

Difference between Dirty Read, Non Repeatable Read and Phantom Read in Database

Dirty Read:-

Dirty read occurs when one transaction is changing the record, and the other transaction can read this record before the first transaction has been committed or rolled back. This is known as a dirty read scenario because there is always the possibility that the first transaction may rollback the change, resulting in the second transaction having read an invalid data.

Dirty Read Example:-

Transaction A begins.
UPDATE EMPLOYEE SET SALARY = 10000 WHERE EMP_ID= '123';

Transaction B begins.
SELECT * FROM EMPLOYEE;
(Transaction B sees data which is updated by transaction A. But, those updates have not yet been committed.)

Non-Repeatable Read:-

Non Repeatable Reads happen when in a same transaction same query yields to a different result. This occurs when one transaction repeatedly retrieves the data, while a difference transactions alters the underlying data. This causes the different or non-repeatable results to be read by the first transaction.

Non-Repeatable Example:-

Transaction A begins.
SELECT * FROM EMPLOYEE WHERE EMP_ID= '123';

Transaction B begins.
UPDATE EMPLOYEE SET SALARY = 20000 WHERE EMP_ID= '123';
(Transaction B updates rows viewed by the transaction A before transaction A commits.) If Transaction A issues the same SELECT statement, the results will be different.

Phantom Read:-

Phantom read occurs where in a transaction execute same query more than once, and the second transaction result set includes rows that were not visible in the first result set. This is caused by another transaction inserting new rows between the execution of the two queries. This is similar to a non-repeatable read, except that the number of rows is changed either by insertion or by deletion.

Phantom Read Example:-

Transaction A begins.
SELECT * FROM EMPLOYEE WHERE SALARY > 10000 ;

Transaction B begins.
INSERT INTO EMPLOYEE (EMP_ID, FIRST_NAME, DEPT_ID, SALARY) VALUES ('111', 'Jamie', 10, 35000);
Transaction B inserts a row that would satisfy the query in Transaction A if it were issued again.



Related Posts:

1. [Difference between Optimistic Locking & Pessimistic Locking in Database](#)
2. [Isolation Levels in Database](#)
3. [Displaying Even and Odd number of Records in Database](#)
4. [How to eliminate the duplicate rows in Database](#)
5. [How to Retrieve Nth Maximum Salary in Database](#)
6. [Performance and Tuning Tips in Database](#)

March 9th, 2015 | Tags: [SQL](#) | Category: [Database, Java](#)