Yuvraj Singh
Data analyst Intern Assignment
singhraj153989@gmail.com
https://yuvraj153989.github.io/Portfolio/
http://linkedin.com/in/yuvraj-singh-3018741b3
+91 8591476799

TASK 01

Your task is to give SQL queries for problem statements with the help of below mentioned tables -
Table - LINK
A. Retrieve the names and salaries of all employees, along with the average salary in their respective
departments.
B. Calculate the total sales amount for each employee, including those who have not made any sales.
Display their names and total sales amount.
C. Rank employees within each department based on their salary in descending order. The ranking should
reset for each department.if two employees have the highest salary, they will both be assigned the rank
1, and the next distinct salary will be assigned the rank 3 (skipping 2).
D. Rank employees within each department based on their salary in descending order. The ranking should
reset for each department.if two employees have the highest salary, they will both be assigned the rank
1, and the next distinct salary will be assigned the rank 2."

Queries

```
use assignment;

-- viewing employee table--
select * from employee;

-- Viewing sales table--
select * from SALES;
```

-- Question 1: Retrieve the names and salaries of all employees, along with the average salary in their respective departments.

```
SELECT
   EMP_ID,
   Name,
   Department,
   Salary,
```

```
   AVG(Salary) OVER (PARTITION BY Department) AS Avg_Salary_In_Department
FROM
   employee;
```

-- Question 2: Calculate the total sales amount for each employee, including those who have not made any sales. Display their names and total sales amount.

```
SELECT
   employee.Name,
   IFNULL(SUM(SALES.Sale_Amount), 0) AS Total_Sales_Amount
FROM
   employee
LEFT JOIN
   SALES ON employee.EMP_ID = SALES.Employee_ID
GROUP BY
   employee.EMP_ID, employee.Name;
```

-- Question 3: Rank employees within each department based on their salary in descending order. The ranking should reset for each department.if two employees have the highest salary, they will both be assigned the rank 1**, and the next distinct salary will be assigned the rank 3 (skipping 2).**

```
SELECT
          EMP_ID,Name,
   Department,
   Salary,
   RANK() OVER (PARTITION BY Department ORDER BY Salary DESC) AS Salary_Rank
from
          employee;
```

-- **Question 4: Rank employees within each department based on their salary in descending order. The ranking should reset for each department.if two employees have the highest salary, they will both be assigned the rank 1, and the next distinct salary will be assigned the rank 2."**

```
SELECT
   Department,
   EMP_ID,
   Name,
   Salary,
   DENSE_RANK() OVER (PARTITION BY Department ORDER BY Salary DESC) AS Salary_Rank
FROM
   employee;
```

TASK 02

Your task is to give queries for problem statements with the help of below mentioned 3 tables -

Table 1 - rm_master - contains info of the raw material sold by the vendor - LINK
Table 2 - rm_mapping_master - contains info about the raw material used in SKU - LINK
Table 3 - sales_master - quantity of SKU sold on a given day - LINK

A. To get the total quantity of each rm(raw material) sold in each month.
B. To get the name of vendors for each SKU.
C. Get the most used and least used raw material based on the SKU sold.

Queries

```
use openinapp;

-- viewing rm_master table
select * from rm_master;

-- viewing rm_mapping_master
select * from rm_mapping_master;

-- viewing sales_master
select * from sales_master;
```

-- **Q1. To get the total quantity of each rm(raw material) sold in each month.**

```
-- formatting date column
UPDATE sales_master
SET date = STR_TO_DATE(date, '%d/%m/%Y');

select sum(Quantity), month(date) as sale_month
from sales_master
group by sale_month;
```

-- Q2. To get the name of vendors for each SKU.

```
SELECT
   rmm.SKU,
   REPLACE(GROUP_CONCAT(DISTINCT rm.VendorName ORDER BY rm.VendorName ASC), ',', ',  ') AS
Vendor_Names
FROM
   rm_mapping_master rmm
LEFT JOIN
   rm_master rm ON rmm.`RID` = rm.`RID` GROUP BY  rmm.SKU;
```

-- Q3: Get the most used and least used raw material based on the SKU sold.

Most Used Raw Material

```
SELECT
   rmm.`RID` AS Raw_Material_ID,
   rm.`RName` AS Raw_Material_Name,
   SUM(rmm.Quantity) AS Total_Quantity_Sold
FROM
   rm_mapping_master rmm
JOIN
   rm_master rm ON rmm.`RID` = rm.`RID`
GROUP BY
   rmm.`RID`, rm.`RName`
ORDER BY
   Total_Quantity_Sold DESC
LIMIT 1;
```

-- Least Used Raw Material

```
SELECT
   rmm.`RID` AS Raw_Material_ID,
   rm.`RName` AS Raw_Material_Name,
   SUM(rmm.Quantity) AS Total_Quantity_Sold
FROM
   rm_mapping_master rmm
JOIN
   rm_master rm ON rmm.`RID` = rm.`RID`
GROUP BY
   rmm.`RID`, rm.`RName`
ORDER BY
   Total_Quantity_Sold ASC
LIMIT 1;
```