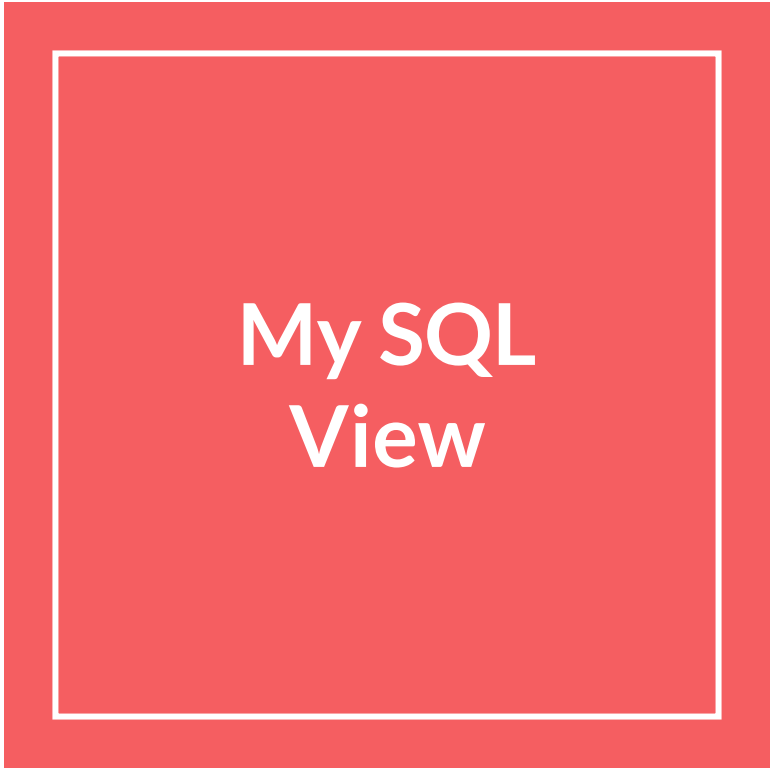

Objectives

1. View definition
2. Creating, modifying and dropping a View
3. Finding View
4. Creating a View with a Union

1. View definition

A view is a specific look on data from one or more tables. It can arrange data in some specific order, highlight or hide some data. A view consists of a stored query accessible as a virtual table composed of the result set of a query. Unlike ordinary tables a view does not form part of the physical schema. It is a dynamic, virtual table computed or collated from data in the database.

A view is a pseudo table. It is a stored query which looks like a table. And it can be referenced like a table.

Views can restrict users to specific rows or columns and thus enhance security. They can be used to join columns from multiple tables, so that they look like a single table. They can be used to provide aggregate information.

1. View definition

- There are several restrictions that apply to views. Here are some of them:
  - The SELECT statement cannot contain a subquery
  - The SELECT statement cannot refer to system or user variables
  - Any table or view referred to in the definition must exist
  - A temporary VIEW cannot be created
  - A VIEW cannot be associated with a trigger

2. Creating, modifying and dropping a View

In the next example, we create a simple view. We use CREATE VIEW syntax to create a view.

This is our data, upon which we create the view.

```
mysql> SELECT * FROM Cars;
+----+------------+--------+
| Id | Name       | Cost   |
+----+------------+--------+
|  1 | Audi       |  52642 |
|  2 | Mercedes   |  57127 |
|  3 | Skoda      |   9000 |
|  4 | Volvo      |  29000 |
|  5 | Bentley    | 350000 |
|  6 | Citroen    |  21000 |
|  7 | Hummer     |  41400 |
|  8 | Volkswagen |  21600 |
+----+------------+--------+
```

2. Creating, modifying and dropping a View

We create a view CheapCars. These are cars which cost under 25000.

```
CREATE VIEW CheapCars AS  SELECT Name FROM Cars WHERE Cost<25000;
```

A view is a database object than can be queried. There are three cars which are considered to be cheap.

```
mysql> SELECT * FROM CheapCars;
+------------+
| Name       |
+------------+
| Skoda      |
| Citroen    |
| Volkswagen |
+------------+
```

2. Creating, modifying and dropping a View

We can redefine a view. Say we now consider a car to be cheap if it costs under 30000. We use the ALTER VIEW statement to modify our view.

```
mysql> ALTER VIEW CheapCars AS SELECT Name FROM Cars WHERE Cost<30000;

mysql> SELECT * FROM CheapCars;
+------------+
| Name       |
+------------+
| Skoda      |
| Volvo      |
| Citroen    |
| Volkswagen |
+------------+
```

2. Creating, modifying and dropping a View

What happens to a view if we delete a table, from which the data is selected?

```
mysql> DROP TABLE Cars;
```

```
mysql> SELECT * FROM CheapCars;
ERROR 1356 (HY000): View 'mydb.CheapCars' references invalid table(s)
or column(s) or function(s) or definer/invoker of view lack rights to use them
```

Querying the view we receive the above error.

```
mysql> SOURCE cars.sql
```

```
mysql> SELECT * FROM CheapCars;
+------------+
| Name       |
+------------+
| Skoda      |
| Citroen    |
| Volkswagen |
+------------+
```

When we recreate the table the view works again.

```
mysql> DROP VIEW CheapCars;
```

Finally, a view is deleted with the DROP VIEW syntax.

3. Finding View

We will mention several ways how to find views in MySQL database.

```
mysql> SHOW FULL TABLES;
+----------------+------------+
| Tables_in_mydb | Table_type |
+----------------+------------+
| AA             | BASE TABLE |
...
| Chars          | BASE TABLE |
| CheapCars      | VIEW       |
| Customers      | BASE TABLE |
| Dates          | BASE TABLE |
| Decimals       | BASE TABLE |
| FavoriteCars   | VIEW       |
...
```

3. Finding View

We can list all tables in a database with a SHOW FULL TABLES statement. In the Table_type column we can see, whether it is a table or a view.

```
mysql> SELECT TABLE_NAME, TABLE_TYPE FROM information_schema.TABLES;
+---------------------------------------+-------------+
| TABLE_NAME                            | TABLE_TYPE  |
+---------------------------------------+-------------+
| CHARACTER_SETS                        | SYSTEM VIEW |
| COLLATIONS                            | SYSTEM VIEW |
| COLLATION_CHARACTER_SET_APPLICABILITY | SYSTEM VIEW |
| COLUMNS                               | SYSTEM VIEW |
| COLUMN_PRIVILEGES                     | SYSTEM VIEW |
| ENGINES                               | SYSTEM VIEW |
...
| Chars                                 | BASE TABLE  |
| CheapCars                             | VIEW        |
| Customers                             | BASE TABLE  |
| Dates                                 | BASE TABLE  |
| Decimals                              | BASE TABLE  |
| FavoriteCars                          | VIEW        |
...
```

3. Finding View

In the information_schema database there is a TABLES table. The TABLE_NAME and TABLE_TYPE columns give us information about table names and their types.

```
mysql> SELECT TABLE_NAME FROM information_schema.VIEWS;
+--------------+
| TABLE_NAME   |
+--------------+
| CheapCars    |
| FavoriteCars |
+--------------+
```

This is the most straightforward way to find views. We query the VIEWS table of the information_schema database.

4. Creating a View with a union

The UNION operator is used to combine result-sets of two or more SELECT statements.

Each select must have the same number of columns.

```
mysql> CREATE VIEW FavoriteCars AS
    -> SELECT * FROM Cars WHERE Id=7
    -> UNION SELECT * FROM Cars WHERE Id=4
    -> UNION SELECT * FROM Cars WHERE Id=5;
```

We create a view called FavoriteCars. In this view, we have three rows which are considered to be favourite. There are three SELECT statements combined with a UNION operator. This is a SELECT from the view.

```
mysql> SELECT * FROM FavoriteCars;
+----+---------+--------+
| Id | Name    | Cost   |
+----+---------+--------+
|  7 | Hummer  |  41400 |
|  4 | Volvo   |  29000 |
|  5 | Bentley | 350000 |
+----+---------+--------+
```