

UNIVERSITATEA „POLITEHNICA” TIMIȘOARA
FACULTATEA AUTOMATICĂ ȘI
CALCULATOARE
DEPARTAMENTUL CALCULATOARE

*Sistem informatic pentru
evaluarea genetică a caracterelor
productive la taurine*

- Teză de doctorat -

CONDUCĂTOR ȘTIINȚIFIC,
PROF. DR. IONEL JIAN

UNIV. „POLITEHNICA” TIMIȘOARA BIBLIOTECA CENTRALĂ		
Nr. volum	276.000	
Dulap	369	Lit. A

DOCTORAND,
ing. ILEANA BRUDIU

2005

**Sistem informatic pentru
evaluarea genetică a caracterelor
productive la taurine**

CUPRINS

INTRODUCERE.....	6
1. MODELE MATEMATICE ÎN AMELIORAREA GENETICĂ.....	9
1.1. Sursele de progres genetic.....	9
1.2. Caracterele care fac obiectul ameliorării.....	13
1.2.1. Caractere care exprimă producția de carne la taurine.....	13
1.2.2. Caractere care exprimă producția de lapte la taurine.....	13
1.2.3. Caracterele de exterior ale taurinelor.....	14
1.2.4. Caracterele reproductive ale taurinelor.....	15
1.3. Valoarea de ameliorare și principiile predicției ei.....	17
1.3.1. Predicția valorii de ameliorare prin metoda celor mai mici pătrate	20
1.3.2. Predicția valorii de ameliorare prin metoda indicilor de selecție.....	21
1.3.3. Predicția valorii de ameliorare prin metodologia BLUP	23
1.3.4. Elemente care influențează aplicarea metodologie BLUP.....	31
2. FACTORI CE INFLUENȚEAZĂ VALOAREA FENOTIPICĂ LA VACILE DE LAPTE	32
2.1. Modelul regresional multifactorial	32
2.2. Factori care influențează producția de lapte	36
2.2.1. Stabilirea modelului matematic.....	37
2.2.2. Analiza rezultatelor obținute prin procesul de modelare	41
2.2.3. Calitatea și credibilitatea modelului și a datelor	43
2.2.4. Concluzii	44
2.3. Predicția valorii de ameliorare utilizând modelul propus.	46
2.4. Compararea modelului propus cu modele existente	48
3. IMPLEMENTAREA SISTEMULUI DE AMELIORARE	50
3.1. Structura bazei de date pentru aplicația Ameliorare.....	50
3.2. Modelul datelor.....	53
3.3. Modelul de date relațional.....	54
3.4. Modelul de date orientat obiect.....	59
3.5. Modelul de date relațional-obiectual	62

3.6. Persistența datelor	64
3.7. Caracteristici și cerințe ale aplicațiilor orientate obiect ce lucrează cu baze de date relaționale	66
3.7.1. Modelarea datelor	66
3.7.2. Implementarea trăsăturilor specifice modelului orientat obiect	68
3.7.3. Accesarea datelor	89
3.7.4. Partajarea datelor	99
3.7.5. Implementarea modelului matematic.....	101
3.7.6. Arhitectura software și patternurile arhitecturale folosite în aplicație.....	109
4.CONCLUZII, CONTRIBUȚII PERSONALE ȘI PERSPECTIVE..	116
ANEXA 1.	120
ANEXA 2	122
ANEXA 3	125
ANEXA 4	127
BIBLIOGRAFIE	132

GLOSAR DE TERMENI

BLP	Best Linear Prediction
BLUP	Best Linear Unbiased Prediction
BP	Best Prediction
DAO	Data Access Object
DIM	Day In Milk
FK	Foreign Key
GOP	Gateway-based object persistence
IA	Însămânțare artificială
MVC	Model – View - Controller
OID	Object Identifier
PK	Primary Key
R2	Coefficient de determinare multiplă
SGBDOO	Sisteme de gestiune a bazelor de date orientate obiect
SGBDRO	Sisteme de gestiune a bazelor de date relațional obiectuale
SQL	Structure Query Language
TDM	Test Day Model
UML	Unified Modeling Language

INTRODUCERE

Cercetarea interdisciplinară reprezintă o direcție a cercetării care poate oferi și asigura domeniul de cercetare nebătătorite, cu perspective de satisfacții. Abordarea unei cercetări interdisciplinare, sau chiar multidisciplinare, presupune cunoașterea domeniilor de interferență și posibilitatea efectuării de cercetări în aceste domenii. Lucrarea de față își propune o cercetare multidisciplinară (ameliorarea animalelor - statistică matematică - informatică) al cărui rezultat îl reprezintă un sistem informatic ce permite studierea factorilor ce influențează procesul de evaluare genetică a animalelor.

Progresele înregistrate în ultimele decenii de statistica matematică, genetica populațiilor și genetica cantitativă au impulsionat studiile referitoare la ameliorarea genetică a animalelor.

Teoria ameliorării s-a dezvoltat mai rapid decât baza computațională pe care o solicita aceasta. Deși teoretic fundamentată, folosirea în practică a unor modele complexe în domeniul ameliorării este întârziată din lipsa unei infrastructuri reprezentate de un sistem informațional la nivel național pentru culegerea și stocarea informațiilor din domeniul ameliorării. Asigurarea unor unelte care să permită cercetarea, reprezintă o primă etapă. Pentru specialistul din domeniul IT, stocarea informațiilor înseamnă mai mult decât o colecție organizată de date. El privește acest domeniu sub două aspecte: modul în care aceste informații sunt stocate și modalitatea de a ajunge cât mai rapid și mai simplu la acestea. În cercetarea de față se propune o soluție pentru un domeniu în care volumul mare, natura diversă și timpul mare în care se colectează datele fac anevoioasă luarea deciziilor. Domeniul propus este cel zootehnic, iar decizia ce trebuie luată este alegerea reproducătorului pentru împerechere, în vederea obținerii unui produs care să asigure ameliorarea speciei în direcția dorită. Cum este ales acesta, ne spune teoria ameliorării, iar metodologia alegerii se bazează pe un calcul statistic.

În procesul de predicție a valorii de ameliorare se pornește de la modelul fundamental al geneticii cantitative pe baza căruia s-au dezvoltat mai multe modele matematice funcție de caracteristicile urmărite și de structura datelor experimentale. Având acces la o bază de date cu toate informațiile necesare se pot dezvolta modele matematice ce studiază influența unei multitudini de factori ce pot influența valoarea fenotipică a animalului.

În **primul capitol** al lucrării este prezentat stadiul cunoașterii în ceea ce privește modelele liniare utilizate în ameliorarea genetică a animalelor. Specia aleasă pentru exemplificare este specia bovină. De ce această specie?

Metodologia utilizată pentru calculul valorii de ameliorare este valabilă și pentru alte specii (suină, ovină, etc.), alegerea speciei bovină s-a făcut din considerente economice, alinierea la legislația europeană impunând o evidență strictă și obligatorie a întregului efectiv indiferent de mărimea fermei sau exploatației. Demersurile pentru evidența efectivului din această specie a fost demarat, iar lucrarea de față completează informațiile stricte de evidență cu cele natură să asigure o valorificare a acestora în scopuri economice.

Sunt prezentate sursele de progres genetic precum și caracterele care fac obiectul ameliorării la specia bovină.

Valoarea de ameliorare reprezintă criteriul corect de alegere a animalelor de reproducție în selecția animalelor. Ea nu poate fi măsurată direct pe animale, poate fi dedusă din valorile fenotipice ale animalelor și/sau ale rudelor acestora.

Predicția valorii de ameliorare este o problemă statistico-matematică, având ca fundament elementele de bază ale geneticii cantitative. Pornind de la principiile teoretice ale regresiei și corelației pot fi deduse relațiile necesare predicției valorii de ameliorare și a parametrilor ei. Sunt prezentate metodele de calcul utilizate în predicția valorii de ameliorare precum și modelele clasice utilizate în evaluarea genetică. Este prezentată de asemenea Modelul testului zilei de control, model utilizat în prezent în evaluarea genetică a caracterelor productive la vacile de lapte.

Capitolul al doilea al lucrării prezintă un studiu cu privire la factorii ce influențează caracterele productive ale vacilor de lapte, finalizat cu propunerea unui model liniar multifactorial. Modelul propus introduce în studiu factori ce caracterizează curba de lactație ale vacilor de lapte, aceasta reprezentând o noutate față de modelele clasice, apropiind modelul propus de cel al testului zilei de control.

Analiza modelului s-a făcut pe baza testelor statistice, interpretarea rezultatelor a condus la concluzii ce confirmă validitatea acestuia. Pe baza modelului propus au fost calculate valorile de ameliorare ale taurilor reproducători, rezultatele obținute au fost comparate cu ce obținute utilizând un model clasic, obținându-se rezultate comparativ asemănătoare.

Calitatea și credibilitatea modelului este susținută de rezultatele obținute în urma analizei și interpretării testelor statistice asupra modelului precum și în urma comparării rezultatelor obținute pentru valorile de ameliorare. Datele de test utilizate în studiu au fost preluate din activitatea de cercetare a facultății de Medicină Veterinară Timișoara.

Interdisciplinaritatea cercetării a fost necesară pentru a asigura unelte de lucru pentru studiu modelului propus. A fost utilizată baza de date și aplicația „Ameliorare”, aplicație ce permite întreținerea bazei de date și obținerea soluțiilor pentru predicția valorii de ameliorare.

În **capitolul trei** al lucrării se prezintă cercetările efectuate pentru realizarea bazei de date și a aplicației „Ameliorare”.

Studiu asupra istoriei bazelor de date este centrat în jurul problemei modelării datelor. Alegerea modelului utilizat (relațional, obiectual, relațional-obiectual) a constituit o primă etapă a cercetării pentru soluționarea acestei probleme. Tendințele actuale în programare și perspectiva viitorului apropiat în domeniul bazelor de date a influențat alegerea scrierii aplicației utilizând un limbaj de programare orientat obiect. Sunt prezentate modalitățile de realizare prin soft de obiecte persistente precum și caracteristici și cerințe aplicațiilor orientate obiect ce lucrează cu baze de date relaționale. Calculul valorilor de ameliorare se realizează prin metodologia BLUP, implementat în aplicație., sunt de asemenea prezentate arhitectura software și patternurile arhitecturale folosite în aplicație

Stabilirea structurii bazei de date s-a făcut în urma unei analize riguroase utilizând limbajul UML. Serverul de baze de date folosit este Microsoft SQL Server 2000, iar aplicația implementată a fost scrisă în limbajul de programare C#.NET.

Capitolul patru al lucrării prezintă concluziile, contribuțiile personale și perspective continuării și valorificării cercetării.

Modelarea matematică permite o analiză a factorilor ce influențează valoarea fenotipică la taurine, statistica matematică reprezintă instrumentul care validează modelul

propus, iar informatica uneltele necesare efectuării cercetării în domeniul ameliorării pe baza modelelor existente, a modelului propus sau a celor care vor fi dezvoltate.

Rezultatul acestei cercetări interdisciplinare reprezintă un sistem informatic ce propune pe lângă un model matematic utilizat în studierea factorilor ce influențează caracterele productive la vacile de lapte, o bază de date și o aplicație ce permite verificarea modelului precum și perspectiva dezvoltării de aplicații din domeniul ameliorării animalelor, crearea de facilități pentru procesul de învățământ și cercetare.

Lucrarea prezintă în **ANEXE**, rasele de taurine de la noi din țară prezente în programul de ameliorare, lista caracterelor care fac obiectul ameliorării la specia bovină, parametri tehnici ai programului de ameliorare și diagramele UML folosite în stabilirea structurii bazei de date și a aplicației.

1. MODELE MATEMATICE ÎN AMELIORAREA GENETICĂ

1.1. Sursele de progres genetic

În exploatarea taurinelor se urmărește obținerea eficienței economice cât mai mari. Acesta se realizează prin îmbunătățirea și creșterea continuă caracterelor productive (producția de lapte și carne).

Caracterele productive sunt caractere cantitative, ele sunt determinate atât de genotip - baza ereditară, cât și de condițiile de mediu în care se dezvoltă acestea. Baza ereditară condiționează potențialul genetic, iar condițiile de mediu pot favoriza sau inhiba capacitatea productivă și producția efectiv realizată pentru un anumit caracter, în fapt exprimarea fenotipică a acestuia.

Sporirea producției individuale a taurinelor se poate realiza pe două căi individuale:

- optimizarea factorilor de mediu;
- mărirea potențialului genetic productiv.

Condițiile de mediu se referă la tehnologia de creștere și exploatare a taurinelor și se realizează prin dirijarea furajării, mulgerii, a lucrărilor de îngrijire, a microclimatului din adăposturi etc.

Prin **optimizarea factorilor de mediu**, producția poate crește până la nivelul maxim al potențialului genetic al animalului. Peste această limită, pe seama modificării în continuare a factorilor de mediu, nu se pot obține sporuri de producție.

Potențialul genetic productiv al taurinelor depinde de construcția lor anatomică și de particularitățile funcționale ale diferitelor sisteme și organe, ambele fiind controlate în toată ontogenza de genotipul animalelor.

Sporirea permanentă și ritmică a valorilor fenotipice productive se poate realiza numai prin modificarea structurii genetice, respectiv îmbunătățirea potențialului genetic la generațiile următoare. Acest lucru îl reprezintă **ameliorarea**.

Ameliorarea presupune un complex de sisteme și metode care aplicate, modifică structura genetică a efectivelor de animale în generații succesive, în sensul îmbunătățirii potențialului genetic al însușirilor morfoproductive [Vi88]

Ameliorarea animalelor, una din cele mai importante laturi ale zootehniei, poate fi considerată ca fiind rezultanta a trei domenii de activitate strâns legate între ele: fiziologia reproducției, genetica și ameliorarea propriu-zisă, adică aplicarea principiilor geneticii pentru găsirea celor mai bune procedee și metode de îmbunătățire a populațiilor de animale.

Scopul principal al ameliorării îl constituie maximizarea progresului genetic pe unitatea de timp și pe unitatea de costuri. Direcția în care se acționează o constituie mărirea gradului de valorificare a condițiilor de hrănire și de creștere intensive practicate în prezent. Legat de aceasta, problema asigurării informațiilor necesare și a modului de utilizare a acestora capătă o importanță deosebită.

Progresul în orice domeniu de activitate depinde de modul în care știm să utilizăm multitudinea de informații, de viteza cu care le putem înțelege, interpreta, de a le transmite și de a lua decizii având la bază aceste cunoștințe.

Decizii pot fi luate și fără a dispune de informații necesare, dar acestea vor fi însă în majoritatea cazurilor decizii eronate, uneori cu urmări grave, iar, analiza desfășurării unor fenomene duce la concluzia că principala cauză a neajunsurilor înregistrate constă în necunoașterea la timp a informațiilor necesare. O etapă laborioasă este legată de culegerea și selectarea informației, apoi, stabilirea și reținerea acelor informații care asigură un grad de certitudine corespunzător în luarea deciziilor. Acuratețea datelor primare constituie o condiție obligatorie pentru trecerea la prelucrarea și analiza lor, o informație falsă, perturbă rezultatele obținute, iar deciziile luate pot fi împotriva propriilor interese.

Scopul ameliorării îl constituie îmbunătățirii potențialul genetic al însușirilor morfoproductive ale animalelor.

Însușirile morfoproductive ale taurinelor se obțin în trei etape.

- I. dobândirea potențialul genetic al noii generații. Aceasta se obține în urma selecției, migrației și a dirijării împerecherilor;
- II. dobândirea capacității productive, care în fapt constă în realizarea potențialității fenotipice productive prin creșterea dirijată a tineretului;
- III. dobândirea producție efectiv realizate, adică obținerea producției reale sau fenotipice prin asigurarea condițiilor de mediu în perioada exploatarei.

Prima etapă în dobândirea potențialul genetic al noii generații presupune „selecția, migrația și dirijarea împerecherilor” [Ge98].

Selecția artificială constituie unul din cei mai importanți factori de realizare a progresului genetic, ea se bazează pe cunoașterea, dirijarea variabilității și eredității animalelor și modificarea structurii genetice a populației, prin sporirea proporției genelor cu efecte fenotipice dorite și reducerea sau înlăturarea genelor cu efecte nedorite [Ge97].

Prin selecție nu se creează noi gene și nici nu se modifică cele existente, dar, ca urmare a discriminării reproductive, se multiplică în populație genotipurile cu combinațiile cele mai favorabile de gene sau a genelor provenite prin mutație .

Sursele de progres genetic sunt [Ge98]:

- taurii amelioratori,
- selecția vacilor primipare
- reforma selectivă la vaci adulte.

Operațiune premergătoare selecției o constituie identificarea genotipurilor superioare din populația supusă selecției. Sursele de informație în vederea selecției sunt fenotipice și genotipice, ele stau la baza stabilirii valorii de ameliorare a reproducătorilor, după care se efectuează ierarhizarea populației supusă selecției.

Selecția fenotipică se aplică în mod curent la vaci iar **selecția genotipică**, datorită costurilor mari se aplică cu precădere la taurii folosiți la însămânțări artificiale.

Ierarhizarea reproducătorilor reținuți la reproducție se face în funcție de valoarea de ameliorare obținută prin selecția individuală [Ge98].

Selecția individuală impune aprecierea indivizilor concomitent după criterii fenotipice, reprezentate de performanțele proprii și criterii genotipice, reprezentate de performanțele ascendenților rudelor colaterale sau descendenților. Actualmente, în acțiunea de selecție a animalelor de rentă se utilizează deferite procedee, forme și metode, în funcție de posibilități, obiective, însușirile urmărite și heritabilitatea acestora.

În perioada de **creștere și exploatare**, în condițiile asigurării de factori optimi de mediu, se poate obține o producție reală cel mult egală cu potențialul genetic productiv.

În procesul de **ameliorare**, direcțiile sunt orientate pe cele două producții principale ale taurinelor: carnea și laptele, acestea diferind de la o zonă geografică la alta, de la o țară la alta, urmărindu-se realizarea unui profit maxim în exploatare [Dr79]. La orientarea direcției de ameliorare a taurinelor se ține seama de structura de rasă, solicitările pieței, posibilitățile de asigurare a cerințelor biologice specifice materialului biologic care se exploatează, preferințele crescătorilor, etc.

Caracterele care au făcut obiectul ameliorării, la începuturile acestei activități, au fost exclusiv orientate pe producția de carne și lapte. În ultima perioadă de timp au apărut noi caractere care sunt supuse ameliorării.

Crescătorul de taurine urmărește să obțină animale cu potențial productiv ridicat, longevive, rezistente și bine adaptate la mediu de exploatare, precoce, bine dezvoltate și conformate, ușor de exploatat și economice. Aceste deziderate pot fi îndeplinite prin utilizarea cu discernământ a caracterelor de ameliorat. Caracterele care necesită a fi ameliorate sunt numeroase, putând fi grupate în: caractere productive (de lapte sau carne), caractere morfologice și caractere de exploatare.

Numărul caracterelor urmărite este mare (ANEXA 1), ele caracterizează însușirile animalului, și evident, în procesul de ameliorare a speciei interesează cunoașterea acestora.

Se disting următoarele categorii de caractere:

- de carne;
- de lapte;
- reproductive;
- de exterior;
- rezistența la îmbolnăviri;
- producția energetică;
- de culoare și particularități de culoare.

Pentru majoritatea însușirilor sunt cunoscuți **coeficienții de heritabilitate** (h^2 = raportul dintre variația genetică aditivă și variația fenotipică totală), iar în funcția de valoarea acestora se stabilește sistemul de ameliorare ce se va practica.

Pentru valori ale h^2 de peste 50%, ponderea determinării caracterelor respective revine varianței genetice aditive și ca urmare, ameliorarea lor prin selecție dă rezultate certe. Pentru însușirile cu heritabilitate scăzută (30%), progresul genetic se realizează numai dacă în selecție se ia în considerare pe lângă fenotipul propriu și fenotipul ascendenților, a rudelor colaterale și a descendenților.

Alegerea caracterelor care fac obiectul ameliorării se face din considerente economice și ameliorative, iar activitatea de urmărire a acestor caractere o reprezintă **testare sau controlul performanțelor productive.**

Controlul performanțelor se realizează în scopul obținerii informațiilor necesare în vederea desfășurării întregii activități de selecție și ameliorare, precum și în scopul obținerii de date obiective necesare în luarea celor mai corecte măsuri organizatorice pentru întreaga activitate de producție.

Datele recoltate prin controlul performanțelor, utilizate în scop de selecție, trebuie să fie suficient de precise și să permită prin metode statistico-matematice, eliminarea influențelor de mediu din manifestarea fenotipică a acestor caractere.

1.2. Caracterele care fac obiectul ameliorării

1.2.1. Caractere care exprimă producția de carne la taurine

Carnea de taurine constituie un aliment complet, valoros sub raport nutritiv, biologic și igienic, relativ convenabil economic, care are un rol particular în viața și sănătatea oamenilor ca urmare există și se justifică preocuparea constantă a amelioratorilor pentru îmbunătățirea continuă a aptitudinilor de carne la taurine. Caracterele de carne se apreciază atât pe animalul viu cât și pe cel sacrificat[Ge98].

Caracterele ce pot fi identificate și luate în considerare pe animalul viu sunt următoarele:

- greutatea și dimensiunile corporale;
- viteza absolută de creștere;
- capacitatea de valorificare a hranei;
- conformația corporală;
- calitatea comercială a animalelor.

Fiecare dintre aceste caractere au un determinism genetic variabil și este influențat de specie, tipul morfo-fiziologic, morfo-productiv, rasă și vârstă.

Masa (greutatea) corporală constituie obiectiv și caracter de selecție.

1.2.2. Caractere care exprimă producția de lapte la taurine

Caracterele de lapte cuprind elemente cantitative (lapte, grăsime, proteine, substanță utilă) și calitative (raport proteine: grăsime, raport între fracțiunile proteice)[Ge98].

Reglementările internaționale prevăd exprimarea producției de lapte a taurinelor drept cantitatea de lapte, cantitatea de grăsime pură și cantitatea de proteină pură pentru 305 zile de lactație ale vacii respective. Aceste caractere sunt singurele care sunt utilizate în munca de selecție și ameliorare.

În aprecierea vacilor din rasele mixte și de lapte, principalul criteriu în ierarhizarea acestora în vederea alcătuirii nucleelor de selecție, nominalizarea vacilor mame de tauri și a candidatelor mame de taur îl constituie producția de lapte. Stabilirea acesteia se face prin control individual periodic în ziua de control și extrapolarea acesteia la nivelul întregii perioade de control. Producția de lapte este dependentă de ordinea lactației, de vârsta primei fătări, în condițiile întreținerii normale.

În ultimul timp se pune un accent deosebit pe **calitatea** producției de lapte. Astfel, unul din principalele criterii de selecție a vacilor cu lapte îl reprezintă conținutul de proteină al acestuia și, mai mult, cantitatea cazeinei din lapte.

Producția de lapte este foarte diferită la rasele de taurine în funcție de:

- gradul de ameliorare genetică,
- direcția de exploatare,
- condițiile de exploatare.

Condițiile de exploatare pot influența semnificativ producția de lapte ale taurinelor din aceeași rasă, iar diferențele se datorează atât factorilor de mediu cât și nivelelor genetice diferite ale efectivelor din aceeași fermă.

Caracterele de lapte au heritabilități diferite, existând corelații între acestea. De exemplu, între elementele cantitative, există corelații (pozitive) foarte strânse (lapte – grăsime 0,85 – 0,90, proteine - grăsime 0,70 – 0,95), între elementele cantitative și cele calitative, corelații negative slabe, toate acestea conducând la concluzia că se poate simplifica și eficientiza aprecierea aptitudinilor prin renunțarea la unele din ele și reținerea celor mai importante.

1.2.3. Caracterele de exterior ale taurinelor

În activitatea de selecție, organismul animal nu trebuie privit ca o simplă însușire de însușiri, ci ca o îmbinare sau, mai exact, ca o integrare a unui însemnat număr de caractere care se realizează după legile eredității și dezvoltării biologice, sub dirijarea sistemului neuro-hormonal. Exteriorul unui animal reprezintă, de fapt, manifestarea posibilităților sale ereditare în condițiile concrete ale mediului înconjurător, iar, ca urmare, forma exterioră a eredității – fenotipul – diferă în funcție de calitatea și modul de reacție al genotipului.

Aprecierea exteriorului la taurine permite depistarea și eliminarea din lotul de reproducție a animalelor cu defecte și malformații care influențează productivitatea și economicitatea exploatarei, permite aprecierea capacității de rezistență a animalelor la efortul de producție și de adaptare la tehnologiile de creștere. De asemenea, exteriorul sau unele însușiri de exterior, se corelează cu producțiile taurinelor. Astfel, principalii indici ai producției de carne se corelează relativ puternic cu însușirile de exterior. Între conformație și producția de lapte nu există relații antagoniste dar coeficientul de corelație genetică este de cele mai multe ori, mic. Producția de lapte, însă este în relație strânsă cu însușirile morfologice și fiziologice ale ugerului. Aprecierea conformației corporale și a constituției au o importanță și semnificație majoră în evaluarea aptitudinilor morfoproductive ale taurinelor constituindu-se în criteriu de selecție pentru materialul de reproducție[Al88].

Bonitatea este operațiunea tehnică de apreciere complexă a taurinelor pe baza însușirilor de exterior, a performanțelor productive proprii precum și a performanțelor rudelor, în scopul stabilirii valorii lor zootehnice.

Aprecierea conformației corporale se face la ora actuală după o metodă considerată mai obiectivă și anume „**descrierea lineară a caracterelor de exterior**”. Dacă inițial, această metodă a fost utilizată pentru aprecierea individuală a primiparelor pe parcursul primelor 100 de zile de lactație, în vederea reținerii sau eliminării de la prăsilă, metoda este utilizată îndeosebi pentru estimarea valorii de ameliorare a taurilor pe baza caracterelor de exterior ale fiicelor în cadrul lucrărilor de testare pe descendenți, precum și la alegerea vacilor – mame de taur și nominalizarea împerecherilor.

În descrierea lineară a exteriorului se pornește de la premisa că majoritatea caracterelor cu importanță economică și funcțională în exploatare se analizează în mod independent, iar în apreciere se folosește o scală numerică, cu note de la 1 la 9, care semnifică manifestarea fenotipică de la o extremă biologică la cealaltă, nota 5 reprezentând, de regulă situația intermediară întâlnită, ca medie pe populație, cu excepția câtorva caractere la care aceasta este considerată optimă.

Sistemul de descriere lineară a exteriorului la taurine se bazează pe următoarele principii:

- Aprecierea caracterelor de exterior cu importanță economică sau funcțională pentru taurine se face independent între acestea.
- Pentru aprecierea fiecărui caracter se utilizează o scară numerică, cuantificându-se prin cifre manifestarea fenotipică a unui caracter de la o extremă la cealaltă. La acest tip apreciere, bonitorul nu mai este pus în situația de a compara manifestarea fenotipică a caracterului cu tipul ideal din memorie, ci este desemnat numai pentru cuantificarea sub formă de cifre a manifestării fenotipice a caracterului respectiv.
- Datele obținute se supun prelucrării computerizate în vederea estimării valorii de ameliorare prin metodologia BLUP.

Caracterele de exterior care fac obiectul descrierii lineare la taurinele mixte și de lapte, în număr de 16- 18, în funcție de rasă, sunt împărțite pe grupe de caractere.

Heritabilitatea caracterelor (h^2) este o proprietate a populațiilor, dependentă de acțiunea genelor, dar influențată de variația datorată mediului. Cea mai importantă funcție a heritabilității în studiul genetic al caracterelor cantitative o reprezintă gradul de siguranță ce se poate atribui, prin intermediul ei, valorii fenotipice în stabilirea capacității de ameliorare a reproducătorilor, ea arată precizia selecției, implicit posibilitatea realizării de progrese mai rapide la un caracter sau la altul, în succesiunea generațiilor[Sa96].

În ANEXA 2 sunt prezentate pentru trei rase de taurine (**Bălțată românească** – Fleckvich – Simmental, **Brună** – Braunvich și **Bălțată cu negru** – Holstein – Friză), caractere incluse în sistemul de descriere lineară a exteriorului precum și valorile medii ale heritabilității acestora, iar în ANEXA 3, parametri tehnici ai programului de ameliorare.

În urma bonitării prin metoda descrierii lineară a exteriorului, funcție de punctajul obținut are loc clasificarea respectiv, încadrarea în clase de calitate.

Descrierea lineară a caracterelor de exterior are ca scop găsirea partenerilor celor mai potriviți pentru împerecheri, în vederea obținerii de noi generații de descendenți, superioare celor din care provin, eliminând punctele slabe și defectele constatate și înlocuindu-le cu caractere dorite[A188].

1.2.4. Caracterele reproductive ale taurinelor

Caracterele reproductive ale taurinelor reprezintă o condiție esențială în procesul de ameliorare. Specialiștii amintesc faptul că aceste caractere trebuie privite ca un complex de caractere, care, fiecare luate separat, reprezintă o mică influență de ameliorare genetică, însă acțiunea lor cumulativă are o importanță hotărâtoare[Hu00].

Principalele caractere reproductive care se iau în considerare la taurine sunt:

- eficiența însămânțării;
- intensitatea activității de reproducție;
- natalitatea;
- prolificitatea.

Se disting caractere reproductive specifice femelelor și masculilor, de asemenea o parte dintre acestea caractere sunt determinate pur genetic altele, în urma unui complex de factori de mediu și genetici.

Pentru ameliorarea genetică interesează în mod special caracterele legate de eficiența însămânțării sau a monteii [Sa01]. Acestea sunt:

- natalitatea
- fecunditatea
- rata non-return
- service-period (intervalul dintre parturiție și monta fecundată)
- indexul de însămânțare (numărul de doze utilizate de material consumat pentru realizarea unei gestații)
- calving-interval (durata de timp dintre două parturiții succesive)

Cu toate că aceste caractere au un determinism genetic extrem de scăzut, aceste caractere sunt luate în considerare în programul de ameliorare pentru rasa respectivă.

Principiile care fundamentează programul de ameliorare al taurinelor sunt multiple și diverse, alegerea de criterii raționale a caracterelor care determină obiectul selecției iau în considerare atât valoarea economică relativă a caracterelor cât și heritabilitatea și corelația genetică dintre caractere. Pentru o populație dată, parametri demografici și de selecție, împreună cu parametri genetici și fenotipici ai caracterelor de selecție, determină efectiv progresul genetic realizat.

1.3. Valoarea de ameliorare și principiile predicției ei

Ameliorarea genetică a animalelor este definită ca un proces de modificare dirijată a potențialului productiv, a caracterelor ereditare, a genofondului populațiilor de animale domestice, în direcția dorită de om. [Ge98]

Deoarece unitatea de lucru a ameliorării este populația și nu individul (genotipul individului nu poate fi modificat după formarea zigotului), singura posibilitate practică de a influența baza ereditară dorită este aceea de a interveni înainte de formarea zigotului, la nivelul populației, influențând astfel generația următoare și nu cea actuală. Ameliorarea se poate defini astfel ca un proces de organizare a reproducerii populației, astfel încât fiecare generație să fie mai bună decât precedentă.

Pentru a realiza o îmbunătățire genetică a populațiilor, trebuie să se selecționeze în generația curentă acele animale care au cea mai mare valoare genetică reală pentru caracterele importante economice. Valoarea genetică reală nu poate fi măsurată direct pe animale, dar poate fi dedusă din valorile fenotipice ale animalelor și/sau ale rudelor acestora.

Valoarea fenotipică a unui animal (P) are următoarele componente:

- **Valoarea genotipică (G)** - este determinată de efectul genelor și al interacțiunii genetice între genele care intră în genotipul caracterului respectiv. Se compune din:
 - Valoarea de ameliorare(A)
 - Valoarea datorată dominației (D)
 - Valoarea datorată epistaziei (I)

$$G = A+D+I$$

- **Valoarea datorată mediului (M)** – este creată de două surse:
 - Mg - efectul de mediu general sau permanent
 - Ms - efectul de mediu special sau temporar

$$M = M_g + M_s$$

Astfel valoarea fenotipică P poate fi scrisă astfel:

$$P = A+D+I+ M_g + M_s$$

În această relație, elementele din partea dreaptă sunt reciproc necorelate. Valoarea lui P este afectată de mediul special Ms, diferit de la un individ la altul, astfel, valorile brute (P) nu sunt comparabile și indivizii nu pot fi aleși la reproducție direct pe baza acestora.

Exprimăm valoarea fenotipică (P) și componentele sale ca abatere de la mediile lor:

$$P - \mu(P) = A - \mu(A) + D - \mu(D) + I - \mu(I) + M_g - \mu(M_g) + M_s - \mu(M_s)$$

Iar cu notațiile:

$$y = P$$

$$a = A - \mu(A)$$

$$d = D - \mu(D)$$

$$i = I - \mu(I)$$

$$p = M_g - \mu(M_g)$$

$$e = M_s - \mu(M_s)$$

Modelul fundamental al geneticii cantitative poate fi scris în termeni de deviații de la medie:

$$y = \mu + a + d + i + p + e$$

μ - media valorii fenotipice,

a, d, i, p, e au media zero și sunt necorelate.

Valoarea de ameliorare a unui individ (A) poate fi exprimată ca medie a valorilor de ameliorare ale părinților săi:

$$A = \frac{A_T + A_M}{2},$$

Performanța unui descendent oarecare este:

$$P = \frac{A_T + A_M}{2} + D + I + M_g + M_s$$

În condițiile în care numărul partenerilor este suficient de mare și au fost luați la întâmplare din populație: D, I, M_g , și M_s se anulează reciproc, rămânând în evidență:

$$\bar{P} = \frac{A_T + A_M}{2} + 0 + 0 + 0 + 0$$

Ceea ce înseamnă că la nivelul populației, media valorilor fenotipice este egală cu media valorilor de ameliorare.

În ipoteza în care femelele partenere constituie o probă reprezentativă din populație, respectiv au aceeași medie ca întreaga populație, rezultă:

$$\bar{P} = \frac{A_T + \mu}{2}$$

Valoarea de ameliorare a unui individ reprezintă de două ori abaterea descendenților lui de la media populației:

$$A_T = 2 \times (\bar{P} - \mu)$$

Afirmația este valabilă când numărul descendenților este infinit și partenerii reprezintă însăși populația, teoretic infinită.

În realitate, valoarea de ameliorare nu poate fi măsurată efectiv, deoarece nu există un număr suficient de mare de descendenți cărora să le fie făcută media performanțelor, având în vedere totalitatea combinațiilor gametice posibile, însă poate fi aproximată.

Statistica matematică ajută la aproximarea valorii de ameliorare reale, pe baza unui număr finit de descendenți. Aprecierea are un caracter probabilistic, în sensul că pe baza numărului descendenților obținuți până la un moment dat, este posibil să se prevadă valoarea de ameliorare a viitorilor produși.

Modele de analiză a datelor primare

Sistemele, ca realitate obiectivă, apar în procesul cunoașterii, ele pot fi înțelese și studiate prin modelare. Modelarea este procesul de studiere a comportamentului sistemelor reale cu ajutorul modelelor. Modelul reprezintă forma simplificată a unui sistem real, păstrând însă ceea ce are el esențial, din punctul de vedere a ceea ce se dorește a fi evidențiat.

Modelul matematic poate fi unul din tipurile de modele utilizate în procesul cunoașterii. Modelarea matematică a sistemelor biologice se numește biometrie, iar modelele utilizate se numesc modele biometrice[Ca00].

Luând în studiu caracterele productive a unei vaci, de exemplu cantitatea de lapte constatăm ca ea este influențată de potențialul ei genetic, vârsta la fătare, sezonul în care a fătata, durata repausului mamar, etc.

În stabilirea modelului matematic trebuie să se stabilească variabila dependentă, precum și variabilele independente, în funcție de gradul în care variația variabilelor independente explică variația variabilei dependente, modelele pot fi deterministe sau stohastice. Modelele utilizate în studiu sistemelor biologice sunt **modele stohastice** deoarece doar o parte din variația variabilei dependente poate fi explicată prin variația variabilelor independente, restul de variație fiind explicată prin variația aflată în componenta eroare[Gr97].

Calitatea oricărei analize statistice este apreciată prin modelul ales să descrie cel mai bine datele primare, numite și observații sau performanțe.

Pornind de la modelul liniar general:

$$y_i = \sum w_{j,i} \beta_j + e_i,$$

- y_i - valoarea observată,
- $w_{j,i}$ - coeficienți constanți, cunoscuți,
- β_j - parametri necunoscuți,
- e_i - eroarea corespunzătoare valorii y_i .

În scriere matriceală: $y = W\beta + e$.

Parametri necunoscuți sunt ca urmare a efectelor fixe, sau aleatoare. Ca urmare, în modelul liniar, termenul $W\beta$ se poate scrie ca suma a doi termeni, ce reflectă această situație.

Modelul liniar general sub formă matriceală este:

$$y = Xb + Zu + e$$

Factorul fix este acel factor cuprins în analiză căruia i se cunosc toate nivelurile. Factorul aleator este factorul al cărui niveluri cuprinse în analiză constituie o probă extrasă la întâmplare dintr-o populație de niveluri normal distribuită, cu o medie și o varianță precizată.

În aceste condiții, variabilele datorate efectelor fixe pot fi estimate, iar cele datorate efectelor aleatoare pot fi predictate.

Dacă modelul liniar prezentat descrie cantitatea de lapte pe care o dă o vacă, constatăm ca ea este influențată de potențialul ei genetic, vârsta la fătare, sezonul în care a fătat, durata repausului mamar, etc. Valoarea fenotipică (producția de lapte) depinde de componenta genetică, care la rândul ei este determinată de valoarea de ameliorare.

Cunoașterea valorii de ameliorare în ameliorare nu constituie un scop în sine, ci un mijloc pentru clasificarea animalelor în vederea selecției. Valoarea de ameliorare este un factor ce se datorează efectelor aleatoare, ca urmare ea poate fi doar predictată.

Predicția valorii de ameliorare necesită cunoașterea următoarelor elemente:

- informația fenotipică reprezentată de performanța proprie;
- relațiile genetice între indivizi (coeficienți de înrudire);
- cunoașterea formei de distribuție (repartiție) a valorilor de ameliorare (a) și a performanțelor observate (v);
- cunoașterea mediei populației (μ) pentru elementul vizat;
- cunoașterea structurii varianțelor și covarianțelor la nivel de populație, elemente care stau la baza estimării parametrilor genetici: heritabilitate (h^2), repetabilitate (R), și corelația genetică r_g ;

În funcție de cunoașterea acestor elemente există trei posibilități în vederea predicției valorii de ameliorare:

1. **metoda celor mai mici pătrate**, cunoscută sub numele de metoda B.P. (Best Prediction = cea mai bună predicție),
2. **metoda indicilor de selecție**, cunoscută sub denumirea B.L.P. (Best Linear Prediction = cea mai bună predicție liniară);
3. **metodologia B.L.U.P.**, (Best Linear Unbiased Prediction = cea mai bună predicție liniară nedeplasată), o combinație a primelor două metode.

1.3.1. Predicția valorii de ameliorare prin metoda celor mai mici pătrate

Aplicarea acestei metode presupune cunoscute elementele:

1. forma de distribuție a valorilor de ameliorare (a) și a performanțelor observate (y);
2. media populației pentru aceste două variabile;
3. structura varianțelor și covarianțelor la nivel de populație.

Principiul metodei constă în condiția ca suma abaterilor la pătrat să fie minimă:

$$Q = \sum_{i=1}^n (y_i - y_k)^2 \rightarrow \text{minim}$$

condiție ce este realizată pentru $y_k = \bar{y}$.

Pornind de la modelul liniar:

$$y_{ijk} = \mu + F_i + T_j + e_{ijk}$$

unde :

y_{ijk} - performanța observată a vacii k , $k = 0, \dots, n_j$;

μ - media populației

F_i - efectul fix al fermei i , $i = 1, \dots, p$;

T_j - efectul fix al taurului $j, j= 1, \dots, q$;

e_{ijk} - efectul aleator al altor factori decât cei analizați, având distribuția normală cu media 0 și varianța σ^2 .

Aplicând condiția ca suma abaterilor să fie minimă asupra expresiei modelului, prin derivare parțială a expresiei matematice, se obțin ecuațiile normale și apoi soluțiile pentru fiecare efect fix (efectul fermei și efectul taurului).

Evaluarea reproducătorilor prin metoda celor mai mici pătrate se face în ipoteza că T este fixat (efectul taurului), ceea ce în practică nu este adevărat.

Deoarece varianța erorilor evaluării genetice nu este minimă, există tendința ca metoda celor mai mici pătrate să supraevalueze sau să subestimeze reproducătorii care au o cantitate mai redusă de informații.

1.3.2. Predicția valorii de ameliorare prin metoda indicilor de selecție

Sunt situații practice când nu se cunoaște forma de distribuție a valorilor de ameliorare (a) și a valorilor fenotipice (y). În aceste condiții metoda celor mai mici pătrate nu poate fi folosită și se propune o relație liniară între a și y .

Pornind de la ecuația regresiei simple (pentru cazuri mai complicate regresia multiplă):

$$y = a + b \cdot X,$$

variabila dependentă (y) poate fi prognozată pe baza lui X (variabilă independentă), exprimată ca abatere de la medie, după relația:

$$\hat{y} = \bar{y} + b(X - \bar{X})$$

Notând :

$X = P$ (performanța)

\hat{A} valoarea de ameliorare estimată,

$$b = \frac{Cov(A, P)}{VarP} = \frac{VarA}{VarP} = h^2$$

Heritabilitatea = h^2 , raportul dintre variația genetică aditivă și variația fenotipică totală)

$$\text{În notație matriceală } b = \frac{Cov(A, P)}{VarP} = \frac{C}{V} = C'V^{-1}$$

A – matricea relațiilor genetice aditive dintre indivizii analizați, C' transpusa matricei varianță - covarianțelor genetică aditivă dintre indivizii analizați și V^{-1} , inversa matricei varianțelor performanțelor.

Se obține relația clasică în ameliorarea animalelor:

$$\hat{A} = \mu_A + h^2 \times (P - \mu_P)$$

cu μ_A media performanțelor estimate și

μ_P media performanțelor observate.

Deoarece $\mu_p = \mu_A = \mu$, atunci, $\hat{A} = \mu + h^2 \times (P - \mu)$, iar exprimată ca abatere de la media populației, valoarea de ameliorare a unui individ oarecare i se poate scrie:

$$\hat{a}_i = h^2 \times (y - \mu) ; \text{ cu } h^2 = C'V^{-1}$$

Observăm că pentru a calcula valoarea de ameliorare este necesară cunoașterea mediei populației (μ) și a coeficientului de heritabilitate (h^2) pentru performanța respectivă.

În practică, media populației (μ) este înlocuită cu performanța medie a contemporanilor (\bar{y}_c), deci media unei probe din populație, ca urmare valorile de ameliorare obținute suferă o deplasare egală cu diferența dintre media adevărată a populației și media unei probe din populație (media grupului de contemporani).

Astfel :

$$y - \mu = (y - \bar{y}_c) + (\bar{y}_c - \mu)$$

$(\bar{y}_c - \mu)$ reprezintă cantitatea de deplasare care afectează valoarea de ameliorare.

Metoda indicilor de selecție (BLP - *Best Linear Prediction*, cea mai bună predicție liniară), furnizează valori de ameliorare deplasate sau cel mult asimptotic nedepasate (cantitatea de deplasare descrește pe măsură ce mărimea probei crește, tinzând către mărimea populației). Cele enunțate constituie motivul pentru care metoda BLP a fost înlocuită cu metodologia BLUP.

În funcție de cantitatea de informație luată în considerare, indicii de selecție se clasifică:

- a. Indici de selecție pentru un singur caracter. Se folosesc ca surse de informație performanțele individului analizat și ale rudelor sale, pentru caracterul considerat;
- b. Indici de selecție pe mai multe caractere măsurate pe candidații la selecție. În această situație sursele de informație sunt performanțele individului analizat;
- c. Indici de selecție pe mai multe caractere măsurate pe candidatul la selecție și pe diferite tipuri de rude.

Limitele indicilor de selecție

În multe situații practice, indicii de selecție nu pot rezolva problemele referitoare la evaluarea genetică a animalelor din următoarele motive [Gr97].

1. BLP presupune cunoscută media y . De exemplu, în cazul folosirii reproducătorilor masculi la însămânțări artificiale, trebuie să se cunoască mediile anului, sezonului și grupului, chiar dacă nu există date anterioare. Soluția la această problemă o oferă metodologia BLUP;
2. În unele cazuri varianțele și covarianțele sunt necunoscute;
3. În cazul folosirii reproducătorilor la însămânțări artificiale, datele obținute sunt grupate pe un număr mare de subclase și acestea pot fi nebalansate. În această situație metoda indicilor de selecție nu poate fi aplicată.

1.3.3. Predicția valorii de ameliorare prin metodologia BLUP

Tehnicile de evaluare genetică au evoluat constant în ultimele decenii sub influența simultană a progresului în statistică, informatică și genetică cantitativă.

Teoria clasică a indicilor de selecție consideră efectele de mediu (efecte fixe) cunoscute fără eroare. În cele mai multe cazuri însă efectele fixe au o influență reală asupra performanțelor și nu sunt cunoscute fără eroare. De exemplu, fiicele taurilor aflați în testare pot încheia testarea în ani și sezoane diferite, precum și în ferme diferite, în condiții de mediu diferite care afectează manifestarea fenotipică a caracterelor [Gr96].

Metodologia BLUP a fost introdusă de către Hendeson C.R., diferența esențială față de indicele de selecție constă în faptul că se abandonează ipoteza cunoașterii mediei observațiilor [He75]. Ca și în cazul indicilor de selecție, se reține elementul \hat{a} , care minimizează media abaterilor pătratică dintre predictor (valoarea de ameliorare predictată) și predictand (valoarea de ameliorare adevărată), respectiv $E(\hat{a} - a)^2$ minim. Această expresie reprezintă totodată și criteriul de calitate al metodei celor mai mici pătrate.

În BLUP, prin estimarea efectelor fixe necunoscute (notate cu \tilde{b}), în care intră μ și alți factori ficși (fermă, sezon, etc), este introdusă o restricție suplimentară: valoarea de ameliorare predictată să fie nedeplasată, adică $E(\hat{a})=E(a)$. Această restricție este obiectivă, ea impunând ca valoarea de ameliorare predictată să nu fie în medie afectată de efectele fixe[He77].

O populație statistică poate fi descrisă prin doi indicatori de bază: media (μ), numită și momentul de ordinul I și varianța (σ^2), cunoscută sub numele de moment de ordinul II. În general primul moment a lui y (vectorul observațiilor), notat Xb nu este cunoscut, însă al doilea moment $V=\sigma^2$ se presupune cunoscut. În acest context, valoarea de ameliorare (cel mai bun predictor liniar nedeplasat) se poate obține pornind de la modelul liniar (forma matriceală):

$$y = Xb + Zu + e$$

în care se consideră că: Xb reprezintă variabilele independente fixe, iar Zu variabilele independente aleatoare (animalele)[Ro91].

Estimarea lui b (\tilde{b}) prin metoda celor mai mici pătrate generalizate se obține din ecuația:

$$[X'V^{-1}X] \times [\tilde{b}] = [X'V^{-1}y]$$

Iar predicția lui u (\hat{u}) din relația clasică în ameliorarea animalelor:

$$\hat{u} = C' \cdot V^{-1} \times (y - X\tilde{b})$$

Pentru început valorile genetice sunt analizate provizoriu ca fixate și nu ca aleatoare.

În această situație, $\text{Var}(y) = V = R$.

Prin rescrierea ecuațiilor celor mai mici pătrate generalizate pentru estimarea lui b (\tilde{b}) și predicția lui u (\hat{u}) prin extensie, se obține:

$$\begin{bmatrix} X' \\ Z' \end{bmatrix} R^{-1} [XZ] \times \begin{bmatrix} \tilde{b} \\ \hat{u} \end{bmatrix} = \begin{bmatrix} X' \\ Z' \end{bmatrix} R^{-1} y$$

de unde rezultă:

$$\begin{bmatrix} X' R^{-1} X & Z' R^{-1} Z \\ Z' R^{-1} X & Z' R^{-1} Z \end{bmatrix} \times \begin{bmatrix} \tilde{b} \\ \hat{u} \end{bmatrix} = \begin{bmatrix} X' R^{-1} y \\ Z' R^{-1} y \end{bmatrix}$$

Dacă la blocul $Z' R^{-1} Z$, unde Z este matricea de incidență a efectelor aleatoare, se adaugă inversa matricei $\text{var}(A)=G$, A – **matricea relațiilor genetice aditive** a însușirii analizate, sistemul furnizează simultan soluții pentru b și u :

$$\begin{bmatrix} X' R^{-1} X & Z' R^{-1} Z \\ Z' R^{-1} X & Z' R^{-1} Z + G^{-1} \end{bmatrix} \times \begin{bmatrix} \tilde{b} \\ \hat{u} \end{bmatrix} = \begin{bmatrix} X' R^{-1} y \\ Z' R^{-1} y \end{bmatrix}$$

Acest sistem de ecuații se numește **sistemul ecuațiilor modelului mixt**, deoarece se consideră simultan efectele fixate și efectele aleatoare [Ma99, McC01].

Frecvent se emite ipoteza că rezidualele modelului sunt distribuite conform unei legi normale de repartiție cu media $\mu = 0$ și varianța σ_e^2

Pentru a simplifica sistemul, se pot înmulți toate ecuațiile cu σ_e^2 , ceea ce duce la eliminarea matricei R^{-1} , deoarece $R^{-1} \cdot R = I$

Pentru simplificarea sistemului, prin înmulțirea cu R se obține:

$$\begin{bmatrix} X' X & X' Z \\ Z' X & Z' Z + G^{-1} \end{bmatrix} \times \begin{bmatrix} \tilde{b} \\ \hat{u} \end{bmatrix} = \begin{bmatrix} X' y \\ Z' y \end{bmatrix}$$

$$\text{Unde } G^{-1} = I \times \frac{\sigma_e^2}{\sigma_u^2},$$

Iar dacă se ține seama de înruderile dintre animale, atunci:

$$G^{-1} = A^{-1} \times \frac{\sigma_e^2}{\sigma_u^2}, \quad \sigma_e^2 \text{ și } \sigma_u^2 \text{ reprezintă varianțele reziduale și cele datorate}$$

reproducătorului.

Utilizarea BLUP permite astfel o evaluare simultană a efectelor de mediu și o predicție a efectelor genetice [He77].

Există mai multe modele biometrice care conduc la rezultate BLUP, dintre acestea cele mai cunoscute sunt: modelul tată, modelul animal individual, modelul animal cu repetabilitate, modelul tată-mamă, modelul animal individual pe mai multe caractere [He84].

În rezolvarea modelelor prin metodologia BLUP, o importanță majoră o reprezintă matricea relațiilor genetice dintre indivizi (A) [Bi00].

Matricea relațiilor genetice dintre indivizi

Animalele dintr-o populație sunt înrudite între ele, cauzele înrudirii lor fiind: mărimea limitată a populației și originea populației dintr-un număr mic de strămoși [Hen75].

Asemănarea genetică între rude poate fi exprimată în trei moduri:

- prin probabilitatea ca o genă luată la întâmplare de la un individ să fie identică prin origine cu o genă luată la întâmplare de la celălalt individ, indice numit coeficient de consangvinizare, în sensul de înrudire.
- Prin corelația dintre valorile de ameliorare ale indivizilor (r), care este estimată de coeficientul de înrudire (*relationship coefficient*). Această metodă poate fi programată pe computer.
- Prin analiza covarianței dintre rude.

Coeficientul de înrudire este un parametru statistic care arată asemănarea cea mai probabilă dintre indivizi. Formula coeficientului de înrudire este:

$$r_{xy} = \frac{\sum (0.5)^{n_1+n_2} + (1 + F_A)}{\sqrt{(1 + F_x) \cdot (1 + F_y)}}$$

Unde r_{xy} reprezintă corelația între valorile de ameliorare ale indivizilor x și z.

F_A , consangvinizarea strămoșului comun

F_x = consangvinizarea individului X

F_y = consangvinizarea individului Z, pe un alt strămoș decât strămoșul comun.

Asemănarea genetică crește dacă un strămoș comun se repetă de mai multe ori, crește dacă strămoșul comun a fost consangvinizat (F_A) și este diminuată dacă indivizii respectivi au fost ei înșiși consangvinizați (F_x și F_y), pe alți strămoși decât strămoșul comun.

r_{xy} reprezintă corelația dintre bazele ereditare ale indivizilor x și y. Relația poate fi scrisă și sub forma:

$$r_{xy} = \frac{a_{xy}}{\sqrt{a_{xx} \times a_{yy}}} \text{ și, deoarece } r_{xy} = \frac{Cov_{xy}}{\sqrt{Var_x \times Var_y}}, \text{ rezultă:}$$

$$a_{xy} = r_{xy} \cdot \sqrt{a_{xx} \times a_{yy}}$$

unde: $a_{xx} = 1 + F_x$ și $a_{yy} = 1 + F_y$

dacă $F_x = F_y = 0$, atunci $a_{xy} = r_{xy}$, iar a_{xy} reprezintă relația genetică dintre indivizii x și y.

Matricea relațiilor genetice A este o matrice simetrică, elementele ei se determină după metoda evaluării concentrice (metoda tubulară)[Un93]. Această metodă este recursivă și poate fi ușor programabilă pe calculator [Br02].

Pentru calcularea elementelor matricei A, animalele din pedigreu sunt trecute într-o listă, notate de la 1 la n și grupate în ordinea generațiilor în așa fel încât părinții să precedă descendenții[Cr05].

Elementele matricei A se calculează după formula:

$$a_{xx} = 1 + 0.5a_{T,M_x}$$

$$a_{xy} = 0.5(a_{xT_x} + a_{xM_x})$$

Unde: a_{T,M_x} = relația genetică aditivă dintre părinții individului x.

a_{xT_x} = relația genetică aditivă dintre individul x și tatăl lui y

a_{xM_x} = relația genetică aditivă dintre individul x și mama lui y

Dacă un singur părinte este cunoscut (de exemplu tatăl)

$$a_{xx} = 1 \text{ și } a_{xy} = 0.5(a_{xT_x})$$

Dacă ambii părinți sunt necunoscuți,

$$a_{xx} = 1 \text{ și } a_{xy} = 0$$

Matricea relațiilor genetice dintre indivizi este elementul cheie în cadrul unui model animal, permițând folosirea tuturor surselor de informație pe o cale optimă [Un92].

Modelul tată

Modelul tată (sire model) pentru descendenți întreținuți în condiții diferite de mediu .

Acest model presupune calculul valorii de ameliorare după modelul:

$$y_{ijk} = \mu + f_i + T_j + e_{ijk}$$

y_{ijk} - performanța măsurată la descendentul k , provenit din reproducătorul j , realizat în cadrul nivelului i al efectului de mediu

μ - media populației comune pentru toate observațiile;

f_i - efectul fix al nivelului i al factorului de mediu (ex. Combinația fermă – an-sezon), $i=1, \dots, p$;

T_j – efectul randomizat al reproducătorului j ($T_j = 0,5 a_j$, unde a_j reprezintă valoarea de ameliorare a reproducătorului j);

e_{ijk} – efectul rezidual randomizat, asociat fiecărei observații.

Modelul tată este un model liniar mixt, el include pe lângă constanta μ (care simbolizează nivelul mediu de producție al populației), efectul fix a fermei f_i , și efectul aleator al reproducătorului T_j . Modelul conține de asemenea efectul aleator rezidual e_{ijk} . [Da00, Be99]. Cele două variabile aleatoare ale modelului, T și e , au mediile nule și varianțele σ_T^2 și σ_e^2 .

Forma matriceală a modelului linear poate fi scrisă sub forma:

$$Y = Xb + Zu + e$$

Pentru „ n ” performanțe, „ m ” efecte fixe și „ p ” efecte aleatoare

- y – vectorul observațiilor, $n \times 1$;

- b – vectorul efectelor fixe, $m \times 1$;
- u – vectorul efectelor randomizate, $p \times 1$
- e – vectorul rezidualelor, $n \times 1$
- X – matrice de design care asociază performanțele la efectele fixe, $n \times m$;
- Z – matrice de design care asociază performanțele la efectele randomizate, $n \times p$;

În cazul modelului, valorile așteptate și matricele varianțe - covarianțe ale efectelor randomizate sunt:

$$\begin{aligned} M(y) &= \mu + f_i = Xb & \text{Var}(T) &= \sigma_T^2 = G \\ M(T) &= 0 & \text{Var}(e) &= \sigma_e^2 = R \\ M(e) &= 0 & \text{Var}(y) &= \sigma_f^2 + \sigma_T^2 = \sigma_y^2 = ZGZ' + R = V \\ \text{Cov}(e, t) &= 0 \end{aligned}$$

Reziduala înglobează ansamblul factorilor neluați în considerare, fie că sunt de natură genetică (valoare genetică neaditivă) sau de mediu, jumătate din valoarea genetică aditivă a mamei descendentului și contribuția riscului de meioză.

Ecuțiile modelului mixt sunt:

$$\begin{bmatrix} X'X & X'Z \\ Z'X & Z'Z + G^{-1} \end{bmatrix} \times \begin{bmatrix} \tilde{b} \\ \tilde{u} \end{bmatrix} = \begin{bmatrix} X'y \\ Z'y \end{bmatrix}$$

unde:

$$G = A \cdot \frac{\sigma_T^2}{\sigma_e^2} \quad \text{și} \quad G^{-1} = A^{-1} \cdot \frac{\sigma_e^2}{\sigma_T^2} = A^{-1} \cdot k$$

- dacă reproducătorii nu sunt înrudiți: $A=I$, unde I este o matrice de identitate, de dimensiuni egale cu numărul de reproducători.

La diagonala ecuațiilor celor mai mici pătrate se adaugă (rezultă din procesul derivării ecuațiilor modelului mixt) un coeficient (k) fiecărui taur. Acesta reprezintă un raport între varianța reziduală și varianța datorată reproducătorilor.

Pentru a ajunge la k se folosesc următoarele relații:

$$\begin{aligned} \sigma_T^2 &= \frac{1}{4} \cdot \sigma_a^2 = \frac{1}{4} h^2 \cdot \sigma_y^2 \\ \sigma_y^2 &= \sigma_T^2 + \sigma_e^2 = 0,25 \cdot h^2 \cdot \sigma_y^2 + \sigma_e^2 \\ \sigma_e^2 &= \sigma_y^2 - \sigma_T^2 = \sigma_y^2 - 0,25 \cdot h^2 \cdot \sigma_y^2 = \sigma_y^2 \cdot (1 - 0,25h^2) \\ \text{deci } k &= \frac{\sigma_e^2}{\sigma_T^2} = \frac{\sigma_y^2 \cdot (1 - 0,25 \cdot h^2)}{\sigma_y^2 \cdot 0,25 \cdot h^2} = \frac{1 - 0,25 \cdot h^2}{0,25 \cdot h^2} \quad \text{sau } k = \frac{4 - h^2}{h^2} \end{aligned}$$

Astfel, ecuațiile modelului mixt conduc prin rezolvarea lui la obținerea soluțiilor care reprezintă mediile caracterului pentru elementul fix și capacitatea de transmitere a caracterului pentru elementul aleator [He75, McP04].

Modelul animal individual

Modelul animal individual este considerat a fi modelul care utilizează sursele de informații (ascendenți, performanță proprie, colaterali, descendenți) pe o cale optimă, indiferent de numărul de caractere considerate. Acesta se apropie de modelul ideal deoarece poate descrie majoritatea elementelor din cadrul unui eveniment biologic [Gr97].

Majoritatea ipotezelor modelului tată nu se regăsesc în practică dacă:

- a. Femelele partenere nu sunt alese la întâmplare, respectiv se practică potrivirea împerecherilor;
- b. Există legături genetice între: reproducătorii masculi; reproducătorii masculi și femelele partenere; femelele partenere;
- c. Candidații la selecție fac parte din mai multe populații.

În aceste condiții se preferă folosirea modelului

$$\text{În modelul } y_{ij} = \mu + F_i + a_j + e_{ij}$$

μ - media generală;

F_i - efectul fix al mediului i ;

a_j - reprezintă valoarea genetică aditivă a animalului j , variabilă aleatoare la care se presupune o repartiție normală $N(\mu, \sigma_a^2)$;

e_{ij} - efectul de mediu randomizat, variabilă aleatoare cu distribuție normală $N(0, \sigma_e^2)$

În model sunt introduse și animale fără performanțe proprii, dar care interesează deoarece sunt înrudite cu animalele care au performanțe proprii.

Matricea relațiilor genetice dintre indivizi se calculează prin metoda evaluării concentrice. Aceasta calculează gradul de consangvinizare plecând de la cel mai îndepărtat strămoș și ajungând la cel mai tânăr, concentrând simultan informații doar dintre două generații [Un93, Br02, Sa01].

Modelul testului zilei de control (Test Day Model)

În modelele prezentate, considerate modele clasice, pentru evaluarea genetică a caracterelor de lapte la taurine, se utilizează cantitatea de lapte pe lactația normală (305 zile), cantitate ce se obține în urma controlului cantitativ al producției prin cântărirea laptelui [Mu04, Pt93].

Modele ce utilizează în locul cantității de lapte pe lactația normală, testul zilei de control (test day milk yields) reprezintă o alternativă în estimarea parametrilor genetici și a valorii de ameliorare (*breeding value*) [Wi97, Ub03]. Aceste modele sunt cunoscute sub numele de Test Day Model (TDM) și sunt cele care se utilizează astăzi în modelarea curbei de lactație și estimarea parametrilor genetici [Mach99, Ja97].

Diferența constă în faptul că în calcul nu se ia cantitatea de lapte (sau alt caracter productiv) realizată într-o lactație normală (305 zile), ci cantitățile de lapte realizate zilnic [Ts00].

Autorii modelului [Sw95, Swa95], susțin că modelul testului zilei de control, comparativ cu cel clasic, asigură o modelare mai apropiată de realitate, fiind surprinse toate efectele genetice și de mediu care influențează valoarea fenotipică. Vacile pot fi incluse în evaluare și dacă au cel puțin producția de pe o singură zi, iar taurii poate fi evaluați cu mai bună precizie având la dispoziție datele fiicelor aflate în evidență.

În realizarea modelului s-a plecat de la modelul clasic, model animal cu repetabilitate, în care testele zilei de control sunt considerate ca măsurători repetate în cadrul aceleași lactații [Ca02, Za95].

Modelul conține un submodel care include factorii care modelează curba de lactație.

Modelul folosit este următorul:

$$y_{ijkmn} = HTD_{im} + A_{jm} + P_{jm} + AS_{km} + \text{submodelul ce modelează curba de lactație} + e_{ijkmn}$$

n teste (305), j vaci, m lactații (se iau maxim 3 lactații); i ferme (cirezii - herds), k subclase vârstă la fătare (de exemplu: 18 - 24 luni, 25 - 29 luni, 30 - 34 luni, 35 - 48 luni, 5 clase).

Y_{ijkmn} - reprezintă al n -lea test al vacii j , la lactația m ;

HTD_{im} - efectul fix al combinației fermă - data controlului;

A_{jm} - efectul genetic aditiv al animalului (randomizat);

P_{jm} - efectul de mediu permanent în interiorul lactației, comun tuturor testelor zilnice din lactația m (randomizat)

AS_{km} - efectul fix al subclasei vârstă la fătare - sezonul fătării, la lactația m .

e_{ijkmn} - reziduala modelului

Pentru submodelul ce modelează curba de lactație se folosește unul din modelele [Po00]:

1. regresie liniară multiplă propusă de Ali și Schaeffer [Ali87, Sch96]:

$$y = b_{km1} \cdot \frac{x}{c} + b_{km2} \cdot \frac{x}{c} + b_{km3} \cdot \ln \frac{c}{x} + b_{km4} \cdot \left(\ln \frac{c}{x}\right)^2$$

Unde x reprezintă ziua de control (DIM - day in milk), iar C o constantă, de obicei valoarea numărului de zile a lactației totale.

2. Curba lui Wilmink [Druet05]:

$$y = a \cdot x^b \cdot e^{-cx}$$

Cu a, b și c parametri asociați cu producția corespunzătoare pantei ascendente, vârfului și pantei descendente curbei de lactație

3. Polinoamele lui Legendre de ordinul cinci.

$$y(x) = \sum a_n \cdot P_n(x)$$

Modelul testului zilei de control a fost extins la lactații multiple și diferite caractere (lapte, grăsime, proteină și celule somatice) [Oth02]. În notație matriceală, modelul Test Day pentru lactație multiplă este:

$$y = Hc + Xb + Wp + Za + e$$

Unde: y – vectorul observațiilor în 3 lactații;
 c – vectorul efectelor fixe (ferma, zi de test, lactație) datorat grupului de contemporane
 b - vectorul efectelor fixe datorat condițiilor de mediu
 p – vectorul efectelor randomizate datorat mediului
 a – vectorul efectelor randomizate datorat efectelor genetice aditive
 H – matricea de incidență asociată grupului de contemporane
 X, W, Z - sunt matricile covarianțelor asociate cu DIM

Matricea covarianțelor așteptate sunt:

$$E \begin{bmatrix} y \\ p \\ a \\ e \end{bmatrix} = \begin{bmatrix} Hb + Xb \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ și } \text{Var} \begin{bmatrix} p \\ a \\ e \end{bmatrix} = \begin{bmatrix} P & 0 & 0 \\ 0 & G & 0 \\ 0 & 0 & R \end{bmatrix}$$

Unde $P = I \otimes P_0$, $G = A \otimes G_0$, $R = \sum R_j$

- A matricea relațiilor genetice aditive;
- P_0 și G_0 matricea covarianțelor condițiilor de mediu și coeficienților genetici de regresie, iar R_{ij} este matricea covarianțelor pentru vaca i în ziua de test j ;

Rezolvarea modelului se face ca și pentru modelele clasice, prin scrierea ecuațiilor modelului mixt, ce are ca soluții estimarea efectelor fixe și predicția celor aleatoare.

Ecuțiile modelului mixt sunt:

$$\begin{bmatrix} H'R^{-1}H & H'R^{-1}X & H'R^{-1}W & H'R^{-1}Z \\ X'R^{-1}H & X'R^{-1}X & X'R^{-1}W & X'R^{-1}Z \\ W'R^{-1}H & W'R^{-1}X & W'R^{-1}W + P^{-1} & W'R^{-1}Z \\ Z'R^{-1}H & Z'R^{-1}X & Z'R^{-1}W & Z'R^{-1}Z + G^{-1} \end{bmatrix} \begin{bmatrix} \hat{c} \\ \hat{b} \\ \hat{p} \\ \hat{a} \end{bmatrix} = \begin{bmatrix} H'R^{-1}y \\ X'R^{-1}y \\ W'R^{-1}y \\ Z'R^{-1}y \end{bmatrix}$$

Soluțiile obținute predictează valoarea de ameliorare și estimează curba de lactație.

1.3.4. Elemente care influențează aplicarea metodologiei BLUP

Rezolvarea modelele prezentate se face prin aplicarea metodologie BLUP, eficacitatea acestuia depinde de mai multe elemente[Gr97].

1. Numărul de animale cuprinse în analiză.

Evaluarea genetică prin BLUP – modelul animal necesită rezolvarea unui sistem de ecuații de mărimea $N_c \times (N_f + N_r)$, N_c fiind numărul de caractere, N_f numărul de niveluri ale efectelor fixe și N_r numărul de niveluri ale efectelor randomizate, adică numărul total de animale de evaluat.

2. Existența unei baze de date.

Pentru o evaluare genetică la nivel național, informațiile trebuie să se regăsească într-o bază de date (sunt necesare informații legate de performanțe precum și cele referitoare la genealogii). Înregistrarea informațiilor trebuie să se realizeze în mod centralizat, eventuala cu legătură cu baze de date similare existente în lume.

3. Nivelul de asigurare a înșămânțărilor artificiale (IA) în populațiile supuse evaluării.

Metodologia BLUP – modelul animal nu permite comparația riguroasă a animalelor din două grupe diferite de contemporani, decât atunci când există legături genetice între ele. Acest lucru se poate realiza prin folosirea IA, care permit obținerea de semifrați situați în ferme diferite, ceea ce duce la existența unor legături genetice între ferme.

4. Fiabilitatea informațiilor.

Creșterea preciziei selecției în urma utilizării metodologiei BLUP, comparativ cu indicii de selecție, se datorează în principal utilizării informațiilor genealogice. Dacă aceste informații conțin erori, câștigul așteptat poate fi diminuat sau chiar anulat.

5. Parametri genetici utilizați.

Parametri genetici (h^2, r_g) ai populației sunt utilizați la predicția valorii de ameliorare. Fiabilitatea acestor valori depinde de precizia cu care au fost estimați parametri genetici. Cea mai sigură metodă pentru estimarea parametrilor genetici este REML (*Restricted Maximum Likelihood*).

2. FACTORI CE INFLUENȚEAZĂ VALOAREA FENOTIPICĂ LA VACILE DE LAPTE

2.1. Modelul regresional multifactorial

Valoarea fenotipică a unui caracter cantitativ este determinată de mai mulți factori cu efect cunoscut sau necunoscut, controlați sau necontrolați.

Una dintre preocupările majore ale statisticii este evidențierea și măsurarea influenței acestor factori asupra variabilelor, demers care comportă o serie de dificultăți legate în primul rând de complexitatea relațiilor, precum și de faptul că relațiile de cauzalitate nu sunt deterministe și se manifestă în general sub formă de tendință.

În procesul cunoașterii, se practică studierea sistemelor prin modelare. Modelele matematice utilizate pentru studierea sistemelor biologice (modele biometrice), permit studierea comportamentului și influența factorilor ce concurează la fenomenul studiat [Cr99].

În evaluarea genetică, pentru modelarea informațiilor fenotipice, se utilizează un **model liniar**, definit ca o reprezentare matematică a efectelor genetice și negenetice care influențează caracterul analizat [Re00]. Alegerea modelului liniar s-a făcut pornind de la modelul fundamental al geneticii cantitative prezentat în capitolul 1.

Pentru a oferi explicații, sau pentru a realiza predicții este nevoie, cel mai adesea, să fie analizate relații între două sau mai multe variabile: între cea a cărei variație încercăm să o explicăm, numită și variabila dependentă, și una sau mai multe variabile independente. În acest caz, relația care există între variabile este o relație regresională.

Modelul de regresie simplă este folosit pentru a descrie relația dintre două variabile. În cazul în care sunt disponibile date despre mai mulți factori cu potențial explicativ, iar aceștia sunt estimați prin variabile cantitative, este de dorit ca analiza să cuprindă simultan toate variabilele și nu doar două dintre acestea. Utilizarea regresiei simple într-un astfel de caz, prin ignorarea unora dintre variabilele independente, ori prin aplicarea succesivă pentru fiecare dintre variabilele independente, poate să conducă la rezultate eronate [Da05].

Metoda regresiei

Cercetarea legăturilor dintre variabile se face cu ajutorul funcțiilor de regresie.

Dacă y este o variabilă dependentă și $x_1, x_2, \dots, x_i, \dots, x_n$, variabile independente, metoda regresiei consideră că y este o funcție de variabile independente $y=f(x_1, x_2, \dots, x_i, x_n)$.

În funcție de forma și gradul funcției, precum și de numărul variabilelor, ecuația de regresie poate să definească o curbă, o suprafață, sau chiar un poliedru.

Modelul de dependență statistică în cazul metodei regresiei înlocuiește modelul teoretic cunoscut: $y= f(x_1, x_2, \dots, x_i, x_n) + \xi$ în care ξ - reprezintă o eroare aleatoare sau mai exact o variabilă reziduală cu dispersia constantă și cu media nulă.

În funcție de numărul factorilor $x_1, x_2, \dots, x_i, \dots, x_n$ care influențează caracteristica rezultativă, există:

- regresie unifactorială sau simplă;
- regresie multifactorială sau multiplă.

Regresie multifactorială

În cazul regresiei multifactoriale legăturile dintre variabile sunt foarte complexe, variabila dependentă este influențată de un număr mare de variabile independente, iar ecuația regresiei multiple are forma :

$$y = f(x_1, x_2, \dots, x_i, x_n) + \xi$$

unde, $x_1, x_2, \dots, x_i, \dots, x_n$ – sunt caracteristicile factoriale, iar ξ - variabila reziduală.

În cazul regresii multiple se utilizează deseori modelul liniar, a cărui expresie are relația:

$$y = a_0 + a_1 x_1 + a_2 x_2 + \dots + a_p x_p + U$$

În acest model a_0 este coeficientul care exprimă influența factorilor neincluși în model, iar $a_i, i=1, n$ se numesc coeficienți de regresie multiplă și arată influența caracteristicii factoriale asupra caracteristicii rezultative, U fiind variabila reziduală.

Legătura multifactorială liniară se poate reprezenta grafic sub forma unui plan. Fiecare din aceste ecuații dă o dreaptă. Intersecția acestor drepte va furniza o suprafață, soluția fiind pe marginea acestuia.

Problema generală care se rezolvă prin modelare statistică se reduce la determinarea următoarelor:

1. y este un fenomen care trebuie explicat ;
2. x_1, x_2, \dots sunt factori explicativi potențiali;
3. Determinarea efectului independent al fiecărui x_i asupra lui y ;
4. Determinarea ierarhia importanței factorilor x_1, x_2, \dots în explicarea lui y .

Analiza și interpretarea modelelor de regresie multiplă

În rezolvarea modelului, în urma determinării coeficienților a_i , se obține pentru variabila dependentă o valoare calculată, notată cu y' .

Coeficientul a_i , numit panta asociată variabilei x_i , reprezintă numărul de unități cu care variază y' atunci când x_i crește cu o unitate iar celelalte variabile independente sunt menținute constante. Altfel spus, a_i arată cum se modifică valoarea așteptată a variabilei dependente atunci când x_i variază iar x_j , sunt constante, $j \neq i$.

De asemenea, în măsura în care datele satisfac anumite proprietăți (dintre care un principiu important este cel al distribuțiilor apropiate de cele normale), a_i indică variația în mediile valorilor lui y care corespund punctelor de forma $(x_1, x_2, \dots, x_i, \dots, x_n)$. iar a_0 , arată care este media lui y atunci când $x_1=0, x_2=0, \dots, x_i=0, \dots, x_n=0$.

Semnul plus al lui a , indică faptul că între x , și y are loc o relație pozitivă, în condiții de control al efectului celorlalte variabile, în timp ce semnul minus indică prezența unei relații negative.

Din interpretarea coeficienților a_i , se vede cum regresia multiplă permite compararea de perechi de grupe de indivizi statistici care sunt identici din perspectiva tuturor variabilelor independente cu excepția unei singure variabile. Diferența observată în valorile variabilei dependente este atribuită variației în variabila independentă care ia valori diferite pentru grupe diferite.

Această proprietate arată faptul că **panta de regresie nu poate fi folosită drept un indicator al intensității relației** dintre variabila dependentă și variabila independentă corespunzătoare, și nici nu permite ierarhizarea variabilelor independente în funcție de contribuția fiecăreia la explicația variației variabilei dependente[Da05].

Eficiența unui model de regresie multiplă

Puterea explicativă a unui model multifactorial poate fi evaluată cu ajutorul indicatorului R^2 , numit **coeficient de determinație multiplă**. El este definit la fel ca și atunci când avem o singură variabilă independentă și are o interpretare similară:

$$R^2 = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2},$$

Raportul dintre măsura varianțelor explicate și măsura varianțelor totale, deci, R^2 indică proporția din variația lui Y care este "explicată" de toate variabilele independente din model. Din definiție rezultă că R^2 poate să ia valori între 0 și 1.

Valorile ridicate ale lui R^2 sunt de dorit în locul celor scăzute pentru că implică faptul că explicația este în mai mare măsură completă. Totuși, această afirmație necesită următoarea precizare: R^2 are proprietatea de a crește cu fiecare variabilă care este introdusă în model și de aceea valorile lui R^2 trebuie judecate și în raport cu numărul de variabile independente. La limită, este posibil să avem $R^2 = 1$ dacă avem un număr de variabile independente suficient de mare, chiar dacă acestea sunt generate aleator.

Concluzia care se desprinde este că, alegerea variabilelor care urmează să fie incluse în model, nu poate fi decisă folosind exclusiv informația de natură statistică (chiar dacă există procedee complexe prin care putem îmbogăți această informație). Numai prin luarea în considerare și a unor aspecte de natură teoretică poate fi decisă includerea sau eliminarea unor variabile în analiză[Da05].

Pentru interpretarea valorilor intermediare ale lui R^2 , este necesar să înțelegem interpretarea valorilor maximale.

$R^2 = 1$ atunci când valorile lui Y sunt complet determinate de combinațiile liniare ale valorilor variabilelor independente.

$R^2 = 0$ indică absența unor relații între variabilele independente și variabila dependentă.

Generalizarea rezultatelor obținute pe eșantion (inferența)

Problema generalizării rezultatelor obținute prin modelare, efectuate asupra unui eșantion, asupra întregii populații se rezolvă prin 2 metode:

- Construirea unui interval de încredere în jurul coeficienților a_i , interval ce poate fi exprimat cu o anumită probabilitate.
- Determinarea valorii maxime a nivelului de semnificație statistică pentru care 0 aparține intervalului de încredere (testul t), cu alte cuvinte, care este probabilitatea ca a_i sa fie nul.

2.2. Factori care influențează producția de lapte

Principalul caracter productiv la vacă îl reprezintă producția de lapte. Acest caracter, de natură cantitativă nu reprezintă unicul caracter productiv, calitatea producției de lapte este reprezentată de un cumul de caractere, dintre care se poate menționa cantitatea de grăsime din lapte, conținutul de proteină din lapte, calitatea cazeinei din lapte etc.

Producția de lapte este influențată de factori de natură genetică, factori legați de furajarea, și factori de mediu. Această triadă este cunoscută ca: „rasă, masă, casă”

În modelul propus, se analizează influența următorilor factori asupra caracterelor productive la vaca, reprezentat de caracterul producția lapte. Aceștia sunt:

- Valoarea de ameliorare a taurului tată;
- Factori de mediu (sezonul fătării, regiunea unde este localizată ferma, etc.);
- Caracteristicile curbei de lactație reprezentată de trei parametri (a, b și c) ce reprezintă: panta ascendentă, vârful curbei de lactație și panta descendentă a acesteia.

Introducerea în model a parametrilor (a, b, c) care caracterizează curba de lactație apropie modelul propus de modelul testului zilei de control [Dr03].

Modelarea curbei de lactație cu ajutorul funcției $y_t = a \cdot t^b \cdot e^{-ct}$, folosește trei parametri (a, b, c) care caracterizează curba de lactație, t fiind ziua de control (de la 0 la 305), y_t producția de lapte din ziua respectivă.

Informațiile culese în vederea analizei sunt:

- Valoarea de ameliorare a taurului tată, caracter de natură cantitativă;
- Curba de lactație, reprezentată prin trei parametri asociați corespunzătoare cu:
 - a - panta ascendentă,
 - b - vârful pantei ascendente,
 - c - panta descendentă
- Condiții de mediu – caractere de natură calitativă:
 - Sezonul fătării categorisită pe două nivele
 - sezon 1 (aprilie – septembrie)
 - sezon 2 (octombrie – martie)
 - Regiunea categorisită pe trei nivele
 - șes
 - deal
 - munte

Factorii enumerați mai sus și luați în analiză, sunt factori care se pot culege din informațiile existente în baza de date, și se deduc în principal din:

- data nașterii
- data primei fătări
- localizarea exploatației
- vârsta la prima fătare
- cod matricol tată

– controalele periodice ale producției de lapte

Curba de lactație (figura 1), poate fi modelată prin diferite tipuri de submodele matematice neliniare, funcție de ziua de control (6-305 zile), ecuația modelului neliniar fiind:

$$y_t = a \cdot t^b \cdot e^{-ct}$$

a,b,c fiind parametri asociați cu producția corespunzătoare pantei ascendente, vârfului și pantei descendente a curbei de lactație, iar t ziua de control, y_t , producția de lapte în ziua de control (DIM day in milk).

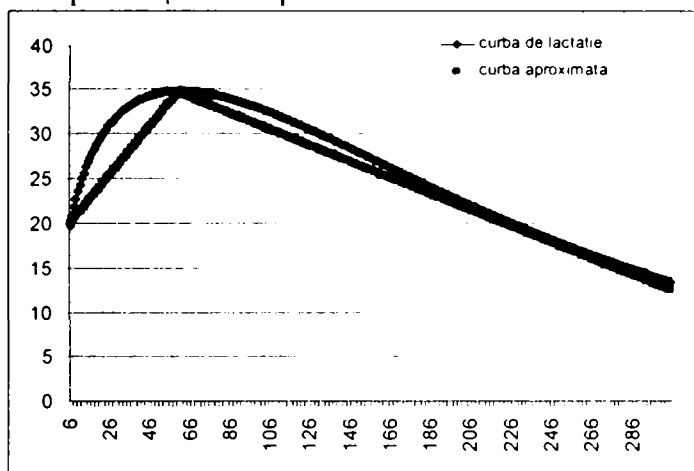


Figura 1. Curba de lactație

2.2.1. Stabilirea modelului matematic

Pentru a putea prezenta ponderea factorilor enunțați mai sus în realizarea producției de lapte pe lactație normală (performanță realizată în primele 305 zile de lactație începând din a doua zi după parturițe) s-a recurs la construirea unui model regresional multiliniar de forma :

$$y = k_0 + k_1x_1 + k_2x_2 + k_3x_3 + k_4x_4 + k_5x_5$$

Alegerea acestui tip de model s-a făcut din următoarele considerente:

1. Alegerea modelului liniar s-a făcut pornind de la modelul fundamental al geneticii cantitative prezentat în capitolul 1, care de asemenea este un model liniar, definit ca o reprezentare matematică a efectelor genetice și negenetice care influențează valoarea fenotipică a producției de lapte.
2. Pentru realizarea predicției este necesar să fie analizate relațiile între două sau mai multe variabile, considerate variabile independente și cea rezultativă, variabila dependentă, care în cazul problemei propuse este producția de lapte.
3. Factorii care influențează caracterul analizat sunt reprezentați de variabile cantitative și calitative dihotomice și trihotomice, acestea permit stabilirea unor legături regresionale.

În modelul regresional multiliniar, variabila dependentă y reprezintă producția de lapte pe lactație normală a fiecărei fiice aflată în testare iar variabilele independente x_i reprezintă factorii de mediu și factorul de natură genetică reprezentat de valoarea de ameliorare a taurului tată ce caracterizează fiecare fiică aflată în testare.

Variabilă independentă	Semnificație
x_1	Sezonul fătării Variabilă calitativă de tip ordinal, dihotomică categorizată astfel: – 1 - sezon vară – 2 - sezon iarnă
x_2	Regiune de exploatație Variabilă calitativă de tip ordinal, trihotomică categorizată astfel: – 1 - câmpie (șes) – 2 - deal – 3 - munte
x_3	Parametru asociat cu producția corespunzătoare pantei ascendentă a curbei de lactație. Variabilă cantitativă
x_4	Parametru asociat cu producția corespunzătoare vârful a curbei de lactație. Variabilă cantitativă
x_5	Parametru asociat cu producția corespunzătoare pantei descendentă a curbei de lactație. Variabilă cantitativă
x_6	Valoarea de ameliorare a taurului tată calculat pe baza metodologie BLUP cu ajutorul aplicației Ameliorare. Variabilă cantitativă

Variabila dependentă o constituie cantitatea de lapte pe lactație normală realizată de fiică. În cazul în care fiica aflată în testare nu a încheiat lactația în momentul aprecierii, valoarea producției de lapte pe lactație normală a fost calculată pe baza cantităților realizat în primele 100 de zile de lactație (între producția de lapte în prima sută de zile și producția de lapte estimată pe lactație normală există o corelație foarte semnificativă $r=0,913+0,003$).

Cantitatea de lapte pe lactația normală este o variabilă cantitativă, se obține în urma controlului cantitativ al producției prin cântărirea laptelui (prin gravimetre), iar exprimarea producției se face în kg de lapte. Controlul cantitativ a fost individual – periodic, realizat prin 13 controale; perioada de control a fost de 28 de zile, prima și ultima perioadă de control pot fi mai lungi (maxim 46 zile) sau mai scurte (minim 14 zile), în funcție de ziua fătării, respectiv a înțărării vacii.

Analiza ecuației regresionale

Date de intrare. Materialul biologic este reprezentat de:

- 10 tauri din rasa Bălțată Românească fără legături genetice.
- 212 de fiice în testare, din aceeași rasă, localizate în exploatații situate în cele trei categorii de regiune; prima lor lactație s-a derulat atât vara cât și iarna.

- Au fost eliminate din eșantion fiicele în testare a căror producții de lapte se aflau în afara intervalului (2500,7500) kg lapte.
- Cele 212 de fiice au fost montate prin însemințare după cum urmează:

Taur	Număr de fiice în testare
T1	11
T2	19
T3	35
T4	16
T5	21
T6	23
T7	17
T8	25
T9	28
T10	17

Modelul regresional multifactorial „Producția de lapte”

În urma prelucrării datelor din eșantionul luat în studiu s-au obținut următoarele valori pentru coeficienții ecuației regresionale prezentate în Tabelul 1:

În figura 2 și figura 3 sunt prezentate histogramele pentru variabilele cantitative (Q lapte, a, b și c), valori obținute din baza de date *Ameliorare*.

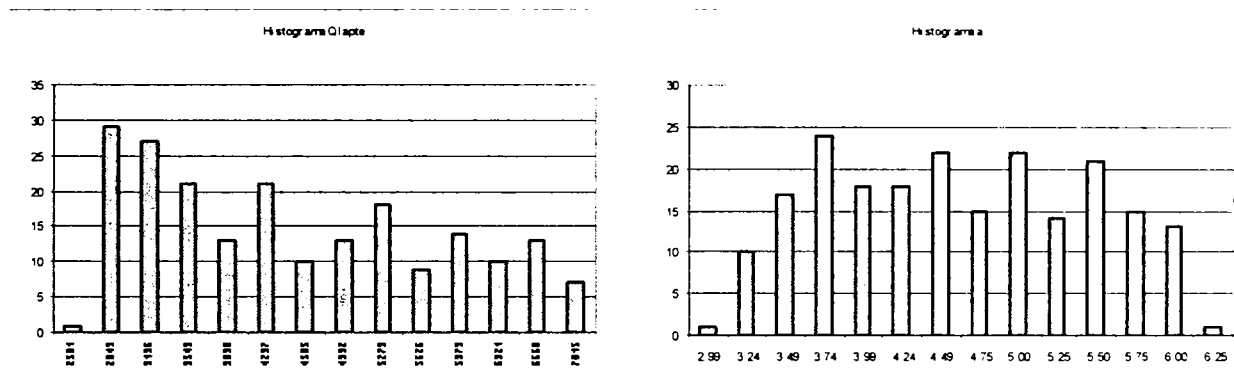


Figura 2 . Histograma variabilelor cantitative Q lapte și a - variabila asociată parametrului ce caracterizează producția corespunzătoare pantei ascendente a curbei de lactație

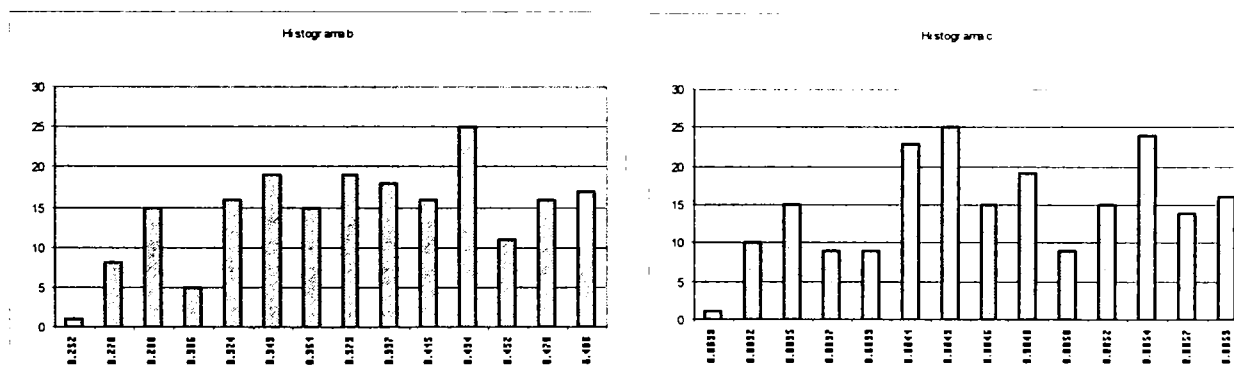


Figura 3 . Histograma variabilelor b, c asociate parametrului ce caracterizează producția corespunzătoare vârfului și pantei ascendente a curbei de lactație

Definiția modelului:

$$Y = a \cdot x_1 + b \cdot x_2 + c \cdot x_3 + d \cdot x_4 + e \cdot x_5 + f \cdot x_6$$

Variabilele de regresie

Tabelul 1

Variabila	Valoarea	Eroarea standard	t-test	Prob(t)
a	-270.04841	65.76683	-4.10614935	0.00006
b	-204.76254	48.12839	-4.25450633	0.00003
c	610.00809	34.07661	17.90108014	0
d	15892.92	438.4939	36.24433378	0
e	-793855.55	41102.94	-19.3138383	0
f	0.4504548	0.205037	2.196949045	0.02914

a, b, c, d, e, f reprezintă coeficienții variabilelor și pot fi interpretați ca pante ale dreptei de regresie $Y = f(x_i)$ considerând ca toate valorile x_j (cu $j > i$) rămân constante.

S-au testat următoarele ipoteze:

- I. **H₀**: variabila x_j nu este necesară, dat fiind că variabilele x_i ($i < j$) sunt incluse în model (ipoteza alternativă fiind: x_j este necesar în modelul pentru predicția valorilor y).

În acest caz, s-a folosit un test t (Student) parțial, calculat pe baza variației explicate de x_j , în afara de ceea ce este explicat de restul variabilelor.

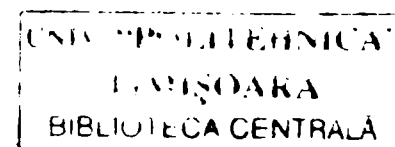
Concluzie:

Probabilitățile **Prob(t)** obținute, prezentate în tabelul 1, pentru toți coeficienții sunt $< 5\%$; ipoteza se respinge. Această analiză, conduce la concluzia ca variabilele cu acești coeficienți nu pot fi înlăturați din model.

- II. **H₀**: $a = b = c = d = e = f = 0$, sau, altfel spus, variabila dependentă Y nu depinde de nici una dintre variabilele independente X_i (ipoteza alternativă fiind: există cel puțin un coeficient diferit de zero).

Concluzie:

În acest caz, s-a folosit testul F (numit testul F global) (Analiza varianței), pentru care probabilitatea obținută este apropiată de zero (testul $F=269,534$ și **Prob(F) ≈ 0**), ca urmare probabilitatea ca ipoteza H_0 să fie adevărată este nulă, ipoteza alternativă fiind cea adevărată., Y (producția de lapte) depinde de factorii luați în studiu.



2.2.2. Analiza rezultatelor obținute prin procesul de modelare

Interpretarea coeficienților de corelare parțială

Calculul coeficienților parțiali de corelare corespunzători variabilelor independente din cadrul întregului model a condus la următoarele rezultate [DataFit01] :

Tabelul 2

Matricea corelațiilor

	X1	X2	X3	X4	X5	X6	Y
X1	1	0.05511	-0.024377	-0.063095	-0.08061	0.02314	-0.03882
X2	0.055111	1	-0.129388	0.032277	-0.01691	-0.05099	-0.04452
X3	-0.02438	-0.12939	1	-0.311324	0.11207	0.02800	0.24631
X4	-0.0631	0.03228	-0.311324	1	0.17816	0.22563	0.76438
X5	-0.08061	-0.01691	0.112074	0.178160	1	0.09995	-0.12621
X6	0.02314	-0.05099	0.028001	0.225625	0.09995	1	0.20344
Y	-0.03882	-0.04452	0.246307	0.764382	-0.12621	0.20344	1

Inspectând rezultatele obținute, sintetizate în tabelul 2, valorile din ultima linie a tabelului arată influența fiecărei variabile în cadrul modelului adoptat.

Aceste valori sunt mici, dar la acest nivel s-a considerat că este mai important de analizat valorile relative ale acestora în raport una față de cealaltă, și nu valorile numerice ca atare.

Reprezentarea grafică a valorilor relative ale coeficienților parțiali de corelare este ilustrată în figura 4.

Concluzii:

▪ O primă concluzie care rezultă se referă la apariția unei ierarhii a factorilor care concură la producția de lapte. Analizând în ordine descrescătoare se poate observa că :

1. Cea mai mare influență asupra variabilei dependente o are variabila asociată parametrului care caracterizează producția corespunzătoare vârful curbei de lactație. Influența sa este pozitivă, în valoare absolută depășește clar toate celelalte variabile prezente în model. Este urmată de variabila asociată parametrului ce caracterizează producția corespunzătoare panta ascendentă, pozitivă, de asemenea, dar legătura este mult mai slabă.
2. Valoarea de ameliorare a taurului tată, are o influență pozitivă, ponderea sa reprezintă 20% din variabilitatea lui Y. Această valoare se află în intervalul de variabilitate a heritabilității pentru caracterul lapte, ($h^2=0,25$ cu limitele de varianță 0,15 – 0,35).

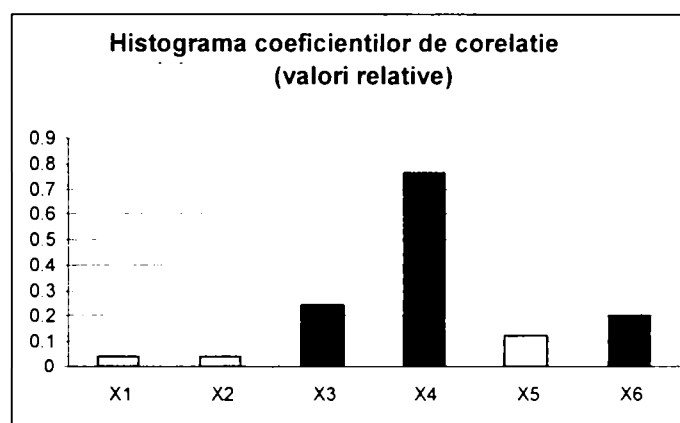


Figura 4. Histograma coeficienților de corelație parțială

3. Influența factorilor de mediu este prezentă, valorile lor sunt mici, dar ținând cont de faptul că aceste variabile sunt de natură calitativă, în model acestea sunt reprezentate ca variabile dihotomice, corectitudinea valorilor obținute pentru coeficienții de corelație nu este sigură.

▪ Reprezentarea sub formă de histogramă a valorilor absolute ale coeficienților parțiali de corelație, permite studierea fenomenului (producție de lapte) ca un ansamblu a mai multor componente, fiecare având o importanță mai mare sau mai mică.

Interpretarea coeficienților de intercorelare

Această analiză are ca scop stabilirea legăturilor ce există între factorii implicați. Este important să se stabilească și să se scoată în evidență în ce măsură un parametru (factor) depinde de altul. Deoarece este vorba de un singur fenomen, este normal ca acești factori să nu fie niște entități singulare, de sine stătătoare, ci între aceștia să apară fenomene de interdependență. Din analiza acestora pot apărea concluzii interesante relative la fenomenul studiat.

Interdependența între factorul genetic reprezentat de valoarea de ameliorare și ceilalți parametri

Procedând ca în cazul coeficienților parțiali de corelare, s-au reprezentat grafic sub formă de histogramă valorile relative ale coeficienților de corelare existenți între parametri. Valorile se regăsesc în figura 5.

Cea mai bună legătură există între factorul genetic reprezentat de valoarea de ameliorare și vârful, respectiv trendul descendent al curbei de lactație.

Acest lucru confirmă faptul că în evaluarea genetică a taurinelor de lapte, este necesară cuprinderea în analiză a valorilor cantităților de lapte pe întreaga curbă de lactație, valori oferite de testul zilei de control.

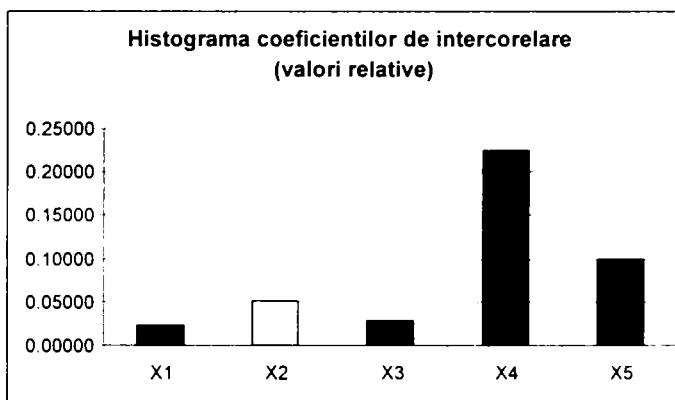


Figura 5. Histograma coeficienților de intercorelare

Concluzii:

1. Există legături între factorii implicați, acestea sunt exprimate de valorile coeficienților de intercorelație. Valorile obținute sunt mici, valoarea lor indică cât de puternică este legătura.
2. A fost analizată și interpretată doar valorile de intercorelație existente între variabila x_6 , corespunzătoare factorului genetic și ceilalți parametri. Aici s-au găsit cele mai importante valori. Legătura între această variabilă și cea asociată cu producția corespunzătoare vârfului curbei de lactație indică o legătură puternică.
3. Între factorii de mediu și trendul ascendent al curbei de lactație există cea mai importantă legătură.

Interpretarea coeficientului de corelare multiplă.

S-a calculat coeficientul de determinare multiplă (R^2), el exprimă ponderea cu care influențează parametri luați în studiu ($X_1, X_2, X_3, X_4, X_5, X_6$), variabila dependentă Y (producția de lapte).

$$R^2 = 0.86741075$$

Proporția varianței explicate = 86.741075%

Concluzie:

- Parametri luați în studiu explică 86.74 % din varianța variabilei independente Y (producția de lapte), restul de 12.26% pot fi explicați prin influența altor parametri care nu au fost luați în studiu. Dintre acești parametri cel mai important îl reprezintă furajarea care are o influență hotărâtoare în producția de lapte. Alcătuirea rațiilor și tehnica furajării reprezintă o problemă de management a fermei, eventual mărirea capacității de ingestie a furajelor de către vacile de lapte reprezintă la fel ca și producția de lapte o problemă de ameliorare genetică.
- Valoarea obținută pentru coeficientul de determinare multiplă este apropiat de 1, el reflectă eficiența modelului, și anume o bună alegere a factorilor incluși în model.
- Ponderea cea mai mare în variabilitatea lui Y o au parametri asociați cu producția corespunzătoare trendul ascendent și vârful curbei de lactație, urmată de ponderea factorului genetic.
- Ponderea variabilelor cantitative cu valori continue, (patru din șase), în studiu modelului, au contribuit la obținerea unui coeficient bun de determinare multiplă.

2.2.3. Calitatea și credibilitatea modelului și a datelor

Calitatea modelului este analizată în primul rând prin eficiența lui, iar valoarea obținută pentru coeficientul de determinare multiplă, coeficient ce indică eficiența modelului este foarte bun (0.86741075). Valoarea ridicată a acestui coeficient a fost influențată și de numărul mare de parametri luați în model.

În modelarea matematică, utilizarea regresiei liniare este condiționată de:

- condiția de lipsă a multicolarității între variabilele modelului;
- natura cantitativă a variabilelor.

Pentru modelul propus, aceste condiții sunt îndeplinite. Variabilelor ce caracterizează condițiile de mediu, sunt variabile nominale, reprezentate de variabile calitative, dihotomice, respectiv trihotomice. Teoria regresiei liniare acceptă aceste condiții pentru aplicare.

Credibilitatea modelului este susținută de următoarele argumente:

1. S-a obținut un coeficient de corelație parțială pentru variabila ce caracterizează factorul genetic foarte apropiată de valoarea heritabilității pentru caracterul lapte ($h^2=0,25$ cu limitele de varianță 0,15 – 0,35).
2. Trendul ascendent și vârful curbei de lactație care se atinge în medie în cea de-a 40-60 zi de lactație sunt elementele care caracterizează întreaga producție de lapte pentru o lactație. Aceste variabile au ponderea cea mai mare în variabilitatea lui y .

Credibilitatea datelor utilizate în model este asigurată în primul rând de îndeplinirea cerințelor pentru aplicarea testelor statistice, și anume:

1. număr suficient de mare de observații (212);
2. repartiție normală pentru variabila dependentă;
3. credibilitatea asupra valorii de ameliorare a taurilor . (valoarea de ameliorare a taurilor tați a fost calculată cu aplicația Ameliorare prin metodologia BLUP).

2.2.4. Concluzii

Producția de lapte, principalul caracter productiv al unor rase de vaci este influențat de o serie de factori printre care, potențialul genetic. Prima sursă de progres genetic o reprezintă taurii amelioratori, care, prin inseminare, pot asigura după cum am obținut din analiza modelului matematic, 20% din varianța producției de lapte. Alături de acest factor, există și alți factori, cum ar fi furajarea, factor de altfel hotărâtor în realizarea unei producții mari de lapte, dar există o multitudine de factori de mediu, factori care pot fi controlați.

Modelul matematic construit ia în studiu șase factori, doi dintre aceștia sunt factori de mediu, valoarea de ameliorare a taurului și factori ce caracterizează curba de lactație

Datele existente în baza de date au permis selectarea datelor necesare construirii modelului, eșantionul luat în studiu asigură credibilitatea modelului.

Factorii de mediu sunt considerate variabile calitative, ordinale, care în model au fost introduse prin categorizarea lor. Modelul conține 4 variabile cantitative ce caracterizează factorul genetic și parametri asociați producției de lapte corespunzătoare pantei ascendente, vârfului și pantei descendente a curbei de lactație.

Pentru construirea modelului matematic s-a ales soluția unui model regresional multiliniar în care ponderea fiecărui parametru este egală cu ponderea celorlalți parametri.

Din studiul efectuat pe eșantionul luat în studiu cu ajutorul modelului construit, au rezultat concluzii care se pot rezuma astfel :

1. Probabilitatea coeficienții variabilelor ce intervin în model să fie nuli este sub 5%.
2. Există legături între toți factorii existenți în model, valorile sunt mici, există corelații pozitive și negative, dar acestea sunt ne semnificative.

3. Coeficientul de determinare multilinară calculat arată că 86,74% din variabilitatea variabilei dependente (producția de lapte) este asigurată de factorii luați în studiu
4. Cocluzia 3 se confirmă prin faptul că heritabilitatea producției de lapte se găsește într-un interval ce conține și valoarea calculată din modelul propus.
5. Trendul ascendent și vârful curbei de lactație care se atinge în media în cea de-a 60 zi de lactație, reprezentate în model prin variabilele x_3 și x_4 sunt elementele a căror pondere în variabilitatea lui y , au valorile cele mai mari, legătura între aceste variabile și variabila dependentă este pozitivă.
6. Factorii de mediu influențează prea puțin variabila dependentă, ei influențează producția corespunzătoare pantei ascendente, vârfului și pantei descendente ale curbei de lactație și ca urmare indirect producția de lapte. La acești factori se poate renunța.

2.3. Predicția valorii de ameliorare utilizând modelul propus.

Predicția valorii de ameliorare pentru caracterul producție lapte s-a făcut aplicând metodologia BLUP, folosind aplicația „Ameliorare” și baza de date ce conține caracterele productive ale celor 212 vaci, fiice ale celor 10 tauri luați în studiu.

Pentru aplicarea metodologiei BLUP, se consideră că modelul liniar propus cuprinde: efectul fix al factorilor de mediu, efectul genetic aditiv al animalului (valoarea de ameliorare) și efectul curbei de lactație ca efecte aleatoare.

Ecuția modelului rescrisă conform cerințelor exprimate mai sus:

$$y_{ij} = s_i + a_j + \sum_{k=1}^3 p_{kj} + e_{ij}$$

y_{ij} - performanța individului j , realizată în mediul i

s_i - efectul fix al factorilor de mediu;

a_j - efectul genetic aditiv al animalului (valoarea de ameliorare);

p_{kj} - efectul datorat curbei de lactație

e_{ij} - efect de mediu temporar particular fiecărei observații. (reziduale)

Pentru efectele aleatoare se consideră că au o distribuție normală cu media 0 și varianțele: σ_a^2 , σ_{p1}^2 , σ_{p2}^2 , σ_{p3}^2 , σ_e^2

Sub formă matriceală modelul devine:

$$y = Xb + Za + Zp_k + e$$

Cu $M(y) = Xb$ și $Var(y) = V = ZAZ' \sigma_a^2 + Z'Z \sigma_{pk}^2 + I \sigma_e^2$

Ecuțiile modelului sunt:

$$\begin{bmatrix} X'X & X'Z & X'Z \\ Z'X & Z'Z + A^{-1}k_a & Z'Z \\ Z'X & Z'Z & Z'Z + I k_p \end{bmatrix} \times \begin{bmatrix} \hat{b} \\ \hat{a} \\ \hat{p} \end{bmatrix} = \begin{bmatrix} X'y \\ Z'y \\ Z'y \end{bmatrix}$$

$$k_a = \frac{\sigma_e^2}{\sigma_a^2}; \quad k_{p_i} = \frac{\sigma_e^2}{\sigma_{p_i}^2}, \quad i = 1, 2, 3$$

$\hat{a}_j + \hat{p}_{jk}$ sunt considerate o măsură a capacității de producție a individului j ;

Evaluarea genetică a indivizilor s-a făcut în situația în care taurii nu au fost selecționați pe descendenți.

Față de modelele clasice, modelul propus conduce la o rezolvare prin metodologia BLUP cu un număr mai mare de ecuații.

Valoarea de ameliorare a unui reproducător poate fi continuu recalculată pe măsură ce noi fiice intră în producție.

La fel ca și cazul celorlalte modele, metodologia BLUP asigură compararea indivizilor în timp (pe durata unei lactații și diferite lactații în ani diferiți) și în spațiu (animale aflate în ferme diferite).

2.4. Compararea modelului propus cu modele existente

Modelele prezentate în capitolul 1 al lucrării, considerate modele clasice (modelele tata, modelul animal individual), modelul „testul zilei de control”, precum și modelul propus în lucrare de față, au fost dezvoltate și folosite pentru evaluarea genetică a taurinelor de lapte. Evaluarea genetică se concretizează prin „predicția” cât mai corectă a valorii de ameliorare a indivizilor selecționați.

Modelele prezentate sunt modele mixte (conțin efecte fixe și efecte aleatoare), în vederea estimării efectelor fixe și predicției efectelor aleatoare se folosește metodologia BLUP.

Modele clasice (modelul tata, modelul animal individual) utilizează sursele de informație (ascendenți, performanță proprie, colaterali, descendenți) pe o cale optimă, indiferent de numărul caracterelor considerate.

Modelul testului zilei de control, așa cum susțin autorii [Swa95, Ali87], consideră că acest model asigură o modelare mai apropiată de realitate. Acest model, pentru evaluarea genetică nu ia în calcul întreaga cantitate de lapte pe lactație ci valorile zilnice măsurate (DIM), astfel sunt surprinse toate efectele genetice și de mediu care influențează valoarea fenotipică. Acest lucru solicită luarea în studiu de **grupuri de contemporane** (aceleași condiții de mediu, sezon, vârstă, lactație, etc.), astfel încât comparația și predicția valorilor genetice a animalelor să se facă în același mod. O vacă care are o producție de lapte de 45 de kg de lapte în ziua de test este comparată cu producția altor vaci din aceeași lactație, contemporane cu ea, aflate în aceeași fermă, în ziua respectivă de test. Acest lucru presupune o evidență foarte riguroasă, dar asigură o creștere a acurateții în predicția însușirilor genetice.

Producția de lapte și componentele acesteia (grăsime, proteină, etc.), variază funcție de stagiul lactației precum și de ziua de control (DIM). Dacă producția de lapte a unui grup de vaci nu este măsurată utilizând același condiții (aceiași zi de test, condiția de contemporaneitate, de timp și mediu, etc.) se pot introduce erori în predicția producției de lapte.

Realizarea grupului de contemporane care intra în analiză prin metoda zilei de control este destul de dificilă în condițiile în care nu există o evidență strictă a producției de lapte ce ține seama de toți factorii care reprezintă criteriile de creare a grupurilor de contemporane. Literatura de specialitate [Mu00] prezintă care sunt acești factori: *TRAPS* (timp, regiune, vârstă, paritate de lactație, sezon) și precizează că pentru toți acești factori se fac ajustări. Valoarea de ameliorare este calculată comparând producția de lapte (grăsime, proteine, celule somatice) cu a tuturor celorlalte contemporane aflate în grup. Este necesară calcularea valorii medii a grupului de contemporane pentru caracterul luat în studiu, apoi are loc o comparare a producției zilnice a fiecărei vaci cu media grupului. Pentru aplicarea acestui model, constituirea grupului de contemporane se face pentru fiecare zi de test, ținând cont de condițiile specifice zilei respective, în acest fel comparația are loc mult mai precis. Realizarea acestor condiții în practică este foarte dificilă în condițiile în care nu există o infrastructură care să asigure culegerea surselor de informații așa cum solicită modelul.

Modelul propus în lucrare, pornește de la modelul clasic (modelul tată), efectele fixe sunt considerate efectele de mediu iar ca efecte aleatoare sunt considerate efectul reproducătorului și efectele datorate curbei de lactație caracterizată prin cei trei parametri.

Introducerea în model a caracteristicilor curbei de lactație ca efecte aleatoare, apropie modelul propus de modelul testului zilei de control, model utilizat în prezent pentru evaluarea genetică a vacilor de lapte. Valoarea genetică este mai exact calculată dacă sunt utilizate ca surse de informare caracteristici ale curbei de lactație.

Față de modelul testului zilei de control, calculul valorii de ameliorare utilizând modelul propus nu presupune crearea grupului de contemporane. Crearea grupului de contemporane, așa cum am prezentat este anevoioasă în condițiile în care nu există o evidență strictă a producției zilnice de lapte. Calculele necesare a fi efectuate pentru estimarea valorii de ameliorare utilizând metoda testului zilei de control solicită costuri mari, care nu întotdeauna se justifică.

Utilizând aplicația „Ameliorare”, pe același lot de animale, s-au calculat valorile de ameliorare pentru cei 10 tauri utilizând modelul clasic (modelul tată – descendenți întreținuți în diferite condiții de mediu) și modelul propus. Valorile obținute diferă ca valoare numerică dar, ierarhizarea reproducătorilor este asemănătoare.

Se disting tauri amelioratori pentru producția de lapte (valori pozitive) și înrăutățitori (valori negative). Rezultatele obținute sunt prezentate în tabelul 3.

Tabelul 3

Ierarhizarea reproducătorilor după rezultatele obținute

	Model clasic		Model propus
T3	59.31	T3	81.35
T1	28.91	T1	71.025
T9	26.88	T9	52.94
T5	18.21	T7	11.23
T7	3.28	T5	9.21
T6	-4.50	T10	0.25
T10	-15.50	T2	-12.5
T2	-33.66	T6	-17.5
T8	-34.96	T8	-64.15
T4	-47.95	T4	-74.15

Prin ambele metode, taurii T3 și T1 sunt cei mai puternic amelioratori iar cel mai înrăutățitor este taurul T4. Prin modelul propus, taurul T10, apare ca ameliorator al caracteristicii producție lapte, față de modelul clasic unde înrăutățitor. Compararea rezultatelor folosind teste statistice (testul t) a condus la obținerea unei probabilități de $p=0,52$ ceea ce semnifică acoperire din punct de vedere statistic.

3. IMPLEMENTAREA SISTEMULUI DE AMELIORARE

3.1. Structura bazei de date pentru aplicația *Ameliorare*

Aplicarea în practică a conceptelor moderne de ameliorare genetică este limitată de factorul computațional. Acesta se reflectă prin:

1. lipsa unei baze de date complete ce stochează informațiile necesare calculelor;
2. sacrificiul de precizie la care se recurge prin simplificarea modelelor folosite și în complexitatea programelor de calcul utilizate.

Folosirea în practică a unor modele complexe, deși teoretic fundamentată, determină costuri prohibitive prin puterea mare de calcul solicitată și prin realizarea infrastructurii absolut necesare în vederea culegerii și realizării bazelor de date ce stochează informațiile necesare calculelor.

Existența unei baze de date complete este imperios necesară, atât pentru o evidență clară în domeniu, cât și pentru asigurarea suportului de informații pentru studiul fenomenelor în domeniu [Po98]. Baza de date trebuie să conțină informațiile care răspund unei game variate de cerințe din partea utilizatorilor, cerințe materializate prin aplicații care vor lucra cu aceste informații. Aplicațiile pot atinge mai multe obiective sau pot să fie specializate pe o anumită problemă.

Principalele obiective ale aplicațiilor din domeniul ameliorării animalelor care solicită informațiile existente în baza de date ar putea fi:

- estimarea componentelor varianței;
- evaluarea genetică a animalelor;
- construirea hărților genetice;
- organizare și suport în exploatare și selecție;
- crearea de facilități pentru procesul de învățământ și cercetare.

Obiectivele aplicațiilor din domeniul ameliorării, enumerate mai sus, se pot referi la specii diferite de animale, specia taurinelor este cea mai intens studiată dintre speciile de animale domestice, și din aceste considerente, proiectarea bazei de date s-a făcut pentru această specie.

Informațiile primare se referă:

- identificarea animalului (cod matricol, sex, rasa, data nașterii, etc.), ascendenții săi (mama, tata), localizare (ferma, proprietar);
- caracterele productive-lapte (lapte, grăsime în lapte, proteină, etc.)
- caractere productive-carne (greutate la fătare, spor mediu zilnic, etc.)
- caractere de exterior (dezvoltare corporală, membre, uger, etc.)
- caractere reproductive (durata gestației, service period, calving interval, taur partener, cod matricol produs, etc.)
- identificarea fermă (exploatației), proprietar, însămânțător, bonitor, controlor, etc.

Utilizatorii bazei de date se împart în două categorii: cei care întrețin baza de date și cei care o exploatează cu una din aplicațiile dezvoltate.

În lucrarea de față a fost implementată o aplicație(C#.NET) care permite calculul valorii de ameliorare a taurilor pentru performanțelor fiicelor aflate în testare. Valoarea de ameliorare este calculată după metodologia BLUP pentru un caracter, indiferent care este acesta (caracter productiv sau caracter de exterior). Modelul matematic utilizat este modelul **tată (sire model)** *pentru descendenți întreținuți în condiții diferite de mediu*.

Diferența între acest model (sire model) și celelalte prezentate, mai puțin modelul testului zilei de control (Test Day Model) constă în calculul matricei relațiilor genetice dintre indivizi (matricea A). În cazul modelului implementat, nu se ține seama de înrudirile dintre animale.

Scopul realizării acestei baze de date și apoi a implementării aplicației, a fost acela de a obține datele necesare în vederea dezvoltării unui model care să permită **analiza și simularea** caracterului productiv „producția de lapte”, funcție de factori de mediu, valoarea de ameliorare și parametri asociați cu producția corespunzătoare pantei ascendente, vârfului și pantei descendente a curbei de lactație[Bet97].

Baza de date a fost proiectată să stocheze informații care să permită calculul valorii de ameliorare folosind modelul testului zilei de control, model care se folosește la ora actuală pe plan mondial. În cazul utilizării modelului testului zilei de control, metodologia obținerii valorii de ameliorare este metodologia BLUP. diferența constă în datele care intră în calcul.

Stabilirea structurii bazei de date, a reprezentat o primă etapă în proiectarea bazei de date[Io04]. Ea a fost stabilită în urma unei riguroase analize a activităților urmărite a se realiza în aplicație, precum și în perspectiva dezvoltării de aplicații ulterioare.

Schema conceptuală a bazei de date este prezentată în figura 5 .

Baza de date astfel concepută conține toate informațiile necesare aplicației implementate și a dezvoltărilor ulterioare ale acesteia. Informațiile existente în baza de date sunt în concordanță cu legislația privind marcarea, identificarea, înregistrarea și mișcarea animalelor de fermă stabilite prin directive și reglementări ale organelor legislative românești și ale Uniunii Europene[Un90].

În ANEXA 4 sunt cuprinse diagramele rezultate în urma analizei cu UML a activităților urmărite pentru baza de date și aplicația „Ameliorare”[Un00, Mi03, [Bru03][Brudiu03].

Au fost analizate următoarele activități:

- Înregistrarea exploatației;
- Înregistrarea taurinei;
- Managementul reproducției;
- Stabilirea valorii de ameliorare.

Pentru acestea au fost stabilite diagrama cazurilor de utilizare și unde a fost necesar, diagramele de secvență și de activități [Together01].

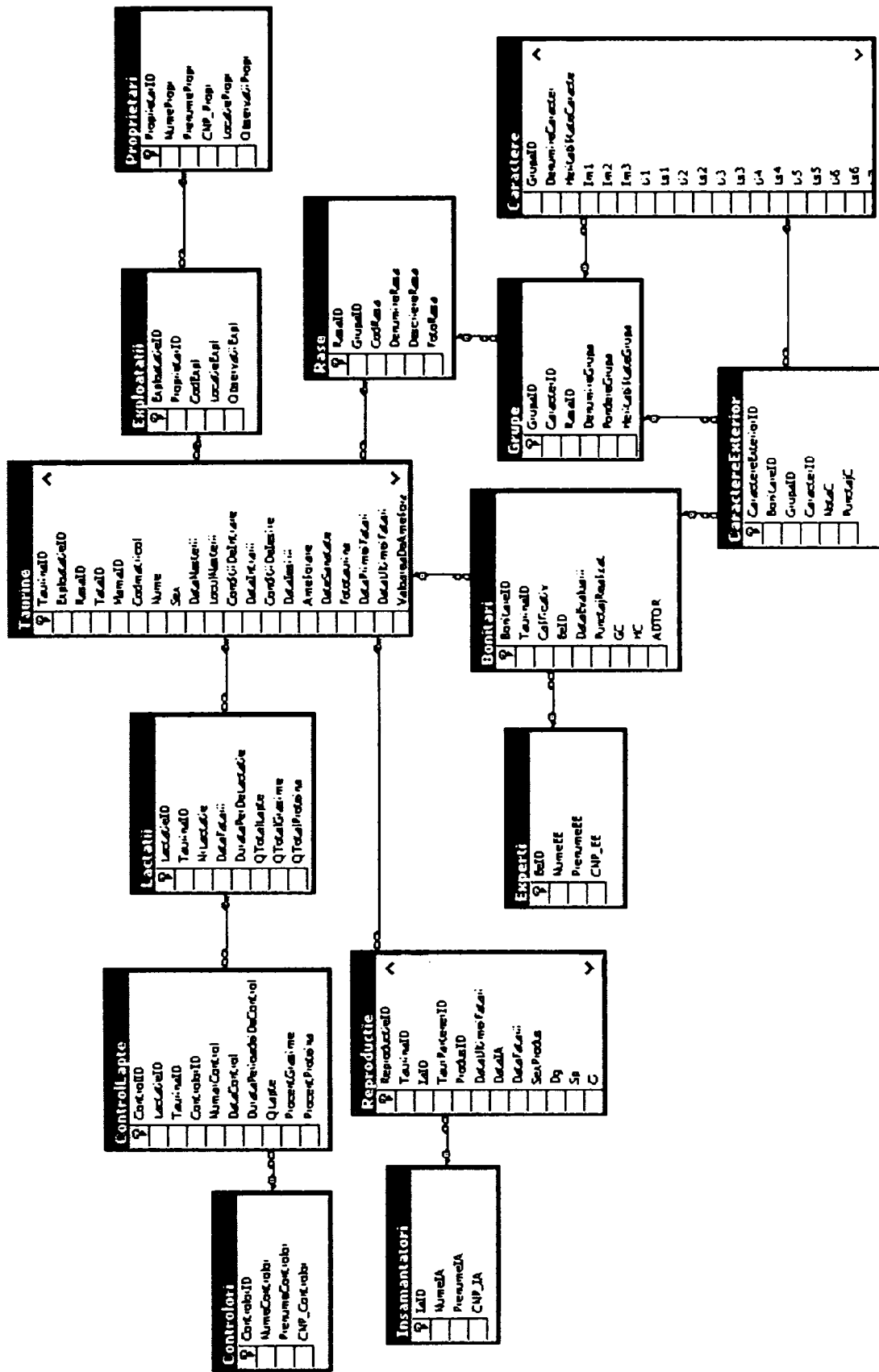


Figura 5. Baza de date *Ameliorare*, schema conceptuală.

3.2. Modelul datelor

Studiul asupra istoriei bazelor de date este centrat în jurul problemei modelării datelor. Bazele de date sunt colecții de date, iar un model de date stabilește regulile de organizare și interpretare a acestora. Există concepte matematice ce exprimă proprietățile statice și dinamice ale unei colecții de date.

Orice model de date se bazează pe trei componente: structurile de date, constrângerile de integritate și operatorii de manipulare a datelor [Co70].

În evoluția bazelor de date două dintre modele sau evidențiat clar, considerându-le la ora actuală de referință în istoria bazelor de date. Acestea sunt *modelul relațional* și *modelul obiectual*.

Locul și rolul celor două modele în evoluția bazelor de date este ilustrat în figura 6.

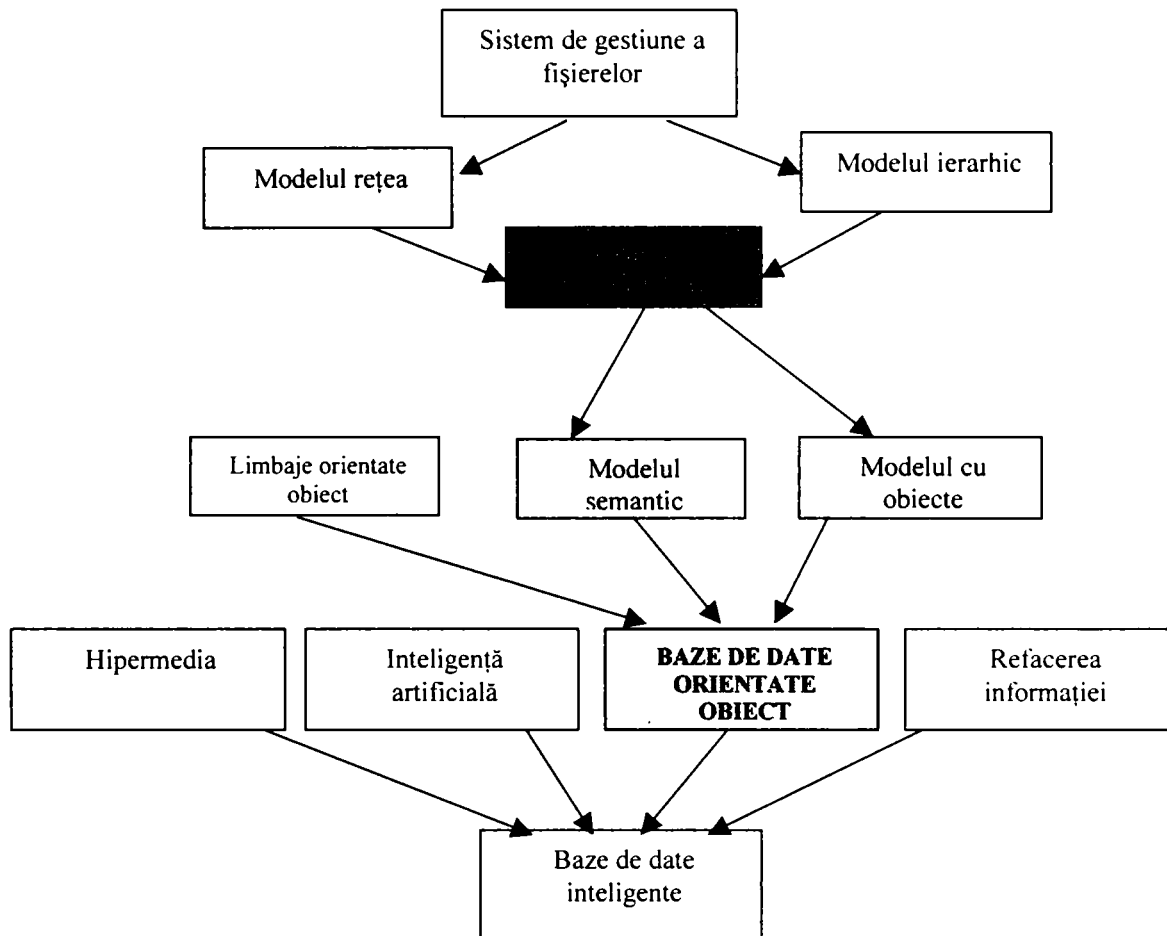


Figura 6. Modelul datelor în evoluția bazelor de date.

3.3. Modelul de date relațional

Modelul relațional al datelor se bazează pe noțiunea de relație din matematică, care corespunde unei mulțimi de entități de același tip și are o reprezentare ușor de înțeles și de manipulat, ce constă dintr-un tabel bidimensional, compus din linii și coloane.

Relația este o mulțime de entități iar tabelul este reprezentarea vizuală a acesteia.

Modelul relațional de baze de date este un model de date formal, el a fost definit de E. Codd în "*A Relational Model of Data for Large Shared Data Banks*" [Co70] și se bazează pe principiile *algebrei relaționale*.

O bază de date relațională este formată dintr-o mulțime finită relații (tabele, fișiere), fiecare relație reprezentând un tip de entitate sau o asociere dintre două sau mai multe tipuri de entități [Co90].

Asupra relațiilor se pot aplica o colecție de operatori care permit manipularea datelor memorate în tabele.

O tabelă (relație) este formată din rânduri și coloane, în algebra relațională n -upleți (rânduri) și constituanți (coloane).

Constituanții sunt informații elementare dintr-o relație. Domeniul constituanților reprezintă mulțimea valorilor pe care le poate lua un constituant. Domeniul se poate defini separat ca un tip abstract. Mai mulți constituanți pot avea același domeniu. Domeniul constituanților sunt valori atomice.

O relație se definește prin:

- n -upletul de constituanți (x_1, x_2, \dots, x_n) ;
- domeniul relației pentru fiecare constituant;
- un predicat care permite pentru orice n -uplet (a_1, a_2, \dots, a_n) , unde a_i aparține domeniului (x_i) să aibe o valoare adevărată sau falsă.

Această relație se notează $R(x_1, x_2, \dots, x_n)$ sau $R(x)$ și este formată din ansamblul n -upleților pentru care predicatul dă propoziții adevărate.

Caracteristicile relației sunt:

- N -upleții în relație sunt neordonați (relația se memorează în fișiere, n -upleții fiind înregistrările fișierului). O relație poate fi vizualizată în orice ordine a rândurilor, poate fi ordonată după fiecare atribut al său.
- Valorile atributelor în n -upleți sunt ordonate. Într-o relație, attributele în n -upleți apar în aceeași ordine (o mulțime ordonată de valori).
- Valorile atributelor în n -upleți sunt atomice. Un atribut în n -uplet nu poate avea decât o valoare. Ea poate fi nulă, să nu existe sau să nu se aplice n -upletului în cauză.

Algebra relațională cuprinde o colecție de operatori. Operatorii se aplică asupra relației în vederea manipulării datelor. În urma aplicării operatorilor rezultă o nouă relație (tabelă).

Operatorii relaționali sunt:

- **selectia** n -upleților din relație care îndeplinesc o anumită condiție;
- **proiecția** retine numai anumite coloane dintr-o relație conform unei liste.

- **combinarea** n-upleților (*JOIN*) din mai multe relații pentru a putea răspunde la întrebări pe baza unor condiții între coloane union compatibile din cele două relații;

Relațiile rezultate pot fi supuse la noi operații.

Operatorii pot fi împărțiți în două categorii:

- operatori din teoria mulțimilor (*UNION, INTERSECT, DIFFERENCE, PRODUS CARTEZIAN*);
- operatori speciali ai algebrei relaționale (*SELECT, PROJECT, JOIN*)

Reprezentarea relațiilor prin tabele

Un tabel este o reprezentare a unei relații și este compus din următoarele:

- Numele tabelului, care este identic cu numele relației pe care o reprezintă;
- Un număr de coloane egal cu numărul de atribute ale relației, fiecare coloană reprezentând un atribut;
- Capul tabelului, în care se înscriu numele atributelor relației, fiecare atribut fiind înscris în coloana corespunzătoare;
- O mulțime de linii, fiecare linie corespunzând unui tuplu, deci unei entități; în fiecare element al unei linii se înregistrează valoarea atributului corespunzător coloanei în care se află elementul respectiv.

În implementările reale, valorile atributelor sunt câmpuri (*fields*) memorate într-o anumită ordine în cadrul înregistrării (*record*) corespunzătoare unui tuplu.

Relațiile unei baze de date reflectă realitatea modelată și de aceea valorile pe care le conțin trebuie să respecte anumite reguli, care corespund celor din realitate. **Constrângerile de integritate** sunt regulile care se definesc la proiectarea unei baze de date și care trebuie să fie respectate de orice stare a acesteia [Si99][Fo97].

O relație este definită de ca o mulțime de tupluri, ele trebuie să fie distincte, nu pot exista într-o relație două tupluri care să conțină aceeași combinație de valori ale tuturor atributelor.

Supercheia (S(K)) este un grup de atribute care determină univoc n-upleții unei relații. Pot exista relații la care S(K) este formată din toate atributele.

Cheia (K) este o supercheie minimală. Ea are proprietatea că, înlocuind un atribut al ei cu un altul, nu mai este supercheie a relației respective.

Cheia primară (PK *Primary Key*) este aleasă dintre cheile pentru identificarea înregistrărilor. Se alege să fie cu un singur atribut sau un număr redus de atribute, să aibe lungime minimă și să aibe sens funcțional.

Asocierile între tipurile de entități definite în modelul conceptual al unei baze de date se realizează în modelul relațional prin intermediul cheilor externe (străine)[Mil03].

Cheia externă (FK *Foreign Key*) este un atribut sau un grup de atribute care este cheie primară într-o altă relație și servește la realizarea unor legături între două relații.

Pe lângă avantajul unui model de date precis și simplu, sistemele de baze de date relaționale mai beneficiază și de un limbaj de programare unanim recunoscut și acceptat, limbajul SQL (*Structure Query Language*). Limbajul SQL este limbajul utilizat de majoritatea sistemelor de baze de date relaționale pentru definirea și manipularea datelor.

SQL, este un limbaj de cereri *non-procedural* care permite accesul la date, precizând *ce* trebuie obținut precizând condițiile[Pe95].

Cu toate că tehnologia relațională a introdus limbajul SQL și el a devenit un standard pentru interogarea bazelor de date, limbajul nu este dependent de tehnologia relațională, el fiind utilizat și în alte tehnologii (orientat obiect, relațional-obiectual). Sistemele de gestiune relaționale asigură că nici un utilizator și nici o aplicație nu pot modifica baza de date, dacă modificarea este în contradicție cu constrângerile de integritate.

Sisteme de gestiune bazelor de date relaționale

Cele mai răspândite sisteme de gestiune a bazelor de date relaționale sunt:

Sistemul Oracle

Sistemul Oracle este un sistem de gestiune a bazelor de date multiutilizator puternic, cu implementări pe toate platformele (Windows, Unix, Linux), care oferă atât performanțe de execuție ridicate, cât și un grad de protecție și securitate a datelor. În toate versiunile, Oracle oferă implementarea completă a caracteristicilor modelului relațional (conform standardului SQL2), iar în ultimele versiuni (Oracle8i, Oracle9i și Oracle 10g) sunt sisteme de gestiune a bazelor de date distribuite, implementând extensiile orientate obiect prevăzute de standardul SQL3 și oferă posibilitatea de dezvoltare a bazelor de date distribuite[Lu95].

Microsoft SQL Server

SQL Server este sistemul de gestiune a bazelor de date relaționale dezvoltat de firma Microsoft pentru sistemele de operare Windows [Microsoft200].

Microsoft SQL Server 2000 (versiunea ianuarie 2004) este un sistem de gestiune a bazelor de date relațional (RDBMS), bazat pe limbajul SQL (*Structured Query Language*), suportă complet standardul SQL2, este scalabil și are integrat limbajul XML (*Extensible Markup Language*), suport pentru aplicațiile internet.

Limbajul utilizat de **Microsoft SQL Server 2000** este **Transact-SQL (T-SQL)**, un dialect al limbajului SQL.

SQL Server 2000 integrează suport complet pentru XML, asigurând o modalitate eficientă de transfer a datelor și posibilități de integrare în mediile care presupun existența unor sisteme informatice eterogene.

Versiunea 2000 integrează tehnologiile XPath și URL Query, simplificând procesele de transfer a datelor în format XML. De asemenea, limbajul procedural Transact SQL (T-SQL) a fost întregit cu noi funcții care facilitează aceste operațiuni.

Microsoft SQL Server 2000:

- suportă accesul în același timp a mai multor utilizatori la baza de date prin intermediul instanțelor bazei de date. Aplicații ce rulează pe calculatoare separate, utilizează componentele de comunicare ale serverului pentru a transmite comenzi instanței bazei de date. În urma conectării (sqlConection), aplicația poate referii datele instanței bazei de date funcție de drepturile de autorizare pe care le are.
- suportă traficul Web precum și conexia a sute de mii de utilizatori în același timp. Tabelele în SQL Server 2000 pot fi partiționate pe mai

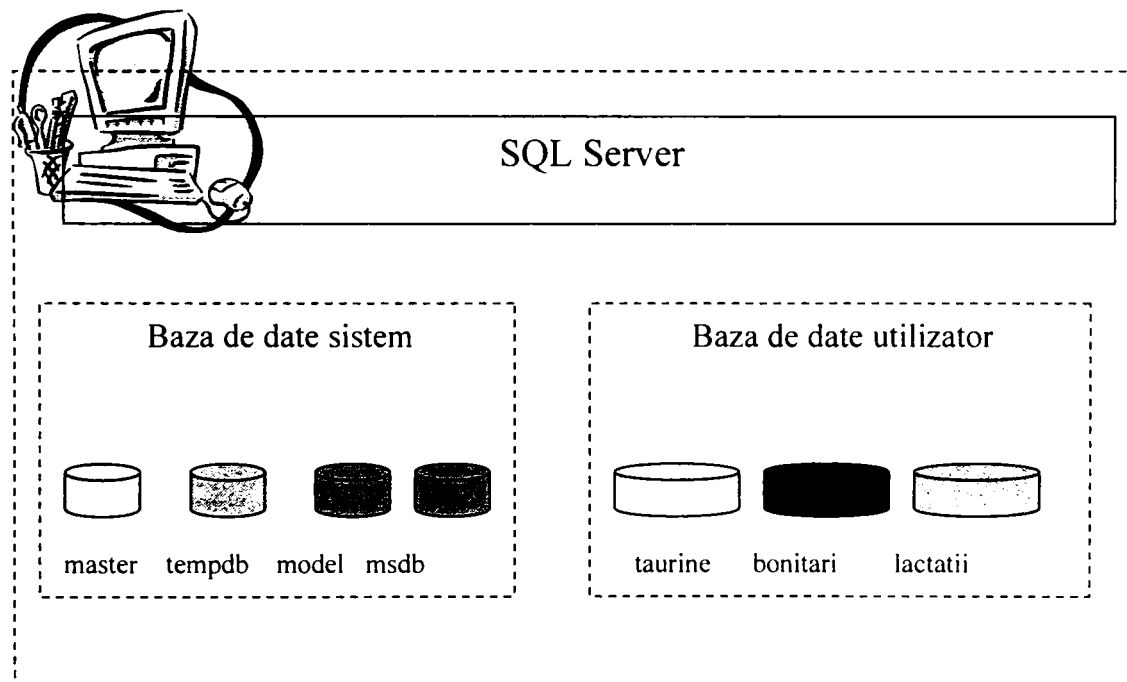
multe servere, calculatoarele multiprocesor cooperează în vederea procesării cererilor foarte mari de date. Grupurile de servere de baze de date sunt numite federații.

- permite să lucreze și în condițiile în care atât baza de date cât și aplicația ce lucrează cu aceasta se găsesc pe același calculator, reducând considerabil cerințele cu privire la resursele solicitate.
- prezintă un nivel ridicat de disponibilitate a bazei de date. Versiunea 2000 oferă posibilitatea de realizare online a backup-ului bazei de date, fără a fi afectate procesele de business derulate prin intermediul acestui server.
- oferă avantaje importante în arhitecturile bazate pe clustere de servere, asigurând un nivel deosebit de disponibilitate și siguranță în exploatarea datelor. Tehnologia Virtual Interface System Area Network (VI SAN) permite integrarea directă a SQL 2000 în cadrul arhitecturilor SAN, fiind asigurat un nivel ridicat de performanță.

Arhitectura bazei de date

În MS SQL Server 2000 datele sunt memorate în baze de date. Există două aspecte cu privire la implementarea arhitecturii bazei de date, cea logică, vizibilă pentru utilizator, organizată în mai multe componente (tabele, vederi, proceduri), și cea fizică. În mai multe fișiere pe disc. Implementarea fizică este transparentă pentru utilizator, administratorul bazei de date este cel care sesizează și lucrează cu în mod concret cu fișierele sistem ale bazei de date.

Fiecare instanță a SQL Server are patru fișiere sistem (master, model, tempdb și msdb) și una sau mai multe baze de date utilizator.



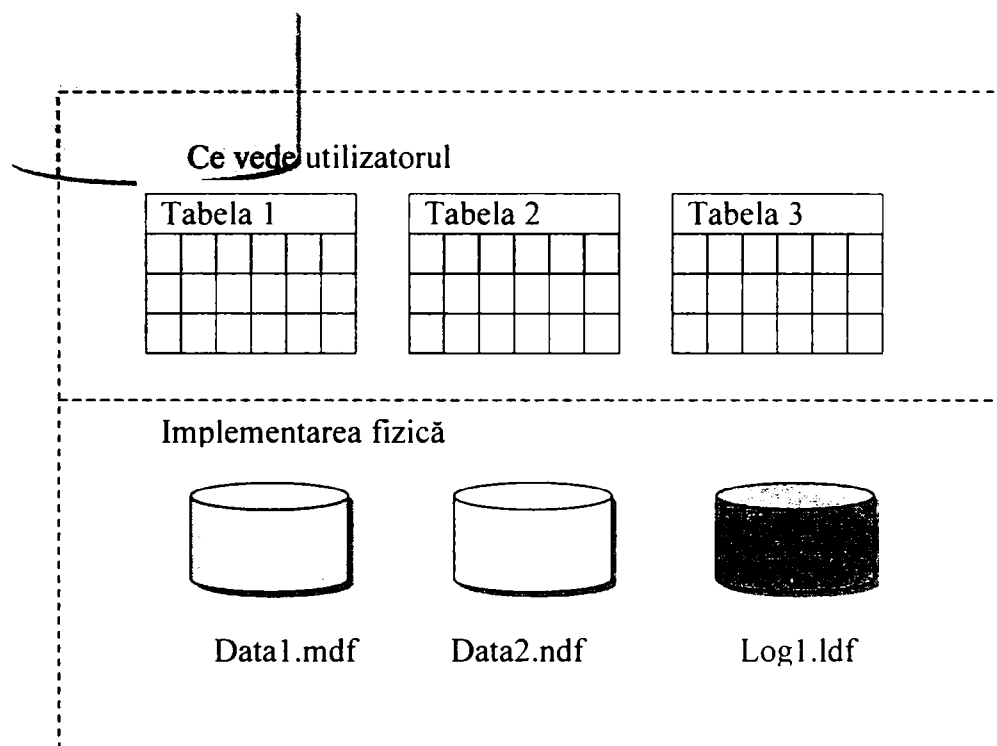
Trăsături caracteristice

Microsoft SQL Server 2000 extinde performanțele, siguranța, calitatea și ușurința în lucru a versiunilor anterioare completându-le pe acestea cu altele, asigurând o excelentă platformă pentru tranzacțiile online (OLTP online transactional processing), depozitarea datelor și aplicații e-commerce.

SQL Server 2000 are asigurată integrarea în internet prin:

- Limbajul XML
- English Query și Microsoft Search Service
- Modelul de programare a SQL Server 2000 este integrat în Windows DNA (Windows Distributed interNet Applications Architecture, platforma Microsoft pentru dezvoltarea aplicațiilor Web ce asigură soluții securizate, fiabile de mare scalabilitate care facilitează integrarea sistemelor și aplicațiilor eterogene.)

Baza de date A



Scalabilitatea și disponibilitatea este asigurată prin faptul că baza de date funcționează pe platforme pornind de la laptop-uri cu Microsoft Windows 98 și până la sisteme multiprosesor cu Microsoft Windows 2000 Data Center Edition.

3.4. Modelul de date orientat obiect

Modelul obiect (*Object Model*) este un concept unificator în știința calculatoarelor, fiind aplicabil în programare, în proiectarea hardware-ului, a interfețelor, a bazelor de date, etc.[At95]

Aplicațiile (*software*) scrise în ultimii ani, indiferent de domeniul de aplicativitate, folosesc modelul orientat obiect. Acest lucru s-a extins și asupra aplicațiilor ce lucrează cu bazele de date. Bazele de date și programarea orientată pe obiecte reprezintă o copie modul de funcționare al creierului uman și ca urmare se aștepta o extindere a modelului orientat obiect și asupra bazelor de date. Dacă modelul relațional a stat la baza sistemelor de gestiune a bazelor de date relaționale, în etapa următoare, programarea orientată obiect și bazele de date au condus la sistemele de gestiune a bazelor de date orientate obiect.

Și în cazul modelului orientat obiect, la fel ca și în cazul modelului relațional regăsim componentele: structuri de date, constrângeri de integritate și operatorii de manipulare a datelor.

Modelul orientat obiect al datelor, constă într-o colecție de:

- proprietăți statice (structuri de date):
 - obiecte;
 - attribute;
 - relații,
- reguli de integritate asupra obiectelor și operațiilor
- proprietăți dinamice: operații sau reguli de definire a unei noi structuri pornind de la vechea structură.

Bazele de date orientate obiect au apărut și au evoluat în urma dezvoltării programării orientate obiect. Îmbinarea programării orientate obiect cu funcțiile sistemelor de gestiune a bazelor de date au condus la sistemele de gestiune a bazelor de date orientate obiect, sisteme care culeg avantajele orientării obiect în dezvoltarea sistemelor software complexe.

În programarea orientată obiect, programele sunt organizate ca și colecții de obiecte cooperante, fiecare obiect fiind o instanță a unei clase. Fiecare clasă reprezintă abstractizarea unui tip de entitate din realitatea modelată, iar clasele sunt membrele unei ierarhii de clase, corelate între ele prin relații de moștenire. Orice obiect este încapsulat, reprezentarea lui (structura internă a obiectului) nu este vizibilă utilizatorilor, aceștia au acces doar la funcțiile (metodele) pe care acel obiect este capabil să le execute[Ma93].

Necesitatea introducerii bazelor de date orientate obiect

Apariția bazele de date orientate obiect este motivată de următoarele[Ne97]:

- nevoia de acces la obiecte persistente prin programe scrise în limbaje orientate obiect.
- noile cerințe din domeniile tehnologice noi apărute, cum ar fi ingineria sau multimedia.

Oricât de folositor este modelul relațional pentru realizarea bazelor de date, există unele domenii (în special acele domenii în care se

manevrează tipuri de date complexe), în care modelul relațional s-a dovedit a fi insuficient de expresiv și cu performanțe de execuție reduse. Domenii ca: proiectarea asistată de calculator, sisteme de informații geografice, medicină (și altele) au impulsivat cercetări pentru găsirea unor modele mai performante, dintre care modelul orientat obiect și modelul relațional obiectual au cunoscut și cunosc în continuare o dezvoltare semnificativă.

- necesitatea introducerii de funcționalități noi: obiecte multimedia, versiuni de obiecte și tranzacții evolute.
- posibilitatea de partajare a codului reutilizabil al aplicațiilor.

Sisteme de gestiune bazelor de date orientate obiect

Sistemele de gestiune a bazelor de date orientate obiect (SGBDOO) sunt rezultatul unirii între sistemele de gestiune a bazelor de date tradiționale și limbajele de programare orientate pe obiecte.

Apariția SGBDOO este o consecință a proliferării limbajelor de programare orientate obiect, acestea vin și adaugă persistentă limbajelor nativ orientate obiect. Principalul atu îl constituie flexibilitatea sporită în prelucrarea tipurilor de date complexe, rezolvându-se astfel o problemă la care erau deficitare sistemele de gestiune a bazelor de date relaționale. În acest cadru, programarea orientată obiect este nu numai o tehnică de programare, ci constituie în mod esențial o tehnică de structurare a programelor care se sprijină pe entități manipulate de sistem și nu de funcțiile sale.

Un sistem de gestiune a bazelor de date orientat obiect, este un sistem de gestiune a bazelor de date capabil să creeze și să administreze obiecte într-un mod transparent și persistent.

Conform „*The Object-Oriented Database System Manifesto*” [At95], un sistem de gestiune al bazelor de date pentru a fi considerat orientat obiect trebuie să satisfacă două criterii:

1. să fie un sistem de gestiune a bazelor de date;
2. sistemul să fie orientat obiect.

În SGBDOO, **limbajele de programare utilizate tratează datele după modelul obiect**, ca urmare, sistemele de gestiune a bazelor de date]au trebuit să extindă și să completeze cu proprietăți ale bazelor de date, structurile de date modelate după modelul obiect. Acestea sunt: persistenta transparentă, controlul concurenței, refacerea datelor, etc.

Figura 7, ilustrează modul în care cele două componente cu caracteristicile lor, contribuie și asigură performanțele noii tehnologii a bazelor de date orientate pe obiecte.

Primele sisteme de gestiune a bazelor de date orientate obiect au apărut la sfârșitul anilor '80 [Martin93], au la bază modelul obiect și suportă conceptele analizei orientate obiect (*Object Oriented Analyses*)[Bo97], proiectării orientate obiect (*Object Oriented Design*) [Co00] și programării orientate obiect (*Object Oriented Languages*)[Bruce22].

Utilizând aceleași modele conceptuale cu cele din programarea orientată obiect, sistemele de gestiune a bazelor de date orientate obiect simplifică dezvoltarea acestora,

valorifică comunicarea între utilizatori, analiști și programatori și micșorează probabilitatea apariției erorilor[Ba96].

Dintre avantajele cele mai importante ale sistemelor de baze de date dezvoltate în modelul obiect se evidențiază capacitatea acestora de a defini și manevra tipuri de date complexe (clase), care se pot extinde prin mecanismul de moștenire, ceea ce contribuie la creșterea performanțelor în aplicațiile de baze de date avansate.

Există și dezavantaje ale sistemelor de baze de date orientate obiect, acestea le fac să aibe o utilizare limitată, mult mai redusă decât cea a sistemelor de baze de date relaționale. Principalul dezavantaj este datorat greutății în stabilirea structurii obiectelor astfel încât să fie prevăzute toate asocierile posibile necesare efectuării ulterioare a interogărilor. Interogări complexe solicită multe asocieri între obiecte, complicând structura acestora.

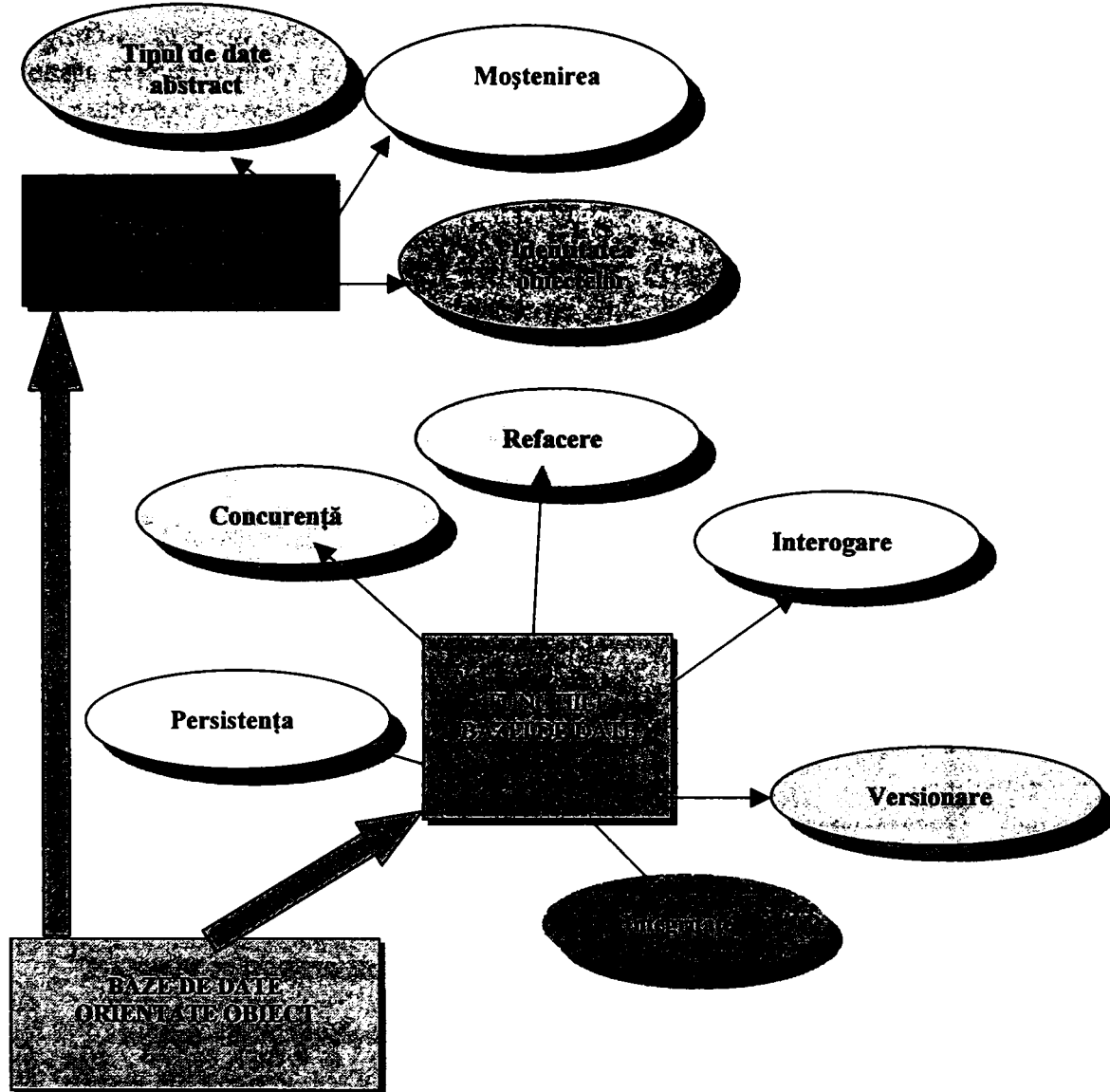


Figura 7. Sisteme de gestiune a bazelor de date orientate obiect.

3.5. Modelul de date relațional-obiectual

Modelul de date relațional-obiectual reprezintă extinderea modelului relațional cu caracteristici ale modelului obiect, extindere necesară pentru realizarea bazelor de date care definesc și prelucrează tipuri de date complexe [St98, St01].

Modelul de date relațional-obiectual păstrează structurarea datelor în relații (reprezentate ca tabele), dar adaugă posibilitatea definirii unor tipuri noi de date pentru domeniile de valori ale atributelor. Tipurile de date definite de utilizator pot fi extinse prin mecanismul de moștenire și pentru fiecare tip sau subtip, de asemenea se pot defini metode pe care le pot executa obiectele de acel tip.

Extinderea sistemelor de baze de date relaționale pentru a deveni relațional obiectuale reprezintă o tendință firească în contextul dezvoltării de aplicațiilor în modul de programare orientat obiect [Da97, Re97]. Experiența și rezultatele obținute cu sistemele relaționale este dezvoltată în contextul utilizării modelului orientat obiect al datelor prin aplicații complexe, programate orientat obiect.

În „matricea lui Stonebraker” [St98, St01], ilustrată în figura 8, sistemele de gestiune a bazelor de date relațional obiectuale se regăsesc în cadranul din dreapta sus, care permite prelucrarea datelor complexe și rezolvarea interogărilor complexe. Modelul relațional obiectual este, evident, cel mai complet, deoarece admite atât tipuri de date definite de utilizator cât și interogări complexe. Stonebraker [St98] denumește sistemele de gestiune a bazelor de date relațional obiectuale ca fiind *sisteme de baze de date universale*.

Complexitatea interogărilor	Cu interogări	Sisteme de gestiune a bazelor de date RELATIONALE	Sisteme de gestiune a bazelor de date RELAȚIONAL OBIECTUALE
	Fără interogări	Sistem clasic de fișiere	Sisteme de gestiune a bazelor de date ORIENTATE OBIECT
		Date simple	Date complexe
			<i>Date complexe</i>

Figura 8 . Clasificarea sistemelor de gestiune a bazelor de date.

Standardele limbajelor de programare din domeniul sistemelor de gestiune relațional-obiectuale sunt extensii ale standardului SQL. Versiunea din anul 1999, denumită SQL3, conține extensii de orientare spre obiecte a limbajului SQL

Ultimele versiuni ale sistemului Oracle (Oracle 8i, Oracle 9i și Oracle 10g) sunt sisteme de gestiune relațional-obiectuale distribuite.

Baza de date Caché [Un05], considerată bază de date performantă „post – relațională” face parte din categoria bazelor de date relațional obiectuale.

Modelul de stocare al datelor în baza de date Caché numit "*hyper-cube*" sau "*spațiu n-dimensional*" este o colecție de șiruri multi – dimensionale numite "*globals*". Informațiile sunt stocate în aceste "*globals*" sub forma de "*subscripts*" care pot fi de forma: șir, întreg, virgula flotanta etc.

Acest mod de stocare al datelor, utilizând un model multidimensional, împreună cu tehnologia orientată obiect, utilizată în aplicațiile ce accesează această bază de date, permit accesul rapid la date (citire, actualizare, ștergere) cu un consum redus de resurse și cu o viteză de lucru sporită a aplicațiilor.

3.6. Persistența datelor

Termenul de obiecte durabile, sau persistente, în conjunctura bazelor de date, este utilizat în problematica memorării acestora într-o memorie externă.

Datele persistente sunt acele date care, se recuperează după terminarea tranzacției sau în caz de eșec, spre deosebire de cele tranziente, care sunt valide doar pe durata rulării programelor sau tranzacțiilor și se pierd atunci când programul sau tranzacția încetează.

În aplicațiile orientate obiect, persistența permite unui obiect să supraviețuiască procesului în care a fost creat. Starea obiectului trebuie memorată în memoria externă și un obiect cu aceeași stare să poată fi recreat pentru condițiile momentului respectiv. O aplicație însă, nu este limitată la numai la un singur obiect, întregul graf al obiectelor trebuie făcut persistent ca apoi, în procesele ulterioare să poată fi recreat.

Persistența este o caracteristică evidentă din punctul de vedere al bazelor de date, dar reprezintă o noutate din punctul de vedere al limbajelor de programare [Ha91].

Din punct de vedere al programării orientate obiect, persistența obiectelor nu este întotdeauna necesară, dar din punct de vedere al bazelor de date, persistența reprezintă o caracteristică obligatorie și este asigurată prin memorarea lor în memoria externă.

Ținând cont de modelele de date utilizate pentru baza de date (relațional, relațional-obiectual și orientat obiect) există trei soluții (tehnici) [Sr96] pentru implementarea persistenței obiectelor:

GOP (*Gateway-based object persistence*) reprezintă tehnica utilizată atunci când într-o aplicație se dorește folosirea unui limbaj de programare orientat obiect pentru date memorate și descrise după o schemă „*non object oriented*”, în această categorie intră și modelul relațional al bazelor de date. Modelul obiectelor aplicației diferă de modelul datelor memorate (persistente), iar această tehnică execută o „mapare” a datelor persistente „*non object oriented*” în date modelate orientat obiect. La timpul rulării, această tehnică asigură translatarea obiectelor din reprezentarea utilizată în aplicație în cea în care ele există sub forma persistentă și invers. Acest mod de implementare asigură o independență a aplicației de date și este cunoscută sub numele de abordarea de mijloc ("*middleware*" approach).

Sisteme de gestiune a bazelor de date relațional-obiectuale au fost construite pentru a răspunde provocării înaintate de limbajele de programare orientate obiect. Sisteme de gestiune a bazelor de date relaționale au cunoscut o dezvoltare de succes datorită rigurozității matematice și utilizării limbajului SQL ca limbaj de interogare. Ele adaugă suport pentru modelarea orientată obiect a datelor și extinde această modelare modelului relațional și limbajului de interogare respectând tehnologia relațională. Această abordare este cunoscută sub numele de „jos în sus” (*bottom-up approach*), punând în centru datele sau baza de date.

Sisteme de gestiune a bazelor de date orientate obiect spre deosebire de cele relațional-obiectuale, au fost concepute astfel încât să fie capabile să lucreze cu obiecte persistente utilizând limbaje de programare orientate obiect. Adesea, datorită faptului că acestea au originile în limbajele de programare orientate obiect, sistemele de gestiune a bazelor de date orientate obiect sunt referite ca sisteme de limbaje persistente. Capabilitățile SGBDOO depășesc sfera lucrului cu obiecte persistente, ele prezintă toate caracteristicile cerute în „*The Object-Oriented Database System Manifesto*”, modelul

orientat obiect al datelor din aplicație corespunde cu cel din baza de date. Implementarea persistenței obiectelor prin SGBDOO este cunoscută sub numele de abordarea de „sus în jos” (*top-down approach*), punând în centru aplicația (sau limbajul de programare)

Primele două dintre sistemele propuse pentru implementarea persistenței au la bază reprezentarea datelor „*non object oriented*”, reprezentant de seamă fiind modelul relațional și bazele de date relaționale. Tehnologia relațională s-a impus categoric, iar multitudinea bazelor de date relaționale precum și a sistemelor de gestiune a bazelor de date relaționale nu este deloc de neglijat. Limbajul utilizat de sistemele de gestiune a bazelor de date relaționale este SQL, de aceea ele mai sunt cunoscute și referite ca **Baze de date SQL** sau **Sisteme de gestiune a bazelor de date SQL**.

Figura 9 prezintă posibilitățile de memorare a obiectelor în memoria externă.

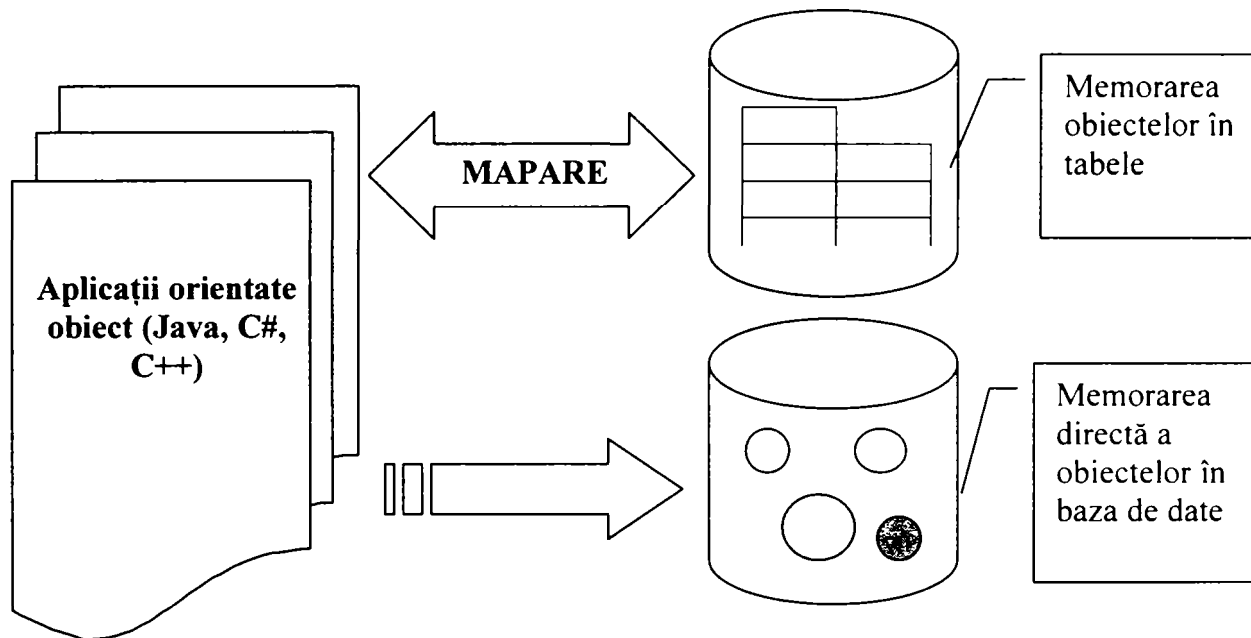


Figura 9. Memorarea obiectelor în baze de date

Pentru baza de date, s-a ales modelul de date relațional, iar sistemul de gestiune a bazei de date este Microsoft SQL Server 2000. Interfața cu baza de date (aplicația *Ameliorare*) a fost dezvoltată într-un limbaj de programare orientat obiect (C#.NET)[Se03, C#04, Wi02].

3.7. Caracteristici și cerințe ale aplicațiilor orientate obiect ce lucrează cu baze de date relaționale

Materializarea persistenței datelor precum și modul de abordare de către aplicațiile orientate obiect a bazelor de date a căror model al datelor este altul decât cel obiect, prezintă unele particularități ce trebuie analizate din trei puncte de vedere:

1. modelarea datelor (memorare, organizare și refacerea structurilor de date);
2. accesarea datelor;
3. accesul concurrent.

3.7.1. Modelarea datelor

Limbajele de programare orientate obiect se bazează și folosesc caracteristicile tipice modelului obiect și anume: obiecte complexe, identitatea obiectelor, încapsularea, moștenirea, legăturile dinamice, etc. Nu toate aceste trăsături, native pentru limbajele de programare orientate obiect, sunt suportate de sistemele de gestiune ale bazelor de date, doar sistemele de gestiune a bazelor de date orientate obiect le suportă în totalitate, deoarece în această idee au fost concepute. Sistemele de gestiune bazelor de date relațional-obiectuale au implementate și suportă doar unele dintre aceste trăsături, sistemele de gestiune a bazelor de date relaționale nu suportă aceste trăsături datorită faptului că folosesc alt model al datelor, cel relațional, total diferit de cel orientat obiect.

Diferența între modelul orientat obiect al datelor și modelele „*non object oriented*” generează o problemă majoră cunoscută sub numele „*impedance mismatch*”. Ea este generată de diferențe fundamentale ce există între cele două modele a datelor, acest termen, este utilizat pentru a exprima incompatibilitatea între conceptele de bază ale celor două sisteme [St98, Da97, Sh02]

Impedance-mismatch este atunci când o bază de date relațională este accesată prin intermediul unei aplicații scrise într-un limbaj de programare orientat obiect.

Diferențe între cele două tipuri de modele a datelor generează următoarele probleme:

1. Există două limbaje de programare distincte, cel în care se scrie aplicația și limbajul de manipulare al datelor care lucrează cu baza de date. Logica aplicației este implementată în limbajul de programare orientat obiect iar crearea, manipularea datelor (CRUD *create, read, update și delete*) și interogarea este realizat cu SQL. Paradigma „*mismatch*” se reduce de fapt între limbajele de programare orientate obiect și SQL
2. Prelucrarea datelor din baza de date în programul de aplicație presupune translatarea lor din modelul utilizat în baza de date (modelul relațional, „*non object oriented*”) în modelul orientat obiect, utilizat de programul de aplicație, la fel, este necesară translatarea datelor din modelul obiect în tabelele bazei de date în urma actualizării acestora. Există diferențe între reprezentarea din memorie internă a datelor și cea persistentă, diferențe ce generează „*impedance mismatch*”, această incompatibilitate este eliminată

atunci când cele două modele ale datelor din aplicație și din baza de date sunt identice.

La aplicațiile ce utilizează sistemele de gestiune a bazelor de date orientate obiect nu se întâlnește „*impedance mismatch*” datorită faptului că modelul datelor utilizat în aplicație este identic cu cel din baza de date.

Sistemele de gestiune a bazelor de date orientat obiect, sunt capabile să creeze și să administreze obiecte într-un mod transparent și persistent. La SGBDOO problemele care apar sunt legate în mod special de modul în care are loc interogarea bazei de date organizată ca bază de obiecte.

Sistemele de gestiune a bazelor de date relațional-obiectuale, în vederea diminuării „*impedance mismatch*”, asigură tot mai multe dintre caracteristicile tipice modelului obiect, amintim aici extinderea tipurilor cu tipul de date UDT (*User Defined Types*) și extensiile orientate obiect prevăzute de standardul SQL3. Incompatibilitatea între modelele datelor utilizate de aplicație și de baza de date se rezolvă într-o oarecare măsură prin translatarea unor porțiuni din aplicație direct pe serverul bazei de date prin posibilitatea executării procedurilor stocate.

Sistemelor a căror model al datelor este altul decât cel orientat obiect („*non object oriented*”), încearcă să reducă „*impedance mismatch*” utilizând unelte software ce **mapează** obiectele în tabele și tabelele în obiecte, creând impresia că se lucrează cu un singur model al datelor, și anume modelul utilizat în aplicație (modelul orientat obiect).

3.7.2. Implementarea trăsăturilor specifice modelului orientat obiect

Modelul relațional și în general modelele „*non-object-oriented*”, nu au căi de reprezentare a caracteristicilor obiectelor în modul în care o face modelul obiect. Implementarea caracteristicilor tipice programării orientate obiect, în cazul utilizării acestor modele de reprezentare a datelor, se face prin „*maparea*” obiectelor în baza de date [Am03, Ke97, Fu97]. Figura 9 prezintă schematic modul de implementare a obiectelor persistente în baze de date orientate obiect și baze de date relaționale.

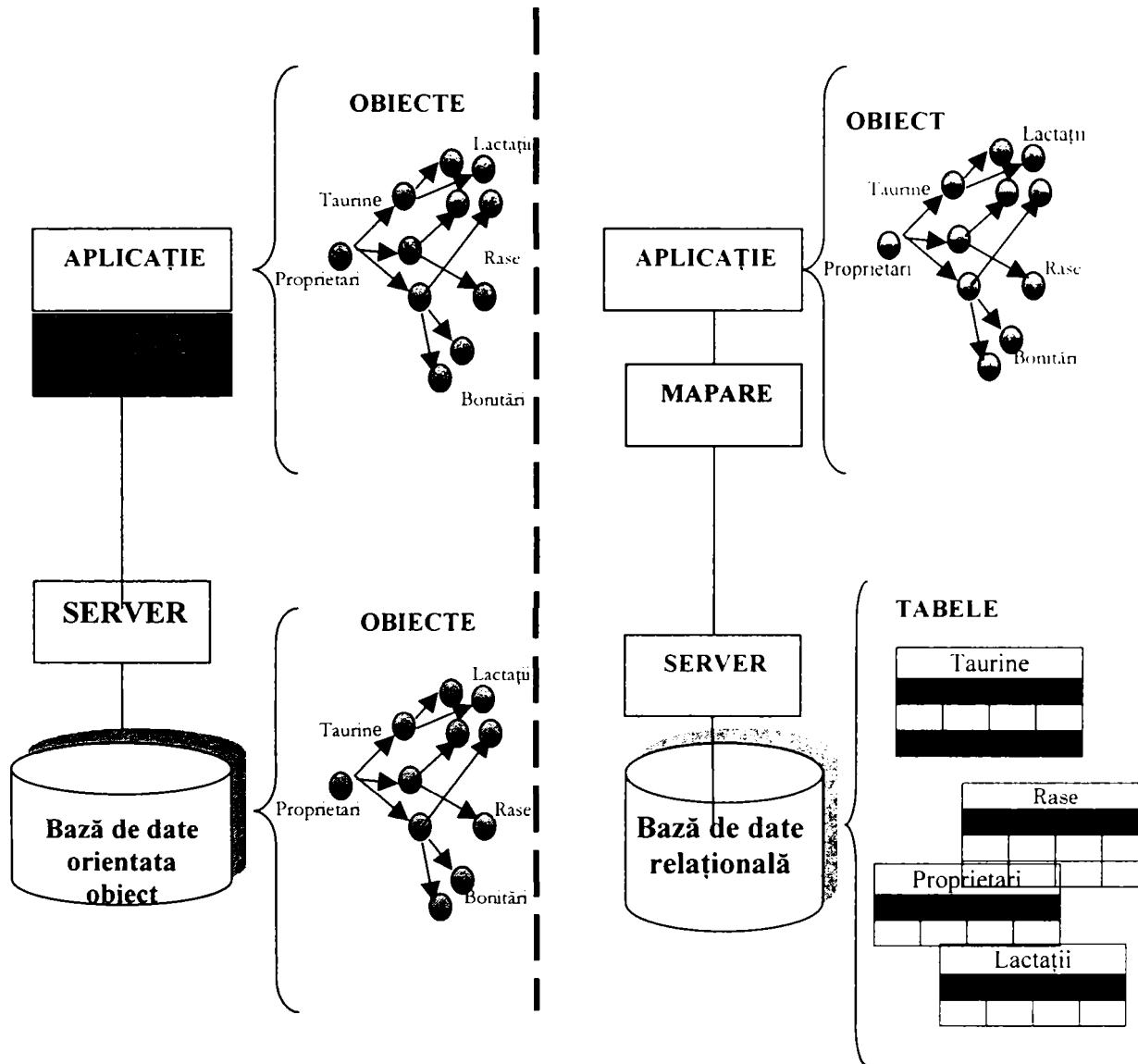


Figura 9. Implementare obiectelor persistente în sistemele bazate pe modelul orientat obiect și modelul „*non object oriented*”

Termenul utilizat este *object/relational mapping (ORM)* sau simplu *object mapping*, în esență înseamnă transformarea datelor dintr-o reprezentare (obiect) în alta (relațional).

Maparea obiectual/relațională este considerată o tehnică în programare, ea asigură legătura între bazele de date relaționale și conceptele limbajelor de programare orientate obiect, creând (efectiv) o bază de date obiecte virtuală.

Maparea poate fi realizată prin scrierea de cod sau prin utilizarea uneltelor construite special în acest scop[Ge04, Mo02], având ca obiectiv automatizarea și creșterea performanței în programare, unelte ce realizează maparea în mod automat (și transparent) a obiectelor în tabelele bazei de date relaționale. În figura este prezentat schematic modul de implementare a obiectelor persistente în sistemele bazate pe modelul orientat obiect și modeleul „*non object oriented*”.

Componentele de bază ale mapării obiectelor în baza de date relațională sunt:

1. Maparea atributelor în coloane;
2. Maparea claselor în tabele;
3. Implementarea moștenirii în bazele de date relaționale;
4. Maparea relațiilor.

Obiective urmărite în evaluarea mapării

Realizarea mapărilor se face cu un efort și un cost. În evaluarea mapărilor sunt urmărite următoarele obiective:

- **Performanța în accesarea memoriei externe.**

Diferența între timpii de acces la hard disk (sau alt mediu de memorare) și al memorie RAM, este de șase ordine de mărime (10^{-9} și 10^{-3}), ca urmare, maparea care accesează des memoria externă scade performanța accesului la date.

- **Performanțe la citire vs. scriere / actualizare.**

Implementarea moștenirii se poate face în mai multe moduri. Funcție de numărul tabelelor în care sunt mapate ierarhia de clase, citirea sau scrierea/actualizarea se poate face printr-un număr diferit de accesări ale bazei de date.

- **Flexibilitatea și costurile întreținerii.**

Flexibilitatea se referă la posibilitățile de modificare (ștergere, adăugare de attribute) ale proprietăților claselor din ierarhie. Aceste modificări se pot face într-un număr diferit de tabele (funcție de implementarea aleasă) cu anumite costuri.

- **Performanță și redundanță vs. cost de întreținere și forma normală.**

Eliminarea redundanței în cazul bazelor de date relaționale se face prin normalizarea bazei de date. Aceasta asigură cele mai bune performanțe cu un cost minim. Alegând diferite moduri de mapare a ierarhiei de clase, nu întotdeauna se respectă regulile de normalizare, ca urmare se reduce redundanța și performanța.

- **Consumul de spațiu vs. Performanță.**

Există mapări care nu fac economie de spațiu de memorie. Funcție de implementare, vor exista tabele în care multe înregistrări vor avea valori <null>. Alegerea metodei de implementare se va face și funcție de acest criteriu.

- **Procesul de interogare.**

Funcție de natura aplicației ce utilizează baza de date (aplicații cu procese online sau simple depozite de date), interogarea acestora se face în mod diferit și ca urmare, proiectarea mapării claselor de obiecte se face ținând cont de normalizarea bazei de date, fără redundanțe și o restructurare care să conferă o performanță optimă.

Maparea atributelor în coloane

Aplicația „Ameliorare” este o aplicație orientată obiect, este organizată ca o colecție de obiecte cooperante, fiecare obiect fiind instanța unei clase. Fiecare clasă reprezintă abstractizarea unui tip de entitate din realitatea domeniului aplicației, obiectele au o structură și un comportament, accesul utilizatorului este permis doar la funcțiile (metodele) pe care obiectul este capabil să le execute.

Corespondența obiect relațională se realizează prin asocierea unei relații (tabelă) cu o clasă, fiecare obiect (instanță a clasei) reprezintă un tuplu din relația respectivă (Figura 10). Clasa conține câte o variabilă membră corespunzătoare fiecărui atribut al relației, iar fiecare obiect are ca valori ale acestor variabile atributele tuplului respectiv din relație. Astfel, Schema conceptuală a bazei de date conține 14 tabele, corespunzătoare entităților specifice aplicației. În BusinessObjects au fost definite clasele corespunzătoare entităților, fiecare clasă are variabile membre corespunzătoare fiecărui atribut al relației, de asemenea sunt prevăzute pentru fiecare atribut metode de citire scriere (*set*, *get*) pe lângă metodele specifice de prelucrare a acestora.

Rasa	Ob1: Rasa
long rasaID	rasaID=1
long grupaID	grupaID=1
string codRasa	codRasa="BR"
string denumireRasa	denumireRasa= "Baltata Romăneasca"
string descriereRasa	descriereRasa="S-a format in nordul
string fotoRasa	fotoRasa=

RasaID	GrupaID	CodRasa	DenumireRasa	DescriereRasa	FotoRasa
1	1	BR	Baltata Romaneasca	S-a format in nord-	<Binary>
12	10	BNR	Baltata cu negru Roma	Tipul morfo-producl	<Binary>
25	6	B	Bruna	Rasa locala, format	<Binary>

Figura 10. Corespondența obiect - relațional între tabela Rasa și clasa Rasa

Tipurile atributelor pot fi:

- de bază
 - definite de utilizator.
- ▶ Dacă tipul atributului este de bază, apare o corespondență între tipul atributului și cel al coloanei din tabelă.
 - ▶ Dacă tipul atributului este unul definit de utilizator, nu mai poate apărea o corespondență unu-la-unu între atribut și coloană, iar

atributele clasei pot mapa în acest caz de la zero la mai multe coloane în baza de date relațională

- ▶ Dacă tipul unui atribut este de tip obiect, mapearea acestui atribut se face printr-o referință la tabela ce mapează clasa obiectului respectiv prin intermediul cheii străine (FK).

În figura 11, este prezentată clasa Taurina (aplicația *Ameliorare*).

În cazul clasei *Taurina*, atributele *Exploatație*, *Rasa*, *Tata*, *Mama*, sunt de tip obiect.

Prin mapare, în tabelul *Taurina* ce mapează clasa *Taurina*, coloanele corespunzătoare acestor câmpuri sunt chei străine (FK), ce referă tabele ce sunt mapate de clasele corespunzătoare obiectelor respective.

În cazul atributelor *Mama* și *Tata*, apare o relație recursivă.

Între atributele unei clase, nu toate sunt persistente. De exemplu, *Q* grăsime este utilizat de instanță pentru a calcula cantitatea de grăsime în lapte. Acest atribut nu este persistent în baza de date.

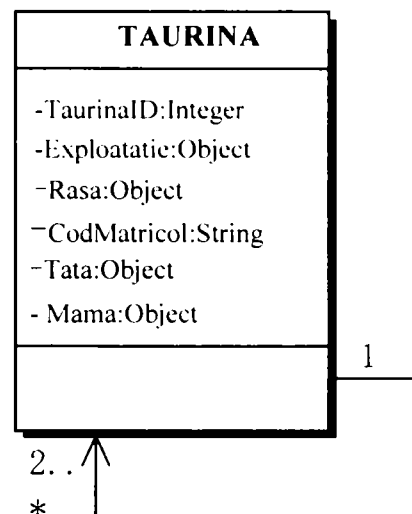


Figura 11. Tipurile atributelor în clasa *Taurina* (UML)

Obiecte complexe

Obiectele complexe conțin un număr arbitrar de câmpuri, fiecare memorând valori atomice ale datelor sau referințe la alte obiecte (de tip arbitrar). Obiecte complexe modelează exact percepția utilizatorului asupra entităților din lumea reală.

Mecanismul de definire al obiectelor complexe permit definirea lor prin atribute ce conțin la rândul lor alte obiecte. Structura obiectelor complexe nu respecta prima forma normală, cerință impusă de structura tabelelor în cazul modelului relațional.

Obiectele complexe sunt construite pornind de la cele simple aplicând asupra lor constructori. Obiecte simple sunt cele de tip întreg, real, boolean, caracter, string, etc., asupra lor se aplică constructori (sets, arrays, lists, bags). Constructorii sunt importanți din punct de vedere al modelării orientate obiect, deoarece captează o trăsătură a obiectului

În cazul obiectelor complexe, operațiile aplicate “rădăcinii” obiectului care considerăm că grupează mai multe obiecte, se propagă asupra tuturor obiectelor care fac parte din grup.

Operațiile ce se pot aplica obiectelor complexe sunt:

- egalitatea
- copierea
- ștergerea
- blocarea (*lock*)

Pentru aceste operații se disting trei nivele: de identificator, de suprafață (*shallow*) și de adâncime (*deep*).

În cazul egalității, compararea la cele trei nivele se face: la nivel de identificator, a valorilor atributelor și respectiv valorilor atributelor obiectelor componente.

Pentru copiere, *shallow copy* creează un obiect nou și copie valorile atributelor, copierea de adâncime creează recursiv un nou obiect.

Propagarea operațiilor de ștergere și blocare înseamnă că dacă obiectul rădăcină este șters sau blocat toate obiectele componente sunt de asemenea șterse sau blocate.

Obiectele care sunt definite în interiorul altor obiecte sunt parte componentă a obiectului în care a fost definit, nu au identitate, sunt automat create sau șterse în momentul creării sau ștergerii obiectului în componenta căruia sunt definite.

Implementarea obiectelor complexe, presupune **maparea** lor în elementele de bază permise de tipul bazei de date. Maparea constă în scrierea de cod ce translatează înspre baza de date și dinspre baza de date, obiecte complexe definite în aplicația scrisă în limbajul orientat obiect. Evident că, scrierea acestui cod și implicit execuția lui îngreunează și scade din performanța aplicației, maparea devenind ineficientă în cazul utilizării bazelor de date ce nu sunt „echipate” cu posibilități de stocare a obiectelor complexe.

Implementarea tipurilor complexe, atunci când acestea pot fi interpretate ca fiind elemente ale unei matrici (*array*) se poate realiza în două moduri:

1. memorarea elementelor matricii în rândurile (tuplurile) tabelului cu o coloană pentru fiecare element al matricii (în cazul în care tipurile elementelor matricii sunt la rândul lor obiecte complexe, vor fi necesare multiple coloane pentru fiecare element al matricii)
2. memorarea elementelor matricii (*array*), separat într-o tabelă, unde fiecare rând (tuplu) al acestei tabeli memorează un singur element al matricii cu indexul acestui element în matrice.

Taurine *			
	Column Name	Data Type	Length
?	TaurinaID	bigint	8
	ExploatareID	bigint	8
	RasaID	bigint	8
	TataID	bigint	8
	MamaID	bigint	8
	Codmatricol	varchar	50
	Nume	varchar	50
	Sex	varchar	1
	DataNasterii	datetime	8
	LoculNasterii	varchar	50
	ConditiiDeIntrare	varchar	50
	DataIntrarii	datetime	8
	ConditiiDeIesire	varchar	50
	DataIesirii	datetime	8
	Ameliorare	varchar	50
	DateSanatate	varchar	1024
	Fototaurina	image	16
	DataPrimeiFatari	datetime	8
	DataUltimeiFatari	datetime	8
	ValoareaDeAmeliorare	float	8
	GreutateNastere	float	8
	Greutate6Luni	float	8
	Greutate12Luni	float	8

În aplicația „Ameliorare”, la implementarea obiectului TAURINA, attributele tata și Mama sunt definite recursiv, implementarea s-a făcut prin intermediul cheii străine (tataID și mamaID) care referă tabela Taurina. Pentru attributele Rasa și Exploatare, implementarea s-a realizat prin referirea tabelului Rasa și Exploatare prin intermediul cheii străine rasaID și exploatare ID. Evident că pentru orice tuplu, valoarea atributului mamaID și tataID trebuie să se regăsească printre valorile cheii primare taurinaID, sau poate fi NULL, dacă pentru taurina respectivă nu se cunosc ascendenții. Situație similară apare și în cazul celorlalte două attribute, Rasa și Exploatare, valorile atributelor rasaID și exploatare ID trebuie să se regăsească printre valorile cheii primare a celor două tabele referite, valori NULL se acceptă pentru taurinele a

căror rasa este incertă. Pentru Exploatație, evidența strictă a localizării taurinei implică existența obligatorie a unei exploatații (ferma) și a unui proprietar.

Relații de asociere între tabele

Asocierile (*relationship*) între tipurile de entități definite în modelul conceptual al unei baze de date se realizează în modelul relațional prin intermediul cheilor străine.

Legătura dintre tabelele bazei de date se realizează prin mecanismul de propagare a cheilor. Tabela care conține cheia străină se numește tabelă (relație) care referă, iar tabela (relația) care conține cheia candidat se numește tabelă (relație) referită. Cheia străină realizează asocierea N:1 între tabele (relații), ea trebuie să respecte anumite condiții pentru a putea asigura această legătură, condiția se numește condiție de integritate și reprezintă o constrângere între cele două relații. Condiția de integritate referențială impune ca mulțimea valorilor unei chei străine să fie inclusă în mulțimea valorilor cheii primare din care s-a propagat.

Operațiile de modificare a stării unei relații (introducere, ștergere, actualizare a tuplurilor relației) pot afecta integritatea referențială a bazei de date, dacă modificările dintr-o relație se fac individual, fără a se ține cont de referințele către și de la această relație.

În modelul relațional asocierea între relații (tabele) se realizează prin egalitatea valorilor atributelor cheii străine cu ale cheii referite, fără să fie necesară memorarea unor legături explicite.

În figura 12, este ilustrată dependențele de ordinul I ale tablei Taurine. Tabela Taurine referă tabelele Bonitari, Lactații și Reproducție, și la rândul său este referită de tabelele Exploatații și Rase.

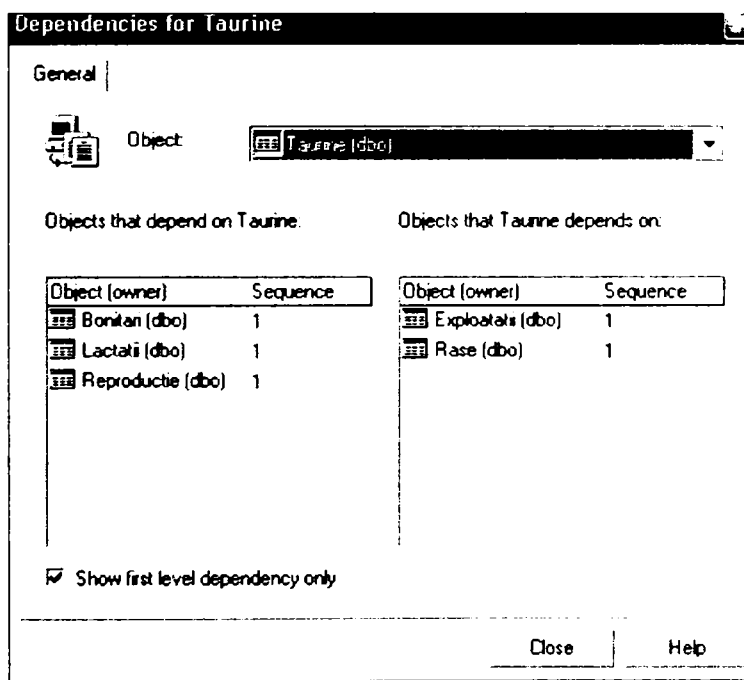


Figura 12. Dependențele tablei Taurine

În figurile 13 și 14, sunt reprezentate tabelele care depind de tabela Taurină, respectiv tabelele de care depinde tabela Taurină, cu cheile străine aferente. Întâmplător, ele au același nume cu cheia primară a tabelului din care este referită, dar acest lucru nu este obligatoriu.

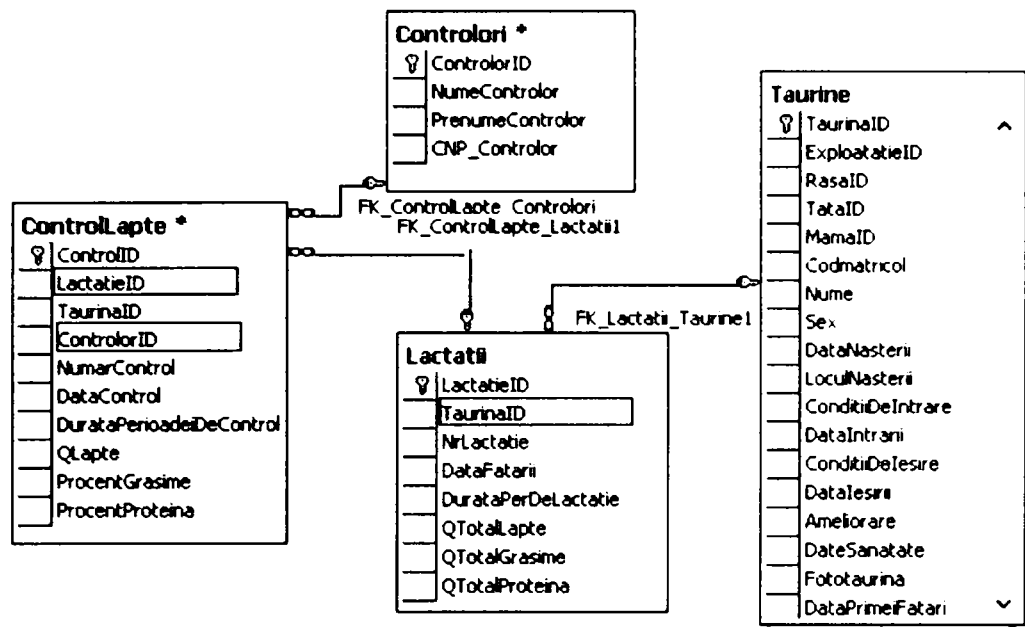


Figura 13. Tabelele referite de Tabela taurină cu cheile străine aferente

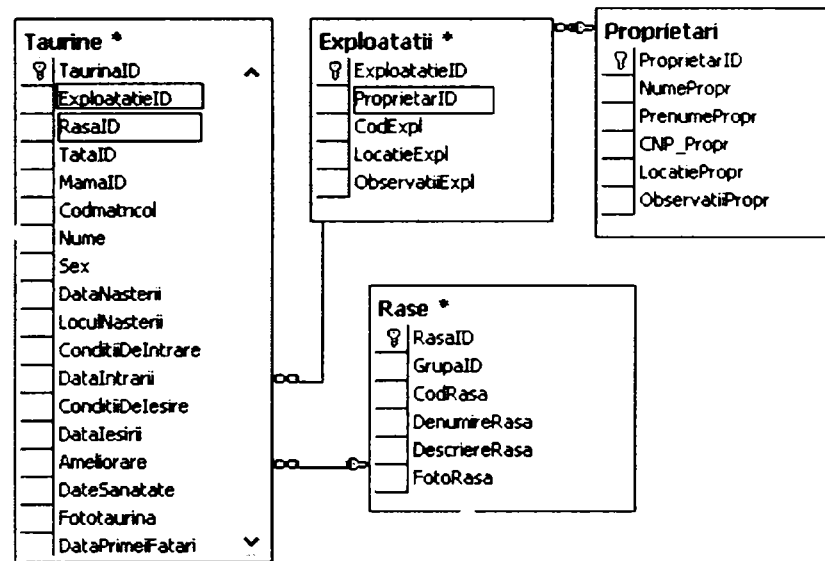


Figura 14. Tabelele care referă tabela Taurină, cu cheile străine aferente

Maparea relațiilor de asociere

În modelarea orientată obiect, asocierea reprezintă relația dintre entități. Maparea asocierii și administrarea entităților asociate, reprezintă conceptul central al oricărei soluții pentru obiecte persistente

Relația de **asociere** se stabilește între obiecte, în sensul că un obiect utilizează alt obiect. Tipurile de relații ce pot exista sunt:

- unu-la-unu (one-to-one)
- unu-la-mulți (one-to-many)
- mai multi-la-mai multi (many-to-many)

Obiecte sunt în relație de **agregare** dacă obiectul întreg se consideră că este format din obiecte componente.

În limbajele de programare orientate obiect, reprezentarea asocierii se face utilizând referințe la obiect (*object references*) și colecții de referințe la obiect.

În sistemele de gestiune a bazelor de date ce utilizează modelul relațional, reprezentarea asocierii se face utilizând cheia străină (*FK*). O coloană în tabelă este reprezentată de cheia străină, care de fapt este o copie a cheii primare a tabelului cu care acesta se află în relație de asociere.

Referințele la obiect sunt direcționate în sensul că asocierea este de la un obiect la altul sau în ambele direcții, arătând modul în care are loc navigarea.

Cheia străină în tabelele relaționale utilizată pentru a materializa relația existentă între tabele nu poate arăta direcția ca urmare navigarea nu poate avea același sens ca în cazul obiectelor referite.

- (1) **Maparea relației de asociere 1:1** Relația de asociere 1:1 presupune că există o corespondență unu la unu între două entități. Dacă cele două entități sunt mapate în două tabele, relația 1 la 1 presupune că fiecărui rând din primul tabel îi corespunde un rând din al doilea tabel.
- (2) **Maparea relației de asociere 1:N** se realizează prin intermediul cheii străine(FK) (Figura 15).

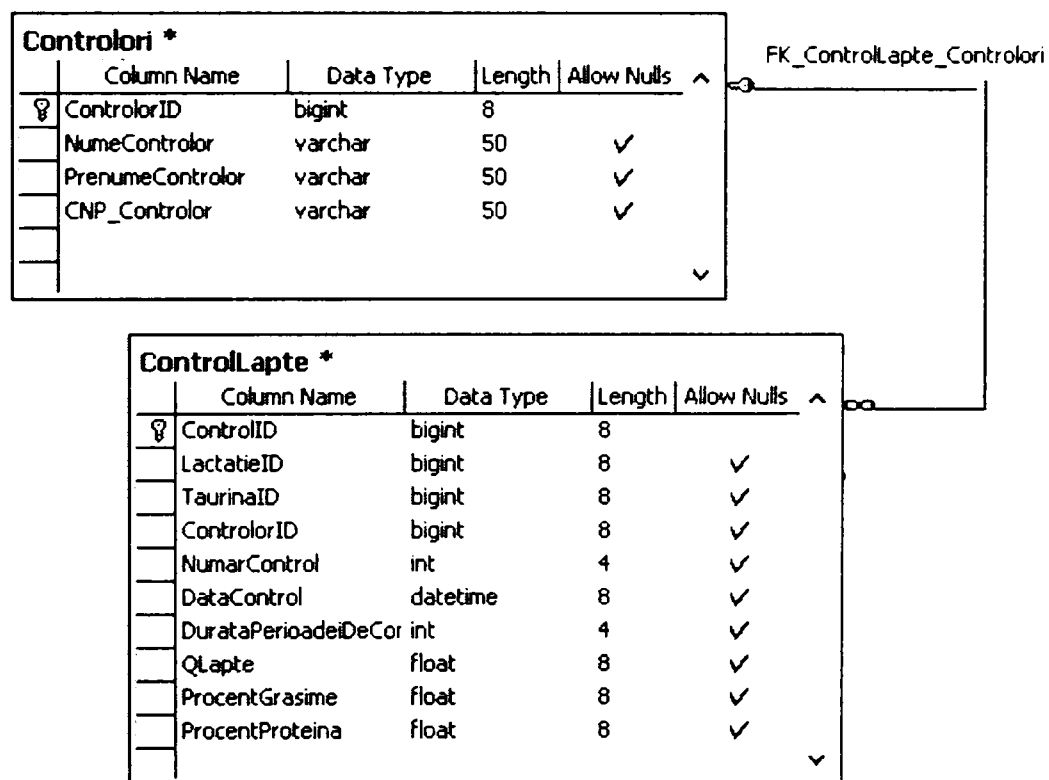


Figura 15. Tabelele Controlori și Control lapte se află în relație de asociere (1:n)

Consecințe:

Performanțe la citire: citirea obiectului Control Lapte presupune o interogare de tip JOIN pentru a afla identitatea controlorului, sau două operații de citire.

Performanțe la scriere: modificarea datelor de identitate ale controlorilor nu presupune decât modificarea acestora în tabela *Controlori*.

Memoria consumată este optimă, se folosește în plus doar coloana ce conține cheia străină, iar din punct de vedere al respectării normalizării nu apar conflicte (Figura 15).

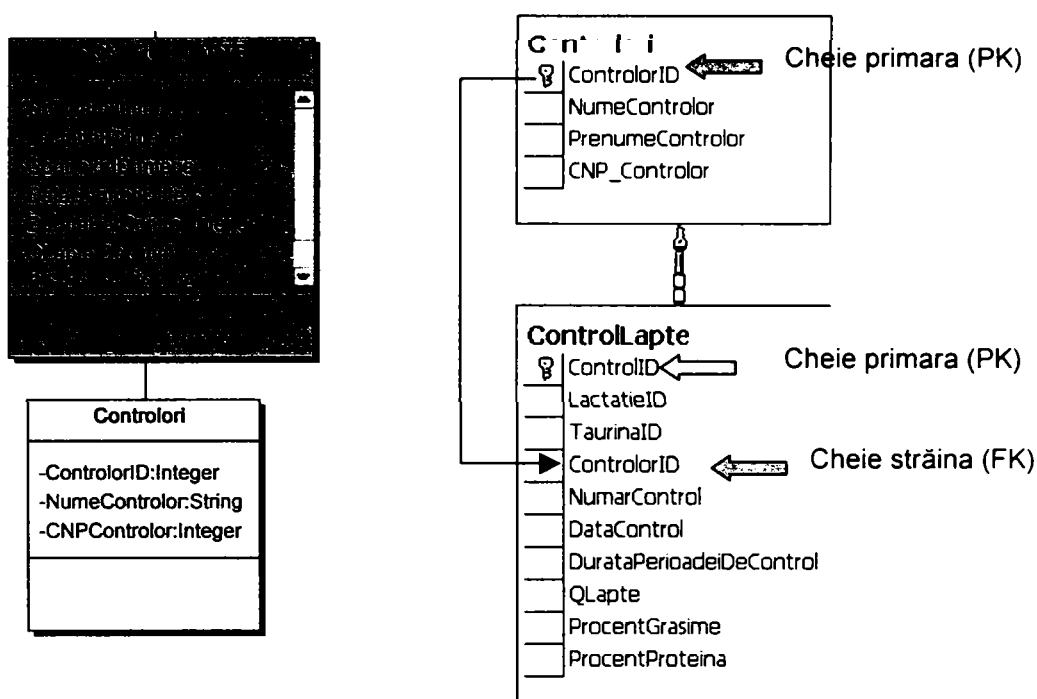


Figura 15. Implementarea relației de asociere 1:n.

Obiectele compozite (agregat)

Obiectele compozite (agregat) sunt acele obiecte individuale care pot fi considerate ca făcând parte dintr-un grup, în sensul de a compune acel grup, de a forma o colecție.

Pentru a ilustra esența obiectelor agregat prezentăm în figura 16, o posibilă interpretare a structurii compozite a obiectului Lactație (aplicația Ameliorare).

Pentru o taurină (Sex: feminin), materializarea unei lactații este concretizată prin maxim 13 controale în care sunt măsurate caracteristicile productive (lapte, % grăsime, % proteină). Lactația nu există dacă nu există cel puțin un Control lapte, ca urmare, obiectul Lactație constă dintr-o colecție de obiecte Control lapte, obiecte de sine stătătoare, care pot fi referite ca atare în aplicație.

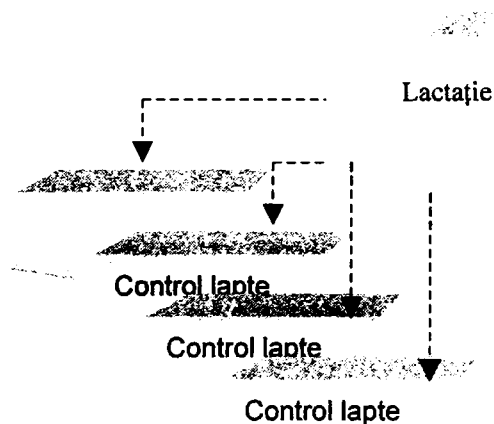


Figura 16. Obiect compozit (agregat)

Asociat cu acest tip de obiect (compozit) apare noțiunea de ștergere și copiere în cascadă. În aplicație, dacă o Lactație este ștearsă, se vor șterge automat toate obiectele compozite (Controalele lapte aferente), de asemenea copierea unei Lactații presupune copierea în adâncime (*deep*) a tuturor Controalelor lapte ce o compun.

În aplicația „Ameliorare” , obiecte compozite pot fi considerate obiectele „Control lapte” și „Caractere exterior”, aflate în relație de asociere N:1 cu obiectele „Lactație”, respectiv „Bonitare”. Constrângerile referențiale impuse asigură ștergerea și actualizarea în cascadă a tuturor tuplurilor corespunzătoare unei Lactații, respectiv unei Bonitări la ștergerea , respectiv actualizarea elementelor referite.

Maparea relației de agregare

Există două modele de implementare.

(1) Maparea atributelor obiectelor agregate într-o singură tabelă (Figura 17).

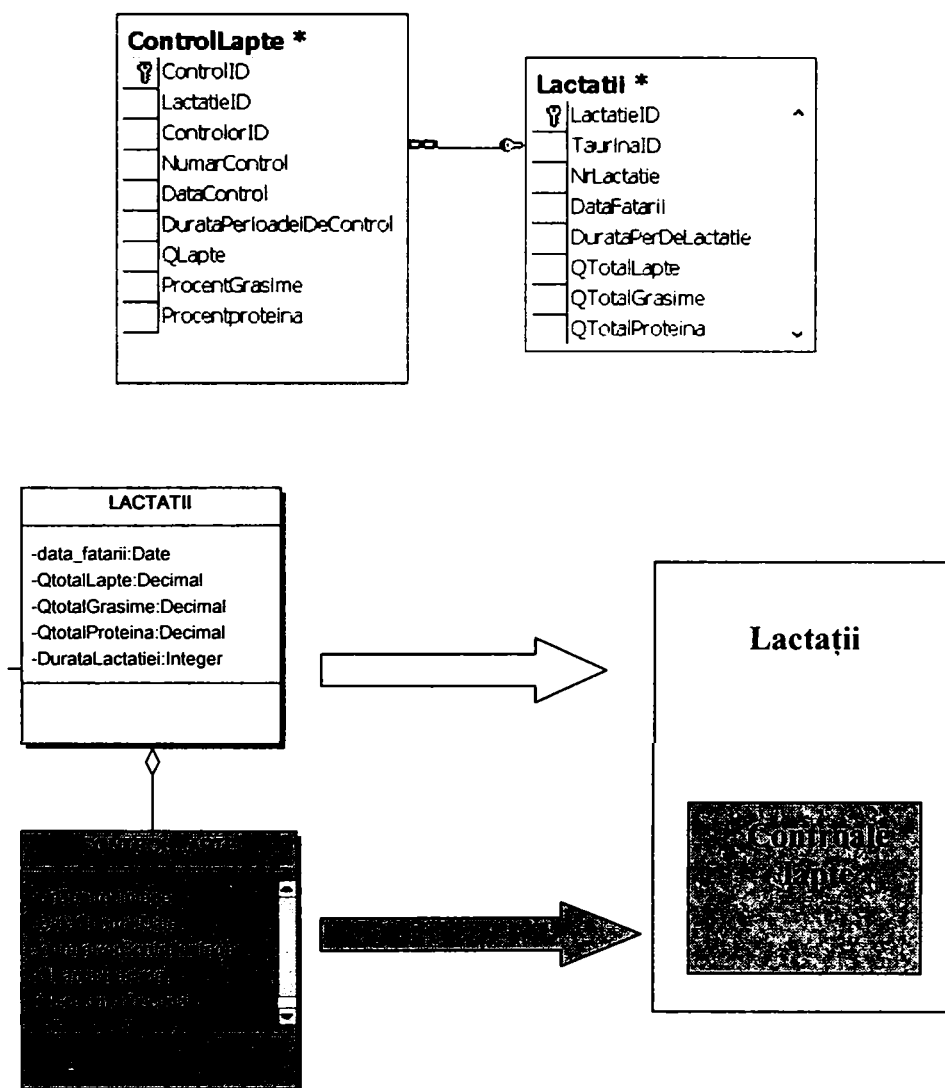


Figura 17. Maparea atributelor obiectelor agregate într-o singură tabelă

Conform acestui model de implementare, maparea se realizează printr-o tabelă ce conține atributele clasei **Lactații** plus atributele clasei **Controale Lapte**.

(2) Utilizarea cheii străine pentru implementarea relației de agregare (Figura 18.).

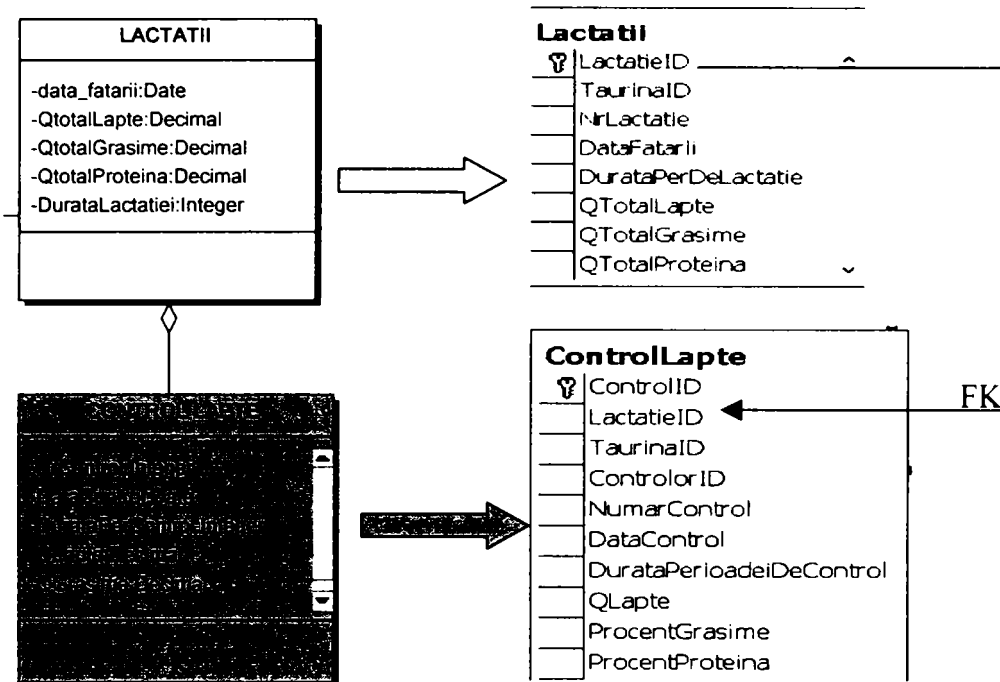


Figura 18. Utilizarea cheii străine pentru implementarea relației de agregare

Implementarea cu ajutorul cheii străine (FK) este mai avantajoasă, acest mod de implementare a fost folosit în aplicația „Ameliorare”.

În ceea ce privește consistența bazei de date, constrângerile impuse trebuie să asigure ștergerea tuturor **Controalelor lapte** pentru o **Lactație**, în cazul în care aceasta se șterge.

Maparea relației de asociere N:M

O relație de asociere N:M poate fi interpretată ca o relație bidirecțională 1 : N. Pentru a crea acest tip de relație este necesar să definim o colecție de atribute referite de fiecare clasă implicată în relație. În modelul relațional, o relație N:M se definește cu ajutorul cheii străine sau utilizând un tabel de legătură (*link table*).

În cazul mapării, tabelul de legătură (*link table*), poate sau nu să fie reprezentat de o clasă. Funcție de situație, tabelul de legătură este **tranzacțional** sau **transparent**.

Tabel de legătură tranzacțional.

Tabelul tranzacțional, este tabelul de legătură conține în afara cheilor străine către cele două tabele implicate în relație și alte informații (persistente). Pentru acest tabel va fi creată o clasă în modelul obiect.

Tabel de legătură transparent.

Dacă tabelul de legătură conține numai cheile străine ce fac legătura cu tabelele ce sunt mapate de clasele aflate în relație, atunci nu este necesară crearea unei clase aferente acestui tabel. Tabelul de legătură este considerat **transparent**.(Figura 19)

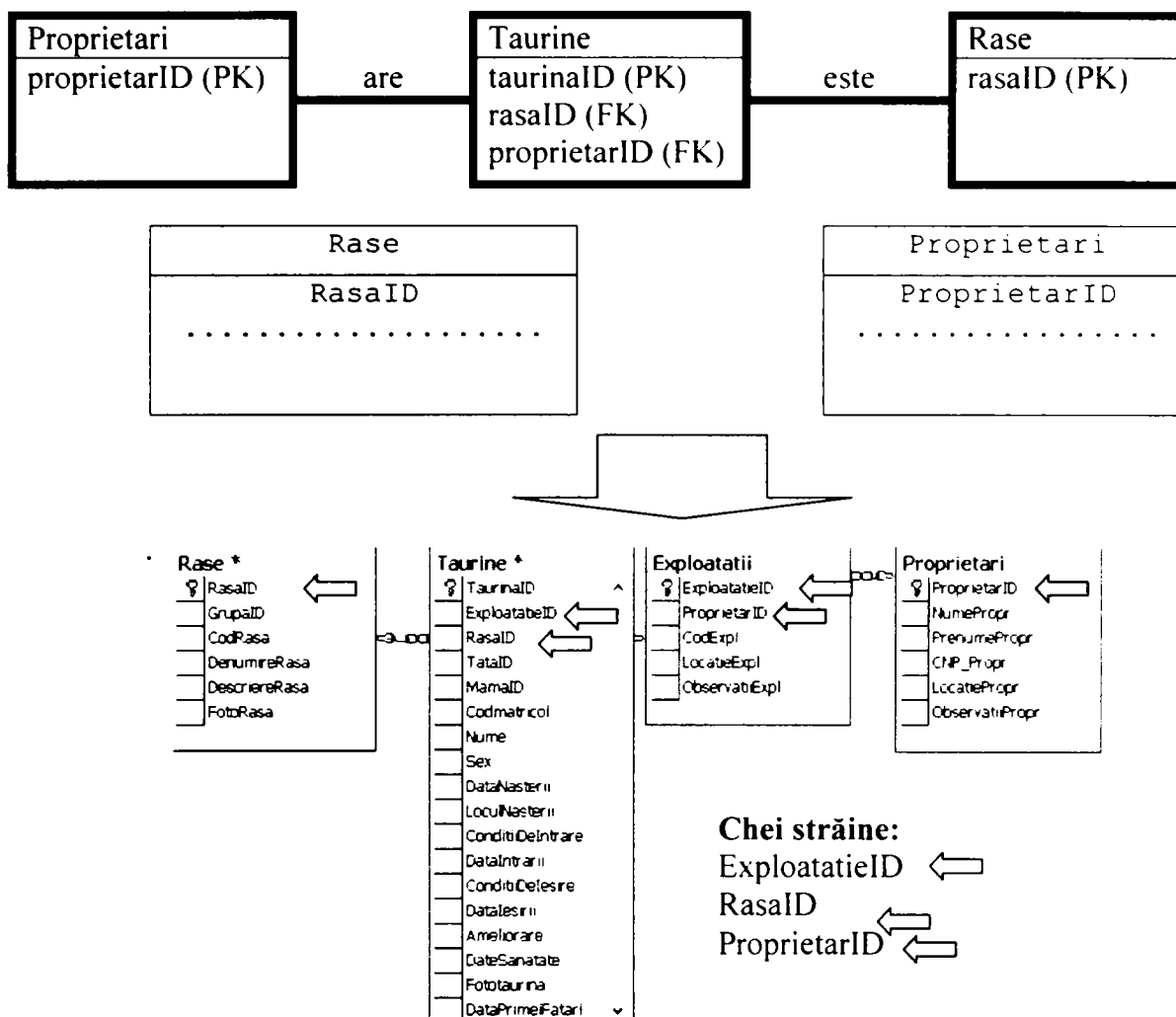


Figura 19. Relația de asociere N:M. Tabel de legătură: tabela Taurine

Maparea moștenirii

Diagrama (UML) ilustrată în figura 20, se prezintă arborele unei moșteniri simple.

Clasa *Taurine* moștenește clasele *Taur* și *Vaca*. Clasa *Taurine* este o clasă abstractă, iar clasele *Taur* și *Vacă* sunt clase concrete. Există trei soluții de implementare a moștenirii claselor în baza de date relațională, cunoscute sub numele de:

- a. mapare filtrată;
- b. mapare orizontală;
- c. maparea verticală.

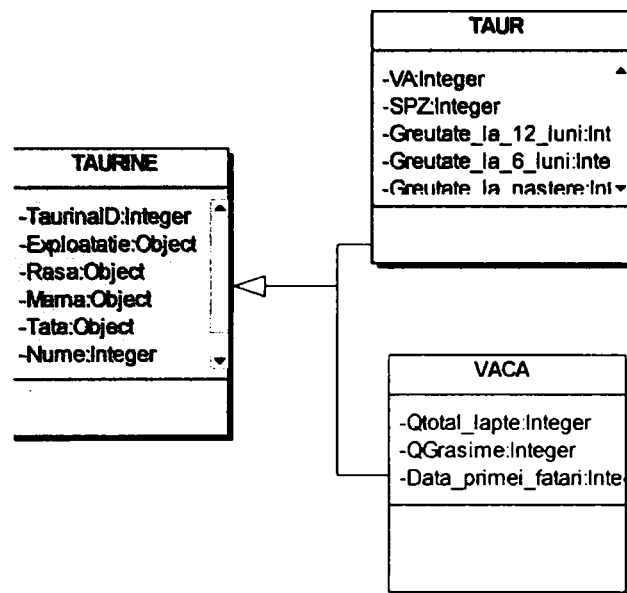


Figura 20. Moștenire simplă, aplicația „Ameliorare”. Diagrama UML

Maparea filtrată

Sunt mapate în aceeași tabelă toate clasele concrete ale ierarhiei.

Tabela va conține coloane pentru toate atributele claselor concrete din ierarhie. Există o filtrare a coloanelor și anume atributele comune sunt luate o singură dată. Clasele abstracte nu sunt mapate în tabelă.

Această abordare asigură o anumită performanță, dar, violează regulile de normalizare. Vor exista în tabelă tuple (rânduri) ale căror coloane vor avea valori NULL. Se preferă această implementare dacă clasa abstractă moștenește multe din atributele claselor concrete.

În ierarhia prezentată în figura 20, maparea filtrată va conduce la o singură tabelă (tabelul Taurina) prezentată în figura 21.

Coloana Sex asigură filtrarea datelor. Coloana DataPrimeiFatari va fi diferită de NULL numai pentru Sex=F, iar SporMZ va fi NULL pentru Sex=F. Din 23 de atribute doar 6 nu se regăsesc în clasa abstractă.

Column Name	Data Type	Length	Allow Nulls
TaurinaID	bigint	8	
ExploatareID	bigint	8	✓
RasaID	bigint	8	✓
TataID	bigint	8	✓
MamaID	bigint	8	✓
Codmatricol	varchar	50	✓
Nume	varchar	50	✓
Sex	varchar	1	✓
DataNasterii	datetime	8	✓
LoculNasterii	varchar	50	✓
ConditiiDeIntrare	varchar	50	✓
DataIntrarii	datetime	8	✓
ConditiiDeIesire	varchar	50	✓
DataIesiri	datetime	8	✓
Ameliorare	varchar	50	✓
DateSanatate	varchar	1024	✓
Fototaurina	image	16	✓
DataPrimeiFatari	datetime	8	✓
ValoareDeAmeliorare	float	8	✓
GreutateNastere	float	8	✓
Greutate6Luni	float	8	✓
Greutate12Luni	float	8	✓
SporMZ	float	6	✓

Figura 21. Maparea filtrată, aplicația „Ameliorare”

Maparea filtrată este cea mai simplă metodă de implementare. Adăugarea de noi clase în ierarhie presupune adăugarea de noi coloane în tabelă. Accesul la date este rapid, toate atributele se regăsesc în aceeași tabelă, nu este nevoie de mai multe accesări ale memoriei pentru citiri și scrieri-actualizări.

Datorită faptului că toate atributele claselor se regăsesc într-o tabelă, iar multe dintre ele pot avea valori <NULL>, apare o risipă a spațiului ocupat, tabela poate fi mare și acest lucru se resimte mai ales în spațiul ocupat în memoria RAM. Acest dezavantaj se accentuează în cazul unei ierarhii complexe.

În aplicația „Ameliorare”, ierarhia simplă prezentată în figura 21, a fost mapată prin această metodă.

Mapare orizontală

În cazul mapării orizontale, fiecare clasă concretă este mapată într-un tabel (Figura 22). În acest caz tabelele vor conține toate atributele clasei concrete, plus atributele moștenite de la clasa părinte abstractă, clasele abstracte nu se vor regăsi mapate în tabele. Aceasta abordare nu este dintre cele mai performante, dar este cea mai simplă de implementat. Dezavantajele sunt multiple, modificarea unei atribut din clasa abstractă va declanșa modificarea respectivă în toate clasele concrete care moștenesc clasa părinte.

Column Name	Data Type	Length	Allow Nulls	Column Name	Data Type	Length	Allow Nulls
TaurID	bigint	8		VaccaID	bigint	8	
ExploatareID	bigint	8	✓	ExploatareID	bigint	8	✓
RasaID	bigint	8	✓	RasaID	bigint	8	✓
TataID	bigint	8	✓	TataID	bigint	8	✓
MamaID	bigint	8	✓	MamaID	bigint	8	✓
Codmatricol	varchar	50	✓	Codmatricol	varchar	50	✓
Nume	varchar	50	✓	Nume	varchar	50	✓
Sex	varchar	1	✓	Sex	varchar	1	✓
DataNasterii	datetime	8	✓	DataNasterii	datetime	8	✓
LoculNasterii	varchar	50	✓	LoculNasterii	varchar	50	✓
ConditiiDeIntrare	varchar	50	✓	ConditiiDeIntrare	varchar	50	✓
DataIntrarii	datetime	8	✓	DataIntrarii	datetime	8	✓
ConditiiDeIesire	varchar	50	✓	ConditiiDeIesire	varchar	50	✓
DataIesirii	datetime	8	✓	DataIesirii	datetime	8	✓
Ameliorare	varchar	50	✓	Ameliorare	varchar	50	✓
DateSanatate	varchar	1024	✓	DateSanatate	varchar	1024	✓
Fototaurina	image	16	✓	Fototaurina	image	16	✓
ValoareDeAmeliorare	float	8	✓	[DataPrimei Fatari]	datetime	8	✓
GreutateNastere	float	8	✓				
Greutate6Luni	float	8	✓				
Greutate12Luni	float	8	✓				
SporMZ	float	8	✓				

Figura 22 . Maparea orizontală.

Maparea verticală

În cazul mapării verticale fiecare clasă din arbore este mapată într-o tabelă, indiferent dacă clasa este abstractă sau concretă. În această situație, fiecare tabelă mapată de clasa abstractă, va trebui legată de clasa părinte, prin intermediul unei cheii străine (FK) care referă cheia primară (PK) din tabela părinte (Figura 23). În cazul

implementării prin această metodă, instanțierea unui obiect din clasa concretă se realizează prin intermediul unei interogări JOIN.

Există performanțe la citire și scriere datorită faptului că citirea și scrierea se face într-o singură tabelă, spațiul consumat este minim, nu există attribute redondante, mai puțin cheia primară a tabelului mapată de clasa părinte care apare ca și cheie străină în tabelele mapate de clasele abstracte.

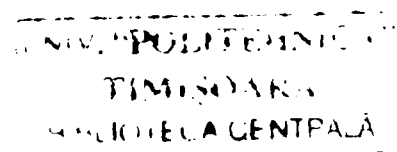
Această metodă simulează perfect moștenirea în arbore, dar consecințele se simt prin performanțele reduse datorită interogărilor necesare la instanțierea obiectelor, mai ales în cazul moștenirilor adânci.

	Column Name	Data Type	Length	Allow Nulls
PK	TaurinaID	bigint	8	
	ExploatareID	bigint	8	✓
	RasaID	bigint	8	✓
	TataID	bigint	8	✓
	MamaID	bigint	8	✓
	Codmatricol	varchar	50	✓
	Nume	varchar	50	✓
	Sex	varchar	1	✓
	DataNasterii	datetime	8	✓
	LoculNasterii	varchar	50	✓
	ConditiiDeIntrare	varchar	50	✓
	DataIntrarii	datetime	8	✓
	ConditiiDeIesire	varchar	50	✓
	DataIesirii	datetime	8	✓
	Ameliorare	varchar	50	✓
	DateSanatate	varchar	1024	✓
	Fototaurina	image	16	✓

	Column Name	Data Type	Length	Allow Nulls
PK	TaurID	bigint	8	
	TaurinaID	bigint	8	✓
	ValoareDeAmeliorare	float	8	✓
	GreutateNasterii	float	8	✓
	Greutate6Luni	float	8	✓
	Greutate12Luni	float	8	✓
	SportMZ	float	8	✓

	Column Name	Data Type	Length	Allow Nulls
PK	VacaID	bigint	8	
	TaurinaID	bigint	8	✓
	DataPrimeiFetari	datetime	8	✓

Figura 23. Maparea verticală.



Identitatea obiectelor

Identitatea este proprietatea unui obiect (instanțiat dintr-o clasă) prin care se distinge de alte obiecte. Identitatea este un concept extern obiectului, introdus în scopul de a asigura o modalitate de reprezentare a individualității lui. Identitatea unui obiect trebuie să se păstrează și după ce obiectul a fost supus unor transformări, sau și-a modificat starea internă

Identitatea obiectului este bine cunoscută în limbajele de programare orientate obiect, dar acest concept are un sens și o semnificație deosebită în cazul bazelor de date [BI93]. Prin identitatea unui obiect trebuie să se înțeleagă că acel obiect există independent de valorile lui.

Se disting două noțiuni legate de echivalența a două obiecte:

- **două obiecte pot fi identice** (ele sunt același obiect) Referințele a două obiecte sunt identice dacă ele referă același obiect în memorie.
- **două obiecte pot fi egale** (ele au aceleași valori ale atributelor).

Două obiecte sunt identice (în aplicație) dacă reprezintă același obiect din baza de date (împart același spațiu de memorare, reprezintă același rând în baza de date dacă obiectul este mapat într-o tabelă, au aceeași valoare a cheii primare).

Dacă, în cazul bazelor de date relaționale, tuplele se identifică prin conținutul lor, în cazul obiectelor, acest lucru nu este realizabil deoarece pot exista două obiecte cu aceleași atribute. Pentru a le identifica, este necesară introducerea identificatorului de obiect (OID *Object Identifier*).

Identitatea obiectului este absolut necesară pentru a asigura persistența obiectului. **În sistemele de gestiune a bazelor de date, OID, ca dată, trebuie să fie persistent.**

Bazele de date „*non-object-oriented*”, asigură accesul la date persistente prin intermediul unei valori care le identifică în mod unic. În cazul bazelor de date relaționale, valoarea care identifică în mod unic un rând (tuplu) din tabelă este cheia primară (PK), ea este folosită pentru accesul la datele persistente. Această formă de acces la date persistente nu poate fi folosită și în cazul modelului orientat obiect datorită faptului că într-o bază de date (obiecte), pot exista două obiecte a căror valori sunt identice. Aplicațiile orientate obiect lucrează cu baze de date ce nu respectă prima formă normală, ca urmare, OID este absolut necesar pentru accesul la obiectele din baza de date.

În cazul soluției de implementare a persistenței obiectelor prin **GOP**, asigurarea unui OID pentru obiecte este limitată și poate fi asigurată doar în bazele de date care au la bază modelul relațional, rețea sau flat file.

Identitatea unui rând (tuplu) din baza de date relațională este exprimată prin valoarea cheii primare. În condițiile existenței asocierilor între tabele, materializarea asocierii se face prin intermediul cheii străine, ca urmare alegerea cheii primare este foarte importantă, aceasta nu poate fi un câmp (coloană) a tablei care să fie actualizabil, deoarece modificarea lui implică modificarea și în tabelele cu care aceasta tabelă este în asociere, ca urmare se recomandă să se alege o cheie primară o coloană a tablei ce nu semnifică nimic pentru utilizator, ea este cunoscută sub numele de cheie surogat (*surrogate keys*), aceasta va fi cheia primară.

În aplicația *Ameliorare*, tabela Proprietari are cheie primară ProprietarID care este o cheie surogat, ea nu semnifică nimic pentru utilizator. Tabela Proprietari se află în

relație 1:n cu tabela Exploatații, a cărei cheie primară, la fel ca și în cazul tabelului Proprietari, este ExploatațieID. ProprietarID și ExploatațieID, cheile primare ale tabelului Proprietari și Exploatații conform definiției nu pot fi modificate în cadrul aplicației, ele nu prezintă nici o semnificație pentru utilizator, au valori unice generate de baza de date sau de aplicație. Figura 24 prezintă câmpurile tabelului Exploatații și Proprietari cu cheile primare definite ca identificatori (Microsoft SQL Server 2000 - Identity Yes, Identity 1, Identity Increment 1)

Column Name	Data Type	Length	Allow Nulls
ExploatațieID	bigint	8	
ProprietarID	bigint	8	✓
CodExpl	varchar	50	✓
LocațieExpl	varchar	50	✓
ObservațieExpl	varchar	1000	✓

Column Name	Data Type	Length	Allow Nulls
ProprietarID	bigint	8	
NumePropr	varchar	50	✓
PrenumePropr	varchar	50	✓
CNP_Propr	varchar	50	✓
LocațiePropr	varchar	50	✓
ObservațiePropr	varchar	50	✓

Figura 24 Cheile primare ale tabelului Exploatații și Proprietari

Implementarea identificatorului ProprietarID în clasa Proprietari (aplicația Ameliorare) este prezentat în figura 25.

```

public class Proprietari:BaseBusinessObject
{
    private long proprietarID = -1;
    private string numePropr = null;
    private string prenumePropr = null;
    private string cnp_Propr = null;
    private string locațiePropr = null;
    private string observatiiPropr=null;

    .....
    public long ProprietarID
    {
        get
        {
            return this.proprietarID;
        }
        set
        {
            this.proprietarID = value;
        }
    }
}

```

Figura 25. Implementarea clasei Proprietari (C# . NET)

Obținerea identității generate pentru o instanță (p) a clasei Proprietari este prezentată în figura 26 :

```
ProprietarID = ReadID(this.sqlConnection);
protected long ReadID(SqlConnection con)
{
    long res = -1;
    //setare comanda pt obtinerea identitatii generate
    this.selectSqlCommand = new SqlCommand("SELECT"+
        "@@IDENTITY AS ID", this.sqlConnection);
    //obtinerea ultimei valori
    res = long.Parse(
selectSqlCommand.ExecuteScalar().ToString());
    return res;
}
```

Figura 26. Obținerea identității generate pentru o instanță (C# . NET)

Unicitatea OID trebuie asigurată la trei nivele:

1. în interiorul clasei;
2. în ierarhia de clase;
3. peste toate clasele

În **SGBDRO**, OID este asigurat prin crearea unui ID (identificator) unic pentru fiecare rând al tabelii. Acest ID este independent de valorile memorate în rândul tabelii, este generat automat prin incrementare de către DDL. Fiecare rând din orice tabelă a bazei de date relaționale poate fi accesată în mod unic prin intermediul acestui ID.

În **SGBDOO**, fiecare obiect primește un identificator care este permanent și este asociat cu acel obiect, nu depinde de structura și starea obiectului. Identitatea obiectului este independentă de locația sau adresa de memorie și permite utilizatorului referirea partajată a obiectului. Faptul că, prin OID, un obiect (din BDOO) poate fi referit în mod partajat, asigură utilizarea BDOO în domenii de aplicații cum ar fi CAD,CAM, CASE sau intelligent offices.

SGBDOO generează automat OID pentru obiecte. El nu se schimbă niciodată pe durata execuției unei aplicații și nu se bazează pe valorile memorate în obiect ca în cazul bazelor de date relaționale unde cheia primară identifică în mod unic un rând (n-uplet) din tabelă. Conceptul de OID fac să fie ușor de controlat memorarea obiectelor și construirea de legături între obiecte.

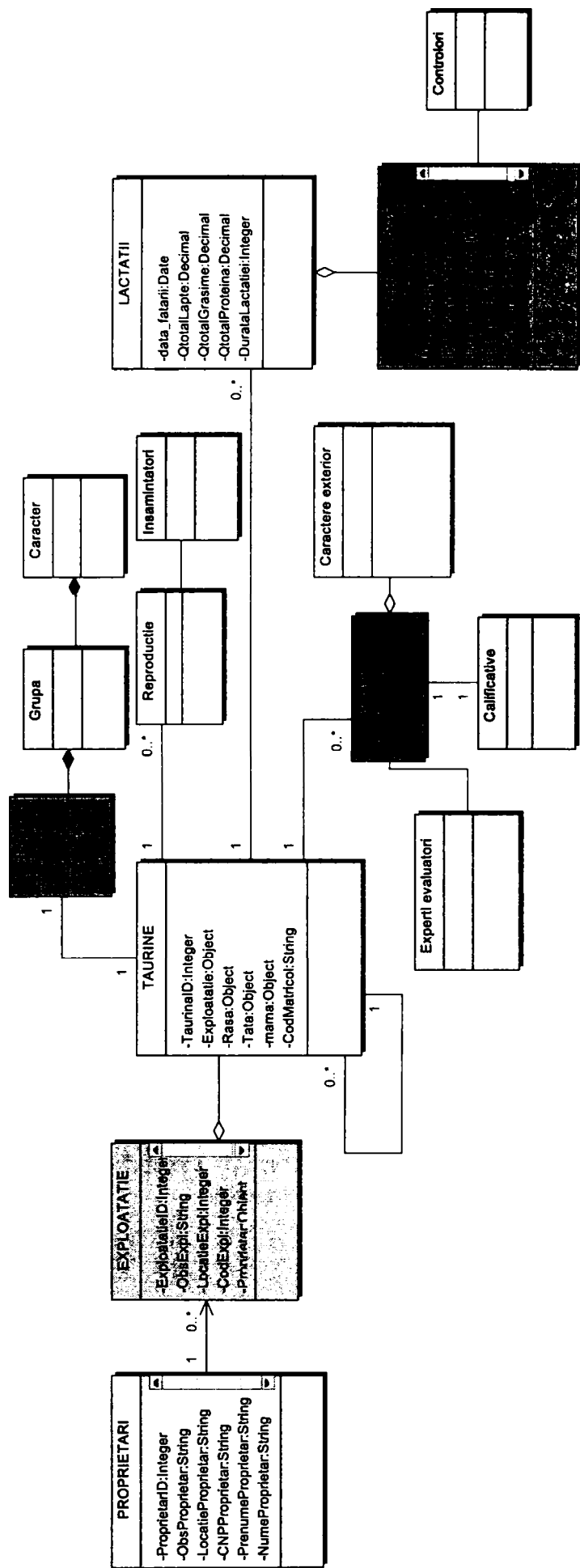


Figura 28. Maparea tabelelor în clase. Diagrama claselor (UML)

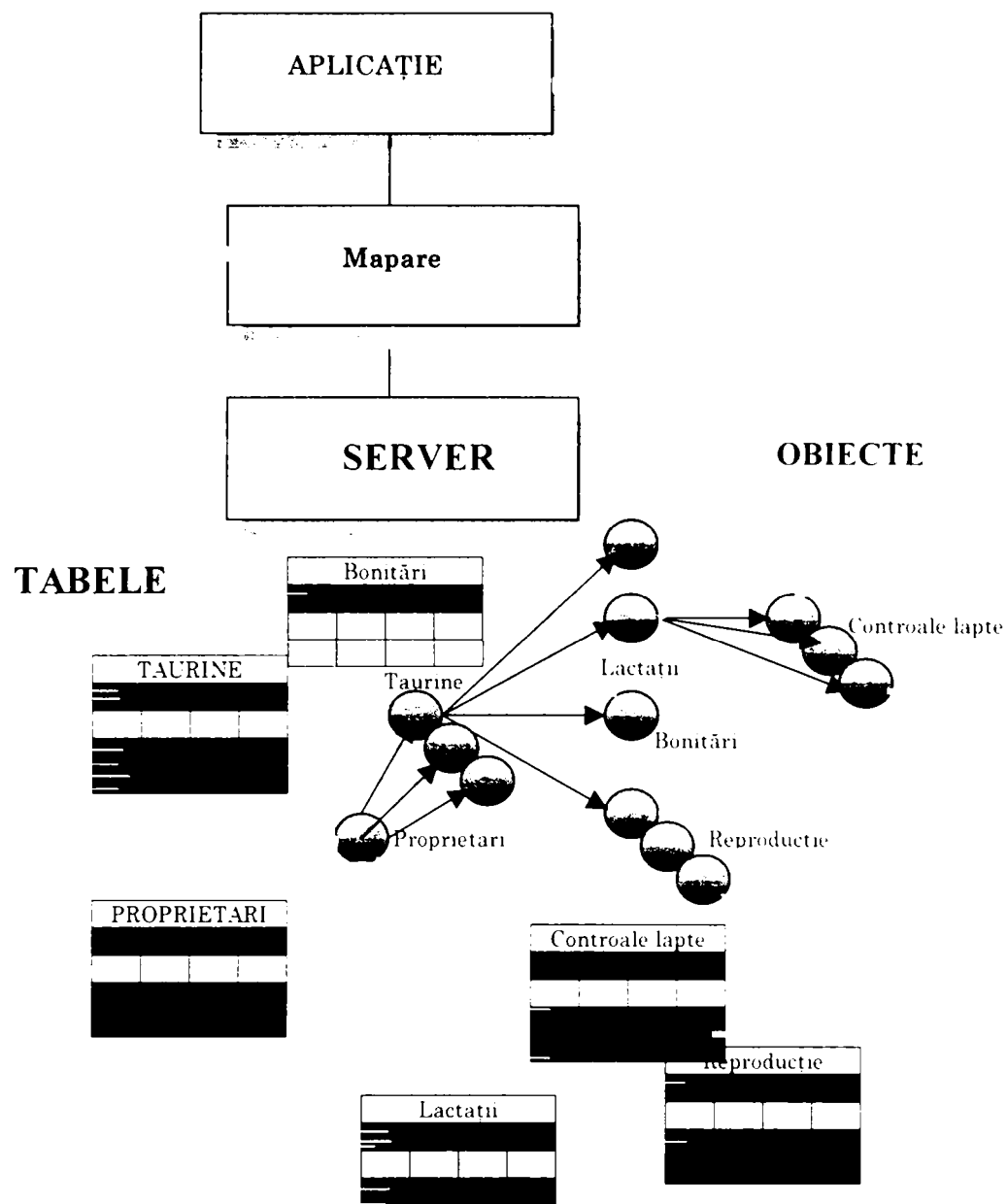


Figura 29 . Maparea obiectelor în tabele și a tabelor în obiecte în aplicațiile orientate obiect ce lucrează cu baze de date relaționale

3.7.3. Accesarea datelor

Analiza din punct de vedere al modului în care se face accesul la date în aplicațiile ce lucrează cu date persistente gravitează în jurul a trei probleme:

1. crearea datelor persistente
2. navigarea în baza de date
3. interogarea bazei de date

Crearea datelor persistente

Există două moduri de a adăuga persistență instanțelor obiectelor :

1. operatorul *new*
2. definirea și implementarea unor clase care se îngrijesc de asigurarea persistenței instanțelor obiectelor în baza de date.

Operatorul *new* are rolul de a crea în aplicație o instanță a clasei, urmând apoi ca tot prin aplicație să se asigure persistența ei.

```
//se creeaza o taurina noua  
Taurina t_tranzient = new Taurina();
```

Această instrucțiune are rolul de a crea un obiect tranzient, urmând apoi ca să fie materializat ca persistent în baza de date.

În unele SGBDOO (ObjectStore și O2) acest lucru se realizează astfel:

```
//se materializează persistența obiectului :  
Taurina t_persistent = new (myDB)Taurina();
```

Definirea și implementarea unor clase care se îngrijesc de asigurarea persistenței instanțelor obiectelor în baza de date reprezintă modalitatea cea mai întâlnită pentru materializarea persistenței.

Clasele de tip DAO (*Data Access Object*) au rolul de a asigura separarea interfeței aplicației de (API) de mecanismul de materializare a persistenței obiectelor.

DAO este un tipar (*pattern*) și prezintă avantajul că modificările survenite și legate de baza de date se efectuează doar în aceste clase, fără a modifica restul aplicației. Prezentarea modului de implementare a DAO în aplicația *Ameliorare* este prezentat în capitolul Arhitectura Software.

Se prezintă în continuare clasa *RasaDAO* ce materializează operațiile *CRUD* (*create, read, update delete*), navigarea și interogarea cu tabela *Rasa* .

```
using System;  
using System.Collections;  
using System.Diagnostics;  
using Ameliorare.BusinessLayer.BusinessObjects;  
using System.Data.SqlClient;  
namespace Ameliorare.DataLayer.DAO  
{
```

```

public class RasaDAO : BaseDAO
{
    private System.Data.SqlClient.SqlCommand deleteSqlCommand;
    private System.Data.SqlClient.SqlCommand insertSqlCommand;
    private System.Data.SqlClient.SqlCommand selectSqlCommand;
    private System.Data.SqlClient.SqlCommand updateSqlCommand;
    private System.ComponentModel.Container components = null;
    public RasaDAO(System.ComponentModel.IContainer
container):base(container)...
    public RasaDAO():base()...
    protected override void Dispose( bool disposing ) ...
    public Rasa FindRasaByPK(long rasaID) ...
    public void LoadAll(ArrayList a) ...
    public void SaveRasa(Rasa ras) ...
    public void Delete(Rasa ras) ...
    public void Delete(long rasaID) ...
    private void UpdateToDB(Rasa ras) ...
    private long InsertIntoDB(Rasa ras) ...
    protected long ReadID(SqlConnection con) ...
}
}

```

Metoda *SaveRasa* este implementată astfel:

```

public void SaveRasa(Rasa ras)
{
    if(ras.RasaID!=-1)
    {
        this.UpdateToDB(ras);
    }
    else
    {
        this.InsertIntoDB(ras);
    }
}

```

Revenind la modul în care se materializează persistența instanțelor obiectelor în aplicațiile ce lucrează cu baze de date relaționale, instanțierea în memorie a clasei Taurină este urmată de apelarea unei instanțe a clasei în care este implementat accesul la baza de date (TaurineDAO) cu salvarea instanței clasei taurină.

```

//se creeaza o taurina noua
Taurina t = new Taurina();
// instanțierea clasei DAO și apelarea metodei de
// materializarea persistenței obiectului
public void SaveTaurina()
{
    TaurineDAO dao = new TaurineDAO();
    dao.SaveTaurina(this.taurinaCurenta);
}

```

La fel ca și în cazul metodei *SaveRasa*, metoda *SaveTaurina*, funcție de valoarea *TaurinaID*, execută adăugarea unei noi înregistrări (tuplu) în baza de date sau o actualizează (*update*) pe cea al cărui ID îi corespunde. Diferența între *UpdateToDB* și *InsertIntoDB* constă în faptul că în condiția în care *taurinaID=-1*, se execută *INSERT* în baza de date, obiectul devine persistent în urma execuției comenzii și închiderea conexiunii, iar dacă *taurinaID* are o valoare diferită de *-1* se execută *UPDATE*, obiectul devine "*detached instance*".

```

private long InsertIntoDB(Rasa ras)
{
    //setare comanda pt sql
    this.insertSqlCommand = new SqlCommand("INSERT
INTO Rase( GrupaID, CodRasa, DenumireRasa, DescriereRasa)" +
" VALUES ( @GrupaID, @CodRasa, @DenumireRasa, @DescriereRasa; ",
this.sqlConnection);
    //adaugare parametrilor
    this.insertSqlCommand.Parameters.Add("@GrupaID", System.Data.
SqlDbType.BigInt);
    this.insertSqlCommand.Parameters.Add("@CodRasa", System.Data.
SqlDbType.VarChar);
    this.insertSqlCommand.Parameters.Add("@DenumireRasa", System.
Data.SqlDbType.VarChar);
    this.insertSqlCommand.Parameters.Add("@DescriereRasa", System
.Data.SqlDbType.VarChar);
    //setare valori pentru parametrilor
    this.insertSqlCommand.Parameters["@GrupaID"].Value=
ras.GrupaID;
    this.insertSqlCommand.Parameters["@CodRasa"].Value =
ras.CodRasa==null?"":ras.CodRasa;
    this.insertSqlCommand.Parameters["@DenumireRasa"].Value =
ras.DenumireRasa==null?"":ras.DenumireRasa;
    this.insertSqlCommand.Parameters["@DescriereRasa"].Value =
ras.DescriereRasa==null?"":ras.DescriereRasa;
    try
    {
        //deschidem conexiunea spre db
        this.sqlConnection.Open();
        //executie comanda sql pe conexiune cu
returnarea numarului de randuri introduse
        insertSqlCommand.ExecuteNonQuery();
        ras.RasaID = ReadID(this.sqlConnection);
    }
    catch(Exception ex)
    {
        Debug.WriteLine("EROARE: InsertIntoDB
"+ras+" "+ex.Message+" "+ex.StackTrace);
    }
    finally
    {
        //inchidere conexiune
        this.sqlConnection.Close();
    }
    return 1;
}

```

```

private void UpdateToDB(Rasa ras)
{
    //setare comanda pt sql
    this.updateSqlCommand= new SqlCommand("UPDATE Rase SET
GrupaID=@GrupaID, CodRasa=@CodRasa, DenumireRasa=@DenumireRasa,
DescriereRasa=@DescriereRasa WHERE RasaID=@RasaID",
this.sqlConnection);

//adaugare parametrii

    this.updateSqlCommand.Parameters.Add("@RasaID", System.Data.S
qlDbType.BigInt);
    this.updateSqlCommand.Parameters.Add("@GrupaID", System.Data.
SqlDbType.BigInt);
    this.updateSqlCommand.Parameters.Add("@CodRasa", System.Data.
SqlDbType.VarChar);
    this.updateSqlCommand.Parameters.Add("@DenumireRasa", System.
Data.SqlDbType.VarChar);
    this.updateSqlCommand.Parameters.Add("@DescriereRasa", System
.Data.SqlDbType.VarChar);
    this.updateSqlCommand.Parameters["@RasaID"].Value =
ras.RasaID;
    this.updateSqlCommand.Parameters["@GrupaID"].Value =
ras.GrupaID;
    this.updateSqlCommand.Parameters["@CodRasa"].Value =
ras.CodRasa;
    this.updateSqlCommand.Parameters["@DenumireRasa"].Value =
ras.DenumireRasa;
    this.updateSqlCommand.Parameters["@DescriereRasa"].Value =
ras.DescriereRasa;
    try
    {
        //deschidem conexiunea spre db
        this.sqlConnection.Open();
        updateSqlCommand.ExecuteNonQuery();
    }
    catch(Exception ex)
    {
        Debug.WriteLine("EROARE: UpdateToDB "+ras+"
"+ex.Message+" "+ex.StackTrace);
    }
    finally
    {
        //inchidere conexiune
        this.sqlConnection.Close();
    }
}

```

Citirea datelor persistente se face la fel, într-un mod transparent. Clasa *TaurinaDAO* are implementată metoda *LoadTaurinaByID(long taurinaID, Taurina t)*, care interoghează baza de date, găsește rândul (tuplu) ce se identifică cu taurină ID, trimis ca parametru și returnează în obiectul t de tip *Taurina*, obiectul persistent.

```

public void LoadTaurinaByID(long taurinaID, Taurina t)
{
    this.selectSqlCommand = new SqlCommand("SELECT * FROM
Taurine WHERE TaurinaID=@TaurinaID", this.sqlConnection);
    this.selectSqlCommand.Parameters.Add("@TaurinaID", System.Data
a.SqlDbType.BigInt);
    this.selectSqlCommand.Parameters["@TaurinaID"].Value =
taurinaID;
        try
        {
            this.sqlConnection.Open();
            SqlDataReader reader =
selectSqlCommand.ExecuteReader();
            while(reader.Read())
            {
                TransferFromReader(reader,t);
            }
        }
        catch(Exception ex)
        {
            Debug.WriteLine("EROARE: FindTuarinaByPK
"+taurinaID+" "+ex.Message+" "+ex.StackTrace);
        }
        finally
        {
            this.sqlConnection.Close();
        }
    }
}

```

Navigarea în baza de date

Accesul la obiectele unei baze de date se face prin navigare, cunoscut și sub numele de „traversare”. Termenul este utilizat datorită faptului că obiectele bazei de date au structura unui „graf”, a cărui traversare presupune trecerea de la un nod la altul, deci de la un obiect la altul, funcție de structura și relațiile ce există între obiecte.

SGBDOO ce lucrează cu „baze de obiecte” nu prezintă facilitatea de a utiliza o valoare ca și cheie pentru navigarea în baza de date. Structura obiectelor prezintă legăturile (asocierile) necesare interogărilor. Cu cât interogările sunt mai complexe, cu atât sunt necesare mai multe asocieri între obiecte, structura acestora se complică și resursele și costurile interogărilor cresc.

SGBDOO existente au fost proiectate să lucreze cu limbaje de programare orientate obiect ca C#, C++, și Java, iar asigurarea persistenței este asigurată într-un mod transparent (*transparent persistence*) Manipularea și traversarea obiectelor persistente este realizată de limbajele de programare în aceeași manieră ca în cazul obiectelor nepersistente memorate în memoria internă (volatilă).

Schematic, limbajele de programate orientate obiect realizează acest lucru în felul următor:

1. Obiectul *Taurină* referă obiectul *Lactație* în baza de date.
2. Pentru a modifica date referitoare la obiectul *Lactație*, aplicație interoghează baza de date;
3. Obține în memorie (*cache*) obiectul *Taurină*;
4. Aplicația solicită serverului obiectul *Lactație*;
5. Efectuează modificările;
6. Trimite spre memorare în baza de date obiectul *Lactație* cu modificările efectuate.

Modul acesta de a implementa persistența transparentă este cunoscută sub numele de *caching*. În figura 30 și figura 31 se prezintă modul de efectuare al modificărilor într-o bază de date orientată obiect.

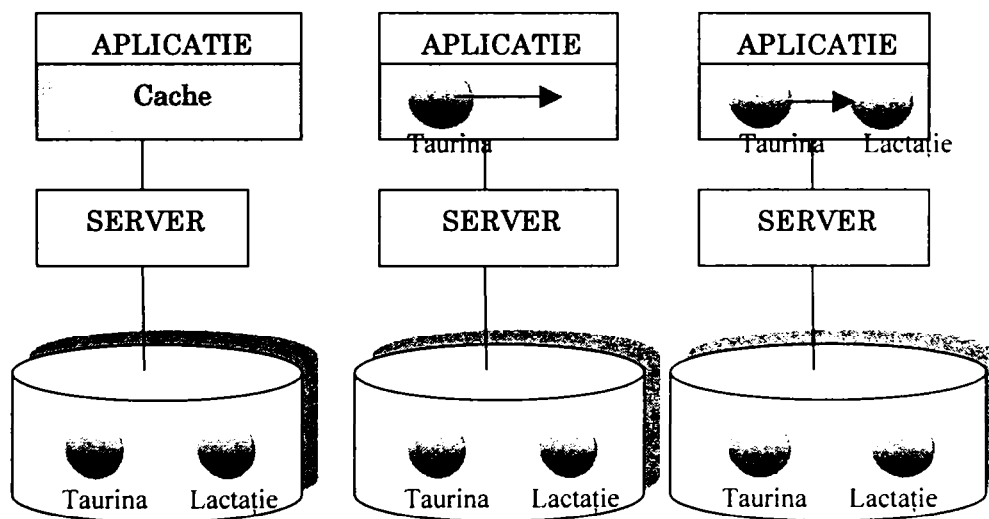


Figura 30. Obținerea în memorie a obiectului lactație pentru efectuarea modificărilor

Aplicația efectuează modificările asupra obiectului Lactație și îl trimite pe server spre memorare în baza de date. Obiectul modificat este colorat în albastru.

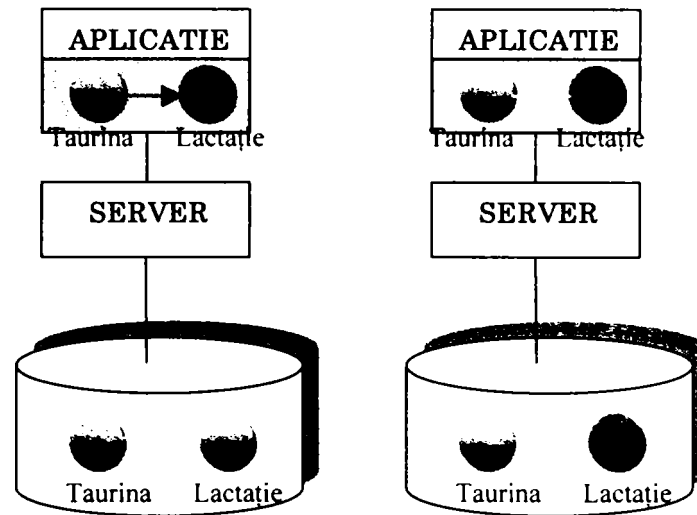


Figura 31. Obiectul Lactație modificat este salvat în baza de date

Cachingul este tehnica utilizată în asigurarea persistenței obiectelor prin păstrarea lor în memorie (*cache*) în vederea efectuării modificărilor și apoi memorarea lor pe disc. *Cache*-ul este utilizat pentru maparea transparentă între obiectele bazei de date și obiectele aplicației, eliminând necesitatea alocării, respectiv dealocării, de memorie din aplicație pentru accesarea obiectelor bazei de date. Aplicația scrisă în limbaj orientat obiect nu trebuie explicit să se ocupe de mutarea obiectelor pe disc, acest lucru se întâmplă automat ca rezultat al interogării (Figura 32).

```

import javax.jdo.*;
import java.util.Collection;

PersistenceManagerFactory pmf;
// assume this variable
// is set
pmf.setConnectionUserName(" ");
pmf.setConnectionPassword(" ");
pmf.setConnectionURL("database URL");
PersistenceManager pm = pmf.getPersistenceManager();
Transaction txn = pm.currentTransaction();

txn.begin();
Extent animal = pm.getExtent(Taurina.class, false);

// perform query
Query query = pm.newQuery(Taurina.class, animal, "name ==
\"Florica\"");
Collection result = (Collection) query.execute();
Iterator iter = result.iterator();
while ( iter.hasNext() )
{
    Taurina taurina = (Taurina) iter.next();

```

```

se solicita obiectul lactat in ved
modificărilor
    taurina.lactatie.nrlactatie = "2";
}

txn.commit();

```

Figura 32 Interfață ce realizează persistența transparentă cu o bază de date orientată obiect sau relațional obiectuală. (Java Data Objects (JDO))

„Navigarea” în sistemul GOP este realizată prin mapare în aplicația ce lucrează cu baza de date „*non object oriented*”. Relațiile (asocierile) ce există între obiecte sunt mapate în tabele, iar accesul la fiecare obiect se face prin intermediul interogărilor (interogări SQL). Figura 33, ilustrează schematic modul în care prin intermediul mapării obiectele ajung în memorie și sunt utilizate de aplicația orientată obiect.

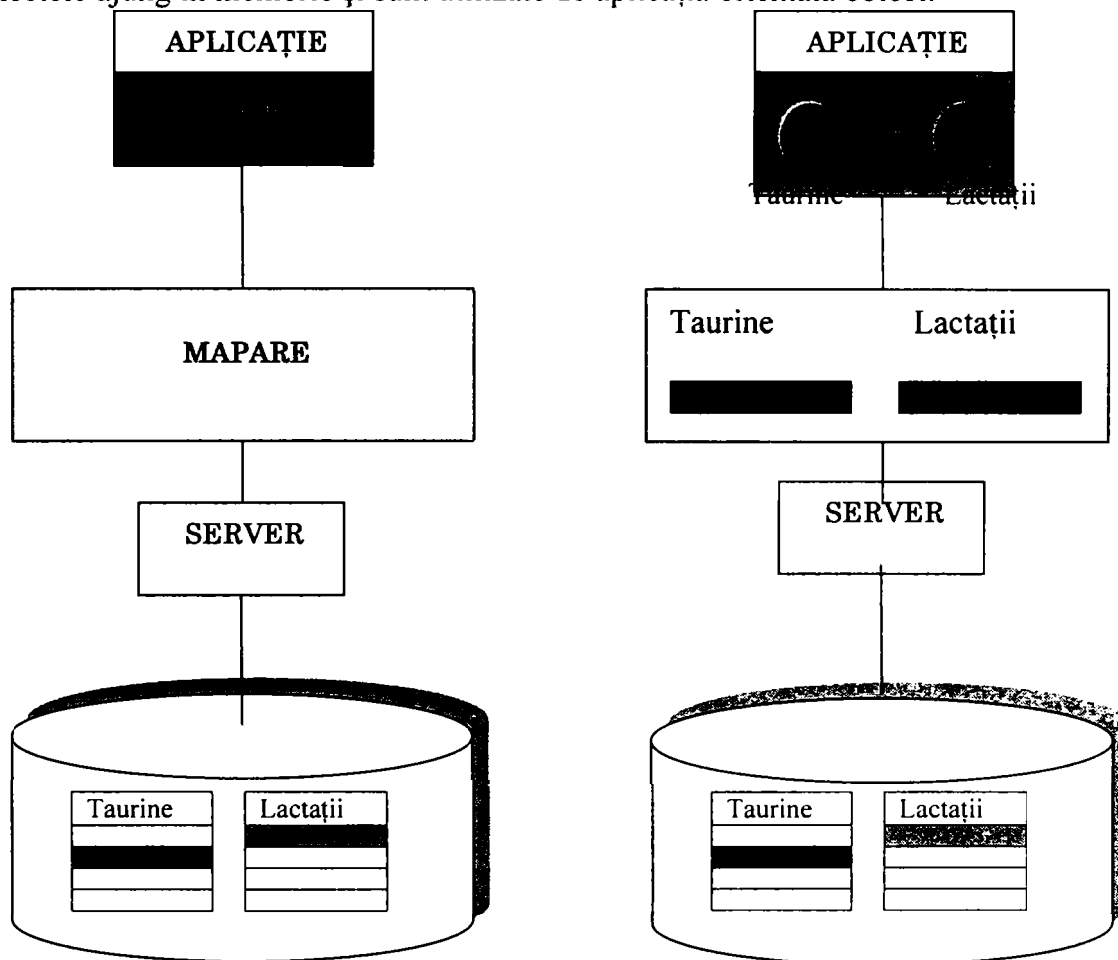


Figura 33. Accesul la date într-o bază de date relațională utilizând o aplicație orientată obiect

Numărul mare de accesări ale memoriei externe reduc performanța, iar creșterea acestora se poate face doar prin minimizarea numărului de interogări ale bazei de date.

Numărul de tabele inclus în interogarea JOIN determină adâncimea grafului de obiecte ce trebuie navigat. Este necesară câte o instrucțiune SELECT pentru fiecare nod ale grafului de obiecte.

Interogarea bazei de date

Interogarea bazei de date reprezintă funcția cea mai importantă operație de manevrare a datelor într-o bază de date.

Interogarea se face în scopul obținerii (recuperării) instanței unei clase din baza de date.

O interogarea complexă solicită resurse care duc la scăderea performanței, acestea se multiplică în condițiile în care există „*impedance mismatch*”.

Pentru formularea interogărilor în bazele de date relaționale se folosesc limbaje de interogare bazate pe algebra relațională și calculul relațional [Am03, Da95], cel mai adesea SQL. Limbajul de interogare lucrează pe date care nu corespund modelului utilizat în aplicație (orientata obiect), ca urmare performanța aplicației este afectată de „*impedance mismatch*”.

Rezolvarea cea mai simplă o găsim în cazul SGBDRO, unde serverele de baze de date (SQL Server 2000, Oracle) posedă un suport solid în vederea executării interogărilor (SQL).

În figura 34, sunt prezentate tabelele din baza de date implicate în interogarea utilizată în aplicația *Ameliorare* în vederea obținerii numărului de fiice ale unui taur în testare aflate într-o exploatație.

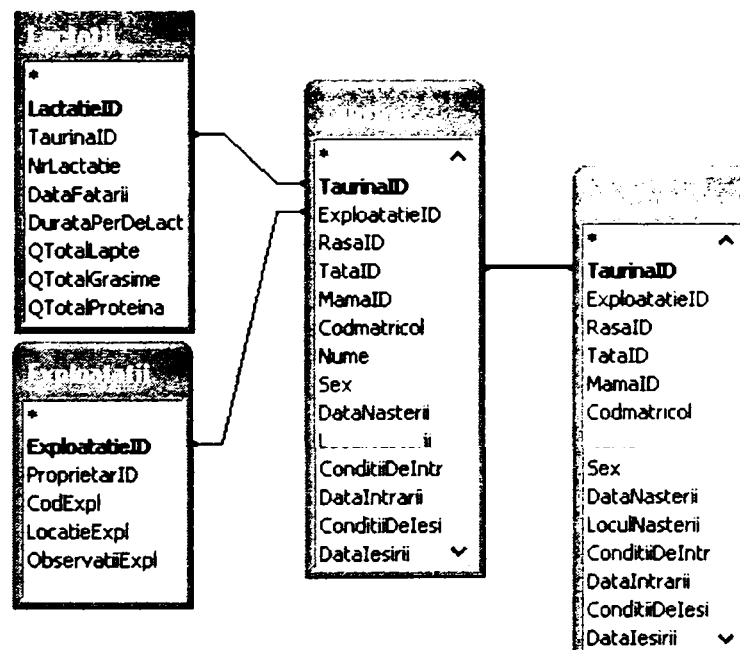


Figura 34. Tabele în interogarea utilizată în aplicația *Ameliorare* în vederea obținerii numărului de fiice ale unui taur în testare aflate într-o exploatație

```

        public long LoadNrFiiceDinFermaDinTata(long
exploatatieID,long tataID)
        {
            Taurina t= new Taurina();
            this.selectSqlCommand = new SqlCommand("SELECT
count(distinct(Taurine.TaurinaID)) as nr_fiice FROM ((Taurine
INNER JOIN Lactatii ON Taurine.TaurinaID = Lactatii.TaurinaID)
INNER JOIN Exploatatii ON Taurine.ExploatatieID =
Exploatatii.ExploatatieID) INNER JOIN Taurine AS Taurine_1 ON
Taurine.TataID = Taurine_1.TaurinaID WHERE
(((Lactatii.NrLactatie)=1)and ( Taurine_1.Ameliorare='TT')and
(Taurine.ExploatatieID=@ExploatatieID) and
(Taurine.tataID=@TataID))", this.sqlConnection);

            //adaugare parametrui

            this.selectSqlCommand.Parameters.Add("@ExploatatieID",System
.Data.SqlDbType.BigInt);
            this.selectSqlCommand.Parameters.Add("@TataID",System.Data.S
qlDbType.BigInt);
            //setare valori pentru parametrui
            this.selectSqlCommand.Parameters["@ExploatatieID"].Value =
exploatatieID;
            this.selectSqlCommand.Parameters["@TataID"].Value = tataID;
            try
            {
                //deschidem conexiunea spre db
                this.sqlConnection.Open();
                //executie comanda sql pe conexiune cu
returnarea rezultatelor intr-un reader
                SqlDataReader reader =
selectSqlCommand.ExecuteReader();
                //transfer din reader in obiect
                while(reader.Read())
                {
                    t.Nr_fiice=(int)reader["Nr_fiice"];
                }
            }
            catch(Exception ex)
            {
                //
                Debug.WriteLine("EROARE:
LoadNrFiiceDinFermaDinTata "+tataID+" "+ex.Message+"
"+ex.StackTrace);
            }
            finally
            {
                //inchidere conexiune
                this.sqlConnection.Close();
            }
            return t.Nr_fiice;
        }
    }

```

Limbajul de interogare suportat de SGBDOO reprezintă o extensie a limbajelor de programare orientate obiect pentru care au fost proiectate, interogarea obiectelor se face cu un limbaj foarte apropiat de SQL, căruia i-au fost adăugate anumite extensii pentru a-l adapta lumii obiectelor:

- suplinirea tuturor claselor cu metoda SELECT;
- extinderea limbajelor de programare pentru a include SQL ca predicat pentru filtrul de selecție al operațiilor;
- introducerea instrucțiunii SQL, SELECT, în limbajele de programare orientate obiect și asigurarea cu un preprocesor.

În felul acesta, SGBDOO asigură un limbaj de interogare declarativ, integrat în limbajul de programare orientat obiect.

3.7.4. Partajarea datelor

Aplicațiilor orientate obiect ce lucrează cu baze de date relaționale trebuie să fie capabile să îndeplinească funcțiile sistemelor de gestiune a bazelor de date referitoare la accesul concurrent și refacerea datelor.

Tranzacțiile

În sistemele de gestiune a bazelor de date, tranzacțiile reprezintă mecanismul utilizat pentru implementarea concurenței și a refacerii [Io04].

Tranzacțiile sunt secvențe de program care ori sunt executate complet ori deloc. Ele trebuie să asigure trecerea datelor din baza de date, dintr-o stare consistentă în alta. Pentru a menține consistența tranzacțiile trebuie să îndeplinească următoarele caracteristici (ACID):

- Atomicitate;
- Consistență;
- Izolare;
- Durabilitate.

Tranzacția este *atomică* dacă operațiile ce o materializează se execută complet sau deloc. *Consistența* tranzacției constă în proprietatea acesteia de a efectua modificări corecte ale bazei de date, adică tranzacția să transforme baza de date dintr-o stare consistentă în altă stare consistentă. Starea unei baze de date este considerată consistentă dacă respectă toate constrângerile de integritate implicite sau explicite.

Atunci când asupra unei baze de date se execută în mod concurrent mai multe tranzacții, sau interferează operații ale tranzacțiilor concurente, ele trebuie să fie *izolate*. Modificările consistente făcute de un utilizator nu sunt văzute în exterior de alți utilizatori decât la validarea tranzacției, ele fiind blocate până la execuția (*commit*)

Durabilitatea reprezintă proprietatea prin care, după validarea unei tranzacții, modificările efectuate de acesta în baza de date nu vor mai fi pierdute datorită unor defectări ulterioare a sistemului. Proprietatea de durabilitate este asigurată prin metode de refacere ale sistemului de gestiune a bazei de date.

Accesul concurrent la date este asigurat de blocare (*locking*).

Există trei comenzi de bază pentru realizarea unei tranzacții: BeginTransaction, Commit, și Rollback. BeginTransaction marchează începutul tranzacției, comenzile incluse între Begin Transaction și oricare dintre comenzile Commit sau Rollback, constituie corpul tranzacției[Se03].

```
SqlConnection myConnection = new SqlConnection
("server = (local) \\ SQLExpress; Integrated Security =SSPI;
database=Ameliorare");
SqlCommand myCommand = new SqlCommand();
SqlTransaction myTrans;
    Deschid conexia.
myConnection.Open();
    // Asignez proprietate conexiei.
myCommand.Connection = myConnection;
    // Incepe tranzacția.
myTrans = myConnection.BeginTransaction();
myCommand.Transaction = myTrans;
    try
    {
        myCommand.CommandText = "DELETE FROM Taurine WHERE
(taurinaID = 100) OR (taurinaID = 101)";
        myCommand.ExecuteNonQuery();
        // Inserez prima inregistrare.
        myCommand.CommandText = "Insert into Taurine (taurinaID,
nume) VALUES (100, 'Florica)";
        myCommand.ExecuteNonQuery();
        // Inserez a doua inregistrare.
        myCommand.CommandText = "Insert into Taurine (taurinaID,
nume) VALUES (101, 'Viorica)";
        myCommand.ExecuteNonQuery();
        myTrans.Commit();
        Console.WriteLine("Ambele inregistrari au fost scrise in
baza de date!");
    }
    catch(Exception e)
    {
        myTrans.Rollback();
        Console.WriteLine(e.ToString());
        Console.WriteLine("Nici o inregistrare nu a fost scrisa in
baza de date!");
    }
    finally
    {
        myConnection.Close();
    }
}
```

3.7.5. Implementarea modelului matematic

Modelele matematice utilizate în predicția valorii de ameliorare prin metodologiei BLUP conduc la un sistem de $p+q$ ecuații, cu $p+q$ necunoscute (p efecte fixe și q efecte aleatoare), a cărui soluții se pot obține prin rezolvarea prin metode numerice a sistemelor de ecuații lineare[Be97].

Soluțiile sistemului de ecuații $\begin{bmatrix} \tilde{b} \\ \hat{u} \end{bmatrix}$ reprezintă:

b – vectorul efectelor fixe, $m \times 1$;

u – vectorul efectelor aleatoare $p \times 1$

Efectul aleator necunoscut reprezintă capacitatea de transmitere a reproducătorilor, iar valoarea de ameliorare este dublul valorii capacității de transmitere deoarece valoarea de ameliorare a unui individ este de două ori deviația mediei descendenților de la medie (media valorilor de ameliorare ale descendenților).

Pornind de la principiile de bază ale programelor de ameliorare și de la premiza că pârghia principală pentru maximizarea progresului genetic în populațiile de taurine o reprezintă exercitarea presiunii de selecție prin masculi (peste 70%), interesează în mod special, posibilitatea predicției valorii de ameliorare a taurilor pentru caracterele productive (lapte, grăsime, proteină) și de exterior ale fiicelor descendente aflate în testare. Ca urmare, în implementarea modelului matematic bazat pe metodologia BLUP, se calculează valoarea de ameliorare pentru caracterele productive, caracterele de exterior și valoarea de ameliorare generală.

Metodele practice de realizare a împerecherilor nominalizate la taurine, folosesc pe lângă valoarea de ameliorare pentru principalele caractere productive și modul de transmitere în descendență a însușirilor morfologice, evaluate prin sistemul de descriere lineară. Scopul îl reprezintă obținerea de informații sistematizate și ierarhizate în ordinea valorică a taurilor pentru fiecare caracter sau însușire luată în considerare.

Ierarhizarea într-un top a celor mai buni tauri, permite selectarea pentru fiecare vacă a taurului cel mai indicat pentru corectarea și selecția celor mai bune exemplare.

În schema din figura se arată modul de derulare al programului SAP („Sire Advisory Program” utilizat de Asociația crescătorilor de taurine din Olanda) unde ierarhizarea taurilor și calculul punctajelor pentru fiecare vacă conduc la obținerea de informații care permit decizia optimă în stabilirea soluției de împerechere.

În figura 35 sunt marcate etapele în care se poate interveni utilizând computerele și programele de calcul[Ge98].

Specialiștii evaluatori (interesați financiar pentru vânzarea materialului seminal) execută lucrările de apreciere pentru mamele de tauri și pentru fiicele taurilor în testare după descendenți. Aprecierea se face pe cele două linii: caractere productive și caractere de exterior, ea poate fi efectuată la solicitarea fermierilor pentru toate vacile din fermă (eventuale candidate de mame de taur), iar datele sunt stocate în baza de date. În figura 35 se prezintă schematic un programul derulat în vederea nominalizării împerecherilor.

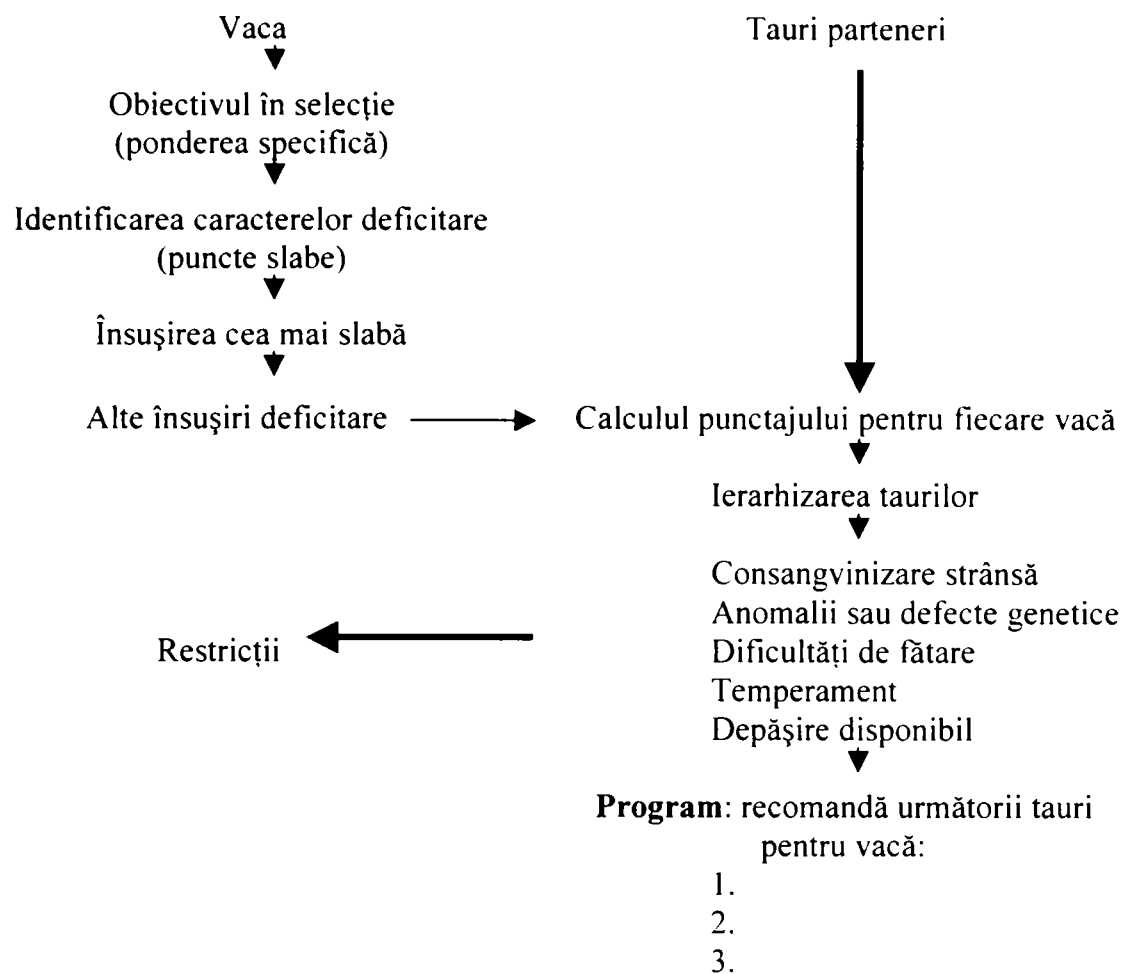


Figura 35. Programul derulat în vederea nominalizării împerecherilor

Datele necesare în vederea calculului valorii de ameliorare se reduc la următoarele:

1. Caractere productive:

- Cantitate lapte
- % grăsime
- %proteină

Conform procedurilor ce există în domeniul zootehnic, evaluarea acestora se face de către *controlorii* desemnați, evaluarea se face periodic pe baza unor proceduri standard pentru candidatele mame de taur și pentru fiicele descendente ale taurilor aflați în testare. Valorile măsurate sunt extrapolate pe întreaga perioadă de control, obținându-se valorile pentru caracterele menționate pentru perioada standard de lactație sau pentru întreaga perioadă de lactație. Aceste valori intervin în calculul valorii de ameliorare [Brudiu05].

2. Caractere de exterior:

Funcție de rasă (la noi în țară, în programul de ameliorare sunt prezente trei rase: Bălțată românească, Brună și Bălțată cu negru) *experții evaluatori*, apreciază conform procedurilor existente, după metoda descrierii lineare, însușirile morfologice pe caractere și grupe de caractere. În urma acestor aprecieri se obțin punctaje care la fel ca și în cazul caracterelor productive, intervin în calculul valorii de ameliorare[Brudiu04].

În scenariul aplicației intervin pe lângă *controlori* și *experții evaluatori*, *însămânțătorii*, rolul acestora este de a efectua însămânțarea artificială a vacilor și junincilor conform procedurilor existente, respectând nominalizările recomandate.

În condițiile existenței unui program unic de identificarea, înregistrare a bovinelor și mișcarea animalelor de fermă, în scenariu intervin **proprietarii** de taurine și **exploatațiile (fermele)** deținute de aceștia.

În figura 36. este prezentată Diagrama *USE CASE* cu actorii ce intervin în scenariul aplicației.

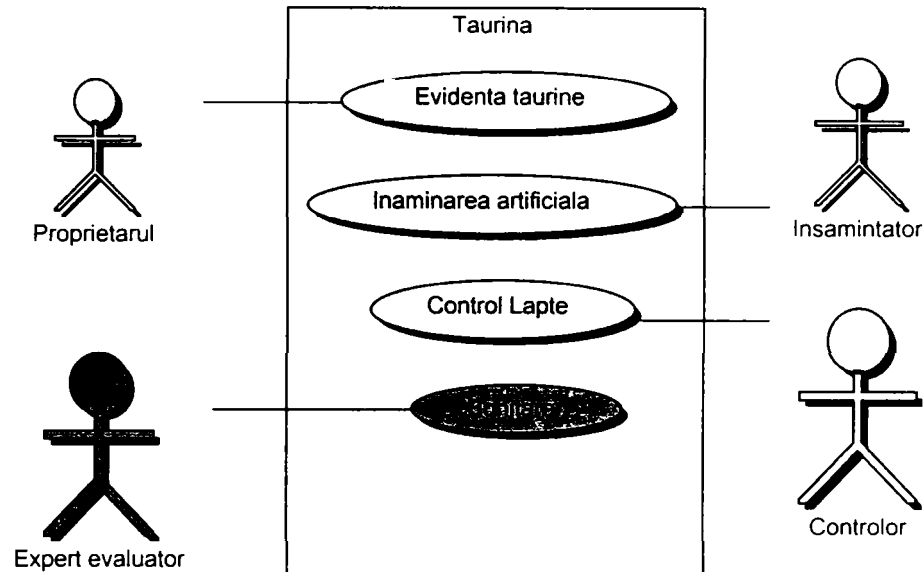


Figura 36. Actorii ce intervin in scenariul aplicației (UML –Diagrama USE CASE)

În aplicația „Ameliorare” se calculează valoarea de ameliorare pentru p tauri, care au în total n fiice aflate în m ferme.

Elementul fix luat în considerare îl reprezintă condițiile de mediu din fermă, elementul aleator îl reprezintă capacitatea de transmitere a caracterului pentru taurii aflați în testare.

În baza de date, fiicele au calculată producția de lapte (standard), precum și cantitatea de grăsime și proteină pentru prima lactație, de asemenea în urma bonității utilizând descrierea lineară a caracterelor de exterior în baza de date se regăsesc valorile punctajelor realizate pe caractere și a punctajelor parțiale pe grupe de caractere pentru trei rase de taurine (Bălțată românească, Bălțată cu negru românească și Brună).

În ecuațiile modelului mixt, matricea coeficienților $(m+p, m+p)$ se compune din patru matrici după cum urmează:

- Matricea $X'X(m,m)$ elementul $x'x_{i,i}$ conține numărul fiicelor din fiecare ferma i ;
- Matricea $X'Z(m,p)$ elementul $x'z_{i,j}$ conține numărul de fiice din ferma i ale taurului j ;
- Matricea $Z'Z(p,p)$ elementul $z'z_{i,j}$ conține numărul de fiice ale taurului i ;
- Matricea $Z'X(p,n)$ reprezintă inversa matricei $X'Z$.

Vectorul termenilor liberi ($m+p$) se compune din 2 vectori după cum urmează:

- Vectorul $X'Y$ (m) elementul $x'i$ conține suma performanțelor realizate de fiice în ferma i ;
- Vectorul $Z'X$ (p) elementul $z'i$ conține suma performanțelor realizate de fiicele taurului i .

Pentru construirea matricii coeficienților și a vectorului termenilor liberi, a fost construită o *clasa Matrice* a cărei membrii sunt:

- Proprietățile *Ferma* și *Tauri*, care în momentul când sunt apelate, interoghează baza de date și furnizează numărul de ferme în care se găsesc fiicele taurilor aflate în testare, respectiv numărul de tauri aflați în testare.
- Metodele `getMatriceXX()`, `getMatriceZZ()`, `getMatriceXZ()`, care în momentul când sunt apelate, interoghează baza de date și furnizează numărul de fiice din fiecare fermă, numărul de fiice ale fiecărui taur, respectiv numărul de fiice din fiecare fermă corespunzătoare fiecărui taur. Metoda `getMatriceInversa()` calculează inversa matricii XZ (Figura 36, 37, 38).
- Metodele `getVectorPX()` și `getVectorPZ()`, în momentul când sunt apelate furnizează suma caracterelor productive (cantitate lapte, grasime, proteină sau punctaje referitoare la caracterele de exterior) realizate pe fiecare fermă respectiv pe fiecare taur (Figura 39,40).
- Metodele `GetMatrice()` și `GetMatriceTermeniLiberi()`, construiesc matricile coeficienților și cea a termenilor liberi pentru a fi transmise ca parametri metodei `Gauss()` ce rezolvă sistemul de ecuații (Figura 41,42).

```
private float[,] getMatriceXX()
{
    float[,] matriceXX=new float[Ferme.Count,Ferme.Count];
    for(int i=0;i<Ferme.Count;i++)
    {
        matriceXX[i,i]=
mainModel.getNrFicedinFerma((long)(Ferme.ToArray()).GetValue(i));
    }
    return matriceXX;
}
```

Figura 36. Implementarea metodei `getMatriceXX()`

```
private float[,] getMatriceZZ()
{
    float[,] matriceZZ=new float[Tauri.Count,Tauri.Count];
    for(int i=0;i<Tauri.Count;i++)
    {
        matriceZZ[i,i]=mainModel.getNrFicedinTata((long)(Tauri.ToArray()).GetValue(i));
    }
    return matriceZZ;
}
```

Figura 37. Implementarea metodei `getMatriceZZ()`


```

private float[,] getMatriceXZ()
{
    float[,] matriceXZ=new float[Ferme.Count,Tauri.Count];
    for(int i=0;i<Ferme.Count;i++)
    {
        for(int j=0;j<Tauri.Count;j++)
        {
            matriceXZ[i,j]=mainModel.getNrFicedinFermadinTata((long)(Ferme.ToArray()).GetValue(i),(long)(Tauri.ToArray()).GetValue(j));
        }
    }
    return matriceXZ;
}

```

Figura 38. Implementarea metodei **getMatriceXZ()**

```

private float[] getVectorPX()
{
    float[] vectorPX=new float[Ferme.Count];
    for(int i=0;i<Ferme.Count;i++)
    {
        vectorPX[i]=mainModel.getSumaCPFiicedinFerma((long)(Ferme.ToArray()).GetValue(i));
    }
    return vectorPX;
}

```

Figura 39. Implementarea metodei **getVectorPX()**

```

private float[] getVectorPZ()
{
    float[] vectorPZ=new float[Tauri.Count];
    for(int i=0;i<Tauri.Count;i++)
    {
        vectorPZ[i]=mainModel.getSumaCPFiicedinTaur((long)(Tauri.ToArray()).GetValue(i));
    }
    return vectorPZ;
}

```

Figura 40 Implementarea metodei **getVectorPZ()**

```

public float[,] GetMatrice()
{
    float[,] matriceXX;
    float[,] matriceXZ;
    float[,] matriceZZ;
    float[,] matriceZX;
    matriceXX=getMatriceXX();
    matriceZZ=getMatriceZZ();
    matriceXZ=getMatriceXZ();
    matriceZX=getMatriceInversa(matriceXZ);
}

```

```

float[,] matriceCoeficienti= new float
[Ferme.Count+Tauri.Count,Ferme.Count+Tauri.Count] :
    # pun matricea XX
    for(int i=0;i<Ferme.Count;i++)
    {
        matriceCoeficienti[i,i]=matriceXX[i,i];
    }
    # pun matricea XZ
    for(int i=0;i<Ferme.Count;i++)
    {
        for(int j=Ferme.Count;j<Ferme.Count+Tauri.Count;j++)
        {
            matriceCoeficienti[i,j]=matriceXZ[i,j-Ferme.Count];
        }
    }
    # pun matricea ZX
    for(int i=Ferme.Count;i<Ferme.Count+Tauri.Count;i++)
    {
        for(int j=0;j<Ferme.Count;j++)
        {
            matriceCoeficienti[i,j]=matriceZX[i-Ferme.Count,j];
        }
    }
    # pun matricea ZZ
    for(int i=Ferme.Count;i<Ferme.Count+Tauri.Count;i++)
    {
        matriceCoeficienti[i,i]=matriceZZ[i-Ferme.Count,i-
Ferme.Count]+15;
    }
    return matriceCoeficienti;
}

```

Figura 41 Implementarea metodei **getMatrice()**

```

public float[] GetMatriceTermeniLiberi ()
{
    float[] vectorPX;
    float[] vectorPZ;
    vectorPX=getVectorPX();
    vectorPZ=getVectorPZ();
    float[] termeniLiberi =new float[Ferme.Count+Tauri.Count];
    # pun vectorul PX
    for(int i=0;i<Ferme.Count;i++)
    {
        termeniLiberi[i]=vectorPX[i];
    }
    # pun vectorul PZ
    for(int i=Ferme.Count;i<Ferme.Count+Tauri.Count;i++)
    {
        termeniLiberi[i]=vectorPZ[i-Ferme.Count];
    }
    return termeniLiberi;
}

```

Figura 42. Implementarea metodei **getMatriceTermeniLiberi()**

Interogarea bazei de date se face prin apelarea unor metode definite în cadrul claselor DAO (Data Acces Object). Aceste metode sunt la rândul lor apelate de proprietăți ale clasei MainModel, respectând astfel tehnologia MVC (Model- View-Controller).

Aplicația furnizează pentru fiecare taur în testare următoarele informații:

- datele de identificare ale taurului (nume, matricolă, data nașterii, rasa, matricolele părinților);
- numărul de fiice a căror caractere productive au fost luate în calcul;
- valoarea de ameliorare absolută și relativă pentru caractere productive: producția de lapte, cantitatea de grăsime în lapte și cantitatea de proteină din lapte, aprecierea caracterelor de exterior.

Conform acestei metode de obținere a informațiilor necesare construirii matricei coeficienților și a vectorului termenilor liberi, accesul la baza de date în vederea interogării se face de $(2+2m+2p+mxp)$ ori, ($m = \text{numărul de ferme}$, $p = \text{numărul de tauri}$), indiferent de numărul fiicelor pe care le au taurii aflați în testare.

Figura 43. prezintă Fisa taurului cu Informații referitoare la valoarea de ameliorare.

Ameliorare v0.1

File Mentenanța About

Adaugare Salvare Stergere

Lista Taurine

Ga50TM
 GE51TM
 GI52TM
 GO53TM
 Horia54TM
 Iancu55TM
 Ia56TM
 Ia57TM
 Ic58TM
 In59TM
 IO60TM
 IO61TM
 Io62TM
 IO63TM
 IR64TM
 Ivan65TM
 Le66TM
 Ma67TM
 Ma68TM
 MA69TM
 Ma70TM
 MI71TM
 NO72TM
 OY73TM
 Paul74TM

Date Generale Caracteristici productive Informatii Ameliorare

Cod Matricol Paul74TM
Nume Paul
Rasa Bruna
Exploatatie Cod E3
Buzias
Proprietar codului marcu
Marcu Aurel

Data nasterii 12/23/1998 12:00:00 A
Data intrării 12/23/1998 12:00:00 A
Conditii intrare nastere

Date sanataate

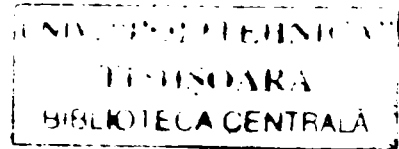
Mama Ed41TM
Tata Zeno87TM
Ameliorare TT

Valoare de Ameliorare lapte 305.5746 Fiice in testare 12
 grasime 9.832557
 proteina 16.32147

CAUTARE AVANSATA

statusBar

Figura 43. Modul de afișare a datelor referitoare la valoarea de ameliorare a taurilor. Aplicația „Ameliorare” C#.NET



3.7.6. Arhitectura software si patternurile arhitecturale folosite in aplicatie

Arhitectura software pe mai multe niveluri (patternul n-tier)

Termenul de *n-tier* se referă la construirea aplicațiilor pe baza unui model arhitectural bazat pe nivele suprapuse.

În arhitectura *n-tier*, aplicația se împarte în părți logice, fiecare parte dezvoltând o porțiune a aplicației (afișarea interfeței cu utilizatorul, accesul la date, etc.), numărul acestora determinând numărul de niveluri [Ma00, Ma02, Ch05]. Cel mai adesea se folosește un număr de trei niveluri, acestea sunt:

- **Nivelul de prezentare** – partea de interfața utilizator a aplicației
- **Logica de business** – în cele mai multe cazuri acest nivel conține codul care forțează regulile de business ale aplicației, dar și entitățile de business, cunoscute împreună sub numele de model de domeniu (domain model)
- **Accesul la date persistente** – codul care face persistarea și încărcarea datelor de pe un suport de stocare (sursa de date) fie acesta sistem de fișiere, soluții XML, baze de date relaționale sau obiectuale etc.

În figura 44, este prezentată arhitectura software 3-tier.

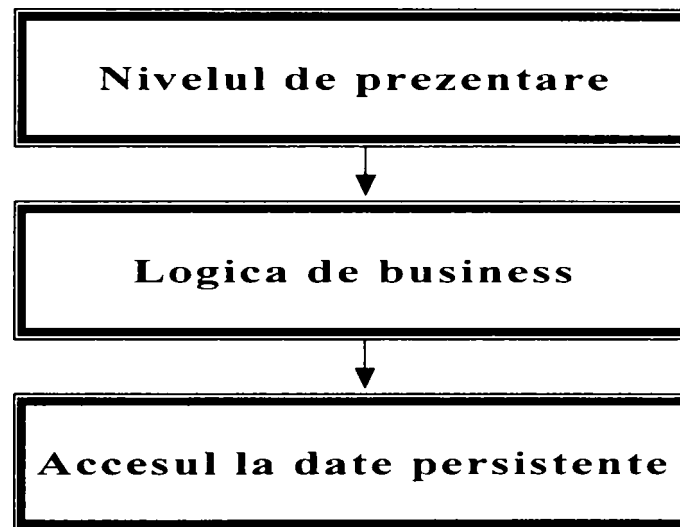


Figura 44. Model tipic n-tier

În aplicația „Ameliorare” s-a utilizat acest tip de arhitectură software datorită avantajelor ce le oferă:

1. în condițiile utilizării aplicației cu alt server de baze de date există posibilitatea modificării rapide în porțiunea de cod reprezentată de nivelul de acces la date persistente, în cazul de față în clasele DAO;

2. utilizarea aplicației pe o interfață Web solicită doar modificări în codul reprezentat de nivelul de prezentare;
3. utilizarea arhitecturii n-tier asigură o claritate bună a codului, ușor de testat asigurând o flexibilitate și o mentenanță sporită.

În figura 45, este prezentată structura pe nivele a aplicației „Ameliorare”

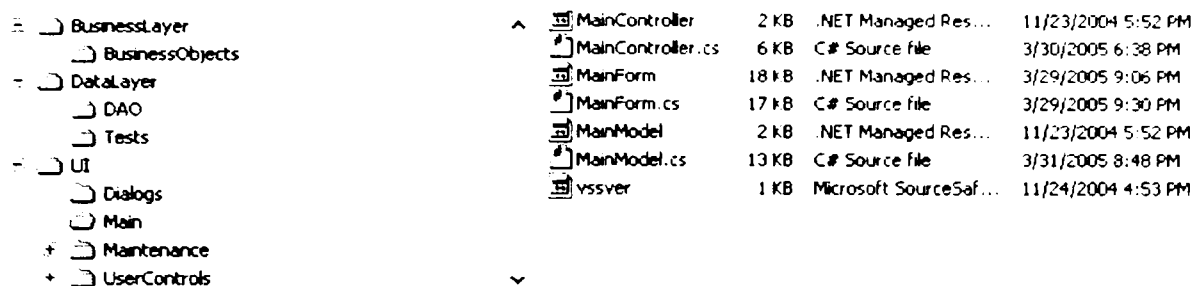


Figura 45 Structura pe nivele a aplicației „Ameliorare” (C#.NET)

Patternul arhitectural Model-View-Controller

Model-View-Controller (MVC) este un design pattern care separă eficient interfața cu utilizatorul de modelul de date în programarea orientată pe obiecte[Bu97, An01, Sc02, Pa03].

Această arhitectură este larg utilizată în programarea în limbajele Java, C++ sau Smalltalk, permițând reutilizarea codului sursă și reducând astfel durata de dezvoltare a aplicațiilor cu interfețe

Arhitectura model-view-controller se constituie din trei componente principale:

- **componenta Model**, reprezentată de structura logică de date a aplicației și clasele de nivel înalt asociate cu ea. Componenta Model nu conține informații despre interfața utilizator. Datele obținute prin componenta Model pot fi văzute de utilizator prin mai multe interfețe. Componenta Model răspunde cererilor făcute de componenta Controller prin afișarea modificărilor asupra datelor.
- **componenta View**, care este o colecție de clase reprezentând interfața cu utilizatorul (toate obiectele pe care utilizatorul le poate vedea pe ecran și cu care poate interacționa, cum ar fi butoane, casete de text, etc.) Dezvoltatorii de software pot implementa pentru aceleași date mai multe componente View, toate utilizând aceleași componente Model și Controller. Componentele View pot fi *WEB-FORMS*, *HTML*, *XML/XSLT*, *XHTML*, și *WML* sau pot fi *Windows forms* etc.
- **componenta Controller**, care reprezintă clasele ce realizează comunicarea între clasele din Model și cele din View. Componenta Controller răspunde la evenimentele și cererile pe care le face utilizatorul prin mouse sau tastatură.

În figura 46 este prezentat schematic modul de lucru al în arhitectura MVC.

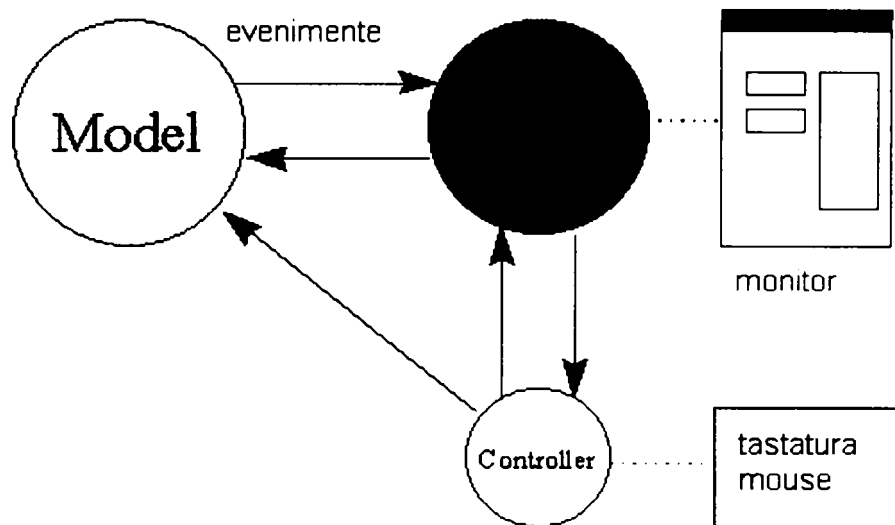


Figura 46. Modul de lucru în arhitectura MVC.

Avantajele utilizării arhitecturii MVC sunt:

1. Există posibilitatea dezvoltării a mai multor vederi pentru aceleași date
2. Decuplarea completă a claselor ce lucrează cu baza de date de colecția de clase ce reprezintă interfața cu utilizatorul asigură un grad ridicat de reutilizare a codului și de modularitate
3. Utilizând arhitectura MVC, aplicația este ușor extensibilă și scalabilă.

În aplicația „Ameliorare”, componenta Controller (**MainController**) prezentată în figura 48, inițializează și închide formul meniului principal prezentată în figura 47 și accesează baza de date prin MainModel în vederea încărcării cu date a controlerelor din apelate din acesta.[Gi02].

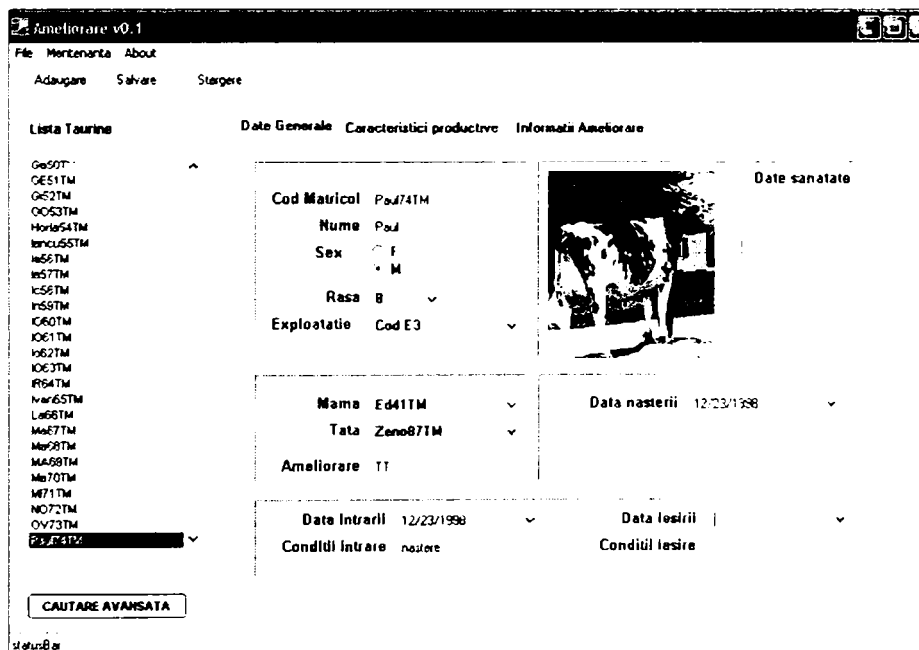


Figura 47. Formul principal la aplicației „Ameliorare” (C#.NET)

```

100         #region initializare si inchidere form
101             //
102             apelata la deschiderea aplicatiei
103             //
104         public void Init()
105             //
106             apelat la inchiderea aplicatiei
107             //
108         public void Exit()...
109     #endregion
110
111     //
112     s-a selectat o taurina din lista
113     //
114     //
115     public void SelectTaurina(object id)
116     public void DisplayBySex()
117     public bool IsTaur()...
118     //
119     public void Save()...
120     //
121     public void Adauga()...
122     //
123     public void Sterge()...
124     public void RefaceListBox()...
125     public void ShowProprietari()...
126     public void ShowExpertii()...
127     public void ShowControlori()...
128     public void ShowInsamintatori()...
129     public void ShowExplicatacii()...
130     public void ShowTaurine()...
131     //
132     )
133     //

```

Figura 48. Descrierea MainController. (C#.NET)

Componenta View (**MainForm**) cuprinde toată colecția de clase ce asigură interfața cu utilizatorul. În cadrul aplicației „Ameliorare” a fost dezvoltată o singură vedere (View1) ce asigură o interfațare pentru o aplicație desktop, existând posibilitatea dezvoltării altor vederi, de exemplu pentru o aplicație Web.

Componenta Model (**MainModel**) prezintă structura logică de date a aplicației și clasele de nivel înalt asociate cu ea (Figura 49).

```

namespace Ameliorare.UI.Main
{
    //summary
    Summary description for MainModel.
    //summary
    public class MainModel : Component
    {
        //summary
        Required designer variable.
        //summary
        private Container components = null;
        public Proprietari proprietarAles= new Proprietari();
        public Proprietari proprietar    = new Proprietari();
        public Expert expert              = new Expert();
        public Expert expertAles         = new Expert();
    }
}

```



```

public Controlor controlor = new Controlor();
public Controlor controlorAles = new Controlor();
public Insamintatori insamintatori = new Insamintatori();
public Insamintatori insamintatorAles = new Insamintatori();
public Exploatatia exploatatia = new Exploatatia();
public Exploatatia exploatatiaAleasa = new Exploatatia();
public Taurina taurinaCurenta = new Taurina();
private ICollection mameDeTaurineList =new ArrayList();
private ICollection tauritestareList =new ArrayList();
private ICollection taurineList = new ArrayList();
private ICollection cMTList = new ArrayList();
private ICollection exploatatiiList = new ArrayList();
private ICollection raseList = new ArrayList();
private ICollection iList =new ArrayList();
private ICollection cList =new ArrayList();
private ICollection eList =new ArrayList();
private ICollection grupeList =new ArrayList();
private ICollection caractereExtRasaList =new ArrayList();
private ICollection caractereExtGrupaList =new ArrayList();
public ICollection proprietariList =new ArrayList();
public ICollection expertiList =new ArrayList();
public ArrayList listaTauriCuFiiceInTestare=new ArrayList();
public ArrayList listaFermeCuFiiceInTestare=new ArrayList();
private ArrayList cE = new ArrayList();
public ArrayList proprietari =new ArrayList();
public ArrayList listaCuExperti = new ArrayList();
public ArrayList listaCuProprietari = new ArrayList();
public ArrayList listaCuControlori = new ArrayList();
public ArrayList listaCuInsamintatori = new ArrayList();
public ArrayList listaCuExploatatii = new ArrayList();
public ArrayList listaCuTaurine = new ArrayList();
public int numarFiiceDinFerma;
public int numarFiiceDinTata;
public int numarFiiceDinFermaDinTata;
private Rasa vrasa ;

```

Figura 49. Descrierea MainModel. (C#.NET)

Soluții pentru accesul la date persistente - Data Access Object

Problema majoră în cazul aplicațiilor orientate obiect o reprezintă accesul la datele persistente. Acestea pot fi memorate în baze de date relaționale, orientate obiect sau baze de date ce utilizează alte modele de stocare a datelor. Arhitectura aplicației trebuie să separe codul ce asigură accesul la date de partea de cod a aplicației ce controlează business-ul logic al aplicației. Design pattern-ul DAO (Data Access Object) a fost introdus ca standard în J2EE și este utilizat și pe alte platforme.[Su03] Prin utilizarea DAO pentru accesul la date persistente se obține într-un mod abstract și încapsulat conexia , accesul și obținerea de date din baza de date.

Un obiect DAO definește metode pentru serie de operații cu baza de date cum ar fi CRUD (*create, read, update, delete*) , metode *find* sau obiect de bussines după un anumit criteriu.

În figura 50, este prezentat clasa DAO TaurineDAO(). Pe lângă metodele **SaveTaurina**, **Delete**, **UpdateToDB** și **InsertToDB**, ce asigură operațiile CRUD, sunt definite metode de caracteristică aplicației cum ar fi **LoadFiiceDinTata()**, ce returnează într-o listă toate fiicele în testare ce au același tata.

```

Ameliorare.DataLayer.DAO.TaurineDAO
(
private SqlCommand deleteSqlCommand;
private SqlCommand insertSqlCommand;
private SqlCommand selectSqlCommand;
private SqlCommand updateSqlCommand;
***
private Container components = null;

public TaurineDAO(IContainer container):base(container)
private void InitializeComponent()
public TaurineDAO():base()
// Summary:
    Clean up any resources being used.
// Summary:
protected override void Dispose( bool disposing )

Component Designer generated code
public long FindTaurinaByPK(long taurinaID)
public long LoadNrFiiceDinFerma(long exploatareID)
public float LoadSumaCPLapteFiiceDinFerma(long exploatareID)
public float LoadSumaCPGrasimeFiiceDinFerma(long exploatareID)
public float LoadSumaCPProteinaFiiceDinFerma(long exploatareID)
public float LoadSumaCPLapteFiiceDinTaur(long taurinaID)
public float LoadSumaCPGrasimeFiiceDinTaur(long taurinaID)
public float LoadSumaCPProteinaFiiceDinTaur(long taurinaID)
public long LoadNrFiiceDinTata(long tataID)
public long LoadNrFiiceDinFermaDinTata(long exploatareID, long tataID)
public void LoadTaurinaByID(long taurinaID, Taurina t)
public void LoadAllTT(ArrayList taurineList)
public void LoadListaTaurinaID(ArrayList taurineList)
public void LoadListaFermeID(ArrayList fermeList)
public void LoadAllNameDeTaurine(ArrayList taurineList)
public void LoadAll(ArrayList taurineList)
public void LoadAllPentruCalculVA(ArrayList CHTList)
private void TransferFromReader(SqlDataReader reader, Taurina t)
public void SaveTaurina(Taurina t)
public void Delete(Taurina t)
public void Delete(long taurinaID)
private void UpdateToDB(Taurina t)
private long InsertIntoDB(Taurina t)
protected long ReadID(SqlConnection con)
)

```

```

Taurina t = new Taurina();
//Taurina t = null;
LoadTaurinaByID(taurinaID,t);
if(t.TaurinaID===-1)
    t=null;
return t;

```

Figura 50. Metoda DAO TaurineDAO(). (C#.NET)

Pentru toate metodele DAO au fost dezvoltate teste ce asigură corectitudinea operațiilor de acces la baza de date. În figura 51, este prezentată metoda **RasaDAOTests()**. În metoda dezvoltată pentru testare este utilizat **NUnit.Framework**. În figura 52 este prezentată structura directorului DAO și Test pentru toate clasele cu acces la baza de date dezvoltate.

```

using NUnit.Framework;
using Ameliorare.BusinessLayer.BusinessObjects;
using Ameliorare.DataLayer.DAO;

[TestFixture]
public class RasaDAOTests
{
    private RasaDAO rasdao = null;
    public long rasaID = -1;

    [Test]
    public void test1_Insert() { ... }
    [Test]
    public void test2_Update() { ... }
    [Test]
    public void test3_Delete() { ... }
    private string GetNewRandom() { ... }
}

```

Figura 51. Teste DAO RasaDAOTests() (C#.NET)

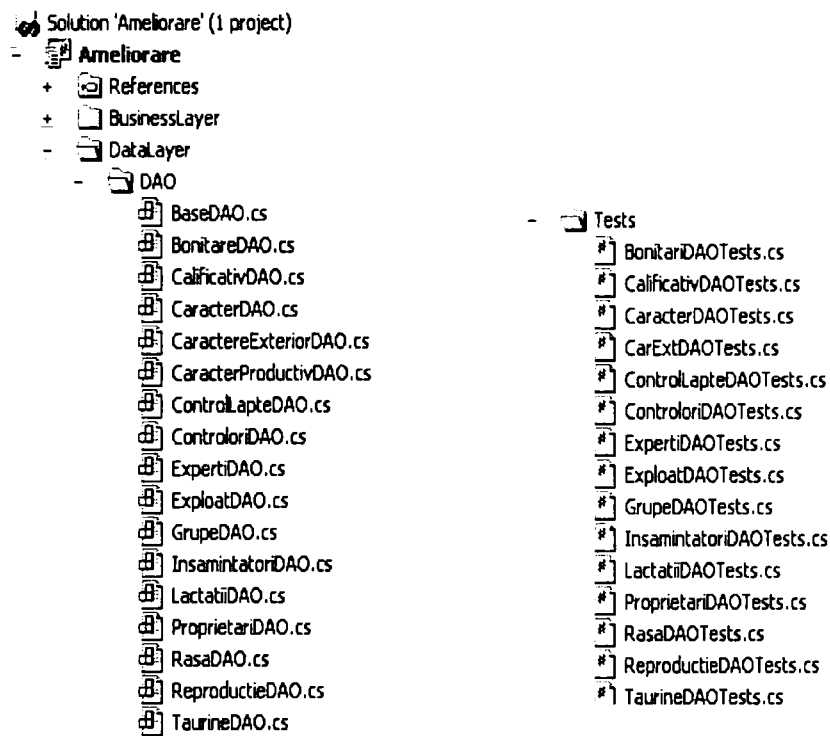


Figura 52. Structura directorului DAO și Tests

4. CONCLUZII, CONTRIBUȚII PERSONALE ȘI PERSPECTIVE

4.1. Concluzii

Cercetările efectuate asupra evaluării genetice a caracterelor productive ale vacilor de lapte și la implementarea aplicației au condus la următoarele **concluzii**:

1. În studiul și evaluarea genetică a caracterelor productive ale vacilor de lapte se folosesc modele matematice iar predicția valorii de ameliorare se poate face prin mai multe metode. În cercetarea efectuată, metodologia utilizată pentru predicția valorii de ameliorare este metodologia BLUP.
2. Pe baza modelelor matematice, statistica matematică ajută la aproximarea valorii de ameliorare reale, pe baza unui număr finit de descendenți. Aprecierea are un caracter probabilistic, în sensul că pe baza descendenților obținuți până la un moment dat, este posibil să se prevadă valoarea medie a viitorilor produși. Predicția valorii de ameliorare folosește pentru alegerea animalelor la reproducție.
3. Modelele clasice utilizate pentru predicția valorii de ameliorare prin metodologia BLUP sunt înlocuite de modele noi ce utilizează informații culese zilnic pentru toate caracterele productive. Aceste metode presupun existența unei infrastructuri ce asigură culegerea, stocarea informațiilor (milioane de înregistrări la nivel național) și aplicații ce asigură prelucrarea lor în timp real.
4. Modelul propus în lucrare ia în studiu factori ce caracterizează curba de lactație apropiind modelul propus de cel al testului zilei de control, fără necesitatea creării grupului de fiice contemporane. Rezultatele obținute în analiza modelului propus precum și cele obținute în predicția valorii de ameliorare utilizând acest model matematic, confirmă posibilitatea utilizării acestui model atât pentru predicția valorii de ameliorare cât și în studiul factorilor ce influențează caracterele productive la vacile de lapte.
5. Indiferent de modelul matematic utilizat sau metoda de predicție a valorii de ameliorare, în vederea cercetării factorilor ce influențează caracterele productive ale vacilor de lapte este necesară existența unei bazei de date cu informațiile fenotipice reprezentate de performanțele proprii ale indivizilor aflați în studiu precum și ale diferitelor tipuri de rude. Baza de date trebuie să conțină toate datele necesare aplicării modelelor cunoscute.
6. Funcție de modelul matematic utilizat, în aplicația ce lucrează cu baza de date este necesară implementarea algoritmilor ce asigură culegerea datelor și rezolvarea sistemului de ecuații ce furnizează valorile predictate pentru valoarea de ameliorare.
7. Crearea bazei de date cu informațiile necesare permite studierea modelelor existente și dezvoltarea de modele noi. Teoria ameliorării s-a dezvoltat mai rapid decât baza computațională pe care o solicita

aceasta, scopul cercetării prezentat în lucrare îl reprezintă crearea acestei baze de date, studierea modelelor existente și crearea de modele noi .

8. Întreținerea bazei de date și implementarea metodologiei de calcul s-a făcut într-o aplicație orientată obiect. Aceasta aplicație lucrează cu o baza de date relațională. Incompatibilitatea între modelul orientat obiect al datelor și cel relațional al datelor persistente generează „*impedance mismatch*”. Reducerea acestei incompatibilități se face utilizând unelte software, prin mapare.

4.2 Contribuțiile personale

În cazul unei cercetări interdisciplinare contribuțiile personale trebuie analizate atât separat, dar mai ales per ansamblu. Importanța și eficiența cercetării trebuie judecate sub aspectul modului în care acestea contribuie la dezvoltarea unui domeniu cu profundă latură practică.

Sistemul informatic realizat reprezintă informatica aplicată în domeniul ameliorării animalelor și presupune abordarea domeniului cu ajutorul matematicii și al informaticii.

Prin cercetările efectuate, **contribuțiile personale** sunt:

1. Propunerea unui model matematic în vederea studierii factorilor ce influențează caracterele productive la vacile de lapte.
2. Calculul valorii de ameliorare pentru caracterul productiv „lapte” prin modelul propus și compararea rezultatelor cu cele obținute utilizând modelul clasic.
3. Proiectarea unei structuri de Bază de date ce conține informațiile necesare cercetării în domeniu ameliorării taurinelor.
4. Realizarea bazei de date („Ameliorare”) și implementarea ei în sistemul Microsoft SQL Server 2000.
5. Implementarea unei aplicații orientate obiect ce lucrează cu baza de date „Ameliorare” care întreține baza de date și calculează valoarea de ameliorare pentru caracterele productive și de exterior la vacile de lapte prin metodologia BLUP. Limbajul de programare utilizat a fost C#.NET.

Analizând din acest punct de vedere contribuțiile enumerate, trebuie acordată importanța cuvenită laturii practice și valorificării cercetării, a legăturii între rezultatele cercetării și experiența practică, cunoscut sub numele de *extensie*.

În societățile consolidate, granițele universității se confundă cu granițele țării, pentru că universitatea are, în întreaga lume, o misiune socială. Societatea recentă cere imperios Universității să înceapă trecerea de la *cercetarea asupra...* și *de la cercetarea despre...* la *cercetarea cu ...*, respectiv *alături* de societate, spre binele comun, n această idee, cercetarea efectuată, asigură legătura între cercetare și practică.

4.3. Perspective

Teoria ameliorării s-a dezvoltat mai rapid decât baza computațională pe care o solicita aceasta. Deși teoretic fundamentată, folosirea în practică a unor modele complexe în domeniul ameliorării este întârziată din lipsa unei infrastructuri reprezentate de un sistem informațional la nivel național pentru culegerea și stocarea informațiilor din domeniul ameliorării.

Realizarea bazei de date ce cuprinde informații în concordanță cu legislația privind marcarea, identificarea, înregistrarea și mișcarea animalelor de fermă stabilite prin directive și reglementări ale organelor legislative românești și ale Uniunii Europene, permite dezvoltarea de aplicații în domeniu ameliorării și alinierea acestora cu cele existente pe plan mondial.

Cercetarea interdisciplinară (statistică, informatică, zootehnie) prezentă în lucrarea de față are ca rezultat cele prezentate în capitolul 4.2, oferă spre valorificare baza de date și o aplicație capabilă să întrețină această bază de date.

1. Caractere de carne ale taurinelor.

- Caractere apreciate pe animal viu:
 - greutatea corporală;
 - dimensiuni corporale;
 - viteza absolută de creștere;
 - capacitatea de valorificare a hranei;
 - conformația și calitatea comercială a animalelor.
- Caractere apreciate pe animalul sacrificat:
 - randamentul la tăiere;
 - carcasa (greutate, dimensiune, structură)
 - carnea (însușiri fizice și organoleptice, compoziția chimică, însușiri tehnologice).
 - gradul de îngrășare

2. Caracterele producției de lapte.

- Caractere cantitative:
 - producția de lapte pe lactația normală
 - producția de lapte pe lactația totală
 - producția de lapte în primele 100 de zile de lactație
 - producția de lapte pe viața productivă
 - producția medie a cirezii
- Caractere calitative:
 - conținutul în proteine
 - calitatea cazeinei din lapte
 - s₁ cazeina
 - s₂ cazeina
 - cazeina
 - k cazeina
- Uniformitatea lactației
- Viteza de muls
- Simetria funcțională (pentru mulsul mecanic)

3. Caractere reproductive

- Caractere generale de reproducție
 - natalitatea
 - fecunditatea
 - rata non-return
 - service-periode
 - indexul de însămânțare
 - calving-intreval
- Caractere reproductive specifice femelelor
 - apariția manifestării căldurilor
 - apariția maturității sexuale
 - prolificitatea
 - comportamentul la fătare
- Caractere reproductive specifice masculilor
 - precocitatea
 - longevitatea
 - instinctul genezic
 - producția și calitatea spermatică
 - capacitatea fecundantă

- Caractere speciale de reproducție (au influență directă asupra biotehnicilor și biotehnologiilor de reproducție)
4. Caractere de exterior ale taurinelor
 - dezvoltarea corporală – talia, perimetrul toracic, lungimea trunchiului
 - adâncimea toracică
 - unghiul crupei
 - lărgimea crupei
 - membre posterioare, ongoane, chișițe
 5. Rezistența la îmbolnăvire
 - rezistența la mamite
 6. Culoarea și particularități de culoare
 7. Producția energetică

Ameliorarea raselor de taurine

Rasele de taurine care fac obiectul programului de ameliorare sunt **Bălțată românească** și **Brună**, cu însușiri de producție mixtă – pentru carne-lapte și lapte-carne și **Bălțată cu negru românească** – specializată pentru producția de lapte. În funcție de stadiul actual al ameliorării și de obiectivele urmărite în selecție, aceste trei rase zonate de principiu în țara noastră se caracterizează, în linii mari, după cum urmează :

Rasa Bălțată românească – de tip Simmental, cu însușiri de producție mixtă,



– este și re rezină principală rasă de taurine a țării, atât prin ponderea numerică, precum și prin însușirile bune pentru ambele producții.

Taurinele din această rasă se preconizează ca, în perspectivă, să fie orientate mai bine spre producția de carne, păstrând însă aptitudinile generale ale tipului productive mixt. Ele rămân principalul furnizor de carne de taurine al țării – și împreună cu celelalte r

participă la acoperirea cerințelor de lapte și produse lactate.

Caracteristicile incluse în sistemul de descriere lineară a exteriorului la rasa Bălțată românească – Fleckvieh - Simmental și valorile medii ale heritabilității acestora (după Varderwestner, 1995 – Germania)

	INSUȘIRI	Coefficient de heritabilitate (h^2)
	DEZVOLTARE CORPORALĂ	0,30
1	Înălțimea la crupă	0,30
2	Lungimea crupei	-
3	Lărgimea crupei (la șolduri)	0,21
4	Adâncimea trunchiului	0,28
5	Unghiul (înclinarea) crupei	0,21
	MUSCULATURĂ	
6	Îmbrăcarea musculară	0,23
	FUNDAMENT (MEMBRE)	0,17
7	Unghiul jaretului	
8	Aspectul (consistența) jaretului	0,17
9	Chișițele	0,24
10	Ongloanele	0,12
	UGER	0,24
11	Extinderea anterioară a ugerului	0,21
12	Înălțimea (prinderea) posterioară a ugerului	0,23
13	Extinderea posterioară a ugerului	0,23
14	Ligamentul suspensor	0,28
15	Adâncimea ugerului	-
16	Poziția (plasarea și orientarea) mameloanelor	0,30
17	Lungimea mameloanelor anterioare	0,23
18	Grosimea mameloanelor	-

Rasa Brună – aparține tipului de producție mixtă pentru lapte – carne, la fel ca în majoritatea țărilor europene care o cresc, și la noi în țară însușirile prioritare sunt în direcția producției de lapte.

Taurinele din această rasă sunt destinate exploatații pentru producția de lapte în sistem semiintensiv, în ferme și gospodării din zonele subcarpatice și de altitudine, precum și în sistem intensive, în ferme cu stabulație liberă.



Caracteristicile incluse în sistemul de descriere lineară a exteriorului la rasa Brună - Brauviech și valorile medii ale heritabilității acestora (după Kanpp, 1993 – Austria)

	INSUȘIRI	Coeficient de heritabilitate (h^2)
	DEZVOLTARE (FORMAT) CORPORALĂ	0,33
1	Înălțimea la greabăn	0,39
2	Adâncimea trunchiului	0,26
3	Linia superioară a trunchiului	-
4	Lărgimea crupei (la șolduri)	0,34
5	Unghiul (înclinarea) crupei	0,21
	MUSCULATURĂ	
6	Îmbrăcarea musculară	0,46
	FUNDAMENT (MEMBRE)	0,24
7	Unghiul jaretului	0,16
8	Aspectul (consistența) jaretului	0,36
9	Chișițele	0,24
10	Ongloanele	0,18
	UGER	0,21
11	Extinderea anterioară a ugerului	0,50
12	Lărgimea posterioară a ugerului	-
13	Înălțimea posterioară a ugerului	0,23
14	Ligamentul suspensor	0,41
15	Adâncimea ugerului	0,29
16	Poziția (plasarea și orientarea) mameloanelor	0,22
17	Lungimea mameloanelor anterioare	0,27

Rasa Bălțată cu negru românească – de tip Holstein – Friză, specializată în direcția producției de lapte, aparține suprarasei Holstein-Friză, recunoscută pe plan mondial ca având cel mai înalt potențial genetic și cele mai ridicate performanțe în producția de lapte. În țara noastră a fost omologată ca rasă în anul 1987, fiind destinată exploatarii intensive, în ferme cu stabulație liberă.



Caracteristicile incluse în sistemul de descriere lineară a exteriorului la rasa Bălțată cu negru românească – Holstein – Friză și valorile medii ale heritabilității acestora [A100]

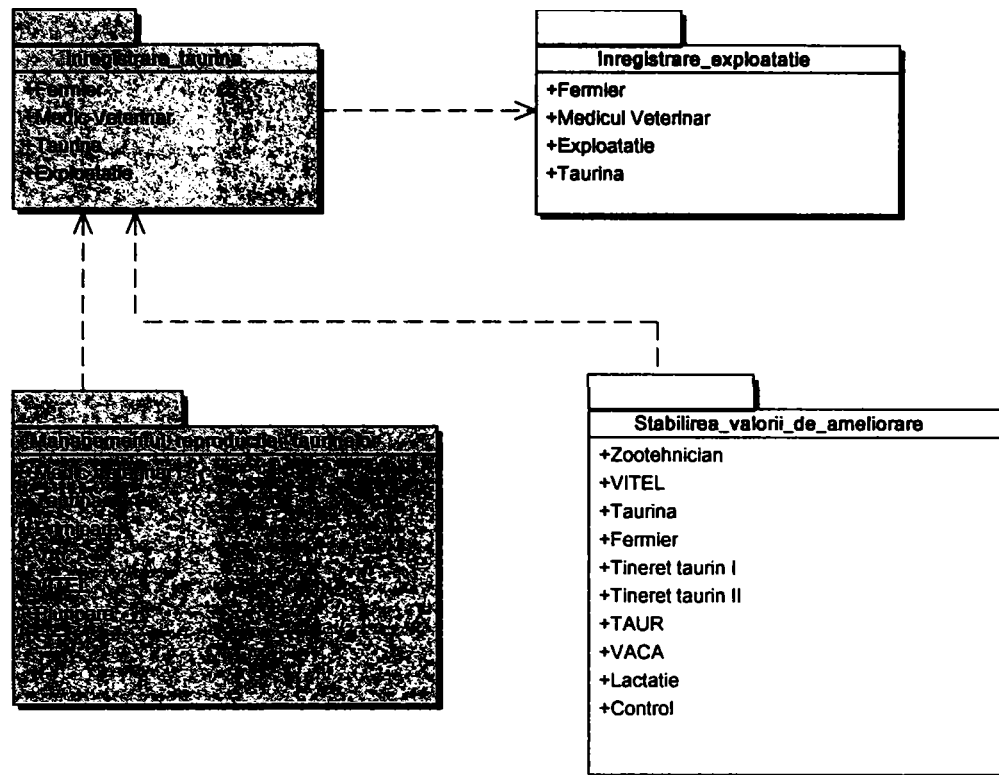
	INSUȘIRI	Coefficient de heritabilitate (h^2)
	TIP MORFOPRODUCTIV	-
1	Caractere specifice producției de lapte	-
	DEZVOLTARE CORPORALĂ	-
2	Înălțimea la crupă	-0,355
3	Adâncimea trunchiului	0,339
4	Lărgimea pieptului (capacitate toracică)	-
5	Lărgimea crupei (la ischii)	0,255
6	Unghiul (înclinarea) crupei	0,273
	FUNDAMENT (MEMBRE)	
7	Unghiul jaretului	-
8	Aspectul (consistența) jaretului	-
9	Ongloanele	0,133
10	Aplombul membrelor posterioare	0,169
	UGER	
11	Extinderea anterioară a ugerului	0,190
12	Înălțimea posterioară a ugerului	-
13	Adâncimea ugerului	0,270
14	Ligamentul suspensor	0,150
15	Poziția (plasarea și orientarea) mameloanelor	0,230
16	Lungimea mameloanelor anterioare	0,230

Parametri tehnici ai programului de ameliorare

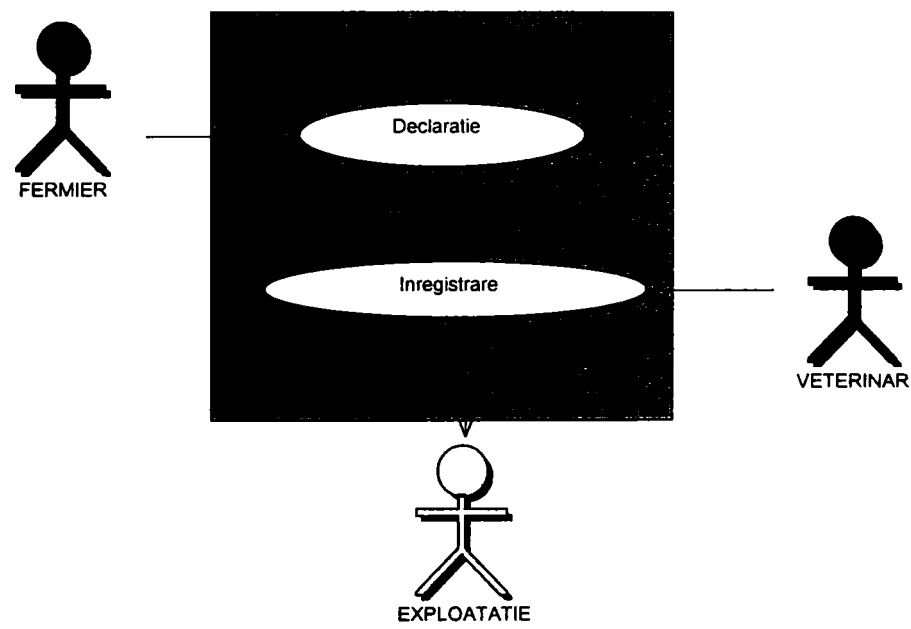
Specificare	UM	Bălțată românească	Brună	Bălțată cu negru românească	
		Rasă mixtă pentru carne - lapte	Rasă mixtă pentru lapte-carne	Rasă specializată pentru lapte	
1. Obiective					
a. Ponderea caracterelor în valoarea de ameliorare globală					
- Lapte	%	60	70	90	
- Carne	%	35	25	-	
- Fitness	%	5	5	10	
b. Producție medie de lapte pe lactație normală la echivalent maturitate					
- Lapte	Kg	5000	4800	6000	
- Grăsime	%	3,90	3,95	4,00	
	Kg	195	190	240	
- Proteină	%	3,30	3,35	3,40	
	kg	170	160	204	
c. Greutate corporală		kg	650 - 700	550 - 600	570 - 650
d. Înălțime	- la greabăn	cm	-	134 - 136	-
	- la crupă	cm	138-140	-	140-142
2. Vaci și vițele însămânțate artificial		cap	300.000	180.000	275.000
		%	46	39	45
3. Vaci în control oficial de producție		cap	80.000	60.000	85.000
		%	12,5	13,0	14,0
4. Tăurași din împerecheri nominalizate introduși în testare după performanțelor proprii		cap	60	40	52
5. Tauri introduși în testare după descendenți		cap	55	30	50
- proprii		cap	40	25	42
- import		cap	15	5	8
6. Capacitatea de testare lapte		cap	44.000	24.000	40.000
		%	55	40	47
7. Tauri în testare după descendenți pentru producția de carne		cap	55	30	-
- Capacitate de testare carne		cap	825	450	-
		cap	10	5	-

- Tauri eliminați după testare carne	cap	10	5	-
8. Tauri eliminați după testare din descendenți lapte	cap	45	25	50
	%	20 80	16 75	100
9. Tauri reușiți la testare – acceptați la IA	cap	9	6	7
	%	16,36	20	14
10. Necesari vaci mame de tauri	cap	360	260	390
11. Vaci candidate mame de taur	cap	Min 720	Min 520	Min 780
12. Tauri tați de tauri folosiți la împerecheri nominalizate	cap	10-12	8-10	10-12
13. Total m.s.c necesar a. de la tăurași în testare, total	doze	600.000	360.000	550.000
	doze	220.000	120.000	212.000
	%	36,6	33,3	38,5
- în populația activă (800 IA1 x 2 doze x nr tăurași)	doze	88.000	48.000	80.000
	%	14,6	13,3	14,5
- în restul populației (1200 IA1 x 2 doze x nr tăurași)	doze	132.000	72.000	132.000
	%	22,0	20,0	24,0
b. tauri în exploatare	doze	350.000	222.000	305.000
	%	58,4	61,7	55,5
	doze	30.000	18.000	33.000
	%	5,0	5,0	6,0
14. Tauri testați în exploatare	cap	27	18	21
15. Rata anuală de înlocuire	%	33,3	33,3	33,3
16. M.S.C în medie pe an și taur în exploatare	doze	13.000	12.300	14.500

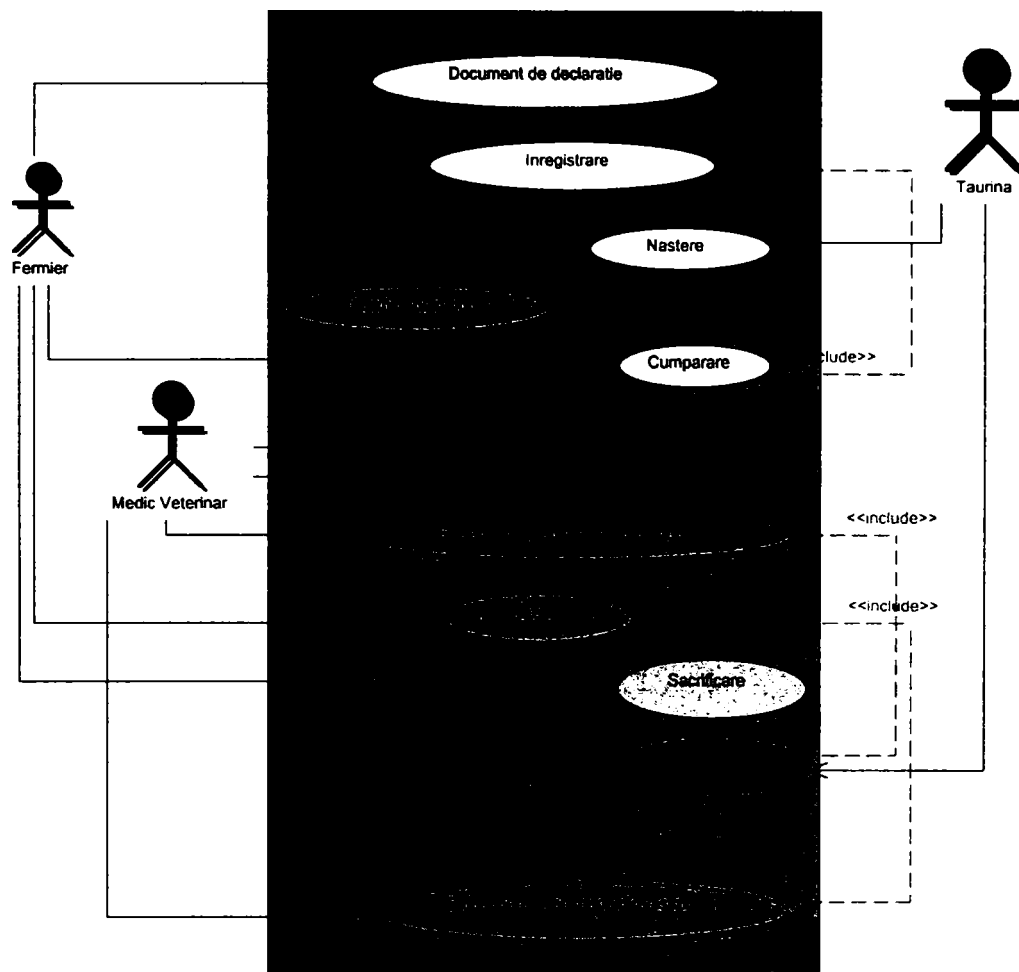
Arhitectura sistemului - ierarhie de pachete



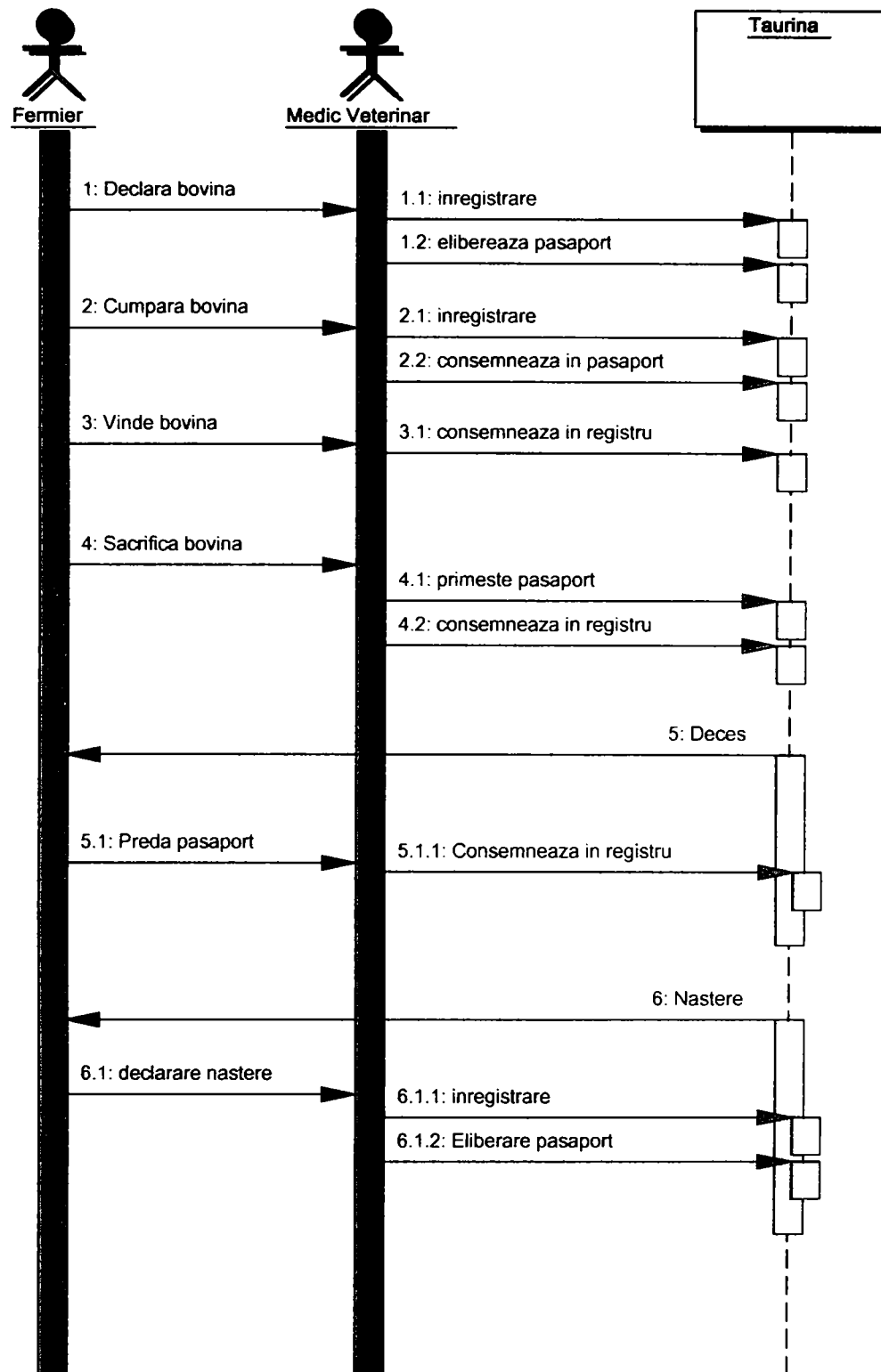
Înregistrarea exploatației - diagrama cazurilor de utilizare



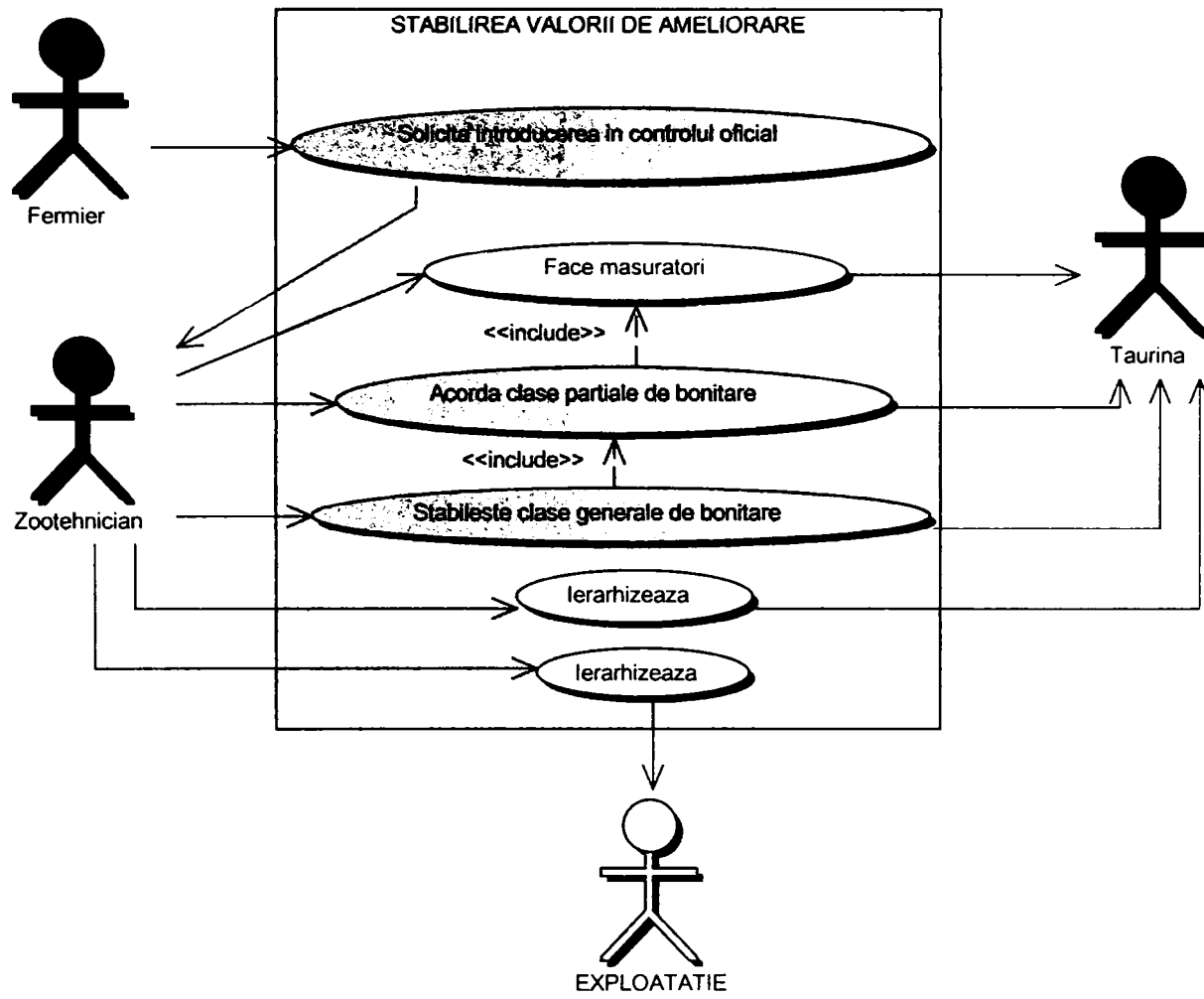
Înregistrarea bovinei - diagrama cazurilor de utilizare



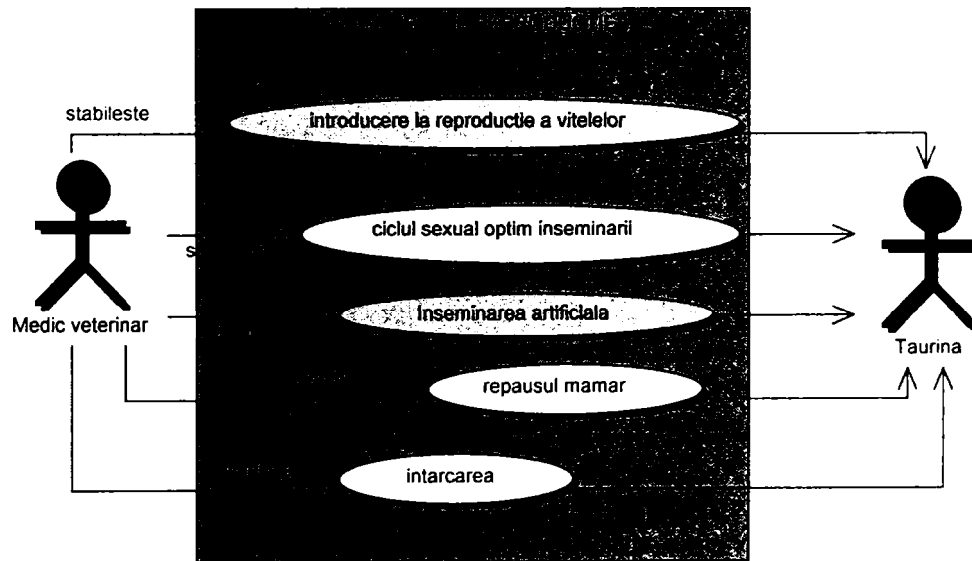
Înregistrarea bovinei - diagrama de secvență



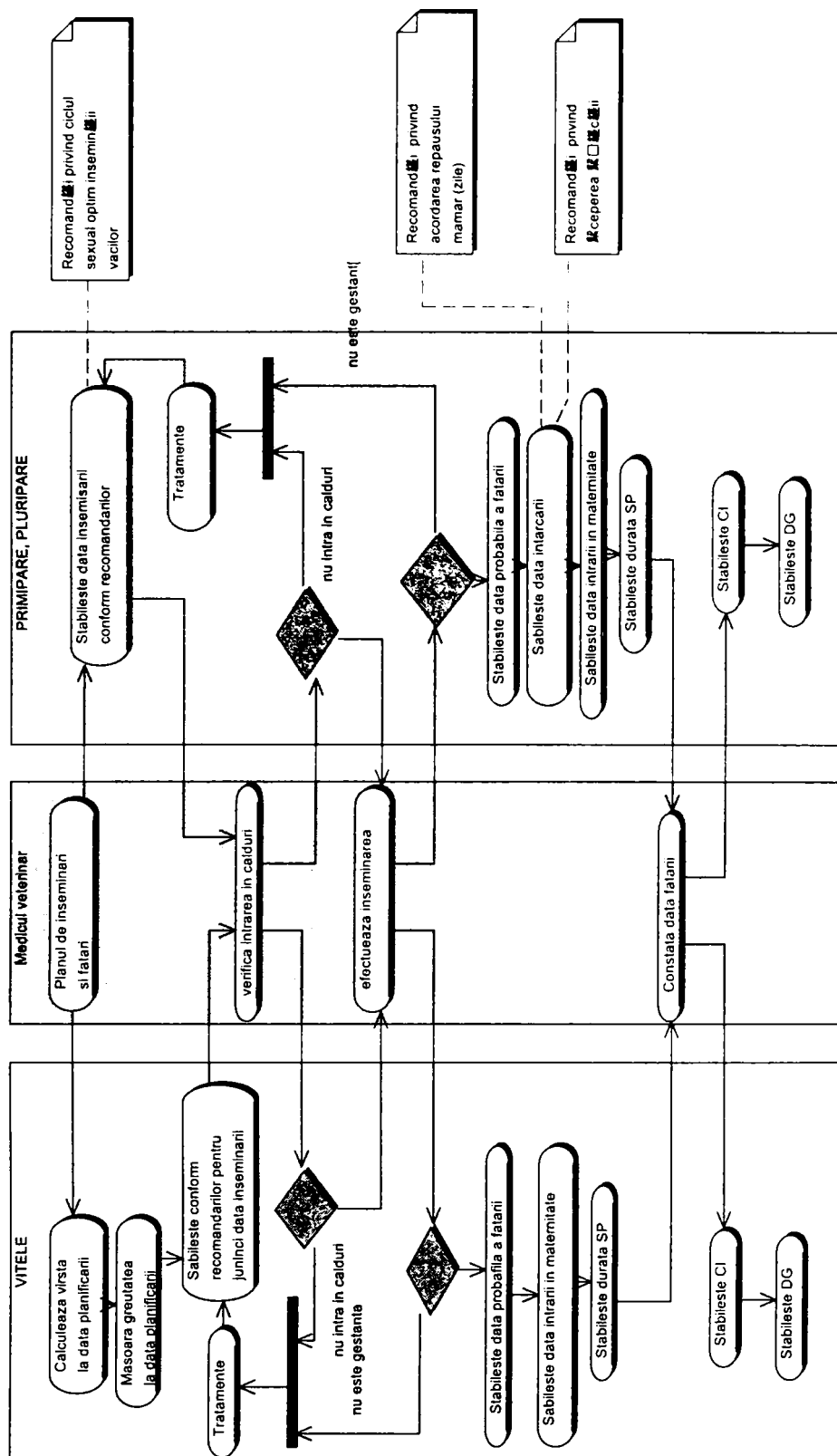
Stabilirea valorii de ameliorare a taurinelor - diagrama cazurilor de utilizare



Managementul reproduçiei- diagrama cazurilor de utilizare



Managementul reproduției - Diagrama de activități



Bibliografie

- 1.[Al88] Alexoiu A., Roșca L. Ghid practic de selecție și dirijare a împerecherilor în fermele de taurine. Edit Ceres, București. 1988.
- 2.[Al00] Alexoiu Aurelian și colab- Aprecierea conformației corporale la taurine, Editura Ceres, 2000.
- 3.[Ali87] Ali, T. E., L. R. Schaeffer. Accounting for covariances among test-day milk yield in dairy cows. Can. J. Anim. Sci. 67:637–644, 1987.
- 4.[Am03] Ambler Scott W. - The Fundamentals of Mapping Objects to Relational Databases, 2003, [http:// www.agiledata.org](http://www.agiledata.org)
- 5.[An01] Angel Jonathan - Tutorial - Middleware Network Magazine, 2001, <http://www.networkmagazine.com/article/NMG20000425S0003>
- 6.[At95] Atkinson M., Bancilhon F, DeWitt D., Dittrich K., Maier D., Zdonik S. The Object-Oriented System Manifesto. 1995, [http://www..cs.cmu.edu/People/clamen/OODBMS/Manifesto/Manifesto.html](http://www.cs.cmu.edu/People/clamen/OODBMS/Manifesto/Manifesto.html)
- 7.[Ba96] Barry D. K.-The Object Database Handbook : How to Select, Implement, and Use Object-Oriented Databases, John Wiley & Sons, 1996.
- 8.[Be97] Corneliu Berbente, Sorin Mitran, Silviu Zancu – Metode numerice, Editura Tehnica, 1997.
- 9.[Be99] Rex Bernardo- Marker-Assisted Best Linear Unbiased Prediction of Single-Cross Performance, Published in Crop Sci. 39:1277–1282, 1999.
10. [Bet97] Greg L. Bethard - A Microcomputer Simulation to Evaluate Management Strategies For Rearing Dairy Replacements, 1997. <http://www.scholar.lib.vt.edu/theses/available/etd-831102339731121/unrestricted/ETD.PDF>
11. [Bi00] Piter Bijmaa, John A. Woolliamsb - Prediction of Rates of Inbreeding in Populations Selected on Best Linear Unbiased Prediction of Breeding Value, Genetics, Vol. 156, 361-373, September 2000.
12. [Bl93] Andrew P. Black / Object Identity , A Position Paper for I-WOOS'93, - www.cs.pdx.edu/~black/publications/ObjectIdentity.pdf
13. [Bo87] Boch, G., - Object Oriented Analysis and Design, Benjamin-Cummings, Redwood City. CA, 1987.
14. [Bru03] Brudiu Ileana, Huțu Ioan - Utilizarea limbajului de modelare unificat (UML) în managementul reproducției taurinelor, Lucrări științifice Zootehnie și Biotehnologii, vol XXXVI, 2003.
15. [Bruce02] Kim B. Bruce - Foundations of Object-Oriented Languages Types and Semantics, MIT Press Cambridge, MA, USA, 2002.

16. [Brudiu02] Brudiu Ileana, I. Huțu, E.Sas- A software program to calculate relationship and inbreeding coefficients with tabular method . *Lucrări științifice medicină veterinară* vol. XXXV, 2002.
17. [Brudiu03] Brudiu Ileana - Identificarea și înregistrarea bovinelor. Stabilirea cerințelor și analiza cu UML , *Lucrări științifice Medicină Veterinară*, vol. XXXVI, pag. 606,2003.
18. [Brudiu04] Brudiu Ileana - Software privind aprecierea caracterelor de exterior la taurine prin metoda descrierii lineare , *Lucrări științifice Medicină Veterinară*, vol. XXXVII, 2004.
19. [Brudiu05] Brudiu Ileana - The prediction of the cattle breeding value through the *BLUP method* – implementation in object oriented software. *Lucrări științifice Medicină Veterinară*, vol. XXXVIII, 2005.
20. [Bu97] Burbeck Steve - Applications Programming in Smalltalk-80(TM):How to use Model-View-Controller (MVC), 1997. <http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html>
21. [C#04] Unknown - C# Standard, Language Specification, Produced by ECMA TC39/TG2, 2004.
22. [Ca00] M. P. L. Calus, R. F. Veerkamp Environmental sensitivity of genetic merit for milk, fat and protein yield estimated by a random regression model, *J Anim. Sci.* Vol. 81, Suppl. 1/*J. Dairy Sci.* Vol. 86, 2000.
23. [Ca02] J. Carneiro, E.J. Pollak, R.L. Quaas, R.W. Blake – An Autoregressive Repetability Animal model for Test-Day Records in Multiple Lactation, *Journal. Dairy Science*, Vol. 85, No.8, 2040:2045, 2002.
24. [Cache05] Unknown - Cache Post-Relational Databases – Portal Intersystem- <http://www.intersystems.com/cache/>,2005.
25. [Ch05] Chartier Robert - Application Architecture: An N-Tier Approach, <http://www.15seconds.com/issue/011023.htm>, 2005
26. [Co00] Constantine, L., *Fundamentals of Object Oriented Design in UML*, Addison-Wesley, Reading M.A., 2000.
27. [Co70] Codd E. F. - A Relational Model of Data for Large Shared Data Banks, *Communications of the ACM*, Vol. 13, No. 6, June 1970, pp. 377-387.
28. [Co90] Codd, E.E., *The Relational Model for Database Management*, Addison-Wesley, Reading M.A., 1990.
29. [Cr05] Knud Cristensen - An Applet for general matrix handling and calculating relationship and inbreeding, http://www.kursus.kvl.dk/shares/vetgen/_Popgen/genetik/applets/tabularapp.htm
30. [Cr99] R. Cristina J. Cosoroabă, **Ileana Brudiu** - Simularea evoluției rezistenței la antihelminticele benzimidazolice a trichostrongilidelor (Model matematic), *Rev. Rom. Med. Vet.* vol. 9 nr. 2, 1999, p. 155 – 162.1999.

31. [Da00] G. S. Datta, P. Lahiri - A unified measure of uncertainty of estimated best linear unbiased predictors in small area estimation problems, *Statistica Sinica* 10, 613-627, 2000.
32. [Da05] David M. Lane - Online Statistics: A Multimedia Course of Study, http://psych.rice.edu/online_stat/, 2005.
33. [Da95] Date, C.J. , An Introduction to Database System, Addison-Wesley, reading, MA, 1995.
34. [Da97] Davis, Judy. "Extended Relational DBMSs: The Technology. Part 1," *DBMS Online*, June 1997. <http://www.dbmsmag.com/9706d13.html>
35. [DataFit01] Unknown - DataFit version 7.1.44, <http://www.curvefitting.com>
36. [Dr03] T. Druet, F. Jaffrezic, D. Boichard, V. Ducrocq – Modeling Lactation Curves and Estimation of genetic Parameters for First Lactation Test-Day Records of French Holstein Cows, *Journal. Dairy Science*. Vol. 86, No.7, 86:2480-2490, 2003.
37. [Dr79] Drăgănescu C., *Ameliorarea animalelor* . Ed. Ceres, București, 1979
38. [Druet05] Tom Druet, Florence Jaffrézic ,Vincent Ducrocq - Estimation of genetic parameters for test day records of dairy traits in the first three lactations, *Genet. Sel. Evol.* 37 (2005) 257-271, 2005.
39. [Fo97] Marin Fotache - Baze de date relaționale. Organizare, interogare și normalizare. *Junimea Iasi* 1997
40. [Fu97] Fussell Mark L, - Foundation of Object Relational mapping , 1997. <http://www.chimu.com/publications/objectRelational/>
41. [Ge04] Gehrtland Justin - NHibernate, 2004, <http://www.theserverside.net/articles/showarticle.tss?id=NHibernate>
42. [Ge97] Georgescu Gh și colab, Studiu corelațiilor între principalele caractere ale producției de lapte. *Lucr. St., seria D. Zootehnie*, vol XXXII-XXXIV, USAMV București, 1997.
43. [Ge98] Georgescu Gh – coordonator, *Tratat de creșterea bovinelor – Ameliorare*, Edit. Ceres, București, 1998.
44. [Gi02] Gielens Paul - Model View Controller written in C# using the .NET Framework, 2002 http://www.codeproject.com/csharp/model_view_controller.asp
45. [Gr96] Grosu H. Și colab , *Predicția valorii de ameliorare la taurinele de lapte*. Comunicare, ICOCB Balotești, 1996.
46. [Gr97] Grosu H., Lungu S., Kremer V.D., *Modele liniare utilizate în ameliorarea genetică a animalelor*. –Edit. Coral Sanivet, București, 1997
47. [Gr98] Grimes Seth, *Modeling Object/Relational Databases*, *DBMS*, Aprilie 1998, <http://www.dbmsmag.com/9712d15.html>
48. [Ha91] Hanson Eric N, Tina M. Harvey - Experiences in Database System Implementation Using a Persistent Programming Language Pages: 314 –

328, 1991, [http:// www.cs.ubc.ca/local/reading/proceedings/ spe91-95/spe/vol23/issue12/spe868.pdf](http://www.cs.ubc.ca/local/reading/proceedings/spe91-95/spe/vol23/issue12/spe868.pdf)

49. [He75] Henderson C.R - Best linear unbiased estimation and prediction under a selection model. *Biometrics* 31(2):423-47., 1975
50. [He77] Henderson C.R - Best linear unbiased prediction of breeding values not in the model for records. *J. Dairy Sci.* 60:783-87. , 1977
51. [He84] Henderson C.R - *Applications of Linear Models in Animal Breeding*. Guelph, Ontario: University of Guelph. , 1984
52. [Hen75] Henderson C.R - A rapid method for computing the inverse of a relationship matrix. *J. Dairy Sci.* 58:1727-30.,1975
53. [Hu00] Huțu I. , E. Sas, **Brudiu Ileana**, Iulian Boboloacă - Managementul computerizat al activității de reproducție, *Lucr. Șt. Med. Vet. Timișoara*, vol.XXXIII, 2000
54. [Io04] Ionescu Felicia – Baze de date relaționale și aplicații. Editura tehnică, București, 2004.
55. [Ja97]J. Jamrozik, L.R. Schaeffer – Estimates of Genetic Parameters for a Test day Model with Random Regression For Yield Traits of first Lactation Holsteins. *Journal. Dairy Science*, Vol. 80, No. 4, 1997.
56. [Ke97] Keller Wolfgang – Mapping Objects to Table. A Pattern Language Proceedings EuroPLoP, 1997, <http://www.objectarchitects.de/>
57. [Lu95] Tom Luers –Bazele Oracle, Ed Teora, 1995.
58. [Ma00] Marston Tony -UNIFACE and the N-Tier Architecture, 2000
59. [Ma02] Marston Tony - The 3-Tier Architecture - some criticisms of Compuware's approach, 2002
60. [Ma93] Martin, James. - *Principles of Object-Oriented Analysis and Design*. Englewood Cliffs, NJ: Prentice Hall, 1993
61. [Ma99] Ma, R. - An Orthodox Blup Approach to Generalized Linear Mixed Models. Ph.D. thesis, Department of Statistics, University of British Columbia, Vancouver, 1999.
62. [Mach99] S.G. Machado, M.A.R. Freitas ,C.H. Gadini - Genetic parameters of test day milk yields of holstein cows .*Genetics and Molecular Biology*, 22, 3, 383-386, 1999.
63. [Martin93] Martin, James. *Principles of Object-Oriented Analysis and Design*. Englewood Cliffs, NJ: Prentice Hall, 1993
64. [McC01] McCulloch, Searle, *Generalized, Linear and Mixed Models*, Wiley, 2001
65. [McP04] Mary Sara McPeck, Xiaodong Wu,Carole Ober - Best Linear Unbiased Allele-Frequency Estimation in Complex Pedigrees, *Biometrics* 60, 359–367 , June 2004.
66. [Mi03] Mariana Milosescu- Baze de date in Visual FoxPro, Ed Teora, 2003.

67. [Mi03] Miller Granville - What's New in UML. 2.0? A Borland White Paper, 2003.
68. [Microsoft200] Microsoft Corporation - Microsoft SQL Sever 2000.
69. [Mo02] Morten Nobel-Jørgensen, Rune Rasmussen, Object Relational Database Framework ORDBFX Theory and implementation, Database Systems, Aalborg University, Denmark 2002, www.eyeweb.dk/ordbfx
70. [Mu00] Blair Murray, Peter Brand - Test Day Model. Genetic Evaluation for Dairy Production Traits, 2000. http://www.gov.on.ca/OMAFRA/english/livestock/dairy/facts/info_cdn.htm
71. [Mu04] Blair Murray, Peter Brand –Test Day Model genetic Evaluation for dairy Production Traits, Canadian Dairy Network website, august, 2004.
72. [Ne97] Nelson King, H. "Object DBMSs: Now or Never," DBMS Online, July 1997, <http://www.dbmsmag.com/9707d13.html>
73. [Ob01] Obasanjo Dare - An exploration of Object oriented database management systems – 2001, <http://www.25hoursaday.com/WhyArentYouUsingAnOODBMS.html>
74. [Oth02] M.H.Othmane, L.F. DeLaFuete, J.A.Carriedo, F. San Primitivo – Heritability and Genetic Correlation of the Test-Day Yield and Composition, Individual laboratory Cheese Yield and Somatic cell Count for Dairy Ewes, Journal. Dairy Science, Vol. 85, No.10, 2692-2698, 2002.
75. [Pa00] Panaitescu Razvan- Descoperirea "CACHE"-ului CHIP 10/2000.
76. [Pa03] Panchal Nimesh - C# Corenr - Implementing MVC Design Pattern in .NET, 2003.
77. [Pe95] Jeff Perkins, Bryan Morgan – SQL fara professor in 14 zile, Ed Teora, 1995.
78. [Po00] M. H. Pool, L.L.G. Janss, and T.H.E. Meuwissen - Genetic Parameters of Legendre Polynomials for First Parity Lactation Curves, Journal of Dairy Science Vol. 83, No. 11, 2000.
79. [Po98] Dorin Mircea Popovici, Ioan Mircea Popopvivi, Jean gabriel Rican – Proiectare si implementare software, Ed Teora, 1998.
80. [Pt93] Ptak E., Schaeffer L.R. Use of the test day yields for genetic evaluation of dairy cattle, Livest.Prod. Science, 34-23, 1993.
81. [Re00] Alvin C. Rencher- Linear Models in Statistics, A Wiley-Interscience Publication, 2000.
82. [Re97] Rennhackkamp Martin, "Extending Relational DBMSs" DBMS, December 1997, <http://www.dbmsmag.com/9712d15.html>
83. [Ro91]Robinson, C.K. - That BLUP is a good thing: the estimation of random effects. Statist. Sci. 6, 15-32, 1991.
84. [Sa00]Sava, Florin - Caiet de seminar pentru statistica sociala. Timisoara: Tipografia Universitatii de Vest din Timisoara, 2000.

85. [Sa01] E. Sas, I. Hutu – Manual de zootehnie practica, Editura Brumar, Timisoara, 2001.
86. [Sa96] E. Sas - Tratat elementar de Zootehnie aplicata, Ed. Helicon, Timisoara, 1996.
87. [Sc02] Scortan Sorin - Dezvoltarea aplicațiilor web folosind arhitectura Model-View- Controller (MVC) model 2 și Apache Struts, 2002.
88. [Sch96] Schaeffer, L. R., J. Jamrozik.. Multiple-trait prediction of lactation yields for dairy cows. J. Dairy Sci. 79:2044–2055, 1996.
89. [Se03] Chris Sells - Windows Forms Programming in C#. Addison-Wesley, 2003
90. [Sh02] Shusman Dan, Oscillating Between Objects and Relational: The Impedance Mismatch, InterSystems Corporation for ITtoolbox Database, 2002, <http://database.ittoolbox.com/documents/document.asp?i=1948>
91. [Si99] Simovici D.A., Tenney , R.L., Relational database System, Academic Press, NY, 1999
92. [Sr96] Srinivasan V., Chang D.T. - Object persistence in object-oriented application, IBM System Journal, Vol.36, No1, 1997, <http://www.research.ibm.com/journal/sj/361/srinivasan.html>
93. [St01] Stonebraker Michael - Object-Relational DBMS - The Next Wave Discussion Report , 2001, <http://www.eecs.wsu.edu/~cdyreson/teaching/advancedDatabase/011/discussions/tworeport.htm#security>
94. [St98] Stonebraker Michael. "Object-Relational DBMS--The Next Wave." Informix website. <http://db.cs.berkeley.edu/papers/Informix/www.informix.com/informix/corpinfo/zines/whitpprs/illuswp/wave.htm>
95. [Su03] Sullivan Sean C. - Advanced DAO programming. Learn techniques for building better DAOs, 2003, <http://www-128.ibm.com/developerworks/java/library/j-dao/>
96. [Sw95] Swalve, H.H.- Test Day Models in the Analysis of Dairy Production Data - A Review. Arch. Tierz., Dummerstorf,; 38: 591-612, 1995.
97. [Swa95] H. H. Swalve - The Effect of Test Day Models on the Estimation of Genetic Parameters and Breeding Values for Dairy Yield Traits, J Dairy Sci 78:929-938, 1995.
98. [Together01] Unknown, User's guide, TogetherSoft Corporation, 2001.
99. [Ts00] S. Tsuruta , I. Misztall, T. J. Lawlor, L Klei - .Estimation of genetic correlations among production, body size, udder, and roductive life traits over time in Holsteins. J Anim. Sci. Vol. 81, Suppl. 1/J. Dairy Sci. Vol. 86, 2000.
100. [Ub03] Übrahim KAYA, Yavuz AKBAP, Can UZMAY Estimation of Breeding Values for Dairy Cattle Using Test-Day Milk Yields, Turk J Vet Anim Sci, 27 , 459-464, 2003.

101. [Un00] Unknown - UML Tutorial - Windows Platform, Advanced Software Technologies, Inc., 2000
102. [Un90] Unknown - Directive92/102/CEE du Conseil du 27 November 1990 concernant l'identification et l'enregistrement animaux.
103. [Un92] Unknown - Coefficient of inbreeding – Wright's Equation and Hardiman's Method, <http://www.highflyer.mpranet.com/coefficient.htm>
104. [Un93] Unknown - Calculation inbreeding and relationship, the tubular method.
<http://www.kursus.kul.dk/shares/vertgen/popgen/genetik/applets/tubulara pp.htm>
105. [Vi88] Vintilă I., Bazele ameliorării genetice a populațiilor de animale domestice, Edit. Facla, Timișoara, 1988.
106. [Wi02] Shawn Wildermuth - Pragmatic ADO.NET Data Access for the Internet World, Addison-Wesley, October 2002
107. [Wi97] G. R. Wiggans, M.E. Goddard - A Computationally Feasible Test day Model For Genetic Evaluation of Yield Traits in the United States, Journal. Dairy Science, Vol. 80, No. 8, 1997.
108. [Za05] L.Zavadilova, E. Nemecova, J. Pribyl, J. Wolf – Definition of subgroups for fixed regression in the test-day animal model for milk production of Holstein Cattle in the Czech Republic, Czech J. Anim Sci, 50, 2005.