|  |  |
| --- | --- |
| |  | | --- | | **Interview Query Solution #39 | Employee Salaries** | |

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |
| --- | --- |
| |  | | --- | | Here's the question and answer for yesterdays sql question.  This question was asked by: **Microsoft** | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **employees table**   | **columns** | **types** | | --- | --- | | id | int | | first\_name | varchar | | last\_name | varchar | | salary | int | | department\_id | int |     **departments table**   | **columns** | **types** | | --- | --- | | id | int | | name | varchar |     1. Given the tables above, select the top 3 departments by the highest percentage of employees making over 100K in salary and have at least 10 employees.  Example output:   | **> 100K %** | **department name** | **number of employees** | | --- | --- | --- | | 90% | engineering | 25 | | 50% | marketing | 50 | | 12% | sales | 12 |   2. Let’s say due to an ETL error, the employee table instead of updating the salaries every year when doing compensation adjustments, did an insert instead. The head of HR still needs the current salary of each employee. Write a query to get the current salary for each employee.  Assume no duplicate combination of first and last names. (I.E. No two John Smiths) | |

|  |  |
| --- | --- |
| |  | | --- | | **Solution:** | |

|  |  |
| --- | --- |
| |  | | --- | | 1. We definitely need a JOIN, a HAVING clause, and a CASE WHEN to differentiate when a employee makes over 100K.  If we first do a join, we will get all employees and their departments. Then all that's left is grouping by the department name and calculating - a function to get all of the employees - a function to get all employees making over 100K - dividing those values by each other  **SELECT**  d.name  , **CAST**(**SUM**(**CASE** **WHEN** salary > 100000 **THEN** 1 **ELSE** 0 **END**) **AS** DECIMAL)/**COUNT**(\*) **AS** percent\_employees\_over\_100K  **FROM** departments **AS** d  **LEFT** **JOIN** employees **AS** e  **ON** d.id = e.department\_id  **GROUP** **BY** 1  **HAVING** **COUNT**(\*) >= 10  **ORDER** **BY** 2 **DESC**  **LIMIT** 3  2. The first step would be to remove duplicates. Given we know there aren't any duplicate first and last name combinations, we can remove duplicates from the employees table by just grouping by first and last name and getting the maximum id from the table which would be the last entry and the most up to date salary.  This way we can rejoin by doing a subquery and not get duplicates in the existing table.  **SELECT** e.first\_name, e.last\_name, e.salary  **FROM** employees **AS** e  **INNER** **JOIN** (  **SELECT** first\_name, last\_name, **MAX**(id) **AS** max\_id  **FROM** employees  **GROUP** **BY** 1,2  ) **AS** m  **ON** e.id = m.max\_id | |

|  |  |
| --- | --- |
| |  | | --- | |  | |