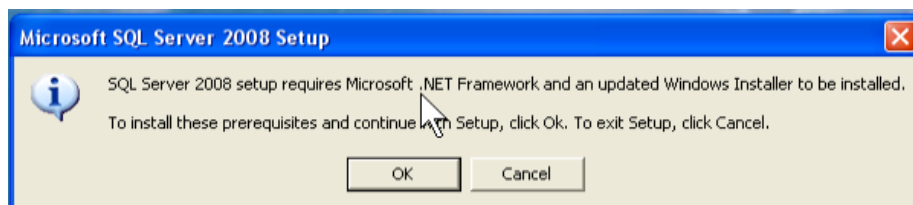


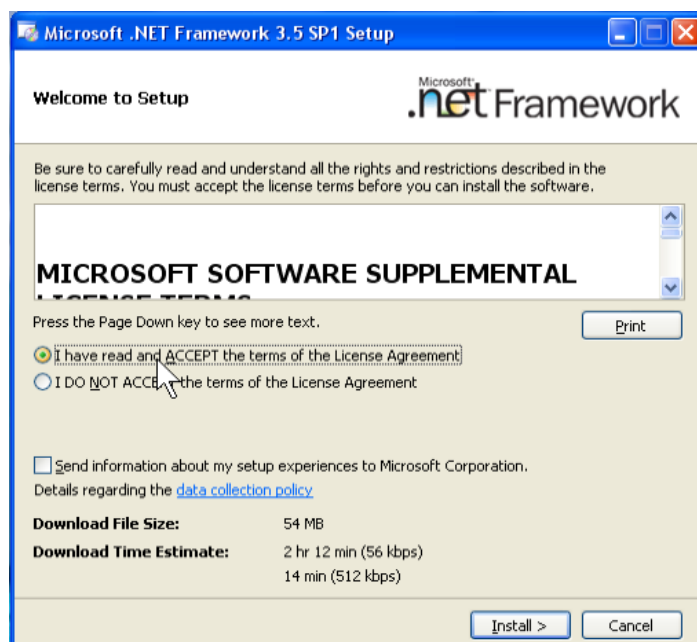
آموزش نصب
SQL Server 2008

جمع آوری : محسن داورزنی

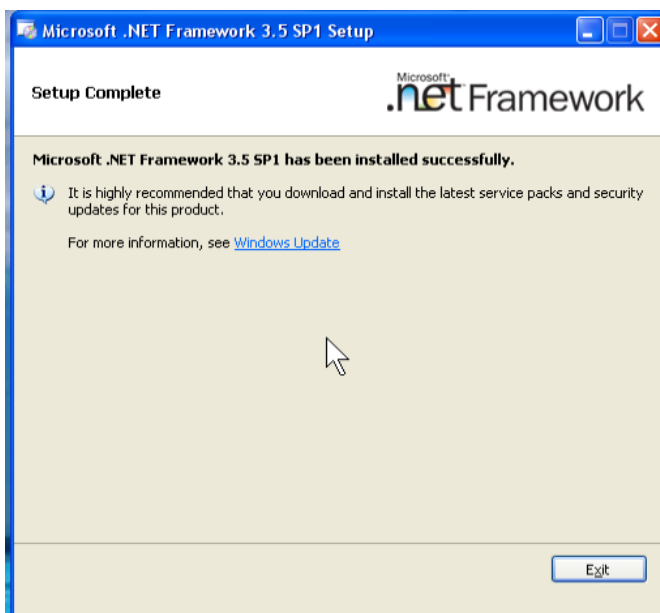
آموزش نصب SQL SERVER 2008 : ابتدا DVD را در دستگاه قرار می دهیم .

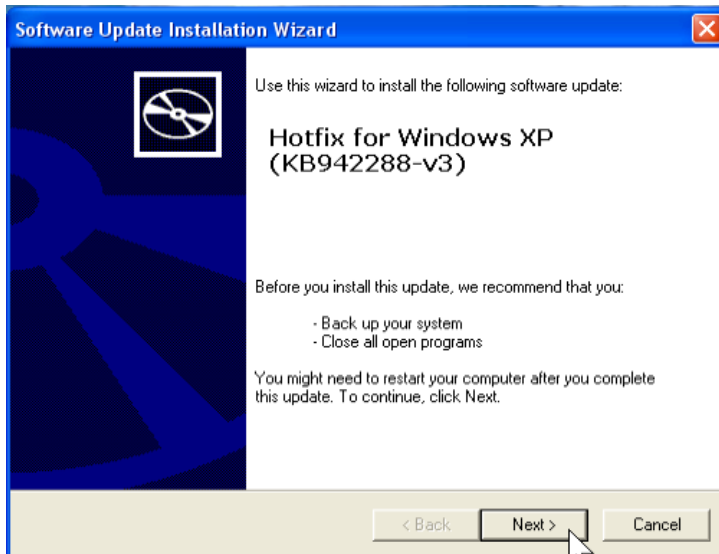


روی گزینه OK کلیک می کنیم ، کادر زیر ظاهر می شود . گزینه I have read and accept ... را انتخاب می کنیم . با کلیک روی Next به مرحله بعدی می رویم و نصب .Net Framework انجام میشود.



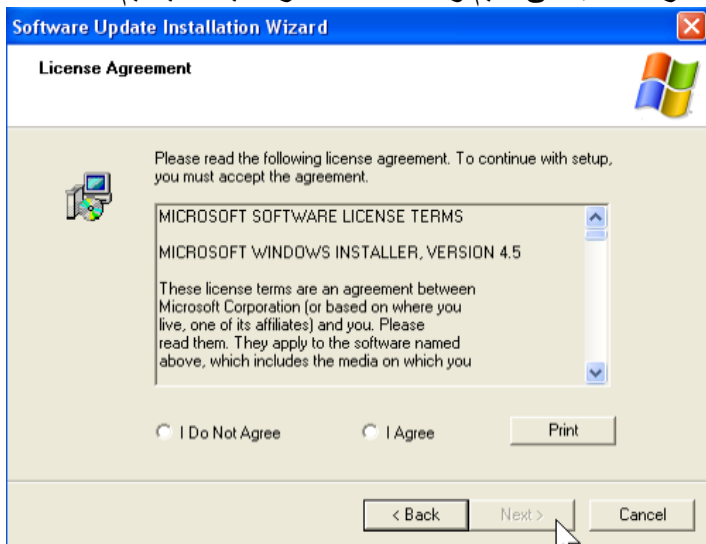
با اتمام نصب Net Framework . کادر زیر ظاهر می شود . روی دکمه Exit کلیک میکنیم.



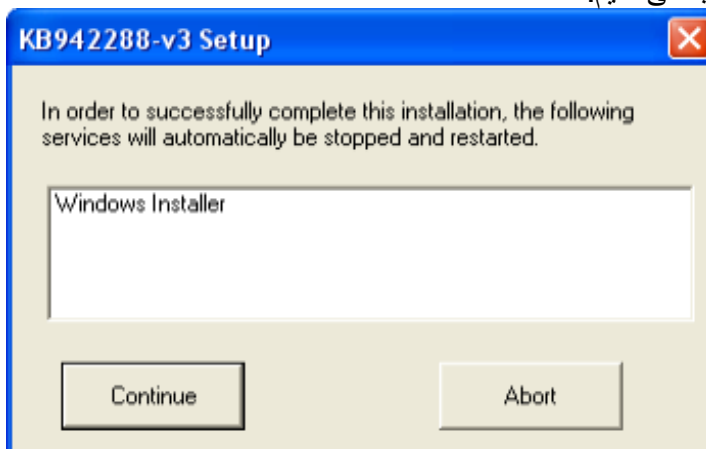


سپس در کادر روبرو دکمه Next را کلیک می کنیم .

آنگاه کادر زیر ظاهر می شود . گزینه Agree را انتخاب می کنیم و دکمه Next را کلیک میکنیم .



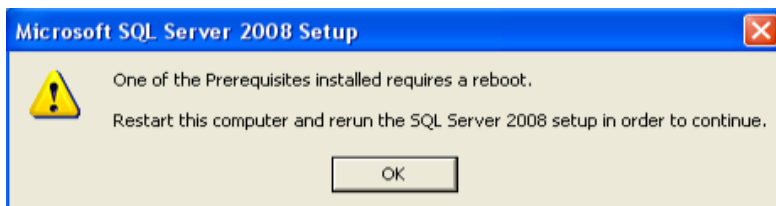
سپس در کادر زیر گزینه continue را انتخاب می کنیم.



سپس روی دکمه finish کلیک کنید .



حال پیغام زیر ظاهر میشود که با کلیک روی آن کامپیوتر restart میشود .

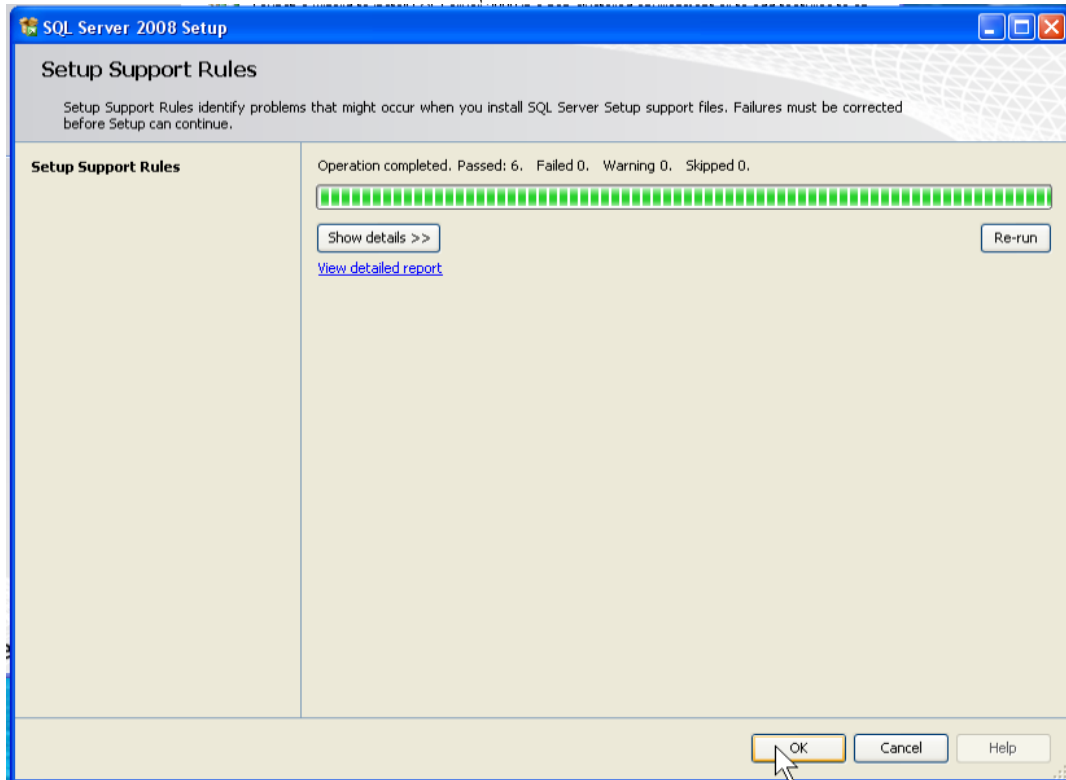


سپس با restart شدن کامپیوتر کادر زیر ظاهر می شود که روی گزینه installation کلیک می کنیم .

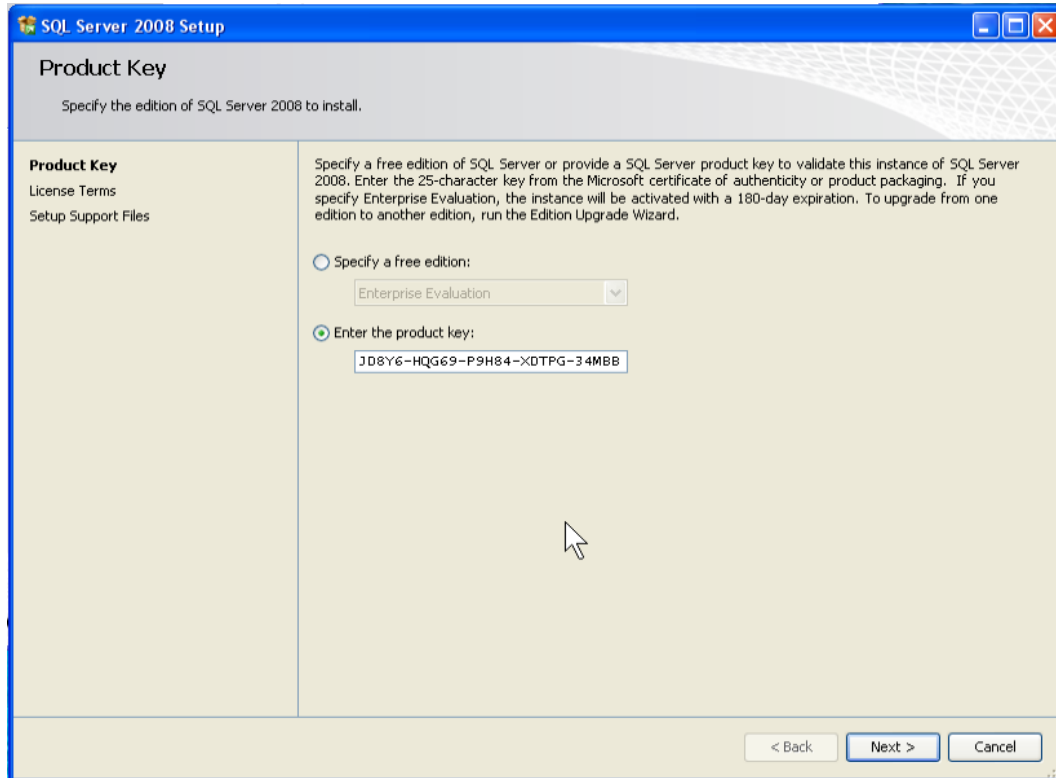


سپس در کادر بالا اولین گزینه را انتخاب میکنیم.

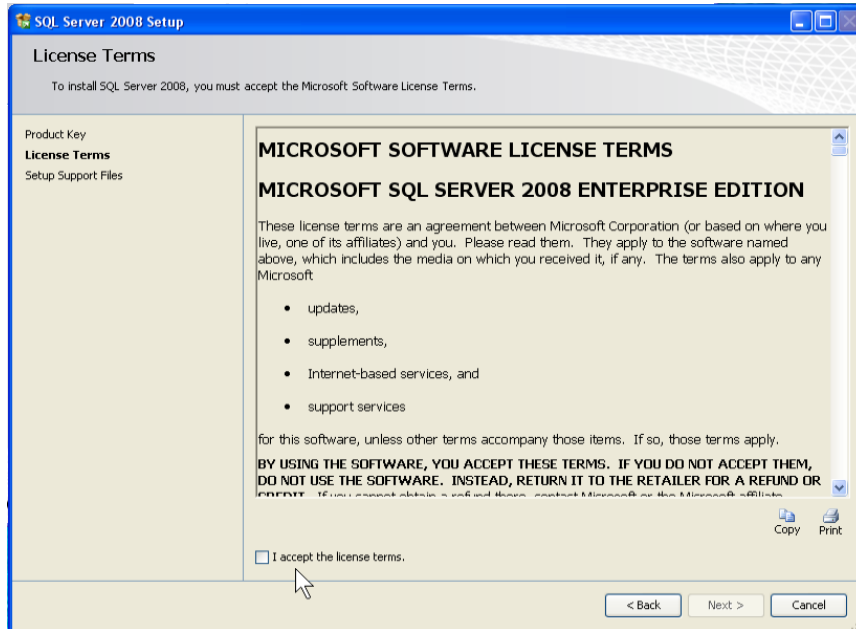
سپس کادر زیر ظاهر میشود. گزینه Ok را کلیک میکنیم.



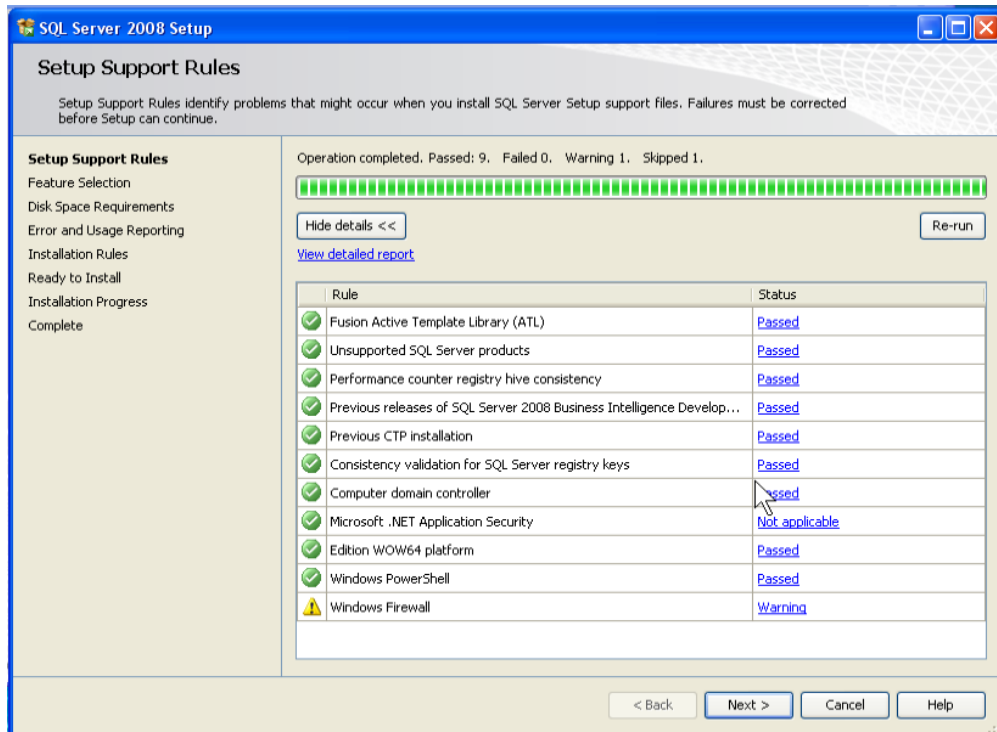
آنگاه پنجره Product Key ظاهر می شود. که به صورت خودکار Product Key وارد شده است.



دکمه Next را کلیک میکنیم . کادر زیر ظاهر می شود . تیک گزینه Accept را میزنیم و سپس روی گزینه Next کلیک میکنیم.

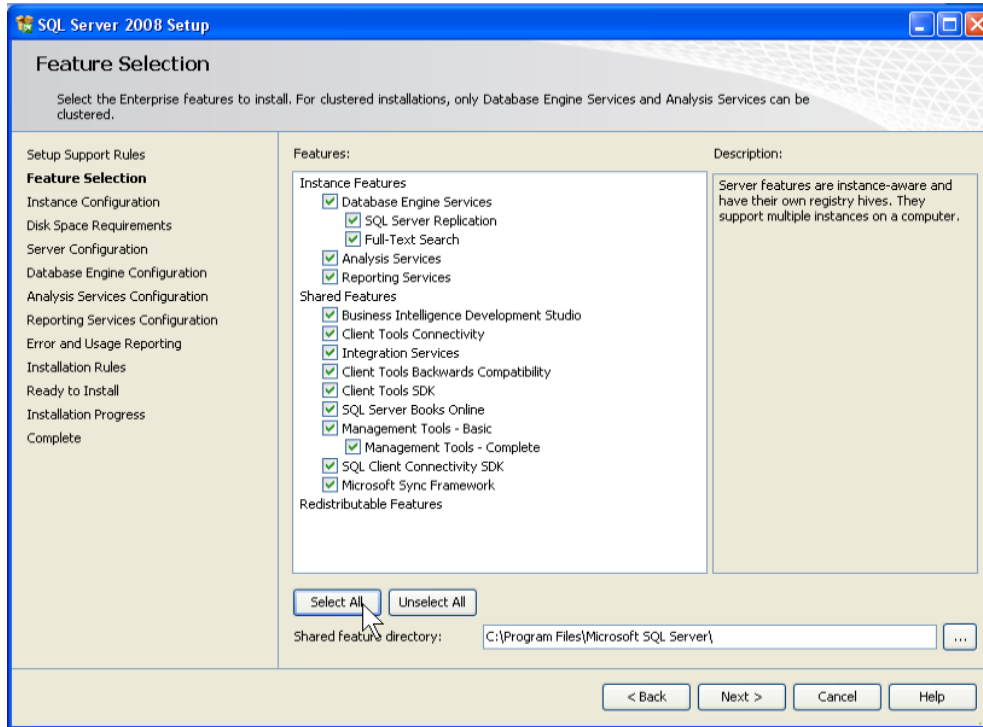


سپس کادر زیر ظاهر میشود . با کلیک روی دکمه install عملیات نصب شروع می شود .
سپس کادر زیر ظاهر میشود .



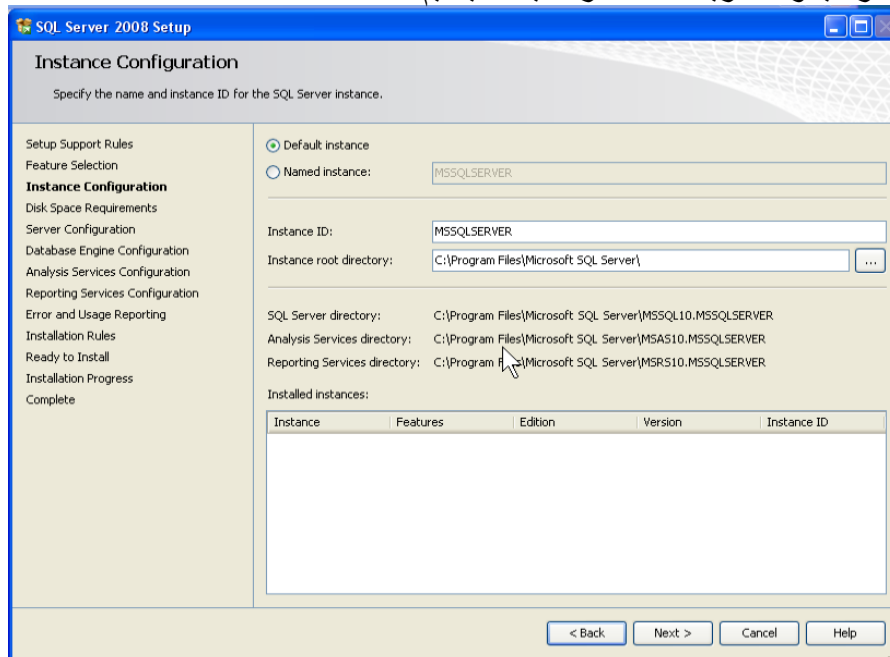
دکمه Next را می زنیم .

کادر زیر ظاهر میشود .

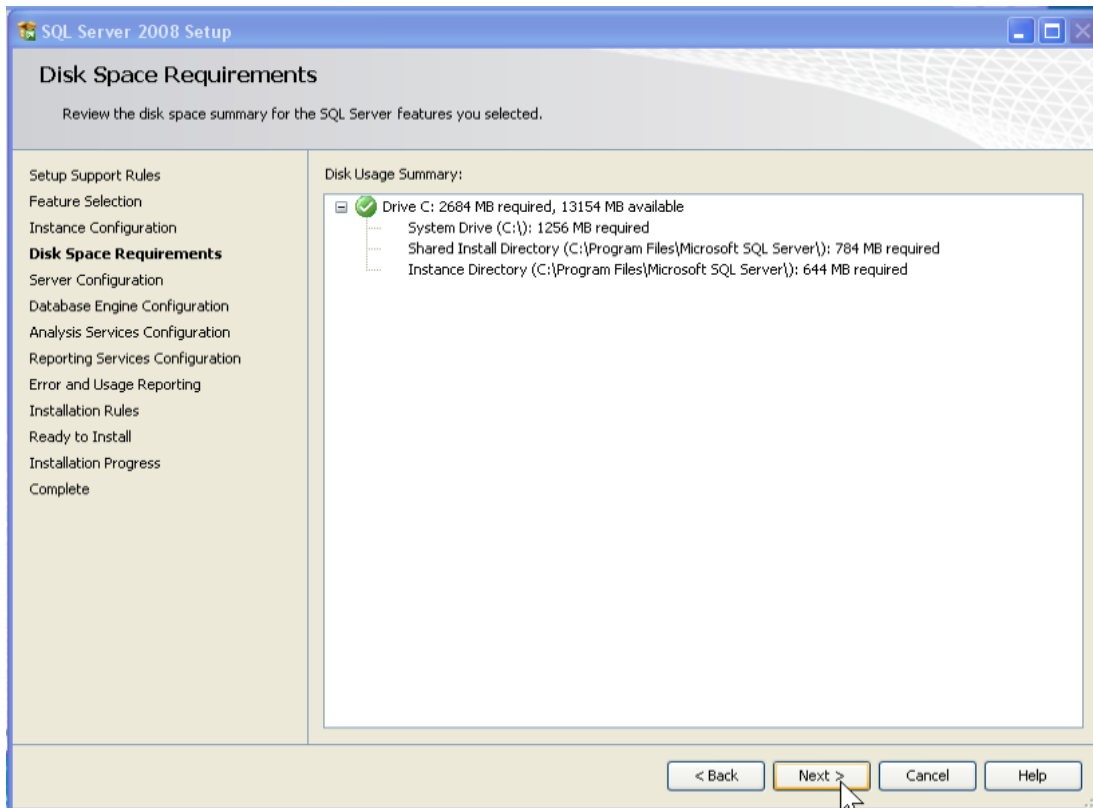


دکمه select all ظاهر میشود روی آن کلیک میکنیم و سپس دکمه Next را می زنیم .

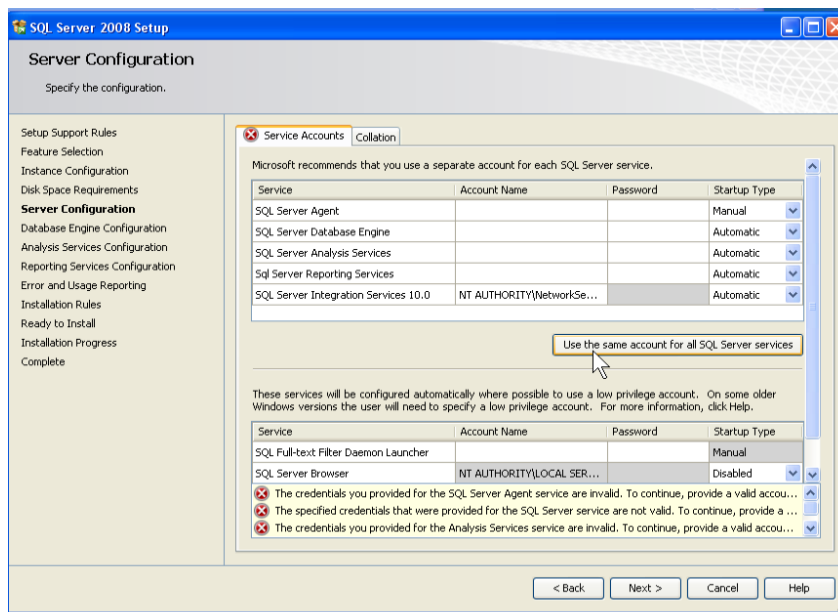
کادر زیر ظاهر میشود . گزینه Next را کلیک میکنیم.



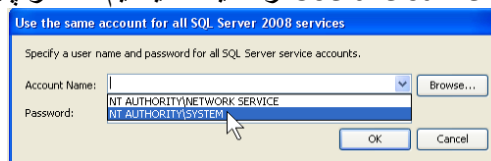
مجدداً دکمه Next را می زنیم. کادر صفحه بعدی ظاهر میشود.



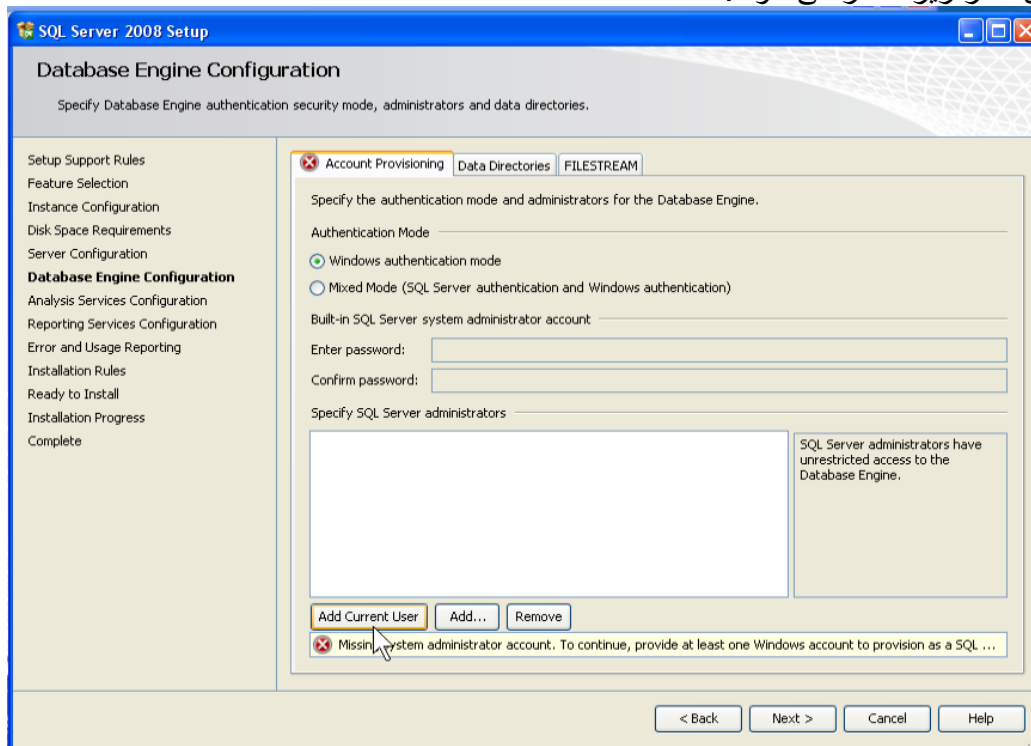
در کادر بالا دکمه Next را کلیک میکنیم . کادر زیر ظاهر میشود.



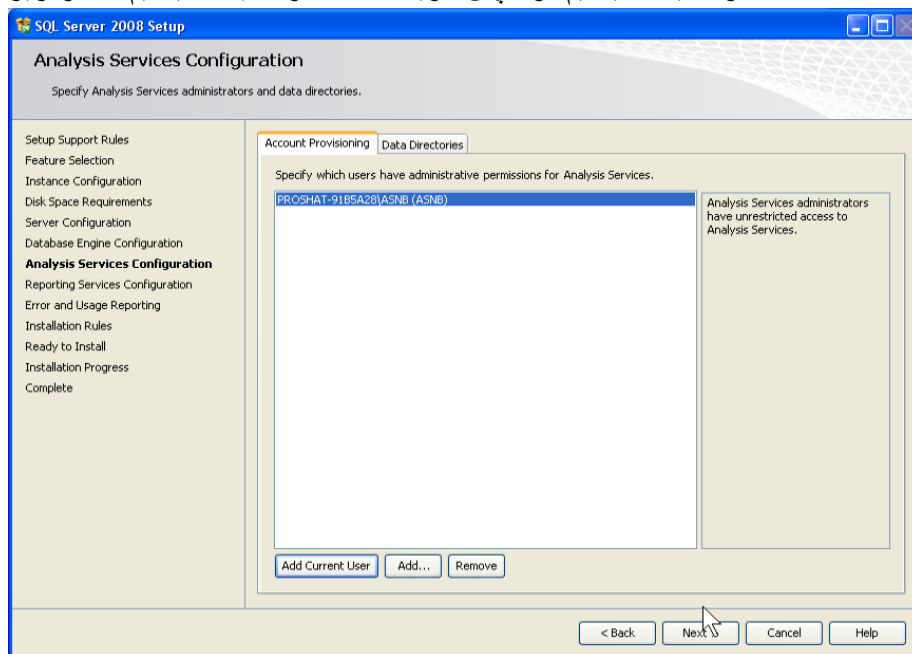
در کادر بالا دکمه Use the same account ... را کلیک میکنیم . کادر پایین ظاهر می شود.



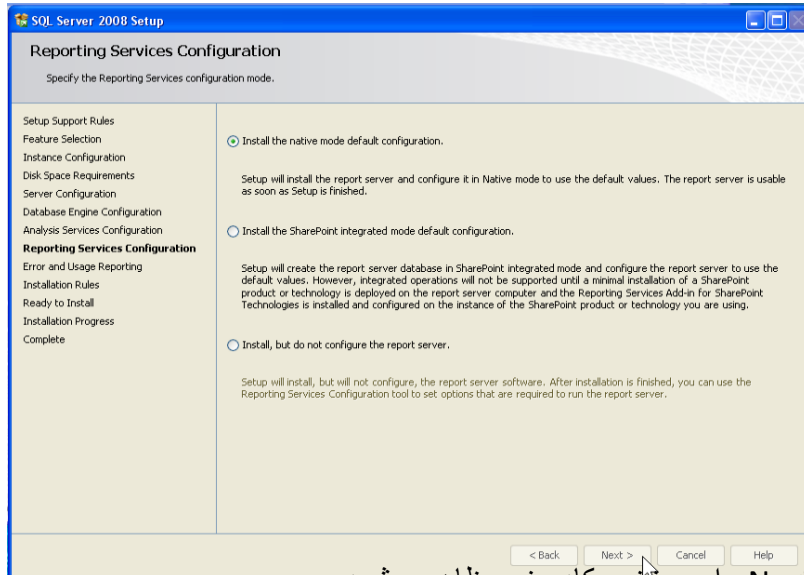
سپس کادر زیر ظاهر می شود .



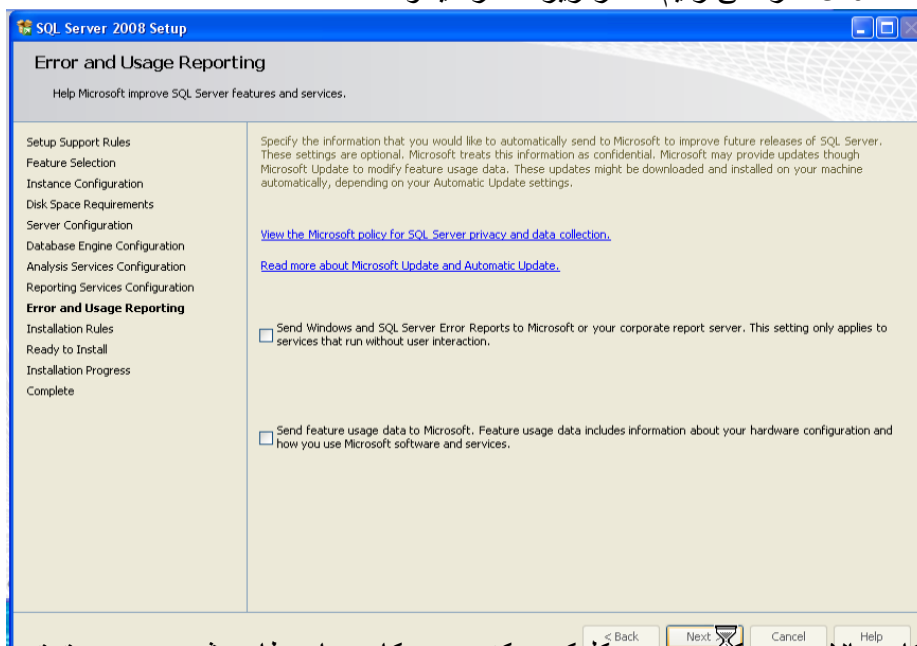
دکمه Add Current User را کلیک میکنیم. و سپس گزینه Next را کلیک میکنیم. کادر زیر ظاهر میشود .



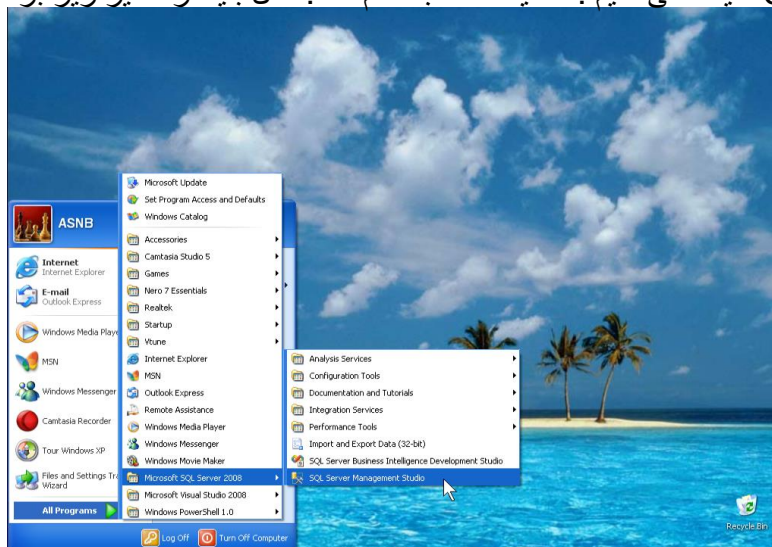
مجدداً " در بالا روی دکمه Add current User کلیک میکنیم و Next را کلیک میکنیم . کادر صفحه بعدی ظاهر میشود.



مجدداً " دکمه Next را می زنیم. کادر زیر ظاهر میشود.



مجدداً در کادر بالا روی دکمه Next کلیک میکنیم. در کادر های ظاهر شده بعدی به ترتیب Next و Close و Next را اجرا کنیم.

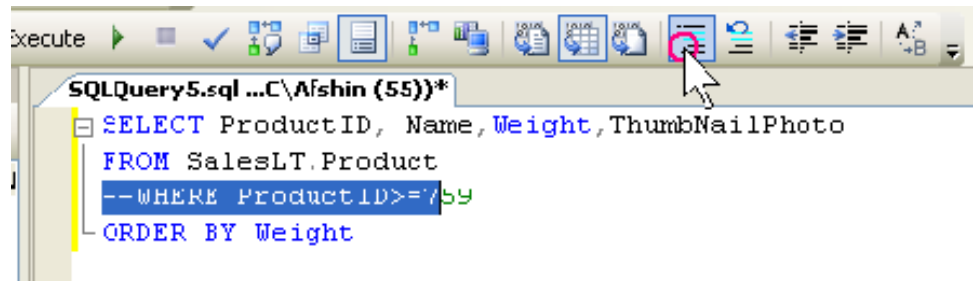


انواع داده ها در SQL SERVER 2008

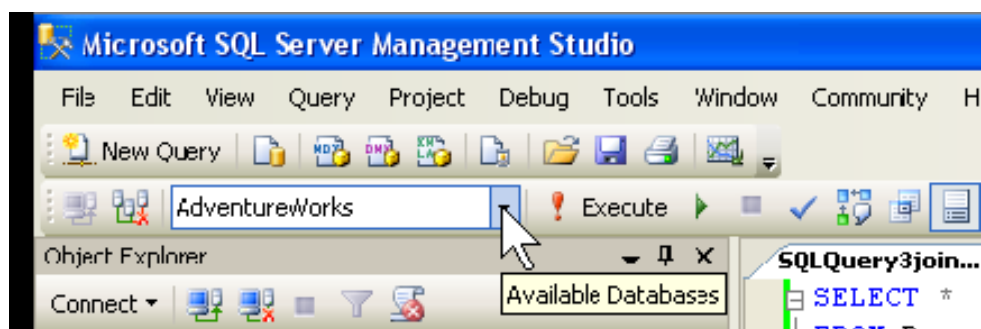
اندازه بر حسب بیت	نام نوع داده
1	Bit
8	Bigint
4	Int
2	SmallInt
1	TinyInt
--	Decimal
8	Money
4	SmallMoney
--	Float
8	DateTime
4	SmallDateTime
3	Date
5	Time

اندازه بر حسب بیت	نام نوع داده
1	Cursor
16	UniquelIdentifier
--	Char
--	VarChar
--	Text
--	Nchar
--	NVarChar
--	Ntext
--	Binary
--	VarBinary
--	Image
--	Table
--	XML

نحوه COMMENT کردن یک خط کد



نحوه تعیین پایگاه داده پیش فرض به صورت زیر است.



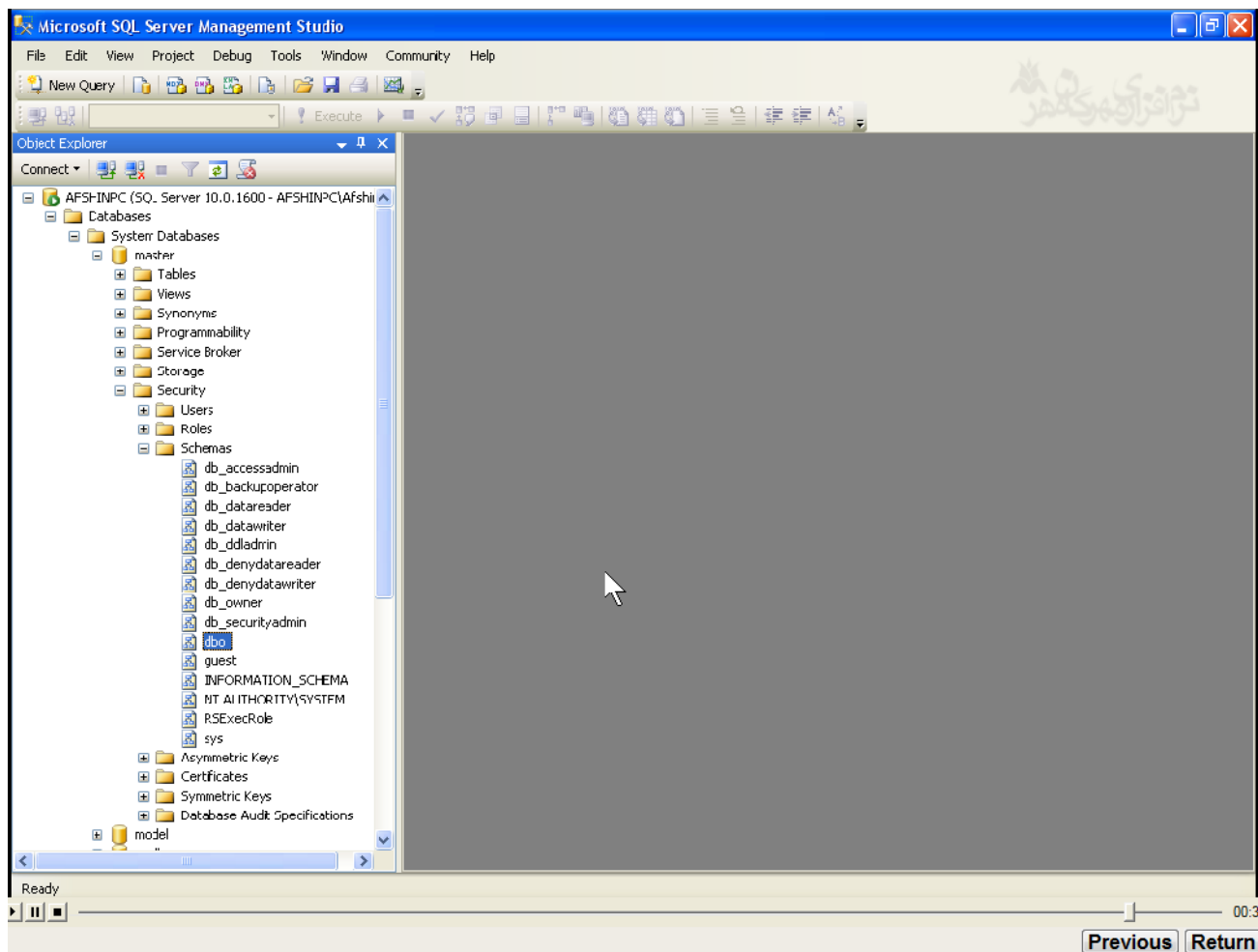
هنگامیکه برای اولین بار کار را با sql-server2008 شروع میکنید، کار با ۴ پایگاه داده سیستم شروع خواهد شد: MASTER , MODEL , MSDB , TEMPDB

برای ساخت هر شیء SQL SERVER از آدرس دهی زیر استفاده می شود .

[ServerName . [DatabaseName . [SchemaName .]]]ObjectName

قسمتها یی که داخل کروشه بیان شده اند ، یعنی اینکته اختیاری هستند .

برای ایجاد اشیاء در پایگاه داده از دستور زیر استفاده میشود . (حتی ایجاد خود database) هر شی ای که در پایگاه داده ایجاد کنیم ، با schema ی پیش فرض dbo فهرست خواهد شد ، نه با نام کاربردی ما .



CREATE <object type> <object name>

به مثال زیر دقت کنید :

```
create database Accounting
on
    (name='Accounting',filename='c:\Accounting.mdf',
    size=10,maxsize=50,filegrowth=5)
log on
    (name='AccountingLog',filename='c:\AccountingLog.ldf',
    size=5,maxsize=25,filegrowth=5)
go

exec sp_helpdb 'Accounting'
```

نکته: علاوه بر stored procedure sp_helpdb در مورد جداول sp_help داریم.

دستور بالا دو مجموعه نتیجه مجزا را اعلام میکند . اولین مورد اطلاعات ترکیب شده ، داده ها و log شده در مورد پایگاه داده است. Db-size مجموعه اندازه پایگاه داده و اندازه log است. مورد دوم مشخصه هایی را در مورد قسمتهای تشکیل دهنده پایگاه داده بیان میکند.

The screenshot shows a SQL query window with the command `exec sp_helpdb sama133` executed. Below the command, two tables are displayed. The first table shows database-level information for 'sama133', and the second table shows file-level information for the database.

name	db_size	owner	dbid	created	status	compatibility_level
sama133	2.81 MB	ali-PC\ali	9	Jul 27 2013	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	100

name	fileid	filename	filegroup	size	maxsize	growth	usage
sama133	1	L:\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQ...	PRIMARY	2304 KB	Unlimited	1024 KB	data only
sama133_log	2	L:\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQ...	NULL	576 KB	2147483648 KB	10%	log only

CREATE <object type> <object name>

برای ایجاد جدول به شکل زیر عمل می کنیم.

CREATE TABLE <object type> <object name>

CREATE TABLE Customers|

برای دیدن تعداد رکورد های جدول

Select count(*) from student

دستور بالا برای ایجاد جدول است. هر script ای که در یک پایگاه داده ایجاد می کنید ، باید به همراه دستور `use databasename` باشد .

```
use Accounting
create table Customers
(
    CustomerNo      int      identity  not null,
    CustomerName    varchar(30)  not null,
    Address1        varchar(30)  not null,
    Address2        varchar(30)  not null,
    city            varchar(20)  not null,
    state           char(2)      not null,
    zip             varchar(10)  not null,
    contact         varchar(25)  not null,
    phone           char(15)     not null,
    FedIDNo        varchar(9)   not null,
    DateInSystem   smalldatetime not null
)
```

The screenshot shows Microsoft SQL Server Management Studio with the following details:

- Object Explorer:** Shows the server 'AFSHINPC (SQL Server 10.0.1600 - AFSH)' with various databases listed, including 'Accounting'.
- Query Window:** Contains the SQL script: `exec sp_help Customers`.
- Results:** Displays the output of the `sp_help` command for the 'Customers' table.

Name	Owner	Type	Created_datetime
Customers	dbo	user table	2010-01-04 05:20:13.733

Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource	Collation
CustomerNo	int	no	4	10	0	no	(n/a)	(n/a)	NULL
CustomerName	varchar	no	30			no	no	no	SQL_Latin1_Genera
Address1	varchar	no	30			no	no	no	SQL_Latin1_Genera
Address2	varchar	no	30			no	no	no	SQL_Latin1_Genera
city	varchar	no	20			no	no	no	SQL_Latin1_Genera
state	char	no	2			no	no	no	SQL_Latin1_Genera
zip	varchar	no	10			no	no	no	SQL_Latin1_Genera
contact	varchar	no	25			no	no	no	SQL_Latin1_Genera

Identity	Seed	Increment	Not For Replication
CustomerNo	1	1	0

RowGUIDCol
No rowguidcol column defined.

Data_located_on_filegroup
PRIMARY

Query executed successfully. AFSHINPC (10.0 RTM) AFSHINPC\Afshin (54) Accounting 00:00:01 1 rows

```

use Accounting
create table Employees
(
    EmployeeID int identity not null,
    FirstName varchar(25) not null,
    MiddleInitial char(1) null,
    LastName varchar(25) not null,
    Title varchar(25) not null,
    SSN varchar(11) not null,
    Salary money not null,
    PriorSalary money not null,
    LastRaise AS Salary - PriorSalary,
    HireDate smalldatetime not null,
    TerminationDate smalldatetime null,
    ManagerEmpID int not null,
    Department varchar(25) not null
)

```

به فیلد LastRaise دقت کنید.

لقب یا سمت کارمند ←

شماره تامین اجتماعی ←

تاریخ استخدام ←

می توان sp_help را روی این جدول اجرا کرد تا ببینیم این جدول به درستی اعمال شده است یا خیر ؟
 برای تغییر جدول از دستور alter به شکل زیر استفاده میکنیم . شکل فرمان آن بسیار شبیه به create است.

ALTER <object type><object name>

```

alter table Employees
Add
    PreviuseEmployer varchar(30) null

```

مثال :

حتی میتوان توسط دستور alter چندین سطر را نیز اضافه کرد . توسط دستور drop میتوان یک شیء را حذف کرد .

DROP <object type> <object name>

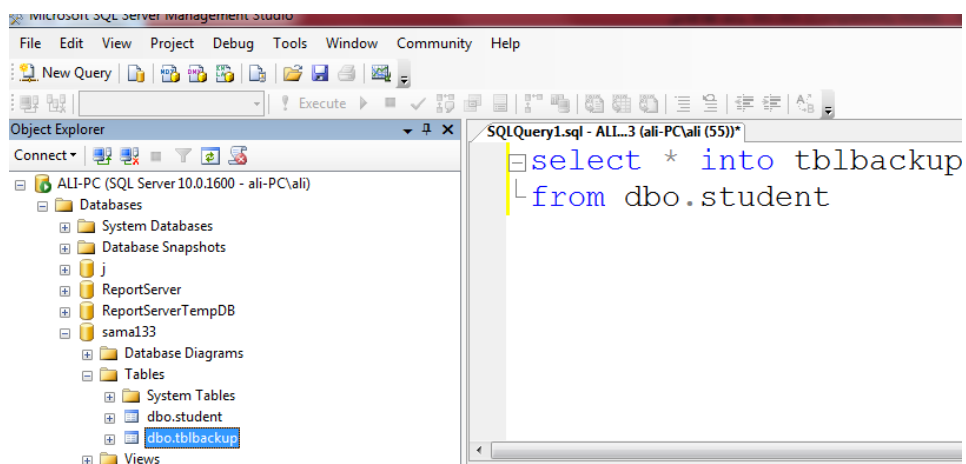
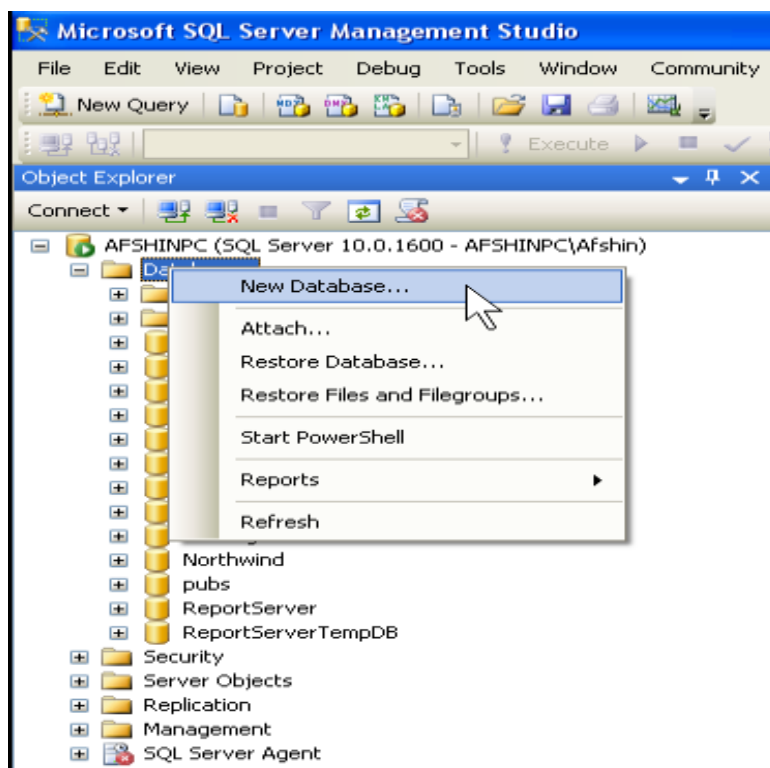
```

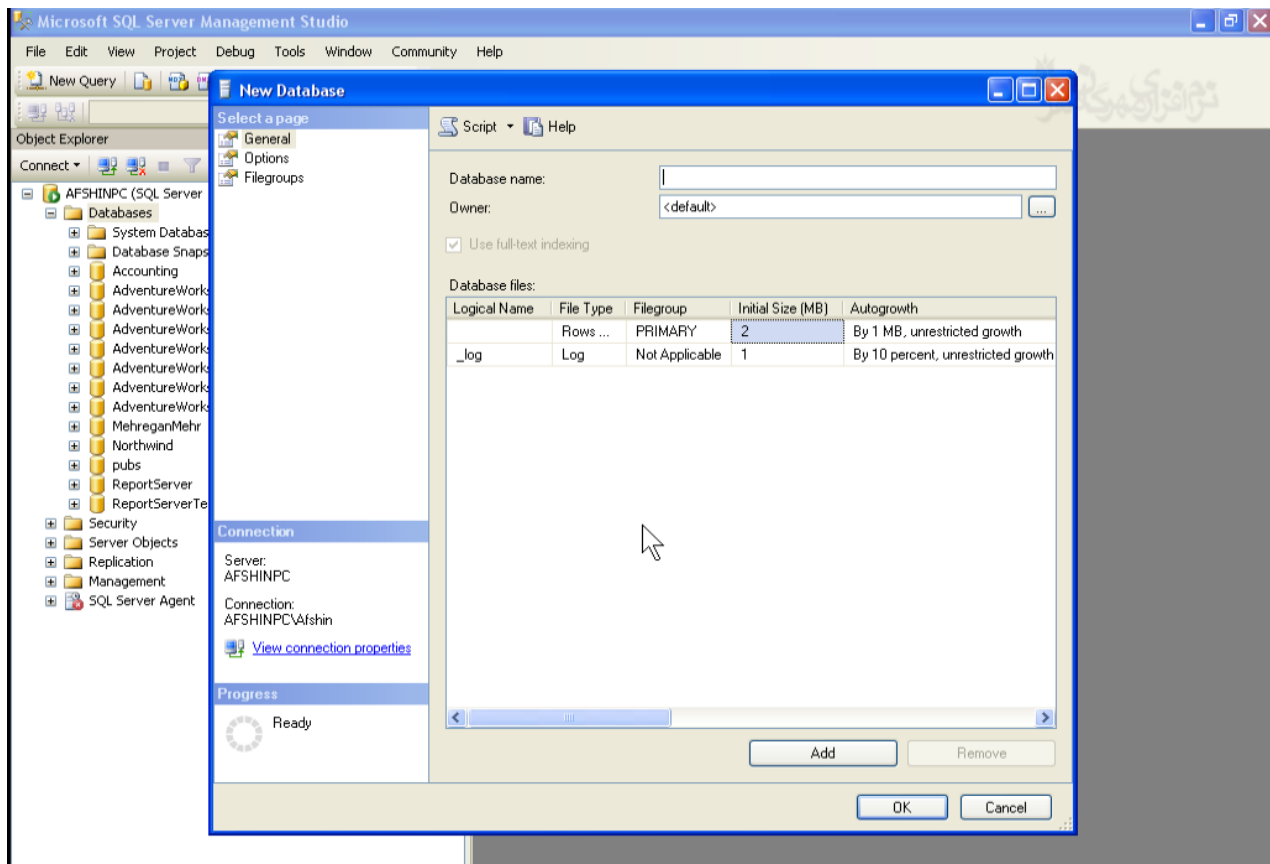
use Accounting
drop table Customers

```

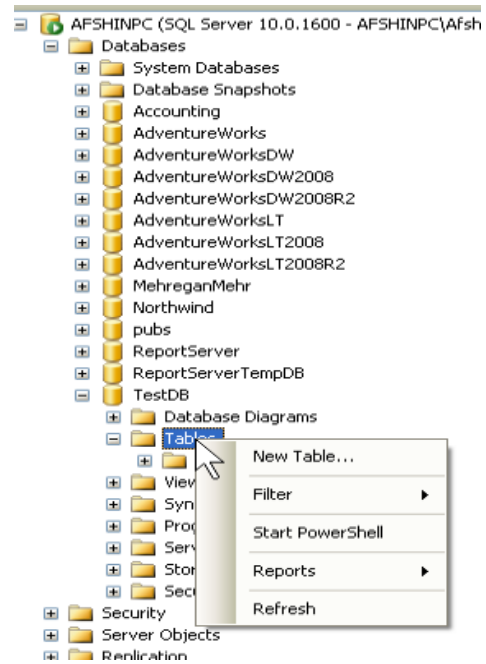
Drop table dbo.customers

در مثال بالا توسط دستور drop جدول customers را پاک کردیم .
نکته : موقع پاک کردن شیء مورد نظر هیچ سئوالی مبنی بر این که آیا می خواهید پاک کنید یا خیر از ما پرسیده نمیشود.
برای پاک کردن database هم میتوان از دستور drop استفاده کرد .
برای حذف database هم از دستور drop استفاده می کنیم .
برای ایجاد database از طریق گرافیکی روی گزینه database کلیک راست کرده و گزینه New database را به شکل زیر انجام می دهیم .

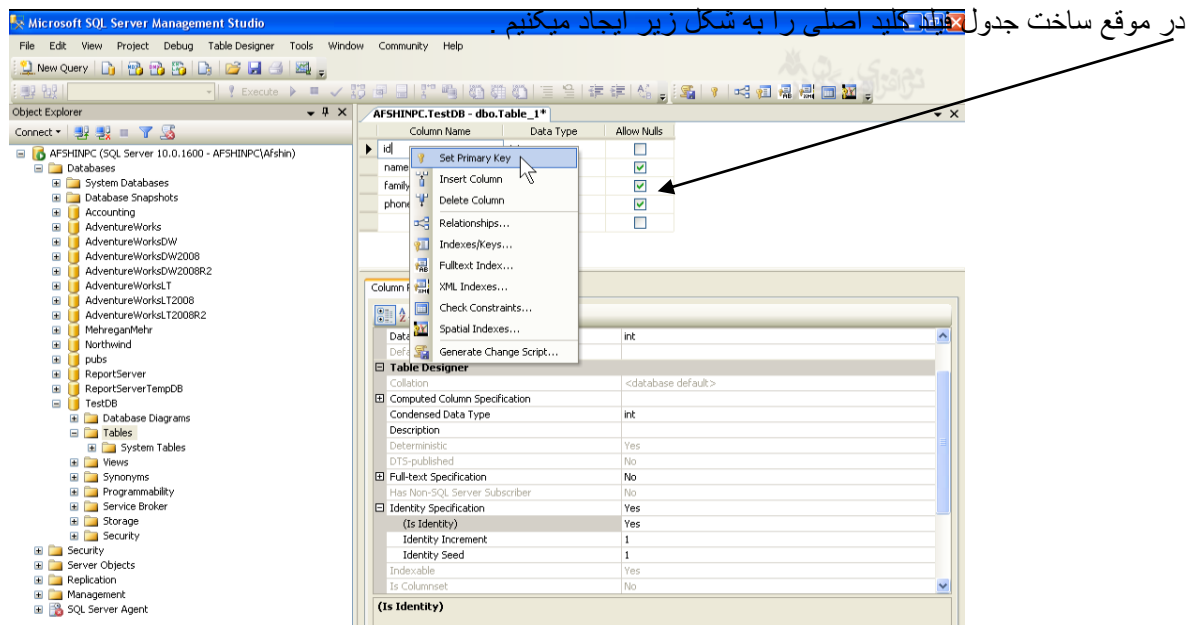
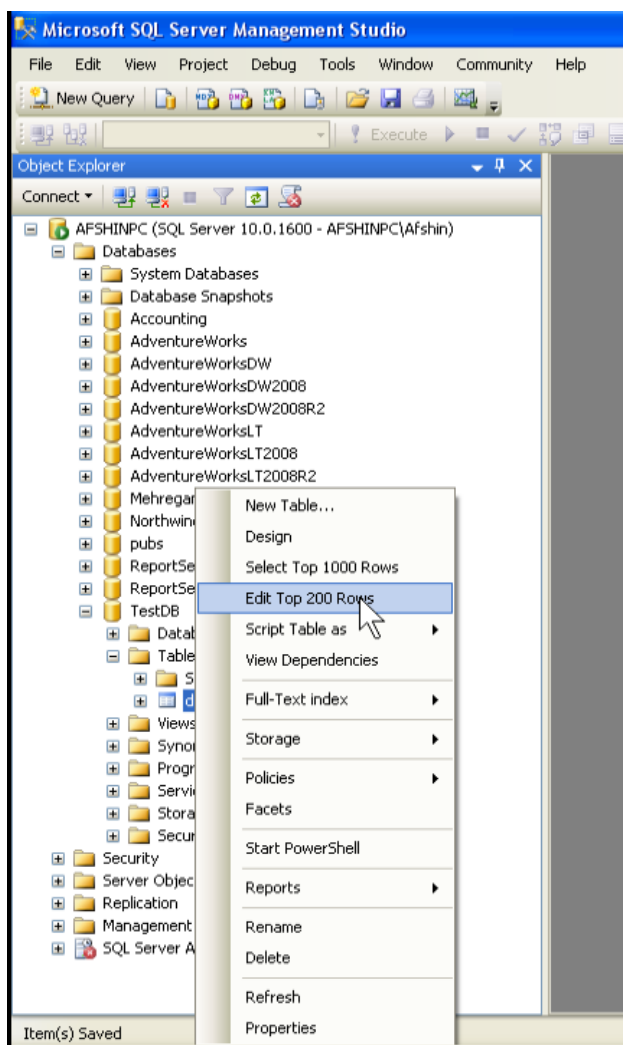




سپس کادر بالا ظاهر میشود. نام database را در کادر database name مینویسیم و OK را می زنیم. برای ایجاد جدول میتوان روی گره Table کلیک راست کرده و گزینه New Table را بزیم.



برای ورود اطلاعات به جدول به شکل زیر داریم .



برخی از دستورات مهم SQL

1) CREATE DATABASE

مثال : CREATE DATABASE database_name

این دستور یک پایگاه داده جدیدی می سازد و میتواند دارای پارامتر های متعددی باشد : مثلاً "اندازه ابتدایی database و محل ساخته شدن آن .

2) CREATE TABLE

```
CREATE TABLE table_name  
(fild1 datatype [NOT NULL],  
fild2 datatype [NOT NULL],...)
```

این دستور یک جدول جدید را در پایگاه داده ها ایجاد می کند.
نام و نوع داده های اجزای جدول نیز در این دستور مشخص می شوند.

3) CREATE USER

```
CREATE USER user
```

این دستور یک کاربر جدید را می سازد و نام و رمز عبور اولیه وی را مشخص می کند.

4) DELETE

```
DELETE FROM table_name  
WHERE condition
```

این دستور داده ها را از داخل بانک اطلاعاتی پاک می کند. به هنگام استفاده از این دستور باید توجه داشت که:

اولاً این دستور بلافاصله پس از ورود اجرا می شود و منتظر تأیید کاربر نمی ماند.
ثانیاً دستور DELETE تنها رکوردها را حذف می کند و قادر به حذف جدول نیست.
برای حذف جدولها باید از دستور DROP TABLE استفاده کنید.

5) DROP DATABASE

```
DROP DATABASE database_name
```

این دستور پایگاه داده ها را کاملاً حذف می کند به نحوی که ساختار فیزیکی پایگاه داده ها و داده های آن به کلی پاک خواهند شد.

6) DROP TABLE

```
DROP TABLE table_name
```

این دستور یک جدول را از پایگاه داده ها حذف می کند.

7) FROM

FROM <table>[,<table>...]

این دستور جدولهای مورد استفاده و یا تلفیق شده را مشخص می‌کند.

8) GROUP BY

GROUP BY <col> [,<col>...]

این دستور همه سطرهایی را که دارای یک مقدار ستون واحد هستند، با یکدیگر دسته‌بندی می‌کند.

9) HAVING

HAVING <search_condition>

این کلمه کلیدی تنها همراه با GROUP BY به کار می‌رود و باعث می‌شود که گروه انتخابی توسط شرط مشخص شده، محدود گردد.

10) INSERT

این دستور امکان اضافه کردن داده‌ها را به بانک اطلاعاتی فراهم می‌آورد و دارای دو شکل کلی است:

INSERT VALUES

-۱

INSERT INTO table_name (col 1, col 2,...)
VALUES(value 1, value 2,...)

این دستور، داده‌ها را به صورت یک رکورد جدید به یک جدول اضافه می‌کند. در عملیاتی که تنها از چند رکورد محدود سروکار دارند، استفاده از این دستور مفید است.

INSERT ... SELECT

-۲

INSERT INTO table_name (col 1, col 2,...)
SELECT col 1, col 2,...
FROM table_name
WHERE search_condition

در مواردی که رکوردهای بسیار زیادی می‌بایست به یک جدول اضافه گردند استفاده از این دستور مطلوب‌تر است. برنامه‌نویسان می‌توانند با به‌کارگیری این دستور، اطلاعات را از یک یا چند جدول استخراج کنند و در جدول دیگری درج نمایند.

11) ORDER BY

ORDER BY <order_list>

این دستور مقادیر برگشتی را بر حسب ستون (یا ستونهای) مشخص‌شده، مرتب می‌کند. ASC یعنی مرتب سازی به صورت صعودی و DESC یعنی مرتب سازی به شکل نزولی

12) GRANT, REVOKE

GRANT role TO user

REVOKE role FROM user

دستور GRANT یک اختیار یا یک نقش را به کاربری که با دستور GRANT USER ساخته شده است اعطا می‌کند و دستور REVOKE آنرا بازپس می‌گیرد. از این دستورات برای اعطا اختیارات سیستمی و نقشها و نیز دادن اجازه‌های دستیابی به ساختارهای داده‌ای در یک بانک اطلاعاتی و یا بازپس‌گیری آنها استفاده می‌شود. این دستورات را فرامین (Data Control Language) DCL می‌نامند.

13) SELECT

SELECT [DISTINCT | ALL]

همه دستورات بازیابی داده‌ها با این دستور آغاز می‌شوند. به‌کارگیری کلمه کلیدی DISTINCT باعث می‌شود تا تنها مقادیر یکتا بازگردانده شوند و از بازگرداندن مقادیر تکراری اجتناب شود. کلمه کلیدی ALL به‌عنوان پیش‌فرض به‌کار می‌رود و در صورت استفاده از آن، مقادیر تکراری نیز مجاز خواهند بود.

14) UPDATE

UPDATE table_name

SET column_name 1 = value 1 [column_name 2 = value 2 ,...]

WHERE search_condition

این دستور مقادیر رکوردهای موجود را تغییر می‌دهد و داده‌ها را در داخل بانک اطلاعاتی به‌روز می‌کند.

15) WHERE

WHERE <search_condition>

این دستور سطرهای برگشتی را توسط شرط محدود می‌کند.

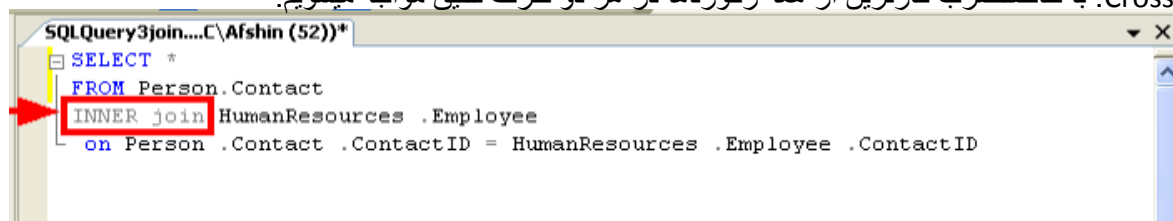
انواع join به شرح زیر هستند (join اطلاعات دو جدول را در یک جدول گرد هم می‌آورد).

الف) inner join : رکورد هایی را بر میگرداند که مطابق فیلد یا فیلدهایی باشد که ما خواسته ایم.

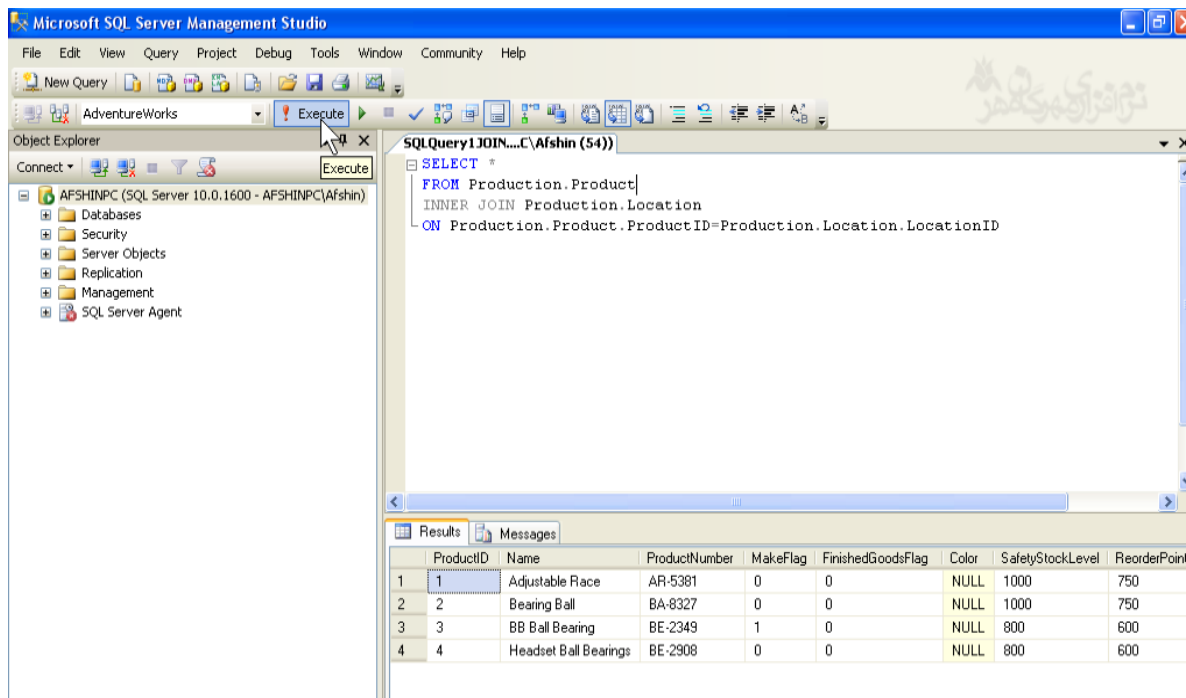
ب) outer join(left or right)

ج) Full join : هنگامی استفاده کنید که تمام رکورد ها را از هر دو طرف تلفیق میخواهید.

د) Cross Join : با حاصلضرب کارتیزین از همه رکوردها در هر دو طرف تلفیق مواجه میشویم.



```
SQLQuery3join....C:\Afshin (52))*
SELECT *
FROM Person.Contact
INNER join HumanResources.Employee
on Person.Contact.ContactID = HumanResources.Employee.ContactID
```



Microsoft SQL Server Management Studio

```
SQLQuery1JOIN....C:\Afshin (54)
SELECT *
FROM Production.Product
INNER JOIN Production.Location
ON Production.Product.ProductID=Production.Location.LocationID
```

ProductID	Name	ProductNumber	MakeFlag	FinishedGoodsFlag	Color	SafetyStockLevel	ReorderPoint
1	Adjustable Race	AR-5381	0	0	NULL	1000	750
2	Bearing Ball	BA-8327	0	0	NULL	1000	750
3	BB Ball Bearing	BE-2349	1	0	NULL	800	600
4	Headset Ball Bearings	BE-2908	0	0	NULL	800	600

```
SQLQuery2.sql - ALL...a (ali-PC\ali (57))* SQLQuery1.sql - ALL...a (ali-PC\ali (56))*
select student.code , student.nam , student.family , mark.vb , mark.access
from student inner join mark
on student.code=mark.code
```

code	nam	family	vb	access
1	1000	ali	hasani	Click to select the whole column
2	2000	alireza	hoseini	19 13
3	2000	alireza	hoseini	19 13

به خروجی ها دقت شود

```
SQLQuery2.sql - ALL...a (ali-PC\ali (57)) SQLQuery1.sql - ALL...a (ali-PC\ali (56))
select student.code , student.nam , student.family , mark.vb , mark.access
from student left join mark
on student.code=mark.code
```

code	nam	family	vb	access
1	1000	ali	hasani	18 20
2	2000	alireza	hoseini	19 13
3	2000	alireza	hoseini	19 13
4	3000	hamid	alzade	NULL NULL
5	5000	ali	javadi	18 19
6	6000	abbas	mobini	NULL NULL

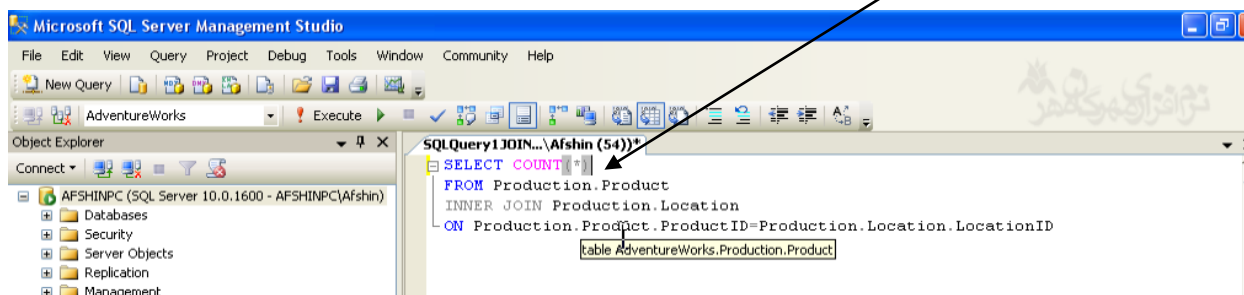
```
SQLQuery2.sql - ALL...a (ali-PC\ali (57))* SQLQuery1.sql - ALL...a (ali-PC\ali (56))*
select student.code , student.nam , student.family , mark.vb , mark.access
from student right join mark
on student.code=mark.code
```

code	nam	family	vb	access
1	1000	ali	hasani	18 20
2	2000	alireza	hoseini	19 13
3	2000	alireza	hoseini	19 13
4	5000	ali	javadi	18 19
5	NULL	NULL	NULL	12 16

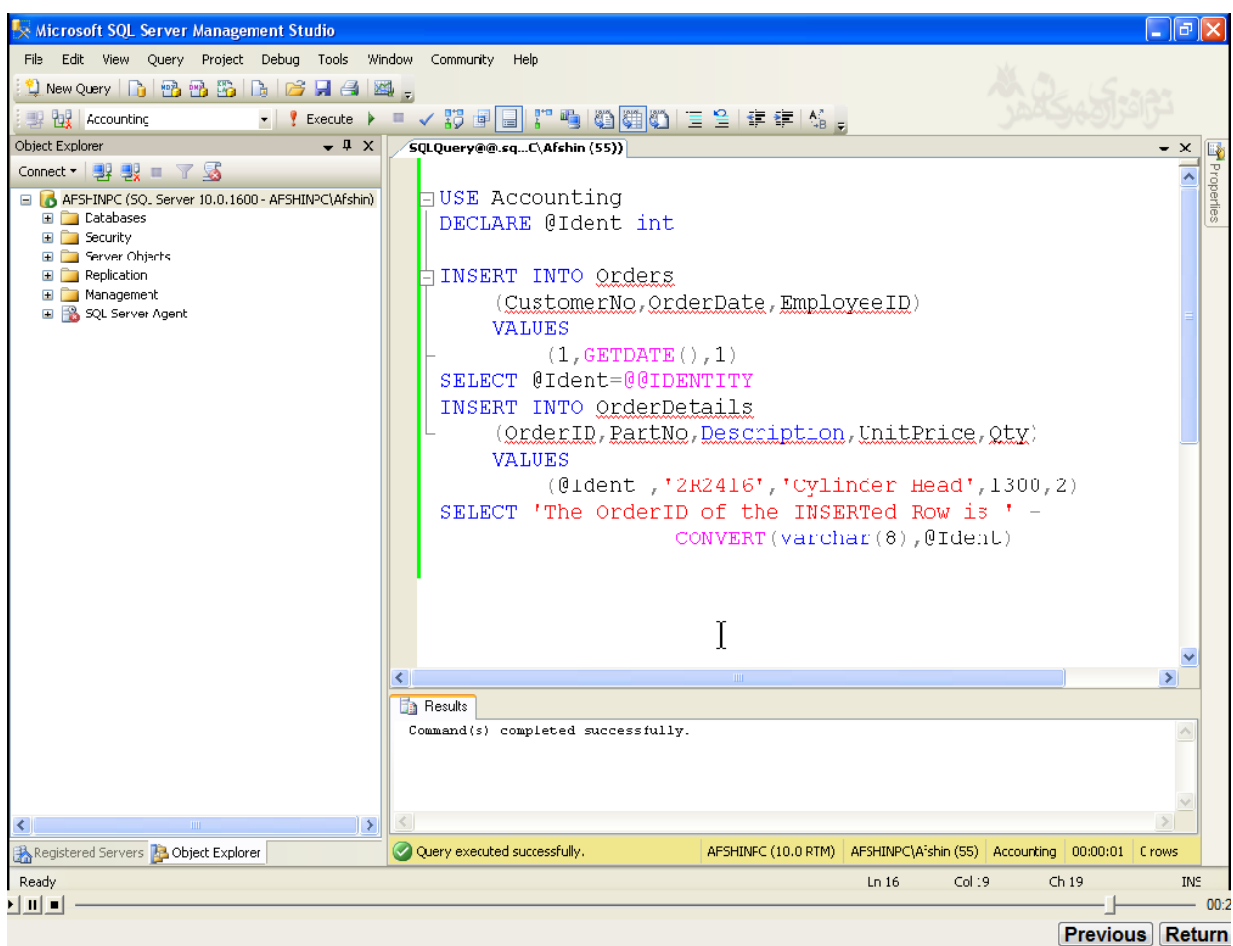
فرم کلی دستورات join به شکل زیر است:

```
--SELECT <SELECT list>  
  
--FROM <LEFT table>  
  
--<LEFT | RIGHT> [ OUTER ] JOIN <RIGHT table>  
  
--ON < join condition >
```


میتوان با استفاده از تابع Count تعداد رکوردهای استخراج شده را بدست آوریم .

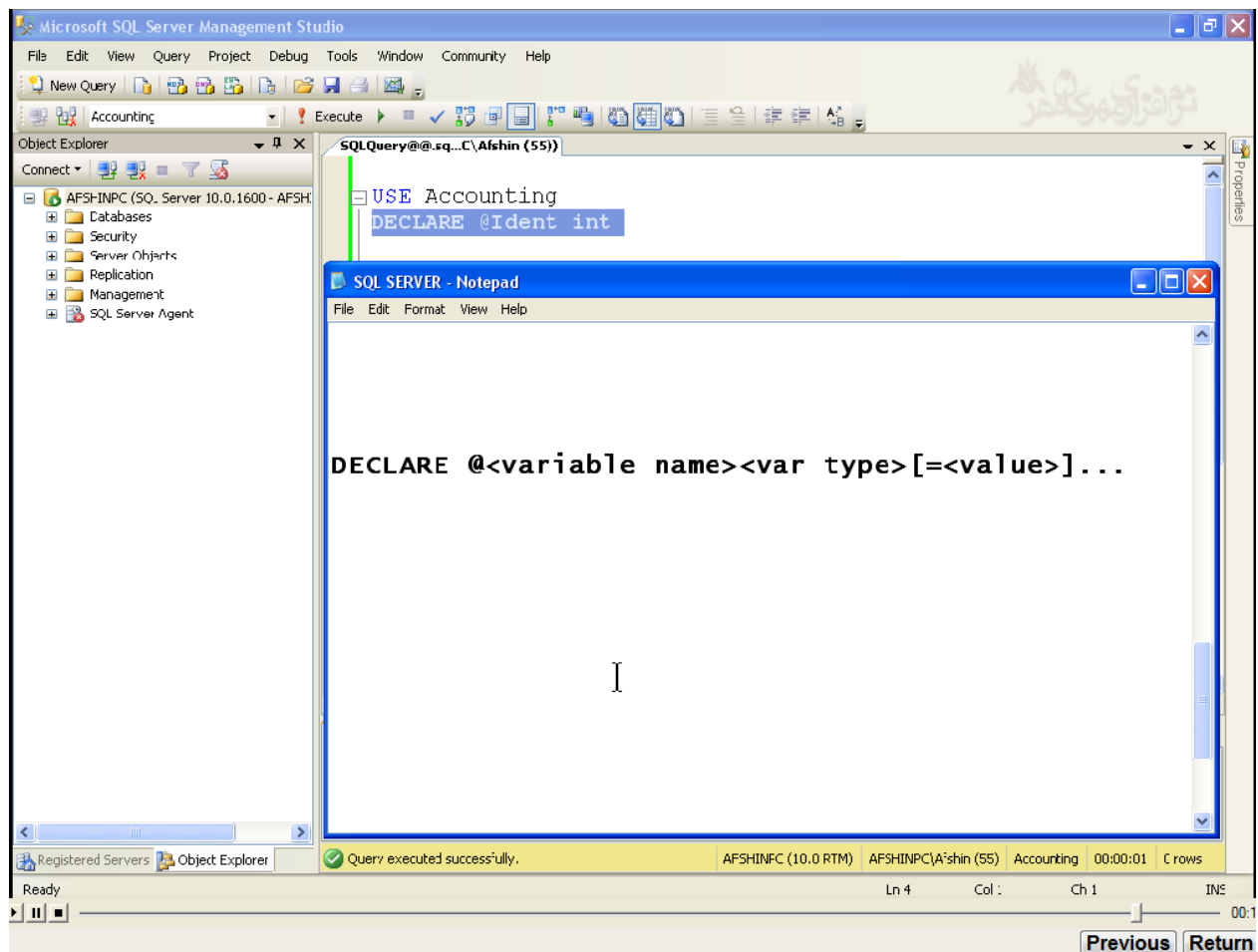


Script نویسی : script ها معمولاً دارای یک هدف معینی هستند یعنی تمام فرامین یک script طبق یک هدف کلی نوشته میشوند. به script زیر توجه کنید . اسکرپت ها در SQL به صورت فایل های متنی ذخیره میشوند.

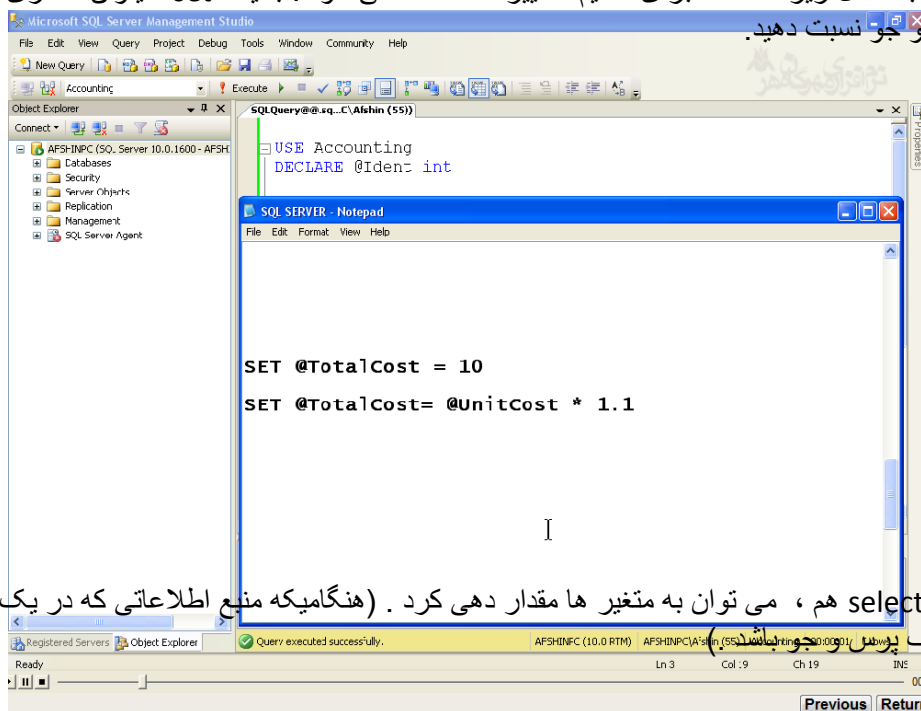


@@identity یک تابع سیستمی است که آخرین مقدار یکتایی را تعیین می کند که در اتصال جاری نسبت داده شده است.

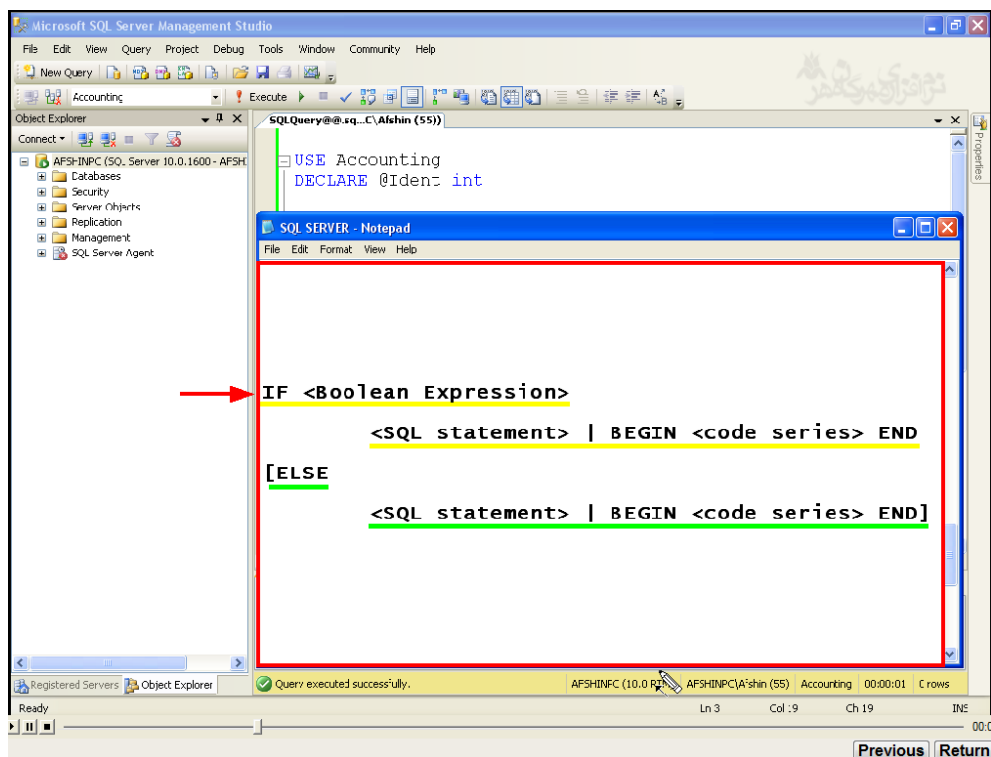
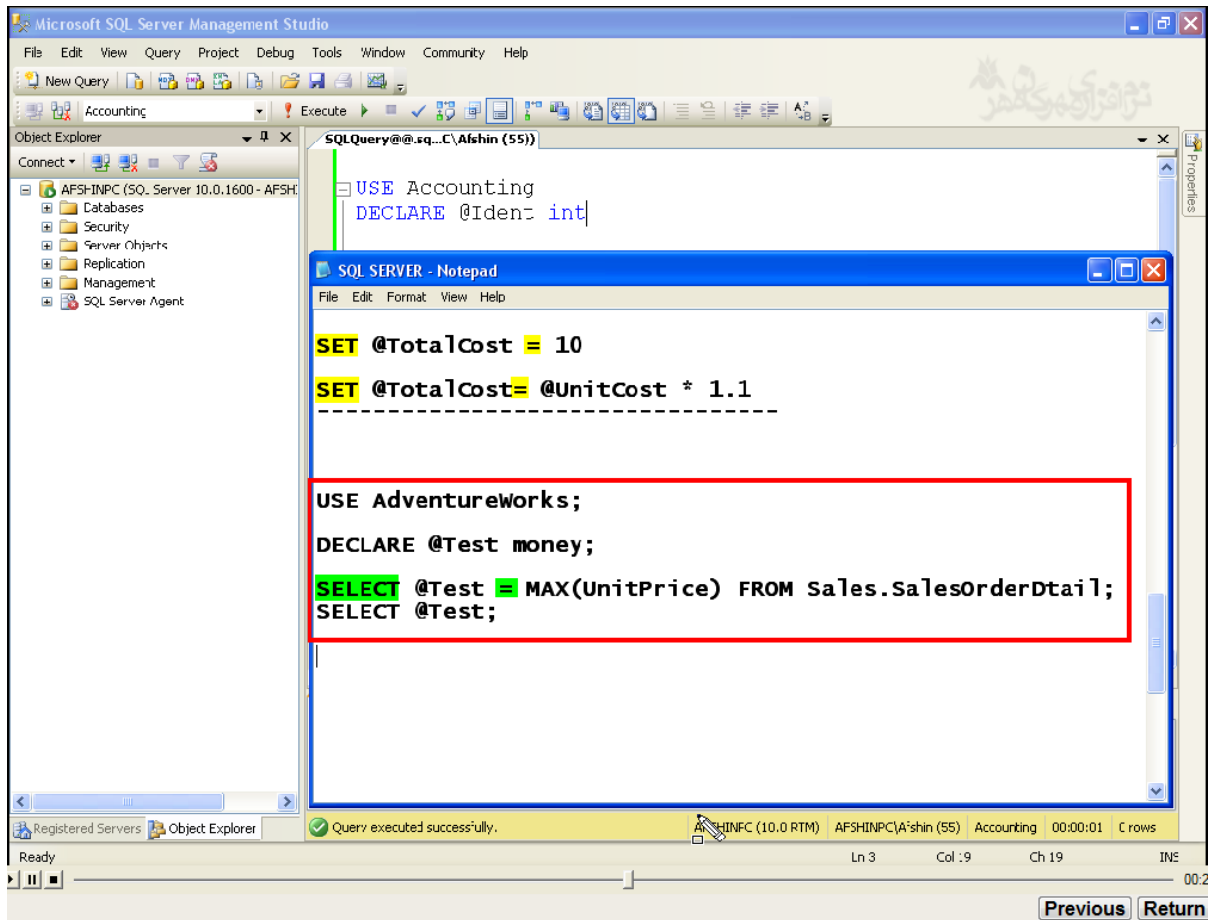
نکته : برای تعریف متغیر از عبارت declare به شکل زیر استفاده می کنیم :



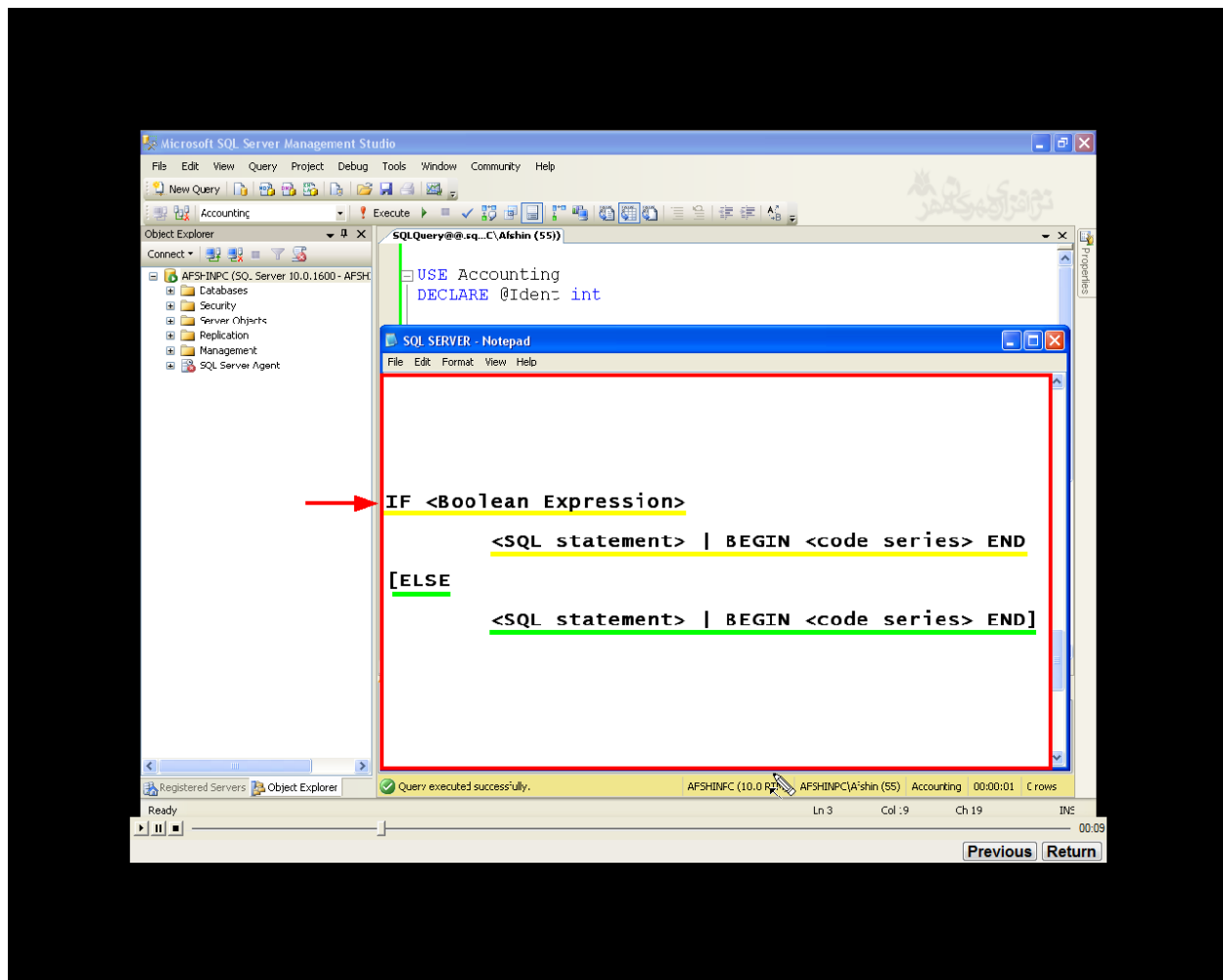
فرم کلی دستور set به شکل زیر است که برای تنظیم متغیرها استفاده می شود. با یک set نمیتوان مقداری را به یک متغیری از یک پرس و جو نسبت دهید.



با استفاده از دستور select هم، می توان به متغیرها مقدار دهی کرد. (هنگامیکه منبع اطلاعاتی که در یک متغیر ذخیره کرده اید از یک پرس و جو باشد)



عبارات کنترل جریان :



```
WHILE <Boolean expression>
    <sql statement>
[BEGIN
    <statement block>
    [BREAK]
    <sql statement>|<statement block>
    [CONTINUE]
END]
```

اغلب مواردی اتفاق می افتد که نمیتوانید جلوی اتفاق افتادن آنرا بگیرید ولی از طرف دیگر نمیخواهید منتظر اجرای چیزی بمانید. در این صورت از عبارت WAITFOR استفاده میشود. delay مقدار زمان انتظار را مشخص میکند.

WAIFOR

DELAY <'time'>|TIME<'time'>

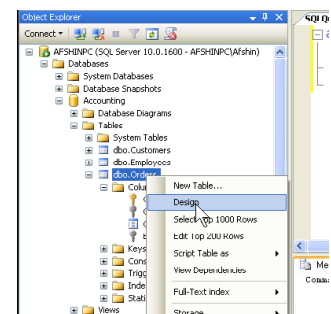
محدودیت یا constraint یک حد و مرز است. محدودیتی که در سطح یک ستون یا جدول اعمال میشود اطمینان حاصل میکند که داده‌ها قوانین جامعیت داده‌های خاص را تامین میکند. جامعیت داده‌ها بر عهده برنامه‌ای که از پایگاه داده‌ها استفاده میکنند نیست بلکه مسئولیت خود پایگاه داده‌هاست.

```
use Accounting
create table Customers
(
    CustomerNo int identity not null
    primary key,
    CustomerName varchar(30) not null,
    Address1 varchar(30) not null,
    Address2 varchar(30) not null,
    city varchar(20) not null,
    state char(2) not null,
    zip varchar(10) not null,
    contact varchar(25) not null,
    phone char(15) not null,
    FedIDNo varchar(9) not null,
    DateInSystem smalldatetime not null
)
```

```
use Accounting
create table Orders
(
    OrderID int identity not null
    primary key,
    CustomerNo int not null
    foreign key references Customers(CustomerNo),
    OrderDate smalldatetime not null,
    EmployeeID int not null
)
```

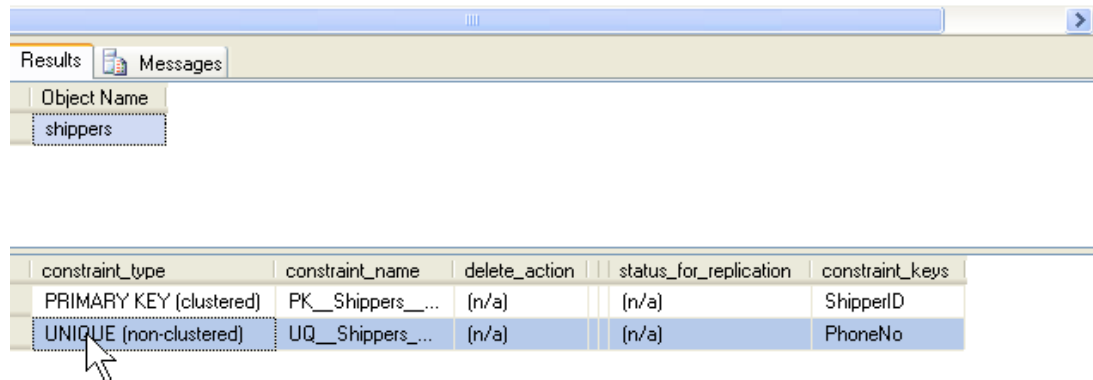
```
alter table orders
add constraint fk_EmployeeCreatesOrder
foreign key (EmployeeID) references Employees(EmployeeID)
```

همچنین میتوان یک جدول را در حالت ویژوالی ببرید و design آنرا تغییر دهید.



Sphep_constraint به ما اطلاعاتی از جدول شامل کلید اصلی و unique را نشان میدهد.

```
exec sp_helpconstraint shippers
```



constraint_type	constraint_name	delete_action	status_for_replication	constraint_keys
PRIMARY KEY (clustered)	PK_Shippers_...	(n/a)	(n/a)	ShipperID
UNIQUE (non-clustered)	UQ_Shippers_...	(n/a)	(n/a)	PhoneNo

```
alter table Customers  
add constraint CN_CustomerDateInSystem  
check  
(DateInSystem <= Getdate ())
```

View ها

دیدگاه ها چیزی جز یک پرس و جوی ذخیره شده نیستند. آنچه مهم است این است که میتوانید داده هایی از جدول پایه یا دیدگاههای دیگر را ترکیب کرده و تطبیق دهید تا جدولی ایجاد شود که درست همانند جدول پایه دیگری عمل میکند ساختار دستوری یک دیدگاه به شکل زیر است.

```
CREATE VIEW <view name>
```

```
AS
```

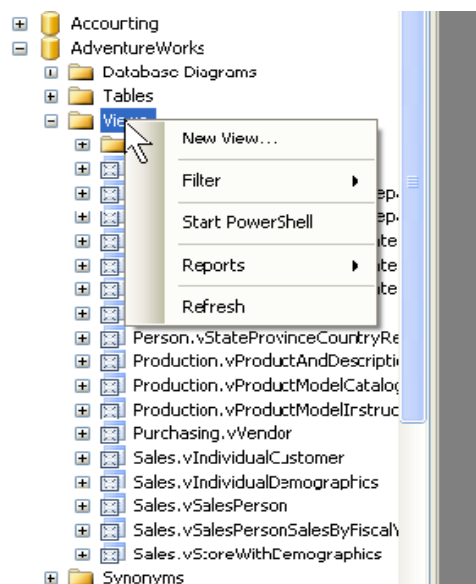
```
<SELECT statement>
```

```
use Accounting  
go  
create view CustomerPhoneList_vw  
AS  
select CustomerName, Contact, Phone  
FROM Customers;
```

مثال : ایجاد یک دیدگاه مربوط به لیست تلفن مشتریان است و آنرا به صورت CustomerPhoneList در پایگاه داده Accounting ایجاد میکنیم.

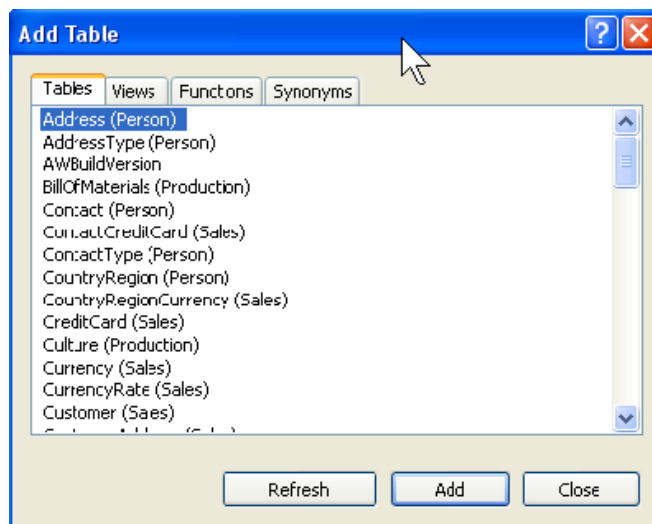
برای حذف یک View از ساختار زیر استفاده میشود:

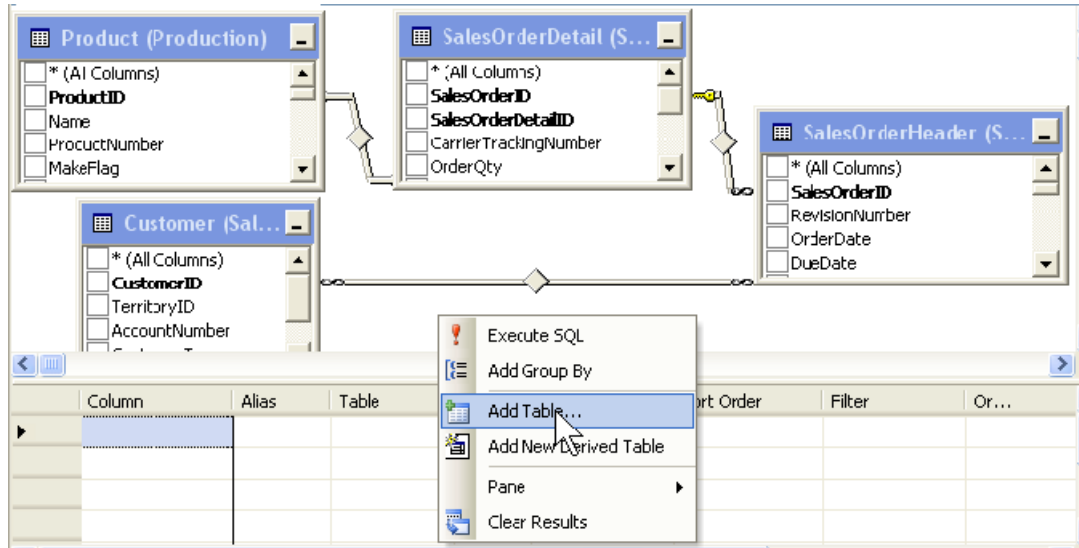
DROP VIEW <view name>



طریقه ساخت view از طریق گرافیکی

سپس پنجره add table به صورت زیر نمایش داده میشود.

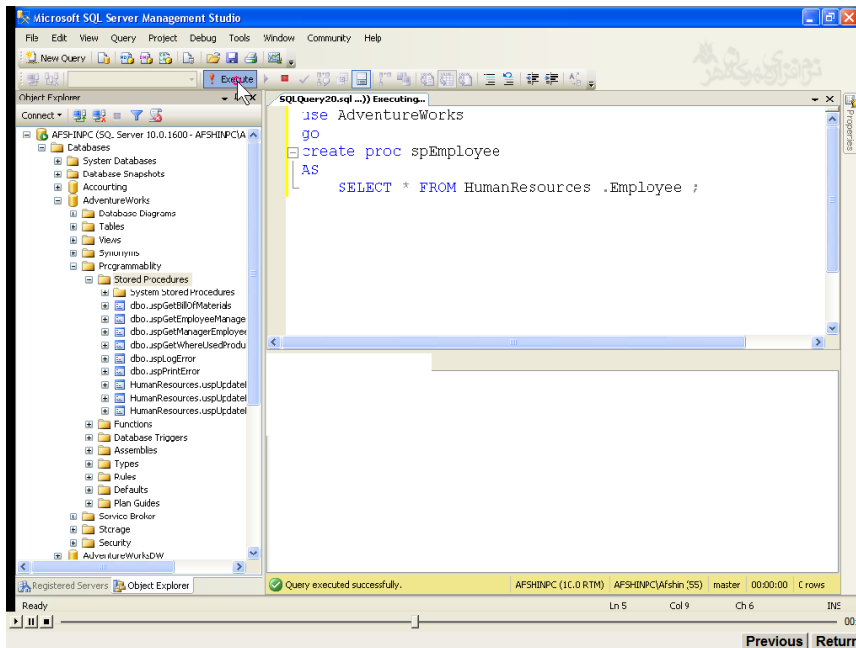




با کلیک راست در پنجره diagram و انتخاب گزینه add table و یا کلیک کردن روی دکمه نوار ابزار add table میتوان جدول دیگری را اضافه کرد.

Stored procedure ها

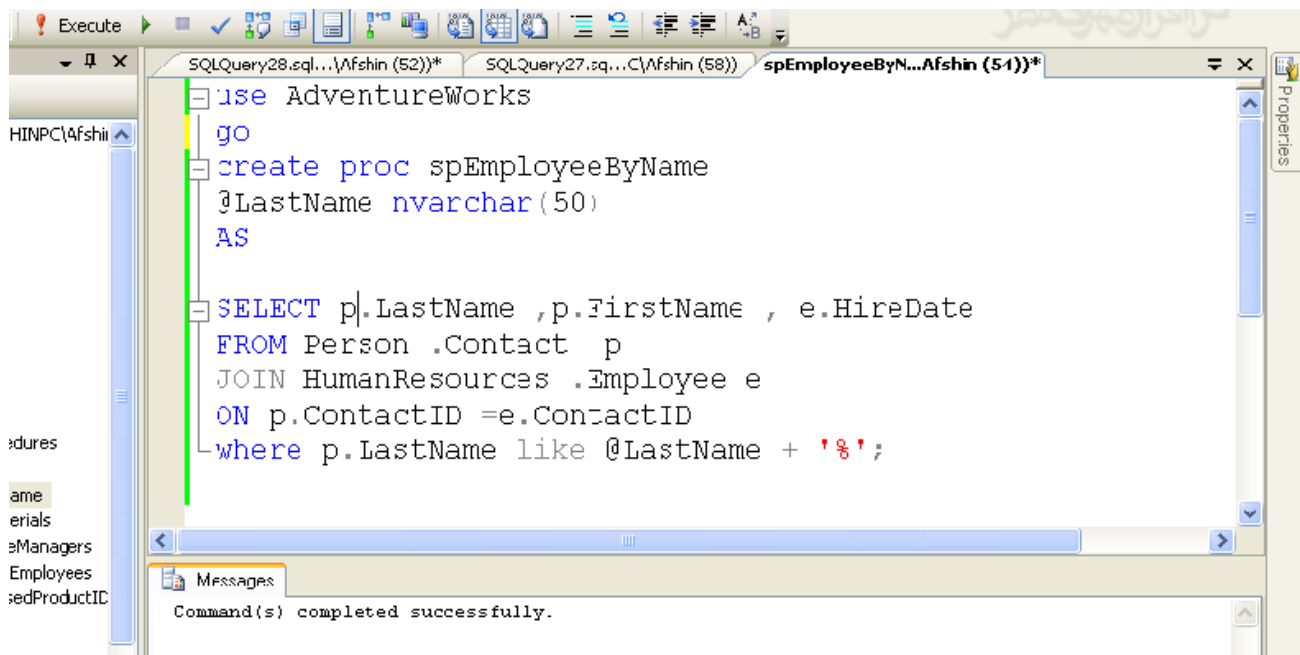
ویژگی های stored procedure ها عبارتست از کارایی بهتر ، لایه امنیتی ، کد قابل استفاده مجدد تفکیک کد و... می باشد.



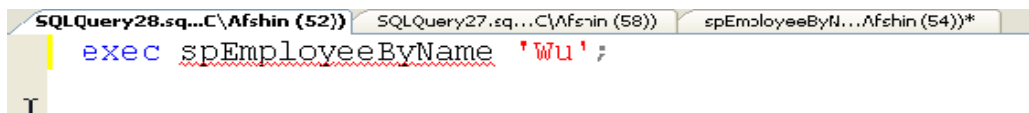
نحوه تعریف پارامتر به شکل زیر است:

@parameter_name[AS] datatype [=default|NULL]

نکته : نام پارامتر حتماً باید با علامت @ شروع شود.



برای اجرای stored procedure از دستور exec که به همراه آن نام stored procedure و پارامتر ورودی وجود دارد ، به صورت زیر استفاده میشود.



برای ارسال مقدار برگشتی از stored procedure به کد فراخوانی از return به صورت زیر استفاده میشود.

RETURN [<integer value to return>]

```

use AdventureWorks ;
go

create proc spTestReturns
AS
declare @MyMessage varchar(50);
declare @MyOtherMessage varchar(50);

select @MyMessage = 'Hi, this line before RETURN...';
print @MyMessage;
return;
select @MyOtherMessage = 'Sorry!';
print @MyOtherMessage;
return;

```

مثال از یک stored procedure که برای اجرای نتیجه حاصل نیاز به script میباشد.

برای اجرای آن به صورت زیر عمل می کنیم:

The screenshot shows a SQL Server Enterprise Manager interface. At the top, there are two tabs: 'SQLQuery32.sql... \Afshin (56))*' and 'SQLQuery29.sql... \Afshin (54))*'. The main window contains the following SQL code:

```

declare @return int;

exec @return = spTestReturns;

select @return ;

```

Below the code editor, there are two tabs: 'Results' and 'Messages'. The 'Messages' tab is active, displaying the message: 'Hi, this line before RETURN..' followed by a cursor. Below the message, it says '(1 row(s) affected)'. There is a vertical cursor on the right side of the message pane.

نحوه پیاده سازی یک stored procedure بازگشتی

```

create proc spFactorial
@ValueIn int,
@ValueOut int OUTPUT
AS
declare @InWorking int;
declare @OutWorking int;
if @ValueIn != 1
begin
select @InWorking = @ValueIn -1;

exec spFactorial @InWorking, @OutWorking OUTPUT;

select @ValueOut = @ValueIn * @OutWorking ;
end
else
begin
select @ValueOut = 1;
end
return;
go

```

بنابراین آنچه که انجام داده ایم ، پذیرش یک مقدار و فراهم کردن مقدار مورد نیاز
حال می‌خواهیم stored procedure برگشتی را با script کوچکی امتحان کنیم.

```

SQLQuery34.sql...Afshin (56))* spFactorial.sql...PC\Afshin (E4))
declare @workingout int;
declare @workingin int;
select @workingin =5;
exec spFactorial @workingin,@workingout OUTPUT;
print cast(@workingin as varchar)+'factorial is : ' +
      cast(@workingout as varchar);

```

|

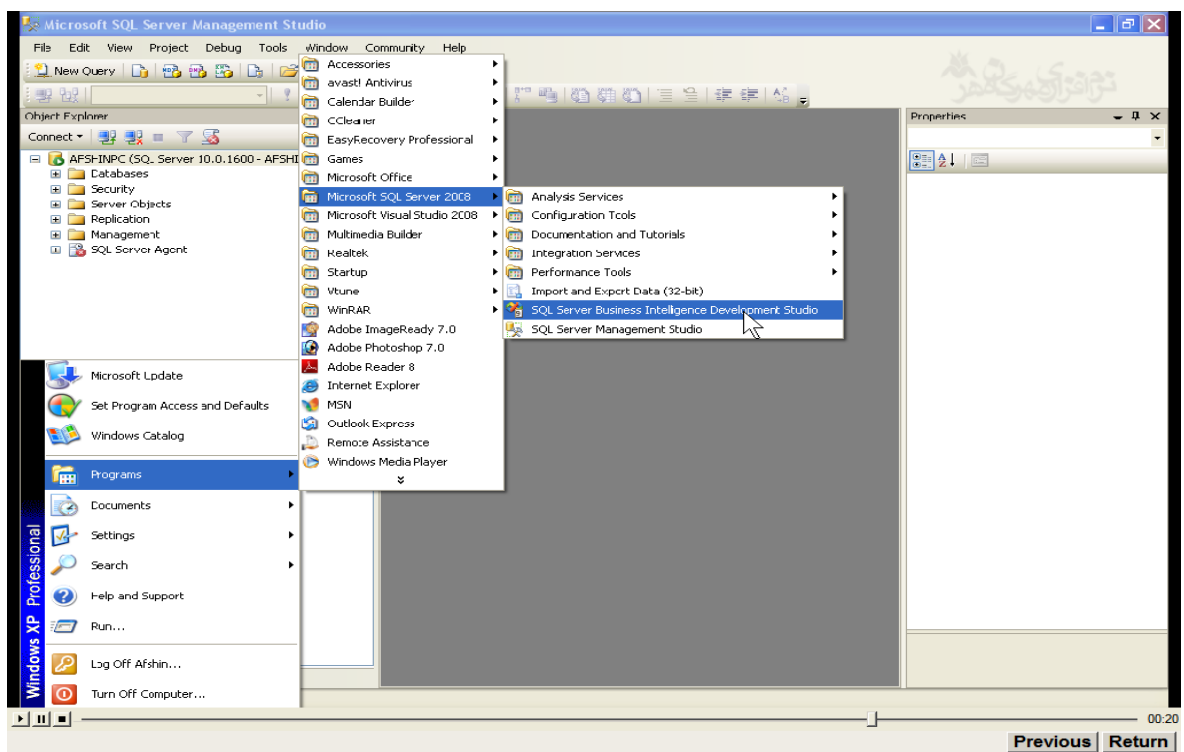
Messages

5:factorial is : 120

که مقدار 120 را به ما می دهد.

گزارشگیری

برای ایجاد گزارش به شرح زیر انجام می دهیم: (برنامه SQL server Business Intelligence Development Studio)



S