1. You have three tables: Products, Categories, and Sales. The Products table contains product information with columns ProductId, ProductName, and CategoryId. The Categories table contains category information with columns CategoryId and CategoryName. The Sales table contains sales information with columns SaleId, ProductId, QuantitySold, SaleDate, and SaleAmount. Write a SQL query to find the top 5 categories with the highest total sales amount for the past year. The result should include the category name and the total sales amount, and it should be ordered by the total sales amount in descending order.

```
create database db1;

use db1;
CREATE TABLE Categories (
  CategoryId INT PRIMARY KEY,
  CategoryName VARCHAR(100) NOT NULL
);

CREATE TABLE Products (
  ProductId INT PRIMARY KEY,
  ProductName VARCHAR(100) NOT NULL,
  CategoryId INT,
  FOREIGN KEY (CategoryId) REFERENCES Categories(CategoryId)
);

CREATE TABLE Sales (
  SaleId INT PRIMARY KEY,
  ProductId INT,
  QuantitySold INT,
  SaleDate DATE,
  SaleAmount DECIMAL(10,2),
```

```
  FOREIGN KEY (ProductId) REFERENCES Products(ProductId)
);


INSERT INTO Categories VALUES
(1, 'Electronics'),
(2, 'Clothing'),
(3, 'Shoes'),
(4, 'Sports'),
(5, 'Cooking'),
(6, 'Toys');

INSERT INTO Products VALUES
(1, 'Laptop', 1),
(2, 'Smartphone', 1),
(3, 'T-Shirt', 2),
(4, 'Formal shoes', 3),
(5, 'Hockey', 4),
(6, 'Toy Car', 6),
(7, 'Induction', 5);


INSERT INTO Sales VALUES
(1, 1, 2, '2024-07-01', 2000.00),
(2, 2, 1, '2024-08-15', 800.00),
(3, 3, 3, '2024-09-10', 90.00),
(4, 4, 2, '2023-10-05', 40.00),
(5, 5, 1, '2024-11-20', 500.00),
(6, 6, 4, '2024-12-25', 120.00),
```

```
(7, 1, 1, '2023-04-15', 1000.00),
(8, 7, 1, '2022-03-15', 1000.00);


SELECT * FROM Categories;
SELECT * FROM Products;
SELECT * FROM Sales;


-- Write a SQL query to find the top 5 categories with the highest total sales amount for the past year.
-- The result should include the category name and the total sales amount, and it should be ordered by the total sales amount
-- in descending order.

SELECT c.CategoryName, SUM(s.SaleAmount) AS TotalSalesAmount
FROM Sales s
INNER JOIN Products p ON s.ProductId = p.ProductId
INNER JOIN Categories c ON p.CategoryId = c.CategoryId
WHERE s.SaleDate >= DATE_SUB(CURDATE(), INTERVAL 1 YEAR)
GROUP BY c.CategoryId, c.CategoryName
ORDER BY TotalSalesAmount DESC
LIMIT 5;
```

2. You have three tables: Employees, Departments, and Salaries. The Employees table contains employee information with columns EmployeeId, EmployeeName, and DepartmentId. The Departments table contains department information with columns DepartmentId and DepartmentName.

The Salaries table contains salary information with columns EmployeeId, Salary, and SalaryDate. Write a SQL query to find the average salary of employees in each department for the current year. The result should include the department name and the average salary, and it should be ordered by the average salary in descending order.

```
create database db2;

use db2;

CREATE TABLE Departments (
  DepartmentId INT PRIMARY KEY,
  DepartmentName VARCHAR(20) NOT NULL
);

INSERT INTO Departments (DepartmentId, DepartmentName) VALUES
(1, 'Engineering'),
(2, 'HR'),
(3, 'Sales');

CREATE TABLE Employees (
  EmployeeId INT PRIMARY KEY,
  EmployeeName VARCHAR(50) NOT NULL,
  DepartmentId INT,
  FOREIGN KEY (DepartmentId) REFERENCES Departments(DepartmentId)
);

INSERT INTO Employees (EmployeeId, EmployeeName, DepartmentId) VALUES
(1, 'Alice', 1),
```

```
(2, 'Bob', 1),
(3, 'Carol', 2),
(4, 'Dave', 3),
(5, 'Eve', 3);

CREATE TABLE Salaries (
  EmployeeId INT,
  Salary DECIMAL(10,2),
  SalaryDate DATE,
  FOREIGN KEY (EmployeeId) REFERENCES Employees(EmployeeId)
);

INSERT INTO Salaries (EmployeeId, Salary, SalaryDate) VALUES
(1, 75000, '2025-01-15'),
(2, 70000, '2025-02-20'),
(3, 60000, '2025-03-10'),
(4, 55000, '2025-04-05'),
(5, 53000, '2025-05-22'),
(1, 76000, '2024-12-10'),
(2, 72000, '2024-11-10');



-- Write a SQL query to find the average salary of employees in each department for the current year.
-- The result should include the department name and the average salary, and it should be ordered by the average salary in descending order.

SELECT d.DepartmentName, AVG(s.Salary) AS AverageSalary
FROM Employees e
Inner JOIN Departments d ON e.DepartmentId = d.DepartmentId
```

```
Inner JOIN Salaries s ON e.EmployeeId = s.EmployeeId
WHERE YEAR(s.SalaryDate) = YEAR(CURDATE())
GROUP BY d.DepartmentId, d.DepartmentName
ORDER BY AverageSalary DESC;
```