

# HR Schema

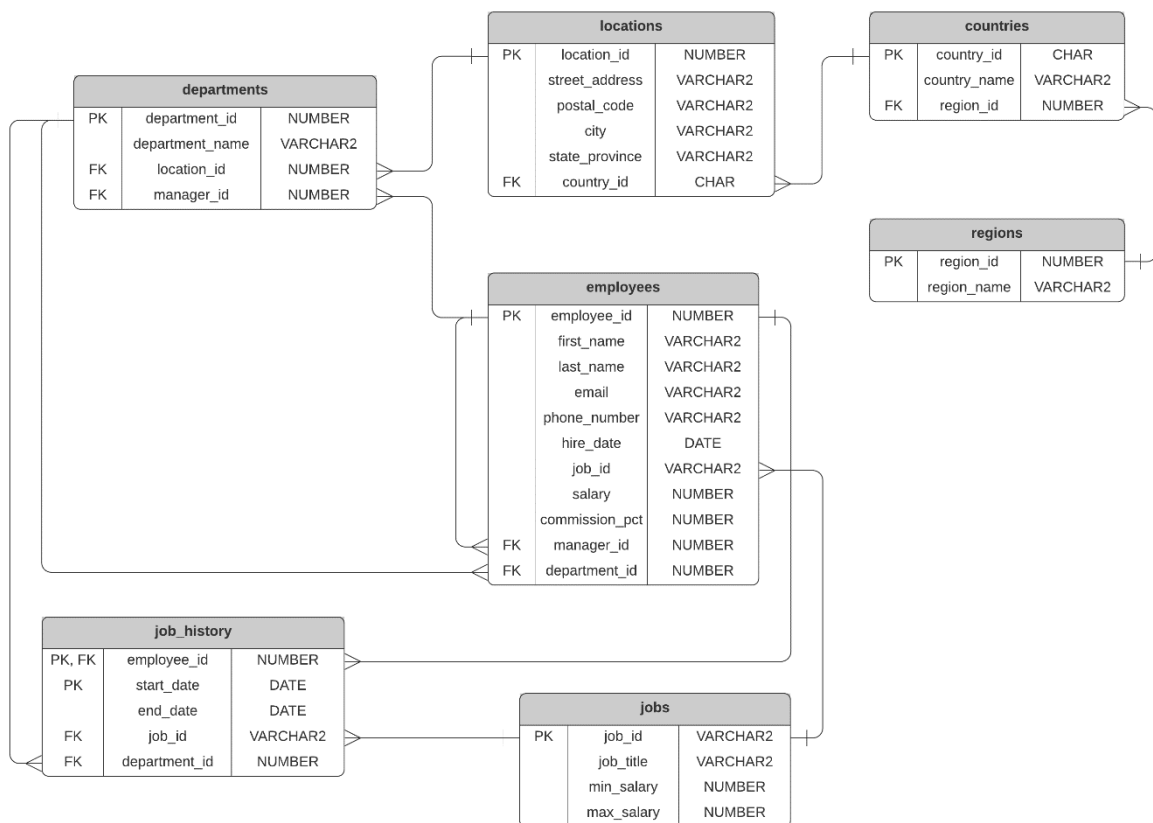
## SQL Practices and Solutions

**On MSSQL Database**

The purpose of this project is to work on HR Schema database to practice SQL queries.

## Practice I-1: Introduction

This figure shows the tables in the HR schema and the columns in each table, as well as dependencies between the tables:



COUNTRIES

DEPARTMENTS

EMPLOYEES

JOB\_HISTORY

JOBS

LOCATIONS

REGIONS

# Queries used in Project

1. Display names and salaries of employees.

```
SELECT first_name  
       ,salary  
FROM employees
```

2. Display the last name and salary of employees earning more than \$12,000

```
SELECT last_name  
       ,salary  
FROM employees  
WHERE salary > 12000
```

3. Display the last name and department number for employee number 176

```
SELECT last_name  
       ,department_id  
FROM employees  
WHERE employee_id = 176
```

4. Display the last name and salary for all employees whose salary is not in the range of \$5,000 to \$12,000

a)

```
SELECT last_name  
       ,salary  
FROM employees  
WHERE salary < 5000  
       OR salary > 12000
```

b)

```
SELECT last_name  
       ,salary  
FROM employees  
WHERE salary NOT BETWEEN 5000  
              AND 12000
```

5. Display the last name, job ID, and start date (hire date) for the employees with the last names of Matos and Taylor. Order the query in ascending order by start date.

```
SELECT last_name
       ,job_id
       ,hire_date
FROM employees
WHERE last_name = 'Matos'
       OR last_name = 'Taylor'
ORDER BY hire_date ASC
```

6. Display the last name and department number of all employees in departments 20 or 50 in ascending alphabetical order by name.

a)

```
SELECT last_name
       ,department_id
FROM employees
WHERE department_id = 20
       OR department_id = 50
ORDER BY first_name ASC
```

b)

```
SELECT last_name
       ,department_id
FROM employees
WHERE department_id IN (
                        20
                        ,50
                        )
ORDER BY first_name ASC
```

7. Display the last name and job title of all employees who do not have a manager.

```
SELECT last_name
       ,job_id
FROM employees
WHERE manager_id IS NULL
```

8. Display the last name, salary, and commission for all employees who earn commissions. Sort data in descending.

```
SELECT last_name
       ,salary
       ,commission_pct
FROM employees
WHERE commission_pct IS NOT NULL
ORDER BY hire_date DESC
```

9. Find the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively.

```
SELECT MAX(salary) AS Maximum
      ,MIN(salary) AS Minimum
      ,SUM(salary) AS Sum
      ,AVG(salary) AS Average
FROM employees
```

10. Display details of jobs where the minimum salary is greater than 10000.

```
SELECT *
FROM jobs
WHERE min_salary > 12000
```

11. Display the first name and join date of the employees who joined between 2002 and 2005.

```
SELECT first_name
      ,hire_date
FROM employees
WHERE hire_date BETWEEN '2002'
                  AND '2005'
```

12. Display first name and join date of the employees who is either IT Programmer or Sales Man.

```
SELECT first_name
      ,hire_date
FROM employees
WHERE job_id = 'IT_PROG'
      OR job_id = 'SA_MAN'
```

13. Display employees who joined after 1st January 2008.

```
SELECT *
FROM employees
WHERE hire_date > '2008-01-01'
```

14. Display details of employee with ID 150 or 160.

```
SELECT *
FROM employees
WHERE employee_id = 150
      OR employee_id = 160
```

15. Display first name, salary, commission pct, and hire date for employees with salary less than 10000.

```
SELECT first_name
       ,salary
       ,commission_pct
       ,hire_date
FROM employees
WHERE salary < 10000
```

16. Display job Title, the difference between minimum and maximum salaries for jobs with max salary in the range 10000 to 20000.

```
SELECT job_title
       ,DIFFERENCE('min_salary', 'max_salary') AS difference
FROM jobs
WHERE max_salary BETWEEN 10000
                  AND 20000
```

17. Display first name, salary, and round the salary to thousands.

```
SELECT first_name
       ,salary
       ,ROUND(salary, - 3) AS 'round_salary'
FROM employees
```

18. Display details of jobs in the descending order of the title.

```
SELECT *
FROM jobs
ORDER BY job_title DESC
```

19. Display employees where the first name or last name starts with S.

```
SELECT *
FROM employees
WHERE last_name LIKE 'S%'
      OR first_name LIKE 'S%'
```

20. Display employees who joined in the month of May.

```
SELECT *
FROM employees
WHERE month(hire_date) = 05
```

21. Display details of the employees where commission percentage is null and salary in the range 5000 to 10000 and department is 30.

```
SELECT *
FROM employees
WHERE commission_pct = NULL
      AND salary BETWEEN 5000
                AND 10000
      AND department_id = 30
```

22. Display first name and date of first salary of the employees.

```
SELECT first_name
       ,eomonth(hire_date)
FROM employees
```

23. Display first name and experience of the employees.

```
SELECT first_name
       ,DATEDIFF(YEAR, hire_date, '08-05-2023') AS year_experience
FROM employees
```

24. Display first name of employees who joined in 2000.

```
SELECT first_name
FROM employees
WHERE year(hire_date) = 2000
```

25. Display first name and last name after converting the first letter of each name to upper case and the rest to lower case.

```
SELECT UPPER(LEFT(first_name, 1)) + LOWER(SUBSTRING(first_name, 2,
len(first_name))) AS first_name
       ,UPPER(LEFT(last_name, 1)) + LOWER(SUBSTRING(last_name, 2,
len(last_name))) AS last_name
FROM employees
```

26. Display the first word in job title.

```
SELECT SUBSTRING(job_title, 1, CHARINDEX(' ', job_title + ' ') - 1)
AS first_word
FROM jobs
```

27. Display the length of first name for employees where last name contain character 'b' after 3rd position.

```
SELECT LEN(first_name)
      , *
FROM employees
WHERE SUBSTRING(last_name, 4, 1) = 'b'
```

28. Display first name in upper case and email address in lower case for employees where the first name and email address are same irrespective of the case.

```
SELECT upper(first_name)
      , lower(email)
FROM employees
```

29. Display employees who joined in the current year.

```
SELECT *
FROM employees
WHERE year(hire_date) = year(current_timestamp)
```

30. Display how many employees joined in each month of the current year.

```
SELECT COUNT(EMPLOYEE_ID) as NumberOfEmployees , DATEPART(month, HIRE_DATE) as Month
FROM EMPLOYEES
where DATEPART(year, HIRE_DATE) = 1998
group by DATEPART(month, HIRE_DATE)
```

31. Display Employees Where The First Name Or Last Name Starts With S.

```
SELECT *
FROM employees
WHERE first_name LIKE 'S%'
      OR last_name LIKE 'S%'
```

32. From The Employees Table Select The Last Name And The Job Id. Using The Case Statement, Give A Detailed Description To The Job Id. For Example: If Job Id Is 'Ad\_Vp' Then Display 'Administrative Vice President', If The Job Id Is 'Ad\_Asst' Then Display 'Administrative Assistant'; Otherwise Display 'Unknow Job Id'

```
SELECT last_name
      , job_id AS 'Administrative Vice President'
FROM employees
WHERE (job_id = 'AD_VP')
```



33. Display Manager Id And Number Of Employees Managed By The Manager.

```
SELECT manager_id
       ,count(employee_id) AS 'number_of_employees'
FROM employees
WHERE manager_id IS NOT NULL
GROUP BY manager_id
ORDER BY COUNT(employee_id)
```

34. Display The Country Id And Number Of Cities We Have In The Country.

```
SELECT country_id
       ,count(city) AS city
FROM locations
GROUP BY country_id
```

35. Display Average Salary Of Employees In Each Department Who Have Commission Percentage.

```
SELECT department_id
       ,AVG(salary) AS average_salary
FROM employees
WHERE commission_pct IS NOT NULL
GROUP BY department_id
```

36. Display Job Id, Number Of Employees, Sum Of Salary, And Difference Between Highest Salary And Lowest Salary Of The Employees Of The Job.

```
SELECT Job_Id
       ,Count(*)
       ,Sum(Salary)
       , (Max(Salary) - Min(Salary)) AS Difference
FROM Employees
GROUP BY Job_Id
```

37. Display Job Id For Jobs With Average Salary More Than 10000.

```
SELECT job_id
       ,AVG(salary)
FROM employees
GROUP BY job_id
HAVING AVG(salary) > 10000
```

38. Display Departments In Which More Than Five Employees Have Commission Percentage.

```
SELECT department_id
       ,count(commission_pct) AS employee
FROM employees
WHERE commission_pct IS NOT NULL
GROUP BY department_id
HAVING count(commission_pct) > 5
```

39. Display Employee Id For Employees Who Did More Than One Job In The Past.

```
SELECT employee_id  
       ,count(*)  
FROM job_history  
GROUP BY employee_id  
HAVING count(*) > 1
```

48. Change Salary Of Employee 115 To 8000 If The Existing Salary Is Less Than 6000.

```
UPDATE employees  
SET salary = 8000  
WHERE employee_id = 116  
      AND salary < 6000
```