



## Save Points – Isolation Levels

### Savepoint in SQL

- Savepoint is a command in SQL that is used with the rollback command.
- It is a command in Transaction Control Language that is used to mark the transaction in a table.
- Consider you are making a very long table, and you want to roll back only to a certain position in a table then; this can be achieved using the savepoint.

To create a table in the database, first, we need to select the database in which we want to create a table.

1. `mysql> USE dbs;`

Then we will write a query to create a table named student in the selected database 'dbs'.

1. `mysql> CREATE TABLE student(ID INT, Name VARCHAR(20), Percentage INT, Location VARCHAR(20), DateOfBirth DATE);`

Now, we will write a single query to insert multiple records in the student table:

1. `mysql> INSERT INTO student(ID, Name, Percentage, Location, DateOfBirth) VALUES(1, "Manthan Koli", 79, "Delhi", "2003-08-20"), (2, "Dev Dixit", 75, "Pune", "1999-06-17"), (3, "Aakash Deshmukh", 87, "Mumbai", "1997-09-12"), (4, "Aaryan Jaiswal", 90, "Chennai", "2005-10-02"), (5, "Rahul Khanna", 92, "Ambala", "1996-03-04"), (6, "Pankaj Deshmukh", 67, "Kanpur", "2000-02-02"), (7, "Gaurav Kumar", 84, "Chandigarh", "1998-07-06"), (8, "Sanket Jain", 61, "Shimla", "1990-09-08"), (9, "Sahil Wagh", 90, "Kolkata", "1968-04-03");`
1. `mysql> SELECT *FROM student;`

ID	Name	Percentage	Location	DateOfBirth
1	Manthan Koli	79	Delhi	2003-08-20
2	Dev Dixit	75	Pune	1999-06-17



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

3	Aakash Deshmukh	87	Mumbai	1997-09-12
4	Aaryan Jaiswal	90	Chennai	2005-10-02
5	Rahul Khanna	92	Ambala	1996-03-04
6	Pankaj Deshmukh	67	Kanpur	2000-02-02
7	Gaurav Kumar	84	Chandigarh	1998-07-06
8	Sanket Jain	61	Shimla	1990-09-08
9	Sahil Wagh	90	Kolkata	1968-04-03

The results show that all ten records are inserted successfully.

To use the TCL commands in SQL, we first need to initiate the transaction by using the BEGIN / START TRANSACTION command.

1. mysql> START **TRANSACTION**;

```
mysql> START TRANSACTION;  
Query OK, 0 rows affected (0.00 sec)
```

We will save our initiated transaction using the SAVEPOINT command along with some specific names of this savepoint.

1. mysql> SAVEPOINT ini;

Here, we have saved the initiated transaction with the name of 'ini'.

```
mysql> SAVEPOINT ini;  
Query OK, 0 rows affected (0.05 sec)
```

1. **Isolation:**

Isolation is referred to as a state of separation. A DBMS's isolation feature ensures that several transactions can take place simultaneously and that no data from one database should have an impact on another. In other words, the process on the second state of the database will start after the operation on the first state is finished.

**Phenomena Defining Isolation Level:**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

- A transaction that reads data that hasn't yet been committed is said to have performed a "**Dirty Read**". Imagine that when Transaction 2 receives the modified row, Transaction 1 modifies the row and leaves it uncommitted. Transaction 2 will have read data that was never intended to exist if transaction 1 reverses the change.
- **Non Repeatable Read** occurs when a transaction reads the same row twice and receives a different value each time. Assume that transaction T1 reads data. Because of concurrency, another transaction, T2, modifies and commits the same data. Transaction T1 will get a different value if it reads the same data a second time.
- When two identical queries are run, but the rows returned by the two are different, this phenomenon is known as a "**Phantom Read**." Assume transaction T1 receives a collection of records that meet some search criteria. Transaction T2 now creates some new data that fit the transaction T1 search criteria. Transaction T1 will acquire a different set of rows if it re-executes the statement that reads the rows.

The SQL standard defines four isolation levels based on these phenomena:

**Levels of Isolation:**

Isolation is divided into four stages. The ability of users to access the same data concurrently is constrained by higher isolation. The greater the isolation degree, the more system resources are required, and the greater the likelihood that database transactions would block one another.

- "**Serializable**," the highest level, denotes that one transaction must be completed before another can start.
- **Repeatable Reads** allow transactions to be accessed after they have begun, even if they have not completed. This level enables phantom reads or the awareness of inserted or deleted rows even when changes to existing rows are not readable.
- **Read Committed** allows you access to information only after it has been committed to the database.
- **Read Uncommitted** is the lowest level of isolation, allowing access to data before modifications are performed.

Users are more prone to experience read phenomena like uncommitted dependencies, often known as dirty reads, where data is read from a row that has been modified by another user but has not yet been committed to the database, and the lower the isolation level