

Sistem de administrare a trenurilor CFR

Oprișan Elena-Grația

Subgrupa 7, Informatică română ,Anul 2

1)Descriere

Scopul proiectului este de a permite organizarea și stocarea eficientă a datelor referitoare la desfășurarea activității sistemului de transport feroviar în România. Călătoriile reprezintă un interes atât național, internațional dar și personal. Iată câteva interese:

- Transportul feroviar este esențial pentru transportul de mărfuri, având un impact direct asupra dezvoltării economice și a facilitării comerțului intern.
- Există trenuri internaționale care leagă orașe și țări, consolidând relațiile bilaterale și favorizând mobilitatea internațională.
- Călătoriile cu trenul pot fi o modalitate ideală de a explora destinații turistice, oferind călătorilor șansa de a se bucura de priveliști unice și de a experimenta cultura locală.

Așadar proiectul urmărește monetizarea călătoriilor, conductorilor dar și a stațiilor vizitate pe parcursul traseului.

2)Dependentele funcționale și forma normala BCNF(utilizând normalizarea)

TREN

TREN.ID->TREN.ACHIZITIE

TREN.ID->TREN.VALABIL

TREN.ID->TREN.TIP

STAȚIE

STATIE.ID->STATIE.RESTAURANT

STATIE.ID->STATIE.ORAS

CĂLATORIE

CALATORIE.ID->CALATORIE.START

CALATORIE.ID->CALATORIE.FINISH

CALATORIE.ID->CALATORIE.PRET

CALATORIE.ID->CALATORIE.DURATA

CONDUCTOR

CONDUCTOR.ID->CONDUCTOR.NUME

CONDUCTOR.ID->CONDUCTOR.PRENUME

CONDUCTOR.ID->NR_TEL

CONDUCTOR.ID->NR_CALATORII

ORAȘ

ORAS.ID->ORAS.NUME

ORAS.ID->ORAS.JUDET

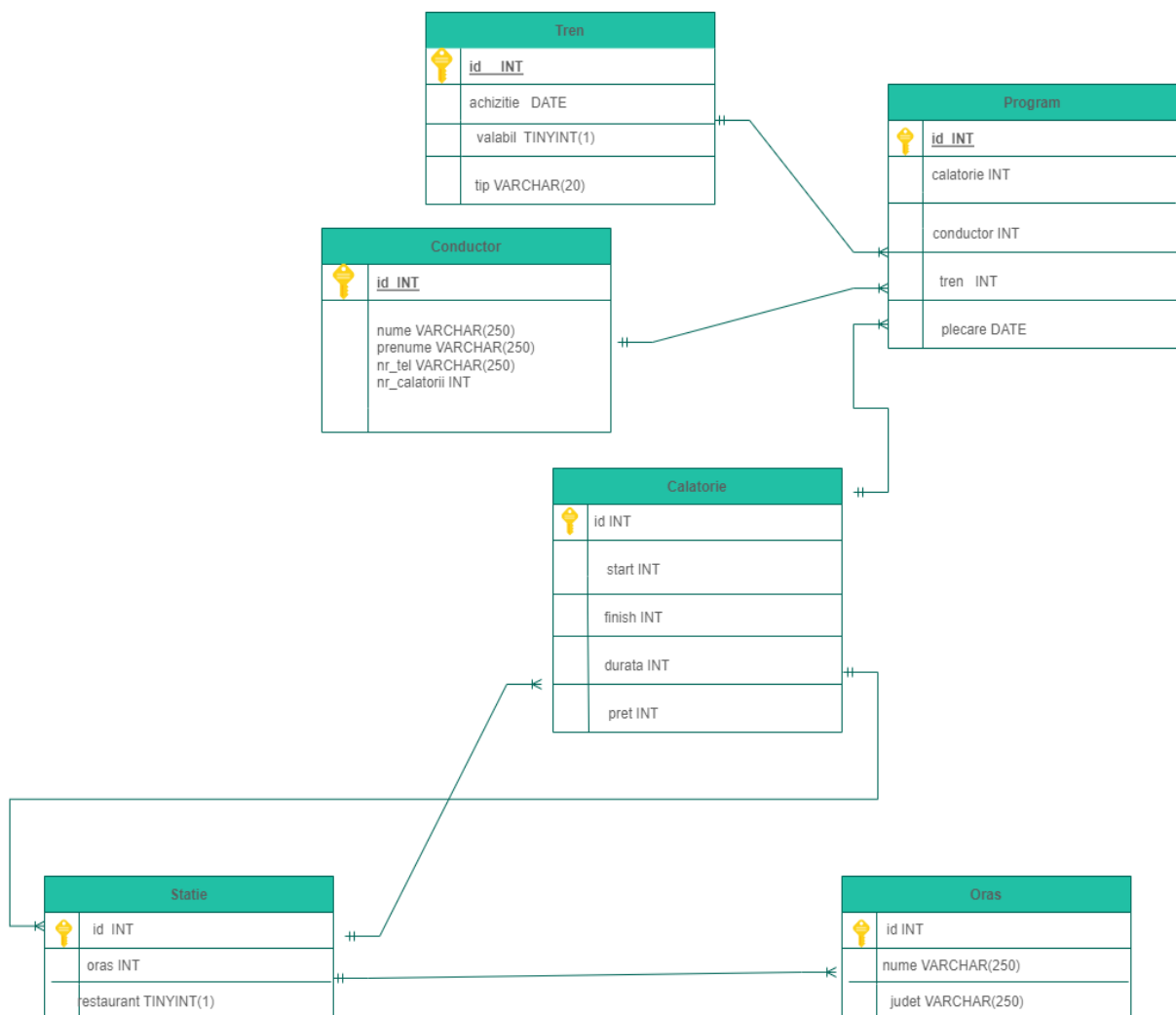
PROGRAM

PROGRAM.ID->PROGRAM.PLECARE

PROGRAM.ID->PROGRAM.SOSIRE

PROGRAM.ID->{PROGRAM.CONDUCTOR,PROGRAM.CALATORIE, PROGRAM.TREN }

3)Diagrama entitate-relație (ERD)



RELATII ÎNTRE TABELE

Tabela TREN:

Este conectată la tabela PROGRAM prin intermediul cheii primare id și cheii externe tren.

Nu are legături directe cu celelalte tabele.

Tabela ORAS:

Tabela STATIE utilizează o cheie externă (oras) care face referire la cheia primară id din ORAS.

Tabela STATIE:

Are o relație cu tabela ORAS prin intermediul cheii externe oras.

Tabela CONDUCTOR:

Tabela PROGRAM utilizează o cheie externă (conductor) care face referire la cheia primară id din CONDUCTOR.

Tabela CALATORIE:

Tabela PROGRAM are două chei externe ("start" și finish), ambele fac referire la cheia primară id din STATIE.

Tabela PROGRAM:

Este conectată la toate celelalte tabele (CALATORIE, CONDUCTOR, TREN).

Utilizează chei externe (calatorie, conductor, tren) care fac referire la cheile primare ale celorlalte tabele.

1. **One-to-Many (Unu-la-Mulți):**

- Tabela **ORAS** și tabela **STATIE**: Un oraș poate avea mai multe stații, dar o stație aparține unui singur oraș. Este o relație one-to-many, unde **ORAS** este tablă părinte, iar **STATIE** este tablă copil.
- Tabela **CONDUCTOR** și tabela **PROGRAM**: Un conducător poate avea mai multe programe, dar un program este asociat cu un singur conducător. Este o relație one-to-many, unde **CONDUCTOR** este tablă părinte, iar **PROGRAM** este tablă copil.
- Tabela **TREN** și tabela **PROGRAM**: Un tren poate fi asociat cu mai multe programe, dar un program este legat de un singur tren. Este o relație one-to-many, unde **TREN** este tablă părinte, iar **PROGRAM** este tablă copil.
- Tabela **CALATORIE** și tabela **PROGRAM**: O călătorie poate fi asociată cu mai multe programe, dar un program este legat de o singură călătorie. Este o relație one-to-many, unde **CALATORIE** este tablă părinte, iar **PROGRAM** este tablă copil.

2. **Many-to-One:**

- Din perspectiva tabelii **PROGRAM**, avem relații many-to-one către **CONDUCTOR**, **CALATORIE** și **TREN**, deoarece un program poate fi legat de un singur conducător, o singură călătorie și un singur tren.
- Din perspectiva tabelii **STATIE**, avem o relație many-to-one către **ORAS**, deoarece o stație aparține unui singur oraș.

3. **Many-to-Many:**

- Este intermediată de tabela **PROGRAM**, care leagă **CONDUCTOR**, **CALATORIE** și **TREN**. Mai exact, un **CONDUCTOR** poate avea mai multe programe, iar o **CALATORIE** și un **TREN** pot fi implicate în mai multe programe.

AFIȘAREA TABELELOR

TREN

ID	ACHIZITIE	AVAILABLE	TIP
3	02/15/2024	1	marfar
21	05/10/2024	0	pasageri
42	02/02/2024	0	pasageri
4	02/01/2024	0	marfar
41	02/01/2024	0	marfar
1	02/01/2024	1	pasageri

PROGRAM

ID	CALATORIE	CONDUCTOR	TREN	PLECARE
24	41	23	42	02/01/2024
45	22	21	4	10/01/2024
23	22	21	4	02/01/2024
26	41	23	42	12/12/2024
25	23	3	1	02/01/2024
46	23	3	1	09/06/2024

6 rows returned in 0.00 seconds [Download](#)

CONDUCTOR

ID	NUME	PRENUME	NR_TEL	NR_CALATORI
22	Dumitrescu	Elena	0720555666	5
1	Popescu	Ion	0720123456	15
2	Popescu	Ana	0720111222	8
21	Ionescu	Mihai	0720333444	12
23	Radu	Ionel	0720777888	12
3	Georgescu	Ionel	0720999000	12

6 rows returned in 0.00 seconds [Download](#)

CALATORIE

ID	start	FINISH	DURATA	PRET
22	2	1	90	30
21	1	3	120	35
41	1	3	120	50
23	3	2	150	40
3	1	3	120	50
61	2	1	90	30

6 rows returned in 0.01 seconds [Download](#)

STATIE

ID	ORAS	RESTAURANT
21	1	0
3	3	0
2	2	1
1	1	1

4 rows returned in 0.00 seconds [Download](#)

ORAS

ID	NUME	JUDET
1	Blaj	Alba
2	Mirau	Sibiu
3	Azuga	Prahova
47	Bistrita	Bistrița-Năsăud
41	Alba Iulia	Alba
45	Oradea	Bihor
42	Arad	Arad
61	Bacău	Bacău

8 rows returned in 0.01 seconds [Download](#)

4)Definirea tabelelor-
creare,modificare,ștergere,redenumire,trunchiere

- Crearea schemei PROIECTBD (în apex nu este nevoie creerea unei scheme)

```
1 CREATE SCHEMA `PROIECTBD` ;
```

- Crearea tabelului **Tren** unde vom stoca datele despre acestea

```
CREATE TABLE TREN (
  id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
  achizitie DATE NOT NULL,
  valabil NUMBER(1, 0) DEFAULT 0,
  tip VARCHAR2(20 CHAR) NOT NULL,
  nume_garaj VARCHAR2(20 CHAR) NOT NULL,
  CONSTRAINT PK_ID PRIMARY KEY(id)
);
```

Table created.

0.06 seconds

- Alterarea tabelului **Tren**, adaugarea constraint CK_TIP, redenumirea coloanei, ștergerea unei coloane

```
10 -- Adăugare constraint CK_TIP
11 ALTER TABLE TREN
12 ADD CONSTRAINT CK_TIP CHECK (tip IN ('marfar', 'pasageri'));
13
14 -- Redenumirea coloanei valabil în available
15 ALTER TABLE TREN
16 RENAME COLUMN valabil TO available;
17
18 -- Ștergerea coloanei in_garaj
19 ALTER TABLE Tren
20 DROP COLUMN nume_garaj;
21
```

Table altered.

0.08 seconds

Se va afișa mesajul *Table altered* pentru fiecare modificare adusă tabelului.

- Crearea tabelului **ORAS**

```

22
23 CREATE TABLE ORAS (
24     id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
25     nume VARCHAR2(250 CHAR) NOT NULL,
26     judet VARCHAR2(250 CHAR) NOT NULL,
27     CONSTRAINT PK_ORAS_ID PRIMARY KEY(id),
28     CONSTRAINT CK_Judet CHECK(judet IN (
29         'Alba', 'Arad', 'Argeş',
30         'Bacău', 'Bihor', 'Bistriţa-Năsăud',
31         'Călăraşi', 'Caraş-Severin', 'Cluj', 'Constanţa',
32         'Dâmboviţa', 'Dolj', 'Galaţi', 'Giurgiu', 'Gorj',
33         'Harghita', 'Hunedoara', 'Ialomiţa', 'Iaşi',
34         'Maramureş', 'Mehedinţi', 'Mureş',
35         'Olt', 'Prahova', 'Sălaj',
36         'Sibiu', 'Suceava',
37         'Tulcea', 'Vaslui', 'Vâlcea', 'Vrancea'
38     )),
39     CONSTRAINT UK_Locatie UNIQUE(nume, judet)
40 );

```

Table created.

0.07 seconds

- Creare tabele **STATIE, CONDUCTOR, CALATORIE, PROGRAM**

```

41
42 CREATE TABLE STATIE (
43     id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
44     oras NUMBER NOT NULL,
45     restaurant NUMBER(1) DEFAULT 0,
46     CONSTRAINT FK_ID_Oras FOREIGN KEY (oras) REFERENCES ORAS(id),
47     CONSTRAINT PK_STATIE_ID PRIMARY KEY(id)
48 );
49
50 CREATE TABLE CONDUCTOR (
51     id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
52     nume VARCHAR2(250 CHAR) NOT NULL,
53     prenume VARCHAR2(250 CHAR) NOT NULL,
54     nr_tel VARCHAR2(50 CHAR),
55     nr_calatori NUMBER DEFAULT 0,
56     CONSTRAINT PK_CONDUCTOR_ID PRIMARY KEY(id)
57 );
58
59 CREATE TABLE CALATORIE (
60     id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
61     "start" NUMBER NOT NULL,
62     finish NUMBER NOT NULL,
63     durata NUMBER NOT NULL,
64     pret NUMBER NOT NULL,
65     CONSTRAINT PK_CALATORIE_ID PRIMARY KEY(id),
66     CONSTRAINT FK_Start FOREIGN KEY ("start") REFERENCES STATIE(id),
67     CONSTRAINT FK_Finish FOREIGN KEY (finish) REFERENCES STATIE(id),
68     CONSTRAINT CK_Pret CHECK(pret > 0),
69     CONSTRAINT CK_Durata CHECK(durata > 0)
70 );
71
72
73 CREATE TABLE PROGRAM (
74     id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
75     calatorie NUMBER NOT NULL,
76     conductor NUMBER NOT NULL,
77     tren NUMBER NOT NULL,
78     plecare DATE NOT NULL,
79     CONSTRAINT PK_ID1 PRIMARY KEY(id),
80     CONSTRAINT FK_Calatorie FOREIGN KEY (calatorie) REFERENCES CALATORIE(id),
81     CONSTRAINT FK_Conductor FOREIGN KEY (conductor) REFERENCES Conductor(id),
82     CONSTRAINT FK_Tren FOREIGN KEY (tren) REFERENCES Tren(id)
83 );
84

```

5) Confirmarea existentei tabelelor create prin interogarea vederilor din dictionarul datelor; vizualizarea structurii acestora si a constrangerilor aferente

```
84  ---vederea structuri tabelor-----
85  CREATE OR REPLACE VIEW structura_tabele AS
86  SELECT
87      table_name,
88      column_name,
89      data_type,
90      data_length,
91      nullable,
92      data_default
93  FROM
94      all_tab_columns
95  WHERE
96      table_name IN ('TREN', 'ORAS', 'STATIE', 'CONDUCTOR', 'CALATORIE', 'PROGRAM');
97
98  -----vederea constrangerilor tabelor-----
99  CREATE OR REPLACE VIEW constrangeri_tabele AS
100 SELECT
101     table_name,
102     constraint_name,
103     constraint_type,
104     search_condition
105 FROM
106     all_constraints
107 WHERE
108     table_name IN ('TREN', 'ORAS', 'STATIE', 'CONDUCTOR', 'CALATORIE', 'PROGRAM');
109
```



```

110 CREATE OR REPLACE VIEW structura AS
111 SELECT
112     st.table_name,
113     st.column_name,
114     st.data_type,
115     st.data_length,
116     st.nullable,
117     st.data_default
118 FROM
119     structura_tabele st;
120
121 CREATE OR REPLACE VIEW constrangeri AS
122 SELECT
123     ct.table_name,
124     ct.constraint_name,
125     ct.constraint_type,
126     ct.search_condition
127 FROM
128     constrangeri_tabele ct;
129
130 -- Verifica structura tabelelor
131 SELECT * FROM structura_tabele;
132
133 -- Verifica constrangerile tabelelor
134 SELECT * FROM constrangeri_tabele;
135

```

Vederea structurii tabelului (**structura_tabele**) este creată pentru a afișa informații despre coloanele tabelelor specificate (**TREN, ORAS, STATIE, CONDUCTOR, CALATORIE, PROGRAM**).

Aceasta folosește dicționarul de date **all_tab_columns** pentru a obține detalii despre coloanele fiecărei tabele, cum ar fi numele coloanei, tipul de date, lungimea datelor, dacă este nulabilă și valoarea implicită.

Vederea constrângerilor tabelului (**constrangeri_tabele**) este creată pentru a afișa informații despre constrângerile fiecărei tabele. Aceasta utilizează dicționarul de date **all_constraints** pentru a obține detalii despre constrângerile precum numele constrângerii, tipul constrângerii și condiția de căutare (în cazul cheilor de tip **CHECK**).

Ulterior, sunt create două vederi finale (**structura** și **constrangeri**) care extrag doar coloanele relevante din vederile anterioare pentru a simplifica afișarea informațiilor.

TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_LENGTH	NULLABLE	DATA_DEFAULT
PROGRAM	ID	NUMBER	22	N	"WKSP_GRATIELA2963"."ISEQ\$\$_235160502".nextval
PROGRAM	CALATORIE	NUMBER	22	N	-
PROGRAM	CONDUCTOR	NUMBER	22	N	-
PROGRAM	TREN	NUMBER	22	N	-
PROGRAM	PLECARE	DATE	7	N	-
CALATORIE	ID	NUMBER	22	N	"WKSP_GRATIELA2963"."ISEQ\$\$_235160239".nextval
CALATORIE	start	NUMBER	22	N	-
CALATORIE	FINISH	NUMBER	22	N	-
CALATORIE	DURATA	NUMBER	22	N	-
CALATORIE	PRET	NUMBER	22	N	-
More than 10 rows available. Increase rows selector to view more rows.					

TABLE_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
PROGRAM	FK_CALATORIE	R	-
PROGRAM	FK_CONDUCTOR	R	-
PROGRAM	FK_TREN	R	-
CALATORIE	FK_START	R	-
CALATORIE	FK_FINISH	R	-
STATIE	FK_ID_ORAS	R	-
PROGRAM	SYS_C00152287664	C	"ID" IS NOT NULL
PROGRAM	SYS_C00152287665	C	"CALATORIE" IS NOT NULL
PROGRAM	SYS_C00152287666	C	"CONDUCTOR" IS NOT NULL
PROGRAM	SYS_C00152287667	C	"TREN" IS NOT NULL
More than 10 rows available. Increase rows selector to view more rows.			

6)Obiecte ale bazei de date, altele de cat tabele

- VEDERI

```

136
137 CREATE OR REPLACE VIEW numar_coloane AS
138 SELECT
139     table_name,
140     COUNT(column_name) AS numar_coloane
141 FROM
142     all_tab_columns
143 WHERE
144     table_name IN ('TREN', 'ORAS', 'STATIE', 'CONDUCTOR', 'CALATORIE', 'PROGRAM')
145 GROUP BY
146     table_name;
147
148
149 SELECT * FROM numar_coloane;
150

```

TABLE_NAME	NUMAR_COLOANE
CONDUCTOR	5
PROGRAM	5
CALATORIE	5
ORAS	3
TREN	4
STATIE	3

6 rows returned in 0.45 seconds [Download](#)

- Secvențe și confirmarea existenței acestora

```
-- Secvența pentru TREN
CREATE SEQUENCE seq_tren START WITH 1 INCREMENT BY 1;

-- Secvența pentru ORAS
CREATE SEQUENCE seq_oras START WITH 1 INCREMENT BY 1;

-- Secvența pentru STATIE
CREATE SEQUENCE seq_statie START WITH 1 INCREMENT BY 1;

-- Verificare existență pentru TREN
SELECT * FROM all_sequences WHERE sequence_name = 'SEQ_TREN';

-- Verificare existență pentru ORAS
SELECT * FROM all_sequences WHERE sequence_name = 'SEQ_ORAS';

-- Verificare existență pentru STATIE
SELECT * FROM all_sequences WHERE sequence_name = 'SEQ_STATIE';
```

7) Adăugarea, modificarea, ștergerea, îmbinarea (merge), selecție

```
-- Inserare în tabelul TREN
INSERT INTO TREN(achizitie, tip)
VALUES(TO_DATE('01-02-2024', 'DD-MM-YYYY'), 'marfar');

INSERT INTO TREN(achizitie, tip)
VALUES(TO_DATE('02-02-2024', 'DD-MM-YYYY'), 'pasageri');

INSERT INTO TREN(achizitie, tip)
VALUES(TO_DATE('03-02-2024', 'DD-MM-YYYY'), 'marfar');

INSERT INTO TREN(achizitie, tip)
VALUES(TO_DATE('10-05-2024', 'DD-MM-YYYY'), 'pasageri');

SELECT * FROM TREN;
```

ID	ACHIZITIE	AVAILABLE	TIP
3	02/03/2024	0	marfar
21	05/10/2024	0	pasageri
2	02/02/2024	0	pasageri
1	02/01/2024	0	marfar

4 rows returned in 0.02 seconds [Download](#)

```
-- Update în tabelul TREN
UPDATE TREN SET available= 1 WHERE tip = 'marfar';

-- Ștergere din TREN
DELETE FROM TREN WHERE ID = 2;

SELECT * FROM TREN;
```

ID	ACHIZITIE	AVAILABLE	TIP
3	02/03/2024	1	marfar
21	05/10/2024	0	pasageri
1	02/01/2024	1	marfar

3 rows returned in 0.01 secondsDownload

```
-- Inserare în ORAS
INSERT INTO ORAS (nume, judet) VALUES ('Berat', 'Alba');
INSERT INTO ORAS (nume, judet) VALUES ('Mirau', 'Alba');
INSERT INTO ORAS (nume, judet) VALUES ('Azuga', 'Prahova');
SELECT * FROM ORAS;
```

ID	NUME	JUDET
1	Berat	Alba
2	Mirau	Alba
3	Azuga	Prahova

3 rows returned in 0.01 secondsDownload

```
INSERT INTO CALATORIE ("start", finish, durata, pret)
VALUES (1, 3, 120, 50.00);

INSERT INTO CALATORIE ("start", finish, durata, pret)
VALUES (2, 1, 90, 30.00);

SELECT * FROM CALATORIE;
```

```
-- Modificare
UPDATE TREN SET tip = 'pasageri' WHERE ID = 1;

-- Modificare
UPDATE TREN SET achizitie = TO_DATE('15-02-2024', 'DD-MM-YYYY') WHERE ID = 3;

-- Modificare
UPDATE ORAS SET nume = 'Blaj' WHERE ID = 1;

-- Modificare
UPDATE ORAS SET judet = 'Sibiu' WHERE ID = 2;
```

ID	ACHIZITIE	AVAILABLE	TIP
3	02/15/2024	1	marfar
21	05/10/2024	0	pasageri
1	02/01/2024	1	pasageri

ID	NUME	JUDET
1	Blaj	Alba
2	Mirau	Sibiu
3	Azuga	Prahova

3 rows returned in 0.00 secondsDownload

```
-- Selecție
SELECT * FROM CALATORIE WHERE durata > 100;

-- Selecție
SELECT * FROM CALATORIE WHERE pret BETWEEN 30 AND 40;

INSERT INTO CALATORIE ("start", finish, durata, pret)
VALUES (1, 3, 120, 35.00);

INSERT INTO CALATORIE ("start", finish, durata, pret)
VALUES (2, 1, 90, 30.00);

INSERT INTO CALATORIE ("start", finish, durata, pret)
VALUES (3, 2, 150, 40.00);
```

ID	start	FINISH	DURATA	PRET
21	1	3	120	35
23	3	2	150	40

ID	start	FINISH	DURATA	PRET
22	2	1	90	30
21	1	3	120	35
23	3	2	150	40

8) Selectia datelor - minim 5 interogari complexe ce pot fi incluse si in definitia unor viewuri, care contin conditii/clauze complexe.

Tabelul CONDUCTOR

ID	NUME	PRENUME	NR_TEL	NR_CALATORI
22	Dumitrescu	Elena	0720555666	5
1	Popescu	Ion	0720123456	15
2	Popescu	Ana	0720111222	8
21	Ionescu	Mihai	0720333444	12
23	Radu	Ionel	0720777888	12
3	Georgescu	Ionel	0720999000	12

1) Se vor afișa doar persoanele a căror nume de familie începe cu litera P sau I și a un număr de călătorii mai mare decât 9.

```
SELECT
    id,
    nume,
    prenume,
    nr_tel,
    nr_calatori
FROM
    CONDUCTOR
WHERE
    substr(nume, 1, 1) IN ('P', 'I')
    AND nr_calatori > 9;
```

ID	NUME	PRENUME	NR_TEL	NR_CALATORII
1	Popescu	Ion	0720123456	15
21	Ionescu	Mihai	0720333444	12

2) Se vor obține detalii despre călătoriile și orașele aferente

```
-- Subinterogare complexă pentru a obține detalii despre călătoriile și orașele aferente---
SELECT
  c.id AS calatorie_id,
  c."start" AS start_calatorie,
  c.finish AS finish_calatorie,
  c.durata AS durata_calatorie,
  c.pret AS pret_calatorie,
  (
    SELECT
      o1.nume || ', ' || o1.judet
    FROM
      ORAS o1
    WHERE
      o1.id = c."start"
  ) AS oras_start,
  (
    SELECT
      o2.nume || ', ' || o2.judet
    FROM
      ORAS o2
    WHERE
      o2.id = c.finish
  ) AS oras_destinatie
FROM
  CALATORIE c;
```

CALATORIE_ID	START_CALATORIE	FINISH_CALATORIE	DURATA_CALATORIE	PRET_CALATORIE	ORAS_START	ORAS_DESTINATIE
22	2	1	90	30	Mirau, Sibiu	Blaj, Alba
21	1	3	120	35	Blaj, Alba	Azuga, Prahova
23	3	2	150	40	Azuga, Prahova	Mirau, Sibiu
3	1	3	120	50	Blaj, Alba	Azuga, Prahova

3) View pentru a afișa toate călătoriile care au o durată mai mare de 100 de minute și un preț mai mic de 50 de unități monetare

```
CREATE OR REPLACE VIEW calatorii_durate_pret AS
SELECT *
FROM CALATORIE
WHERE durata > 100 AND pret < 50.00;

-- Verificare view
SELECT * FROM calatorii_durate_pret;
```

ID	start	FINISH	DURATA	PRET
21	1	3	120	35
23	3	2	150	40

4) View pentru a afișa toate orașele din județul "Alba":

```
CREATE OR REPLACE VIEW orase_alba AS
SELECT *
FROM ORAS
WHERE judet = 'Alba';

-- Verificare view
SELECT * FROM orase_alba;
```

ID	NUME	JUDET
1	Blaj	Alba

1 rows returned in 0.01 seconds [Download](#)

5) Se vor afișa câte trenuri sunt de pasageri și câte de marfă.

```
SELECT
(
SELECT COUNT(*)
FROM TREN
WHERE tip = 'pasageri'
) AS numar_trenuri_pasageri,
(
SELECT COUNT(*)
FROM TREN
WHERE tip = 'marfar'
) AS numar_trenuri_marfar
FROM DUAL;
```

NUMAR_TRENURI_PASAGERI	NUMAR_TRENURI_MARFAR
2	2

6) Raport privind numarul total de călătorii pentru fiecare tip de tren

Nota s-au realizat inserari în prealabil pentru a se observa un rezultat concret

```
SELECT
T.tip AS tip_tren,
COUNT(C.id) AS numar_calatorii
FROM
TREN T
LEFT JOIN PROGRAM P ON T.id = P.tren
LEFT JOIN CALATORIE C ON P.calatorie = C.id
GROUP BY
T.tip;
```

TIP_TREN	NUMAR_CALATORII
marfar	2
pasageri	4

2 rows returned in 0.00 seconds [Download](#)

7) FULL OUTER JOIN între tabelele CONDUCTOR și Program

Afișează o listă cu toți conducătorii (CONDUCTOR) și programele atribuite (PROGRAM), inclusiv cazurile în care nu există o potrivire.

```
SELECT
    CO.id AS conductor_id,
    CO.nume,
    CO.prenume,
    P.id AS program_id,
    P.plecare
FROM
    CONDUCTOR CO
FULL OUTER JOIN PROGRAM P ON CO.id = P.conductor;
```

CONDUCTOR_ID	NUME	PRENUME	PROGRAM_ID	PLECARE
23	Radu	Ionel	24	02/01/2024
21	Ionescu	Mihai	45	10/01/2024
21	Ionescu	Mihai	23	02/01/2024
23	Radu	Ionel	26	12/12/2024
3	Georgescu	Ionel	25	02/01/2024
3	Georgescu	Ionel	46	09/06/2024
1	Popescu	Ion	-	-
2	Popescu	Ana	-	-
22	Dumitrescu	Elena	-	-

8) Left Join între tabelele ORAS și STATIE:

Afișează o listă de toate orașele (ORAS) și stațiile asociate (STATIE)

```
SELECT
    O.nume AS nume_oras,
    O.judet AS judet_oras,
    S.id AS id_statie,
    S.restaurant
FROM
    ORAS O
LEFT JOIN STATIE S ON O.id = S.oras
WHERE
    S.id IS NOT NULL AND S.restaurant IS NOT NULL;
```


NUME_ORAS	JUDET_ORAS	ID_STATIE	RESTAURANT
Blaj	Alba	21	0
Blaj	Alba	1	1
Mirau	Sibiu	2	1
Azuga	Prahova	3	0

4 rows returned in 0.01 seconds [Download](#)

Această interogare utilizează o clauză WHERE care exclude rândurile în care atât S.id cât și S.restaurant sunt nule din rezultatul final. Prin urmare, doar acele rânduri care au corespondență în ambele tabele (ORAS și STATIE) și nu au valori nule pentru S.id și S.restaurant vor fi incluse în rezultat.

9) Limitarea numarului de linii returnate de o interogare folosind clauzele

OFFSET si FETCH

```
SELECT
    T.id AS tren_id,
    T.achizitie,
    T.tip,
    C.id AS calatorie_id,
    C."start" AS start_calatorie,
    C.finish AS finish_calatorie,
    C.durata AS durata_calatorie,
    C.pret AS pret_calatorie,
    P.plecare,
    CO.numa AS nume_conductor,
    CO.prenume AS prenume_conductor
FROM
    TREN T
    LEFT JOIN PROGRAM P ON T.id = P.tren
    LEFT JOIN CALATORIE C ON P.calatorie = C.id
    LEFT JOIN CONDUCTOR CO ON P.conductor = CO.id
ORDER BY
    T.id, P.plecare;
```

```
-- Use OFFSET and FETCH to get rows 11 to 20
SELECT
    T.id AS tren_id,
    T.achizitie,
    T.tip,
    C.id AS calatorie_id,
    C."start" AS start_calatorie,
    C.finish AS finish_calatorie,
    C.durata AS durata_calatorie,
    C.pret AS pret_calatorie,
    P.plecare,
    CO.numa AS nume_conductor,
    CO.prenume AS prenume_conductor
FROM
    TREN T
    LEFT JOIN PROGRAM P ON T.id = P.tren
    LEFT JOIN CALATORIE C ON P.calatorie = C.id
    LEFT JOIN CONDUCTOR CO ON P.conductor = CO.id
ORDER BY
    T.id, P.plecare
OFFSET 5 ROWS FETCH NEXT 10 ROWS ONLY;
```

TREN_ID	ACHIZITIE	TIP	CALATORIE_ID	START_CALATORIE	FINISH_CALATORIE	DURATA_CALATORIE	PRET_CALATORIE	PLECARE	NUME_CONDUCTOR	PRENUME_CONDUCTOR
1	02/01/2024	pasageri	25	5	2	150	40	02/01/2024	Georgescu	Ionel
1	02/01/2024	pasageri	25	5	2	150	40	09/06/2024	Georgescu	Ionel
3	02/15/2024	marfar	-	-	-	-	-	-	-	-
4	02/01/2024	marfar	22	2	1	90	30	02/01/2024	Ionescu	Mihai
4	02/01/2024	marfar	22	2	1	90	30	10/01/2024	Ionescu	Mihai
21	05/10/2024	pasageri	-	-	-	-	-	-	-	-
41	02/01/2024	marfar	-	-	-	-	-	-	-	-
42	02/02/2024	pasageri	41	1	3	120	50	02/01/2024	Radu	Ionel
42	02/02/2024	pasageri	41	1	3	120	50	12/12/2024	Radu	Ionel

TREN_ID	ACHIZITIE	TIP	CALATORIE_ID	START_CALATORIE	FINISH_CALATORIE	DURATA_CALATORIE	PRET_CALATORIE	PLECARE	NUME_CONDUCTOR	PRENUME_CONDUCTOR
21	05/10/2024	pasageri	-	-	-	-	-	-	-	-
41	02/01/2024	marfar	-	-	-	-	-	-	-	-
42	02/02/2024	pasageri	41	1	3	120	50	02/01/2024	Radu	Ionel
42	02/02/2024	pasageri	41	1	3	120	50	12/12/2024	Radu	Ionel

Prima Interogare (fără OFFSET și FETCH):

Interogarea principală obține informații despre trenuri, călătorii, program și conducători.

Datele sunt ordonate după T.id (ID-ul trenului) și P.plecare (data de plecare din program).

Aceasta returnează întregul set de rezultate, fără limitare.

A Doua Interogare (cu OFFSET și FETCH):

Această interogare utilizează clauza OFFSET 5 ROWS FETCH NEXT 10 ROWS ONLY.

OFFSET 5 ROWS indică să se sara primele 5 rânduri din rezultat.

FETCH NEXT 10 ROWS ONLY indică să se recupereze următoarele 10 rânduri din rezultat .

În esență, această interogare returnează rândurile 6 până la 9 din setul de rezultate obținut în prima interogare.