--------*LECTURE- 12*--------

1️⃣ Handling Duplicates using GROUP BY / HAVING

What are duplicates?

Duplicates are **rows where certain column values repeat** (e.g., same email, same phone, same order_id).

How GROUP BY helps

GROUP BY groups rows with the same values.

HAVING filters groups **after grouping**.

Why HAVING, not WHERE?

- WHERE filters **rows**
- HAVING filters **groups**

Example idea

If you want to find duplicate emails:

- Group by email
- Count how many times each email appears
- Keep only those with count > 1

👉 Used widely in **data cleanup and validation**.

2️⃣ JOIN-based Duplicates

What does this mean?

Duplicates that occur **because of joins**, not because data is actually duplicated.

Why does it happen?

- One-to-many relationships
  (e.g., one customer → many orders)

When you join:

- Customer table + Orders table
- Customer data repeats for every order

Why this is important?

- Can inflate counts
- Can give wrong reports
- Common issue in production queries

How to handle?

- Use proper join conditions
- Use DISTINCT
- Aggregate before joining (best practice)

👉 Critical skill for **reporting and performance tuning**.

3️⃣ MySQL Characteristics

Key features of MySQL:

- Open-source RDBMS
- Supports **ACID** (with InnoDB)

- Widely used in web and enterprise apps
- Supports:
    - Indexes
    - Transactions
    - Stored procedures
    - Triggers
    - Events
- Scales well with replication and clustering

👉 Very popular for **high-traffic production systems**.

4️⃣ Built-in Functions

What are they?

Predefined functions provided by MySQL to perform operations.

Main categories:

- **String**: CONCAT, LENGTH, UPPER, LOWER, REPLACE
- **Numeric**: ROUND, CEIL, FLOOR, ABS
- **Date/Time**: NOW, CURDATE, DATEDIFF
- **Aggregate**: COUNT, SUM, AVG, MAX, MIN

Why important?

- Reduce complex logic
- Improve readability
- Used heavily in **real-time queries**

5️⃣ UDFs (User Defined Functions)

What are UDFs?

Functions **created by users**, not built-in.

Purpose:

- Reuse business logic
- Return a **single value**

Difference from stored procedures:

UDF	Stored Procedure
Returns single value	Can return multiple values
Used in SELECT	Cannot be used directly in SELECT

👉 Useful when logic needs to be reused in queries.

6️⃣ Stored Procedures

What are they?

A **set of SQL statements stored in the database** and executed as a unit.

Why use them?

- Reusability
- Better performance
- Reduced network traffic
- Security (logic hidden from users)

Features:

- Can take input/output parameters
- Can contain loops and conditions
- Used for **business logic at DB level**

☞ Common in enterprise applications.

7️⃣ Triggers

What is a trigger?

A special procedure that runs **automatically** when a specific event occurs.

Events:

- INSERT
- UPDATE
- DELETE

Use cases:

- Audit logging
- Data validation
- Auto-updating columns (e.g., timestamps)

Important:

- Triggers fire automatically
- Can impact performance if misused

☞ Powerful but must be used carefully.

8️⃣ Events

What are events?

Scheduled tasks inside MySQL (like a cron job).

Example use cases:

- Daily cleanup of old data
- Monthly report generation
- Archiving records

Difference from triggers:

Trigger	Event
Event-based	Time-based
Auto on DML	Runs on schedule

☞ Useful for **automation inside DB**.

9️⃣ ERD Basics (Entity Relationship Diagram)

What is ERD?

A visual representation of:

- Tables (Entities)
- Columns (Attributes)
- Relationships

Key components:

- Primary Key
- Foreign Key

- One-to-One
- One-to-Many
- Many-to-Many

Why ERD matters?

- Helps design databases
- Avoids redundancy
- Improves normalization

👉 Foundation of **database design**.