**SubQuery**

A **subquery** in SQL is a query nested inside another query. It can be used in different parts of a SQL statement, such as the SELECT, FROM, WHERE, or HAVING clauses.

There are two types of subqueries:

1. **Single-row subquery**: Returns a single value.
2. **Multi-row subquery**: Returns multiple values (like a list).

Employees table;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Employee\_id | First\_name | Last\_name | Salary | Department |
| 1 | John | Doe | 6000 | 101 |
| 2 | Jana | Smith | 7500 | 102 |
| 3 | Alex | White | 5000 | 101 |
| 4 | Emily | Brown | 8000 | 103 |
| 5 | David | Black | 4000 | 102 |

Departments table;

|  |  |  |
| --- | --- | --- |
| department\_id | department\_name | location\_id |
| 101 | Salers | 1700 |
| 102 | HR | 1800 |
| 103 | IT | 1700 |

Find all employees who are in departments located in location\_id = 1700.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SELECT** **CONCAT**(first\_name +last\_name)  **FROM** employees  **WHERE** department\_id **IN** (**SELECT** department\_id  **FROM** departments  **WHERE** location\_id = 1700  ); | |  | | --- | | First\_name+last\_name | | John doe | | Alex white | | Emily brown |   . |

For each employee, show their salary and the average salary across all employees.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SELECT** department\_id, total\_salary  **FROM** (**SELECT** department\_id, **SUM**(salary) **AS** total\_salary  **FROM** employees  **GROUP** **BY** department\_id) **AS** dept\_salaries; | .   |  |  | | --- | --- | | department\_id | location\_id | | 101 | 1700 | | 102 | 1800 | | 103 | 1700 |   .. |

Calculate the total salary for each department using a subquery as a temporary table.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SELECT first\_name, last\_name, salary,  (SELECT AVG(salary) FROM employees) AS avg\_salary  FROM employees; | .   |  |  |  |  | | --- | --- | --- | --- | | First\_name | Last\_name | Salary | Avg\_sal | | John | Doe | 6000 | 6100 | | Jana | Smith | 7500 | 6100 | | Alex | White | 5000 | 6100 | | Emily | Brown | 8000 | 6100 | | David | Black | 4000 | 6100 |   . |

Find all departments where the total salary is greater than the average total salary of all departments.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SELECT department\_id, SUM(salary) AS total\_salary  FROM employees  GROUP BY department\_id  HAVING SUM(salary) > (SELECT AVG(total\_salary)  FROM (SELECT SUM(salary) AS total\_salary  FROM employees  GROUP BY department\_id) AS dept\_avg); | .   |  |  | | --- | --- | | Department\_d | Total\_salary | | 102 | 1150 |   . |

**SELECT** ROOM\_TYPE , (**SELECT** **SUM**(TOTAL\_PRICE)

**FROM** RESERVATIONS

**WHERE** ROOMS.ROOM\_ID = RESERVATIONS.ROOM\_ID) **AS** TOTAL

**FROM** ROOMS;