MySQL Notes

(From Basics to Intermediate Level)

◆ What is MySQL?

- **MySQL** is a **Relational Database Management System (RDBMS)**.

- It uses **SQL (Structured Query Language)** to manage databases.

- Open-source and widely used with web technologies like PHP.

🧩 Basic Concepts

Term	Definition
Database	Collection of structured data.
Table	Stores data in rows and columns.
Row (Record)	A single entry in a table.
Column (Field)	A specific category of data in a table.
Query	A command written in SQL to perform operations.

🧾 MySQL Data Types

Type	Example	Use
INT	10, 200	Integer values
VARCHAR(n)	'Dhanush'	Text up to n characters
DATE	'2025-07-26'	Stores date

BOOLEAN	TRUE / FALSE	Logical values
FLOAT	3.14	Decimal numbers
TEXT	Large text	Articles, descriptions

🛠️ Common SQL Commands

🧱 DDL – Data Definition Language

sql
CopyEdit
```sql
CREATE TABLE Students (
  ID INT,
  Name VARCHAR(100),
  Age INT
);

ALTER TABLE Students ADD Email VARCHAR(100);

DROP TABLE Students;
```

✍️ DML – Data Manipulation Language

sql
CopyEdit
```sql
INSERT INTO Students VALUES (1, 'Dhanush', 21);
UPDATE Students SET Age = 22 WHERE ID = 1;
DELETE FROM Students WHERE ID = 1;
```

🔍 DQL – Data Query Language

sql
CopyEdit
```sql
SELECT * FROM Students;
SELECT Name, Age FROM Students WHERE Age > 20;
```

🔐 DCL – Data Control Language

```sql
CopyEdit
GRANT SELECT ON Students TO 'user1';
REVOKE SELECT ON Students FROM 'user1';
```

🔁 TCL – Transaction Control Language

```sql
CopyEdit
START TRANSACTION;
UPDATE Students SET Age = 23 WHERE ID = 1;
COMMIT;
-- or
ROLLBACK;
```

📊 Constraints in MySQL

Constraint	Description
PRIMARY KEY	Uniquely identifies a row
FOREIGN KEY	Refers to primary key in another table
NOT NULL	Field cannot be empty
UNIQUE	No duplicate values
DEFAULT	Sets a default value

Example:

```sql
CopyEdit
CREATE TABLE Users (
  ID INT PRIMARY KEY,
  Name VARCHAR(100) NOT NULL,
```

```
  Age INT DEFAULT 18
);
```

🔗 Joins in MySQL

Type	Description
INNER JOIN	Returns matching records from both
LEFT JOIN	All records from left + matching right
RIGHT JOIN	All records from right + matching left
FULL JOIN*	All records from both sides *(not in MySQL directly)*

Example:

sql
CopyEdit
```sql
SELECT A.Name, B.Course
FROM Students A
INNER JOIN Courses B ON A.ID = B.StudentID;
```

🧠 Aggregate Functions

sql
CopyEdit
```sql
SELECT COUNT(*) FROM Students;
SELECT AVG(Age) FROM Students;
SELECT MAX(Age), MIN(Age) FROM Students;
```

🧰 Useful Clauses

Clause	Use
WHERE	Filters rows

`ORDER BY`	Sorts results
`GROUP BY`	Groups rows
`HAVING`	Filters groups
`LIMIT`	Limits number of results

💡 Sample Project Table

sql
CopyEdit

```sql
CREATE TABLE Products (
    ID INT PRIMARY KEY,
    Name VARCHAR(100),
    Price FLOAT,
    InStock BOOLEAN
);

INSERT INTO Products VALUES (1, 'Laptop', 54999.99, TRUE);
```

✅ MySQL Advantages

- Free and open-source

- Easy integration with PHP and Java

- Fast and secure

- Cross-platform support

📘 MySQL Notes – Part 2 (Advanced Topics)

🔹 1. Aliases (AS)

Used to give a **temporary name** to a table or column.

```sql
CopyEdit
SELECT Name AS StudentName FROM Students;
SELECT A.Name, B.Course FROM Students AS A JOIN Courses AS B ON A.ID = B.StudentID;
```

🔹 2. Pattern Matching with LIKE

```sql
CopyEdit
SELECT * FROM Students WHERE Name LIKE 'A%';   -- Starts with A
SELECT * FROM Students WHERE Name LIKE '%sh';  -- Ends with sh
SELECT * FROM Students WHERE Name LIKE '%an%'; -- Contains "an"
```

🔹 3. IN, BETWEEN, IS NULL

```sql
CopyEdit
SELECT * FROM Students WHERE Age IN (18, 20, 22);
SELECT * FROM Students WHERE Age BETWEEN 18 AND 25;
SELECT * FROM Students WHERE Email IS NULL;
```

🔹 4. Indexes

Indexes speed up data retrieval from tables.

```sql
CopyEdit
CREATE INDEX idx_name ON Students(Name);
DROP INDEX idx_name ON Students;
```

◆ 5. Views

A view is a **virtual table** based on the result of a query.

```sql
CopyEdit
CREATE VIEW StudentNames AS
SELECT Name, Age FROM Students;

SELECT * FROM StudentNames;
```

◆ 6. Stored Procedures

Reusable blocks of SQL code.

```sql
CopyEdit
DELIMITER //

CREATE PROCEDURE GetAllStudents()
BEGIN
  SELECT * FROM Students;
END //

DELIMITER ;

CALL GetAllStudents();
```

◆ 7. Triggers

Triggers are SQL code that runs **automatically** when an event occurs (INSERT, UPDATE, DELETE).

```sql
CopyEdit
CREATE TRIGGER before_insert_students
BEFORE INSERT ON Students
FOR EACH ROW
SET NEW.Age = IF(NEW.Age IS NULL, 18, NEW.Age);
```

◆ 8. Subqueries

A query inside another query.

```sql
CopyEdit
SELECT Name FROM Students
WHERE Age = (SELECT MAX(Age) FROM Students);
```

◆ 9. Transactions

Transactions are used to execute **multiple SQL statements** as a single unit.

```sql
CopyEdit
START TRANSACTION;
UPDATE Account SET Balance = Balance - 500 WHERE ID = 1;
UPDATE Account SET Balance = Balance + 500 WHERE ID = 2;
COMMIT;
```

Use ROLLBACK to undo.

🔹 10. User Management

Create and manage database users.

```sql
CopyEdit
CREATE USER 'dhanush'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON mydb.* TO 'dhanush'@'localhost';
FLUSH PRIVILEGES;
```

🔹 11. Backup and Restore

- Backup:

```bash
CopyEdit
mysqldump -u root -p mydb > mydb_backup.sql
```

- Restore:

```bash
CopyEdit
mysql -u root -p mydb < mydb_backup.sql
```

🧠 Practice Queries

```sql
CopyEdit
-- Top 3 oldest students
SELECT Name, Age FROM Students ORDER BY Age DESC LIMIT 3;

-- Count students in each course
SELECT Course, COUNT(*) FROM Students GROUP BY Course;

-- Students not in course 'BCA'
```

```sql
SELECT * FROM Students WHERE Course <> 'BCA';
```

🧾 Project Suggestions

1. **Student Management System**

2. **E-commerce Product Catalog**

3. **Library Management System**

4. **Online Quiz Database**

5. **Hospital Appointment System**