+ - #ccc

Step 1 of 7

Hands-on Lab: Using Views in MySQL using phpMyAdmin

Estimated time needed: 20 minutes

In this lab, you will learn how to create tables and load data in the MySQL database service using the phpMyAdmin graphical user interface (GUI) tool.

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 an internal database. You will be working on a sample HR database. This HR database schema consists of 5 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_ID
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_SALARY
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
5	Software	30004	L0004

LOCATIONS

LOCT_ID	DEP_ID_LOC
L0001	2
L0002	5
L0003	7

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

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.

How does the syntax of a CREATE VIEW statement look?

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

How does the syntax of a REPLACE VIEW statement look?

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

How does the syntax of a DROP VIEW statement look?

```
DROP VIEW view_name;
```

Exercise 1: Create a View

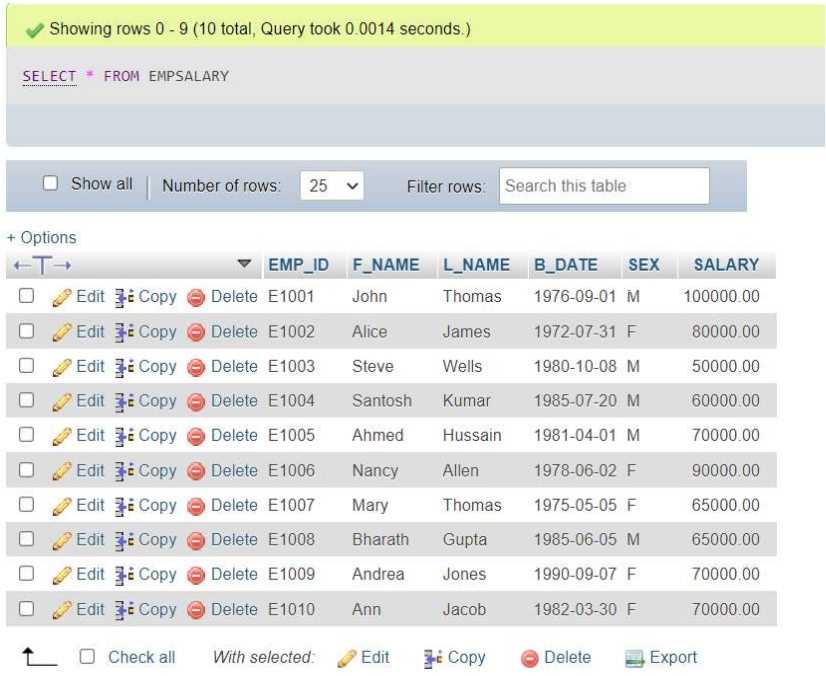
In this exercise, you will create a View and show a selection of data for a given table.

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;
```

2. Using SELECT, query the **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;
```



Exercise 2: Update a View

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

1. It now seems that the **EMPSALARY** view we created in exercise 1 doesn't contain enough salary information, such as max/min salary and the job title of the employees. Let's update the **EMPSALARY** view:

    - combining two tables **EMPLOYEES** and **JOBS** so that we can display our desired information from the HR database.
    - including the columns **JOB*TITLE, MIN*SALARY, MAX_SALARY** of the **JOBS** table as well as excluding the **SALARY** column of the **EMPLOYEES** table.

    Copy the code below and paste it to the textarea of the **SQL** page. Click **Go**..

    ```
    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;
    ```

    > **NOTE:** Don't worry if you don't understand how to combine to two tables using implicit inner join. You will learn more about joins later on. For now, just think 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 diagram (at the beginning of the lab) showing the tables for the HR database to observe how the **JOB*ID*** *and* ***JOB*IDENT** columns from the **EMPLOYEES** and **JOBS** tables respectively contain common unique data.

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;
    ```

image

Exercise 3: Drop a View

In this exercise, you will drop a created View.

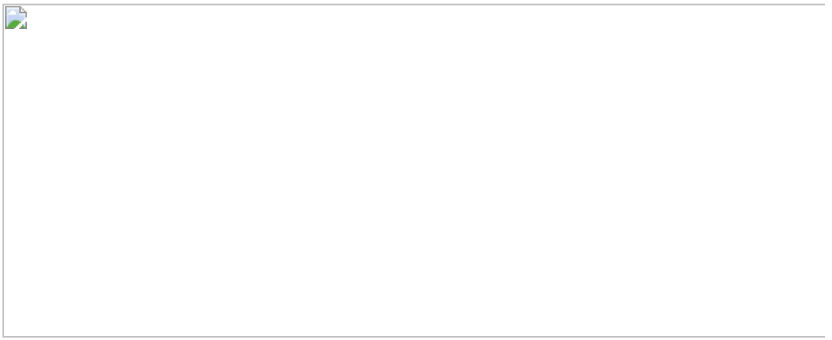
1. Let's delete the created **EMPSALARY** view. Copy the code below and paste it to the paste it to the textarea of the **SQL** page. Click **Go**..

    ```
    DROP VIEW EMPSALARY;
    ```

2. 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;
```



Congratulations! You have completed this lab, and you are ready for the next topic.

Author(s)

Lakshmi Holla

Malika Singla

Changelog

Date	Version	Changed by	Change Description
2021-11-01	0.1	Lakshmi Holla, Malika Singla	Initial Version

Continue