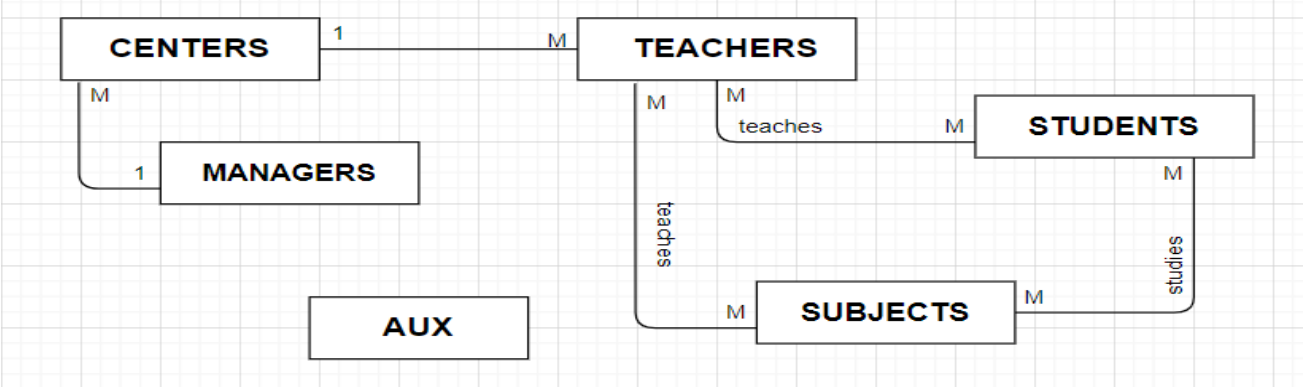
Bucur Andreea
Grupa 233
Baza de date pentru un centru de meditatii

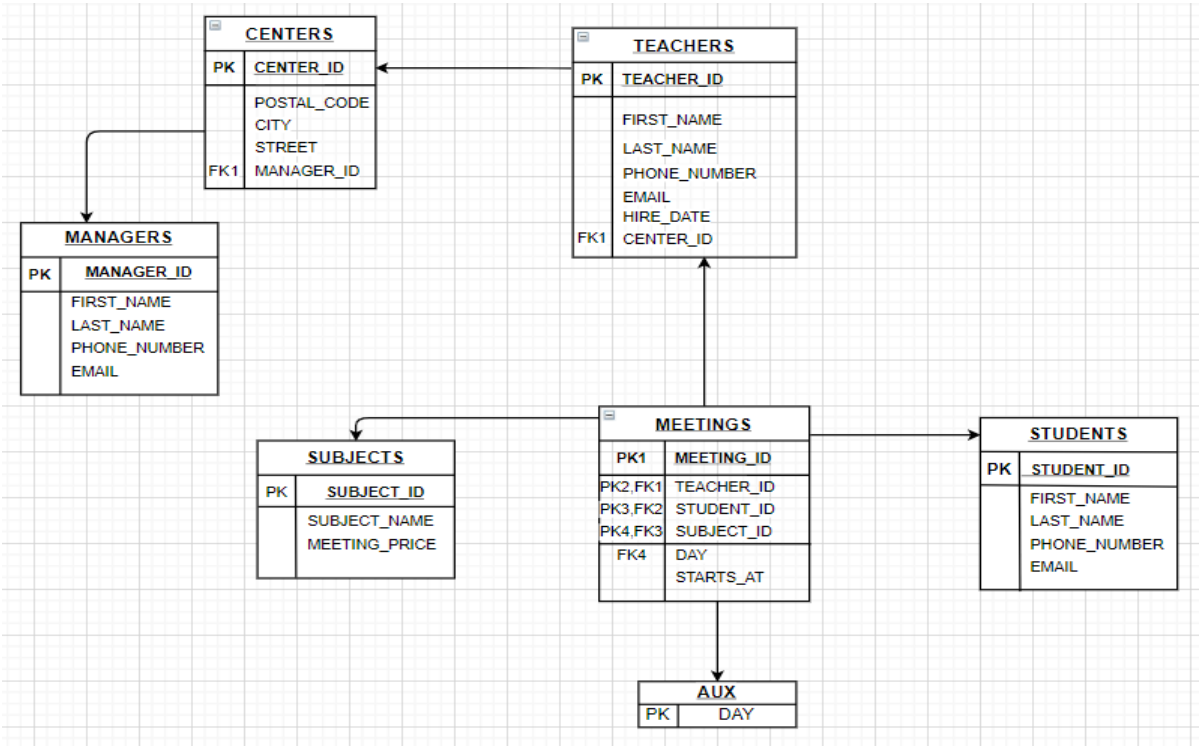
1. Prezentaţi pe scurt baza de date (utilitatea ei).

Baza de date a fost proiectata pentru a tine evidenta unor centre de meditatii ale unui antreprenor. In baza de date se gasesc toti profesorii care predau in aceste centre, elevii care fac meditatii, managerul fiecarui centru si toate sedintele. Tabela meetings contine id-ul profesorului, id-ul elevului ,id-ul materiei la care elevul face pregatire cu profesorul respectiv si ziua si ora in care se tine sedinta. O sedinta se tine in fiecare saptamana in aceeasi zi si la aceeasi ora.

2. Realizaţi diagrama entitate-relaţie (ERD).



3. Pornind de la diagrama entitate-relaţie realizaţi diagrama conceptuală a modelului propus, integrând toate atributele necesare.



4. Implementaţi în Oracle diagrama conceptuală realizată: definiţi toate tabelele, implementând toate constrângerile de integritate necesare (chei primare, cheile externe etc).

```
CREATE TABLE MANAGERS(
   MANAGER_ID NUMBER(3),
   first_name varchar2(20),
   last_name  varchar2(30),
   phone_number varchar(12),
   email varchar(30),
   PRIMARY KEY (MANAGER_ID)
   );


create table centers(
   center_id number(2),
   postal_code number(7),
   city varchar2(30) ,
   street varchar2(30),
   manager_id number(2),
   primary key (center_id),
   foreign key (manager_id) references managers(manager_id)
   );


create table students(
   student_id number(3),
   first_name varchar2(20),
   last_name  varchar2(30),
   phone_number varchar(12),
   email varchar(30),
   primary key (student_id)
   );

create table teachers(
   teacher_id number(3),
   first_name varchar2(20),
   last_name  varchar2(30),
   phone_number varchar(12),
   email varchar(30),
   hire_date date,
   center_id number(3),
   primary key (teacher_id),
   FOREIGN KEY (center_id) REFERENCES centers(center_id)
               );
```

```
Table CENTERS created.

Table STUDENTS created.

Table TEACHERS created.

Table SUBJECTS created.

Table AUX created.

Table MEETINGS created.
```

```
create table subjects(
    subject_id number(2),
    subject_name varchar2(30),
    meeting_price number(2),
    primary key (subject_id)
    );
```

```
create table aux(
    day varchar2(11),
    primary key (day)
    );
```

```
create table meetings(
    meeting_id number(3),
    subject_id number(2),
    student_id number(3),
    teacher_id number(3),
    day varchar(11),
    start_at varchar(5),
    FOREIGN KEY (student_id) REFERENCES students(student_id),
    FOREIGN KEY (teacher_id) REFERENCES teachers(teacher_id),
    FOREIGN KEY (subject_id) REFERENCES subjects(subject_id),
    FOREIGN KEY (day) references aux(day)
    );
```

5. Adăugaţi informaţii coerente în tabelele create (minim 3-5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).

```
CREATE SEQUENCE teacher_id_seq  START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE center_id_seq    START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE meeting_id_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE subject_id_seq   START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE student_id_seq   START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE manager_id_seq START WITH 1 INCREMENT BY 1;
```

```
INSERT INTO MANAGERS VALUES
(MANAGER_ID_SEQ.NEXTVAL,'CLAIRE','BROOK','0785395410','clairebrook@email.com');

INSERT INTO MANAGERS VALUES
(MANAGER_ID_SEQ.NEXTVAL,'JOHN','BROWN','0745892012','johnbrown@email.com');

INSERT INTO MANAGERS VALUES
(MANAGER_ID_SEQ.NEXTVAL,'THERESA','MUND','0762010365','theresamund@email.com');

INSERT INTO MANAGERS VALUES
(MANAGER_ID_SEQ.NEXTVAL,'MELINDA','WALTER','0748901202','melindawalter@email.com');

INSERT INTO MANAGERS VALUES
(MANAGER_ID_SEQ.NEXTVAL,'ROBERT','ZANE','0785223658','robertzane@email.com');

INSERT INTO MANAGERS VALUES
(MANAGER_ID_SEQ.NEXTVAL,'Kate','Peralta','0778545668','kateperalta@email.com');
```

	MANAGER_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER	EMAIL
1	1	CLAIRE	BROOK	0785395410	clairebrook@email.com
2	2	JOHN	BROWN	0745892012	johnbrown@email.com
3	3	THERESA	MUND	0762010365	theresamund@email.com
4	4	MELINDA	WALTER	0748901202	melindawalter@email.com
5	5	ROBERT	ZANE	0785223658	robertzane@email.com
6	6	Kate	Peralta	0778545668	kateperalta@email.com

```
insert into centers values (center_id_seq.nextval,127324,'New York','Maiden Lane',1);
insert into centers values (center_id_seq.nextval,124856,'Chicago','Prairie Avenue',2);
insert into centers values (center_id_seq.nextval,138453,'Los Angeles','Sunset Boulevard',3);
insert into centers values (center_id_seq.nextval,178513,'Miami','Montgomery Street',4);
insert into centers values (center_id_seq.nextval,178963,'Seattle','First Avenue',5);
insert into centers values (center_id_seq.nextval,134532,'San Francisco','Lombard Street',6);
```

	CENTER_ID	POSTAL_CODE	CITY	STREET	MANAGER_ID
1	1	127324	New York	Maiden Lane	1
2	2	124856	Chicago	Prairie Avenue	2
3	3	138453	Los Angeles	Sunset Boulevard	3
4	4	178513	Miami	Montgomery Street	4
5	5	178963	Seattle	First Avenue	5
6	6	134532	San Francisco	Lombard Street	6

```
insert into teachers values
(teacher_id_seq.nextval,'Susan','Connor','0761714909','susanconnor@email.com',to_date('2018-02-23','yyyy-mm-dd'),1);

insert into teachers values
(teacher_id_seq.nextval,'Margaret','Adelman','0765466544','margaretadelman@email.com',to_date('2018-07-13','yyyy-mm-dd'),1);

insert into teachers values
(teacher_id_seq.nextval,'Christinne','Novak','0762321239','christinnenovak@email.com',to_date('2017-05-21','yyyy-mm-dd'),5);

insert into teachers values
(teacher_id_seq.nextval,'Jeff','Johnsonn','0752354100','jeffjohnsonn@email.com',to_date('2019-03-23','yyyy-mm-dd'),3);

insert into teachers values
(teacher_id_seq.nextval,'Melvis','Forbin','0754213200','melvisforbin@email.com',to_date('2020-08-23','yyyy-mm-dd'),4);

insert into teachers values
(teacher_id_seq.nextval,'Mike','James','0754254330','mikejames@email.com','23-JUN-20',2);

insert into teachers values
(teacher_id_seq.nextval,'Elena','Miller','0758525850','elenamiller@email.com','23-JAN-19',6);
```

	TEACHER_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER	EMAIL	HIRE_DATE	CENTER_ID
1	1	Susan	Connor	0761714909	susanconnor@email.com	23-FEB-18	1
2	2	Margaret	Adelman	0765466544	margaretadelman@email.com	13-JUL-18	1
3	3	Christinne	Novak	0762321239	christinnenovak@email.com	21-MAY-17	5
4	4	Jeff	Johnsonn	0752354100	jeffjohnsonn@email.com	23-MAR-19	3
5	5	Melvis	Forbin	0754213200	melvisforbin@email.com	23-AUG-20	4
6	6	Mike	James	0754254330	mikejames@email.com	23-JUN-20	2
7	7	Elena	Miller	0758525850	elenamiller@email.com	23-JAN-19	6

```
insert into subjects values (subject_id_seq.nextval,'English',20);
insert into subjects values (subject_id_seq.nextval,'Mathematics',25);
insert into subjects values (subject_id_seq.nextval,'Chemistry',25);
insert into subjects values (subject_id_seq.nextval,'Economics',25);
```

	SUBJECT_ID	SUBJECT_NAME	MEETING_PRICE
1	1	English	20
2	2	Mathematics	25
3	3	Chemistry	25
4	4	Economics	25

```
insert into aux values('Monday');
insert into aux values('Tuesday');
insert into aux values('Wednesday');
insert into aux values('Thursday');
insert into aux values('Friday');
insert into aux values('Saturday');
insert into aux values('Sunday');
```

	DAY
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday
7	Sunday

```
insert into students
values(student_id_seq.nextval,'Nicole','Green','0741258962','nicolegreen@email.com');
insert into students
values(student_id_seq.nextval,'Lena','Smith','075896325','lenasmith@email.com');
insert into students
 values(student_id_seq.nextval,'Paul','Miller','0786523698','paulmiller@email.com');
insert into students
values(student_id_seq.nextval,'David','King','0784210025','davidking@email.com');
insert into students
values(student_id_seq.nextval,'Emma','Jones','0725002510','emmajones@email.com');
insert into students
values(student_id_seq.nextval,'Ashley','HILL','0727852150','ashleyhill@email.com');
insert into students
values(student_id_seq.nextval,'BETTY','GRACE','0728521422','bettygrace@email.com');
```

	STUDENT_ID	FIRST_...	LAST_NAME	PHONE_NUMBER	EMAIL
1	1	Nicole	Green	0741258962	nicolegreen@email.com
2	2	Lena	Smith	075896325	lenasmith@email.com
3	3	Paul	Miller	0786523698	paulmiller@email.com
4	4	David	King	0784210025	davidking@email.com
5	5	Emma	Jones	0725002510	emmajones@email.com
6	6	Ashley	HILL	0727852150	ashleyhill@email.com
7	7	BETTY	GRACE	0728521422	bettygrace@email.com

```
insert into meetings
values(meeting_id_seq.nextval,1,3,1,'Monday','14:00');
insert into meetings
values(meeting_id_seq.nextval,2,1,3,'Wednesday','14:00');
insert into meetings
values(meeting_id_seq.nextval,2,2,2,'Monday','13:30');
insert into meetings
values(meeting_id_seq.nextval,3,3,4,'Tuesday','15:00');
insert into meetings
values(meeting_id_seq.nextval,3,4,1,'Sunday','11:00');
insert into meetings
values(meeting_id_seq.nextval,4,4,1,'Tuesday','09:00');
insert into meetings
values(meeting_id_seq.nextval,1,2,2,'Friday','09:30');
insert into meetings
values(meeting_id_seq.nextval,4,1,3,'Thursday','16:30');
insert into meetings
values(meeting_id_seq.nextval,2,6,7,'Thursday','16:00');
insert into meetings
values(meeting_id_seq.nextval,2,6,7,'Thursday','16:00');
insert into meetings
values(meeting_id_seq.nextval,2,7,7,'Thursday','12:00');
```

	MEETING_ID	TEACHER_ID	STUDENT_ID	SUBJECT_ID	DAY	START_AT
1	1	1	3	1	Monday	14:00
2	2	3	1	2	Wednesday	14:00
3	3	2	2	2	Monday	13:30
4	4	4	3	3	Tuesday	15:00
5	5	1	4	3	Sunday	11:00
6	6	1	4	4	Tuesday	09:00
7	7	2	2	1	Friday	09:30
8	8	3	1	4	Thursday	16:30
9	9	7	6	2	Thursday	16:00
10	10	7	7	2	Thursday	12:00

6. Definiţi un subprogram stocat care să utilizeze un tip de colecţie studiat. Apelaţi subprogramul.

Subprogramul numara cati elevi fac meditatii la cel putin 2 materii intr-un oras dat ca parametru. Tratati cazul in care orasul dat ca parametru nu exista, respectiv cazul in care in orasul dat nu invata niciun elev. Inserati in tabelul info_ANB informatiile corespunzatoare fiecarui caz determinat de valoarea data pentru parametru.

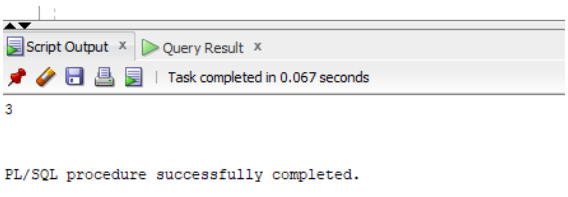
```
create table info(
   "Numar elevi" number(3),
   eroare varchar(60));

create or replace FUNCTION f6_anb (
   oras centers.city%TYPE
) RETURN NUMBER AS
   numar NUMBER;
   oras_inexistent EXCEPTION;
   elevi_inexistenti EXCEPTION;
BEGIN
   SELECT  COUNT(city)
   INTO numar
   FROM centers
   WHERE city = oras;
   IF numar = 0 THEN
      RAISE  oras_inexistent;
   END IF;
   SELECT COUNT(s.last_name)
   INTO numar
   FROM students   s JOIN meetings   m ON ( s.student_id = m.student_id )
                     JOIN teachers   t ON ( m.teacher_id = t.teacher_id )
                     JOIN centers    c ON ( t.center_id = c.center_id )
   WHERE c.city = oras;
   IF ( numar = 0 ) THEN
      RAISE elevi_inexistenti;
   END IF;
   SELECT  count(distinct s.last_name)
   INTO numar
   FROM students   s JOIN meetings   m ON ( s.student_id = m.student_id )
                     JOIN teachers   t ON ( m.teacher_id = t.teacher_id )
                     JOIN centers    c ON ( t.center_id = c.center_id )
   WHERE city = oras AND ( SELECT COUNT(meeting_id)
                           FROM meetings mm
                           WHERE mm.student_id = s.student_id) >= 2;
   INSERT INTO info VALUES ( numar, NULl);
   RETURN numar;
EXCEPTION
   WHEN oras_inexistent THEN
      INSERT INTO info VALUES (0,'Nu exista orasul dat');
      COMMIT;
      raise_application_error(-20000, 'There is no center in this city');
   WHEN elevi_inexistenti THEN
      INSERT INTO info VALUES (0, ‘Nu exista elevi in orasul dat');
     COMMIT;
      raise_application_error(-20001, 'There are no students in this city');
   WHEN OTHERS THEN
      INSERT INTO info VALUES (0,'Alta eroare');
      COMMIT;
      raise_application_error(-20002, 'Another error');
```

END;/

EXEMPLE:

```
begin
   DBMS_OUTPUT.PUT_LINE(f6_ANB('New York'));
end;
/
```




```
begin
   DBMS_OUTPUT.PUT_LINE(f6_ANB('Oxford'));
end;
/
```

```
Error starting at line : 282 in command -
begin
    DBMS_OUTPUT.PUT_LINE(f6_ANB('Oxford'));
end;
Error report -
ORA-20000: There is no center in this city
ORA-06512: at "HOMEUSER.F6_ANB", line 65
ORA-06512: at line 2
20000. 00000 -  "%s"
*Cause:    The stored procedure 'raise_application_error'
           was called which causes this error to be generated.
*Action:   Correct the problem as described in the error message or contact
           the application administrator or DBA for more information.
```

```
begin
   DBMS_OUTPUT.PUT_LINE(f6_ANB('Miami'));
end;
/
```

```
Error starting at line : 284 in command -
begin
    DBMS_OUTPUT.PUT_LINE(f6_ANB('Miami'));
end;
Error report -
ORA-20001: There are no students in this city
ORA-06512: at "HOMEUSER.F6_ANB", line 73
ORA-06512: at line 2
```

Dupa apelarea functiei f6_ANB pentru cele 3 orase, in tabelul info se vor insera datele corespunzatoare.

```
Select *
from info;
```

	Numar elevi	EROARE
1	3	(null)
2	0	Nu exista orasul dat
3	0	Nu exista elevi in orasul dat

7. Definiţi un subprogram stocat care să utilizeze un tip de cursor studiat. Apelați subprogramul.

Subprogramul obtine toate sedintele care se tin intr-o zi data ca parametru. Se vor trata cazurile in care ziua nu este corect scrisa sau nu exista sedinte in ziua respectiva.

```
CREATE OR REPLACE PROCEDURE p7_ANB(v_day aux.day%type)
IS
   CURSOR meeting IS
      SELECT meeting_id, student_id, m.teacher_id, subject_id, c.center_id
      FROM meetings m join teachers t on (m.teacher_id = t.teacher_id)
                  join centers  c on (t.center_id  = c.center_id)
      Where v_day = m.day;
       nr number(2);
       nicio_sedinta EXCEPTION;
       zi_inexistenta EXCEPTION;



BEGIN
      nr:=0;
      SELECT COUNT(v_day)
       INTO nr
      FROM aux
      WHERE v_day = day;

      if nr = 0 then
         raise zi_inexistenta;
      end if;
      nr:=0;
      FOR i in meeting LOOP
         DBMS_OUTPUT.PUT_LINE('Meeting id: ' || i.meeting_id);
         DBMS_OUTPUT.PUT_LINE('Teacher: ' || i.teacher_id );
         DBMS_OUTPUT.PUT_LINE('Student: ' || i.student_id );
         DBMS_OUTPUT.PUT_LINE('Subject: ' ||i.subject_id);
         DBMS_OUTPUT.PUT_LINE('Center: ' || i.center_id);
         DBMS_OUTPUT.NEW_LINE();
         nr:=nr+1;
      END LOOP;
      if nr = 0 then
         raise nicio_sedinta;
      end if;

      EXCEPTION
   WHEN nicio_sedinta THEN
         raise_application_error(-20003, 'No meetings on this day');
   WHEN zi_inexistenta THEN
         raise_application_error(-20004, 'This day does not exist');
   WHEN OTHERS THEN
      raise_application_error(-20005, 'Another error');

END;
/
```

EXEMPLE:

```
begin
   p7_ANB('Luni');
end;
/
```

```
Error starting at line : 347 in command -
begin
    p7_ANB('Luni');
end;
Error report -
ORA-20004: This day does not exist
ORA-06512: at "HOMEUSER.P7_ANB", line 42
ORA-06512: at line 2
```

```
begin
   p7_ANB('Monday');
end;
/
```

```
    Meeting id: 1
    Teacher: 1
    Student: 3
    Subject: 1
    Center: 1

    Meeting id: 3
    Teacher: 2
    Student: 2
    Subject: 2
    Center: 1
```

```
begin
  p7_ANB('Saturday');
end;
/
```

```
Error starting at line : 355 in command -
begin
    p7_ANB('Saturday');
end;
Error report -
ORA-20003: No meetings on this day
ORA-06512: at "HOMEUSER.P7_ANB", line 40
ORA-06512: at line 2
```

8. Definiți un subprogram stocat de tip funcție care să utilizeze 3 dintre tabelele definite. Tratați toate excepțiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

Subprogramul insereaza in tabelul info2 toti elevii care fac meditatii cu profesorul cu id-ul dat ca parametru. Se trateaza urmatoarele cazuri:nu exista profesorul cu id-ul respectiv;niciun elev nu face pregatire cu profesorul cu id-ul respectiv.

```
create table info2(
   "Numar sedinte" number(3),
   eroare varchar(60)
   );

create or replace function f8_ANB(id teachers.teacher_id%type)
   RETURN number AS
   numar NUMBER;
   profesor_inexistent EXCEPTION;
   elevi_inexistenti EXCEPTION;
BEGIN
   select count(teacher_id)
   into numar
   from teachers
   where teacher_id = id;

   if numar = 0 then
      raise profesor_inexistent;
   end if;

   select count(distinct s.student_id)
   into numar
   from students s
   join meetings m on (s.student_id = m.student_id)
   join teachers t on (t.teacher_id = m.teacher_id)
   where t.teacher_id = id;

   if numar = 0 then
      raise elevi_inexistenti;
   end if;

   insert into info2 values ( numar,NULL);
   commit;
   return numar;

   EXCEPTION
   WHEN profesor_inexistent THEN
      INSERT INTO info2 VALUES (0, 'Nu exista profesorul cu id-ul dat');
      COMMIT;
      raise_application_error(-20006, 'This teacher doesn''t exist');
   WHEN elevi_inexistenti THEN
      INSERT INTO info2 VALUES (0, 'Profesorul nu are sedinte');
      COMMIT;
      raise_application_error(-20007, 'This teacher has no meetings');
   WHEN OTHERS THEN
      INSERT INTO info2 VALUES (0, 'Alta eroare');
      COMMIT;
      raise_application_error(-20008, 'Another error');
END;
```

/

EXEMPLE:

```
begin
   DBMS_OUTPUT.PUT_LINE(f8_ANB(1));
end;
/
```

```
2


PL/SQL procedure successfully completed.
```

```
begin
   DBMS_OUTPUT.PUT_LINE(f8_ANB(8));
end;
/
```

```
Error starting at line : 428 in command -
begin
    DBMS_OUTPUT.PUT_LINE(f8_ANB(8));
end;
Error report -
ORA-20006: This teacher doesn't exist
ORA-06512: at "HOMEUSER.F8_ANB", line 37
ORA-06512: at line 2
```

```
begin
   DBMS_OUTPUT.PUT_LINE(f8_ANB(5));
end;
/
```

```
Error starting at line : 432 in command -
begin
    DBMS_OUTPUT.PUT_LINE(f8_ANB(5));
end;
Error report -
ORA-20007: This teacher has no meetings
ORA-06512: at "HOMEUSER.F8_ANB", line 45
ORA-06512: at line 2
```

Dupa apelarea functiei in cele 3 cazuri, se vor insera in tabela info2 datele corespunzatoare.

EXEMPLU:

```
select *
from info2;
```

	Numar sedinte	EROARE
1	2	(null)
2	0	Nu exista profesorul cu id-ul dat
3	0	Profesorul nu are sedinte

9. Definiţi un subprogram stocat de tip procedură care să utilizeze 5 dintre tabelele definite. Trataţi toate excepţiile care pot apărea. Apelaţi subprogramul astfel încât să evidenţiaţi toate cazurile tratate.

Subprogramul primeste 2 parametri (id,day) si afiseaza toate sedintele care au loc in ziua day in centrul al carui manager are id-ul id.

```
create or replace procedure p9_ANB(v_id managers.manager_id%type, v_day aux.day%type)
IS
   CURSOR meets IS
      SELECT meeting_id, (ss.first_name ||' ' || ss.last_name) as Student,(t.first_name ||' '||t.last_name) as
Teacher,s.subject_name,m.subject_id
      FROM meetings m join teachers t on (m.teacher_id = t.teacher_id)
                  join centers  c on (t.center_id  = c.center_id)
                  join managers mg on (c.manager_id = mg.manager_id)
                  join subjects s on (s.subject_id = m.subject_id)
                  join students ss on (m.student_id = ss.student_id)
      Where v_day = m.day and mg.manager_id = v_id;
      nr number(2);
      nicio_sedinta EXCEPTION;
      manager_inexistent EXCEPTION;
      zi_inexistenta  EXCEPTION;
BEGIN
      SELECT COUNT(v_day)
      INTO nr
      FROM aux
      WHERE v_day = day;
      if nr = 0 then raise zi_inexistenta;
      end if;
      SELECT COUNT(manager_id)
      into nr
      from managers
      where manager_id = v_id;
      if nr = 0 then raise manager_inexistent;
      end if;
      nr:=0;
      for i in meets loop
         DBMS_OUTPUT.PUT_LINE('Meeting id: ' || i.meeting_id);
         DBMS_OUTPUT.PUT_LINE('Teacher: ' || i.Student );
         DBMS_OUTPUT.PUT_LINE('Student: ' || i.Teacher );
         DBMS_OUTPUT.PUT_LINE('Subject: ' ||i.subject_name);
         DBMS_OUTPUT.NEW_LINE();
         nr:=nr+1;
      END LOOP;
      if nr = 0 then   raise nicio_sedinta;
      end if;
      Exception
      WHEN nicio_sedinta THEN
         raise_application_error(-20009, 'No meetings found');
      WHEN zi_inexistenta THEN
         raise_application_error(-20010, 'This day does not exist');
      WHEN manager_inexistent then
         raise_application_error(-20011,'No manager has this id');
      WHEN OTHERS THEN
         raise_application_error(-20012, 'Another error');

END;
/
```

EXEMPLE:

```
BEGIN
   p9_ANB(2,'Sunday');
end;
/
```

```
Error starting at line : 506 in command -
BEGIN
    p9_ANB(2,'Sunday');
end;
Error report -
ORA-20009: No meetings found
ORA-06512: at "HOMEUSER.P9_ANB", line 51
ORA-06512: at line 2
```

```
BEGIN
   p9_ANB(1,'Monday');
end;
/
```

```
Meeting id: 1
Teacher: Paul Miller
Student: Susan Connor
Subject: English

Meeting id: 3
Teacher: Lena Smith
Student: Margaret Adelman
Subject: Mathematics
```

```
BEGIN
   p9_ANB(8,'Friday');
end;
/
```

```
Error starting at line : 516 in command -
BEGIN
    p9_ANB(8,'Friday');
end;
Error report -
ORA-20011: No manager has this id
ORA-06512: at "HOMEUSER.P9_ANB", line 55
ORA-06512: at line 2
```

```
BEGIN
   p9_ANB(2,'Vineri');
end;
/
```

```
Error starting at line : 521 in command -
BEGIN
    p9_ANB(2,'Vineri');
end;
Error report -
ORA-20010: This day does not exist
ORA-06512: at "HOMEUSER.P9_ANB", line 53
ORA-06512: at line 2
```

10. Definiţi un trigger de tip LMD la nivel de comandă. Declanşaţi trigger-ul.

Triggerul permite modificarile in tabela meetings doar în intervalul de ore 8:00 - 20:00, de luni până vineri.

```
CREATE OR REPLACE TRIGGER trig10
 BEFORE INSERT OR UPDATE OR DELETE ON meetings
BEGIN
IF (TO_CHAR(SYSDATE,'D') = 1)
OR (TO_CHAR(sysdate,'D') = 7)
OR (TO_CHAR(SYSDATE,'HH24') NOT  BETWEEN 8 AND 20)
THEN  RAISE_APPLICATION_ERROR(-20013,'Can''t update this table now');
END IF;
END;
/
```

EXEMPLU

```
insert into meetings values(
meeting_id_seq.nextval,
2,7,7,'Thursday','12:00');
```

```
Trigger TRIG10 compiled
```

```
Error starting at line : 542 in command -
insert into meetings values(meeting_id_seq.nextval,2,7,7,'Thursday','12:00')
Error report -
ORA-20013: Can't update this table now
ORA-06512: at "HOMEUSER.TRIG10", line 6
ORA-04088: error during execution of trigger 'HOMEUSER.TRIG10'
```

11. Definiţi un trigger de tip LMD la nivel de linie. Declanşaţi trigger-ul.

Triggerul nu permite adaugarea sedintelor in intervalul orar 18:00-08:00.

```
CREATE OR REPLACE TRIGGER trig11
 BEFORE UPDATE OF start_at ON meetings
 FOR EACH ROW
BEGIN
 IF (to_char(:NEW.start_at) < '08:00') then
      RAISE_APPLICATION_ERROR(-20012,'Meeting can''t start at this time');
END IF;

 IF (to_char(:NEW.start_at) > '18:00') then
      RAISE_APPLICATION_ERROR(-20013,'Meeting can''t start at this time');
END IF;
END;
/
```

```
Trigger TRIG11 compiled
```

Exemplu:

```
UPDATE MEETINGS
```

```
Error starting at line : 562 in command -
UPDATE MEETINGS
SET START_AT = '00:00'
WHERE MEETING_ID = 8
Error report -
ORA-20012: Meeting can't start at this time
ORA-06512: at "HOMEUSER.TRIG11", line 3
ORA-04088: error during execution of trigger 'HOMEUSER.TRIG11'
```

```
SET START_AT = '00:00'
WHERE MEETING_ID = 8;
```

12. Definiți un trigger de tip LDD. Declanșați trigger-ul.

Doar utilizatorii cu numele 'Manager' pot modifica, sterge sau adauga tabele

```
CREATE OR REPLACE TRIGGER trig12
BEFORE ALTER OR CREATE OR DROP ON schema
BEGIN
IF USER <> UPPER('manager') THEN
   RAISE_APPLICATION_ERROR(-20014,'You  don''t have the right to  modify, create or delete tables');
end if;
END;
/
```

EXEMPLU

```
create table ex(
        nr number
                );
```

```
Trigger TRIG12 compiled


Error starting at line : 581 in command -
create table ex(nr number)
Error report -
ORA-00604: error occurred at recursive SQL level 1
ORA-20014: You  don't have the right to  modify, create or delete tables
ORA-06512: at line 3
00604. 00000 -  "error occurred at recursive SQL level %s"
*Cause:    An error occurred while processing a recursive SQL statement
           (a statement applying to internal dictionary tables).
*Action:   If the situation described in the next error on the stack
           can be corrected, do so; otherwise contact Oracle Support.
```

13. Definiți un pachet care să conțină toate obiectele definite în cadrul proiectului.

```
CREATE OR REPLACE PACKAGE package13
AS
   FUNCTION f6 (
      oras centers.city%TYPE
   ) RETURN NUMBER;

   PROCEDURE p7 (
      v_day aux.day%TYPE
   );

   FUNCTION f8 (
      id teachers.teacher_id%TYPE
   ) RETURN NUMBER;

   PROCEDURE p9 (
      v_id    managers.manager_id%TYPE,
      v_day   aux.day%TYPE
   );

END package13;
/
```

```
Package PACKAGE13 compiled


Package Body PACKAGE13 compiled
```

```
CREATE OR REPLACE PACKAGE BODY package13 as
   FUNCTION f6 (
      oras centers.city%TYPE
   ) RETURN NUMBER AS
      numar NUMBER;
      oras_inexistent EXCEPTION;
      elevi_inexistenti EXCEPTION;
   BEGIN
      SELECT COUNT(city)
      INTO numar
      FROM centers
      WHERE city = oras;
      IF numar = 0 THEN
         RAISE oras_inexistent;
      END IF;
      SELECT COUNT(s.last_name)
      INTO numar
      FROM students   s JOIN meetings   m ON ( s.student_id = m.student_id )
         JOIN teachers   t ON ( m.teacher_id = t.teacher_id )
         JOIN centers    c ON ( t.center_id = c.center_id )
      WHERE c.city = oras;
      IF ( numar = 0 ) THEN
         RAISE elevi_inexistenti;
      END IF;
      SELECT COUNT(DISTINCT s.last_name)
      INTO numar
      FROM students   s JOIN meetings   m ON ( s.student_id = m.student_id )
         JOIN teachers   t ON ( m.teacher_id = t.teacher_id )
         JOIN centers    c ON ( t.center_id = c.center_id )
      WHERE city = oras AND ( SELECT COUNT(meeting_id)
                              FROM meetings mm
                              WHERE mm.student_id = s.student_id ) >= 2;
      INSERT INTO info VALUES ( numar,NULL);
      RETURN numar;
   EXCEPTION
      WHEN oras_inexistent THEN
         INSERT INTO info VALUES (0,'Nu exista orasul dat' );
         COMMIT;
         raise_application_error(-20000, 'There is no center in this city');
      WHEN elevi_inexistenti THEN
         INSERT INTO info VALUES (0,'Nu exista elevi in orasul dat');
         COMMIT;
         raise_application_error(-20001, 'There are no students in this city');
      WHEN OTHERS THEN
         INSERT INTO info VALUES (0,'Alta eroare');
         COMMIT;
         raise_application_error(-20002, 'Another error');
   END f6;


   begin
   DBMS_OUTPUT.PUT_LINE(package13.f6('New York'));
end;
/
```

```
3

PL/SQL procedure successfully completed.
```

```
PROCEDURE p7 (
      v_day aux.day%TYPE
   ) IS

      CURSOR meeting IS
      SELECT meeting_id, student_id, m.teacher_id, subject_id, c.center_id
      FROM meetings   m JOIN teachers   t ON ( m.teacher_id = t.teacher_id )
                        JOIN centers    c ON ( t.center_id = c.center_id )
      WHERE v_day = m.day;
      nr NUMBER(2);
      nicio_sedinta EXCEPTION;
      zi_inexistenta EXCEPTION;
   BEGIN
      nr := 0;
      SELECT COUNT(v_day)
      INTO nr
      FROM aux
      WHERE v_day = day;
      IF nr = 0 THEN
         RAISE zi_inexistenta;
      END IF;
      nr := 0;
      FOR i IN meeting LOOP
         dbms_output.put_line('Meeting id: ' || i.meeting_id);
         dbms_output.put_line('Teacher: ' || i.teacher_id);
         dbms_output.put_line('Student: ' || i.student_id);
         dbms_output.put_line('Subject: ' || i.subject_id);
         dbms_output.put_line('Center: ' || i.center_id);
         dbms_output.new_line();
         nr := nr + 1;
      END LOOP;
      IF nr = 0 THEN
         RAISE nicio_sedinta;
      END IF;
   EXCEPTION
      WHEN nicio_sedinta THEN
         raise_application_error(-20003, 'No meetings on this day');
      WHEN zi_inexistenta THEN
         raise_application_error(-20004, 'This day does not exist');
      WHEN OTHERS THEN
         raise_application_error(-20005, 'Another error');
   END p7;
```

EXEMPLE:

```
begin
   package13.p7('Luni');
end;/
```

```
Error starting at line : 316 in command -
begin
    package13.p7('Luni');
end;
Error report -
ORA-20004: This day does not exist
ORA-06512: at "HOMEUSER.PACKAGE13", line 138
ORA-06512: at line 2
```

```
begin
   package13.p7('Monday');
end;/
```

```
Meeting id: 12
Teacher: 1
Student: 3
Subject: 1
Center: 1

Meeting id: 1
Teacher: 2
Student: 2
Subject: 2
Center: 1
```

```
FUNCTION f8 (
      id teachers.teacher_id%TYPE
   ) RETURN NUMBER AS
      numar NUMBER;
      profesor_inexistent EXCEPTION;
      elevi_inexistenti EXCEPTION;
   BEGIN
      SELECT COUNT(teacher_id)
      INTO numar
      FROM teachers
      WHERE teacher_id = id;
      IF numar = 0 THEN
         RAISE profesor_inexistent;
      END IF;
      SELECT COUNT(DISTINCT s.student_id)
      INTO numar
      FROM
         students   s
         JOIN meetings   m ON ( s.student_id = m.student_id )
         JOIN teachers   t ON ( t.teacher_id = m.teacher_id )
      WHERE
         t.teacher_id = id;
      IF numar = 0 THEN
         RAISE elevi_inexistenti;
      END IF;
      INSERT INTO info2 VALUES (numar, NULL);
      COMMIT;
      RETURN numar;
   EXCEPTION
      WHEN profesor_inexistent THEN
         INSERT INTO info2 VALUES (0,'Nu exista profesorul cu id-ul dat');
         COMMIT;
         raise_application_error(-20006, 'This teacher doesn''t exist');
      WHEN elevi_inexistenti THEN
         INSERT INTO info2 VALUES ( 0,'Profesorul nu are sedinte');
         COMMIT;
         raise_application_error(-20007, 'This teacher has no meetings');
      WHEN OTHERS THEN
         INSERT INTO info2 VALUES ( 0,'Alta eroare');

         COMMIT;
         raise_application_error(-20008, 'Another error');
   END f8;
```

EXEMPLE:

```
1
```
Save File (Ctrl+S)

```
PL/SQL procedure successfully completed.
```

```
begin
   DBMS_OUTPUT.PUT_LINE(package13.f8('2'));
end;/
```

```
begin
   DBMS_OUTPUT.PUT_LINE(package13.f8('8'));
end;/
```

```
Error starting at line : 328 in command -
begin
    DBMS_OUTPUT.PUT_LINE(package13.f8('8'));
end;
Error report -
ORA-20006: This teacher doesn't exist
```

```
begin
   DBMS_OUTPUT.PUT_LINE(package13.f8('5'));
end;/
```

```
Error starting at line : 328 in command -
begin
    DBMS_OUTPUT.PUT_LINE(package13.f8('5'));
end;
Error report -
ORA-20007: This teacher has no meetings
```

```
PROCEDURE p9(
      v_id    managers.manager_id%TYPE,
      v_day   aux.day%TYPE
   ) IS
      CURSOR meets IS
      SELECT meeting_id,student_id,m.teacher_id,subject_name,s.subject_id
      FROM meetings   m JOIN teachers   t ON ( m.teacher_id = t.teacher_id )
         JOIN centers    c ON ( t.center_id = c.center_id )
         JOIN managers   mg ON ( c.manager_id = mg.manager_id )
         JOIN subjects   s ON ( s.subject_id = m.subject_id )
      WHERE v_day = m.day  AND mg.manager_id = v_id;
      nr NUMBER(2);
      nicio_sedinta EXCEPTION;
      manager_inexistent EXCEPTION;
      zi_inexistenta EXCEPTION;
   BEGIN
      SELECT  COUNT(v_day)
      INTO nr
      FROM  aux
      WHERE v_day = day;
      IF nr = 0 THEN RAISE zi_inexistenta;
      END IF;
      SELECT COUNT(manager_id) INTO nr
      FROM managers
      WHERE  manager_id = v_id;
      IF nr = 0 THEN RAISE manager_inexistent;
      END IF;
      nr := 0;
      FOR i IN meets LOOP
         dbms_output.put_line('Meeting id: ' || i.meeting_id);
         dbms_output.put_line('Teacher id: ' || i.teacher_id);
         dbms_output.put_line('Student id: ' || i.student_id);
         dbms_output.put_line('Subject : ' || i.subject_name);
         dbms_output.new_line();
         nr := nr + 1;
      END LOOP;
      IF nr = 0 THEN RAISE nicio_sedinta;
      END IF;
   EXCEPTION
      WHEN nicio_sedinta THEN raise_application_error(-20009, 'No meetings found');
      WHEN zi_inexistenta THEN  raise_application_error(-20010, 'This day does not exist');
      WHEN manager_inexistent THEN   raise_application_error(-20011, 'No manager has this id');
      WHEN OTHERS THEN  raise_application_error(-20012, 'Another error');
   END p9;
 END package13;
 /
```

EXEMPLE:

```
BEGIN
   p9_ANB(2,'Sunday');
end;/
```

```
Error report -
ORA-20009: No meetings found
ORA-06512: at "HOMEUSER.P9_ANB", line 51
ORA-06512: at line 2
```

```
BEGIN
   p9_ANB(1,'Monday');
```

```
Meeting id: 12
Teacher: Paul Miller
Student: Susan Connor
Subject: English

Meeting id: 14
Teacher: Lena Smith
Student: Margaret Adelman
Subject: Mathematics
```

```
end;/
```

```
Error starting at line : 333 in command -
BEGIN
    p9_ANB(8,'Friday');
end;
Error report -
ORA-20011: No manager has this id
ORA-06512: at "HOMEUSER.P9_ANB", line 55
ORA-06512: at line 2
```

```
BEGIN
   p9_ANB(8,'Friday');
end;/
```

```
Error starting at line : 338 in command -
BEGIN
    p9_ANB(2,'Vineri');
end;
Error report -
ORA-20010: This day does not exist
ORA-06512: at "HOMEUSER.P9_ANB", line 53
ORA-06512: at line 2
```

```
BEGIN
   p9_ANB(2,'Vineri');
end;/
```