

PROIECT BAZE DE DATE

Firmă de Organizare Evenimente

Florescu Iuliana

An universitar: 2022-2023

Grupa: 212, seria: 21, An II

Matematică-Informatică

Cuprins

1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.
2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.
3. Descrierea entităților, incluzând precizarea cheii primare.
4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.
5. Descrierea atributelor, incluzând tipul de date și eventualele constrângerii, valori implicite, valori posibile ale atributelor.
6. Descrierea atributelor, incluzând tipul de date și eventualele constrângerii, valori implicite, valori posibile ale atributelor.
7. Descrierea atributelor, incluzând tipul de date și eventualele constrângerii, valori implicite, valori posibile ale atributelor.
8. Descrierea atributelor, incluzând tipul de date și eventualele constrângerii, valori implicite, valori posibile ale atributelor.
9. Realizarea normalizării până la forma normală 3 (FN1-FN3).
10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).
11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative).
12. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:
 - subcereri sincronizate în care intervin cel puțin 3 tabele
 - subcereri nesincronizate în clauza FROM
 - grupări de date cu subcereri nesincronizate în care intervin cel puțin 3 tabele, funcții grup, filtrare la nivel de grupuri (în cadrul aceleiasi cereri)
 - ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiasi cereri)
 - utilizarea a cel puțin 2 funcții pe siruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE
 - utilizarea a cel puțin 1 bloc de cerere (clauza WITH)
13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.

14. Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.
15. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outerjoin pe minimum 4 tabele, o cerere ce utilizează operația division și o cerere care implementează analiza top-n.
16. Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale. Cererea va fi exprimată prin expresie algebraică, arbore algebraic și limbaj (SQL), atât anterior cât și ulterior optimizării.
17.
 - a. Realizarea normalizării BCNF, FN4, FN5
 - b. Aplicarea denormalizării, justificând necesitatea acesteia.

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

Modelul de date oferă informații despre o firmă de organizare evenimente, angajații și clienții acesteia, cât și despre acțiunile pe care partenerii le au pentru promovarea ei. Modelul de date este util pentru fiecare eveniment, astfel încât angajații pot urmări mai ușor statusul proiectelor, dar au și o evidență asupra recenziilor oferite de către clienți și plățile efectuate de aceștia.

Fiecare eveniment are asignat angajați ce provin din diferite departamente, de exemplu PR, IT.

Firma colaborează cu diversi parteneri ce se ocupă de promovare, prin diverse moduri (clipuri de promovare, imagini, colaborări cu vedete) sau de posibile oferte ce au loc în anumite perioade ale anului.

De asemenea, la fiecare sfârșit de sezon important din an se scriu statisticile despre plăți, ce se pot efectua cu cardul sau numerar, sume încasate, recenziile ce sunt oferite de către clienți la finalul colaborării, cele mai căutate evenimente și perioada de desfășurare a acestora.

2. Prezentarea constrângerilor (restrictii, reguli) impuse asupra modelului.

Fiecare departament poate avea mai mulți angajați, dar măcar unul (pentru a putea exista departamentul), dar un angajat poate lucra la un singur eveniment.

Un eveniment poate să fie promovat de anumiți parteneri sau de niciunul.

La finalul colaborării dintre companie și client, aceștia din urma sunt datorii să ofere o recenzie, sau mai multe (în funcție de câte colaborării au avut) care să fie unică la nivel de individ.

Numărul de telefon al clienților sau al partenerilor este unic și nu conține prefixele țărilor.

Fiecare eveniment poate avea mai multe oferte pe an, dar minim una pentru a putea fi gestionat proiectul "Oferte".

3. Descrierea entităților, incluzând precizarea cheii primare.

	Nume Entitate	Descriere	Cheie Primară
1.	Eveniment	Conține detalii despre un eveniment, precum: numele, dăurile de început și sfârșit ale evenimentului.	id_eveniment
2.	Tip_eveniment	Conține detalii despre tipul de eveniment la care se lucrează, de exemplu: nuntă, botez, o delegație, petrecerea unei firme. Este o entitate separată de entitatea “Eveniment”, pentru că fiecare tip de eveniment are asignat diferite tipuri de angajați, promovării, aduce alte finanțării în firmă.	id_tipEveniment
3.	Angajați	Conține informații despre angajați, de exemplu: nume, salariu, funcție, vîrstă.	cnp_angajat

4.	Departament	Conține informații despre pozițiile pe care trebuie să le aibă un angajator ca să facă parte dintr-un departament.	id_departament
5.	Parteneri	Conține informații despre partenerii ce se ocupa de partea de promovare sau organizare oferte ale firmei, de exemplu: nume, locație, număr de telefon, email.	id_partener
6.	Promovare	Conține informații despre acțiunea de promovare a evenimentului. Conține un slogan.	id_promovare
7.	Categorie_promovare	Conține informații despre tipul de promovare utilizat. Audio – prin conferințe de presă. Video – clipuri de promovare. Vizual – pliante/reviste.	id_categorie
8.	Client	Conține informații despre clienți, de exemplu: nume, număr de telefon, adresă, email.	cnp_client

9.	Plată	Conține informații despre plătile efectuate.	id_plată
10.	Recenzie	Conține informații despre recenziile pe care clienții sunt datorii să le ofere la finalul colaborării.	id_recenzie și an_recenzie(există posibilitatea să avem același număr de recenzi, însă anul să fie diferit)
11.	Statistică	Conține informații despre sumele încasate, cel mai căutat eveniment, cea mai căutată perioadă și evidența recenziilor.	id_statistică
12.	Ofertă	Conține informații despre ofertele posibile.	id_ofertă
13.	Locații_departament	Conține informații despre locatiile posibile	Id_locatii
14.	Plangere	Conține informații despre plangerile efectuate	Id_plangere

4. Descrierea relațiilor, incluzând precizarea cardinalității acestora

Eveniment are Angajat = relație care leagă entitățile EVENIMENT și ANGAJAT reflectând legătura dintre acestea (un eveniment are mai mulți angajați asignați). Are cardinalitatea M(1) : 1 (un eveniment trebuie să aibă minim un angajat dar poate avea mai mulți, un angajat poate să lucreze la un singur eveniment).

Angajat aparține Departament = relație care leagă entitățile ANGAJAT și DEPARTAMENT reflectând legătura dintre acestea(un angajat aparține unui departament). Are cardinalitatea 1 : M(1) (un angajat poate să aparțină unui singur

departament, dar un departament poate avea mai mulți angajați, dar minim unul pentru a putea exista departamentul).

Eveniment este TipEveniment = relație care leagă entitățile EVENIMENT și TIP_EVENIMENT reflectând legătura dintre acestea. Are cardinalitatea 1 : M(2) (un eveniment este de un singur tip, dar pot exista mai multe tipuri de evenimente (minim 2 pentru a putea exista entitatea tip_eveniment).

Eveniment are Parteneri = relație care leagă entitățile EVENIMENT și PARTENERI reflectând legătura dintre acestea(un eveniment are parteneri ce se ocupă de promovare). Are cardinalitatea M(0) : 1 (un eveniment poate avea mai mulți parteneri sau niciunul, dar un partener are asignat un singur eveniment de care se poate ocupa).

Eveniment afisează Statistici = relație care leagă entitățile EVENIMENT și STATISTICI reflectând legătura dintre acestea(la finalul unui eveniment se scriu și afișează statisticile). Are cardinalitatea 1 : 1 (un eveniment poate să ofere o singură evidență a statisticilor, iar o statistică aparține unui singur eveniment).

Partener face Ofertă = relație care leagă entitățile PARTENERI și OFERTĂ reflectând legătura dintre acestea (un partener face o ofertă). Are cardinalitatea M(1) : 1 (un partener poate să facă mai multe oferte, dar minim una, iar o ofertă este propusă de un singur partener).

Promovare are CategoriePromovare = relație care leagă entitățile PROMOVARE și CATEGORIE_PROMOVARE reflectând legătura dintre acestea (promovarea este de mai multe tipuri cu caracteristici diferite). Are cardinalitatea 1 : M(2) (o promovare este de un singur tip, dar pot exista minim 2 tipuri pentru a exista aceasta entitate).

Client oferă recenzie = relație care leagă entitățile CLIENT și RECENZIE reflectând legătura dintre acestea (un client oferă recenzie). Are cardinalitatea M(1) : 1 (un client sau mai mulți pot oferi recenzii, dar o recenzie este oferită de un singur client).

Client efectuează Plată = relație care leagă entitățile CLIENT și PLATĂ reflectând legătura dintre acestea (clientul efectuează o plată). Are cardinalitatea M(1) : 1 (un client sau mai mulți pot face plăti, dar o plată este făcută de un singur client).

Client_reclamă_Angajat = relație care leagă entitățile CLIENT și ANGAJAT, reflectând legătura dintre acestea (care sunt angajații cărora un client le face o reclamație/plângere). Ea are cardinalitatea M(0) :M(0) (un client nu este obligat să reclame un angajat, dar un client poate reclama mai mulți angajați iar un angajat nu este obligat să aibă o reclamație făcută, dar un angajat poate fi reclamat de mai mulți clienți) .

Recenzie_se_afișează_Statistică = relație care leagă RECENZIE si STATISTICĂ, reflectand legătura dintre acestea (o recenzie contribuie la statisticile evenimentului). Are cardinalitatea M(1) : 1 (o recenzie/ mai multe este afișată într-o singură statistică).

Departament_are_Locatii_Departament = relație care leagă DEPARTAMENT și LOCATII_DEPARTAMENT, reflectând legătura dintre acestea (un departament are o singură locație). Are cardinalitatea 1 :M(1) (o locație poate avea mai multe departamente).

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

1. Entitate Eveniment

ATRIBUT	TIP DE DATE	DIMENSIUNE	OBSERVAȚII
id_eveniment	number	lungime maximă 5	identificator unic pentru fiecare eveniment (PK)
dată_început	date	lungime maxima 8	NOT NULL deoarece reprezintă data de început a unei oferte
dată_final	date	lungime maxima 8	- NOT NULL deoarece reprezintă data de sfârșit a unei oferte
Nume_eveniment	Varchar2(100)	lungime maximă 100	NOT NULL deoarece reprezintă numele unui angajat.
cnp_angajat	number	lungime maximă 13	reprezintă codul angajatului care lucrează pentru eveniment. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul ANGAJAT, adică este cheie externă în tabelul EVENIMENT(FK).
id_partener	number	lungime maximă 5	reprezintă codul partenerului care

			lucrează pentru eveniment. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul PARTENER, adică este cheie externă în tabelul EVENIMENT(FK).
id_tip_eveniment	number	lungime maximă 5	reprezintă codul tipului de eveniment. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul TIP_EVENIMENT, adică este cheie externă în tabelul EVENIMENT(FK).
id_recenzie	number	lungime maximă 5	reprezintă codul recenziei oferite pentru un eveniment. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul RECENZIE, adică este cheie externă în tabelul EVENIMENT(FK).
id_plată	number	lungime maximă 5	reprezintă codul plății făcute pentru eveniment. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul PLATĂ, adică este cheie externă în tabelul EVENIMENT(FK).
id_promovare	number	lungime maximă 5	reprezintă codul promovarii făcute pentru eveniment. Atributul trebuie să

			corespondă la o valoare a cheii primare din tabelul PROMOVARE, adică este cheie externă în tabelul EVENIMENT(FK).
id_ofertă	number	lungime maximă 5	reprezintă codul ofertei făcute pentru eveniment. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul OFERTA, adică este cheie externă în tabelul EVENIMENT(FK).

2. Entitate Angajat

ATRIBUT	TIP DE DATE	DIMENSIUNE	OBSERVAȚII
Cnp_angajat	number	lungime fixă 13	identificator unic pentru fiecare eveniment (PK)
Nume_angajat	varchar2	lungime maximă 10	NOT NULL deoarece reprezintă numele unui angajat.
salariu	number	lungime maximă de 6 caractere	reprezintă salariul unui angajat este o variabilă de tip numeric(real), pozitivă
funcție	varchar2	lungime maximă 20	NOT NULL căci va conține numele funcției pe care un angajat o are
vârstă	number	lungime maximă 2	- NOT NULL deoarece reprezintă vârstă pe care o are angajatul, valoare mai mare sau egală cu 18

id_departament	number	lungime maxima 5	reprezintă codul departamentului La care lucrează angajatul. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul DEPARTAMENT, adică este cheie externă în tabelul ANGAJAT(FK).
----------------	--------	------------------	--

3. Entitate Client

ATRIBUT	TIP DE DATE	DIMENSIUNE	OBSERVAȚII
cnp_client	number	lungime fixă 13	identificator unic pentru fiecare eveniment (PK)
Nume_client	varchar2	lungime maximă 10	NOT NULL deoarece reprezintă numele unui angajat.
Telefon_client	varchar2	lungime maximă 10	NOT NULL pentru a salva metoda de contact a unui client.
adresă	varchar2	lungime maximă 30	NOT NULL deoarece va conține orașul în care se află clientul
mail	varchar2	lungime maximă 20	NOT NULL deoarece va conține mail-ul clientului, sa aiba 'gmail.com'
id_plată	number	lungime maximă 5	reprezintă id-ul platii efectuate de client Atributul trebuie să corespundă la o valoare a cheii primare din tabelul PLATA, adică este cheie

			externă în tabelul ANGAJAT(FK).
id_recenzie	number	lungime maximă 5	Reprezintă id-ul recenziei efectuate de client Atributul trebuie să corespundă la o valoare a cheii primare din tabelul RECENZIE, adică este cheie externă în tabelul ANGAJAT(FK)
id_plânger e	number	lungime maximă 5	Reprezintă id-ul plangerii efectuate de client Atributul trebuie să corespundă la o valoare a cheii primare din tabelul PLANGERE, adică este cheie externă în tabelul ANGAJAT(FK)

4. Entitate Plată

ATRIBUT	TIP DE DATE	DIMENSIUNE	OBSERVAȚII
id_plată	number	lungime maximă 5	identificator unic pentru fiecare plată (PK)
tip	varchar2	lungime maximă 20	NOT NULL deoarece va conține modalitatea de plată
sumă	number	lungime maximă de 6 caractere	NOT NULL căci va conține suma plătită, valoare pozitiva

5. Entitate Recenzie

ATRIBUT	TIP DE DATE	DIMENSIUNE	OBSERVAȚII
id_recenzie	number	lungime maximă 5	identificator unic pentru fiecare recenzie (PK)
Text_recenzie	varchar10	lungime maxima 30	NOT NULL căci va conține suma recenzia
Nr_stele	number	Lungime maximă de 1	atribut ce reține cât de satisfăcut a fost clientul pe durata unui eveniment, pe o scară de la 1 la 5
an_recenzie	date	lungime maximă 4	identificator unic pentru fiecare recenzie (PK)

6. Entitate Parteneri

ATRIBUT	TIP DE DATE	DIMENSIUNE	OBSERVAȚII
id_partener	number	lungime maximă 5	identificator unic pentru fiecare recenzie (PK)
Nume_partener	varchar2	lungime maximă 10	NOT NULL deoarece reprezintă numele unui partener.
locatie	varchar2	lungime maximă 10	NOT NULL căci va conține locul de unde se află partenerul
Telefon_partener	varchar2	lungime maximă 10	NOT NULL pentru a salva metoda de contact a unui partener.

7. Entitate Promovare

ATRIBUT	TIP DE DATE	DIMENSIUNE	OBSERVAȚII
----------------	--------------------	-------------------	-------------------

id_promovare	number	lungime maximă 5	identificator unic pentru fiecare promovare (PK)
slogan	varchar2	lungime maximă 30	NOT NULL deoarece va conține sloganul de promovare
id_categorie_promovare	number	Lungime maximă 5	reprezintă codul tipului de eveniment care este promovat. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul CATEGORIE_EVENT, adică este cheie externă în tabelul PROMOVARE (FK).

8. Entitate Departament

ATRIBUT	TIP DE DATE	DIMENSIUNE	OBSERVATII
id_departament	number	lungime maximă 5	identificator unic pentru fiecare departament (PK)
Nume_departament	varchar2	Lungime maxima 100	NOT NULL, conține numele departamentului
Numar_angajati	number	Lungime maxima 5	NOT NULL, conține numar de angajati per departament
Id_locatii	number	Lungime maxima 5	reprezintă codul locatiei in care se afla departamentul Atributul trebuie să corespundă la o valoare a cheii primare din tabelul

			LOCATII_DEPARTAMENT, adică este cheie externă în tabelul DEPARTAMENT(FK).
--	--	--	---

9. Entitate Statistică

ATRIBUT	TIP DE DATE	DIMENSIUNE	OBSERVAȚII
id_statistică	number	lungime maximă 5	identificator unic pentru fiecare statistică (PK)
sumă_încasată	number	lungime maximă de 6 caractere	reprezintă suma incasată pe un eveniment. este o variabilă de tip numeric(real), valoare pozitiva
id_recenzie	number	lungime maximă 5	reprezintă codul recenziei făcută. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul RECENZIE, adică este cheie externă în tabelul STATISTICĂ (FK).

10. Entitate Ofertă

ATRIBUT	TIP DE DATE	DIMENSIUNE	OBSERVAȚII
id_ofertă	number	lungime maximă 5	identificator unic pentru fiecare ofertă (PK)
Nume_oferta	varchar2	lungime maximă 10	NOT NULL deoarece reprezintă numele ofertei

reducere	number	lungime maximă2	NOT NULL deoarece reprezintă procentul reducerii
perioada	date	lungime maxima 8	NOT NULL deoarece reprezintă perioada destinață ofertei

11. Entitate Ctegorie_Promovare

ATRIBUT	TIP DE DATE	DIMENSIUNE	OBSERVAȚII
id_categorie_promovare	number	lungime maximă5	identificator unic pentru fiecare categorie (PK)
Nume_categorie_promovare	varchar2	Lungime maxima 30	NOT NULL deoarece reprezintă numele categoriei evenimentului

12. Entitate Tip_Eveniment

ATRIBUT	TIP DE DATE	DIMENSIUNE	OBSERVAȚII
id_tip_eveniment	number	lungime maximă5	identificator unic pentru fiecare tip de eveniment (PK)
Nume_tipeveniment	Varchar2	Lungime maxima 30	NOT NULL deoarece reprezinta numele tipului de eveniment

13. Relația CLIENT_reclamă_ANGAJAT transformată în entitatea PLÂNGERE

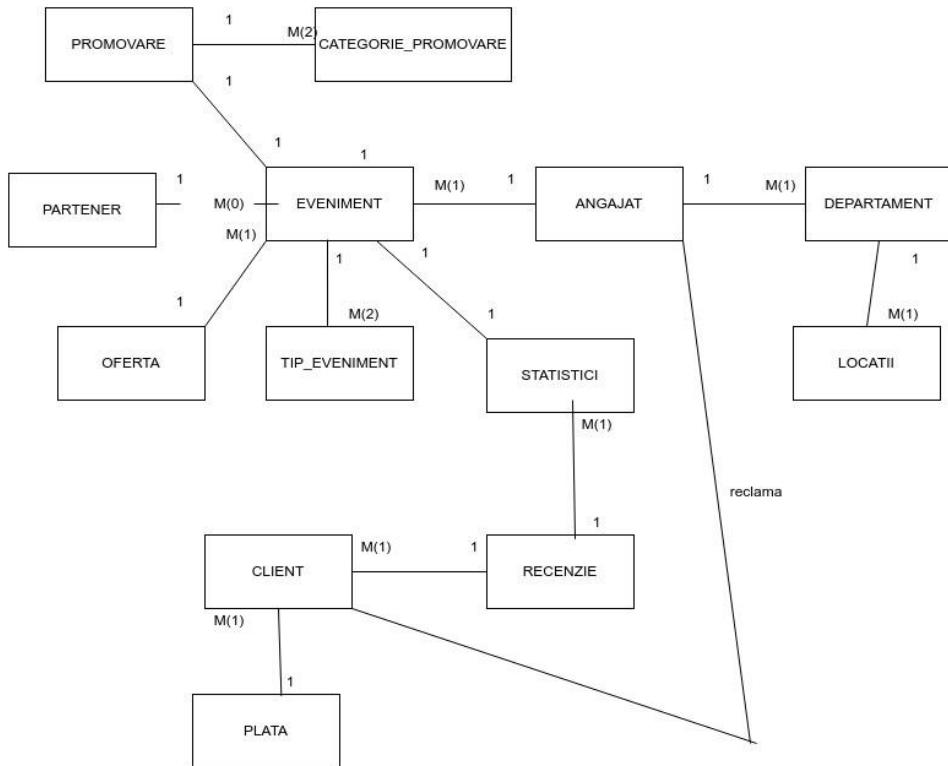
ATRIBUT	TIP DE DATE	DIMENSIUNE	OBSERVAȚII
id_plângere	number	lungime maximă 15	- identificator unic pentru fiecare plângere/

			reclamație făcută de un client (PK)
observație_făcută	varchar2		- NOT NULL căci va conține nemulțumirea unui client față de un angajat
cnp_angajat	number	lungime Fixa 13	- reprezintă codul angajatului ce a primit plângere/reclamație Atributul trebuie să corespundă la o valoare a cheii primare din tabelul ANGAJAT, adică este cheie externă în tabelul PLÂNGERE (FK).

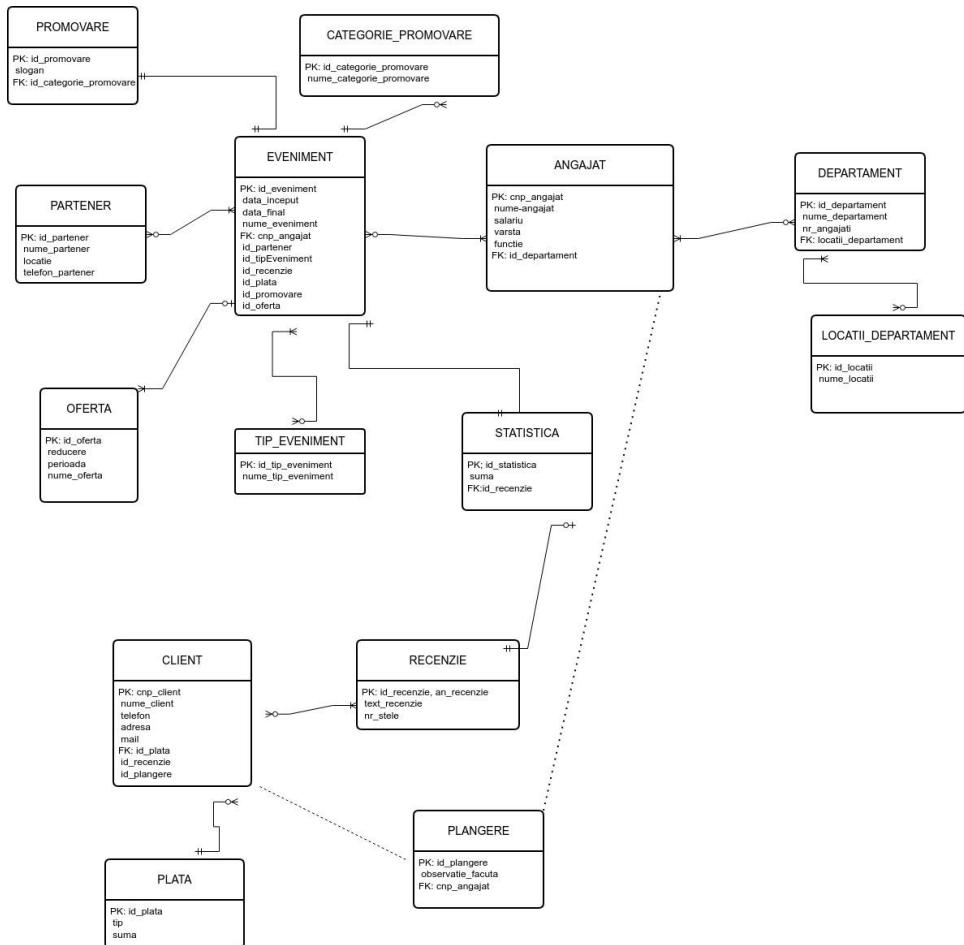
14.ENTITATE LOCATII_DEPARTAMENT

ATRIBUT	TIP DE DATE	DIMENSIUNE	OBSERVATII
Id_locatii	number	Lungime maxima 5	- identificator unic pentru fiecare locatie in care se afla un departament (PK)
Nume_locatii	Varchar2	Lungime maximă - 100 de caractere	- NOT NULL căci va conține locatia unui departament

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



7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 6 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.



8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7

EVENIMENT (id_eveniment#, data_început, data_final, nume, cnp_angajat, id_partener, id_tip_eveniment, id_recenzie, id_plată, id_promovare, id_ofertă)

ANGAJAT (cnp_angajat#, nume, salariu, funcție, vârstă, id_departament)

CLIENT (cnp_client#, nume, telefon, adresă, mail, id_plată, id_recenzie, id_plângere)

PLATĂ (id_plată#, tip, sumă)

RECENTIE (id_recenzie#, an_recenzie#, text, stelet)

PARTENER (id_partener#, nume, locație, telefon, mail)

PROMOVARE (id_promovare#, slogan, id_categorie_promovare)

DEPARTAMENT (id_departament#, cnp_angajat, id_locatii)

STATISTICĂ (id_statistica#, suma_încasată, id_recenzie)

OFERTĂ (id_ofertă#, nume, reducere, perioadă)

CATEGORIE_PROMOVARE (id_categorie#_promovare)

TIP_EVENIMENT (id_tip_eveniment#)

PLÂNGERE (nr_plângere#, observație_făcută, cnp_angajat)

LOCATII_DEPARTAMENT (#id_locatii, nume_locatie)

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

Forma normală 1 (FN1)

O relație se află în FN1 dacă fiecărui atribut care o compune îi corespunde o valoare indivizibilă.

- În entitatea EVENIMENT cheia primară este id_eveniment ceea ce înseamnă că este un identificator unic. Astfel, atributul id_eveniment fiind o valoare indivizibilă, relația se află în FN1 (există identificator unic pentru toate intrările în tabel).
- În entitatea RECENZIE cheia primară este compusă din id_recenzie și an_recenzie. Această cheie primară este un identificator unic pentru entitatea RECENZIE. În acest fel, atributele id_recenzie și an_recenzie sunt împreună o valoare indivizibilă, deci relația se află în FN1 (există identificator unic pentru toate intrările din tabel).
- Pentru a exemplifica această normalizare, presupunem că pentru o plată există mai multe tarife (suma).
→ La început relația *plata_nonFN1* arată:

id_plată#	sumă
1	100, 150, 200
2	300
3	400, 500

Astfel, se poate observa că relația de mai sus nu este FN1, deoarece atributului sumă nu îi corespunde o valoare indivizibilă.

→ După ce a fost aplicată FN1 relația arată:

id_plată#	sumă
1	100
1	150
1	200
2	300
3	400
3	500

Deci, după normalizare fiecărui atribut care o compune îi corespunde o valoare indivizibilă.

Forma normală 2 (FN2)

O relație se află în FN2 dacă și numai dacă aceasta se află în prima formă normală și dacă fiecare atribut care nu face parte din cheie este dependent de întreaga cheie primară.

→ La început relația RECENZIE fără FN2 arăta:

Id_recenzie#	An_recenzie#	text	stele	Id_statistică	Sumă_încasată
1	2022	Mulțumit	5/5	1	5000
1	2023	Total nemulțumit	2/5	2	2000
4	2021	Plăcut surprins	4/5	3	6530
6	2023	Mulțumit	3/5	2	3456

- Relația de mai sus se află în FN1, deoarece există identificator unic pentru toate intrările din tabel.
- Pentru a fi în FN2 trebuie ca fiecare atribut care nu este cheie să fie dependent de întregă cheie primară. În acest caz, attributele *text* și *stele* trebuie să depindă de întregă cheie primară *id_recenzie#* și *id_statistică#*, *an_recenzie#*, *cnp_client#*, dar se poate observa faptul că acestea nu depind direct de toată cheia primară. Acest fapt se explică prin existența dependenței directe dintre *id_statistică#*, *text* și *sumă_încasată*, ceea ce înseamnă că *text*, *sumă_încasată* depind doar de o parte a cheii primare, anume de *id_statistică#*. Astfel, relația prezentată mai sus nu se află în a doua formă normală.

→ Astfel, dependențele vor fi:

$$\{id_statistică\} \rightarrow \{sumă_încasată, text\}.$$

→ Transformarea în FN2:

STATISTICĂ

Id_statistică	Sumă_încasată	Id_recenzie
1	5000	1
2	2000	1
3	6530	4

2	3456	6
---	------	---

RECENZIE

Id_recenzie	An_recenzie	text	Nr_stele
1	2022	Mulțumit	5/5
1	2023	Total nemulțumit	2/5
4	2021	Plăcut surprins	4/5
6	2023	Mulțumit	3/5

Forma normală 3 (FN3)

O relație se află în FN3 dacă și numai dacă se află în FN2 și dacă fiecare atribut care nu este cheie, depinde direct de cheia primară și numai de ea.

- Pentru exemplificarea acestei forme normale s-a ales să se realizeze un exemplu fictiv pentru ANGAJAT (se presupune că atributul *salariu* depinde de *funcție*, fapt ce nu este adevărat în cazul diagramei construite).

→ La început relația *angajat_nonFN1* arată:

Cnp_angajat	nume	funcție	salariu	vârstă	Id_departament
1230987	Andrei	manager	4500	32	1
6754123	Ioana	secretară	2600	20	1
7645209	Laura	organizator	3500	28	3

- Această relație este în prima formă normală, pentru că există identificator unic pentru toate intrările din tabel, anume *cnp_angajat#*.
- Se află și în FN2 deoarece se poate remarcă faptul că attributele *nume*, *funcție*, *salariu*, *vârstă*, *id_departament* care nu sunt în componența cheii primare sunt dependente de întreaga cheie primară *cnp_angajat#*.
- În acest exemplu se observă că atributul *salariu* depinde tranzitiv de cheia primară *cnp_angajat#* prin intermediul atributului *funcție*.
 - * {*cnp_angajat#*} → {*nume*, *funcție*, *varstă*, *id_departament*}
 - * {*id_angajat#*} → {*funcție*} → {*salariu*}

- Pentru a aduce relația angajat_nonFN3 în forma normală 3 se aplică regula Casey-Delobel care presupune descompunerea relației prin eliminarea dependenței funcționale tranzitive. În acest mod, din cauza dependenței funcționale funcție > salariu care surprinde faptul că relația nu se află în FN3, se înlocuiește relația prin două proiecții:
 - ✗ ANGAJAT_FN3 (cnp_angajat#, nume, funcție, vârstă, id_departament)
 - ✗ FUNCȚIE_SALARIU (funcție, salariu)
- Transformarea:

ANGAJAT_FN3

cnp_angajat	nume	funcție	vârstă	id_departament
1230987	Andrei	manager	32	1
6754123	Ioana	secretară	20	1
7645209	Laura	organizator	28	3

FUNCȚIE_SALARIU

funcție	salariu
manager	4500
secretară	2600
organizator	3500

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

A se vedea exercițiul 11, unde s-a făcut inserarea datelor în tabele. Am realizat secventa tabelului Locatii_Departament.

11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative).

EVENIMENT

```
create table Eveniment
(
    id_eveniment number(5) primary key,
    data_inceput date not null,
    data_final date not null,
    nume_eveniment varchar2(100) not null,
    cnp_angajat number(13),
    foreign key (cnp_angajat) references Angajat (cnp_angajat),
    id_partener number(5),
    foreign key (id_partener) references Partener (id_partener),
    id_TipEveniment number(5),
    foreign key (id_TipEveniment) references TipEveniment (id_TipEveniment),
    id_recenzie number(5),
    foreign key (id_recenzie) references Recenzie (id_recenzie),
    id_plata number(5),
    foreign key (id_plata) references Plata (id_plata),
    id_promovare number(5),
    foreign key (id_promovare) references Promovare (id_promovare),
    id_oferta number(5),
    foreign key (id_oferta) references Oferta (id_oferta)
);
```

ANGAJAT

```
create table Angajat
(
    cnp_angajat number(13) primary key,
    nume_angajat varchar(10) not null,
    salariu number(5) not null,
    functie varchar(20) not null,
    varsta number(2) not null,
    id_departament number(5),
    foreign key (id_departament) references Departament (id_departament)
);

alter table Angajat
add check( length(cnp_angajat)=10) and (REGEXP_LIKE(cnp_angajat, '[0-9]'))));
alter table Angajat
add check(salariu > 0);
```

CLIENT

```

create table Client
(
cnp_client number(13) primary key,
nume_client varchar(10) not null,
telefon_client varchar(10) not null,
adresa varchar(20) not null,
mail varchar(20) not null,
id_plata number(5),
foreign key (id_plata) references Plata (id_plata),
id_recenzie number(5),
foreign key (id_recenzie) references Recenzie (id_recenzie),
id_plangere number(5),
foreign key (id_plangere) references Plangere (id_plangere)
);

alter table Client
add check( lenght(cnp_client)=10) and (REGEXP_LIKE(cnp_client, '[0-9]'))));
alter table Client
add check( lenght(telefon_client)=10) and (REGEXP_LIKE(telefon_client, '[0-9]'))));
alter table Client
add check(mail_client like ('gmail.com'));

```

PLATA

```

create table Plata
(
id_plata number(5) primary key,
tip varchar(2) not null,
suma number(6) not null
);

```

	Worksheet	Query Builder
1	ALTER TABLE Plata	
2	ADD CHECK (suma > 0);	
3		
4		

RECENZIE

```
create table Recenzie

(
    id_recenzie number(5) not null,
    text_recenzie varchar(30) not null,
    nr_stele number(1) not null,
    an_recenzie date not null,
    primary key(id_recenzie, an_recenzie)
);
```

```
ALTER TABLE Recenzie
ADD CHECK (nr_stele < 6);
```

PARTENER

```
create table Partener
(
    id_partener number(5) primary key,
    nume_partener varchar(10) not null,
    locatie varchar(10) not null,
    telefon_partener varchar(10) not null,
    add check( (length(telefon_partener)=10) and (REGEXP_LIKE(telefon_partener, '[0-9]')));
```

PROMOVARE

```
create table Promovare
(
    id_promovare number(5) primary key,
    slogan varchar(30) not null,
    id_CategoriePromovare number(5),
    foreign key (id_CategoriePromovare) references CategoriePromovare (id_CategoriePromovare)
);
```

```
alter table Promovare
modify
(
    slogan varchar2(100)
);
```

DEPARTAMENT

```
create table Departament
(
    id_departament number(5) primary key,
    nume_departament VARCHAR2(100) not null,
    numar_angajati number(5) not null
);
```

STATISTICA

```
create table Statistica
(
    id_statistica number(5) primary key,
    suma_incasata number(6) not null,
    id_recenzie number(5),
    foreign key (id_recenzie) references Recenzie (id_recenzie)
);
```

```
ALTER TABLE Statistica
ADD CHECK (suma_incasata > 0);
```

OFERTA

```
create table Oferta
(
    id_oferta number(5) primary key,
    nume_oferta varchar(10) not null,
    reducere number(2) not null,
    perioada date not null
);
```

CATEGORIEPROMOVARE

```
15
16 create table CategoriePromovare
17 (
18     id_categoriePromovare number(5) primary key,
19     numecategoriepromovare varchar2(100) not null
20 );
```

TIPEVENIMENT

```
15
16 create table TipEveniment
17 (
18     id_TipEveniment number(5) primary key,
19     numetipeveniment varchar2(100) not null
20 );
```

PLANGERE

```
Query Builder
create table Plangere
(
    id_plangere number(5) primary key,
    observatie_facuta varchar(30) not null,
    foreign key (cnp_angajat) references Angajat (cnp_angajat)
);
```

LOCATII_DEPARTAMENT

```
create table LocatiiDepartament
(
    id_locatii number(5) primary key,
    NumeLocatii varchar2(100) not null
);
```

INSERARE DATE

DEPARTAMENT

```
insert into Departament(id_departament)
values
(124);

insert into Departament(id_departament)
values
(125);
```

```
insert into Departament(id_departament)
values
(126);

insert into Departament(id_departament)
values
(127);

insert into Departament(id_departament)
values
(128);
```



```
insert into Departament(id_departament)
values
(129);
```



```
UPDATE departament
SET numar_angajati = 3
WHERE id_departament = 123;
```



```
UPDATE departament
SET numar_angajati = 50
WHERE id_departament = 124;
```



```
UPDATE departament
SET numar_angajati = 25
WHERE id_departament = 125;
```



```
UPDATE departament
SET numar_angajati = 20
WHERE id_departament = 126;
```



```
UPDATE departament
SET numar_angajati = 5
WHERE id_departament = 127;
```



```
UPDATE departament
SET numar_angajati = 1
WHERE id_departament = 128;
```



```
UPDATE departament
SET numar_angajati = 10
WHERE id_departament = 129;
```

SQL | All Rows Fetched: 7 in 0,001 seconds

ID_DEPARTAMENT	NUME_DEPARTAMENT	NUMAR_ANGAJATI
1	123 Management	3
2	124 Organizare Evenimente	50
3	125 Colaboratori	25
4	126 Promovare	20
5	127 IT	5
6	128 Contabilitate	1
7	129 Curatenie	10

```

1 update departament
2 set id_locatii = 1
3 where id_departament = 123;
4 update departament
5 set id_locatii = 2
6 where id_departament = 124;
7 update departament
8 set id_locatii = 3
9 where id_departament = 125;
10 update departament
11 set id_locatii = 4
12 where id_departament = 126;
13 update departament
14 set id_locatii = 5
15 where id_departament = 127;
16 update departament
17 set id_locatii = 1
18 where id_departament = 128;
19 update departament
20 set id_locatii = 2
21 where id_departament = 129;

```

ANGAJAT

```
insert into Angajat(cnp_angajat, nume_angajat, salariu, functie, varsta, id_departament)
values
(1234567891234, 'Ana', 2300, 'secretara', 20, 123);
insert into Angajat(cnp_angajat, nume_angajat, salariu, functie, varsta, id_departament)
values
(9876543212345, 'Ion', 4500, 'inginer', 26, 123);
insert into Angajat(cnp_angajat, nume_angajat, salariu, functie, varsta, id_departament)
values
(1230978542675, 'Andrei', 7640, 'manager', 40, 127);
insert into Angajat(cnp_angajat, nume_angajat, salariu, functie, varsta, id_departament)
values
(6093451238745, 'Andreea', 6000, 'organizator', 50, 127);
insert into Angajat(cnp_angajat, nume_angajat, salariu, functie, varsta, id_departament)
values
(6542138065237, 'Ilinca', 4600, 'colaborator', 26, 124);
```

```
UPDATE angajat
SET id_departament = 128
WHERE cnp_angajat = 1234567891234;
```

```
UPDATE angajat
SET id_departament = 127
WHERE cnp_angajat = 9876543212345;
```

```
UPDATE angajat
SET id_departament = 123
WHERE cnp_angajat = 1230978542675;
```

```
UPDATE angajat
SET id_departament = 124
WHERE cnp_angajat = 6093451238745;
```

```
insert into Angajat(cnp_angajat, nume_angajat, salariu, functie, varsta, id_departament)
values
(1928374561029, 'Ionela', 2500, 'femeie de servicii', 30, 129);
insert into Angajat(cnp_angajat, nume_angajat, salariu, functie, varsta, id_departament)
values
(8976123407564, 'Ionel', 4100, 'organizator', 27, 124);
insert into Angajat(cnp_angajat, nume_angajat, salariu, functie, varsta, id_departament)
values
(1256340986123, 'Mihai', 7640, 'promoveaza even', 40, 126);
insert into Angajat(cnp_angajat, nume_angajat, salariu, functie, varsta, id_departament)
values
(5470923419742, 'Silviu', 6500, 'colaborator', 30, 125);
insert into Angajat(cnp_angajat, nume_angajat, salariu, functie, varsta, id_departament)
values
(9870345213496, 'Alexandru', 7000, 'organizator', 26, 124);
```

	CNP_ANGAJAT	NUME_ANGAJAT	SALARIU	FUNCTIE	VARSTA	ID_DEPARTAMENT
1	1928374561029	Ionela	2500	femeie de servici	30	129
2	8976123407564	Ionel	4100	organizator	27	124
3	1256340986123	Mihai	7640	promoveaza even	40	126
4	5470923419742	Silviu	6500	colaborator	30	125
5	9870345213496	Alexandru	7000	organizator	26	124
6	1234567891234	Ana	2300	contabil	20	128
7	9876543212345	Ion	4500	inginer	26	127
8	1230978542675	Andrei	7640	manager	40	123
9	6093451238745	Andreea	6000	organizator	50	124
10	6542138065237	Ilinca	4600	colaborator	26	125

Plângere

```

2 insert into Plangere(id_plangere, observatie_facuta, cnp_angajat)
3 values
4 (1, 'neserios', 1234567891234);
5 insert into Plangere(id_plangere, observatie_facuta, cnp_angajat)
6 values
7 (2, 'prea vorbaret', 9876543212345);
8 insert into Plangere(id_plangere, observatie_facuta, cnp_angajat)
9 values
0 (3, 'sta prea mult in pauza', 1230978542675);
1 insert into Plangere(id_plangere, observatie_facuta, cnp_angajat)
2 values
3 (4, 'prea timid', 6093451238745);
4 insert into Plangere(id_plangere, observatie_facuta, cnp_angajat)
5 values
6 (5, 'nu colaboreaza cu colegii', 6542138065237);
```

```

insert into Plangere(id_plangere, observatie_facuta, cnp_angajat)
values
(6, 'necomunicativ', 9870345213496);
insert into Plangere(id_plangere, observatie_facuta, cnp_angajat)
values
(7, 'nededicat', 5470923419742);
insert into Plangere(id_plangere, observatie_facuta, cnp_angajat)
values
(8, 'rezolva task-urile prea greu', 1256340986123);
insert into Plangere(id_plangere, observatie_facuta, cnp_angajat)
values
(9, 'intarzie', 8976123407564);
insert into Plangere(id_plangere, observatie_facuta, cnp_angajat)
values
(10, 'nu asculta toate parerile', 1928374561029);
```

ID_PLANGERE	OBSERVATIE_FACUTA	CNP_ANGAJAT
1	6 necomunicativ	9870345213496
2	7 nededicat	5470923419742
3	8 rezolva task-urile prea greu	1256340986123
4	9 intarzie	8976123407564
5	10 nu asculta toate parerile	1928374561029
6	1 neserios	1234567891234
7	2 prea vorbaret	9876543212345
8	3 sta prea mult in pauza	1230978542675
9	4 prea timid	6093451238745
10	5 nu colaboreaza cu colegii	6542138065237

Categorie Promovare

```

insert into categoriepromovare(id_categoriepromovare)
values
(12);
insert into categoriepromovare(id_categoriepromovare)
values
(13);
insert into categoriepromovare(id_categoriepromovare)
values
(14);
insert into categoriepromovare(id_categoriepromovare)
values
(15);
insert into categoriepromovare(id_categoriepromovare)
values
(16);

```

```

UPDATE CategoriePromovare
SET NUMECATEGORIEPROMOVARE = 'video'
WHERE ID_CAREGORIEPROMOVARE = 12;

UPDATE CategoriePromovare
SET NUMECATEGORIEPROMOVARE = 'audio'
WHERE ID_CAREGORIEPROMOVARE = 13;

UPDATE CategoriePromovare
SET NUMECATEGORIEPROMOVARE = 'imagine'
WHERE ID_CAREGORIEPROMOVARE = 14;

UPDATE CategoriePromovare
SET NUMECATEGORIEPROMOVARE = 'mass-media'
WHERE ID_CAREGORIEPROMOVARE = 15;

UPDATE CategoriePromovare
SET NUMECATEGORIEPROMOVARE = 'evenimente'
WHERE ID_CAREGORIEPROMOVARE = 16;

```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0,001 seconds

ID_CAREGORIEPROMOVARE	NUMECATEGORIEPROMOVARE
1	12 video
2	13 audio
3	14 imagine
4	15 mass-media
5	16 evenimente

PLATA

```

insert into Plata(id_plata, tip, suma)
values
(1, 'numerar', 1200);
insert into Plata(id_plata, tip, suma)
values
(2, 'card', 3876);
insert into Plata(id_plata, tip, suma)
values
(3, 'numerar', 6000);
insert into Plata(id_plata, tip, suma)
values
(4, 'numerar', 4320);
insert into Plata(id_plata, tip, suma)
values
(5, 'card', 6500);|

```

	ID_PLATA	SUMA	TIP
1	1	1200	numerar
2	2	3876	card
3	3	6000	numerar
4	4	4320	numerar
5	5	6500	card

Promovare

```

select * from Promovare;

insert into Promovare(id_promovare, slogan)
values
(1, 'nunta frumoasa');
insert into Promovare(id_promovare, slogan)
values
(2, 'botez Cristian');
insert into Promovare(id_promovare, slogan)
values
(3, 'intrunire');
insert into Promovare(id_promovare, slogan)
values
(4, 'revedere');
insert into Promovare(id_promovare, slogan)
values
(5, 'intalnire de afaceri');

```

```

insert into Promovare(id_promovare, slogan, ID_CATEGORIEPROMOVARE)
values
(7, 'bine v-am regasit', 12);
insert into Promovare(id_promovare, slogan, ID_CATEGORIEPROMOVARE)
values
(8, 'revedere placuta', 13);
insert into Promovare(id_promovare, slogan, ID_CATEGORIEPROMOVARE)
values
(9, 'botez magic', 14);
insert into Promovare(id_promovare, slogan, ID_CATEGORIEPROMOVARE)
values
(10, 'afaceri prospere', 15);

UPDATE Promovare
SET ID_CATEGORIEPROMOVARE = 12
WHERE id_promovare = 1;
UPDATE Promovare
SET ID_CATEGORIEPROMOVARE = 13
WHERE id_promovare = 2;
UPDATE Promovare
SET ID_CATEGORIEPROMOVARE = 14
WHERE id_promovare = 3;
UPDATE Promovare
SET ID_CATEGORIEPROMOVARE = 16
WHERE id_promovare = 5;
UPDATE Promovare
SET ID_CATEGORIEPROMOVARE = 12
WHERE id_promovare = 1;

insert into Promovare(id_promovare, slogan, ID_CATEGORIEPROMOVARE)
values
(6, 'nunta de neuitat', 16);

```

ID_PROMOVARE	SLOGAN	ID_CATEGORIEPROMOVARE
1	6 nunta de neuitat	16
2	7 bine v-am regasit	12
3	8 revedere placuta	13
4	9 botez magic	14
5	10 afaceri prospere	15
6	1 nunta frumoasa	12
7	2 botez Cristian	13
8	3 intrunire	14
9	4 revedere	15
10	5 intalnire de afaceri	16

Recenzie

```

3 | insert into Recenzie(id_recenzie, text_recenzie, nr_stele, an_recenzie)
4 | values
5 | (1, 'un eveniment frumos', 5, '23-07-2022');
6 | insert into Recenzie(id_recenzie, text_recenzie, nr_stele, an_recenzie)
7 | values
8 | (1, 'un eveniment reusit', 4, '01-09-2022');
9 | insert into Recenzie(id_recenzie, text_recenzie, nr_stele, an_recenzie)
10 | values
11 | (1, 'dezastru', 1, '10-02-2020');
12 | insert into Recenzie(id_recenzie, text_recenzie, nr_stele, an_recenzie)
13 | values
14 | (1, 'frumos', 4, '01-01-2023');
15 | insert into Recenzie(id_recenzie, text_recenzie, nr_stele, an_recenzie)
16 | values
17 | (1, 'nereusit', 2, '12-12-2019');|
```

```

UPDATE recenzie
SET id_recenzie = 3
WHERE an_recenzie = '10-02-2020';
UPDATE recenzie
SET id_recenzie = 4
WHERE an_recenzie = '01-01-2023';
UPDATE recenzie
SET id_recenzie = 5
WHERE an_recenzie = '12-12-2019';
```

Script Output x | Query Result 1 x | Query Result 2 x

SQL | All Rows Fetched: 5 in 0,001 seconds

ID_RECENZIE	TEXT_RECENZIE	NR_STELE	AN_RECENZIE
1	1 un eveniment frumos	5	23-07-2022
2	2 un eveniment reusit	4	01-09-2022
3	3 dezastru	1	10-02-2020
4	4 frumos	4	01-01-2023
5	5 nereusit	2	12-12-2019

Statistica

```

} insert into Statistica(id_statistica, suma_incasata)
values
(10, 2300);
insert into Statistica(id_statistica, suma_incasata)
values
(11, 7000);
insert into Statistica(id_statistica, suma_incasata)
values
(12, 3200);
insert into Statistica(id_statistica, suma_incasata)
values
(13, 5412);
insert into Statistica(id_statistica, suma_incasata)
values
(14, 7011); |



| UPDATE Statistica
| SET id_recenzie = 1
| WHERE ID_statistica = 10;
| UPDATE Statistica
| SET id_recenzie = 2
| WHERE ID_statistica = 11;
| UPDATE Statistica
| SET id_recenzie = 3
| WHERE ID_statistica = 12;
| UPDATE Statistica
| SET id_recenzie = 4
| WHERE ID_statistica = 13;
| UPDATE Statistica
| SET id_recenzie = 5
| WHERE ID_statistica = 14;
|



insert into Statistica(id_statistica, suma_incasata, id_recenzie)
values
(15, 3000, 3);
insert into Statistica(id_statistica, suma_incasata, id_recenzie)
values
(16, 2700, 1);
insert into Statistica(id_statistica, suma_incasata, id_recenzie)
values
(17, 4300, 2);
insert into Statistica(id_statistica, suma_incasata, id_recenzie)
values
(19, 6000, 5);
insert into Statistica(id_statistica, suma_incasata, id_recenzie)
values
(18, 6700, 4); |

```

ID_STATISTICA	SUMA_INCASATA	ID_RECENZIE
1	15	3000
2	16	2700
3	17	4300
4	18	6700
5	19	6000
6	10	2300
7	11	7000
8	12	3200
9	13	5412
10	14	7011

OFERTA

```

3 | insert into Oferta(id_oferta, nume_oferta, reducere, perioada)
4 | values
5 | (1, 'red iunie', 12, '12-07-2023');
6 | insert into Oferta(id_oferta, nume_oferta, reducere, perioada)
7 | values
8 | (2, 'red mai', 30, '23-05-2023');
9 | insert into Oferta(id_oferta, nume_oferta, reducere, perioada)
10| values
11|(3, 'red ian', 40, '20-01-2023');
12| insert into Oferta(id_oferta, nume_oferta, reducere, perioada)
13| values
14|(4, 'red oct', 5, '03-10-2023');
15| insert into Oferta(id_oferta, nume_oferta, reducere, perioada)
16| values
17|(5, 'red iulie', 34, '23-07-2024');

```

ID_OFERTA	NUME_OFERTA	REDUCERE	PERIOADA
1	1 red iunie	12	12-07-2023
2	2 red mai	30	23-05-2023
3	5 red iulie	34	23-07-2024
4	3 red ian	40	20-01-2023
5	4 red oct	5	03-10-2023

Partener

```

insert into Partener(id_partener, nume_partener, locatie, telefon_partener)
values
(1, 'Ioana', 'Bucuresti', '0761223498');
insert into Partener(id_partener, nume_partener, locatie, telefon_partener)
values
(2, 'Mihai', 'Buftea', '0743221874');
insert into Partener(id_partener, nume_partener, locatie, telefon_partener)
values
(3, 'Ana', 'Sibiu', '0723442109');
insert into Partener(id_partener, nume_partener, locatie, telefon_partener)
values
(4, 'Alexandru', 'Iasi', '0712334097');
insert into Partener(id_partener, nume_partener, locatie, telefon_partener)
values
(5, 'Maria', 'Cluj', '0756443218');

```

script Output x Query Result x

SQL | All Rows Fetched: 5 in 0,002 seconds

ID_PARTENER	NUME_PARTENER	LOCATIE	TELEFON_PARTENER
1	1 Ioana	Bucuresti	0761223498
2	2 Mihai	Buftea	0743221874
3	3 Ana	Sibiu	0723442109
4	4 Alexandru	Iasi	0712334097
5	5 Maria	Cluj	0756443218

CLIENT

```

insert into Client(cnp_client, nume_client, telefon_client, adresa, mail, id_recenzie, id_plangere, id_plata)
values
(1209348745612, 'Stelu', '0763221231', 'Bucuresti', 'stelu@gmail.com', 1, 4, 3 );
insert into Client(cnp_client, nume_client, telefon_client, adresa, mail, id_recenzie, id_plangere, id_plata)
values
(1029384756748, 'Andu', '0723889712', 'Iasi', 'andu@gmail.com', 1, 4, 3 );
insert into Client(cnp_client, nume_client, telefon_client, adresa, mail, id_recenzie, id_plangere, id_plata)
values
(2039481074561, 'Ada', '0769987120', 'Ilfov', 'ada@gmail.com', 5, 3, 4 );
insert into Client(cnp_client, nume_client, telefon_client, adresa, mail, id_recenzie, id_plangere, id_plata)
values
(9801234875612, 'Mara', '0765443421', 'Constanta', 'mara@gmail.com', 3, 1, 2 );
insert into Client(cnp_client, nume_client, telefon_client, adresa, mail, id_recenzie, id_plangere, id_plata)
values
(9876554312906, 'Laur', '076889654', 'Bucuresti', 'laur@gmail.com', 2, 2, 1 );

```

```

insert into Client(cnp_client, nume_client, telefon_client, adresa, mail, id_recenzie, id_plangere, id_plata)
values
(1023984576123, 'Liviu', '0786551234', 'Cluj', 'liviu@gmail.com', 4, 10, 3);
insert into Client(cnp_client, nume_client, telefon_client, adresa, mail, id_recenzie, id_plangere, id_plata)
values
(9801237563812, 'Alex', '0789654312', 'Branesti', 'alex@gmail.com', 1, 9, 5);
insert into Client(cnp_client, nume_client, telefon_client, adresa, mail, id_recenzie, id_plangere, id_plata)
values
(5694312905634, 'Miruna', '0764334210', 'Iasi', 'miruna@gmail.com', 2, 7, 1);
insert into Client(cnp_client, nume_client, telefon_client, adresa, mail, id_recenzie, id_plangere, id_plata)
values
(9867451230987, 'Magda', '0769005431', 'Constanta', 'magda@gmail.com', 3, 8, 2);
insert into Client(cnp_client, nume_client, telefon_client, adresa, mail, id_recenzie, id_plangere, id_plata)
values
(3476512389765, 'Valentina', '0743221309', 'Bucuresti', 'valentina@gmail.com', 5, 6, 4);

```

	CNP_CLIENT	NUME_CLIENT	TELEFON_CLIENT	ADRESA	MAIL	ID_PLATA	ID_RECENZIE	ID_PLANGERE
1	1023984576123	Liviu	0786551234	Cluj	liviu@gmail.com	3	4	10
2	9801237563812	Alex	0789654312	Branesti	alex@gmail.com	5	1	9
3	5694312905634	Miruna	0764334210	Iasi	miruna@gmail.com	1	2	7
4	9867451230987	Magda	0769005431	Constanta	magda@gmail.com	2	3	8
5	3476512389765	Valentina	0743221309	Bucuresti	valentina@gmail.com	4	5	6
6	1209348745612	Stelu	0763221231	Bucuresti	stelu@gmail.com	3	1	4
7	1029384756748	Andu	0723889712	Iasi	andu@gmail.com	3	1	4
8	2039481074561	Ada	0769987120	Ilfov	ada@gmail.com	4	5	3
9	9801234875612	Mara	0765443421	Constanta	mara@gmail.com	2	3	1
10	9876554312906	Laur	076889654	Bucuresti	laur@gmail.com	1	2	2

Tip Eveniment

```

1 insert into tipeveniment(id_tipeveniment)
2 values
3 (1);
4 insert into tipeveniment(id_tipeveniment)
5 values
6 (2);
7 insert into tipeveniment(id_tipeveniment)
8 values
9 (3);
10 insert into tipeveniment(id_tipeveniment)
11 values
12 (4);
13 insert into tipeveniment(id_tipeveniment)
14 values
15 ...

```

```

UPDATE TipEveniment
SET numeTipEveniment = 'afaceri'
WHERE id_tipeveniment = 5;

```

int Output X | Query Result X

Script Output X | Query Result X

SQL | All Rows Fetched: 5 in 0,001 seconds

ID_TIPEVENIMENT	NUMETIPEVENIMENT
1	1 nunta
2	2 botez
3	3 intruniri
4	4 revedere
5	5 afaceri

Eveniment

```

| insert into Eveniment(id_eveniment, data_inceput, data_final, nume_eveniment, cnp_angajat, id_promovare, id_oferta, id_tipeveniment)
| values
| (1, '12-07-2023', '13-07-2023', 'nunta', 1234567891234, 1, 1, 1);
| insert into Eveniment(id_eveniment, data_inceput, data_final, nume_eveniment, cnp_angajat, id_promovare, id_oferta, id_tipeveniment)
| values
| (2, '20-01-2023', '20-01-2023', 'botez', 9876543212345, 2, 3, 2);
| insert into Eveniment(id_eveniment, data_inceput, data_final, nume_eveniment, cnp_angajat, id_promovare, id_oferta, id_tipeveniment)
| values
| (3, '23-05-2023', '23-05-2023', 'intrunire', 1230978542675, 3, 2, 3);
| insert into Eveniment(id_eveniment, data_inceput, data_final, nume_eveniment, cnp_angajat, id_promovare, id_oferta, id_tipeveniment)
| values
| (4, '03-10-2023', '04-10-2023', 'revedere', 6093451238745, 4, 4, 4);
| insert into Eveniment(id_eveniment, data_inceput, data_final, nume_eveniment, cnp_angajat, id_promovare, id_oferta, id_tipeveniment)
| values
| (5, '23-07-2024', '24-07-2024', 'afaceri', 6542138065237, 5, 5, 5);

| UPDATE eveniment
| SET nume_Eveniment = 'botez Alin'
| WHERE id_eveniment = 2;

| UPDATE eveniment
| SET nume_Eveniment = '10 ani de la terminarea liceului'
| WHERE id_eveniment = 4;

| UPDATE eveniment
| SET nume_Eveniment = 'sedinta firma'
| WHERE id_eveniment = 3;

| UPDATE eveniment
| SET nume_Eveniment = 'colaborare firme'
| WHERE id_eveniment = 5;

```

```

insert into Eveniment(id_eveniment, data_inceput, data_final, nume_eveniment, cnp_angajat, id_promovare, id_oferta, id_tipeveniment)
values
(6, '23-06-2024', '14-06-2024', 'nunta', 9870345213496, 6, 1, 1);
insert into Eveniment(id_eveniment, data_inceput, data_final, nume_eveniment, cnp_angajat, id_promovare, id_oferta, id_tipeveniment)
values
(7, '15-01-2023', '16-01-2023', 'botez', 5470923419742, 9, 3, 2);
insert into Eveniment(id_eveniment, data_inceput, data_final, nume_eveniment, cnp_angajat, id_promovare, id_oferta, id_tipeveniment)
values
(8, '03-05-2023', '03-05-2023', 'intrunire', 1256340986123, 8, 2, 3);
insert into Eveniment(id_eveniment, data_inceput, data_final, nume_eveniment, cnp_angajat, id_promovare, id_oferta, id_tipeveniment)
values
(9, '15-10-2023', '15-10-2023', 'revedere', 8976123407564, 7, 4, 4);
insert into Eveniment(id_eveniment, data_inceput, data_final, nume_eveniment, cnp_angajat, id_promovare, id_oferta, id_tipeveniment)
values
(10, '20-07-2024', '21-07-2024', 'afaceri', 1928374561029, 10, 5, 5);

```

```

UPDATE eveniment
SET nume_eveniment = 'nunta Alin si Mirela'
WHERE id_eveniment = 6;
UPDATE eveniment
SET nume_eveniment = 'botez Iris'
WHERE id_eveniment = 7;
UPDATE eveniment
SET nume_eveniment = 'discutie firma'
WHERE id_eveniment = 8;
UPDATE eveniment
SET nume_eveniment = 'revedere dupa 10 ani de la terminarea facultatii'
WHERE id_eveniment = 9;
UPDATE eveniment
SET nume_eveniment = 'afaceri firma'
WHERE id eveniment = 10;

```

SQL | All Rows fetched: 10 in 0 seconds

ID_EVENIMENT	DATA_INCEPUT	DATA_FINAL	NUME_EVENIMENT	CNP_ANGAJAT	ID_PROMOVARE	ID_OFERTA	ID_TIPEVENIMENT
1	23-06-2024	14-06-2024	nunta Alin si Mirela	9870345213496	6	1	
2	15-01-2023	16-01-2023	botez Iris	5470923419742	9	3	
3	03-05-2023	03-05-2023	discutie firma	1256340986123	8	2	
4	15-10-2023	15-10-2023	revedere dupa 10 ani de la terminarea facultatii	8976123407564	7	4	
5	20-07-2024	21-07-2024	afaceri firma	1928374561029	10	5	
6	12-07-2023	13-07-2023	Nunta Maria si Ion	1234567891234	1	1	
7	20-01-2023	20-01-2023	botez Alin	9876543212345	2	3	
8	23-05-2023	23-05-2023	sedinta firma	1230978542675	3	2	
9	03-10-2023	04-10-2023	10 ani de la terminarea liceului	6093451238745	4	4	
10	23-07-2024	24-07-2024	colaborare firme	6542138065237	5	5	

Locații Departament

```

select * from locatiidepartament;
create SEQUENCE SEQ_LOCATII start with 1 increment by 1 NOCYCLE NOCACHE;
insert into locatiidepartament
values (seq_locatii.nextval, 'Bucuresti');
insert into locatiidepartament
values (seq_locatii.nextval, 'Iasi');
insert into locatiidepartament
values (seq_locatii.nextval, 'Cluj');
insert into locatiidepartament
values (seq_locatii.nextval, 'Sibiu');
insert into locatiidepartament
values (seq_locatii.nextval, 'Oradea');

```

Script Output Query Result

SQL | All Rows Fetched: 5 in 0,003 seconds

ID_LOCATII	NUMELOCATII
1	1 Bucuresti
2	2 Iasi
3	3 Cluj
4	4 Sibiu
5	5 Oradea

12. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:

- subcereri sincronizate în care intervin cel puțin 3 tabele
- subcereri nesincronizate în clauza FROM
- grupări de date cu subcereri nesincronizate în care intervin cel puțin 3 tabele, funcții grup, filtrare la nivel de grupuri (în cadrul aceleiasi cereri)
- ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiasi cereri)
- utilizarea a cel puțin 2 funcții pe siruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE
- utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

```
--Pentru fiecare departament, să se afișeze numele acestuia, numele și salariul celor mai prost plătiți
--angajați din cadrul său.
--subcerere sincronizată
select nume_departament, nume_angajat, salariu
from departament d, (select id_departament, nume_angajat, salariu
                      from angajat a
                     where salariu = (select min(salariu)
                                       from angajat m
                                       where a.id_departament = m.id_departament
                                         group by m.id_departament)
                     group by id_departament, nume_angajat, salariu) sub
where d.id_departament= sub.id_departament;
```

Query Result x | Query Result 1 x

SQL | All Rows Fetched: 7 in 0,008 seconds

NUME_DEPARTAMENT	NUME_ANGAJAT	SALARIU
1 Management	Andrei	7640
2 Organizare Evenimente	Ionel	4100
3 Colaboratori	Ilinca	4600
4 Promovare	Mihai	7640
5 IT	Ion	4500
6 Contabilitate	Ana	2300
7 Curatenie	Ionela	2500

```
--sa se gaseasca tipul evenimentului dupa atributul din urma a evenimentelor la care lucreaza angajati
--cu salariul maxim de 3500
--grupări de date cu subcereri nesincronizate in care intervin cel putin 3 tabele,
--functii grup, filtrare la nivel de grupuri (in cadrul aceleiasi cereri)
select numetipeveniment, id_eveniment, cnp_angajat
  from eveniment join tipeveniment using (id_tipeveniment)
 where id_eveniment in (select id_eveniment
                           from eveniment
                          where cnp_angajat in (select cnp_angajat
                                                   from angajat|
                                                   where salariu in ( select max(salariu)
                                                       from angajat
                                                       group by cnp_angajat
                                                       having max(salariu) > 3500
                                                       )))
```

Query Result x

SQL | All Rows Fetched: 8 in 0,005 seconds

NUMETIPEVENIMENT	ID_EVENTIMENT	CNP_ANGAJAT
1 nuntă	6 9870345213496	
2 botez	7 5470923419742	
3 botez	2 9876543212345	
4 intruniri	8 1256340986123	
5 intruniri	3 1230978542675	
6 revedere	9 8976123407564	

```

1 --sa se afiseze departamentele si numele locatiilor din care salariul maxim al angajatilor este mai mare de 3500
2 --functii grup, filtrare la nivel de grupuri (in cadrul aceleiasi cereri)
3 select d.id_departament, l.numelocatii, d.nume_departament, max(a.salariu) as Salariu
4 from departament d join angajat a on (d.id_departament = a.id_departament)
5      join locatiidepartament l on (l.id_locatii = d.id_locatii)
6 group by d.id_departament, d.nume_departament, l.numelocatii
7 having max(salariu) > 3500;
8
9
10
11

```

Script Output | Query Result | SQL | All Rows Fetched: 5 in 0,005 seconds

	ID_DEPARTAMENT	NUMELOCATII	NUME_DEPARTAMENT	SALARIU
1	123	Bucuresti	Management	7640
2	127	Oradea	IT	4500
3	125	Cluj	Colaboratori	6500
4	126	Sibiu	Promovare	7640
5	124	Iasi	Organizare Evenimente	7000

```

--sa se afiseze reclamatia facuta de clientul cu cnp-ul 1023984576123
--subcerere nesincronizata pe 2 tabele in clauza from

select id_plangere, observatie_facuta
from plangere
WHERE id_plangere IN(select id_plangere
from client
where cnp_client IN(select cnp_client
from client
where cnp_client = 1023984576123));

```

Query Result | SQL | All Rows Fetched: 1 in 0,001 seconds

	ID_PLANGERE	OBSERVATIE_FACUTA
1	10	nu asculta toate parerile

```

--Pentru fiecare departament să se afișeze id-ul departamentului, suma totală a salariilor din acel departament
-- și într-o coloană cu numele STATUS SALARII să se afișeze un
-- mesaj care indică statusul salariilor:
-- pentru sumele mai mari de 2500 se va afișa mesajul: 'Suma angajatilor este mai mare de 2500'
-- pentru sumele între 1200 și 1400 se va afișa mesajul: 'Suma angajatilor este între 1200 și 2500'
-- iar pentru restul se va afișa mesajul: 'Suma angajatilor este mai mică de 1200'

-- s-au folosit: clauza WITH, group by, order si CASE

with salariu as(select id_departament, sum(salariu) SUMA
                  from angajat
                  group by id_departament
                  order by id_departament
)
select id_departament, SUMA,
CASE WHEN SUMA > 2500 THEN 'Suma angajatilor este mai mare de 2500'
WHEN SUMA < 2500 AND SUMA > 1200 THEN 'Suma angajatilor este între 1200 și 2500'
ELSE 'Suma angajatilor este mai mică de 1200'
END as "STATUS SALARII"
from salariu;

```

ID_DEPARTAMENT	SUMA	STATUS SALARII
1	123	7640 Suma angajatilor este mai mare de 2500
2	124	17100 Suma angajatilor este mai mare de 2500
3	125	11100 Suma angajatilor este mai mare de 2500
4	126	7640 Suma angajatilor este mai mare de 2500
5	127	4500 Suma angajatilor este mai mare de 2500
6	128	2300 Suma angajatilor este intre 1200 si 2500
7	129	2500 Suma angajatilor este mai mica de 1200

```
|--| Să se creeze o cerere prin care să se afișeze numărul total de evenimente și, din acest total, numărul evenimentelor din 2023.  
-- ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)  
select id_eveniment, count(id_eveniment), sum(decode(nvl(to_char(data_inceput, 'yyyy'), 0), '2023', 1, 0)) an2023  
from eveniment  
group by id_eveniment;
```

Query Result | All Rows Fetched: 10 in 0,003 seconds

ID_EVENIMENT	COUNT(ID_EVENIMENT)	AN2023
1	6	0
2	7	1
3	8	1
4	9	1
5	10	0
6	1	1
7	2	1
8	3	1
9	4	1
10	5	0

```
ksheet Query Builder

1 --Scriptă o cerere prin care să se afișeze prenumele salariatului cu prima literă majusculă și
2 --toate celelalte litere mici, numele acestuia cu majuscule și lungimea numelui, pentru
3 --angajații al căror nume începe cu A sau M. Rezultatul
4 --va fi ordonat descrescător după lungimea numelui. Se vor eticheta coloanele corespunzător.
5
6 --utilizarea a cel puțin 2 funcții pe șiruri de caractere
7
8 select initcap(nume_angajat) nume, length(nume_angajat) lungime
9 from angajat
10 where initcap(nume_angajat) like 'A%' or initcap(nume_angajat) like 'M%'
11 order by length(nume_angajat) desc;
```

Script Output x | Query Result x

SQL | All Rows Fetched: 5 in 0,002 seconds

	NUME	LUNGIME
1	Alexandru	9
2	Andreea	7
3	Andrei	6
4	Mihai	5
5	Ana	3

```

Sheet | Query Builder
--Să se afișeze numele evenimentului, data de început și
--data negocierii pretului, care este prima zi de Luni după 1 luna de la eveniment. Etichetați această coloană "Negociere".
--2 funcții pe date calendaristice
select nume_eveniment "NUME", data_inceput ,
       next_day(add_months(data_inceput, 1), 'LUNI') "Negociere"
from eveniment;

```

Script Output x | Query Result x

SQL | All Rows Fetched: 10 in 0,004 seconds

NUME	DATA_INCEPUT	Negociere
1 nunta Alin si Mirela	23-06-2024	29-07-2024
2 botez Iris	15-01-2023	20-02-2023
3 discutie firma	03-05-2023	05-06-2023
4 revedere dupa 10 ani de la terminarea facultatii	15-10-2023	20-11-2023
5 afaceri firma	20-07-2024	26-08-2024
6 Nunta Maria si Ion	12-07-2023	14-08-2023
7 botez Alin	20-01-2023	27-02-2023
8 sedinta firma	23-05-2023	26-06-2023
9 10 ani de la terminarea liceului	03-10-2023	06-11-2023
10 colaborare firme	23-07-2024	26-08-2024

```

7 ;
8 --sa se gaseasca tipul evenimentului dupa atributul din urma a evenimentelor la care lucreaza angajati
9 --cu cnp-ul 8976123407564
0 --grupări de date cu subcereri nesincronizate in care intervin cel putin 3 table,
1 select numetipeveniment, id_eveniment, cnp_angajat
2 from eveniment join tipeveniment using (id_tipeveniment)
3 where id_eveniment in (select id_eveniment
4                         from eveniment
5                           where cnp_angajat in (select cnp_angajat
6                                     from angajat
7                                       where cnp_angajat =8976123407564))
8
9
0
1

```

Query Result x | Query Result 1 x

SQL | All Rows Fetched: 1 in 0,002 seconds

NUMETIPEVENIMENT	ID_EVENTIMENT	CNP_ANGAJAT
1 revedere	9	8976123407564

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

```

4 --sa li se creasca salariile angajatilor cu 20% care lucreaza in departamentul 124
5
6
7 update angajat
8 set salariu = salariu + 0.2 * salariu
9 where cnp_angajat in (select cnp_angajat
0         from angajat
1             where id_departament = ( select id_departament
2                             from departament
3                                 where id_departament = 124)
4
5 select * from angajat;

```

Script Output x Query Result x

SQL | All Rows Fetched: 10 in 0,002 seconds

	CNP_ANGAJAT	NUME_ANGAJAT	SALARIU	FUNCTIE	VARSTA	ID_DEPARTAMENT
1	1928374561029	Ionela	2500	femeie de servici	30	129
2	8976123407564	Ionel	4920	organizator	27	124
3	1256340986123	Mihai	7640	promoveaza even	40	126
4	5470923419742	Silviu	6500	colaborator	30	125
5	9870345213496	Alexandru	8400	organizator	26	124
6	1234567891234	Ana	2300	contabil	20	128
7	9876543212345	Ion	4500	inginer	26	127
8	1230978542675	Andrei	7640	manager	40	123
9	6093451238745	Andreea	7200	organizator	50	124
10	6542138065237	Ilinca	4600	colaborator	26	125

	CNP_ANGAJAT	NUME_ANGAJAT	SALARIU	FUNCTIE	VARSTA	ID_DEPARTAMENT
1	1928374561029	Ionela	2500	femeie de servici	30	129
2	8976123407564	Ionel	4100	organizator	27	124
3	1256340986123	Mihai	7640	promoveaza even	40	126
4	5470923419742	Silviu	6500	colaborator	30	125
5	9870345213496	Alexandru	7000	organizator	26	124
6	1234567891234	Ana	2300	contabil	20	128
7	9876543212345	Ion	4500	inginer	26	127
8	1230978542675	Andrei	7640	manager	40	123
9	6093451238745	Andreea	6000	organizator	50	124
10	6542138065237	Ilinca	4600	colaborator	26	125

Înainte de cerere

```
-- sa se stearga ofertele care au reducere mai mica sau egala cu 12%. Sa se anuleze modificarile
delete from oferta
where reducere in (select reducere
                     from oferta
                     where reducere <= 12);
```

Script Output x | Query Result x

SQL | All Rows Fetched: 3 in 0,001 seconds

ID_OFERTA	NUME_OFERTA	REDUCERE	PERIOADA
1	2 red mai	30	23-05-2023
2	5 red iulie	34	23-07-2024
3	3 red ian	40	20-01-2023

3 | ROLLBACK;

Script Output x | Query Result x

SQL | All Rows Fetched: 5 in 0,001 seconds

ID_OFERTA	NUME_OFERTA	REDUCERE	PERIOADA
1	1 red iunie	12	12-07-2023
2	2 red mai	30	23-05-2023
3	5 red iulie	34	23-07-2024
4	3 red ian	40	20-01-2023
5	4 red oct	5	03-10-2023

```
--sa creasca suma_incasata cu 10% pentru statisticile cu id -ul recenziei 3
update statistica
set suma_incasata = suma_incasata + 0.1 * suma_incasata
where id_statistica in (select id_statistica
                           from statistica
                           where id_recenzie = ( select id_recenzie
                                                 from recenzie
                                                 where id_recenzie = 3)
                           );
select * from statistica;
rollback;
```

Script Output x | Query Result x | Query Result 1 x

SQL | All Rows Fetched: 10 in 0,001 seconds

ID_STATISTICA	SUMA_INCASATA	ID_RECENZIE
1	15	3300
2	16	2700
3	17	4300
4	18	6700
5	19	6000
6	10	2300
7	11	7000
8	12	3520
9	13	5412
10	14	7011

```

        );
select * from statistica;
rollback;

```

Script Output x | Query Result x | Query Result 1 x

SQL | All Rows Fetched: 10 in 0,001 seconds

ID_STATISTICA	SUMA_INCASATA	ID_RECENZIE
1	15	3000
2	16	2700
3	17	4300
4	18	6700
5	19	6000
6	10	2300
7	11	7000
8	12	3200
9	13	5412
10	14	7011

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

```

create view Oferta_Eveniment as select o.id_oferta, nume_oferta, nume_eveniment
from oferta o join eveniment e on o.id_oferta = e.id_oferta;

```

Script Output x | Query Result x | Query Result 1 x

SQL | All Rows Fetched: 10 in 0,002 seconds

ID_OFERTA	NUME_OFERTA	NUME EVENIMENT
1	1 red iunie	nunta Alin si Mirela
2	1 red iunie	Nunta Maria si Ion
3	2 red mai	discutie firma
4	2 red mai	sedinta firma
5	3 red ian	botez Iris
6	3 red ian	botez Alin
7	4 red oct	10 ani de la terminarea liceului
8	4 red oct	revedere dupa 10 ani de la terminarea facultatii
9	5 red iulie	colaborare firme
10	5 red iulie	afaceri firma

```

--lmd nepermisa
delete from Oferta_Eveniment where id_oferta = 3;
rollback;
--lmd permisa
update Oferta_Eveniment set nume_eveniment = '15 ani de la terminarea liceului' where id_oferta = 4;

```

15. Formulați în limbaj natural și 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

```
--division  
--afisati cat este jumata (1/2) din salariul angajatului cu cnp-ul 1928374561029  
SELECT salariu, salariu / 1.2 AS "jumatare din salariu"  
FROM angajat  
where cnp angajat = '1928374561029';
```

```
--selectati numele evenimentului, numele angajatului care are departamentul in locatia 'Iasi'  
--o cerere ce utilizeaza operatia outer-join pe minimum 4 tabele  
3 select nume_eveniment, nume_angajat, numelocatii  
from eveniment e FULL OUTER JOIN angajat a on (e.cnp_angajat = a.cnp_angajat)  
      FULL OUTER JOIN departament d on (d.id_departament = a.id_departament)  
      FULL OUTER JOIN locatiidepartament ld on (ld.id_locatii = d.id_locatii)  
where numelocatii like '%Iasi%';
```

NUME_EVENTIMENT	NUME_ANGAJAT	NUMELOCATII
1 revedere dupa 10 ani de la terminarea facultatii	Ionel	Iasi
2 afaceri firma	Ionela	Iasi
3 nunta Alin si Mirela	Alexandru	Iasi
4 10 ani de la terminarea liceului	Andreea	Iasi

```
--analiza top-n
-- În instrucțiunea SQL de mai jos, câmpurile obligatorii sunt afișate pentru angajații cu primele 3 cele mai mici salarii.
--Rezultatul este afișat în ordinea crescătoare a salariilor lor.

SELECT ROWNUM as RANG, nume_angajat, cnp_angajat, salarui
FROM (SELECT salarui, nume_angajat, cnp_angajat
      FROM Angajat
      ORDER BY salarui)
WHERE ROWNUM <= 3;
```

RANG	NUME_ANGAJAT	CNP_ANGAJAT	SALARIU
1	Ana	1234567891234	2300
2	Ionela	1928374561029	2500
3	Ionel	8976123407564	4100