**Lab 8 – Advanced Databases**

**Section A**

1. What is meant by the term *transaction*?
   1. A transaction is when a user or an application program carries out a series of actions which can access or alter content held in a database.
2. List 5 reasons why a transaction may fail to complete successfully.
   1. A transaction may fail to complete successfully in the following areas system crashes, physical disasters, carelessness, sabotage or application software errors.
3. A number of problems can occur when concurrent access to a database is allowed – provide a list of these problems.
   1. One of several problems that may occur include when two processes attempt to update the same data the same time. This creates a race condition. The process that updates the data first ‘loses’ and its data is overwritten by the second process while the process that updates the data first ‘loses’ and its data is overwritten by the second process. This is named the lost update problem.
   2. The second occurs when an application process tries to read uncommitted data. Here an application process is allowed to see the intermediate results of another transaction prior to that other transaction being committed to the database. This is called the uncommitted dependency problem.
   3. The third occurs when an application process submits the same query in a transaction but gets different results each time. This shows that transactions focus on reading the database to produce inaccurate results if they are allowed read the partial results of other incomplete transactions that are engaged in updating the database. This is named the inconsistent analysis problem.
   4. The final occurs when application process submits the same query in a transaction but get additional rows in the result set each time. Subsequent fetches of the same data in the same transaction returns additional rows that were not in the original result set. This is the phantom read problem.
4. In each of the examples below, indicate what type of problem is encountered:
   1. Lost update problem

|  |  |  |
| --- | --- | --- |
| Time | T1 | T2 |
| t1 | read\_item(X) |  |
| t2 | X:=X\*10 |  |
| t3 |  | read\_item(X) |
| t4 | write\_item(X) |  |
| t5 |  | write\_item(X) |

1. The Inconsistent Analysis Problem

|  |  |  |
| --- | --- | --- |
| Time | T1 | T2 |
| t1 | read\_item(X) |  |
| t2 | X:=X\*10 |  |
| t3 | write\_item(X) |  |
| t4 |  | Total:=0 |
| t5 |  | read\_item(X) |
| t6 |  | Total:=Total+X |
| t7 |  | read\_item(Y) |
| t8 |  | Total:=Total+Y |
| t9 | read\_item(Y) |  |
| t10 | Y:=Y\*10 |  |
| t11 | write\_item(Y) |  |

1. Uncommitted Dependency Problem

|  |  |  |
| --- | --- | --- |
| Time | T1 | T2 |
| t1 | read\_item(X) |  |
| t2 | X:=X\*10 |  |
| t3 | write\_item(X) |  |
| t4 |  | read\_item(X) |
| t5 |  | write\_item(X) |
| t6 | ABORT |  |

1. What are the ACID properties?
   1. The A in ACID stands for Atomicity which means that a transaction is either performed in its entirety or not performed at all, there’s no in-between.
   2. The C in ACID stands for Consistency which means that a transaction must transform the database from one consistent state to another.
   3. The I stand for Isolation this means that transactions must be executed independently of one another. The partial effects of incomplete transactions should not be visible to other transactions.
   4. The D stands for Durability which permanently records the effects of a committed transaction in the database and must not be lost because of any subsequent failure.
2. What is meant by the term *serializable* with regards to database transactions?
   1. Serializable means that transactions run in such a way that they appear to be executed one at a time rather than concurrently.

**Section B**

1. Open two instances of MySQL Workbench.
2. Using one of the sessions create a new database called **transtest**

CREATE DATABASE transtest;

1. Create a table called **concurrency\_test**

USE transtest;

CREATE TABLE concurrency\_test ( cid INTEGER, cdesc VARCHAR(50) ) ENGINE=InnoDB;

1. Insert some records into the table, select the records and note result

INSERT INTO concurrency\_test VALUES (1, "First Record");

INSERT INTO concurrency\_test VALUES (2, "Second Record");

SELECT \* FROM concurrency\_test;

1. Perform the following commands in the relevant MySQL session window and note results

|  |  |
| --- | --- |
| **SESSION 1** | **SESSION 2** |
| START TRANSACTION;  INSERT INTO concurrency\_test VALUES (3, "Third Record");  INSERT INTO concurrency\_test VALUES (4, "Fourth Record");  SELECT \* FROM concurrency\_test; |  |
|  | USE transtest;  SELECT \* FROM concurrency\_test; |
| COMMIT; |  |
|  | SELECT \* FROM concurrency\_test; |

1. Perform the following commands in the relevant MySQL session window and note results

|  |  |
| --- | --- |
| **SESSION 1** | **SESSION 2** |
| INSERT INTO concurrency\_test VALUES (5, "Fifth Record");  SELECT \* FROM concurrency\_test;  ROLLBACK;  SELECT \* FROM concurrency\_test; |  |

1. Perform the following commands in the relevant MySQL session window and note results
2. Rem to turn off Safe Mode : SET SQL\_SAFE\_UPDATES = 0;

|  |  |
| --- | --- |
| **SESSION 1** | **SESSION 2** |
| START TRANSACTION;  UPDATE concurrency\_test SET cdesc="Updated" WHERE cid = 4; |  |
|  | SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;  SELECT \* FROM concurrency\_test WHERE cid = 4; |
| ROLLBACK;  SELECT \* FROM concurrency\_test; |  |

Which problem is manifesting itself here?