Answers:

1. Create a stored procedure GetEmployeeInfo which takes @dept\_no as an input parameter, and outputs a result set which includes the following fields: emp\_no, employee full name, department name. Provide a screenshot of output results using ‘d1’ as input parameter.

**CREATE** **PROCEDURE** GetEmployeeInfo\_1

@dept\_no **char**(4)

**AS**

**BEGIN**

**select** emp\_no, emp\_fname + ' ' + emp\_lname **as** employee\_name , s.dept\_name **from** employee e , subdivision s

**where** e.dept\_no = s.dept\_no

**and** e.dept\_no = @dept\_no

**END**

**exec** GetEmployeeInfo\_1 @dept\_no = 'D1'

A screenshot of a social media post

Description automatically generated

1. Create a stored procedure IncreaseBudgetAmount which takes @project\_no and @new\_budget as input parameters and returns @message as an output parameter. The stored procedure must perform the following business rules:
   1. If @project\_no is not found, it returns the message “Invalid Project Number”
   2. If @new\_ budget is greater than the current budget amount, it must update the project budget and return the message “budget amount increased”
   3. If @new\_ budget is less than or equal to the current budget, it does nothing and return the message “New budget must be greater than the current budget”

Provide sample execution commands for all three business cases, along with a screen shot of results for each.

**CREATE** **PROCEDURE** IncreaseBudgetAmout (@project\_no **char**(4), @new\_budget **float** , @message **varchar**(25) **OUTPUT** )

**AS**

**BEGIN**

**IF** **EXISTS** (**SELECT** \* **FROM** project **WHERE** project\_no=@project\_no)

**BEGIN**

**IF** @new\_budget > (**SELECT** budget **FROM** project **WHERE** project\_no=@project\_no)

**BEGIN**

**UPDATE** project

**SET** budget=@new\_budget

**WHERE** project\_no=@project\_no

**PRINT** 'budget amount increased.'

**END**

**ELSE**

**BEGIN**

**PRINT** 'New budget must be greater than the current budget.'

**END**

**END**

**ELSE**

**BEGIN**

**PRINT** 'Invalid Project Number.'

**END**

**END**

**EXEC** IncreaseBudgetAmout @project\_no = 'p1', @new\_budget = 60000, @message = 'Project Details';

A screenshot of a social media post

Description automatically generated

A screenshot of a social media post

Description automatically generated

1. Create a User Defined Function GetBudgetAmount which takes @project\_name and returns the budget for a given project. If it cannot find the record it returns NULL. Show a SQL example of a function being used to the budget for “CRM system”

A screenshot of a social media post

Description automatically generated

**CREATE** **FUNCTION** GetBudgetAmount (@project\_name **nvarchar**(20))

**RETURNS** **int**

**AS**

**BEGIN**

**Declare** @budg **int**;

**SELECT** @budg= budget

**FROM** project

**WHERE** project\_name='CRM System';

**IF**(@budg **IS** **NULL**)

**SET** @budg= 0;

**RETURN** @budg;

**END**;

**SELECT** **DISTINCT** dbo.GetBudgetAmount('CRM system') **FROM** project

A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

Description automatically generated

1. What is the name of a single logical operation on the data to satisfy ACID property?  
   **TRANSACTION**
2. Which ACID property does the following DDLs satisfy?
   1. CREATE TABLE Customer (CustomerID int PRIMARY KEY, CustomerName varchar(100) NOT NULL)  
        
      **DURABILITY**
3. Which ACID property ensures the integrity of data reads?  
   **Consistency** :

Consistency guarantees that a transaction will not allow the database to contain inconsistent data. In other words, the transactional transformations on data bring the database from one consistent state to another.

1. Failure to write data to non-volatile memory violates which property?
   1. Atomicity
   2. Consistency
   3. **Isolation**
   4. Durability
2. State the reasons why concurrency control needed?

Concurrency control is used to address such conflicts which mostly occur with a multi-user system. It helps you to make sure that database transactions are performed concurrently without violating the data integrity of respective databases.

Concurrency control is the procedure in DBMS for managing simultaneous operations without conflicting with each another. Concurrent access is quite easy if all users are just reading data. There is no way they can interfere with one another. Though for any practical database, would have a mix of reading and WRITE operations and hence the concurrency is a challenge.

1. What is the difference between a local transaction and a distributed transaction?

A transaction is considered to be a local transaction when it is a single-phase transaction and is handled by the database directly. A transaction is considered to be a distributed transaction when it is coordinated by a transaction monitor and uses fail-safe mechanisms (such as two-phase commit) for transaction resolution.

1. When should you use the SAVE TRANSACTION statement?

We should use SAVE TRANSACTION statement for short transactions

The SAVE TRANSACTION statement, in combination with the IF or WHILE statement, is a useful transaction feature for the execution of parts of an entire transaction. On the other hand, the use of this statement is contrary to the principle of operational databases that a transaction should be as **short** as possible, because long transactions generally reduce data availability.

1. Discuss the difference between row-level and page-level locking.

Row Level: A row is the smallest resource that can be locked. The support of row-level locking includes both data rows and index entries. Row-level locking means that only the row that is accessed by an application will be locked. Hence, all other rows that belong to the same page are free and can be used by other applications. The Database Engine can also lock the page on which the row that has to be locked is stored.

Images    **ROWLOCK** Replaces the existing shared table lock with shared row locks for each qualifying row of the table.

Images    **PAGLOCK** Replaces a shared table lock with shared page locks for each page containing qualifying rows.

row-level locking maximizes concurrency because it leaves all but one row on the page unlocked. On the other hand, system overhead is increased because each locked row requires one lock.

Page-level locking (and table-level locking) restricts the availability of data but decreases the system overhead.

1. Can a user explicitly influence the locking behavior of the system?

Specifies that exclusive locks are to be taken and held until the transaction completes. If **XLOCK** is specified with ROWLOCK, PAGLOCK, or TABLOCK, the exclusive locks apply to the appropriate level of granularity.