PROIECTAREA BAZELOR DE DATE-Anul 3
Laborator 1

Limbajul de manipulare a datelor (LMD) - I
Limbajul de definire a datelor (LDD) - II

Limbajul de manipulare a datelor (LMD)

- Comenzile SQL care alcătuiesc **LMD** permit:
  - regăsirea datelor (*SELECT*);
  - adăugarea de noi înregistrări (*INSERT*);
  - modificarea valorilor coloanelor din înregistrările existente (*UPDATE*);
  - adăugarea sau modificarea condiționată de înregistrări (*MERGE*);
  - suprimarea de înregistrări (*DELETE*).

- **Tranzacția** este o unitate logică de lucru, constituită dintr-o secvență de comenzi care trebuie să se execute atomic (ca un întreg) pentru a menține consistența bazei de date.
- *Server*-ul *Oracle* asigură consistența datelor pe baza tranzacțiilor, inclusiv în eventualitatea unei anomalii a unui proces sau a sistemului. Tranzacțiile oferă mai multă flexibilitate şi control în modificarea datelor.
- Comenzile SQL care alcătuiesc **LCD** sunt:
  - ROLLBACK – pentru a renunța la modificările aflate în aşteptare se utilizează instrucțiunea *ROLLBACK*. În urma execuției acesteia, se încheie tranzacția, se anulează modificările asupra datelor, se restaurează starea lor precedentă şi se eliberează blocările asupra liniilor.
  - COMMIT - determină încheierea tranzacției curente şi permanentizarea modificărilor care au intervenit pe parcursul acesteia. Instrucțiunea suprimă toate punctele intermediare definite în tranzacție şi eliberează blocările tranzacției.

    Obs: O comandă LDD (*CREATE*, *ALTER*, *DROP*) determină un COMMIT implicit.
  - SAVEPOINT - Instrucțiunea *SAVEPOINT* marchează un punct intermediar în procesarea tranzacției. În acest mod este posibilă împărțirea tranzacției în subtranzacții. Această instrucțiune nu face parte din standardul *ANSI* al limbajului *SQL*.

I. Comanda *INSERT*

1. Inserări mono-tabel

Comanda INSERT are următoarea sintaxă simplificată:

INSERT INTO *obiect* [***AS*** *alias*] [(*nume_coloană* [, *nume_coloană …*])]
{***VALUES*** ({*expr* | ***DEFAULT***} [, {*expr* | ***DEFAULT***} …])
| *subcerere*}

Subcererea specificată în comanda *INSERT* returnează linii care vor fi adăugate în tabel.

Dacă în tabel se introduc linii prin intermediul unei subcereri, coloanele din lista *SELECT* trebuie să corespundă, ca număr şi tip, celor precizate în clauza *INTO*. În absența unei liste de coloane în

clauza *INTO*, subcererea trebuie să furnizeze valori pentru fiecare atribut al obiectului destinație, respectând ordinea în care acestea au fost definite.

Observații (tipuri de date):

- Pentru claritate, este recomandată utilizarea unei liste de coloane în clauza *INSERT*.
- În clauza *VALUES*, valorile de tip caracter şi dată calendaristică trebuie incluse între apostrofuri. Nu se recomandă includerea între apostrofuri a valorilor numerice, întrucât aceasta ar determina conversii implicite la tipul *NUMBER*.
- Pentru introducerea de valori speciale în tabel, pot fi utilizate funcții.

Adăugarea unei linii care va conține valori *null* se poate realiza în mod:

- implicit, prin omiterea numelui coloanei din lista de coloane;
- explicit, prin specificarea în lista de valori a cuvântului cheie *null*

În cazul şirurilor de caractere sau al datelor calendaristice se poate preciza şirul vid ('').

Observații (erori):

Server-ul *Oracle* aplică automat toate tipurile de date, domeniile de valori şi constrângerile de integritate. La introducerea sau actualizarea de înregistrări, pot apărea erori în următoarele situații:

- nu a fost specificată o valoare pentru o coloană *NOT NULL*;
- există valori duplicat care încalcă o constrângere de unicitate;
- a fost încălcată constrângerea de cheie externă sau o constrângere de tip *CHECK*;
- există o incompatibilitate în privința tipurilor de date;
- s-a încercat inserarea unei valori având o dimensiune mai mare decât a coloanei corespunzătoare.

2. Inserari multi-tabel

O inserare multi-tabel presupune introducerea de linii calculate pe baza rezultatelor unei subcereri, într-unul sau mai multe tabele. Acest tip de inserare, introdus de *Oracle9i*, este util în mediul *data warehouse*.

Pentru o astfel de inserare, în versiunile anterioare lui *Oracle9i* erau necesare *n* operații independente *INSERT INTO…SELECT…*, unde *n* reprezintă numărul tabelelor destinație. Aceasta presupunea *n* procesări ale aceleiaşi surse de date şi, prin urmare, creşterea de *n* ori a timpului necesar procesului.

Sintaxa comenzii *INSERT* în acest caz poate fi:

- Pentru inserări necondiționate:

INSERT ALL INTO… [***INTO***…]
subcerere;

- Pentru inserări condiționate:

INSERT [***ALL*** | ***FIRST***]
WHEN *condiție* ***THEN INTO***…
[***WHEN*** *condiție* ***THEN INTO***…
[***ELSE INTO*** …]]
subcerere;

- *ALL* determină evaluarea tuturor condițiilor din clauzele *WHEN*. Pentru cele a căror valoare este *TRUE*, se inserează înregistrarea specificată în opțiunea *INTO* corespunzătoare.

- *FIRST* determină inserarea corespunzătoare primei clauze *WHEN* a cărei condiție este evaluată *TRUE*. Toate celelalte clauze *WHEN* sunt ignorate.

Exerciții [I]

1. Să se creeze tabelele *EMP*, *DEPT* (în şirul de caractere “pnu”, *p* reprezintă prima literă a prenumelui, iar *nu* reprezintă primele două litere ale numelui dumneavoastră), prin copierea structurii şi conținutului tabelelor *EMPLOYEES*, respectiv *DEPARTMENTS*.
2. Listați structura tabelelor sursă şi a celor create anterior. Ce se observă?
3. Listați conținutul tabelelor create anterior.
4. Să se insereze departamentul 300, cu numele *Programare* în *DEPT*.
5. Să se insereze un angajat corespunzător departamentului introdus anterior în tabelul *EMP*, precizând valoarea *NULL* pentru coloanele a căror valoare nu este cunoscută la inserare (metoda implicită de inserare). Determinați ca efectele instrucțiunii să devină permanente.
6. Să se mai introducă un angajat corespunzător departamentului 300, precizând după numele tabelului lista coloanelor în care se introduc valori (metoda explicita de inserare). Se presupune că data angajării acestuia este cea curentă (*SYSDATE*). Salvați înregistrarea.
7. Creați un nou tabel, numit *EMP1*, care va avea aceeaşi structură ca şi *EMPLOYEES*, dar nicio înregistrare. Copiați în tabelul *EMP1* salariații (din tabelul *EMPLOYEES*) al căror comision depăşeşte 25% din salariu.
8. Creați 2 tabele *emp2* şi *emp3* cu aceeaşi structură ca tabelul *EMPLOYEES*, dar fără înregistrări (acceptăm omiterea constrângerilor de integritate). Prin intermediul unei singure comenzi, copiați din tabelul *EMPLOYEES*:

- în tabelul *EMP1* salariații care au salariul mai mic decât 5000;
- în tabelul *EMP2* salariații care au salariul cuprins între 5000 şi 10000;
- în tabelul *EMP3* salariații care au salariul mai mare decât 10000.

Verificați rezultatele, apoi ştergeți toate înregistrările din aceste tabele.

II. Comanda UPDATE

Sintaxa simplificată a comenzii ***UPDATE*** este:

```
UPDATE nume_tabel [alias]
SET col1 = expr1[, col2=expr2]
[WHERE conditie];
```

sau

```
UPDATE nume_tabel [alias]
SET (col1,col2,...) = (subcerere)
[WHERE conditie];
```

Observații:

- de obicei pentru identificarea unei linii se foloseşte o condiție ce implică cheia primară;
- dacă nu apare clauza *WHERE* atunci sunt afectate toate liniile tabelului specificat;
- cazurile în care instrucțiunea *UPDATE* nu poate fi executată sunt similare celor în care eşuează instrucțiunea *INSERT*. Acestea au fost menționate anterior.

Exerciții [II]

9. Măriți salariul tuturor angajaților din tabelul *EMP* cu 5%. Vizualizati, iar apoi anulați modificările.

10. Schimbați jobul tuturor salariaților din departamentul 80 care au comision în 'SA_REP'. Anulați modificările.
11. Să se promoveze Douglas Grant la manager în departamentul 20, având o creştere de salariu cu 1000$.
12. Schimbați salariul şi comisionul celui mai prost plătit salariat din firmă, astfel încât să fie egale cu salariul si comisionul directorului.
13. Să se modifice jobul şi departamentul angajatului având codul 114, astfel încât să fie la fel cu cele ale angajatului având codul 205.

III. Comanda DELETE

Sintaxa simplificată a comenzii ***DELETE*** este:

DELETE FROM nume_tabel
[WHERE conditie];

Daca nu se specifica nici o conditie, vor fi şterse toate liniile din tabel.

Exercitii [III]

14. Ştergeți toate înregistrările din tabelul *DEPT*. Anulați modificările.
15. Ştergeți angajații care nu au comision. Anulați modificările.
16. Suprimați departamentele care un au nici un angajat. Anulați modificările.

Limbajul de definire a datelor (LDD)

- În general, instrucțiunile *LDD* sunt utilizate pentru definirea structurii corespunzătoare obiectelor unei scheme : tabele, vizualizări, vizualizări materializate, indecşi, sinonime, clustere, proceduri şi funcții stocate, declanşatori, pachete stocate etc.
- Aceste instrucțiuni permit:
  - crearea, modificarea şi suprimarea obiectelor unei scheme şi a altor obiecte ale bazei de date, inclusiv baza însăşi şi utilizatorii acesteia (*CREATE*, *ALTER*, *DROP*);
  - modificarea numelor obiectelor unei scheme (*RENAME*);
  - ştergerea datelor din obiectele unei scheme, fără suprimarea structurii obiectelor respective (*TRUNCATE*).
- Implicit, o instrucțiune *LDD* permanentizează (*COMMIT*) efectul tuturor instrucțiunilor precedente şi marchează începutul unei noi tranzacții.
- Instrucțiunile *LDD* au efect imediat asupra bazei de date şi înregistrează informația în dicționarul datelor.
- Definirea unui obiect presupune : crearea (*CREATE*), modificarea (*ALTER*) şi suprimarea sa (*DROP*).
- **Reguli de numire a obiectelor bazei de date**
  - Identificatorii obiectelor trebuie să înceapă cu o literă şi să aibă maxim 30 de caractere, cu excepția numelui bazei de date care este limitat la 8 caractere şi celui al legăturii unei baze de date, a cărui lungime poate atinge 128 de caractere.

- Numele poate conține caracterele *A-Z*, *a-z*, 0-9, _, *$* şi #.
- Două obiecte ale aceluiaşi utilizator al *server*-ului *Oracle* nu pot avea acelaşi nume.
- Identificatorii nu pot fi cuvinte rezervate ale *server*-ului *Oracle*.
- Identificatorii obiectelor nu sunt *case-sensitive*.

Definirea tabelelor

1. Crearea tabelelor

➢ Formele simplificate ale comenzii de creare a tabelelor sunt :

CREATE TABLE *nume_tabel* **(**
coloana_1 tip_date [constrangere_nivel_coloana]
[DEFAULT *valoare****],***
. . . . . .
coloana_n tip_date [constrangere_nivel_coloana]
[DEFAULT *valoare****],***
[*constrangeri_nivel_tabel****]***
);

sau

CREATE TABLE *nume_tabel [(coloana_1,..., coloana_n)]*
AS *subcerere;*

➢ **Constrângerile definite asupra unui tabel pot fi de următoarele tipuri:**

- NOT NULL - coloana nu poate conține valoarea *Null*; (*NOT NULL*)
- UNIQUE – pentru coloane sau combinații de coloane care trebuie să aibă valori unice în cadrul tabelului; (*UNIQUE* (*col1, col2, …*))
- PRIMARY KEY - identifică în mod unic orice înregistrare din tabel. Echivalent cu NOT NULL + UNIQUE; (*PRIMARY KEY* (*col1, col2, …*))
- FOREIGN KEY - stabileşte o relație de cheie externă între o coloană a tabelului şi o coloană dintr-un tabel specificat.
  [*FOREIGN KEY nume_col*]
  REFERENCES nume_tabel(nume_coloana)
  [ON DELETE {CASCADE| *SET NULL}]*
  - *FOREIGN KEY* este utilizat într-o constrângere la nivel de tabel pentru a defini coloana din tabelul „copil“;
  - *REFERENCES* identifică tabelul „părinte“ şi coloana corespunzătoare din acest tabel;
  - *ON DELETE CASCADE* determină ca, odată cu ştergerea unei linii din tabelul „părinte“, să fie şterse şi liniile dependente din tabelul „copil“;
  - *ON DELETE SET NULL* determină modificarea automată a valorilor cheii externe la valoarea *null*, atunci când se şterge valoarea „părinte“.
- CHECK- o condiție care să fie adevărată la nivel de coloană sau linie (*CHECK* (*conditie*)).

Obs:

- Constrângerile pot fi create odată cu tabelul sau adăugate ulterior cu o comandă ALTER TABLE.

- Constrângerile de tip CHECK se pot implementa la nivel de coloană doar dacă nu referă o altă coloană a tabelului.
- In cazul în care cheia primară este compusă, ea nu poate fi definită la nivel de coloane, ci doar la nivel de tabel.
- Constrângerea de tip NOT NULL se poate declara doar la nivel de coloană.

➢ Principalele **tipuri de date** pentru coloanele tabelelor sunt următoarele :

Tip de date	Descriere
VARCHAR2(n) [BYTE | CHAR]	Defineşte un şir de caractere de dimensiune variabilă, având lungimea maximă de n octeți sau caractere. Valoarea maximă a lui n corespunde la 4000 octeți, iar cea minimă este de un octet sau un caracter.
CHAR(n) [BYTE | CHAR]	Reprezintă un şir de caractere de lungime fixă având n octeți sau caractere. Valoarea maximă a lui n corespunde la 2000 octeți. Valoarea implicită şi minimă este de un octet.
NUMBER(p, s)	Reprezintă un număr având p cifre, dintre care s cifre formează partea zecimală
LONG	Conține şiruri de caractere având lungime variabilă, care nu pot ocupa mai mult de 2GB.
DATE	Reprezintă date calendaristice valide, între 1 ianuarie 4712 i.Hr. şi 31 decembrie 9999 d.Hr.

2. Modificarea (structurii) tabelelor

➢ **Modificarea structurii unui tabel** se face cu ajutorul comenzii ***ALTER TABLE***. Forma comenzii depinde de tipul modificării aduse:

- adăugarea unei noi coloane (nu se poate specifica poziția unei coloane noi în structura tabelului; o coloană nouă devine automat ultima în cadrul structurii tabelului)

  ALTER TABLE *nume_tabel*
  ADD (*coloana tip_de_date* ***[DEFAULT*** *expr][, ...]);*

- modificarea unei coloane (schimbarea tipului de date, a dimensiunii sau a valorii implicite a acesteia; schimbarea valorii implicite afectează numai inserările care succed modificării)

  ALTER TABLE *nume_tabel*
  MODIFY (*coloana tip_de_date* ***[DEFAULT*** *expr][, ...]);*

- eliminarea unei coloane din structura tabelului:

  ALTER TABLE *nume_tabel*
  DROP COLUMN *coloana;*
  sau
  ALTER TABLE *nume_tabel*
  DROP (*coloana*)***;***

Obs:

- dimensiunea unei coloane numerice sau de tip caracter poate fi mărită, dar nu poate fi micşorată decât dacă acea coloană conține numai valori *null* sau dacă tabelul nu conține nici o linie.
- tipul de date al unei coloane poate fi modificat doar dacă valorile coloanei respective sunt *null*.

- o coloană *CHAR* poate fi convertită la tipul de date *VARCHAR2* sau invers, numai dacă valorile coloanei sunt *null* sau dacă nu se modifică dimensiunea coloanei.

➢ Comanda *ALTER* permite **adăugarea unei constrângeri într-un tabel existent, eliminarea, activarea sau dezactivarea constrângerilor**.

- Pentru adăugare de constrângeri, comanda are forma:

  ALTER TABLE *nume_tabel*
  ADD [CONSTRAINT *nume_constr] tip_constr (coloana);*

- Pentru eliminare de constrângeri:

  ALTER TABLE *nume_tabel*
  DROP [CONSTRAINT *nume_constr] tip_constr (coloana);*

- Pentru activare/dezactivare constrângere:

  ALTER TABLE *nume_tabel*
  MODIFY CONSTRAINT *nume_constr* ***ENABLE|DISABLE****;*
  sau
  ALTER TABLE *nume_tabel*
  ENABLE| DISABLE *nume_constr;*

3. Suprimarea tabelelor

➢ Ştergerea fizică a unui tabel, inclusiv a înregistrărilor acestuia, se realizează prin comanda:
DROP TABLE *nume_tabel;*

➢ Pentru ştergerea conținutului unui tabel şi păstrarea structurii acestuia se poate utiliza comanda :
TRUNCATE TABLE *nume_tabel ;*

!!!Obs: Fiind operație LDD, comanda TRUNCATE are efect definitiv.

4. Redenumirea tabelelor

Comanda ***RENAME*** permite redenumirea unui tabel, vizualizare sau secvență.

RENAME *nume1_obiect* ***TO*** *nume2_obiect;*

Obs:

- În urma redenumirii sunt transferate automat constrângerile de integritate, indecşii şi privilegiile asupra vechilor obiecte.
- Sunt invalidate toate obiectele ce depind de obiectul redenumit, cum ar fi vizualizări, sinonime sau proceduri şi funcții stocate.

Exerciții [III]

17. Să se creeze tabelul ANGAJATI corespunzător schemei relaționale:
ANGAJATI(cod_ang, nume, prenume, email, data_ang, job, cod_sef, salariu, cod_dep)
în următoarele moduri:
   a) fără precizarea vreunei chei sau constrângeri
   b) cu precizarea cheilor primare la nivel de linie si a constrangerilor NOT NULL pentru coloanele nume şi salariu

c) cu precizarea cheii primare la nivel de tabel si a constrângerilor NOT NULL pentru coloanele nume şi salariu.

Se presupune că valoarea implicită a coloanei data_ang este SYSDATE.

Obs: Nu pot exista două tabele cu acelaşi nume în cadrul unei scheme, deci creerea unui tabel va fi precedata de suprimarea sa prin comanda : *DROP TABLE ANGAJATI ;*

18. Adăugați următoarele înregistrări în tabelul ANGAJATI :

Cod_ang	Nume	Prenume	Email	Data_ang	Job	Cod_sef	Salariu	Cod_dep
100	Nume1	Prenume1	Null	Null	Director	null	20000	10
101	Nume2	Prenume2	Nume2	02-02-2004	Inginer	100	10000	10
102	Nume3	Prenume3	Nume3	05-06-2000	Programator	101	5000	20
103	Nume4	Prenume4	Null	Null	Inginer	100	9000	20
104	Nume5	Prenume5	Nume5	Null	Programator	101	3000	30

Completați valorile coloanelor nume şi prenume cu cele corespunzătoare unor colegi din grupa.
Prima si a patra înregistrare vor fi introduse specificând coloanele pentru care introduceți date efectiv (inserare explicită), iar celelalte vor fi inserate fără precizarea coloanelor în comanda INSERT (inserare explicită). Salvați comenzile de inserare într-un fişier p2l3.sql.

19. Creați tabelul ANGAJATI_10, prin copierea angajaților din departamentul 10 din tabelul *ANGAJATI*. Listați structura noului tabel. Ce se observă ?
20. Introduceti coloana comision in tabelul *ANGAJATI*. Coloana va avea tipul de date *NUMBER*(4,2).
21. Este posibilă modificarea tipului coloanei salariu în NUMBER(6,2)?
22. Setați o valoare *DEFAULT* pentru coloana salariu.
23. Modificați tipul coloanei comision în *NUMBER(2, 2)* şi al coloanei salariu la *NUMBER(10,2)*, în cadrul aceleiaşi instrucțiuni.
24. Actualizati coloana comision, setând-o la valoarea 0.1 pentru salariații al căror job începe cu litera P. (*UPDATE*)
25. Modificați tipul de date al coloanei email în *VARCHAR2*.
26. Adăugați coloana *nr_telefon* în tabelul *ANGAJAT*, setându-i o valoare implicită.
27. Vizualizați înregistrările existente. Suprimați coloana nr_telefon.

Ce efect ar avea o comandă *ROLLBACK* în acest moment?

28. Redenumiți tabelul *ANGAJATI* în *ANGAJATI3*.
29. Consultați vizualizarea TAB din dicționarul datelor. Redenumiți *ANGAJAT3* în *ANGAJATI*.
30. Suprimați conținutul tabelului *ANGAJATI_10*, fără a suprima structura acestuia.
31. Creați şi tabelul *DEPARTAMENTE*, corespunzător schemei relaționale :
DEPARTAMENTE (cod_dep, nume, cod_director) specificând doar constrângerea NOT NULL pentru nume (nu precizați deocamdată constrângerea de cheie primară).

32. Introduceți următoarele înregistrări în tabelul DEPARTAMENTE :

Cod_dep	**Nume**	**Cod_director**
10	Administrativ	100
20	Proiectare	101
30	IT	Null

33. Se va preciza apoi cheia primara *cod_dep* (comanda *ALTER*).

Obs:

- Introducerea unei constrângeri după crearea tabelului, presupune ca toate liniile existente în tabel la momentul respectiv să satisfacă noua constrângere.
- Acest mod de specificare a constrângerilor permite numirea acestora.
- In situația in care constrângerile sunt precizate la nivel de coloană sau tabel (în *CREATE TABLE*) ele vor primi implicit nume atribuite de sistem, dacă nu se specifică vreun alt nume într-o clauză *CONSTRAINT*.

34. Să se precizeze constrângerea de cheie externă pentru coloana *cod_dep* din *ANGAJAT*.
a) fără suprimarea tabelului (*ALTER TABLE*);
b) prin suprimarea şi recrearea tabelului, cu precizarea noii constrângeri la nivel de coloană ({*DROP*, *CREATE*} *TABLE*). De asemenea, se vor mai preciza constrângerile (la nivel de coloană, dacă este posibil):
- *PRIMARY KEY* pentru *cod_ang*;
- *FOREIGN KEY* pentru *cod_sef*;
- *UNIQUE* pentru combinația *nume* + *prenume*;
- *UNIQUE* pentru *email* ;
- *NOT NULL* pentru *nume*, *email* ;
- verificarea *cod_dep* >0 ;
- verificarea ca salariul sa fie mai mare decât *comisionul**100 ;

35. Suprimați şi recreați tabelul, specificând toate constrângerile la nivel de tabel (în măsura în care este posibil).

36. Reintroduceți date în tabel, utilizând şi modificând, dacă este necesar fişierul p2l3.sql.

37. Ce se întâmplă dacă se încearcă suprimarea tabelului *DEPARTAMENTE*?
DROP TABLE departamente ;

38. Introduceți constrângerea *NOT NULL* asupra coloanei email.

39. (Incercați să) adăugați o nouă înregistrare în tabelul *ANGAJATI*, care să corespundă codului de departament 50. Se poate?

40. Adăugați un nou departament, cu numele *Analiza*, codul 60 şi directorul null în *DEPARTAMENTE*. *COMMIT*.

41. (Incercați să) ştergeți departamentul 20 din tabelul *DEPARTAMENTE*. Comentați.

42. Ştergeți departamentul 60 din *DEPARTAMENTE*. *ROLLBACK*.

43. (Incercați să) introduceți un nou angajat, specificând valoarea 114 pentru *cod_sef*. Ce se obține ?

44. Adăugați un nou angajat, având codul 114. Incercați din nou introducerea înregistrării de la exercițiul precedent.

?? Ce concluzii reies din exercițiile precedente ? Care este ordinea de inserare, atunci când avem constrângeri de cheie externă ?

45. Se doreşte ştergerea automată a angajaților dintr-un departament, odată cu suprimarea departamentului. Pentru aceasta, este necesară introducerea clauzei *ON DELETE CASCADE* în definirea constrângerii de cheie externă. Suprimați constrângerea de cheie externă asupra tabelului *ANGAJATI* şi reintroduceți această constrângere, specificând clauza *ON DELETE CASCADE*.

46. Ştergeți departamentul 20 din *DEPARTAMENTE*. Ce se întâmplă ? *Rollback*.

47. Introduceți constrângerea de cheie externă asupra coloanei *cod_director* a tabelului *DEPARTAMENTE*. Se doreşte ca ştergerea unui angajat care este director de departament să atragă după sine setarea automată a valorii coloanei *cod_director* la null.

48. Ştergeți angajatul având codul 114 din tabelul *ANGAJAT*. Analizați efectele comenzii. Rollback.

49. Adăugați o constrângere de tip *check* asupra coloanei salariu, astfel încât acesta să nu poată depăşi 30000.

50. Incercați actualizarea salariului angajatului 100 la valoarea 35000.

51. Dezactivați constrângerea creată anterior şi re–încercați actualizarea. Ce se întâmplă dacă încercăm reactivarea constrângerii?