Popescu Paullo Robertto Karloss

Grupa 131

Tema 3

TEMĂ

Diagrama ER:

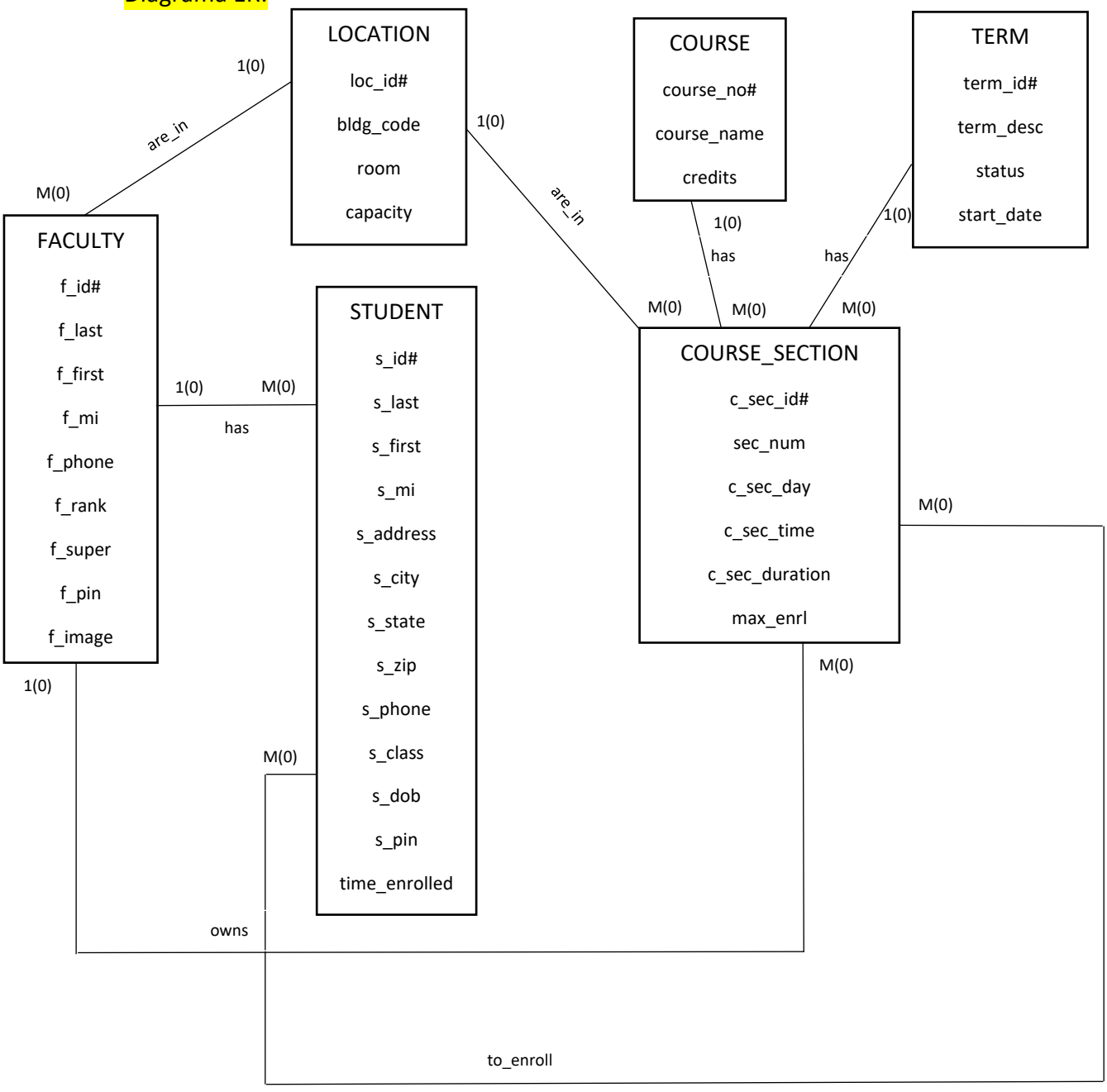
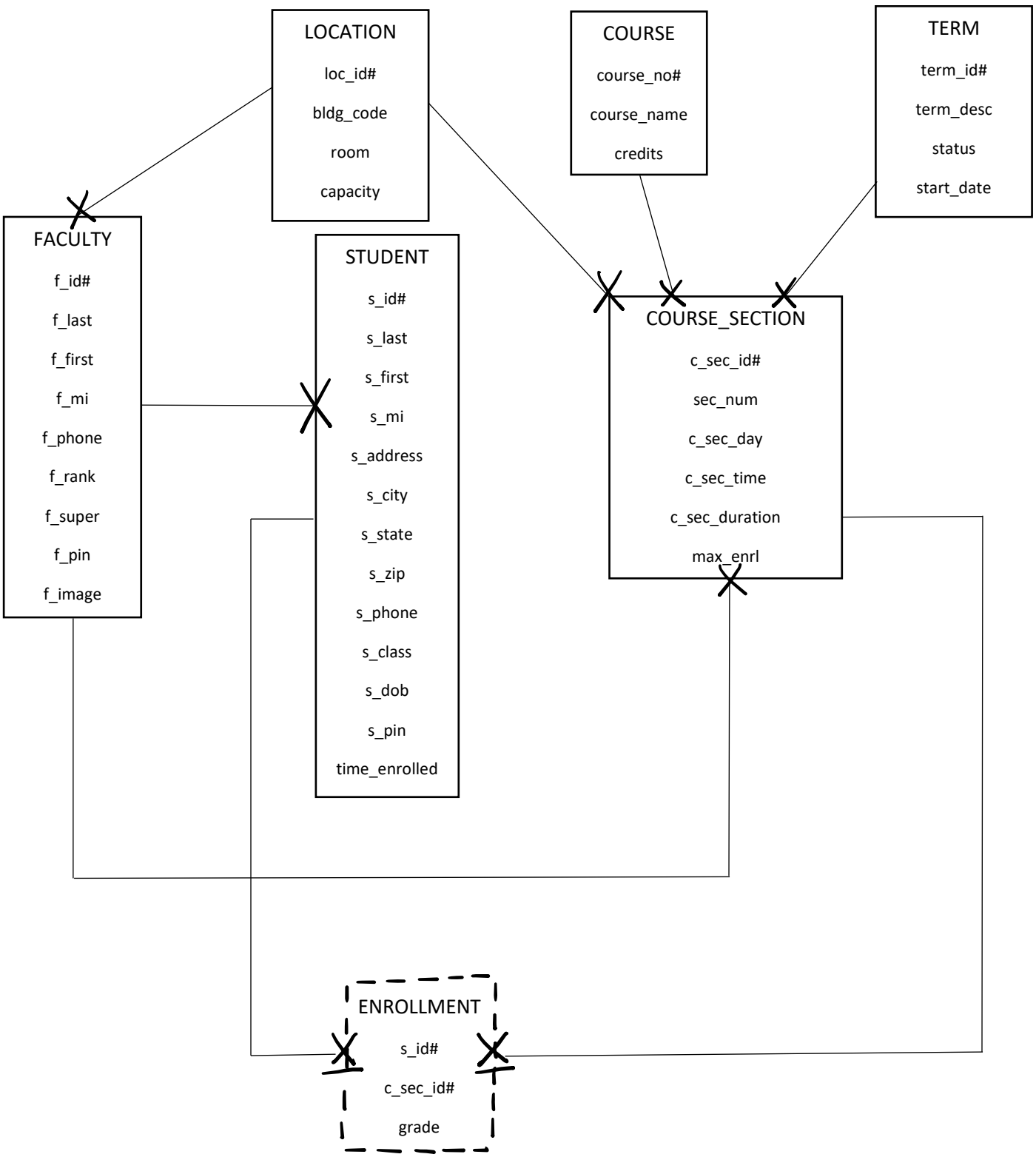


Diagrama conceptuală:



! Înainte de a trece la rezolvarea exerciţiilor, pornind de la schemele relaţionale ale tabelelor, realizaţi **diagrama conceptuală a modelului**. Schemele relaţionale sunt:
LOCATION (loc_id#, bldg_code, room, capacity)
FACULTY (f_id#, f_last, f_first, f_mi, loc_id, f_phone, f_rank, f_super, f_pin, f_image BLOB)
STUDENT (s_id#, s_last, s_first, s_mi, s_address, s_city, s_state, s_zip, s_phone, s_class, s_dob, s_pin, f_id, time_enrolled)
TERM (term_id#, term_desc, status, start_date)
COURSE (course_no#, course_name, credits)
COURSE_SECTION (c_sec_id#, course_no, term_id, sec_num, f_id, c_sec_day, c_sec_time, c_sec_duration, loc_id, max_enrl)
ENROLLMENT (s_id#, c_sec_id#, grade)

Obs: Tabelul *FACULTY* conţine cadrele didactice ale facultăţii.

REZOLVĂRI (LE PUTEȚI GĂSI ȘI ÎN FIȘIERUL .SQL ATAȘAT) !

1) ***Cerința:*** Să se determine, printr-o singură cerere, codul şi numele studenţilor coordonaţi de profesorul Brown, precum şi codul şi numele cursurilor ţinute de acesta. Etichetaţi coloanele „Cod” şi „Student sau curs”.

Adăugaţi o coloană denumită „tip”, care precizează tipul valorii de pe coloana a doua („student” sau „curs”).

Rezolvare:

```
select s_id "COD", s_last || ' ' || s_first "STUDENT SAU CURS", 'Student' "TIP"
from student
--aici am crezut ca exista doar un singur profesor pe care il cheama BROWN
--de aceea n-am folosit in si am folosit =
where f_id = (select f_id
        from faculty
        where upper(f_last) = 'BROWN')
union
select course_no "COD", course_name "STUDENT SAU CURS", 'Curs'
from course
where course_no in (select course_no
            from course_section
```

--la fel si aici

where f_id = (select f_id

from faculty

where upper(f_last) = 'BROWN'));

Print-Screen:

```
--1
select s_id "COD", s_last || ' ' || s_first "STUDENT SAU CURS", 'Student' "TIP"
from student
--aici am crezut ca exista doar un singur profesor pe care il cheama BROWN
--de aceea n-am folosit in si am folosit =
where f_id = (select f_id
              from faculty
              where upper(f_last) = 'BROWN')
union
select course_no "COD", course_name "STUDENT SAU CURS", 'Curs'
from course
where course_no in (select course_no
                    from course_section
                    --la fel si aici
                    where f_id = (select f_id
                                  from faculty
                                  where upper(f_last) = 'BROWN'));
```

Query Result | SQL | All Rows Fetched: 2 in 0.062 seconds

	COD	STUDENT SAU CURS	TIP
1	JO101	Johnson Lisa	Student
2	MIS 301	Systems Analysis	Curs

2) ***Cerința:*** Determinați studenţii (cod, nume) care au urmat un curs de Baze de date, dar nu şi unul de Programare în C++.

Rezolvare:

select s.s_id, s.s_last from student s, course c, enrollment e, course_section cs

where s.s_id = e.s_id

and cs.c_sec_id = e.c_sec_id

and cs.course_no = c.course_no

and c.course_name = 'Database Management'

and c.course_name not in (select course_name from course

where course_name = 'Programming in C++');

Print-Screen:

```
--2
select s.s_id, s.s_last from student s, course c, enrollment e, course_section cs
where s.s_id = e.s_id
and cs.c_sec_id = e.c_sec_id
and cs.course_no = c.course_no
and c.course_name = 'Database Management'
and c.course_name not in (select course_name from course
                          where course_name = 'Programming in C++');
```

Query Result | SQL | All Rows Fetched: 4 in 0.306 seconds

	S_ID	S_LAST
1	JO100	Jones
2	JO101	Johnson
3	MA100	Marsh
4	PE100	Perez

3) ***Cerința:*** Determinaţi studenţii (cod, nume) care au obţinut nota C la cel puţin un examen sau care au cel puţin o notă necunoscută.

Rezolvare:

select distinct s.s_id as "COD", s.s_last as "NUME"

from student s

where (select count(1)

from enrollment

where s_id = s.s_id

and (grade = 'C') or grade = null) > 0;

Print-Screen:

```
--3
select distinct s.s_id as "COD", s.s_last as "NUME"
from student s
where (select count(1)
        from enrollment
        where s_id = s.s_id
        and (grade = 'C') or grade = null) > 0;
```

Query Result | SQL | All Rows Fetched: 3 in 0.012 seconds

	COD	NUME
1	PE100	Perez
2	MA100	Marsh
3	JO101	Johnson

4) ***Cerința:*** Afişaţi locaţiile care au capacitate maximă (codul locaţiei, codul clădirii, capacitatea).

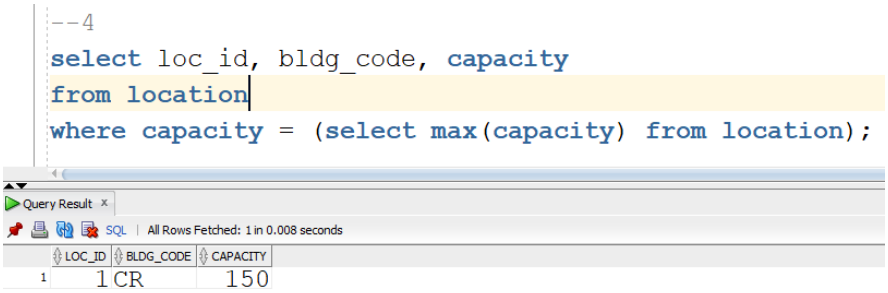
Rezolvare:

select loc_id, bldg_code, capacity

from location

where capacity = (select max(capacity) from location);

Print-Screen:



5) ***Cerința:*** Executaţi comenzile următoare:

CREATE TABLE t (id NUMBER PRIMARY KEY);

INSERT INTO t VALUES(1);

INSERT INTO t VALUES(2);

INSERT INTO t VALUES(4);

INSERT INTO t VALUES(6);

INSERT INTO t VALUES(8);

INSERT INTO t VALUES(9);

Determinaţi id-ul minim, respectiv maxim, disponibil.

De exemplu, pentru valorile introduse, cererea va întoarce valorile 3 şi 7.

Rezolvare:

CREATE TABLE t (id NUMBER PRIMARY KEY);

INSERT INTO t VALUES(1);

INSERT INTO t VALUES(2);

INSERT INTO t VALUES(4);

INSERT INTO t VALUES(6);

INSERT INTO t VALUES(8);

INSERT INTO t VALUES(9);

select min(id)+1 as Minim, null as Maxim

from T

where id+1 not in (select * from t)

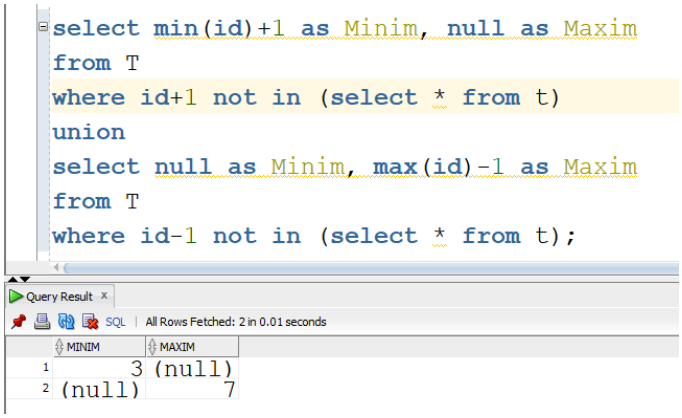
union

select null as Minim, max(id)-1 as Maxim

from T

where id-1 not in (select * from t);

Print-Screen:



6) ***Cerința:*** Să se obţină un rezultat de forma: cod profesor, nume profesor, student, curs. Pentru fiecare profesor, coloanele student şi curs vor afişa „Da” dacă există vreun student coordonat, respectiv vreun curs prezentat de acel profesor, şi „Nu” altfel. În cazul afirmativ, se va specifica între paranteze numărul de studenţi coordonaţi, respectiv numărul de cursuri ţinute.

Rezolvare:

select f.f_id "Cod profesor", f.f_last || ' ' || f.f_first "Nume Profesor",

decode(nvl(s.nr, 0), 0, 'Nu', 'Da (' || s.nr || ')') "Student",

decode(nvl(c.nr, 0), 0, 'Nu', 'Da (' || c.nr || ')') "Curs"

```
from faculty f, (select
                    count(1) as nr, f_id
                    from student
                    group by f_id) s,
                   (select
                    count(distinct course_no) nr, f_id
                    from course_section
                    group by f_id) c
where s.f_id (+) = f.f_id
and c.f_id (+) = f.f_id;
```

Print-Screen:

```
--6
select f.f_id "Cod profesor", f.f_last || ' ' || f.f_first "Nume Profesor",
decode(nvl(s.nr, 0), 0, 'Nu', 'Da (' || s.nr || ')') "Student",
decode(nvl(c.nr, 0), 0, 'Nu', 'Da (' || c.nr || ')') "Curs"
from faculty f, (select
                 count(1) as nr, f_id
                 from student
                 group by f_id) s,
                (select
                 count(distinct course_no) nr, f_id
                 from course_section
                 group by f_id) c
where s.f_id (+) = f.f_id
and c.f_id (+) = f.f_id;
```

Query Result | SQL | All Rows Fetched: 5 in 0.009 seconds

	Cod profesor	Nume Profesor	Student	Curs
1	1	Marx Teresa	Da (3)	Da (2)
2	2	Zhulin Mark	Da (1)	Da (3)
3	3	Langley Colin	Da (1)	Da (2)
4	4	Brown Jonnel	Da (1)	Da (1)
5	5	Sealy James	Nu	Da (1)

7) ***Cerința:*** Determinaţi perechile posibile de semestre a căror descriere (term_desc) diferă doar pe ultimul caracter.

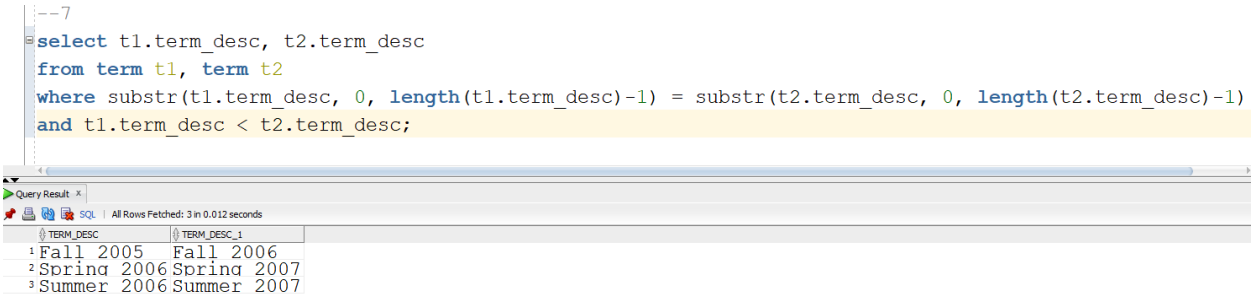
Rezolvare:

```
select t1.term_desc, t2.term_desc
from term t1, term t2
```

where substr(t1.term_desc, 0, length(t1.term_desc)-1) = substr(t2.term_desc, 0, length(t2.term_desc)-1)

and t1.term_desc < t2.term_desc;

Print-Screen:



8) ***Cerința:*** Determinaţi studenţii care au urmat cel puţin două cursuri al căror cod (course_no) diferă pe al cincilea caracter.

Rezolvare:

with cursuri as (select s_id,s_last,s_first,course_no

from student s join enrollment e using(s_id) join course_section c1 using(c_sec_id)

order by s_id asc)

select distinct c1.s_id,c1.s_last,c1.s_first,c1.course_no,c2.course_no

from cursuri c1 join cursuri c2 on(c1.s_id = c2.s_id)

where (substr(c1.course_no, 5, 1) != substr(c2.course_no, 5, 1))

and (c1.course_no != c2.course_no)

and (c1.course_no < c2.course_no)

order by c1.s_id asc;

Print-Screen:

```
--8
with cursuri as (select s_id,s_last,s_first,course_no
from student s join enrollment e using(s_id) join course_section c1 using(c_sec_id)
order by s_id asc)
select distinct c1.s_id,c1.s_last,c1.s_first,c1.course_no,c2.course_no
from cursuri c1 join cursuri c2 on(c1.s_id = c2.s_id)
where (substr(c1.course_no, 5, 1) != substr(c2.course_no, 5, 1))
      and (c1.course_no != c2.course_no)
      and (c1.course_no < c2.course_no)
order by c1.s_id asc;
```

Query Result

SQL | All Rows Fetched: 20 in 0.575 seconds

	S_ID	S_LAST	S_FIRST	COURSE_NO	COURSE_NO_1
1	JO100	Jones	Tammy	MIS 101	MIS 301
2	JO100	Jones	Tammy	MIS 101	MIS 441
3	JO100	Jones	Tammy	MIS 101	MIS 451
4	JO100	Jones	Tammy	MIS 301	MIS 441
5	JO100	Jones	Tammy	MIS 301	MIS 451
6	JO101	Johnson	Lisa	MIS 101	MIS 301
7	JO101	Johnson	Lisa	MIS 101	MIS 441
8	JO101	Johnson	Lisa	MIS 101	MIS 451
9	JO101	Johnson	Lisa	MIS 301	MIS 441
10	JO101	Johnson	Lisa	MIS 301	MIS 451
11	MA100	Marsh	John	MIS 101	MIS 301
12	MA100	Marsh	John	MIS 101	MIS 441
13	MA100	Marsh	John	MIS 301	MIS 441
14	NG100	Nguyen	Ni	MIS 101	MIS 301
15	PE100	Perez	Jorge	MIS 101	MIS 301
16	PE100	Perez	Jorge	MIS 101	MIS 441
17	PE100	Perez	Jorge	MIS 101	MIS 451
18	PE100	Perez	Jorge	MIS 301	MIS 441
19	PE100	Perez	Jorge	MIS 301	MIS 451
20	SM100	Smith	Mike	MIS 101	MIS 301

9) ***Cerința:*** Determinaţi perechile de coduri de cursuri care s-au ţinut pe acelaşi semestru. Perechile se vor considera neordonate (dacă se determină (x,y), nu se va include în rezultat şi (y, x)). Codul mai mare va fi pe prima coloană.

Rezolvare:

select cs1.course_no, cs2.course_no

from course_section cs1

join course_section cs2 using (term_id)

where cs1.course_no > cs2.course_no;

Print-Screen:

```
--9
select cs1.course_no, cs2.course_no
from course_section cs1
join course_section cs2 using (term_id)
where cs1.course_no > cs2.course_no;
```

Query Result | SQL | All Rows Fetched: 19 in 0.012 seconds

	COURSE_NO	COURSE_NO_1
1	MIS 301	MIS 101
2	MIS 301	MIS 101
3	MIS 301	MIS 101
4	MIS 451	MIS 301
5	MIS 451	MIS 301
6	MIS 441	MIS 301
7	MIS 441	MIS 301
8	MIS 451	MIS 441
9	MIS 451	MIS 441
10	MIS 451	MIS 441
11	MIS 451	MIS 441
12	MIS 451	CS 155
13	MIS 451	CS 155
14	MIS 441	CS 155
15	MIS 441	CS 155
16	MIS 301	CS 155
17	MIS 441	MIS 101
18	MIS 301	MIS 101
19	MIS 441	MIS 301

10) ***Cerința:*** Să se determine codul, numele cursului, denumirea semestrului şi numărul de locuri (max_enrl) pentru cursurile al căror număr de locuri este mai mic decât numărul de locuri corespunzător oricărui curs ţinut în locaţia 1.

Rezolvare:

select c_sec_id "Cod", course_no "Numele cursului", term_desc "Denumirea semestrului", max_enrl "Nr. locuri"

from course_section join term using (term_id)

where max_enrl < (select min(max_enrl)

from course_section

where loc_id = 1);

Print-Screen:

```
--10
select c_sec_id "Cod", course_no "Numele cursului", term_desc "Denumirea semestrului", max_enrl "Nr. locuri"
from course_section join term using (term_id)
where max_enrl < (select min(max_enrl)
                  from course_section
                  where loc_id = 1);
```

Query Result | SQL | All Rows Fetched: 11 in 0.01 seconds

	Cod	Numele cursului	Denumirea semestrului	Nr. locuri
1	2	MIS 101	Fall 2006	35
2	3	MIS 101	Fall 2006	35
3	4	MIS 301	Fall 2006	35
4	5	MIS 301	Spring 2007	35
5	6	MIS 441	Spring 2007	30
6	7	MIS 441	Spring 2007	30
7	8	CS 155	Spring 2007	35
8	9	MIS 451	Spring 2007	35
9	10	MIS 451	Spring 2007	35
10	12	MIS 301	Summer 2007	35
11	13	MIS 441	Summer 2007	35

11) ***Cerința:*** Determinaţi cursurile cu număr minim de locuri. Se vor afişa numele cursului şi numărul de locuri.

Rezolvare:

select distinct course_name, cs.max_enrl

from course_section cs join course using (course_no)

where cs.max_enrl = (select min(cs2.max_enrl)

from course_section cs2);

Print-Screen:

```
--11
select distinct course_name, cs.max_enrl
from course_section cs join course using (course_no)
where cs.max_enrl = (select min(cs2.max_enrl)
                     from course_section cs2);
```

Query Result | SQL | All Rows Fetched: 1 in 0.009 seconds

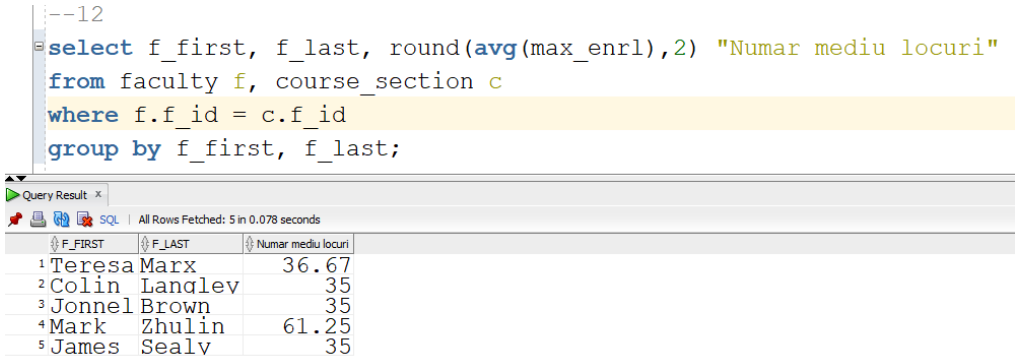
	COURSE_NAME	MAX_ENRL
1	Database Management	30

12) ***Cerința:*** Pentru fiecare profesor, să se afişeze numele acestuia şi numărul mediu de locuri corespunzătoare cursurilor sale.

Rezolvare:

```
select f_first, f_last, round(avg(max_enrl),2) "Numar mediu locuri"
from faculty f, course_section c
where f.f_id = c.f_id
group by f_first, f_last;
```

Print-Screen:

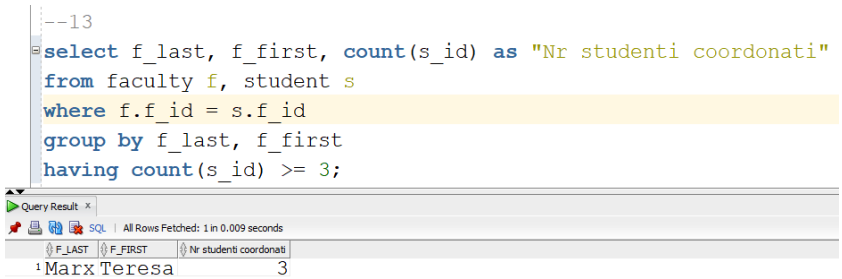


13) ***Cerința:*** Determinaţi profesorii care coordonează cel puţin 3 studenţi. Afişaţi numele profesorului şi numărul de studenţi coordonaţi.

Rezolvare:

```
select f_last, f_first, count(s_id) as "Nr studenti coordonati"
from faculty f, student s
where f.f_id = s.f_id
group by f_last, f_first
having count(s_id) >= 3;
```

Print-Screen:



```
--13
select f_last, f_first, count(s_id) as "Nr studenti coordonati"
from faculty f, student s
where f.f_id = s.f_id
group by f_last, f_first
having count(s_id) >= 3;
```

Query Result — All Rows Fetched: 1 in 0.009 seconds

	F_LAST	F_FIRST	Nr studenti coordonati
1	Marx	Teresa	3

14) ***Cerința:*** Determinaţi, pentru fiecare curs, capacitatea maximă a locaţiilor în care s-a desfăşurat. Se vor afişa numele cursului, capacitatea maximă şi codul locaţiei corespunzătoare.

Rezolvare:

select distinct c.course_name, l.capacity, l.loc_id

from course c

join course_section cs on (c.course_no = cs.course_no)

join location l on (cs.loc_id = l.loc_id)

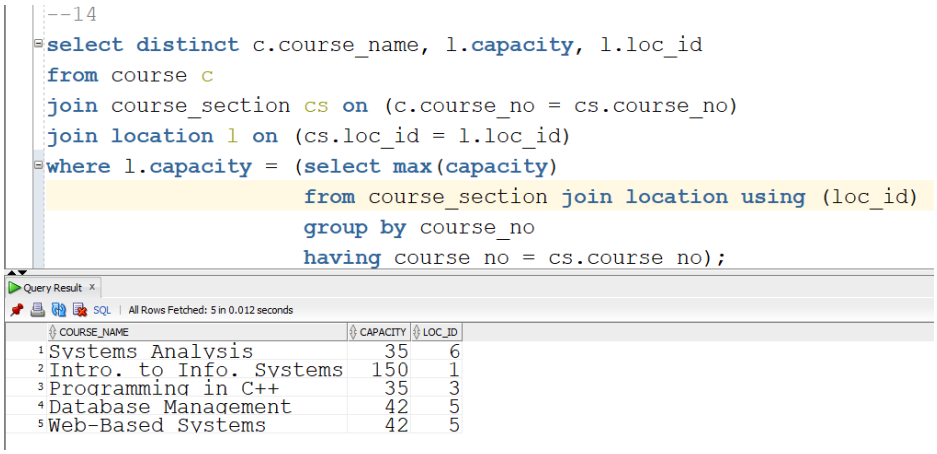
where l.capacity = (select max(capacity)

from course_section join location using (loc_id)

group by course_no

having course_no = cs.course_no);

Print-Screen:



```
--14
select distinct c.course_name, l.capacity, l.loc_id
from course c
join course_section cs on (c.course_no = cs.course_no)
join location l on (cs.loc_id = l.loc_id)
where l.capacity = (select max(capacity)
                    from course_section join location using (loc_id)
                    group by course_no
                    having course_no = cs.course_no);
```

Query Result — All Rows Fetched: 5 in 0.012 seconds

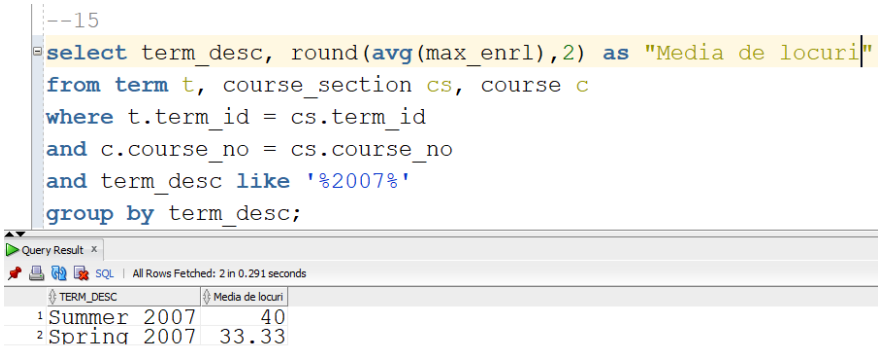
	COURSE_NAME	CAPACITY	LOC_ID
1	Systems Analysis	35	6
2	Intro. to Info. Systems	150	1
3	Programming in C++	35	3
4	Database Management	42	5
5	Web-Based Systems	42	5

15) ***Cerința:*** Pentru fiecare semestru din 2007, să se afle valoarea medie a numărului de locuri la cursurile din semestrul respective.

Rezolvare:

```
select term_desc, round(avg(max_enrl),2) as "Media de locuri"

from term t, course_section cs, course c

where t.term_id = cs.term_id

and c.course_no = cs.course_no

and term_desc like '%2007%'

group by term_desc;
```

Print-Screen:



```
--15
select term_desc, round(avg(max_enrl),2) as "Media de locuri"
from term t, course_section cs, course c
where t.term_id = cs.term_id
and c.course_no = cs.course_no
and term_desc like '%2007%'
group by term_desc;
```

Query Result

All Rows Fetched: 2 in 0.291 seconds

	TERM_DESC	Media de locuri
1	Summer 2007	40
2	Spring 2007	33.33