**ACADEMIA DE STUDII ECONOMICE DIN BUCUREŞTI**

**FACULTATEA DE CIBERNETICĂ, STATISTICĂ ŞI INFORMATICĂ ECONOMICĂ**

**PROIECT BAZE DE DATE:**

***GESTIUNE LA O SOCIETATE DE TRANSPORT PUBLIC***

**Student: BATOG FLORIN**

**Grupa: 1045**

**Seria C**

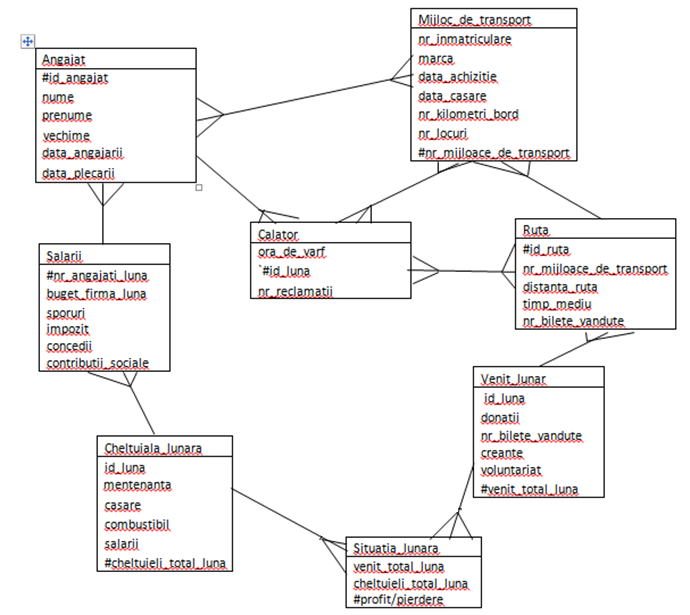
**I.DESCRIEREA BAZEI DE DATE**

Proiectul are ca scop evidenţa unei societăţi care are ca obiect de activitate transportul public de persoane. Acest proiect este realizat prin programul SQL developer şi este prezentat într-un format word. Proiectul conţine noţiuni fundamentale de baze de date, tabele, atribute, relaţii, restricţii, schema tabelelor, codul SQL pentru tabele şi comenzi sql.

Baza de date conţine informaţii despre angajat (id\_angajat, nume, prenume, vechime, data\_angajarii, data\_plecarii), salari (nr\_angajati\_luna, buget\_firma\_luna, sporuri, impozit, concedii, contributii\_sociale), mijloace de transport în comun (nr\_inmatriculare, marca, data\_achizitie, data\_casare, nr\_kilometri\_bord, nr\_locuri, nr\_mijloace\_de\_transport), călător (ora\_de\_varf, id\_luna, nr\_reclamatii), ruta (id\_ruta, nr\_mijloace\_de\_transport, distanta\_ruta, timp\_mediu, nr\_bilete\_vandute) pe care merge mijlocul de transport în comun, venitul lunar (id\_luna, donatii, nr\_bilete\_vandute, creante, voluntariat, venit\_total\_luna), cheltuiala lunară (id\_luna, mentenanta, casare, combustibil, salarii, cheltuieli\_total\_luna), situaţia lunară (venit\_total\_luna, cheltuieli\_total\_luna, profit\_pierdere). Tema aleasă are ca scop organizarea, evidenţa şi gestiunea societăţii de transport în comun.

**II.SCHEMA BAZEI DE DATE**

|  |  |  |
| --- | --- | --- |
| ENTITATI | TIP RELATIE | RELATII |
| 1.Angajat-Mijloc\_de\_transport | Many to many | Un angajat poate conduce mai multe mijloace de transport iar un mijloc de transport poate avea într-o săptămână mai mulţi angajaţi care să conducă acel mijloc de transport. |
| 2. Salarii- Angajat | One to many | Salariile pot fi distribuite la mai mulţi angajaţi dar un angajat nu poate avea mai multe salarii. |
| 3.Angajat- Calator | One to many | Un angajat poate avea mai mulţi călători dar un călător nu poate fi într-un mijloc de transport condus de mai mulţi angajaţi la un moment dat. |
| 4.Mijloc\_de\_transport- Calator | Many to many | Un călător poate urca în mai multe mijloace de transport iar un mijloc de transport poate avea mai mulţi călători. |
| 5.Ruta- Mijloc\_de\_transport | One to many | Un mijloc de transport poate fi direcţionat pe o singură rută dar o ruta poate avea mai multe mijloace de transport la dispoziţie. |
| 6.Ruta- Calator | Many to many | Un călător poate circula pe mai multe rute iar o rută poate avea mai mulţi călători . |
| 7.Venit\_lunar- Ruta | One to many | Rutele generază venituri prin biletele vândute însa veniturile nu pot genera rute. |
| 8.Cheltuiala\_lunara- Salarii | One to many | Salariile generează cheltuieli însă cheltuielile nu pot genera bani pentru mărirea salariilor. |
| 9. Cheltiuala\_lunara- Situatia\_lunara | One to many | Situaţia lunară este dată de cheltuieli. |
| 10. Venit\_lunar- Situatia\_lunara | One to many | Situaţia lunară este dată de venituri. |



**III. CREAREA TABELELOR**

**1. Să se creeze tabela Angajat. Tabela va avea coloanele: id\_angajat, nume, prenume, vechime, data\_angajarii, data\_plecarii. Tabela va avea cheie primară.**

CREATE TABLE Angajat(

id\_angajat NUMBER(3) NOT NULL,

CONSTRAINT PK\_Angajat PRIMARY KEY(id\_angajat),

nume VARCHAR2(20) NOT NULL,

prenume VARCHAR2(20) NOT NULL,

vechime NUMBER(2),

data\_angajarii DATE NOT NULL,

data\_plecarii DATE NOT NULL

);

**2. Să se creeze tabela Salarii. Tabela va avea coloanele: nr\_angajati\_luna, buget\_firma\_luna, sporuri, impozitconcedii, contributii\_sociale. Tabela va avea cheie primară.**

CREATE TABLE Salarii(

nr\_angajati\_luna NUMBER(3) NOT NULL,

CONSTRAINT PK\_Salarii PRIMARY KEY(nr\_angajati\_luna),

buget\_firma\_luna NUMBER(6) NOT NULL,

sporuri NUMBER(4) NOT NULL,

impozit NUMBER(3) NOT NULL,

concedii NUMBER(5) NOT NULL,

contributii\_sociale NUMBER(3) NOT NULL

);

**3. Să se creeze tabela Mijloc\_de\_transport. Tabela va avea coloanele: nr\_inmatriculare, marca, data\_achizitie, data\_casare, nr\_kilometri\_bord, nr\_locuri, nr\_mijloace\_de\_transport. Tabela va avea cheie primară.**

CREATE TABLE Mijloc\_de\_transport(

nr\_inmatriculare VARCHAR2(20) NOT NULL ,

marca VARCHAR2(20) NOT NULL,

data\_achizitie DATE NOT NULL,

data\_casare DATE NOT NULL,

nr\_kilometri\_bord NUMBER(8) NOT NULL,

nr\_locuri NUMBER(2) NOT NULL,

nr\_mijloace\_de\_transport NUMBER(4) NOT NULL,

CONSTRAINT PK\_Mijloc\_de\_transport PRIMARY KEY(nr\_mijloace\_de\_transport)

);

**4. Să se creeze tabela Calator. Tabela va avea coloanele: ora\_de\_varf, id\_luna, nr\_reclamatii. Tabela va avea cheie primară.**

CREATE TABLE Calator(

ora\_de\_varf TIMESTAMP(6) NOT NULL,

id\_luna DATE NOT NULL,

CONSTRAINT PK\_Calator PRIMARY KEY(id\_luna),

nr\_reclamatii NUMBER(3)

);

**5. Să se creeze tabela Ruta. Tabela va avea coloanele: id\_ruta, nr\_mijloace\_de\_transport, distanta\_ruta, timp\_mediu, nr\_bilete\_vandute. Tabela va avea o cheie primară.**

CREATE TABLE Ruta(

id\_ruta NUMBER(3) NOT NULL,

CONSTRAINT PK\_Ruta PRIMARY KEY(id\_ruta),

nr\_mijloace\_de\_transport NUMBER(3) NOT NULL,

distanta\_ruta NUMBER(2) NOT NULL,

timp\_mediu TIMESTAMP(6) NOT NULL,

nr\_bilete\_vandute NUMBER(6)

);

**6. Să se creeze tabela Venit\_lunar. Tabela va avea coloanele: id\_luna, donatii, nr\_bilete\_vandute, creante, voluntariat, venit\_total\_luna. Tabela va avea o cheie primară.**

CREATE TABLE Venit\_lunar(

id\_luna NUMBER(2) NOT NULL,

donatii NUMBER(3) NOT NULL,

nr\_bilete\_vandute NUMBER(6) NOT NULL,

creante NUMBER(6),

voluntariat NUMBER(3),

venit\_total\_luna NUMBER(6),

CONSTRAINT PK\_Venit\_lunar PRIMARY KEY(venit\_total\_luna)

);

**7. Să se creeze tabela Cheltuiala\_lunara. Tabela va avea coloanele: id\_luna, mentenanta, casare, combustibil, salarii, cheltuieli\_total\_luna. Tabela va avea cheie primară.**

CREATE TABLE Cheltuiala\_lunara(

id\_luna NUMBER(2) NOT NULL,

mentenanta NUMBER(6),

casare NUMBER(6),

combustibil NUMBER(6),

salarii NUMBER(6),

cheltuieli\_total\_luna NUMBER(6),

CONSTRAINT PK\_Cheltuiala\_lunara PRIMARY KEY(cheltuieli\_total\_luna )

);

**8. Să se creeze tabela Situatia\_lunara. Tabela va avea coloanele: venit\_total\_luna, cheltuieli\_total\_luna, profit\_pierdere. Tabela va avea cheie primară şi două chei externe.**

CREATE TABLE Situatia\_lunara(

venit\_total\_luna NUMBER(6),

CONSTRAINT FK\_Situatia\_lunara FOREIGN KEY(venit\_total\_luna) REFERENCES Venit\_lunar(venit\_total\_luna),

cheltuieli\_total\_luna NUMBER(6),

FOREIGN KEY(cheltuieli\_total\_luna) REFERENCES Cheltuiala\_lunara(cheltuieli\_total\_luna),

profit\_pierdere NUMBER(6)

);

**IV.ADĂUGAREA ȊNREGISTRĂRILOR ȊN TABELE + ACTUALIZAREA DE DATE, JONCȚIUNI, LUCRU CU FUNCȚII DE GRUP**

OBSERVAȚIE: De fiecare dată când am adăugat înregistrări pentru un tabel am efectuat şi comenzile legate de tabelul respectiv

**9.Să se adauge înregistrări pentru tabela Angajat.**

INSERT INTO Angajat(id\_angajat, nume, prenume, vechime, data\_angajarii, data\_plecarii) VALUES(100,'Petrache','Ion',20,TO\_DATE('11.10.1999','DD.MM.YYYY'),TO\_DATE('11.10.2019','DD.MM.YYYY'));

INSERT INTO Angajat(id\_angajat, nume, prenume, vechime, data\_angajarii, data\_plecarii) VALUES(101,'Marius','Tudor',19,TO\_DATE('12.08.2000','DD.MM.YYYY'),TO\_DATE('12.09.2019','DD.MM.YYYY'));

INSERT INTO Angajat(id\_angajat, nume, prenume, vechime, data\_angajarii, data\_plecarii) VALUES(102,'Vintila','Petru',13,TO\_DATE('18.04.2006','DD.MM.YYYY'),TO\_DATE('11.03.2019','DD.MM.YYYY'));

INSERT INTO Angajat(id\_angajat, nume, prenume, vechime, data\_angajarii, data\_plecarii) VALUES(103,'Popescu','Emil',4,TO\_DATE('7.07.2015','DD.MM.YYYY'),TO\_DATE('11.10.2019','DD.MM.YYYY'));

INSERT INTO Angajat(id\_angajat, nume, prenume, vechime, data\_angajarii, data\_plecarii) VALUES(104,'Vasile','Iulian',5,TO\_DATE('4.09.2014','DD.MM.YYYY'),TO\_DATE('11.11.2019','DD.MM.YYYY'));

INSERT INTO Angajat(id\_angajat, nume, prenume, vechime, data\_angajarii, data\_plecarii) VALUES(105,'Cezar','Dumitru',8,TO\_DATE('12.12.2010','DD.MM.YYYY'),TO\_DATE('12.12.2018','DD.MM.YYYY'));

INSERT INTO Angajat(id\_angajat, nume, prenume, vechime, data\_angajarii, data\_plecarii) VALUES(106,'Ilie','Florin',10,TO\_DATE('15.11.2009','DD.MM.YYYY'),TO\_DATE('15.11.2019','DD.MM.YYYY'));

INSERT INTO Angajat(id\_angajat, nume, prenume, vechime, data\_angajarii, data\_plecarii) VALUES(107,'Moraru','Nicolae',24,TO\_DATE('17.02.1994','DD.MM.YYYY'),TO\_DATE('19.05.2018','DD.MM.YYYY'));

INSERT INTO Angajat(id\_angajat, nume, prenume, vechime, data\_angajarii, data\_plecarii) VALUES(108,'Popescu','Grigore',15,TO\_DATE('11.01.2004','DD.MM.YYYY'),TO\_DATE('12.10.2019','DD.MM.YYYY'));

INSERT INTO Angajat(id\_angajat, nume, prenume, vechime, data\_angajarii, data\_plecarii) VALUES(109,'Baltag','Alexandru',9,TO\_DATE('1.02.2010','DD.MM.YYYY'),TO\_DATE('1.10.2019','DD.MM.YYYY'));

INSERT INTO Angajat(id\_angajat, nume, prenume, vechime, data\_angajarii, data\_plecarii) VALUES(110,'Petru','Iustin',17,TO\_DATE('11.02.2002','DD.MM.YYYY'),TO\_DATE('11.10.2019','DD.MM.YYYY'));

INSERT INTO Angajat(id\_angajat, nume, prenume, vechime, data\_angajarii, data\_plecarii) VALUES(111,'Iuliu','Mircea',12,TO\_DATE('23.07.2007','DD.MM.YYYY'),TO\_DATE('23.07.2019','DD.MM.YYYY'));

INSERT INTO Angajat(id\_angajat, nume, prenume, vechime, data\_angajarii, data\_plecarii) VALUES(112,'Popescu','Daniel',20,TO\_DATE('6.05.1999','DD.MM.YYYY'),TO\_DATE('6.08.2019','DD.MM.YYYY'));

**10. Să se steargă ultimul rând din tabela Angajat.**

DELETE FROM Angajat WHERE nume='Popescu' AND prenume='Daniel';

**11. Să se afişeze prenumele şi vechimea angajaţilor care au vechimea mai mare decât a lui Mircea.**

SELECT prenume, vechime FROM Angajat WHERE vechime>(SELECT vechime FROM Angajat WHERE prenume='Mircea');

**12. Să se afişeze numele şi prenumele angajaţilor a căror prenume încep cu litera I şi au vechime mai mare de 8 ani.**

SELECT nume, prenume FROM Angajat WHERE prenume LIKE 'I%' AND vechime>8;

**13. Să se afişeze numele şi data angajării angajatului care şi-a început activitatea ȋn cadrul societăţii după anul 2013.**

SELECT nume, data\_angajarii FROM Angajat WHERE EXTRACT(YEAR FROM TO\_DATE(data\_angajarii, 'DD-MON-RR'))>2013;

**14**. **Să se modifice prenumele angajatului Petru din Iustin în Iosif.**

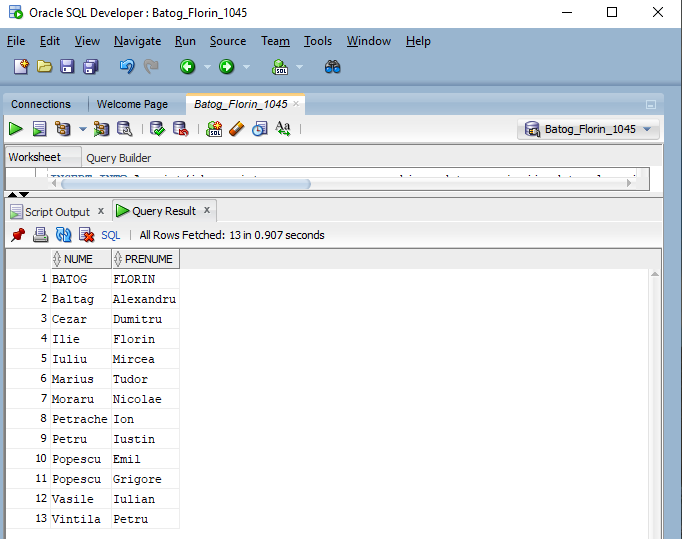
UPDATE Angajat SET prenume='Iosif' WHERE nume='Petru';

SELECT \* FROM Angajat;

**15. Să se adauge un nou angajat urmând a se afişa numele şi prenumele angajaţilor ordonate în funcţie de nume.** (**RȂNDUL CARE CONȚINE NUMELE ŞI PRENUMELE STUDENTULUI)**

INSERT INTO Angajat(id\_angajat, nume, prenume, vechime, data\_angajarii, data\_plecarii) VALUES(112,'BATOG','FLORIN',0,TO\_DATE('25.04.2009','DD.MM.YYYY'),TO\_DATE('11.02.2019','DD.MM.YYYY'));

SELECT nume, prenume FROM Angajat ORDER BY nume;



**16. Să se adauge înregistrări pentru tabela Salarii.**

INSERT INTO Salarii(nr\_angajati\_luna, buget\_firma\_luna, sporuri, impozit, concedii, contributii\_sociale) VALUES(123,990000,3000,980,22000,981);

INSERT INTO Salarii(nr\_angajati\_luna, buget\_firma\_luna, sporuri, impozit, concedii, contributii\_sociale) VALUES(122,980000,4000,830,21000,900);

INSERT INTO Salarii(nr\_angajati\_luna, buget\_firma\_luna, sporuri, impozit, concedii, contributii\_sociale) VALUES(125,970000,5000,850,22000,980);

INSERT INTO Salarii(nr\_angajati\_luna, buget\_firma\_luna, sporuri, impozit, concedii, contributii\_sociale) VALUES(120,965000,3500,900,23000,800);

INSERT INTO Salarii(nr\_angajati\_luna, buget\_firma\_luna, sporuri, impozit, concedii, contributii\_sociale) VALUES(119,960000,4300,870,24000,760);

INSERT INTO Salarii(nr\_angajati\_luna, buget\_firma\_luna, sporuri, impozit, concedii, contributii\_sociale) VALUES(116,950000,4330,800,22500,650);

INSERT INTO Salarii(nr\_angajati\_luna, buget\_firma\_luna, sporuri, impozit, concedii, contributii\_sociale) VALUES(118,940000,5000,843,21000,600);

INSERT INTO Salarii(nr\_angajati\_luna, buget\_firma\_luna, sporuri, impozit, concedii, contributii\_sociale) VALUES(112,930000,4330,980,20000,570);

INSERT INTO Salarii(nr\_angajati\_luna, buget\_firma\_luna, sporuri, impozit, concedii, contributii\_sociale) VALUES(120,920000,4000,950,19500,550);

INSERT INTO Salarii(nr\_angajati\_luna, buget\_firma\_luna, sporuri, impozit, concedii, contributii\_sociale) VALUES(113,910000,3500,900,19000,540);

INSERT INTO Salarii(nr\_angajati\_luna, buget\_firma\_luna, sporuri, impozit, concedii, contributii\_sociale) VALUES(114,905000,3000,850,18000,530);

INSERT INTO Salarii(nr\_angajati\_luna, buget\_firma\_luna, sporuri, impozit, concedii, contributii\_sociale) VALUES(115,900000,2500,800,17000,500);

**17. Să se afişeze minimul dintre sumele contributiilor pe care firma le are de luat de la angajatii sai lunar si bugetul maxim alocat lunar.**

SELECT MIN(sporuri+impozit+contributii\_sociale), MAX(buget\_firma\_luna) FROM Salarii WHERE nr\_angajati\_luna>=118;

**18. Să se adauge înregistrări pentru tabela Mijloc\_de\_transport.**

INSERT INTO Mijloc\_de\_transport(nr\_inmatriculare, marca, data\_achizitie, data\_casare, nr\_kilometri\_bord, nr\_locuri, nr\_mijloace\_de\_transport) VALUES('BV-01-RBV','Karsan',TO\_DATE('11.10.2009','DD.MM.YYYY'),TO\_DATE('11.10.2019','DD.MM.YYYY'),10000,23,100);

INSERT INTO Mijloc\_de\_transport(nr\_inmatriculare, marca, data\_achizitie, data\_casare, nr\_kilometri\_bord, nr\_locuri, nr\_mijloace\_de\_transport) VALUES('BV-02-RBV','BMC',TO\_DATE('7.10.2008','DD.MM.YYYY'),TO\_DATE('7.10.2018','DD.MM.YYYY'),800000,70,101);

INSERT INTO Mijloc\_de\_transport(nr\_inmatriculare, marca, data\_achizitie, data\_casare, nr\_kilometri\_bord, nr\_locuri, nr\_mijloace\_de\_transport) VALUES('BV-03-RBV','Mercedes',TO\_DATE('9.04.2001','DD.MM.YYYY'),TO\_DATE('9.09.2019','DD.MM.YYYY'),10500,23,120);

INSERT INTO Mijloc\_de\_transport(nr\_inmatriculare, marca, data\_achizitie, data\_casare, nr\_kilometri\_bord, nr\_locuri, nr\_mijloace\_de\_transport) VALUES('BV-04-RBV','Euro 6',TO\_DATE('11.10.2000','DD.MM.YYYY'),TO\_DATE('11.10.2019','DD.MM.YYYY'),900000,23,100);INSERT INTO Mijloc\_de\_transport(nr\_inmatriculare, marca, data\_achizitie, data\_casare, nr\_kilometri\_bord, nr\_locuri, nr\_mijloace\_de\_transport) VALUES('BV-05-RBV','Euro 5',TO\_DATE('11.10.2002','DD.MM.YYYY'),TO\_DATE('11.10.2017','DD.MM.YYYY'),9000,23,103);

INSERT INTO Mijloc\_de\_transport(nr\_inmatriculare, marca, data\_achizitie, data\_casare, nr\_kilometri\_bord, nr\_locuri, nr\_mijloace\_de\_transport) VALUES('BV-06-RBV','Otokar',TO\_DATE('1.03.2004','DD.MM.YYYY'),TO\_DATE('11.10.2014','DD.MM.YYYY'),850000,23,102);

INSERT INTO Mijloc\_de\_transport(nr\_inmatriculare, marca, data\_achizitie, data\_casare, nr\_kilometri\_bord, nr\_locuri, nr\_mijloace\_de\_transport) VALUES('BV-07-RBV','Ikarus',TO\_DATE('11.10.2009','DD.MM.YYYY'),TO\_DATE('11.10.2019','DD.MM.YYYY'),790000,23,99);

INSERT INTO Mijloc\_de\_transport(nr\_inmatriculare, marca, data\_achizitie, data\_casare, nr\_kilometri\_bord, nr\_locuri, nr\_mijloace\_de\_transport) VALUES('BV-08-RBV','Solaris',TO\_DATE('13.06.2007','DD.MM.YYYY'),TO\_DATE('11.10.2017','DD.MM.YYYY'),9800,23,110;

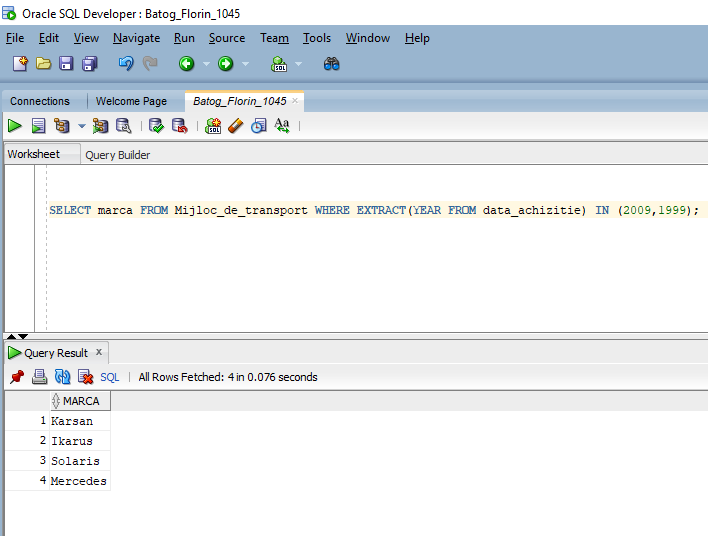
INSERT INTO Mijloc\_de\_transport(nr\_inmatriculare, marca, data\_achizitie, data\_casare, nr\_kilometri\_bord, nr\_locuri, nr\_mijloace\_de\_transport) VALUES('BV-09-RBV','Solaris',TO\_DATE('11.10.2009','DD.MM.YYYY'),TO\_DATE('11.10.2019','DD.MM.YYYY'),860000,23,98);INSERT INTO Mijloc\_de\_transport(nr\_inmatriculare, marca, data\_achizitie, data\_casare, nr\_kilometri\_bord, nr\_locuri, nr\_mijloace\_de\_transport) VALUES('BV-10-RBV','BMC',TO\_DATE('13.12.2006','DD.MM.YYYY'),TO\_DATE('12.03.2019','DD.MM.YYYY'),8000,23,104);

INSERT INTO Mijloc\_de\_transport(nr\_inmatriculare, marca, data\_achizitie, data\_casare, nr\_kilometri\_bord, nr\_locuri, nr\_mijloace\_de\_transport) VALUES('BV-11-RBV','Mercedes',TO\_DATE('15.02.1999','DD.MM.YYYY'),TO\_DATE('14.10.2009','DD.MM.YYYY'),780000,23,106);

INSERT INTO Mijloc\_de\_transport(nr\_inmatriculare, marca, data\_achizitie, data\_casare, nr\_kilometri\_bord, nr\_locuri, nr\_mijloace\_de\_transport) VALUES('BV-12-RBV','Berkhoff',TO\_DATE('1.06.1993','DD.MM.YYYY'),TO\_DATE('10.01.2013','DD.MM.YYYY'),12345,60,10);

**19. Să se afişeze marca mijloacelor de transport care au fost achiziţionate în anul 2009 sau 1999.**

SELECT marca FROM Mijloc\_de\_transport WHERE EXTRACT(YEAR FROM data\_achizitie) IN (2009,1999);



**20. Să se ştearga coloana ora\_de\_varf.**

ALTER TABLE Calator DROP COLUMN ora\_de\_varf;

**21. Să se adauge înregistrări pentru tabela Calator.**

INSERT INTO Calator(id\_luna, nr\_reclamatii) VALUES(TO\_DATE('11.12.2019','DD.MM.YYYY'),122);

INSERT INTO Calator(id\_luna, nr\_reclamatii) VALUES(TO\_DATE('1.01.2013','DD.MM.YYYY'),111);

INSERT INTO Calator(id\_luna, nr\_reclamatii) VALUES(TO\_DATE('13.06.2014','DD.MM.YYYY'),100);

INSERT INTO Calator(id\_luna, nr\_reclamatii) VALUES(TO\_DATE('5.08.2015','DD.MM.YYYY'),109);

INSERT INTO Calator(id\_luna, nr\_reclamatii) VALUES(TO\_DATE('6.02.2009','DD.MM.YYYY'),113);

INSERT INTO Calator(id\_luna, nr\_reclamatii) VALUES(TO\_DATE('7.12.2010','DD.MM.YYYY'),120);

INSERT INTO Calator(id\_luna, nr\_reclamatii) VALUES(TO\_DATE('8.11.2010','DD.MM.YYYY'),101);

INSERT INTO Calator(id\_luna, nr\_reclamatii) VALUES(TO\_DATE('9.10.2011','DD.MM.YYYY'),111);

INSERT INTO Calator(id\_luna, nr\_reclamatii) VALUES(TO\_DATE('19.05.2016','DD.MM.YYYY'),101);

INSERT INTO Calator(id\_luna, nr\_reclamatii) VALUES(TO\_DATE('14.04.2017','DD.MM.YYYY'),105);

INSERT INTO Calator(id\_luna, nr\_reclamatii) VALUES(TO\_DATE('12.03.2018','DD.MM.YYYY'),106);

INSERT INTO Calator(id\_luna, nr\_reclamatii) VALUES(TO\_DATE('12.05.2008','DD.MM.YYYY'),117);

**22. Să se redenumească coloana id\_luna în data\_inregistrare.**

ALTER TABLE Calator RENAME COLUMN id\_luna TO data\_inregistrare;

**23. Să se adauge înregistrări pentru tabela Ruta.**

INSERT INTO Ruta(id\_ruta, nr\_mijloace\_de\_transport, distanta\_ruta, timp\_mediu, nr\_bilete\_vandute) VALUES(100,134,2,20,123);

INSERT INTO Ruta(id\_ruta, nr\_mijloace\_de\_transport, distanta\_ruta, timp\_mediu, nr\_bilete\_vandute) VALUES(101,143,3,30,100);

INSERT INTO Ruta(id\_ruta, nr\_mijloace\_de\_transport, distanta\_ruta, timp\_mediu, nr\_bilete\_vandute) VALUES(102,123,1,10,154);

INSERT INTO Ruta(id\_ruta, nr\_mijloace\_de\_transport, distanta\_ruta, timp\_mediu, nr\_bilete\_vandute) VALUES(103,132,2,20,134);

INSERT INTO Ruta(id\_ruta, nr\_mijloace\_de\_transport, distanta\_ruta, timp\_mediu, nr\_bilete\_vandute) VALUES(104,193,4,40,153);

INSERT INTO Ruta(id\_ruta, nr\_mijloace\_de\_transport, distanta\_ruta, timp\_mediu, nr\_bilete\_vandute) VALUES(105,139,5,50,121);

INSERT INTO Ruta(id\_ruta, nr\_mijloace\_de\_transport, distanta\_ruta, timp\_mediu, nr\_bilete\_vandute) VALUES(106,141,1,20,125);

INSERT INTO Ruta(id\_ruta, nr\_mijloace\_de\_transport, distanta\_ruta, timp\_mediu, nr\_bilete\_vandute) VALUES(107,114,3,30,109);

INSERT INTO Ruta(id\_ruta, nr\_mijloace\_de\_transport, distanta\_ruta, timp\_mediu, nr\_bilete\_vandute) VALUES(108,131,2,20,113);

INSERT INTO Ruta(id\_ruta, nr\_mijloace\_de\_transport, distanta\_ruta, timp\_mediu, nr\_bilete\_vandute) VALUES(109,113,1,10,112);

INSERT INTO Ruta(id\_ruta, nr\_mijloace\_de\_transport, distanta\_ruta, timp\_mediu, nr\_bilete\_vandute) VALUES(110,160,5,50,120);

INSERT INTO Ruta(id\_ruta, nr\_mijloace\_de\_transport, distanta\_ruta, timp\_mediu, nr\_bilete\_vandute) VALUES(111,106,2,20,129);

**24. Să se afişeze numărul mijloacelor de transport care au id-ul între 103 şi 109.**

SELECT id\_ruta, COUNT(nr\_mijloace\_de\_transport) FROM Ruta GROUP BY id\_ruta HAVING id\_ruta BETWEEN 103 AND 109;

**25. Să se afişeze marca mijlocului de transport şi numărul de bilete vândute în acel mijloc de transport.**

SELECT marca,nr\_bilete\_vandute FROM Mijloc\_de\_transport INNER JOIN Ruta ON Mijloc\_de\_transport.nr\_mijloace\_de\_transport =Ruta.nr\_mijloace\_de\_transport;

**26. Să se afişeze id\_ruta şi numărul mijloacelor de transport disponibile pe rutele respective. Rezultatul va lua valorile comune dintre criteriile id\_ruta>=105 sau nr\_mijloace\_de\_transport>140 şi id\_ruta<101 sau nr\_mijloace\_de\_transport<180.**

SELECT id\_ruta, nr\_mijloace\_de\_transport FROM Ruta WHERE id\_ruta>=105 OR nr\_mijloace\_de\_transport>140 INTERSECT (SELECT id\_ruta, nr\_mijloace\_de\_transport FROM Ruta WHERE id\_ruta<101 OR nr\_mijloace\_de\_transport<180);

**27. Să se afişeze id\_ruta în funcţie de câte bilete sunt vândute astfel: 1 dacă ruta este slab circulată, 2 dacă ruta este circulată la nivel mediu iar cu 3 rutele care sunt intens circulate. Coloana care va avea aceste informaţii va purta denumirea de Mesaj.**

SELECT id\_ruta, CASE WHEN nr\_bilete\_vandute<110 THEN 1 WHEN nr\_bilete\_vandute BETWEEN 110 AND 145 THEN 2 WHEN nr\_bilete\_vandute>110 THEN 3 ELSE 0 END Mesaj FROM Ruta;

**28. Să se adauge înregistrări pentru tabela Venit\_lunar.**

INSERT INTO Venit\_lunar(id\_luna, donatii, nr\_bilete\_vandute, creante, voluntariat, venit\_total\_luna) VALUES (1,12,1000,130,100,1521);

INSERT INTO Venit\_lunar(id\_luna, donatii, nr\_bilete\_vandute, creante, voluntariat, venit\_total\_luna) VALUES (2,21,1200,230,99,1545);

INSERT INTO Venit\_lunar(id\_luna, donatii, nr\_bilete\_vandute, creante, voluntariat, venit\_total\_luna) VALUES (3,32,1300,450,98,1657);

INSERT INTO Venit\_lunar(id\_luna, donatii, nr\_bilete\_vandute, creante, voluntariat, venit\_total\_luna) VALUES (4,23,1400,670,97,1700);

INSERT INTO Venit\_lunar(id\_luna, donatii, nr\_bilete\_vandute, creante, voluntariat, venit\_total\_luna) VALUES (5,45,1500,870,96,1800);

INSERT INTO Venit\_lunar(id\_luna, donatii, nr\_bilete\_vandute, creante, voluntariat, venit\_total\_luna) VALUES (6,54,1600,543,87,1900);

INSERT INTO Venit\_lunar(id\_luna, donatii, nr\_bilete\_vandute, creante, voluntariat, venit\_total\_luna) VALUES (7,32,1700,123,97,2000);

INSERT INTO Venit\_lunar(id\_luna, donatii, nr\_bilete\_vandute, creante, voluntariat, venit\_total\_luna) VALUES (8,21,1700,321,100,2500);

INSERT INTO Venit\_lunar(id\_luna, donatii, nr\_bilete\_vandute, creante, voluntariat, venit\_total\_luna) VALUES (9,56,1800,576,88,2500);

INSERT INTO Venit\_lunar(id\_luna, donatii, nr\_bilete\_vandute, creante, voluntariat, venit\_total\_luna) VALUES (10,66,1900,876,84,2400);

INSERT INTO Venit\_lunar(id\_luna, donatii, nr\_bilete\_vandute, creante, voluntariat, venit\_total\_luna) VALUES (11,77,2000,342,80,2300);

INSERT INTO Venit\_lunar(id\_luna, donatii, nr\_bilete\_vandute, creante, voluntariat, venit\_total\_luna) VALUES (12,88,3000,130,100,2800);

**29. Să se adauge o restricţie la id\_luna astfel încât id\_luna să ia valori între 1 şi 12.**

ALTER TABLE Venit\_lunar ADD CONSTRAINT CHK\_id\_luna CHECK(id\_luna BETWEEN 1 AND 12);

**30. Să se afişeze veniturile totale lunare ierarhizate în funcţie de suma celorlalte coloane exceptând id\_luna.**

SELECT venit\_total\_luna FROM Venit\_lunar ORDER BY venit\_total\_luna=creante+donatii+nr\_bilete\_vandute+voluntariat;

**31. Să se adauge înregistrări pentru tabela Cheltuiala\_lunara.**

INSERT INTO Cheltuiala\_lunara(id\_luna, mentenanta, casare, combustibil, salarii, cheltuieli\_total\_luna) VALUES(1,1500,1200,1300,1800,2300);

INSERT INTO Cheltuiala\_lunara(id\_luna, mentenanta, casare, combustibil, salarii, cheltuieli\_total\_luna) VALUES(2,1600,1300,1200,1900,2200);

INSERT INTO Cheltuiala\_lunara(id\_luna, mentenanta, casare, combustibil, salarii, cheltuieli\_total\_luna) VALUES(3,1700,1400,1100,2000,2100);

INSERT INTO Cheltuiala\_lunara(id\_luna, mentenanta, casare, combustibil, salarii, cheltuieli\_total\_luna) VALUES(4,1800,1500,1000,2100,2000);

INSERT INTO Cheltuiala\_lunara(id\_luna, mentenanta, casare, combustibil, salarii, cheltuieli\_total\_luna) VALUES(5,1900,1600,1300,2200,1900);

INSERT INTO Cheltuiala\_lunara(id\_luna, mentenanta, casare, combustibil, salarii, cheltuieli\_total\_luna) VALUES(6,2000,1700,1400,2300,1800);

INSERT INTO Cheltuiala\_lunara(id\_luna, mentenanta, casare, combustibil, salarii, cheltuieli\_total\_luna) VALUES(7,1800,1800,1500,2400,1700);

INSERT INTO Cheltuiala\_lunara(id\_luna, mentenanta, casare, combustibil, salarii, cheltuieli\_total\_luna) VALUES(8,1700,1900,1600,2500,1600);

INSERT INTO Cheltuiala\_lunara(id\_luna, mentenanta, casare, combustibil, salarii, cheltuieli\_total\_luna) VALUES(9,1600,2200,1700,2000,1500);

INSERT INTO Cheltuiala\_lunara(id\_luna, mentenanta, casare, combustibil, salarii, cheltuieli\_total\_luna) VALUES(10,1500,2000,1800,1700,1400);

INSERT INTO Cheltuiala\_lunara(id\_luna, mentenanta, casare, combustibil, salarii, cheltuieli\_total\_luna) VALUES(11,1400,1700,1900,1600,1300);

INSERT INTO Cheltuiala\_lunara(id\_luna, mentenanta, casare, combustibil, salarii, cheltuieli\_total\_luna) VALUES(12,1300,1600,2000,1500,1200);

**32. Să se adauge o restricţie la id\_luna astfel încât id\_luna să ia valori între 1 şi 12.**

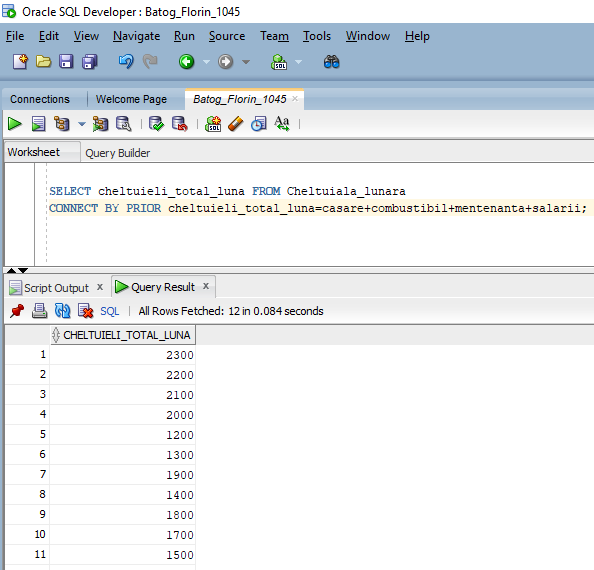
ALTER TABLE Cheltuiala\_lunara ADD CONSTRAINT CHK\_id\_lunar CHECK(id\_luna BETWEEN 1 AND 12);

**33. Să se afişeze veniturile şi cheltuielile pentru fiecare lună.**

SELECT venit\_total\_luna, cheltuieli\_total\_luna FROM Venit\_lunar INNER JOIN Cheltuiala\_lunara ON Venit\_lunar.id\_luna=Cheltuiala\_lunara.id\_luna;

**34. Să se afişeze cheltuielile totale lunare ierarhizate în funcţie de suma celorlalte coloane exceptând id\_luna.**

SELECT cheltuieli\_total\_luna FROM Cheltuiala\_lunara ORDER BY cheltuieli\_total\_luna=casare+combustibil+mentenanta+salarii;



**35. Să se afişeze id\_luna şi cheltuielile cu salariile din lunile respective unde valoarea este situată între 1700 şi 2800 exceptând valorile cuprinse între 2100 şi 2400.**

SELECT id\_luna, salarii FROM Cheltuiala\_lunara WHERE salarii BETWEEN 1700 AND 2800 MINUS (SELECT id\_luna, salarii FROM Cheltuiala\_lunara WHERE salarii BETWEEN 2100 AND 2400);

**36. Să se adauge înregistrări pentru tabela Situatia\_financiara.**

INSERT INTO Situatia\_lunara(venit\_total\_luna, cheltuieli\_total\_luna, profit\_pierdere) VALUES(1800,1500,300);

INSERT INTO Situatia\_lunara(venit\_total\_luna, cheltuieli\_total\_luna, profit\_pierdere) VALUES(1900,1800,100);

INSERT INTO Situatia\_lunara(venit\_total\_luna, cheltuieli\_total\_luna, profit\_pierdere) VALUES(1700,1900,200);

INSERT INTO Situatia\_lunara(venit\_total\_luna, cheltuieli\_total\_luna, profit\_pierdere) VALUES(2500,1500,1000);

INSERT INTO Situatia\_lunara(venit\_total\_luna, cheltuieli\_total\_luna, profit\_pierdere) VALUES(2300,1400,900);

INSERT INTO Situatia\_lunara(venit\_total\_luna, cheltuieli\_total\_luna, profit\_pierdere) VALUES(2300,1800,100);

INSERT INTO Situatia\_lunara(venit\_total\_luna, cheltuieli\_total\_luna, profit\_pierdere) VALUES(1900,1800,700);

INSERT INTO Situatia\_lunara(venit\_total\_luna, cheltuieli\_total\_luna, profit\_pierdere) VALUES(2400,1600,0);

**37. Să se şteargă rândul unde profit\_pierdere=0;**

DELETE FROM Situatia\_lunara WHERE profit\_pierdere=0;

**38**. **Să se şteargă rândul unde profit\_pierdere=700;**

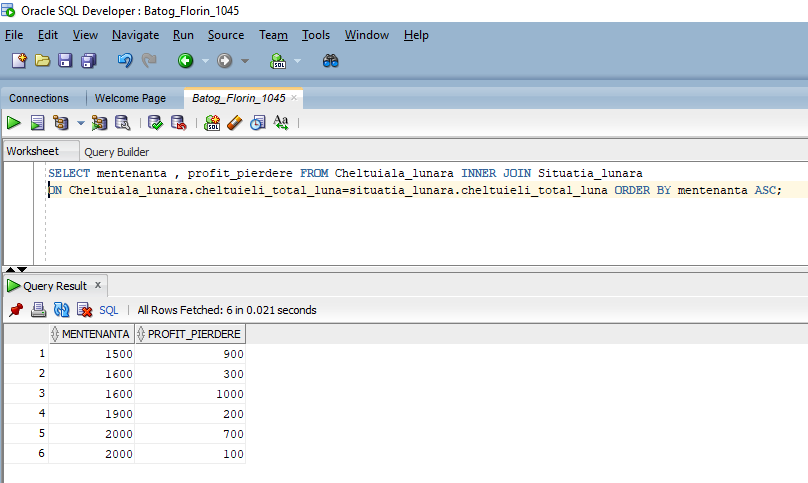
DELETE FROM Situatia\_lunara WHERE profit\_pierdere=700;

**39. Să se afişeze diferenţa dintre veniturile şi cheltuielile lunare înregistrate de societatea de transport.**

SELECT venit\_total\_luna, cheltuieli\_total\_luna, profit\_pierdere FROM Situatia\_lunara WHERE profit\_pierdere=venit\_total\_luna-cheltuieli\_total\_luna;

**40. Să se afişeze costul întreţinerii mijloacelor de transport şi contul de profit şi pierderi.**

SELECT mentenanta , profit\_pierdere FROM Cheltuiala\_lunara INNER JOIN Situatia\_lunara ON Cheltuiala\_lunara.cheltuieli\_total\_luna=situatia\_lunara.cheltuieli\_total\_luna ORDER BY mentenanta ASC;



**CONCLUZIE**

Baza de date realizată pentru societatea de transport public este utilă din punct de vedere organizatoric, ajută la eficientizarea timpului de lucru şi la luarea deciziilor privind alocarea bugetului societăţii în funcţie de rezultatele precedente, pentru a evita greşelile anterioare.

**BIBLIOGRAFIE**

1. Conf. Dr. Diaconița Vlad, “*Baze de date Curs 7 Tipuri de baze de date, arhitecturi pentru bazele de date”*
2. <http://bd.ase.ro/>