4

tread.sql -> runs at isolation level READ ONLY, as it only performs a READ/SELECT and therefore cannot corrupt the database in any way.

twrite.sql -> runs at isolation level READ COMMITTED, as there's just one update query

twrite2.sql -> runs at isolation level SERIALIZABLE, as there are multiple update queries and they all need to read from one snapshot of data

**Justification for twrite.sql running at READ COMMITTED**

If twrite.sql was running at isolation level read,

| **twrite2.sql** | **twrite.sql** |
|---|---|
| **UPDATE POSITION**<br>**SET** SALARY = 1.1*SALARY<br>**WHERE** P# = &1; | |
| | **UPDATE POSITION**<br>**SET** SALARY = SALARY + 10<br>**WHERE** P# = &1;<br><br>This query works on uncommitted data from the previous query (twrite2.sql query 1), corrupting the database |
| **UPDATE POSITION**<br>**SET SALARY = (SELECT SALARY**<br>**FROM POSITION**<br>**WHERE P# = &1)**<br>**WHERE P# = &2;** | |

### Justification for twrite2.sql running at SERIALIZABLE

If twrite2.sql was running at isolation level READ COMMITTED,

| **twrite2.sql** | **twrite.sql** |
|---|---|
| **UPDATE POSITION**<br>**SET** SALARY = 1.1*SALARY<br>**WHERE** P# = &1; | |
| | **UPDATE POSITION**<br>**SET** SALARY = SALARY + 10<br>**WHERE** P# = &1; |
| **UPDATE POSITION**<br>**SET SALARY = (SELECT SALARY**<br>**FROM POSITION**<br>**WHERE P# = &1)**<br>**WHERE P# = &2;**<br><br>This query would work on the result of twrite.sql corrupting the database if it wasn't SERIALIZABLE | |