Hands-on Lab: Using Views in MySQL using phpMyAdmin

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:

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;
```

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

```
SELECT * FROM EMPSALARY;
```

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;
```

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;
```

Task 3: Drop a View

In this exercise, you will drop the created View `EMPSALARY`.
Use the code below.

```
DROP VIEW EMPSALARY;
```

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;
```

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