



TechFlow

-PROIECT BAZE DE DATE -

CUPRINS :

Cerința 1.	3
Cerința 2.	5
Cerința 3.	6
Cerința 4.	7
Cerința 5.	8
Cerința 6.	10
Cerința 7.	11
Cerința 8.	12
Cerința 9.	13
Cerința 10.	15
Cerința 11.	16
Cerința 12.	19
Cerința 13.	22
Cerința 14.	23
Cerința 15.	24
Cerința 16.	26
Cerinta 17.	28
Cerinta 18.	30
Cerinta 19.	32

Cerința 1.

Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.

Numele Proiectului : TechFlow

Tema : Bază de date pentru gestionarea vânzărilor de produse IT

Descrierea modelului real:

TechFlow este o afacere specializată în vânzarea de produse IT, cum ar fi componente hardware, accesorii și echipamente IT, prin intermediul magazinelor fizice și a site-ului lor de comerț electronic.

Aceasta gestionează informații detaliate despre **produse, categorii, producători, clienți, comenzi, detalii ale comenzilor, magazine fizice, angajați, recenzii, plăți, transport și wishlist**.

TechFlow oferă o gamă largă de produse IT, iar fiecare **produs** are un nume unic, o descriere detaliată, un preț și o cantitate de stoc disponibilă. Produsele sunt împărțite în **categorii**, iar fiecare categorie are un nume unic. De asemenea, fiecare produs este asociat cu un **producător**, care are un nume unic și o țară de origine.

Clienții TechFlow pot plasa **comenzi** pentru produsele disponibile prin intermediul site-ului de comerț electronic. **Comenzile** conțin informații despre data comenzii și statusul comenzii, care indică starea acesteia, cum ar fi "nouă", "confirmată" sau "anulată". **Detaliile despre produsele comandate**, cum ar fi cantitatea și prețul unitar, sunt înregistrate în sistem. De asemenea în momentul plasării comenzii, clienții aleg modalitatea de **transport**.

Pentru a gestiona stocul, TechFlow înregistrează informații despre **magazinele fizice** pe care le deține, inclusiv numele magazinului și adresa. De asemenea, **angajații** care lucrează în magazinele fizice sunt înregistrați în sistem, având funcții precum vânzător, casier sau manager.

Clienții TechFlow au opțiunea de a lăsa **recenzii** și feedback despre produsele achiziționate. Aceste recenzii conțin informații despre autor, produsul recenzat și nota acordată.

Plățile pentru comenzile plasate sunt înregistrate în sistem, împreună cu informații despre metoda de plată utilizată.

De asemenea, TechFlow oferă funcționalitatea de **wishlist**, unde clienții pot adăuga produsele preferate pentru a le urmări și a le achiziționa ulterior.

Utilitate:

Prin intermediul bazei de date TechFlow, se realizează gestiunea detaliată a produselor, categoriilor de produse, comenzilor , plăților și recenziilor. Aceasta facilitează procesul de vânzare, stocare, gestionare a comenzilor și interacțiune cu clienții. De asemenea, baza de date permite generarea de rapoarte și analize pentru a obține o înțelegere mai profundă a activităților de vânzare și a preferințelor clienților, contribuind astfel la **luarea deciziilor strategice** și la **îmbunătățirea experienței clienților** .

Reguli de functionare:

- **Adăugarea, actualizarea și ștergerea** produselor, categoriilor, producătorilor, clienților, comenzilor, recenziilor, plăților și wishlist-urilor.
- **Căutarea și afișarea** informațiilor despre produse, categorii, producători, clienți, comenzile și recenziile asociate acestora.
- **Gestionarea** stocurilor de produse și actualizarea nivelurilor de stoc în funcție de comenzile primite și plățile efectuate.
- **Înregistrarea și gestionarea** comenzilor efectuate de clienți, inclusiv actualizarea stării comenzilor.
- **Adăugarea și gestionarea** recenziilor și notelor acordate de clienți pentru produsele achiziționate.
- **Administrarea** datelor despre clienți, inclusiv înregistrarea și actualizarea detaliilor personale, gestionarea wishlist-urilor și a istoricului de comenzi.
- **Gestionarea** producătorilor și categoriilor de produse, inclusiv adăugarea și actualizarea informațiilor despre aceștia.
- **Administrarea** datelor despre angajați, inclusiv înregistrarea și actualizarea informațiilor despre nume, prenume, funcție și asocierea acestora cu magazinele TechFlow.

Cerința 2.

Prezentarea constrângerilor impuse asupra modelului.

- Fiecare **produs** trebuie să aibă un nume unic, o descriere detaliată, un preț și o cantitate de stoc disponibilă.
- **Produsele** trebuie să fie împărțite în **categorii**, iar fiecare categorie trebuie să aibă un nume unic.
- Fiecare **produs** trebuie să fie asociat cu un **producător**, care are un nume unic și o țară de origine.
- **Clienții** trebuie să poată plasa **comenzi** pentru produsele disponibile prin intermediul site-ului de comerț electronic.
- **Comenzile** trebuie să conțină informații despre data comenzii și statusul comenzii, care indică starea acesteia, cum ar fi "nouă", "confirmată" sau "anulată".
- **Clienții** trebuie să poată alege modalitatea de transport în momentul plasării comenzii.
- TechFlow trebuie să înregistreze informații despre **magazinele fizice** pe care le deține, inclusiv numele magazinului, adresa.
- **Angajații** care lucrează în magazinele fizice trebuie să fie înregistrați în sistem, având funcții precum vânzător, casier sau manager.
- Clienții TechFlow trebuie să aibă opțiunea de a lăsa **recenzii** și feedback despre produsele achiziționate, care conțin informații despre autor, produsul recenzat și nota acordată.
- **Plățile** pentru comenzile plasate trebuie să fie înregistrate în sistem, împreună cu informații despre metoda de plată utilizată și tipul plății.
- TechFlow trebuie să ofere funcționalitatea de **wishlist**, unde clienții pot adăuga produsele preferate pentru a le urmări și a le achiziționa ulterior.

Cerința 3.

Descrierea entităților, incluzând precizarea cheii primare.

1. **"Produse"**: reprezintă un produs IT vândut de TechFlow
Cheia primară: ID_produc
2. **"Producatori"**: reprezintă compania care a fabricat produsul IT vândut de TechFlow
Cheia primară: ID_producator
3. **"Categorii"**: reprezintă o categorie în care sunt împărțite produsele TechFlow
Cheia primară: ID_categorie
4. **"Clienți"**: reprezintă o persoană care a cumpărat un produs sau a plasat o comandă de produse de pe site-ul TechFlow
Cheia primară: ID_client
5. **"Comenzi"**: reprezintă o comandă plasată de un client
Cheia primară: ID_comanda
6. **"Magazine"**: reprezintă un magazin fizic deținut de TechFlow
Cheia primară: ID_magazin
7. **"Angajați"**: reprezintă o persoană care lucrează într-un magazin fizic TechFlow
Cheia primară: ID_angajat
8. **"Recenzii"**: reprezintă feedback-ul lăsat de un client cu privire la un produs achiziționat
Cheia primară: ID_recenzie
9. **"Plati"**: reprezintă o plată înregistrată pentru o comandă plasată de un client
Cheia primară: ID_plata
10. **"Transport"**: reprezintă modalitatea de transport în momentul plasării comenzii
Cheia primară: ID_transport
11. **"Wishlist"**: reprezintă o listă de produse preferate de un client
Cheia primară: ID_wishlist

Cerința 4.

Descrierea relațiilor, incluzând precizarea cardinalității acestora.

Relatii:

1. **"Produse" - "Producatori":** Many-to-One
(o companie poate fabrica mai multe produse, dar fiecare produs este asociat cu un singur producător)
2. **"Produse" - "Categorii":** Many-to-One
(un produs poate fi încadrat într-o singură categorie, dar o categorie poate avea mai multe produse)
3. **"Clienți" - "Comenzi":** One-to-Many
(un client poate plasa mai multe comenzi, dar o comandă este plasată de un singur client)
4. **"Comenzi" - "Plati":** One-to-One
(fiecare comandă are o singură înregistrare de plată)
5. **"Comenzi" - "Transport":** Many-to-One
(o comandă poate fi asociată cu o singură modalitate de transport, dar o modalitate de transport poate fi utilizată pentru mai multe comenzi)
6. **"Clienți" - "Recenzii":** One-to-Many
(un client poate lăsa mai multe recenzii, dar o recenzie este scrisă de un singur client)
7. **"Clienți" - "Wishlist":** One-to-One
(fiecare client are o singură listă de produse preferate)
8. **"Magazine" - "Angajați":** One-to-Many
(un magazin poate avea mai mulți angajați, dar un angajat lucrează într-un singur magazin)
9. **"Comenzi" - "Produse":** Many-to-Many
(o comandă poate include mai multe produse și un produs poate fi achiziționat în mai multe comenzi)
10. **"Produse" - "Wishlist":** Many-to-Many
(un produs poate fi adăugat în mai multe liste de dorințe și o listă de dorințe poate include mai multe produse)
11. **"Produse" - "Magazine":** Many-to-Many
(un produs poate fi adăugat în mai multe magazine și un magazin poate include mai multe produse)
12. **"Produse" - "Recenzii":** One-to-Many
(un produs poate avea mai multe recenzii și o recenzie poate include un singur produs)

Cerința 5.

Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.

Precizari :

Cheile Primare/Straine - tip de date number, valori implicite generate automat

Toate attributele inafara de cheile primare si straine au : constrângere not null, valoare implicită null

1."Produse"

cheie primară: ID_produș , nume_produș (varchar(50)), descriere_produș (text) , pret_produș (float), stoc_produș (int), ID_producator , ID_categorie

2."Producatori"

cheie primară: ID_producator , nume_producator (varchar(50)), tara_producator (varchar(50))

3."Categorii"

cheie primară: ID_categorie, nume_categorie (varchar(50)), descriere_categorie (text)

4."Clienti"

cheie primară: ID_client, nume_client (varchar(50)), prenume_client (varchar(50)), email_client (varchar(35)), telefon_client (varchar(20)), data_inregistrare (date)

5."Comenzi"

cheie primară: ID_comanda, data_plasare (date), status_comanda (varchar(50)), ID_client, ID_transport, ID_plata

6."Magazine"

cheie primară: ID_magazin, nume_magazin (varchar), adresa_magazin (varchar)

7."Recenzii"

cheie primara: ID_recenzie, nota_recenzie (number(1,1)), ID_client, ID_produc

8."Plati"

cheie primara: ID_plata, total_plata (number(10,2)) , tip_plata(varchar(50))

9."Transport"

cheie primara: ID_transport, firma_transport (varchar(50)), cost_transport (number(10,2))

10."Wishlist"

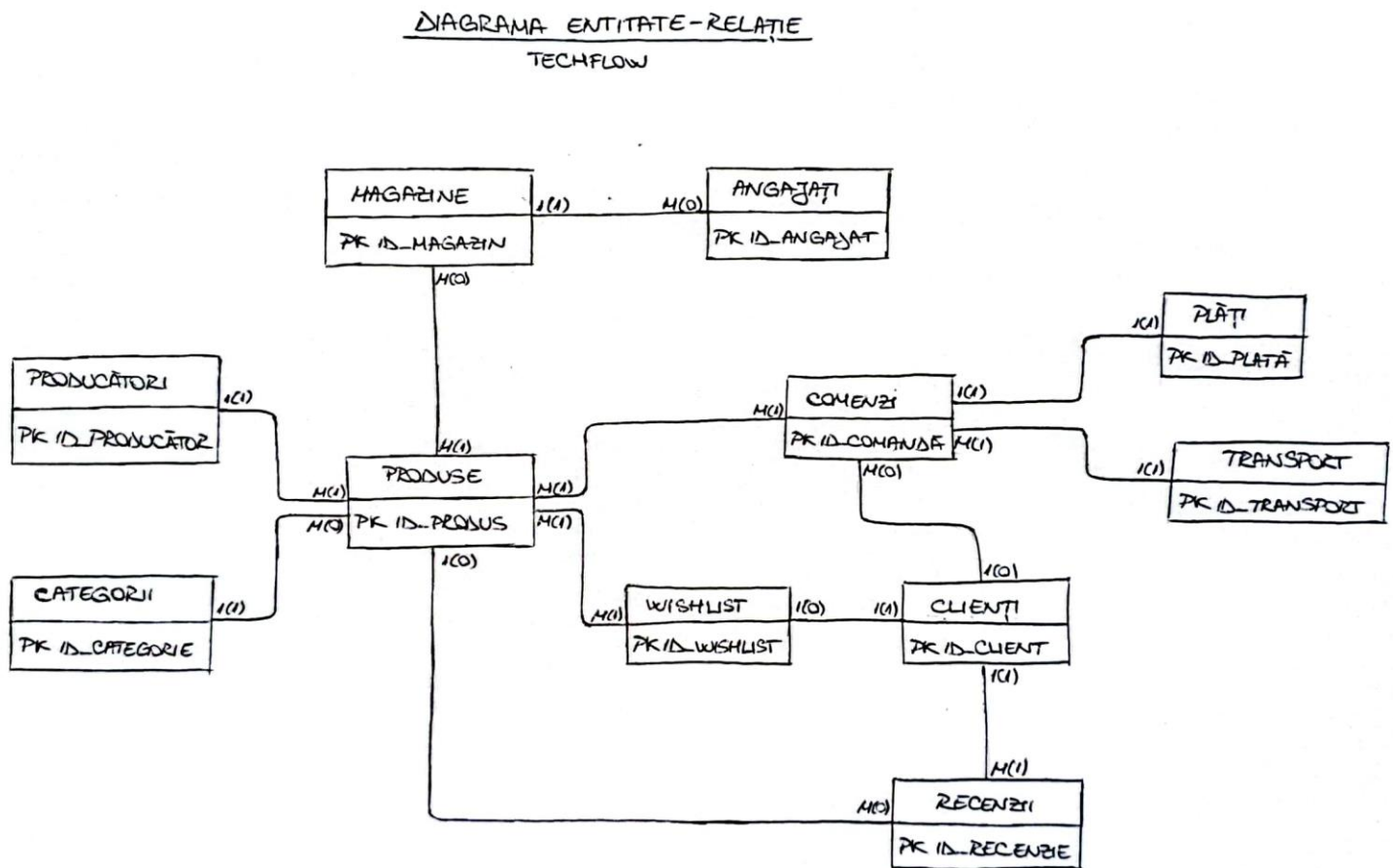
ID_wishlist, data_adaugare (date) , ID_client

11."Angajati"

ID_angajat, nume_angajat (varchar(50)), prenume_angajat (varchar(50)), email_angajat (varchar(50)), salariu_angajat (number(10,2)), job_angajat (varchar(50)), ID_magazin

Cerința 6.

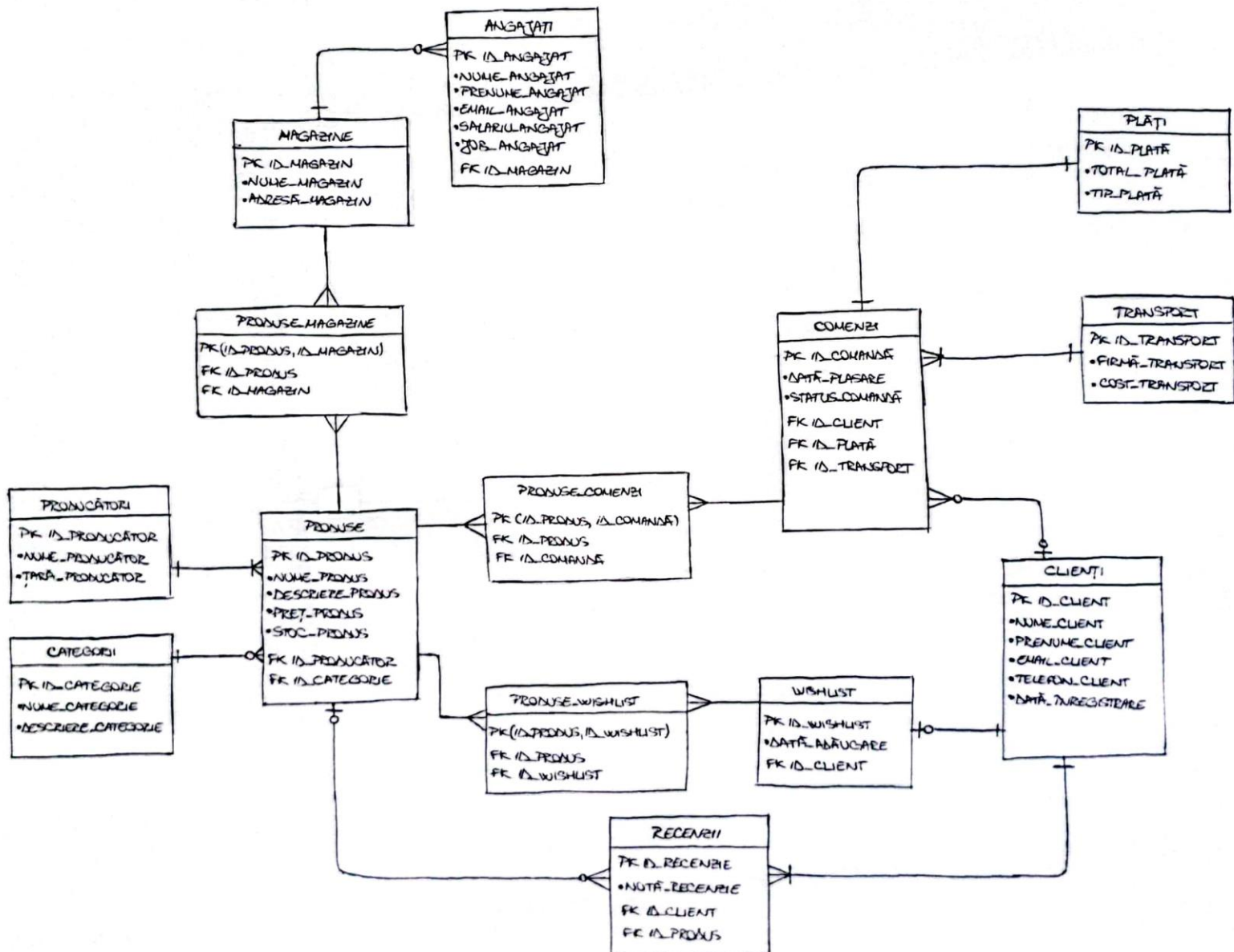
Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.



Cerința 7.

Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6.

DIAGRAMA CONCEPTUALĂ
TECHFLOW



Cerința 8.

Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.

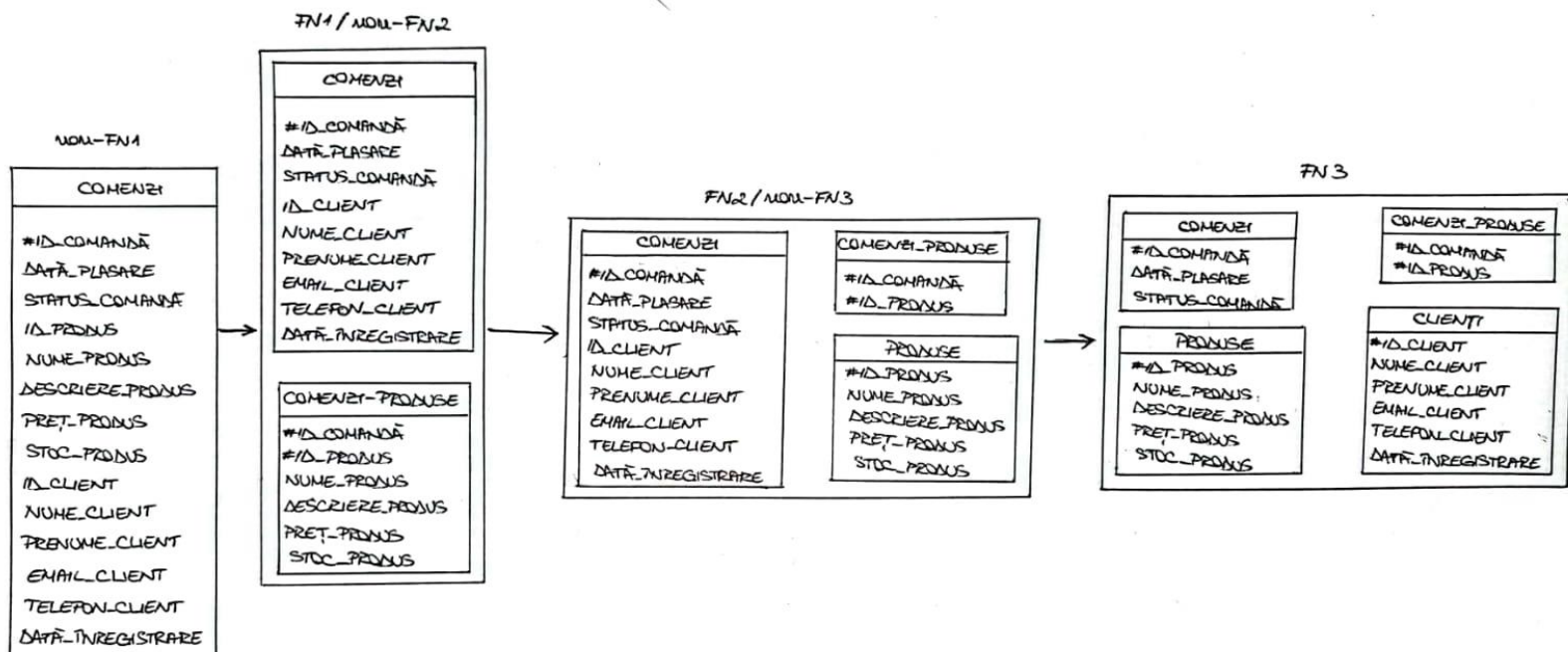
1. **angajati**(id_angajat, nume_angajat, prenume_angajat, email_angajat, salariu_angajat, job_angajat, id_magazin)
2. **magazine**(id_magazin, nume_magazin, adresa_magazin)
3. **produse**(id_produș, nume_produș, descriere_produș, pret_produș, stoc_produș, id_producator, id_categorie)
4. **producatori**(id_producator, nume_producator, tara_producator)
5. **categorii**(id_categorie, nume_categorie, descriere_categorie)
6. **wishlist**(id_wishlist, data_adaugare, id_client)
7. **clienti**(id_client, nume_client, prenume_client, email_client, telefon_client, data_inregistrare)
8. **recenzii**(id_recenzie, nota_recenzie, id_client, id_produș)
9. **comenzi**(id_comanda, data_plasare, status_comanda, id_client, id_plata, id_transport)
10. **plati**(id_plata, total_plata, tip_plata)
11. **transport**(id_transport, firma_transport, cost_transport)

Tabele asociative:

12. **produse_magazine**(id_produș, id_magazin)
13. **produse_wishlist**(id_produș, id_wishlist)
14. **produse_comenzi**(id_produș, id_comanda)

Cerința 9.

Realizarea normalizării până la forma normală 3 (FN1-FN3).



Prima formă normală (FN1) :

O relație este în FN1 dacă domeniile pe care sunt definite attributele relației sunt constituite numai din valori atomice. Un tuplu nu trebuie să conțină attribute sau grupuri de attribute repetitive. Aducerea relațiilor în FN1 presupune eliminarea atributelor compuse și a celor repetitive.

Deoarece o comandă poate avea unul sau mai multe produse, informațiile legate de produse vor fi separate într-un alt tabel.

A doua formă normală (FN2) :

O relație se află în a doua formă normală FN2 dacă:

1. se află în forma normală FN1
2. fiecare atribut care nu este cheie este dependent de întreaga cheie primară.

Deoarece nume_produs, descriere_produs, pret_produs si stoc_produs depinde doar de id_produs, se va adăuga un tabel separat.

A treia formă normală (FN3) :

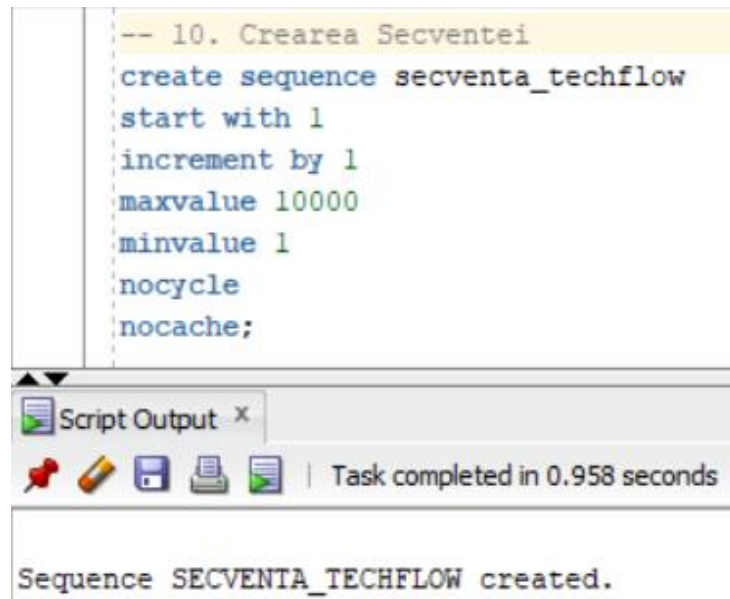
O relație este în forma normală trei FN3 dacă:

1. se găsește în FN2
2. fiecare atribut care nu este cheie depinde direct de cheia primară.

În tabelul COMENZI se observa ca atributul id_client determina in mod unic attributele : nume_client, prenume_client, email_client, telefon_client, data_inregistrare. Astfel se va adauga un nou tabel numit CLIENTI cu cheia primara id_client avand attributele mentionate.

Cerința 10.

Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).



```
-- 10. Crearea Secventei  
create sequence secventa_techflow  
start with 1  
increment by 1  
maxvalue 10000  
minvalue 1  
nocycle  
nocache;
```

Script Output x

Task completed in 0.958 seconds

Sequence SECVENTA_Techflow created.

Cerința 11.

Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea.

a). Crearea Tabelelor

Table MAGAZINE created.	Table PRODUSE_MAGAZINE created.	Table PRODUSE_COMENZI created.
Table ANGAJATI created.	Table PLATII created.	Table WISHLIST created.
Table PRODUCATORI created.	Table TRANSPORT created.	Table PRODUSE_WISHLIST created.
Table CATEGORII created.	Table CLIENTII created.	Table RECENZII created.
Table PRODUSE created.	Table COMENZI created.	

b). Inserarea Datelor

ID_ANGAJAT	NUME_ANGAJAT	PRENUME_ANGAJAT	EMAIL_ANGAJAT	SALARIU_ANGAJAT	JOB_ANGAJAT	ID_MAGAZIN
1	6 Popescu	Ion	ion.popescu@example.com	2500	Casier	1
2	7 Ionescu	Maria	maria.ionescu@example.com	3000	Manager	1
3	8 Popa	Andrei	andrei.popa@example.com	2200	Casier	2
4	9 Dragomir	Elena	elena.dragomir@example.com	2800	Manager	2
5	10 Georgescu	Alexandru	alexandru.georgescu@example.com	2400	Casier	3
6	11 Gheorghe	Mihai	mihai.gheorghe@example.com	3100	Manager	3
7	12 Pop	Adrian	adrian.pop@example.com	2300	Casier	4
8	13 Vasile	Ioana	ioana.vasile@example.com	2900	Manager	4
9	14 Stan	Daniel	daniel.stan@example.com	2600	Casier	5
10	15 Nistor	Cristina	cristina.nistor@example.com	3300	Manager	5

ID_CATEGORIE	NUME_CATEGORIE	DESCRIERE_CATEGORIE
1	21 Placi de baza	Categorie dedicata placilor de baza
2	22 Procesoare	Categorie dedicata procesoarelor
3	23 Placi video	Categorie dedicata placilor video
4	24 Memorii RAM	Categorie dedicata modulelor de memorie RAM
5	25 Stocare ROM	Categorie dedicata solutiilor de stocare

ID_C...	NUME_CLIENT	PRENUME_CLIENT	EMAIL_CLIENT	TELEFON_CLIENT	DATA_INREGISTRARE
1	46 Avram	Gabriela	gabriela.avram@example.com	0776123456	12-MAY-23
2	47 Constantin	Mihai	mihai.constantin@example.com	0787123456	15-MAR-23
3	48 Dumitru	Adriana	adriana.dumitru@example.com	0798123456	14-JAN-23
4	49 Florescu	Valentin	valentin.florescu@example.com	0709123456	13-FEB-23
5	50 Gheorghe	Cristina	cristina.gheorghe@example.com	0710123456	12-MAR-23

ID_COM...	DATA_PLASARE	STATUS_COMANDA	ID_CLIENT	ID_PLATA	ID_TRANSPORT
1	51 15-MAY-23	Noua	46	36	41
2	52 10-MAY-23	Confirmata	47	37	41
3	53 15-APR-23	Anulata	46	38	41
4	54 15-MAY-23	Noua	48	39	42
5	55 09-MAY-23	Confirmata	49	40	44

ID_MAGAZIN	NUME_MAGAZIN	ADRESA_MAGAZIN
1	1 TechFlow Red	Bacau
2	2 TechFlow Blue	Bucuresti
3	3 TechFlow Black	Cluj
4	4 TechFlow Green	Iasi
5	5 TechFlow Yellow	Brasov

ID_COMANDA	DATA_PLASARE	STATUS_COMANDA	ID_CLIENT	ID_PLATA	ID_TRANSPORT
1	51 15-MAY-23	Noua	46	36	41
2	52 10-MAY-23	Confirmata	47	37	41
3	53 15-APR-23	Anulata	46	38	41
4	54 15-MAY-23	Noua	48	39	42
5	55 09-MAY-23	Confirmata	49	40	44

ID_PLATA	TOTAL_PLATA	TIP_PLATA
1	36	2598.99 Card
2	37	1699.97 Numerar
3	38	4199.98 Transfer bancar
4	39	5899.98 Card
5	40	2499.98 Numerar

Humă Ștefan-Dorian
Grupa 143

ID_PR...	NUME_PRODUS	DESCRIERE_PRODUS	PRET_PRODUS	STOC_PRODUS	ID_PRODUCATOR	ID_CATEGORIE
1	26 ASUS ROG Strix X570-E Gaming	Placă de bază ATX pentru gaming, compatibilă cu procesoare AMD Ryzen	1599.99	10	16	21
2	27 Intel Core i9-10900K	Procesor Intel de ultimă generație cu 10 nuclee și 20 fire de execuție	2899.99	5	16	22
3	28 NVIDIA GeForce RTX 3080	Placă video high-end pentru gaming, cu 10 GB memorie GDDR6X	5299.99	3	17	23
4	29 Corsair Vengeance RGB Pro	Modul de memorie DDR4 cu iluminare RGB și capacitate de 16 GB	699.99	20	17	24
5	30 Samsung 970 EVO Plus	Solid State Drive NVMe cu capacitate de 1 TB și viteză de transfer ridicată	999.99	8	17	25
6	31 MSI B450 TOMAHAWK MAX	Placă de bază ATX pentru procesoare AMD Ryzen, suportă memorie DDR4	599.99	15	18	21
7	32 AMD Ryzen 7 5800X	Procesor AMD cu 8 nuclee și 16 fire de execuție, frecvență de bază 3.8 GHz	1999.99	7	18	22
8	33 Gigabyte GeForce GTX 1660 SUPER OC	Placă video performantă pentru gaming, 6 GB memorie GDDR6	1799.99	10	19	23
9	34 RAM Kingston HyperX Fury RGB	Modul de memorie DDR4 cu iluminare RGB și capacitate de 32 GB	1299.99	12	19	24
10	35 SSD Western Digital Blue	Solid State Drive SATA cu capacitate de 500 GB și viteză de transfer mare	399.99	25	20	25

ID_PRODUS	ID_COMANDA
1	26 51
2	30 51
3	29 52
4	35 52
5	31 52
6	27 53
7	34 53
8	28 54
9	31 54
10	29 55
11	33 55

ID_PRODUS	ID_MAGAZIN
1	26 1
2	27 1
3	26 2
4	29 2
5	30 3
6	27 3
7	32 4
8	33 4
9	34 5
10	33 5

ID_PRODUS	ID_WISHLIST
1	27 56
2	28 56
3	29 57
4	31 57
5	31 58
6	27 58
7	34 59
8	27 59
9	31 60
10	28 60
11	26 60

ID_RECENZIE	NOTA_RECENZIE	ID_CLIENT	ID_PRODUS
1	66	46	26
2	67	47	27
3	68	48	28
4	69	49	29
5	70	50	30

ID_TRANS...	FIRMA_TRANSPORT	COST_TRANSPORT
1	41 Cargus	100
2	42 Fan Courier	150.5
3	43 Same Day	200.75
4	44 DPD	180.25
5	45 DHL	220

ID_WISHLIST	DATA_ADAUGARE	ID_CLIENT
1	56 14-MAY-23	46
2	57 13-MAY-23	47
3	58 12-MAY-23	48
4	59 11-MAY-23	49
5	60 10-MAY-23	50

Cerința 12.

Formulați în limbaj natural și implementați 5 cereri SQL complexe.

```
-- 1.
-- Afiseaza numele produselor și prețurile acestora din toate comenzile plasate de clientul Avram Gabriela.

-- subcerere sincronizata în care intervin 3 tabele ( produse, produse_comenzi, clienti)
-- utilizarea a doua functii pe siruri de caractere

select p.num_e_produs, p.pret_produs
from produse p
where exists (
  select 'x'
  from produse_comenzi pc
  join comenzi c on pc.id_comanda = c.id_comanda
  join clienti cl on c.id_client = cl.id_client
  where pc.id_produs = p.id_produs
  and lower(cl.num_e_client) = 'avram'
  and lower(cl.prenume_client) = 'gabriela'
);
```

Query Result x

SQL | All Rows Fetched: 4 in 0.776 seconds

NUME_PRODUS	PRET_PRODUS
1 ASUS ROG Strix X570-E Gaming	1599.99
2 Intel Core i9-10900K	2899.99
3 Samsung 970 EVO Plus	999.99
4 RAM Kingston HyperX Fury RGB	1299.99

```
-- 2.
-- Pentru fiecare magazin, sa se afiseze denumirea acestuia,
-- precum si numele și salariul celor mai slab platiti angajati din cadrul sau.

-- subcerere nesincronizata în clauza FROM

select m.num_e_magazin, ang.num_e, ang.salariu
from magazine m,
  (select a.id_magazin, a.num_e_angajat as num_e, a.salariu_angajat as salariu
   from angajati a
   where a.salariu_angajat = (select min(a2.salariu_angajat)
                             from angajati a2
                             where a2.id_magazin = a.id_magazin)
  ) ang
where m.id_magazin = ang.id_magazin;
```

Query Result x

SQL | All Rows Fetched: 5 in 0.065 seconds

NUME_MAGAZIN	NUME	SALARIU
1 TechFlow Red	Popescu	2500
2 TechFlow Blue	Popa	2200
3 TechFlow Green	Pop	2300
4 TechFlow Yellow	Stan	2600
5 TechFlow Black	Georgescu	2400


```
-- 3.
-- Afisati codul magazinului, numele magazinului, salariul mediu si numarul de produse pentru fiecare magazin.
-- Filtrati rezultatele astfel incat sa includa doar magazinele care au cel putin 2 produse si salariul fiecarui
-- angajat sa fie mai mic decat cel mai mic salariu mediu al angajatilor de la oricare alt magazin.

-- grupări de date cu subcereri nesincronizate in care intervin cel putin 3 tabele, funcții grup, filtrare la nivel de grupuri

select m.id_magazin, m.nume_magazin, avg(a.salariu_angajat) as salariu_meniu, count(pm.id_produș) as numar_produș
from magazine m
join angajati a on m.id_magazin = a.id_magazin
join produse_magazine pm on m.id_magazin = pm.id_magazin
where a.salariu_angajat < ( select min(avg_salariu)
                           from (select avg(salariu_angajat) AS avg_salariu
                                from angajati
                                group by id_magazin
                                )
                           )
group by m.id_magazin, m.nume_magazin
having count(pm.id_produș) >= 2;
```

Query Result x

SQL | All Rows Fetched: 3 in 0.011 seconds

ID_MAGAZIN	NUME_MAGAZIN	SALARIU_MEDIU	NUMAR_PRODUSE
1	2 TechFlow Blue	2200	2
2	3 TechFlow Black	2400	2
3	4 TechFlow Green	2300	2

```
-- 4.
-- Afisati numele produsului, nota recenziei (inlocuind valorile nule cu 0) si starea recenziei
-- (afisand "Fara recenzii" pentru valorile nule si "Cu recenzii" pentru valorile nenule) pentru fiecare produs.
-- Rezultatele sunt sortate in ordine descrescatoare dupa numele produsului.

-- ordonare si utilizarea funcțiilor NVL și DECODE (in cadrul aceleiasi cereri)

select p.nume_produș, nvl(r.nota_recenzie, 0) as nota_recenzie,
       decode(r.nota_recenzie, null, 'Fără recenzii', 'Cu recenzii') as stare_recenzie
from produse p
left join recenzii r on p.id_produș = r.id_produș
order by p.nume_produș desc;
```

Query Result x

SQL | All Rows Fetched: 10 in 0.036 seconds

NUME_PRODUS	NOTA_RECENZIE	STARE_RECENZIE
1 Samsung 970 EVO Plus	3.5	Cu recenzii
2 SSD Western Digital Blue	0	Fără recenzii
3 RAM Kingston HyperX Fury RGB	0	Fără recenzii
4 NVIDIA GeForce RTX 3080	4.2	Cu recenzii
5 MSI B450 TOMAHAWK MAX	0	Fără recenzii
6 Intel Core i9-10900K	3.8	Cu recenzii
7 Gigabyte GeForce GTX 1660 SUPER OC	0	Fără recenzii
8 Corsair Vengeance RGB Pro	5	Cu recenzii
9 ASUS ROG Strix X570-E Gaming	4.5	Cu recenzii
0 AMD Ryzen 7 5800X	0	Fără recenzii

```
-- 5.
-- Sa se afiseze numele clientului, suma totala a platilor efectuate de fiecare client, luna si ziua inregistrarii clientului si
-- o comparatie a platilor fiecarui client cu media platilor pentru toti clientii.
-- Media platilor sa fie calculata pe baza sumelor totale ale fiecarui client.

-- 2 funcții pe date calendaristice , clauza WITH
with
plata_totala_client as ( select cl.id_client,cl.numa_client,cl.preume_client,sum(p.total_plata) as plata_client,cl.data_inregistrare
                        from clienti cl
                        join comenzi co on cl.id_client = co.id_client
                        join plati p on co.id_plata = p.id_plata
                        group by cl.id_client,cl.numa_client,cl.preume_client,cl.data_inregistrare
                        ),
plata_medie_client as (select avg(plata_client) AS plata_medie
                       from plata_totala_client
                       )
select ptc.numa_client, ptc.preume_client,ptc.plata_client,
to_char(ptc.data_inregistrare,'MM') luna_inregistrare,to_char(ptc.data_inregistrare,'DD') zi_inregistrare,
case
  when ptc.plata_client > pmc.plata_medie then 'Peste medie'
  when ptc.plata_client < pmc.plata_medie then 'Sub medie'
  else 'Medie'
end as comparatie
from
  plata_totala_client ptc,
  plata_medie_client pmc;
```

Query Result x

SQL | All Rows Fetched: 4 in 0.058 seconds

	NUME_CLIENT	PRENUME_CLIENT	PLATA_CLIENT	LUNA_INREGISTRARE	ZI_INREGISTRARE	COMPARATIE
1	Florescu	Valentin	2499.98 02	13		Sub medie
2	Avram	Gabriela	6798.97 05	12		Peste medie
3	Constantin	Mihai	1699.97 03	15		Sub medie
4	Dumitru	Adriana	5899.98 01	14		Peste medie

Cerința 13.

Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.

```
--1 Actualizarea unui client
update clienti
set nume_client = 'Noul Nume', prenume_client = 'Noul Prenume'
where id_client = (
    select id_client
    from clienti
    where nume_client = 'Vechiul Nume' and prenume_client = 'Vechiul Prenume'
);

--2 Stergerea comenzilor unui client
delete from comenzi
where id_client = (
    select id_client
    from clienti
    where nume_client = 'Nume Client' and prenume_client = 'Prenume Client'
);

--3 Actualizarea platilor dupa o anumita data cu un adaos de 10%
update plati
set total_plata = total_plata * 1.1
where id_plata in (
    select id_plata
    from comenzi
    where data_plasare > to_date('2022-01-01', 'YYYY-MM-DD')
);
```

Cerința 14.

Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.

```
-- Cerinta 14

-- Rezultatul final al vizualizarii va fi un set de date care va afisa numele si prenumele clientului,
-- numarul total de comenzi, suma totala a platilor si data cea mai recenta de plasare
-- a unei comenzi pentru fiecare client.

create view vizualizare_comenzi as

select c.num_client, c.prenume_client, count(co.id comanda) as numar_comenzi,
sum(p.total_plata) AS suma_totala_plati, max(co.data_plasare) AS ultima_data_plasare
from clienti c
join comenzi co on c.id_client = co.id_client
join plati p on co.id_plata = p.id_plata
group by c.num_client, c.prenume_client;
```

Script Output x Query Result x
Task completed in 0.125 seconds

View VIZUALIZARE_COMENZI created.

```
-- Exemple de operatii LMD permise
-- SELECT: Puteti executa operatii de selectare asupra vizualizarii

select * from vizualizare_comenzi;

-- JOIN: Puteti efectua operatii de imbinare cu alte tabele

select vc.*,c.email_client,c.telefon_client
from vizualizare_comenzi vc
join clienti c on vc.num_client=c.num_client;

-- Exemplu de operatie LMD nepermisa pe vizualizare
-- UPDATE:
update vizualizare_comenzi
set numar_comenzi = 10
where num_client = 'Dumitru';
```


Cerința 15.

Implementați în SQL: o cerere ce utilizează operația outer-join pe minimum 4 tabele, o cerere ce utilizează operația division și o cerere care implementează analiza top-n.

```
-- o cerere ce utilizează operația outerjoin pe minimum 4 tabele
-- Sa se afiseze numele magazinului, id-ul magazinului, id-ul produsului
-- si id-ul recenziei pentru fiecare inregistrare disponibila in baza de date.
select m.nume_magazin, pm.id_magazin, p.id_produs, r.id_recenzie
from magazine m
left join produse_magazine pm on m.id_magazin = pm.id_magazin
inner join produse p on pm.id_produs = p.id_produs
right join recenzii r on pm.id_produs = r.id_produs;
```

Query Result x

SQL | All Rows Fetched: 7 in 0.101 seconds

	ID_MAGAZIN	ID_PRODUS	ID_RECENZIE
1 TechFlow Red	1	27	67
2 TechFlow Red	1	26	66
3 TechFlow Blue	2	26	66
4 TechFlow Blue	2	29	69
5 TechFlow Black	3	30	70
6 TechFlow Black	3	27	67
7 (null)	(null)	(null)	68

```
-- o cerere care implementeaza analiza top-n
-- Afiseaza primele 5 produse (descrescator) in functie de numarul de recenzii
select *
from (
  select p.nume_produs, count(r.id_recenzie) as numar_recenzii
  from produse p
  left join produse_magazine pm on p.id_produs = pm.id_produs
  left join recenzii r on pm.id_produs = r.id_produs
  group by p.nume_produs
  order by count(r.id_recenzie) desc
)
where rownum <= 3;
```

Query Result x

SQL | All Rows Fetched: 3 in 0.04 seconds

ID_PRODUS	NUMAR_RECENZII
1 ASUS ROG Strix X570-E Gaming	2
2 Intel Core i9-10900K	2
3 Corsair Vengeance RGB Pro	1


```
-- o cerere ce utilizează operația division
-- Verificati daca exista produse care au fost comandate de toti clientii.
select
case when count(*) > 0 then 'Exista produse'
     else 'Nu exista produse'
end as rezultat
from produse p
where not exists (
  select c.id_client
  from clienti c
  where not exists (
    select pc.id_produș
    from produse_comenzi pc
    where pc.id_produș = p.id_produș
    and pc.id_comanda in (
      select co.id_comanda
      from comenzi co
      where co.id_client = c.id_client
    )
  )
);
```

Query Result x

SQL | All Rows Fetched: 1 in 0.024 seconds

REZULTAT

1 Nu exista produse

Cerința 16.

Două instrucțiuni select echivalente semantic, de comparat din punct de vedere al execuției.

```
-- Cerinta 16

-- sa se afiseze clientii si informatii despre tipul de plata si costul transportului
--pentru toate comenzile nou aparute efectuate de acestia.

-- Metoda 1.
-- 3 join-uri
select cl.ume_client,cl.prenume_client,p.tip_plata,t.cost_transport,co.status_comanda
from clienti cl
join comenzi co on (cl.id_client=co.id_client)
join plati p on (co.id_plata=p.id_plata)
join transport t on (t.id_transport=co.id_transport)
where lower(co.status_comanda) like 'noua';

-- Metoda 2.
-- folosirea unui view pentru optimizare
create view vizualizare_comenzi_noi as
select co.id_client,p.tip_plata,t.cost_transport,co.status_comanda
from comenzi co
join plati p on (co.id_plata=p.id_plata)
join transport t on (t.id_transport=co.id_transport)
where lower(co.status_comanda) like 'noua';

select * from vizualizare_comenzi_noi;

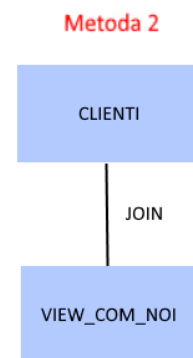
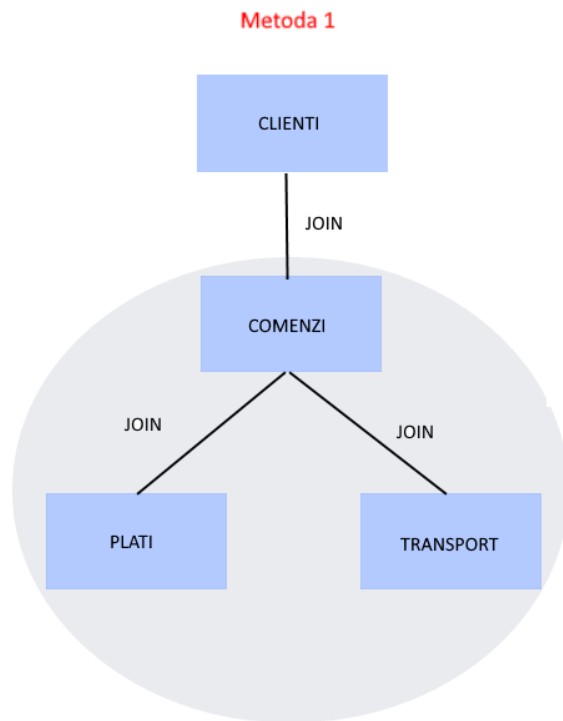
select cl.ume_client,cl.prenume_client,vcn.tip_plata,vcn.cost_transport,vcn.status_comanda
from clienti cl
join vizualizare_comenzi_noi vcn on (cl.id_client=vcn.id_client);
```

pt Output x Query Result x

SQL | All Rows Fetched: 2 in 0.156 seconds

NUME_CLIENT	PRENUME_CLIENT	TIP_PLATA	COST_TRANSPORT	STATUS_COMANDA
1 Avram	Gabriela	Card	100	Noua
2 Dumitru	Adriana	Card	150.5	Noua

Plan de executie:



Cerinta 17.

a. Realizarea normalizării BCNF

-pentru tabelele PRODUSE, CATEGORII, PRODUCATORI (se afla in forma BCNF), dar presupun ca exista un singur tabel PRODUSE care cuprinde toate attributele.

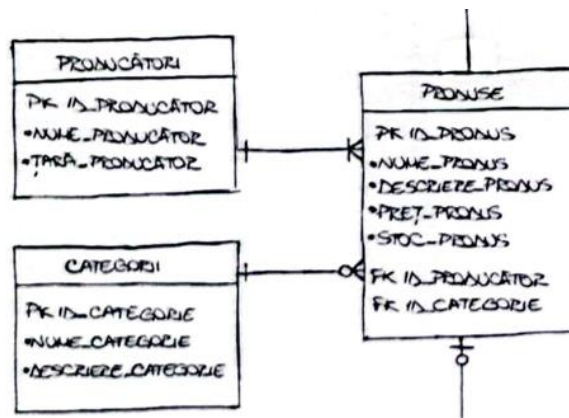
Dependente functionale:

id_produs → nume_produs, descriere_produs, pret_produs, stoc_produs, id_producator, id_categorie

id_producator → nume_producator, tara_producator

id_categorie → nume_categorie, descriere_categorie

**Descompunerea tabelor in functie de dependente functionale:
PRODUSE, CATEGORII, PRODUCATORI.**



b. alegerea unor relații/join-uri din model și reprezentarea acestora într-o bază de date NoSql

(am folosit VSCode si MongoDB Atlas)

Crearea si Inerarea colectiilor : mongodb_creatie_inserare.txt

Exemple : mongodb_exemple.txt

```
156 // Sa se afiseze memoriile ram din baza de date
157 use('mongodbVSCodePlaygroundDB');
158 db.produse.aggregate([
159   {
160     $lookup: {
161       from: "categorii",
162       localField: "id_categorie",
163       foreignField: "id_categorie",
164       as: "categorie"
165     }
166   },
167   {
168     $unwind: "$categorie"
169   },
170   {
171     $lookup: {
172       from: "producatori",
173       localField: "id_producator",
174       foreignField: "id_producator",
175       as: "producator"
176     }
177   },
178   {
179     $unwind: "$producator"
180   },
181   {
182     $project: {
183       nume_produs: 1,
184       descriere_produs: 1,
185       pret_produs: 1,
186       stoc_produs: 1,
187       "categorie.nume_categorie": 1,
188       "producator.nume_producator": 1
189     }
190   },
191   {
192     $match: {
193       "categorie.nume_categorie": "Memorii RAM"
194     }
195   }
196 ]);
```

```
1 {
2   "_id": {
3     "$oid": "6470e16b8a05443e84b92874"
4   },
5   "nume_produs": "Corsair Vengeance RGB Pro",
6   "descriere_produs": "Modul de memorie DDR4 cu iluminare RGB și capacitate de 16 GB",
7   "pret_produs": 699.99,
8   "stoc_produs": 20,
9   "categorie": {
10     "nume_categorie": "Memorii RAM"
11   },
12   "producator": {
13     "nume_producator": "PC Master"
14   }
15 },
16 {
17   "_id": {
18     "$oid": "6470e16b8a05443e84b92879"
19   },
20   "nume_produs": "RAM Kingston HyperX Fury RGB",
21   "descriere_produs": "Modul de memorie DDR4 cu iluminare RGB și capacitate de 32 GB",
22   "pret_produs": 1299.99,
23   "stoc_produs": 12,
24   "categorie": {
25     "nume_categorie": "Memorii RAM"
26   },
27   "producator": {
28     "nume_producator": "VideoMaster"
29   }
30 }
31 ]
```

```
// Sa se afiseze produsele cu un pret mai mare de 2500
use('mongodbVSCodePlaygroundDB');
db.produse.find({ pret_produs: { $gt: 5000 } });
```

```
1 {
2   "_id": {
3     "$oid": "6470e16b8a05443e84b92873"
4   },
5   "nume_produs": "NVIDIA GeForce RTX 3080",
6   "descriere_produs": "Placă video high-end pentru gaming, cu 10 GB memorie GDDR6X",
7   "pret_produs": 5299.99,
8   "stoc_produs": 3,
9   "id_producator": 17,
10  "id_categorie": 23
11 }
12 ]
```

Cerinta 18.

Tranzacții: ilustrarea consistency levels in Oracle cu tranzacții care operează asupra modelului ales.

```
-- Cerinta 18

--T1
create table ang as select * from angajati;
select * from ang;

--dirty write

--T1
select salariu_angajat from ang where id_angajat = 7;
-- valoare: 3000

update ang
set salariu_angajat = salariu_angajat + 1000
where id_angajat = 7;

--T2
update ang
set salariu_angajat = salariu_angajat + 2000
where id_angajat = 7;

--T1
rollback;

--T2
select salariu_angajat from ang where id_angajat = 7;
--valoare : 5000
commit;

--T1
select salariu_angajat from ang where id_angajat = 7;
--valoare: 5000 (nu se inregistreaza 6000)
```

```
--lost update

--T1
select salariu_angajat from ang where id_angajat = 7;
-- valoare: 5000

update ang
set salariu_angajat = salariu_angajat + 1000
where id_angajat = 7;

--T2
update ang
set salariu_angajat = salariu_angajat + 2000
where id_angajat = 7;

--T1
commit;

--T2

select salariu_angajat from ang where id_angajat = 7;
commit;
-- valoare: 8000, nu se inregistreaza 7000

--T1
select salariu_angajat from ang where id_angajat = 7;
-- valoare: 8000
```

Cerinta 19.

Optimizarea a două cereri utilizând indexare.

```
-- Cerinta 19

create index idx_pret_produș on produse (pret_produș);
create index idx_nume_producator on producatori (nume_producator);
create index idx_nume_categorie on categorii (nume_categorie);

-- 1. Afisati toate produsele din interval de pret [1000,2000] si numele producatorilor.
-- optimizare : va fi folosit idx_pret_produș si idx_nume_producator

select p.nume_produș, pr.nume_producator
from produse p
join producatori pr on p.id_producator = pr.id_producator
where p.pret_produș between 1000 and 2000;

-- 2. Afisati numarul de produse din fiecare categorie si numele categoriilor.
-- optimizare : va fi folosit idx_nume_categorie and idk pt id_produș deoarece este PK
-- (indexul este creat o data cu adaugarea constrangerii)

select c.nume_categorie, count(p.id_produș) as numar_produș
from categorii c
join produse p on p.id_categorie = c.id_categorie
group by c.nume_categorie;
```