QUESTIONS FOR HUMAN RESOURCES DATABASE

1. Display the current and previous details of all employees.

```
SELECT      employee_id, job_id
FROM        employees
UNION
SELECT      employee_id, job_id
FROM        job_history
ORDER BY    employee_id;
```

2. Display the current and previous departments of all employees.

```
SELECT      employee_id, job_id, department_id
FROM        employees
UNION ALL
SELECT      employee_id, job_id, department_id
FROM        job_history
ORDER BY    employee_id;
```

3. Display the employee IDs and job IDs of those employees who currently have a job title that is the same as their previous one (that is, they changed jobs but have now gone back to doing the same job they did previously)

```
SELECT      employee_id, job_id
FROM        employees
INTERSECT
SELECT      employee_id, job_id
FROM        job_history;
```

4. Display the employee IDs of those employees who have not changed their jobs even once.

```
SELECT      employee_id
FROM        employees
MINUS
SELECT      employee_id
FROM        job_history;
```

5. Using UNION operator, display the location Id, department name, and the state where it is located.
NOTE: You must match the data type(using the TO_CHAR function or any other conversion functions) when columns do not exists in one or the other table)

```
SELECT      location_id, department_name “Department”, TO_CHAR(NULL) “WAREHOUSE
LOCATION”
FROM        departments
UNION
SELECT      location_id, TO_CHAR(NULL) “Department”, state_province
FROM        locations;
```

6. Using the UNION operator, display the employee ID, job ID, and salary of all employees.

```
SELECT    employee_id, job_id, salary
FROM      employees
UNION
SELECT    employee_id, job_id, 0
FROM      job_history;
```

7. Which employees have salaries greater than Abel's salary?

```
SELECT    last_name, salary
FROM      employees
WHERE     salary > (SELECT salary
                    FROM employees
                    WHERE last_name = 'Abel');
```

8. Find the employees who have same job_id but have greater salary than 'Taylor'.

```
SELECT    last_name, job_id, salary
FROM      employees
WHERE     job_id =(SELECT job_id
                   FROM employees
                   WHERE last_name = 'Taylor')
AND       salary > (SELECT salary
                   FROM employees
                   WHERE last_name = 'Taylor');
```

9. Find the names, job IDs and salary of those employees whose salary is smaller than any one of the salaries of IT_Programmers.

```
SELECT    employee_id, last_name, job_id, salary
FROM      employees
WHERE     salary < ANY    (SELECT salary
                           FROM employees
                           WHERE job_id = 'IT_PROG')
AND       job_id <> 'IT_PROG';
```

10. Find the names, job IDs and salary of those employees whose salary is smaller than ALL one of the salaries of IT_Programmers.

```
SELECT    employee_id, last_name, job_id, salary
FROM      employees
WHERE     salary < ALL    (SELECT salary
                           FROM employees
                           WHERE job_id = 'IT_PROG')
AND       job_id <> 'IT_PROG';
```

11. Find the managers whose salary is less than any one of his/her employees.

```
SELECT    employee_id, salary, last_name
FROM      employees M
WHERE     M. Salary < ANY (SELECT salary
                           FROM employees W
```

```
                                WHERE W.manager_id =  M. Employee_id) ;
```

12. Find the names of managers who has the workers earning more then 10.000

```
SELECT      employee_id, salary, last_name
FROM        employees M
WHERE EXISTS        (SELECT employee_id
                     FROM employees W
                     WHERE (W.manager_id =  M. Employee_id) AND W.salary>10000);
```

13. Find the names of those departments where there is no employees working for them.

```
SELECT *
FROM       departments
WHERE      NOT EXISTS    (SELECT *
                          FROM employees
                          WHERE employees.department_id =
departments.department_id);
```

14. Find the names of employees who are not manager. NOTE: What happens if sub query has at least one null value?)

```
SELECT      emp.last_name
FROM        employees emp
WHERE       emp. Employee_id NOT IN    (SELECT mgr.manager_id
                                        FROM employees mgr
                                        WHERE mgr.manager_id is NOT NULL);
```