

# מיני פרויקט בבסיסי נתונים שלב 3

## תוכן עניינים

2	קוד ראשון – השמת מרצים כרכזי קורסים:
5	קוד שני – ריקון מעונות:
9	שינויים בבסיס הנתונים:

## קוד ראשון – השמת מרצים כרכזי קורסים:

בשביל קוד זה, כתבנו שתי פונקציות, פרוצדורה אחת, ותוכנית ראשית. הפונקציה הראשונה, היא `check_free_lecturer`. בהינתן מרצה, וקורס מסוים כלשהו, היא בודקת האם המרצה הזה פנוי בסמסטר הזה להיות רכז של הקורס הזה. לשם כך, היא בודקת קודם כל שהמרצה הזה עובד במחלקה של הקורס, ואח"כ בודקת שהוא לא מלמד כבר יותר מקורס אחד נוסף באותו סמסטר. קוד התוכנית:

```
--given a course and a lecturer, check if the lecturer is free to teach the course
create or replace function check_free_lecturer
(cy number, sem number, ci number, li number)
return number is
    cnd number;
    numCrs number;
    nm varchar2(20);
    d_ser varchar2(20);
    flag boolean := false;
    --the departments name that lecturer is working for
    cursor dpnm is
        ((select w.dep_name
        from working_for w
        where w.l_id = li)
        union
        (select d.dep_name
        from department d
        where d.l_id = li));
begin
    --get the department name of the course
    select c.dep_name into nm
    from course_type c
    where c.c_id=ci
    fetch first 1 rows only;

    --check if the lecturer works for that department
    open dpnm;
    loop
        fetch dpnm into d_ser;
        exit when dpnm%notfound;
        if d_ser=nm then flag := true; end if;
    end loop;
    close dpnm;
    if not flag then return 0; end if;

    --check that he doesnt teach too much courses in that semester (at most one more course)
    select count(*) into numCrs
    from teaches t
    where t.course_year=cy and t.semester=sem and t.c_id!=ci and t.l_id=li;
    if numCrs >= 2 then return 0; end if;

    return 1;

exception
    when others then
        raise;
end;
/
```

הפונקציה השנייה שהגדרנו, היא `get_free_lecturers`, המחזירה בהינתן קורס מסוים, את רשימת המרצים שיכולים לשמש בתור רכז הקורס שלו. הפונקציה מחזירה `ref cursor`, לרשימת ה-`l_id` של המרצים המתאימים. היא קוראת לפונקציה הקודמת שהגדרנו, `check_free_lecturer`, על מנת לבדוק אם מרצה הוא מתאים או לא. קוד התוכנית:

```

create or replace function get_free_lecturers(cy number, sem number, ci number)
return sys_refcursor is
    l_cursor sys_refcursor;
begin
    --send only the free lecturers, as a ref cursor
    open l_cursor for
        select l.l_id
        from lecturer l
        where (check_free_lecturer(cy,sem,ci,l.l_id)=1)
        order by dbms_random.value;
    return l_cursor;

    exception
        when others then
            raise;
end;
/

```

הגדרנו פרוצדורה שקראנו לה assign\_course\_coordinator. בהינתן קורס מסוים ומרצה, היא מגדירה את המרצה להיות רכז הקורס של הקורס הזה. הפרוצדורה בודקת שהוא עומד בתנאים, ולאחר מכן מעדכנת את בסיס הנתונים בהתאם. קוד התכנית:

```

create or replace procedure assign_course_coordinator
(ci number, cy number, sem number, li number) is
    not_good_lecturer exception;
begin
    --check that the given lecturer is free
    if (check_free_lecturer(cy,sem,ci,li)=0) then
        raise not_good_lecturer;
    end if;

    --update the coordinator of that course
    update course
    set l_id = li
    where (c_id=ci and semester=sem and course_year=cy);

    --update the table teaches, with the values of the lecturer and the course
    insert into SYSTEM.TEACHES (SEMESTER, COURSE_YEAR, C_ID, L_ID)
    values (sem, cy, ci, li);
    commit;

    exception
        when not_good_lecturer then
            rollback;
            raise_application_error(-20001, 'the lecturer is not free');
        when others then
            rollback;
            raise;
end;
/

```

לבסוף, כתבנו תכנית ראשית, בה הגדרנו את הפרמטרים של הקורס, קראנו לפונקציה `get_free_lecturers` כדי למצוא מרצה יחיד מתאים, הוצאנו מתוך `cursor` שקיבלו רשומה, וקראנו לפרוצדורה `assign_course_coodinator` כדי להקצות את המרצה שקיבלנו כרכז קורס לקורס המתאים, וכמובן עם טיפול בחריגות. קוד התכנית:

```
declare
    ci number := 14246;
    cy number := 2012;
    sem number := 2;
    li number;
    l_cursor sys_refcursor;
    no_free_lecturer exception;
begin
    --get one random free lecturer
    l_cursor := get_free_lecturers(cy, sem, ci);
    fetch l_cursor into li;

    --check that the cursor is not empty
    if l_cursor%notfound then raise no_free_lecturer; end if;

    --assign the lecturer to the course
    assign_course_coodinator(ci, cy, sem, li);

    --handle exceptions
exception
    when no_free_lecturer then
        dbms_output.put_line('there is no free lecturer to teach the course');
    when others then
        dbms_output.put_line('An error occurred: ' || sqlerrm);
end;
/
```

כעת, בדקנו את הפעלת התכנית. הרצנו פעמיים את הקוד הבא, בפעם הראשונה לפני הפעלת התכנית, ובפעם השנייה אחרי הפעלת התכנית:

```
select * from course
where c_id=14246 and course_year=2012 and semester=2;
```

לפני הפעלת התכנית, קיבלנו:

	SEMESTER	COURSE_YEAR	GRADE_TO_PASS	TEST_PERCENTAGE	EXRECISE_PERCENTAGE	C_ID	L_ID
▶ 1	2	2012	60	0.5	0.5	14246	6509965

לאחר מכן הפעלנו את התכנית, הרצנו שוב, וקיבלנו:

	SEMESTER	COURSE_YEAR	GRADE_TO_PASS	TEST_PERCENTAGE	EXRECISE_PERCENTAGE	C_ID	L_ID
▶ 1	2	2012	60	0.5	0.5	14246	1161454

כלומר כפי שאפשר לראות הוקצה מרצה חדש להיות רכז הקורס, כנדרש.

## קוד שני – ריקון מעונות:

בקוד זה, כתבנו שתי פונקציות, פרוצדורה אחת ותכנית ראשית. מטרת התכנית הראשית, היא לרוקן מעונות, כלומר בהינתן איזשהם מעונות להעביר את כל הסטודנטים הנמצאים בהם למעונות אחרים. כדי להפוך חלק זה ליותר מעניין, שינינו קצת את בסיס הנתונים, והוספנו עמודה של free\_beds\_num בטבלה dormitories, כדי שתשמור את כמות המיטות הפנויות בכל בניין של מעונות (נראה את הקוד ונסביר בהמשך). הפונקציה הראשונה שכתבנו, get\_one\_not\_full\_dormitory, מחזירה בהינתן שם של בניין מעונות, שם של איזשהו בניין מעונות רנדומלי אחר לא מלא, בשביל שיהיה אפשר להעביר אליו סטודנטים. קוד התכנית:

```
--get one not full dormitory
create or replace function get_one_not_full_dormitory(dnm varchar2)
return varchar2 is
    new_nm varchar2(20);
begin
    --check that it's not full, and that it's not the original dormitory
    select d_name into new_nm
    from dormitories
    where d_name!=dnm
        and free_beds_num > 0
    order by dbms_random.value
    fetch first 1 rows only;
    return new_nm;

    exception
        when others then
            raise;
end;
/
```

כתבנו את פונקציית העזר get\_student\_name\_from\_id, הנותנת בהינתן s\_id של סטודנט את השם שלו, בה השתמשנו לצרכי הדפסה בתכנית. קוד התכנית:

```
create or replace function get_student_name_from_id(sti number)
return varchar2 is
    nm varchar2(20);
begin
    select s_name into nm
    from student
    where s_id=sti;
    return nm;

    exception
        when others then
            raise;
end;
/
```

כתבנו את הפרוצדורה move\_dormitory, אשר בהינתן סטודנט כלשהו ובניין מעונות כלשהו מעבירה את הסטודנט לבניין המעונות הזה. הפרוצדורה בודקת שאכן קיים בניין מעונות כזה, וסטודנט כזה, ושיש מקום בבניין המעונות החדש, ואם כל הבדיקות מצליחות, מעבירה את הסטודנט הזה לבניין החדש ומעדכנת את הנתונים בבסיס הנתונים. קוד התכנית:

```
create or replace procedure move_dormitory
(sti number, dnm varchar2) is
  old_nm varchar2(20);
  full_dormitory exception;
  tmp number;
  snm varchar2(20);
begin
  select count(*) into tmp
  from student
  where s_id=sti;
  if tmp=0
  then raise value_error;
  end if;

  select count(*) into tmp
  from dormitories
  where d_name=dnm;
  if tmp=0
  then raise value_error;
  end if;

  select free_beds_num into tmp
  from dormitories
  where d_name=dnm;
  if tmp=0
  then raise full_dormitory;
  end if;

  select d_name into old_nm
  from student
  where s_id=sti;

  update dormitories
  set free_beds_num = free_beds_num+1
  where d_name=old_nm;

  update dormitories
  set free_beds_num = free_beds_num-1
  where d_name=dnm;

  update student
  set d_name = dnm
  where s_id=sti;

  commit;

  snm := get_student_name_from_id(sti);
  dbms_output.put_line('moved ' || snm || ' from ' || old_nm || ' to ' || dnm);

  exception
  when full_dormitory then
    raise_application_error(-20001, 'the dormitory is full');
  when others then
    rollback;
    raise;
end;
/
```

לבסוף, כתבנו תכנית ראשית, הקולטת מהמשתמש שם של בניין מעונות, ועבור כל הסטודנטים בבניין הזה מעבירה אותם לבניין מעונות אחר. קוד התכנית:

```
declare
--get the dormitory name to empty as parameter
dnm varchar2(20) := '&dorm_name';
--the students who live there
cursor dorm_students is
    select s_id from student where d_name=dnm;
sti number;
new_nm varchar2(20);
begin
--for each student, pick a random not full dormitory, and move him there
open dorm_students;
loop
    fetch dorm_students into sti;
    exit when dorm_students%notfound;
    --pick a dormitory
    new_nm := get_one_not_full_dormitory(dnm);
    --move the student to the new dormitory
    move_dormitory(sti,new_nm);
end loop;
close dorm_students;

--handle exceptions
exception
    when others then
        dbms_output.put_line('An error occurred: ' || sqlerrm);
end;
/
```

כעת בדקנו את התוכנית. הרצנו פעמיים את הקוד הבא:

```
select * from dormitories where d_name='Stuart';
select * from student where d_name = 'Stuart';
```

בפעם הראשונה קיבלנו:

		D_NAME	ROOMS_NUM	BEDS_NUM	FREE_BEDS_NUM
▶ 1	Stuart	...	3	12	7

  

	S_ID	S_NAME	PHONE_NUMBER	EMAIL	BIRTH_DATE	LEARNING_YEAR	D_NAME
▶ 1	2146662	Courtney Lewis	643-252-5319	garrettsteven@example.net	09/10/1998	...	2 Stuart
2	9299120	Scott Mahoney	367.811.4998x17209	nathanielmitchell@example.net	06/11/1998	...	5 Stuart
3	4060223	Jacob Ray	+1-821-239-3593	fmarsh@example.org	03/09/2004	...	4 Stuart
4	5868906	Vickie Powell	580-594-4358	hayesmargaret@example.net	04/04/1988	...	5 Stuart
5	8566038	Cody Fischer	(438)638-6078	thompsonangela@example.net	18/09/1999	...	3 Stuart

לאחר מכן הרצנו את התכנית, הכנסנו בתיבה את הפרמטר Stuart:

Variables

Name	Value
▶ dorm_name	Stuart

OK
Cancel
Clear



והרצנו שוב את שאילתות הselect, וכעת קיבלנו:

	D_NAME	ROOMS_NUM	BEDS_NUM	FREE_BEDS_NUM
▶ 1	Stuart	3	12	12

S_ID	S_NAME	PHONE_NUMBER	EMAIL	BIRTH_DATE	LEARNING_YEAR	D_NAME
------	--------	--------------	-------	------------	---------------	--------

כלומר הקוד אכן רוקן את בניין המעונות Stuart, כנדרש.

## שינויים בבסיס הנתונים:

כפי שכתבנו, עשינו שינוי קטן בבסיס הנתונים, והוספנו עוד עמודה לטבלה של dormitories, שתשמור את מספר המיטות הפנויות בכל בניין מעונות, וחישבנו אותו עבור כל בניין מעונות כדי לאתחל אותו עם הערך הנכון. מצורף קוד השינוי:

```
alter table dormitories
add(free_beds_num number);

update dormitories d
set d.free_beds_num = d.beds_num - (select count(*)
                                     from student s
                                     where s.d_name=d.d_name);

commit;
```

וכפי שכבר ראינו בתוכנית שהרצנו, השינוי עבד כנדרש.