

1. Introducere

1.1. Descrierea temei alese

Tema proiectului este crearea unei baze de date pentru o școală. Scopul acestei baze de date este de a gestiona informațiile referitoare la părinți, elevi, personal administrativ, personal didactic, personal auxiliar, profiluri, orare, materii, săli și cataloage. Această temă a fost aleasă datorită nevoii de a gestiona informațiile referitoare la școală. O bază de date bine proiectată poate ajuta la simplificarea proceselor administrative.

1.2. Obiective propuse

- crearea unei baze de date bine structurate pentru gestionarea informațiilor școlare;
- asigurarea integrității datelor prin utilizarea constrângерilor și de tipuri de date adecvate;
- crearea unor interogări eficiente pentru extragerea datelor relevante.

1.3. Structura proiectului

Această documentație este împărțită în mai multe capituloare. Primul capitol este dedicat introducerii, incluzând descrierea temei alese, obiectivele propuse și o prezentare detaliată a structurii lucrării. Al doilea capitol detaliază tehnologiile și instrumentele utilizate în realizarea proiectului. Al treilea capitol descrie implementarea proiectului, cu accent pe proiectarea bazei de date și conceptele privind reprezentarea datelor, iar capitolul al patrulea conține codul SQL utilizat pentru crearea tabelelor, inserarea datelor și interogările.

2. Tehnologii utilizate

2.1. SQL

SQL (Structured Query Language) este un limbaj de programare standardizat, utilizat pentru gestionarea bazelor de date relaționale. SQL permite crearea, modificarea și interogarea bazelor de date, precum și definirea și modificarea structurii tabelelor și a altor obiecte asociate cu baza de date.

SQL a fost inițial dezvoltat la compania IBM de către Donald D. Chamberlin și Raymond F. Boyce în anul 1974. La început, numele era SEQUEL (tot de la Structured Query Language) și avea ca scop manipularea și regăsirea datelor stocate în bazele de date relaționale IBM.

SQL este un limbaj declarativ, ceea ce înseamnă că utilizatorii specifică ceea ce doresc să realizeze, și nu modul în care trebuie să fie realizate aceste operațiuni. Aceasta se bazează pe conceptul de relații, astfel încât datele sunt organizate în tabele care sunt interconectate prin intermediul cheilor primare și a celor străine.

Prin intermediul SQL, se pot realiza o gamă largă de operațiuni, cum ar fi crearea, citirea, actualizarea și ștergerea datelor dintr-o bază de date. De asemenea, SQL permite interogarea și agregarea datelor, sortarea, filtrarea și gruparea acestora într-o varietate de moduri.

Există mai multe dialecte de SQL, dar majoritatea interogărilor sunt portabile între diferite sisteme de gestiune a bazelor de date, deoarece majoritatea producătorilor de baze de date au implementat același standard SQL. Cu toate acestea, există și unele diferențe semnificative între diferitele implementări ale standardului SQL, ceea ce poate duce la probleme de compatibilitate între sistemele de gestiune a bazelor de date.

2.2. Oracle SQL Developer

Oracle SQL Developer este un mediu de dezvoltare integrat (IDE) pentru bazele de date Oracle, care permite utilizatorilor să creeze, să editeze și să gestioneze bazele de date Oracle și să execute interogări SQL.

Acesta oferă o interfață grafică utilizatorului și suportă o varietate de caracteristici, cum ar fi editarea de cod, vizualizarea datelor și modelarea datelor. De asemenea, Oracle SQL Developer include un editor de interogări SQL puternic și versatil, care oferă sugestii de completare și verificare de sintaxă.

2.3. Visual Studio Code

Visual Studio Code (VS Code) este un editor de cod sursă dezvoltat de Microsoft, ideal pentru multiple limbaje, inclusiv SQL, datorită completării automate (IntelliSense) și integrării cu baze de date. Este ușor extensibil, permitând conectarea la Oracle Database și alte sisteme, iar interfața sa prietenoasă, alături de suportul pentru Git, îl fac excelent pentru gestionarea și modificarea codului. În acest proiect, VS Code a fost utilizat pentru editarea fișierelor SQL, ulterior salvate și executate în Oracle SQL Developer, oferind o metodă rapidă și eficientă de lucru.

3. Realizarea proiectului

3.1. Modul de descriere a datelor

Baza de date este structurată pentru a gestiona informațiile unei școli. Aceasta include 11 tabele: ȘCOALĂ, ELEVI, PĂRINȚI, PERSONAL ADMINISTRATIV, PERSONAL DIDACTIC, PERSONAL AUXILIAR, PROFILURI, ORARE, MATERII, SĂLI, CATALOAGE.

Fiecare tabel are coloane cu tipuri de date specifice, iar cheile primare și cheile străine sunt definite pentru a asigura integritatea referențială a datelor în baza de date. Atributele fiecărei entități au denumiri clare și unice, iar tipurile de date atribuite au fost alese în funcție de natura fiecărui câmp (de exemplu, NUMBER, VARCHAR2).

Parametrul „NOT NULL” este utilizat pentru a asigura că un câmp de text nu poate rămâne necompletat, în timp ce parametrul „PRIMARY KEY” desemnează un atribut unic care identifică în mod clar fiecare înregistrare din tabel. Cheile primare sunt esențiale, deoarece permit crearea de indici pentru gestionarea mai rapidă a datelor.

Atributele care reprezintă chei străine sunt definite în tabelele secundare, menținând o corespondență exactă cu valorile din tabelele primare asociate. După definirea lor, aceste atrbute sunt declarate ca „FOREIGN KEY” pentru a garanta integritatea relațiilor dintre tabele.

3.2. Respectarea formelor normale

Pentru respectarea formei normale și evitarea redundanțelor, fiecare tabel din baza de date a fost proiectat să reprezinte un set specific de informații legate de școală, fără relații redundante sau directe de tip many-to-many. De exemplu, tabela CATALOAGE este utilizată pentru a stoca informații despre cataloagele școlare, precum numărul materiilor, notele, absențele sau învoiri, dar nu are o relație directă cu elevii sau materiile. Această abordare ajută la păstrarea unei structuri bine definite și organizate, simplificând gestionarea datelor în sistemul educațional.

3.3. Diagrama entitate-asociere (EA)

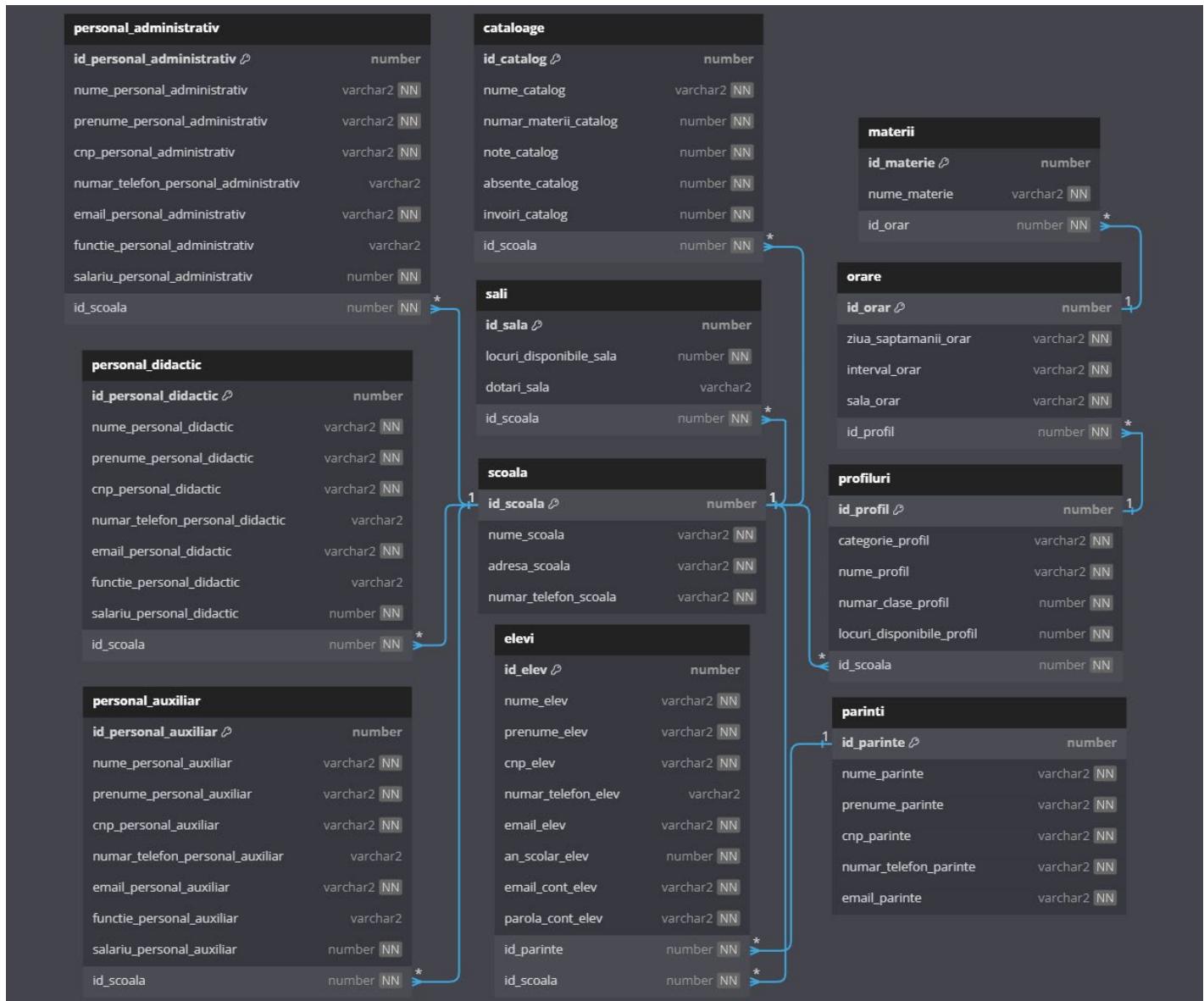


Figura 1. Diagrama entitate-asociere

3.4. Descrierea diagramei, a relațiilor, a entităților și a atributelor

Diagrama bazei de date include 11 tabele: ȘCOALĂ, ELEVI, PĂRINȚI, PERSONAL ADMINISTRATIV, PERSONAL DIDACTIC, PERSONAL AUXILIAR, PROFILURI, ORARE, MATERII, SĂLI, CATALOAGE, care sunt interconectate prin chei primare și străine. Tabelele descriu diferite aspecte specifice gestionării datelor școlare.

Tabela ȘCOALĂ conține informații despre instituția școlară, precum numele școlii, adresa și numărul de telefon al acesteia. Cheia primară este id_scoala. Această tabelă este punctul central al bazei de date, toate celelalte entități fiind legate de ea pentru a arăta apartenența lor la o anumită școală.

Tabela ELEVI este folosită pentru a stoca informații despre elevi, cum ar fi numele, prenumele, CNP-ul, anul școlar, numărul de telefon și emailul elevului. Cheia primară este id_elev, iar id_scoala este o cheie străină care leagă elevii de școala lor.

Tabela PĂRINȚI conține informații despre părinți, inclusiv nume, prenume, număr de telefon, CNP-ul și email-ul acestora. Cheia primară este id_parinte, iar id_elev este o cheie străină care leagă părinții de elevii lor.

Tabela PERSONAL_ADMINISTRATIV include informații despre personalul administrativ al școlii, cum ar fi numele, prenumele, CNP-ul numărul de telefon și salariul. Cheia primară este id_personal_administrativ, iar id_scoala este o cheie străină care conectează personalul la școala în care lucrează.

Tabela PERSONAL DIDACTIC stochează informații despre profesorii din școală, cum ar fi numele, prenumele, CNP-ul, funcția și salariul acestora. Cheia primară este id_personal_didactic, iar id_scoala este cheia străină care indică școala la care aceștia predau.

Tabela PERSONAL_AUXILIAR conține informații despre personalul auxiliar, cum ar fi numele, prenumele, CNP-ul, salariul și funcția acestora. Cheia primară este id_personal_auxiliar, iar id_scoala este cheia străină care indică școala la care aceștia lucrează.

Tabela PROFILURI este folosită pentru a defini profilurile educaționale disponibile în școală, cum ar fi „Matematică-Informatică” sau „Științe Sociale”. Cheia primară este id_profil.

Tabela ORARE stochează informații despre programul școlar, inclusiv ziua săptămânii, intervalul orar și sălile în care se predau diferite materii. Cheia primară este id_orar, iar id_profil este o cheie străină care indică profilul asociat orarului.

Tabela MATERII stochează informații despre materiile predate, cum ar fi numele materiei. Cheia primară este id_materie, iar id_orar este o cheie străină care leagă materiile de orare.

Tabela SĂLI conține informații despre sălile din școală, cum ar fi locurile disponibile și dotările din acestea. Cheia primară este id_sala, iar id_scoala este o cheie străină care leagă sălile de școala în care sunt localizate.

Tabela CATALOAGE este utilizată pentru a înregistra rezultatele elevilor la diferite materii. Aceasta include numele clasei căruia îi aparține catalogul, notele elevilor, absențele și învoiriile acestora. Cheia primară este id_catalog, iar id_scoala este cheia străină care indică școala căreia aparțin cataloagele.

3.5. Modelul relațional al bazei de date

Modelul relațional reprezintă o metodă de organizare și gestionare a datelor într-o bază de date. Organizarea datelor se face în tabele cu rânduri și coloane, iar definirea relațiilor dintre acestea se face prin chei primare și străine.

În implementarea modelului relațional, atributele sunt utilizate ca și chei logice. Baza de date este reprezentată ca un ansamblu de tabele care conțin datele și legăturile dintre acestea. Acest model se bazează pe noțiunea matematică de relație, care reprezintă o mulțime de entități de același tip.

Într-un model relațional, fiecare tabel conține date referitoare la un anumit subiect, iar fiecare rând din tabel reprezintă o instanță a acestei informații. Coloanele din tabel reprezintă atributele entității și descriu caracteristicile acesteia. De exemplu, entitatea ELEV are atributele id_elev, nume_elev, prenume_elev, cnp_elev, numar_telefon_elev, email_elev, an_scolar_elev, email_cont_elev, parola_cont_elev și cheile străine id_parinte, id_scoala.

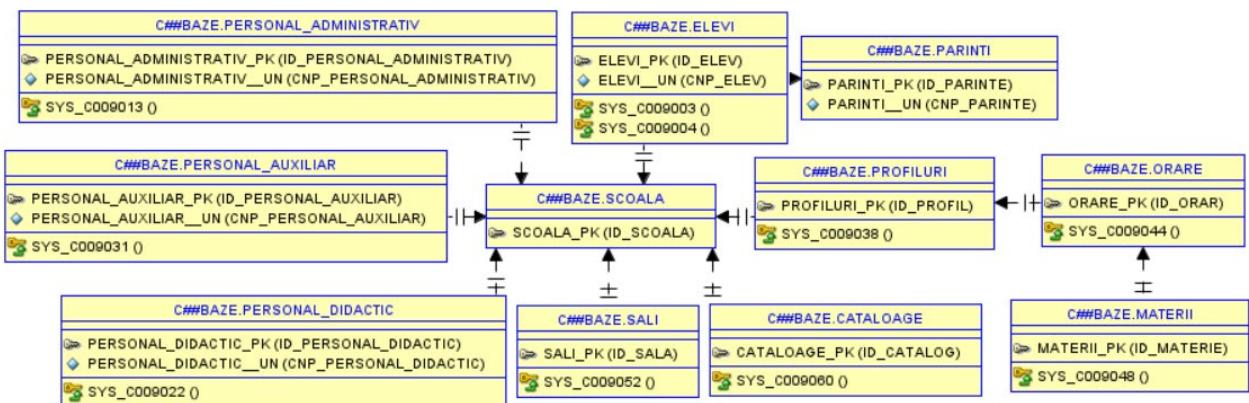


Figura 2. Modelul relațional al bazei de date

4. Codul SQL

4.1. Crearea tabelelor

Crearea tabelelor se face după cum urmează:

```
CREATE TABLE scoala (
    id_scoala NUMBER PRIMARY KEY,
    nume_scoala VARCHAR2(100) NOT NULL,
    adresa_scoala VARCHAR2(150) NOT NULL,
    numar_telefon_scoala VARCHAR2(15) NOT NULL
);
```

| COLUMN_NAME | DATA_TYPE | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|------------------------|--------------------|----------|--------------|-----------|----------|
| 1 ID_SCOALA | NUMBER | No | (null) | 1 | (null) |
| 2 NUME_SCOALA | VARCHAR2(100 BYTE) | No | (null) | 2 | (null) |
| 3 ADRESA_SCOALA | VARCHAR2(150 BYTE) | No | (null) | 3 | (null) |
| 4 NUMAR_TELEFON_SCOALA | VARCHAR2(15 BYTE) | No | (null) | 4 | (null) |

Figura 3. Crearea tăblei SCOALĂ

```
CREATE TABLE parinti (
    id_parinte NUMBER PRIMARY KEY,
    nume_parinte VARCHAR2(50) NOT NULL,
    prenume_parinte VARCHAR2(50) NOT NULL,
    cnp_parinte VARCHAR2(13) NOT NULL UNIQUE,
    numar_telefon_parinte VARCHAR2(15) NOT NULL,
    email_parinte VARCHAR2(100) NOT NULL
);
```

| COLUMN_NAME | DATA_TYPE | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|-------------------------|--------------------|----------|--------------|-----------|----------|
| 1 ID_PARINTE | NUMBER | No | (null) | 1 | (null) |
| 2 NUME_PARINTE | VARCHAR2(50 BYTE) | No | (null) | 2 | (null) |
| 3 PRENUME_PARINTE | VARCHAR2(50 BYTE) | No | (null) | 3 | (null) |
| 4 CNP_PARINTE | VARCHAR2(13 BYTE) | No | (null) | 4 | (null) |
| 5 NUMAR_TELEFON_PARINTE | VARCHAR2(15 BYTE) | No | (null) | 5 | (null) |
| 6 EMAIL_PARINTE | VARCHAR2(100 BYTE) | No | (null) | 6 | (null) |

Figura 4. Crearea tăblei PĂRINȚI


```

CREATE TABLE profiluri (
    id_profil NUMBER PRIMARY KEY,
    categorie_profil VARCHAR2(50) NOT NULL,
    nume_profil VARCHAR2(50) NOT NULL,
    numar_clase_profil NUMBER NOT NULL,
    locuri_disponibile_profil NUMBER NOT NULL,
    id_scoala NUMBER NOT NULL,
    FOREIGN KEY (id_scoala) REFERENCES scoala(id_scoala)
);

```

| COLUMN_NAME | DATA_TYPE | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|-----------------------------|-------------------|----------|--------------|-----------|----------|
| 1 ID_PROFIL | NUMBER | No | (null) | 1 | (null) |
| 2 CATEGORIE_PROFIL | VARCHAR2(50 BYTE) | No | (null) | 2 | (null) |
| 3 NUME_PROFIL | VARCHAR2(50 BYTE) | No | (null) | 3 | (null) |
| 4 NUMAR_CLASE_PROFIL | NUMBER | No | (null) | 4 | (null) |
| 5 LOCURI_DISPONIBILE_PROFIL | NUMBER | No | (null) | 5 | (null) |
| 6 ID_SCOALA | NUMBER | No | (null) | 6 | (null) |

Figura 9. Crearea tабelei PROFILURI

```

CREATE TABLE orare (
    id_orar NUMBER PRIMARY KEY,
    ziua_saptamanii_orar VARCHAR2(15) NOT NULL,
    interval_orar VARCHAR2(20) NOT NULL,
    sala_orar VARCHAR2(20) NOT NULL,
    id_profil NUMBER NOT NULL,
    FOREIGN KEY (id_profil) REFERENCES profiluri(id_profil)
);

```

| COLUMN_NAME | DATA_TYPE | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|-----------------------|-------------------|----------|--------------|-----------|----------|
| 1 ID_ORAR | NUMBER | No | (null) | 1 | (null) |
| 2 ZIUA_SAPTMANII_ORAR | VARCHAR2(15 BYTE) | No | (null) | 2 | (null) |
| 3 INTERVAL_ORAR | VARCHAR2(20 BYTE) | No | (null) | 3 | (null) |
| 4 SALA_ORAR | VARCHAR2(20 BYTE) | No | (null) | 4 | (null) |
| 5 ID_PROFIL | NUMBER | No | (null) | 5 | (null) |

Figura 10. Crearea tабelei ORARE

```

CREATE TABLE materii (
    id_materie NUMBER PRIMARY KEY,
    nume_materie VARCHAR2(50) NOT NULL,
    id_orar NUMBER NOT NULL,
    FOREIGN KEY (id_orar) REFERENCES orare(id_orar)
);

```

| COLUMN_NAME | DATA_TYPE | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|----------------|-------------------|----------|--------------|-----------|----------|
| 1 ID_MATERIE | NUMBER | No | (null) | 1 | (null) |
| 2 NUME_MATERIE | VARCHAR2(50 BYTE) | No | (null) | 2 | (null) |
| 3 ID_ORAR | NUMBER | No | (null) | 3 | (null) |

Figura 11. Crearea tабelei MATERII

```

CREATE TABLE sali (
    id_sala NUMBER PRIMARY KEY,
    locuri_disponibile_sala NUMBER NOT NULL,
    dotari_sala VARCHAR2(150),
    id_scoala NUMBER NOT NULL,
    FOREIGN KEY (id_scoala) REFERENCES scoala(id_scoala)
);

```

| COLUMN_NAME | DATA_TYPE | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|---------------------------|--------------------|----------|--------------|-----------|----------|
| 1 ID_SALA | NUMBER | No | (null) | 1 | (null) |
| 2 LOCURI_DISPONIBILE_SALA | NUMBER | No | (null) | 2 | (null) |
| 3 DOTARI_SALA | VARCHAR2(150 BYTE) | Yes | (null) | 3 | (null) |
| 4 ID_SCOALA | NUMBER | No | (null) | 4 | (null) |

Figura 12. Crearea tăblei SĂLI

```

CREATE TABLE cataloage (
    id_catalog NUMBER PRIMARY KEY,
    nume_catalog VARCHAR2(50) NOT NULL,
    numar_materii_catalog NUMBER NOT NULL,
    note_catalog NUMBER NOT NULL,
    absente_catalog NUMBER NOT NULL,
    invoiri_catalog NUMBER NOT NULL,
    id_scoala NUMBER NOT NULL,
    FOREIGN KEY (id_scoala) REFERENCES scoala(id_scoala)
);

```

| COLUMN_NAME | DATA_TYPE | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|-------------------------|-------------------|----------|--------------|-----------|----------|
| 1 ID_CATALOG | NUMBER | No | (null) | 1 | (null) |
| 2 NUME_CATALOG | VARCHAR2(50 BYTE) | No | (null) | 2 | (null) |
| 3 NUMAR_MATERII_CATALOG | NUMBER | No | (null) | 3 | (null) |
| 4 NOTE_CATALOG | NUMBER | No | (null) | 4 | (null) |
| 5 ABSENTE_CATALOG | NUMBER | No | (null) | 5 | (null) |
| 6 INVOIRI_CATALOG | NUMBER | No | (null) | 6 | (null) |
| 7 ID_SCOALA | NUMBER | No | (null) | 7 | (null) |

Figura 13. Crearea tăblei CATALOAGE

21. Afisează numele materiilor care se predau în ziua de luni:

```
SELECT m.nume_materie  
FROM materii m  
JOIN orare o ON m.id_orar = o.id_orar  
WHERE o.ziua_saptamanii_orar = 'Luni'
```

| NUME_MATERIE | |
|---------------|------------|
| 1 Matematica | 5 Istorie |
| 2 Informatica | 6 Muzica |
| 3 Pedagogie | 7 Biologie |
| 4 Geografie | 8 Muzica |

Figura 45. Afişarea materiilor care se predau în ziua de luni

22. Afisează salariile tuturor angajaților din categoriile administrative, didactice și auxiliare:

```
SELECT salariu_personal_administrativ AS salariu  
FROM personal_administrativ  
UNION  
SELECT salariu_personal_didactic  
FROM personal_didactic  
UNION  
SELECT salariu_personal_auxiliar  
FROM personal_auxiliar;
```

| SALARIU | 5 | 10 | 15 | 5800 |
|---------|--------|---------|---------|------|
| 1 4500 | 6 4200 | 11 7500 | 16 3300 | |
| 2 5000 | 7 3800 | 12 6800 | 17 3500 | |
| 3 5200 | 8 7000 | 13 6200 | 18 4000 | |
| 4 6000 | 9 6400 | 14 5600 | 19 3000 | |
| | | | 20 6500 | |

Figura 46. Afişarea salariilor angajaților

23. Afisează elevii care nu au număr de telefon înregistrat:

```
SELECT nume_elev, prenume_elev  
FROM elevi  
WHERE numar_telefon_elev IS NULL;
```

| NUME_ELEV | PRENUME_ELEV | 4 Ghinea | Adrian |
|-----------|--------------|-----------|--------|
| 1 Ciolacu | Florin | 5 Florea | Anca |
| 2 Stan | Alex | 6 Grigore | Teodor |
| 3 Stan | Gabriela | 7 Manole | Ioana |

Figura 47. Afişarea elevilor fără un număr de telefon

30. Construiește o propoziție cu detalii despre elevi și școala lor:

```
SELECT 'Elevul ' || e.nume_elev || ' ' || e.prenume_elev || ' este înregistrat la școala ' || s.nume_scoala AS Detalii  
FROM elevi e JOIN scoala s ON e.id_scoala = s.id_scoala;
```

| DETALII |
|--|
| 1 Elevul Buruiana Ana este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 2 Elevul Pavel Bianca este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 3 Elevul Ionescu Maria este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 4 Elevul Ionescu Mihai este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 5 Elevul Iacob George este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 6 Elevul Iacob Bianca este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 7 Elevul Ciolacu Ioana este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 8 Elevul Ciolacu Florin este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 9 Elevul Stan Alex este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 10 Elevul Stan Gabriela este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 11 Elevul Radu Cristian este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 12 Elevul Radu Dana este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 13 Elevul Ioan Dan este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 14 Elevul Ioan Gabriela este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 15 Elevul Petre Elena este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 16 Elevul Petre Radu este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 17 Elevul Tudor Vlad este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 18 Elevul Tudor Alexia este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 19 Elevul Radulescu Ana este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 20 Elevul Radulescu Mihai este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 21 Elevul Ghinea Adrian este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 22 Elevul Mocanu Ema este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 23 Elevul Dinca Rares este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 24 Elevul Lungu Anca este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 25 Elevul Gheorghiu Teodor este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 26 Elevul Dragomir Ioana este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 27 Elevul Avram Mihai este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 28 Elevul Simion Alina este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 29 Elevul Costea Alex este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 30 Elevul Ciobanu Bianca este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 31 Elevul Popa Adrian este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 32 Elevul Dima Ema este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 33 Elevul Toma Rares este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 34 Elevul Florea Anca este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 35 Elevul Grigore Teodor este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 36 Elevul Manole Ioana este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 37 Elevul Morar Mihai este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 38 Elevul Ganea Alina este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 39 Elevul Ghinoiu Alex este înregistrat la școala Colegiul National de Informatica Matei Basarab |
| 40 Elevul Buruiana Andrei este înregistrat la școala Colegiul National de Informatica Matei Basarab |

Figura 54. Afisarea și concatenarea textului

Bibliografie

1. <https://docs.oracle.com/en/database/oracle/oracle-database/21/sqlrf/index.html>
2. https://en.wikipedia.org/wiki/Oracle_SQL_Developer