MIS602 Data Modeling & Database Design

Name

Institutional Affiliation

Contents

[Introduction 3](#_Toc55060049)

[Reflection 3](#_Toc55060050)

[References 6](#_Toc55060051)

# Introduction

The task involved completing the various tasks as outlined in the assignment. This required the use of SQL statements and commands. It was accomplished on MySQL database management system and also using PHP MyAdmin. The SQL statements and commands were written in an SQL file. In order to ensure that the SQL file created worked as expected, the file was then imported into the MySQL using PHP MyAdmin. All the commands worked as expected.

# Reflection

The first task required creating a database schema and the various tables as per the assignment. The first attempt failed due to a schema not being selected. This was corrected by selecting the Schema using MySQL USE command (Bell, 2018). Once the tables had been created, the correct relationships were setup and referential integrity enforced by altering the tables. The second and third tasks involved populating the tables with data, which was easily achieved by using the SQL INSERT command.

The fourth, fifth and seventh tasks involved selecting data from the tables. This was achieved by combining both the asterisk operator to select all columns in a table as well as column names to select specific columns from the table (DuBois, 2003). The fifth task was an update query that did not affect any row since none of the rows in the building table met the criteria used to select the table to update. The eighth task involved selecting using a WHERE clause which was achieved with any problems.

For the ninth task, it was required that the correct apartments’ rent be updated. Various approaches were tried in order to determine the amount to increase the rent by. Ultimately, in order to increase the rent by 2%, the current rent was multiplied with 1.02. To determine which apartments to update, a sub query was used in order to select the building id of the correct building given its name.

Tasks 10 to 13 involved the use of SELECT queries to get the desired results. In task 13, the ORDER BY clause was used so as to arrange the results in descending order. For task 14, the total number of apartments was determined using the SUM function (Kofler, 2004). Additionally, the resulting total was renamed using an alias to provide an easier to understand name for the results of the query.

For task 15, the WHERE clause compared the values in Apartment and Owner tables so as to combine their values as desired. Without using table aliases, the statement failed since the column Owner\_ID was in both tables and the DBMS could not determine which table’s column was being referred to (McLaughlin, 2013). This was solved by using table aliases although using the fully qualified name of a column i.e. Table.Column was also found to work. Without the GROUP BY clause, all the owners ended up having the same number of apartments which was the total of all the apartments.

For Task 16, a new owner was inserted in the Owner table. To delete a row containing a given value, the LIKE clause was used in combining with the % operator. The % operator ensured that any row that contained the enclosed value would be deleted. Task 17 also required the use of an alias when selecting the desired columns since both tables had the same desired column Building\_ID, although it was only necessary to alias one table. Task 18 and 19 however had both tables involved in the query aliased although since the where claused required the column with the same name to be used for selecting the desired rows to display.

For task 20, the SUM function was used in conjunction with the GROUP BY clause in order to determine the number of apartments in a building. The total number of apartments per building was then aliased so as to make the name more meaningful and to use it in determining which building had more than 5 apartments. Trying use the alias in the WHERE clause resulted in the query failing since the WHERE clause could not detect the alias created. This necessitated the use of the HAVING clause in order to correctly reference the newly created alias name.

For task 21, the SUM function was used in conjunction with the GROUP BY clause in order to determine the number of apartments owned by each owner. The total number of apartments per owner was then aliased so as to make the name more meaningful and to use it in determining which how many apartments each owner had. Trying use the alias in the WHERE clause resulted in the query failing since the WHERE clause could not detect the alias created. This necessitated the use of the HAVING clause in order to correctly reference the newly created alias name (Vaswani, 2003). Furthermore, without having a default value of 0 in the apartment that could be owned by all owners, it was not possible to determine who had no apartments since there was no corresponding value in the apartments table for the query to refer to.

For task 22, a sub query was used to select the building ID. The query was ordered by the apartment rent column value and limited to return only one value using the LIMIT clause, ensuring only the building with the lowest rent had its ID returned. Task 23 and 24 required the use of table aliases due to both tables having a column with the same name. For task number 24, the asterisk wildcard was used in selecting the columns rather than having to list out the columns in all the three tables explicitly.

# References

Bell, C. (2018). *Introducing the MySQL 8 Document Store*.

DuBois, P. (2003). *MySQL Cookbookbook*.

Kofler, M. (2004). *The Definitive Guide to MySQL*.

McLaughlin, M. (2013). *MySQL Workbench: Data Modeling & Development*.

Vaswamni, V. (2003). *MySQL: The Complete Reference*.