Proiect Baze de date-
Catalog Online pentru liceu

Cod github: https://github.com/andreiserban123/BD-project

Obiectivul bazei de date:

Baza de date este destinata gestionarii notelor elevilor unui liceu. Platforma permite parintilor, studentilor si profesorilor sa vizualizeze/adauge note in functie de permisiuni si rol.

1. Descrierea tabelelor si atributelor:

SIT_PERSON -> contine datele personale, nume, prenume, data nasterii l, cnp si adresa

SIT_USER -> fiecare user are un email si parola. In spatele fiecarui user se afla un id din SIT_PERSON

SIT_USER_ROLES -> fiecare user are nevoie de cel putin un rol. Exista cazuri in care un profesor poate fi si parinte => un user are 2 roluri

SIT_GRADE -> nomenclator de clase (9-12)

SIT_PARENTS -> tabela care uneste parintele si copilul

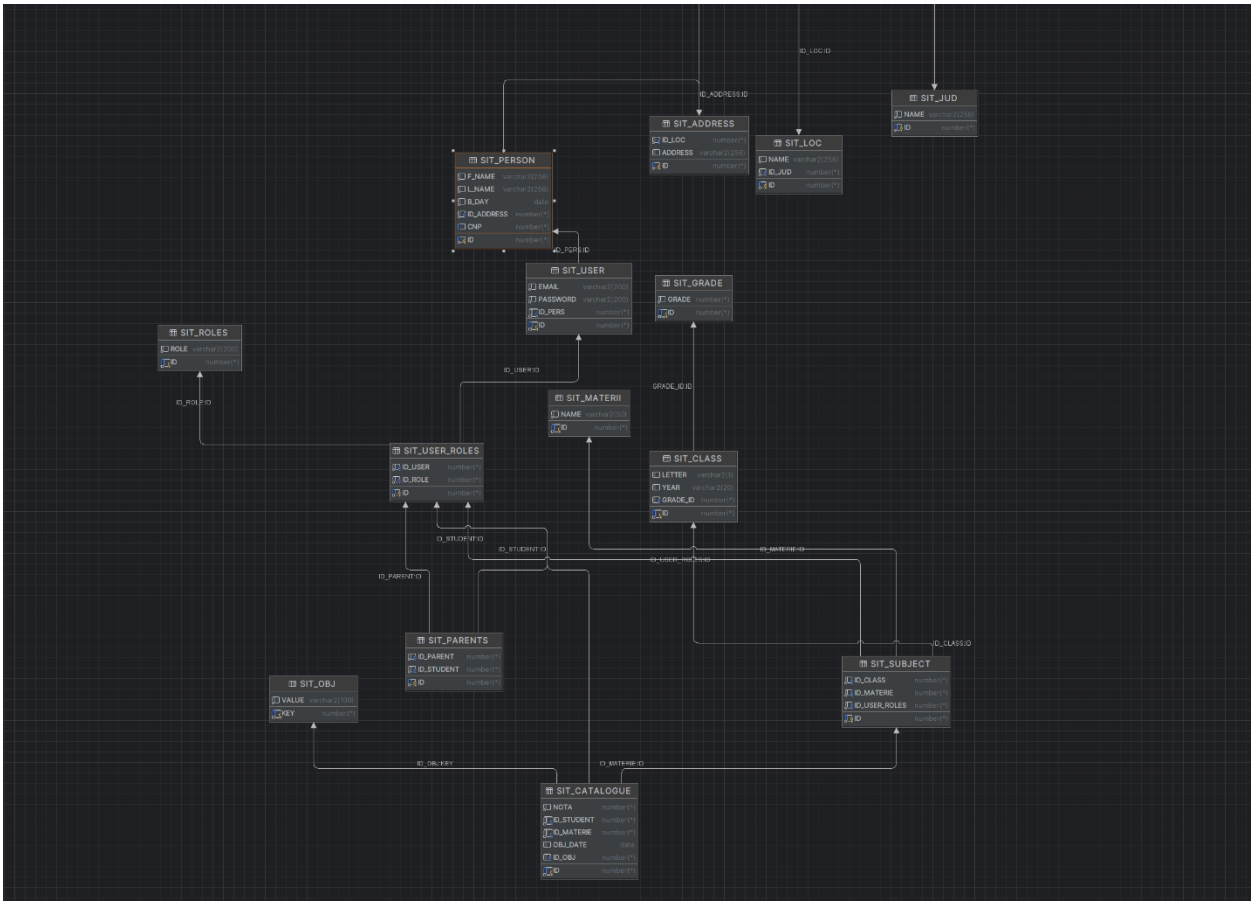
SIT_CLASS -> anul, litera clasei (‘A’,’B’,’C’, etc.)

SIT_SUBJECT -> id_class (id-ul clasei la care se face materia), id_materia(FK catre un nomenclator de materii), id_user_roles=> id profesorului

SIT_OBJ -> o tabela de tip key value pair in care value poate fi ‘N’, ‘T’, ‘ABS’ reprezentand nota , teza si abs

SIT_CATALOGUE -> avem nota(nu este obligatorie in caz de ‘ABS’) , id_student, id_materie, obj_date(data la care s-a luat nota/abs/teza), id_obj (care poate fi ‘N’, ‘T’, ‘ABS’)

2. Schema bazei de date



3. Creearea tabelelor

```
create table sit_jud
(
    id   number primary key not null,
    name varchar2(256)      not null
);
/

create table sit_loc
(
    id     number primary key not null,
    name   varchar2(256)      not null,
    id_jud number             not null,
    FOREIGN KEY (id_jud) REFERENCES SIT_JUD (id)
);
/

create table sit_address
(
    id      number primary key not null,
    id_loc  number             not null,
    address varchar2(256),
    FOREIGN KEY (id_loc) references sit_loc (id)
);
/

create table sit_person
(
    id         number primary key not null,
    f_name     varchar2(256)      not null,
    l_name     varchar2(256)      not null,
    b_day      date               not null,
    id_address number             not null,
    cnp        number,
    FOREIGN KEY (id_address) references SIT_ADDRESS (id)
);
```

```
create table SIT_USER
(
    id       number primary key not null,
    email    varchar2(200)      not null,
    password varchar2(200)      not null,
    id_pers  number             not null,
    FOREIGN KEY (id_pers) references SIT_PERSON (id)
);

create table sit_roles
(
    id   number primary key not null,
    role varchar2(200)      not null
);
/

create table sit_user_roles
(
    id      number primary key not null,
    id_user number             not null,
    id_role number             not null,
    FOREIGN KEY (id_user) references SIT_USER (id),
    FOREIGN KEY (id_role) references SIT_ROLES (id)
);
/

create table SIT_PARENTS
(
    id         number primary key not null,
    id_parent  number             not null,
    id_student number             not null,
    FOREIGN KEY (id_parent) references SIT_USER_ROLES (id),
    FOREIGN KEY (id_student) references SIT_USER_ROLES (id)
);
/
```

```
create table SIT_GRADE
(
    id    number primary key not null,
    grade number             not null
);
/

create table SIT_MATERII
(
    id   number primary key not null,
    name varchar2(50)       not null
);
/
create table SIT_CLASS
(
    id       number primary key not null,
    letter   varchar2(1),
    year     varchar2(20),
    grade_id number,
    FOREIGN KEY (grade_id) references SIT_GRADE (id)
);
/
create table SIT_SUBJECT
(
    id            number primary key not null,
    id_class      number             not null,
    id_materie    number             not null,
    id_user_roles number             not null,
    FOREIGN KEY (id_class) references SIT_CLASS (id),
    FOREIGN KEY (id_materie) references SIT_MATERII (id),
    FOREIGN KEY (id_user_roles) references SIT_USER_ROLES (id)
);
/

create table SIT_OBJ
(
    key   number primary key not null,
    value varchar2(100)      not null
);
```

```
create table SIT_CATALOGUE
(
    id         number primary key not null,
    nota       number,
    id_student number             not null,
    id_materie number             not null,
    obj_date   date,
    id_obj     number,
    FOREIGN KEY (id_student) references SIT_USER_ROLES (id),
    FOREIGN KEY (id_materie) references SIT_SUBJECT (id),
    foreign key (id_obj) references SIT_OBJ (key),
    check ( round(nota) >= 1 and round(nota) <= 10 )
);
```

4. Actualizarea structurii tabelelor si modificarea restrictiilor de integritate

4.1 Alter table – modificare structurii

```
alter table sit_person
add telefon varchar2(10);
```

4.2 Alter Table – modificarea restrictiilor de integritate:

```
ALTER TABLE SIT_PERSON
    ADD CONSTRAINT uniqueCNP UNIQUE (CNP);

alter table sit_user
    add constraint uniqueEmail unique (email);
```

4.3 Drop table:

```
create table SIT_EMAIL
(
    id         number primary key not null,
    email      varchar2(200)      not null,
    id_student number             not null,
    FOREIGN KEY (id_student) references SIT_USER_ROLES (id)
);

drop table sit_email cascade constraints ;
```

5. Adaugarea inregistrarilor

```sql
INSERT INTO sit_jud (id, name)
VALUES (1, 'BUCURESTI');
INSERT INTO sit_jud (id, name)
VALUES (2, 'CLUJ');
INSERT INTO sit_jud (id, name)
VALUES (3, 'TIMIS');
INSERT INTO sit_jud (id, name)
VALUES (4, 'BRASOV');
INSERT INTO sit_jud (id, name)
VALUES (5, 'CONSTANTA');
INSERT INTO sit_jud (id, name)
VALUES (6, 'DOLJ');
INSERT INTO sit_jud (id, name)
VALUES (7, 'GALATI');
INSERT INTO sit_jud (id, name)
VALUES (8, 'IASI');
INSERT INTO sit_jud (id, name)
VALUES (9, 'PRAHOVA');
INSERT INTO sit_jud (id, name)
VALUES (10, 'SIBIU');
INSERT INTO sit_jud (id, name)
VALUES (11, 'Suceava');


INSERT INTO sit_loc (id, name, id_jud)
VALUES (1, 'Sectorul 6', 1);
INSERT INTO sit_loc (id, name, id_jud)
VALUES (2, 'Cluj-Napoca', 2);
INSERT INTO sit_loc (id, name, id_jud)
VALUES (3, 'TIMISOARA', 3);
INSERT INTO sit_loc (id, name, id_jud)
VALUES (4, 'Brasov City', 4);
INSERT INTO sit_loc (id, name, id_jud)
VALUES (5, 'Constanta City', 5);
INSERT INTO sit_loc (id, name, id_jud)
VALUES (6, 'Craiova', 6);
INSERT INTO sit_loc (id, name, id_jud)
VALUES (7, 'Galati', 7);
INSERT INTO sit_loc (id, name, id_jud)
VALUES (8, 'Iasi', 8);
INSERT INTO sit_loc (id, name, id_jud)
VALUES (9, 'Ploiesti', 9);
INSERT INTO sit_loc (id, name, id_jud)
VALUES (10, 'Sibiu', 10);
INSERT INTO sit_loc (id, name, id_jud)
VALUES (11, 'Suceava', 11);
```

```sql
INSERT INTO sit_address (id, id_loc, address)
VALUES (1, 1, 'STR. Chibzuintei');
INSERT INTO sit_address (id, id_loc, address)
VALUES (2, 2, 'STR.  21 Decembrie 1989');
INSERT INTO sit_address (id, id_loc, address)
VALUES (3, 3, 'STR.  Mihai Eminescu');
INSERT INTO sit_address (id, id_loc, address)
VALUES (4, 4, 'STR. Centrului');
INSERT INTO sit_address (id, id_loc, address)
VALUES (5, 5, 'Bd. Tomis');
INSERT INTO sit_address (id, id_loc, address)
VALUES (6, 6, 'STR. 1 Decembrie');
INSERT INTO sit_address (id, id_loc, address)
VALUES (7, 7, 'STR. 1 Mai');
INSERT INTO sit_address (id, id_loc, address)
VALUES (8, 8, 'STR. 2 Mai');
INSERT INTO sit_address (id, id_loc, address)
VALUES (9, 9, 'STR. 3 Mai');
INSERT INTO sit_address (id, id_loc, address)
VALUES (10, 10, 'STR. 4 Mai');
INSERT INTO sit_address (id, id_loc, address)
VALUES (11, 11, 'STR. 5 Mai');



INSERT INTO sit_person (id, f_name, l_name, b_day, id_address, cnp)
VALUES (1, 'ION', 'IONESCU', to_date('01-01-1990', 'dd-mm-yyyy'), 1, 1234567890123);
INSERT INTO sit_person (id, f_name, l_name, b_day, id_address, cnp)
VALUES (2, 'VASILE', 'VASILESCU', to_date('01-02-2003', 'dd-mm-yyyy'), 2, 1234567890123);
INSERT INTO sit_person (id, f_name, l_name, b_day, id_address, cnp)
VALUES (3, 'MARIUS', 'POPESCU', to_date('12-03-2005', 'dd-mm-yyyy'), 3, 1234567890123);
INSERT INTO sit_person (id, f_name, l_name, b_day, id_address, cnp)
VALUES (4, 'ANA', 'ANESCU', to_date('15-04-1995', 'dd-mm-yyyy'), 1, 1234567890123);
INSERT INTO sit_person (id, f_name, l_name, b_day, id_address, cnp)
VALUES (5, 'GEORGE', 'GEORGESCU', to_date('10-08-1988', 'dd-mm-yyyy'), 2, 1234567890123);
INSERT INTO sit_person (id, f_name, l_name, b_day, id_address, cnp)
VALUES (6, 'MARIA', 'MARIAN', to_date('05-05-1992', 'dd-mm-yyyy'), 2, 5234562891323);
INSERT INTO sit_person (id, f_name, l_name, b_day, id_address, cnp)
VALUES (7, 'RADU', 'RADULESCU', to_date('08-08-1985', 'dd-mm-yyyy'), 3, 1234537892123);
INSERT INTO sit_person (id, f_name, l_name, b_day, id_address, cnp)
VALUES (8, 'MIRCEA', 'MIRCEANU', to_date('09-09-1999', 'dd-mm-yyyy'), 4, 1234537890123);
INSERT INTO sit_person (id, f_name, l_name, b_day, id_address, cnp)
VALUES (9, 'ANDREI', 'ANDREESCU', to_date('10-10-2000', 'dd-mm-yyyy'), 5, 1234167890123);
INSERT INTO sit_person (id, f_name, l_name, b_day, id_address, cnp)
VALUES (10, 'MIHAI', 'MIHAILESCU', to_date('11-11-2001', 'dd-mm-yyyy'), 6, 1234667890123);
INSERT INTO sit_person (id, f_name, l_name, b_day, id_address, cnp)
VALUES (11, 'IONEL', 'IONEL', to_date('12-12-2002', 'dd-mm-yyyy'), 7, 12345678912123);
INSERT INTO sit_person (id, f_name, l_name, b_day, id_address, cnp)
VALUES (12, 'IONELA', 'IONELA', to_date('13-01-2003', 'dd-mm-yyyy'), 8, 12345678320123);
```

```sql
INSERT INTO sit_user (id, email, password, id_pers)
VALUES (1, 'ion.ionescu@gmail.com', '123456', 1);
INSERT INTO sit_user (id, email, password, id_pers)
VALUES (2, 'vasile.vasilescu@gmail.com', 'fdsfds', 2);
INSERT INTO sit_user (id, email, password, id_pers)
VALUES (3, 'marius.popescu@gmail.com', 'acoperis', 3);
INSERT INTO sit_user (id, email, password, id_pers)
VALUES (4, 'ana.anescu@gmail.com', 'parola123', 4);
INSERT INTO sit_user (id, email, password, id_pers)
VALUES (5, 'george.georgescu@gmail.com', 'parola456', 5);
INSERT INTO sit_user (id, email, password, id_pers)
VALUES (6, 'maria.marinescu', 'parola789', 6);
INSERT INTO sit_user (id, email, password, id_pers)
VALUES (7, 'radu.radulescu', 'parola101112', 7);
INSERT INTO sit_user (id, email, password, id_pers)
VALUES (8, 'mircea.mirceanu', 'parola131415', 8);
INSERT INTO sit_user (id, email, password, id_pers)
VALUES (9, 'andrei.andreescu', 'parola161718', 9);
INSERT INTO sit_user (id, email, password, id_pers)
VALUES (10, 'mihai.mihailescu', 'parola192021', 10);
INSERT INTO sit_user (id, email, password, id_pers)
VALUES (11, 'ionel.ionel', 'parola222324', 11);
INSERT INTO sit_user (id, email, password, id_pers)
VALUES (12, 'ionela.ionela', 'parola252627', 12);



INSERT INTO sit_roles (id, role)
VALUES (1, 'STUDENT');
INSERT INTO sit_roles (id, role)
VALUES (2, 'PARENT');
INSERT INTO sit_roles (id, role)
VALUES (3, 'TEACHER');
INSERT INTO sit_roles (id, role)
VALUES (4, 'ADMIN');

INSERT INTO sit_user_roles (id, id_user, id_role)
VALUES (1, 1, 3); -- teacher
INSERT INTO sit_user_roles (id, id_user, id_role)
VALUES (2, 2, 1); -- student
INSERT INTO sit_user_roles (id, id_user, id_role)
VALUES (3, 3, 1); -- student
INSERT INTO sit_user_roles (id, id_user, id_role)
VALUES (4, 4, 2); -- parent
INSERT INTO sit_user_roles (id, id_user, id_role)
VALUES (5, 5, 1); -- student
INSERT INTO sit_user_roles (id, id_user, id_role)
VALUES (6, 6, 1); -- student
INSERT INTO sit_user_roles (id, id_user, id_role)
VALUES (7, 7, 1); -- student
INSERT INTO sit_user_roles (id, id_user, id_role)
VALUES (8, 8, 3); -- teacher
INSERT INTO sit_user_roles (id, id_user, id_role)
VALUES (9, 9, 1); -- student
INSERT INTO sit_user_roles (id, id_user, id_role)
VALUES (10, 10, 1); -- student
INSERT INTO sit_user_roles (id, id_user, id_role)
VALUES (11, 11, 1); -- student
INSERT INTO sit_user_roles (id, id_user, id_role)
VALUES (12, 12, 1); -- student
```

```sql
INSERT INTO sit_parents (id, id_parent, id_student)
VALUES (1, 1, 2);
INSERT INTO sit_parents (id, id_parent, id_student)
VALUES (2, 4, 5);


INSERT INTO sit_grade (id, grade)
VALUES (1, 9);
INSERT INTO sit_grade (id, grade)
VALUES (2, 10);
INSERT INTO sit_grade (id, grade)
VALUES (3, 11);
INSERT INTO sit_grade (id, grade)
VALUES (4, 12);


INSERT INTO sit_materii (id, name)
VALUES (1, 'MATEMATICA');
INSERT INTO sit_materii (id, name)
VALUES (2, 'FIZICA');
INSERT INTO sit_materii (id, name)
VALUES (3, 'CHIMIE');
INSERT INTO sit_materii (id, name)
VALUES (4, 'INFORMATICA');
INSERT INTO sit_materii (id, name)
VALUES (5, 'BIOLOGIE');
INSERT INTO sit_materii (id, name)
VALUES (6, 'GEOGRAFIE');
INSERT INTO sit_materii (id, name)
VALUES (7, 'ISTORIE');
INSERT INTO sit_materii (id, name)
VALUES (8, 'LIMBA ROMANA');
INSERT INTO sit_materii (id, name)
VALUES (9, 'LIMBA ENGLEZA');
INSERT INTO sit_materii (id, name)
VALUES (10, 'LIMBA FRANCEZA');
INSERT INTO sit_materii (id, name)
VALUES (11, 'LIMBA GERMANA');
INSERT INTO sit_materii (id, name)
VALUES (12, 'Dirigentie');

INSERT INTO sit_class (id, letter, year, grade_id)
VALUES (1, 'A', '2019-2020', 1);
INSERT INTO sit_class (id, letter, year, grade_id)
VALUES (2, 'B', '2019-2020', 1);
INSERT INTO sit_class (id, letter, year, grade_id)
VALUES (3, 'C', '2019-2020', 1);
INSERT INTO sit_class (id, letter, year, grade_id)
VALUES (4, 'D', '2019-2020', 1);
INSERT INTO sit_class (id, letter, year, grade_id)
VALUES (5, 'E', '2019-2020', 1);
INSERT INTO sit_class (id, letter, year, grade_id)
VALUES (6, 'A', '2019-2020', 2);
INSERT INTO sit_class (id, letter, year, grade_id)
VALUES (7, 'B', '2019-2020', 2);
INSERT INTO sit_class (id, letter, year, grade_id)
VALUES (8, 'C', '2019-2020', 2);
INSERT INTO sit_class (id, letter, year, grade_id)
VALUES (9, 'D', '2019-2020', 2);
INSERT INTO sit_class (id, letter, year, grade_id)
VALUES (10, 'E', '2019-2020', 2);
```

```sql
INSERT INTO sit_subject (id, id_class, id_materie, id_user_roles)
VALUES (1, 1, 1, 1);
INSERT INTO sit_subject (id, id_class, id_materie, id_user_roles)
VALUES (2, 1, 2, 1);
INSERT INTO sit_subject (id, id_class, id_materie, id_user_roles)
VALUES (3, 1, 3, 1);
INSERT INTO sit_subject (id, id_class, id_materie, id_user_roles)
VALUES (4, 1, 4, 1);
INSERT INTO sit_subject (id, id_class, id_materie, id_user_roles)
VALUES (5, 1, 5, 1);
INSERT INTO sit_subject (id, id_class, id_materie, id_user_roles)
VALUES (6, 1, 6, 8);
INSERT INTO sit_subject (id, id_class, id_materie, id_user_roles)
VALUES (7, 1, 7, 8);
INSERT INTO sit_subject (id, id_class, id_materie, id_user_roles)
VALUES (8, 1, 8, 8);
INSERT INTO sit_subject (id, id_class, id_materie, id_user_roles)
VALUES (9, 1, 9, 8);


INSERT INTO sit_obj (key, value)
VALUES (1, 'N');
INSERT INTO sit_obj (key, value)
VALUES (2, 'T');
INSERT INTO sit_obj (key, value)
VALUES (3, 'Abs');

INSERT INTO sit_catalogue (id, nota, id_student, id_materie, obj_date, id_obj)
VALUES (1, 9, 2, 1, to_date('01-01-2020', 'dd-mm-yyyy'), 1);
INSERT INTO sit_catalogue (id, nota, id_student, id_materie, obj_date, id_obj)
VALUES (2, 10, 2, 1, to_date('01-01-2020', 'dd-mm-yyyy'), 1);
```

6. Actualizarea inregistrarilor

```
update SIT_MATERII
set name = 'Spaniola'
where id = 8;

update sit_roles
set role = 'STUDENT'
where id = 1;

update SIT_CATALOGUE
set nota = 10
where id = 1;
```

+ delete

```
delete
from SIT_MATERII
where id = 8;
```

7. Stergerea si recuperarea unei tabele

```
drop table sit_jud cascade constraints;
flashback table sit_jud to before drop;
```

8. Interogari ale bazei de date

1. selecteaza toate notele studentului cu id 2

```
SELECT c.nota, m.name AS materie
FROM SIT_CATALOGUE c
         JOIN SIT_SUBJECT s ON c.id_materie = s.id_materie
         JOIN SIT_MATERII m ON s.id_materie = m.id
WHERE c.id_student = 2;
```

Output | Result 9 | 2 rows

	NOTA	MATERIE
1	9	MATEMATICA
2	10	MATEMATICA

2. afiseaza id-ul si rolul utilizatorilor

```
SELECT id,
       DECODE(id_role, 1, 'Student', 2, 'Parent', 3, 'Teacher', 4, 'Admin', 'Other') AS role
FROM sit_user_roles;
```

Output | 2. afiseaza id-ul si... rolul utilizatorilor | 12 rows

	ID	ROLE
1	6	Student
2	7	Student
3	8	Teacher
4	9	Student
5	10	Student
6	11	Student
7	12	Student
8	1	Teacher
9	2	Student
10	3	Student
11	5	Student
12	4	Parent

3. media notelor pe fiecare materie

```
SELECT id_materie, AVG(nota) AS medie_nota
FROM sit_catalogue
GROUP BY id_materie;
```

4. Notele maxime pentru fiecare student:

```sql
SELECT id_student, MAX(nota) AS nota_maxima
FROM sit_catalogue
GROUP BY id_student;
```

5.Utilizatori cu mai mult de un rol:

```sql
SELECT id_user, COUNT(DISTINCT id_role) AS numar_role
FROM sit_user_roles
GROUP BY id_user
HAVING COUNT(DISTINCT id_role) > 1;
```

6. Notele medii pentru fiecare clasă:

```sql
SELECT s.id_class, AVG(c.nota) AS medie_nota
FROM sit_catalogue c
         JOIN sit_subject s ON c.id_materie = s.id_materie
GROUP BY s.id_class;
```

7. Numărul de utilizatori din fiecare județ:

```sql
SELECT l.id_jud, COUNT(DISTINCT u.id) AS numar_utilizatori
FROM sit_user u
         JOIN sit_person p ON u.id_pers = p.id
         JOIN sit_address a ON p.id_address = a.id
         JOIN sit_loc l ON a.id_loc = l.id
GROUP BY l.id_jud;
```

8. Numărul total de studenți în fiecare clasă:

```sql
SELECT s.id_class, COUNT(DISTINCT u.id) AS numar_studenti
FROM sit_user u
         JOIN sit_user_roles r ON u.id = r.id_user
         JOIN sit_subject s ON r.id = s.id_user_roles
         JOIN sit_class c ON s.id_class = c.id
WHERE r.id_role = 1
GROUP BY s.id_class;
```

9. Materiile în care media notelor este mai mare decât 7:

```
SELECT id_materie, AVG(nota) AS medie_nota
FROM sit_catalogue
GROUP BY id_materie
HAVING AVG(nota) > 7;
```

10. Utilizatorii care nu au rolul de student:

```
SELECT u.id, u.email, u.password, u.id_pers
FROM sit_user u
         JOIN sit_user_roles r ON u.id = r.id_user
WHERE r.id_role != 1;
```

11.Utilizatorii cu date de naștere în aceeași lună:

```
SELECT TO_CHAR(b_day, 'MM') AS luna_nastere, COUNT(id) AS numar_utilizatori
FROM sit_person
GROUP BY TO_CHAR(b_day, 'MM');
```

12. Utilizatorii care au rolul de parinte si au copii in clasa a 12-a:

```
SELECT u.id, u.email, u.password, u.id_pers
FROM sit_user u
         JOIN sit_user_roles r ON u.id = r.id_user
         JOIN sit_parents p ON r.id = p.id_parent
         JOIN sit_user_roles r2 ON p.id_student = r2.id_user
         JOIN sit_subject s ON r2.id = s.id_user_roles
         JOIN sit_class c ON s.id_class = c.id
WHERE r.id_role = 2
  AND c.grade_id = 4;
```

13. Intersectarea parintilor și profesorilor:

```sql
SELECT id_user
FROM sit_user_roles
WHERE id_role = 2

INTERSECT

SELECT id_user
FROM sit_user_roles
WHERE id_role = 3;
```

14. Studenții care au absențe și note:

```sql
SELECT id_student, nota, NULL AS absenta
FROM sit_catalogue
WHERE id_obj != (SELECT key FROM sit_obj WHERE value = 'Abs')

UNION

SELECT id_student, NULL AS nota, obj_date AS absenta
FROM sit_catalogue
WHERE id_obj = (SELECT key FROM sit_obj WHERE value = 'Abs');
```

15. Utilizatorii care au același nume și prenume cu părintele lor:

```sql
SELECT u.id, u.email, u.password, u.id_pers
FROM sit_user u
         JOIN sit_user_roles r ON u.id = r.id_user
         JOIN sit_parents p ON r.id = p.id_parent
         JOIN sit_user_roles r2 ON p.id_student = r2.id_user
         JOIN sit_person p2 ON r2.id = p2.id
WHERE p2.f_name = p2.f_name
  AND p2.l_name = p2.l_name;
```

16. Notele la examenele susținute în anul 2023 (utilizând funcții de data și expresia CASE):

```sql
SELECT id,
       obj_date,
       nota,
       CASE
           WHEN EXTRACT(YEAR FROM obj_date) = 2023 THEN 'Examen 2023'
           ELSE 'Alta perioada'
           END AS perioada
FROM sit_catalogue;
```

17. Materiile la care au fost susținute cele mai multe examene (primele 5)

```sql
SELECT id_materie, COUNT(*) AS numar_examene
FROM sit_catalogue
GROUP BY id_materie
ORDER BY numar_examene DESC
    FETCH FIRST 5 ROWS ONLY;
```

18. Utilizatorii care au absențe la mai mult de 3 materii

```sql
SELECT id_student, COUNT(DISTINCT id_materie) AS numar_materii
FROM sit_catalogue
WHERE id_obj = (SELECT key FROM sit_obj WHERE value = 'Abs')
GROUP BY id_student
HAVING COUNT(DISTINCT id_materie) > 3;
```

19.Utilizatorii care au predat cel puțin o materie (utilizând INNER JOIN)

```sql
SELECT DISTINCT u.id, u.email
FROM sit_user u
         JOIN sit_user_roles ur ON u.id = ur.id_user
         JOIN sit_subject s ON ur.id_user = s.id_user_roles
where ur.ID_ROLE = 3;
```

20. Utilizatorii care nu au predat nicio materie (utilizând LEFT JOIN)

```sql
SELECT DISTINCT u.id, u.email
FROM sit_user u
         LEFT JOIN sit_user_roles ur ON u.id = ur.id_user
         LEFT JOIN sit_subject s ON ur.id_user = s.id_user_roles
WHERE s.id_user_roles IS NULL;
```

9. Gestiunea altor obiecte ale bazei de date: vederi, indecsi, sinonime, secvente.

9.1 Un view care sa contina studentul si notele lui

```sql
CREATE VIEW vw_student_grades AS
SELECT
    u.id AS student_id,
    u.email AS student_email,
    m.name AS materie,
    c.nota
FROM
    sit_user u
JOIN
    sit_user_roles ur ON u.id = ur.id_user
JOIN
    sit_catalogue c ON u.id = c.id_student
JOIN
    sit_subject s ON c.id_materie = s.id_materie
JOIN
    sit_materii m ON s.id_materie = m.id;
```

9.2 Index pe mail pentru a facilita selecturile pe acest camp:

```
CREATE INDEX idx_email ON sit_user(email);
```

9.3 Sinonim:

```
create synonym catalog for sit_catalogue;

select * from catalog;
```

9.4 Secventiator:

```
create sequence seq_sit_jud start with 1 increment by 1 nocache nocycle;
select seq_sit_jud.nextval from dual;
```

Output | nextval:INTEGER

1 row

	NEXTVAL
1	1

Poate fi utilizat ca si ID unic.