LOOKML-DEVELOPER Q&As

LookML Developer

Pass Google LOOKML-DEVELOPER Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.geekcert.com/lookml-developer.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Google Official Exam Center

QUESTION 1

The code below shows a view order_items with its measures total_revenue and user_count

```
view: order_items {

measure: total_revenue {

type: sum

sql: ${TABLE}.sale_price ;;

}

measure: user_count {

type: count_distinct

sql: ${users.id} ;;

}
}
```

Which code correctly represents a new measure that calculates average revenue per user?

A.
```
measure: average_revenue_per_user {
type: number
sql: ${total_revenue}/${user_count} ;;
}
```

B.
```
measure: average_revenue_per_user {
type: average
sql: ${total_revenue}/${user_count} ;;
}
```

C.
```
measure: average_revenue_per_user {
type: number
sql: ${total_revenue}/${users.id};;
}
```

D.
```
measure: average_revenue_per_user {
type: average sql: ${total_revenue}/${users.id};;
}
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: C

QUESTION 2

A developer needs to build a new dimension that offers an age-based cohort representation of users. Which LookML code should the developer use to meet the requirement?

A.
```
dimension: age_field {
type: bins
bins_size: 30
style: classic
sql: ${age} ;;
}
```

B.
```
dimension: age_field {
type: groups
groups: [<30, 30-60, >60]
sql: ${age} ;;
}
```

C.
```
dimension: age_field {
type: string tiers: [0 to 30, 30 to 60, 60 and above]
style: classic
sql: ${age} ;;
}
```

D.
```
dimension: age_field {
type: tier tiers: [0, 30, 60]
style: classic A sql: ${age} ;;
}
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: B

QUESTION 3

A LookML developer has created a model with many Explores in it. Business users are having a difficult time locating the Explore they want in the long list displayed.

Which two actions can the LookML developer take to improve the user interface? (Choose two.)

A. Apply the hidden parameter with a value of yes to Explores that only exist to power specific Looks, dashboards, or suggestion menus.

B. Modify the business users' roles so they do not have this model in their model set.

C. Combine the Explores into just a few Explores that each join to many views.

D. Apply the group_label parameter to organize the Explores under different headings.

E. Apply the fields parameter so that each Explore has fewer fields in it.

Correct Answer: BC

QUESTION 4

A developer has a persistent derived table view called user_facts that contains aggregated data for each user. The developer needs to query the data from this table in another derived table view called user_region_facts.

Which strategy should the developer use to write the query for user_region_facts that will leverage the existing derived table?

A. Use ${user_facts.SQL_TABLE_NAME} to reference the user_facts derived table.

B. Copy the name of the database table in the scratch schema for the user_facts derived table.

C. Writhe the query form user_facts into a common table expression (WITH user_facts AS...).

D. Write a subquery in the FROM clause and alias with ${user_facts}.

Correct Answer: C

QUESTION 5

A developer needs to add an Explore built off of the orders view, which surfaces only completed orders. An orders Explore exists that contains all order information. Fields from the orders view are also referenced in other existing views such as ${orders.fieldname}.

How should developer define a new Explore for completed orders and keep all field references working correctly?

A.
```
explore: completed_orders {
sql_always_where: ${orders.status} = "complete" ;;
view_name: orders
}
```

B.
```
explore: completed_orders {
sql_always_where: ${orders.status} = "complete" ;;
from: orders
}
```

C.
```
explore: completed_orders {
always_filter: {
A field: orders.status
A value: "complete"
}
from: orders
view_name: orders
}
```

D.
```
explore: completed_orders {
always_filter: {
A field: orders.status
A value: "complete"
}
from: completed_orders
view_name: orders
}
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: C