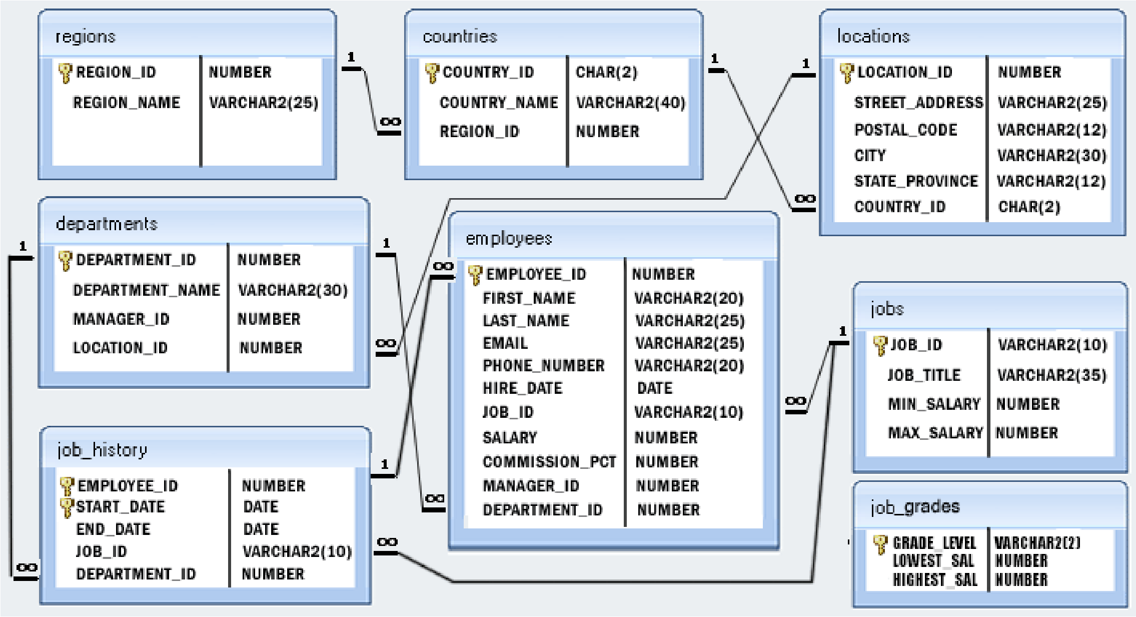
Employees , department and locations table are given in excel format sqlassignment.xlsx



regions

REGION_ID	NUMBER
REGION_NAME	VARCHAR2(25)

countries

COUNTRY_ID	CHAR(2)
COUNTRY_NAME	VARCHAR2(40)
REGION_ID	NUMBER

locations

LOCATION_ID	NUMBER
STREET_ADDRESS	VARCHAR2(25)
POSTAL_CODE	VARCHAR2(12)
CITY	VARCHAR2(30)
STATE_PROVINCE	VARCHAR2(12)
COUNTRY_ID	CHAR(2)

departments

DEPARTMENT_ID	NUMBER
DEPARTMENT_NAME	VARCHAR2(30)
MANAGER_ID	NUMBER
LOCATION_ID	NUMBER

employees

EMPLOYEE_ID	NUMBER
FIRST_NAME	VARCHAR2(20)
LAST_NAME	VARCHAR2(25)
EMAIL	VARCHAR2(25)
PHONE_NUMBER	VARCHAR2(20)
HIRE_DATE	DATE
JOB_ID	VARCHAR2(10)
SALARY	NUMBER
COMMISSION_PCT	NUMBER
MANAGER_ID	NUMBER
DEPARTMENT_ID	NUMBER

jobs

JOB_ID	VARCHAR2(10)
JOB_TITLE	VARCHAR2(35)
MIN_SALARY	NUMBER
MAX_SALARY	NUMBER

job_history

EMPLOYEE_ID	NUMBER
START_DATE	DATE
END_DATE	DATE
JOB_ID	VARCHAR2(10)
DEPARTMENT_ID	NUMBER

job_grades

GRADE_LEVEL	VARCHAR2(2)
LOWEST_SAL	NUMBER
HIGHEST_SAL	NUMBER

1. 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');
```

2. 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 manager_id in (select employee_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')));
```

3. 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.
Hint : Write single-row and multiple-row subqueries

```
SELECT first_name, last_name FROM employees
WHERE manager_id in (select employee_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')));
```

4. Write a query to find the name (first_name, last_name) of the employees who are managers.

```
SELECT first_name, last_name
FROM employees
WHERE (employee_id IN (SELECT manager_id FROM employees));
```

5. 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);
```

6. 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);
```

7. 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);
```

8. 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;
```

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

```
SELECT department_name AS 'Department Name',
COUNT(*) AS 'No of Employees'
FROM departments
INNER JOIN employees
ON employees.department_id = departments.department_id
GROUP BY departments.department_id, department_name
ORDER BY department_name;
```

10. Write a query to get 3 maximum salaries.

```
SELECT DISTINCT salary
FROM employees a
WHERE 3 >= (SELECT COUNT(DISTINCT salary)
FROM employees b
WHERE b.salary >= a.salary)
ORDER BY a.salary DESC;
```

11. Write a query to get 3 minimum salaries.

```
SELECT DISTINCT salary
FROM employees a
WHERE  3 >= (SELECT COUNT(DISTINCT salary)
FROM employees b
WHERE b.salary <= a.salary)
ORDER BY a.salary DESC;
```

12. Write a query to get nth max salaries of employees.

```
SELECT *
FROM employees emp1
WHERE (1) = (
SELECT COUNT(DISTINCT(emp2.salary))
FROM employees emp2
WHERE emp2.salary > emp1.salary);
```

13. Write a query to find the addresses (location_id, street_address, city, state_province, country_name) of all the departments.
Hint : Use NATURAL JOIN.

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

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

```
SELECT first_name, last_name, department_id, department_name
FROM employees
JOIN departments USING (department_id);
```

15. 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
JOIN departments d
ON (e.department_id = d.department_id)
JOIN locations l ON
(d.location_id = l.location_id)
WHERE LOWER(l.city) = 'London';
```

16. Write a query to find the employee id, name (last_name) along with their manager_id and name (last_name).

```
SELECT e.employee_id 'Emp_Id', e.last_name 'Employee',
m.employee_id 'Mgr_Id', m.last_name 'Manager'
FROM employees e
join employees m
ON (e.manager_id = m.employee_id);
```

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

```
SELECT d.department_name, e.first_name, l.city
FROM departments d
JOIN employees e
ON (d.manager_id = e.employee_id)
JOIN locations l USING (location_id);
```

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

```
SELECT job_title, AVG(salary)
FROM employees
NATURAL JOIN jobs
GROUP BY job_title;
```

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

```
SELECT job_title, first_name, salary-min_salary 'Salary - Min_Salary'
FROM employees
NATURAL JOIN jobs;
```

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

```
SELECT jh.* FROM job_history jh
JOIN employees e
ON (jh.employee_id = e.employee_id)
WHERE salary > 10000;
```

21. 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 first_name, last_name, hire_date, salary,
(DATEDIFF(now(), hire_date))/365 Experience
FROM departments d JOIN employees e
ON (d.manager_id = e.employee_id)
WHERE (DATEDIFF(now(), hire_date))/365>15;
```