**ACADEMIA DE STUDII ECONOMICE DIN BUCUREŞTI**

**FACULTATEA DE CIBERNETICĂ, STATISTICĂ ȘI INFORMATICĂ ECONOMICĂ**



**PROIECT ORACLE**

**„** **Baza de date a unui restaurant”**

Chelaru Cristian-Antonio

Grupa 1099, seria F

Prof. Ene Gabriela

București 2023

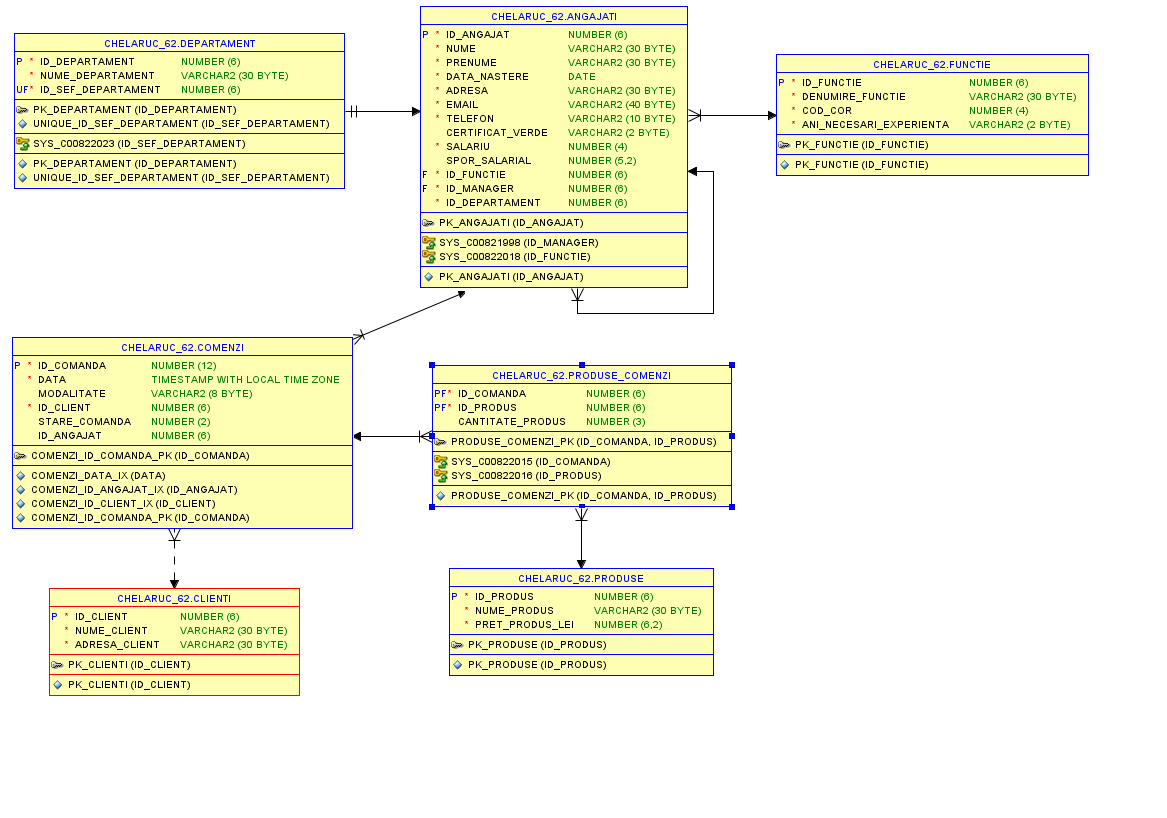
**Cuprins**

1. Descrierea bazei de date..................................................3
2. Schema bazei de date.......................................................3
3. Relațiile dintre tabele.......................................................4
4. Crearea tabelelor..............................................................5
5. Popularea tabelelor (actualizare, adăugări)....................11
6. Exemple de interogări variate........................................26
7. Gestiunea altor obiecte ale bazei de date.......................35
8. Descrierea bazei de date

Tema proiectului este „Proiectarea și implementarea unei baze de date pentru gestiunea activității unui restaurant”. Se consideră activitatea de evidență a funcționării unui restaurant.

Activitatea unui restaurant este dependentă de un personal bine organizat și pregătit. De aceea, managementul unui restaurant trebuie să fie clar, astfel încât fiecare persoană să-și cunoască misiunea în cadrul societății. Pentru a face acest lucru, definirea funcțiilor și a departamentelor este critică în succesul companiei. Totodată, stabilirea clară a unor preparate culinare însoțite de o procedură standard, lipsită de ambiguitate, este, de asemenea, importantă.

1. Schema bazei de date



1. Relațiile dintre tabele
2. Relația dintre tabelul „Angajați” și:
3. Tabelul „Funcție”

* Many to one (deoarece o funcție poate fi ocupată de mai mulți angajați, dar un angajat poate avea o singură funcție)

1. Tabelul „Departament”

* Many to many (deoarece mai mulți angajați pot face parte dintr-un departament, iar un angajat poate fi șeful mai multor departamente)

1. Tabelul „Angajați”

* Many to one (doarece un angajat poate avea un singur manager, dar un angajat-manager poate avea mai mulți angajați în subordine)

1. Tabelul „Comenzi”

* Many to one (deoarece o comandă poate să fie pregătită de mai mulți angajați, dar un angajat poate pregăti doar o singură comandă)

1. Relația dintre tabelul „Comenzi” și:
2. Tabelul „Produse”

* Many to many (deoarece un produs poate fi în mai multe comenzi, iar o comandă poate să aibă mai multe produse)

1. Tabelul „Clienți

* Many to one (deoarece un client poate plasa mai multe comenzi, dar o comandă poate să aibă atribuit un singur client)

1. Crearea tabelelor

Momentan, pentru realizarea activității cu succes a întreprinderii, am creat patru tabele cu câmpurile aferente acestora:

1. Tabelul „Produse”

Motivul principal pentru care restaurantele sunt profitabile este reprezentat de prepatele gustoase ale acestora:

|  |  |  |
| --- | --- | --- |
| **Nume coloană** | **Restricție** | **Tip de date** |
| id\_produs | PRIMARY KEY | NUMBER(6) |
| nume\_produs | CHECK – NOT NULL | VARCHAR(30) |
| pret\_produs\_lei | CHECK (> 0) | NUMBER(6, 2) |

Acest tabel poate fi creat cu ușurință folosind următoarele comenzi:

DROP TABLE PRODUSE CASCADE CONSTRAINTS;

CREATE TABLE PRODUSE(

id\_produs NUMBER(6) CONSTRAINT pk\_produse PRIMARY KEY,

nume\_produs VARCHAR(30) NOT NULL,

pret\_produs\_lei NUMBER(6, 2) CONSTRAINT ck\_pret\_produs\_lei CHECK (pret\_produs\_lei > 0) NOT NULL

);

1. Tabelul „Angajați”

Restaurantul fără un personal dedicat și bine pregătit nu poate exista, de aceea este primul tabel la care m-am gândit:

|  |  |  |
| --- | --- | --- |
| **Nume coloană** | **Restricție** | **Tip de date** |
| id\_angajat | PRIMARY KEY | NUMBER(6) |
| nume | CHECK – NOT NULL | VARCHAR(30) |
| prenume | CHECK – NOT NULL | VARCHAR(30) |
| data\_nastere | CHECK - NOT NULL | DATE |
| adresa | CHECK – NOT NULL | VARCHAR(30) |
| email | CHECK – NOT NULL | VARCHAR(40) |
| telefon | CHECK – NOT NULL | VARCHAR(10) |
| certificat\_verde | CHECK – („DA” / „NU”) | VARCHAR(2) |
| salariu | CHECK ( >= 1386) | NUMBER(4) |
| spor\_salarial | CHECK (between 0 and 100) | DECIMAL(5, 2) |
| id\_functie | FOREIGN KEY | NUMBER(6) |
| id\_manager | FOREIGN KEY | NUMBER(6) |
| id\_departament | FOREIGN KEY | NUMBER(6) |

Acest tabel poate fi creat cu ușurință folosind următoarele comenzi:

DROP TABLE ANGAJATI CASCADE CONSTRAINTS;

CREATE TABLE ANGAJATI(

id\_angajat NUMBER(6) CONSTRAINT pk\_angajati PRIMARY KEY,

nume VARCHAR(30) NOT NULL,

prenume VARCHAR(30) NOT NULL,

data\_nastere DATE NOT NULL,

adresa VARCHAR(30) NOT NULL,

email VARCHAR(40) NOT NULL,

telefon VARCHAR(10) NOT NULL,

certificat\_verde VARCHAR(2) CONSTRAINT ck\_certificat\_verde CHECK (upper(certificat\_verde) IN (upper('Da'), upper('Nu'))),

salariu NUMBER(4) CONSTRAINT ck\_salariu CHECK (salariu >= 1386) NOT NULL,

spor\_salarial DECIMAL(5, 2) CONSTRAINT ck\_spor\_salarial CHECK (spor\_salarial BETWEEN 0 AND 100),

id\_functie NUMBER(6) NOT NULL,

id\_manager NUMBER(6) NOT NULL,

id\_departament NUMBER(6) NOT NULL

);

ALTER TABLE ANGAJATI

MODIFY id\_manager REFERENCES ANGAJATI(id\_angajat);

ALTER TABLE ANGAJATI

MODIFY id\_functie REFERENCES FUNCTIE(id\_functie);

ALTER TABLE ANGAJATI

MODIFY id\_departament REFERENCES DEPARTAMENT(id\_departament);

1. Tabelul „Departamente”

Pentru o organizare bună, fiecare angajat trebuie să facă parte dintr-un departament care are o misiune bine definită:

|  |  |  |
| --- | --- | --- |
| **Nume coloană** | **Restricție** | **Tip de date** |
| id\_departament | PRIMARY KEY | NUMBER(6) |
| nume\_departament | CHECK – NOT NULL | VARCHAR(30) |
| id\_sef\_departament | FOREIGN KEY | NUMBER(6) |

Acest tabel poate fi creat cu ușurință folosind următoarele comenzi:

DROP TABLE DEPARTAMENT CASCADE CONSTRAINTS;

CREATE TABLE DEPARTAMENT(

id\_departament NUMBER(6) CONSTRAINT pk\_departament PRIMARY KEY,

nume\_departament VARCHAR(30) NOT NULL,

id\_sef\_departament NUMBER(6) REFERENCES ANGAJATI(id\_angajat) NOT NULL CONSTRAINT unique\_id\_sef\_departament UNIQUE

);

1. Tabelul „Funție”

În cadrul fiecărui departament, fiecare angajat are atribuții diferite, de aceea am creat acest tabel, pentru a crește transparența și claritatea:

|  |  |  |
| --- | --- | --- |
| **Nume coloană** | **Restricție** | **Tip de date** |
| id\_functie | PRIMARY KEY | NUMBER(6) |
| denumire\_functie | CHECK – NOT NULL | VARCHAR(30) |
| cod\_cor | CHECK (dimensiune = 4) | NUMBER(4) |
| ani\_necesari\_experienta | CHECK - NOT NULL | VARCHAR(2) |

Acest tabel poate fi creat cu ușurință folosind următoarele comenzi:

DROP TABLE FUNCTIE CASCADE CONSTRAINTS;

CREATE TABLE FUNCTIE(

id\_functie NUMBER(6) CONSTRAINT pk\_functie PRIMARY KEY,

denumire\_functie VARCHAR(30) NOT NULL,

cod\_cor NUMBER(4) CONSTRAINT ck\_cod\_cor CHECK (length(cod\_cor) = 4) NOT NULL,

ani\_necesari\_experienta VARCHAR(2) NOT NULL

);

1. Tabelul „Clienți”

Fiecare restaurant are nevoie de clienți, pe aceștia o să-i salvăm pentru a putea să le procesăm comenzile:

|  |  |  |
| --- | --- | --- |
| **Nume coloană** | **Restricție** | **Tip de date** |
| id\_client | PRIMARY KEY | NUMBER(6) |
| nume\_client | CHECK - NOT NULL | VARCHAR(30) |
| adresa\_client | CHECK - NOT NULL | VARCHAR(30) |

Acest tabel poate fi creat cu ușurință folosind următoarele comenzi:

DROP TABLE CLIENTI CASCADE CONSTRAINTS;

CREATE TABLE CLIENTI (

id\_client NUMBER(6) CONSTRAINT pk\_clienti PRIMARY KEY,

nume\_client VARCHAR(30) NOT NULL,

adresa\_client VARCHAR(30) NOT NULL

);

1. Tabelul „Comenzi”

Restaurantul pentru a produce bani, are nevoie să-și vândă produsele, de aceea avem nevoie de acest tabel:

|  |  |  |
| --- | --- | --- |
| **Nume coloană** | **Restricție** | **Tip de date** |
| id\_comanda | PRIMARY KEY | NUMBER(6) |
| id\_client | FOREIGN KEY | VARCHAR(30) |
| id\_angajat | FOREIGN KEY | VARCHAR(3) |
| data\_plasare\_comanda | CHECK – NOT NULL | DATE |
| data\_finalizare\_comanda | - | DATE |

Acest tabel poate fi creat cu ușurință folosind următoarele comenzi:

DROP TABLE COMENZI CASCADE CONSTRAINTS;

CREATE TABLE COMENZI (

id\_comanda NUMBER(6) CONSTRAINT pk\_comenzi PRIMARY KEY,  
id\_client NUMBER(6) CONSTRAINT fk\_id\_client REFERENCES CLIENTI(id\_client),  
id\_angajat NUMBER(6) CONSTRAINT fk\_id\_angajat REFERENCES ANGAJATI(id\_angajat),  
data\_plasare\_comanda DATE NOT NULL,  
data\_finalizare\_comanda DATE);

1. Tabelul „Produse\_Comenzi”

Pentru a ști ce produse conține fiecare comandă ne vom ajuta de acest tabel:

|  |  |  |
| --- | --- | --- |
| **Nume coloană** | **Restricție** | **Tip de date** |
| id\_comanda | PRIMARY KEY | NUMBER(6) |
| id\_produs | PRIMARY KEY | NUMBER(6) |
| cantitate\_produs | CHECK > 0 | NUMBER(3) |

Acest tabel poate fi creat cu ușurință folosind următoarele comenzi:

DROP TABLE PRODUSE\_COMENZI CASCADE CONSTRAINTS;

CREATE TABLE PRODUSE\_COMENZI (

id\_comanda NUMBER(6) REFERENCES COMENZI(id\_comanda),

id\_produs NUMBER(6) REFERENCES PRODUSE(id\_produs),

cantitate\_produs NUMBER(3) CONSTRAINT ck\_cantitate\_produs CHECK (cantitate\_produs > 0)

);

ALTER TABLE PRODUSE\_COMENZI

ADD CONSTRAINT produse\_comenzi\_pk PRIMARY KEY(id\_comanda, id\_produs);

1. Popularea tabelelor (actualizare, adăugări)
2. Popularea tabelei „Produse”

INSERT INTO PRODUSE (id\_produs, nume\_produs, pret\_produs\_lei)   
VALUES (1, 'Omleta', 15);

INSERT INTO PRODUSE (id\_produs, nume\_produs, pret\_produs\_lei)   
VALUES (2, 'Paste carbonara', 25);

INSERT INTO PRODUSE (id\_produs, nume\_produs, pret\_produs\_lei)   
VALUES (3, 'Salata greceasca', 25);

INSERT INTO PRODUSE (id\_produs, nume\_produs, pret\_produs\_lei)   
VALUES (4, 'Ciorba de legume', 15);

INSERT INTO PRODUSE (id\_produs, nume\_produs, pret\_produs\_lei)   
VALUES (5, 'Piept de pui', 31);

INSERT INTO PRODUSE (id\_produs, nume\_produs, pret\_produs\_lei)   
VALUES (6, 'Ceafa de porc', 32);

INSERT INTO PRODUSE (id\_produs, nume\_produs, pret\_produs\_lei)   
VALUES (7, 'Coaste de porc', 35);

INSERT INTO PRODUSE (id\_produs, nume\_produs, pret\_produs\_lei)   
VALUES (8, 'Snitel de pui', 30);

INSERT INTO PRODUSE (id\_produs, nume\_produs, pret\_produs\_lei)   
VALUES (9, 'Somon la gratar', 46);

INSERT INTO PRODUSE (id\_produs, nume\_produs, pret\_produs\_lei)   
VALUES (10, 'Pastrav' , 31);

INSERT INTO PRODUSE (id\_produs, nume\_produs, pret\_produs\_lei)   
VALUES (11, 'Aripioare de pui', 28);

INSERT INTO PRODUSE (id\_produs, nume\_produs, pret\_produs\_lei)   
VALUES (12, 'Papanasi', 13);

INSERT INTO PRODUSE (id\_produs, nume\_produs, pret\_produs\_lei)   
VALUES (13, 'Clatite', 11);

1. Popularea tabelei „Angajați”:

INSERT INTO ANGAJATI (id\_angajat, nume, prenume, data\_nastere, adresa, email, telefon, certificat\_verde, salariu, spor\_salarial, id\_functie, id\_manager, id\_departament)  
VALUES (1, 'Chelaru', 'Cristian-Antonio', to\_date('30.07.2002', 'dd.mm.yyyy'), 'Strada Primaverii', 'toni.chelaru@gmail.com', '0786450450', 'da', '9000', null, 1, 1, 1);

INSERT INTO ANGAJATI (id\_angajat, nume, prenume, data\_nastere, adresa, email, telefon, certificat\_verde, salariu, spor\_salarial, id\_functie, id\_manager, id\_departament)  
VALUES (2, 'Badic', 'Mihai', to\_date('20.09.2002', 'dd.mm.yyyy'), 'Strada Verii', 'badic\_mihai@gmail.com', '0738435600', 'nu', '8300', null, 2, 1, 1);

INSERT INTO ANGAJATI (id\_angajat, nume, prenume, data\_nastere, adresa, email, telefon, certificat\_verde, salariu, spor\_salarial, id\_functie, id\_manager, id\_departament)  
VALUES (3, 'Coatu', 'Sorin', to\_date('18.08.2001', 'dd.mm.yyyy'), 'Strada Iernii', 'coatu\_sorin@gmail.com', '0744843023', 'da', '5400', 5.50, 3, 2, 2);

INSERT INTO ANGAJATI (id\_angajat, nume, prenume, data\_nastere, adresa, email, telefon, certificat\_verde, salariu, spor\_salarial, id\_functie, id\_manager, id\_departament)  
VALUES (4, 'Craciun', 'Ionut', to\_date('29.11.2001', 'dd.mm.yyyy'), 'Strada Anotimpurilor', 'craciunionut20@gmail.com', '0794300123', 'da', '5300', 6.01, 3, 2, 2);

INSERT INTO ANGAJATI (id\_angajat, nume, prenume, data\_nastere, adresa, email, telefon, certificat\_verde, salariu, spor\_salarial, id\_functie, id\_manager, id\_departament)  
VALUES (5, 'Cozma', 'Aurel', to\_date('15.10.2001', 'dd.mm.yyyy'), 'Strada Toamnei', 'cozma\_aurel@gmail.com', '0788009913', 'da', '3200', 10.05, 4, 3, 3);

INSERT INTO ANGAJATI (id\_angajat, nume, prenume, data\_nastere, adresa, email, telefon, certificat\_verde, salariu, spor\_salarial, id\_functie, id\_manager, id\_departament)  
VALUES (6, 'Leonida', 'Tudor', to\_date('22.01.2002', 'dd.mm.yyyy'), 'Strada Soarelui', 'leonida.tudor@gmail.com', '0776699334', 'da', '3100', null, 4, 3, 3);

INSERT INTO ANGAJATI (id\_angajat, nume, prenume, data\_nastere, adresa, email, telefon, certificat\_verde, salariu, spor\_salarial, id\_functie, id\_manager, id\_departament)  
VALUES (7, 'Cristescu', 'George', to\_date('23.06.2001', 'dd.mm.yyyy'), 'Strada Noptii', 'cristescugeorge23@gmail.com', '0793868727', 'nu', '3150', 1.02, 4, 4, 3);

INSERT INTO ANGAJATI (id\_angajat, nume, prenume, data\_nastere, adresa, email, telefon, certificat\_verde, salariu, spor\_salarial, id\_functie, id\_manager, id\_departament)  
VALUES (8, 'Macovei', 'Gheorghe', to\_date('12.06.2003', 'dd.mm.yyyy'), 'Strada Zilei', 'gheorghe.macovei@gmail.com', '0701315632', 'da', '3175', null, 4, 4, 3);

INSERT INTO ANGAJATI (id\_angajat, nume, prenume, data\_nastere, adresa, email, telefon, certificat\_verde, salariu, spor\_salarial, id\_functie, id\_manager, id\_departament)  
VALUES (9, 'Marinescu', 'Larisa', to\_date('12.05.2002', 'dd.mm.yyyy'), 'Strada Lunii', 'larisa\_lari@gmail.com', '0780913567', 'nu', '2200', null, 5, 5, 3);

INSERT INTO ANGAJATI (id\_angajat, nume, prenume, data\_nastere, adresa, email, telefon, certificat\_verde, salariu, spor\_salarial, id\_functie, id\_manager, id\_departament)  
VALUES (10, 'Coman', 'Claudia', to\_date('26.02.2001', 'dd.mm.yyyy'), 'Strada Anului', 'claudiaco20@gmail.com', '0741347425', 'nu', '2200', 4.20, 5, 6, 3);

INSERT INTO ANGAJATI (id\_angajat, nume, prenume, data\_nastere, adresa, email, telefon, certificat\_verde, salariu, spor\_salarial, id\_functie, id\_manager, id\_departament)  
VALUES (11, 'Popescu', 'Ana', to\_date('26.10.2003', 'dd.mm.yyyy'), 'Strada Zilei', 'anapop20@gmail.com', '0743674251', 'da', '5500', 7.05, 6, 2, 4);

INSERT INTO ANGAJATI (id\_angajat, nume, prenume, data\_nastere, adresa, email, telefon, certificat\_verde, salariu, spor\_salarial, id\_functie, id\_manager, id\_departament)  
VALUES (12, 'Credu', 'Costin', to\_date('21.09.2001', 'dd.mm.yyyy'), 'Strada Telefon 1', 'credu\_costin@gmail.com', '0712345678', 'da', '5400', null, 9, 2, 5);

INSERT INTO ANGAJATI (id\_angajat, nume, prenume, data\_nastere, adresa, email, telefon, certificat\_verde, salariu, spor\_salarial, id\_functie, id\_manager, id\_departament)  
VALUES (13, 'Aciu', 'Elena', to\_date('11.09.2001', 'dd.mm.yyyy'), 'Strada Calarasi 1', 'elena\_ac@gmail.com', '0712345678', 'da', '5450', null, 12, 2, 6);

INSERT INTO ANGAJATI (id\_angajat, nume, prenume, data\_nastere, adresa, email, telefon, certificat\_verde, salariu, spor\_salarial, id\_functie, id\_manager, id\_departament)  
VALUES (14, 'Dinu', 'Maria', to\_date('14.03.2001', 'dd.mm.yyyy'), 'Strada Dubai 1', 'maria\_dinu@gmail.com', '0712345678', 'da', '5300', null, 15, 2, 7);

INSERT INTO ANGAJATI (id\_angajat, nume, prenume, data\_nastere, adresa, email, telefon, certificat\_verde, salariu, spor\_salarial, id\_functie, id\_manager, id\_departament)  
VALUES (15, 'Stoileru', 'Valentina', to\_date('22.01.2001', 'dd.mm.yyyy'), 'Strada Busteni 1', 'valentina\_s@gmail.com', '0712345678', 'da', '5505', 14.05, 18, 2, 8);

INSERT INTO ANGAJATI (id\_angajat, nume, prenume, data\_nastere, adresa, email, telefon, certificat\_verde, salariu, spor\_salarial, id\_functie, id\_manager, id\_departament)  
VALUES (16, 'Mihai', 'Catalin', to\_date('12.07.2001', 'dd.mm.yyyy'), 'Strada Grecia 1', 'mihai.catalin@gmail.com', '0712345678', 'da', '5500', 12.05, 21, 2, 9);

INSERT INTO ANGAJATI (id\_angajat, nume, prenume, data\_nastere, adresa, email, telefon, certificat\_verde, salariu, spor\_salarial, id\_functie, id\_manager, id\_departament)  
VALUES (17, 'Constantin', 'Andrei', to\_date('13.08.2003', 'dd.mm.yyyy'), 'Strada Revelionului', 'constanti\_andrei@gmail.com', '0712346587', 'da', '5055', null, 24, 2, 10);

1. Popularea tabelei „Departament”:

INSERT INTO DEPARTAMENT (id\_departament, nume\_departament, id\_sef\_departament)  
VALUES(1, 'Conducere', 1);

INSERT INTO DEPARTAMENT (id\_departament, nume\_departament, id\_sef\_departament)  
VALUES(2, 'Manageri', 2);

INSERT INTO DEPARTAMENT (id\_departament, nume\_departament, id\_sef\_departament)  
VALUES(3, 'Bucatari', 3);

INSERT INTO DEPARTAMENT (id\_departament, nume\_departament, id\_sef\_departament)  
VALUES(4, 'Ospatari', 11);

INSERT INTO DEPARTAMENT (id\_departament, nume\_departament, id\_sef\_departament)  
VALUES(5, 'Salubrizare', 12);

INSERT INTO DEPARTAMENT (id\_departament, nume\_departament, id\_sef\_departament)  
VALUES(6, 'Aparare', 13);

INSERT INTO DEPARTAMENT (id\_departament, nume\_departament, id\_sef\_departament)  
VALUES(7, 'Contabilitate', 14);

INSERT INTO DEPARTAMENT (id\_departament, nume\_departament, id\_sef\_departament)  
VALUES(8, 'Promovare', 15);

INSERT INTO DEPARTAMENT (id\_departament, nume\_departament, id\_sef\_departament)  
VALUES(9, 'Resurse umane', 16);

INSERT INTO DEPARTAMENT (id\_departament, nume\_departament, id\_sef\_departament)  
VALUES(10, 'IT', 17);

1. Popularea tabelei „Funcții”:

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta)   
VALUES (1, 'Director', 1120 , 20);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (2, 'Manager', 1120 , 10);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (3, 'Sef-Bucatar', 3434 , 5);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (4, 'Bucatar', 5120 , 2);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (5, 'Ajutor-Bucatar', 9411 , 0);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (6, 'Chelner-sef', 5131, 5);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (7, 'Chelner', 5131, 2);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (8, 'Ajutor-chelner', 5131, 0);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (9, 'Muncifor sef salubritate', 9611, 5);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (10, 'Muncifor salubritate', 9611, 2);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (11, 'Ajutor muncifor salubritate', 9611, 0);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (12, 'Paznic-sef', 5414, 5);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (13, 'Paznic', 5414, 2);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (14, 'Ajutor-paznic', 5414, 0);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (15, 'Contabil-Sef', 1211, 5);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (16, 'Contabil', 3313, 2);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (17, 'Ajutor contabil', 2411, 0);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (18, 'Agent de publicitate sef', 1222, 5);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (19, 'Agent de publicitate', 2431, 2);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (20, 'Ajutor agent de publicitate', 2431, 0);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (21, 'Agent resurse umane sef', 2423, 5);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (22, 'Agent resurse umane', 2423, 2);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (23, 'Ajutor agent resurse umane', 2423, 0);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (24, 'Programator sef', 2512, 5);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (25, 'Programator', 2512, 2);

INSERT INTO FUNCTIE (id\_functie, denumire\_functie, cod\_cor, ani\_necesari\_experienta) VALUES (26, 'Ajutor programator', 2512, 0);

1. Popularea tabelei „Clienți”

INSERT INTO CLIENTI (id\_client, nume\_client, adresa\_client)   
VALUES (1, 'Simion Emil', 'Drumul Invingatorilor');

INSERT INTO CLIENTI (id\_client, nume\_client, adresa\_client)   
VALUES (2, 'Voicu Violeta', 'Aleea Reusitei');

INSERT INTO CLIENTI (id\_client, nume\_client, adresa\_client)   
VALUES (3, 'Popov Petrica', 'Strada Luptelor');

INSERT INTO CLIENTI (id\_client, nume\_client, adresa\_client)   
VALUES (4, 'Mihai Constatin', 'Strada Luminii');

INSERT INTO CLIENTI (id\_client, nume\_client, adresa\_client)   
VALUES (5, 'Petre Ionut', 'Strada Intunericului');

INSERT INTO CLIENTI (id\_client, nume\_client, adresa\_client)   
VALUES (6, 'Popa Nicu', 'Strada Razelor');

INSERT INTO CLIENTI (id\_client, nume\_client, adresa\_client)   
VALUES (7, 'Pana Gheorghe', 'Strada Curcubeului');

INSERT INTO CLIENTI (id\_client, nume\_client, adresa\_client)   
VALUES (8, 'Duma Denisa', 'Strada Perfectiunii');

INSERT INTO CLIENTI (id\_client, nume\_client, adresa\_client)   
VALUES (9, 'Balan Ionut', 'Strada Imperfectiunii');

INSERT INTO CLIENTI (id\_client, nume\_client, adresa\_client)   
VALUES (10, 'Helerea Elena', 'Strada Perfectului Imperfect');

1. Popularea tabelei „Comenzi”

INSERT INTO COMENZI (id\_comanda, id\_client, id\_angajat, data\_plasare\_comanda, data\_finalizare\_comanda)   
VALUES (1, 4, 5, TO\_DATE('15.10.2021', 'DD-MM-YYYY'), TO\_DATE('16.10.2021', 'DD-MM-YYYY'));

INSERT INTO COMENZI (id\_comanda, id\_client, id\_angajat, data\_plasare\_comanda, data\_finalizare\_comanda)   
VALUES (2, 9, 6, TO\_DATE('15.10.2021', 'DD-MM-YYYY'), TO\_DATE('15.10.2021', 'DD-MM-YYYY'));

INSERT INTO COMENZI (id\_comanda, id\_client, id\_angajat, data\_plasare\_comanda, data\_finalizare\_comanda)  
VALUES (3, 10, 9, TO\_DATE('16.10.2021', 'DD-MM-YYYY'), TO\_DATE('17.10.2021', 'DD-MM-YYYY'));

INSERT INTO COMENZI (id\_comanda, id\_client, id\_angajat, data\_plasare\_comanda, data\_finalizare\_comanda)  
VALUES (4, 1, 10, TO\_DATE('16.10.2021', 'DD-MM-YYYY'), TO\_DATE('19.10.2021', 'DD-MM-YYYY'));

INSERT INTO COMENZI (id\_comanda, id\_client, id\_angajat, data\_plasare\_comanda, data\_finalizare\_comanda)  
VALUES (5, 6, 7, TO\_DATE('16.10.2021', 'DD-MM-YYYY'), TO\_DATE('16.10.2021', 'DD-MM-YYYY'));

INSERT INTO COMENZI (id\_comanda, id\_client, id\_angajat, data\_plasare\_comanda, data\_finalizare\_comanda)  
VALUES (6, 5, 8, TO\_DATE('17.10.2021', 'DD-MM-YYYY'), TO\_DATE('18.10.2021', 'DD-MM-YYYY'));

INSERT INTO COMENZI (id\_comanda, id\_client, id\_angajat, data\_plasare\_comanda, data\_finalizare\_comanda)  
VALUES (7, 2, 4, TO\_DATE('18.10.2021', 'DD-MM-YYYY'), TO\_DATE('22.10.2021', 'DD-MM-YYYY'));

INSERT INTO COMENZI (id\_comanda, id\_client, id\_angajat, data\_plasare\_comanda, data\_finalizare\_comanda)  
VALUES (8, 3, 5, TO\_DATE('19.10.2021', 'DD-MM-YYYY'), TO\_DATE('21.10.2021', 'DD-MM-YYYY'));

INSERT INTO COMENZI (id\_comanda, id\_client, id\_angajat, data\_plasare\_comanda, data\_finalizare\_comanda)   
VALUES (9, 7, 9, TO\_DATE('19.10.2021', 'DD-MM-YYYY'), TO\_DATE('22.10.2021', 'DD-MM-YYYY'));

INSERT INTO COMENZI (id\_comanda, id\_client, id\_angajat, data\_plasare\_comanda, data\_finalizare\_comanda)  
VALUES (10, 8, 6, TO\_DATE('19.10.2021', 'DD-MM-YYYY'), TO\_DATE('22.10.2021', 'DD-MM-YYYY'));

INSERT INTO COMENZI (id\_comanda, id\_client, id\_angajat, data\_plasare\_comanda, data\_finalizare\_comanda)  
VALUES (11, 5, 10, TO\_DATE('20.10.2021', 'DD-MM-YYYY'), TO\_DATE('24.10.2021', 'DD-MM-YYYY'));

INSERT INTO COMENZI (id\_comanda, id\_client, id\_angajat, data\_plasare\_comanda, data\_finalizare\_comanda)  
VALUES (12, 4, 10, TO\_DATE('21.10.2021', 'DD-MM-YYYY'), TO\_DATE('24.10.2021', 'DD-MM-YYYY'));

INSERT INTO COMENZI (id\_comanda, id\_client, id\_angajat, data\_plasare\_comanda, data\_finalizare\_comanda)  
VALUES (13, 7, 4, TO\_DATE('22.10.2021', 'DD-MM-YYYY'), TO\_DATE('24.10.2021', 'DD-MM-YYYY'));

INSERT INTO COMENZI (id\_comanda, id\_client, id\_angajat, data\_plasare\_comanda, data\_finalizare\_comanda)   
VALUES (14, 9, 5, TO\_DATE('22.10.2021', 'DD-MM-YYYY'), TO\_DATE('25.10.2021', 'DD-MM-YYYY'));

INSERT INTO COMENZI (id\_comanda, id\_client, id\_angajat, data\_plasare\_comanda, data\_finalizare\_comanda)  
VALUES (15, 4, 7, TO\_DATE('22.10.2021', 'DD-MM-YYYY'), TO\_DATE('26.10.2021', 'DD-MM-YYYY'));

INSERT INTO COMENZI (id\_comanda, id\_client, id\_angajat, data\_plasare\_comanda, data\_finalizare\_comanda)  
VALUES (16, 8, 8, TO\_DATE('01.11.2021', 'DD-MM-YYYY'), null);

INSERT INTO COMENZI (id\_comanda, id\_client, id\_angajat, data\_plasare\_comanda, data\_finalizare\_comanda)  
VALUES (17, 1, 7, TO\_DATE('11.10.2021', 'DD-MM-YYYY'), null);

INSERT INTO COMENZI (id\_comanda, id\_client, id\_angajat, data\_plasare\_comanda, data\_finalizare\_comanda)  
VALUES (18, 2, 4, sysdate, null);Popularea tabelei „Produse\_Comenzi”

1. Pouplarea tabelei „Produse\_Comenzi”

INSERT INTO PRODUSE\_COMENZI(id\_comanda, id\_produs, cantitate\_produs)  
VALUES (1, 1, 5);

INSERT INTO PRODUSE\_COMENZI(id\_comanda, id\_produs, cantitate\_produs)  
VALUES (1, 2, 4);

INSERT INTO PRODUSE\_COMENZI(id\_comanda, id\_produs, cantitate\_produs)  
VALUES (1, 5, 1);

INSERT INTO PRODUSE\_COMENZI(id\_comanda, id\_produs, cantitate\_produs)  
VALUES (1, 9, 2);

INSERT INTO PRODUSE\_COMENZI(id\_comanda, id\_produs, cantitate\_produs)  
VALUES (2, 3, 10);

INSERT INTO PRODUSE\_COMENZI(id\_comanda, id\_produs, cantitate\_produs)  
VALUES (2, 9, 50);

INSERT INTO PRODUSE\_COMENZI(id\_comanda, id\_produs, cantitate\_produs)  
VALUES (3, 6, 40);

INSERT INTO PRODUSE\_COMENZI(id\_comanda, id\_produs, cantitate\_produs)  
VALUES (3, 7, 2);

INSERT INTO PRODUSE\_COMENZI(id\_comanda, id\_produs, cantitate\_produs)  
VALUES (3, 4, 1);

INSERT INTO PRODUSE\_COMENZI(id\_comanda, id\_produs, cantitate\_produs)  
VALUES (4, 8, 6);  
  
INSERT INTO PRODUSE\_COMENZI(id\_comanda, id\_produs, cantitate\_produs)

VALUES (4, 2, 2);

INSERT INTO PRODUSE\_COMENZI(id\_comanda, id\_produs, cantitate\_produs)

VALUES (4, 9, 10);

INSERT INTO PRODUSE\_COMENZI(id\_comanda, id\_produs, cantitate\_produs)

VALUES (4, 5, 22);

1. Exemple de interogări variate

--**Jonctiuni externe**

--1. Sa se afiseze produsele comandate impreuna cu cele care nu se regasesc pe nici o comanda.

SELECT p.id\_produs, p.nume\_produs, pc.id\_produs

FROM PRODUSE p, PRODUSE\_COMENZI pc

WHERE p.id\_produs = pc.id\_produs (+);

--2. Sa se afiseze toti angajatii care au incheiat comenzi, precum si cei care nu au incheiat comenzi.

SELECT a.id\_angajat, a.nume, a.prenume, c.id\_comanda

FROM ANGAJATI A, COMENZI c

WHERE a.id\_angajat = c.id\_angajat (+);

--**Realizarea interogarilor pe baza unor conditii**

--**Expresia CASE**

--1. Sa se identifice ierarhia angajatilor responsabili cu mancarea.

SELECT a.id\_angajat, a.nume, a.prenume, f.denumire\_functie, LEVEL

FROM ANGAJATI a, FUNCTIE f

WHERE a.id\_functie = f.id\_functie

CONNECT BY NOCYCLE a.id\_angajat = PRIOR a.id\_manager

START WITH f.denumire\_functie = 'Ajutor-Bucatar';

2. Sa se calculeze sporul salarial angajatiilor in functie de pozitia (functia) ocupata:

• 0.5% din valoarea salariului daca functia este Sef-Bucatar

• 0.9% din valoarea salariului daca functia este Programator-Sef

• 1.5% din valoarea salariului daca functia este Manager

Pentru celelalte functii comisionul va fi 0.

SELECT a.id\_angajat, a.nume, a.prenume, f.denumire\_functie,

CASE

WHEN UPPER(f.denumire\_functie) = 'SEF-BUCATAR' THEN 0.5

WHEN UPPER(f.denumire\_functie) = 'PROGRAMATOR-SEF' THEN 0.9

WHEN UPPER(f.denumire\_functie) = 'MANAGER' THEN 1.5

ELSE 0

END as spor\_salarial

FROM ANGAJATI a, FUNCTIE f

WHERE a.id\_functie = f.id\_functie;

--Aplicand sporul, salariul devine:

SELECT a.id\_angajat, a.nume, a.prenume, f.denumire\_functie, a.salariu AS salariu\_initial,

CASE

WHEN UPPER(f.denumire\_functie) = 'SEF-BUCATAR' THEN 0.5

WHEN UPPER(f.denumire\_functie) = 'PROGRAMATOR-SEF' THEN 0.9

WHEN UPPER(f.denumire\_functie) = 'MANAGER' THEN 1.5

ELSE 0

END as sporul\_salarial,

(CASE

WHEN UPPER(f.denumire\_functie) = 'SEF-BUCATAR' THEN 0.5

WHEN UPPER(f.denumire\_functie) = 'PROGRAMATOR-SEF' THEN 0.9

WHEN UPPER(f.denumire\_functie) = 'MANAGER' THEN 1.5

ELSE 0

END) \* a.salariu + a.salariu AS valoare\_salariu\_cu\_spor

FROM ANGAJATI a, FUNCTIE f

WHERE a.id\_functie = f.id\_functie;

--**Functia Decode**

SELECT a.id\_angajat, a.nume, a.prenume, f.denumire\_functie,

DECODE(upper(f.denumire\_functie), 'SEF-BUCATAR', 0.5, 'PROGRAMATOR-SEF', 0.9, 'MANAGER', 1.5, 0) AS sporul\_salarial

FROM ANGAJATI a, functie f

WHERE a.id\_functie = f.id\_functie;

2) Sa se calculeze taxa de transport pentru clienti astfel:

• daca clientul a comandat intr-o cantitate mai mica de 10 bucati, atunci taxa este de 10%;

• daca cantitatea e intre 10-20 de bucati, atunci taxa este de 5%;

• daca cantitatea e mai mare de 20 de bucati, atunci taxa este 0;

SELECT pc.id\_comanda, pc.id\_produs, pc.cantitate\_produs, p.nume\_produs, p.pret\_produs\_lei,

CASE

WHEN UPPER(pc.cantitate\_produs) < 10 THEN 0.1

WHEN UPPER(pc.cantitate\_produs) BETWEEN 10 AND 20 THEN 0.05

ELSE 0

END as taxa\_transport

FROM PRODUSE\_COMENZI pc, PRODUSE p

WHERE pc.id\_produs = p.id\_produs;

/\*

**Operatorii algebrei relationale:** UNION, INTERSECT, MINUS

**Operatorul MINUS**

1) Sa se afiseze angajatii care au salariul intre 2222 si 7777 fara cei care au salariul 3333 si 5555.

SELECT id\_angajat, nume, prenume, salariu

FROM angajati

WHERE salariu BETWEEN 2222 AND 7777

MINUS

SELECT id\_angajat, nume, prenume, salariu

FROM angajati

WHERE salariu IN (3333, 5555);

2) Sa se calculeze taxa de transport pentru clienti astfel:

• daca clientul a comandat intr-o cantitate mai mica de 10 bucati, atunci taxa este de 10%;

• daca cantitatea e intre 10-20 de bucati, atunci taxa este de 5%;

• daca cantitatea e mai mare de 20 de bucati, atunci taxa este 0;

Din acestea sa se elimine inregistrarile a caror produse incep cu litera G.

Ordonati descrescator in functie de numele produselor.

Observatie: clauza Order by se poate mentiona o singura data la sfarsitul intregii cereri.

SELECT pc.id\_comanda, pc.id\_produs, pc.cantitate\_produs, p.nume\_produs, p.pret\_produs\_lei,

CASE

WHEN UPPER(pc.cantitate\_produs) < 10 THEN 0.1

WHEN UPPER(pc.cantitate\_produs) BETWEEN 10 AND 20 THEN 0.05

ELSE 0

END as taxa\_transport

FROM PRODUSE\_COMENZI pc, PRODUSE p

WHERE pc.id\_produs = p.id\_produs

MINUS

SELECT pc.id\_comanda, pc.id\_produs, pc.cantitate\_produs, p.nume\_produs, p.pret\_produs\_lei,

CASE

WHEN UPPER(pc.cantitate\_produs) < 10 THEN 0.1

WHEN UPPER(pc.cantitate\_produs) BETWEEN 10 AND 20 THEN 0.05

ELSE 0

END as taxa\_transport

FROM PRODUSE\_COMENZI pc, PRODUSE p

WHERE pc.id\_produs = p.id\_produs

AND UPPER(p.nume\_produs) LIKE 'G%'

ORDER BY nume\_produs DESC;

**Operatorul UNION**

3) Sa se calculeze taxa de transport pentru clienti folosind operatorul UNION:

• daca clientul a comandat intr-o cantitate mai mica de 10 bucati, atunci taxa este de 10%;

• daca cantitatea e intre 10-20 de bucati, atunci taxa este de 5%;

• daca cantitatea e mai mare de 20 de bucati, atunci taxa este 0;

SELECT pc.id\_comanda, pc.id\_produs, pc.cantitate\_produs, p.nume\_produs, p.pret\_produs\_lei,

(p.pret\_produs\_lei \* 0.1) AS taxa\_transport

FROM PRODUSE\_COMENZI pc, PRODUSE p

WHERE pc.id\_produs = p.id\_produs

AND pc.cantitate\_produs < 10

UNION

SELECT pc.id\_comanda, pc.id\_produs, pc.cantitate\_produs, p.nume\_produs, p.pret\_produs\_lei,

(p.pret\_produs\_lei \* 0.05) AS taxa\_transport

FROM PRODUSE\_COMENZI pc, PRODUSE p

WHERE pc.id\_produs = p.id\_produs

AND pc.cantitate\_produs BETWEEN 10 AND 20

UNION

SELECT pc.id\_comanda, pc.id\_produs, pc.cantitate\_produs, p.nume\_produs, p.pret\_produs\_lei,

(p.pret\_produs\_lei \* 0) AS taxa\_transport

FROM PRODUSE\_COMENZI pc, PRODUSE p

WHERE pc.id\_produs = p.id\_produs

AND pc.cantitate\_produs >= 20;

**Operatorul INTERSECT**

4) Sa se afiseze angajatii care au salariul intre 5555 si 7777 si au certificat verde.

SELECT \*

FROM ANGAJATI

WHERE salariu BETWEEN 5555 AND 7777

INTERSECT

SELECT \*

FROM ANGAJATI

WHERE LOWER(certificat\_verde) LIKE 'da';

1. Gestiunea altor obiecte ale bazei de date

**TABELE VIRTUALE**

1. Sa realizeze o tabela virtuala cu toti angajatii din departamentul 3. Actualizăm salariul.

CREATE OR REPLACE VIEW v\_angajati\_3

AS SELECT \* FROM ANGAJATI

WHERE id\_departament = 3;

SELECT \* FROM v\_angajati\_3;

UPDATE v\_angajati\_3

SET SALARIU = SALARIU - 500;

--2. Stocarea unei interogări care să permită adăugarea unor condiţii ulterioare

SELECT \* FROM v\_angajati\_3 WHERE EXTRACT(MONTH FROM data\_nastere) = 6;

--3. Actualizarea tabelelor virtuale

CREATE OR REPLACE VIEW v\_produse

AS SELECT \* FROM PRODUSE;

UPDATE v\_produse

SET pret\_produs\_lei = pret\_produs\_lei + 7;

SELECT \* FROM v\_produse;

--4. Opţiunea WITH READ ONLY

CREATE OR REPLACE VIEW v\_functie

AS SELECT \* FROM functie

WITH READ ONLY;

SELECT \* FROM v\_functie;

--5. Sa se stearga tabela virtuală v\_functie

DROP VIEW v\_functie;

--6. Vizualizarea informatiilor despre tabelele virtuale:

SELECT VIEW\_NAME, TEXT FROM USER\_VIEWS;

**INDECSI**

--1. Sa se creeze un index pe tabela angajati pe coloana prenume:

SELECT \* FROM angajati WHERE nume = 'Coman';

CREATE INDEX idx\_nume ON angajati(nume);

SELECT \* FROM angajati WHERE nume = 'Coman';

SELECT \* FROM angajati WHERE UPPER(nume) = 'COMAN';

CREATE INDEX idx\_upper\_nume ON angajati(UPPER(nume));

--2. Vizualizarea indecşilor unui anumit utilizator:

SELECT \* FROM USER\_INDEXES;

--3. Sa se stearga indexul creat anterior:

DROP INDEX idx\_nume;

**SECVENTE**

1. Sa se creeze o secventa pentru asigurarea unicitatii cheii primare din tabela Comenzi.

CREATE SEQUENCE seq\_idcomanda

START WITH 19 INCREMENT BY 1

MAXVALUE 1000 NOCYCLE;

INSERT INTO comenzi VALUES (seq\_idcomanda.NEXTVAL, 5, 4, TO\_DATE('25.12.2021', 'dd.mm.yyyy'), NULL);

SELECT \* FROM COMENZI;

--2. Sa se afiseze valoarea curenta a secventei:

SELECT seq\_idcomanda.CURRVAL FROM DUAL;

--3. Să se modifice pasul de incrementare şi valoarea maximă pentru secvenţa anterioară:

ALTER SEQUENCE seq\_idcomanda INCREMENT BY 2;

ALTER SEQUENCE seq\_idcomanda MAXVALUE 2000;

INSERT INTO comenzi VALUES (seq\_idcomanda.NEXTVAL, 5, 4, TO\_DATE('27.12.2021', 'dd.mm.yyyy'), NULL);

SELECT \* FROM COMENZI;

--4. Să se vizualizeze informaţiile depre secvenţele utilizatorilor:

SELECT \* FROM USER\_SEQUENCES;

--5. Să se steargă secvenţa seq\_idcomanda :

DROP SEQUENCE seq\_idcomanda;

**SINONIME**

1. Sa se creeze un sinonim pentru tabela comenzi:

CREATE SYNONYM c FOR COMENZI;

--2. Vizualizarea sinonimelor se realizeaza astfel:

SELECT \* FROM USER\_SYNONYMS;

--3. Sa se stearga sinonimul creat anterior:

DROP SYNONYM c;

**GROUP BY**

Sa se calculeze valoarea fiecarui produs (valoare = cantitate \* pret)

--si sa se afiseze numele produsului, pretul, cantitatea si valoarea.

SELECT DISTINCT p.nume\_produs, p.pret\_produs\_lei, (SELECT COUNT(\*)   
FROM PRODUSE\_COMENZI pc  
WHERE pc.id\_produs = p.id\_produs  
GROUP BY pc.id\_produs) AS cantitate\_produs,  
(p.pret\_produs\_lei \* (  
(SELECT COUNT(\*)   
FROM PRODUSE\_COMENZI pc  
WHERE pc.id\_produs = p.id\_produs  
GROUP BY pc.id\_produs))) AS valoare  
FROM PRODUSE\_COMENZI pc, PRODUSE p   
WHERE pc.id\_produs = p.id\_produs;