COURSE CODE: 1372

PROGRAM CODE- PROG8651

SECTION – 10

ASSIGNMENT TITLE: PROJECT-3 SQL Administration (A)

GROUP MEMBERS

Saraswat, Gaurav

Patel, Shivamkumar Prakashbhai

Parmar, Kaushalkumar Jagdishbhai

Kaur, Jaspreet

Deepika, Deepika

Bathul Khanam, Bathul Khanam

Section B: Expanding database functionality.

Question 3: -

From: Database Administrator

To: Matt Kozi

Subject: Proposal for simplified Data Cleansing Process using Stored Procedures

I would like to present a solution to improve our data cleansing processes, particularly focusing on the Salary field in the Employee table. My primary aim is to use stored procedures to automate formatting operations, which will help me save time on repetitive tasks.

To make it easier to create consistent information formats for the Salary field in the Employee table, I'm suggesting a method of implementation for stored procedures that achieves the following:

- **Convert the INT to a CHAR**: The salary field will be converted from INT to CHAR data type using SQL commands in the suggested stored procedure.

- **Format Salary with Commas, Decimal Places, and Dollar Signs**: The stored procedure will prefix the salary attribute with a dollar sign, add commas as separators for thousands, and include decimal places for cents if necessary.

By implementing this stored method into practice, we can streamline the data cleansing step for the Salary field and minimize the amount of manual labour needed for formatting each time.

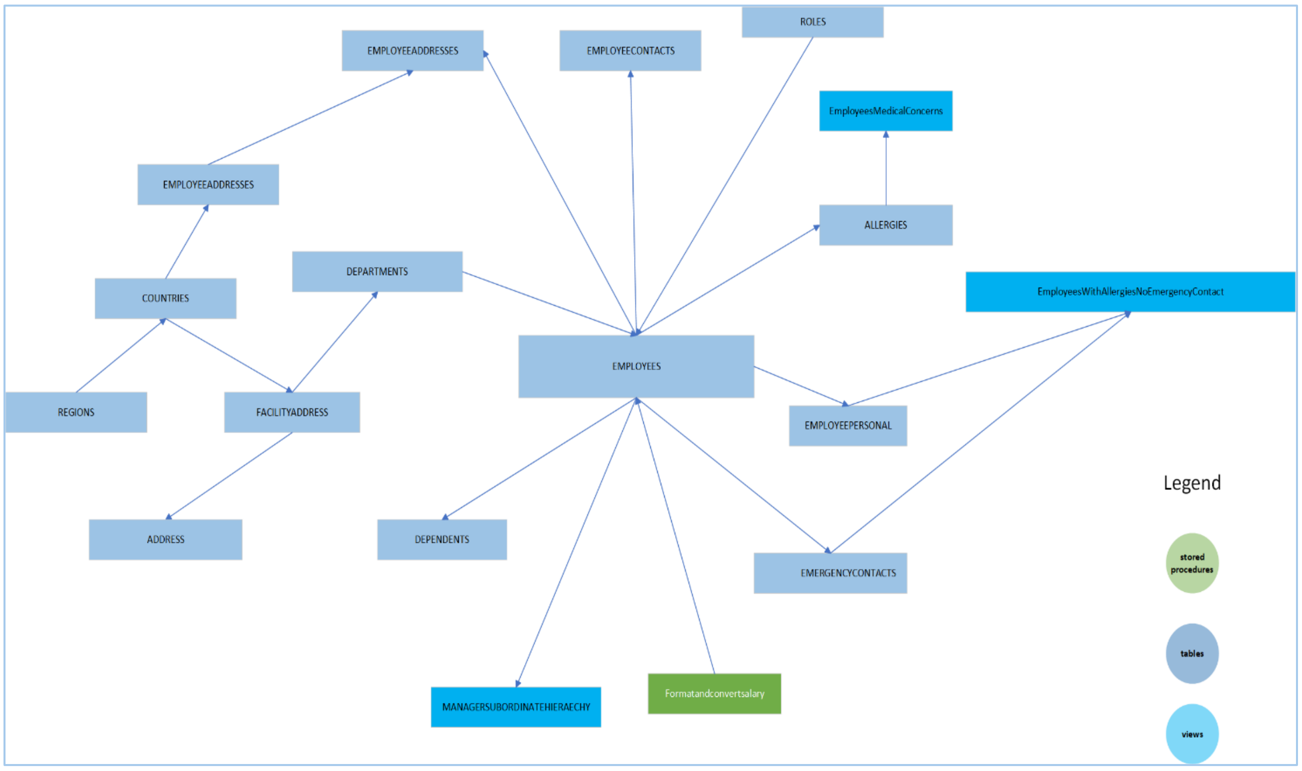
I would like to talk more about this concept and answer any queries or worries you might have.

I'm eager to move on with implementing the Stored procedure with your approval.

Thank you.

SECTION: C

1) Please find dependency graph below:



2) Explain why you need to develop dependency graphs which include views and database objects such as stored procedures when we already have logical and physical data models. How do dependency graphs and data models Differentiate?

- **Why do we need to develop dependency graphs alongside logical and physical data models?**

  - In database redesign, dependency graphs are essential. Dependency graphs give a dynamic perspective of the database structure, whereas logical and physical data models provide a complete image of the entities, attributes, relationships, and constraints. They give an example of the interdependencies between various database objects, such as tables, views, and stored procedures.

    The logical and physical data models establish the relationships between data items, which depict the data and how it is kept in the database. Dependency graphs are helpful because these models must clearly show the relationships between various database elements.

- **How are dependency graphs different from data models?**

  - The database's schema and structure are the main topics of data models. On the other hand, dependency graphs show the relationships between table, view, and stored procedure objects in a database.
  - Data models help comprehend the original design of the database. On the other hand, dependency graphs—particularly for views and stored procedures—help determine the consequences of database modifications.
  - Entities, attributes, and relationships are displayed in data models. Dependency graphs, however, take things a step further. They show how views and stored procedures interact and function while the database is in use.
  - Data models establish entity relationships. Dependency graphs become more intricate, illustrating the connections between views, tables, and stored procedures. This gives a clearer picture of how various database components are interdependent.

3) Through your recent work, you implemented a new stored procedure, while stored procedures can be helpful, they can also present challenges to data stability. Assume the following scenario is true; A stored procedure that exports health data from the EmployeePersonal table was running, and another stored procedure that reformats emergency contact phone data could also be running.

a. Assume we just executed the stored procedure you have created for this assessment while the other two stored procedures were also running, and an error occurred. Give an example of a dirty read, a nonrepeatable read, and a phantom read among this group of stored procedures.

- If the newly created "FormatAndConvertSalary" stored procedure accesses data from the EmployeePersonal database that has been modified but not yet committed by the other stored procedures that are currently in use, this could result in a dirty read.

- o If our newly created stored procedure "FormatAndConvertSalary" reads the same information from the EmployeePersonal database more than once and each time it receives a different value because of modifications made by the other stored procedures that are currently in use, this could create a nonrepeatable read.

- A phantom read can happen when our newly created stored procedure, "FormatAndConvertSalary," reads a set of rows from the EmployeePersonal table and then, while it's being executed, other stored procedures insert new rows. As a result, when the "FormatAndConvertSalary" stored procedure reads the same set of rows, it sees extra rows.

b. What concurrency control measures are appropriate for the stored procedure that you are creating?

- It is advised that concurrency control be provided for our newly stored process called "FormatAndConvertSalary." The stored procedure can stop other concurrent processes from changing the data it is working on until it has finished executing by obtaining locks on the pertinent tables or rows in the EmployeePersonal table. In addition to ensuring data stability, this will stop phantom, unclean, and nonrepeatable reads.

c. What concurrency control measures are appropriate for the two other stored procedures?

- To ensure data consistency and avoid conflicts with concurrent processes, the other two stored procedures should incorporate appropriate concurrency control techniques, including locking the tables or rows they deal with. This method helps prevent problems such as phantom reads, dirty reads, and nonrepeatable reads. Furthermore, to prevent deadlocks—a situation in which several processes are left waiting endlessly for resources locked by one another—these stored procedures should lock resources in a consistent order.