DAY-1

1. Data Definition Language (DDL)

DDL consists of commands that define and manage database structures.

a. Creating a Table

```
sql
CopyEdit
CREATE TABLE sample_data(cust_id number(4,0), cust_name VARCHAR(20),
email_id VARCHAR(20));
```

- Creates a table named `sample_data` with three columns:
    - `cust_id` (Numeric, 4 digits)
    - `cust_name` (Variable character, max 20 characters)
    - `email_id` (Variable character, max 20 characters)

b. Viewing Table Structure

```
sql
CopyEdit
DESCRIBE sample_data;
```

- Displays the structure of the table, including column names, data types, and constraints.

c. ALTER TABLE Command (Modifying Table Structure)

i. Adding a Column

```
sql
CopyEdit
ALTER TABLE sample_data ADD mobile_no CHAR(10);
```

- Adds a new column `mobile_no` (Fixed character length 10) to the table.

ii. Removing a Column

```
sql
CopyEdit
ALTER TABLE sample_data DROP COLUMN mobile_no;
```

- Removes the `mobile_no` column from the table.

iii. Renaming a Column

```
sql
CopyEdit
ALTER TABLE sample_data RENAME COLUMN cust_id TO customer_id;
```

- Changes the column name from `cust_id` to `customer_id`.

iv. Renaming the Table

```
sql
CopyEdit
ALTER TABLE sample_data RENAME TO customer;
```

- Changes the table name from `sample_data` to `customer`.

v. Modifying Column Size and Data Type

```
sql
CopyEdit
ALTER TABLE customer MODIFY cust_name VARCHAR(30);
ALTER TABLE customer MODIFY cust_name VARCHAR(10);
```

- Increases and then reduces the size of `cust_name`.

vi. Changing Constraints on Columns

```
sql
CopyEdit
ALTER TABLE customer ADD mobile_no number(10,0);
ALTER TABLE customer MODIFY mobile_no CHAR(10);
```

- Adds a `mobile_no` column as a number, then changes it to a character type.

d. Deleting Data vs. Deleting the Table

i. Truncating a Table (Removes Data but Keeps Structure)

```
sql
CopyEdit
TRUNCATE TABLE customer;
```

- Deletes all rows but keeps the table structure intact.

ii. Dropping a Table (Removes Data & Structure Permanently)

```
sql
CopyEdit
DROP TABLE customer;
```

- Completely removes the table and all its data.

2. Data Manipulation Language (DML)

DML includes commands for inserting, updating, and deleting data.

a. Creating an Employee Table

```
sql
CopyEdit
CREATE TABLE employee(empno number, ename varchar2(20), city
varchar2(15), salary number(8,2));
```

- Creates an employee table with four columns.

b. INSERT INTO Statement (Adding Data)

```
sql
CopyEdit
INSERT INTO employee (empno , ename , city, salary) VALUES
(101,'Ajit','Hyderabad',10000.00);
```

- Adds a new row to the employee table.

i. Inserting All Columns (Alternate Syntax)

```
sql
CopyEdit
INSERT INTO employee VALUES (101,'Ajit','Hyderabad',10000.00);
```

- Inserts values without specifying column names (but must follow table structure).

ii. Inserting Partial Data (Causes Error if Constraints Exist)

```
sql
CopyEdit
INSERT INTO employee VALUES(102,'Chithra'); -- Error
```

- Causes an error because all columns require values unless they allow NULL.

iii. Correcting Partial Insert

```
sql
CopyEdit
INSERT INTO employee(empno, ename) VALUES(102,'Chithra');
```

- Works because only empno and ename are provided.

c. INSERT ALL (Multiple Inserts at Once)

```
sql
CopyEdit
INSERT ALL
    INTO employee (empno, ename, city, salary) VALUES (104,
'srijit', 'Mumbai', 14000.00)
    INTO employee (empno, ename, city, salary) VALUES (105, 'Elsa',
'Pune', 15500)
SELECT * FROM dual;
```

- Inserts multiple rows into the table in a single statement.

d. Updating Data

```
sql
CopyEdit
```

```
UPDATE employee SET empno=103 WHERE ename='Chithra';
```

- Changes empno to 103 for Chithra.

```
sql
CopyEdit
UPDATE employee SET salary = salary*1.1;
```

- Increases everyone's salary by 10%.

```
sql
CopyEdit
UPDATE employee SET city = 'Delhi' WHERE ename='Chithra';
```

- Updates city for Chithra.

```
sql
CopyEdit
UPDATE employee SET salary = salary*1.2 WHERE empno < 103;
```

- Increases salary by 20% for employees with empno less than 103.

e. DELETE Statement (Removing Data)

```
sql
CopyEdit
DELETE FROM employee WHERE salary > 20000;
```

- Removes all employees earning more than 20,000.

```
sql
CopyEdit
DELETE FROM employee;
```

- Deletes all records but keeps the table structure.

f. Differences Between DELETE and TRUNCATE

Feature	DELETE	TRUNCATE
Type	DML	DDL
Condition Allowed?	Yes	No
Rollback Possible?	Yes	No
Space Released?	No	Yes
Performance	Slower	Faster

3. Constraints in SQL

Constraints enforce rules on data.

a. Unique Constraint

```
sql
CopyEdit
CREATE TABLE emp (empno NUMBER UNIQUE, ename VARCHAR2(20));
```

- Ensures empno values are unique.

b. Not NULL Constraint

```
sql
CopyEdit
ALTER TABLE emp MODIFY ename VARCHAR2(20) NOT NULL;
```

- Ensures ename cannot be NULL.

c. Primary Key (Unique & Not NULL)

```
sql
CopyEdit
CREATE TABLE emp (
    empno NUMBER,
    ename VARCHAR2(20),
    CONSTRAINT emp_pk_cons PRIMARY KEY (empno)
);
```

- empno is now a unique identifier and cannot be NULL.

d. Foreign Key (Referencing Another Table)

```
sql
CopyEdit
CREATE TABLE project (
    proj_id NUMBER PRIMARY KEY,
    duration NUMBER,
    empid NUMBER,
    CONSTRAINT project_fk_cons FOREIGN KEY(empid) REFERENCES
emp(empno)
);
```

- Links `empid` in `project` table to empno in emp table.

e. Foreign Key Characteristics

1. Can contain NULL values.
2. Can have duplicate values.
3. Must match a value in the referenced primary key.

f. Handling Foreign Key Deletions

i. ON DELETE SET NULL

```
sql
CopyEdit
CREATE TABLE project (
    proj_id NUMBER PRIMARY KEY,
    duration NUMBER,
    empid NUMBER,
    CONSTRAINT project_fk_cons FOREIGN KEY(empid) REFERENCES
emp(empno) ON DELETE SET NULL
);
```

- If an employee is deleted, their `empid` in `project` becomes NULL.

ii. ON DELETE CASCADE

```
sql
CopyEdit
CREATE TABLE project (
    proj_id NUMBER PRIMARY KEY,
    duration NUMBER,
    empid NUMBER,
    CONSTRAINT project_fk_cons FOREIGN KEY(empid) REFERENCES
emp(empno) ON DELETE CASCADE
);
```

- If an employee is deleted, all their related projects are also deleted.

Summary

This document covered:

1. **DDL (CREATE, ALTER, DROP, TRUNCATE)**
2. **DML (INSERT, UPDATE, DELETE)**
3. **Constraints (UNIQUE, PRIMARY KEY, FOREIGN KEY, NOT NULL)**
4. **Foreign Key behaviors (ON DELETE CASCADE, ON DELETE SET NULL)**