Course: SQL

Final Project Problem Statement !

From the course you learnt various fundamental concepts and skills related to working with various databases (PostgreSQL / Redshift/DynamoDB).such as

1. Creating databases and tables
2. Inserting and manipulating data,
3. Querying and filtering data, Aggregating and analyzing data.
4. Data manipulation and transformation.

By completing these tasks and understanding the concepts, you will develop a solid foundation in working with databases using PostgreSQL/RedShift. These skills are essential for various roles in data management, software development, and data analysis.

Instructions to upload your project document.

Follow the instructions below to upload your code and output

1. Create an account in Github (https://github.com/)

2. Create a new repository name" Project-your name"(Example Project-Sam)

3. The repository should be made "Public" and "Add a README file" should be added while creating.

4. Add all the files to be submitted in that folder created

5.Commit and push them from VS Code software or online compiler or any method and make sure all the files get uploaded into the Github Repository that you created.

6. The folder to be uploaded should have all the following content. Names of the files with code and All the screenshots of the outputs.

7. A README File describing the project that the candidate created .

8. Once everything is successfully uploaded, make sure to submit the repository link in the **"My Assignment"** menu as a PDF document and click the **"Submit Assignment"** button. (Copy and paste the Github link in your document).

Task1

Project Title: Academic Management System (using SQL)

Project Description:

Design and develop an Academic Management System using SQL. The projects should involve three tables 1.StudentInfo 2. CoursesInfo 3.EnrollmentInfo. The Aim is to create a system that allows for managing student information and course enrollment. The project will include the following tasks:

1. **Database Creation:**

a) Create the StudentInfo table with columns STU_ ID, STU_NAME, DOB, PHONE_NO, EMAIL_ID,ADDRESS.

b) Create the CoursesInfo table with columns COURSE_ID, COURSE_NAME,COURSE_INSTRUCTOR NAME.

c) Create the EnrollmentInfo with columns ENROLLMENT_ID, STU_ ID, COURSE_ID,

ENROLL_STATUS(Enrolled/Not Enrolled). The FOREIGN KEY constraint in the EnrollmentInfo table references the STU_ID column in the StudentInfo table and the COURSE_ID column in the CoursesInfo table.

2. **Data Creation:**

Insert some sample data for StudentInfo table , CoursesInfo table, EnrollmentInfo with respective fields.

3) Retrieve the Student Information

a) Write a query to retrieve student details, such as student name, contact informations, and Enrollment status.

b) Write a query to retrieve a list of courses in which a specific student is enrolled.

c) Write a query to retrieve course information, including course name, instructor information.

d) Write a query to retrieve course information for a specific course .

e) Write a query to retrieve course information for multiple courses.

f) Test the queries to ensure accurate retrieval of student information.(execute the queries and verify the results against the expected output.)

4. Reporting and Analytics (Using joining queries)

a) Write a query to retrieve the number of students enrolled in each course

b) Write a query to retrieve the list of students enrolled in a specific course

c) Write a query to retrieve the count of enrolled students for each instructor.

d) Write a query to retrieve the list of students who are enrolled in multiple courses

e) Write a query to retrieve the courses that have the highest number of enrolled students(arranging from highest to lowest)

Task2

Project: Student Database Management System(PostgreSQL)

Objective: Design and implement a student database management system using PostgreSQL that allows storing and retrieving student information efficiently. The project will include the following tasks:

1. Database Setup

Create a database named "student_database."

Create a table called " student_table " with the following columns: Student_id (integer), Stu_name (text), Department (text), email_id (text),Phone_no (numeric), Address (text), Date_of_birth (date), Gender (text), Major (text), GPA (numeric),Grade (text) should be A,B,C etc.

2. **Data Entry**

Insert 10 sample records into the "student_table" using INSERT command.

3. **Student Information Retrieval**

Develop a query to retrieve all students' information from the "student_table" and sort them in descending order by their grade.

4. **Query for Male Students:**

.Implement a query to retrieve information about all male students from the "student_table."

5. **Query for Students with GPA less than 5.0**

Create a query to fetch the details of students who have a GPA less than 5.0 from the "student_table."

6. **Update Student Email and Grade**

Write an update statement to modify the email and grade of a student with a specific ID in the "student_table."

7. **Query for Students with Grade "B"**

Develop a query to retrieve the names and ages of all students who have a grade of "B" from the "student_table."

8. **Grouping and Calculation**

Create a query to group the "student_table" by the "Department" and "Gender" columns and calculate the average GPA for each combination.

9. **Table Renaming**

Rename the "student_table" to "student_info" using the appropriate SQL statement.

10. **Retrieve Student with Highest GPA**

Write a query to retrieve the name of the student with the highest GPA from the "student_info" table.

Task 3

Project: Event Management System using PostgreSQL.

Objective: To develop the application that allows users to create and manage events, track attendees, and handle event registrations efficiently. The project will include the following tasks:

1. Database Creation

Create a database named "EventsManagement."

Create tables for Events, Attendees, and Registrations.

Events- Event_Id, Event_Name, Event_Date, Event_Location, Event_Description

Attendees- Attendee_Id, Attendee_Name, Attendee_Phone, Attendee_Email, Attendee_City

Registrations-Registration_id, Event_Id, Attendee_Id,Registration_Date,Registration_Amount.

The FOREIGN KEY constraint in the Registrations table references the Event_Id column in the Events table and the Attendee_Id column in the Attendees table.

2. Data Creation

Insert some sample data for Events, Attendees, and Registrations tables with respective fields.

3. Manage Event Details

a) Inserting a new event.

b) Updating an event's information.

c) Deleting an event.

4) Manage Track Attendees & Handle Events

a)Inserting a new attendee.

b)Registering an attendee for an event.

5.Develop queries to retrieve event information, generate attendee lists, and calculate event attendance statistics.

Task4

Develop the queries to retrieve information from the OLAP operations performed and to gain a deeper understanding of the sales data through different dimensions, aggregations, and filters.

Project: OLAP Operations (using Redshift or PostgreSQL)

Objective: Perform OLAP operations (Drill Down, Rollup, Cube, Slice, and Dice) on the "sales_sample" table to analyze sales data. The project will include the following tasks:

1. **Database Creation**

Create a database to store the sales data (Redshift or PostgreSQL).

Create a table named "sales_sample" with the specified columns:

Product_Id (Integer)

Region (varchar(50))-like East ,West etc

Date (Date)

Sales_Amount (int/numeric)

2. **Data Creation**

Insert 10 sample records into the "sales_sample" table, representing sales data.

3. **Perform OLAP operations**

a) Drill Down-Analyze sales data at a more detailed level. Write a query to perform drill down from region to product level to understand sales performance.

b) Rollup- To summarize sales data at different levels of granularity. Write a query to perform roll up from product to region level to view total sales by region.

c) Cube - To analyze sales data from multiple dimensions simultaneously. Write a query to Explore sales data from different perspectives, such as product, region, and date.

d) Slice- To extract a subset of data based on specific criteria. Write a query to slice the data to view sales for a particular region or date range.

e) Dice - To extract data based on multiple criteria. Write a query to view sales for specific combinations of product, region, and date