 **Dirty Read** → Read uncommitted data.

 **Non-Repeatable Read** → Same row’s value **changed** between reads.

 **Phantom Read** → **New row(s)** appeared/disappeared between reads.

**SQL Isolation Levels (ANSI Standard)**

There are **4 standard levels** (from weakest → strongest):

**1. READ UNCOMMITTED**

* Transactions can **read uncommitted (dirty) data** from other transactions.
* Fastest, but least safe.
* Problems: **Dirty Reads, Non-Repeatable Reads, Phantom Reads**.

✅ Example: Reading a balance that is updated but not committed yet.

**2. READ COMMITTED (Default in Oracle, SQL Server, PostgreSQL)**

* Transactions can **only read committed data**.
* Prevents **dirty reads**, but still allows:
  + **Non-Repeatable Reads**
  + **Phantom Reads**

✅ Example: You read salary = 5000, another transaction commits salary = 6000, you read again → changed.

**3. REPEATABLE READ (Default in MySQL InnoDB)**

* Ensures that if a transaction reads the same row twice, it gets the **same value**.
* Prevents **Dirty Reads** and **Non-Repeatable Reads**.
* But **Phantom Reads** are still possible.

✅ Example: You query "all employees with salary > 5000". Another transaction inserts a new employee with salary 6000. When you query again, you see an extra row (phantom).

**4. SERIALIZABLE**

* Strongest isolation level.
* Transactions are executed as if they run **one after another (serially)**.
* Prevents **Dirty Reads, Non-Repeatable Reads, and Phantom Reads**.
* Safest, but slowest (uses locks on ranges).

✅ Example: If you select employees with salary > 5000, no one can insert new rows in that range until you finish.

……………………………………………………………………………………………………………………….