Metropolitan State University

ICS 311 —Database Management Systems

Term Project – Step 3

Database Implementation on MySQL

Due: See Syllabus

Part 1:

In this part you are required to implement your database on MySQL. Your implementation should include the following:

1- Create all the tables that you specified in your relational schema (Part 2, of Step 2). Make sure to include primary keys and foreign keys.

2- Populate all tables with data of your choice. Make sure that each table includes at least 10 rows.

.SQL file was submitted for this part

Part 2:

Design and clearly explain **at least ten** query scenarios that may be useful on your database.

Write SQL syntax to answer each query.

A query like this: select count(*) from tableOne, can only be used as one of the ten.
Strongly suggested to include some queries using Table Joins, Aggregates, Group By, Order By, and Having.
At least one query must use a view (that you created from your application tables) in the FROM clause.
Note: the SELECT used for in the creation of the view, does not count as one of the 10 queries.

Query 1

USE OF COUNT(*)

```
SELECT COUNT(*) FROM Movie;
```

Query-2

USE OF VIEW-

View-A View is a temporary table and it is same as the main table but we can perform any operation on it without modifying main table.

```
CREATE VIEW Count_Movie AS(SELECT MovieID,Genre FROM Movie);
```

Now Display data of view-

```
SELECT * FROM Count_Movie;
```

Query-4

USE OF AGGREGATE FUNCTION-

```
SELECT SUM(Show.ShowID) AS SUM FROM Movie

INNER JOIN Show ON Show.MovieID=Movie.MovieID;
```

Query-5

USE OF GROUP BY-

```
SELECT Show.ShowID FROM Show

GROUP BY Show.MovieID,Show.ShowID;
```

Query-6

USE OF ORDER BY

```
SELECT Show.ShowID FROM Show

ORDER BY Show.ShowID DESC;
```

Query-7

USE OF HAVING

```
SELECT Show.ShowID FROM Show

GROUP BY Show.ShowID

HAVING Show.ShowID>1;
```

Query-9

USE of GROUP BY AND ORDER BY TOGETHER-

```
SELECT Movie.MovieID FROM Movie

GROUP BY Movie.MovieID

ORDER BY Movie.MovieID DESC;
```

Query-10

Use of Above all functions-

```
SELECT Movie.Genre,Show.ShowID FROM Movie

INNER JOIN Show ON Show.MovieID=Movie.MovieID

WHERE Movie.MovieID>1

GROUP BY Movie.Genre,Show.ShowID

HAVING COUNT(Show.ShowID)>2
```

```
ORDER BY Show.ShowID DESC;
```

ELECT is used to display all(*) or specified column/columns

COUNT( ) is used to count number of rows inside specified column.

table_name1.column_name1,Here it specifies that column_name1 is of table table_name1

INNER JOIN is used to join two tables based on a common column.

WHERE is used to specify condition.

We cannot use aggregate function directly inside where so HAVING is used.

GROUP BY is used to group result as per the column specified.

ORDER BY is used to sort result either in ascending(ASC)increasing or descending(DESC)decreasing order.

What to submit:

1- A word document to explain your query scenarios (in good detail) along with SQL syntax to answer these queries. Make sure to include at least two aggregate queries, also some doing table joins, some order by's and group by's (in other words have a good variety that incorporates the material we have been learning).

2- There is a dropbox folder on D2L to turn in your project code. You need to turn in the '.sql' scripts that include:
a. SQL statements to create the tables (from Part 1 above).
b. SQL statements to populate tables with data (from Part 1 above).
c. SQL statements for your query scenarios (from Part 2).
d. SQL used to create the view (from Part 2).

Note that you need to include all the required SQL statements in file. This could be a Word Doc. or a text file that ends in ".sql". **Make sure to test your file before you submit it.**