1. Retrieve the names of all customers (in alphabetical order) from the "Customers" table.
2. List the names of products from the "Products" table that have a price greater than $50.
3. Retrieve the total number of orders for each customer from the "Orders" table. Display the results in descending order of order count.
4. Find the average salary of employees from the "Employees" table, but only for those whose job title is "Manager."
5. Retrieve the names and ages of customers who have placed at least three orders. Display the results in alphabetical order of customer names.
6. List the product names and their respective prices from the "Products" table, ordered by price in descending order.
7. Retrieve the department names and the average salary of employees in each department from the "Employees" table. Display only those departments where the average salary is greater than $60,000.
8. Find the total number of orders placed by each customer who is located in a city, starting with the letter "S." Display the results in ascending order of customer names.
9. Retrieve the names and ages of customers who have placed orders with a total value of at least $1,000. Display the results in alphabetical order of customer names.
10. List the names of employees from the "Employees" table who have the same job title and are part of the same department. Sort the results by department name and then by employee name.
11. Consider a table named "Employees" with the following columns:
    a. EmployeeID (int): Unique identifier for each employee
    b. FirstName (varchar): First name of the employee
    c. LastName (varchar): Last name of the employee
    d. DepartmentID (int): Identifier for the department the employee belongs to
    e. Salary (decimal): The salary of the employee.

Write an SQL query to retrieve all columns for all rows from the "Employees" table.

12. Consider a table named "Products" with the following columns:

a. ProductID (int): Unique identifier for each product
b. ProductName (varchar): Name of the product
c. CategoryID (int): Identifier for the product category
d. Price (decimal): The price of the product.

Write an SQL query to retrieve only the "ProductID" and "ProductName" columns for all rows from the "Products" table.

13. Write an SQL query to retrieve a list of distinct cities from the "Customers" table.

14. Consider a table named "Orders" with the following columns:
    a. OrderID (int): Unique identifier for each order
    b. ProductID (int): Identifier for the product in the order
    c. Quantity (int): Quantity of the product ordered
    d. PricePerUnit (decimal): The price per unit of the product.

Write an SQL query to calculate the total price (Quantity * PricePerUnit) for each order.

15. Write an SQL query to retrieve the details (CustomerID, FirstName, LastName, Age) of all customers who are exactly 30 years old.

16. Write an SQL query to retrieve the details (ProductID, ProductName, Price) of all products with a price between $50 and $100.

17. Write an SQL query to retrieve the details (CustomerID, FirstName, LastName, Age) of all customers who are either 25 years old or younger (Age <= 25) or 45 years old or older (Age >= 45).

18. Write an SQL query to retrieve the details (ProductID, ProductName, Price, Rating) of all products with a price less than $50 (Price < 50) and a rating of at least 4 (Rating >= 4).

19. Write an SQL query to retrieve the details (EmployeeID, FirstName, LastName, DepartmentID) of all employees who work in either the Sales department (DepartmentID = 1) or the Marketing department (DepartmentID = 2).

20. Consider a table named "Orders" with the following columns:
    a. OrderID (int): Unique identifier for each order
    b. OrderDate (date): Date of the order
    c. OrderStatus (varchar): Status of the order (e.g., "Pending," "Shipped," "Delivered").

Write an SQL query to retrieve the details (OrderID, OrderDate, OrderStatus) of all orders that are either "Pending" or "Shipped."

21. Write an SQL query to retrieve the details (EmployeeID, FirstName, LastName, Salary) of all employees with salaries between $40,000 and $60,000.

22. Write an SQL query to retrieve the details (OrderID, OrderDate, OrderStatus) of all orders placed between January 1, 2023, and December 31, 2023.

23. Write an SQL query to retrieve the details (ProductID, ProductName, QuantityInStock) of all products with a quantity in stock between 10 and 100.

24. Write an SQL query to retrieve the details (CustomerID, FirstName, LastName, Age) of all customers who are between 18 and 30 years old.

25. Write an SQL query to retrieve the details (CustomerID, FirstName, LastName) of customers with IDs 101, 204, and 305.

26. Write an SQL query to retrieve the details (ProductID, ProductName) of products that do not belong to category 4.

27. Write an SQL query to retrieve the details (CustomerID, FirstName, LastName) of customers who are not listed with IDs 102, 205, and 306.

28. Consider a table named "Products" with the following columns:
    a. ProductID
    b. ProductName

    Consider another table named "ProductReviews" with the following columns:
    a. ReviewID
    b. ProductID

    Write an SQL query to retrieve the details (ProductID, ProductName) of products that have no reviews.

29. Consider a table named "Customers" with the following columns: CustomerID, FirstName and LastName.

    Consider another table named "Orders" with the following columns: OrderID and CustomerID.

Write an SQL query to retrieve the details (CustomerID, FirstName, LastName) of customers who have not placed any orders.

30. Write an SQL query to retrieve the details (CustomerID, FirstName, LastName) of customers whose first names start with the letter 'A'.

31. Write an SQL query to retrieve the details (ProductID, ProductName) of products whose names contain the word 'Phone'.

32. Consider a table named "Employees" with the following columns:
    a. EmployeeID (int): Unique identifier for each employee
    b. FirstName (varchar): First name of the employee
    c. LastName (varchar): Last name of the employee
    d. ManagerID (int): Identifier for the employee's manager. This field contains NULL if the employee has no manager.

    Write an SQL query to retrieve the details (EmployeeID, FirstName, LastName) of employees who do not have an assigned manager.

33. Consider a table named "Orders" with the following columns:
    a. OrderID (int): Unique identifier for each order.
    b. CustomerID (int): Identifier for the customer who placed the order. This field contains NULL if the customer information is missing.
    c. OrderDate (date): Date of the order.

    Write an SQL query to retrieve the details (OrderID, OrderDate) of orders with missing customer information.

34. Write an SQL query to retrieve the details (EmployeeID, FirstName, LastName, DepartmentID, Salary) of employees sorted first by department (in ascending order) and then by salary (in descending order).

35. Write an SQL query to retrieve the details (OrderID, CustomerID, OrderDate) of orders sorted first by customer (ascending) and then by order date (ascending).

36. Consider two tables: "Customers" and "Orders" with the following columns:

    "Customers" Table:

    - CustomerID (int): Unique identifier for each customer.

- FirstName (varchar): First name of the customer.
- LastName (varchar): Last name of the customer.

"Orders" Table:

- OrderID (int): Unique identifier for each order.
- CustomerID (int): Identifier for the customer who placed the order.

Write an SQL query to count the total number of orders and the number of distinct customers who have placed orders.

37. Consider a table named "OrderDetails" with the following columns:

    a. OrderDetailID (int): Unique identifier for each order detail.
    b. OrderID (int): Identifier for the order.
    c. ProductID (int): Identifier for the product.

    Write an SQL query to count the number of unique products sold across all orders.

38. Consider a table named "Employees" with the following columns:

    a. EmployeeID (int): Unique identifier for each employee.
    b. FirstName (varchar): First name of the employee.
    c. LastName (varchar): Last name of the employee.
    d. DepartmentID (int): Identifier for the department the employee belongs to.
    e. Salary (decimal): The salary of the employee.

    Write an SQL query to find the minimum, maximum, and average salary for each department and retrieve the department ID, minimum salary, maximum salary, and average salary.

39. Consider the same "Employees" table as in the previous question. Write an SQL query to find the departments with an average salary above $60,000 and retrieve the department ID and average salary.

40. Consider two tables: "Customers" and "Orders" with the following columns:

    "Customers" Table:

    - CustomerID (int): Unique identifier for each customer.

- FirstName (varchar): First name of the customer.
- LastName (varchar): Last name of the customer.

"Orders" Table:

- OrderID (int): Unique identifier for each order.
- CustomerID (int): Identifier for the customer who placed the order.
- OrderAmount (decimal): The total amount of the order.

Write an SQL query to find the minimum, maximum, and average order amount for customers who have placed more than one order. Retrieve the customer's first name, last name, minimum order amount, maximum order amount, and average order amount.

41. Consider two tables: "Customers" and "Orders" with the following columns:

    a. "Customers" Table: CustomerID, FirstName, LastName

    b. "Orders" Table: OrderID, CustomerID, OrderDate

    Write an SQL query to find the first and last names of customers who have placed more than two orders.

42. Consider two tables: "Customers" and "Orders" with the following columns:

    a. "Customers" Table: CustomerID, FirstName, LastName

    b. "Orders" Table: OrderID, CustomerID, OrderDate

    You have a specific customer with the ID of 101. Write an SQL query to find the first and last names of customers who placed orders on the same date as customer 101.

43. Consider two tables: "Customers" and "Orders" with the following columns:

    a. "Customers" Table: CustomerID, FirstName, LastName

    b. "Orders" Table: OrderID, CustomerID, OrderAmount

    Write an SQL query to find the first and last names of customers who have placed orders with a total amount greater than the average order amount.

44. Consider a table named "Products" with the following columns: ProductID, ProductName, CategoryID, Price.

    Write an SQL query to find the names of products with a price greater than the average price within their respective categories.

45. Consider a table named "Employees" with the following columns: EmployeeID, FirstName, LastName, DepartmentID, Salary.

    Write an SQL query to find the first and last names of employees with a salary greater than the average salary within their respective departments.

46. Consider three tables: "Customers," "Orders," and "Products" with the following columns:

    a. "Customers" Table: CustomerID, FirstName, LastName

    b. "Orders" Table: OrderID, CustomerID, ProductID

    c. "Products" Table: ProductID, CategoryID

    Write an SQL query to find the first and last names of customers who have placed orders in multiple product categories.

47. Consider two tables: "Customers" and "Orders” with the following columns:

    a. "Customers" Table: CustomerID, FirstName, LastName, Location

    b. "Orders" Table: OrderID, CustomerID, ProductName

    Write an SQL query to find the distinct product names purchased by customers who are located in "New York."

48. Consider two tables: "Customers" and "Orders" with the following columns:

    a. "Customers" Table: CustomerID, FirstName, LastName

    b. "Orders" Table: OrderID, CustomerID, OrderAmount

    Write an SQL query to find the first and last names of customers who have placed orders with a total amount greater than any order amount placed by customer 101.

49. Consider a table named "Products" with the following columns:

ProductID, ProductName, CategoryID, Price

Write an SQL query to find the names of products with a price greater than any product in category 1.

50. Consider two tables: "Customers" and "Orders" with the following columns:

    a. "Customers" Table: CustomerID, FirstName, LastName

    b. "Orders" Table: OrderID, CustomerID, OrderAmount

Write an SQL query to find the first and last names of customers who have placed orders with a total amount greater than all order amounts placed by customer 101.

51. Consider a table named "Products" with the following columns:

ProductID, ProductName, CategoryID, Price

Write an SQL query to find the names of products with a price greater than all products in category 1.

52. Consider two tables: "Customers" and "Orders" with the following columns:

    a. "Customers" Table: CustomerID, FirstName, LastName

    b. "Orders" Table: OrderID, CustomerID, OrderAmount

Write an SQL query to find the first and last names of customers who have placed orders with a total amount greater than some order amounts placed by customer 101.

53. Consider a table named "Products" with the following columns:

ProductID, ProductName, CategoryID, Price

Write an SQL query to find the names of products with a price greater than some products in category 1.

54. Consider three tables: "Customers," "Orders," and "OrderDetails" with the following columns:

"Customers" Table:

- CustomerID (int): Unique identifier for each customer.

- FirstName (varchar): First name of the customer.
- LastName (varchar): Last name of the customer.

"Orders" Table:

- OrderID (int): Unique identifier for each order.
- CustomerID (int): Identifier for the customer who placed the order.
- OrderDate (date): Date of the order.

"OrderDetails" Table:

- OrderDetailID (int): Unique identifier for each order detail.
- OrderID (int): Identifier for the order.
- ProductID (int): Identifier for the product.
- Quantity (int): Quantity of the product ordered.

Write an SQL query to find the first and last names of customers who have purchased products in category 1.

55. Consider a table named "Employees" with the following columns:

"Employees" Table:

- EmployeeID (int): Unique identifier for each employee.
- FirstName (varchar): First name of the employee.
- LastName (varchar): Last name of the employee.
- ManagerID (int): Identifier for the employee's manager.

Write an SQL query to find the first and last names of employees who manage other employees (i.e., employees with at least one direct report).

56. Consider two tables: "Customers" and "Orders" with the following columns:

"Customers" Table:

- CustomerID (int): Unique identifier for each customer.
- FirstName (varchar): First name of the customer.
- LastName (varchar): Last name of the customer.

"Orders" Table:

- OrderID (int): Unique identifier for each order.
- CustomerID (int): Identifier for the customer who placed the order.
- OrderDate (date): Date of the order.

Write an SQL query to retrieve the first and last names of customers along with the order IDs and order dates of their orders.

57. Consider a table named "Employees" with the following columns:

"Employees" Table:

- EmployeeID (int): Unique identifier for each employee.
- FirstName (varchar): First name of the employee.
- LastName (varchar): Last name of the employee.
- ManagerID (int): Identifier for the employee's manager.

Write an SQL query to retrieve the first and last names of employees along with the first and last names of their respective managers.

58. Consider two tables: "Customers" and "Orders" with the following columns:

"Customers" Table:

- CustomerID (int): Unique identifier for each customer.
- FirstName (varchar): First name of the customer.
- LastName (varchar): Last name of the customer.

"Orders" Table:

- OrderID (int): Unique identifier for each order.
- CustomerID (int): Identifier for the customer who placed the order.
- OrderDate (date): Date of the order.

Write an SQL query to retrieve the first and last names of all customers and their associated order dates, if they have placed any orders. Use a LEFT JOIN for this purpose.

59. Consider a table named "Employees" with the following columns:

"Employees" Table:

- EmployeeID (int): Unique identifier for each employee.
- FirstName (varchar): First name of the employee.
- LastName (varchar): Last name of the employee.
- ManagerID (int): Identifier for the employee's manager.

Write an SQL query to retrieve the first and last names of employees and their respective managers' first and last names. Use a self-join for this purpose.

60. Consider two tables: "Products" and "Categories" with the following columns:

"Products" Table:

- ProductID (int): Unique identifier for each product.
- ProductName (varchar): Name of the product.
- CategoryID (int): Identifier for the product category.

"Categories" Table:

- CategoryID (int): Unique identifier for each category.
- CategoryName (varchar): Name of the category.

Write an SQL query to retrieve the product names and their associated category names, sorted first by category name in ascending order and then by product name in ascending order within each category.

61. Consider three tables: "Customers," "Orders," and "OrderDetails" with the following columns:

"Customers" Table:

- CustomerID (int): Unique identifier for each customer.
- FirstName (varchar): First name of the customer.
- LastName (varchar): Last name of the customer.

"Orders" Table:

- OrderID (int): Unique identifier for each order.
- CustomerID (int): Identifier for the customer who placed the order.
- OrderDate (date): Date of the order.

"OrderDetails" Table:

- OrderDetailID (int): Unique identifier for each order detail.
- OrderID (int): Identifier for the order.
- ProductID (int): Identifier for the product.
- Quantity (int): Quantity of the product ordered.

Write an SQL query to retrieve the first and last names of customers, along with the order dates and product names for their orders.

62-a. Consider three tables: "Employees," "Teams," and "TeamMembers" with the following columns:

"Employees" Table:

- EmployeeID (int): Unique identifier for each employee.
- FirstName (varchar): First name of the employee.
- LastName (varchar): Last name of the employee.

"Teams" Table:

- TeamID (int): Unique identifier for each team.
- TeamName (varchar): Name of the team.

"TeamMembers" Table:

- TeamMemberID (int): Unique identifier for each team member.
- EmployeeID (int): Identifier for the employee.
- TeamID (int): Identifier for the team.

Write an SQL query to retrieve the first and last names of employees, along with the team names of the teams they manage.

62-b. Consider three tables: "Products," "Orders," and "OrderDetails" with the following columns:

"Products" Table:

- ProductID (int): Unique identifier for each product.
- CategoryID (int): Identifier for the product category.

"Orders" Table:

- OrderID (int): Unique identifier for each order.
- OrderDate (date): Date of the order.

"OrderDetails" Table:

- OrderDetailID (int): Unique identifier for each order detail.
- OrderID (int): Identifier for the order.
- ProductID (int): Identifier for the product.
- Quantity (int): Quantity of the product ordered.
- UnitPrice (decimal): Price per unit of the product.

Write an SQL query to find the total sales amount for each product category and year. Group the results by both the product category and the year of the order date.

63. Consider a table named "Employees" with the following columns:

"Employees" Table:

- EmployeeID (int): Unique identifier for each employee.
- FirstName (varchar): First name of the employee.
- LastName (varchar): Last name of the employee.
- DepartmentID (int): Identifier for the department the employee belongs to.
- JobTitle (varchar): Job title of the employee.
- Salary (decimal): The salary of the employee.

Write an SQL query to find the average salary for each combination of department and job title. Group the results by both the department and job title.

64. Consider two tables: "Customers" and "Orders" with the following columns:

"Customers" Table:

- CustomerID (int): Unique identifier for each customer.
- FirstName (varchar): First name of the customer.
- LastName (varchar): Last name of the customer.

"Orders" Table:

- OrderID (int): Unique identifier for each order.
- CustomerID (int): Identifier for the customer who placed the order.
- OrderDate (date): Date of the order.

Write an SQL query to retrieve the first and last names of all customers, along with their associated order dates, if they have placed any orders. Include customers who have not placed any orders in the result.

65. Consider two tables: "Products" and "Categories" with the following columns:

"Products" Table:

- ProductID (int): Unique identifier for each product.
- ProductName (varchar): Name of the product.
- CategoryID (int): Identifier for the product category.

"Categories" Table:

- CategoryID (int): Unique identifier for each category.
- CategoryName (varchar): Name of the category.

Write an SQL query to retrieve the product names and their associated category names, including categories that have no products associated with them.

66. Consider two tables: "Employees" and "Projects" with the following columns:

"Employees" Table:

- EmployeeID (int): Unique identifier for each employee.
- FirstName (varchar): First name of the employee.
- LastName (varchar): Last name of the employee.

"Projects" Table:

- ProjectID (int): Unique identifier for each project.
- ProjectName (varchar): Name of the project.
- EmployeeID (int): Identifier for the employee assigned to the project.

Write an SQL query to retrieve the first and last names of all employees, along with the names of the projects they are assigned to. Include employees who are not assigned to any projects in the result.

67. Consider two tables: "Employees" and "Teams" with the following columns:

"Employees" Table:

- EmployeeID (int): Unique identifier for each employee.
- FirstName (varchar): First name of the employee.
- LastName (varchar): Last name of the employee.
- ManagerID (int): Identifier for the employee's manager.

"Teams" Table:

- TeamID (int): Unique identifier for each team.
- TeamName (varchar): Name of the team.

Write an SQL query to retrieve the first and last names of all employees, along with the names of the teams they manage, if they manage any teams. Include employees who do not manage any teams in the result.

68. Consider two tables: "Customers" and "Orders" with the following columns:

"Customers" Table:

- CustomerID (int): Unique identifier for each customer.
- FirstName (varchar): First name of the customer.
- LastName (varchar): Last name of the customer.

"Orders" Table:

- OrderID (int): Unique identifier for each order.
- CustomerID (int): Identifier for the customer who placed the order.
- OrderDate (date): Date of the order.

Write an SQL query to retrieve the first and last names of all customers, along with their associated order dates, if they have placed any orders. Include customers who have not placed any orders and orders that have no associated customers in the result.

69. Consider two tables: "Employees" and "Teams" with the following columns:

"Employees" Table:

- EmployeeID (int): Unique identifier for each employee.
- FirstName (varchar): First name of the employee.
- LastName (varchar): Last name of the employee.
- ManagerID (int): Identifier for the employee's manager.

"Teams" Table:

- TeamID (int): Unique identifier for each team.
- TeamName (varchar): Name of the team.

Write an SQL query to retrieve the first and last names of all employees, along with the names of the teams they manage, if they manage any teams. Include employees who do not manage any teams and teams with no associated managers in the result.

70. Consider two tables: "Customers" and "Orders" with the following columns:

"Customers" Table:

- CustomerID (int): Unique identifier for each customer.
- FirstName (varchar): First name of the customer.
- LastName (varchar): Last name of the customer.

"Orders" Table:

- OrderID (int): Unique identifier for each order.
- CustomerID (int): Identifier for the customer who placed the order.
- OrderDate (date): Date of the order.

Write an SQL query to retrieve the first and last names of all customers, along with their associated order dates, if they have placed any orders. Additionally, retrieve the order IDs and order dates for orders that have no associated customers and the customer IDs and first names for customers who have not placed any orders.

71. Consider two tables: "Customers" and "Orders" with the following columns:

"Customers" Table:

- CustomerID (int): Unique identifier for each customer.
- FirstName (varchar): First name of the customer.
- LastName (varchar): Last name of the customer.

"Orders" Table:

- OrderID (int): Unique identifier for each order.
- CustomerID (int): Identifier for the customer who placed the order.
- OrderDate (date): Date of the order.

Write an SQL query to retrieve the first and last names of customers who have placed at least one order.

72. Consider two tables: "Products" and "OrderDetails" with the following columns:

"Products" Table:

- ProductID (int): Unique identifier for each product.
- ProductName (varchar): Name of the product.

"OrderDetails" Table:

- OrderDetailID (int): Unique identifier for each order detail.
- ProductID (int): Identifier for the product in the order.
- Quantity (int): Quantity of the product ordered.

Write an SQL query to retrieve the names of products that have never been ordered.

73. Consider two tables: "Customers" and "Suppliers" with the following columns:

"Customers" Table:

- CustomerID (int): Unique identifier for each customer.
- CustomerName (varchar): Name of the customer.

"Suppliers" Table:

- SupplierID (int): Unique identifier for each supplier.
- SupplierName (varchar): Name of the supplier.

Write an SQL query to retrieve a list of unique names that includes both customer names and supplier names.

74. Consider a table named "Products" with the following columns:

"Products" Table:

- ProductID (int): Unique identifier for each product.
- ProductName (varchar): Name of the product.
- CategoryID (int): Identifier for the product category.

Write an SQL query to find the product names that belong to both category 1 and category 2.

75. Consider a "Products" table with the following columns:

"Products" Table:

- ProductID (int): Unique identifier for each product.
- ProductName (varchar): Name of the product.
- Price (decimal): Price of the product.
- CategoryID (int): Identifier for the product category.

Write an SQL query to increase the prices of all products in category 1 by 15%.

76. Consider a "Customers" table with the following columns:

"Customers" Table:

- CustomerID (int): Unique identifier for each customer.
- FirstName (varchar): First name of the customer.
- LastName (varchar): Last name of the customer.
- Email (varchar): Email address of the customer (optional).

Write an SQL query to insert two new customers into the "Customers" table using both the VALUES and DEFAULTS clauses. One customer should have all values specified explicitly, and the other customer should use the DEFAULT value for the "Email" column.

77. Consider two tables: "Customers" and "Orders" with the following columns:

"Customers" Table:

- CustomerID (int): Unique identifier for each customer.
- FirstName (varchar): First name of the customer.
- LastName (varchar): Last name of the customer.

"Orders" Table:

- OrderID (int): Unique identifier for each order.
- CustomerID (int): Identifier for the customer who placed the order.
- OrderDate (date): Date of the order.

Write an SQL query to copy all orders placed by a specific customer (e.g., CustomerID = 1) into the "Orders" table for a different customer (e.g., CustomerID = 2) using INSERT... SELECT.

78. Consider a "Products" table with the following columns:

"Products" Table:

- ProductID (int): Unique identifier for each product.
- ProductName (varchar): Name of the product.
- Price (decimal): Price of the product.

Write an SQL query to increase the price of all products by 10%.

79. Consider the same "Products" table as before. Write an SQL query to append "(New)" to the end of the product names for all products.

80. Consider an "Orders" table with the following columns:

"Orders" Table:

- OrderID (int): Unique identifier for each order.
- OrderDate (date): Date of the order.

Write an SQL query to delete all rows from the "Orders" table.

81. Consider a "Products" table with the following columns:

"Products" Table:

- ProductID (int): Unique identifier for each product.
- ProductName (varchar): Name of the product.
- IsActive (bit): Indicates whether the product is active (1) or inactive (0).

Write an SQL query to delete all inactive products (products with IsActive = 0).

82. Consider an "Orders" table with the following columns:

"Orders" Table:

- OrderID (int): Unique identifier for each order.

- OrderDate (date): Date of the order.

Write an SQL query to delete all orders placed before January 1, 2023.

83. Consider a "Products" table with the following columns:

"Products" Table:

- ProductID (int): Unique identifier for each product.
- ProductName (varchar): Name of the product.
- Price (decimal): Price of the product.
- StockQuantity (int): Quantity of the product in stock.

Write an SQL query to increase the price by 5% and decrease the stock quantity by 10% for all products.

84. Consider a "Customers" table with the following columns:

"Customers" Table:

- CustomerID (int): Unique identifier for each customer.
- FirstName (varchar): First name of the customer.
- LastName (varchar): Last name of the customer.
- Email (varchar): Email address of the customer.

Write an SQL query to update the customer's first name and email address for all customers.

85. Consider a "Customers" table with the following columns:

"Customers" Table:

- CustomerID (int): Unique identifier for each customer.
- FirstName (varchar): First name of the customer.
- LastName (varchar): Last name of the customer.
- Email (varchar): Email address of the customer.
- City (varchar): City where the customer resides.

Write an SQL query to update the email addresses of customers who live in "New York City" to use a new domain, changing "@example.com" to "@nycmail.com."

86. Consider an "Employees" table with the following columns:

"Employees" Table:

- EmployeeID (int): Unique identifier for each employee.
- FirstName (varchar): First name of the employee.
- LastName (varchar): Last name of the employee.
- Salary (decimal): Salary of the employee.
- YearsOfExperience (int): Number of years of experience.

Write an SQL query to increase the salaries of employees with more than 5 years of experience by 10%.

87. Consider a "Products" table with the following columns:

"Products" Table:

- ProductID (int): Unique identifier for each product.
- ProductName (varchar): Name of the product.
- Description (text): Description of the product.
- Brand (varchar): Brand name of the product.

Write an SQL query to update the product descriptions for all products with the brand name "ABC" to include additional information.

88. Consider a "Customers" table with the following columns:

"Customers" Table:

- CustomerID (int): Unique identifier for each customer.
- FirstName (varchar): First name of the customer.
- LastName (varchar): Last name of the customer.
- DiscountPercentage (decimal): Discount percentage for the customer's purchases.
- TotalPurchases (decimal): Total amount spent by the customer.

Write an SQL query to update the discount percentages for customers who have made total purchases exceeding $1,000. Set their discount percentage to 10%.