

سوال یک

سه حرف مخفف شده ماه های میلادی: `char(n)`
قیمت دلاری محصولاتی که همگی دو رقم اعشار دارند: `numeric(p,2)`
نام و نام خانوادگی کاربر: `varchar(30)`
کد ملی کاربر: `char(10)`
ذخیره ی قیمت لحظه ای ارز های دیجیتال: `real, double precision, float`
تعداد بازدید های یک ویدیو: `int`

سوال دو

`char(10)` به این معنا است که دقیقاً باید ده کارکتر داشته باشیم ولی `varchar(10)` به این معنا است که حداکثر ده کارکتر داریم به عنوان مثال زمانی که میخواهیم برای کد ملی نوع داده را مشخص کنیم از آنجایی که طول کد ملی ثابت است از `char(n)` استفاده میکنیم و زمانی که میخواهیم برای نام و نام خانوادگی نوع داده را مشخص کنیم چون اسم ها طول های متفاوتی دارند از `varchar(n)` استفاده میکنیم.

سوال سه

Where `substring(first_name,1,2) = 'me'` and `right (last_name,3) = 'avi'`

Select `concat(first_name,last_name)`

سوال چهار

a

	id [PK] character varying (5)	name character varying (20)
1	25525	Moreira
2	10204	Mediratta
3	69225	Mejia
4	83462	Mehra

b

	title character varying (50)
1	Tort Law

c

	name character varying (20)	title character varying (50)
1	Preuss	Embedded Systems
2	Date	Embedded Systems
3	Aiken	Embedded Systems
4	Noga	Embedded Systems
5	Karande	Embedded Systems
6	Prabhakaran	Embedded Systems
7	Hakkinen	Embedded Systems
8	Drig	Embedded Systems
9	Harshman	Embedded Systems
10	Won	Embedded Systems

d

	prereq_id character varying (8)	sum numeric
1	852	11
2	130	10
3	818	8
4	403	8
5	696	8
6	947	7

e

	room_number character varying (7) 🔒
1	707
2	812

f

	t_name character varying (20) 🔒	t_count bigint 🔒
1	Mahmoud	1
2	Kean	1
3	Tung	1
4	Mingoz	1
5	Atanassov	1

g

	course_id [PK] character varying (8) 🔒	sec_id [PK] character varying (8) 🔒	semester [PK] character varying (6) 🔒	year [PK] numeric (4) 🔒	building character varying (15) 🔒	room_number character varying (7) 🔒	time_slot_id character varying (4) 🔒
1	408	1	Spring	2007	Taylor	812	C
2	631	1	Spring	2007	Taylor	183	E

h

	name character varying (20) 🔒	sum numeric 🔒
1	Scibili	21
2	Towsey	34
3	Hameed	25
4	Kocsis	30
5	Frolova	40
6	Coppens	44
7	Raïev	53
8	Katehakis	16
9	Hayrapetyan	50
10	Richter	28

سوال پنج

a

dept_total مجموع حقوق را بر اساس دپارتمان حساب می کند در واقع به ما می گوید مجموع حقوق اساتید در هر دپارتمان چند است و پس یک زوج است که شامل نام دپارتمان و مجموع حقوق اساتید است و dept_total_avg میانگین مجموع حقوق را بر اساس دپارتمان ها محاسبه میکند و در نهایت میانگین مجموع حقوق همه ی دپارتمان ها را نشان میدهد در نهایت دپارتمان هایی سلکت شده اند که مجموع حقوق دپارتمان آن ها از میانگین مجموع حقوق همه ی دپارتمان ها بیشتر یا مساوی بوده است.

```
select dept_name
from(
select dept_name, sum(salary) as s
from instructor
group by dept_name
) as t1
where t1.s >
(
select avg (s)
from (
select dept_name, sum(salary) as s
from instructor
group by dept_name
) as t2
)
```

	dept_name character varying (20)
1	Pol. Sci.
2	Marketing
3	English
4	Cybernetics
5	Physics
6	Athletics
7	...

b

اسم اساتید را به همراه تعداد درسی که تدریس کرده اند نشان میدهد به شرطی که تعداد درسی که در سال ۲۰۰۳ تدریس کرده اند از تعداد درسی که به صورت میانگین توسط اساتید در سال ۲۰۰۳ تدریس شده است بیشتر باشد.

```
with avg_ins_count(value) as
(
SELECT AVG(t.TeachCount)
FROM (SELECT instructor.id,COUNT(*) AS TeachCount
FROM teaches,instructor
WHERE teaches.id=instructor.id AND year=2003
GROUP BY instructor.id ) as t
),
ins_count (name,value) as
(
SELECT instructor.name,COUNT(*),COUNT(*)
FROM teaches,instructor
WHERE teaches.id=instructor.id AND year=2003
GROUP BY instructor.id
)
SELECT name,sum(ins_count.value)
from ins_count,avg_ins_count
where ins_count.value > avg_ins_count.value
group by name
```

	name character varying (20)	sum numeric
1	D'Agostino	2
2	Dale	3
3	Ullman	2

سوال شش

a

```
-- q6
-- a
create table uni_data
(
    stu_id          varchar(5),
    stu_name        varchar(20) not null,
    stu_dept_name    varchar(20),
    year            numeric(4,0),
    semester        varchar(6),
    course_name      varchar(50),
    score           float,
    is_rank         int
);
```

stu_id	stu_name	stu_dept_name	year	semester	course_name	score	is_rank
character varying (5)	character varying (20)	character varying (20)	numeric (4)	character varying (6)	character varying (50)	double precision	integer

b

	stu_id	stu_name	stu_dept_name	year	semester	course_name	score	is_rank
	character varying (5)	character varying (20)	character varying (20)	numeric (4)	character varying (6)	character varying (50)	double precision	integer
1	97101	Marek	Psychology	2005	Spring	Shakespeare	85	1
2	26695	Hac	Finance	2003	Spring	Mechanics	90	1
3	6990	Hoshi	Biology	2007	Fall	Journalism	70	0
4	74530	Ranno	Physics	2006	Fall	Tort Law	100	1
5	23270	Bouras	Biology	2004	Fall	Graphics	65	0
6	16297	Sacchi	Marketing	2005	Spring	Calculus	70	0
7	52187	Fritsch	Math	2008	Spring	Surfing	65	0
8	86001	Meise	English	2004	Fall	Graphics	100	1
9	34322	Baba	Biology	2007	Fall	Japanese	85	1
10	73542	Dooley	Geology	2001	Spring	Arabic	90	1
11	65753	Plantl	Geology	2002	Fall	Optics	100	1

```
-- b
INSERT INTO uni_data (stu_id,stu_name,stu_dept_name,year,semester,course_name,score)
SELECT student.id, student.name, student.dept_name,takes.year,takes.semester,course.title,
CASE
    WHEN takes.grade = 'A+' THEN 100
    WHEN takes.grade = 'A ' THEN 95
    WHEN takes.grade = 'A-' THEN 90
    WHEN takes.grade = 'B+' THEN 85
    WHEN takes.grade = 'B ' THEN 80
    WHEN takes.grade = 'B-' THEN 75
    WHEN takes.grade = 'C+' THEN 70
    WHEN takes.grade = 'C ' THEN 65
    WHEN takes.grade = 'C-' THEN 60
    ELSE 0
END AS score
FROM student,takes,course
where student.id=takes.id
and course.course_id = takes.course_id;

update uni_data
set is_rank = case
    when score > 70
    then 1
    else 0
end;
```

c

688	75123	Chowdhury	Physics	2003	Spring	Finite Element Analysis	105	1
689	12711	Malagoli	Physics	2004	Fall	Video Gaming	110	1
690	41890	Srivastava	Physics	2006	Fall	The IBM 360 Architecture	110	1
691	21552	Bradshaw	Physics	2008	Spring	World History	110	1
692	53451	Hasan	Physics	2009	Fall	Rock and Roll	95	1
693	18338	Kangs	Physics	2006	Spring	Video Gaming	95	1
694	29260	Verhoeven	Physics	2008	Spring	Race Car Driving	95	1
695	7498	Chang	Physics	2006	Spring	Operating Systems	95	1
696	77000	Cherchi	Physics	2001	Fall	Aquatic Chemistry	90	1
697	41675	Wheeler	Physics	2006	Fall	Multimedia Design	100	1

```
-- c
select * from uni_data;
update uni_data
set score= score+10
where score > 75 and stu_dept_name='Physics';

update uni_data
set score= score+15
where score <= 75 and stu_dept_name='Physics';
```

d

```
-- d
delete from uni_data;
where stu_name like 'T%' and score < (select avg (score) from uni_data);

--
```