

آموزش سطح مقدماتی SQL Server 2008

نویسنده : علیرضا بهارلویی

سایت علمی دانشجویان ایران

www.Daneshju.ir



اصطلاح پایگاه داده یکی از اصلاحات رایج در شاخه انفورماتیک است.

امروزه بحث پایگاه داده را در بسیاری از کتابها و مقالات میتوان دید ولی کار در محیط عملی و یادگیری مفاهیمی به جز مفاهیم پایه مهمترین مطلبی هست که در بازار کار مورد استفاده قرار می گیرد و اصولا کمتر در کتاب ها به آن پرداخته شده یا به صورت غیر ملموس می باشد.

ما در این آموزش سعی می کنیم محیط عملیاتی پایگاه داده را مورد بحث قرار داده و مفاهیم ابتدایی آن را برای یادگیری مراحل پیشرفته بازگو کنیم. اما این نکته را نمی شود نادیده گرفت که مفاهیم بنیادی اگرچه پیش پا افتاده است ولی برای شروع کار با محیط عملیاتی باید نیم نگاهی به آنها داشته باشیم برای همین منظور، در ابتدا این مفاهیم بنیادی را مطرح می کنیم.

سیستم مدیریت پایگاه داده یکی از سیستم های ذخیره و بازیابی اطلاعات است. چرا ذخیره و بازیابی اطلاعات؟

در محیط ذخیره و بازیابی اطلاعات، ما با امکاناتی چون ذخیره داده ها، بازیابی داده ها یا تولید داده جدید سر و کار داریم. در محیط عملیاتی پایگاه داده ها هم همین عملیات ها را داریم پس دلیلی که باعث شده پایگاه داده یکی از سیستم های ذخیره باشد این است.

ولی نکته اساسی در تفاوت سیستم ذخیره و بازیابی با پایگاه داده در مورد نوع همین پردازش و ذخیره و بازیابی است. در سیستم های پایگاه داده ما با داده های پیچیده سر و کار داریم به همین خاطر باید فراتر از ذخیره و بازیابی عمل کنیم.

تاریخچه ای از پایگاه داده:

پایگاه داده از اوایل دهه ۶۰ میلادی ایجاد شد، در دهه ۷۰ توسعه یافت و در دهه ۸۰ سیستم مدیریت پایگاه داده های رابط وارد شد و مورد توجه قرار گرفت. در دهه ۹۰ میلادی طرح کامل این پایگاه، سیستم مدیریت پایگاه شی گرا وارد دنیا شد و مورد استقبال قرار گرفت و تا امروز پای برجای مانده است.

پایگاه داده شامل سیستم هایی چون سیستم فیلینگ، سیستم مدیریت داده ها، سیستم مدیریت پایگاه داده، سیستم پایگاه داده های فضایی، سیستم اطلاعات اجرایی و ... می باشد.

تعریف کامل پایگاه داده:

پایگاه داده عبارت است از مجموعه ای از فقره داده نامدار و پایا.

سیستم پایگاه داده شامل سخت افزار، نرم افزار، کاربر و داده می باشد که ما از شرح آنها به خاطر نا کارآمدی پرهیز می کنیم.

برای طراحی یک جدول و برقراری ارتباط واقعی بین اجزاء، نیاز به نمودارهای مدلسازی داریم که در پایگاه داده می توان از نمودارهای ERD، EER و UML استفاده کرد. چون این نمودارها مربوط به بحث تحلیل و طراحی سیستم است که از توضیح آنها در اینجا پرهیز می کنیم. اگر می خواهید پایگاه داده دقیقی داشته باشید باید قدرت طراحی نمودارهای فوق را داشته باشید. برای این منظور می توانید به منابع معتبر برای این مطالب مراجعه کنید. برای طراحی جدول پایگاه داده، نیاز به صفات داریم. صفات در اصل همان اشیاء دنیای واقعی ما هستند، مانند شماره ملی، شماره دانشجویی و ...

اکنون زمان آن رسیده تا وارد محیط عملیاتی پایگاه داده شویم.

محیطی که برای آموزش در نظر گرفته ایم، محیط SQL 2008 می باشد. این نسخه قابلیت پردازش زبان فارسی را هم دارد. همچنین سرعت پردازش و بازیابی بالاتری نسبت به نسخه های قبلی خود دارد.

ابتدا برخی از مقادیری که در این محیط میتوان به کار برد را در زیر نام می بریم.

مقادیر قابل قبول انواع داده

مقادیر عددی

مقادیر اعداد صحیح bight

مقادیر اعداد صحیح Int

مقادیر اعداد صحیح Smallint

مقادیر اعداد صحیح Tinyint

مقادیر اعداد صحیح با ارزش ۰ و ۱ bit

مقادیر مقیاس با دقت ثابت شده decimal

مقادیر صحیح شناور (متغیر Float)

مقادیر صحیح شناور متغیر واقعی real

مقادیر تاریخ و زمان datetime

مقادیر کاراکتری

مقادیر حرفی کدگذاری نشده با طول ثابت Char

مقادیر حرفی کدگذاری نشده با طول متغیر Varchar

داده کدگذاری شده با طول متغیر Text

داده کدگذاری شده با طول ثابت Nchar

داده کدگذاری نشده با طول متغیر nvarchar

داده کدگذاری شده با طول متغیر ntext

مقادیر باینری (۱ و ۰)

داده باینری با طول ثابت binary

داده باینری با طول متغیر Varbinary

داده باینری با طول متغیر Image

حال قصد داریم که ساخت جدول، مقدار دهی جدول ها، واکنشی اطلاعات مورد نیاز و ... را با هم یاد بگیریم. با ما همراه باشید تا به مرور مقدمات این نرم افزار را آموزش ببینیم.

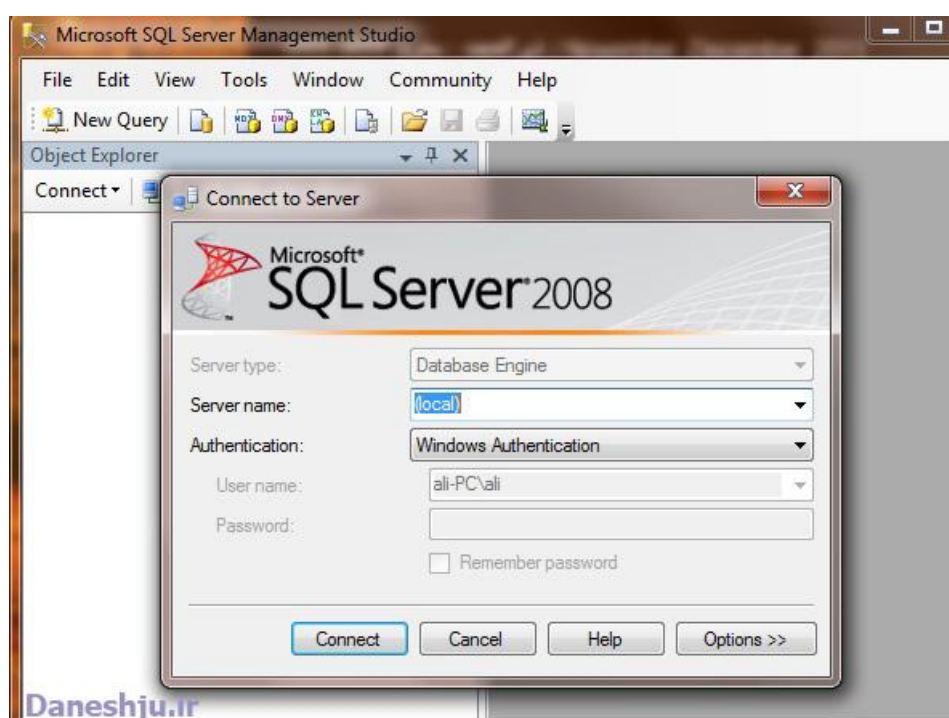
توجه داشته باشید مثالی که برای شرح آموزش استفاده می شود مثال دانشجو و دروس وی می باشد.

قسمت اول:

پس از نصب SQL Server 2008 صفحه ای باز می شود که از شما درخواست یک Server name را دارد. ما به طور پیش فرض آن را (local) قرار می دهیم.

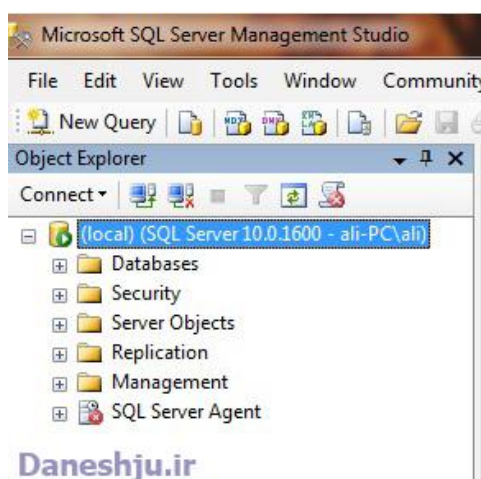
همانطور که مشاهده می کنید در قسمت User name به طور خودکار هر اسمی که روی سیستم عامل شما باشد قرار می گیرد. همچنین شما می توانید پسوردی برای خود قرار دهید.

حال گزینه Connect را انتخاب کرده تا وارد محیط اصلی شویم.

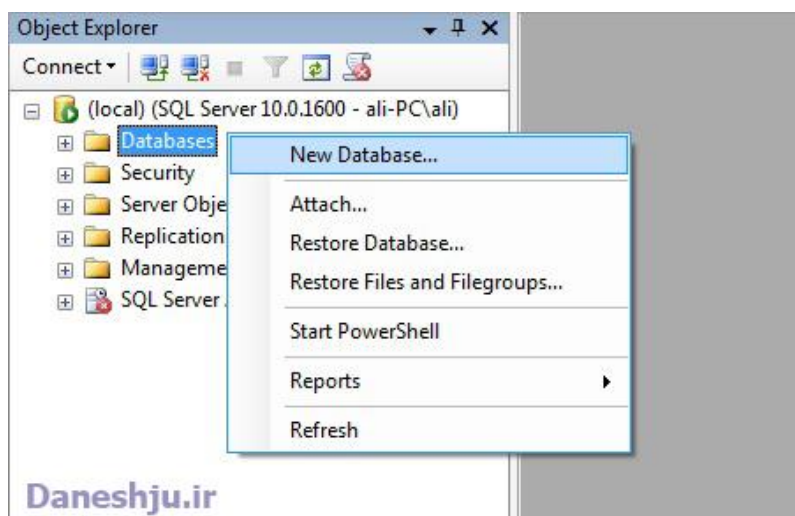


وقتی وارد محیط می شویم (همانطور که در شکل می بینید) وارد Server name که ساختیم شده ایم.

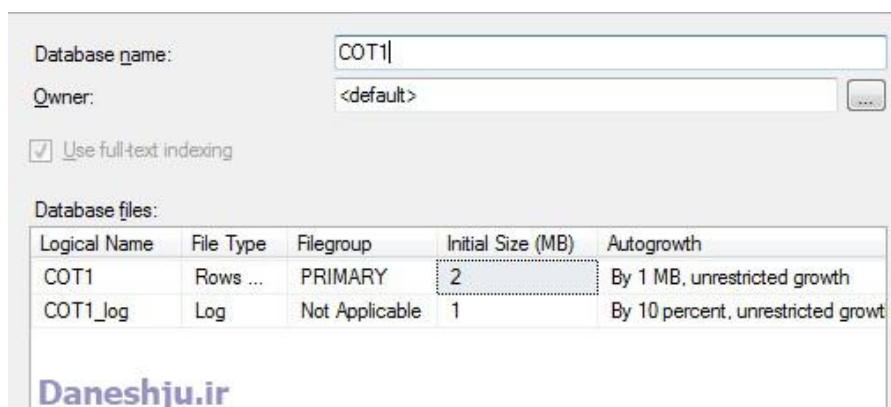
اکثر مباحث این آموزش روی قسمت Databases است.



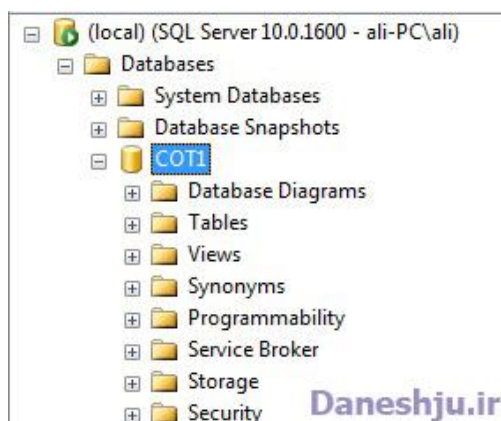
روی Databases کلیک راست کرده و گزینه New Database را انتخاب می کنیم. حال یک دیتابیس جدید تولید کرده ایم.



همانطور که در شکل می بینید سیستم از ما یک نام برای دیتابیس درخواست می کند. در اینجا اسم این دیتابیس را COT1 قرار می دهیم.



حال وارد دیتابیس خود می شویم. هر دیتابیس شامل زیرمجموعه هایی همچون tables, views و ... هستند که ما بیشتر روی همین دو قسمت کار می کنیم.



روی گزینه tables کلیک راست کرده و گزینه New tables را انتخاب می کنیم.


حال می توانیم جدول هایی برای داده های خود داشته باشیم. برای این مثال به سه جدول نیاز داریم. جدولی برای دانشجو، جدولی برای دروس و جدول سومی برای ترکیب این دو جدول.

همچنین باید در هر یک از جدول ها یک صفت شاخص یا همان کلید انتخاب کنیم. همانطور که می دانید کلید اصلی باید صفتی باشد که خاص هر داده باشد و موارد تکراری نداشته باشد.


برای انتخاب کلید اصلی روی هر صفتی که مدنظر است کلیک راست کرده و گزینه primary key را انتخاب کنید.

هر صفتی که به عنوان کلید اصلی انتخاب شود یک کلید کنار آن ظاهر می شود.

جدول زیر که به نام COT است. شامل ستون هایی برای شماره درس، عنوان درس، تعداد درس، نوع درس و شماره گروه آموزش ارائه کننده درس می باشد. COID به عنوان کلید اصلی انتخاب شده است.

	Column Name	Data Type	Allow Nulls
	COID	smallint	<input type="checkbox"/>
	COTITLE	nvarchar(50)	<input type="checkbox"/>
	CREDIT	smallint	<input type="checkbox"/>
	COTYPE	nvarchar(50)	<input checked="" type="checkbox"/>
	CODEID	smallint	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

جدول بعدی به نام STT که شامل ستون هایی برای شماره دانشجویی، نام دانشجو، سطح تحصیلات، رشته تحصیلی و شماره گروه آموزش دانشجو می باشد. STID به عنوان کلید اصلی انتخاب شده است.

	Column Name	Data Type	Allow Nulls
	STID	smallint	<input type="checkbox"/>
	STNAME	nvarchar(50)	<input type="checkbox"/>
	STLEV	nchar(10)	<input checked="" type="checkbox"/>
	STMJR	nchar(20)	<input checked="" type="checkbox"/>
	STDEID	smallint	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

جدول سوم به نام STCOT که ترکیب دانشجو و دروس است. این جدول با استفاده از کلیدهای اصلی دو جدول قبل ایجاد شده و شامل ستون هایی برای ترم، سال تحصیلی و نمره درسی دانشجو است.

Column Name	Data Type	Allow Nulls
STID	smallint	<input type="checkbox"/>
COID	smallint	<input type="checkbox"/>
TR	smallint	<input type="checkbox"/>
YRYR	smallint	<input type="checkbox"/>
GRADE	smallint	<input type="checkbox"/>
		<input type="checkbox"/>

Daneshju.ir

نکته ۱: در حین ساخت جدول ها همانطور که می بینید کنار هر سطر در انتها یک گزینه Allow Nulls هست. این گزینه اگر فعال شود به ما اجازه می دهد در هنگام وارد کردن اطلاعات آن صفت را خالی بگذاریم. مثلا می توان نوع مدرک یک دانشجویی را درج نکرد و در آینده آن را وارد کرد.

نکته ۲: توجه داشته باشید کلید اصلی نمیتواند nulls باشد.

نکته ۳: با استفاده از نوع داده nvarchar می توانیم مقادیر را فارسی وارد کنیم.

حال میبینید که در زیرمجموعه Tables سه جدول ایجاد شده وجود دارند.

Tables
System Tables
dbo.COT1
dbo.STCOT
dbo.STT

Daneshju.ir

با کلیک راست روی هر یک از جداول و انتخاب گزینه Edit Top 200 Rows می توانیم وارد جدول ها شویم و مقادیر مربوط به هر صفت را وارد کنیم.

ما در اینجا به طور فرضی مقادیری را وارد نموده ایم.

	STID	STNAME	STLEV	STMJR	STDEID
▶	8800	محسن	کارشناسی	مهندسی کامپیو...	1
	8801	نادر	کارشناسی	علوم کامپیوتر	1
	8802	مریم	کارشناسی	علوم کامپیوتر	1
	8803	حامد	کاردانی	مهندسی کامپیو...	1
	8804	مینا	کاردانی	مهندسی صنایع	1
	8805	مرضیه	کاردانی	مهندسی صنایع	1
	8806	علی	کارشناسی	مهندسی کامپیو...	1

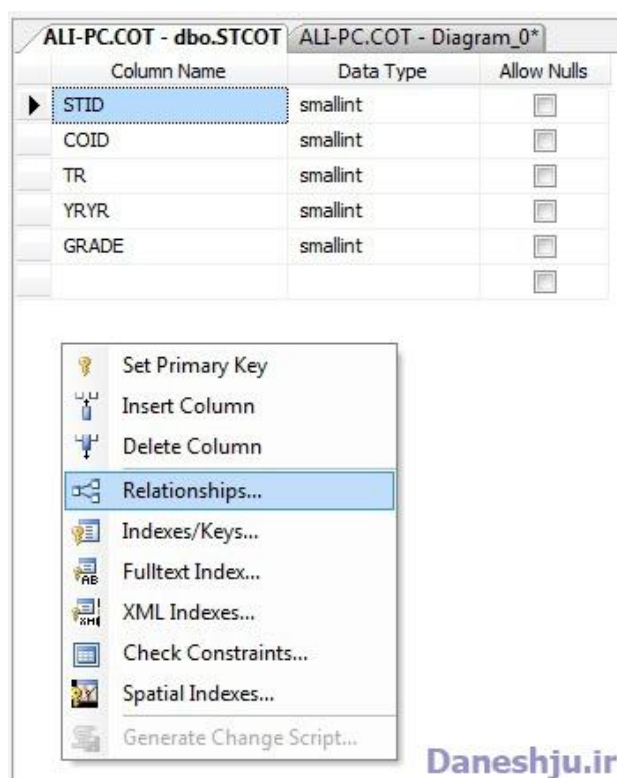
Daneshju.ir

	STID	COID	TR	YRZR	GRADE
▶	8800	1	1	9091	14
	8800	2	1	9091	10
	8800	4	2	9091	15
	8801	1	2	9091	8
	8801	6	1	9091	18
	8802	7	2	9091	14
	8802	5	1	9091	19
	8802	10	2	9091	17

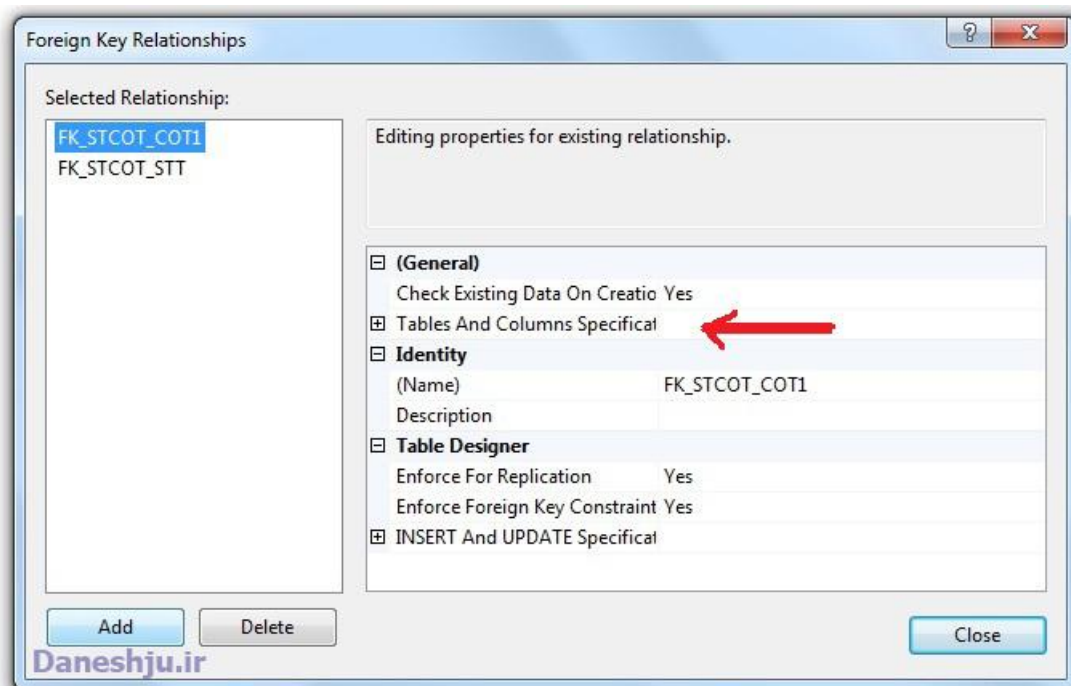
	COID	COTITLE	CREDIT	COTYPE	CODEID
▶	1	اصول طراحی پی	3	تخصصی	1
	2	شبکه های کامپیوتر	3	اصلی	1
	3	مهندسی ترم 1	3	تخصصی	1
	4	مهندسی ترم 2	3	تخصصی	1
	5	سیستم عامل	3	اصلی	1
	6	اندیشه 1	2	عمومی	1
	7	اندیشه 2	2	عمومی	1

در آخرین مرحله باید برای ترکیب جدولها و ایجاد رابطه بین جدول COT & STCOT و STT & STCOT اقدام نمود تا در حین بازیابی اطلاعات از جدول سوم مشکلی نداشته باشیم.

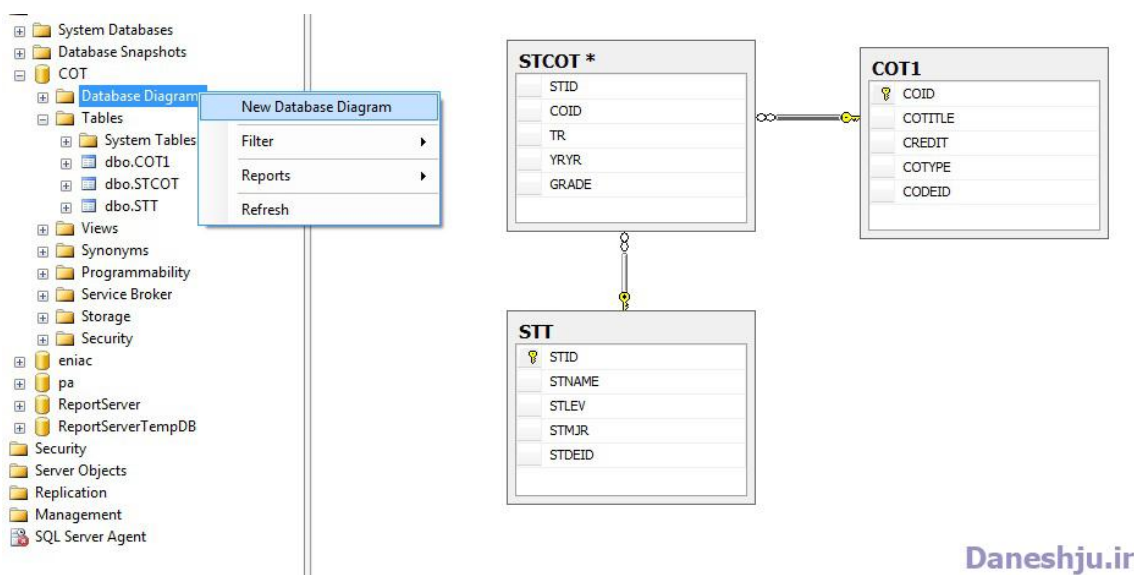
برای این کار باید به روی جدول COT رفته و گزینه Desing را انتخاب کنیم. در آنجا روی فضای خالی کلیک راست کرده و گزینه Relationship را انتخاب می کنیم.



همانطور که در شکل می بینید وقتی وارد محیط شدیم گزینه Tables and Columns را انتخاب کرده و در آنجا به ترتیب کلیدهای اصلی هر کدام از جدول ها را به جدول STCOT وصل می کنیم.



سپس روی گزینه Database Diagram کلیک کرده و New Database را انتخاب می کنیم تا ترکیب جدول ها را به صورت نموداری مشاهده کنیم.



حال ما سه جدول و مقادیر آنها را داریم و توانسته ایم آنها را به هم وصل کنیم تا در واکنشی اطلاعات مشکلی نداشته باشیم.

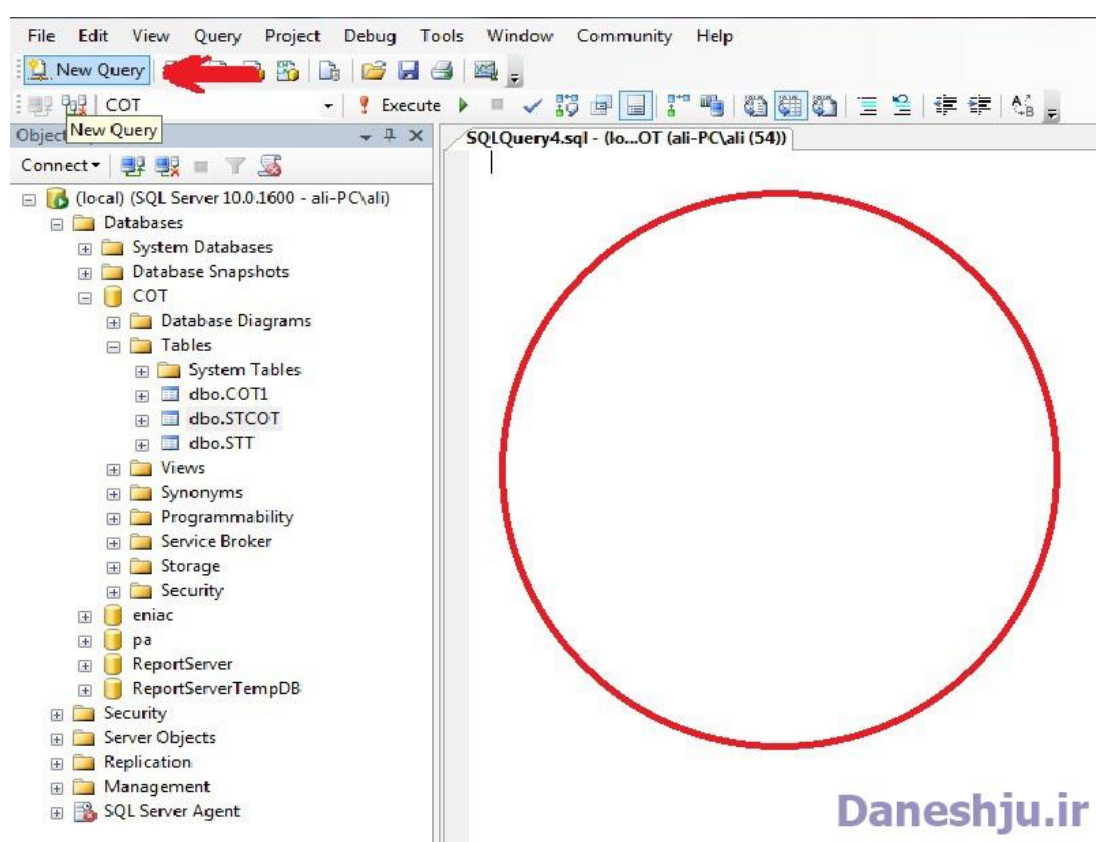
مرحله ساخت جدول به اتمام رسید. در قسمت بعد ما به سراغ پرسش و پاسخ ها خواهیم رفت.

قسمت دوم:

بخش دوم آموزش SQL 2008 را با مبحث پرس و جو در محیط آن پی می گیریم.

همانطور که مشاهده می کنید در قسمت بالای صفحه اصلی SQL، گزینه ای به نام NEW Query داریم. با انتخاب این گزینه وارد محیط پرس و جو خواهیم شد.

در محیط پرس و جو می توانیم با استفاده از کدهای دستور العمل از جدول هایی که ساختیم اطلاعاتی را به هر طریقی که می خواهیم به دست بیاوریم. مثلاً اعمالی نظیر میانگین نمرات یا دانشجویانی که در یک ترم، درسی را قبول شده اند، نمره قبولی آن درس و ... که در ادامه، این امکانات و پرس و جوها را خواهیم دید.



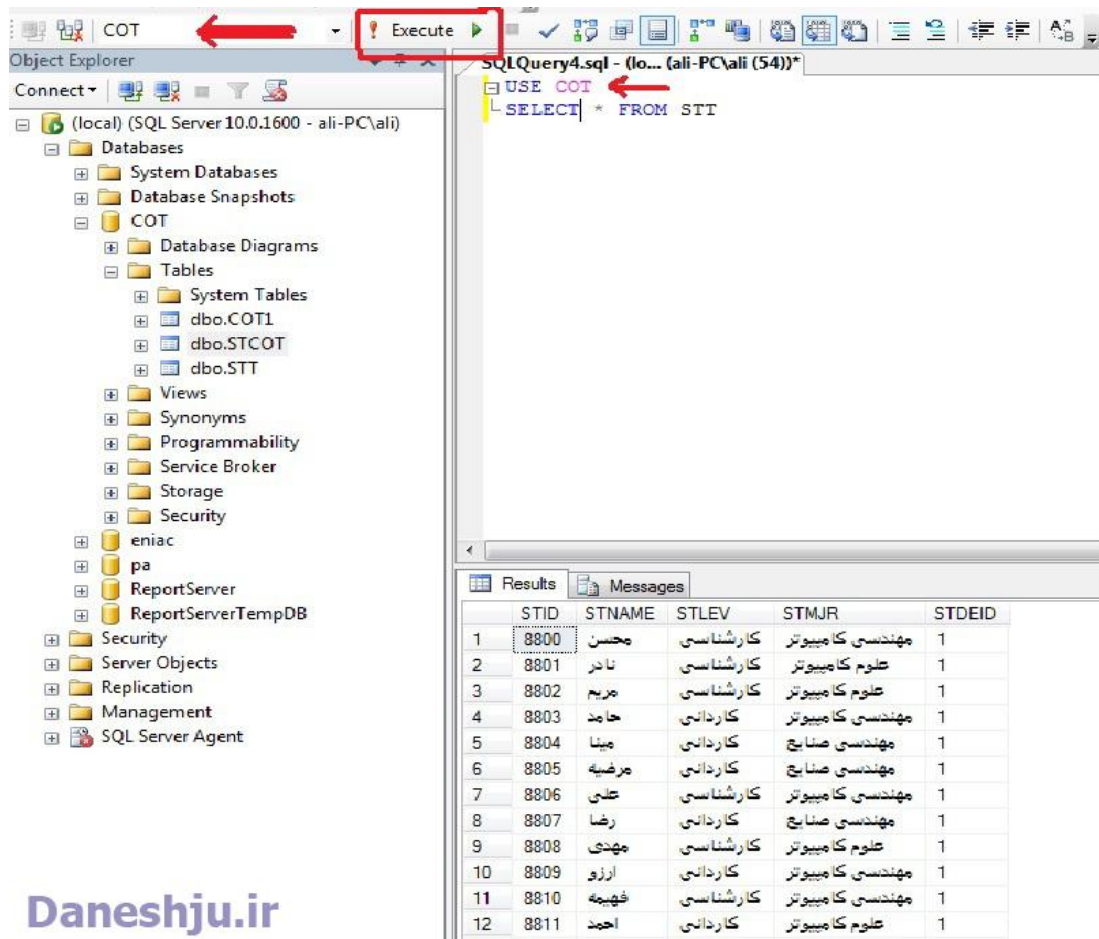
برای اینکه جدول مورد نظر خود را برای پرس و جو انتخاب کنیم باید در قسمت مشخص شده در عکس، جدول مورد نظر را انتخاب کنیم. در اینجا جدول COT که مربوط به مشخصات دروس و دانشجو و ... می باشد را انتخاب کردیم.

SELECT اولین دستوری هست که اینجا نوشته ایم. در این دستور، باید از FROM استفاده کنیم، زیرا این دستور مشخص می کند از کدام جدول باید واکشی انجام شود.

می توانیم FROM را با یک * به SELECT متصل نمود. البته در آینده خواهیم دید که نیازی به * هم نداریم.

نقش دستور USE مشخص کردن جدول اصلی هست که می خواهیم پرس و جوی داده هایمان را در آن داشته باشیم، که این کار را در بالا به روش دیگری مشخص کردیم و دیگر نیازی به آن نداریم.

در این مثال، داده های جدول STT را واکشی کردیم. همانطور که مشاهده می کنید کل داده های این جدول در پایین نمایش داده شده است. برای اجرای هر دستوری که نوشته می شود باید گزینه Execute را کلیک کنیم تا نتایج نمایش داده شوند.



The screenshot shows the SQL Server Enterprise Manager interface. In the top toolbar, the 'Execute' button (a green play icon) is highlighted with a red box and a red arrow. The Object Explorer on the left shows the database structure, including the 'COT' database and its tables: 'dbo.COT1', 'dbo.STCOT', and 'dbo.STT'. The SQL Query window on the right contains the following SQL code:

```
USE COT
SELECT * FROM STT
```

The Results pane at the bottom shows the output of the query, which is a table with 12 rows and 5 columns: STID, STNAME, STLEV, STMJR, and STDEID. The data is as follows:

STID	STNAME	STLEV	STMJR	STDEID
1	8800	مهندس	کارشناسی کامپیوتر	1
2	8801	نادر	کارشناسی علوم کامپیوتر	1
3	8802	مریم	کارشناسی علوم کامپیوتر	1
4	8803	حامد	کارشناسی کامپیوتر	1
5	8804	مینا	کارشناسی صنایع	1
6	8805	مرضیه	کارشناسی صنایع	1
7	8806	علی	کارشناسی کامپیوتر	1
8	8807	رضا	کارشناسی صنایع	1
9	8808	مهدی	کارشناسی علوم کامپیوتر	1
10	8809	ارزو	کارشناسی کامپیوتر	1
11	8810	فهیبه	کارشناسی کامپیوتر	1
12	8811	احمد	کارشناسی علوم کامپیوتر	1

برای بقیه داده های جدول های دیگر نیز این دستور را نوشته و اجرا می کنیم.



The screenshot shows the SQL Query window with the following three queries:

```
SELECT STNAME , STMJR
FROM STT

SELECT COTITLE , COTYPE
FROM COT1

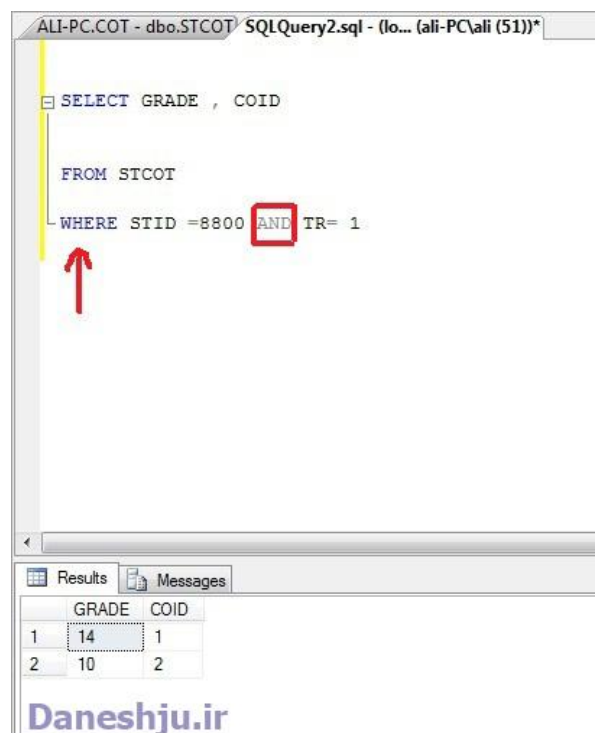
SELECT TR , YRYR
FROM STCOT
```

STNAME	STMJR	TR	YR	COTITLE	COTYPE
1	محسن	12	2	1	تخصصی
2	ناذر	13	1	2	اصلی
3	مریم	14	1	3	تخصصی
4	حامد	15	2	4	تخصصی
5	مینا	16	1	5	اصلی
6	مرغیبه	17	1	6	عمومی
7	علی	18	2	7	عمومی
8	رضا	19	1	8	اصلی

با استفاده از دستور های بالا، از هر جدولی دو صفت و مقادیرش را استخراج کرده ایم.

در دستور اول نام و رشته تحصیلی دانشجو را درخواست کرده ایم و در دستور بعدی نام درس و نوع آن و در دستور سوم سال تحصیلی و ترم آن را در خواست داده ایم که نتایج را در جدول های زیر آن مشاهده می کنید.

حال می خواهیم دستوری همراه با شرط نوشته و اجرا کنیم. برای قرار دادن شرط باید از WHERE استفاده کنیم. در اینجا یک دستور همراه با شرط را نوشته و اجرا می کنیم. مشاهده می کنید که در جلوی دستور شرط، مقدار ۸۸۰۰ که همان شماره دانشجویی و ۱ که ترم دانشجو است را نوشته ایم. اگر توجه داشته باشید می بینید که با استفاده از دستور AND صفت های مورد نظرمان را به هم وصل کرده ایم. با این کار می توانیم چندین صفت را درخواست کنیم. جواب این دستور هم در زیر آن مشاهده می کنید.



وقتی بخواهیم پس از مدتی یک داده را به جدول اضافه کنیم چون انجام این کار در هر جدول و به صورت اولیه وقت گیر و در بعضی مواقع کاری مشکل است، می توانیم در محیط پرس و جو از دستور INSERT استفاده کنیم. این دستور یک یا چند سطر را با صفت هایی که خودمان مشخص می کنیم به جدول یا جداول اضافه می کند.

پس از اجرای این دستور در قسمت پایین می توانیم مشاهده کنیم که آیا دستور با موفقیت اجرا شده یا خیر. در کادر قرمز رنگ مشاهده می کنید که پیام نشان دهنده موفقیت بوده است. همچنین در زیر آن، جدول را می بینید که سطر جدید به آن اضافه گردیده است.



Messages

(1 row(s) affected)

Daneshju.ir

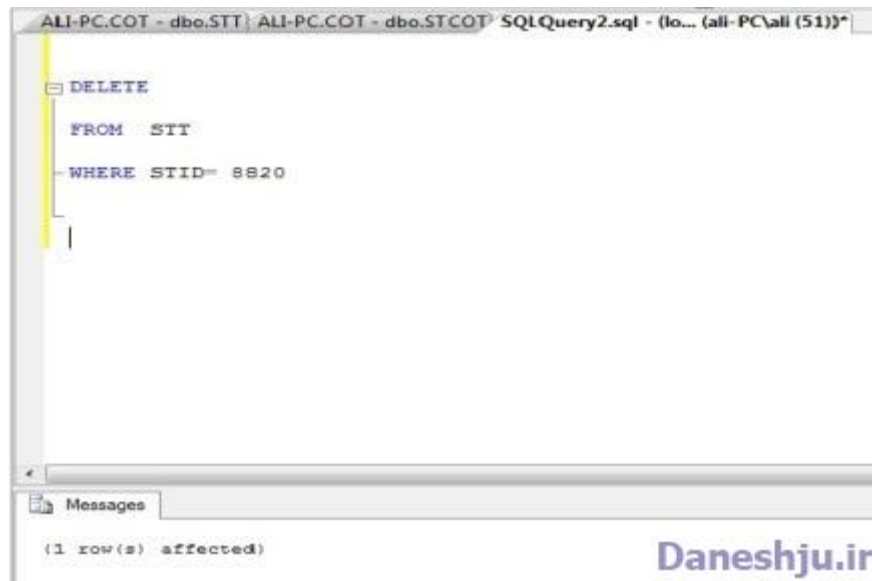
8810	فهمیه	کارشناسی	مهندسی کامپیو...	1
8811	احمد	کاردانی	علوم کامپیوتر	1
8820	ALI	BS	COMP	5
8822	رضا	کارشناسی	علوم کامپیوتر	1

همانطور که می توانیم سطری اضافه کنیم، در مقابل آن می توان سطری را حذف کرد. برای این کار با استفاده از دستور DELETE این کار را انجام می دهیم. برای درک بهتر ما این دو مثال را انجام می دهیم.

توجه داشته باشید که اگر یک صفت عام، مثل مدرک تحصیلی کاردانی اگر در شرط حذف قرار داده شود، تمام سطرهایی که این صفت را دارا می باشند حذف می گردند.



Daneshju.ir



در بعضی از مواقع مانند حذف یا درج و ... باید یک بهنگام سازی در جدول خود ایجاد کنیم تا داده ها به صورت صحیح نمایش داده شوند. برای این کار باید از دستور UPDATE استفاده کنیم.

در مواقع دیگری هم می توان از این دستور استفاده کرد. مثلا شماره درسی را در نظر بگیرید که پس از مدتی تغییر می کند. حال ما هم باید در جدول درس خود این کار را انجام داده تا با دنیای واقعی هماهنگ باشیم. برای این کار نیز از دستور UPDATE استفاده می کنیم.

برای درک این موضوع، شماره درسی را از جدول COT1 از ۶ به ۲۰ تغییر می دهیم.



نظم یکی مواردی است که به دقتی تر کار کردن و بهره بردن به ما کمک می کند. در جدول ها هم می توانیم نظم را اعمال کنیم. برای این امر، باید از دستوری به نام ORDER BY استفاده کنیم. توجه داشته باشید، می توانیم هم به صورت نزولی و هم به صورت صعودی، عمل نظم بخشیدن به داده ها را انجام دهیم. نرم افزار SQL به طور پیش فرض مرتب کردن صعودی را در نظر می گیرد. برای اینکه بتوانیم داده ها را به صورت نزولی مرتب کنیم، باید بعد از دستور ORDER BY از DESC استفاده کنیم.

مثالی از این مفاهیم را در شکل مشاهده می کنید.

The image shows two side-by-side screenshots of the SQL Server Enterprise Manager interface. Both windows show a query in the 'SQLQuery2.s' file, connected to the 'ALI-PC.COT - dbo.COT1' database. The query in both is: `SELECT GRADE FROM STCOT ORDER BY GRADE`. The left window's query is explicitly `ORDER BY GRADE DESC`, while the right window's query is `ORDER BY GRADE`. Below each query, the 'Results' tab is active, displaying a table with 12 rows and one column named 'GRADE'.

	GRADE
1	19
2	18
3	18
4	17
5	16
6	16
7	15
8	14
9	14
10	13
11	12
12	12

	GRADE
1	5
2	8
3	8
4	9
5	10
6	10
7	11
8	12
9	12
10	13
11	14
12	14

یکی از حسن هایی که در کار با داده ها مورد توجه است توابع داده ای هستند. توابعی همچون بیشترین مقدار، کمترین مقدار، میانگین و ... SQL هم از این توابع بی نصیب نبوده است. چند تا از این دستور های مهم و پرکاربرد را اینجا نام برده و مثال هایی می زنیم.

۱. دستور COUNT : این دستور به ما نشان می دهد که داده های مورد نظر چه تعداد هستند. در اصل این دستور، تابع شمارش است.

نکته ای که باید در این تابع در نظر بگیریم این است که اگر بخواهیم در شمارش های خود تکرار را هم داشته باشیم، باید پس از نوشتن دستور از یک (*) استفاده کنیم.

۲. دستورات MAX & MIN : این دو دستور، بیشترین و کمترین مقادیر را خواهند داد.

در اینجا یک مثال از این دستور ها را نمایش خواهیم داد. در دستور اول مقادیر کمتر و بیشتر و در دومی تعداد را محاسبه کرده ایم.

The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays two SQL queries. The first query is:

```
SELECT MIN (STID) , MAX (STID)
FROM STCOT
```

The second query is:

```
SELECT COUNT (*)
FROM STT
```

The bottom pane shows the results of these queries. The first query result is a table with two columns, both labeled '(No column name)', containing the values 8800 and 8811. The second query result is a table with one column labeled '(No column name)' containing the value 13.

(No column name)	(No column name)
1	8800
	8811

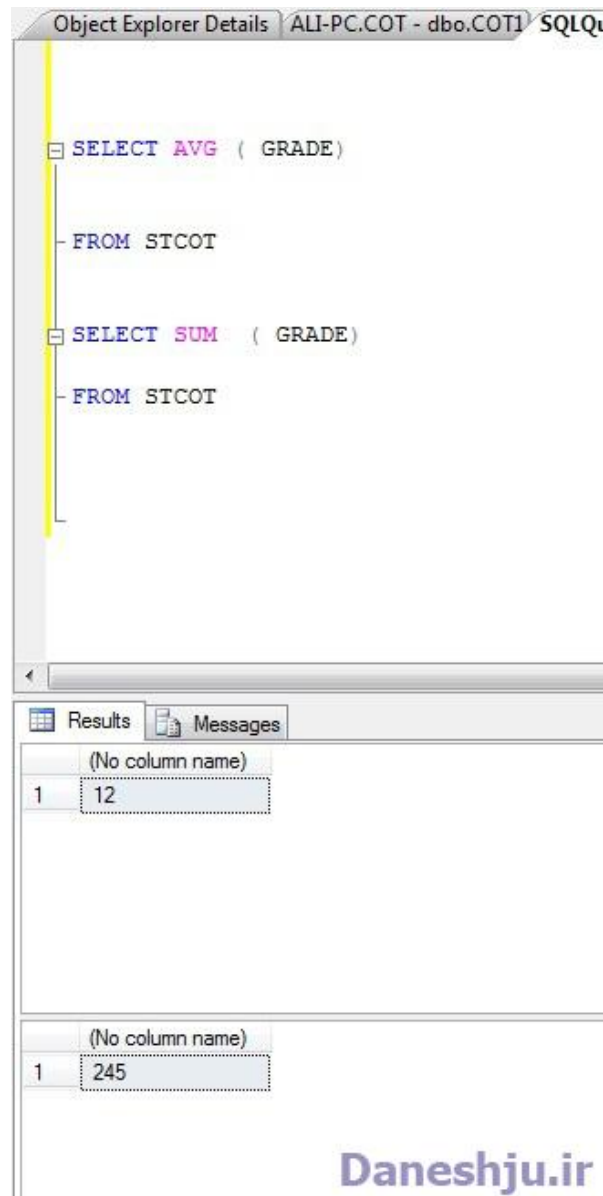
(No column name)
1
13

Daneshju.ir

۳. دستور دیگر AVG یا همان میانگین است. با این دستور، می توانیم میانگین یک سری از مقادیر را محاسبه کنیم.

۴. آخرین دستوری که بیان می کنیم، SUM است. این دستور، جمع مقادیر مورد نظر را به ما می دهد.

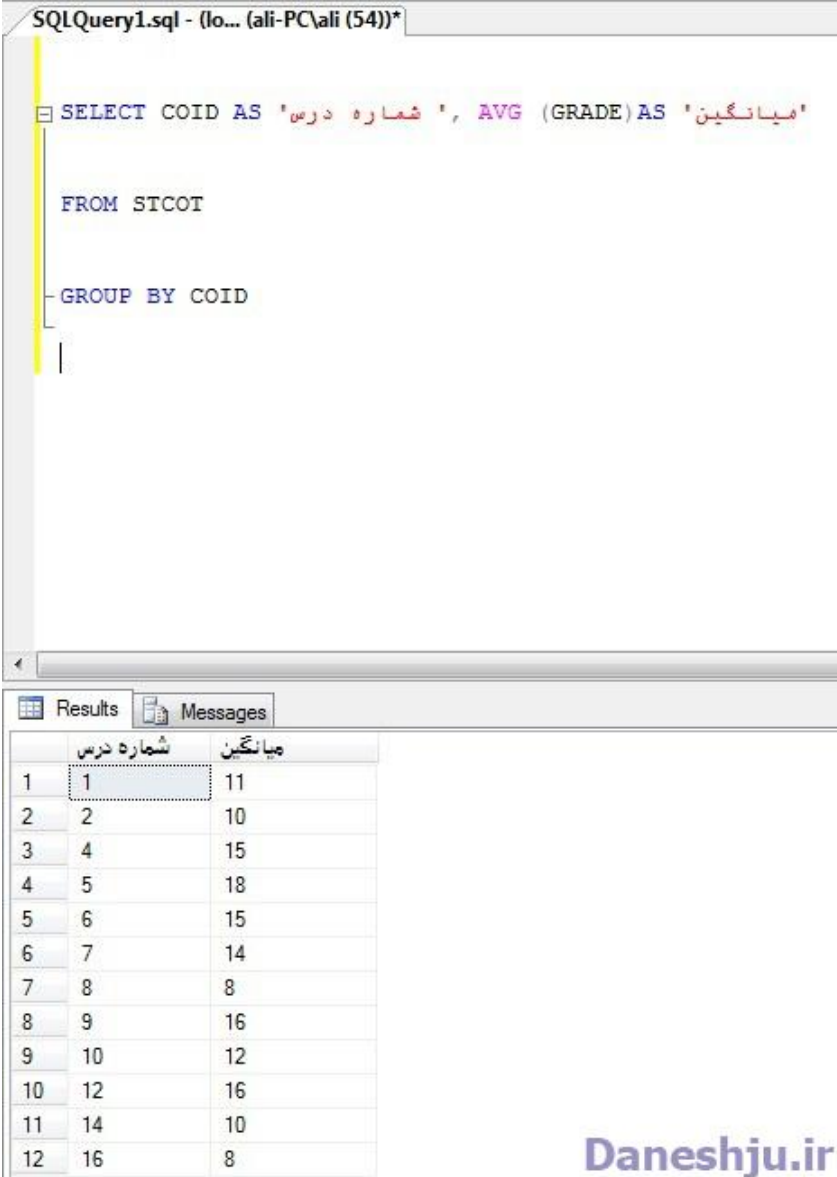
در زیر، مثال هایی از این دو دستور را مشاهده می نمایید.



یکی از دستور های جالب و مهم در پرسش و پاسخ ها، دستور GROUP BY است. با استفاده از این دستور می توان مقادیر یک ستون یا بیشتر از یک ستون را گروه بندی کرد، البته مقدار ستون ها یکسان است.

برای درک بهتر، یک مثال از این دستور ارائه می دهیم.

همانطور که مشاهده می کنید، میانگین نمره و شماره هر درس را به دست آورده ایم و هر کدام را با استفاده از دستور GROUP BY در یک ستون مجزا قرار داده ایم.



The screenshot shows a SQL query window titled 'SQLQuery1.sql - (lo... (ali-PC\ali (54))*'. The query is as follows:

```
SELECT COID AS 'شماره درس', AVG (GRADE) AS 'میانگین'
FROM STCOT
GROUP BY COID
```

Below the query window, the 'Results' tab is active, displaying the following data:

	شماره درس	میانگین
1	1	11
2	2	10
3	4	15
4	5	18
5	6	15
6	7	14
7	8	8
8	9	16
9	10	12
10	12	16
11	14	10
12	16	8

Daneshju.ir

همچنین می توان برای گروه بندی ها، ناظری هم قرار داد. برای این کار، باید از دستور HAVING استفاده کنیم. در اصل این دستور همان کار شرط را انجام می دهد. در شکل مشاهده می کنید، تعداد درس هایی که کمتر از ۲ نفر آن را گرفته اند شمرده و گروه بندی شده است.

ALI-PC.COT - dbo.STT SQLQuery1.sql - (lo... (ali-PC\ali (54))*

```

[اند گرفته را آن نفر 2 از کمتر که درسهایی شماره]
SELECT STCOT.COID AS
FROM STCOT
WHERE TR=2
GROUP BY COID
HAVING COUNT (*) < 2

```

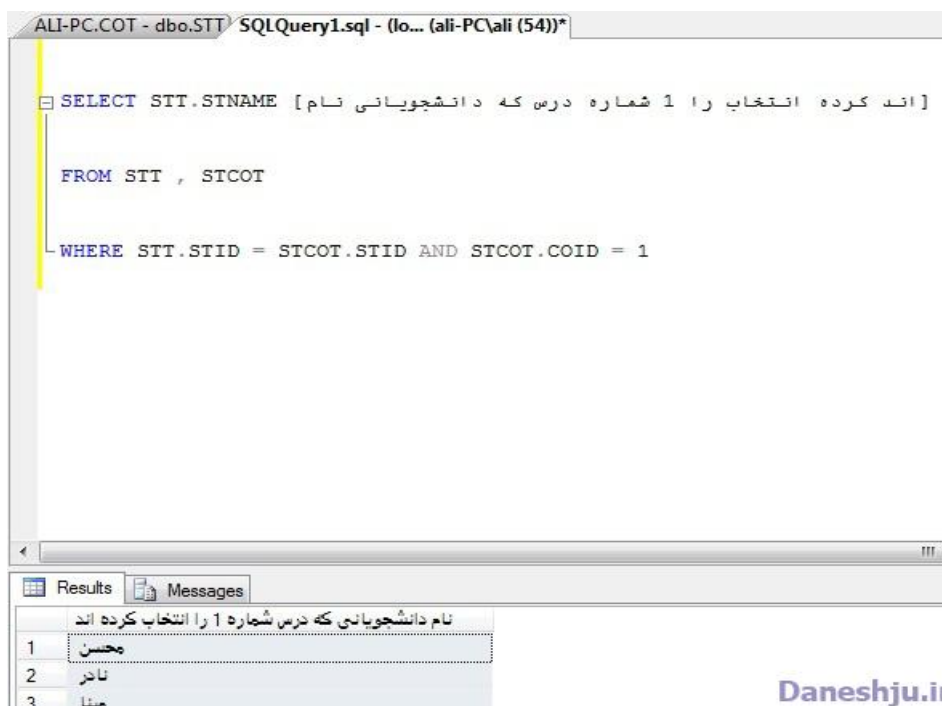
Results Messages

شماره درسهایی که کمتر از 2 نفر آن را گرفته اند	
1	1
2	4
3	5
4	6
5	7
6	8
7	14

Daneshju.ir

بیشتر اوقات برای پرس و جو، داده های چند جدول را نیاز داریم. تا اینجا روش هایی که یاد گرفتیم فقط برای بازیابی داده ها از یک جدول بود. حال می خواهیم داده هایمان را از چند جدول بازیابی کنیم. برای این کار باید از دستور JOIN استفاده نماییم. این دستور که همان پیوند است با استفاده از علامت = انجام می شود.

برای مثال در شکل مشاهده می کنید که نام هر جدول را با داده ای که می خواهیم با (.) کنار هم نوشته و با یکدیگر پیوند زدیم.



Daneshju.ir

گاهی اوقات نیاز داریم که چندین پرسش را همزمان با هم مطرح کنیم. برای این کار باید از پرسش های تو در تو استفاده نمود. البته این روش کمی پیچیده و سخت است که با تمرین می توان مهارت لازم را پیدا نمود. این دستور روش جدیدی نیست و در واقع همان روش SELECT می باشد که با IN می توان چند دستور را داخل هم قرار داد. تنها نکته ای که مهم است مکان قرار دادن این دستورات در داخل یکدیگر می باشد.

برای مثال، می خواهیم نام دانشجوئی که درس مورد نظر را انتخاب کرده اند را بدانیم. برای این کار باید نام دانشجو و شرط وجود داشتن درس مورد نظر در لیست دانشجو را درخواست کنیم.



Daneshju.ir

توجه داشته باشید که پرسش ها را می توان هر چقدر که بخواهید پیچیده کرد و اطلاعات بیشتری بازیابی نمود، ولی همانطور که قبلا هم اشاره کردیم باید مهارت کافی در این زمینه را با تمرین کسب نمایید تا اشکال و خطاها را در این نوع پرسش ها کاهش داد. یک مثال دیگر از این نوع پرسش ها را در زیر مشاهده می کنید، این بار کمی پیچیده تر. در شکل مشاهده می کنید که نام دانشجویانی بازیابی شده که گروه آموزشی درس آنها ۱ بوده و در ترم ۱ آن را انتخاب کرده اند.

The screenshot shows a SQL query window with the following query:

```
SELECT STT.STNAME [نام دانشجویان نام]
FROM STT
WHERE STT.STID IN ( SELECT STCOT.STID
FROM STCOT
WHERE TR=1 AND STCOT.COID IN
(SELECT COT1.COID
FROM COT1
WHERE COT1.CODEID = 1 ))
```

Below the query window, the 'Results' tab is active, displaying a table with 8 rows and 1 column titled 'نام دانشجویان'.

نام دانشجویان
1
2
3
4
5
6
7
8

Daneshju.ir

نکته جالبی در این نوع پرسش ها وجود دارد که در واقع وقتی از IN استفاده می کنیم این امر کاربرد دارد. صورت نایی IN یا همان NOT IN یک کار جالبی را انجام می دهد.

در اینجا با یک مثال این روش را مطرح می کنیم تا درک بهتری داشته باشد. مثلا می خواهیم نام دانشجویانی را داشته باشیم که درس شماره ۱ را انتخاب نکرده اند. می توانیم همان پرسشی که در مورد نام دانشجویانی که درس مورد نظر را انتخاب کرده اند را بنویسیم و یک NOT IN در شرط قبل از شروع پرسش تو در تو بیان کرده تا مخالف پرسش ما را جواب دهد.

SQLQuery1.sql - (lo... (ali-PC\ali (54))*

```

SELECT STNAME [نام دانشجوینی که شماره درس 1 را انتخاب نکرده اند]
FROM STI
WHERE STID NOT IN ( SELECT STID
                    FROM STCOT
                    WHERE COID = 1 )

```

Results Messages

نام دانشجوینی که شماره درس 1 را انتخاب نکرده اند
1
2
3
4
5
6
7
8
9
10

Daneshju.ir

تا اینجا پرسش هایمان را با IN مطرح کردیم، اما باید توجه داشت که در SQL روشی ساده تر هم برای جایگزینی IN وجود دارد. در این روش می توانیم به جای نوشتن IN از عملگرهایی چون (< , <= , > , >= , = , ...) استفاده نماییم. با این عملگرها می توانیم مستقیماً آن سوالی که مد نظر هست را مطرح کنیم. مثلاً معدل دانشجویانی که کمتر از ۱۷ یا نمره دانشجویی که کمتر از ۱۰ و ...

در مثالی که در اینجا مطرح کردیم، درخواستی داده ایم برای نام دانشجویانی که با دانشجوی مد نظر ما هم رشته هستند.

ALI-PC.COT - dbo.COT1 | ALI-PC.COT - dbo.STI | SQLQuery1.sql - (lo... (ali-PC\ali (54))*

```

SELECT STNAME [نام دانشجوین هم رشته با شماره 8809]
FROM STI
WHERE STMJR = ( SELECT STMJR
                FROM STI
                WHERE STID = 8809 )

```

Results Messages

نام دانشجوین هم رشته با شماره 8809
1
2
3
4
5

Daneshju.ir

مطلب آخری که برای پرس و جوها خیلی کاربرد دارد، ترکیب دو جدول هست. گاهی نیاز هست به عنوان مثال، جدول مشخصات دانشجو، جدول دروس و یا ... را در هم ترکیب و یک جدول را در جواب داشته باشیم. برای این کار باید از دستور INNER JOIN استفاده کنیم. کار این دستور در اصل ضرب دو جدول در هم می باشد.

نکته پرکاربرد دیگر که باید ذکر شود تغییر نام جدول های نمایشی در هنگام پاسخ سیستم به ما می باشد. برای این کار در حین درخواست ها در پرس و جو می توانیم با استفاده از کلمه AS نام یک ستون از جدول را که می خواهیم در جواب به ما نشان دهد را به یک زبان صریح مثلا فارسی بیان کند.

در شکل زیر مشاهده می کنید نام هر ستون را با فارسی با کلمه AS جلوی آن نوشته ایم تا در هنگام نمایش پاسخ به همین سبک باشد. برای این کار، باید ابتدا یک PERCENT در شروع کار نوشته تا سیستم، این مفاهیم را تشخیص دهد.

در ابتدا جلوی دستور SELECT یک (100) TOP نوشته ایم. این در اصل تعیین کننده تعداد جواب سیستم به ما می باشد. مثلا در اینجا ۱۰۰ نفر را درخواست کرده ایم.

این یک مثالی کلی و زیباست که مفاهیمی که تاکنون یاد گرفته ایم را در بردارد. هر سه جدول را با هم ترکیب کرده و از آن اطلاعات را واکنشی نموده ایم.

SQLQuery1.sql - (lo... (ali-PC\ali (51)) - ALL-PC.COT - dbo.View_1*

```

SELECT TOP (100)
    PERCENT
    STT.STIID AS [شماره دانشجویی], STT.STNAME AS [نام دانشجو], COT1.COTITLE AS [نام درس], STCOT.GRADE AS [نمره]
FROM
    COT1 INNER JOIN STCOT
        ON COT1.COID = STCOT.COID
    INNER JOIN STT
        ON STCOT.STIID = STT.STIID
    ORDER BY STCOT.GRADE DESC
  
```

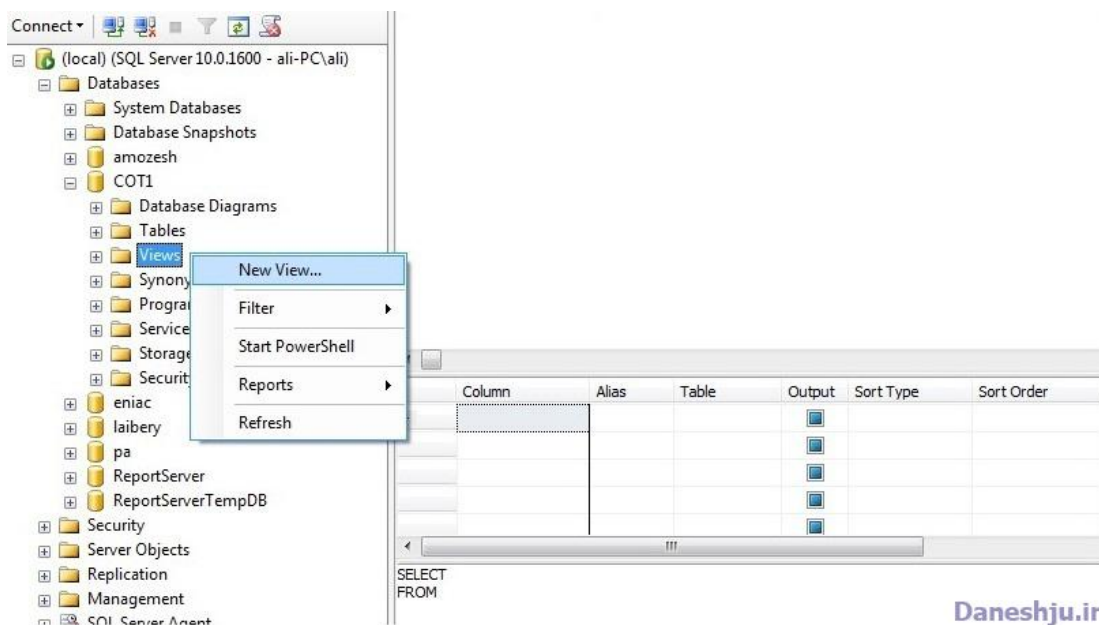
	شماره دانشجویی	نام دانشجو	نام درس	نمره
1	8802	مریم	سیستم عامل	19
2	8801	ناذر	اندیشه 1	18
3	8805	مرضیه	سیستم عامل	18
4	8802	مریم	فیزیک 2	17
5	8808	مهدی	فیزیک 1	16
6	8811	احمد	آز فیزیک 2	16
7	8800	محسن	مهندسی نرم	15
8	8800	محسن	اصول طراحی پیگاه داده	14
9	8802	مریم	اندیشه 2	14
10	8804	مینا	اصول طراحی پیگاه داده	13
11	8806	علی	اندیشه 1	12
12	8809	ارزو	آز سیستم عامل	12
13	8803	حامد	فیزیک 2	11

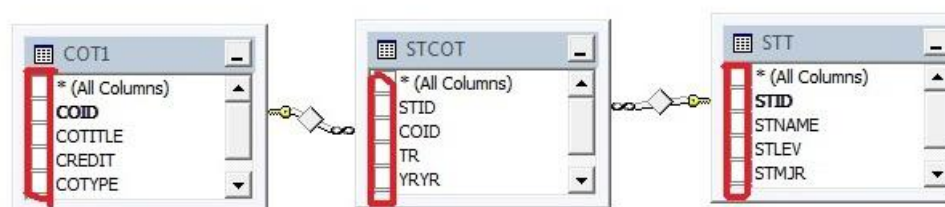
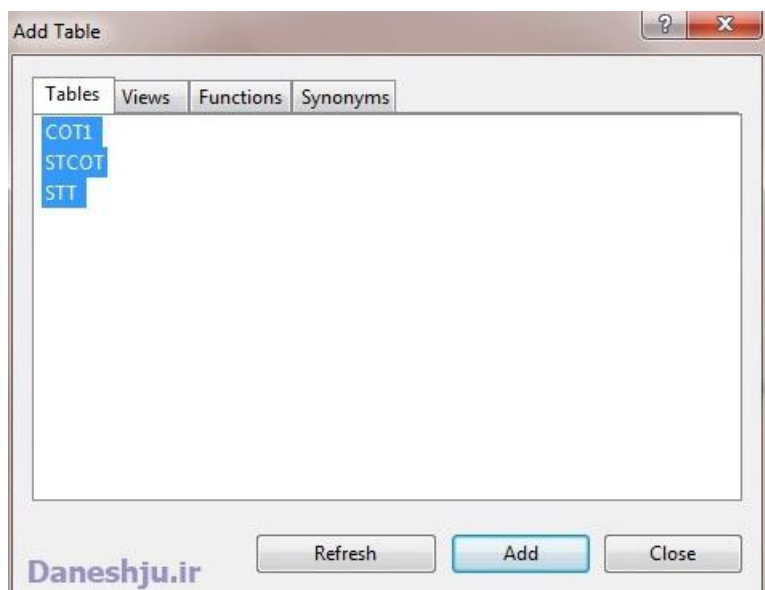
Daneshju.ir

توجه داشته باشید دستور INNER JOIN را هم می توانیم به صورت راحت تری انجام دهیم. با استفاده از دستور VIEWS که در تصویر، مراحل رفتن به این دستور نشان داده شده را می توان انجام داد. VIEWS در اصل یک جدول مجازی است و به صورت واقعی در پایگاه داده وجود ندارد.

نکته بسیار مهمی که باید توجه داشته باشید این است که این بخش، یک بخش محدود برای دسترسی است و این بخش بیشتر کاربردش برای یک مهمان یا کاربر می باشد. برای کسی که طراح و ادمین یک پایگاه داده است باید تمام دسترسی ها وجود داشته باشد که این امر فقط از همان محیط پرس و جو فراهم می گردد. پس سعی کنید زیاد به این محیط وابسته نشوید زیرا این محیط اگر چه بسیار راحت تر از محیط پرس و جو است اما سطح دسترسی محدودی دارد که وقتی به طور عملی با آن کار کنید این حقیقت را خواهید فهمید.

در تصویر مراحل باز کردن یک VIEWS را مشاهده می کنید. پس از کلیک روی گزینه NEW VIEWS یک صفحه مانند شکل دوم باز می شود که درخواست می کند که جدول های مورد نظر خود را به محیط انتخاب و اضافه کنید. ما هر سه جدول خود را اضافه نموده ایم. در شکل سه هر سه جدول و ستون هایی که در آن وجود دارد برای ما نمایش داده می شود. با استفاده از گزینه های کنار نام ستون ها که در تصویر با کادر قرمز مشخص شده می توانیم اطلاعات خود را انتخاب و واکنشی نماییم. با انتخاب هر گزینه، اطلاعات مورد نظر نمایش داده می شود. در اصل همان اعمالی که قبلا در محیط پرس و جو انجام دادیم اینجا اتفاق می افتد، ولی باید دانست که سطح دسترسی محدود است و خیلی از کارهای اضافه را نمی توانیم در این محیط انجام دهیم.





Daneshju.ir

در حال حاضر شما در سطحی از SQL 2008 قرار دارید که می توانید جدول هایی در پایگاهتان ساخته و انواع دستورات را در محیط پرس و جوی آن انجام داده و به راحتی با محیط VIEWS آن کار کنید. شما همچنین می توانید با تمرین بر روی این مفاهیم تسلط پیدا کرده و به خوبی با مقدمات ساخت و پرس و جوی SQL کار نمایید.

سعی کنید تمرینات خود را معطوف به دنیای واقعی خود کرده و روی موسساتی که با آنها در ارتباط هستید، مثلا دانشگاه، کتابخانه و ... انجام داده تا با محیط کار واقعی این نرم افزار بیشتر آشنا شوید.

همیشه موفق و موید باشید

منابع:

- کتاب اصول طراحی پایگاه داده ؛ رانکوهی
- کتاب پایگاه داده ؛ انتشارات پیروز الوان

