



SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with ‘A’ Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING(IoT and
Cybersecurity Including BCT)**

COURSE NAME : 19SB504 DATABASE MANAGEMENT SYSTEMS

III YEAR / V SEMESTER

Unit II- SQL

**Topic :Transaction Control Commands – Commit,
Rollback, Save point**



Transaction Control Command

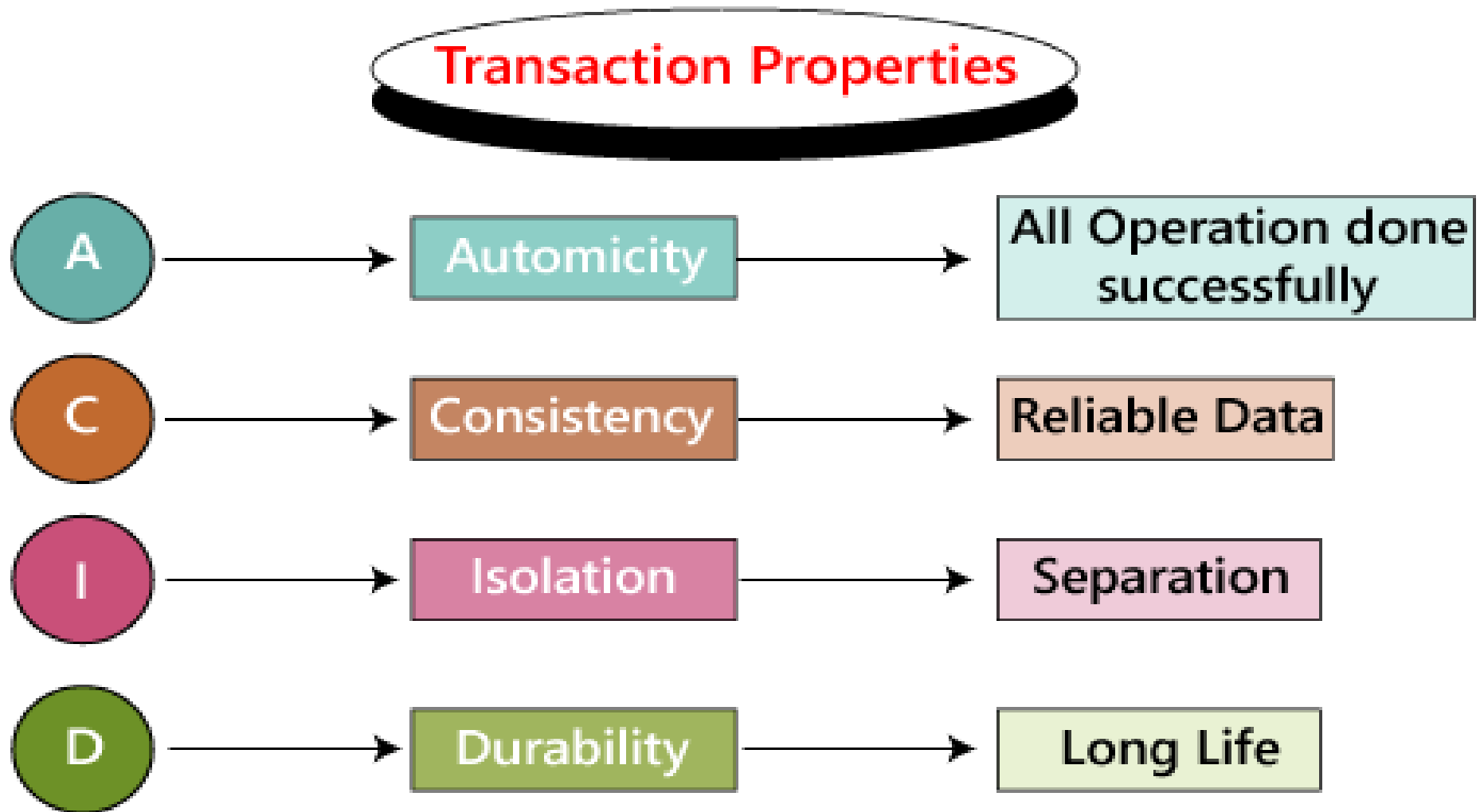
The **Transaction** is the unit of work performed opposite of the database. Any transaction that reads from or writes to a database.

Transaction is the **spread of one or several changes to a database.**

For example, if we are **recording, updating, or deleting** the history of the table, then we create the transaction of the table.

To control the operations of the **data integrity is essential**, and it is used to handle the error of the database. We add some SQL queries to the group and execute the transaction part.

Properties of Transactions





- **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** – to save the changes.
- **ROLLBACK** – to roll back the changes.
- **SAVEPOINT** – creates points within the groups of transactions in which to ROLLBACK.
- **SET TRANSACTION** – Places a name on a transaction.



COMMIT Command



The COMMIT command is the transactional command **used to save changes invoked by a transaction to the database.**

The COMMIT command is the transactional command used to save changes invoked by a transaction to the database. The COMMIT command saves all the transactions to the database since the last COMMIT or ROLLBACK command.

Syntax

COMMIT;

Example

Consider the CUSTOMERS table having the following records –



ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

Following is an example which would delete those records from the table which have age = 25 and then COMMIT the changes in the database.

```
SQL> DELETE FROM CUSTOMERS WHERE AGE = 25;  
SQL> COMMIT;
```



Output

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
3	kaushik	23	Kota	2000.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00



ROLLBACK Command

The ROLLBACK command is the transactional command **used to undo** transactions that have not already been saved to the database. This command can only be used to undo transactions since the last COMMIT or ROLLBACK command was issued.

Syntax

ROLLBACK;

Example

Consider the CUSTOMERS table having the following records –

```
SQL> DELETE FROM CUSTOMERS WHERE AGE = 25;
```

```
SQL> ROLLBACK;
```



Output

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00



SAVEPOINT Command(Book Marks)



A SAVEPOINT is a point in a transaction when you can roll the transaction back to a certain point without rolling back the entire transaction.

Syntax

```
SAVEPOINT SAVEPOINT_NAME;
```

This command serves only in the creation of a SAVEPOINT among all the transactional statements. The ROLLBACK command is used to undo a group of transactions.

The syntax for rolling back to a SAVEPOINT is as shown below.

```
ROLLBACK TO SAVEPOINT_NAME;
```



```
SQL> SAVEPOINT SP1;
```

Savepoint created.

```
SQL> DELETE FROM CUSTOMERS WHERE ID=1;
```

1 row deleted.

```
SQL> SAVEPOINT SP2;
```

Savepoint created.

```
SQL> DELETE FROM CUSTOMERS WHERE ID=2;
```

1 row deleted.

```
SQL> SAVEPOINT SP3;
```

Savepoint created.

```
SQL> DELETE FROM CUSTOMERS WHERE ID=3;
```

1 row deleted.



Now that the three deletions have taken place, let us assume that you have changed your mind and decided to **ROLLBACK** to the **SAVEPOINT** that you identified as **SP2**.

Because **SP2** was created after the first deletion, the last two deletions are undone –

```
SQL> ROLLBACK TO SP2;  
Rollback complete.
```

Output



```
SQL> SELECT * FROM CUSTOMERS;
```

```
+-----+-----+-----+-----+-----+
| ID | NAME   | AGE | ADDRESS | SALARY |
+-----+-----+-----+-----+-----+
| 2 | Khilan | 25 | Delhi   | 1500.00 |
| 3 | kaushik | 23 | Kota    | 2000.00 |
| 4 | Chaitali | 25 | Mumbai | 6500.00 |
| 5 | Hardik | 27 | Bhopal  | 8500.00 |
| 6 | Komal | 22 | MP      | 4500.00 |
| 7 | Muffy | 24 | Indore  | 10000.00 |
+-----+-----+-----+-----+-----+
```

6 rows selected.



Any Query????

Thank you.....