Readings:

* ACID Property: <https://en.wikipedia.org/wiki/ACID_(computer_science)>
* Concurrency control <https://en.wikipedia.org/wiki/Concurrency_control>
* chapter 13: “Concurrency Control” (Petkovic)

Use the sample database created in previous lecture to answer the following three questions.



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 @dept\_no char(4)

AS

BEGIN

SELECT e.emp\_no, e.emp\_fname + ' ' + e.emp\_lname as full\_name,

d.dept\_name

FROM employee e

join department d on e.dept\_no = d.dept\_no

where d.dept\_no =@dept\_no

END

exec GetEmployeeInfo 'd1'

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 IncreaseBudgetAmount @project\_no char(4), @new\_budget float,

@message varchar(100) output

AS

BEGIN

IF NOT EXISTS (

SELECT 1 from dbo.project p where p.project\_no = @project\_no

)

SET @message ='Invalid Project Number'

ELSE IF EXISTS (

SELECT 1 from dbo.project p where p.project\_no = @project\_no

and @new\_budget> p.budget

)

BEGIN

UPDATE dbo.project set budget = @new\_budget where project\_no = @project\_no;

SET @message ='budget amount increased'

END;

ELSE IF EXISTS (

SELECT 1 from dbo.project p where p.project\_no = @project\_no

and @new\_budget<= p.budget

)

BEGIN

SET @message ='New budget must be greater than the current budget'

END

END

declare @message varchar(100), @projec\_no varchar(4), @new\_budget float

set @projec\_no ='f1'

set @new\_budget =10

exec IncreaseBudgetAmount @projec\_no, @new\_budget, @message output

print @message

set @projec\_no ='p1'

set @new\_budget =10

exec IncreaseBudgetAmount @projec\_no, @new\_budget, @message output

print @message

set @projec\_no ='p1'

set @new\_budget =400000

exec IncreaseBudgetAmount @projec\_no, @new\_budget, @message output

print @message

-----results of execution

Invalid Project Number

New budget must be greater than the current budget

(1 row affected)

budget amount increased

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”

create function GetBudgetAmount (@project\_name varchar(50) )

returns float

as

begin

return

(

select budget from project where project\_name = @project\_name

)

end

select dbo.GetBudgetAmount('CRM system')

1. What is the name of a single logical operation on the data to satisfy ACID property?

Transaction. A Transaction can either be implicit such as an INSERT or UPDATE or explicit when defined to BEGIN TRANSACTION …. COMMIT/ROLLBACK TRANSACTION.

1. Which ACID property does the following DDLs satisfy?

CREATE TABLE Customer (CustomerID int PRIMARY KEY, CustomerName varchar(100) NOT NULL)

Consistency is and ACID property which states that database must be valid according to all defined rules, including constraints, cascades, triggers, and any combination thereof. In this case the DDL above creates a table with PK constraint, non-nullable columns. All these rules must be valid according to ACID property( specifically Consistentency)

1. Which ACID property ensures the integrity of data reads?

Isolation. Isolation ensures that concurrent execution of transactions leaves the database in the same state that would have been obtained if the transactions were executed sequentially.

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 is need because DBMS are designed to support multiple users and processes operating various set of tasks such as Inserting records, Updating records while other processes read from those records at the same time. If concurrency is uncontrolled, the following issues with be present: dirty reds, lost update problem, non-repeatable read, and the phantom read problem.

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

A local transaction occurs within a database. Distributed transaction involves two or more databases, and often time distributed across the network(s).

1. When should you use the SAVE TRANSACTION statement?

A save point marks a specified point within the transaction so that all **updates that follow** can be canceled without canceling the entire transaction

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

Locking means that the transaction marks the data that it accesses so that the DBMS knows not to allow other transactions to modify it until the first transaction succeeds or fails.

Row versioning provides each reading transaction the prior, unmodified version of data that is being modified by another active transaction. This allows readers to operate without acquiring locks, i.e., writing transactions do not block reading transactions, and readers do not block writers.

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

YES, by either changing the isolation level using the SET TRANSACTION ISOLATION LEVEL statement or by using hints. For example, for databases in “READ COMMITED” Isolation Level, NOLOCK hint allows the user to read non-committed data which essentially skips the waiting of exclusive placed by the writer process