```
-- Step 1: Create Database
CREATE DATABASE PracticalDB;
USE PracticalDB;

-- Step 2: Create Tables

-- Employees Table
CREATE TABLE employees (
    employee_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    salary DECIMAL(10,2),
    department_id INT
);

-- Departments Table
CREATE TABLE departments (
    department_id INT PRIMARY KEY,
    department_name VARCHAR(50),
    location VARCHAR(50)
);

-- Customers Table
CREATE TABLE customers (
    customer_id INT PRIMARY KEY,
    customer_name VARCHAR(50)
);

-- Products Table
CREATE TABLE products (
    product_id INT PRIMARY KEY,
    product_name VARCHAR(50),
    supplier_id INT
);
```

```
-- Suppliers Table
CREATE TABLE suppliers (
    supplier_id INT PRIMARY KEY,
    supplier_name VARCHAR(50)
);

-- Orders Table
CREATE TABLE orders (
    order_id INT PRIMARY KEY,
    customer_id INT,
    product_id INT,
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id),
    FOREIGN KEY (product_id) REFERENCES products(product_id)
);

-- Step 3: Insert Sample Data

-- Insert Data into Employees Table
INSERT INTO employees (employee_id, first_name, last_name, salary, department_id) VALUES
(1, 'John', 'Doe', 60000, 1),(2, 'Jane', 'Smith', 75000, 2),(3, 'Michael', 'Johnson', 80000, 1),
(4, 'Emily', 'Davis', 50000, 3),(5, 'Daniel', 'Brown', 67000, 2),(6, 'Jessica', 'Miller', 72000, 3),
(7, 'David', 'Wilson', 55000, 1),(8, 'Sarah', 'Moore', 51000, 2),(9, 'James', 'Taylor', 88000, 3),
(10, 'Laura', 'Anderson', 65000, 1);

-- Insert Data into Departments Table
INSERT INTO departments (department_id, department_name, location) VALUES
(1, 'HR', 'New York'),(2, 'Sales', 'Chicago'),(3, 'IT', 'San Francisco'),(4, 'Marketing', 'Los Angeles'),
(5, 'Finance', 'Houston'),(6, 'Operations', 'Seattle'),(7, 'Legal', 'Boston'),(8, 'Engineering', 'Austin'),
(9, 'Support', 'Denver'),(10, 'Logistics', 'Miami');

-- Insert Data into Customers Table
INSERT INTO customers (customer_id, customer_name) VALUES
(1, 'Alice Johnson'),(2, 'Robert Brown'),(3, 'Charlie Davis'),(4, 'Emma Wilson'),(5, 'Oliver White'),(6, 'Sophia Thomas'),
(7, 'Liam Martinez'),(8, 'Isabella Clark'),(9, 'Mason Hall'),(10, 'Mia Young');
```

-- Insert Data into Products Table

INSERT INTO products (product_id, product_name, supplier_id) VALUES

(1, 'Laptop', 1),(2, 'Smartphone', 2),(3, 'Tablet', 3),(4, 'Monitor', 4),(5, 'Keyboard', 5),(6, 'Mouse', 6),(7, 'Printer', 7),

(8, 'Headphones', 8),(9, 'Camera', 9),(10, 'Smartwatch', 10);

-- Insert Data into Suppliers Table

INSERT INTO suppliers (supplier_id, supplier_name) VALUES

(1, 'TechCorp'),(2, 'MobileWorld'),(3, 'TabTech'),(4, 'Display Inc.'),(5, 'KeyBoards Ltd.'),(6, 'MouseHouse'),

(7, 'PrintMasters'),(8, 'AudioSolutions'),(9, 'PhotoGear'),(10, 'WearableTech');

-- Insert Data into Orders Table

INSERT INTO orders (order_id, customer_id, product_id) VALUES

(1, 1, 2),(2, 2, 4),(3, 3, 6),(4, 4, 8),(5, 5, 10),(6, 6, 1),(7, 7, 3),(8, 8, 5),(9, 9, 7),(10, 10, 9);

1 **Basic SELECT Query**

SELECT first_name, last_name, salary FROM employees;

First Name	Last Name	Salary
John	Doe	60000
Jane	Smith	75000
Michael	Johnson	80000
Emily	Davis	50000
Daniel	Brown	67000
Jessica	Miller	72000
David	Wilson	55000
Sarah	Moore	51000
James	Taylor	88000
Laura	Anderson	65000

SELECT with WHERE Clause

SELECT first_name, last_name, salary FROM employees WHERE salary > 50000;

First Name	Last Name	Salary
John	Doe	60000
Jane	Smith	75000
Michael	Johnson	80000
Daniel	Brown	67000
Jessica	Miller	72000
David	Wilson	55000
Sarah	Moore	51000
James	Taylor	88000
Laura	Anderson	65000

Subqueries (Nested Queries)

SELECT column1, column2 FROM table_name WHERE column3 = (SELECT column3 FROM table_name WHERE condition);

first_name	last_name
John	Doe
Michael	Johnson
David	Wilson
Laura	Anderson

1: Single-row Subquery

SELECT first_name, last_name, salary FROM employees

WHERE salary > (SELECT AVG(salary) FROM employees);

First Name	Last Name	Salary
John	Doe	60000
Jane	Smith	75000
Michael	Johnson	80000
Daniel	Brown	67000
Jessica	Miller	72000
David	Wilson	55000
Sarah	Moore	51000
James	Taylor	88000
Laura	Anderson	65000

2: Multiple-row Subquery

SELECT first_name, last_name FROM employees

WHERE department_id IN (SELECT department_id FROM departments WHERE location = 'New York');

First Name	Last Name
John	Doe
Michael	Johnson
David	Wilson
Laura	Anderson

3: Correlated Subquery

SELECT first_name, last_name, salary, department_id FROM employees e

WHERE salary > (SELECT AVG(salary) FROM employees WHERE department_id = e.department_id);

first_name	last_name	salary	department_id
Michael	Johnson	80000	1
Jane	Smith	75000	2
Daniel	Brown	67000	2
Jessica	Miller	72000	3
James	Taylor	88000	3

JOIN

1: Simple Inner Join

SELECT e.first_name, e.last_name, d.department_name FROM employees e INNER JOIN departments d ON e.department_id = d.department_id;

First Name	Last Name	Department Name
John	Doe	HR
Jane	Smith	Sales
Michael	Johnson	HR
Emily	Davis	IT
Daniel	Brown	Sales
Jessica	Miller	IT
David	Wilson	HR
Sarah	Moore	Sales
James	Taylor	IT
Laura	Anderson	HR

2: Using JOIN with Multiple Tables

SELECT c.customer_name, p.product_name FROM customers c

INNER JOIN orders o ON c.customer_id = o.customer_id

INNER JOIN products p ON o.product_id = p.product_id;

Customer Name	Product Name
Alice Johnson	Smartphone
Robert Brown	Monitor
Charlie Davis	Mouse
Emma Wilson	Headphones
Oliver White	Smartwatch
Sophia Thomas	Laptop
Liam Martinez	Tablet
Isabella Clark	Keyboard
Mason Hall	Printer
Mia Young	Camera

3: Using LEFT JOIN

SELECT e.first_name, e.last_name, d.department_name FROM employees e

LEFT JOIN departments d ON e.department_id = d.department_id;

First Name	Last Name	Department Name
John	Doe	HR
Jane	Smith	Sales
Michael	Johnson	HR
Emily	Davis	IT
Daniel	Brown	Sales
Jessica	Miller	IT
David	Wilson	HR
Sarah	Moore	Sales
James	Taylor	IT
Laura	Anderson	HR

In this practical we use this joins