

Academia de Studii Economice

Facultatea de Cibernetică, Statistică și Informatică Economică

**PROIECT BAZE DE DATE**

**Gestiunea datelor dintr-un magazin de haine**

**Mihai Irina Maria**

Grupa 1058

Seria D

* **Obiectivul proiectului**

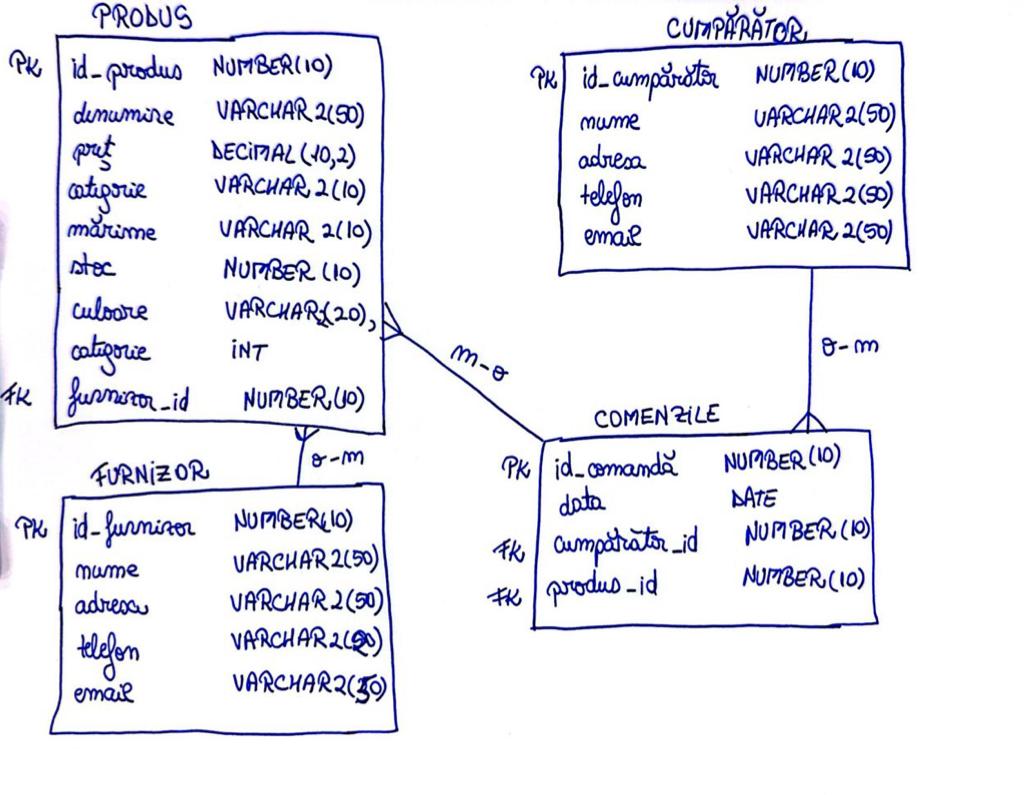
Aceaatsa bază de date pentru un magazin de haine are ca obiectiv să gestioneze informațiile despre produsele, clienții, furnizorii și comenzile magazinului. De asemenea, aceasta poate fi utilizata pentru a urmari stocul de produse, a gestiona comenzile si a monitoriza vanzarile.

Pentru a gestiona informatiile despre produse, baza de date include detalii despre fiecare produs, cum ar fi denumirea, pretul, categoria, dimensiunile si culorile disponibile. De asemenea, baza de date include informatii si despre furnizorii produselor, cum ar fi numele, adresa, umarul de telefon si email-ul.

Pentru a gestiona informatiile despre cumparatori, baza de date include detalii despre fiecare client, cum ar fi numele, adresa, numarul de telefon si adresa de e-mail.

Pentru a gestiona informatiile despre comenzile magazinului, baza de date include detalii despre fiecare comanda, cum ar fi numarul comenzii, data plasarii, produsele comandate si adresa de livrare.

* **Descrierea tabelelor**



* **Atrubute si restrictii**

**PRODUS:**

Id\_produs NUMBER(10) (PK)

Denumire VARCHAR2(50)

Pret NUMBER(10,2)

Categorie VARCHAR2(20)

Marime VARCHAR2(10)

Stoc NUMBER(10)

Furnizor\_id NUMBER(10) (FK)

**CLIENTI :**

Id\_client NUMBER(10) (PK)

Nume VARCHAR2(50)

Adresa VARCHAR2(50)

Telefon VARCHAR2(20)

Email VARCHAR2(50)

**FURNIZORI:**

Id\_furnizori NUMBER(10) (PK)

Nume VARCHAR2(50)

Adresa VARCHAR2(50)

Telefon VARCHAR2(50)

Email VARCHAR2(50)

**COMENZI:**

Id\_comenzi NUMBER(10)

Data DATE

Id\_client NUMBER(10) (FK)

Id\_produs NUMBER(10) (FK)

* **Tipurile de legaturi**

Tabela **PRODUS** are o legatura MANY – TO – ONE cu tabela **FURNIZORI**. Un furnizor poate furniza mai multe produse, dar un produs este furnizat de un singur furnizor .

Tabela **COMENZI** are o relatie MANY – TO – ONE cu tabela **CLIENTI**. Un client poate plasa mai multe comenzi, dar o comanda este plasata de un singur client.

Tabela **COMENZI** are o relatie MANY – TO – ONE cu tabela **PRODUS**. Un produs poate fi comandat de mai multi clienti, dar o comanda este plasata pentru un singur produs.

2.

* Crearea tabelelor (**CREATE**)

1. Tabela **PRODUS**

CREATE TABLE produs(

id\_produs NUMBER(10) PRIMARY KEY,

denumire VARCHAR2(50),

pret DECIMAL(10,2),

categorie VARCHAR2(20),

marime VARCHAR2(10),

stoc NUMBER(10),

culoare VARCHAR2(20),

cantitate INT,

furnizor\_id NUMBER(10),

FOREIGN KEY (furnizor\_id) REFERENCES furnizori(id\_furnizor)

);

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Tabela **CUMPARATOR**

CREATE TABLE cumparator (

id\_cumparator NUMBER(10) PRIMARY KEY,

nume VARCHAR2(50),

adresa VARCHAR2(50),

telefon VARCHAR2(50),

email VARCHAR2(50)

);

A screenshot of a computer

Description automatically generated

1. Tabela **FURNIZOR**

CREATE TABLE furnizori(

id\_furnizor NUMBER(10) PRIMARY KEY,

nume VARCHAR2(50),

adresa VARCHAR2(50),

telefon VARCHAR2(20),

email VARCHAR2(50)

);

A screenshot of a computer

Description automatically generated

1. Tabela **COMENZILE**

CREATE TABLE comenzile(

id\_comanda NUMBER(10) PRIMARY KEY,

data DATE,

cumparator\_id NUMBER(10),

FOREIGN KEY (cumparator\_id) REFERENCES cumparator(id\_cumparator),

produs\_id NUMBER(10),

FOREIGN KEY (produs\_id) REFERENCES produs(id\_produs)

);

A screenshot of a computer

Description automatically generated

* **DROP**
* Stergerea tabelei PRODUS

A screenshot of a computer

Description automatically generated

* **ALTER**
* Stergerea coloanei STOC

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

* Adaugarea coloanei STOC

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Exemple cu operatiile de actualizare a datelor

* **INSERT**

1. Populam tabela **FURNIZORI**

INSERT INTO furnizori (id\_furnizor, nume, adresa, telefon, email)

VALUES (1, 'Furnizor1', 'Strada Libertatii, nr. 10, Bucuresti', '0722000001', 'furnizor1@email.com');

INSERT INTO furnizori (id\_furnizor, nume, adresa, telefon, email)

VALUES (2, 'Furnizor2', 'Bulevardul Unirii, nr. 20, Bucuresti', '0722000002', 'furnizor2@email.com');

INSERT INTO furnizori (id\_furnizor, nume, adresa, telefon, email)

VALUES (3, 'Furnizor3', 'Calea Victoriei, nr. 30, Bucuresti', '0722000003', 'furnizor3@email.com');

INSERT INTO furnizori (id\_furnizor, nume, adresa, telefon, email)

VALUES (4, 'Furnizor4', 'Strada Mihai Viteazu, nr. 40, Bucuresti', '0722000004', 'furnizor4@email.com');

INSERT INTO furnizori (id\_furnizor, nume, adresa, telefon, email)

VALUES (5, 'Furnizor5', 'Bulevardul Carol I, nr. 50, Bucuresti', '0722000005', 'furnizor5@email.com');

INSERT INTO furnizori (id\_furnizor, nume, adresa, telefon, email)

VALUES (6, 'Furnizor6', 'Calea Dorobantilor, nr. 60, Bucuresti', '0722000006', 'furnizor6@email.com');

INSERT INTO furnizori (id\_furnizor, nume, adresa, telefon, email)

VALUES (7, 'Furnizor7', 'Strada Ion Mihalache, nr. 70, Bucuresti', '0722000007', 'furnizor7@email.com');

INSERT INTO furnizori (id\_furnizor, nume, adresa, telefon, email)

VALUES (8, 'Furnizor8', 'Bulevardul Magheru, nr. 80, Bucuresti', '0722000008', 'furnizor8@email.com');

INSERT INTO furnizori (id\_furnizor, nume, adresa, telefon, email)

VALUES (9, 'Furnizor9', 'Calea Mosilor, nr. 90, Bucuresti', '0722000009', 'furnizor9@email.com');

INSERT INTO furnizori (id\_furnizor, nume, adresa, telefon, email)

VALUES (10, 'Furnizor10', 'Strada Polona, nr. 100, Bucuresti', '0722000010', 'furnizor10@email.com');

SELECT \* FROM furnizori;

A screenshot of a computer

Description automatically generated

1. Populam tabela **CUMPARATOR**

INSERT INTO cumparator (id\_cumparator, nume, adresa, telefon, email)

VALUES (1, 'Ion Popescu', 'Strada Libertatii, nr. 10, Bucuresti', '0722000001', 'ion.popescu@email.com');

INSERT INTO cumparator (id\_cumparator, nume, adresa, telefon, email)

VALUES (2, 'Maria Ionescu', 'Bulevardul Unirii, nr. 20, Bucuresti', '0722000002', 'maria.ionescu@email.com');

INSERT INTO cumparator (id\_cumparator, nume, adresa, telefon, email)

VALUES (3, 'Vasile Dumitru', 'Calea Victoriei, nr. 30, Bucuresti', '0722000003', 'vasile.dumitru@email.com');

INSERT INTO cumparator (id\_cumparator, nume, adresa, telefon, email)

VALUES (4, 'Elena Popa', 'Strada Mihai Viteazu, nr. 40, Bucuresti', '0722000004', 'elena.popa@email.com');

INSERT INTO cumparator (id\_cumparator, nume, adresa, telefon, email)

VALUES (5, 'Ana Vasilescu', 'Bulevardul Carol I, nr. 50, Bucuresti', '0722000005', 'ana.vasilescu@email.com');

INSERT INTO cumparator (id\_cumparator, nume, adresa, telefon, email)

VALUES (6, 'Mihai Radu', 'Calea Dorobantilor, nr. 60, Bucuresti', '0722000006', 'mihai.radu@email.com');

INSERT INTO cumparator (id\_cumparator, nume, adresa, telefon, email)

VALUES (7, 'Ioana Pop', 'Strada Ion Mihalache, nr. 70, Bucuresti', '0722000007', 'ioana.pop@email.com');

INSERT INTO cumparator (id\_cumparator, nume, adresa, telefon, email)

VALUES (8, 'Radu Ionescu', 'Bulevardul Magheru, nr. 80, Bucuresti', '0722000008', 'radu.ionescu@email.com');

INSERT INTO cumparator (id\_cumparator, nume, adresa, telefon, email)

VALUES (9, 'Elena Dumitrescu', 'Calea Mosilor, nr. 90, Bucuresti', '0722000009', 'elena.dumitrescu@email.com');

INSERT INTO cumparator (id\_cumparator, nume, adresa, telefon, email)

VALUES (10, 'Ionut Popescu', 'Strada Polona, nr. 100, Bucuresti', '0722000010', 'ionut.popescu@email.com');

SELECT \* FROM cumparator;

A screenshot of a computer

Description automatically generated

1. Populam tabela **PRODUS**

INSERT INTO produs (id\_produs, denumire, pret, categorie, marime, stoc, culoare, furnizor\_id)

VALUES (1, 'Tricou', 49.99, 'Imbracaminte', 'M', 100, 'Alb', 1);

INSERT INTO produs (id\_produs, denumire, pret, categorie, marime, stoc, culoare, furnizor\_id)

VALUES (2, 'Pantaloni', 99.99, 'Imbracaminte', 'L', 200, 'Negru', 2);

INSERT INTO produs (id\_produs, denumire, pret, categorie, marime, stoc, culoare, furnizor\_id)

VALUES (3, 'Geaca', 149.99, 'Imbracaminte', 'XL', 150, 'Albastru', 3);

INSERT INTO produs (id\_produs, denumire, pret, categorie, marime, stoc, culoare, furnizor\_id)

VALUES (4, 'Pulover', 79.99, 'Imbracaminte', 'S', 120, 'Rosu', 4);

INSERT INTO produs (id\_produs, denumire, pret, categorie, marime, stoc, culoare,furnizor\_id)

VALUES (5, 'Blugi', 89.99, 'Imbracaminte', 'M', 80, 'Albastru', 1);

INSERT INTO produs (id\_produs, denumire, pret, categorie, marime, stoc, culoare, furnizor\_id)

VALUES (6, 'Camasa', 69.99, 'Imbracaminte', 'L', 90, 'Alb', 2);

INSERT INTO produs (id\_produs, denumire, pret, categorie, marime, stoc, culoare, furnizor\_id)

VALUES (7, 'Rochie', 119.99, 'Imbracaminte', 'S', 70, 'Rosu', 3);

INSERT INTO produs (id\_produs, denumire, pret, categorie, marime, stoc, culoare, furnizor\_id)

VALUES (8, 'Fusta', 79.99, 'Imbracaminte', 'M', 60, 'Negru', 4);

INSERT INTO produs (id\_produs, denumire, pret, categorie, marime, stoc, culoare, furnizor\_id)

VALUES (9, 'Costum', 299.99, 'Imbracaminte', 'XL', 50, 'Gri', 1);

INSERT INTO produs (id\_produs, denumire, pret, categorie, marime, stoc, culoare, furnizor\_id)

VALUES (10, 'Palton', 249.99, 'Imbracaminte', 'L', 40, 'Maro', 2);

SELECT \* FROM produs;

A screenshot of a computer

Description automatically generated

1. Populam tabela **COMENZILE**

INSERT INTO comenzile (id\_comanda, data, cumparator\_id, produs\_id)

VALUES (1, TO\_DATE('2024-01-01', 'YYYY-MM-DD'), 1, 1);

INSERT INTO comenzile (id\_comanda, data, cumparator\_id, produs\_id)

VALUES (2, TO\_DATE('2024-01-02', 'YYYY-MM-DD'), 2, 2);

INSERT INTO comenzile (id\_comanda, data, cumparator\_id, produs\_id)

VALUES (3, TO\_DATE('2024-01-03', 'YYYY-MM-DD'), 3, 3);

INSERT INTO comenzile (id\_comanda, data, cumparator\_id, produs\_id)

VALUES (4, TO\_DATE('2024-01-04', 'YYYY-MM-DD'), 4, 4);

INSERT INTO comenzile (id\_comanda, data, cumparator\_id, produs\_id)

VALUES (5, TO\_DATE('2024-01-05', 'YYYY-MM-DD'), 1, 5);

INSERT INTO comenzile (id\_comanda, data, cumparator\_id, produs\_id)

VALUES (6, TO\_DATE('2024-01-06', 'YYYY-MM-DD'), 2, 6);

INSERT INTO comenzile (id\_comanda, data, cumparator\_id, produs\_id)

VALUES (7, TO\_DATE('2024-01-07', 'YYYY-MM-DD'), 3, 7);

INSERT INTO comenzile (id\_comanda, data, cumparator\_id, produs\_id)

VALUES (8, TO\_DATE('2024-01-08', 'YYYY-MM-DD'), 4, 8);

INSERT INTO comenzile (id\_comanda, data, cumparator\_id, produs\_id)

VALUES (9, TO\_DATE('2024-01-09', 'YYYY-MM-DD'), 1, 9);

INSERT INTO comenzile (id\_comanda, data, cumparator\_id, produs\_id)

VALUES (10, TO\_DATE('2024-01-10', 'YYYY-MM-DD'), 2, 10);

SELECT \* FROM comenzile;

A screenshot of a computer

Description automatically generated

* **UPDATE**

1. Pentru tabela **FURNIZORI**

* Sa se actualizeze numele furnizorilor cu id\_furnizor egal cu 1 in FURNIZOR NOU din tabela furnizori.

UPDATE furnizori

SET nume = 'FURNIZOR NOU'

WHERE id\_furnizor = 1;

A screenshot of a computer

Description automatically generated

1. Pentru tabela **CUMPARATOR**

* Sa se actualizeze adresa cumparatorilor care au numarul de telefon care incepe cu : 07220000, in ‘STRADA NOUA, nr.1, Bucuresti’.

UPDATE cumparator

SET adresa = 'STRADA NOUA, nr. 1, Bucuresti'

WHERE telefon LIKE '07220000%';

A screenshot of a computer

Description automatically generated

1. Pentru tabela **PRODUS**

* Sa se creasca pretul produselor din categoria Imbracaminte cu 10%.

UPDATE produs

SET pret = pret \* 1.1

WHERE categorie = 'Imbracaminte';

A screenshot of a computer

Description automatically generated

1. Pentru tabela **COMENZILE**

* Actualizați data comenzilor pentru produsele care au un preț mai mare de 100 la data de 11 ianuarie 2024.

UPDATE comenzile

SET data = TO\_DATE('2024-01-11', 'YYYY-MM-DD')

WHERE produs\_id IN (SELECT id\_produs FROM produs WHERE pret > 100);

A screenshot of a computer

Description automatically generated

* **DELETE**

1. Pentru tabela **FURNIZORI**

- Sa se stearga toți furnizorii care nu mai au niciun produs în stoc și a căror preț este mai mare de 200

A screenshot of a computer

Description automatically generated

1. Pentru tabela **CUMPARATOR**

* Sa se stearga toți cumpărătorii care nu au plasat nicio comandă după data de 1 ianuarie 2024.

A screenshot of a computer

Description automatically generated

1. Pentru tabela **PRODUS**

* Sa se stearga toate produsele care au un stoc mai mic de 5 unități și a căror preț este mai mare de 150.

A screenshot of a computer

Description automatically generated

1. Pentru tabela **COMENZILE**

* Sa se stearga toate comenzile plasate înainte de data de 1 ianuarie 2024 pentru produsele a căror preț este mai mare de 150

A screenshot of a computer

Description automatically generated

1. Exemple de interogari variate (**SELECT**)

1.

* Utilizarea operatorilor de **comparatie**
* Sa se selecteze toate coloanele pentru produsele care au pretul > 100.

SELECT \*

FROM produs

WHERE pret > 100;

A screenshot of a computer

Description automatically generated

2.

* **Join**-uri
* Să se afișeze numele cumpărătorilor care au plasat comenzi și denumirea produselor pe care le-au comandat.

SELECT c.nume AS nume\_cumparator, p.denumire AS nume\_produs

FROM cumparator c

INNER JOIN comenzile co ON c.id\_cumparator = co.cumparator\_id

INNER JOIN produs p ON co.produs\_id = p.id\_produs;

A screenshot of a computer

Description automatically generated

3.

- Să se afișeze denumirea tuturor produselor și numele cumpărătorilor care le-au comandat.

SELECT c.nume AS nume\_cumparator, p.denumire AS nume\_produs

FROM cumparator c

RIGHT JOIN comenzile co ON c.id\_cumparator = co.cumparator\_id

RIGHT JOIN produs p ON co.produs\_id = p.id\_produs;

A screenshot of a computer

Description automatically generated

4.

- Să se afișeze numele tuturor cumpărătorilor și denumirea produselor pe care le-au comandat.

SELECT c.nume AS nume\_cumparator, p.denumire AS nume\_produs

FROM comenzile co

LEFT JOIN cumparator c ON c.id\_cumparator = co.cumparator\_id

LEFT JOIN produs p ON co.produs\_id = p.id\_produs;

A screenshot of a table

Description automatically generated

5.

- Selectați numele cumpărătorului și denumirea produsului pentru toți cumpărătorii, indiferent dacă au plasat o comandă sau nu, și pentru toate produsele, indiferent dacă au fost comandate sau nu.

SELECT c.nume, p.denumire

FROM cumparator c

LEFT JOIN comenzile co ON c.id\_cumparator = co.cumparator\_id

LEFT JOIN produs p ON co.produs\_id = p.id\_produs;

A screenshot of a table

Description automatically generated

6.

- Sa se afiseze cumparatorii care au cheltuit cel mai mult

SELECT cu.nume, SUM(p.pret) AS total\_cheltuit

FROM cumparator cu

JOIN comenzile co ON cu.id\_cumparator = co.cumparator\_id

JOIN produs p ON co.produs\_id = p.id\_produs

GROUP BY cu.nume

ORDER BY total\_cheltuit DESC;

A screenshot of a computer

Description automatically generated

* Sa se afiseze cumparatorul care a cheltuit cel mai mult.

SELECT cu.nume, SUM(p.pret) AS total\_cheltuit

FROM cumparator cu

JOIN comenzile co ON cu.id\_cumparator = co.cumparator\_id

JOIN produs p ON co.produs\_id = p.id\_produs

GROUP BY cu.nume

HAVING SUM(p.pret) = (

SELECT MAX(total\_cheltuit)

FROM (

SELECT SUM(p.pret) AS total\_cheltuit

FROM cumparator cu

JOIN comenzile co ON cu.id\_cumparator = co.cumparator\_id

JOIN produs p ON co.produs\_id = p.id\_produs

GROUP BY cu.nume

)

);

A screenshot of a computer

Description automatically generated

7.

- Sa se identifice cumparatorii care au plasat cele mai multe comenzi

SELECT cu.nume, COUNT(\*) AS numar\_comenzi

FROM comenzile co

INNER JOIN cumparator cu ON co.cumparator\_id = cu.id\_cumparator

GROUP BY cu.nume

ORDER BY numar\_comenzi DESC;

A screenshot of a computer

Description automatically generated

8.

- Sa se identifice categoria de produse care apare cel mai des in comenzile plasate.

A screenshot of a computer

Description automatically generated

9.

* Utilizarea **functiilor de grup** si conditii asupra acestora
* Selectați categoria și prețul mediu al produselor pentru fiecare categorie, unde prețul mediu este mai mare de 50.

SELECT categorie, AVG(pret) AS Pret\_Mediu

FROM produs

GROUP BY categorie

HAVING AVG(pret) > 50;

A screenshot of a computer

Description automatically generated

10.

* Selectați categoria, prețul mediu și numărul de produse pentru fiecare categorie de produse.

SELECT categorie, AVG(pret) AS Pret\_Mediu, COUNT(\*) AS Numar\_Produse

FROM produs

GROUP BY categorie;

A screenshot of a calculator

Description automatically generated

11.

* Utilizarea functiilor **numerice**, de tip caracter, pentru data si timp
* Selectați prețul rotunjit la întregul cel mai apropiat, denumirea produsului în litere mari și data comenzii în format ‘YYYY-MM-DD’ pentru fiecare produs comandat.

SELECT ROUND(pret, 0) Pret\_Rotunjit, UPPER(denumire) Denumire, TO\_CHAR(data, 'YYYY-MM-DD') Data

FROM produs

JOIN comenzile ON produs.id\_produs = comenzile.produs\_id;

A screenshot of a table

Description automatically generated

12.

* Construirea de expresii cu **DECODE** si **CASE**
* Selectați denumirea produsului, categoria nouă (unde ‘Imbracaminte’ este înlocuită cu ‘Haine’, iar toate celelalte categorii sunt înlocuite cu ‘Altele’) și clasificarea prețului (unde produsele cu prețul mai mic de 50 sunt clasificate ca ‘Ieftin’, cele cu prețul între 50 și 100 sunt ‘Moderat’, iar restul sunt ‘Scump’) pentru fiecare produs.

SELECT denumire,

DECODE(categorie, 'Imbracaminte', 'Haine', 'Altele') AS Categorie\_Noua,

CASE

WHEN pret < 50 THEN 'Ieftin'

WHEN pret BETWEEN 50 AND 100 THEN 'Moderat'

ELSE 'Scump'

END AS Clasificare\_Pret

FROM produs;

A screenshot of a computer

Description automatically generated

13.

* **UNION**

* Să se afișeze o listă cu toți furnizorii și cumpărătorii, fără duplicate, ordonată alfabetic.

SELECT nume FROM furnizori

UNION

SELECT nume FROM cumparator

ORDER BY nume;

A screenshot of a table

Description automatically generated

14.

* **MINUS**
* Să se afișeze numele furnizorilor care nu sunt și cumpărători.

A screenshot of a computer

Description automatically generated

15.

* **INTERSECT**
* Să se afișeze numele entităților care sunt și furnizori și cumpărători.

A screenshot of a computer

Description automatically generated

16.

* Subcereri (**cereri imbricate**)
* Selectați toate produsele care sunt furnizate de furnizorii ale căror nume încep cu ‘Furnizor’.

SELECT \*

FROM produs

WHERE furnizor\_id IN (

SELECT id\_furnizor

FROM furnizori

WHERE nume

LIKE 'Furnizor%');

A screenshot of a computer

Description automatically generated

17.

* Functii single-row
* Selectați numele cumpărătorilor în litere mari și lungimea numelui pentru fiecare cumpărător.

SELECT UPPER(nume) AS Nume\_Upper, LENGTH(nume) AS Lungime\_Nume

FROM cumparator;

A screenshot of a computer

Description automatically generated

18.

- Sa se afiseze produsele cu pretul peste medie

SELECT denumire, pret

FROM produs

WHERE pret > (SELECT AVG(pret) FROM produs);

A screenshot of a table

Description automatically generated

19.

- Sa se afiseze cumparatorii cu cele mai lungi nume.

SELECT nume, LENGTH(nume) AS lungime\_nume

FROM cumparator

ORDER BY lungime\_nume DESC;

A screenshot of a computer

Description automatically generated

20.

- Sa se afiseze produsele cu cel mai mare stoc.

SELECT denumire, stoc

FROM produs

ORDER BY stoc DESC;

A screenshot of a computer

Description automatically generated

21.

- Sa se afiseze cumpărătorii care au cheltuit mai mult decât media totală a cheltuielilor tuturor cumpărătorilor

SELECT cu.nume, SUM(p.pret) AS total\_cheltuit

FROM cumparator cu

JOIN comenzile co ON cu.id\_cumparator = co.cumparator\_id

JOIN produs p ON co.produs\_id = p.id\_produs

GROUP BY cu.nume

HAVING SUM(p.pret) > (

SELECT AVG(total\_cheltuit)

FROM (

SELECT SUM(p.pret) AS total\_cheltuit

FROM cumparator cu

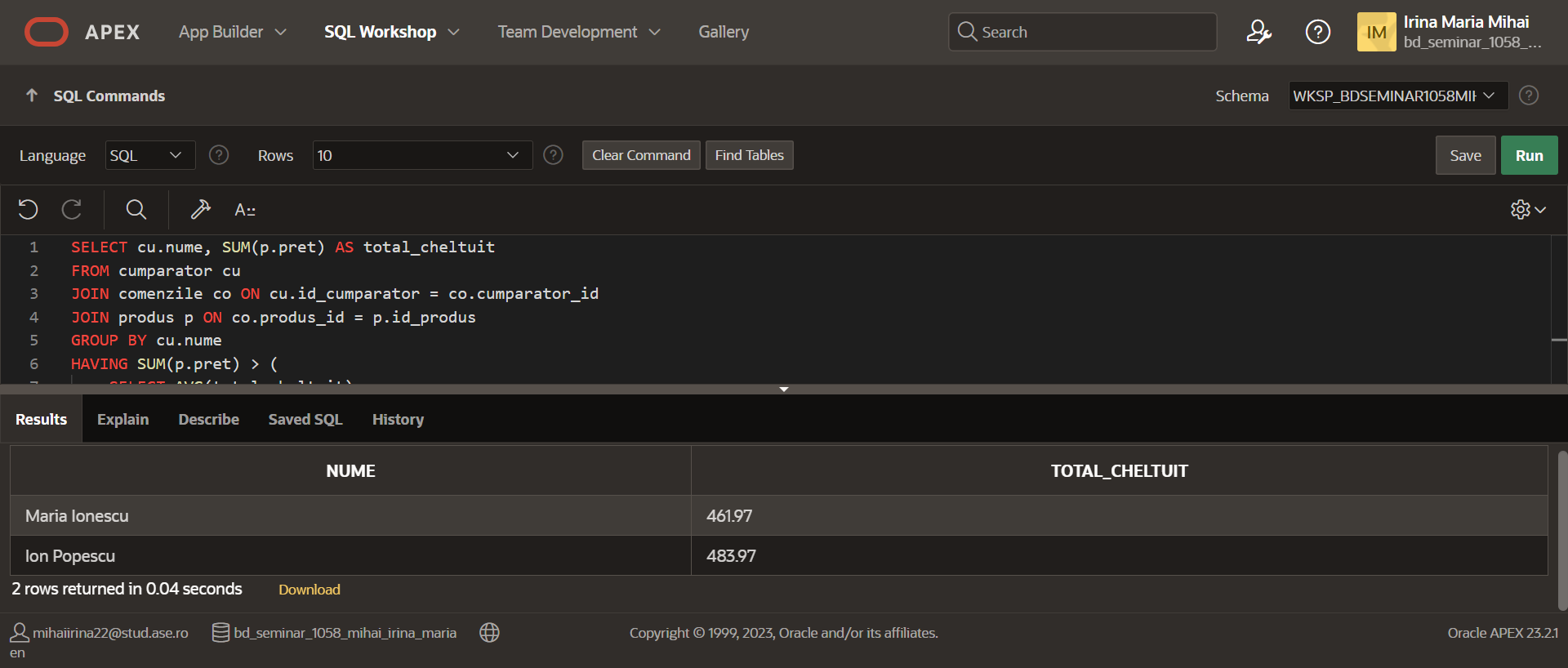
JOIN comenzile co ON cu.id\_cumparator = co.cumparator\_id

JOIN produs p ON co.produs\_id = p.id\_produs

GROUP BY cu.nume

)

);



1. Gestiunea altor obiecte ale bazei de date: vederi, indecsi, sinonime, secvente etc.

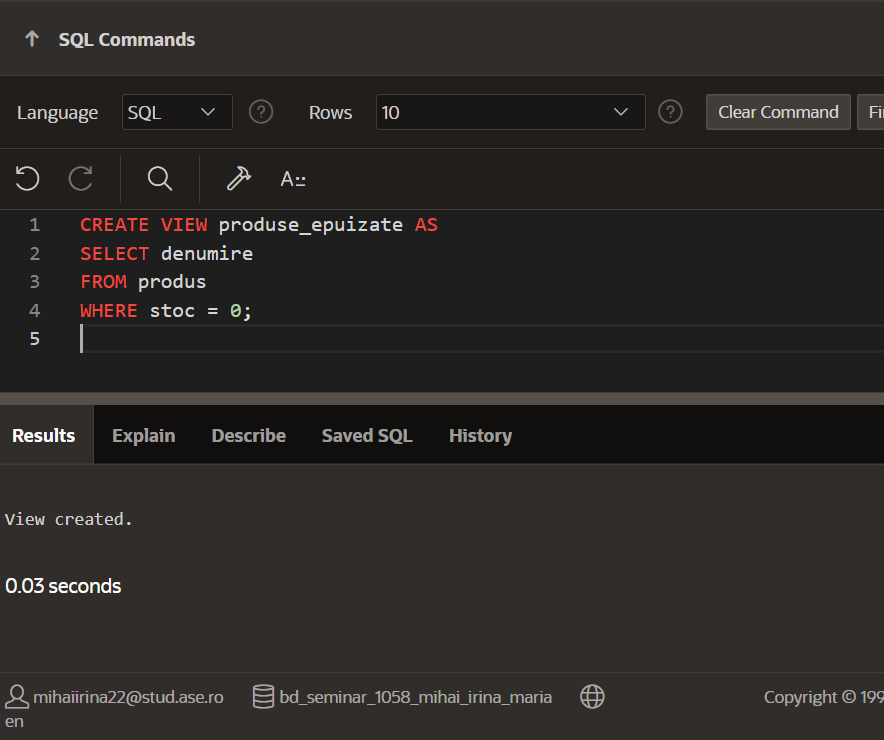
* **VEDERI**
* Sa se creeze o vedere care sa contina produsele cu stocul epuizat.

CREATE VIEW produse\_epuizate AS

SELECT denumire

FROM produs

WHERE stoc = 0;

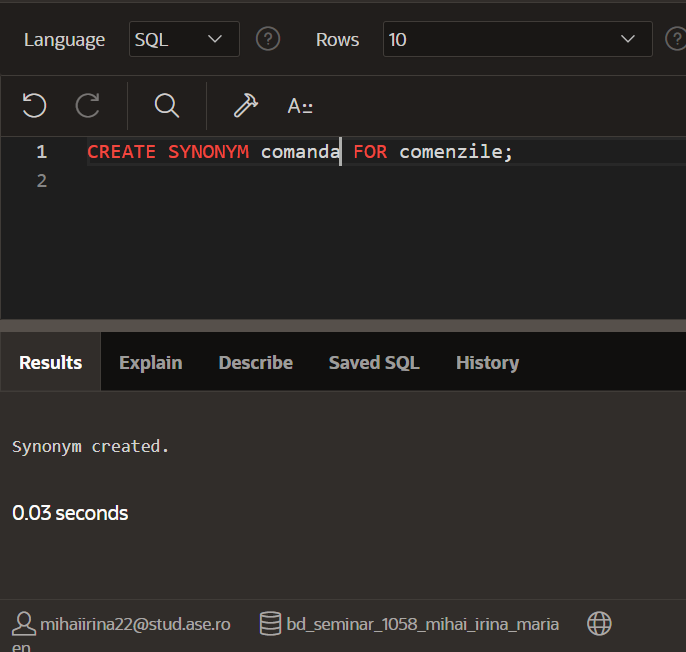


* **INDECSI**
* Creeaza un index pe coloana prêt din tabela PRODUS pentru a imbunatati performanta interogarilor care filtreaza dupa prêt.

A screenshot of a computer

Description automatically generated

* **SINONIME**
* Creeaza un sinonim pentru tabela COMENZILE .



* **SECVENTE**
* Creeaza o secventa pentru a genera automat ID-uri pentru comenzile noi.

A screenshot of a computer

Description automatically generated