Week 08 Research Assignment

Note: All answers are a synthesis of what I've learned from the class materials, unless a source is linked specifically.

Prompt:

Besides the creation and dropping of tables and columns (Data Definition Requests or DDL), what other types of requests can you make to a MySQL database? How is data retrieved from a MySQL database? What clauses exist for restricting the data that is returned to the user? Please include the syntax for this type of request, and examples.

Response:

In addition to the **Data Definition Language *(DDL)*** subset of SQL, there are three other subsets: **Data Manipulation Language *(DML)***, **Data Query Language *(DQL)***, and **Data Control Language *(DCL)***.

- **DML** deals with requests that modify the actual data, including adding, updating, and deleting it. This *is not* the same as DDL requests, which are used to define schemas and tables; DML only conerns the *content* of schema and tables, not their definition. Some common SQL constructs that fall into the DML subset are `DELETE FROM`, `INSERT INTO ... VALUES`, and `UPDATE ... SET`. Here are some examples:

  ```
  DELETE FROM table_name WHERE condition;

  INSERT INTO table_name (column_name_1, column_name_2,
  ...column_name_n) VALUES (column_value_1, column_value_2,
  ...column_value_n);

  /*Note that the following example would almost always be
  followed by a ...WHERE condition... construct to specify
  which rows are updated, unless one wants to set that
  column to a single value for all existing rows.*/

  UPDATE table_name SET column_name = new_value;
  ```

- **DQL** deals solely with *retrieving* data from a database. The `SELECT...` construct is the best example of this language subset. The wikipedia article on **DQL** (linked at the end of this response) seems to indicate that combining `SELECT` with `FROM` and/or `WHERE` moves it into the realm of **DML**, but the article on **DML** appears to limit this to cases where `SELECT` is followed by `INTO` (because this results in insertion). My intuition leans toward `SELECT ... FROM ... WHERE` constructions being part of the **DQL** subset, as such constructs don't appear to modify any data. Here's an example:

```
SELECT column_name FROM table_name WHERE condition;

/*or, to select all fields from an entry that matches
the condition...*/

SELECT * FROM table_name WHERE condition
```

- **DCL** is the subset of SQL dealing with controlling user access, usually through explicitly identifying which privileges a user can access, and from which host. It includes constructs such as `GRANT` and `REVOKE`, as well as `CREATE USER` and `CREATE ROLE`. While SQL is largely treated the same by different implementations (like Microsoft SQL, MySQL, Oracle, etc...), it can especially differ in its **DCL** subset. The following example are quite specific to MySQL.

```
--Create roles. These act as "groups" of privileges
CREATE ROLE 'role_admin', 'role_view';

/*Create some users. NOTE: The same username followed
by a different host would be creating different users
than the ones below.
Additionally, the example passwords suck as passwords*/
CREATE USER 'user_viewer'@'host_name' IDENTIFIED BY 'view';
CREATE USER 'user_admin'@'host_name' IDENTIFIED BY 'admin';


/*Grant privileges to the different roles for
all tables in a schema*/
GRANT SELECT ON schema_name.* to 'role_view';
GRANT ALL ON schema_name.* to 'role_admin';

--Grant different roles to different users
GRANT 'role_view' TO 'user_viewer'@'host_name';
GRANT 'role_admin' TO 'user_admin'@'host_name';

--Oops, we hired an evil hacker admin.
--Remove their admin role
REVOKE 'role_admin' FROM 'user_admin'@'host_name';

--and just in case
ALTER USER 'user_admin'@'host_name' ACCOUNT LOCK;

--Now get your security guy and your boss
--on the phone, stat!
```

In the above response, information is paraphrased, condensed, and/or synthesized from the following wikipedia articles, and furthermore informed by my own *practice with MySQL:*

- https://en.wikipedia.org/wiki/Data_manipulation_language
- https://en.wikipedia.org/wiki/Select_(SQL)
- https://en.wikipedia.org/wiki/Delete_(SQL)
- https://en.wikipedia.org/wiki/Data_query_language
- https://en.wikipedia.org/wiki/Data_control_language

In addition, the following source was used to flesh out my understanding of DCL:

- https://dev.mysql.com/doc/refman/8.0/en/roles.html

Prompt:

What's your favorite thing you learned this week?

Response:

My favorite thing I learned this week was the `JOIN` and `GROUP BY` constructs. They make it easy to retrieve data from multiple tables in informative ways. I had a lot of fun playing with the Northwind example database on W3Schools SQL editor (https://www.w3schools.com/sql/trysql.asp?filename=trysql_editor) and am particularly proud of this snippet I came up with:

```sql
SELECT Employees.FirstName, Employees.LastName, Employees.EmployeeID,
SUM(OrderDetails.Quantity * Products.Price) FROM OrderDetails
JOIN Orders ON OrderDetails.OrderID = Orders.OrderID
JOIN Products ON OrderDetails.ProductID = Products.ProductID
JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID
GROUP BY Employees.EmployeeID;

/*returns the total dollar amount in sales each employee made
across all their orders. From there you could probably get
their average dollar amount per order, etc...*/
```