Transactions and data concurrency in DB2

Task. 1. **Isolation level CURSOR STABILITY and option READ COMMITTED**

1. Start command line processor (terminal A)
2. Start DB2 instance (`db2start`)
3. Connect to SAMPLE db as STUDENT
4. List tables in db SAMPLE
5. If the table TEST exists, drop it and then create a table TEST(kol integer)
6. Insert 10 rows to table TEST:
   db2 insert into test select 1 from syscat.tables fetch first 10 rows only
7. Disconnect from SAMPLE (db2 connect reset)
8. Read the value of the db config parameter cur_commit for db SAMPLE
   get db cfg for sample
9. Disable option cur_commit
   update db cfg for sample using cur_commit disabled
10. Enter the interactive mode; disable the auto-commit option
11. Connect to SAMPLE db; display all rows from TEST table
12. Start new command line processor window (terminal B) and connect to SAMPLE db
13. In window A run update test set kol=10
14. In window B select all rows from table TEST; what happened?
15. In window A finish the transaction with COMMIT (observe the window B)
16. Disconnect from SAMPLE in windows A and B
17. Enable the cur_commit option for SAMPLE
    update db cfg for sample using cur_commit on
18. Connect to SAMPLE db in windows A and B
19. In window A, run the command update test set kol=20
20. In window B select all rows from table TEST; what happened? What data is returned?
21. In window A finish the transaction with COMMIT and again in widow B select all rows from table TEST; compare the result set with the set of rows previously obtained.
22. Disconnect from SAMPLE in windows A and B

Task 2. **Isolation levels Repeatable Read and Read Stability**

1. In window A change isolation level for session to Repeatable Read (RR)
   change isolation to RR
2. Connect to SAMPLE in widows A and B
3. In window A, run the statement: select * from test where kol=20
4. In window B insert new row to TEST: insert into test values (20)
   Why the statement waits?
5. Commit the transaction in window A, observing the effect in window B
6. Disconnect from SAMPLE in windows A and B
7. In window A change isolation level for session to Read Stability (RS)
   change isolation to RS
8. Connect to SAMPLE in widows A and B
9. In window A, run the query: select * from test where kol=20
10. In window B insert new row to TEST: insert into test values (20)
    Why the query does not wait for window A to commit?
11. In window A, run the query: select * from test where kol=20
    How many rows are returned? What effect occurred?
12. Commit transaction in window A; disconnect from db in windows A and B

Task. 3. Isolation level Uncommitted Read (UR)

1. In window A change isolation level for session to Repeatable Read (RR)
2. Connect to SAMPLE in widows A and B
3. In window A run the statement `update test set kol=30`
4. In window B try to select all rows from TEST, is this possible?
5. In window A commit the transaction, observe the effect in window B
6. Disconnect from db in window B
7. In window B change isolation level for session to Uncommitted Read (UR)
8. Connect to db in window B
9. In window A run the statement `update test set kol=40`
10. In window B try to select all rows from TEST, is this possible? Why?
11. In window A commit the transaction
12. In window B select all rows from TEST, compare the result set with the previously obtained
13. In window A run the statement `update test set kol=99`
14. In window B select all rows from TEST
15. Rollback the transaction in window A
16. In window B select all rows from TEST, compare the result set with the previously obtained. What effect occurred?
17. Disconnect from db in windows A and B