

به نام خدا

## کلیاتی درباره SQL و املا (Syntax) دستورات

SQL به دو قسمت تقسیم میشود :

۱- زبان دستکاری داده ها (DML (Data Manipulation Language):  
دستورات DML شامل بخش‌های زیر می‌شوند

SELECT واکنشی (انتخاب) اطلاعات از دیتابیس  
UPDATE ویرایش اطلاعات دیتابیس  
DELETE پاک کردن اطلاعات از دیتابیس  
INSERT INTO اضافه کردن اطلاعات جدید به دیتابیس

۲- زبان تعریف داده ها (DDL (Data Definition Language):  
دستورات DDL شامل بخش‌های زیر می‌شوند

CREATE DATABASE ایجاد یک دیتابیس جدید  
ALTER DATABASE ایجاد تغییرات در دیتابیس  
CREATE TABLE ایجاد یک table جدید  
ALTER TABLE اعمال تغییرات در table  
DROP TABLE پاک کردن یک table  
CREATE INDEX ایجاد یک شاخصه  
DROP INDEX حذف یک شاخص

Syntax دستورات DML

۱- دستور واکنشی اطلاعات از پایگاه داده

دستور ساده : نام جدول یا جدولها from نام فیلد یا فیلدها Select

دستور کامل : شرط(ها) Where نام جدول(ها) from نام فیلد(ها) Select  
Orderby نام فیلد(ها) +ASC/DESC

## ۲- دستور ویرایش اطلاعات دیتابیس

Update (ها) شرط Where ... , مقدار جدید = فیلد , مقدار جدید = فیلد Set نام جدول Update

## ۳- دستور پاک کردن اطلاعات از دیتابیس

Delete from جدول Where (ها) شرط

## ۴- دستور اضافه کردن اطلاعات جدید به دیتابیس

Insert Into نام جدول (نام فیلدها) نام جدول Select نام فیلدها from دستور نوع اول نام جدول (پشتیبان)

Insert Into نام جدول (نام فیلدها) Values (مقدار فیلدها) دستور نوع دوم

## پیوند جدولها

برای انتخاب اطلاعات از دو یا چند جدول :

الف- می توان در قسمت From نام جدولها را با علامت , از هم جدا و نوشت و در قسمت Where ارتباط جدولها را بصورت شرایط دو به دو بیان کرد:

مثال:

```
SELECT CourseCode,CourseName FROM Tbl_Course,Tbl_RegisterItem WHERE  
Tbl_Course.CourseCode=TblRegisterItem.CourseCode
```

ب- می توان جدولها را پیوند زد:

پیوند (JOIN) دو جدول در واقع نوع خاصی از ضرب دو جدول (رابطه) است که عمل پیوند دو جدول (رابطه) را بر اساس ستون (ستونهای) مشترک انجام میدهد.

مثال: فرض کنید رابطه های R1 و R2 بصورت:

## R1 R2 نتیجه ضرب

S#	Sname	S#	P#	QTY	=	S.S#	Sname	SP.S#	P#	QTY
S1	رضا	S1	P1	100		S1	رضا	S1	P1	100
S2	علی	S1	P2	80		S1	رضا	S1	P2	80
						S2	علی	S1	P1	100
						S2	علی	S1	P2	80

S2 که علی است S1 از قطعه P1 مقدار 100 تا را تولید کرده است. (چه جمله بی ربطی)

S2 که علی است S1 از قطعه P2 مقدار 80 تا را تولید کرده است. (چه جمله بی ربطی)

اما عمل Join که نوع خاصی ضرب است عمل ضرب را فقط بر اساس ستون (ستونهای) مشترک انجام می دهد که حاصل آن بصورت ذیل خواهد بود که فاقد دو سطر آخر است که بی مفهوم بودند.

## R1 R2 JOIN نتیجه

S#	Sname	JOIN	S#	P#	QTY	=	S.S#	Sname	SP.S#	P#	QTY
S1	رضا		S1	P1	100		S1	رضا	S1	P1	100
S2	علی		S1	P2	80		S1	رضا	S1	P2	80

## انواع JOIN

الف – INNER JOIN: اتصال دو جدول در زمانی که اطلاعات متناظر در هر دو جدول وجود داشته باشند یعنی اطلاعاتی را نمایش می‌دهد که در هر دو جدول وجود داشته باشد ولی اطلاعاتی که در یک جدول باشد ولی دیگری وجود نداشته باشد را در نتیجه پرس و جو نمایش نمی‌دهد.

Syntax: SELECT نام فیلدها FROM نام جدول اول INNER JOIN نام جدول دوم ON فیلد مشترک در جدول دوم = فیلد مشترک در جدول اول

مثال: اطلاعات معلمان هر درس را بدست آورید.

جواب: پرس و جو را از اطلاعات دو جدول خواسته پس باید دو جدول را پیوند زده و فیلدها را واکنشی کرد از طرفی چون شرطی بیان نکرده پس جدول معلمان با جدول درس را باید INNER JOIN کرد.

ب – (Right/Left) OUTER JOIN: اتصال دو جدول بصورتی که اطلاعات متناظر دو جدول در نتیجه پرس و جو مشاهده می‌شود حتی اگر در جدول دیگر وجود نداشته باشد. حال اگر تاکید روی جدول اول باشد LEFT JOIN ولی اگر تاکید روی جدول دوم باشد RIGHT JOIN خواهد بود.

مثال ۱: اطلاعات معلمان هر درس را بدست آورید در ضمن معلمانی که در این ترم درسی ارائه نداده‌اند نیز آورده شود.

جواب: جدول معلمان جدول اولی و جدول درس دومی است اولاً چون شرط آورده OUTER JOIN است ثانیاً چون تاکید روی جدول اولی است پس LEFT JOIN است. پس جدول معلمان با جدول درس را باید LEFT JOIN کرد.

مثال ۲: اطلاعات معلمان هر درس را بدست آورید در ضمن دروسی که در این ترم ارائه نشده‌اند نیز آورده شود.

جواب : جدول معلمان جدول اولی و جدول درس دومی است اولاً چون شرط آورده OUTER JOIN است ثانیاً چون تاکید روی جدول دومی است پس RIGHT JOIN است. پس جدول معلمان با جدول درس را باید RIGHT JOIN کرد.

## چند عبارت مهم

۱- عبارت ORDER BY : زمانی که در دستور SELECT بخواهیم رکوردهای واکنشی شده را مرتب سازی هم کنیم از این عبارت در انتهای دستور SELECT استفاده میکنیم که اگر مرتب سازی صعودی بخواهیم ASC ولی اگر نزولی بخواهیم DESC بکار می‌بریم.

Select نام فیلد(ها) from نام جدول(ها) Where شرط(ها) Orderby نام فیلد(ها)+ASC/DESC

۲- عبارت GROUP BY :

در پرس و جویهای تجمعی (Aggregate) شامل جمع کل ، تعداد کل و... اگر بخواهیم بر اساس یک فیلد خاص گروه بندی کنیم از این عبارت استفاده می‌کنیم.

مثال : در رابطه ذیل

رابطه R

S#	City
S1	نقده
S2	نقده
S3	اشنویه
S4	اشنویه
S5	اشنویه

می‌خواهیم بگوئیم که فرخنا نقده ۲ و اشنویه ۳

نه اینکه S1 نقده ، S2 نقده ، S3 اشنویه ...

که بایستی از گروه‌بندی استفاده کنیم.

پرسش : نام شهر و تعداد تولید کنندگان ساکن آن را شهر را از رابطه فوق استخراج کنید.

جواب : `SELECT City,Count(*) FROM R GROUP BY(City)`

نکته مهم : هنگام استفاده از عبارت `GROUP BY` بایستی تمام فیلدهایی که در مقابل `SELECT` آمده‌اند بغیر از آنهایی که در توابع تجمعی `Sum` ، `Min` ، `Count` استفاده شده‌اند را در لیست قسمت `GROUP BY` آورد.

۳- عبارت `HAVING` :

میدانیم که در املاء دستور `SELECT` شرط را با عبارت `Where` بیان می‌کردیم اما در پرس و جوهای تجمعی که از عبارت `GROUP BY` استفاده می‌کنیم شرط را بایستی با عبارت `HAVING` بیاوریم نه `Where`

عبارت `HAVING` همیشه با `GROUP BY` همراه می‌شود و هیچگاه به تنهایی استفاده نمی‌شود و شرط را روی گروه‌های ساخته شده اعمال می‌کند.

مثال : در رابطه مثال قبل ، نام هر شهر و تعداد تولید کنندگان آن شهر را در صورتیکه تعداد آنها بیشتر از ۱۰ باشد را برگردانید.

جواب : `SELECT City,Count(*) FROM R GROUP BY(City) HAVING Count(*)>10`