Baza de date

„FonoDesign”

Nițe Dan-Alexandru

Grupa 144

Anul I

Baza de date a unei firme care se ocupă de producerea si comercializarea difuzoarelor acustice

**Tema proiectului:** Dezvoltarea unei baze de date destinată să optimizeze gestionarea si organizarea datelor associate cu sistemul de fabricație și distribuție a difuzoarelor. Aceasta va include detalii despre produse, componente, loturi de producție, clienți, comenzi, expediții și retururi. Produsele sunt clasificate în diverse modele, fiecare cu specificații tehnice proprii. Componentele necesare pentru fabricația difuzoarelor provin de la diferiți furnizori și sunt gestionate într-un inventar detaliat.

Loturile de producție sunt planificate pentru a optimiza fluxul de fabricație și sunt asociate cu date specifice de începere și terminare. Comenzile de la clienți sunt procesate în sistem, fiecare comandă fiind urmărită de la plasare până la finalizare, inclusiv detaliile expedițiilor și statusul acestora. Fiecare expediție este asociată cu un vehicul din departamentul de transport și un curier, asigurându-se că produsele ajung la destinație în mod eficient. În cazul retururilor, sistemul înregistrează motivele returnării, produsul implicat și acțiunile finale, fie acestea rambursare sau înlocuire.

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

Această bază de date este destinată să optimizeze gestionarea și organizarea datelor asociate cu sistemul de fabricație și distribuție a difuzoarelor. Modelul include tabele care stochează informații despre furnizori, produse, componente, inventar, loturi de producție, clienți, comenzi, expediții și retururi.

#### Entitățile și relațiile principale sunt:

* **Furnizori**: Stochează informații despre furnizorii de componente, inclusiv nume, adresă și email.
* **Componente**: Include detalii despre componentele utilizate în fabricația difuzoarelor, fiecare componentă fiind asociată cu un furnizor specific.
* **Produs**: Conține informații despre produsele finale, inclusiv numele, descrierea și tipul produsului.
* **Inventar**: Găzduiește date despre stocurile de componente, inclusiv cantitatea disponibilă și cea comandată.
* **Loturi\_Producție**: Gestionează planificarea producției, inclusiv datele de începere și terminare ale fiecărui lot.
* **Client**: Înregistrează informații despre clienți, inclusiv numele, prenumele, adresa, telefonul și emailul.
* **Comenzi**: Urmărește comenzile de la clienți, inclusiv data comenzii, totalul și statusul.
* **Detalii\_Comenzi**: Include detalii despre produsele comandate, cantitatea și prețul per unitate.
* **Expedieri**: Gestionează expedițiile comenzilor, inclusiv data expedierii, data estimată de livrare și statusul.
* **Retururi**: Înregistrează retururile, inclusiv data returului, motivul și acțiunile finale.
* **Departament\_Transport**: Conține informații despre vehiculele din departamentul de transport, inclusiv modelul, marca, numărul de licență și statusul.
* **Curier**: Stochează detalii despre curieri, inclusiv numele, prenumele, telefonul și emailul, precum și vehiculul alocat.

Utilitatea acestei baze de date constă în îmbunătățirea eficienței gestionării și organizării datelor referitoare la fabricația și distribuția difuzoarelor.

#### Beneficiile includ:

* **Optimizarea producției**: Planificarea eficientă a loturilor de producție pentru a maximiza fluxul de fabricație și a reduce timpul de inactivitate.
* **Gestionarea inventarului**: Monitorizarea stocurilor de componente pentru a preveni supra-stocarea sau deficitul de materiale.
* **Urmărirea comenzilor**: Gestionarea completă a comenzilor de la plasare până la finalizare, incluzând expedițiile și retururile.
* **Eficientizarea transportului**: Alocarea eficientă a vehiculelor și curierilor pentru a asigura livrarea promptă a produselor.
* **Rapoarte detaliate**: Generarea de rapoarte privind producția, stocurile, comenzile și expedițiile, utile pentru analiza performanței și luarea deciziilor informate.

Funcționalitățile bazei de date includ:

* **Stocarea și gestionarea datelor**: Informații detaliate despre furnizori, produse, componente și inventar.
* **Planificarea producției**: Gestionarea loturilor de producție, incluzând datele de începere și terminare.
* **Urmărirea comenzilor**: Înregistrarea și monitorizarea comenzilor de la clienți, inclusiv detaliile expedițiilor și statusul acestora.
* **Gestionarea expedițiilor și retururilor**: Urmărirea expedițiilor și înregistrarea retururilor, incluzând motivele și acțiunile finale.
* **Administrarea transportului**: Gestionarea vehiculelor și curierilor, incluzând alocarea vehiculelor pentru expediții.
* **Generarea rapoartelor**: Crearea de rapoarte detaliate despre producție, inventar, comenzi, expediții și retururi.

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

**1. Tabela Furnizor**

* **ID\_Furnizor**: Cheia primară, unică pentru fiecare furnizor.
* **Nume**: Numele furnizorului, cu o lungime maximă de 255 de caractere.
* **Adresă**: Adresa furnizorului, cu o lungime maximă de 255 de caractere.
* **Email**: Adresa de email a furnizorului, cu o lungime maximă de 255 de caractere.
* **Numar\_Contact**: Numărul de telefon pentru contactarea furnizorului, cu o lungime maxima de 20 de caractere.

**2. Tabela Componenta**

* **ID\_Componenta**: Cheia primară, unică pentru fiecare componentă.
* **Nume**: Numele componentei, cu o lungime maximă de 255 de caractere.
* **Tip**: Tipul componentei, cu o lungime maximă de 100 de caractere.
* **ID\_Furnizor**: Cheie externă către tabela Furnizori, referindu-se la furnizorul componentei.
* **Pret**: Prețul componentei, cu o lungime maxima de 6 caractere.
* **ID\_Produs**: Cheie externă către tabela Produs, referindu-se la produsul căruia îi este specifică componenta.

**3. Tabela Produs**

* **ID\_Produs**: Cheia primară, unică pentru fiecare produs.
* **Nume**: Numele produsului, cu o lungime maximă de 255 de caractere.
* **Descriere**: Descrierea produsului, cu o lungime maximă de 255 de caractere.
* **Pret\_Vanzare**: Prețul cu care se vinde produsul.
* **Cost\_Producere:** Prețul cu care se fabrică produsul.

**4. Tabela Inventar**

* **ID\_Inventar**: Cheia primară, unică pentru fiecare înregistrare de inventar.
* **ID\_Componenta**: Cheie externă către tabela Componente, referindu-se la componenta specifică.
* **Cantitate\_Disponibila**: Cantitatea disponibilă în inventar.
* **Cantitate\_Comandata**: Cantitatea comandată din inventar.

**5. Tabela Loturi\_Produs**

* **ID\_Lot**: Cheia primară, unică pentru fiecare lot de producție.
* **ID\_Produs**: Cheie externă către tabela Produs, referindu-se la produsul specific fabricat.
* **Data\_Începere**: Data de începere a lotului de producție.
* **Data\_Terminare**: Data de terminare a lotului de producție.
* **Cantitate\_Produsa**: Câte produse se găsesc în lot.

**6. Tabela Client**

* **ID\_Client**: Cheia primară, unică pentru fiecare client.
* **Nume**: Numele clientului, cu o lungime maximă de 255 de caractere.
* **Prenume**: Prenumele clientului, cu o lungime maximă de 255 de caractere.
* **Adresă**: Adresa clientului, cu o lungime maximă de 255 de caractere.
* **Cod\_Poștal**: Codul poștal al clientului, cu o lungime maxima de 6 caractere.
* **Telefon**: Numărul de telefon al clientului, cu o lungime maximă de 20 de caractere.
* **Email**: Adresa de email a clientului, cu o lungime maximă de 255 de caractere.

**7. Tabela Comandă**

* **ID\_Comanda**: Cheia primară, unică pentru fiecare comandă.
* **ID\_Client**: Cheie externă către tabela Client, referindu-se la clientul care a plasat comanda.
* **Data\_Comanda**: Data plasării comenzii.
* **Pret\_Total**: Totalul comenzii.
* **Status**: Statusul comenzii, cu o lungime maximă de 50 de caractere.

**8. Tabela Detalii\_Comanda**

* **ID\_Detalii\_Comanda**: Cheia primară, unică pentru fiecare detaliu de comandă.
* **ID\_Comanda**: Cheie externă către tabela Comenzi, referindu-se la comanda specifică.
* **ID\_Produs**: Cheie externă către tabela Produs, referindu-se la produsul specific comandat.
* **Cantitate**: Cantitatea comandată.
* **Preț\_Per\_Unitate**: Prețul per unitate al produsului.

**9. Tabela Expediere**

* **ID\_Expediere**: Cheia primară, unică pentru fiecare expediere.
* **ID\_Comanda**: Cheie externă către tabela Comenzi, referindu-se la comanda specifică.
* **Data\_Expediere**: Data expedierii comenzii.
* **Data\_Estimare\_Livrare**: Data estimată a livrării.
* **ID\_Zona\_Transport**: Cheie externă către tabela Departament\_Transport, referindu-se la hub-ul de unde se va efectua transportul.
* **Status**: Statusul expedierii, cu o lungime maximă de 50 de caractere.

**10. Tabela Retur**

* **ID\_Retur**: Cheia primară, unică pentru fiecare retur.
* **ID\_Comanda**: Cheie externă către tabela Comenzi, referindu-se la comanda specifică.
* **ID\_Produs**: Cheie externă către tabela Produs, referindu-se la produsul specific.
* **Data\_Retur**: Data returului.
* **Motiv**: Motivul returului, cu o lungime maximă de 255 de caractere.
* **Acțiuni\_Finalizare**: Acțiunile întreprinse pentru finalizarea returului, cu o lungime maximă de 100 de caractere.

**11. Tabela Departament\_Transport**

* **ID\_Transport**: Cheia primară, unică pentru fiecare vehicul din departamentul de transport.
* **Mașină:** Mașina asociată departamentului de transport respectiv.

**12. Tabela Curier**

* **ID\_Curier**: Cheia primară, unică pentru fiecare curier.
* **Nume**: Numele curierului, cu o lungime maximă de 255 de caractere.
* **Prenume**: Prenumele curierului, cu o lungime maximă de 255 de caractere.
* **Telefon**: Numărul de telefon al curierului, cu o lungime maximă de 20 de caractere.
* **Email**: Adresa de email a curierului, cu o lungime maximă de 255 de caractere.
* **ID\_Zona\_Transport**: Cheie externă către tabela Departament\_Transport, referindu-se la vehiculul alocat curierului.

3.Descrierea entităților, incluzând precizarea cheii primare

1. **Entitatea Furnizor**: Această tabelă conține informații despre furnizorii care livrează componente pentru producția de difuzoare. Fiecare înregistrare este identificată prin cheia primară ID\_Furnizor și conține informații despre numele furnizorului (Nume), adresa (Adresa), numărul de contact (Numar\_Contact) și adresa de email (Adresa\_Mail). Aceste informații sunt esențiale pentru gestionarea relațiilor cu partenerii care furnizează materialele necesare producției.
2. **Entitatea Componenta**: Această tabelă stochează date despre componentele utilizate în asamblarea difuzoarelor. Fiecare componentă este unic identificată prin cheia primară ID\_Componenta și include detalii despre numele (Nume), tipul (Tip) și prețul (Pret) componentei, precum și ID-ul furnizorului (ID\_Furnizor - FK) care o livrează. Aceste detalii ajută la urmărirea costurilor de producție și la gestionarea inventarului.
3. **Entitatea Produs**: Această tabelă conține informații despre diferitele tipuri de difuzoare produse de companie. Fiecare produs este identificat prin cheia primară ID\_Produs și include informații precum numele (Nume), descrierea (Descriere), prețul de vânzare (Pret\_Vanzare) și costul de producție (Cost\_Productie). Informațiile din această tabelă sunt vitale pentru stabilirea strategiilor de preț și pentru analiza profitabilității.
4. **Entitatea Loturi\_Produs**: Această tabelă urmărește loturile de producție pentru fiecare tip de produs. Identificată prin cheia primară ID\_Lot, aceasta conține legături la produsele specifice (ID\_Produs - FK), data de începere a producției (Data\_Incepere), data de terminare (Data\_Terminare) și cantitatea produsă (Cantitate\_Produsa). Informațiile ajută la planificarea și eficientizarea producției.
5. **Entitatea Inventar**: Această tabelă gestionează stocurile de componente necesare producției. Fiecare înregistrare este identificată prin cheia primară ID\_Inventar și conține informații despre componenta stocată (ID\_Componenta - FK), cantitatea disponibilă (Cantitate\_Disponibila) și cantitatea comandată (Cantitate\_Comandata). Aceste informații sunt esențiale pentru a asigura un flux constant de materiale pentru producție.
6. **Entitatea Client**: Această tabelă documentează informațiile despre clienții care cumpără produse. Fiecare client este unic identificat prin cheia primară ID\_Client și include numele de familie (Numele\_Familie), prenumele (Prenumele), adresa (Adresa), codul poștal (Cod\_Postal), numărul de telefon (Numar\_Telefon) și adresa de email (Email). Această tabelă este esențială pentru gestionarea relațiilor cu clienții și pentru activitățile de marketing.
7. **Entitatea Comanda**: Această tabelă gestionează comenzile plasate de clienți. Identificată prin cheia primară ID\_Comanda, aceasta include referințe la clientul specific (ID\_Client - FK), data comenzii (Data\_Comanda), prețul total (Pret\_Total) și starea comenzii (Status). Această tabelă este crucială pentru monitorizarea și gestionarea fluxului de comenzi.
8. **Entitatea Detalii\_Comanda**: Această tabelă conține detaliile specifice ale fiecărei comenzi, inclusiv produsele comandate. Fiecare detaliu este identificat prin cheia primară ID\_Detalii\_Comanda și include referințe la comanda și produsul specific (ID\_Comanda - FK, ID\_Produs - FK), cantitatea comandată (Cantitate) și prețul per unitate (Pret\_Per\_Unitate). Aceste informații permit o gestionare detaliată a fiecărei comenzi.
9. **Entitatea Expediere**: Această tabelă urmărește expedierea comenzilor către clienți. Identificată prin cheia primară ID\_Expediere, aceasta include referințe la comanda specifică (ID\_Comanda - FK), data expedierii (Data\_Expediere), data estimată pentru livrare (Data\_Estimata\_Livrare), zona de transport (ID\_Zona\_Transport - FK) și statusul expediției (Status). Această tabelă este vitală pentru coordonarea livrărilor și asigurarea satisfacției clienților.
10. **Entitatea Departament\_Transport**: Această tabelă conține informații despre zonele de transport și vehiculele utilizate pentru expedierea produselor. Fiecare înregistrare este identificată prin cheia primară ID\_Zona\_Transport și conține detalii despre vehicul (Masina). Aceste informații sunt importante pentru gestionarea eficientă a resurselor logistice.
11. **Entitatea Angajat**: Această tabelă stochează informații despre angajații care lucrează în departamentul de transport. Fiecare angajat este identificat prin cheia primară ID\_Angajat și include informații precum numele de familie (Numele\_Familie), prenumele (Prenumele), bonusul lunar (Bonus\_Lunar), salariul lunar (Salariu\_Lunar) și zona de transport (ID\_Zona\_Transport - FK). Aceste date sunt esențiale pentru gestionarea personalului și a logisticii de livrare.
12. **Entitatea Retur**: Această tabelă gestionează returnările produselor de către clienți. Fiecare retur este identificat prin cheia primară ID\_Retur și include referințe la comanda și produsul returnat (ID\_Comanda - FK, ID\_Produs - FK), motivul returnării (Motiv), data returnării (Data\_Retur) și acțiunea finală (Actiune\_Finala). Această tabelă este crucială pentru procesarea eficientă a returnărilor și asigurarea satisfacției clienților.

4.Descrierea relațiilor, incluzând precizarea cardinalității acestora.

 **Relația între Furnizor și Componenta**:

* **Descriere**: Un furnizor poate furniza mai multe componente, dar fiecare componentă este furnizată de un singur furnizor.
* **Cardinalitate**: Furnizor (1) --- (0,N) Componenta
* **Chei**: ID\_Furnizor (PK în Furnizor, FK în Componenta)

 **Relația între Componenta și Produs**:

* **Descriere**: O componentă poate face parte din mai multe produse, iar un produs poate include mai multe componente.
* **Cardinalitate**: Componenta (0,N) --- (0,N) Produs
* **Chei**: ID\_Produs (PK în Produs, FK în Componenta)

 **Relația între Produs și Loturi\_Produs**:

* **Descriere**: Un produs poate fi fabricat în mai multe loturi, iar fiecare lot este asociat cu un singur produs.
* **Cardinalitate**: Produs (1) --- (0,N) Loturi\_Produs
* **Chei**: ID\_Produs (PK în Produs, FK în Loturi\_Produs)

 **Relația între Componenta și Inventar**:

* **Descriere**: O componentă poate fi stocată în inventar de mai multe ori (de exemplu, în diferite locații), iar fiecare intrare în inventar se referă la o singură componentă.
* **Cardinalitate**: Componenta (1) --- (0,N) Inventar
* **Chei**: ID\_Componenta (PK în Componenta, FK în Inventar)

 **Relația între Client și Comanda**:

* **Descriere**: Un client poate plasa mai multe comenzi, dar fiecare comandă este plasată de un singur client.
* **Cardinalitate**: Client (1) --- (0,N) Comanda
* **Chei**: ID\_Client (PK în Client, FK în Comanda)

 **Relația între Comanda și Detalii\_Comanda**:

* **Descriere**: O comandă poate conține mai multe detalii despre produse, iar fiecare detaliu de comandă se referă la o singură comandă.
* **Cardinalitate**: Comanda (1) --- (0,N) Detalii\_Comanda
* **Chei**: ID\_Comanda (PK în Comanda, FK în Detalii\_Comanda)

 **Relația între Produs și Detalii\_Comanda**:

* **Descriere**: Un produs poate apărea în detalii multiple ale comenzilor, iar fiecare detaliu de comandă se referă la un singur produs.
* **Cardinalitate**: Produs (1) --- (0,N) Detalii\_Comanda
* **Chei**: ID\_Produs (PK în Produs, FK în Detalii\_Comanda)

 **Relația între Comanda și Expediere**:

* **Descriere**: O comandă poate fi expediată o singură dată, iar fiecare expediere este asociată cu o singură comandă.
* **Cardinalitate**: Comanda (1) --- (0,1) Expediere
* **Chei**: ID\_Comanda (PK în Comanda, FK în Expediere)

 **Relația între Departament\_Transport și Expediere**:

* **Descriere**: Un departament de transport poate gestiona multiple expediții, iar fiecare expediere este gestionată de un singur departament de transport.
* **Cardinalitate**: Departament\_Transport (1) --- (0,N) Expediere
* **Chei**: ID\_Zona\_Transport (PK în Departament\_Transport, FK în Expediere)

 **Relația între Departament\_Transport și Angajat**:

* **Descriere**: Un departament de transport poate avea mai mulți angajați, iar fiecare angajat lucrează într-un singur departament de transport.
* **Cardinalitate**: Departament\_Transport (1) --- (0,N) Angajat
* **Chei**: ID\_Zona\_Transport (PK în Departament\_Transport, FK în Angajat)

 **Relația între Comanda și Retur**:

* **Descriere**: O comandă poate avea mai multe retururi, iar fiecare retur este asociat cu o singură comandă.
* **Cardinalitate**: Comanda (1) --- (0,N) Retur
* **Chei**: ID\_Comanda (PK în Comanda, FK în Retur)

 **Relația între Produs și Retur**:

* **Descriere**: Un produs poate apărea în mai multe retururi, iar fiecare retur se referă la un singur produs.
* **Cardinalitate**: Produs (1) --- (0,N) Retur
* **Chei**: ID\_Produs (PK în Produs, FK în Retur)

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

 **Tabela Furnizor**:

* ID\_Furnizor (NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY): identificator unic al furnizorului.
* Nume (VARCHAR2(100)): numele furnizorului, stocat sub formă de șir de caractere cu lungimea maximă de 100.
* Adresa (VARCHAR2(255)): adresa furnizorului, stocată sub formă de șir de caractere cu lungimea maximă de 255.
* Numar\_Contact (VARCHAR2(15)): numărul de contact al furnizorului, stocat sub formă de șir de caractere cu lungimea maximă de 15.
* Adresa\_Mail (VARCHAR2(100)): adresa de email a furnizorului, stocată sub formă de șir de caractere cu lungimea maximă de 100.

 **Tabela Componenta**:

* ID\_Componenta (NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY): identificator unic al componentei.
* Nume (VARCHAR2(100)): numele componentei, stocat sub formă de șir de caractere cu lungimea maximă de 100.
* Tip (VARCHAR2(50)): tipul componentei, stocat sub formă de șir de caractere cu lungimea maximă de 50.
* Pret (NUMBER(10, 2)): prețul componentei, stocat ca valoare zecimală cu 2 zecimale.
* ID\_Furnizor (NUMBER): identificatorul furnizorului care furnizează componenta, cheie străină referindu-se la Furnizor(ID\_Furnizor).

 **Tabela Produs**:

* ID\_Produs (NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY): identificator unic al produsului.
* Nume (VARCHAR2(100)): numele produsului, stocat sub formă de șir de caractere cu lungimea maximă de 100.
* Descriere (CLOB): descrierea produsului, stocată ca text de tip CLOB.
* Pret\_Vanzare (NUMBER(10, 2)): prețul de vânzare al produsului, stocat ca valoare zecimală cu 2 zecimale.
* Cost\_Productie (NUMBER(10, 2)): costul de producție al produsului, stocat ca valoare zecimală cu 2 zecimale.

 **Tabela Loturi\_Produs**:

* ID\_Lot (NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY): identificator unic al lotului de producție.
* ID\_Produs (NUMBER): identificatorul produsului asociat lotului, cheie străină referindu-se la Produs(ID\_Produs).
* Data\_Incepere (DATE): data începerii producției lotului.
* Data\_Terminare (DATE): data terminării producției lotului.
* Cantitate\_Produsa (NUMBER): cantitatea de produse fabricate în acest lot.

 **Tabela Inventar**:

* ID\_Inventar (NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY): identificator unic al înregistrării din inventar.
* ID\_Componenta (NUMBER): identificatorul componentei stocate în inventar, cheie străină referindu-se la Componenta(ID\_Componenta).
* Cantitate\_Disponibila (NUMBER): cantitatea disponibilă a componentei.
* Cantitate\_Comandata (NUMBER): cantitatea componentei comandată, dar încă nelivrată.

 **Tabela Client**:

* ID\_Client (NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY): identificator unic al clientului.
* Numele\_Familie (VARCHAR2(100)): numele de familie al clientului, stocat sub formă de șir de caractere cu lungimea maximă de 100.
* Prenumele (VARCHAR2(100)): prenumele clientului, stocat sub formă de șir de caractere cu lungimea maximă de 100.
* Adresa (VARCHAR2(255)): adresa completă a clientului, stocată sub formă de șir de caractere cu lungimea maximă de 255.
* Cod\_Postal (VARCHAR2(10)): codul poștal al clientului, stocat sub formă de șir de caractere cu lungimea maximă de 10.
* Numar\_Telefon (VARCHAR2(15)): numărul de telefon al clientului, stocat sub formă de șir de caractere cu lungimea maximă de 15.
* Email (VARCHAR2(100)): adresa de email a clientului, stocată sub formă de șir de caractere cu lungimea maximă de 100.

 **Tabela Comanda**:

* ID\_Comanda (NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY): identificator unic al comenzii.
* ID\_Client (NUMBER): identificatorul clientului care a plasat comanda, cheie străină referindu-se la Client(ID\_Client).
* Data\_Comanda (DATE): data plasării comenzii.
* Pret\_Total (NUMBER(10, 2)): prețul total al comenzii, stocat ca valoare zecimală cu 2 zecimale.
* Status (VARCHAR2(50)): statusul comenzii (e.g., "procesată", "expediată", "finalizată"), stocat sub formă de șir de caractere cu lungimea maximă de 50.

 **Tabela Detalii\_Comanda**:

* ID\_Detalii\_Comanda (NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY): identificator unic al detaliilor comenzii.
* ID\_Comanda (NUMBER): identificatorul comenzii asociate, cheie străină referindu-se la Comanda(ID\_Comanda).
* ID\_Produs (NUMBER): identificatorul produsului comandat, cheie străină referindu-se la Produs(ID\_Produs).
* Cantitate (NUMBER): cantitatea produsului comandat.
* Pret\_Per\_Unitate (NUMBER(10, 2)): prețul pe unitate al produsului, stocat ca valoare zecimală cu 2 zecimale.

 **Tabela Expediere**:

* ID\_Expediere (NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY): identificator unic al expedierii.
* ID\_Comanda (NUMBER): identificatorul comenzii asociate, cheie străină referindu-se la Comanda(ID\_Comanda).
* Data\_Expediere (DATE): data expedierii comenzii.
* Data\_Estimata\_Livrare (DATE): data estimată pentru livrarea comenzii.
* ID\_Zona\_Transport (NUMBER): identificatorul zonei de transport responsabilă de expediere, cheie străină referindu-se la Departament\_Transport(ID\_Zona\_Transport).
* Status (VARCHAR2(50)): statusul expedierii (e.g., "în curs de livrare", "livrat"), stocat sub formă de șir de caractere cu lungimea maximă de 50.

 **Tabela Departament\_Transport**:

* ID\_Zona\_Transport (NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY): identificator unic al zonei de transport.
* Masina (VARCHAR2(50)): identificarea vehiculului utilizat pentru transport, stocată sub formă de șir de caractere cu lungimea maximă de 50.

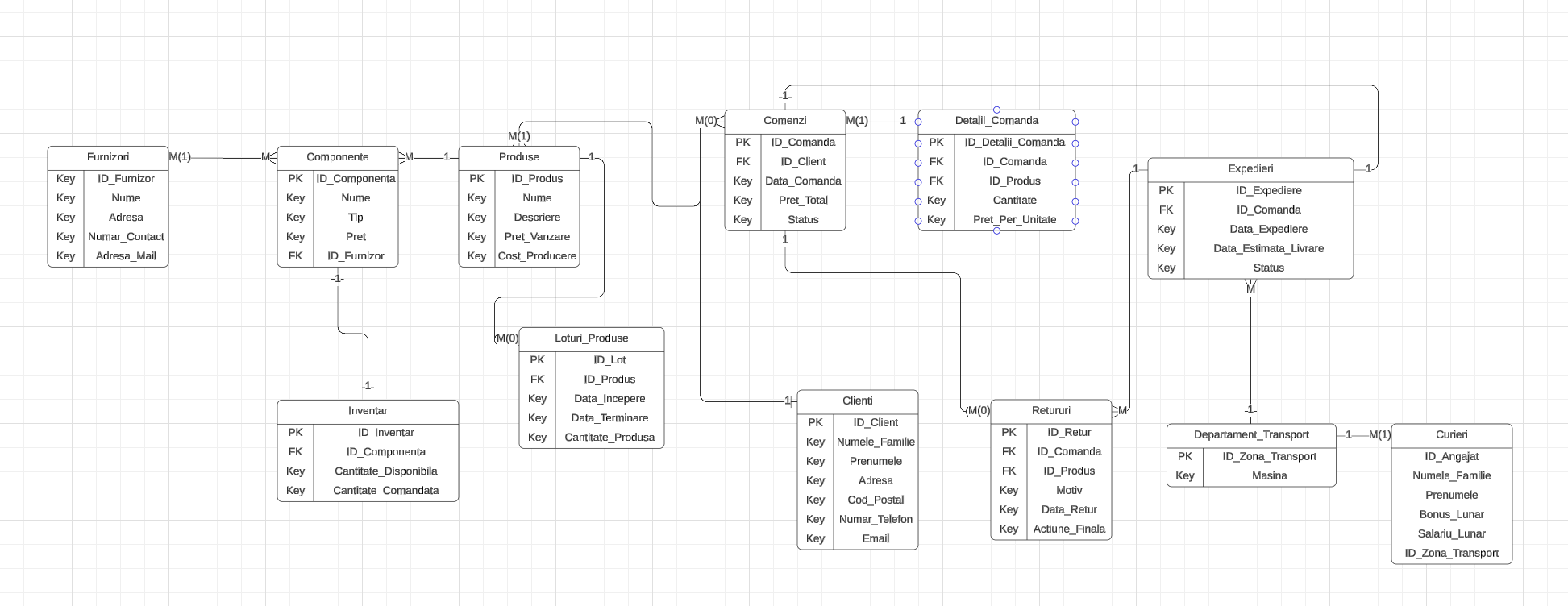
 **Tabela Angajat**:

* ID\_Angajat (NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY): identificator unic al angajatului.
* Numele\_Familie (VARCHAR2(100)): numele de familie al angajatului, stocat sub formă de șir de caractere cu lungimea maximă de 100.
* Prenumele (VARCHAR2(100)): prenumele angajatului, stocat sub formă de șir de caractere cu lungimea maximă de 100.
* Bonus\_Lunar (NUMBER(10, 2)): bonusul lunar al angajatului, stocat ca valoare zecimală cu 2 zecimale.
* Salariu\_Lunar (NUMBER(10, 2)): salariul lunar al angajatului, stocat ca valoare zecimală cu 2 zecimale.
* ID\_Zona\_Transport (NUMBER): identificatorul zonei de transport la care este alocat angajatul, cheie străină referindu-se la Departament\_Transport(ID\_Zona\_Transport).

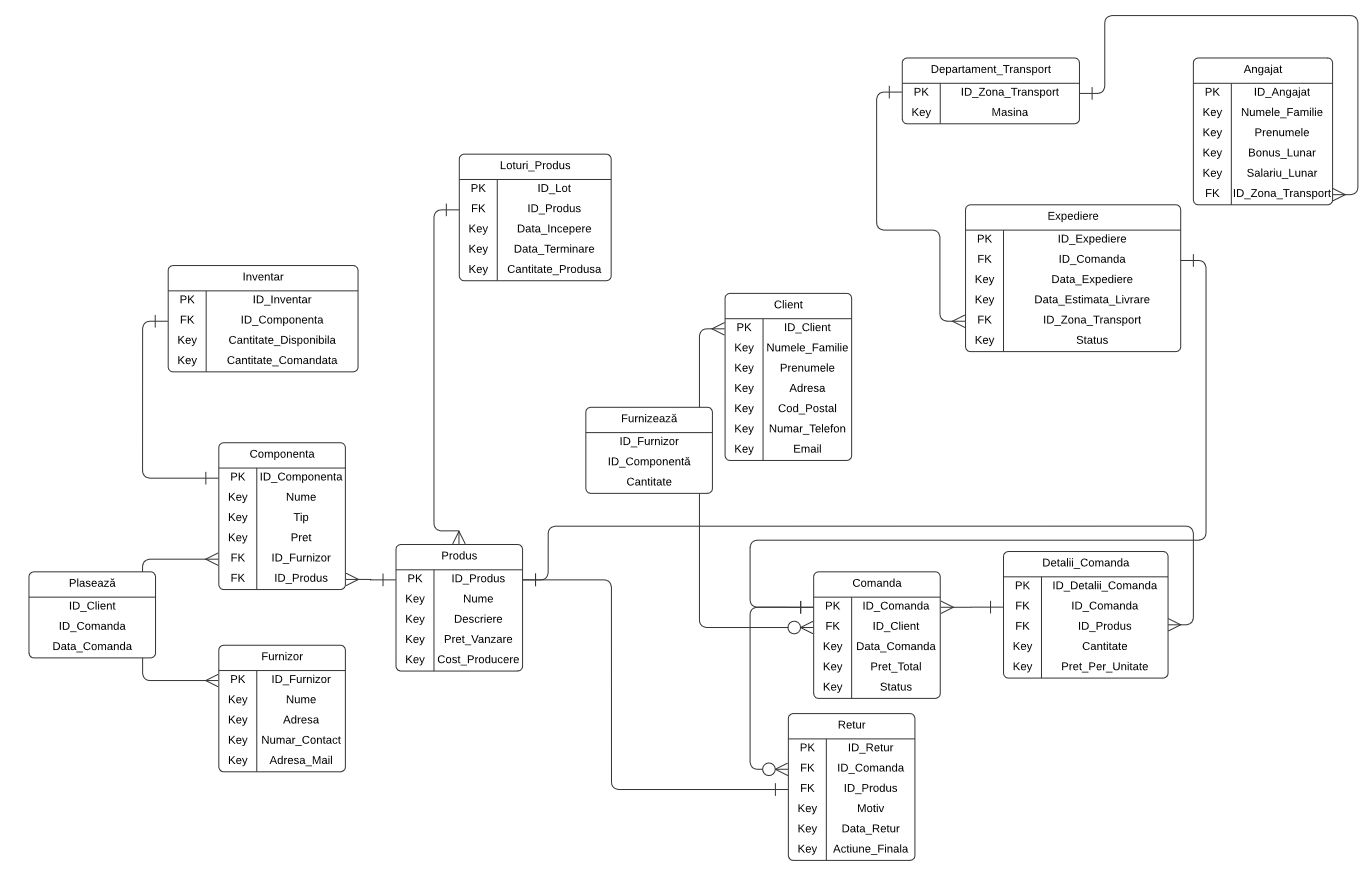
 **Tabela Retur**:

* ID\_Retur (NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY): identificator unic al returului.
* ID\_Comanda (NUMBER): identificatorul comenzii asociate returului, cheie străină referindu-se la Comanda(ID\_Comanda).
* ID\_Produs (NUMBER): identificatorul produsului returnat, cheie străină referindu-se la Produs(ID\_Produs).
* Motiv (VARCHAR2(255)): motivul returului, stocat sub formă de șir de caractere cu lungimea maximă de 255.
* Data\_Retur (DATE): data la care a fost efectuat returul.
* Actiune\_Finala (VARCHAR2(50)): acțiunea finală luată pentru retur (e.g., "rambursare", "înlocuire"), stocată sub formă de șir de caractere cu lungimea maximă de 50.

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



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



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

Schemele relaționale:

 **Furnizor** (ID\_Furnizor, Nume, Adresa, Numar\_Contact, Adresa\_Mail)

 **Componenta** (ID\_Componenta, Nume, Tip, Pret, ID\_Furnizor, ID\_Produs)

 **Produs** (ID\_Produs, Nume, Descriere, Pret\_Vanzare, Cost\_Productie)

 **Loturi\_Produs** (ID\_Lot, ID\_Produs, Data\_Incepere, Data\_Terminare, Cantitate\_Produsa)

 **Inventar** (ID\_Inventar, ID\_Componenta, Cantitate\_Disponibila, Cantitate\_Comandata)

 **Client** (ID\_Client, Numele\_Familie, Prenumele, Adresa, Cod\_Postal, Numar\_Telefon, Email)

 **Comanda** (ID\_Comanda, ID\_Client, Data\_Comanda, Pret\_Total, Status)

 **Detalii\_Comanda** (ID\_Detalii\_Comanda, ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate)

 **Expediere** (ID\_Expediere, ID\_Comanda, Data\_Expediere, Data\_Estimata\_Livrare, ID\_Zona\_Transport, Status)

 **Departament\_Transport** (ID\_Zona\_Transport, Masina)

 **Angajat** (ID\_Angajat, Numele\_Familie, Prenumele, Bonus\_Lunar, Salariu\_Lunar, ID\_Zona\_Transport)

 **Retur** (ID\_Retur, ID\_Comanda, ID\_Produs, Motiv, Data\_Retur, Actiune\_Finala)

 **Furnizeaza** (ID\_Furnizor, ID\_Componenta, Cantitate)

 **Plaseaza** (ID\_Client, ID\_Comanda, Data\_Comanda)

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

### Forma Normală 1 (FN1)

O relație se află în FN1 dacă fiecare atribut al tabelului conține doar valori atomice, adică fiecare coloană are valori indivizibile.

**Exemplu de NON-FN1: Tabela Produs**

Să presupunem că avem un atribut Specificatii în tabela Produs care conține valori concatenate, de exemplu: "Dimensiune: 30cm; Culoare: Negru; Putere: 50W".

CREATE TABLE Produs (

ID\_Produs NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

Nume VARCHAR2(100),

Specificatii VARCHAR2(255) -- exemplu de atribut non-atomic

);

Pentru a aduce această tabelă în FN1, trebuie să descompunem Specificatii în atribute separate:

CREATE TABLE Produs (

ID\_Produs NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

Nume VARCHAR2(100),

Dimensiune VARCHAR2(50),

Culoare VARCHAR2(50),

Putere VARCHAR2(50)

);

### Forma Normală 2 (FN2)

O relație se află în FN2 dacă este în FN1 și fiecare atribut non-cheie este complet dependent de cheie primară (adică nu există dependențe parțiale).

**Exemplu de NON-FN2: Tabela Detalii\_Comanda**

Să presupunem că în tabela Detalii\_Comanda avem un atribut Pret\_Total care depinde de Pret\_Per\_Unitate și Cantitate.

CREATE TABLE Detalii\_Comanda (

ID\_Detalii\_Comanda NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

ID\_Comanda NUMBER,

ID\_Produs NUMBER,

Cantitate NUMBER,

Pret\_Per\_Unitate NUMBER(10, 2),

Pret\_Total NUMBER(10, 2) -- dependență parțială de `Pret\_Per\_Unitate` și `Cantitate`

);

Pentru a aduce această tabelă în FN2, eliminăm Pret\_Total și calculăm această valoare când este necesar.

CREATE TABLE Detalii\_Comanda (

ID\_Detalii\_Comanda NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

ID\_Comanda NUMBER,

ID\_Produs NUMBER,

Cantitate NUMBER,

Pret\_Per\_Unitate NUMBER(10, 2)

);

### Forma Normală 3 (FN3)

O relație se află în FN3 dacă este în FN2 și nu există dependențe tranzitive (adică atributele non-cheie nu depind de alte atribute non-cheie).

**Exemplu de NON-FN3: Tabela Comanda**

Să presupunem că în tabela Comanda avem un atribut Nume\_Client care depinde de ID\_Client.

CREATE TABLE Comanda (

ID\_Comanda NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

ID\_Client NUMBER,

Data\_Comanda DATE,

Pret\_Total NUMBER(10, 2),

Status VARCHAR2(50),

Nume\_Client VARCHAR2(100) -- dependență tranzitivă de `ID\_Client`

);

Pentru a aduce această tabelă în FN3, eliminăm Nume\_Client din Comanda și păstrăm această informație doar în tabela Client.

CREATE TABLE Comanda (

ID\_Comanda NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

ID\_Client NUMBER,

Data\_Comanda DATE,

Pret\_Total NUMBER(10, 2),

Status VARCHAR2(50),

CONSTRAINT fk\_client FOREIGN KEY (ID\_Client) REFERENCES Client(ID\_Client)

);

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

Codul sursă SQL:

CREATE SEQUENCE Furnizor\_SEQ START WITH 1;

CREATE SEQUENCE Componenta\_SEQ START WITH 1;

CREATE SEQUENCE Produs\_SEQ START WITH 1;

CREATE SEQUENCE Loturi\_Produs\_SEQ START WITH 1;

CREATE SEQUENCE Inventar\_SEQ START WITH 1;

CREATE SEQUENCE Client\_SEQ START WITH 1;

CREATE SEQUENCE Comanda\_SEQ START WITH 1;

CREATE SEQUENCE Detalii\_Comanda\_SEQ START WITH 1;

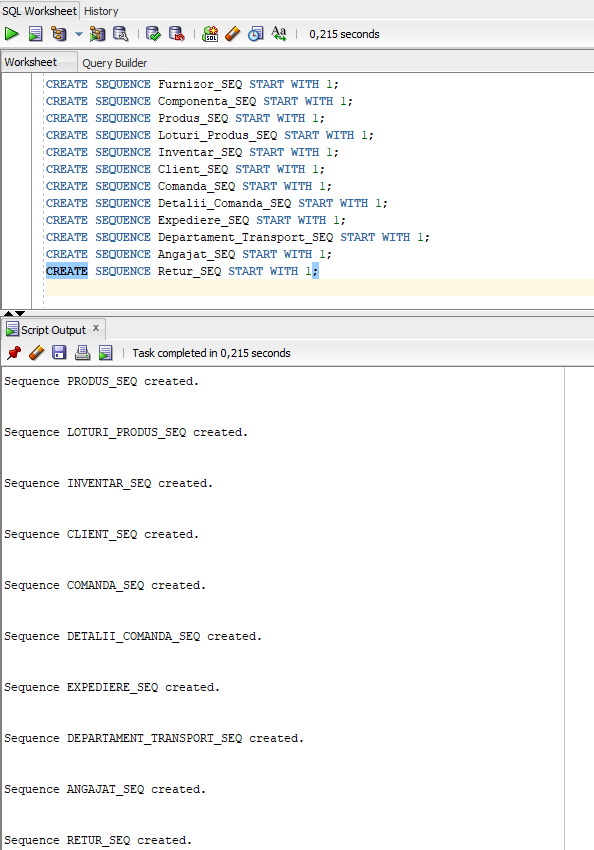
CREATE SEQUENCE Expediere\_SEQ START WITH 1;

CREATE SEQUENCE Departament\_Transport\_SEQ START WITH 1;

CREATE SEQUENCE Angajat\_SEQ START WITH 1;

CREATE SEQUENCE Retur\_SEQ START WITH 1;

PrintScreen cu execuția și cu rezultatul:



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).

Pentru tabelul **Furnizor**:

*Codul SQL:*

CREATE TABLE Furnizor (

ID\_Furnizor NUMBER DEFAULT Furnizor\_SEQ.NEXTVAL PRIMARY KEY,

Nume VARCHAR2(100),

Adresa VARCHAR2(255),

Numar\_Contact VARCHAR2(15),

Adresa\_Mail VARCHAR2(100)

);

INSERT INTO Furnizor (Nume, Adresa, Numar\_Contact, Adresa\_Mail) VALUES

('SC Sunet Premium SRL', 'Strada Muzicii, Nr. 10', '0722345678', 'contact@sunetpremium.com');

INSERT INTO Furnizor (Nume, Adresa, Numar\_Contact, Adresa\_Mail) VALUES

('Acoustic Solutions', 'Bulevardul Armoniei, Nr. 23', '0722345679', 'info@acousticsolutions.com');

INSERT INTO Furnizor (Nume, Adresa, Numar\_Contact, Adresa\_Mail) VALUES

('High Fidelity Acoustics', 'Calea Melodiei, Nr. 45', '0722345680', 'support@highfidelity.com');

INSERT INTO Furnizor (Nume, Adresa, Numar\_Contact, Adresa\_Mail) VALUES

('SC Amplify SRL', 'Strada Bass, Nr. 7', '0722345681', 'contact@amplify.com');

INSERT INTO Furnizor (Nume, Adresa, Numar\_Contact, Adresa\_Mail) VALUES

('VibraSound Inc.', 'Bulevardul Frecvențelor, Nr. 33', '0722345682', 'info@vibrasound.com');

INSERT INTO Furnizor (Nume, Adresa, Numar\_Contact, Adresa\_Mail) VALUES

('Pure Audio Systems', 'Aleea Decibelilor, Nr. 9', '0722345683', 'support@pureaudio.com');

INSERT INTO Furnizor (Nume, Adresa, Numar\_Contact, Adresa\_Mail) VALUES

('Waveform Electronics', 'Strada Rezonanței, Nr. 56', '0722345684', 'contact@waveform.com');

INSERT INTO Furnizor (Nume, Adresa, Numar\_Contact, Adresa\_Mail) VALUES

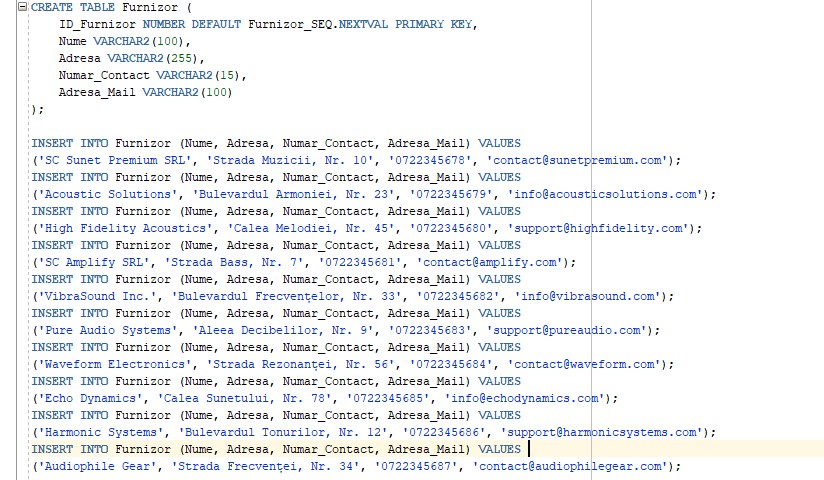
('Echo Dynamics', 'Calea Sunetului, Nr. 78', '0722345685', 'info@echodynamics.com');

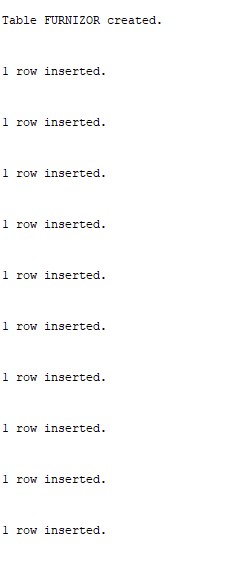
INSERT INTO Furnizor (Nume, Adresa, Numar\_Contact, Adresa\_Mail) VALUES

('Harmonic Systems', 'Bulevardul Tonurilor, Nr. 12', '0722345686', 'support@harmonicsystems.com');

INSERT INTO Furnizor (Nume, Adresa, Numar\_Contact, Adresa\_Mail) VALUES

('Audiophile Gear', 'Strada Frecvenței, Nr. 34', '0722345687', 'contact@audiophilegear.com');





Pentru tabelul **Produs**:

*Codul SQL:*

CREATE TABLE Produs (

ID\_Produs NUMBER DEFAULT Produs\_SEQ.NEXTVAL PRIMARY KEY,

Nume VARCHAR2(100),

Descriere CLOB,

Pret\_Vanzare NUMBER(10, 2),

Cost\_Productie NUMBER(10, 2)

);

INSERT INTO Produs (Nume, Descriere, Pret\_Vanzare, Cost\_Productie) VALUES

('EchoMaster', 'Difuzor de înaltă performanță, ideal pentru utilizare profesională.', 1200.00, 800.00);

INSERT INTO Produs (Nume, Descriere, Pret\_Vanzare, Cost\_Productie) VALUES

('SoundWave', 'Difuzor compact, perfect pentru sisteme audio casnice.', 800.00, 500.00);

INSERT INTO Produs (Nume, Descriere, Pret\_Vanzare, Cost\_Productie) VALUES

('BassBoom', 'Difuzor de medii, optimizat pentru claritatea sunetului.', 950.00, 600.00);

INSERT INTO Produs (Nume, Descriere, Pret\_Vanzare, Cost\_Productie) VALUES

('TrebleMax', 'Difuzor de bas cu răspuns rapid și profund.', 1300.00, 900.00);

INSERT INTO Produs (Nume, Descriere, Pret\_Vanzare, Cost\_Productie) VALUES

('PortablePro', 'Difuzor portabil, cu baterie încorporată și conectivitate Bluetooth.', 700.00, 400.00);

INSERT INTO Produs (Nume, Descriere, Pret\_Vanzare, Cost\_Productie) VALUES

('OutdoorX', 'Difuzor de exterior, rezistent la intemperii.', 1100.00, 700.00);

INSERT INTO Produs (Nume, Descriere, Pret\_Vanzare, Cost\_Productie) VALUES

('AllSound', 'Sistem audio all-in-one, cu amplificator integrat.', 1500.00, 1000.00);

INSERT INTO Produs (Nume, Descriere, Pret\_Vanzare, Cost\_Productie) VALUES

('MegaSound', 'Difuzor de coloană, pentru utilizare în spații mari.', 2000.00, 1300.00);

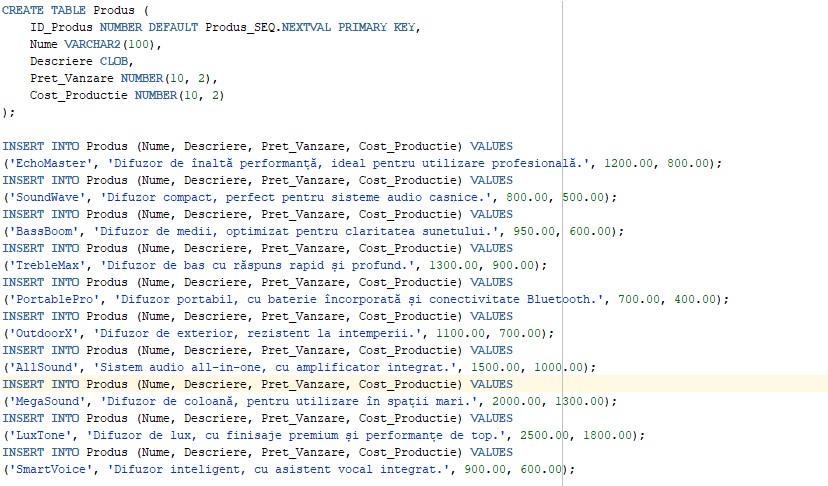
INSERT INTO Produs (Nume, Descriere, Pret\_Vanzare, Cost\_Productie) VALUES

('LuxTone', 'Difuzor de lux, cu finisaje premium și performanțe de top.', 2500.00, 1800.00);

INSERT INTO Produs (Nume, Descriere, Pret\_Vanzare, Cost\_Productie) VALUES

('SmartVoice', 'Difuzor inteligent, cu asistent vocal integrat.', 900.00, 600.00);

PrintScreen cu codul si cu rezultatul executiei:





Pentru tabelul **Componente**:

*Codul SQL:*

CREATE TABLE Componenta (

ID\_Componenta NUMBER DEFAULT Componenta\_SEQ.NEXTVAL PRIMARY KEY,

Nume VARCHAR2(100),

Tip VARCHAR2(50),

Pret NUMBER(10, 2),

ID\_Furnizor NUMBER,

ID\_Produs NUMBER,

CONSTRAINT fk\_furnizor FOREIGN KEY (ID\_Furnizor) REFERENCES Furnizor(ID\_Furnizor),

CONSTRAINT fk\_produs FOREIGN KEY (ID\_Produs) REFERENCES Produs(ID\_Produs)

);

INSERT INTO Componenta (Nume, Tip, Pret, ID\_Furnizor, ID\_Produs) VALUES

('Difuzor de înalte', 'Tweeter', 150.00, 1, 1);

INSERT INTO Componenta (Nume, Tip, Pret, ID\_Furnizor, ID\_Produs) VALUES

('Difuzor de medii', 'Midrange', 200.00, 2, 2);

INSERT INTO Componenta (Nume, Tip, Pret, ID\_Furnizor, ID\_Produs) VALUES

('Difuzor de bas', 'Woofer', 250.00, 3, 3);

INSERT INTO Componenta (Nume, Tip, Pret, ID\_Furnizor, ID\_Produs) VALUES

('Crossover pasiv', 'Crossover', 100.00, 4, 4);

INSERT INTO Componenta (Nume, Tip, Pret, ID\_Furnizor, ID\_Produs) VALUES

('Amplificator intern', 'Amplifier', 300.00, 5, 5);

INSERT INTO Componenta (Nume, Tip, Pret, ID\_Furnizor, ID\_Produs) VALUES

('Cablu de conexiune', 'Cable', 50.00, 1, 2);

INSERT INTO Componenta (Nume, Tip, Pret, ID\_Furnizor, ID\_Produs) VALUES

('Carcasă difuzor', 'Cabinet', 400.00, 2, 3);

INSERT INTO Componenta (Nume, Tip, Pret, ID\_Furnizor, ID\_Produs) VALUES

('Panou de control', 'Control Panel', 350.00, 3, 4);

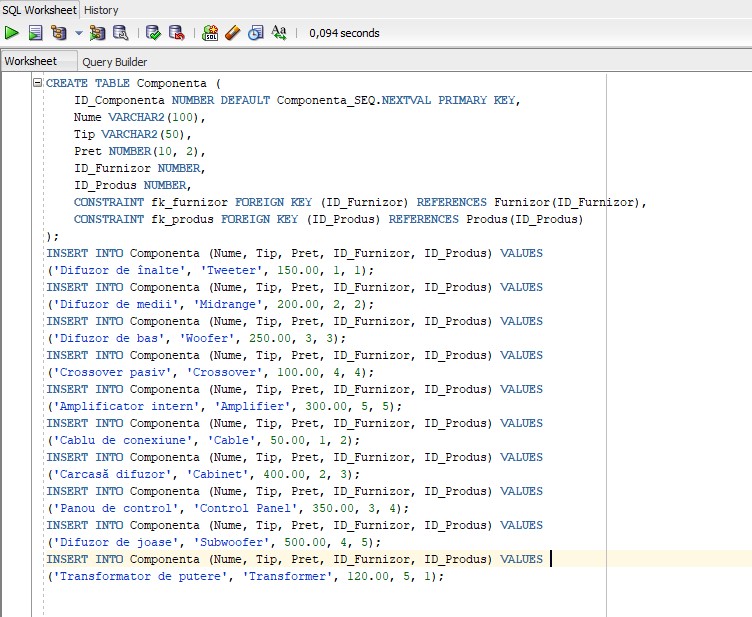
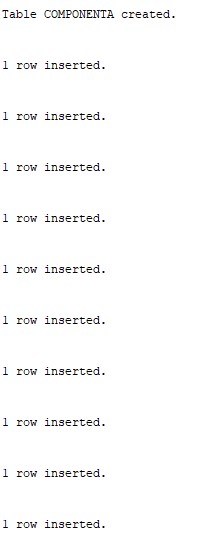
INSERT INTO Componenta (Nume, Tip, Pret, ID\_Furnizor, ID\_Produs) VALUES

('Difuzor de joase', 'Subwoofer', 500.00, 4, 5);

INSERT INTO Componenta (Nume, Tip, Pret, ID\_Furnizor, ID\_Produs) VALUES

('Transformator de putere', 'Transformer', 120.00, 5, 1);

*PrintScreen cu codul si cu rezultatul executiei lui:*



Pentru tabelul Loturi\_Produs:

*Codul SQL:*

CREATE TABLE Loturi\_Produs (

ID\_Lot NUMBER DEFAULT Loturi\_Produs\_SEQ.NEXTVAL PRIMARY KEY,

ID\_Produs NUMBER,

Data\_Incepere DATE,

Data\_Terminare DATE,

Cantitate\_Produsa NUMBER,

CONSTRAINT fk\_produs\_loturi\_produs FOREIGN KEY (ID\_Produs) REFERENCES Produs(ID\_Produs)

);

INSERT INTO Loturi\_Produs (ID\_Produs, Data\_Incepere, Data\_Terminare, Cantitate\_Produsa) VALUES

(1, TO\_DATE('01-01-2023', 'DD-MM-YYYY'), TO\_DATE('10-01-2023', 'DD-MM-YYYY'), 100);

INSERT INTO Loturi\_Produs (ID\_Produs, Data\_Incepere, Data\_Terminare, Cantitate\_Produsa) VALUES

(2, TO\_DATE('01-02-2023', 'DD-MM-YYYY'), TO\_DATE('10-02-2023', 'DD-MM-YYYY'), 150);

INSERT INTO Loturi\_Produs (ID\_Produs, Data\_Incepere, Data\_Terminare, Cantitate\_Produsa) VALUES

(3, TO\_DATE('01-03-2023', 'DD-MM-YYYY'), TO\_DATE('10-03-2023', 'DD-MM-YYYY'), 200);

INSERT INTO Loturi\_Produs (ID\_Produs, Data\_Incepere, Data\_Terminare, Cantitate\_Produsa) VALUES

(4, TO\_DATE('01-04-2023', 'DD-MM-YYYY'), TO\_DATE('10-04-2023', 'DD-MM-YYYY'), 250);

INSERT INTO Loturi\_Produs (ID\_Produs, Data\_Incepere, Data\_Terminare, Cantitate\_Produsa) VALUES

(5, TO\_DATE('01-05-2023', 'DD-MM-YYYY'), TO\_DATE('10-05-2023', 'DD-MM-YYYY'), 300);

INSERT INTO Loturi\_Produs (ID\_Produs, Data\_Incepere, Data\_Terminare, Cantitate\_Produsa) VALUES

(6, TO\_DATE('01-06-2023', 'DD-MM-YYYY'), TO\_DATE('10-06-2023', 'DD-MM-YYYY'), 350);

INSERT INTO Loturi\_Produs (ID\_Produs, Data\_Incepere, Data\_Terminare, Cantitate\_Produsa) VALUES

(7, TO\_DATE('01-07-2023', 'DD-MM-YYYY'), TO\_DATE('10-07-2023', 'DD-MM-YYYY'), 400);

INSERT INTO Loturi\_Produs (ID\_Produs, Data\_Incepere, Data\_Terminare, Cantitate\_Produsa) VALUES

(8, TO\_DATE('01-08-2023', 'DD-MM-YYYY'), TO\_DATE('10-08-2023', 'DD-MM-YYYY'), 450);

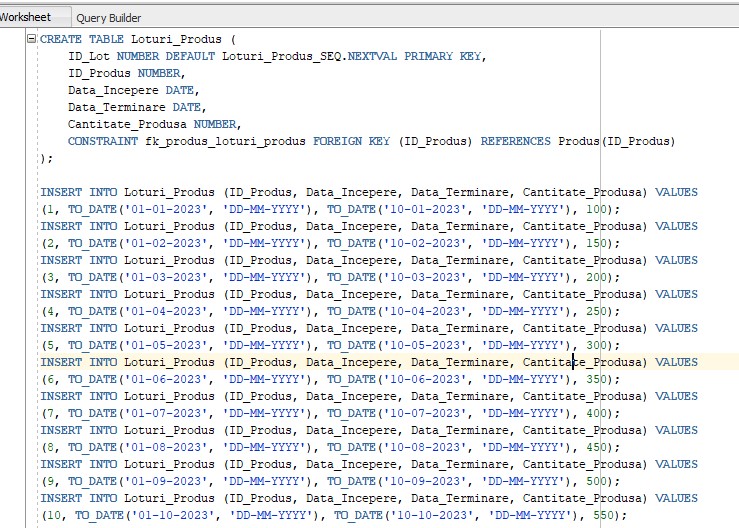
INSERT INTO Loturi\_Produs (ID\_Produs, Data\_Incepere, Data\_Terminare, Cantitate\_Produsa) VALUES

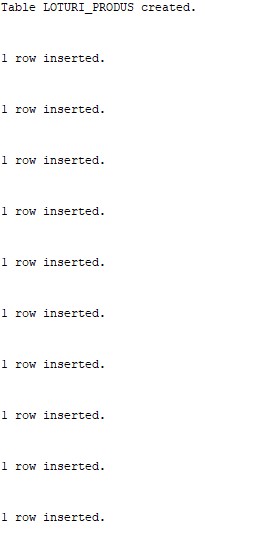
(9, TO\_DATE('01-09-2023', 'DD-MM-YYYY'), TO\_DATE('10-09-2023', 'DD-MM-YYYY'), 500);

INSERT INTO Loturi\_Produs (ID\_Produs, Data\_Incepere, Data\_Terminare, Cantitate\_Produsa) VALUES

(10, TO\_DATE('01-10-2023', 'DD-MM-YYYY'), TO\_DATE('10-10-2023', 'DD-MM-YYYY'), 550);

*PrintScreen cu codul si cu rezultatul executiei lui:*





Pentru tabelul **Inventar**:

*Codul SQL:*

CREATE TABLE Inventar (

ID\_Inventar NUMBER DEFAULT Inventar\_SEQ.NEXTVAL PRIMARY KEY,

ID\_Componenta NUMBER,

Cantitate\_Disponibila NUMBER,

Cantitate\_Comandata NUMBER,

CONSTRAINT fk\_componenta\_inventar FOREIGN KEY (ID\_Componenta) REFERENCES Componenta(ID\_Componenta)

);

INSERT INTO Inventar (ID\_Componenta, Cantitate\_Disponibila, Cantitate\_Comandata) VALUES

(1, 500, 100);

INSERT INTO Inventar (ID\_Componenta, Cantitate\_Disponibila, Cantitate\_Comandata) VALUES

(2, 600, 150);

INSERT INTO Inventar (ID\_Componenta, Cantitate\_Disponibila, Cantitate\_Comandata) VALUES

(3, 700, 200);

INSERT INTO Inventar (ID\_Componenta, Cantitate\_Disponibila, Cantitate\_Comandata) VALUES

(4, 800, 250);

INSERT INTO Inventar (ID\_Componenta, Cantitate\_Disponibila, Cantitate\_Comandata) VALUES

(5, 900, 300);

INSERT INTO Inventar (ID\_Componenta, Cantitate\_Disponibila, Cantitate\_Comandata) VALUES

(6, 400, 100);

INSERT INTO Inventar (ID\_Componenta, Cantitate\_Disponibila, Cantitate\_Comandata) VALUES

(7, 350, 150);

INSERT INTO Inventar (ID\_Componenta, Cantitate\_Disponibila, Cantitate\_Comandata) VALUES

(8, 450, 200);

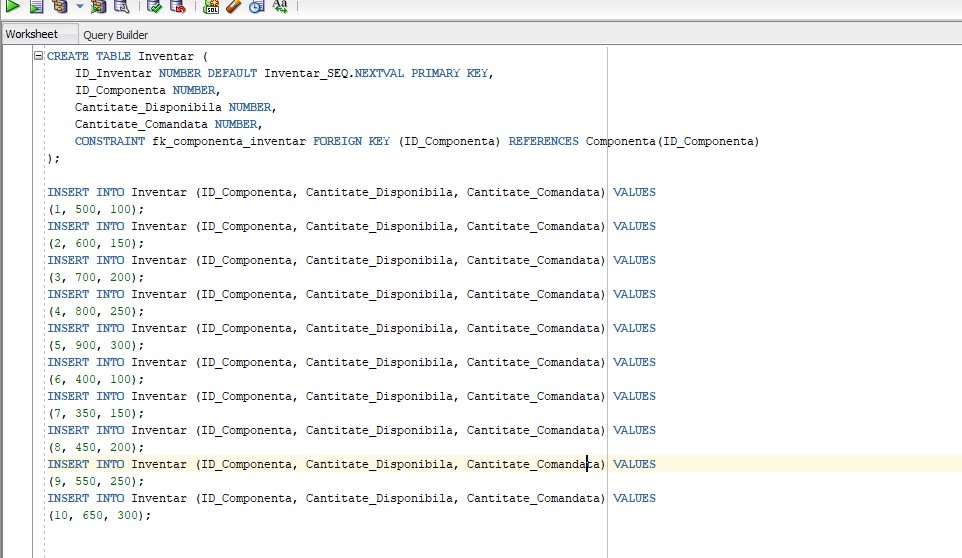
INSERT INTO Inventar (ID\_Componenta, Cantitate\_Disponibila, Cantitate\_Comandata) VALUES

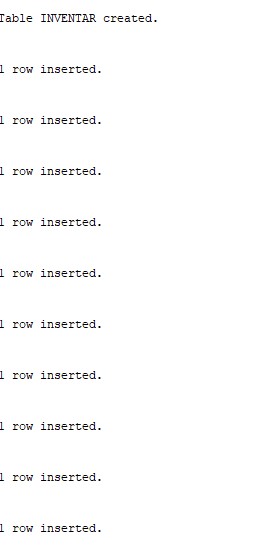
(9, 550, 250);

INSERT INTO Inventar (ID\_Componenta, Cantitate\_Disponibila, Cantitate\_Comandata) VALUES

(10, 650, 300);

*PrintScreen cu codul si cu rezultatul executiei lui:*





Pentru tabelul Client:

*Codul SQL:*

CREATE TABLE Client (

ID\_Client NUMBER DEFAULT Client\_SEQ.NEXTVAL PRIMARY KEY,

Numele\_Familie VARCHAR2(100),

Prenumele VARCHAR2(100),

Adresa VARCHAR2(255),

Cod\_Postal VARCHAR2(10),

Numar\_Telefon VARCHAR2(15),

Email VARCHAR2(100)

);

INSERT INTO Client (Numele\_Familie, Prenumele, Adresa, Cod\_Postal, Numar\_Telefon, Email) VALUES

('Popescu', 'Ion', 'Strada Fictiva, Nr. 10', '010101', '0712345678', 'ion.popescu@example.com');

INSERT INTO Client (Numele\_Familie, Prenumele, Adresa, Cod\_Postal, Numar\_Telefon, Email) VALUES

('Ionescu', 'Maria', 'Bulevardul Inventat, Nr. 23', '020202', '0723456789', 'maria.ionescu@example.com');

INSERT INTO Client (Numele\_Familie, Prenumele, Adresa, Cod\_Postal, Numar\_Telefon, Email) VALUES

('Georgescu', 'Andrei', 'Calea Imaginara, Nr. 45', '030303', '0734567890', 'andrei.georgescu@example.com');

INSERT INTO Client (Numele\_Familie, Prenumele, Adresa, Cod\_Postal, Numar\_Telefon, Email) VALUES

('Vasilescu', 'Ana', 'Strada Noua, Nr. 7', '040404', '0745678901', 'ana.vasilescu@example.com');

INSERT INTO Client (Numele\_Familie, Prenumele, Adresa, Cod\_Postal, Numar\_Telefon, Email) VALUES

('Dumitrescu', 'Dan', 'Aleea Fictiunii, Nr. 33', '050505', '0756789012', 'dan.dumitrescu@example.com');

INSERT INTO Client (Numele\_Familie, Prenumele, Adresa, Cod\_Postal, Numar\_Telefon, Email) VALUES

('Marinescu', 'Elena', 'Piata Inchipuita, Nr. 5', '060606', '0767890123', 'elena.marinescu@example.com');

INSERT INTO Client (Numele\_Familie, Prenumele, Adresa, Cod\_Postal, Numar\_Telefon, Email) VALUES

('Stanciu', 'Florin', 'Parcul Inexistent, Nr. 12', '070707', '0778901234', 'florin.stanciu@example.com');

INSERT INTO Client (Numele\_Familie, Prenumele, Adresa, Cod\_Postal, Numar\_Telefon, Email) VALUES

('Radu', 'Ioana', 'Bulevardul Fantaziei, Nr. 19', '080808', '0789012345', 'ioana.radu@example.com');

INSERT INTO Client (Numele\_Familie, Prenumele, Adresa, Cod\_Postal, Numar\_Telefon, Email) VALUES

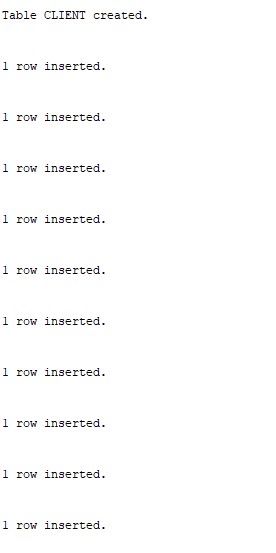
('Preda', 'Mihai', 'Strada Misterului, Nr. 21', '090909', '0790123456', 'mihai.preda@example.com');

INSERT INTO Client (Numele\_Familie, Prenumele, Adresa, Cod\_Postal, Numar\_Telefon, Email) VALUES

('Iliescu', 'Carmen', 'Aleea Iluziilor, Nr. 27', '100101', '0701234567', 'carmen.iliescu@example.com');

*PrintScreen cu codul si cu rezultatul executiei lui:*





Pentru tabelul **Comanda**:

*Codul SQL:*

CREATE TABLE Comanda (

ID\_Comanda NUMBER DEFAULT Comanda\_SEQ.NEXTVAL PRIMARY KEY,

ID\_Client NUMBER,

Data\_Comanda DATE,

Pret\_Total NUMBER(10, 2),

Status VARCHAR2(50),

CONSTRAINT fk\_client\_comanda FOREIGN KEY (ID\_Client) REFERENCES Client(ID\_Client)

);

INSERT INTO Comanda (ID\_Client, Data\_Comanda, Pret\_Total, Status) VALUES

(1, TO\_DATE('01-01-2023', 'DD-MM-YYYY'), 1000.00, 'procesată');

INSERT INTO Comanda (ID\_Client, Data\_Comanda, Pret\_Total, Status) VALUES

(2, TO\_DATE('02-01-2023', 'DD-MM-YYYY'), 1200.00, 'expediată');

INSERT INTO Comanda (ID\_Client, Data\_Comanda, Pret\_Total, Status) VALUES

(3, TO\_DATE('03-01-2023', 'DD-MM-YYYY'), 1400.00, 'finalizată');

INSERT INTO Comanda (ID\_Client, Data\_Comanda, Pret\_Total, Status) VALUES

(4, TO\_DATE('04-01-2023', 'DD-MM-YYYY'), 1600.00, 'procesată');

INSERT INTO Comanda (ID\_Client, Data\_Comanda, Pret\_Total, Status) VALUES

(5, TO\_DATE('05-01-2023', 'DD-MM-YYYY'), 1800.00, 'expediată');

INSERT INTO Comanda (ID\_Client, Data\_Comanda, Pret\_Total, Status) VALUES

(6, TO\_DATE('06-01-2023', 'DD-MM-YYYY'), 2000.00, 'finalizată');

INSERT INTO Comanda (ID\_Client, Data\_Comanda, Pret\_Total, Status) VALUES

(7, TO\_DATE('07-01-2023', 'DD-MM-YYYY'), 2200.00, 'procesată');

INSERT INTO Comanda (ID\_Client, Data\_Comanda, Pret\_Total, Status) VALUES

(8, TO\_DATE('08-01-2023', 'DD-MM-YYYY'), 2400.00, 'expediată');

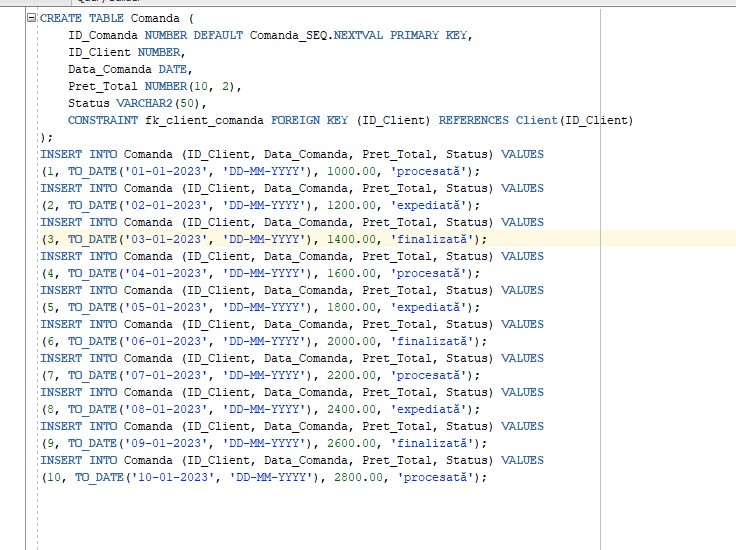
INSERT INTO Comanda (ID\_Client, Data\_Comanda, Pret\_Total, Status) VALUES

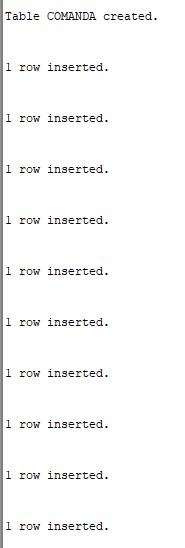
(9, TO\_DATE('09-01-2023', 'DD-MM-YYYY'), 2600.00, 'finalizată');

INSERT INTO Comanda (ID\_Client, Data\_Comanda, Pret\_Total, Status) VALUES

(10, TO\_DATE('10-01-2023', 'DD-MM-YYYY'), 2800.00, 'procesată');

*PrintScreen cu codul si cu rezultatul executiei lui:*





Pentru tabelul **Detalii\_Comanda**:

*Codul SQL:*

CREATE TABLE Detalii\_Comanda (

ID\_Detalii\_Comanda NUMBER DEFAULT Detalii\_Comanda\_SEQ.NEXTVAL PRIMARY KEY,

ID\_Comanda NUMBER,

ID\_Produs NUMBER,

Cantitate NUMBER,

Pret\_Per\_Unitate NUMBER(10, 2),

CONSTRAINT fk\_comanda\_detalii FOREIGN KEY (ID\_Comanda) REFERENCES Comanda(ID\_Comanda),

CONSTRAINT fk\_produs\_detalii FOREIGN KEY (ID\_Produs) REFERENCES Produs(ID\_Produs)

);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(1, 1, 10, 120.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(1, 2, 5, 160.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(2, 3, 8, 130.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(2, 4, 4, 110.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(3, 5, 12, 125.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(3, 1, 6, 100.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(4, 2, 9, 105.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(4, 3, 7, 150.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(5, 4, 10, 200.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(5, 5, 3, 250.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(6, 1, 5, 120.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(6, 2, 7, 160.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(7, 3, 9, 130.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(7, 4, 8, 110.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(8, 5, 6, 125.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(8, 1, 4, 100.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(9, 2, 3, 105.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(9, 3, 2, 150.00);

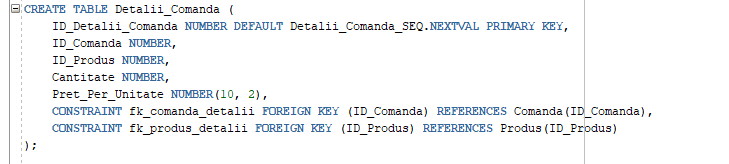
INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

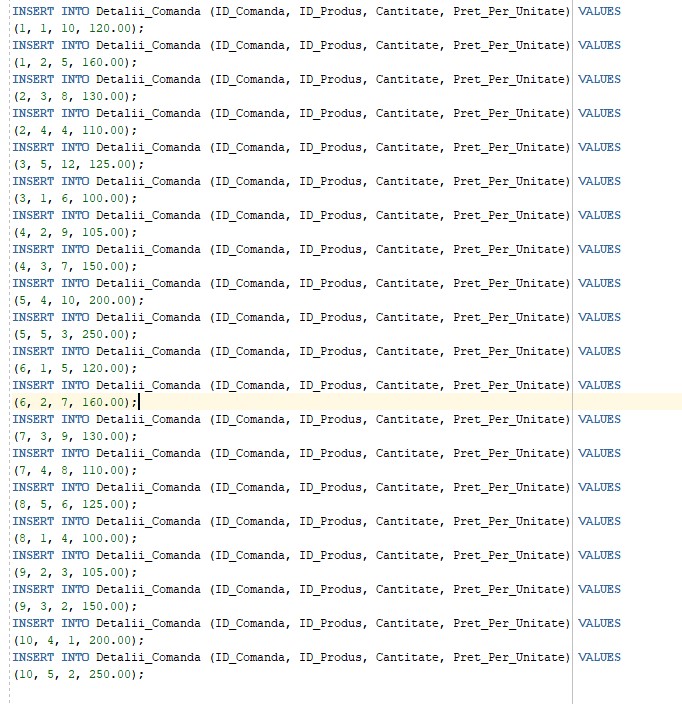
(10, 4, 1, 200.00);

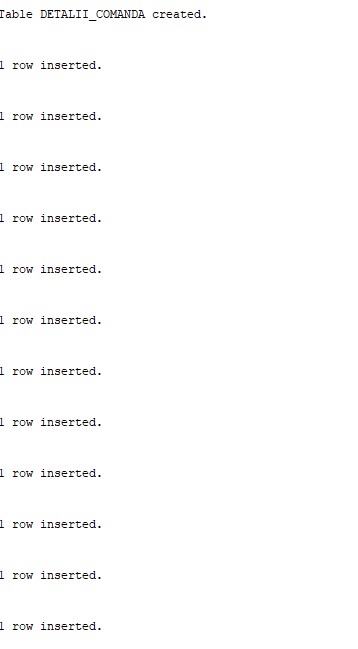
INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(10, 5, 2, 250.00);

*PrintScreen cu codul si cu rezultatul executiei lui:*

**





Pentru tabelul **Expediere**:

*Codul SQL:*

CREATE TABLE Expediere (

ID\_Expediere NUMBER DEFAULT Expediere\_SEQ.NEXTVAL PRIMARY KEY,

ID\_Comanda NUMBER,

Data\_Expediere DATE,

Data\_Estimata\_Livrare DATE,

ID\_Zona\_Transport NUMBER,

Status VARCHAR2(50),

CONSTRAINT fk\_comanda FOREIGN KEY (ID\_Comanda) REFERENCES Comanda(ID\_Comanda),

CONSTRAINT fk\_zona\_transport FOREIGN KEY (ID\_Zona\_Transport) REFERENCES Departament\_Transport(ID\_Zona\_Transport)

);

INSERT INTO Expediere (ID\_Comanda, Data\_Expediere, Data\_Estimata\_Livrare, ID\_Zona\_Transport, Status) VALUES

(1, TO\_DATE('11-01-2023', 'DD-MM-YYYY'), TO\_DATE('15-01-2023', 'DD-MM-YYYY'), 1, 'în curs de livrare');

INSERT INTO Expediere (ID\_Comanda, Data\_Expediere, Data\_Estimata\_Livrare, ID\_Zona\_Transport, Status) VALUES

(2, TO\_DATE('12-01-2023', 'DD-MM-YYYY'), TO\_DATE('16-01-2023', 'DD-MM-YYYY'), 2, 'livrat');

INSERT INTO Expediere (ID\_Comanda, Data\_Expediere, Data\_Estimata\_Livrare, ID\_Zona\_Transport, Status) VALUES

(3, TO\_DATE('13-01-2023', 'DD-MM-YYYY'), TO\_DATE('17-01-2023', 'DD-MM-YYYY'), 3, 'în curs de livrare');

INSERT INTO Expediere (ID\_Comanda, Data\_Expediere, Data\_Estimata\_Livrare, ID\_Zona\_Transport, Status) VALUES

(4, TO\_DATE('14-01-2023', 'DD-MM-YYYY'), TO\_DATE('18-01-2023', 'DD-MM-YYYY'), 4, 'livrat');

INSERT INTO Expediere (ID\_Comanda, Data\_Expediere, Data\_Estimata\_Livrare, ID\_Zona\_Transport, Status) VALUES

(5, TO\_DATE('15-01-2023', 'DD-MM-YYYY'), TO\_DATE('19-01-2023', 'DD-MM-YYYY'), 5, 'în curs de livrare');

INSERT INTO Expediere (ID\_Comanda, Data\_Expediere, Data\_Estimata\_Livrare, ID\_Zona\_Transport, Status) VALUES

(6, TO\_DATE('16-01-2023', 'DD-MM-YYYY'), TO\_DATE('20-01-2023', 'DD-MM-YYYY'), 1, 'livrat');

INSERT INTO Expediere (ID\_Comanda, Data\_Expediere, Data\_Estimata\_Livrare, ID\_Zona\_Transport, Status) VALUES

(7, TO\_DATE('17-01-2023', 'DD-MM-YYYY'), TO\_DATE('21-01-2023', 'DD-MM-YYYY'), 2, 'în curs de livrare');

INSERT INTO Expediere (ID\_Comanda, Data\_Expediere, Data\_Estimata\_Livrare, ID\_Zona\_Transport, Status) VALUES

(8, TO\_DATE('18-01-2023', 'DD-MM-YYYY'), TO\_DATE('22-01-2023', 'DD-MM-YYYY'), 3, 'livrat');

INSERT INTO Expediere (ID\_Comanda, Data\_Expediere, Data\_Estimata\_Livrare, ID\_Zona\_Transport, Status) VALUES

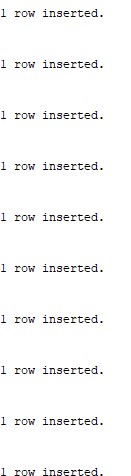
(9, TO\_DATE('19-01-2023', 'DD-MM-YYYY'), TO\_DATE('23-01-2023', 'DD-MM-YYYY'), 4, 'în curs de livrare');

INSERT INTO Expediere (ID\_Comanda, Data\_Expediere, Data\_Estimata\_Livrare, ID\_Zona\_Transport, Status) VALUES

(10, TO\_DATE('20-01-2023', 'DD-MM-YYYY'), TO\_DATE('24-01-2023', 'DD-MM-YYYY'), 5, 'livrat');

*PrintScreen cu codul si cu rezultatul executiei lui:*





Pentru tabelul **Departament\_Transport**:

*Codul SQL:*

CREATE TABLE Departament\_Transport (

ID\_Zona\_Transport NUMBER DEFAULT Departament\_Transport\_SEQ.NEXTVAL PRIMARY KEY,

Masina VARCHAR2(50)

);

INSERT INTO Departament\_Transport (Masina) VALUES

('Camion 1');

INSERT INTO Departament\_Transport (Masina) VALUES

('Camion 2');

INSERT INTO Departament\_Transport (Masina) VALUES

('Camion 3');

INSERT INTO Departament\_Transport (Masina) VALUES

('Camion 4');

INSERT INTO Departament\_Transport (Masina) VALUES

('Camion 5');

INSERT INTO Departament\_Transport (Masina) VALUES

('Furgonetă 1');

INSERT INTO Departament\_Transport (Masina) VALUES

('Furgonetă 2');

INSERT INTO Departament\_Transport (Masina) VALUES

('Furgonetă 3');

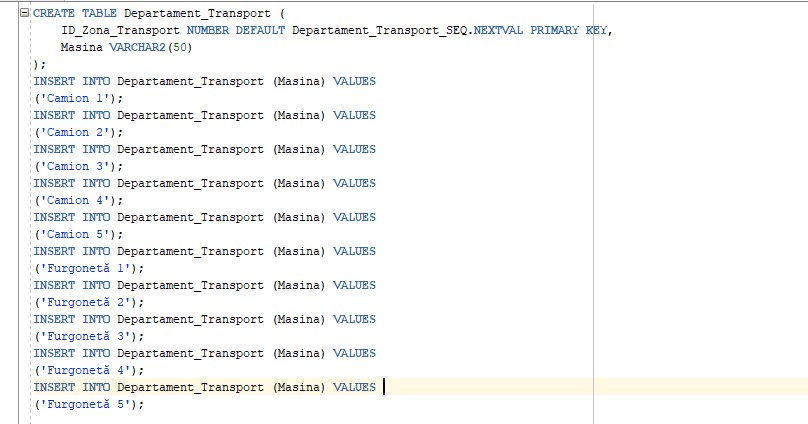
INSERT INTO Departament\_Transport (Masina) VALUES

('Furgonetă 4');

INSERT INTO Departament\_Transport (Masina) VALUES

('Furgonetă 5');

*PrintScreen cu codul si cu rezultatul executiei lui:*





Pentru tabelul Angajat:

*Codul SQL:*

CREATE TABLE Angajat (

ID\_Angajat NUMBER DEFAULT Angajat\_SEQ.NEXTVAL PRIMARY KEY,

Numele\_Familie VARCHAR2(100),

Prenumele VARCHAR2(100),

Bonus\_Lunar NUMBER(10, 2),

Salariu\_Lunar NUMBER(10, 2),

ID\_Zona\_Transport NUMBER,

CONSTRAINT fk\_zona\_transport\_angajat FOREIGN KEY (ID\_Zona\_Transport) REFERENCES Departament\_Transport(ID\_Zona\_Transport)

);

INSERT INTO Angajat (Numele\_Familie, Prenumele, Bonus\_Lunar, Salariu\_Lunar, ID\_Zona\_Transport) VALUES

('Popescu', 'Alex', 100.00, 3000.00, 1);

INSERT INTO Angajat (Numele\_Familie, Prenumele, Bonus\_Lunar, Salariu\_Lunar, ID\_Zona\_Transport) VALUES

('Ionescu', 'Cristina', 150.00, 3200.00, 2);

INSERT INTO Angajat (Numele\_Familie, Prenumele, Bonus\_Lunar, Salariu\_Lunar, ID\_Zona\_Transport) VALUES

('Georgescu', 'Mihai', 200.00, 3400.00, 3);

INSERT INTO Angajat (Numele\_Familie, Prenumele, Bonus\_Lunar, Salariu\_Lunar, ID\_Zona\_Transport) VALUES

('Vasilescu', 'Elena', 250.00, 3600.00, 4);

INSERT INTO Angajat (Numele\_Familie, Prenumele, Bonus\_Lunar, Salariu\_Lunar, ID\_Zona\_Transport) VALUES

('Dumitrescu', 'Adrian', 300.00, 3800.00, 5);

INSERT INTO Angajat (Numele\_Familie, Prenumele, Bonus\_Lunar, Salariu\_Lunar, ID\_Zona\_Transport) VALUES

('Marinescu', 'Raluca', 180.00, 3100.00, 6);

INSERT INTO Angajat (Numele\_Familie, Prenumele, Bonus\_Lunar, Salariu\_Lunar, ID\_Zona\_Transport) VALUES

('Stanciu', 'Florin', 170.00, 3300.00, 7);

INSERT INTO Angajat (Numele\_Familie, Prenumele, Bonus\_Lunar, Salariu\_Lunar, ID\_Zona\_Transport) VALUES

('Radu', 'Ioana', 190.00, 3500.00, 8);

INSERT INTO Angajat (Numele\_Familie, Prenumele, Bonus\_Lunar, Salariu\_Lunar, ID\_Zona\_Transport) VALUES

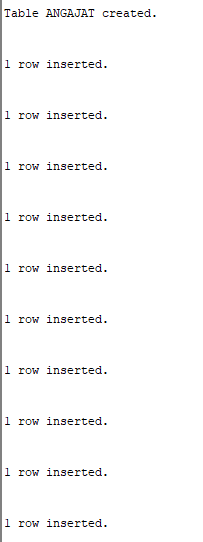
('Preda', 'Mihai', 220.00, 3700.00, 9);

INSERT INTO Angajat (Numele\_Familie, Prenumele, Bonus\_Lunar, Salariu\_Lunar, ID\_Zona\_Transport) VALUES

('Iliescu', 'Carmen', 210.00, 3900.00, 10);

*PrintScreen cu codul si cu rezultatul executiei lui:*





Pentru tabelul **Departament\_Transport**:

*Codul SQL:*

CREATE TABLE Departament\_Transport (

ID\_Zona\_Transport NUMBER DEFAULT Departament\_Transport\_SEQ.NEXTVAL PRIMARY KEY,

Masina VARCHAR2(50)

);

INSERT INTO Departament\_Transport (Masina) VALUES

('Camion 1');

INSERT INTO Departament\_Transport (Masina) VALUES

('Camion 2');

INSERT INTO Departament\_Transport (Masina) VALUES

('Camion 3');

INSERT INTO Departament\_Transport (Masina) VALUES

('Camion 4');

INSERT INTO Departament\_Transport (Masina) VALUES

('Camion 5');

INSERT INTO Departament\_Transport (Masina) VALUES

('Furgonetă 1');

INSERT INTO Departament\_Transport (Masina) VALUES

('Furgonetă 2');

INSERT INTO Departament\_Transport (Masina) VALUES

('Furgonetă 3');

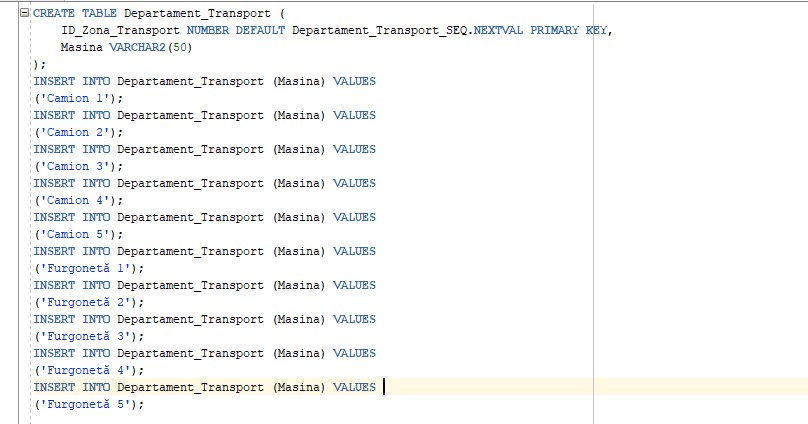
INSERT INTO Departament\_Transport (Masina) VALUES

('Furgonetă 4');

INSERT INTO Departament\_Transport (Masina) VALUES

('Furgonetă 5');

*PrintScreen cu codul si cu rezultatul executiei lui:*





Pentru tabelul **Retururi:**

*Codul SQL:*

CREATE TABLE Retur (

ID\_Retur NUMBER DEFAULT Retur\_SEQ.NEXTVAL PRIMARY KEY,

ID\_Comanda NUMBER,

ID\_Produs NUMBER,

Motiv VARCHAR2(255),

Data\_Retur DATE,

Actiune\_Finala VARCHAR2(50),

CONSTRAINT fk\_comanda\_retur FOREIGN KEY (ID\_Comanda) REFERENCES Comanda(ID\_Comanda),

CONSTRAINT fk\_produs\_retur FOREIGN KEY (ID\_Produs) REFERENCES Produs(ID\_Produs)

);

INSERT INTO Retur (ID\_Comanda, ID\_Produs, Motiv, Data\_Retur, Actiune\_Finala) VALUES

(1, 1, 'Produs defect', TO\_DATE('15-01-2023', 'DD-MM-YYYY'), 'rambursare');

INSERT INTO Retur (ID\_Comanda, ID\_Produs, Motiv, Data\_Retur, Actiune\_Finala) VALUES

(2, 2, 'Produs nepotrivit', TO\_DATE('16-01-2023', 'DD-MM-YYYY'), 'înlocuire');

INSERT INTO Retur (ID\_Comanda, ID\_Produs, Motiv, Data\_Retur, Actiune\_Finala) VALUES

(3, 3, 'Comandă greșită', TO\_DATE('17-01-2023', 'DD-MM-YYYY'), 'rambursare');

INSERT INTO Retur (ID\_Comanda, ID\_Produs, Motiv, Data\_Retur, Actiune\_Finala) VALUES

(4, 4, 'Calitate scăzută', TO\_DATE('18-01-2023', 'DD-MM-YYYY'), 'înlocuire');

INSERT INTO Retur (ID\_Comanda, ID\_Produs, Motiv, Data\_Retur, Actiune\_Finala) VALUES

(5, 5, 'Probleme tehnice', TO\_DATE('19-01-2023', 'DD-MM-YYYY'), 'rambursare');

INSERT INTO Retur (ID\_Comanda, ID\_Produs, Motiv, Data\_Retur, Actiune\_Finala) VALUES

(6, 1, 'Nu se potrivește', TO\_DATE('20-01-2023', 'DD-MM-YYYY'), 'înlocuire');

INSERT INTO Retur (ID\_Comanda, ID\_Produs, Motiv, Data\_Retur, Actiune\_Finala) VALUES

(7, 2, 'Defect la livrare', TO\_DATE('21-01-2023', 'DD-MM-YYYY'), 'rambursare');

INSERT INTO Retur (ID\_Comanda, ID\_Produs, Motiv, Data\_Retur, Actiune\_Finala) VALUES

(8, 3, 'Greșeală de comandă', TO\_DATE('22-01-2023', 'DD-MM-YYYY'), 'înlocuire');

INSERT INTO Retur (ID\_Comanda, ID\_Produs, Motiv, Data\_Retur, Actiune\_Finala) VALUES

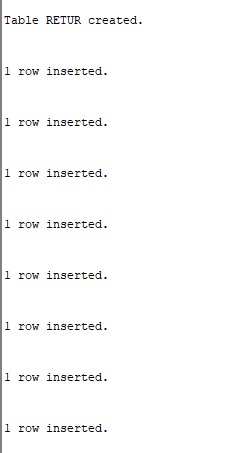
(9, 4, 'Probleme de funcționare', TO\_DATE('23-01-2023', 'DD-MM-YYYY'), 'rambursare');

INSERT INTO Retur (ID\_Comanda, ID\_Produs, Motiv, Data\_Retur, Actiune\_Finala) VALUES

(10, 5, 'Nu îndeplinește așteptările', TO\_DATE('24-01-2023', 'DD-MM-YYYY'), 'înlocuire');

*PrintScreen cu codul si cu rezultatul executiei lui:*





Pentru tabelul **Furnizeaza** (asociativ)**:**

*Codul SQL:*

CREATE TABLE Furnizeaza (

ID\_Furnizor NUMBER,

ID\_Componenta NUMBER,

Cantitate NUMBER,

CONSTRAINT fk\_furnizeaza\_furnizor FOREIGN KEY (ID\_Furnizor) REFERENCES Furnizor(ID\_Furnizor),

CONSTRAINT fk\_furnizeaza\_componenta FOREIGN KEY (ID\_Componenta) REFERENCES Componenta(ID\_Componenta),

PRIMARY KEY (ID\_Furnizor, ID\_Componenta)

);

INSERT INTO Furnizeaza (ID\_Furnizor, ID\_Componenta, Cantitate) VALUES

(1, 1, 100);

INSERT INTO Furnizeaza (ID\_Furnizor, ID\_Componenta, Cantitate) VALUES

(2, 2, 150);

INSERT INTO Furnizeaza (ID\_Furnizor, ID\_Componenta, Cantitate) VALUES

(3, 3, 200);

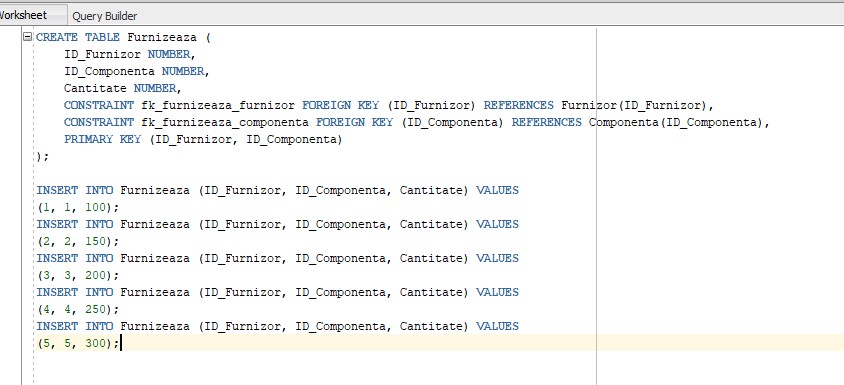
INSERT INTO Furnizeaza (ID\_Furnizor, ID\_Componenta, Cantitate) VALUES

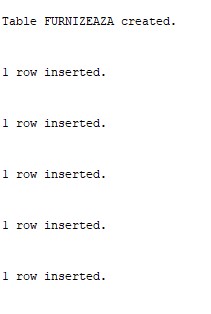
(4, 4, 250);

INSERT INTO Furnizeaza (ID\_Furnizor, ID\_Componenta, Cantitate) VALUES

(5, 5, 300);

*PrintScreen cu codul si cu rezultatul executiei lui:*





Pentru tabelul **Plaseaza** (asociativ)**:**

*Codul SQL:*

CREATE TABLE Plaseaza (

ID\_Client NUMBER,

ID\_Comanda NUMBER,

Data\_Comanda DATE,

CONSTRAINT fk\_plaseaza\_client FOREIGN KEY (ID\_Client) REFERENCES Client(ID\_Client),

CONSTRAINT fk\_plaseaza\_comanda FOREIGN KEY (ID\_Comanda) REFERENCES Comanda(ID\_Comanda),

PRIMARY KEY (ID\_Client, ID\_Comanda)

);

INSERT INTO Plaseaza (ID\_Client, ID\_Comanda, Data\_Comanda) VALUES

(1, 1, TO\_DATE('01-01-2023', 'DD-MM-YYYY'));

INSERT INTO Plaseaza (ID\_Client, ID\_Comanda, Data\_Comanda) VALUES

(2, 2, TO\_DATE('02-01-2023', 'DD-MM-YYYY'));

INSERT INTO Plaseaza (ID\_Client, ID\_Comanda, Data\_Comanda) VALUES

(3, 3, TO\_DATE('03-01-2023', 'DD-MM-YYYY'));

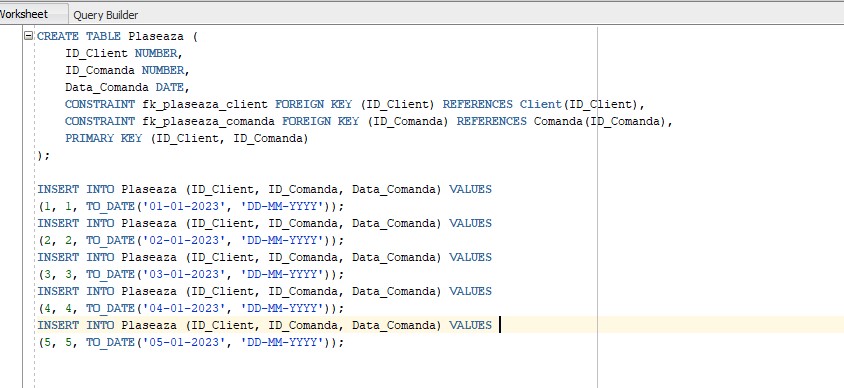
INSERT INTO Plaseaza (ID\_Client, ID\_Comanda, Data\_Comanda) VALUES

(4, 4, TO\_DATE('04-01-2023', 'DD-MM-YYYY'));

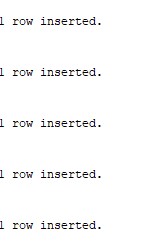
INSERT INTO Plaseaza (ID\_Client, ID\_Comanda, Data\_Comanda) VALUES

(5, 5, TO\_DATE('05-01-2023', 'DD-MM-YYYY'));

*PrintScreen cu codul si cu rezultatul executiei lui:*







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

a)subcereri sincronizate în care intervin cel puțin 3 tabele

b)subcereri nesincronizate în clauza FROM

c)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)

d)ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)

e) utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE

f) utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

**1. Cerere SQL cu subcereri sincronizate în care intervin cel puțin 3 tabele**

Formulare în limbaj natural:

Selectați numele și prenumele clienților care au plasat comenzi în anul 2023, împreună cu numărul total de produse comandate și suma totală plătită pentru aceste produse. Rezultatele vor fi ordonate descrescător după suma totală plătită.

Implementare:

sql

SELECT

C.Numele\_Familie || ' ' || C.Prenumele AS Numele\_Complet,

COUNT(D.ID\_Produs) AS Total\_Produse\_Comandate,

SUM(D.Cantitate \* D.Pret\_Per\_Unitate) AS Suma\_Totala

FROM

Client C

JOIN Comanda O ON C.ID\_Client = O.ID\_Client

JOIN Detalii\_Comanda D ON O.ID\_Comanda = D.ID\_Comanda

WHERE

EXTRACT(YEAR FROM O.Data\_Comanda) = 2023

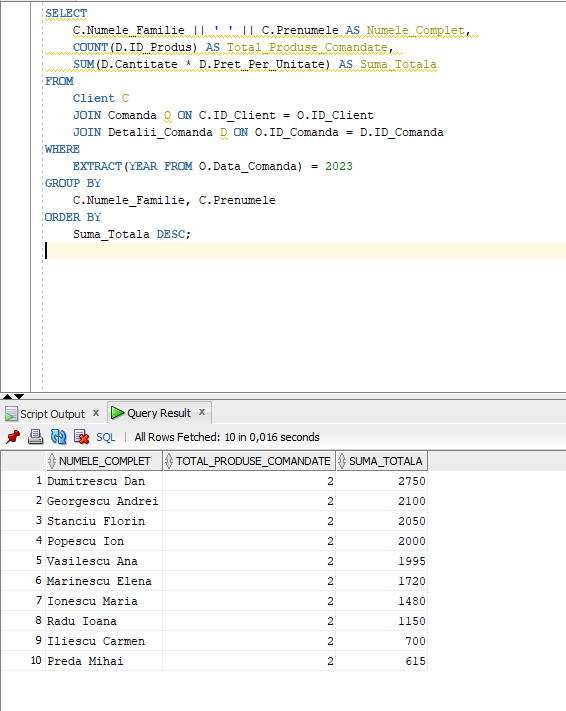
GROUP BY

C.Numele\_Familie, C.Prenumele

ORDER BY

Suma\_Totala DESC;

*PrintScreen cu codul rulat si cu rezultatul:*

**

**2. Cerere SQL cu subcereri nesincronizate în clauza FROM**

Formulare în limbaj natural:

Selectați produsele care au fost comandate cel puțin o dată, împreună cu numărul total de comenzi pentru fiecare produs și suma totală generată din vânzarea acestora. Rezultatele vor fi ordonate descrescător după numărul total de comenzi.

Implementare:

SELECT

P.Nume AS Nume\_Produs,

COALESCE(Comenzi\_Total, 0) AS Total\_Comenzi,

COALESCE(Vanzari\_Total, 0) AS Total\_Vanzari

FROM

Produs P

LEFT JOIN (

SELECT

D.ID\_Produs,

COUNT(D.ID\_Comanda) AS Comenzi\_Total,

SUM(D.Cantitate \* D.Pret\_Per\_Unitate) AS Vanzari\_Total

FROM

Detalii\_Comanda D

GROUP BY

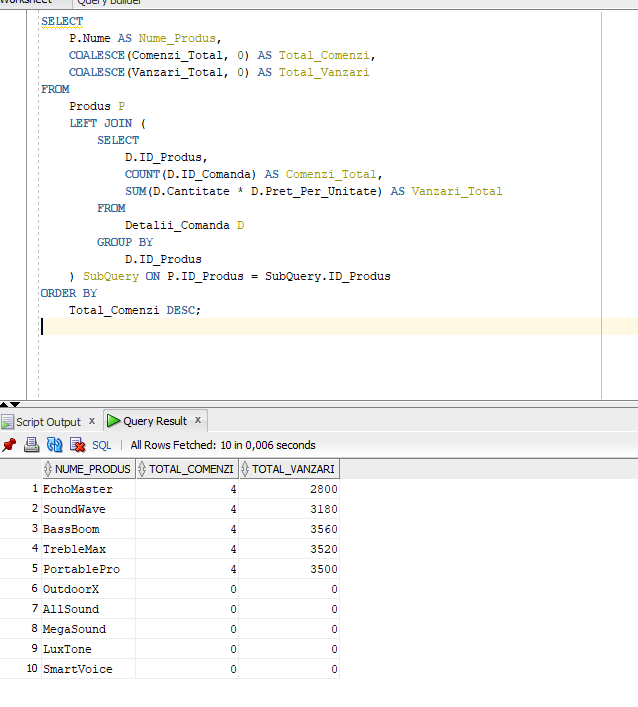
D.ID\_Produs

) SubQuery ON P.ID\_Produs = SubQuery.ID\_Produs

ORDER BY

Total\_Comenzi DESC;

*PrintScreen cu codul rulat si cu rezultatul:*



**3. Cerere SQL cu grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate în clauza de HAVING în care intervin cel puțin 3 tabele**

Formulare în limbaj natural:

Selectați furnizorii care au furnizat componente pentru cel puțin 2 produse diferite și a căror cantitate totală furnizată este mai mare decât media cantităților furnizate de toți furnizorii. Se vor afișa numele furnizorului și cantitatea totală furnizată, ordonate descrescător după cantitate.

Implementare:

SELECT

F.Nume AS Nume\_Furnizor,

SUM(FZ.Cantitate) AS Cantitate\_Totala

FROM

Furnizor F

JOIN Furnizeaza FZ ON F.ID\_Furnizor = FZ.ID\_Furnizor

JOIN Componenta C ON FZ.ID\_Componenta = C.ID\_Componenta

GROUP BY

F.Nume

HAVING

COUNT(DISTINCT C.ID\_Produs) >= 2

AND SUM(FZ.Cantitate) > (

SELECT

AVG(SubQuery.Cantitate\_Totala)

FROM (

SELECT

SUM(FZ2.Cantitate) AS Cantitate\_Totala

FROM

Furnizeaza FZ2

GROUP BY

FZ2.ID\_Furnizor

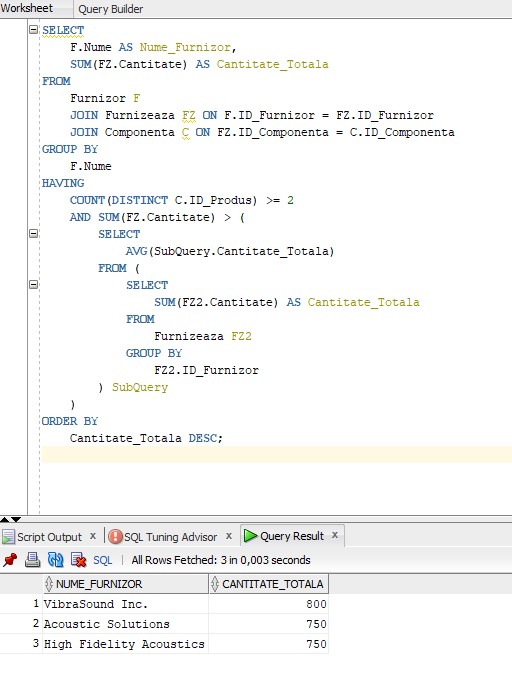
) SubQuery

)

ORDER BY

Cantitate\_Totala DESC;

*PrintScreen cu codul rulat si cu rezultatul:*



**4. Cerere SQL cu ordonări și utilizarea funcțiilor NVL și DECODE**

Formulare în limbaj natural:

Selectați comenzile și statusul acestora, înlocuind valorile null cu "Necunoscut" și codificând statusul "procesată" ca 1, "expediată" ca 2 și alte statusuri ca 0. Rezultatele vor fi ordonate după ID-ul comenzii.

Implementare:

SELECT

O.ID\_Comanda,

O.Data\_Comanda,

NVL(O.Status, 'Necunoscut') AS Status\_Comanda,

DECODE(O.Status, 'procesată', 1, 'expediată', 2, 0) AS Cod\_Status

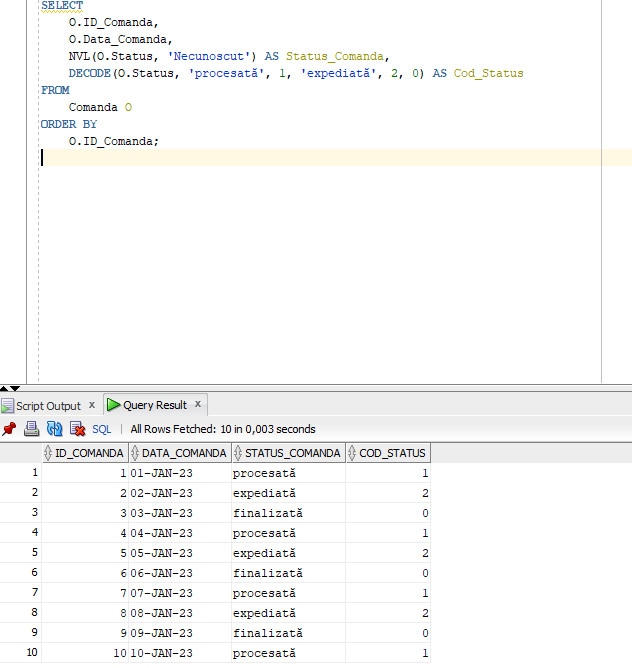
FROM

Comanda O

ORDER BY

O.ID\_Comanda;

*PrintScreen cu codul rulat si cu rezultatul:*



**5. Cerere SQL cu funcții pe șiruri de caractere, funcții pe date calendaristice și expresii CASE, utilizarea unui bloc de cerere (clauza WITH)**

Formulare în limbaj natural:

Selectați clienții care au plasat comenzi în ultimul an, formatând numele complet în formatul "Nume, Prenume" și data comenzii în formatul "DD-Month-YYYY". Calculați vechimea comenzii pe baza datei comenzii și clasificați comenzile în funcție de vechime (< 30 zile, 30-60 zile, > 60 zile).Implementare:

WITH Recent\_Comenzi AS (

SELECT

C.ID\_Client,

C.Numele\_Familie,

C.Prenumele,

O.Data\_Comanda

FROM

Client C

JOIN Comanda O ON C.ID\_Client = O.ID\_Client

WHERE

O.Data\_Comanda >= ADD\_MONTHS(SYSDATE, -12)

)

SELECT

C.Numele\_Familie || ', ' || C.Prenumele AS Nume\_Complet,

TO\_CHAR(C.Data\_Comanda, 'DD-Month-YYYY') AS Data\_Comanda\_Format,

ROUND((SYSDATE - C.Data\_Comanda)) AS Zile\_De\_La\_Comanda,

CASE

WHEN ROUND((SYSDATE - C.Data\_Comanda)) < 30 THEN '< 30 zile'

WHEN ROUND((SYSDATE - C.Data\_Comanda)) BETWEEN 30 AND 60 THEN '30-60 zile'

ELSE '> 60 zile'

END AS Categorie\_Vechime\_Comanda

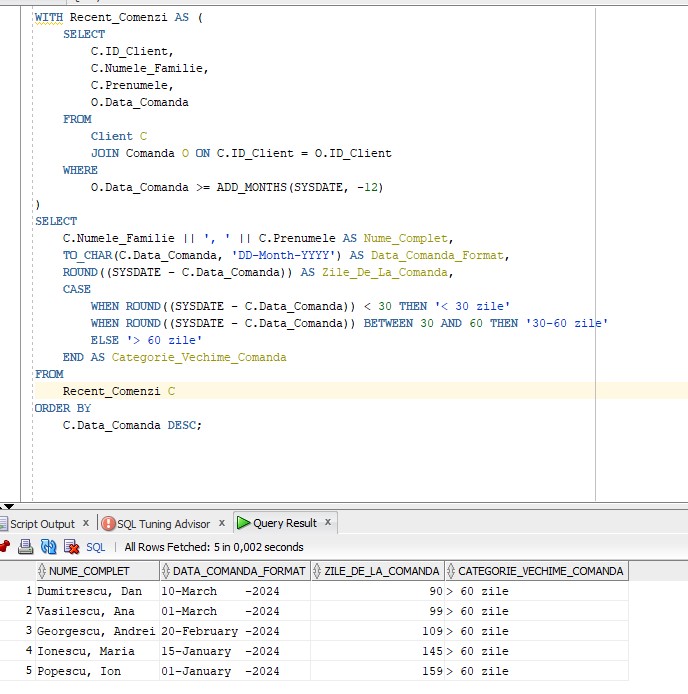
FROM

Recent\_Comenzi C

ORDER BY

C.Data\_Comanda DESC;

*PrintScreen cu codul rulat si cu rezultatul:*



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

1. Actualizarea datelor: Reducerea prețului produselor care au stoc mare în inventar.

Formulare în limbaj natural:

Reduceți prețul de vânzare al produselor cu 10% dacă cantitatea disponibilă în inventar depășește 500 de unități.

Implementare SQL:

UPDATE Produs

SET Pret\_Vanzare = Pret\_Vanzare \* 0.90

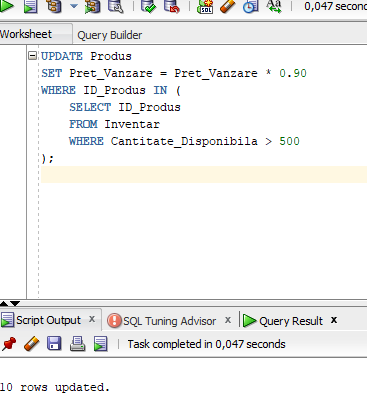
WHERE ID\_Produs IN (

SELECT ID\_Produs

FROM Inventar

WHERE Cantitate\_Disponibila > 500

);



2. Actualizarea datelor: Schimbarea statusului comenzilor la "finalizată" pentru comenzile care au toate produsele livrate.

Formulare în limbaj natural:

Actualizați statusul comenzilor la "finalizată" dacă toate produsele din comandă au fost livrate.

Implementare SQL:

UPDATE Comanda

SET Status = 'finalizată'

WHERE ID\_Comanda IN (

SELECT ID\_Comanda

FROM Detalii\_Comanda DC

WHERE NOT EXISTS (

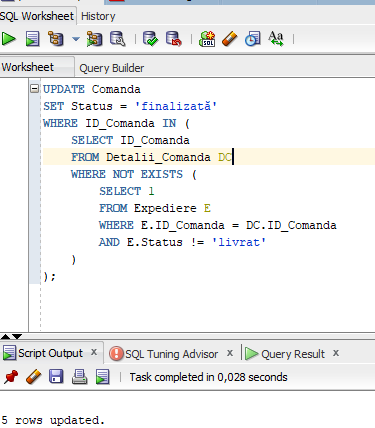
SELECT 1

FROM Expediere E

WHERE E.ID\_Comanda = DC.ID\_Comanda

AND E.Status != 'livrat'

)

);

3. Actualizarea datelor: Reducerea prețului produselor care au stoc mare în inventar.

Formulare în limbaj natural:

Reduceți prețul de vânzare al produselor cu 10% dacă cantitatea disponibilă în inventar depășește 500 de unități.

Implementare SQL:

UPDATE Produs

SET Pret\_Vanzare = Pret\_Vanzare \* 0.90

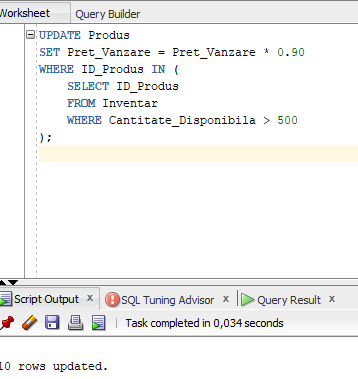
WHERE ID\_Produs IN (

SELECT ID\_Produs

FROM Inventar

WHERE Cantitate\_Disponibila > 500

);



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

Crearea vizualizării complexe

Formulare în limbaj natural:

Creați o vizualizare care să conțină informații despre comenzile clienților, inclusiv detalii despre produse, starea expedițiilor și detalii despre clienți.

Implementare SQL:

CREATE OR REPLACE VIEW V\_Complexa AS

SELECT

C.ID\_Comanda,

CL.Numele\_Familie || ' ' || CL.Prenumele AS Nume\_Client,

CL.Adresa,

CL.Numar\_Telefon,

P.Nume AS Nume\_Produs,

P.Pret\_Vanzare,

E.Data\_Expediere,

E.Status AS Status\_Expediere,

DC.Cantitate,

DC.Pret\_Per\_Unitate,

(DC.Cantitate \* DC.Pret\_Per\_Unitate) AS Total\_Produs

FROM

Comanda C

JOIN Client CL ON C.ID\_Client = CL.ID\_Client

JOIN Detalii\_Comanda DC ON C.ID\_Comanda = DC.ID\_Comanda

JOIN Produs P ON DC.ID\_Produs = P.ID\_Produs

LEFT JOIN Expediere E ON C.ID\_Comanda = E.ID\_Comanda;

Operație LMD permisă: Actualizarea statusului expedițiilor

Formulare în limbaj natural:

Actualizați statusul expediției pentru o anumită comandă.

Implementare SQL:

UPDATE Expediere

SET Status = 'livrat'

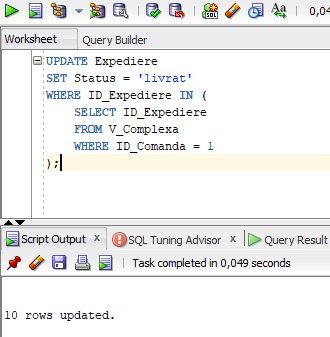
WHERE ID\_Expediere IN (

SELECT ID\_Expediere

FROM V\_Complexa

WHERE ID\_Comanda = 1

);



Operație LMD nepermisă: Inserarea unui nou rând în vizualizare

Formulare în limbaj natural:

Încercați să inserați o nouă înregistrare în vizualizarea complexă.

Implementare SQL:

INSERT INTO V\_Complexa (ID\_Comanda, Nume\_Client, Adresa, Numar\_Telefon, Nume\_Produs, Pret\_Vanzare, Data\_Expediere, Status\_Expediere, Cantitate, Pret\_Per\_Unitate, Total\_Produs)

VALUES (11, 'Popescu Mihai', 'Strada Noua, Nr. 1', '0712345678', 'EchoMaster', 1200, TO\_DATE('15-01-2023', 'DD-MM-YYYY'), 'în curs de livrare', 5, 120, 600);

