# SQL Queries

**Database Creation:**

CREATE DATABASE EmployeeManagementSystem;

**Table Creation:**

CREATE TABLE employees (

emp\_id INT PRIMARY KEY,

emp\_name VARCHAR(50),

emp\_salary DECIMAL(10, 2),

emp\_department VARCHAR(50)

);

CREATE TABLE departments (

dept\_id INT PRIMARY KEY,

dept\_name VARCHAR(50),

dept\_location VARCHAR(50)

);

**Data Insertion:**

INSERT INTO employees (emp\_id, emp\_name, emp\_salary, emp\_department) VALUES (1, 'John Doe', 50000.00, 'IT');

INSERT INTO employees (emp\_id, emp\_name, emp\_salary, emp\_department) VALUES (2, 'Jane Smith', 60000.00, 'HR');

INSERT INTO departments (dept\_id, dept\_name, dept\_location) VALUES (1, 'IT', 'New York');

INSERT INTO departments (dept\_id, dept\_name, dept\_location) VALUES (2, 'HR', 'London');

**Dropping a Table:**

DROP TABLE departments;

Table Alteration:

ALTER TABLE employees ADD emp\_email VARCHAR(100);

**Table Updating:**

UPDATE employees SET emp\_salary = 55000.00 WHERE emp\_id = 1;

Different Forms of Select Queries:

SELECT \* FROM employees;

SELECT \* FROM employees WHERE emp\_salary > 55000.00;

SELECT \* FROM employees WHERE emp\_department = 'IT';

SELECT \* FROM employees WHERE emp\_salary BETWEEN 40000.00 AND 60000.00;

SELECT MAX(emp\_salary) AS highest\_salary FROM employees;

SELECT emp\_department, COUNT(\*) AS employee\_count FROM employees GROUP BY emp\_department;

**Inner join**

SELECT employees.emp\_name, employees.emp\_salary, departments.dept\_name

FROM employees

INNER JOIN departments ON employees.emp\_department = departments.dept\_name;

**Left Join**

SELECT employees.emp\_name, employees.emp\_salary, departments.dept\_name

FROM employees

LEFT JOIN departments ON employees.emp\_department = departments.dept\_name;

**Right Join:**

SELECT employees.emp\_name, employees.emp\_salary, departments.dept\_name

FROM employees

RIGHT JOIN departments ON employees.emp\_department = departments.dept\_name;

**Full outer Join**

SELECT employees.emp\_name, employees.emp\_salary, departments.dept\_name

FROM employees

FULL OUTER JOIN departments ON employees.emp\_department = departments.dept\_name;

# MST Python Practice Problem Day 7

**Practice Problem**

**Database Creation:**

Create a new database named "OnlineBookstore".

**Tables Creation:**

Create the "books" table with columns: book\_id (primary key), title, author\_id (foreign key), price, and publication\_year.

Create table books(book\_id int primary key,title varchar(50),author\_id int,price decimal(10,2),publication\_year year, constraint fk\_author foreign key(author\_id) references authors(author\_id));

Query OK, 0 rows affected (0.074 sec)

Create the "authors" table with columns: author\_id (primary key), author\_name, and country.

Create the "orders" table with columns: order\_id (primary key), book\_id (foreign key), customer\_name, and order\_date.

Create table orders(order\_id int primary key,book\_id int,customer\_name varchar(50),order\_date date, constraint fk\_order foreign key(book\_id) references books(book\_id));

**Data Insertion:**

Insert sample data into the "books" table.

Insert sample data into the "authors" table.

Insert sample data into the "orders" table.

**Data Retrieval:**

Retrieve all books from the "books" table.

Retrieve all authors from the "authors" table.

Retrieve all orders from the "orders" table.

**Data Relationships:**

Update the "books" table to include a new column, genre.

Update the "orders" table to include a new column, quantity.

**Data Retrieval with Relationships:**

Retrieve all books along with their author information.

Retrieve all orders along with the book title and customer information.