Assignment 3: Explain the ACID properties of a transaction in your own words. Write SQL statements to simulate a transaction that includes locking and demonstrate different isolation levels to show concurrency control.

1. **Atomicity**: This property ensures that a transaction is treated as a single, indivisible unit, which either completely succeeds or completely fails. If any part of the transaction fails, the entire transaction is rolled back, and the database state is left unchanged as if the transaction never occurred.
2. Consistency: Transactions must transition the database from one valid state to another valid state. Consistency ensures that the database rules, constraints, and relationships are maintained across all transactions.
3. Isolation: This property ensures that transactions executed concurrently lead to the same state that would be obtained if they were executed serially. Isolation helps prevent transactions from interfering with each other and prevents issues such as dirty reads, non-repeatable reads, and phantom reads.
4. Durability: Once a transaction has been committed, it must remain so, even in the event of errors, crashes, or power loss. Durability is typically ensured by using transaction logs that can replay changes made during the transaction to recover the committed state.

Transaction Simulation with Explicit Locking

START TRANSACTION;

-- Lock the rows in the Accounts table for Alice and Bob

SELECT * FROM Accounts WHERE AccountID IN (1, 2) FOR UPDATE;

-- Deduct from Alice's account

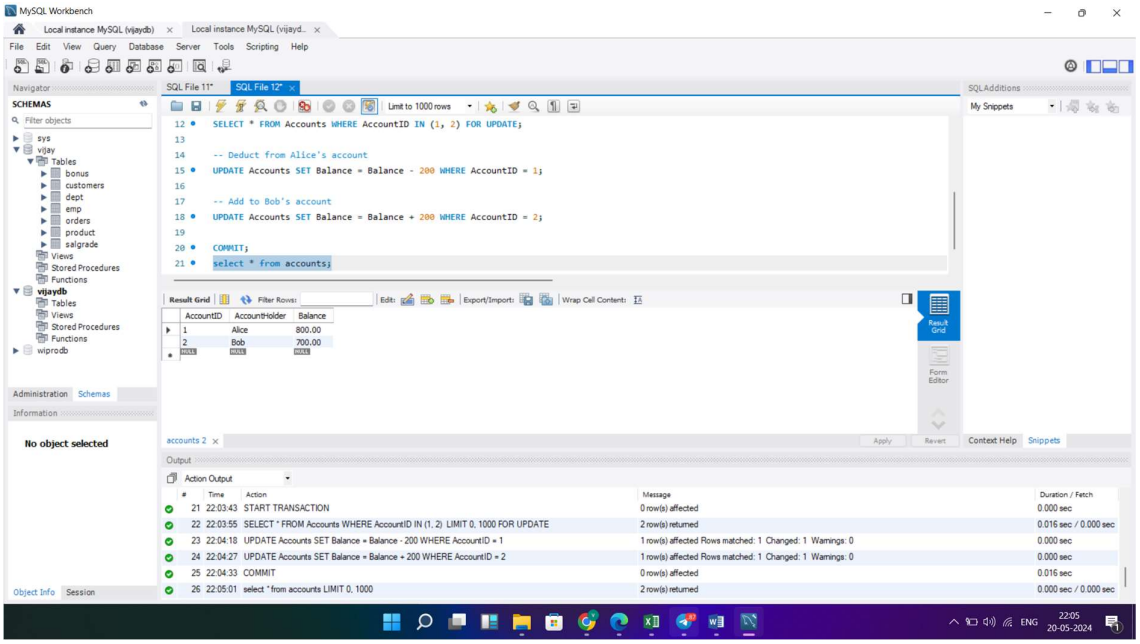
UPDATE Accounts SET Balance = Balance - 200 WHERE AccountID = 1;

-- Add to Bob's account

UPDATE Accounts SET Balance = Balance + 200 WHERE AccountID = 2;

COMMIT;

N n ,n,mnkmml/



Demonstrating Different Isolation Levels

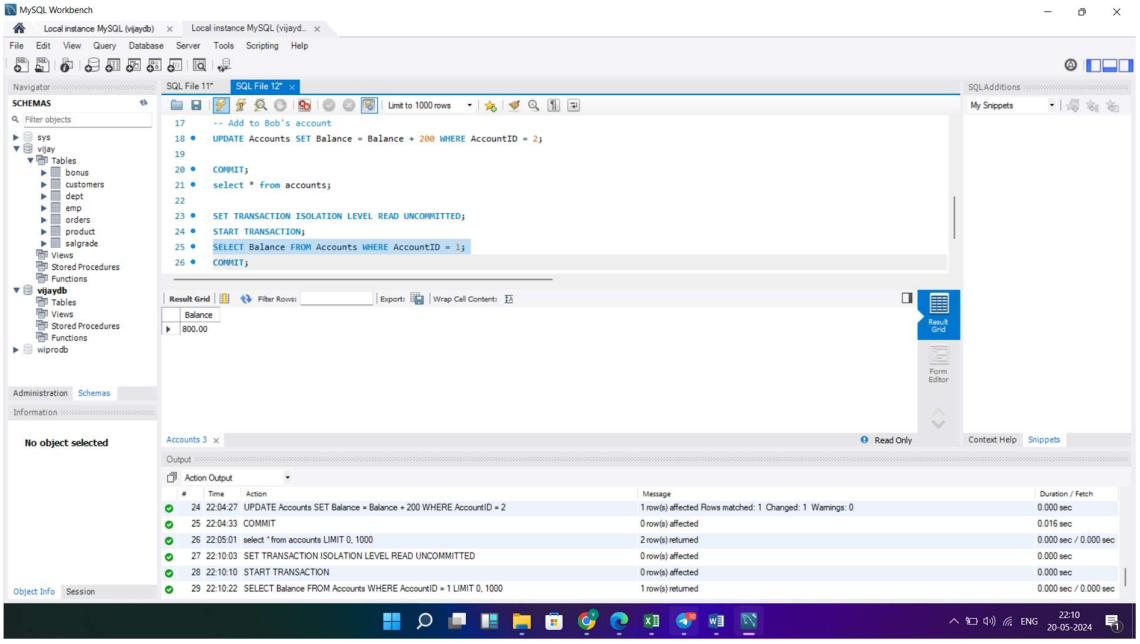
READ UNCOMMITTED: Allows dirty reads. One transaction may see uncommitted changes made by another.

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;

START TRANSACTION;

SELECT Balance FROM Accounts WHERE AccountID = 1;

COMMIT;



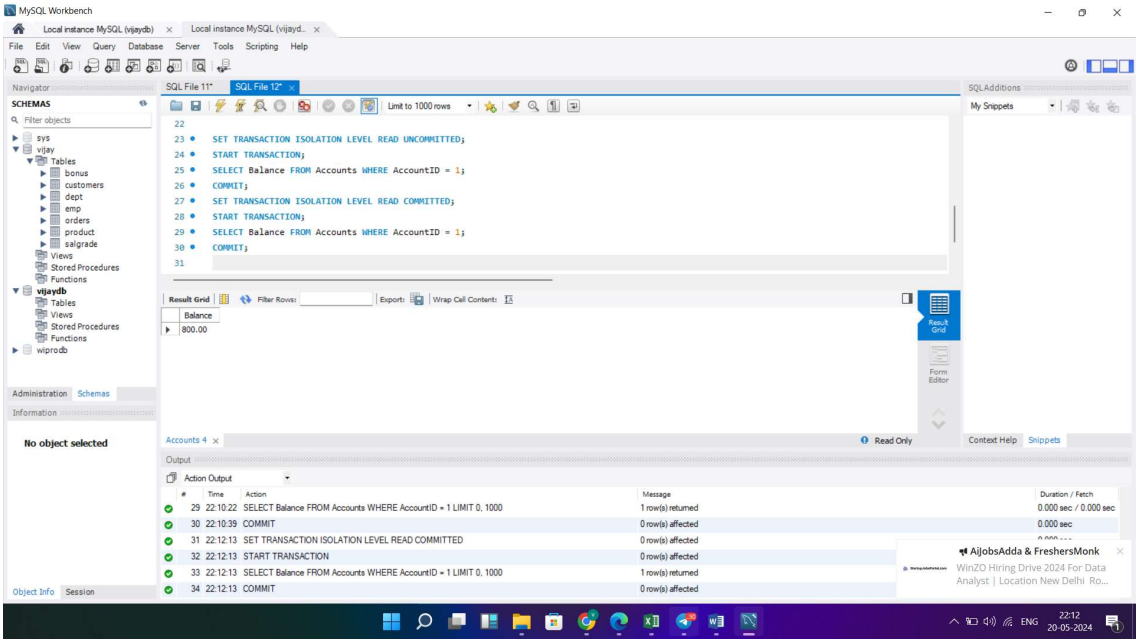
READ COMMITTED: Prevents dirty reads. Each statement within the transaction sees only data that has been committed before the statement begins

SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

START TRANSACTION;

SELECT Balance FROM Accounts WHERE AccountID = 1;

COMMIT;


MySQL Workbench
Local instance MySQL (vijaydb)
Local instance MySQL (vijayd..
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sys
vijay
Tables
bonus
customers
dept
emp
orders
product
salgrade
Views
Stored Procedures
Functions
vijaydb
Tables
Views
Stored Procedures
Functions
wiprodb
Administration Schemas
Information
No object selected
Object Info Session
SQL File 11*
SQL File 12*
Limit to 1000 rows
22
23 SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
24 START TRANSACTION;
25 SELECT Balance FROM Accounts WHERE AccountID = 1;
26 COMMIT;
27 SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
28 START TRANSACTION;
29 SELECT Balance FROM Accounts WHERE AccountID = 1;
30 COMMIT;
31
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Balance
800.00
Result Grid
Form Editor
Accounts 4
Read Only
SQLAdditions
My Snippets
Context Help
Snippets
Output
Action Output
Time Action Message Duration / Fetch
29 22:10:22 SELECT Balance FROM Accounts WHERE AccountID = 1 LIMIT 0, 1000 1 row(s) returned 0.000 sec / 0.000 sec
30 22:10:39 COMMIT 0 row(s) affected 0.000 sec
31 22:12:13 SET TRANSACTION ISOLATION LEVEL READ COMMITTED 0 row(s) affected
32 22:12:13 START TRANSACTION 0 row(s) affected
33 22:12:13 SELECT Balance FROM Accounts WHERE AccountID = 1 LIMIT 0, 1000 1 row(s) returned
34 22:12:13 COMMIT 0 row(s) affected
AijobsAdda & FreshersMonk
WinZO Hiring Drive 2024 For Data Analyst | Location New Delhi Ro...
ENG
22:12
20-05-2024