

## آموزش زبان اس کیوال - استاد رمزی

### زبان SQL :

SQL مخفف Structerd Query Language یکی از محبوب ترین زبان ها برای ذخیره سازی اطلاعات میباشد. این زبان قادر است تا برای یک پایگاه داده عمل جستجو و گزینش اطلاعات را انجام دهد و همچنین اطلاعات ذخیره شده در یک پایگاه را بازیابی، حذف، اضافه و یا به روز رسانی کند. زبان Sql جنبه های پیاده سازی جبر رابط ای را پشتیبانی میکند.

| S - عرضه کنندگان |            |       |        |
|------------------|------------|-------|--------|
| S#               | SName      | City  | Status |
| S1               | فن آوران   | تهران | 10     |
| S2               | ایران قطعه | تبریز | 20     |
| S3               | پولادین    | تبریز | 20     |

| P - قطعات |       |           |        |       |
|-----------|-------|-----------|--------|-------|
| P#        | Color | Type      | Weight | City  |
| P1        | قرمز  | آهن       | 5      | تهران |
| P3        | سبز   | مس        | 7      | تبریز |
| P4        | آبی   | برنز      | 5      | شیراز |
| P5        | قرمز  | آلومینیوم | 6      | تهران |

| SP - واسط |    |     |
|-----------|----|-----|
| S#        | P# | QTY |
| S1        | P1 | ۳۰۰ |
| S1        | P2 | 200 |
| S1        | P3 | 400 |
| S2        | P1 | 300 |
| S2        | P2 | 400 |
| S3        | P2 | 200 |

- ایجاد جدول (Create Table) :

جداول مهمترین عناصر سیستم های پایگاه داده هستند، که برای ذخیره و نگهداری سازمان یافته اطلاعات مورد استفاده قرار میگیرند. به سطر های یک جدول رکورد (Record) و به هر یک از خانه های ستون یک جدول فیلد (Feild) میگویند.

```
Create table "Name Table" (  
    S# char(5) not null,  
    P# char(5), not null,  
    QTY int,  
    Primary key (S#,P#),  
    Foreign key (S#) reference S(S#),  
        On delete cascade,  
        On update cascade,  
    Foreign key (P#) reference p(P#),  
        On delete cascade,  
        On update cascade,  
)
```

- Alter Table (تغییر دادن جداول) :

با استفاده از این دستور میتوانید تغییراتی را در یک جدول موجود اعمال کنید. برای مثال اضافه کردن و یا حذف صفاتی خاص.

```
Alter table S  
Add Scount Smallint
```

✓ دستور فوق بندی به نام Scount از نوع Smallint را به جدول S اضافه میکند.

```
Alter table S
```

```
Alter( SName char(100))
```

✓ دستور فوق نوع داده را تغییر نمیدهد بلکه طول SName را افزایش میدهد.

```
Alter table S
```

```
Drop SName
```

✓ حذف فیلد SName از جدول S

• Drop table :

از این دستور برای از بین بردن جدول و محتوای آن استفاده می شود.

```
Drop table S
```

• SELECT

مهمترین دستور SQL است و برای بازیابی اطلاعات از آن استفاده میشود. ساده ترین فرمت دستور

Select به شکل زیر است:

مراحلی که داخل براکت هستند اختیاری اند

```
Select [Distinct] نام فیلد ها
```

```
From نام جدول
```

```
[Where جستجو]
```

```
[Group By فیلد ها]
```

```
[Having شرط]
```

```
[Order By فیلد ها]
```

مثال: شماره و نام عرضه کنندگانی را که در شهر C2 هستند را به دست آورید.

```
Select S#,SName
```

```
From S
```

```
Where city = 'C2'
```

مثال: شماره تمام عرضه کنندگان را به دست آورید.

```
Select S# from S
```

مثال: شماره تمام قطعات را بدون تکرار از جدول SP به دست آورید.

```
Select distincts P# from SP
```

مثال: شماره تمام قطعات را با تکرار از جدول SP به دست آورید.

```
Select P# from SP
```

مثال: شماره و وزن (به گرم) قطعات تولید شده را به دست آورید.

```
Select P#, weight * 1000 as weight from P
```

خروجی :

| P# | Weight |
|----|--------|
| P1 | 5000   |
| P2 | 7000   |
| P3 | 5000   |
| P4 | 6000   |

• ORDER BY :

برای مرتب سازی است نزولی و صعودی (ASSENDING/DESENDING) و پیش فرض آن ASSENDING است.

مثال: عرضه کنندگانی را که ساکن شهر تهران هستند را بر اساس مقادیر نزولی وضعیت نمایش دهید.

```
Select S#,Status From S
Where city = 'تهران'
Order by status DESC
```

خروجی :

| S# | Status |
|----|--------|
| S3 | 30     |
| S1 | 20     |
| S4 | 20     |

نکته : زمانی که از دستور Order By استفاده می کنیم می بایست حتما فیلد مورد نظر در خروجی وجود داشته باشد: دستور زیر اجرا نمی شود چون فیلد City در خروجی وجود ندارد.

```
Select S# from S
Order By City
```

○ مرتب سازی بر اساس ۲ فیلد

در مثال زیر در صورتی که فیلد weight برابر باشد بر اساس P# آنها را مرتب میکند.

```
Select P#, weight from P
Order by weight, p#
```

خروجی :

| P# | weight |
|----|--------|
| P1 | 5      |
| P3 | 5      |
| P4 | 6      |
| P2 | 7      |

• عملکرد In و Between در SELECT

به کمک Between میتوان وجود یک مقدار در یک بازه و به کمک IN می توان وجود یک مقدار را در مجموعه ای از مقادیر بررسی کرد.

Between  $x \in [a, b]$

In  $x \in \{a, b, c, d, \dots\}$

مثال : شماره قطعه، رنگ و وزن قطعاتی را به دست آورید که وزن آنها بین ۶ تا ۱۹ باشد :

```
Select P#, color, weight From P
Where weight Between 6 And 19
```

خروجی :

| P# | Color | Weight |
|----|-------|--------|
| P2 | سبز   | ۷      |
| P4 | قرمز  | ۶      |

مثال : شماره قطعه و وزن قطعاتی را به دست آورید که وزن آنها بین ۶ تا ۱۹ نباشد :

```
Select P#, Weight From P
Where weight not Between 6 And 19
```

خروجی :

| P# | Weight |
|----|--------|
| P1 | 5      |
| P3 | 5      |

مثال : شماره قطعه، رنگ و وزن قطعاتی را به دست آورید که وزن آنها در مجموعه (2,5,7) باشد :

Select P#, weight,Color From P

Where weight in (2,5,7)

خروجی :

| P# | Weight | Color |
|----|--------|-------|
| ۱  | ۵      | قرمز  |
| ۲  | ۷      | سبز   |
| ۳  | ۵      | آبی   |

TXT

|       |      |
|-------|------|
| جستجو | حسین |
|-------|------|

• جستجوی عین عبارت :

Select x from Student

Where Family=TXT

• عملگر Like (بخشی از عبارت) :

از این عملگر برای جستجو در بخشی از عبارت مورد نظر استفاده قرار میگیرد.

○ علامت % به جای مجموعی ای از کاراکترها

○ علامت \_ به جای یک کاراکتر

مثال : مشخصات قطعاتی را گزارش دهید که با حرف " آ " شروع شده اند.

Select \* from P

Where Type like 'آ%'

خروجی :

| P# | Color | Type      | City  | Weight |
|----|-------|-----------|-------|--------|
| ۱  | قرمز  | آهن       | تهران | 5      |
| ۴  | قرمز  | آلومینیوم | تهران | ۶      |

مثال : مشخصات قطعاتی که شامل حرف " آ " نیستند.

```
Select * from P
Where P# Not like '%آ%'
```

خروجی :

| P# | Color | Type | City  | Weight |
|----|-------|------|-------|--------|
| ۲  | سبز   | مس   | تبریز | ۷      |
| ۳  | آبی   | برنز | شیراز | ۵      |

مثال : مشخصات قطعاتی را گزارش دهید که 2 حرفی باشند و حرف " م " در ابتدا قرار داشته باشند.

```
Select * from P
Where Type like 'م_'
```

خروجی :

| P# | Color | Type | City  | Weight |
|----|-------|------|-------|--------|
| ۲  | سبز   | مس   | تبریز | ۷      |

مثال : مشخصات قطعاتی را گزارش دهید که حرف " ر " دومین حرف باشد و با حرف " ز " خاتمه یابد.

```
Select * from P
Where Type Like 'ز%ر_'
```



خروجی :

| P# | Color | Type | City  | Weight |
|----|-------|------|-------|--------|
| ۳  | آبی   | برنز | شیراز | ۵      |

• پیوند جداول :

پیوند نوع خاصی از پرس و جو (Query) است که طی آن عمل بازیابی اطلاعات از بیش از یک جدول صورت میگیرد.

○ پیوند طبیعی :

مثال : شماره و نام شهر های قطعات و عرضه کنندگانی را که در یک شهر هستند را به دست آورید.

```
Select S.S#, S.city, P.P#, P.city
From S, P
Where S.city = P.city
```

خروجی :

| S# | City  | P# | City  |
|----|-------|----|-------|
| S1 | تهران | P1 | تهران |
| S1 | تهران | P4 | تبریز |
| S2 | تبریز | P2 | تبریز |
| S3 | تبریز | P2 | تهران |

○ ضرب دکارتی :

مثال : مقادیر ۲ جدول S و P را به دست آورید.

```
Select S.*, P.*
From S, P
```

نمایش دیگر :

```
Select S.*, P.*  
From S Cross Join P
```

○ INNER JOIN (پیوند طبیعی): تمام سطرهایی را از ۲ جدول مورد نظر به دست میآورد که در یک فیلد مشترک باشند

مثال: نام تمام شرکت های تولید کننده قطعات را به همراه اطلاعات قطعات تولید کرده هر شرکت (نام عرضه کنندگان قطعات و اطلاعات قطعات عرضه شده را) به دست آورید.

```
Select S.SName , P.*  
From S INNER JOIN SP  
On S.S# = SP.S#  
SP INNER JOIN P  
On SP.P# = P.P#
```

خروجی :

| SName      | P# | Color | City  | Weight | Type |
|------------|----|-------|-------|--------|------|
| فن آوران   | ۱  | قرمز  | تهران | ۵      | آهن  |
| فن آوران   | ۲  | سبز   | تهران | ۷      | مس   |
| فن آوران   | ۳  | آبی   | شیراز | ۵      | برنز |
| ایران قطعه | ۱  | قرمز  | تهران | ۵      | آهن  |
| ایران قطعه | ۲  | سبز   | تهران | ۷      | مس   |
| پولادین    | ۲  | سبز   | تهران | ۷      | مس   |

● SELECT متداخل:

مثال: نام عرضه کنندگانی را بیآورید که قطعه P2 را تهیه کرده اند.

○ روش اول

```
Select SName From S,SP  
Where (S.S# = SP.S#) and (SP.P# = p2)
```

○ روش دوم

```
Select SName From S
```

```
Where S# in (Select S# from SP where P# = p2)
```

○ روش سوم (اولویت ها)

```
Select SName From S INNER JOIN SP ON S.S# = SP.S# & P# = P2
```

خروجی هر ۳ مورد به شکل زیر است :

|            |
|------------|
| SName      |
| فن آوران   |
| ایران قطعه |
| پولادین    |

مثال : شماره عرضه کنندگانی را بیابید که در همان شهر ساکن باشند که عرضه کنندگان S2 ساکن هستند.

```
Select S# From S
```

```
Where city in (Select city from S Where S# = S2)
```

خروجی :

|    |
|----|
| S# |
| 2  |
| 3  |

● : Group By

مثال : تعداد کل عرضه شده از هر قطعه را همراه با شماره قطعه گزارش دهید.

```
Select P#, SUM(QTY) AS SUM From SP
```

```
Group by P#
```

QTY را جمع بزن و در SUM قرار بده

بر اساس P# گروه بندی کن

خروجی :

| P# | SUM |
|----|-----|
| 1  | 600 |
| 2  | 800 |
| 3  | 400 |

نکته: صفتی که گروه بندی بر اساس آن صورت میگیرد حتما می بایست در خروجی آورده شود.

● : Having

همواره با Group By استفاده میشود. نقش Having در گروه مانند نقش Where در سطر است. به عبارت دیگر Having برای اعمال شروط بر روی گروه ها مورد استفاده قرار میگیرد. مثال : شماره قطعه، قطعاتی را گزارش دهید که توسط بیش از یک عرضه کننده ، عرضه شده باشد.

```
Select P#  
From SP  
Group By P#  
Having Count(*) > 1
```

خروجی :

| P# |
|----|
| 1  |
| 2  |

توابع ستونی در دستوری Select :

Count ○

مثال : شماره عرضه کنندگانی را بدهید که بیش از یک قطعه تولید میکنند.

```
Select S#  
From SP  
Group By S# Having COUNT(*) > 1
```

خروجی :

| S# |
|----|
| 1  |
| 2  |

مثال:

```
Select COUNT(*)
From Students
Where family = 'Dana'
```

AVG ○

Sum ○

مثال: تعداد کل عرضه شده از قطعه P2 را به دست آورید.

```
Select SUM(QTY) AS SUM
From SP
Where p# = 2
```

خروجی :

| SUM |
|-----|
| 800 |

نکته: میتوان با استفاده از کلمه کلیدی AS نام یک ستون را تغییر داد.

Max ○

مثال: قطعه ای که از آن بیشترین تعداد تولید میشود را به دست آورید.

```
Select Max(SUMQTY)AS MAX From(
Select SUM(QTY) AS SUMQTY
From SP GROUP BY P#) F
```

خروجی :

|     |
|-----|
| MAX |
| 800 |

توضیح: در قطعه کد بالا ابتدا مجموع تولید شده از هر قطعه را به دست می آوریم و نامش را SUMQTY قرار میدهیم، سپس در بین آنها بزرگترین را به دست می آوریم. کلمه F را در نهایت برای اینکه عمل اولیه یعنی به دست آوردن مجموع از هر قطعه را به صورت یک جدول به عمل بعدی معرفی کنیم می نویسیم.

MIN ○

مثال: قطعه ای که از آن کمترین تعداد تولید میشود را به دست آورید.

همانند بالا با این تفاوت که بجای MAX از MIN استفاده می کنیم.

COUNT ○

مثال: تعداد عرضه کنندگان شهر تبریز چننا است.

```
Select COUNT(S#) AS COUNT From S
Where city = 'تبریز'
```

خروجی :

|       |
|-------|
| COUNT |
| 2     |

• Insert :

از این دستور برای ثبت اطلاعات استفاده میشود و به شکل زیر می تواند باشد:

○ روش اول

```
Insert into <نام جدول> values (نام فیلد ها)
```

مثال :

```
Insert into P values ('P','C1',24)
```

مثال :

```
Insert into Students (S#,Family,Group)
Values('9143061','Dana','Computer sience')
```

○ روش دوم

```
Insert into <نام جدول> [نام فیلد ها] <Query>
```

مثال :

```
Insert into Temp(P#,total QTY)
Select P#, Sum(QTY) From
Group By P#
```

● Update :

از این دستور برای به روز رسانی مقادیر فیلد ها استفاده میشود و به شکل زیر استفاده میشود.

```
Update <نام جدول> Set <فیلد ها> = <مقدار> Where <شرط>
```

مثال : شماره عرضه کنندگان S1 را به S10 تغییر دهید.

```
Update S set S# = S10 where S# = S1
```

مثال : تعداد را در محموله های عرضه کنندگان ساکن C3 صفر کنید.

```
Update SP Set QTY = 0
Where S# in (
select S# From S where city = C3)
```

مثال :

Update Students

Set Family = 'Dashti' where Family 'Dana'

• Delete :

برای حذف کردن یک یا چند رکورد از یک جدول خاص. در صورتی که شرط خاصی برای این دستور استفاده نکنیم تمامی فیلدهای جدول مورد نظر از بین می‌رود.

Delete From <نام جدول> [Where <شرط>]

مثال: عرضه کنندگان S5 را حذف کنید.

Delete From S where S# = S5

مثال: مقادیر جدول SP را حذف کنید.

Delete From SP

مثال :

Delete From Students Where Family = 'Dashti'

• Drop :

برای از بین بردن یک جدول استفاده میشود. با اجرای این دستور تمام کلیدهای خارجی جدول به طور خودکار از بین می‌رود.  
مثال: از بین بردن جدول S.

Drop table S



## • View

در SQL یک View یک جدول مجازی است که نتیجه حاصل از عبارات و دستورات SQL می باشد. یک view شامل سطر و ستونهایی دقیقا شبیه یک جدول واقعی است. فیلدها در یک View فیلدهایی هستند که از یک یا چند جدول واقعی در پایگاه داده گرفته شده اند.

```
CREATE VIEW AS نام جدول مجازی
SELECT نام فیلدها
FROM نام جدول
WHERE شرط
```

مثال :

### Creat View

```
SELECT dbo.P.Color, dbo.P.Type, dbo.SP.QTY, dbo.S.Status,
       dbo.S.City, dbo.S.SName
FROM   dbo.P INNER JOIN
       dbo.SP ON dbo.P.P# = dbo.SP.P# INNER JOIN
       dbo.S ON dbo.SP.S# = dbo.S.S#
```