



عنوان مقاله: دستور SELECT در SQL Server

نویسنده مقاله: تیم فنی نیک‌آموز

تاریخ انتشار: ۱۷ دی ۱۴۰۲

منبع: <https://nikamooz.com/select-statement-in-sql-server/>

دستور Select در SQL Server کاربردهای متنوعی دارد و شما می‌توانید آن را در انواع کوئری‌های زبان T-SQL به کار ببرید. این دستور به‌عنوان یکی از اجزای کلیدی در اسکریپت‌نویسی و به هدف استخراج داده‌ها مورد استفاده قرار می‌گیرد. در این مقاله، دستور SELECT و کاربردهای آن به‌همراه مثال‌های قابل درک شرح داده می‌شوند؛ به طوری که پس از مطالعه آن، با ضرورت و اهمیت Select آشنا خواهید شد.

زبان SQL چیست و چه تفاوتی با T-SQL دارد؟

زبان اسکریپت‌نویسی SQL یک زبان کوئری‌نویسی ساختاریافته است که به منظور مدیریت و دستکاری در **پایگاه داده رابطه‌ای** (RDBMS) طراحی شده است. این زبان، با ارائه یک راه استانداردشده در اموری همچون تعامل با پایگاه داده، اجرای کوئری‌ها، به‌روزرسانی رکوردها، درج داده‌های جدید و مدیریت ساختار خود دیتابیس مورد استفاده قرار می‌گیرد. در حال حاضر، SQL به‌صورت گسترده در انواع حوزه‌های صنعت و آکادمیک به کار می‌رود و اغلب سیستم‌های مدیریت پایگاه داده (DBMS) از آن پشتیبانی می‌کنند.

زبان T-SQL نسخه تعمیم‌یافته SQL محسوب می‌شود که توسط شرکت مایکروسافت پیاده‌سازی شده است. T-SQL به طور خاص با Microsoft SQL Server کار می‌کند و در آن، قابلیت‌های توسعه‌یافته‌ای مانند **استور پروسیجر** (Stored Procedures)، توابع تعریف‌شده توسط کاربر (User-Defined)، Trigger ها و... افزوده شده است.

دستور SELECT در SQL Server چیست؟

SQL Server به‌عنوان یک سیستم مدیریت پایگاه داده قدرتمند، امکانات و قابلیت‌های گوناگونی در اختیار توسعه‌دهنده و **مدیر پایگاه داده** (DBA) قرار می‌دهد. دستور Select در SQL Server یکی از اصلی‌ترین **دستورات SQL Server** به شمار می‌رود که به واسطه آن، امکان استخراج داده‌ها وجود دارد؛ به طوری که گاهی به این دستور، تحت عنوان سنگ بنای کوئری‌نویسی و استخراج داده‌ها از پایگاه‌های داده SQL Server اشاره می‌شود. در بخش بعدی، به موارد استفاده دستور Select پرداخته خواهد شد تا نقش آن در کوئری‌نویسی شفاف‌تر شود.



کاربرد Select در SQL Server

در ادامه، متداولترین موارد استفاده Select در SQL Server شرح داده می‌شوند.

۱- استخراج داده

اصلی‌ترین کاربرد دستور Select در SQL Server، استخراج داده‌ها از یک یا چند جدول پایگاه داده است. این عمل، واکنشی یک یا چند ستون خاص از جدول را شامل می‌شود. به‌عنوان مثال، فرض کنید قصد دارید نام و نام خانوادگی کارکنان سازمان را از جدول مربوط به آن‌ها (Employee) استخراج کنید. در چنین شرایطی، باید دستور Select را به صورت زیر اجرا کنید:

```
SELECT FirstName, LastName FROM Employees;
```

	FirstName	LastName
1	Nancy	Davolio
2	Andrew	Fuller
3	Janet	Leverling
4	Margaret	Peacock
5	Steven	Buchanan
6	Michael	Suyama
7	Robert	King
8	Laura	Callahan
9	Anne	Dodsworth

۲- فیلترینگ داده ها با Where

شما می‌توانید با استفاده از WHERE در SQL Server ، سطرها را براساس شرایط خاصی فیلتر کنید. به این ترتیب، این امکان فراهم می‌شود که زیرمجموعه‌ای از داده‌های جدول استخراج شوند. برای درک بهتر، به مثال زیر توجه کنید. فرض کنید می‌خواهیم تنها قیمت و نام محصولاتی را از جدول محصول استخراج کنیم که CategoryID آن‌ها برابر با ۲ باشد. در چنین شرایطی، لازم است دستور Select به همراه شرط زیر به کار رود:

```
SELECT ProductName, UnitPrice, CategoryID FROM Products WHERE CategoryID = 2;
```

	ProductName	UnitPrice	CategoryID
1	Aniseed Syrup	10.00	2
2	Chef Anton's Cajun Seasoning	22.00	2
3	Chef Anton's Gumbo Mix	21.35	2
4	Grandma's Boysenberry Spread	25.00	2
5	Northwoods Cranberry Sauce	40.00	2
6	Genen Shouyu	15.50	2
7	Gula Malacca	19.45	2
8	Sirop d'érable	28.50	2
9	Vegie-spread	43.90	2
10	Louisiana Fiery Hot Pepper Sauce	21.05	2
11	Louisiana Hot Spiced Okra	17.00	2
12	Original Frankfurter grüne Soße	13.00	2

۳- مرتب سازی نتایج با Order By

در برخی مواقع، شرایط حکم می‌کند که نتایج حاصل از اجرای یک کوئری را مرتب‌سازی (Sort) کنید. برای این کار، کافیست دستور Select به همراه Order By استفاده شود. مرتب‌سازی نتایج در مواقعی مناسب است که بخواهید داده‌ها در یک ترتیب خاص، صعودی (Ascending) یا نزولی (Descending)، نمایش داده شوند. برای مثال، در ادامه قصد داریم نام محصولات و قیمت آن‌ها را به صورت نزولی نمایش دهیم:

```
SELECT ProductName, UnitPrice, CategoryID FROM Products ORDER BY UnitPrice DESC;
```

	ProductName	UnitPrice	CategoryID
1	Côte de Blaye	263.50	1
2	Thüringer Rostbratwurst	123.79	6
3	Mishi Kobe Niku	97.00	6
4	Sir Rodney's Marmalade	81.00	3
5	Camaron Tigers	62.50	8
6	Raclette Courdavault	55.00	4
7	Manjimup Dried Apples	53.00	7
8	Tarte au sucre	49.30	3
9	Ipoh Coffee	46.00	1
10	Rössle Sauerkraut	45.60	7
11	Schoggi Schokolade	43.90	3
12	Vegie-spread	43.90	2
13	Northwoods Cranberry Sauce	40.00	2
14	Alice Mutton	39.00	6

۴- حذف سطرهای تکراری با Distinct

به واسطه کلمه کلیدی Distinct و دستور Select در SQL Server می‌توان سطرهای تکراری را از نتیجه حذف کرد و یک لیست یکتا از مقادیر به دست آورد. با اجرای دستور زیر، CategoryID های منحصر به فرد از جدول محصولات (Products) استخراج می‌شوند.

```
SELECT DISTINCT CategoryID FROM Products;
```

	CategoryID
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8

برای درک بهتر این Use-Case دستور Select، پیشنهاد می‌شود [مقاله دستور Select Distinct در SQL](#) نیز مطالعه شود.

۵- استفاده از دستور Select به همراه توابع تجمیعی

با کمک Group By و دستور Select ، می‌توان سطرها را براساس یک یا چند ستون خاص، گروه‌بندی کرد. معمولاً در این مورد استفاده دستور Select، توابع تجمیعی (Aggregation Functions) مانند COUNT ، SUM ، AVG و... به همراه Group By به کار می‌روند. برای درک بهتر، به دستور زیر توجه کنید.

```
SELECT CategoryID, AVG(UnitPrice) AS AvgUnitPrice FROM Products GROUP BY CategoryID;
```

	CategoryID	AvgUnitPrice
1	1	37.9791
2	2	23.0625
3	3	25.16
4	4	28.73
5	5	20.25
6	6	54.0066
7	7	32.37
8	8	20.6825

با اجرای مثال فوق، نتیجه این کوئری در قالب دو ستون CategoryID و AvgUnitPrice نمایش داده می‌شود؛ به طوری که هر سطر، نمایان‌گر یک Category خاص باشد و در ستون AvgUnitPrice مرتبط به آن، میانگین قیمت محصولات آن Category نمایش داده شود.

۶- پیاده سازی Join روی جداول

دستور Select به شما این اجازه را می‌دهد که چندین جدول پایگاه داده را Join کنید. در واقع، زمانی که شما بخواهید داده‌هایی از دو یا چند جدول پایگاه داده رابطه‌ای را با یکدیگر ترکیب کنید، لازم است دستور Select را به همراه کلاز Join استفاده کنید. برای درک بهتر عملیات Join و کاربرد دستور Select ، به مثال زیر توجه کنید:

```
SELECT
  Orders.OrderID,
  Customers.CompanyName,
  Orders.OrderDate
FROM Orders
JOIN Customers ON
  Orders.CustomerID = Customers.CustomerID;
```

	OrderID	CompanyName	OrderDate
1	10248	Vins et alcools Chevalier	1996-07-04 00:00:00.000
2	10249	Toms Spezialitäten	1996-07-05 00:00:00.000
3	10250	Hanari Carnes	1996-07-08 00:00:00.000
4	10251	Victuailles en stock	1996-07-08 00:00:00.000
5	10252	Suprêmes délices	1996-07-09 00:00:00.000
6	10253	Hanari Carnes	1996-07-10 00:00:00.000
7	10254	Chop-suey Chinese	1996-07-11 00:00:00.000
8	10255	Richter Supermarkt	1996-07-12 00:00:00.000
9	10256	Wellington Importadora	1996-07-15 00:00:00.000
10	10257	HILARION-Abastos	1996-07-16 00:00:00.000

۷- زیر کوئری (SubQuery) برای شرط های پیچیده

با به کارگیری Subquery ها درون دستور Select ، امکان رسیدگی به شرط های پیچیده و استخراج داده ها بر مبنای نتایج حاصل از یک کوئری دیگر وجود دارد. فرض کنید بخواهیم نام شرکت مشتریانی را استخراج کنیم که در تاریخ '۱۹۹۶-۱۰-۳۰' ثبت سفارش انجام داده اند. برای این کار، لازم است دستور Select به صورت زیر مورد استفاده قرار بگیرد.

SELECT

CompanyName

FROM Customers

WHERE CustomerID IN (SELECT CustomerID FROM Orders WHERE OrderDate = '1996-10-30');

	CompanyName
1	Frankenversand

۸- عملگرهای SET

شما می توانید با استفاده از عملگرهای SET، شامل UNION، INTERSECT و EXCEPT، در یک دستور Select، نتایج حاصل از اجرای چند کوئری را ترکیب یا مقایسه کنید. برای مثال، نحوه استفاده از عملگر UNION به همراه دستور Select به شکل زیر است:

SELECT ProductID AS RecID, ProductName AS RecTitle, 'Products' AS RecType
FROM Products

UNION

SELECT EmployeeID AS RecID, LastName AS RecTitle, 'Employees' AS RecType
FROM Employees

	RecID	RecTitle	RecType
73	7	Uncle Bob's Organic Dried P...	Products
74	50	Valkoinen suklaa	Products
75	63	Vegie-spread	Products
76	64	Wimmers gute Semmelknödel	Products
77	47	Zaanse koeken	Products
78	1	Davolio	Employees
79	2	Fuller	Employees
80	3	Leverling	Employees
81	4	Peacock	Employees
82	5	Rochanan	Employees

با اجرای کوئری فوق، نتایج حاصل از دو کوئری مختلف در قالب یک نتیجه، ترکیب و نمایش داده می‌شوند.

۹- استفاده از Window Function ها برای کوئری نویسی تحلیلی

به واسطه استفاده از Window Function ها در دستور Select ، می‌توان محاسبات را روی بازه خاصی از ستون‌های مرتبط با ستون فعلی انجام داد. در مثال زیر، با کمک AVG() و کلاز OVER، میانگین قیمت هر Category از محصول محاسبه می‌شود.

SELECT

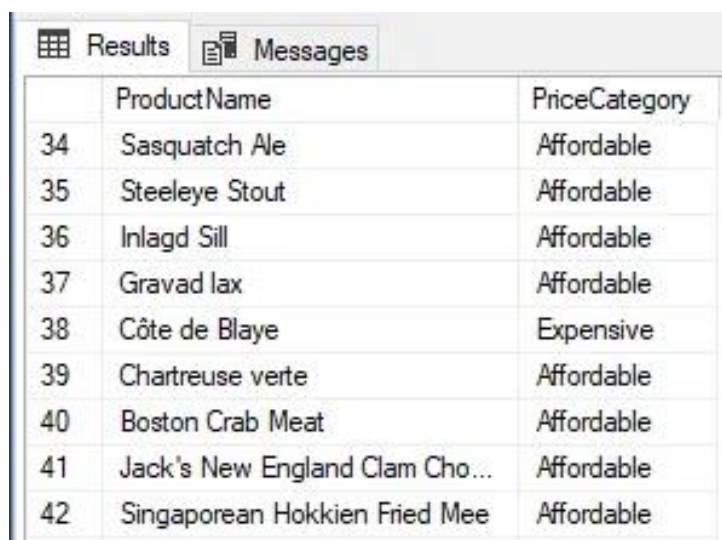
```
ProductID, UnitPrice,
AVG(UnitPrice) OVER (PARTITION BY CategoryID) AS AvgUnitPriceInCategory
FROM Products;
```

	ProductID	UnitPrice	AvgUnitPriceInCategory
1	1	18.00	37.9791
2	2	19.00	37.9791
3	24	4.50	37.9791
4	34	14.00	37.9791
5	35	18.00	37.9791
6	38	263.50	37.9791
7	39	18.00	37.9791
8	43	46.00	37.9791
9	70	15.00	37.9791

۱۰- منطق شرطی و استفاده از عبارت Case When

شما می‌توانید با بهره‌مندی از عبارت CASE WHEN در دستور Select ، منطق شرطی را معرفی کرده و خروجی را براساس شرایط خاصی شخص‌سازی کنید. در مثال زیر، از CASE به منظور طبقه‌بندی محصولات براساس قیمت آن‌ها استفاده خواهد شد:

```
SELECT
    ProductName,
    CASE
        WHEN UnitPrice > 100 THEN 'Expensive'
        ELSE 'Affordable'
    END AS PriceCategory
FROM Products;
```



	ProductName	PriceCategory
34	Sasquatch Ale	Affordable
35	Steeleye Stout	Affordable
36	Inlagd Sill	Affordable
37	Gravad lax	Affordable
38	Côte de Blaye	Expensive
39	Chartreuse verte	Affordable
40	Boston Crab Meat	Affordable
41	Jack's New England Clam Cho...	Affordable
42	Singaporean Hokkien Fried Mee	Affordable

علاوه بر این، [مقاله استفاده از عبارت Case در SQL Server](#) می‌تواند برای شما به‌عنوان مطلب مکمل مفید باشد.

۱۱- اعمال محدودیت روی سطرها با TOP

شما می‌توانید با کمک کلید واژه TOP در دستور Select ، هر تعداد خاصی از ستون‌ها را از ابتدای مجموعه جواب استخراج کنید. به‌عنوان مثال، با اجرای کوئری زیر، «۱۰ محصول بالایی» از جدول Products، به همراه نام و قیمت آن‌ها استخراج شده و براساس هزینه، به‌صورت نزولی (DESC) مرتب‌سازی می‌شوند و درنهایت، در خروجی نشان داده خواهند شد.

```
SELECT TOP 10
    ProductName, UnitPrice
FROM Products
ORDER BY UnitPrice DESC;
```


	ProductName	UnitPrice
1	Côte de Blaye	263.50
2	Thüringer Rostbratwurst	123.79
3	Mishi Kobe Niku	97.00
4	Sir Rodney's Marmalade	81.00
5	Camarvon Tigers	62.50
6	Raclette Courdavault	55.00
7	Manjimup Dried Apples	53.00
8	Tarte au sucre	49.30
9	Ipoh Coffee	46.00
10	Rössle Sauerkraut	45.60

۱۲- Pivot و Unpivot داده ها

Pivot و Unpivot در SQL Server دو عملگری به شمار می‌روند که برای تبدیل داده‌ها به فرمت‌های ستونی و سطری استفاده می‌شوند. در کوئری زیر، از Pivot در دستور Select استفاده شده است و به کمک آن، تعداد رخداد هر CategoryID برای هر محصول شمرده می‌شود.

SELECT

*

```
FROM (SELECT ProductName, CategoryID FROM Products) AS SourceTable
PIVOT
(
    COUNT(CategoryID)
    FOR CategoryID IN ([1], [2], [3], [4])
) AS PivotTable;
```

	ProductName	1	2	3	4
1	Alice Mutton	0	0	0	0
2	Aniseed Syrup	0	1	0	0
3	Boston Crab Meat	0	0	0	0
4	Camembert Pierrot	0	0	0	1
5	Camarvon Tigers	0	0	0	0
6	Chai	1	0	0	0
7	Chang	1	0	0	0
8	Chartreuse verte	1	0	0	0
9	Chef Anton's Cajun Seasoning	0	1	0	0

پیشنهاد می‌شود [مقاله PIVOT و UNPIVOT در SQL Server - قسمت اول](#) را نیز مطالعه کنید تا درک بهتری از آن به دست آورید.

۱۳- استفاده از دستور Select در Insert Into

شما می‌توانید دستور Select را به صورتی ترکیبی با دستور Insert Into به کار ببرید. با این کار، براساس کوئری خاصی، داده‌ها به جدول مورد نظر درج می‌شوند. معمولاً به این عبارت، تحت عنوان INSERT INTO SELECT نیز اشاره می‌شود. برای درک بهتر، به مثال‌های زیر توجه کنید.

```
INSERT INTO NewEmployees
SELECT * FROM Employees;
```

با اجرای کوئری فوق، تمام ستون‌های جدول Employees به جدول NewEmployees درج می‌شوند. این موضوع، در مواقعی مناسب است که بخواهید داده‌ها را از یک جدول به جدول دیگر کپی کنید.

در مثال زیر، تنها ستون‌های مشخصی به جدول SalesEmployees درج می‌شوند که شرط جلوی کلمه WHERE برای آن‌ها صدق کند.

```
INSERT INTO SalesEmployees (EmployeeID, FirstName, LastName)
SELECT
EmployeeID, FirstName, LastName
FROM Employees
WHERE Department = 'Sales';
```

۱۴- استفاده از دستور Select به همراه کلاز Having

در SQL Server می‌توان Having را به همراه دستور Select استفاده کرد تا بدین طریق، نتایج حاصل از کوئری را براساس مقادیر تجمیعی فیلتر کرد. توجه کنید که با به کارگیری Where، پیش از گروه‌بندی و تجمیع سطرها، فرآیند فیلترکردن سطرها انجام می‌شود؛ در حالی که با استفاده از Having، فرآیند فیلترینگ نتایج پس از گروه‌بندی و تجمیع آن‌ها اعمال خواهد شد. به منظور درک بهتر، به کوئری زیر توجه کنید:

```
SELECT
CustomerID, SUM(Freight) as TotalFreight
FROM orders
GROUP BY CustomerID
HAVING SUM(Freight)>300;
```

	CustomerID	TotalFreight
1	AROUT	471.95
2	BERGS	1559.52
3	BLONP	623.66
4	BONAP	1357.87
5	BOTTM	793.95
6	CHOPS	367.24
7	DRACD	306.04
8	EASTC	832.34
9	ERNSH	6205.39

در مثال فوق، لیست مقادیر CustomerID ها به همراه «مجموعه مقدار هزینه حمل» بازگردانده می‌شوند؛ البته در صورتی که مجموعه هزینه حمل آن‌ها بزرگ‌تر از ۳۰۰ باشد.

نکته حائز اهمیت این است که Having با توابع تجمیعی به کار می‌رود و پس از اعمال Group By اعمال خواهد شد.

۱۵- Index Optimizer Hint در SQL Server

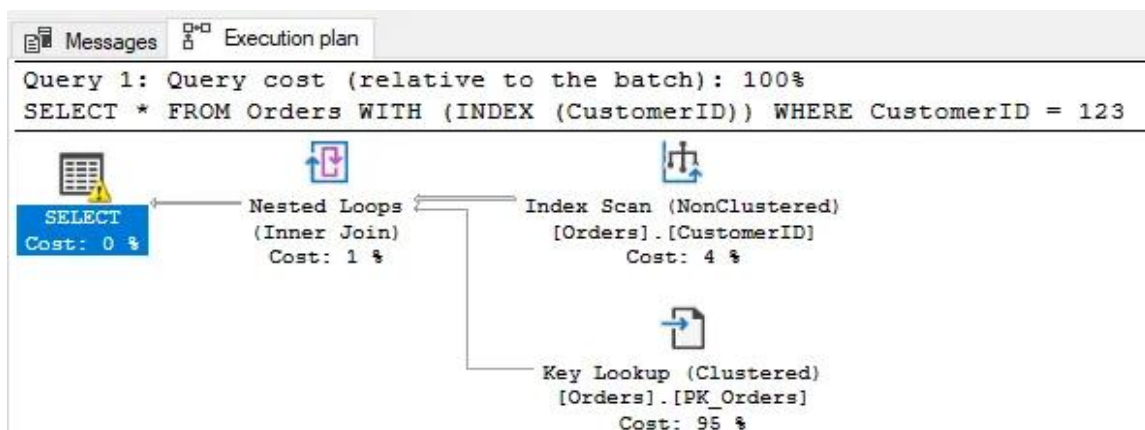
شما می‌توانید با استفاده از Optimizer Hint خاصی که INDEX نام دارد، یک [ایندکس](#) مشخص برای بهینه‌ساز کوئری پیشنهاد کنید. INDEX Hint به شما این امکان را می‌دهد که انتخاب پیش‌فرض بهینه‌ساز را برای یک کوئری مشخص، نادیده بگیرید. در مثال زیر، با به‌کارگیری WITH(INDEX(CustomerID)) به عنوان Hint، بهینه‌سازی متوجه خواهد شد که شما یک ایندکس خاص برای کوئری پیشنهاد کرده‌اید.

SELECT

*

FROM Orders WITH (INDEX (CustomerID))

WHERE CustomerID = 123;



در این بخش، مهم‌ترین کاربردهای دستور Select در SQL Server مورد بررسی قرار گرفتند. با توجه به نقش کلیدی Select، بدیهی است که امکان استفاده از آن در [انواع اسکریپت های SQL Server](#) وجود دارد.

کلام آخر: دستور Select در SQL Server

دستور Select در SQL Server به‌عنوان یکی از عبارات پایه و متداول کوئری‌نویسی شناخته شده است. به واسطه این دستور، می‌توانید اموری همچون استخراج داده‌ها، فیلترینگ داده‌ها، پیاده‌سازی جوین روی جداول، حذف مقادیر تکراری، استفاده از Subquery ها و سایر موارد را انجام داد. در این مطلب، پراهمیت‌ترین Use Case های دستور Select در SQL Server به‌همراه مثال‌های کاربردی شرح داده شدند و در مقالات آتی، نحوه استفاده از سایر کوئری‌های T-SQL آموزش داده خواهند شد.