Tema 4. Crearea bazei de date. Modificarea bazei de date. Suprimarea bazei de date.

Instrucțiunea de creare a unei scheme relaționale conține comanda **CREATE TABLE**. Ea trebuie să includă, de asemenea, numele relației, numele de atribute care formează schema relațională, cheia primară, posibilele chei externe şi restricțiile impuse asupra valorilor pe care le pot lua atributele. Orice definiție a unei componente se separă de următoarea prin virgulă.

Sintaxa generală a instrucțiunii este:

CREATE TABLE <nume relație>(

<definiție atribut I> [. <definiție atribut 2>...).

<definiție cheie primara>,

[<definiție cheie externă 1>

[, <definiție cheie externă 2> …],]

[<definiție cheie secundară 1>

[,<definiție cheie secundară 2>...],]

[<definitie constrângere 1>

[,<definiție constrângere 2>...]]);

Ordinea de realizare a definiției este mai flexibilă, cu toate că este evident că înainte de a defini un atribut, ca parte componentă a unei chei primare sau externe, sau de a defini o constrângere asupra valorilor posibile, este necesară definire acestui atribut.

Fie date 2 tabele:

Funcționari	**Nume**	**Prenume**	**Dept ID**
	Bobu	Angela	Dpt 01
	Mutu	Andrei	Dpt 02
	Matei	Ion	Dpt 02
	Paiu	Adrian	Dpt 03
	Bobu	Matei	Dpt 02

Departamente	**Dept ID**	**Denumire**
	Dpt 01	Software
	Dpt 02	Hardware
	Dpt 03	Sistem de operare

Exemplu 1. Să se creeze o schemă relațională, unde atributul **Adresă** este o secvență de caractere de lungime fixă şi ia valoarea implicită "*necunoscută*".

CREATE TABLE funcționari(

Nume VARCHAR(25) NOT NULL,

Prenume VARCHAR(30).

Adresă CHAR(50) **DEFAULT** necunoscută');

În acest exemplu, se forțează un atribut să ia o valoare prin specificarea unei valori implicite cu ajutorul opțiunii *DEFAULT*. Atunci când, în relația funcționari, se va insera un tuplu fără a fi indicată o adresă, sistemul va insera automat valoarea necunoscută'.

Deseori, se imbină constrângerile *UNIQUE* şi *NOT NULL*. In consecința atributul asupra căruia se aplică aceste opțiuni nu acceptă valori repetate şi nici valori nule. Astfel, atributul este definit cheie secundară al schema respective.

Exemplu 2. Se creează o schemă relațională cu o constrangere de comportament al domeniului. Atributul **Data_naştere** este de tip **DATE** și nu ia valori mai mici de *'1980-01-01'* şi mai mari de

'1990-12-31'.

CREATE TABLE funcționari (

Nume VARCHAR(25) NOT NULL,

Prenume VARCHAR(30),

Adresă CHAR(50) DEFAULT 'necunoscuta '

Data naştere DATE

CHECK(*Data_naştere > ’1980-01-01’*

AND Data_naştere<`1990-12-31'));

Trebuie menţionat că SGBD-ul verifică constrângerile definite de fiecare dată când se inserează tupluri noi in relație, se suprimă tupluri sau se modifică valorile unui tuplu. Bineinteles, că această verificare necesita timp, din care motiv randamentul sistemului poate fi negativ afectat, dacă se definesc un numar mare de constrângeri. In sectiunea consacrată constrângerilor, au fost examinate mai detaliat metode mai eficiente de definire a constrângerilor

Exemplu 3. Atributul *ID_funcţionar* este declarat cheie primară a relației funcționari:

CREATE TABLE funcționari(

ID_functionar CHAR(6)

Nume VARCHAR(25) NOT NULL,

Prenume VARCHAR(30) NOT NULL,

Várstа INTEGER CHECK(Värstа>=18

AND Vârsta<30).

PRIMARY KEY (ID_functionar));

De asemenea, se poate specifica că o mulțime de atribute iau valori unice. Aceasta permite, de fapt, declararea *cheilor secundare*. Astfel, aplicând opțiunea **UNIQUE**, se poate indica că doi funcționari nu pot avea acelaşi nume și același prenume. Menționăm că opțiunea **UNIQUE** nu se aplică asupra valorilor necunoscute, adică **NULL.**

Exemplu 4. Mulțimea de atribute *{Nume, Prenume}* este declarată cheie secundară.

CREATE TABLE functionari(

ID_Funcționar CHAR(6)

Nume VARCHAR(25) NOT NULL ,

Prenume VARCHAR(30) NOT NULL,

Vârsta INTEGER CHECK(Vârsta >=18

AND Värsta<30),

PRIMARY KEY (ID fanctionar),

UNIQUE *Nume, Prenume);*

Norma limbajului SQL2 permite declararea cheilor externe ale unei relații. Cu alte cuvinte se indică mulțimea de atribute care, univoc, determină cheia altei relații. Cheia externă se defineste similar cheii primare, indicând clauza **FOREIGN KEY** urmata de lista de atribute ce alcatuiesc cheia externă separate prin virgula, urmata de cuvantul rezervat **REFERENCES** care indică numele relației referite.

Exemplul 5.

Atributul *ID_departament* este declarat cheie externă, care se referă la cheia primară a relaţiei *departamente*. Atributul *Denumire* a relatii departamente este definită cheie externă.

CREATE TABLE *departamente (*

ID departament CHAR(8) PRIMARY KEY,

Denumire VARCHAR(15) UNIQUE NOT NULL):

CREATE TABLE *funcționari (*

ID departament CHAR(6)

Nume VARCHAR(25) NOT NULL

Prenume VARCHAR(30) NOT NULL

Vârsta INTEGER CHECK (Värsta> 18

AND Vârsta< 30),

PRIMARY KEY (ID funcționar),

UNIQUE*(Nume, Prenume),*

FOREGN KEY *(ID departament)*

REFERENCES *departamente);*

De fiecare dată când **SGBD-ul** actualizează baza de date creată de exemplul 5, va verifica dacă nu sunt afectate legăturile dintre relațiile *departamente* şi *funcționari*. Pentru ca definiția să fie corectă, atributele care compun această definiție şi relaţia referită trebuie să existe anterior. De aceea, existenţa cheilor externe restrâng ordinea de creare a schemelor relationale. Adică

îndată ce relația r1, are o cheie externă a relației r2 , ultima trebuie să fie cheie primară.

Adăugarea, modificarea şi suprimarea atributelor

Standardul SQL2 permite *adăugarea, suprimarea şi modificarea* unui stribut din schemă. Cu toate acestea, Oracle, până la versiunea 8.0 inclusiv, permite numai adăugarea şi modificarea atributelor, dar nu şi stergerea. Pentru adăugarea unui atribut, care, de obicei, se adauga la sfârşitul definiţiilor celor existente, se utilizează următoarea sintaxa:

ALTER TABLE <nume relație>

ADD <definiție atribut>;

Exemplul 6. In relația *funcționari*, trebuie inclusă și adresa funcționarului. Următoarea instrucțiune adaugă atributul Adresă in această relație:

ALTER TABLE *funcționari*

ADD *Adresă VARCHAR(50):*

Atributul nou se introduce în schema relației, evident, pentru toate plurile. Tuplurile deja existente in relație vor include valoarea **NULL** pentru atributul adăugat sau valoarea implicită, dacă este utilizată clauza **DEFAULT.** Nu este posibilă adăugarea unui atribut definit cu constrângerea **NOT NULL** și fără valoarea implicită, dacă relaâia conține date.

Exemplu 7. Prima instrucțiune este incorectă, a doua – corectă.

ALTER TABLE *funcționari*

ADD *Adresă VARCHAR(50) NOT NULL,*

ALTER TABLE *funcționari ADD Adresă VARCHAR(50*)

DEFAULT *'*necunoscută*'* **NOT NULL;**

Pentru modificarea unui atribut existent (poate fi modificat doar tipul, nu și numele) se utilizează instrucțiunea:

ALTER TABLE <nume relatie>

MODIFY <nume atribut vechi> <definiție nouă>:

Exemplu 8. Pentru a indica că este nevoie ca dimensiunea atributului *Adresă* să se mărească până la 120 de caractere, se va utiliza următoarea instructiune:

ALTER TABLE *funcţionari*

MODIFY *Adresă VARCHAR(120);*

Sau Modificarea structurii unui table mai poate fi prezentat şi astfel:

ALTER TABLE table_name <opţiuni>

În lista opţiunilor pot fi:

- adăugate coloane noi: ADD *nume_coloana* <parametri>;

```
ALTER TABLE table_name
  ADD column_name <data_type> [ NULL | NOT NULL ]
      [IDENTITY [ ( seed ,increment ) ]
    | [CONSTRAINT constraint_name ] DEFAULT constant_expression ]

    | [CONSTRAINT constraint_name
      PRIMARY KEY
        |UNIQUE
        |FOREIGN KEY REFERENCES referenced_table_name [(ref_column)]
        |CHECK (logical_expression )
```

- adăugate noi constrângeri pentru coloanele existente: ADD WITH sau CONSTRAINT;

```
ALTER TABLE table_name
  ADD CONSTRAINT constraint_name
      PRIMARY KEY | UNIQUE  (column)
    | FOREIGN KEY (column)
          REFERENCES referenced_table_name [(ref_column)]
    | DEFAULT constant_expression FOR column
    |CHECK (logical_expression )
```

- făcute modificări pentru coloanele existente: ALTER COLUMN <parametri>;
- suprimarea unei coloane: DROP COLUMN ;
- suprimarea unei constrângeri:

```
ALTER TABLE table_name
```

DROP CONSTRAINT constraint_name

Pentru aceasta trebuie să ştim numele constrângerii.

Exemple:

```
alter table mytable
  add column5 varchar(100) check (column5<>'petrica')

alter table mytable
  add constraint dv_column5
    default  'gigel' for column5

alter table mytable
  drop constraint dv_column5
```

Suprimarea (lichidarea) unei tabele.

```
drop table mytable
```