CERTSWARRIOR

Cloudera

CDP-4001

CDP Data Analyst- Certification Exam

Questions&AnswersPDF

ForMoreInformation:

https://www.certswarrior.com/

Features:

- 90DaysFreeUpdates
- 30DaysMoneyBackGuarantee
- InstantDownloadOncePurchased
- 24/7OnlineChat Support
- ItsLatestVersion

Latest Version: 6.0

Question: 1

You are analyzing customer purchase data in a Hive table called 'customer_purchaseS. You want to create a bar chart visualization showing the total revenue generated by each product category. Which of the following HiveQL queries would you use to generate the data for this visualization?

A.
```
SELECT product_category, SUM(purchase_amount) AS total_revenue FROM customer_purchases GROUP BY product_category;
```
B.
```
SELECT product_category, COUNT(*) AS total_purchases FROM customer_purchases GROUP BY product_category;
```
C.
```
SELECT product_category, AVG(purchase_amount) AS average_revenue FROM customer_purchases GROUP BY product_category;
```
D.
```
SELECT product_category, MAX(purchase_amount) AS highest_purchase FROM customer_purchases GROUP BY product_category;
```
E.
```
SELECT product_category, MIN(purchase_amount) AS lowest_purchase FROM customer_purchases GROUP BY product_category;
```

Answer: A

Explanation:
The correct query is Option A. It uses the 'SUM()" aggregate function to calculate the total revenue for each product category, grouping the results by 'product_category'. This query provides the data needed for a bar chart visualization showing revenue per category.

Question: 2

You have a Hive table named •web_logs' containing website visit data. You want to create a line chart visualization showing the number of unique visitors to your website each day. Which HiveQL query would you use to extract the data for this visualization?

○
```
SELECT DATE(visit_timestamp) AS visit_date, COUNT(DISTINCT user_id) AS unique_visitors FROM web_logs GROUP BY visit_date;
```
○
```
SELECT DATE(visit_timestamp) AS visit_date, COUNT(*) AS total_visits FROM web_logs GROUP BY visit_date;
```
○
```
SELECT DATE(visit_timestamp) AS visit_date, SUM(visit_duration) AS total_visit_duration FROM web_logs GROUP BY visit_date;
```
○
```
SELECT DATE(visit_timestamp) AS visit_date, AVG(visit_duration) AS average_visit_duration FROM web_logs GROUP BY visit_date;
```
○
```
SELECT DATE(visit_timestamp) AS visit_date, COUNT(page_views) AS total_page_views FROM web_logs GROUP BY visit_date;
```

A. Option A
B. Option B

C. Option C
D. Option D
E. Option E

Answer: A

Explanation:
The correct query is Option A. It uses the 'DATE()' function to extract the date from the 'visit_timestamp' column and the 'COUNT(DISTINCT user_id)' function to count the number of unique user IDs for each date. This query provides the data needed for a line chart visualization showing daily unique visitors.

Question: 3

You need to analyze website traffic data from a Hive table called 'website_traffic'. The table has columns 'visit_date' page_urr, 'user_id', and 'visit_duration'. You want to create a heatmap visualization showing the most popular pages on your website for each day of the week. Which of the following HiveQL queries would you use to prepare the data for this visualization?

A.
```
SELECT DAYOFWEEK(visit date) AS dav of week, page_url, COUNT(*) AS page_views FROM website_traffic GROUP BY day_of_week, page_url ORDER BY day_of_week, page_views DESC;
```
B.
```
SELECT DAYOFWEEK(visit date) AS day of week, page url, COUNT(DISTINCT user_id) AS unique_visitors FROM website_traffic GROUP BY day_of_week, page_url ORDER BY day_of_week, unique_visitors DESC;
```
C.
```
SELECT DAYOFWEEK(visit_date) AS day_of_week, page_url, SUM(visit_duration) AS total_visit_duration FROM website_traffic GROUP day_of_week, page_url ORDER BY day_of_week, total_visit_duration DESC;
```
D.
```
SELECT DAYOFWEEK(visit_date) AS day_of_week, page_url, AVG(visit_duration) AS average_visit_duration FROM website_traffic GROUP BY BY day_of_week, page_url ORDER BY day_of_week, average_visit_duration DESC;
```
E.
```
SELECT DAYOFWEEK(visit_date) AS day_of_week, page_url, COUNT(DISTINCT user_id) AS unique_visitors FROM website_traffic GROUP BY day_of_week, page_url ORDER BY day_of_week, unique_visitors ASC;
```

Answer: A,B,C

Explanation:
The correct queries are Options A, B, and C. These queries all extract the day of the week (using 'DAYOFWEEK()") and page URL, then group the results and aggregate the data. Option A uses to get the total number of page views, Option B uses 'COUNT(DISTINCT user_id)' to get the number of unique visitors, and Option C uses 'SUM(visit_duration)' to get the total visit duration. All three options provide suitable data for a heatmap visualization, depending on the specific metric you want to visualize.

Question: 4

You are building a dashboard in Cloudera Data Visualizations (CDV) to analyze website traffic data. You want to display a line chart showing the number of unique visitors per hour over the last 24 hours. Which CDV widget would you use to achieve this?

A. Time Series
B. Bar Chart
C. Pie Chart
D. Scatter Plot
E. Table

Answer: A

Explanation:
The Time Series widget is specifically designed for displaying data over time, making it the ideal choice for visualizing unique visitors per houn The other options are not suitable for this scenario. Bar charts are best for comparing categorical data, pie charts represent proportions of a whole, scatter plots show relationships between two variables, and tables display raw data in a tabular format.

Question: 5

You want to create a CDV dashboard that dynamically filters data based on user selections. You have a dataset with sales data by region and product category. You want to allow users to choose a specific region and see the corresponding sales data for all product categories. Which CDV feature would you use to enable this dynamic filtering?

A. Data Filters
B. Drill-Down
C. Data Linking
D. Cross-Filtering
E. Data Blending

Answer: D

Explanation:
Cross-filtering in CDV allows you to apply filters from one widget to other widgets on the dashboard. In this scenario, you would use a filter widget to let users select a region, and then the selected region would automatically filter the sales data in the other widgets displaying sales by product category. Data Filters are for selecting specific values, Drill-Down is for exploring data in more detail, Data Linking connects different data sources, and Data Blending combines data from multiple sources.

Question: 6

You're building a CDV dashboard to monitor website performance metrics. You want to display a gauge widget that shows the average website load time in seconds, with a green color if the average is below 2 seconds, yellow for 2-4 seconds, and red for above 4 seconds. Which CDV feature would you use to create this dynamic color-based thresholding?

A. Conditional Formatting
B. Data Linking
C. Drill-Down

D. Data Blending
E. Data Filters

Answer: A

Explanation:
Conditional Formatting in CDV allows you to apply different formatting rules based on specific conditions. In this scenario, you would use conditional formatting on the gauge widget to change the color based on the average website load time. The other options are not relevant to this task. Data Linking connects different data sources, Drill-Down is for exploring data in more detail, Data Blending combines data from multiple sources, and Data Filters are used for selecting specific values.

Question: 7

You need to retrieve data from a table named 'customer_orders' within the database 'retail_data' in Impala. Which of the following Impala commands will successfully display the table schema?

A. DESCRIBE retail data.customer orders;
B. SHOW TABLES IN retail data LIKE 'customer orders';
C. SELECT* FROM retail data.customer orders LIMIT 1;
D. SHOW CREATE TABLE retail data.customer orders;
E. SELECT* FROM customer orders;

Answer: A,D

Explanation:
Both options 'A' and 'D' are correct. Option 'A' uses the 'DESCRIBE command, which provides a concise overview of the table schema, including column names and data types. Option 'D', using 'SHOW CREATE TABLE' , presents the complete DDL statement used to create the table, offering a more detailed view of the table structure and its properties. Option 'B' lists tables in the specified database matching the given pattern but doesn't show the schema. Option 'C' retrieves the first row of data from the table, not the schema, and Option 'E' assumes the database is already in context.

Question: 8

You are tasked with identifying all tables in an Impala database named 'sales_analytics' that contain data related to customer demographics. Which Impala command would most efficiently achieve this goal?

- ○ SHOW TABLES IN sales_analytics LIKE 'customer%';
- ○ DESCRIBE sales_analytics.*;
- ○ SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA='sales_analytics' AND TABLE_NAME LIKE 'customer%';
- ○ SHOW TABLES IN sales_analytics;
- ○ SELECT * FROM sales_analytics.customer_data;

A. Option A
B. Option B

C. Option C
D. Option D
E. Option E

Answer: C

Explanation:
Option 'C' is the most accurate and efficient approach. It utilizes the 'INFORMATION SCHEMA' database, which provides metadata about database objects. The query filters tables in the 'sales_analytics' database by matching the table name pattern 'customer%' to effectively identify tables related to customer demographics. Options 'A' and 'D' list all tables in the database, not specifically those related to customer demographics. Option 'B' attempts to describe all tables in the database, which would be inefficient. Option 'E' retrieves data from a specific table, not a general list of tables.

Question: 9

You need to pull data from tables named 'product_sales' and 'customer_activity' within the Impala database 'ecommerce'. Which of the following Impala commands will retrieve data from both tables, combining it into a single result set, based on a shared column named 'customer id'?

- ☐ SELECT * FROM ecommerce.product_sales UNION ALL ecommerce.customer_activity;
- ☐ SELECT * FROM ecommerce.product_sales JOIN ecommerce.customer_activity ON product_sales.customer_id = customer_activity.customer_id;
- ☐ SELECT * FROM ecommerce.product_sales INTERSECT ecommerce.customer_activity;
- ☐ SELECT * FROM ecommerce.product_sales EXCEPT ecommerce.customer_activity;
- ☐ SELECT * FROM ecommerce.product_sales WHERE customer_id IN (SELECT customer_id FROM ecommerce.customer_activity);

A. Option A
B. Option B
C. Option C
D. Option D
E. Option E

Answer: B,E

Explanation:
Both options 'B' and 'E' achieve the goal of combining data from two tables based on a shared column. Option 'B' uses a 'JOIN' clause, specifically an 'INNER JOIN', to retrieve matching rows from both tables based on the 'customer_id' column. Option 'E' utilizes a subquery within the 'WHERE' clause to filter rows in 'product_sales' to those whose 'customer_id' values exist in the 'customer_activity' table. Option 'A' uses 'UNION ALL' , which concatenates the results of both queries, not joining them based on a common column. Option 'C' retrieves only the rows that are common to both tables, not the complete combined dataset. Option 'D' finds rows present in the first table but not in the second, not the desired combination of data.

Question: 10

You have a table 'customer data' in Hive with a column 'birth date' stored as a string in the format 'YYYY-MM-DD'. You need to convert this column to a date type and calculate the age of each customer. Which HiveQL query would achieve this?

```
SELECT *, date_format(birth_date, 'yyyy-MM-dd') AS birth_date, datediff(current_date, birth_date) AS age FROM customer_data;
```

```
SELECT *, cast(birth_date as date) AS birth_date, year(current_date) - year(birth_date) AS age FROM customer_data;
```

```
SELECT *, to_date(birth_date, 'yyyy-MM-dd') AS birth_date, years_between(current_date, birth_date) AS age FROM customer_data;
```

```
SELECT *, convert(birth_date, date) AS birth_date, age(birth_date, current_date) AS age FROM customer_data;
```

```
SELECT *, cast(birth_date as timestamp) AS birth_date, datediff(current_date, birth_date) / 365.25 AS age FROM customer_data;
```

A. Option A
B. Option B
C. Option C
D. Option D
E. Option E

Answer: B

Explanation:
Option B is the correct answer. It uses the 'cast' function to convert the string 'birth_date' to a date data type. Then, it uses the 'year' function to calculate the age by subtracting the year of birth from the current year. The other options are incorrect because they either use incorrect functions or do not properly handle the date conversion and age calculation.

CERTSWARRIOR

FULL PRODUCT INCLUDES:

Money Back Guarantee

Instant Download after Purchase

90 Days Free Updates

PDF Format Digital Download

24/7 Live Chat Support

Latest Syllabus Updates

For More Information – Visit link below:

https://www.certswarrior.com

16 USD Discount Coupon Code: U89DY2AQ