Cerinte BD

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

**Modelul real** al bazei de date pentru GadgetHaven include gestionarea produselor electronice, stocurilor, comenzilor, clienților și furnizorilor. Baza de date va facilita operațiuni precum adăugarea, modificarea și ștergerea produselor, plasarea comenzilor, urmărirea stocurilor și generarea rapoartelor.

**Utilitate:**

* Gestionarea eficientă a produselor și stocurilor.
* Facilitatea procesului de comandă și facturare.
* Urmărirea activităților clienților și furnizorilor.
* Optimizarea proceselor de inventariere și logistică.

**Reguli de funcționare:**

* Fiecare produs trebuie să aparțină unei categorii.
* Fiecare comandă trebuie să fie asociată cu un client și poate conține unul sau mai multe produse.
* Fiecare produs poate fi aprovizionat de unul sau mai mulți furnizori.
* Stocul produselor trebuie actualizat automat în urma plasării comenzilor.

1. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.

**Constrangeri:**

* **Integritatea referențială**: Cheile externe trebuie să fie respectate pentru a menține legăturile între tabele.
* **Integritatea domeniului**: Atributele trebuie să respecte tipurile de date definite și valorile permise.
* **Unicitate**: Cheile primare și cheile candidate trebuie să fie unice.
* **Non-null**: Anumite câmpuri, cum ar fi numele produsului, prețul, etc., nu pot fi null.
* **Valori implicite**: Atributele cu valori implicite trebuie să le folosească atunci când nu sunt specificate alte valori.

1. Descrierea entităților, incluzând precizarea cheii primare.
2. **Clienți** (Customers)
   * ID\_Clienți (PK)
   * Nume
   * Prenume
   * Email
   * Telefon
   * Adresă
3. **Produse** (Products)
   * ID\_Produse (PK)
   * Nume
   * Descriere
   * Preț
   * ID\_Categorie (FK)
4. **Categorii** (Categories)
   * ID\_Categorie (PK)
   * Nume\_Categorie
5. **Furnizori** (Suppliers)
   * ID\_Furnizori (PK)
   * Nume\_Furnizor
   * Contact
6. **Stocuri** (Inventory)
   * ID\_Stoc (PK)
   * ID\_Produse (FK)
   * Cantitate
   * Data\_Ultimei\_Aprovizionari
7. **Comenzi** (Orders)
   * ID\_Comenzi (PK)
   * ID\_Clienți (FK)
   * Data\_Comenzii
   * Total
8. **Detalii Comenzi** (Order\_Details)
   * ID\_Detalii (PK)
   * ID\_Comenzi (FK)
   * ID\_Produse (FK)
   * Cantitate
   * Preț\_Unitar
9. **Aprovizionări** (Supplies)
   * ID\_Aprovizionare (PK)
   * ID\_Produse (FK)
   * ID\_Furnizori (FK)
   * Cantitate
   * Data\_Aprovizionarii
10. Descrierea relațiilor, incluzând precizarea cardinalității acestora.

* **Clienți - Comenzi**: Un client poate avea mai multe comenzi (1).
* **Categorii - Produse**: O categorie poate conține mai multe produse (1).
* **Produse - Stocuri**: Fiecare produs are un stoc asociat (1:1).
* **Comenzi - Detalii Comenzi**: O comandă poate avea mai multe detalii de comandă (1).
* **Produse - Detalii Comenzi**: Fiecare detaliu de comandă se referă la un produs (1).
* **Furnizori - Aprovizionări**: Un furnizor poate aproviziona mai multe produse (1).
* **Produse - Aprovizionări**: Un produs poate fi aprovizionat de mai mulți furnizori (M), implementat prin tabelul Supplies.

1. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.
2. **Clienți**:
   * ID\_Clienți: INTEGER, PRIMARY KEY, AUTO\_INCREMENT
   * Nume: VARCHAR(50), NOT NULL
   * Prenume: VARCHAR(50), NOT NULL
   * Email: VARCHAR(100), NOT NULL, UNIQUE
   * Telefon: VARCHAR(15), NOT NULL
   * Adresă: VARCHAR(200), NOT NULL
3. **Produse**:
   * ID\_Produse: INTEGER, PRIMARY KEY, AUTO\_INCREMENT
   * Nume: VARCHAR(100), NOT NULL
   * Descriere: TEXT, NULL
   * Preț: DECIMAL(10, 2), NOT NULL
   * ID\_Categorie: INTEGER, FOREIGN KEY, NOT NULL
4. **Categorii**:
   * ID\_Categorie: INTEGER, PRIMARY KEY, AUTO\_INCREMENT
   * Nume\_Categorie: VARCHAR(50), NOT NULL
5. **Furnizori**:
   * ID\_Furnizori: INTEGER, PRIMARY KEY, AUTO\_INCREMENT
   * Nume\_Furnizor: VARCHAR(100), NOT NULL
   * Contact: VARCHAR(100), NOT NULL
6. **Stocuri**:
   * ID\_Stoc: INTEGER, PRIMARY KEY, AUTO\_INCREMENT
   * ID\_Produse: INTEGER, FOREIGN KEY, NOT NULL
   * Cantitate: INTEGER, NOT NULL
   * Data\_Ultimei\_Aprovizionari: DATE, NOT NULL
7. **Comenzi**:
   * ID\_Comenzi: INTEGER, PRIMARY KEY, AUTO\_INCREMENT
   * ID\_Clienți: INTEGER, FOREIGN KEY, NOT NULL
   * Data\_Comenzii: DATE, NOT NULL
   * Total: DECIMAL(10, 2), NOT NULL
8. **Detalii Comenzi**:
   * ID\_Detalii: INTEGER, PRIMARY KEY, AUTO\_INCREMENT
   * ID\_Comenzi: INTEGER, FOREIGN KEY, NOT NULL
   * ID\_Produse: INTEGER, FOREIGN KEY, NOT NULL
   * Cantitate: INTEGER, NOT NULL
   * Preț\_Unitar: DECIMAL(10, 2), NOT NULL
9. **Aprovizionări**:
   * ID\_Aprovizionare: INTEGER, PRIMARY KEY, AUTO\_INCREMENT
   * ID\_Produse: INTEGER, FOREIGN KEY, NOT NULL
   * ID\_Furnizori: INTEGER, FOREIGN KEY, NOT NULL
   * Cantitate: INTEGER, NOT NULL
   * Data\_Aprovizionarii: DATE, NOT NULL
10. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.
11. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 7 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.
12. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.
13. Realizarea normalizării până la forma normală 3 (FN1-FN3).
14. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).
15. 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).
16. 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) în care intervin cel puțin 3 tabele (in cadrul aceleiași cereri)
    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)