

تست تھی بودن یک صفت :

□ کلمه Is Null برای تست تھی بودن و Is Not Null برای تھی نبودن استفاده میگردد.

مثال: Query بنویسید که نام و شماره دانشجویی دانشجویان را برده که در مقابل شهر آنها مقداری وارد نشده است.

```
select sn , sname from stud
where city Is NULL
```

Nested Query (Query تودرتو)

اگر در part3 یا part4 ، دستور select داشته باشیم به چنین Query هایی Query تو در تو کوئیم و به Query های ذکر شده در این تست ها sub Query کوئیم.

تست عضو مجموعه بودن :

برای این کار از کلمات IN (sub Query) و Not IN (sub Query) استفاده میگردد ، از آنجایی که In و Not in بروی یک Sub Query تست میشوند. پس در part3 و part4 از دستور select استفاده میگردد.

مثال : Query بنویسید که نام درسهایی را برده که نمره دانشجویان در آن درس بزرگتر از 15 باشد .

```
select cname from crs
where c# IN (select c# from sec
where score > 15)
```

در subQuery که در دستور بالا داریم از جدول sec شماره درسهایی بدست می آید که نمره کسب شده در آنها بزرگتر از 15 است، حال اگر شماره درسی از جدول crs عضو این مجموعه بدست آمده باشد یعنی آن درس ارائه شده و نمره کسب شده در آن بیشتر از 15 است.

□ دستور Query قبل را می توان با استفاده از پیوند طبیعی نیز نوشت، به این ترتیب که دروسی که ارائه شده اند از پیوند طبیعی crs , sec حاصل میگردد و می توان شرط $score > 15$ را بروی آن اعمال نمود .

□ اگر در part1 نام دو رابطه ذکر گردد، حاصلضرب دکارتی دو رابطه مناسبه میگردد ، برای پیوند طبیعی باید شرطی را مبنی بر مقایسه ستونهای همنام دو رابطه بنویسیم .

```
select cname from crs , sec
where (crsc#=secc#)
And (score > 15)
```

1. حاصلضرب دکارتی دو رابطه مناسبه میگردد.

2. با تست شرط مساوی بودن ستونهای همنام

3. پیوند طبیعی حاصل میگردد.

مثال. Query بنویسید که مشخصات دانشجویانی را برده که در شهرهای تهران، اصفهان، یزد و تبریز زندگی نمیکنند.

```
select * from stud
where city Not IN ("تهران", "اصفهان", "یزد", "تبریز")
```

در Query نوشته شده شهر ها از رابطه stud که عضو مجموعه ذکر شده نیستند، همراه با

مشخصات دانشجویان آنها داده می شود.

مقایسه مجموعه ها:

مثال. Query بنویسید که مشخصات دانشجویانی را برده که میانگین معدل آنها بزرگتر از معدل تمام دانشجویان دانشکده شماره 5 باشد.

```
select * from stud
where avge > all (select avge from stud where clgn = 5)
```

از subQuery رابطه ای بدست می آید که معدل تمام دانشجویان دانشکده شماره 5 در آن می باشد. حال می توان معدل هر یک از دانشجویان را از طریق $all >$ با تمام این مقادیر مقایسه کرد. و آنهایی را که معدل شان از تمام معدل های موجود در رابطه بدست آمده از subQuery بزرگتر است بدست آورد.



> som	بزرگتر از حداقل یکی
< som	کوچکتر از حداقل یکی
<> som	مخالف با حداقل یکی
= som	مساوی با حداقل یکی

> all	بزرگتر از همه
< all	کوچکتر از همه
<> all	مخالف همه
= all	مساوی همه

☐ all <> معادل است با NOT IN

☐ som <> معادل است با IN

مثال. Query بنویسید که مشخصات دانشجویانی را بدهد که معدل آنها بیشتر از معدل حداقل یکی از دانشجویان دانشکده شماره 5 باشد.

تست رابطه های خالی (Not Exist) و غیر خالی (Exist):

☐ از Exist و Not Exist برای تست غیر خالی و خالی بودن یک Query یا subQuery استفاده می گردد.

مثال. نام دانشکده هایی که درسی ارائه می دهند.

مثال. Query بنویسید که مشخصات درس هایی را بدهد که تا به حال ارائه شده اند.

```
select * from stud
where c# NOT IN (select C# from sec)
```

عملگر تقسیم :

برای پیاده سازی تقسیم در SQL مستقیماً عملگری وجود ندارد ولی می توان با تابع تجمعی count() پیاده سازی نمود.

مثال. Query بنویسید که شماره دانشجویانی را بدهد که همه درس ها را گرفته اند.

```
Select sn from Sec
Group by (Sn)
Having (count (Distinct C#))=(select Count(C#) from Crs)
```

در این Query رابطه Sec بر اساس شماره دانشجویی گروه بندی می گردد

و با استفاده از تابع Count() تعداد درس هایی را که در هر گروه و در واقع

برای هر دانشجو موجود است شمرده می شود، حال اگر مقدار این درس ها با مقدار تمام درس ها برابر باشد، یعنی دانشجو تمام درس ها را گرفته است. از Distinct به این خاطر استفاده شده که ممکن است دانشجویی درسی را افتاده باشد، و دوباره آن را بگیرد.

```
Select pname from Sec
Group by (pname)
Having (Count (Distinct C#))=(Select Count(C#) from Crs)
```

مثال. نام اساتیدی که همه درس ها را ارائه کرده اند.

مثال. مشخصات اساتیدی که همه درس ها را ارائه کرده اند.

```
Select * from prof
where pname IN (Select pname from Sec
Group by (pname)
Having (Count (Distinct C#))=(Select Count(C#) from Crs))
```

در Query مثال قبل نام اساتیدی که همه درس ها را ارائه کرده بودند

را بدست آوردیم حال با مقایسه نام اساتید با نام اساتید موجود در Query مثال قبل بدین صورت که اگر نام استادی در آن Query باشد

یعنی تمام درس ها را ارائه کرده است و می توان از رابطه prof مشخصات کامل اساتید را بدست آورد.

مثال. نام اساتید با مدرک دکتری که همه درس ها را ارائه کرده اند.

در مثال بالا کافی است در شرط where عبارت " (دکتری degree, And) اضافه شود.

دستورات اضافه کردن داده به جدول:

صورت کلی.

```
insert into table - name
values (v1...vn)
```

دستور بالا مقادیر v_1 تا v_n را به رابطه $table-name$ اضافه می نماید و در واقع یک تاپل به رابطه اضافه می کند.

```
insert into stud
```

```
values (10, 13.22, 'مشور', 'کریمی', 743622)
```

مثال.

در دستور بالا مشخصات ذکر شده به ترتیب به هر یک از ستون های جدول Stud اضافه می شوند.

□ در دستور insert مشخصات v_1 تا v_n به گونه ای در جدول قرار می گیرند که v_1 به اولین ستون و v_n به آخرین ستون اضافه می شود.

حال در صورتی که بخواهیم، مشخصاتی که وارد می کنیم به درستی وارد ستون های مربوطه خود شوند از فرمت زیر استفاده می کنیم

```
insert into table - name (Ai, ..., Aj)
values (vi, ..., vj) 1 ≤ i, j ≤ n
```

با استفاده از این فرمت هر یک از مقادیر v_i تا v_j به ترتیب وارد ستون های A_i تا A_j میشوند

```
insert into stud (city, sn, avge, sname)
```

```
values ('تهران', 7425306, 14.25, 'کریمی')
```

مثال.

□ می توان فروبی دستور select را در یک جدول ذخیره کرد یا به تاپل های یک جدول اضافه نمود.

```
insert into good - student
```

```
select * from stud
```

```
where avge > 17
```

مثال.

$good-stud$ رابطه ای است همانند رابطه Stud، ولی دانشجویانی که معدل بالای 17 هستند در آن قرار می دهیم.

دستور حذف کردن:

صورت کلی.

```
Delete from table - name
[where conditions]
```

مثال. Delete from stud

این دستور تمام مقادیر جدول Stud را حذف می نماید، ام شمای جدول همپنان باقی است

□ دستور Drop table مفتویات آن را همراه با شمای آن حذف خواهد کرد.

```
Delete from stud
where avge > 17
```

□ در صورتی که بخواهیم تنها مقادیر خاصی از جدول حذف شوند از شرط استفاده می کنیم مثلا به این شکل

دستور بهنگام سازی (Update):

صورت کلی.

```
Update table - name set Ai = vi, ..., Ak = vk
[where conditions]
```

□ در بهنگام سازی رکوردها موجودند و تنها مقادیر صفات خاصه آن را تغییر می دهیم.

```
Update Crs set cnamee = 'شیمی آلی'
Where Cname = 'شیمی معدنی'
```

مثال. Query بنویسید که نام درس شیمی آلی را به شیمی معدنی تغییر دهد.

مثال. دستوری بنویسید که معدل دانشجویان کمتر از 14 را در ضریب 1.2 و بزرگتر از 14 را در ضریب 1.1 ضرب کند. اگر ابتدا معدل دانشجویانی که معدل کمتر از 14 دارند در 1.2 ضرب کنیم ممکن است معدل آنها مقداری بیشتر از 14 گردد، و سپس دوباره معدل آنها را چون مقداری بیش از 14 شده در 1.1 ضرب خواهد شد. بنابراین ابتدا باید معدل های بیشتر از 14 را در 1.1 ضرب نمائیم و سپس معدل های کمتر از 14 را در 1.2 ضرب شوند.

```
Update stud Set avge=
Case
Where avge>14 Then avge=avge*1.1
else avge=avge*1.2
end
```

□ می توان در عبارت Case برای مقایسه چند دستور از when else نیز استفاده نمود.

مثال. Query بنویسید که نام درس ریاضی عمومی 1 به ریاضیات پایه و تعداد واحد آن را برابر 3 قرار دهد.

```
Update Crs Set cname='ریاضیات پایه',unit=3
where cname='1 ریاضیات'
```

□ در صورتی که از شرط where استفاده نشود، نام تمام دروس به ریاضیات پایه و تعداد واحد آنها به 3 تغییر می کند.