

Proiect - Baze de date

Gestiunea unui Service GSM

Ailioaei Sorina-Elena

Seria C, Grupa 1051

1.Descrierea temei

În baza de date sunt organizate, după modelul relațional, datele cu privire la activitatea unei firme de reparații de telefoane mobile și anume, a unui Service GSM.

Această bază de date conține informații despre telefoanele mobile defecte, ce urmează a fi reparate, despre componenta telefonului ce trebuie schimbată, detalii despre client, vânzările pentru fiecare client în parte, și, de asemenea, alte accesorii ce pot fi achiziționate de aceștia, indiferent dacă doresc repararea telefonului sau nu.

Prima tabela este TELEFON, unde se găsesc următoarele atribute: COD_TELEFON (codul telefonului adus la reparat, care este și cheie primară în această tabelă), BRAND, MODEL, DEFECTIONE.

A doua tabelă este denumită CLIENT_TEL și conține detalii referitoare la client cu următoarele coloane: COD_CLIENT(cheie primară), COD_TELEFON(cheie externă ce face legătură cu tabela TELEFON), NUME, PRENUME, NR_CONTACT. Telefonul este unic și va aparține unui singur client, însă, clientul poate să dețină mai multe telefoane, aceasta fiind o relație 1-n.

A treia tabelă o reprezintă COMPONENTA_SCHIMBATA, având detalii despre componenta defectă a telefonului ce trebuie schimbată, cu următoarele coloane: COD_COMP (cheie primară), DENUMIRE, PRET.

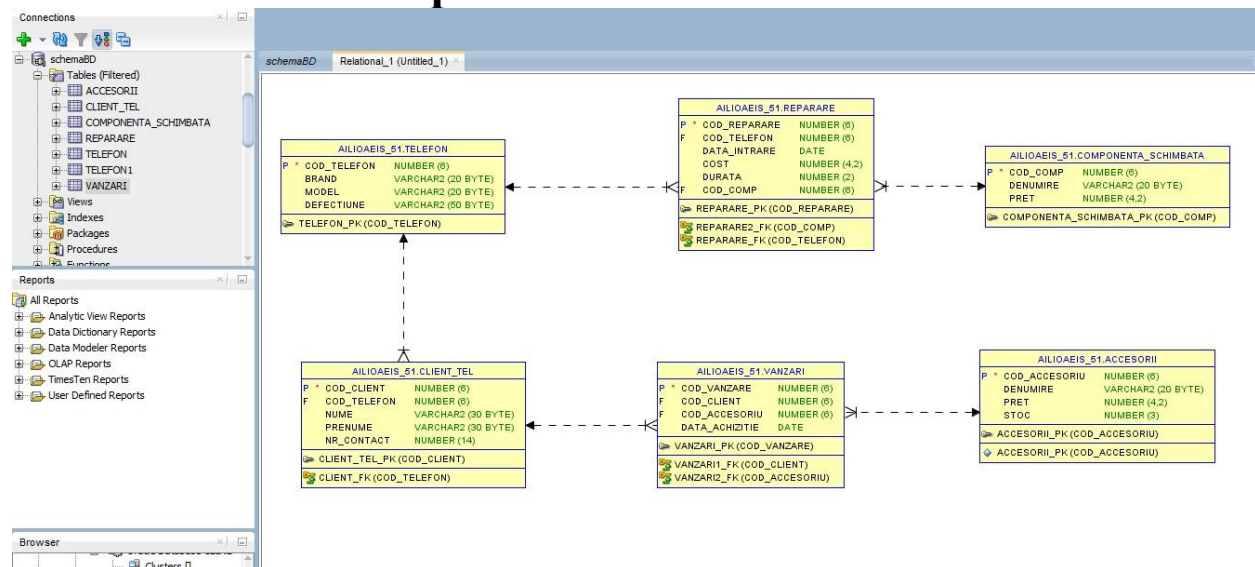
A patra tabelă denumită REPARARE, aduce informații despre procesul de reparare, cum ar fi: COD_REPARARE (cheie primară), COD_TELEFON(cheie externă ce face legătură cu tabela TELEFON), COD_COMP(cheie externă ce face legătură cu tabela COMPONENTA_SCHIMBATA), DATA_INTRARE(regăsim data în care telefonul a fost adus la reparat), COST(costul suplimentar, aferent procesului de reparație, un cost în afară de prețul componentei schimbate), DURATA(numărul de zile necesare angajatului pentru a repara

telefonul). Între această tabelă și TELEFON avem o relație 1:1 și anume, repararea se face pe un singur telefon, si invers, un telefon este supus unei singure reparări.

Următoarea tabelă, ACCESORII, cuprinde anumite accesorii pentru telefoane mobile cum ar fi: huse, încărcătoare, etc., de care dispune firma respectiva și care sunt puse la vânzare pentru clienți indiferent dacă au un telefon de reparat sau nu. Tabela are următoarele atribute: COD_ACCESORIU(cheie primară), DENUMIRE, PREȚ.

Ultima tabelă, VANZARI, reține detalii despre vânzarea de accesorii făcută cu fiecare client în parte, aceasta deține atributele: COD_VANZARE(cheie primară), COD_CLIENT(cheie externă ce face legătură cu tabela CLIENT_TEL), COD_ACCESORIU(cheie externă ce face legătură cu tabela ACCESORII), DATA.

2.Schema conceptuală



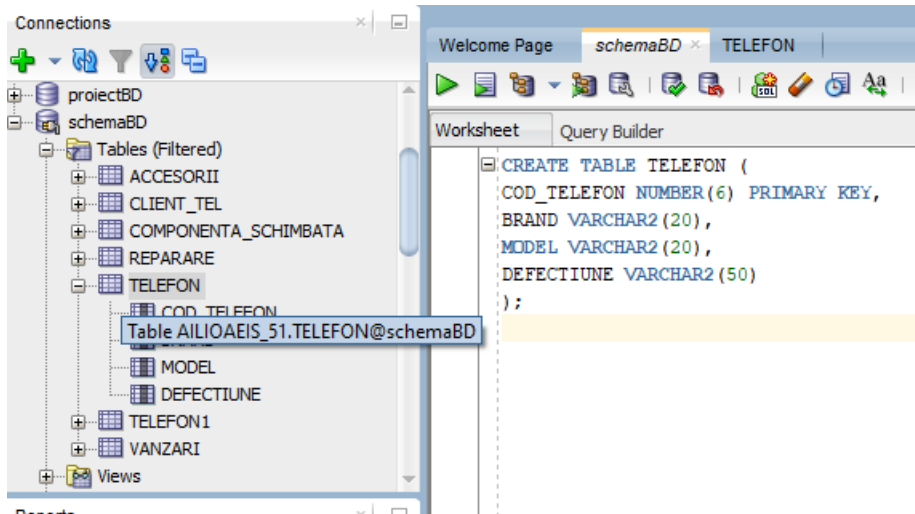
3. Construirea bazei de date – tabele și restricții de integritate.

Construirea celor 6 tabele:

1. Tabela TELEFON:

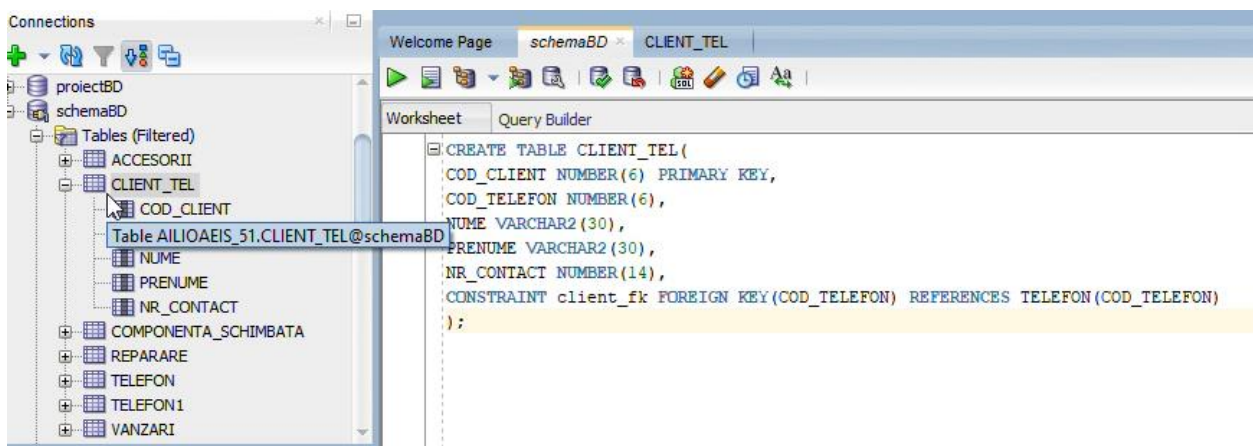
```

CREATE TABLE TELEFON (
    COD_TELEFON NUMBER(6) PRIMARY KEY,
    BRAND VARCHAR2(20),
    MODEL VARCHAR2(20),
    DEFECTIUNE VARCHAR2(50)
);
    
```



2. Tabela CLIENT_TEL

```
CREATE TABLE CLIENT_TEL(
COD_CLIENT NUMBER(6) PRIMARY KEY,
COD_TELEFON NUMBER(6),
NUME VARCHAR2(30),
PRENUME VARCHAR2(30),
NR_CONTACT NUMBER(14),
CONSTRAINT client_fk FOREIGN KEY(COD_TELEFON) REFERENCES
TELEFON(COD_TELEFON)
);
```



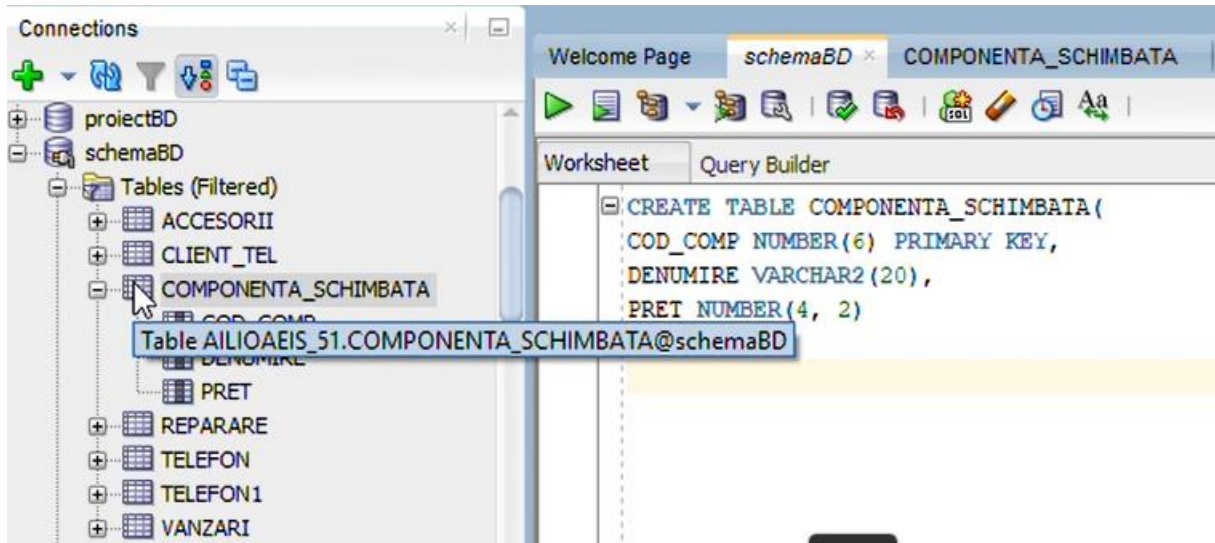
3. Tabela COMPONENTA_SCHIMBATA

```
CREATE TABLE COMPONENTA_SCHIMBATA(
COD_COMP NUMBER(6) PRIMARY KEY,
```

```

DENUMIRE VARCHAR2(20),
PRET NUMBER(4, 2)
);

```

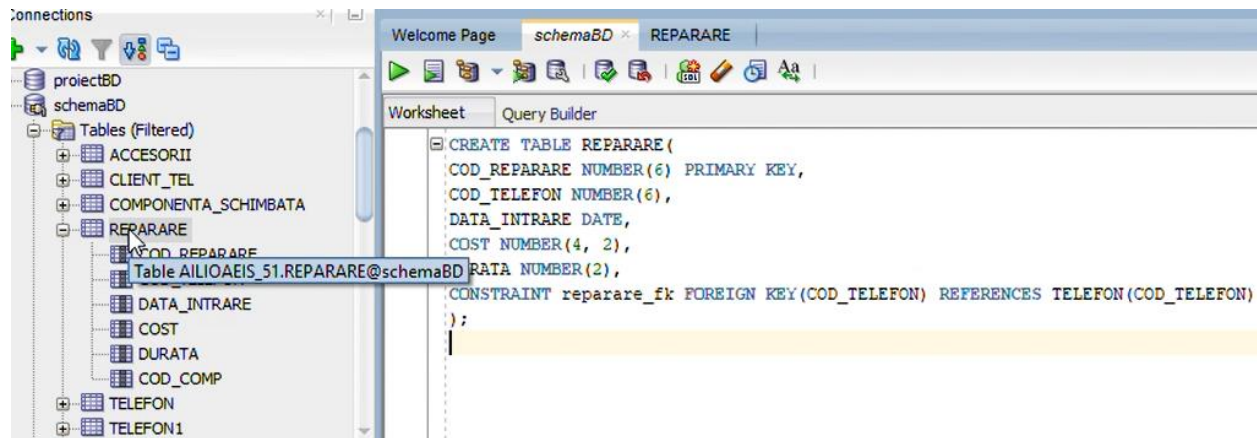


4. Tabela REPARARE

```

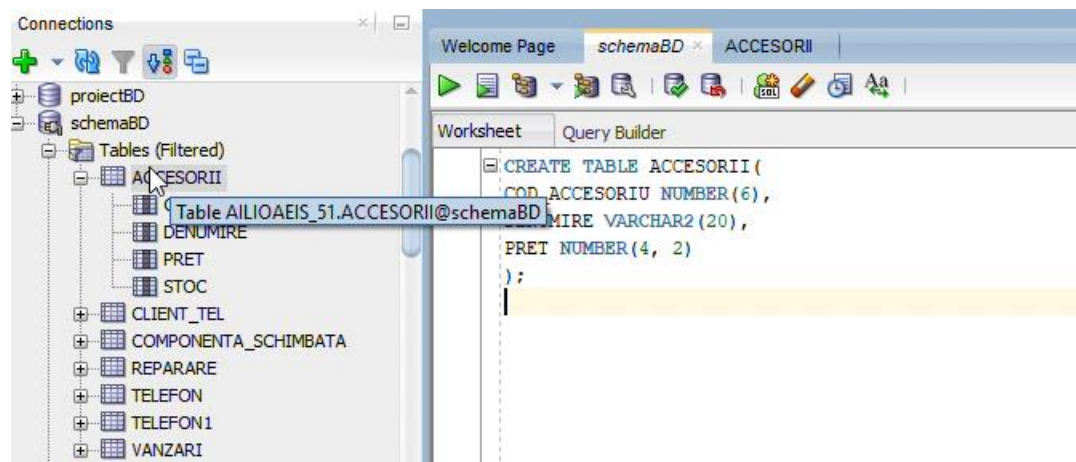
CREATE TABLE REPARARE(
  COD_REPARARE NUMBER(6) PRIMARY KEY,
  COD_TELEFON NUMBER(6),
  DATA_INTRARE DATE,
  COST NUMBER(4, 2),
  DURATA NUMBER(2),
  CONSTRAINT reparare_fk FOREIGN KEY(COD_TELEFON) REFERENCES
  TELEFON(COD_TELEFON)
);

```



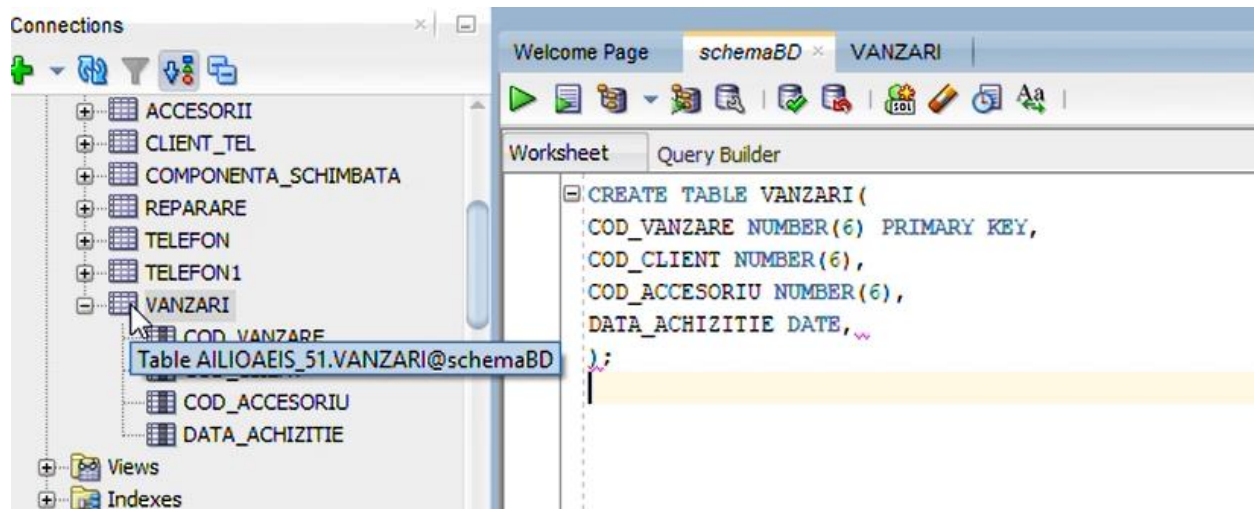
5. Tabela ACCESORII

```
CREATE TABLE ACCESORII(
COD_ACCESORIU NUMBER(6),
DENUMIRE VARCHAR2(20),
PRET NUMBER(4, 2)
);
```



6. Tabela VANZARI

```
CREATE TABLE VANZARI(
COD_VANZARE NUMBER(6) PRIMARY KEY,
COD_CLIENT NUMBER(6),
COD_ACCESORIU NUMBER(6),
DATA_ACHIZITIE DATE
);
```



Exemple cu operațiile LDD:

Adaugare cheie primara pentru tabela ACCESORII

- ALTER TABLE ACCESORII ADD CONSTRAINT accesorii_pk PRIMARY KEY(COD_ACCESORIU);

Adăugați coloana **STOC** în tabela **ACCESORII** având tipul Number(3)

- ALTER TABLE ACCESORII ADD(STOC NUMBER(3));

Cheie externa a tabeli **VANZARI** cu tabela **CLIENT_TEL**

- ALTER TABLE VANZARI ADD CONSTRAINT vanzari1_fk FOREIGN KEY(COD_CLIENT) REFERENCES CLIENT_TEL(COD_CLIENT);

Cheie externa a tabeli **VANZARI** cu tabela **ACCESORII**.

- ALTER TABLE VANZARI ADD CONSTRAINT VANZARI2_fk FOREIGN KEY(COD_ACCESORIU) REFERENCES ACCESORII(COD_ACCESORIU);

Stergeti tabela COMPONENTA_SCHIMBATA

- DROP TABLE COMPONENTA_SCHIMBATA; (dupa care o creem din nou)

Adăugați restricția de integritate **reparare_ck** care să nu permită introducerea în câmpul **COST** a unor valori mai mici de 0.

- ALTER TABLE REPARARE ADD CONSTRAINT reparare_ck CHECK (COST >0);

Adăugați restricția de integritate **reparare2_ck** care să nu permită introducerea în câmpul **DURATA** a unor valori mai mici de 0 și mai mari decât 20.

- ALTER TABLE REPARARE ADD CONSTRAINT reparare2_ck CHECK(DURATA BETWEEN 0 AND 20);

Adăugați coloana **COD_COMP** în tabela **REPARARE** având tipul Number(6).

- ALTER TABLE REPARARE ADD(COD_COMP NUMBER(6));

Cheie externa a tabelii **REPARARE** cu tabela **COMPONENTA_SCHIMBATA**.

- ALTER TABLE REPARARE ADD CONSTRAINT REPARARE2_fk FOREIGN KEY(COD_COMP) REFERENCES COMPONENTA_SCHIMBATA(COD_COMP);

Modificati precizia în 5 coloanei PRET în tabela COMPONENTA_SCHIMBATA.

- ALTER TABLE COMPONENTA_SCHIMBATA MODIFY (PRET NUMBER(5,2));

Modificati valoarea maxima de caractere pe care o poate avea coloana DENUMIRE în tabela COMPONENTA_SCHIMBATA

- ALTER TABLE COMPONENTA_SCHIMBATA MODIFY(DENUMIRE VARCHAR2(40));

4.Exemple cu operații de actualizare a datelor.

Adăugarea de înregistrări în fiecare tabelă: INSERT:

➤ Tabela TELEFON

```
INSERT INTO TELEFON ( COD_TELEFON, BRAND, MODEL, DEFECTIUNE) VALUES(
100, 'iPhone', '14', 'mufa de incarcare defecta');
```

```
INSERT INTO TELEFON ( COD_TELEFON, BRAND, MODEL, DEFECTIUNE) VALUES(
101, 'Samsung', 'S9', 'ecran spart');
```

```
INSERT INTO TELEFON ( COD_TELEFON, BRAND, MODEL, DEFECTIUNE) VALUES(
102, 'iPhone', 'X', 'difuzor infundat');
```

```
INSERT INTO TELEFON ( COD_TELEFON, BRAND, MODEL, DEFECTIUNE) VALUES(
103, 'Huawei', 'P8', 'ecran spart');
```

```
INSERT INTO TELEFON ( COD_TELEFON, BRAND, MODEL, DEFECTIUNE) VALUES(
104, 'iPhone', '7', 'mufa de incarcare defecta');
```

```
INSERT INTO TELEFON ( COD_TELEFON, BRAND, MODEL, DEFECTIUNE) VALUES(
105, 'Samsung', 'Hold', 'buton deschidere defect');
```

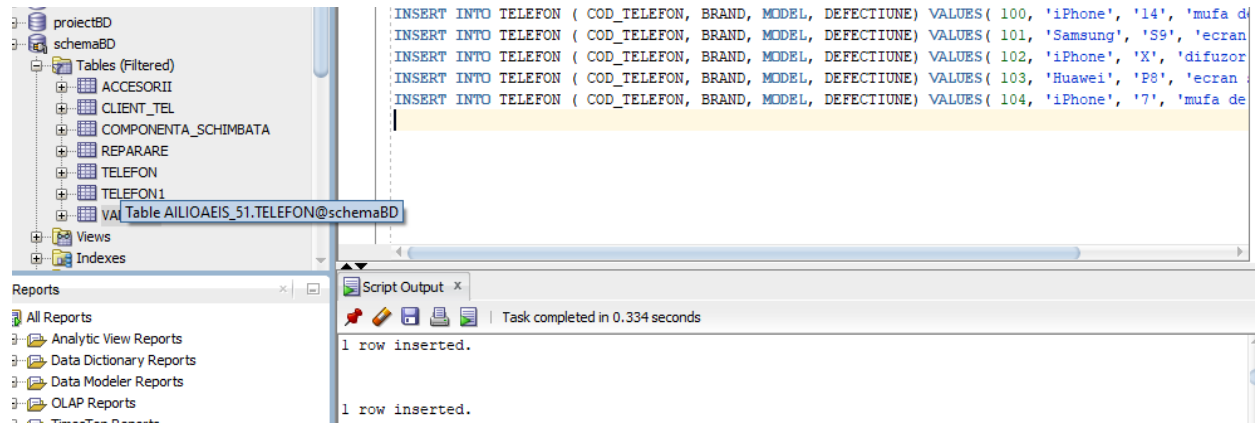
```
INSERT INTO TELEFON ( COD_TELEFON, BRAND, MODEL, DEFECTIUNE) VALUES(
106, 'iPhone', '12', 'camera frontala defecta');
```

```
INSERT INTO TELEFON ( COD_TELEFON, BRAND, MODEL, DEFECTIUNE) VALUES(
107, 'Motorola', 'Razr40', 'mufa de incarcare defecta');
```

```
INSERT INTO TELEFON ( COD_TELEFON, BRAND, MODEL, DEFECTIUNE) VALUES(
108, 'Huawei', 'P11', 'ecran spart');
```

```
INSERT INTO TELEFON ( COD_TELEFON, BRAND, MODEL, DEFECTIUNE) VALUES(
109, 'iPhone', '11', 'buton meniu defect');
```


INSERT INTO TELEFON (COD_TELEFON, BRAND, MODEL, DEFECTIONE) VALUES(110, 'Samsung', 'Flip 4', 'ecran spart');



➤ Tabela CLIENT_TEL

INSERT INTO CLIENT_TEL (COD_CLIENT, COD_TELEFON, NUME, PRENUME, NR_CONTACT) VALUES(1,100, 'Ailioaei', 'Sorina', 0757658685);

INSERT INTO CLIENT_TEL (COD_CLIENT, COD_TELEFON, NUME, PRENUME, NR_CONTACT) VALUES(5,101, 'Dimitra', 'Miruna', 0754231876);

INSERT INTO CLIENT_TEL (COD_CLIENT, COD_TELEFON, NUME, PRENUME, NR_CONTACT) VALUES(3,102, 'Ionescu', 'Rares', 0756437654);

INSERT INTO CLIENT_TEL (COD_CLIENT, COD_TELEFON, NUME, PRENUME, NR_CONTACT) VALUES(2,103, 'Popescu', 'Alin', 0756776543);

INSERT INTO CLIENT_TEL (COD_CLIENT, COD_TELEFON, NUME, PRENUME, NR_CONTACT) VALUES(4,105, 'Toader', 'Vlad', 0756123456);

INSERT INTO CLIENT_TEL (COD_CLIENT, COD_TELEFON, NUME, PRENUME, NR_CONTACT) VALUES(6,106, 'Aanei', 'Andra', 0756548850);

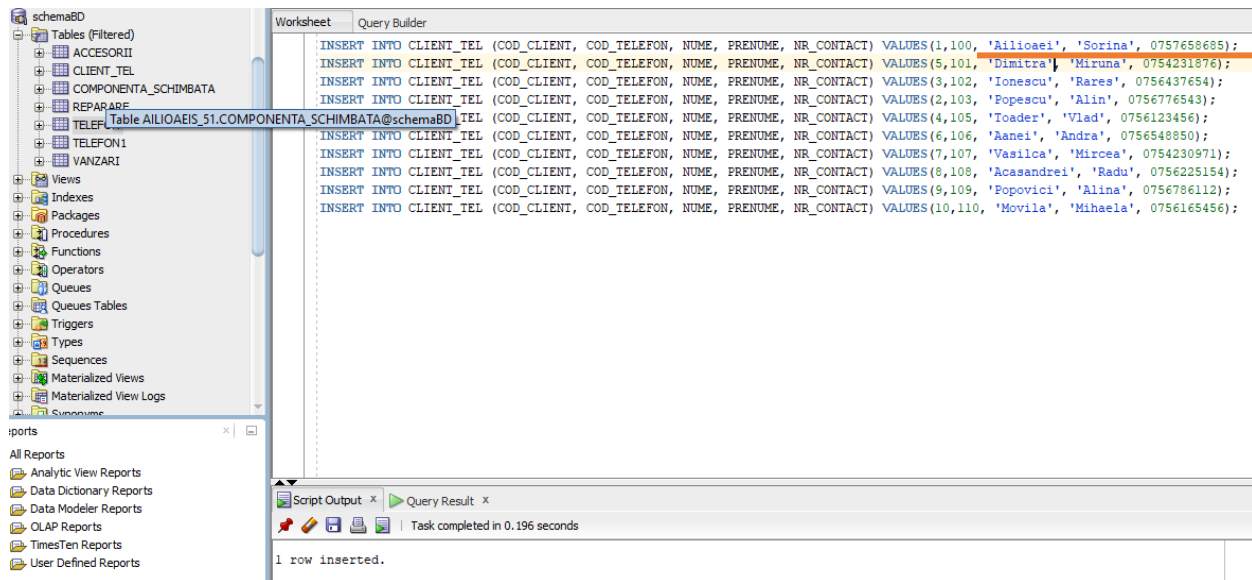
INSERT INTO CLIENT_TEL (COD_CLIENT, COD_TELEFON, NUME, PRENUME, NR_CONTACT) VALUES(7,107, 'Vasilca', 'Mircea', 0754230971);

INSERT INTO CLIENT_TEL (COD_CLIENT, COD_TELEFON, NUME, PRENUME, NR_CONTACT) VALUES(8,108, 'Acasandrei', 'Radu', 0756225154);

INSERT INTO CLIENT_TEL (COD_CLIENT, COD_TELEFON, NUME, PRENUME, NR_CONTACT) VALUES(9,109, 'Popovici', 'Alina', 0756786112);

INSERT INTO CLIENT_TEL (COD_CLIENT, COD_TELEFON, NUME, PRENUME, NR_CONTACT) VALUES(10,110, 'Movila', 'Mihaela', 0756165456);

INSERT INTO CLIENT_TEL (COD_CLIENT, COD_TELEFON, NUME, PRENUME, NR_CONTACT) VALUES(11,104, 'Butnariu', 'Alexandra', 0756440226);



➤ Tabela COMPONENTA_STRICATA

INSERT INTO COMPONENTA_SCHIMBATA (COD_COMP, DENUMIRE, PRET)
VALUES (001, 'ECRAN IPHONE', 420);

INSERT INTO COMPONENTA_SCHIMBATA (COD_COMP, DENUMIRE, PRET)
VALUES (002, 'ECRAN SAMSUNG', 400);

INSERT INTO COMPONENTA_SCHIMBATA (COD_COMP, DENUMIRE, PRET)
VALUES (003, 'ECRAN HUAWEI', 380);

INSERT INTO COMPONENTA_SCHIMBATA (COD_COMP, DENUMIRE, PRET)
VALUES (004, 'MUFA INCARCARE TYPE C', 150);

INSERT INTO COMPONENTA_SCHIMBATA (COD_COMP, DENUMIRE, PRET)
VALUES (005, 'MUFA INCARCARE LIGHTNING', 170);

INSERT INTO COMPONENTA_SCHIMBATA (COD_COMP, DENUMIRE, PRET)
VALUES (006, 'DIFUZOR IPHONE', 250);

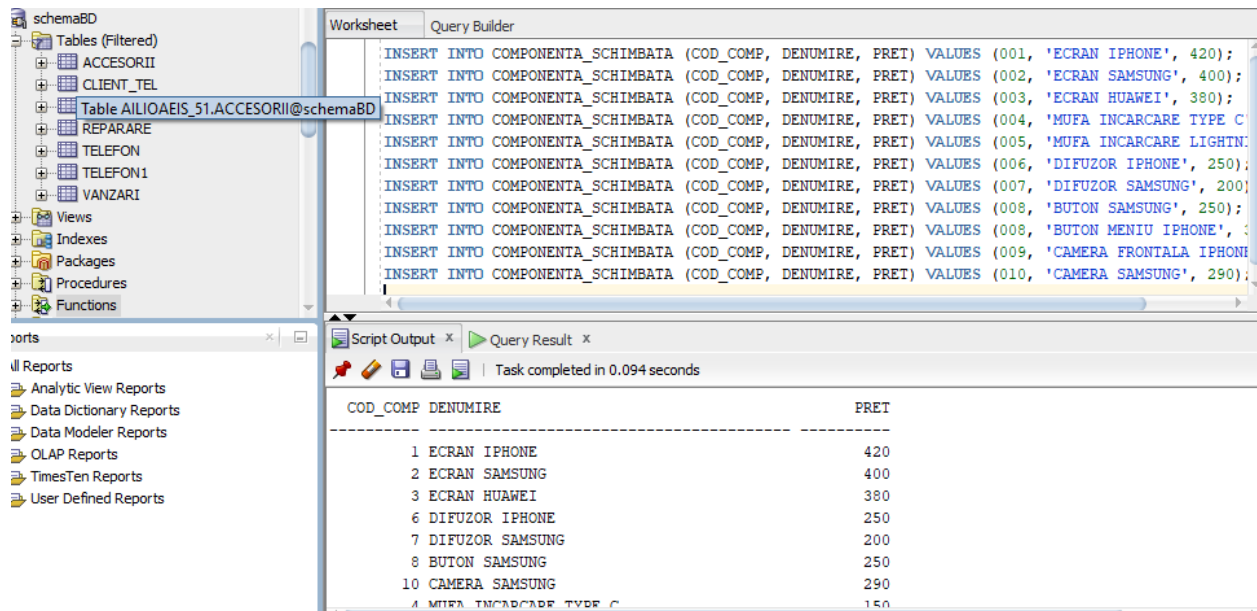
INSERT INTO COMPONENTA_SCHIMBATA (COD_COMP, DENUMIRE, PRET)
VALUES (007, 'DIFUZOR SAMSUNG', 200);

INSERT INTO COMPONENTA_SCHIMBATA (COD_COMP, DENUMIRE, PRET)
VALUES (008, 'BUTON SAMSUNG', 250);

INSERT INTO COMPONENTA_SCHIMBATA (COD_COMP, DENUMIRE, PRET)
VALUES (008, 'BUTON MENIU IPHONE', 300);

INSERT INTO COMPONENTA_SCHIMBATA (COD_COMP, DENUMIRE, PRET)
VALUES (009, 'CAMERA FRONTALA IPHONE', 350);

INSERT INTO COMPONENTA_SCHIMBATA (COD_COMP, DENUMIRE, PRET)
VALUES (010, 'CAMERA SAMSUNG', 290);



➤ Tabela ACCESORII

INSERT INTO ACCESORII (COD_ACCESORIU, DENUMIRE, PRET, STOC) VALUES(1, 'HUSA IPHONE', 50, 80);

INSERT INTO ACCESORII (COD_ACCESORIU, DENUMIRE, PRET, STOC) VALUES(2, 'HUSA SAMSUNG', 40, 100);

INSERT INTO ACCESORII (COD_ACCESORIU, DENUMIRE, PRET, STOC) VALUES(3, 'HUSA HUAWEI', 35, 90);

INSERT INTO ACCESORII (COD_ACCESORIU, DENUMIRE, PRET, STOC) VALUES(4, 'FOLIE PLASTIC', 10, 400);

INSERT INTO ACCESORII (COD_ACCESORIU, DENUMIRE, PRET, STOC) VALUES(5, 'FOLIE STICLA IPH', 50, 100);

INSERT INTO ACCESORII (COD_ACCESORIU, DENUMIRE, PRET, STOC) VALUES(6, 'FOLIE STICLA SMG', 40, 100);

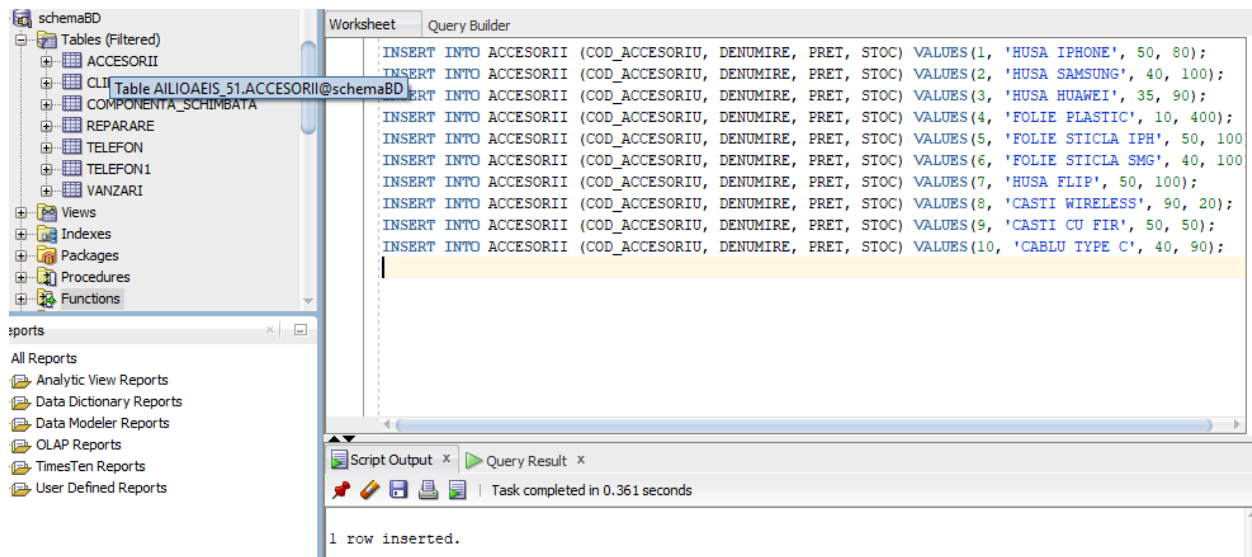
INSERT INTO ACCESORII (COD_ACCESORIU, DENUMIRE, PRET, STOC) VALUES(7, 'HUSA FLIP', 50, 100);

INSERT INTO ACCESORII (COD_ACCESORIU, DENUMIRE, PRET, STOC) VALUES(8, 'CASTI WIRELESS', 90, 20);

INSERT INTO ACCESORII (COD_ACCESORIU, DENUMIRE, PRET, STOC) VALUES(9, 'CASTI CU FIR', 50, 50);

INSERT INTO ACCESORII (COD_ACCESORIU, DENUMIRE, PRET, STOC) VALUES(10, 'CABLU TYPE C', 40, 90);

INSERT INTO ACCESORII (COD_ACCESORIU, DENUMIRE, PRET, STOC) VALUES(11, 'CABLU TYPE A', 30, 10);



➤ Tabela VÂNZĂRI

```
INSERT INTO VANZARI(COD_VANZARE, COD_CLIENT, COD_ACCESORIU,
DATA_ACHIZITIE) VALUES (111, 2, 2, TO_DATE('02.01.2024', 'DD.MM.YYYY'));
```

```
INSERT INTO VANZARI(COD_VANZARE, COD_CLIENT, COD_ACCESORIU,
DATA_ACHIZITIE) VALUES (222, 1, 5, TO_DATE('30.12.2023', 'DD.MM.YYYY'));
```

```
INSERT INTO VANZARI(COD_VANZARE, COD_CLIENT, COD_ACCESORIU,
DATA_ACHIZITIE) VALUES (333, 3, 2, TO_DATE('29.12.2023', 'DD.MM.YYYY'));
```

```
INSERT INTO VANZARI(COD_VANZARE, COD_CLIENT, COD_ACCESORIU,
DATA_ACHIZITIE) VALUES (444, 5, 2, TO_DATE('29.12.2023', 'DD.MM.YYYY'));
```

```
INSERT INTO VANZARI(COD_VANZARE, COD_CLIENT, COD_ACCESORIU,
DATA_ACHIZITIE) VALUES (555, 4, 6, TO_DATE('02.01.2024', 'DD.MM.YYYY'));
```

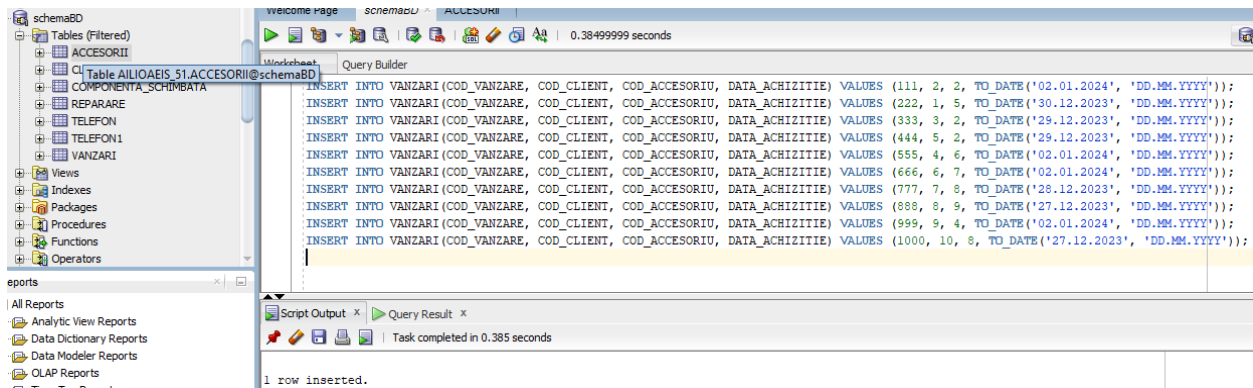
```
INSERT INTO VANZARI(COD_VANZARE, COD_CLIENT, COD_ACCESORIU,
DATA_ACHIZITIE) VALUES (666, 6, 7, TO_DATE('02.01.2024', 'DD.MM.YYYY'));
```

```
INSERT INTO VANZARI(COD_VANZARE, COD_CLIENT, COD_ACCESORIU,
DATA_ACHIZITIE) VALUES (777, 7, 8, TO_DATE('28.12.2023', 'DD.MM.YYYY'));
```

```
INSERT INTO VANZARI(COD_VANZARE, COD_CLIENT, COD_ACCESORIU,
DATA_ACHIZITIE) VALUES (888, 8, 9, TO_DATE('27.12.2023', 'DD.MM.YYYY'));
```

```
INSERT INTO VANZARI(COD_VANZARE, COD_CLIENT, COD_ACCESORIU,
DATA_ACHIZITIE) VALUES (999, 9, 4, TO_DATE('02.01.2024', 'DD.MM.YYYY'));
```

```
INSERT INTO VANZARI(COD_VANZARE, COD_CLIENT, COD_ACCESORIU,
DATA_ACHIZITIE) VALUES (1000, 10, 8, TO_DATE('27.12.2023', 'DD.MM.YYYY'));
```



➤ Tabela REPARARE

INSERT INTO REPARARE (COD_REPARARE, COD_TELEFON, DATA_INTRARE, COST, DURATA, COD_COMP) VALUES(11,100,TO_DATE('02.01.2024', 'DD.MM.YYYY'), 70, 2, 005);

INSERT INTO REPARARE (COD_REPARARE, COD_TELEFON, DATA_INTRARE, COST, DURATA, COD_COMP) VALUES(22,101,TO_DATE('02.01.2024', 'DD.MM.YYYY'), 80, 3, 002);

INSERT INTO REPARARE (COD_REPARARE, COD_TELEFON, DATA_INTRARE, COST, DURATA, COD_COMP) VALUES(33,102,TO_DATE('31.12.2023', 'DD.MM.YYYY'), 50, 1, 006);

INSERT INTO REPARARE (COD_REPARARE, COD_TELEFON, DATA_INTRARE, COST, DURATA, COD_COMP) VALUES(44,103,TO_DATE('29.12.2023', 'DD.MM.YYYY'), 80, 3, 003);

INSERT INTO REPARARE (COD_REPARARE, COD_TELEFON, DATA_INTRARE, COST, DURATA, COD_COMP) VALUES(55,104,TO_DATE('03.01.2024', 'DD.MM.YYYY'), 70, 2, 005);

INSERT INTO REPARARE (COD_REPARARE, COD_TELEFON, DATA_INTRARE, COST, DURATA, COD_COMP) VALUES(66,105,TO_DATE('22.12.2023', 'DD.MM.YYYY'), 50, 2, 008);

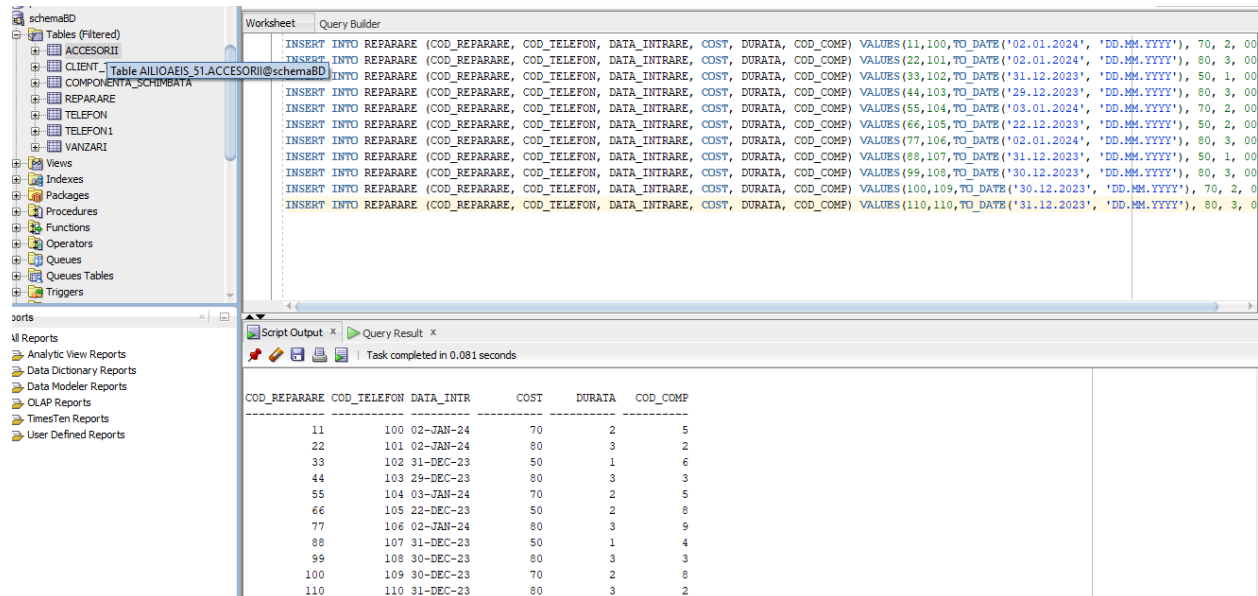
INSERT INTO REPARARE (COD_REPARARE, COD_TELEFON, DATA_INTRARE, COST, DURATA, COD_COMP) VALUES(77,106,TO_DATE('02.01.2024', 'DD.MM.YYYY'), 80, 3, 009);

INSERT INTO REPARARE (COD_REPARARE, COD_TELEFON, DATA_INTRARE, COST, DURATA, COD_COMP) VALUES(88,107,TO_DATE('31.12.2023', 'DD.MM.YYYY'), 50, 1, 004);

INSERT INTO REPARARE (COD_REPARARE, COD_TELEFON, DATA_INTRARE, COST, DURATA, COD_COMP) VALUES(99,108,TO_DATE('30.12.2023', 'DD.MM.YYYY'), 80, 3, 003);

```
INSERT INTO REPARARE (COD_REPARARE, COD_TELEFON, DATA_INTRARE,
COST, DURATA, COD_COMP) VALUES(100,109,TO_DATE('30.12.2023',
'DD.MM.YYYY'), 70, 2, 008);
```

```
INSERT INTO REPARARE (COD_REPARARE, COD_TELEFON, DATA_INTRARE,
COST, DURATA, COD_COMP) VALUES(110,110,TO_DATE('31.12.2023',
'DD.MM.YYYY'), 80, 3, 002);
```



COD_REPARARE	COD_TELEFON	DATA_INTR	COST	DURATA	COD_COMP
11	100	02-JAN-24	70	2	5
22	101	02-JAN-24	80	3	2
33	102	31-DEC-23	50	1	6
44	103	29-DEC-23	80	3	3
55	104	03-JAN-24	70	2	5
66	105	22-DEC-23	50	2	8
77	106	02-JAN-24	80	3	9
88	107	31-DEC-23	50	1	4
99	108	30-DEC-23	80	3	3
100	109	30-DEC-23	70	2	8
110	110	31-DEC-23	80	3	2

Actualizarea înregistrărilor:

UPDATE:

- Să se actualizeze stocul=50 pentru accesoriul cu codul 2.

```
UPDATE ACCESORII SET STOC=50 WHERE COD_ACCESORIU=2;
```

- Sa se actualizeze MODEL=Razr20 pentru brandul Motorola.

```
UPDATE TELEFON SET MODEL='Razr20' WHERE BRAND='Motorola';
```

- Sa se modifice pretul pentru schimbul de ecran al brandului Iphone, la 450.
UPDATE COMPONENTA_SCHIMBATA SET PRET=450 WHERE
DENUMIRE='ECRAN IPHONE';

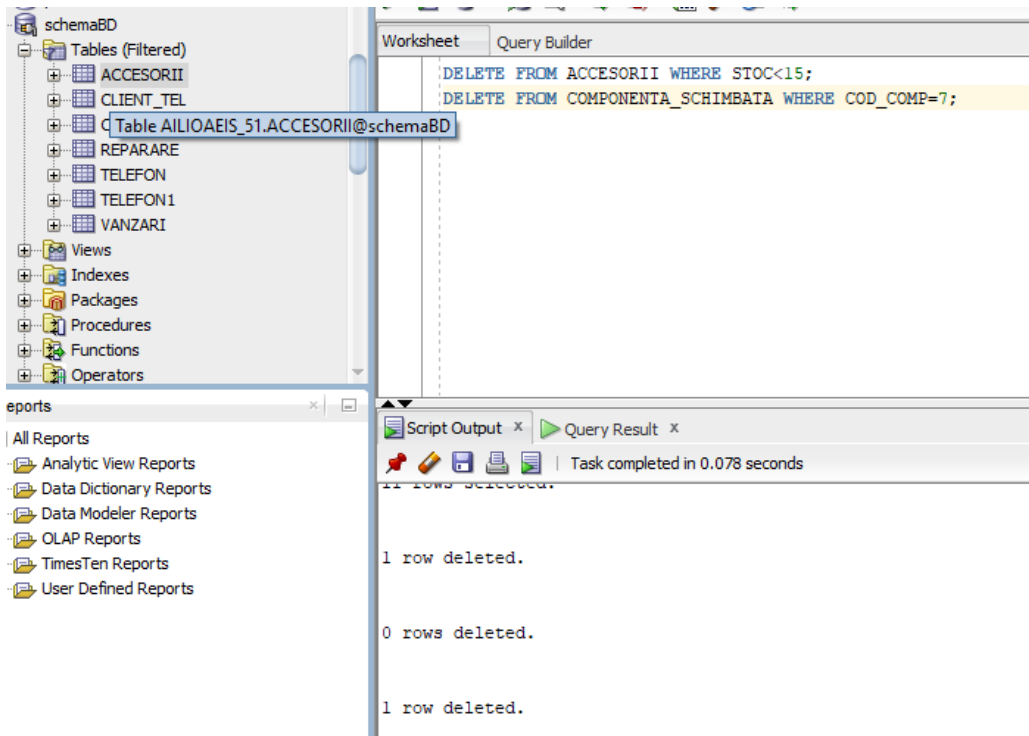
DELETE:

- Sa se stearga accesoriul cu stocul mai mic de 15.

```
DELETE FROM ACCESORII WHERE STOC<15;
```

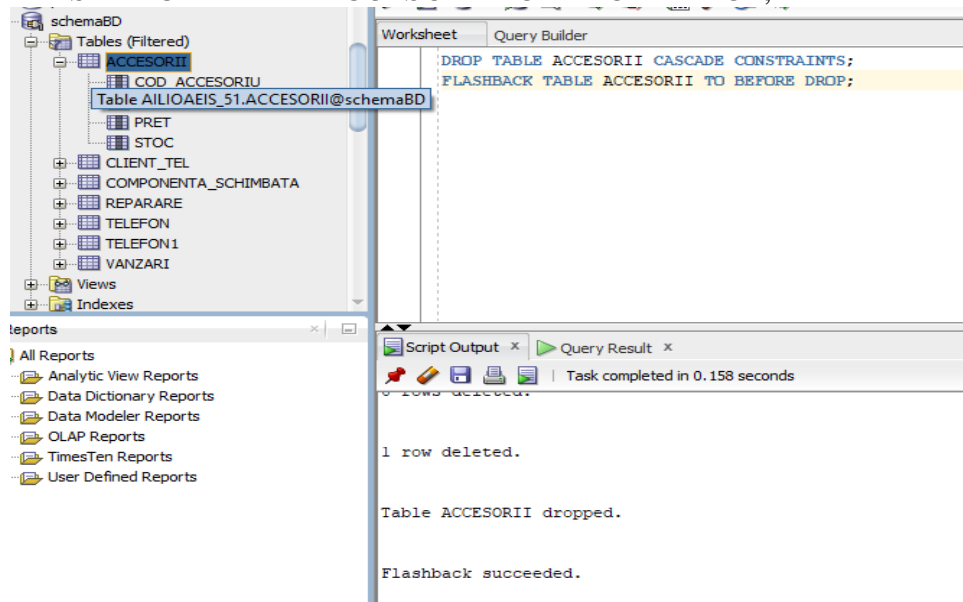
- Sa se stearga componenta de schimbat cu codul 7.

```
DELETE FROM COMPONENTA_SCHIMBATA WHERE COD_COMP=7;
```



Ștergerea și recuperarea unei tabele:

-Ștergerea tablei ACCESORII si recuperarea acesteia cu flashback.
DROP TABLE ACCESORII CASCADE CONSTRAINTS;
FLASHBACK TABLE ACCESORII TO BEFORE DROP;



4.Exemple de interogări.

- Sa se afiseze denumirea si pretul componentelor care au pretul cuprins intre 100 si 200.

SELECT DENUMIRE, PRET FROM COMPONENTA_SCHIMBATA WHERE PRET BETWEEN 100 AND 200;

The screenshot shows the SQL Developer interface. On the left, the 'schemaBD' tree is expanded, showing tables like ACCESORII, CLIENT_TEL, COM, REPARARE, TELEFON, TELEFON1, and VANZARI. The 'Table AILIOAEIS_51.ACCESORII@schemaBD' is selected. The 'Query Builder' tab is active, showing the query: `SELECT DENUMIRE, PRET FROM COMPONENTA_SCHIMBATA WHERE PRET BETWEEN 100 AND 200;`. The 'Script Output' tab shows the execution results:

DENUMIRE	PRET
MUFA INCARCARE TYPE C	150
MUFA INCARCARE LIGHTNING	170

- Afisati accesoriile ce au denumirea incepand cu litera 'c'

SELECT * FROM ACCESORII WHERE DENUMIRE LIKE 'C%';

The screenshot shows the SQL Developer interface. On the left, the 'schemaBD' tree is expanded, showing tables like ACCESORII, CLIENT_TEL, COM, REPARARE, TELEFON, TELEFON1, and VANZARI. The 'Table AILIOAEIS_51.ACCESORII@schemaBD' is selected. The 'Query Builder' tab is active, showing the query: `SELECT * FROM ACCESORII WHERE DENUMIRE LIKE 'C%';`. The 'Script Output' tab shows the execution results:

no rows selected

COD_ACCESORIU	DENUMIRE	PRET	STOC
8	CASTI WIRELESS	90	20
9	CASTI CU FIR	50	50
10	CABLU TYPE C	40	90

- Afisati denumirea si pretul accesoriilor cu pretul<40.

SELECT DENUMIRE, PRET FROM ACCESORII WHERE PRET<40;

The screenshot shows a database management tool interface. On the left, a tree view displays the database schema, including tables like ACCESORII, CLIENT_TEL, and VANZARI. The main area shows a query in the Query Builder: `SELECT DENUMIRE, PRET FROM ACCESORII WHERE PRET < 40;`. Below the query, the Script Output window displays the results of the query, showing two rows of data:

DENUMIRE	PRET
HUSA HUAWEI	35
FOLIE PLASTIC	10

- Sa se afiseze PRENUMELE clientilor care nu au nr de telefon.

`SELECT PRENUME FROM CLIENT_TEL WHERE NR_CONTACT IS NULL;`

The screenshot shows the same database management tool interface. The query in the Query Builder is now: `SELECT PRENUME FROM CLIENT_TEL WHERE NR_CONTACT IS NULL;`. The Script Output window displays the results of the query, showing "no rows selected".

- Sa se afiseze codul produselor vandute in anii 2022 si 2023 si ordonati crescator dupa data achizitiei;

`SELECT COD_VANZARE, DATA_ACHIZITIE FROM VANZARI
WHERE EXTRACT(YEAR FROM DATA_ACHIZITIE) IN (2022, 2023)
ORDER BY DATA_ACHIZITIE;`

The screenshot shows a database management tool interface. On the left, a tree view displays the database schema 'schemaBD' with various tables and views. The 'Tables (Filtered)' section is expanded, showing tables like 'ACCESORII', 'CLIENT_TEL', 'COMPONENTA_SCHIMBATA', 'REPARARE', 'TELEFON', 'TELEFON1', and 'VANZARI'. The 'Query Builder' window on the right contains the following SQL query:

```
SELECT COD_VANZARE, DATA_ACHIZITIE FROM VANZARI
WHERE EXTRACT(YEAR FROM DATA_ACHIZITIE) IN (2022, 2023)
ORDER BY DATA_ACHIZITIE;
```

The 'Script Output' window at the bottom right displays the results of the query, showing a list of purchase codes and dates:

COD_VANZARE	DATA_ACHIZITIE
888	27-DEC-23
1000	27-DEC-23
777	28-DEC-23
333	29-DEC-23
444	29-DEC-23
222	30-DEC-23

- Sa se afiseze numele si prenumele clientului cu telefon de brand Samsung.

```
SELECT NUME, PRENUME FROM CLIENT_TEL C, TELEFON T
WHERE C.COD_TELEFON=T.COD_TELEFON AND BRAND='Samsung';
```

The screenshot shows the same database management tool interface. The 'Query Builder' window now contains the following SQL query:

```
SELECT NUME, PRENUME
FROM CLIENT_TEL C, TELEFON T
WHERE C.COD_TELEFON=T.COD_TELEFON AND BRAND='Samsung';
```

The 'Script Output' window at the bottom right displays the results of the query, showing the names of clients with Samsung phones:

NUME	PRENUME
Dimitra	Miruna
Toader	Vlad
Movila	Mihaela

- Sa se afiseze costul de reparare al telefoanelor de brand iPhone, in ordine descrescatoare

```
SELECT R.COST FROM REPARARE R ,TELEFON T
WHERE R.COD_TELEFON=T.COD_TELEFON AND T.BRAND='iPhone'
ORDER BY R.COST DESC;
```

The screenshot shows a database management tool interface. On the left, a tree view displays the database schema, including tables like ACCESORII, CLIENT_TEL, REPARARE, TELEFON, and VANZARI. The main area shows a query in the Query Builder:

```
SELECT R.COST FROM REPARARE R, TELEFON T
WHERE R.COD_TELEFON=T.COD_TELEFON AND T.BRAND='iPhone'
ORDER BY R.COST DESC;
```

The Script Output window shows the results of the query:

COST
80
70
70
70
50

- Sa se afiseze media costurilor componentelor pentru fiecare tip de brand.

```
SELECT T.BRAND, ROUND(AVG(R.COST),2) AS MEDIE FROM TELEFON T, REPARARE R
WHERE T.COD_TELEFON=R.COD_TELEFON
GROUP BY BRAND;
```

The screenshot shows the same database management tool interface. The query in the Query Builder is:

```
SELECT T.BRAND, ROUND(AVG(R.COST),2) AS MEDIE
FROM TELEFON T, REPARARE R
WHERE T.COD_TELEFON=R.COD_TELEFON
GROUP BY BRAND;
```

The Script Output window shows the results of the query:

BRAND	MEDIE
iPhone	68
Samsung	70
Huawei	80
Motorola	50

- Sa se afiseze media mai mica de 70 a costurilor de reparare pentru fiecare tip de brand.

```
SELECT T.BRAND, ROUND(AVG(R.COST),2) AS MEDIE FROM TELEFON T, REPARARE R
WHERE T.COD_TELEFON=R.COD_TELEFON
GROUP BY BRAND
HAVING ROUND(AVG(R.COST),2)<70;
```

The screenshot shows a database query tool interface. On the left, a tree view displays the database schema 'schemaBD' with tables like ACCESORII, CLIENT_TEL, REPARARE, TELEFON, and VANZARI. The 'Table AILIOAEIS_51.CLIENT_TEL@schemaBD' is selected. The main window shows a SQL query in the 'Query Builder' tab:

```
SELECT T.BRAND, ROUND(AVG(R.COST), 2) AS MEDIE
FROM TELEFON T, REPARARE R
WHERE T.COD_TELEFON=R.COD_TELEFON
GROUP BY BRAND
HAVING ROUND(AVG(R.COST), 2) < 70;
```

The 'Script Output' tab shows the execution result:

BRAND	MEDIE
iPhone	68
Motorola	50

- Sa se afiseze BRAND-ul si MODELUL telefoanelor reparate pe 2 ianuarie 2024 folosind TO_DATE.

```
SELECT T.BRAND, T.MODEL, R.DATA_INTRARE
FROM TELEFON T, REPARARE R
WHERE T.COD_TELEFON=R.COD_TELEFON AND R.DATA_INTRARE = TO_DATE('02.01.2024',
'DD.MM.YYYY');
```

The screenshot shows the same database query tool interface. The 'Table AILIOAEIS_51.ACCESORII@schemaBD' is selected in the tree view. The SQL query in the 'Query Builder' tab is:

```
SELECT T.BRAND, T.MODEL, R.DATA_INTRARE
FROM TELEFON T, REPARARE R
WHERE T.COD_TELEFON=R.COD_TELEFON AND R.DATA_INTRARE = TO_DATE('02.01.2024', 'DD.MM.YYYY');
```

The 'Script Output' tab shows the execution result:

BRAND	MODEL	DATA_INTR
iPhone	14	02-JAN-24
Samsung	S23	02-JAN-24
iPhone	12	02-JAN-24

- Afisati telefoanele reparate in anul 2024 folosind functia EXTRACT.

```
SELECT T.BRAND, T.MODEL, R.DATA_INTRARE FROM TELEFON T, REPARARE R
WHERE R.COD_TELEFON=T.COD_TELEFON AND EXTRACT(YEAR FROM
R.DATA_INTRARE)=2024;
```

The screenshot shows a database management tool interface. On the left, a tree view displays the database schema, including tables like ACCESORII, CLIENT_TEL, REPARARE, TELEFON, and VANZARI. The 'Table AILIOAEIS_51.ACCESORII@schemaBD' is selected. The main window shows a SQL query in the 'Query Builder' tab:

```
SELECT T.BRAND, T.MODEL, R.DATA_INTRARE
FROM TELEFON T, REPARARE R
WHERE R.COD_TELEFON=T.COD_TELEFON
AND EXTRACT(YEAR FROM R.DATA_INTRARE)=2024;
```

Below the query, the 'Script Output' window shows the results of the query, which are displayed in a table format:

BRAND	MODEL	DATA_INTR
iPhone	14	02-JAN-24
Samsung	S23	02-JAN-24
iPhone	13	03-JAN-24
iPhone	12	02-JAN-24

- Sa se afiseze telefoanele reparate si data intrarii acestora in service in format MM/YY folosind functia TO_CHAR.

```
SELECT COD_REPARARE, TO_CHAR(DATA_INTRARE, 'MM/YY') DATA_INTRARE
FROM REPARARE;
```

The screenshot shows the same database management tool interface. The 'Table AILIOAEIS_51.ACCESORII@schemaBD' is still selected. The main window shows a new SQL query in the 'Query Builder' tab:

```
SELECT COD_REPARARE, TO_CHAR(DATA_INTRARE, 'MM/YY') DATA_INTRARE
FROM REPARARE;
```

Below the query, the 'Script Output' window shows the results of the query, which are displayed in a table format:

COD_REPARARE	DATA_INTRARE
33	12/23
44	12/23
55	01/24
66	12/23
77	01/24
88	12/23
99	12/23
100	12/23
110	12/23

- Sa se afiseze numele, prenumele si ultimele 3 cifre al numarului de contact pentru fiecare angajat.

```
SELECT NUME,PRENUME, SUBSTR(NR_CONTACT, LENGTH(NR_CONTACT) - 2, 3) AS
Ultimele_Trei_Cifre
FROM CLIENT_TEL;
```

Query Builder

```
SELECT NUME, PRENUME, SUBSTR(NR_CONTACT, LENGTH(NR_CONTACT) - 2, 3) AS Ultimele_Trei_Cifre
FROM CLIENT_TEL;
```

Script Output x

Task completed in 0.091 seconds

Ailioaei	Sorina	685
Dimitra	Miruna	876
Ionescu	Rares	654
Popescu	Alin	543
Toader	Vlad	456
Aanei	Andra	850
Vasilca	Mircea	971
Acasandrei	Radu	154

- Sa se afiseze clientii cu numele Ailioaei utilizand functiile INITCAP, UPPER SI LOWER.

SELECT COD_CLIENT, nume FROM CLIENT_TEL WHERE INITCAP(num) like 'Ailioaei%';

SELECT COD_CLIENT, nume FROM CLIENT_TEL WHERE upper(num) like 'AILIOAEI%';

SELECT COD_CLIENT, nume FROM CLIENT_TEL WHERE lower(num) like 'ailioaei%';

Query Builder

```
SELECT COD_CLIENT, nume FROM CLIENT_TEL WHERE INITCAP(num) like 'Ailioaei%';
SELECT COD_CLIENT, nume FROM CLIENT_TEL WHERE upper(num) like 'AILIOAEI%';
SELECT COD_CLIENT, nume FROM CLIENT_TEL WHERE lower(num) like 'ailioaei%';
```

Script Output x

Task completed in 0.14 seconds

1 Ailioaei

- Sa se afiseze diferenta de zile intre data curenta si data achizitiei al fiecarui produs vandut in parte si ordonati crescator acestea dupa data achizitiei.

SELECT COD_VANZARE, DATA_ACHIZITIE, trunc(SYSDATE - DATA_ACHIZITIE) AS Diferenta_Zile

FROM VANZARI

ORDER BY DATA_ACHIZITIE;

The screenshot shows a database management interface. On the left, a tree view displays the database schema, including tables like ACCESORII, CLIENT_TEL, REPARARE, TELEFON, and VANZARI. The main area shows a SQL query in the Query Builder:

```
SELECT COD_VANZARE, DATA_ACHIZITIE, trunc(SYSDATE - DATA_ACHIZITIE) AS Diferenta_Zile
FROM VANZARI
ORDER BY DATA_ACHIZITIE;
```

Below the query, the Script Output window shows the results of the query, indicating that the task was completed in 0.122 seconds. The results are as follows:

COD_VANZARE	DATA_ACHIZITIE	Diferenta_Zile
888	27-DEC-23	7
1000	27-DEC-23	7
777	28-DEC-23	6
333	29-DEC-23	5
444	29-DEC-23	5
222	30-DEC-23	4
999	02-JAN-24	1
666	02-JAN-24	1

- Sa se afiseze mesaje urmatoare pentru vanzarile de accesorii:
 - Daca pretul este mai mare decat 50 se afiseaza mesajul “Pret ridicat”
 - Daca pretul este cuprins intre 30 si 50 se afiseaza mesajul “Pret moderat”
 - Iar daca pretul este mai mic decat 30 se afiseaza mesajul “Pret mic”.

```
SELECT V.COD_VANZARE, A.COD_ACCESORIU, A.PRET,
CASE
WHEN A.PRET>50 THEN 'PRET RIDICAT'
WHEN A.PRET BETWEEN 30 AND 50 THEN 'PRET MODERAT'
WHEN A.PRET<30 THEN 'PRET MIC'
END
FROM VANZARI V, ACCESORII A
WHERE V.COD_ACCESORIU=A.COD_ACCESORIU;
```


The screenshot shows a database query tool interface. On the left, a tree view displays the database schema 'schemaBD' with tables like ACCESORII, CLIENT_TEL, and VANZARI. The 'Table AILIOAEIS_51.CLIENT_TEL@schemaBD' is selected. The main window shows a SQL query in the 'Query Builder' tab:

```
SELECT V.COD_VANZARE, A.COD_ACCESORIU, A.PRET,
CASE
WHEN A.PRET>50 THEN 'PRET RIDICAT'
WHEN A.PRET BETWEEN 30 AND 50 THEN 'PRET MODERAT'
WHEN A.PRET<30 THEN 'PRET MIC'
END
FROM VANZARI V, ACCESORII A
WHERE V.COD_ACCESORIU=A.COD_ACCESORIU;
```

The 'Script Output' tab shows the results of the query, indicating 'Task completed in 0.074 seconds'. The output is a table with columns: COD_VANZARE, COD_ACCESORIU, PRET, and CASEWHENA.PR.

COD_VANZARE	COD_ACCESORIU	PRET	CASEWHENA.PR
111	2	40	PRET MODERAT
333	2	40	PRET MODERAT
444	2	40	PRET MODERAT
999	4	10	PRET MIC
222	5	50	PRET MODERAT
555	6	40	PRET MODERAT
666	7	50	PRET MODERAT
777	8	90	PRET RIDICAT
1000	8	90	PRET RIDICAT
888	9	50	PRET MODERAT

- Sa se afiseze id-ul clientilor ce nu a participat la nicio vanzare.

SELECT C.COD_CLIENT FROM CLIENT_TEL C MINUS SELECT V.COD_CLIENT FROM VANZARI V;

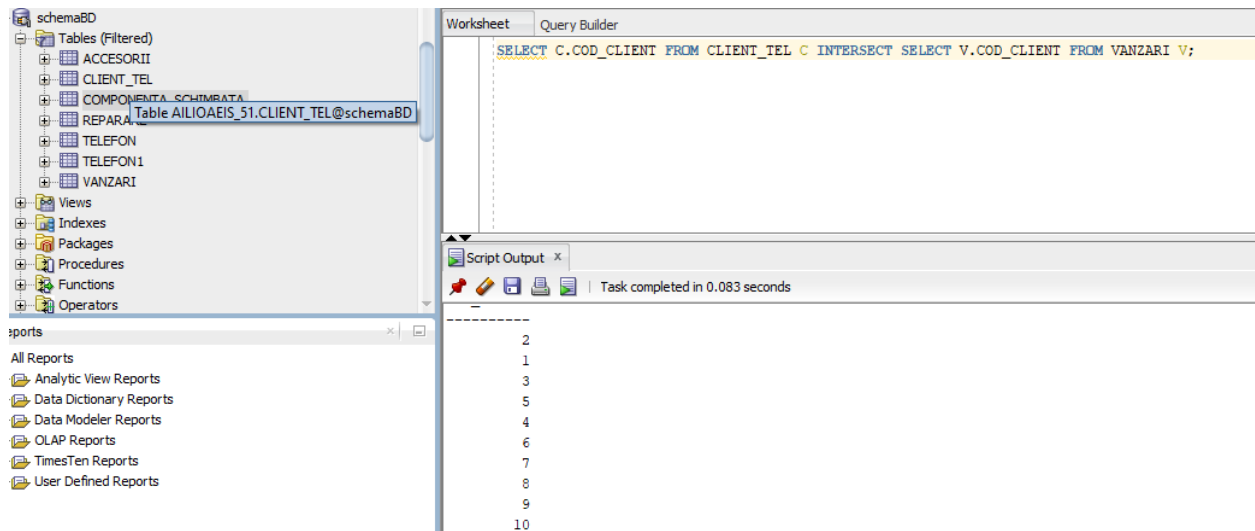
The screenshot shows the same database query tool interface. The 'Table AILIOAEIS_51.CLIENT_TEL@schemaBD' is selected. The main window shows a SQL query in the 'Query Builder' tab:

```
SELECT C.COD_CLIENT FROM CLIENT_TEL C MINUS SELECT V.COD_CLIENT FROM VANZARI V;
```

The 'Script Output' tab shows the results of the query, indicating 'Task completed in 0.078 seconds'. The output is a table with the column: COD_CLIENT.

COD_CLIENT
11

- Sa se afiseze id-ul clientilor ce au participat la vanzare.
SELECT C.COD_CLIENT FROM CLIENT_TEL C INTERSECT SELECT V.COD_CLIENT FROM VANZARI V;

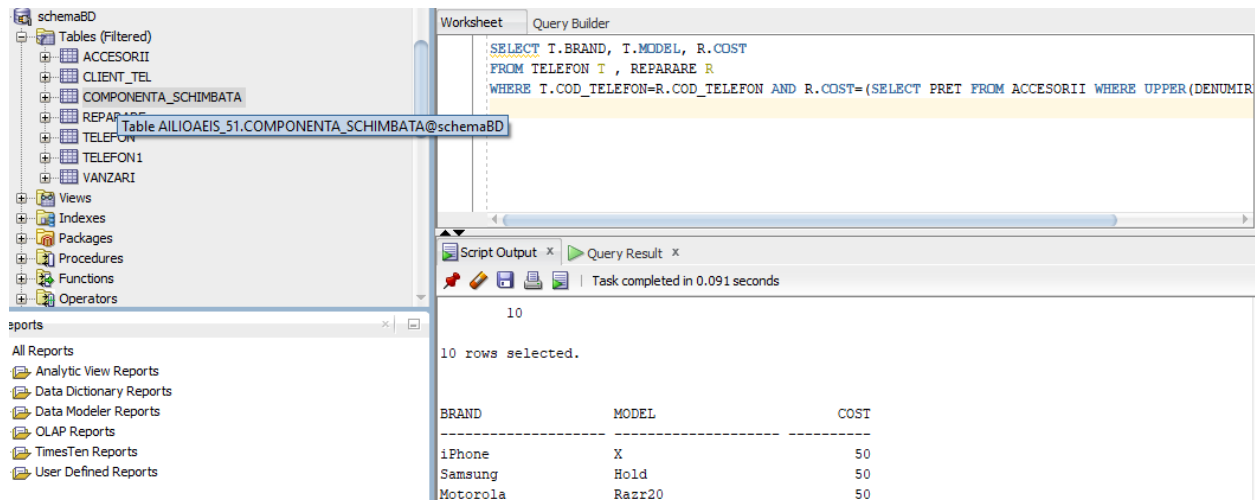


- Sa se afiseze telefoanele ce au costul de reparare egal cu pretul accesoriiului HUSA IPHONE.

```
SELECT T.BRAND, T.MODEL, R.COST
```

```
FROM TELEFON T , REPARARE R
```

```
WHERE T.COD_TELEFON=R.COD_TELEFON AND R.COST=(SELECT PRET FROM ACCESORII  
WHERE UPPER(DENUMIRE)='HUSA IPHONE');
```



- Sa se modifice durata de reparare a telefonului primului client cu durata de reparare a telefonului celui de-al doilea client.

```
UPDATE REPARARE
```

```
SET DURATA=(SELECT DURATA FROM REPARARE WHERE COD_REPARARE=22)
```

```
WHERE COD_REPARARE=11;
```

```

UPDATE REPARARE
SET DURATA=(SELECT DURATA FROM REPARARE WHERE COD_REPARARE=22)
WHERE COD_REPARARE=11;
select * from reparaare;

```

Script Output x Query Result x

Task completed in 0.112 seconds

1 row updated.

COD_REPARARE	COD_TELEFON	DATA_INTR	COST	DURATA	COD_COMP
11	100	02-JAN-24	70	3	5
22	101	02-JAN-24	80	3	2

- Sa se creeze o tabela virtuala cu toate reparatiile ce au costul mai mare decat 60 de lei. Afisati tabela virtuala.

```

CREATE VIEW reparatii_scumpe
AS SELECT * FROM REPARARE
WHERE COST > 60;
SELECT * FROM reparatii_scumpe;

```

```

CREATE VIEW reparatii_scumpe
AS SELECT * FROM REPARARE
WHERE COST > 60;
SELECT * FROM reparatii_scumpe;

```

Script Output x Query Result x

Task completed in 0.122 seconds

8 rows selected.

COD_REPARARE	COD_TELEFON	DATA_INTR	COST	DURATA	COD_COMP
11	100	02-JAN-24	70	3	5
22	101	02-JAN-24	80	3	2
44	103	29-DEC-23	80	3	3
55	104	03-JAN-24	70	2	5
77	106	02-JAN-24	80	3	9
99	108	30-DEC-23	80	3	3
100	109	30-DEC-23	70	2	8
110	110	31-DEC-23	80	3	2

- Sa se stearga din tabela virtuala creata mai sus, reparatiile care au costul egal cu 70. Afisati rezultatul.

```

delete from reparatii_scumpe
where cost=70;

```

SELECT * FROM reparatii_scumpe;

Worksheet Query Builder

```
delete from reparatii_scumpe
where cost=70;
SELECT * FROM reparatii_scumpe;
```

Script Output x Query Result x

Task completed in 0.102 seconds

3 rows deleted.

COD_REPARARE	COD_TELEFON	DATA_INTR	COST	DURATA	COD_COMP
22	101	02-JAN-24	80	3	2
44	103	29-DEC-23	80	3	3
77	106	02-JAN-24	80	3	9
99	108	30-DEC-23	80	3	3
110	110	31-DEC-23	80	3	2

- Sa se creeze si sa se stearga un index pe tabela CLIENT_TEL pe coloana NUME.

create index clienti_num_idx on client_tel(num);

drop index clienti_num_idx;

Worksheet Query Builder

```
create index clienti_num_idx on client_tel(num);
drop index clienti_num_idx;
```

Script Output x Query Result x

Task completed in 0.185 seconds

Index CLIENTI_NUM_IDX created.

Index CLIENTI_NUM_IDX dropped.

- Sa se creeze o secventa pentru asigurarea unicitatii cheii primare din tabela TELEFON.

CREATE SEQUENCE seq_cod_telefon START WITH 100 INCREMENT BY 10
MAXVALUE 150 NOCYCLE;

INSERT INTO TELEFON (COD_TELEFON, BRAND, MODEL, DEFECTIUNE)

```
VALUES (seq_cod_telefon.NEXTVAL, 'Samsung', 'Galaxy S21', 'ecran spart');  
select * from telefon;
```

- Afisati valoarea curenta a secventei:

```
SELECT seq_cod_telefon.CURRVAL FROM DUAL;
```

```
drop sequence seq_cod_telefon;
```

The screenshot displays a database management interface. On the left, a tree view shows the database schema, including tables like ACCESORII, CLIENT_TEL, COMPONENTA_SCHIMBATA, REPARARE, TELEFON, TELEFON1, and VANZARI. The 'Table AILIOAEIS_51.TELEFON1@schemaBD' is selected. The main window shows a 'Query Builder' with a script containing the following SQL commands:

```
--CREATE SEQUENCE seq_cod_telefon START WITH 110 INCREMENT BY 10 MAXVALUE 150 NOCYCLE;  
--INSERT INTO TELEFON (COD_TELEFON, BRAND, MODEL, DEFECTIUNE)  
--VALUES (seq_cod_telefon.NEXTVAL, 'Samsung', 'Galaxy S21', 'ecran spart');  
drop sequence seq_cod_telefon;  
--select * from telefon;  
--SELECT seq_cod_telefon.CURRVAL FROM DUAL;
```

Below the script, the 'Script Output' tab shows the execution results:

COD_TELEFON	BRAND	MODEL	DEFECTIUNE
110	Samsung	Galaxy S21	ecran spart
120	Samsung	Galaxy S21	ecran spart

12 rows selected.
Sequence SEQ_COD_TELEFON dropped.

- Sa se creeze un sinonim pentru tabela REPARARE.

```
CREATE SYNONYM detalii_reparare FOR REPARARE;
```

```
select * from detalii_reparare;
```

```
DROP SYNONYM detalii_reparare;
```

schemaBD

Tables (Filtered)

ACCESORII

CLIENT_TEL

COMPONENTA SCHIMBATA

REPARARE

TELEFON

TELEFON1

VANZARI

Views

Indexes

Packages

Procedures

Functions

Operators

reports

All Reports

Analytic View Reports

Data Dictionary Reports

Data Modeler Reports

OLAP Reports

TimesTen Reports

User Defined Reports

Worksheet

Query Builder

CREATE SYNONYM detalii_reparare FOR REPARARE;

select * from detalii_reparare;

DROP SYNONYM detalii_reparare;

Script Output x

Query Result x

Task completed in 0.164 seconds

66	105	22-DEC-23	80	2	0
77	106	02-JAN-24	80	3	9
88	107	31-DEC-23	50	1	4
99	108	30-DEC-23	80	3	3
110	110	31-DEC-23	80	3	2

8 rows selected.

Synonym DETALII_REPARARE dropped.