SQL Table Creation and Data Insertion:

```SQL
-- Create the Employee table:
    ➢  CREATE TABLE Employee (
EmployeeID INT PRIMARY KEY,
FirstName VARCHAR(50),
LastName VARCHAR(50),
Department VARCHAR(50),
Salary DECIMAL(10, 2)
);


-- Insert 20 sample records into the Employee table:
    ➢  INSERT INTO Employee (EmployeeID, FirstName, LastName, Department, Salary)
VALUES
    (1, 'John', 'Doe', 'HR', 50000.00),
    (2, 'Jane', 'Smith', 'IT', 60000.00),
    (3, 'Michael', 'Johnson', 'Finance', 55000.00),
    (4, 'Emily', 'Brown', 'Marketing', 52000.00),
    (5, 'David', 'Wilson', 'Sales', 55000.00),
    (6, 'Sarah', 'Lee', 'IT', 62000.00),
    (7, 'James', 'Anderson', 'Finance', 58000.00),
    (8, 'Olivia', 'Martinez', 'HR', 52000.00),
    (9, 'Daniel', 'Taylor', 'Sales', 56000.00),
    (10, 'Sophia', 'Jackson', 'Marketing', 53000.00),
    (11, 'Liam', 'Harris', 'IT', 61000.00),
    (12, 'Ava', 'White', 'Finance', 57000.00),
    (13, 'William', 'Miller', 'HR', 51000.00),
    (14, 'Ella', 'Thompson', 'Marketing', 54000.00),
    (15, 'Alexander', 'Walker', 'Sales', 57000.00),
    (16, 'Mia', 'Clark', 'IT', 59000.00),
```

(17, 'Henry', 'Lewis', 'Finance', 59000.00),

(18, 'Sofia', 'Hall', 'HR', 53000.00),

(19, 'Jackson', 'Young', 'Sales', 58000.00),

(20, 'Luna', 'Wright', 'Marketing', 55000.00);
```


**General Questions And Answers:**

1. **What is the structure of the Employee table?**
   - ❖ **DESCRIBE** in SQL is a command that explains the structure of the table. Actual databases are structured, programmed, and used by a group of people. Such scenarios describe functionality that makes it easier for multiple people to access the database.

   - ❖ The **DESCRIBE** command will show all the details**,**
     - Columns present in the table
     - Each column's current value types such as **VARCHAR, CHAR, INT, FLOAT, DATE,** ETC.
     - Default value of columns where exists
     - **NULL** OR **NOT NULL** values that a column can contain.

   - ❖ **SYNTAX** FOR GETTING THE STRUCTURE OF THE EMPLOYEE TABLE:
     - ➢ **DESC/DESCRIBE Employee;**

**OUTPUT:**

- [Table structure](Table structure)

| | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action | | |
|---|---|------|------|-----------|-----------|------|---------|----------|-------|--------|---|---|
| ☐ | 1 | EmployeeID | int(11) | | | No | *None* | | | Change | Drop | More |
| ☐ | 2 | FirstName | varchar(50) | latin1_swedish_ci | | Yes | *NULL* | | | Change | Drop | More |
| ☐ | 3 | LastName | varchar(50) | latin1_swedish_ci | | Yes | *NULL* | | | Change | Drop | More |

| | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action | | |
|---|---|------|------|-----------|------------|------|---------|----------|-------|--------|---|---|
| ☐ | 4 | Department | varchar(50) | latin1_swedish_ci | | Yes | *NULL* | | | Change | Drop | More |
| ☐ | 5 | Salary | decimal(10,2) | | | Yes | *NULL* | | | Change | Drop | More |

**2. How many columns are there in the Employee table?**

❖ To find that we can use a simple function COUNT() with INFORMATION_SCHEMA view in a relational database. This INFORMATION_SCHEMA is an ANSI standard set of views that provides read-only access to the details of databases and their objects like tables, constraints, procedures, etc. See the below query:
❖
  ➢ SELECT count(*) as No_of_Column FROM information_schema.columns WHERE table_name ='Employee';

❖ Here, **COUNT(*)** counts the number of columns returned by the INFORMATION_SCHEMA .columns one by one and provides the final count of the columns. Here table_name selects the table in which we wish to work.

**OUTPUT:**

**No_of_Column**

   5

**3. What is the primary key of the Employee table?**

  ➢ **To get the primary key of a table in SQL, you can use the following command:**

**SELECT column_name**

**FROM information_schema.key_column_usage**

**WHERE table_name = 'Employee' AND constraint_name = 'PRIMARY';**

> **OUTPUT:**

> column_name

> **EmployeeID**

❖ **EmployeeID** is the Primary Key of the Employee table.

4. **Can you retrieve all the records from the Employee table?**
❖ To retrieve all the records from the Employee table we will use the below syntax:

> **SELECT * FROM Employee;**

▪ **OUTPUT:**

| ←T→ | | | | EmployeeID | FirstName | LastName | Department | Salary |
|---|---|---|---|---|---|---|---|---|
| ☐ | Edit | Copy | Delete | 1 | John | Doe | HR | 50000.00 |
| ☐ | Edit | Copy | Delete | 2 | Jane | Smith | IT | 60000.00 |
| ☐ | Edit | Copy | Delete | 3 | Michael | Johnson | Finance | 55000.00 |
| ☐ | Edit | Copy | Delete | 4 | Emily | Brown | Marketing | 52000.00 |
| ☐ | Edit | Copy | Delete | 5 | David | Wilson | Sales | 55000.00 |
| ☐ | Edit | Copy | Delete | 6 | Sarah | Lee | IT | 62000.00 |
| ☐ | Edit | Copy | Delete | 7 | James | Anderson | Finance | 58000.00 |
| ☐ | Edit | Copy | Delete | 8 | Olivia | Martinez | HR | 52000.00 |
| ☐ | Edit | Copy | Delete | 9 | Daniel | Taylor | Sales | 56000.00 |
| ☐ | Edit | Copy | Delete | 10 | Sophia | Jackson | Marketing | 53000.00 |
| ☐ | Edit | Copy | Delete | 11 | Liam | Harris | IT | 61000.00 |
| ☐ | Edit | Copy | Delete | 12 | Ava | White | Finance | 57000.00 |
| ☐ | Edit | Copy | Delete | 13 | William | Miller | HR | 51000.00 |

| ←T→ | | | | EmployeeID | FirstName | LastName | Department | Salary |
|---|---|---|---|---|---|---|---|---|
| ☐ | Edit | Copy | Delete | 14 | Ella | Thompson | Marketing | 54000.00 |
| ☐ | Edit | Copy | Delete | 15 | Alexander | Walker | Sales | 57000.00 |
| ☐ | Edit | Copy | Delete | 16 | Mia | Clark | IT | 59000.00 |
| ☐ | Edit | Copy | Delete | 17 | Henry | Lewis | Finance | 59000.00 |
| ☐ | Edit | Copy | Delete | 18 | Sofia | Hall | HR | 53000.00 |
| ☐ | Edit | Copy | Delete | 19 | Jackson | Young | Sales | 58000.00 |
| ☐ | Edit | Copy | Delete | 20 | Luna | Wright | Marketing | 55000.00 |

5. **How many records (rows) are there in the Employee table?**

❖ **COUNT(*)** counts the total number of rows in the table:

❖ **SYNTAX:**

**SELECT COUNT(*) as count_record
FROM Employee;**

- **OUTPUT:**

**count_record**
     20

**Data Retrieval:**

6. **Show me the first 5 records from the Employee table.**

➢ To get the first five records from a table in MySQL, you can use the **LIMIT** clause in your SQL query. Here's an example:
➢ **SYNTAX:**
➢ **SELECT * FROM EMPLOYEE
LIMIT 5;**

- **OUTPUT:**

| EmployeeID | FirstName | LastName | Department | Salary |
|---|---|---|---|---|
| 1 | John | Doe | HR | 50000.00 |
| 2 | Jane | Smith | IT | 60000.00 |
| 3 | Michael | Johnson | Finance | 55000.00 |
| 4 | Emily | Brown | Marketing | 52000.00 |
| 5 | David | Wilson | Sales | 55000.00 |

7. **Display the names of all employees in the Marketing department.**

➢ **SELECT FirstName, LastName FROM employee
WHERE Department = 'Marketing';**

▪ **OUTPUT:**

| ←T→ | | | FirstName | LastName |
|---|---|---|---|---|
| ☐  Edit  Copy  Delete | | | Emily | Brown |
| ☐  Edit  Copy  Delete | | | Sophia | Jackson |
| ☐  Edit  Copy  Delete | | | Ella | Thompson |
| ☐  Edit  Copy  Delete | | | Luna | Wright |

8. **Retrieve the highest salary in the Employee table.**

➢ **SELECT MAX(Salary) FROM Employee;**

• **OUTPUT:**

**MAX(Salary)**

  **62000.00**

9. **List the employees whose salary is above $55,000.**

- ➤ **SELECT * FROM Employee**
  **WHERE Salary > 55000;**
- ▪ **This query will list all the employees from the Employee table whose salary is above $55,000.**

- ▪ **OUTPUT:**

| ←T→ | | | | EmployeeID | FirstName | LastName | Department | Salary |
|---|---|---|---|---|---|---|---|---|
| ☐ | Edit | Copy | Delete | 2 | Jane | Smith | IT | 60000.00 |
| ☐ | Edit | Copy | Delete | 6 | Sarah | Lee | IT | 62000.00 |
| ☐ | Edit | Copy | Delete | 7 | James | Anderson | Finance | 58000.00 |
| ☐ | Edit | Copy | Delete | 9 | Daniel | Taylor | Sales | 56000.00 |
| ☐ | Edit | Copy | Delete | 11 | Liam | Harris | IT | 61000.00 |
| ☐ | Edit | Copy | Delete | 12 | Ava | White | Finance | 57000.00 |
| ☐ | Edit | Copy | Delete | 15 | Alexander | Walker | Sales | 57000.00 |
| ☐ | Edit | Copy | Delete | 16 | Mia | Clark | IT | 59000.00 |
| ☐ | Edit | Copy | Delete | 17 | Henry | Lewis | Finance | 59000.00 |
| ☐ | Edit | Copy | Delete | 19 | Jackson | Young | Sales | 58000.0 |

10. **Who has the lowest salary in the IT department?**
- ➤ **SELECT MIN(Salary) FROM Employee;**
- ➤ **OUTPUT:**

| MIN(Salary) |
|---|
| **50000.00** |

11. **Find the employee with EmployeeID 7.**
   - ➤ **SELECT * FROM Employee WHERE EmployeeID = 7;**

**OUTPUT:**

| ←T→ | | | EmployeeID | FirstName | LastName | Department | Salary |
|---|---|---|---|---|---|---|---|
| ☐ | Edit | Copy | Delete | 7 | James | Anderson | Finance | 58000.00 |

**Data Modification:**

12. Update the salary of John Doe to $52,000.
    - ❖ You can update the salary of an employee in a MySQL database by using the UPDATE statement. Here is an example of how you can do it:
    - ❖
    - ❖ UPDATE Employee
    - ❖ SET salary = 52000
    - ❖ WHERE name = 'John Doe';
    - ❖ This SQL command will set the salary field to 52000 for all records in the Employee table where the name is John Doe.
    - ❖ UPDATED Employee Table OUTPUT:

| ←T→ | | | | EmployeeID | FirstName | LastName | Department | Salary |
|---|---|---|---|---|---|---|---|---|
| ☐ | Edit | Copy | Delete | 1 | **John** | **Doe** | **HR** | **52000**.00 |
| ☐ | Edit | Copy | Delete | 2 | Jane | Smith | IT | 60000.00 |
| ☐ | Edit | Copy | Delete | 3 | Michael | Johnson | Finance | 55000.00 |
| ☐ | Edit | Copy | Delete | 4 | Emily | Brown | Marketing | 52000.00 |
| ☐ | Edit | Copy | Delete | 5 | David | Wilson | Sales | 55000.00 |
| ☐ | Edit | Copy | Delete | 6 | Sarah | Lee | IT | 62000.00 |
| ☐ | Edit | Copy | Delete | 7 | James | Anderson | Finance | 58000.00 |
| ☐ | Edit | Copy | Delete | 8 | Olivia | Martinez | HR | 52000.00 |
| ☐ | Edit | Copy | Delete | 9 | Daniel | Taylor | Sales | 56000.00 |
| ☐ | Edit | Copy | Delete | 10 | Sophia | Jackson | Marketing | 53000.00 |
| ☐ | Edit | Copy | Delete | 11 | Liam | Harris | IT | 61000.00 |
| ☐ | Edit | Copy | Delete | 12 | Ava | White | Finance | 57000.00 |
| ☐ | Edit | Copy | Delete | 13 | William | Miller | HR | 51000.00 |
| ☐ | Edit | Copy | Delete | 14 | Ella | Thompson | Marketing | 54000.00 |
| ☐ | Edit | Copy | Delete | 15 | Alexander | Walker | Sales | 57000.00 |
| ☐ | Edit | Copy | Delete | 16 | Mia | Clark | IT | 59000.00 |
| ☐ | Edit | Copy | Delete | 17 | Henry | Lewis | Finance | 59000.00 |
| ☐ | Edit | Copy | Delete | 18 | Sofia | Hall | HR | 53000.00 |
| ☐ | Edit | Copy | Delete | 19 | Jackson | Young | Sales | 58000.00 |
| ☐ | Edit | Copy | Delete | 20 | Luna | Wright | Marketing | 55000.00 |

13. **Insert a new employee with EmployeeID 21, FirstName 'Isabella,' LastName 'Davis,' Department 'Sales,' and Salary $57,000.**
    - ❖ **You can insert a new employee record into a MySQL database by using the INSERT INTO statement. Here is an example of how you can do it:**
    - ❖
    - ❖
    - ❖ **INSERT INTO Employee (EmployeeID, FirstName, LastName, Department, Salary)**
    - ❖ **VALUES (21, 'Isabella', 'Davis', 'Sales', 57000.00);**
    - ❖ **This SQL command will insert a new record into the Employee table with the EmployeeID as 21, FirstName as 'Isabella', LastName as 'Davis', Department as 'Sales', and Salary as 57000.**
    - ❖ **UPDATED Employee Table:**
    - ❖

| ←T→ | | | | EmployeeID | FirstName | LastName | Department | Salary |
|---|---|---|---|---|---|---|---|---|
| ☐ | Edit | Copy | Delete | 1 | John | Doe | HR | 52000.00 |
| ☐ | Edit | Copy | Delete | 2 | Jane | Smith | IT | 60000.00 |
| ☐ | Edit | Copy | Delete | 3 | Michael | Johnson | Finance | 55000.00 |
| ☐ | Edit | Copy | Delete | 4 | Emily | Brown | Marketing | 52000.00 |
| ☐ | Edit | Copy | Delete | 5 | David | Wilson | Sales | 55000.00 |
| ☐ | Edit | Copy | Delete | 6 | Sarah | Lee | IT | 62000.00 |
| ☐ | Edit | Copy | Delete | 7 | James | Anderson | Finance | 58000.00 |
| ☐ | Edit | Copy | Delete | 8 | Olivia | Martinez | HR | 52000.00 |
| ☐ | Edit | Copy | Delete | 9 | Daniel | Taylor | Sales | 56000.00 |
| ☐ | Edit | Copy | Delete | 10 | Sophia | Jackson | Marketing | 53000.00 |
| ☐ | Edit | Copy | Delete | 11 | Liam | Harris | IT | 61000.00 |
| ☐ | Edit | Copy | Delete | 12 | Ava | White | Finance | 57000.00 |
| ☐ | Edit | Copy | Delete | 13 | William | Miller | HR | 51000.00 |
| ☐ | Edit | Copy | Delete | 14 | Ella | Thompson | Marketing | 54000.00 |
| ☐ | Edit | Copy | Delete | 15 | Alexander | Walker | Sales | 57000.00 |
| ☐ | Edit | Copy | Delete | 16 | Mia | Clark | IT | 59000.00 |
| ☐ | Edit | Copy | Delete | 17 | Henry | Lewis | Finance | 59000.00 |
| ☐ | Edit | Copy | Delete | 18 | Sofia | Hall | HR | 53000.00 |

| ←T→ | | | | EmployeeID | FirstName | LastName | Department | Salary |
|---|---|---|---|---|---|---|---|---|
| ☐ | Edit | Copy | Delete | 19 | Jackson | Young | Sales | 58000.00 |
| ☐ | Edit | Copy | Delete | 20 | Luna | Wright | Marketing | 55000.00 |
| ☐ | Edit | Copy | Delete | **21** | **Isabella** | **Davis** | **Sales** | **57000.00** |

14. **Delete the record of employee with EmployeeID 9.**
   - ❖ **You can delete a record of an employee in a MySQL database by using the DELETE FROM statement. Here is an example of how you can do it:**
   - ❖
   - ❖ **DELETE FROM Employee**
   - ❖ **WHERE EmployeeID = 9;**
   - ❖ **This SQL command will delete the record from the Employee table where the EmployeeID is 9.**
   - ❖ **UPDATED Employee Table OUTPUT:**

| ←T→ | | | | EmployeeID | FirstName | LastName | Department | Salary |
|---|---|---|---|---|---|---|---|---|
| ☐ | Edit | Copy | Delete | 1 | John | Doe | HR | 52000.00 |
| ☐ | Edit | Copy | Delete | 2 | Jane | Smith | IT | 60000.00 |
| ☐ | Edit | Copy | Delete | 3 | Michael | Johnson | Finance | 55000.00 |
| ☐ | Edit | Copy | Delete | 4 | Emily | Brown | Marketing | 52000.00 |
| ☐ | Edit | Copy | Delete | 5 | David | Wilson | Sales | 55000.00 |
| ☐ | Edit | Copy | Delete | 6 | Sarah | Lee | IT | 62000.00 |
| ☐ | Edit | Copy | Delete | 7 | James | Anderson | Finance | 58000.00 |
| ☐ | Edit | Copy | Delete | **8** | **Olivia** | **Martinez** | **HR** | **52000.00** |
| ☐ | Edit | Copy | Delete | **10** | **Sophia** | **Jackson** | **Marketing** | **53000.00** |
| ☐ | Edit | Copy | Delete | 11 | Liam | Harris | IT | 61000.00 |
| ☐ | Edit | Copy | Delete | 12 | Ava | White | Finance | 57000.00 |
| ☐ | Edit | Copy | Delete | 13 | William | Miller | HR | 51000.00 |
| ☐ | Edit | Copy | Delete | 14 | Ella | Thompson | Marketing | 54000.00 |
| ☐ | Edit | Copy | Delete | 15 | Alexander | Walker | Sales | 57000.00 |
| ☐ | Edit | Copy | Delete | 16 | Mia | Clark | IT | 59000.00 |
| ☐ | Edit | Copy | Delete | 17 | Henry | Lewis | Finance | 59000.00 |
| ☐ | Edit | Copy | Delete | 18 | Sofia | Hall | HR | 53000.00 |
| ☐ | Edit | Copy | Delete | 19 | Jackson | Young | Sales | 58000.00 |

| ←T→ | | | | EmployeeID | FirstName | LastName | Department | Salary |
|---|---|---|---|---|---|---|---|---|
| ☐ | Edit | Copy | Delete | 20 | Luna | Wright | Marketing | 55000.00 |
| ☐ | Edit | Copy | Delete | 21 | Isabella | Davis | Sales | 57000.00 |

15. **Change the department of employee with EmployeeID 14 to 'Finance.'**
   - ❖ **You can update the department of an employee in a MySQL database by using the UPDATE statement. Here is an example of how you can do it:**
   - ❖
   - ❖ **UPDATE Employee**
   - ❖ **SET Department = 'Finance'**
   - ❖ **WHERE EmployeeID = 14;**
   - ❖ **This SQL command will set the Department field to 'Finance' for the record in the Employee table where the EmployeeID is 14.**
   - ❖ **UPDATED Employee Table OUTPUT:**
   - ❖

| ←T→ | | | | EmployeeID | FirstName | LastName | Department | Salary |
|---|---|---|---|---|---|---|---|---|
| ☐ | Edit | Copy | Delete | 1 | John | Doe | HR | 52000.00 |
| ☐ | Edit | Copy | Delete | 2 | Jane | Smith | IT | 60000.00 |
| ☐ | Edit | Copy | Delete | 3 | Michael | Johnson | Finance | 55000.00 |
| ☐ | Edit | Copy | Delete | 4 | Emily | Brown | Marketing | 52000.00 |
| ☐ | Edit | Copy | Delete | 5 | David | Wilson | Sales | 55000.00 |
| ☐ | Edit | Copy | Delete | 6 | Sarah | Lee | IT | 62000.00 |
| ☐ | Edit | Copy | Delete | 7 | James | Anderson | Finance | 58000.00 |
| ☐ | Edit | Copy | Delete | 8 | Olivia | Martinez | HR | 52000.00 |
| ☐ | Edit | Copy | Delete | 10 | Sophia | Jackson | Marketing | 53000.00 |
| ☐ | Edit | Copy | Delete | 11 | Liam | Harris | IT | 61000.00 |
| ☐ | Edit | Copy | Delete | 12 | Ava | White | Finance | 57000.00 |
| ☐ | Edit | Copy | Delete | 13 | William | Miller | HR | 51000.00 |
| ☐ | Edit | Copy | Delete | **14** | **Ella** | **Thompson** | **Finance** | **54000.00** |
| ☐ | Edit | Copy | Delete | 15 | Alexander | Walker | Sales | 57000.00 |
| ☐ | Edit | Copy | Delete | 16 | Mia | Clark | IT | 59000.00 |

| | | | | EmployeeID | FirstName | LastName | Department | Salary |
|---|---|---|---|---|---|---|---|---|
| ☐ | Edit | Copy | Delete | 17 | Henry | Lewis | Finance | 59000.00 |
| ☐ | Edit | Copy | Delete | 18 | Sofia | Hall | HR | 53000.00 |
| ☐ | Edit | Copy | Delete | 19 | Jackson | Young | Sales | 58000.00 |
| ☐ | Edit | Copy | Delete | 20 | Luna | Wright | Marketing | 55000.00 |
| ☐ | Edit | Copy | Delete | 21 | Isabella | Davis | Sales | 57000.00 |

**Aggregation and Statistics:**

16. **Calculate the average salary of all employees.**
    - ❖ You can calculate the average salary of all employees in a MySQL database by using the AVG() function. Here is an example of how you can do it:
    - ❖
    - ❖
    - ❖ SELECT AVG(Salary) AS AverageSalary
    - ❖ FROM Employee;
    - ❖ This SQL command will calculate the average of the Salary field for all records in the Employee table and display it as AverageSalary.
    - ❖ SQL OUTPUT:

| AverageSalary |
|---|
| 56000.000000 |

17. **What is the total salary expenditure for the company?**
    - ❖ You can calculate the total salary expenditure for the company in a MySQL database by using the SUM() function. Here is an example of how you can do it:
    - ❖
    - ❖
    - ❖ SELECT SUM(Salary) AS TotalSalaryExpenditure
    - ❖ FROM Employee;
    - ❖ This SQL command will calculate the sum of the Salary field for all records in the Employee table and display it as TotalSalaryExpenditure.
    - ❖ SQL OUTPUT FOR THE SYNTAX:

| TotalSalaryExpenditure |
|---|
| 1120000.00 |

18. **Find the department with the highest average salary.**

- ❖ **You can find the department with the highest average salary in a MySQL database by using the AVG() function along with GROUP BY and ORDER BY clauses. Here is an example of how you can do it:**
- ❖
- ❖
- ❖ **SELECT Department, AVG(Salary) AS AverageSalary**
  **FROM Employee**
  **GROUP BY Department**
  **ORDER BY AverageSalary DESC**
  **LIMIT 1;**
  **This SQL command will calculate the average of the Salary field for each Department in the Employee table, sort them in descending order of the average salary, and display the department with the highest average salary.**

     **SQL OUTPUT FOR THE SYNTAX:**

| ←T→ | | | | Department | AverageSalary |
|---|---|---|---|---|---|
| ☐ | Edit | Copy | Delete | IT | 60500.000000 |

19. **Count the number of employees in each department.**
    - ❖ **You can count the number of employees in each department in a MySQL database by using the COUNT() function along with the GROUP BY clause. Here is an example of how you can do it:**
    - ❖
      **SELECT Department, COUNT(*) AS NumberOfEmployees**
      **FROM Employee**
      **GROUP BY Department;**
    - ❖ **This SQL command will count the number of records (employees) for each Department in the Employee table and display it as NumberOfEmployees.**
    - ❖ **SQL OUTPUT FOR THE SYNTAX:**

| Department | NumberOfEmployees |
|---|---|
| Finance | 5 |
| HR | 4 |
| IT | 4 |
| Marketing | 3 |
| Sales | 4 |

20. **What is the sum of salaries for employees in the IT department?**
    - ❖ **You can calculate the sum of salaries for employees in the IT department in a MySQL database by using the SUM() function along with a WHERE clause. Here is an example of how you can do it:**

❖
❖
  **SELECT SUM(Salary) AS TotalSalary**
  **FROM Employee**
  **WHERE Department = 'IT';**
❖ **This SQL command will calculate the sum of the Salary field for all records in the Employee table where the Department is 'IT', and display it as TotalSalary.**
❖ **SQL OUTPUT FOR THE SYNTAX:**

| TotalSalary |
|---|
| 242000.00 |

**\*\*Sorting and Filtering:\*\***

21. **List employees in ascending order of their salaries.**
   ❖ **you can list employees in ascending order of their salaries in a MySQL database by using the ORDER BY clause. Here is an example of how you can do it:**
   ❖
     **SELECT \***
     **FROM Employee**
     **ORDER BY Salary ASC;**
   ❖ **This SQL command will select all records from the Employee table and order them in ascending order of the Salary.**
   ❖ **SQL OUTPUT FOR THE SYNTAX:**
   ❖

| ←T→ | | | | EmployeeID | FirstName | LastName | Department | Salary_1 |
|---|---|---|---|---|---|---|---|---|
| ☐ | Edit | Copy | Delete | 13 | William | Miller | HR | 51000.00 |
| ☐ | Edit | Copy | Delete | 8 | Olivia | Martinez | HR | 52000.00 |
| ☐ | Edit | Copy | Delete | 1 | John | Doe | HR | 52000.00 |
| ☐ | Edit | Copy | Delete | 4 | Emily | Brown | Marketing | 52000.00 |
| ☐ | Edit | Copy | Delete | 10 | Sophia | Jackson | Marketing | 53000.00 |
| ☐ | Edit | Copy | Delete | 18 | Sofia | Hall | HR | 53000.00 |
| ☐ | Edit | Copy | Delete | 14 | Ella | Thompson | Finance | 54000.00 |
| ☐ | Edit | Copy | Delete | 5 | David | Wilson | Sales | 55000.00 |
| ☐ | Edit | Copy | Delete | 3 | Michael | Johnson | Finance | 55000.00 |
| ☐ | Edit | Copy | Delete | 20 | Luna | Wright | Marketing | 55000.00 |

| ←T→ | | | | EmployeeID | FirstName | LastName | Department | Salary_1 |
|---|---|---|---|---|---|---|---|---|
| ☐ | Edit | Copy | Delete | 15 | Alexander | Walker | Sales | 57000.00 |
| ☐ | Edit | Copy | Delete | 12 | Ava | White | Finance | 57000.00 |
| ☐ | Edit | Copy | Delete | 21 | Isabella | Davis | Sales | 57000.00 |
| ☐ | Edit | Copy | Delete | 7 | James | Anderson | Finance | 58000.00 |
| ☐ | Edit | Copy | Delete | 19 | Jackson | Young | Sales | 58000.00 |
| ☐ | Edit | Copy | Delete | 16 | Mia | Clark | IT | 59000.00 |
| ☐ | Edit | Copy | Delete | 17 | Henry | Lewis | Finance | 59000.00 |
| ☐ | Edit | Copy | Delete | 2 | Jane | Smith | IT | 60000.00 |
| ☐ | Edit | Copy | Delete | 11 | Liam | Harris | IT | 61000.00 |
| ☐ | Edit | Copy | Delete | 6 | Sarah | Lee | IT | 62000.00 |

22. **Show the top 3 highest-paid employees.**
   - ❖ **You can list the top 3 highest-paid employees in a MySQL database by using the ORDER BY clause along with the LIMIT clause. Here is an example of how you can do it:**
   - ❖
   - ❖
     **SELECT \***
     **FROM Employee**
     **ORDER BY Salary DESC**
     **LIMIT 3;**
   - ❖ **This SQL command will select all records from the Employee table, order them in descending order of the Salary, and limit the output to the top 3 records.**
   - ❖ **SQL OUTPUT FOR THE SYNTAX:**
   - ❖

| ←T→ | | | | EmployeeID | FirstName | LastName | Department | Salary_1 |
|---|---|---|---|---|---|---|---|---|
| ☐ | Edit | Copy | Delete | 6 | Sarah | Lee | IT | 62000.00 |
| ☐ | Edit | Copy | Delete | 11 | Liam | Harris | IT | 61000.00 |
| ☐ | Edit | Copy | Delete | 2 | Jane | Smith | IT | 60000.00 |

23. **Display employees with a salary between $50,000 and $60,000.**
   - ❖ **You can display employees with a salary between 50,000 and 60,000 in a MySQL database by using the BETWEEN keyword. Here is an example of how you can do it:**

❖

❖

**SELECT ***
**FROM Employee**
**WHERE Salary BETWEEN 50000 AND 60000;**

❖ **This SQL command will select all records from the Employee table where the Salary is between 50000 and 60000.**

❖ **SQL OUTPUT FOR THE SYNTAX:**

| ←T→ | | | | EmployeeID | FirstName | LastName | Department | Salary |
|---|---|---|---|---|---|---|---|---|
| ☐ | Edit | Copy | Delete | 1 | John | Doe | HR | 52000.00 |
| ☐ | Edit | Copy | Delete | 2 | Jane | Smith | IT | 60000.00 |
| ☐ | Edit | Copy | Delete | 3 | Michael | Johnson | Finance | 55000.00 |
| ☐ | Edit | Copy | Delete | 4 | Emily | Brown | Marketing | 52000.00 |
| ☐ | Edit | Copy | Delete | 5 | David | Wilson | Sales | 55000.00 |
| ☐ | Edit | Copy | Delete | 7 | James | Anderson | Finance | 58000.00 |
| ☐ | Edit | Copy | Delete | 8 | Olivia | Martinez | HR | 52000.00 |
| ☐ | Edit | Copy | Delete | 10 | Sophia | Jackson | Marketing | 53000.00 |
| ☐ | Edit | Copy | Delete | 12 | Ava | White | Finance | 57000.00 |
| ☐ | Edit | Copy | Delete | 13 | William | Miller | HR | 51000.00 |
| ☐ | Edit | Copy | Delete | 14 | Ella | Thompson | Finance | 54000.00 |
| ☐ | Edit | Copy | Delete | 15 | Alexander | Walker | Sales | 57000.00 |
| ☐ | Edit | Copy | Delete | 16 | Mia | Clark | IT | 59000.00 |
| ☐ | Edit | Copy | Delete | 17 | Henry | Lewis | Finance | 59000.00 |
| ☐ | Edit | Copy | Delete | 18 | Sofia | Hall | HR | 53000.00 |
| ☐ | Edit | Copy | Delete | 19 | Jackson | Young | Sales | 58000.00 |
| ☐ | Edit | Copy | Delete | 20 | Luna | Wright | Marketing | 55000.00 |
| ☐ | Edit | Copy | Delete | 21 | Isabella | Davis | Sales | 57000.00 |

24. **Sort employees by their last names in alphabetical order.**
    - ❖ **You can sort employees by their last names in alphabetical order in a MySQL database by using the ORDER BY clause. Here is an example of how you can do it:**
    - ❖
    - ❖

        **SELECT ***
        **FROM Employee**
        **ORDER BY LastName ASC;**
    - ❖ **This SQL command will select all records from the Employee table and order them in ascending order of the LastName.**
    - ❖ **SQL OUTPUT FOR THE SYNTAX:**
    - ❖

←T→

| | | | | EmployeeID | FirstName | LastName 1 | Department | Salary |
|---|---|---|---|---|---|---|---|---|
| ☐ | Edit | Copy | Delete | 7 | James | Anderson | Finance | 58000.00 |
| ☐ | Edit | Copy | Delete | 4 | Emily | Brown | Marketing | 52000.00 |
| ☐ | Edit | Copy | Delete | 16 | Mia | Clark | IT | 59000.00 |
| ☐ | Edit | Copy | Delete | 21 | Isabella | Davis | Sales | 57000.00 |
| ☐ | Edit | Copy | Delete | 1 | John | Doe | HR | 52000.00 |
| ☐ | Edit | Copy | Delete | 18 | Sofia | Hall | HR | 53000.00 |
| ☐ | Edit | Copy | Delete | 11 | Liam | Harris | IT | 61000.00 |
| ☐ | Edit | Copy | Delete | 10 | Sophia | Jackson | Marketing | 53000.00 |
| ☐ | Edit | Copy | Delete | 3 | Michael | Johnson | Finance | 55000.00 |
| ☐ | Edit | Copy | Delete | 6 | Sarah | Lee | IT | 62000.00 |
| ☐ | Edit | Copy | Delete | 17 | Henry | Lewis | Finance | 59000.00 |
| ☐ | Edit | Copy | Delete | 8 | Olivia | Martinez | HR | 52000.00 |
| ☐ | Edit | Copy | Delete | 13 | William | Miller | HR | 51000.00 |
| ☐ | Edit | Copy | Delete | 2 | Jane | Smith | IT | 60000.00 |
| ☐ | Edit | Copy | Delete | 14 | Ella | Thompson | Finance | 54000.00 |
| ☐ | Edit | Copy | Delete | 15 | Alexander | Walker | Sales | 57000.00 |
| ☐ | Edit | Copy | Delete | 12 | Ava | White | Finance | 57000.00 |
| ☐ | Edit | Copy | Delete | 5 | David | Wilson | Sales | 55000.00 |
| ☐ | Edit | Copy | Delete | 20 | Luna | Wright | Marketing | 55000.00 |

| ←T→ | | | | EmployeeID | FirstName | LastName  1 | Department | Salary |
|---|---|---|---|---|---|---|---|---|
| ☐ | Edit | Copy | Delete | 19 | Jackson | Young | Sales | 58000.00 |

25. **Find employees with names starting with 'A.'**
   - ❖ **You can find employees with names starting with 'A' in a MySQL database by using the LIKE keyword. Here is an example of how you can do it:**
   - ❖
   - ❖
     **SELECT ***
     **FROM Employee**
     **WHERE FirstName LIKE 'A%';**
   - ❖ **This SQL command will select all records from the Employee table where the FirstName starts with 'A'.**
   - ❖ **SQL OUTPUT FOR THE SYNTAX:**

| ←T→ | | | | EmployeeID | FirstName | LastName | Department | Salary |
|---|---|---|---|---|---|---|---|---|
| ☐ | Edit | Copy | Delete | 12 | Ava | White | Finance | 57000.00 |
| ☐ | Edit | Copy | Delete | 15 | Alexander | Walker | Sales | 57000.00 |

**\*\*Grouping and Aggregation:\*\***

26. **Group employees by department and show the total count in each department.**
   - ❖ **You can group employees by department and show the total count in each department in a MySQL database by using the COUNT() function along with the GROUP BY clause. Here is an example of how you can do it:**
   - ❖
   - ❖
     **SELECT Department, COUNT(*) AS NumberOfEmployees**
     **FROM Employee**
     **GROUP BY Department;**
   - ❖ **This SQL command will count the number of records (employees) for each Department in the Employee table and display it as NumberOfEmployees.**
   - ❖ **SQL OUTPUT FOR THE SYNTAX:**

| Department | NumberOfEmployees |
|---|---|
| Finance | 5 |
| HR | 4 |
| IT | 4 |
| Marketing | 3 |
| Sales | 4 |

27. **Calculate the average salary for each department.**

- ❖ You can calculate the average salary for each department in a MySQL database by using the AVG() function along with the GROUP BY clause. Here is an example of how you can do it:
- ❖
- ❖

  SELECT Department, AVG(Salary) AS AverageSalary
  FROM Employee
  GROUP BY Department;
- ❖ This SQL command will calculate the average of the Salary field for each Department in the Employee table and display it as AverageSalary.
- ❖ SQL OUTPUT FOR THE SYNTAX:

| ←T→ | | | | Department | AverageSalary |
|---|---|---|---|---|---|
| ☐ | Edit | Copy | Delete | Finance | 56600.000000 |
| ☐ | Edit | Copy | Delete | HR | 52000.000000 |
| ☐ | Edit | Copy | Delete | IT | 60500.000000 |
| ☐ | Edit | Copy | Delete | Marketing | 53333.333333 |
| ☐ | Edit | Copy | Delete | Sales | 56750.000000 |

28. **What is the highest salary in the Sales department?**
    - ❖ You can get the highest salary in the Sales department in a MySQL database by using the MAX() function along with a WHERE clause. Here is an example of how you can do it:
    - ❖
    - ❖
    - ❖

      SELECT MAX(Salary) AS HighestSalary
      FROM Employee
      WHERE Department = 'Sales';
    - ❖ This SQL command will calculate the maximum of the Salary field for all records in the Employee table where the Department is 'Sales', and display it as HighestSalary.
    - ❖ SQL OUTPUT FOR THE SYNTAX:

| HighestSalary |
|---|
| 58000.00 |

29. **List departments with more than 3 employees.**

- ❖ **You can list departments with more than 3 employees in a MySQL database by using the COUNT() function along with the GROUP BY and HAVING clauses. Here is an example of how you can do it:**
- ❖
- ❖
  **SELECT Department, COUNT(*) AS NumberOfEmployees**
  **FROM Employee**
  **GROUP BY Department**
  **HAVING COUNT(*) > 3;**
- ❖ **This SQL command will count the number of records (employees) for each Department in the Employee table, and display those departments where the count is more than 3 as NumberOfEmployees.**
- ❖ **SQL OUTPUT FOR THE SYNTAX:**

| Department | NumberOfEmployees |
|------------|-------------------|
| Finance | 5 |
| HR | 4 |
| IT | 4 |
| Sales | 4 |

30. **Find the department with the most employees.**
- ❖ **You can find the department with the most employees in a MySQL database by using the COUNT() function along with the GROUP BY and ORDER BY clauses. Here is an example of how you can do it:**
- ❖
- ❖
  **SELECT Department, COUNT(*) AS NumberOfEmployees**
  **FROM Employee**
  **GROUP BY Department**
  **ORDER BY NumberOfEmployees DESC**
  **LIMIT 1;**
- ❖ **This SQL command will count the number of records (employees) for each Department in the Employee table, order them in descending order of the count, and display the department with the most employees.**
- ❖ **SQL OUTPUT FOR THE SYNTAX:**

| Department | NumberOfEmployees |
|------------|-------------------|
| Finance | 5 |

**Joins and Relationships:**

31. **Do you have any information about the managers of each department?**
    - ❖ **To retrieve information about the managers of each department, you would need to have a field in your Employee table that specifies who the manager of each employee is. This is typically done by having a ManagerID field in the Employee table that references the EmployeeID of the manager.**
    - ❖
    - ❖ **Assuming you have such a field and it's named ManagerID, and you want to find out who the manager of each department is, you could use a query like this:**
    - ❖
    - ❖

      **SELECT Department, ManagerID**
      **FROM Employee**
      **GROUP BY Department;**
    - ❖ **This will give you the ManagerID of one employee from each department. If the ManagerID is the same for all employees in a department, this will effectively give you the manager of each department.**
    - ❖
    - ❖ **However, please note that this is a simplified scenario. In reality, data organization might be more complex and you might need to use JOIN operations or subqueries depending on how your data is structured.**
    - ❖ **SQL OUTPUT FOR THE SYNTAX:**

## Error

**SQL query:** Copy

```
SELECT Department, ManagerID
FROM Employee
GROUP BY Department LIMIT 0, 25
```
**MySQL said:**

#1054 - Unknown column 'ManagerID' in 'field list'

32. **Show employees who do not belong to any department.**
    - ❖ **You can show employees who do not belong to any department in a MySQL database by using the IS NULL keyword. Here is an example of how you can do it:**
    - ❖
    - ❖

      **SELECT \***
      **FROM Employee**
      **WHERE Department IS NULL;**
    - ❖ **This SQL command will select all records from the Employee table where the Department is NULL.**
    - ❖ **SQL OUTPUT FOR THE SYNTAX:**

| EmployeeID | FirstName | LastName | Department | Salary |
|---|---|---|---|---|

Query results operations

33. **List employees along with their department names.**
    - ❖ **You can list employees along with their department names in a MySQL database by using a simple SELECT statement. Here is an example of how you can do it:**
    - ❖
    - ❖
      **SELECT FirstName, LastName, Department**
      **FROM Employee;**
    - ❖ **This SQL command will select the FirstName, LastName, and Department fields from all records in the Employee table.**
    - ❖ **SQL OUTPUT FOR THE SYNTAX:**

| ←T→ | | | | FirstName | LastName | Department |
|---|---|---|---|---|---|---|
| ☐ | Edit | Copy | Delete | John | Doe | HR |
| ☐ | Edit | Copy | Delete | Jane | Smith | IT |
| ☐ | Edit | Copy | Delete | Michael | Johnson | Finance |
| ☐ | Edit | Copy | Delete | Emily | Brown | Marketing |
| ☐ | Edit | Copy | Delete | David | Wilson | Sales |
| ☐ | Edit | Copy | Delete | Sarah | Lee | IT |
| ☐ | Edit | Copy | Delete | James | Anderson | Finance |
| ☐ | Edit | Copy | Delete | Olivia | Martinez | HR |
| ☐ | Edit | Copy | Delete | Sophia | Jackson | Marketing |
| ☐ | Edit | Copy | Delete | Liam | Harris | IT |
| ☐ | Edit | Copy | Delete | Ava | White | Finance |
| ☐ | Edit | Copy | Delete | William | Miller | HR |
| ☐ | Edit | Copy | Delete | Ella | Thompson | Finance |
| ☐ | Edit | Copy | Delete | Alexander | Walker | Sales |
| ☐ | Edit | Copy | Delete | Mia | Clark | IT |
| ☐ | Edit | Copy | Delete | Henry | Lewis | Finance |
| ☐ | Edit | Copy | Delete | Sofia | Hall | HR |
| ☐ | Edit | Copy | Delete | Jackson | Young | Sales |
| ☐ | Edit | Copy | Delete | Luna | Wright | Marketing |
| ☐ | Edit | Copy | Delete | Isabella | Davis | Sales |

34. **Retrieve employees who have the same last name.**

- ❖ **You can use the following SQL query to retrieve employees who have the same last name in the employee table:**
- ❖
- ❖

  **SELECT a.***
  **FROM employee a**
  **JOIN (**
   **SELECT LastName**
   **FROM employee**
   **GROUP BY LastName**
   **HAVING COUNT(*) > 1**
  **) b ON a.LastName = b.LastName;**
- ❖ **This query works by first creating a subquery that groups the employee table by last_name and selects only those names that appear more than once (i.e., HAVING COUNT(*) > 1). The main query then joins this subquery with the original employee table on the last_name field, effectively filtering out employees who have a unique last name. The result is a list of all employees who share their last name with at least one other employee.**
- ❖
- ❖ **Please note that this query will return all columns for these employees. If you only want to see specific columns, you can replace a.* with the column names you are interested in, separated by commas. For example, if you only want to see their first and last names, you could use a.FirstName, a.LastName instead of a.*.**
- ❖ **SQL OUTPUT FOR THE SYNTAX:**

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0097 seconds.)

| EmployeeID | FirstName | LastName | Department | Salary |
|------------|-----------|----------|------------|--------|

Query results operations

**35. Find employees with the same department and salary.**

- ❖ **You can use the following SQL query to retrieve employees who have the same department and salary in the employee table:**
- ❖
- ❖

  **SELECT a.***
  **FROM employee a**
  **JOIN (**
   **SELECT department, salary**
   **FROM employee**
   **GROUP BY department, salary**
   **HAVING COUNT(*) > 1**
  **) b ON a.department = b.department AND a.salary = b.salary;**
- ❖ **This query works by first creating a subquery that groups the employee table by department and salary and selects only those combinations that appear more than once (i.e., HAVING COUNT(*) > 1). The main query then joins this subquery with the original employee table on the department and salary fields, effectively filtering out employees who have a unique combination of department and salary.**

The result is a list of all employees who share their department and salary with at least one other employee. Please replace employee, department, and salary with your actual table and column names if they are different.

❖

❖ Please note that this query will return all columns for these employees. If you only want to see specific columns, you can replace a.* with the column names you are interested in, separated by commas. For example, if you only want to see their names, department, and salary, you could use a.name, a.department, a.salary instead of a.*.

❖ SQL OUTPUT FOR THE SYNTAX ::

| EmployeeID | FirstName | LastName | Department | Salary |
|---|---|---|---|---|
| 1 | John | Doe | HR | 52000.00 |
| 8 | Olivia | Martinez | HR | 52000.00 |
| 15 | Alexander | Walker | Sales | 57000.00 |
| 21 | Isabella | Davis | Sales | 57000.00 |

**Advanced Queries:**

36. Find the employee(s) with the second-highest salary.
    ❖ You can use the following SQL query to retrieve the employee(s) with the second-highest salary in the employee table:
    ❖
    ❖
    ```
    SELECT *
    FROM employee
    WHERE salary = (
      SELECT MAX(salary)
      FROM employee
      WHERE salary < (
        SELECT MAX(salary)
        FROM employee
)
    );
    ```
    ❖ This query works by first finding the highest salary in the employee table, then finding the highest salary that is less than this value. The main query then selects all employees whose salary matches this second-highest value.
    ❖
    ❖ Please note that this query will return all columns for these employees. If you only want to see specific columns, you can replace * with the column names you are interested in, separated by commas. For example, if you only want to see their names and salaries, you could use name, salary instead of *.
    ❖ SQL OUTPUT FOR THE SYNTAX ::
    ❖

| | | | EmployeeID | FirstName | LastName | Department | Salary |
|---|---|---|---|---|---|---|---|
| ←T→ | | | | | | | |
| ☐ | Edit | Copy Delete | 11 | Liam | Harris | IT | 61000.00 |

37. **Display the employee(s) with the longest first name.**
- ❖ **You can use the following SQL query to retrieve the employee(s) with the longest first name in the employee table:**
- ❖

  **SELECT ***
  **FROM employee**
  **WHERE LENGTH(FirstName) = (**
  **SELECT MAX(LENGTH(FirstName))**
  **FROM employee**
  **);**
- ❖ **This query works by first finding the maximum length of first_name in the employee table. The main query then selects all employees whose first_name length matches this maximum length.**

- ❖ **Please note that this query will return all columns for these employees. If you only want to see specific columns, you can replace * with the column names you are interested in, separated by commas. For example, if you only want to see their names, you could use FirstName, LastName instead of *.**
- ❖ **SQL OUTPUT FOR THE SYNTAX ::**
- ❖

| | | | EmployeeID | FirstName | LastName | Department | Salary |
|---|---|---|---|---|---|---|---|
| ←T→ | | | | | | | |
| ☐ | Edit | Copy Delete | 15 | **Alexander** | Walker | Sales | 57000.00 |

38. **Calculate the salary difference between employees with EmployeeID 3 and 6.**
- ❖ **you can use the following SQL query to calculate the salary difference between employees with EmployeeID 3 and 6 in the employee table:**
- ❖
- ❖

  **SELECT ABS(**
  **(SELECT salary FROM employee WHERE EmployeeID = 3) -**
  **(SELECT salary FROM employee WHERE EmployeeID = 6)**
  **) AS salary_difference;**
- ❖ **This query works by first selecting the salary of the employee with EmployeeID 3 and subtracting from it the salary of the employee with EmployeeID 6. The ABS function is used to ensure that the result is always positive, regardless of which salary is larger. The result is aliased as salary_difference.**
- ❖ **SQL OUTPUT FOR THE SYNTAX ::**

| salary_difference |
|---|
| **7000.00** |

39. List employees who earn more than the average salary in their department.
   ❖ You can use the following SQL query to list employees who earn more than the average salary in their department in the employee table:
   ❖
   ❖

   **SELECT e1.***
   **FROM employee e1**
   **JOIN (**
     **SELECT department, AVG(salary) AS avg_salary**
     **FROM employee**
     **GROUP BY department**
   **) e2 ON e1.department = e2.department**
   **WHERE e1.salary > e2.avg_salary;**

   ❖ This query works by first creating a subquery that calculates the average salary for each department in the employee table. The main query then joins this subquery with the original employee table on the department field and selects all employees whose salary is greater than the average salary of their department. Please replace employee, department, and salary with your actual table and column names if they are different.
   ❖
   ❖ Please note that this query will return all columns for these employees. If you only want to see specific columns, you can replace e1.* with the column names you are interested in, separated by commas. For example, if you only want to see their names and salaries, you could use e1.name, e1.salary instead of e1.*.
   ❖ SQL OUTPUT FOR THE SYNTAX ::

| EmployeeID | FirstName | LastName | Department | Salary |
|---|---|---|---|---|
| 6 | Sarah | Lee | IT | 62000.00 |
| 7 | James | Anderson | Finance | 58000.00 |
| 11 | Liam | Harris | IT | 61000.00 |
| 12 | Ava | White | Finance | 57000.00 |
| 15 | Alexander | Walker | Sales | 57000.00 |
| 17 | Henry | Lewis | Finance | 59000.00 |
| 18 | Sofia | Hall | HR | 53000.00 |
| 19 | Jackson | Young | Sales | 58000.00 |
| 20 | Luna | Wright | Marketing | 55000.00 |
| 21 | Isabella | Davis | Sales | 57000.00 |

40. Show the employees with the highest salary within each department.

❖ **You can use the following SQL query to show the employees with the highest salary within each department in the employee table:**

❖

❖

**SELECT e1.***
**FROM employee e1**
**JOIN (**
  **SELECT department, MAX(salary) AS max_salary**
  **FROM employee**
  **GROUP BY department**
**) e2 ON e1.department = e2.department AND e1.salary = e2.max_salary;**

❖ **This query works by first creating a subquery that calculates the maximum salary for each department in the employee table. The main query then joins this subquery with the original employee table on the department field and the salary field, effectively selecting all employees who have the highest salary in their department. Please replace employee, department, and salary with your actual table and column names if they are different.**

❖

❖ **Please note that this query will return all columns for these employees. If you only want to see specific columns, you can replace e1.* with the column names you are interested in, separated by commas. For example, if you only want to see their names and salaries, you could use e1.name, e1.salary instead of e1.*.**

❖ **SQL OUTPUT FOR THE SYNTAX:**

| EmployeeID | FirstName | LastName | Department | Salary |
|---|---|---|---|---|
| 6 | Sarah | Lee | IT | 62000.00 |
| 17 | Henry | Lewis | Finance | 59000.00 |
| 18 | Sofia | Hall | HR | 53000.00 |
| 19 | Jackson | Young | Sales | 58000.00 |
| 20 | Luna | Wright | Marketing | 55000.00 |

# ####### END OF ASSIGNMENT #######