Free Questions for Databricks-Certified-Data-Analyst-Associate by vceexamstest

Shared by Quinn on 24-05-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

A data analyst creates a Databricks SQL Query where the result set has the following schema:

region STRING

number_of_customer INT

When the analyst clicks on the "Add visualization" button on the SQL Editor page, which of the following types of visualizations will be selected by default?

Options:

A- Violin Chart

B- Line Chart

C- IBar Chart

D- Histogram

E- There is no default. The user must choose a visualization type.

Answer:

C

Explanation:

According to the Databricks SQL documentation, when a data analyst clicks on the "Add visualization" button on the SQL Editor page, the default visualization type isBar Chart. This is because the result set has two columns: one of type STRING and one of type INT. The Bar Chart visualization automatically assigns the STRING column to the X-axis and the INT column to the Y-axis. The Bar Chart visualization is suitable for showing the distribution of a numeric variable across different categories.Reference:Visualization in Databricks SQL,Visualization types

Question 2

Question Type: MultipleChoice

Which of the following statements about a refresh schedule is incorrect?

Options:

A- A query can be refreshed anywhere from 1 minute lo 2 weeks

B- Refresh schedules can be configured in the Query Editor.

C- A query being refreshed on a schedule does not use a SQL Warehouse (formerly known as SQL Endpoint).

D- A refresh schedule is not the same as an alert.

E- You must have workspace administrator privileges to configure a refresh schedule

Answer:

C

Explanation:

Refresh schedules are used to rerun queries at specified intervals, and these queries typically require computational resources to execute. In the context of a cloud data service like Databricks, this would typically involve the use of a SQL Warehouse (or a SQL Endpoint, as they were formerly known) to provide the necessary computational resources. Therefore, the statement is incorrect because scheduled query refreshes would indeed use a SQL Warehouse/Endpoint to execute the query.

Question 3

Question Type: **MultipleChoice**

How can a data analyst determine if query results were pulled from the cache?

Options:

A- Go to the Query History tab and click on the text of the query. The slideout shows if the results came from the cache.

B- Go to the Alerts tab and check the Cache Status alert.

C- Go to the Queries tab and click on Cache Status. The status will be green if the results from the last run came from the cache.

D- Go to the SQL Warehouse (formerly SQL Endpoints) tab and click on Cache. The Cache file will show the contents of the cache.

E- Go to the Data tab and click Last Query. The details of the query will show if the results came from the cache.

Answer:

A

Explanation:

Databricks SQL uses a query cache to store the results of queries that have been executed previously. This improves the performance and efficiency of repeated queries. To determine if a query result was pulled from the cache, you can go to the Query History tab in the Databricks SQL UI and click on the text of the query. A slideout will appear on the right side of the screen, showing the query details, including the cache status. If the result came from the cache, the cache status will show "Cached". If the result did not come from the cache, the cache status will show "Not cached". You can also see the cache hit ratio, which is the percentage of queries that were served from the cache.Reference: The answer can be verified from Databricks SQL documentation which provides information on how to use the query cache and how to check the cache status. Reference link: Databricks SQL - Query Cache

Question 4

Question Type: **MultipleChoice**

Which of the following is a benefit of Databricks SQL using ANSI SQL as its standard SQL dialect?

Options:

A- It has increased customization capabilities

B- It is easy to migrate existing SQL queries to Databricks SQL

C- It allows for the use of Photon's computation optimizations

D- It is more performant than other SQL dialects

E- It is more compatible with Spark's interpreters

Answer:

B

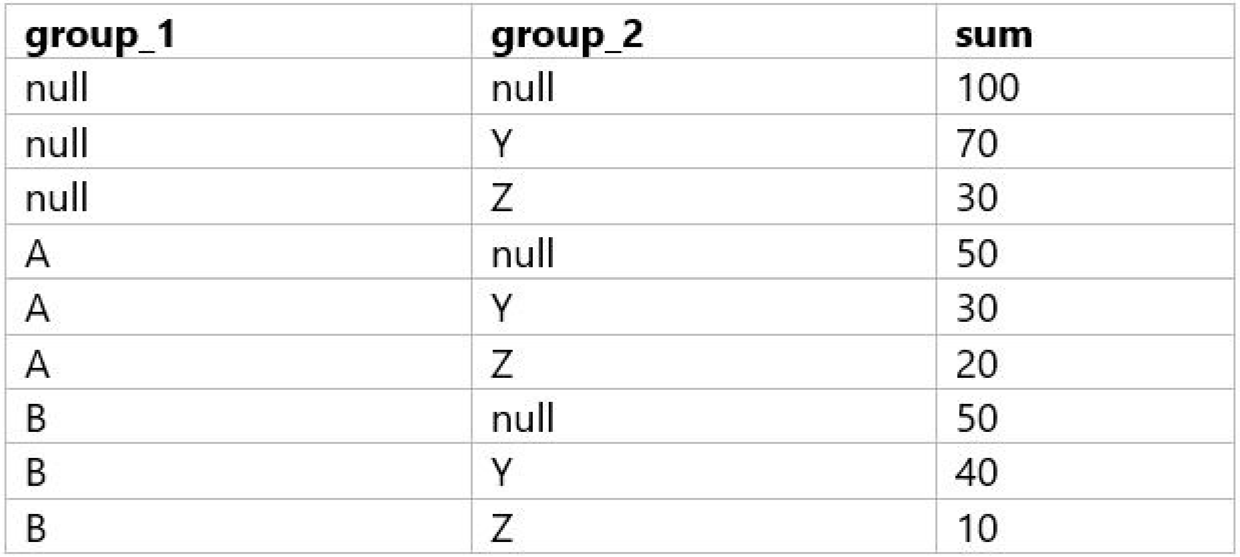
Explanation:

Databricks SQL uses ANSI SQL as its standard SQL dialect, which means it follows the SQL specifications defined by the American National Standards Institute (ANSI). This makes it easier to migrate existing SQL queries from other data warehouses or platforms that also use ANSI SQL or a similar dialect, such as PostgreSQL, Oracle, or Teradata.By using ANSI SQL, Databricks SQL avoids surprises in behavior or unfamiliar syntax that may arise from using a non-standard SQL dialect, such as Spark SQL or Hive SQL12.Moreover, Databricks SQL also adds compatibility features to support common SQL constructs that are widely used in other data warehouses, such as QUALIFY, FILTER, and user-defined functions2.Reference:ANSI compliance in Databricks Runtime,Evolution of the SQL language at Databricks: ANSI standard by default and easier migrations from data warehouses

Question 5

Question Type: MultipleChoice

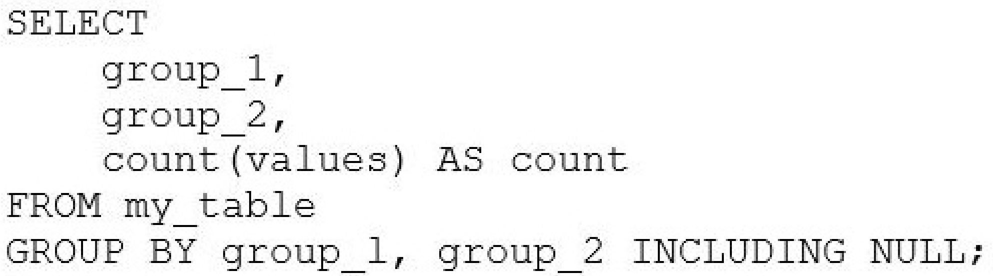
A data analyst is processing a complex aggregation on a table with zero null values and their query returns the following result:



group_1	group_2	sum
null	null	100
null	Y	70
null	Z	30
A	null	50
A	Y	30
A	Z	20
B	null	50
B	Y	40
B	Z	10

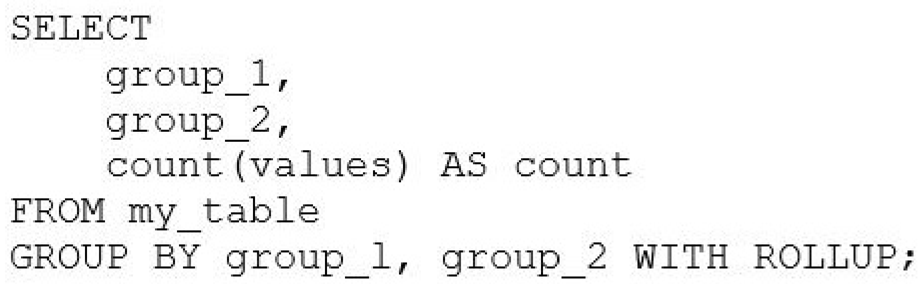
Which of the following queries did the analyst run to obtain the above result?

A)



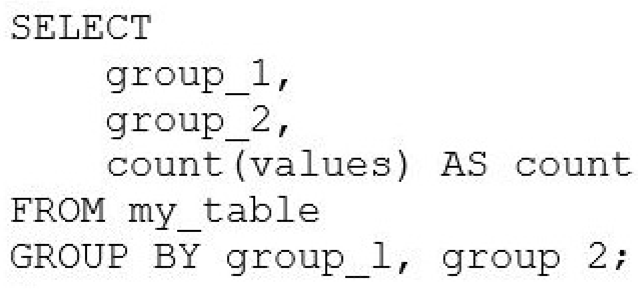
```
SELECT
    group_1,
    group_2,
    count(values) AS count
FROM my_table
GROUP BY group_1, group_2 INCLUDING NULL;
```

B)



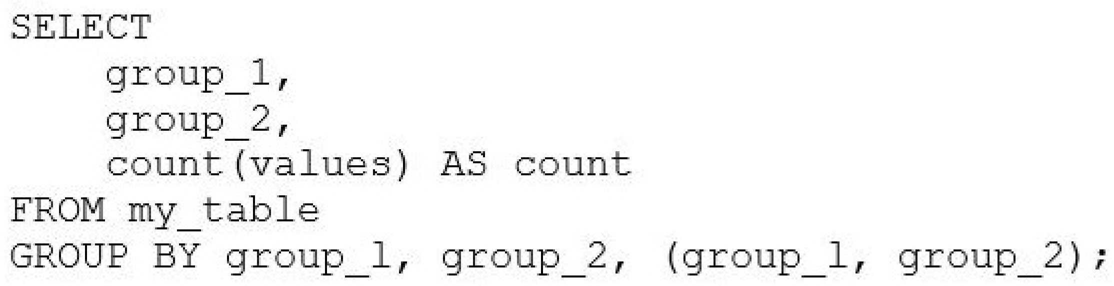
```
SELECT
    group_1,
    group_2,
    count(values) AS count
FROM my_table
GROUP BY group_l, group_2 WITH ROLLUP;
```

C)



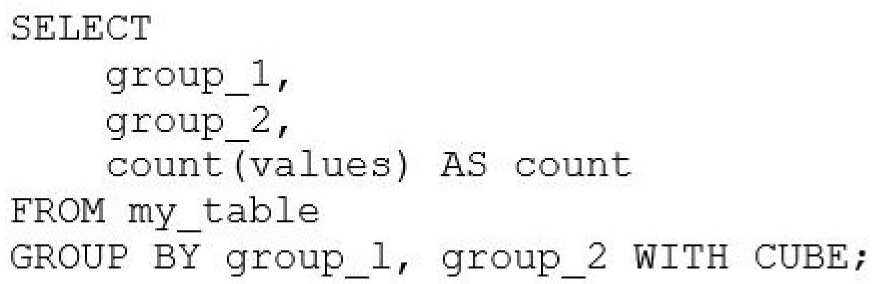
```
SELECT
    group_1,
    group_2,
    count(values) AS count
FROM my_table
GROUP BY group_l, group 2;
```

D)



```
SELECT
    group_1,
    group_2,
    count(values) AS count
FROM my_table
GROUP BY group_l, group_2, (group_l, group_2);
```

E)



```
SELECT
    group_1,
    group_2,
    count(values) AS count
FROM my_table
GROUP BY group_1, group_2 WITH CUBE;
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

E- Option E

Answer:

B

Explanation:

The result set provided shows a combination of grouping by two columns (group_1 and group_2) with subtotals for each level of grouping and a grand total. This pattern is typical of a GROUP BY ... WITH ROLLUP operation in SQL, which provides subtotal rows and a grand total row in the result set.

Considering the query options:

A) Option A: GROUP BY group_1, group_2 INCLUDING NULL - This is not a standard SQL clause and would not result in subtotals and a grand total.

B) Option B: GROUP BY group_1, group_2 WITH ROLLUP - This would create subtotals for each unique group_1, each combination of group_1 and group_2, and a grand total, which matches the result set provided.

C) Option C: GROUP BY group_1, group 2 - This is a simple GROUP BY and would not include subtotals or a grand total.

D) Option D: GROUP BY group_1, group_2, (group_1, group_2) - This syntax is not standard and would likely result in an error or be interpreted as a simple GROUP BY, not providing the subtotals and grand total.

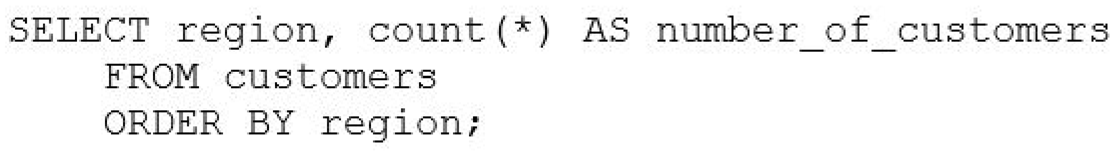
E) Option E: GROUP BY group_1, group_2 WITH CUBE - The WITH CUBE operation produces subtotals for all combinations of the selected columns and a grand total, which is more than what is shown in the result set.

The correct answer is Option B, which uses WITH ROLLUP to generate the subtotals for each level of grouping as well as a grand total. This matches the result set where we have subtotals for each group_1, each combination of group_1 and group_2, and the grand total where both group_1 and group_2 are NULL.

Question 6

Question Type: **MultipleChoice**

A data analyst has been asked to count the number of customers in each region and has written the following query:



```
SELECT region, count(*) AS number_of_customers
    FROM customers
    ORDER BY region;
```

If there is a mistake in the query, which of the following describes the mistake?

Options:

A- The query is using count('). which will count all the customers in the customers table, no matter the region.

B- The query is missing a GROUP BY region clause.

C- The query is using ORDER BY. which is not allowed in an aggregation.

D- There are no mistakes in the query.

E- The query is selecting region but region should only occur in the ORDER BY clause.

Answer:

B

Explanation:

In the provided SQL query, the data analyst is trying to count the number of customers in each region. However, they made a mistake by not including the "GROUP BY" clause to group the results by region. Without this clause, the query will not return counts for each distinct region but rather an error or incorrect result.Reference: The need for a GROUP BY clause in such queries can be understood from Databricks SQL documentation:Databricks SQL.

I also noticed that you uploaded an image with your question. The image shows a snippet of an SQL query written in plain text on a white background. The query is attempting to select regions and count customers from a "customers" table and order the results by region. There's no visible syntax highlighting or any other color - it's monochromatic. The query is the same as the one in your question. I'm not sure why you included the image, but maybe you wanted to show me the exact format of your query. If so, you can also use code blocks to display formatted content such as SQL queries. For example, you can write:

SELECT region, count(*) AS number_of_customers

FROM customers

ORDER BY region;

This way, you can avoid uploading images and make your questions more clear and concise. I hope this helps.

Question 7

Question Type: **MultipleChoice**

A data analyst has created a user-defined function using the following line of code:

```
CREATE FUNCTION price(spend DOUBLE, units DOUBLE)

RETURNS DOUBLE

RETURN spend / units;
```

Which of the following code blocks can be used to apply this function to the customer_spend and customer_units columns of the table customer_summary to create column customer_price?

Options:

A- SELECT PRICE customer_spend, customer_units AS customer_price FROM customer_summary

B- SELECT price FROM customer_summary

C- SELECT function(price(customer_spend, customer_units)) AS customer_price FROM customer_summary

D- SELECT double(price(customer_spend, customer_units)) AS customer_price FROM customer_summary

E- SELECT price(customer_spend, customer_units) AS customer_price FROM customer_summary

Answer:

E

Explanation:

A user-defined function (UDF) is a function defined by a user, allowing custom logic to be reused in the user environment1.To apply a UDF to a table, the syntax isSELECT udf_name(column_name) AS alias FROM table_name2. Therefore, option E is the correct way to use the UDFpriceto create a new columncustomer_pricebased on the existing columnscustomer_spendandcustomer_unitsfrom the tablecustomer_summary.Reference:

What are user-defined functions (UDFs)?

User-defined scalar functions - SQL

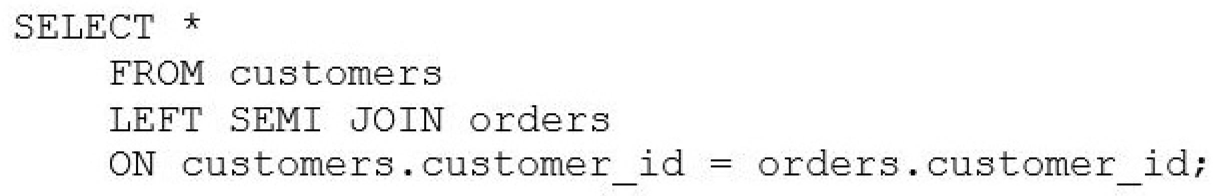
V

Question 8

Question Type: **MultipleChoice**

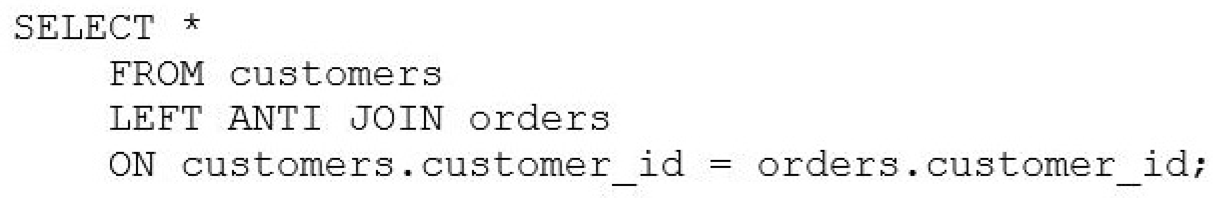
Consider the following two statements:

Statement 1:



```
SELECT *
    FROM customers
    LEFT SEMI JOIN orders
    ON customers.customer_id = orders.customer_id;
```

Statement 2:



```
SELECT *
    FROM customers
    LEFT ANTI JOIN orders
    ON customers.customer_id = orders.customer_id;
```

Which of the following describes how the result sets will differ for each statement when they are run in Databricks SQL?

Options:

A- The first statement will return all data from the customers table and matching data from the orders table. The second statement will return all data from the orders table and matching data from the customers table. Any missing data will be filled in with NULL.

B- When the first statement is run, only rows from the customers table that have at least one match with the orders table on customer_id will be returned. When the second statement is run, only those rows in the customers table that do not have at least one match with the orders table on customer_id will be returned.

C- There is no difference between the result sets for both statements.

D- Both statements will fail because Databricks SQL does not support those join types.

E- When the first statement is run, all rows from the customers table will be returned and only the customer_id from the orders table will be returned. When the second statement is run, only those rows in the customers table that do not have at least one match with the orders table on customer_id will be returned.

Answer:

B

Explanation:

Based on the images you sent, the two statements are SQL queries for different types of joins between the customers and orders tables. A join is a way of combining the rows from two table references based on some criteria. The join type determines how the rows are matched and what kind of result set is returned. The first statement is a query for a LEFT SEMI JOIN, which returns only the rows from the left table reference (customers) that have a match with the right table reference (orders) on the join condition (customer_id). The second statement is a query for a LEFT ANTI JOIN, which returns only the rows from the left table reference (customers) that have no match with the right table reference (orders) on the join condition (customer_id). Therefore, the result sets for the two statements will differ in the following way:

The first statement will return a subset of the customers table that contains only the customers who have placed at least one order. The number of rows returned will be less than or equal to the number of rows in the customers table, depending on how many customers have orders. The number of columns returned will be the same as the number of columns in the customers table, as the LEFT SEMI JOIN does not include any columns from the orders table.

The second statement will return a subset of the customers table that contains only the customers who have not placed any order. The number of rows returned will be less than or equal to the number of rows in the customers table, depending on how many customers have no orders. The number of columns returned will be the same as the number of columns in the customers table, as the LEFT ANTI JOIN does not include any columns from the orders table.

The other options are not correct because:

A) The first statement will not return all data from the customers table, as it will exclude the customers who have no orders. The second statement will not return all data from the orders table, as it will exclude the orders that have a matching customer. Neither statement will fill in any missing data with NULL, as they do not return any columns from the other table.

C) There is a difference between the result sets for both statements, as explained above. The LEFT SEMI JOIN and the LEFT ANTI JOIN are not equivalent operations and will produce different outputs.

D) Both statements will not fail, as Databricks SQL does support those join types. Databricks SQL supports various join types, including INNER, LEFT OUTER, RIGHT OUTER, FULL OUTER, LEFT SEMI, LEFT ANTI, and CROSS. You can also use NATURAL, USING, or LATERAL keywords to specify different join criteria.

E) The first statement will not return only the customer_id from the orders table, as it will return all columns from the customers table. The second statement is correct, but it is not the only difference between the result sets.

Question 9

Question Type: MultipleChoice

In which of the following situations should a data analyst use higher-order functions?

Options:

A- When custom logic needs to be applied to simple, unnested data

B- When custom logic needs to be converted to Python-native code

C- When custom logic needs to be applied at scale to array data objects

D- When built-in functions are taking too long to perform tasks

E- When built-in functions need to run through the Catalyst Optimizer

Answer:

C

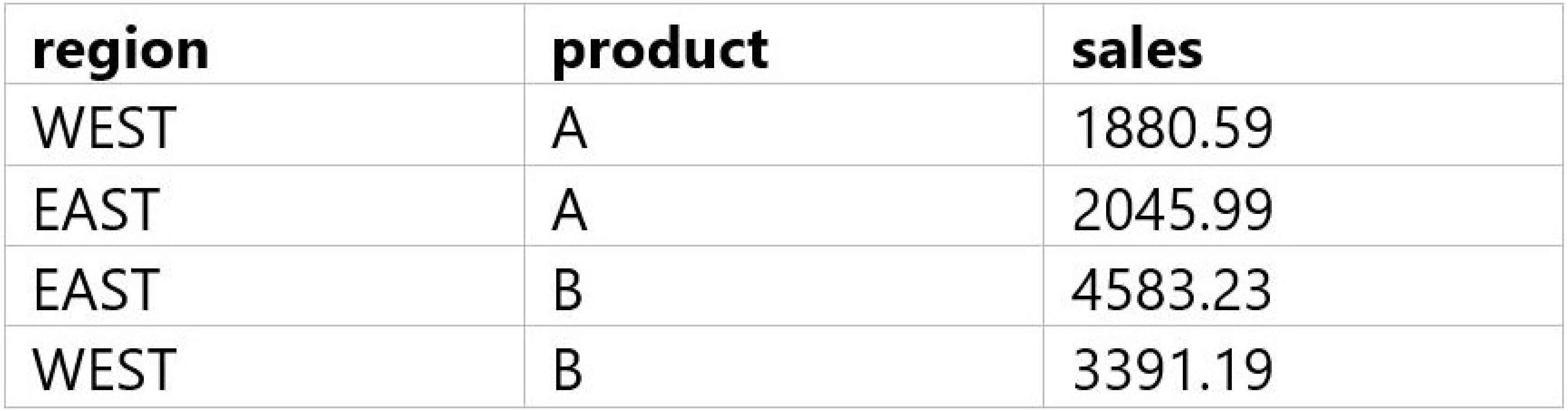
Explanation:

Higher-order functions are a simple extension to SQL to manipulate nested data such as arrays. A higher-order function takes an array, implements how the array is processed, and what the result of the computation will be. It delegates to a lambda function how to process each item in the array. This allows you to define functions that manipulate arrays in SQL, without having to unpack and repack them, use UDFs, or rely on limited built-in functions. Higher-order functions provide a performance benefit over user defined functions.Reference:Higher-order functions | Databricks on AWS,Working with Nested Data Using Higher Order Functions in SQL on Databricks | Databricks Blog,Higher-order functions - Azure Databricks | Microsoft Learn,Optimization recommendations on Databricks |

Question 10

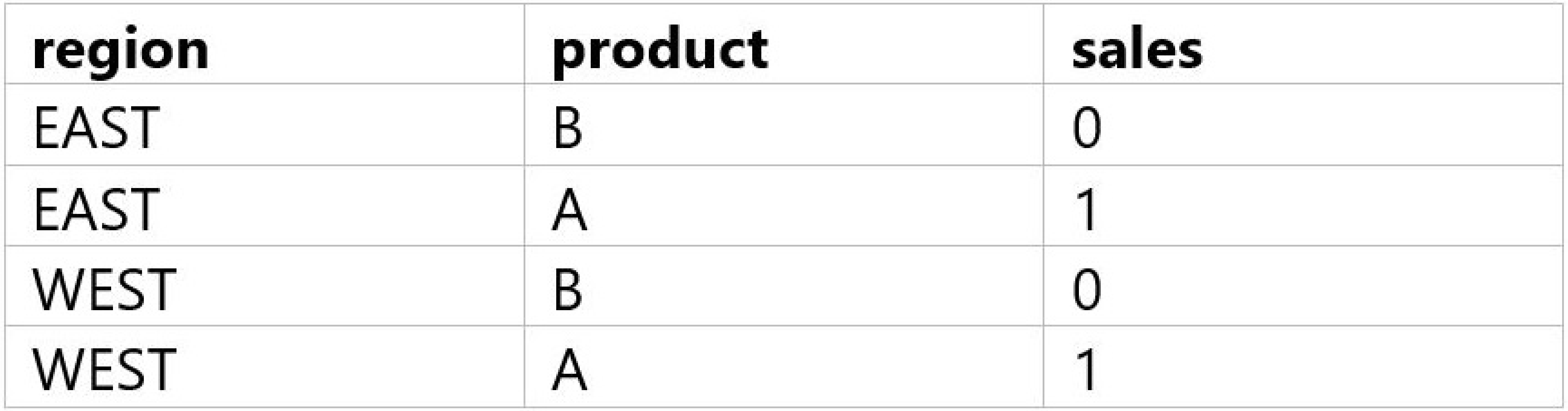
Question Type: MultipleChoice

A data analyst has been asked to use the below table sales_table to get the percentage rank of products within region by the sales:



region	product	sales
WEST	A	1880.59
EAST	A	2045.99
EAST	B	4583.23
WEST	B	3391.19

The result of the query should look like this:



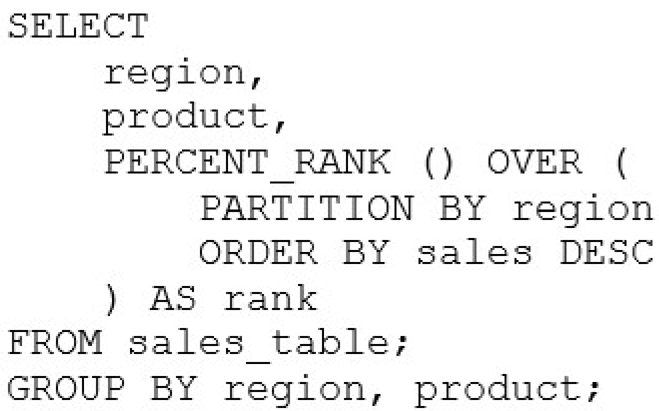
region	product	sales
EAST	B	0
EAST	A	1
WEST	B	0
WEST	A	1

Which of the following queries will accomplish this task?

A)

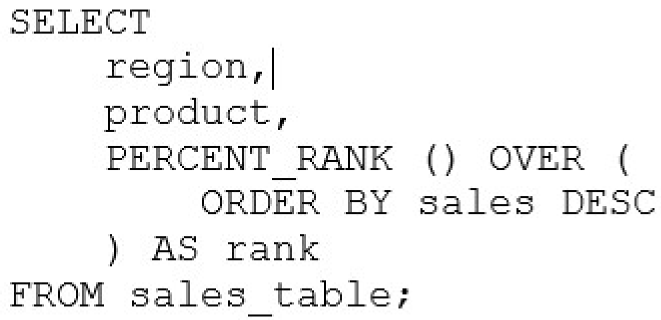
```
SELECT
    region,
    product,
    RANK() OVER (
        PARTITION BY region
        ORDER BY sales DESC
    ) AS rank
FROM sales_table;
GROUP BY region, product;
```

B)

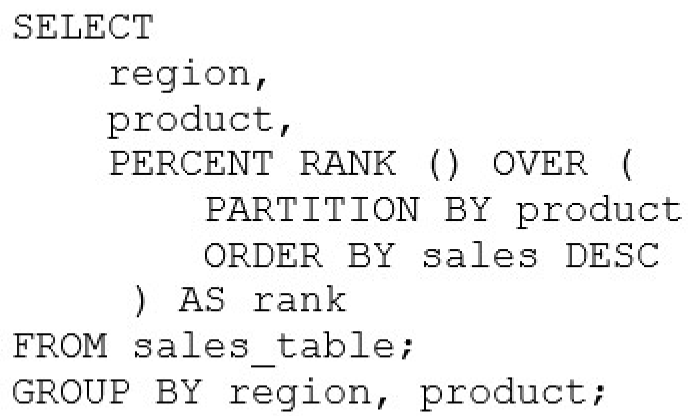


```
SELECT
    region,
    product,
    PERCENT_RANK () OVER (
        PARTITION BY region
        ORDER BY sales DESC
    ) AS rank
FROM sales_table;
GROUP BY region, product;
```

C)



```
SELECT
    region,
    product,
    PERCENT_RANK () OVER (
        ORDER BY sales DESC
    ) AS rank
FROM sales_table;
```



```
SELECT
    region,
    product,
    PERCENT RANK () OVER (
        PARTITION BY product
        ORDER BY sales DESC
     ) AS rank
FROM sales_table;
GROUP BY region, product;
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

B

Explanation:

The correct query to get the percentage rank of products within region by the sales is option B. This query uses the PERCENT_RANK() window function to calculate the relative rank of each product within each region based on the sales amount. The window function is partitioned by region and ordered by sales in descending order. The result is aliased as rank and displayed along with the region and product columns. The other options are incorrect because:

A) Option A uses the RANK() window function instead of the PERCENT_RANK() function. The RANK() function returns the rank of each row within the partition, but not the percentage rank. Also, the query does not have a GROUP BY clause, which is required for aggregate functions like SUM().

C) Option C uses the DENSE_RANK() window function instead of the PERCENT_RANK() function. The DENSE_RANK() function returns the rank of each row within the partition, but not the percentage rank. Also, the query does not have a GROUP BY clause, which is required for aggregate functions like SUM().

D) Option D uses the ROW_NUMBER() window function instead of the PERCENT_RANK() function. The ROW_NUMBER() function returns the sequential number of each row within the partition, but not the percentage rank. Also, the query does not have a GROUP BY clause, which is required for aggregate functions like SUM().Reference:

1: PERCENT_RANK (Transact-SQL)

2: Window functions in Databricks SQL

3: Databricks Certified Data Analyst Associate Exam Guide

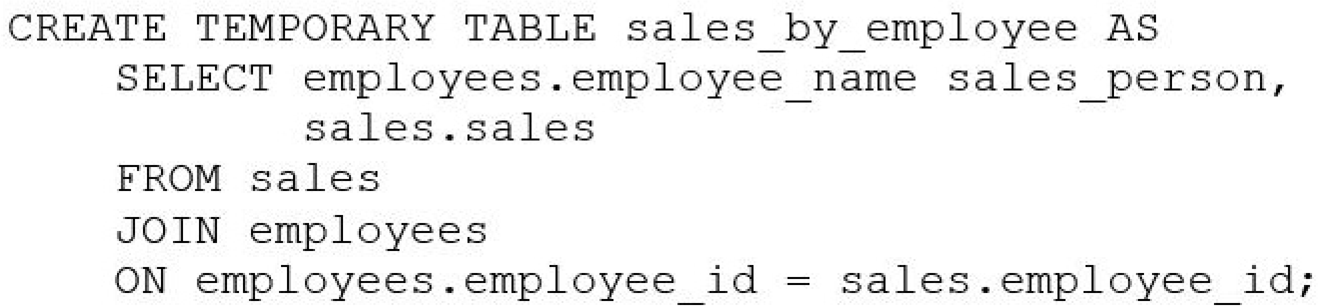
Question 11

Question Type: MultipleChoice

A business analyst has been asked to create a data entity/object called sales_by_employee. It should always stay up-to-date when new data are added to the sales table. The new entity should have the columns sales_person, which will be the name of the employee from the employees table, and sales, which will be all sales for that particular sales person. Both the sales table and the employees table have an employee_id column that is used to identify the sales person.

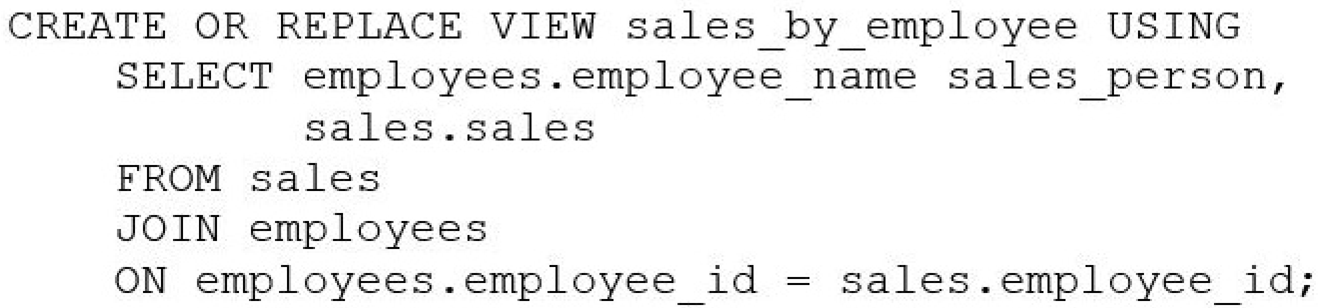
Which of the following code blocks will accomplish this task?

A)



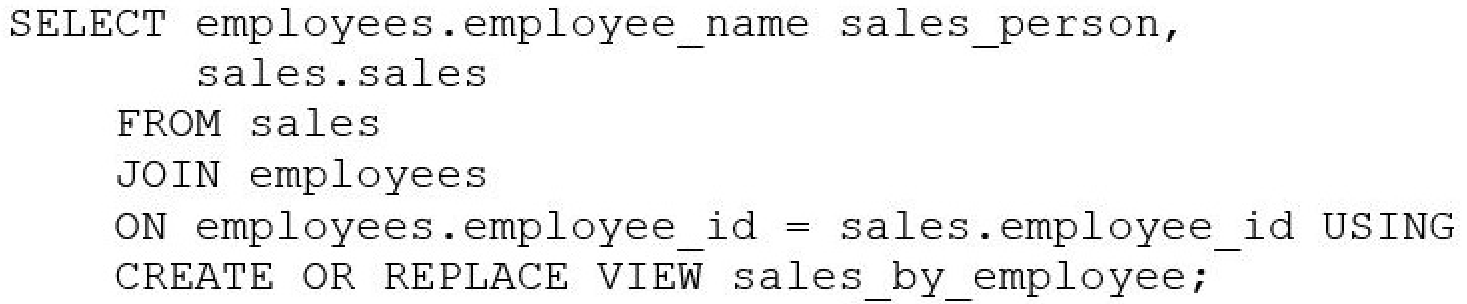
```
CREATE TEMPORARY TABLE sales_by_employee AS
    SELECT employees.employee_name sales_person,
           sales.sales
    FROM sales
    JOIN employees
    ON employees.employee_id = sales.employee_id;
```

B)



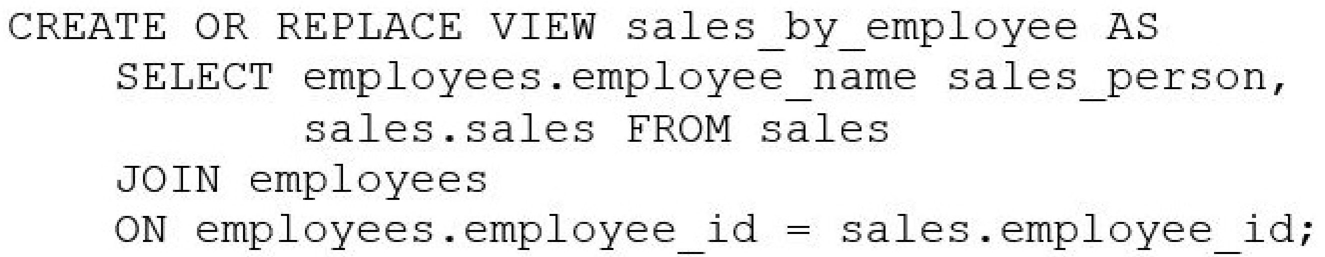
```
CREATE OR REPLACE VIEW sales_by_employee USING
    SELECT employees.employee_name sales_person,
           sales.sales
    FROM sales
    JOIN employees
    ON employees.employee_id = sales.employee_id;
```

C)



```
SELECT employees.employee_name sales_person,
       sales.sales
    FROM sales
    JOIN employees
    ON employees.employee_id = sales.employee_id USING
    CREATE OR REPLACE VIEW sales_by_employee;
```

D)



```
CREATE OR REPLACE VIEW sales_by_employee AS
    SELECT employees.employee_name sales_person,
           sales.sales FROM sales
    JOIN employees
    ON employees.employee_id = sales.employee_id;
```

Options:

A- Option

B- Option

C- Option

D- Option

Answer:

D

Explanation:

The SQL code provided in Option D is the correct way to create a view namedsales_by_employeethat will always stay up-to-date with the sales and employees tables. The code uses theCREATE OR REPLACE VIEWstatement to define a new view that joins the sales and employees tables on the employee_id column. It selects the employee_name as sales_person and all sales for each employee, ensuring that the data entity/object is always up-to-date when new data are added to these tables.

Question 12

Question Type: **MultipleChoice**

A data analysis team is working with the table_bronze SQL table as a source for one of its most complex projects. A stakeholder of the project notices that some of the downstream data is duplicative. The analysis team identifies table_bronze as the source of the duplication.

Which of the following queries can be used to deduplicate the data from table_bronze and write it to a new table table_silver?

A)

```
CREATE TABLE table_silver AS

SELECT DISTINCT *

FROM table_bronze;
```

B)

```
CREATE TABLE table_silver AS

INSERT *

FROM table_bronze;
```

C)

```
CREATE TABLE table_silver AS

MERGE DEDUPLICATE *

FROM table_bronze;
```

D)

INSERT INTO TABLE table_silver

SELECT * FROM table_bronze;

E)

INSERT OVERWRITE TABLE table_silver

SELECT * FROM table_bronze;

Options:

A- Option A

B- Option B

C- Option C

D- Option D

E- Option E

Answer:

A

Explanation:

Option A uses theSELECT DISTINCTstatement to remove duplicate rows from thetable_bronzeand create a new tabletable_silverwith the deduplicated data.This is the correct way to deduplicate data using Spark SQL12. Option B simply inserts all the rows fromtable_bronzeintotable_silver, without removing any duplicates. Option C is not a valid syntax for Spark SQL, as there is noMERGE DEDUPLICATEstatement. Option D appends all the rows fromtable_bronzeintotable_silver, without removing any duplicates. Option E overwrites the existing data intable_silverwith the data fromtable_bronze, without removing any duplicates.Reference:Delete Duplicate using SPARK SQL,Spark SQL - How to Remove Duplicate Rows

20%
DISCOUNT