Transactions

When the application logic needs to execute a sequence of SQL commands in an atomic fashion, then the commands need to be grouped as a logical unit of work (LUW) called SQL transaction In everyday life, people conduct different kind of business transactions buying products, ordering travels, changing or canceling orders, buying tickets to concerts, paying rents, electricity bills, insurance invoices, etc. Transactions are recoverable units of data access tasks in terms of database content manipulation.

Transactions group a set of tasks into a single execution unit. Each transaction begins with a specific task and ends when all the tasks in the group successfully complete. If any of the tasks fail, the transaction fails. Therefore, a transaction has only two results: **success** or **failure**.

Incomplete steps result in the failure of the transaction. A database transaction, by definition, must be atomic, consistent, isolated and durable. These are popularly known as **ACID properties**.

Properties of Transactions

Transactions have the following four standard properties, usually referred to by the acronym **ACID**.

- **Atomicity**: ensures that all operations within the work unit are completed successfully. Otherwise, the transaction is aborted at the point of failure and all the previous operations are rolled back to their former state.
- **Consistency:** ensures that the database properly changes states upon a successfully committed transaction.
- **Isolation:** enables transactions to operate independently of and transparent to each other.
- **Durability:** ensures that the result or effect of a committed transaction persists in case of a system failure.

Transaction Control

The following commands are used to control transactions.

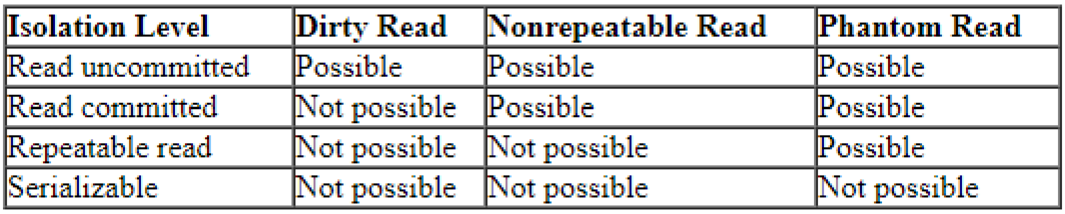
- **Commit:** Any successful execution of the transaction is ended by a COMMIT command.
- **Rollback:** Unsuccessful execution need to be ended by a ROLLBACK command which automatically recovers from the database all changes made by the transaction.
- **Savepoint:** savepoint command is used to temporarily save a transaction so that you can roll back to that point whenever required.
  Savepoint savepoint_name;
- **SET TRANSACTION:** Places a name on a transaction.

Transaction Isolation Levels

As we know that, in order to maintain consistency in a database, it follows ACID properties. Among these four properties (Atomicity, Consistency, Isolation and Durability) Isolation determines how transaction integrity is visible to other users and systems.

Isolation levels define the degree to which a transaction must be isolated from the data modifications made by any other transaction in the database system. A transaction isolation level is defined by the following phenomena.

- **Dirty Read:** A Dirty read is the situation when a transaction reads a data that has not yet been committed. For example, Let's say transaction 1 updates a row and leaves it uncommitted, meanwhile, Transaction 2 reads the updated row. If transaction 1 rolls back the change, transaction 2 will have read data that is considered never to have existed.
- **Non-Repeatable read:** Non Repeatable read occurs when a transaction reads same row twice, and get a different value each time. For example, suppose transaction T1 reads data. Due to concurrency, another transaction T2 updates the same data and commit, Now if transaction T1 rereads the same data, it will retrieve a different value.
- **Phantom Read:** Phantom Read occurs when two same queries are executed, but the rows retrieved by the two, are different. For example, suppose transaction T1 retrieves a set of rows that satisfy some search criteria. Now, Transaction T2 generates some new rows that match the search criteria for transaction T1. If transaction T1 re-executes the statement that reads the rows, it gets a different set of rows this time.



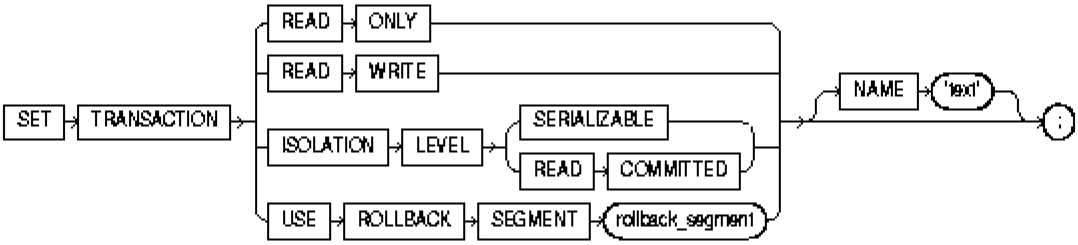
Isolation Level	Dirty Read	Nonrepeatable Read	Phantom Read
Read uncommitted	Possible	Possible	Possible
Read committed	Not possible	Possible	Possible
Repeatable read	Not possible	Not possible	Possible
Serializable	Not possible	Not possible	Not possible

The Oracle database uses the read committed and serializable isolation levels in addition to a read-only mode. By default, Oracle uses the read committed isolation level for database transactions.

Thus, according to the information in the table, dirty reads are only possible at the read uncommitted isolation level; however, the serializable isolation level is sufficient to prevent dirty reads as well as non-repeatable and phantom reads.

SET TRANSACTION Statement

The **SET TRANSACTION** statement begins a read-only or read-write transaction, establishes an isolation level, or assigns the current transaction to a specified rollback segment. Read-only transactions are useful for running multiple queries against one or more tables while other users update the same tables.



READ ONLY

Establishes the current transaction as read-only, so that subsequent queries see only changes committed before the transaction began. The use of READ ONLY does not affect other users or transactions.

READ WRITE

Establishes the current transaction as read-write. The use of READ WRITE does not affect other users or transactions. If the transaction executes a data manipulation statement, Oracle assigns the transaction to a rollback segment.

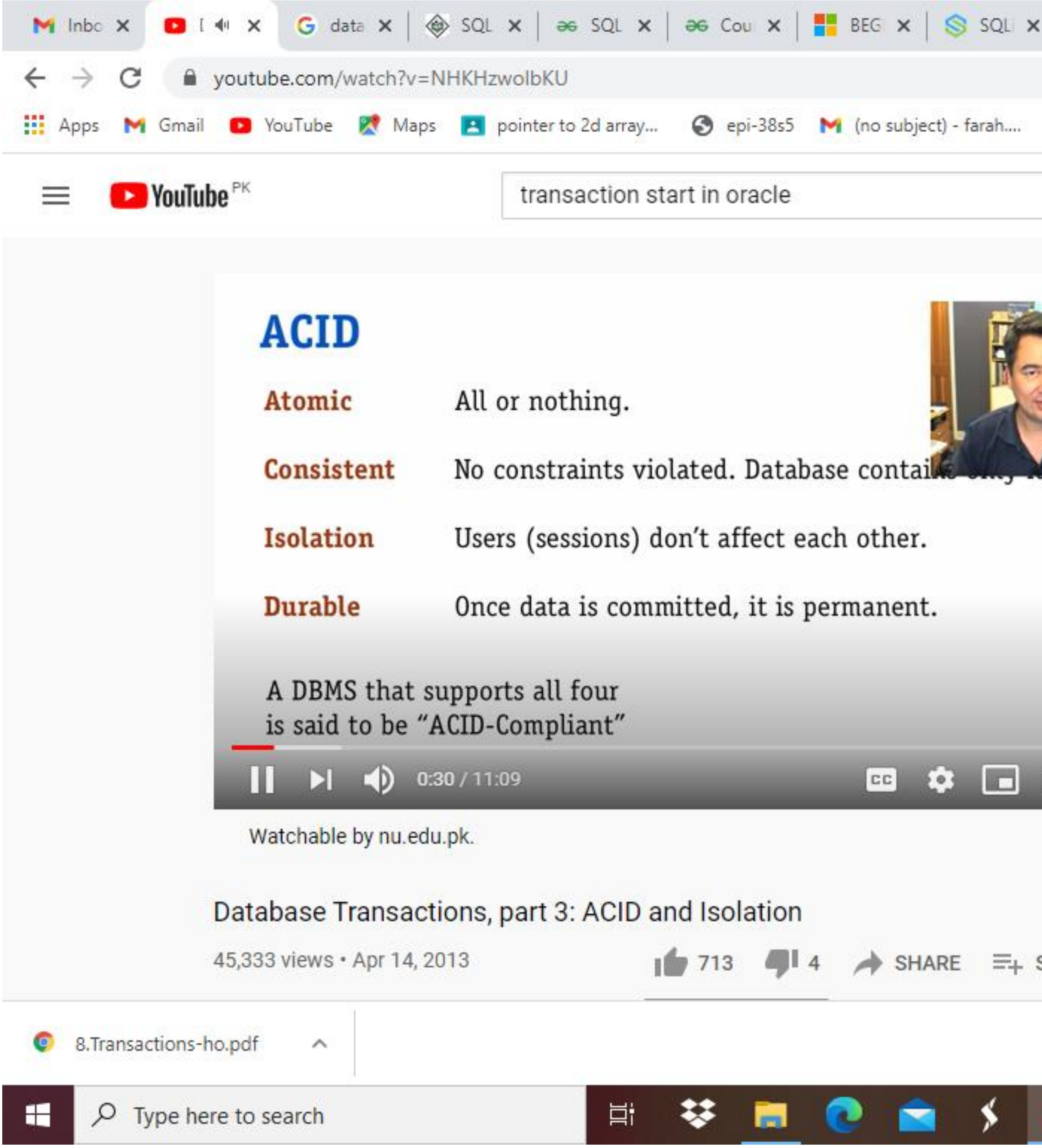
ISOLATION LEVEL

Specifies how to handle transactions that modify the database.

- **SERIALIZABLE:** If a serializable transaction tries to execute a SQL data manipulation statement that modifies any table already modified by an uncommitted transaction, the statement fails.
- **READ COMMITTED:** If a transaction includes SQL data manipulation statements that require row locks held by another transaction, the statement waits until the row locks are released.

USE ROLLBACK SEGMENT

Assigns the current transaction to the specified rollback segment and establishes the transaction as read-write. You cannot use this parameter with the READ ONLY parameter in the same transaction because read-only transactions do not generate rollback information.



ACID

Atomic	All or nothing.
Consistent	No constraints violated. Database contai
Isolation	Users (sessions) don't affect each other.
Durable	Once data is committed, it is permanent.

A DBMS that supports all four is said to be "ACID-Compliant"

Watchable by nu.edu.pk.

Database Transactions, part 3: ACID and Isolation

45,333 views • Apr 14, 2013