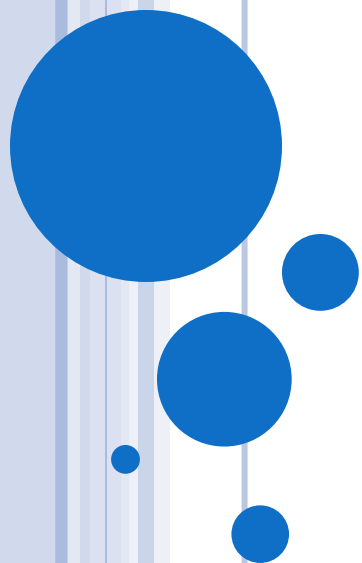


فصل چهارم

زبان رابطه ای SQL



اصولاً هر RDBMS دارای نسخه SQL خاص خود است. ویژگی های SQL عبارت است از:

- ۱- زبانی غیر رویه ای
- ۲- دارای عملگرهای بسیار قوی
- ۳- تامین کننده استقلال داده ای
- ۴- دارای اکمال ساختاری
- ۵- شامل تمام داده های استاندارد
- ۶- قابل استفاده هم به صورت مستقل و هم به صورت ادغام شده

انواع عملگرها در SQL

توسط عملگرها، عمل خاصی را انجام می دهیم. انواع عملگرها عبارتند از:

۱- محاسباتی (+, -, *, /, %)

۲- رابطه ای (=, >, <, <=, >=, !=)

۳- منطقی (AND, OR, NOT)

۴- بیتی (^, |, &, ~)

۵- انتساب (=)

۶- ویژه (IN, BETWEEN, ALL, ANY, LIKE)

دستورهای SQL

دستورات در SQL را می توان به سه دسته زیر تقسیم بندی کرد:

۱- DML : (SELECT , INSERT , UPDATE , DELETE)

۲- DDL : (CREATE , ALTER , DROP)

۳- DCL : (GRANT , REVOKE)

ایجاد جدول (CREATE TABLE)

مثال

ایجاد جدول دانشجو (S) با فیلد های شماره دانشجویی (کلید) و نام دانشجو:

```
CREATE TABLE S  
(  
    ID      Char (10)  PRIMARY KEY ,  
    Name    Char (20),  
);
```

حذف جدول (DROP TABLE)

برای حذف یک جدول از دستور DROP TABLE ، استفاده می شود. وقتی جدولی حذف شود، فایل متناظر جدول حذف شده و تعریف جدول از کاتالوگ خارج می شود. برای حذف جدول S ، از دستور زیر استفاده می کنیم:

```
DROP TABLE S;
```

درج (INSERT)

از این دستور برای اضافه کردن رکورد به جدول بانک اطلاعاتی استفاده می شود.

مثال

درج اطلاعات یک دانشجو به جدول STUDENT:

```
INSERT INTO S (ID,Name)  
VALUES ('110' , 'Ali' );
```

بهنگام سازی (UPDATE)

برای ویرایش رکوردهای جدول از دستور UPDATE استفاده می کنیم.

مثال

شماره دانشجویی 100 را به 200 تغییر دهید.

```
UPDATE S  
SET      ID=200  
WHERE    ID=100;
```

بازیابی (SELECT)

مثال

بازیابی شماره و نام دانشجویان از جدول S :

```
SELECT ID, Name  
FROM S;
```

مثال

بازیابی شماره دانشجویانی که نام آنها ALI باشد از جدول S :

```
SELECT ID  
FROM S  
WHERE Name='ALI';
```

مثال

بازیابی شماره و نام دانشجویانی که نام آنها ALI باشد:

```
SELECT *  
FROM S  
WHERE Name='ALI';
```

مثال

بازیابی نام دانشجویان بدون نمایش نام های تکراری:

```
SELECT DISTINCT SNAME  
FROM S;
```

توابع جمعی (Aggregate Functions)

توابع جمعی در قسمت جبر رابطه ای توضیح داده شد. این توابع عبارتند از:

COUNT , MAX , MIN , SUM , AVG

مثال

پرس و جویی که تعداد دانشجویان را بر می گرداند:

```
SELECT COUNT(*)  
FROM S;
```

مرتب سازی رکوردها

توسط امکان ORDER BY می توان جدول جواب را برحسب یک یا بیش از یک ستون به صورت صعودی (ASC) یا نزولی (DESC) مرتب کرد. (ASC پیش فرض است).

```
SELECT *  
FROM S  
ORDER BY ID DESC;
```

تست مقدار تهی فیلد

توسط این امکان می توان وجود مقدار هیچ در یک ستون را بررسی کرد.

مثال

شماره دانشجویانی را بدهید که نمره آنها در درس C1 هنوز اعلام نشده است.

```
SELECT SID  
FROM SC  
WHERE CID='C1' AND GRADE IS NULL;
```


عملگر LIKE

توسط این امکان در SQL می توان عمل بازیابی را بر اساس نشانوند جستجوی کاراکتر با شرایط مورد نظر انجام داد. به عبارتی مشخص می کند که آیا رشته ای در قسمتی از فیلد قرار دارد یا خیر. تذکر: به جای کاراکتر - می تواند یک کاراکتر و به جای کاراکتر % می تواند تعدادی کاراکتر قرار گیرد.

مثال

مشخصات دانشجویانی را بدهید که نام آنها ۴ حرفی است و حرف دوم در نام آنها A باشد، مانند SARA.

```
SELECT *  
FROM ST  
WHERE SNAME LIKE 'A--';
```

مثال

مشخصات دانشجویانی را بدهید که نام آنها با کاراکتر A شروع شود.

```
SELECT *  
FROM ST  
WHERE SNAME LIKE 'A%';
```

مثال

مشخصات دانشجویانی را بدهید که نام آنها ۳ حرفی است و حرف اول، یکی از کاراکترهای A,B,C,D باشد.

```
SELECT *  
FROM ST  
WHERE SNAME LIKE '[A-D]--';
```

مثال

مشخصات دانشجویانی را بدهید که نام آنها ۳ حرفی است و حرف اول یکی از کاراکترهای A,B,C,D نباشد:

```
SELECT *  
FROM ST  
WHERE SNAME LIKE '[^A-D]--';
```

عملگر UNION

مثال

با فرض اینکه اطلاعات دانشجویان در بانک S و اطلاعات اساتید در بانک T قرار دارد، برای مشخص کردن نام دانشجویان و اساتید از دستور زیر استفاده می کنیم:

```
(SELECT NAME FROM S )  
UNION  
(SELECT TNAME FROM T );
```

عملگر BETWEEN

توسط این امکان می توان وجود یک مقدار را در یک محدوده بررسی کرد.

مثال

شماره دانشجویانی را بدهید که نمره آنها در درس C1 بین ۱۰ تا ۱۲ باشد.

```
SELECT SC.SID  
FROM SC  
WHERE CID='C1' AND GRADE BETWEEN 10 AND 12;
```

گروه بندی اطلاعات

توسط امکان گروه بندی، می توان سطرهای یک جدول را برحسب مقادیر یک ستون گروه بندی کرد به نحوی که در هر گروه مقدار آن ستون یکسان باشد.

مثال

گروه بندی جدول R بر حسب ستون Y :

```
SELECT *  
FROM R  
GROUP BY Y;
```

X	Y	Z
X1	Y1	Z1
X1	Y2	Z2
X2	Y1	Z2
X3	Y1	Z3
X3	Y3	Z4

 $\Rightarrow$

X	Y	Z
X1	Y1	Z1
X2	Y1	Z2
X3	Y1	Z3
X1	Y2	Z2
X3	Y3	Z4

مثال

گروه بندی جدول دانشجو - درس برحسب مقادیر ستون شماره CID :

```
SELECT SID , AVG(GRADE)
FROM SC
GROUP BY CID;
```

استفاده از HAVING در دستور SELECT

اگر بخواهیم از توابع جمعی در شرط WHERE استفاده کنیم، پیام خطا صادر می شود و باید از HAVING به جای WHERE استفاده کرد.

بنابراین دستور زیر نادرست است:

```
SELECT S# , AVG(QTY)
FROM SP
WHERE AVG(QTY) > 100
GROUP BY S#;
```

دستور صحیح به صورت زیر می باشد:

```
SELECT S#,AVG(QTY)
FROM SP
GROUP BY S#
HAVING AVG(QTY) > 100;
```

پیوند رابطه ها

برای پاسخ به بعضی از پرسش ها نیاز است که به بیش از یک جدول مراجعه شود. در این حالت از پیوند باید استفاده شود.

مثال

نام دانشجویانی را بدهید که درس شماره C3 را انتخاب کرده اند.

```
SELECT ST.SNAME
FROM ST, SC
WHERE ST.SID = SC.SID AND SC.CID = 'C3 ';
```