

بِسْمِ تَعَالَى

آموزش دستورات T-SQL و راهنمای استفاده در Delphi7

WWW.SoftGozar.Com

در این تاپیک میخوام آموزشهای مربوط به استفاده از دستورات SQL در دلفی رو بذارم.
به چند دلیل:

- برای استفاده از اکثر دستورات sql نوع دیتابیس مورد استفاده در پروژه مهم نیست (paradox,ado,interbase,...)
- بیشتر مشکلاتی که در تالارهای برنامه نویسی مطرح شدند رو میشه با استفاده از دستورات sql خیلی راحت تر حل کرد
- SQL در زبانهای برنامه نویسی دیگه هم قابل استفاده هست پس یادگیری اون لازمه

چند نکته برای شروع:

- در تمام آموزشها فرض بر این هست که دوستان در ارتباط دادن پایگاه داده با برنامه کاربردی مشکلی ندارند
- دستورات sql به حروف کوچک و بزرگ حساس نیستند (select با SELECT فرقی ندارد)
- تمامی دستوراتی که در اینجا گفته میشه، تست شده هست
- دستورات در کامپوننت های ADO به کار گرفته میشه

مقدمه ای بر SQL (Structured Query Language) :

SQL زبان پرس و جویی هست که توسط ANSI (American National Standards Institute) استاندارد شده و برای دستکاری دیتابیس ها، بازیابی و update داده ها در بانک های اطلاعاتی مفید می باشد. SQL در دیتابیسهای MS Access, Paradox, DB2, Informix, MS SQL Server, My SQL, Oracle, Sybase استفاده میشود.

متاسفانه ورژن های مختلفی از این زبان وجود دارد (SQL/PSM, SQL PL, T-SQL, MySQL, PL/SQL, PL/pgSQL , ...) اما برای اینکه همه آنها با استاندارد ANSI سازگاری داشته باشند باید از کلمات کلیدی یکسان در استفاده های مشابه پشتیبانی کنند. (مانند SELECT, UPDATE, DELETE, INSERT, WHERE و غیره)

چرا SQL ؟

SQL زبان پرس و جوی ساخت یافته است
SQL به شما اجازه میدهد تا به آسانی به دیتابیس ها دسترسی داشته باشید
SQL میتواند برای دیتابیس ها پرس و جو ایجاد کند
SQL میتواند داده ها را از دیتابیس بازیابی کند
SQL میتواند رکورد جدیدی در دیتابیس درج کند
SQL میتواند رکوردی را از دیتابیس حذف کند
SQL میتواند رکوردهای دیتابیس را به روز کند
SQL برای یادگیری آسان است

: SELECT

اولین دستوری رو که میخوام بگم دستور select هست.
شکل کلی دستور:

```
SELECT column_name(s) FROM table_name
```

فرض کنید جدولی با نام inventory مربوط به مشخصات کالا داریم که دارای فیلدهای , name , cost , color , weight هست.

```
select name, cost from inventory
```

تمامی فیلدهای مربوط به ستون name, cost رو به ما نشون میده
حالا اگر بخوایم که تمامی فیلدها رو ببینیم نیازی نیست که نام همه فیلدها رو بنویسیم، کافیه به جای نام فیلدها " * " رو قرار بدیم. به صورت زیر:

```
select * from inventory
```

حالا این دستورات رو چطور در دلفی استفاده کنیم؟ از تب ADO به کامپوننت ADOQuery اضافه کنید و کد رو در یه button به صورت زیر بنویسید:

```
ADOQuery1.Close;  
ADOQuery1.SQL.Clear;  
ADOQuery1.SQL.Add('select * from table_name');  
ADOQuery1.Open;
```

: WHERE

سوالی که اینجا هست اینه که چطور نتایج دلخواه رو بر اساس یه مقدار متغیر نمایش بدیم؟ برای این کار باید از دستور where استفاده کرد.

```
SELECT column FROM table WHERE condition
```

در قسمت condition میتونیم از شرط های مختلف استفاده کنیم
فرض کنید در همون جدول مشخصات کالا میخوایم کالاهایی که نام اونها در یک edit نوشته میشه رو نمایش بدیم:

```
ADOQuery1.Close;  
ADOQuery1.SQL.Clear;  
ADOQuery1.SQL.Add('select * from inventory where name=:param');  
ADOQuery1.Parameters.ParamValues['param']:=Edit1.Text;  
ADOQuery1.Open;
```

همونطور که می بینید اینجا از "پارامتر" استفاده شده.
عملگرهای مقایسه ای در sql (که میتونیم برای شرط های مختلف در قسمت condition استفاده کنیم):

```
= , <> , < , > , <= , >=
```

<> به معنای نا مساوی هست.
همچنین میتونید از عملگرهای منطقی AND و OR برای پیاده سازیهای مختلف استفاده کنید.

: ORDER BY

دستور دیگه ای که کاربرد زیادی در select داره ORDER BY هست که برای مرتب سازی نتایج بر اساس یه فیلد خاص استفاده میشه. شکل کلی دستور:

```
SELECT * FROM table_name ORDER BY column
```

توجه: اگر به جای * از نام فیلد استفاده کنید، مرتب سازی باید بر اساس فیلدی باشه که در نتیجه هم نمایش داده میشه. همیشه در قسمت order by از نام چند فیلد استفاده کرد. مزیت این کار این هست که اگر در فیلد اولی دو مقدار یکسان وجود داشته باشه، همون مقادیر بر اساس فیلد دیگه ای هم مرتب میشن.

```
select * from inventory order by name,weight
```

در اینجا اگر دو کالا با نام یکسان وجود داشته باشه، بر اساس وزن مرتب میشن بعد از نام ستونها در قسمت order by اگر DESC رو قرار بدین، نتایج به صورت نزولی مرتب میشن و در صورتی که ASC قرار داشته باشه به صورت صعودی. حالت پیش فرض در زمانی که شما حالت مرتب سازی رو مشخص نکنین، صعودی هست.

: BETWEEN

برای اینکه دستورات رو بهتر متوجه بشین، مثالها رو با همون جدول inventory که در اولین آموزش گفتم پیگیری میکنم.

فرض کنید شما میخواین تمام مقادیر بین دو مقدار مشخص رو داشته باشین. مثلا مقادیری که وزن اونها از ۱۵ بیشتر و از ۷۵ کمتر باشه. با توجه به آموزشهایی که تا حالا گفته شده با استفاده از where و عملگر and میشه این کار رو به صورت زیر انجام داد:

```
select * from inventory where weight>15 and weight<75
```

اما یه راه دیگه هم هست. میتونین به جای نوشتن شرط های مختلف از between استفاده کنید:

```
with ADOQuery1 do begin
  close;
  sql.Clear;
  sql.add('select * from inventory where weight between 15 and 75');
  open;
  end;
```

کد بالا مشخصات تمام کالاهایی رو برمیگردونه که وزن اونها بین ۱۵ و ۷۵ هست.

: IN

بعضی وقتها پیش میاد که چند مقدار خاص از یه فیلد رو میخوایم. باید از ۵ یا ۶ شرط مختلف استفاده کرد؟ جواب منفیه. شما میتونین از گزینه in در دستور select استفاده کنین (فرض کنید دنبال کالاهایی هستیم که رنگ اونها قرمز، سفید، آبی و سبز هست) :

```
with ADOQuery1 do begin
  close;
  sql.Clear;
  sql.add('select * from inventory where color in
("قرمز","سفید","آبی","سبز");
  open;
  end;
```

نوی پست قبلی یادم رفت که در مورد in , between به توضیح بدم که چطور میشه مقدار مورد جستجو رو از کاربر دریافت کنن.

اگر قرار باشه کاربر از edit استفاده کنه و مقادیر مورد نظر رو وارد کنه، در اکثر دستورات sql مقادیر متنی حتما باید بین "" قرار بگیره ("متغیر") ولی برای مقادیر عددی نیازی به "" نیست.

مثال برای between زمانی که در فیلد عددی جستجو میکنیم:

```
with adoquery1 do begin
  close;
  sql.clear;
  sql.Add('select * from inventory where weight between '+Edit1.Text+' and
'+Edit2.Text);
  open;
  end;
```

مثال برای in زمانی که در فیلد متنی (رشته) جستجو میکنیم:

```
ADOQuery1.Close;
ADOQuery1.SQL.Clear;
ADOQuery1.SQL.Add('select * from inventory where name in
(''+Edit1.Text+'',''+Edit2.Text+'',''+Edit3.Text+'',''+Edit4.Text+'')');
ADOQuery1.Open;
```

توجه: در دستور in فرقی نداره که مقدار عددی باشه یا رشته، حتما از پرانتز استفاده کنید.

: AS

در حالت عادی وقتی که از query ها استفاده میشه title مربوط به dbgrid نام فیلدهایی هست که در دستورات sql استفاده شده و معمولا به زبان انگلیسی هست. پس تکلیف برنامه های ما که به زبان فارسی هست چیه؟ برای حل این مشکل میتونید از Alias استفاده کنید. شکل کلی دستور به صورت زیر هست:

```
SELECT column AS column_alias FROM table
```

حالا میخوايم که نتايج در dbgrid به صورتی نشون داده بشه که دو فیلد name,color انتخاب شده و title مربوط به اونها "نام و رنگ" باشه:

```
with ADOQuery1 do begin
  close;
  sql.Clear;
  sql.add('select name as نام, color as رنگ from inventory');
  open;
end;
```

حتی اگر نام جدول هم خیلی طولانی هست و شما رو کلافه کرده میتونید از Alias به همین صورت که توضیح داده شده برای کوتاه کردن نام جدول استفاده کنید و جدول رو با نام دلخواه در برنامه خودتون به کار بگیرید.

: DISTINCT

تا حالا پیش اومده که بخواین تمام مقادیر یک فیلد رو توی combobox یا یه جدول دیگه داشته باشین؟ اگر همچین موردی بوده حتما میدونید که تمام مقادیر تکراری موجود در فیلد، در نتیجه حاصل هست. ممکنه چندین مقدار تکراری در یک فیلد وجود داشته باشه و بی دلیل به لیست مورد نظر شما اضافه شده باشه. راه حل این مشکل استفاده از DISTINCT در SQL هست.

شکل کلی دستور به صورت زیر هست :

```
SELECT DISTINCT column FROM table
```

به طور کلی این مورد به شما کمک میکنه مقادیر منحصر به فرد یک فیلد رو داشته باشید.

فرض کنید لیست کالاهای موجود در جدول نمونه رو میخوایم تا بدونیم که چه کالاهایی موجود هست، مطمئنا نیازی به نام های تکراری نداریم:

```
with adoquery1 do begin  
close;  
sql.clear;  
sql.add('select distinct name from inventory');  
open;  
end;
```

: LIKE

یکی دیگه از دستوراتی که زمان جستجو خیلی کارآمد هست، دستور LIKE هست که به همراه where استفاده میشه. این دستور برای زمانی هست که ما به دنبال تمام مقادیری هستیم که با یک یا چند حرف مشخص شروع میشه و یا تمام میشه، و یا حتی این چند حرف میتونه جزئی از نتیجه باشه.

شکل کلی دستور:

```
SELECT column FROM table  
WHERE column LIKE pattern
```

در قسمت pattern مشخص میکنیم که نتیجه حاصل بر چه اساسی باشه.

به مثال توجه کنید:

```
with ADOQuery1 do begin  
close;  
sql.Clear;  
sql.Add('select * from inventory where name like '''+Edit6.Text+'%''');  
open;  
end;
```

فرض میکنیم که مقادیر فیلد name قطعات سخت افزاری هستند. در مثال بالا اگر در edit1 مقدار "مانیتور" وارد شه، تمام مقادیری که با "مانیتور" شروع شده مثل "مانیتور Samsung ، مانیتور LG ، مانیتور BenQ و ..." به نمایش در میاد.

با کمی خلاقیت اگر این کد رو در رویداد OnChange یک ادیت قرار بدیم با تایپ اولین حرف مثلا "م" تمام مقادیر "مانیتور، ماوس، مادر برد" و بقیه مقادیری که با "م" شروع میشه به نمایش در میاد.

اگر به محل کاربرد "%" دقت کنید، خودتون میتونید شرط هایی که با نیاز شما سازگارتر هستند بنویسید.

: JOIN

نکته ای که فراموش کردم در مورد select بگم اینه که select میتونه اطلاعات چند جدول رو با هم در یک query به شما نمایش بده. پس شما میتونید اطلاعاتی رو که میخواین از دو یا چند جدول انتخاب کنید.

این کار به JOIN معروف هست.

```
SELECT table_1.column1, table_1.column2, table_2.column  
FROM table_1, table_2  
WHERE condition
```

کاربرد join در جداول master/detail بیشتر هست که جدولها بر اساس یک فیلد کلید با هم ارتباط دارند. فرض کنید در همین جدول inventory فیلد no مربوط به کد کالا، کلید هست. و جدول دیگه ای هم وجود داره به نام جدول orders که سفارش اشخاص بر اساس کد کالا در اون ذخیره میشه (پس فیلد no که در inventory کلید اصلی هست، در orders به عنوان کلید خارجی استفاده میشه) فیلدهای جدول orders میتونه "person_id (کلید اصلی) - fname - lname - no" باشه.

حالا ما میخوایم ببینیم چه افرادی چه کالاهایی رو سفارش دادند:

```
with adoquery1 do begin  
close;  
sql.Clear;  
sql.Add('select orders.fname, orders.lname, inventory.name from orders,  
inventory where inventory.no=orders.no');  
open;  
end;
```

نتیجه حاصل شامل نام و نام خانوادگی اشخاص از جدول orders و نام کالا از جدول inventory هست.

: LEFT JOIN – RIGHT JOIN - INNER JOIN

JOIN در sql موارد INNER JOIN , LEFT JOIN , RIGHT JOIN رو داره که توضیح میدم:

شکل کلی دستور در: INNER JOIN

```
SELECT field1, field2, field3
FROM first_table
INNER JOIN second_table
ON first_table.keyfield = second_table.foreign_keyfield
```

شکل کلی دستور در LEFT JOIN :

```
SELECT field1, field2, field3
FROM first_table
LEFT JOIN second_table
ON first_table.keyfield = second_table.foreign_keyfield
```

شکل کلی دستور در RIGHT JOIN :

```
SELECT field1, field2, field3
FROM first_table
RIGHT JOIN second_table
ON first_table.keyfield = second_table.foreign_keyfield
```

مثال برای: Inner Join

```
with adoquery1 do begin
close;
sql.Clear;
sql.Add('select inventory.name, inventory."cost", orders.fname,
orders.lname');
sql.Add('from inventory');
sql.Add('inner join orders');
sql.Add('on inventory.no=orders.no');
open;
end;
```

تفاوت: inner , left , right

در Inner join تمام رکوردهای دو جدول، هر جا که شرط مطابقت داشته باشه نمایش داده میشه و اگر در جدول اول رکوردی وجود داشته باشه که با هیچ رکوردی از جدول دوم مطابقت نداشته باشه، در نتیجه نهایی نمایش داده نمیشه.

در Left join تمام رکوردها از جدول اول نمایش داده میشه، حتی اگر با هیچ رکوردی از جدول دوم مطابقت نداشته باشه.

Right join برعکس left join عمل میکنه و تمام رکوردهای جدول دوم رو نمایش میده، حتی اگر با هم مطابقت نداشته باشند.

با روش آزمون و خطا و مثالهای مختلف میتونین دقیقا متوجه شین که کار این دستورات چی هست. زمانی که از inner join استفاده میشه تمام رکوردهایی لیست میشه که با شرط ما مطابقت داشته باشه. یعنی در جدول orders کالایی سفارش داده شده که در جدول inventory موجود هست. اگر از left join استفاده کنیم علاوه بر لیست شدن رکوردهایی که با شرط مطابقت دارند، بقیه رکوردهای جدول اول هم در نتیجه نهایی هست. یعنی کالاهایی که کسی اونها رو سفارش نداده.

و در صورتی که از right join استفاده کنیم، باز هم علاوه بر لیست شدن رکوردهایی که با شرط مطابقت دارند، بقیه رکوردهای جدول دوم هم در نتیجه نهایی هست. یعنی سفارشاتی که کالای مورد سفارش در جدول inventory معرفی نشده.

توضیح: مثال این قسمت توسط دوست عزیز [DrXoX](#) آماده شده:

برای درک بهتر مطلب شکل زیر فکر کنم خوب باشه

Table1

ID	Name
1	علی
2	رضا
3	محمد

Table2

ID	City
1	تهران
2	مشهد
4	اصفهان

Select Name, City From Table1 **Inner join** Table2 On Table1.ID = Table2.ID

Inner Join

Name	City
علی	تهران
رضا	مشهد

Select Name, City From Table1 **Left join** Table2 On Table1.ID = Table2.ID

Left Join

Name	City
علی	تهران
رضا	مشهد
محمد	Null

Select Name, City From Table1 **Right join** Table 2 On Table1.ID = Table2.ID

Right join

Name	City
علی	تهران
رضا	مشهد
Null	اصفهان

: IS NULL

یکی دیگه از توابعی که میتونه مفید باشه IS NULL هست.
این تابع زمانی کاربرد داره که شما میخوای تمام فیلدهای null (خالی) رو پیدا کنی.
شکل کلی دستور:

```
SELECT Column FROM table WHERE column IS NULL
```

کد زیر تمام رکوردهایی رو برمیگردونه که فیلد name اونها خالی هست:

```
with adoquery1 do begin
close;
sql.Clear;
sql.Add('select * from inventory where name is null');
open;
end;
```

توجه کنید که null با (" ") space تفاوت داره.

منظور از " " ، کاراکتر فضای خالی هست که با فشردن کلید space کیبورد ایجاد میشه .یعنی اگر فیلدی در ظاهر خالی باشه ولی این کاراکتر در اون وجود داشته باشه، به عنوان null شناخته نمیشه.
Null مربوط به زمانی هست که هیچ کاراکتری یا داده ای در فیلد وجود نداره.

اگر خواستید رکوردهای پر شده رو پیدا کنید، دستور رو به صورت زیر تغییر بدید:

```
select * from table where FieldName is not null
```


استفاده از محاسبات ریاضی:

در حالت عادی و وقتی که از table ها استفاده میشه، برای محاسبات روی تمام مقادیر به فیلد، میومدیم و به فیلد جدید از نوع calculate درست میکردیم و در OnCalcFields مربوط به جدول چند خط کد مینوشتیم. اینکار با استفاده از دستورات sql فوق العاده راحت هست.

شکل کلی دستور:

```
SELECT field1,field2,field3 <arithmetic operation> AS AliasName FROM table
```

به جای قسمت <arithmetic operation> به عبارت محاسباتی قرار میگیره.

به مثال توجه کنید (تمام مقادیر فیلد cost با ۲۰۰۰ جمع میشه و در ستون new_cost قرار میگیره) :

```
with adoquery1 do begin
close;
sql.Clear;
sql.Add('select name,cost+2000 as new_cost from inventory');
open;
end;
```

: Scalar function و Aggregate Functions

Sql به سری تابع خیلی مفید برای انجام عملیات روی فیلدها داره که میتونه خیلی مفید باشه. این توابع در دو دسته کلی قرار گرفته Aggregate Functions و Scalar functions :
توابع Aggregate :

SUM(column)

مجموع مقادیر فیلد

AVG(column)

میانگین یک فیلد

MAX(column)

بیشترین مقدار فیلد

MIN(column)

کمترین مقدار فیلد

COUNT(column)

(تعداد سطرهای یک فیلد (بدون محاسبه مقادیر Null))

COUNT(*)

تعداد کل رکوردهای جدول

FIRST(column)

اولین مقدار فیلد

LAST(column)

آخرین مقدار فیلد

توابع Scalar :

برعکس توابع Aggregate که بر روی تمام مقادیر فیلد عمل میکنند، این توابع مقادیر رو تک تک پردازش میکنند و نتیجه رو نمایش میدهند.

```
UCASE (column)
```

تبدیل کاراکترهای فیلد به حروف بزرگ (برای کاراکترهای a تا z)

```
LCASE (column)
```

تبدیل کاراکترهای فیلد به حروف کوچک (برای کاراکترهای A تا Z)

```
MID (column, start, end)
```

از محل start به میزان end کاراکتر از فیلد جدا کرده و نمایش میدهد (start و end اعداد صحیح هستند)

```
LEN (column)
```

مشخص کردن طول رشته های فیلد

```
INSTR (column, "ch")
```

نمایش محل وقوع کاراکتر ch در فیلد

```
LEFT (column, number)
```

جدا کردن تعداد کاراکتر مشخص، از سمت چپ مقادیر فیلد

```
RIGHT (column, number)
```

جدا کردن تعداد کاراکتر مشخص، از سمت راست مقادیر فیلد

```
ROUND (column, decimals)
```

رند کردن اعداد اعشاری و مشخص کردن دقت اعشار

```
NOW ()
```

نمایش تاریخ و زمان

روش استفاده:


شکل کلی دستور:

```
SELECT function(column) FROM table
```

یک مثال برای حاصل جمع فیلد:

```
with adoquery1 do begin
close;
sql.Clear;
sql.Add('select sum(cost) from inventory');
open;
```

سوال:

نوشته شده توسط **مرد مباح**  فرهاد جان.
 خروجی تابع بالا قاعدتا یک عدد هست. درسته؟
 اونوقت چطوری باید از اون توی برنامه استفاده کرد؟ اگه میشه یک توضیح کوچیک به ما بده.
 ممنون.

پاسخ: بله درسته. خروجی توابعی مثل max,min,first,count,avg,... یک عدد هست.

اگر در به کار بردن این توابع از نام مستعار استفاده کنید (استفاده از AS) به صورت زیر میتونید مقداری که query برمیگردونه رو داشته باشید (با table ها فرقی نداره):

```
select sum(cost) as new_cost from inventory
```

روش دسترسی:

```
ADOQuery1.FieldValues['new_cost'];
```

ولی اگر نام مستعار به کار نبرید، با استفاده از شماره فیلد که از ۰ شماره گذاری میشه، میتونید به مقدار فیلد دسترسی داشته باشید:

```
select sum(cost) from inventory
```

روش دسترسی:

```
ADOQuery1.Fields[0].Value;
```

سوال:

نوشته شده توسط hp1361  سلام

دوستان هر رکورد من ۱۰ تا ستون داره اسم ستونها رو باید در قسمت Sum بنویسم؟
اگه بخوام هر رکورد جمع فیلدهاش در یک فیلد دیگه در همون رکورد (فیلد جدید باشه
ها) قرار بگیره چیکار کنم؟
ممنون و منتظرم

پاسخ توسط: MasterData

شما میخواهید ستون های یک رکورد رو جمع بزنی یا یک ستون از همه رکوردها؟
اگه میخواهید مجموع ستونهای یک رکورد رو به عنوان یک ستون جدید داشته اشی

```
Select field1+field2+....+fieldN as NewField from TableName
```


اما اگه میخواهید یک ستون رو از چند رکورد جمع بزنی

```
Select Sum(FieldName) As NewField From TableName
```

ممکنه شرط یا گروه بندی داشته باشی

توجه: توضیحات کامل در مورد Group by در ادامه آمده است.

نقل قول:

نوشته شده توسط MasterData  شما میخواهید ستون های یک رکورد رو جمع بزنی یا یک ستون از همه رکوردها؟
تگه میخواهید مجموع ستونهای یک رکورد رو به عنوان یک ستون جدید داشته اشی
کد:

```
Select field1+field2+....+fieldN as NewField from TableName
```

اما اگه میخواهید یک ستون رو از چند رکورد جمع بزنی
کد:

```
Select Sum(FieldName) As NewField From TableName
```

ممکنه شرط یا گروه بندی داشته باشی

کاملاً درسته. فقط این نکته رو اضافه میکنم که اگر میخواهید یک ستون رو جمع بزنی و بر اساس مقدار یکتا در فیلد دیگه نشون بدی، باید از group by استفاده کنی. نام فیلدی که در قسمت group by نوشته میشه، باید حتماً در جلوی select هم اومده باشه.

```
with adoquery1 do begin
close;
sql.Clear;
sql.Add('select field1,sum(field2) from table ');
sql.Add('group by field1');
open;
```

: UNION

توی این پست میخوام در مورد UNION توضیح بدم. این دستور برای انتخاب اطلاعات مربوط به هم از دو یا چند جدول هست، مثل Join .

شکل کلی دستور:

```
SELECT field FROM table1
UNION
SELECT field FROM table2
```

-تعداد فیلدهایی که در دستورها select قرار میگیره باید برابر باشه
-نوع داده فیلدهایی که زیر هم قرار میگیرن باید یکی باشه

این دستور اطلاعاتی رو که از دو جدول بدست میاد به صورت یکتا (Distinct) و sort شده در یک ستون قرار میده. اگر نیاز به انتخاب مقادیر تکراری باشه باید از UNION ALL استفاده کرد. شکل کلی دستور همونی هست که گفته شد، فقط به جای union all از union استفاده میشه و البته دیگه sort نمیشه.

: GROUP BY

یکی از مواردی که کاربرد زیادی هم در sql داره گروه بندی داده ها هست.

زمانی که از توابع aggregate استفاده میکنید، تمام داده های جدول رو به عنوان نتیجه بر میگردونه. با استفاده از group by میتونید بر اساس یک مقدار یکتا در فیلد، داده ها رو گروه بندی کنید.

شکل کلی دستور:

```
SELECT column, aggregate function(column) FROM table GROUP BY column
```

فرض کنید در جدول inventory ارزش تمام کالاها رو به تفکیک هر کالا میخوانین:

```
select name,sum(cost) from inventory group by name
```

در کد بالا اگر در جدول چند کالا با یک نام وجود داشته باشه، query که نوشته شده مجموع فیلد cost مربوط به کالاهای همنام رو به عنوان نتیجه برمیگردونه.

نکته: فیلدی که در جلوی group by نوشته میشه باید در قسمت select هم باشه.

: HAVING

زمانی که داده ها رو گروه بندی میکنید، اگر نیاز به دستورات شرطی باشه همیشه از where استفاده کرد و باید از عبارت having استفاده کنید.

شکل کلی دستور:

```
SELECT column, aggregate function (column) FROM table
GROUP BY column
HAVING aggregate function (column) condition value
```

فرض کنید در جدول inventory ارزش تمام کالاها رو به تفکیک هر کالا و کالاهایی که ارزش کلی آنها بیشتر از ۲۵۰۰۰ هست:

```
select name,sum(cost) from inventory group by name having sum(cost)>=25000
```


امیدوارم تا اینجا خسته نشده باشید. دستورات مربوط به بازیابی داده ها تقریبا تمام شده، از امروز میخوام دستوراتی رو بگم که مربوط به ویرایش و اعمال تغییرات در دیتابیس و جداول هست. پس برای این دستورات دیگه از adoquery استفاده نمیکنم و به جای اون از ADOCommand استفاده میشه که در تب ADO هست.

: UPDATE

یکی از کارهایی که همیشه در دیتابیسها انجام میشه ویرایش داده هایی هست که قبلا وارد شده، برای این کار از دستور UPDATE استفاده میشه.

همینطور که از معنی این کلمه مشخص هست، این دستور برای ویرایش و اصلاح داده ها در جدول ها استفاده میشه.

شکل کلی دستور:

```
UPDATE table_name
SET column_name = new_value
WHERE column_name = some_value
```

در مقابل set مقدار جدید new_value برای فیلد column_name نوشته میشه.

فرض کنید در جدول inventory میخوایم مقدار فیلد cost رو برای فیلد name تغییر بدیم با این شرط که هر جا name=sample هست cost=new_cost بشه.

```
ADOCommand1.CommandText:='update inventory set cost=new_cost where
name=sample';
ADOCommand1.Execute;
```

: DELETE

تا حالا پیش اومده بخواید چند سطر رو با هم از یک جدول حذف کنید؟ حتما برای این کار به حلقه مینوشتید و... با استفاده از delete در sql میتونید خیلی راحت هر تعداد سطر رو از جدول با شرایط خاص حذف کنید.

شکل کلی دستور:

```
DELETE FROM table_name
WHERE column_name = some_value
```

: INSERT INTO

برای درج داده های جدید در بانک استفاده میشه.

شکل کلی دستور:

```
1:
INSERT INTO table_name
VALUES (value1, value2,....)

2:
INSERT INTO table_name (column1, column2,...)
VALUES (value1, value2,....)
```

روش اول برای زمانی هست که قراره تمام فیلدها مقدار دهی شه و لی دومی زمانی کاربرد داره که فیلدهای خاصی رو وارد میکنید.

فرض کنید قرار هست در جدول inventory تمام فیلدها رو مقدار دهی کنید:

```
ADOCCommand1.CommandText='insert into inventory values
('+edit1.Text+', '"+edit2.Text+', '"+edit3.Text+', '"+edit4.Text+', '"+edit5.Te
xt+')';
ADOCCommand1.Execute;
```

میبینید که مقدار دهی ۵ فیلد در ۲ خط انجام شد!
نکته مهم: مقادیر فیلدهای متنی (text) حتما باید بین " " داشته باشند.

: SELECT INTO

این دستور اغلب برای Backup گرفتن از جدول ها مورد استفاده قرار میگیره. دقت کنید که شما با کامپوننت ها هم میتونید backup بگیرید اما از تمام دیتابیس. پس زمانی که نیازی به backup تمام دیتابیس نیست بهتره که از select into استفاده کنید.

شکل کلی دستور:

```
SELECT column_name(s) INTO newtable [IN externaldatabase]
FROM source
```

این دستور فوق العاده انعطاف پذیر هست، چون هم میتونید پشتیبان جدولها رو در همون دیتابیس ذخیره کنید و هم میتونید به یک دیتابیس دیگه انتقال بدید.
حالت اول:

```
SELECT * INTO table_backup FROM table
```

حالت دوم:

```
SELECT * INTO table_backup IN backup.mdb FROM table
```

حالتهای دیگر: انتخاب چند فیلد مشخص به جای استفاده از * ، به کار بردن عبارات شرطی مانند where و گرفتن پشتیبان از وضعیت جاری، پشتیبان گیری از چند جدول درون یک جدول.

استفاده از کاراکتر های خاص برای دستورات شرطی:

قبلا به همراه LIKE کاراکتر % رو به کار برده بودم. این کاراکتر تمام فیلدهایی رو که با شرط دستور مطابقت داشتند و دارای طول مساوی و یا بیشتر از رشته بودند رو بر می گردوند. امروز میخوام در مورد کاراکتر _ و [] توضیح بدم. فرض کنید به دنبال فیلد هایی هستیم که سه حرفی هستند و دو حرف اول اونها AB هست. اگر شرط رو به صورت

```
LIKE AB_
```

بنویسیم تمام فیلدهای ... abc, aba, abr, abz, در نتیجه حاصل هست. حالا اگر از این که فیلد با چه کاراکتری شروع شده اطلاعی نداشته باشیم و مجموعه ای از کاراکترهای فیلدها رو بخوایم چي؟ از [] استفاده میکنیم
مثال:

```
LIKE [a-g]%
```

تمام فیلدها رو که با حروف a,b,c,d,e,f,g شروع میشوند رو در نتیجه حاصل بر می گردونه.

: ALTER TABLE

این دستور یکی از دستورات DDL زبان T-SQL می باشد که به عنوان نمونه جهت اضافه نمودن فیلد به جدول توضیح داده می شود.
اضافه کردن فیلد در زمان اجرا:
برای این کار باید از ALTER TABLE استفاده کنیم:
شکل کلی دستور:

```
ALTER TABLE table_name  
ADD column_name datatype
```

نام جدول : table_name
نام فیلد : column_name
نوع فیلد : datatype

فرض کنید میخواهیم فیلد MyField رو با طول ۵۰ کاراکتر به جدول inventory اضافه کنیم:

```
alter table inventory add MyField varchar(50)
```

دستورات مربوط به ذخیره و بازیابی داده ها (DML) تقریباً به صورت کامل گفته شده و فقط دستورات ساختن اشیا در پایگاه داده و تنظیم ویژگی ها و خصوصیات پایگاه داده (DDL) باقی مانده است که مطمئناً گفتن تمام آنها ضرورتی ندارد و اگر دوستان به این موارد نیاز پیدا کردند می توانند مشکلات خود را در لینک هایی که در صفحه ۲ مقاله ذکر شده پیگیری نمایند.

مطمئناً آموزش ها خالی از اشکال نیست. از تمامی دوستان عزیز درخواست می کنم مشکلات موجود را به بنده اعلام کنند تا در بهبود سطح مقاله سهیم باشند.

موفق و پیروز باشید

[فرهاد دهداران](#)

پایان