

# SQL Assignment

Data Source Information:

Sales Table:

Table Name: sales

Columns:

product\_category (VARCHAR): The category of the product being sold.

revenue (DECIMAL): The revenue generated from each sale.

date (DATE): The date when the sale occurred.

Employees Table:

Table Name: employees

Columns:

employee\_id (INT): Unique identifier for each employee.

employee\_name (VARCHAR): The name of the employee.

salary (DECIMAL): The salary of the employee.

Orders Table:

Table Name: orders

Columns:

order\_id (INT): Unique identifier for each order.

customer\_id (INT): Unique identifier for each customer.

purchase\_amount (DECIMAL): The total purchase amount for each order.

## 1. \*\*Aggregate Functions:\*\*

- Write a SQL query to calculate the total revenue generated by each product category. Use the SUM() function for this calculation.

The screenshot displays the Snowflake web interface. On the left, a sidebar shows a tree view of databases and tables, with 'SALES' selected under 'SNOWFLAKE'. The main area shows a SQL query editor with the following code:

```
-- ANALYZE AND CUSTOMER'S EMPLOYMENT AM, CUSTOMER_HIERARCHY VALUE
-- (101, 'Alice Johnson'),
-- (102, 'Bob Smith'),
-- (103, 'Charlie Brown'),
-- (104, 'Emma Davis');
SELECT product_category, SUM(revenue) AS total_revenue
FROM sales
GROUP BY product_category;
```

Below the query editor, the 'Results' tab is active, showing a table with 3 rows and 2 columns: 'PRODUCT\_CATEGORY' and 'TOTAL\_REVENUE'.

PRODUCT_CATEGORY	TOTAL_REVENUE
1 Electronics	4500.00
2 Clothing	1400.00
3 Furniture	2500.00

On the right side of the results table, there is a 'Query Details' panel showing 'Query duration: 128ms' and 'Query ID: 01b53893-0000-aabc...'. At the bottom of the interface, a Windows taskbar is visible with the date '24-06-2024' and time '03:25 PM'.

## 2. \*\*Ranking Functions:\*\*

- Use the RANK() function to assign ranks to employees based on their salary. Display the employee name, salary, and rank in the result set.

The screenshot shows the Snowflake SQL IDE interface. The query editor contains the following SQL code:

```

63 -- SELECT product_category, sum(revenue) as total_revenue
64 -- FROM sales
65 -- GROUP BY product_category;
66
67
68 SELECT
69     employee_name,
70     salary,
71     RANK() OVER (ORDER BY salary DESC) AS salary_rank
72 FROM employees;
73

```

The results table shows the following data:

EMPLOYEE_NAME	SALARY	SALARY_RANK
Jane Smith	60000.00	1
Sarah Clark	59000.00	2
Emily Johnson	58000.00	3
Michael Brown	55000.00	4
David Lee	52000.00	5
John Doe	50000.00	6

### 3. \*\*Window Functions:\*\*

- Write a SQL query to calculate the running total of sales revenue over time. Use the ROW\_NUMBER() function to achieve this.

The screenshot shows the Snowflake SQL IDE interface. The query editor contains the following SQL code:

```

67 -- SELECT
68 --     employee_name,
69 --     salary,
70 --     RANK() OVER (ORDER BY salary DESC) AS salary_rank
71 -- FROM employees;
72
73
74 SELECT
75     date,
76     SUM(revenue) OVER (ORDER BY date) AS running_total
77 FROM sales;
78

```

The results table shows the following data:

DATE	RUNNING_TOTAL
2023-01-01	1500.00
2023-01-02	2300.00
2023-01-03	3500.00
2023-01-04	6000.00
2023-01-05	6600.00
2023-01-06	8400.00

### 4. \*\*Common Table Expressions (CTEs):\*\*

- Create a CTE that lists the top 5 customers based on their total purchase amount. Include customer ID, name, and total purchase amount in the result.

The screenshot shows the Snowflake web interface. On the left, the 'Databases' sidebar is open, showing a tree structure with 'CUSTOMERS', 'EMPLOYEES', 'ORDERS', and 'SALES' under the 'SNOWFLAKE' database. The 'SALES' table is selected, showing 6 rows. The main panel displays a SQL query in the 'ASSIGNMENT3.SALES' workspace:

```

80 customer_id,
81 SUM(purchase_amount) AS total_purchase_amount
82 FROM orders
83 GROUP BY customer_id
84 ORDER BY total_purchase_amount DESC
85 LIMIT 5
86 )
87 SELECT
88 tc.customer_id,
89 c.customer_name,

```

The 'Results' tab shows a table with 4 rows and 3 columns: CUSTOMER\_ID, CUSTOMER\_NAME, and TOTAL\_PURCHASE\_AMOUNT.

	CUSTOMER_ID	CUSTOMER_NAME	TOTAL_PURCHASE_AMOUNT
1	101	Alice Johnson	1800.00
2	102	Bob Smith	2600.00
3	103	Charlie Brown	3500.00
4	104	Emma Davis	3200.00

Query Details on the right show a duration of 258ms and 4 rows. The bottom status bar shows the system time as 04:01 PM on 24-06-2024.

## 5. \*\*NVL vs COALESCE:\*\*

- Explain the difference between the NVL() and COALESCE() functions in SQL. Provide an example for each function demonstrating its usage.

The screenshot shows the Snowflake web interface. On the left, the 'Databases' sidebar is open, showing a tree structure with 'CUSTOMERS', 'EMPLOYEES', 'ORDERS', and 'SALES' under the 'SNOWFLAKE' database. The 'SALES' table is selected, showing 6 rows. The main panel displays a SQL query in the 'ASSIGNMENT3.SALES' workspace:

```

103
104 -- Insert sample data into Employees Table
105 -- INSERT INTO Employees_WITH_NULLS (employee_id, employee_name, department, first_name, nickname) VALUES
106 -- (1, 'John Doe', 'HR', 'John', NULL),
107 -- (2, 'Jane Smith', NULL, 'Jane', 'J'),
108 -- (3, 'Michael Brown', 'IT', NULL, 'Mike'),
109 -- (4, 'Emily Johnson', 'Finance', NULL, NULL),
110 -- (5, 'David Lee', NULL, NULL, 'Dave'),
111 -- (6, 'Sarah Clark', 'Marketing', 'Sarah', NULL);
112 SELECT * FROM Employees_WITH_NULLS
113
114
115

```

The 'Results' tab shows a table with 6 rows and 5 columns: EMPLOYEE\_ID, EMPLOYEE\_NAME, DEPARTMENT, FIRST\_NAME, and NICKNAME.

	EMPLOYEE_ID	EMPLOYEE_NAME	DEPARTMENT	FIRST_NAME	NICKNAME
1	1	John Doe	HR	John	null
2	2	Jane Smith	null	Jane	J
3	3	Michael Brown	IT	null	Mike
4	4	Emily Johnson	Finance	null	null
5	5	David Lee	null	null	Dave
6	6	Sarah Clark	Marketing	Sarah	null

Query Details on the right show a duration of 306ms and 6 rows. The bottom status bar shows the system time as 04:10 PM on 24-06-2024.

app.snowflake.com/corgmids/gz16346/w9lrQX/VKQx#query

2024-06-24 11:37am 2024-06-24 12:25pm 2024-06-24 12:36pm 2024-06-24 12:36pm world\_city 2024-06-24 3:17pm 2024-06-24 3:17pm

Databases Worksheets

Search objects

- CUSTOMERS
- EMPLOYEES
- ORDERS
- SALES

SALES 6 Rows

PRODUCT\_CATEGORY VARCHAR(50)  
REVENUE NUMBER(10,2)  
DATE DATE

ASSIGNMENT3.SALES Settings

```
-- (1, 'John Doe', 'HR', 'John', NULL),  
-- (2, 'Jane Smith', NULL, 'Jane', 'J'),  
-- (3, 'Michael Brown', 'IT', NULL, 'Mike'),  
-- (4, 'Emily Johnson', 'Finance', NULL, NULL),  
-- (5, 'David Lee', NULL, NULL, 'Dave'),  
-- (6, 'Sarah Clark', 'Marketing', 'Sarah', NULL);  
-- SELECT * FROM Employees_WITH_NULLS;  
  
SELECT employee_name, NVL(department, 'Not specified') AS department  
FROM Employees_WITH_NULLS;
```

Results Chart

EMPLOYEE_NAME	DEPARTMENT
1 John Doe	HR
2 Jane Smith	Not specified
3 Michael Brown	IT
4 Emily Johnson	Finance
5 David Lee	Not specified
6 Sarah Clark	Marketing

Query Details

Query duration 113ms

Rows 6

Query ID 01b538c1-0000-aaca-0-

Show more

31°C Mostly sunny

ENG IN 04:11 PM 24-06-2024

app.snowflake.com/corgmids/gz16346/w9lrQX/VKQx#query

2024-06-24 11:37am 2024-06-24 12:25pm 2024-06-24 12:36pm 2024-06-24 12:36pm world\_city 2024-06-24 3:17pm 2024-06-24 3:17pm

Databases Worksheets

Search objects

- CUSTOMERS
- EMPLOYEES
- ORDERS
- SALES

SALES 6 Rows

PRODUCT\_CATEGORY VARCHAR(50)  
REVENUE NUMBER(10,2)  
DATE DATE

ASSIGNMENT3.SALES Settings

```
-- (4, 'Emily Johnson', 'Finance', NULL, NULL),  
-- (5, 'David Lee', NULL, NULL, 'Dave'),  
-- (6, 'Sarah Clark', 'Marketing', 'Sarah', NULL);  
-- SELECT * FROM Employees_WITH_NULLS;  
  
-- SELECT employee_name, NVL(department, 'Not specified') AS department  
-- FROM Employees_WITH_NULLS;  
  
SELECT  
    employee_name,  
    COALESCE(first_name, nickname, 'No name provided') AS preferred_name  
FROM Employees_WITH_NULLS;
```

Results Chart

EMPLOYEE_NAME	PREFERRED_NAME
1 John Doe	John
2 Jane Smith	Jane
3 Michael Brown	Mike
4 Emily Johnson	No name provided
5 David Lee	Dave
6 Sarah Clark	Sarah

Query Details

Query duration 46ms

Rows 6

Query ID 01b538c2-0000-aaca-0-

Show more

31°C Mostly sunny

ENG IN 04:12 PM 24-06-2024