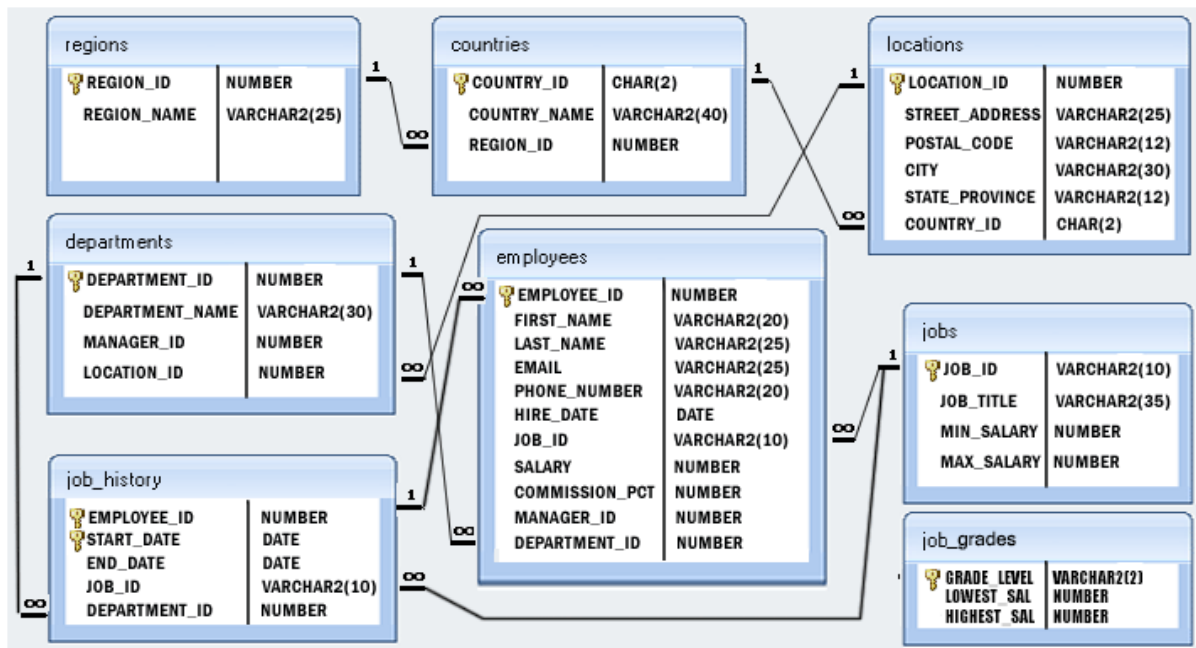


MYSQL HR DATABASE QUERIES:



1. Write a query to display the names (first_name, last_name) using alias name "First Name", "Last Name".

```
SELECT FIRST_NAME as "FIRST NAME", LAST_NAME as "LAST NAME" from employees;
```

2. Write a query to get unique department ID from employee table.

```
SELECT DISTINCT(DEPARTMENT_ID) from employees;
```

3. Write a query to get all employee details from the employee table order by first name, descending.

```
SELECT * from employees ORDER BY FIRST_NAME DESC;
```

4. Write a query to get the names (first_name, last_name), salary, PF of all the employees (PF is calculated as 15% of salary).

```
SELECT FIRST_NAME, LAST_NAME, SALARY, SALARY * 0.15 AS PF FROM employees;
```

5. Write a query to get the employee ID, names (first_name, last_name), salary in ascending order of salary.

```
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, SALARY FROM employees ORDER BY SALARY ASC;
```

6. Write a query to get the total salaries payable to employees.

```
SELECT SUM(SALARY) from employees;
```

7. Write a query to get the maximum and minimum salary from employees table.

```
SELECT MIN(SALARY) AS "MINIMUM SALARY", MAX(SALARY) AS "MAXIMUM SALARY" from employees;
```

8. Write a query to get the average salary and number of employees in the employees table.

```
SELECT AVG(SALARY), COUNT(EMPLOYEE_ID) AS "NUMBER OF EMPLOYEES" from employees;
```

9. Write a query to get the number of jobs available in the employees table.

```
SELECT COUNT(DISTINCT JOB_ID) FROM employees;
```

10. Write a query get all first name from employees table in upper case.

```
SELECT UPPER(FIRST_NAME) from employees;
```

11. Write a query to get the first 3 characters of first name from employees table.

```
SELECT LEFT(FIRST_NAME, 3) from employees;
```

12. Write a query to calculate $171 \times 214 + 625$.

```
SELECT 171*214+625 AS "RESULT";
```

13. Write a query to get the names (for example Ellen Abel, Sundar Ande etc.) of all the employees from employees table.

```
SELECT CONCAT(FIRST_NAME, " ", LAST_NAME) AS "EMPLOYEE NAME" from employees;
```

14. Write a query to get first name from employees table after removing white spaces from both side.

```
SELECT TRIM(FIRST_NAME) from employees;
```

15. Write a query to get the length of the employee names (first_name, last_name) from employees table.

```
SELECT FIRST_NAME, LAST_NAME, LENGTH(FIRST_NAME) + LENGTH(LAST_NAME) AS "FULL  
NAME LENGTH" from employees;
```

16. Write a query to check if the first_name fields of the employees table contains numbers.

```
SELECT FIRST_NAME FROM employees WHERE FIRST_NAME REGEXP '[0-9]';
```

17. Write a query to select first 10 records from a table.

```
SELECT * FROM `employees` LIMIT 10;
```

18. Write a query to get monthly salary (round 2 decimal places) of each and every employee.

```
SELECT FIRST_NAME, LAST_NAME, ROUND(SALARY/12, 2) AS "SALARY PER MONTH ROUNDED  
2" from employees;
```

19. Write a query to display the name (first_name, last_name) and salary for all employees whose salary is not in the range \$10,000 through \$15,000.

```
SELECT FIRST_NAME, LAST_NAME, SALARY FROM employees WHERE SALARY NOT BETWEEN  
10000 AND 15000;
```

20. Write a query to display the name (first_name, last_name) and department ID of all employees in departments 30 or 100 in ascending order.

```
SELECT DEPARTMENT_ID, CONCAT(FIRST_NAME, " ", LAST_NAME) AS "EMPLOYEE  
NAME" FROM employees WHERE DEPARTMENT_ID = 30 OR DEPARTMENT_ID = 100 ORDER  
BY DEPARTMENT_ID ASC;
```

OR

```
SELECT FIRST_NAME, LAST_NAME, DEPARTMENT_ID FROM employees WHERE DEPARTMENT  
_ID IN (30, 100) ORDER BY DEPARTMENT_ID ASC;
```

21. Write a query to display the name (first_name, last_name) and salary for all employees whose salary is not in the range \$10,000 through \$15,000 and are in department 30 or 100.

```
SELECT FIRST_NAME, LAST_NAME, DEPARTMENT_ID FROM employees WHERE DEPARTMENT_ID IN (30, 100) AND SALARY NOT BETWEEN 10000 AND 15000;
```

22. Write a query to display the name (first_name, last_name) and hire date for all employees who were hired in 1987.

```
SELECT FIRST_NAME, LAST_NAME, HIRE_DATE FROM employees WHERE YEAR(HIRE_DATE) = 1987;
```

23. Write a query to display the first_name of all employees who have both "b" and "c" in their first name.

```
SELECT FIRST_NAME FROM employees WHERE FIRST_NAME LIKE '%b%' AND FIRST_NAME LIKE '%c%';
```

24. Write a query to display the last name, job, and salary for all employees whose job is that of a Programmer or a Shipping Clerk, and whose salary is not equal to \$4,500, \$10,000, or \$15,000.

```
SELECT LAST_NAME, JOB_ID, SALARY FROM employees WHERE JOB_ID IN ('IT_PROG', 'SH_CLERK') AND SALARY NOT IN (4500, 10000, 15000);
```

25. Write a query to display the last name of employees whose names have exactly 6 characters.

```
SELECT LAST_NAME FROM employees WHERE LENGTH(LAST_NAME) = 6;
```

OR

```
SELECT LAST_NAME FROM employees WHERE LAST_NAME LIKE '_____';
```

26. Write a query to display the last name of employees having 'e' as the third character.

```
SELECT LAST_NAME FROM employees WHERE LAST_NAME LIKE '%_e%';
```

27. Write a query to display the jobs/designations available in the employees table.

```
SELECT DISTINCT JOB_ID FROM employees;
```

28. Write a query to select all record from employees where last name in 'BLAKE', 'SCOTT', 'KING' and 'FORD'.

```
SELECT * FROM employees WHERE LAST_NAME IN ('BLAKE', 'SCOTT', 'KING', 'FORD') ;
```

29. Write a query to list the number of jobs available in the employees table.

```
SELECT COUNT(DISTINCT(JOB_ID)) FROM employees;
```

30. Write a query to get the total salaries payable to employees.

```
SELECT SUM(SALARY) FROM employees;
```

31. Write a query to get the minimum salary from employees table.

```
SELECT MIN(SALARY) FROM employees;
```

32. Write a query to get the maximum salary of an employee working as a Programmer.

```
SELECT MAX(SALARY) FROM employees WHERE JOB_ID = "IT_PROG";
```

33. Write a query to get the average salary and number of employees working the department 90.

```
SELECT AVG(SALARY), COUNT(EMPLOYEE_ID) FROM employees WHERE DEPARTMENT_ID = 90;
```

34. Write a query to get the highest, lowest, sum, and average salary of all employees.

```
SELECT MIN(SALARY), MAX(SALARY), AVG(SALARY), SUM(SALARY) FROM employees;
```

35. Write a query to get the number of employees with the same job.

```
SELECT JOB_ID, COUNT(EMPLOYEE_ID) FROM employees GROUP BY JOB_ID;
```

36. Write a query to get the difference between the highest and lowest salaries

```
SELECT (MAX(SALARY) - MIN(SALARY)) AS "SALARY DIFFERENCE" FROM employees;
```

37. Write a query to find the manager ID and the salary of the lowest-paid employee for that manager.

```
SELECT MANAGER_ID, MIN(SALARY) FROM employees WHERE MANAGER_ID IS NOT NULL GROUP BY MANAGER_ID ORDER BY MIN(SALARY) ASC;
```

38. Write a query to get the department ID and the total salary payable in each department.

```
SELECT DEPARTMENT_ID, SUM(SALARY) FROM employees WHERE DEPARTMENT_ID IS NOT NULL GROUP BY DEPARTMENT_ID;
```

39. Write a query to get the average salary for each job ID excluding programmer.

```
SELECT JOB_ID, AVG(SALARY) FROM employees WHERE JOB_ID IS NOT NULL AND JOB_ID <> 'IT_PROG' GROUP BY JOB_ID;
```

40. Write a query to get the total salary, maximum, minimum, average salary of employees (job ID wise), for department ID 90 only.

```
SELECT JOB_ID, SUM(SALARY), MAX(SALARY), MIN(SALARY), AVG(SALARY) FROM employees WHERE DEPARTMENT_ID = 90 GROUP BY JOB_ID;
```

41. Write a query to get the job ID and maximum salary of the employees where maximum salary is greater than or equal to \$4000.

```
SELECT JOB_ID, MAX(SALARY) FROM employees GROUP BY JOB_ID HAVING MAX(SALARY) >= 4000;
```

42. Write a query to get the average salary for all departments employing more than 10 employees.

```
SELECT COUNT(EMPLOYEE_ID), DEPARTMENT_ID, AVG(SALARY) FROM employees GROUP BY DEPARTMENT_ID HAVING COUNT(EMPLOYEE_ID) > 10;
```

43. Write a query to find the name (first_name, last_name) and the salary of the employees who have a higher salary than the employee whose last_name='Bull'.

```
SELECT FIRST_NAME, LAST_NAME, SALARY FROM employees WHERE SALARY > (SELECT SALARY FROM employees WHERE LAST_NAME = 'BULL');
```

44. Write a query to find the name (first_name, last_name) of all employees who works in the IT department.

```
SELECT FIRST_NAME, LAST_NAME FROM employees WHERE DEPARTMENT_ID = (SELECT DEPARTMENT_ID FROM departments WHERE DEPARTMENT_NAME = "IT");
```

45. Write a query to find the name (first_name, last_name) of the employees who have a manager and worked in a USA based department.

```
SELECT first_name, last_name FROM employees WHERE manager_id IN (select manager_id FROM employees WHERE department_id IN (SELECT department_id FROM departments WHERE location_id IN (SELECT location_id from locations where country_id='US')));
```

46. Write a query to find the name (first_name, last_name), and salary of the employees whose salary is greater than the average salary.

```
SELECT FIRST_NAME, LAST_NAME, SALARY FROM employees WHERE SALARY > (SELECT AVG(SALARY) FROM employees);
```

47. Write a query to find the name (first_name, last_name), and salary of the employees whose salary is equal to the minimum salary for their job grade.

```
SELECT FIRST_NAME, LAST_NAME, SALARY FROM employees WHERE employees.SALARY = (SELECT MIN_SALARY FROM jobs WHERE employees.JOB_ID = jobs.JOB_ID);
```

48. Write a query to find the name (first_name, last_name), and salary of the employees who earns more than the average salary and works in any of the IT departments.

```
SELECT FIRST_NAME, LAST_NAME, SALARY FROM employees WHERE DEPARTMENT_ID IN (SELECT DEPARTMENT_ID FROM departments WHERE DEPARTMENT_NAME LIKE '%IT%') AND SALARY > (SELECT AVG(SALARY) FROM employees);
```

49. Write a query to find the name (first_name, last_name), and salary of the employees who earns more than the earning of Mr. Bell.

```
SELECT FIRST_NAME, LAST_NAME, SALARY FROM employees WHERE salary > (SELECT salary FROM employees WHERE LAST_NAME="Bell") ORDER BY FIRST_NAME;
```

50. Write a query to find the name (first_name, last_name), and salary of the employees who earn the same salary as the minimum salary for all departments.

```
SELECT FIRST_NAME, LAST_NAME, SALARY FROM employees WHERE SALARY IN (SELECT MIN(SALARY) FROM employees WHERE DEPARTMENT_ID IN (SELECT DEPARTMENT_ID FROM departments));
```

OR

```
SELECT * FROM employees WHERE SALARY = (SELECT MIN(SALARY) FROM employees);
```

51. Write a query to find the name (first_name, last_name), and salary of the employees whose salary is greater than the average salary of all departments.

```
SELECT * FROM employees WHERE SALARY > ALL (SELECT AVG(SALARY) FROM employees GROUP BY DEPARTMENT_ID);
```

52. Write a query to find the name (first_name, last_name) and salary of the employees who earn a salary that is higher than the salary of all the Shipping Clerk (JOB_ID = 'SH_CLERK'). Sort the results of the salary of the lowest to highest.

```
SELECT FIRST_NAME, LAST_NAME, SALARY FROM employees WHERE SALARY > ALL (SELECT SALARY FROM employees WHERE JOB_ID = 'SH_CLERK') ORDER BY SALARY ASC;
```

53. Write a query to display the employee ID, first name, last name, and department names of all employees.

```
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, (SELECT DEPARTMENT_NAME FROM departments D WHERE D.DEPARTMENT_ID = E.DEPARTMENT_ID) DEPARTMENT FROM employees E ORDER BY DEPARTMENT;
```

54. Write a query to display the employee ID, first name, last name, salary of all employees whose salary is above average for their departments.

```
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, SALARY FROM employees A WHERE SALARY > (SELECT AVG(SALARY) FROM employees WHERE DEPARTMENT_ID = A.DEPARTMENT_ID);
```

55. Write a query to fetch even numbered records from employees table.

```
SELECT EMPLOYEE_ID FROM employees WHERE EMPLOYEE_ID % 2 = 0;
```


56. Write a query to find the 5th maximum salary in the employees table.

```
SELECT SALARY FROM employees ORDER BY SALARY DESC LIMIT 5;
```

57. Write a query to select last 10 records from a table.

```
SELECT * FROM (SELECT * FROM employees ORDER BY EMPLOYEE_ID DESC LIMIT 10)  
RESULT ORDER BY EMPLOYEE_ID ASC;
```

58. Write a query to list the department ID and name of all the departments where no employee is working.

```
SELECT * FROM departments WHERE DEPARTMENT_ID NOT IN (SELECT DEPARTMENT_ID  
FROM employees);
```

59. Write a query to get 3 maximum salaries.

```
SELECT DISTINCT(SALARY) FROM employees ORDER BY SALARY DESC LIMIT 3;
```

60. Write a query to get 3 minimum salaries.

```
SELECT DISTINCT(SALARY) FROM employees ORDER BY SALARY ASC LIMIT 3;
```

61. Write a query to find the addresses (location_id, street_address, city, state_province, country_name) of all the departments.

```
SELECT location_id,street_address,city,state_province,country_name FROM loc  
ations NATURAL JOIN countries;
```

62. Write a query to find the name (first_name, last name), department ID and name of all the employees.

```
SELECT e.FIRST_NAME,e.LAST_NAME,e.DEPARTMENT_ID,d.DEPARTMENT_NAME FROM empl  
oyees e INNER JOIN departments d ON e.DEPARTMENT_ID = d.DEPARTMENT_ID;
```

63. Write a query to find the name (first_name, last_name), job, department ID and name of the employees who works in London.

```
SELECT e.FIRST_NAME,e.LAST_NAME,e.JOB_ID,e.DEPARTMENT_ID,d.DEPARTMENT_NAME  
FROM employees e INNER JOIN departments d ON (e.DEPARTMENT_ID = d.DEPARTMEN  
T_ID) INNER JOIN locations l ON (d.LOCATION_ID = l.LOCATION_ID) WHERE LOWER  
(l.CITY) = "London";
```

64. Write a query to find the name (first_name, last_name) and hire date of the employees who was hired after 'Jones'.

```
SELECT e.FIRST_NAME, e.LAST_NAME, e.HIRE_DATE FROM employees e JOIN employees f ON (f.last_name = 'Jones') WHERE f.hire_date < e.hire_date;
```

65. Write a query to get the department name and number of employees in the department.

```
SELECT DEPARTMENT_NAME, COUNT(EMPLOYEE_ID) FROM departments INNER JOIN employees ON employees.DEPARTMENT_ID = departments.DEPARTMENT_ID GROUP BY departments.DEPARTMENT_ID, DEPARTMENT_NAME ORDER BY DEPARTMENT_NAME;
```

66. Write a query to find the employee ID, job title, number of days between ending date and starting date for all jobs in department 90.

```
SELECT EMPLOYEE_ID, JOB_TITLE, END_DATE - START_DATE FROM job_history NATURAL JOIN jobs WHERE DEPARTMENT_ID = 90;
```

67. Write a query to display the department ID and name and first name of manager.

```
SELECT departments.DEPARTMENT_ID, departments.DEPARTMENT_NAME, departments.MANAGER_ID, employees.FIRST_NAME FROM departments INNER JOIN employees ON departments.MANAGER_ID = employees.EMPLOYEE_ID;
```

68. Write a query to display the department name, manager name, and city.

```
SELECT departments.DEPARTMENT_NAME, employees.FIRST_NAME, locations.CITY FROM departments JOIN employees ON (departments.MANAGER_ID = employees.EMPLOYEE_ID) JOIN locations USING (LOCATION_ID);
```

69. Write a query to display the job title and average salary of employees.

```
SELECT jobs.JOB_TITLE, AVG(employees.SALARY) FROM employees JOIN jobs ON (jobs.JOB_ID = employees.JOB_ID) GROUP BY JOB_TITLE;
```

70. Write a query to display job title, employee name, and the difference between salary of the employee and minimum salary for the job.

```
SELECT jobs.JOB_TITLE, employees.FIRST_NAME, employees.LAST_NAME, jobs.MAX_SALARY - jobs.MIN_SALARY AS "DIFFERENCE IN SALARY" FROM employees INNER JOIN jobs ON (jobs.JOB_ID = employees.JOB_ID);
```

71. Write a query to display the job history that were done by any employee who is currently drawing more than 10000 of salary.

```
SELECT * FROM job_history JOIN employees ON (job_history.EMPLOYEE_ID = employees.EMPLOYEE_ID) WHERE employees.SALARY > 10000;
```

72. Write a query to display department name, name (first_name, last_name), hire date, salary of the manager for all managers whose experience is more than 15 years.

```
SELECT departments.DEPARTMENT_NAME, employees.FIRST_NAME, employees.LAST_NAME, employees.HIRE_DATE, employees.SALARY, (DATEDIFF(now(), employees.HIRE_DATE)) / 365 "EXPERIENCE" FROM departments INNER JOIN employees ON (departments.MANAGER_ID = employees.EMPLOYEE_ID) WHERE (DATEDIFF(now(), employees.HIRE_DATE)) / 365 > 15;
```