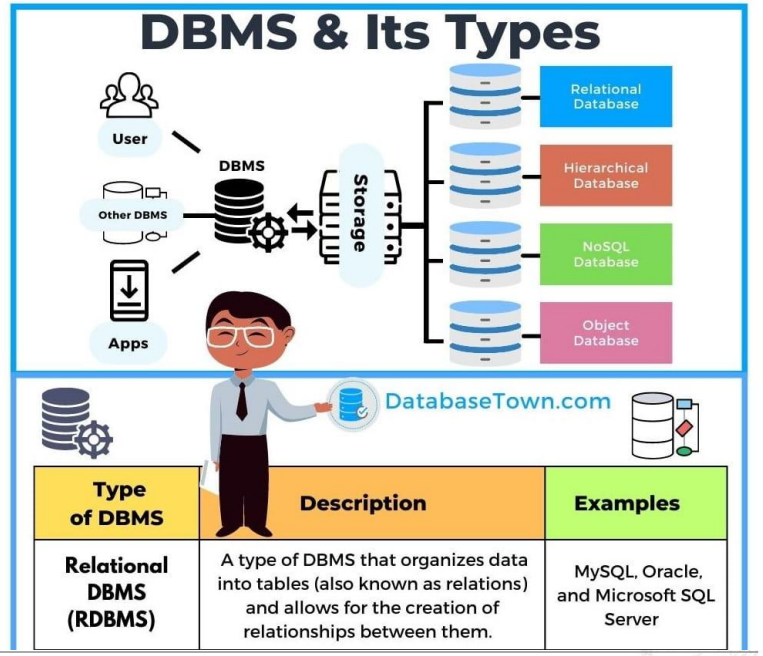
Introduction to Databases, DBMS, SQL, and More



NIELIT Chandigarh/Ropar

"A SQL query walks into a bar. He approaches two tables and says, "Mind if I join you?""

A structured collection of data stored electronically.

- **Definition:** A structured collection of data stored electronically.
- **Examples:**
  - School student records
  - Hospital patient data
  - Online shopping product details
- **A simple database table:**

employees

id	name	job_level_id
54378	Darius	3
94722	Raven	3
45783	Eduardo	1

What is a DBMS? (Database Management System)

- **Definition:** Software that manages databases (stores, retrieves, updates).
- **Examples:**
  - MySQL
  - Oracle
  - Microsoft SQL Server
  - PostgreSQL

Why Normalization?

- **Definition:** Organizing data to reduce redundancy and improve efficiency.
- **Normal Forms (1NF, 2NF, 3NF):**
  - **1NF:** No repeating groups
  - **2NF:** No partial dependencies
  - **3NF:** No transitive dependencies

Not-Normalized Database Table

STUDENT

StudentID	StudentName	MajorName	NoOfCreditHours
111	Kirsten	Accounting	152
222	Eve	IS	138
333	Zoe	IS	138
444	Ben	Accounting	152

Normalized Database Tables

MAJOR

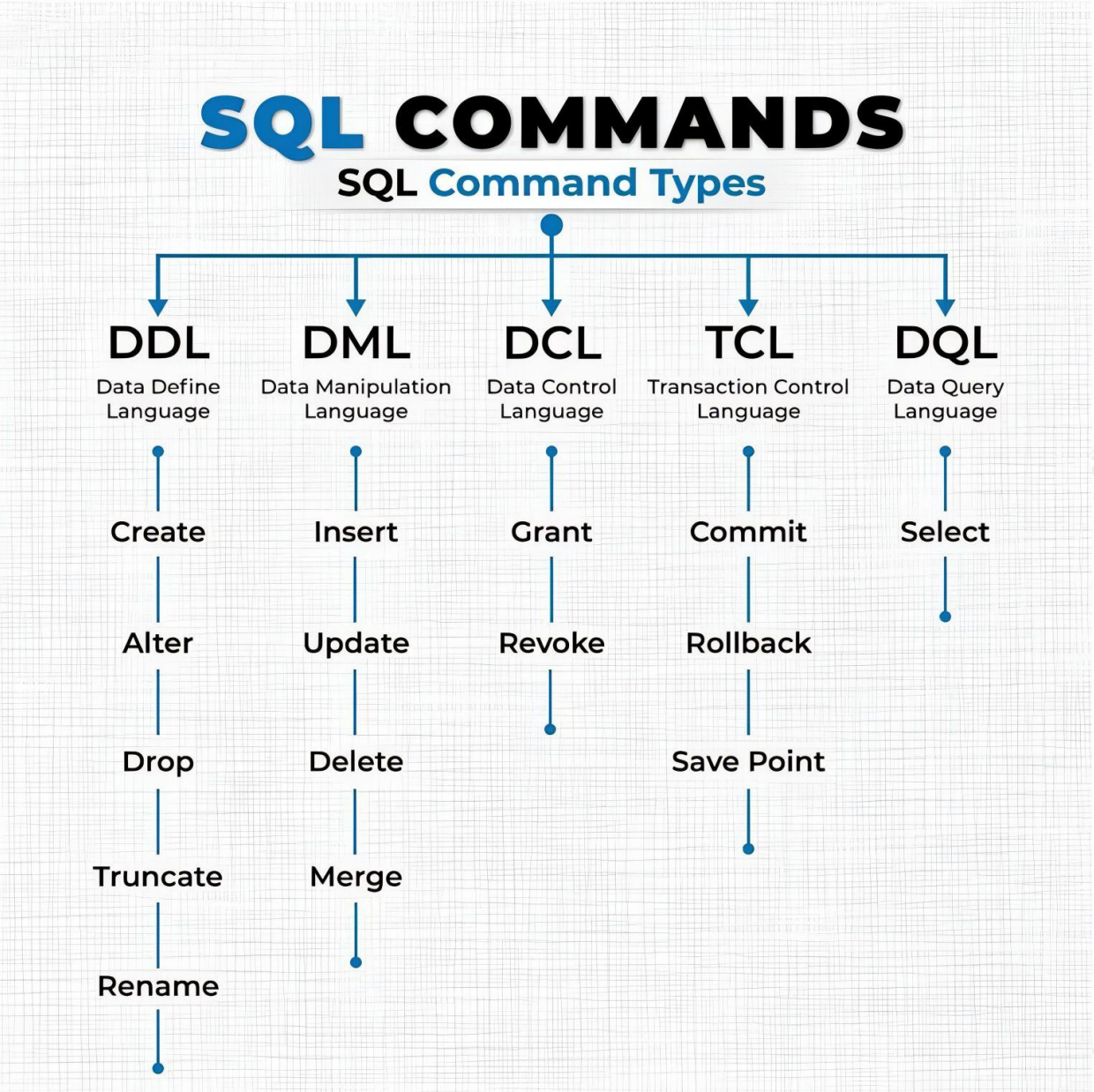
MajorName	NoOfCreditHours
Accounting	152
IS	138

STUDENT

StudentID	StudentName	MajorName
111	Kirsten	Accounting
222	Eve	IS
333	Zoe	IS
444	Ben	Accounting

Introduction to SQL

- **Definition:** Structured Query Language – used to communicate with databases.
- **Types of SQL Commands:**
  - DDL (Data Definition Language)
  - DML (Data Manipulation Language)
  - DCL (Data Control Language)
  - TCL (Transaction Control Language)



Types of SQL Commands

Type	Full Form	Key Commands	Purpose
DDL	Data Definition Lang.	CREATE, ALTER, DROP, TRUNCATE	Define/modify database structure
DML	Data Manipulation Lang.	SELECT, INSERT, UPDATE, DELETE	Manage data in tables
DCL	Data Control Lang.	GRANT, REVOKE	Control access permissions
TCL	Transaction Control Lang.	COMMIT, ROLLBACK, SAVEPOINT	Manage transactions

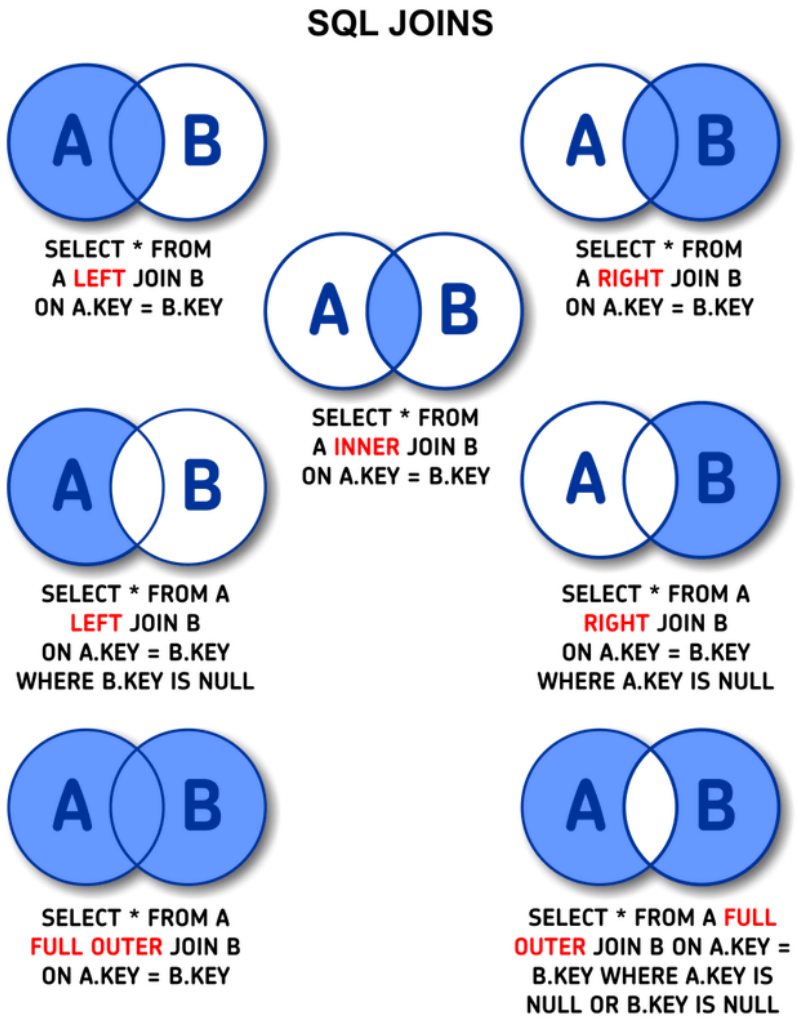
Basic SQL Commands

Command	Description	Example
CREATE	Creates a database/table	CREATE TABLE Students (...);
SELECT	Retrieves data	SELECT * FROM Students;
INSERT	Adds new data	INSERT INTO Students VALUES (...);
UPDATE	Modifies data	UPDATE Students SET name='John' WHERE id=1;
DELETE	Removes data	DELETE FROM Students WHERE id=2;

Joins in SQL

- Combines data from multiple tables.
- **Types:**
  - **INNER JOIN:** Returns matching rows
  - **LEFT JOIN:** All rows from left table + matches from right
  - **RIGHT JOIN:** All rows from right table + matches from left
  - **FULL JOIN:** All rows when there's a match in either table
- **Example:**

```
SELECT Orders.OrderID, Customers.CustomerName
FROM Orders INNER JOIN Customers ON
Orders.CustomerID = Customers.CustomerID;
```



Constraints in SQL

Rules applied to table columns for data integrity.

- **Common Constraints:**
  - PRIMARY KEY – Uniquely identifies a row
  - FOREIGN KEY – Links two tables
  - NOT NULL – Column cannot be empty
  - UNIQUE – All values must be different
  - CHECK – Ensures condition is met
  - DEFAULT – Sets a default value

- **Example:**

```
CREATE TABLE Employees (
    ID int PRIMARY KEY,
    Name varchar(50) NOT NULL,
    Age int CHECK (Age >= 18)
);
```

Summary

- **Database** → Organized data storage
- **DBMS** → Manages databases
- **Normalization** → Reduces redundancy
- **SQL** → Language for database operations
- **Joins** → Combine tables
- **Constraints** → Ensure data integrity