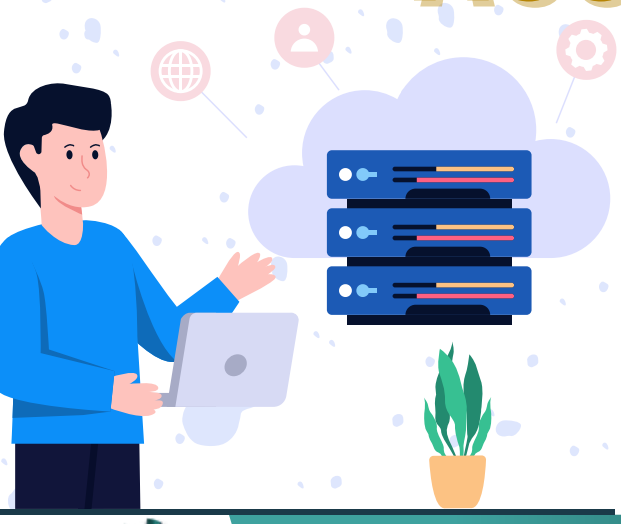
SQL ASSIGNMENT



Database , Tables , Constraints

1. Create new database & employee table (based on give sample data)
    - create employee table with primary key (EmployeeID)
2. Insert sample data into the table.
3. Write a query to create a clone of an existing table using Create Command.
4. Write a query to get all employee detail from "employee" table
5. Select only top 1 record from employee table
6. Select only bottom 1 record from employee table
7. How to select a random record from a table?
8. Write a query to get
    - “first_name” in upper case as "first_name_upper"
    - ‘first_name’ in lower case as ‘first_name_lower”
    - Create a new column “full_name” by combining “first_name” & “last_name” with space as a separator.
    - Add 'Hello ' to first_name and display result.
9. Select the employee details of
    - Whose “first_name” is ‘Malli’
    - Whose “first_name” present in ("Malli","Meena", "Anjali")
    - Whose “first_name” not present in ("Malli","Meena", "Anjali")
    - Whose “first_name” starts with “v”
    - Whose “first_name” ends with “i”
    - Whose “first_name” contains “o”
    - Whose "first_name" start with any single character between 'm-v'
    - Whose "first_name" not start with any single character between 'm-v'
    - Whose "first_name" start with 'M' and contain 5 letters
10. Write a query to get all unique values of"department" from the employee table.
11. Query to check the total records present in a table.
12. Write down the query to print first letter of a Name in Upper Case and all other letter in Lower Case.(EmployDetail table)
13. Write down the query to display all employee name in one cell separated by ',' ex:-"Vikas, nikita, Ashish, Nikhil , anish"(EmployDetail table).

14. Query to get the below values of "salary" from employee table

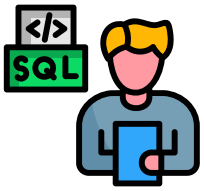
- Lowest salary
- Highest salary
- Average salary
- Highest salary - Lowest salary as diff_salary
- % of difference between Highest salary and lowest salary. (sample output format: 10.5%)

15. Select “first_name” from the employee table after removing white spaces from
    - Right side spaces
    - Left side spaces
    - Both right & left side spaces
16. Query to check no.of records present in a table where employees having 50k salary.
17. Find the most recently hired employee in each department.

Case When Then End Statement Queries

1. Display first_name and gender as M/F.(if male then M, if Female then F)
2. Display first_name, salary, and a salary category. (If salary is below 50,000, categorize as 'Low'; between 50,000 and 60,000 as 'Medium'; above 60,000 as 'High')
3. Display first_name, department, and a department classification. (If department is 'IT', display 'Technical'; if 'HR', display 'Human Resources'; if 'Finance', display 'Accounting'; otherwise, display 'Other')
4. Display first_name, salary, and eligibility for a salary raise. (If salary is less than 50,000, mark as 'Eligible for Raise'; otherwise, 'Not Eligible')
5. Display first_name, joining_date, and employment status. (If joining date is before '2022-01-01', mark as 'Experienced'; otherwise, 'New Hire')
6. Display first_name, salary, and bonus amount. (If salary is above 60,000, add10% bonus; if between 50,000 and 60,000, add 7%; otherwise, 5%)
7. Display first_name, salary, and seniority level.
8. (If salary is greater than 60,000, classify as 'Senior'; between 50,000 and 60,000 as 'Mid-Level'; below 50,000 as 'Junior')

9. Display first_name, salary, and seniority level.
10. (If salary is greater than 60,000, classify as 'Senior'; between 50,000 and 60,000 as 'Mid-Level'; below 50,000 as 'Junior')
11. Display first_name, department, and job level for IT employees. (If department is 'IT' and salary is greater than 55,000, mark as 'Senior IT Employee'; otherwise, 'Other').
12. Display first_name, joining_date, and recent joiner status. (If an employee joined after '2024-01-01', label as 'Recent Joiner'; otherwise, 'Long-Term Employee')
13. Display first_name, joining_date, and leave entitlement. (If joined before '2021-01-01', assign '10 Days Leave'; between '2021-01-01' and '2023-01-01', assign '20 Days Leave'; otherwise, '25 Days Leave')
14. Display first_name, salary, department, and promotion eligibility. (If salary is above 60,000 and department is 'IT', mark as 'Promotion Eligible'; otherwise, 'Not Eligible')
15. Display first_name, salary, and overtime pay eligibility. (If salary is below 50,000, mark as 'Eligible for Overtime Pay'; otherwise, 'Not Eligible')
16. Display first_name, department, salary, and job title. (If department is 'HR' and salary is above 60,000, mark as 'HR Executive'; if department is 'Finance' and salary is above 55,000, mark as 'Finance Manager'; otherwise, 'Regular Employee')
17. Display first_name, salary, and salary comparison to the company average. (If salary is above the company's average salary, mark as 'Above Average'; otherwise, 'Below Average')



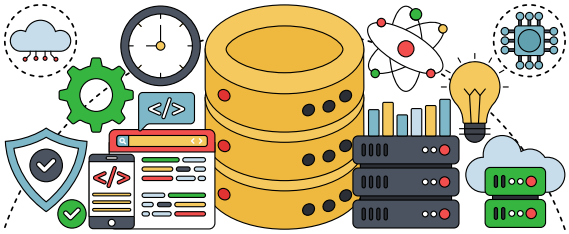
Group by

1. Write the query to get the department and department wise total(sum) salary, display it in ascending and descending order according to salary.
2. Write down the query to fetch Project name assign to more than one Employee
3. Write the query to get the department, total no. of departments, total(sum) salary with respect to department from "employee table" table.
4. Get the department-wise salary details from the "employee table" table:
   - What is the average salary? (Order by salary ascending)
   - What is the maximum salary? (Order by salary ascending)

5. Display department-wise employee count and categorize based on size. (If a department has more than 5 employees, label it as 'Large'; between 3 and 5 as 'Medium'; otherwise, 'Small')
6. Display department-wise average salary and classify pay levels. (If the average salary in a department is above 60,000, label it as 'High Pay'; between 50,000 and 60,000 as 'Medium Pay'; otherwise, 'Low Pay').
7. Display department, gender, and count of employees in each category. (Group by department and gender, showing total employees in each combination)
8. Display the number of employees who joined each year and categorize hiring trends. (If a year had more than 5 hires, mark as 'High Hiring'; 3 to 5 as 'Moderate Hiring'; otherwise, 'Low Hiring')
9. Display department-wise highest salary and classify senior roles. (If the highest salary in a department is above 70,000, label as 'Senior Leadership'; otherwise, 'Mid-Level')
10. Display department-wise count of employees earning more than 60,000. (Group employees by department and count those earning above 60,000, labeling departments with more than 2 such employees as 'High-Paying Team')

Date time related queries

1. Query to extract the below things from joining_date column. (Year, Month, Day, Current Date)
2. Create two new columns that calculate the difference between joining_date and the current date. One column should show the difference in months, and the other should show the difference in days
3. Get all employee details from the employee table whose joining year is 2020.
4. Get all employee details from the employee table whose joining month is Feb.
5. Get all employee details from employee table whose joining date between "2021-01-01" and "2021-12-01"



Joins Related queries

1. Get the employee name and project name from the "employee table" and "ProjectDetail" for employees who have been assigned a project, sorted by first name.
2. Get the employee name and project name from the "employee table" and "ProjectDetail" for all employees, including those who have not been assigned a project, sorted by first name.
3. Get the employee name and project name from the "employee table" and "ProjectDetail" for all employees. If an employee has no assigned project, display "-No Project Assigned," sorted by first name.
4. Get all project names from the "ProjectDetail" table, even if they are not linked to any employee, sorted by first name from the "employee table" and "ProjectDetail" table.
5. Find the project names from the "ProjectDetail" table that have not been assigned to any employee using the "employee table" and "ProjectDetail" table.
6. Get the employee name and project name for employees who are assigned to more than one project.
7. Get the project name and the employee names of employees working on projects that have more than one employee assigned.
8. Get records from the "ProjectDetail" table where the corresponding employee ID does not exist in the "employee table."

Ranking Functions

1. Get all project names from the "ProjectDetail" table, even if they are not linked to any employee, sorted by first name from the "employee table" and "ProjectDetail" table.
2. Find the project names from the "ProjectDetail" table that have not been assigned to any employee using the "employee table" and "ProjectDetail" table.
3. Get the employee name and project name for employees who are assigned to more than one project.
4. Get the project name and the employee names of employees working on projects that have more than one employee assigned.
5. Get records from the "ProjectDetail" table where the corresponding employee ID does not exist in the "employee table."

Partitioning Data

1. Assign a row number to each employee within their department based on salary in descending order.
2. Rank employees within each department based on salary using RANK().
3. Rank employees within each department based on salary using DENSE_RANK().
4. Find the highest-paid employee in each department using RANK().
5. Find the second highest-paid employee in each department using RANK().
6. Rank employees based on their years of experience within each department.
7. Find the employee with the earliest join date in each department using RANK().

Complex Ranking Scenarios

1. Find the employees who earn more than the average salary of their department.
2. Rank employees within each job title in every department based on salary.
3. Find the top 3 highest-paid employees in each department.
4. Find employees who belong to the top 10% earners within their department using PERCENT_RANK().
5. Assign row numbers to employees based on their joining year using PARTITION BY YEAR(join_date).
6. Rank employees based on the number of projects handled within each department.
7. Find employees who are the only ones in their department (departments with only one employee).
8. Find the highest-paid employee in each job role within a department.
9. Find employees who have been working in the company the longest in each department.

Thank You

To know more about our courses

+91 89517 96123 | +91 9916961234 | +91 89517 85123

SCAN NOW