Subject: ۳ نمره تبدیل ER به مدل رابطه‌ای — Final در ANSI — ۳ نمره ER — ۱ نمره ۳ تمرین سه سطحی و ۲ نمره ویژگی‌های مدل‌ها

Year Month Date — ۲ نمره از تعاریف جلسه پیش و این جلسه تا ۳۶۲ — ۶ نمره کدنویسی SQL و مفهوم‌های این جلسه

Duplicate key : اگر در جدول C سعی کنیم update کنیم جای C_1 ، C_2

بذار میثیم عمیقه، Duplicate key خطای اصلی نباید تکراری بشه ۲ تا نباید مشابه هم باشه

در کدنویسی SQL بخش CREATE TABLE نباید اسم جدول تکراری بشه.

Data Type نوع داده

عددی

integer ، int : عدد صحیح

float ، real : اعشاری

numeric (x, y)
decimal (x, y)

x: حداکثر طول عدد
y: حداقل طول اعشار

رشته: رشته در SQL به صورت char(n) است.

nchar(n) ← رشته کاراکتر با طول معین

nvarchar(n) * ۱- رشته کاراکتر ۲- طول متغیر — این بیشتر استفاده میشه

varchar(n) ← طول رشته‌ها حساب می‌کنه.

numeric (3,1) — numeric (4,2)

مثلاً این غلط ۲۱۴/۰ X

1
12
123
~~1231~~

```
Create table student
(
    S#  char(10),   Null & Not Null   ، Unique اینجا می‌تونیم بنویسیم
    PRIMARY key   می‌تونیم تعریف کنیم — نوشتن اش مهمه
    S(S#)  مثلاً foreign key نوشتن اینکه
```

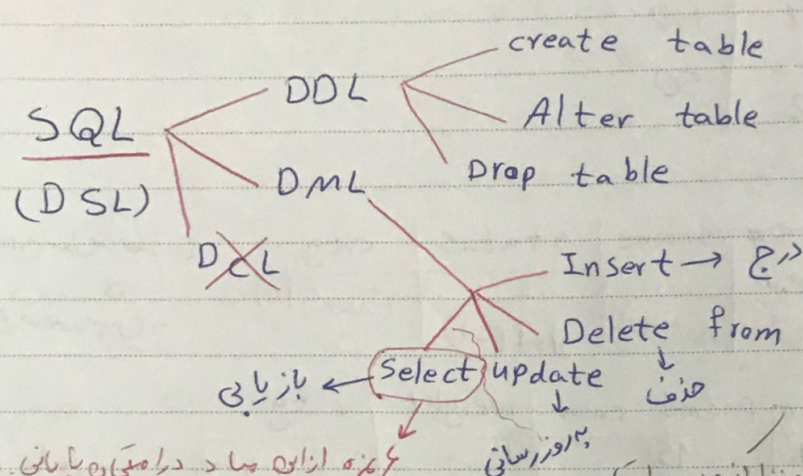
Vazik

Scanned by CamScanner

Subject:
Year Month Date ( )

چهارشنبه امتحان میان ترم معماری سه سطحی و ویژگی نسل ها زیاد میاد.
ER یک حلقه میاد.

از این جلسه ۳-۴ نمره در امتحان پایانی می آید، ۳ نمره ER میره جدول ما میکشیم ۲ نمره تعریفی.



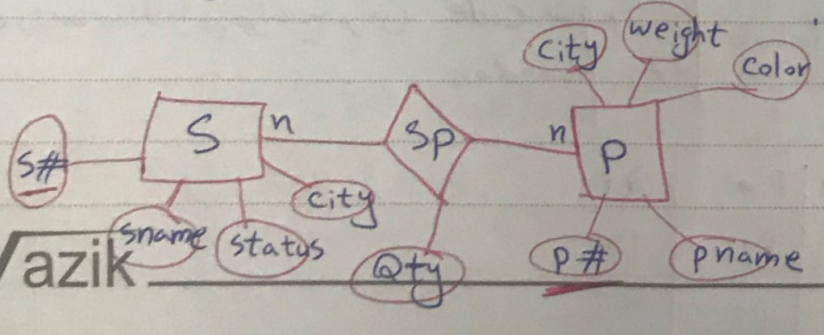
۶ نمره از این میاد در امتحان پایانی.

باید اطلاعاتش را حفظ کنیم سره امتحان

بانک اطلاعاتی قطعه ـ عرضه کننده:

در یک محیط عملیاتی هر عرضه کننده (S) دارای کلی منحصر بفرد (S#)، نام (Sname)، کد وضعیت (status)، شهر (city) است و اقدام به عرضهٔ (SP) تعداد معینی (Qty) از انواع قطعه می کند، به طوری که هر قطعه دارای کلی منحصر بفرد (P#)، نام قطعه (Pname)، رنگ (color)، وزن قطعه (weight) می باشد همچنین شهری که در آن قطعه انبار (city) می شود نیز ثبت می شود.

(P) Part



Vazik

Scanned by CamScanner

Subject:

Year ..... Month ..... Date ..... ( )

دراسکن ER دیگه نمی‌آد.

S#	P#	SP#
S_1	P_1	100
S_1	P_2	50

۲ نوع داریم

طراحی مدل رابطه‌ای صفحه‌ی ER.

S

S#	sname	status	city
کلید اصلی char(5)	char(10) not null	int def(-1)	char(10)

P

P#	pname	color	weight	city
کلید اصلی char(6)	char(15) not null	char(10) def(???)	int	char(10)

کلیدها نوعشون باید یکسان باشه.

سوال اول امتحان پایان ترم ۳ نمره

Sp

S#	P#	Qty
کلید خارجی مرجع S	کلید خارجی مرجع P	int
char(5)	char(6)	

برای تمام موجودیت‌ها یک جدول رسم می‌کنیم

چون رابطه n-n بین دو جدول جداگانه برای رابطه رسم می‌کنیم.

کلید اصلی

Delete cascade
update restrict

Delete cascade
update

DataBase : Sp

رشته

طول رشته مشخص است: char(n)

طول رشته مشخص نیست: Varchar(n)

nchar(n)

nvarchar(n)

unicode

عدد

اعشاری: float / real

صحیح: int

S#	sname	status	city
S_1	sn1	17	London
S_2	sn2	19	Rome

P#	pname	color	weight	city
P1	Pn1	Blue	19	Rome
P2	Pn2	Red	12	Paris
P3	Pn3	Yellow	17	London

جدول‌های بالا را می‌خواهیم برای نوع Code

S#	P#	Qty
S_1	P_1	100
S_1	P_2	300
S_1	P_3	400
S_2	P_1	100
S_2	P_3	200

Vazik

Scanned by CamScanner

Subject:
Year Month Date ( )

update برای این مرحله بهتره.

① Restrict (محدودیت) → اگر در جدول S، s1 را حذف کنیم در جدول sp تمام s1 ها بدون مرجع می‌مونن برای این مشکل از محدودیت استفاده می‌کنیم.

② Cascade (آبشاری) یعنی اول می‌آیم توی sp همه s1 ها را حذف می‌کنیم بعد در جدول S، s1 ها را حذف می‌کنیم ولی در cascade برعکس.

در SQL جدول‌ها:

```
Create table S (
S#   char(5),
sname   char(10)  Not null,
status   int  default (-1),
city   char(10),
primary key (S#));
```

```
create table P(
P# → pno = pnumber   char(6),
pname   char(15)  Not Null,
color   char(10)  default('???'),
weight   int,
city   char(10),
primary key (P#));
```

کلید اصلی

ادامه جدول

```
create   table   sp(
S#   char(5),
P#   char(6),
Qty   int,
primary key (S#, P#),
foreign key (S#)
references   S(S#)
ON   Delete   cascade
ON   update   restrict,
foreign key (P#)   references P(P#)
ON Delete cascade
ON update cascade
);
```

کلید خارجی

ادامه کد

اینجا باید ویرگول بزاریم

Vazik

Scanned by CamScanner

Subject:

Year Month Date ( )

کلید خارجی (S#, city) — که کلید خارجی ترکیبی بود ترتیب مهم.

references S(S#, city)

هر تغییری که می‌خواهیم توی جدول انجام دهیم از ALTER TABLE استفاده می‌کنیم.

(حذف، اضافه و ...)

S#	P#	Qty
S_1	P_4	100
S_1	P_4	300

Duplicate Key ← error — غیره اینو اضافه کرد

جلسه V: هر کد SQL که می‌خواهد نوشته شود تمام قواعد جامعیت آن رعایت شده

S

S#	sname	status	city
S_1	sn_1	17	London
S_2	sn_2	18	Rom
S_3	sn_3	12	Paris
S_5	sn_5	-1	London

مثلاً درج می‌خواهیم انجام دهیم اول بینیم NULL نباشه کلید اصلی.

P

P#	pname	Color	weight	city
P_1	Pn_1	Red	17	London
P_2	Pn_2	Blue	19	Rom
P_3	Pn_3	Red	23	Paris

مشخصات جداول را حتماً باید حفظ باشیم یعنی اینا رو

SP

S#	P#	Qty
S_1	P_1	100
S_1	P_2	200
S_2	P_1	300
S_2	P_2	400
S_4	P_3	100

Vazir

Scanned by CamScanner

Subject:
Year Month Date

۲ تا ۳ نمره ③ قواعد جامعیت و انجام عملیات ۳ نمره ④ تبدیل ERD به جداول رابطه‌ای } میانترم
۲ نمره ⑤ کدنویسی SQL ۷ تا ۸ نمره ② تعاریف مدل رابطه‌ای ۲ نمره }

عملیات : (درج) Insert

Insert into S (city, Sname, S#)
Values('London', 'Sn5', 'S5')

درج سه تایی در جدول S خودمان به ردیف S5، Sn5 و London می‌دهیم ولی

Insert into sp (Qty, P#)
Values('400', 'P2')

status مقدار پیش‌فرض default تعریف کرده‌ایم -1

عملیات انجام نمی‌شود ← جامعیت نهاد (عضو S#)

Insert into sp (Qty, P#, S#)
Values(300, 'P2', 'S3')

درج انجام نمی‌شود چون قبلاً درج شده در جدول sp پس Duplicate Key روی می‌دهد.

تکراری نباید بدهیم.

Insert into P(P#, color, weight, city)
Values('P4', 'Blue', 25, 'Rome')

درج انجام نمی‌شود چون Pname عضو Not Null

Insert into sp (S#, P#, Qty)
Values('S5', 'P5', 100)

نداریم

انجام نمی‌شود چون (P5)

Refrential Integrity (جامعیت ارجاعی)

Vazik

Subject:

Year ..... Month ..... Date ..... ( )

جدول S ← عرضه کننده / جدول P ← قطعه

SP ← عرضه

update S

Set status = status + 2

where city = 'Rome'

عرضه کننده ← S

عرضه کننده شهر Rome را ۲ واحد اضافه کن.

update اسم جدول

set تغییرات

where شرایط

مثال) به که وضعیت عرضه کننده‌هایی که با قطعه P2 در یک شهر هستند ۲ واحد اضافه شود.

update S

set status = status + 2

where city = (select city

from P

where p# = 'P2')

در جدول P شهر P2 ← Rome است می‌بریم در جدول S ردیف‌هایی که شهرشون Rome است با P2 هم شهره.

ولی اگه شهر Rome و ۲ مقدار هم شهر بود با شرط نامساوی دیگه مساوی نمی‌نوشتیم بذاریم.

نامساوی در یک شهر نباشه

update P

Set P# = 'P4'

where color = 'Red'

(۱) لا انجام میشه ولی تأثیری نداره

(۲) $(P_1, P_3) \rightarrow P_4$

P_1 و P_3 ی خواد شه P_4

(۳) چون در جدول P دوتا رنگ Red داریم پس انجام نمی‌شود Duplicate key

اگه تکرار نباشه پس کلید ارجاعی داریم ولی اگه یک تکرار داشته باشیم

پیغام انجام نمی‌شود

Duplicate key

Vazik

Scanned by CamScanner

Subject:
Year Month Date ( )

update p
set p# = 'p4'
where color = 'Blue'

اگر کلید خارجی p# در SP : Cascade : انجام می‌شود و کلیه‌ی p2 های SP هم p4 می‌شوند.

ولی اگر لفت Restrict : انجام نمی‌شود.

اگر چیزی لفت بدهد یعنی Restrict است.

update SP
set p# = 'p5'
where p# = 'p3'

جامعیت ارجاعی، در جدول P سطر P_5 موجود نیست.

update SP
set p# = 'p2'
where p# = 'p3'

انجام نمی‌شود چون در SP ستون P# اگر جای P_3 بیاید P_2 داریم چون ردیف بالایی دقیقاً تکراری میشه duplicate key ← تکراری

Select : خروجی هر Select خودش یک جدول.

از جدول ← Select
from

شرط ← where
Restrict

دسته‌بندی ← group by
Having

order By

مثال) نام و شهری که status آنها بزرگتر از 15 باشد:

Select Sname, City
عرضه کنندگان
from S
where status > 15

Vazik

Scanned by CamScanner

Subject:
Year Month Date ( )

select color
رنگ قطعه‌ی P_1

from P → قطعه

where P# = 'P1'

نام قطعات قرمز رنگی که در شهر Rome انتخاب شده‌اند.

select pname
from P
where color = 'Red'
AND city = 'Rome'

چون دو تا شرط داریم.

نام قطعات قرمز رنگ شهر Rome که وزن آنها ۲ رقمی است:

select pname
from P
where color = 'Red'
AND city = 'Rome'
AND weight >= 10
AND weight <= 99

نام قطعات قرمز رنگی که در شهر rm قرار دارند یا وزن آنها دو رقمی است:

select pname
from P
where color = 'Red'
AND (city = 'Rome'
OR (weight >= 10
AND weight <= 99))

پرانتز ها باید گذاشته شود.

Vazik

Scanned by CamScanner

Subject:
Year Month Date ( )

نام عرضه کننده گانی که با P1 در یک شهر قرار دارند.

```
select  sname
from    S
where   city = ( select  city
                 from    p
                 where   p# = 'P1' )
```

نام سنگین وزن ترین قطعه:

```
select  pname
from    p
where   weight = ( select  max (weight)
                   from p )
```

max ← بیشترین مقدار
min ← کمترین مقدار
avg → میانگین
sum → جمع

نام سنگین وزن ترین قطعه قرمز رنگ:

```
select  pname
from    p
where   color = 'Red'
AND     weight = ( select  max (weight)
                   from    p
                   where   color = 'Red' )
```

قطعه قرمز رنگ

سنگین ترین وزن بین قطعات قرمز

Vazik

Scanned by CamScanner

Subject:

Year ..... Month ..... Date ..... ( )

S

S #	sname	status	City
S_1	Sn_1	17	Rome
S_2	Sn_2	19	paris
S_3	Sn_3	12	London
S_4	Sn_4	19	paris

نام عرضه کننده ها

(مثال

P

P #	Pname	color	weight	city
P_1	Pn_1	Red	17	London
P_2	Pn_2	Blue	13	paris
P_3	Pn_3	Red	11	London

SP

S#	P #	Qty
S_1	P_1	100
S_1	P_2	200
S_2	P_1	400
S_2	P_3	100
S_3	P_1	200

(مثال) عرضه کننده‌ای که قطعه P_1 را عرضه کرده‌اند.

[redacted] است یعنی طلبی در جدول sp.

```
select   S #
from     SP
where    P # = 'P1'
```

Vazik

Scanned by CamScanner

نام عرضه کننده‌گانی که قطعه P_1 را عرضه کرده‌اند.

Sname در جدول S هست.

```
Select *
from S inner join Sp on
```

S.S# = Sp.S#

S.S#	Sname	status	city	Sp.S#	P#	Qty
S_1	Sn_1	17	Rome	S_1	P_1	۱۰۰
S_1	Sn_1	17	Rome	S_1	P_2	۲۰۰
S_2	Sn_2	19	Paris	S_2	P_1	۴۰۰
S_2	Sn_2	19	Paris	S_2	P_2	۱۰۰
S_3	Sn_3	12	London	S_3	P_1	۲۰۰

join کنیم

```
select sname
from S inner join Sp on
```

S.S# = Sp.S#

```
where P# = 'P1'
```

نام قطعاتی که توسط S_1 عرضه شده‌اند.

از جدول SP

و فهمیم که S_1 چی قطعاتی تهیه کرده بعد رجوع و لینم به جدول P

Vazik

Scanned by CamScanner

Subject:

Year Month Date ( )

```
select   color
from     p inner join sp on p.p# = sp.p#
where    s# = 's1'
```

۲ تا جدول را join کردیم.

---

۴ قطعاتی که توسط عرضه کنندگان لندنی عرضه شده اند در کدام شهر انبار شده اند.

جدول S ← ۳ تا جدول را باید join کنیم.

s# ستون S در جدول S می‌بریم فقط s_3 هست که London هست پس باید برویم جدول sp

$p_1 \leftarrow s_3$ را عرضه می‌کند می‌بریم جدول P

```
select   p.city          ← درخروجی ظاهر شود
from     s inner join sp on s.s# = sp.s#
         inner join p on sp.p# = p.p#
where    s.city = 'London'
```

s.city ← city را خالی بنویسیم error میده باید

بنویسیم.

قطعاتی که توسط s_1 یا s_2 عرضه شده اند.

یعنی طبق اصلی ← p#

```
select   p#
from     sp
where    s# = 's1'  OR  s# = 's2'
```

Vazik

Scanned by CamScanner

Subject:

Year: Month: Date:

نام قطعاتی که توسط S_1 یا S_2 عرضه شده اند.

```
select Pname
from P inner join SP on P.P# = SP.P#
        where S# = 'S1' OR S# = 'S2'
```

قطعاتی که توسط S_1 و S_2 عرضه شده اند.

```
select P#
from SP
where S# = 'S1' AND S# = 'S2'
```

اشتراک — غلط — اجتماع

چون در یک ردیف نمی‌توان AND را اجرا کرد. ها P# هم باید یکی باشد

در جدول S

SP.S#	P.P#	Qty	SP.S#	SP.P#	Qty
X حذف S_1	P_1	۱۰۰	S_1	P_1	۱۰۰
X S_1	P_1	۱۰۰	S_1	P_2	۲۰۰
✓ S_1	P_1	۱۰۰	S_2	P_1	۴۰۰
X S_1	P_1	۱۰۰	S_2	P_3	۱۰۰
X S_1	P_1	۱۰۰	S_3	P_1	۲۰۰

جدول → جدول خود

```
select P#
from SP inner join SPX
    ON SP.P# = SPX.P#
where SP.S# = 'S1'
AND SPX.S# = 'S2'
```

SOBHAN

Scanned by CamScanner

Subject: Year: Month: Date:

نام قطعاتی که توسط S_1 و S_2 عرضه شده اند.

```
select pname
from P inner join SP on P.P#=SP.P#
     inner join SP SPX
     ON SP.P#=SPX.P#
where SP.S#='S1'
AND SPX.S#='S2'
```

نام قطعاتی که عرضه کننده ای با رنگ قرمز دارند.

۳ جدول را join میکنیم.

```
Select pname
from P inner join SP on P.P#=SP.P#
     inner join S on S.S#=SP.S#
where S.city='Rome' oR S.city='London'
```

$A = \{ \quad \}$

$a \in A$

$a \notin A$

in های تودرتو

مثال) قطعاتی که توسط S_1 یا S_2 عرضه شده اند

```
select P#
from SP
where S# in ('S1','S2')
```

SOBHAN

Scanned by CamScanner

Subject: Year: Month: Date:

مثال) نام محصولاتی که عرضه کننده‌ای از شهر رم یا لندن دارند.

```
select pname
from P inner join SP on P.P# = SP.P#
       inner join S on S.S# = SP.S#
       where S.city in('Rome','London')
```

مجموعه عرضه کننده‌ای که محصول P_1 را از SP خارج می‌کنیم

```
select S#
from S
where P# = 'P1'
```

مثال) نام عرضه کننده‌ای که عضو این مجموعه هستند.

```
Select Sname
from S
where S# in (select S#
             from SP
             where P# = 'P1')
```

مثال) نام عرضه کننده‌ای که محصول P_1 یا P_2 را عرضه می‌کنند.

```
select sname
from S
where S# in (select S#
             from SP
             where P# = 'P')
```

ادامه صفحه بعد

SOBHAN

Scanned by CamScanner

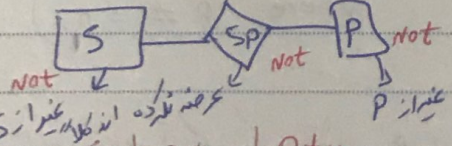
Subject: Year: Month: Date:

OR S# IN (select S#
from SP
where P# = 'P۲')

حالا اگه بخوایم
فقط کالای OR
و نداریم AND

select Pname
from P
where P# IN (select P#
from SP
where S# IN (select S#
from S
where city IN
('London', 'Rome'))

select Sname
from S
where S# in (select S#
from SP
where P# != 'P۱')



S#	P#	Qty
S۱	P۱	۱۰۰ X
S۱	P۲	۲۰۰ ✓
S۲	P۱	۴۰۰ X
S۲	P۲	۱۰۰ ✓
S۴	P۱	۲۰۰ X

SOBHAN

Scanned by CamScanner

```
select sname
from S
where S# Not in (select S#
                 from SP
                 where P# = 'P1')
```

"S#" هایی که در جدول SP برای P1 عرضه نکرده اند ← S_2 , S_4

پس در جدول S فقط S_1 و S_3 را نگه میداریم و بقیه را حذف میکنیم

```
select sname
from S
where S# Not in (select S#
                 from SP
                 where P# != 'P1')
      AND S# IN (select S# from SP)
```

برای حذف

فقط P_1 را عرضه کرده اند

① غیر از P_1 را عرضه نکرده اند ،

② حداقل یک قطعه را عرضه کرده باشند .

Scanned by CamScanner

عرضه کننده هایی که P_1 را عرضه کرده و P_2 را عرضه کرده اند.

```
select sname
from S
where S# Not in (select S#
                 from SP
                 where P# = 'P1')
AND S# IN (select S#
           from SP
           where P# = 'P2')
```

نام مشتریانی که توسط عرضه کننده‌ای عرضه شده اند دارای بالاترین درجه وضعیت در بین عرضه کننده‌ها می‌باشد.

```
Select pname
from P
where P# In (select P#
             from SP
             where S# IN(select S#
                         from S
                         where city=
                         'London'
                         AND status=
                         (select Max(status)
                          from S
                          where city='London')
                         from SP)
```


Scanned by CamScanner