**Proiect Restaurant Baze de Date**

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

Prin acest proiect îmi propun să creez o interfață complexă de gestiune pentru un restaurant, luând în considerare majoritatea aspectelor necesare unei astfel de întreprinderi la scară cât mai largă. Acesta propune o soluție pentru restaurantele de toate tipurile, fiind conceput atât pentru rezervări obișnuite, cât și pentru evenimente speciale, cum ar fi nunți, majorate, etc.

Este creată o infrastructură care să organizeze eficient întregul proces. Astfel, se pot observa mai multe secțiuni pe care diagrama le ia în considerare. Printre acestea există o tabelă de livrări, pentru care în supraentitatea “Angajați” există o subentitate special creată și anume “distribuitori\_livrări”. De asemenea, se asigură în permanență monitorizarea felurilor de mâncare disponibile într-un inventar. Se asigură o varietate constantă prin tabela care conține meniuri.

Lista de reduceri asigură popularizarea restaurantului prin actualizarea unei liste cu reduceri.

În cele din urmă, baza de date permite gestionarea eficientă a evenimentelor și se include scenariul în care sunt implicate persoane terțe. La nunți se poate nota colaborarea cu formații, muzicieni sau la petreceri de botez se poate lua în calcul prezența spre exemplu a unor ursitoare sau animatori.

**2.** Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.

Un client va putea mereu să facă o rezervare – presupunem că restaurantul are sufficient spațiu.

Un invitat poate participa la evenimente doar dacă este înregistrat în sistem ca invitat\_eveniment asociat unui client.

Planificarea unui eveniment presupune selectarea unei săli disponibile în data respectivă și alocarea cel puțin a unui angajat organizatoric.

O sală nu poate fi asociată simultan cu două evenimente care se desfășoară în aceeași zi.

Un animator nu poate fi planificat la două evenimente care se suprapun în timp.

Fiecare eveniment trebuie să aibă cel puțin un angajat alocat în tabelul Planificări\_organizatorice.

O ofertă expirată (data\_expirare < SYSDATE) nu poate fi asociată în tabelele Liste\_reduceri sau aplicată în comenzi.

Un angajat nu poate avea în același timp două funcții (ex: nu poate fi și distribuitor și restaurant), fiind în tabele specializate distincte.

Un invitat la eveniment nu poate fi repartizat la o masă deja completă (în funcție de capacitate definită la sală).

Fiecare produs este legat de un singur meniu, dar poate apărea în mai multe comenzi.

Un client poate fi atat mesean cat si invitat sau client online simultan.

| **Entitate** | **Cheie Primară** | **Descriere** |
| --- | --- | --- |
| **Angajați** | id\_angajat | Reprezintă angajații generali ai firmei, incluzând toate categoriile de personal. |
| **Distribuitori\_livrări** | id\_angajat\_distribuitor | Angajați responsabili cu livrările comenzilor către clienți. |
| **Angajați\_logistică** | id\_angajat\_logistica | Angajați implicați în logistica evenimentelor și a comenzilor. |
| **Personal\_restaurant** | id\_angajat\_restaurant | Angajați care deservesc în locația fizică, precum ospătari sau bucătari. |
| **Comenzi** | id\_comanda | Înregistrarea comenzilor efectuate de către clienți. |
| **Note\_plată** | id\_nota\_plata | Notele de plată care conțin totalul sumei comenzilor din restaurant. |
| **Rezervări** | id\_rezervare | Reprezintă rezervările făcute de clienți pentru anumite servicii sau evenimente. |
| **Clienți** | id\_client | Clienții care interacționează cu serviciile oferite. |
| **Meseni** | id\_mesean | Reprezintă persoanele care participă la mese, înregistrați ca parte a unui client. |
| **Invitați\_eveniment** | id\_invitat\_eveniment | Persoane invitate la evenimente, asociate unui client. |
| **Clienți\_online** | Id\_client\_online | Persoană care dorește să comande mâncare online. |
| **Oferte** | id\_oferta | Ofertele speciale ce pot conține reduceri pentru meniuri. |
| **Meniuri** | id\_meniu | Lista de meniuri disponibile pentru comenzi sau evenimente. |
| **Produse\_inventar** | id\_produs | Inventarul de produse folosite în meniuri. |
| **Evenimente** | id\_eveniment | Evenimentele organizate, ce pot implica săli, meniuri și animatori. |
| **Săli** | id\_sala | Spațiile fizice în care au loc evenimentele. |
| **Animatori** | id\_animator | Persoane (de exemplu, animatori) care participă în cadrul evenimentelor. |

**3.** Descrierea entităților, incluzând precizarea cheii primare.

**4.** Descrierea relațiilor, incluzând precizarea cardinalității acestora.

| **Relație** | **Cardinalitate** | **Descriere și Implementare** |
| --- | --- | --- |
| livrează | Distribuitori\_livrări - Comenzi: 1(1):M(0) | Un distribuitor poate livra mai multe comenzi (M). Fiecare comandă este livrată de un singur distribuitor (1). → FK în Comenzi. |
| preia | Angajați\_logistică - Comenzi: 1(1):M(0) | Un angajat logistică poate prelua mai multe comenzi (M). Fiecare comandă este preluată de un singur angajat logistică (1). → FK în Comenzi. |
| gestionează | Personal\_restaurant - Comenzi: 1(1):M(0) | Un angajat al restaurantului poate gestiona mai multe comenzi (M). Fiecare comandă este gestionată de un singur angajat al restaurantului (1). → FK în Comenzi. |
| conține | Comenzi – Produse\_inventar: M(0):M(1) | O comandă poate include mai multe produse, iar un produs poate fi inclus în mai multe comenzi. → tabel asociativ (Comenzi\_produse). |
| include | Meniuri - Produse\_inventar: 1(1):M(1) | Un meniu conține mai multe produse din inventar (M). Fiecare produs aparține unui singur meniu (1). → FK în Produse\_inventar. |
| comandă | Client\_online – Comenzi: 1(1):M(0) | Un client online commandă de mai multe ori (M). Fiecare comandă aparține unui singur client online (1). → FK în Comenzi. |
| generează | Meseni - Note\_plată: 1(1):M(0) | Un mesean poate genera mai multe note de plată (M). Fiecare notă de plată corespunde unui singur mesean (1). → FK în Note\_plată. |
| efectuează | Meseni - Rezervări: 1(1):M(0) | Un client poate efectua mai multe rezervări (M). Fiecare rezervare este efectuată de un singur client (1). → FK în Rezervări. |
| facturează | Note\_plată – Produse\_inventar: M(0):M(1) | O notă poate factura mai multe produse (M), iar un produs poate fi facturat în mai multe comenzi. → tabel asociativ (Detalii\_comandă). |
| participă | Invitați\_eveniment - Evenimente: M(1):M(1) | Un invitat poate participa la mai multe evenimente, iar un eveniment poate avea mai mulți invitați. → tabel asociativ (Participări\_evenimente). |
| sunt\_servite | Meniuri - Evenimente: M(1):M(0) | Un meniu poate fi servit în mai multe evenimente – M. Un eveniment poate include mai multe meniuri – M. → tabel asociativ (Meniuri\_evenimente) |
| planifică | Relație de tip 3 | Angajații, sălile și evenimentele sunt codependente, iar tabelul ajută la vizualizarea detaliilor de organizare și planificare. |
| participă\_eveniment | Animatori - Evenimente: M(0):M(0) | Un animator poate participa la mai multe evenimente, iar un eveniment poate avea mai mulți animatori. → tabel asociativ (Colaborări). |
| beneficiază | Produse\_inventar - Oferte: M(1):M(0) | Un produs poate beneficia de mai multe oferte, iar o ofertă poate cuprinde mai multe produse. → tabel asociativ (Liste\_reduceri). |
| este\_un | Cele 3 subentități de la Angajți și Clienți cu supraentitățile: 1(0):1(1) | Un angajat poate fi de 3 tipuri (distribuitor, logistician, personal -chelner/barman/bucatar) și clientul la fel (online, mesean – fizic, invitat) |

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

**1. Angajati**

| **ATRIBUT** | **TIP** | **CONSTRÂNGERI** | **VALORI POSIBILE ȘI VALORI DEFAULT** | **OBSERVAȚII** |
| --- | --- | --- | --- | --- |
| id\_angajat | NUMBER | PK | Generat automat | Identificator unic al angajatului |
| nume | VARCHAR2 | NOT NULL |  | Numele angajatului |
| prenume | VARCHAR2 | NOT NULL |  | Prenumele angajatului |
| functie | VARCHAR2 | NOT NULL |  | Funcția angajatului |
| salariu | NUMBER | NOT NULL |  | Salariul angajatului |
| dată\_angajare | DATE | NOT NULL | DD:MM:YY | Data angajării |
| mail | VARCHAR2 | NOT NULL |  | Mail-ul angajatului asociat |

**2. Distribuitori\_livrări**

| **ATRIBUT** | **TIP** | **CONSTRÂNGERI** | **VALORI POSIBILE ȘI VALORI DEFAULT** | **OBSERVAȚII** |
| --- | --- | --- | --- | --- |
| id\_angajat\_distribuitor | NUMBER | PK | Generat automat | Identificatorul distribuitorului |
| zona\_livrare | VARCHAR2 | NOT NULL |  | Zona de livrare |
| număr\_livrari | NUMBER | NOT NULL |  | Numărul de livrări effectuate |
| id\_angajat | NUMBER | FK → Angajați |  | Identificatorul angajatului asociat |

**3. Angajați\_logistică**

| **ATRIBUT** | **TIP** | **CONSTRÂNGERI** | **VALORI POSIBILE ȘI VALORI DEFAULT** | **OBSERVAȚII** |
| --- | --- | --- | --- | --- |
| id\_angajat\_logistica | NUMBER | PK | Generat automat | Identificatorul angajatului de logistică |
| nivel\_acces | VARCHAR2 | NOT NULL |  | Nivelul de acces în sistem |
| id\_angajat | NUMBER | FK → Angajați |  | Identificatorul angajatului asociat |

**4. Personal\_restaurant**

| **ATRIBUT** | **TIP** | **CONSTRÂNGERI** | **VALORI POSIBILE ȘI VALORI DEFAULT** | **OBSERVAȚII** |
| --- | --- | --- | --- | --- |
| id\_angajat\_restaurant | NUMBER | PK | Generat automat | Identificatorul angajatului restaurantului |
| rol | VARCHAR2 | NOT NULL |  | Rolul în restaurant |
| id\_angajat | NUMBER | FK → Angajați |  | Identificatorul angajatului asociat |

**5. Comenzi**

| **ATRIBUT** | **TIP** | **CONSTRÂNGERI** | **VALORI POSIBILE ȘI VALORI DEFAULT** | **OBSERVAȚII** |
| --- | --- | --- | --- | --- |
| id\_comanda | NUMBER | PK | Generat automat | Identificator unic al comenzii |
| data\_comanda | DATE | NOT NULL | SYSDATE | Data la care a fost înregistrată comanda |
| ora\_livrare | VARCHAR2 | NOT NULL | format HH:MM | Ora estimată pentru livrare |
| status | VARCHAR2 | NOT NULL | Plasată / În curs / Livrată | Starea actuală a comenzii |
| preț | NUMBER | NOT NULL |  | Pretul comenzii plasate |
| id\_distribuitor | NUMBER | FK → Distribuitori\_livrări |  | Angajat responsabil de livrare |
| id\_logistician | NUMBER | FK → Angajați\_logistică |  | Angajat care a preluat comanda |
| id\_personal\_restaurant | NUMBER | FK → Personal\_restaurant |  | Angajat care gestionează comanda |
| id\_client\_online | NUMBER | FK → Client\_online |  | Client care a comandat |

**6. Note\_plată**

| **ATRIBUT** | **TIP** | **CONSTRÂNGERI** | **VALORI POSIBILE ȘI VALORI DEFAULT** | **OBSERVAȚII** |
| --- | --- | --- | --- | --- |
| id\_notă\_plată | NUMBER | PK | Generat automat | Identificatorul notei de plată |
| valoare\_totală | NUMBER | NOT NULL |  | Valoarea totală a comenzii |
| metodă\_plată | VARCHAR2 | NOT NULL | Cash / Card | Metoda de plată aleasă de client |
| id\_mesean | NUMBER | FK → Meseni |  | Comanda corespunzătoare |

**7. Rezervări**

| **ATRIBUT** | **TIP** | **CONSTRÂNGERI** | **VALORI POSIBILE ȘI VALORI DEFAULT** | **OBSERVAȚII** |
| --- | --- | --- | --- | --- |
| id\_rezervare | NUMBER | PK | Generat automat | Identificatorul rezervării |
| data\_rezervare | DATE | NOT NULL |  | Data rezervării |
| numar\_persoane | NUMBER | NOT NULL |  | Numărul de persoane pentru rezervare |
| ora\_rezervare | VARCHAR2 | NOT NULL | hh:mm | Verificarea valabilității |
| id\_client | NUMBER | FK → Clienți |  | Clientul care a făcut rezervarea |

**8. Clienți**

| **ATRIBUT** | **TIP** | **CONSTRÂNGERI** | **VALORI POSIBILE ȘI VALORI DEFAULT** | **OBSERVAȚII** |
| --- | --- | --- | --- | --- |
| id\_client | NUMBER | PK | Generat automat | Identificator unic al clientului |
| nume | VARCHAR2 | NOT NULL |  | Numele clientului |
| prenume | VARCHAR2 | NOT NULL |  | Prenumele clientului |
| nr\_tel | VARCHAR2 | NOT NULL |  | Număr de telefon |
| mail | VARCHAR2 | NOT NULL |  | Adresa de email |

**9. Meseni**

| **ATRIBUT** | **TIP** | **CONSTRÂNGERI** | **VALORI POSIBILE ȘI VALORI DEFAULT** | **OBSERVAȚII** |
| --- | --- | --- | --- | --- |
| id\_mesean | NUMBER | PK | Generat automat | Identificator unic al meseanului |
| loc\_masa | VARCHAR2 | NOT NULL |  | Locul la masă |
| id\_client | NUMBER | FK → Clienți |  | Identificatorul clientului asociat |

**10. Invitați\_eveniment**

| **ATRIBUT** | **TIP** | **CONSTRÂNGERI** | **VALORI POSIBILE ȘI VALORI DEFAULT** | **OBSERVAȚII** |
| --- | --- | --- | --- | --- |
| id\_invitat\_eveniment | NUMBER | PK | Generat automat | Identificator unic al invitatului de eveniment |
| tip\_invitat | VARCHAR2 | NOT NULL | VIP, normal | Tipul invitației |
| masa\_invitat | NUMBER | NOT NULL |  | Masa unde e repartizat |
| id\_client | NUMBER | FK → Clienți |  | Identificatorul clientului invitat |

**11. Clienți\_online**

| **ATRIBUT** | **TIP** | **CONSTRÂNGERI** | **VALORI POSIBILE ȘI VALORI DEFAULT** | **OBSERVAȚII** |
| --- | --- | --- | --- | --- |
| id\_client\_online | NUMBER | PK | Generat automat | Identificator unic al meseanului |
| adresă | VARCHAR2 | NOT NULL |  | Adresa comenzii |
| id\_client | NUMBER | FK → Clienți |  | Identificatorul clientului asociat |

| **ATRIBUT** | **TIP** | **CONSTRÂNGERI** | **VALORI POSIBILE ȘI VALORI DEFAULT** | **OBSERVAȚII** |
| --- | --- | --- | --- | --- |
| id\_meniu | NUMBER | PK | Generat automat | Identificator unic al meniului |
| Denumire | VARCHAR2 | NOT NULL |  | Denumirea meniului |
| Descriere | VARCHAR2 | NOT NULL |  | Descriere scurtă a meniului |
| categorie\_produse | VARCHAR2 | NOT NULL |  | Categoria de produse |

**12. Meniuri**

**13. Produse\_inventar**

| **ATRIBUT** | **TIP** | **CONSTRÂNGERI** | **VALORI POSIBILE ȘI VALORI DEFAULT** | **OBSERVAȚII** |
| --- | --- | --- | --- | --- |
| id\_produs | NUMBER | PK | Generat automat | Identificatorul produsului |
| nume\_produs | VARCHAR2 | NOT NULL |  | Numele produsului |
| cantitate | NUMBER | NOT NULL |  | Cantitatea disponibilă |
| unitate\_masura | VARCHAR2 | NOT NULL | buc / portie / sticla / pahar | Unitatea de măsură pentru cantitate |
| pret\_unitar | NUMBER | NOT NULL |  | Pretul unei portii |
| id\_meniu | NUMBER | FK → Produse\_inventar |  | Identificatorul meniului din care face parte |

**14. Evenimente**

| **ATRIBUT** | **TIP** | **CONSTRÂNGERI** | **VALORI POSIBILE ȘI VALORI DEFAULT** | **OBSERVAȚII** |
| --- | --- | --- | --- | --- |
| id\_eveniment | NUMBER | PK | Generat automat | Identificator unic al evenimentului |
| tip\_eveniment | VARCHAR2 | NOT NULL |  | Tipul evenimentului (ex: nuntă, petrecere) |
| data\_eveniment | DATE | NOT NULL |  | Data la care are loc evenimentul |
| numar\_persoane | NUMBER | NOT NULL |  | Număr estimat de participanți |

**15. Oferte**

| **ATRIBUT** | **TIP** | **CONSTRÂNGERI** | **VALORI POSIBILE ȘI VALORI DEFAULT** | **OBSERVAȚII** |
| --- | --- | --- | --- | --- |
| id\_oferta | NUMBER | PK | Generat automat | Identificator unic al ofertei |
| discount | NUMBER | NOT NULL |  | Reducerea aplicată comenzii |
| data\_expirare | DATE | NOT NULL |  | Data expirării ofertei |

**16. Săli**

| **ATRIBUT** | **TIP** | **CONSTRÂNGERI** | **VALORI POSIBILE ȘI VALORI DEFAULT** | **OBSERVAȚII** |
| --- | --- | --- | --- | --- |
| id\_sala | NUMBER | PK | Generat automat | Identificatorul sălii |
| nume\_sala | VARCHAR2 | NOT NULL |  | Numele sălii |
| capacitate | NUMBER | NOT NULL |  | Numărul maxim de persoane |

**17. Animatori**

| **ATRIBUT** | **TIP** | **CONSTRÂNGERI** | **VALORI POSIBILE ȘI VALORI DEFAULT** | **OBSERVAȚII** |
| --- | --- | --- | --- | --- |
| id\_animator | NUMBER | PK | Generat automat | Identificatorul animatorului |
| nume\_animator | VARCHAR2 | NOT NULL |  | Numele animatorului |
| tip\_activitate | VARCHAR2 | NOT NULL |  | Serviciile prestate |
| pret\_ora | NUMBER | NOT NULL |  | Pretul serviciilor prestate |

**18. Planificări\_organizatorice – relația de tip 3**

| **Atribut** | **Tip** | **Constrângeri** | **VALORI POSIBILE ȘI VALORI DEFAULT** | **OBSERVAȚII** |
| --- | --- | --- | --- | --- |
| id\_angajat | NUMBER | FK → Angajati + PK (face parte din compunere) |  | Identificatorul angajatului |
| id\_eveniment | NUMBER | FK → Evenimente + PK (face parte din compunere) |  | Identificatorul evenimentului |
| id\_sala | NUMBER | FK → Sali + PK (face parte din compunere) |  | Identificatorul sălii |
| observații | VARCHAR2 | Opțional |  | Specificarea detaliilor cerute de clienți |

Tabelele asociative contin doar ID-urile tabelelor pe care le leagă, având cheie primară compusă și cele două coloane sunt FK (exact ca în diagrama conceptuală).

**6.** Realizarea diagramei entitate-relațiecorespunzătoare descrierii de la punctele 3-5.

1

Planificări\_organizatorice

Angajați

1

M

Meniuri

Săli

Produse\_inventar

Meseni

Clienți

Invitați\_eveniment

Distribuitori\_livrări

1

Angajați\_logistică

Personal\_restaurant

M

1 to 1 Angajați

1

M

M

Evenimente

M

Comenzi\_online\_online

M

Rezervări

M

M

M

M

Oferte

M

Animatori

Comenzi\_restaurant

M

M

M

M

M

1

M

1

1 to 1 Clienți

1

M

M

M

1 to 1 Clienți

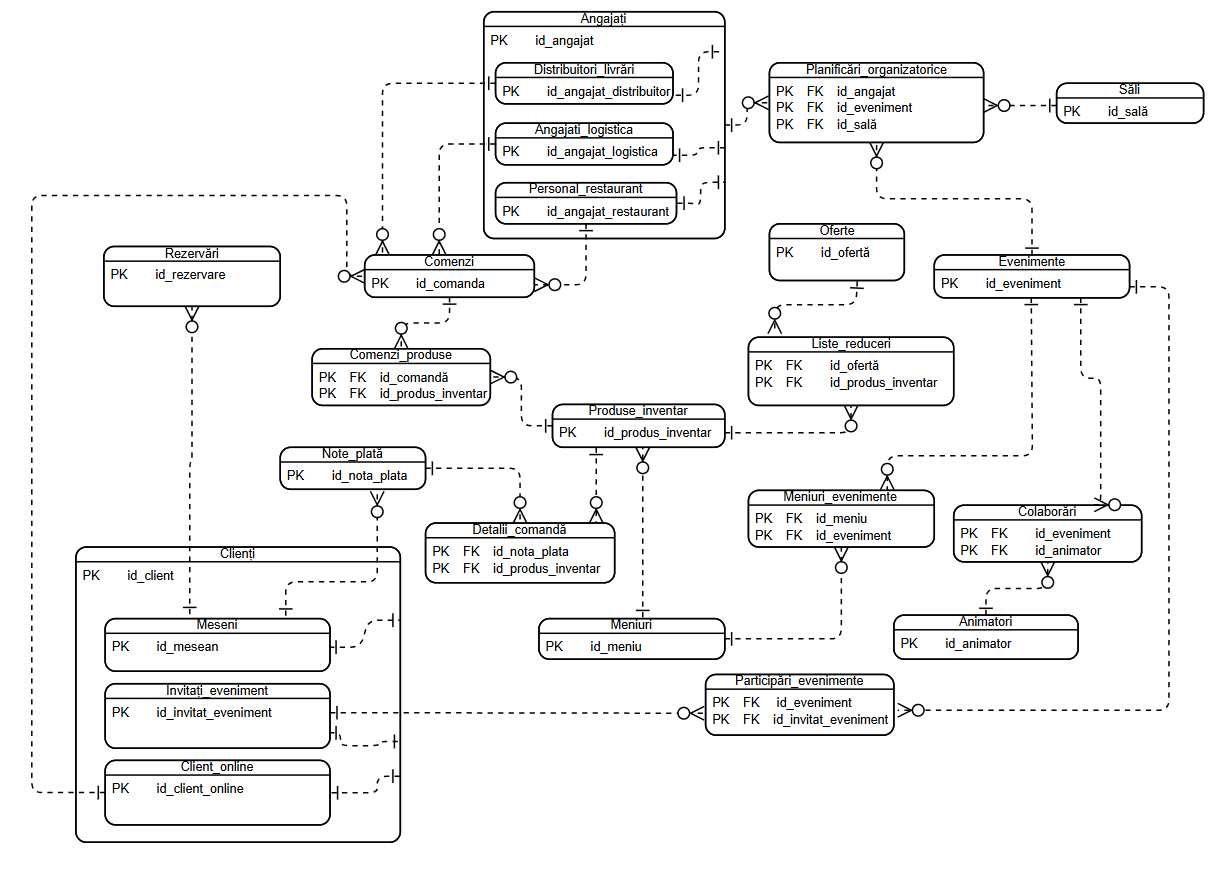
M

1 to 1 Clienți

1

Client\_online

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



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

**ANGAJAȚI**(#id\_angajat, nume, prenume, functie, salariu, data\_angajare, mail)

**DISTRIBUITORI\_LIVRĂRI**(#id\_distribuitor, zona\_livrare, numar\_livrari, id\_angajat)

**ANGAJATI\_LOGISTICĂ**(#id\_angajat\_logistica, nivel\_acces, id\_angajat)

**PERSONAL\_RESTAURANT**(#id\_personal\_restaurant, rol, id\_angajat)

**CLIENȚI**(#id\_client, nume, prenume, nr\_tel, mail)

**MESENI**(#id\_mesean, loc\_masa, id\_client)

**INVITAȚI\_EVENIMENT**(#id\_invitat\_eveniment, tip\_invitat, masa\_invitat, id\_client)

**CLIENȚI\_ONLINE**(#id\_client\_online, adresa, id\_client)

**COMENZI**(#id\_comanda, data\_comanda, ora\_livrare, status, pret, id\_distribuitor, id\_logistician, id\_personal\_restaurant, id\_client\_online)

**NOTE\_PLATĂ**(#id\_nota\_plata, valoare\_totala, metoda\_plata, id\_mesean)

**REZERVĂRI**(#id\_rezervare, data\_rezervare, ora\_rezervare, numar\_persoane, id\_mesean)

**MENIURI**(#id\_meniu, denumire, descriere, categorie\_produse)

**PRODUSE\_INVENTAR**(#id\_produs\_inventar, nume\_produs, cantitate, unitate\_masura, pret\_unitar, id\_meniu)

**EVENIMENTE**(#id\_eveniment, tip\_eveniment, data\_eveniment, numar\_persoane)

**OFERTE**(#id\_oferta, discount, data\_expirare)

**SĂLI**(#id\_sala, nume\_sala, capacitate)

**ANIMATORI**(#id\_animator, nume\_animator, tip\_activitate, pret\_ora)

**COMENZI\_PRODUSE**(#id\_comanda, #id\_produs\_inventar)

**DETALII\_COMANDĂ**(#id\_nota\_plata, #id\_produs\_inventar)

**PARTICIPĂRI\_EVENIMENTE**(#id\_eveniment, #id\_invitat\_eveniment)

**MENIURI\_EVENIMENTE**(#id\_meniu, #id\_eveniment)

**COLABORĂRI**(#id\_eveniment, #id\_animator)

**LISTE\_REDUCERI**(#id\_oferta, #id\_produs\_inventar)

**PLANIFICĂRI\_ORGANIZATORICE**(#id\_angajat, #id\_eveniment, #id\_sala, observatii)

**9.** Realizarea normalizării până la forma normal 3 (FN1 – FN3)

**Exemplul 1 – Normalizarea tabelei Note\_plata**

Presupunem că inițial tabela Note\_plata conținea, pe lângă informațiile despre plată, și denumiri de produse, astfel:

| **id\_nota\_plata** | **valoare\_totala** | **metoda\_plata** | **Produse** |
| --- | --- | --- | --- |
| 201 | 55.00 | Card | Pizza, Cola |

Această structură este într-o formă pre-FN1. Produsele sunt incluse în mod neatomic – două produse într-un singur rând, ceea ce contravine formei normale 1 (FN1).

Pentru a aduce tabela în FN2, am creat o tabelă asociativă Detalii\_comandă, unde fiecare produs este reprezentat pe un rând propriu.

| **id\_nota\_plata** | **valoare\_totala** | **metoda\_plata** |
| --- | --- | --- |
| 201 | 55.00 | Card |

| **id\_nota\_plata** | **produs** |
| --- | --- |
| 201 | Pizza |
| 201 | Cola |

**Exemplul 3 – Normalizarea tabelei Comenzi\_produse**

| **id\_comanda** | **id\_produs\_inventar** | **pret\_unitar** |
| --- | --- | --- |
| 101 | 5 | 15.50 |
| 101 | 7 | 21.00 |
| 102 | 5 | 15.50 |

Nu se află în FN-2 pentru că pret\_unitar nu depinde de id\_comanda, ci doar de produs. Așadar separăm prețul împreună cu produsul într-un table dedicate produsului. Am adăugat și alte attribute specifice care apar actualmente în table.

| **id\_comanda** | **id\_produs\_inventar** |
| --- | --- |
| 101 | 5 |
| 101 | 7 |
| 102 | 5 |

| **id\_produs\_inventar** | **nume\_produs** | **cantitate** | **unitate\_masura** | **pret\_unitar** |
| --- | --- | --- | --- | --- |
| 5 | Pizza | 100 | buc | 15.50 |
| 7 | Cola | 200 | sticla | 21.00 |

**Exemplul 3 – Normalizarea tabelei Rezervări**

Presupunem că inițial tabela Rezervări conținea, pe lângă informațiile despre rezervare, și datele clientului:

| **id\_rezervare** | **data\_rezervare** | **nr\_persoane** | **nume\_client** | **nr\_tel** |
| --- | --- | --- | --- | --- |
| 301 | 2024-06-10 | 4 | Ionescu Maria | 0723 456 789 |
| 302 | 2024-06-11 | 6 | Ionescu Maria | 0723 456 789 |

Această structură este într-o formă pre-FN3. Atributele precum nume\_client și nr\_tel nu depind de cheia primară id\_rezervare, ci de o entitate externă – clientul. Nu există coloane compuse sau liste, deci forma FN1 este respectată. De asemenea, cheia este simplă, deci FN2 este implicit satisfăcută. Problema constă în dependențele tranzitive.

Pentru a aduce tabela în FN3, am extras datele clientului într-o tabelă separată (Clienți) și am păstrat în Rezervări doar id\_client, adică cheia externă care leagă rezervarea de clientul respectiv.

În forma finală FN3, fiecare atribut depinde în mod direct de cheia primară a tabelei sale și nu există dependențe tranzitive.

* Rezervări (id\_rezervare, data\_rezervare, nr\_persoane, id\_client)
* Clienți (id\_client, nume\_client, nr\_tel) – supraentitate ce conține Meseni

**Exemplul 4 – Normalizarea tabelei Produse\_inventar**

Presupunem că inițial tabela Produse\_inventar conținea, pe lângă datele despre produse, și denumirea meniului din care face parte fiecare produs:

| **id\_produs** | **nume\_produs** | **cantitate** | **pret** | **denumire\_meniu** |
| --- | --- | --- | --- | --- |
| 501 | Pizza | 15 | 25.00 | Meniu Italian |
| 502 | Tiramisu | 8 | 14.00 | Meniu Italian |

Această structură este într-o formă pre-FN3. Atributul denumire\_meniu nu depinde de cheia primară id\_produs, ci de o entitate externă – meniul. Nu există coloane compuse sau liste, deci forma FN1 este respectată. De asemenea, cheia este simplă, deci FN2 este implicit satisfăcută. Problema constă în dependențele tranzitive.

Pentru a aduce tabela în FN3, am extras informațiile despre meniuri într-o tabelă separată (Meniuri) și am păstrat în Produse\_inventar doar id\_meniu, adică cheia externă care leagă produsul de meniul din care face parte.

În forma finală FN3, fiecare atribut depinde în mod direct de cheia primară a tabelei sale și nu există dependențe tranzitive. Exemplu:

* Produse\_inventar (id\_produs, nume\_produs, cantitate, pret, id\_meniu)
* Meniuri (id\_meniu, meniu\_denumire, descriere)

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

Vezi creare\_insert\_complet.txt

**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; maxim 30 de înregistrări în fiecare tabel).

Vezi creare\_insert\_complet.txt

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

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

Vezi queries.txt

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

Vezi queries.txt

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

Vezi queries.txt

**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.

Vezi queries.txt

**16.** Prezentarea planului de execuție a unei cereri complexe, optimizare/compare plan alternativ folosind hint-uri și obiecte specifice optimizării cererilor (spre exemplu indexi).

Vezi queries.txt pentru cod. Am folosit primul query de la exercițiul 12, cu subcereri sincronizate.

Explicare optimizre:

Tabela comenzi este accesată de 3 ori, pentru join în query-ul mare și în cele două subcereri sincronizate, care prelucrează datele câte un rând o dată, ceea ce creează un blocaj.

De asemenea, se observă accesarea frecventă a tabelei comenzi. Aceasta ar avea într-un model real mult mai multe intrări, așa că un index ar optimiza accesarea acesora și gruparea și agregarea lor.

De aceea, am optimizat interogarea prin:

1. Rescrierea logicii de agregare (eliminarea subcererilor corelate) folosind un JOIN direct și un GROUP BY global.
2. Crearea unui index pe coloana ID\_CLIENT\_ONLINE din tabela COMENZI, care este utilizată atât în JOIN, cât și pentru agregări. Acest index permite înlocuirea TABLE ACCESS FULL cu INDEX RANGE SCAN, scăzând semnificativ costul execuției.

**17. a.** Realizarea normalizării BCNF, FN4, FN5.

**Exemplu BCNF – Colaborări**

|  |  |  |  |
| --- | --- | --- | --- |
| id\_eveniment | id\_animator | tip\_eveniment | tip\_activitate |
| 1 | 1 | Nuntă | Muzică Live |
| 2 | 2 | Botez | Clown |
| 3 | 1 | Aniversare | Muzică Live |
| 4 | 3 | Majorat | DJ |

Verificăm dependențele tranzitive ale atributelor non-cheie. Avem: id\_animator → tip\_activitate.

Aici, id\_animator este o parte a cheii primare (id\_eveniment, id\_animator). tip\_activitate este un atribut non-cheie.

Dependența tranzitivă s-ar manifesta dacă am avea PK → NonKey1 → NonKey2. În cazul nostru, (id\_eveniment, id\_animator) → tip\_activitate, și id\_animator → tip\_activitate.

Conform definiției FN3, un tabel este în FN3 dacă nu există atribute non-cheie care depind tranzitiv de cheia primară. Asta înseamnă că niciun atribut non-cheie nu depinde de un alt atribut non-cheie.

În exemplul nostru, tip\_activitate depinde de id\_animator. Deși id\_animator este parte a cheii primare, tip\_activitate nu depinde de un alt atribut non-cheie. Astfel, tabelul este în FN3.

Regula BCNF: Fiecare determinant trebuie să fie o cheie candidat.

Avem dependența: id\_animator → tip\_activitate.

Aici, id\_animator este determinantul.

Este id\_animator o cheie candidat? Nu. Un singur animator (id\_animator = 1) poate participa la mai multe evenimente (1, 3), deci id\_animator singur nu poate identifica un rând unic în tabelul Colaborari.

Deoarece determinantul id\_animator nu este o cheie candidat, tabelul nu este în BCNF.

Pentru a aduce tabelul în BCNF, trebuie să eliminăm dependența id\_animator → tip\_activitate prin descompunerea tabelului în alte două tabele (Animatori și colaborări elimină tipul de activitate care se duce la Animatori):

|  |  |
| --- | --- |
| **id\_animator** | **tip\_activitate** |
| 1 | Muzică Live |
| 2 | Clown |
| 3 | DJ |
| 4 | Iluzionism |

|  |  |
| --- | --- |
| **id\_eveniment** | **id\_animator** |
| 1 | 1 |
| 2 | 2 |
| 3 | 1 |
| 4 | 3 |

**Exemplu FN4 – Furnizori**

| **id\_meniu** | **id\_produs\_inventar** | **nume\_furnizor\_ingredient** |
| --- | --- | --- |
| 1 | 1 (Făină) | AgroPro SRL |
| 1 | 1 (Făină) | Grădinari SRL |
| 1 | 2 (Ouă) | Ferma Găinilor |
| 2 | 3 (Carne) | Carnex SA |
| 2 | 4 (Legume) | Verde Bio |
| 2 | 4 (Legume) | Grădinari SRL |

Presupunem că vrem să adăugăm o tabelă pentru furnizorii de ingrediente.

Tabelul nu este în FN4 deoarece are dependențe de tip multivaloare netriviale care nu sunt dependențe funcționale. Nu este suficient să știm id\_meniu și id\_produs\_inventar pentru a determina un singur nume\_furnizor\_ingredient, și invers, id\_meniu și nume\_furnizor\_ingredient nu determină un singur id\_produs\_inventar.

Faptul că meniul 1 folosește făină nu implică faptul că făina este furnizată de toți furnizorii de făină pentru acest meniu. Informația despre furnizor este o proprietate a ingredientului, nu a relației dintre meniu și ingredient.

Pentru a aduce tabelul în FN4, trebuie să eliminăm dependențele multivaloare prin descompunerea sa în tabele mai mici:

| **id\_meniu** | **id\_produs\_inventar** |
| --- | --- |
| 1 | 1 (Făină) |
| 1 | 2 (Ouă) |
| 2 | 3 (Carne) |
| 2 | 4 (Legume) |

| **id\_produs\_inventar** | **nume\_furnizor\_ingredient** |
| --- | --- |
| 1 (Făină) | AgroPro SRL |
| 1 (Făină) | Grădinari SRL |
| 2 (Ouă) | Ferma Găinilor |
| 3 (Carne) | Carnex SA |
| 4 (Legume) | Verde Bio |
| 4 (Legume) | Grădinari SRL |

**Exemplu FN5 – Planificări organizatorice**

| **id\_angajat** | **id\_eveniment** | **id\_sala** |
| --- | --- | --- |
| 201 | 10 | 1 |
| 202 | 11 | 2 |

Este correct în forma FN-5. Dacă spargem în:

* Angajat\_Eveniment(id\_angajat, id\_eveniment)
* Eveniment\_Sala(id\_eveniment, id\_sala)
* Angajat\_Sala(id\_angajat, id\_sala)

și apoi facem JOIN, obținem linii care nu existau precum:

| **id\_angajat** | **id\_eveniment** | **id\_sala** |
| --- | --- | --- |
| 201 | 10 | 1 |
| 201 | 10 | 2 |

**b.** Aplicarea denormalizării, justificând necesitatea acesteia.

Se propune denormalizarea tabelelor Comenzi și Produse\_inventar.

De ce merită denormalizată această parte?

1. Frecvența Interogărilor: În cazul unui restaurant, operațiile de vizualizare a detaliilor comenzilor (pentru angajații din restaurant, bucătărie, distribuitori, sau pentru generarea de facturi/bonuri) sunt extrem de frecvente.
2. Complexitatea Interogărilor: Fiind cele mai utilizate date, complexitatea interogărilor care le accesează crește
3. Volumul de date: Într-un restaurant cele mai multe date reținute vor fi legate de consumabile – produse și comenzi

CREATE MATERIALIZED VIEW MV\_DETALII\_COMENZI\_PRODUSE

REFRESH COMPLETE ON DEMAND

AS

SELECT

    -- Coloane din tabela COMENZI

    C.id\_comanda AS comanda\_id,

    C.data\_comanda AS comanda\_data,

    C.ora\_livrare AS comanda\_ora\_livrare,

    C.status AS comanda\_status,

    C.pret AS comanda\_pret\_total,

    C.id\_distribuitor AS comanda\_distribuitor,

    C.id\_logistician AS comanda\_logistician,

    C.id\_personal\_restaurant AS comanda\_personal\_restaurant,

    C.id\_client\_online AS comanda\_id\_client\_online,

    -- Coloane din tabela COMENZI\_PRODUSE

    -- CP.id\_comanda AS cp\_comanda\_id,

    -- CP.id\_produs\_inventar AS cp\_produs\_inventar\_id,

    -- Coloane din tabela PRODUSE\_INVENTAR

    PI.id\_produs\_inventar AS produs\_inventar\_id,

    PI.nume\_produs AS produs\_inventar\_nume,

    PI.cantitate AS produs\_inventar\_cantitate\_stoc,

    PI.unitate\_masura AS produs\_inventar\_unitate\_masura,

    PI.pret\_unitar AS produs\_inventar\_pret\_unitar,

    PI.id\_meniu AS produs\_inventar\_id\_meniu\_asociat

FROM

    COMENZI C

JOIN

    COMENZI\_PRODUSE CP ON C.id\_comanda = CP.id\_comanda

JOIN

    PRODUSE\_INVENTAR PI ON CP.id\_produs\_inventar = PI.id\_produs\_inventar;

**18.** Exemplificarea ***isolation levels*** prin exemple de tranzacții care se execută în paralel în condiții de concurență, evidențiind efectele diferitelor niveluri de izolare asupra concurenței și integrității datelor.

Nivele de Izolare SQL Standard:

1. READ UNCOMMITTED
2. READ COMMITTED
3. REPEATABLE READ
4. SERIALIZABLE

Vom folosi un produs ipotetic din inventar cu id\_produs\_inventar = 101 și cantitate\_disponibila = 50.

Stare inițială:

UPDATE PRODUSE\_INVENTAR SET CANTITATE\_DISPONIBILA = 50   
WHERE ID\_PRODUS\_INVENTAR = 101;

COMMIT;

**Permite:** Dirty Reads, Non-Repeatable Reads, Phantom Reads, Lost Updates

**1. READ UNCOMMITTED**

**Scenariu: Dirty Read**

**Tranzacția 1 (T1): Actualizează Stocul (fără Commit)**

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;

BEGIN TRANSACTION;

UPDATE PRODUSE\_INVENTAR

SET CANTITATE\_DISPONIBILA = CANTITATE\_DISPONIBILA - 10

WHERE ID\_PRODUS\_INVENTAR = 101;

-- Nu se face COMMIT încă

**Tranzacția 2 (T2): Verifică Stocul**

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;

BEGIN TRANSACTION;

-- Un alt angajat verifică stocul disponibil

SELECT CANTITATE\_DISPONIBILA FROM PRODUSE\_INVENTAR WHERE ID\_PRODUS\_INVENTAR = 101;

-- Va returna 40 (date necommited de T1)

COMMIT;

-- Dintr-un motiv oarecare, comanda T1 este anulată

ROLLBACK;

**Efect:** T2 a citit o valoare (40) care ulterior a fost anulată (stocul a revenit la 50).

**2. READ COMMITTED (nivelul implicit în oracle)**

**Scenariu: Non-Repeatable Read**

**Stare inițială:** CANTITATE\_DISPONIBILA = 50 pentru ID\_PRODUS\_INVENTAR = 101.

**Evită:** Dirty Reads

**Permite:** Non-Repeatable Reads, Phantom Reads, Lost Updates

**Tranzacția 1 (T1): Procesează Comanda și Actualizează Stocul**

SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

BEGIN TRANSACTION;

SELECT CANTITATE\_DISPONIBILA FROM PRODUSE\_INVENTAR WHERE ID\_PRODUS\_INVENTAR = 101;

-- Returnează 50

-- Simulăm o operație complexă care durează

**Tranzacția 2 (T2): Procesează Altă Comandă și Actualizează Stocul**

SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

BEGIN TRANSACTION;

-- Altă comandă reduce stocul

UPDATE PRODUSE\_INVENTAR

SET CANTITATE\_DISPONIBILA = CANTITATE\_DISPONIBILA - 5

WHERE ID\_PRODUS\_INVENTAR = 101;

COMMIT;

-- Stocul este acum 45

**Revenire Tranzacția 1:**

SQL

-- T1 își reia execuția

SELECT CANTITATE\_DISPONIBILA FROM PRODUSE\_INVENTAR WHERE ID\_PRODUS\_INVENTAR = 101;

-- Returnează 45. T1 a citit de două ori, și a primit valori diferite.

COMMIT;

**Efect:** T1 a văzut stocul ca fiind 50 inițial, dar când a verificat din nou, a găsit 45.

**3. REPEATABLE READ**

**Scenariu: Phantom Read**

**Stare inițială:** Avem produse în inventar. Să zicem că avem produse cu CANTITATE\_DISPONIBILA > 0.

**Evită:** Dirty Reads, Non-Repeatable Reads

**Permite:** Phantom Reads, Lost Updates

**Tranzacția 1 (T1): Verifică Produse Disponibile**

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;

BEGIN TRANSACTION;

SELECT COUNT(\*) FROM PRODUSE\_INVENTAR WHERE CANTITATE\_DISPONIBILA > 0;

-- Returnează, de exemplu, 10

-- Continua executia cu alta logica

**Tranzacția 2 (T2): Adaugă un Produs Nou în Inventar**

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;

BEGIN TRANSACTION;

INSERT INTO PRODUSE\_INVENTAR (ID\_PRODUS\_INVENTAR, NUME\_PRODUS, PRET\_UNITAR, CANTITATE\_DISPONIBILA, ID\_MENIU)

VALUES (102, 'Salată de fructe', 25.00, 30, 5); -- Un produs nou, cu cantitate > 0

COMMIT;

**Revenire Tranzacția 1:**

-- T1 ajunge în cele din urmă la linia

SELECT COUNT(\*) FROM PRODUSE\_INVENTAR WHERE CANTITATE\_DISPONIBILA > 0;

-- Returnează 11. T1 a citit un "rând fantomă" (produsul nou inserat de T2).

COMMIT;

**Efect:** Deși T1 a blocat rândurile pe care le-a citit inițial pentru a preveni modificări (Non-Repeatable Reads), nu a prevenit inserarea de noi rânduri care se potrivesc criteriilor sale de căutare.

**4. SERIALIZABLE**

**Scenariu: Prevenirea Phantom Read și asigurarea consistenței complete**

**Stare inițială:** Avem produse în inventar.

**Evită:** Dirty Reads, Non-Repeatable Reads, Phantom Reads

**Asigură:** Cel mai înalt nivel de izolare. Tranzacțiile par a se executa secvențial, una după alta

**Tranzacția 1 (T1): Verifică și Procesează Produse Disponibile (simulăm o logică complexă)**

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

BEGIN TRANSACTION;

-- Verifică câte produse sunt în stoc

SELECT COUNT(\*) FROM PRODUSE\_INVENTAR WHERE CANTITATE\_DISPONIBILA > 0;

-- Returnează 10

-- ... Ruleaza alt cod în continuare ...

**Tranzacția 2 (T2): Adaugă un Produs Nou în Inventar**

SQL

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

BEGIN TRANSACTION;

INSERT INTO PRODUSE\_INVENTAR (ID\_PRODUS\_INVENTAR, NUME\_PRODUS, PRET\_UNITAR, CANTITATE\_DISPONIBILA, ID\_MENIU)

VALUES (103, 'Tort de ciocolată', 60.00, 20, 5);

-- Această inserare va fi blocată și va aștepta finalizarea T1, sau va eșua cu un eroare de timeout/deadlock în funcție de SGBD și configurare.

**Revenire Tranzacția 1:**

SQL

-- T1 se finalizează

COMMIT;

**Revenire Tranzacția 2 (după finalizarea T1):**

SQL

-- Acum T2 poate executa INSERT-ul.

COMMIT;

**Efect:** T1 vede exact același set de rânduri pe parcursul întregii sale execuții, indiferent de inserările sau ștergerile altor tranzacții. Tranzacția T2 este forțată să aștepte până când T1 se finalizează, asigurând că T1 nu va vedea rânduri "fantomă".