-- part d

-- set autocommit off

--1

select score from movie where title = 'My Cousin Vinny';

update movie set score = (select max(score) from movie) where title = 'My Cousin Vinny';

-- My partner gets 7.5 while I get 10. for step 6 and 7.

-- Step 8 My partner tries to update up it hangs. This is because I have a lock on the record and his process cannot update

until I release the lock.

--step 9 he cancels then requires and still sees 7.5 he can read but I have not committed the changes yet.

-- I commit then he sees the correct score of 10.

--IT releases the locks once the original process commits. When a process commits then it will release all of the locks that it has obtained.

step 16;

after I commit my partner still sees the score as 7.5; This is because the serializable transaction level my partner's transaction will read the score as 7.5 until his transaction commits. I can update the score of My Cousin Vinny as much as i want be he will not be able to see any other score.

read commited snapshot reads the score of the last commited transation. once my partner commits his trasaction then he will be able to see the updated score.

--part e

step 5: My partner is allowed to update the score of My Cousin Vinny;

SELECT \* FROM movie WHERE title = 'My Cousin Vinny' FOR UPDATE OF score NOWAIT;

--step 8 My partner is not allowed to update the score. THe FOR UPDATE locks records that are being used for the query. the nowait clause tells oracle not to wait if the recored has been locked by another trasaction.

step 11;

I receive and error ORA-0054;resource busy and acquire with NOWAIT specified or timeout expired. This is expected becuase The nowaite clause is set so my trasaction will not wait for resources that are locked by another trasaction.

--part F:

INSERT INTO movie(id, title, yr, score, votes, director) VALUES(2632454,'Kingsman: The Secret Service', 2014, 8.2, 84535, 'Matthew Vaughn');

step 7; My partner saw the inserted row only. This was different from his first select where he saw all 164 movies that started with K. This is a non-repeatble read at the default isolation level.

Phantom reads can also occur at this level because i can be inserting and/or updating a recored while my partner is querying these records. This can be prevented in oracle by changing the isolation level to serializable.

--Part G:

1: Oracles CCT is not conservative;

2. oracle cct is strict;

3. Oracle obays 2-phase locking.

4. Oracle's cct can lead to phatom problems in the default isolation level. We saw this in step 7 of part F.

5. Oracle's default cct can lead to deadlocks because the database does not employ a conservative 2-phase locking stratigy.

6. Oracle's CCT cannot lead to a livelock.

7. Oracle's cct default isolation level read committed does not allow dirty reads this means that it is not possible for a cascading rollback. It is possible to set the isolation level to read uncommmitted which does allow for dirty reads. In this case it would be possible for oracle to have a cascading rollback.

8. Oracle's cct dedauld isolation level read committed is recoverable. It is possible to set the isolation level to read uncommitted which would allow,dirty reads, and consequently allow schedules to become unrecoverable.