

Veritabanı Programlama

Bölüm 5

Şema ve İç İçe Sorgular

Dr. Öğr. Üyesi Murat TAŞYÜREK (kayubmprogramlama1@gmail.com)

1 Kasım 2023

Kayseri Üniversitesi, Bilgisayar Mühendisliği Bölümü

Schema (Şema)


- **Schema (Şema)** veritabanındaki tablo, view veya stored procedure gibi objeleri gruplamaya yarayan bir ifadedir.
- Default **schema** dbo'dur.
- Veri tabanı nesnelerinin fazla olduğu ve yetki bazlı erişime ihtiyaç duyulduğunda **şema** yaygın olarak kullanılır.
- SQL server'da şema bazlı yetkilendirme mevcuttur.
- İzin (yetki) verdiğiniz kullanıcı sadece izin verilen **şemanın** altındaki nesneleri görür diğerlerini göremez.

Schema (Şema)

- Veritabanında bulunan nesne sayısı arttıkça yetki vermek ve yönetmek zorlaşacaktır.
- Bu gibi sorunların önüne geçmek için **şema** oluşturulur ve yetkilendirir.
- Bir objeyi bir **şema** altında oluşturabilir veya başka **şemanın** altına transfer edebilirsiniz.

```
-- create schemas  
CREATE SCHEMA SEMA_ADI ;
```

- Open-data olarak yayımlanmış şema ve altındaki nesneleri indirip kurup örnekleri onun üzerinden yapacağız.
- <https://www.sqlservertutorial.net/sql-server-sample-database/> adresinde bulunan "Download SQL Server Sample Database" veritabanını bilgisayarımıza indirmek için **tıklayınız** .

 <https://www.sqlservertutorial.net/sql-server-sample-database/>

sel Sayfam  <https://yoksis.yok.gov...>  Murat Taşyürek (0000-...  Zimbra: Gelen Kutusu  Makaleler | Kayseri Üni...  TÜBİTAK TEYDEB F

```
CREATE TABLE production.stocks (  
    store_id INT,  
    product_id INT,  
    quantity INT,  
    PRIMARY KEY (store_id, product_id),  
    FOREIGN KEY (store_id)  
    REFERENCES sales.stores (store_id)  
    ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (product_id)  
    REFERENCES production.products (product_id)  
    ON DELETE CASCADE ON UPDATE CASCADE  
);
```

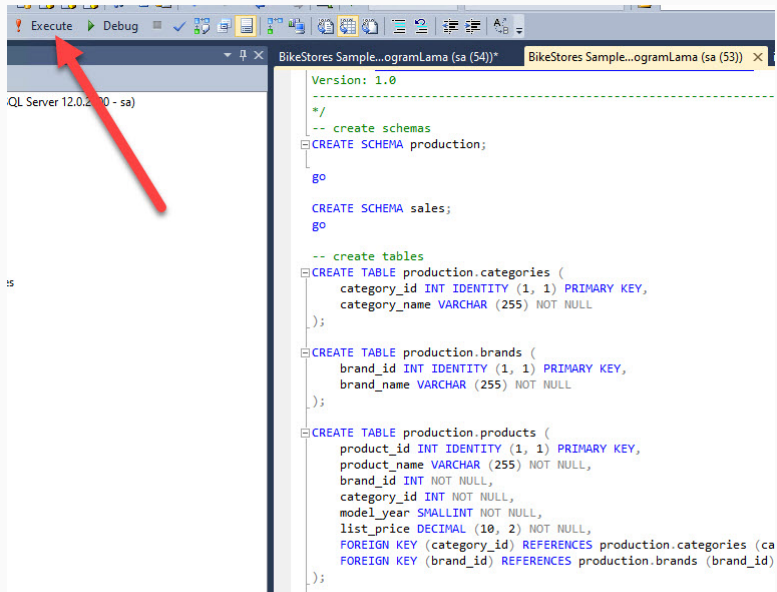
Click the following link to download the sample database script:

[Download SQL Server Sample Database](#)

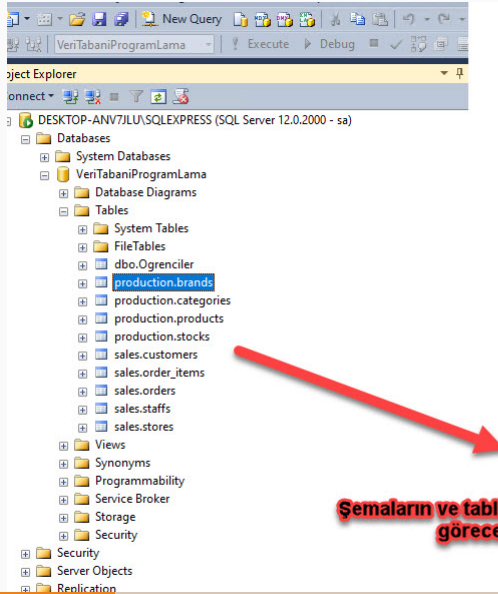


- İndirdiğimiz dizinde 3 adet dosya bulunmaktadır.
- Sql server management studio (SSMS) açalım.
- BikeStores Sample Database - create objects.sql isimli dosyayı SSMS ile açalım ve çalıştırıp şema ve tabloları oluşturmasını sağlayalım.

Örnek Şema ve Tablolar



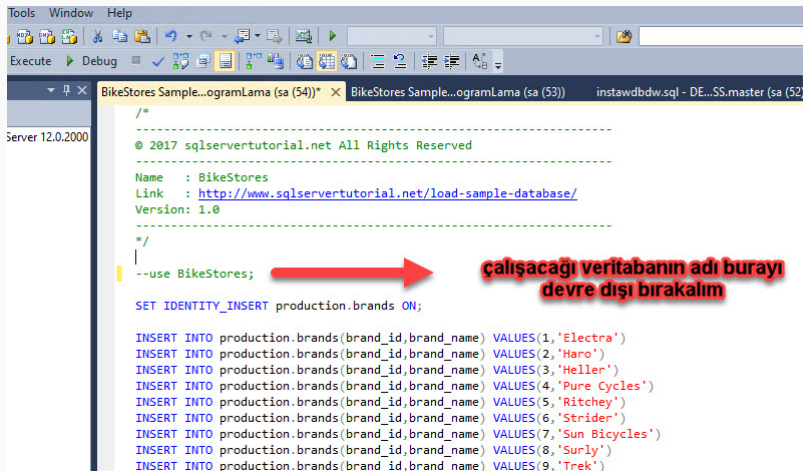
Şemalar ve Tablolar



**Şemaların ve tabloların oluştuğunu
göreceksiniz.**

Tablolara Kayıt Ekleme

- BikeStores Sample Database - load data.sql isimli dosyayı SSMS ile açalım ve çalıştırıp tablolara kayıt eklemesini sağlayalım.



```
Tools Window Help
Execute Debug
BikeStores Sample...ogramLama (sa (54))* x BikeStores Sample...ogramLama (sa (53)) instawdbdw.sql - DE...SS.master (sa (52))
Server 12.0.2000

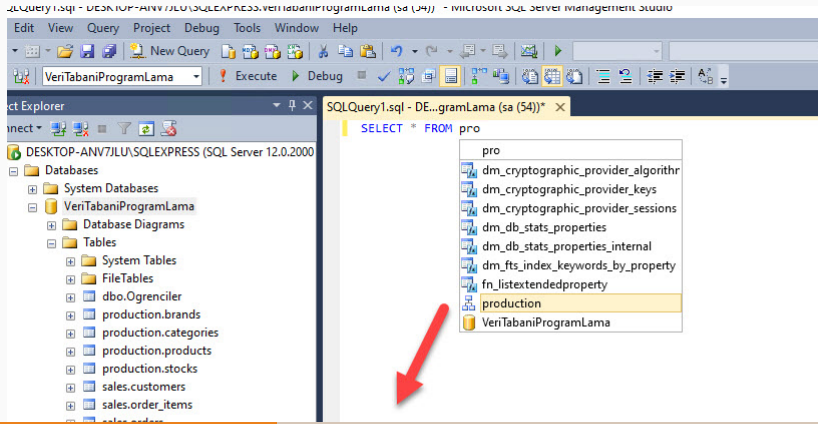
/*
-----
© 2017 sqlservertutorial.net All Rights Reserved
-----
Name : BikeStores
Link : http://www.sqlservertutorial.net/load-sample-database/
Version: 1.0
-----
*/
|
--use BikeStores;
SET IDENTITY_INSERT production.brands ON;

INSERT INTO production.brands(brand_id,brand_name) VALUES(1,'Electra')
INSERT INTO production.brands(brand_id,brand_name) VALUES(2,'Haro')
INSERT INTO production.brands(brand_id,brand_name) VALUES(3,'Heller')
INSERT INTO production.brands(brand_id,brand_name) VALUES(4,'Pure Cycles')
INSERT INTO production.brands(brand_id,brand_name) VALUES(5,'Ritchey')
INSERT INTO production.brands(brand_id,brand_name) VALUES(6,'Strider')
INSERT INTO production.brands(brand_id,brand_name) VALUES(7,'Sun Bicycles')
INSERT INTO production.brands(brand_id,brand_name) VALUES(8,'Surly')
INSERT INTO production.brands(brand_id,brand_name) VALUES(9,'Trek')
```

çalışacağı veritabanının adı burayı devre dışı bırakalım

Şemalar üzerinde sorgu

- Şema üzerinde sorgu yaparken **SELECT ... FROM şema adı.tablo adı** şeklinde yapılır.
- **SELECT * FROM production.products**



Sorgu sonucu

2LQuery1.sql - DESKTOP-ANV7JLU\SQLEXPRESS.VeriTabaniProgramLama (sa (54))* - Microsoft SQL Server Management Studio

Edit View Query Project Debug Tools Window Help

VeriTabaniProgramLama Execute Debug

ct Explorer

DESKTOP-ANV7JLU\SQLEXPRESS (SQL Server 12.0.2000)

Databases

- System Databases
- VeriTabaniProgramLama
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - dbo.Ogrenciler
 - production.brands
 - production.categories
 - production.products
 - production.stocks
 - sales.customers
 - sales.order_items
 - sales.orders
 - sales.staffs
 - sales.stores
 - Views
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
 - Security

SQLQuery1.sql - DE...gramLama (sa (54))*

```
SELECT * FROM production.products
```

100 %

Results Messages

	product_id	product_name	brand_id	category_id	model_year	list_price
1	1	Trek 820 - 2016	9	6	2016	379.99
2	2	Ritchey Timberwolf Frameset - 2016	5	6	2016	749.99
3	3	Surly Wednesday Frameset - 2016	8	6	2016	999.99
4	4	Trek Fuel EX 8 29 - 2016	9	6	2016	2899.99
5	5	Heller Shagawaw Frame - 2016	3	6	2016	1320.99
6	6	Surly Ice Cream Truck Frameset - 2016	8	6	2016	469.99
7	7	Trek Slash 8 27.5 - 2016	9	6	2016	3999.99
8	8	Trek Remedy 29 Carbon Frameset - 2016	9	6	2016	1799.99
9	9	Trek Conduit+ - 2016	9	5	2016	2999.99
10	10	Surly Straggler - 2016	8	4	2016	1549.00
11	11	Surly Straggler 650b - 2016	8	4	2016	1680.99
12	12	Electra Townie Original 21D - 2016	1	3	2016	549.99
13	13	Electra Cruiser 1 (24-Inch) - 2016	1	3	2016	269.99
14	14	Electra Girl's Hawaii 1 (16-inch) - 2015/2016	1	3	2016	269.99
15	15	Electra Moto 1 - 2016	1	3	2016	529.99
16	16	Electra Townie Original 20 EQ - 2016	1	3	2016	509.99

Örnek Sorgular

- **in** komutu birden fazla öge için şartı sağlayan kayıtları bulmak için kullanılan bir komuttur.
- Aynı kuralın olumsuzunu kullanmak için **NOT IN** kullanılır.
- Genel kullanımı aşağıdaki şekilde gibidir.
- Kategori adı c ile başlayan ürünleri listeleyen SQL cümleciğini kodlayalım.

Syntax

The syntax for the IN condition in SQL is:

```
expression IN (value1, value2, .... value_n);
```

OR

```
expression IN (subquery);
```

IN Sorgu sonucu

2000

```
SELECT * FROM production.products where  
category_id in (Select category_id from production.categories  
c where c.category_name LIKE 'C%')
```

214 %

Results Messages

	product_id	product_name	brand_id	category_id	model_year	list_price
1	10	Surly Straggler - 2016	8	4	2016	1549.00
2	11	Surly Straggler 650b - 2016	8	4	2016	1680.99
3	12	Electra Townie Original 21D - 2016	1	3	2016	549.99
4	13	Electra Cruiser 1 (24-Inch) - 2016	1	3	2016	269.99
5	14	Electra Girl's Hawaii 1 (16-inch) - 2015/2016	1	3	2016	269.99
6	15	Electra Moto 1 - 2016	1	3	2016	529.99
7	16	Electra Townie Original 7D EQ - 2016	1	3	2016	599.99
8	17	Pure Cycles Vine 8-Speed - 2016	4	3	2016	429.00
9	18	Pure Cycles Western 3-Speed - Women's - 2015/2016	4	3	2016	449.00
10	19	Pure Cycles William 3-Speed - 2016	4	3	2016	449.00
11	20	Electra Townie Original 7D EQ - Women's - 2016	1	3	2016	599.99
12	21	Electra Cruiser 1 (24-Inch) - 2016	1	1	2016	269.99
13	22	Electra Girl's Hawaii 1 (16-inch) - 2015/2016	1	1	2016	269.99
14	23	Electra Girl's Hawaii 1 (20-inch) - 2015/2016	1	1	2016	299.99

SQL JOIN

- SQL join ifadesi, aralarındaki ortak alana dayalı olarak iki veya daha fazla tablodaki verileri veya satırları birleştirmek için kullanılır. Aşağıda sunulduğu şekilde türleri mevcuttur.
- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL JOIN
- NATURAL JOIN

INNER JOIN

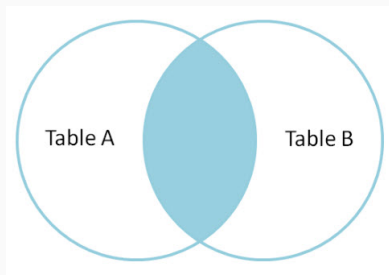
- **INNER JOIN** anahtar sözcüğü, koşul sağlandığı sürece her iki tablodaki tüm satırları seçer. Bu anahtar kelime, koşulun karşılandığı her iki tablodaki tüm satırları birleştirerek sonuç kümesini oluşturacaktır; yani ortak alanın değeri aynı olacaktır.

```
SELECT table1.column1,table1.column2,table2.column1,...  
FROM table1  
INNER JOIN table2  
ON table1.matching_column = table2.matching_column;
```

table1: First table.

table2: Second table

matching_column: Column common to both the tables.



LEFT JOIN

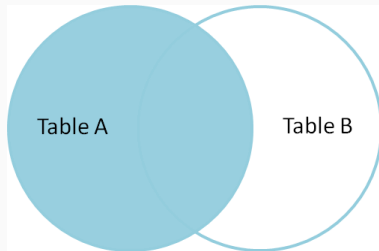
- Bu birleştirme, birleştirmenin sol tarafındaki tablonun tüm satırlarını döndürür ve birleştirmenin sağ tarafındaki tablonun satırlarını eşleştirir. Sağ tarafta eşleşen satır bulunmayan satırlar için sonuç kümesi null değerini içerecektir. **LEFT JOIN** aynı zamanda **LEFT OUTER JOIN** olarak da bilinir.

```
SELECT table1.column1,table1.column2,table2.column1,...  
FROM table1  
LEFT JOIN table2  
ON table1.matching_column = table2.matching_column;
```

table1: First table.

table2: Second table

matching_column: Column common to both the tables.



RIGHT JOIN

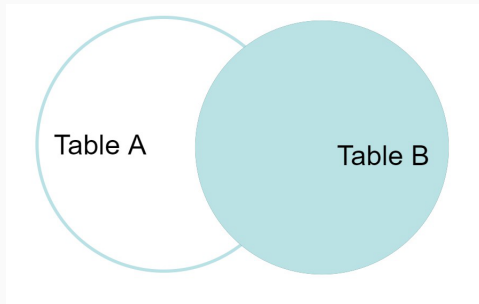
- **RIGHT JOIN**, **LEFT JOIN**'e benzer. Bu birleştirme, birleştirmenin sağ tarafındaki tablonun tüm satırlarını ve birleştirmenin sol tarafındaki tablo ile eşleşen satırları döndürür. Sol tarafta eşleşen satır bulunmayan satırlar için sonuç kümesi null değerini içerecektir. **RIGHT JOIN** aynı zamanda **RIGHT OUTER JOIN** olarak da bilinir.

```
SELECT table1.column1,table1.column2,table2.column1,...  
FROM table1  
RIGHT JOIN table2  
ON table1.matching_column = table2.matching_column;
```

table1: First table.

table2: Second table

matching_column: Column common to both the tables.



FULL JOIN

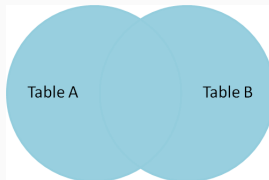
- **FULL JOIN**, hem **LEFT JOIN** hem de **RIGHT JOIN** sonuçlarını birleştirerek sonuç kümesini oluşturur. Sonuç kümesi her iki tablodaki tüm satırları içerecektir. Eşleşmenin olmadığı satırlar için sonuç kümesi NULL değerleri içerecektir.

```
SELECT table1.column1,table1.column2,table2.column1,...  
FROM table1  
FULL JOIN table2  
ON table1.matching_column = table2.matching_column;
```

table1: First table.

table2: Second table

matching_column: Column common to both the tables.



- `production.products` tablosunu ve `production.categories` tablosunu `inner join` kullanarak `category_id` sütunları üzerinden birleştirelim.
- `inner join` yerine `left join`, `right join` ve `full join` ile değiştirerek sonuçları görelim.
- `inner join` yerine `where` şartı kullanarak aynı sonuçları listeyelim.

Inner JOIN Sorgu

Debug

SQLQuery2.sql - DES...SS.master (sa (52)) SQLQuery1.sql - DE...gramLama (sa (54))* X

```
SELECT p.*, c.category_name FROM production.products p inner join  
production.categories c on p.category_id= c.category_id
```

214 %

Results Messages

	product_id	product_name	brand_id	category_id	model_year	list_price	category_name
1	1	Trek 820 - 2016	9	6	2016	379.99	Mountain Bikes
2	2	Ritchey Timberwolf Frameset - 2016	5	6	2016	749.99	Mountain Bikes
3	3	Surly Wednesday Frameset - 2016	8	6	2016	999.99	Mountain Bikes
4	4	Trek Fuel EX 8 29 - 2016	9	6	2016	2899.99	Mountain Bikes
5	5	Heller Shogamaw Frame - 2016	3	6	2016	1320.99	Mountain Bikes
6	6	Surly Ice Cream Truck Frameset - 2016	8	6	2016	469.99	Mountain Bikes
7	7	Trek Slash 8 27.5 - 2016	9	6	2016	3999.99	Mountain Bikes
8	8	Trek Remedy 29 Carbon Frameset - 2016	9	6	2016	1799.99	Mountain Bikes
9	9	Trek Conduit+ - 2016	9	5	2016	2999.99	Electric Bikes
10	10	Surly Straggler - 2016	8	4	2016	1549.00	Cyclocross Bicycles
11	11	Surly Straggler 650b - 2016	8	4	2016	1680.99	Cyclocross Bicycles
12	12	Electra Townie Original 21D - 2016	1	3	2016	549.99	Cruisers Bicycles
13	13	Electra Cruiser 1 (24-Inch) - 2016	1	3	2016	269.99	Cruisers Bicycles
14	14	Electra Girl's Hawaii 1 (16-inch) - 2015/2016	1	3	2016	269.99	Cruisers Bicycles
15	15	Electra Moto 1 - 2016	1	3	2016	529.99	Cruisers Bicycles
16	16	Electra Townie Original 7D EQ - 2016	1	3	2016	599.99	Cruisers Bicycles
17	17	Pure Cycles Vine 8-Speed - 2016	4	3	2016	429.00	Cruisers Bicycles
18	18	Pure Cycles Western 3-Speed - Women's - 2015/2016	4	3	2016	449.00	Cruisers Bicycles
19	19	Pure Cycles William 3-Speed - 2016	4	3	2016	449.00	Cruisers Bicycles
20	20	Electra Townie Original 7D EQ - Women's - 2016	1	3	2016	599.99	Cruisers Bicycles
21	21	Electra Cruiser 1 (24-Inch) - 2016	1	1	2016	269.99	Children Bicycles
22	22	Electra Girl's Hawaii 1 (16-inch) - 2015/2016	1	1	2016	269.99	Children Bicycles
23	23	Electra Girl's Hawaii 1 (20-inch) - 2015/2016	1	1	2016	299.99	Children Bicycles

WHERE ile inner JOIN Sorgu

SQLQuery2.sql - DES...SS.master (sa (52)) SQLQuery1.sql - DE...gramLama (sa (54))

```
SELECT p.*, c.category_name FROM production.products p,  
production.categories c where p.category_id= c.category_id
```

214 %

Results Messages

	product_id	product_name	brand_id	category_id	model_year	list_price	category_name
1	1	Trek 820 - 2016	9	6	2016	379.99	Mountain Bikes
2	2	Ritchey Timberwolf Frameset - 2016	5	6	2016	749.99	Mountain Bikes
3	3	Surly Wednesday Frameset - 2016	8	6	2016	999.99	Mountain Bikes
4	4	Trek Fuel EX 8 29 - 2016	9	6	2016	2899.99	Mountain Bikes
5	5	Heller Shagawaw Frame - 2016	3	6	2016	1320.99	Mountain Bikes
6	6	Surly Ice Cream Truck Frameset - 2016	8	6	2016	469.99	Mountain Bikes
7	7	Trek Slash 8 27.5 - 2016	9	6	2016	3999.99	Mountain Bikes
8	8	Trek Remedy 29 Carbon Frameset - 2016	9	6	2016	1799.99	Mountain Bikes
9	9	Trek Conduit+ - 2016	9	5	2016	2999.99	Electric Bikes
10	10	Surly Straggler - 2016	8	4	2016	1549.00	Cyclocross Bicycles
11	11	Surly Straggler 650b - 2016	8	4	2016	1680.99	Cyclocross Bicycles
12	12	Electra Townie Original 21D - 2016	1	3	2016	549.99	Cruisers Bicycles
13	13	Electra Cruiser 1 (24-Inch) - 2016	1	3	2016	269.99	Cruisers Bicycles
14	14	Electra Girl's Hawaii 1 (16-inch) - 2015/2016	1	3	2016	269.99	Cruisers Bicycles
15	15	Electra Moto 1 - 2016	1	3	2016	529.99	Cruisers Bicycles
16	16	Electra Townie Original 7D EQ - 2016	1	3	2016	599.99	Cruisers Bicycles
17	17	Pure Cycles Vine 8-Speed - 2016	4	3	2016	429.00	Cruisers Bicycles
18	18	Pure Cycles Western 3-Speed - Women's - 2015/2016	4	3	2016	449.00	Cruisers Bicycles
19	19	Pure Cycles William 3-Speed - 2016	4	3	2016	449.00	Cruisers Bicycles
20	20	Electra Townie Original 7D EQ - Women's - 2016	1	3	2016	599.99	Cruisers Bicycles
21	21	Electra Cruiser 1 (24-Inch) - 2016	1	1	2016	269.99	Children Bicycles
22	22	Electra Girl's Hawaii 1 (16-inch) - 2015/2016	1	1	2016	269.99	Children Bicycles
23	23	Electra Girl's Hawaii 1 (20-inch) - 2015/2016	1	1	2016	299.99	Children Bicycles

DISTINCT

- **DISTINCT** anahtar sözcüğü, select anahtar sözcüğüyle birlikte kullanılır.
- Belirli bir sütunda/tablodaki yinelenen değerlerden kaçınmaya ihtiyaç duyulduğunda faydalıdır.
- **DISTINCT** kullandığımızda yalnızca benzersiz değerleri elde ederiz.
- **DISTINCT** faydalı bir kullanımı olmasına rağmen yavaş çalışır. Büyük kayıtları analiz ederken çok tavsiye etmem.

```
SELECT DISTINCT column1, column2
```

```
FROM table_name
```

DISTINCT Sorgular

- production.products tablosunda bulunan kayıtlardan tekil category_id değerlerini listeyelim.

SQL Server 12.0.2000

```
select distinct p.category_id from production.products p
```

```
select p.category_id from production.products p
```

100 %

Results Messages

	category_id
1	1
2	2
3	3
4	4
5	5
6	6
7	7

	category_id
1	6
2	6
3	6
4	6
5	6
6	6
7	6
8	6
9	5
10	4
11	4
12	3

MAX-MIN

- **MAX()** fonksiyonu seçilen sütunun en büyük değerini sağlar.
- SQL sorularında **MAX(sütun adı)** işlevleri sütundaki en büyük değeri döndürür.
- Benzer şekilde **MIN()** fonksiyonu da kayıtlardan min değerini döndürecektir.

```
SELECT MAX(column_name)
```

```
FROM table_name
```

```
WHERE condition;
```


- production.products tablosunda bulunan kayıtlardan en düşük ve en yüksek fiyata sahip olan ürünleri karegorisi ile birlikte sorgulayın.
- ilgili sorgu cümleciğini iç içe sorgularak kullanarak tek sorgu cümleciği içerisinde yapınız.

MAX-MIN Sorgu Sonucu

```
select p.*, (select c.category_name from production.categories c where c.category_id=p.category_id) as CATEGORI_NAME  
from production.products p where p.list_price=(select MIN(b.list_price) from production.products b)  
or p.list_price=(select MAX(a.list_price) from production.products a)
```

146 %

Results Messages

	product_id	product_name	brand_id	category_id	model_year	list_price	CATEGORI_NAME
1	155	Trek Domane SLR 9 Disc - 2018	9	7	2018	11999.99	Road Bikes
2	263	Strider Classic 12 Balance Bike - 2018	6	1	2018	89.99	Children Bicycles