Database Design

Introduction

In this lab you will execute some SQL operations, practice using Joins and use Explain.

Estimated Lab Time: 20 minutes

Objectives

In this lab, you will: -Work with SQL CRUD -Work with SQL Index -Work with SQL JOINS -Work with SQL EXPLAINS

> **Note:** -Server: mysql1

Task 1: Work with SQL CRUD

1. If not already connected to app-srv and mysql1 then do the following

    - a. Connect with your SSH client using the public IP and the provided ssh Example of connections from Linux, MAC, Windows Powershell

      ```
      <span style="color:green">shell></span> <copy> ssh -i id_rsa_app-srv
      opc@<public_ip></copy>
      ```

    - b. Connect to shell-mysql1

      ```
      <span style="color:green">shell-app-srv$</span> <copy> ssh -i
      $HOME/sshkeys/id_rsa_mysql1 opc@mysql1 </copy>
      ```

2. Connect to your mysql-advanced with admin user

   ```
   <span style="color:green">shell></span> <copy>mysqlsh
   admin@mysql1:3307</copy>
   ```

3. Create a new table poi

   ```
   <span style="color:blue">mysql></span> <copy>use world;</copy>
   ```

   ```
   <span style="color:blue">mysql></span> <copy>SHOW TABLES;</copy>
   ```

```
<span style="color:blue">mysql></span> <copy>CREATE TABLE if not exists poi
(x Int, y INT, z INT);</copy>
```

4. We forgot to add a primary key! Let's add a new column for id used for large integer values

```
<span style="color:blue">mysql></span> <copy>alter table poi add id bigint;
</copy>
```

```
<span style="color:blue">mysql></span> <copy>ALTER TABLE poi ADD PRIMARY KEY
(id);</copy>
```

5. Create a copy of your city table

```
<span style="color:blue">mysql></span> <copy>create table city_part as
select * from city;</copy>
```

6. How many records does it contain?

```
<span style="color:blue">mysql></span> <copy>SELECT count(*) FROM city_part;
</copy>
```

7. How many records city table contain?

```
<span style="color:blue">mysql></span> <copy>SELECT count(*) FROM city;
</copy>
```

8. Verify the difference of the two table creation (there is a big one!)

```
<span style="color:blue">mysql></span> <copy>show create table city\G</copy>
```

```
<span style="color:blue">mysql></span> <copy>show create table
city_part\G</copy>
```

Task 2: Work with SQL INDEX

1. Create address table with an index

```
<span style="color:blue">mysql></span> <copy>CREATE TABLE address (
id INT unsigned NOT NULL PRIMARY KEY,
email VARCHAR(100),
lname CHAR(30),
fname CHAR(30),
INDEX name (lname, fname)
);</copy>
```

2. Load data into address table

```
<span style="color:blue">mysql></span> <copy>INSERT INTO address (id, email,
lname, fname) VALUES
```

(1, 'john@example.com', 'Smith', 'John'), (2, 'jane@example.com', 'Doe', 'Jane'), (3, 'bob@example.com', 'Johnson', 'Bob'), (4, 'alice@example.com', 'Brown', 'Alice'), (5, 'mike@example.com', 'Williams', 'Mike'); ```

2. List address table

```
<span style="color:blue">mysql></span> <copy>select * from address;</copy>
```

```
<span style="color:blue">mysql></span> <copy>select * from address where
lname='smith';</copy>
```

Task 3: Work with JOINS

1. Join (Inner Join): An inner join returns rows where there is a match between the joined tables based on the specified condition.

   - List cities along with the countries they belong to

   ```
   <span style="color:blue">mysql></span> <copy>SELECT city.Name AS City,
   country.Name AS Country FROM city INNER JOIN country ON city.CountryCode =
   country.Code;</copy>
   ```

2. Left Join (Outer Join): A left join returns all rows from the left table and the matched rows from the right table. If there's no match, NULL values are included for the right table's columns.

   - List all countries along with their cities.

   ```
   <span style="color:blue">mysql></span> <copy>SELECT country.Name AS Country,
   city.Name AS City  FROM country LEFT JOIN city ON country.Code =
   ```

```
    city.CountryCode order by city.Name;</copy>
```

3. Left Semijoin (Outer Join): A left join returns all rows from the left table and the matched rows from the right table. If there's no match, NULL values are included for the right table's columns.

   - List all countries along with their official languages.

```
<span style="color:blue">mysql></span> <copy>SELECT country.Name AS Country,
countrylanguage.Language AS Language FROM country LEFT JOIN countrylanguage
ON country.Code = countrylanguage.CountryCode AND countrylanguage.IsOfficial
= 'T';</copy>
```

4. Right Join (Outer Join): A right join is similar to a left join but returns all rows from the right table and the matched rows from the left table.

   - List all official languages along with the countries that speak them.

```
<span style="color:blue">mysql></span> <copy>SELECT countrylanguage.Language
AS Language, country.Name AS Country FROM countrylanguage RIGHT JOIN country
ON countrylanguage.CountryCode = country.Code AND countrylanguage.IsOfficial
= 'T';</copy>
```

Task 4: Work with EXPLAINS

1. Use the EXPLAIN command to show how MySQL will process this index statement

```
<span style="color:blue">mysql></span> <copy>explain select fname from
address where lname='smith';</copy>
```

   - **Interpretation:** The optimizer plans to use the "name" index to match rows in the "address" table based on a constant value. The index is used to efficiently locate the relevant rows, and a WHERE clause is applied for further filtering. The optimizer estimates that only one row will be examined, and all rows are expected to be included in the result set.

2. Use the EXPLAIN command to show how MySQL will process this Join/where statement

```
<span style="color:blue">mysql></span> <copy>EXPLAIN SELECT
countrylanguage.Language AS Language, country.Name AS Country FROM
countrylanguage RIGHT JOIN country ON countrylanguage.CountryCode =
country.Code AND countrylanguage.IsOfficial = 'T';</copy>
```

   - **Interpretation:** The optimizer plans to perform a full table scan for the "country" table and use a non-unique index for the "countrylanguage" table, matching rows based on the "Code" column

from the "country" table. The output provides valuable information for understanding and optimizing query performance.

3. You can now exit from MySQL Shell

```
<span style="color:blue">mysql</span> <copy>\q</copy>
```

Acknowledgements

- **Author** - Marco Carlessi, Principal Sales Consultant
- **Contributors** - Perside Foster, Principal Sales Consultant, Selena Sánchez, MySQL Solutions Engineer
- **Last Updated By/Date** - Perside Foster, Partner Solutions Engineer, November 2024