

Proiect SGBD - Oracle

Georgescu Ionuț-Bogdan

Baza de date este destinată gestionării unui sistem de transport feroviar pentru pasageri unde sunt gestionate informații despre trenuri, vagoane, pasageri, rute, bilete și angajați. Aceasta poate fi utilizată de către companiile feroviare pentru a urmări rutele disponibile, biletele vândute și angajații care operează trenurile. Este de menționat faptul că toate atributele au restricția NOT NULL.

Tabele, atribute și restricții

Tabela **RUTE** cuprinde informații despre rutele operatorului: **ID_RUTA** [PRIMARY KEY]; **DURATA** – durata călătoriei în minute; **NR_KM** – numărul de km al rutei; **PLECARE** – stația de plecare; **SOSIRE** – stația de sosire;

Tabela **PASAGERI** cuprinde informații despre pasageri: **ID_PASAGER** [PRIMARY KEY]; **NUME** – numele pasagerului; **PRENUME** – prenumele pasagerului; **TELEFON** – numărul de telefon al pasagerului; **EMAIL** – adresa de e-mail a pasagerului; **TIP_BILET** – Din gară/Fizic, Online; **ID_RUTA** [FOREIGN KEY];

Tabela **BILETE** cuprinde informații despre biletele pasagerilor: **ID_BILET** [PRIMARY KEY]; **TIP** – Clasa 1, Clasa 2; **LOC** – numărul locului pasagerului; **VAGON** – nr. Vagonului; **DATA_EMITERII** – data la care a fost emis biletul; **DATA_CALATORIEI** [CHECK] – data călătoriei (diferența dintre data emiterii biletului și data călătoriei nu trebuie să fie mai mare de 30 de zile); **PRET** – pretul biletului; **ID_RUTA** [FOREIGN KEY]; **ID_PASAGER** [FOREIGN KEY];

Tabela **TRENURI** cuprinde informații despre trenurile operatorului: **ID_TREN** [PRIMARY KEY]; **TIP_TREN** – Regio, InterRegio; **NR_VAGOANE** NUMBER(2) – numărul vagoanelor aflate în compunerea trenului; **NR_PASAGERI** – numărul călătorilor din tren; **ID_RUTA** [FOREIGN KEY];

Tabela **VAGOANE** cuprinde informații despre vagoanele aflate în compunerea trenurilor: **ID_VAGON** [PRIMARY KEY]; **TIP_VAGON** – Vagon salon, Cușetă, etc.; **CAPACITATE** – capacitatea vagonului; **ID_TREN** [FOREIGN KEY];

Tabela **PERSONAL_TREN** cuprinde informații despre personal: **ID_ANGAJAT** [PRIMARY KEY]; **NUME** – numele angajatului; **PRENUME** – prenumele angajatului; **FUNCTIE** – denumirea funcției pe care angajatul o ocupă; **SALARIUL** – salariul angajatului; **ID_TREN** [FOREIGN KEY];

Tabela **STATII_RUTE** reprezintă o tabelă de ascoiere pentru a evita apariția unei relații N:N și cuprinde informații precum **ID_RUTA** [PRIMARY AND FOREIGN KEY]; **ID_STATIE** [PRIMARY AND FOREIGN KEY]; **SOSIRE** – ora sosirii; **PLECARE** – ora plecării;

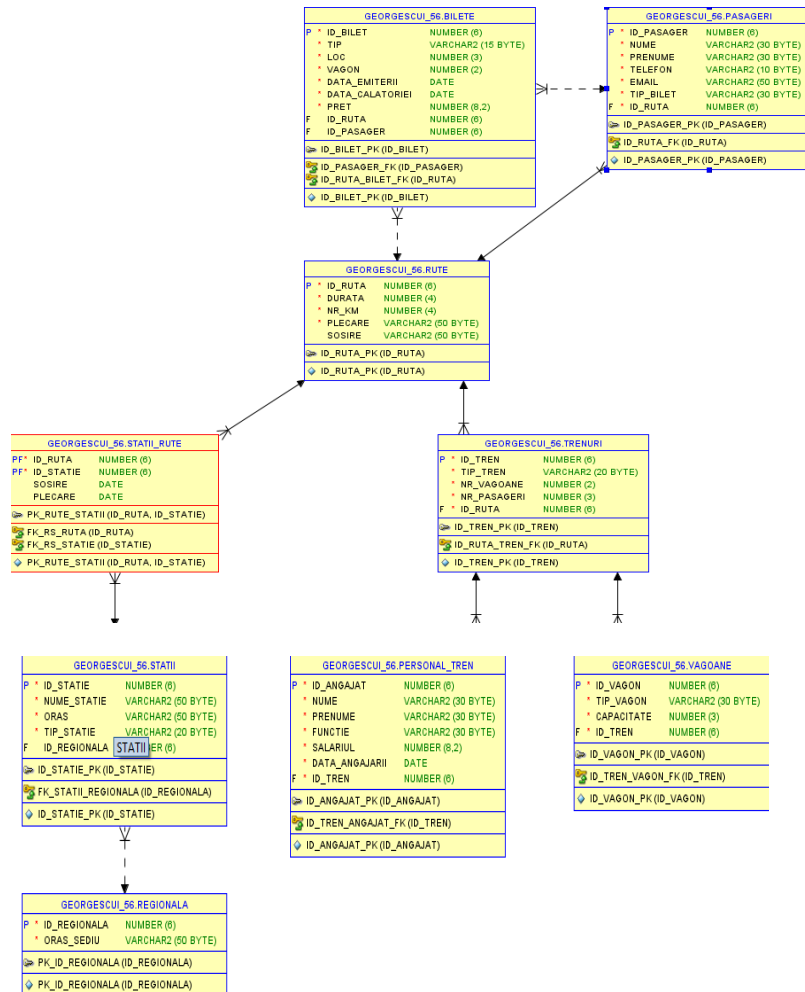
Tabela **STATII** cuprinde informații despre stațiile aflate pe o rută cum ar fi **ID_STATIE** [PRIMARY KEY]; **NUME_STATIE**; **ORAS**; **TIP_STATIE** – joncțiune, intermediară, terminus etc; **ID_REGIONALA** [FOREIGN KEY];

Tabela **REGIONALA** cuprinde informații despre regionalele care administrează stațiile: **ID_REGIONALA** [PRIMARY KEY]; **ORAS_SEDIU**;

Legăturile între tabele:

- Între tabela **RUTE** și **TRENURI** există o relație de 1:N prin coloana **ID_RUTA**.
- Între tabela **RUTE** și **PASAGERI** există o relație de 1:N prin coloana **ID_RUTA**.
- Între tabela **RUTE** și **BILETE** există o relație de 1:N prin coloana **ID_RUTA**.
- Între tabela **PASAGERI** și **BILETE** există o relație de 1:N prin coloana **ID_PASAGER**.
- Între tabela **TRENURI** și **VAGOANE** există o relație de 1:N prin coloana **ID_TREN**.
- Între tabela **TRENURI** și **PERSONAL_TREN** există o relație de 1:N prin coloana **ID_TREN**.
- Între tabela **RUTE** și **STATII_RUTE** există o relație de 1:N prin coloana **ID_RUTA**.
- Între tabela **STATII** și **STATII_RUTE** există o relație 1:N prin coloana **ID_STATIE**.
- Între tabela **REGIONALA** și **STATII** există o relație 1:N prin coloana **ID_REGIONALA**.

Schema bazei de date



Actualizarea inregistrarilor

/*Sa se scada capacitatea cu 2 locuri pentru vagoanele de tip 'Automotor'.*/

UPDATE VAGOANE

SET CAPACITATE = CAPACITATE - 2

WHERE UPPER(TIP_VAGON) = 'AUTOMOTOR';

/*Sa se mareasca salariile tuturor angajatilor de pe trenurile cu mai mult de 100 de pasageri*/

UPDATE PERSONAL_TREN

SET SALARIUL = 1.1 * SALARIUL

WHERE ID_TREN IN (SELECT ID_TREN FROM TRENURI WHERE NR_PASAGERI > 100);

/*Sa se mareasca cu 20% salariul controlorilor de bilete care au un salariu mai mic decat salariul mediu al controlorilor de bilete*/

UPDATE PERSONAL_TREN

SET SALARIUL = 1.2 * SALARIUL

WHERE UPPER(FUNCTIE) = 'CONTROLOR BILETE' AND SALARIUL < (SELECT
AVG(SALARIUL) FROM PERSONAL_TREN WHERE UPPER(FUNCTIE) =
'CONTROLOR BILETE');

/*Sa se mareasca cu 20% pretul biletelor de pe rutele mai mari de 50 de km*/

UPDATE BILETE

SET PRET = 1.2 * PRET

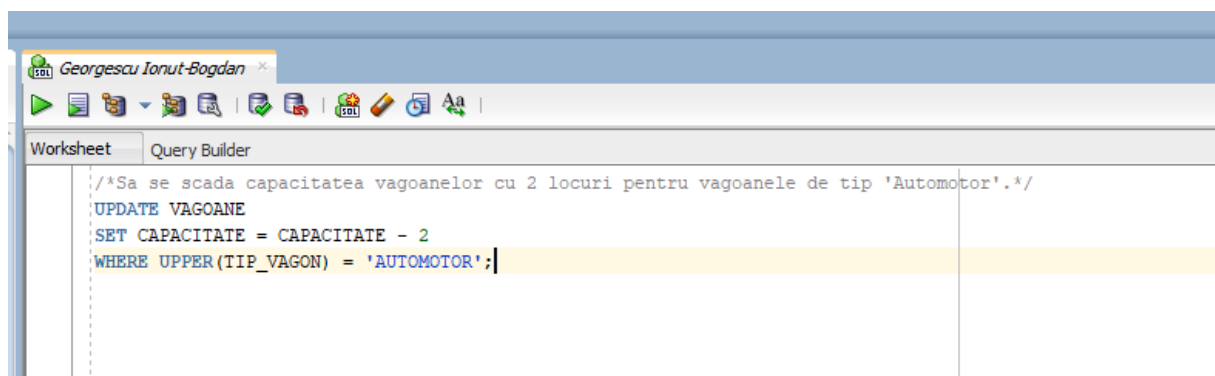
WHERE ID_RUTA IN (SELECT ID_RUTA FROM RUTE WHERE NR_KM > 50);

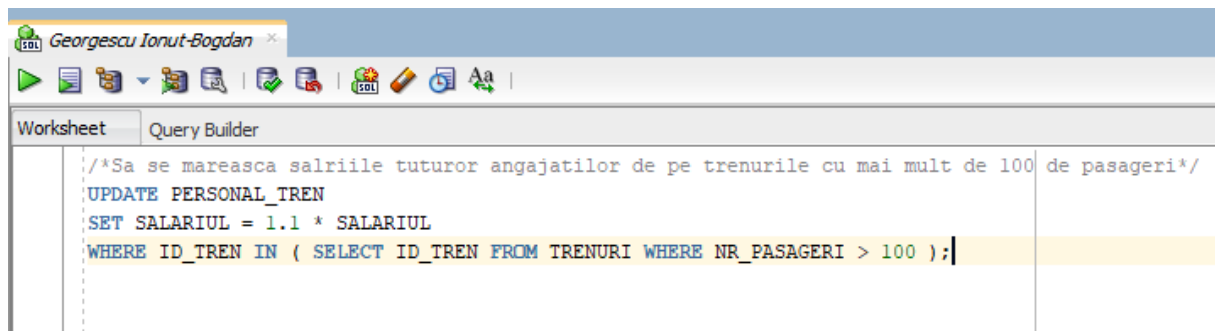
/*Sa se elimine toate bilete de pe ruta cu ID_RUTA = 171218. La final sa se anuleze comanda.*/

DELETE FROM BILETE

WHERE ID_RUTA = 171218;

ROLLBACK;

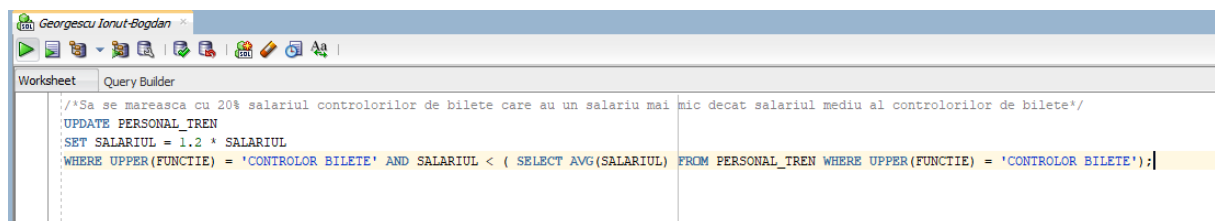




Georgescu Ionut-Bogdan

Worksheet Query Builder

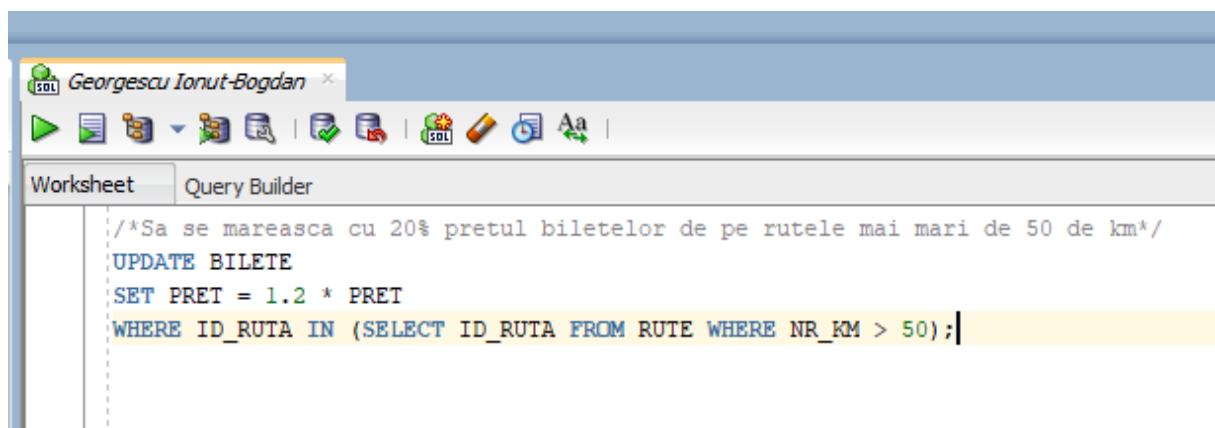
```
/*Sa se mareasca salariile tuturor angajatilor de pe trenurile cu mai mult de 100 de pasageri*/  
UPDATE PERSONAL_TREN  
SET SALARIUL = 1.1 * SALARIUL  
WHERE ID_TREN IN ( SELECT ID_TREN FROM TRENURI WHERE NR_PASAGERI > 100 );
```



Georgescu Ionut-Bogdan

Worksheet Query Builder

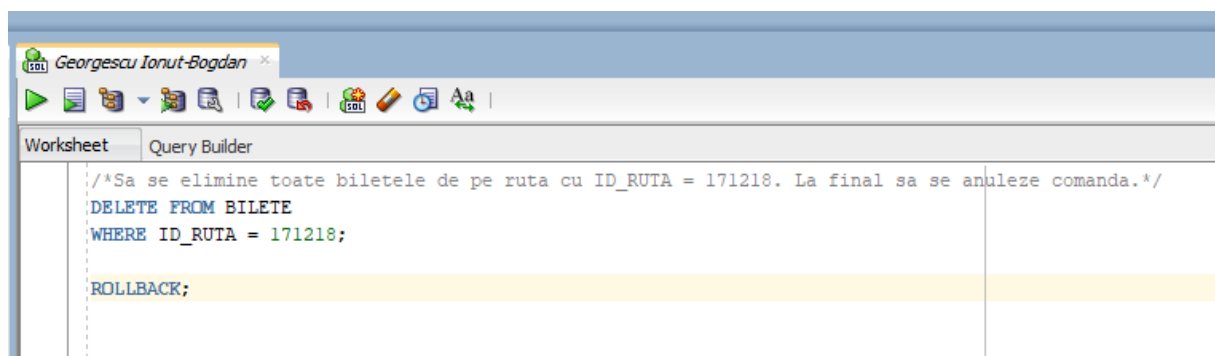
```
/*Sa se mareasca cu 20% salariul controlorilor de bilete care au un salariu mai mic decat salariul mediu al controlorilor de bilete*/  
UPDATE PERSONAL_TREN  
SET SALARIUL = 1.2 * SALARIUL  
WHERE UPPER(FUNCTIE) = 'CONTROLOR BILETE' AND SALARIUL < ( SELECT AVG(SALARIUL) FROM PERSONAL_TREN WHERE UPPER(FUNCTIE) = 'CONTROLOR BILETE');
```



Georgescu Ionut-Bogdan

Worksheet Query Builder

```
/*Sa se mareasca cu 20% pretul biletelor de pe rutele mai mari de 50 de km*/  
UPDATE BILETE  
SET PRET = 1.2 * PRET  
WHERE ID_RUTA IN (SELECT ID_RUTA FROM RUTE WHERE NR_KM > 50);
```

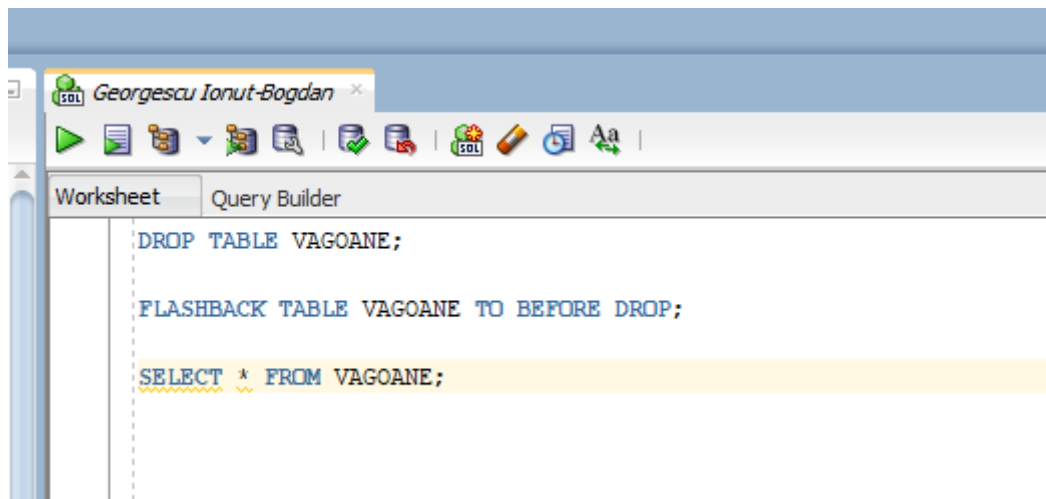


Georgescu Ionut-Bogdan

Worksheet Query Builder

```
/*Sa se elimine toate biletele de pe ruta cu ID_RUTA = 171218. La final sa se anuleze comanda.*/  
DELETE FROM BILETE  
WHERE ID_RUTA = 171218;  
  
ROLLBACK;
```

Stergerea tabelii VAGOANE și recuperarea acesteia



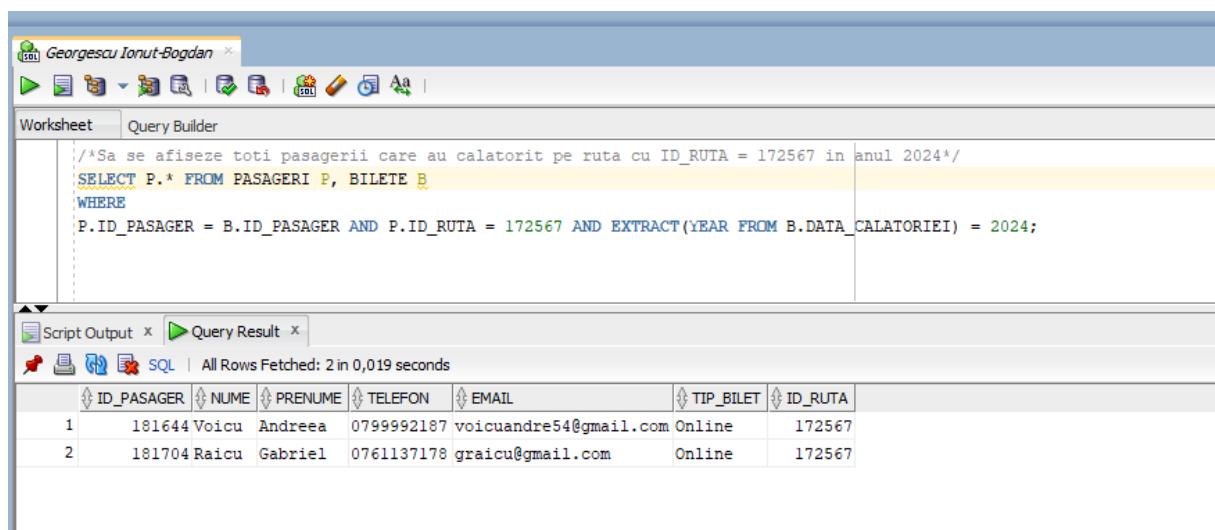
Exemple de interogari

1. Să se afișeze toți pasagerii care au călătorit pe ruta cu ID_RUTA = 172567 în anul 2024.

SELECT P.* FROM PASAGERI P, BILETE B

WHERE

P.ID_PASAGER = B.ID_PASAGER AND P.ID_RUTA = 172567 AND EXTRACT(YEAR FROM B.DATA_CALATORIEI) = 2024;



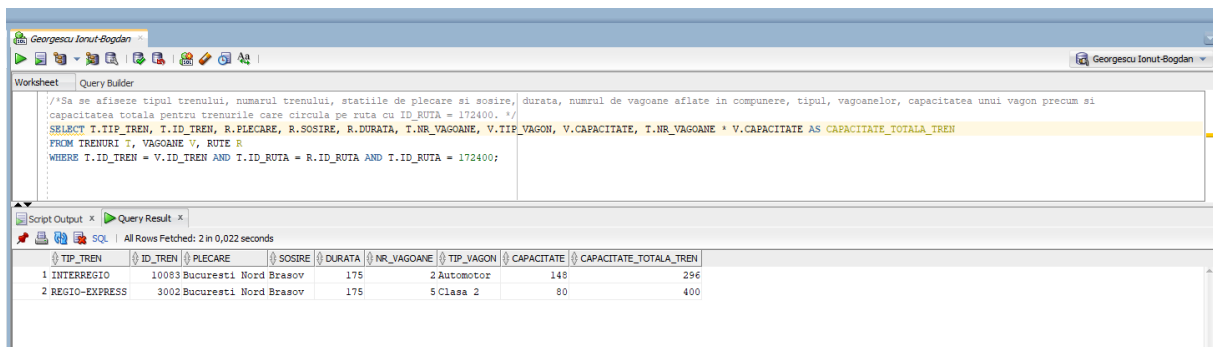
2. Sa se afiseze tipul trenului, numarul trenului, statiile de plecare si sosire, durata, numrul de vagoane aflate in compunere, tipul, vagoanelor, capacitatea unui vagon

precum si capacitatea totala pentru trenurile care circula pe ruta cu ID_RUTA = 172400.

```
SELECT T.TIP_TREN, R.PLECARE, R.SOSIRE, R.DURATA, T.NR VAGOANE,
V.TIP_VAGON, V.CAPCITATE, T.NR_VAGOANE * V.CAPACITATE AS
CAPACITATE_TREN

FROM TRENURI T, VAGOANE V, RUTE R

WHERE T.ID_TREN = VI.ID_TREN AND T.ID_RUTA = R.ID_RUTA AND T.ID_RUTA =
172400;
```

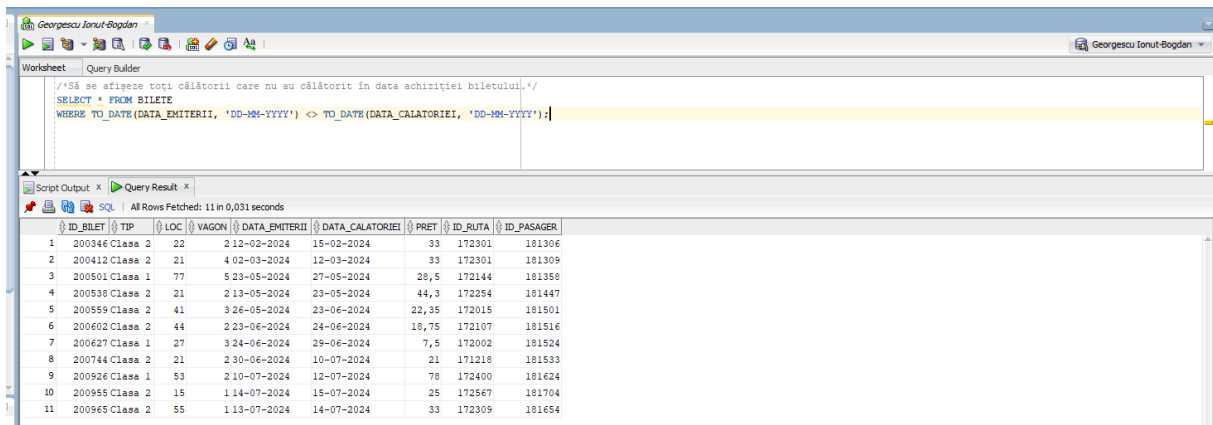


The screenshot shows a SQL Query Builder window with a query that filters for route ID 172400. The results table has 7 columns: TIP_TREN, ID_TREN, PLECARE, SOSIRE, DURATA, NR_VAGOANE, and CAPACITATE_TREN. Two rows are displayed.

TIP_TREN	ID_TREN	PLECARE	SOSIRE	DURATA	NR_VAGOANE	CAPACITATE_TREN
1 INTERREGIO	10063	Bucuresti Nord Brasov	175	2	Automotor	148
2 REGIO-EXPRESS	3002	Bucuresti Nord Brasov	175	5	Clasa 2	80

3. Să se afișeze toți călătorii care nu au călătorit în data achiziției biletului.

```
SELECT * FROM BILETE
WHERE TO_DATE(DATA_EMITERII, 'DD-MM-YYYY') <>
TO_DATE(DATA_CALATORIEI, 'DD-MM-YYYY');
```



The screenshot shows a SQL Query Builder window with a query that filters for tickets where the issue date is not equal to the travel date. The results table has 10 columns: ID_BILET, TIP, LOC, VAGON, DATA_EMITERII, DATA_CALATORIEI, PRET, ID_RUTA, and ID_PASAGER. 11 rows are displayed.

ID_BILET	TIP	LOC	VAGON	DATA_EMITERII	DATA_CALATORIEI	PRET	ID_RUTA	ID_PASAGER
1	200346	Clasa 2	22	212-02-2024	15-02-2024	33	172301	181306
2	200412	Clasa 2	21	4 02-03-2024	12-03-2024	33	172301	181309
3	200501	Clasa 1	77	5 23-05-2024	27-05-2024	28,5	172144	181358
4	200538	Clasa 2	21	213-05-2024	23-05-2024	44,3	172254	181447
5	200559	Clasa 2	41	3 26-05-2024	23-06-2024	22,35	172015	181501
6	200602	Clasa 2	44	2 23-06-2024	24-06-2024	18,75	172107	181516
7	200627	Clasa 1	27	3 24-06-2024	29-06-2024	7,5	172002	181524
8	200744	Clasa 2	21	2 30-06-2024	10-07-2024	21	171218	181533
9	200926	Clasa 1	53	210-07-2024	12-07-2024	78	172400	181624
10	200955	Clasa 2	15	114-07-2024	15-07-2024	25	172567	181704
11	200965	Clasa 2	55	113-07-2024	14-07-2024	33	172309	181654

4. Să se afișeze numele și prenumele precum și adresa de e-mail pentru toți călătorii care au adresa de e-mail de tip '@gmail.com' ordonați după nume.

```
SELECT NUME, PRENUME, EMAIL FROM PASAGERI
WHERE UPPER(EMAIL) LIKE '%@GMAIL.COM'
ORDER BY NUME;
```


Georgescu Ionut-Bogdan

Worksheet Query Builder

```
/*Să se afișeze numele și prenumele precum și adresa de e-mail pentru toți călătorii care au adresa de e-mail de tip '@gmail.com' ordonați după nume.*/
SELECT NUME, PRENUME, EMAIL FROM PASAGERI
WHERE UPPER(EMAIL) LIKE '@GMAIL.COM'
ORDER BY NUME;
```

Script Output x Query Result x

All Rows Fetched: 12 in 0,276 seconds

	NUME	PRENUME	EMAIL
1	Andrescu	Ioana	ioana.andr@gmail.com
2	Erenia	Alina	erenia.alina22@gmail.com
3	Ionescu	Marian	mar.ion@gmail.com
4	Irimia	Alina	i.alina253@gmail.com
5	Marinescu	Georgiana	marinescu2004@gmail.com
6	Matei	Ionut	i.matei117@gmail.com
7	Nica	Octavian	noctavian232@gmail.com
8	Nicolaescu	Teodor	teo.nicolaescu33@gmail.com
9	Nistor	Elena	nistorelena87@gmail.com
10	Raicu	Gabriel	graicu@gmail.com
11	Sandulescu	Georgiana	gsandulescu54@gmail.com
12	Voicu	Andreea	voicuandree54@gmail.com

5. Să se afișeze toți angajații care lucrează pe trenuri de tip ‘Automotor’.

SELECT P.*

FROM PERSONAL_TREN P, TRENURI T, VAGOANE V

WHERE P.ID_TREN = T.ID_TREN AND T.ID_TREN = V.ID_TREN

AND UPPER(TIP_VAGON) = ‘AUTOMOTOR’;

Georgescu Ionut-Bogdan

Worksheet Query Builder

```
/*5. Să se afișeze toți angajații care lucrează pe trenuri de tip 'Automotor'.*/
SELECT P.*
FROM PERSONAL_TREN P, TRENURI T, VAGOANE V
WHERE P.ID_TREN = T.ID_TREN AND T.ID_TREN = V.ID_TREN
AND UPPER(TIP_VAGON) = 'AUTOMOTOR';
```

Script Output x Query Result x

All Rows Fetched: 6 in 0,026 seconds

	ID_ANGAJAT	NUME	PRENUME	FUNCTIE	SALARIU	DATA_ANGAJARII	ID_TREN
1	202913	Dimitrie	Alexandru	Mecanic Locomotiva	6600	18-04-2022	10165
2	202915	Alexandrescu	Gabriel	Controlor Bilete	4730	23-02-2017	10165
3	202926	Badea	Alin	Controlor Bilete	4300	25-02-2017	9457
4	202935	Negut	Constantin	Controlor Bilete	4300	11-01-2015	9417
5	202936	Florescu	Gabriela	Controlor Bilete	4730	13-02-2022	10083
6	202946	Toma	Gabriela	Mecanic Locomotiva	6930	13-11-2022	10083

6. Să se afișeze toate detaliile despre trenurile care nu au repartizați niciun angajat.

SELECT * FROM TRENURI T

WHERE NOT EXISTS (SELECT 1 FROM PERSONAL_TREN P WHERE P.ID_TREN = T.ID_TREN);

Georgescu Ionut-Bogdan

Worksheet Query Builder

```
SELECT * FROM TRENURI T
WHERE NOT EXISTS (SELECT 1 FROM PERSONAL_TREN P WHERE P.ID_TREN = T.ID_TREN);
```

Script Output x Query Result x

All Rows Fetched: 8 in 0,012 seconds

	ID_TREN	TIP_TREN	NR_VAGOANE	NR_PASAGERI	ID_RUTA
1	472	INTERREGIO	11	232	171544
2	531	INTERCITY	6	89	171218
3	7522	REGIO-EXPRESS	4	280	172002
4	9001	REGIO	4	320	172015
5	9205	REGIO-EXPRESS	3	160	171407
6	9434	REGIO	2	87	171509
7	10268	REGIO	2	100	172029
8	10356	REGIO	3	256	172107

7. Să se afișeze stațiile de plecare și sosire, durata, numărul de km precum și tipul rutei astfel: dacă există cel puțin 1 tren pe ruta respectivă atunci să se afișeze 'Ruta Operabila', altfel 'Ruta Neoperabila'.

```
SELECT R.PLECARE, R.SOSIRE, R.NR_KM,
```

```
CASE
```

```
WHEN EXISTS ( SELECT 1 FROM TRENURI T WHERE T.ID_RUTA = R.ID_RUTA )  
THEN 'Ruta Operabila'
```

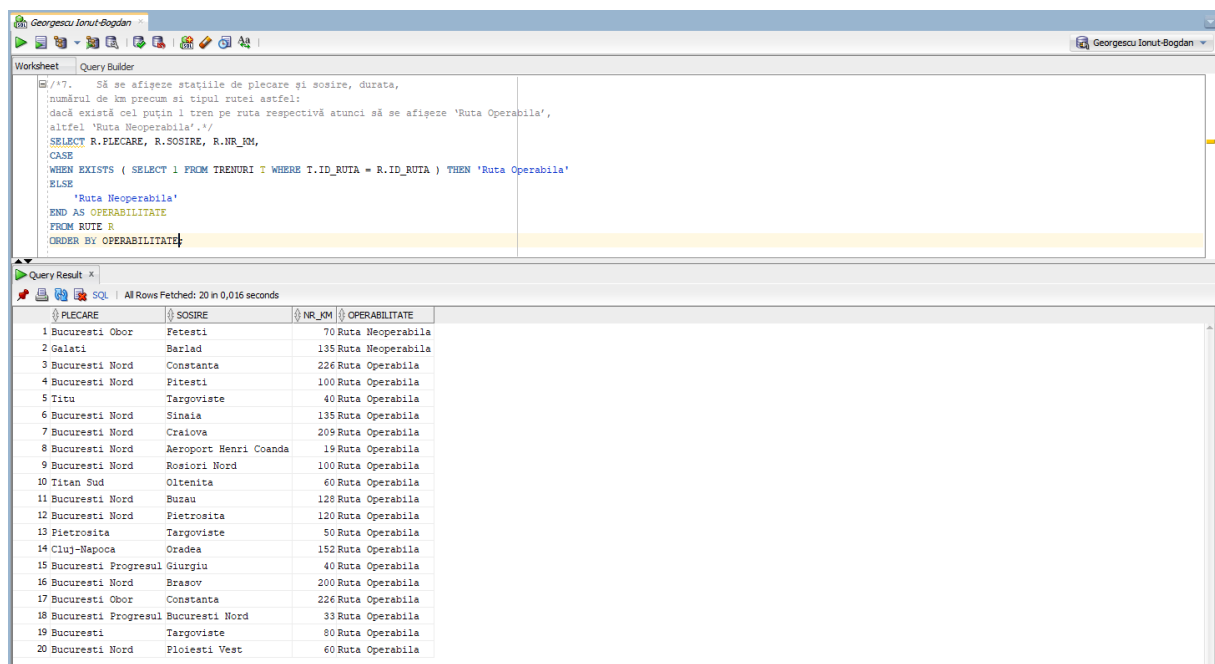
```
ELSE
```

```
'Ruta Neoperabila'
```

```
END AS OPERABILITATE
```

```
FROM RUTE R
```

```
ORDER BY OPERABILITATE;
```



Georgescu Ionut-Bogdan

Worksheet Query Builder

7. Să se afișeze stațiile de plecare și sosire, durata, numărul de km precum și tipul rutei astfel: dacă există cel puțin 1 tren pe ruta respectivă atunci să se afișeze 'Ruta Operabila', altfel 'Ruta Neoperabila'.

```
SELECT R.PLECARE, R.SOSIRE, R.NR_KM,  
CASE  
WHEN EXISTS ( SELECT 1 FROM TRENURI T WHERE T.ID_RUTA = R.ID_RUTA ) THEN 'Ruta Operabila'  
ELSE  
'Ruta Neoperabila'  
END AS OPERABILITATE  
FROM RUTE R  
ORDER BY OPERABILITATE;
```

Query Result

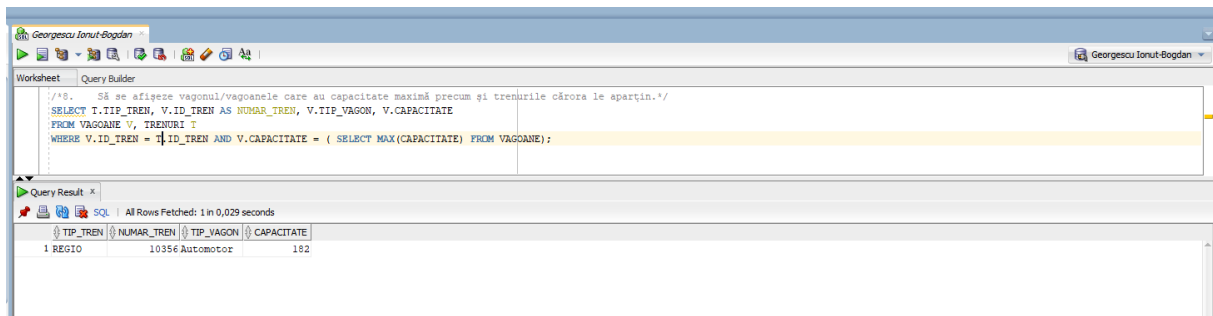
All Rows Fetched: 20 in 0,016 seconds

	PLECARE	SOSIRE	NR_KM	OPERABILITATE
1	Bucuresti Obor	Fetesti	70	Ruta Neoperabila
2	Galati	Barlad	135	Ruta Neoperabila
3	Bucuresti Nord	Constanta	226	Ruta Operabila
4	Bucuresti Nord	Pitesti	100	Ruta Operabila
5	Titu	Targoviste	40	Ruta Operabila
6	Bucuresti Nord	Sinaia	135	Ruta Operabila
7	Bucuresti Nord	Craiova	209	Ruta Operabila
8	Bucuresti Nord	Aeroport Henri Coanda	19	Ruta Operabila
9	Bucuresti Nord	Rosiori Nord	100	Ruta Operabila
10	Titan Sud	Oltenita	60	Ruta Operabila
11	Bucuresti Nord	Buzau	128	Ruta Operabila
12	Bucuresti Nord	Pietrosita	120	Ruta Operabila
13	Pietrosita	Targoviste	50	Ruta Operabila
14	Cluj-Napoca	Oradea	152	Ruta Operabila
15	Bucuresti Progresul	Gurgiu	40	Ruta Operabila
16	Bucuresti Nord	Brasov	200	Ruta Operabila
17	Bucuresti Obor	Constanta	226	Ruta Operabila
18	Bucuresti Progresul	Bucuresti Nord	33	Ruta Operabila
19	Bucuresti	Targoviste	80	Ruta Operabila
20	Bucuresti Nord	Ploiesti Vest	60	Ruta Operabila

8. Să se afișeze vagonul/vagoanele care au capacitate maximă precum și trenurile cărora le aparțin.

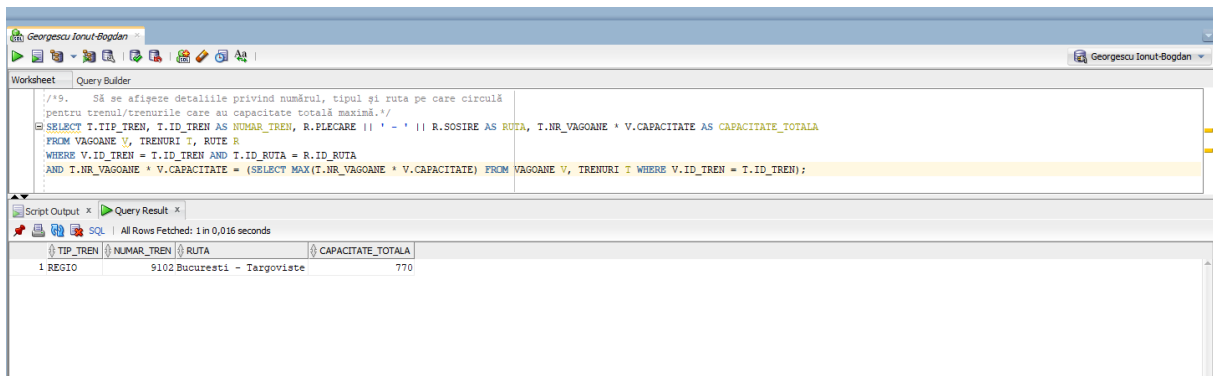
```
SELECT T.TIP_TREN, V.ID_TREN AS NUMAR_TREN, V.TIP_VAGON, V.CAPACITATE  
FROM VAGOANE V, TRENURI T
```

WHERE V.ID_TREN = T.ID_TREN AND V.CAPACITATE = (SELECT
MAX(CAPACITATE) FROM VAGOANE);



9. Să se afişeze detaliile privind numărul, tipul şi ruta pe care circulă pentru trenul/trenurile care au capacitate totală maximă.

SELECT T.TIP_TREN, T.ID_TREN AS NUMAR_TREN, R.PLECARE || ' - ' || R.SOSIRE
AS RUTA, T.NR_VAGOANE * V.CAPACITATE AS CAPACITATE_TOTALA
FROM VAGOANE V, TRENURI T, RUTE R
WHERE V.ID_TREN = T.ID_TREN AND T.ID_RUTA = R.ID_RUTA
AND T.NR_VAGOANE * V.CAPACITATE = (SELECT MAX(T.NR_VAGOANE *
V.CAPACITATE) FROM VAGOANE V, TRENURI T WHERE V.ID_TREN = T.ID_TREN);



10. Să se afişeze toate trenurile care au o capacitate totală mai mare decât capacitatea totală medie a trenurilor înregistrate.

SELECT T.TIP_TREN, T.ID_TREN, R.PLECARE || ' - ' || R.SOSIRE AS RUTA,
T.NR_VAGOANE * V.CAPACITATE AS CAPACITATE_TOTALA
FROM VAGOANE V, TRENURI T, RUTE R
WHERE V.ID_TREN = T.ID_TREN AND T.ID_RUTA = R.ID_RUTA
AND T.NR_VAGOANE * V.CAPACITATE >= (SELECT AVG(T.NR_VAGOANE *
V.CAPACITATE) FROM VAGOANE V, TRENURI T WHERE V.ID_TREN = T.ID_TREN)
ORDER BY T.NR_VAGOANE * V.CAPACITATE DESC;

The screenshot shows a SQL query in the Query Builder window. The query selects train details and capacity from two tables, VAGOANE and TRENURI, joined on their respective IDs. The results are ordered by capacity in descending order.

TIP_TREN	ID_TREN	Ruta	Capacitate_Totala
1 REGIO	9102	Bucuresti - Targoviste	770
2 REGIO	9102	Bucuresti - Targoviste	770
3 INTERCITY	551	Bucuresti Nord - Buzau	720
4 INTERREGIO	472	Bucuresti Nord - Sinaia	583
5 REGIO	10356	Titan Sud - Oltenita	546
6 REGIO	8001	Bucuresti Obor - Constanta	450
7 REGIO-EXPRESS	9900	Bucuresti Nord - Craiova	400
8 REGIO-EXPRESS	3002	Bucuresti Nord - Brasov	400
9 INTERCITY	531	Bucuresti Nord - Ploiesti Vest	390

11. Să se afișeze toate trenurile care au o durată a călătoriei de 1 oră și 20 de minute mai mare puțin cele care nu pleacă din stația ‘Bucuresti Nord’.

```
SELECT T.TIP_TREN, T.ID_TREN, R.PLECARE || ' - ' || R.SOSIRE AS RUTA, R.DURATA
FROM TRENURI T, RUTE R
WHERE T.ID_RUTA = R.ID_RUTA AND R.DURATA >= 80
MINUS
SELECT T.TIP_TREN, T.ID_TREN, R.PLECARE || ' - ' || R.SOSIRE AS RUTA, R.DURATA
FROM TRENURI T, RUTE R
WHERE T.ID_RUTA = R.ID_RUTA AND UPPER(R.PLECARE) <> 'BUCURESTI NORD';
```

The screenshot shows a SQL query in the Query Builder window. The query uses a MINUS operator to filter out trains that do not depart from 'Bucuresti Nord'. The results show trains with a duration of 80 minutes or more.

TIP_TREN	ID_TREN	Ruta	Durata
1 INTERREGIO	1583	Bucuresti Nord - Constanta	168
2 REGIO-EXPRESS	9205	Bucuresti Nord - Pitesti	160
3 INTERREGIO	472	Bucuresti Nord - Sinaia	105
4 REGIO-EXPRESS	9900	Bucuresti Nord - Craiova	282
5 REGIO	9001	Bucuresti Nord - Rosiori Nord	159
6 REGIO	10165	Bucuresti Nord - Pietrosita	175
7 INTERCITY	551	Bucuresti Nord - Buzau	110
8 INTERREGIO	10083	Bucuresti Nord - Brasov	175
9 REGIO-EXPRESS	3002	Bucuresti Nord - Brasov	175

12. Să se afișeze angajații care au fost angajați înainte de anul 2007.

```
SELECT * FROM PERSONAL_TREN
WHERE EXTRACT(YEAR FROM DATA_ANGAJARII) < 2007
ORDER BY DATA_ANGAJARII;
```

Georgescu Ionut-Bogdan

Worksheet Query Builder

```

WHERE T.ID_RUTA = R.ID_RUTA AND UPPER(R.PLECARE) <> 'BUCURESTI NORD';

/*12. Să se afișeze angajații care au fost angajați înainte de anul 2007.*/

SELECT * FROM PERSONAL_TREN
WHERE EXTRACT(YEAR FROM DATA_ANGAJARII) < 2007
ORDER BY DATA_ANGAJARII;

```

Script Output x Query Result x

All Rows Fetched: 8 in 0,018 seconds

ID_ANGAJAT	NUME	PRENUME	FUNCTIE	SALARIU	DATA_ANGAJARII	ID_TREN
1	202901	Marinescu	Ion	Controlor Bilete	6050 12-09-1990	9102
2	202907	Filipescu	Stefania	Controlor Bilete	4940 21-01-1998	9900
3	202911	Ungureanu	Valeria	Controlor Bilete	5610 24-05-1999	1634
4	202903	Ionescu	Marinela	Controlor Bilete	5610 23-09-1999	1583
5	202902	Vasile	Cristian	Mecanic Locomotiva	8250 22-08-2000	9102
6	202912	Dumitrescu	Eugen	Controlor Bilete	5555 15-06-2001	367
7	202906	Nicolae	Marian	Mecanic Locomotiva	7590 19-02-2003	1583
8	202908	Tudor	Andrei	Mecanic Locomotiva	7700 13-04-2004	9900

13. Să se afișeze angajații trenurilor precum și tipul trenului, numărul și ruta pentru trenurile care operează pe rute mai lungi de 200 de km.

```

SELECT T.TIP_TREN, T.ID_TREN AS NUMAR_TREN, R.PLECARE || ' - ' || R.SOSIRE
AS RUTA

FROM TRENURI T, RUTE R

WHERE T.ID_RUTA = R.ID_RUTA AND R.NR_KM > 200;

```

Georgescu Ionut-Bogdan

Worksheet Query Builder

```

ORDER BY DATA_ANGAJARII;

/*13. Să se afișeze angajații trenurilor precum și tipul trenului, numărul și
ruta pentru trenurile care operează pe rute mai lungi de 200 de km.*/

SELECT T.TIP_TREN, T.ID_TREN AS NUMAR_TREN, R.PLECARE || ' - ' || R.SOSIRE AS RUTA
FROM TRENURI T, RUTE R
WHERE T.ID_RUTA = R.ID_RUTA AND R.NR_KM > 200;

```

Script Output x Query Result x

All Rows Fetched: 3 in 0,211 seconds

TIP_TREN	NUMAR_TREN	RUTA
1 INTERREGIO	1583	Bucuresti Nord - Constanta
2 REGIO-EXPRESS	9900	Bucuresti Nord - Craiova
3 REGIO	8001	Bucuresti Obor - Constanta

14. Să se afișeze trenurile și numărul de controlori de bilete alocați.

```

SELECT T.TIP_TREN, T.ID_TREN AS NUMAR_TREN, R.PLECARE || ' - ' || R.SOSIRE
AS RUTA, COUNT(P.ID_ANGAJAT) AS NUMAR_CONTROLORI

FROM PERSONAL_TREN P, TRENURI T, RUTE R

WHERE P.ID_TREN = T.ID_TREN AND T.ID_RUTA = R.ID_RUTA AND
UPPER(P.FUNCTIE) = 'CONTROLOR BILETE'

GROUP BY T.TIP_TREN, T.ID_TREN, R.PLECARE, R.SOSIRE;

```

Georgescu Ionut-Bogdan

Worksheet Query Builder

```

/*14. Să se afișeze trenurile și numărul de controlori de bilete alocați.*/
SELECT T.TIP_TREN, T.ID_TREN AS NUMAR_TREN, R.PLECARE || ' - ' || R.SOSIRE AS RUTA, COUNT(P.ID_ANGAJAT) AS NUMAR_CONTROLORI
FROM PERSONAL_TREN P, TRENURI T, RUTE R
WHERE P.ID_TREN = T.ID_TREN AND T.ID_RUTA = R.ID_RUTA AND UPPER(P.FUNCTIE) = 'CONTROLOR BILETE'
GROUP BY T.TIP_TREN, T.ID_TREN, R.PLECARE, R.SOSIRE;

```

Script Output x Query Result x

SQL All Rows Fetched: 12 in 0,043 seconds

TIP_TREN	NUMAR_TREN	RUTA	NUMAR_CONTROLORI
1 REGIO	9102	Bucuresti - Targoviste	1
2 INTERREGIO	1634	Bucuresti Nord - Ploiesti Vest	1
3 INTERREGIO	1583	Bucuresti Nord - Constanta	1
4 REGIO-EXPRESS	9900	Bucuresti Nord - Craiova	1
5 INTERCITY	551	Bucuresti Nord - Buzau	1
6 REGIO	10165	Bucuresti Nord - Pietrosita	1
7 INTERREGIO	367	Cluj-Napoca - Oradea	2
8 REGIO	9417	Bucuresti Progresul - Giurgiu	1
9 REGIO-EXPRESS	3002	Bucuresti Nord - Brasov	1
10 INTERREGIO	10083	Bucuresti Nord - Brasov	1
11 REGIO	8001	Bucuresti Obor - Constanta	1
12 REGIO-EXPRESS	9457	Bucuresti Progresul - Bucuresti Nord	1

15. Să se afișeze detalii despre trenurile și rutele pe care circulă pentru toate trenurile care pleacă sau sosesc in Targoviste.

SELECT T.TIP_TREN, T.ID_TREN AS NUMAR_TREN, R.PLECARE || ' - ' || R.SOSIRE
AS RUTA

FROM TRENURI T, RUTE R

WHERE T.ID_RUTA = R.ID_RUTA AND (UPPER(R.PLECARE) = 'TARGOVISTE' OR
UPPER(R.SOSIRE) = 'TARGOVISTE');

Georgescu Ionut-Bogdan

Worksheet Query Builder

```

GROUP BY T.TIP_TREN, T.ID_TREN, R.PLECARE, R.SOSIRE;
/*15. Să se afișeze detalii despre trenurile și rutele pe care circulă
pentru toate trenurile care pleacă sau sosesc in Targoviste.*/
SELECT T.TIP_TREN, T.ID_TREN AS NUMAR_TREN, R.PLECARE || ' - ' || R.SOSIRE AS RUTA
FROM TRENURI T, RUTE R
WHERE T.ID_RUTA = R.ID_RUTA AND ( UPPER(R.PLECARE) = 'TARGOVISTE' OR UPPER(R.SOSIRE) = 'TARGOVISTE' );
/*16. Să se afișeze detalii despre mecanicii de locomotivă precum și o
coloană bonus astfel: pentru parcurgerea unei rute de până în 200 de km bonusul este de 10%, pentru rutele
intre 200 și 300 de km bonusul este 20%, iar pentru rutele parcurse de peste 300km bonusul este 30%.*/
/*17. Să se afișeze toate trenurile de tip Automotor și care pleacă din stația "Bucuresti Nord".*/
/*18. Să se afișeze toți salariații care au salariul mai mare decât salariul mediu al unui controlor de bilete.*/
/*19. Să se afișeze controlorii de bilete precum și coloana bonus astfel:

```

Script Output x Query Result x

SQL All Rows Fetched: 3 in 0,013 seconds

TIP_TREN	NUMAR_TREN	RUTA
1 REGIO	9102	Bucuresti - Targoviste
2 REGIO	9434	Titu - Targoviste
3 REGIO	10268	Pietrosita - Targoviste

16. Să se afișeze detalii despre mecanicii de locomotivă precum și o coloană bonus astfel:
pentru parcurgerea unei rute de până în 200 de km bonusul este de 10%, pentru rutele
intre 200 și 300 de km bonusul este 20%, iar pentru rutele parcurse de peste 300km
bonusul este 30%.

SELECT P.NUME, P.PRENUME, P.SALARIUL, P.ID_TREN,

CASE

WHEN R.NR_KM < 200 THEN 0.1 * SALARIUL

```

WHEN R.NR_KM BETWEEN 200 AND 300 THEN 0.2 * SALARIUL

WHEN R.NR_KM > 300 THEN 0.3 * SALARIUL

ELSE 0

END AS BONUS

FROM PERSONAL_TREN P, TRENURI T, RUTE R

WHERE P.ID_TREN = T.ID_TREN AND T.ID_RUTA = R.ID_RUTA;

```

Georgescu Ionut-Bogdan

Worksheet Query Builder

```

--16. Să se afișeze detalii despre mecanicii de locomotivă precum și o
--coloană bonus astfel: pentru parcurgerea unei rute de până în 200 de km bonusul este de 10%, pentru rutele
--între 200 și 300 de km bonusul este 20%, iar pentru rutele parcurse de peste 300km bonusul este 30%.*/

SELECT P.NUME, P.PRENUME, P.SALARIUL, P.ID_TREN,
CASE
WHEN R.NR_KM < 200 THEN 0.1 * SALARIUL
WHEN R.NR_KM BETWEEN 200 AND 300 THEN 0.2 * SALARIUL
WHEN R.NR_KM > 300 THEN 0.3 * SALARIUL
ELSE 0
END AS BONUS
FROM PERSONAL_TREN P, TRENURI T, RUTE R
WHERE P.ID_TREN = T.ID_TREN AND T.ID_RUTA = R.ID_RUTA;

```

Script Output x Query Result x

All Rows Fetched: 20 in 0,128 seconds

	NUME	PRENUME	SALARIUL	ID_TREN	BONUS
1	Marinescu	Ion	6050	9102	605
2	Vasile	Cristian	8250	9102	825
3	Simionescu	Aurelia	7370	1634	737
4	Ungureanu	Valeria	5610	1634	561
5	Ionescu	Marinela	5610	1583	1122
6	Nicolae	Marian	7590	1583	1518
7	Tudor	Andrei	7700	9900	1540
8	Filipescu	Stefania	4840	9900	968
9	Gheorghe	Alina	4290	551	429
10	Alexandrescu	Gabriel	4730	10165	473
11	Dimitrie	Alexandru	6600	10165	660
12	Marinescu	Victor	4950	367	495
13	Radu	Elena	7260	367	726
14	Dumitrescu	Eugen	5555	367	555,5
15	Negut	Constantin	4300	9417	430
16	Andrescu	Gabriel	4730	3002	946
17	Toma	Gabriela	6930	10083	1386
18	Florescu	Gabriela	4730	10083	946
19	Toma	Matei	4730	8001	946
20	Badea	Alin	4300	9457	430

17. Să se afișeze toate trenurile de tip Automotor și care pleacă din stația “Bucuresti Nord”.

```

SELECT T.TIP_TREN, T.ID_TREN AS NUMAR_TREN, R.PLECARE || ' - ' || R.SOSIRE
AS RUTA

FROM TRENURI T, RUTE R, VAGOANE V

WHERE T.ID_TREN = V.ID_TREN AND T.ID_RUTA = R.ID_RUTA AND
UPPER(V.TIP_VAGON) = 'AUTOMOTOR' AND UPPER(R.PLECARE) = 'BUCURESTI
NORD'

UNION

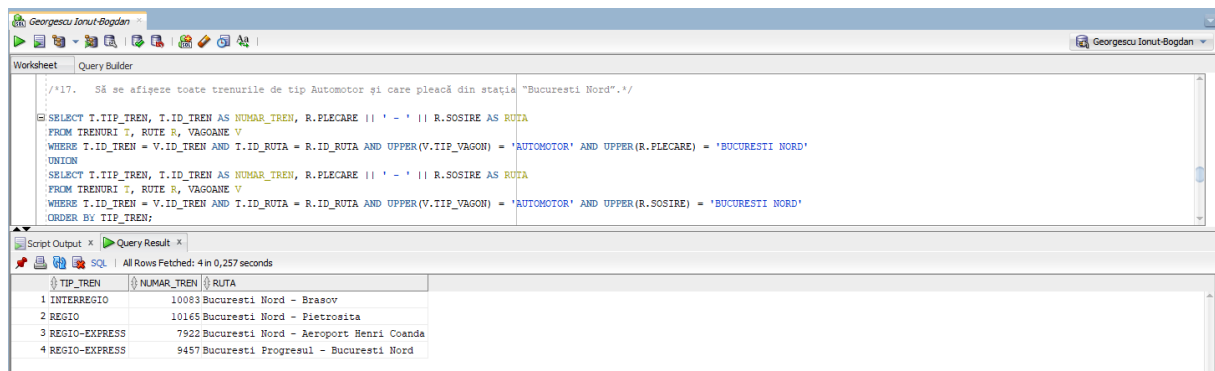
SELECT T.TIP_TREN, T.ID_TREN AS NUMAR_TREN, R.PLECARE || ' - ' || R.SOSIRE
AS RUTA

FROM TRENURI T, RUTE R, VAGOANE V

WHERE T.ID_TREN = V.ID_TREN AND T.ID_RUTA = R.ID_RUTA AND
UPPER(V.TIP_VAGON) = 'AUTOMOTOR' AND UPPER(R.SOSIRE) = 'BUCURESTI
NORD'

```

ORDER BY TIP_TREN;



The screenshot shows a SQL Developer window with a query in the 'Query Builder' tab. The query is designed to list train types and routes from 'Bucuresti Nord'. The results are displayed in a table with 4 rows and 3 columns: TIP_TREN, NUMAR_TREN, and RUTA.

TIP_TREN	NUMAR_TREN	RUTA
1 INTERREGIO	10083	Bucuresti Nord - Brasov
2 REGIO	10165	Bucuresti Nord - Pietrosita
3 REGIO-EXPRESS	7922	Bucuresti Nord - Aeroport Henri Coanda
4 REGIO-EXPRESS	9457	Bucuresti Progresul - Bucuresti Nord

18. Să se afișeze toți salariații care au salariul mai mare decât salariul mediu al unui controlor de bilete.

SELECT * FROM PERSONAL_TREN

WHERE SALARIUL > (SELECT AVG(SALARIUL) FROM PERSONAL_TREN)

ORDER BY SALARIUL DESC;

Georgescu Ionut-Bogdan

Worksheet

Query Builder

WHERE T.ID_TREN = V.ID_TREN AND T.ID_RUTA = R.ID_RUTA AND UPPER(V.TIF_VAGON) = 'AUTOMOTOR' AND UPPER(R.SOSTRE) = 'BUCURESTI NORD'

ORDER BY TIF_TREN;

/*18. Să se afișeze toți salariații care au salariul mai mare decât salariul mediu al unui controlor de bilete.*/

SELECT * FROM PERSONAL_TREN

WHERE SALARIUL > (SELECT AVG(SALARIUL) FROM PERSONAL_TREN)

ORDER BY SALARIUL DESC;

Script Output x

Query Result x

SQL

All Rows Fetched: 8 in 0,015 seconds

ID_ANGAJAT	NUME	PRENUME	FUNCTIE	SALARIUL	DATA_ANGAJARII	ID_TREN	
1	202902	Vasile	Cristian	Mecanic Locomotiva	8250	22-08-2000	9102
2	202908	Tudor	Andrei	Mecanic Locomotiva	7700	13-04-2004	9900
3	202906	Nicolae	Marian	Mecanic Locomotiva	7590	19-02-2003	1583
4	202914	Simionescu	Aurelia	Mecanic Locomotiva	7370	17-03-2008	1634
5	202910	Radu	Elena	Mecanic Locomotiva	7260	19-07-2014	367
6	202946	Toma	Gabriela	Mecanic Locomotive	6930	13-11-2022	10083
7	202913	Dimitrie	Alexandru	Mecanic Locomotive	6600	18-04-2022	10165
8	202901	Marinescu	Ion	Controlor Bilete	6050	12-09-1990	9102

19. Să se afișeze controlorii de bilete precum și coloana bonus astfel:

dacă lucrează pe un tren REGIO bonusul este 10%, dacă lucrează pe un tren REGIO-EXPRESS bonusul este 15%, dacă lucrează pe un tren INTERREGIO bonusul este 20%, iar dacă lucrează pe un tren INTERCITY bonusul este 25%.

SELECT P.*,

CASE

WHEN UPPER(T.TIP_TREN) = 'REGIO' THEN 0.1

WHEN UPPER(T.TIP_TREN) = 'REGIO-EXPRESS' THEN 0.15


```

WHEN UPPER(T.TIP_TREN) = 'INTERREGIO' THEN 0.2
WHEN UPPER(T.TIP_TREN) = 'INTERCITY' THEN 0.25
ELSE 0
END AS DISCOUNT
FROM PERSONAL_TREN P, TRENURI T
WHERE P.ID_TREN = T.ID_TREN AND UPPER(P.FUNCTIE) = 'CONTROLOR BILETE';

```

Georgescu Ionut-Bogdan

Georgescu Ionut-Bogdan

Worksheet

Query Builder

```
/*19. Să se afișeze controlorii de bilete precum și coloana bonus astfel:
dacă lucrează pe un tren REGIO bonusul este 10%, dacă lucrează pe un tren
REGIO-EXPRESS bonusul este 15%, dacă lucrează pe un tren INTERREGIO bonusul este 20%,
iar dacă lucrează pe un tren INTERCITY bonusul este 25%*/

SELECT P.*,
CASE
WHEN UPPER(T.TIP_TREN) = 'REGIO' THEN 0.1
WHEN UPPER(T.TIP_TREN) = 'REGIO-EXPRESS' THEN 0.15
WHEN UPPER(T.TIP_TREN) = 'INTERREGIO' THEN 0.2
WHEN UPPER(T.TIP_TREN) = 'INTERCITY' THEN 0.25
ELSE 0
END AS DISCOUNT
FROM PERSONAL_TREN P, TRENURI T
WHERE P.ID_TREN = T.ID_TREN AND UPPER(P.FUNCTIE) = 'CONTROLOR BILETE';
```

Script Output

Query Result

All Rows Fetched: 13 in 0,015 seconds

	ID_ANGAJAT	NUME	PRENUME	FUNCTIE	SALARUL	DATA_ANGAJARII	ID_TREN	DISCOUNT
1	202901	Marinescu	Ion	Controlor Bilete	6050	12-09-1990	9102	0,1
2	202903	Ionescu	Marinela	Controlor Bilete	5610	23-09-1999	1583	0,2
3	202915	Alexandrescu	Gabriel	Controlor Bilete	4730	23-02-2017	10165	0,1
4	202904	Gheorghe	Alina	Controlor Bilete	4290	24-10-2019	551	0,25
5	202907	Filipescu	Stefania	Controlor Bilete	4840	21-01-1998	9900	0,15
6	202909	Marinescu	Victor	Controlor Bilete	4950	11-01-2007	367	0,2
7	202912	Dumitrescu	Eugen	Controlor Bilete	5555	15-06-2001	367	0,2
8	202911	Ungureanu	Valeria	Controlor Bilete	5610	24-05-1999	1634	0,2
9	202925	Toma	Matei	Controlor Bilete	4730	23-02-2018	8001	0,1
10	202916	Andrescu	Gabriel	Controlor Bilete	4730	23-04-2017	3002	0,15
11	202926	Badea	Alin	Controlor Bilete	4300	25-02-2017	9457	0,15
12	202935	Negut	Constantin	Controlor Bilete	4300	11-01-2015	9417	0,1
13	202936	Florescu	Gabriela	Controlor Bilete	4730	13-02-2022	10083	0,2

20. Să se afișeze numărul de trenuri specific fiecărui tip.

```

SELECT T.TIP_TREN, COUNT(T.ID_TREN)
FROM TRENURI T
GROUP BY T.TIP_TREN;

```

The screenshot shows a SQL query in the Query Builder window. The query is as follows:

```

/*20. Să se afișeze numărul de trenuri specific fiecărui tip.*/

SELECT T.TIP_TREN, COUNT(T.ID_TREN)
FROM TRENURI T
GROUP BY T.TIP_TREN;

```

The results are displayed in the Query Result window, showing 4 rows of data. The columns are: TIP_TREN and COUNT(T.ID_TREN).

TIP_TREN	COUNT(T.ID_TREN)
1 REGIO	8
2 INTERREGIO	5
3 INTERCITY	2
4 REGIO-EXPRESS	5

21. Să se afișeze biletele cumpărate pe rutele operate cu trenuri de tip Automotor.

```

SELECT B.* FROM BILETE B, RUTE R, TRENURI T, VAGOANE V
WHERE
B.ID_RUTA = R.ID_RUTA AND R.ID_RUTA = T.ID_RUTA AND T.ID_TREN =
V.ID_TREN AND UPPER(V.TIP_VAGON) = 'AUTOMOTOR';

```

Georgescu Ionut-Bogdan

Worksheet Query Builder

```
/*21. Să se afișeze biletele cumpărate pe rute operate cu trenuri de tip Automotor.*/
SELECT B.* FROM BILETE B, RUTE R, TRENURI T, VAGOANE V
WHERE
B.ID_RUTA = R.ID_RUTA AND R.ID_RUTA = T.ID_RUTA AND T.ID_TREN = V.ID_TREN AND UPPER(V.TIP_VAGON) = 'AUTOMOTOR';
```

Query Result x

All Rows Fetched: 5 in 0,186 seconds

ID_BILET	TIP	LOC	VAGON	DATA_EMITERII	DATA_CALATORIEI	PRET	ID_RUTA	ID_PASAGER
200501	Clasa 1	77	123	05-2024	27-05-2024	28,5	172144	181358
200602	Clasa 2	44	223	06-2024	24-06-2024	18,75	172107	181516
200627	Clasa 1	27	324	06-2024	29-06-2024	7,5	172002	181524
200926	Clasa 1	53	210	07-2024	12-07-2024	78	172400	181624
200965	Clasa 2	55	113	07-2024	14-07-2024	33	172309	181654

22. Să se afișeze toate rutele disponibile indiferent dacă există pasageri care călătoresc pe aceste rute sau nu.

```
SELECT R.ID_RUTA, R.DURATA, R.NR_KM, R.PLECARE, R.SOSIRE ,
COUNT(B.ID_BILET) AS NUMAR_BILETE_VANDUTE
FROM RUTE R, BILETE B
WHERE B.ID_RUTA (+) = R.ID_RUTA
GROUP BY R.ID_RUTA, R.DURATA, R.NR_KM, R.PLECARE, R.SOSIRE;
```

Georgescu Ionut-Bogdan

Worksheet Query Builder

```
/*22. Să se afișeze toate rutele disponibile indiferent dacă călătoresc pasageri pe aceste rute sau nu.*/
SELECT R.ID_RUTA, R.DURATA, R.NR_KM, R.PLECARE, R.SOSIRE , COUNT(B.ID_BILET) AS NUMAR_BILETE_VANDUTE
FROM RUTE R, BILETE B
WHERE B.ID_RUTA (+) = R.ID_RUTA
GROUP BY R.ID_RUTA, R.DURATA, R.NR_KM, R.PLECARE, R.SOSIRE;
```

Query Result x

All Rows Fetched: 20 in 0,009 seconds

ID_RUTA	DURATA	NR_KM	PLECARE	SOSIRE	NUMAR_BILETE_VANDUTE
171011	105	60	Bucuresti	Targoviste	1
171218	38	60	Bucuresti Nord	Ploiesti Vest	2
171324	169	226	Bucuresti Nord	Constanta	2
171407	160	100	Bucuresti Nord	Pitesti	1
171509	45	40	Titu	Targoviste	0
171544	105	135	Bucuresti Nord	Sinaia	1
171903	282	209	Bucuresti Nord	Craiova	1
172002	21	19	Bucuresti Nord	Aeroport Henri Coanda	1
172015	159	100	Bucuresti Nord	Rosiori Nord	1
172029	58	50	Pietrosita	Targoviste	0
172107	100	60	Titan Sud	Oltenita	1
172144	175	120	Bucuresti Nord	Pietrosita	1
172254	110	128	Bucuresti Nord	Buzau	1
172301	122	152	Cluj-Napoca	Oradea	2
172309	65	40	Bucuresti Progresul	Giurgiu	1
172346	158	135	Galati	Barlad	0
172400	175	200	Bucuresti Nord	Brasov	1
172458	205	226	Bucuresti Obor	Constanta	1
172567	105	70	Bucuresti Obor	Pitesti	2
172603	60	33	Bucuresti Progresul	Bucuresti Nord	0

23. Să se afișeze capacitatea totală maximă per tip de tren.

```
SELECT T.TIP_TREN, MAX(V.CAPACITATE * T.NR_VAGOANE) AS
CAPACITATEA_TOTALA_MAXIMA
FROM TRENURI T, VAGOANE V
WHERE T.ID_TREN = V.ID_TREN
GROUP BY T.TIP_TREN;
```

Georgescu Ionut-Bogdan

Worksheet Query Builder

/*23. Să se afișeze capacitatea totala maxima pe rang de tren.*/

```
SELECT T.TIP_TREN, MAX(V.CAPACITATE * T.NR_VAGOANE) AS CAPACITATEA_TOTALA_MAXIMA
FROM TRENURI T, VAGOANE V
WHERE T.ID_TREN = V.ID_TREN
GROUP BY T.TIP_TREN;
```

Query Result X

All Rows Fetched: 4 in 0,019 seconds

TIP_TREN	CAPACITATEA_TOTALA_MAXIMA
1 INTERREGIO	563
2 INTERCITY	720
3 REGIO-EXPRESS	400
4 REGIO	770

24. Să se afișeze dacă există rute neoperabile cu plecare din Bucuresti.

```
SELECT R.PLECARE || ' - ' || R.SOSIRE AS RUTA
FROM RUTE R
WHERE NOT EXISTS (SELECT 1 FROM TRENURI T WHERE T.ID_RUTA =
R.ID_RUTA);
```

Georgescu Ionut-Bogdan

Worksheet Query Builder

/*24. Să se afișeze dacă există rute neoperabile cu plecare din Bucuresti.*/

```
SELECT R.PLECARE || ' - ' || R.SOSIRE AS RUTA
FROM RUTE R
WHERE NOT EXISTS (SELECT 1 FROM TRENURI T WHERE T.ID_RUTA = R.ID_RUTA);
```

Query Result X

All Rows Fetched: 2 in 0,019 seconds

RUTA
1 Galati - Barlad
2 Bucuresti Obor - Fetesti

25. Să se afișeze coloana DISCOUNT, 10% din pret, pentru biletele online achizitionate la trenurile INTERCITY.

```
SELECT B.*,
CASE
WHEN UPPER(P.TIP_BILET) = 'ONLINE' THEN 0.1 * B.PRET
ELSE 0
END AS DISCOUNT
FROM PASAGERI P, BILETE B, RUTE R, TRENURI T
WHERE P.ID_PASAGER = B.ID_PASAGER AND B.ID_RUTA = R.ID_RUTA AND
R.ID_RUTA = T.ID_RUTA AND UPPER(T.TIP_TREN) = 'INTERCITY';
```

Georgescu Ionut-Bogdan

Worksheet Query Builder

```

--125. Sa se afisere coloana DISCOUNT, 10% din pret, pentru biletele online achizitionate la trenurile INTERCITY.V
SELECT B.*,
CASE
WHEN UPPER(P.TIP_BILET) = 'ONLINE' THEN 0.1 * B.PRET
ELSE 0
END AS DISCOUNT
FROM PASAGERI P, BILETE B, RUTE R, TRENURI T
WHERE P.ID_PASAGER = B.ID_PASAGER AND B.ID_RUTA = R.ID_RUTA AND R.ID_RUTA = T.ID_RUTA AND UPPER(T.TIP_TREN) = 'INTERCITY';

```

Query Result x

All Rows Fetched: 3 in 0,025 seconds

	ID_BILET	TIP	LOC	VAGON	DATA_EMITERII	DATA_CALATORIEI	PRET	ID_RUTA	ID_PASAGER	DISCOUNT
1	200538	Clasa 2	21		213-05-2024	23-05-2024	44,3	172254	181447	0
2	200629	Clasa 2	25		130-06-2024	30-06-2024	21	171218	181529	2,1
3	200744	Clasa 2	21		230-06-2024	10-07-2024	21	171218	181533	0

Gestionarea altor obiecte – sinonime, indexi, views

Creare, exemplu utilizare, ștergere

- VIEWS

CREATE VIEW TRENURI_V

AS SELECT * FROM TRENURI;

SELECT * FROM TRENURI_V;

Georgescu Ionut-Bogdan

Worksheet Query Builder

```

CREATE VIEW TRENURI_V
AS SELECT * FROM TRENURI;

SELECT * FROM TRENURI_V;

```

Script Output x Query Result x

Task completed in 0,116 seconds

Commit complete.

View TRENURI_V created.

UPDATE TRENURI_V

SET NR_PASAGERI = NR_PASAGERI + 15

WHERE ID_TREN = 9205;

Georgescu Ionut-Bogdan

Worksheet Query Builder

```

UPDATE TRENURI_V
SET NR_PASAGERI = NR_PASAGERI + 15
WHERE ID_TREN = 9205;

```

Script Output x Query Result x

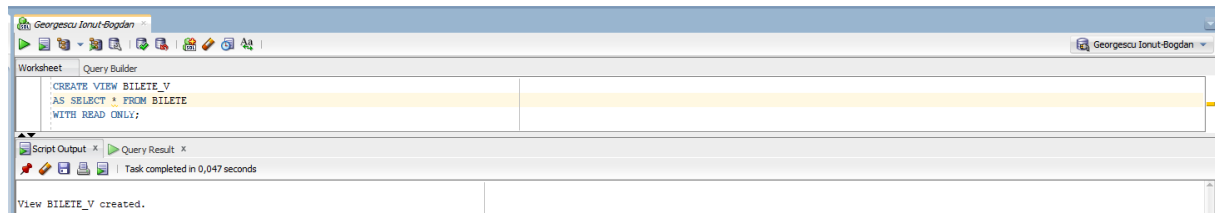
Task completed in 0,356 seconds

1 row updated.

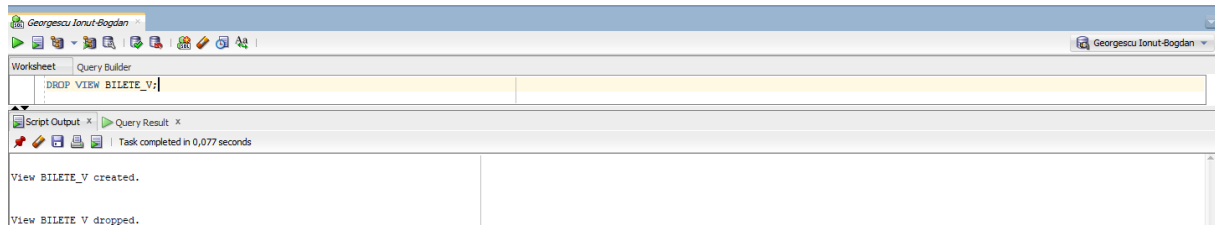
CREATE VIEW BILETE_V

AS SELECT * FROM BILETE

WITH READ ONLY;

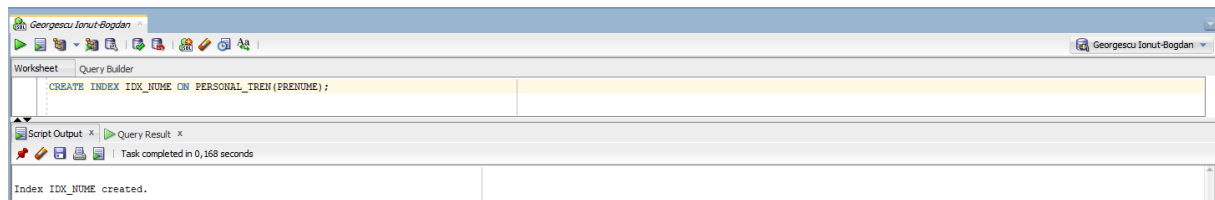


DROP VIEW BILETE_V;

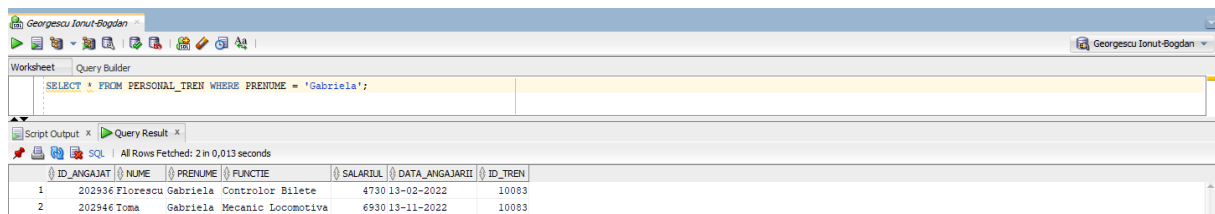


- **INDECȘI**

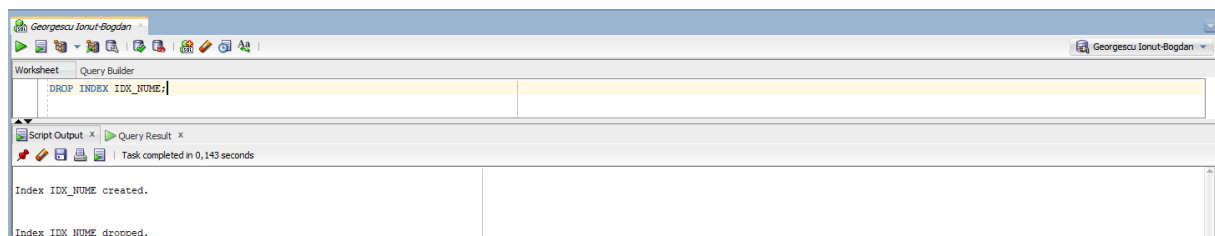
CREATE INDEX IDX_NUME ON PERSONAL_TREN(PRENUME);



SELECT * FROM PERSONAL_TREN WHERE PRENUME = 'Gabriela';



DROP INDEX IDX_NUME;



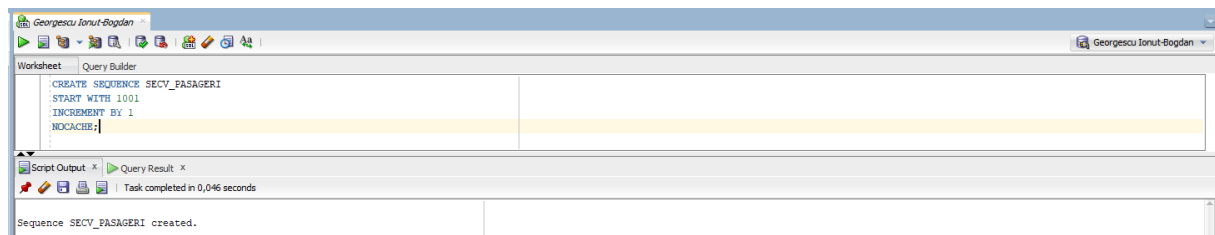
- **SECVENȚE**

CREATE SEQUENCE SECV_PASAGERI

START WITH 1001

INCREMENT BY 1

NOCACHE;

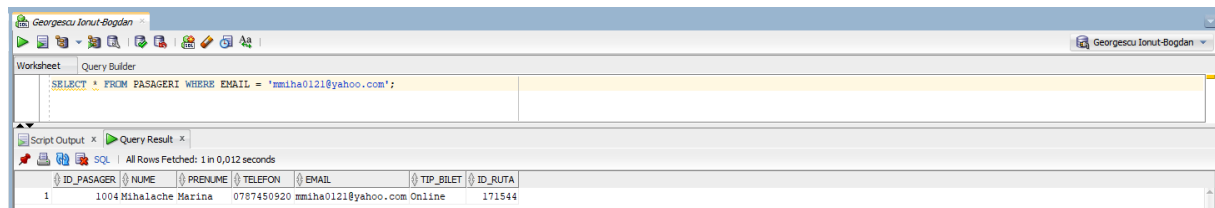


INSERT INTO PASAGERI

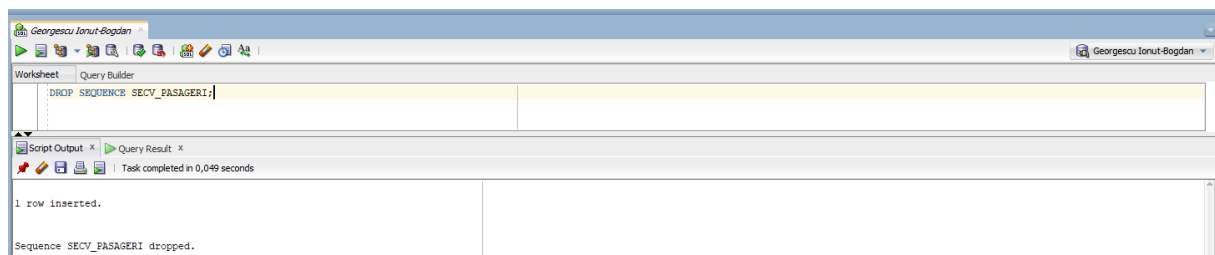
VALUES(SECV_PASAGERI.NEXTVAL, 'Mihalache', 'Marina', '0787450920', 'mmiha0121@yahoo.com', 'Online', 171544);



SELECT * FROM PASAGERI WHERE EMAIL = 'mmiha0121@yahoo.com';

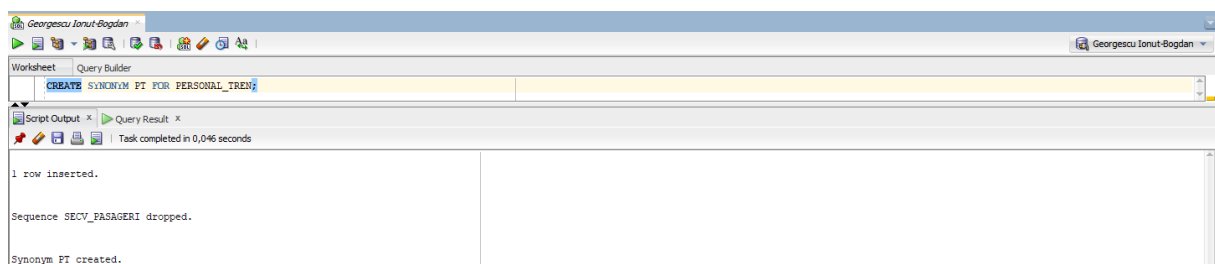


DROP SEQUENCE SECV_PASAGERI;



- SINONIME

CREATE SYNONYM PT FOR PERSONAL_TREN;



SELECT * FROM PT;

Georgescu Ionut-Bogdan

Worksheet Query Builder

```
CREATE SYNONYM PT FOR PERSONAL_TREN;
```

```
SELECT * FROM PT;
```

Script Output x Query Result x

All Rows Fetched: 20 in 0,098 seconds

	ID_ANGAJAT	NUME	PRENUME	FUNCTIE	SALARIU	DATA_ANGAJARII	ID_TREN
1	202901	Marinescu	Ion	Controlor Bilete	6050	12-09-1990	9102
2	202902	Vasile	Cristian	Mecanic Locomotiva	8250	22-08-2000	9102
3	202903	Ionescu	Marinela	Controlor Bilete	5610	23-09-1999	1583
4	202904	Gheorghe	Alina	Controlor Bilete	4290	24-10-2019	551
5	202906	Nicolae	Marian	Mecanic Locomotiva	7590	19-02-2003	1583
6	202907	Filipescu	Stefania	Controlor Bilete	4840	21-01-1998	9900
7	202908	Tudor	Andrei	Mecanic Locomotiva	7700	13-04-2004	9900
8	202909	Marinescu	Victor	Controlor Bilete	4950	11-01-2007	367
9	202910	Radu	Elena	Mecanic Locomotiva	7260	19-07-2014	367
10	202911	Ungureanu	Valeria	Controlor Bilete	5610	24-05-1999	1634
11	202912	Dumitrescu	Eugen	Controlor Bilete	5555	15-06-2001	367
12	202913	Dimitrie	Alexandru	Mecanic Locomotiva	6600	18-04-2022	10165
13	202914	Simionescu	Aurelia	Mecanic Locomotiva	7370	17-03-2008	1634
14	202915	Alexandrescu	Gabriel	Controlor Bilete	4730	23-02-2017	10165
15	202916	Andrescu	Gabriel	Controlor Bilete	4730	23-04-2017	3002
16	202925	Toma	Matei	Controlor Bilete	4730	23-02-2018	8001
17	202926	Badea	Alin	Controlor Bilete	4300	25-02-2017	9457
18	202935	Negut	Constantin	Controlor Bilete	4300	11-01-2015	9417
19	202936	Florescu	Gabriela	Controlor Bilete	4730	13-02-2022	10083
20	202946	Toma	Gabriela	Mecanic Locomotiva	6930	13-11-2022	10083

DROP SYNONYM PT;

Georgescu Ionut-Bogdan

Worksheet Query Builder

```
DROP SYNONYM PT;
```

Script Output x

Task completed in 0,088 seconds

Synonym PT dropped.

Interogări PL/SQL

- Să se adauge o nouă înregistrare in tabela PASAGERI.

begin

insert into pasageri values (

140505, 'Tanase', 'Ana', '0722123456', 'ana.tanase@gmail.com', 'Fizic',
171324);

commit;

end;

/

	ID_PASAGER	NUME	PRENUME	TELEFON	EMAIL	TIP_BILET	ID_RUTA
1	140505	Tanase	Ana	0722123456	ana.tanase@gmail.com	Fizic	171324

- Să se modifice adresa de e-mail a pasagerului cu id-ul 181524.

declare

v_email pasageri.email%type := 'tnicolaescu@gmail.com';

begin

update pasageri

set email = v_email where id_pasager = 181524;

commit;

end;

/

	ID_PASAGER	EMAIL
1	181524	teo.nicolae335@gmail.com

	ID_PASAGER	EMAIL
1	181524	tnicolaescu@gmail.com

- Să se șteargă înregistrarea din tabela PASAGERI cu id-ul 1004;

```
begin
  delete from pasageri
  where id_pasager = 1004;
  commit;
end;
/
```

ID_PASA... EMAIL

- Să se afișeze numele și prenumele pasagerului cu id 181524.

```
set serveroutput on
declare
  v_id_pasager pasageri.id_pasager%type := 181524;
  v_nume pasageri.nume%type;
  v_prenume pasageri.prenume%type;
begin
  select nume, prenume into v_nume, v_prenume
  from pasageri where id_pasager = v_id_pasager;

  dbms_output.put_line('Pasagerul cu ID = ' || v_id_pasager || ' este: ' || v_nume
  || ' ' || v_prenume);
end;
/
```

Pasagerul cu ID = 181524 este: Nicolaescu Teodor

PL/SQL procedure successfully completed.

- Să se afișeze tipul și capacitatea vagonului cu id 182944.

```
set serveroutput on
declare
  v_id_vagon vagoane.id_vagon%type := 182944;
  v_tip_vagon vagoane.tip_vagon%type;
  v_capacitate vagoane.capacitate%type;
begin
  select tip_vagon, capacitate into v_tip_vagon, v_capacitate
  from vagoane where id_vagon = v_id_vagon;
  dbms_output.put_line('Vagonul cu ID-ul ' || v_id_vagon || ' este de tip '
  || v_tip_vagon || ' si are capacitatea totala de ' || v_capacitate || ' locuri.');
```

Vagonul cu ID-ul 182944 este de tip Clasa 2 si are capacitatea totala de 80 locuri.

PL/SQL procedure successfully completed.

- Să se afișeze orașul și tipul stației PLOIESTI VEST.

```

set serveroutput on
declare
    v_ume_statie statii.ume_statie%type := 'Ploiesti Vest';
    v_oras statii.oras%type;
    v_tip_statie statii.tip_statie%type;
begin
    select oras, tip_statie into
    v_oras, v_tip_statie from statii where ume_statie like '%Ploiesti Vest%';
    dbms_output.put_line('Statia ' || v_ume_statie || ' este de tip '
    || v_tip_statie || ' si se afla in localitatea ' || v_oras);
end;
/

Statia Ploiesti Vest este de tip Jonctiune si se afla in localitatea Ploiesti

PL/SQL procedure successfully completed.

```

- Să se afișeze cât este prețul biletului cu id 200526 pentru un student. Atenție studenții au 90% reducere pentru biletele la Clasa 2 și 50% pentru biletele la Clasa 1.

```

set serveroutput on
declare
    v_id_bilet bilete.id_bilet%type := 200526;
    v_tip bilete.tip%type;
    v_pret bilete.pret%type;
begin
    select tip, pret into v_tip, v_pret
    from bilete where id_bilet = v_id_bilet;
    dbms_output.put_line('Pretul intreg al biletului este ' || v_pret);
    if v_tip = 'Clasa 1' then
        v_pret := 0.5 * v_pret;
    else
        v_pret := 0.1 * v_pret;
    end if;
    dbms_output.put_line('Pretul cu reducere pentru studenti este ' || v_pret);
end;
/

```

```

Pretul intreg al biletului este 33.5
Pretul cu reducere pentru studenti este 3.35

PL/SQL procedure successfully completed.

```

- Să se afișeze tipul vagonului cu id 183002.

```

set serveroutput on
declare
    v_id_vagon vagoane.id_vagon%type := 183002;
    v_tip_vagon vagoane.tip_vagon%type;
    v_mesaj varchar2(100);
begin
    select tip_vagon into v_tip_vagon from vagoane
    where id_vagon = v_id_vagon;
    v_mesaj := case v_tip_vagon
    when 'Cuseta' then 'Acesta este un vagon de tip CUSETA.'
    when 'Restaurant' then 'Acesta este un vagon RESTAURANT.'
    when 'Etajat' then 'Acetsa este un vagon de tip ETAJAT.'
    when 'Clasa 1' then 'Acesta este un vagon de tip SALON CLASA 1'
    when 'Clasa 2' then 'Acesta este un vagon de tip SALON CLASA 2'
    when 'Automotor' then 'Acesta este un vagon de tip AUTOMOTOR.'
    else 'Acest vagon este de alt tip'
    end;
    dbms_output.put_line(v_mesaj);
end;
/

```

```
Acesta este un vagon de tip AUTOMOTOR.
```

```
PL/SQL procedure successfully completed.
```

- Să se afișeze tipul capacității vagonului cu id 183002.

```

set serveroutput on
declare
    v_id_vagon vagoane.id_vagon%type := 183002;
    v_capacitate vagoane.capacitate%type;
    v_mesaj varchar2(100);
begin
    select capacitate into v_capacitate from vagoane
    where id_vagon = v_id_vagon;
    case
        when v_capacitate < 50 then v_mesaj := 'capacitate mica.';
        when v_capacitate between 50 and 90 then v_mesaj := 'capacitate medie';
        when v_capacitate > 90 then v_mesaj := 'capacitate medie';
        else v_mesaj := 'capacitate necunocuta';
    end case;
    dbms_output.put_line('Vagonul cu ID = ' || v_id_vagon || ' are ' || v_capacitate
    || ' locuri. Acesta are o '

```

```

        || v_mesaj);
end;
/
Vagonul cu ID = 183002 are 60 locuri. Acesta are o capacitate medie

PL/SQL procedure successfully completed.

```

- Să se afișeze numele și prenumele tuturor pasagerilor din tabela PASAGERI.

```

set serveroutput on
declare
    v_id_pasager pasageri.id_pasager%type;
    v_nume pasageri.nume%type;
    v_prenume pasageri.prenume%type;

    cursor c_pasager is select id_pasager, nume, prenume from pasageri;
begin
    open c_pasager;

    loop
        fetch c_pasager into v_id_pasager, v_nume, v_prenume;
        exit when c_pasager%notfound;
        dbms_output.put_line('ID: ' || v_id_pasager || ' | NUME: ' || v_nume ||
            ' | PRENUME: ' || v_prenume);
    end loop;

    close c_pasager;
end;
/

```

```

ID: 140505 | NUME: Tanase | PRENUME: Ana
ID: 180902 | NUME: Popescu | PRENUME: Ion
ID: 181101 | NUME: Andreescu | PRENUME: Ioana
ID: 181306 | NUME: Marinescu | PRENUME: Cristian-Stefan
ID: 181309 | NUME: Bivol | PRENUME: Ilie
ID: 181334 | NUME: Nistor | PRENUME: Elena
ID: 181358 | NUME: Radoi | PRENUME: Alexandra
ID: 181423 | NUME: Sorescu | PRENUME: Damian
ID: 181447 | NUME: Mihaescu | PRENUME: Adriana
ID: 181501 | NUME: Nica | PRENUME: Octavian
ID: 181516 | NUME: Rusu | PRENUME: Maria
ID: 181524 | NUME: Nicolaescu | PRENUME: Teodor
ID: 181529 | NUME: Sandulescu | PRENUME: Georgiana
ID: 181533 | NUME: Ionescu | PRENUME: Marian
ID: 181549 | NUME: Vasilache | PRENUME: Daria
ID: 181604 | NUME: Irimia | PRENUME: Alina
ID: 181624 | NUME: Matei | PRENUME: Ionut
ID: 181634 | NUME: Eremia | PRENUME: Alina
ID: 181644 | NUME: Voicu | PRENUME: Andreea
ID: 181654 | NUME: Marinescu | PRENUME: Georgiana
ID: 181704 | NUME: Raicu | PRENUME: Gabriel

```

PL/SQL procedure successfully completed.

- Să se afișeze id-ul biletului, data călătoriei, vagonul și locul pentru toată înregistrările din tabela BILETE;

```
set serveroutput on
```

```
begin
```

```
  for rec in ( select b.id_bilet, b.data_calatoriei, b. vagon, b.loc from bilete b order by
b.data_calatoriei) loop
```

```
    dbms_output.put_line('Bilet: ' || rec.id_bilet || ' | Data calatoriei: ' ||
rec.data_calatoriei
```

```
  || ' | Vagon: ' || rec.vagon || ' | Loc: ' || rec.loc);
```

```
  end loop;
```

```
end;
```

```
/
```

```

Bilet: 200312 | Data calatoriei: 12-FEB-24 | Vagon: 2 | Loc: 44
Bilet: 200433 | Data calatoriei: 12-FEB-24 | Vagon: 1 | Loc: 65
Bilet: 200346 | Data calatoriei: 15-FEB-24 | Vagon: 2 | Loc: 22
Bilet: 200412 | Data calatoriei: 12-MAR-24 | Vagon: 4 | Loc: 21
Bilet: 200322 | Data calatoriei: 12-APR-24 | Vagon: 5 | Loc: 55
Bilet: 200526 | Data calatoriei: 23-MAY-24 | Vagon: 3 | Loc: 45
Bilet: 200538 | Data calatoriei: 23-MAY-24 | Vagon: 2 | Loc: 21
Bilet: 200501 | Data calatoriei: 27-MAY-24 | Vagon: 1 | Loc: 77
Bilet: 200559 | Data calatoriei: 23-JUN-24 | Vagon: 3 | Loc: 41
Bilet: 200602 | Data calatoriei: 24-JUN-24 | Vagon: 2 | Loc: 44
Bilet: 200627 | Data calatoriei: 29-JUN-24 | Vagon: 3 | Loc: 27
Bilet: 200629 | Data calatoriei: 30-JUN-24 | Vagon: 1 | Loc: 25
Bilet: 200855 | Data calatoriei: 10-JUL-24 | Vagon: 3 | Loc: 47
Bilet: 200744 | Data calatoriei: 10-JUL-24 | Vagon: 2 | Loc: 21
Bilet: 200925 | Data calatoriei: 10-JUL-24 | Vagon: 2 | Loc: 55
Bilet: 200935 | Data calatoriei: 11-JUL-24 | Vagon: 2 | Loc: 55
Bilet: 200926 | Data calatoriei: 12-JUL-24 | Vagon: 2 | Loc: 53
Bilet: 200945 | Data calatoriei: 12-JUL-24 | Vagon: 1 | Loc: 51
Bilet: 200965 | Data calatoriei: 14-JUL-24 | Vagon: 1 | Loc: 55
Bilet: 200955 | Data calatoriei: 15-JUL-24 | Vagon: 1 | Loc: 15

```

- Să se afișeze informații despre trenuri

```
set serverout on
declare
    v_id_tren trenuri.id_tren%type;
    v_tip_tren trenuri.tip_tren%type;
    v_nr_pasageri trenuri.nr_pasageri%type;
    v_index number := 1;
    v_total number;
begin
    select count(*) into v_total from trenuri;

    while v_index <= v_total loop
        select id_tren, tip_tren, nr_pasageri
        into v_id_tren, v_tip_tren, v_nr_pasageri
        from (select id_tren, tip_tren, nr_vagoane, nr_pasageri, row_number()
        over (order by id_tren) as rn from trenuri)
        where rn = v_index;

        dbms_output.put_line('Tren ID: ' || v_id_tren || ' | Tip tren: ' || v_tip_tren ||
        ' | Numar pasageri: ' || v_nr_pasageri);

        v_index := v_index + 1;

    end loop;
end;
/
```

```

Tren ID: 367 | Tip tren: INTERREGIO | Numar pasageri: 300
Tren ID: 472 | Tip tren: INTERREGIO | Numar pasageri: 232
Tren ID: 531 | Tip tren: INTERCITY | Numar pasageri: 89
Tren ID: 551 | Tip tren: INTERCITY | Numar pasageri: 612
Tren ID: 1583 | Tip tren: INTERREGIO | Numar pasageri: 260
Tren ID: 1634 | Tip tren: INTERREGIO | Numar pasageri: 201
Tren ID: 3002 | Tip tren: REGIO-EXPRESS | Numar pasageri: 300
Tren ID: 7922 | Tip tren: REGIO-EXPRESS | Numar pasageri: 280
Tren ID: 8001 | Tip tren: REGIO | Numar pasageri: 256
Tren ID: 9001 | Tip tren: REGIO | Numar pasageri: 320
Tren ID: 9102 | Tip tren: REGIO | Numar pasageri: 400
Tren ID: 9205 | Tip tren: REGIO-EXPRESS | Numar pasageri: 175
Tren ID: 9417 | Tip tren: REGIO | Numar pasageri: 100
Tren ID: 9434 | Tip tren: REGIO | Numar pasageri: 87
Tren ID: 9457 | Tip tren: REGIO-EXPRESS | Numar pasageri: 55
Tren ID: 9900 | Tip tren: REGIO-EXPRESS | Numar pasageri: 298
Tren ID: 10083 | Tip tren: INTERREGIO | Numar pasageri: 110
Tren ID: 10165 | Tip tren: REGIO | Numar pasageri: 165
Tren ID: 10268 | Tip tren: REGIO | Numar pasageri: 100
Tren ID: 10356 | Tip tren: REGIO | Numar pasageri: 256

PL/SQL procedure successfully completed.

```

- Să se afișeze informații despre trenuri și ruta pe care circulă fiecare tren

```

set serveroutput on
declare
    cursor tren_c is
        select t.id_tren, t.tip_tren, r.plecare, r.sosire
        from trenuri t join rute r on t.id_ruta = r.id_ruta order by t.id_tren;

    v_id_tren trenuri.id_tren%type;
    v_tip_tren trenuri.tip_tren%type;
    v_plecare rute.plecare%type;
    v_sosire rute.sosire%type;
begin
    open tren_c;
    loop
        fetch tren_c into v_id_tren, v_tip_tren, v_plecare, v_sosire;
        exit when tren_c%notfound;

        dbms_output.put_line('Tren: ' || v_tip_tren || ' ' || v_id_tren || ' Ruta: ' ||
v_plecare || ' - ' || v_sosire);

    end loop;
    close tren_c;

```

end;

/

```
Tren: INTERREGIO 367 Ruta: Cluj-Napoca - Oradea
Tren: INTERREGIO 472 Ruta: Bucuresti Nord - Sinaia
Tren: INTERCITY 531 Ruta: Bucuresti Nord - Ploiesti Vest
Tren: INTERCITY 551 Ruta: Bucuresti Nord - Buzau
Tren: INTERREGIO 1583 Ruta: Bucuresti Nord - Constanta
Tren: INTERREGIO 1634 Ruta: Bucuresti Nord - Ploiesti Vest
Tren: REGIO-EXPRESS 3002 Ruta: Bucuresti Nord - Brasov
Tren: REGIO-EXPRESS 7922 Ruta: Bucuresti Nord - Aeroport Henri Coanda
Tren: REGIO 8001 Ruta: Bucuresti Obor - Constanta
Tren: REGIO 9001 Ruta: Bucuresti Nord - Rosiori Nord
Tren: REGIO 9102 Ruta: Bucuresti - Targoviste
Tren: REGIO-EXPRESS 9205 Ruta: Bucuresti Nord - Pitesti
Tren: REGIO 9417 Ruta: Bucuresti Progresul - Giurgiu
Tren: REGIO 9434 Ruta: Titu - Targoviste
Tren: REGIO-EXPRESS 9457 Ruta: Bucuresti Progresul - Bucuresti Nord
Tren: REGIO-EXPRESS 9900 Ruta: Bucuresti Nord - Craiova
Tren: INTERREGIO 10083 Ruta: Bucuresti Nord - Brasov
Tren: REGIO 10165 Ruta: Bucuresti Nord - Pietrosita
Tren: REGIO 10268 Ruta: Pietrosita - Targoviste
Tren: REGIO 10356 Ruta: Titan Sud - Oltenita
```

PL/SQL procedure successfully completed.

- Să se creeze o procedură care arată numărul angajaților pentru fiecare funcție din PERSONAL_TREN.

set serveroutput on

create or replace procedure statistica_personal is v_total number;

begin

select count(*) into v_total from personal_tren;

if v_total = 0 then

dbms_output.put_line('Nu exista anagajati!');

else

for rec in (

select functie, count(*) as nr_ang from personal_tren
group by functie)

loop

dbms_output.put_line('Functia: ' || rec.functie
|| ', Nr. angajati: ' || rec.nr_ang);

end loop;

end if;

exception

when OTHERS then

```

        dbms_output.put_line('A aparut o eroare neasteptata!');
    end;
/

```

```

exec statistica_personal;

```

```

Procedure STATISTICA_PERSONAL compiled

```

```

Functia: Controlor Bilete, Nr. angajati: 13

```

```

Functia: Mecanic Locomotiva, Nr. angajati: 7

```

```

PL/SQL procedure successfully completed.

```

- Să se scrie o procedură care mărește salariul cu 15% angajatului cu ID-ul trimis ca parametru dacă acesta are o vechime mi mare de 5 ani.

```

set serveroutput on

```

```

create or replace procedure modifica_salariul(p_id_angajat in
personal_tren.id_angajat%type)

```

```

is

```

```

    v_salariul_vechi personal_tren.salariul%type;

```

```

    v_salariul_nou personal_tren.salariul%type;

```

```

begin

```

```

    select salariul into v_salariul_vechi from personal_tren

```

```

    where id_angajat = p_id_angajat;

```

```

    update personal_tren

```

```

    set salariul = salariul * 1.15

```

```

    where trunc((months_between(sysdate, data_angajarii))/12) >= 5;

```

```

    select salariul into v_salariul_nou from personal_tren

```

```

    where id_angajat = p_id_angajat;

```

```

        dbms_output.put_line('Salariatul cu ID: ' || p_id_angajat || ', salariul vechi: ' ||
v_salariul_vechi || ', salariul nou: ' || v_salariul_nou);

```

```

exception

```

```

    when NO_DATA_FOUND then

```

```

        raise_application_error(-20001, 'Nu exista niciun angajat cu acest ID!');

```

```

end;

```

```

/

```

```

exec modifica_salariul(202906);

```

```

exec modifica_salariul(1001);

```


Procedure MODIFICA_SALARIUL compiled

Salariatul cu ID: 202906, salariul vechi: 7590, salariul nou: 8728.5

PL/SQL procedure successfully completed.

Error starting at line : 25 in command -

BEGIN modifica_salariul(1001); END;

Error report -

DRA-20001: Nu exista niciun angajat cu acest ID!

DRA-06512: at "GEORGESCUI_56.MODIFICA_SALARIUL", line 19

DRA-06512: at line 1

- Să se scrie o procedură cu parametrii In și OUT care returnează informații despre un tren în funcție de ID-ul acestuia.

set serveroutput on

create or replace procedure show_tren(p_id_tren IN trenuri.id_tren%type,
p_tip_tren OUT trenuri.tip_tren%type, p_plecure OUT rute.plecure%type, p_sosire
OUT rute.sosire%type)

is

begin

select t.tip_tren, r.plecure, r.sosire into p_tip_tren, p_plecure, p_sosire
from trenuri t, rute r
where t.id_ruta = r.id_ruta and id_tren = p_id_tren;

dbms_output.put_line('Trenul ' || p_tip_tren || ' ' || p_id_tren
|| ' Ruta: ' || p_plecure || ' - ' || p_sosire);

exception

when NO_DATA_FOUND then

raise_application_error(-20001, 'Nu exista niciun tren cu acest ID!');

when OTHERS then

raise_application_error(-20002, 'A aparut o eroare neasteptata');

end;

/

declare

v_tip_tren trenuri.tip_tren%type;

v_plecure rute.plecure%type;

v_sosire rute.sosire%type;

begin

show_tren(551, v_tip_tren, v_plecure, v_sosire);

end;

/

```
Procedure SHOW_TREN compiled
```

```
Trenul INTERCITY 551 Ruta: Bucuresti Nord - Buzau
```

```
PL/SQL procedure successfully completed.
```

- Să se scrie o procedură cu 1 parametru de tip IN OUT care returneaza ID-ul unui pasager.

```
set serveroutput on
```

```
create or replace procedure cauta_pasager(p_id_pasager IN OUT varchar2)
```

```
is
```

```
    v_id_pasager pasageri.id_pasager%type;
```

```
begin
```

```
    select id_pasager into v_id_pasager from pasageri
```

```
    where id_pasager = to_number(p_id_pasager);
```

```
    p_id_pasager := to_char(v_id_pasager);
```

```
exception
```

```
    when NO_DATA_FOUND then
```

```
        p_id_pasager := '-1';
```

```
    when OTHERS then
```

```
        p_id_pasager := '-1';
```

```
end;
```

```
/
```

```
declare
```

```
    v_id_pasager varchar2(10);
```

```
begin
```

```
    v_id_pasager := 181101;
```

```
    cauta_pasager(v_id_pasager);
```

```
    dbms_output.put_line(v_id_pasager);
```

```
end;
```

```
/
```

```
Procedure CAUTA_PASAGER compiled
```

```
181101
```

```
PL/SQL procedure successfully completed.
```

- Să se creeze o funcție care afișează numărul pasagerilor de pe o anumită rută.

```

set serveroutput on
create or replace function nr_pasageri_ruta(p_id_ruta IN number)
return number is
    v_total number := 0;
begin
    select count(distinct b.id_pasager)
    into v_total from bilete b
    join rute r on b.id_ruta = r.id_ruta
    where r.id_ruta = p_id_ruta;

    return nvl(v_total, 0);
exception
    when NO_DATA_FOUND then
        return 0;
    when OTHERS then
        return -1;
end;
/

```

```

select nr_pasageri_ruta(172567) as nr_pasageri_ruta from dual;

```

NR_PASAGERI_RUTA	
1	2

- Să se creeze un pachet numit pachet_feroviar care să cuprindă subprogramele statistica_tren, afisare_detalii_statie, salariul_mediu_angajati, viteza_medie_trenuri.

```

set serveroutput on
create or replace package pachet_feroviar
is
    procedure statistica_tren(p_id_tren in number);
    procedure afisare_detalii_statie(p_id_statie in number);
    function salariu_mediu_angajati return number;
    function viteza_medie_trenuri return number;
end pachet_feroviar;
/

```

```

create or replace package body pachet_feroviar
is
    procedure statistica_tren(p_id_tren in number)
    is
        v_tip_tren trenuri.tip_tren%type;
        v_nr_vagoane trenuri.nr_vagoane%type := 0;

```

```

v_nr_locuri vagoane.capacitate%type := 0;
v_nr_personal number := 0;
v_total_locuri number := 0;
begin
    select count(*) into v_nr_personal from personal_tren
    where id_tren = p_id_tren;

    select distinct min(tip_tren), min(nr_vagoane) into v_tip_tren, v_nr_vagoane
    from trenuri where id_tren = p_id_tren;

    select sum(capacitate) into v_nr_locuri
    from vagoane where id_tren = p_id_tren;

    v_total_locuri := v_nr_vagoane * v_nr_locuri;

    dbms_output.put_line('Trenul ' || v_tip_tren || p_id_tren || ' | Total locuri pasageri: '
        || v_total_locuri || ' | Nr. angajati: ' || v_nr_personal);
exception
    when NO_DATA_FOUND then
        raise_application_error(-20001, 'Nu exista trenul cu ID = ' || p_id_tren);
    when OTHERS then
        raise_application_error(-20002, 'A aparut o eroare in afisarea statisticii trenului ' ||
p_id_tren);
end statistica_tren;

procedure afisare_detalii_statie(p_id_statie in number)
is
    v_nume statii.nume_statie%type;
    v_oras statii.oras%type;
    v_regionala regionala.oras_sediu%type;
begin
    select s.nume_statie, s.oras, r.oras_sediu into v_nume, v_oras, v_regionala
    from statii s join regionala r on s.id_regionala = r.id_regionala
    where id_statie = p_id_statie;

    dbms_output.put_line('Regionala C.F.R. ' || v_regionala || ', Statia ' || v_nume || ' din
orasul ' || v_oras);
exception
    when NO_DATA_FOUND then
        raise_application_error(-20003, 'Nu exista statia cu ID = ' || p_id_statie);
    when OTHERS then
        raise_application_error(-20001, 'A aparut o eroare in afisarea detaliilor statiei ' ||
p_id_statie);
end afisare_detalii_statie;

```

```

function salariu_mediu_angajati return number
is
    v_sal_mediu personal_tren.salariul%type := 0;
begin
    select avg(salariul) into v_sal_mediu from personal_tren;
    return v_sal_mediu;
exception
    when NO_DATA_FOUND then
        return 0;
    when OTHERS then
        return -1;
end salariu_mediu_angajati;

```

```

function viteza_medie_trenuri return number
is
    v_durata number := 0;
    v_distanta number := 0;
    v_viteza_medie number := 0;
begin
    select sum(durata), sum(nr_km) into v_durata, v_distanta
    from rute;

    v_durata := v_durata / 60;

    v_viteza_medie := v_distanta / v_durata;

    return trunc(v_viteza_medie, 2);
exception
    when NO_DATA_FOUND then
        return 0;
    when OTHERS then
        return -1;
end viteza_medie_trenuri;
end pachet_feroviar;
/

```

```

exec pachet_feroviar.statistica_tren(9102);
exec pachet_feroviar.afisare_detalii_statie(1003);
select pachet_feroviar.salariu_mediu_angajati, pachet_feroviar.viteza_medie_trenuri
from dual;

```

	⚡ SALARIU_MEDIU_ANGAJATI	⚡ VITEZA_MEDIE_TRENURI
1	6540.24	54.21

```

Package PACHET_FEROVIAR compiled

Package Body PACHET_FEROVIAR compiled

Trenul REGIO9102 | Total locuri pasgaeri: 1540 | Nr. angajati: 2

PL/SQL procedure successfully completed.

Regionala C.F.R. Bucuresti, Statia Targoviste din orasul Targoviste

PL/SQL procedure successfully completed.

>>Query Run In:Query Result

```

- Să se creeze un trigger care nu permite stergerea rutelor pe care s-au emis bilete.

```

create or replace trigger trg_no_delete_rute_with_bilet
before delete on rute
for each row
declare
    v_count number;
begin
    select count(*) into v_count
    from bilete
    where id_ruta = :old.id_ruta;

    if v_count > 0 then
        raise_application_error(-20020,
            'Nu se poate sterge acest tren! S-au emis bilete pentru acest tren!');
    end if;
end;
/

delete from rute where id_ruta = 171324;

```

```

Trigger TRG_NO_DELETE_RUTE_WITH_BILET compiled

Error starting at line : 18 in command -
delete from rute where id_ruta = 171324
Error at Command Line : 18 Column : 13
Error report -
SQL Error: ORA-20020: Nu se poate sterge acest tren! S-au emis bilete pentru acest tren!
ORA-06512: at "GEORGESCUI_56.TRG_NO_DELETE_RUTE_WITH_BILET", line 9
ORA-04088: error during execution of trigger 'GEORGESCUI_56.TRG_NO_DELETE_RUTE_WITH_BILET'

```

- Să se creeze un trigger care nu permite inserarea unui bilet cu data călătoriei înainte de sysdate.

```

create or replace trigger trg_bilete_data_calatorie
before insert on bilete
for each row
begin
    if :new.data_calatorie < sysdate then
        raise_application_error(-20010,
            'Nu se poate insera un bilet cu data calatoriei in trecut');
    end if;
end;
/
insert into bilete
values(0008, 'Clasa 2', 55, 2, '19-MAY-25', '20-MAY-25', 17, 17011, 1);

```

```

Trigger TRG_BILETE_DATA_CALATORIE compiled

Error starting at line : 11 in command -
insert into bilete
values(0008, 'Clasa 2', 55, 2, '19-MAY-25', '20-MAY-25', 17, 17011, 1)
Error at Command Line : 11 Column : 13
Error report -
SQL Error: ORA-20010: Nu se poate insera un bilet cu data calatoriei in trecut
ORA-06512: at "GEORGESCUI_56.TRG_BILETE_DATA_CALATORIE", line 3
ORA-04088: error during execution of trigger 'GEORGESCUI_56.TRG_BILETE_DATA_CALATORIE'

```

- Să se creeze un trigger care scrie operatiunile facute asupra pretului unui bilet.

```

create table bilete_audit (
    id_audit number generated by default as identity primary key,
    id_bilet number, pret_vechi number, pret_nou number, data_modificarii date);
create or replace trigger trg_audit_bilet
after update of pret on bilete
for each row
begin
    if :old.pret <> :new.pret then
        insert into bilete_audit (id_bilet, pret_vechi, pret_nou, data_modificarii)
        values(:old.id_bilet, :old.pret, :new.pret, sysdate);
    end if;
end;

```

```
end;
/
```

```
update bilete
set pret = pret * 1.1
where id_bilet = 200312;
```

```
select * from bilete_audit;
```

	ID_AUDIT	ID_BILET	PRET_VECHI	PRET_NOU	DATA_MODIFICARII
1	1	200312	17	18.7	22-MAY-25

- Căutați un pasager cu ID = 992134.

```
set serveroutput on
declare
    v_nume pasageri.nume%type;
    v_id_pasager number := 992134;
begin
    select nume into v_nume
    from pasageri
    where id_pasager = v_id_pasager;
    dbms_output.put_line('Nume pasager: ' || v_nume);
exception
    when NO_DATA_FOUND then
        dbms_output.put_line('Nu exista cu ID-ul ' || v_id_pasager);
end;
/
```

```
Nu exista cu ID-ul 992134
```

```
PL/SQL procedure successfully completed.
```

- Să se găsească ruta cu plecare din București Nord.

```
set serveroutput on
declare
    v_id_ruta rute.id_ruta%type;
    v_plecare rute.plecare%type := 'Bucuresti Nord';
begin
    select id_ruta into v_id_ruta
    from rute
    where plecare = v_plecare;
    dbms_output.put_line('ID ruta: ' || v_id_ruta);
exception
    when TOO_MANY_ROWS then
        dbms_output.put_line('Exista mai multe rute cu plecare din ' || v_plecare);
```



```
end;  
/  
Exista mai multe rute cu plecare din Bucuresti Nord
```

```
PL/SQL procedure successfully completed.
```

- Să se calculeze prețul biletelor de pe o anumită rută.

```
set serveroutput on
```

```
declare
```

```
    v_id_ruta rute.id_ruta%type := 999999;
```

```
    v_total_pret number := 0;
```

```
    v_nr_bilete number := 0;
```

```
    v_media number;
```

```
begin
```

```
    select sum(pret), count(*) into v_total_pret, v_nr_bilete  
    from bilete  
    where id_ruta = v_id_ruta;
```

```
    v_media := v_total_pret / v_nr_bilete;
```

```
    dbms_output.put_line('Media preturilor: ' || v_media);
```

```
exception
```

```
    when ZERO_DIVIDE then
```

```
        dbms_output.put_line('Nu exista bilete pentru ruta cu ID-ul ' || v_id_ruta);
```

```
end;
```

```
/
```

```
Media preturilor:
```

```
PL/SQL procedure successfully completed.
```