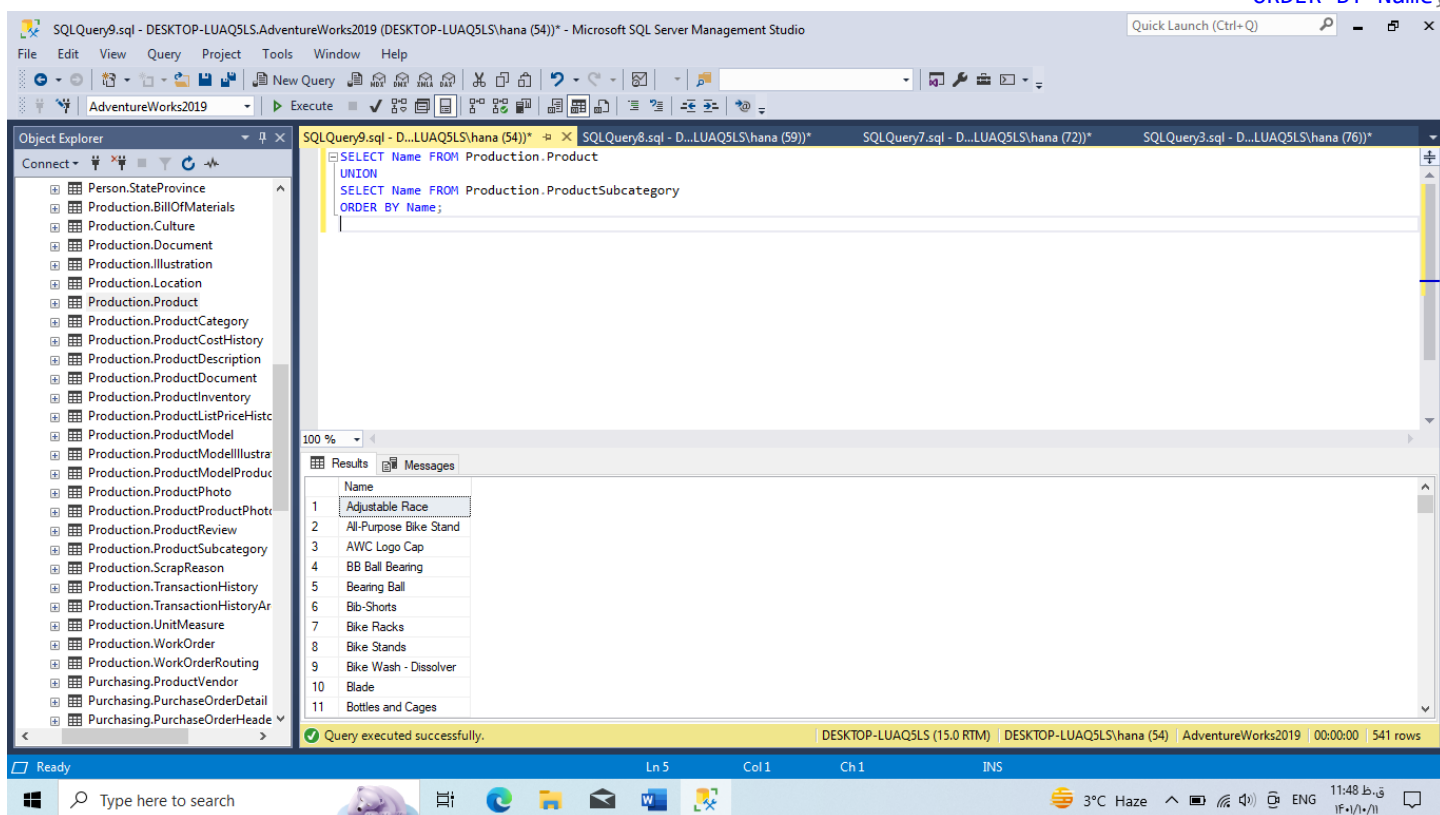


UNION : عملگر UNION برای ترکیب مجموعه نتیجه دو یا چند عبارت SELECT استفاده می شود.

هر عبارت SELECT در UNION باید دارای همان تعداد ستون باشد. ستون ها نیز باید دارای انواع داده های مشابه باشند. ستون های هر دستور SELECT نیز باید به همان ترتیب باشند

```
SELECT Name FROM Production.Product
UNION
SELECT Name FROM Production.ProductSubcategory;
```

ORDER BY Name;



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
SELECT Name FROM Production.Product
UNION
SELECT Name FROM Production.ProductSubcategory
ORDER BY Name;
```

The Results pane displays the output of the query, showing a list of product names sorted alphabetically:

Name
Adjustable Race
All-Purpose Bike Stand
AWC Logo Cap
BB Ball Bearing
Bearing Ball
Bib-Shorts
Bike Racks
Bike Stands
Bike Wash - Dissolver
Blade
Bottles and Cages

UNION ALL: اگر برخی از مشتریان یا تامین کنندگان شهر یکسانی داشته باشند، هر شهر فقط یک بار فهرست می شود، زیرا UNION فقط مقادیر متمایز را انتخاب می کند. از UNION ALL برای انتخاب مقادیر تکراری نیز استفاده کنید!

```
SELECT Name FROM Production.Product
UNION ALL
SELECT Name FROM Production.ProductSubcategory
ORDER BY Name;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The 'Object Explorer' on the left lists the database schema, including tables like 'Production.Product' and 'Production.ProductSubcategory'. The 'Query Editor' in the center contains the following SQL query:

```
SELECT Name FROM Production.Product
UNION ALL
SELECT Name FROM Production.ProductSubcategory
ORDER BY Name;
```

The 'Results' pane at the bottom displays the output of the query, which is a list of product names. The first row is 'Adjustable Race', and the last row is 'Bottles and Cages'. The status bar at the bottom indicates that the query was executed successfully and returned 541 rows.

Name
Adjustable Race
All-Purpose Bike Stand
AWC Logo Cap
BB Ball Bearing
Bearing Ball
Bib-Shorts
Bike Racks
Bike Stands
Bike Wash - Dissolver
Blade
Bottles and Cages

UNION With WHERE این دستور زیر (فقط مقادیر متمایز) را از هر دو جدول برمی‌گرداند:

```
SELECT name FROM Production.Product
WHERE name='chains'
UNION
SELECT name FROM Production.ProductSubcategory
WHERE name='chains'
```

ORDER BY name;

The screenshot shows the Microsoft SQL Server Management Studio interface. The 'Object Explorer' on the left lists various database objects. The 'Query Editor' in the center contains the following SQL query:

```
SELECT name FROM Production.Product
WHERE name='chains'
UNION
SELECT name FROM Production.ProductSubcategory
WHERE name='chains'
ORDER BY name;
```

The 'Results' pane at the bottom shows the output of the query, which is a single row with the value 'Chains' in the 'name' column.

name
Chains

The status bar at the bottom indicates that the query was executed successfully and returned 1 row.

GROUP BY: عبارت GROUP BY ردیف‌هایی را که مقادیر یکسانی دارند در ردیف‌های خلاصه گروه‌بندی می‌کند، مانند «تعداد مشتریان در هر کشور را بیابید» دستور GROUP BY اغلب با توابع انبوه (COUNT(), MAX(), MIN(), SUM(), AVG()) برای گروه‌بندی مجموعه نتایج توسط یک یا چند ستون استفاده می‌شود

```
SELECT COUNT(750), ComponentID
from Production.BillOfMaterials
GROUP BY ComponentID;
```

```
SELECT COUNT(750), ComponentID
FROM Production.BillOfMaterials
GROUP BY ComponentID
ORDER BY COUNT(ComponentID) DESC;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists the AdventureWorks2019 database schema. The query window contains two SQL queries. The first query is a GROUP BY statement, and the second is a GROUP BY statement with an ORDER BY clause. The Results window shows the output of the first query, displaying two tables of data.

Query 1:

```
SELECT COUNT(750), ComponentID
FROM Production.BillOfMaterials
GROUP BY ComponentID;
```

Query 2:

```
SELECT COUNT(750), ComponentID
FROM Production.BillOfMaterials
GROUP BY ComponentID
ORDER BY COUNT(ComponentID) DESC;
```

Results:

(No column name)	ComponentID
278	1
279	1
280	1
281	11
282	1
310	1
311	1
312	1
313	1
314	1

GROUP BY With JOIN

```
SELECT Shippers.ShipperName,
COUNT(Orders.OrderID) AS NumberOfOrders
FROM Orders
LEFT JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID
GROUP BY ShipperName;
```

```
select*
from Production.ProductSubcategory
```

```
select*
from Production.ProductModel
```

CREATE DATABASE دستور برای ایجاد یک پایگاه داده SQL جدید استفاده می شود

```
CREATE DATABASE testDB;
```

CREATE TABLE دستور برای ایجاد یک جدول جدید در پایگاه داده استفاده می شود.

پارامترهای ستون نام ستون های جدول را مشخص می کند.

پارامتر datatype نوع داده ای را که ستون می تواند نگه دارد) به عنوان مثال varchar ، عدد صحیح، تاریخ و غیره (مشخص می کند.

```
CREATE TABLE Persons (  
    PersonID int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)  
);
```

DROP TABLE : برای حذف کردن یک جدول موجود در پایگاه داده استفاده می شود.

```
DROP TABLE Person.PhoneNumberType;
```

DROP DATABASE : برای حذف کردن یک پایگاه داده یکجا از این دستور استفاده می شود.

```
DROP DATABASE testDB;
```

BACKUP DATABASE در SQL Server برای ایجاد یک نسخه پشتیبان کامل از پایگاه داده SQL موجود استفاده می شود.

```
BACKUP DATABASE databasename  
TO DISK = 'filepath'  
WITH DIFFERENTIAL;
```

ALTER TABLE : برای افزودن، حذف یا تغییر ستون ها در جدول موجود استفاده می شود. دستور ALTER TABLE همچنین برای افزودن و حذف محدودیت های مختلف در جدول موجود استفاده می شود.

```
ALTER TABLE Customers  
ADD Email varchar(255);
```

ALTER TABLE - ADD Column : برای اضافه کردن یک ستون در جدول، از دستور زیر استفاده کنید:

```
ALTER TABLE Customers  
ADD Email varchar(255);
```

ALTER TABLE - DROP COLUMN : برای حذف یک ستون در جدول، از نحو زیر استفاده کنید (توجه داشته باشید که برخی از سیستم های پایگاه داده اجازه حذف یک ستون را نمی دهند):

```
ALTER TABLE Customers
DROP COLUMN Email;
```

PRIMARY KEY : به طور منحصر به فرد هر رکورد را در یک جدول شناسایی می کند. کلیدهای اصلی باید حاوی مقادیر UNIQUE باشند و نمی توانند حاوی مقادیر NULL باشند. یک جدول فقط می تواند یک کلید اصلی داشته باشد. و در جدول، این کلید اولیه می تواند از یک یا چند ستون (فیلد) تشکیل شده باشد.

```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    CONSTRAINT PK_Person PRIMARY KEY (ID,LastName) ;
```

PRIMARY KEY برای ایجاد محدودیت **PRIMARY KEY** در ستون "ID" زمانی که جدول از قبل ایجاد شده است:

برای اجازه دادن به نامگذاری یک محدودیت کلید اصلی و برای تعریف یک محدودیت کلید اصلی در چندین ستون، از دستور SQL زیر استفاده کنید:

توجه: اگر از **ALTER TABLE** برای افزودن کلید اصلی استفاده می کنید، ستون(های) کلید اصلی باید فاقد مقادیر NULL باشند (زمانی که جدول برای اولین بار ایجاد شد).

```
ALTER TABLE Persons
ADD PRIMARY KEY (ID);
```

Comments نظرات برای توضیح بخش هایی از دستورات SQL یا جلوگیری از اجرای دستورات SQL استفاده می شود.

توجه: مثال های این فصل در فایرفاکس و مایکروسافت اج کار نمی کنند!

Comments تک خطی با -- شروع می شود. هر متنی بین -- و انتهای خط نادیده گرفته می شود (اجرا نمی شود).

```
--SELECT * FROM Customers;
SELECT * FROM Products;
```