

1. Preliminarii
2. Standarde SQL
3. SQL în sistemul *Oracle*
4. Studiu de caz

1. PRELIMINARII

🔗 **Info :** istoric

- ✓ *SQL (Structured Query Language)* este limbajul standard de tratare a sistemelor relaționale
 - Prima varianta: mijlocul anilor '70, *SEQUEL* (pt proiectul *System R*), E.F. CODD, *IBM*,
 - Preluata in 1979 de Larry Elliot, *Relational Software* (=Oracle)
 - ✓ Acum un standard internațional (recunoscut *ANSI*):
 - Teoretic: acceptat de ~ toate produsele de baze de date
 - Comercial: inca domina piata
- motivul: baza teoretica riguroasa:
modelul relational.

1. PRELIMINARII

- ☞ **Info :** SQL = limbaj relational (neprocedural)
- ✓ SQL ~ limbaj neprocedural, declarativ
utilizatorii specifică CE CUM trebuie obținut,
=> compilatorul limbajului SQL generează automat o
procedură care accesează BD și execută comanda
- ✓ SQL = limbaj relational (nivel de abstractizare mai
ridicat → productivitate superioara)
bazat pe
- algebra relationala,
 - calculul relational,
 - etc., in functie caracteristicile necesare.

1. PRELIMINARII

 **Info :** 3 metode pt.implementarea limbajului SQL:

1. apelarea directă (*Direct Invocation*): constă în introducerea instrucțiunilor direct de la prompter (interactiv)
2. modularizarea (*Modul Language*): se folosesc proceduri apelate de programele aplicație
3. încapsularea (*Embedded SQL*): instrucțiunile SQL pot fi combinate cu instrucțiunile limbajului de programare al programului respectiv (Standardul SQL include suport pentru limbajele C, C++, COBOL, Java, Ada, M, Fortran, Pascal, PL/1)

1. PRELIMINARII

☞ **Info :** SQL și securitatea/ integritatea datelor

2 abordări ale problemei securității datelor, acceptate de orice SGBD:

- ✓ controlul discreționar
 - accesul la un anumit obiect este la discreția proprietarului acestuia,
- ✓ controlul obligatoriu
 - fiecare obiect are un nivel de clasificare,
 - fiecare utilizator are un nivel de permisiune.

1. Preliminarii
2. Standarde SQL
3. SQL în sistemul *Oracle*
4. Studiu de caz

2. STANDARDE SQL

- 1986: SQL a fost adoptat ca standard ANSI/ și ISO în 1987:
 - ❑ erau omise regulile de integritate referențială și anumiți operatori relaționali =>redundanța ridicată,
- 1989, ISO a publicat o revizie minimă, un addendum:
 - ❑ e introdusă o posibilitate de definire a integrității,
- 1992: prima revizie serioasă => **SQL2** sau **SQL-92**:
 - sunt definite tipurile de date simplu structurate,
- 1999: **SQL3** sau **SQL:99**:
 - ❑ e implementată tehnologia obiectuală (prin introducerea tipurilor complex structurate: tipuri obiect, referință și colecție și a tipurilor nestructurate: *Large Objects*.
 - ❑ se pot defini cereri recursive,
 - ❑ se introduc clase speciale de declanșatori și
 - ❑ se introduc caracteristici pentru gestiunea datelor complexe obiectuale,
=>definirea de extensii asupra tipurilor de date și, implicit, asupra tabelor care stochează date de acest tip.
 - ❑ se introduc structuri de control pentru a completa limbajul SQL din punct de vedere computațional (crearea, prelucrarea și interogarea structurilor de date obiectuale ce sunt stocate persistent într-o BD.

2. STANDARDE SQL

SQL:2003 și *SQL:2006*: definesc modalități în care SQL poate fi utilizat în conjuncție cu *XML*,

- *SQL:2003*:
 - ☐ este introdusă instrucțiunea *MERGE*, o combinație a comenzilor *INSERT* și *DELETE*;
- *SQL:2006*:
 - ☐ sunt definite modalitățile în care sunt importate și stocate date *XML* într-o bază de date *SQL*,
 - ☐ sunt definite modalitățile de prelucrare a acestora în baza de date,
 - ☐ sunt definite modalitățile de afișare a datelor *XML* și specifice *SQL* în format *XML* etc.
 - ☐ sunt furnizate facilități care permit aplicațiilor să integreze *Xquery* în codul *SQL*, permițând acces simultan atât la date *SQL*, cât și la documente *XML*.

1. Preliminarii
2. Standarde SQL
3. *SQL în sistemul Oracle*
4. Studiu de caz

3. SQL ÎN SISTEMUL ORACLE

3. SQL în sistemul Oracle

3.1. Oracle SQL

3.2. Modalitati de executare a comenzilor SQL

3.3. Clasificarea comenzilor SQL (LDD, LMD, LCT, LCD, alte tipuri de comenzi)

3.4. Procesarea comenzilor SQL

3.5. Optimizarea comenzilor SQL

3. SQL ÎN SISTEMUL ORACLE

3.1. Oracle SQL

SQL = asigură comunicarea cu serverul *Oracle*

=> reduce timpul necesar creării și întreținerii aplicațiilor de baze de date,

Oracle SQL = include extensii ale limbajului SQL standard *ANSI/ISO*

Instrumentele și aplicațiile *Oracle* = furnizează instrucțiuni suplimentare.

Utilitarele permit

- executarea instrucțiunilor limbajului SQL standard asupra unei baze de date *Oracle*,
- executarea instrucțiunilor sau funcțiilor suplimentare disponibile.

3. SQL ÎN SISTEMUL ORACLE

3.2. Modalități de a executa instrucțiunile SQL

și de a obține rezultate din baza de date:

- ✓ scrierea de programe utilizând *Oracle Forms and Reports*,
- ✓ încapsularea instrucțiunilor SQL în programe scrise în alte limbaje:
 - ❑ *Oracle Pro*C* : permite adăugarea de instrucțiuni SQL într-un program C,
 - ❑ interfeța *JDBC* : permite adăugarea de instrucțiuni SQL într-un program *Java*;
- ✓ utilizarea unor utilitare (*SQL*Plus* și *iSQL*Plus*) care permit execuția de instrucțiuni SQL și listarea rezultatelor acestora ≡ comunicarea dintre utilizator și BD (idem: produsul *Oracle SQL Developer* dar acesta nu este integrat în sistemul *Oracle*).

3. SQL ÎN SISTEMUL ORACLE

3.3. Clasificarea comenzilor SQL

Instrucțiunile SQL se împart în mai multe categorii, în funcție de tipul acțiunii pe care o realizează:

- a. limbajul de definire a datelor (*LDD*);
- b. limbajul de prelucrare a datelor (*LMD*);
- c. limbajul de control al tranzacțiilor (*LCT*);
- d. limbajul de control al datelor (*LCD*);
- e. comenzi speciale:
 - i. instrucțiunile pentru controlul sesiunii,
 - ii. instructiuni pentru controlul sistemului,
 - iii. instrucțiunile SQL încapsulate.

3. SQL ÎN SISTEMUL ORACLE

(a) Limbajul de definire a datelor (*LDD*):

- specific fiecărui SGBD
- funcțiile principale sunt aceleași
- conceptual, LDD realizează:
 - definirea entităților și a atributelor acestora prin: nume, formă de memorare, lungime
 - precizarea relațiilor dintre date și strategiile de acces la ele
 - stabilirea criteriilor diferențiate de confidențialitate
 - stabilirea criteriilor de validare automată a datelor utilizate.

3. SQL ÎN SISTEMUL ORACLE

=> **LDD** contine comenzi pt. definirea structurii obiectelor unei scheme:

1. crearea, modificarea și ștergerea obiectelor schemei și a altor structuri ale BD, inclusiv a bazei însăși și a utilizatorilor acesteia (*CREATE*, *ALTER*, *DROP*);
2. redenumirea obiectelor schemei (*RENAME*);
3. ștergerea tuturor datelor din obiectele schemei, fără a suprima structura acestor obiecte (*TRUNCATE*);
4. pornirea și oprirea opțiunilor de audit (*AUDIT*, *NOAUDIT*);
5. analiza informațiilor asupra unui tabel, index sau cluster (*ANALYZE*, *ASSOCIATE STATISTICS*, *DISASSOCIATE STATISTICS*);
6. adăugarea unui comentariu în dicționarul datelor (*COMMENT*);
7. suprimarea definitivă a obiectelor (*PURGE*) și revenirea la o stare anterioară a obiectelor (*FLASHBACK*).

O comandă **LDD**

- salvează, implicit, tranzacția curentă,
- marchează începutul unei noi tranzacții;

După execuția unei instrucțiuni **LDD**, sistemul *Oracle* declanșează automat instrucțiuni **SQL** care modifică informația din dicționarul datelor.

3. SQL ÎN SISTEMUL ORACLE

(b) Limbajul de prelucrare a datelor (*LMD*):

- permite formalizarea operațiilor care trebuie executate asupra unei baze de date sub forma unor comenzi
- o comandă are următoarea structură:
 - operația (deschidere/închidere, calcul aritmetic/logic, editare, extragere, adăugare, ștergere, căutare, reactualizare etc.),
 - criterii de selecție,
 - mod de acces (secvențial, indexat etc.),
 - format de editare;
- 2 tipuri de LMD:
 - procedurale (specifică modul în care se obține rezultatul unei comenzi LMD)
 - neprocedurale (descriu doar datele ce vor fi obținute și nu modalitatea de obținere a acestora).

3. SQL ÎN SISTEMUL ORACLE

⇒ **LMD** contine comenzi pt. interogarea si prelucrarea datelor din obiectele unei scheme:

- ✓ interogarea datelor din unul sau mai multe tabele (*SELECT*);
- ✓ adăugarea de înregistrări în tabele sau vizualizări (*INSERT*);
- ✓ actualizarea valorilor unor coloane din înregistrările existente în tabele sau vizualizări (*UPDATE*);
- ✓ adăugarea sau actualizarea condiționată a înregistrărilor în tabele sau vizualizări (*MERGE*);
- ✓ ștergerea de înregistrări din tabele sau vizualizări (*DELETE*);
- ✓ obținerea planului de execuție a instrucțiunilor SQL (*EXPLAIN PLAN*);
- ✓ blocarea unui tabel sau a unei vizualizări, limitând temporar accesul celorlalți utilizatori (*LOCK TABLE*).

3. SQL ÎN SISTEMUL ORACLE

(c) Limbajul de control al tranzactiilor (LCT):

contine comenzi pt. gestionarea modificărilor efectuate de către comenzile LMD și grupează aceste comenzi în unități logice, numite tranzacții:

- ✓ salvarea modificărilor unei tranzacții (*COMMIT*);
- ✓ anularea modificărilor dintr-o tranzacție fie în întregime, fie începând de la un punct intermediar (*ROLLBACK*);
- ✓ definirea unui punct intermediar până la care tranzacția poate fi anulată (*SAVEPOINT*);
- ✓ stabilirea de proprietăți ale tranzacției (*SET TRANSACTION TO*).

3. SQL ÎN SISTEMUL ORACLE

(d) Limbajul de control al datelor (*LCD*):

contine comenzi pt. gestionarea accesului la date
(asigurarea confidențialității și integrității datelor, salvarea informației în cazul unor defecțiuni, obținerea unor performanțe, rezolvarea unor probleme de concurență):

- ✓ acordarea de privilegii și *role*-uri (*GRANT*);
- ✓ revocarea privilegiilor și *role*-urilor acordate (*REVOKE*).

3. SQL ÎN SISTEMUL ORACLE

(e) Comenzi speciale:

- i. Instrucțiunile pentru controlul sesiunii
- ii. Instrucțiunile pentru controlul sistemului
- iii. Instrucțiunile SQL încapsulate.

3. SQL ÎN SISTEMUL ORACLE

i. Instrucțiunile pentru controlul sesiunii

permit gestionarea proprietăților sesiunii unui utilizator; ex.

- ✓ *ALTER SESSION*: permite modificarea sesiunii curente, astfel încât aceasta să îndeplinească funcțiuni specializate,
- ✓ *SET ROLE*: determină activarea sau dezactivarea *role*-urilor pentru sesiunea curentă;

ii. Instrucțiunile pentru controlul sistemului

- ✓ *ALTER SYSTEM* permite controlul sistemului, modificand, în mod dinamic, proprietățile instanței serverului *Oracle* (schimbarea anumitor setări: numărul minim de servere partajate, restricționarea sau suprimarea unei sesiuni, golirea zonei *shared pool* din *SGA*, suspendarea tuturor operațiilor *I/O* etc.).

3. SQL ÎN SISTEMUL ORACLE

iii. Instrucțiunile SQL încapsulate

= sunt reprezentare de comenzi *LDD*, *LMD* și *LCT* care pot fi încorporate în programe scrise în limbaje procedurale, urmând să fie utilizate prin intermediul precompilatoarelor *Oracle*

permit:

- ✓ definirea, alocarea și eliberarea cursorilor (*DECLARE CURSOR, OPEN, CLOSE*),
- ✓ specificarea unei baze de date și conectarea la sistemul *Oracle* (*DECLARE DATABASE, CONNECT*),
- ✓ declararea de variabile (*DECLARE STATEMENT*),
- ✓ inițializarea de descriptori (*DESCRIBE*),
- ✓ specificarea modului în care urmează să fie tratate erorile și avertismentele (*WHENEVER*),
- ✓ analizarea și executarea instrucțiunilor SQL (*PREPARE, EXECUTE, EXECUTE IMMEDIATE*),
- ✓ regăsirea informațiilor din baza de date (*FETCH*).

3. SQL ÎN SISTEMUL ORACLE

3.3. *Procesarea comenzilor SQL*

✓ procesarea unei **comenzi SQL** presupune parcurgerea urmatoarelor etape:

- A. crearea unui cursor,
- B. analiza instructiunii,
- C. legarea variabilelor,
- D. [paralelizarea],
- E. executarea instructiunii,
- F. închiderea cursorului;

✓ procesarea unei **interogari SQL**: presupune parcurgerea catorva etape suplimentare:

- descrierea rezultatelor,
- definirea modului de prezentare a rezultatelor
- recuperarea liniilor selectate.

3. SQL ÎN SISTEMUL ORACLE

(A) Crearea unui cursor

Pentru analizarea sintactică și semantică explicită a instrucțiunilor SQL încapsulate într-o aplicație se utilizează niste resurse de memorie speciale, numite cursoare



Definitii Cursor =

= o zonă de memorie în care se rețin:

- instrucțiunea SQL care urmează a fi analizată (procesată) și
- informații necesare procesării acesteia

= o structura de control care:

- asigură parcurgerea înregistrărilor dintr-o BD
- prelucrarea corepunzătoare a acestora: returnarea, adăugarea, ștergerea înregistrărilor din BD

= un pointer către o înregistrare dintr-un set de înregistrări; el poate accesa înregistrările rand pe rand și – la nevoie – se poate deplasa la o anumită înregistrare.

3. SQL ÎN SISTEMUL ORACLE



Observatii

- Sistemul *Oracle* gestionează în mod automat cursoarele
- Un cursor este creat independent de instrucțiunea *SQL*. El este generat automat, în așteptarea instrucțiunii *SQL* căreia să îi fie asociat
- Fiecare sesiune poate deschide mai multe cursoare, până la limita stabilită de parametrul de inițializare *OPEN_CURSORS*
- Pentru a elibera memoria sistemului, se recomandă ca aplicațiile să prevadă închiderea cursoarelor care nu mai sunt necesare.

3. SQL ÎN SISTEMUL ORACLE

Clasificarea cursorurilor

α) in functie de directia de deplasare:

scrollable și non-scrollable,

β) in functie de relatia cu datele:

modificabile și nemodificabile;

(α)

Execuția unui cursor plasează rezultatul cererii asociate într-o mulțime de linii (mulțime rezultat), care pot fi regăsite secvențial sau nesecvențial;

Există interfețe care permit regăsirea de linii recuperate anterior, i.e.:

- recuperarea liniei n din mulțimea rezultat sau
- recuperarea celei de a n -a linii de la poziția curentă.

3. SQL ÎN SISTEMUL ORACLE

Cursoarele *scrollable* =

- = cele pentru care operațiile *LMD* și de regăsire **nu** trebuie să se desfășoare secvențial, într-un singur sens (de la început către sfârșit);
- = pot accesa orice înregistrare din multimea rezultat astfel:
 - i. se declara cursorul,
 - ii. se specifica pozitia cursorului; ea poate fi:
 - **absoluta** (specificata in raport cu prima înregistrare din multimea rezultat),
 - **relativa** (specificata in raport cu pozitia curenta a cursorului).

3. SQL ÎN SISTEMUL ORACLE

Sintaxa

DECLARE *nume_cursor* sensitivity **SCROLL** CURSOR
FOR SELECT ... FROM ...

FETCH [NEXT | PRIOR | FIRST | LAST] FROM
nume_cursor

FETCH ABSOLUTE *n* FROM *nume_cursor*

FETCH RELATIVE *n* FROM *nume_cursor*

Observatie

este posibil ca un cursor *scrollable* sa acceseze aceiasi inregistrare din multimea-rezultat de mai multe ori => modificarile generate de alte tranzactii (*insert*, *update*, *delete*) pot conduce la modificarea multimii-rezultat.

3. SQL ÎN SISTEMUL ORACLE

Cursoarele *nonscrollable (forward-only)* =

- pot accesa fiecare inregistrare cel mult o data,
- apoi cursoul se deplaseaza automat la urmatoarea inregistrare;
- daca inregsitrarea returnata a fost ultima, o noua comanda FETCH va lasa cursorul tot pe aceasta și va da un mesaj de atentionare.

3. SQL ÎN SISTEMUL ORACLE

β)

Un cursor poate fi **senzitiv / nesenzitiv / asenzitiv** la aceste modificari ale datelor (le sesizeaza / nu le sesizeaza / le sesizeaza in conformitate cu specificatiile SGBD respectiv).

3. SQL ÎN SISTEMUL ORACLE

(B) Analiza comenzii

- În timpul analizei, instrucțiunea este transferată de la procesul utilizator la sistemul *Oracle*.
- Se efectueaza mai multe operatii:
 - ✓ interpretarea instrucțiunii,
 - ✓ verificarea validității instrucțiunii,
 - ✓ verificarea definițiilor de tabele și coloane în dicționarul datelor,
 - ✓ verificarea privilegiilor de acces la obiectele referite în instrucțiune,
 - ✓ determinarea planului de execuție optim al instrucțiunii,
 - ✓ încărcarea instrucțiunii într-o zonă SQL partajată,
 - ✓ direcționarea instrucțiunilor distribuite către nodurile distante care conțin informația referită.,
 - ✓ identificarea acelor erori care pot fi depistate înaintea execuției instrucțiunii.

3. SQL ÎN SISTEMUL ORACLE

(C) Legarea variabilelor (*variables binding*)

= procesul prin care se obtin valorile variabilelor la care face referință instrucțiunea și care trebuie cunoscute în vederea execuției acesteia

(D) [Paralelizarea]

= o etapă opțională, care se poate efectua înaintea execuției efective a unei instrucțiuni

Sistemul *Oracle* poate paraleliza comenzile *LMD* și unele operații *LDD* (crearea indecșilor, crearea unui tabel cu ajutorul unei subcereri, operații asupra partițiilor etc.)

Paralelizarea determină execuția instrucțiunii SQL de către mai multe procese server, ceea ce reduce timpul necesar operației.

3. SQL ÎN SISTEMUL ORACLE

(E) Executia comenzii

- ✓ Execuția unei instrucțiuni *UPDATE* sau *DELETE* determină blocarea liniilor afectate de aceasta.
 - Scopul unei astfel de operații este asigurarea integrității datelor
 - Liniile sunt deblocate la permanentizarea sau anularea tranzacției corespunzătoare comenzii respective
- ✓ Execuția unei instrucțiuni *SELECT* și *INSERT* nu determină blocări

(F) Inchiderea cursorului

- ✓ Procesarea oricărei instrucțiuni SQL se finalizează prin închiderea cursorului asociat, adică eliberarea zonei de memorie utilizate în scopul prelucrării.

3. SQL ÎN SISTEMUL ORACLE

Procesarea unei interogari SQL

- ✓ parcurgerea etapelor necesare executiei unei comenzi,
- ✓ parcurgerea unor etape suplimentare:
 - descrierea rezultatelor interogarii și definirea modului de prezentare a rezultatelor:
 - se indica tipul, dimensiunea numele datelor,
 - dacă este necesar, sistemul efectuează conversii implicite ale valorilor returnate;
 - recuperarea liniilor unei interogari:
 - presupune selectarea și, eventual, ordonarea acestora.

3. SQL ÎN SISTEMUL ORACLE

3.4. Optimizarea comenzilor SQL

- ❑ constituie o etapa importantă în procesarea oricărei instrucțiuni *LMD*
- ❑ există mai multe posibilități de a executa o astfel de instrucțiune (datorate, de exemplu, ordinii în care sunt accesate tabelele sau indecșii)
- ❑ se realizeaza
 - i. cu ajutorul unui modul *software* al sistemului *Oracle* numit **optimizor**,
 - ii. prin directivele (*hint*) date de catre proiectantul aplicatiei sub forma de comentarii atasate instructiunilor SQL.

3. SQL ÎN SISTEMUL ORACLE

i. Sistemul *Oracle* dispune de:

- ☐ **optimizori pe bază de cost** (utilizati preponderent în versiunile cele mai recente ale lui *Oracle Server*)
- ☐ **optimizori pe bază de reguli** (nu au fost actualizati în noile versiuni dar au ramas disponibil pentru compatibilitate).

ii **Proiectantul** unei aplicații deține mai multe informații despre datele specifice acesteia =>

- ☐ el poate stabili căi mai eficiente pentru execuția instrucțiunilor SQL =>
- ☐ el include în textul comenzii SQL respective, sub formă de comentarii, niste **directive** (*hint*) de executie.

3. SQL ÎN SISTEMUL ORACLE

Pentru rularea unei instrucțiuni LMD:

- este necesar ca sistemul să efectueze o succesiune de pași care definesc un **plan de execuție**;

Un plan de execuție include:

- o metodă de acces și
- o ordonare a tabelelor referite în instrucțiune;

O schema stocata (*stored outline*) =

= o abstractizare a unui plan de execuție generat de către optimizor

- consta dintr-o mulțime de *hint-uri*,
- este editabila (programatorul poate interveni pentru a modifica un plan de execuție).

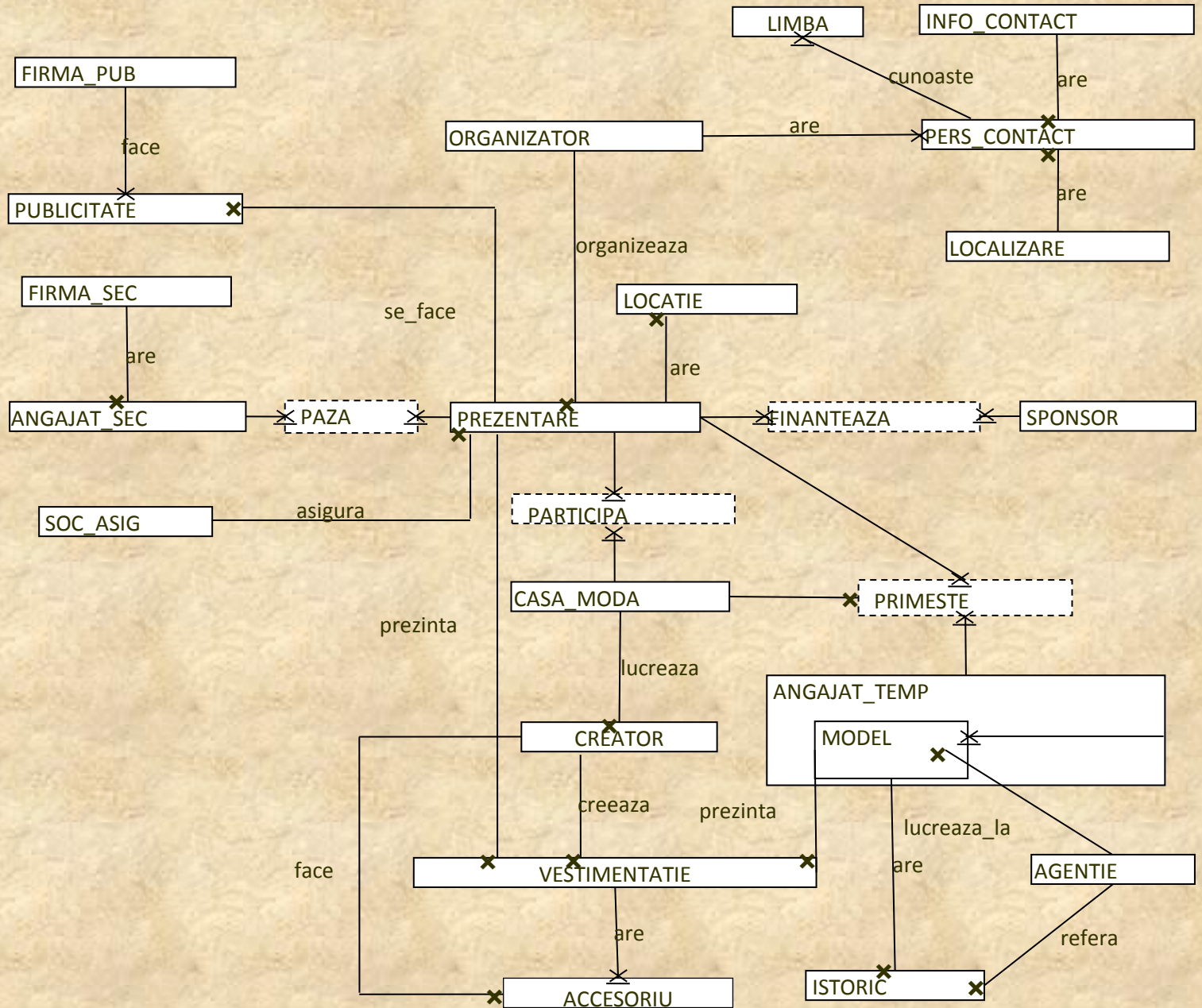
1. Preliminarii
2. Standarde SQL
3. SQL în sistemul *Oracle*
4. Studiu de caz

4. *STUDIU DE CAZ*

Un model de gestiune a prezentărilor de modă:

- organizatorii acestor prezentări:
- societățile de asigurare,
- societățile de pază și protecție,
- firmele de publicitate,
- sponsorii care finanțează prezentările,
- casele de modă care participă la prezentări,
- creatorii de modă care lucrează în cadrul fiecărei case de modă și realizează vestimentațiile expuse,
- modelele care prezintă aceste vestimentații,
- istoricul acestor modele referitor la agențiile cu care au colaborat.

Curs 4: Generalitati despre SQL



1. Preliminarii
2. Standarde SQL
3. SQL în sistemul *Oracle*
4. Studiu de caz