ACADEMIA DE STUDII ECONOMICE DIN BUCUREȘTI

FACULTATEA DE CIBERNETICĂ, STATISTICĂ SI INFORMATICĂ ECONOMICĂ

SPECIALIZAREA DE INFORMATICĂ ECONOMICĂ

**PROIECT – BAZE DE DATE**

PROFESOR COORDONATOR:

BOTHA IULIANA

STUDENT:

SIMIONESCU CORINA ANDREEA

BUCUREȘTI

2023-2024

Table of Content

[**I.**](#_heading=h.gjdgxs) **DESCRIEREA PROBLEMEI ȘI SCHEMA CONCEPTUALĂ** 4

[**1. Descrierea problemei alese** 4](#_heading=h.30j0zll)

[**2. Schema conceptuală a bazei de date** 5](#_heading=h.1fob9te)

[**II.**](#_heading=h.3znysh7) **DEFINIREA SCHEMEI BAZEI DE DATE** 6

[**1. Implementarea operaţiilor de definire a datelor: CREATE, ALTER, DROP** 6](#_heading=h.2et92p0)

[**2. Schema conceptuală a bazei de date generată de Oracle SQL Developer** 13](#_heading=h.tyjcwt)

[**III.**](#_heading=h.3dy6vkm) **IMPLEMENTAREA OPERAȚIILOR DE ACTUALIZARE A DATELOR** 14

[**1. INSERT** 14](#_heading=h.1t3h5sf)

[**1.1 Datele inserate in tabele** 23](#_heading=h.4d34og8)

[**2. UPDATE** 26](#_heading=h.2s8eyo1)

[**3. DELETE** 26](#_heading=h.17dp8vu)

[**IV.**](#_heading=h.3rdcrjn) **IMPLEMENTAREA INTEROGĂRILOR** 27

[**1. JONCȚIUNI** 27](#_heading=h.26in1rg)

[**1.1. INNER JOIN** 27](#_heading=h.lnxbz9)

[**1.2. OUTER JOIN** 28](#_heading=h.35nkun2)

[**1.3. LEFT JOIN** 29](#_heading=h.1ksv4uv)

[**1.4. RIGHT JOIN** 30](#_heading=h.44sinio)

[**2. FUNCȚII DE GRUP** 31](#_heading=h.2jxsxqh)

[**2.1. COUNT** 31](#_heading=h.z337ya)

[**2.2. SUM** 32](#_heading=h.3j2qqm3)

[**2.3. MIN** 34](#_heading=h.1y810tw)

[**2.4. MAX** 35](#_heading=h.4i7ojhp)

[**2.5. AVG** 35](#_heading=h.2xcytpi)

[**3. FUNCȚII SINGLE-ROW** 36](#_heading=h.1ci93xb)

[**3.1. FUNCȚII NUMERICE** 36](#_heading=h.3whwml4)

[**3.1.1. ROUND** 36](#_heading=h.2bn6wsx)

[**3.1.2. TRUNC** 36](#_heading=h.qsh70q)

[**3.2. FUNCTII DE CARACTERE** 37](#_heading=h.3as4poj)

[**3.2.1. LOWER** 37](#_heading=h.1pxezwc)

[**3.2.2. UPPER** 37](#_heading=h.49x2ik5)

[**3.2.3. INITCAP** 38](#_heading=h.2p2csry)

[**3.2.4. OPERATORUL DE CONCATENARE: ||** 39](#_heading=h.147n2zr)

[**3.2.5. SUBSTR** 39](#_heading=h.3o7alnk)

[**3.2.6. CONCAT SI LENGTH** 40](#_heading=h.23ckvvd)

[**3.3. FUNCȚII DE TIP DATĂ CALENDARISTICĂ** 41](#_heading=h.ihv636)

[**3.3.1. SYSDATE** 41](#_heading=h.32hioqz)

[**3.3.2. MONTH\_BETWEEN() , NEXT\_DAY() , LAST\_DAY(), ADD\_MONTHS()** 41](#_heading=h.1hmsyys)

[**4. REALIZAREA INTEROGĂRILOR PE BAZA UNOR CONDIȚII** 42](#_heading=h.41mghml)

[**4.1. DECODE** 42](#_heading=h.2grqrue)

[**4.2. CASE** 43](#_heading=h.vx1227)

[**5. OPERATORII ALGEBREI RELAȚIONALE** 44](#_heading=h.3fwokq0)

[**5.1. MINUS** 44](#_heading=h.1v1yuxt)

[**5.2. INTERSECT** 45](#_heading=h.4f1mdlm)

[**5.3. UNION** 45](#_heading=h.2u6wntf)

[**6. SUBCERERI** 46](#_heading=h.19c6y18)

[**V.**](#_heading=h.3tbugp1) **GESTIUNEA ALTOR OBIECTE ALE BAZEI DE DATE** 50

[**1. TABELE VIRTUALE** 50](#_heading=h.28h4qwu)

[**1.1. CREARE TABELĂ VIRTUALĂ** 50](#_heading=h.nmf14n)

[**1.2. INTEROGARE TABELĂ VIRTUALĂ** 52](#_heading=h.37m2jsg)

[**1.3. STERGERE TABELĂ VIRTUALĂ** 53](#_heading=h.1mrcu09)

[**2. INDECȘI** 53](#_heading=h.46r0co2)

[**2.1. CREARE INDEX** 53](#_heading=h.2lwamvv)

[**2.2. VIZUALIZARE INDECȘI** 54](#_heading=h.111kx3o)

[**2.3. EFICIENTA INDECȘILOR** 54](#_heading=h.3l18frh)

[**2.4. STERGERE INDECȘI** 56](#_heading=h.206ipza)

[**3. SINONIME** 56](#_heading=h.4k668n3)

[**3.1. CREARE SINONIM** 56](#_heading=h.2zbgiuw)

[**3.2. UTILIZAREA SINONIMULUI CREAT** 57](#_heading=h.1egqt2p)

[**3.3. VIZUALIZARE SINONIME** 57](#_heading=h.3ygebqi)

[**3.4. STERGERE SINONIM** 57](#_heading=h.2dlolyb)

[**4. SECVENȚE** 58](#_heading=h.sqyw64)

[**4.1. CREARE SECVENȚĂ** 58](#_heading=h.3cqmetx)

[**4.2. UTILIZAREA SECVENȚEI CREATE** 58](#_heading=h.1rvwp1q)

[**4.3. AFISARE VALOARE CURENTĂ A SECVENȚEI** 59](#_heading=h.4bvk7pj)

[**4.4. STERGERE SECVENȚĂ** 60](#_heading=h.2r0uhxc)

# **DESCRIEREA PROBLEMEI ȘI SCHEMA CONCEPTUALĂ**

## **1. Descrierea problemei alese**

Baza de date este concepută pentru a gestiona informațiile unui site de vânzare de laptopuri.

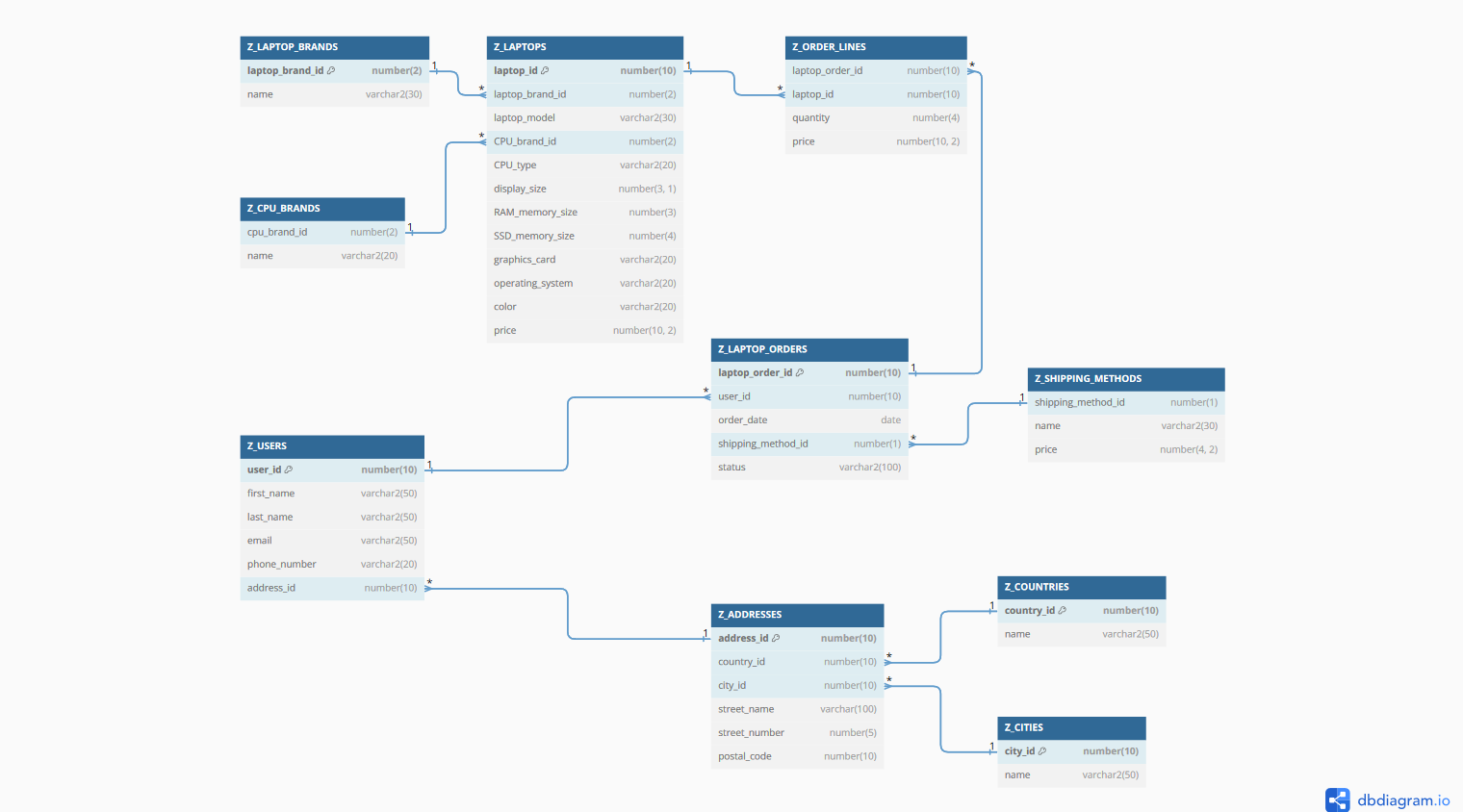
Atunci când un utilizator se loghează pe site, acesta trebuie să introducă informații precum nume, prenume, adresă de e-mail, număr de telefon și adresă. Adresa este asociată cu o țară și un oraș, iar acestea sunt preluate din tabelele Z\_COUNTRIES și Z\_CITIES, asigurând astfel integritatea referențială.

După logare, utilizatorul poate naviga prin catalogul de laptopuri, care sunt stocate în tabela Z\_LAPTOPS. Fiecare laptop are detalii precum brand, model, tip de procesor, dimensiunea ecranului, capacitatea de stocare SSD, placă grafică, sistem de operare și preț.

Atunci când un utilizator decide să plaseze o comandă, este creată o înregistrare în tabela Z\_LAPTOP\_ORDERS, cu detalii precum data comenzii, metoda de livrare și statusul comenzii. Metodele de livrare sunt gestionate în tabela Z\_SHIPPING\_METHODS.

Fiecare linie a comenzii este detaliată în tabela Z\_ORDER\_LINES, indicând laptopul comandat, cantitatea și prețul. Aceste detalii sunt legate de tabelul Z\_LAPTOP\_ORDERS și Z\_LAPTOPS prin foreign keys.

## **2. Schema conceptuală a bazei de date**



# **DEFINIREA SCHEMEI BAZEI DE DATE**

## **1. Implementarea operaţiilor de definire a datelor: CREATE, ALTER, DROP**

DROP TABLE Z\_USERS CASCADE CONSTRAINTS;

DROP TABLE Z\_ADDRESSES CASCADE CONSTRAINTS;

DROP TABLE Z\_COUNTRIES CASCADE CONSTRAINTS;

DROP TABLE Z\_CITIES CASCADE CONSTRAINTS;

DROP TABLE Z\_LAPTOPS CASCADE CONSTRAINTS;

DROP TABLE Z\_LAPTOP\_BRANDS CASCADE CONSTRAINTS;

DROP TABLE Z\_CPU\_BRANDS CASCADE CONSTRAINTS;

DROP TABLE Z\_ORDER\_LINES CASCADE CONSTRAINTS;

DROP TABLE Z\_LAPTOP\_ORDERS CASCADE CONSTRAINTS;

DROP TABLE Z\_SHIPPING\_METHODS CASCADE CONSTRAINTS;

prompt

prompt Creating table Z\_COUNTRIES

prompt ======================

prompt

CREATE TABLE Z\_COUNTRIES

(

COUNTRY\_ID NUMBER(10) CONSTRAINT PK\_COUNTRY\_ID PRIMARY KEY,

NAME VARCHAR2(50) CONSTRAINT NN\_COUNTRY\_NAME NOT NULL

CONSTRAINT UNQ\_COUNTRY\_NAME UNIQUE

);

prompt

prompt Creating table Z\_CITIES

prompt ======================

prompt

CREATE TABLE Z\_CITIES

(

CITY\_ID NUMBER(10) CONSTRAINT PK\_CITY\_ID PRIMARY KEY,

NAME VARCHAR2(50) CONSTRAINT NN\_CITY\_NAME NOT NULL

CONSTRAINT UNQ\_CITY\_NAME UNIQUE

);

prompt

prompt Creating table Z\_ADDRESSES

prompt ======================

prompt

CREATE TABLE Z\_ADDRESSES

(

ADDRESS\_ID NUMBER(10) CONSTRAINT PK\_ADDRESS\_ID PRIMARY KEY,

COUNTRY\_ID NUMBER(10) CONSTRAINT NN\_COUNTRY\_ID NOT NULL,

CITY\_ID NUMBER(10) CONSTRAINT NN\_CITY\_ID NOT NULL,

STREET\_NAME VARCHAR2(100),

STREET\_NUMBER NUMBER(5),

POSTAL\_CODE NUMBER(10)

);

ALTER TABLE Z\_ADDRESSES

ADD CONSTRAINT FK\_COUNTRY\_ID FOREIGN KEY (COUNTRY\_ID)

REFERENCES Z\_COUNTRIES (COUNTRY\_ID);

ALTER TABLE Z\_ADDRESSES

ADD CONSTRAINT FK\_CITY\_ID FOREIGN KEY (CITY\_ID)

REFERENCES Z\_CITIES (CITY\_ID);

prompt

prompt Creating table Z\_USERS

prompt ======================

prompt

CREATE TABLE Z\_USERS

(

USER\_ID NUMBER(10) CONSTRAINT PK\_USER\_ID PRIMARY KEY,

FIRST\_NAME VARCHAR2(50) CONSTRAINT NN\_FIRST\_NAME NOT NULL,

LAST\_NAME VARCHAR(50) CONSTRAINT NN\_LAST\_NAME NOT NULL,

EMAIL VARCHAR2(50) CONSTRAINT NN\_EMAIL NOT NULL,

PHONE\_NUMBER VARCHAR2(20),

ADDRESS\_ID NUMBER(10)

);

ALTER TABLE Z\_USERS

ADD CONSTRAINT FK\_ADDRESS\_ID FOREIGN KEY (ADDRESS\_ID)

REFERENCES Z\_ADDRESSES (ADDRESS\_ID);

prompt

prompt Creating table Z\_SHIPPING\_METHODS

prompt ======================

prompt

CREATE TABLE Z\_SHIPPING\_METHODS

(

SHIPPING\_METHOD\_ID NUMBER(1) CONSTRAINT PK\_SHIPPING\_METHOD\_ID PRIMARY KEY,

NAME VARCHAR2(30) CONSTRAINT NN\_SHIPPING\_METHOD\_NAME NOT NULL

CONSTRAINT UNQ\_SHIPPING\_METHOD\_NAME UNIQUE,

PRICE NUMBER(4,2) CONSTRAINT NN\_PRICE NOT NULL

);

ALTER TABLE Z\_SHIPPING\_METHODS

ADD CONSTRAINT CHK\_SHIPPING\_PRICE CHECK (PRICE >= 0);

prompt

prompt Creating table Z\_SHIPPING\_METHODS

prompt ======================

prompt

CREATE TABLE Z\_LAPTOP\_ORDERS

(

LAPTOP\_ORDER\_ID NUMBER(10) CONSTRAINT PK\_LAPTOP\_ORDER\_ID PRIMARY KEY,

USER\_ID NUMBER(10) CONSTRAINT NN\_USER\_ID NOT NULL,

ORDER\_DATE DATE,

SHIPPING\_METHOD\_ID NUMBER(1) CONSTRAINT NN\_SHIPPING\_METHOD\_ID NOT NULL,

STATUS VARCHAR2(100)

);

ALTER TABLE Z\_LAPTOP\_ORDERS

ADD CONSTRAINT FK\_USER\_ID FOREIGN KEY (USER\_ID)

REFERENCES Z\_USERS (USER\_ID);

ALTER TABLE Z\_LAPTOP\_ORDERS

ADD CONSTRAINT FK\_SHIPPING\_METHOD\_ID FOREIGN KEY (SHIPPING\_METHOD\_ID)

REFERENCES Z\_SHIPPING\_METHODS (SHIPPING\_METHOD\_ID);

prompt

prompt Creating table Z\_LAPTOP\_BRANDS

prompt ======================

prompt

CREATE TABLE Z\_LAPTOP\_BRANDS

(

LAPTOP\_BRAND\_ID NUMBER(2) CONSTRAINT PK\_LAPTOP\_BRAND\_ID PRIMARY KEY,

NAME VARCHAR2(30) CONSTRAINT NN\_LAPTOP\_BRAND\_NAME NOT NULL

CONSTRAINT UNQ\_LAPTOP\_BRAND\_NAME UNIQUE

);

prompt

prompt Creating table Z\_CPU\_BRANDS

prompt ======================

prompt

CREATE TABLE Z\_CPU\_BRANDS

(

CPU\_BRAND\_ID NUMBER(2) CONSTRAINT PK\_CPU\_BRAND\_ID PRIMARY KEY,

NAME VARCHAR2(20) CONSTRAINT NN\_CPU\_BRAND\_NAME NOT NULL

CONSTRAINT UNQ\_CPU\_BRAND\_NAME UNIQUE

);

prompt

prompt Creating table Z\_LAPTOPS

prompt ======================

prompt

CREATE TABLE Z\_LAPTOPS

(

LAPTOP\_ID NUMBER(10) CONSTRAINT PK\_LAPTOP\_ID PRIMARY KEY,

LAPTOP\_BRAND\_ID NUMBER(2) CONSTRAINT NN\_LAPTOP\_BRAND\_ID NOT NULL,

LAPTOP\_MODEL VARCHAR2(30),

CPU\_BRAND\_ID NUMBER(2) CONSTRAINT NN\_CPU\_BRAND\_ID NOT NULL,

CPU\_TYPE VARCHAR2(20),

DISPLAY\_SIZE NUMBER(3,1),

RAM\_MEMORY\_SIZE NUMBER(3),

SSD\_MEMORY\_SIZE NUMBER(4),

GRAPHICS\_CARD VARCHAR2(20),

OPERATING\_SYSTEM VARCHAR2(20),

COLOR VARCHAR2(20),

PRICE NUMBER(10, 2)

);

ALTER TABLE Z\_LAPTOPS

ADD CONSTRAINT FK\_LAPTOP\_BRAND\_ID FOREIGN KEY (LAPTOP\_BRAND\_ID)

REFERENCES Z\_LAPTOP\_BRANDS (LAPTOP\_BRAND\_ID);

ALTER TABLE Z\_LAPTOPS

ADD CONSTRAINT FK\_CPU\_BRAND\_ID FOREIGN KEY (CPU\_BRAND\_ID)

REFERENCES Z\_CPU\_BRANDS (CPU\_BRAND\_ID);

prompt

prompt Creating table Z\_ORDER\_LINES

prompt ======================

prompt

CREATE TABLE Z\_ORDER\_LINES

(

LAPTOP\_ORDER\_ID NUMBER(10) CONSTRAINT NN\_LAPTOP\_ORDER\_ID NOT NULL,

LAPTOP\_ID NUMBER(10) CONSTRAINT NN\_LAPTOP\_ID NOT NULL,

QUANTITY NUMBER(4) CONSTRAINT NN\_QUANTITY NOT NULL,

PRICE NUMBER(10, 2) CONSTRAINT NN\_ORDER\_LINE\_PRICE NOT NULL

);

ALTER TABLE Z\_ORDER\_LINES

ADD CONSTRAINT FK\_LAPTOP\_ORDER\_ID FOREIGN KEY (LAPTOP\_ORDER\_ID)

REFERENCES Z\_LAPTOP\_ORDERS (LAPTOP\_ORDER\_ID);

ALTER TABLE Z\_ORDER\_LINES

ADD CONSTRAINT FK\_LAPTOP\_ID FOREIGN KEY (LAPTOP\_ID)

REFERENCES Z\_LAPTOPS (LAPTOP\_ID);

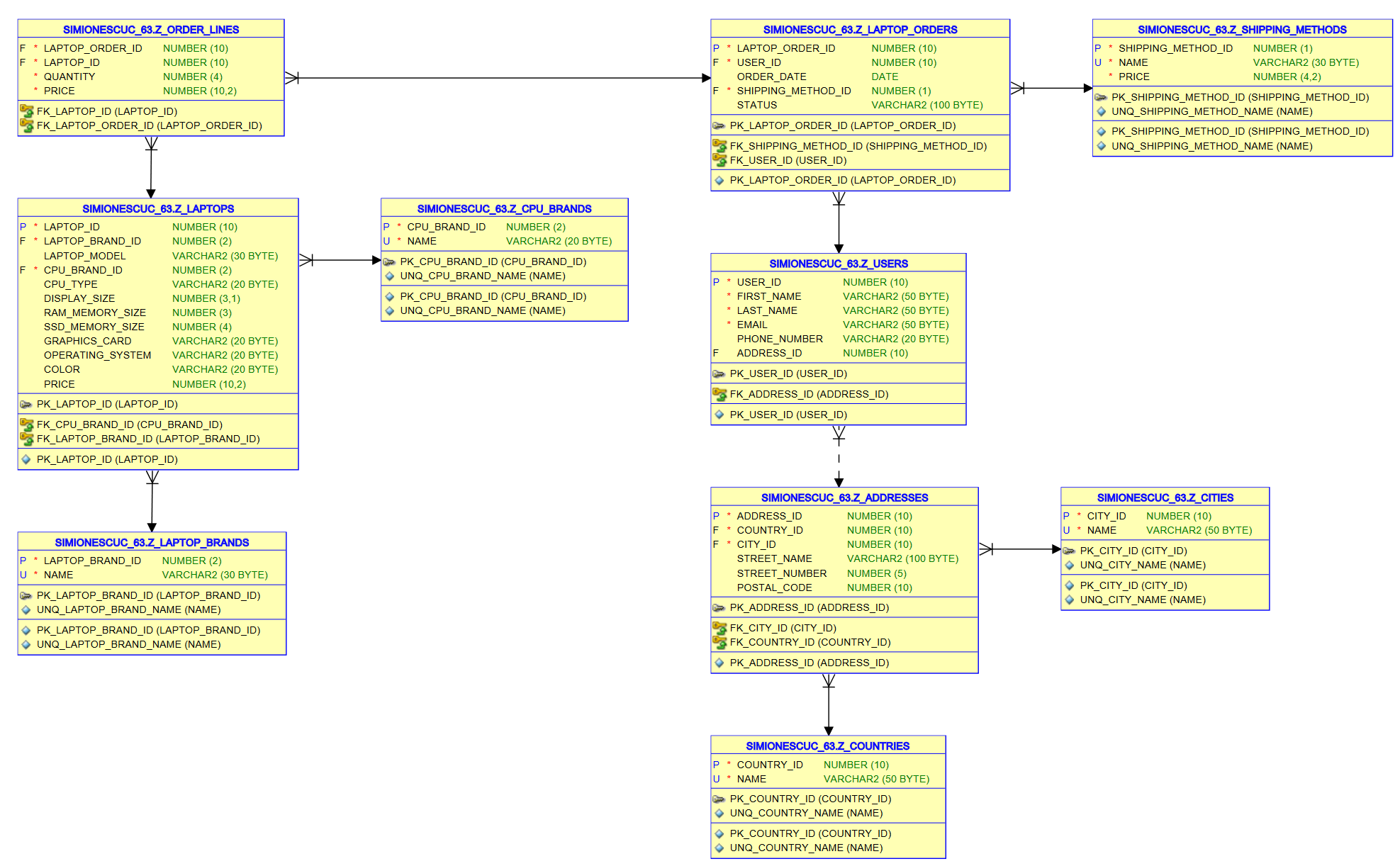
ALTER TABLE Z\_ORDER\_LINES

ADD CONSTRAINT CHK\_ORDER\_LINE\_QUANTITY CHECK (QUANTITY >= 1);

ALTER TABLE Z\_ORDER\_LINES

ADD CONSTRAINT CHK\_ORDER\_LINE\_PRICE CHECK (PRICE >= 0);

## **2. Schema conceptuală a bazei de date generată de Oracle SQL Developer**



# **IMPLEMENTAREA OPERAȚIILOR DE ACTUALIZARE A DATELOR**

## **1. INSERT**

prompt Loading Z\_CITIES

INSERT INTO Z\_CITIES (CITY\_ID, NAME)

VALUES (1, 'Bucuresti');

INSERT INTO Z\_CITIES (CITY\_ID, NAME)

VALUES (2, 'Cluj');

INSERT INTO Z\_CITIES (CITY\_ID, NAME)

VALUES (3, 'Caracal');

INSERT INTO Z\_CITIES (CITY\_ID, NAME)

VALUES (4, 'Craiova');

INSERT INTO Z\_CITIES (CITY\_ID, NAME)

VALUES (5, 'Targu Mures');

INSERT INTO Z\_CITIES (CITY\_ID, NAME)

VALUES (6, 'Roma');

INSERT INTO Z\_CITIES (CITY\_ID, NAME)

VALUES (7, 'Milano');

INSERT INTO Z\_CITIES (CITY\_ID, NAME)

VALUES (8, 'Barcelona');

INSERT INTO Z\_CITIES (CITY\_ID, NAME)

VALUES (9, 'Madrid');

INSERT INTO Z\_CITIES (CITY\_ID, NAME)

VALUES (10, 'Salamanca');

prompt 10 records loaded

prompt Loading Z\_COUNTRIES

INSERT INTO Z\_COUNTRIES (COUNTRY\_ID, NAME)

VALUES (1, 'Romania');

INSERT INTO Z\_COUNTRIES (COUNTRY\_ID, NAME)

VALUES (2, 'Italia');

INSERT INTO Z\_COUNTRIES (COUNTRY\_ID, NAME)

VALUES (3, 'Spania');

INSERT INTO Z\_COUNTRIES (COUNTRY\_ID, NAME)

VALUES (4, 'Austria');

INSERT INTO Z\_COUNTRIES (COUNTRY\_ID, NAME)

VALUES (5, 'Anglia');

prompt 5 records loaded

prompt Loading Z\_ADDRESSES

INSERT INTO Z\_ADDRESSES (ADDRESS\_ID, COUNTRY\_ID, CITY\_ID, STREET\_NAME, STREET\_NUMBER, POSTAL\_CODE)

VALUES (1, 1, 1, 'Strada 1', 10, 1000);

INSERT INTO Z\_ADDRESSES (ADDRESS\_ID, COUNTRY\_ID, CITY\_ID, STREET\_NAME, STREET\_NUMBER, POSTAL\_CODE)

VALUES (2, 1, 1, 'Strada 2', 20, 2000);

INSERT INTO Z\_ADDRESSES (ADDRESS\_ID, COUNTRY\_ID, CITY\_ID, STREET\_NAME, STREET\_NUMBER, POSTAL\_CODE)

VALUES (3, 1, 2, 'Strada 3', 30, 3000);

INSERT INTO Z\_ADDRESSES (ADDRESS\_ID, COUNTRY\_ID, CITY\_ID)

VALUES (4, 2, 7);

INSERT INTO Z\_ADDRESSES (ADDRESS\_ID, COUNTRY\_ID, CITY\_ID)

VALUES (5, 2, 6);

INSERT INTO Z\_ADDRESSES (ADDRESS\_ID, COUNTRY\_ID, CITY\_ID)

VALUES (6, 2, 6);

INSERT INTO Z\_ADDRESSES (ADDRESS\_ID, COUNTRY\_ID, CITY\_ID)

VALUES (7, 3, 8);

INSERT INTO Z\_ADDRESSES (ADDRESS\_ID, COUNTRY\_ID, CITY\_ID)

VALUES (8, 3, 8);

INSERT INTO Z\_ADDRESSES (ADDRESS\_ID, COUNTRY\_ID, CITY\_ID)

VALUES (9, 3, 8);

prompt 9 records loaded

prompt Loading Z\_USERS

INSERT INTO Z\_USERS(USER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, ADDRESS\_ID)

VALUES (1, 'Corina', 'Simionescu', 'corina.simionescu12@gmail.com', '0766111111', 1);

INSERT INTO Z\_USERS(USER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, ADDRESS\_ID)

VALUES (2, 'Mihai', 'Ionescu', 'mihai.ionescu@gmail.com', '0766222222', 1);

INSERT INTO Z\_USERS(USER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, ADDRESS\_ID)

VALUES (3, 'Ana', 'Aremere', 'ana.aremere@gmail.com', '0766333333', 2);

INSERT INTO Z\_USERS(USER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, ADDRESS\_ID)

VALUES (4, 'Matei', 'Florin', 'matei.florin@gmail.com', '0766444444', 3);

INSERT INTO Z\_USERS(USER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL)

VALUES (5, 'Cristian', 'Cantemir', 'cristian.cantemir@gmail.com');

INSERT INTO Z\_USERS(USER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL)

VALUES (6, 'Oana', 'Diaconu', 'oana.diaconu@gmail.com');

INSERT INTO Z\_USERS(USER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL)

VALUES (7, 'Ionut', 'Volintiru', 'ionut.volintiru@gmail.com');

INSERT INTO Z\_USERS(USER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL)

VALUES (8, 'Daria', 'Ispas', 'daria.ispas@gmail.com');

INSERT INTO Z\_USERS(USER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL)

VALUES (9, 'Miron', 'Stanciu', 'miron.stanciu@gmail.com');

INSERT INTO Z\_USERS(USER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, ADDRESS\_ID)

VALUES (10, 'Leonardo', 'Davinci', 'leonardo.davinci@gmail.com', 4);

INSERT INTO Z\_USERS(USER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, ADDRESS\_ID)

VALUES (11, 'Elena', 'Agostini', 'elena.agostini@gmail.com', 4);

INSERT INTO Z\_USERS(USER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, ADDRESS\_ID)

VALUES (12, 'Sofia', 'Lombardi', 'sofia.lombardi@gmail.com', 5);

INSERT INTO Z\_USERS(USER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, ADDRESS\_ID)

VALUES (13, 'Alessandro', 'Fontana', 'alessandro.fontana@gmail.com', 6);

INSERT INTO Z\_USERS(USER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, ADDRESS\_ID)

VALUES (14, 'Pablo', 'Escobar', 'pablo.escobar@gmail.com', 7);

INSERT INTO Z\_USERS(USER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, ADDRESS\_ID)

VALUES (15, 'Mateo', 'Fernanez', 'mateo.fernandez@gmail.com', 8);

INSERT INTO Z\_USERS(USER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, ADDRESS\_ID)

VALUES (16, 'Martina', 'Lopez', 'martina.lopez@gmail.com', 9);

prompt 16 records loaded

prompt Loading Z\_SHIPPING\_METHODS

INSERT INTO Z\_SHIPPING\_METHODS(SHIPPING\_METHOD\_ID, NAME, PRICE)

VALUES (1, 'Fan Courier', 15);

INSERT INTO Z\_SHIPPING\_METHODS(SHIPPING\_METHOD\_ID, NAME, PRICE)

VALUES (2, 'Sameday', 30);

INSERT INTO Z\_SHIPPING\_METHODS(SHIPPING\_METHOD\_ID, NAME, PRICE)

VALUES (3, 'EasyBox', 0);

prompt 3 records loaded

prompt Loading Z\_LAPTOP\_BRANDS

INSERT INTO Z\_LAPTOP\_BRANDS(LAPTOP\_BRAND\_ID, NAME)

VALUES (1, 'Apple');

INSERT INTO Z\_LAPTOP\_BRANDS(LAPTOP\_BRAND\_ID, NAME)

VALUES (2, 'Lenovo');

INSERT INTO Z\_LAPTOP\_BRANDS(LAPTOP\_BRAND\_ID, NAME)

VALUES (3, 'Asus');

prompt 3 records loaded

prompt Loading Z\_CPU\_BRANDS

INSERT INTO Z\_CPU\_BRANDS(CPU\_BRAND\_ID, NAME)

VALUES (1, 'Apple');

INSERT INTO Z\_CPU\_BRANDS(CPU\_BRAND\_ID, NAME)

VALUES (2, 'Intel');

INSERT INTO Z\_CPU\_BRANDS(CPU\_BRAND\_ID, NAME)

VALUES (3, 'AMD');

prompt 3 records loaded

prompt Loading Z\_LAPTOPS

INSERT INTO Z\_LAPTOPS(LAPTOP\_ID, LAPTOP\_BRAND\_ID, LAPTOP\_MODEL, CPU\_BRAND\_ID, CPU\_TYPE, DISPLAY\_SIZE, RAM\_MEMORY\_SIZE, SSD\_MEMORY\_SIZE, GRAPHICS\_CARD, OPERATING\_SYSTEM, COLOR, PRICE)

VALUES (1, 1, 'MacBook Pro', 1, 'M1', 14, 16, 512, NULL, 'MacOS', 'grey', 10000);

INSERT INTO Z\_LAPTOPS(LAPTOP\_ID, LAPTOP\_BRAND\_ID, LAPTOP\_MODEL, CPU\_BRAND\_ID, CPU\_TYPE, DISPLAY\_SIZE, RAM\_MEMORY\_SIZE, SSD\_MEMORY\_SIZE, GRAPHICS\_CARD, OPERATING\_SYSTEM, COLOR, PRICE)

VALUES (2, 2, 'Yoga Slim 7 Pro', 3, 'Ryzen 7', 14, 16, 512, 'Integrata', NULL, 'cloud grey', 5000);

INSERT INTO Z\_LAPTOPS(LAPTOP\_ID, LAPTOP\_BRAND\_ID, LAPTOP\_MODEL, CPU\_BRAND\_ID, CPU\_TYPE, DISPLAY\_SIZE, RAM\_MEMORY\_SIZE, SSD\_MEMORY\_SIZE, GRAPHICS\_CARD, OPERATING\_SYSTEM, COLOR, PRICE)

VALUES (3, 2, 'IdeaPad Slim 5', 2, 'I5', 16, 16, 512, 'Integrata', 'Windows 11', 'black', 3000);

INSERT INTO Z\_LAPTOPS(LAPTOP\_ID, LAPTOP\_BRAND\_ID, LAPTOP\_MODEL, CPU\_BRAND\_ID, CPU\_TYPE, DISPLAY\_SIZE, RAM\_MEMORY\_SIZE, SSD\_MEMORY\_SIZE, GRAPHICS\_CARD, OPERATING\_SYSTEM, COLOR, PRICE)

VALUES (4, 2, 'LOQ', 2, 'I5', 15.6, 16, 512, 'Dedicata', 'Linux', 'black', 5000);

INSERT INTO Z\_LAPTOPS(LAPTOP\_ID, LAPTOP\_BRAND\_ID, LAPTOP\_MODEL, CPU\_BRAND\_ID, CPU\_TYPE, DISPLAY\_SIZE, RAM\_MEMORY\_SIZE, SSD\_MEMORY\_SIZE, GRAPHICS\_CARD, OPERATING\_SYSTEM, COLOR, PRICE)

VALUES (5, 2, 'ThinkPad', 2, 'Xeon', 15.6, 32, 1000, 'Dedicata', 'Windows 10 Pro', 'black', 35000);

INSERT INTO Z\_LAPTOPS(LAPTOP\_ID, LAPTOP\_BRAND\_ID, LAPTOP\_MODEL, CPU\_BRAND\_ID, CPU\_TYPE, DISPLAY\_SIZE, RAM\_MEMORY\_SIZE, SSD\_MEMORY\_SIZE, GRAPHICS\_CARD, OPERATING\_SYSTEM, COLOR, PRICE)

VALUES (6, 1, 'MacBook Air', 1, 'M2', 15.3, 8, 512, 'Integrata', 'MacOS', 'starlight', 9000);

INSERT INTO Z\_LAPTOPS(LAPTOP\_ID, LAPTOP\_BRAND\_ID, LAPTOP\_MODEL, CPU\_BRAND\_ID, CPU\_TYPE, DISPLAY\_SIZE, RAM\_MEMORY\_SIZE, SSD\_MEMORY\_SIZE, GRAPHICS\_CARD, OPERATING\_SYSTEM, COLOR, PRICE)

VALUES (7, 1, 'MacBook Air', 1, 'M2', 15.3, 8, 512, 'Integrata', 'MacOS', 'midnight', 8900);

INSERT INTO Z\_LAPTOPS(LAPTOP\_ID, LAPTOP\_BRAND\_ID, LAPTOP\_MODEL, CPU\_BRAND\_ID, CPU\_TYPE, DISPLAY\_SIZE, RAM\_MEMORY\_SIZE, SSD\_MEMORY\_SIZE, GRAPHICS\_CARD, OPERATING\_SYSTEM, COLOR, PRICE)

VALUES (8, 1, 'MacBook Air', 1, 'M2', 15.3, 8, 512, 'Integrata', 'MacOS', 'space grey', 12300);

INSERT INTO Z\_LAPTOPS(LAPTOP\_ID, LAPTOP\_BRAND\_ID, LAPTOP\_MODEL, CPU\_BRAND\_ID, CPU\_TYPE, DISPLAY\_SIZE, RAM\_MEMORY\_SIZE, SSD\_MEMORY\_SIZE, GRAPHICS\_CARD, OPERATING\_SYSTEM, COLOR, PRICE)

VALUES (9, 1, 'MacBook Pro', 1, 'M3', 14, 8, 1000, 'Integrata', 'MacOS', 'silver', 1100);

INSERT INTO Z\_LAPTOPS(LAPTOP\_ID, LAPTOP\_BRAND\_ID, LAPTOP\_MODEL, CPU\_BRAND\_ID, CPU\_TYPE, DISPLAY\_SIZE, RAM\_MEMORY\_SIZE, SSD\_MEMORY\_SIZE, GRAPHICS\_CARD, OPERATING\_SYSTEM, COLOR, PRICE)

VALUES (10, 1, 'MacBook Pro', 1, 'M3 Max', 16, 96, 1000, 'Integrata', 'MacOS', 'space black', 25500);

INSERT INTO Z\_LAPTOPS(LAPTOP\_ID, LAPTOP\_BRAND\_ID, LAPTOP\_MODEL, CPU\_BRAND\_ID, CPU\_TYPE, DISPLAY\_SIZE, RAM\_MEMORY\_SIZE, SSD\_MEMORY\_SIZE, GRAPHICS\_CARD, OPERATING\_SYSTEM, COLOR, PRICE)

VALUES (11, 3, 'Vivobook', 2, 'I3', 15.6, 8, 256, 'Integrata', NULL, 'terra cota', 1800);

INSERT INTO Z\_LAPTOPS(LAPTOP\_ID, LAPTOP\_BRAND\_ID, LAPTOP\_MODEL, CPU\_BRAND\_ID, CPU\_TYPE, DISPLAY\_SIZE, RAM\_MEMORY\_SIZE, SSD\_MEMORY\_SIZE, GRAPHICS\_CARD, OPERATING\_SYSTEM, COLOR, PRICE)

VALUES (12, 3, 'Vivobook', 2, 'I7', 15.6, 16, 512, 'Integrata', NULL, 'indie black', 3000);

INSERT INTO Z\_LAPTOPS(LAPTOP\_ID, LAPTOP\_BRAND\_ID, LAPTOP\_MODEL, CPU\_BRAND\_ID, CPU\_TYPE, DISPLAY\_SIZE, RAM\_MEMORY\_SIZE, SSD\_MEMORY\_SIZE, GRAPHICS\_CARD, OPERATING\_SYSTEM, COLOR, PRICE)

VALUES (13, 3, 'Zephyrus', 3, 'Ryzen 9', 16, 32, 2000, 'Dedicata', 'Windows 11', 'black', 33900);

prompt 13 records loaded

prompt Loading Z\_LAPTOP\_ORDERS

INSERT INTO Z\_LAPTOP\_ORDERS(LAPTOP\_ORDER\_ID, USER\_ID, ORDER\_DATE, SHIPPING\_METHOD\_ID, STATUS)

VALUES (1, 1, to\_date('12-11-2023', 'dd-mm-yyyy'), 3, 'Comanda ajunsa in punctul de livrare');

INSERT INTO Z\_LAPTOP\_ORDERS(LAPTOP\_ORDER\_ID, USER\_ID, ORDER\_DATE, SHIPPING\_METHOD\_ID, STATUS)

VALUES (2, 2, to\_date('01-01-2023', 'dd-mm-yyyy'), 1, 'Comanda predata curierului');

INSERT INTO Z\_LAPTOP\_ORDERS(LAPTOP\_ORDER\_ID, USER\_ID, ORDER\_DATE, SHIPPING\_METHOD\_ID, STATUS)

VALUES (3, 3, to\_date('02-02-2023', 'dd-mm-yyyy'), 2, 'Comanda se indreapta spre punctul de livrare');

INSERT INTO Z\_LAPTOP\_ORDERS(LAPTOP\_ORDER\_ID, USER\_ID, ORDER\_DATE, SHIPPING\_METHOD\_ID, STATUS)

VALUES (4, 5, to\_date('03-03-2023', 'dd-mm-yyyy'), 3, 'Comanda se indreapta spre punctul de livrare');

INSERT INTO Z\_LAPTOP\_ORDERS(LAPTOP\_ORDER\_ID, USER\_ID, ORDER\_DATE, SHIPPING\_METHOD\_ID, STATUS)

VALUES (5, 1, to\_date('18-12-2023', 'dd-mm-yyyy'), 3, 'Comanda se proceseaza');

INSERT INTO Z\_LAPTOP\_ORDERS(LAPTOP\_ORDER\_ID, USER\_ID, ORDER\_DATE, SHIPPING\_METHOD\_ID, STATUS)

VALUES (6, 6, to\_date('04-04-2023', 'dd-mm-yyyy'), 1, 'Comanda predata curierului');

INSERT INTO Z\_LAPTOP\_ORDERS(LAPTOP\_ORDER\_ID, USER\_ID, ORDER\_DATE, SHIPPING\_METHOD\_ID, STATUS)

VALUES (7, 10, to\_date('05-05-2023', 'dd-mm-yyyy'), 2, 'Comanda se proceseaza');

INSERT INTO Z\_LAPTOP\_ORDERS(LAPTOP\_ORDER\_ID, USER\_ID, ORDER\_DATE, SHIPPING\_METHOD\_ID, STATUS)

VALUES (8, 10, to\_date('06-06-2023', 'dd-mm-yyyy'), 2, 'Comanda se indreapta spre punctul de livrare');

INSERT INTO Z\_LAPTOP\_ORDERS(LAPTOP\_ORDER\_ID, USER\_ID, ORDER\_DATE, SHIPPING\_METHOD\_ID, STATUS)

VALUES (9, 11, to\_date('06-06-2023', 'dd-mm-yyyy'), 2, 'Comanda se indreapta spre punctul de livrare');

INSERT INTO Z\_LAPTOP\_ORDERS(LAPTOP\_ORDER\_ID, USER\_ID, ORDER\_DATE, SHIPPING\_METHOD\_ID, STATUS)

VALUES (10, 16, to\_date('03-03-2023', 'dd-mm-yyyy'), 1, 'Comanda ajunsa in punctul de livrare');

prompt 10 records loaded

prompt Loading Z\_ORDER\_LINES

INSERT INTO Z\_ORDER\_LINES(LAPTOP\_ORDER\_ID, LAPTOP\_ID, QUANTITY, PRICE)

VALUES (1, 2, 1, 5000);

INSERT INTO Z\_ORDER\_LINES(LAPTOP\_ORDER\_ID, LAPTOP\_ID, QUANTITY, PRICE)

VALUES (2, 1, 1, 10000);

INSERT INTO Z\_ORDER\_LINES(LAPTOP\_ORDER\_ID, LAPTOP\_ID, QUANTITY, PRICE)

VALUES (3, 3, 1, 3000);

INSERT INTO Z\_ORDER\_LINES(LAPTOP\_ORDER\_ID, LAPTOP\_ID, QUANTITY, PRICE)

VALUES (4, 4, 2, 5000);

INSERT INTO Z\_ORDER\_LINES(LAPTOP\_ORDER\_ID, LAPTOP\_ID, QUANTITY, PRICE)

VALUES (5, 1, 1, 10000);

INSERT INTO Z\_ORDER\_LINES(LAPTOP\_ORDER\_ID, LAPTOP\_ID, QUANTITY, PRICE)

VALUES (6, 6, 3, 9000);

INSERT INTO Z\_ORDER\_LINES(LAPTOP\_ORDER\_ID, LAPTOP\_ID, QUANTITY, PRICE)

VALUES (7, 10, 1, 25500);

INSERT INTO Z\_ORDER\_LINES(LAPTOP\_ORDER\_ID, LAPTOP\_ID, QUANTITY, PRICE)

VALUES (8, 11, 1, 1800);

INSERT INTO Z\_ORDER\_LINES(LAPTOP\_ORDER\_ID, LAPTOP\_ID, QUANTITY, PRICE)

VALUES (9, 2, 4, 5000);

INSERT INTO Z\_ORDER\_LINES(LAPTOP\_ORDER\_ID, LAPTOP\_ID, QUANTITY, PRICE)

VALUES (10, 1, 1, 10000);

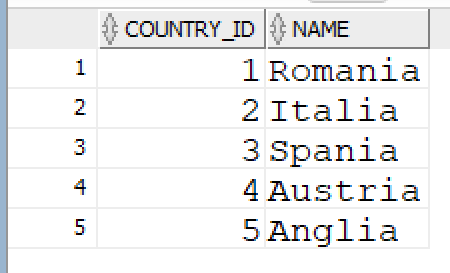
prompt 10 records loaded

### **1.1 Datele inserate in tabele**

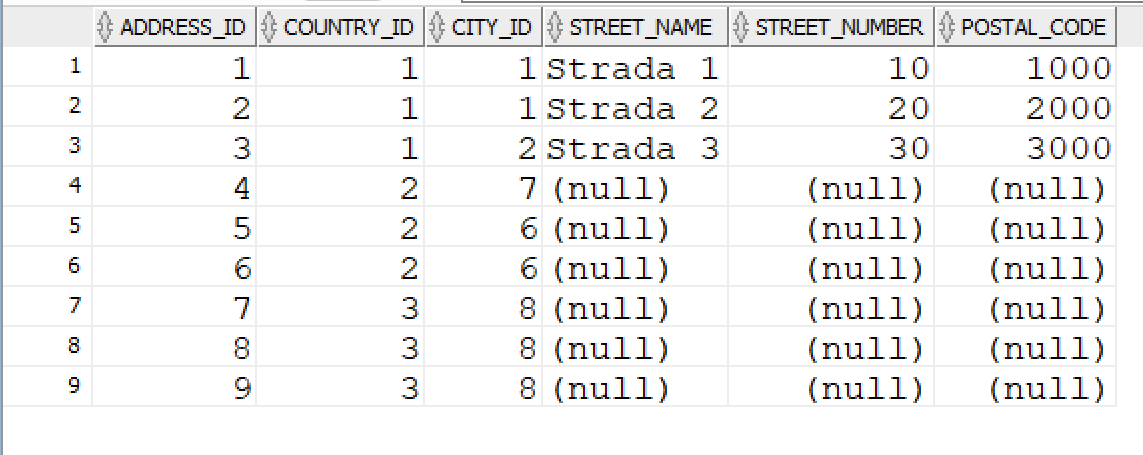
* Z\_CITIES



* Z\_COUNTRIES



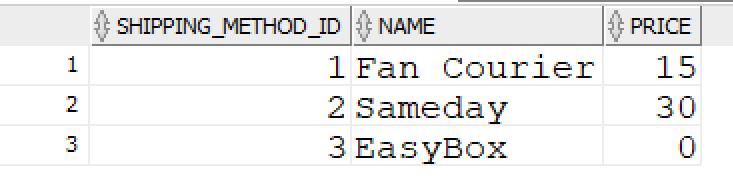
* Z\_ADDRESSES



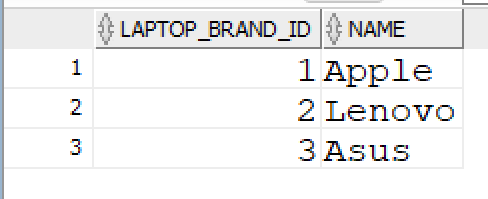
* Z\_USERS



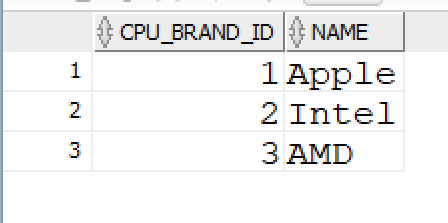
* Z\_SHIPPING\_METHODS



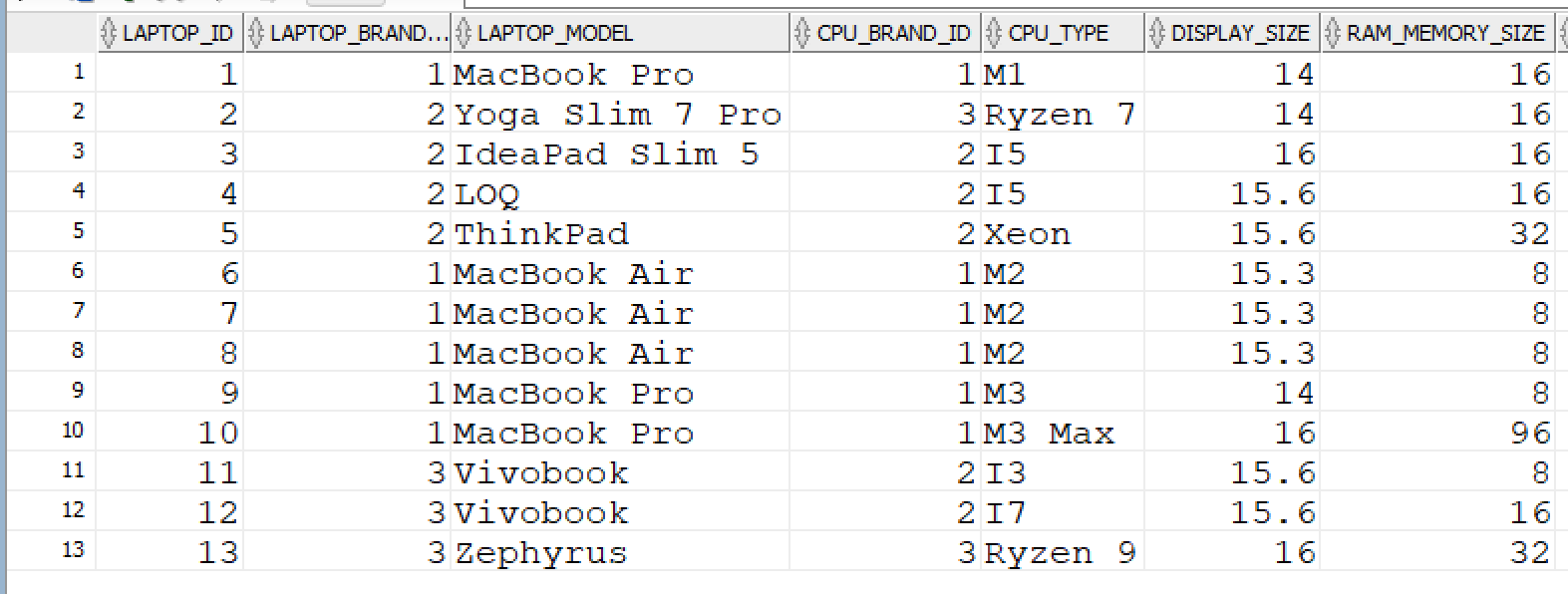
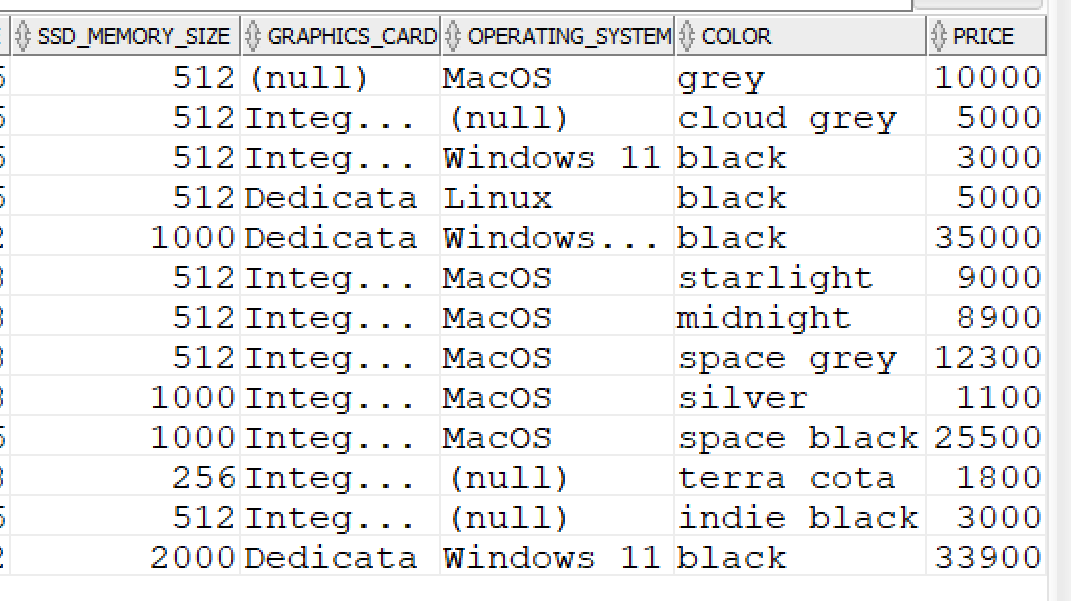
* Z\_LAPTOP\_BRANDS



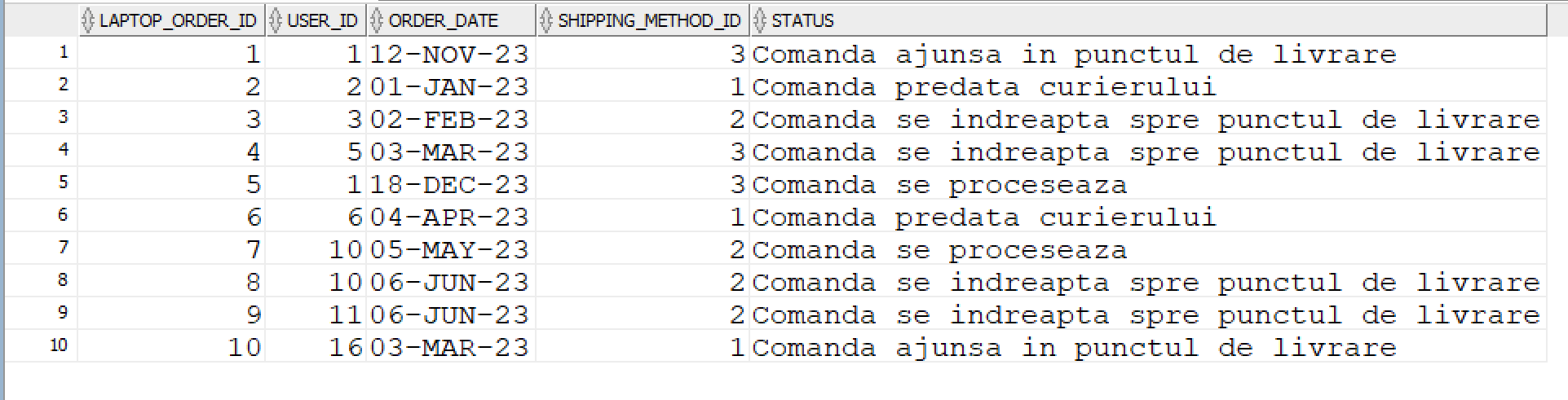
* Z\_CPU\_BRANDS



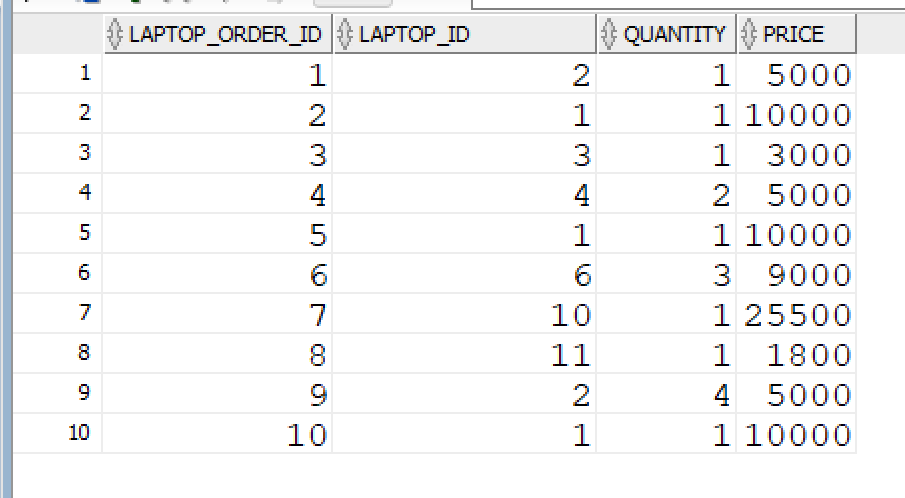
* Z\_LAPTOPS

* Z\_LAPTOP\_ORDERS



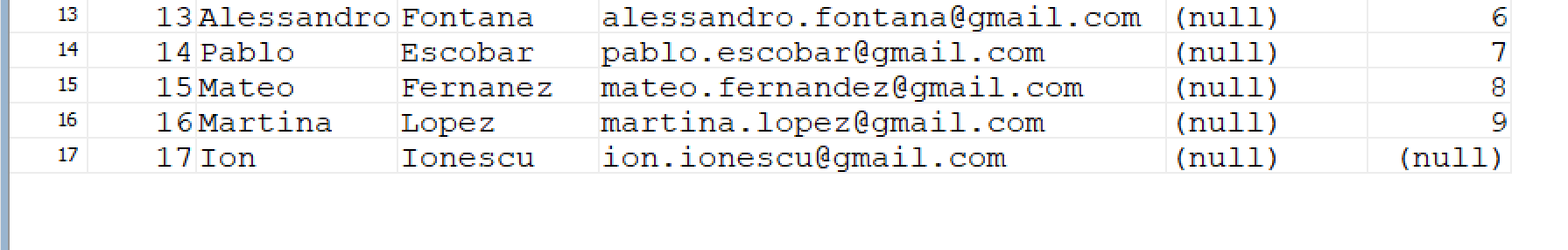
* Z\_ORDER\_LINES



Pentru exemplificarea comenzilor UPDATE si DELETE am inserat un user in tabela Z\_USERS:

INSERT INTO Z\_USERS(USER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL)

VALUES (17, ‘Ion’, ‘Ionescu’, ‘ion.ionescu@gmail.com’);



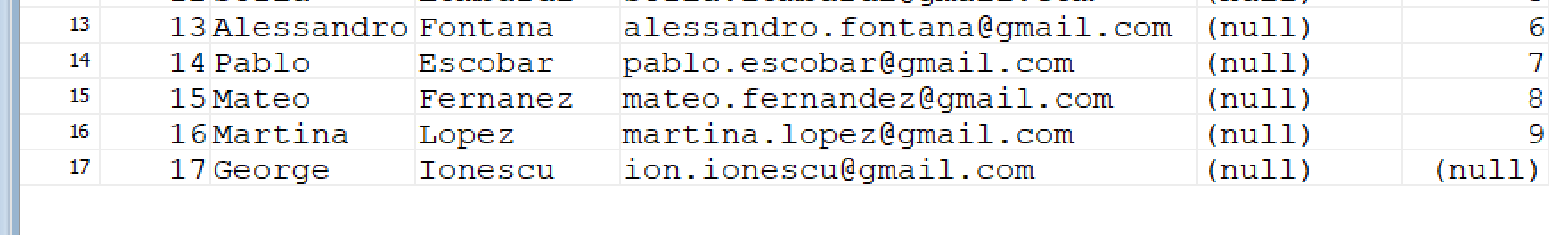
## **2. UPDATE**

-- Modificarea prenumeului in ‘George’ pentru userul cu id-ul = 17

UPDATE Z\_USERS

SET FIRST\_NAME = ‘George’

WHERE USER\_ID = 17;

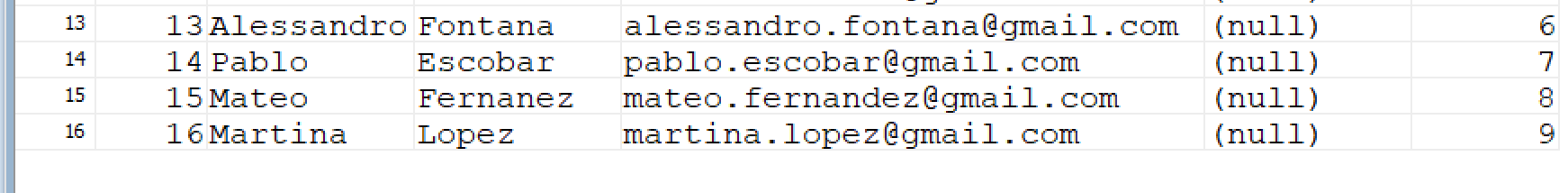


## **3. DELETE**

-- Stergerea userului cu id-ul = 17

DELETE FROM Z\_USERS

WHERE USER\_ID = 17;



# **IMPLEMENTAREA INTEROGĂRILOR**

## **1. JONCȚIUNI**

### **1.1. INNER JOIN**

-- Afisare tara si orasul in care locuieste fiecare user

SELECT U.FIRST\_NAME, U.LAST\_NAME, CO.NAME AS COUNTRY, CI.NAME AS CITY

FROM Z\_USERS U

JOIN Z\_ADDRESSES A ON U.ADDRESS\_ID = A.ADDRESS\_ID

JOIN Z\_COUNTRIES CO ON A.COUNTRY\_ID = CO.COUNTRY\_ID

JOIN Z\_CITIES CI ON A.CITY\_ID = CI.CITY\_ID;



-- Afisare tara si orasul in care locuieste fiecare user care a comandat macar un laptop

SELECT U.FIRST\_NAME, U.LAST\_NAME, CO.NAME AS COUNTRY, CI.NAME AS CITY

FROM Z\_USERS U

JOIN Z\_ADDRESSES A ON U.ADDRESS\_ID = A.ADDRESS\_ID

JOIN Z\_COUNTRIES CO ON A.COUNTRY\_ID = CO.COUNTRY\_ID

JOIN Z\_CITIES CI ON A.CITY\_ID = CI.CITY\_ID

JOIN Z\_LAPTOP\_ORDERS LO ON U.USER\_ID = LO.USER\_ID

WHERE U.USER\_ID = LO.USER\_ID;



### **1.2. OUTER JOIN**

-- Selectarea tuturor userilor si a tarii in care locuiesc (daca au)

-- si selectarea tuturor tarilor indiferent daca au sau nu un user asociat

SELECT U.USER\_ID, U.FIRST\_NAME, CO.NAME AS COUNTRY\_NAME

FROM Z\_USERS U

LEFT JOIN Z\_ADDRESSES A ON U.ADDRESS\_ID = A.ADDRESS\_ID

LEFT JOIN Z\_COUNTRIES CO ON A.COUNTRY\_ID = CO.COUNTRY\_ID

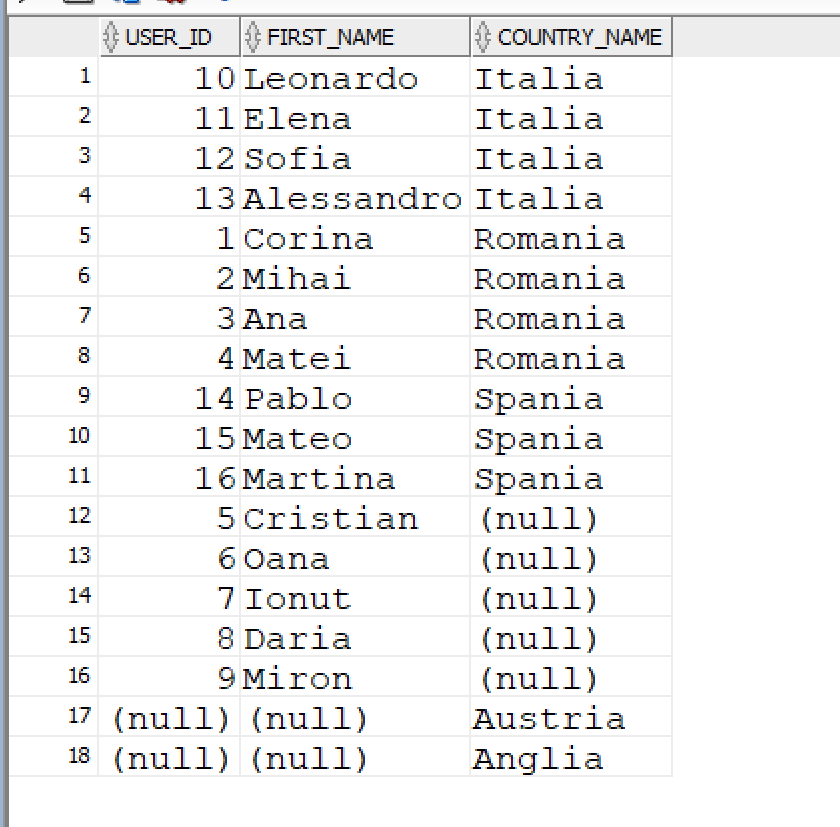
UNION

SELECT USER\_ID, FIRST\_NAME, CO.NAME AS COUNTRY\_NAME

FROM Z\_COUNTRIES CO

LEFT JOIN Z\_ADDRESSES A ON CO.COUNTRY\_ID = A.COUNTRY\_ID

LEFT JOIN Z\_USERS U ON A.ADDRESS\_ID = U.ADDRESS\_ID;



### **1.3. LEFT JOIN**

-- Afisare toti userii si numarul de laptopuri comandate

SELECT

U.USER\_ID,

U.FIRST\_NAME,

U.LAST\_NAME,

U.EMAIL,

SUM(QUANTITY) AS NO\_ORDERED\_LAPTOPS

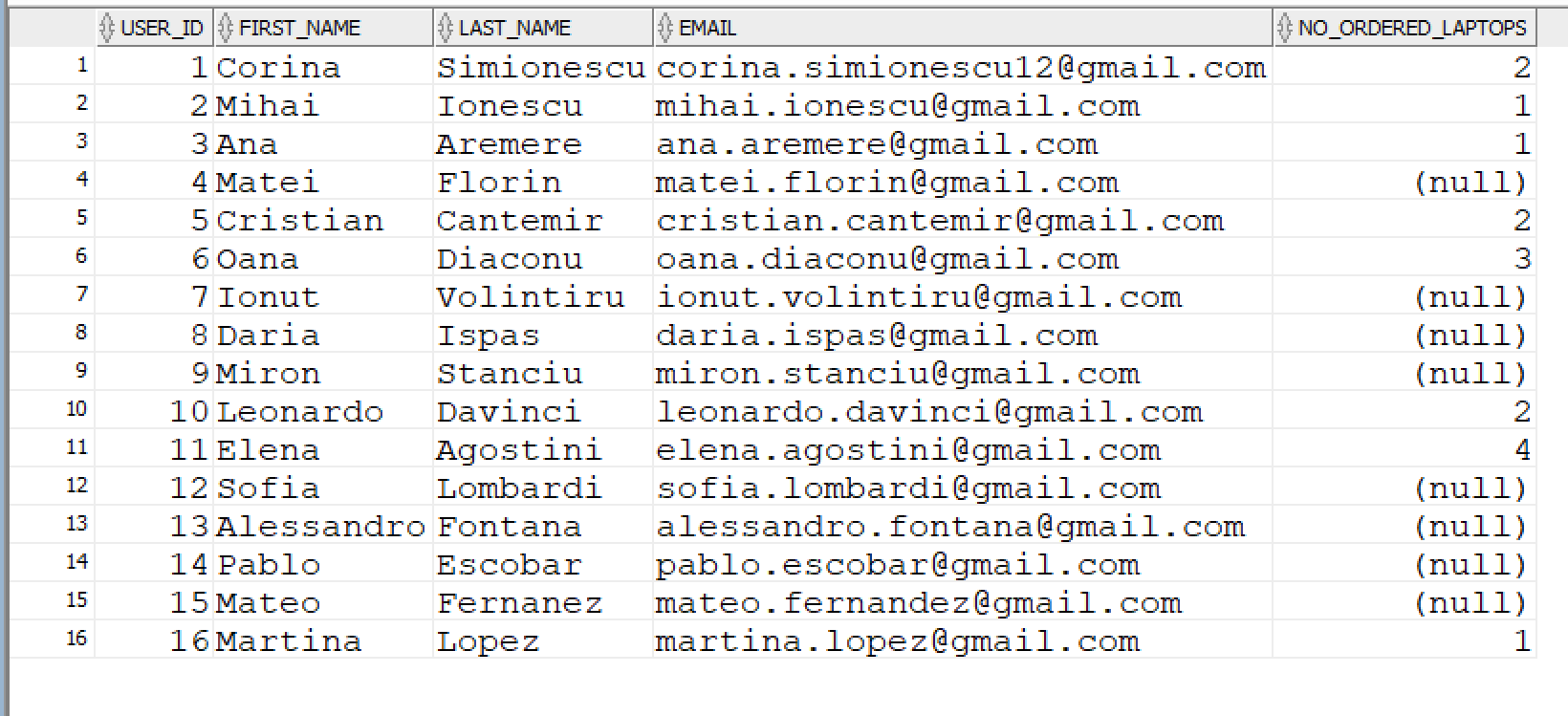
FROM Z\_USERS U

LEFT JOIN Z\_LAPTOP\_ORDERS LO ON U.USER\_ID = LO.USER\_ID

LEFT JOIN Z\_ORDER\_LINES OL ON LO.LAPTOP\_ORDER\_ID = OL.LAPTOP\_ORDER\_ID

GROUP BY U.USER\_ID, U.FIRST\_NAME, U.LAST\_NAME, U.EMAIL

ORDER BY U.USER\_ID;



### **1.4. RIGHT JOIN**

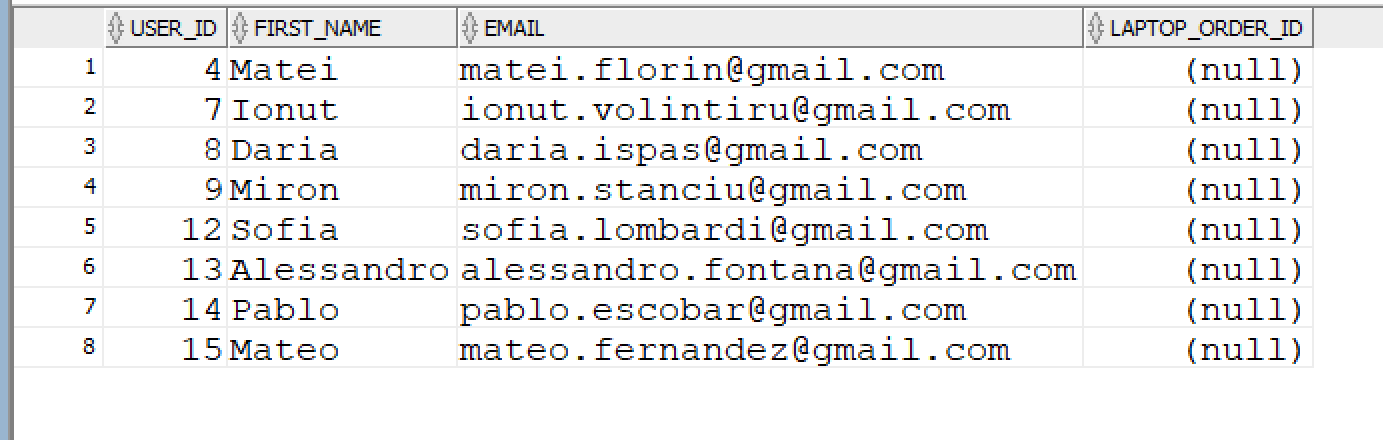
-- Afisare toti userii care nu au dat comanda

SELECT U.USER\_ID, U.FIRST\_NAME, U.EMAIL, LO.LAPTOP\_ORDER\_ID

FROM Z\_LAPTOP\_ORDERS LO

RIGHT JOIN Z\_USERS U ON LO.USER\_ID = U.USER\_ID

WHERE LAPTOP\_ORDER\_ID IS NULL;



## **2. FUNCȚII DE GRUP**

### **2.1. COUNT**

-- Afisare toate laptopurile, de cate ori au fost comandate si cantitatea totala comandata

SELECT

L.LAPTOP\_ID,

LB.NAME,

L.LAPTOP\_MODEL,

COUNT(OL.LAPTOP\_ID) AS NO\_ORDERS,

SUM(OL.QUANTITY) AS ORDERED\_QUANTITY

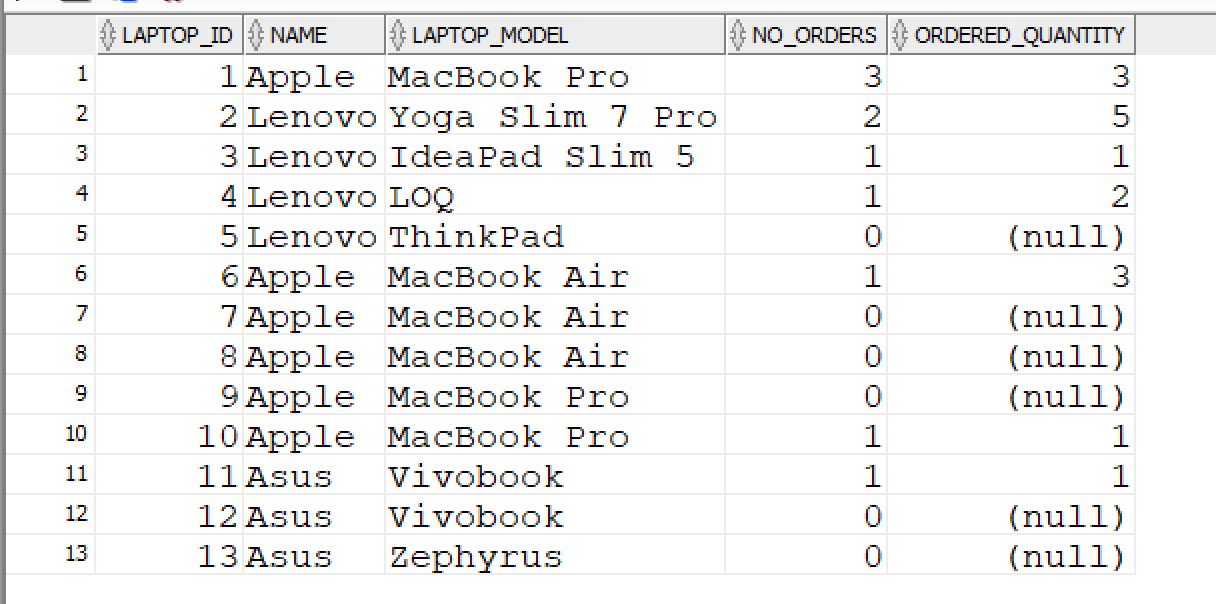
FROM Z\_LAPTOPS L

JOIN Z\_LAPTOP\_BRANDS LB ON L.LAPTOP\_BRAND\_ID = LB.LAPTOP\_BRAND\_ID

LEFT JOIN Z\_ORDER\_LINES OL ON L.LAPTOP\_ID = OL.LAPTOP\_ID

GROUP BY L.LAPTOP\_ID, LB.NAME, L.LAPTOP\_MODEL

ORDER BY L.LAPTOP\_ID ASC;



-- Afisare laptopuri care au fost comandate intre 1 si 3 ori

SELECT

L.LAPTOP\_ID,

LB.NAME,

L.LAPTOP\_MODEL,

COUNT(OL.LAPTOP\_ID) AS NO\_ORDERS

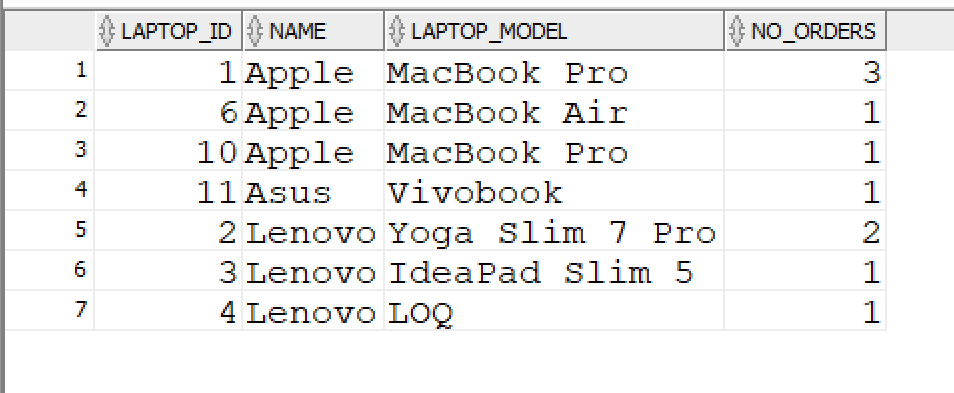
FROM Z\_LAPTOPS L

JOIN Z\_ORDER\_LINES OL ON L.LAPTOP\_ID = OL.LAPTOP\_ID

JOIN Z\_LAPTOP\_BRANDS LB ON L.LAPTOP\_BRAND\_ID = LB.LAPTOP\_BRAND\_ID

HAVING COUNT(OL.LAPTOP\_ID) BETWEEN 1 AND 3

GROUP BY L.LAPTOP\_ID, LB.NAME, L.LAPTOP\_MODEL;



### **2.2. SUM**

-- Afisare tara cu cele mai multe laptopuri comandate

SELECT CO.NAME, SUM(OL.QUANTITY) AS ORDERED\_QUANTITY

FROM Z\_COUNTRIES CO

JOIN Z\_ADDRESSES A ON CO.COUNTRY\_ID = A.COUNTRY\_ID

JOIN Z\_USERS U ON A.ADDRESS\_ID = U.ADDRESS\_ID

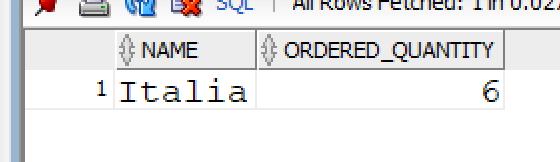
JOIN Z\_LAPTOP\_ORDERS LO ON U.USER\_ID = LO.USER\_ID

JOIN Z\_ORDER\_LINES OL ON LO.LAPTOP\_ORDER\_ID = OL.LAPTOP\_ORDER\_ID

GROUP BY CO.NAME

ORDER BY SUM(OL.QUANTITY) DESC

FETCH FIRST 1 ROW ONLY;



-- Afisare toti userii si numarul de laptopuri comandate

SELECT

U.USER\_ID,

U.FIRST\_NAME,

U.LAST\_NAME,

U.EMAIL,

SUM(QUANTITY) AS NO\_ORDERED\_LAPTOPS

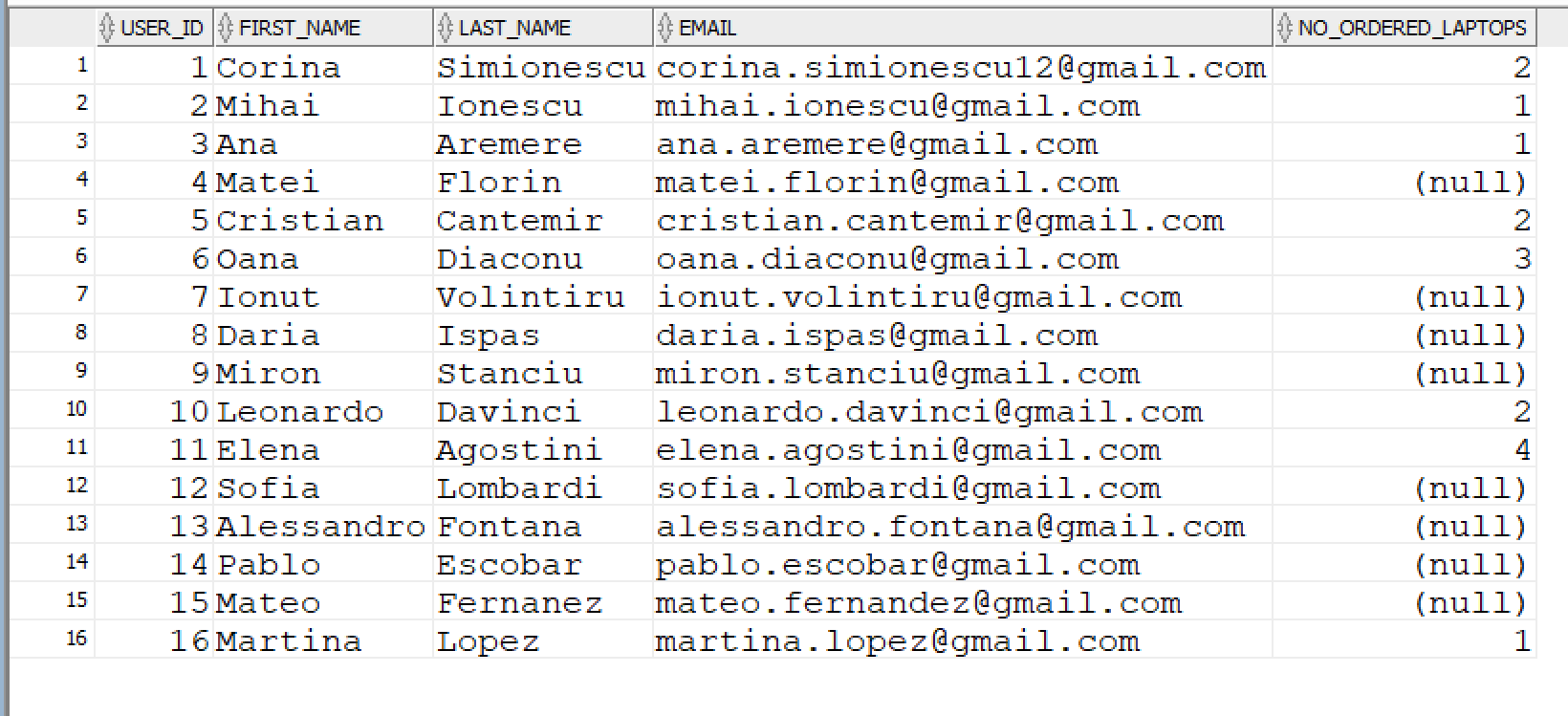
FROM Z\_USERS U

LEFT JOIN Z\_LAPTOP\_ORDERS LO ON U.USER\_ID = LO.USER\_ID

LEFT JOIN Z\_ORDER\_LINES OL ON LO.LAPTOP\_ORDER\_ID = OL.LAPTOP\_ORDER\_ID

GROUP BY U.USER\_ID, U.FIRST\_NAME, U.LAST\_NAME, U.EMAIL

ORDER BY U.USER\_ID;



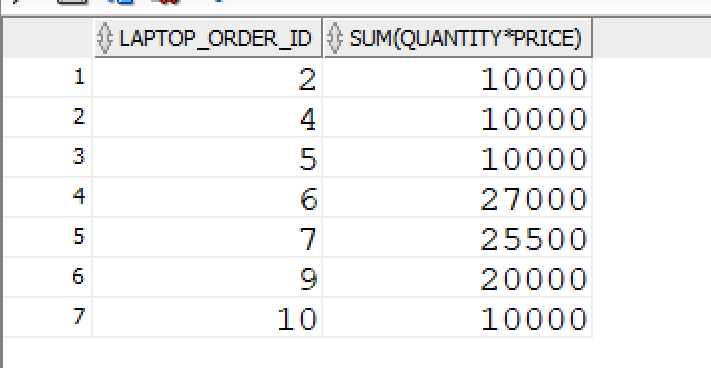
-- Afisare comenzile de peste 5000 de lei

SELECT LAPTOP\_ORDER\_ID, SUM(QUANTITY\*PRICE)

FROM Z\_ORDER\_LINES

HAVING SUM(QUANTITY\*PRICE) > 5000

GROUP BY LAPTOP\_ORDER\_ID;



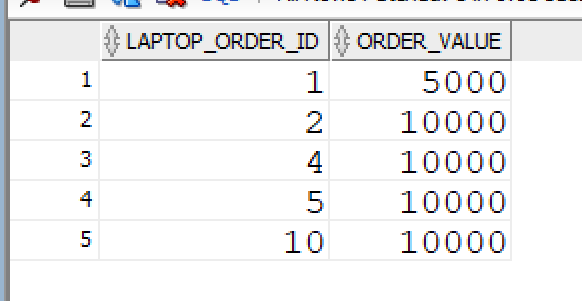
-- Afisare comenzi care au valoarea de 5000 de lei sau 10000 de lei

SELECT LAPTOP\_ORDER\_ID, SUM(QUANTITY\*PRICE) AS ORDER\_VALUE

FROM Z\_ORDER\_LINES

HAVING SUM(QUANTITY\*PRICE) IN (5000, 10000)

GROUP BY LAPTOP\_ORDER\_ID;

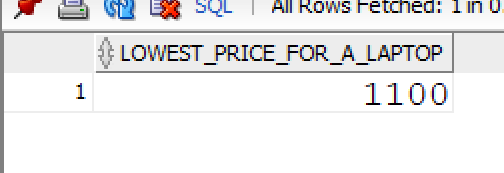


### **2.3. MIN**

--Afisare cel mai mic pret dintre preturile toate laptopurilor

SELECT MIN(PRICE) AS LOWEST\_PRICE\_FOR\_A\_LAPTOP

FROM Z\_LAPTOPS;

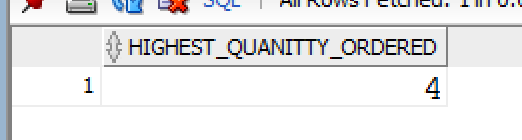


### **2.4. MAX**

-- Afisare cea mai mare cantitate de laptopuri comandate intr-o singura comanda

SELECT MAX(QUANTITY) AS HIGHEST\_QUANITTY\_ORDERED

FROM Z\_ORDER\_LINES;

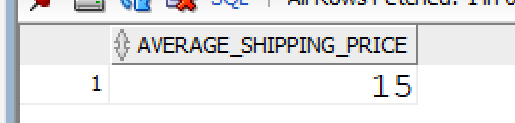


### **2.5. AVG**

-- Afisare pretul in medie pentru metoda de livrare

SELECT AVG(PRICE) AS AVERAGE\_SHIPPING\_PRICE

FROM Z\_SHIPPING\_METHODS;



## **3. FUNCȚII SINGLE-ROW**

### **3.1. FUNCȚII NUMERICE**

#### **3.1.1. ROUND**

-- Afisare dimensiunea medie a displayului pentru fiecare brand de laptopuri

SELECT

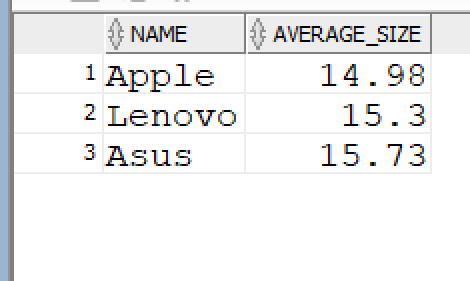
LB.NAME,

ROUND(AVG(L.DISPLAY\_SIZE), 2) AS AVERAGE\_SIZE

FROM Z\_LAPTOPS L

JOIN Z\_LAPTOP\_BRANDS LB ON L.LAPTOP\_BRAND\_ID = LB.LAPTOP\_BRAND\_ID

GROUP BY LB.NAME;



#### **3.1.2. TRUNC**

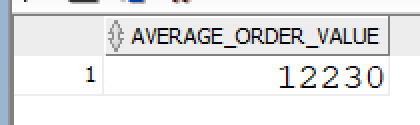
-- Afisare valoarea medie a comenzilor

SELECT TRUNC(AVG(ONE\_ORDER\_VALUE)) AS AVERAGE\_ORDER\_VALUE

FROM (SELECT LAPTOP\_ORDER\_ID, SUM(PRICE\*QUANTITY) AS ONE\_ORDER\_VALUE

FROM Z\_ORDER\_LINES

GROUP BY LAPTOP\_ORDER\_ID);



### **3.2. FUNCTII DE CARACTERE**

#### **3.2.1. LOWER**

-- Afisare cate laptopuri contin in numele culorii 'black' si pretul lor mediu

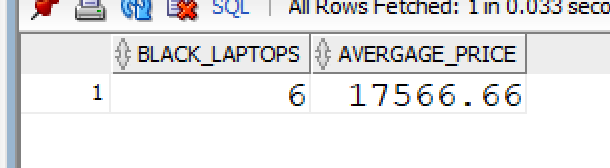
SELECT

COUNT(LAPTOP\_ID) AS BLACK\_LAPTOPS,

TRUNC(AVG(PRICE), 2) AS AVERGAGE\_PRICE

FROM Z\_LAPTOPS

WHERE LOWER(COLOR) LIKE '%black%';



#### **3.2.2. UPPER**

-- Afisare cate comenzi au fost livrate de Fan Courier, valoarea comenzii (cu tot cu pretul

-- metodei de livrare si totalul userilor care au dat aceste comenzi

SELECT

COUNT(LO.LAPTOP\_ORDER\_ID) AS LAPTOPS\_FANCOURIER,

COUNT(LO.USER\_ID) AS USERS,

(SUM(OL.PRICE) + (COUNT(LO.LAPTOP\_ORDER\_ID) \* SM.PRICE)) AS TOTAL\_VALUE

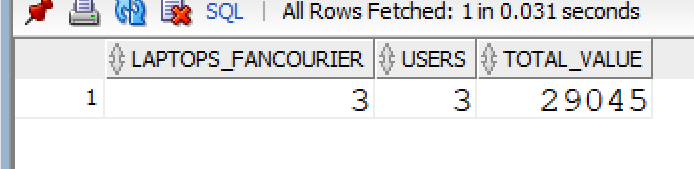
FROM Z\_LAPTOP\_ORDERS LO

JOIN Z\_SHIPPING\_METHODS SM ON LO.SHIPPING\_METHOD\_ID = SM.SHIPPING\_METHOD\_ID

JOIN Z\_ORDER\_LINES OL ON LO.LAPTOP\_ORDER\_ID = OL.LAPTOP\_ORDER\_ID

WHERE UPPER(SM.NAME) LIKE 'FAN%COURIER'

GROUP BY SM.PRICE;



#### **3.2.3. INITCAP**

-- Afisare laptopuri al caror tip de procesor este 'Ryzen'. Pentru fiecare laptop

-- sa se afiseze si pretul, cantitatea in care a fost cumparat si profitul generat.

SELECT

L.LAPTOP\_ID,

LB.NAME AS LAPTOP\_BRAND,

L.LAPTOP\_MODEL,

L.CPU\_TYPE,

L.PRICE,

SUM(OL.QUANTITY) AS ORDERED\_QUANTITY,

SUM(OL.PRICE\*OL.QUANTITY) AS PROFIT

FROM Z\_LAPTOPS L

JOIN Z\_LAPTOP\_BRANDS LB ON L.LAPTOP\_BRAND\_ID = LB.LAPTOP\_BRAND\_ID

LEFT JOIN Z\_ORDER\_LINES OL ON L.LAPTOP\_ID = OL.LAPTOP\_ID

WHERE INITCAP(CPU\_TYPE) LIKE 'Ryzen%'

GROUP BY

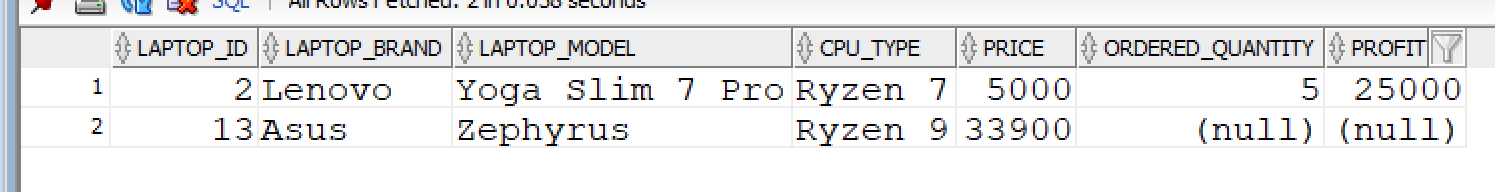
L.LAPTOP\_ID,

LB.NAME,

L.LAPTOP\_MODEL,

L.CPU\_TYPE,

L.PRICE;

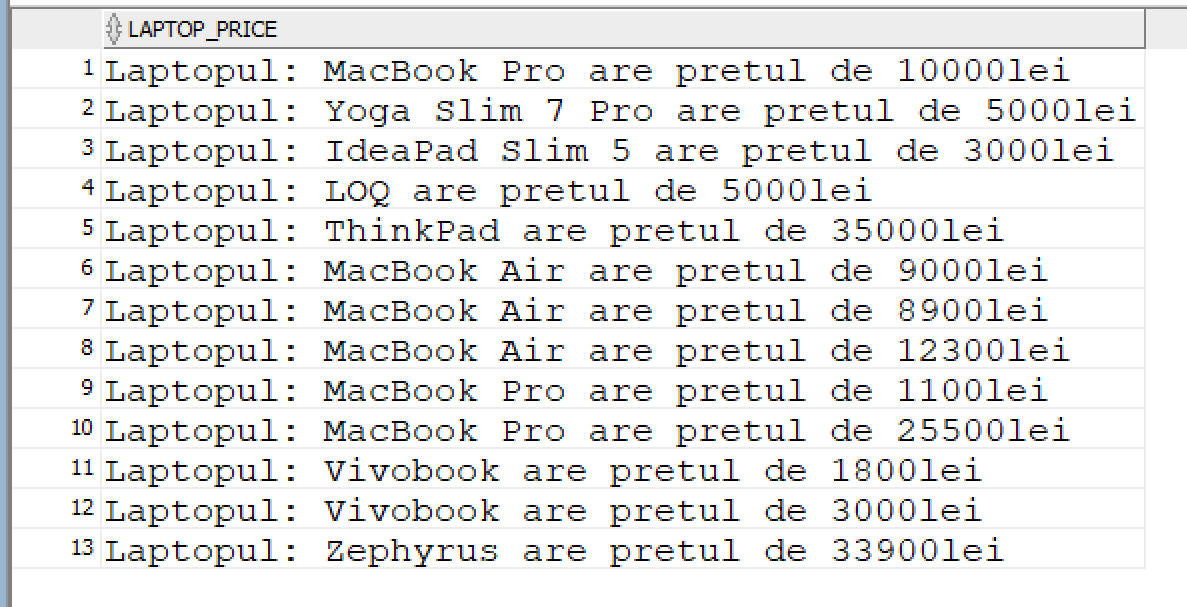


#### **3.2.4. OPERATORUL DE CONCATENARE: ||**

-- Pentru fiecare laptop sa se afiseze modelul si pretul folosind operatorul de concatenare

SELECT 'Laptopul: ' || LAPTOP\_MODEL || ' are pretul de ' || PRICE || 'lei' LAPTOP\_PRICE

FROM Z\_LAPTOPS;



#### **3.2.5. SUBSTR**

-- Afisare cate laptopuri comandate au modelul MacBook si profitul adus de acestea

SELECT

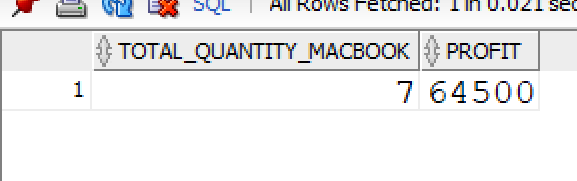
SUM(OL.QUANTITY) AS TOTAL\_QUANTITY\_MACBOOK,

SUM(OL.PRICE) AS PROFIT

FROM Z\_ORDER\_LINES OL

JOIN Z\_LAPTOPS L ON OL.LAPTOP\_ID = L.LAPTOP\_ID

WHERE LOWER(SUBSTR(L.LAPTOP\_MODEL, 1, 7)) = 'macbook';



#### **3.2.6. CONCAT SI LENGTH**

-- Afisare user id, numele de familie, numarul de telefon concatenat cu primele 2 litere din

-- numele tarii, lungimea numarului de telefon

SELECT

U.USER\_ID,

U.LAST\_NAME,

CONCAT(U.PHONE\_NUMBER, SUBSTR(CO.NAME,1,2)) AS PHONE\_COUNTRY,

LENGTH(PHONE\_NUMBER)

FROM Z\_USERS U

JOIN Z\_ADDRESSES A ON U.ADDRESS\_ID = A.ADDRESS\_ID

JOIN Z\_COUNTRIES CO ON A.COUNTRY\_ID = CO.COUNTRY\_ID;



### **3.3. FUNCȚII DE TIP DATĂ CALENDARISTICĂ**

#### **3.3.1. SYSDATE**

-- Afisare zilele dintre ultima comanda plasata si ziua de azi

SELECT ROUND(SYSDATE - ORDER\_DATE) AS DAYS

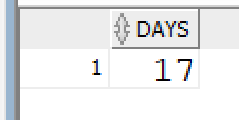
FROM Z\_LAPTOP\_ORDERS

WHERE ORDER\_DATE = (SELECT ORDER\_DATE

FROM Z\_LAPTOP\_ORDERS

ORDER BY ORDER\_DATE DESC

FETCH FIRST 1 ROW ONLY);



#### **3.3.2. MONTH\_BETWEEN() , NEXT\_DAY() , LAST\_DAY(), ADD\_MONTHS()**

-- Afisare idul comenzii, data in care a fost plasata comanda, numarul de luni intre

-- data de azi si data plasarii comenzii, urmatoarea zi de luni dupa data plasarii,

-- ultima zi din luna din care face parte data plasarii si data corespunzatoare dupa 10

-- luni de la data plasarii comenzii

SELECT

LAPTOP\_ORDER\_ID,

ORDER\_DATE,

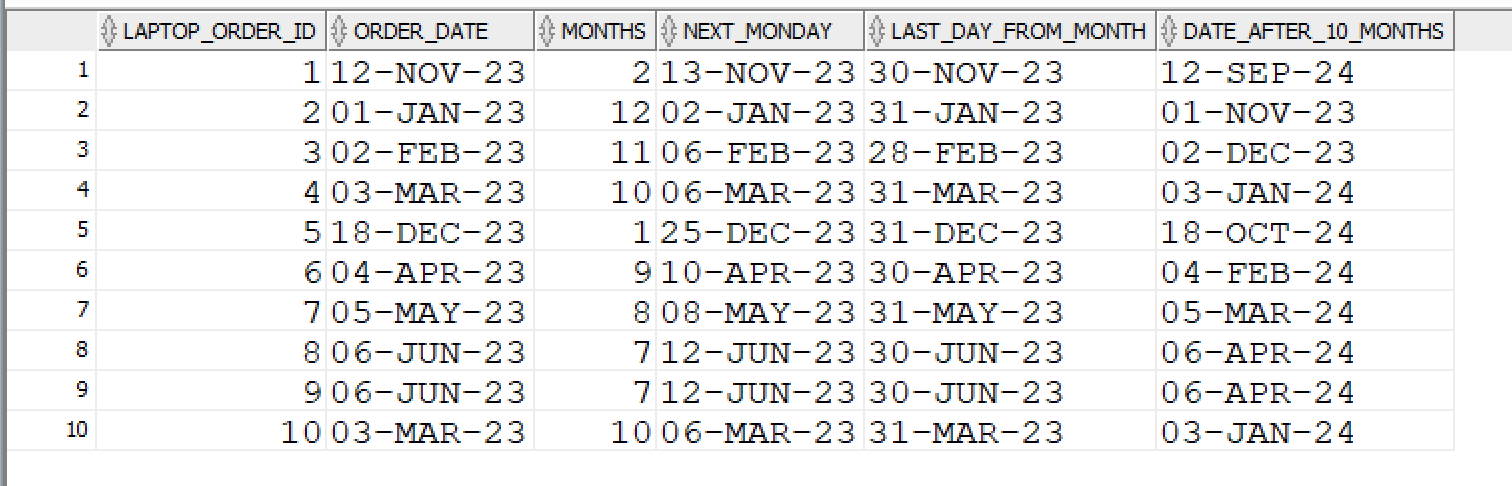
ROUND(MONTHS\_BETWEEN(SYSDATE, ORDER\_DATE)) AS MONTHS,

NEXT\_DAY(ORDER\_DATE, 'MONDAY') AS NEXT\_MONDAY,

LAST\_DAY(ORDER\_DATE) AS LAST\_DAY\_FROM\_MONTH,

ADD\_MONTHS(ORDER\_DATE, 10) AS DATE\_AFTER\_10\_MONTHS

FROM Z\_LAPTOP\_ORDERS;



## **4. REALIZAREA INTEROGĂRILOR PE BAZA UNOR CONDIȚII**

### **4.1. DECODE**

-- Categorisirea laptopurilor in functie de marimea display-ului.

-- dimensiune display < 15.3, atunci laptopul intrain categoria 'Mic'

-- dimensiune dispkay = 15.3, categorie 'Mediu'

-- dimensiune display > 15.3, categorie 'Mare'

SELECT

LAPTOP\_ID,

LAPTOP\_MODEL,

DISPLAY\_SIZE,

DECODE(

SIGN(DISPLAY\_SIZE - 15.3),

-1, 'Small',

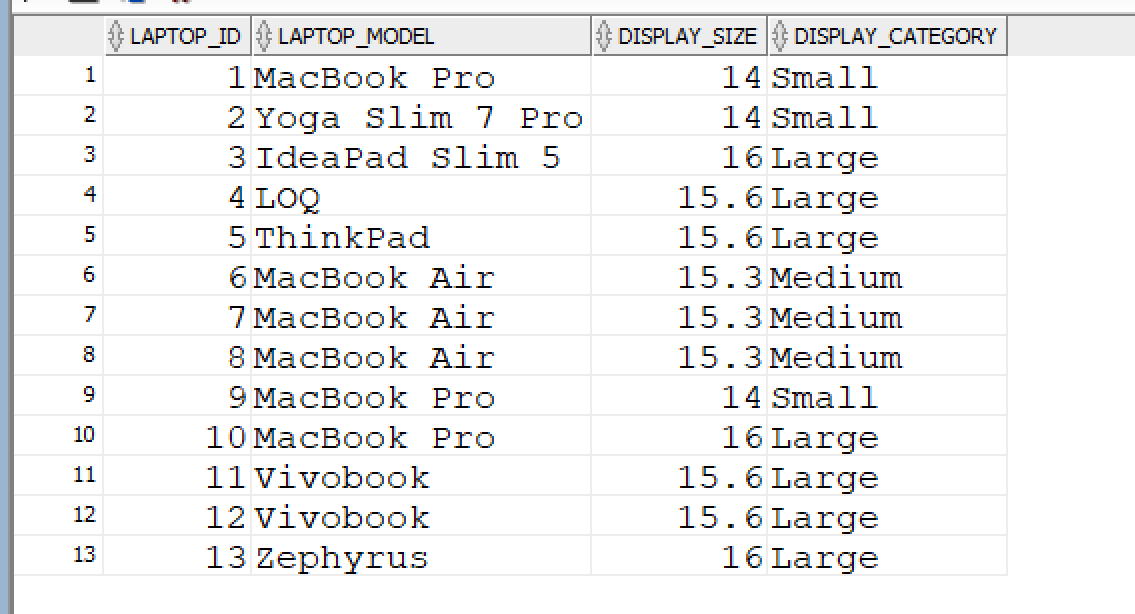
0, 'Medium',

1, 'Large'

) AS DISPLAY\_CATEGORY

FROM

Z\_LAPTOPS;



### **4.2. CASE**

-- Categorisirea laptopurilor in functie de dimensiunea memoriei de tip SSD

SELECT

LAPTOP\_ID,

LAPTOP\_MODEL,

SSD\_MEMORY\_SIZE,

CASE

WHEN SSD\_MEMORY\_SIZE <= 256 THEN 'Small'

WHEN SSD\_MEMORY\_SIZE >= 512 AND SSD\_MEMORY\_SIZE <= 512 THEN 'Medium'

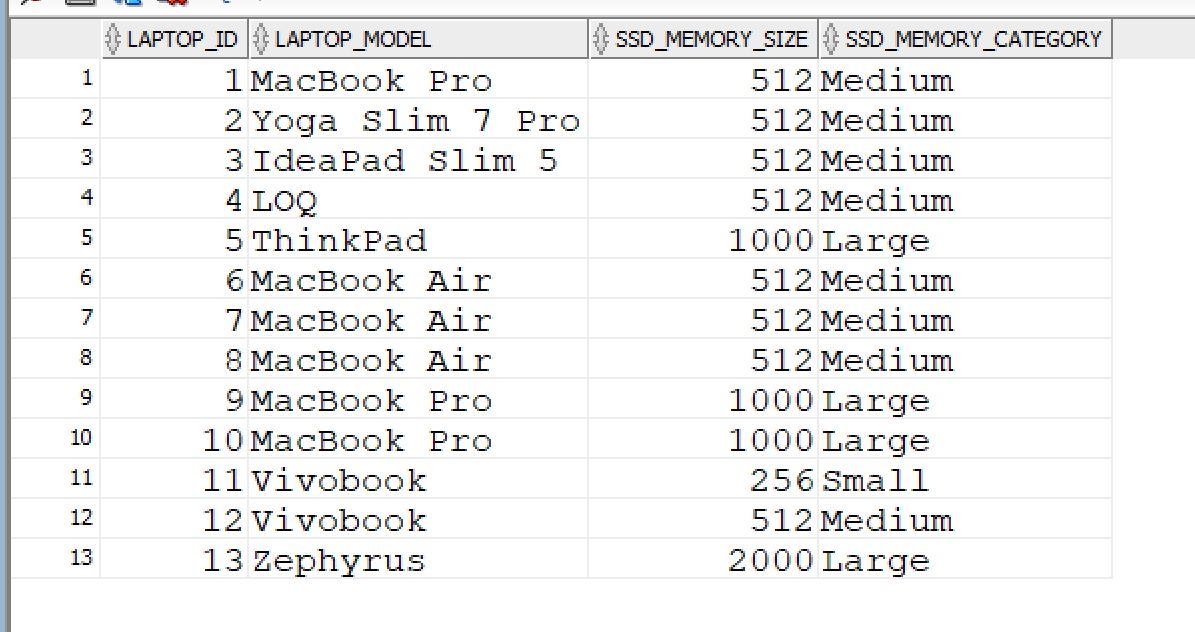
WHEN SSD\_MEMORY\_SIZE >= 1000 THEN 'Large'

ELSE 'Unknown'

END AS SSD\_MEMORY\_CATEGORY

FROM

Z\_LAPTOPS;



## **5. OPERATORII ALGEBREI RELAȚIONALE**

### **5.1. MINUS**

-- Selectarea userilor care nu au o tara

SELECT U.USER\_ID, U.FIRST\_NAME, CO.NAME AS COUNTRY\_NAME

FROM Z\_USERS U

LEFT JOIN Z\_ADDRESSES A ON U.ADDRESS\_ID = A.ADDRESS\_ID

LEFT JOIN Z\_COUNTRIES CO ON A.COUNTRY\_ID = CO.COUNTRY\_ID

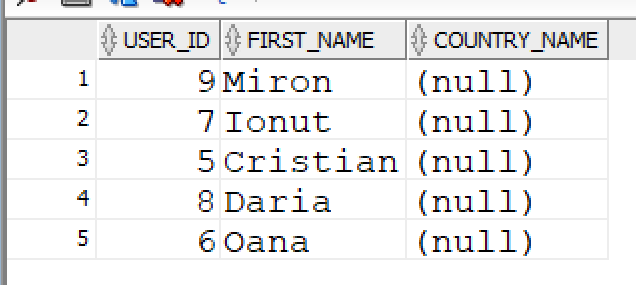
MINUS

SELECT USER\_ID, FIRST\_NAME, CO.NAME AS COUNTRY\_NAME

FROM Z\_COUNTRIES CO

JOIN Z\_ADDRESSES A ON CO.COUNTRY\_ID = A.COUNTRY\_ID

JOIN Z\_USERS U ON A.ADDRESS\_ID = U.ADDRESS\_ID;



### **5.2. INTERSECT**

-- Selectarea userilor cu id-ul cuprins intre 1 si 4 si al caror prenume incepe cu litera M

SELECT USER\_ID, FIRST\_NAME

FROM Z\_USERS

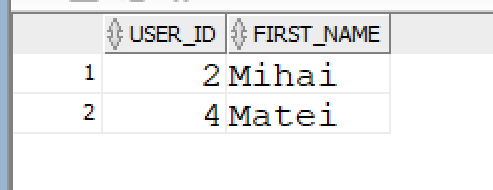
WHERE USER\_ID BETWEEN 1 AND 4

INTERSECT

SELECT USER\_ID, FIRST\_NAME

FROM Z\_USERS

WHERE FIRST\_NAME LIKE 'M%';



### **5.3. UNION**

-- Selectarea userilor din Romania si a celor din Italia, utilizand UNION

SELECT U.USER\_ID, U.FIRST\_NAME, U.LAST\_NAME, CO.NAME

FROM Z\_USERS U

JOIN Z\_ADDRESSES A ON U.ADDRESS\_ID = A.ADDRESS\_ID

JOIN Z\_COUNTRIES CO ON A.COUNTRY\_ID = CO.COUNTRY\_ID

WHERE UPPER(CO.NAME) = 'ROMANIA'

UNION

SELECT U.USER\_ID, U.FIRST\_NAME, U.LAST\_NAME, CO.NAME

FROM Z\_USERS U

JOIN Z\_ADDRESSES A ON U.ADDRESS\_ID = A.ADDRESS\_ID

JOIN Z\_COUNTRIES CO ON A.COUNTRY\_ID = CO.COUNTRY\_ID

WHERE UPPER(CO.NAME) = 'ITALIA';

## **6. SUBCERERI**

-- Afisare cel mai scump laptop si cantitatea in care a fost cumparat

SELECT

L.LAPTOP\_ID,

LB.NAME AS BRAND\_NAME,

L.LAPTOP\_MODEL,

L.PRICE,

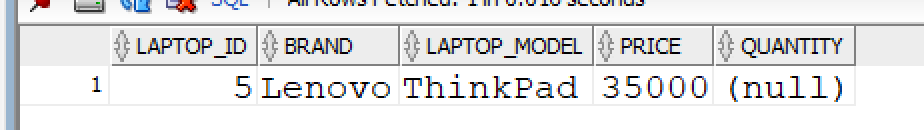
OL.QUANTITY

FROM Z\_LAPTOPS L

JOIN Z\_LAPTOP\_BRANDS LB ON L.LAPTOP\_BRAND\_ID = LB.LAPTOP\_BRAND\_ID

LEFT JOIN Z\_ORDER\_LINES OL ON L.LAPTOP\_ID = OL.LAPTOP\_ID

WHERE L.PRICE = (SELECT MAX(PRICE) FROM Z\_LAPTOPS);



-- Afisare laptopul care a generat cele mai multe vanzari

SELECT

L.LAPTOP\_ID,

LB.NAME AS BRAND,

L.LAPTOP\_MODEL,

SUM(OL.QUANTITY\*OL.PRICE) AS GENERATED\_SALES\_VALUE

FROM Z\_LAPTOPS L

JOIN Z\_LAPTOP\_BRANDS LB ON L.LAPTOP\_BRAND\_ID = LB.LAPTOP\_BRAND\_ID

JOIN Z\_ORDER\_LINES OL ON L.LAPTOP\_ID = OL.LAPTOP\_ID

WHERE L.LAPTOP\_ID =(SELECT LAPTOP\_ID

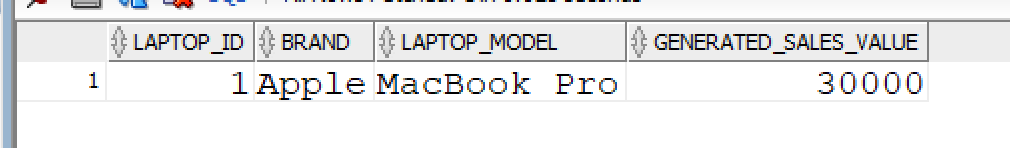
FROM Z\_ORDER\_LINES

GROUP BY LAPTOP\_ID

ORDER BY SUM(QUANTITY\*PRICE) DESC

FETCH FIRST 1 ROW ONLY)

GROUP BY L.LAPTOP\_ID, LB.NAME, L.LAPTOP\_MODEL;



-- Afisare toate laptopurile care au un pret mai mare ca pretul mediu per laptop si cantitatea

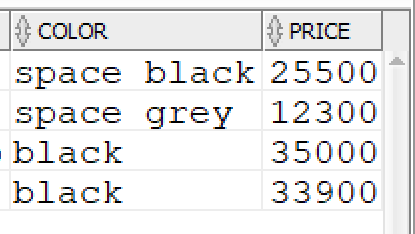
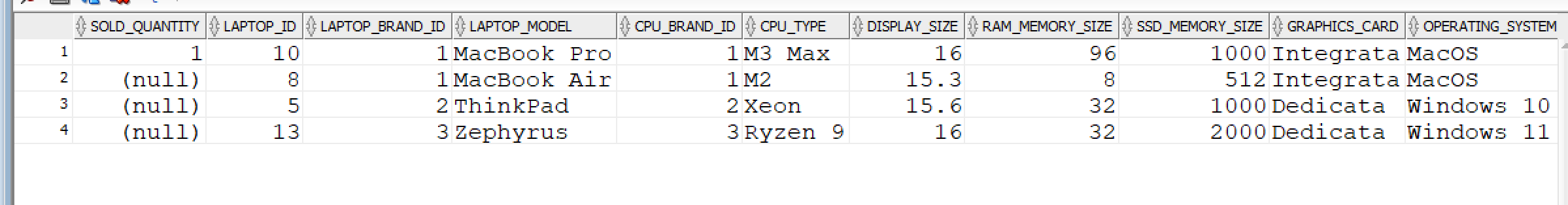
-- vanduta pentru fiecare

SELECT OL.QUANTITY AS SOLD\_QUANTITY, L.\*

FROM Z\_LAPTOPS L

LEFT JOIN Z\_ORDER\_LINES OL ON L.LAPTOP\_ID = OL.LAPTOP\_ID

WHERE L.PRICE > (SELECT AVG(LA.PRICE) FROM Z\_LAPTOPS LA);



-- Afisare laptopuri care au un pret mai mare fata de pretul mediu al laptopurilor

-- cu acelasi brand

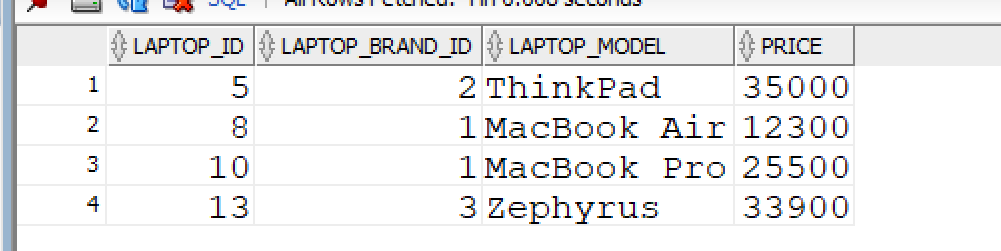
SELECT L.LAPTOP\_ID, L.LAPTOP\_BRAND\_ID, L.LAPTOP\_MODEL, L.PRICE

FROM Z\_LAPTOPS L

WHERE L.PRICE > (SELECT AVG(LB.PRICE)

FROM Z\_LAPTOPS LB

WHERE LB.LAPTOP\_BRAND\_ID = L.LAPTOP\_BRAND\_ID);



-- Afisare toti userii si cati bani au dat pe laptopuri, dar si pe livrare

SELECT

U.USER\_ID,

U.FIRST\_NAME,

U.LAST\_NAME,

(SELECT SUM(OL.PRICE \* OL.QUANTITY)

FROM Z\_ORDER\_LINES OL

JOIN Z\_LAPTOP\_ORDERS LO ON OL.LAPTOP\_ORDER\_ID = LO.LAPTOP\_ORDER\_ID

WHERE LO.USER\_ID = U.USER\_ID) AS AMOUNT\_SPENT\_ON\_LAPTOPS,

(SELECT SUM(SM.PRICE)

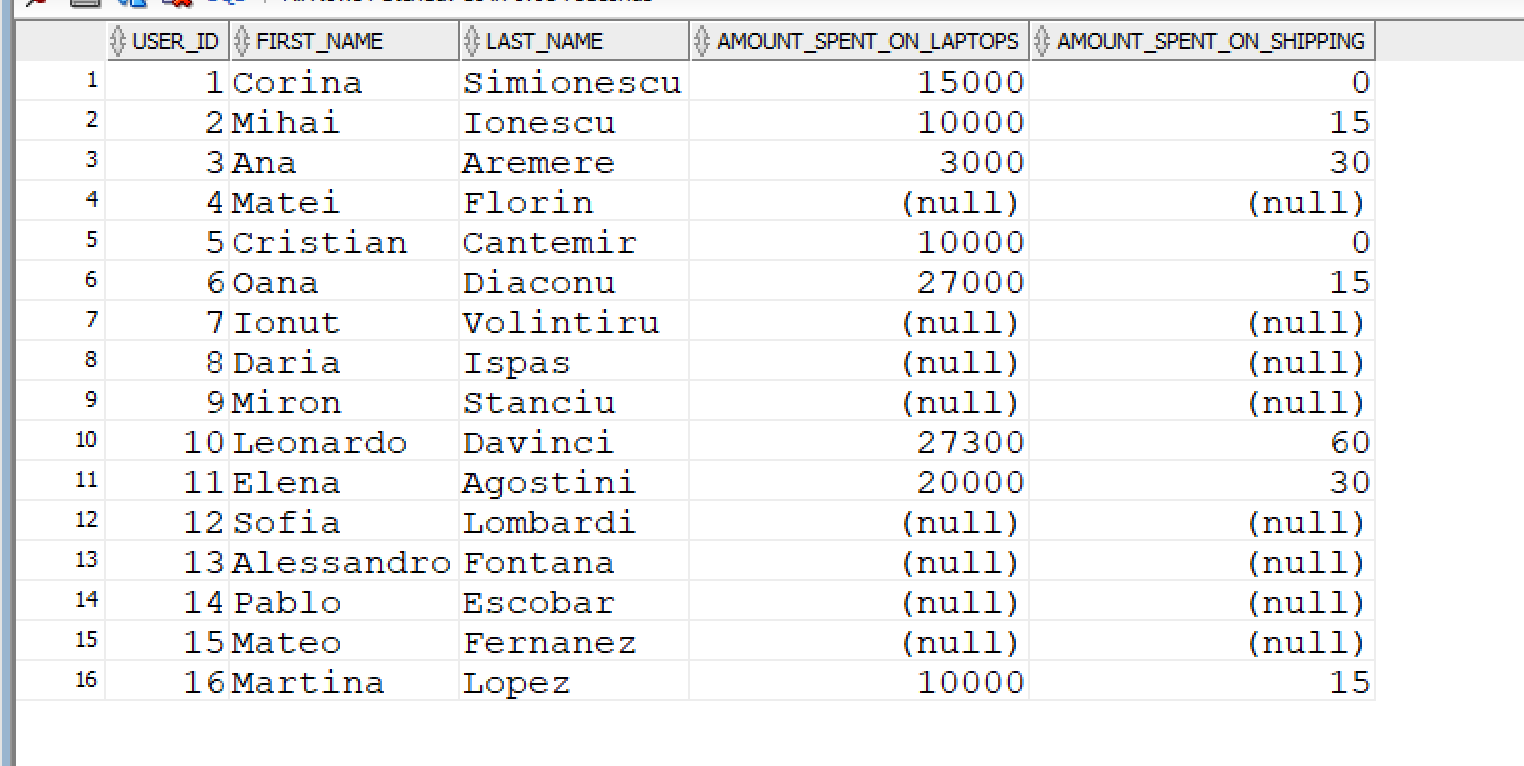
FROM Z\_SHIPPING\_METHODS SM

JOIN Z\_LAPTOP\_ORDERS LO ON SM.SHIPPING\_METHOD\_ID = LO.SHIPPING\_METHOD\_ID

WHERE LO.USER\_ID = U.USER\_ID) AS AMOUNT\_SPENT\_ON\_SHIPPING

FROM Z\_USERS U

ORDER BY U.USER\_ID ASC;



# **GESTIUNEA ALTOR OBIECTE ALE BAZEI DE DATE**

## **1. TABELE VIRTUALE**

### **1.1. CREARE TABELĂ VIRTUALĂ**

-- Crearea unei tabele virtuale cu urmatoarele coloane:

-- id-ul comenzii,id-ul userului, pretul total al laptopurilor din comanda,

-- pretul metodei de livrare si pretul total

CREATE OR REPLACE VIEW ORDERS\_PRICES\_DETAILS AS

SELECT

OL.LAPTOP\_ORDER\_ID,

LO.USER\_ID,

SUM(OL.PRICE\*OL.QUANTITY) AS LAPTOPS\_PRICE,

SM.PRICE AS SHIPPING\_PRICE,

SUM(OL.PRICE\*OL.QUANTITY) + SM.PRICE AS TOTAL\_PRICE

FROM Z\_LAPTOP\_ORDERS LO

JOIN Z\_ORDER\_LINES OL ON LO.LAPTOP\_ORDER\_ID = OL.LAPTOP\_ORDER\_ID

JOIN Z\_SHIPPING\_METHODS SM ON LO.SHIPPING\_METHOD\_ID = SM.SHIPPING\_METHOD\_ID

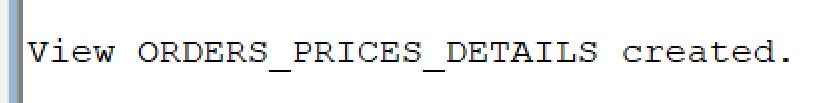
GROUP BY

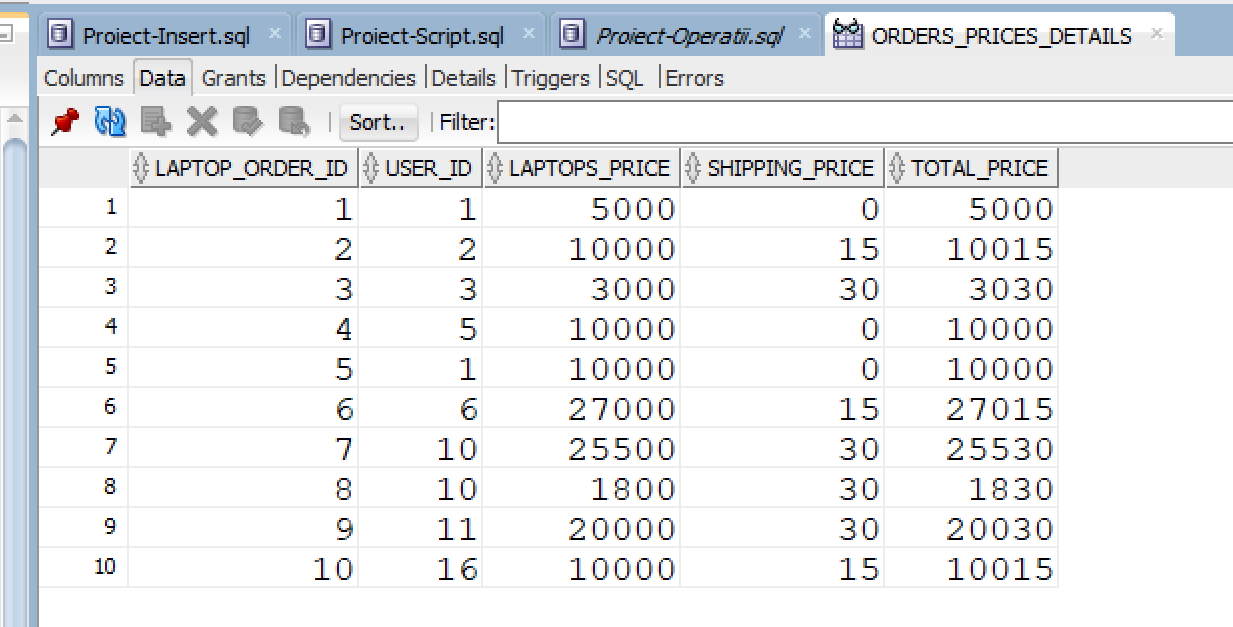
OL.LAPTOP\_ORDER\_ID,

LO.USER\_ID,

SM.PRICE

ORDER BY LAPTOP\_ORDER\_ID;





-- Crearea unei tabele virtuale cu urmatoarele coloane:

-- id user, nume, prenume, tara si orasul din adresa

CREATE OR REPLACE VIEW USERS\_COUNTRY\_CITY AS

SELECT

U.USER\_ID,

U.FIRST\_NAME,

U.LAST\_NAME,

CO.NAME AS COUNTRY,

CI.NAME AS CITY

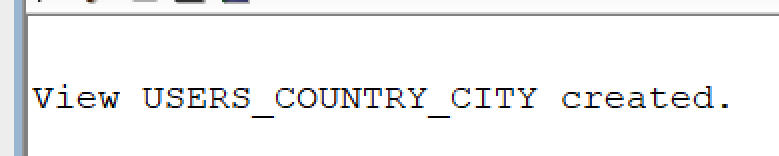
FROM Z\_USERS U

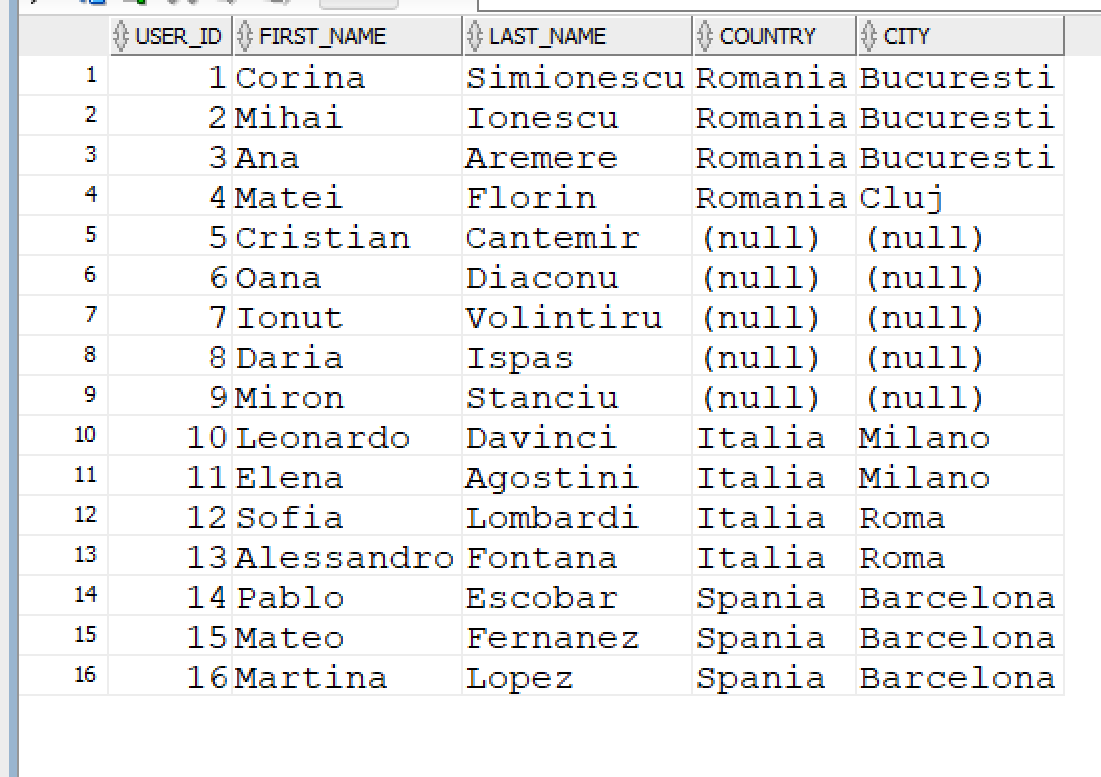
LEFT JOIN Z\_ADDRESSES A ON U.ADDRESS\_ID = A.ADDRESS\_ID

LEFT JOIN Z\_COUNTRIES CO ON A.COUNTRY\_ID = CO.COUNTRY\_ID

LEFT JOIN Z\_CITIES CI ON A.CITY\_ID = CI.CITY\_ID

ORDER BY U.USER\_ID ASC;





### **1.2. INTEROGARE TABELĂ VIRTUALĂ**

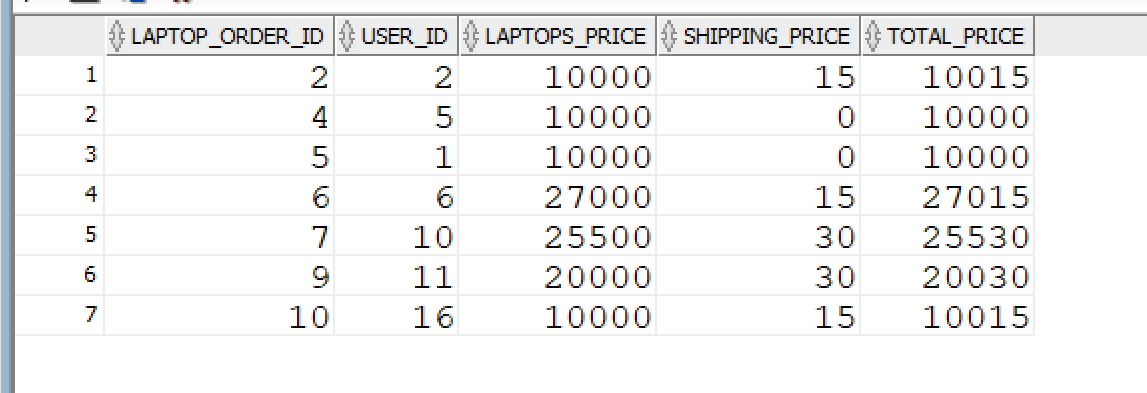
-- Afisare toti userii pt care suma totala a comenzilor este mai mare de 10000 de lei

SELECT \*

FROM ORDERS\_PRICES\_DETAILS

WHERE TOTAL\_PRICE >= 10000

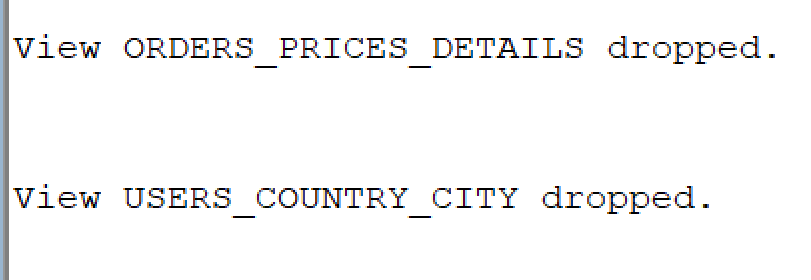
ORDER BY LAPTOP\_ORDER\_ID;



### **1.3. STERGERE TABELĂ VIRTUALĂ**

DROP VIEW ORDERS\_PRICES\_DETAILS;

DROP VIEW USERS\_COUNTRY\_CITY;

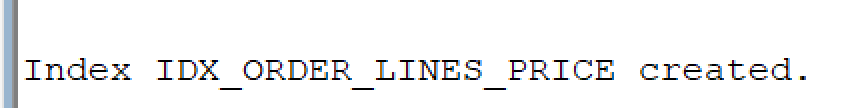


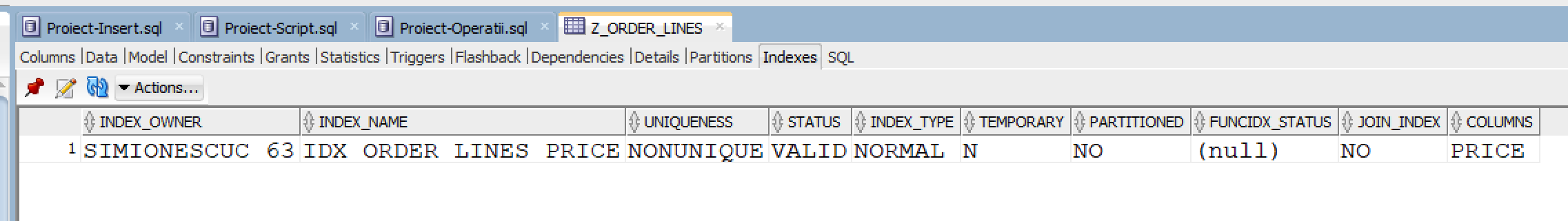
## **2. INDECȘI**

### **2.1. CREARE INDEX**

-- Creare index pentru coloana "price" din tabelul Z\_ORDER\_LINES

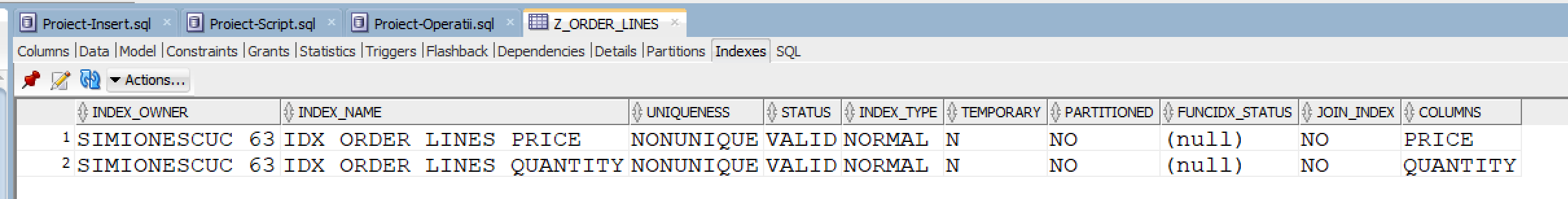
CREATE INDEX IDX\_ORDER\_LINES\_PRICE ON Z\_ORDER\_LINES(PRICE);





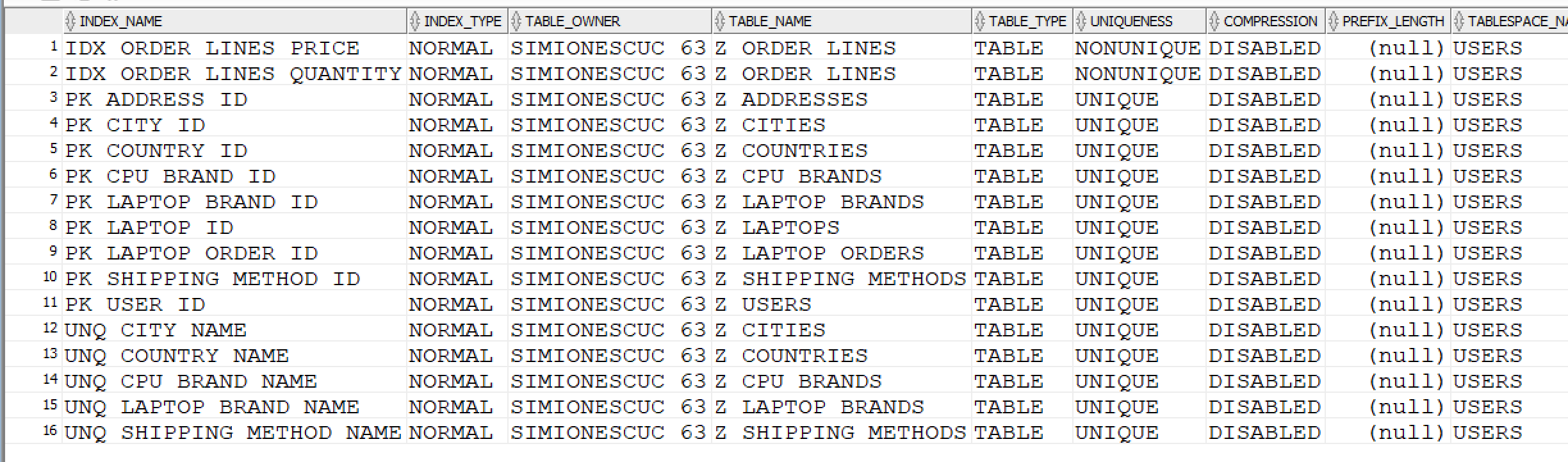
-- Creare index pentru coloana "quantity" din tabelul Z\_ORDER\_LINES

CREATE INDEX IDX\_ORDER\_LINES\_QUANTITY ON Z\_ORDER\_LINES(QUANTITY);



### **2.2. VIZUALIZARE INDECȘI**

SELECT \* FROM USER\_INDEXES;



### **2.3. EFICIENTA INDECȘILOR**

-- Afisare useri si valoarea laptopurilor comandate daca pretul >= 10000 si cantitatea >= 1

SELECT

U.USER\_ID,

U.FIRST\_NAME,

U.LAST\_NAME,

SUM(OL.PRICE\*OL.QUANTITY)

FROM Z\_ORDER\_LINES OL

JOIN Z\_LAPTOP\_ORDERS LO ON OL.LAPTOP\_ORDER\_ID = LO.LAPTOP\_ORDER\_ID

JOIN Z\_USERS U ON LO.USER\_ID = U.USER\_ID

WHERE

OL.PRICE >= 10000

AND

OL.QUANTITY >= 1

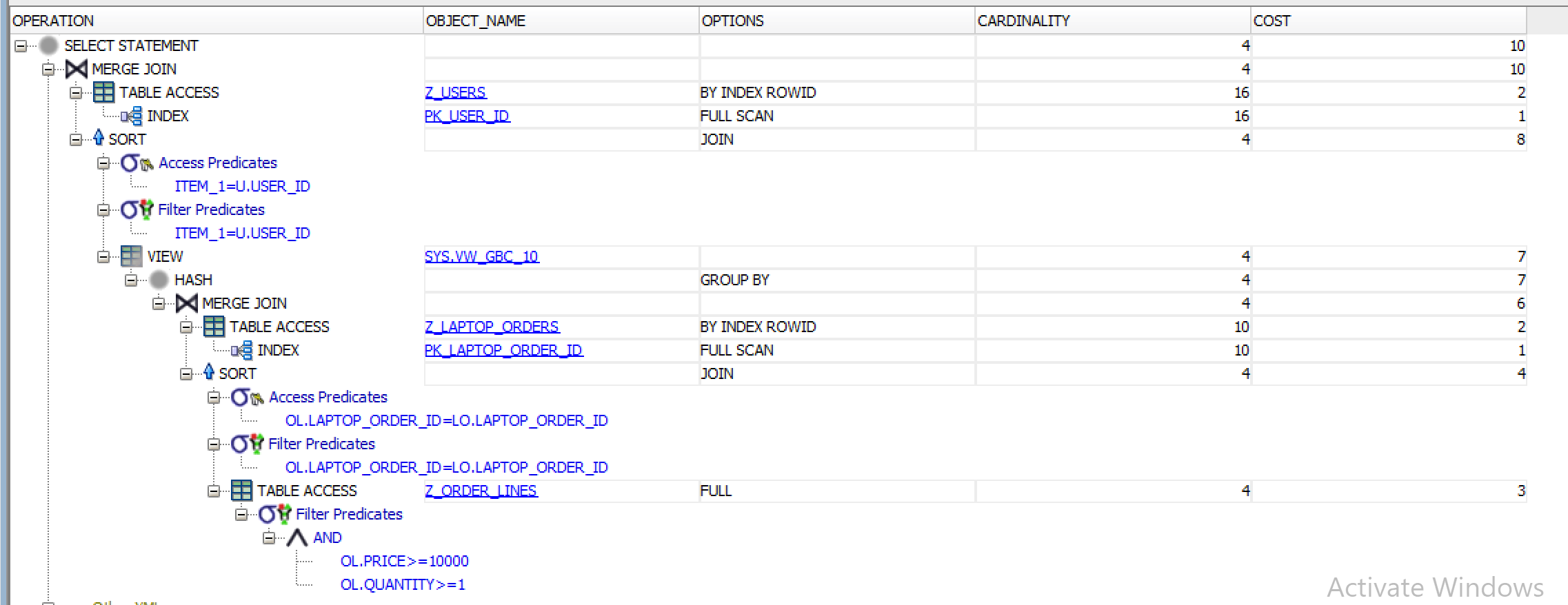
GROUP BY

U.USER\_ID,

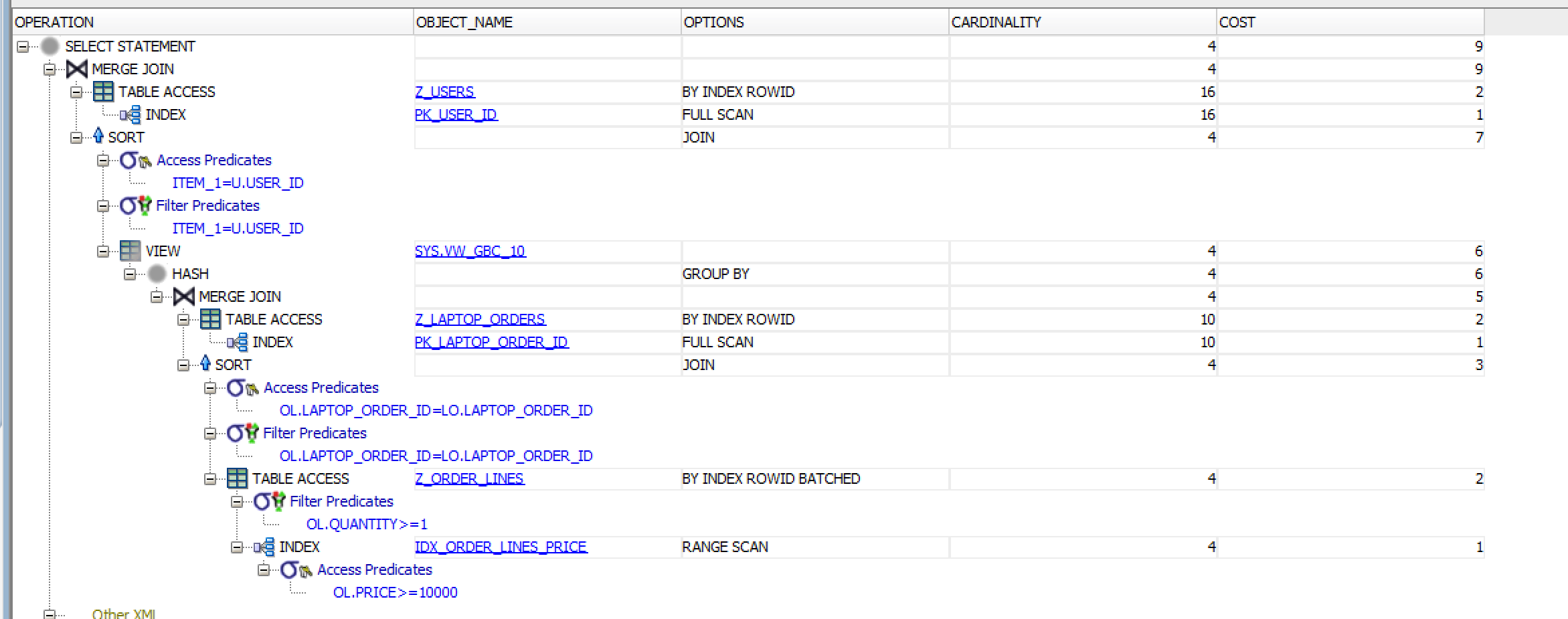
U.FIRST\_NAME,

U.LAST\_NAME;

**Costul de execuție inainte de crearea indecșilor:**



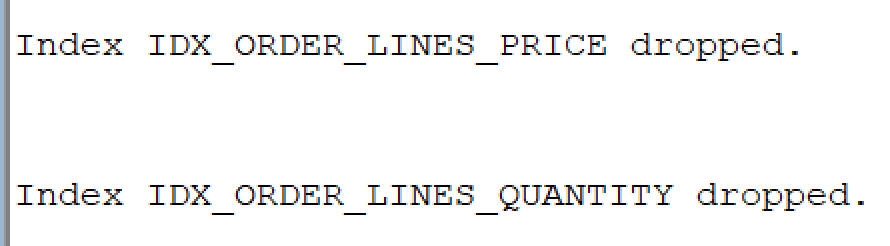
**Costul de execuție după de crearea indecșilor:**



### **2.4. STERGERE INDECȘI**

DROP INDEX IDX\_ORDER\_LINES\_PRICE;

DROP INDEX IDX\_ORDER\_LINES\_QUANTITY;

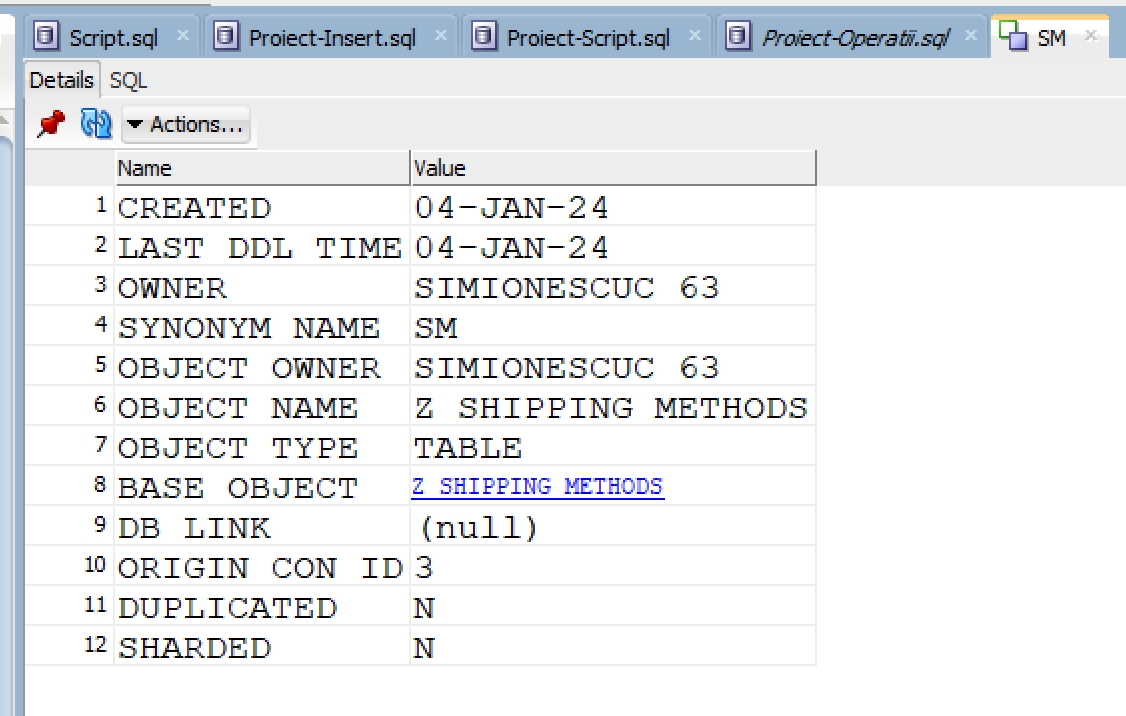
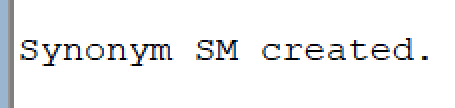


## **3. SINONIME**

### **3.1. CREARE SINONIM**

-- Creare sinonim privat pentru tabela Z\_SHIPPING\_METHODS

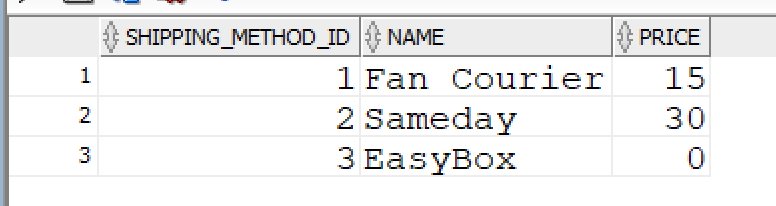
CREATE OR REPLACE SYNONYM SM FOR Z\_SHIPPING\_METHODS;



### **3.2. UTILIZAREA SINONIMULUI CREAT**

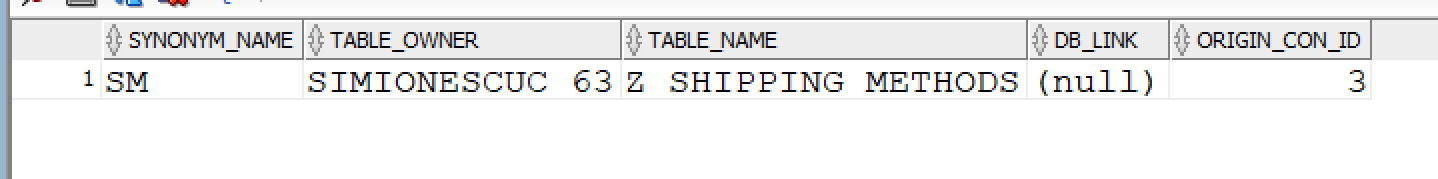
-- Afisare metode de livrare, folodind sinonimul creat anterior

SELECT \* FROM SM;



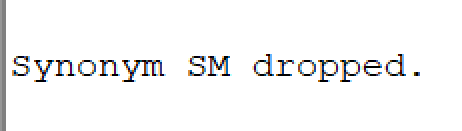
### **3.3. VIZUALIZARE SINONIME**

SELECT \* FROM USER\_SYNONYMS;



### **3.4. STERGERE SINONIM**

DROP SYNONYM SM;



## **4. SECVENȚE**

### **4.1. CREARE SECVENȚĂ**

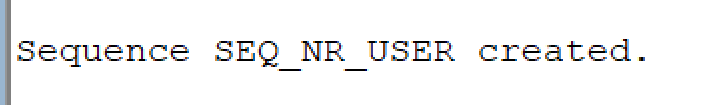
-- Crearea unei secvente pentru asigurarea unicitatii cheii primare din tabela

-- Z\_USERS

CREATE SEQUENCE SEQ\_NR\_USER

START WITH 100 INCREMENT BY 1

MAXVALUE 1000 NOCYCLE;



### **4.2. UTILIZAREA SECVENȚEI CREATE**

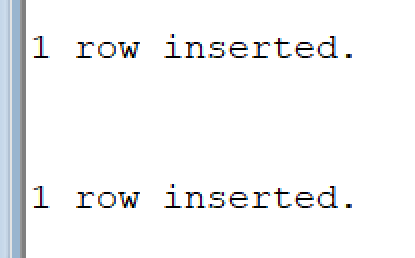
-- Sa se insereze 2 useri in tabela Z\_USERS, folosind secventa creata anterior

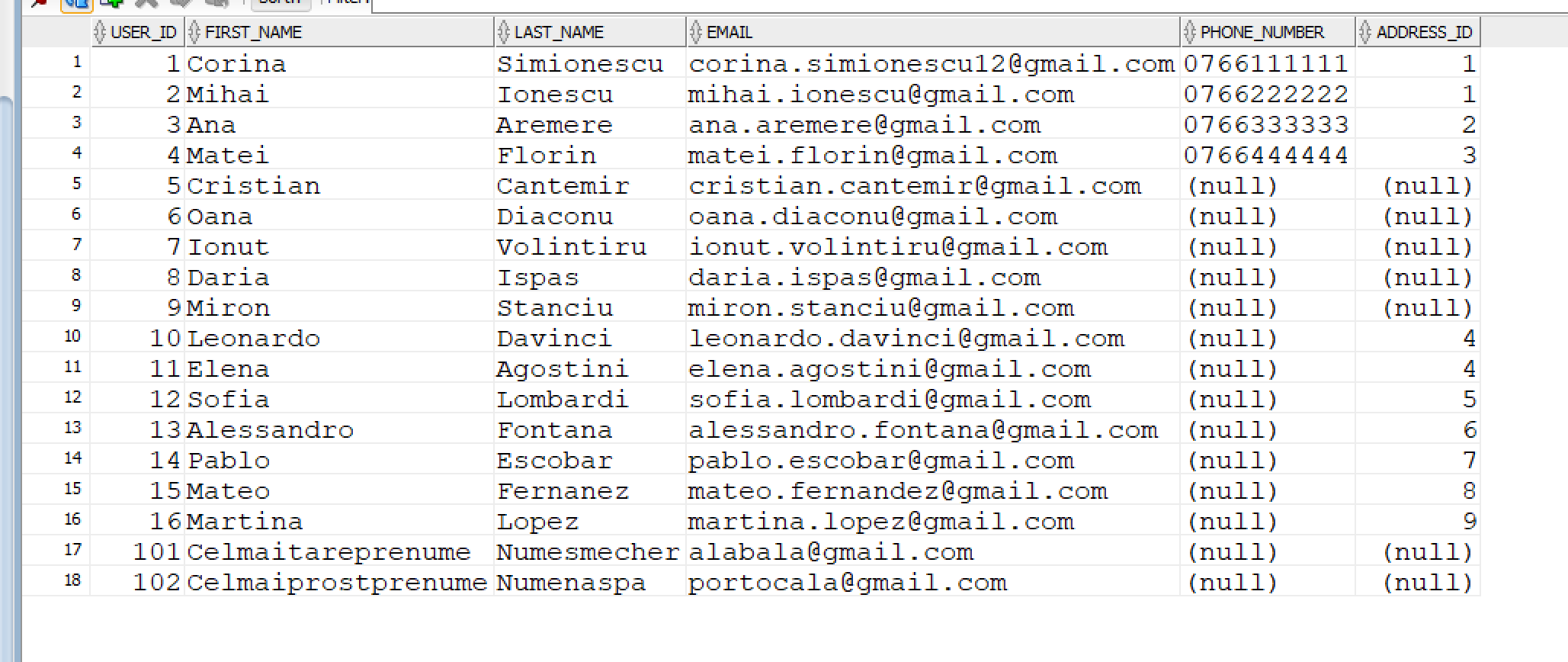
INSERT INTO Z\_USERS (USER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL)

VALUES(SEQ\_NR\_USER.NEXTVAL, 'Celmaitareprenume', 'Numesmecher', 'alabala@gmail.com');

INSERT INTO Z\_USERS (USER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL)

VALUES(SEQ\_NR\_USER.NEXTVAL, 'Celmaiprostprenume', 'Numenaspa', 'portocala@gmail.com');

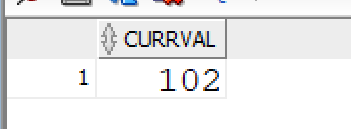




### **4.3. AFISARE VALOARE CURENTĂ A SECVENȚEI**

-- Afisare valoare curenta a secventei create anterior

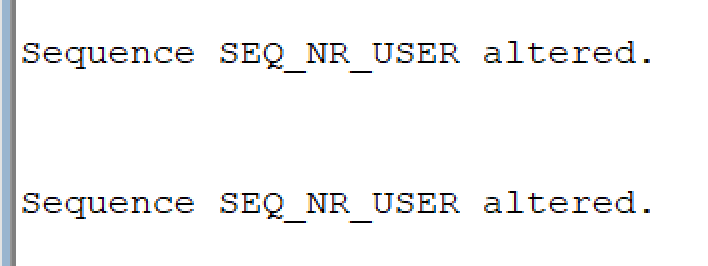
SELECT SEQ\_NR\_USER.CURRVAL FROM DUAL;



-- Modificarea pasului de incrementare si valoarea maxima a secventei anterioare

ALTER SEQUENCE SEQ\_NR\_USER INCREMENT BY 10;

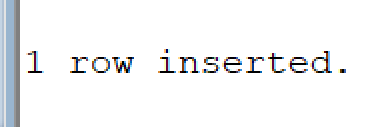
ALTER SEQUENCE SEQ\_NR\_USER MAXVALUE 5000;

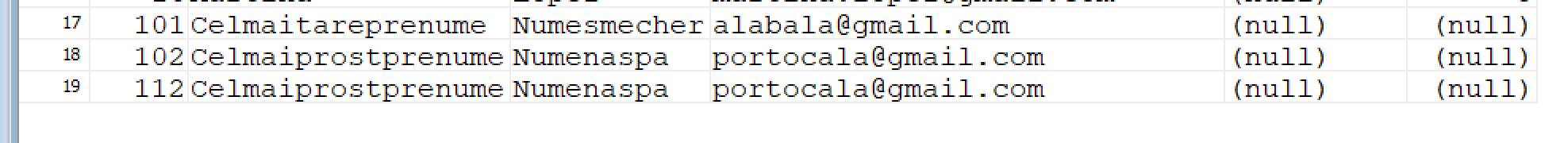


-- Exemplificare ca s-a marit pasul de incrementare

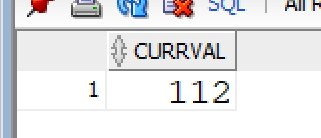
INSERT INTO Z\_USERS (USER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL)

VALUES(SEQ\_NR\_USER.NEXTVAL, 'Celmaiprostprenume', 'Numenaspa', 'portocala@gmail.com');





SELECT SEQ\_NR\_USER.CURRVAL FROM DUAL;



### **4.4. STERGERE SECVENȚĂ**

DROP SEQUENCE SEQ\_NR\_USER;

