**Lab 1 – Advanced Databases**

**Pre-requisites:**

For Section A:

1. Read through and watch all lectures and videos concerning transactions and concurrency.

For Section B:

1. Download a local instance of MySQL workbench and MySQL shell (these often install together). Follow instructions on Moodle on how to do this.
2. Follow along with video tutorials on how to use MySQL shell with transaction commands i.e., Begin transaction, commit, rollback, setting isolation levels etc.

**Section A**

1. What is meant by the term *transaction*?

A change in the database

1. List 5 reasons why a transaction may fail to complete successfully.
2. A number of problems can occur when concurrent access to a database is allowed – provide a list of these problems.
3. In each of the examples below, indicate what type of problem is encountered:

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

|  |  |  |
| --- | --- | --- |
| 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) |  |

|  |  |  |
| --- | --- | --- |
| 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?
2. What is meant by the term *serializable* with regards to database transactions?

**Section B**

1. Open two MySQL monitor prompts.
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; |

A screenshot of a computer screen

Description automatically generated

First window inserts new rows and views the updates to those rows.

Second window uses same database and attempts to view rows. Initially this window cannot see the updates from window 1. Window 1 commits and finally window 2 can observe the updates. This is because of the isolation level set by default in the sql shell (see REPEATABLE READ).

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 (5, "Fifth Record");  SELECT \* FROM concurrency\_test;  ROLLBACK;  SELECT \* FROM concurrency\_test; |  |

Session 1 starts a transaction and inserts a new row. The table is observed after the insertion. A rollback undoes the insertion. This is then observable in the read.

1. Ensure that SQL\_SAFE\_UPDATES (Safe Mode) is turned off:

SHOW VARIABLES LIKE ‘SQL\_SAFE\_UPDATES’;

By default, SQL\_SAFE\_UPDATES is turned on (= 1). We need it to be turned off.

SET SQL\_SAFE\_UPDATES = 0;

Ensure that it is turned off:

SHOW VARIABLES LIKE ‘SQL\_SAFE\_UPDATES’;

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

|  |  |
| --- | --- |
| **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; |  |
|  | If continued here there will be a mistake |

Which problem is manifesting itself here?

The problem here is the uncommitted dependency problem.

A screenshot of a computer screen

Description automatically generated