**2. If a failed transaction was increasing a salary in the database, if redo is applied to that transaction, will this result in a salary increase that’s double the desired increase?  Explain.**

Ans: The redo operation will make sure that the salary is increased. It will increase the salary only if the previous operation was unable to make the required changes. Hence applying the redo operation will not result in double the desired salary.

**4. Find and describe the four levels of transaction isolation in the SQL Standard.  Which of these is implemented in Oracle SQL and what are the SQL statements?**

Ans: The four levels of transaction isolation are:

**Read Uncommitted**: Here the transactions can see uncommitted changes made by other transactions. Dirty reads, non-repeatable reads, and phantom reads are possible.

**Read Committed:** Here thetransactions only see data that has been committed by other transactions. This level prevents dirty reads but allows non-repeatable reads and phantom reads.

**Repeatable Read:** This levelensures that within a transaction, any data read will not change. This level prevents dirty reads and non-repeatable reads, but phantom reads are still possible.

**Serializable:** This level ensures that concurrent execution of transactions behaves as if they were executed sequentially. This level prevents dirty reads, non-repeatable reads, and phantom reads.

Read Commit and Serializable can be implemented in Oracle.

**SET TRANSACTION ISOLATION LEVEL READ COMMITTED;**

**SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;**

**6. Locking can occur at different levels of granularity.  Two typical examples are the row level and the database page level.  For a database system designer, what are the tradeoffs between these two different levels?  Which would you choose for a database product that is intended to be sold for a wide range of applications?**

Ans: The Row level locking provides high concurrency and minimizes the chances of lock contention. But it becomes difficult to manage large number of individual locks possibility of deadlock arises if not handles correctly.

The page level locking is easier to implement and has reduced risk of deadlocks. But it provides lower concurrency and flexibility. And transactions may need to access rows within same locked page, which may lead to performance bottlenecks.

I would choose row-level locking for a database product intended for a wide range of applications. It is more versatile and flexible and allows the database system to handle diverse workloads efficiently, especially in scenarios where multiple users or transactions are accessing and modifying different rows concurrently.

**8. Suppose a database is to be split between two different servers, with some tables on one server and some on another.  Do you think the operation of COMMIT and ROLLBACK will work in the same way across two servers?**

Ans: Yes, to achieve this functionality we use two phase commit protocol.

In first phase each server is sent its part of transaction. After the processing each server reports back READY TO COMMIT to the initiating server.

In second phase, if the any server is unable to complete its part of transaction, all receive rollback command. Only when all servers indicate they are ready to commit, the COMMIT command is issued.