



## Worksheet 1

**Student Name:** Sujal Dhiman

**Branch:** MCA

**Semester:** 2nd

**Subject Name:** Technical Training

**UID:** 25MCA20186

**Section/Group:** MCA1(A)

**Date of Performance:** 06/01/2026

**Subject Code:** 25CAP-652

### AIM:

To design and implement a sample database system using DDL, DML, and DCL commands, including database creation, data manipulation, schema modification, and role-based access control to ensure data integrity and secure, read-only access for authorized users.

### Objective:

To gain practical experience in implementing Data Definition Language (DDL), Data Manipulation Language (DML), and Data Control Language (DCL) operations in a real database environment. This will also include implementing role-based privileges to secure data

### Coding:

```
CREATE TABLE Division (
    division_id INT PRIMARY KEY,
    division_name VARCHAR(30) UNIQUE NOT NULL
);
```

```
CREATE TABLE Staff (
    staff_id INT PRIMARY KEY,
    staff_name VARCHAR(30) NOT NULL,
    pay INT CHECK (pay > 0),
```

---

```
division_id INT,  
CONSTRAINT fk_div FOREIGN KEY (division_id) REFERENCES Division(division_id)  
);
```

```
CREATE TABLE Assignment (  
assignment_id INT PRIMARY KEY,  
assignment_title VARCHAR(30) NOT NULL,  
division_id INT,  
CONSTRAINT fk_assign_div FOREIGN KEY (division_id)  
REFERENCES Division(division_id)  
);
```

```
INSERT INTO Division VALUES  
(10, 'ADMIN'),  
(20, 'ACCOUNTS'),  
(30, 'TECH'),  
(40, 'MARKETING');
```

```
INSERT INTO Staff VALUES  
(501, 'RAVI SHARMA', 60000, 20),  
(502, 'NEHA VERMA', 48000, 10),  
(503, 'ARJUN MEHTA', 52000, 30),  
(504, 'PRIYA GUPTA', 55000, 40);
```

```
INSERT INTO Assignment VALUES  
(901, 'Budget Planning', 20),
```

---

(902, 'System Upgrade', 30),

(903, 'Client Outreach', 40);

UPDATE Staff

SET pay = 51000

WHERE staff\_id = 502;

DELETE FROM Assignment

WHERE assignment\_id = 902;

SELECT CURRENT\_USER;

CREATE ROLE view\_only\_staff

WITH LOGIN PASSWORD 'view123';

SELECT CURRENT\_USER;

GRANT SELECT ON Division TO view\_only\_staff;

GRANT SELECT ON Staff TO view\_only\_staff;

GRANT SELECT ON Assignment TO view\_only\_staff;

REVOKE CREATE ON SCHEMA public FROM view\_only\_staff;

REVOKE INSERT, UPDATE, DELETE ON Division FROM view\_only\_staff;

REVOKE INSERT, UPDATE, DELETE ON Staff FROM view\_only\_staff;

REVOKE INSERT, UPDATE, DELETE ON Assignment FROM view\_only\_staff;

## ALTER TABLE Assignment

```
ALTER COLUMN assignment_title TYPE VARCHAR(100);
```

```
DROP TABLE Assignment;
```

```
SELECT * FROM division;
```

```
SELECT * FROM assignment;
```

```
SELECT * FROM staff;
```

## OUTPUT :

	division_id [PK] integer	division_name character varying (30)
1	10	ADMIN
2	20	ACCOUNTS
3	30	TECH
4	40	MARKETING

**Fig 1.1: Division Table**

	assignment_id [PK] integer	assignment_title character varying (30)	division_id integer
1	901	Budget Planning	20
2	903	Client Outreach	40

**Fig 1.2: Assignment Table**

	staff_id [PK] integer	staff_name character varying (30)	pay integer	division_id integer
1	501	RAVI SHARMA	60000	20
2	503	ARJUN MEHTA	52000	30
3	504	PRIYA GUPTA	55000	40
4	502	NEHA VERMA	51000	10

**Fig 1.2: Staff Table**

## 5. Learning outcomes (What I have learnt) :

1. Understand the working principle of the Quick Sort algorithm, including pivot selection, partitioning, and recursive sorting.
2. Analyze the time complexity of Quick Sort in best, average, and worst cases, and relate it to practical runtime measurements in Python.
3. Implement Quick Sort using recursion in Python, including correct handling of indices and in-place array manipulation.
4. Evaluate algorithm performance by measuring execution time, interpreting results using Python's time module.
5. Compare sorted and original datasets, demonstrating the correctness of the algorithm and reinforcing debugging and verification skills.

## 6. Evaluation Grid :

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Worksheet Completion		08 marks
2.	Conduct		12 marks
3.	Viva Voice		10 marks
4.	Total		30 marks