KARNAVATI UNIVERSITY | UIT

Unitedworld Institute Of Technology

B.Tech. Computer Science & Engineering

Semester : 3rd

Introduction To Database Management System

Course Code: 71203002003

Unit – 3 : STRUCTURED QUERY LANGUAGE - SQL AND PL/SQL

Prepared by:
Mr. Utsav Kapadiya
Assistant Professor (UIT)

Transaction Control Commands in SQL

What is a Transaction?

A Transaction is a group of SQL statements executed as a single unit. It ensures data consistency — meaning either all operations succeed or none do.
This prevents partial updates that could corrupt your data. If one operation fails, the entire transaction can be rolled back to maintain database integrity.

Example Transaction

```
BEGIN;UPDATE accounts SET balance = balance - 500 WHERE id = 1;UPDATE accounts SET balance = balance + 500 WHERE id = 2;COMMIT;
```

Transaction Control Commands (TCC)

There are 3 main commands used to manage transactions effectively:

COMMIT

Save all changes permanently to the database.

ROLLBACK

Undo (cancel) all changes made since the last COMMIT or SAVEPOINT.

SAVEPOINT

Set a temporary marker in a transaction to which you can roll back later.

1. COMMIT

Definition

COMMIT permanently saves all the changes made during the transaction to the database.

Syntax

```
COMMIT;
```

After COMMIT, all changes are permanent and cannot be undone.

Example

```
BEGIN;UPDATE Students SET department_id = 102 WHERE student_id = 1;INSERT INTO Students VALUES (4, 'Priya', 103);COMMIT;
```

2. ROLLBACK

Definition

ROLLBACK cancels all changes made in the current transaction and restores the database to the last committed state.

Syntax

```
ROLLBACK;
```

After ROLLBACK, changes are undone — as if they never happened.

Example

```
BEGIN;UPDATE Students SET department_id = 105 WHERE student_id = 2;DELETE FROM Students WHERE student_id = 3;-- Oops! Wrong data deletedROLLBACK;
```

Both the UPDATE and DELETE are completely undone.

3. SAVEPOINT

SAVEPOINT sets a checkpoint inside a transaction. You can roll back to that specific point without undoing the entire transaction.

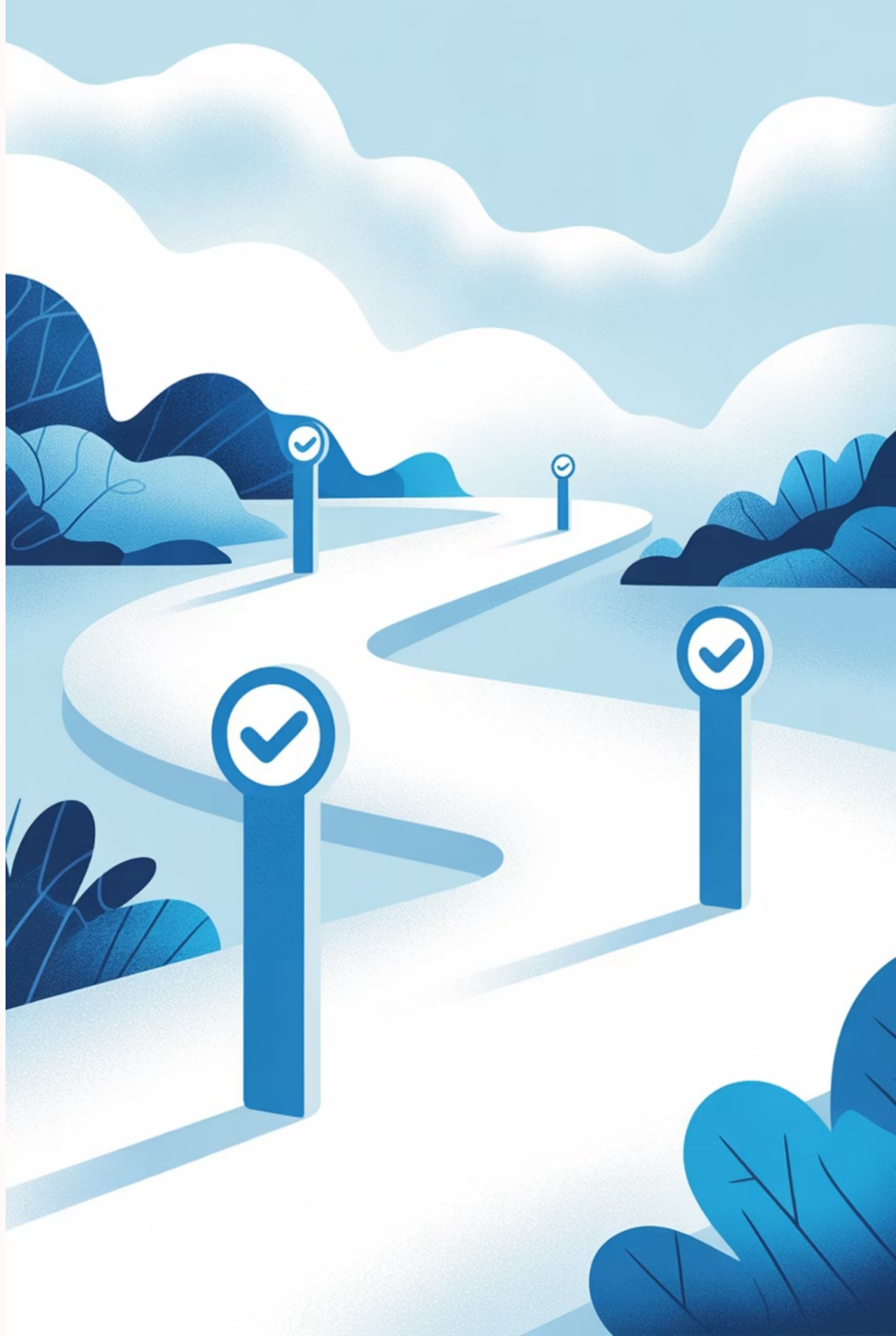
Syntax

```
SAVEPOINT savepoint_name;ROLLBACK TO savepoint_name;
```

Example

```
BEGIN;UPDATE Students SET department_id = 102 WHERE student_id = 1;SAVEPOINT A;UPDATE Students SET department_id = 103 WHERE student_id = 2;SAVEPOINT B;DELETE FROM Students WHERE student_id = 3;-- Only undo the deleteROLLBACK TO B;COMMIT;
```

Changes before SAVEPOINT B are kept. The DELETE is undone. Then COMMIT saves everything else.



Summary Table

COMMIT	Ends a transaction and saves changes permanently	Changes cannot be undone
ROLLBACK	Cancels all uncommitted changes	Restores database to last COMMIT
SAVEPOINT	Creates a temporary rollback point	Allows partial rollback within transaction

Analogy: Easy to Remember

Imagine you're editing a document:

SAVE (COMMIT)

You save your work permanently to disk.

UNDO (ROLLBACK)

You undo everything since your last save.

BOOKMARK (SAVEPOINT)

You mark a place, so you can undo changes only after that point.

ACID Properties in SQL

ACID ensures database transactions are reliable and consistent.

1. Atomicity – All or Nothing

Definition

A transaction is treated as one single unit — either every step succeeds or none do.

Real-life analogy

When sending a parcel, if you can't hand over the parcel and collect the payment together, the deal is canceled — not half done.

Example code

```
BEGIN;UPDATE accounts SET
balance = balance - 500 WHERE
id = 1;UPDATE accounts SET
balance = balance + 500 WHERE
id = 2;COMMIT;
```

If the second UPDATE fails, the first one will also be rolled back automatically.

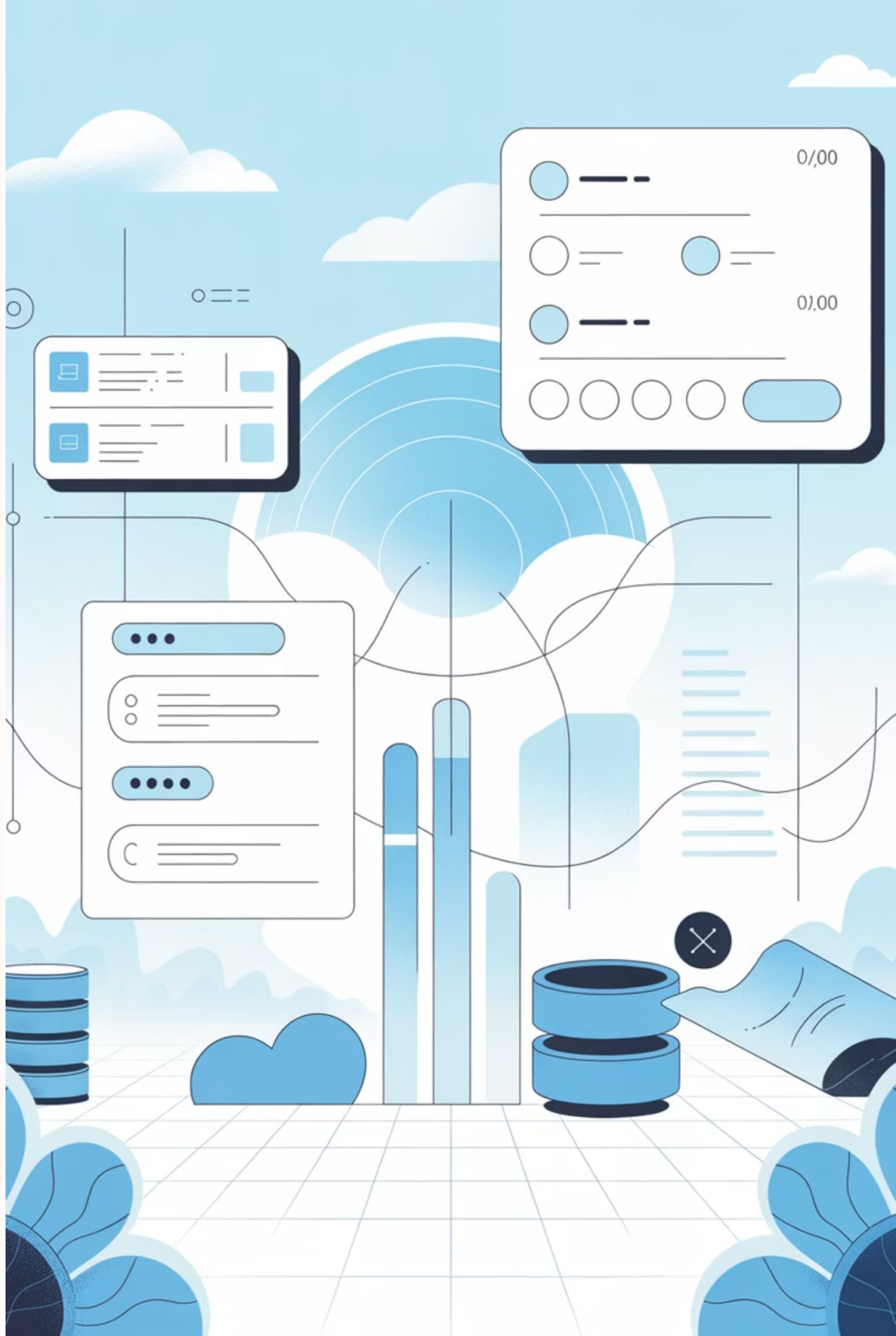
2. Consistency – Database Stays Valid

A transaction must keep the database in a valid state , means data rules, constraints, and relationships remain correct.

Example: If Account A had ₹2000 and Account B had ₹1000, after transferring ₹500, total money = ₹3000 should still remain.

The transaction must not violate constraints like:

- Foreign key relationships
- Data types
- Check constraints
- Balance totals



3. Isolation – Transactions Don't Disturb Each Other

Definition: Each transaction works independently. Even if 10 users are updating the database at once, it should feel like each one is running alone.

Example: If two users try to buy the last ticket at the same time:

- Isolation ensures only one of them gets it
- The other will see the updated data after the first transaction commits

Prevents problems like:

- Dirty reads
- Lost updates
- Inconsistent data

4. Durability – Once Saved, Always Saved

Definition: After a transaction is committed, the changes are permanent, even if there's a power failure or crash.

Example: If you COMMIT a transaction that adds ₹500 to your account, even if the database server restarts, your ₹500 will still be there.

Databases achieve durability using:

- Transaction logs
- Backups
- Write-ahead logging mechanisms

ACID Properties Summary

ATOMICITY

All or Nothing

CONSISTENCY

Database Stays Valid

ISOLATION

Transactions Don't Disturb Each Other

DURABILITY

Once Saved, Always Saved

ACID properties ensure reliable database transactions.

Wrap-up:

Real-Life Example

Bank money transfers use COMMIT, ROLLBACK, and SAVEPOINT to ensure both debit and credit operations succeed together — or neither happens at all.

Key Takeaway

Transaction control commands are essential for maintaining data integrity and consistency in any database application.

Practice these commands in phpMyAdmin or your SQL terminal to master transaction control and ace your interviews!

Thank You!