**TRANSACTIONS**

**Non-conflicting transactions: Those transactions that do not change/modify the data object they access. They simply read the data object**

Schedule 1: Read the points of customer ID = 5

|  |  |
| --- | --- |
| **Transaction 1** | **Transaction 2** |
| Read(p) |  |
|  | Read(p) |
| Commit |  |
|  | Commit |

Transaction 1:

**START TRANSACTION;**

**SELECT points FROM customers**

**WHERE customer\_id=5;**

**COMMIT;**

Transaction 2:

**START TRANSACTION;**

**SELECT points FROM customers**

**WHERE customer\_id=5;**

**COMMIT;**

Schedule 2: Read the unit price of Maggi and Hoodie

|  |  |
| --- | --- |
| **Transaction 3** | **Transaction 4** |
| Read(m) |  |
|  | Read(h) |
| Commit |  |
|  | Commit |

Transaction 3:

**START TRANSACTION;**

**SELECT unit\_price**

**FROM products**

**WHERE product\_id=4;**

**COMMIT;**

Transaction 4:

**START TRANSACTION;**

**SELECT unit\_price**

**FROM products**

**WHERE product\_id=5;**

**COMMIT;**

**Conflicting transactions: Those transactions that change the data object they access. E.g., Write-Read (WR), Read-Write (RW), Write-Write (WW)**

1. If you run the transactions in different sessions,

Schedule 3: Write the points of customer ID = 5, without committing. In another transaction, read the points of the customer.

|  |  |
| --- | --- |
| **Transaction 5** | **Transaction 6** |
| Write (p) |  |
|  | Read(p) |
|  | Commit |

Transaction 5:

**SELECT TRANSACTION;**

**UPDATE customers**

**SET points = 100**

**WHERE customer\_id = 5;**

Transaction 6:

**START TRANSACTION;**

**SELECT points FROM customers**

**WHERE customer\_id=5;**

**COMMIT;**

This will return the points as 0 in the 2nd session, even though we updated them in the 1st session. This is a conflict.

1. If you run the transactions in the same session, you would need to create save points and use rollbacks to exhibit conflicts.

Schedule 4: Begin Transaction 7, create a save point, write the points of customer ID = 5, rollback to the save point, read the points of the customer ID = 5, commit transaction 7.

|  |  |
| --- | --- |
| **Transaction 7** | **Transaction 8** |
| Savepoint() |  |
| Write(p) |  |
|  | Rollback() |
|  | Read(p) |
|  | Commit |
| Commit |  |

Transaction 7,8:

**START TRANSACTION;**

**SAVEPOINT transaction\_7;**

**UPDATE customers**

**SET points = 100**

**WHERE customer\_id=5;**

**ROLLBACK to transaction\_7;**

**SELECT points**

**FROM customers**

**WHERE customer\_id=5;**

**COMMIT;**

This nullifies the effect of transaction 7, which can lead to undesired results. This is a conflict.