بسمه تعالی
پاسخ تکلیف دوم درس پایگاه داده ها 1

1. برای هر یک از داده های زیر وارد کنید کدام نوع داده مناسب تر است:

نوع متغیر (char, varchar, int,)	داده	
char	سه حرف مخفف شده ماه های میلادی	
numeric	قیمت دلاری محصولات که همگی دو رقم اعشار دارند	
varchar	نام و نام خانوادگی کاربر	
varchar char	کد ملی کاربر	
float	ذخیره قیمت لحظه ای ارز های دیجیتال دلار	
int	تعداد بازدید یک ویدئو	

2. تفاوت (char(10) با (varchar(10) را توضيح دهيد. انتخاب كدام يك در چه حالاتي بهتر است؟

توضیحات کاملتر در صفحه 67 کتاب آمده است.

char (10): یک رشته از کاراکترها با طول ثابت 10 است.

varchar (10): یک رشته از کاراکترها با طول متغیر حداکثر 10 است.

اگر رشته هایی با طول بسیار متغیر مانند نام را ذخیره می کنید، از VARCHAR استفاده کنید، اگر طول همیشه یکسان است، از CHAR استفاده کنید زیرا از نظر اندازه کارآمدتر و همچنین سریعتر است.

- 3. دستورات زیر را با توجه به خواسته سوال در صورت امکان بازنویسی کنید:
 - a. بازنویسی بدون استفاده از دستور like

WHERE first name LIKE 'me%' AND last name LIKE '%avi'

```
WHERE LEFT(first_name,2) = 'me' AND RIGHT(last_name,3) = 'avi'
```

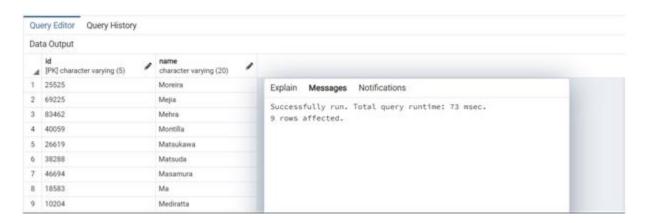
b. بازنویسی بدون اپراتور ||

SELECT first_name || last_name

SELECT CONCAT(first name, last name);

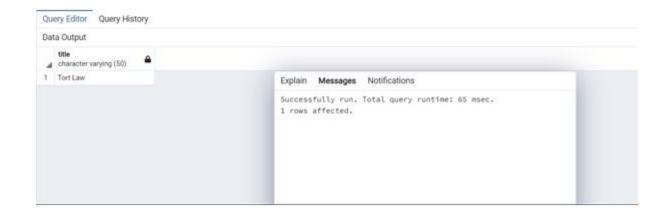
4. با توجه به پایگاه داده (University: Large Version) برای هر یک از موارد زیر دستور SQL مناسب بنویسید . a. شناسه و نام دانشجوبانی که نام آنها با M شروع و با a پایان می پابد.

SELECT id,name
FROM student
WHERE Name LIKE 'M%a'



d. نام دروسی را بیابید که در دانشکده های مهندسی ارائه شده باشند(نام آنها با Eng. ختم شوند) و در ترم پاییز 2009 ارائه شده باشند.

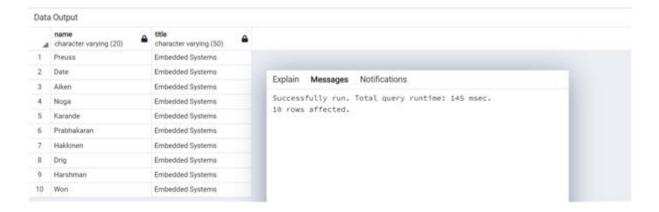
SELECT title
FROM section,course
where course.course_id=section.course_id AND year=2009
AND semester='Fall' AND dept_name LIKE '%Eng.'



c. زوج نام دانشجو و نام درس هایی را بیابید که دانشجو درس مورد نظر را 3 مرتبه و یا بیشتر اخذ کرده

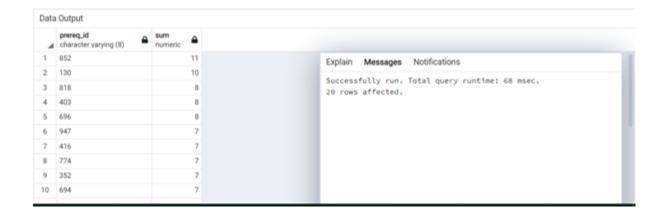
SELECT Name,Title
FROM Takes,Student,Course
WHERE Takes.Id=Student.Id AND Takes.Course_Id=Course.Course_Id
GROUP BY Student.Id,Course.Course Id

HAVING Count(*)>2



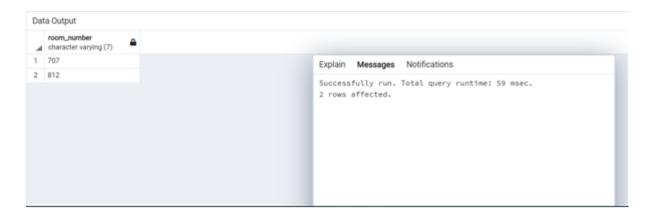
d. شناسه درس و جمع تعداد واحدهایی که این درس به طور مستقیم پیش نیاز درس های دیگر است را به دست بیاورید که درس مورد نظر پیش نیاز بیش از 4 واحد درسی دیگر هستند. (به ترتیب نزولی بر اساس جمع تعداد واحد هایی که درس پیش نیاز درس های دیگر است)

```
SELECT prereq_id,SUM(credits)
FROM prereq , course
WHERE prereq.course_id=course.course_id
GROUP BY prereq_id
HAVING SUM(credits)>4
ORDER BY sum DESC
```



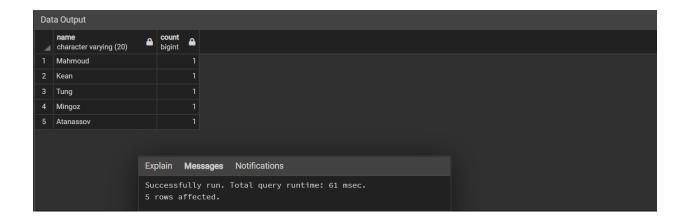
e. شماره اتاق هایی را بدست بیاورید که در ترم (Spring سال 2008)در طی هفته در مجموع 2 ساعت یا بیشتر اشغال هستند.(فقط ساعت مد نظر است و دقیقه مدنظر نیست)

```
SELECT room_number
FROM section , time_slot
where section.time_slot_id=time_slot.time_slot_id AND year=2008 AND
semester='Spring'
GROUP BY room_number
HAVING SUM(end_hr-start_hr)>=2
```



f. نام استاد به همراه تعداد دروس ارائه شده در سال 2003 را برای اساتیدی که در سال 2003 از میانگین تعداد دروس ارائه شده توسط هر استاد در آن سال تعداد درس های کمتری ارائه داده باشند به دست آورید.

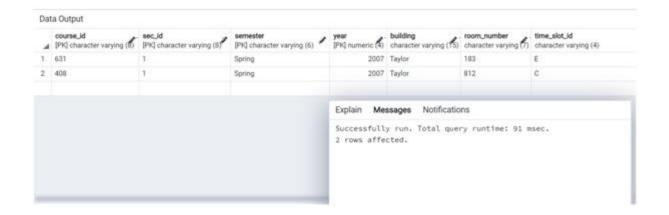
```
SELECT instructor.name,COUNT(*)
FROM teaches,instructor
WHERE teaches.id=instructor.id AND year=2003
GROUP BY instructor.id
HAVING COUNT(*)<(SELECT AVG(InsTeachCount.TeachCount)
FROM (SELECT instructor.id,COUNT(*) AS TeachCount
FROM teaches,instructor
WHERE teaches.id=instructor.id AND year=2003
GROUP BY instructor.id) AS InsTeachCount)
```



g. كلاس هايي كه در سال 2007 در ساختمان Taylor، حداقل يكي از ساعات شروع ارائه آن بين ساعات 8 صبح الي 12 برگزار ميشوند را به دست آوربد.

```
SELECT * FROM Section
WHERE year=2007 AND building='Taylor' AND
(SELECT COUNT(*)
```

```
FROM time_slot
WHERE section.Time_Slot_id=time_slot.Time_slot_id AND
start_hr BETWEEN 8 AND 12)>=1
```



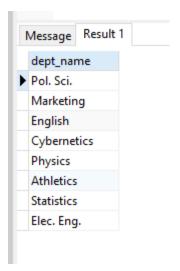
h. فرض کنید واحد تعدادی از دروس تغییر کرده است .تعداد واحد های گذرانده دانشجویان را مجدد محاسبه و به همراه نام آن ها نمایش دهید.(واحد های گذرانده را تنها با گرید های دسته A و گرید های دسته B در نظر بگرید)

```
SELECT s.name,SUM(credits)
    FROM takes, course , student as s
    WHERE course.course_id=takes.course_id AND    s.id=takes.id
    AND (grade like 'A%' or grade like 'B%')
    GROUP BY s.id
```

Message Result 1		
name	sum	
Scibili		21
Towsey		34
Hameed		25
Kocsis		30
Frolova		40
Coppens		44
Raïev		53
Katehakis		16
Hayrapetyan		50
Richter		28
Lagendijk		35
Folkers		36

5. برای هر یک از دستورات زیر بنویسید که چه نتیجه ای بر می گردانند و WITH را به SUB QUERY و برعکس تبدیل کنید. a. تبدیل از WITH به SUB QUERY و توضیح خروجی دستور

```
with dept total (dept name, value) as
(select dept name, sum (salary)
from instructor
group by dept name),
dept total avg(value) as
(select avg(value)
from dept total)
select dept name
from dept total, dept total avg
where dept total.value >= dept total avg.value;
SELECT dept name FROM
(SELECT dept name, sum(salary) FROM instructor GROUP BY dept name) as
dept total(dept name, value) ,
(SELECT avg(value) FROM (select dept name, sum(salary) FROM instructor GROUP
BY dept name) as dept total(dept name, value)) as dept total avg(value)
WHERE dept_total.value >= dept_total_avg.value;
کوئری مربوطه در واقع جمع حقوقی که به اساتید تعلق میگیرد به تفکیک دپارتمان محاسبه کرده،سپس میانگین این حقوق ها را
  محاسبه کرده، در نهایت کوئری کلی دپارتمان هایی را می آورد که جمع حقوقی پرداختی آن دپارتمان ها به اساتیدشّان از میانگین
                                                     حقوق پرداختی دیارتمان ها به استاد ها بیشتر باشد.
```

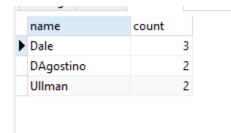


b. تبدیل از SUB QUERY به WITH و توضیح خروجی دستور

SELECT instructor.name,COUNT(*)
FROM teaches,instructor

```
WHERE teaches.id=instructor.id AND year=2003
GROUP BY instructor.id
HAVING COUNT(*)>(SELECT AVG(InsTeachCount.TeachCount)
                 FROM (SELECT instructor.id, COUNT(*) AS TeachCount
                     FROM teaches, instructor
                     WHERE teaches.id=instructor.id AND year=2003
                       GROUP BY instructor.id) AS InsTeachCount)
WITH InsTeachAvg(TCAvg) AS( SELECT AVG(InsTeachCount.TeachCount)
            FROM (SELECT instructor.id, COUNT(*) AS TeachCount
                  FROM teaches, instructor
                 WHERE teaches.id=instructor.id AND year=2003
                      GROUP BY instructor.id, year) AS InsTeachCount )
SELECT instructor.name,COUNT(*)
FROM teaches, instructor
WHERE teaches.id=instructor.id AND year=2003
GROUP BY instructor.id, year
HAVING COUNT(*)>(SELECT TCAvg FROM InsTeachAvg)
```

نام استاد به همراه تعداد دروس ارائه شده آن در سال 2003 را برای اساتیدی که در سال 2003 از میانگین تعداد دروس ارائه شده توسط هر استاد در آن سال تعداد درس های بیشتری ارائه داده باشند را لیست می کند.



"score" int4,
"is rank" int2

```
Oniversity: Large Version) موارد زیر را به ترتیب با کمک دستورات (University: Large Version) انجام دهید .a دولی با اطلاعات .a uni_data(stu_id, stu_name,stu_dept_name, year,semester,course_name, score, is_rank)

بسازید. نوع فیلد ها با توجه فیلد های پایگاه انتخاب کنید.

CREATE TABLE "public"."uni_data" (
"stu_id" varchar(5),
"stu_name" varchar(20),
"stu_dept_name" varchar(30),
"year" numeric(4),
"semester" varchar(6),
"course name" varchar(50),
```

```
);
Micaauge
 CREATE TABLE "public". "uni_data" (
   "stu_id" varchar(5) NOT NULL,
   "stu_name" varchar(20),
   "stu_dept_name" varchar(255),
   "year" numeric(4),
   "semester" varchar(6),
   "course_name" varchar(50),
   "score" int4,
   "is_rank" int2
 )
 > OK
 > Time: 0.039s
        d. با توجه به جدولی که در قسمت قبل ساختید دستوری بنویسید که داده های دانشجو ها و درس های
                   که برداشته اند را در جدول وارد کند و score دانشجو را مطابق grade زیر ثبت کند.
             A + = 100, A = 95, A - = 90
              B+ = 85, B = 80, B- = 75
             C+ = 70, C = 65, C- = 60
              Else: 0
               و مقدار ستون is rank اگر امتیاز شخصی بزرگتر از 70 بود یک باشد و در غیر اینصورت 0
INSERT INTO uni data
SELECT s."id", s."name", s.dept name, T."year", T. semester, C. title,
(CASE
       WHEN T.grade = 'A+' THEN 100
      WHEN T.grade = 'A ' THEN 95
      WHEN T.grade = 'A-' THEN 90
      WHEN T.grade = 'B+' THEN 85
      WHEN T.grade = 'B ' THEN 80
      WHEN T.grade = 'B-' THEN 75
      WHEN T.grade = 'C+' THEN 70
      WHEN T.grade = 'C ' THEN 65
      WHEN T.grade = "C-" THEN 60 ELSE 0 END ),
     ( CASE WHEN T.grade IN ( 'A+', 'A ', 'A-', 'B+', 'B', 'B-' ) THEN 1 ELSE
0 END )
    FROM takes AS T JOIN student AS s ON T."id" = s."id"
    JOIN course AS C ON C.course id = T.course id
```

```
INSERT INTO uni_data
 SELECT s."id", s."name",s.dept_name,T."year",T.semester,C.title,
       WHEN T.grade = 'A+' THEN 100
       WHEN T.grade = 'A ' THEN 95
       WHEN T.grade = 'A-' THEN 90
       WHEN T.grade = 'B+' THEN 85
       WHEN T.grade = 'B ' THEN 80
       WHEN T.grade = 'B-' THEN 75
       WHEN T.grade = 'C+' THEN 70
       WHEN T.grade = 'C ' THEN 65
     WHEN T.grade = 'C-' THEN 60 ELSE 0 END ),
( CASE WHEN T.grade IN ( 'A+', 'A ', 'A-', 'B+', 'B ', 'B-' ) THEN 1 ELSE 0 END )
     FROM takes AS T JOIN student AS s ON T. "id" = s. "id"
     JOIN course AS C ON C.course_id = T.course_id
 > Affected rows: 30000
 > Time: 0.868s
          c. دستوری بنویسید که امتیاز درس های دانشجوبان دانشکده فیزیک را در جدولی که ساخته اید، اگر
                             کوچکتر از 75 بود 10 امتیاز و در غیر اینصورت 15 امتیاز افزایش دهد.
UPDATE uni data
SET score = ( CASE WHEN score < 75 THEN score + 10 ELSE score + 15 END )
WHERE
  uni data.stu dept name = 'Physics'
               Message
                UPDATE uni_data
                SET score = ( CASE WHEN score < 75 THEN score + 10 ELSE score + 15 END )
                  uni_data.stu_dept_name = 'Physics'
                > Affected rows: 1416
                > Time: 0.157s
       d. دستوری بنویسید که رکورد هایی که اسم دانشجو با T شروع می شوند و امتیازشان از میانگین امتیاز ها
                                                     در آن دانشکده کمتر است حذف شوند.
DELETE FROM uni data AS ud
WHERE
  stu name LIKE 'T%'
  AND score < ( SELECT AVG ( score ) FROM uni data AS sub ud WHERE
sub_ud.stu_dept_name = ud.stu_dept_name)
```

```
DELETE FROM uni_data AS ud
WHERE
stu_name LIKE 'T%'
AND score < ( SELECT AVG ( score ) FROM uni_data AS sub_ud WHERE sub_ud.stu_dept_name = ud.stu_dept_name)
> Affected rows: 557
> Time: 6.948s
```