



دانشگاه فنی و حرفه‌ای
دانشکده فنی دکتر شریعتی

جلسه چهارم

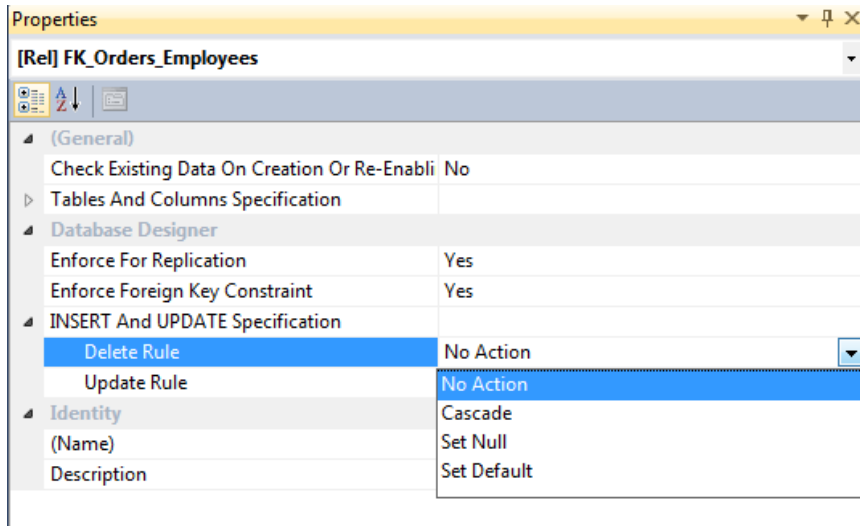
آزمایشگاه پایگاه داده ها

دستور select در SQL Server 2014

ایجاد ارتباطات در جدول ها

- برای ایجاد ارتباط بعد از طراحی جداول می توان وارد محیط دیاگرام شد و سپس ارتباط بین جدول ها را با استفاده از کلیدها مشخص کنیم.
- راه دیگر تعیین ارتباطات کلیک راست در محیط دیزاین جدول و زدن گزینه **relationships...** است.
- توجه : مشخصه ای در ارتباطات بین جدول ها وجود دارد که نحوه برخورد با رکوردهای جدولی که شامل کلید خارجی است، را تعیین می کند، این مشخصه به شکل زیر در **propertise** ارتباطات وجود دارد:
- وارد دیتابیس دیاگرام شده و یک جدول را انتخاب و سپس **relationships** را انتخاب کرده و **propertise** را بررسی می کنیم:

ایجاد ارتباطات در جدول ها

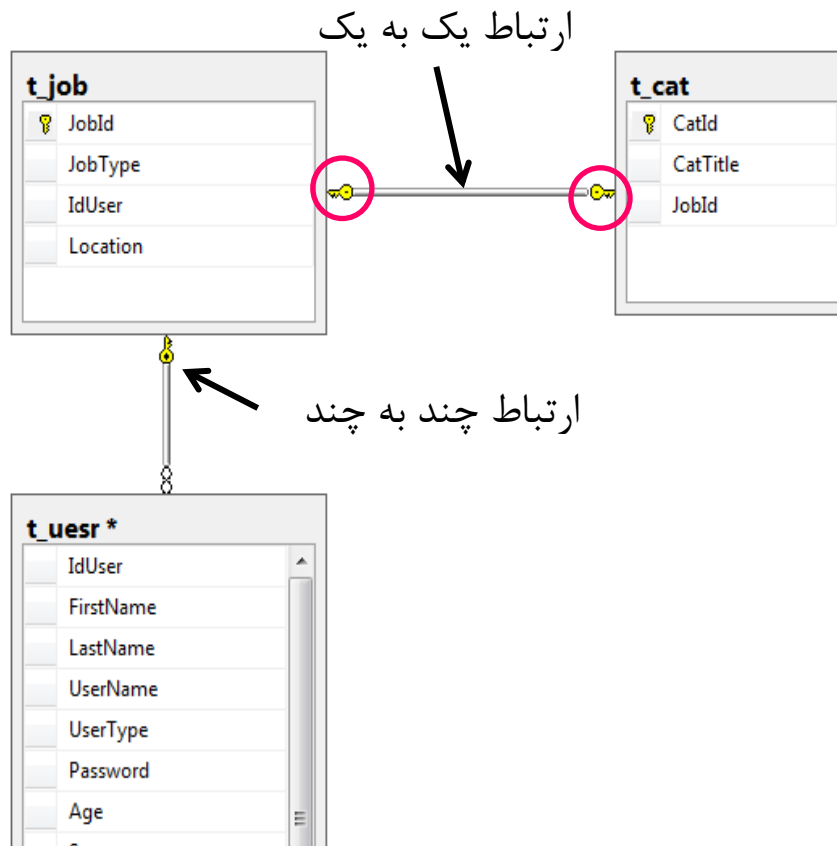


■ مثال:

■ با حذف اطلاعات مربوط به یک کتابدار از کتابخانه، باید اطلاعات تمام کتاب های امانت داده شده توسط آن کتابدار، حذف شود.

■ برای انجام این کار از گزینه **Cascade** استفاده می کنیم.

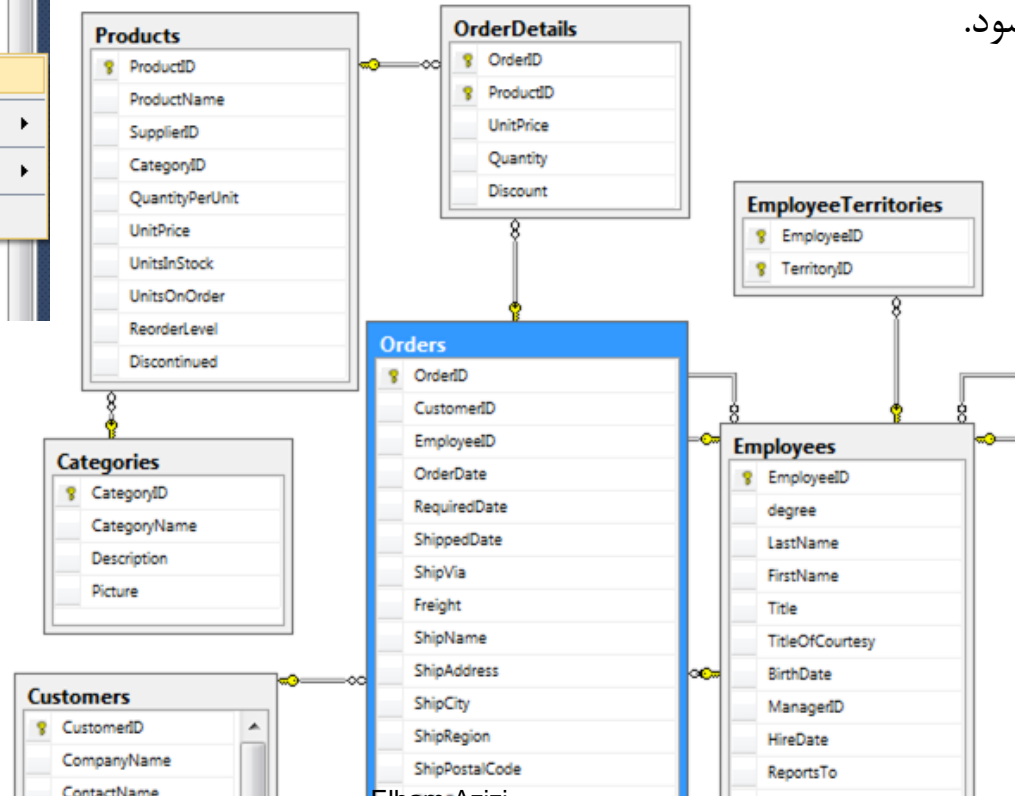
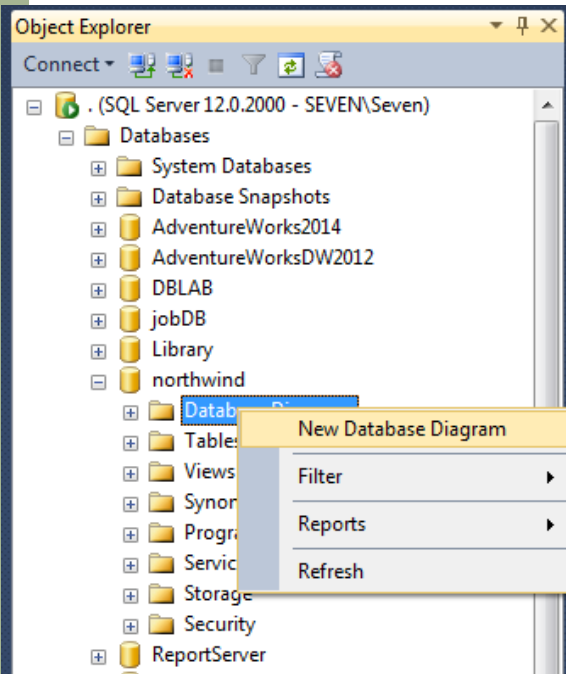
نمایش ارتباطات بین جدول ها



■ برای ایجاد ارتباط یک به یک بین دو جدول، باید از **Unique Key** برای فیلد کلید خارجی استفاده کنیم.

دیتابیس دیاگرام

با استفاده از شاخه Database Diagram می توانید چندین نمودار پایگاه داده ایجاد نمائید. توسط این نمودارها رابطه بین جداول بیشتر مشخص می شود.

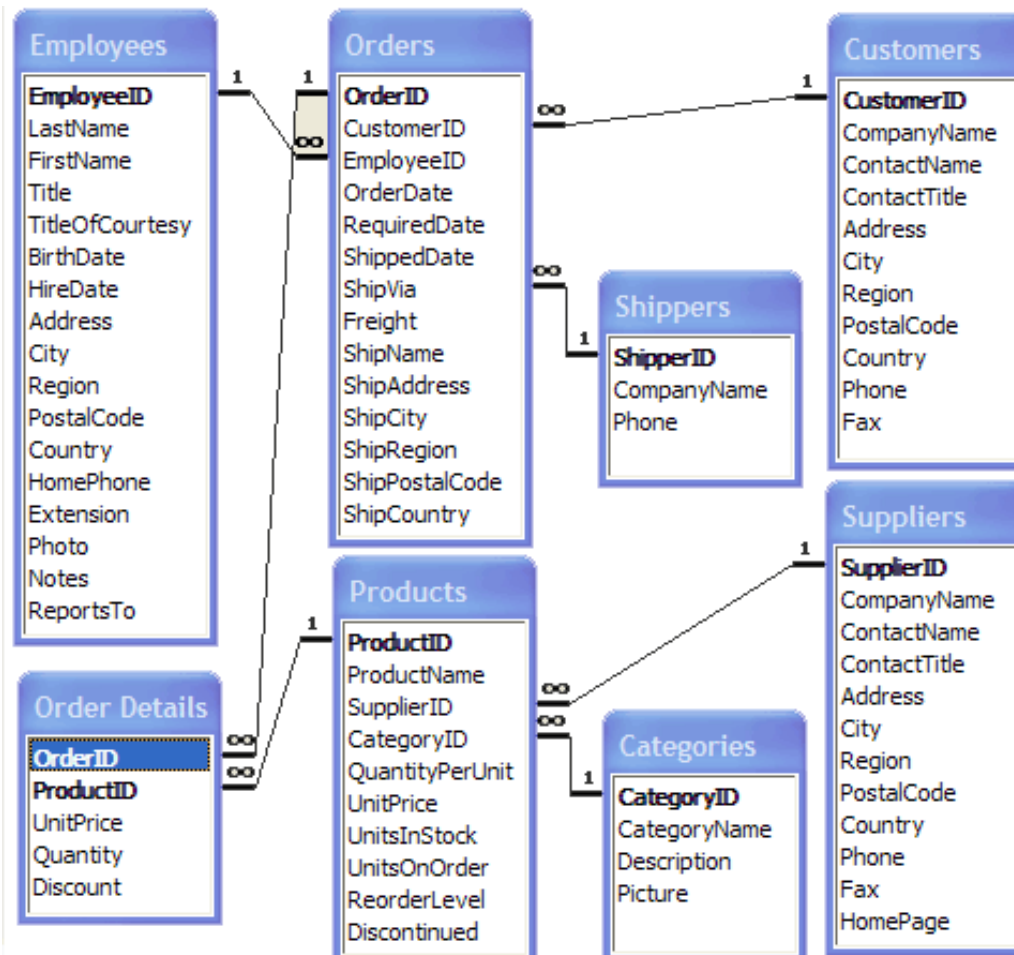


معرفی پایگاه داده Northwind

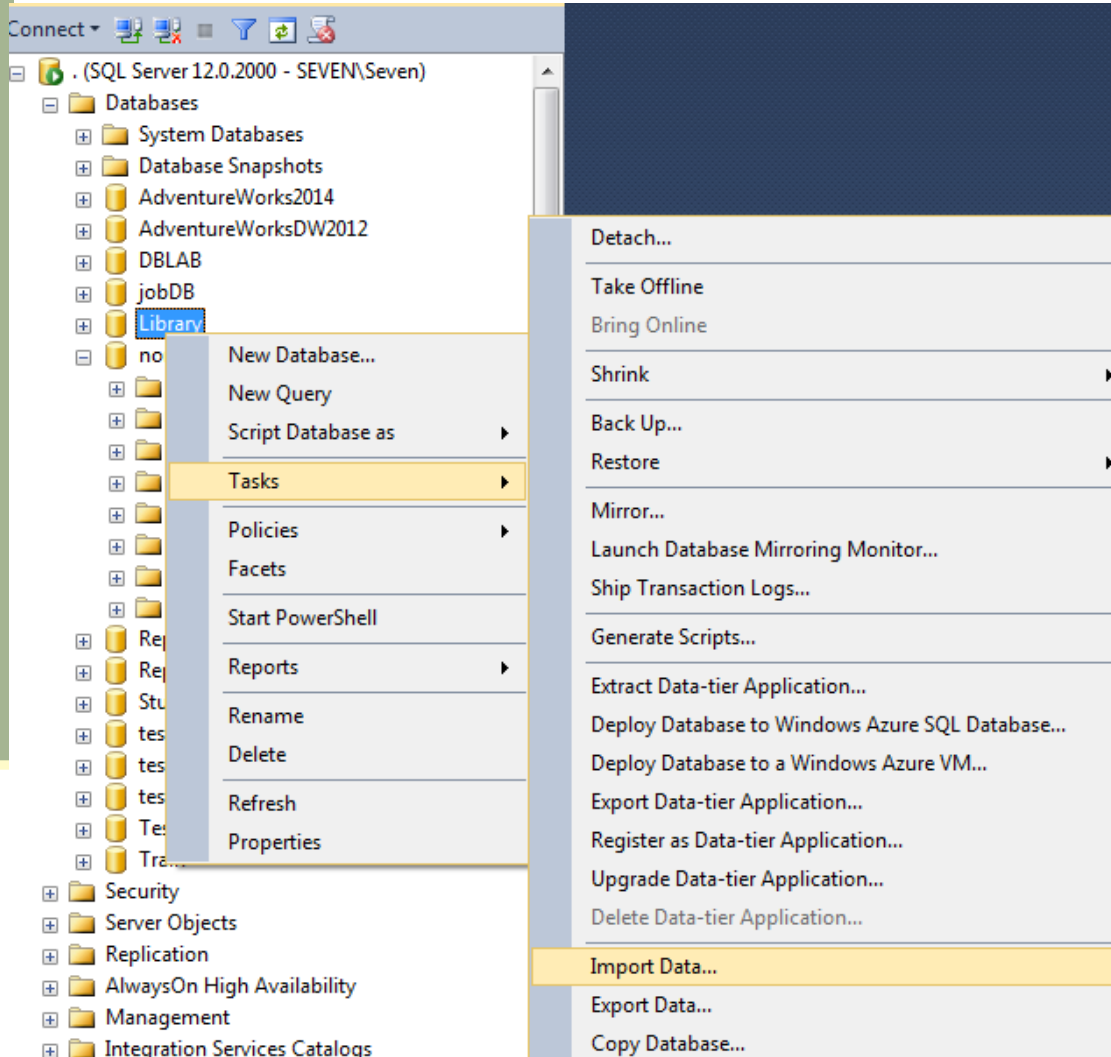
- این دیتابیس شامل جدول های زیر می باشد:
- محصولات (products) < ۸۱ رکورد
- تولیدکنندگان (Suppliers) < ۲۹ رکورد
- مشتریان (Customers) < ۹۱ رکورد
- کارمندان (Employees) < ۹ رکورد
- سفارشات (Orders) < ۸۳۰ رکورد
- جزئیات سفارش (OrderDetails)

The database contains the sales data for Northwind Traders, a fictitious specialty foods export-import company.

معرفی پایگاه داده Northwind



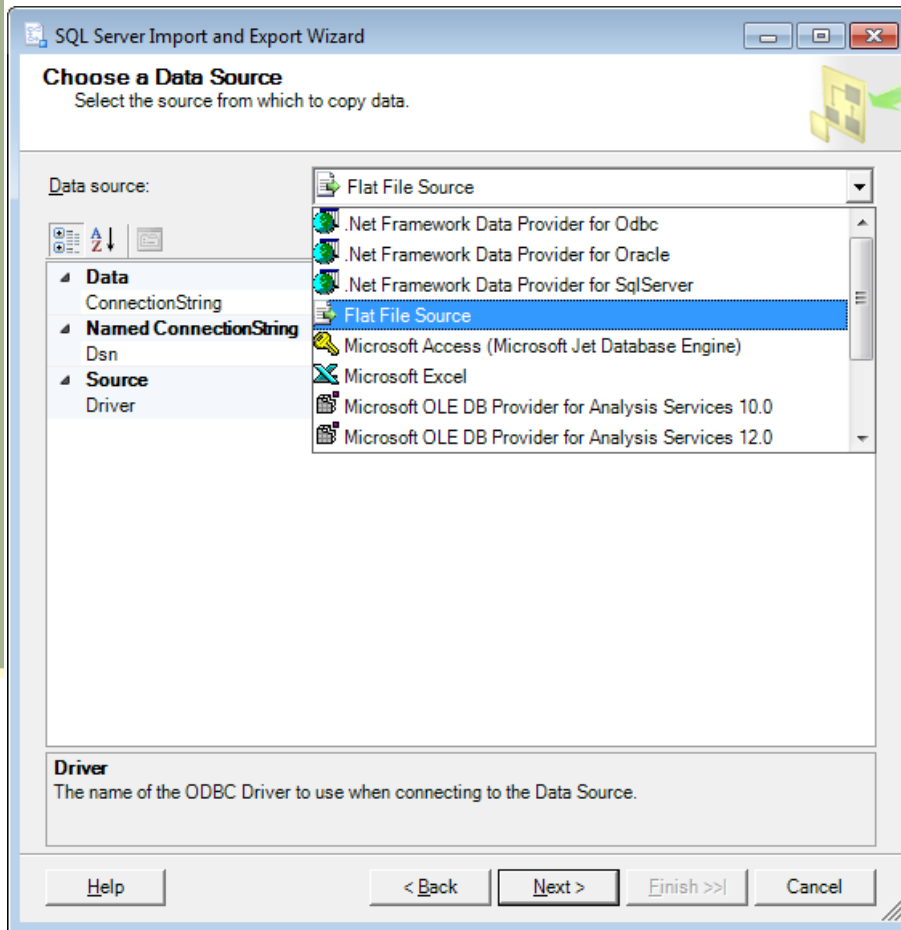
نحوه Import داده ها



برای Import دیتابیس ها و فایل های مختلف به sql server از روش زیر استفاده می کنیم:

روی دیتابیس مورد نظر کلیک راست کرده و از بخش Tasks گزینه Import Data را می زنیم.

نحوه Import داده ها



بخش **Data source** در این پنجره نوع فایل مورد نظر برای **Import** را مشخص می کند:

■ Flat File Source

■ فایل های text و CSV و ...

■ Microsoft Excel

■ Microsoft Access


■ فایل های *.mdb

■ و ...

- **SQL** is a query language to operate on sets.
 - It is more or less standardized, and used by almost all relational database management systems: SQL Server, Oracle, MySQL, PostgreSQL, DB2, Informix, etc.
- **PL/SQL** is a proprietary procedural language used by Oracle
- **PL/pgSQL** is a procedural language used by PostgreSQL
- **TSQL** is a proprietary procedural language used by [Microsoft](#) in SQL Server.

- Distinct
- Sorting
- Filtering
- Wildcards
- New Functions in SQL 2012
- Case

مثال هایی از دستور Select

- `select * from customers`
- `select CompanyName from customers`
- `select CompanyName from customers As 'نام شرکت'`
- `select FirstName + ' ' + lastName as Fullname`
 Concatenating
- `select 1+2`
- `select 'hello world'`

مرتب سازی در دستور Select (ادامه)

مرتب سازی صعودی شهر در جدول مشتریان

```
■ select * from customers order by city
```

مرتب سازی نزولی شهر در جدول مشتریان

```
■ select * from customers order by city desc
```

مرتب سازی نزولی شهر و در صورت برابر بودن شهر ، مرتب سازی صعودی بر اساس آدرس

```
■ select * from customers  
order by city desc , address asc
```

مرتب سازی بر اساس طول سمت شغلی مشتریان

```
■ select * from customers order by len(ContactTitle)
```

دستور Select (ادامه)

شهرهای تکراری فقط یکبار نشان داده می شود

■ `select distinct city from customers`

فقط پنج رکورد اول از جدول محصولات نمایش داده می شود

■ `select top (5) ProductName from Products`

نمایش 25 درصد از رکوردهای جدول سفارشات

■ `select top 25 percent * from Orders`

تمرین

۱. نمونه ای از `distinct` برای دو فیلد

استفاده از دستور where

TABLE 1: WHERE Clause Operators

OPERATOR	EXAMPLE USAGE
=, >, <, >=, <=, <>, !=, !>, !<	<Column Name> = <Other Column Name> <Column Name> = 'Bob'
AND, OR, NOT	<Column1> = <Column2> AND <Column3> >=<Column 4> <Column1> != "MyLiteral" OR <Column2> = "MyOtherLiteral"
BETWEEN	<Column1> BETWEEN 1 AND 5
LIKE	<Column1> LIKE "ROM%"
IN	<Column1> IN (List of Numbers) <Column1> IN ("A", "b", "345")
ALL, ANY, SOME	<column expression> (comparison operator) <ANY SOME> (subquery)
EXISTS	EXISTS (subquery)

فیلتر کردن در دستور Select

انتخاب کارمندان شهر لندن

```
■ select * from employees where city='london'
```

انتخاب کارمندانی که نام آن ها با حرف A شروع می شود

```
■ select firstname from employees where firstname  
like 'A%'
```

انتخاب کارمندانی که نام آن ها با حرف A تمام می شود

```
■ select firstname from employees where firstname  
like '%A'
```

انتخاب کارمندانی که در نام آن ها حرف A وجود دارد

```
■ select firstname from employees where firstname  
like '%A%'
```


Wildcards در دستور Like

TABLE 3-1 Wildcards used in LIKE patterns

Wildcard	Meaning	Example
% (percent sign)	Any string including an empty one	'D%': string starting with D
_ (underscore)	A single character	'_D%': string where second character is D
[<character list>]	A single character from a list	'[AC] %': string where first character is A or C
[<character range>]	A single character from a range	'[0-9] %': string where first character is a digit
[^<character list or range>]	A single character that is not in the list or range	'[^0-9] %': string where first character is not a digit

فیلتر کردن در دستور Select (ادامه)

انتخاب کارمندانی که نام خانوادگی آن ها Davilio یا Davolio است.

- `select * from employees where lastname like 'Dav[oi]lio'`
- `select * from employees where lastname like 'Dav_liO'`

انتخاب کارمندانی که نام خانوادگی آن ها به حرف n ختم نمی شود.

- `select * from employees where LastName like '%[^n]'`

فیلتر کردن در دستور Select (ادامه)

انتخاب سفارشات که تعداد محصولات سفارش داده شده آن بین 5 تا 10 است.

```
select * from OrderDetails where Quantity between 5  
and 10
```

نمایش اطلاعات تولید کنندگان کشور انگلیس یا آمریکا

```
select * from Suppliers where Country='uk' or  
Country='usa'
```

```
select * from Suppliers where Country in  
( 'uk' , 'usa' , 'japan' )
```

دستور Case

```
CASE WHEN <predicate> THEN <true_result> ELSE <false_or_unknown_result> END
```

■ دو مدل ساده و جستجویی دارد:

■ مثال فرم ساده دستور **Case** که برای محصولات که مقدار فیلد تخفیف آن ها ۰ می باشد ، عبارت **No** و در صورتی که این مقدار یک باشد، عبارت **yes** را نمایش می دهد.

```
Select ProductID,ProductName,UnitPrice,Discontinued,  
Case Discontinued  
When 0 Then 'No'  
When 1 Then 'Yes'  
Else 'Unknown'  
End As Discontinued_Desc  
From Products
```

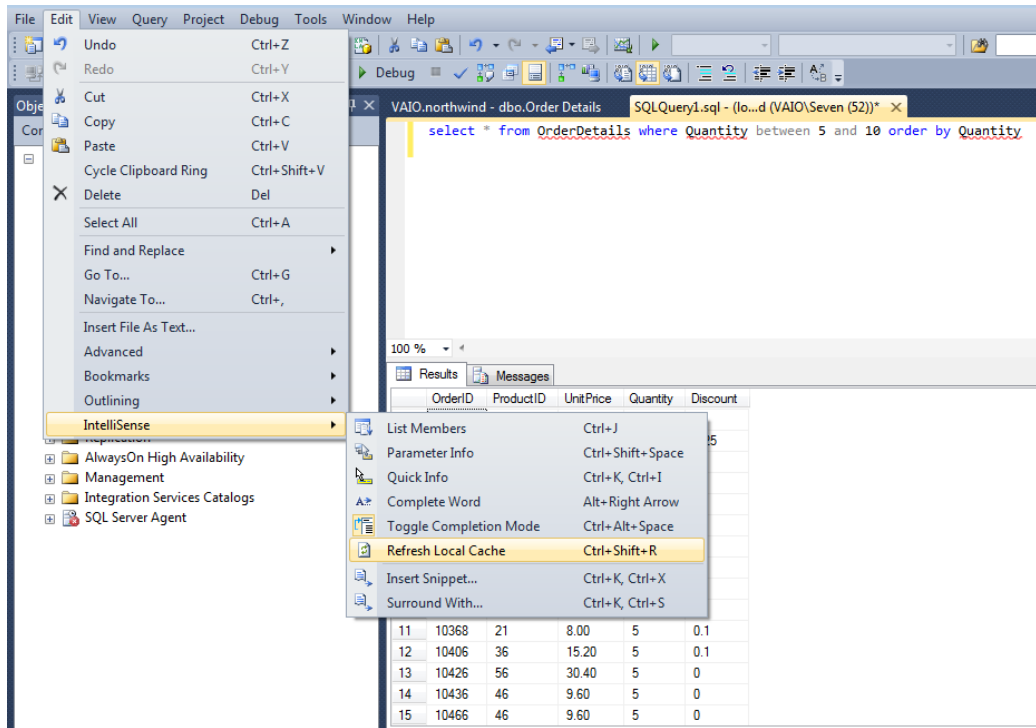
مدل جستجویی دستور Case

■ مثال فرم جستجویی که با توجه به مقدار قیمت هر واحد محصول، در ستونی دیگر سه مقدار پایین، بالا و متوسط برای آن نمایش می دهد.

```
Select ProductID,ProductName,UnitPrice,  
  Case  
    When UnitPrice <20 Then 'Low'  
    When UnitPrice <40 Then 'Medium'  
    When UnitPrice >=40 Then 'High'  
    Else 'Unknown'  
  End As Pricerange  
From Products
```

استفاده از Refresh Local Cache

زمانی که تغییری مثل عوض کردن نام جدول و ... در دیتابیس به وجود می آوری و علامت قرمز رنگ زیر کلمات ظاهر میشود که با استفاده از منوی edit و بخش IntelliSense میتوان عمل refresh را برای دیتابیس انجام داد.



تابع فرمت (از توابع جدید SQL Server 2012)

■ شماره محصول از جدول محصولات را به صورت ده رقمی نشان دهید.

■ مثلاً 42 به شکل 0000000042 نشان داده شود.

```
SELECT productid,  
       RIGHT(REPLICATE('0', 10) + CAST(productid AS VARCHAR(10)), 10) AS str_productid  
FROM Production.Products;
```

■ در SQL 2012 تابعی معرفی شده که فرایند بالا را آسان تر انجام می دهد:

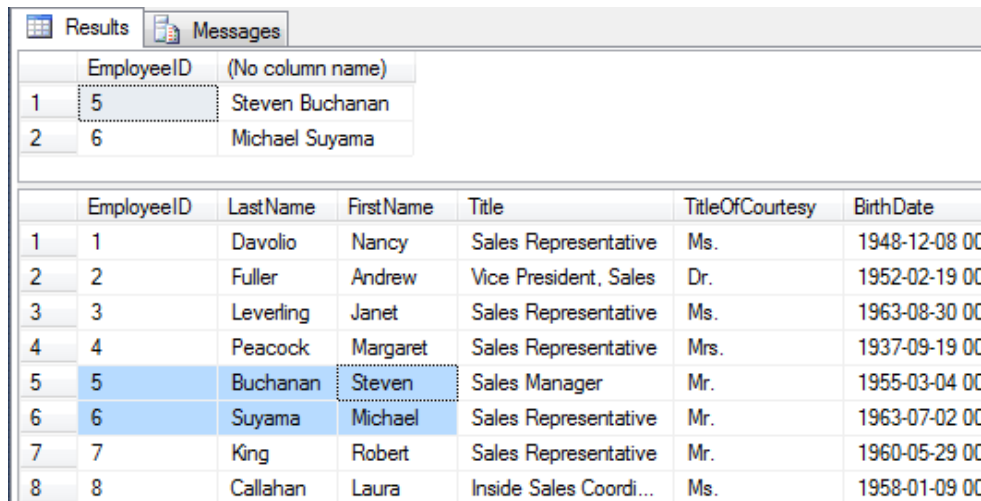
```
SELECT productid,  
       FORMAT(productid, 'd10') AS str_productid  
FROM Production.Products;
```



تابع Offset Fetch (از توابع جدید SQL Server 2012)

- در SQL Server 2012 تابعی معرفی شده که امکان انتخاب ناحیه مشخصی از Result set را فراهم می کند.
- توجه داشته باشید این تابع فقط با دستور order by باید استفاده شود.

```
SELECT EmployeeID, ,FirstName + ' ' + LastName  
FROM Employees  
ORDER BY EmployeeID ASC OFFSET 4 ROWS FETCH NEXT 2 ROWS ONLY
```



	EmployeeID	(No column name)
1	5	Steven Buchanan
2	6	Michael Suyama

	EmployeeID	LastName	FirstName	Title	TitleOfCourtesy	BirthDate
1	1	Davolio	Nancy	Sales Representative	Ms.	1948-12-08 00
2	2	Fuller	Andrew	Vice President, Sales	Dr.	1952-02-19 00
3	3	Leverling	Janet	Sales Representative	Ms.	1963-08-30 00
4	4	Peacock	Margaret	Sales Representative	Mrs.	1937-09-19 00
5	5	Buchanan	Steven	Sales Manager	Mr.	1955-03-04 00
6	6	Suyama	Michael	Sales Representative	Mr.	1963-07-02 00
7	7	King	Robert	Sales Representative	Mr.	1960-05-29 00
8	8	Callahan	Laura	Inside Sales Coordi...	Ms.	1958-01-09 00

این دستور از خروجی select کارمندان که به صورت صعودی شماره کارمند، مرتب شده است، از سطر ۴ به بعد، به تعداد دو سطر انتخاب می کند و نمایش می دهد.

نکات استفاده از Offset Fetch

- استفاده از دستور `order by` در این دستور اجباری است.
- استفاده از عبارت `Fetch` بدون دستور `Offset` امکان پذیر نیست.
- امکان ترکیب دستور `Top` و دستور `Offset Fetch` به طور همزمان در یک کوئری نیست.
- برای مشخص کردن تعداد رکورد در این دستور از هر مقدار عددی، محاسباتی، ثابت می توان استفاده کرد اما امکان استفاده از `scalar sub query` ها نیست.

مثال هایی از دستور Offset Fetch

■ مثال ۱

■ نمایش نام و نام خانوادگی همه کارمندان از سطر سوم به بعد در result set

- Skip first 3 rows from the sorted result set and return the remaining rows.

```
SELECT EmployeeID, FirstName + ' ' + LastName  
FROM Employees  
ORDER BY EmployeeID ASC OFFSET 3 ROWS
```

تمرین

1. کارمندی که بعد از تاریخ ۱/۱۷/۱۹۹۳ استخدام شده اند را نمایش دهید.
2. مشتریانی که نام شرکت آنان با حرف **B** شروع می شود را نمایش دهید.

پاسخ تمرینات

1. `SELECT FirstName, LastName, Title FROM Employees WHERE HireDate > '1/17/1993'`
2. `select CompanyName from customers where CompanyName LIKE 'B%'`

- یک کوئری بنویسید برای انتخاب سه مشتری به صورت تصادفی، با هر بار اجرای کوئری سه نفر از مشتریان لیست شود.
- تابع `newID()` : تابعی است که با هر بار اجرا مقداری یکتا بر می گرداند.
- حل:

Select top 3 customerID

From customers

Order by newID()