



CHAPTER 17: VIEWS

2023

CHAPTER 17: VIEWS

Contents:

1. Introduction to views
2. Creating views
3. Viewing data using a view
4. Updating a view
5. Materialized views

VIEWS

A VIEW in SQL Server is like a virtual table that contains data from one or multiple tables.

You can get the result set by creating a query with the data from one or more tables using the SELECT statement.

For example, the following statement returns the product name, brand, and list price of all products from the products and brands tables:

```
SELECT
```

```
    product_name, brand_name, list_price
```

```
FROM  products p
```

```
INNER JOIN brands b
```

```
    ON b.brand_id = p.brand_id;
```

VIEWS CONTINUED..

The next time you need the same result set, you can save this query to a text file, open it, and run it again.

SQL Server provides a better way to store this query through a view in the database catalog.

The view is a named query that is stored in the database catalog for later reference.

Advantages of a view:

1. Security
2. It's Simple and not complex
3. Consistency

CREATE A VIEW EXAMPLE

1. Create the view

```
CREATE VIEW sales.product_info
AS
SELECT
    product_name,
    brand_name,
    list_price
FROM
    production.products p
INNER JOIN production.brands b
    ON b.brand_id = p.brand_id;
```

100 %

Messages

Commands completed successfully.

Completion time: 2022-01-13T10:51:28.6437701+02:00

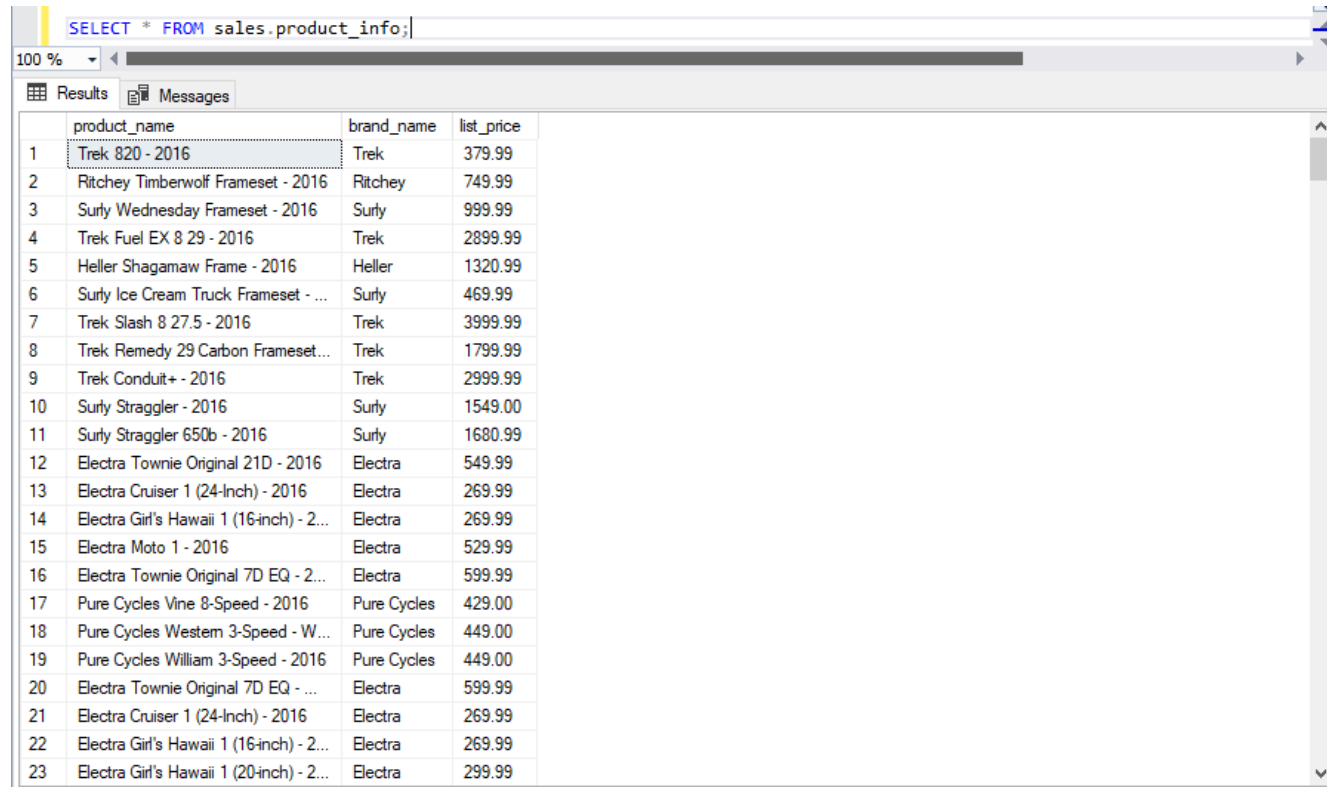
CREATE A VIEW EXAMPLE

```
CREATE VIEW view_name AS  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

```
CREATE VIEW [Brazil Customers] AS  
SELECT CustomerName, ContactName  
FROM Customers  
WHERE Country = 'Brazil';
```

VIEWING THE VIEW

2. you can reference to the view in the SELECT statement like a table



```
SELECT * FROM sales.product_info;
```

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

VIEWS (CREATING AND RUNNING VIEWS)

```
CREATE VIEW view_name AS  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

```
CREATE VIEW [Products Above Average Price] AS  
SELECT ProductName, Price  
FROM Products  
WHERE Price > (SELECT AVG(Price) FROM Products);
```

```
SELECT * FROM [Products Above Average Price];
```

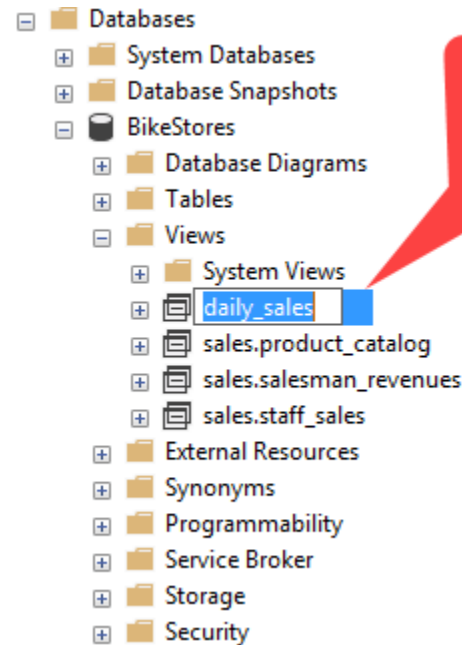
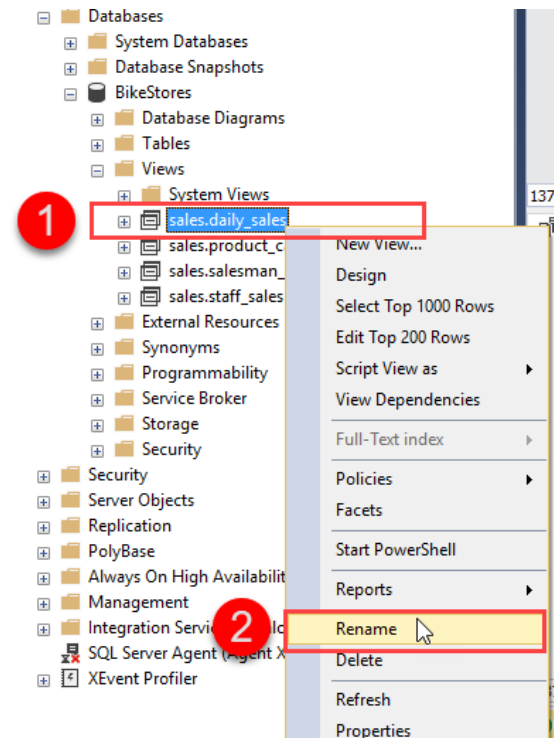
```
CREATE VIEW [Brazil Customers] AS  
SELECT CustomerName, ContactName  
FROM Customers  
WHERE Country = 'Brazil';
```

```
SELECT * FROM [Brazil Customers];
```


RENAME VIEW

First, in Object Explorer, expand the Databases, choose the database name which contains the view that you want to rename and expand the Views folder.

Second, right-click the view that you want to rename and select Rename.



UPDATING/DELETING VIEWS

A view can be updated with the CREATE OR REPLACE VIEW statement

```
CREATE OR REPLACE VIEW view_name AS
```

```
SELECT column1, column2, ...
```

```
FROM table_name
```

```
WHERE condition;
```

Deleting a view

```
DROP VIEW view_name;
```

```
DROP VIEW [Brazil Customers];
```

```
CREATE OR REPLACE VIEW [Brazil  
Customers] AS
```

```
SELECT CustomerName, ContactName, City
```

```
FROM Customers
```

```
WHERE Country = 'Brazil'
```

UPDATING A VIEW

```
CREATE OR REPLACE VIEW view_name AS  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

```
CREATE OR REPLACE VIEW [Brazil Customers] AS  
SELECT CustomerName, ContactName, City  
FROM Customers  
WHERE Country = 'Brazil';
```

DROP VIEW

To remove a view from a database, you use the DROP VIEW statement as follows:

The syntax: `DROP VIEW [IF EXISTS] schema_name.view_name;`

The above syntax specifies the name of the view to remove after the DROPVIEW keyword.

If the view belongs to a schema, you must also explicitly specify the name of the schema to which the view belongs.

DROPPING A VIEW

```
DROP VIEW view_name;
```

```
DROP VIEW [Brazil Customers];
```



SQL Server returns an error when trying to delete a view that does not exist.

The **IF EXISTS** clause prevents you from getting an error when you delete a view that does not exist.

The syntax to drop multiple views:

```
DROP VIEW [IF EXISTS] schema_name.view_name1, schema_name.view_name2,...;
```

A vertical decorative bar on the left side of the slide featuring a repeating geometric pattern of interlocking circles in teal and white.

Practice VIEWS

1. Create a view called `listings_under_100` where the query is all listings that are below 100 in price.
2. Create a view called `listings_private_rooms` where the query is all listings that are private rooms.
3. Delete the view `listings_under_100`.
4. Update the view `listings_private_rooms` to show hotel rooms

MATERIALIZED VIEWS

A materialized view is a database object that contains the result of a query that has been precomputed and stored in the database. Materialized views can be thought of as a cache of query results that can be used to improve performance and reduce the workload on the database server.


```
CREATE MATERIALIZED VIEW mv_sales_by_region
```

```
AS
```

```
SELECT region, SUM(sales) AS total_sales
```

```
FROM sales_table
```

```
GROUP BY region;
```

Once the materialized view is created, you can refresh it using the REFRESH MATERIALIZED VIEW statement, which updates the view with the latest data from the underlying table(s).

```
REFRESH MATERIALIZED VIEW mv_sales_by_region;
```