

Tema curs s10 – gestionarea unei platforme de e-learning

A.

Diagrama E/R

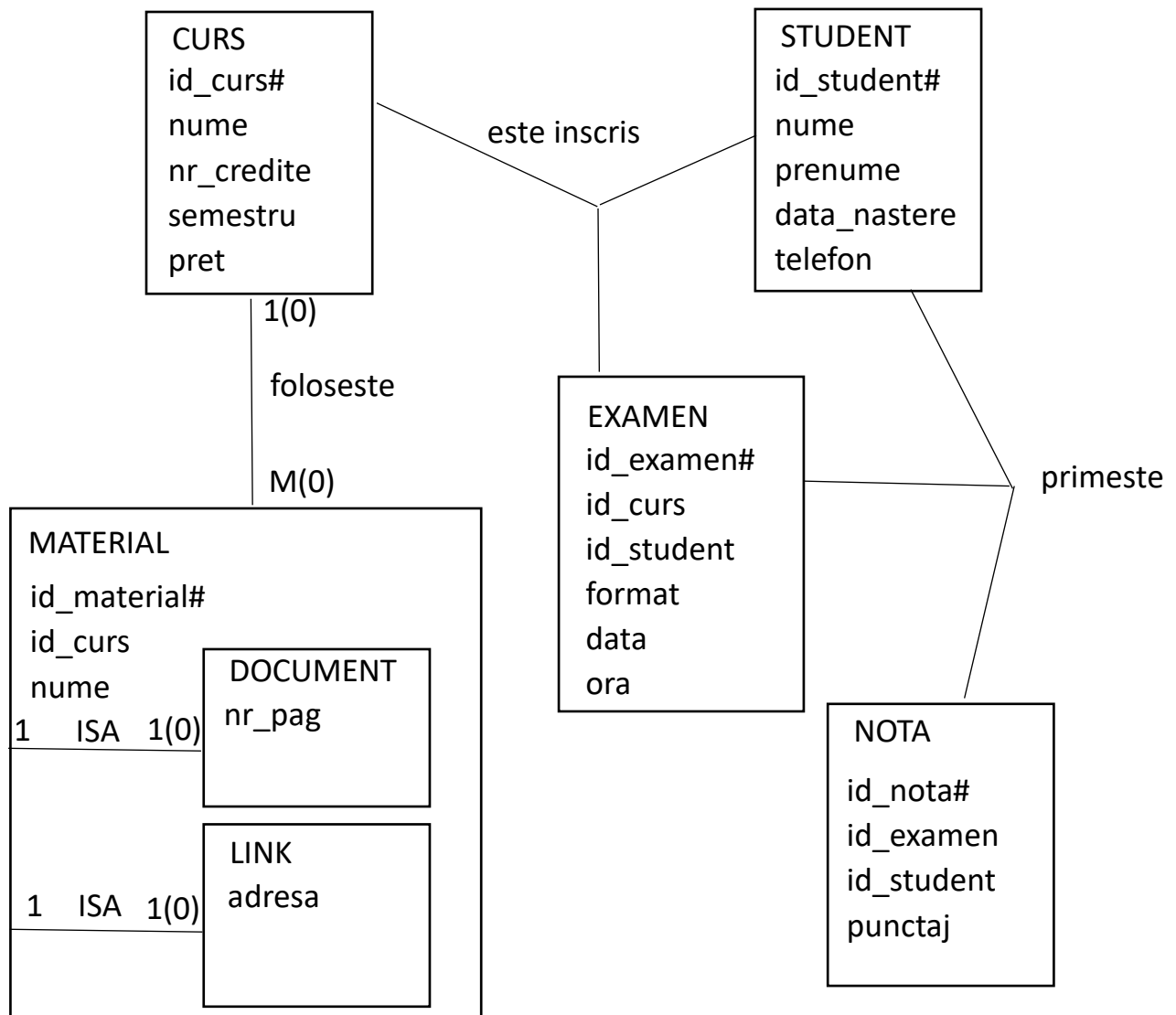
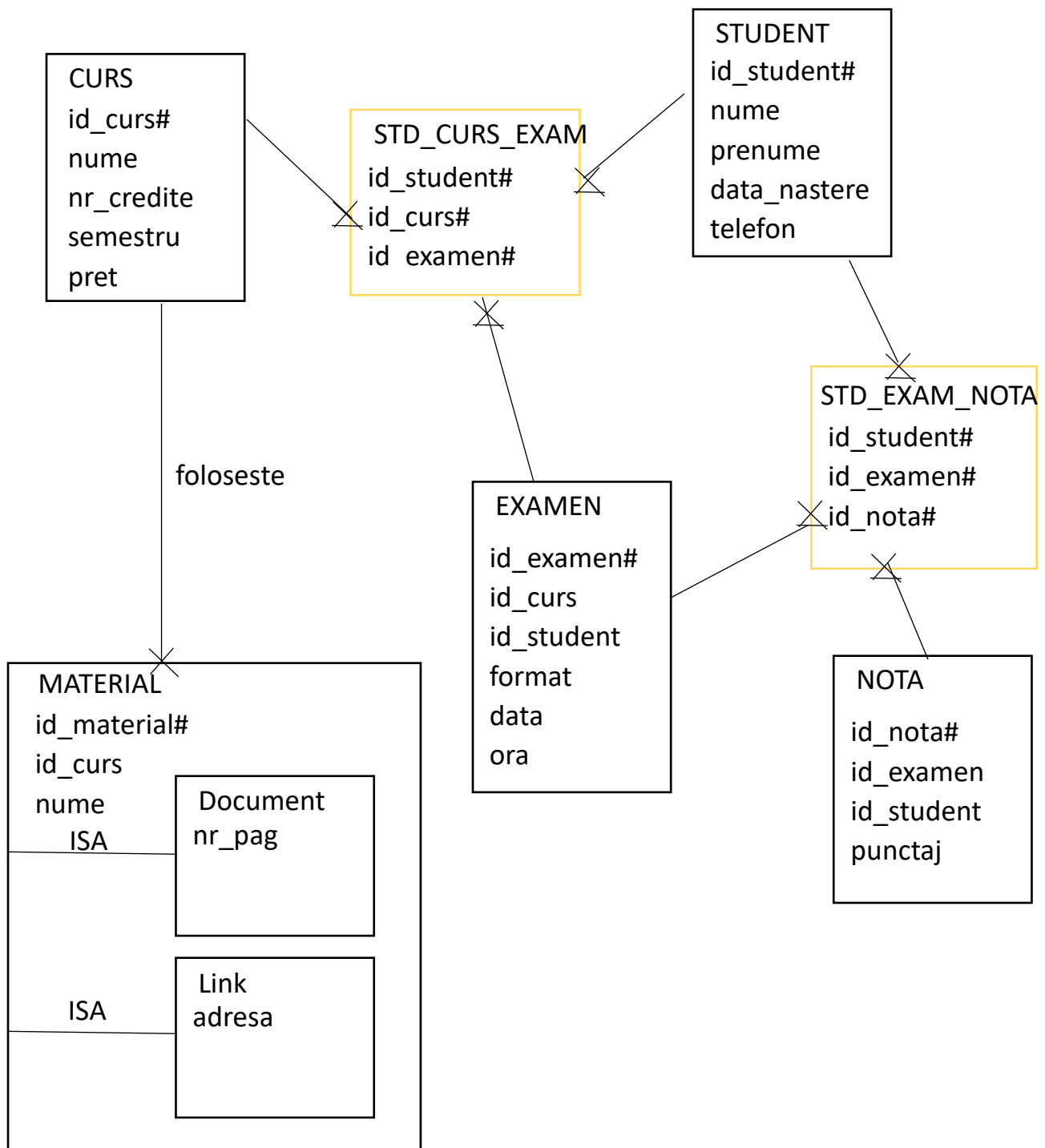


Diagrama conceptuala



Schemele relationale

STUDENT (id_student#, nume, prenume, data_nastere, telefon)

CURS (id_curs#, nume, nr_credite, semestru, pret)

EXAMEN(id_examen#, id_curs, id_student, format, data, ora)

STD_CURS_EXAM (id_student#, id_curs#, id_examen#)

NOTA(id_nota#, id_examen, id_student, punctaj)

STD_EXAM_NOTA(id_student#, id_examen#, id_nota#)

MATERIAL (id_material#, id_curs, nume)

DOCUMENT(id_material#, nr_pag)

LINK(id_material#, adresa)

B.

1. Care sunt cursurile la care sunt înrolați studenții în semestrul curent?

```
select nume  
from curs  
where semestru == 2;
```

2. Care sunt materialele (documente, link-uri externe etc) de curs/laborator/seminar pe care trebuie să le parcurgă un student pentru un curs?

```
select m.id_material  
from material m, curs c  
where c.id_curs == m.id_curs;
```

3. Care sunt notele studenților la materiile studiate?

```
select n.punctaj as "Nota", s.numa as "Nume student", c.numa as "Nume  
curs"  
from nota n, student s, examen e, curs c,  
where n.id_examen == e.id_examen and n.id_student == s.id_student and  
e.id_curs == c.id_curs;
```

4. Când au loc sesiunile online de curs sau examinare?

```
select c.numa as "Materie", e.data as "Data", e.ora as "Ora"  
from examen e, curs c  
where e.id_curs == c.id_curs;
```

C. Dați exemple ale operatorilor algebrei relaționale (11 operatori): cerere în limbaj natural, expresie în algebra relațională și cerere în SQL.

1. Sa se afiseze pretul, numele si id-ul tuturor cursurilor .

```
PROJECT(CURS, pret, nume, id_curs)
```

```
select pret, nume, id_curs  
from curs;
```

2. Sa se afiseze toate informatiile cursurilor din semestrul 1.

```
SELECT(CURS, semestru=1)
```

```
select *  
from curs  
where semestrul == 1;
```

3. Sa se afiseze materialele necesare pentru fiecare curs.

```
R = PROJECT(CURS, nume);  
S = PROJECT(MATERIAL, nume);  
UNION(R, S);
```

```
select nume  
from curs  
union  
select nume  
from material;
```

4. Sa se afiseze numele, prenumele si data nasterii studentilor care sunt nascuti inainte de data 20.11.2002, dar care nu au numele de familie „Popescu”.

```
R = PROJECT(SELECT(STUDENT, data_nastere<"20.11.2002", nume, prenume,  
data_nastere);
```

```
S = PROJECT(SELECT(STUDENT, nume="Popescu"), nume, prenume,  
data_nastere);  
DIFFERENCE(R, S);
```

```
select nume, prenume, data_nastere  
from student  
where data_nastere <to_date("20.11.2002" , "dd.mm.yyyy" )  
minus  
select nume, prenume, data_nastere  
from student  
where nume="Popescu";
```

5. Sa se afiseze materialele utile pentru cursul de „Structuri de date” si „Tehnici web”.

```
R = PROJECT(SELECT(CURS, nume = "structuri de date");  
S = PROJECT(SELECT(CURS, nume = "tehnici web");  
P = PROJECT(SELECT(MATERIAL, id_curs in R), nume);  
Q = PROJECT(SELECT(MATERIAL, id_curs in S), nume);  
INTERSECT(P, Q);
```

```
select nume  
from material  
where id_curs in (select id_curs  
                  from curs  
                  where lower(nume) = "structuri de date")  
intersect  
select nume  
from material  
where id_curs in (select id_curs  
                  from curs  
                  where lower(nume) = "tehnici web");
```

6. Sa se afiseze toate posibilitatile de note pe care le poate obtine un student.

```
R = PROJECT(STUDENT, id_student, nume, prenume);  
S = PROJECT(NOTA, punctaj);  
PRODUCT(R, S);
```

```
select s.id_student, s.nume, s.prenume, n.punctaj  
from student s, nota n;
```

7. Sa se afiseze id cursurilor mai scumpe de 200 de lei la care se foloseste materialul cu numele „Algebra”.

```
R = PROJECT(SELECT(CURS, pret >= 200), id_curs);  
S = PROJECT(SELECT(MATERIAL, nume="algebra"), id_curs);  
DIVISION(R, S);
```

```
select id_curs from curs c where not exists (select * from material m where  
nume != "Algebra") and not exists (select * from curs c1 where c.id_curs =  
m.id_curs and c.id_curs = c1.id_curs and c1.pret >= 200);
```

8. Sa se afiseze codurile studentilor si notele care au fost obtinute la examene.

```
JOIN(STUDENT, NOTA);  
  
select s.id_student, n.punctaj  
from student s, nota n  
where s.id_student == n.id_student;
```

9. Sa se afiseze numele cursurilor si al materialelor asociate care au numele materialelor diferit de "Poligoane".

```
R = PROJECT(CURS, nume);
```

```
S = PROJECT(MATERIAL, nume);  
JOIN(R, S, MATERIAL.nume <> "poligon");
```

```
select c.nume, m.nume  
from curs c, material m  
where lower(c.nume) <> "poligon";
```

10. Sa se afiseze numele si prenumele studentilor care sunt inscrisi la cursul de „Tehnici web”

```
R = SELECT(CURS, nume == 'tehnici web') ;  
S = JOIN(STUDENT, R) ;  
PROJECT(S, nume, prenume);
```

```
select s.nume, s.prenume  
from student s, curs c, std_curs_exam sce  
where s.id_student=sce.id_student and c.id_curs=sce.id_curs and lower(c.nume)  
= 'tehnici web';
```

11. Sa se afiseze cursurile la care nu s-a inscris niciun student

```
R = RIGHT OUTER JOIN(STUDENT, CURS);  
  
select c.nume  
from curs c right outer join std_curs_exam sce  
on c.id_curs = sce.id_curs;
```