

Snowflake.DAA-C01.v2026-10-13.q66

□□□□:	DAA-C01
□□□□:	SnowPro Advanced: Data Analyst Certification Exam
□□□:	Snowflake
□□ □□ □□□:	66
□□:	v2026-10-13
# □□ □:	145
# □□ □□□:	660
https://www.krdump.com/Snowflake.DAA-C01.v2026-10-13.q66.html	

NEW QUESTION: 1

- □□ □□ □□ □□ □□□□ □□□□ □□□□ □□□□□?
- A. □□□□□□ □□ □□□ □□ □□□□ □□□ □□□□□ □□□□□.
- B. □□□□ □□ □□ □□□ □□□□ □□□□ □□ □□□ □□□□□□.
- C. □□□□ □□ □□□ □□□ □□□□ □□□ □□□□□.
- D. □□ □□ □□□ □□□ □□□ □□□□□ □□□ □□□□ □□□.

Answer: (SHOW ANSWER)

□□□□ □□ □□ □□□ □□□□ □□□□ □□ □□□ □□□□□□.

NEW QUESTION: 2

- □□ □□(JSON, CSV, Parquet)□□ □□□□ □□□ Snowflake□ □□ □□ □□□□ □□□□ □□□ □□□□□□. □□ □□□□□ □□□ □□, null □ □□ □□, □□ □ □□□ □□□□ □□□□□. □□□□ □□□ □□□ □□ □□□ □□□□ □□□ □□ □□□□ □□□□ □□□ □□□ □□ □□□□□□ □□□□ □□□□. □ □□□ □□□ □□ □□ □□□□ Snowflake □□ □ □□ □□ □□□ □□□□□?
- A. □□□ □□□ □□□□ Snowpipe□ □□□□ Snowflake□ □□ □□□ □□ □□□□ □□□□□. □□ □□□ □□□ □□ □□□ □□□□□.
- B. Snowflake□ □□□ □□□□ □□□ □□□ □□□□□□ □□ □□□□, □□□ □□□□□ □□□ □□ □□□ □□□□, Snowpipe□ COPY INTO□ COPY INTO □ □□ □□□ □□□ □□ □□□□ □□□□□.
- C. Snowflake Connector for Python□ □□□□ □□□ □□ □□□ □□ □□□□□□□ □□□□, Snowflake□ □□□□ □□□□ □□ □□□□□□ □□□ □□□ □□□ □□ □ □□ □□□ □□□□□. □□ □□□ □□ □□ □□□□ □□□ □□□□□.
- D. Snowflake□ □□□ □□ ETL □□□ □□□□, ETL □□ □□□ □□□ □□ □ □□ □□□ □□□ □□, □□□□ □□□□ □□□□ □□□ □□□□ Snowflake□ □□□□□.
- E. □□□ □□ □ □□□□ □□□□ □□□ □□□ □□□□ □□ □□□ □□ □□□ □□□ □□ □□□□ □□□□□□. □□□□ □□□ □□□ □□ Snowflake□ □□□ □□□□ □□□ □□ □□□□ □□ □□□□□□.

Answer: (SHOW ANSWER)

□□ B, C, D, E□ □□ □□□ □□ □□□ □□□□□. B□ □□□ □□□□□ □□ □□ □□□ □□□□□. C□ Python □□□□ □□□□ □□□ □□ □□□ □□□ □ □□□ □□ □. D□ □□ ETL □□□ □□□□□. E□ □□ □□□□ □□□□ □□ □□ □ □□□ □□ □□□ □□□□□. A□ □□ □□□ □□□□ □□□□ □□ □□□ □□□□□□ □□□ □□□. □□□ □□ □□□ □□□ □□□□□□ □□ □□□□ □□□□, □□□ □□ □□ □□ □□ □□□□ □□ □□□ □□ □□□ □□□□, □□□ □□□□ □□□ □□ □□□ □□□□□ □□□□.

NEW QUESTION: 3

1. 如何计算转化率？请提供 SQL 查询。

- A.** `SELECT * FROM LEADS JOIN CUSTOMERS ON LEADS.CUSTOMER_ID = CUSTOMERS.ID`, Snowflake `JOIN` `ON` `LEADS.CUSTOMER_ID = CUSTOMERS.ID`. Snowflake `JOIN` `ON` `LEADS.CUSTOMER_ID = CUSTOMERS.ID`.
- B.** Snowsight `SELECT * FROM LEADS JOIN CUSTOMERS ON LEADS.CUSTOMER_ID = CUSTOMERS.ID`. Snowsight `SELECT * FROM LEADS JOIN CUSTOMERS ON LEADS.CUSTOMER_ID = CUSTOMERS.ID`.
- C.** `SELECT * FROM LEADS JOIN CUSTOMERS ON LEADS.CUSTOMER_ID = CUSTOMERS.ID`. `ETL` `SELECT * FROM LEADS JOIN CUSTOMERS ON LEADS.CUSTOMER_ID = CUSTOMERS.ID`.
- D.** `SELECT * FROM LEADS JOIN CUSTOMERS ON LEADS.CUSTOMER_ID = CUSTOMERS.ID`. `SELECT * FROM LEADS JOIN CUSTOMERS ON LEADS.CUSTOMER_ID = CUSTOMERS.ID`.
- E.** `SELECT * FROM LEADS JOIN CUSTOMERS ON LEADS.CUSTOMER_ID = CUSTOMERS.ID` Python `SELECT * FROM LEADS JOIN CUSTOMERS ON LEADS.CUSTOMER_ID = CUSTOMERS.ID` (UDF) `SELECT * FROM LEADS JOIN CUSTOMERS ON LEADS.CUSTOMER_ID = CUSTOMERS.ID`. Snowpark DataFrame API `SELECT * FROM LEADS JOIN CUSTOMERS ON LEADS.CUSTOMER_ID = CUSTOMERS.ID`.

Answer: A (LEAVE A REPLY)

Snowflake is a cloud data warehouse that provides a single source of truth for your data. It is a fully managed, multi-tenant, and scalable data warehouse that can be used for a wide range of data processing and analytics tasks. Snowflake is designed to be easy to use, flexible, and secure, and it can be integrated with a variety of data sources and tools. Snowflake is a cloud-native data warehouse that is built on a distributed architecture, which allows it to scale horizontally and vertically. It is designed to be easy to use, flexible, and secure, and it can be integrated with a variety of data sources and tools. Snowflake is a cloud-native data warehouse that is built on a distributed architecture, which allows it to scale horizontally and vertically. It is designed to be easy to use, flexible, and secure, and it can be integrated with a variety of data sources and tools.

NEW QUESTION: 4

'PRODUCT SALES'□□ Snowflake □□□□ □□, □ □□□□□ 'PRODUCT ID (INT)', 'SALE DATE (DATE)', 'SALES AMOUNT' □□ □□□□. 'PRODUCT ID'□ 'SALE DATE'□ □ □□□ □□ □□□□ □□□□ □□□ □□□ □□□ □□□□□ □□□. □□ □ Snowflake□□ □ □□□ □□□□ □□ □□□□ □□□ □□□□, □ □□□ □□□□□?

- A.** 'ROW NUMBER()'은 데이터베이스 쿼리에서 'PRODUCT ID'와 'SALE DATE'를 기준으로 데이터를 정렬하고 번호 매기는 데 사용됩니다. 이는 데이터의 순서를 지정하고 중복된 값을 식별하는 데 유용합니다.
- B.** 'ROW'와 'ROWID'는 데이터베이스 테이블에서 각 행을 식별하는 데 사용됩니다. 'ROWID'는 Snowflake에서 행의 물리적 위치를 나타내며, 'ROW'는 논리적 행을 나타냅니다.
- C.** 'ROW ID'와 'ROWID'는 데이터베이스 테이블에서 각 행을 식별하는 데 사용됩니다. Snowflake에서는 'ROWID'가 Snowflake ID를 나타내며, 'ROW ID'는 Snowflake ID를 나타냅니다.
- D.** Snowflake에서는 'ROWID'가 Snowflake ID를 나타내며, 'ETL'은 데이터 추출, 변환, 적재 프로세스를 나타냅니다. Snowflake에서는 'ROWID'가 Snowflake ID를 나타내며, 'ETL'은 데이터 추출, 변환, 적재 프로세스를 나타냅니다.
- E.** Snowflake에서는 'UDF'가 사용자 정의 함수를 나타내며, 'UDF'는 사용자 정의 함수를 나타냅니다. Snowflake에서는 'UDF'가 사용자 정의 함수를 나타내며, 'UDF'는 사용자 정의 함수를 나타냅니다.

Answer: C,D (LEAVE A REPLY)

□□ C□ D□ □□ □□□□ □□□□□. □□ UNIQUE □□ □□□ □□□ □□ □□□ □□□□□ □□□□□. ETL □□□□□□□□ □□□ □□□ □□□□ □(D)□ □□□
 □□□□□□□ □□□□ □□ □□□□□. □(A)□ □□□ □□□ □□ □□□, □□ □□□□(B)□ □□ □□□□ □□□□ □□ □□ □□□ □□□□□ □□□□□. □□□□ □□ □□
 (UDF)□ □□□ □ □ □□□ □ □□□□□□□□ □□□ □□□ □□□□□.

NEW QUESTION: 5

1. CSV to Snowflake: `copy into 'HHMISS.csv' from (select * from 'data_20240101 120000.csv'). '2024-01-01'`
 2. Snowflake to target table: `copy into 'target table' from (select * from 'HHMISS.csv').`

- A.** `COPY INTO` is used to copy data from a table to a new table or a new location in a data lake. It is used to create a new table or a new location in a data lake.
- B.** `COPY INTO` is used to copy data from a table to a new table or a new location in a data lake. It is used to create a new table or a new location in a data lake.
- C.** `COPY INTO` is used to copy data from a table to a new table or a new location in a data lake. It is used to create a new table or a new location in a data lake.

D. □□ □□ □□ □□□ □□ Snowpipe□ □□□□ '2024-01-01' □□□ □□□ □□□ □□□□ □□□□ □□□□□ □□□□□.

E. METADATA\$FILENAME □ □ □ □ □ □ □ □ □ □ COPY INTO □ □ □ □ □ □ . □ □ □ □ □ □ □ □ □ □ COPY INTO □ □ □ □ □ □ □ □ □ □.

Answer: B,E (LEAVE A REPLY)

```

COPY B0 00 000 0000000. COPY INTO 000 METADATA$FILENAME 0000 0000 00 000 0000 000 000 0 00000. 00 E0 'METADATA
$FILENAME' 00 00 0000 00 00000 000 000 0000 00 0000 000 0000 000000 'COPY INTO' 0 000 00 000 00000 0 0000
0000 00000. 00 A0 00000 00 00000000. 00 C0 00000 00 000 00000. 00 D0 00 E00 000 0 00000. Snowpipe0 00000 0
0 COPY 000 00 0000 00 000 00000.

```

NEW QUESTION: 6

□ □□□□ □□□□□ □□ □□□ □□ □□□□ □□□□ □□□□ □□□□ □□□□. □□□□ □□□ SQL □□□□ □□□□ □□□□ □□□□ □□□□, □□□□ □□□□ □ □ □□
 □□□□□. □□□□ □□ □□ 8□ □□□□ □□□□ □□□ □□□□. □□, □□ □□□ □□□□□□ □□ □□ □□ □□ □ □□ □□□□ □□□□□□. □□□ □□ □□□□ □□□
 □, □ □□□□ □□□□□ □□□□□ □□□□□ □□ □□ □□□ □□□□□?

A. Snowflake □□□ □□□□ □□□ SQL □□□ □□ □□□□ Snowflake□ □□□ □□ □□□ □□□□ □□□ □□□□ □□□□□□□□. □□ □□□ □□□ □□□ □□□□□□□□
□□□□□ □□□.

B. 下列 SQL 语句中，哪条语句是错误的？
 A. 在 SQL 中，SELECT 语句的 WHERE 子句可以放在 HAVING 子句之前。
 B. 在 SQL 中，SELECT 语句的 WHERE 子句可以放在 HAVING 子句之后。
 C. 在 SQL 中，SELECT 语句的 WHERE 子句可以放在 HAVING 子句之前，也可以放在 HAVING 子句之后。
 D. 在 SQL 中，SELECT 语句的 WHERE 子句可以放在 HAVING 子句之前，也可以放在 HAVING 子句之后，也可以放在 HAVING 子句之前和之后。

C. □□ ETL □□□ □□□□ □□□□ □□, □□ □ □□□ □□. ETL □□□ □□ □ □□□□ □□□□ □□□□ □□□□□.

D. Snowpipe □ □□□ □□□□ □□□ □□□□ □□□□□ □□□□□. □□ □□□□ □□ □□□□ SQL □□□ □□□□ □□□□ □□□□□. □□□□ □□□□ □□□ □□□□ □□□□ □□□ □□□□□.

E. Snowflake □ □□□ □□□□□□□ □□□□ □□ □□□ □□ □□ □□□ □□ □□□□ □□□□□.

Answer: B (LEAVE A REPLY)

[illegible]

NEW QUESTION: 7

[illegible]

A. 1. 'ST_SPHERE_GRID' □□□ □□□□ 'city_boundary'□ □□ □□□ GEOGRAPHY □□ □□□□ □□□□□. 2. 'ST_CONTAINS' □□□ □□□□ □ □□□ □ □□ □□□□
□□ □□□ □□□□□. 3. □ □□ □□ □□□ □□□□□.

B. 1. 'ST_HEXGRID' □□□ □□□□ 'city_boundary'□ □□ □□□ GEOGRAPHY □□ □□□□ □□□□□. 2. 'ST_INTERSECTS' □□□ □□□□ □ □□□ □□ □□□□ □□ □□ □□□□□. 3. □ □□ □□ □□ □□ □□□□□.

C. 1. 'ST GRID' □□□□ □□□□ 'city_boundary'□ □□ □□□□ GEOGRAPHY □□ □□□□ □□□□□. 2. 'ST CONTAINS' □□□□ □□□□ □ □□□□ □ □□ □□□□ □□ □□□ □□□□□. 3. □ □□ □□ □□□ □□□□□.

D. 1. 'ST_HEXGRI' □□□□ □□□□ 'city_boundary'□ □□ □□□ GEOGRAPHY □□ □□□□ □□□□□. 2. 'ST_WITHIN' □□□□ □□□□ □ □□□ □□ □□□ □□ □□□□ □□□□□. 3. □ □□ □□ □□ □□□ □□□□□.

E. 1. 'ST_HEXGRID' □□□ □□□□ 'city_boundary'□ □□ □□□ GEOGRAPHY □□ □□□□ □□□□□. 2. 'ST CONTAINS' □□□ □□□□ □ □□□ □□□ □□□ □□ □□ □□□. 3. □ □□ □□ □□□ □□□□□.

NEW QUESTION: 10

Which of the following Snowflake SQL queries will create a model named SALES_FORECAST_MODEL using the SALES_DATA table as input and the SALE_DATE column as the target column? The model should have a horizon of 7 days and a confidence of 95%.

```
CONFIDENCE = 0.95; SELECT FROM TABLE(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 HORIZON = 7 CONFIDENCE = 0.95; SELECT FROM TABLE(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 HORIZON = 7; SELECT FROM TABLE(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 CONFIDENCE = 0.95; SELECT FROM TABLE(FORECAST(MODEL_NAME=>'SALES_FORECAST_MODEL', INPUT => (SELECT DATEADD(day, row_number() OVER (ORDER BY NULL), CURRENT_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 CONFIDENCE = 0.95; SELECT FROM TABLE(FORECAST(MODEL_NAME=>'SALES_FORECAST_MODEL', INPUT => (SELECT DATEADD(day, seq4(), CURRENT_DATE()) AS SALE_DATE FROM TABLE(GENERATOR(rowcount => 7))));
```

- A. ☐ A
- B. ☐ B
- C. ☐ C
- D. ☐ D
- E. ☐ E

Answer: (SHOW ANSWER)

The correct answer is B. The query in option B creates a model named SALES_FORECAST_MODEL using the SALES_DATA table as input and the SALE_DATE column as the target column. The model has a horizon of 7 days and a confidence of 95%. The query uses the FORECAST function to create the model and the GENERATOR function to generate a sequence of dates for the input.

NEW QUESTION: 11

Which of the following Snowflake SQL queries will create a model named SALES_FORECAST_MODEL using the SALES_DATA table as input and the ORDER_DATE column as the target column? The model should have a horizon of 7 days and a confidence of 95%.

- A. ☐ The query in option A creates a model named SALES_FORECAST_MODEL using the SALES_DATA table as input and the ORDER_DATE column as the target column. The model has a horizon of 7 days and a confidence of 95%. The query uses the FORECAST function to create the model and the GENERATOR function to generate a sequence of dates for the input.
- B. ☐ The query in option B creates a model named SALES_FORECAST_MODEL using the SALES_DATA table as input and the ORDER_DATE column as the target column. The model has a horizon of 7 days and a confidence of 95%. The query uses the FORECAST function to create the model and the GENERATOR function to generate a sequence of dates for the input.
- C. ☐ The query in option C creates a model named SALES_FORECAST_MODEL using the SALES_DATA table as input and the ORDER_DATE column as the target column. The model has a horizon of 7 days and a confidence of 95%. The query uses the FORECAST function to create the model and the GENERATOR function to generate a sequence of dates for the input.
- D. ☐ The query in option D creates a model named SALES_FORECAST_MODEL using the SALES_DATA table as input and the ORDER_DATE column as the target column. The model has a horizon of 7 days and a confidence of 95%. The query uses the FORECAST function to create the model and the GENERATOR function to generate a sequence of dates for the input.
- E. ☐ The query in option E creates a model named SALES_FORECAST_MODEL using the SALES_DATA table as input and the ORDER_DATE column as the target column. The model has a horizon of 7 days and a confidence of 95%. The query uses the FORECAST function to create the model and the GENERATOR function to generate a sequence of dates for the input.

Answer: B (LEAVE A REPLY)

[illegible]

Answer: A,C (LEAVE A REPLY)

[illegible]

NEW QUESTION: 14

□□ □□ □□ □□ □ □□□ □□□□ □□ □□□ □□□□ □□□?

A. CREATE TABLE DEV.PUBLIC.TRANS_HIST LIKE PROD.PUBLIC.TRANS_HIST;

B. CREATE TABLE DEV.PUBLIC.TRANS_HIST AS (SELECT * FROM PROD.PUBLIC.TRANS_HIST);

C. CREATE TABLE DEV.PUBLIC.TRANS_HIST CLONE PROD.PUBLIC.TRANS_HIST;

D. CREATE TABLE DEV.PUBLIC.TRANS_HIST AS (SELECT * FROM PROD.PUBLIC.TRANS_HIST WHERE extract(year from (TRANS_DATE)) = 2019);

Answer: C (LEAVE A REPLY)

Snowflake 0 0 0 0 (Zero-Copy Cloning) 0 0 0 0 0 0 0 0 0 0. 0 0 0 0 0 0 0 0 0 0 (0 0 0 0 0 0) 0 0 0 0 0 0 0 0 0 0 0 0, 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0. CLONE 0 0 0 0 Snowflake 0.

Snowflake es un proveedor de almacenamiento de datos en la nube, diseñado para ser escalable, seguro y fácil de usar. Se integra con una amplia gama de aplicaciones y servicios de la nube, lo que lo convierte en una opción popular para empresas que buscan optimizar su infraestructura de datos.

□ □ □ □ □ □ :

* □□ A(LIKE)□ □□□□ □ □□□ □□□ □□□□□□. □□□□ □□□ □□□□ □□□□ □□ □□□ □□□□□ □□□, □□□□□□ □□□ "□□□"□ □□□□□ □□□□□.

* □□ B(CTAS - Create Table As Select)□ □□□ □□ □ □□□ □□□□□. □□ □□ □□□ □□□ □□□□ □□□□ □□□□ □□□□ □□ □□□ □□ □□□□
□.

* □□ D□ □□□□ CTAS □□□□□. □□ □□□□□ □□□ □□ □□ □ □□□, 2019□ □□□ □□ □□□ □□□ □□ □□□ □□□□□ □□□ □□ □□□ □□□□□.

* □□ C□ 100% □□□□□. □ □□□ Snowflake□ □□ □□□ CLONE □□□□ □□□□□, □□ □□/□□□ □□□ □□ □□□□ □□ □□ □□ □□□ □□□ □□□□□ □□ □□□□□. □□ □□□ □□ □ □□□ □□□ □□□ □□ □□□□□.

NEW QUESTION: 15

[illegible]

- A.** 'TRANSACTIONS' □□□□ □□ □ □□ □□ □□ □□□□ 'DEPARTMENT_ACCESS' □□□□ □□□□ □□ □□□□□. □ □□ 'Compliance' □□□ 'SELECT' □□□ □□□□□. □ □□ □□□ □□□ □ □□ □□□ □□□□, Compliance □□□ □□□ □□ □□□ □□□□ □□ □□□ □□□ □□ □□□ □□□ □□□□□.
- B.** 'ACCOUNT_NUMBER' □□ □□ □□ □□□ □□□ □□□ □□□□ □□ □□□□ □□□ '□□ □□'□□ □ □□□ □□□□ □□ □□□ □□ □□□□ □□□□□ □□□□ □□□□ □□ □□□□□. □ □□□ □□ □□□ □□□ □□□□□ □□ 'ACCOUNT_NUMBER'□ □□□□, □□□ □□□ □□□□ □□□□□.
- C.** '□□□(CALLER)' □□□□ □□□□ □□ □□□□□ □□□□□. □□ □□□□ □□□ Snowflake □□□□ □□□ □□□□ □□□□ □□□ □□ □□□ □□□□□. □□□□ '□□ □□(Compliance)' □□□□ □□ □□ □□ □□□ □□, □□ □□□□□ □□□□□ □□ '□□ □□(ACCOUNT_NUMBER)'□ □□ □□□□□. □□□□ □□ □□, □□□□ □ □□□□□. '□□ □□' □□□ □□ □□□□□ □□ □□ □□□ □□□□□.
- D.** □□□□ □□ □□ □□ □□□ □□□□□. □ □□□ □□□□ □□□□□□ □□ □□□ □□□□ '□□ □□' □□ □□ □□□ □□□□□. □ □ □□□ □□□□ '□□ □□'□ □□□ □□□□ □□□ □□□ □□□ □□□ □□□□□. □ □ □□□ □□□□ □□ □□□ □□□ □□□□□.
- E.** '□□ □□' □□□ □□ 'UNMASK' □□□□ □□□□□□□. □□□□ □□ □□ □□ □□□□ 'TRANSACTIONS' □□□□ □□□ □ 'ACCOUNT_NUMBER' □□ □□ □□ □□□ □□ □ □□□ □□□□□□ □ □□□ □□□ □□□□□.

Answer: ([SHOW ANSWER](#))

00 B0 00 0000 000 0000 0000000. 00 000 000 000 0000 CASE 00 00000, 0000 '00 00' 000 00 00 0 000 0000 00 0
0 0000 00 000 000 000 0000 'ACCOUNT_NUMBER' 00 0000 000 000 0 0000. 000 00 000 000 00000 000 0000 0000
0 000 000 00 0000 00000. 00 E0 UNMASK 000 00 00000 00 0000 00 0000 0000 000 000 0000. 000 00(00 C)0 000
0 00 0 000 000 00000. 00 00 000 0000 0000 000 000 00 00 000 00000. 00 A0 0000 0000 000 000 0 000, 00 0
0000000 0 00 00 0000 00 0000 000 00000.

NEW QUESTION: 16

Google Cloud Storage(GCS) 1000 1000 1000 JSON 10000 'CUSTOMER DATA'100 Snowflake 10000 10000 10000 1000. 1000 100 1000000 100 1000000 10000 1000000 10000 1000. JSON 1000 GZIP100 10000 1000, Snowflake100 DATE 10000 10000 100 '1000(registration_date)' 1000 10000 10000. 1000 100 100000 'registration_date' 1000 1000 1000 1000 10000 10000 10000. 1000 1000 100000 100000 100000 10000 100 1000 1000 1000 10000, 100 100 100 10000 1000 1000 1000 10000 10000. 100 100000 100 1000 100000 1000. 100 100 100 100 1000 100000 100 1000 1000 100000?

- A.** 'TYPE = JSON' □ 'COMPRESSION = GZIP'□ □□□□ □□ □□ □□□ □□□□□. 'ON_ERROR = SKIP_FILE' □□□ □□□□ COPY INTO □□□ □□□□□. □□□□ □□□□ □□□□ COPY HISTORY □□ □□□□ □□□ □□□ □□□ □ □□□□ □□□ □□□□□.
- B.** 'TYPE = JSON' □ 'COMPRESSION = GZIP'□ □□□□ □□ □□ □□□ □□□□□. □□ □□ 'TO DATE(registration_datey)'□ 'SON ERROR = CONTINUE'□ □□□□ COPY INTO □□□ □□□□□. □□□ □□□□ □□□□ □□ □□□ □□ □□□□ □□□□□.
- C.** 'TYPE = JSON' □ 'COMPRESSION = GZIP'□ □□□□ □□ □□ □□□ □□□□□. □□ □□ 'TRY TO DATE(registration_date)' □ 'ON ERROR = SKIP FILE'□ □□□□ COPY INTO □□□ □□□□□. □□□ □□□□□ 'registration_date' □□ NULL□□ □□□□ □□□ □□□□□ □□□□□.
- D.** 'TYPE = JSON' □ 'COMPRESSION = GZIP'□ □□□□ □□ □□ □□□ □□ Snowpipe□ □□□□□□□. □□□□ □□ □□ □□□ □□□□ □□□ □□□□ □□□□□□.
- E.** 'TYPE = JSON' □ 'COMPRESSION = GZIP'□ □□□□ □□ □□ □□□ □□□□□. □□ □□ 'TRY'□ □□□□ COPY INTO □□□ □□□□□. 'CUSTOMER DATA' □□□□ 'registration_date'□ □□ □□□□ □□□□ 'ON ERROR = CONTINUE'□ □□□□□.

Answer: E (LEAVE A REPLY)

0000 E0000. E0 00000 00000 00 0000 NULL0 00000 00000 00000, 00 00000 00000 00000 0000 0 00000. 'ON ERROR = CONTINUE' 0000 0
 0000 00 0000000 00000 00000. 0000 00000 00000 0000 00000 0000 000000. A 0000 00 0000 0000000 00 000000 00000 00 00000 0
 0000 00000. B 0000 00 0000 00 0000 0 00 00 00 0000000 000000. C 0000 0000000 0000 0000000 0000000. D 0000 Snowpipe0 00 0000
 00 00 0000 0000 00 0000000 0000 00 0000 0 00000000. 0000 E 0000 00 00000. 00000 00 00 0000 0000000 000000 0000 0000 0000
 00000.

- B. ''
- C. NULL
- D. □□□□ □□ □□□□ □□ □□□ □ □□□□ □□□ □□□□□.
- E. □□ □□□□□ □□□□ □□ □□□ □ □□□□ □□□ □□□□□.

Answer: (SHOW ANSWER)

'CUSTOMER DATA' □□□□ 'SSN' □□ 'mask_ssn' □□□ □□□ □□□□□. □□□ □□□□ □□□□ □□□ 'ANALYST_ROLE'□□□, □□□ □□□ □□ □□ □□□ □□□□ □□ 'SSN' □□ '-'□ □□□□□. □□ □□ □□□□ □□ □□□□ □□ □□□ □□ □□□ □ □□□□. □□ □□ □□□□ □□□□ □□ □□ □□□□ □□□□ □□□ □ □□□ □ □□□ □□ □□□□□.

NEW QUESTION: 19

Snowflake□□ BI □□ □□□ □□□□ □□ □□ □□□ □□ □□□ □□ □ □□ □□ □□□□ □□□ □ □□ □□□ □□□ □□□ □□ □□□ □□□□□? (□□□□ □□ □□ □ □□□□□□.)

- A. □□ □□ □□□
- B. □□□ □□ □□ □□ □□
- C. □□ □□ □□ □□ □□
- D. □□□ □□□□ □ □□□ □□ □□

Answer: A,D (LEAVE A REPLY)

□□ □□ □□□ □ □□□ □□□□ □□ □□□ □□ □□□ □□ □□□ □□ □□□ □□□ □ □□□ □□□□ □ □□□ □□□, □□ □□ □□□ □ BI □□□ □□□ □□□ □□□ □□□ □□□.

NEW QUESTION: 20

'website_visits'□□ □□□□□ 'user_id', 'visit_timestamp', 'page_view' □□ □□ □□□ □□ □□□□ □□□□ □□□□. □ □□□□ □□□□ □□□ □□ □□ □□(□)□ □□□□ □□□. 'visit_timestamp' □□ □□□ □□□ TIMESTAMP NTZ□□ □□□ □, □□ Snowflake SQL □□□ □□□□ □ □□ □□□ □□□□ □□□ □ □□□□?

- ☐ SELECT user_id, visit_timestamp, visit_timestamp - LAG(visit_timestamp, 1, visit_timestamp) OVER (PARTITION BY user_id ORDER BY visit_timestamp) AS time_difference_minutes FROM website_visits;
- ☐ SELECT user_id, visit_timestamp, DATEDIFF(minute, LAG(visit_timestamp, 1, visit_timestamp) OVER (PARTITION BY user_id ORDER BY visit_timestamp), visit_timestamp) AS time_difference_minutes FROM website_visits;
- ☐ SELECT user_id, visit_timestamp, TIMEDIFF(minute, LAG(visit_timestamp, 1, visit_timestamp) OVER (PARTITION BY user_id ORDER BY visit_timestamp), visit_timestamp) AS time_difference_minutes FROM website_visits;
- ☐ SELECT user_id, visit_timestamp, ROUND(TIMESTAMPDIFF(minute, LAG(visit_timestamp, 1) OVER (PARTITION BY user_id ORDER BY visit_timestamp), visit_timestamp)) AS time_difference_minutes FROM website_visits;
- ☐ SELECT user_id, visit_timestamp, AVG(visit_timestamp - LAG(visit_timestamp, 1) OVER (PARTITION BY user_id ORDER BY visit_timestamp)) AS time_difference_minutes FROM website_visits;

- A. □□ A
- B. □□ B
- C. □□ C
- D. □□ D

[illegible]

- A.** `SELECT * FROM TABLE1 WHERE ID = 1` is a valid SQL query to retrieve all data from `TABLE1` where `ID` is 1.
- B.** `SELECT * FROM TABLE1 WHERE ID = 1` is a valid SQL query to retrieve all data from `TABLE1` where `ID` is 1.
- C.** `SELECT * FROM TABLE1 WHERE ID = 1` is a valid SQL query to retrieve all data from `TABLE1` where `ID` is 1.
- D.** `SELECT * FROM TABLE1 WHERE ID = 1` is a valid SQL query to retrieve all data from `TABLE1` where `ID` is 1.
- E.** `SELECT * FROM TABLE1 WHERE ID = 1` is a valid SQL query to retrieve all data from `TABLE1` where `ID` is 1.

Answer: B (LEAVE A REPLY)

□□□□ □□ □□ □□□□ □□□□ □□□ □□□□□. □□□ □□ □□□ □□ □□□ □□ □□ □□□□ □□ □□□□ □□ □
□□ □□□□ □□□□ □□□□ □□□ □□□ □□□□□. □□ □□□ □□□□ □□□ □ □□□, □□ □□□ □□□□ □□□□ □□ □□□□. □□ □□□□ □
□□□ □□ □□□□□ □□ □□□□□□□. □□ □□□□ □□□ □□□□ □□ □□□ □□□ □□ □□□ □□□ □□ □□ □□□ □□□ □ □□□, □□□ □□□ □□□
□ □□□□.

NEW QUESTION: 24

[illegible]

- A.** `ALTER MATERIALIZED VIEW sales_mv REFRESH` forces Snowflake to refresh the materialized view immediately.
- B.** `ALTER MATERIALIZED VIEW sales_mv REFRESH` forces Snowflake to refresh the materialized view at the next scheduled refresh time.
- C.** `ALTER MATERIALIZED VIEW sales_mv REFRESH` forces Snowflake to refresh the materialized view at the next scheduled refresh time, but only if the view is stale.
- D.** `ALTER MATERIALIZED VIEW sales_mv REFRESH` forces Snowflake to refresh the materialized view at the next scheduled refresh time, but only if the view is not stale.
- E.** `ALTER MATERIALIZED VIEW sales_mv REFRESH` forces Snowflake to refresh the materialized view at the next scheduled refresh time, but only if the view is stale and the refresh is forced.

Answer: ([SHOW ANSWER](#))

□□□ □□ □□ □□□□ □□ □□ □□□ □□ □□□□ □□ □□□□ □□ □□ □□□□ □□ □□□ □□ □□□□. □□□□□ □□□ □□□ □(A) □□□ □□ □
□□. □□ □□ □□□□ □(B) □□□□□ □□ □□ □□□ □□□□□□. □□ □□ □□(D) □□ □□ □□□□ □□ □□□ □□□□□ □□□□□. □□ □□ □□□□ □(E)
□ □□ □□□□□□.

NEW QUESTION: 25

Snowflake ()

- A.** □□~~+~~□□ □□□ □□ □□
- B.** □□□ □□□ □□
- C.** □□ □□ □□ □□
- D.** □□□□ □□ □ □□□□

Answer: A,D (LEAVE A REPLY)

[illegible]

NEW QUESTION: 26

00000 000 0000 0000 000 000 000000 000 00000 0000 0000. 00000 000 000, 00 0, 0000 00, 00 000 0000, 0
 0000 Snowflake 000000 00000000. 00000 00 00(000 000 -> 00 0 -> 0000 00 -> 00) 000000 00 000 00000 000. 00 000
 0000 0000 0: "sql CREATE OR REPLACE TABLE website_events (event_timestamp TIMESTAMP NTZ, event_type VARCHAR(50), user_id VARCHAR(100), page_url

C. □□ □□ □□(□: Tableau, Power BI)□ □□□□ Snowflake□ □□□□ □□□□ □□□ □ CSV □□ □□□ □□ □□ □□□□ □□□□□ □□□□□. □□ □□□ □□□ □□□□ □□ □□□ □□ □□□ □□□□□□.

D. '01 01' 0000 Snowflake 0000 000000. 000 000 0000 0000 00 00 00(0: AWS Lambda)0 00000. Lambda 000 0000 0000 CSV 00 00 000 0 S30 00000 0000 000 000 00000 00000.

E. '0 0' 00000 0 00000 Snowflake 000 00000. 0 000 00 00000 0000 0000 0000, Snowflake 00000 0000 CSV 0000 000 0, CSV 000 00 00000 00000. 00000 Python 000 00 00(UDF) 00000 CSV 000 00 000 00000 00000. 000 000 000 00 0000.

Answer: ([SHOW ANSWER](#))

[illegible]

NEW QUESTION: 30

[illegible]

A. □□□ □□ □□□ □ □□ □□□ □□ □□□ □□ □□□ □□□□□.

B. ☐☐☐ ☐☐ ☐☐☐ ☐ ☐☐☐☐☐☐ ☐☐☐ ☐☐☐☐.

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

D. □□□□ □□ □□□ □□ □□□ □□□□□.

Answer: A (LEAVE A REPLY)

□□□ □□ □□□ □□□ □□□ □□ □□□ □□□□ □□□ □ □□□□□□ □ □□ □□□ □□□□□.

NEW QUESTION: 31

[illegible]

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

B. □□ □□□□ □□ □□ □□□ □□□□ □□ □□ □□□□□. Snowflake □□ □□□□□ □□□□ □□□ □□□□ □□□ □□ □□□□ □□□ □□□□□□□□.

C. □□□ □□□□ □□□□ □□□□□ □□□□ □□□ □□□□□ □□□ X-Large□ □□□□ Snowflake□ □□ □□□ □□□ □□□ □ □□□□.

D. □□ □□ □□□ □□□□ □□ □□□□ □□□□□. Snowflake □□ □□□ □□□□ □□ □□□□ □□□□ □□ □□□□□.

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

Answer: ([SHOW ANSWER](#))

[illegible]

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**)

NEW QUESTION: 32

Snowflake□□ □□ □□□ □□□ □□□□ □□□ □□ □□□ □□□□ □□□ □□□□ □□□□ □ □□□ □□□ □□□?

- A. □□□ □□□ □□□□ □□□□□.
- B. □□ □□□ □□□ □□ □□□ □□□ □□□□□.
- C. □□ □□□ □□□□ □□□ □ □□□□.
- D. □□□ □□□ □□□□□ □□□□□

Answer: C ([LEAVE A REPLY](#))

Snowflake□ □□ □□□ □□□ □□□□ □□ □□□□□ □□□ □□□□ □□□ □ □□ □□□ □□ □□□ □□□□ □□ □□□ □□□ □□□□ □□□ □ □□ □□□□ □□□□ □□□□ □ □□ □□□.

NEW QUESTION: 33

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

- A. □□□□ □□□□□ □□ □□□ □ □□□□ □□□□ □□□ □□ □□□.
- B. □□□□ □□□□□ □□ □□□ □□□□ □□□ □□(□□□□)□ □□□ □□ □□□ □□□□ □□□□ □□ □□□□, □□ □□ □□□□ □ □ □□ □□□□ □□□ □□□□ □.
- C. □□□□ □□ □□□□□ □ □□□ □□ □□□□ □□□□ □□□ □□ □□.
- D. □□ □□□ □□ □□ □□□□□ □□ □ □□□ □□□ □□□ □□ □□□□ □ □□□.
- E. □□ □□□ □□□□ □□□□ □□□□ □□□ □ □□□□ □□ □□□ □□□ □□□ □□□ □□ □□□□□ □□□□ □□ □□□, □□□ □□ □□ □□□ □□□□ □□□□ □ □□ □□ □□□ □□□□□.

Answer: ([SHOW ANSWER](#))

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

NEW QUESTION: 34

Snowflake Marketplace□ □□□□ □□□□ □□□ □, □□ □□□□ □□ □□□ □□□ □□□□ □ □□□ □□□ □□ □□□ □□□?

- A. □□□ □□□ Snowflake□□ □□□□ □□□ □□□□□ □□□□□.
- B. □□□ □□□ □□□ □□ □□□ □□□ □□□ □□□□ □□□.
- C. □□ □□□□ □□ □□□ □□□□□.
- D. □□□ □ □□ □□□ □□□ □□□□□.

Answer: B ([LEAVE A REPLY](#))

□□□ □□□ □□ □□□ □□□ □□□ □□□□, □□ □□□□ □□ □□□ □□□ □□□ □ □□□□.

NEW QUESTION: 35

Which of the following is a valid Snowflake table name?

- A. table with spaces and special characters.
- B. table with spaces and special characters.
- C. table with spaces and special characters.
- D. table with spaces and special characters.

Answer: D ([LEAVE A REPLY](#))

Which of the following is a valid Snowflake table name?

NEW QUESTION: 36

You have a table named 'EVENTS' with columns: 'EVENT ID', 'EVENT URL', 'EVENT DATE', 'EVENT TYPE', 'EVENT LOCATION'. You want to find the number of unique users who have visited the website. Which of the following SQL queries would you use?

- A. 'COUNT(DISTINCT USER_ID)'
- B. 'COUNT(DISTINCT USER_ID)'
- C. 'COUNT(DISTINCT USER_ID)'
- D. 'COUNT(DISTINCT USER_ID)'
- E. 'COUNT(DISTINCT USER_ID)'

Answer: A ([LEAVE A REPLY](#))

You have a table named 'EVENTS' with columns: 'EVENT ID', 'EVENT URL', 'EVENT DATE', 'EVENT TYPE', 'EVENT LOCATION'. You want to find the number of unique users who have visited the website. Which of the following SQL queries would you use?

NEW QUESTION: 37

Which of the following is a valid Snowflake table name?

- A. table with spaces and special characters.
- B. table with spaces and special characters.
- C. BI table with spaces and special characters.
- D. table with spaces and special characters.

Answer: ([SHOW ANSWER](#))

Which of the following is a valid Snowflake table name?

NEW QUESTION: 38

You have a table named 'CUSTOMER DATA' with columns: 'CUSTOMER ID', 'JOIN DATE', 'LAST ACTIVE DATE', 'REVENUE', 'CHURNED' (BOOLEAN). You want to find the number of unique customers who have churned. Which of the following SQL queries would you use?

- A. 'COUNT(DISTINCT CUSTOMER_ID)'
- B. 'COUNT(DISTINCT CUSTOMER_ID)'
- C. 'COUNT(DISTINCT CUSTOMER_ID)'

Answer: A (LEAVE A REPLY)

Which of the following is a correct statement about Snowflake's security model?

NEW QUESTION: 42

Which of the following is a correct statement about Snowflake's security model? Snowflake uses a role-based access control (RBAC) model. Roles are assigned to users, and permissions are granted to roles. Snowflake also supports a least-privilege security model, where users are granted only the permissions they need to perform their tasks. Snowflake also supports a separation of duties (SoD) model, where different users are granted different permissions to ensure that no single user has too much power. Snowflake also supports a data loss prevention (DLP) model, where sensitive data is protected by encryption and access controls.

- A. Snowflake uses a role-based access control (RBAC) model. Roles are assigned to users, and permissions are granted to roles.
- B. Snowflake uses a least-privilege security model, where users are granted only the permissions they need to perform their tasks.
- C. Snowflake also supports a separation of duties (SoD) model, where different users are granted different permissions to ensure that no single user has too much power.
- D. Snowflake also supports a data loss prevention (DLP) model, where sensitive data is protected by encryption and access controls.
- E. Snowflake also supports a data loss prevention (DLP) model, where sensitive data is protected by encryption and access controls.

Answer: (SHOW ANSWER)

Which of the following is a correct statement about Snowflake's security model? Snowflake uses a role-based access control (RBAC) model. Roles are assigned to users, and permissions are granted to roles. Snowflake also supports a least-privilege security model, where users are granted only the permissions they need to perform their tasks. Snowflake also supports a separation of duties (SoD) model, where different users are granted different permissions to ensure that no single user has too much power. Snowflake also supports a data loss prevention (DLP) model, where sensitive data is protected by encryption and access controls.

NEW QUESTION: 43

Which of the following is a correct statement about Snowflake's security model? Snowflake uses a role-based access control (RBAC) model. Roles are assigned to users, and permissions are granted to roles. Snowflake also supports a least-privilege security model, where users are granted only the permissions they need to perform their tasks. Snowflake also supports a separation of duties (SoD) model, where different users are granted different permissions to ensure that no single user has too much power. Snowflake also supports a data loss prevention (DLP) model, where sensitive data is protected by encryption and access controls.

- A. Snowflake uses a role-based access control (RBAC) model. Roles are assigned to users, and permissions are granted to roles.
- B. Snowflake uses a least-privilege security model, where users are granted only the permissions they need to perform their tasks.
- C. Snowflake also supports a separation of duties (SoD) model, where different users are granted different permissions to ensure that no single user has too much power.
- D. Snowflake also supports a data loss prevention (DLP) model, where sensitive data is protected by encryption and access controls.
- E. Snowflake also supports a data loss prevention (DLP) model, where sensitive data is protected by encryption and access controls.

Answer: (SHOW ANSWER)

Which of the following is a correct statement about Snowflake's security model? Snowflake uses a role-based access control (RBAC) model. Roles are assigned to users, and permissions are granted to roles. Snowflake also supports a least-privilege security model, where users are granted only the permissions they need to perform their tasks. Snowflake also supports a separation of duties (SoD) model, where different users are granted different permissions to ensure that no single user has too much power. Snowflake also supports a data loss prevention (DLP) model, where sensitive data is protected by encryption and access controls.

NEW QUESTION: 44

Which of the following is a correct statement about Snowflake's security model? Snowflake uses a role-based access control (RBAC) model. Roles are assigned to users, and permissions are granted to roles. Snowflake also supports a least-privilege security model, where users are granted only the permissions they need to perform their tasks. Snowflake also supports a separation of duties (SoD) model, where different users are granted different permissions to ensure that no single user has too much power. Snowflake also supports a data loss prevention (DLP) model, where sensitive data is protected by encryption and access controls.

- A. SELECT COUNT() FROM ORDERS WHERE SUBSTRING(ORDER_DATE, 6, 2) = '01' AND SUBSTRING(ORDER_DATE, 1, 4) = '2023';

- B. `SELECT COUNT( ) FROM ORDERS WHERE ORDER_DATE LIKE '2023-01%';`
- C. `SELECT COUNT( ) FROM ORDERS WHERE TO_DATE(ORDER_DATE) BETWEEN '2023-01-01' AND '2023-01-31';`
- D. `SELECT COUNT( ) FROM ORDERS WHERE MONTH(TO_TIMESTAMP(ORDER_DATE)) = 1 AND YEAR(TO_TIMESTAMP(ORDER_DATE)) = 2023;`
- E. `SELECT COUNT( ) FROM ORDERS WHERE DATE(CONVERT_TIMEZONE('UTC', SESSION_TIMEZONE(), TO_TIMESTAMP(ORDER_DATE))) BETWEEN '2023-01-01' AND '2023-01-31';`

Answer: E ([LEAVE A REPLY](#))

□□ E□ VARCHAR□ TIMESTAMP□ □□□□□ □□□□, UTC□ □□ □□□□ □□□ □□, □□□ □□□□ □□ DATE □□□ □□□□□ □□ □□□□□□. □□ □□ □□□ □□□ □□ □□ □□□ □□□ □□□□□. □□ A□ B□ □□□ □□□ □□□□□, □□ □□ □□ □□□□ □□ □□ □□□□ □□□□ □□□□. □□ C□ □□□□ □□□□ □□ □□□ □□□ □□□□□. □□ D□ MONTH() □ YEAR() □□□ □□□□ □□ □□□ □□□ □□□□ □□ □□□ □□□□ □□□□□.

NEW QUESTION: 45

□□□ □□□□ □□□ □□ □ □□ □□□ □ □□□ □□□□ □□□□ □□□□□.



□□□ □□□□ □□□ □□□□□?

- A. □□ □□□ ACCOUNTADMIN □□□ □□ □□□□ □□□ □ □□□□.
- B. ACCOUNT_USAGE □□ INFORMATION_SCHEMA□ □□□□ □□□.
- C. DB1□ ACCOUNT_USAGE□ □□ SELECT □□□ □□□ □□□ □□□.
- D. □□ □□□□□□ □□□□□ SNOWFLAKE□ □□□□ □□□.

Answer: D ([LEAVE A REPLY](#))

□ □□□ Snowflake□ □□ □□□ □□□□ □□□ □□ Snowflake □□□□□□□ □□□ □□ □□□□ □□□□□. Snowflake□□ ACCOUNT_USAGE □□□□ □□□ □□□, □□ □□, □□ □□□ □□□ □□ □□□ □□ □□□□ □□ □□□□ □□□□ □ □□□□□. □□□ □□ SNOWFLAKE□□ □□□ □□ □□□□□□□ □□□□□. □□□ □□□□ □□□□ □□□ □□□ □□□ □, □□ □□□ ACCOUNT_USAGE□ □□□□ □□□ DB1 □□□□□□ □□□□□ PUBLIC □□□ □□□□ □□□ □□□□ □□ □.

WAREHOUSE_METERING_HISTORY□ □□□□ □□□ □□□ Snowflake □□□□□ □□ □□□□□□ □□□□□□ □□□ □□□□□ □□□□□. □□□□□ DB1 □□□ ACCOUNT_USAGE□□ □□□□ □□□□. □□ □□□□□ □□□□ □□□□□ □□□ 'DB1.ACCOUNT_USAGE'□ □□□□ □□□□.

□ □□□ □□□□ □□ □□□□ □ □□ □□□ □□□ □ □□□□.

* □□ □□□ □□□ □□□□□: □□□ □□□□ □□ □□□ □□□□□ □□□: FROM SNOWFLAKE.

ACCOUNT_USAGE.WAREHOUSE_METERING_HISTORY. □□ □□□□ □□ □□ □□□□□ □□□□ □□□ □□□ □ □□□ □□□□ □□ □□□□ □□ □□□□□.

* □□□□ □□: □□□ □□□□ □□ 'USE DATABASE SNOWFLAKE;' □□□ □□□□□□.

□□□□ □□ □□□ □□□□□□ □□□□□ □□□□□ ACCOUNT_USAGE□ □□ □ □□ □□□.

(ACCOUNTADMIN) ACCOUNTADMIN 000 00 000 0 000 00 00 000000. 000000 000 00000 000, ACCOUNTADMIN 000 00000 00 00 00 0 SNOWFLAKE 00000000 00 00000 0000 000 0 00000. 0000 00000 0000 00 00 00 00(000000 "00 00"0000 0000)0 0000 00000000 00 000000. 0000 0000000 0000000 0000 0000 00000 00 0 00 0000 000000 000000 000000.

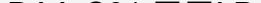
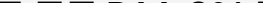
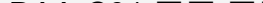
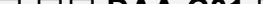
NEW QUESTION: 46

[illegible]

- A.** `SELECT STAGE_NAME FROM STAGE_HISTORY ORDER BY STAGE_NAME;`
- B.** `SELECT STAGE_NAME FROM STAGE_HISTORY ORDER BY STAGE_NAME, STAGE_HISTORY_ID;`
- C.** `TIMESTAMPDIFF('SECOND', start_timestamp, end_timestamp) AS duration FROM STAGE_HISTORY;`
- D.** `SUM(STAGE_HISTORY_ID) FROM STAGE_HISTORY;`
- E.** `SELECT STAGE_NAME FROM STAGE_HISTORY WHERE start_timestamp < end_timestamp;`

Answer: C (LEAVE A REPLY)

00 00(C) 0 000000 00 00000 0000 000000. 0 000000 00 0 0 0000 00 0 0000 000000 000000 00 0000 000000 00 00000. 0000 00 0000 0 0000 00000 00 'TIMESTAMPDIFF()' 0000 00000 0 0000 00 0000 00000 0000. 00 0000 00 000(A, D) 00 00(B) 0 000000, 0 000000 00 0 00 00 0000 0000 000000 000000 00000. 00000 0 0000 0000 00000 00000.

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**)

NEW QUESTION: 47

[illegible]

- A.** □□□□□ □□□□ □□□□ □□ □□ □□□, □□□□ □□ □□ □□□ □□□ □□□□ □ □□□, □□□□ □□□□ □□□ □□□□ □□ □□□.
- B.** □□□□ □ □□□ □□□□ □□□□ □□□, □□ □□□ □□□ □□□□ □□□ □□, □□□ □ □□□□□□ □□ □□□□ □□□□ □□□ □□□□ □□□.
- C.** □ □□□ □□□□ □□□□ □□□□ □□□, □□□□ □□□□ □□□□ □□ □□□, □□□□ □□ □□ □□ □□□□ □□□□ □□□ □□.
- D.** □□□□ □□□□ □□□□ □□□□ □□□□ □□□□ □□□ □□, □□ □□□ □□□ □□□□ □□□ □□, □□ □□□□□ □□□□ □□□ □ □□ □□□.
- E.** □ □□ □ □□□□□□ □□ □□ □□ □□□ □□□□ □□ □□, □□ □□ □□□ □□□□ □□□ □□, □□□□□ □□□ □□□□ □□□□ □□□ □□□□□.

Answer: B (LEAVE A REPLY)

[illegible]

NEW QUESTION: 48

□□ □□□□ □□ □□ □□□ □□□□ □□ □□ □□□□ □□ □□□ □□□ □□□□□?

- A.** □□ □□ □□□□ □□□□ □□□ □□□ □□□
B. □□□□ □□□□□□ □□□ □□□
C. □□ □□□ □□□ □□ □□ □□
D. □□ □□□ □□□□ □□□□□.

Answer: C (LEAVE A REPLY)

□□ □□□□ □□ □□ □□□ □□□□ □□ □□ □□□□ □□ □□ □□□ □□□□ □ □□ □□□□□.

NEW QUESTION: 49

□□□ □□□□ SNOWFLAKE.ML.FORECAST□ □□□□ modelX□□ □□□ □□□□□□. □□□□ □□ □ □□ □□ □□□□ □ □□□ tableX□ □□ □□□□ □□□. modelX:
FORECAST □□□ □□□ □ □□□□ □□ □□□ □□□□ □□□?

- A.** □□ □□ □□□ tableX□ □□ □□□□□.
- B.** □ □□□□ □□ □□□ □□□□□.
- C.** RESULT_SCAN□ □□□□ □□□□ □□□□□.
- D.** □□ □□□ □□□ □□ RESULT_SCAN□ □□□ □□□□□ tableX□ □□□□□.

Answer: C (LEAVE A REPLY)

Snowflake Cortex ML □□(□: FORECAST)□ □□□□ □□□ □□(□: CALL modelX!FORECAST(...))□ □□□□ □□□ □ □ □□□ □□ □□□ □□□□□. □ □□□ Snowsight
□□ □□□ □ □ □□□, CALL □ □□□ □□ INSERT INTO □□ CREATE TABLE AS SELECT(CTAS) □ □□□ □□ □□□ □□ □□□ □ □□□□.

□□ □□ □□□ □□ □□□(tableX)□ □□□□□ □□□ □□□□ RESULT_SCAN □□□ □□□ □□□□ □□□. Snowflake□ □□ □□ □ □□ □□ □□□ 24□□ □□ □□ □
□□ □□□□□. RESULT_SCAN □□□ □□□□ □ □□□ □□ □□□ □□□□□ □□□ □ □□□□.

□□ □□□□□ □□□ □□□□.

```
* □□ □□: CALL modelX!FORECAST(FORECASTING_PERIODS => 12);
```

* □□ □□□ LAST_QUERY_ID() □□□ □□□□ □□ □□□ □□□ □□□ □□□□□.

```
* □□ □□□ □□□□ □□□□ □□□□: CREATE TABLE tableX AS SELECT * FROM TABLE (RESULT_SCAN(LAST_QUERY_ID())); □□ □□:
```

* □□ A□ CALL □□□ □□□ □□□ □□ □□□□ INTO □□ □□□□ □□□□ □□□□ □□□□.

* □□ B□ □□□□ □□□ □□□□ □□ □□ □□ □□ □□□ □□□□ □□ □□ □□□ □□□ □□□□ □□□□□□□□.

* □□ D□ □□□□ □□□□ □□□ □□("□□ □□□ □□")□ □□□□ □□□□.

* C 100% . Snowflake Data Cloud Snowflake " " .

NEW QUESTION: 50

Snowflake□□ □□□□ □□□ □□ □□□ □□□ □□ □□□□ □□ □□□ □□□□?

- A.** □□□□□ □□ □□□ □□□□□.
- B.** □□ □□□ □□□□□ □□ □□ □□ □□□ □□□□.
- C.** □□ □□□ □□□ □□ □□□ □□□ □□□□□.
- D.** □□ □□□ □□□□□ □□ □□ □□ □□□ □□□□□□.

Answer: B (LEAVE A REPLY)

□□□□ □□□□ □□□ Snowflake□□ □□□ □□ □ □□ □□□□ □□□□□ □□ □□□ □□□□□□.

NEW QUESTION: 51

Which of the following is a valid Snowflake SQL query to calculate the total number of unique users who have purchased a product, grouped by product ID (CPA)?

- A. `SELECT CPA, COUNT(DISTINCT user_id) FROM purchases GROUP BY CPA;`
- B. `SELECT CPA, COUNT(DISTINCT user_id) FROM purchases;`
- C. `SELECT CPA, COUNT(DISTINCT user_id) FROM purchases GROUP BY CPA, user_id;`
- D. `SELECT CPA, COUNT(DISTINCT user_id) FROM purchases;`
- E. `SELECT CPA, COUNT(DISTINCT user_id) FROM purchases GROUP BY CPA;`

Answer: (SHOW ANSWER)

The correct answer is A. The query `SELECT CPA, COUNT(DISTINCT user_id) FROM purchases GROUP BY CPA;` calculates the total number of unique users who have purchased a product, grouped by product ID (CPA). The other options are incorrect because they either do not group by CPA or do not use the `DISTINCT` keyword to count unique users.

NEW QUESTION: 52

Snowsight is a tool that allows you to view and analyze data from Snowflake. Which of the following is a valid Snowsight query to calculate the total number of unique users who have purchased a product, grouped by product ID (CPA)?

- A. `SELECT CPA, COUNT(DISTINCT user_id) FROM purchases GROUP BY CPA;`
- B. `SELECT CPA, COUNT(DISTINCT user_id) FROM purchases;`
- C. `Snowsight SELECT CPA, COUNT(DISTINCT user_id) FROM purchases GROUP BY CPA;`
- D. `SELECT CPA, COUNT(DISTINCT user_id) FROM purchases;`

Answer: D (LEAVE A REPLY)

The correct answer is D. The query `SELECT CPA, COUNT(DISTINCT user_id) FROM purchases;` calculates the total number of unique users who have purchased a product, grouped by product ID (CPA). The other options are incorrect because they either do not group by CPA or do not use the `DISTINCT` keyword to count unique users.

NEW QUESTION: 53

'website traffic' table in Snowflake. Which of the following is a valid Snowflake SQL query to calculate the total number of unique users who have purchased a product, grouped by product ID (CPA)?

- A. `SELECT DATE(timestamp), COUNT(DISTINCT user_id), device_type FROM website_traffic GROUP BY DATE(timestamp), device_type;`
- B. `SELECT DATE(timestamp), APPROX_COUNT_DISTINCT(user_id), device_type FROM website_traffic GROUP BY DATE(timestamp), device_type;`
- C. `SELECT DATE(timestamp), COUNT(user_id), device_type FROM website_traffic GROUP BY DATE(timestamp), device_type;`
- D. `SELECT DATE(timestamp), COUNT(DISTINCT user_id) as dau, device_type, COUNT( ) OVER (PARTITION BY DATE(timestamp)) as total_users FROM website_traffic GROUP BY DATE(timestamp), device_type, user_id;`
- E. `SELECT DATE(timestamp), COUNT(DISTINCT user_id) as dau, device_type FROM website_traffic GROUP BY DATE(timestamp), device_type ORDER BY DATE(timestamp), dau DESC;`

Answer: E (LEAVE A REPLY)

The correct answer is E. The query `SELECT DATE(timestamp), COUNT(DISTINCT user_id) as dau, device_type FROM website_traffic GROUP BY DATE(timestamp), device_type ORDER BY DATE(timestamp), dau DESC;` calculates the total number of unique users who have purchased a product, grouped by product ID (CPA). The other options are incorrect because they either do not group by CPA or do not use the `DISTINCT` keyword to count unique users.

NEW QUESTION: 54

Which of the following is a valid Snowflake SQL query to calculate the total number of unique users who have purchased a product, grouped by product ID (CPA)?

- A. `SELECT CPA, COUNT(DISTINCT user_id) FROM purchases GROUP BY CPA;`

- D.** ☐☐ ☐☐ ☐☐☐☐ ☐☐ ☐☐ ☐☐ ☐☐☐☐.

Answer: A (LEAVE A REPLY)

□□□□ □□ □□□□ □□□□ □□□□ □□□□ □□□ □□ □□ □□□□.

NEW QUESTION: 55

Snowsight□□ □□□□ □□□ □□ □ □□□ □□ □□□ □□ □□□ □□□□ □□ □□□ □□□ □□□□□?

- A.** Snowsight □ □□ □□ □□□ □□□□.
B. □□ □□□ □□□ □□□□ □□□□ □□ □□□□.
C. Snowsight □ □□□ □□ □□□ □□□ □□□ □□□
D. □□□ □□ □□□ □□□ □□□ □□ □□□ □□□□□.

Answer: ([SHOW ANSWER](#))

□□□ □□ □□□ □□□ □□ □□ □□□□ Snowsight□□ □□ □ □□□ □□□□.

NEW QUESTION: 56

□□□ □□□ □ □□□ □□□ □□ □□□□ □□□ □□□□□? (□□□□ □□ □□□ □□□□□)

- A.** □□ □□□□ □□ □ □□ □□ □□
B. □□ □□□ □□□ □□ □□□ □□ □□
C. □□□ □□ □□ □□ □□□ □□
D. □□ □□□ □□□□ □□ □□□ □□ □□□□ □□□□ □

Answer: ([SHOW ANSWER](#))

□□□ □□□ □□□□ □□□ □□□ □□□□ □□/□□ □□□ □□□□ □□ □□ □□□□□.

NEW QUESTION: 57

[illegible]

	DEPT	JAN	FEB	MAR	APRIL
1	electronics	100	200	300	100
2	clothes	100	300	150	200
3	cars	200	400	100	50

□□□ □□□ □□□□ □ □□ □□□□ □□ □□□□□?

	DEPT	MONTH	SALES
1	electronics	JAN	100
2	electronics	FEB	200
3	electronics	MAR	300
4	electronics	APRIL	100
5	clothes	JAN	100
6	clothes	FEB	300
7	clothes	MAR	150
8	clothes	APRIL	200
9	cars	JAN	200
10	cars	FEB	400
11	cars	MAR	100
12	cars	APRIL	50

- A. ☐☐

- A.** ☐☐☐ ☐☐☐ ☐☐☐☐ ☐☐

- B. Snowflake uses a columnar storage format.
- C. Snowflake uses a row-oriented storage format.
- D. Snowflake uses a hybrid storage format.

Answer: (SHOW ANSWER)

Snowflake uses a columnar storage format. This format allows for efficient storage and retrieval of data, particularly for analytical workloads. Snowflake's columnar storage is implemented using a compressed columnar format called Columnar Store.

NEW QUESTION: 61

A Snowflake user wants to create a table named CUSTOMERS with columns for ID, NAME, and EMAIL_ADDRESS. The user wants to ensure that the EMAIL_ADDRESS column is encrypted using a secure algorithm. Which of the following SQL statements would correctly create the table and encrypt the EMAIL_ADDRESS column?

- A. CREATE TABLE CUSTOMERS (ID INT, NAME VARCHAR(50), EMAIL_ADDRESS VARCHAR(100) ENCRYPTED);
- B. CREATE TABLE CUSTOMERS (ID INT, NAME VARCHAR(50), EMAIL_ADDRESS VARCHAR(100) SECURE);
- C. IS_ROLE_IN_SESSION() returns true if the current user is an Analyst role, false otherwise. The user wants to ensure that the EMAIL_ADDRESS column is encrypted using a secure algorithm. Which of the following SQL statements would correctly create the table and encrypt the EMAIL_ADDRESS column?
- D. 'MD5()' is a secure algorithm for encrypting data. The user wants to ensure that the EMAIL_ADDRESS column is encrypted using a secure algorithm. Which of the following SQL statements would correctly create the table and encrypt the EMAIL_ADDRESS column?
- E. 'REGEXP_REPLACE()' is a secure algorithm for encrypting data. The user wants to ensure that the EMAIL_ADDRESS column is encrypted using a secure algorithm. Which of the following SQL statements would correctly create the table and encrypt the EMAIL_ADDRESS column?

Answer: (SHOW ANSWER)

The correct answer is A. The SQL statement correctly creates the table CUSTOMERS with columns ID, NAME, and EMAIL_ADDRESS. The EMAIL_ADDRESS column is encrypted using the ENCRYPTED keyword. The other options are incorrect. Option B uses the SECURE keyword, which is not a valid Snowflake keyword. Option C is a function call, not a SQL statement. Option D uses the MD5() function, which is not a secure algorithm for encrypting data. Option E uses the REGEXP_REPLACE() function, which is not a secure algorithm for encrypting data.

DAA-C01 is a collection of Snowflake dumps. DumpTop is a tool that can be used to download the dumps. DAA-C01 is a collection of Snowflake dumps. DumpTop is a tool that can be used to download the dumps. **DAA-C01** is a collection of Snowflake dumps. DumpTop is a tool that can be used to download the dumps. <https://www.dumptop.com/Snowflake/DAA-C01-dump.html> (67 Q&As Dumps, **30%OFF Special Discount: KrDump**)

NEW QUESTION: 62

A Snowflake user wants to create a table named CUSTOMERS with columns for ID, NAME, and EMAIL_ADDRESS. The user wants to ensure that the EMAIL_ADDRESS column is encrypted using a secure algorithm. Which of the following SQL statements would correctly create the table and encrypt the EMAIL_ADDRESS column?

- A. WHERE clause is used to filter rows based on a condition. GROUP BY clause is used to group rows based on one or more columns.
- B. SELECT clause is used to select rows from a table. GROUP BY clause is used to group rows based on one or more columns.
- C. WHERE clause is used to filter rows based on a condition. SELECT clause is used to select rows from a table.
- D. WHERE clause is used to filter rows based on a condition. GROUP BY clause is used to group rows based on one or more columns.

Answer: (SHOW ANSWER)

The correct answer is A. The SQL statement correctly creates the table CUSTOMERS with columns ID, NAME, and EMAIL_ADDRESS. The EMAIL_ADDRESS column is encrypted using the ENCRYPTED keyword. The other options are incorrect. Option B uses the SECURE keyword, which is not a valid Snowflake keyword. Option C is a function call, not a SQL statement. Option D uses the MD5() function, which is not a secure algorithm for encrypting data.

The user wants to ensure that the EMAIL_ADDRESS column is encrypted using a secure algorithm. Which of the following SQL statements would correctly create the table and encrypt the EMAIL_ADDRESS column? The user wants to ensure that the EMAIL_ADDRESS column is encrypted using a secure algorithm. Which of the following SQL statements would correctly create the table and encrypt the EMAIL_ADDRESS column?

```

`datebucket` 0000 000 000 000000. 0 0000 0000 000 00 0000 000 0000 0000 0000 000 0 0000. 00 00 000 "00" 000
0 "00" 000 000 0 0000. 00 00 `datebucket` SELECT 000 00 00 00 00 0000, 000 00 0000 0000 000000 `GROUP BY` 00 000
000000 000 0000(0: `GROUP BY 1` 00 `GROUP BY :datebucket`).

```

□ □ □ □ □ □ :

```
* A .:datebucket` WHERE  000 00000 0000 0000 0000/000 00 0000.
```

```
* B000000000.`:daterange`000 0000 0000, SELECT 000 000 000 00 000 000000.
```

* □□ C□ A □ B□ □□□ □□□ □□□□□□□□.

```
* □□ D□ 100% □□□□□. □□ □□ Snowsight □□□ □□□ □□□□. `:daterange`□ WHERE □□ □□□ □ □□□ □□□□, `:datebucket`□ GROUP BY □□ □□□ □□ □
□□ □□□□□.
```

NEW QUESTION: 63

□□□ □□□□ Amazon S3 □□□□ □□□ Parquet □□□ □□□□□□. □□□□□ Parquet □□□ □□□□ □□ □□□□ □ □□ □□□□ □□□ □□□□□?



B)

C)

D)

A. ☐ ☐ A

B. ☐ ☐ B

C. ☐ ☐ C

D. ☐ ☐ D

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

□□, □□ □ □□ □□□□ □□ □□□ □□ VARIANT □□ □□□□, SQL□□□□ □□ □□□□ \$1□ □□□□ □□□□□.

□□ □□□□ □□□□ □□(□□ A□ □□□□ □□)□ Parquet □□□ \$1, \$2, \$3□ □□ □□ □□ □□□□ □□ □□ □□□□ □□□□ □□□□.

Parquet □□□ □□ □ \$2 □□□ □□ □□□□ \$1□ □□ □□ □□□□ □□ □□□ NULL□ □□□□□.

□□□ □□□□ □□□□□ "□□"□□□ □□□□□ □□□ □□ □□ □□□ □□□□□ □□□□ □□ □□□□ □□□□ □□□. □□ □□ □□(\$1)□ □□□ □□ □□
(:)□ □□□ □□ □□ □□ □□□□ □□□□□(□: \$1:o_custkey). □□, Variant□□ □□□ □□ □□□□□ □□□ Variant □□□□□ □□ □□□□ □□□ □□□ □□□□ □□□
□□ □□ □□ □□(□: ::number, ::date)□ □□□□ □□□□□ □□□ □□□ □□□□ □□□□□ □□□□ □□□.

□□□ □□□□:

- * □□ A□ □□ □□(\$2, \$3 □)□ □□□□ □□□ □□□□□□□□. □□ □□□ CSV□ □□ □□□□ □□□□□ □□□□□.
- * □□ B□ □□ □□ □□(\$1)□ □□ □□ □□ □□ □□ □□ □□□□□ □□□□ □□□ □□□□□□□□.
- * □□ D□ Snowflake SQL□□ □□ □□□ □□□□ □□ □□□ parse() □□□ □□□□□ □□□□ □□□□.
- * □□ C□ 100% □□□ □□□□□. Parquet □□□□ \$1□ □□□□ □□□ □□□□ □□□□, □□□ □□□□ □□ □□ □□□□, □□□ □□□□ □□□□□. □ □□ □□□ "□
□□ □□ □ □□"□□□□ □□, □□ SnowPro □□ □□□ □□□□ □□ □□□□□.

NEW QUESTION: 64

□□□ □□ □□□□ 'customer_id'□ □□□□ □□□□□□ □□□□□□□ □□□□. □□□□□ 'customer_id'□ 'transaction_amount' □□ □□□□. □□ □□□
'transaction_date'□ □□□□ □□□□ □ 'customer_id'□ □□□□ □□□ □□□□□ □□□. □□ □□□□ □ □□□□□□ □□□ □, □□ □□ □ □□ □□□ □□□ □□□□ □
□□□□□□ □□□ □□ □□□ □□ □□□□□ □□□□ □ □□□□?

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

B. 'customer_id'□ 'transaction_date'□ □□□□ 'transaction_amount'□ □□ □□□□ □□□□ □□ □□□□□.

C. 'transaction_date' □□ □□ □□□□ □□□□□.

D. 'ALTER TABLE fact_transactions RECLUSTER;' □□□□ □□□□ □□□□ □□ □□□□□□□□□□.

E. 'customer_id'□ □□□□ □□□ □ □□□□□□ □ □□□□ □□□□ □□□□ □□□□ □□□□□□□□□□. □□ □□□□ □□□□□.

Answer: B (LEAVE A REPLY)

□□ B□ □□□ □□□□□□. 'transaction_date'□ □□□□ □□□□ □□ □□□□ □□□□ □□ □□□□ □ □□ □□□ □□ □□□ □ □□□□. □□□□ □□ □□□ □□□□ □
□□□□ □□□ □□□□□ □□□□□ □□□□□. 'transaction_date'□ □□□□□ □□□ □□ □□ □ □□ □□ □□□ □□□□□ □□□ □ □□□□. □ □□□(□□) □□ □□
□□ 'customer_id'□ □□□□ □□□□□□□□□ □□□ □□□ □□□□□□□. □□□□ □□ □□□ □□□ □□□□ □□ A□ □□□ □□ □□□ □□□□ □□ □ □□□□. □□ C□
□□ □□□ □□ □□□□□□□, □□□□ □□ □□□□ □□□□□ □□□□ □□□ □□□□□ □□□□□ □□□□ □□□□ □□□□. □□ D□ □□ □□ □□ □□□□□□□□□ □□□ □□ □
□ □□ □□□ □□ □□□ □□□ □□ □ □□□□. □□ E□ □□□ □□ □□ □□□ □□ □□□ □□□ □□□□□□□ □□□ □ □□□□. □□□ □□□□ □□ □□□□ □□
□□ □□□□.

NEW QUESTION: 65

Snowflake□□ □□ □□□□ □□□□ □□□ □□□□□□ □□□□ □□□□. □□ □□□ □□□ □□ □□ □□□□ □□ 'RAW ORDERS' □□□□ JSON □□□□ □□□□ □□
□□. □□□□ □□ □□□□ □□ □□ □□□□ □□□□□ □□□□ 'CLEANED ORDERS' □□□□ □□□□ □□ □□□ □□□ □□□ □□□□□ □□□. □□, □□□ 'ORDER
LINEAGE' □□□□□ □□□ □□(□□ □□ □□□ □□ □□□ □□□□ □□□□□)□ □□□□□ □□□. □□ □ □□□ □□ □□□ □□□□□ □□□□□ □□□□ □□ □□□
□□□□□? (□ □□□ □□□□□□)

A. 'RAW_ORDERS' □□□□ 'email' □□ □□□ □□□ □□□□□. □□ □□ 'RAW_ORDERS' □□□□ □□ □□□□ 'CLEANED_ORDERS' □□□□ □□□□ □□□ □□□□□.

B. 'RAW ORDERS' □□□□□ □□□□ □□ SHA256 □□□ □□□□ □□□ □□□ □□□□□, □□□ □□□□ 'CLEANED ORDERS' □□□□ □□□□ □□□, □□ □□ □□□
'ORDER LINEAGE' □□□□ □□□□ □□ □□□□□ □□□□□□□. □ □□ □□□□□ □□□□□ □□□□□ □□□□□□.

C. Snowpipe□ □□□□ 'RAW ORDERS' □□□□ □□□□ □□□□□. 'RAW ORDERS' □□□□ □□□□ □□□□□. □□ □□□□ □□□ □□□□ □□□□ □□□□□ □ □□
□□ □□, □□□□ □□□□□, □□□ □□□□ 'CLEANED_ORDERS' □□□□ □□□□, □□ □□□ 'ORDER_LINEAGE' □□□□ □□□□□.

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

E. 'RAW ORDERS' □□□□ □□□ □□□ □□□□□ □□ □□□□□. Snowpipe□ □□□□ □□□□ □□ □□ □□□□ 'RAW ORDERS' □□□□ □□□□□□. □□□□ □□□ □□□□ □□□□ □□ 'CLEANED_ORDERS' □□□□ □□□□, 'ORDER_LINEAGE' □□□□ □□ □□ □□□ □□□□.

Answer: ([SHOW ANSWER](#))

B: The following table lists the tables in the database. The table 'CLEANED ORDERS' is the main table, and 'ORDER LINEAGE' is a table that contains the lineage of the 'CLEANED ORDERS' table. C: Snowpipe is a feature that allows you to load data from a cloud storage into a table. RAW_ORDERS is a table that contains the raw data from the cloud storage. A: The table 'CLEANED_ORDERS' is a table that contains the cleaned data from the cloud storage. D: The table 'ORDER LINEAGE' is a table that contains the lineage of the 'CLEANED_ORDERS' table. E: The table 'RAW_ORDERS' is a table that contains the raw data from the cloud storage.

NEW QUESTION: 66

□□□ □□ □□□ □□□□ □ □□□□ □□□ □□□□□?

- A.** □□ □□□ □□□□ □□ □□ □□□□ □□□□ □
B. □□ □□□ □□□□ □□ □ □□ □□□ □□□□□.
C. □□ □□□ □□□□ □□□□ □□□□□.
D. □□ □□ □□ □□□ □□□□ □□

Answer: B (LEAVE A REPLY)

□□ □□□ □□ □□□ □□□ □□ □□□ □□ □□□ □□□ □□ □□□ □□□□ □□ □□□ □□□□ □□ □□□□□.

DAA-C01 00 000 000000 00 DumpTop 00 0000 000 DAA-C01 00! DumpTop 0 00 **DAA-C01** 00 000 000000, DumpTop DAA-C01 00 000 0 0000000 000 00000000. 0000 000 0000 00 DumpTop DAA-C01 000 000000. <https://www.dumptop.com/Snowflake/DAA-C01-dump.html> (67 Q&As Dumps, **30%OFF Special Discount: KrDump**)