

— **Write a query to display the names (first_name, last_name) of all the employees :**

```
select FIRST_NAME, LAST_NAME from employees;
```

— **Write a query to get all employee details and order them by first name, descending:**

```
select FIRST_NAME from employees order by FIRST_NAME DESC;
```

```
select FIRST_NAME, last_name, email, phone_number from employees order by FIRST_NAME DESC;
```

— **Write a query to get the total salaries payable to employees.**

```
select sum(salary) from employees;
```

— **Write a query to get the maximum and minimum salary from employees table:**

```
SELECT MAX(salary), min(salary) FROM employees;
```

— **Write a query to get monthly salary (round 2 decimal places) of each and every employee. To get the monthly salary of an employee, divide the annual salary by 12,2**

```
sqlite> .mode column
sqlite> .headers on
sqlite> select first_name,
...> last_name,
...> round(salary/12,2)
...> as "salary monthly"
...> from employees;
```

— **Write a query to display the names (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 from employees where salary not between 10000 and 15000;
```

— **Write a query to display the names (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, salary, department_id from employees where salary not between 10000 and 15000 and department_id in(30,100);
```

— **Write a query to display the last names of employees having 'e' as the third character.**

```
SELECT last_name FROM employees WHERE last_name LIKE '__e%';
```

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

```
SELECT COUNT(DISTINCT job_id)
...> FROM employees;
COUNT(DISTINCT job_id)
```

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

```
SELECT department_id, SUM(salary)
...> FROM employees
...> GROUP BY department_id;
```

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

```
SELECT job_id, AVG(salary)
...> FROM employees
...> WHERE job_id <> 'IT_PROG'
...> GROUP BY job_id;
```

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

```
SELECT department_id, AVG(salary), COUNT(*)
...> FROM employees
...> GROUP BY department_id
...> HAVING COUNT(*) > 10;
```

— Write a query to find the names (first_name, last_name) and salaries 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');
```

— Write a query to find the names (first_name, last_name) of the employees who have a manager who works for a department based in the United States.

```
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')));
```

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

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

— **Write a query to display job title, employee name, and the difference between the 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;
```