DEMO

WEEK 1 & WEEK 2

**DDL (Data Definition Language)**

**CREATE Table**

CREATE TABLE table\_name (

column1 INT PRIMARY KEY AUTO\_INCREMENT,

column2 VARCHAR(50) NOT NULL,

column3 DECIMAL(10,2)

);

**ALTER Table**

ALTER TABLE table\_name ADD COLUMN new\_column VARCHAR(100);

**DROP Table**

DROP TABLE table\_name;

**DML (Data Manipulation Language)**

**INSERT**

INSERT INTO table\_name (column1, column2, column3)

VALUES (value1, 'value2', value3);

**UPDATE**

UPDATE table\_name

SET column2 = 'new\_value'

WHERE column1 = value1;

**DELETE**

DELETE FROM table\_name

WHERE column1 = value1;

**SELECT with Clauses**

**SELECT**

SELECT column1, column2 FROM table\_name;

**WHERE**

SELECT \* FROM table\_name

WHERE column3 > 1000;

**ORDER BY**

SELECT column1, column3

FROM table\_name

ORDER BY column3 DESC;

**GROUP BY**

SELECT column1, AVG(column3) AS avg\_value

FROM table\_name

GROUP BY column1;

**HAVING**

SELECT column1, AVG(column3) AS avg\_value

FROM table\_name

GROUP BY column1

HAVING AVG(column3) > 5000;

**Joins**

**INNER JOIN**

SELECT t1.column1, t2.column2

FROM table1 t1

INNER JOIN table2 t2 ON t1.common\_col = t2.common\_col;

**LEFT JOIN**

SELECT t1.column1, t2.column2

FROM table1 t1

LEFT JOIN table2 t2 ON t1.common\_col = t2.common\_col;

**RIGHT JOIN**

SELECT t1.column1, t2.column2

FROM table1 t1

RIGHT JOIN table2 t2 ON t1.common\_col = t2.common\_col;

**Aggregate Functions**

SELECT SUM(column1) AS total\_value FROM table\_name;

SELECT COUNT(\*) AS total\_rows FROM table\_name;

SELECT AVG(column1) AS avg\_value FROM table\_name;

SELECT MIN(column1) AS min\_value FROM table\_name;

SELECT MAX(column1) AS max\_value FROM table\_name;

**Subqueries**

**Nested Query**

SELECT column1, column2

FROM table\_name

WHERE column3 > (SELECT AVG(column3) FROM table\_name);

**Correlated Subquery**

SELECT column1, column2

FROM table1 t1

WHERE column3 > (

SELECT AVG(column3) FROM table1 WHERE column4 = t1.column4

);

**Set Operations**

**UNION**

SELECT column1 FROM table1

UNION

SELECT column1 FROM table2;

**UNION ALL**

SELECT column1 FROM table1

UNION ALL

SELECT column1 FROM table2;

**Procedural Objects**

**Trigger**

CREATE TABLE deleted\_records (

column1 INT,

column2 VARCHAR(50),

column3 DECIMAL(10,2)

);

DELIMITER //

CREATE TRIGGER trg\_after\_delete

AFTER DELETE ON table\_name

FOR EACH ROW

BEGIN

INSERT INTO deleted\_records (column1, column2, column3)

VALUES (OLD.column1, OLD.column2, OLD.column3);

END //

DELIMITER ;

**Stored Procedure**

DELIMITER //

CREATE PROCEDURE get\_records(IN input\_val INT)

BEGIN

SELECT \* FROM table\_name WHERE column1 = input\_val;

END //

DELIMITER ;

CALL get\_records(10);

**Function**

DELIMITER //

CREATE FUNCTION calculate\_yearly(val DECIMAL(10,2))

RETURNS DECIMAL(12,2)

DETERMINISTIC

BEGIN

RETURN val \* 12;

END //

DELIMITER ;

SELECT column1, calculate\_yearly(column3) AS yearly\_value FROM table\_name;

**Views**

CREATE VIEW view\_name AS

SELECT t1.column1, t1.column2, t2.column3

FROM table1 t1

JOIN table2 t2 ON t1.common\_col = t2.common\_col;

SELECT \* FROM view\_name;

**Transactions**

**Start Transaction**

START TRANSACTION;

or

BEGIN;

**COMMIT (Save changes permanently)**

COMMIT;

**ROLLBACK (Undo changes)**

ROLLBACK;

**SAVEPOINT (Create a checkpoint inside a transaction)**

SAVEPOINT sp1;

**ROLLBACK TO SAVEPOINT**

ROLLBACK TO sp1;

**Release Savepoint**

RELEASE SAVEPOINT sp1;

**String Functions in MySQL**

**Concatenate Strings**

SELECT CONCAT(column1, ' ', column2) AS full\_name

FROM table\_name;

**Concatenate with Separator**

SELECT CONCAT\_WS('-', column1, column2, column3) AS combined

FROM table\_name;

**Length of String**

SELECT LENGTH(column1) AS str\_length

FROM table\_name;

**Convert to Uppercase / Lowercase**

SELECT UPPER(column1), LOWER(column1)

FROM table\_name;

**Substring**

SELECT SUBSTRING(column1, 1, 5) AS first\_five\_chars

FROM table\_name;

**Position of Substring**

SELECT LOCATE('sub\_str', column1) AS position

FROM table\_name;

**Replace Substring**

SELECT REPLACE(column1, 'old', 'new') AS replaced\_value

FROM table\_name;

**Trim Spaces**

SELECT TRIM(column1) AS trimmed,

LTRIM(column1) AS left\_trimmed,

RTRIM(column1) AS right\_trimmed

FROM table\_name;

**Reverse String**

SELECT REVERSE(column1) AS reversed\_str

FROM table\_name;

**Left / Right Substring**

SELECT LEFT(column1, 3) AS first\_three,

RIGHT(column1, 3) AS last\_three

FROM table\_name;

**Index Queries in MySQL**

**Create Index**

CREATE INDEX idx\_column1 ON table\_name(column1);

**Create Unique Index**

CREATE UNIQUE INDEX idx\_unique\_col ON table\_name(column2);

**Show Indexes of a Table**

SHOW INDEX FROM table\_name;

**Drop Index**

DROP INDEX idx\_column1 ON table\_name;

**Composite Index (Multiple Columns)**

CREATE INDEX idx\_multi ON table\_name(column1, column2);

**Use EXPLAIN to Check Index Usage**

EXPLAIN SELECT \* FROM table\_name WHERE column1 = 'value';