Q1. Create a New Database and Table for Employees Task:

Create a new database named **company_db** and create a table named **employees** with the following columns:

Column Name	Data Type	Constraint
employee_id	INT	PRIMARY KEY
first_name	VARCHAR(50)	
last_name	VARCHAR(50)	
department	INT	
salary	VARCHAR(50)	
hire_date	DATE	

Ans:

```
CREATE DATABASE company_db;

USE company_db;

CREATE TABLE employees
(
    Employee_id     INT PRIMARY KEY,
    first_name      VARCHAR(50),
    last_name       VARCHAR(50),
    department      VARCHAR(50),
    salary          INT,
    hire_date       DATE
                    );
```

Q2. Insert Data into Employees Table Task:

Insert the following sample records into the employees table.

employee_id	first_name	last_name	department	salary	hire_date
101	Amit	Sharma	HR	50000	2020-01-15
102	Riya	Kapoor	Sales	75000	2019-03-22
103	Raj	Mehta	IT	90000	2018-07-11
104	Neha	Verma	IT	85000	2021-09-01
105	Arjun	Singh	Finance	60000	2022-02-10

Ans:

```
INSERT INTO employees VALUES
        (101, 'Amit', 'Sharma', 'HR', 50000, '2020-01-15'),
        (102, 'Riya', 'Kapoor',    'Sales', 75000,  '2019-03-22'),
        (103, 'Raj', 'Mehta', 'IT', 90000, '2018-07-11'),
        (104, 'Neha', 'Verma', 'IT', 85000, '2021-09-01'),
        (105, 'Arjun', 'Singh', 'Finance', 60000, '2022-02-10');
```

Q3. Display All Employee Records Sorted by Salary (Lowest to Highest)

Hint: Use the ORDER BY clause on the salary column.

Ans:

```
SELECT * FROM employees
order by salary ASC;
```

employee_id	first_name	last_name	department	salary	hire_date
101	Amit	Sharma	HR	50000	2020-01-15
105	Arjun	Singh	Finance	60000	2022-02-10
102	Riya	Kapoor	Sales	75000	2019-03-22
104	Neha	Verma	IT	85000	2021-09-01
103	Raj	Mehta	IT	90000	2018-07-11

Q4. Show Employees Sorted by Department (A–Z) and Salary (High → Low).

Ans:

```
SELECT employee_id, first_name, last_name, department, salary, hire_date
        FROM employees
ORDER BY department ASC,
          salary DESC;
```

Q5. List All Employees in the IT Department, Ordered by Hire Date (Newest First).

Ans:

```
SELECT employee_id, first_name, last_name, department, salary, hire_date
        FROM employees
 WHERE department = 'IT'
ORDER BY hire_date desc;
```

Q6. Create and Populate a Sales Table

Task: Create a table sales to track sales data:

Ans:

```
CREATE TABLE sales
(
 Sale_id          INT PRIMARY KEY,
 Customer_name    VARCHAR(50),
 amount           INT,
 sale_date        DATE
 );

INSERT INTO sales VALUES (1, 'Aditi', 1500,  '2024-08-01'),
                         (2, 'Rohan', 2200,  '2024-08-03'),
                         (3, 'Aditi', 3500,  '2024-09-05'),
                         (4, 'Meena', 2700,  '2024-09-15'),
                         (5, 'Rohan', 4500,  '2024-09-25');
```

Q7. Display All Sales Records Sorted by Amount (Highest → Lowest)
Hint: Use ORDER BY amount DESC.

Ans:

```
SELECT * FROM sales
        ORDER BY amount DESC;
```

Q8. Show All Sales Made by Customer "Aditi"
Hint: Use WHERE customer_name = 'Aditi'

Ans:

```
SELECT * FROM sales
        where customer_name ='Aditi';
```

Q9. What is the Difference Between a Primary Key and a Foreign Key?

Ans: A **Primary Key** uniquely identifies each record in a table. It cannot contain NULL values.
A table can have only one primary key.
A **Foreign Key** is used to create a link between two tables. It references the primary key of another table.
It can contain NULL unless restricted.

Q10. What Are Constraints in SQL and Why Are They Used?

Ans: Constraints are rules applied on table columns in SQL to maintain data accuracy, reliability, and integrity in the database. Ex- PRIMARY KEY, FOREIGN KEY, UNIQUE, NOT NULL

They ensure that only valid data is stored and prevent mistakes like duplicates, missing values, or invalid relationships.