SIT103/SIT772: Database Fundamentals

7.2C: Mini-project Part 2

Overview

In this task, you will implement the database designed in Task 4.2C Mini-project Part 1. You will write SQL statements using MySQL to create tables, insert records and interact with the database to manipulate data. It is assumed that you have completed Task 4.2C and you have designed the normalized database for a company of your choice.

Tasks to do

1. Write SQL statements to create all tables in the final normalized ERD in Task 4.2C.

   - Use appropriate data types and required constraints for attributes.
   - Ensure all relationships and relationship constraints are maintained.

   Note that the order of table creation can be important when tables have Foreign Keys.
   Provide **SQL statements to create all tables** in your report.

2. Insert at least FIVE records in each table using SQL.

   Provide **all SQL statements** used to insert data and **screenshots of results of 'SELECT * FROM <tablename>;'** for each table in your report.

3. Write and run an SQL statement to add one additional column (attribute) in any one of the existing tables with a default value.

   First, provide the following information:

   - Name of table you want to add a new column
   - Name, data type and default value of the new column you want to add in the table

   Then write an SQL statement to do it. Your SQL statement must do what you said you want to do in the first part of this question. When you run the SQL, it populates the default value in the newly added column for all existing records in the table.

   Provide your **SQL statement** and a **screenshot of records in the table using** '**SELECT * FROM <tablename>**' to show that existing records have default values for the newly added column in your report.

4. Write and run an SQL statement to update the default value of the newly added column to a different value for certain rows based on a condition using any other column.

   NOTE: You are NOT changing the values of the new column for all records; you are ONLY changing for the records that match your condition.

   First, provide the following information:

   - What is the condition you want to use to filter the rows? You can use any comparison operator.

   - What is the new value of the newly added column you want to set for those selected rows?

   Then, write an SQL statement to do it. Your SQL command must do what you said you want to do in the first part of this question. When you run the SQL, it changes the value of the newly added column for the records matching the condition from the default value to the new value.

   Provide your **SQL statement** and a **screenshot of records in the table using** '`SELECT * FOM <tablename>;`' to show that the value of newly added column is successfully updated for records matching the condition in your report.

5. In this section, you are required to write SQL queries to interact with the database you implemented. Answer each SQL question in this section with the following:

   - First you provide what you want to do

   - Provide an **SQL statement to do what you want to do** and **provide a screenshot of results after successful execution** of the SQL command

   a) Write an SQL query to demonstrate the use of SELECT with INNER JOIN and ORDER BY.

   b) Write an SQL query to demonstrate the use of SELECT with WHERE and IN.

   c) Write an SQL query to demonstrate the use of at least one DATE function.

   d) Write an SQL statement to create a VIEW using a SELECT statement with a JOIN. Provide the statement to create the VIEW you want and demonstrate the output of the VIEW using '`SELECT * FROM <viewname>;`'.

Submission Requirements:

Submit one PDF/WORD file as a database design document for the company answering all tasks/questions above.

Submission Due

The due date for each task is stated on its OnTrack task information dashboard.