

Databricks.Databricks-Certified-Professional-Data-Engineer.v2026-10-04.q201

□□□□:	Databricks-Certified-Professional-Data-Engineer
□□□□:	Databricks Certified Professional Data Engineer Exam
□□□:	Databricks
□□ □□ □□□:	201
□□:	v2026-10-04
# □□ □:	108
# □□ □□□:	2010
https://www.krdump.com/Databricks.Databricks-Certified-Professional-Data-Engineer.v2026-10-04.q201.html	

NEW QUESTION: 1

□□□ □□□□□ □□ Databricks Unity Catalog□□ □□□ □□□ □□□□ □□□□. □□□ analyst_group□ sales □□□□□ □□ SELECT □□□ □□□□□, □ □□□ □□□□ □□□□ □□ □ □□ □ □□ □□□, □□□, □□ □□ SELECT □□□ □□□ □□□□ □□ □ □□□ □□□□□□.

Unity Catalog□□ □□ □□ □□□ □□□ □□□□□?

- A. □□□□□ SELECT □□□ □□□□ □□ □□□□ □□ □□ □ □□ □□ □□□, □□□, □□ SELECT □□□ □□□□ □□□□□.
- B. Unity □□□□□ □□□ □□□□□ □□□□ □□□□. □□□□ □□□□ □□□ □□□□□□□ □ □□□ □ □□□□ □□ SELECT □□□ □□□□□ □□□□ □□□.
- C. □□□□ □□□□ SELECT □□□ □□□□ □□ □□□ □ □□□□□ □□□□□ □□ □□□ □□□ □ □□□□□ □□□□ □□□□.
- D. □□□ □□□□ □□□ □□□ □□□□ □□□ □□□□ □□□□ □□□□ □□□□ □□ □ □□□ □□□□□.

Answer: (SHOW ANSWER)

Databricks □□□ □□□□ □□□□ □□□ □□□ □□□□ □ □□□□□ □□□ □□:

Unity Catalog□□ □□□ □□□□□ □□□□ □□□□. □, □□□□□ □□ □□(□: SELECT)□ □□□□□ □□ □□□□□ □□□ □□(□□□, □□□ □□ □)□ □□□ □□□ □□□□ □□□□ □ □□□□. □ □□ □□□ □□□□ □□□□ □□ □□ □□ □□□ □□□ □□□□.

Databricks□ □□ □□ □□□ □□□ "□□□□ □□□□ □□□□ □□□ □□ □□□ □□□□ □□□□ □□□□ □□□□." □□□□ □□ □□□ □□□□ □□ □□ □□□□ □ □□□□ □□□□□ □□□□ □□□□ □□□. □□□□ □□□ □□ □□□ □□□ □□ □□ □□ □□□ □□□□□. □□□□ □□ B□ Unity Catalog□ □□ □□□ □□□□ □□□□ □□, A□ D□ □□ □□□ □□ □□□□ □.

NEW QUESTION: 2

□□□ □□□□ □□□ □□ □□□ □□ □□□□□ □□□□□□.

□□ □□ □□□□ □ □□□ □□□□ □□ SQL DDL □□□□ □□□□ □□ □□□ □□ □ □□ □□□□?

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

- A. 1. □□□ table_name□ □□ □□ □□□□□.
- 2. DELTA□ □□□□ (id □□□□, birthDate □□, avgRating □□) □□ □□□□□.
- B. 1. CREATE TABLE IF NOT EXISTS table_name (id STRING, birthDate DATE, avgRating FLOAT)
- C. 1. CREATE OR REPLACE TABLE table_name AS
- 2. id(□□□), birthDate(□□), avgRating(□□)□ □□ □□□□ □□□□ □□□□□.
- D. 1. □□□_□□□ □□□ □□ □□□□□.
- 2. id (□□□), birthDate (□□), avgRating (□□)□ □□□□□.

E. 1. CREATE OR REPLACE TABLE table_name (id STRING, birthDate DATE, avgRating FLOAT)

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

NEW QUESTION: 3

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

```
Cmd 1
rawDF = spark.table("raw_data")

Cmd 2
rawDF.printSchema()

Cmd 3
flattenedDF = rawDF.select("...", "values.*")

Cmd 4
finalDF = flattenedDF.drop("values")

Cmd 5
finalDF.explain()

Cmd 6
display(finalDF)

Cmd 7
finalDF.write.mode("append").saveAsTable("flat_data")
```

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

- A. □□ 2
- B. □□ 3
- C. □□ 4
- D. □□ 5
- E. □□ 6

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

□□ 6□ □□□□ □□□□ □□□□ □□ □□□□ □□ □□□□□□. □ □□□ finalDF □□□□□□□□ □□ □□ □□□□ □□□□ □□□□□□. finalDF □□□□□□□□ □□ □□ 7□□ □□□□ □ □□□□□□ □ □□□ □□□□ □□□□. □□□□ □□□□□□□□ □□□□ □□ □□□□ □□□ □□□ □ □□ □□□□ □□□ □□□ □□□□.

□□□ Cmd 6□ □□□□□□ □□□□ □□□.

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

* Cmd 1: raw_data □□□□ rawDF□□ Spark □□□□□□□□ □□□□□□□.

* Cmd 2: rawDF □□□□□□□□ □□□□ □□□□□□. □□ □□□ □ □□□ □□ □□□ □□□□□.

* Cmd 3: rawDF □□□□□□□□ □□ □□ values □□□□ □□ □□□ □□ □□ □□□□ flattenedDF□□ □ □□□□□□□□ □□□□□.

* Cmd 4: flattenedDF □□□□□□□□ □ □□ □□□□ □□ values □□ □□□□ finalDF□□ □ □□□□□□□ □□□□□.

* Cmd 5: finalDF □□□□□□□ □□□ □□ □□□ □□□□□□. □□ □□ □□□ □□□□□ □□□□ □ □□□□□.

* Cmd 7: finalDF □□□□□□□ flat_data□□ □□□□ □□□□. □□ □□□□ □ □□□□ □□□□ append □□□ □□□□□□.

NEW QUESTION: 4

□□ SQL □ □ □□ □□□□ □□ □□ □□□□□ □□□□ □□□□ □ □□□ □ □□ □□ □□□□□?

- A. SELECT DISTINCT * FROM table_name
- B. SELECT DISTINCT * FROM table_name HAVING COUNT(*) > 1
- C. SELECT DISTINCT_ROWS (*) FROM table_name
- D. SELECT * FROM table_name GROUP BY * HAVING COUNT(*) < 1
- E. SELECT * FROM table_name GROUP BY * HAVING COUNT(*) > 1

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

□□

□□□ SELECT DISTINCT * FROM table_name□□□.

NEW QUESTION: 5

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

- A. DROP TABLE table_name CASCADE
- B. □□□_□□ □□
- C. DROP TABLE table_name □□□□ □□□□, dbutils.fs.rm("/path",True) □□□□ □□□□ □□□ □□□□ □□□□□□.
- D. DROP TABLE table_name INCLUDE_FILES
- E. □□□□ □□□□ VACUUM □□□ □□□□□.

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

□□

□□□ DROP TABLE table_name□□□.

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

NEW QUESTION: 6

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

- A. 1. □□□ □□ = "□□□"
- 2.schema_name = "bronze"
- 3.query = f"select * from schema_name.table_name"
- B. 1. □□□ □□ = "□□□"
- 2.schema_name = "bronze"
- 3. □□ = "select * from {schema_name}.{table_name}"
- C. 1.table_name = "sales"
- 2.schema_name = "bronze"
- 3.query = f"select * from { schema_name}.{table_name}"
- D. 1.table_name = "sales"
- 2.schema_name = "bronze"
- 3.query = f"select * from " + schema_name + "." + table_name"

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

□□

□□□

□□□ □□ = "□□"

query = f"select * from {schema_name}.{table_name}"

`f` □□□□ □□□ □□□ □□□□ □ □□□ □ □□□□. `f`□□□□ □□□ {□□□□ □□}□□□□."

<https://realpython.com/python-f-strings/>

NEW QUESTION: 7

□□□□ `department`□□ □□□ □□□ □□□□ □□ □□□□□ □□□□. □ □□□□□ □□□□□□, □ □□□□□ □□□□ □□□ □□ □□□ □□□□□ □□□. `department` □□□ □□□□□□ □□□□, □□□□ □□□ □□□□□, □□□□ □□□ □□ □□□ □□□□□ □□ □□□. □□□□ □□□□ □□ □□□ □□□ □ □□□□□?

A. 1. □□□ None□ □□ □□:

2. #□□ □□

3. □□□ □□□:

4. □□

(□□)

B. 1. (□□□ None□ □□ □□)

2. #□□ □□

3.□□□ □□□

4. □□

C. 1. □□□ None□ □□ □□:

2. #□□ □□

3.□:

4. □□

D. 1. □□□ None□ □□ □□:

2. #□□ □□

3. □□ □□:

4. □□

E. 1. □□□ None□ □□:

2. #□□ □□

3. □□□ □□□:

4. □□

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

□□

□□□ □□□ □□□□.

1. □□□ None□ □□ □□:

2. #□□ □□

3. □□□ □□□:

4. □□

NEW QUESTION: 8

Apache Kafka `String` 数据类型在 Delta Lake 中存储为什么类型。

`BINARY`, `BOOLEAN`, `STRING`, `LONG`, `DOUBLE`, `DECIMAL` 类型。默认情况下，`String` 数据类型在 Delta Lake 中存储为 `STRING` 类型。如果配置 `delta.string.storage.dataFormat` 为 `PII`，则 `String` 数据类型在 Delta Lake 中存储为 `PII` 类型。PII 类型在 Delta Lake 中存储为 `STRING` 类型，但会应用额外的加密和访问控制。PII 类型在 Delta Lake 中存储为 `STRING` 类型，但会应用额外的加密和访问控制。PII 类型在 Delta Lake 中存储为 `STRING` 类型，但会应用额外的加密和访问控制。

- A. `String` 数据类型在 Delta Lake 中存储为 `STRING` 类型。
- B. `String` 数据类型在 Delta Lake 中存储为 `PII` 类型，但会应用额外的加密和访问控制。
- C. `String` 数据类型在 Delta Lake 中存储为 `PII` 类型，但不会应用额外的加密和访问控制。
- D. `String` 数据类型在 Delta Lake 中存储为 `PII` 类型，但会应用额外的加密和访问控制，但不会应用额外的加密和访问控制。
- E. `String` 数据类型在 Delta Lake 中存储为 `PII` 类型，但不会应用额外的加密和访问控制。

Answer: B (LEAVE A REPLY)

在 Delta Lake 中，`String` 数据类型默认存储为 `STRING` 类型。如果配置 `delta.string.storage.dataFormat` 为 `PII`，则 `String` 数据类型在 Delta Lake 中存储为 `PII` 类型。PII 类型在 Delta Lake 中存储为 `STRING` 类型，但会应用额外的加密和访问控制。PII 类型在 Delta Lake 中存储为 `STRING` 类型，但会应用额外的加密和访问控制。PII 类型在 Delta Lake 中存储为 `STRING` 类型，但会应用额外的加密和访问控制。

* `String` 数据类型在 Delta Lake 中存储为 `PII` 类型，但不会应用额外的加密和访问控制。

* `String` 数据类型在 Delta Lake 中存储为 `PII` 类型，但会应用额外的加密和访问控制。

NEW QUESTION: 9

在 Delta Lake 中，`customer_id` 数据类型在 Delta Lake 中存储为什么类型。默认情况下，`customer_id` 数据类型在 Delta Lake 中存储为 `STRING` 类型。如果配置 `delta.string.storage.dataFormat` 为 `PII`，则 `customer_id` 数据类型在 Delta Lake 中存储为 `PII` 类型。PII 类型在 Delta Lake 中存储为 `STRING` 类型，但会应用额外的加密和访问控制。PII 类型在 Delta Lake 中存储为 `STRING` 类型，但会应用额外的加密和访问控制。PII 类型在 Delta Lake 中存储为 `STRING` 类型，但会应用额外的加密和访问控制。

- A. `customer_id` 数据类型在 Delta Lake 中存储为 `STRING` 类型。
- B. `customer_id` 数据类型在 Delta Lake 中存储为 `PII` 类型，但会应用额外的加密和访问控制。
- C. `customer_id` 数据类型在 Delta Lake 中存储为 `PII` 类型，但不会应用额外的加密和访问控制。
- D. `customer_id` 数据类型在 Delta Lake 中存储为 `PII` 类型，但会应用额外的加密和访问控制，但不会应用额外的加密和访问控制。
- E. `customer_id` 数据类型在 Delta Lake 中存储为 `PII` 类型，但不会应用额外的加密和访问控制。

Answer: B,E (LEAVE A REPLY)

Liquid Clustering 是 Delta Lake 中的一种优化技术，用于提高查询性能。它通过重新组织数据块来提高查询效率。Liquid Clustering 是 Delta Lake 中的一种优化技术，用于提高查询性能。它通过重新组织数据块来提高查询效率。Liquid Clustering 是 Delta Lake 中的一种优化技术，用于提高查询性能。它通过重新组织数据块来提高查询效率。

在 Delta Lake 中，`customer_id` 数据类型在 Delta Lake 中存储为什么类型。默认情况下，`customer_id` 数据类型在 Delta Lake 中存储为 `STRING` 类型。如果配置 `delta.string.storage.dataFormat` 为 `PII`，则 `customer_id` 数据类型在 Delta Lake 中存储为 `PII` 类型。PII 类型在 Delta Lake 中存储为 `STRING` 类型，但会应用额外的加密和访问控制。PII 类型在 Delta Lake 中存储为 `STRING` 类型，但会应用额外的加密和访问控制。PII 类型在 Delta Lake 中存储为 `STRING` 类型，但会应用额外的加密和访问控制。

NEW QUESTION: 10

在 Delta Lake 中，`customer_id` 数据类型在 Delta Lake 中存储为什么类型。默认情况下，`customer_id` 数据类型在 Delta Lake 中存储为 `STRING` 类型。如果配置 `delta.string.storage.dataFormat` 为 `PII`，则 `customer_id` 数据类型在 Delta Lake 中存储为 `PII` 类型。PII 类型在 Delta Lake 中存储为 `STRING` 类型，但会应用额外的加密和访问控制。PII 类型在 Delta Lake 中存储为 `STRING` 类型，但会应用额外的加密和访问控制。PII 类型在 Delta Lake 中存储为 `STRING` 类型，但会应用额外的加密和访问控制。

`(DATE)`, `(INT)`, `(FLOAT)`, `(FLOAT)`, `(FLOAT)`

在 Delta Lake 中，`customer_id` 数据类型在 Delta Lake 中存储为什么类型。默认情况下，`customer_id` 数据类型在 Delta Lake 中存储为 `STRING` 类型。如果配置 `delta.string.storage.dataFormat` 为 `PII`，则 `customer_id` 数据类型在 Delta Lake 中存储为 `PII` 类型。PII 类型在 Delta Lake 中存储为 `STRING` 类型，但会应用额外的加密和访问控制。PII 类型在 Delta Lake 中存储为 `STRING` 类型，但会应用额外的加密和访问控制。PII 类型在 Delta Lake 中存储为 `STRING` 类型，但会应用额外的加密和访问控制。

`customer_id` > 66.3

在 Delta Lake 中，`customer_id` 数据类型在 Delta Lake 中存储为什么类型。默认情况下，`customer_id` 数据类型在 Delta Lake 中存储为 `STRING` 类型。如果配置 `delta.string.storage.dataFormat` 为 `PII`，则 `customer_id` 数据类型在 Delta Lake 中存储为 `PII` 类型。PII 类型在 Delta Lake 中存储为 `STRING` 类型，但会应用额外的加密和访问控制。PII 类型在 Delta Lake 中存储为 `STRING` 类型，但会应用额外的加密和访问控制。PII 类型在 Delta Lake 中存储为 `STRING` 类型，但会应用额外的加密和访问控制。

- A. `customer_id` 数据类型在 Delta Lake 中存储为 `STRING` 类型。
- B. Parquet 格式在 Delta Lake 中存储为 `PII` 类型，但会应用额外的加密和访问控制。

C. ☐☐ ☐☐☐☐ ☐☐☐ ☐☐☐☐ ☐☐ ☐ ☐☐☐☐☐.

D. ☐☐ ☐☐☐☐ ☐☐ ☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐.

E. Hive □□□□□□ □□ □□ □□□□ □□□ □□□ □□□□□.

Answer: D (LEAVE A REPLY)

[illegible]

□□: [Databricks Certified Data Engineer Professional], "Delta Lake" □□; [Databricks Documentation], "Optimizations - Data Skipping" □□.

Delta Lake □ □□□□ □□□□ □□□□ □ □□□ □□□□ □□ □□□□□. □□□□ □□□□ □□□□ □□□□ □□ □□□□ □□□□□.

- □ □ □ □ □

- □□ □□ 32□ □ □□□□ □□□

- □□ □□ 32□ □ □□□□ □□□

- 0000 00 32 0 0 000 00 null 00 00000. 000 000 000 000 0000 00 000 0, 00 000 00000 000 000 0000 00 000 00000. 000 0000

[illegible]

```
SELECT * FROM EMPLOYEE WHERE EMPLOYEE_ID = 1000;
```

NEW QUESTION: 11

□□ □□□ □□□□□ □□ □□□ □□ □□□, □□□ □□□ □□□ □□ □□□ □□□ □□□□□□ □□□ □□□□□□. □□□ □□ □□□ □□□ □□□ □□□ □□□□□□ □□□□ □□□, □ □ □□ □□□□ □ □□□ □ □□ □□□ □□ □ □□□□□?

A. ☐☐ ☐☐☐

B. ☐☐ ☐☐☐ ☐☐☐

C. ☐☐ ☐ ☐☐

D. UNITY ☐☐☐☐ ☐ ☐☐☐☐☐☐

E. □□□□ □□□ □□□□ □□□□

Answer: B (LEAVE A REPLY)

11

□□□ □□ □□□ □□□□□□.

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

□ □ □ □ □ □ □ □ □ □

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

□□□□ □□ □□□ □□□□□□:

□□□□ □□ □□□□ □□□□□ `expect` □□□□ □□□□□. □□□□ □□ □□□□ □□□□ □□□□ □□ □□ □□□□ □□□□□.

33

```
1.@dlt.expect("□□□ □□□□□", "col("□□□□□") > '2012-01-01'")
```

SQL

1. `valid_timestamp EXPECT (timestamp > '2012-01-01')`

□□□□ □□ □□□□ □□□□□.

```
`expect`  □□ `drop`  □□□□ □□□□ □□□□ □□ □□□□ □□□ □□□ □ □□□□. □□□ □□ □□□□ □□ □□□ □□□□ □□□□□.
```

333

1. @dlt.expect_or_drop("valid_current_page", "current_page_id IS NOT NULL AND current_page_title IS NOT NULL") SQL

[illegible]

1. @dlt.expect_or_fail("valid_count", "count > 0")
 SQL
 1. SELECT valid_count FROM table WHERE valid_count > 0

NEW QUESTION: 12

1. SELECT valid_count FROM table WHERE valid_count > 0

```

-----
AnalysisException                                Traceback (most recent call last)
<command-3293767849433948> in <module>
----> 1 display(df.select(3*"heartrate"))

/databricks/spark/python/pyspark/sql/dataframe.py in select(self, *cols)
    1690         [Row(name='Alice', age=12), Row(name='Bob', age=15)]
    1691         """
-> 1692         jdf = self._jdf.select(self._jcols(*cols))
    1693         return DataFrame(jdf, self.sql_ctx)
    1694

/databricks/spark/python/lib/py4j-0.10.9-src.zip/py4j/java_gateway.py in __call__(self, *args)
    1302
    1303         answer = self.gateway_client.send_command(command)
-> 1304         return_value = get_return_value(
    1305             answer, self.gateway_client, self.target_id, self.name)
    1306

/databricks/spark/python/pyspark/sql/utils.py in deco(*a, **kw)
    121         # Hide where the exception came from that shows a non-Pythonic
    122         # JVM exception message.
--> 123         raise converted from None
    124     else:
    125         raise

AnalysisException: cannot resolve 'heartrateheartrateheartrate' given input columns:
[spark_catalog.database.table.device_id, spark_catalog.database.table.heartrate,
spark_catalog.database.table.mrn, spark_catalog.database.table.time];
'Project ['heartrateheartrateheartrate]
+- SubqueryAlias spark_catalog.database.table
   +- Relation[device_id#75L,heartrate#76,mrn#77L,time#78] parquet

```

□□□ □□□ □□□□ □□□ □□□□□?

- A. □□□ □□□ PvSoark□□□ Scala □□□□□ □□□□□□□.
- B. □□□□ heartrateheartrateheartrate□□ □□□ □□ □□□□.
- C. □□ □□□ □□ □ □□□□ □□ □□□ □□□□□□□.
- D. DataFrame □□□ □□□ □ □ □□□□ □□ □□□ □□□□□□□.
- E. □□□ □□ □□ □□□□ □□□□ □□ □□ □□□ □□□□□□□.

Answer: ([SHOW ANSWER](#))

□□

df.withColumn("heartrate", df["heartrate"] * 3) df.withColumn("heartrate", df.heartrate * 3) df.withColumn("heartrate", col("heartrate") * 3) □□□ □□: [Databricks Certified Data Engineer Professional], "Spark Core" □□; Py4J □□, "Py4J□ □□□□?" □□; Databricks □□, "□□ □□ - □□ □□" □□; Databricks □□, "□ □□" □□.

NEW QUESTION: 13

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

```
A. spark.sql( " ALTER TABLE my_table SET TBLPROPERTIES ( ' delta.deletedFileRetentionDuration '
= '15    ' ) " )
```

B. delta.tables □ □ □ □ □ *

```
deltaTable = DeltaTable.forPath(spark, " /mnt/data/my_table " )
```

```
deltaTable.deletedFileRetentionDuration = " 15  "

```

C. `spark.sql( " VACUUM my table RETAIN 15 HOURS " )`

D. `spark.conf.set( " spark.databricks.delta.deletedFileRetentionDuration " , " 15 days " )`

Answer: A (LEAVE A REPLY)

Delta Lake□□ delta.deletedFileRetentionDuration □□□ VACUUM □□ □□ □□□ □□□ □□□□ □□□□ □□ □□□□ □□□ □□□□□.

* □□□□□ □ □□ □□□ 7□□ □□□□ □□□□.

```
* 00 000 00 000 0000 00 000 ALTER TABLE 00 0000 000 000 000000 000000 0 0000.
```

□□ A□ □□□ SQL □□□□ □□□□□.

```
ALTER TABLE my_table SET TBLPROPERTIES ( 'delta.deletedFileRetentionDuration' = 'interval 15 days' )
```

□ □ □ □ □ □ □ □ □ □ ?

[illegible]

ALTER TABLE `□□ DataFrameWriter □□□ □□ □□□□ □□□.`

* C : VACUUM ... RETAIN□ □□ □□ □□ □□(□: 15□□ □□□ □□ □□)□ □□□□ □□□ □□□□ □□ □□□ □□□□ □□ □□□□. □□□□ □□ □□□ □□□□ □ □□□ □ □□□□.

* D: spark.conf □□□ □□ □□ □□□ □□□ □□□ □□□□, □□□□ □□ □□□□□□ □□□□□ □□□□□□ □□□□□. □□□ □□□□ □ □□□ □□□□□.

□□□ □□ A□ Delta Lake□□ □□□ □□□ 15□ □□ □□□ □□□□□ □□□□ □ □□ □□□□ □□□□ □□ □□□□□.

NEW QUESTION: 14

□□ □□□ □□□□□ □□□ □□□□□□ □□□□ □□□□□ Databricks Lakehouse□ □□□□□□□□ □□□□.

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

Which of the following is a supported operation for Delta Lake?

- A. Delta Lake supports ACID transactions, Databricks supports upsert.
- B. Databricks supports ACID transactions, Delta Lake supports upsert.
- C. Delta Lake supports ACID transactions, Databricks supports upsert.
- D. Delta Lake supports ACID transactions, Databricks supports upsert.

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

Databricks Delta Lake supports ACID transactions, Delta Lake supports upsert. Delta Lake supports ACID transactions, Databricks supports upsert. Delta Lake supports ACID transactions, Databricks supports upsert. Delta Lake supports ACID transactions, Databricks supports upsert.

Delta Lake supports ACID transactions, Databricks supports upsert. Delta Lake supports ACID transactions, Databricks supports upsert. Delta Lake supports ACID transactions, Databricks supports upsert. Delta Lake supports ACID transactions, Databricks supports upsert.

* Databricks Delta Lake supports ACID transactions, Delta Lake supports upsert.

* Delta Lake supports ACID transactions, Databricks supports upsert.

NEW QUESTION: 15

Which of the following is a supported operation for Databricks Asset Bundle?

my_project_job supports ACID transactions, Databricks supports upsert.

my_project_job supports ACID transactions, Databricks supports upsert.

- A. databricks my_project_job -e prod
- B. databricks my_project_job --env prod
- C. databricks run my_project_job -t prod
- D. databricks bundle run my_project_job -t prod

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

Databricks Asset Bundles(DAB) supports ACID transactions, Databricks supports upsert. Databricks Asset Bundles(DAB) supports ACID transactions, Databricks supports upsert. Databricks Asset Bundles(DAB) supports ACID transactions, Databricks supports upsert. Databricks Asset Bundles(DAB) supports ACID transactions, Databricks supports upsert.

-t supports ACID transactions, Databricks supports upsert. Databricks Asset Bundles(DAB) supports ACID transactions, Databricks supports upsert. Databricks Asset Bundles(DAB) supports ACID transactions, Databricks supports upsert.

NEW QUESTION: 16

- A. Delta Lake supports ACID transactions, Databricks supports upsert.
- B. Delta Lake supports ACID transactions, Databricks supports upsert.
- C. Delta Lake supports ACID transactions, Databricks supports upsert.
- D. Delta Lake supports ACID transactions, Databricks supports upsert.

Answer: ([SHOW ANSWER](#))

Delta Lake Delta UniForm(Delta Universal Format) supports ACID transactions, Databricks supports upsert. Delta Lake Delta UniForm(Delta Universal Format) supports ACID transactions, Databricks supports upsert. Delta Lake Delta UniForm(Delta Universal Format) supports ACID transactions, Databricks supports upsert.

□ □□□ □□ □□:

* (A) □□□□ □□ □□□□ □□□□ □□□ □□□□□□ □□□□□□.

* □□□ □□: Delta Lake□ □□ □□□□ □□□, □□ □□□ □□□□ □□ □□ □□□□□ □□ □□□ □□□□ □□□□.

* (B) □□ □□□□□ 'iceberg'□ □□□□□ □□□□□ Iceberg □□□□ □□ □ □□□ □□□.

* □□□:

* □□ □ □□□ □□□ □□□□□ □□□.

* □□□□ □□ □□□□ □□□ □□□□ □□ □□□□ □ □□□□.

* (C) □□□□□ □□□ □ □□□ □□ □□ □□□□ □□□□□ □□□□ □□□□□.

* □□□:

* Iceberg □□□ □□□ □□□□ □□ □□□ □□□□ □□ □□ □□□□ □□□□□.

* Delta UniForm□ Iceberg□ □□ □□□□ □□□□ □ □□□ □□□□ □□□□.

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

* □□:

* □□ □□ □□□ Databricks Delta to Iceberg □□□□□□ □□□ □□□□ □□ Delta □□□□□ Iceberg□ □□□□ □□□□.

* □□ □, □□□□ □□□□□ Delta □ Iceberg □□ □□□□ □□□□ □□ □□□□ □ □□□□.

□□:

Delta□ Iceberg □□ □□ □□□□ □□ □□□ □□□ Delta □□□□□ Iceberg□ □□□□ □□□□ □□□□□ □□□ □□ □□ □□ □□□□ □□□□ □□□□. □□□ □□ (D)□ □□□□□.

□□ □□:

* Apache Iceberg□ Delta UniForm - Databricks □□

* Delta□ Iceberg□ □□ - Databricks

Databricks-Certified-Professional-Data-Engineer □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ Databricks-Certified-Professional-Data-Engineer □□! DumpTop □ □□ **Databricks-Certified-Professional-Data-Engineer** □□ □□□ □□□□□□, DumpTop Databricks-Certified-Professional-Data-Engineer □□ □□□ □□□□□□□□ □□□ □□□□□□□□. □□□□□ □□□ □□□□ □□ DumpTop Databricks-Certified-Professional-Data-Engineer □□□ □□□□□. <https://www.dumptop.com/Databricks/Databricks-Certified-Professional-Data-Engineer-dump.html> (217 Q&As Dumps, **30%OFF Special Discount: KrDump**)

NEW QUESTION: 17

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

□□□ □□□□□ □□ □□ □□□ □□□□ □□□?

A. Databricks Jobs□ □□□□ □□ □□ □□□□ □□□□□ □□□□□ □□□ □□ □□□□ □□□□□.

B. □□ □□□ □□□□ □□ □□ □□ □□□□ □□□□□ □□ □□□ □□□□ □□□□ □□□□□.

C. □□ □□□□ □□ Auto CDC □□□□□□ □□□□ □□ □□□ □□□ □□□□□□.

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

Answer: D (LEAVE A REPLY)

Databricks □□□ □□□□ □□□□ □□□ □□□ □□□□ □ □□□□□ □□□ □□:

□□□□□ □□□□□□ □□□ □□ □□□□□ □□□ □□□ □□□□ □□ □□ □□□□ □□□□ □□□□ □□□ □□□□□. □□□□ □□□□□ □□□□□□□ Databricks□ □□□ □□□□□ □□□

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

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

□□ □□□(A) □□ □□ □□□□ □□□□(B)□ □□□□□□□ □□□ □□ □□ □□, □□ CDC(C)□ □□□□ □□□ □□□ □□ □□ □□□□ □□ □□□ □□□□□.

```

, updates.*
ers
customers.customer_id
= true AND updates.address <> customers.address
mergeKey
.current = true AND customers.address <> staged_u
se, end_date = staged_updates.effective_date
ess, current, effective_date, end_date)
stomer id, staged_updates.address, true, staged_u

```

databricks

```
viewupdates□ customers □□□□ □□ □□ □□□□□ □□ □□ □□□ □□□□ □□ □□□ □□□□□.
□□□ □□□ □□□□ □□□ □□□ □□ □□□ □□□□□.
```

```

MERGE INTO customers
USING (
    SELECT updates.customer_id as merge_key, updates.*
    FROM updates

    UNION ALL

    SELECT NULL as merge_key, updates.*
    FROM updates JOIN customers
    ON updates.customer_id = customers.customer_id
    WHERE customers.current = true AND updates.address <> customers.address
) staged_updates
ON customers.customer_id = mergeKey
WHEN MATCHED AND customers.current = true AND customers.address <> staged_updates.address THEN
    UPDATE SET current = false, end_date = staged_updates.effective_date
WHEN NOT MATCHED THEN
    INSERT(customer_id, address, current, effective_date, end_date)
    VALUES(staged_updates.customer_id, staged_updates.address, true, staged_updates.effective_date,
null)

```

databricks

- A.** □□ □□□□ □□ 3 □□□□ □□□□□□, □□ □□ □□ □□ □□ □□□□ □□ □□□□□.
- B.** □□ □□□□ □□ 2 □□□□ □□□□□□. □□ □□ □□□□□□ □ □□ □□□□ □□ □□□ □□□□ □ □□ □□□□□□.
- C.** □□ □□□□ Type 0 □□□□ □□□□ □□□, □□ □□ □□□ □□ □□ □□□□ □□ □□□ □□ □□□□ □□□□□ □□□□□□.
- D.** □□ □□□□ Type 1 □□□□ □□□□ □□□, □□ □□ □ □□□ □□□□□□ □□□ □□□□ □□□□.
- E.** □□ □□□□ Type 2 □□□□ □□□□ □□□, □□ □□ □□□□□□ □□ □□□ □□□□□□.

□ □□□ MERGE INTO □□□ □□□□ □ □□□□□ □ □□□□ customers □□□□ □□□□□. MERGE INTO □□□ □□ □□□□ □□ □□□ □□ □, □ □□ □□□ □□□□. □□ □□□ □□ □□ □□□ □□ □□□ □□ □□□ □□ □□□ □□ □□□ □□□□□. □ □□ □□□ customers □□□□ □□ □□ customer_id□ □□□□ □□□□ □□□□□ □□□□□. □□□ □□ □□□□ □ □□□□ □□ □□□□ □ □□□ □□□□□□ current_flag□ false□ □□□□ □□ □□□□ □ □□ □□□ □□□ □□□□ □□□□. □□ □□ □□□□ □□ □□□□ □ □□□ □□□□ □ □□□ □□□□ □□ □□□□ □□ □□□□ □□ □□□□ □□□□ □□□□ □□□□ □□□□. □, □□ □□ □□□□□ □ □□ □□□ □□ □□□ □□□□ □ □□ □□□□□, □□ Type 2 □□□□ □□□□□. □□ □ □□ □□: [Databricks Certified Data Engineer Professional], "Delta Lake" □□; Databricks □□□□ "Databricks□ Delta Lake □□" □□□□ □□□□□□.

NEW QUESTION: 19

□□□□ □□□□ □□□□□ □□□ □□□□ □□□ □□□□□□ □□□. □□□ □□, □□ □ □□□ □□□□ □□□□□□ □ □□ □□□ 10□□□□. □□□□ □□ □□□ □□□ □, □□ □□ □ □□□ □□□ □□ □□(SLA) □□ □□□ □□□□ □□□ □□□□□?

- A. □□ □□□ □□□□□□ □□□□□□ □ □□□ □ □ □□□□□ □□□ □□□□□.
- B. □ □□ □□□□□□ □□□□□□ □ □□□□ □□□□□ □□□ □□□□□.
- C. 60□ □□□□ □□□□ □□□□ □□□ □□□□□.
- D. □□□ □□□□□ □ □□□□ □□□ □□□ □□□ □□□ □□□□□.

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

* □□□□ □□: "□□ □□□□□ □□ □□□ □□ □□□□ □□□ □□□□ □□□□□□." □□: □□□□ □□(□□□□ vs. □□); Databricks □□ □□□□.

NEW QUESTION: 20

□□ □□□□ □□ Delta □□□□ device_readings □□□□ □□□□ □□ □□□□ □□□ Power BI □□□□□ □□□□ □□□□.

- * device_id □□□ - □□ □□ ID
- * event_ts TIMESTAMP - □□□□ □□ □□□□□(UTC)
- * temperature_c DOUBLE - □□ □□
- * □□ □□ STRING

□ □□□ □□, □□ 2□ □□□□ □□□ □□ 5□ □□(□: 00:02-00:07, 00:07-00:12 □)□□ □ □□□ □□□□ □□ □□□ □□ □□□ □□□□ □□□.

□□□□ □ □□□ □□ □ □□ □□□□□□ □□□□□ □□ □□□□ □□□ □□□ □□□ □□ □ □□□□. □□ □□ □ □ □□ □□□ □□□□ □□□ □□□□□?

- A. □□ AS (
□□□□
□□ ID,
window(event_ts, '5□', '5□', '2□') AS win, temperature_c FROM device_readings) SELECT device_id, win.start AS bucket_start, win.end AS bucket_end, AVG(temperature_c) AS avg_temp_5m FROM buckets GROUP BY device_id, win ORDER BY device_id, bucket_start;
- B. □□
□□ ID,
□□□ □□□ □□ □□□□,
□□□ □□ □□ □□□,
AVG(□□_□□) AS avg_temp_5m
device_readings□□
GROUP BY device_id, window(event_ts, ' 5 minutes ' , ' 5 minutes ' , ' 2 minutes ') ORDER BY device_id, bucket_start;
- C. □□
□□ ID,
date_trunc('minute', event_ts - INTERVAL 2 MINUTES) + INTERVAL 2 MINUTES AS bucket_start, date_trunc('minute', event_ts - INTERVAL 2 MINUTES) + INTERVAL 7 MINUTES AS bucket_end,
AVG(temperature_c) AS avg_temp_5m FROM device_readings GROUP BY device_id, date_trunc('minute', event_ts - INTERVAL 2 MINUTES) ORDER BY device_id, bucket_start;
- D. □□
□□ ID,
□□□_ts,
AVG(□□_c) OVER (
device_id□ □□
event_ts□ □□□□ □□

□□ 5□ □□□ □□ □ □□□ □□

) AS □□ □□ 5m

device_readings□□

WINDOW w AS (window(event_ts, ' 5 minutes ' , ' 2 minutes '));

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

Spark □□□□□ □□ □□□□ □□ `window(timeColumn, windowDuration, slideDuration=None, startTime=None)`□ □□□□□. `startTime` □□□ □□□ □□□ □□□□ □ □□□□ □□□□□□□ □□□□□, □□□ □□ □ □□ □□□ □□ □□□ □□□□□□. □□ 2□ □□□□ □□□ □□ 5□ □□□ □□ □□□ □□□ □□□□□. (Apache Spark) □□ A□ `window(event\_ts, '5 minutes', '5 minutes', '2 minutes')`□ □□□□ □□□□□, □□ □□□□ □□□□□.

5□ □□□ □□□□ 2□□ □□□□□ win.start□ win.end□ □□□□□. □□ B□ □□□ □□□ □□ □□□□ □□□ □□□□□, □□ C□ □□ 5□ □□□ □□□ □□□□ □□ □ □□□ □□□ □□ □ □□□□, □□ D□ □□□ □□ □□ □□□ □ □□ □□□□ □□ □□ □□□ □□□ □□□□□. (Apache Spark)

NEW QUESTION: 21

□□ 26. 5000□□ □□ □□ □□□ □□ □□□, □□ 1200□□ □□□□□□. □ □ □□□□ □□□□□?

□□ □ □□□ □□□□ "□□□" □□□ □□□□ □□□ □□ □□□ □□□□ □□□□□?

- A. 2.4
- B. 24 0
- C. .24
- D. .48
- E. 4.8

Answer: ([SHOW ANSWER](#))

□□
□□ □□□ □□ □□, □□□ □□□ □□ □□□ □□□ □□ □□□□□□□(MLE)□ □□ □□□ □□□ □□□ □□□□.
□□ □□□□. □□ □□□□ □ □□□ □□ □□ □□□ □□□□□.
□□□□□ □□ □□, □□ □□ □□□ □□ □□□ □□ □ □□□, □□□□ □□ □ □□ □□□ □□□□.
□□□□ □□ □□ □□□ □□□ □ □□ □□□ □□ □□□□ □□ □□□□□□. □□ □□□□,
□□ □□□□ □□ □□□ □□□ □□ □□(□□□□) □□□ □□□.
□□ □□□□ □□ □□ □□ □□ □□□□ □□□□□□□(MLE)□ □□□□ □□□ □ □□□□.
□□ □□□□ □□□□ □□□ □□□ □□□□□ □□ □□ □□□□ □□ □□□ □□□.
□□ □□□ □□□ (□□□□ □□□□ □) □□ □□□□ □□□ □□ □□□□□.
□□□□□, □□□ □□□ □□□ □□ □□ □□□ □□ □□ □□□□ □□□ □□□□□.
□□ □□□ □□□□□ □□ □□□□ □□ □□□□□. □□□□□, □□ □□□ □□□□□□.
□□□ □□□ □□ □□□□ "□□" □□, □□□ □□ □□ □□□ □□ □□□ □□□□□.
□□ □□ □□□ □□ □□□□ □□□ □□□□□□. □□□□□
□□□ □□□ □□ □□□□ □□ □□□ □□□□, □□ □□ □□□ □□□ □ □□□□ □□□□.
□□□ □ □□□ □□ □□□□ □□□□. □□□ □□ □□□ □□□□□ □□□□ □□□□ □□□. □□□ □□□□□□,
□□□□□□□□□ □□□□ □□□ □□□□ □□□□.

NEW QUESTION: 22

□□□ □□□□ GDPR □□□ □□ □□□ □□ □□□ □□□□ □□□□. □□□ □□ □□□□□ □□□ □□ □□□□ □□ □□□ □□□□ □□ □□□ □□ □□□ □□□□□□□.

```
(spark.read
  .format("delta")
  .option("readChangeData", True)
  .option("startingTimestamp", '2021-08-22 00:00:00')
  .option("endingTimestamp", '2021-08-29 00:00:00')
  .table("user_lookup")
  .createOrReplaceTempView("changes"))

spark.sql("""
DELETE FROM user_aggregates
WHERE user_id IN (
  SELECT user_id
  FROM changes
  WHERE _change_type='delete'
)
""")
```

user_id is a column in user_lookup table. user_aggregates table has a column user_id. What is the result of the above code?

- A. Error: Delta Lake does not support BACUM operation.
- B. Error: Delta Lake ACID does not support DELETE operation.
- C. Error: Delta Lake does not support WHERE clause.
- D. Success: Delta Lake ACID DELETE operation is successful.

Answer: A (LEAVE A REPLY)

Delta Lake ACID does not support DELETE operation. It only supports INSERT, UPDATE, and MERGE operations. To delete data from a Delta Lake table, you can use the VACUUM command. VACUUM removes deleted data from the table and compacts the remaining data.

NEW QUESTION: 23

A company is using Delta Lake to store customer churn data. The data is loaded into a table named customer_churn_params. The table has a column named current_timestamp. The company wants to update the churn parameters for all customers who have churned in the last 24 hours. What is the correct SQL statement to achieve this?

- A. customer_churn_params UPDATE SET churn_params = 1 WHERE current_timestamp() < 24.
- B. customer_churn_params UPDATE SET churn_params = 1 WHERE current_timestamp() < 24.
- C. customer_churn_params UPDATE SET churn_params = 1 WHERE current_timestamp() < 24.
- D. customer_churn_params UPDATE SET churn_params = 1 WHERE current_timestamp() < 24.

Answer: (SHOW ANSWER)

customer_churn_params UPDATE SET churn_params = 1 WHERE current_timestamp() < 24.

❏❏❏ ❏❏: "❏❏ ❏❏❏ ❏❏(CDF)❏ ❏❏ ❏❏❏❏ ❏❏ ❏ ❏❏ ❏❏ ❏❏❏ ❏❏❏❏❏❏." ❏❏❏ ❏❏: "table_changes❏ ❏❏❏❏ ❏ ❏❏(❏❏ ❏❏❏❏❏) ❏❏❏ ❏❏, ❏❏❏❏ ❏❏ ❏❏❏ ❏ ❏❏ ❏ ❏❏❏❏." ❏❏❏ ❏❏: "MERGE INTO❏ ❏❏❏ ❏❏ ❏❏❏❏❏❏ ❏❏❏❏❏." ❏❏ ❏ ❏❏❏❏ ❏❏❏❏ ❏❏❏ ❏❏ ❏❏❏❏ ❏❏❏❏❏. CDF❏ ❏❏❏❏❏ MERGE❏ ❏❏❏❏ ❏❏❏ ❏❏ ❏❏❏❏ ❏❏❏❏ ❏❏❏❏❏ ❏ ❏❏❏❏ ❏❏ 24❏❏ ❏❏ CDF❏❏ ❏❏❏ ❏❏ ❏❏❏ ❏❏❏ ❏❏ ❏ ❏❏ ❏ ❏❏❏❏ ❏❏❏❏❏❏.

❏❏:

NEW QUESTION: 24

3❏❏ ❏❏ ❏❏❏ ❏❏❏ ❏❏ ❏❏❏❏❏ Ganglia ❏❏❏❏ ❏❏❏ ❏, ❏❏ ❏❏❏ VM ❏❏❏❏ ❏❏❏ ❏❏❏ ❏❏❏❏❏❏?

- A. ❏ ❏❏❏ ❏❏❏ ❏❏❏❏ ❏❏❏❏❏.
- B. 5❏ ❏❏ ❏❏❏❏ ❏❏❏❏ ❏❏❏❏❏/❏❏❏❏ ❏❏❏❏❏
- C. ❏❏ ❏❏❏ ❏❏ ❏❏ 8❏❏ ❏❏❏❏ ❏❏❏❏ ❏❏❏❏.
- D. CPU ❏❏❏❏ ❏ 75%❏❏❏.
- E. ❏❏❏❏ I/O❏ ❏❏ ❏❏❏❏ ❏❏❏❏.

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

NEW QUESTION: 25

❏❏ ❏❏ ❏❏ ❏❏❏ ❏❏❏❏ ❏❏❏❏ ❏❏ ❏❏❏❏ ❏❏❏❏❏❏ ❏❏❏ ❏❏❏ ❏❏❏❏❏. ❏ ❏❏❏ ❏❏ ❏❏ ❏❏ ❏❏ ❏❏❏❏❏ ❏❏❏ ❏❏ ❏❏❏❏ ❏❏❏❏❏. ❏❏❏ ❏❏❏❏ ❏❏❏❏❏ ❏❏ ❏❏❏❏ ❏❏ ❏❏❏ ❏❏❏❏ ❏❏❏❏ ❏❏❏❏ ❏❏❏❏ ❏❏❏❏ ❏❏❏❏ ❏❏❏❏ ❏❏❏ ❏❏❏ ❏❏ ❏❏❏❏ ❏❏ ❏❏❏❏❏. `user\_id` ❏❏❏ ❏❏❏❏ ❏❏ ❏❏ ❏❏❏❏, ❏❏❏ ❏❏ ❏❏❏❏ ❏❏❏❏.

user_id❏ BIGINT, username❏ STRING, user_utc❏ STRING, user_region❏ STRING, last_login❏ BIGINT, auto_pay❏ BOOLEAN, last_updated❏ BIGINT❏❏❏. ❏❏ ❏ ❏❏❏❏ account_history❏❏ ❏❏❏❏ ❏❏❏❏, ❏ ❏❏❏❏ ❏❏❏ ❏❏❏ ❏❏❏❏ ❏❏ ❏❏❏❏ ❏❏ ❏❏❏ ❏❏❏❏❏. ❏❏❏❏ ❏❏ ❏❏❏❏ account_current❏❏, ❏ ❏❏ user_id❏ ❏❏ ❏❏ ❏❏ ❏❏❏❏ Type 1 ❏❏❏❏ ❏❏❏❏❏.

❏❏❏ ❏❏ ❏❏❏ ❏❏❏ ❏❏❏ ❏❏ ❏❏ ❏❏❏❏ ❏❏❏❏❏ ❏❏❏ ❏, ❏❏❏ ❏❏ ❏❏❏ ❏❏❏ describedaccount_currenttable❏ ❏❏❏❏❏ ❏❏❏❏❏❏ ❏ ❏❏❏ ❏ ❏❏ ❏❏ ❏❏❏ ❏❏❏❏ ❏❏❏❏❏?

- A. ❏❏ ❏❏❏ ❏❏❏❏ ❏❏ ❏❏ ❏❏❏❏❏ ❏ ❏❏❏ ❏❏❏❏, ❏❏❏❏ ❏❏❏❏ ❏❏❏❏ ❏ ❏ ❏❏❏❏ ❏❏ ❏❏❏ ❏❏❏ ❏❏ ❏❏❏❏ ❏❏ ❏❏❏❏❏❏❏ ❏❏❏❏❏.
- B. ❏❏❏ ID❏❏ ❏❏❏❏❏ ❏❏❏ ❏❏❏❏ ❏❏ ❏❏❏❏ ❏❏❏ ❏❏❏❏ ❏❏❏❏❏ ❏❏ ❏❏ ❏❏❏❏ ❏❏ ❏❏ ❏❏❏ ❏❏❏❏ ❏ ❏❏❏❏ ❏❏ ❏❏ ❏❏❏❏ ❏❏❏❏❏.
- C. ❏❏ ❏❏❏❏ ❏❏❏ ❏❏❏❏ ❏❏❏ ❏❏ ❏❏ ❏❏, ❏❏❏ ❏❏❏ ID❏ ❏❏ ❏❏❏ ❏❏❏ ❏❏❏ ❏❏❏❏ ❏❏❏❏❏ ❏❏❏❏❏, ❏ ❏❏❏ ID❏ ❏❏ ❏❏ ❏❏ ❏❏ ❏❏❏❏❏❏❏❏ ❏❏❏❏ ❏❏ ❏❏ ❏❏❏❏❏.
- D. Delta Lake ❏❏ ❏❏❏ ❏❏❏❏ ❏❏ ❏❏❏ ❏❏ ❏❏❏ ❏❏ ❏❏ ❏❏ ❏❏❏❏ ❏❏❏ ❏❏ ❏❏❏ ❏❏❏ ❏❏ ❏❏ ❏❏❏ ❏❏❏❏❏.
- E. ❏❏ ❏❏❏❏ ❏❏❏ ❏❏❏❏ ❏❏❏ ❏❏ ❏❏ ❏❏❏ ❏❏❏❏ ❏❏❏❏ ❏❏❏❏❏, ❏❏❏ ❏❏❏ ❏❏❏❏ ❏❏❏ ❏❏❏. ❏ ❏❏❏ ❏❏❏ ❏❏ ❏❏ ❏❏ ❏❏❏❏❏❏❏❏ ❏❏❏❏ ❏❏ ❏❏ ❏ ❏❏❏❏.

Answer: ([SHOW ANSWER](#))

❏ ❏❏❏ ❏❏❏ ❏❏❏ ❏ ❏❏❏ ID❏ ❏❏ ❏❏ ❏❏ ❏❏ ❏❏❏❏ ❏❏ ❏❏ ❏❏❏❏ ❏❏❏❏❏❏ ❏❏❏❏❏❏. ❏ ❏❏❏ ❏❏❏ ❏❏❏❏ ❏❏❏ ❏❏ ❏❏ ❏❏❏ ❏❏❏❏ ❏❏ ❏❏❏ ❏❏❏❏❏❏ ❏❏ ❏❏❏ ❏❏❏❏ ❏❏❏❏❏. ❏❏ ❏❏❏ ID❏ ❏❏❏ ❏❏❏ ❏❏❏ ❏❏ ❏ ❏❏❏ ❏❏❏❏ ❏❏❏❏❏❏ ❏❏ ❏❏ ❏❏❏ ❏ ❏❏❏ ID❏ ❏❏ ❏❏ ❏❏ ❏❏❏❏❏ ❏❏❏❏❏ ❏. ❏❏ ❏❏ ❏❏ ❏❏ ❏❏❏❏ ❏ ❏❏❏ ID❏ ❏❏ ❏❏ ❏❏ ❏❏ ❏❏ ❏❏❏❏ ❏❏❏❏❏❏❏❏ ❏❏❏❏❏. ❏, ❏❏❏ ID ❏❏ ❏❏❏❏ ❏❏❏ ❏❏❏ ❏❏❏❏❏. ❏❏❏ ❏❏ ❏❏:

[Databricks Certified Data Engineer Professional]❏ "Delta Lake" ❏❏ ❏ Databricks ❏❏❏ "❏❏❏ ❏❏❏❏ ❏❏❏❏ ❏❏❏" ❏❏❏ ❏❏❏❏❏❏.

NEW QUESTION: 26

❏❏❏ ❏❏❏❏❏ ❏❏❏ ETL ❏❏❏❏❏❏ ❏❏ Databricks CLI❏ ❏❏❏❏ ❏❏❏❏❏ ❏❏❏❏❏ ❏❏❏. ❏ ❏❏❏❏❏ ❏❏ 5❏, i3.xlarge ❏❏❏❏ 1❏, ❏❏❏ '14.3.x-scala2.12' ❏❏❏❏ ❏❏❏❏ ❏❏❏.

❏❏❏ ❏❏❏❏❏ ❏❏ ❏❏❏❏ ❏❏❏❏ ❏❏❏❏?

- A. databricks clusters create 14.3.x-scala2.12 --num-workers 5 --node-type-id i3.xlarge --cluster-name DataEngineer_cluster

D. ☐ ☐ ☐ ☐ ☐.

E. □□□□□□□.

Answer: (SHOW ANSWER)

□ □□□ □□ □□ □ □□ □□□□ □□□ event_id □□ □ □□□□ □□ □□ □□ □□□□ □□□□ □□□□ □□□□□. □ □□□ INSERT INTO □□□□ □□□□ new_events □□ □ □□□□ events □□□□ □□□□□. □□□□ INSERT INTO □□□□ □□ □ □(event_id) □□ □□ □□□□ □□□ □□ □□□□ □□ □□ □□□ □□□□ □□□□ □□□□. □□□□ □□ □ □□□ □□□□ event_id □□ □ □□□□ □□ □□ □□ □□□□ □□□□ events □□□□ □□□□ □□□□. □□□□ □□ □□: [Databricks Certified Data Engineer Professional], "Delta Lake" □□; Databricks □□, "INSERT INTO □□□□ □□□□ □□" □□. "□□ merge_condition □□□□ □□ □ □□ □ □□ □□ WHEN MATCHED □□ □ □□ □□ □□□ □□□□ □□□ □□ □□ □□□□ □□ □□□ □□□□□." <https://docs.databricks.com/en/sql/language-manual/delta-merge-into.html#:~:text=If%20none%20of%20the%20WHEN%20MATCHED%20conditions%20are%20not%20satisfied%20then%20the%20insert%20will%20be%20skipped%20and%20no%20update%20will%20be%20performed.> 20□□ □ □□ □ □□ □□□□□ □□□ □□□□□. 20□□ □□%2C%20□□□%20□□%20□□%20□□□□ □□ □□□ □□ □□□□.

NEW QUESTION: 28

□□ □□□□ □□□ □□□□□ □□□□ □□ □□□ □□□□□ Databricks Auto Loader □ □□□ □□ □ □□ □□□ □□□□ □□ □□□ □□□□ □□□□ □ □□□, □□ □□□□□ JSON □□□ □□□□ □□ □□□□□ □□□□, □ □□□ □□□□ □□□□ □□□□ □□□□ □□□□□□□□. □□□□ □□□ □□ □□□□ □□□□□.

```
def auto_load_json(source_path: str,
                  checkpoint_path: str,
                  target_table_path: str):
    (spark.readStream
     .format("cloudFiles")
     .option("cloudFiles.format", "json")
     .option("cloudFiles.schemaLocation", checkpoint_path)
     .load(source_path)
     )
```

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

```
.writeStream
A. .option("mergeSchema", True)
   .start(target_table_path)
   .writeStream
   .option("checkpointLocation", checkpoint_path)
B. .option("mergeSchema", True)
   .trigger(once=True)
   .start(target_table_path)
   .write
   .option("checkpointLocation", checkpoint_path)
C. .option("mergeSchema", True)
   .outputMode("append")
   .save(target_table_path)
   .write
D. .option("mergeSchema", True)
   .mode("append")
   .save(target_table_path)
   .writeStream
   .option("checkpointLocation", checkpoint_path)
E. .option("mergeSchema", True)
   .start(target_table_path)
```

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

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

□□ B□ □□□ □□ □□□ □□□□ □□ □□□ □□□□ □□□□. □□ B□ □□□ □□□□□.

□□ B□ Databricks Auto Loader□ □□□ □□ □ □□ □□□ □□□ "cloudFiles.schemaLocation" □□□ □□□□□. □□ □□ B□ Databricks Auto Loader□ □□□ □□ □□□ □□□ "mergeSchema" □□□ □□□□□. □□□□□ □□ B□ □□ □□□□□ □□□□ JSON □□□ □□□□□ □□□□ □ □□□ "writeStream" □□□□ □□□□□. □□ □□□□ □□ □□□ □□□□ □, □□□ □□□□ □□□□□, □□□ □□□ □□□□ □□□ □□□□ □□□□. □□ □□:

* AutoLoader□□ □□□ □□ □ □□□ □□□□□.

<https://docs.databricks.com/en/ingestion/auto-loader/schema.html>

* □□□□ □□□ □□:

<https://docs.databricks.com/spark/latest/structured-streaming/writing-streaming-data.html>

NEW QUESTION: 29

□□□ □□□□□ □□ □□ □□ □□ 8□□ □□ □□□□ □□□□ □□□□ □□□ □□□□□ □□□□ □□□, □ □□□ □ 20□ □□ □□□□□. □□ □□ □ □□□□ □□□□ □ □□ □□□ □ □□ □□ □□ □□ □□ □□ □□ □□□□□. □□□ □□ □□□ □□□□□ □□□□ □□ □□ □□□ □□□□□?

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

B. □□□ □□ □□□□ □□ 8□ 20□□ □□□ □ □□ □□□ □□□□□.

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

D. □□ □□□ □□□□ 20□□□ □□□□ □□ □□□□ □□, □□□□ 1□□ □□□□ □ □□ □□□ □□□□□.

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

Answer: ([**SHOW ANSWER**](#)**)**

□□

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

Task name * ?

OperationsReporting

Type * Source * ?

Notebook | Workspace

Path * ?

/Users/ [redacted] [edit] [new] [copy]

Cluster * ?

cluster (126.00 GB | 36 Cores | DBR 10.4 LTS | Spark 3.2.1 | Scala 2.12) [edit] [dropdown]

Parameters ? UI | JSON

Add

Depends on

Dataengineering x [dropdown]

Advanced options

Cancel Create task

□□□□□ □□□ □□ □□□ □□□□ □□ □□□□□□□□.



CI/CD process to integrate Databricks with Git?

- A. Yes
- B. Databricks CI/CD process
- C. Yes, but only for production jobs
- D. No, Databricks does not support CI/CD
- E. Yes, but only for development jobs

Answer: (SHOW ANSWER)

Yes

Databricks CI/CD process

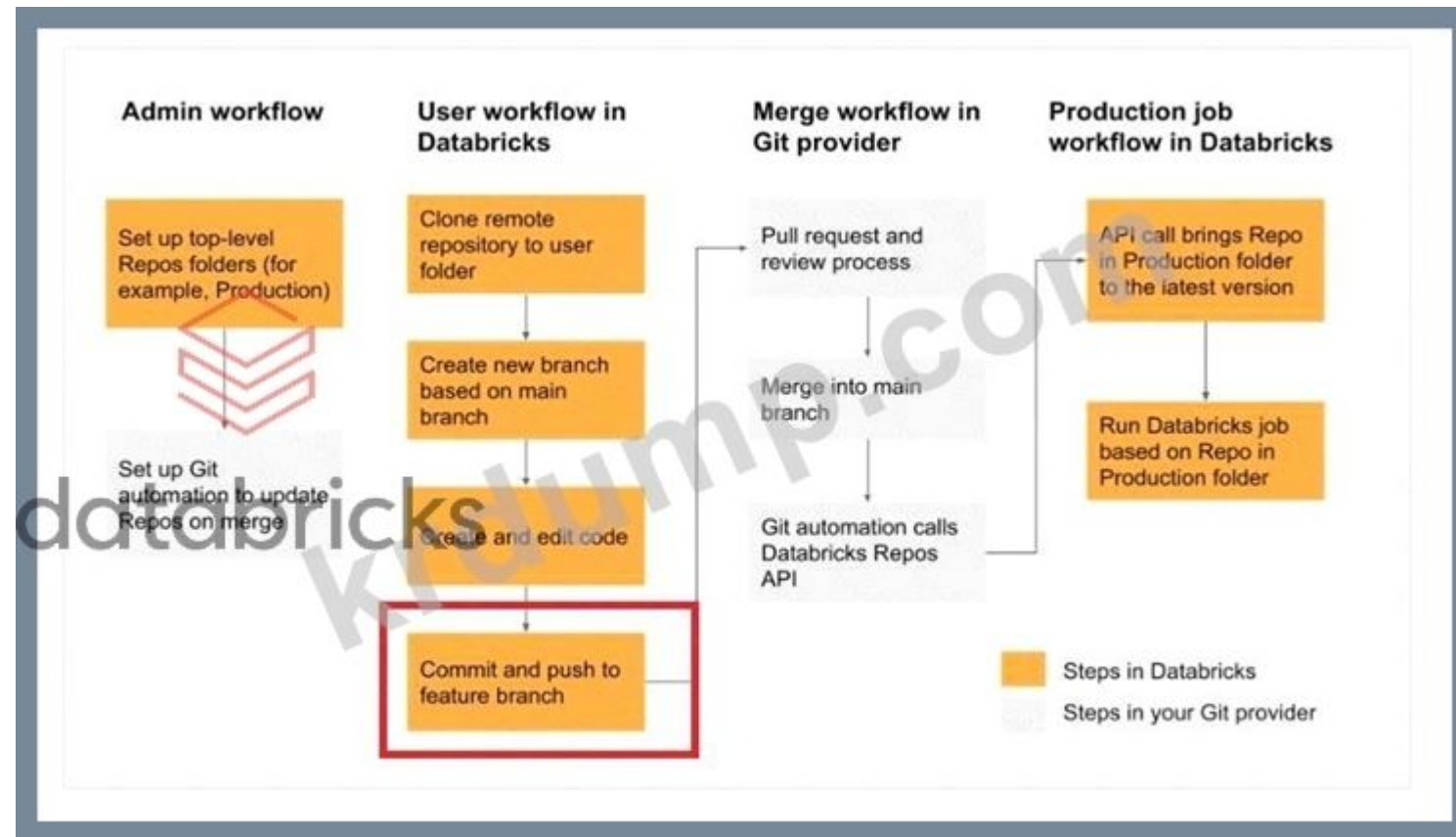
Databricks CI/CD process integrates Databricks with Git using the Databricks Git Provider.

The process involves setting up a Databricks Repo in the Production folder, which is then linked to a Git repository using the Databricks Git Provider.

The process involves setting up a Databricks Repo in the Production folder, which is then linked to a Git repository using the Databricks Git Provider.

The process involves setting up a Databricks Repo in the Production folder, which is then linked to a Git repository using the Databricks Git Provider.

The process involves setting up a Databricks Repo in the Production folder, which is then linked to a Git repository using the Databricks Git Provider.



NEW QUESTION: 31

The Databricks CI/CD process integrates Databricks with Git using the Databricks Git Provider.

The process involves setting up a Databricks Repo in the Production folder, which is then linked to a Git repository using the Databricks Git Provider.

The process involves setting up a Databricks Repo in the Production folder, which is then linked to a Git repository using the Databricks Git Provider.

The process involves setting up a Databricks Repo in the Production folder, which is then linked to a Git repository using the Databricks Git Provider.

The process involves setting up a Databricks Repo in the Production folder, which is then linked to a Git repository using the Databricks Git Provider.

The process involves setting up a Databricks Repo in the Production folder, which is then linked to a Git repository using the Databricks Git Provider.

The process involves setting up a Databricks Repo in the Production folder, which is then linked to a Git repository using the Databricks Git Provider.

Answer: D (LEAVE A REPLY)

Databricks Git □□ □□ □□□□ □□□ □□ □□□ □□□□ UI□ □□□□ □□ □□□ □□ □□□□ □□□□□□. □□□□ □ □□□□□ □□□ □□ □□□ □□□□ □□ □□, □□□□ UI□□ □□ □□□□, □□ □□□ □□□ □ □□□□. □□□□ □□□ □ □□□□□ □□□ □□□□□□ □□□□ □□□ □□ □□ □□ □□□ □□□□ □□□. □ □□□ □□ □ □□□□ □□□ □□ □□□□□ □□□□ □□□□ □□ □□□ □□□□□. □□□ □□(C)□□□ □□□□ □□(B)□□ □□□ □□□□ □□ □□ □□□ □□□□□. □□ □□ □□(A)□□ □□ □□ □□ □□ □□□ □□□□ □□□□. □□□□ D□ □□□ □□ □□□ □□□□ □□□□ □□□□ Databricks□□ □□□ □□□□□.

Databricks-Certified-Professional-Data-Engineer □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ Databricks-Certified-Professional-Data-Engineer □□! DumpTop □ □□ **Databricks-Certified-Professional-Data-Engineer** □□ □□□ □□□□□□, DumpTop Databricks-Certified-Professional-Data-Engineer □□ □□□ □□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □□ DumpTop Databricks-Certified-Professional-Data-Engineer □□□ □□□□□. <https://www.dumptop.com/Databricks/Databricks-Certified-Professional-Data-Engineer-dump.html> (217 Q&As Dumps, **30%OFF Special Discount: KrDump**)

NEW QUESTION: 32

□□□ □□□□□ Lakeflow Declarative Pipelines□ Expectations □□□ □□□□ □□□□ □□ □□□□ □□□ □□□□ □□□□. □□□□ □□□□□ □□□ □□□ □□□ □□□□ □□□□□, □ □□ □□ □□ □□ □□□□ □□□□ □□ □□□□ □□ □□ □□□□ □□□□ □□□□. □□□ □□□ □□ □□□ □□□□□. □□□ □□ □□□ □□ □□□□ □□□□ □ □□ □□ □□□□ □ □□□□ □□□□ □□□□ □□□□.

□□□ □□□ □□ □□□□ □□ □□ □□□□□:

```
@dlt.table
@dlt.expect( " valid_sensor_reading " , " reading < 120 " )
def silver_sensor_readings():
spark.readStream.table(" bronze_sensor_readings " )□ □□□□□.
□□ □□□ □□ □□□ □□□□□?
```

A. @dlt.table

```
@dlt.expect( " valid_sensor_reading " , " reading < 120 " )
def silver_sensor_readings():
spark.readStream.table(" bronze_sensor_readings " )□ □□□□□.
@dlt.table
@dlt.expect( " invalid_sensor_reading " , " reading > = 120 " )
def quarantine_sensor_readings():
spark.readStream.table(" bronze_sensor_readings " )□ □□□□□.
```

B. @dlt.table

```
@dlt.expect_or_drop( " valid_sensor_reading " , " reading < 120 " )
def silver_sensor_readings():
spark.readStream.table(" bronze_sensor_readings " )□ □□□□□.
@dlt.table
@dlt.expect( " invalid_sensor_reading " , " reading < 120 " )
def quarantine_sensor_readings():
spark.readStream.table(" bronze_sensor_readings " )□ □□□□□.
```

```
@dlt.expect_or_drop( " valid_sensor_reading " , " reading < 120 " )
def silver_sensor_readings():
    spark.readStream.table(" bronze_sensor_readings ")
```

@dlt.table

```
@dlt.expect_or_drop( " invalid_sensor_reading " , " reading > = 120 " ) def quarantine_sensor_readings():
spark.readStream.table(" bronze_sensor_readings ") □ □ □ □ □.
```

D. @dlt.table

```
@dlt.expect_or_drop( " valid_sensor_reading " , " reading < 120 " )
def silver_sensor_readings():
    spark.readStream.table(" bronze_sensor_readings ")
```

@dlt.table

```
@dlt.expect( " invalid_sensor_reading " , " reading > = 120 " )
```

```
def quarantine_sensor_readings():
```

```
spark.readStream.table(" bronze_sensor_readings ")
```

Answer: ([SHOW ANSWER](#))

Lakeflow Declarative Pipelines(DLT) @dlt.expect, @dlt

expect_or_drop □ @dlt.expect_all.

* @dlt.expect□ □□□ □□□□ □□ □□□ □□□□□ □□□□□ □□□□□, □□□ □□ □□□□□ □□□□. □□, □□□ □□ □□ □□□□ □□□ □ □□□□.

* @dlt.expect_or_drop□ □□□□□□□□□□ □□ □□ □□□□□□ □□□□□□ □□, □□□□ □□□□ □□ □□□□□□□□□□ □□□□□□.

☐ ☐☐ ☐☐ ☐☐☐ ☐☐☐ ☐☐☐☐.

* □□ □(□□ □ < 120)□ □□ □□□□ □□□□□.

* □□ □(□□ □ >= 120)□ □□ □□□□ □□□□□.

* □□ □□ □□□ □□□□□ □□□ □□□.

□□□ □□ □□□ □□ A□□, □□□:

```
* 0000 @dlt.expect 0000 120 000 00 00 0000 00000.000 00 00000 00000.
```

* □□ □□□□ □□ □□ 120 □□□□□ □□□ □□□ □□□□ □□ □□□□ □□□ □□□□□.

□□ □□□□ □□□□□□□□.

* □□ B/D: □ □□ expect_or_drop□ □□ □□□□□ □□□ □□□ □□□□ □□ □□□ □□□□ □□ □□□□□.

* □□ C: □ □□□ □□□ □□ expect or drop□ □□□□□. □□□□ □□ □□□ □□ □□ □□□□ □□□□ □□ □□□□□.

[illegible]

□□: Databricks □□ - Delta Live Tables(DLT) □□ □□: @dlt.expect, @dlt.

expect_or_drop ☐ ☐ ☐ ☐ ☐.

NEW QUESTION: 33

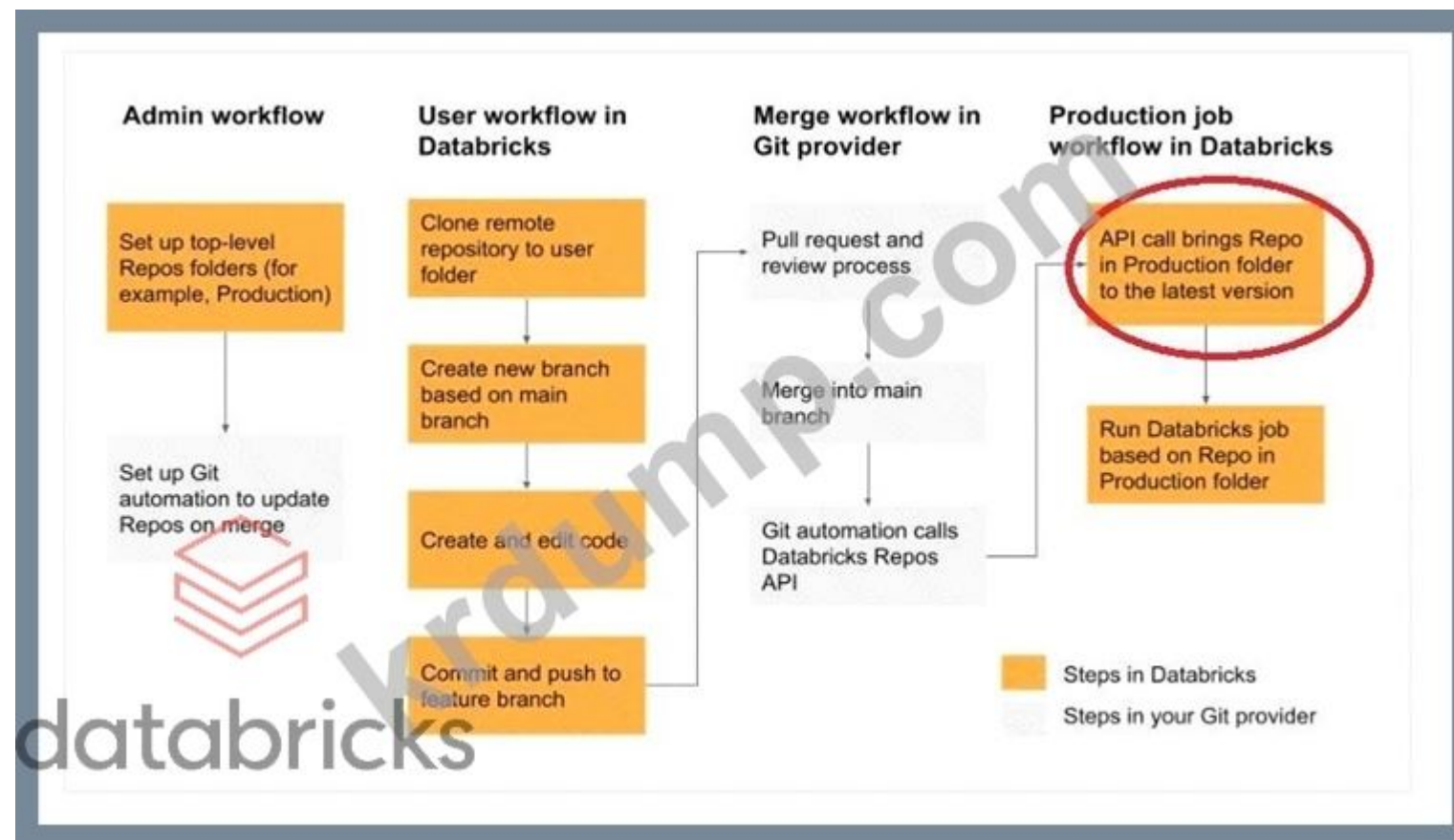
Databricks 300 000 000 000, 0 000 000 Databricks 000 000. 000 A 00 0000 0000 0000. 000 B C 000 0000, 00 000 A 00 0000 0000.

□□ A□ □□□ □□ □□ □□□ □□, □ □□ □□□ □□□□ □□□ □□□□□?

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

B. □□ B□ C□ □□□ □□ □□□ □□□□□. □□ A□□ □□□ □□□ □□ □□□ □□ □□□□□□.

C. □□ □□□ □□□□□ □□□□ □□□ □□□□□□□ □□ □□□ □□□□ □□□□. □□ A□ □□□□□□ □□ □□□ □□□□ □□□□□.



NEW QUESTION: 36

Which of the following is a valid Delta Lake table name?

```
from pyspark.sql.functions import current_timestamp, input_file_name, col
from pyspark.sql.column import Column
```

```
def ingest_daily_batch(time_col: Column, year:int, month:int, day:int):
    (spark.read
     .format("parquet")
     .load(f"/mnt/daily_batch/{year}/{month}/{day}")
     .select(
         time_col.alias("ingest_time"),
         input_file_name().alias("source_file")
     )
     .write
     .mode("append")
     .saveAsTable("bronze")
    )
```

Which of the following is a valid Delta Lake table name?

Which of the following is a valid Delta Lake table name?

B. `spark.readStream.load("bronze")` ☐ ☐ ☐ ☐ ☐.

C.

```
return (spark.read
        .table("bronze")
        .filter(col("source_file") == f"/mnt/daily_batch/{year}/{month}/{day}")
)
```

E.

11

NEW QUESTION: 37

[illegible][illegible]

NEW QUESTION: 38

$$Y = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n$$

D. □□ □□□□□

$$\square\square: RY = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n$$

1. 在多元线性回归模型中，如果自变量之间存在多重共线性，会对模型的估计和预测产生什么影响？请简要说明原因。

NEW QUESTION: 39

1. 在多元线性回归模型中，如果自变量之间存在多重共线性，会对模型的估计和预测产生什么影响？请简要说明原因。

- A. 模型的估计量仍然是无偏的，但方差会增大。
 B. 模型的估计量是有偏的，且方差也会增大。
 C. 模型的估计量仍然是无偏的，且方差也会增大。
 D. 模型的估计量是有偏的，但方差不会增大。
 E. 模型的估计量仍然是无偏的，且方差不会增大。

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

1. 在多元线性回归模型中，如果自变量之间存在多重共线性，会对模型的估计和预测产生什么影响？请简要说明原因。

1. 在多元线性回归模型中，如果自变量之间存在多重共线性，会对模型的估计和预测产生什么影响？请简要说明原因。

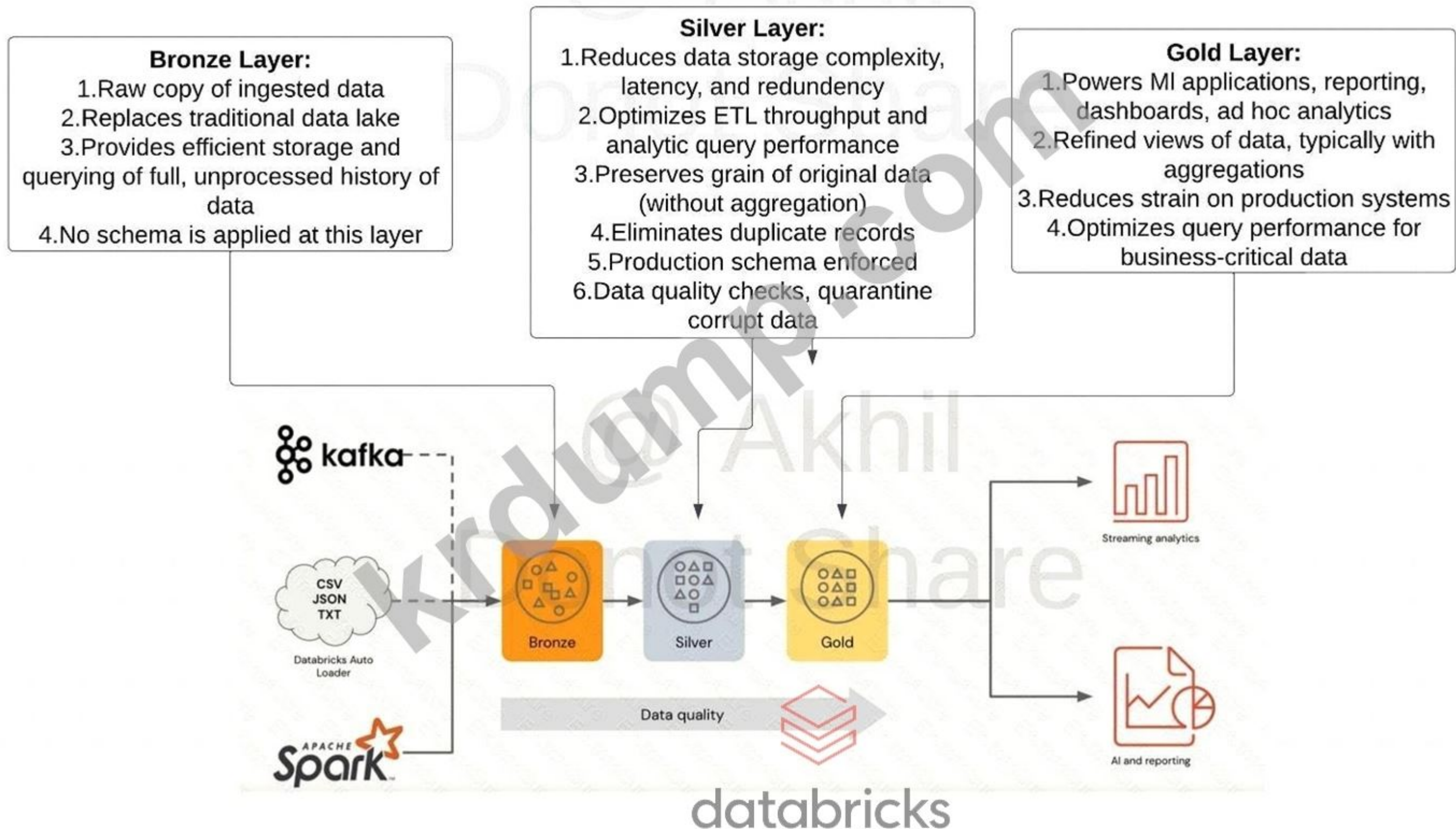
1. 在多元线性回归模型中，如果自变量之间存在多重共线性，会对模型的估计和预测产生什么影响？请简要说明原因。

1. 在多元线性回归模型中，如果自变量之间存在多重共线性，会对模型的估计和预测产生什么影响？请简要说明原因。
 2. ETL 过程是指数据抽取、转换和加载的过程。
 3. 在数据仓库中，数据通常以星型模型或雪花模型进行组织。
 4. 数据仓库的主要目的是为决策支持系统提供数据。
 5. 数据仓库中的数据通常是历史数据。
 6. 数据仓库中的数据通常是集成数据。

1. 在多元线性回归模型中，如果自变量之间存在多重共线性，会对模型的估计和预测产生什么影响？请简要说明原因。

1. 在多元线性回归模型中，如果自变量之间存在多重共线性，会对模型的估计和预测产生什么影响？请简要说明原因。

1. 在多元线性回归模型中，如果自变量之间存在多重共线性，会对模型的估计和预测产生什么影响？请简要说明原因。



NEW QUESTION: 40

□□ □□ □ DELTA □□□□ □□□□ □ □□□ □ □□ □□□□□?

A. □□□ □□ □□

B. DROP DELTA table_name

C. DROP TABLE table_name FORMAT DELTA

D. _

Answer: D (LEAVE A REPLY)

NEW QUESTION: 41

[illegible]

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

A. Scalaは、Javaと互換性のある型システムを持つ。ScalaはJARファイルで配布され、PySparkやSpark SQLなど、他のデータ処理フレームワークとも連携する。

B. □□ □□□□ □□ □□ □□□ □□ □□ □□□□ □□□ □□□ □□□□ □□□ □□□□ □□□□ □□□□ "□□ □□" □□□□□□ □□□□.

C. □□ □□ □□□□ □□ □□□ □□□ IDE□ □□□□ □□□□ □□□. □□ □□ Spark □ Delta Lake□ □□ □□ □□□□ □□□ □□□□ □□ □□ □□□□□ □□ □□□ □□ □□□□ □□ □□□□.

[illegible]

E. Photon □ □□□ □□□ □□ □□□ □□□□□□□□ □□□□ □ □□□□, □□ □□ □□ □□ □□□□ □□□□ □□□□ □□ □□□ □□□□□ Jobs UI□ □□□□ □□□.

Answer: D (LEAVE A REPLY)

0000 000000. 00000 00 00 00 0 00 00 00000 00000 00 0000 0 00000 00000 0 0000 0000 00000 000000. 00 0000 `display()` 0000 0000 0000 0000
 00 00, 00 0000 0000 00 00000 000000 00000. 00000 00 0000 0000 000000 00000 00 00 0000 0000 00000. Databricks 000000 0000 0000 00 Spark
 0 0000 0000 00000 0000 000000 0000. 0000 00 DataFrame 00 Dataset00 0 DataFrame 00 Dataset0 00000 00000, `filter`, `select`, `join` 00 00000. 0000 DataFrame 0
 0 Dataset0 00 0000 000000 00000 00000000 0000 000000 00000 00000 00000, `count`, `show`, `save` 00 00000. DataFrame 00 Dataset00 `display()` 0 00000 0
 00 0000 000000 0000 00 0000 00000 000000. Spark0 0000 00 00 0000 000000, 00 0000 0000 0000 0000 00000 0000 000000. 00 Spark0 0000 00
 00 00 0000 0000 00 00000 00000 00 00000 0 0000 00000 0 0000 0000. 0000 display()0 00000 0000 0000 00000000, 00 00 0000 0000 00 00000 0
 00000. 00000 00 0000 0000 000000 00000 00 00 0000 00 0 00000. 00000 00000 00 0000 00 00000 0000000 display()0 00 00 00000 0000 0 00
 00000 00 0000 0000 0000. 0000 00 00: [Databricks Certified Data Engineer Professional], "Spark Core" 00; Databricks 00, "00 00" 00; Databricks 00, "00" 00.

NEW QUESTION: 42

0000 000000 00 00 00 0000 00 0000 00000000 0000 00000000 0000000000 00000. 0000 0000, 00, 00 0000 00 0000 00000 00 000000 00
 0 000000. 0000 00000 00 00000 0000 0000000 000000 00000, 00 00000 0000 0000000 0 00 00 000000 00 00000 0 000000. 00 00000 00 00
 00 000000 0 00 00000 000000. 00 0000 0000 00 00 00(PII)0 000000, 00 0 00 0000 00 0000000 0000 0 0000 0000 000000.

[illegible][illegible][illegible][illegible]

C. Storing □□ □□□□ □□□□ □□ □□□□□□ □□□□ □□□□□□ □□□□ □□ □□□ □□ □□□ □□□ □□ □□□ □□□ □□□□, □□ □□□□□ □□ □□□□□□□ □□ □□
□□□ □□□□ □□□ □□□□□□.

[illegible]

E. ☐ I agree to pay my child's tuition at the private school named above for the year ending _____, and I understand that if I fail to do so, the State may remove my child from the public schools.

Answer: A (LEAVE A REPLY)

00 00 000 000 00 00 000 00000 000000. 0000 00 00(0: 000, 00, 00)0 00 00000 000 00000000 0000000 000 000000 00 00 00 000 00 0 00000. 00, 0000000 ACL(00 00 00)0 00 0000 0000 0 0000 00 0000 00 0000 0 0000, 00 00 0000000, 0000 00 00 00 00 0000 000000 0 00 0 00000. 00, 0 0000000000 00000 00000 00 00 0000 000000 00000 0000 000000 0000 0000 0 00000. 00, 0 000000000 00000 00000 0000 0000 00000 00 00000 00000 0000 0 00000. 00 000 00000 00 00000 00000 0000 0 00000. 00 00: [Databricks Certified Data Engineer Professional], "Lakehouse" 00; Databricks 00, "00000000 00 00" 00.

NEW QUESTION: 43

0000 0000000000 0 00 00000 0 00 0000 000000 000000000. 0 00 0000 00 000000 Parquet 00000 /mnt/raw_orders/0 00000 0000000 00000 00000. 0 00000 0000 0, 00 0000 0000 Databricks 0000 000000. (0000.000000) .format(" parquet ") .load(" /mnt/raw_orders/ ") .withWatermark("00 " , "200 ") .dropDuplicates([" customer_id " , " order_id "]) .writeStream .0000(once=True) .table(" 00 ")) customer_id0 order_id 0000 0 0000 00000 00000 00 0 0000 00, time 0000 00 000000 00000 00000 0000 0000 000000 000000. 00 000000 00 0000 00 0 00 00000 00 0000 00000 00000 0000 0000 00 0000 0000 0000, 00 0 00 000 000000? A. 00 000000 0000 0000 0000, 200 00 00 0000 0000 000000 000000. B. 00 000000 00 200 0000 0000 00000 0000 0000 00000. C. 00 0000 00 00 0 00 00000 00000 00 00 00000 200 00 000000. D. 200 00 0000 00 000000 0000 00 00000 0000 0 0000, 00 000000 customer_id0 order_id0 00 00 00000 0000 0 00000.

Answer: A (LEAVE A REPLY)
 * 0000 00: "dropDuplicates with watermark0 00000 00 00 00 00 00 00 00 0000 000000."
 * 0000 00: "0000 00 00000000 0000 00000 0000 0000 00000 0000 0 00000." 00 00: 00000 00000 00000 0 00 00; 0000 0000.

NEW QUESTION: 44

00 0000 00 0000 00000 Delta Lake 00000 00000 000000.

Which of the following is the correct Delta Live query to create a table sales_orders_cleaned from the Delta Live table sales_orders, partitioned by city, with a partition for sales_order_in_chicago?

1. sales_order_in_chicago FROM sales_orders_cleaned

2. AS

3. SELECT order_date, city, sum(price) as sales,

4. FROM sales_orders_cleaned

5. WHERE city = 'chicago'

6. GROUP BY order_date, city

A. FROM sales_orders_cleaned WHERE city = 'chicago'

B. FROM sales_orders_cleaned WHERE city = 'chicago'

C. FROM sales_orders_cleaned WHERE city = 'chicago'

D. Sales_orders_cleaned FROM LIVE sales_orders_cleaned

E. FROM sales_orders_cleaned WHERE city = 'chicago'

Answer: (SHOW ANSWER)

Which of the following is the correct Delta Live query to create a table sales_orders_cleaned from the Delta Live table sales_orders, partitioned by city, with a partition for sales_order_in_chicago?

1. sales_order_in_chicago FROM sales_orders_cleaned

2. AS

3. SELECT order_date, city, sum(price) as sales,

4. FROM LIVE.sales_orders_cleaned

5. WHERE city = 'chicago'

6. GROUP BY order_date, city

NEW QUESTION: 46

Which of the following is the correct command to clone the repository https://github.com/foo/data_loader into the directory /dbfs/mnt/new_data?

```
%sh
git clone https://github.com/foo/data_loader;
python ./data_loader/run.py;
mv ./output /dbfs/mnt/new_data
```

Which of the following is the correct command to clone the repository https://github.com/foo/data_loader into the directory /dbfs/mnt/new_data?

Which of the following is the correct command to clone the repository https://github.com/foo/data_loader into the directory /dbfs/mnt/new_data?

A. %sh Git is the correct command to clone the repository https://github.com/foo/data_loader into the directory /dbfs/mnt/new_data.

B. %sh pip install is the correct command to clone the repository https://github.com/foo/data_loader into the directory /dbfs/mnt/new_data.

C. %sh mv is the correct command to clone the repository https://github.com/foo/data_loader into the directory /dbfs/mnt/new_data.

D. Databricks is the correct command to clone the repository https://github.com/foo/data_loader into the directory /dbfs/mnt/new_data.

E. %sh mv is the correct command to clone the repository https://github.com/foo/data_loader into the directory /dbfs/mnt/new_data.

Answer: E (LEAVE A REPLY)

https://www.databricks.com/blog/2020/08/31/introducing-the-databricks-web-terminal.html is the correct command to clone the repository https://github.com/foo/data_loader into the directory /dbfs/mnt/new_data.

Which of the following is the correct command to clone the repository https://github.com/foo/data_loader into the directory /dbfs/mnt/new_data?

□ □□□, Spark□ □□ □□ □ □□□ □□□□ □□□□. □□ □□, Databricks Connect □□□ □□□□ □□ Databricks □□□□□□ Python □□□ □□□□□ Spark Git Connector□ □□□□ Git □□□□□□□ Spark DataFrame□□ □□□□ □□ □ □□□□.

Databricks-Certified-Professional-Data-Engineer □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ Databricks-Certified-Professional-Data-Engineer □□! DumpTop □ □□ **Databricks-Certified-Professional-Data-Engineer** □□ □□□ □□□□□□, DumpTop Databricks-Certified-Professional-Data-Engineer □□ □□□ □□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □□ DumpTop Databricks-Certified-Professional-Data-Engineer □□□ □□□□□. <https://www.dumptop.com/Databricks/Databricks-Certified-Professional-Data-Engineer-dump.html> (217 Q&As Dumps, **30%OFF Special Discount: KrDump**)

NEW QUESTION: 47

□□ □□□ Parquet □□□□ □□ □□□ □□ □□□ □□□ □□□□□ □□ □□□□ □□□□□. □□ □□ □□□ □□ □□□ □□□ □□□ □□ □□□□ □□ □□□□ □□ □□ □□□ □□□□ □.



customer_id□ order_id □□□ □ □□□ □□□□ □□□□ □□ □ □□□ □□□ □□□□□.

□□ □□□□□ □□ □□□ □□ □ □□ □□□□ □□ □□□ □□□□ □□□ □□ □□□ □□□ □□□, □□ □ □□ □□□ □□□□?

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

Answer: (SHOW ANSWER)

□ □□□ □□□ □□□ `dropDuplicates` □□□□ □□□□ □ □□□ □□ □□ □□ □□□□ □□□ □ `orders` □□□□ □□ □□□□□□. □□□ □ □□□□ □□ □□ □□ □□ □□ □□□ □□ □ □□□□ □□□□ □□ □□□ □□□□ □□ □□□□ □□ □□□□ □□ □□□□ □□ □□□□ □□□ □ □□□□. □□ □□□□□ Delta Lake□ □□□□ `mergeInto`□ □□□□ □□□(upsert) □□□ □□□ □ □□ □ □□ □□□□□. □□□□ □□ □□: [Databricks Certified Data Engineer Professional], "Delta Lake" □□; Databricks □□, "DROP DUPLICATES" □□.

NEW QUESTION: 48

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

□□□ □□□□□ □□ □ □□□ □□□ □□□□.

dlt□ □□□□□

pyspark.sql.functions□□ expr□ □□□□□.

□□ = {

" valid_lat " : " (lat IS NOT NULL) " ,

```

" valid_long " : " (long IS NULL OR long IS NOT NULL) "
}

quarantine_rules = " NOT({0})" .format( " AND " .join(rules.values()))

@dlt.view
def raw_trips_data():
    spark.readStream.table(" ride_and_go.telemetry.trips ") \
        .select(
            col("lat"), col("long"), col("time"), col("trip_id"),
            col("driver_id"), col("passenger_id"), col("vehicle_id"),
            col("status"), col("type"), col("category"), col("sub_category")
        )

A. @dlt.table(partition_cols=[ " is_quarantined " ])
@dlt.expect_all(10)
def trips_data_quarantine():
    (
        spark.readStream.table( " raw_trips_data " )
        withColumn( " is_quarantined " , expr(quarantine_rules))
    )

B. @dlt.table
@dlt.expect_all_or_drop(rules)
def trips_data_quarantine():
    spark.readStream.table("raw_trips_data") \
        .select(
            col("lat"), col("long"), col("time"), col("trip_id"),
            col("driver_id"), col("passenger_id"), col("vehicle_id"),
            col("status"), col("type"), col("category"), col("sub_category")
        )

C. @dlt.table(name= " trips_data_quarantine " )
def trips_data_quarantine():
    (
        spark.readStream.table( " raw_trips_data " )
        .select(
            col("lat"), col("long"), col("time"), col("trip_id"),
            col("driver_id"), col("passenger_id"), col("vehicle_id"),
            col("status"), col("type"), col("category"), col("sub_category")
        )
    )

D. @dlt.view
@dlt.expect_or_drop( " lat_long_present " , " (lat IS NOT NULL AND long IS NOT NULL) " ) def trips_data_quarantine():
    spark.readStream.table(" ride_and_go.telemetry.trips ") \
        .select(
            col("lat"), col("long"), col("time"), col("trip_id"),
            col("driver_id"), col("passenger_id"), col("vehicle_id"),
            col("status"), col("type"), col("category"), col("sub_category")
        )

```

Answer: ([SHOW ANSWER](#))

Databricks Lakeflow Spark □□□ □□□□□□ □□ □□ □□□ □□□□□□□□. □ □□□□□ □□□ □□ □□□□ □□ □□ □□ □□□□ □□□□□ □□□□□, □□ □□□ □□ `is\_quarantined` □□□□ □□□ □□, □□ □□□□□ □□□ □□□□□ □□ □□□ □□□□□□. □□□□□ □□□ □□ □□ □□□□ □□□□ □□ □□ □□□□□ □□□□ □□ □□ □□□ □□□□□□. (Databricks □□) □□ A□ □□□□ □□□ □□□□ □□□ □□□□□□□□. □□ □□□□ □□□□□, □□□□□ □□ □□ `is\_quarantined` □□□□□ □□□□□, □□□□□ □□□□□ □□□ □□ □□□ □□□□□□. □□ B□ □□□□ □□ □□ □□□□□ □□□ □□□□□ □□□□ □□ □□ □□□ □□□□□ □□□□□. □□ C□ □□ □□ □□□□□□ □□ □□□ □□□ □□ □□□□□. □□ D □□ □□□□ □□ □□ □□ □□□ □□□□□ □□□ □□□□□ □□□ □□ □□□ □□□□ □□□□□. (Databricks □□)

NEW QUESTION: 49

□□□□□ □□□□ □□□□ □□□□□ □□□ □□□□ □□□□ □ □□□ □ □□ REST API □□□ □□□□□?

- A. /jobs/runs/list**
- B. /jobs/runs/get-output**
- C. /jobs/runs/get**
- D. /jobs/get**
- E. /jobs/list**

□□ □□

* □□□□ □□□□ □□□□□ □□□ - □□□□

Databricks □□ □□□□

NEW QUESTION: 51

Databricks SQL □□ □□ □□□ □□□□ □□ □□□□□□. □□ □□ □□□ □□ □□□ □□□□ □□ □□ □□□ □□□□ □□□. □□□ □□□ □□ □□ □□□ □□□ □□□□ □□□ □□ □□ □□□□ SQL □□□□□(SQL Warehouse)□ □□□□ □□□ □□ □□□□□. □□ □ □□ □□/□□ □□□ □□□□ □□ □□ □ □□ □□□ □□□□□?

*□□□ Databricks□ □□ SQL □□□□□□ □□□ SQL □□□□□□ □□□□□□.

- A. SQL □□□□□(SQL □□□□□)□ □□ □□□□ □□□ □□□□ □ □□□□.
- B. SQL □□□□□(SQL □□□□□)□ □□ □□□ □□□□ □□ □ □□□□.
- C. SQL □□□□□(SQL □□□□□)□ □□□□□ □□□ 2□ □□□□ 4□ □□□□ □□□ □ □□□□.
- D. SQL □□□□□(SQL □□□□□)□ □□ □□ □□ □□□ □ □ □□□□.
- E. SQL □□□□□(SQL □□□□□)□ □□ □□□□ □□□ □□□□□ □□ □□□□ □□□ "□□□□ □□□"□ □□□ □ □□□□.

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

□□

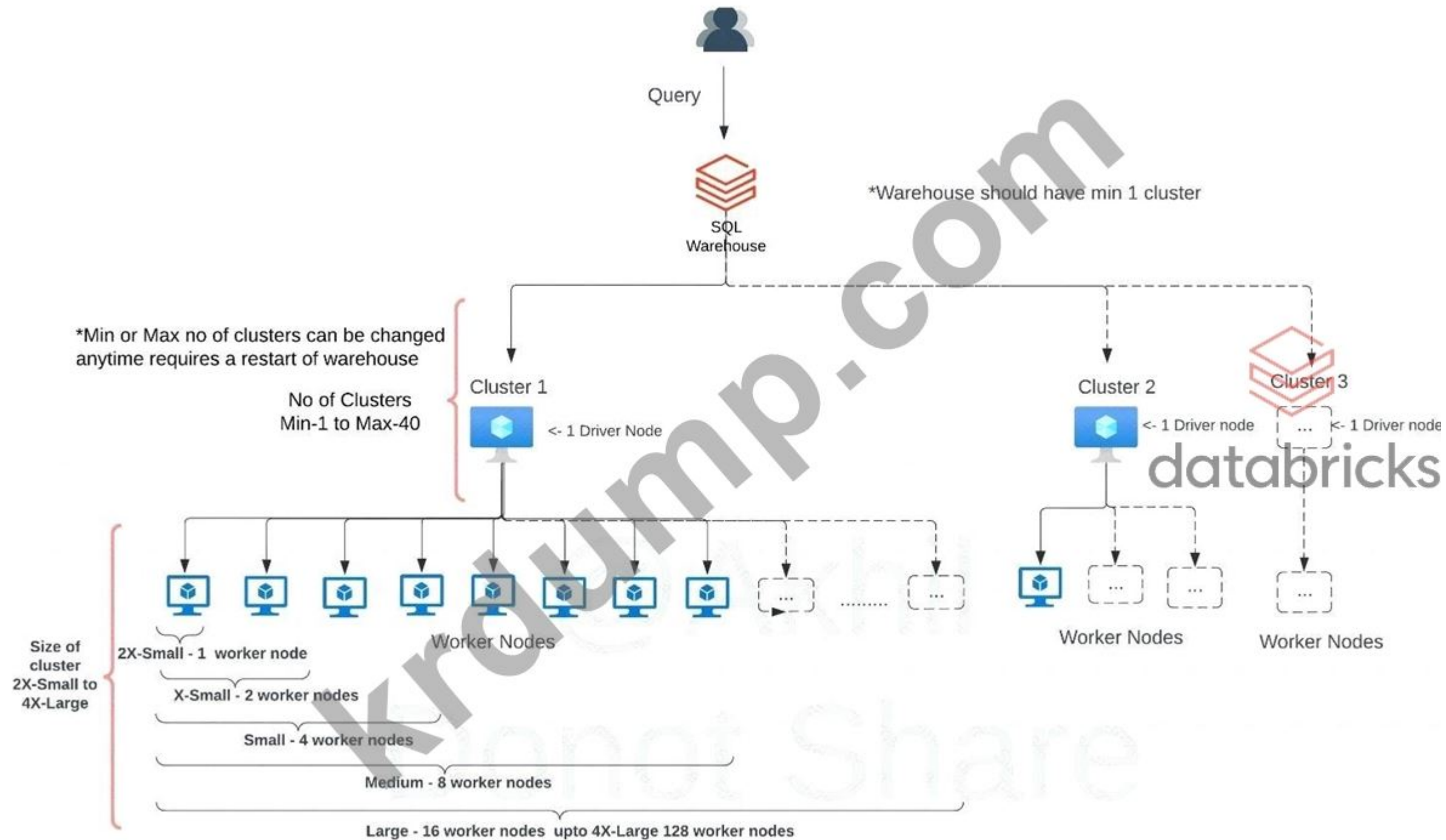
□□ SQL □□□□□□ □□ □□ □□□□ □□ □ □□□ □□□□. □□ □□ □□□ □□□ □ □□ □□□□□ □□□□□ □□□ □□□□□ □□□□ □□ □□ □□□ □□□□□ □□□□ □□ □□ □□□□ □□□□ □□ □□ □□□□.

□ □□□ SQL □□□□□(SQL □□□□□)□ □□ □□□ □□□□ □□□□ □□□□ □□ □□□, □□□ □□□□□ □□□□□ □□□ □□□□□ □□□□ □□□. □□□ □□□□□ □□□□ □□ □□□□ □□□ 2□ □□□□ 4□ □□□□ □□(□□□□ □)□□, □□□ □□□ □□□□□ □□□□ □□□□ □□ □□□□□ □□□□ □□(□□□□ □□)□□ □□□.

SQL □□□□□(SQL □□□□□) □□: (□□□□ □□ □□□ □□ □□□ □□□□□□ □□□□□ □□□□□.)

1. SQL □□□□□□□□ □□ □□ □□□ □□□□□ □□□ □□□.
2. □□□□□ □□□ □□□□ □□□ □□ □□□ □□ □□□ □□□□□.
3. □□□□□ □□ □□ □□ □□□□ □□□ □□ □□□□□(2X-Small -> □□ □□ 1□, X-Small -> □□ □□ 2□... □□ 4X-Large -> □□ □□ 128□). □□ □□□□□□□ □□□.
4. □□□□ □□(2X-Smal□□ ...4XLarge□□)□ □□□□ □□ □□□□□ □□□□ 3X-Large □□□□ □□□ □□□□ □□□(□□)□ □□ □□□ □□□□□□ 20□□ □□□ □ □□ □□□□ □□ □□ □□ □□ 10□□ □□□ □□□ □ □□□□.
- 1, max1) 10□□ □□□ □□□ □□□□ □□ □□□ 10□□ □□□ □ 10□□ □□□ □□□ □□□ □□□□□ □□□□□.
5. □□□□□ □□□□ □□□ □□□ □□ □□□ □□□ □ □□□□. □□ □□, 2X-Small □□□ □□□□□□□ □□ □□□ 1□□ □□□ □□, □□□□□ □□□ X-Small□ □□□□ 30□ □□ □□ □ □ □□□□. □□ 2X-Small□□ □□ □□□ 1□□□□ X-Small□□ □□ □□□ 2□□□□ □□□ □ □□ □□□ □□□□ □ □□□ □□□□ □□□□□. (□□: □□ □□□□ □□□□, □□ □□□ □ □□□ □□ □□□ □□ □□□□□ □□ □□□□□ □□□□ □□ □□□□.)
6. □□□ □□□□□□ □□ □□ □□□□□ □□ □ □□□, □□ □□□ □□□□□ □□□. □□□□□□ □□□□ □□□□(□□ 1□, □□ 2□)□ □□□ X-Small □□□□ □□□ □□□ □□, Databricks□ □□□□ □□□ □□□□ □□ □□□□□ □□□□□. □□ □□ □□□□□□ 2□□ □□□□(□□ 1□, □□ 2□)□ □□□□□ □□□□ □□ □□□□ 20□□ □□□ □□□□ 10□□ □ □□ □□□□ □□□ □□□ □□□□ □□ □□□. □□□□ Databricks□ □□□□ □ □□ □□□□□ □□□□ □□□□ □□ 10□□ □□□ □ □□ □□□□□ □□□□□□□□.
7. □□ □□□ □ □ □□□ □□□□□ □□ □□□□ □□□□. □□□ □□□□□ □□□□ □□ □□□ □□□□ □□ □□□□ □□ □□□ □□□ □□ □□□□□ □□□□□.
- □□□ □□□□□ □□ □□□ □□□□□□□.

SQL Warehouse Overview



*All clusters in a warehouse should be of same size
X-Small or Medium or Large ...

Warehouse cluster size(Small,Medium ... No of worker nodes) can be changed anytime but a restart is required

*All worker nodes in Classic warehouse use Standard_E8ds_v4

*Driver size changes(bigger virtual machine) as cluster size(Small,Medium.. No of worker nodes) increases

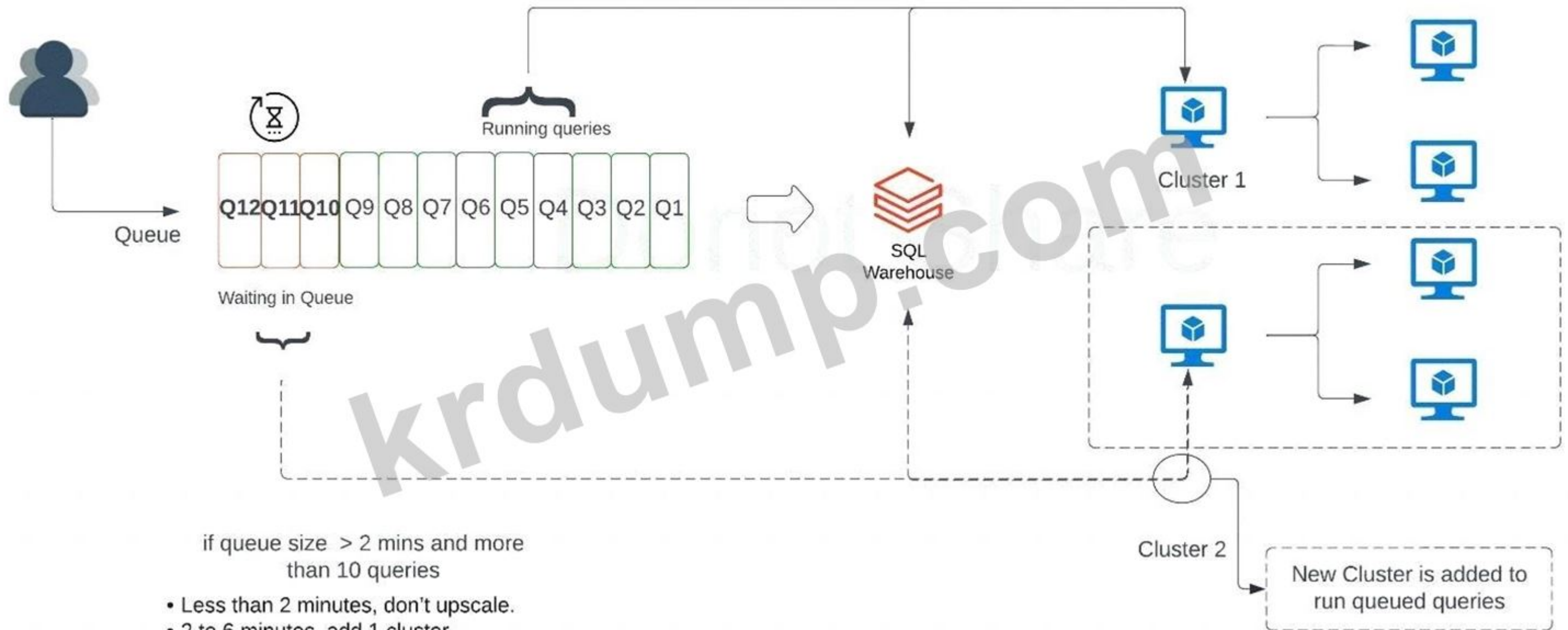
SQL Warehouse (SQL Warehouse) is a managed service that allows you to run SQL queries on your data. It is designed to be easy to use and scale.

SQL Warehouse -> SQL queries are executed on a cluster of worker nodes. The cluster size can be changed to match the workload.

Databricks SQL Warehouse is a managed service that allows you to run SQL queries on your data. It is designed to be easy to use and scale. The warehouse can be configured with a minimum of 1 cluster and a maximum of 40 clusters. The cluster size can be changed to match the workload.

SQL Warehouse Scale out

Scaling: Min 1 Max 3
Cluster Size: XSmall



if queue size > 2 mins and more than 10 queries

- Less than 2 minutes, don't upscale.
- 2 to 6 minutes, add 1 cluster.
- 6 to 12 minutes, add 2 clusters.
- 12 to 22 minutes, add 3 clusters.

* Please note 2 mins wait is only for classic warehouses for server less the wait can be much less.


databricks

Cluster size (2X-Small...4XLarge) SQL Endpoint(SQL Warehouse) Auto stop After 60 minutes of inactivity. Scaling Min. 1 Max. 2 clusters (6 to 12 size)

Starter Endpoint

Name

Starter Endpoint

Cluster size ⓘ

X-Small

6 DBU / cluster

Auto stop

After 60 minutes of inactivity

Scaling ⓘ

Min. 1

Max. 2

clusters (6 to 12 size)

Cluster size (2X-Small...4XLarge) SQL Endpoint(SQL Warehouse) Auto stop After 60 minutes of inactivity. Scaling Min. 1 Max. 2 clusters (6 to 12 size)



NEW QUESTION: 52

□□ □□□ □□□□ activity_details □□ Delta □□□□ CHECK □□ □□□ □□□□□ □□□□□□□□.

```
ALTER TABLE activity_details
```

```
ADD CONSTRAINT valid_coordinates
```

```
CHECK (
```

```
latitude >= -90 AND
```

```
longitude <= 90 AND
```

```
longitude >= -180 AND
```

```
longitude <= 180);
```

□□ □□□ □□□□ □ □□□□ □□□□□ □□□□ □□□, □□□□ □□ = 45.50, □□ = 212.67 □ □□□□ □□□□□□.

Which of the following is a valid Delta Lake constraint?

- A. The `valid_coordinates` column must be of type `Boolean`.
- B. The `valid_coordinates` column must be of type `String`.
- C. The `valid_coordinates` column must be of type `Integer`.
- D. The `valid_coordinates` column must be of type `Boolean`.
- E. The `valid_coordinates` column must be of type `String`.

Answer: (SHOW ANSWER)

CHECK is a Delta Lake constraint that can be used to enforce data integrity. It is a column constraint that can be used to enforce data integrity. Delta Lake ACID transactions are supported. ACID transactions are supported. Delta Lake ACID transactions are supported. Delta Lake ACID transactions are supported. Delta Lake ACID transactions are supported.

* `valid_coordinates`: <https://docs.delta.io/latest/delta-constraints.html>

* ACID transactions: <https://docs.delta.io/latest/delta-intro.html#acid-transactions>

NEW QUESTION: 53

Which of the following is a valid Delta Lake constraint? validation_copy is a valid Delta Lake constraint. validation_copy is a valid Delta Lake constraint. validation_copy is a valid Delta Lake constraint. validation_copy is a valid Delta Lake constraint.

DLT is a Delta Lake constraint that can be used to enforce data integrity. DLT is a Delta Lake constraint that can be used to enforce data integrity. DLT is a Delta Lake constraint that can be used to enforce data integrity. DLT is a Delta Lake constraint that can be used to enforce data integrity.

- A. The `validation_copy` column must be of type `Boolean`.
- B. The `validation_copy` column must be of type `String`.
- C. The `validation_copy` column must be of type `Integer`.
- D. The `validation_copy` column must be of type `Boolean`.

Answer: D (LEAVE A REPLY)

validation_copy is a Delta Lake constraint that can be used to enforce data integrity. validation_copy is a Delta Lake constraint that can be used to enforce data integrity. validation_copy is a Delta Lake constraint that can be used to enforce data integrity. validation_copy is a Delta Lake constraint that can be used to enforce data integrity. validation_copy is a Delta Lake constraint that can be used to enforce data integrity.

:

Databricks Delta Live Tables is a Delta Lake constraint that can be used to enforce data integrity. Databricks Delta Live Tables is a Delta Lake constraint that can be used to enforce data integrity. Databricks Delta Live Tables is a Delta Lake constraint that can be used to enforce data integrity. Databricks Delta Live Tables is a Delta Lake constraint that can be used to enforce data integrity.

NEW QUESTION: 54

Which of the following is a valid Delta Lake constraint? AnalysisException: cannot resolve 'heartrateheartrateheartrate' is a valid Delta Lake constraint. AnalysisException: cannot resolve 'heartrateheartrateheartrate' is a valid Delta Lake constraint. AnalysisException: cannot resolve 'heartrateheartrateheartrate' is a valid Delta Lake constraint.

- A. DataFrame is a valid Delta Lake constraint.
- B. AnalysisException is a valid Delta Lake constraint.
- C. heartrateheartrateheartrate is a valid Delta Lake constraint.
- D. AnalysisException is a valid Delta Lake constraint.

Answer: C (LEAVE A REPLY)

* `heartrateheartrateheartrate`: "select() is a valid Delta Lake constraint." is a valid Delta Lake constraint. PySpark DataFrame select; is a valid Delta Lake constraint.

NEW QUESTION: 55

Which of the following is a valid Delta Lake constraint? Delta Lake ACID transactions are supported. Delta Lake ACID transactions are supported. Delta Lake ACID transactions are supported. Delta Lake ACID transactions are supported.

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

Answer: C (LEAVE A REPLY)

11

□□□ □□□ □□ □□□ □□□ □□□□□ □□□□.

□□ □□□ □□□□ DLT □□□□□□□ □□□ □□□□ □□□□ □□ □ □□ □□□ □□□□ □□□□□□. □□ □□ □□□ □□□□ □□□ □□□□ □□□□□□. (□□□□□□ □□□ □□ □□□ □□□□ □□ □□□□□□□.)



databricks

```
%sql
CREATE OR REPLACE LIVE TABLE orders_valid
(CONSTRAINT valid_timestamp EXPECT (timestamp > '2020-01-01'))
SELECT * FROM LIVE.orders_vw

%python
@dlt.expect("valid timestamp", "timestamp > '2012-01-01'")
```

Retain Invalid Records

Records violate expectations are
Added to target table
Flagged in invalid in event log
Pipeline continues

```
%sql
CREATE OR REPLACE LIVE TABLE orders_valid
(CONSTRAINT valid_timestamp EXPECT (timestamp > '2020-01-01') ON VIOLATION DROP ROW)
SELECT * FROM LIVE.orders_vw

%python
@dlt.expect_or_drop("valid timestamp", "timestamp > '2012-01-01'")
```

Drop Invalid Records

Records violate expectation are
Dropped from target table
Flagged invalid in event log
Pipeline Continues

```
%sql
CREATE OR REPLACE LIVE TABLE orders_valid
(CONSTRAINT valid_timestamp EXPECT (timestamp > '2020-01-01') ON VIOLATION FAIL)
SELECT * FROM LIVE.orders_vw

%python
@dlt.expect_or_fail("valid timestamp", "timestamp > '2012-01-01'")
```

Fail on Invalid Records

Records violate expectation
Cause the job to fail

□□□□ □□ □□□:

□□□ □□ □□□□ □□□□□ `expect` □□□□ □□□□□. □□□ □□ □□□□ □□□ □□□□ □□ □□ □□□ □□□ □□□□□.

SQL

CONSTRAINT valid_timestamp EXPECT (timestamp > '2020-01-01')

□□□□ □□ □□□□ □□□□□.

`expect` □□ `drop` □□□□ □□□□ □□□□ □□ □□□□ □□□ □□□□ □□□□□□. □□□ □□ □□□□ □□ □□□ □□□□ □□□□□□.

SQL

□□ □□ valid_timestamp EXPECT (timestamp > '2020-01-01') ON VIOLATION DROP ROW □□□□ □□ □□□□ □□ □□ □□:

□□□□ □□ □□□□ □□□□ □□ □□, expect □□ fail □□□□ □□□□ □□□ □□□ □□□ □□□□ □□ □□□ □□□□□□. □□□ □□□ □□□□□□ □□ □□□□ □□□□□□ □□□□□□ □□□□□□.

SQL

□□ □□ valid_timestamp □□□(timestamp > '2020-01-01') □□ □ □□□□ □□

NEW QUESTION: 59

Delta Live Tables□ □□□□ □□□ □□□ □□□□□□ □□ □□ □□ □□□□ □□□□.

1. CONSTRAINT valid_timestamp EXPECT (timestamp > '2020-01-01')

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

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

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

NEW QUESTION: 60

□□□ □□□□□ databricks.yml □□□ □□ □□□ □□□□□ Databricks Asset Bundle□ □□□□ □□□□ □□□□□□□□ □□□□□□□□. □□ □□□ □□□□ □□ □□□□ □□ □□□□ □□□ my_project_job□□□□ □□□ □□□ □□□□ □□□□. □□□□ □□ □□□□□□□ □□□□□, □□□□ □□□ □□□ □□□□□□ □□□□ □□□ □□□□ □□□□ □□□□ □□□□ □□□□ □□□□ □□□□□□□?

- A. databricks□□ my_project_job -e prod□ □□□□□□.
- B. databricks □□ □□ my_project_job --env prod
- C. databricks run my_project_job -t prod
- D. databricks bundle run my_project_job -t prod

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

Databricks □□□ □□□□ □□□□ □□□ □□□ □□□□ □ □□□□□□ □□□ □□:

Databricks Asset Bundles(DAB)□ □□□□ □□ □□□ □□ □□, □□□□□□ □ □□□□□□ □□ Databricks □□□□□ □□□□□□ □□□□□ □□□□ □ □□□□□.

□□ □□□□□, `databricks bundle run` □□□□ □□□□□ □□ □□ □□□□□□□ □□□□ □□□□ □ □□□□, □□□□ □□ □□ □□□□ □□□□ □□ □□□□ □□(□: □□□□□, □□□□□ □□ □ □□ □□ URL) □ □□□□□□.

-t □□□□□ □□ □□(□: dev, staging □□ prod)□ □□□□□□□. □□□□ □□ databricks.yml □□□□□ □□ □□□ □□□ □□□ □□□□ □□□□ □□□□□□□.

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

NEW QUESTION: 61

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

- A. notebook.utils.run("□□□ □□□□ □□")
- B. execute.utils.run("□□□ □□□□ □□")
- C. dbutils.notebook.run("□□□ □□□□ □□")

D. □□□□ □□ □□□□□□□ □□□ □ □□□□.

E. spark.notebook.run("□□ □□□ □□")

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

□□

□□□ dbutils.notebook.run("□□ □□□ □□")□□□.

□□□ □□ □□□ □□□ □□ □□□□□□.

run(path: String, timeout_seconds: int, arguments: Map): String

1.dbutils.notebook.run("ful-notebook-name", 60, {"argument": "data", "argument2": "data2", ...})

Databricks-Certified-Professional-Data-Engineer □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ Databricks-Certified-Professional-Data-Engineer □□! DumpTop □ □□ **Databricks-Certified-Professional-Data-Engineer** □□ □□□ □□□□□□, DumpTop Databricks-Certified-Professional-Data-Engineer □□ □□□ □□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □□ DumpTop Databricks-Certified-Professional-Data-Engineer □□□ □□□□□. <https://www.dumptop.com/Databricks/Databricks-Certified-Professional-Data-Engineer-dump.html> (217 Q&As Dumps, **30%OFF Special Discount: KrDump**)

NEW QUESTION: 62

□□□ □□□□□ □□□□ □□ □□□□ □□□□ □□ Lakeflow □□□ □□□□□□ □□□□ □□□□. □ □□□□□□□ Auto Loader□ □□□□ □□□□ □□□□, customer_id□ amount □□ 0□ □ □□ □□□□ □□□ □□□ □□□□ □□□. □□□□ □□ □□□□ □□□□ □□□.

Lakeflow Declarative Pipelines □□ □ □□ □□□ Python□ □□□□ □ □□ □□□ □□□□□?

A. @dlt.table

def silver_orders():

□□ (

dlt.read_stream(" bronze_orders ")

.expect_or_drop(" valid_customer " , " customer_id IS NOT NULL ")

.expect_or_drop(" valid_amount " , " amount > 0 ")

)

B. @dlt.table

@dlt.expect(" valid_customer " , " customer_id IS NOT NULL ")

@dlt.expect(" valid_amount " , " amount > 0 ")

def silver_orders():

dlt.read_stream(" bronze_orders ")□ □□□□□□.

C. @dlt.table

def silver_orders():

□□ (

dlt.read_stream(" bronze_orders ")

.expect(" valid_customer " , " customer_id IS NOT NULL ")

.expect(" valid_amount " , " amount > 0 ")

)

D. @dlt.table

@dlt.expect_or_drop(" valid_customer " , " customer_id IS NOT NULL ")

```
dlt.read_stream(" bronze_orders ")
```

Answer: A (LEAVE A REPLY)

Lakeflow Declarative Pipelines(LDP), □□ Delta Live Tables(DLT)□ □□□□ □□□□□□□□ □□□ □□□□□ □□□ □□□□□□. □□□□□ □□ □ □□ □□ □ □□□□□.

* □□ □□(□□) # □□□ □□□□ □□ □□□□ □□□□ □□□□ □□□□ □□□□ □□ □□□□□□.

* □□ □□(expect_or_drop) # □□□ □□□□ □□ □□□□ □□ □□□□□ □□□□□.

* □□ □ □□□□□ □□(expect_or_fail) # □□ □□□ □□□□□□ □□□□ □□□□□□.

[illegible]

```
0) □ □□□□ □□□. □□ □□□ □□□ □□□ □□□ □□□□ □□□ .expect_or_drop("expectation_name", "SQL_predicate") □ □□□□ □□□□.
```

* □□ A□ □□□□□. □ □□ □□□ □□ □□ □□ □□□ .expect_or_drop□ □□ □□□□ □□□ □□□□ □□ □□□□ □□□□ □□□□.

* □□ B□ @dlt.expect □□□□□□ □□ □□□□ □□□, □ □□□□□ □□ □□□ □□□ □ □□□□ □□ □□ □□□□□ □□□□.

* □□ C□ .expect□ □□□□□, □ □□ □□ □□□□ □□ □□□ □□□□ □□□□□.

* `□□ D□ Python API□□ □□□□ □□ @dlt.expect or drop □□□□□ □□□ □□□□□. expect or drop□ □□□□□□ □□ DataFrame□ □□□□ □□□□ □□□□.`

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

□□: Databricks Lakeflow □□□ □□□□□ □□ - □□□(expect, expect_or_drop, expect_or_fail)

NEW QUESTION: 63

□□□□ □□□□ □□ □□ □□ □□ □□ □□□□ □□□□□?

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

B. Gold ☐ ETL ☐☐☐ ☐☐☐☐ ☐☐, Silver ☐ ☐☐ ☐☐ ☐☐☐ ☐☐.

C. □□ □□□ □□□□□, □□ □□ □□□□□□.

D. □□ □□ □□□, □□ □□□□ □□□□ □□□□.

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

Answer: A (LEAVE A REPLY)

11

□□□□ □□□□ - □□□□□□

□ □ □ :

1. □□□□ □□□□□□, □□□, □□□□, □□ □□ □□□ □□□□□.

2. □□□ □□ □□ □□□□ □□ □□□□□ □□□ □□

3. □□ □□□□ □□□□ □□□ □□□□□.

4.

□□ □□ □□: □□ □□□□仔□ □□□□ □□□□ □□□□ □ □(□□, □, □)□ □□□ □□□□□□. □ □□ □□□□ □□□ □□ □□□ □□□ □□□ □□□□.

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

□□ □□ □□ (□□□ □□)



A. □□ □□□□ □□ 3 □□□□ □□□□□□, □□ □□ □□ □□ □□ □□□□ □□ □□□□□.

B. □□ □□□□ □□ 2 □□□□ □□□□□□. □□ □□ □□□□□ □ □□ □□□□ □□ □□□ □□□□ □ □□ □□□□□.

C. □□ □□□□ Type 0 □□□□ □□□□ □□□, □□ □□ □□□ □□ □□ □□□□ □□ □□□ □□ □□□□ □□□□□ □□□□□.

D. □□ □□□□ Type 1 □□□□ □□□□ □□□, □□ □□ □ □□□ □□□□□□ □□□ □□□□ □□□□.

E. □□ □□□□ Type 2 □□□□ □□□□ □□□, □□ □□ □□□□□□ □□ □□□ □□□□□.

□ □□□ `MERGE INTO` □□□□ □□□□ □ □□□□□□ □□□ □ □□□□ `customers` □□□□ □□□□□. `MERGE INTO` □□□□ □□ □□□□ □□ □□□ □□ □, □ □□ □□□ □□□ □. □□ □□□ □□ □□ □□□ □□ □□□ □□ □□□ □□ □□□ □□□□□. □ □□ □□□ `customers` □□□□ □□ □□ `customer\_id` □ □□□□ □□□□ □□□□□ □□□□□. □□□ □□ □□□□ □□ □□□□ □□ □□□□ □ □□□ □□□□□□ `current\_flag` □ `false` □ □□□□ □□ □□□□ □ □□ □□□ □□□ □□□□ □□□□. □□ □□ □□□□ □□ □□□□ □ □□ □□□□ □□□□ □□□□ □□□□ `current\_flag` □ `true` □ □□□□ □□ □□□□ □□□□ □□□□ □□□□. □, □□ □□ □□□□□ □ □□ □□□ □□ □□□ □□□□ □ □□ □□□□□, □□ □□ 2 □□ □□ □□□ □□□□□. □□□ □□ □□: [Databricks Certified Data Engineer Professional], "Delta Lake" □□; Databricks □□□□ "Databricks □ Delta Lake □□" □□□□ □□□□□□.

[illegible][illegible]

Delta Live Tables는 데이터 스트림을 처리하는 데 사용되는 테이블로, 테이블의 데이터가 변경될 때 자동으로 업데이트됩니다. CON-CONSTRAINT valid_timestamp EXPECT (timestamp > '2020-01-01') 제약 조건을 사용하여 테이블의 데이터가 2020-01-01 이후에 생성된 것임을 보장합니다. 이 제약 조건을 사용하여 테이블의 데이터를 필터링하는 방법은 무엇입니까?

- A. 테이블의 데이터를 필터링하는 쿼리를 작성하여 timestamp > '2020-01-01' 조건을 사용합니다.
- B. 테이블의 데이터를 필터링하는 쿼리를 작성하여 timestamp < '2020-01-01' 조건을 사용합니다.
- C. 테이블의 데이터를 필터링하는 쿼리를 작성하여 timestamp > '2020-01-01' 조건을 사용합니다.
- D. 테이블의 데이터를 필터링하는 쿼리를 작성하여 timestamp < '2020-01-01' 조건을 사용합니다.
- E. 테이블의 데이터를 필터링하는 쿼리를 작성하여 timestamp > '2020-01-01' 조건을 사용합니다.

Answer: ([SHOW ANSWER](#))

정답

정답은 C입니다. Delta Live Tables는 데이터 스트림을 처리하는 데 사용되는 테이블로, 테이블의 데이터가 변경될 때 자동으로 업데이트됩니다. CON-CONSTRAINT valid_timestamp EXPECT (timestamp > '2020-01-01') 제약 조건을 사용하여 테이블의 데이터가 2020-01-01 이후에 생성된 것임을 보장합니다. 이 제약 조건을 사용하여 테이블의 데이터를 필터링하는 방법은 무엇입니까?

정답은 C입니다. Delta Live Tables는 데이터 스트림을 처리하는 데 사용되는 테이블로, 테이블의 데이터가 변경될 때 자동으로 업데이트됩니다. CON-CONSTRAINT valid_timestamp EXPECT (timestamp > '2020-01-01') 제약 조건을 사용하여 테이블의 데이터가 2020-01-01 이후에 생성된 것임을 보장합니다. 이 제약 조건을 사용하여 테이블의 데이터를 필터링하는 방법은 무엇입니까? (정답은 C입니다.)



databricks

```
%sql
CREATE OR REPLACE LIVE TABLE orders_valid
(CONSTRAINT valid_timestamp EXPECT (timestamp > '2020-01-01'))
SELECT * FROM LIVE.orders_vw

%python
@dlt.expect("valid timestamp", "timestamp > '2012-01-01'")
```

Retain Invalid Records

Records violate expectations are
Added to target table
Flagged in invalid in event log
Pipeline continues

```
%sql
CREATE OR REPLACE LIVE TABLE orders_valid
(CONSTRAINT valid_timestamp EXPECT (timestamp > '2020-01-01') ON VIOLATION DROP ROW)
SELECT * FROM LIVE.orders_vw

%python
@dlt.expect_or_drop("valid timestamp", "timestamp > '2012-01-01'")
```

Drop Invalid Records

Records violate expectation are
Dropped from target table
Flagged invalid in event log
Pipeline Continues

```
%sql
CREATE OR REPLACE LIVE TABLE orders_valid
(CONSTRAINT valid_timestamp EXPECT (timestamp > '2020-01-01') ON VIOLATION FAIL)
SELECT * FROM LIVE.orders_vw

%python
@dlt.expect_or_fail("valid timestamp", "timestamp > '2012-01-01'")
```

Fail on Invalid Records

Records violate expectation
Cause the job to fail

NEW QUESTION: 70

□□ □ Silver □□□□□ Gold □□□□□ □□ □□□□□ □□□□ □□□□ □□□□ □□□□□?

A. 1.(spark.table("sales"))

2..groupBy("store")

```
3..agg(sum("sales"))
```

4..writeStream

```
5..□□("checkpointLocation", checkpointPath)
```

```
6..outputMode("complete")
```

```
7..table("aggregatedSales") )
```

 $(\square\square)$

B. 1.`(spark.table("sales"))`

2..writeStream

```
3..□□("□□□□□ □□", □□□□□ □□)
```

```
4..outputMode("complete")
```

```
5..□□□("□□□") )
```

C. 1.(spark.table("sales"))

```
2..withColumn("avgPrice", col("sales") / col("units"))
```

3..writeStream

```
4..□□("checkpointLocation", checkpointPath)
```

```
5..outputMode("append")
```

```
6..table("cleanedSales") )
```

D. 1.(spark.readStream.load(rawSalesLocation))

2..writeStream

```
3..□□("□□□□□ □□", □□□□□ □□)
```

```
4..outputMode("append")
```

```
5..table("uncleanedSales") )
```

E. 1.(spark.read.load(rawSalesLocation))

2. .writeStream

```
3. .option("checkpointLocation", checkpointPath)
```

```
4. .outputMode("append")
```

```
5. .table("uncleanedSales") )
```

Answer: A (LEAVE A REPLY)

11

33

```
1.(spark.table("sales"))
```

```
2..groupBy("store")
```

```
3..agg(sum("sales"))
```

4..writeStream

```
5..□□("checkpointLocation", checkpointPath)
```

```
6..outputMode("complete")
```

```
7..table("aggregatedSales") )
```

□□ □□□□ □□□□□ □□□ □□□□ □□□□ □ □□□□□.

□□□ □□□ □□ □□□ □□□□□□.

□□□□ □□□□ - □□□□□□

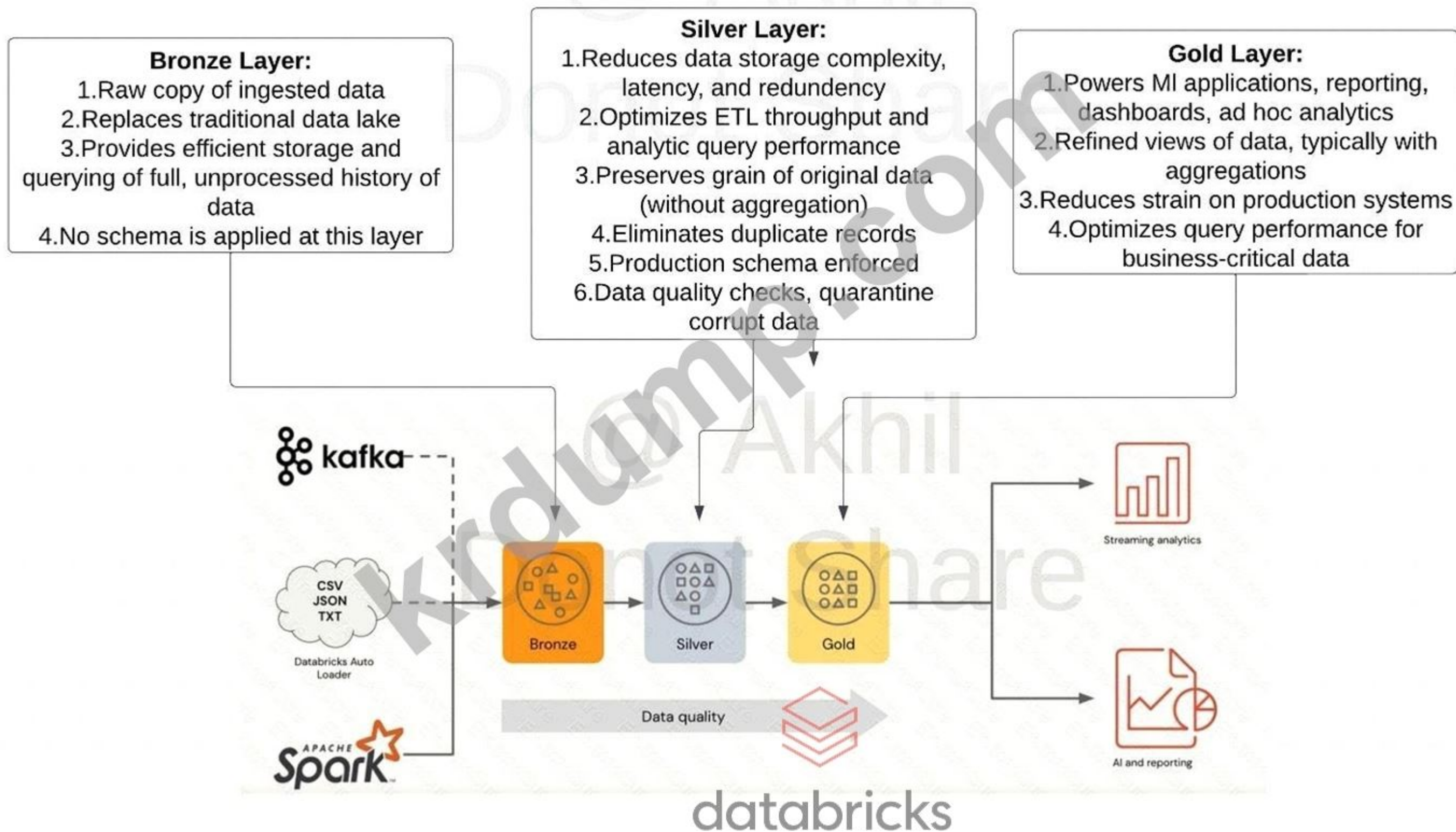
□ □ □ :

1. □□□□ □□□□□□, □□□, □□□□, □□ □□ □□□□□□□□.
2. □□□ □□ □□ □□□□ □□ □□□□ □□□ □□
3. □□ □□□□ □□□□ □□□ □□□□□□.
4. □□□□ □□ □□□□ □□ □□ □□□ □□□□□□□.

□□ □□ □□: □□ □□□□仔□ □□□□ □□□□ □□□□ □ □(□□, □, □)□ □□□ □□□□□□. □ □□ □ □□□ □□□ □□ □□□ □□□ □□□ □□□□.

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

□□ □□ □□ (□□□ □□)



□□□ □□□□□ Lakeflow Declarative Pipelines□ Expectations □□□ □□□□ □□□□ □□ □□□□ □□□ □□□□ □□□□. □□□□ □□□□□ □□□ □□□ □□□ □□□□ □□□□□, □ □□ □□ □□ □□ □□□□ □□□□ □□ □□□□ □□ □□ □□□□ □□□□ □□□□. □□□□ □□□□ □□ □□□□□. □□□□ □□ □□□ □□ □□□□ □□□□ □ □□ □□ □□□□ □ □□□□ □□□□ □□□□ □□□□ □□□□.

□□□ □□□ □□ □□□□ □□ □□ □□□□□:

@dlt.table

@dlt.expect("valid_sensor_reading", "reading < 120")

def silver_sensor_readings():

spark.readStream.table("bronze_sensor_readings")□ □□□□□.

□□ □□□ □□ □□□ □□□□□?

A. @dlt.table

@dlt.expect("valid_sensor_reading", "reading < 120")

def silver_sensor_readings():

spark.readStream.table("bronze_sensor_readings")□ □□□□□.

@dlt.table

@dlt.expect("invalid_sensor_reading", "reading >= 120")

def quarantine_sensor_readings():

spark.readStream.table("bronze_sensor_readings")□ □□□□□.

B. @dlt.table

@dlt.expect_or_drop("valid_sensor_reading", "reading < 120")

def silver_sensor_readings():

spark.readStream.table("bronze_sensor_readings")□ □□□□□.

@dlt.table

@dlt.expect("invalid_sensor_reading", "reading < 120")

def quarantine_sensor_readings():

spark.readStream.table("bronze_sensor_readings")□ □□□□□.

C. @dlt.table

@dlt.expect_or_drop("valid_sensor_reading", "reading < 120")

def silver_sensor_readings():

spark.readStream.table("bronze_sensor_readings")□ □□□□□.

@dlt.table

@dlt.expect_or_drop("invalid_sensor_reading", "reading >= 120")

def quarantine_sensor_readings():

spark.readStream.table("bronze_sensor_readings")□ □□□□□.

D. @dlt.table

@dlt.expect_or_drop("valid_sensor_reading", "reading < 120")

def silver_sensor_readings():

spark.readStream.table("bronze_sensor_readings")□ □□□□□.

@dlt.table

@dlt.expect("invalid_sensor_reading", "reading >= 120")

def quarantine_sensor_readings():

spark.readStream.table("bronze_sensor_readings")□ □□□□□.

0000 000000. 00000000 0000 000000 00 00000, 00 00 0 000000 0000 000000. 00000000 000000 0000 00 0000 00 0000 DBFS 00 S3 00 00000
 000000 000000 0000000. 0 000000 0000 00 000000 000000 0000 00 0000000 0000000 0000 0000. 0 00 000000 0000 0000 0000000 0000000 000000 00 00
 000000 0000 00 0000 0000 0000 0000 0 000000. 0000 00 00: [Databricks Certified Data Engineer Professional], "0000 00000" 00; Databricks 00, "0000000" 00.

NEW QUESTION: 76

SQL DDL □□ `CREATE TABLE sales (id int, units int)` □ □□□ □□□□ □□□□ □□□ □□□□□?

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

Answer: (SHOW ANSWER)

11

□ □ □ □ □ □ □ □ □ □ □ □ □ □ .

Location □□□ □□ □□□□ □□□□ □□□□□ □□ □□□ □□□□ □□□ □□□□□. □□ □□□ □□: CREATE TABLE table name (column column data type...)

Databricks-Certified-Professional-Data-Engineer □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ Databricks-Certified-Professional-Data-Engineer □□! DumpTop □ □□ **Databricks-Certified-Professional-Data-Engineer** □□ □□□ □□□□□□, DumpTop Databricks-Certified-Professional-Data-Engineer □□ □□□ □□□□□□□ □□□ □□□□□□□. □□□□ □□□ □□□□ □□ DumpTop Databricks-Certified-Professional-Data-Engineer □□□ □□□□□. <https://www.dumptop.com/Databricks/Databricks-Certified-Professional-Data-Engineer-dump.html> (217 Q&As Dumps, **30%OFF Special Discount: KrDump**)

NEW QUESTION: 77

[illegible]

- A. □□□ □□□□ □□□□ □□ □□ □□□ □□□□ □□
- B. □□□ □□□□ □□□□ □□ □□□□□ □□□□□ □□□□ □□
- C. □□□□ □□□□ □□ □□□□ Lakehouse□ □□□□ □□
- D. □□□□ □□□□□□□ □□ □□ □□□ □□□□ □□
- E. □□□□ □□□□□□ □□□ □□ □□ □□□□ □□□□ □□□□ □□□□ □□□□.

Answer: C (LEAVE A REPLY)

11

☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐.

[illegible]

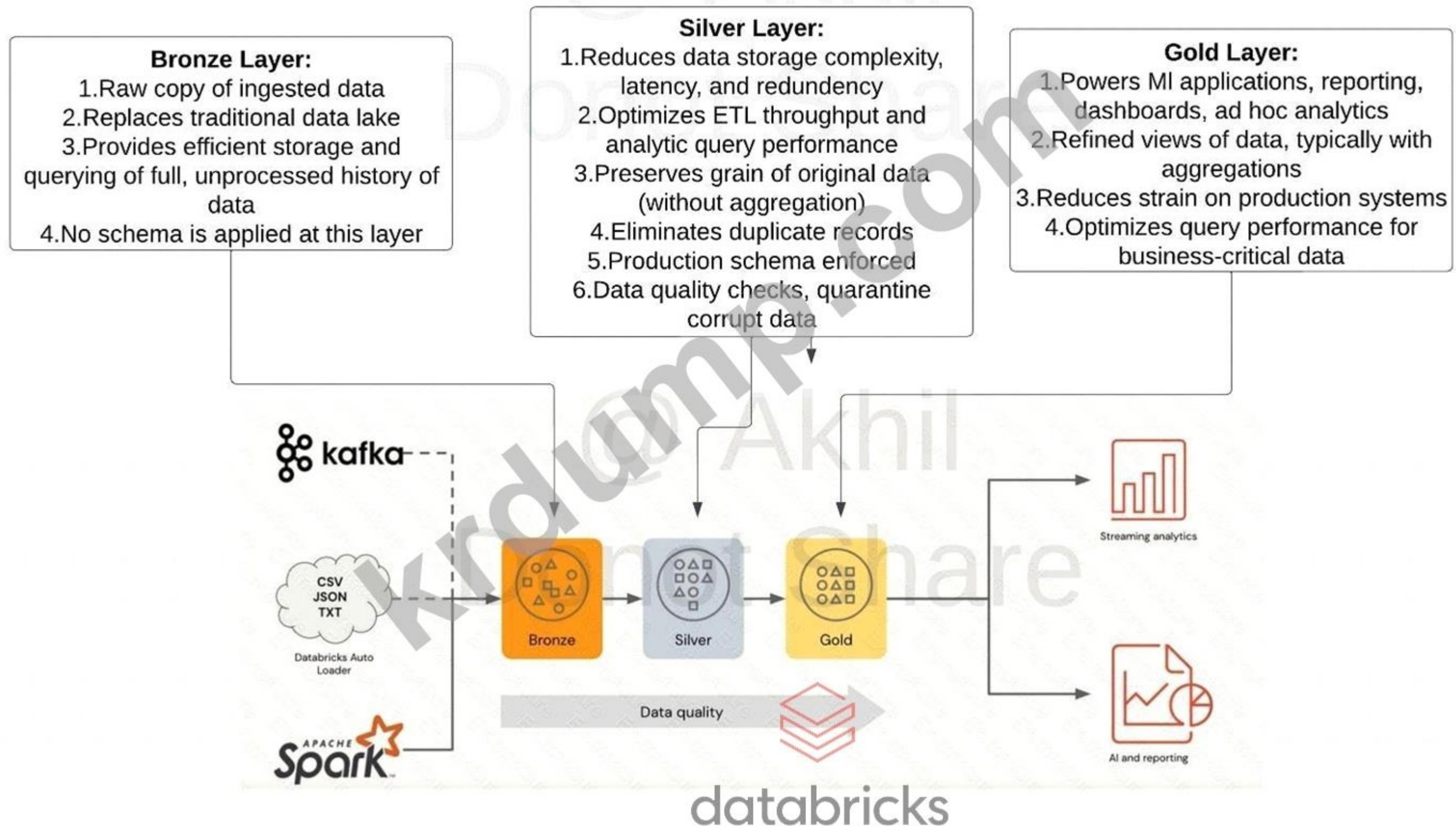
□ □ □ □ □ □ □ □ - □ □ □ □ □ □ □ □

□ □ □ :

1. □□□ □□□□ □□ □□
2. □□ □□□ □□□□ □□□□□.
3. □□□□ □□ □□ □□□ □□□ □□□□□ □□□□ □□□ □ □□□□.
4. □ □□□□□ □□□□ □□□□ □□□□.

□□ □□ □□: □□ □□□□仔□ □□□□ □□□□ □□□□ □ □(□□, □, □)□ □□□ □□□□□□. □ □□ □ □□□ □□□ □□ □□□ □□□ □□□ □□□□.

□□□□□ □□□□□ □□□□ □□□□. □□□□□ □□ □□□□ □ □□□□ □□□□ □□□□.
□□□□ □□□□ □ □□□□ □□□ □□□□□?




```
from pyspark.sql.functions import current_timestamp, input_file_name, col
from pyspark.sql.column import Column
```

```
def ingest_daily_batch(time_col: Column, year:int, month:int, day:int):
    (spark.read
     .format("parquet")
     .load(f"/mnt/daily_batch/{year}/{month}/{day}")
     .select("time_col.alias('ingest_time'),
            input_file_name().alias('source_file')")
     .write
     .mode("append")
     .saveAsTable("bronze"))
```

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

□□ □□ □□ □□ □□ □□ □□ □□□□□□?

def new_records():

A. spark.readStream.table("bronze") □ □ □ □ □ □.

B. spark.readStream.load("bronze") □ □ □ □ □ □.

C.

```
return (spark.read
        .table("bronze")
        .filter(col("ingest_time") == current_timestamp()))
```

D. spark.read.option("readChangeFeed", "true").table("bronze") □ □ □ □ □ □.

E.

```
return (spark.read
        .table("bronze")
        .filter(col("source_file") == f"/mnt/daily_batch/{year}/{month}/{day}")
        )
```

Answer: E (LEAVE A REPLY)

<https://docs.databricks.com/en/delta/delta-change-data-feed.html>

NEW QUESTION: 81

```
dbutils.secrets.get("secrets","API_KEY")
```

NEW QUESTION: 83

□□□ □□□□ □□ □□□□ Databricks Lakehouse□ □□□□ □□□□ □□□ □□□ □□□□ □□ □□□□ □□□ □ □□ □□ □□□ □□□□□ □□□□□□. □□ □□□□□ □□□ □□ □□□□□□□ □□ □□□ □□ □□□ □□□□ □□ □□□ □□ □□□ □□□□□□□. prod □□□□□□□ □□ □□ □□□ eng□ □□□□□.

GRANT SELECT ON DATABASE prod TO eng;

eng □□□ □□□ □□□ □□□□□ □ □□□□□ □□□□□□ □□□□ □□□□ □□□ □, □□□ □□□ □□□□ □□□ □□□□□?

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

Answer: D (LEAVE A REPLY)

`GRANT USAGE ON DATABASE prod TO eng` □□□ eng □□□ prod □□□□□□□ □□□ □ □□ □□□ □□□□□. □, eng □□□ prod □□□□□□□ □□□□ □□ □□□□ □□□ □ □□□ □. `GRANT SELECT ON DATABASE prod TO eng` □□□ eng □□□ prod □□□□□□□ □□□□ □□□ □□□□ □□□ □ □□ □□□ □□□□□. □, eng □□□ SQL □□ DataFrame API□ □□ □□ □□□□ □□□ □ □□□□.

□□□ □□□ □□□ eng □□□ □□□ □ □□□, □□ □□ □□, □□□ □□ □□ □□□ □□ □□ □□□ □□□□ □□□□. □□□ eng □□ □□□□ prod □□□□□□□ □□ □□□□ □□ □□ □ □□□ □□□□□□□□ □□□□ □□□□□ □□□ □ □□□□. □□ □□:

* □□□□□□□ □□ □□ □□: <https://docs.databricks.com/en/security/auth-authz/table-acls/grant-privileged-database.html>

Hive □□□□□ □□□ □□□ □ □□ □□: [https://docs.databricks.com/en/security/auth-authz](https://docs.databricks.com/en/security/auth-authz/table-acls/privileges.html)

/table-acls/privileges.html

NEW QUESTION: 84

□□□ □□□□□ Databricks□ Jobs API□ □□□□ □□ □□□□ □ □□□ □□□□□□ □□□. □□ □□□ □□□□, □□□ □□□ □□□ □ □□ □□□□ □□□. □□□ □□□□□ □□ □□□ API □□□ □□□□ □□□?

- A. jobs/list □□□□□□ □□□□ □□ □□□ □□□□, jobs/runs/list□ □□□□ □□ □□ □□□ □□□□, jobs/run-now□ □□□□ □□□ □□□ □□ □□□□□.
- B. jobs/get □□□□□□ □□□□ □□ □□ □□□ □□□ □□ jobs/update□ □□□□ □□□ □□□ □□ □□□□□.
- C. jobs/list □□□□□□ □□□□ □□ □□□ □□□ □□, jobs/create □□□□□□ □□□□ □ □□□ □□□□, jobs/run-now□ □□□□ □ □□□ □□□□□.
- D. jobs/cancel □□□□□□ □□□□ □□□ □□□ □□□ □□ jobs/create□ □□□□ □ □□□ □□ □□□□ □□□□□□.

Answer: A (LEAVE A REPLY)

Databricks □□□ □□□□ □□□□ □□□ □□□ □□□□ □ □□□□□ □□□ □□:

Databricks Jobs REST API□ □□□□ □□ □□ □□□□□□ □□□□□. □□□ □□□□ □ □□□ □□□ □□ □ □□ □□ □□□ □□□□□.

GET /api/2.1/jobs/list - □□□□□□ □□ □□ □□ □□□ □□□ □□□□□.

GET /api/2.1/jobs/runs/list - □□ □□□ □□ □□ □□ □□□ □□ □□(□: □□□: □□)□ □□□□□.

POST /api/2.1/jobs/run-now - □□□ □□□ □□ □□ □□□□□.

□ □□□ Databricks□□ □□□□ □□ □□ □ □□ □□□ □□□ □□□□□. `jobs/update` □□□ □□□□□□□ □□□□□ □□□ □□ □□□□□ □□□, `jobs/create` □□□ □ □□□ □□□□ □ □□□□ □□□ □□□ □□ □□□□ □□ □□□□ □□□□. □□□ □□□□ □□ □□□□ □□ □□□□ □□□ □□□□□. □□□ □□ A□ □□□ □□□ □□ □□□□□□□.

NEW QUESTION: 85

□□ □□ □□□□ □□□□□ /mnt/finance_eda_bucket □□□ □□□□□□□□□.

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

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

□□□□ □□ □□□□ □□ □□□ □□□□ □□, tx_sales □□□□ □□□□ □□□ □□□□ □□□ □□□□□□?

A. □□ □□□□ Databricks □□□ □□□□□ Hive □□□□□□□ □□ □□□ □□□□□□.

B. /mnt/finance eda □□□ □□□□ □□□□ □□□□□ □□ □□□□ □□□□□□.

C. □□ □□□□ □□□ □□□□ Databricks □□ □□□□ Hive □□□□□□□ □□□□□□.

D. /mnt/finance eda □□□ □□□□ □□□□ □□□□□ □□□ □□□□ □□□□□□.

E. DBFS □□ □□□□ □□□□□□ □□□ □□□□ □□□□□□.

Answer: [\(SHOW ANSWER\)](#)
<https://docs.databricks.com/en/lakehouse/data-objects.html>

NEW QUESTION: 86

□□□ □□□□□□ □□□ □□ □□□ □□□□□□□.

1. □□ *

2. JSON.`/path/to/json/file.json` □□;

□□□ □□□□□□ □□□□ □ □□□ □□ □□□ □□□(DLT)□□ □□□ □ □□□ □□□□ □ □□□ □□□□□□.

□□□□□□. □□ □□□□ DLT □□□□□□□ □ □□ □□□□ □□□□ □□□.

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

A. JSON □□ □□□ cloud_files(...) □□□ □□□□ □□□.

B. □□ □□ □□□ `CREATE DELTA LIVE TABLE table\_name AS` □□ □□□□ □□□.

C. □□ □□ □□□ COMMENT □□ □□□□ □□□.

D. □□ □□ □□□ `CREATE LIVE TABLE table\_name AS` □□ □□□□ □□□.

E. FROM □□□ json □□ live. □□□□ □□□□ □□□.

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

NEW QUESTION: 87

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

Cmd 1

```
rawDF = spark.table("raw_data")
```



databricks

Cmd 2

```
rawDF.printSchema()
```

Cmd 3

```
flattenedDF = rawDF.select("*", "values.*")
```

Cmd 4

```
finalDF = flattenedDF.drop("values")
```

Cmd 5

```
finalDF.explain()
```

Cmd 6

```
display(finalDF)
```

Cmd 7

```
finalDF.write.mode("append").saveAsTable("flat_data")
```

□□□□ □□□□ □□□□ □□ □□ □□□□ □□□□ □□□?

- A. □□ 2
- B. □□ 3
- C. □□ 4
- D. □□ 5
- E. □□ 6

Answer: ([SHOW ANSWER](#))

□□ 6□ □□□□ □□□□ □□□□ □□ □□□□ □□ □□□□□. □ □□□ finalDF □□□□□□□□ □□ □□ □□□□ □□□□ □□□□□. finalDF □□□□□□□□ □□ □□ 7□□ □□□□ □□□□□□□ □ □□□ □□□□ □□□□. □□□□ □□□□□□□□ □□□□ □□ □□□□ □□□ □□ □□ □□□□ □□□ □□□ □□□□.

□□□ Cmd 6□ □□□□□□ □□□□ □□□.

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

* Cmd 1: raw_data □□□□ rawDF□□ Spark □□□□□□□□ □□□□□□.

* Cmd 2: rawDF □□□□□□□□ □□□□ □□□□□. □□ □□□ □ □□□ □□ □□□ □□□□□.

* Cmd 3: rawDF □□□□□□□□ □□ □□ values □□□□ □□ □□□ □□ □□ □□□□ flattenedDF□□ □ □□□□□□□□ □□□□□.

- * Cmd 4: flattenedDF `flatMap { (value, _) => Seq(value) }` values `flatMap { (value, _) => Seq(value) }` finalDF `flatMap { (value, _) => Seq(value) }`.
- * Cmd 5: finalDF `flatMap { (value, _) => Seq(value) }` `flatMap { (value, _) => Seq(value) }` `flatMap { (value, _) => Seq(value) }` `flatMap { (value, _) => Seq(value) }` `flatMap { (value, _) => Seq(value) }`.
- * Cmd 7: finalDF `flatMap { (value, _) => Seq(value) }` flat_data `flatMap { (value, _) => Seq(value) }` `flatMap { (value, _) => Seq(value) }` `flatMap { (value, _) => Seq(value) }` append `flatMap { (value, _) => Seq(value) }` `flatMap { (value, _) => Seq(value) }`.

NEW QUESTION: 88

Which of the following is the correct way to write a Spark SQL query to select all columns from a table named 'table' and filter the results to only include rows where the 'id' column is greater than 100?

- A. `SELECT * FROM table WHERE id > 100`
- B. `SELECT * FROM table WHERE id >= 100`
- C. `SELECT * FROM table WHERE id > 100 AND id < 200`
- D. `SELECT * FROM table WHERE id > 100`
- E. `SELECT * FROM table WHERE id > 100`

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

Which of the following is the correct way to write a Spark SQL query to select all columns from a table named 'table' and filter the results to only include rows where the 'id' column is greater than 100?

NEW QUESTION: 89

Which of the following is the correct way to write a Spark SQL query to select all columns from a table named 'table' and filter the results to only include rows where the 'id' column is greater than 100?

```
%sh
git clone https://github.com/foo/data_loader;
python ./data_loader/run.py;
mv ./output /dbfs/mnt/new_data
```

Which of the following is the correct way to write a Spark SQL query to select all columns from a table named 'table' and filter the results to only include rows where the 'id' column is greater than 100?

- A. `%sh git clone https://github.com/foo/data_loader; python ./data_loader/run.py; mv ./output /dbfs/mnt/new_data`
- B. `%sh git clone https://github.com/foo/data_loader; python ./data_loader/run.py; mv ./output /dbfs/mnt/new_data`
- C. `%sh git clone https://github.com/foo/data_loader; python ./data_loader/run.py; mv ./output /dbfs/mnt/new_data`
- D. Databricks `%sh git clone https://github.com/foo/data_loader; python ./data_loader/run.py; mv ./output /dbfs/mnt/new_data`
- E. `%sh git clone https://github.com/foo/data_loader; python ./data_loader/run.py; mv ./output /dbfs/mnt/new_data`

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

<https://www.databricks.com/blog/2020/08/31/introducing-the-databricks-web-terminal.html> is a new feature that allows you to run Spark SQL queries directly from the Databricks web interface. Which of the following is the correct way to write a Spark SQL query to select all columns from a table named 'table' and filter the results to only include rows where the 'id' column is greater than 100?

NEW QUESTION: 90

Which of the following is the correct way to write a Spark SQL query to select all columns from a table named 'table' and filter the results to only include rows where the 'id' column is greater than 100?

- A. `SELECT * FROM table WHERE id > 100`
- B. `SELECT * FROM table WHERE id >= 100`

C. ☐ ☐ ☐ ☐

D. ☐ ☐ ☐ ☐ ☐

E. ☐☐ ☐☐

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

0000 000000. 00000 0000 00000 000 00000 00 000000 0000 0 00 00 0000 00000 000000. 0000 0000 Databricks 000000000 00000 00 00000 00
 00 0 000000. 0000 00000 00, 00, 00, 0000 0 00 0000 00000. 00 0000 00, 00, 00, 00000, 00 0000 00000 00000 00 00 0000 000000. 00 0000
 00000 00000 0000 0 0000 0000 0000000 0000 00 00000. 00 0000 00000 0000 00 00000000 0000 0000 0 0000 0000000 0000 00 00000. 00 0000 0
 00 0000 0 0 0000 0000000 0000 00 00000. 0 00 000000 00 0000 00000 00000 00000 0000 0 0 00 0000 0000000 00000 00 00000 00000 0000 0
 00 0 00000. 0000 00 00: [Databricks Certified Data Engineer Professional], "Databricks Workspace" 00; Databricks 00, "Notebook permissions" 00.

NEW QUESTION: 91

□□□□ □□□ □□□ □□□□ □□□ □□ □□ □ □□□□ □□ □□□□ □□ □□□ □□□ □□□□. □□ □□□□ □□ □□□ □ □□□□□□ □□□ □□ □□□ 3□ □□□□□□. □□□ □□□□ □□ □□□□□ □ □□□□□□□ □□ □□□ □□ □□□□□□, □□□ 30□□ □□□□□ □□□. □□ □□□□ □□ □□□ 10□ □□□□□ □□□□□□ □□□□ □□□□.

□□ □□ □□□ □□□□ □□□□ □□□□ 10□ □□□ □□□□ □□□ □□□ □, □□ □□□ □□ □□□ □□□□□?

A. □□□ □□□ 5□□ □□□□. □□ □□□ □ □□ □□□□□ □□ □□□ □□ □□□ □ □□□ □□□□ □□ □□ □□□□ □□ □□ □□□ □□□ □ □□□□.

B. □□□ □□□ 30□□ □□□□□. □ □□□ □□ □□□ □□ □□ □□□ □□□ □□□ □□□ □□□ □□ □□□ □□□ □□□ □□□ □□ □□ □□ □□□□□.

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

[illegible]

E. 000 000 500 000000. 00 000 0 00 0000 00000 0000 00 00 000 00 0000 0000 00 000 0 00000.

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

□□□ □□□□□ □□ □□□□ □□□□ □□□□ □□□□ □□ □□ □□□ □□□□ □□ □□□ □□□□, □□□□ 10□ □□□ □□□□□ □□□□ □□□. □□□□ □□□ □□□□ □ □
□ □□□ □□ □□□ '□ □□ □□□' □□□□□.

□ □ □ □ □ □ □ □ :

□□ □□ □□□□(□□ □□): □□ 10□□ □□□ □□□ □□□□ □□ □□ □□□ □□ □□ □□□□ □□□ □□□ □□□ □ □□□□. □□ □□ □□□ □□□□ □□ □□□ □□□□ □□ □□□ □ □□□□.

[illegible]

Trigger Once ☐ ☐:

□□: □□□□□ □□□□ □□ Databricks □□□ □□□□ 10□□□ □□□□□ □□□ □ □□□□. □ □□ □□□ □□□ □□ □□□ □□□□ □□□□ □ □□ □□□□ □□ □□□□ □□ □□□ □□ □□□ □□□□, □□ □□□ □□□□ □□□□ □□□ □□□ □□□.

[illegible]

□□ □□□□ □□ □□ □□:

□□ A □ E(□□ □□): □□□ □□□ 5□□ □□□ □□ □□□ □□□ □□□□ □□ □□ □□ □□□ □□□□ □□□ □□□ □ □□□, □□□□□ □□□□□ □□□ □□ □□□ □□□ □ □□□□.

□□ B(□□ □□): □□□ □□□ 30□□ □□□ □□□ □□ □□□ □□□ □ □□□, □□ 10□ □□□ □□□□ □□□□ □□□ □□ □□□ □□□□□.

□□ C(□□□□ □□): □□ □□□□ □□ □□ □□□ □□□□□□ □□□□ □□□ □ □□□, □□ □□ □□□□□□ □□□□ □□□ □□□□ □□□ □□ □□ □ □□□ □□□ □□□ □□□ □ □□□□.

□ □ :

[illegible]

11

□□□□ □□□□ □□□□□ □□□ - □□□□

Databricks-Certified-Professional-Data-Engineer 00 0000 000000 00 DumpTop 00 0000 000 Databricks-Certified-Professional-Data-Engineer 00! DumpTop 0 00 **Databricks-Certified-Professional-Data-Engineer** 00 0000 000000, DumpTop Databricks-Certified-Professional-Data-Engineer 00 0000 0000000000 0000 00000000. 00000 0000 00000 00 DumpTop Databricks-Certified-Professional-Data-Engineer 0000 000000. <https://www.dumptop.com/Databricks/Databricks-Certified-Professional-Data-Engineer-dump.html> (217 Q&As Dumps, **30%OFF Special Discount: KrDump**)

NEW QUESTION: 92

user_ltv00 00000 00 00 0000 000000 0000 00 00000 0 000000. 000000000 000000 00000 00000 0000, 0000 0000 ACL0 00000 0000 00 0000 0000 0 0 000000.
user_ltv 00000 0000 00 00000 0000 000000.
0000 0000, 00 00, 00 00 00 00
00 0 0000 000000.

```
CREATE VIEW email_ltv AS
SELECT
CASE WHEN
  is_member('marketing') THEN email
ELSE 'REDACTED'
END AS email,
ltv
FROM user_ltv
```

- 0000 00 0000 00 000000 0000 00 0000 000000.
SELECT * FROM email_ltv
00 0 0 0000 0000 00 0 00000 0000 000000?
A. 0 00 00 000000, 00 0 00 "0000"0000 00000 00000 null 00 000000.
B. 00000 ITV 00 00000, 0000 0000 00 00 null0 000000.
C. 0000 0 LTV 0000 0000 ITV 00 000000.
D. 0000, 00, LTV 00 0000 LTV 00 00 000000.
E. 00000 LTV 00 00000, 0000 00 0 0000 "REDACTED" 00000 000000.

Answer: (SHOW ANSWER)

0 0000 `email\_ltv`00 00 00000 `user\_ltv` 0000000 `email`0 `ltv` 00 000000. `user\_ltv` 00000 00000 `email STRING`, `age INT`, `ltv INT`0000. 00, `CASE WHEN` 00000 00 00 00000 0000 0000 00000 00 00 `email` 00 "REDACTED" 00000 0000000. 0000 000000 00000 0000 0000 0000000000 `email`0 `ltv` 00 00000, 0 00 `email` 0000 "REDACTED" 00000 0000000. 0000 00 00: [Databricks Certified Data Engineer Professional], "Lakehouse" 00; Databricks 00, "CASE 0000" 00.

NEW QUESTION: 93

0000 0000000 0000 000000 00000 0000 00000 00 0000 00000000. 00 00 0000 0000 00000.
0000:

- ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐.
- ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐?
- ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐?
- A. .read ☐ ☐ .readStream☐ ☐ ☐ ☐.
- B. .read ☐ ☐ ☐ ☐ .stream ☐ ☐ ☐ ☐ ☐.
- C. `format("cloudFiles")` ☐ ☐ `format("stream")`☐ ☐ ☐ ☐.
- D. .load(dataSource) ☐ ☐ ☐ ☐ .stream ☐ ☐ ☐ ☐ ☐.
- E. spark ☐ ☐ ☐ ☐ ☐ .stream ☐ ☐ ☐ ☐ ☐.

NEW QUESTION: 94

```

Cards {id LONG, items ARRAY<STRUCT<id: LONG, count: INT>>|
id LONG, email email
1001[{id: "DESW6", count: 1}] "u1@domain.com"
1002[{id: "HYND4", count: 1}, {id: "M2", count: 2}] "u2@domain.com"
1003[{id: "M2", count: 1}] "u3@domain.com"

```

```

MERGE INTO TEST t
USING updates u
ON t.id = u.id
WHEN MATCHED THEN
  UPDATE SET *

```

How would the following update be handled?

```

tnew created from t, adding existing column:
50 items
2021/12/01 08:03, count: 2, columns: "BOOK

```

- Answer: ([SHOW ANSWER](#))**

* Delta Lake       Databricks : <https://docs.databricks.com/delta/delta-batch.html#schema-evolution>

NEW QUESTION: 95

Lakeflow □□□ □□□□□□ □□ □□□ Spark □□□ □□□□□ □□□ □□□□?

- A.** Lakeflow□ □□□ □□□□□□ □□□ □□□□□□ □□□□□□□□ □□□□ □□□□ □□, □□□ □□□□□ □□□ □□□□ □□□□ □□ □□ □□□□□□□□ □□□□□.
- B.** □□□□ □□□□□ □□□□ □□□ □□□□ □□□ □ □□□, Lakeflow □□□ □□□□□□ □□□ □□□□.
- C.** Lakeflow □□□ □□□□□□ Delta Lake □□□□□ □ □ □□□, □□□ □□□□□□ □ □ □□□□.
- D.** Lakeflow□ □□□ □□□□□□ □□□ □□□ □□□□ □□□□ □□, □□□ □□□□□ □□ □□ □□□ □□□ □□□□□.

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

Databricks □□□ □□□□ □□□□ □□□ □□□ □□□□ □ □□□□□ □□□ □□:

Databricks □□□ □□□ Lakeflow Declarative Pipelines□ Structured Streaming□ □□□□ □□□, □ □□ □□□ □□□□□□□ □ □□□ □□□ □□□□□. Airflow□ Azure Data Factory□ □□ □ □□□□□□□ □□ □□□ □□□ □□□ □□ □□□ □□ □□□, □□□ □ □□□ □□□□ □□□□□. □□ Structured Streaming□ □ □□ □□□□ □□□□□ □□□□ □□□□ □□ □□ □ □□□□□, □□□ □ □□□□ □□□□ □□□□ □□□. □ □ Delta Lake □□ □ □□□ □□□ □□□□□, Lakeflow Declarative Pipelines□ □□ □ □□□ □□ □□□□ □□□□□ □□□□ □□ □ □□□□□□. □□□ □□□ □□□ A - Lakeflow Declarative Pipelines□ □□□□ □□□□□□□□ □ □□□□□.

NEW QUESTION: 96

- □□□□□ □□□□□ Databricks SQL □□□□□ □ □□□ □□□□ □ □ □ □□ □□□□ □□□ □□□□□□.
- "□□ □□" □□□ □□□□ □ □□□□□□ □ □ □□ □□□□. □□□ □ □□□ □□□□□ □□□.
- □□□□ □□□□ □ □□□□ □□□ □□□□ □□□ □□□ □□□□□.
- □ □□□□ □□□□□ □ □ □□ □□□□ □□□ □□□□ □□□ □□ □□□□□?
- A.** □ □□□□ □□□□□ SQL □□□□□□□ □□□□ □ □ □ □□ □□□ □ □□□□.
- B.** □□□□□ □□□ □□□ □□□ □□□□ □□□□ □ □ □ □□ □□□ □ □□□□.
- C.** □□□□□ □□□ □□□ □□ □□□ □□□□ □□□ □□□□ □□□ □□□ □ □□□□.
- D.** □□□□ □□□□□ □□□ □□□ □□□□ □□ □□□□□□ □□□ □ □□□□.
- E.** □□□□□ □□□ □□□ □□ □□□□ □□ Databricks □□□□□ □□□□ □□ □ □□□□.

Answer: ([SHOW ANSWER](#)**)**

NEW QUESTION: 97

- □□ □□□□ □□□□□ /mnt/finance_eda_bucket □□□ □□□□□□□□.
- □□ □□□□□□□ □□□□ □□ □□□ □□ □□□ □□□□□□□.

```
CREATE DATABASE finance_eda_db
LOCATION '/mnt/finance_eda_bucket';
GRANT USAGE ON DATABASE finance_eda_db TO finance;
GRANT CREATE ON DATABASE finance_eda_db TO finance;
```

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

```
CREATE TABLE finance_eda_db.tx_sales AS
SELECT *
FROM sales
WHERE state = "TX";
```

- □□ □□□□ financegroup□ □□□□ □□, tx_sales □□□□ □□□□ □□□ □□□□ □□□ □□□□□?
- A.** □□ □□□□ Databricks □□□ □□□□□ Hive □□□□□□□ □□ □□□ □□□□□.
- B.** /mnt/finance_eda □□□ □□□□ □□□□ □□□□□ □□ □□□□ □□□□□.

- C. `dbfs:/mnt/finance.eda` is a Databricks Hive path.
- D. `/mnt/finance.eda` is a local file path.
- E. DBFS is a distributed file system.

Answer: D (LEAVE A REPLY)

<https://docs.databricks.com/en/lakehouse/data-objects.html>

NEW QUESTION: 98

Which of the following is a valid Databricks Auto Loader configuration? `cloudFiles.schemaLocation` is a required configuration. `mergeSchema` is an optional configuration. `writeStream` is a required configuration. `cloudFiles.format` is a required configuration.

- A. `cloudFiles.schemaLocation`
- B. `mergeSchema`
- C. `writeStream`
- D. `cloudFiles.format`
- E. `cloudFiles.location`

Answer: B (LEAVE A REPLY)

Which of the following is a valid Databricks Auto Loader configuration? `cloudFiles.schemaLocation` is a required configuration. `mergeSchema` is an optional configuration. `writeStream` is a required configuration. `cloudFiles.format` is a required configuration.

* `AutoLoader` is a Databricks configuration.

<https://docs.databricks.com/en/ingestion/auto-loader/schema.html>

* `cloudFiles` is a Databricks configuration.

<https://docs.databricks.com/spark/latest/structured-streaming/writing-streaming-data.html>

NEW QUESTION: 99

- A. `SET Timezone = America/New_York` is a valid SQL statement.
- B. `SET Timezone = America/New_York` is a valid SQL statement.
- C. `SET Timezone = America/New_York` is a valid SQL statement.
- D. `SET Timezone = America/New_York` is a valid SQL statement.
- E. `SET Timezone = America/New_York` is a valid SQL statement.

Answer: D (LEAVE A REPLY)

- 1. `SET Timezone = America/New_York` is a valid SQL statement.
- 2. `SET Timezone = America/New_York` is a valid SQL statement.

NEW QUESTION: 101

Which of the following MLflow Spark UDFs can be used to write predictions to a Delta Lake table? The table schema is "customer_id LONG, predictions DOUBLE, date DATE".

from pyspark.sql.functions import current_date

```
model = mlflow.pyfunc.spark_udf(spark, model_uri="models:/churn/prod")
df = spark.table("customers")
columns = ["account_age", "time_since_last_seen", "app_rating"]
preds = (df.select(
    model(*columns).alias("predictions"),
    current_date().alias("date")
))
```

Which of the following MLflow Spark UDFs can be used to write predictions to a Delta Lake table? The table schema is "customer_id LONG, predictions DOUBLE, date DATE".

A. preds.write.format("delta").save("/preds/churn_preds")

Which of the following MLflow Spark UDFs can be used to write predictions to a Delta Lake table? The table schema is "customer_id LONG, predictions DOUBLE, date DATE".

```
(preds.write
    .format("delta")
    .mode("overwrite")
    .saveAsTable("churn_preds"))
```

B.)

C. preds.write.mode("append").saveAsTable("churn_preds")

```
(preds.writeStream
    .outputMode("append")
    .option("checkpointPath", "/_checkpoints/churn_preds")
    .table("churn_preds"))
```

D.) databricks

Answer: (SHOW ANSWER)

NEW QUESTION: 102

Which of the following MLflow Spark UDFs can be used to write predictions to a Delta Lake table? The table schema is "customer_id LONG, predictions DOUBLE, date DATE".

```

-----
AnalysisException                                Traceback (most recent call last)
<command-3293767849433948> in <module>
----> 1 display(df.select(3*"heartrate"))

/databricks/spark/python/pyspark/sql/dataframe.py in select(self, *cols)
    1690         [Row(name='Alice', age=12), Row(name='Bob', age=15)]
    1691         """
-> 1692         jdf = self._jdf.select(self._jcols(*cols))
    1693         return DataFrame(jdf, self.sql_ctx)
    1694

/databricks/spark/python/lib/py4j-0.10.9-src.zip/py4j/java_gateway.py in __call__(self, *args)
    1302
    1303         answer = self.gateway_client.send_command(command)
-> 1304         return_value = get_return_value(
    1305             answer, self.gateway_client, self.target_id, self.name)
    1306

/databricks/spark/python/pyspark/sql/utils.py in deco(*a, **kw)
    121         # Hide where the exception came from that shows a non-Pythonic
    122         # JVM exception message.
--> 123         raise converted from None
    124     else:
    125         raise

AnalysisException: cannot resolve 'heartrateheartrateheartrate' given input columns:
[spark_catalog.database.table.device_id, spark_catalog.database.table.heartrate,
spark_catalog.database.table.mrn, spark_catalog.database.table.time];
'Project ['heartrateheartrateheartrate]
+- SubqueryAlias spark_catalog.database.table
   +- Relation[device_id#75L,heartrate#76,mrn#77L,time#78] parquet

```

□□□ □□□ □□□□ □□□ □□□□□?

- A. □□□ □□□ PvSoark□□□ Scala □□□□□ □□□□□□□.
- B. □□□□ heartrateheartrateheartrate□□ □□□ □□ □□□□.
- C. □□ □□□ □□ □ □□□□ □□ □□□ □□□□□□□.
- D. DataFrame □□□ □□□ □ □ □□□□ □□ □□□ □□□□□□□.
- E. □□□ □□ □□ □□□□ □□□□ □□ □□ □□□ □□□□□□□.

Answer: ([SHOW ANSWER](#))

□□

AnalysisException. AnalysisException Spark SQL. ¹ .
 'heartrateheartrateheartrate' .
 'heartrate' 'age' . . 'heartrateheartrateheartrate' .
 ' '. : AnalysisException

NEW QUESTION: 103

,
Lakehouse ?
A. Lakehouse .
B. Lakehouse SQL .
C. Lakehouse .
D. Lakehouse ,
E. ACID .

Answer: D (LEAVE A REPLY)

NEW QUESTION: 104

. , 10 .
 ?
A. .
B. 60 .
C. .
D. .

Answer: C (LEAVE A REPLY)

. ,
 . 10 ,
 :
Databricks : https://docs.databricks.com/jobs.html

NEW QUESTION: 105

.

Cmd 1

```
rawDF = spark.table("raw_data")
```

Cmd 2

```
rawDF.printSchema()
```

Cmd 3

```
flattenedDF = rawDF.select("*", "values.*")
```

Cmd 4

```
finalDF = flattenedDF.drop("values")
```

Cmd 5

```
finalDF.explain()
```

Cmd 6

```
display(finalDF)
```

Cmd 7

```
finalDF.write.mode("append").saveAsTable("flat_data")
```

□□□□ □□□□ □□□□ □□ □□ □□□□ □□□□ □□□?

- A. □□ 2
- B. □□ 3
- C. □□ 4
- D. □□ 5
- E. □□ 6

Answer: E (LEAVE A REPLY)

□□ 6□ □□□□ □□□□ □□□□ □□ □□ □□□□ □□□□ □□□□. □ □□□ finalDF □□□□□□□□ □□ □□ □□□□ □□□□ □□□□□□. finalDF □□□□□□□□ □□ □□ 7□□ □□□□ □□□□□□□□ □ □□□ □□□□ □□□□. □□□□ □□□□□□□□ □□□□ □□ □□□□ □□□ □□ □□ □□□□ □□□ □□□ □□□□.

□□□ Cmd 6□ □□□□□□ □□□□ □□□.

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

* Cmd 1: raw_data □□□□ rawDF□□ Spark □□□□□□□□□ □□□□□□.

* Cmd 2: rawDF □□□□□□□□ □□□□ □□□□□□. □□ □□□ □ □□□ □□ □□□ □□□□□.

* Cmd 3: rawDF □□□□□□□□ □□ □□ values □□□ □□ □□□ □□ □□ □□□□ flattenedDF□□ □ □□□□□□□ □□□□□.

* Cmd 4: flattenedDF □□□□□□□□□ □ □□ □□□□ □□ values □□ □□□□ finalDF□□ □ □□□□□□□ □□□□□.

* Cmd 5: finalDF □□□□□□□□ □□□ □□ □□□ □□□□□□. □□ □□ □□□ □□□□□ □□□□ □ □□□□□.

* Cmd 7: finalDF □□□□□□□□ flat_data□□ □□□□ □□□□. □□ □□□□ □ □□□□ □□□□ □□□□ □□□□□.

NEW QUESTION: 106

□□ □ Delta Lake □ □□□ □□□ □□□□ □□□ □□□□□?

- A. □□ □□ □□ □□□ □□□□ □□□□□□□ □ □□□□ □□□□ □□□□ □□ □□ □□ □□ □□ □□ □□ □□□□.
- B. □□□□ □□ □□□ □□□□ □□□ □□ □□ □□□□ □□□□, □□□ □□□ □□□□□□□ □□□□□ □ □□ □□ □□ □□□ □□□□□.
- C. □□ □□□ □□□ □ □□□ □□□ □□□ □ □□□ □□□□ □□ □□□ □□□ □□□□□. □, □□□□ 1GB □ □□□ □□□ □□□□□.
- D. □□ □□□□□ □□□□ □□ □□ □□ □□ □□ □□□ □□ □□□□ □□ OPTIMIZE □□□□□.

Answer: A (LEAVE A REPLY)

Delta Lake □□□ □□□ Delta □□□□ □□□□ □□□□ □□ □□ □□□ □□□□□. □ □□ □□□ □□□ □□ □□□□ □□□□ □□□□□ □□ □□ □□ □□ □□ □□ □□ □□□ □□□ □ □□□ □□□. □□□ □□ □□□ □□□□ □□ □□ □□ □□ □□□, □□□□□ □□□□□ □□□□ □□ □□□ □□□□□, □□ □□ □□□ □□ □□□□□ □□□ □□ □□ □□□□□ □□□□ □ □□□□.

□□ □□:

* Databricks □ Delta Lake □□ □□ □□ □□:

<https://docs.databricks.com/delta/optimizations/auto-optimize.html>

Databricks-Certified-Professional-Data-Engineer □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ Databricks-Certified-Professional-Data-Engineer □□! DumpTop □ □□ **Databricks-Certified-Professional-Data-Engineer** □□ □□□ □□□□□□, DumpTop Databricks-Certified-Professional-Data-Engineer □□ □□□ □□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □□ DumpTop Databricks-Certified-Professional-Data-Engineer □□□ □□□□□. <https://www.dumptop.com/Databricks/Databricks-Certified-Professional-Data-Engineer-dump.html> (217 Q&As Dumps, **30%OFF Special Discount: KrDump**)

NEW QUESTION: 107

user_ltv □□ □□□ □□□□ □□ □□ □□□ □□□□□ □□□ □□ □□□□ □ □□□□ □□□□.

□□□□□□□ □□□□□ □□□□ □□□□, □ □□□ ACL □□□□ □□□ □□ □□□ □□□□ □ □□□□□.

user_ltv □□□□ □□□ □□ □□□□ □□□ □□□□.

□□□ □□□, □□ □□, □□ □□ □□ □□

□□ □ □□□ □□□□□.

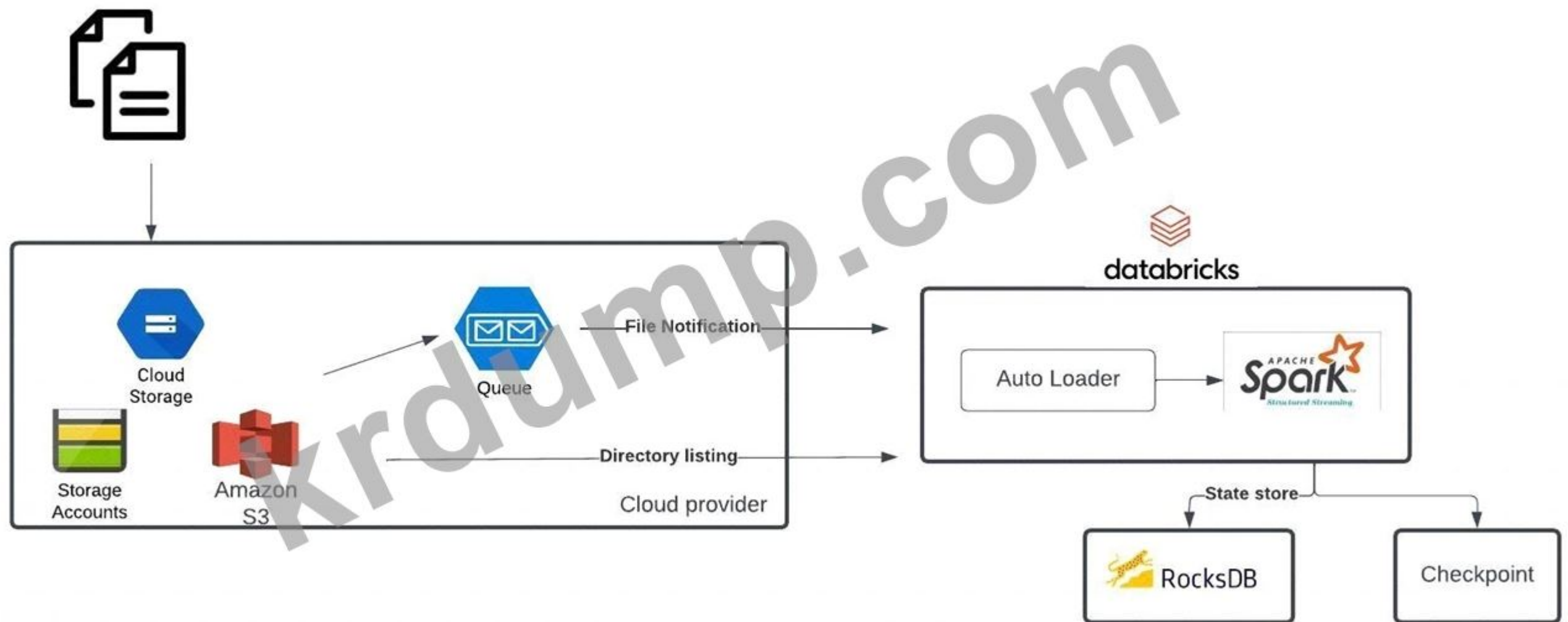
```
CREATE VIEW email_ltv AS
SELECT
CASE WHEN
    is_member('marketing') THEN email
ELSE 'REDACTED'
END AS email,
ltv
FROM user_ltv
```

□□□ □□ □□□ □□ □□□□ □□□ □□ □□□ □□□□□.

SELECT * FROM email_ltv

□□ □ □ □□□ □□□ □□ □ □□□□ □□□ □□□□□?

Auto Loader & Cloud Storage Integration



*Directory listing also supports incremental file listing

□□ □□ □□ □□□□ □□□ □□ □□□□ □□□ □□□ □□ □□□ □□□□. □□ □□ □□ □□□ □□ □□ □□□□ | AWS □ Databricks

NEW QUESTION: 111

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

A. □□□□ □□□□

B. □□ □□ □□ □□□□

□□ □□ □□ □□□ □□□□□□.

'□□□'. □□: AnalysisException

NEW QUESTION: 114

Databricks□□ □□ □□□□ □□□ □□ □ □□□ DBFS □□ □□□□□ dbutils.fs.mount()□ □□□□ □□□□ □□ □□ □□□□□□□□.

□□ □ □□ □□□ □□□□□?

- A. DBFS□ Unix □□ □□□□ □□□ □□□ □□□ □□□ □□ □□□□□ □□□ □□□ □□ □□□ □ □□□ □□ □□ □□□ □□□□□□□□.
- B. □□□□□ DBFS □□□ □□□□ □□□ □□□ □□□□□□ □□□□ □□□ □ □□□□□.
- C. DBFS □□□ □□□□ □□□□ □□ □□□ □□□□□□. □□□□ □□□□ □□□ □□□ □□ □□ □ □□ □□□ □□□ □□ □□□□□□.
- D. Databricks □□□□□ %sh□ □□□ □ DBFS □□ □□ □□□□ □□□□ □□□□ □ □□□□.
- E. DBFS □□□ □□□□□ □□□ □□ □□ □□□ □□□ □□□□ □□, □□□□ □□□□□ □□ □□ □□□ □□□□ □□ □□ □□□□ □□□□□□.

Answer: (SHOW ANSWER)

DBFS□ Unix □□ □□□□ □□□ □□□ □□□ □□□□ □□□□ □□ □□□□□ □□□ □□□ □□ □□□ □□ □□ □□□ □□□□□□□□.1 DBFS□ □□□ □□ □□□□ □□□ □□ □□□□ □□ □□□ □□□□□, □□□ □□□ □□□ □□ □□□ □□□ □□ □□□□□.1 □□□□□ DBFS □□□ □□□□□□□□ □□ □□□□ □□□ □ □□□, □□□□ □□□ □□□ □□ □□ □□□ □□□□ □□ □□ □□□□□ □□□ □□ □□□□□□.2 □□□□ □□□□ □□□ □□□ □□ □□ □ □□ □□□ □□□□ □□□□ □□□, □□□ □ □□□ □□ □□□ □□ □□□ □□ □□□□ □ □□.3 □□□□□ FUSE□ □□□□□ □□ □□, Databricks □□□□□ %sh□ □□□□ □ DBFS □□□ □□□□ □□□□ □□□ □□□ □ □□□□.4 DBFS □□□ □□□□□ □□□ □□ □□ □□ □□□ □□ □□□□□□□ □□□ □□ □□□□□ □□□ □□□□□□.1 □□□□ □□□□□ □□□ □□□□□ □□□□ □□ □ □□ □□ □□□ □□□□ □□ □□□□□ □□□□□□.3 □□ □□: DBFS, Azure Databricks□□ □□ □□, Azure Databricks□ □□□□ □□ □□□□ □□□, FUSE□ □□□□ □ DBFS □□□

NEW QUESTION: 115

□□□ □□□□□ □□□ □□□□ □□ □□□ □□ □□□□ □□□□□ □□□□□ Databricks □□□□□ □□□□□ □□□□□.

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

- A. □□□□□□ □□ □□□ □ □ □□ □□□□ □□□□ □□ CDC(□□ □□□ □□) □□□□□□ □□□□□□.
- B. □□□□ □□□ □□ 1□ □□□ □□ □□ □□□□ Apache Kafka □□□□ □□□□ □□□□□.
- C. □□ □□ □□ □□□□□ □□□ □□ □ □□□ □□ □□□□ □□ □□□□ □□□ □□□□□□.
- D. □□□□□□ □□□□ □□□ □□□□ □□□□□ □□□□ □□□□ □□□ □□□ □□□□□ □□□□□□□□.

Answer: C (LEAVE A REPLY)

Databricks□ □□□□ □□ □□□ □□ SQL □□, □□ □ □□□ □□□ □□ □□□□ □□□□ □ □□□□□ □□□□□. □□ □□□ □□□□□ □□□□ □□□ □□ □□ □□ □□ □□ □□ □□□ □□ □□□ □□ □□ □□ □□□□□ □□□ □□□ □□□□□.

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

Databricks □□□□□□□ □□□□ □□□□□ □□□ □□ □□□□□□ □□□□ □□, □□□□□ □□□□□□ □□□□ □□□□ □□□□ □□□□□ □□□□□□. □□□ □□□ □□□□□□ □□□ □□□ □□□ □ □□□□ □□ □□□ □□ □□□ □□ □□ □□□ □□□ □ □□□□ □□ □□□ □□□□ □□□□.

NEW QUESTION: 116

□□□ □□□□□ □□ □□(□□□□ □□□ □□) □□□□ □□□□ □□ □□ □□□□ □□□□ □□□ □□ □□□ □□□□ □□□ □□ □□□□□ □□□.



A. Joining on event time constraint: `clickTime == impressionTime` using a `leftOuter` join

B. Joining on event time constraint: `clickTime >= impressionTime - interval 3 hours` and removing watermark

C. Joining on event time constraint: `clickTime + 3 hours < impressionTime - 2 hours`

D. Joining on event time constraint: `clickTime >= impressionTime AND clickTime <= impressionTime + interval 1 hour`

1. 在代码中，我们使用 `clickTime == impressionTime` 来判断广告是否被点击。如果为 `true`，则表示广告被点击，我们将 `clickTime` 设置为 `impressionTime`。

2. 在代码中，我们使用 `clickTime != impressionTime` 来判断广告是否未被点击。如果为 `true`，则表示广告未被点击，我们将 `clickTime` 设置为 `0`。

3. 在代码中，我们使用 `clickTime < impressionTime` 来判断广告是否被点击的时间早于展示时间。如果为 `true`，则表示广告被点击的时间早于展示时间，我们将 `clickTime` 设置为 `0`。

4. 在代码中，我们使用 `clickTime > impressionTime` 来判断广告是否被点击的时间晚于展示时间。如果为 `true`，则表示广告被点击的时间晚于展示时间，我们将 `clickTime` 设置为 `0`。

A. Delta □□□□□ □□ □□ □□□ □□□□ □□ □□□□ □□□□□.
B. □□□ □□ □□□ □□ □□□ □□□□ □□□ □□□ □□□□□.
C. event_date □□ □□□□ □□□□□ □ Z-□□ □□ □□□ □□□□□.
D. □□ □□□ □□□ □□□□□ □□□□□ □□□ □□□ □□□ □□□□□ □□□□□.

Delta Lakeは、データレイク上で、Apache Spark、Apache Flink、Apache Iceberg、Databricksなどのデータエンジンと連携して、データを安全に保存し、管理するための技術です。Delta Lakeは、データの追記、更新、削除、スキーマの進化、パーティションの最適化、データの追跡と監査などの機能を提供します。Delta Lakeは、データの信頼性と整合性を確保するための重要な技術です。

NEW QUESTION: 118

SQL□ □□□□ DBFS □□□□□ □□ □□ CSV □□□□ □□□□ □□ □□ □□□ □□□□□?

- A. 'dbfs:/location/csv_files/'□□ CSV □□□□ *□ □□□□□.
- B. 'dbfs:/location/csv_files/'□□ CSV.*□ □□□□□.
- C. CSV. 'dbfs:/location/csv_files/'□□ *□ □□□□□.
- D. □□ □□□ □□ □□□ □ □□□□. □□ COPY INTO□ □□□□ □□□□ □□□□ □□□□ □□□.
- E. 'dbfs:/location/csv_files/'□□ CSV□ □□□□ *□ □□□□□.

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

□□
□□□ SELECT * FROM CSV. 'dbfs:/location/csv_files/'□□□.
□□ □□□ □□□□□ □□□□□ □□□ □□ □□□ □□□ □ □□□□.
`/Location` □□□□ *□ □□□□□.
□□ - CSV, JSON, PARQUET, TEXT

NEW QUESTION: 119

□□□□□ □□□□□ □□□□□ □□□□□ □□□ □□□□ □□□□ □ □□□ □ □□ REST API □□□□ □□□□□?

- A. /jobs/runs/list
- B. /jobs/runs/get-output
- C. /jobs/runs/get
- D. /jobs/get
- E. /jobs/list

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

□□□□ □□□□□. REST API □□□□ □□□□□ □□□□□ □□□□ □□□□□ □□□ □□□□ □□□ □ □□ □□□□□. REST API□ □□□□, □□, □□□, □□□□□ □□ Databricks □□ □□ □□□□□ □□□□□ □□ □□□ □ □□ □□□□□□□□□. REST API□ GET, POST, PUT, DELETED □□ HTTP □□□□□ □□□□□ □□□ □□□□□ □□ □□□ □□□□□. /jobs/get □□□□□ □ □□ ID□ □□□□□ □□□ □□ □□□ □□□□□ GET □□□□□□□. □ □□□□□ □□ □□, □□, □□ □□, □□□, □□□□ □□ □□ □□□ □□□□□. □□□□□ □□□ □□□ □ □□□□□. □□□□□ □□□ □□□□□□ □□□□□ □□□□ □□□ □□□□□ DBFS□ □□□□ JAR □□□□ □□□□□ JAR □□□□, □□ □□□ □□□□□□ □□□ □ DBFS□ □□□□ Python □□□□ □□□□□ Python □□□□□ □ □□□□□. □□□□□□ □□□ □□ □□ □□□ □□□□□ □□□ □□□□□ □ □□ □□ □□□□□. /jobs/get □□□□□□□ □□□ □ □□□□□ □□□□□ □□□□□ □□□□□ □□□ □ □□□□□. □□□□ □□ □□: [Databricks □□ □□□□ □□□□□], "Databricks Jobs" □□; Databricks □□, "Get" □□; Databricks □□, "JobSettings" □□.

NEW QUESTION: 120

□□ □□□□ □□□□□ Databricks Jobs UI□ □□□□□ □□□ □□□ □□□□□ □□□□□□□. □□□□ □□□□□ □, □□□□□□ □□□ □ □□□ "□□□□"□ □□□□□ □□ □□ □□□□□□□. □□ □□□ □□□□ □□□□□□ □□□□□□.

"DevOps" □□□□ □□ "□□□□" □□□□ □□□□ □ □□□□ □□□□□ □□□□ □ □□□□.

□□ □ □□ □□□ □□□□ □□ □□□ □□ □□□ □□□□□?

- A. Databricks □□□□□ □□□□□ □ □□ □□□ □□□□. "□□□□" □□□□ □□□□ □□□ □ □□□□.
- B. Databricks □□ □□□□□ □□ "□□□□" □□□□ □□□□, □ □□□□ □□□□ □ □□□□.
- C. □□ "admins" □□ □□□□ □□ □□□□□□□□ □□□□ □□ □□□□ □□□□ □ □□□□.
- D. □□□□□ □□□□ □□ □□□ □□□□ □□□□ □□ □□□□ □□□□ □□□□ □□□□ □ □□□□.
- E. □□□□□□□ □□□□□ □□□□ "□□□□" □□□□ □□□□ □ □□□□.

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

□□□ □□□ □□□□□ "□□□" □□□ "DevOps" □□□□ □□□ □ □□ □□□ Databricks □□□□ □□□□ □ □□ □□□ □□, □□□□ □□□ □□ □□ □□□□□ □□ □□□□□. □□□□ □□□□ □ □□ □□ □ □□□, □□□ □□□□ □□□ □□ □□□□. □□ □□□□ □□□ □□□ □□□ □□□□□□ □□ □□□□□□ □□□□ □□□□ □□□□□□. □□ □□□□ □□ □□□ □ □□ □□□ □□□ □□□, □□ □□□□ □□□□ □□□ □□□□□ □□□ □ □□□□. □□□ □□□□ □□□□ □□□□ □□□ □ □□, □□ □□□□□□ □□□ □ □□□□. □□□ □□□ □ □□ □□□□□ "□□□" □□□ "DevOps" □□□□ □□□□□ □□□ □□□□□□□. □□ □□:

* □□ □□ □□: <https://docs.databricks.com/security/access-control/table-acls/index.html>

* □□ □□:

<https://docs.databricks.com/security/access-control/table-acls/privileges.html#job-permissions>

NEW QUESTION: 121

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

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

```
store_id INT, total_sales_qtd FLOAT, avg_daily_sales_qtd FLOAT, total_sales_ytd
FLOAT, avg_daily_sales_ytd FLOAT, previous_day_sales FLOAT, total_sales_7d FLOAT,
avg_daily_sales_7d FLOAT, updated TIMESTAMP
```

□□ □□□ □□ □□□□□□□ □□ □□□ □□ □□□ □□□ □□□ □□□□ □□□□ □□□, □ □□□□ □□ □□□□□ □□□□□ □□□□□□□□. products_per_order□□ □□□ □ □□□□ □ □□□ □□ □□□ □□□□ □□□□.

```
store_id INT, order_id INT, product_id INT, quantity INT, price FLOAT,
order_timestamp TIMESTAMP
```

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

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

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

B. Delta Cache□ □□□□ products_per_order □□□□ □□□□ □□□□□□ □ □□ □ □□□□□ □□□ □□□ □ □□□□.

C. Databricks □□□ □□ products_per_order □□□□ □□ □□□ □□□□□ □□□□□ Structure Streaming□ □□□□□□.

D. products_per_order □□□□ □□ □□ □□□□, □ □□ □□ □□□□□ □□□□□.

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

Databricks-Certified-Professional-Data-Engineer □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ Databricks-Certified-Professional-Data-Engineer □□! DumpTop □ □□ **Databricks-Certified-Professional-Data-Engineer** □□ □□□ □□□□□□, DumpTop Databricks-Certified-Professional-Data-Engineer □□ □□□ □□□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □□ DumpTop Databricks-Certified-Professional-Data-Engineer □□□ □□□□□. <https://www.dumptop.com/Databricks/Databricks-Certified-Professional-Data-Engineer-dump.html> (217 Q&As Dumps, **30%OFF** Special Discount: **KrDump**)

NEW QUESTION: 122

□□ □□□ □□ □□□ □□□□ Delta Lake □□□□ □□□□ □□□□□.

```
from pyspark.sql.functions import current_timestamp, input_file_name, col
from pyspark.sql.column import Column
```

```
def ingest_daily_batch(time_col: Column, year:int, month:int, day:int):
    (spark.read
     .format("parquet")
     .load(f"/mnt/daily_batch/{year}/{month}/{day}")
     .select("time_col.alias('ingest_time'),
            input_file_name().alias('source_file')")
     .write
     .mode("append")
     .saveAsTable("bronze"))
```

Which of the following code snippets will correctly read the data from the 'bronze' table and filter it based on the 'ingest_time' column?

def new_records():

A. spark.readStream.table("bronze").

B. spark.readStream.load("bronze").

C.

```
return (spark.read
        .table("bronze")
        .filter(col("ingest_time") == current_timestamp()))
```

D. spark.read.option("readChangeFeed", "true").table("bronze").

E.

```
return (spark.read
        .table("bronze")
        .filter(col("source_file") == f"/mnt/daily_batch/{year}/{month}/{day}")
        )
```

Answer: E (LEAVE A REPLY)

<https://docs.databricks.com/en/delta/delta-change-data-feed.html>

NEW QUESTION: 123

[illegible]

- A.** 所有 的 数 据 都 是 正 数 且 有 限 个。
- B.** 所有 的 数 据 都 是 正 数 且 有 限 个， 且 每 个 数 据 都 是 正 数 且 有 限 个。
- C.** 所有 的 数 据 都 是 正 数 且 有 限 个。 且 每 个 数 据 都 是 正 数 且 有 限 个。
- D.** 所有 的 数 据 都 是 正 数 且 有 限 个， 且 每 个 数 据 都 是 正 数 且 有 限 个 API 的 数 据 都 是 正 数 且 有 限 个。
- E.** 所有 的 数 据 都 是 正 数 且 有 限 个。

Answer: C (LEAVE A REPLY)

11

Jobs□ Tasks□ □□□ Job □□ □ Tasks□ □□ □□ □□ □□□□□ □□□□□.

NEW QUESTION: 124

□□□ □□□□□ □□ □□□ □□□ □□□□ □□ □□□□ Databricks□ □□□□□□. □□ □□ □□ □□□□ Apache Iceberg □□□ □□□ □□ □□ □□□ □□ □□□□□ □□□. □□□ □□□□□ □□□ □□ □□□?

- A.** □□□□ □□ □□□□ □□□□ □□□ □□□□□ □□□□□.
- B.** □□ □□□□□ 'iceberg' □□□ □□□□ □□□□ □□□□ Iceberg □□□□ □□ □ □□□ □□□.
- C.** □□□□□ □□□ □ □□□ □□ □□ □□□□ □□□□□ □□□□□ □□□□□.
- D.** □□□□ □□ □□□□ □□□□□ □□□□ □□□□ □□□□ □□□□□ □□□□ □□ □□□□ □□ □ □□□ □□□.

Answer: D (LEAVE A REPLY)

Delta Lake  Delta Universal Format(Delta UniForm)  Delta Lake  Apache Iceberg      .  , Delta Iceberg Delta .

□ □ □ □ □ □ □ □ :

- (A) □□□□ □□ □□□□ □□□□ □□□ □□□□□ □□□□□.
- □□: Delta Lake □□ □□□□ □□□, □□ □□□ □□□□ □□ □□ □□□□□ □□ □□□ □□□□ □□□□.
- (B) □□ □□□□□ 'iceberg' □□□□□ □□□□ Iceberg □□□□ □□ □ □□□ □□□□. (□□□□ □)
- □ □□□ □□□ □□□□□ □□□.

□□□□ □□ □□□□ □□□ □□□□ □□ □□□□ □□□□.

- (C) □□□□□ □□□ □ □□□ □□ □□ □□□□ □□□□□ □□□□ □□□□□. (□□)

Iceberg □□□ □□□ □□□□ □□ □□□ □□□□ □□ □□ □□□□ □□□□.

Delta UniForm□ Iceberg□ □□ □□□□□ □□□□□ □□□ □□□ □□□ □□□□ □□□□.

- [illegible]

□□ □□ □□□ Databricks Delta to Iceberg □□□□□□ □□□ □□□□ □□ Delta □□□□ Iceberg□ □□□□ □□□□.

□□ □ □□□□ □□□□□ Delta □ Iceberg □□ □□□□ □□□□ □□ □□□□ □ □□□□.

□ □ :

Delta Iceberg Delta Iceberg

□□□ (D)□□ □□□□□.

11

Apache Iceberg ☐ Delta UniForm - Databricks ☐

Delta ☒ Iceberg ☒ ☐ ☐ - Databricks

NEW QUESTION: 125

□□ Spark Stream □□□□□ □□□□ □□□ □□□□ □□ □□□ □□ □□ □□□ □□ □□ □□□□ □□□□ □□□ □□ □□□□. □□ □□□□ □□□□□ □□□□□ □□□ □□□□.

- 1.spark.table("source_table")
- 2..writeStream
- 3..□□("____", "dbfs:/location/silver")
- 4..outputMode("____")
- 5..□□□(□ □=____)
- 6..table("target_table")

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

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

NEW QUESTION: 126

Databricks □□□ 3□□ □□□□ □□□□□□□, □ □□□□ Databricks □□□□□□□. □□□ A□ □□ □□□□ □□□□ □□□□. □□□ B□ C□ □□□ □□□□, □□ □□□ A□ □□ □□□□ □□□□.

□□ A□ B□ □□□□□ □□□□□□ □□ C□ □□□ □□ □□ □□□□ □□, □□□□ □□□ □□□□ □□□ □□□□□?

- A. □□ A□ B□ □□□ □□□□ □□□ □□ □□□ □□□□□ □□□□□□□□. □□ C□ □□ □□□ □□□□□ □□□□□ □ □□□□.
- B. □□ A□ B□ □□□ □□□□ □□□ □□ □□□ □□□□□ □□□□□□□□. □□ C□□ □□□ □□□ □□ □□□ □□ □□□□□□.
- C. □□ A□ □□□ □□□□ □□□ □□ □□□ □□□□□ □□□□□□□□. □□ □□□ □□ □□ B□ C□ □□ □□□ □□□□ □□□□.
- D. □□ □□□ □□□ □□□□ □□□□□ □□ □□□ □□□□□ □□□ □□□ Lakehouse□ □□ □□□ □□□□ □□□□.
- E. □□ □□□ □□□□□ □□□□ □□□ □□□□□□□□ □□ □□□ □□□□ □□□□. □□ C□ □□□□□□□ □□ □□□ □□□□ □□□□□.

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

□ □□□ `CREATE TABLE USING DELTA` □□□ □□□□ DBFS□ □□□ □□ Parquet □□□□ Delta Lake □□□□ □□□□□. □□ `LOCATION` □□□□ □□□□ Parquet □□□ □□□□
`/mnt/finance\_eda\_bucket/tx\_sales.parquet`□□ □□□□□. `LOCATION` □□□□ □□□□□□□ □□□ □□ □□□□ □□□□□. □□ □□□□ □□ □□□□□ □□□□ □□□ □□□□ □□□□□□
□ Databricks□□ □□□□ □□□□. □□ □□□□ DBFS □□ S3□ □□ □□□□ □□□□ □□□□ □□ □□□□□ Parquet □□ CSV□ □□ □□□□ □□□ □□□ □□□ □□ □□ □□□ □ □□□□.

□ □□ □□□ □□□ □□ □□□ /mnt/finance_eda_bucket□ □□□□ □□□□□ □□□□□ prod.sales_by_store□□ □ □□□ □□ □□□□ □□□□ □□□□. □ □□□ □□□□ □□□□□ □□□□
□ □□□□□ □□□ □□□□ □□□□. □□ □□□□□□ □□□ □□□ □□□□□□ □□□ □□□ □□□□ □□ □□□□ Delta □□□□ □□□ □□□□□□. □□□□ □□: [Databricks Certified Data Engineer Professional], "Delta Lake" □□; Databricks □□, "ALTER TABLE RENAME TO" □□; Databricks □□, "□□ □□□ □□" □□.

NEW QUESTION: 127

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

□□□□ □□□ □□□□ □□□ □□□ □□ □□□ □□□□□□.

Databricks Delta Lake □□ □□: Delta Lake □□□

Delta Lake ACID Databricks: Delta Lake ACID

NEW QUESTION: 131

Delta Lake (Change Data Feed) `delta.enableChangeDataFeed = true` `abronzetable` 1 .

```
from pyspark.sql.functions import col
```

```
(spark.read.format("delta")
    .option("readChangeFeed", "true")
    .option("startingVersion", 0)
    .table("bronze")
    .filter(col("_change_type").isin(["update_postimage", "insert"]))
    .write
    .mode("append")
    .table("bronze_history_type1")
)
```

?

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

Answer: (SHOW ANSWER)

Spark.read CDF startingVersion .

NEW QUESTION: 132

DDL .

```
CREATE TABLE inventory (id INT, units FLOAT)
```

A. SET TBLPROPERTIES business_owner = 'supply chain'

```
CREATE TABLE inventory (id INT, units FLOAT)
```

B. TBLPROPERTIES(business_owner = '')

```
CREATE TABLE inventory (id INT, units FLOAT)
```

```
SET(business_owner = '')
```

```
D. CREATE TABLE inventory (id INT, units FLOAT)
```

```
SET PROPERTY (business_owner = 'supply chain')
```

```
E. CREATE TABLE inventory (id INT, units FLOAT)
```

```
SET TAG (business_owner = '')
```

Answer: B (LEAVE A REPLY)

CREATE TABLE inventory (id INT, units FLOAT) TBLPROPERTIES (business_owner = 'supply chain')   (Databricks SQL) | Databricks on AWS ALTER TABLE   TBLPROPERTIES    . ALTER TABLE inventory SET TBLPROPERTIES(business_owner , 'operations')

Answer: E (LEAVE A REPLY)

NEW QUESTION: 136

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

□□ □□□□ □□ □□□□□ □ □□□□□ □□□ □□ □□□ 3□ □□□□□. □□□ □□□□ □□ □□□□□ □ □□□□□□□ □□ □□□ □□ □□□□□□, □□□ 30□□ □□□□□ □□ □. □□ □□□□ □□□ 10□ □□□□ □□□□□□ □□□□ □□□□.

□□ □□ □□□ □□□□ □□□□ □□□□ 10□ □□□ □□□□ □□□ □□□ □, □□ □□□ □□ □□□ □□□□□?

A. □□□ □□□ 5□□ □□□□. □□ □□□ □ □□ □□□□□ □□ □□□ □□ □□□ □□□□ □□ □□ □□□□ □□ □□ □□□□ □□ □□ □□□□ □□ □□□□.

B. □□□ □□□ 30□□ □□□□. □ □□□ □□ □□□ □□ □□ □□□ □□□ □□□ □□□ □□□ □□ □□□ □□□ □□□ □□□ □□ □□ □□ □□□□.

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

[illegible]

E. □□□ □□□ 5□□ □□□□□. □□ □□□ □ □□ □□□□ □□□□ □□□□ □□ □□ □□□ □□ □□□□ □□□□ □□ □□□ □ □□□□.

Answer: D (LEAVE A REPLY)

11

0000 00000. 000000 000000 000000 0000 00 100 000 0000 0000 00 00 000 000 0 00 00000. '0 0 000' 000 0000 000 0 00
 0000 00 000 00 0000 000 0 0000 000 000 000000. 0 000 0000 000 00000 0000 00 000 00 00 00000 00000 00000 0000
 0. '0 0 000' 000 0000 Databricks 000 10000 000 00000 0000 0 0000 00 000 0000 0000 00 000 0000 00 000 000 0 0000. 000 00
 00: [Databricks Certified Data Engineer Professional], "0000 0000" 00; Databricks 00, "0 0 000" 00.

Databricks-Certified-Professional-Data-Engineer ☐☐ ☐☐☐ ☐☐☐☐☐ ☐☐ DumpTop ☐☐ ☐☐☐☐ ☐☐☐ Databricks-Certified-Professional-Data-Engineer ☐☐! DumpTop ☐ ☐☐ **Databricks-Certified-Professional-Data-Engineer** ☐☐ ☐☐☐ ☐☐☐☐☐☐, DumpTop Databricks-Certified-Professional-Data-Engineer ☐☐ ☐☐☐ ☐☐☐☐☐☐☐☐ ☐☐☐ ☐☐☐☐☐☐☐☐. ☐☐☐☐ ☐☐☐☐ ☐☐☐☐☐ ☐☐ DumpTop Databricks-Certified-Professional-Data-Engineer ☐☐☐ ☐☐☐☐☐. <https://www.dumptop.com/Databricks/Databricks-Certified-Professional-Data-Engineer-dump.html> (217

Q&As Dumps, **30%OFF** Special Discount: **KrDump**)

NEW QUESTION: 137

□□□□ □□□ □□□□ □□□□ □□□ □□□ □, □□ □□ □ □□□□ □□□□ □□□ □□ □□□□ □□□ □□□□□?

A. □□□□: □□□ □□ □□□□;

□ □ □ □ □ : □ □ □ □ ;

□□ □□ □□ □□: □□□

B. □□□□: □□□ □□ □□□□:

□ □ □ □ □ : □ □ :

□□ □□ □□ □:1

C. :

□□□ □□: □□□;

□□ □□ □□ □: 1

D. :

□ □ □ □ □ □ : □ □ □ □ :

□□ □□ □□ □:1

E. □□□□: □□ □□□ □□□□:

Answer: ([SHOW ANSWER](#))

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

* □□ □□ □□ □□□ 1□ □□□□ □ □□ □□□ □□ □□□□□ □□□ □ □□□□□. □□□ □□ □□ □□□ □□□ □□□□ □□ □□ □□□ □□□ □□ □□□□ □□ □□□□ □
□ □□ □□ □□□ □□□ □□ □ □□□□.

□□ □□: □□ □□□□, □□ □□□, □□ □□ □□ □□

SQL Delta

- Answer: ([SHOW ANSWER](#))**

```

00 00 0000 00000 /mnt/finance_eda_bucket 000 000000000.
0000 00 00000000 0000 00 000 00 0000 00000000.

```

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

□□□□ □□ □□□□ □□ □□□ □□□□ □□, tx_sales □□□□ □□□□ □□□ □□□□ □□□ □□□□□?

- A.** □□ □□□□ Databricks □□□ □□□□ Hive □□□□□□ □□ □□□ □□□□□.
- B.** /mnt/finance eda □□□ □□□□ □□□□ □□□□□ □□ □□□□ □□□□□.

Unity Catalog □ □□□(□□ C)□ □□ □ □□ □□□□ □□□ □ □□□ □□ □□ □□(PII) □□□ □□□□□ □□□□. □□□ □□ D□ □□□□ □□ □□□□ □□□□□ PII□ □□□□ □□ □□ □□ □□ □□□□□.

NEW QUESTION: 142

DevOps □□ Jobs UI□ □□□□ □□ □□□□□ □□□ □□□ □□□□ □□□□ □□□□□□□□. □□ □□□ □□□ □□□□□ □□□□ □□□ □□□□ □□ □□□ □□□ □ □□□ □ □□□ □□ □□□□□□.

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

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

Answer: D (LEAVE A REPLY)

Databricks□□ □□□□□ □□□□ □□ '□□' □□□ □□□□ □□ □□□□ □□□ □□□ □ □□ □□□ □□□□□ □□□ □□ □□□□. □□□ □□ □□□ □□ □□□ □□□ □□□ □□□ □□□ □□□ □ □□□□□ □□□ □ □□ □□ □□ □□ □□ □□□□ □□□□ □□□ □□□ □ □□□□. □ □□ □□□ □□□ □□ □□□ □□□□ □□ □□□ □□□ □□□ □□□□ □□□□ □□□ □ □□□□ □□□□ □□ □□ □□□ □□□□□.

Databricks □□□□□□ □ □□□□ □□ □□□ □□ □ □□□ □□ □□□ □□ □□□ □□□□□(https://docs.databricks.com/security/access-control/workspace-acl.html).

NEW QUESTION: 143

□□ □ Delta Lake□ □□□□ □□□ □□□□ □□□ □□□□□?

- A. □□ □□ □□ □□□ □□□□ □□□□ □□□□□□□ □ □□□□ □□□□ □□□□ □□ □□ □□□ □□ □□ □ □□ □□ □□□ □□□□□.
- B. □□□□ □□ □□□ □□□□ □□□ □□ □□ □□□□ □□□□, □□□ □□□ □□□□□□□ □□□□□ □ □□ □□ □□ □□□ □□□□□.
- C. □□ □□□ □□□ □ □□□ □□□ □□□ □ □□□ □□□□ □□ □□□ □□□ □□□□□. □, □□□□ 1GB□ □□□ □□□ □□□ □□□□□.
- D. □□ □□□□□ □□□□ □□ □□ □□ □□ □□ □□□ □□ □□□□ □□ □□□□ □□ OPTIMIZE□ □□□□□.

Answer: A (LEAVE A REPLY)

Delta Lake □□□ □□□ Delta □□□□ □□□□ □□□□ □□ □□ □□□ □□□□□. □ □□ □□□ □□□ □□ □□□□ □□□□ □□□□□ □□ □□ □□ □□□, □□ □□ □□ □□ □□ □□ □ □ □□□ □□□ □ □□□ □□□. □□□ □□ □□□ □□□□ □□ □□ □□ □□ □□□, □□□□□ □□□□□ □□□□ □□ □□□ □□□□□, □□ □□ □□□ □□ □□□□□□ □□□ □□ □□ □□□□□ □□□□ □ □□□□.

:
Databricks Delta Lake □□ □□ □□ □□□ □□ □□□□ □□□ □ □□□□: https://docs.databricks.com/delta/optimizations /□□ □□□.html

NEW QUESTION: 144

□□□□ □□□ □□□ □□□□ □□□□ □□□ □□ □ □□□□ □□ □□ □□□□ □□ □□□ □□□ □□□ □□□□. □□ □□□□ □□ □□□□□ □ □□□□□□ □□□ □□ □□□ 3□ □□□□□. □□□ □□□□ □□ □□□□□ □ □□□□□□□ □□ □□□ □□ □□□□□□, □□□ 30□□ □□□□□ □□ □□. □□ □□□□ □□□ 10□ □□□□ □□□□□□ □□□□ □□□□.

□□ □□ □□□ □□□□ □□□□ □□□□ 10□ □□□ □□□□ □□□ □□□ □, □□ □□□ □□ □□□ □□□□□?

- A. □□□ □□□ 5□□ □□□□. □□ □□□ □ □□ □□□□□ □□ □□□ □□ □□□ □ □□□ □□□□ □□ □□ □□□□ □□ □□ □□□ □□□ □ □□□□.
- B. □□□ □□□ 30□□ □□□□□. □ □□□ □□ □□□ □□ □□ □□□ □□□ □□□ □□ □□□□ □□□□ □□□ □□ □□ □□ □□□□□.
- C. □□□□□ □□□□□ □□□□ □□□□ □□□ □□□ □□□□. □□ □□□ □□□ □□□□□ □□ □□□ □□ □□ □□ □□ □□□ □□□□□□□.
- D. '□ □□ □□□' □□□□ □□□□□ Databricks □□□□ □□□□ □□□ □□□□ □□□ □□□□□ □□□□□□□. □□□ □□ □□ □□□ □□□□ □ □□□□ □□□□□.


```
(spark.read
  .format("parquet")
  .load(f"/mnt/raw_orders/{date}")
  .dropDuplicates(["customer_id", "order_id"])
  .write
  .mode("append")
  .saveAsTable("orders")
)
```

customer_id order_id □□□ □ □□□ □□□□ □□□□ □□ □ □□□ □□□ □□□□□.

[illegible][illegible]

B. orders

[illegible][illegible][illegible]

Answer: B (LEAVE A REPLY)

`dropDuplicates()` removes duplicate rows based on specified columns. In this case, it's used on the `orders` DataFrame. The `orderBy("Delta Lake")` clause sorts the resulting DataFrame by the `Delta Lake` column. Finally, `mergeInto(upsert)` writes the sorted DataFrame back to the Delta Lake table, inserting new rows or updating existing ones.

[Databricks Certified Data Engineer Professional], "Delta Lake"; Databricks, "DROP DUPLICATES".

NEW QUESTION: 147

□□□□ □□□ □□□□ □□□ □□□ □, □□ □□ □ □□□ □□□□ □□□ □□ □□□□ □□□ □□□□□?

A. □□□□: □□□ □□ □□□□:

□ □ □ □ □ : □ □ □ :

□□ □□ □□ □□: □□□

B. □□□□: □□□ □□ □□□□:

□ □ □ □ □ : □ □ :

□□ □□ □□ □: 1

C. : :

□□□ □□: □□□:

□□ □□ □□ □: 1

D. □□□□: □□ □□□ □□□□:

□□□ □□: □□□;

□□ □□ □□ □: 1

E. □□□□: □□ □□□ □□□□:

□ □ □ □ □ : □ □ :

□□ □□ □□ □:1

Answer: (SHOW ANSWER)

11

0 000 0000 0000 0000 0000 000 0000 0 00 000 00000. 00 00 0 0000 0000 000 000 0 00 00000. 00 00 000 000 00 00000
 0000 00 00 0 00000 000 0000 000 000 00000. 0000 0000 000 00000 0 00 0 000 0000 00 0 0000 00000 0000 0000 0000.
 00 0000 000 000 0000 00 00 00 00 00 10 0000 000. 000 00 00: Databricks 00 000 0000 000 000 "0000 0 00" 00, Databricks 000 "00
 00 00 00" 00.

NEW QUESTION: 148

1. 数据库设计：数据库设计是数据库系统开发的关键环节，包括需求分析、概念设计、逻辑设计和物理设计。数据库设计的好坏直接影响到数据库系统的性能、可靠性和可维护性。数据库设计的主要任务是根据用户的需求，设计出一个能够满足其要求的数据模型。数据库设计的主要步骤包括：需求分析、概念设计、逻辑设计和物理设计。

1. `orders_summary`
 2. `orders_summary`
 3. `as`
 4. `product_id`
 5. `order_count`
 6. `(`
 7. `orders_instore`
 8. `product_id`
 9. `order_count`
 10. `)`
 11. `product_id`
 12. `--`
 13. `supply_summary`
 14. `as`
 15. `select product_id, sum(supply_count) supply_count`
 16. `product_id`
 17. `product_id`
 18. `.`
 19. `product_id`
 20. `.`
 21. `stock_cte`
 22. `as (`
 23. `select nvl(s.product_id,o.product_id) as product_id,`
 24. `nvl(supply_count,0) - nvl(order_count,0) as on_hand`
 25. `product_id`
 26. `product_id`
 27. `s.product_id = o.product_id`
 28. `)`
 29. `*`
 30. `product_id`
 31. `stock_cte`
 32. `product_id`
- A.** SQL `orders_summary`
- B.** SQL `orders_summary`
- C.** SQL `orders_summary`

D. SQL 쿼리 실행을 위한 클러스터를 생성합니다.

E. SQL 쿼리 실행을 위한 클러스터를 생성하고, 클러스터에 "테스트 쿼리"를 실행합니다.

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

답변

이 문제는 SQL 쿼리를 실행하기 위해 Databricks 클러스터를 생성하고, 클러스터에 쿼리를 실행하는 방법을 묻고 있습니다. Databricks 클러스터는 SQL 쿼리를 실행하기 위해 생성되며, 클러스터에 쿼리를 실행할 수 있습니다.

Databricks 클러스터를 생성하고, 클러스터에 쿼리를 실행하는 방법을 묻고 있습니다. Databricks 클러스터는 SQL 쿼리를 실행하기 위해 생성되며, 클러스터에 쿼리를 실행할 수 있습니다.

이 문제는 Azure Databricks 클러스터를 생성하고, 클러스터에 쿼리를 실행하는 방법을 묻고 있습니다.

Cluster size	Instance type for driver(applies only to Classic Warehouses/SQL Endpoints)	Worker count
2X-Small	Standard_E8ds_v4	1
X-Small		2
Small	Standard_E16ds_v4	4
Medium	Standard_E32ds_v4	8
Large		16
X-Large		32
2X-Large		64
3X-Large		128
4X-Large	Standard_E64ds_v4	256

*The instance size of all workers is Standard_E8ds_v4.

이 문제는 SQL 쿼리를 실행하기 위해 Databricks 클러스터를 생성하고, 클러스터에 쿼리를 실행하는 방법을 묻고 있습니다.

이 문제는 SQL 쿼리를 실행하기 위해 Databricks 클러스터를 생성하고, 클러스터에 쿼리를 실행하는 방법을 묻고 있습니다.

이 문제는 SQL 쿼리를 실행하기 위해 Databricks 클러스터를 생성하고, 클러스터에 쿼리를 실행하는 방법을 묻고 있습니다.

이 문제는 SQL 쿼리를 실행하기 위해 Databricks 클러스터를 생성하고, 클러스터에 쿼리를 실행하는 방법을 묻고 있습니다.

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

Starter Endpoint

Name

Starter Endpoint

Cluster size ⓘ

X-Small

6 DBU / cluster ▾

Auto stop

☒

After 60 minutes of inactivity.

Scaling ⓘ

Min. 1

Max. 2

clusters (6 to 12 DBU)

Scale up

Scale out

databricks

NEW QUESTION: 149

□□□ □□□□□ □□ □□ □□□ □□ □□□□□□.

Which of the following is a valid data type for the products_per_order column in the orders table? products_per_order is a column in the orders table.

```
store_id INT, order_id INT, product_id INT, quantity INT, price FLOAT,
order_timestamp TIMESTAMP
```

Which of the following is a valid data type for the products_per_order column in the orders table? products_per_order is a column in the orders table.

Which of the following is a valid data type for the products_per_order column in the orders table? products_per_order is a column in the orders table?

- A. Delta Cache products_per_order column is a column in the orders table.
- B. products_per_order column is a column in the orders table.
- C. Databricks products_per_order column is a column in the orders table.
- D. products_per_order column is a column in the orders table.

Answer: D (LEAVE A REPLY)

Which of the following is a valid data type for the products_per_order column in the orders table? products_per_order is a column in the orders table.

Which of the following is a valid data type for the products_per_order column in the orders table? products_per_order is a column in the orders table.

Which of the following is a valid data type for the products_per_order column in the orders table? products_per_order is a column in the orders table.

Which of the following is a valid data type for the products_per_order column in the orders table? products_per_order is a column in the orders table.

Which of the following is a valid data type for the products_per_order column in the orders table? products_per_order is a column in the orders table.

Databricks products_per_order column is a column in the orders table.

NEW QUESTION: 151

Which of the following is a valid data type for the products_per_order column in the orders table? products_per_order is a column in the orders table.

- A. products_per_order column is a column in the orders table.
- B. products_per_order column is a column in the orders table.
- C. Kafka products_per_order column is a column in the orders table.
- D. MySQL products_per_order column is a column in the orders table.
- E. products_per_order column is a column in the orders table.

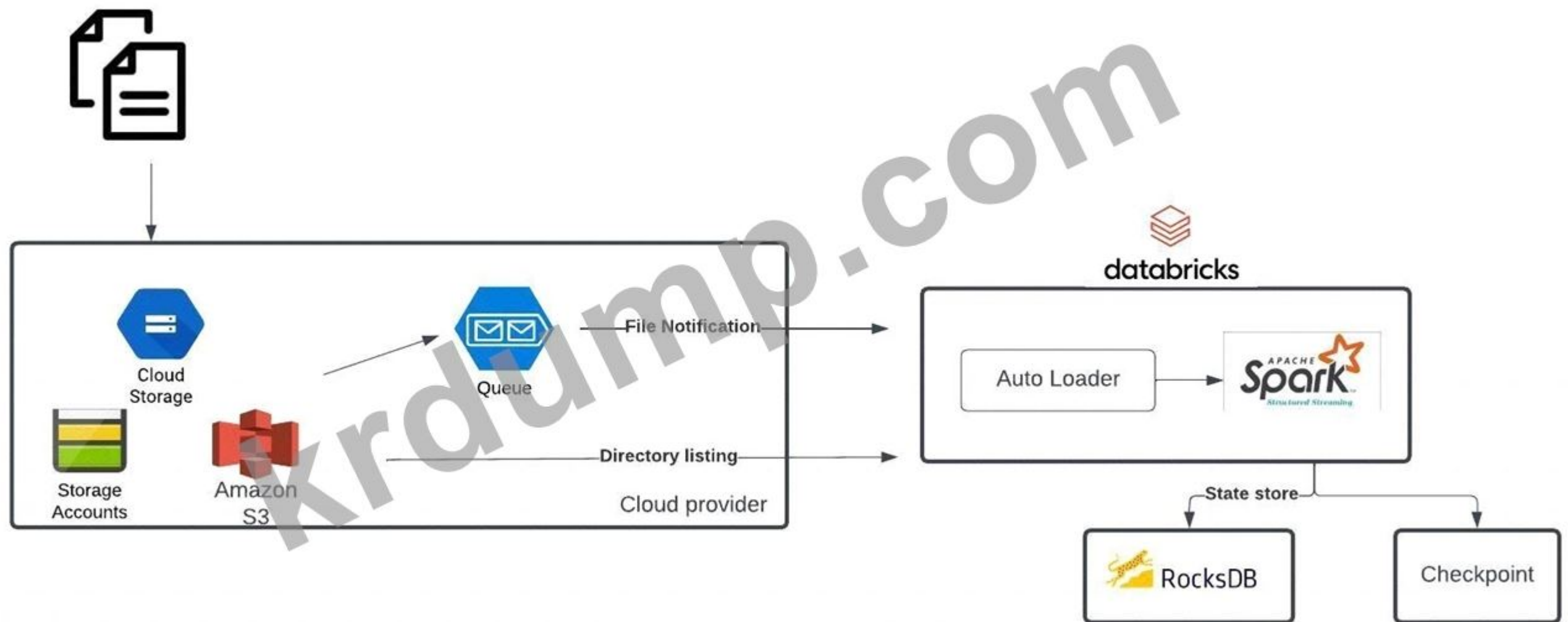
Answer: A (LEAVE A REPLY)

Which of the following is a valid data type for the products_per_order column in the orders table? products_per_order is a column in the orders table.

Which of the following is a valid data type for the products_per_order column in the orders table? products_per_order is a column in the orders table.

Which of the following is a valid data type for the products_per_order column in the orders table? products_per_order is a column in the orders table.

Auto Loader & Cloud Storage Integration



*Directory listing also supports incremental file listing

□□ □□ □□□□ □□□□ □□

Auto Loader□ □□□□ □□□□□ □□□□ □ □□ □□□□□.

1. □□□□ □□ - □□□□ □□□ □□□□ RocksDB□ □□□ □□□□, □□ □□□ □□□□□ □□□□□.

2. □□ □□ - □□ □□□ □□□□ □□ □□□□ □□ □□□□, □□□□ □□ □□□ □□ □□□ □□ □□□ □□□□ □ □□□ □ □□□□ □□□□ □□□□ □□□□ □□□□ □□□□.

[□□ □□]

□□ □□ vs □□□□ □□□□?

□□ □□

Auto Loader□ □□□ □□ □□ □□□□ □□□□□ □ □□□ □□□ □□ □□□□□ □□□□□. Auto Loader□ cloudFiles□□ □□□ □□□□ □□□□ □□□ □□□□□. □□□□ □□ □□□□□ □□□□ □□□ □□□□ cloudFiles □□□ □ □□□ □□□□ □□ □□□□ □□□□, □□ □□□□□ □□ □□ □□□ □□ □□□□ □□□ □□□□□.

□□ □□□ □□ □□ □□ □□□□ □□ □□□ □□□□□?

*□□□ □ □□□ □□□ □□□ □□ □□□□ □□□□ □□□□□ □□, □□ □□□ COPY INTO SQL □□□□ □□□□□ □□□ □□□□ □□ □□□ □□ □□□ □□□ □ □□□□.

*□□□ □□□□ □□□ □□ □□□ □□□ □□□ □□□□□? □□ □□□ □□□□ □□□ □□ □□□ □□ □□□□ □□ □ □□□ □ □□□□. □□□ □□ □□ □□□□ □□□ □□□□ □□ COPY INTO SQL □□□ □□□□ □□□ □□ □□□ □□ □□□ □ □□□□.

Databricks-Certified-Professional-Data-Engineer □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ Databricks-Certified-Professional-Data-Engineer □□! DumpTop □ □□ **Databricks-Certified-Professional-Data-Engineer** □□ □□□ □□□□□□, DumpTop Databricks-Certified-Professional-Data-Engineer □□ □□□ □□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □□ DumpTop Databricks-Certified-Professional-Data-Engineer □□□ □□□□□. <https://www.dumptop.com/Databricks/Databricks-Certified-Professional-Data-Engineer-dump.html> (217 Q&As Dumps, **30%OFF Special Discount: KrDump**)

NEW QUESTION: 152

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

```
Carts (id LONG, items ARRAY<STRUCT<id: LONG, count: INT>>)
id items email
1001[{"id: "DESW65", count: 1}] "u1@domain.com"
1002[{"id: "MYED48", count: 1}, {"id: "M27", count: 2}] "a2@domain.com"
1003[{"id: "M27", count: 1}] "a3@domain.com"
```

□□ MERGE □□ □□□ □□□ □□□□ □ □□□□ □□□□ □□ □□□□ □□□□□□ □ □□□□□.



□□ □□□□□ □□□ □□□□ □□□?

- A. □□ □□□□ □□□ □□ □□□□□ □□□ □□□□ □□□ □□□ "□□□" □□ □□□□□□□.
- B. □□□ □□□ □□□ □□ □□□□ □□□□, □□□□ □□ □□ □□□□ □□ □□□□ NULL□ □□□□.
- C. □□ □□□□ □□ □□ □□ □□□ □□□□ □□□□ □□□□□□ □□□ □□□□□.
- D. □ □□ □□□ □□ □□□□ □□□□, □□ □□□□ □□ □□□ □□□□□□ □ □□□ □□ NULL □□ □□□□□.

Answer: D (LEAVE A REPLY)

Databricks Delta □□□□□ □□□ □□ □□□ □□□□ □□, MERGE □□□ □□ □□□□ □ □□□ □□□□ Databricks□ □□□□ □□□ □□□□ □□□□ □ □□□ □□□□□□□. □□ □□□ □ □ □□□ □□ □□, Databricks□ □□ □□□□ NULL □□ □□□□□. □□□ □□ □□□□ □□ □□□□□ □□□ □□□□ □□□□, □□ □□ □□□ □□□ □□□□□ NULL □□□□□ □□ □ □□□ □ □□□ □□□□ □□□.

□□:

Delta Lake□ □□□ □□□ □□ Databricks □□: <https://docs.databricks.com/delta/delta-batch.html#schema-evolution>

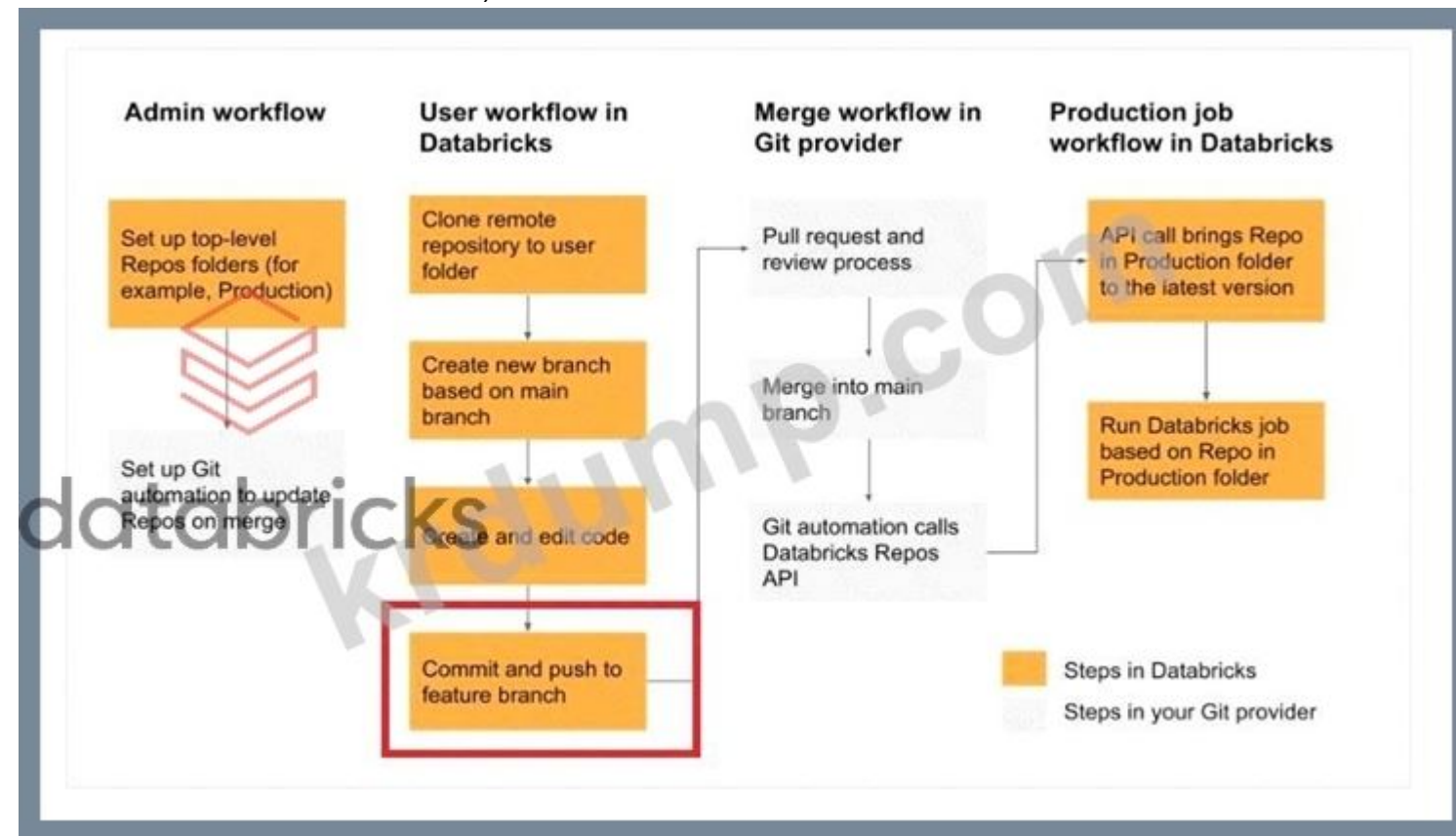
NEW QUESTION: 153

□□ □ Databricks □□□□□□ □□ □□□□ □□ □□ □□□□□?

- A. Databricks □□□□ □□□□□ □ □□□□□ □□□ □ □□□□.

E. Databricks □□□□ □□□□ □□ □□ □□□ □□□ □□ □□□ □□ □□ □□□□ □□□ □ □□□□.

□□□ □□□□ □□□□□□□□.)



INSERT

B. □□□ □□□□□□.

Answer: D (LEAVE A REPLY)

NEW QUESTION: 157

Databricks □□□ 3□□ □□□□ □□□□□□, □ □□□□ Databricks □□□□□□. □□□ A□ □□ □□□□ □□□□ □□□□. □□□ B□ C□ □□□ □□□□, □□ □□□ A□ □□ □□□□ □ □□□.

□□ A□ □□□ □□ □□ □□□ □□, □ □□ □□□ □□□□ □□□ □□□□□?

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

Answer: ([SHOW ANSWER](#))

11

Databricks 是一个云原生数据湖构建在 Apache Spark 之上，运行在 AWS、Azure 和 Google Cloud 上。它提供了一个统一的平台，用于数据摄入、存储、处理、分析和治理。Databricks 与 Lakehouse 架构兼容，支持结构化、半结构化和非结构化数据。Delta Lake 是 Databricks 开发的一个开源存储格式，用于在数据湖中实现 ACID 事务。它提供了数据完整性、一致性和可靠性。Delta Lake 与 Databricks 紧密集成，但也可以与其他数据引擎（如 Apache Spark、Presto、Trino 等）一起使用。Delta Lake 的 'V' 版本功能允许对数据进行版本控制，类似于数据库中的表。Delta Lake 还支持时间旅行，允许用户查看历史数据快照。Delta Lake 的架构设计旨在解决数据湖中的数据一致性和可靠性问题，使其成为企业级数据湖的理想选择。Delta Lake 的文档和教程可以在 Databricks 官方网站上找到。

□□□□ □□: <https://docs.databricks.com/delta/delta-intro.html#transactional-writes> □□ □□: <https://docs.databricks.com/en/workflows/jobs/conditional-tasks.html>

NEW QUESTION: 158

Databricks Auto Loader□ □□ □□ □□□ □□□□ □□□ □□□□□?

- A.** □ □□□ □□ □□□□ □□□ □□ □□□□, □ □□□ □□ Delta Lake □□□□ □□□□□ □□□ □□□□□ □□□□□.
- B.** □□□□ □□□□□ □ □□□□ □ □□ □□□□ □□□□ □□ □□□□ □□□ □□□□□. □ □□□ □□ Delta Lake □□□□ □□□□□ □□□ □□□ □□□ □□□□□.
- C.** □□ □□□□ □□ □□ □□□□□ □ □□□□ □□□ □□□ Databricks □□□ □□□□□. □ □□□□ □□□□□ □□□ □□□ □□□□ □□ □□□□ □□□□ □□□□□.
- D.** □ □□□ □□ □□□□□ □□□□ □□□□, □□ □□□□ □□ □□□□□ □□ □□□ □□□ □□□□ □□□□ □□□□□□.

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

Databricks Auto Loader¹ Delta Lake² □□□□ □□□□ □□□□□ □□□□□ □□□□□□. Auto Loader¹ □□ □□ □□□ □□ □□□□□ □□□□ □ □□□ □□□□□. □□ □□ □□□ □ □□ □□ □□ Delta Lake □□□□ □□□□□ □□□ □□□□□ □□□□□. □ □□□ □□ □□ □□ □□ □□□ □ □□ □□□□□ □□□ □□□ □□□ □ □□□□. □□ □□□ Auto Loader¹ □□ □□ □□ □□□□ □□ □□□ □□□ □□ □□ □□□ □□□□□.

□ □ □ □ :

* Databricks □□ □□ □□: □□ □□ □□□

* :

NEW QUESTION: 159

Delta Live Tables □□□□□□ □□□ □□□□□ □□□ □□□□ □□ □□□□ □□□□□ □□□□ □□□□.

□□□□□ □□□□□ □□ '□□' □□□ □□□ □ □□□□ □□□ □□□□□?

- A. 100 000 000 0 0 0000000 0000000 000000. 0000 00000 000000.
- B. 100 000 000 0000000 0000 0000 0000 00000 000000000. 0000 00000 0000000 00 00000 00000000 00000 0000000.
- C. 100 000 000 0000000 0000 0000 0000 00000 000000000. 00000000 0000 0000 0000 00000 000000 00 00 0 00000 0000000.
- D. 100 000 000 0 0 00000000 00000000 0000000. 0000 000000 00 00 0 000000 00 0000000.
- E. 100 000 000 0000000 00000000 000000000 000000 00000. 0000 000000 000000000 00 0000000.

Answer: E (LEAVE A REPLY)

□□
□□ □□ □□□ □□□ □ □□□□□ □ □□□□□□ □□□□□ □□□□□. □□ □□□□ □□ □□□ □□□□ □□□□□.
DLT □□□□□□ □□ □□□ □□□□ □□ □ □□□ □□□□□, □□ □ □□ □□□□□□ □□□ □□ □ □□ □□ □□□ □ □□□□.
□□ □ □□ □□
□□ □□□□ □□□□□□ □□□□ Delta Live Tables □□□□ □□□ □□ □□□ □□□□□.
*□□□□ □□□□ □□ □□□□□ □□ □□□□□ □□□□□.
*□□□□□ □□□ □□□ □□□□□□ □□□ □□ □□□□ □□□ □ □□□□.
□□ □□□□ Delta Live Tables □□□□ □□□ □□□□.
*□□□□ □□ □ □□□ □□ □□□ □□□□ □□ □□□ □□ □□□ □□ □□□□□ □□ □□□□□.
*□□ □□(□: □□□□ □□ □□) □□ □ □□□ □□□□□□.
□□□□□□ UI□ □□□ □□□□ □□ □□□ □□□□ □□□ □□□ □ □□□□. □□□□□ □□□□□□ □□ □□□ □□□□□.
□□ □□□ □□□□ □□ □ □□□ □□□□ □ □□□□□ □□ □□□ □□□□□.
□□ □□□ □□□□□ □□□ □□□ □□□□ □□ □□ □□ □ □□□ □□ □□□□.
□□ □□□ □□ DLT □□ □□ □□□ □□□ □□□.

<https://docs.databricks.com/data-engineering/delta-live-tables/delta-live-tables-concepts.html#delta-live-tables-c>

NEW QUESTION: 160

PySpark□□ ETL □□□ DataFrame.transform□ □□□□ □□□ □ □□□ □□□ □□□ □□□□ □□ □□□ □□□□□?

A. □□□□□ □□□:

```
def transform(self, df):  
df.withColumn("value_upper", upper(col("value")))  
□ □□□□□.  
□□□□□ = □□□□□()  
assertDataFrameEqual(pipeline.transform(test_input), expected)
```

B. def upper_value(df):

```
df.withColumn("value_upper", upper(col("value")))  
□ □□□□□.  
def filter_positive(df):  
df.filter(df["id"] > 0)  
□ □□□□□.  
pipeline_df = df.transform(upper_value).transform(filter_positive)
```

C. def upper_transform(df):

```
df.withColumn("value_upper", upper(col("value")))  
□ □□□□□.  
□□□ = □□□ □□.□□ □□  
assertDataFrameEqual(actual, expected)
```

D. def transform_data(input_df):

```
# □□ □□□ □□□ □□□□  
□□_df□ □□□□□.  
test_input = spark.createDataFrame([(1, "a")], ["id", "value"])  
assertDataFrameEqual(transform_data(test_input), expected)
```

Answer: (SHOW ANSWER)

□□□□□ □□□

Databricks Apache Spark DataFrame.transform() API को किस प्रकार का ETL ऑपरेशन माना जाता है? यह ऑपरेशन किस प्रकार का ऑपरेशन है?

A: यह ऑपरेशन (डेटा फ्रेम) को ऑपरेशन ऑपरेशन, ऑपरेशन ऑपरेशन ऑपरेशन. यह, DataFrame.transform() को ऑपरेशन ऑपरेशन, ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन.

B: यह ऑपरेशन ऑपरेशन ऑपरेशन. यह ऑपरेशन ऑपरेशन DataFrame को ऑपरेशन ऑपरेशन DataFrame को ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन (upper_value, filter_positive) को ऑपरेशन ऑपरेशन. df.transform(func) को ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ETL ऑपरेशन ऑपरेशन ऑपरेशन. यह ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन, ETL ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन.

C: यह ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन. यह, यह ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन.

D: यह ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन. DataFrame.transform() को ऑपरेशन ऑपरेशन, ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन.

यह ऑपरेशन B को PySpark ETL ऑपरेशन ऑपरेशन DataFrame.transform() को ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन.

Databricks को ऑपरेशन DataFrame.transform() को ऑपरेशन ऑपरेशन "ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ETL ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन. यह ऑपरेशन B को ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन.

NEW QUESTION: 161

user_ltv को ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन.

ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन, यह ऑपरेशन ACL को ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन.

user_ltv को ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन.

```
email STRING, age INT, ltv INT

The following view definition is executed:

CREATE VIEW user_ltv_no_minors AS
SELECT email, age, ltv
FROM user_ltv
WHERE
CASE
WHEN 'is_member'='auditing' THEN TRUE
ELSE age >= 18
END
```

यह ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन.

```
SELECT * FROM user_ltv_no_minors
```

यह ऑपरेशन ऑपरेशन ऑपरेशन?

A. 18 को ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन, यह ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन.

B. यह 17 को ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन, यह ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन.

C. 18 को ऑपरेशन ऑपरेशन ऑपरेशन null को ऑपरेशन ऑपरेशन. यह ऑपरेशन user_ltv को ऑपरेशन ऑपरेशन.

D. यह ऑपरेशन ऑपरेशन ऑपरेशन user_ltv को ऑपरेशन ऑपरेशन.

Answer: (SHOW ANSWER)

यह ऑपरेशन CASE को ऑपरेशन, यह ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ELSE को ऑपरेशन ऑपरेशन. ELSE को ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन. यह ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन 'is_member('auditing')' को ऑपरेशन ऑपरेशन ऑपरेशन, यह ऑपरेशन 18 को ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन. यह 18 को ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन.

NEW QUESTION: 162

यह ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन. यह ऑपरेशन Apache Kafka को ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन, यह ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन. यह ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन.

यह ऑपरेशन ऑपरेशन ऑपरेशन ऑपरेशन.



□□□ □□□□□□ □ □□□□□ □□□□ □□□ □□□□ □□□ □□□□ □□□ □□□□, □ □□□ □□□□□?

- A. □□□; Delta Lake □ □□□ □□□□ □□□□ □□□□.
- B. □, □ □□□ □□ □□□ □□□□ □□□ □ □□□□.
- C. □□□, Delta Lake □□□□ □□ □□□ □□□ □ □□ □□□□ □□□□□□.
- D. □; Delta Lake □ □□□ □□ □□ □□□□.
- E. □□□, □ □□□□□ □□ □□□□□ □□□□□ □□□□□.

Answer: ([SHOW ANSWER](#))

0000 000000. 00000000 0000 000000 00 00000, 00 00 0 000000 0000 000000. 00000000 000000 0000 00 0000 00 0000 DBFS 00 S3 00 00000
 000000 000000 000000. 0 000000 0000 00 000000 000000 0000 00 0000000 0000000 0000 0000. 0 00 000000 0000 0000 0000000 0000000 000000 00 00
 000000 0000 00 0000 0000 0000 0000 0 000000. 0000 00 00: [Databricks Certified Data Engineer Professional], "0000 000000" 00; Databricks 00, "0000000" 00.

NEW QUESTION: 163

DLT □□□□□□□ □□□ □□ □□□□ □□□□ □□□□□.

Raw lot□ □□□ □□ □□□□ □□ □□ □□ □□□□ □□□□□.

Bgm_stats raw_lot BPM □□□□ □□□□ □□□ □□□ □□□□□ □□□□□.

```

000 00000 00000 0000 00 0 00 0000 00 0000 00 raw_iot 00000 0000 000000 00000 0000 000 0 000 0 0000000 000 0000 00
0?

```

- A.** bpm_stats□□ skipChangeCommits □□□□ true□ □□□□□
- B.** SkipChangeCommits □□□□ true□ □□ raw_lot
- C.** bpm_stats□□ □□□□□, □□□, □□ □□□ false□ □□□□□.
- D.** raw_lot□□ □□□□□, □□□, □□ □□□ false□ □□□□□.

Answer: D (LEAVE A REPLY)

Databricks Lakehouse `raw_iottable` `pipelines.reset.allowed` `false`

[illegible]

theraw iot □□□□ □□ □□ □□ □□□□ □□□□, bpm stats□ □□ □□ □□□□ □□□□□ □ □□□□ □□ □□□□ □□ □□□□□.

□ □ □ □ :

* Databricks DLT Overview: <https://docs.databricks.com/data-engineering/delta-live-tables/delta-live-tables-overview.html>

NEW QUESTION: 164

[illegible]

- A.**
1. □□ □□ □□ □□ view_name
 2. AS SELECT * FROM table_name

B. 1. □□ □□ □ □□ □□ □□ view_name
2. AS SELECT * FROM table_name

C. 1. □□ □□ □ view_name □□ □□ □□
2. AS SELECT * FROM table_name

D. 1. □ □□ □□ □□ view_name
2. AS SELECT * FROM table_name

E. 1. □□ □ □□ □□ □□ view_name
2. AS SELECT * FROM table_name

1. `□□ □ □□ □□ □□ view_name`
2. `AS SELECT * FROM table_name`

`[REDACTED]`(Regex) `[REDACTED]` `[REDACTED]` `[REDACTED]` `[REDACTED]`. Spark Driver log4j `[REDACTED]` `[REDACTED]`, `[REDACTED]`, `[REDACTED]`, `[REDACTED]`, `[REDACTED]`, `[REDACTED]` `[REDACTED]` `[REDACTED]` `[REDACTED]` `[REDACTED]` `[REDACTED]`

`[REDACTED]` `[REDACTED]`. `[REDACTED]` `[REDACTED]` Scala, Python, Java, Spark SQL, Databricks `[REDACTED]` `[REDACTED]` `[REDACTED]` `[REDACTED]` `[REDACTED]`. `[REDACTED]`:

□□ □□□ Parquet □□□□ □□ □□□ □□ □□□ □□□ □□□□□ □□ □□□□ □□□□□. □□ □□ □□□ □□ □□□ □□□□ □□ □□□□□ □□ □□□□□.

```
(spark.read
  .format("parquet")
  .load(f"/mnt/raw_orders/{date}")
  .dropDuplicates(["customer_id", "order_id"])
  .write
  .mode("append")
  .saveAsTable("orders"))
```


<https://docs.microsoft.com/en-us/azure/databricks/security/access-control/table-acls/object-privileges#dynamic-view>

NEW QUESTION: 171

[illegible]

```
(spark.read
  .format("parquet")
  .load(f"/mnt/raw_orders/{date}")
  .dropDuplicates(["customer_id", "order_id"])
  .write
  .mode("append")
  .saveAsTable("orders")
)
```

customer_id order_id

□□ □□□□□□ □□ □□□ □□ □□□ □□□□□ □□ □□□ □□□□ □□□ □□ □□□ □□□ □□□, □□ □□ □□□ □□□□?

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

B. orders _____ , _____

C. orders □□□□ □□ □ □□ □□□□ □□□ □□□□ □□□□□. □□ □□□□ □□□ □□ □□ □□ □□□□ □□ □□ □□ □□□□ □□□□□□.

D. orders □□□□ □□ □ □□ □□□ □□□□ □□□□ □□□□□□. □□ □□□□ □□□□ □□ □□ □□ □□□□ □□ □□ □□□□ □□□□□□.

E. □□ □□□□ □□□□ □□□ □□□ □□□□ □□□□ □□□□ □□ □□ □□□ □□□□ □□ □□□□ □□□□ □□□□ □□□.

Answer: B (LEAVE A REPLY)

□ □□□ □□□ □□□ `dropDuplicates` □□□□ □□□□ □ □□□ □□ □□ □□ □□□□ □□□ □ `orders` □□□□ □□ □□□□□. □□□ □ □□□□ □□ □□ □□ □□ □□ □□ □□ □□ □□□□ □□ □□ □□ □□□□ □□ □□□□ □□ □□□□ □□ □□□□ □□ □□□□ □□ □□□□ □□ □□□□. □□ □□□□□ Delta Lake □ □□□□ `mergeInto` □ □□□ □□□(upsert) □□□ □□□ □ □□ □ □□ □□□□□. □□□ □□ □□: [Databricks Certified Data Engineer Professional], "Delta Lake" □□; Databricks □□, "DROP DUPLICATES" □□.

NEW QUESTION: 172

00 000 00000 00 000 00 000, 00 0 000 ETL 0000 00 000 00000 00 000 00 00 00000. 000 000 00000 00 00 00 0 00 00 000 0 0
 000?

A. ☐☐ ☐☐☐

B. ☐ ☐ ☐ ☐ ☐

C. SQL ☐☐☐☐☐

D. ☐☐ ☐☐☐ ☐☐☐

E. □□□□ □□□ □□□□ □□□□

Answer: ([SHOW ANSWER](#))

11

[illegible]

DLT□ □□□ □□□ □□ DAG □□ □□□ □□□□ □□□ □□□□ □□□□□□. □□□ □□ □□□□□ □□ □□□ □□□□ □□□ DAG□ □□□□□.

1. □□□ □ □□ □□ □□ □□

2. $\square\square\square\square\square\square * \square\square\square\square\square\square.$

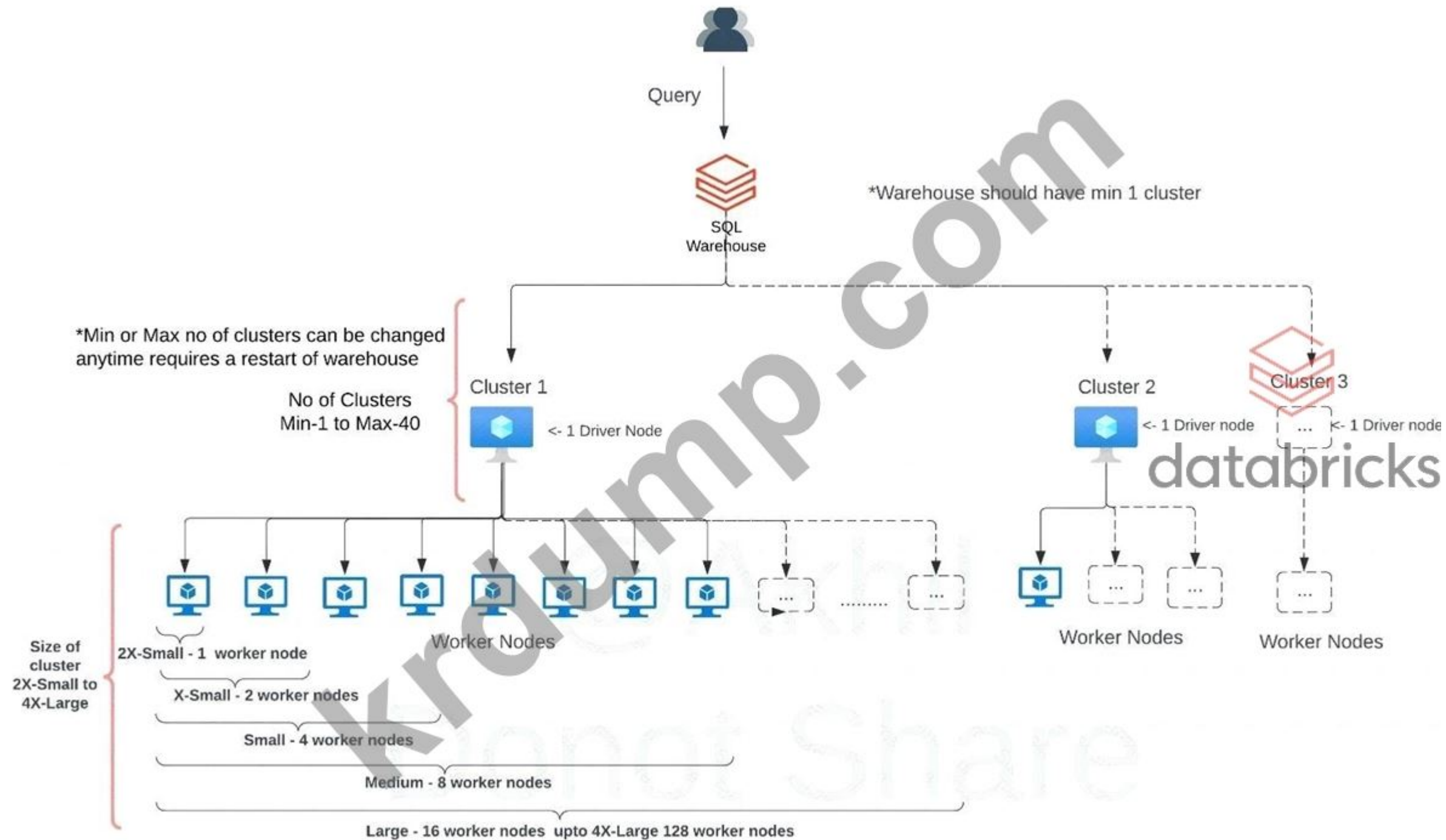
- 3.
4. `sales_orders_raw` `sales_orders_raw`.
5. `select * from sales_orders;`
- 6.
7. `sales_orders_cleaned` `sales_orders_cleaned`.
8. `as`
9. `sales_orders_raw.*`
10. `live.sales_orders_raw` `s`
11. `customer_id` `customer_id`
12. `c.customer_id = s.customer_id`
13. `c.city = 'LA';`
- 14.
15. `sales_orders_in_la` `sales_orders_in_la`.
16. `sales_orders_cleaned` `sales_orders_cleaned`.
- `sales_orders_cleaned` `sales_orders_cleaned`.



DELTA LIVE TABLES `sales_order_in_la`,
<https://databricks.com/product/delta-live-tables>

4. □□□□ □□(2X-Smal□□ ...4XLarge□□)□ □□□□ □□ □□□□□ □□□□ 3X-Large □□□□ □□□ □□□□ □□□(□□)□ □□ □□□ □□□□□□ 20□□ □□□ □ □□ □□□□ □□ □□ □□ 10□□ □□□ □□□ □ □□□□.
- 1, max1) 10□□ □□□ □□□ □□□□ □□ □□□ 10□□ □□□ □ 10□□ □□□ □□□ □□□ □□□□□ □□□□□.
5. □□□□□ □□□□ □□□ □□□ □□ □□□ □□□ □ □□□□. □□ □□, 2X-Small □□□ □□□□□□□ □□ □□□ 1□□ □□□ □□, □□□□□ □□□ X-Small□ □□□□ 30□ □□ □□ □ □ □□□□.
- 2X-Small□ □□ □□□ 1□□□ X-Small□ □□ □□□ 2□□□□ □□□ □ □□ □□□ □□□□ □ □□□ □□□□ □□□□□. (□□: □□ □□□□ □□□□, □□ □□□ □□□□ □□ □□□ □□ □□□□□ □□ □□□□□ □□□□ □□ □□□□.)
6. □□□ □□□□□□ □□ □□ □□□□□ □□ □ □□□, □□ □□□ □□□□□ □□□. □□□□□□ □□□□ □□□□(□□ 1□, □□ 2□)□ □□□ X-Small □□□□ □□□ □□□ □□, Databricks□ □□□□ □□□ □□□□ □□ □□□□□ □□□□□□. □□ □□ □□□□□□ 2□□ □□□□(□□ 1□, □□ 2□)□ □□□□□ □□□□ □□ □□□□ 20□□ □□□ □□□□ 10□□ □ □□ □□□□ □□□ □□□ □□□□ □□ □□□. □□□□ Databricks□ □□□□ □ □□ □□□□□ □□□□ □□□□ □□ 10□□ □□□ □ □□ □□□□□ □□□□□□□□.
7. □□ □□□ □ □ □□□ □□□□□ □□ □□□□ □□□□. □□□□ □□□□□ □□□□ □□ □□□ □□□□ □□ □□□□ □□ □□□ □□□ □□ □□□□□ □□□□□. □□ □□□ □□□□□ □□ □□□ □□□□□□.

SQL Warehouse Overview



*All clusters in a warehouse should be of same size
X-Small or Medium or Large ...

Warehouse cluster size(Small,Medium ... No of worker nodes) can be changed anytime but a restart is required

*All worker nodes in Classic warehouse use `Standard_E8ds_v4`

*Driver size changes(bigger virtual machine) as cluster size(Small,Medium.. No of worker nodes) increases

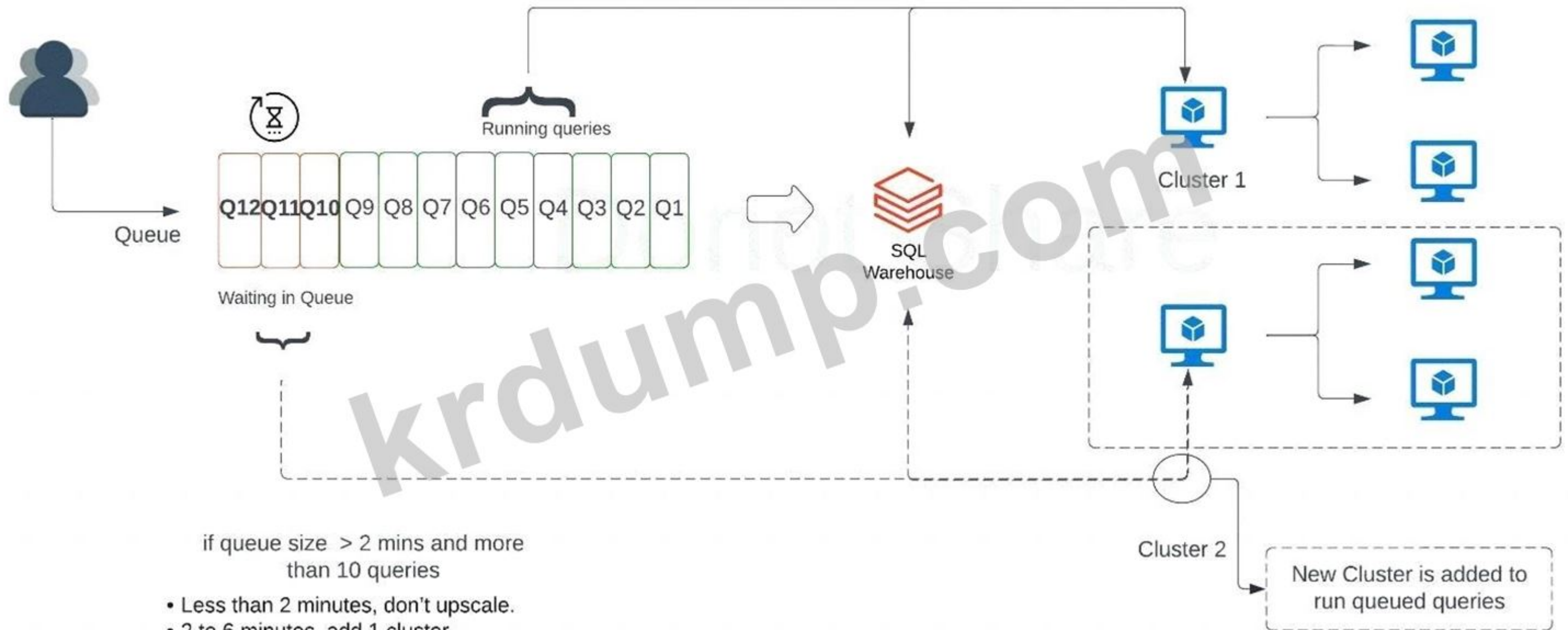
SQL Warehouse (SQL Warehouse) is a managed service that allows you to run SQL queries on a cluster of worker nodes.

The SQL Warehouse is a managed service that allows you to run SQL queries on a cluster of worker nodes.

Databricks SQL Warehouse is a managed service that allows you to run SQL queries on a cluster of worker nodes. The warehouse is a managed service that allows you to run SQL queries on a cluster of worker nodes.

SQL Warehouse Scale out

Scaling: Min 1 Max 3
Cluster Size: XSmall



if queue size > 2 mins and more than 10 queries

- Less than 2 minutes, don't upscale.
- 2 to 6 minutes, add 1 cluster.
- 6 to 12 minutes, add 2 clusters.
- 12 to 22 minutes, add 3 clusters.

* Please note 2 mins wait is only for classic warehouses for server less the wait can be much less.


databricks

□□ □□ □ □□ □□ □ □□ □□ □ □□ □□(2X-Small...□□)□ □□□□□□.
...4XLarge)□ □□□□ □□ □□□ □□□□ SQL Endpoint(SQL Warehouse) □□□□□□ □ □□ □□□□□ □□□ □ □□ □□ □□□ □□□□□□□. □□ □□□□□□ □□□□ □□ □□ □□□ □
□□□□ □□□□□□ □□ □□□□ □ □ □□□□□.

Starter Endpoint

Name

Starter Endpoint

Cluster size ⓘ

X-Small

6 DBU / cluster

Auto stop

After 60 minutes of inactivity

Scaling ⓘ

Min. 1

Max. 2

clusters (6 to 12 size)

Scale up

Scale out

□□□ □□□□ □□□ □□□ □ □ □□□(□□ □□□□ □□ □□ □□)?
□□ □□□ □□□□□□ □□□□ □□□□ □□ □□□□ □□□□□ □□□ □□□ □ □□□□(□□ □□). □□□□□ □□ □□□ □□ □□□ □□□ □□□ □ □□ □□□□ □□□□□. □
□□ □□□□ □□ □□□ □□ □□□□□□ □□□□□ □□□□ □□ □□□ □□□ □ □□□□. □□□□□ □□□ □□ □□ DBU □□□□ □□□□ □□□ □□ □□ □□□ □□□□□ □□□□
□.



NEW QUESTION: 175

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

□□ □□□□ □□□ □□ □□□□ □□ Parquet □□□□ □□□□ □□□□.

item_id INT, user_id INT, review_id INT, rating FLOAT, review STRING

□□ □□□ □□□□ □□ □□ □□□ □□□□ □□□□. □□□ □□ □□ □□ □ □□□ 30□□ □□ □□□ □□□ □□□□□ □□□.

□□ □□□ □□□□□ □ □□□□□ Delta Lake□ □□□□ □□ □□□ □□□ □□□□□□□□.

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

A. Delta Lake □□□ □□□□□□□ □□ □□ □□□ □□□ □□□□□ □□ □□□□.

B. Delta Lake□□□ □□□ □□□□ □□□ □ □□□□.

C. □□ □□□ □□□□□ ZORDER ON □□□ □□□□ □□□.

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

E. □□ □□ □□□ □□ □ 4□ □□□ □□□□□.


```
(spark.read
  .format("delta")
  .option("readChangeData", True)
  .option("startingTimestamp", '2021-08-22 00:00:00')
  .option("endingTimestamp", '2021-08-29 00:00:00')
  .table("user_lookup")
  .createOrReplaceTempView("changes"))
```

```
spark.sql("""
DELETE FROM user_aggregates
WHERE user_id IN (
  SELECT user_id
  FROM changes
  WHERE _change_type='delete'
)
""")
```

user_id is a column in user_lookup table, and user_aggregates table has a column user_id. How can you delete rows from user_aggregates where user_id is in the list of user_id from changes table where _change_type is 'delete'?

- A. Use: DELETE FROM user_aggregates WHERE user_id IN (SELECT user_id FROM changes WHERE _change_type='delete').
- B. Use: Delta Lake ACID DELETE FROM user_aggregates WHERE user_id IN (SELECT user_id FROM changes WHERE _change_type='delete').
- C. Use: DELETE FROM user_aggregates WHERE user_id IN (SELECT user_id FROM changes WHERE _change_type='delete').
- D. Use: Delta Lake DELETE FROM user_aggregates WHERE user_id IN (SELECT user_id FROM changes WHERE _change_type='delete').

Answer: (SHOW ANSWER)

Delta Lake DELETE FROM user_aggregates WHERE user_id IN (SELECT user_id FROM changes WHERE _change_type='delete'). is the correct answer. Delta Lake is a storage layer that sits on top of S3, and it supports ACID transactions. This means that you can delete rows from a table in Delta Lake, and the delete operation will be atomic and consistent. The other options are incorrect because they either use the wrong syntax or they don't use Delta Lake's ACID capabilities.

NEW QUESTION: 178

A table named pii_test has a column named name of type STRING. The table has a comment "Contains PII". The table is created with the following SQL DDL:

CREATE TABLE pii_test (id INT, name STRING COMMENT "PII") COMMENT "Contains PII" TBLPROPERTIES ('contains_pii' = True)

```
CREATE TABLE dev.pii_test
(id INT, name STRING COMMENT "PII")
COMMENT "Contains PII"
TBLPROPERTIES ('contains_pii' = True)
```

Which of the following statements is true?

- A. dev.pii_test is a table in the dev database.
- B. dev.pii_test is a table in the pii database.

C. dev.pii ☐☐☐☐ TBL ☐☐ ☐☐

D. ☐ ☐ ☐ ☐ dev.pii ☐ ☐ ☐

E. ☐☐ ☐☐☐☐ ☐☐

Answer: A (LEAVE A REPLY)

0000 0000 0000 0 00 00 0000 00000000 00000 0000 0 00 000000. 00 0000 00 00 00(PII)0 0000 00 00000 00000 0000 0000 0000 00000. 00000
 0 00, 0000 0000 00000 0000 00 0000 00 "contains_pii"0 true0 00000 00 000000. DESCRIBE EXTENDED 0000 000000 0000, 00, 00 0 0000 00 0000 0000 00
 00 0 000000. dev.pii_test 000000 0 0000 000000 SQL DDL 00 0000 00 0000 0 00, 0000 00 0 0000 00 0000 00000 000000 00000 000000. 00
 0 00: [Databricks Certified Data Engineer Professional], "Lakehouse" 00; Databricks 00, "DESCRIBE EXTENDED" 00.

NEW QUESTION: 179

□□□ □□□□ □□ GDPR □□□ □□ □□□ □□□ □□□□ □□□□ □□□□. □□□ Delta Lake□ namedusers □□□□□ □□□□ □□□□ □ □□□ □□ □□□ □□□□□ □□ □□ □□□□.

```
DELETE FROM users
WHERE user_id IN
    (SELECT user_id FROM delete_requests)
```

user_id 00 00 000 delete_requests 0000 0000 00 00000 00000 0000 0000 0, 00 0000 000000 00000 0000 00000 0 00 0000 0 0000 00 000000 00
0 00000 0000 00000 0 0000 000000?

A. □; Delta Lake□ ACID □□□ □□ □□□ □□□ □□□□□ □□□□ □□ □□□□ □□□□□ □□□□□□ □□□□□□.

B. □□□. □□ □□□ □□□□□ □□□□ □□□ □□□□ □□ □□□□ □□□ □□□□ □□□ □ □□□□.

C. □; □□ □□□ □□□□ □□□ □□ □□□ □□□ □□□□ □□ □□□□□□□.

D. □□□. Delta Lake □ □□ □□□ □□ □□□ □□ □□□ □□ ACID □□□ □□□□□.

E. □□□; □□□ □□□□ □□□ □□□ vacuum □□□ □□□□ □□□□ □□□ □□□ □□□□ □□□□ □□ □□□□ □□ □□ □□□ □ □□□□.

Answer: E (LEAVE A REPLY)

```

0 0000 `DELETE FROM` 0000 00000 0000 0000 00 00000 0000 `delete_requests`00 000000 0000 00000 0000 00000 0000 00000 00000 000000.

```

[illegible]

□□□ □□ □□: [Databricks □□ □□□ □□□□ □□□], "Delta Lake" □□; Databricks □□, "□□□□□ □□" □□; Databricks □□, "Delta □□□□□ □ □□ □□□□ □□ □□ □□" □□.

NEW QUESTION: 180

□□□□ □□□□ □□□ □□□ □□ □□□□ □□□□□?

A. ☐☐☐☐ ☐☐☐ ☐☐☐☐ ☐☐☐ ☐ ☐☐☐☐.

B. □□□□ □□□ □□□□ □□□ □ □□□□.

C. □□□ □□ □□□ □□□□ □□□□□.

D. □□□ □□□ □□□□ □□□□□□.

E. ☐☐ ☐☐ ☐☐☐☐ ☐☐☐☐☐☐

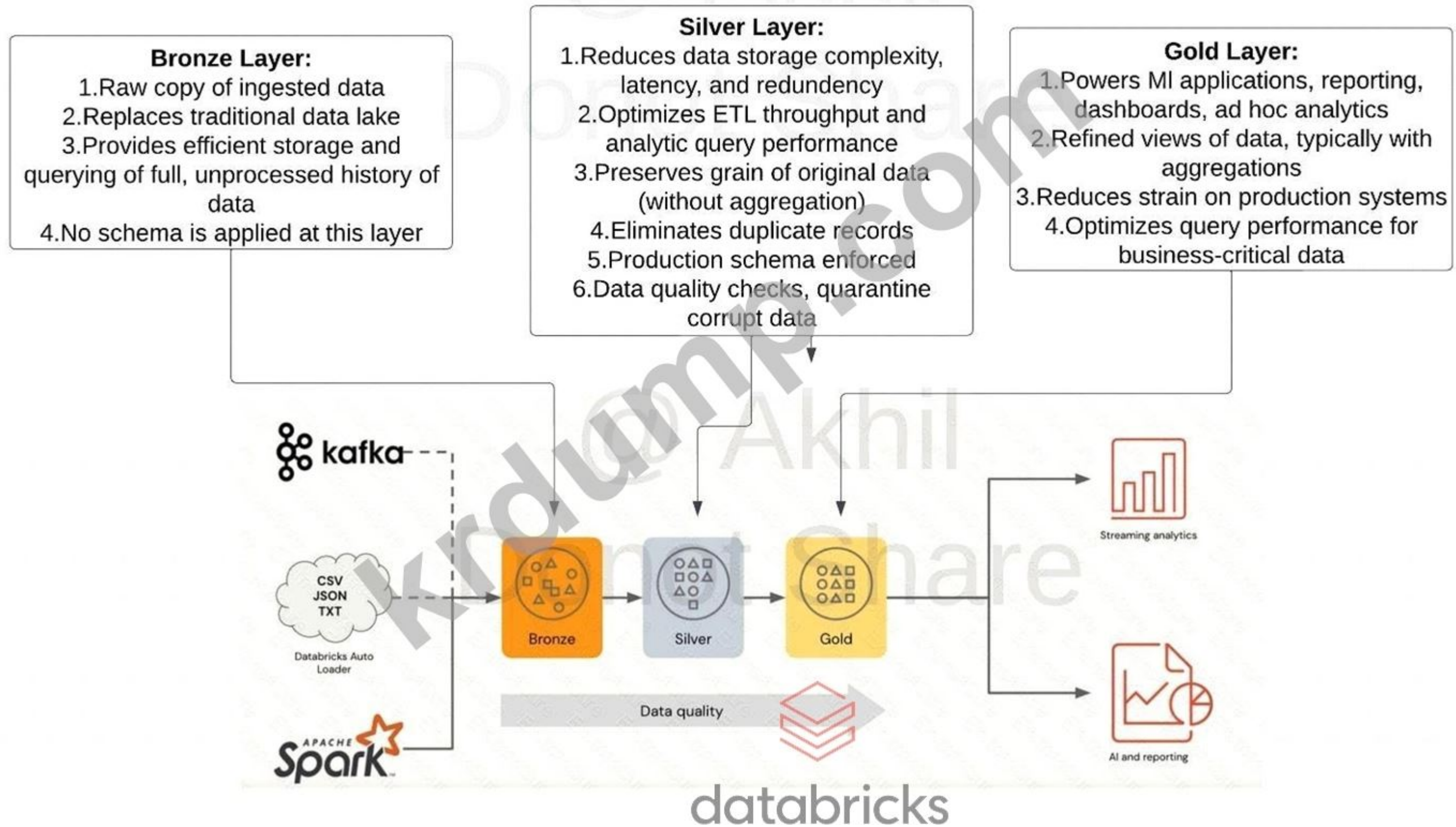
Answer: B (LEAVE A REPLY)

11

□□□□ □□□□ - □□□□□□

□□ □□ □□: □□ □□□□仔□ □□□□ □□□□ □□□□ □ □(□□, □, □)□ □□□ □□□□□□. □ □□ □□□□ □□□ □□ □□□ □□□ □□□ □□□□.

□□□□□ □□□□□ □□□□ □□□□. □□□□□ □□ □□□□ □ □□□□ □□□□ □□□□.
□□ □□ □□ (□□□□ □□)



□□□ □ □□□ □□□□□ □ 400GB□ RAM, □ 160□□ □□, □□□ VM□ □□□ Executor□ □□□□ □□□ □□□□□.
□□ □□□ □□□□ □□□ □□□ □□□□ □, □□ □□□□ □□ □ □□ □□□ □□ □□□ □□□□□?

A. * $\square$ VM $\square$; 1

* □□□□ 400GB

* 160□ □□/□□□

B. * ☐ VM ☐: 8

* □□□□ 50GB

* 20□ □□/□□□

C. * ☐ VM ☐: 4

* □□□□ 100GB

* 40□ □□/□□□

D. * $\square$ VM $\square$: 2

* □□□□ 200GB

* 80□ □□/□□□

Answer: B (LEAVE A REPLY)

1. 数据源。数据源是数据仓库的入口，可以是数据库、数据湖、数据流等。在数据仓库中，数据源可以是关系型数据库、NoSQL 数据库、数据湖、数据流等。数据源可以是内部数据源，也可以是外部数据源。数据源可以是结构化数据，也可以是非结构化数据。数据源可以是实时数据，也可以是批量数据。数据源可以是单一数据源，也可以是多个数据源。数据源可以是单一数据源，也可以是多个数据源。数据源可以是单一数据源，也可以是多个数据源。

[illegible]

```
[Databricks □□ □□□ □□□□ □□□]□ "□□ □□" □□ □ Databricks □□□□ "□□□□ □□" □□□ □□□□□□.
```

Databricks-Certified-Professional-Data-Engineer □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ Databricks-Certified-Professional-Data-Engineer □□! DumpTop □ □□ **Databricks-Certified-Professional-Data-Engineer** □□ □□□ □□□□□□, DumpTop Databricks-Certified-Professional-Data-Engineer □□ □□□ □□□□□□□ □□□ □□□□□□□. □□□□ □□□ □□□□ □□ DumpTop Databricks-Certified-Professional-Data-Engineer □□□ □□□□□. <https://www.dumptop.com/Databricks/Databricks-Certified-Professional-Data-Engineer-dump.html> (217 Q&As Dumps, **30%OFF Special Discount: KrDump**)

NEW QUESTION: 182

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

□□ □□□ □□□ □□□□ □ □□□, □□ □ □□ □□ □□□ □□□□ □□□ □□□□ □□□□ □□□ □ □□□ □□□ □□□□ □□□ □□□□ □□□ □ □□□□?

A. delta.deduplicate = true ☐ ☐☐ ☐☐☐

[illegible]

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

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

E. □□ □□□□ □□□□□ Delta Lake □□□ □□ □□ □□□ □□□□□.

Answer: ([SHOW ANSWER](#))

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

`MERGE INTO target\_table USING source\_table ON target\_table.unique\_key = source\_table.unique\_key WHEN NOT MATCHED THEN INSERT \*` □ □□□ □□ □□□□□ □□ □□□□ □□ □□ □ □□□ □□□□, □□ □□□□ □□□□ □□□□□. □□□ □□ □□ □□□□ □□ □□□□ □□□□ □□ □□□ □ □□□□.

□□ □□:

* <https://docs.databricks.com/delta/delta-update.html#upsert-into-a-table-using-merge>

* <https://docs.databricks.com/delta/delta-update.html#insert-only-merge>

NEW QUESTION: 183

□□□ □□□□□ □□ Delta Lake □□□□ □□ □□□□□□ □□ Databricks SQL □□□ □□□ □□□□□□. recent_sensor_recordings □□□□□ □□ 5□ □□□ □□□ □□ □□ ID, □□□□□ □ □□□ □□□□ □□□□.

□□ □□□ □□□ □□□□ □ □□□□□.

```
SELECT MEAN(temperature), MAX(temperature), MIN(temperature)
FROM recent_sensor_recordings
GROUP BY sensor_id
```

□□ □□□ 1□□□ □□□□□ □□□□ □□□ □□ 10□ □□□ □□□□□. □□ □□(□□)□ 120°F(□□ □ 38□)□ □□□□ □□□ □□□□□ □□□□ □□□□. □□□ □□ 1□ □□□□ □□□ □□.

□ □□□ 3□ □□ □□□□ □□□□□ □□□□, □□ □ □□ □□□ □□□ □□□□ □□□?

A. □□ □□□ □□ □□□ □□□ 3□ □□ □□□□ □ 120□□ □□□□□□.

B. recent_sensor_recordings □□□□ □□□ □ □ □□ □□□□ □□ □□□□ □□□□□.

C. □□ □□□ 3□ □□ □□□ □□□□□□ □□ □ □□ □□□□□□□.

D. □□ □□□ □□□□ □□□ □□ □□□ □□□ 3□ □□ □□□□ □ 120□□ □□□□□□.

E. □□□ □□□ □□□□ □□□ □□ □□□ □□□ 3□ □□ □□□□ □ 120□□ □□□□□□.

Answer: E (LEAVE A REPLY)

□□□ □□□□□. □□□□ sensor_id □□ GROUP BY □□ □□□□ □□ □□□ □ □□□ □□ □□□ □□□□□ □□□□□. □□ □ □□□□ □□ □□□ 120°F(□□)□□ □□□ □□□ □□□□ □. □, □□ □□□ □□□□ 3□ □□□□ □□ □□□ 120°F(□□)□ □□□ □□□□□. □□ □□□ □□ □□□ 120°F(□□) □□□□ □□□□ □□□ □□□□□. □□□□ □□: [Databricks Certified Data Engineer Professional], "SQL Analytics" □□; Databricks □□, "Alerts" □□.

NEW QUESTION: 184

DLT □□□□□□□ □□□ □□ □□□□ □□□□ □□□□□.

Raw_lot□ □□□ □□ □□□□ □□ □□ □□ □□□□ □□□□□.

Bgm_stats□ raw_lot□ BPM □□□□ □□□□ □□□ □□□ □□□□□ □□□□□.

□□□ □□□□□ □□□□□ □□□□ □□ □ □□ □□□□ □□ □□□□ □□ raw_iot □□□□□ □□□□ □□□□□ □□□□□ □□□□ □□□ □ □□□ □ □□□□□□ □□□ □□□□ □□ □?

A. bpm_stats□□ □□□□□, □□□, □□ □□□ false□ □□□□□.

B. bpm_stats□□ skipChangeCommits □□□□ true□ □□□□□

C. raw_iot□□ □□□□□, □□□, □□ □□□ false□ □□□□□.

D. raw_lot□ SkipChangeCommits □□□□ true□ □□

Answer: C (LEAVE A REPLY)

NEW QUESTION: 185

□□ □□□ □□□□□□ DataFrame df□ □□□□ □□□□ □□□ □□□□ □□□□ □□□ □□□□□□ □□□□□ □□□ □□□□□. □ □□□□□□ □□□ □□ 5□ □□□□ □□ □□□ □□ □□□ □□□□ □□□. □□, □□ □□□□ □□□□ □□□□ 10□ □□ □□ □□ □□□ □□□□ □□□.

□□□□ □□□□□□ df□ □□□ □□ □□□□ □□□ □□□□.
"□□ ID INT, □□□ □□ TIMESTAMP, □□ FLOAT, □□ FLOAT"

□□ □□:

```
if._____  
    .groupBy(  
        window("event_time", "5 minutes").alias("time")  
        "device_id"  
    )  
    .agg(  
        avg("temp").alias("avg_temp"),  
        avg("humidity").alias("avg_humidity")  
    )  
    .writeStream  
    .format("delta")  
    .saveAsTable("sensor_avg")
```

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

- A. withWatermark("event_time", "10□")
- B. awaitArrival("event_time", "10□")
- C. await("event_time + '10□'")
- D. slidingWindow("event_time", "10□")
- E. delayWrite("event_time", "10□")

Answer: A (LEAVE A REPLY)

□□□ A. withWatermark("event_time", "10 minutes")□□□. □□□□ □□ □□ □□□□ □□ 10□ □□ □□ □□ □□□ □□□□□ □□□□ □□□□□. withWatermark □□□□ □□ □□□□ □□ □□□□□ □□□□ □ □□□□□. □□□□□ □□□□□ □□ □□ □□□□ □□□□ □□□ □□□□ □□□□ □□□□□ □□□□□. □ □□ □□□□□ 10□□□ □□□□□. □□ □□□

Structured Streaming□□ □□□□□ □□ □□□ □□□ □□ □□□ □□□□□□□□. □□ □□:

* □□□□: <https://docs.databricks.com/spark/latest/structured-streaming/watermarks.html>

* □□□□ □□: <https://docs.databricks.com/spark/latest/structured-streaming/window-operations.html>

NEW QUESTION: 186

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

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

Answer: A (LEAVE A REPLY)

□□□ □□□□□. □□□ □□□□(Structured Streaming)□□□ □□□ □ □□□ □□ □□ □□□□ □□ □□□-□□ □□□ □□□□□. □□ □□ □□□□ □□□□ □□ □□ □□ □□ □□ □□□ □□ □□ □□ □□□ □□ □□□□□□ □□ □□ □□□□□□. □ □□ □□□-□□ □□□ □ □□□□□□□□ □□ □□□□□□ □□□ □□ □□ □□□□ □□ □□□ □□□□ □□□, □ □□□□

A lakehouse has the following key features:

- **Transaction support:** In an enterprise lakehouse many data pipelines will often be reading and writing data concurrently. Support for ACID transactions ensures consistency as multiple parties concurrently read or write data, typically using SQL.
- **Schema enforcement and governance:** The Lakehouse should have a way to support schema enforcement and evolution, supporting DW schema architectures such as star/snowflake-schemas. The system should be able to **reason about data integrity**, and it should have robust governance and auditing mechanisms.
- **BI support:** Lakehouses enable using BI tools directly on the source data. This reduces staleness and improves recency, reduces latency, and lowers the cost of having to operationalize two copies of the data in both a data lake and a warehouse.
- **Storage is decoupled from compute:** In practice this means storage and compute use separate clusters, thus these systems are able to scale to many more concurrent users and larger data sizes. Some modern data warehouses also have this property.
- **Openness:** The storage formats they use are open and standardized, such as Parquet, and they provide an API so a variety of tools and engines, including machine learning and Python/R libraries, can efficiently access the data **directly**.
- **Support for diverse data types ranging from unstructured to structured data:** The lakehouse can be used to store, refine, analyze, and access data types needed for many new data applications, including images, video, audio, semi-structured data, and text.
- **Support for diverse workloads:** including data science, machine learning, and SQL and analytics. Multiple tools might be needed to support all these workloads but they all rely on the same data repository.
- **End-to-end streaming:** Real-time reports are the norm in many enterprises. Support for streaming eliminates the need for separate systems dedicated to serving real-time data applications.

□□ □□
□□□ □□□

NEW QUESTION: 189

Delta Lake □□□□ □□□□ □□ □□□□□ □□□ □□ □□□ □□ □□□□□□ □□□ □□□□ □□□ □□□□ □□□□. □□, activity_details □□□□ □□□ □□ □ □□ □□□ □□
□□ □□ □□ □□ □□ □□□ □□□ □ □□ □□□ □□□.
□□ □□□□□ Delta Lake □□□□ CHECK □□ □□□ □□□□ □□ □□□ □□□□□□.

CONSTRAINT valid_coordinates
(
 cude >= -90 AND
 cude <= 90 AND
 ... >= 100 AND



databricks

☐☐ ☐ ☐ ☐☐☐ ☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐☐

- Answer: C (LEAVE A REPLY)**

[illegible]

NEW QUESTION: 190

☐☐☐ ☐☐☐☐ ☐☐☐ ☐☐ ☐☐☐☐☐☐

□ □□□ □□□ □□□ □□□?

- Answer: B (LEAVE A REPLY)**

Databricks□□ □□ □□□□ □□□□ □□ □□ □□□□ DBFS□□ □□□□□□ dbutils.fs.mount()□ □□□□□ □□□□□□ □□ □□□□□□□□□□.

Which of the following is true?

- A. DBFS is a Unix-style file system that is used to store data in the Hadoop Distributed File System (HDFS).
- B. DBFS is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS).
- C. DBFS is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS).
- D. Databricks is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS).
- E. DBFS is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS).

Answer: A (LEAVE A REPLY)

DBFS is a Unix-style file system that is used to store data in the Hadoop Distributed File System (HDFS). 1 DBFS is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS). 2 DBFS is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS). 3 DBFS is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS).

FUSE is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS). 4 DBFS is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS). 5 DBFS is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS).

DBFS, Azure Databricks, Azure Databricks, FUSE, DBFS

NEW QUESTION: 192

Which of the following is true? DBFS is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS). DBFS is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS).

SQL is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS).

- A. DBFS
- B. DBFS
- C. DBFS
- D. GO
- E. DBFS

Answer: (SHOW ANSWER)

NEW QUESTION: 193

Which of the following is true? DBFS is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS). DBFS is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS). DBFS is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS).

- A. DBFS is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS).
- B. DBFS is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS).
- C. DBFS is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS).
- D. Spark is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS).

Answer: A (LEAVE A REPLY)

DBFS is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS). DBFS is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS). DBFS is a distributed file system that is used to store data in the Hadoop Distributed File System (HDFS).

Databricks Delta Lake : <https://docs.databricks.com/delta/delta-batch.html#schema-management>

Delta Live Table ☐☐☐☐☐☐ STREAMING LIVE TABLE ☐ ☐☐☐☐ ☐ ☐☐ ☐☐ ☐☐☐☐.

□□□ □□□□□ □□□ □□□□ □□ □□□□ □□□□□.

□□□□□□ □□□□□□□□ '□□'□ □□□□□□□□□?

□ □ □

□ □ □ □ □ □ □ □ □ □ □

□□□□□ □□ □□□□ □□□□□□ □□□□ □□□□□.

□□□□□ □□□ □□□ □□ □□□□ □□ □□□□□□.

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

Answer: B (LEAVE A REPLY)

[illegible]

B. □□□□ □□□□ □□□□ □□□□.

C. □□□□ □□□□ □□□□ □□□□.

D. □□ □□□□ □□□□ □□□□□□.

E. D. # □□□□□□□□□□□□□□□□

[illegible]

Answer: ([SHOW ANSWER](#))

11

00000000 - 00000000 00000000 00000000 00000000. 00000000 00000000.

```

Della Lake 00000 0000 0000 000000 00 00000000 00000 0000 00 00000 00000.
year_id LONG post_text STRING post_id STRING longitude FLOAT latitude FLOAT post_time TIMEST

```

□□ < 20 □□ > -20

□□□ □□□ □□□ □

^ □□ □□□□ □□□□ □□□□□ □□□□ □□□□ □□□

[illegible]

```
{
  "chicago": [
    {
      "date": "01-01-2021",
      "temp": [25, 28, 45, 56, 39, 25]
    },
    {
      "date": "01-02-2021",
      "temp": [25, 28, 49, 54, 38, 25]
    },
    {
      "date": "01-03-2021",
      "temp": [25, 28, 49, 58, 38, 25]
    }
  ]
}
```

□□ □□: 58

- A. 1.select max(raw.chicago.temp[3]) from temp_data
- B. 1.select array_max(raw.chicago[*].temp[3]) from temp_data
- C. 1.select array_max(from_json(raw['chicago'].temp[3],'array<int>')) from temp_data
- D. 1.select array_max(from_json(raw:chicago[*].temp[3],'array<int>')) from temp_data
- E. 1.select max(from_json(raw:chicago[3].temp[3],'array<int>')) from temp_data

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

□□
□□: □ □□□ □□□□ □□□□. □□ □□□ □ □□ □□□ □□□ □□□□ □□□, □□ □□□ □□□ □□□ □□ □□□□ □□□□□.
□□ □□□ □□□□□ □ □□ □□□ □□ □ □□□□. □□ □□□ □ □□□.
<https://docs.databricks.com/optimizations/semi-structured.html>
□□□ □□□ □□ □□□□□.
□□□ □□□ □□□□ □□□□□□□□.

```
select raw:chicago[*] from temp_data
```



Select Chicago and all elements inside Chicago

```
[{"date": "2021-01-01", "temp": [25, 28, 45, 56, 39, 25]}, {"date": "2021-01-02", "temp": [25, 28, 49, 54, 38, 25]}, {"date": "2021-01-03", "temp": [25, 28, 49, 58, 38, 25]}]
```

```
select raw:chicago[*].temp[3] from temp_data
```



select 4th element in the temp array, index starts from 0,1,2,3

```
'[56,54,58]'
```

```
select from_json(raw:chicago[*].temp[3], 'array<int>') from temp_data
```



```
[56,54,58]
```

convert string to array using from_json

```
select array_max(from_json(raw:chicago[*].temp[3], 'array<int>')) from temp_data
```



```
58
```

use array_max to calculate the max temp

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

<https://docs.databricks.com/optimizations/semi-structured.html>

□ □□ □□□ □□□ □□□ □□ DDL□ □□□□□□.

1. temp_data □□□□ □□ □□ □□□□□.

2. as select ' {

3. "□□□":[

4. {"□□□": "2021□ 1□ 1□",

5. "temp": [25, 28, 45, 56, 39, 25]

6. },

7. {"□□□": "2021□ 2□ 1□",

8. "temp": [25, 28, 49, 54, 38, 25]

9. },


```
%sql
CREATE TABLE transactions(transactionId int, transactionDate timestamp,unitsSold int)
USING DELTA LOCATION "dbfs:/mnt/bronze/transactions"
```

OK

Command took 1.78 seconds -- by akhil at 10/10/2022, 5:02:40 PM on Akhil Vangala's Cluster

```
%sql
DESCRIBE EXTENDED transactions
```

OK

Command took 0.04 seconds -- by akhil at 10/10/2022, 5:02:40 PM on Akhil Vangala's Cluster

2

```
%sql
DESCRIBE EXTENDED transactions
```

OK

Command took 0.04 seconds -- by akhil at 10/10/2022, 5:02:40 PM on Akhil Vangala's Cluster

table Data Profile

	col_name	data_type	comment
10	Database	default	
11	Table	transactions	
12	Location	dbfs:/mnt/bronze/transactions	
13	Provider	delta	
14	Owner	root	
15	Type	EXTERNAL	
16	Table Properties	[delta.minReaderVersion=1,delta.minWriterVersion=2,external=true]	

NEW QUESTION: 200

6. Which of the following is a valid Delta Lake table property?

- A. `delta.minReaderVersion=1,delta.minWriterVersion=2`
- B. `delta.minReaderVersion=1,delta.minWriterVersion=2,external=true`
- C. `delta.minReaderVersion=1,delta.minWriterVersion=2,external=true`
- D. `delta.minReaderVersion=1,delta.minWriterVersion=2,external=true`
- E. `delta.minReaderVersion=1,delta.minWriterVersion=2,external=true`

Answer: C (LEAVE A REPLY)

□□

□□ □□□ □□□ □□□□.

□□□ □□□ 6□□ □□□□ □□□ □□□□ □□□□ □ □□□□ □□ □□ □□□□□ □□□□□. □□□ □□□ □□□□□, □□□, □□□□□□, □□□ □□□ □□□□ □□□□□.

Task name * ?

sales

Type *

Notebook

Source * ?

Workspace

Path * ?

/Repos/[REDACTED]/cli-demo/notebooks/job-demo

You are using a Local repository. To automatically track the upstream repository in staging or production Jobs, select Git in the Source field.

Cluster * ?

Process_all_departments_cluster (28.00 GB | 8 Cores | DBR 10.4 LTS | Spark 3.2.1 | S...

Parameters ?

UI | JSON

```
{
  "department": "accounting"
}
```

Depends on

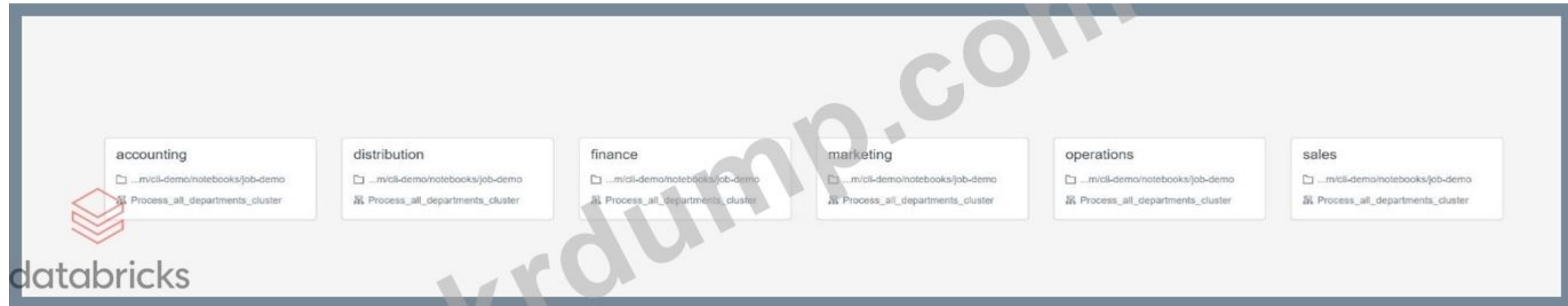
Select task dependencies...

▼ Advanced options

Cancel

Save task

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



NEW QUESTION: 201

□□ □□□□□ □□□□ □□ □□□□□□□ □□□□ □□□ □□□ □□□□□ □□□ □□□□□. □□ □ □□□ □□ □□□□□?

- A. □□□ □□□ DataFrame□ □□□ □ □□□□.
- B. DataFrame□ □□□□ SQL UDF□ □□□□□
- C. □□□ □□□ □□□□ □□□ □ □□ SQL UDF□ □□□□□.
- D. □□□ □□□ □□ □□□ □□□□ □□□ □□ □□□ □□□ □ □□□□.
- E. □□□ □□□ DataFrame□ □□□ □ □□□□.

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

□□

□□□ □□□ □□□ □□□□□□□ □□□ □ □□□ □□□□.

□□□ □□ □□□ □□ □□□□.

1.get_source_dataframe(tablename):

2. df = spark.read.table(tablename)

3. df□ □□□□□.

df = get_source_dataframe('test_table')

□□□ □□□ □□□□□ Spark□ □□□□□□□ □□□□ □□ □□□ □□ `df` □ □□□□□.

Databricks-Certified-Professional-Data-Engineer ☐☐ ☐☐☐ ☐☐☐☐☐ ☐☐ DumpTop ☐☐ ☐☐☐☐ ☐☐☐ Databricks-Certified-Professional-Data-Engineer ☐☐! DumpTop ☐ ☐☐ **Databricks-Certified-Professional-Data-Engineer** ☐☐ ☐☐☐ ☐☐☐☐☐☐, DumpTop Databricks-Certified-Professional-Data-Engineer ☐☐ ☐☐☐ ☐☐☐☐☐☐☐☐ ☐☐☐ ☐☐☐☐☐☐☐☐. ☐☐☐☐ ☐☐☐☐☐☐☐ ☐☐ DumpTop Databricks-Certified-Professional-Data-Engineer ☐☐☐ ☐☐☐☐☐. <https://www.dumptop.com/Databricks/Databricks-Certified-Professional-Data-Engineer-dump.html> (217 Q&As Dumps, **30%OFF** Special Discount: **KrDump**)