



Laborator 2 - CONCEPTE DE BAZA DIN TEORIA GENERALA A BAZELOR DE DATE

1. O DEFINIȚIE A NOȚIUNII DE BAZĂ DE DATE

Una din caracteristicile ultimilor ani este explozia informațională. Volumul imens de informații nu mai poate fi utilizat eficient prin intermediul metodelor tradiționale. Prelucrarea automată a informațiilor cu ajutorul sistemelor electronice de calcul a devenit o necesitate pentru toate domeniile de activitate. Cea mai evoluată metodă de organizare a informațiilor în vederea prelucrării lor automate o întâlnim la bazele de date. În baza de date se memorează datele conform unui anumit model. Utilizatorul vede baza de date conform acestui model. Datele memorate urmează să fie gestionate (adaugare, regasire, ștergere, modificare) conform acestui model. Pentru ca aceste operații să se poată efectua cât mai ușor este necesar să se facă o descriere a datelor memorate în baza de date, și această descriere să fie memorată în diferite tabele.

În momentul în care se execută o anumită operație, trebuie să se consulte aceste tabele de descriere pentru a verifica dacă datele respective există, și eventual unde se află ele memorate.

*O bază de date este formată dintr-o colecție organizată de date. Aceste date se prelucrează cu un sistem de programe numit sistem de gestiune a bazelor de date. De obicei, se folosește notația **BD** pentru **Bazele de Date** și notația **SGBD** pentru **Sistemele de Gestiune a Bazelor de Date** (în engleză **Data Base Management System – DBMS**).*

Nivelurile de organizare a datelor în bazele de date sunt:

- *Nivelul conceptual*, dat de viziunea administratorului bazei de date;
- *Nivelul logic*, dat de viziunea programatorului de aplicații;
- *Nivelul fizic*, dat de viziunea inginerului de sistem.

Modelul de date este un ansamblu de concepte și instrumente pentru realizarea unei scheme conceptuale de bază de date. Această schemă permite descrierea fenomenelor din lumea reală prin entități care au atribute și valori, împreună cu legăturile dintre entități.

Un model de date pentru baze de date are următoarele elemente obligatorii:

- *definirea structurii modelului*, care presupune definirea obiectelor (entităților) și a caracteristicilor (atributelor) lor, precum și definirea legăturilor (asocierilor) între obiecte;
- *operatorii*, care acționează asupra structurilor de date;
- *regulile de integritate*, care asigură corectitudinea datelor.

Modelele de date pentru baze de date pot fi de următoarele *tipuri* (fundamentale): ierarhice, rețea, relaționale, orientate obiect.



Modelul ierarhic. Datele sunt reprezentate după structura unui arbore. Legăturile dintre date sunt ordonate unic, orice acces făcându-se prin vârful ierarhiei (rădăcină).

Un subordonat nu poate avea decât un singur superior, iar un superior poate avea oricâți subordonați.

Modelul rețea. Datele sunt reprezentate printr-o mulțime de ierarhii legate între ele. Un nod al rețelei poate avea oricâți subordonați și oricâți superiori. La un subordonat se poate ajunge pe mai multe căi.

Modelul relațional. Datele sunt reprezentate sub forma unor tabele (relații). Tabela este o submulțime a produsului cartezian a unor domenii de valori. În tabelă sunt coloane (attribute-caracteristici) și linii (tupluri-înregistrări). Legăturile între tabele sunt logice, realizate prin valori.

Modelul orientat obiect. Structura de baza folosită este cea de clasă de obiecte, definită prin abstractizarea entității fizice din lumea reală. Obiectele sunt o colecție de proprietăți care se referă la aceeași entitate, împreună cu operațiile permise (metodele). Comunicarea între obiecte se face prin mesaje (cereri de regăsire). Caracteristicile fundamentale ale obiectelor sunt: încapsularea, polimorfismul, succesiunea.

2. SISTEME DE GESTIUNE A BAZELOR DE DATE

Rolul unui SGBD pentru o baza de date este asemanator unui sistem de operare pentru un calculator.

Un sistem de gestiune a bazelor de date este format din mai multe programe care asigură, în principal, următoarele funcții:

- **Funcția de definire a datelor.** Prin această funcție se pot defini tipul datelor, structura lor, precum și relațiile dintre ele.
- **Funcția de creare a bazelor de date.** Această funcție asigură introducerea datelor în bazele de date. Operația de introducere a datelor într-o bază de date este numită încărcarea bazei de date.
- **Funcția de actualizare a bazelor de date.** Această funcție asigură ținerea la zi a datelor din bazele de date prin operații de adăugare a unor date noi, ștergere a unor date devenite inutile, actualizarea unor date, etc.
- **Funcția de interogare a bazelor de date.** Această funcție asigură accesul la datele înregistrate în bazele de date.
- **Funcția de administrare a bazelor de date.** Prin această funcție se stabilesc criteriile de actualizare a datelor, drepturile de acces la date, modul de protejare a datelor, etc.

Primele baze de date au apărut în anii '60. Unul din primii cercetători în domeniu, care a introdus de fapt conceptul de bază de date, este Ch. W. Bachman. Bazele de date de tip relațional sunt cele mai răspândite. Ele au fost introduse în 1970 de E.F. Codd.

Baze de date - SQL



Fundamentul teoretic al acestui tip de baze de date îl reprezintă teoria matematică a relațiilor. Larga răspândire a tipului relațional de baze de date se datorează și faptului că celelalte tipuri de baze de date se pot reduce ușor la bazele de tip relațional. În acest curs ne vom ocupa de bazele de date de tip relațional.

Cele mai cunoscute sisteme de gestiune a bazelor de date relaționale care s-au impus înainte de apariția PC-urilor sunt: **SQL (Structured Query Language)** și **QBE (Query By Example)**. Apariția PC-urilor a reorientat specialiștii în baze de date. Astfel, au apărut numeroase sisteme de gestiune a bazelor de date de tip relațional pentru PC-uri autonome sau conectate în rețele locale. Dintre acestea menționăm aici: dBase, Paradox, FoxPro, Oracle și Access.

3. BAZE DE DATE RELATIONALE

Înainte de a începe să realizăm o aplicație cu baze de date relaționale trebuie să cunoaștem **principalele concepte din teoria relațională**. Acestea sunt absolut necesare, atât pentru analiza și proiectarea bazei de date relaționale, cât și pentru elaborarea programelor de aplicație într-un SGBD relațional ales. Prezentăm în continuare aceste concepte:

DOMENIUL reprezintă un ansamblu de valori, caracterizat printr-un nume. Domeniul se poate defini explicit, prin enumerarea tuturor valorilor care aparțin acestuia, sau implicit, prin precizarea proprietăților pe care le au valorile domeniului respectiv.

RELAȚIA (tabela) reprezintă un subsansamblu al produsului cartezian dintre mai multe domenii caracterizat printr-un nume.

ATRIBUTUL este coloana unei relații caracterizate printr-un nume. Fiecare atribut își ia valorile dintr-un domeniu. Mai multe atribute pot lua valori din același domeniu.

TUPLUL este o linie dintr-o relație și nu are nume. Valorile dintr-un tuplu aparțin produsului cartezian dintre domeniile relației.

SCHEMA RELAȚIEI este un ansamblu format din numele relației (R) urmat între paranteze rotunde de lista atributelor (A_i), pentru fiecare atribut precizându-se domeniul asociat (D_i):

$$R (A_1: D_1, A_2: D_2, \dots, A_n: D_n),$$

unde $n \geq m$ sunt numere întregi.

CHEIA este un atribut sau un ansamblu de atribute cu ajutorul căruia se poate identifica un tuplu dintr-o relație.

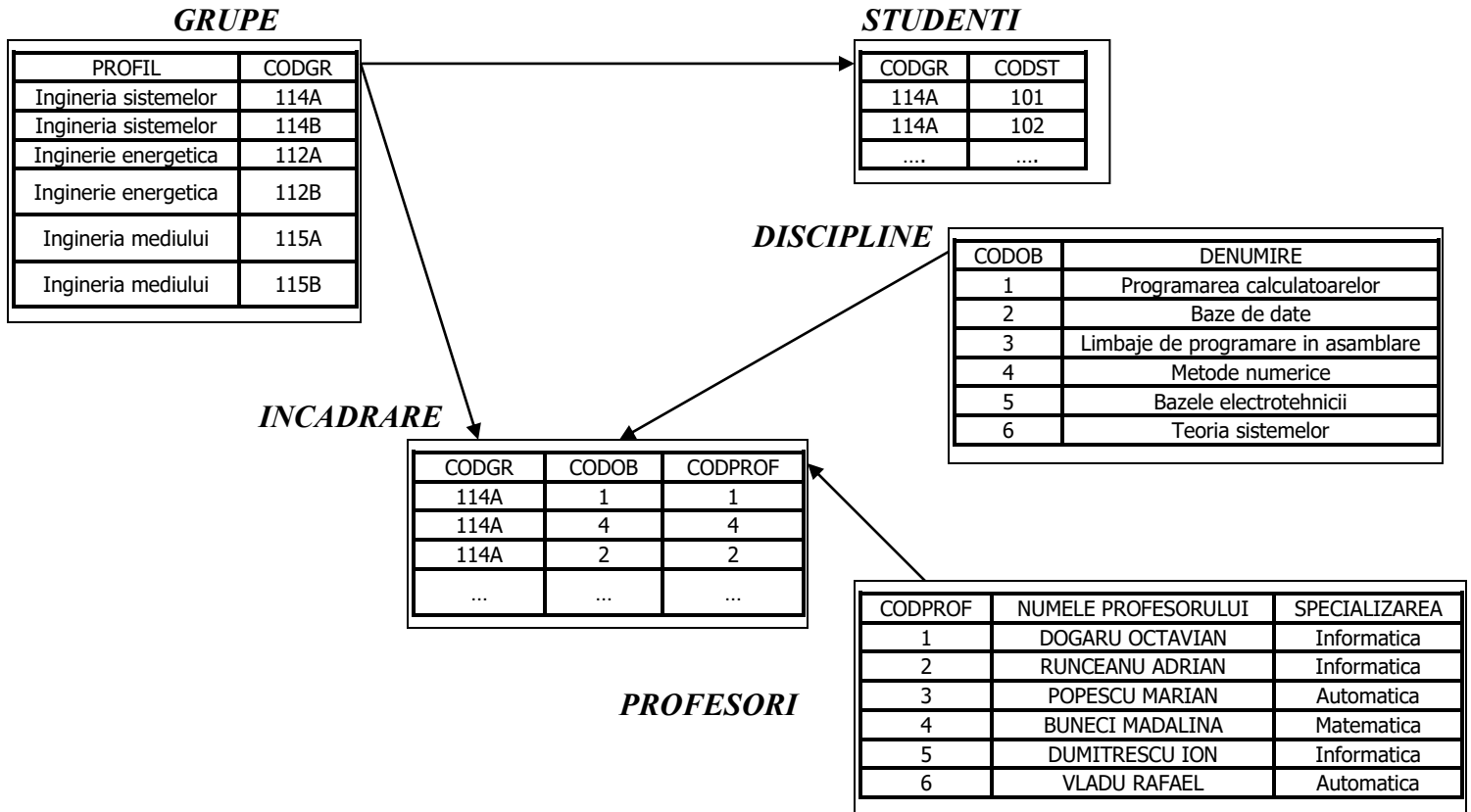
Numim **cheie primară (primary key)** a unei relații, un atribut (sau un grup de atribute) care identifică fără ambiguitate fiecare linie a relației. (De exemplu: atributul COD este cheie primară deoarece nu există două facultăți cu același cod)

Numim **cheie straină (foreign key)** a unei relații un grup de atribute care pune în legătură linia din două tabele (relații).



Exemplu:

Pentru exemplificare considerăm următoarea baza de date **FACULTATE** care contine cinci tabele:



1. Tabela GRUPE cu cheile: profil si codgr=codul grupei
2. Tabela STUDENTI cu cheile: codgr si codst=codul studentului
3. Tabela DISCIPLINE cu cheile codob=codul disciplinei si denumire
4. Tabela PROFESORI cu cheile codprof=codul profesorului, numele si specializarea
5. Tabela INCADRARE care are chei de la celelalte tabele: codgr, codob si codprof.

Cheile primare(identificatorii unici) sunt:

- în tabela GRUPE – codgr
- în tabela STUDENTI – codst
- în tabela DISCIPLINE – codob
- în tabela PROFESORI - codprof
- în tabela INCADRARE – atributul compus – codgr+codob+codprof

Cheile străine sunt:

- atributul GRUPE.codgr pentru tabela STUDENTI (refera tabela GRUPE)
- atributul INCADRARE.codob pentru tabela INCADRARE (refera tabela DISCIPLINE)



- atributul INCADRARE.codgr pentru tabela INCADRARE (refera tabela GRUPE)
- atributul INCADRARE.codprof pentru tabela INCADRARE (refera tabela PROFESORI).

Bibliografie

1. C.J. Date, An introduction to Database Systems, Addison-Wesley Publishing Company, 1995
2. I. Despi, G. Petrov, R. Reisz, A. Stepan, Teoria generală a bazelor de date, Editura Mirton, Timișoara, 2000

Probleme propuse spre rezolvare

1. Să se creeze tabelele următoare:

AN_XY

CodAn number(2) -- cheie primara

DenAn varchar2(10) -- nenula

NrGrupe number(2) -- implicit 0

GRUPA_XY

CodGrupa number(2) -- cheie primara

DenGrupa varchar2(10) -- nenula

NrStudenti number(3) -- implicit 0

CodAn number(2) -- referinta la cheia tabelii AN_XY,

MedieGrupa number(2) -- nenula

unde XY sunt inițialele voastre, astfel încât tabelele create de voi să fie distincte.

Să se folosească apoi comanda **DESCRIBE** pentru a se vedea tabelele create.

Solutie - pentru crearea tabelii **AN_XY**

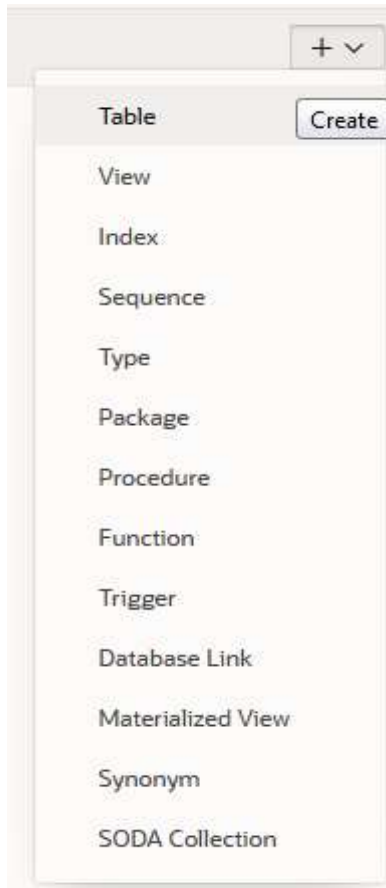
Pasul 1 – Din meniul principal se selecteaza optiunea **SQL Workshop** si apoi **Object Browser**.

Va aparea urmatoarea fereastră:

Baze de date - SQL



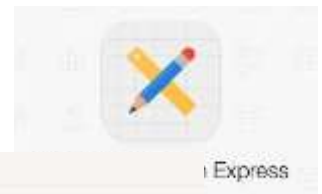
Din zona dreapta se selecteaza optiunea + si se alege **Table**:



Apoi va aparea urmatoarea fereastra de creare a unei tabele:

Pasul 2 - Se introduc denumirile si tipurile de date pentru fiecare din coloanele tabelului (**Columns**):

Baze de date - SQL



Create Table

Columns

Table Name: AN_AR

Preserve Case

Column Name	Type	Precision	Scale	Not Null	Identity	Move
CODAN	NUMBER					▲ ▼
DENAN	VARCHAR2					▲ ▼
NRGRUPE	NUMBER					▲ ▼
	- Select Datatype -					▲ ▼
	- Select Datatype -					▲ ▼
	- Select Datatype -					▲ ▼

Apoi se apasa butonul **Next**.

Create Table

Columns

Table Name: AN_AR

Preserve Case

Column Name	Type	Precision	Scale	Not Null	Identity	Move
CODAN	NUMBER					▲ ▼
DENAN	VARCHAR2					▲ ▼
NRGRUPE	NUMBER					▲ ▼
	- Select Datatype -					▲ ▼
	- Select Datatype -					▲ ▼
	- Select Datatype -					▲ ▼

Pasul 3 - Se precizeaza care este coloana(coloanele) care vor fi cheie primara (**Primary Key**), cat si modalitatea de adaugare implicita sau un a unei secvente de valori in acea coloana de tip cheie primara. Vom selecta optiunea **Not populated** pentru a avea posibilitatea de a introduce valori nomunale in coloana CodAn. Se selecteaza denumirea CodAn in zona **Primary Key** si apoi butonul **Next**.

Baze de date - SQL



Create Table

Primary Key

Table name: **AN_AR**

Primary Key: No Primary Key
 Populated from a new sequence
 Populated from an existing sequence
 Not populated
 Populated by Identity column

* Primary Key Constraint Name: **AN_AR_PK**

* Primary Key: **CODAN(NUMBER)**

Composite Primary Key: **- Select Composite Primary Key -**

Primary Key
A primary key allows each row in a table to be uniquely identified.
If you select to populate your primary key from a new sequence, you will be prompted to enter the new sequence's name. If you select to populate your primary key from an existing sequence, you will be prompted to select the sequence. Both these methods result in the generation of a trigger against your table. You can also select to not populate your primary. This is the only method that allows you to define a composite primary key made up of more than two columns.

Pasul 4 – Se stabileste ce coloana va fi cheia straina (**Foreign Key**) pentru legatura cu alta tabela. Pentru exemplul nostru un este necesara stabilirea unei astfel de informatii, drept pentru care se apasa pe butonul **Next**, fara a face nicio alta selectie.

Create Table

Foreign Key

Foreign Keys

Foreign Key | Column | Referenced Table | Referenced Columns | Action

Add Foreign Key

Name: **AN_AR_FK**

Disallow Delete
 Cascade Delete
 Set Null on Delete

Select Key Column(s): **CODAN
DENAN
NRGRUPE**

References Table:

Referenced Column(s):

Baze de date - SQL



Pasul 5 - se foloseste pentru stabilirea constrangerilor de integritate (notiunile teoretice vor fi prezentate intr-un curs ulterior). Nu se selecteaza nimic, doar se apasa pe butonul **Next**.

The screenshot shows the 'Create Table' wizard in Oracle Application Express, specifically the 'Constraints' step. At the top, a progress bar indicates that the first three steps are completed (green checkmarks) and the current step, 'Constraints', is active (green circle). Below the progress bar, there is a table with columns 'Constraint Name', 'Type', and 'Column(s)/Check'. To the right of this table is a green 'Add' button. Underneath, the 'Constraint Type' is set to 'Check' (selected with a radio button) and 'Unique' is unselected. A 'Check Condition' text area is empty. Below that, the 'Key Column(s)' section shows a list of columns: 'CODMAN', 'DENMAN', and 'NRGRUPE'. To the right of this list is a large empty text area for defining the constraint. At the bottom left, there are links for '> Available Columns' and '> Example Check Constraints'. A 'Name' input field is also visible.

Pasul 6 – Se selecteaza butonul **Create Table**.

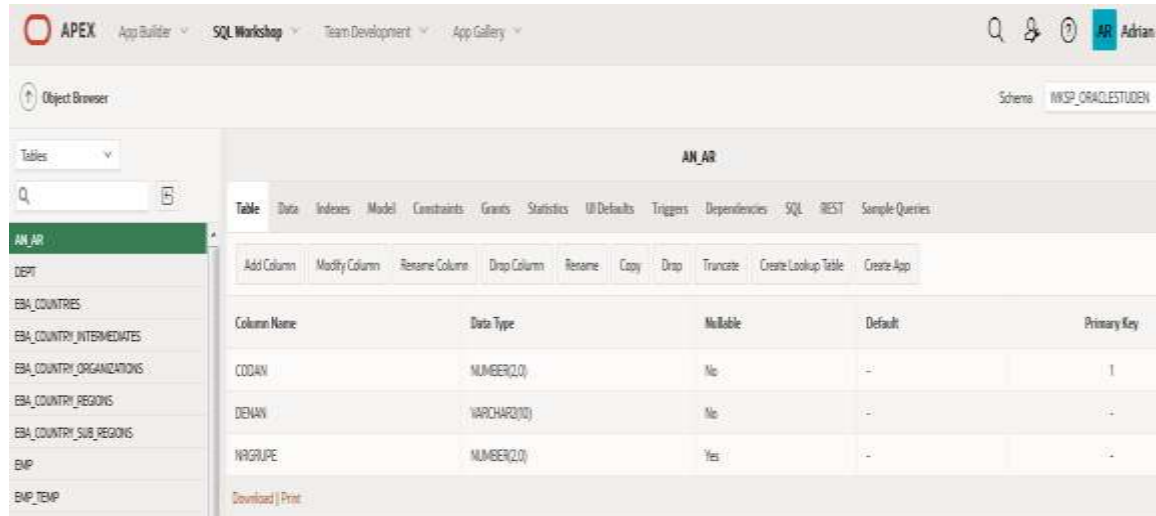
Nota: Daca se selecteaza zona **SQL**, va aparea comanda SQL care va crea efectiv tabela a carei structuri a fost stabilita in pasii anteriori.

The screenshot shows the 'Create Table' wizard in Oracle Application Express, specifically the 'Confirm' step. At the top, a progress bar indicates that all four previous steps are completed (green checkmarks) and the current step, 'Confirm', is active (green circle). Below the progress bar, the text 'Please confirm your request.' is displayed. Underneath, the 'Schema' is set to 'WKSP_ORACLESTUDENTIAIA' and the 'Table name' is 'AN_AR'. Below these fields, there is a link for '> SQL'. At the bottom left, there is a 'Cancel' button with a left arrow. At the bottom right, there is a green 'Create Table' button.

Baze de date - SQL



Rezultatul obtinut este urmatorul:



2. a. Să se introducă 4 înregistrări distincte complete în tabela AN_XY, de tipul:

CodAn DenAn NrGrupe

- 1 Anul I 4
- 2 Anul II 3
- 3 Anul III 4
- 4 Anul IV 3

Ce se întâmplă dacă încercați să introduceți un al 5-lea an:

1 Anul V 2?

Solutie:

Din suboptiunea **Data** se selecteaza butonul **Insert Row** si astfel se adauga, pe rand, fiecare linie (rand) dorita(dorit):



Se apasa pe butonul **Create and Create Another** de cate ori doriti sa introduceti o noua linie in tabela respectiva.

Baze de date - SQL



La final apar urmatoarele informatii:

ZSET	CODAN	DENAN	NRGRUPE
1	Anul I	4	
4	Anul IV	3	
3	Anul III	4	
2	Anul II	3	

Din Meniul principal **SQL Workshop**, optiunea **SQL Commands** se scrie comanda **describe numele tablei** si se apasa butonul **Run**

Rezultatul executiei comenzii este:

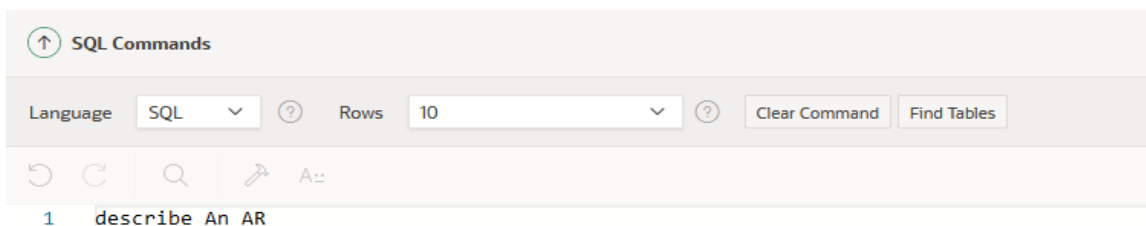


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
AN_AR	CODAN	NUMBER	-	2	0	1	-	-	-
	DENAN	VARCHAR2	10	-	-	-	-	-	-
	NRGRUPE	NUMBER	-	2	0	-	✓	-	-

Baze de date - SQL



Ce se întâmplă dacă încercați să introduceți un al 5-lea an:
1 Anul V 2?

Rezultatul încercării de introducere a unei noi linii care are exact aceeași informație în câmpul (atributul) **CodAn**, este obținerea următoarei erori prin care se precizează faptul că nu se pot introduce informații care se repetă într-un câmp (atribut) de tip **Cheie primară**:

error ORA-00001: unique constraint (WKSP_ORACLESTUDENTIAIA.AN_AR_PK) violated

AN_AR

error ORA-00001: unique constraint (WKSP_ORACLESTUDENTIAIA.AN_AR_PK) violated

Create Row

Table: AN_AR

* CodAn: 1

* DenAn: Anul V

NrGrupa: 2

Cancel Create Create and Create Another

Rezolvati cerintele urmatoare, pe baza solutiei prezentate anterior.

b. Să se introducă 6 înregistrări distincte complete în tabela **GRUPE_XY**, de tipul:

CodGrupa	DenGrupa	NrStudenti	CodAn
1	Grupa nr 1	10	1
2	Grupa nr 2	11	2
3	Grupa nr 3	12	3
4	Grupa nr 4	13	4
5	Grupa nr 5	14	1
6	Grupa finala	15	2

Ce se întâmplă dacă încercați să introduceți a 7-a grupă:

7 Grupa nr 7 16 5 ?

Baze de date - SQL



3. Să se creeze tabelele următoare:

DEPARTAMENT_XY

Cod_dep number(3) cheie primara

Den_dep varchar2(20) nenula

Nr_angajati number(2) default 0

Total_salarii number default 0

ANGAJATI_XY

Matricol number(3) cheie primara

Cod_dep number(3) referinta la tabela **DEPARTAMENT_XY**

Nume varchar2(40)

Salariu number(7)

unde XY sunt inițialele voastre, astfel încât tabelele create de voi să fie distincte între ele

Să se folosească apoi comanda **DESCRIBE** pentru a se vedea tabelele create.

4. Să se introduca înregistrările:

DEPARTAMENT_XY

Cod_dep	Den_dep	Nr_ang	Total_salarii
1	'Depart. VANZARI'	NULL	NULL
2	'Depart. MARKETING'	NULL	NULL
3	'Depart. PRODUCTIE'	NULL	NULL

ANGAJATI_XY

Matricol	Cod_dep	Nume	Salariu
1	1	'Pops J'	100
2	3	'Marc V'	200
3	1	'Johnson T'	150
4	2	'Mathieu R'	400
5	3	'Gerald D'	800