

## I. BAZE DE DATE – Noțiuni introductive:

Tipuri de legaturi între tabele:

- **1-1** – legatura dintre cele doua tabele se face pe baza cheilor primare
- **1-n** – cheia primara din tabela “parinte” se adauga in tabela “copil” sub forma de cheie externa
- **n-n** – nu se poate implementa practice si se recurge la adaugarea unei tabele suplimentare care va contine cheile primare ale tabelelor initiale si campuri referitoare la asocierea dintre tabelele “parinti”. Cheia primara din tabela intermediara va fi o cheie compusa.

**Structura bazei de date utilizata ca exemplu la seminar** - Se considera activitatea de evidenta a comenzilor incheiate de o societate comerciala cu diverse firme prin intermediul agentilor angajati in cadrul societatii. Comenzile contin produse aflate in depozitul societatii, iar pentru fiecare produs se cunoaste in permanenta stocul existent.

### FIRME

<u>CODFIRMA</u>	DENFIRMA	LOC	CONTBANCA	ZONA
-----------------	----------	-----	-----------	------

### AGENTI

<u>CODAGENT</u>	NUMEAGENT	DATAANG	DATANAST	ZONA	FUNCTIE	CODSEF
-----------------	-----------	---------	----------	------	---------	--------

### COMENZI

<u>NRCOM</u>	<u>CODFIRMA</u>	<u>CODAGENT</u>	DATA
--------------	-----------------	-----------------	------

### RANDCOM

<u>NRCOM</u>	<u>CODPRODUS</u>	CANT	PRET	TERMENLIVR
--------------	------------------	------	------	------------

### PRODUSE

<u>CODPRODUS</u>	DENPRODUS	UM	STOC
------------------	-----------	----	------

## II. NUCLEUL SQL\*PLUS

Limbajul **SQL – Structured Query Language** este un limbaj standard de interogare a bazelor de date. Implementat în majoritatea SGBD-urilor, dar nu numai. În ORACLE este implementat nucleul SQL cu anumite particularități.

### 1. Tipuri de date:

TIP	Caracteristici
<b>VARCHAR2(size)</b>	sir de caractere de lungime variabila
<b>NUMBER(p,s)</b>	Numeric: p (precision) = 1..38, s (scale) = -84..127
<b>LONG</b>	sir de caractere de lung>2GB
<b>DATE</b>	data calendaristica
<b>RAW(size)</b>	date de tip binar. Maxim size este 2000 bytes.
<b>LONG RAW</b>	lungime variabila > 2GB se stocheaza grafice, sunete, documente
<b>ROWID</b>	adresa fiecarui rind din tabela
<b>CHAR(size)</b>	sir de caractere de lungime fixa
<b>BLOB</b>	binary large object (max 4GB) se stocheaza date nestructurate (text, imagine, video, date spatiale)
<b>BFILE</b>	contine un pointer catre un fisier binar stocat in afara bazei de date

### 2. Operatori SQL

Operator	Caracteristici
<, >, =, >=, <=, NOT	Operatori de comparatie
<b>BETWEEN ... AND ...</b>	între două valori (inclusiv).
<b>IN(listă)</b>	egal cu oricare valoare din listă
<b>LIKE</b>	similar cu un șablon % - oricâte caractere; _ - un caracter;
<b>IS NULL</b>	are valoarea NULL

### 3. Comenzi SQL

Comenzile referitoare la bazele de date sunt comune tuturor versiunilor de SQL și sunt împartite în următoarele categorii:

#### 1) Comenzi LDD (Limbaj de Definire a Datelor - Data Definition Language)

Comanda	Scop
<b>CREATE</b>	Creeaza un obiect nou (tabela, utilizator, rol, etc)
<b>ALTER</b>	Modifica o parte dintre proprietatile unui obiect
<b>DROP</b>	Elimina un obiect din baza de date

**2) Comenzi LMD (Limbaaj de Manipulare a Datelor - Data Manipulation Language)**

<b>Comanda</b>	<b>Scop</b>
<b>SELECT</b>	Regaseste inregistrari in tabele sau vederi.
<b>DELETE</b>	Sterge inregistrari din tabele
<b>INSERT</b>	Adauga o noua inregistrare in tabela
<b>UPDATE</b>	Modifica valorile unor inregistrari din tabele

**3) Comenzi LPT (Limbaaj pentru Procesarea Tranzactiilor)**

<b>Comanda</b>	<b>Scop</b>
<b>COMMIT</b>	Finalizeaza o tranzactie
<b>ROLLBACK</b>	Anuleaza o tranzactie
<b>SAVEPOINT</b>	Defineste un punct de salvare, in locul in care tranzactia salveaza toate operatiile efectuate pana in momentul respective, inainte de a continua urmatorul set de modificari efectuate in baza de date.

**4) Comenzi LCD (Limbaaj de Control al Datelor - Data Control Language)**

<b>Comanda</b>	<b>Scop</b>
<b>GRANT</b>	Acorda utilizatorilor drepturile necesare pentru accesul si manipularea obiectelor din baza de date
<b>REVOKE</b>	Anuleaza anumite drepturi utilizatorilor

## **PARTEA I. Comenzi LDD**

Comanda	Scop
<b>CREATE</b>	Creaza un obiect nou: TABLE, INDEX, CLUSTER, TABLESPACE, SEQUENCE, VIEW, MATERIALIZED VIEW, USER, ROLE, PROCEDURE, FUNCTION, TRIGGER
<b>ALTER</b>	Modifica o parte dintre proprietatile unui obiect
<b>DROP</b>	Sterge un obiect din baza de date

## **CREAREA SI GESTIUNEA TABELELOR**

### **1.1. CREAREA TABELELOR – COMANDA *CREATE TABLE***

Pentru a crea o tabela se specifica urmatoarele:

- definitiile atributelor
- modul de organizare a tabeli
- restrictiile de integritate
- spatiul de tabela
- caracteristicile de stocare
- datele daca se preiau dintr-o alta tabela

#### ***RESTRICTII DE INTEGRITATE***

- Tipuri de restrictii:
  - o NOT NULL
  - o UNIQUE
  - o PRIMARY KEY
  - o FOREIGN
  - o CHECK
- Precizarea restrictiilor se poate face in 2 moduri:
  - *in-line* la definirea coloanei pe care se impune restrictia si
  - *out-of-line* separat, fie la sfarsitul sintaxei create table, fie prin comanda alter

**Exemplu – crearea tabelelor pentru aplicatia utilizata la seminar.**

```
DROP TABLE FIRME CASCADE CONSTRAINTS;  
DROP TABLE AGENTI CASCADE CONSTRAINTS;  
DROP TABLE COMENZI CASCADE CONSTRAINTS;  
DROP TABLE RINDCOM CASCADE CONSTRAINTS;  
DROP TABLE PRODUSE CASCADE CONSTRAINTS;
```

```
create table firme  
(codfirma number(2) constraint PKey_firme primary key,  
denfirma varchar2(20) not null,  
loc varchar2(20),  
contbanca varchar(15),
```

```
zona varchar2(15) CONSTRAINT FZONA_CK check (zona in  
( 'MOLDOVA','ARDEAL','BANAT','MUNTENIA','DOBROGEA','TRANSILVANIA')));
```

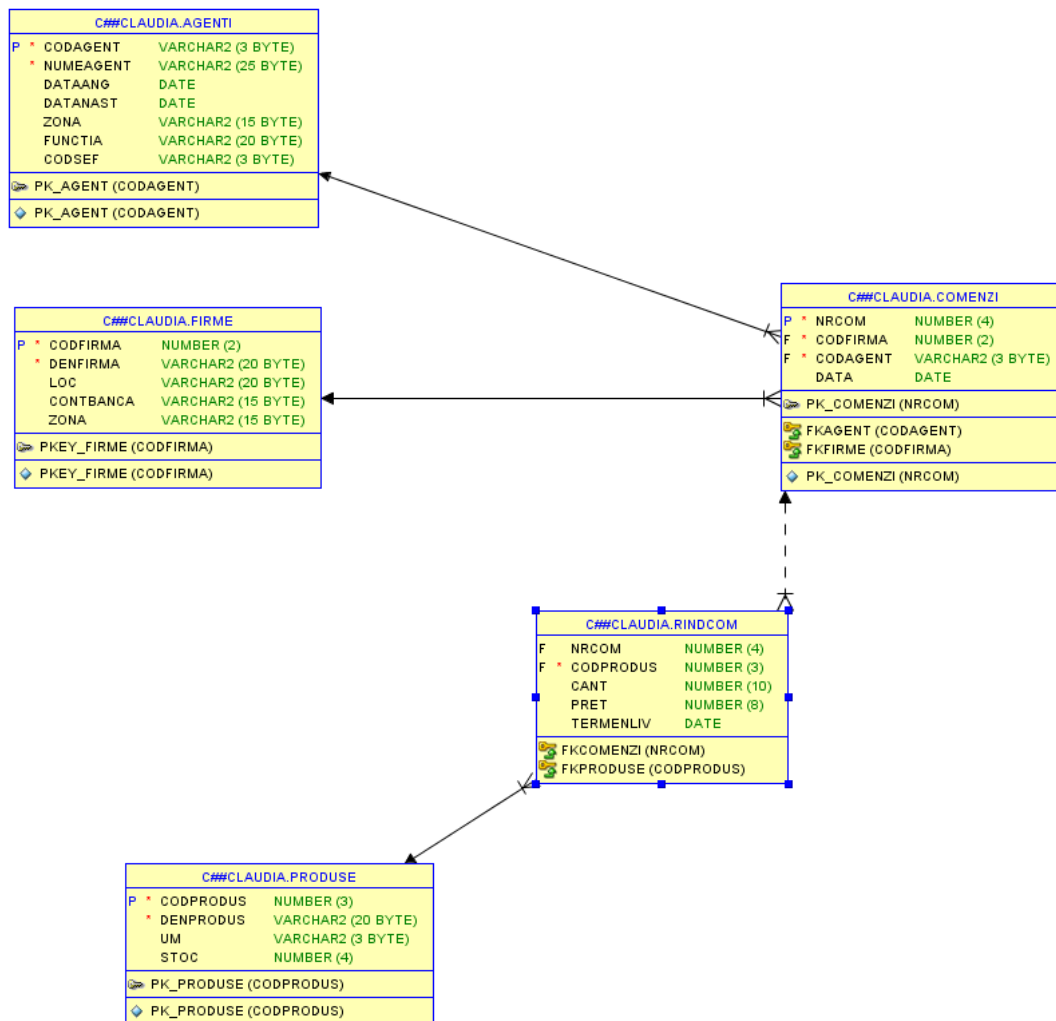
```
create table agenti  
(codagent varchar2(3) constraint pk_agent primary key,  
numeagent varchar2(25) not null,  
dataang date default sysdate,  
datanast date,  
zona varchar2(15) CONSTRAINT AGZONA_CK check (zona  
in('MOLDOVA','ARDEAL','BANAT','MUNTENIA','DOBROGEA','TRANSILVANIA')),  
functia varchar2(20),  
codsef varchar2(3));
```

```
create table comenzi  
(nrcom number(4) constraint pk_comenzi primary key,  
codfirma number(2) not null,  
codagent varchar2(3) not null,  
data date default sysdate,  
CONSTRAINT FKAgent FOREIGN KEY (codagent) REFERENCES agenti(codagent),  
CONSTRAINT FKFirme FOREIGN KEY (codfirma) REFERENCES firme(codfirma));
```

```
create table produse  
(codprodus number(3) constraint pk_produse primary key,  
denprodus varchar2(20) not null,  
um varchar2(3),  
stoc number(4));
```

```
create table rindcom  
(nrcom number(4),  
codprodus number(3) not null,  
cant number(10),  
pret number(8),  
termenliv date,  
CONSTRAINT FKComenzi FOREIGN KEY (nrcom) REFERENCES comenzi(nrcom),  
CONSTRAINT FKProduse FOREIGN KEY (codprodus) REFERENCES  
produse(codprodus));
```

Creati tabele in SQLDeveloper si generati apoi ER diagram.



Relational 1

### Exercitiu :

Alegeti un domeniu si stabiliti entitatile si relatiile dintre acestea. Trasformati entitatile si relatiile in tabele si chei primare, respectiv straine si construti ER diagram.

### CREAREA UNEI TABELE PE BAZA CAMPURILOR DIN ALTA TABELA:

```

CREATE TABLE nume_tabela
AS
SELECT [* , nume campuri] FROM nume_tabela_sursa
WHERE [conditie];
    
```

**Exemplu:** Tabela firme\_buc va contine firmele din Bucuresti

```

CREATE TABLE FIRME_BUC
AS
SELECT * FROM FIRME
WHERE LOC='BUCURESTI';
    
```

## ***1.2. MODIFICAREA STRUCTURII TABELELOR - COMANDA ALTER***

Realizeaza urmatoarele:

- Modificarea structurii tabeli: ADD, MODIFY, DROP COLUMN, SET UNUSED
- Modificarea restrictiilor de integritate: ADD, MODIFY, DROP, DISABLE CONSTRAINT
- Redenumeste tabela: RENAME

**Exemple - dupa fiecare comanda vizualizati structura tabeli cu DESCRIBE sau DESC:**

```
ALTER TABLE agenti RENAME TO personal;
```

Sau

```
RENAME AGENTI TO PERSONAL;
```

```
ALTER TABLE PERSONAL  
ADD (EMAIL VARCHAR2(10),  
VARSTA NUMBER(2));
```

```
ALTER TABLE PERSONAL  
MODIFY (EMAIL VARCHAR2(30));
```

```
ALTER TABLE PERSONAL  
DROP COLUMN EMAIL;
```

```
ALTER TABLE PERSONAL  
SET UNUSED COLUMN FUNCTIA;
```

### Logical Delete

On large tables the process of physically removing a column can be very time and resource consuming. For this reason you may decide to logically delete it.

```
ALTER TABLE table_name SET UNUSED (column_name);  
ALTER TABLE table_name SET UNUSED (column_name1, column_name2);
```

Once this is done the columns will no longer be visible to the user. If at a later date you have time to physically delete the columns this can be done using the following.

```
ALTER TABLE PERSONAL  
DROP UNUSED COLUMNS;
```

```
ALTER TABLE PERSONAL
```

```
ADD (CONSTRAINT check_varsta CHECK (varsta>18 and varsta<60));
```

```
ALTER TABLE PERSONAL  
DISABLE CONSTRAINT check_varsta;
```

```
ALTER TABLE PERSONAL  
DROP CONSTRAINT check_varsta;
```

### ***1.3. STERGEREA TABELELOR – COMANDA DROP***

DROP permite stergerea unei tabele.

```
DROP TABLE PERSONAL;  
SAU  
DROP TABLE PERSONAL CASCADE CONSTRAINTS;
```

### **VIZUALIZAREA OBIECTELOR CE APARTIN UNUI ANUMIT UTILIZATOR:**

```
SELECT * FROM USER_TABLES;  
SELECT TABLE_NAME, cluster_name FROM USER_TABLES;
```

```
SELECT * FROM USER_CATALOG;
```

```
SELECT DISTINCT OBJECT_TYPE  
FROM USER_OBJECTS;
```

```
SELECT TABLE_NAME, CONSTRAINT_TYPE, CONSTRAINT_NAME  
FROM USER_CONSTRAINTS;
```

```
SELECT TABLE_NAME , COMMENTS  
FROM USER_TAB_COMMENTS;
```

-pentru a defini comments :

```
comment on table nume_tabel IS 'Tabelul cu carti';  
comment on table books IS 'Tabelul cu carti';
```