

Ministerul Educației al Republicii Moldova
Universitatea Tehnică a Moldovei
Departamentul Inginerie, Software și Automatică

Proiect de curs

Disciplina: Baze de date și cunoștințe

Efectuat de:

st.gr.TI-152 Lașcu Mihai

Controlat de:

lector asistent Cojanu Irina

Chișinău 2018

CUPRINS

1. Proiectarea logică a bazei de date.....	4
Analiza domeniului	4
Descrierea Atributelor.....	4
Stabilirea dependentelor functionale între attribute.....	4
Stabilirea dependentelor functionale între attribute	5
Sinteza bazei de date	5
Decodificarea atributelor.....	13
Descrierea asocierilor între relații	14
2. Implimentarea bazei de date în SQL	15
Crearea bazei de date noi în T-SQL.....	15
Crearea relațiilor și constrângerilor de integritate.....	15
Crearea cheilor externe.....	17
Crearea Diagramei Bazei de Date.....	18
Popularea cu tupluri a relațiilor BD.....	18
Crearea schemelor	20
Crearea Utilizatorilor.....	20
Crearea viziunilor	22
Crearea sinonimelor.....	23
3. Interogarea Bazei de Date	24
Crearea interogărilor ce exprimă operațiile algebrei relaționale	24
Joncțiune externă.....	27
Semijoncțiunea	28
Crearea interogărilor cu funcții de agregare.....	29
Crearea interogărilor cu grupare.....	30
Subinterogări	31
Subinterogări în alte locuri decât clauza WHERE	32
Crearea interogărilor de actualizare a relațiilor.....	33
Crearea procedurilor stocate	34

Concluzie.....	36
ANEXE.....	37

1. Proiectarea logică a bazei de date

Analiza domeniului

Am creat baza de date pentru un magazin de tehnică online. Aceasta conține informații despre produse și comenzile efectuate. La procurarea unui produs clientul trebuie să introducă datele sale personale conform buletinului de identitate. Fiecare comandă, produs și client va fi nominalizată după ID. Pentru a afla mai multe informații despre datele personale ale clientului vom accesa căutarea acestora după IDNP.

Descrierea Atributelor

Cod Atribut	Nume Atribut
A	ID_produc
B	Nume_produc
C	Descriere
D	Pret
E	ID_comanda
F	ClientID
G	Data_Efecturarii
I	IDNP
J	Nume
K	Prenume
L	Sex
M	Tara
N	Virsta

Stabilirea dependentelor functionale între attribute

Multimea dependentei functionale F:

$F = \{ \text{ID_produc} \rightarrow \text{Nume_produc}, \text{Descriere}, \text{Pret} \}$

$\text{ID_comanda} \rightarrow \text{ClientID}, \text{Data_efectuarii}, \text{Cost}$

$\text{ID_Produc} \rightarrow \text{ID_Comanda}$

$\text{ClientID} \rightarrow \text{IDNP}, \text{Nume}, \text{Prenume}$

$\text{IDNP} \rightarrow \text{Sex}, \text{Tara}, \text{Virsta} \}$

Stabilirea dependentelor functionale intre attribute

$F = \{ A \rightarrow BCD, E \rightarrow FGD, A \rightarrow E, F \rightarrow IJK, I \rightarrow LMN \}$

Sinteza bazei de date

a) Multimea bazei de date dupa codificare

$F = \{ A \rightarrow BCD, E \rightarrow FGD, A \rightarrow E, F \rightarrow IJK, I \rightarrow LMN \}$

b) Aducerea dependentelor la forma $X \rightarrow A$

$F = \{ A \rightarrow B, A \rightarrow C, A \rightarrow D, A \rightarrow E, E \rightarrow F, E \rightarrow G, E \rightarrow D, F \rightarrow I, F \rightarrow J, F \rightarrow K, I \rightarrow L, I \rightarrow M, I \rightarrow N \}$

c) Se adauga in multimea de dependente functionale dependenta de forma $R \rightarrow Q$ unde R este universal bazei de date, iar Q nu apartine lui R .

$F = \{ A \rightarrow B, A \rightarrow C, A \rightarrow D, A \rightarrow E, E \rightarrow F, E \rightarrow G, E \rightarrow D, F \rightarrow I, F \rightarrow J, F \rightarrow K, I \rightarrow L, I \rightarrow M, I \rightarrow N, ABCDEFGHIJKLMNOP \rightarrow Q \}$

d) Se construiesc acoperirea neredundanta pentru F :

$A \rightarrow B$:

$(A)^+ \setminus [F \setminus A \rightarrow B] = \langle ACDE, ACDEFG, ACDEFGHIJK, ACDEFGHIJKLMN \rangle$

B nu este submultime a lui $(A)^+$ de aceea $A \rightarrow B$ nu este redundanta in F

$A \rightarrow C$:

$(A)^+ \setminus [F \setminus A \rightarrow C] = \langle ABDE, ABDEFG, ABDEFGHIJK, ABDEFGHIJKLMN \rangle$

C nu este submultime a lui $(A)^+$ de aceea $A \rightarrow C$ nu este redundanta in F

$A \rightarrow D$:

$(A)^+ \setminus [F \setminus A \rightarrow D] = \langle ABCE, ABCEDFG, ABCEDFGHIJK, ABCEDFGHIJKLMN \rangle$

D este submultime a lui $(A)^+$ de aceea $A \rightarrow D$ este redundanta in F

$A \rightarrow E$:

$(A)^+ \setminus [F \setminus A \rightarrow E] = \langle ABCD \rangle$ E nu este submultime a lui $(A)^+$ de aceea $A \rightarrow E$ nu este redundanta in F

$E \rightarrow F$:

$(E) + [F \setminus E \rightarrow F] = \langle EGD \rangle$ F nu este submultime a lui $(E)^+$ de aceea $E \rightarrow F$ nu este redundanta in F

$E \rightarrow G$:

$(E) + [F \setminus E \rightarrow G] = \langle EFD, EFDIJK, EFDIJKLMNOP \rangle$ G nu este submultime a lui $(E)^+$ de aceea $E \rightarrow G$ nu este redundanta in F

$E \rightarrow D$:

$(E) + [F \setminus E \rightarrow D] = \langle EFG, EFGIJK, EFGIJKLMNOP \rangle$ D nu este submultime a lui $(E)^+$ de aceea $E \rightarrow D$ nu este redundanta in F

$F \rightarrow I$:

$(F) + [F \setminus F \rightarrow I] = \langle FJK \rangle$ I nu este submultime a lui $(F)^+$ de aceea $F \rightarrow I$ nu este redundanta in F

$F \rightarrow J$:

$(F) + [F \setminus F \rightarrow J] = \langle FIK, FIKLMN \rangle$ J nu este submultime a lui $(F)^+$ de aceea $F \rightarrow J$ nu este redundanta in F

$F \rightarrow K$:

$(F) + [F \setminus F \rightarrow K] = \langle FIJ, FIJLMN \rangle$ K nu este submultime a lui $(F)^+$ de aceea $F \rightarrow K$ nu este redundanta in F

$I \rightarrow L$:

$(I) + [F \setminus I \rightarrow L] = \langle IMN \rangle$ L nu este submultime a lui $(I)^+$ de aceea $I \rightarrow L$ nu este redundanta in F

$I \rightarrow M$:

$(I) + [F \setminus I \rightarrow M] = \langle ILN \rangle$ M nu este submultime a lui $(I)^+$ de aceea $I \rightarrow M$ nu este redundanta in F

$I \rightarrow N$:

$(I) + [F \setminus I \rightarrow N] = \langle ILM \rangle$ N nu este submultime a lui $(I)^+$ de aceea $I \rightarrow N$ nu este redundanta in F

$ABCDEFGHIJKLMN \rightarrow Q$:

$(ABCDEFGHIJKLMN) + [F \setminus ABCDEFGHIJKLMN \rightarrow Q] = \langle ABCDEFGHIJKLMN \rangle$ Q neredundanta

$FN = \{ A \rightarrow B, A \rightarrow C, A \rightarrow E, E \rightarrow F, E \rightarrow G, E \rightarrow D, F \rightarrow I, F \rightarrow J, F \rightarrow K, I \rightarrow L, I \rightarrow M, I \rightarrow N, ABCDEFGHIJKLMNOP \rightarrow Q \}$

e) Se trece mulțimea de dependențe funcționale prin algoritmul reducerii la stânga

$BCDEFGHIJKLMNOP \rightarrow Q$:

$(BCDEFGHIJKLMNOP) + [F] = \langle BCDEFGHIJKLMNOP \rangle$ Q nu este o submultime a lui $(BCDEFGHIJKLMNOP)$ de aceea A nu este redundant în F

$ACDEFGHIJKLMNOP \rightarrow Q$:

$(ACDEFGHIJKLMNOP) + [F] = \langle ACDEFGHIJKLMNOP, ABCDEFGHIJKLMNOPQ \rangle$ Q este submultime a lui $(ACDEFGHIJKLMNOP)$ de aceea B este redundantă în F

$ADEFGHIJKLMNOP \rightarrow Q$:

$(ADEFGHIJKLMNOP) + [F] = \langle ADEFGHIJKLMNOP, ABCDEFGHIJKLMNOPQ \rangle$ Q este submultime a lui $(ADEFGHIJKLMNOP)$ de aceea C este redundantă în F

$ADEFGHIJKLMNOP \rightarrow Q$:

$(AEFGHIJKLMNOP) + [F] = \langle AEFGHIJKLMNOP, ABCDEFGHIJKLMNOPQ \rangle$ Q este submultime a lui $(AEFGHIJKLMNOP)$ de aceea D este redundantă în F

$AEFGHIJKLMNOP \rightarrow Q$:

$(AFGHIJKLMNOP) + [F] = \langle AFGHIJKLMNOP, ABCDEFGHIJKLMNOPQ \rangle$ Q este submultime a lui $(AFGHIJKLMNOP)$ de aceea E este redundantă în F

$AFGHIJKLMNOP \rightarrow Q$:

$(AGHIJKLMNOP) + [F] = \langle AGHIJKLMNOP, ABCDEFGHIJKLMNOPQ \rangle$ Q este submultime a lui $(AGHIJKLMNOP)$ de aceea F este redundantă în F

$AGHIJKLMNOP \rightarrow Q$:

$(AHIJKLMNOP) + [F] = \langle AHIJKLMNOP, ABCDEFGHIJKLMNOPQ \rangle$ Q este submultime a lui $(AHIJKLMNOP)$ de aceea G este redundantă în F

AIJKLMN- \rightarrow Q:

$(AJKLMN)+[F] = \langle AJKLMN, ABCDEFGHIJKLMNQ \rangle$ Q este submultime a lui (AJKLMN) de aceea I este redundanta in F

AJKLMN- \rightarrow Q:

$(AKLMN)+[F] = \langle AKLMN, ABCDEFGHIJKLMNQ \rangle$ Q este submultime a lui (AKLMN) de aceea J este redundanta in F

AKLMN- \rightarrow Q:

$(ALMN)+[F] = \langle ALMN, ABCDEFGHIJKLMNQ \rangle$ Q este submultime a lui (ALMN) de aceea K este redundanta in F

ALMN- \rightarrow Q:

$(AMN)+[F] = \langle AMN, ABCDEFGHIJKLMNQ \rangle$ Q este submultime a lui (AMN) de aceea L este redundanta in F

AMN- \rightarrow Q:

$(AN)+[F] = \langle AN, ABCDEFGHIJKLMNQ \rangle$ Q este submultime a lui (AN) de aceea M este redundanta in F

AN- \rightarrow Q:

$(A)+[F] = \langle A, ABCDEFGHIJKLMNQ \rangle$ Q este submultime a lui (A) de aceea N este redundanta in F

$Flr = \{ A \rightarrow B, A \rightarrow C, A \rightarrow E, E \rightarrow F, E \rightarrow G, E \rightarrow D, F \rightarrow I, F \rightarrow J, F \rightarrow K, I \rightarrow L, I \rightarrow M, I \rightarrow N, A \rightarrow Q \}$

f) Formam clasele de echivalenta pentru multimea Flr

A- \rightarrow B:

$(A)+ = \langle A, ABCEQ, ABCEQFGD, ABCEQFGDIJK, ABCEQFGDIJKLMN \rangle$

A- \rightarrow C:

$(A)+ = \langle A, ABCEQ, ABCEQFGD, ABCEQFGDIJK, ABCEQFGDIJKLMN \rangle$

A->E:

(A)+ = <A, ABCEQ, ABCEQFGD, ABCEQFGDIJK, ABCEQFGDIJKLMN>

A->Q:

(A)+ = <A, ABCEQ, ABCEQFGD, ABCEQFGDIJK, ABCEQFGDIJKLMN>

E->G:

(E)+ = <E, EGFD, EGFDIJK, EGFDIJKLMN>

E->F:

(E)+ = <E, EGFD, EGFDIJK, EGFDIJKLMN>

E->D:

(E)+ = <E, EGFD, EGFDIJK, EGFDIJKLMN>

F->I:

(F)+ = <F, FIJK, FIJKLMN>

F->J:

(F)+ = <F, FIJK, FIJKLMN>

F->K:

(F)+ = <F, FIJK, FIJKLMN>

I->L:

(I)+ = <I, ILMN>

I->M:

(I)+ = <I, ILMN>

I->N:

(I)+ = <I, ILMN>

Avem urmatoarele clase de echivalenta

- 1) $E(A) = \{ A \rightarrow B, A \rightarrow C, A \rightarrow E \}$
 - 2) $E(E) = \{ E \rightarrow F, E \rightarrow G, E \rightarrow D \}$
 - 3) $E(F) = \{ F \rightarrow I, F \rightarrow J, F \rightarrow K \}$
 - 4) $E(I) = \{ I \rightarrow L, I \rightarrow M, I \rightarrow N \}$
 - 5) $E(A) = \{ A \rightarrow Q \}$
- g) Construim mulțimea J din F, cu determinanții X și Y, unde $X \leftrightarrow Y$ schimbă mulțimea $J = J \cup \{ X \rightarrow Y, Y \rightarrow X \}$

În rezultat avem:

$J = \{$
1) $\{ \}$
2) $\{ \}$
3) $\{ \}$
4) $\{ \}$
5) $\{ \}$
 $\}$

Pentru oricare A din Y, dacă $X \rightarrow A$ se află în F, atunci $F = F \setminus \{ X \rightarrow A \}$

Clase de echivalență primite

- 1) $E(A) = \{ A \rightarrow B, A \rightarrow C, A \rightarrow E \}$
- 2) $E(E) = \{ E \rightarrow F, E \rightarrow G, E \rightarrow D \}$
- 3) $E(F) = \{ F \rightarrow I, F \rightarrow J, F \rightarrow K \}$
- 4) $E(I) = \{ I \rightarrow L, I \rightarrow M, I \rightarrow N \}$
- 5) $E(A) = \{ A \rightarrow Q \}$

h) Stergem din F dependent tranzitiva:

$A \rightarrow B$:

$(A) + [F \setminus A \rightarrow B] = \langle ACDE, ACDEFG, ACDEFGIJK, ACDEFGIJKLMN \rangle$

B nu este submultime a lui (A)+ de aceea $A \rightarrow B$ nu este redundanta in F

A->C:

$(A)+[F \setminus A \rightarrow C] = \langle ABDE, ABDEFG, ABDEFGIJK, ABDEFGIJKLMN \rangle$

C nu este submultime a lui $(A)^+$ de aceea A->C nu este redundanta in F

A->E:

$(A)+[F \setminus A \rightarrow E] = \langle ABCD \rangle$ E nu este submultime a lui $(A)^+$ de aceea A->E nu este redundanta in F

E->F:

$(E)+[F \setminus E \rightarrow F] = \langle EGD \rangle$ F nu este submultime a lui $(E)^+$ de aceea E->F nu este redundanta in F

E->G:

$(E)+[F \setminus E \rightarrow G] = \langle EFD, EFDIJK, EFDIJKLMN \rangle$ G nu este submultime a lui $(E)^+$ de aceea E->G nu este redundanta in F

E->D:

$(E)+[F \setminus E \rightarrow D] = \langle EFG, EFGIJK, EFGIJKLMN \rangle$ D nu este submultime a lui $(E)^+$ de aceea E->D nu este redundanta in F

F->I:

$(F)+[F \setminus F \rightarrow I] = \langle FJK \rangle$ I nu este submultime a lui $(F)^+$ de aceea F->I nu este redundanta in F

F->J:

$(F)+[F \setminus F \rightarrow J] = \langle FIK, FIKLMN \rangle$ J nu este submultime a lui $(F)^+$ de aceea F->J nu este redundanta in F

F->K:

$(F)+[F \setminus F \rightarrow K] = \langle FIJ, FIJLMN \rangle$ K nu este submultime a lui $(F)^+$ de aceea F->K nu este redundanta in F

I->L:

$(I)+[F \setminus I \rightarrow L] = \langle IMN \rangle$ L nu este submultime a lui $(I)^+$ de aceea I->L nu este redundanta in F

I->M:

$(I)+[F \setminus I \rightarrow M] = \langle ILN \rangle$ M nu este submultime a lui (I)+ de aceea I->M nu este redundanta in F

I->N:

$(I)+[F \setminus I \rightarrow N] = \langle ILM \rangle$ N nu este submultime a lui (I)+ de aceea I->N nu este redundanta in F

$(A)+[F \setminus A \rightarrow Q] = \langle A, ABCDEFGHIJKLMNOP \rangle$ Q nu este submultime a lui (A)+ de aceea A->Q nu este redundanta in F

Avem multimea F1:

$$F1 = \{ E(A) = \{ A \rightarrow B, A \rightarrow C, A \rightarrow E \} \}$$

$$E(E) = \{ E \rightarrow F, E \rightarrow G, E \rightarrow D \}$$

$$E(F) = \{ F \rightarrow I, F \rightarrow J, F \rightarrow K \}$$

$$E(I) = \{ I \rightarrow L, I \rightarrow M, I \rightarrow N \}$$

$$E(A) = \{ A \rightarrow Q \}$$

Nu avem dependente în mulțimea J, deci vom crea mulțimea G care va fi egală cu F1

$$G = \{ E(A) = \{ A \rightarrow B, A \rightarrow C, A \rightarrow E \} \}$$

$$E(E) = \{ E \rightarrow F, E \rightarrow G, E \rightarrow D \}$$

$$E(F) = \{ F \rightarrow I, F \rightarrow J, F \rightarrow K \}$$

$$E(I) = \{ I \rightarrow L, I \rightarrow M, I \rightarrow N \}$$

$$E(A) = \{ A \rightarrow Q \}$$

}

i) Construim schemele bazei de date, din R5 eliminăm Q

$$1) R1 = ABCE \quad k1 = \{ A \}$$

$$2) R2 = EDFG \quad k2 = \{ E \}$$

$$3) R3 = FIJK \quad k3 = \{ F \}$$

$$4) R4 = ILMN \quad k4 = \{ I \}$$

$$5) R5 = A \quad k7 = \{ A \}$$

Baza de date sintetizată în 3FN:

Db = {(ABCE, {A}), (EDFG, {E}), (FIJK, {F}), (ILMN, {I}), (A, {A})}

Decodificarea atributelor

Nume codificate	Nume reale de atribute si relatii
<i>R1=ABCE,</i> <i>K={A}</i>	<i>Produsei,</i> <i>K={ID_produc}</i>
A	ID_produc
B	Nume_produc
C	Descriere
E	ID_comanda
<i>R2=EDFG,</i> <i>K={E}</i>	<i>Comenzi,</i> <i>K={ID_comanda}</i>
E	ID_comanda
D	Pret
F	ClientID
G	Data_Efectuarii
<i>R3=FIJK,</i> <i>K={F}</i>	<i>Clienti,</i> <i>K={ClientID}</i>
F	ClientID
I	IDNP
J	Nume
K	Prenume
<i>R4=ILMN,</i> <i>K={I}</i>	<i>Buletin,</i> <i>K={IDNP}</i>
I	IDNP
L	Sex
M	Tara
N	Virsta

Descrierea asocierilor între relații

Asocierile între relații:

- a) Clienti - Comenzi 1:n
- b) Comenzi –Produse 1:n
- c) Clienti - Buletin 1:1

Ordinea în care pot fi înserate tuplurile în relațiile BD va fi următoarea:

- a) Produse
- b) Comenzi
- c) Clienti
- d) Buletin

2. Implimentarea bazei de date în SQL

Crearea bazei de date noi în T-SQL

```
1. USE master
2. GO
3. IF DB_ID('MagTehGSM') IS NOT NULL
4. DROP DATABASE MagTehGSM
5. GO
6. CREATE DATABASE MagTehGSM
7. ON(
8. NAME = MagTehGSM,
9. FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\MagTehGSM
   .mdf',
10. SIZE = 5MB,
11. MAXSIZE = UNLIMITED,
12. FILEGROWTH = 1MB)
13. LOG ON(
14. NAME = MagTehGSM_log,
15. FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\MagTehGSM
   _log.ldf',
16. SIZE = 1MB,
17. MAXSIZE = UNLIMITED,
18. FILEGROWTH = 5MB)
19. GO
```

Вид

диск (C:) > Program Files > Microsoft SQL Server > MSSQL12.MSSQLSERVER > MSSQL > DATA



Имени	Дата изменения	Тип	Размера
 MagTehGSM	19.05.2018 19:47	SQL Server Databa...	4 288 КБ
 MagTehGSM_log	19.05.2018 19:47	SQL Server Databa...	1 072 КБ

Figura 1. Crearea bazei de date MagTehGSM

Crearea relațiilor și constrângerilor de integritate

```
1. use MagTehGSM
2. --relatia produse
3. IF OBJECT_ID('Produse', 'U') IS NOT NULL
4. DROP TABLE Produse
5. CREATE TABLE Profesori(
6.     ID_produs int NOT NULL,
7.     Nume_produs nvarchar(50) NOT NULL,
```

```

8.     Descriere varchar(50) NOT NULL,
9.     ID_comanda int NOT NULL,
10.    PRIMARY KEY (ID_produs)
11.    )
12. --relatie comenzi
13. IF OBJECT_ID('Comenzi', 'U') IS NOT NULL
14. DROP TABLE Comenzi
15. CREATE TABLE Comenzi(
16.     ID_comanda int NOT NULL,
17.     Pret int NOT NULL,
18.     ClientID int NOT NULL,
19.     Data_efectuarii date NULL,
20.     PRIMARY KEY (ID_comanda)
21.    )
22. --relatie clienti
23. IF OBJECT_ID('Clienti', 'U') IS NOT NULL
24. DROP TABLE Clienti
25. CREATE TABLE Clienti(
26.     ClientID int NOT NULL,
27.     IDNP varchar(13) NULL,
28.     NUME nvarchar(50) NULL,
29.     Prenume nvarchar(50) NULL,
30.     PRIMARY KEY (ClientID)
31.    )
32. --relatie buletin
33. IF OBJECT_ID('Buletin', 'U') IS NOT NULL
34. DROP TABLE Buletin
35. CREATE TABLE Buletin(
36.     IDNP varchar(13) NOT NULL,
37.     Sex char(1) NULL,
38.     Tara nvarchar(20) NULL,
39.     Virsta int NULL,
40.     PRIMARY KEY (IDNP)
41.    )

```

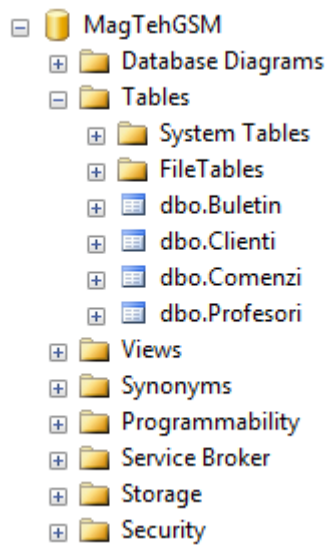


Figura 2. Crearea tabelelor

Crearea cheilor externe

```
1. --Chei externe
2. USE MagTehGSM
3. --Produce
4. ALTER TABLE Produce ADD CONSTRAINT [FK_ComenziProduce]
5. FOREIGN KEY(ID_comanda) REFERENCES Comenzi (ID_comanda)
6. ON UPDATE CASCADE;
7.
8. --Comenzi
9. ALTER TABLE Comenzi ADD CONSTRAINT [FK_ClientiComenzi]
10. FOREIGN KEY(ClientID) REFERENCES Clienti(ClientID)
11. ON UPDATE CASCADE;
12. --CHEIE UNICA
13. ALTER TABLE Clienti
14. ADD CONSTRAINT UC_Clienti UNIQUE(IDNP)
```

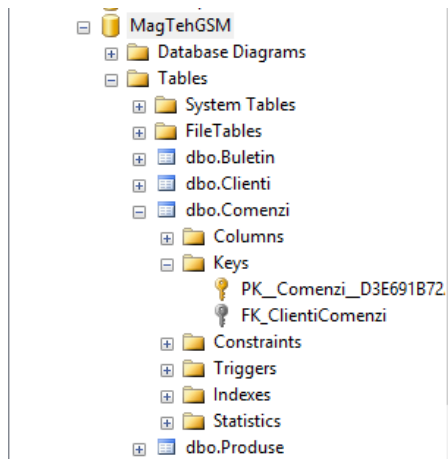


Figura 3. Chei externe

Crearea Diagramei Bazei de Date

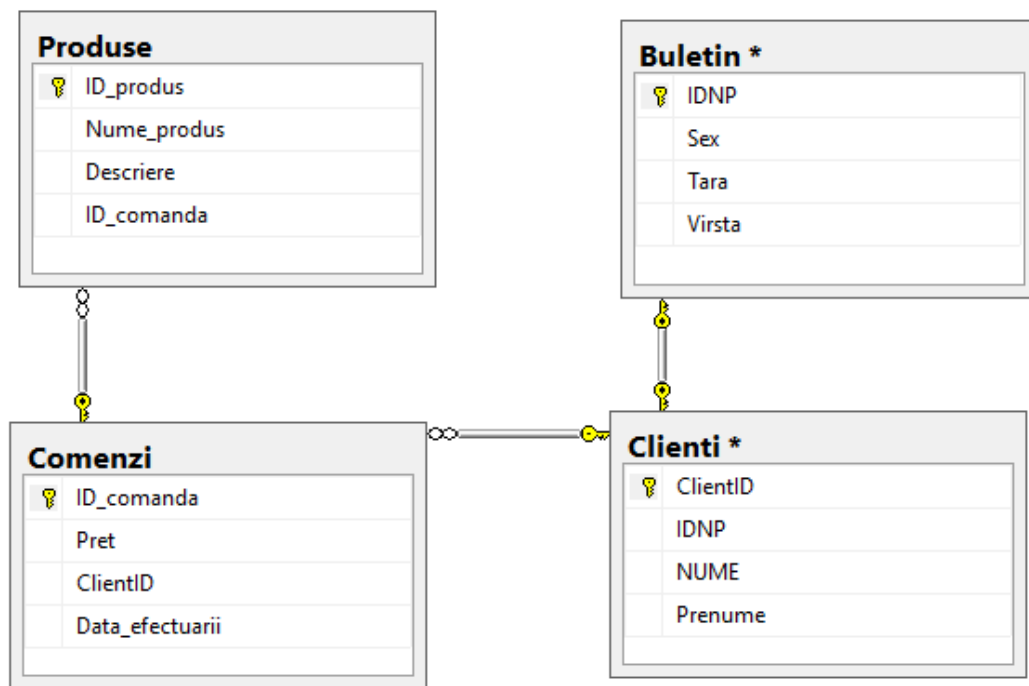


Figura 4. Diagrama bazei de date

Popularea cu tupluri a relatiilor BD

```
1. --POPULAREA TUPLURILOR
2. GO
3. INSERT INTO Buletin VALUES('1234567891001', 'm', 'Romania', 75)
4. INSERT INTO Buletin VALUES('1234567891002', 'm', 'Moldova', 42)
5. INSERT INTO Buletin VALUES('1234567891003', 'm', 'Romania', 44)
6. INSERT INTO Buletin VALUES('1234567891004', 'm', 'Moldova', 34)
7. INSERT INTO Buletin VALUES('1234567891005', 'f', 'Italia', 22)
8. INSERT INTO Buletin VALUES('1234567891006', 'f', 'Moldova', 25)
9. INSERT INTO Buletin VALUES('1234567891007', 'm', 'Moldova', 22)
10. INSERT INTO Buletin VALUES('1234567891008', 'm', 'Romania', 25)
11. INSERT INTO Buletin VALUES('1234567891009', 'm', 'Anglia', 39)
12. INSERT INTO Buletin VALUES('1234567891010', 'm', 'Franta', 60)
13. GO
14. INSERT INTO Clienti VALUES(1001, '1234567891001', 'Creanga', 'Vasile')
15. INSERT INTO Clienti VALUES(1002, '1234567891002', 'Zagareanu', 'Denis')
16. INSERT INTO Clienti VALUES(1003, '1234567891003', 'Sisea', 'Alexandru')
17. INSERT INTO Clienti VALUES(1004, '1234567891004', 'Ziozic', 'Andrei')
18. INSERT INTO Clienti VALUES(1005, '1234567891005', 'Cobra', 'Ana')
19. INSERT INTO Clienti VALUES(1006, '1234567891006', 'Vulpe', 'Lidia')
20. INSERT INTO Clienti VALUES(1007, '1234567891007', 'Sendrila', 'Ion')
```

```

21. INSERT INTO Clienti VALUES(1008, '1234567891008', 'Covrig', 'Adrean')
22. INSERT INTO Clienti VALUES(1009, '1234567891009', 'Ciudic', 'Mihail')
23. INSERT INTO Clienti VALUES(1010, '1234567891010', 'Magari', 'Igor')
24. GO
25. INSERT INTO Comenzi VALUES(1, 4000, 1001, '2017-10-10')
26. INSERT INTO Comenzi VALUES(2, 6000, 1002, '2017-10-11')
27. INSERT INTO Comenzi VALUES(3, 9000, 1002, '2017-10-12')
28. INSERT INTO Comenzi VALUES(4, 5000, 1003, '2017-10-13')
29. INSERT INTO Comenzi VALUES(5, 3000, 1004, '2017-10-14')
30. INSERT INTO Comenzi VALUES(6, 799, 1005, '2017-10-15')
31. INSERT INTO Comenzi VALUES(7, 2500, 1006, '2017-10-16')
32. INSERT INTO Comenzi VALUES(8, 16000, 1007, '2017-10-17')
33. INSERT INTO Comenzi VALUES(9, 20000, 1008, '2017-10-18')
34. INSERT INTO Comenzi VALUES(10, 25000, 1009, '2017-10-19')
35. INSERT INTO Comenzi VALUES(11, 14000, 1010, '2017-10-20')
36. GO
37. INSERT INTO Produse VALUES(1, 'Samsung', 'Galaxy S8', 1)
38. INSERT INTO Produse VALUES(2, 'Nokia', 'Lumia N9', 2 )
39. INSERT INTO Produse VALUES(3, 'Lenovo', 'Vibe X', 3 )
40. INSERT INTO Produse VALUES(4, 'Samsung', 'Galaxy Edge 7', 3 )
41. INSERT INTO Produse VALUES(5, 'Huawei', 'Y200', 4 )
42. INSERT INTO Produse VALUES(6, 'Sony', 'Z2', 5 )
43. INSERT INTO Produse VALUES(7, 'Samsung', 'NOTE 2', 6 )
44. INSERT INTO Produse VALUES(8, 'Alcatel', 'N300', 7 )
45. INSERT INTO Produse VALUES(9, 'Cubot', 'C100', 8 )
46. INSERT INTO Produse VALUES(10, 'Apple', 'Iphone X', 9 )
47. INSERT INTO Produse VALUES(11, 'Apple', 'Iphone 8', 10 )
48. GO

```

	ClientID	IDNP	NUME	Prenume
▶	1001	1234567891001	Creanga	Vasile
	1002	1234567891002	Zagareanu	Denis
	1003	1234567891003	Sisea	Alexandru
	1004	1234567891004	Ziozic	Andrei
	1005	1234567891005	Cobra	Ana
	1006	1234567891006	Vulpe	Lidia
	1007	1234567891007	Sendrila	Ion
	1008	1234567891008	Covrig	Adrean
	1009	1234567891009	Ciudic	Mihail
	1010	1234567891010	Magari	Igor
*	NULL	NULL	NULL	NULL

Figura 5. Exemplu de tabel populat

Crearea schemelor

```
1. --scheme
2. IF EXISTS(SELECT sys.schemas.name FROM sys.schemas WHERE sys.schemas.name = 'Clienti')
3. DROP SCHEMA Clienti
4. GO
5. CREATE SCHEMA Clienti
6. GO
7. ALTER SCHEMA Clienti TRANSFER dbo.Clienti
8. GO
9. IF EXISTS(SELECT sys.schemas.name FROM sys.schemas WHERE sys.schemas.name = 'Buletin')
10. DROP SCHEMA Buletin
11. GO
12. CREATE SCHEMA Buletin
13. GO
14. ALTER SCHEMA Buletin TRANSFER dbo.Buletin
```

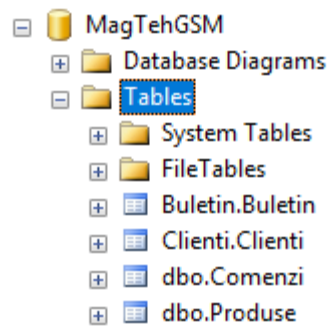


Figura 6. Schemele create

Crearea Utilizatorilor

```
1. --crearea userilor
2. USE master
3. GO
4. -- Primul user
5. IF EXISTS (SELECT loginname FROM master.dbo.syslogins WHERE name = 'User1')
6. BEGIN
7. DROP LOGIN User1;
8. END
9. GO
10. USE MagTehGSM;
11. GO
12. IF EXISTS (SELECT * FROM sys.database_principals WHERE name = 'User1')
13. BEGIN
14. DROP LOGIN User1;
15. END
16. CREATE LOGIN User1 WITH PASSWORD = '12345';
17. CREATE USER User1 FROM LOGIN User1;
18. GO
19. GRANT SELECT TO User1;
```

```

20. GO
21. --al doilea user
22. IF EXISTS (SELECT loginname FROM master.dbo.syslogins WHERE name = 'admin1')
23. BEGIN
24. DROP LOGIN admin1;
25. END
26. GO
27. USE MagTehGSM;
28. GO
29. IF EXISTS (SELECT * FROM sys.database_principals WHERE name = 'admin1')
30. BEGIN
31. DROP USER admin1;
32. END
33. CREATE LOGIN admin1 WITH PASSWORD = '54321';
34. CREATE USER admin1 FROM LOGIN admin1;
35. GO
36. GRANT SELECT, UPDATE, INSERT, DELETE TO admin1;

```

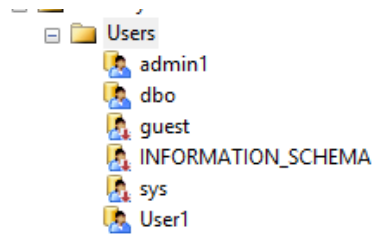


Figura 7. Utilizatorii creați

1 SELECT *

2 FROM Clienti.Clienti

100 %

Results Messages

	ClientID	IDNP	NUME	Prenume
1	1001	1234567891001	Creanga	Vasile
2	1002	1234567891002	Zagareanu	Denis
3	1003	1234567891003	Sisea	Alexandru
4	1004	1234567891004	Ziozic	Andrei
5	1005	1234567891005	Cobra	Ana
6	1006	1234567891006	Vulpe	Lidia
7	1007	1234567891007	Sendrila	Ion
8	1008	1234567891008	Covrig	Adrean
9	1009	1234567891009	Ciudic	Mihail
10	1010	1234567891010	Magari	Igor

Figura 8. Exemplu pentru utilizarea User1

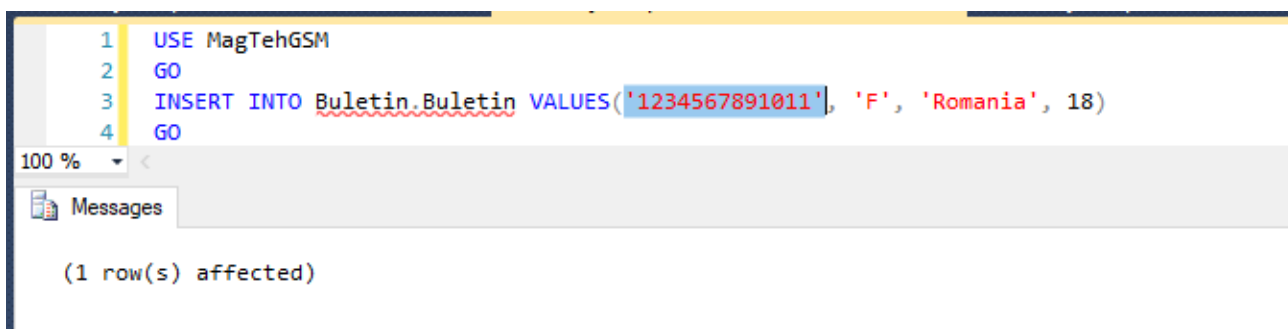


Figura 9. Exemplu pentru utilizarea admin1

Crearea viziunilor

```

1. --viziuni
2. USE MagTehGSM
3. GO
4. --prima viziune
5. IF OBJECT_ID('View1', 'V') IS NOT NULL
6. DROP VIEW View1
7. GO
8. CREATE VIEW View1 WITH SCHEMABINDING AS
9. SELECT ClientID, NUME, Prenume
10. FROM Clienti.Clienti
11. WHERE IDNP = '1234567891005'
12. GO
13. SELECT * FROM View1
14. GO
15. --a doua viziune
16. IF OBJECT_ID('View2', 'V') IS NOT NULL
17. DROP VIEW View2
18. GO
19. CREATE VIEW View2 AS
20. SELECT ID_produs, Nume_produs, Descriere FROM Produse
21. WITH CHECK OPTION
22. GO
23. UPDATE View2
24. SET View2.Descriere = 'Iphone SE' WHERE ID_produs = 11
25. SELECT * FROM View2

```

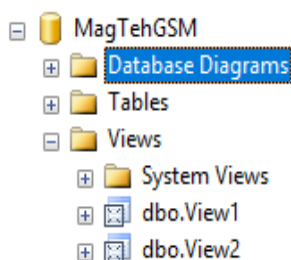


Figura 10. Viziunile create

Crearea sinonimelor

```
1. --sinonime
2. --Sinonim pt buletin
3. GO
4. CREATE SYNONYM Bul for Buletin.Buletin
5. GO
6. --sinonim pt clienti
7. CREATE SYNONYM Cli for Clienti.Clienti
8. GO
9. --SINONIM PENTRU COMENZI
10. CREATE SYNONYM Com for dbo.Comenzi
11. GO
12. --SINONIM PT PRODUSE
13. CREATE SYNONYM Prod for dbo.Produse
14. GO
```

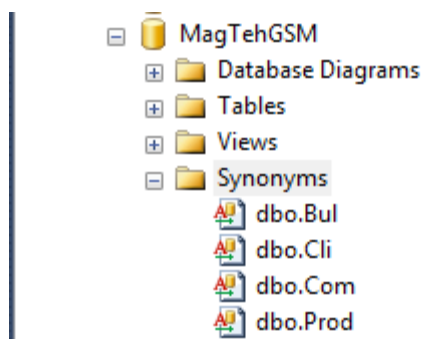


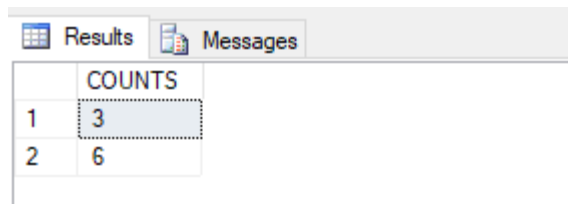
Figura 11. Crearea Sinonimelor

3. Interogarea Bazei de Date

Crearea interogărilor ce exprimă operațiile algebrei relaționale

Să se selecteze numărul de produse de la compania Samsung și numărul de comenzi efectuate cu preț mai mare de 5000\$. Rezultatul să se afișeze în reuniune.

```
1. SELECT COUNT(ID_produs) AS COUNTS FROM Produse
2. WHERE Nume_produs = 'Samsung'
3. UNION
4. SELECT Count(ID_comanda) FROM Comenzi
5. WHERE Pret > 5000
```

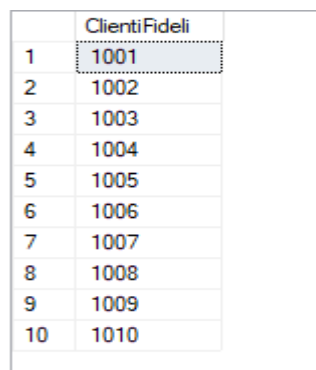


	COUNTS
1	3
2	6

Figura 3.1. Uniunea

Să se selecteze clienții care au efectuat comenzi de la magazinul de tehnică online.

```
1. SELECT ClientID AS ClientiFideli FROM Cli
2. INTERSECT
3. SELECT ClientID FROM Com
```



	ClientiFideli
1	1001
2	1002
3	1003
4	1004
5	1005
6	1006
7	1007
8	1008
9	1009
10	1010

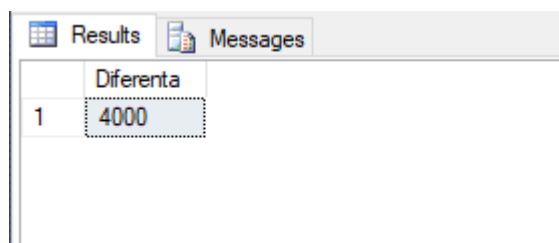
Figura 3.2. Intersect

Sa se determine diferența de preț dintre produsele de la Samsung si Huawei.

```

1. SELECT Pret AS [Diferenta] FROM Comenzi
2. WHERE ID_comanda = 1
3. EXCEPT
4. SELECT Pret FROM Comenzi
5. WHERE ID_comanda = 5

```



	Diferenta
1	4000

Figura 3.3 Diferența

Să se combine produsele cu prețul acestora din tabela comenzi.

```

1. ;WITH
2.     Produs AS(
3.         SELECT ID_produs, Nume_produs, Descriere
4.         FROM Prod
5.     ),
6.     Comanda AS(
7.         SELECT Pret
8.         FROM Com
9.     )
10. SELECT * FROM Produs CROSS JOIN Comanda

```

	ID_produs	Nume_produs	Descriere	Pret
1	1	Samsung	Galaxy S8	4000
2	2	Nokia	Lumia N9	4000
3	3	Lenovo	Vibe X	4000
4	4	Samsung	Galaxy Edge 7	4000
5	5	Huawei	Y200	4000
6	6	Sony	Z2	4000
7	7	Samsung	NOTE 2	4000
8	8	Alcatel	N300	4000
9	9	Cubot	C100	4000
10	10	Apple	Iphone X	4000
11	11	Apple	Iphone SE	4000
12	1	Samsung	Galaxy S8	6000
13	2	Nokia	Lumia N9	6000
14	3	Lenovo	Vibe X	6000
15	4	Samsung	Galaxy Edge 7	6000
16	5	Huawei	Y200	6000
17	6	Sony	Z2	6000
18	7	Samsung	NOTE 2	6000
19	8	Alcatel	N300	6000
20	9	Cubot	C100	6000
21	10	Apple	Iphone X	6000
22	11	Apple	Iphone SE	6000
23	1	Samsung	Galaxy S8	9000

Figura 3.4 Produsul Cartezian

Să se afișeze numele și prenumele clienților care au efectuat comenzi în magazinul de tehnică online.

1. **SELECT** ID_comanda, Pret, Nume, Prenume
2. **FROM** Com A, Cli B
3. **WHERE** A.ClientID = B.ClientID

Results		Messages			
	ID_comanda	Pret	Nume	Prenume	
1	1	4000	Creanga	Vasile	
2	2	6000	Zagareanu	Denis	
3	3	9000	Zagareanu	Denis	
4	4	5000	Sisea	Alexandru	
5	5	3000	Ziozic	Andrei	
6	6	799	Cobra	Ana	
7	7	2500	Vulpe	Lidia	
8	8	16000	Sendrila	Ion	
9	9	20000	Covrig	Adrean	
10	10	25000	Ciudic	Mihail	
11	11	14000	Magari	Igor	

Figura 3.5 Teta Joncțiune

Să se selecteze clienții și buletinele lor.

1. **SELECT** ClientID, Nume, Prenume, Cli.IDNP, Sex, Tara, Virsta
2. **FROM** Cli **INNER JOIN** Bul **ON** Cli.IDNP = Bul.IDNP

	ClientID	Nume	Prenume	IDNP	Sex	Tara	Virsta
1	1001	Creanga	Vasile	1234567891001	m	Romania	75
2	1002	Zagareanu	Denis	1234567891002	m	Moldova	42
3	1003	Sisea	Alexandru	1234567891003	m	Romania	44
4	1004	Ziozic	Andrei	1234567891004	m	Moldova	34
5	1005	Cobra	Ana	1234567891005	f	Itala	22
6	1006	Vulpe	Lidia	1234567891006	f	Moldova	25
7	1007	Sendrila	Ion	1234567891007	m	Moldova	22
8	1008	Covrig	Adrean	1234567891008	m	Romania	25
9	1009	Ciudic	Mihail	1234567891009	m	Anglia	39
10	1010	Magari	Igor	1234567891010	m	Franta	60

Figura 3.6 Joncțiunea naturala

Joncțiune externă

1. **SELECT** ClientID, Nume, Prenume, Cli.IDNP, Sex, Tara, Virsta
2. **FROM** Cli **LEFT JOIN** Bul **ON** Cli.IDNP = Bul.IDNP

	ClientID	Nume	Prenume	IDNP	Sex	Tara	Virsta
1	1001	Creanga	Vasile	1234567891001	m	Romania	75
2	1002	Zagareanu	Denis	1234567891002	m	Moldova	42
3	1003	Sisea	Alexandru	1234567891003	m	Romania	44
4	1004	Ziozic	Andrei	1234567891004	m	Moldova	34
5	1005	Cobra	Ana	1234567891005	f	Itala	22
6	1006	Vulpe	Lidia	1234567891006	f	Moldova	25
7	1007	Sendrila	Ion	1234567891007	m	Moldova	22
8	1008	Covrig	Adrean	1234567891008	m	Romania	25
9	1009	Ciudic	Mihail	1234567891009	m	Anglia	39
10	1010	Magari	Igor	1234567891010	m	Franta	60

Figura 3.7. Left Join

Să se afișeze numele și prenumele clienților care au efectuat comenzi în magazinul de tehnică online.

1. **SELECT** ID_comanda, Pret, Nume, Prenume
2. **FROM** Com A **RIGHT JOIN** Cli B **ON** A.ClientID = B.ClientID

	ID_comanda	Pret	Nume	Prenume
1	1	4000	Creanga	Vasile
2	2	6000	Zagareanu	Denis
3	3	9000	Zagareanu	Denis
4	4	5000	Sisea	Alexandru
5	5	3000	Ziozic	Andrei
6	6	799	Cobra	Ana
7	7	2500	Vulpe	Lidia
8	8	16000	Sendrila	Ion
9	9	20000	Covrig	Adrean
10	10	25000	Ciudic	Mihail
11	11	14000	Magari	Igor

Figura 3.8. Right Join

Semijoncțiunea

1. **SELECT DISTINCT** ID_comanda, Pret, B.ClientID, Nume, Pnume
2. **FROM** Com A **join** Cli B **ON** A.ClientID = B.ClientID

	ID_comanda	Pret	ClientID	Nume	Pnume
1	1	4000	1001	Creanga	Vasile
2	2	6000	1002	Zagareanu	Denis
3	3	9000	1002	Zagareanu	Denis
4	4	5000	1003	Sisea	Alexandru
5	5	3000	1004	Ziozic	Andrei
6	6	799	1005	Cobra	Ana
7	7	2500	1006	Vulpe	Lidia
8	8	16000	1007	Sendrila	Ion
9	9	20000	1008	Covrig	Adrean
10	10	25000	1009	Ciudic	Mihail
11	11	14000	1010	Magari	Igor

Figura 3.9. Semijoncțiunea

Să se afișeze clienții care au comenzi mai scumpe de 4000. Deasemenea informațiile despre ei.

1. **SELECT** Diviziunea1.ClientID, IDNP, NUME, Pnume, ID_comanda, Pret, Data_efectuarii
2. **FROM**
3. (**SELECT** * **FROM** Cli) **AS** [Diviziunea1]
4. **JOIN**
5. (**SELECT** * **FROM** Com **WHERE** Pret > 4000) **AS** Diviziunea2
6. **ON** [Diviziunea1].ClientID = [Diviziunea2].ClientID

	ClientID	IDNP	NUME	Pnume	ID_comanda	Pret	Data_efectuarii
1	1002	1234567891002	Zagareanu	Denis	2	6000	2017-10-11
2	1002	1234567891002	Zagareanu	Denis	3	9000	2017-10-12
3	1003	1234567891003	Sisea	Alexandru	4	5000	2017-10-13
4	1007	1234567891007	Sendrila	Ion	8	16000	2017-10-17
5	1008	1234567891008	Covrig	Adrean	9	20000	2017-10-18
6	1009	1234567891009	Ciudic	Mihail	10	25000	2017-10-19
7	1010	1234567891010	Magari	Igor	11	14000	2017-10-20

Figura 3.10. Diviziuni

Crearea interogărilor cu funcții de agregare

Afișează prețul maxim, minim, venitul și suma acestuia. Numărul de comenzi.

```
1. SELECT MAX(PRET) AS Maxim, MIN(Pret) AS Minim,  
2. SUM(Pret) AS Suma, AVG(Pret) AS Venit,  
3. count(*) AS Comenzi  
4. FROM Com
```

	Maxim	Minim	Suma	Venit	(No column name)
1	25000	799	105299	9572	11

Figua 3.11. funcții de agregare

Afișează vârsta minimă maximă, suma și venitul. De asemenea și numărul de clienți.

```
1. SELECT MAX(Virsta) AS Maxim, MIN(Virsta) AS Minim,  
2. SUM(Virsta) AS Suma, AVG(Virsta) AS Venit,  
3. count(*) AS clients  
4. FROM Bul
```

	Maxim	Minim	Suma	Venit	clients
1	75	22	388	38	10

Figura 3.12. funcții de agregare

```
1. SELECT COUNT(1) FROM Prod
```

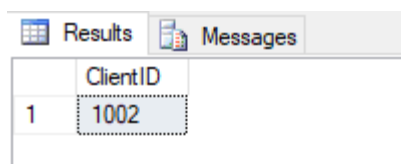
	(No column name)
1	11

Figura.3.13. Count

Crearea interogărilor cu grupare

Să se găsească clienții care au efectuat cel puțin 2 comenzi.

```
1. SELECT ClientID FROM Com
2. WHERE Pret > 1000
3. GROUP BY ClientID
4. Having Count(Pret) >= 2
```



ClientID
1002

Figura 3.14 Grupare

Să se găsească prețul mediu a comenzilor pentru fiecare Client. Să se afișeze clientul și prețul.

```
1. SELECT AVG(Pret) as PretMediu, Com.ClientID FROM Com
2. join Cli ON Cli.ClientID = Com.ClientID
3. GROUP BY Com.ClientID
```

	PretMediu	ClientID
1	4000	1001
2	7500	1002
3	5000	1003
4	3000	1004
5	799	1005
6	2500	1006
7	16000	1007
8	20000	1008
9	25000	1009
10	14000	1010

Figura 3.15 Grupare

Subinterogări

Să se afișeze comezile clienților a căror nume începe cu liter 'C'.

```
1. SELECT ID_comanda, Data_efectuării, Pret
2. FROM Com
3. WHERE ClientID IN (SELECT ClientID FROM Cli WHERE Nume LIKE 'C%')
4. ORDER BY Pret
```

	ID_comanda	Data_efectuării	Pret
1	6	2017-10-15	799
2	1	2017-10-10	4000
3	9	2017-10-18	20000
4	10	2017-10-19	25000

Figura 3.16. Folosirea IN

Să se găsească clienții care sunt în lista de Comenzi.

```
1. SELECT Nume, Prenume, IDNP
2. FROM Cli
3. WHERE EXISTS (SELECT ClientID FROM Com WHERE ClientID = Cli.ClientID);
```

	Nume	Prenume	IDNP
1	Creanga	Vasile	1234567891001
2	Zagareanu	Denis	1234567891002
3	Sisea	Alexandru	1234567891003
4	Ziozic	Andrei	1234567891004
5	Cobra	Ana	1234567891005
6	Vulpe	Lidia	1234567891006
7	Sendrila	Ion	1234567891007
8	Covrig	Adrean	1234567891008
9	Ciudic	Mihail	1234567891009
10	Magari	Igor	1234567891010

Figura 3.17. Folosirea EXISTS

Să se găsească clienții care au efectuat comenzi mai scumpe de 4000.

```
1. SELECT Nume, Prenume, IDNP
2. FROM Cli
3. WHERE ClientID = ANY (SELECT ClientID FROM Com WHERE Pret > 4000);
```

	Nume	Prenume	IDNP
1	Zagareanu	Denis	1234567891002
2	Sisea	Alexandru	1234567891003
3	Sendrila	Ion	1234567891007
4	Covrig	Adrean	1234567891008
5	Ciudic	Mihail	1234567891009
6	Magari	Igor	1234567891010

Figura 3.18. Folosirea ANY

Subinterogări în alte locuri decât clauza WHERE

Să se găsească datele despre fiecare client și numărul său de comenzi.

```
1. SELECT Nume, Prenume,
2. ClientID = (SELECT Count(*) FROM Com WHERE Com.ClientID = C.ClientID)
3. FROM Cli C
```

	Nume	Prenume	ClientID
1	Creanga	Vasile	1
2	Zagareanu	Denis	2
3	Sisea	Alexandru	1
4	Ziozic	Andrei	1
5	Cobra	Ana	1
6	Vulpe	Lidia	1
7	Sendrila	Ion	1
8	Covrig	Adrean	1
9	Ciudic	Mihail	1
10	Magari	Igor	1

Figura 3.19. Subinterogare

Crearea interogărilor de actualizare a relațiilor

Să se scrie o interogare de ștergere condiționată a tuplurilor.

```
1. USE MagTehGSM
2. GO
3. IF OBJECT_ID('stergere_client', 'TR') IS NOT NULL DROP TRIGGER stergere_client
4. GO
5. CREATE TRIGGER stergere_student ON Buletin.Buletin
6. AFTER DELETE
7. AS
8. BEGIN
9.     DELETE FROM Buletin.Buletin
10.    WHERE IDNP = (SELECT MAX(Virsta) FROM Buletin.Buletin)
11. END
12. GO
```

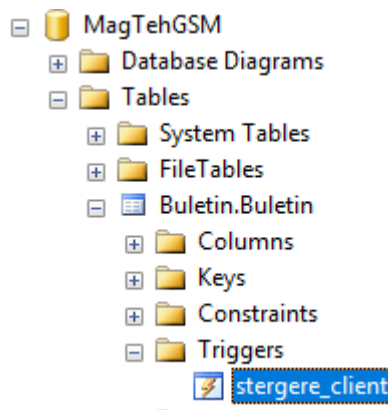


Figura 3.20. Ștergerea declanșatorului

Să se scrie o interogare de actualizare condiționată a tuplurilor.

```
1. USE MagTehGSM
2. GO
3. IF OBJECT_ID('inregistrare_noua', 'TR') IS NOT NULL
4. DROP TRIGGER inregistrare_noua;
5. GO
6. CREATE TRIGGER inregistrare_noua ON Produse
7. AFTER UPDATE
8. AS SET NOCOUNT ON;
9. IF UPDATE(Descriere)
10. SELECT 'Produsul cu ID-ul ' + RTRIM(CAST(inserted.ID_produs AS tinyint)) +
11. 'are o descriere noua ' + CAST(inserted.Descriere AS varchar(50))
12. FROM inserted
13. SELECT 'Descrierea veche era ' + RTRIM(CAST(deleted.Descriere AS varchar(50)))
14. FROM deleted
15. GO
16. UPDATE produse SET Descriere = 'Galaxy S9' WHERE ID_produs = 1;
```

	(No column name)
1	Produsul cu ID-ul 1 are o descriere noua Galaxy S9
	(No column name)
1	Descrierea veche era Galaxy S8

Figura 3.21. Crearea declanșatorului

Crearea procedurilor stocate

Sondaj categorie de vîrstă a clienților.

```

1. USE MagTehGSM
2. GO
3. IF OBJECT_ID ('sondaj', 'P') IS NOT NULL DROP PROCEDURE sondaj
4. GO
5. CREATE PROCEDURE sondaj
6. @Virsta INT
7. AS
8. SELECT c.IDNP, c.Nume, c.Prenume, b.Tara, b.Virsta
9. FROM Clienti.Clienti AS c JOIN Buletin.Buletin AS b ON c.IDNP = b.IDNP
10. WHERE Virsta < @Virsta
11. GO
12. EXEC sondaj 40

```

	IDNP	Nume	Prenume	Tara	Virsta
1	1234567891004	Ziozic	Andrei	Moldova	34
2	1234567891005	Cobra	Ana	Itala	22
3	1234567891006	Vulpe	Lidia	Moldova	25
4	1234567891007	Sendrila	Ion	Moldova	22
5	1234567891008	Covrig	Adrean	Romania	25
6	1234567891009	Ciudic	Mihail	Anglia	39

Figura 3.22. Rezultatul procedurii

Lista de clienți care au efectuat comenzi începînd cu prețul 15 000 și mai mult.

```
1. USE MagTehGSM
2. GO
3. IF OBJECT_ID ('GetSuperComands', 'P') IS NOT NULL DROP PROCEDURE GetSuperComands
4. GO
5. CREATE PROCEDURE GetSuperComands
6. @pret INT
7. AS
8. SELECT c.IDNP, c.Nume, c.Prenume, com.ID_comanda, com.Pret
9. FROM Comenzi com join Clienti.Clienti c ON com.ClientID = c.ClientID
10. WHERE Pret >= @pret
11. GO
12. EXEC GetSuperComands 15000
```

	IDNP	Nume	Prenume	ID_comanda	Pret
1	1234567891007	Sendrila	Ion	8	16000
2	1234567891008	Covrig	Adrean	9	20000
3	1234567891009	Ciudic	Mihail	10	25000

Figura 3.23. Rezultatul procedurii

Concluzie

În lucrarea dată am efectuat sinteza bazei de date MagTehOnline. La început am creat un tabel cu atributele din bază și am stabilit dependențele funcționale, după care le-am codificat. Am normalizat baza de date magazin de tehnică online și am eliminat dependențele redundante, în cazul nostru am avut doar o singură redundanță. Normalizarea este o tehnică de generare a unor relații cu proprietățile dorite, în scopul memorării corecte a datelor unei întreprinderi. În următoarea etapă am aplicat algoritmul reducerii la stânga și am stabilit clasele de echivalență. În mulțimea J nu am dependențe, deci nu sunt modificări în clasele de echivalență. După pasul dat am construit schemele bazei de date. Parțial întreaga mulțime de dependențe funcționale a fost conservată. Joncțiunea este satisfăcută fără pierderi. Economia de memorie se face prin eliminarea atributelor duble, adăugând chei externe și grupînd atributele pe tabele. Totodată datele vor rămîne întregi după operații de modificare a datelor (Update, Insert) din tabel din cauza că au fost eliminate redundanțele. În acest fel vom scăpa de date inconsistente. După crearea schemelor, am decodificat atributele și am obținut relațiile într-un tabel, cu cheia primară în fiecare relație. Avem 2 tipuri de asociere în baza de date: 1 la 1 și 1 la mulți. Baza a fost normalizată cu succes. Relațiile sale sunt identice cu cele din diagram de baza.

După părerea mea normalizarea reprezintă un element foarte important, deoarece ea garantează funcționarea efectivă a bazei de date la un volum de date mai mare.

ANEXE

```
--Crearea Bazei de date MagTehGSM
USE master
GO
IF DB_ID('MagTehGSM') IS NOT NULL
DROP DATABASE MagTehGSM
GO
CREATE DATABASE MagTehGSM
ON(
NAME = MagTehGSM,
FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\MagTehGSM.mdf',
SIZE = 5MB,
MAXSIZE = UNLIMITED,
FILEGROWTH = 1MB)
LOG ON(
NAME = MagTehGSM_log,
FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\MagTehGSM_log.ldf',
SIZE = 1MB,
MAXSIZE = UNLIMITED,
FILEGROWTH = 5MB)
GO

--Creaza tabele

use MagTehGSM
--relatia produse
IF OBJECT_ID('Produse', 'U') IS NOT NULL
DROP TABLE Produse
CREATE TABLE Produse(
    ID_produs int NOT NULL,
    Nume_produs nvarchar(50) NOT NULL,
    Descriere varchar(50) NOT NULL,
    ID_comanda int NOT NULL,
    PRIMARY KEY (ID_produs)
)
--relatie comenzi
IF OBJECT_ID('Comenzi', 'U') IS NOT NULL
DROP TABLE Comenzi
CREATE TABLE Comenzi(
    ID_comanda int NOT NULL,
    Pret int NOT NULL,
    ClientID int NOT NULL,
    Data_efectuarii date NULL,
    PRIMARY KEY (ID_comanda)
)
--relatie clienti
IF OBJECT_ID('Clienti', 'U') IS NOT NULL
DROP TABLE Clienti
CREATE TABLE Clienti(
    ClientID int NOT NULL,
    IDNP varchar(13) NULL,
    NUME nvarchar(50) NULL,
    Prenume nvarchar(50) NULL,
    PRIMARY KEY (ClientID)
)
```

```

--relatie buletin
IF OBJECT_ID('Buletin', 'U') IS NOT NULL
DROP TABLE Buletin
CREATE TABLE Buletin(
    IDNP varchar(13) NOT NULL,
    Sex char(1) NULL,
    Tara nvarchar(20) NULL,
    Virsta int NULL,
    PRIMARY KEY (IDNP)
)

--Chei externe
USE MagTehGSM
--Produce
ALTER TABLE Produce ADD CONSTRAINT[FK_ComenziProduce]
FOREIGN KEY(ID_comanda) REFERENCES Comenzi (ID_comanda)
ON UPDATE CASCADE;

--Comenzi
ALTER TABLE Comenzi ADD CONSTRAINT[FK_ClientiComenzi]
FOREIGN KEY(ClientID) REFERENCES Clienti(ClientID)
ON UPDATE CASCADE;
--Clienti
ALTER TABLE Clienti ADD CONSTRAINT [FK_clienti_buletin] FOREIGN KEY(IDNP)
REFERENCES Buletin (IDNP)
ON UPDATE CASCADE;
--CHEIE UNICA
ALTER TABLE Clienti
ADD CONSTRAINT UC_Clienti UNIQUE(IDNP)

USE MagTehGSM
--POPULAREA TUPLURILOR
GO
INSERT INTO Buletin VALUES('1234567891001', 'm', 'Romania', 75)
INSERT INTO Buletin VALUES('1234567891002', 'm', 'Moldova', 42)
INSERT INTO Buletin VALUES('1234567891003', 'm', 'Romania', 44)
INSERT INTO Buletin VALUES('1234567891004', 'm', 'Moldova', 34)
INSERT INTO Buletin VALUES('1234567891005', 'f', 'Itala', 22)
INSERT INTO Buletin VALUES('1234567891006', 'f', 'Moldova', 25)
INSERT INTO Buletin VALUES('1234567891007', 'm', 'Moldova', 22)
INSERT INTO Buletin VALUES('1234567891008', 'm', 'Romania', 25)
INSERT INTO Buletin VALUES('1234567891009', 'm', 'Anglia', 39)
INSERT INTO Buletin VALUES('1234567891010', 'm', 'Franta', 60)
GO
INSERT INTO Clienti VALUES(1001, '1234567891001', 'Creanga', 'Vasile')
INSERT INTO Clienti VALUES(1002, '1234567891002', 'Zagareanu', 'Denis')
INSERT INTO Clienti VALUES(1003, '1234567891003', 'Sisea', 'Alexandru')
INSERT INTO Clienti VALUES(1004, '1234567891004', 'Ziozic', 'Andrei')
INSERT INTO Clienti VALUES(1005, '1234567891005', 'Cobra', 'Ana')
INSERT INTO Clienti VALUES(1006, '1234567891006', 'Vulpe', 'Lidia')
INSERT INTO Clienti VALUES(1007, '1234567891007', 'Sendrila', 'Ion')
INSERT INTO Clienti VALUES(1008, '1234567891008', 'Covrig', 'Adrean')
INSERT INTO Clienti VALUES(1009, '1234567891009', 'Ciudic', 'Mihail')
INSERT INTO Clienti VALUES(1010, '1234567891010', 'Magari', 'Igor')
GO
INSERT INTO Comenzi VALUES(1, 4000, 1001, '2017-10-10')
INSERT INTO Comenzi VALUES(2, 6000, 1002, '2017-10-11')
INSERT INTO Comenzi VALUES(3, 9000, 1002, '2017-10-12')

```

```

INSERT INTO Comenzi VALUES(4, 5000, 1003, '2017-10-13')
INSERT INTO Comenzi VALUES(5, 3000, 1004, '2017-10-14')
INSERT INTO Comenzi VALUES(6, 799, 1005, '2017-10-15')
INSERT INTO Comenzi VALUES(7, 2500, 1006, '2017-10-16')
INSERT INTO Comenzi VALUES(8, 16000, 1007, '2017-10-17')
INSERT INTO Comenzi VALUES(9, 20000, 1008, '2017-10-18')
INSERT INTO Comenzi VALUES(10, 25000, 1009, '2017-10-19')
INSERT INTO Comenzi VALUES(11, 14000, 1010, '2017-10-20')
GO
INSERT INTO Produse VALUES(1, 'Samsung', 'Galaxy S8', 1)
INSERT INTO Produse VALUES(2, 'Nokia', 'Lumia N9', 2 )
INSERT INTO Produse VALUES(3, 'Lenovo', 'Vibe X', 3 )
INSERT INTO Produse VALUES(4, 'Samsung', 'Galaxy Edge 7', 3 )
INSERT INTO Produse VALUES(5, 'Huawei', 'Y200', 4 )
INSERT INTO Produse VALUES(6, 'Sony', 'Z2', 5 )
INSERT INTO Produse VALUES(7, 'Samsung', 'NOTE 2', 6 )
INSERT INTO Produse VALUES(8, 'Alcatel', 'N300', 7 )
INSERT INTO Produse VALUES(9, 'Cubot', 'C100', 8 )
INSERT INTO Produse VALUES(10, 'Apple', 'Iphone X', 9 )
INSERT INTO Produse VALUES(11, 'Apple', 'Iphone 8', 10 )
GO

--scheme
IF EXISTS(SELECT sys.schemas.name FROM sys.schemas WHERE sys.schemas.name = 'Clienti')
DROP SCHEMA Clienti
GO
CREATE SCHEMA Clienti
GO
ALTER SCHEMA Clienti TRANSFER dbo.Clienti
GO
IF EXISTS(SELECT sys.schemas.name FROM sys.schemas WHERE sys.schemas.name = 'Buletin')
DROP SCHEMA Buletin
GO
CREATE SCHEMA Buletin
GO
ALTER SCHEMA Buletin TRANSFER dbo.Buletin

--crearea userilor
USE master
GO
-- Primul user
IF EXISTS (SELECT loginname FROM master.dbo.syslogins WHERE name = 'User1')
BEGIN
DROP LOGIN User1;
END
GO
USE MagTehGSM;
GO
IF EXISTS (SELECT * FROM sys.database_principals WHERE name = 'User1')
BEGIN
DROP LOGIN User1;
END
CREATE LOGIN User1 WITH PASSWORD = '12345';
CREATE USER User1 FROM LOGIN User1;
GO
GRANT SELECT TO User1;
GO
--al doilea user

```

```

IF EXISTS (SELECT loginname FROM master.dbo.syslogins WHERE name = 'admin1')
BEGIN
DROP LOGIN admin1;
END
GO
USE MagTehGSM;
GO
IF EXISTS (SELECT * FROM sys.database_principals WHERE name = 'admin1')
BEGIN
DROP USER admin1;
END
CREATE LOGIN admin1 WITH PASSWORD = '54321';
CREATE USER admin1 FROM LOGIN admin1;
GO
GRANT SELECT, UPDATE, INSERT, DELETE TO admin1;

--viziuni
USE MagTehGSM
GO
--prima viziune
IF OBJECT_ID('View1', 'V') IS NOT NULL
DROP VIEW View1
GO
CREATE VIEW View1 WITH SCHEMABINDING AS
SELECT ClientID, NUME, Prenume
FROM Clienti.Clienti
WHERE IDNP = '1234567891005'
GO
SELECT * FROM View1
GO
--a doua viziune
IF OBJECT_ID('View2', 'V') IS NOT NULL
DROP VIEW View2
GO
CREATE VIEW View2 AS
SELECT ID_produs, Nume_produs, Descriere FROM Produce
WITH CHECK OPTION
GO
UPDATE View2
SET View2.Descriere = 'Iphone SE' WHERE ID_produs = 11
SELECT * FROM View2

--sinonime
--Sinonim pt buletin
GO
CREATE SYNONYM Bul for Buletin.Buletin
GO
--sinonim pt clienti
CREATE SYNONYM Cli for Clienti.Clienti
GO
--SINONIM PENTRU COMENZI
CREATE SYNONYM Com for dbo.Comenzi
GO
--SINONIM PT PRODUSE
CREATE SYNONYM Prod for dbo.Produce
GO

```



```

USE MagTehGSM
GO
SELECT COUNT(ID_produs) AS COUNTS FROM Produe
WHERE Nume_produs = 'Samsung'
UNION
SELECT Count(ID_comanda) FROM Comenzi
WHERE Pret > 5000
GO

-----
SELECT ClientID AS ClientiFideli FROM Cli
INTERSECT
SELECT ClientID FROM Com
GO

-----
SELECT Pret AS [Diferenta] FROM Comenzi
WHERE ID_comanda = 1
EXCEPT
SELECT Pret FROM Comenzi
WHERE ID_comanda = 5
GO

-----
;WITH
    Produs AS(
        SELECT ID_produs, Nume_produs, Descriere
        FROM Prod
    ),
    Comanda AS(
        SELECT Pret
        FROM Com
    )
    SELECT * FROM Produs CROSS JOIN Comanda
GO

-----

SELECT ID_comanda, Pret, Nume,Prenume
FROM Com A,Cli B
WHERE A.ClientID = B.ClientID
GO

-----
SELECT ClientID, Nume, Prenume, Cli.IDNP, Sex, Tara, Virsta
FROM Cli INNER JOIN Bul ON Cli.IDNP = Bul.IDNP
GO

-----
SELECT ClientID, Nume, Prenume, Cli.IDNP, Sex, Tara, Virsta
FROM Cli LEFT JOIN Bul ON Cli.IDNP = Bul.IDNP
GO

-----
SELECT DISTINCT ID_comanda, Pret, Nume,Prenume
FROM Com A RIGHT join Cli B ON A.ClientID = B.ClientID
GO

-----
SELECT Diviziunea1.ClientID, IDNP, NUME, Prenume, ID_comanda, Pret, Data_efectuarii
FROM
(SELECT * FROM Cli) AS [Diviziunea1]
JOIN
(SELECT * FROM Com WHERE Pret > 4000) AS Diviziunea2
ON [Diviziunea1].ClientID = [Diviziunea2].ClientID
GO

```

```

-----
SELECT MAX(PRET) AS Maxim, MIN(Pret) AS Minim,
SUM(Pret) AS Suma, AVG(Pret) AS Venit,
count(*) AS Comenzi
FROM Com
GO

-----
SELECT MAX(Virsta) AS Maxim, MIN(Virsta) AS Minim,
SUM(Virsta) AS Suma, AVG(Virsta) AS Venit,
count(*) AS clients
FROM Bul
GO

-----
SELECT COUNT(1) FROM Prod
GO

-----
SELECT ClientID FROM Com
WHERE Pret > 1000
GROUP BY ClientID
Having Count(Pret) >= 2
GO

-----
SELECT AVG(Pret) as PretMediu, Com.ClientID FROM Com
join Cli ON Cli.ClientID = Com.ClientID
GROUP BY Com.ClientID
GO

-----
SELECT ID_comanda, Data_efectuarii, Pret
FROM Com
WHERE ClientID IN (SELECT ClientID FROM Cli WHERE NUME LIKE 'C%')
ORDER BY Pret
GO

-----
SELECT Nume, Prenume, IDNP
FROM Cli
WHERE EXISTS (SELECT ClientID FROM Com WHERE ClientID = Cli.ClientID);
GO

-----
SELECT Nume, Prenume, IDNP
FROM Cli
WHERE ClientID = ANY (SELECT ClientID FROM Com WHERE Pret > 4000);
GO

-----
SELECT Nume, Prenume,
ClientID = (SELECT Count(*) FROM Com WHERE Com.ClientID = C.ClientID)
FROM Cli C
GO

-----
USE MagTehGSM
GO
IF OBJECT_ID('stergere_client', 'TR') IS NOT NULL
DROP TRIGGER stergere_client
GO
CREATE TRIGGER stergere_client ON Buletin.Buletin
AFTER DELETE
AS
BEGIN
DELETE FROM Buletin.Buletin

```

```

WHERE IDNP = (SELECT MAX(Virsta) FROM Buletin.Buletin )
END
GO
-----
USE MagTehGSM
GO
IF OBJECT_ID('inregistrare_noua', 'TR') IS NOT NULL
DROP TRIGGER inregistrare_noua
IF EXISTS (SELECT * FROM sys.triggers WHERE object_id = OBJECT_ID('inregistrare_noua'))
DROP TRIGGER inregistrare_noua
GO
CREATE TRIGGER inregistrare_noua ON Produce
AFTER UPDATE
AS SET NOCOUNT ON;
IF UPDATE(Descriere)
SELECT 'Produsul cu ID-ul ' + RTRIM(CAST(inserted.ID_produs AS tinyint)) +
'are o descriere noua '+CAST(inserted.Descriere AS varchar(50))
FROM inserted
SELECT 'Descrierea veche era ' +RTRIM(CAST(deleted.Descriere AS varchar(50)))
FROM deleted
GO
UPDATE produse SET Descriere = 'Galaxy S9' WHERE ID_produs = 1;
GO
-----
USE MagTehGSM
GO
IF OBJECT_ID ('sondaj', 'P') IS NOT NULL DROP PROCEDURE sondaj
GO
CREATE PROCEDURE sondaj
@Virsta INT
AS
SELECT c.IDNP, c.Nume, c.Prenume, b.Tara, b.Virsta
FROM Clienti.Clienti AS c JOIN Buletin.Buletin AS b ON c.IDNP = b.IDNP
WHERE Virsta < @Virsta
GO
EXEC sondaj 40
GO
-----
USE MagTehGSM
GO
IF OBJECT_ID ('GetSuperComands', 'P') IS NOT NULL DROP PROCEDURE GetSuperComands
GO
CREATE PROCEDURE GetSuperComands
@pret INT
AS
SELECT c.IDNP, c.Nume, c.Prenume, com.ID_comanda, com.Pret
FROM Comenzi com join Clienti.Clienti c ON com.ClientID = c.ClientID
WHERE Pret >= @pret
GO
EXEC GetSuperComands 15000
GO

```