INF20010 / 60014 Database Systems **Week 6 Tutorial Questions**

Lost Update Concurrency
Transactions & VB.NET Transactions

Tutorial Tasks

1. **What is a statement-level rollback?**

   Statement-level rollback is a transaction control mechanism that allows a group of SQL statements to be rolled back when one of the statements fails to execute. In the event of a rollback, any changes made by the said transaction will be returned to its initial state.

2. **Describe a business scenario where a transaction has more than one SQL insert, update or delete statement to take the database from one consistent state to another.**

   Suppose we have an imaginary business that sell some kind of services, a transaction that take the database from one consistent state to another might be:

   1. Check if the customer has sufficient funds in their account to make the purchase.
   2. Minus the customer balance with the price of the purchase.
   3. Add the deducted transfer to the store owner's balance.
   4. Update the transaction history.
   5. Commit the transaction.

3. **Using the previous example, what would happen if only part of the transaction was executed and committed? How does this compare to the first two ACID principles.**

   If only part of the transaction was executed and committed, this would violate the Atomicity principle of ACID, Atomicity states that a transaction must be treated as a single, indivisible unit of work and all changes are either committed or rolled back.

   In the event of Atomicity being violated, Consistency would also be violated. Consistency refers to the fact that database should remain in a consistent state throughout a transaction, and any changes made by a transaction must be adhered to the rules and constraint defined by the schema. If only part of the transaction was executed and committed, this would mean that the rules and constraints of the database schema has been violated.

4. **What is the difference between transactions that are executed serially and a transaction schedule that is serializable?**

   Transactions that are executed serially are transactions that are executed one after another transactions has finished.

   On the other hand, a transaction schedule that is serializable is a set of transactions that can be executed in any order but till produce the same result as if they are executed serially. Therefore, it will ensure data consistency and integrity even if the transaction is executed concurrently.

Lost Update Concurrency
Transactions & VB.NET Transactions

5. What is meant by the term Concurrent Transactions?

Concurrent Transaction refers to a situation where multiple transactions are executed simultaneously or concurrently.

6. Describe how a Dirty Read may occur?

Dirty Read may occur when transaction A updates a record in a table but does not commit the change, at the same time, another transaction B attempts to read the same record before transaction A commits the change. This will mean that Transaction B will retrieve the uncommitted data from transaction A and it may perform some calculation on the said data.

7. What is another name for a Dirty Read?

Read Uncommitted is another name for a Dirty Read.

8. Imagine that Customer 123 has a balance of $100
Transaction A is supposed to increase customer 123's balance by $40.
Transaction B is supposed to decrease customer 123's balance by $10.
Both procedures run concurrently executing statements is the order shown:

Statement	Transaction A	Transaction B
1	**vbalance number :=0;**	
2	**vcustid number := 123;**	
3		**vbal number :=0;**
4		**vcustid number := 123;**
5	**Begin**	
6		**Begin**
7	**SELECT balance INTO vbalance FROM customer WHERE custid = pcustid;**	
8		**SELECT balance INTO vbal FROM customer WHERE custid = pcustid;**
9	**vbalance := vbalance + 40;**	
10	**UPDATE customer SET balance = vbalance WHERE custid = pcustid;**	
11	**Commit;**	
12	**End;**	
13		**vbal := vbal - 10;**
14		**UPDATE customer SET balance = vbal WHERE custid = pcustid;**
15		**Commit;**
16		**End;**

At the completion of Transactions A and B, customer 123's balance <u>should be</u> $130

a) **What is customer 123's actual balance at the end of statement 12?** 140

b) What is customer 123's actual balance at the end of statement 16? 90

c) A problem has occurred. What name do we give to this problem? A Lost Update

Lost Update Concurrency
Transactions & VB.NET Transactions

9. This scenario is identical to the question above.
However, statements 7 & 8 below have been modified to include a FOR UPDATE clause.

   a. **What is the effect of the FOR-UPDATE clause in the SELECT statement?** An exclusive lock has been created from customer to prevent other transactions from modifying until the locking transaction is complete.
   b. **List the sequence in which these statements will be executed.** Procedure A starting from line 7 to line 12 will be executed first, and Procedure B starting at line 8 will have to wait until Procedure A release the lock
   c. **What is customer 123's balance at the end of statement 16** $130

Statement	Procedure A	Procedure B
1	vbalance number :=0;	
2		vbalance number :=0;
3	vcustid number := 123;	
4		vcustid number := 123;
5	Begin	
6		Begin
7	SELECT balance INTO vbalance FROM customer WHERE custid = pcustid FOR UPDATE;	
8		SELECT balance INTO vbalance FROM customer WHERE custid = pcustid FOR UPDATE;
9	vbalance := vbalance + 40;	
10	UPDATE customer SET balance = vbalance WHERE custid = pcustid;	
11	Commit;	
12	End;	
13		vbalance := vbalance - 10;
14		UPDATE customer SET balance = vbalance WHERE custid = pcustid;
15		Commit;
16		End;

10. State whether each of these statements is True or False:

   a. **True / False. An employee row is locked via the Select…For Update statement. No other transactions can read data about that employee while the lock is in place**

      True, it will be an exclusive lock.

   b. **True / False. An employee row is locked via the Select…For Update statement. No other transactions can update data about that employee while the lock is in place**

      True, modification of all kinds will have to wait until the lock is released.

   c. **True / False. A transaction that contains a Select …For Update clause MUST also have an Update statement that modifies the locked row(s).**

False, in this case the transaction does not necessarily need an Update statement, as lock is used to ensure exclusive access to the selected rows, therefore, it can be accessed without making any modification.

d. True / False. A transaction that contains a Select… For Update statement does NOT have to have ANY Insert, Update or Delete SQL statements.

True, as mentioned above, a lock is to ensure exclusive access, and an access does not necessarily mean modification.

e. True / False. A Select… For Update lock remains in place until the row(s) is updated

False, the lock remains in place until the row is committed or rolled back.

f. True / False. A commit statement will unlock a row(s) locked by a Select… For Update statement

True, once a transaction is committed, the lock will be released.

g. True / False. A rollback statement will unlock a row(s) locked by a Select… For Update statement

True, once a transaction is rolled back, the lock will be released.

h. True / False. An exception that causes a transaction to end will cause a rollback statement.

True, when a transaction encounters an error or exception, it can either be handled by the transaction or rolled back by the transaction and effectively undoing any changes made to the selected rows.

INF20010 / 60014 Database Systems **Week 6 Tutorial Questions**

Lost Update Concurrency
Transactions & VB.NET Transactions

Consider the following VB Code:

```
1  Imports Oracle.DataAccess.Client
2  Public Class Form1
3      Private Sub TestOracleButton_Click(ByVal sender As System.Object, ByVal e As System.E
4          TestTrans()
5      End Sub
6      Public Sub TestTrans()
7          Dim rvConn As Oracle.DataAccess.Client.OracleConnection
8          rvConn = CreateConnection()
9          Dim vOutcome As String = ""
10         Dim rvTran As Oracle.DataAccess.Client.OracleTransaction = Nothing
11         Try
12             rvConn.Open()
13             rvTran = rvConn.BeginTransaction(IsolationLevel.ReadCommitted)
14             Update_Table_1(rvConn)
15             Update_Table_2(rvConn)
16             Update_Table_3(rvConn)
17             rvTran.Commit()
18             vOutcome = ("Transaction Finished OK")
19         Catch ex As Exception
20             rvTran.Rollback()
21             vOutcome = ex.Message
22         Finally
23             rvConn.Close()
24             MessageBox.Show(vOutcome)
25         End Try
26     End Sub
27     Public Sub Update_Table_1(ByVal rvConn As Oracle.DataAccess.Client.OracleConnection)
28         Dim rvCmd As New Oracle.DataAccess.Client.OracleCommand
29         rvCmd.Connection = rvConn
30         rvCmd.CommandType = CommandType.StoredProcedure
31         rvCmd.CommandText = "SP_UPDATE_TABLE1"
32         rvCmd.ExecuteNonQuery()
33     End Sub
```

Assume that code for sub procedures **update_table_2** & **update_table_3** are similar to lines 28-34 above.

11. **What is the purpose of line 10?** Initialize a transaction variable to the value of nothing, as the transaction has yet to begin.
12. **What is the purpose of line 13?** Set the isolation level to read committed, meaning that the transaction can only read committed data from other transaction.
13. **What happens when VB executes line 12 & 13?** Line 12 will open the connection and line 13 will begin the transaction
14. **What happens when VB executes line 14?** It will call Update_Table_1
15. **What would happen if line 33 causes a Raise_Application_Error within Oracle?** If there were to be an exception it will be caught by line 19 and will raise an exception which will subsequently rollback all the uncommitted data to its initial state.
16. **Do you think that sp_update_table1 & sp_update_table2 & sp_update_table3 will each contain a commit statement? Why / Why not?**

    As all the procedures are parts of a larger transaction that involves multiple operations and should be committed as a single unit, a COMMIT statement for each procedure would not be necessary, as we can also see a single COMMIT statement for all the procedures in line 17.

Lost Update Concurrency
Transactions & VB.NET Transactions

17. **Do you think that sp_update_table1 & sp_update_table2 & sp_update_table3 will each contain a rollback statement? Why / Why not?**

    As all the procedures are parts of a larger transaction that involves multiple operations and should be rolled back as a single unit, a rollback statement for each procedure would not be necessary, as we can also see the rollback mechanism from line 19 to line 21.

18. **What circumstances cause a Commit to occur in this code?**

    If line 13 to line 17 occur without any exception, then the commit statement will occur.

19. **What circumstances cause a Rollback to occur in this code?**

    If line 13 to line 17 raise an exception, then a roll back will be raised.

20. **What is the purpose of the Finally block of code?**

    It is used to ensure that certain actions are taken regardless of whether an exception was thrown. It is usually used to close the connection to a database.

Lab Tasks

Continue with Assignment work