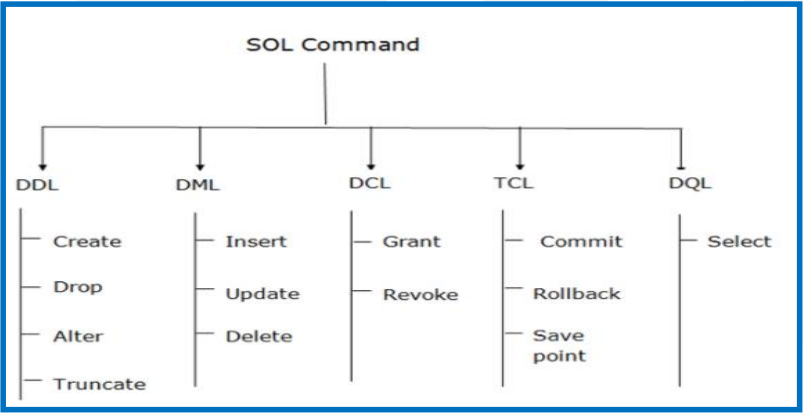
**Chapter 2: STRUCTURED QUERY LANGUAGE**

**Topic – 1: Introductory Concepts**

**Introduction**

* **ANSI:** American National Standard Institutes
* **ISO:** International Organization for Standardization

**SQL Commands**



**Topic – 2: Data Definition Language**

**Introduction**

* All changes are saved permanently.
* **VARCHAR2** is **not** any kind of keyword.
* **DROP** deletes both **structure** and **records** both.
* **ALTER** is used to modify existing/ or add new attribute.
* **TRUNCATE** deletes all rows, not the structure and frees space.

**DDL Syntaxes**

***DROP TABLE employee;***

***ALTER TABLE students ADD (ADDRESS VARCHAR(20));***

***ALTER TABLE students MODIFY (NAME VARCHAR(20));***

***TRUNCATE TABLE employee;***

**Topic – 3: Data Manipulation Language**

**Introduction**

* **Not** auto-committed, thus can be rolled back.

**DML Syntaxes**

***UPDATE students SET id = 4, name = ‘Gourav’ WHERE roll\_no = ‘1’;***

***DELETE FROM book WHERE author = ‘Gourav’;***

**Topic – 4: Data Control Language**

**DCL Syntaxes**

***GRANT SELECT, UPDATE ON students TO user1,user2;***

***REVOKE SELECT, UPDATE ON students FROM user1,user2;***

**Topic – 5: Transaction Control Language**

**Transaction Control Language**

* Used along DML commands only.
* Are auto-committed.
* Always use the word **transaction** while mentioning DCL commands.

**TCL Syntaxes**

***COMMIT;***

***ROLLBACK;***

***SAVEPOINT savepoint\_name;***

**Topic – 6: Special Operations**

**Logical Operators**

***SELECT \* FROM employee WHERE salary >= ALL(25000,60000);***

***SELECT \* FROM students WHERE job = ‘manager’ AND salary > 70000;***

***SELECT name FROM customers UNION SELECT name FROM investors;***

**Like Queries**

***SELECT \* FROM people WHERE name LIKE ‘G%’;***

***SELECT \* FROM people WHERE name LIKE ‘%v’;***

***SELECT \* FROM people WHERE name LIKE ‘\_\_\_\_\_\_%’;***

***SELECT \* FROM people WHERE name LIKE ‘\_\_u%’;***

**Aggregate Functions**

* Falls under **DML** category.

***SELECT SUM(salary) FROM employee;***

***SELECT AVG(salary) FROM employee;***

***SELECT MAX(salary) FROM employee;***

***SELECT MIN(salary) FROM employee;***

***SELECT COUNT(salary) FROM employee;***

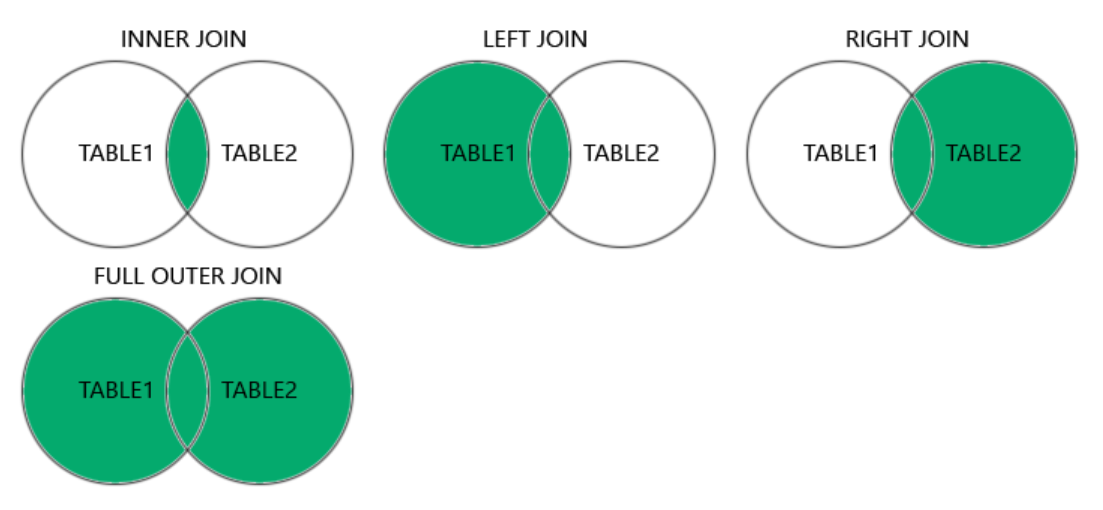
***SELECT COUNT(DISTINCT(salary)) FROM employee;***

**Renaming Temporary Column, Temporarily**

***SELECT price \* 1.5 AS “new\_price” FROM product;***

**Join Queries**

***SELECT students.s\_name FROM students INNER JOIN interns ON students.s\_ID = interns.i\_ID;***



* Write keywords in commands accordingly.
* In ***SELF JOIN***, take both tables as same table.