SQL COMPLETE NOTES

==========
■ DML (Data Manipulation Language)
==========
- INSERT: Add new records
- SELECT: Retrieve data
- UPDATE: Modify existing records
- DELETE: Remove records
- MERGE: Insert + Update in one

Example:
INSERT INTO Employees (emp_id, emp_name, salary, dept)
VALUES (1, 'Amit', 50000, 'HR');

SELECT * FROM Employees WHERE dept = 'IT';

UPDATE Employees SET salary = salary+5000 WHERE emp_name='Ravi';

DELETE FROM Employees WHERE dept='HR';

==========
■ Operators in SQL
==========
1. Arithmetic: +, -, *, /, %
2. Comparison: =, !=, <, >, <=, >=
3. Logical: AND, OR, NOT
4. Special: IN, BETWEEN, LIKE, IS NULL, EXISTS
5. Set Operators: UNION, UNION ALL, INTERSECT, MINUS
6. Assignment: =

Example:
SELECT * FROM Employees WHERE salary BETWEEN 40000 AND 60000;

==========
■ Joins in SQL
==========
1. INNER JOIN – Matching rows only
2. LEFT JOIN – All from left + matches from right
3. RIGHT JOIN – All from right + matches from left
4. FULL JOIN – All from both tables
5. CROSS JOIN – Cartesian product
6. SELF JOIN – Table joins with itself

Example:
SELECT e.emp_name, d.dept_name
FROM Employees e
INNER JOIN Departments d
ON e.dept_id = d.dept_id;

==========
■ Subqueries in SQL
==========
1. Single-row: returns one value

2. Multi-row: returns many values (IN, ANY, ALL)
3. Correlated: depends on outer query
4. Nested: subquery inside subquery
5. In SELECT clause
6. In FROM clause

Example:
SELECT emp_name, salary
FROM Employees
WHERE salary > (SELECT AVG(salary) FROM Employees);

==========
■ Common Table Expression (CTE)
==========
WITH cte_name AS (
SELECT ...
)
SELECT * FROM cte_name;

- Simple CTE
- Recursive CTE (hierarchies)
- Multiple CTEs

Example:
WITH AvgSalary AS (
SELECT AVG(salary) AS avg_sal FROM Employees
)
SELECT emp_name, salary FROM Employees, AvgSalary
WHERE Employees.salary > AvgSalary.avg_sal;

==========
■ SQL Clauses
==========
- WHERE → filter rows
- ORDER BY → sort rows
- GROUP BY → group rows
- HAVING → filter groups
- LIMIT/TOP → restrict rows
- DISTINCT → remove duplicates
- FROM → choose tables
- JOIN → combine tables
- IN → match list
- BETWEEN → range check
- LIKE → pattern match
- IS NULL → null check
- EXISTS → subquery check
- ALL/ANY → compare with subquery
- UNION/INTERSECT/MINUS → set operations

Example:
SELECT dept_id, AVG(salary) AS avg_salary
FROM Employees
GROUP BY dept_id
HAVING AVG(salary) > 60000;