

Experiment No 10

Aim: Implementation of Views and Triggers

Class: SE Comp

Year: 2020-21

Performed by: Danyl Fernandes, 72

Performance Date: 22-04-2021

Views:

- A view is a database object that has no values. Its contents are based on the base table.
- It contains rows and columns similar to the real table. In MySQL, the View is a **virtual table** created by a query by joining one or more tables.
- It is operated similarly to the base table but does not contain any data of its own. The View and table have one main difference that the views are definitions built on top of other tables (or views).
- If any changes occur in the underlying table, the same changes reflected in the View also.

Syntax:

```
CREATE [OR REPLACE] VIEW view_name AS  
SELECT columns  
FROM tables  
[WHERE conditions];
```

Parameters:

The view syntax contains the following parameters:

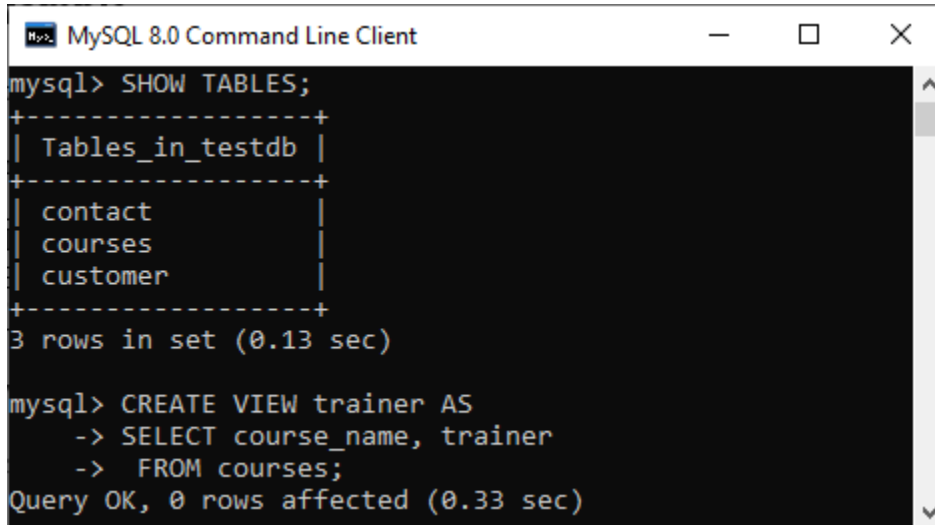
OR REPLACE: It is optional. It is used when a VIEW already exists. If you do not specify this clause and the VIEW already exists, the CREATE VIEW statement will return an error.

view_name: It specifies the name of the VIEW that you want to create in MySQL.

WHERE conditions: It is also optional. It specifies the conditions that must be met for the records to be included in the VIEW.

Example:

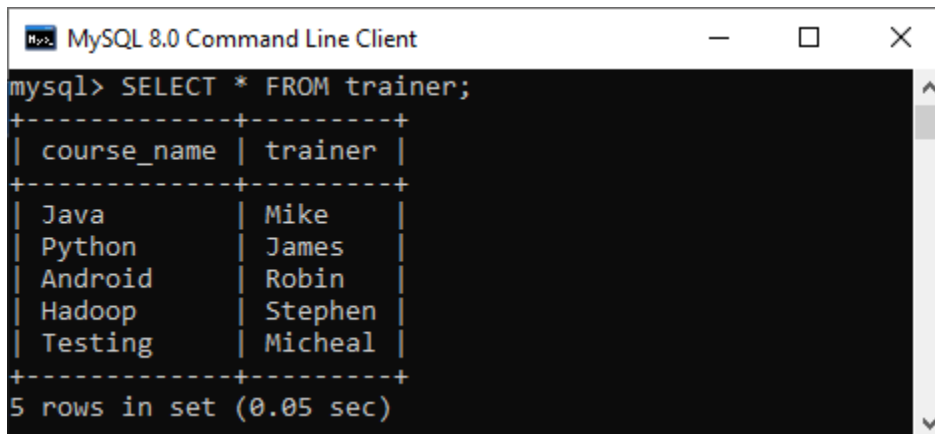
```
CREATE VIEW trainer AS  
SELECT course_name, trainer  
FROM courses;
```



The screenshot shows a MySQL 8.0 Command Line Client window. The user has entered the command `SHOW TABLES;` and the output shows three tables: `contact`, `courses`, and `customer`. Then, the user enters the command `CREATE VIEW trainer AS SELECT course_name, trainer FROM courses;` and the output shows "Query OK, 0 rows affected (0.33 sec)".

```
mysql> SHOW TABLES;  
+-----+  
| Tables_in_testdb |  
+-----+  
| contact          |  
| courses          |  
| customer         |  
+-----+  
3 rows in set (0.13 sec)  
  
mysql> CREATE VIEW trainer AS  
-> SELECT course_name, trainer  
-> FROM courses;  
Query OK, 0 rows affected (0.33 sec)
```

```
SELECT * FROM trainer;
```



The screenshot shows a MySQL 8.0 Command Line Client window. The user has entered the command `SELECT * FROM trainer;` and the output shows five rows of data: `Java` by `Mike`, `Python` by `James`, `Android` by `Robin`, `Hadoop` by `Stephen`, and `Testing` by `Micheal`. The output is displayed in a table format with headers `course_name` and `trainer`.

```
mysql> SELECT * FROM trainer;  
+-----+-----+  
| course_name | trainer |  
+-----+-----+  
| Java       | Mike   |  
| Python     | James  |  
| Android    | Robin  |  
| Hadoop     | Stephen|  
| Testing    | Micheal|  
+-----+-----+  
5 rows in set (0.05 sec)
```

Triggers:

- A trigger in MySQL is a set of SQL statements that reside in a system catalog.
- **It is a special type of stored procedure that is invoked automatically in response to an event.**
- Each trigger is associated with a table, which is activated on any DML statement such as **INSERT, UPDATE, or DELETE**.
- A trigger is called a special procedure because it cannot be called directly like a stored procedure.
- The main difference between the trigger and procedure is that a trigger is called automatically when a data modification event is made against a table.
- In contrast, a stored procedure must be called explicitly.

Syntax:

```
CREATE TRIGGER trigger_name
(AFTER | BEFORE) (INSERT | UPDATE | DELETE)
ON table_name FOR EACH ROW
BEGIN
    --variable declarations
    --trigger code
END;
```

Types of Triggers:

We can define the maximum six types of actions or events in the form of triggers:

- **Before Insert:** It is activated before the insertion of data into the table.
- **After Insert:** It is activated after the insertion of data into the table.
- **Before Update:** It is activated before the update of data in the table.
- **After Update:** It is activated after the update of the data in the table.
- **Before Delete:** It is activated before the data is removed from the table.
- **After Delete:** It is activated after the deletion of data from the table.

Limitations of Using Triggers:

- MySQL triggers do not allow the use of all validations; they only provide extended validations. **For example**, we can use the NOT NULL, UNIQUE, CHECK and FOREIGN KEY constraints for simple validations.
- Triggers are invoked and executed invisibly from the client application. Therefore, it isn't easy to troubleshoot what happens in the database layer.
- Triggers may increase the overhead of the database server.

Query:

```
create view full_sailor_view as select * from sailor;
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0060 seconds.)

```
create view full_sailor_view as select * from sailor
```

✓ Showing rows 0 - 20 (21 total, Query took 0.0019 seconds.)

```
SELECT * FROM `full_sailor_view`
```

☐ Show all | Number of rows: Filter rows:

+ Options

<div><div><div></div><div></div><div></div></div></div>				sid	sname	address	rating	age
<input type="checkbox"/>		Edit	<div><div><div></div><div></div><div></div></div></div> Copy <div><div></div></div> Delete	1	Rahul	Mumbai	5	34
<input type="checkbox"/>		Edit	<div><div><div></div><div></div><div></div></div></div> Copy <div><div></div></div> Delete	2	Danyl	Pune	5	21
<input type="checkbox"/>		Edit	<div><div><div></div><div></div><div></div></div></div> Copy <div><div></div></div> Delete	3	Shrinath	Hyderabad	5	22
<input type="checkbox"/>		Edit	<div><div><div></div><div></div><div></div></div></div> Copy <div><div></div></div> Delete	4	Jack	Pennsylvania	4	32
<input type="checkbox"/>		Edit	<div><div><div></div><div></div><div></div></div></div> Copy <div><div></div></div> Delete	5	Sean	Ontario	3	30
<input type="checkbox"/>		Edit	<div><div><div></div><div></div><div></div></div></div> Copy <div><div></div></div> Delete	6	Matt	Texas	3	29
<input type="checkbox"/>		Edit	<div><div><div></div><div></div><div></div></div></div> Copy <div><div></div></div> Delete	7	Hans	Berlin	4	28
<input type="checkbox"/>		Edit	<div><div><div></div><div></div><div></div></div></div> Copy <div><div></div></div> Delete	8	Drishyam	Telangana	2	29

Query:

```
create view rating_3 as select * from sailor where rating = 3;
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0065 seconds.)

```
create view rating_3 as select * from sailor where rating = 3
```

✓ Showing rows 0 - 4 (5 total, Query took 0.0012 seconds.)
















```
SELECT * FROM `rating_3`
```

☐ Show all

Number of rows: 25 ▼

Filter rows:

+ Options

				sid	sname	address	rating	age
<input type="checkbox"/>		 Copy	 Delete	5	Sean	Ontario	3	30
<input type="checkbox"/>		 Copy	 Delete	6	Matt	Texas	3	29
<input type="checkbox"/>		 Copy	 Delete	10	Bob	Alabama	3	54
<input type="checkbox"/>		 Copy	 Delete	15	Tyrone	SF	3	41
<input type="checkbox"/>		 Copy	 Delete	16	Daniel	Connecticut	3	49

Query:

```
insert into sailor values (22, "Eddie", "Phoenix", 3, 30);
```

✓ Showing rows 0 - 5 (6 total, Query took 0.0011 seconds.)



















```
SELECT * FROM `rating_3`
```

☐ Show all

Number of rows: 25 ▾

Filter rows:

+ Options

				sid	sname	address	rating	age
<input type="checkbox"/>	 Edit	 Copy	 Delete	5	Sean	Ontario	3	30
<input type="checkbox"/>	 Edit	 Copy	 Delete	6	Matt	Texas	3	29
<input type="checkbox"/>	 Edit	 Copy	 Delete	10	Bob	Alabama	3	54
<input type="checkbox"/>	 Edit	 Copy	 Delete	15	Tyrone	SF	3	41
<input type="checkbox"/>	 Edit	 Copy	 Delete	16	Daniel	Connecticut	3	49
<input type="checkbox"/>	 Edit	 Copy	 Delete	22	Eddie	Phoenix	3	30

Delete Trigger:

Query:

```
create table trash (  
    sid int,  
    sname varchar(30),  
    rating int,  
    age int  
);
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0293 seconds.)

```
create table trash(sid int, sname varchar(30), rating int, age int)
```

Query:

```
DELIMITER $  
CREATE or REPLACE TRIGGER old_record_to_trash BEFORE DELETE on  
sailor  
FOR EACH ROW  
BEGIN  
    INSERT INTO trash VALUES (old.sid, old.sname, old.rating,  
old.age);  
END;  
$
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0126 seconds.)

```
CREATE or REPLACE TRIGGER old_record_to_trash BEFORE DELETE on sailor FOR EACH ROW BEGIN INSERT INTO trash VALUES (old.sid, old.sname,  
old.rating, old.age); END;
```

[\[Edit inline\]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

Query:


```
select * from trash
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)

```
SELECT * FROM `trash`
```

sid	sname	rating	age
-----	-------	--------	-----

Query results operations

 Create view

Query:

```
delete from sailor where sid = 21
```

✓ 1 row affected. (Query took 0.0059 seconds.)

```
delete from sailor where sid = 21
```

Query:

```
select * from trash
```

✓ Showing rows 0 - 0 (1 total, Query took 0.0009 seconds.)

```
SELECT * FROM `trash`
```

☐ Show all | Number of rows: Filter rows:

+ Options

sid	sname	rating	age
21	George	5	45

Before Update Trigger:

Query:

```
create table update_logs (  
    sid int,  
    old_sname varchar(30),  
    updated_by varchar(30),  
    Updated_on varchar(30)  
);
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0428 seconds.)

```
create table update_logs ( sid int, old_sname varchar(30), updated_by varchar(30), updated_on varchar(30) )
```

[\[Edit inline\]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

Query:

```
DELIMITER $  
CREATE or REPLACE TRIGGER update_logs BEFORE UPDATE on sailor  
FOR EACH ROW  
BEGIN  
    INSERT INTO update_logs VALUES (  
        old.sid,  
        old.sname,  
        USER(),  
        SYSDATE()  
    );  
END;  
$
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0124 seconds.)

```
CREATE or REPLACE TRIGGER update_logs BEFORE UPDATE on sailor FOR EACH ROW BEGIN INSERT INTO update_logs VALUES ( old.sid, old.sname,  
USER(), SYSDATE() ); END;
```

[\[Edit inline\]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

Query:


```
select * from update_logs
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0010 seconds.)

```
SELECT * FROM `update_logs`
```

sid	old_sname	updated_by	updated_on
-----	-----------	------------	------------

Query results operations

 Create view

Query:

```
update sailor set name = "Abraham" where sid = 11
```

✓ 0 rows affected. (Query took 0.0098 seconds.)

```
update sailor set sname = "Abraham" where sid = 11
```

Query:

```
select * from update_logs
```

✓ Showing rows 0 - 1 (2 total, Query took 0.0018 seconds.)

```
SELECT * FROM `update_logs`
```

☐ Show all | Number of rows: Filter rows:

+ Options

sid	old_sname	updated_by	updated_on
11	Abraham	root@localhost	2021-04-21 14:20:28

After Update Trigger:

Query:

```
create table update_logs (  
    sid int,  
    new_sname varchar(30),  
    updated_by varchar(30),  
    updated_on varchar(30)  
);
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0302 seconds.)

```
create table update_logs ( sid int, new_sname varchar(30), updated_by varchar(30), updated_on varchar(30) )
```

[\[Edit inline\]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

Query:

```
DELIMITER $  
CREATE or REPLACE TRIGGER after_update_logs AFTER UPDATE on  
sailor  
FOR EACH ROW  
BEGIN  
    INSERT INTO update_logs VALUES (  
        old.sid,  
        new.sname,  
        USER(),  
        SYSDATE()  
    );  
END;  
$
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0170 seconds.)

```
CREATE or REPLACE TRIGGER after_update_logs AFTER UPDATE on sailor FOR EACH ROW BEGIN INSERT INTO update_logs VALUES ( old.sid,  
new.sname, USER(), SYSDATE() ); END;
```

[\[Edit inline\]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

Query:


```
select * from update_logs
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0015 seconds.)

```
select * from update_logs
```

sid	new_sname	updated_by	updated_on
-----	-----------	------------	------------

Query results operations

 Create view

Query:

```
update sailor set sname = "Shayla" where sid = 18
```

✓ 1 row affected. (Query took 0.0116 seconds.)

```
update sailor set sname = "Shayla" where sid = 18
```

Query:

```
select * from update_logs
```

✓ Showing rows 0 - 0 (1 total, Query took 0.0017 seconds.)

```
SELECT * FROM `update_logs`
```

☐ Show all | Number of rows: 25 ▾ Filter rows:

+ Options

sid	new_sname	updated_by	updated_on
18	Shayla	root@localhost	2021-04-21 14:31:41

After Insert Trigger:

Query:

```
create table insert_logs (  
    sid int,  
    new_sname varchar(30),  
    updated_by varchar(30),  
    updated_on varchar(30)  
);
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0385 seconds.)

```
create table insert_logs ( sid int, new_sname varchar(30), updated_by varchar(30), updated_on varchar(30) )
```

[\[Edit inline\]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

Query:

```
DELIMITER $  
CREATE or REPLACE TRIGGER after_update_logs AFTER INSERT on  
sailor  
FOR EACH ROW  
BEGIN  
    INSERT INTO update_logs VALUES (  
        old.sid,  
        new.sname,  
        USER(),  
        SYSDATE()  
    );  
END;  
$
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0143 seconds.)

```
CREATE or REPLACE TRIGGER after_insert_logs AFTER INSERT on sailor FOR EACH ROW BEGIN INSERT INTO insert_logs VALUES ( new.sid,  
new.sname, USER(), SYSDATE() ); END;
```

[\[Edit inline\]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

Query:


```
select * from insert_logs
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0016 seconds.)

```
SELECT * FROM `insert_logs`
```

sid	new_sname	updated_by	updated_on
-----	-----------	------------	------------

Query results operations

 Create view

Query:

```
insert into sailor values (23, "Bob", "NYC", 2, 32)
```

✓ 1 row inserted. (Query took 0.0053 seconds.)

```
insert into sailor values (23, "Bob", "NYC", 2, 32)
```

Query:

```
select * from insert_logs
```

✓ Showing rows 0 - 0 (1 total, Query took 0.0018 seconds.)

```
SELECT * FROM `insert_logs`
```

☐ Show all | Number of rows: Filter rows:

+ Options

sid	new_sname	updated_by	updated_on
23	Bob	root@localhost	2021-04-21 15:13:52