

Departments(DEPARTMENT_ID, DEPARTMENT_NAME, MANAGER_ID, LOCATION_ID)
Employees(EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER,
HIRE_DATE, JOB_ID, SALARY, COMMISSION_PCT, MANAGER_ID,
DEPARTMENT_ID)

Locations(LOCATION_ID, STREET_ADDRESS, POSTAL_CODE, CITY, STATE_PROVINCE,
COUNTRY_ID)

Countries(COUNTRY_ID, COUNTRY_NAME, REGION_ID)

1. From the following table, write a SQL query to find those employees who receive a higher salary than the employee with ID 163. Return first name, last name.

```
SELECT first_name, last_name
FROM employees
WHERE salary >
( SELECT salary
FROM employees
WHERE employee_id=163
);
```

2. From the following table, write a SQL query to find out which employees have the same designation as the employee whose ID is 169. Return first name, last name, department ID and job ID.

```
SELECT first_name, last_name, salary, department_id, job_id
FROM employees
WHERE job_id =
( SELECT job_id
FROM employees
WHERE employee_id=169
);
```

3. From the following table, write a SQL query to find those employees whose salary matches the lowest salary of any of the departments. Return first name, last name and department ID.

```
SELECT first_name, last_name, salary, department_id
FROM employees
WHERE salary IN
( SELECT MIN(salary)
FROM employees
GROUP BY department_id
);
```

4. From the following table, write a SQL query to find those employees who earn more than the average salary. Return employee ID, first name, last name.

```
SELECT employee_id, first_name, last_name
FROM employees
```

```
WHERE salary >
( SELECT AVG(salary)
FROM employees
);
```

5. From the following table, write a SQL query to find those employees who report to that manager whose first name is 'Payam'. Return first name, last name, employee ID and salary.

```
SELECT first_name, last_name, employee_id, salary
FROM employees
WHERE manager_id =
(SELECT employee_id
FROM employees
WHERE first_name = 'Payam'
);
```

6. From the following table, write a SQL query to find those employees whose ID matches any of the numbers 134, 159 and 183. Return all the fields.

```
SELECT *
FROM employees
WHERE employee_id IN (134,159,183);
```

7. From the following table, write a SQL query to find those employees whose salary is in the range of 1000, and 3000 (Begin and end values have included.). Return all the fields.

```
SELECT * FROM employees
WHERE salary BETWEEN 1000 and 3000;
```

8. From the following table and write a SQL query to find those employees whose salary falls within the range of the smallest salary and 2500. Return all the fields.

```
SELECT *
FROM employees
WHERE salary BETWEEN
(SELECT MIN(salary)
FROM employees) AND 2500;
```

9. From the following tables, write a SQL query to find those employees who do not work in the departments where managers' IDs are between 100 and 200 (Begin and end values are included.). Return all the fields of the employees.

```
SELECT *
FROM employees
WHERE department_id NOT IN
(SELECT department_id
FROM departments
```

```
WHERE manager_id BETWEEN 100 AND 200));
```

10. From the following table, write a SQL query to find those employees who get second-highest salary. Return all the fields of the employees.

```
SELECT *  
FROM employees  
WHERE employee_id IN  
(SELECT employee_id  
FROM employees  
WHERE salary =  
(SELECT MAX(salary)  
FROM employees  
WHERE salary <  
(SELECT MAX(salary)  
FROM employees)));
```

11. From the following tables, write a SQL query to find those employees who work in the same department as 'Clara'. Exclude all those records where first name is 'Clara'. Return first name, last name and hire date.

```
SELECT first_name, last_name, hire_date  
FROM employees  
WHERE department_id =  
( SELECT department_id  
FROM employees  
WHERE first_name = 'Clara')  
AND first_name <> 'Clara';
```

12. write a SQL query to find those employees who work in a department where the employee's first name contains the letter 'T'. Return employee ID, first name and last name.

```
SELECT employee_id, first_name, last_name  
FROM employees  
WHERE department_id IN  
( SELECT department_id  
FROM employees  
WHERE first_name LIKE '%T%' );
```

13. write a SQL query to find those employees who earn more than the average salary and work in the same department as an employee whose first name contains the letter 'J'. Return employee ID, first name and salary.

```
SELECT employee_id, first_name , salary  
FROM employees  
WHERE salary >  
(SELECT AVG (salary)  
FROM employees )  
AND department_id IN
```

```
( SELECT department_id
FROM employees
WHERE first_name LIKE '%J%');
```

14. write a SQL query to find those employees whose department is located at 'Toronto'. Return first name, last name, employee ID, job ID.

```
SELECT first_name, last_name, employee_id, job_id
FROM employees
WHERE department_id =
(SELECT department_id
FROM departments
WHERE location_id =
(SELECT location_id
FROM locations
WHERE city = 'Toronto'));
```

15. write a SQL query to find those employees whose salary is lower than that of employees whose job title is 'MK_MAN'. Return employee ID, first name, last name, job ID.

```
SELECT employee_id, first_name, last_name, job_id
FROM employees
WHERE salary < ANY
( SELECT salary
FROM employees
WHERE job_id = 'MK_MAN' );
```

16. write a SQL query to find those employees whose salary is lower than that of employees whose job title is "MK_MAN". Exclude employees of Job title 'MK_MAN'. Return employee ID, first name, last name, job ID.

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

17. write a SQL query to find those employees whose salary exceeds the salary of all those employees whose job title is "PU_MAN". Exclude job title 'PU_MAN'. Return employee ID, first name, last name, job ID.

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

18. write a SQL query to find those employees whose salaries are higher than the average for all departments. Return employee ID, first name, last name, job ID.

```
SELECT employee_id, first_name, last_name, job_id
FROM employees
WHERE salary > ALL
( SELECT AVG(salary)
  FROM employees
  GROUP BY department_id
);
```

19. write a SQL query to check whether there are any employees with salaries exceeding 3700. Return first name, last name and department ID.

```
SELECT first_name, last_name, department_id
FROM employees
WHERE EXISTS
(SELECT *
 FROM employees
 WHERE salary > 3700 );
```

20. write a SQL query to find all those departments where at least one employee is employed. Return department name.

```
SELECT department_name
FROM departments
WHERE department_id IN
(SELECT DISTINCT(department_id)
 FROM employees);
```

21. write a SQL query to find employees who work in departments located in the United Kingdom. Return first name.

```
SELECT first_name
FROM employees
WHERE department_id IN
(SELECT department_id
 FROM departments
 WHERE location_id IN
 (SELECT location_id
  FROM locations
  WHERE country_id =
  (SELECT country_id
   FROM countries
   WHERE country_name='United Kingdom')));
```

22. write a SQL query to find out which employees are earning more than the average salary and who work in any of the IT departments. Return last name.

```
SELECT last_name
FROM employees
```

```

WHERE department_id IN
(SELECT department_id
FROM departments
WHERE department_name LIKE 'IT%')
AND salary >
(SELECT avg(salary)
FROM employees);

```

23. write a SQL query find the employees who report to a manager based in the United States. Return first name, last name.

```

SELECT first_name, last_name, salary
FROM employees
WHERE salary >
(SELECT salary
FROM employees
WHERE last_name='Ozer')
ORDER BY last_name;

```

24. write a SQL query to find those employees whose salaries exceed 50% of their department's total salary bill. Return first name, last name.

```

SELECT e1.first_name, e1.last_name
FROM employees e1
WHERE salary >
( SELECT (SUM(salary))*0.5
FROM employees e2
WHERE e1.department_id=e2.department_id);

```

25. write a SQL query to find those employees who are managers. Return all the fields of employees table.

```

SELECT *
FROM employees
WHERE employee_id IN
(SELECT DISTINCT manager_id FROM employees);

```