

**PROIECT BAZE DE DATE
GESTIUNEA UNUI MAGAZIN DE
CALCULATOARE**

DUMITRU ANDREEA-EMILIA

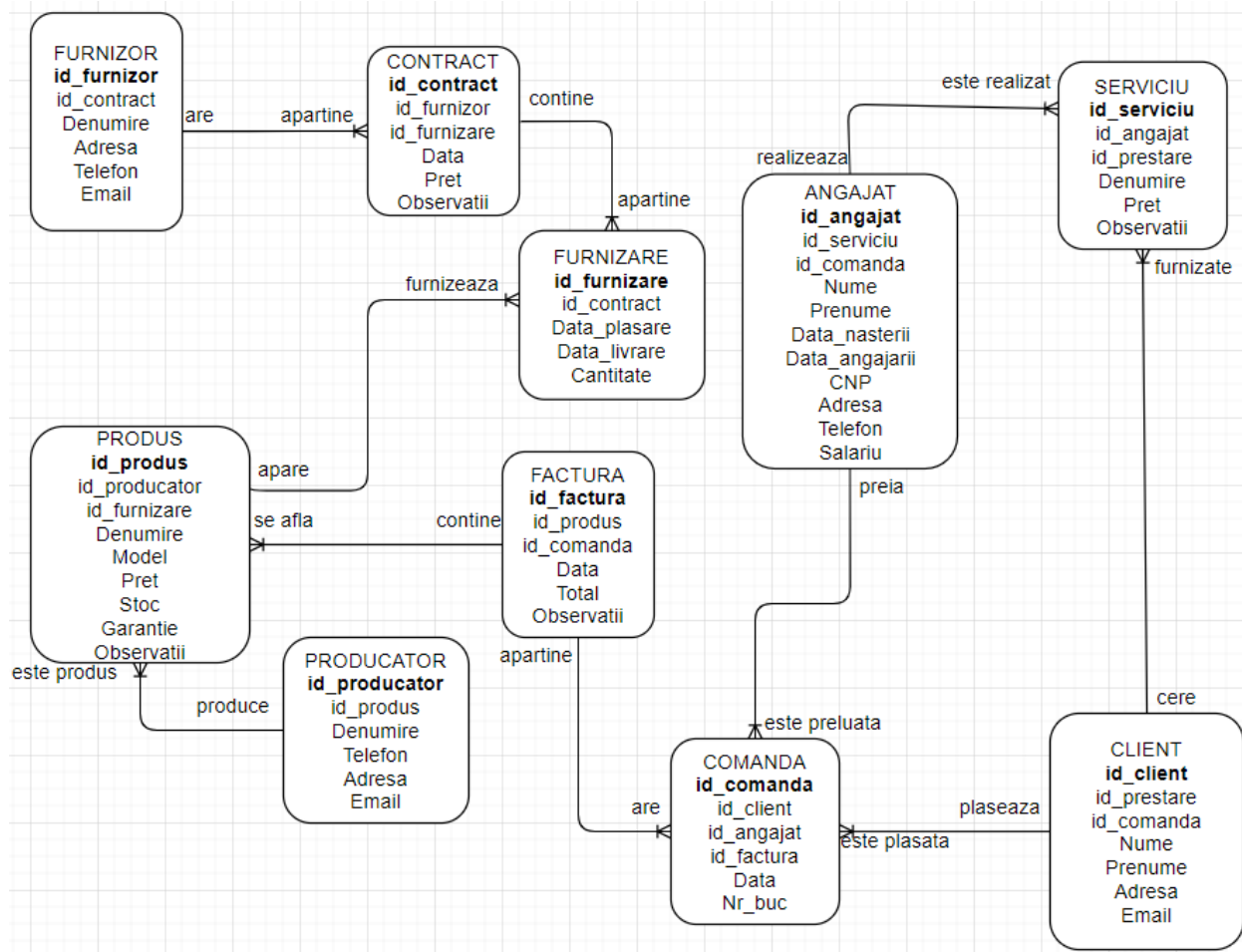
2021

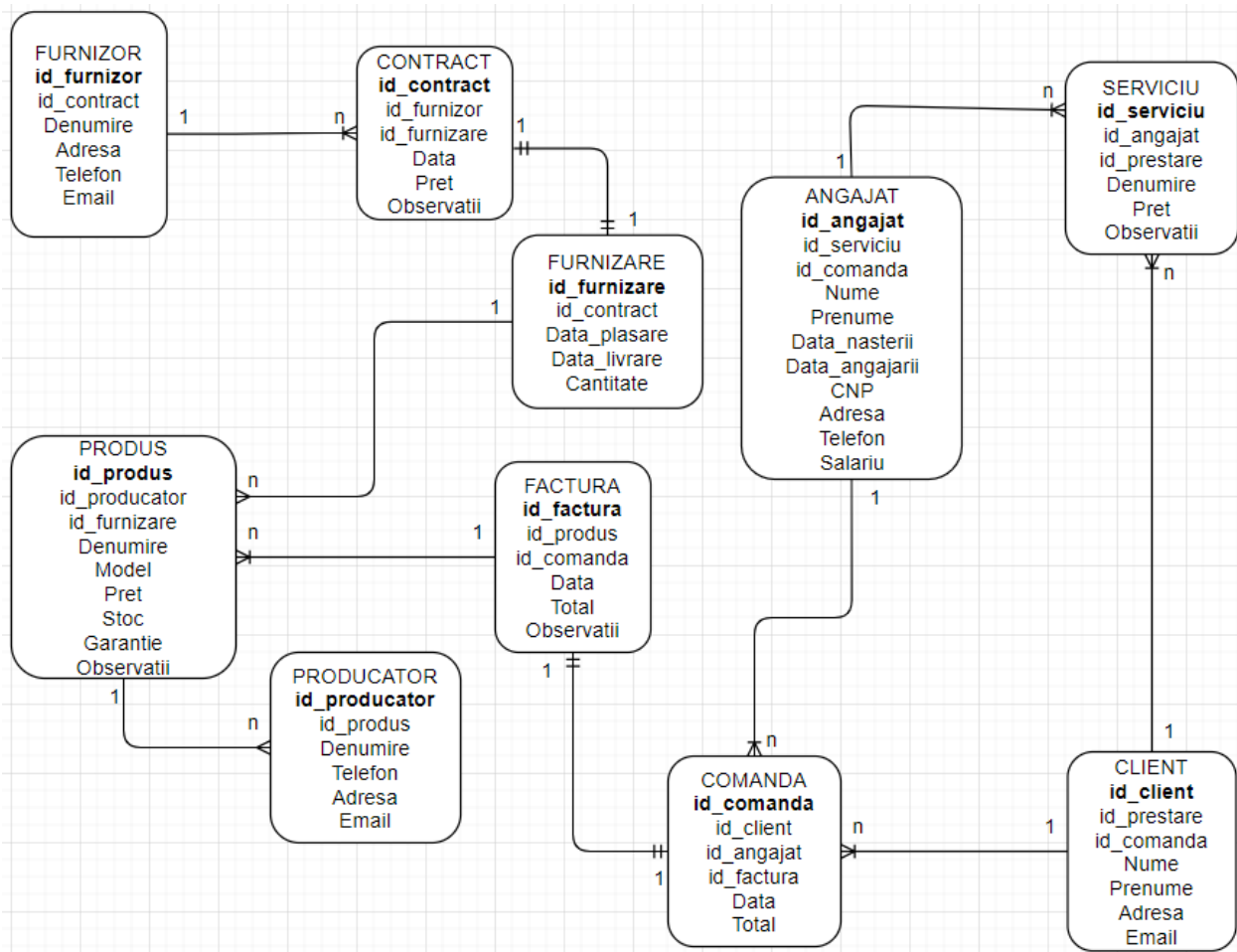
1. Descrierea temei

Pentru gestiunea unui magazin de calculatoare s-a implementat o baza de date in Oracle, care cuprinde tabele cu toate datele relevante din cadrul magazinului. Baza de date reprezinta o colectie de informatii referitoare la un anumit subiect, in vederea colectarii, memorarii, regasirii si gestiunii acestora. Baza de date cuprinde date despre furnizori, contracte, produse, producatori, comenzi, facturi, angajati, clienti si servicii.

Baza de date a magazinului de calculatoare pastreaza evidenta clara a angajatilor cu ajutorul datelor personale (nume, prenume, data nasterii, data angajarii, CNP, etc). De asemenea, firma isi propune sa pastreze si evidenta clientilor (nume, prenume, adresa, etc). Firma poate avea colaborari cu diferiti producatori, asa ca informatiile despre acestia trebuiesc inregistrate in baza de date. Totodata, datele producatorilor sunt relevante si in procesul de returnare a unui produs catre garantie. Deoarece firma este un magazin de calculatoare, aceasta este specializata in comercializarea produselor si serviciilor IT. Astfel, este necesar sa fie stocate date referitoare la produse (denumire, model, pret, stoc, garantie etc). Firma trebuie sa tina si o evidenta asupra veniturilor si a cheltuielilor, obtinuta cu ajutorul facturilor. Evidenta serviciilor se memoreaza prin prestarile de servicii pe care angajatii magazinului le ofera clientilor (se memoreaza date referitoare la serviciu in sine – denumire, total, etc).

2. Schema conceptuala pentru modelarea magazinului de calculatoare





3. Construirea bazei de date

Creare tabel „PRODUS”

create table "PRODUS"

(

"ID_PRODUS" number not null,

"ID_PRODUCATOR" number not null,

"ID_FURNIZARE" number not null,

"DENUMIRE" varchar2(255),

"MODEL" varchar2(255),

"PRET" number,

"STOC" number,

"GARANTIE" number,
 "OBSERVATII" varchar2(255),
 constraint "PRODUS_PK" primary key ("ID_PRODUS") enable
);

Column Name	Data Type	Nullable	Default	Primary Key
ID_PRODUS	NUMBER	No	-	1
ID_PRODUCATOR	NUMBER	No	-	-
ID_FURNIZARE	NUMBER	No	-	-
DENUMIRE	VARCHAR2(255)	Yes	-	-
MODEL	VARCHAR2(255)	Yes	-	-
PRET	NUMBER	Yes	-	-
STOC	NUMBER	Yes	-	-
GARANTIE	NUMBER	Yes	-	-
OBSERVATII	VARCHAR2(255)	Yes	-	-
				1 - 9

Creare tabel „ANGAJAT”

```
CREATE table "ANGAJAT" (
  "ID_ANGAJAT" NUMBER NOT NULL,
  "ID_SERVICIU" NUMBER NOT NULL,
  "ID_COMANDA" NUMBER NOT NULL,
  "NUME" VARCHAR2(255),
  "PRENUME" VARCHAR2(255),
  "DATA_NASTERII" DATE,
  "DATA_ANGAJARII" DATE,
  "CNP" VARCHAR2(255),
  "ADRESA" VARCHAR2(255),
  "TELEFON" NUMBER,
  "SALARIU" NUMBER,
```

constraint "ANGAJAT_PK" primary key ("ID_ANGAJAT")

);

Column Name	Data Type	Nullable	Default	Primary Key
ID_ANGAJAT	NUMBER	No	-	1
ID_SERVICIU	NUMBER	No	-	-
ID_COMANDA	NUMBER	No	-	-
NUME	VARCHAR2(255)	Yes	-	-
PRENUME	VARCHAR2(255)	Yes	-	-
DATA_NASTERII	DATE	Yes	-	-
DATA_ANGAJARII	DATE	Yes	-	-
CNP	VARCHAR2(255)	Yes	-	-
ADRESA	VARCHAR2(255)	Yes	-	-
TELEFON	NUMBER	Yes	-	-
SALARIU	NUMBER	Yes	-	-
				1 - 11

Creare tabel „CLIENT”

CREATE table "CLIENT" (

"ID_CLIENT" NUMBER NOT NULL,

"ID_COMANDA" NUMBER NOT NULL,

"NUME" VARCHAR2(255),

"PRENUME" VARCHAR2(255),

"ADRESA" VARCHAR2(255),

"EMAIL" VARCHAR2(255),

constraint "CLIENT_PK" primary key ("ID_CLIENT")

);

Column Name	Data Type	Nullable	Default	Primary Key
ID_CLIENT	NUMBER	No	-	1
ID_COMANDA	NUMBER	No	-	-
NUME	VARCHAR2(255)	Yes	-	-
PRENUME	VARCHAR2(255)	Yes	-	-
ADRESA	VARCHAR2(255)	Yes	-	-
EMAIL	VARCHAR2(255)	Yes	-	-
				1 - 6

Creare tabel „FURNIZOR”

```
CREATE table "FURNIZOR" (
  "ID_FURNIZOR" NUMBER NOT NULL,
  "ID_CONTRACT" NUMBER NOT NULL,
  "DENUMIRE" VARCHAR2(255),
  "ADRESA" VARCHAR2(255),
  "TELEFON" NUMBER,
  "EMAIL" VARCHAR2(255),
  constraint "FURNIZOR_PK" primary key ("ID_FURNIZOR")
);
```

Column Name	Data Type	Nullable	Default	Primary Key
ID_FURNIZOR	NUMBER	No	-	1
ID_CONTRACT	NUMBER	No	-	-
DENUMIRE	VARCHAR2(255)	Yes	-	-
ADRESA	VARCHAR2(255)	Yes	-	-
TELEFON	NUMBER	Yes	-	-
EMAIL	VARCHAR2(255)	Yes	-	-
				1 - 6

Creare tabel „CONTRACT”

```
CREATE table "CONTRACT" (  
    "ID_CONTRACT" NUMBER NOT NULL,  
    "ID_FURNIZOR" NUMBER NOT NULL,  
    "ID_FURNIZARE" NUMBER NOT NULL,  
    "DATA" DATE,  
    "PRET" NUMBER,  
    "OBSERVATII" VARCHAR2(255),  
    constraint "CONTRACT_PK" primary key ("ID_CONTRACT")  
);
```

Column Name	Data Type	Nullable	Default	Primary Key
ID_CONTRACT	NUMBER	No	-	1
ID_FURNIZOR	NUMBER	No	-	-
ID_FURNIZARE	NUMBER	No	-	-
DATA	DATE	Yes	-	-
PRET	NUMBER	Yes	-	-
OBSERVATII	VARCHAR2(255)	Yes	-	-
				1 - 6

Creare tabel „FURNIZARE”

```
CREATE table "FURNIZARE" (  
    "ID_FURNIZARE" NUMBER NOT NULL,  
    "ID_CONTRACT" NUMBER NOT NULL,  
    "DATA_PLASARE" DATE,  
    "DATA_LIVRARE" DATE,  
    "CANTITATE" NUMBER,  
    constraint "FURNIZARE_PK" primary key ("ID_FURNIZARE")  
);
```


Column Name	Data Type	Nullable	Default	Primary Key
ID_FURNIZARE	NUMBER	No	-	1
ID_CONTRACT	NUMBER	No	-	-
DATA_PLASARE	DATE	Yes	-	-
DATA_LIVRARE	DATE	Yes	-	-
CANTITATE	NUMBER	Yes	-	-
				1 - 5

Creare tabel „PRODUCATOR”

```
CREATE table "PRODUCATOR" (
    "ID_PRODUCATOR" NUMBER NOT NULL,
    "ID_PRODUS"    NUMBER NOT NULL,
    "DENUMIRE"     VARCHAR2(255),
    "TELEFON"      NUMBER,
    "ADRESA"       VARCHAR2(255),
    "EMAIL"        VARCHAR2(255),
    constraint "PRODUCATOR_PK" primary key ("ID_PRODUCATOR")
);
```

Column Name	Data Type	Nullable	Default	Primary Key
ID_PRODUCATOR	NUMBER	No	-	1
ID_PRODUS	NUMBER	No	-	-
DENUMIRE	VARCHAR2(255)	Yes	-	-
TELEFON	NUMBER	Yes	-	-
ADRESA	VARCHAR2(255)	Yes	-	-
EMAIL	VARCHAR2(255)	Yes	-	-
				1 - 6

Creare tabel „FACTURA”

```
CREATE table "FACTURA" (  
    "ID_FACTURA" NUMBER NOT NULL,  
    "ID_PRODUS" NUMBER NOT NULL,  
    "ID_COMANDA" NUMBER NOT NULL,  
    "DATA" DATE,  
    "TOTAL" NUMBER,  
    "OBSERVATII" VARCHAR2(255),  
    constraint "FACTURA_PK" primary key ("ID_FACTURA")  
);
```

Column Name	Data Type	Nullable	Default	Primary Key
ID_FACTURA	NUMBER	No	-	1
ID_PRODUS	NUMBER	No	-	-
ID_COMANDA	NUMBER	No	-	-
DATA	DATE	Yes	-	-
TOTAL	NUMBER	Yes	-	-
OBSERVATII	VARCHAR2(255)	Yes	-	-
				1 - 6

Creare tabel „COMANDA”

```
CREATE table "COMANDA" (  
    "ID_COMANDA" NUMBER NOT NULL,  
    "ID_CLIENT" NUMBER NOT NULL,  
    "ID_ANGAJAT" NUMBER NOT NULL,  
    "ID_FACTURA" NUMBER NOT NULL,  
    "DATA" DATE,  
    "NR_BUC" NUMBER,
```

constraint "COMANDA_PK" primary key ("ID_COMANDA")

);

Column Name	Data Type	Nullable	Default	Primary Key
ID_COMANDA	NUMBER	No	-	1
ID_CLIENT	NUMBER	No	-	-
ID_ANGAJAT	NUMBER	No	-	-
ID_FACTURA	NUMBER	No	-	-
DATA	DATE	Yes	-	-
NR_BUC	NUMBER	Yes	-	-
				1 - 6

Creare tabel „SERVICIU”

CREATE table "SERVICIU" (

"ID_SERVICIU" NUMBER NOT NULL,

"ID_ANGAJAT" NUMBER NOT NULL,

"ID_PRESTARE" NUMBER NOT NULL,

"DENUMIRE" VARCHAR2(255),

"PRET" NUMBER,

"OBSERVATII" VARCHAR2(255),

constraint "SERVICIU_PK" primary key ("ID_SERVICIU"));

Column Name	Data Type	Nullable	Default	Primary Key
ID_SERVICIU	NUMBER	No	-	1
ID_ANGAJAT	NUMBER	No	-	-
ID_PRESTARE	NUMBER	No	-	-
DENUMIRE	VARCHAR2(255)	Yes	-	-
PRET	NUMBER	Yes	-	-
OBSERVATII	VARCHAR2(255)	Yes	-	-
				1 - 6

Utilizarea comenzii ALTER TABLE in tabelul „PRODUS”

Column Name	Data Type	Nullable	Default	Primary Key
ID_PRODUS	NUMBER	No	-	1
ID_PRODUCATOR	NUMBER	No	-	-
ID_FURNIZARE	NUMBER	No	-	-
DENUMIRE	VARCHAR2(255)	Yes	-	-
MODEL	VARCHAR2(255)	Yes	-	-
PRET	NUMBER	Yes	-	-
STOC	NUMBER	Yes	-	-
GARANTIE	NUMBER	Yes	-	-
OBSERVATII	VARCHAR2(255)	Yes	-	-
				1 - 9

ALTER TABLE PRODUS

DROP COLUMN OBSERVATII;

Column Name	Data Type	Nullable	Default	Primary Key
ID_PRODUS	NUMBER	No	-	1
ID_PRODUCATOR	NUMBER	No	-	-
ID_FURNIZARE	NUMBER	No	-	-
DENUMIRE	VARCHAR2(255)	Yes	-	-
MODEL	VARCHAR2(255)	Yes	-	-
PRET	NUMBER	Yes	-	-
STOC	NUMBER	Yes	-	-
GARANTIE	NUMBER	Yes	-	-
				1 - 8

ALTER TABLE PRODUS

MODIFY MODEL NUMBER;

Column Name	Data Type	Nullable	Default	Primary Key
ID_PRODUS	NUMBER	No	-	1
ID_PRODUCATOR	NUMBER	No	-	-
ID_FURNIZARE	NUMBER	No	-	-
DENUMIRE	VARCHAR2(255)	Yes	-	-
MODEL	NUMBER	Yes	-	-
PRET	NUMBER	Yes	-	-
STOC	NUMBER	Yes	-	-
GARANTIE	NUMBER	Yes	-	-
				1 - 8

Utilizarea comenzii ALTER TABLE in tabelul „ANGAJAT”

Column Name	Data Type	Nullable	Default	Primary Key
ID_ANGAJAT	NUMBER	No	-	1
ID_SERVICIU	NUMBER	No	-	-
ID_COMANDA	NUMBER	No	-	-
NUME	VARCHAR2(255)	Yes	-	-
PRENUME	VARCHAR2(255)	Yes	-	-
DATA_NASTERII	DATE	Yes	-	-
DATA_ANGAJARII	DATE	Yes	-	-
CNP	VARCHAR2(255)	Yes	-	-
ADRESA	VARCHAR2(255)	Yes	-	-
TELEFON	NUMBER	Yes	-	-
SALARIU	NUMBER	Yes	-	-
				1 - 11

ALTER TABLE ANGAJAT

DROP COLUMN DATA_NASTERII

Column Name	Data Type	Nullable	Default	Primary Key
ID_ANGAJAT	NUMBER	No	-	1
ID_SERVICIU	NUMBER	No	-	-
ID_COMANDA	NUMBER	No	-	-
NUME	VARCHAR2(255)	Yes	-	-
PRENUME	VARCHAR2(255)	Yes	-	-
DATA_ANGAJARII	DATE	Yes	-	-
CNP	VARCHAR2(255)	Yes	-	-
ADRESA	VARCHAR2(255)	Yes	-	-
TELEFON	NUMBER	Yes	-	-
SALARIU	NUMBER	Yes	-	-
				1 - 10

ALTER TABLE ANGAJAT

DROP COLUMN DATA_ANGAJARII

Column Name	Data Type	Nullable	Default	Primary Key
ID_ANGAJAT	NUMBER	No	-	1
ID_SERVICIU	NUMBER	No	-	-
ID_COMANDA	NUMBER	No	-	-
NUME	VARCHAR2(255)	Yes	-	-
PRENUME	VARCHAR2(255)	Yes	-	-
CNP	VARCHAR2(255)	Yes	-	-
ADRESA	VARCHAR2(255)	Yes	-	-
TELEFON	NUMBER	Yes	-	-
SALARIU	NUMBER	Yes	-	-
				1 - 9

Utilizarea comenzii DROP TABLE in tabelul „PRODUS”

DROP TABLE PRODUS

PRODUS										Create ▼
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL
Add Column Modify Column Rename Column Drop Column Rename Copy Drop Truncate Create Lookup Table										
No data found.										

Obs! Dupa un Refresh asupra bazei de date, tabelul PRODUS dispare complet.

4. Exemple de operatii de actualizare a datelor

Utilizarea comenzii INSERT INTO

Comanda INSERT INTO este utilizata pentru a adauga date in tabelele deja existente. In exemplul de mai jos este prezentat tabelul CLIENT inainte si dupa utilizarea comenzii INSERT INTO cu ajutorul careia este adaugat un nou client in baza de date.







EDIT	ID_CLIENT	ID_COMANDA	NUME	PRENUME	ADRESA	EMAIL
	1	1	Neagu	Cristian	Ploiesti	dodeva6086@isecv.com
	2	2	Lautaru	Mihai	Ploiesti	xavaqikak-2051@yopmail.com
	3	3	Constantin	Costin	Baicoi	rjose.ortiz@bulegoblog.com
	4	4	Nutescu	Catalina	Ploiestiori	nuttummobo-6478@yopmail.com
	5	5	Luchian	Andrei	Bucuresti	atocexy-5362@yopmail.com
	6	6	Dima	Laurentiu	Ploiesti	perufforra-7557@yopmail.com
	7	7	Dinescu	Sebastian	Baicoi	xafynowell-0561@yopmail.com
row(s) 1 - 7 of 7						

INSERT INTO CLIENT

VALUES (8,8,'Alexandrescu','Catalin','Ploiesti','bikexenur-5801@yopmail.com');








EDIT	ID_CLIENT	ID_COMANDA	NUME	PRENUME	ADRESA	EMAIL
	1	1	Neagu	Cristian	Ploiesti	dodeva6086@isecv.com
	2	2	Lautaru	Mihai	Ploiesti	xavaqikak-2051@yopmail.com
	3	3	Constantin	Costin	Baicoi	rjose.ortiz@bulegoblog.com
	4	4	Nutescu	Catalina	Ploiestiori	nuttummobo-6478@yopmail.com
	5	5	Luchian	Andrei	Bucuresti	atocexy-5362@yopmail.com
	6	6	Dima	Laurentiu	Ploiesti	perufforra-7557@yopmail.com
	7	7	Dinescu	Sebastian	Baicoi	xafynowell-0561@yopmail.com
	8	8	Alexandrescu	Catalin	Ploiesti	bikexenur-5801@yopmail.com
row(s) 1 - 8 of 8						

In exemplul de mai jos este prezentat tabelul FACTURA inainte si dupa utilizarea comenzii INSERT INTO cu ajutorul careia este adaugata o noua factura in baza de date existenta.







EDIT	ID_FACTURA	ID_PRODUS	ID_COMANDA	DATA	TOTAL	OBSERVATII
	1	1	1	05/22/2021	300	-
	2	2	2	05/22/2021	1100	-
	3	3	3	05/21/2021	500	-
	4	4	4	05/20/2021	700	-
	5	5	5	05/21/2021	1900	-
	6	6	6	05/22/2021	1100	-
row(s) 1 - 6 of 6						

INSERT INTO FACTURA

VALUES (7,7,7,'05/19/2021',2100,'-');








EDIT	ID_FACTURA	ID_PRODUS	ID_COMANDA	DATA	TOTAL	OBSERVATII
	7	7	7	05/19/2021	2100	-
	1	1	1	05/22/2021	300	-
	2	2	2	05/22/2021	1100	-
	3	3	3	05/21/2021	500	-
	4	4	4	05/20/2021	700	-
	5	5	5	05/21/2021	1900	-
	6	6	6	05/22/2021	1100	-
row(s) 1 - 7 of 7						

In exemplul de mai jos este prezentat tabelul COMANDA inainte si dupa utilizarea comenzii INSERT INTO cu ajutorul careia este adaugata o noua comanda in baza de date existenta.






EDIT	ID_COMANDA	ID_CLIENT	ID_ANGAJAT	ID_FACTURA	DATA	NR_BUC
	1	1	1	1	05/22/2021	1
	2	2	2	2	05/22/2021	2
	3	3	3	3	04/20/2021	1
	4	4	4	4	05/19/2021	1
	5	5	5	5	05/19/2021	2
	6	6	6	6	05/21/2021	1
row(s) 1 - 6 of 6						

INSERT INTO COMANDA

VALUES (7,7,7,7,'05/22/2021',1)







EDIT	ID_COMANDA	ID_CLIENT	ID_ANGAJAT	ID_FACTURA	DATA	NR_BUC
	1	1	1	1	05/22/2021	1
	2	2	2	2	05/22/2021	2
	3	3	3	3	04/20/2021	1
	4	4	4	4	05/19/2021	1
	5	5	5	5	05/19/2021	2
	6	6	6	6	05/21/2021	1
	7	7	7	7	05/22/2021	1
row(s) 1 - 7 of 7						

In exemplul de mai jos este prezentat tabelul FURNIZOR inainte si dupa utilizarea comenzii INSERT INTO cu ajutorul careia este adaugat un nou furnizor de piese/calculatoare in baza de date existenta.

EDIT	ID_FURNIZOR	ID_CONTRACT	DENUMIRE	ADRESA	TELEFON	EMAIL
	1	1	GeFors Computers	Ploiesti	7115554292	usocutomy-8583@yopmail.com
	2	2	Bocris	Ploiesti	7115555545	aloppurre-6655@yopmail.com
	3	3	Mival Group	Ploiesti	7115554543	enonninaxi-3752@yopmail.com
	4	4	PCZone	Bucuresti	7555549993	uppozemis-5060@yopmail.com
	5	5	ABD Computer	Bucuresti	7115555416	hannibybo-8887@yopmail.com
row(s) 1 - 5 of 5						

INSERT INTO FURNIZOR

VALUES (6,6,'DensiTron','Ploiesti',712395435,'ronnymmevovo-7659@yopmail.com');

EDIT	ID_FURNIZOR	ID_CONTRACT	DENUMIRE	ADRESA	TELEFON	EMAIL
	6	6	DensiTron	Ploiesti	712395435	ronnymmevovo-7659@yopmail.com
	1	1	GeFors Computers	Ploiesti	7115554292	usocutomy-8583@yopmail.com
	2	2	Bocris	Ploiesti	7115555545	aloppurre-6655@yopmail.com
	3	3	Mival Group	Ploiesti	7115554543	enonninaxi-3752@yopmail.com
	4	4	PCZone	Bucuresti	7555549993	uppozemis-5060@yopmail.com
	5	5	ABD Computer	Bucuresti	7115555416	hannibybo-8887@yopmail.com
row(s) 1 - 6 of 6						

Utilizarea comenzii UPDATE

In exemplul urmator, comanda UPDATE este utilizata pentru a modifica adresa unui angajat in cazul relocarii acestuia in alt oras, pentru a tine evidenta adreselor angajatilor in baza de date. Sunt prezentate stadiile tabelului inainte si dupa utilizarea comenzii UPDATE.

EDIT	ID_ANGAJAT	ID_SERVICIU	ID_COMANDA	NUME	PRENUME	CNP	ADRESA	TELEFON	SALARIU	DATA_ANGAJARII
	1	1	1	Cantemir	Adrian	1982890821308	Bucuresti	721001011	2500	10/17/2018
	2	1	2	Dumitrescu	Bogdan	1467459270513	Ploiesti	721017829	2500	10/10/2019
	3	2	3	Bibescu	Ciprian	1804490312364	Bucuresti	721099085	3000	05/03/2020
	4	3	4	Baciu	Livia	6074547896537	Bucuresti	721001099	2100	05/04/2020
	5	4	5	Oprisan	Alexandru	1242297234155	Balcoi	729199101	3500	09/04/2019
	6	4	6	Antonescu	Mihaela	2790468310916	Ploiestiori	732190191	2700	12/19/2020
	7	5	7	Ionescu	Andrei	1399052661959	Mizil	741001213	2800	09/29/2020
	8	6	8	Daban	Iulian	1824378030144	Ploiesti	768988101	2800	07/21/2019
	9	7	9	Diaconu	Matei	1544273038014	Bucuresti	761213141	2900	01/31/2018
	10	7	10	Naftan	Laura	6345466949752	Balcoi	721099980	2900	05/17/2020
row(s) 1 - 10 of 10										

UPDATE ANGAJAT

SET ADRESA='Ploiesti'

WHERE ID_ANGAJAT=1

row(s) 1 - 10 of 10

In exemplul urmatoar, comanda UPDATE este utilizata pentru a modifica pretul unui contract in cazul in care acesta a fost introdus eronat in baza de date. Sunt prezentate stadiile tabelului inainte si dupa utilizarea comenzii UPDATE.

row(s) 1 - 5 of 5

UPDATE CONTRACT

```
SET PRET=3800
```

WHERE ID CONTRACT=2

row(s) 1 - 5 of 5

In exemplul urmatoar, comanda UPDATE este utilizata pentru a modifica modelul unui produs in cazul in care acesta a fost introdus eronat sau daca contine denumiri dupa care acesta nu poate fi gasit prin interogarea bazei de date. Sunt prezentate stadiile tabelului inainte si dupa utilizarea comenzii UPDATE.

[illegible]

UPDATE PRODUS

```
SET MODEL='HP'
```

WHERE ID_PRODUS=13

[illegible]




Utilizarea comenzii DELETE

In exemplul urmatoare se utilizeaza comanda DELETE pentru a sterge datele unui angajat din tabelul ANGAJAT in cazul in care acesta este concediat sau isi da demisia din cadrul firmei. Sunt prezentate stadiile tabelului inainte si dupa utilizarea comenzii DELETE.

EDIT	ID_ANGAJAT	ID_SERVICIU	ID_COMANDA	NUME	PRENUME	CNP	ADRESA	TELEFON	SALARIU	DATA_ANGAJARII
	1	1	1	Cantemir	Adrian	1982890821308	Ploiesti	721001011	2500	10/17/2018
	2	1	2	Dumitrescu	Bogdan	1467459270513	Ploiesti	721017829	2500	10/10/2019
	3	2	3	Bibescu	Ciprian	1804490312364	Bucuresti	721099085	3000	05/03/2020
	4	3	4	Baciu	Livia	6074547896537	Bucuresti	721001099	2100	05/04/2020
	5	4	5	Oprisan	Alexandru	1242297234155	Balcoi	729199101	3500	09/04/2019
	6	4	6	Antonescu	Mihaela	2790468310916	Ploiestiori	732190191	2700	12/19/2020
	7	5	7	Ionescu	Andrei	1399052661959	Mizil	741001213	2800	09/29/2020
	8	6	8	Daban	Iulian	1824378030144	Ploiesti	768988101	2800	07/21/2019
	9	7	9	Diaconu	Matei	1544273038014	Bucuresti	761213141	2900	01/31/2018
	10	7	10	Naftan	Laura	6345466949752	Balcoi	721099980	2900	05/17/2020
row(s) 1 - 10 of 10										

DELETE FROM ANGAJAT

WHERE ID_ANGAJAT=9

EDIT	ID_ANGAJAT	ID_SERVICIU	ID_COMANDA	NUME	PRENUME	CNP	ADRESA	TELEFON	SALARIU	DATA_ANGAJARII
	1	1	1	Cantemir	Adrian	1982890821308	Ploiesti	721001011	2500	10/17/2018
	2	1	2	Dumitrescu	Bogdan	1467459270513	Ploiesti	721017829	2500	10/10/2019
	3	2	3	Bibescu	Ciprian	1804490312364	Bucuresti	721099085	3000	05/03/2020
	4	3	4	Baciu	Livia	6074547896537	Bucuresti	721001099	2100	05/04/2020
	5	4	5	Oprisan	Alexandru	1242297234155	Balcoi	729199101	3500	09/04/2019
	6	4	6	Antonescu	Mihaela	2790468310916	Ploiestiori	732190191	2700	12/19/2020
	7	5	7	Ionescu	Andrei	1399052661959	Mizil	741001213	2800	09/29/2020
	8	6	8	Daban	Iulian	1824378030144	Ploiesti	768988101	2800	07/21/2019
	10	7	10	Naftan	Laura	6345466949752	Balcoi	721099980	2900	05/17/2020
row(s) 1 - 9 of 9										

In exemplul urmatoare se utilizeaza comanda DELETE pentru a sterge datele unui client din baza de date atunci cand acesta nu mai face parte din clientela magazinului. Sunt prezentate stadiile tabelului inainte si dupa utilizarea comenzii DELETE.

EDIT	ID_CLIENT	ID_COMANDA	NUME	PRENUME	ADRESA	EMAIL
	1	1	Neagu	Cristian	Ploiesti	dodeva6086@isecv.com
	2	2	Lautaru	Mihai	Ploiesti	xavaqikak-2051@yopmail.com
	3	3	Constantin	Costin	Baicoi	rjose.ortiz@bulegoblog.com
	4	4	Nutescu	Catalina	Ploiestiori	nuttummobo-6478@yopmail.com
	5	5	Luchian	Andrei	Bucuresti	atocexy-5362@yopmail.com
	6	6	Dima	Laurentiu	Ploiesti	perufforra-7557@yopmail.com
	7	7	Dinescu	Sebastian	Baicoi	xafynowell-0561@yopmail.com
	8	8	Alexandrescu	Catalin	Ploiesti	bikexenur-5801@yopmail.com
row(s) 1 - 8 of 8						

DELETE FROM CLIENT

WHERE ID_CLIENT=8

EDIT	ID_CLIENT	ID_COMANDA	NUME	PRENUME	ADRESA	EMAIL
	1	1	Neagu	Cristian	Ploiesti	dodeva6086@isecv.com
	2	2	Lautaru	Mihai	Ploiesti	xavaqikak-2051@yopmail.com
	3	3	Constantin	Costin	Baicoi	rjose.ortiz@bulegoblog.com
	4	4	Nutescu	Catalina	Ploiestiori	nuttummobo-6478@yopmail.com
	5	5	Luchian	Andrei	Bucuresti	atocexy-5362@yopmail.com
	6	6	Dima	Laurentiu	Ploiesti	perufforra-7557@yopmail.com
	7	7	Dinescu	Sebastian	Baicoi	xafynowell-0561@yopmail.com
row(s) 1 - 7 of 7						

In exemplul urmatoar se utilizeaza comanda DELETE pentru a sterge datele unui produs din cadrul bazei de date atunci cand acesta nu mai face parte din oferta magazinului. Sunt prezentate stadiile tabelului inainte si dupa utilizarea comenzii DELETE.

EDIT	ID_PRODUS	ID_PRODUCATOR	ID_FURNIZARE	DENUMIRE	MODEL	PRET	STOC	GARANTIE	OBSERVATII
	1	1	1	SSD	Kingston A400 240GB	150	20	36	-
	2	1	2	SSD	Kingston A400 480GB	250	14	36	-
	3	2	3	SSD	ADATA SU630 240GB	150	15	36	-
	4	3	4	HDD	WD Blue 1TB	180	10	24	-
	5	3	5	HDD	WD Purple 2TB	305	6	36	-
	6	4	6	HDD	Seagate BarraCuda 2TB	240	17	24	-
	7	4	7	HDD	Seagate IronWolf NAS 2TB	400	3	36	-
	8	5	8	Monitor	LED IPS Dell 24"	650	21	36	-
	9	6	9	Monitor	LED IPS Lenovo 27"	1000	10	36	-
	10	7	10	Monitor	LED TN Samsung 21.5"	480	16	24	-
	11	8	11	Desktop PC	Seriousx	2200	4	24	-
	12	8	12	Desktop PC	Seriousx	3600	7	24	-
	13	9	13	Desktop PC	HP	6500	3	2	-
row(s) 1 - 13 of 13									

DELETE FROM PRODUS

WHERE ID_PRODUS=12

EDIT	ID_PRODUS	ID_PRODUCATOR	ID_FURNIZARE	DENUMIRE	MODEL	PRET	STOC	GARANTIE	OBSERVATII
	1	1	1	SSD	Kingston A400 240GB	150	20	36	-
	2	1	2	SSD	Kingston A400 480GB	250	14	36	-
	3	2	3	SSD	ADATA SU630 240GB	150	15	36	-
	4	3	4	HDD	WD Blue 1TB	180	10	24	-
	5	3	5	HDD	WD Purple 2TB	305	6	36	-
	6	4	6	HDD	Seagate BarraCuda 2TB	240	17	24	-
	7	4	7	HDD	Seagate IronWolf NAS 2TB	400	3	36	-
	8	5	8	Monitor	LED IPS Dell 24"	650	21	36	-
	9	6	9	Monitor	LED IPS Lenovo 27"	1000	10	36	-
	10	7	10	Monitor	LED TN Samsung 21.5"	480	16	24	-
	11	8	11	Desktop PC	Seriousx	2200	4	24	-
	13	9	13	Desktop PC	HP	6500	3	2	-
row(s) 1 - 12 of 12									

5. Exemple de interogari

In exemplul de mai jos se utilizeaza comanda SELECT DISTINCT in tabelul PRODUS pentru a vedea care este varietatea de produse din cadrul magazinului de calculatoare.

SELECT DISTINCT DENUMIRE FROM PRODUS

DENUMIRE
SSD
HDD
SSD
Monitor
Desktop PC

In exemplul urmator este prezentata utilizarea comenzii LEFT JOIN in tabelul FURNIZOR pentru a vedea corespondenta intre id-urile contractelor existente si id-urile furnizorilor din baza de date. Se poate observa faptul ca unul dintre furnizori nu are un contract creat sau introdus in baza de date.

```
SELECT FURNIZOR.ID_FURNIZOR, CONTRACT.ID_CONTRACT FROM FURNIZOR
LEFT JOIN CONTRACT ON FURNIZOR.ID_FURNIZOR = CONTRACT.ID_FURNIZOR
```

ID_FURNIZOR	ID_CONTRACT
1	1
2	2
3	3
4	4
5	5
6	-

In exemplul urmator este prezentata utilizarea comenzii INNER JOIN in tabelul ANGAJAT pentru a vedea corespondenta intre id-urile serviciilor executate de angajati si id-urile serviciilor existente in baza de date. Astfel, firma se asigura ca fiecare angajat are un serviciu pe care il executa si nu sunt foarte multi angajati care executa acealasi serviciu sau servicii care nu sunt executate de niciun angajat.

```
SELECT ANGAJAT.ID_SERVICIU, SERVICIU.ID_SERVICIU FROM ANGAJAT
INNER JOIN SERVICIU ON ANGAJAT.ID_SERVICIU = SERVICIU.ID_SERVICIU
```

ID_SERVICIU	ID_SERVICIU
1	1
1	1
2	2
3	3
4	4
4	4
5	5
6	6
7	7

In exemplul urmator se utilizeaza comanda SELECT AVG in tabelul FACTURA pentru a face o medie a totalului facturilor. Astfel, pentru contabilitatea firmei, este mai simplu de evaluat care este media facturilor tiparite si pentru a deduce pierderi si profit.

SELECT AVG(TOTAL) AS MEDIE_FACTURI FROM FACTURA

MEDIE_FACTURI
1100

In exemplul urmator se utilizeaza functia SELECT in tabelul ANGAJAT pentru a vizualiza adresele angajatilor. Astfel, firma ia la cunostinta provenienta fiecarui angajat si poate oferi servicii precum asigurarea transportului sau decontarea abonamentelor de transport.

SELECT ADRESA AS INFO_CONTACT FROM ANGAJAT

INFO_CONTACT
Ploiesti
Ploiesti
Bucuresti
Bucuresti
Baicoi
Ploiestiori
Mizil
Ploiesti
Baicoi

In exemplul urmator este utilizata comanda SELECT – WHERE in tabelul PRODUS pentru a furniza informatii cu privire la stocul produselor care se afla in cantitate mai mica de 5. Astfel, firma poate comanda produsele respective pentru a nu ramane cu stocul epuizat.

SELECT DENUMIRE,MODEL,STOC FROM PRODUS WHERE STOC<5

DENUMIRE	MODEL	STOC
HDD	Seagate IronWolf NAS 2TB	3
Desktop PC	Serious	4
Desktop PC	HP	3

In exemplul de mai jos este utilizata comanda ORDER BY pentru a vizualiza in cadrul tabelii FURNIZOR care este cel mai recent furnizor de piese obtinut de catre firma.

SELECT ID_CONTRACT, DENUMIRE FROM FURNIZOR

ORDER BY ID_CONTRACT DESC

ID_CONTRACT	DENUMIRE
6	DensiTron
5	ABD Computer
4	PCZone
3	Mival Group
2	Bocris
1	GeFors Computers

In exemplul urmator este utilizata comanda SELECT *, ORDER BY in tabela CLIENT pentru a vizualiza cati dintre clientii magazinului de calculatoare sunt din orasul Ploiesti. Dupa cum se poate vedea, un numar destul de mic sunt clienti locali ai magazinului, restul fiind clienti externi.

```
SELECT * FROM CLIENT WHERE ADRESA='Ploiesti' ORDER BY NUME ASC
```

ID_CLIENT	ID_COMANDA	NUME	PRENUME	ADRESA	EMAIL
6	6	Dima	Laurentiu	Ploiesti	perufforra-7557@yopmail.com
2	2	Lautaru	Mihai	Ploiesti	xavaqikak-2051@yopmail.com
1	1	Neagu	Cristian	Ploiesti	dodeva6086@isecv.com

In exemplul urmator este utilizata comanda LIKE in tabela ANGAJAT pentru a vizualiza angajatii firmei care au un numar de telefon care incepe cu 72, adica cel mai probabil in reseaua Vodafone. Astfel, firma poate face un sondaj si decide asupra unor beneficii pentru angajatii care se afla majoritar intr-o anumita retea de telefonie.

```
SELECT NUME, PRENUME, TELEFON FROM ANGAJAT
WHERE TELEFON LIKE '72%'
```

NUME	PRENUME	TELEFON
Cantemir	Adrian	721001011
Dumitrescu	Bogdan	721017829
Bibescu	Ciprian	721099085
Baciu	Livia	721001099
Oprisan	Alexandru	729199101
Naftan	Laura	721099980

In exemplul urmator este utilizata comanda BETWEEN in tabela ANGAJAT pentru a vizualiza angajatii care au salarii pana la 2500 de lei, acestia fiind cei mai predispusi unei mariri in cazul in care salariile lor sunt prea mici raportat la salariul de piata.

SELECT NUME, PRENUME, SALARIU FROM ANGAJAT

WHERE SALARIU BETWEEN 0 AND 2500

NUME	PRENUME	SALARIU
Cantemir	Adrian	2500
Dumitrescu	Bogdan	2500
Baciu	Livia	2100

In exemplul urmator se utilizeaza comanda SELECT – AS in tabela PRODUS pentru a avea o privire de ansamblu asupra produselor si modelului produselor, denumite in continuare specificatii.

SELECT DENUMIRE, MODEL AS SPECIFICATII FROM PRODUS

DENUMIRE	SPECIFICATII
SSD	Kingston A400 240GB
SSD	Kingston A400 480GB
SSD	ADATA SU630 240GB
HDD	WD Blue 1TB
HDD	WD Purple 2TB
HDD	Seagate BarraCuda 2TB
HDD	Seagate IronWolf NAS 2TB
Monitor	LED IPS Dell 24"
Monitor	LED IPS Lenovo 27"
Monitor	LED TN Samsung 21.5"
More than 10 rows available. Increase rows selector to view more rows.	

In exemplul urmator se utilizeaza functia SUM in tabela FURNIZARE pentru a calcula cantitatea totala de produse care se vor livra in ziua de 05/22/2021. Astfel, se poate face loc in depozit pentru cantitatea asteptata la livrare sau se pot calcula stocurile produselor care urmeaza a intra in depozitul magazinului.

```
SELECT SUM(CANTITATE) AS CANTITATE_TOTALA FROM FURNIZARE
WHERE DATA_LIVRARE='05/22/2021'
```

CANTITATE_TOTALA
62

In exemplul urmator se utilizeaza comanda SELECT – IN in tabela CLIENT pentru a vizualiza clientii care apartin orasului Ploiesti dar si cei care apartin municipiului Bucuresti. Astfel, firma are o vedere de ansamblu asupra interesului intern vs extern al clientilor magazinului.

```
SELECT NUME, PRENUME, ADRESA FROM CLIENT
WHERE ADRESA IN ('Ploiesti', 'Bucuresti')
```

NUME	PRENUME	ADRESA
Neagu	Cristian	Ploiesti
Lautaru	Mihai	Ploiesti
Luchian	Andrei	Bucuresti
Dima	Laurentiu	Ploiesti

In exemplul urmator este utilizata comanda INNER JOIN pentru a vizualiza id-urile comenzilor din tabelele FACTURA si COMANDA in vederea observarii corespondentei intre cele doua. Astfel, firma este interesata daca fiecarei comenzi ii corespunde o factura si vice-versa.

```
SELECT COMANDA.ID_COMANDA, FACTURA.ID_COMANDA FROM COMANDA
INNER JOIN FACTURA ON COMANDA.ID_COMANDA = FACTURA.ID_COMANDA
```

ID_COMANDA	ID_COMANDA
7	7
1	1
2	2
3	3
4	4
5	5
6	6

In exemplul urmator se utilizeaza comanda SELECT *, BETWEEN in tabelul CONTRACT pentru a vizualiza contractele incheiate de catre firma in ultimele 2 zile. Astfel este oferita o perspectiva asupra gradului de recunoastere in piata.

```
SELECT * FROM CONTRACT WHERE DATA BETWEEN '05/21/2021' AND '05/22/2021'
```

ID_CONTRACT	ID_FURNIZOR	ID_FURNIZARE	DATA	PRET	OBSERVATII
1	1	1	05/22/2021	2500	-
2	2	2	05/21/2021	3800	-
4	4	4	05/22/2021	850	-
5	5	5	05/22/2021	3000	-