

Hands-on Lab: Using Views in MySQL using phpMyAdmin



**Skills
Network**

Estimated time needed: 20 minutes

In this lab, you will learn about using views. In SQL, a view is an alternative way of representing data that exists in one or more tables. Just like a real table, it contains rows and columns. The fields in a view are fields from one or more real tables in the database. Though views can be queried like a table, views are dynamic; only the definition of the view is stored, not the data.

Objectives

After completing this lab, you will be able to:

- Create a View and show a selection of data for a given table
- Update a View to combine two or more tables in meaningful ways
- Drop a created View

Software Used in this Lab

In this lab, you will use [MySQL](#). MySQL is a Relational Database Management System (RDBMS) designed to efficiently store, manipulate, and retrieve data.



To complete this lab you will utilize MySQL relational database service available as part of IBM Skills Network Labs (SN Labs) Cloud IDE. SN Labs is a virtual lab environment used in this course.

Database Used in this Lab

The database used in this lab is a sample HR database. This HR database schema consists of five tables called EMPLOYEES, JOB_HISTORY, JOBS, DEPARTMENTS, and LOCATIONS. Each table has a few rows of sample data. The following diagram shows the tables for the HR database:

SAMPLE HR DATABASE TABLES

EMPLOYEES

EMP_ID	F_NAME	L_NAME	SSN	B_DATE	SEX	ADDRESS	JOB_ID	SALARY	MANAGER_ID	DEP
E1001	John	Thomas	123456	1976-01-09	M	5631 Rice, OakPark,IL	100	100000	30001	2
E1002	Alice	James	123457	1972-07-31	F	980 Berry Ln, Elgin,IL	200	80000	30002	5
E1003	Steve	Wells	123458	1980-08-10	M	291 Springs, Gary,IL	300	50000	30002	5

JOB_HISTORY

EMPL_ID	START_DATE	JOBS_ID	DEPT_ID
E1001	2000-01-30	100	2
E1002	2010-08-16	200	5
E1003	2016-08-10	300	5

JOBS

JOB_IDENT	JOB_TITLE	MIN_SALARY	MAX_SA
100	Sr. Architect	60000	100000
200	Sr.SoftwareDeveloper	60000	80000
300	Jr.SoftwareDeveloper	40000	60000

DEPARTMENTS

DEPT_ID_DEP	DEP_NAME	MANAGER_ID	LOC_ID
2	Architect Group	30001	L0001
5	Software Development	30002	L0002
7	Design Team	30003	L0003

LOCATIONS

LOCT_ID	DEP_ID_LOC
L0001	2
L0002	5
L0003	7

Follow the steps below to create the database and the tables.

1. Open the MySQL interface from Skills Network menu.
2. Create a new database and name it HR.

3. Load and execute the script shared in the link below to create the necessary tables.

[HR Database Create Tables Script.sql](#)

4. Load all the tables with the data available in the CSV files shared below.

[Departments.csv](#)
[Employees.csv](#)
[Jobs.csv](#)
[Locations.csv](#)
[JobsHistory.csv](#)

Note: Please refer to the instruction in the lab "[Create and Load Tables using SQL Scripts](#)" for instructions regarding loading scripts in MySQL.

Task 1: Create a View

In this exercise, you will create a View and show a selection of data for a given table.

1. Let's create a view called EMPSALARY to display salary along with some basic sensitive data of employees from the HR database. To create the EMPSALARY view from the EMPLOYEES table, Copy the code below and paste it to the textarea of the **SQL** page. Click Go.

```
CREATE VIEW EMPSALARY AS  
SELECT EMP_ID, F_NAME, L_NAME, B_DATE, SEX, SALARY  
FROM EMPLOYEES;
```

The screenshot shows a MySQL query editor interface. In the main text area, there is a code block with three numbered lines:

```
1 CREATE VIEW EMPSALARY AS  
2   SELECT EMP_ID, F_NAME, L_NAME, B_DATE, SEX, SALARY  
3     FROM EMPLOYEES;
```

Below the text area are several buttons: SELECT*, SELECT, INSERT, UPDATE, DELETE, Clear, Format, and Get auto-saved query. There is also a checkbox for Bind parameters and another for Show this query here again. At the bottom, there are settings for Delimiter (with a dropdown menu), Retain query box, Rollback when finished, and Enable foreign key checks. A checked checkbox for Enable foreign key checks is visible.

[Hide query box](#)

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0116 seconds.)

```
CREATE VIEW EMPSALARY AS SELECT EMP_ID, F_NAME, L_NAME, B_DATE, SEX, SALARY FROM EMPLOYEES
```

2. Using SELECT, query the EMPSALARY view to retrieve all the records. Use the following statement.

```
SELECT * FROM EMPSALARY;
```

Showing rows 0 - 9 (10 total, Query took 0.0014 seconds.)

SELECT * FROM EMPSALARY

Proc

Show all

Number of rows:

25

Filter rows:

Search this table

+ Options

	EMP_ID	F_NAME	L_NAME	B_DATE	SEX	SALARY
<input type="checkbox"/>	E1001	John	Thomas	1976-09-01	M	100000.00
<input type="checkbox"/>	E1002	Alice	James	1972-07-31	F	80000.00
<input type="checkbox"/>	E1003	Steve	Wells	1980-10-08	M	50000.00
<input type="checkbox"/>	E1004	Santosh	Kumar	1985-07-20	M	60000.00
<input type="checkbox"/>	E1005	Ahmed	Hussain	1981-04-01	M	70000.00
<input type="checkbox"/>	E1006	Nancy	Allen	1978-06-02	F	90000.00
<input type="checkbox"/>	E1007	Mary	Thomas	1975-05-05	F	65000.00
<input type="checkbox"/>	E1008	Bharath	Gupta	1985-06-05	M	65000.00
<input type="checkbox"/>	E1009	Andrea	Jones	1990-09-07	F	70000.00
<input type="checkbox"/>	E1010	Ann	Jacob	1982-03-30	F	70000.00



Check all

With selected:

Task 2: Update a View

In this exercise, you will update a View to combine two or more tables in meaningful ways.

Assume that the EMPSALARY view we created in Task 1 doesn't contain enough salary information, such as max/min salary and the job title of the employees. For this, we need to get information from other tables in the database. You need all columns from EMPLOYEES table used above, except for SALARY. You also need the columns JOB_TITLE, MIN_SALARY, MAX_SALARY of the JOBS table.

The command to be used is as follows:

```
CREATE OR REPLACE VIEW EMPSALARY AS
SELECT EMP_ID, F_NAME, L_NAME, B_DATE, SEX, JOB_TITLE,
MIN_SALARY, MAX_SALARY
FROM EMPLOYEES, JOBS
WHERE EMPLOYEES.JOB_ID = JOBS.JOB_IDENT;
```

Run SQL query/queries on table HR.EMPLOYEES: 

```
1 CREATE OR REPLACE VIEW EMPSALARY AS
2     SELECT EMP_ID, F_NAME, L_NAME, B_DATE, SEX, JOB_TITLE, MIN_SALARY, MAX_SALARY
3     FROM EMPLOYEES, JOBS
4     WHERE EMPLOYEES.JOB_ID = JOBS.JOB_IDENT;
```

Bind parameters 

[Delimiter] Show this query here again Retain query box Rollback when finished Enable foreign key checks

[Hide query box](#)

 MySQL returned an empty result set (i.e. zero rows). (Query took 0.0461 seconds.)

```
CREATE OR REPLACE VIEW EMPSALARY AS SELECT EMP_ID, F_NAME, L_NAME, B_DATE, SEX, JOB_TITLE, MIN_SALARY, MAX_SALARY F
```

NOTE: The technique used here to combine data from two tables is called implicit inner join. You will learn more about joins later on. For now, just assume you are combining the data of two different tables, EMPLOYEES and JOBS by connecting their respective columns JOB_ID and JOB_IDENT, since both the columns contain common unique data. You can have a look at the database description, shared at the beginning of the lab, to verify this.

2. Using SELECT, query the updated EMPSALARY view to retrieve all the records. Copy the code below and paste it to the textarea of the SQL page. Click Go.

```
SELECT * FROM EMPSALARY;
```

Showing rows 0 - 9 (10 total, Query took 0.0019 seconds.)

SELECT * FROM EMPSALARY

Proc

Show all

Number of rows:

25

Filter rows:

+ Options

	<input type="checkbox"/> Edit	Copy	Delete	EMP_ID	F_NAME	L_NAME	B_DATE	SEX	JOB_TITLE	MIN_SALARY	MAX_SALARY
<input type="checkbox"/>				E1001	John	Thomas	1976-09-01	M	Sr. Architect	60000.00	100000.00
<input type="checkbox"/>				E1002	Alice	James	1972-07-31	F	Sr.Software Developer	60000.00	80000.00
<input type="checkbox"/>				E1003	Steve	Wells	1980-10-08	M	Jr.Software Developer	40000.00	60000.00
<input type="checkbox"/>				E1004	Santosh	Kumar	1985-07-20	M	Jr.Software Developer	40000.00	60000.00
<input type="checkbox"/>				E1005	Ahmed	Hussain	1981-04-01	M	Jr. Architect	50000.00	70000.00
<input type="checkbox"/>				E1006	Nancy	Allen	1978-06-02	F	Lead Architect	70000.00	100000.00
<input type="checkbox"/>				E1007	Mary	Thomas	1975-05-05	F	Jr. Designer	60000.00	70000.00
<input type="checkbox"/>				E1008	Bharath	Gupta	1985-06-05	M	Jr. Designer	60000.00	70000.00
<input type="checkbox"/>				E1009	Andrea	Jones	1990-09-07	F	Sr. Designer	70000.00	90000.00
<input type="checkbox"/>				E1010	Ann	Jacob	1982-03-30	F	Sr. Designer	70000.00	90000.00

Check all With selected: Edit Copy Delete Export

Task 3: Drop a View

In this exercise, you will drop the created View EMPSALARY.

Use the code below.

```
DROP VIEW EMPSALARY;
```

Run SQL query/queries on table HR.EMPLOYEES: 

```
1 DROP VIEW EMPSALARY;
```

Bind parameters 

[Delimiter] Show this query here again Retain query box Rollback when finished Enable foreign key checks

[Hide query box](#)

 MySQL returned an empty result set (i.e. zero rows). (Query took 0.0056 seconds.)

~~DROP VIEW EMPSALARY~~

[Console](#)

Using SELECT, you can verify whether the EMPSALARY view has been deleted or not. Copy the code below and paste it to the textarea of the SQL page. Click Go.

`SELECT * FROM EMPSALARY;`

```
9  
10  
11 SELECT * FROM EMPSALARY;
```

[SELECT *](#) [SELECT](#) [INSERT](#) [UPDATE](#) [DELETE](#) [Clear](#) [Format](#) [Get auto-saved query](#)

Bind parameters [?](#)

[Delimiter :] Show this query here again Retain query box Rollback when finished Enable foreign key checks

[Hide query box](#)

Error

SQL query: [Copy](#) [?](#)

```
SELECT * FROM EMPSALARY LIMIT 0, 25
```

MySQL said: [?](#)

```
#1146 - Table 'HR.EMPSALARY' doesn't exist
```

[Console](#)

Practice Problems

Try to solve the following practice problems based on your learning in this lab.

1. Create a view “EMP_DEPT” which has the following information.
EMP_ID, FNAME, LNAME and DEP_ID from EMPLOYEES table

► Click here for the solution

2. Modify “EMP_DEPT” such that it displays Department names instead of Department IDs. For this, we need to combine information from EMPLOYEES and DEPARTMENTS as follows.

EMP_ID, FNAME, LNAME from EMPLOYEES table and
DEP_NAME from DEPARTMENTS table, combined over the columns DEP_ID and DEPT_ID_DEP.

► Click here for the solution

3. Drop the view “EPM_DEPT”.

► Click here for the solution

Conclusion

Congratulations on completing this lab. You now have hands-on knowledge of how to use Views in SQL.

You have now learned how to:

- Create a new View as per the requirement
- Modify a view to include data from multiple tables in the data set
- Drop a view

Author(s)

[Lakshmi Holla](#)

[Malika Singla](#)

Additional Contributor(s)

[Abhishek Gagneja](#)