CONESTOGA
Connect Life and Learning

COURSE CODE: 1372

PROGRAM CODE- PROG8651

SECTION – 10

ASSIGMENT TITTLE: PROJECT-3 SQL Administration (A)

STUDENT NAME(s):

- ✓ PRIYANSH PATEL (8894179)
- ✓ SAURABH KANOJIA (8913722)
- ✓ VAISHNAVI GOPAL WADHWA (8939083)

DATE: 24 Nov 2023

SECTION – B : Expanding database functionality

You will submit your SQL answers for questions 1,2, and 4 for this section as a .SQL file. Your response for Question 3 will need to be included with the .PDF submission for Section C.

3. Part of your responsibilities as a DBA has you routinely cleansing data to ensure proper data formats are being enforced across all tables. This activity, while important, consumes a significant part of your time that could be better spent working on more value-add projects. You want to partially automate the process through the development of stored procedures but would need approval from your boss. Write a brief memo to Matt Kozi, Solutions Architect, explaining the purpose of stored procedures and how you would implement stored procedures to simplify administration of the database.

For your example to Mr. Kozi, you will be focusing on the Salary field in the Employee table.
Stored Procedure:

a. Convert the INT to a CHAR (SQL command)
b. Insert commas, decimal places, and dollar signs where appropriate in the data for the salary attribute for the Employee entity.

To answer this question, you are to provide a written explanation, not the SQL script.

➔ Subject: Simplifying Database Administration through Stored Procedures

Hi Matt,

I trust this message reaches you in good health. I would like to explore the idea of integrating stored procedures into our database management approach. This could greatly streamline our administrative tasks and enhance our overall efficiency.

Stored procedures are sets of SQL statements with assigned names, stored in a relational database management system (RDBMS) for reuse and sharing among multiple programs. They offer several benefits, including enhanced security, data integrity, and productivity. By using stored procedures, we can reduce network traffic between clients and servers, as the commands are executed as a single batch of code.

To implement stored procedures, we should follow few standards and conventions for designing and implementing them.

This involves:

- ✓ documenting the purpose, inputs, outputs, and logic of each stored procedure in a consistent format.
- ✓ using naming conventions for readability and organization.
- ✓ writing clear and concise code following SQL programming best practices.

- ✓ we should optimize the execution time.
- ✓ resource consumption.
- ✓ network traffic of our stored procedures.

After deploying stored procedures, it is essential to maintain and document them regularly to keep them up to date and functional.

In our case, I have created a stored procedure named, “UpDated_Salary” which is been executed in separate batch of command with the help of GO command, with the help of both CREATE OR ALTER command in case we might feel a need to update anything in it.

Actual line of code includes, ALTER command where we convert the Salary field of EMPLOYEES table in Char(17) datatype as demanded as well as I have tried to implement the comma structure with decimal places and dollar signs in updated Salary field.

To run a stored procedure → execute the line below:

```
EXEC UpDated_Salary;
```

If successful ran will give something like this in output:

```
Salary has been converted to char.
```

We should use version control, backup, and recovery tools to manage our stored procedures, and document any changes, issues, or enhancements in a change log or repository. Periodic review and refactoring of stored procedures are also recommended to eliminate any obsolete or redundant code.

I believe incorporating stored procedures into our database management strategy aligns with our goal of optimizing performance, enhancing security, and simplifying administration. If you have any questions or would like to discuss this further, please feel free to reach out.

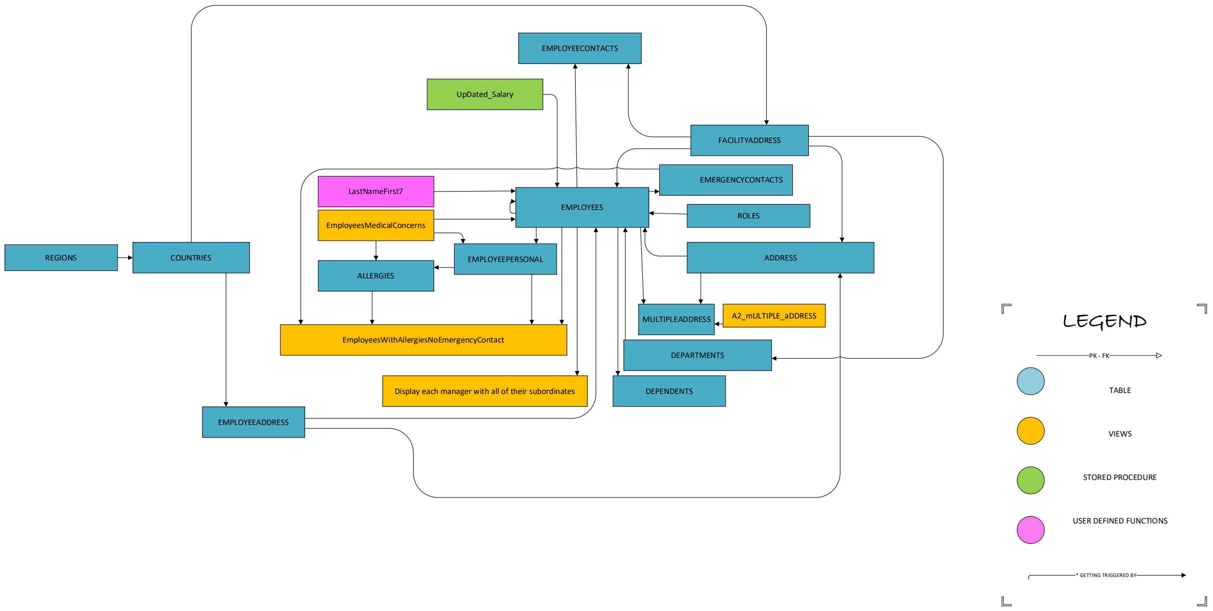
Best regards,
Group7

SECTION – C : Updating documentation

The following questions require a written explanation that must be included as a separate file from your SQL scripts. You may include Section B, Question 3's written response in this part of the submission.

1) **Create a dependency graph that shows dependencies among all tables, views, stored procedures, and functions.**

DEPENDENCY GRAPH



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?**

➔ Dependency graphs are crucial in database management, even when logical and physical data models are already in place because they provide a different perspective on the relationships and dependencies between various database objects, including views and stored procedures.

Logical and physical data models represent the data and how it is stored in the database, setting the relationship between data items. However, these models do not explicitly illustrate the dependencies between different database objects, which is where dependency graphs come in.

Dependency graphs are powerful tools for managing complicated chains of calculations, making it easier to understand calculation flows, analyse dependencies, and efficiently recalculate values after data changes. They are used to represent the flow of information among the attributes in a parse tree, helping to determine the evaluation order for the attributes.

In the context of database management, dependency graphs can help visualize and manage the complex relationships between different database objects, such as views and stored procedures. Views are used for SELECT statements, storing commonly used JOIN queries and specific columns to build virtual tables of an exact set of data. Stored procedures, on the other hand, hold more complex logic, such as INSERT, DELETE, and UPDATE statements to automate large SQL workflows.

By developing dependency graphs that include these views and stored procedures, we can better understand the impact of changes made to one component on the other parts of the database system. This understanding is essential when updating or replacing database components, helping to identify potential issues and prevent system failures.

Moreover, dependency graphs can assist in managing software dependencies, understanding the complexity of all the individual components tied to a project, and bringing it under control. They can also help anticipate blockers and bottlenecks so teams can work more efficiently.

While here are some differences between dependency graphs and data models:

Dependency Graphs	Data Models
Represent the dependencies and relationships between different database objects, such as tables, views, and stored procedures.	Represent the structure of data and how it is stored in the database, setting the relationship between data items.
Provide a visual representation of the connections and interactions between database objects, helping to understand the impact of changes and identify potential issues.	are used to describe the data contained in the database, the relationships between data items, and the constraints on data.
Help in managing complicated chains of calculations, making it easier to understand calculation flows and efficiently recalculate values after data changes.	Include three main types: conceptual, logical, and physical data models, each serving a specific purpose in the database design process.
Can assist in managing software dependencies, understanding the complexity of individual components tied to a project, and anticipating blockers and bottlenecks for more efficient work.	Logical data models define the structure of data elements and set the relationships between them, while physical data models describe the database-specific implementation of the data model.
are essential for understanding the flow of information among attributes in a parse tree, helping to determine the evaluation order for the attributes.	Logical data models are designed and developed independently from the DBMS, while physical data models are developed for a specific version of a DBMS, location, data storage, or technology to be used in the project.

In summary, while logical and physical data models provide a structural representation of data and its relationships, dependency graphs offer a detailed view of the dependencies between different database objects, including views and stored procedures. This additional layer of understanding can significantly enhance database management and administration.

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.**

➔

Stored Procedure1: exports health data from the EmployeePersonal table

Stored Procedure2: reformats emergency contact phone data from the EmployeeContacts table

Stored Procedure3: reformats Salary field in the Employee table

- ■ After executing the current stored procedure, examples are mentioned below:

1. A dirty read:
   A dirty read occurs when a transaction reads data that has not yet been committed.

   Explanation: if transaction 1 updates a row and leaves it uncommitted, and transaction 2 reads the updated row, if transaction 1 rolls back the change, transaction 2 will have read data that is considered never to have existed

   E.g., Suppose Stored Procedure1 is exporting health data from the EmployeePersonal table. If another transaction updates the health data in the same table but does not commit the changes, and then Stored Procedure1 reads this uncommitted data, it would result in a dirty read.

2. A nonrepeatable read:
   A nonrepeatable read occurs when a transaction reads the same row twice but gets different data each time.

Explanation: If transaction T1 reads data, and due to concurrency, another transaction T2 updates the same data and commits, if transaction T1 rereads the same data, it will retrieve a different value

E.g., Consider Stored Procedure3, which reformats the Salary field in the Employee table. If a transaction reads the salary of an employee, and before it completes, another transaction updates the salary of the same employee, when the first transaction reads the salary again, it will retrieve a different value, leading to a nonrepeatable read

3. A phantom read:
   A phantom read occurs when two identical queries are executed, but the rows retrieved by the two queries are different.

   Explanation: if transaction T1 retrieves a set of rows that satisfy some search criteria, and then transaction T2 generates some new rows that match the search criteria for transaction T1, this can result in phantom reads

   E.g., Let's say Stored Procedure2 reformats emergency contact phone data from the EmployeeContacts table. If a transaction retrieves a set of emergency contact phone numbers that satisfy certain criteria, and before it completes, another transaction inserts new emergency contact phone numbers that match the same criteria, the first transaction might retrieve a different set of emergency contact phone numbers when it re-executes the query, resulting in a phantom read

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

➔

For Stored Procedure3, which reformats the Salary field in the Employee table, appropriate concurrency control measures include techniques such as:

- Two-Phase Locking Protocol
- Time Stamp Ordering Protocol
- Multi-Version Concurrency Control
- Validation Concurrency Control

These techniques help in managing simultaneous operations without conflicting with each other and ensure that database transactions are performed concurrently and accurately to produce correct results without violating data integrity.

A. Two-Phase Locking Protocol:

It is a widely used concurrency control technique that ensures serializability of transactions.

It involves two phases:

1. The growing phase, during which a transaction acquires locks
2. The shrinking phase, during which a transaction releases lock.

This protocol prevents issues such as dirty reads and nonrepeatable reads by ensuring that transactions acquire all the locks they need before releasing any locks.

Time Stamp Ordering Protocol and Multi-Version Concurrency Control are other techniques that can be used to manage concurrency and ensure data consistency. These techniques use timestamps to order and manage concurrent transactions, allowing for high levels of concurrency while maintaining data integrity.

Validation Concurrency Control, also known as optimistic concurrency control, is another approach that can be suitable for scenarios where conflicts are less likely to occur. It allows transactions to execute without restrictions until they are committed, and then validates the transactions to ensure that they have not caused any conflicts. This approach can be beneficial in scenarios where conflicts are infrequent, as it minimizes the overhead associated with locking and timestamp management.

The choice of the appropriate concurrency control measure for Stored Procedure3 should consider factors such as the frequency of concurrent access, the likelihood of conflicts, and the performance overhead associated with each technique.

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

➔

Based on the provided stored procedures, appropriate concurrency control measures for the two other stored procedures can include the following techniques:

1. Stored Procedure1 (exports health data from the EmployeePersonal table):

- Appropriate Concurrency Control Measures: Given that this stored procedure involves exporting data, a suitable concurrency control measure could be the Two-Phase Locking Protocol. This protocol ensures that transactions acquire all the locks they need before releasing any locks, thus preventing issues such as dirty reads and nonrepeatable reads.

2. Stored Procedure2 (reformats emergency contact phone data from the EmployeeContacts table):

- Appropriate Concurrency Control Measures: For this stored procedure, a suitable concurrency control measure could be Multi-Version Concurrency Control. This technique keeps old versions of data items to increase concurrency, which can be beneficial in scenarios where multiple transactions are accessing and modifying the same data concurrently.

These concurrency control measures are designed to manage simultaneous operations without conflicting with each other and ensure data integrity in a multi-user database environment. The choice of the appropriate concurrency control measure for each stored procedure should consider factors such as the nature of the data being accessed and modified, the frequency of concurrent access, and the likelihood of conflicts.