Snowflake.DAA-C01.v2026-09-03.q67

□□□□:	DAA-C01
□□□□:	SnowPro Advanced: Data Analyst Certification Exam
□□□:	Snowflake
□□ □□ □□□:	67
□□:	v2026-09-03
# □□ □:	242
# □□ □□□:	670
https://www.krdump.com/Snowflake.DAA-C01.v2026-09-03.q67.html	

NEW QUESTION: 1
'PRODUCTS'□□ □□□□ 'PRICE'□□ □□ VARCHAR □□□□ □□□□ □□□□. □ □□ □ □□□ □□□ □□ □□□(□: '12.99')□ □□, □□□□ □□ □□(□: '12.99USD', 'N/A')□ □□□□ □□ □□. □□ □□□ □□□ □□ □□□ □□□□ □□□. □□ □ □□□ □□ □□ □□□□ □□ □□□ □□□ □□□□□ □□□□ □□□ □□□□□? □□□□ □□ □□□ □□□□□□.

A.
```
SELECT AVG(TRY_CAST(PRICE AS DECIMAL(10,2))) FROM PRODUCTS WHERE IS_DECIMAL(PRICE);
```

B.
```
SELECT AVG(CAST(PRICE AS DECIMAL(10,2))) FROM PRODUCTS WHERE REGEXP_LIKE(PRICE, '^[0-9]+(.[0-9]+)?$');
```

C.
```
SELECT AVG(TO_NUMBER(PRICE)) FROM PRODUCTS;
```

D.
```
SELECT AVG(CASE WHEN IS_DECIMAL(PRICE) THEN CAST(PRICE AS DECIMAL(10,2)) ELSE NULL END) FROM PRODUCTS;
```

□□□:
```
SELECT AVG(TO_DECIMAL(PRICE, 10, 2)) FROM PRODUCTS;
```

Answer: (SHOW ANSWER)
□□ B□ □□□ □□□ PRICE □□ □□□ □□ □□(□□□ □□□ □□□)□ □□□□ □□□ □□□□□ □□□□ □□ □□□□ □□□□ □□□□□. 'REGEXP LIKE'□ □□□ □□ □□ □□□□□ □□□ □ □□□ □□□□, 'CAST'□ □□ □□□ □□□ □□ □□□□ □□□ □ □□□□□. □□ D □□ □□□□, □□□□ DECIMAL□ □□□ □ □□□ □□□□ □□□ □□ □□(UDF)□ 'IS DECIMAL'□ □□□□ 'CASE' □□ □□□□ □□□□□. 'IS DECIMAL'□ □□□ PRICE□ DECIMAL□ □□□□□, □□□ □□□ NULL□ □□□□□. AVG □□□ NULL □□ □□□□ □□□□□ □□ □ □□ □□ □□□□□. □□ A□ 'DECIMAL' □□□ □□ □□□ □□□□ □□□ □□ □□□ □□□□ □□□. 'TO_NUMBER'□ 'TO_DECIMAL'□ □□ □□□ □□□□ □□□ □□□□□□ □□□□ □ □□□□. □□ E□ PRICE □□ □□□□□ □□□ □ □□ □□ □□□ □□□□□□, '□□□ □□ □□'□ □□□ □□□□□ □□ NULL□ □□□□□.

NEW QUESTION: 2
□□□ □□ □□ □□□ □□ □□□□ □□□ Snowsight □□□□□ □□□□□□. □ □□□□□ □□ □□□ □□□ □□□□ □□□, □□ □□□ □□□□ □ □ □□ □□□□ □□□ □□ □□□ □ □□ □ □□□ □□ □□□. □□□□□ □□□□ □□□□□ □□ □ □□ □□□ □□□□ □□□?
A. '□□ □□' □□□ □□□□ □□□□□ □□□ □□□□□. □□ □□, □□ □□□ □□□□□ □□□□ □□□ □□□□□□ □ □□□□ □□ '□□' □□□ □□□□□.
B. '□□ □□' □□□ □□□□ □□□ □□□□□ □□□□□. □□ □□□ □□□ □□□ □□□/□□ □□ 'SELECT' □□□ □□□□□□ □ □□□□ □□ 'USAGE' □□□ □□□ □□□□□. □□ □□ □□□□ □□□□ □□ □□ □□□□□ □□□ □□□ □□□□□□ □□□□□.
C. '□□ □□' □□□ □□□□ □□□□□ □□□ □□□□□. □□ □□ □□□ □□□□□ Snowsight□□ □□□□ □□ □□□□□□ □□ '□□□□' □□□ □□□□□.
D. □□□□□ PDF □□□ □□□□ □□□ □□ □□□□ □□□□ □□ □□□ □□□□□.
E. '□□ □□' □□□□ □□□□□ □□□□ □□ □□□ □□□□ □□□□.

Answer: B (LEAVE A REPLY)
'□□ □□'(B)□□ □□□□ □□□□ □□□□ □□□ □□□ □ □□□□. '□□ □□□/□□ □□ SELECT □□ □ □□□□□□/□□□□ □□ USAGE □□'□ □□□□ □□□□□ □□□□ □□□□ □□□ □□□ □ □□□□ □ □ □□□□. '□□ □□'(A)□ □□□□ □□□□□ □□□ □ □□□ □□□□□ □□□□□ □□□□. '□□□□'(C)□ □□□ □□□ □□ □□□ □□□□□ □□□□ □□ □□□□ □□ □□□□. PDF□ □□□□(D)□ □□□□□□□ □□ □□□□ □□□□ □□□□ □□ □□□ □□□□. E□ □□□ □□□ □□ □□□ □□□ □□□□ □□□ □□□□ □□□□.

NEW QUESTION: 3
□□□ □ □□□□□ □□□ □ □□ □ □□□□□ □□□□ □□ □□□ □□□ □□□□□?
A. □□□□ □□□ □□□□ □□ □, □□□□ □□□ □□□□.
B. □□□ □□□ □□□□ □□□□ □□□□ □□□□□.
C. □□□□ □□□□ □□□ □□ □□□ □□□□ □□□□□.
D. □□ □ □□□□□ □□ □□□□□ □□ □□□ □□□□□.
Answer: D (LEAVE A REPLY)
□□ □ □□□□□ □□□□□ □□ □□□ □□□□, □□ □□□ □ □□□□ □□ □□□ □□ □□□ □□□□□.

NEW QUESTION: 4
Snowflake□□ □□□□□ □□□ □□ □□□□ □□□□ □□□□. □□ □□□□□ □□□ □□□□□ □□□□ □□ □□□□□□. 'SALES' □□□□□ 'order_id', 'order date', 'product_id', 'quantity', 'price' □□ □□□□. □□□□ □□ □□□□ □□□ □□ □□□□ □□ □□□□ □□□ □, □□□ □□ □□ □□□ □□□ □□□□ □ □□ □□□□ SQL □□□ □□ □□□ □□ □ □□□□□?
A. □□□ □□ □□ □□□□ □□□□ □□□ □□□□ □□□ □□□□□ □□□ □□□ □□□ □□□□□. □□ □□ "DAYOFWEEK" □□□ □□□□□.
B. 'product_id'□ □□□ □□ □□□ □□□□ □□ □□ □□□□□ □□□ □□ □□□ □□□ □□□□□ □□□□□. □□ □□□ □□ 'PIVOT' □□□□ □□□□□.
C. □□□ □□ □□□ □□□□ □ □□ □□□ □□□ □□ □□□ □ □□ □□□ □□□ □□□ □□□ □□□ □□□□ □□□□□. 'SALES' □□□□ 'SHIPPING' □□□□ □□□□ 'order_date'□ 'ship_date'□ □□□ □□□□□.
D. □□□□ □□□ □□□□ □□□□ □□□ □□□ □□ □□ □□□ □□□ □ □ □□□ □□□□□□. 'INFORMATION SCHEMA.LOAD HISTORY'□ □□□□ □□□□ □□□ □□□ □□□ □□□ □□□.
E. □□□□□ □□□ □□ □□ □□ □□ □□□ □□□□ □□□□□ □□□□ □□□ □□ □□ □ □□□ □□□ □□ □ □□ □□□ □□□ □□□□□□.
Answer: A,B,C,E (LEAVE A REPLY)
□□ A, B, C, E□ □□ □□□□ □□□□. □□ A□ □□□□ □□□□ □ □□□ □□□. □□ B□ □□□ □□□ □□□□□. □□ C□ □□ □□ □□□ □□□□□. □□ E□ □□□□□ □□ □□□□ □□□□, □□ D□ □□□ □□ □□□ □□□ □□□ □□ □□□ □□□□□.

NEW QUESTION: 5
'SALES DATA'□□ □□□□ □□ □□ □□□ □□□□. 'SALES AMOUNT' □□□ □□□□ □□□□ □□□□ □□□. □□□□□, □□□□ 3□□□□ □□□ □□□□ 'SALES AMOUNT' □□ □□□ □□ □□□□□ □□□. Snowflake□□ □□□ □□□ □□□ □□□□ □□ □□□□ □□□ □□□□□?
A. □□□ □□ □□□ □□ □□□ □□□ □□□□ □□□ □□, WHERE □□ □□□ UPDATE □□ □□□□ □□□□ □□□□ □□□□□.
B. □ □□ □□□□ □□□□ □□□□ 'SALES_AMOUNT'□ □□ □□ □□□□□□ □□ □□□□□ □□□□□.
C. □□□ □□□ CASE □□□□ □□□□ □□ 'CREATE OR REPLACE TABLE AS SELECT (CTAS)' □□ □□□□ □□, □□ □□ □ □□□□ □□□□ □□□ □□□□ □□□□ □□□□□.
D. □□□□ □□ □□□□□ □□□□, □□□ □□□ □□ □ □□□ □□□ □□, □□□ □□□□ □□ Snowflake□ □□□□□.
E. □□□□ □□□□, □□ □□□□ □□, □□□□ □ □□□□ □□□ □, □□ CASE □□ □□□□, □□□□ □□□□ □□ □□□ □□ □□ □□ □□□□ □□□ □□□□ □□□□□.
Answer: C (LEAVE A REPLY)
□□ C□ □□ □□□□ □□ □□□□□. □□□ □□□ □□ CTAS □□ □□□□ □□□□ □ □□ □□□□ □□, □□ □□ □ □□□□ □□□ □ □□□□. 'CASE' □□□□ □□□□ □□□□ □□□ □□ □□□□ □□□ □ □□□□ □□ □□□ □□□□ □□□□□ □□□ □□□ □□□□. □□ A□ □□ □□□ □□□ □□, □□ B□ □□ □□□□□□ □ □□□ □□□□ □□□ □□□□□□, □

□ D□ □□ □□□□□ □□□□ □□□ □□□ □□□□□. □□ E□ C□ □□□□□ □□□□□ □□□ □□□□ □□□□ □□□ □□□□□□. □□□ □□ □□□ □□□ □□ □□ □□□ □□□□ □ □□□□□. CTAS □□□ □□ □ □□□ □□□ □□ □□□ □□□□□.

NEW QUESTION: 6
SQL□□ □□□ □□ □□(UDF)□ □□□ □□□ □□□ □□□□□?
A. □□ □□□ □□□ □□ SQL □□□ □□□ □□ □ □□□ □□□□□.
B. UDF□ □□□ □□□ □□ □□□ □□□□ □□□□□.
C. UDF□ □□□□ □□□□ □□□ □ □□□□.
D. UDF□ □□ □□□□ □□□□□.
Answer: (SHOW ANSWER)
□□□ □□ □□(UDF)□ □□□□ SQL □□□ □□□ □□ □ □□□ □□□□□, □□ □□□ □□□ □□ □□ □□□ □□□ □□□ □ □□□□.

NEW QUESTION: 7
□□ □□□ □□ □□□□ □□□□ □□□□. 'sales' □□□□□ 'product id'(INT), 'sale date'(DATE), 'sale_amount'(NUMBER) □□ □□□□. □ □□□ □□□ □□ □ □□□ □□□□ □□□ □□□□ □□□. □□□□ 'sale date', 'product_id', 'sale_amount', 'daily total', 'percentage_contribution'□ □□□□□ □□□. □□ Snowflake □□ □ □ □□□ □□□□ □□□□ □□ □□□□□?

□ SELECT sale_date, product_id, sale_amount, SUM(sale_amount) OVER (PARTITION BY sale_date) AS daily_total, (sale_amount / daily_total) 100 AS percentage_contribution FROM sales;

□ SELECT sale_date, product_id, sale_amount, SUM(sale_amount) OVER () AS daily_total, (sale_amount / daily_total) 100 AS percentage_contribution FROM sales;

□ SELECT sale_date, product_id, sale_amount, (SELECT SUM(sale_amount) FROM sales) AS daily_total, (sale_amount / daily_total) 100 AS percentage_contribution FROM sales;

□ SELECT s.sale_date, s.product_id, s.sale_amount, dt.daily_total, (s.sale_amount / dt.daily_total) 100 AS percentage_contribution FROM sales s JOIN (SELECT sale_date, SUM(sale_amount) AS daily_total FROM sales GROUP BY sale_date) dt ON s.sale_date = dt.sale_date;

□ SELECT sale_date, product_id, sale_amount, SUM(sale_amount) OVER (ORDER BY sale_date) AS daily_total, (sale_amount / daily_total) 100 AS percentage contribution FROM sales;

A. □□ A
B. □□ B
C. □□ C
D. □□ D
E. □□ E
Answer: A,D (LEAVE A REPLY)
□□ A□ D□ □□□□□. □□ A□ □□□ □□ OVER(PARTITION BY)□ □□□□ □ □□ □ □□□ □□□□□. □□ □ □□ □□□□ □ □□ □ □□□ □□□□ □□□ □□□□□□. □□ D□ □□ □□□ □□□□ □ □□ □ □□□ □□□ □, □ □□□ □□ □□□□ □□ □□□□□. □□ B□ □□□ □□ □□ □□□□ □ □□□ □ □□ □□□□□. □□ C□ □□ □□□□ □ □□□ □ □□ □□ □□ □□ □□ □□ □□□□□, □□ □□□ □□□□□. □□ E□ □□□□ □□□ □□□□□ 'sale_date'□ □□□□ □□□□□□ □□ □□ □□□ □□□□□.

NEW QUESTION: 8

□□□ □□□□ □□□ □□ □□□ Snowflake □□□ 'ORDERS'□ □□□□ □□□□ □□□ □□□□□. 'CREATE TABLE ORDERS (ORDER ID INT, CUSTOMER ID INT, ORDER DATE DATE, TOTAL_AMOUNT). □□□ □□□□ 'ORDER ID'□ □□□□ null□ □□□□ □□, 'CUSTOMER ID'□ 'CUSTOMERS' □□□(□ □□ 'CUSTOMER ID')□□ □□□ □□□ □□□□ □□, 'ORDER DATE'□ □□ □□□ □□□□ □□□. □□ □ Snowflake□□ □□□ □□ □□□ □□□□ □□ □□□□□ □□□ □□□ □□□□□? 'CREATE TABLE CUSTOMERS (CUSTOMER ID INT PRIMARY KEY, CUSTOMER_NAME VARCHAR(255));'.

- Add a PRIMARY KEY constraint on `ORDER_ID`, a FOREIGN KEY constraint on `CUSTOMER_ID` referencing `CUSTOMERS(CUSTOMER_ID)`, and use a CHECK constraint `ORDER_DATE <= CURRENT_DATE()`.
- Add a UNIQUE constraint on `ORDER_ID` and `NOT NULL` constraint, a FOREIGN KEY constraint on `CUSTOMER_ID` referencing `CUSTOMERS(CUSTOMER_ID)`, and create a TRIGGER to prevent future dates.
- Add a PRIMARY KEY constraint on `ORDER_ID`, a FOREIGN KEY constraint on `CUSTOMER_ID` referencing `CUSTOMERS(CUSTOMER_ID)`, and use a NOT NULL constraint for ORDER_DATE.
- Add a UNIQUE constraint and NOT NULL constraint on `ORDER_ID`, a FOREIGN KEY constraint on `CUSTOMER_ID` referencing `CUSTOMERS(CUSTOMER_ID)`, and create a view with a filter on `ORDER_DATE <= CURRENT_DATE()`.
- Add a UNIQUE constraint and NOT NULL constraint on `ORDER_ID`, a FOREIGN KEY constraint on `CUSTOMER_ID` referencing `CUSTOMERS(CUSTOMER_ID)`, and use a CHECK constraint `ORDER_DATE <= CURRENT_DATE()`.

A. □□ A
B. □□ B
C. □□ C
D. □□ D
E. □□ E
Answer: E (LEAVE A REPLY)
□□ E□ □□ □□□□ □□□□□□. □□ □□ □□□□ □□□□ □□□□□, □□ □□□□ □□□□□ □□ □□□ □ null □□ □□□ □□ □□ □□□ □□□□ □□□□□ □□□□.
'ORDER_DATE <=' □□□ □□□□ □□□ □ □□ □□ □□□ □□□ □□ □□□ □□□□ □□ □□□ □□□□ □□ □□ □□□□□.

NEW QUESTION: 9
□□□ □□□□ Snowflake Marketplace□□ □□ □□ □□ □□□ □□□□ Snowflake □□□□□□□ □□ □□□□ □□□□ □□□. □□□ □□ □□□ □□ □ □□□ □□□□ □□□□□. □□ SQL □□ □ □□ □□□□ Marketplace □□ □□□ □□ □□ □□□ □□□□ □□□ □□□□ □□□□□□□ □ □□□□ □□□□ □□ □□□□□ □□□ □□□ □□□□, Marketplace □□□□□ □□□ □□ □□□□□ □□ □□□ □□□□□?

A.
```
CREATE TABLE enriched_customers AS SELECT c. , m. FROM customer_table c JOIN marketplace_view m ON c.customer_id = m.customer_id;
```

B.
```
CREATE TABLE enriched_customers AS SELECT c. , m.age, m.income FROM customer_table c JOIN marketplace_view m ON c.customer_id = m.customer_id;
```

C.
```
CREATE VIEW enriched_customers AS SELECT c. , m.age, m.income FROM customer_table c JOIN marketplace_view m ON c.customer_id = m.customer_id;
```

D.
```
CREATE OR REPLACE TABLE enriched_customers AS SELECT c. , m.age, m.income FROM customer_table c JOIN marketplace_view m ON c.customer_id =
m.customer_id;
```

【□□□】:
```
CREATE TABLE enriched_customers AS SELECT c. , m.age, m.income FROM customer_table c, marketplace_view m WHERE c.customer_id = m.customer_id;
```

Answer: (SHOW ANSWER)

□□ D□ □□□ □□ □□□ □□ □□□□□ □□□□□. □ □□□□ □□□□□ □□ □□□□ □□□□, □□□□□□ □□□ □□□ □(□□, □□)□ □□□□□ □□□□ □□□□ □□□ □□□ □ □□□□. □□ □□□□□ JOIN □□□ □□□□□. □□□ □□□□ □□□□ □□□ □□ □ □□□ □□□□ □□□□ □□ □ □□ □ □□□□ □□□□ □□□.

NEW QUESTION: 10
□□ □□□, □□□, □□□ □ □□□ □□ □□□ □□□ □ □□□□□□ □□□□ □□□ □□ □ □□□□ □□□ □□□□□?
A. □□□ □□□□ □□□□□ □□□ □□□□ □□□ □□□ □□□□.
B. □□□□ □□ □□□ □□ □□□ □□□□□.
C. □□□ □□ □□□ □ □□ □□□ □□ □□□ □□□ □□ □□□ □□□□□.
D. □□□ □□ □□□ □□□ □ □□□□□□ □□□ □□□ □□□□□.
Answer: (SHOW ANSWER)
□□□ □□ □□□ □□□ □□□ □□ □□□ □□□□ □□□ □ □□□□□□ □ □□ □□□ □□□□□.

NEW QUESTION: 11
□□□□ □□□□ □□□ □□□ □□ □□ □□□ □□□□ 'CUSTOMER DATA'□□ □□□□ □□□□□□□ □□□ □□ □□ □□□□ □□□□ □□□. □□□ □□□□□□ 'CUSTOMER DATA TEST'□ □□□□ □□ □□□ □□□□ □□□. □□ Snowflake □□ □ □ □□□ □□□□ □□ □□□□ □□□ □□□□□?
A. `CREATE DATABASE CUSTOMER_DATA_TEST AS REPLICATION OF CUSTOMER_DATA;`
B. `CREATE DATABASE CUSTOMER_DATA_TEST AS COPY OF CUSTOMER_DATA;`
C. `CREATE DATABASE CUSTOMER_DATA_TEST CLONE CUSTOMER_DATA;`
D. `CREATE DATABASE CUSTOMER_DATA_TEST LIKE CUSTOMER_DATA;`
【□□】:
`CREATE OR REPLACE TABLE CUSTOMER_DATA_TEST AS SELECT FROM CUSTOMER_DATA;`
Answer: C (LEAVE A REPLY)
'CREATE DATABASE CUSTOMER DATA TEST CLONE CUSTOMER DATA;' □□□□ Snowflake□ □□ □□ □□□ □□□ □□□□ □□□ □□ □□□□□□. □□□□ □□□□□□□ □□□□□ □□□□ □□□□, □□□□ □□□ □□ □□□ □□□ □□ □□□□□ □□□□□. □□□ □□□□ □□□□ □□□□□ □□□□ □□□□□□ □□□ □□□ □□□ □□□□□□. □□ A□ □□/□ □ □ □□□ □□□□□. □□ B□ □□□ Snowflake □□□□ □□□□. □□ D□ □□□ □□□□ □□□□ □□□□ □□□□. □□ E□ □□ □□□□□□ □□□ □□□□ □□ □□□□ □ □□□□ □□ □□ □□ □□□□□□ □□ □□□□ □□□□□.

NEW QUESTION: 12
Snowflake□□ □□□ □□□, □□ □□□□□□ □□□ □□□□ □□□ □□□□ □□□ □□□□ □□□ □□□□. □ □□ □□ □□□ □□□□ □□□□ □□□, □□□□□ □□□ □□□□□□ □ □□□. □ □□□ □□□□□□ 0□□ □□ □ □□ □□□ □□□□□ □□□. □□ □□ □□ □□□ □□ □□□ □ □□ □ □□ □□□ □□ □□□□□□?
A. □□ □□□□□ □□□□ □□ □□□ □□□□, □□ □□□ □□□□ □□ □□□ □□□□ 0□ □□□ □□□ □□□□□.
B. □□ □□ □□ □□ □□□ □□□□ □□□ □□□□ □□□□, COALESCE □□□ □□□□ □□□ □□□ □□□□ 0□□ □□ □ □□ □□□ □□□□ □□□□ □□□ □□□□ LEFT JOIN□□□.
C. □□□ CTE(□□ □□□ □□□)□ □□□□ □□□ □□□ □□□ □□ □□ □□□□ UNION ALL□ □□□□, □□ □□□ □□□□ □□ NVL□ □□□□ □□□ □□□ □□□□ 0□□ □□□□.
D. 'GENERATE_SERIES' □□(UDF□ □□ □□ □□□ □□)□ □□□□ □□ □□□□ □□□□, □□□□ □□□ □□□□ LEFT JOIN□ □□□ □□, COALESCE□ □□□□ □□ □□ □□ □□□ □ □□ □□□□ 0□□ □□□□.
E. LATERAL FLATTEN □□□ □□□□ □□ □□□ □□□□, □□ □□□□ □□□ □□□□ LEFT JOIN□ □□, ZEROIFNULL□ □□□□ □□□ □□□ □□□□ □□ □□ □□ 0□□ □□□□.
Answer: B (LEAVE A REPLY)
□□□ □□ □□□□ □□□□ COALESCE□ □□□ LEFT JOIN□ □□ □□□□ □□ □□□□□. □□ □□□□(A)□ □□□□□ □□□ □□□ □□□□ □□□ □□□□. □□ CTE(C)□ □□□ □□ □□ □ □□□□. UDF □□ Lateral Flatten(D □ E)□ □□□□ □□ □□ JOIN□ □□ □□□□ □□□□□ □□□□ □ □□□□. □□□ Snowflake□ □□□□ □□ □□ □□□ □□□□ □□□□.

NEW QUESTION: 13
□□□□ □□□ □□ □□□ □□□□□□□□ □□□ □□□ □□□□ □□ □ □□□□□?
A. □□□ □□□ □□□□ □□ □□□ □□□ □□□ □□□□□.
B. □□□ □□□ □□□□ □□□ □□ □□□ □□□□□.
C. □□□ □□□ □□□□ □□ □□□ □□□ □□□□□.
D. □□□ □□□ □□□□ □□ □□□□ □□□ □□□ □□□.
Answer: (SHOW ANSWER)
□□□ □□□ □□□□ □□ □□□□ □□ □□□□□□ □□□ □□ □□□ □□□□ □ □□□ □□□.

NEW QUESTION: 14
BI □□□□□ □□ □□□ □□□□□ □□□ □□ □□ □□□□ □□□ □□□ □□□□ □□ □ □□ □□□ □□□□□?
A. □□□ □□□ □□□ □□□ □□
B. □□□□ □□□ □□□ □□□ □□□□□
C. □□ □□□ □□□□ □□□□ □□□ □□
D. □□□ □□□ □□□ □□□ □□
Answer: D (LEAVE A REPLY)
□□□□ □□□ □□□ □□□ □□□□□ □□ □□□ □□□□ □□□□ □□□ □ □□□, □□ □ □□□ □□□□□.

NEW QUESTION: 15
Snowflake□ 'LOG DATA'□□ □□□□ □□, 'log_message'□□ 'VARIANT' □□ □□□□. □ □□□ □□□□□□ □□□ JSON □□□□ □□□□□. 'log_message' □□□ 'context'□□ □□□ JSON □□□ □□□□, □ □□□ □□ □□ □□□ □□□ □□ □□ □□□□ □□ □□ □□□□. 'context' □□ □□□ 'user_id', 'session_id' □/□□ 'request_id'□ □□□ □□ □-□ □□ □□□□□. □□□ □ □ □□□□ □□□□ □□ □□□ 'user_id'□ 'request_id' □□□ □□□□ □□□. □□ 'context' □□ □□ □□ ID □ □□□□ □□□ □□(NULL□ □□□)□ □□□□ □□□. □□□ □□□ □□ □□□ □□□□ □□□ □□ □□ □□□□□. □ □□□ □□□ □ □□ □□ □□□ □□□□□□.
□ ```sql SELECT DISTINCT log_message:context.user_id::STRING, log_message:context.session_id::STRING, log_message:context.request_id::STRING FROM LOG_DATA; ```

□ ```sql SELECT DISTINCT TRY_TO_STRING(log_message:context.user_id), TRY_TO_STRING(log_message:context.session_id), TRY_TO_STRING(log_message:context.request_id) FROM LOG_DATA; ```

□ ```sql SELECT DISTINCT NVL(log_message:context.user_id::STRING, 'NULL'), NVL(log_message:context.session_id::STRING, 'NULL'), NVL(log_message:context.request_id::STRING, 'NULL') FROM LOG_DATA; ```

□ ```sql SELECT DISTINCT IFF(IS_NULL(log_message:context.user_id), 'NULL', log_message:context.user_id::STRING), IFF(IS_NULL(log_message:context.session_id), 'NULL', log_message:context.session_id::STRING), IFF(IS_NULL(log_message:context.request_id), 'NULL', log_message:context.request_id::STRING) FROM LOG_DATA; ```

□ ```sql SELECT DISTINCT COALESCE(log_message:context.user_id::STRING, 'NULL'), COALESCE(log_message:context.session_id::STRING, 'NULL'), COALESCE(log_message:context.request_id::STRING, 'NULL') FROM LOG_DATA; ```
A. □□ A

B. □□ B
C. □□ C
D. □□ D
E. □□ E
Answer: B,E (LEAVE A REPLY)
□□ B□ E□ □□ □□□ □□□□□□. □□ B□ 'TRY TO_STRING' □□□ □□□□□. □□ □□□ NULL□ □□, □□ □□ □□□□ NULL□ □□□□□. □□ E□ 'COALESCE' □□□ □□□□ 'context' □□□ □□□ 'context' □□ □□ ID□ □□□ □□□□ □□□ □□□□□ □□ □□□□□. □ □ □ □□□□ NULL□ □□, 'NULL' □□□□ □□□□□. □□ □□□□ □□□ □□□□ □□ □ □□ □□□ □□□□. □□ A□ NULL □□ □□□ □□ □□□ □□□□□ □□□ □□□□ □□□□. □□ C□ D□ □□□ □□□□□□, NVL □ IFF □□□ 'TRY TO_STRING' □ 'COALESCE' □□□□ □□□ □□□ □□□ □□ □□□ □□□□□. □□□ □□□ □□□ □□□□ □□□□.

NEW QUESTION: 16
□□□ □□□□ □□□□□□ □□ □□□ □□□□□ □□□ □□□. □□ □□□□ □□□ □□ □□ □□□□ □□, □□ □□□□ □□□ □□ □□□□. □□ □□□ □□□□ □□□?
A. □□/□□□
B. □□
C. □□
D. □□ □□
Answer: (SHOW ANSWER)
Snowsight□□ □□□□ □□□□□ □□□□□ □□(□□ □□ □□□)□ □□□□ □□□ □□□□ □□□ □□□□ □□□. □□□□ □□ □□ □□□□ □□ □□□ □□□□□ □□ □□□□ □□□ □ □□□□□ □□□ □□□□□ '□□ □□' □□□ □□□ □□□□□.
□□□ □□ □□□□ □□□□□ □□ □□□□□□ □□□□□ □□□□□ □□□ □□ □□□□.
□□ □□□ □□□□ "□□□□□"□□□. □□ □□□ □□□ SQL □□□ □□□□ □□□□ □□□□□ □□□□ □□ □□(□□ C)□ □□□ □□□□□□. □□ □□□ □□ □□□□ □□□ □□□□ □ "□□"□ □ □□□□. □□□ □□□□□ □□□□□ □□□□□□ □□□ □□□□□, □□ □□□□ □□□ □ □□□□ □□□ □□□ □□□□ □□ □□□ □ □□□ □□ □□ □□□□□.
□□□ □□□□:
* □□ A(□□/□□□)□ Snowsight□□ □□□□ □□ □□□ □□ □□ □□□ □□□ □□□□. □□ □□□ □□□□□ □□□ □□ □□ □□□ □□ □□□□□.
* □□ B(□□)□ □□□ □ □□ □□□□ □□□□□. □ □□□ □□□□ □□□□□ □□ □□□ "□□□" □□ □□ □□□ □□□□ □□□, □□□ □□□□□ □ □□□□ □□□□.
* □□ C(□□)□ □□□ □□□□□. □□□□□ □□□□ □□□ □□□□ □□ □□□□ □□□ □□□□ □□□□□. □□□ □□□□ □□□□ SQL□ □□ □□□□ □□□□ □□□□ □□ □□□□ □□□.
* □□ D□ 100% □□□□□. □□ □□ □□□ Snowsight□ "□□□ □□" □□□□□. □ □□□ □□□□ "□□ □□□ □□" □□□□□ □□□ □□□□ □□□ □□□ □□□ □ □□□□. □ □□□ □ □ □□□ □□□□□□ □□ □□ □□ □□□□ □□□□ □□ □ □□□□ □□ □□ □□□ □□□□ □□ □□□□□ □□□□□□.

NEW QUESTION: 17
Snowflake□□ □□ □□□ □□□ □□□□ □□□□. □□□ □□□□ '□□ ID', '□□ ID', '□□ □□' □ '□□ □□'□ □□□□ □□□□. □□ '□□ □□' □□ □□□ □□ □□□□□, □□ □□□□□ □□ □□□□. □□ '□□ ID', '□□ ID' □ '□□ □□'□ □□□ □□□□ □□□□□ □□□□ □□□□. □□□□□, □□□□ □□□ □□ □□ □□□ 10,000□□ □□□□□ □□□. □□□ □□□ □□ □□□ □□□□ □□ □□□□□ □□□ Snowflake SQL □ □□□ □□□□□?

- CREATE OR REPLACE TABLE sales AS SELECT customer_id, product_id, sale_date, ABS(sale_amount) AS sale_amount FROM sales WHERE sale_amount > 0 AND sale_amount < 10000 GROUP BY customer_id, product_id, sale_date, sale_amount;
- CREATE OR REPLACE TABLE sales AS SELECT customer_id, product_id, sale_date, IFF(sale_amount < 0, 0, sale_amount) AS sale_amount FROM sales WHERE sale_amount < 10000 QUALIFY ROW_NUMBER() OVER (PARTITION BY customer_id, product_id, sale_date ORDER BY sale_date) = 1;
- CREATE OR REPLACE TABLE sales AS SELECT customer_id, product_id, sale_date, CASE WHEN sale_amount < 0 THEN 0 ELSE sale_amount END AS sale_amount FROM sales WHERE sale_amount BETWEEN 0 AND 10000 QUALIFY ROW_NUMBER() OVER (PARTITION BY customer_id, product_id, sale_date ORDER BY sale_date) = 1;
- CREATE OR REPLACE TABLE sales AS SELECT customer_id, product_id, sale_date, ABS(sale_amount) AS sale_amount FROM sales WHERE ABS(sale_amount) < 10000 QUALIFY ROW_NUMBER() OVER (PARTITION BY customer_id, product_id, sale_date ORDER BY sale_date) = 1;
- CREATE OR REPLACE TABLE sales AS SELECT customer_id, product_id, sale_date, CASE WHEN sale_amount < 0 THEN 0 ELSE sale_amount END AS sale_amount FROM sales WHERE sale_amount < 10000 QUALIFY ROW_NUMBER() OVER (PARTITION BY customer_id, product_id, sale_date ORDER BY sale_date) = 1;

A. □□ D
B. □□ A
C. □□ C
D. □□ B
E. □□ E
Answer: E (LEAVE A REPLY)

□□ E□ □□□ □□ □□□ □□ □□□ □□□□ □□□□□. □□ □□ □□□ 0□□ □□□□, □□□□ □□(sale_amount < 10000)□ □□□□, QUALIFY □□ ROW_NUMBER() □□□ □□□ □□ □□ 'customer_id', 'product_id', 'sale_date'□ □□□□ □□ □□□□ □□□□ □ □□ □□□ □ □□ □□□ □□□□□. □□ A□ D□ □□ □□ □□□ □□□□ □□□□ □□□, □□ B□ IFF(sale_amount < 0, 0, sale_amount)□ □□□□□ sale_amount < 10000□ □□□□ □□□□ □□□, □□ C□ □□ □□□ 0□□ 10000 □□□ □□□□□.

NEW QUESTION: 18
□□□ □□(JSON □ CSV □□)□ □□ □□□□ □□□ TEXT □□ □□□ □□ Snowflake □□□ 'raw_data'□ □□□□ □□□□. □□□ □□ □□□□ □□ □□□ □□□□ □□□□ □□□, JSON □ □□ CSV □□□ □□□ □□□□ □□□. □□□□□, JSON □□ □□('{l'□ □□)□□□ 'user_id' □□ □□ □□□□, CSV □□ □□(□□□ □□)□□□ □ □□ □□(□□□ □□□ □□□□ □□)□ □ □ □□□□ □□□. □□ □□ □ □□□ □□□ □□□ □□ □□□□□ □□□□ □□□□ □□□ □□□□□?

- SELECT CASE WHEN STARTS_WITH(log_entry, '{') THEN PARSE_JSON(log_entry):user_id ELSE SPLIT(log_entry, ',')[1] END AS extracted_value FROM raw_data;
- SELECT CASE WHEN SUBSTR(log_entry, 1, 1) = '{' THEN PARSE_JSON(log_entry):user_id ELSE SPLIT_PART(log_entry, ',', 2) END AS extracted_value FROM raw_data;
- SELECT IFF(STARTS_WITH(log_entry, '{'),GET_PATH(log_entry, 'user_id'),SPLIT(log_entry, ',')[1]) AS extracted_value FROM raw_data;
- SELECT CASE WHEN STARTS_WITH(log_entry, '{') THEN log_entry:user_id ELSE SPLIT(log_entry, ',')[1] END AS extracted_value FROM raw_data;
- SELECT CASE WHEN LEFT(log_entry, 1) = '{' THEN GET(PARSE_JSON(log_entry), 'user_id') ELSE SPLIT_TO_TABLE(log_entry, ',')[2] END AS extracted_value FROM raw_data;

A. □□ A
B. □□ B
C. □□ C
D. □□ D

E. □□ E
Answer: (SHOW ANSWER)
□□ B□ □□ □□□□□ □□□ □□□□□□. `1, 1) = '{'`□ `STARTS_WITH`□ `PARSE_JSON`□ □□ □□□ □□□□ □□□ □□ □□□ '{'□ □□□□□ □□□□□ □□□□□. JSON □□□ □□, `VARIANT` □□□□ 'user_id'□ □□ □□□□ □ □□□□ □□□□□. CSV □□□ □□, □□□ □□□□□ □□ □□□ □□□□□ □□□ □□□ `,', 2)`□ □□□□, □□ `SPLIT`□□ □□□□□□. □ □ A□ 'log_entry' □□ JSON □□□ □□□ □□□□ □□□□ □□□□ □□□ □□□ □□□□ `PARSE_JSON`□ □□□□ □□□□□ □□□□ □□□. □□ C□ JSON□ □□ □□□□ □□□ □□□ □ □□ `GET_PATH`□ □□□□□. □□ D□ `TEXT` □□ JSON □ □□□□ □□ □□□ □ □□□□ □□□ □□□□□. □□ E□ `SPLIT TO TABLE`□ □□□□□, □□ `SPLIT PART`□□ □□□ □ □ □□□.

NEW QUESTION: 19
Snowflake Marketplace□□ □□ □□□□ □□□□ □□□□ □ □□ □□□ □□□ □□ □□□ □□□□ □□□□□ □□□□ □□ □□□□ □□□ □ □□□□?
A. □□ □□□ □□□ □□□□□□
B. □□ □□□ □□□
C. □□□□□□ □□□□ □□□ □ □□
D. □□□ □□ □□□ □□
Answer: (SHOW ANSWER)
□□ □□□ □□□ □□ □□□□□ □□□□ □□ □□ □□ □□ □□□ □□ □□□ □□□ □□ □□□□ □□□ □ □□□□.

NEW QUESTION: 20
□□□ □□□□ □ □□□□ □ □□□□ □□□□ □□ □□□ □□ □□ □□ 3□□ □□□□ □□□ □□□□□. □□ □□□□ 'sales data'□□ □□□□ 'region', 'sales_rep', 'sales_amount' □□ □□□ □ □□□□. □ □□□ □□□□□ □□□ □ □□ Snowflake SQL □□ □□□□□?

```
] SELECT region, sales_rep, sales_amount FROM (SELECT region, sales_rep, sales_amount, ROW_NUMBER() OVER (PARTITION BY region ORDER BY sales_amount
ESC) as rn FROM sales_data) WHERE rn <= 3;

] SELECT region, sales_rep, sales_amount FROM (SELECT region, sales_rep, sales_amount, RANK() OVER (PARTITION BY region ORDER BY sales_amount DESC)
s rnk FROM sales_data) WHERE rnk <= 3;

] SELECT region, sales_rep, sales_amount FROM (SELECT region, sales_rep, sales_amount, DENSE_RANK() OVER (PARTITION BY region ORDER BY sales_amount
ESC) as dr FROM sales_data) WHERE dr <= 3;

] SELECT region, sales_rep, sales_amount, LAG(sales_amount, 1, 0) OVER (PARTITION BY region ORDER BY sales_amount DESC) as prev_sales FROM sales_data
HERE prev_sales <= 3;

] SELECT region, sales_rep, sales_amount FROM (SELECT region, sales_rep, sales_amount, NTILE(3) OVER (PARTITION BY region ORDER BY sales_amount DESC)
s tile FROM sales_data) WHERE tile = 1;
```


A. □□ A
B. □□ B
C. □□ C
D. □□ D
E. □□ E
Answer: A,B,C (LEAVE A REPLY)

□□ A, B, C□ □□□ □□□ □□□□ □□□□ □ □□ □ □□ □□□□ □□□ □□□□. `LAG`□ □□□ □□□ □□□□, `DENSE RANK()`□ □□□ □□ □□□ □□□ □□□□ □□ □□□ □□□ □, `DENSE RANK()`□ □□□ □□ □□□ □□□ □□□□□ □□□ □□□□ □□□□. □ □□ □□ □□ `WHERE` □□ □□□□ □□ 3□□ □□□□ □ □□□□. □□ D□ `LAG`□ □□ □□□□ □ □□ □□□ □□ □□□□. □□ E□ `NTILE`□ □□□□□, □□ □□ □ □□□□ □□□□ `tile = 1`□ □□□□ □□ □□ □□ □□ 3□□ □□ □□□□ □□ □ □□□□.

NEW QUESTION: 21
□□ □□ □□ □□□ □, □□ □□ □□ □ □□□ □□□□ □□□□ □□ □□□□ □□ □□□□ □ □□□ □□□ □□ □□□ □□□□ □□□?
A. □□ □□
B. □□ □□
C. □□ □□□
D. □□ □□ □□□ □□
Answer: B (LEAVE A REPLY)
Snowflake □□□ □□ □□□□□ □□□□ □□□□ □□□□ □□□ □□□ □□□ □□□□ □□ □□□ □□ □□□ □□□□□. □□□□ □ □□□ □□(RAP)□□ □□□□ □□ □□ □□□□ □ □, CURRENT_ROLE □□ CURRENT_USER□ □□ □□ □□□ □□□□ □□ □□□ □□□□ □□□ □□ □□□□□□ □□□ □□ □□ □□□□□.
□□ □□□ □□□ □□ □□□□□ □□□□ □□□□□ □□□□□□ CURRENT_ACCOUNT() □□□ □□□□ □□□□□. □ □□□ □□ □□□□ □□□□ □□□□ □□ □□ □□□□ □□□□□.
□ □□ □□ □□□ CURRENT_ACCOUNT() □□□ □□□□□(□: WHERE shared_account_id = CURRENT_ACCOUNT()), □□□ □□□□ □ □□□ □□□ □□□□ □□□ □□ □ □□□ □ □ □□ □ □□□ □ □□□□.
□□□ □□□□:
* □□ A □ D: CURRENT_ROLE □ CURRENT_AVAILABLE_ROLES□ □□□□□ □□ □□ □□□ □□□□□. □□□□ □□□□ □□ □□□ □□□ □□□ □□□ □□□ □□□ □□□□ □□□□ □□□ □□□ □□ □ □ □□ □□□□ □□□ □ □□□□.
* □□ C: CURRENT_USER □□ □□ □□□ □□□ □□□□□□ □□□□□.
* □□ B: 100% □□□□□. □□ □□□ □□ □□□□ □□□ □□□ □□□□□ □□□ □□ □□□ □□□□, □□□ □□□□□ □□□ □□□ □□□□ □□□. □□□ □□□□ □ □□ □□□□ □□ □□□ □□□□, "□□□ □□□ □□"□ □□□□ □□□□ □□□□□ □□□□□.

NEW QUESTION: 22
Snowflake□ '□□ □□□' □□□□□ □□ □□□ □□□ □□ □□ □□ □□(PII)□ □□□□ □□□□. □□ □□□□ □ □□□□ □□ □□ □□□ □□□ □□□ □□□□ □□□. □□□ □□□ □□ □□□□ □□□□, □□ □□ ID□ □□□□ □□ □□□□□ □□ □□□□ □□□□ □□□. □□□ □□□ □□□□ □□□□□ □□□□ □□ □□ Snowflake □□□ □□□ □□□□ □□□□ □□□?
A. Snowflake□ □□□ □□□ □□□ □□□□ □□ □□ □□ □□ □□ □□(PII) □□ □□□□ □□□□□□.
B. Snowflake□ HASH □□□ □□□□ □□ ID□ □□□□□, □□□ □□ □□(UDF)□ □□□□ □□ □□□ □□□ □□□□ □□□□ □□□□ □□□ □ □□□□ □□□□□. □□□ □□□ □□□ □□ □□□□ □□□□ □□ □□ □□□ □□□□□.
C. 'customer_data' □□□□ □□ □□□ □□□□, □□ □□□□ □□(□: Python)□ □□□□ □□□□ □□□□ □□, □□□□ □□□□ Snowflake□ □□ □□□□□.
D. Snowflake□ □□ □□□ □□□□ □□ □□ □□ □□ □□□ □□□□ □□ □□□□ □□□□□.
E. Snowflake□ □□□ □□□ □ □□ □□□ □□□□ □□ □□□ □□ □□ □□□□, □□ □□□□ □□□□ □□□□ □ □ □□□ □□□.
Answer: B,D (LEAVE A REPLY)
□□ B□ D□ □□□ □□□□□. □□ B□ □□ □□□□ □□□□ HASH□ □□□□ □□ □□□□ □□□□□□ □□□□ □□□□ □□ □□ □□□ □□□□□ □□□□□. □□□ □□□ □□□ □□ □□ □□□ □□□□□. □□ D□ □□ □□□ □□□□ □□ □□□ □□□□ □□□□□. □□ A□ E□ □□ □□□□ □□□□□, □□□ □□□□ □□□ □□□ □□□□ □□□□ □□ □□ □□□□ □ □□ □□ □□(PII)□ □□□ □ □□□□. □□ C□ □□□ □□□□□ □□□□□ □□□ □□ □□□ □□ □□□□ □□□□□.

NEW QUESTION: 23
□□ □□□ □□□ 'SALES DATA' □□□□ □□□□ □□□□ □□□. 'SALE AMOUNT' □□ VARCHAR□ □□□□ □□□, □□□□ NUMERIC □□□□□ □□□. □□ □□□ □□□ □□ □□□ 'NULL'□ □□□ NULL □□ □□□□ □□□□. 'SALE AMOUNT' □□ NUMERIC□□ □□□□ □ □□□ □□□□ □□ □□□□ Snowflake SQL□ □□□□ □□□ □□□ □□□□ □□□. □□ SQL

□ □ □□□□ □□□□ □ □□□ □□ 'SALE AMOUNT' □□ □□□□□ □□□□ □□□□ □□ □□□□□? □□: 'REPLACE INVALID CHARACTERS' □□□ □□ □□(UDF)□ □□□ □□ □□□ □ □□□□ □□□□ □□□□□ □□□□□.

A. ○ ```sql CREATE OR REPLACE TEMPORARY TABLE INVALID_SALES AS SELECT FROM SALES_DATA WHERE TRY_TO_NUMBER(REPLACE_INVALID_CHARACTERS(SALE_AMOUNT)) IS NULL AND SALE_AMOUNT <> 'NULL'; UPDATE SALES_DATA SET SALE_AMOUNT = '0' WHERE SALE_AMOUNT IN (SELECT SALE_AMOUNT FROM INVALID_SALES); DELETE FROM SALES_DATA WHERE SALE_AMOUNT = 'NULL'; ```

B. ○ ```sql UPDATE SALES_DATA SET SALE_AMOUNT = NULL WHERE SALE_AMOUNT = 'NULL'; CREATE OR REPLACE TEMPORARY TABLE INVALID_SALES AS SELECT FROM SALES_DATA WHERE TRY_TO_NUMBER(REPLACE_INVALID_CHARACTERS(SALE_AMOUNT)) IS NULL AND SALE_AMOUNT IS NOT NULL; UPDATE SALES_DATA SET SALE_AMOUNT = '0' WHERE SALE_AMOUNT IN (SELECT SALE_AMOUNT FROM INVALID_SALES); ```

C. ○ ```sql CREATE OR REPLACE TEMPORARY TABLE INVALID_SALES AS SELECT FROM SALES_DATA WHERE NOT REGEXP_LIKE(SALE_AMOUNT, '^[0-9]+(\.[0-9]+)?$') AND SALE_AMOUNT <> 'NULL'; UPDATE SALES_DATA SET SALE_AMOUNT = '0' WHERE SALE_AMOUNT IN (SELECT SALE_AMOUNT FROM INVALID_SALES); UPDATE SALES_DATA SET SALE_AMOUNT = NULL WHERE SALE_AMOUNT = 'NULL'; ```

D. ○ ```sql UPDATE SALES_DATA SET SALE_AMOUNT = NULL WHERE SALE_AMOUNT = 'NULL'; CREATE OR REPLACE TEMPORARY TABLE INVALID_SALES AS SELECT FROM SALES_DATA WHERE TRY_TO_DECIMAL(REPLACE_INVALID_CHARACTERS(SALE_AMOUNT),38,2) IS NULL; UPDATE SALES_DATA SET SALE_AMOUNT = '0' WHERE SALE_AMOUNT IN (SELECT SALE_AMOUNT FROM INVALID_SALES); ```

[□□□]:

○ ```sql CREATE OR REPLACE TEMPORARY TABLE INVALID_SALES AS SELECT FROM SALES_DATA WHERE TRY_TO_NUMBER(REPLACE_INVALID_CHARACTERS(SALE_AMOUNT)) IS NULL; UPDATE SALES_DATA SET SALE_AMOUNT = '0' WHERE SALE_AMOUNT IN (SELECT SALE_AMOUNT FROM INVALID_SALES); ```

Answer: D (LEAVE A REPLY)

□□ D□ □□□ □□ □□□ □□□□ □□□□□. □□, 'NULL' □□□ □□ □□ NULL □□□ □□□□□. □□ □□, □□□□ □□ □□□ □□□ □ '□□ □□(SALE AMOUNT)'□ □□□ □□□ □ □ □ □□□ □□□□ □□ □□□ 'INVALID SALES'□ □□□□□. □□□□□, □□ □□□□□ □□□ □□□□ □□ □□□ □□□ '0'□□ □□□□□□□. □□□ □ □□ □□ □□ □□□□ □□ TRY_TO_DECIMAL□ □□□□ □□ □□□□□.

NEW QUESTION: 24

□□□□ □□□□□ □□□□ □□□ □□□ □□□□□ □□□ □□□□. □□□, □□, □□ □□□□□ □□ □ □□ □□□ □□ □□□ □□□□ □□□. □□□ □□□ □ □□ □□□□□ □□□□ □ □□, □ □□ □□□□ □□ □□□□□□ □□□ □□ □□□□□□ □□ □ □□□□ □□ □□ □□□ □□□□ □□ □□□. □□ □ Snowflake□□ □□□ □□□ □□ □□□□□□ □□□□ □ □□ □ □□ □□□ □□□□□? (□□ □□)

A. □□□ □□□□ □□□□ □□ □□ □□ □□□□ □□□□.

B. □□□ □□ □□□ □□ 'MERGE □'□ □□□□ □□ □□□□ □□□□ □□□□ □□□□□□□ □□□□□.

C. □□□ □□□ 'INSERT INTO' □□□ □□□□□.

D. □□ □□□□ □□□□ □□ □□□□□□ □□□□ □□□□ □□ □□ □□ □□ □□□□ □□□□ □□□ □□□.

E. □□□ □□ □□□ □□ □□ □□□□ □□□□□.

Answer: A,B,D (LEAVE A REPLY)

□□ A, B, D□ □□□□□. □ □□ □□ □□ □□□□ □□□ □(A)□ □□□ □□□ □□□□□. 'MERGE □'□ □□□□ □(B)□ □□□□□ □□□ □□ □□□□ □□ □□□ □□□□ □□ □□□□ □□ □□□ □□□ □□□□□ □□□. □□□ □□□□ □(D)□ □□ □□ □ □□□ □□ □□□□□□ □□ □□ □□□□ □□□□ □□□□ □□□□□. □□ C□ □□□ 'INSERT INTO'□ □□□□□

□□□ □ □□□□ □□□□ □□□□ □□□□ □□□□. □□ E□ □□ □□ □ □□ □□□□ □□□□□ □□□□ □□□□□ □□□□□ □□□□. □□ □□□□ □□□□ □□, □□ □□□□□ □□ □□ □□□ □□□□ □□□□ □□□ □□□□ □□□□ □□□□. □□□ □□□□ □□□□ □□□ □□□ □□□□ □□□ □ □□□□. MERGE□ □□□□ □□□□ □□□□ □□ □□ □□□□ □ □□ □□ □ □□□□□ □□ □ □□□□□□ □□□ □□□ □□□□□.

NEW QUESTION: 25
Snowflake□ 'sensor_data'□□ □□□□ □□, 'reading'□□□ □□ JSON □□□□ □□□□ □□□□. JSON □□□ □□□□□, □□□ □□ □□(UDF)□ □□□□ □□ □□ □□ 'temperature'□ □□ □□□ □□□. 'temperature'□ □□□ 'sensor_type'□ □□ □□ □ □□□□. □□ □□□ '$.metrics.temperature'□ □□□ □□□□, □□ □□□ '$.reading.temp_c'□ □□□□□. □□ □□□ 'sensor_type' □□ □□□□ □□□□. JSON 'reading'□ 'sensor_type'□ □□□□ □□ □□□ □□□□, JSON□ □□ □□□ □□□□ □□□ NULL□ □□□□ UDF□ □□□ □□□□. □□□ □□□ □□ JavaScript UDF□ Snowflake□ JSON □□ □□□ □□□□ □ □□□ □□□ □□□ □ □□□□?

- ```sql CREATE OR REPLACE FUNCTION get_temperature(reading VARCHAR, sensor_type VARCHAR) RETURNS FLOAT LANGUAGE JAVASCRIPT AS $$ var reading_json = JSON.parse(reading); if (sensor_type === 'type1') { return reading_json.metrics.temperature; } else if (sensor_type === 'type2') { return reading_json.reading.temp_c; } else { return null; } $$; ```
- ```sql CREATE OR REPLACE FUNCTION get_temperature(reading VARIANT, sensor_type VARCHAR) RETURNS FLOAT LANGUAGE JAVASCRIPT AS $$ if (sensor_type === 'type1') { return reading.metrics.temperature; } else if (sensor_type === 'type2') { return reading.reading.temp_c; } else { return null; } $$; ```
- ```sql CREATE OR REPLACE FUNCTION get_temperature(reading VARIANT, sensor_type VARCHAR) RETURNS FLOAT LANGUAGE JAVASCRIPT AS $$ try { if (sensor_type === 'type1') { return reading.metrics.temperature; } else if (sensor_type === 'type2') { return reading.reading.temp_c; } else { return null; } } catch (e) { return null; } $$; ```
- ```sql CREATE OR REPLACE FUNCTION get_temperature(reading VARCHAR, sensor_type VARCHAR) RETURNS FLOAT LANGUAGE JAVASCRIPT AS $$ var reading_json = JSON.parse(reading); try { if (sensor_type === 'type1') { return reading_json.metrics.temperature; } else if (sensor_type === 'type2') { return reading_json.reading.temp_c; } else { return null; } } catch (e) { return null; } $$; ```
- ```sql CREATE OR REPLACE FUNCTION get_temperature(reading VARIANT, sensor_type VARCHAR) RETURNS FLOAT LANGUAGE JAVASCRIPT AS $$ if (sensor_type === 'type1') { return reading['metrics']['temperature']; } else if (sensor_type === 'type2') { return reading['reading']['temp_c']; } else { return null; } $$; ```

A. □□ A
B. □□ B
C. □□ C
D. □□ D
E. □□ E
Answer: C (LEAVE A REPLY)
□□ C□ □□ □□□ □□□□□. '□□' □ □□ □□□ □□□□ 'VARIANT'□ □□□□□, Snowflake□ JSON □□□□ □□□□ 'VARIANT' □□□□ □□□□ □□□ □□□□ JSON □□□ □□□□ □. □□, □□□ □□□ JSON□ □□□□ □□ □□□ □□□□ □□ 'try...catch' □□□ □□□□ □□□ □□ 'NULL'□ □□□□□. □□ □□ □□□ □□ □□□ □□□□ □□ □□□□□. 'VARIANT'□ □□ □□□□ □□ □□□ □□ □□ □□□ □□□ □□□□□. □□ A□ D□ 'VARIANT'□ □□□□ □□□□ □□ □□□□□ □□□□□. □□ B□ 'try...catch' □□□ □□ □□ □□□ □□ □□□ □ □□ □ □□ □□ □□□ □□□□ □□□□. □□ E□ □□ □□□□ □□□□ □□□ □□ C□□ □□□□ □□□□□.

NEW QUESTION: 26
□□□ □□□ □□□□ □□□ □□□□ □□□□ □□□□□ □□ □□□ □□□ □□□□. □□□□ □□□□ □□□□□□ Snowpipe□ □□□□ 'WEB EVENTS'□□ Snowflake □□□□ □□□□□. 'WEB EVENTS' □□□□□ '□□□ □□□□□', '□□□ URL', '□□□ ID' □□ □□□□ □□□□. □□□□□ □□ □□□□□ □□□□□□□ □□□, □□□ □□□ □□□□ □□□□. □□ □ □□ □ □□ □□□□ □□□□ □ □□□□ □□□□ □□□ □□□ □□□□ □□ □□□□?

A. 'COPY INTO' □□ □□□□ □□ □□□□ □□□ □□ □□ □□(□: □□, □□ □□ □□)□ □□□□□ Snowpipe □□□ □□□□□□.
B. □□□□□ □□□ □□□□ □□ □□□□ □□□□ □□ □□□□ □□ □□ □□□ □□□ □□□□□.
C. Snowpipe □□□ □□□ □□□□ Snowflake □□ □□□□ □□□ □□□□.
D. □□□□ □□□ □□□□□□ □□□ □□ □□ '□ □□□' □□□□ □□ □□□□□ □□□.
E. Snowpipe□ 'AUTO_INGEST' □□□□ □□ □□ □□□ □□ □□□ □□□□□ □□□□ □□ □□□ □□ □□□ □□□□.
Answer: E (LEAVE A REPLY)
□□□□ □□ □□□ □□ □□□ □□□ □(□□ E)□ □□□□ □□□□ □□□ □□□ □□□□ □□ □□□, □□□ □□□□ □□□□. □□□□□ □□ □□□□□ □□□□□□□ □□□ □□□ □ □ □□□ □□□ □□□ □□ □□□ □□□□□ □□□□□. □□ □□ □□□ □□□□ □□ □□ □□□□□□□. Snowpipe □□ □□□, □□ □□□ □□ □□□□ □ □□, Snowpipe □□□ □□□□ □ □□ □□ □□ □□□ □ □ □□□□. □□□□□ □□□ □□ □□□ □□□ □□□□ □□□□□ □□□ □□□.

NEW QUESTION: 27
Snowflake□□ □□ □□ □□□ □□□□ □□□□ □□ □□ □□□□ □□□ □□□□□?
A. □□ □□□ □□ □□□ □□□□□.
B. □□□□ □□ □□□ □□□□ □ □□□ □□□.
C. □□□ □□□ □□□□ □□□ □□□ □□□.
D. □□ □□ □□ □ □□ □□□ □□□□ □ □□□ □□□.
Answer: D (LEAVE A REPLY)
□□ □□ □□□ □□□□ □□ □□ □□□ □□□□, □□ □□□ □□□□, □□ □□□ □□□□□ □□ □□□ □□□□□ □ □□□ □□□.

NEW QUESTION: 28
□□ □□□ □□□□ Snowflake Marketplace □□□ □□□ □□□□ □□□□. □□□□ □□ □□□□□□, □□ 'STOCK PRICES' □□□□ □ □□□□ □□□□ □□□□ □□□□□ □□□□ □□ □. Marketplace □□□ □□□ 'DATE'(□□), 'SYMBOL'(VARCHAR), 'PRICE'(NUMBER) □□ □□ 'MARKETPLACE STOCK PRICES'□□ □□ □□□□□. □□□□ 'STOCK PRICES' □□□□□ □□□ □□ □□□□. □□□ 'DATE'□ 'SYMBOL' □□ □□ □□ □□□ □□□□□ □□ □ □□□□ □□□□ □□□□ □ □□ □□□ Snowflake □□ □□ □□□ □□ □ □□□□□?
A. □□ □□□□ □□□ 'INSERT INTO STOCK_PRICES SELECT FROM' □□ □□□□ Snowflake □□□ □□□□□.
B. □□□ □□□□ □□ □□□ □□ □□□□ 'STOCK_PRICES' □□□□ □□□□ □□□□ Snowflake Pipe□ □□□□□.
C. □□ □□□□ Snowflake □□□□ □□□□ 'STOCK PRICES' □□□□ 'MERGE' □□ □□□□, 'SYMBOLS'□ 'DATE'□ □□□□ □□□□ □□□ □□□ □□ □□□□ □□□□□□, □□□□ □□ □ □ □□□□ □□□□□. □□ 'DATE'□ 'SYMBOL'□ □□ □□ □□□ □□□□ □□□ □□□□□.
D. Snowflake Marketplace □□□ □□□ □□□□ □□ □□□□□ □□ Snowpipe □□ □□ □□□ □□□□□□. Marketplace□ □□ □□□□□□ □ □□□ □□□□ □□□□□ □□□□ □□□□□ □.
E. '□□ □□' □□□□ □□□□ □□□ □□, □□ □□□□ □□ □□□□ □□□□ □□□□ □□□ □□□□ □□□□ □□□□ Snowflake □□□ □□□□□. □□□ □□ □□ □□□ □ □□□ □□ □□□ □□ □□ □ □□□ □□□□□.
Answer: C (LEAVE A REPLY)
□□ C□ □□ □□□□□. 'MERGE' □□ □□□□ □□□□ □□(□ □□ 'SYMBOL')□ □□ □ □□□□ □□□□ □□ □□□□ □□□□□ □ □□□□. □□ □□ □ □□□□ □□□□ □□□ □□ □ □ □□□□ □□□□□□□□. □□ □□ □□□ □□□ □□□□□. □□ A□ □□ □□□□ □□□ □□□□□ □□□ □□□□□. □□ B□ Snowpipe□ □□ □□ □□□□□□ □□□□ □□□□□ □□ □ □ □□□□. □□ D□ Snowpipe□ □□ □□□□□□ □□□ □□ □□ □□□□□□□ □□□□□□□ □□□□□□□. □□ E□ □□ □□□ □□□ □□□ □□ □□□□□, □□□ □□ □□□□ □ □□□□ □□□□□, □□□□□□ □□□ □□□□ □□ □□ □□□ □□□ □□□□ □□□□.

NEW QUESTION: 29
Snowsight□ □□□□ Snowflake□ □□□□ □□□ □, □ □□□ □□ □□□ □□□□□?
(□□□□ □□ □□□ □□□□□)

A. □□/□□ □□□□ □□□□ □□□ □□□□
B. □□ □□□ □□
C. □□□ □□□ □□
D. □□□□ □□□ □□□ □□□□□.
Answer: A,B (LEAVE A REPLY)
Snowsight□ □□ □□□ □□□ □□ □ □□ □□□□ □□□□ □□□□ □□□ □□□□□ □□□□□.

NEW QUESTION: 30
□□□ □□□□ Snowsight□ □□□□ □□ □□ □□□□□ □□□□□□. □□□□ □□□ □□ □□ □□□ □□ □□ □□ □□□□ □□ □□ □ □□ □□□□ □□□ □□□ □□□□□ □□□□ □ □□ □ □□□ □□□□□□.
□□□ □□ □□□ □□□ □□□□.

□□□□ □□□□□ □□ □□ □□□ □□□ □□□□□ □□□.
□□□ WHERE □□ □□□□ □□□ □□ □□□ □□□□□?
A. start_time >= dateadd('days', -7, SYSDATE())
B. start_time >= dateadd('days', -7, CURRENT_TIMESTAMP())
C. start_time = :date_filter□ □□
D. □□ □□ = :daterange□ □□
Answer: (SHOW ANSWER)
Snowflake□ □□ □ □□□□□□ Snowsight□□ □□□ □□□ □□□□□ □□□□□ □□ □□ □□□ □□ □□ □□□ □□□□ □□ □□□□□□. □□□ □□□ □□□□ □□□□ □□□ □□ □ □□□ □□□□ □□ SQL □□□ □□□□ □□□□ □□□ □□□□ □□(□: "□□ 7□"□□ "□□ 12□□"□ □□)□ □□□ □ □□□□.
□□ □□ □□□□□ □□ □□□ □□□□ `:daterange`□□□. □□□ □□□ □□ □□□□□ □□ □□ □□□ □□ □□, `:daterange` □□□□ □□□□ □□□ □□ □□□ □□□□ □□ □□ □□□ □□□ □□□. □□ □□□ □□□□ □□ □□□ □□, `:daterange`□ □□ □□□ □□□ □ □□□ □□□□□ □□□□□□□□. `WHERE` □□ □□□□ Snowflake□ □ □□□□ □□□□ □□□□ □ □□ □□□ □□□□ □□□□□ □□□ □□□ □□□□□.
□□□ □□□□:
* □□ A□ B□ □□□□□ □□(-7□)□ □□□□ □□□ □□□□ □□□□. □□□ □□□ □□□ □□□□ □□□□□ □□□□ □□□□. □□□□ UI□□ □□□ □□□□ □□□ □□□□ □□□ □□ □ □□□□ □□ □□□ □□ "□□□ □□ □□"□□ □□□ □□ □□□ □□□□ □□□□.
* □□ C□ Snowsight□□ ':date_filter'□ □□□ □□□ □□□□ □□□□ □□□□ □□□□. □□□□ □□ □□□□ □□□ □□ □□□ □□ □□ □□□, □□□□ □□□ □□ □□□ □□ □□□ □ □□□ □□□□□ □□□□.

* □□ D□ 100% □□□□□. WHERE <column> = :daterange □□□ Snowsight □□□ □□ □□□ □□□ □□□□ □ □□□□ □□□□ □□□□□. □□ □□ □□□□□ □□ □□□ □□□□ □□ □□ □□□ □□□□□ □□□ □□ □□ □□□□ □□ □□□ □□□□□.

NEW QUESTION: 31
Snowflake□□ □□□□ DML(□□□ □□ □□) □□□ □□□□ □□□ □□ □□□ □□□□□? (□□□□ □□ □□□ □□□□□□)
A. □□ □□□□ □□□ □□
B. □□ □□□ □□□□
C. □□□□ □□□ □□□□□
D. □ □□□ □□
Answer: B,C,D (LEAVE A REPLY)
Snowflake□□ □□□□ DML □□□□ □□□ □□, □□□□ □ □□□ □□□□□.

NEW QUESTION: 32
□□ □□□□ □□ □□□ □□□ □ □□ □□□□ □□□ □□□□□ □□ □□□ □□□□? (□□□□ □□ □□□ □□□□□□)
A. □□□ □□ □□ □□□ □□□
B. □□ □□□ □□□ □□ □□□□ □□ □□
C. □□ □□ □□ □□□□ □□□□ □□ □□□ □□□□□.
D. □□ □□□ □□ □□□ □□ □□□ □□□
Answer: A,B (LEAVE A REPLY)
□□ □□ □ □□ □□□ □□□□ □ □□ □□ □□□□ □□□ □□□□ □ □□□ □□□□ □□ □□ □□□□□.

NEW QUESTION: 33
Snowflake□□ Parquet □□□ □□□□ □□ □□ □□ □ □□□□ □□□ □□□□ □□□ □□□□□?
A. □□ □□ □□□ □□□□□
B. □□□ □□ □ □□ □□ □□□ □□□□□.
C. Parquet □□□ □□□□□ □□ □□□ □□□□□□.
D. □□ □□□ □□□□□ □□ □□ □□ □□□ □□□□.
Answer: D (LEAVE A REPLY)
Parquet □□□ Snowflake□□ □□□ □□□ □□□□ □□□□ □□ □□□ □□□ □□ □□ □□□ □□ □□ □□□ □□□□ □□□□ □□□□□□.

NEW QUESTION: 34
□□□ □□□□ □□ □□□□□ □□□(□□□ □□ □□□□□ □□ □□□□ □□□□ □□□)□ □□ □□□ □□□□□ □□□ □□□□□. □□ □ □□□□ □□□□ □ □□□ □ □□ □□□ □□ □□□ □□□□□. □□ □□□□ 'SALES DATA'□□ 'ORDER DATE(DATE)', 'PRODUCT CATEGORY(VARCHAR)', 'NUMBER' □□ □□□□. □□ SQL □□□ □□□□ □□ □□□ □□□ □□ □□□ □□ □ □□ □□□□ □□□□□□ □□□ □□ □ □□□□□?
A. □□□□ □□□ SQL □□□ □□□□ □□□ □□ □□□□.

B. □□□□ □□□ SQL □□□ □□□□ □□□ □□□□ □□□□.
C. 'SALES_DATR'□ □□□ □□□ □□□ □□□□ □□ □□□□□□.
D. □□□□ □□□□ □□ □□□□ □□ □□□□□□.
E. □□□□ □□ □□□□ □□□ □□□□□.
Answer: (SHOW ANSWER)
□□□□ □□ □□ □□□ □□□□□ □□ □□□□ □□ □□□ □□□ □□ □□□ □ □□□□. □□ □□ □□□□ □□□ □□□ □□□ □□ □□□ □□□ □□ □□□ □□ □□□□□. □□ □□ □□□ □□□ □ □□□□ □□□□ □□□□. □□ □□□□ Snowflake □□□ □□ □□□□ □□□□ □ □□□□□. □□ □□□□ □□□ □□ □□□□ □□□□ □□□□ □□□ □□□ □□ □□□ □□□□□. □□□ □□□ □□ □□ □□□ □□ □□ □□□ □□□□ □ □□□□□.

NEW QUESTION: 35
□□ □ □□ □□□ □□□ □□ □□□ □□□ □□□□□?
A. □□□ □□□ □□□ □□ □□□ □□□ □ □□□ □□ □□□□ □□□□□.
B. □□□ □□□ □□ □□□ □□□ □□□□ □□□ □□□.
C. □□□ □□ □□ □□□ □□□ □□□ □□□□□.
D. □□ □ □□□ □□□ □□□□ □□□□ □□□ □□ □□□ □□□□□.
Answer: A (LEAVE A REPLY)
□□ □ □□ □□□ □□ □□□ □□ □□□ □□□ □□□ □ □□ □□□□ □□□□□.

NEW QUESTION: 36
□ □□ □□□ Snowflake□ □□□□ □□ □□ □□□□ □□□□ □□□□. □ □□□ □□ □□□□ □□□□ □□□□ □□□□ □□□□ □□□□ □□ □ □□□□ □□□□□ □□□. □□ □□□□ 'SALES DATA'□□ □□□□ 'SALE DATE'(□□)□ 'SALES AMOUNT'(□□) □□ □□□□ □□□□. □□ SQL □ □ Snowflake□ □□ □□□ □□□□ 95% □□ □□□□ □□□□ □□ □□□□□ □ □□ □□□ □□□□□?

```
ONFIDENCE = 0.95; SELECT FROM TABLE(FORECAST(MODEL_NAME=>'SALES_FORECAST_MODEL', INPUT => SELECT DATEADD(day, seq4(), CURRENT_DATE()) AS SALE_DATE
ROM TABLE(GENERATOR(timeLimit => 1))));

CREATE OR REPLACE MODEL SALES_FORECAST_MODEL USING TABLE SALES_DATA TARGET SALES_AMOUNT INPUT SALE_DATE HORIZON = 7 CONFIDENCE = 0.95; SELECT
ROM TABLE(FORECAST(MODEL_NAME=>'SALES_FORECAST_MODEL', INPUT => SELECT DATEADD(day, seq4(), CURRENT_DATE()) AS SALE_DATE FROM
ABLE(GENERATOR(timeLimit => 1))));

CREATE OR REPLACE MODEL SALES_FORECAST_MODEL USING TABLE SALES_DATA TARGET SALES_AMOUNT INPUT SALE_DATE HORIZON = 7; SELECT FROM
ABLE(FORECAST(MODEL_NAME=>'SALES_FORECAST_MODEL', INPUT => SELECT DATEADD(day, seq4(), CURRENT_DATE()) AS SALE_DATE FROM TABLE(GENERATOR(timeLimit
> 1))));

CREATE OR REPLACE MODEL SALES_FORECAST_MODEL USING TABLE SALES_DATA TARGET SALES_AMOUNT INPUT SALE_DATE TIMESTAMP_INPUT = TRUE HORIZON = 7
ONFIDENCE = 0.95; SELECT FROM TABLE(FORECAST(MODEL_NAME=>'SALES_FORECAST_MODEL', INPUT => (SELECT DATEADD(day, row_number() OVER (ORDER BY NULL),
URRENT_DATE()) AS SALE_DATE FROM table(generator(rowcount => 7))));

CREATE OR REPLACE MODEL SALES_FORECAST_MODEL USING TABLE SALES_DATA TARGET SALES_AMOUNT INPUT SALE_DATE TIMESTAMP_INPUT = TRUE HORIZON = 7
ONFIDENCE = 0.95; SELECT FROM TABLE(FORECAST(MODEL_NAME=>'SALES_FORECAST_MODEL', INPUT => (SELECT DATEADD(day, seq4(), CURRENT_DATE()) AS SALE_DATE
```

A. □□ A
B. □□ B
C. □□ C

D. □□ D
E. □□ E
Answer: E (LEAVE A REPLY)
□□ E□ □□ □□□□□. □□ A□ □□□□□ 'timeLimit'□ □□□□□, □□ □□ □□□ □□□□ □□□. □□ E□ 'TIMESTAMP_INPUT = TRUE'□ □□□□ 'SALE_DATE' □□ □□□□ □□□□□. □□ □□ □□(7□)□ □□ □□(0.95)□ □□□□ □□□□, 'TABLE(GENERATOR(rowcount => 7))'□ □□□□ 7□ □□ □□□ □□□□□. □□ D□ 'OVER (ORDER BY NULL)'□ □□□□□, □□ □ □□□□ seq4()□□ □□□ □□□□□.

NEW QUESTION: 37
Snowflake□ □□ □ □ □□□□ □□ □□ □□ □□ □ □□□ □□ □□□□□? (□ □□ □□□□□□)
A. □□ □□ □□□□ □□ □□ □□□ □ □ □□□ □□□ □□, □□□□ □□ □□ □□□ □□□□□ □□ □□ □□□ □□□□ □□□□□.
B. □□□□ □□ □□□□□ □□□ □ □□□, □□ □□ □□□□ □□□ □ □□□□.
C. □□ □□ □□ □□ □□ □□ □□□□□ □□□□□, □□□□ □□ □□□□□ □□□ □□□□□□ □□ □□□ □□□□□□.
D. □□ □□ □□□□ □ □□ □□ □□□□ □□□ □□□ □□ □□□□ □□ □□□ □ □□□□.
E. □□ □□ □ □□□ □□□ □□ □□□□ □□□□ □□□□ □□□ □□□□ □□□ □ □□□□. □□□□ □□ □ □□□ □□□ □□ □□□ □ □□□□.
Answer: (SHOW ANSWER)
□□ □□ □□ □□□ □□□ □□□ □□ □□□ □□□□□. □□□□ □□ □□□□ □□ □□□□ □□□□ □□□ □□□□□□. □□ □□ □□ □□□ □□□□□ □□ □□□□□ □□□ □ □□□ □. □□□□ □□ Snowflake□□ □□□□ □□ □□□□□. □□□□ □□ SELECT □□□□ WHERE □□ □□□ □□ □□□ □□□□ □□□ □ □□□□. □□□□ □□□□ □ □□□ □□□ □□□ □ □□□□.

NEW QUESTION: 38
□□ □□□ □□□ □□□□□ □□□, REST API, □□ □□ □ □□□ □□□ □□□ □□ □□□□ Snowflake□ □□□□□□□□ □□□□. □ □□□ Snowflake□ □□□ □□□□ □□□□ □□□ □ □ □ □ □□□ □□□□□ □□□. □□□ □□□□□ □□ □□□□ □□□□ □□□□ □□□□ □□□□ Snowflake □□□ □□ □□□ □□□ □□□□□?
A. AWS Lambda □□□ □□□□ S3□□ □□□ □□□□ REST API □□ □□□ □□ Snowpipe□ □□ □□□□ □ □□□□□ □□□□ □□□□ Snowpark □□□□□.
B. □□□□□ □□ □□□□ □□□□ □□ □□□□□□ □□□□□ □□□□ □□□□□ □□□ Snowflake □□□, □□ □□□ □□□ □□ □□□□ □□ □□□□□.
C. □□ □□□□ □□ □□□ □ □□ □□ □□□□, □□□□ □□ □ □□□□ □□ □□□□□ □□□□ □□□□□.
D. Informatica □□ Matillion□ □□ □□ ETL □□□ □□□□ JDBC/ODBC □□□ □□ □□□□ □□, □□ □ Snowflake□ □□□□, □□□□ □□□ □□ □□□ □□□□□.
E. □□ □□ □□ □□ □□□ □□□□ Snowpipe, Snowflake□ □□□□□□ REST API □□□□ □□ Kafka □□□, □□□ □□□□ □□□□□ □□□ □□□□□ □□□ □□□ □□□□ □□ Snowflake □□ □□□ □□□□ □□□ □□ Python □□□□□ □□□□ □□□□.
Answer: A,E (LEAVE A REPLY)
□□ A□ E□ □□ □□□□ □□□□ □□□□ □□□□□. □□ □□ □□□ □□ Snowpipe□ □□ □□□ □□□□□ □□□□□. Kafka □□□□ REST API □□□□ □□□□□ □□□ □ □□□ □□ □□□. AWS Lambda□ Snowflake □□ □□□ □□□□□ □□□ □□□ Snowflake□ □□□□ □□□□ □□□ □□□ □□□□□. □□ B□ □□□ □□□ □□□□ □□ □□□□□, □□ □□□□ □ □ □□□ □□□ □□□□ □□□□ □□□□. □□ C□ □□ □□□□ □□ □□□□ □□ □□ □□□□ □□□□□ □□□ □□ □ □□ □□□ □□□ □ □□□□. □□ D□ □□ ETL □□□ □□□□ □ □□□ □□□□ □□□□ Snowflake□ □□ □□□ □□□□□ □□□□ □□□□.

NEW QUESTION: 39
□□□ □□□ □□□ □□ □□□□□ □□□ □□□□ □□□ □□□ □□□□ □□□ □□□□□. □ □□□□□ □□□ □□□□ □□ □□□ □□ □□□□ □□□□ □□□. □□ □ □□ □□□ □□ □ □□□ □□□□□?
A. □□□□ □□□□□ □□□□ Parquet □□□□ AWS S3 □□□ □□□□□.
B. □□□ □□ □□□□ □□□ □□□□□□.
C. □□ □□□□ □□□□ IoT □□ □□□.

D. □□ □□ □□ □ □□ □□□ □□□□ Salesforce □□□□□□.
E. □□ API□ □□ □□□ □□ □□□ □□ □□□□ □□□ □□□□□□□ □□□□□.
Answer: (SHOW ANSWER)
□□ A, B, C□ □□ □□□ □□□ □□ □□□ □□ □□□□□ □□ □□□ □□□ □□□□□. □□□□ □□□□□ □□□(A)□ □□□□□□□ □□□ □□□□□ □□ □□□□ □□□□□. □□ □□ □ □□ □□□(B)□ □□ □□□ □□□□ □□ □□□□ □□□□□. Salesforce □□□(C)□ □□ □□□□ □ □□ □□□ □□ □□□ □□□□□. HR □□ □□□(D)□ IoT □□ □□□(E)□ □□ □□ □ □□□ □□□ □□□□ □□□□□ □□□□.

NEW QUESTION: 40
Snowflake□□ □□ □□□ □□□□ JSON □□□□ □□, □□□ □ □□□ □ □□□ □ □□ □□□□ □□□□□? (□□□□ □□ □□□ □□□□□□.)
A. □□□ □□ □□ □□ □□
B. □□ □□□ □□ □□□ □□□
C. □□□ JSON □□□ □□ □□ □ □□ □ □□□□ □□□
D. □□ JSON □□ □□ □□□ □□ □□
Answer: (SHOW ANSWER)
Snowflake□□ JSON □□□□ □□□ □ □□□ □ □□ □□ □□ □□ □ □□□ JSON □□□ □□ □□ □ □□□ □□□□ □□ □□□ □□□□.

NEW QUESTION: 41
□□ □□□□□ □□ □□ □□□□ □□□ Parquet □□□ □□□□. □□□ □□□ □□ □□□□ □□□□□ Snowflake □□ □□ □□□□ □□□. '□□ □□□(SALES REP)' □□□ □□□□ □□ □ □□ □□ □□□□ □ □ □□□ □□, '□□□(MANAGER)' □□□ □□□□ □□ □□□□ □ □ □□□ □□ □□□. □□□ □□ □□ □□□ □□□□□ □□□ □, Parquet □□ □□□□ □ □□ □□ □□□ □□□□ □□ □□□□ □□□ □□□□□?
A. Parquet □□□ □□ □□□ □□□□ □□□□ □□ □□□□ □□□□□. □□□ □□□ □□ □□ □□□□□ □□□ □□□ □□□□ □□□ □□□ □□□ □□□□□. □□ □□□□ □□□□ □□ □□ □□□□□. □ □□□ □□ □□ □□□ □□□ □□□□□.
B. □□ □□□□ □□□□ □□ □□ □□□□□. □□□ □□□ □□□ □□ □□□□ □□□□□ □ □□□ □□□ □□□□□. □□ □□□ □□ □□□□ □□□□□. □□□□ 'ACCOUNTADMIN'□□ □□□□□.
C. Parquet □□□ □□ □□□□ □□ □□□ □□□□ □□ □□ □□□□□. □ □□□ WHERE □□ □□□□ □□□□ □□□ □□ □□□□□□. RLS □□□ □□□□ □□□□.
D. □□ □□□□ □□□□ □□ □□ □□□□□. □□ □□ □□□ □□□ □□□ □□ □□□□ □□□□□ □ □□□ □□□ □□□□ □□ □□□ □□ □□ □□□□□. □□□ □□□ □□□ □□□ □ □□□ □ □□□□ □□□. □□□□ 'ACCOUNTADMIN'□□ □□□□□.
E. 'JSON EXTRACT PATH TEXT'□ □□□□ Parquet □□□□ □□ □□□ □□ □□□□ □□ □□ □□□□□. □□ □□□□ JOIN □□□ □□□□ □□□□ □□□ □□ □□□□ □□□□□□.
Answer: D (LEAVE A REPLY)
□ □□□ □□(RAP)□ □ □□ □□□ □□ □□□□□□ □□□ □□ □□ □□□□□. □□□□ □□ □□ □□□□□ □□ □□□□ □□□□ □□□ □□ □□ □□□□□ □□□. ' □□□ □□ □□ □ □□ □□□□ □□ □□□ □□□□ □ □□□□□. □□ □□ □□ □□□□ □□ □□□ □□ □□□□□□ □□□□□. 'ACCOUNTADMIW'□ □□□□ □□□□ □□ □□ □□□ □□□□ □□□ □□ □□ □□ □□□□□. □□□ □□□ □□□□□ □□□□ □□□ □ □□□□□ □ □□□□ □□□□ □□□□.

NEW QUESTION: 42
□□ □□ □□□□ □□ □□ □□□ □□□□ □□□□. 'FACT SUBSCRIPTIONS' □□□□□ 'subscription_id', 'customer id', 'start date', 'end date', 'subscription_amount'□ □□ □□ □□□□. □□□□ □□□ □□ □□ □□(MRR)□ □□□□ □□□□□ □□□. □□□ □□□ □□□□ □□□□□ □□□ □□□ □□□ □□□□□ □□□. □□□ □□□ □□□□ □□ □□ □□ □□□ □□□□□ □□ □□□ □ □□ □□ □□□ □□□□□□.
A. 'DIM_DATE' □□□□ □□□□ 'start_date' □ 'end_date' □□ □□□□ 'FACT_SUBSCRIPTIONS' □□□□ □□ □□□□ □□□□□.
B. □□ □ □ □□□ □□□ □□□□ □□□ □□ □□□ 'FACT SUBSCRIPTION SNAPSHOTS'□ □□□□□. □ □□□□□ 'subscription_id', 'customer_id', 'snapshot_date', 'is_active', 'subscription_amount' □□□ □□□□□ □□□.

C. □□□ □□ □□□ □□□□ □□ 'FACT SUBSCRIPTIONS' □□□□□ 'start_date'□ 'end_date'□ VARCHAR □□ □□□□□.
D. 'subscription_id'□ □□ □□ □□ 'DIM_SUBSCRIPTIOW' □□□□ □□□□ □□ □□ □□□ □□ □□□□ □□□□ □□ □□□□ □□□□ □□□ □□□.
E. '□ ID', '□ □□□', '□ □□□'□ □□ □□ □□ 'DIM MONTH' □□□□ □□□□, 'start_date'□ □□ □□ □□□ □□□ □□□□ 'FACT SUBSCRIPTIONS' □□□□ □□ □□□□ □□□□□.
Answer: (SHOW ANSWER)
□□□ □□ □□□(□□ B)□ □□□□ □□ □□□ □□ □□ □□(MRR)□ □□□□ □□ □□□ □ □□□□. □□ □□ □□ □□□ □□□ □□□ □□□□□. 'DIM_MONTH' □□□(□□ E)□ □□□ □ □□ □□ □□□ □ □□□ □□□□□□. □□ A(□□□□ □□□□ □□ □□□□ 'DIM DATE'□ □□)□ □□ □□□ □□□□ □□□ □ □□□ MRR/□□□ □□□□ □□ □□□ □ □□□□. □□ □ VARCHAR □□□□ □□□□ □(□□ C)□ □□ □□ □□□□ □□ □□□ □□□□□. □□□ □□ □□□(□□ D)□ □□□ □□□ □□□ □□□ □□□ □□□□ □□□□.

NEW QUESTION: 43
□□□ □□□□ □□□ □□ □□ □□□ □□ □□□ □□□ 'SALES DATA'□ □□□□□□. □□□□ 5□ □□ □□□□□□, Snowflake □□□ □□□ □□ □□□ □□□□□ □□□□ □□□□. □□ □□ □□ □□□□ □□□ □□□□ □□□□ □□□. Snowflake□□ 'SALES DATA' □□□□ □□□□ □□ □□□□□ □□□ □ □□ □□□ □□□□□?
A. 'UNDROP TABLE SALES DATA' □□□□ □□□□ □□□□ □□□□□□.
B. 'CREATE TABLE CLONE SALES DATA BEFORE(STATEMENT y)' □□□□ □□□□ □□□□ □□□□ □ □□□□ □□ □□□□ □□□ □□□□ □□□□ □□□□□.
C. Snowflake □□□□ □□□□ □□□□ □□□□□□ □□□□□.
D. SELECT □□□ 'TIME TRAVEL' □□□ □□□□ □□□□ □□□□ □□□□ □□ □□□□□.
E. 'CREATE TABLE CLONE SALES DATAAT (OFFSET -86400 5)' □□□□ □□□□ □□□□ □□□□ □ □□□□ □□ □□□□ □□ □□□□ □□□□ □□□□□.
Answer: (SHOW ANSWER)
'UNDROP TABLE' □□□ □□□ □□ □□ □□□ □□□ □□□□ □□□□ □□ □□□□□ □□□□ □□□□□. □□□□ 5□ □□ □□□□□ □□ □□ □□(Enterprise Edition□□□ □□ 90□)□ □□□□□ □□□□□ 'UNDROP TABLE' □□□□ □□□ □ □□□□. □□ B□ E□ □□□ □ □□ □□□□ □□□□ □□□ □□□□□, □ □□□□ □□□□ □□□ 'UNDROP TABLE'□□ □□□□ □□□ □□□□. □□□ □□□ □□□□ Snowflake □□□□ □□□□ □(C)□ □□□□□□. □□ D□ □□□ □□ □ □□□ □□ □□□□ □□□□□ 'UNDROP' □□□ □□ □□ □□□□ □□□ □ □ □□□□□.

NEW QUESTION: 44
□□□ □□□□ □□□□ □□□□□ □□ □□□ □ □□□□□ □□ □□□□ □ □□ □□□ □□□?
A. □□□□ □□□□□ □□□□ □□□□ □□□ □□□□□.
B. □□□ □□□□ □□□□ □□□□ □□□□ □□□ □□□□.
C. □□□ □□□□□ □□□□ □□□□ □□□□ □□□ □□□□.
D. □□□□ □□ □□ □□□ □□□□ □□□□ □□□ □□□□□□.
Answer: A (LEAVE A REPLY)
□□□□ □□ □□ □□□ □□□□ □□□□ □□□□□ □□□□ □□□□□ □□□ □ □□□□.

NEW QUESTION: 45
□□□□□□ □□ □□□ □□□□□□ □□□□□ □□□□ □□□. □□ □□□ □□ □□□(□□, □□□□□, □□, □□□ □□), □□ □□□(□□, □□, □□ □□) □ □□ □□□□ □□□□ □□□ □. □□□□ □□ □□□ □□□□ □□□ □□□□□ □□□□□ □□□. □□ □□□ □ Snowflake □□ □ □□ □□ □□□□ □□□□ □□ □ □□□ □□ □□□□□□? (□ □□□ □□□□□)
A. Snowflake□ □□□ □□□ □□ □□□ □□□□ □□ □□ □□□□ □3□ □□ □□□ □□□□□.
B. Power BI□ Snowflake□ □□□□ □□ □□□ □□□□ □□□ □□, □□□ □□□ □□, □□ □□□ □□ □□ □□□ □□□□ □□□□ □□□ □□□□□ □□□□□. Snowflake□ □□ □□ □□ □□(Snowpark □□ Python □□ □□□□ □□)□ □□□□ □□□□□ □□□ □□ □□□□□. Power BI□□ □□□ □□ □□□ □□□ □□□□ □□ □□□□ □□□□□.
C. Snowflake□ □□ □□□ □□□□ □□ □□ □□ □ □□□ □□□□ □□ □□□□ □□□□□.
D. Tableau □□□□□□ □□□ □□□□ □□□□ □□□ □□, □□ □□ □ □□ □□ □□ □□□□□ □□□□□. □□ □□, □□ □ □□□ □□ □□□ □□□ □□□□□. Snowflake □ □□□ □□ □ □□□□ □□ □□□□ □□□□ □□□ □ □□□ □□□□□.

E. □□ □□□□ □□□ □□□□ □□□□ Spark□ □□□□ □□ □□□ □□□□□.
Answer: B,D (LEAVE A REPLY)
□□ B□ D□ □□ □□□□□□. □□ B□ Power BI□ □□□ □□□ □□□□□ □□ □□□, □□□, □□□ □□□□ □□□□ □□□ □□□□□. Snowflake□ □□ □□ □□□□□ □□ □□□ □□ □□□□□. □□ D□ □□□□ □□□□ □□ □□ □□ □□□□□ □□□□, □□ □□□ □□ □□□ □□□ □□□□□. □ □□□ □□□ □□□□ □□□ □□□ □□□□ □□□ □□ □□□□ □□ □□□ □□□ □ □□□□. □□ A□ □□□□ □□□□ □□□□ □□□ □□□□. □□ C□ □□□□ □□ □□□ □□□□□. □□ E□ Snowflake □□□ □□□□ □□□□ □□□□, □□ Snowflake□ □□ □□□ □□□□ □□□ □□□□ □□□□ □□□□ □□□ □ □□□□.

NEW QUESTION: 46
□□□ □□□□ 'SALES' □□□(10□ □)□ 'CUSTOMERS' □□□(1□□ □)□ 'CUSTOMER ID'□ □□□□ □□□ □ □□ □□□ □□□□ □□□ □□ □□□□. 'SALES' □□□□ □□□□ □□□□□ □□. □□□□ □□□ □□□ □□□□. SELECT s. , c. FROM SALES s JOIN CUSTOMERS c ON s.CUSTOMER_ID = c.CUSTOMER_ID WHERE s.SALE_DATE DATEADD(day, -30, CURRENT DATE()); □□□ □□□ □ □□ □□□ □□□ □□ □□□ □□ □□□□□ □□□ □, □□ □ □□ □□□ □□ □□□ □□ □□□□□ □□□□ □ □□□□?
A. □ □□□□ □□□□ □□□□ □□ □□□□ 'SALE_DATE'□ □□□□ □□□□□□. □□□□ □□ □□ □□ □□□□.
B. 'SALES.CUSTOMER ID'□ 'CUSTOMERS.CUSTOMER ID'□ □□ □□□□ □□□□□.
C. 'SALES.CUSTOMER' □ 'CUSTOMERS.CUSTOMER ID'□ □□ □□ □□□□ □□□□□.
D. 'SALES' □□□□ 'CUSTOMER_ID'□ □□□□ □□ □□□□ □□□ □□ □ □□□□ □□ □□□□□□□□.
E. □□ 30□□□ □□ □□□□ □□□□ 'SALES LAST DAYS' □□□□ □□□□, □□ □□□□ 'CUSTOMERS' □□□□ □□□□□. 'SALES LAST 30 DAYS' □□□□ □□ □□ □□□□.
Answer: E (LEAVE A REPLY)
'□□ 30□□ □□' □□□□ □□□□ □□□□ □□ □□ □□□□ □□□□□. □□ □□ □□ SALES □□□□ □□□ □□ □□ □ □□ □□□□□. □□□□ □(A)□ □□□□□ □□□□□□, SALES □□□□ □□ □□□□ □□□ □□ □□□□ □ □□ □□□ □□□ □□□□□ □□□ □ □□□□. □□ □□□(B)□ Snowflake□□ □□□□ □□□□. □□ □□□(C)□ □□□□□□ □□ □ □□□ □ □□□□ □□ □ □□□□. □□□□□(D)□ □□□ □□□□ □ □□□, □□ □□□ □□ □ □□□□□□ □□□ □□□ □□□ □□□ □□ □ □□□, □□ □□ □□□ □□□□ □□□□.

NEW QUESTION: 47
SQL□ □□□ □□□ □□□□ □□ □□□□□ □□□ □□□ □□□□□?
A. □□□ □□□ □□□□ □□□□□.
B. □□□□ □□ □□□ □□□□ □□□□ □□□□□□.
C. □□ □□□□□ □□□ □□ □ □□□□ □□□ □□□ □□□□ □□ □□□□ □□□□□□.
D. □□ □□□□□ □□□ □□□□ □□□□ □□ □□□ □□□.
Answer: (SHOW ANSWER)
□□ □□□□□ □□□ □□ □ □□□□ □□□ □□□ □□□□ □□ □□□ □□□ □□□□ □□□□ □□□□□□.

NEW QUESTION: 48
□□□ □□□□ □□ □□□ □□, □□□ □□□□ □□□ □□ □□□□ □□□ □□□□. SALES_DATE□ □□ □□ □□□ □, □□□ 7□ □□ □□□ □□□□ □□□ □□□ □□□□□?
A. SUM(SALES) OVER (ORDER BY SALES_DATE ROWS BETWEEN 6 PRECEDING AND
□□ □)
B. SUM(SALES) OVER (ORDER BY SALES_DATE ROWS BETWEEN 7 PRECEDING AND

□□ □)
C. AVG(SALES) OVER (ORDER BY SALES_DATE ROWS BETWEEN 6 PRECEDING AND
□□ □)
D. AVG(SALES) OVER (ORDER BY SALES_DATE ROWS BETWEEN 7 PRECEDING AND
□□ □)
Answer: C (LEAVE A REPLY)
□□□□(□□ □□ □□) □□□ □□□□ □□□ □□□□ □□□□ □□□ □□□□ □ □□□□ □□ □□□ □□ □□□□□. Snowflake□□□ □□□ □□□ ROWS □□□ □□ □□□□ □□ □□□ □□.
□□□ □□□ □□□□ □□ □□ 7□ □□ □□□ □□□□□ "□□□" □□ "□□□"□ □□□ 7□□ □□ □□□□ □□□. SQL □□□ □□□□ □□ □□ □ 7□□ □ □ □□□ □□□□□.
□□□ □□□ 7□ □□□ □□ 6□ □□ □□ □□□□ □□□($6 + 1 = 7$).
□□□ □□□□:
* □□ A□ B□ SUM() □□□ □□□□□. □□□ □□□ □□□□□, □□□□□ □□ □□□ □□□□ □□□□□ □□□□ □□□□.
* □□ D□ □□□□□□□. □□□□ "7 PRECEDING AND CURRENT ROW"□ □□□ 8□ □□(□□ □□□ □ □□ 7□□ □□ □□)□ □□□□ □□□□□.
* □□ C□ 100% □□□□□. AVG() □□ □□□ □□□□ □□□□ "□□ □ □ □□ □ 6□"□ □□□□ □□□□ □□ □□□ □□□ □□□ □□□□□ □□□□ □□□ □□□□□.

NEW QUESTION: 49
Snowsight□ □□□□ □□□□ □□ □□□□ □□□□□ □□□□ □□□□. □□□□ 'WEB EVENTS'□□ □□□□ □□□, 'EVENT_TIME'(TIMESTAMP_NTZ), 'EVENT_TYPE'(VARCHAR, □: 'page_view', 'button_click'), 'USER_ID'(VARCHAR), 'PAGE URL'(VARCHAR) □□ □□□□. □□ 7□ □□ □ □□□□ □□□□ 'page_view' □□□ □ □□ □□□ □□□□ □□□ □□□□ □□□. □□ □□ □□□□□ □□□□ □□□ □□□□□ □□□ □ □□□□. □ □□□□ □□, page_view □□□□ □□□ □□ □□□ □□□□ □□□ □□□□□. □□ SQL □□ □ Snowsight □□□ □□□□ □ □□ □ □□ □□ □□□ □ □□□ □□□□ □□□□ □□□ □□□□□?

```
AG(EVENT_TIME) OVER (PARTITION BY USER_ID ORDER BY EVENT_TIME) AS PREVIOUS_EVENT_TIME FROM WEB_EVENTS WHERE EVENT_TYPE
 'page_view' AND EVENT_TIME >= DATEADD(day, -7, CURRENT_TIMESTAMP()) ) WHERE PREVIOUS_EVENT_TIME IS NOT NULL GROUP BY USER_ID;
```

) ```sql SELECT USER_ID, AVG(EVENT_TIME - LAG(EVENT_TIME) OVER (PARTITION BY USER_ID ORDER BY EVENT_TIME)) FROM WEB_EVENTS
VHERE EVENT_TYPE = 'page_view' AND EVENT_TIME >= DATEADD(day, -7, CURRENT_TIMESTAMP()) GROUP BY USER_ID;```

) ```sql SELECT USER_ID, AVG(DATEDIFF(second, EVENT_TIME, LAG(EVENT_TIME) OVER (PARTITION BY USER_ID ORDER BY EVENT_TIME))) FROM
VEB_EVENTS WHERE EVENT_TYPE = 'page_view' AND EVENT_TIME >= DATEADD(day, -7, CURRENT_TIMESTAMP()) GROUP BY USER_ID;```

) ```sql SELECT USER_ID, AVG(TIMESTAMPDIFF(second, EVENT_TIME, FIRST_VALUE(EVENT_TIME) OVER (PARTITION BY USER_ID ORDER BY
VENT_TIME))) FROM WEB_EVENTS WHERE EVENT_TYPE = 'page_view' AND EVENT_TIME >= DATEADD(day, -7, CURRENT_TIMESTAMP()) GROUP BY
SER_ID;```

) ```sql SELECT USER_ID, AVG(TIMESTAMPDIFF(second, PREVIOUS_EVENT_TIME, EVENT_TIME)) FROM WEB_EVENTS WHERE EVENT_TYPE =
age_view' AND EVENT_TIME >= DATEADD(day, -7, CURRENT_TIMESTAMP()) QUALIFY PREVIOUS_EVENT_TIME IS NOT NULL GROUP BY USER_ID;```

A. □□ A
B. □□ B
C. □□ C
D. □□ D
E. □□ E
Answer: (SHOW ANSWER)

□ □□□ □□□ □□□ □□□□ □ □□□□ □□ □□□ □□□ □□□ □□, 'TIMESTAMP_DIFF' □□□ □□□□ □□□□ □□□ □□ □□ □□□ □ □□□ □□□□□. □□ □□ □□ □□□□ □ □□□□ □□ □□□ □□□ □□□□□. 'WHERE PREVIOUS_EVENT_TIME IS NOT NULL' □□ □ □□□□ □ □□ □□□(□□ □□□ □□□ null□ □□)□ □□□□ □ □□□□□. □□ B□ □□□ □□□ □□ □□□ □□□□□, □□ Snowflake□□ □ □□ □□□ □□□ □□□ □□□ □□□□. □□ C□ 'DATEDIFF' □□□ □□□□□, □ □□□ □□□ □□□ □□ □□□□ □□□ □□□□□□ □. □□ D□ 'FIRST_VALUE'□ □□ □□□□□□. □□ E□ LAG □□□ □□□□ □□□□ □□ □□□ □□□□□ □□□□□□□.

NEW QUESTION: 50
Snowflake□□ □□ □□(□: □□□□, □□ □□, □□□)□ □□□ □, □□□ □□□ □□□ □□□ □□□ □□□□□?
A. □□□□ □□□ □□□ □□□ □□□ □□□□□.
B. □□ □□□ □□ □□□ □□□□ □□□ □□□□.
C. □□□ □□□ □□ □ □□ □□□ □□□ □□□.
D. □□□□ □□□□ □□□□ □□□□ □□□□□.
Answer: (SHOW ANSWER)
□□ □□□□ □□ □□□□ □□□ □□□□ □□□□ □□□ □□□ □ □□□ □□, □□ □□ □□□ □□□□ □□ □□ □□□□ □□□ □□□ □ □□□□.

NEW QUESTION: 51
□□□ □□ □□□ □□(□□□, □□□□, □□ □□□)□ □□ □□ □□□□ □□□□□□ □□□. □□ □□□□ Snowflake□ 'CUSTOMER ENGAGEMENT' □□□□ □□□□ □□□, 'CUSTOMER ID', 'CHANNEL', 'EVENT TYPE', 'EVENT TIMESTAMP', 'EVENT VALUE'(□□ □□) □□ □□□□. □□□ □□ 7□□□ □□□ □ □□ □□□ □□□□□ □□□□, □□ □□ □□□ □□□ □□□ □ □ □ □□□□□ □□□□□. □□□ □□□□ □□□ □□ □□□ □□□□□ □□□□ □, □ □□□□□ □□ □□□□ □□ □ □ □□□ □□□□ □□ □□□ □□□□□?
A. □□□□ □□ □□□□ 'CUSTOMER_ENGAGEMENT' □□□□ □□ 7□□ □□□□ □□ □□□□ □□ □□□□□. 'GROUP BY CHANNEL'□ 'SUM(EVENT_BALUE)'□ □□□□□.
B. Snowflake □□□ □□□□ □□ □□□□ □□ 7□□□ □□□ □ □□ □□ □□□ □□ □□□□ □□□□ □ 'MV CUSTOMER ENGAGEMENT'□ □□□□□. □□□□ □□□ □ □□□□ □□□ □ □ □□□□ □□□□□.
C. '□□□ □□□'□ □□□□ '□□ □□' □□□□ □□□□ □□ □□□ □□□□, □□ 7□ □□□□ □□□□□ □□□ □□□. □□□□ □□ □□□ □□ 7□ □□□□ □□□□□□ □□□.
D. '□□ □□' □□□ □□□ □□□□ □□□ □□ □□□□ □□□□ □□□ □□ □ □□□□ □□□□□.
E. □□ □□□□□ □□□□ □□□ □□□ □□□□ □□ □□□□ □□ □□□□□□□. □□□□□ □ □□ □□□□ □□□□□.
Answer: B (LEAVE A REPLY)
□□□□ □(B)□ □□□ □□□□ □□ □□□□ □□□□ □□□ □□ □□□ □□□ □□□□□. Snowflake □□□ □□□□ □□ □□ □□□□ □□□□ □□ □□ □□□ □□□□□. □□ A□ □□□ □□ □□□ □□□ □□□ □□□□ □□□. □□ C□ □□ □ □□□□ □□□□ □□□ B□□ □□□□ □□□□□. □□ D□ □□□ □□□ □□□ □□ □□□□ □□□□□. □□ E□ □□ □□□□□ □□ □□□□ □□ □□□□ □□□□ □□□□ □□ □□ □□□ □□□ □ □□□□.

NEW QUESTION: 52
Snowflake□ 'USER ACTIVITY' □□□□□ VARCHAR □□□ 'ACTIVITY TIMESTAMP' □□ □□□□. □ □□□ ISO 8601(□: '2023-10-27'), Unix □□□ □(□: '1698400800'), □□□(□: '1698400800000') □ □□□ □□□ □□□□□□ □□□□□. □□□ □□ □□□□□□ Snowflake□□ □□□ TIMESTAMP NTZ □□□ □□□□ □□□□ □□□. □□□ □□□ □□□□□ □□□ □ □□ TIMESTAMP NTZ□ □□□□ □ □□ □□□□□ □□□ □□□ □□□□□?
A. 'TO_TIMESTAMP'□ □□ □□ □□□□ □□□□ □□□ 'CASE' □□ □□□□ □ □□□ □□□□□ □□□□□.
B. 'TO_TIMESTAMP'□ □□□□ □ □□□□□ □□□□ □□□ □□ □□□□ □□□ □□, □□□ □□ UNION□□ □□□□.
C. □□□□□ □□□ □□□□□ □□□□ □□□ □□ □□(□: CTO TIMESTAMP □)□ □□□□ □□□ □□ □□(UDF)□ □□□□□.
D. □□ □□□ □□□□ 1000□□ □□□ □□□ □□ □□□□□. □□ □□ □□ ISO 8601 □□□ □□□□ 'TO TIMESTAMP' □□□ □ □ □□□□□.
E. Snowflake□ □□ □□□ □□ □□ □□□ □□□□□ ALTER TABL □□□□ □□□□ □□ □□ □□□ □□□ TIMESTAMP_NTZ□ □□□□□ □□ □□□.
Answer: (SHOW ANSWER)

□□ C□ □□ □□□□□ □□□ □□□□ □□□□□. 'CASE' □(□□ A)□ □□□□□, □□□ □□□ □□□ □□ □□□□□ □□ □□□ □□□□ □ □□□□. □□ □□□□ □□□□ □□(□□ B)□ □□□□□□□. □□□□ ISO 8601 □□□□ □□□□ □□(□□ D)□ □□ ISO 8601 □□□□□ □□□ □□□□ □□ □□□ □□□ □□□ □ □□□□. Snowflake□ □□ □□ □□ □□(□□ E)□ □ □ □□□ □□□□ □□□□. □□□ □□ □□(UDF, □□ C)□ □□ □□ □ □□ □□□ □□□□□ □□ □□□□□ □□ □□□□ □□□□□□. UDF □□□ □□□ Snowflake □□□ □□□□ □□ □ □□ □□ □□□ □ □□□, □□ □□ □□□ TIMESTAMP NTZ □□□ □□□□ □□□□ □□□ □ □□□□.

NEW QUESTION: 53
'customer_orderS'□□ □□□□ □□□ □□□ □□□□□. □ □□□□□ 'order_id'(INT), 'customer_id'(INT), 'order_date'(DATE), 'order_total'(NUMBER) □□□ □□□□. □□□□ 'order_date'□ □□□ □ □□□□□ □□□□. □□□, □□ □ □□ □□□ □□□□ □□□□ □□ □□□□ □□□. □□ □ □□ □□□□ □ □□□ □□□ □□□ □□□□ □□□ □□□□□ □□□□□ □□□□ □□□□ □□ □□□ □□□□ □□□?

C CREATE MATERIALIZED VIEW monthly_customer_orders AS SELECT customer_id, DATE_TRUNC('month', order_date) AS order_month, SUM(order_total) AS total_order_value FROM customer_orders GROUP BY customer_id, DATE_TRUNC('month', order_date);

C CREATE MATERIALIZED VIEW monthly_customer_orders AS SELECT customer_id, order_date, SUM(order_total) AS total_order_value FROM customer_orders GROUP BY customer_id, order_date;

C CREATE MATERIALIZED VIEW monthly_customer_orders AS SELECT customer_id, MONTH(order_date) AS order_month, SUM(order_total) AS total_order_value FROM customer_orders GROUP BY customer_id, MONTH(order_date);

C CREATE MATERIALIZED VIEW monthly_customer_orders AS SELECT customer_id, CONVERT_TIMEZONE('UTC', 'America/Los_Angeles', order_date) AS order_month, SUM(order_total) AS total_order_value FROM customer_orders GROUP BY customer_id, CONVERT_TIMEZONE('UTC', 'America/Los_Angeles', order_date);

C CREATE MATERIALIZED VIEW monthly_customer_orders AS SELECT customer_id, CAST(order_date AS VARCHAR(7)) AS order_month, SUM(order_total AS total_order_value FROM customer_orders GROUP BY customer_id, CAST(order_date AS VARCHAR(7));

C CREATE MATERIALIZED VIEW monthly_customer_orders AS SELECT customer_id, CAST(order_date AS VARCHAR(7)) AS order_month, SUM(order_total AS total_order_value FROM customer_orders GROUP BY customer_id, CAST(order_date AS VARCHAR(7));

A. □□ A
B. □□ B
C. □□ C
D. □□ D
E. □□ E
Answer: A (LEAVE A REPLY)
□□ A□ □□ □□ □□□□□. □ □□□ 'DATE TRUNC('month', order_date)'□ □□□□ □□ □□□ □□□□□. □□ □□□□ 'order_date'□ □□□□□ □□□□ □□□□ □ □□ □□ □ □□ □□ □□□ □□□□ □□□ □□□□□ □□□□□ □□□ □ □□□□. □□ B□ □□ 'order_date'□ □□□□□ □□□ □□ □□ □□□ □□□□□. □□ C□ □□□□ □□□□ □□□ □□□□□ □□ □ □ □□□ □□□□ □□□□. □□ D□ 'CONVERT_TIMEZONE'□ □□□□□, □□ □□□ □□ □□□ □□□ □□□□□ □□□□□ □□□□ □□□□. □□ E□ 'CAST(order_date AS VARCHAR(7))'□ □□□□□, □□ □□ □□□ □□□ □□□□□.

NEW QUESTION: 54
□□□□ □□□□ □□□□□□ □□□□□ □□□□ □□□□. □□□ □□ □□ □□□ □□□□. 1. □□ □ □□□ □□ □□□□□. 2. □□□□ □□ □□(□□□□, □□□, □□□)□□ □□□□ □□ □□ □ □□□ □□□. 3. □□ □□□ □□ □ □□□ □□ □□□□ □□□□□. 4. □□□□□ 15□□□ □□ □□□□ □□ □□□□ □□□. 5. □□□□□ □□□ □□□□ □□□□ □□□ □□□□ □ □□. □□□ □□□ □□□□□.

```
CREATE OR REPLACE TABLE website_traffic (
    event_time TIMESTAMP_NTZ,
    user_id VARCHAR,
    device_type VARCHAR
);
```

□□ □ Snowflake□□ □□ □□□□□ □□□□ □□□ □□□□ □□□□□? □□□□ □□ □□□ □□□□□□.

A. □□ □ □□□ □□ □□ □□□ □□□□ □□ □□□□ □□□□ □□ □□□□□. Snowflake □□□ □□□□ 15□□□ □□□□ □□ □□ □□□□. □□□□□ □ □□□□ □□ □□□□□.

B. Snowflake □□□□ □□□□ 'website_traffic' □□□□ □□ □□□ □□□□□. 15□□□ □□□□ □□□□ □□ □ □□□ □□ □□ □□□ □□□□ □□ □□□□ □□□□□□ □□□ □□□□□. □□□□□ □ □□ □□□□ □□□□□.

C. □□ □ □□□ □□ □□ □□□ □□□ □□ □□□□ □□ Snowflake □□ □□□□□. □□□□□ □□ □□□□ □□□□□ □□ □□ □□ □□□□□. □□□□ □□□□ □□□□ □□□□.

D. □□□□ □□□ □□□ □□□ □□ □□□ □□□□ '□□□□ □□□' □□□□□ □□ □□ □ □□□ □□ □□ □□□ □□□ □□ □□□□□ □□□ □ □□□□.

E. □□ □ □□□ □□ □□ □□□ □□□ □□ □□□□ □□ □□□□□ □□□□□. □ □□ □□□□□ 15□□□ □□□□□ □□□□, □□□□ □□□□□□□. □□□□□ □ □□□□ □□□□□.

Answer: (SHOW ANSWER)

□□□□ □(□□ A)□ □□□□ □□□ □□□(□□ B)□ □□□ □□□ □□□ □□□ □□□□□ □□□□ □ □□ □□□□ □□□□□. □□□□ □□ □□□ □□ □□□□□ □□ □□□ □□ □□□ □□. □□□□ □□□□ □□ □□□□ 15□□□ □□□ □□□□ □□□□ □□ □□□ □□ □□□ □□□ □ □□□□. □□ □□ □□ □□□ □□□ □□□□ □□□□ □□ □ □□□□. □□ □(□□ C)□ □□□ □□□ □□□ □□□□□ □□□ □□□ □□□□ □□□ □□□□□. □□□□ □□□ □□ □□(□□ D)□ □□□□ □□ □□□□□ Snowflake□ □□□ □□□ □□□□ □□□ □□□□ □□□□□. □□ □□□□(□□ E)□ □□□ □ □□□ □ □□□□□□□ □□□□ □□□ □□□□ □□□□ □□□ □□□□.

NEW QUESTION: 55

□□□□□ □□□□ □□ □□ □□□ □□□□□□ Snowsight □□□□□ □□□□ □□□. □□ □□□□ 'SALES_DATR'□□□ □□□□ □□□□ □□□, 'SALE_DATE'(DATE), 'PRODUCT_ID'(NFT), 'SALES AMOUNT'(FLOAT), 'CUSTOMER REGION'(VARCHAR) □□ □□□□ □□□□. □□ 30□□□ □□ □ □□□□ □□□□ □□□ □□□ □ □□□ □□ □□ 5□ □□□ □□□□ □□ □□□□ □□□. Snowsight□□ □ □□□ □□□□ □□ □□□□ □□□ □□ □ □□□□□? (□ □□□ □□□□□)

A. Snowsight □□□□□ □ □□ □□ □□□ □□□□□. □□□ 'SELECT SALE_DATE, SUM(SALES_AMOUNT) FROM SALES_DATA WHERE SALE_DATE DATEADD(day, -30, GROUP BY SALE_DATE ORDER BY SALE_DATE;' □□□ □□□□□ □□□ □□□ □□□□, □□ □□□ 'SELECT CUSTOMER REGION, SUM(SALES AMOUNT) FROM SALES DATA GROUP BY CUSTOMER REGION ORDER BY DESC LIMIT 5;' □□□ □□□□□ □□□□ □□□□□.

B. Snowsight□□ □□ □□□ □□□□ □□ □□□ □□□□□. x□□□ '□□□'□, y□□□ '□□□ □□'□ □□□□ □□□□ □□□□□. □□ □□, □ □□ □□□ □□□□ □□ □□ □□□ □□□ □□□□ □□□□□.

C. Snowsight□ '□□□' □□□ □□□□ □□□□ □□□ □□□□ □□ □□ □□□ □□ □□□ □□, □□□ □□□□□□□ □□□□ □□□ □□□ □□□□□. □□□□□ SQL□ □□ □□□□ □□ □□.

D. □ □ □□ □□ □□ □□□ □□□□□. □ □□ □□□ □□□□ □□□ □□□ □□□ □□□□, □ □□ □□□ □□□ □□ □□□□ □□□□□. □ □ □□ □□□ SQL □□□ □□□□□ □□□ □ □□ □□□□□□.

E. Snowsight□□ □ □□ □□ □□□□□ □□□□□. □ □□ □□□□□□ □□□ □□□ □□□□. □ □□ □□□□□□ □□□ □□ □□ □□□□. □□ □□ □ □□□□□ □□□ □□ □□□□□ □□□□□.

Answer: A,C (LEAVE A REPLY)

□□ A□ □□ □□□ □□□□ Snowsight□□ □□□ □□□□ □□□□□ □□□□□ □□□□□ □□□□□. □□ C□ □□□□□. Snowsight□ □□□ □□□ □□□□ □□ □□□ □□□□□ □□ □□□□ □□ □□□□ □□ □□□□ □□□ □□□ □ □ □□ □□□ □□□□□□ □□□□□. □□ B, D, E□ □□ Snowsight □□□□ □□□□ □□ □□□□ □□□ □□□□ □□□□ □□□□□.

NEW QUESTION: 56

□□ □ Snowflake□□ □□□ □□ □□(UDF)□ □□□□ □□ □□□ □□□□□ □□ □□□□, □□ □□□ SQL □□□□ □□□□ □ □□ □□□ □□□ □□□□□□. (□□□□ □□ □□□ □□□ □□□.)

A. UDF□ □□□ □□□ □□□□□ □□ □□□□ □□□□ □ □□□, □□□ □□ □□□□ □□□□ □□□ □□□□.
B. Snowflake UDF, □□ Java □□ Python□□ □□□ UDF□ □□ □□□ □□□ □□□□ □ □□ □□□□ □□ □□ □□□□□ □□□ □ □□□□.
C. SQL UDF□ □□□□□ Snowflake □□ □□□ □□□□□ □□□□□ Java □□ Python UDF□□ □□□ □ □□□□.
D. UDF□ □□ □□ □□□ □□□ □□□ □□□ □□ □ □□□□ □ □ □□□□, □□ □□□ □□□ □□ □□□□ □□□□ □□□□□ □□□□ □□ □□□ □□□□□ □□□□ □ □□□□.
E. □□ □□□ UDF□ □□□□, □ □ Snowflake □□□ □□□□.
Answer: A,B,C,D (LEAVE A REPLY)
□□□ □□ □□(UDF)□ □□ □□□□ □□□□ □ □□□ □□ □□ □□□□ □□□□ □□ □□□□. Java □ Python UDF□ Snowflake □□ □□□□ □□□□□ □□□□□ □□□□□. SQL UDF□ □□□□□ □ □□□□. UDF□ □□ □□□□ □□□□□. □□ □□□ UDF□ □□ □□□□. UDF□ Snowflake □□□ □□□□□.

NEW QUESTION: 57
Snowflake □□ □□ □□□□ □□□ □□□ □□□□ '□□' □□□ □□□□□□. □□ □□ □□□□ 'calculate_score(columnl, column2)'□□ □□□ □□ □□(UDF)□ □□□□□. □ UDF□ Python□ □ □□□□□□□. □□ □□□ □□□ □□ □ □□□□ □□□□ □ □□ □□ □□ □□ □□□□□□. □ □□□□□ □□□□□ □□ □□ □□□ □□□□ □□□ □□ □□□?
A. Snowflake□ □□ □□ □□□ □□□□ □□ UDF□ Python □□ SQL□ □□ □□□□□.
B. UDF □□□ □ □□ □□□□ □□□□ □□ □□□□□ □□□ □□□□.
C. 'calculate_score' □□□ □□ □□(UDF)□ □□□□ □□□ □□ □□□□ □□□ □□□□ □□□□ □□ □□□□□. □□ □□ □□□ □□ □□□□ □□ □□□□ □□□□□ □□□.
D. □□ □□□ □□ UDF□ □□ □□□□□ □□□□ □□ □□□ □□□□□□.
E. UDF □□ □□□ □□□□ □□□ □□□ □□□ □□□□ □□□ □□□ □□ □□□□□□.
Answer: (SHOW ANSWER)
□□ A□ C□ □□□ □□ □□□□□□. A□ □□□ Snowflake □□□□ □□ □□□ □□□□ □□□□ □□□□. C□ □□□ □□ □□□□ UDF □□□ □□□□ □□□ □□□. □□□□□ □□□ □□ □ □(B)□ □□□ □ □ □□□, UDF□ □□□□ □□□□□ □□□□ □□ □□□□. UDF □□□ □□□□ □(E)□ □□□ □□□□ □□ □□□ □□ □□ □□□ □□□□ □ □□□, □ □□□□ SQL □□ □□□ □□□□ □□ □ □□ □□□□□□. □□ □□□□ □□□ □□□ □□□ □ □□ □□ □□□ □□□□□.

NEW QUESTION: 58
'raw data'□□ □□□ Snowflake □□□□ CSV □□□ □□□□ □□□□. □ □□□□ □□ □□ □□□□ □□□□ □□□, □□ □□ 'order date' □□ □□□□ □□□□. □□□ 'order date' □□ '1900-01-01'□ □□□□ □□□ □□□□ □□□□ 'cleaned data'□□ □ □□□□ □□□□ □□□. Snowflake □□□ □□□□ □ □□□ □□□□ □□ □□□□□ □□□ □□□ □□ □ □□□□□?
A.
```
CREATE TABLE cleaned_data AS SELECT FROM raw_data WHERE order_date IS NOT NULL;
```
B.
```
CREATE TABLE cleaned_data AS SELECT ,IFF(order_date IS NULL,'1900-01-01', order_date) as order_date_new FROM raw_data;
```
C.
```
CREATE TABLE cleaned_data AS SELECT ,NVL(order_date,'1900-01-01') as order_date_new FROM raw_data;
```
D.
```
CREATE TABLE cleaned_data AS SELECT ,COALESCE(order_date,'1900-01-01') as order_date_new FROM raw_data;
```
E.
```
CREATE TABLE cleaned_data AS SELECT other_columns, COALESCE(order_date, TO_DATE('1900-01-01')) AS order_date FROM raw_data;
```
Answer: E (LEAVE A REPLY)
□□ E□ □□ □□□□□ □□□□□. 'COALESCE'□ NULL □□ □□□□□ □□□□, □□ □□ □□□ □□□ □□(DATE)□□ □□□□□. □□ □□ □□ □□ □□□□□ □□□□□. □□ A□ order_date□ NULL□ □□ □□□□□□. □□ B, C, D□ □ □□ □□□□□. □□ □□ □□ □□□□□ □□□□ □□□□, □□ □□ □□□□ □□ □ □□□□□.

NEW QUESTION: 59
□□□□□ □□ □□□ □□□ □□□□ □□ □□□ □□□□ □□□ □ □□□□□ □□ □□□□ □ □□□ □□□□□?
A. □□□□ □□□ □□□□ □□□ □□□□ □□□□□□.
B. □□□ □ □□□□□ □□□□ □□□□□.
C. □□□□ □□□ □□□□ □□□□ □□□□ □□□□□.
D. □□ □□□ □□□ □□□□□ □□□ □□□□□ □□□□□□.

Answer: C (LEAVE A REPLY)
□□□□ □□□ □□□□ □□□□ □□□ □ □□□□ □□ □□□ □□□ □□□□ □□□□□.

NEW QUESTION: 60
□□□ □□ □□□□ □□□ □, □□□ □□ □□□ □□□ □ □□ □□□□ □□□□ □□□? (□□□□ □□ □□□ □□□□□)
A. □□□ □□
B. □□□□ □□□□
C. □□□ □□ □□□ □□
D. □□□ □□ □□□
Answer: A,B (LEAVE A REPLY)
□□□ □□ □□□ □□□ □□ □□□□ □□ □□□ □□□ □□□ □□□ □□ □□□□ □□□.

NEW QUESTION: 61
Snowflake□□ 'TRANSACTIONS'□□ □□□ □□□ □□ □□ □□□□ □□□□ □□□ □□ □□□□□ □□□□ □□□□. □□ □□□□□ □□□□ □□□ □□□□ □□ 'transaction_id', 'customer_id', 'amount'□ □□ □□ □ □□ □□□□ □ □□ □□ □□ □□ □□□□ □□□. □□□ □□□ □ □ □ □□□□ NULL □□ □□□ □ □□□, □□ □□ □□□□ □□□ NULL □□ □□□ □ □□□□ □□□. Snowflake□ □□□ □□□ □□□□ □□ □□ □□ □ □□ □ □□ NULL □□ □□□ □□□□ □□ □□ □□ □□ □□□□□ □□□ □□ □□ □□□□ □□□ □□□□□? (□ □□□ □□□□□□)
A. Snowflake□ □□ □□□□ NULL □□ □□□□ □□□□□ customer_id, amount, transaction_date □□□ □□ □□□□□.
B. 'SHA2(CONCAT(NVL(transaction_id, 'I), NVL(customer_id, ''), NVL(amount, ''), NVL(transaction_date, '')))' □□□ □□□□, □□□□ □□ NVL □□□ □□□□ NULL □□ □ □□□□ □□□□.
C. 'II customer_id II amount II transaction_date' □□□ □□□□□. Snowflake□ □□□ □□ □□ NULL □□ □□□□□ □□□□□ □□□□□.
D. WS('I', transaction_id, customer_id, amount,) □□□ □□□□□. □ □□□ □□□□ □□□□ □□□□ □□□□, NULL □□ □□ □□□□ □□□□ □□□□□ □□□□ □□□ □ □□□□.
E. AS VARCHAR), AS VARCHAR), AS VARCHAR), NVL(CAST(transaction_date AS VARCHAR), □□□ □□□□ □ □□ □□□□□ □□□□ □□□□ NVL□ □□□□ NULL□ □ □□□□ □□□□.
Answer: B,E (LEAVE A REPLY)
□□ B□ E□ □□ □□□ □ □□□□. □□ B□ □ □□ □□□□ □□□□ SHA2 □□□ □□□□, NULL □□ □ □□□□ □□□□□ . □□ □ □□ NULL□ □□ SHA2□ NULL□ □□□□ □□□ □□ □□ □□□□ □□□□□. □□ E □□ □□ □ □□ □□□ □ SHA2□ □□□□ □□□□□□, □□ □□ varchar□ □□□□□□. □□ □□□ □□ □ □□□ □□ □□□□ □□ □□□ □□□ □□ □ □□ □□□ □□□□□. □ □□ □□□ Snowflake□ HASH □□□ □□□ □ □□□□ NULL□□ □□□□ NULL□ □□□□□ □□□□ □□□□. □□ C□ '1/2' □□□□ □□□□ □□ □□□□□, □ □□□□ Snowflake□ NULL □□ □□□□□. □□ D□ □□□□ □□□□ □□□□ □□□□, NULL □□ □□ □□□□ □□□□ □□□□□ □□□□ □□□ □ □□□□.

NEW QUESTION: 62
Snowflake□□ CSV, JSON, Parquet □□□ □□□ □□□ □ □□□ □□□ □□□ □□□□ □□ □□ □□□ □□□□ □□□□ □□ □□□ □□□□?
A. CSV □□□ □□ □□□ □□□□□ □□ □□ □□ □□□ □□□□□□.
B. Parquet □□□ □□ □□□ □□□□□ □□ □□ □□ □□□ □□□□.
C. □□ □□□ □□ □□ □ □□ □□□ □□□ □□□ □□□□.
D. JSON □□□ □□ □□ □□□ □□□□ □□ □□ □□ □□□ □□□□□□.
Answer: (SHOW ANSWER)

Snowflake□ Parquet □□□ CSV □ JSON□ □□ □□ □□□ □□ □□□ □□□ □□□□ □□ □□ □□□□ □□□ □□ □□ □□□ □□ □□ □□□ □□ □□□□ □□□□□□.

NEW QUESTION: 63
□□ □□□, □□□, □□□ □ □□□ □□ □□□ □□□ □ □□□□□□ □□□□ □□□ □□ □ □□□□ □□□ □□□□□?
A. □□□ □□ □□□ □ □□ □□□ □□ □□□ □□□ □□ □□□ □□□□□.
B. □□□ □□ □□□ □□□ □ □□□□□□ □□□ □□□ □□□□□.
C. □□□ □□□□ □□□□□ □□□ □□□□ □□□ □□□ □□□□.
D. □□□□ □□ □□□ □□ □□□ □□□□□.
Answer: (SHOW ANSWER)
□□□ □□ □□□ □□□ □□□ □□ □□□ □□□□ □□□ □ □□□□□□ □ □□ □□□ □□□□□.

NEW QUESTION: 64
□□ □□□□□ Parquet □□ □□□□ □□□ □□□ □□ □□□□□ □□□□. Parquet □□□ Snappy □□□ □□□□ □□□□ □□□□. □□□□ □□□ □□□□□ □□ □□□ □□□□□□ □ □□□□ □□□□□ □□□ □ □□ Snowflake □□□□ □□□□ □□□. □□□□ □□□ □□ □□□ □□ □□□□ □□□ □□ □□ □□□ □□□ □, □□ □□ □ □□ □□□□ □□ □□□ □□□ □□?
A. □□ □□□ □□□□ □□ □□ □□□□ □□□□□. Snowflake□ □□□□ □□□ □□□□□. □□ □□□□ □□ □□□□ □□ □□□□□.
B. VARIANT □□□ □□ Snowflake □□□□ □□□□□. □□ □□□□□□ □ □□□□ □□□□ □□□ □ □□ □□□ '(TYPE = PARQUET)'□ □□□□□. VARIANT □□□ □□ □□□□ □□□□ □□□□ □□□□ □□□□ □□ □□□□□.
C. 'FILE_FORMAT = (TYPE = PARQUET, COMPRESSION = 'AUTO')'□ Parquet □□□ □□□□ □□ □□□□ □□□□□. Snowflake□ □□□ □□□□ □□□□□. □□ □□□□ □□□ □□□□ □ □□□□.
D. □□ □□ □□□ □□□□ □□ 'COPY INTO'□ □□□□ Parquet □□□ Snowflake □□ □□□□ □□ □□□□□. Snowflake□ Snappy □□□ □□□□ □□□□ □□ □□□□ □□□□□□.
E. 'AUTO_REFRESH = TRUE'□ □□□ □□ □□□□ □□□□ Parquet □□□ □□ □□□□□. □□ □□□□□ □□□ □ 'FILE_FORMAT = (TYPE = PARQUET, COMPRESSION = 'SNAPPY')' □□ □ □□□□□. □ □□ □□□□ □□ □□ □□ □□□□□. □□ □□□□□ '□□□ □□'□ □□□□□.
Answer: B (LEAVE A REPLY)
□□ □□□□ □□□□ 'COMPRESSION = 'SNAPPY"□ □□ □□□ □□□□□ □□□□ Snowflake□ □□□□ □□□□□ □□□ □ □□□□. 'AUTO REFRESH = TRUE'□ □□□ □□□ □□□□, □□ □□□□ □□ □□□□ Parquet□ □□□ □□□□□ □□□ □ □□□□. 'ANALYZE TABLE'□ □□□□ Snowflake□ □□ □□ □□□ □□□□ □ □□□□. Snowflake□ □□□ □□ □□□ □□ □□ □□□ □ □□□, □□□□□ □□□□ □ □□ □□ □□□ □□□□ □□□ □□□□ □ □□□□. 'OPTIMIZE TABLE'□ □□ □□□□□ □□□□ □□□□. □□ □□□ □□ □□□□ □□ □□□ □ □□ □□ □□□ □□ □□□. □□ □□□ □□□□ □□ Parquet □□□ □□□ □□□□ □□□□ □□□□ □□ □□□ □□□□.

NEW QUESTION: 65
□□ □□□ □□□ 'CUSTOMER DATA'□□ Snowflake □□□□ □□□□. Snowflake Marketplace□□ □□□□ □ □□ □□□ □□(Share A, Share B)□ □□□□ □ □□□□ □□□□□ □□□. Share A□ □□ □□ □□□ □□□□, Share B□ □□ □□ □□□ □□□□□. □ Share □□ 'CUSTOMER ENRICHMENT'□□ □□□ □□ □□□□ □□□, 'CUSTOMER ID'□□ □□ □□ □□□□. □□ □□ □□□□ □□ □□ □□ □□□ □□ □□□(□: 700) □□□ □□□□□ □□ □□ □□□□ □□□□□ □□ □□□. □□ □ □□ □□ □□□ □□ □□□□ □□□□ □□□□□, □□□ □□ □□ □□□ □□□ □□□□□ □□□ □ □□□□?
A. □ □□□ □□□ □□ □□□ □ □ □□ □□□ □□, □□ □□□ □□ □□ □□□□ □□□□ □□□□□ □ □□ □□□□□.
B. 'CUSTOMER_DATA' □□□□ □□□□ □ □□□ □□ □□ □□□ □□ □□ □□□ □□□□, □□ □□ □□□□ □□□ □, □□□ □□□□ □ □□□□ □□□□ □□ □□□□□ □□□□□□.
C. 'CUSTOMER_DATA' □□ □□ □('CUSTOMER_ENRICHMENT' □(Share A □ Share B))□ □□ □□□ □□□(CTE)□ □□□□ □□□□ □□ □□ □□□□, Share B□ □□□ □□□ □□□ □□□ □□□ □□ □□ □□□ □□ □□□ □□□□□ □□□□□□.
D. □ □□□ □□ □□□□□ □□□□ □□□□ □□□ □□□□ □□ □□ □ □□ □□ □□□□ □□□□□.

E. '□□ □□□'□ □□ □(□□ □ A□ □□ □ B□ □□ □□ □□) □ □□ □□□□ □□ □□ □□□□ □□□□ □□□□□ □□□ □□□□□ □□□□ □□□ □□□□□. □□□ □□□ □□ □□□ □ □□□□□.

Answer: C (LEAVE A REPLY)

□□ C□ □□ □□□□ □□□□□ □□□ □□□□ □□ □□□□□. CTE□ □□□□ □□ □□ □□ □□ □□ □□□ □□□ □ □□ □□ □□□□□ □□□□ □□□□ □□□ □□□□ □□□□ □□ □□□ □ □ □□□□. □□ □□□ □□ □□ □□□ □□□ □□ □□□□□ □□□□ □□ □□□□□. □□ □□□□ □□ □ □□ □□□ □□□□□. □□ A□ □ □□□ □□□ □□□ □□□□ □□ □□, □□ B□ □□□ □□□□ □□□□ □□ □□□ □□□□□. □□ D□ □□ □□□□□□ □□□□ □□□□. □□ E□ □□ □□ □□□ □□□□□. CTE□ □□□□ □□ □□ □□□□ □□□□ □□□ □□□ □□□ □ □□□□.

NEW QUESTION: 66

□□□ □□□□ □□ □□□□□□ □□ □□□□ □□□ □□□□. □ □□□□□□ □□□□ □□ □□□□ □□□ □□□□ □□□□ □□□ □□□ □□□□ □□□□. □□ □□□□ Snowsight□□ □□ □□ □□□□□ □□□□□□. □□□□ □□ □□□□□ □□□□□□ □□ □□ □□□□□□□ □□□□□ □□□□□ □□□ □□□□ □□□?

A. □□□□□ □□□□□ □□□□□.

B. □□□□□ □□□□□ □□□□ □□□□□□.

C. □ □□□□□ □□ □□□ □□□ □□ □□□□□ □□□□□□.

D. QUERY_HISTORY □□ □□□□ □□ □□□ □□ □□ □□□□□ □□□□□.

Answer: A (LEAVE A REPLY)

Snowsight□□□ □□□□□ □□□□□ □□□□ □□□□ □□□ □□□□ □□ □□□ □□□□ □□□ □ □□□□. □□□ □□□□ □□□ □□□□□□ □□□ □□ □ □□□ □□ "□□ □□"□ □ □ □□□ □□, Snowflake□ □□ □□□□□ □□□□ □□□ □□ □□□□ □□□ □□□□□.

□□ "□□ □□□□□ □□" □□□ □□ □□□□□ □□□□□ □□□□ □□□□. □□□□ □□□□ □□□ □□□□ □□□ □□□ □□ □□□□ □□ □□□□□ □□□ □ □□□□. □ "□□" □□ "□□□□□□ □□" □□□ □□□□ □□ SQL □, □□□ □□□ □□, □□□□□ □□ □□□ □□□ □□(□□)□ □□ □□□□□.

□□□ □□□□:

* □□ B□ C□ □□□□□ □□□□□ □□ □ □□ □□(□□ □□ □□□□, □ □□, □□ □□ □□)□ □□□□□. □□ □□ □□□ □□□ □□□ Snowsight□□ □□□□ □□□□□ □□□□ □□□ □.

* □□ D□ □□ □□□□ □□□□ □□ □□□□, □□□□ □□□ □□□ □□□ □□ □□ □□□ □□□□□□□ □□ □□□ □□□ □□□□ □□□.

* □□ A□ 100% □□□□□. □□□□□ □□□□□ □□ □□□ □□ □□□□□ □□□□ □□□□ □ □□ □□□□□ □□ □□□ □□ □□□□□□ □□ □□□ □□□ □ □□□□. □□□□□ □□ □□□ □□□□ □□□ "□□"□ □□ □□□□ □□□ □□□ □□□□□ □□ □□□□□.

NEW QUESTION: 67

□□□ Snowflake□ □□□□ □□ □□□□ □□□□ □□□□. □□ □□ □□□ □□□□ □□□□□ □□□ □□□□□. □□ □□□□ 'SALES DATA'□□ 'SALE DATE', 'PRODUCT ID', 'CUSTOMER D', 'SALE AMOUNT'□ □□ □□ □□ □□□ □□ □□ □□□□ □□□□. □□□□ □□□ □□□ □ □□□□ 'SALE DATE'□ □□□□□. □□ □□ □□□ □□ □□ □□□ □□□ □□□□ □□□□. □□□ □□ □□□□□ □□□ □□□ □□□□□□ □□□□ □□□ □□□□□ □□□. □□ □□□ □□□□ □□ □□ □□ □ □□ □□□ □□ □□□□ □□□?

A. 'SALES DATA' □□□□ 'SALE DATE' □□□ □□□□ □□□□□ □□□□ □□□□□.

B. 'PRODUCT_ID'□ 'CUSTOMER ID'□ □□□□ □□ □□ □□□□ □□□□ □□□□ □□ □□□□□.

C. □□ □□□ □□□□□ □□□□.

D. '□□ □□□' □□□□□ '□□ □□'□ □□ □□ □□□□ □□□□□.

E. 'SALES_DATX' □□□□ 'SALE_DAT' □□□□ □□□□□.

Answer: A (LEAVE A REPLY)

'□□□'□ □□□□ □□□□ □□□□□□□ □□ □□ □□□□ □□ □□□□□. □□□ □□ □□□□ □□□□ □□□□ □□□□ '□□□'□ □□□□ □□□□ □□□□□ □□□□□ □□ □□ □ □ □□□ □□□ □□ □□□□□. □□□□ □□ □□□□ □□ □□□ □□□□ □□ □□□□ □□□□ □□ □ □□□□. □□□□□ □□□ □□□ □□□ □□□□ □□□□□ □□ □□□ □□□□□ □

□□□ □□□ □□□□ □□□, □□ □□ □□□□ □□ □□ □□□□ □□□□□□□ □□□□ □□□□□. Snowflake□ □□□ □□ □□□□□ □□□□ □□□□. □□□ □□ A□ □□ □□□ □□ □□□.

DAA-C01 □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ DAA-C01 □□! DumpTop □ □□ **DAA-C01** □□ □□□ □□□□□□, DumpTop DAA-C01 □□ □□□ □□□□□□□□ □□ □ □□□□□□□**.** □□□□ □□□ □□□□ □□ DumpTop DAA-C01 □□□ □□□□□. https://www.dumptop.com/Snowflake/DAA-C01-dump.html (**67** Q&As Dumps, **30%OFF Special Discount: KrDump**)