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.passapply.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

Users report that every time they change the filter on their Explore, the filters take a very long time to populate.

How can the developer improve the filtering experience with this Explore?

A. Limit the filter suggestions using the suggestions parameter.

B. Add an always_filter parameter to restrict the filter suggestions.

C. Use an access_filter parameter to automatically apply filters.

D. Add persistence to the base view of the Explore.

Correct Answer: A

QUESTION 3

A developer wants to create a new Explore based on the order_items view. The developer creates an

Explore in the ecommerce model file with the following definition:

explore: order_items {}

After saving and validations, the developer receives this LookML validator error:

Inaccessible view "inventory_items", "inventory_items" is not accessible in explore" "order_items". Check

for typos and missing joins in explore "order_items".

What caused this error to appear?

A. A field in the order_items view references a field in the inventory_items view.

B. A field in the inventory_items view references a field in the order_items view.

C. There is an Explore named inventory_items which references the order_items view.

D. There is another Explore named order_items which references the inventory_items view.

Correct Answer: A

QUESTION 4

A LookML developer has a transactions view with several measures that each perform complex calculations involving multiple fields. The LookML developer creates an Explore based on the transactions view. The product team wants to perform further functions on these measures, such as SUM, AVG, MIN, MAX, and RANK. The team wants these further functions to be performed at different levels of detail: weekly, monthly, and yearly.

How can the LookML developer model these requirements and minimize the amount of code rewriting?

A. Add measures to the transactions view of type: number to apply the required functions.

B. Change the existing measures in the transactions view to dimensions, and add measures of the different required types.

C. Create a constant for each measure so it can be reused across other areas of the LookML project.

D. Create native derived tables using transactions as the explore_source.

Correct Answer: D

QUESTION 5

The developer is creating an Explore that includes the product users, and orders views that will meet the following guidelines.

Joins between the orders and users views should not incur high performance costs.

Users of this Explore will primarily be looking at data from the orders view.

Users of this Explore should only be able to see orders from the retailer "Fashion.ly".

The only field the users need from the products view is product.name.

Which LookML should the developer use?

A.
```
explore: orders {

    join: product {

    fields: [product.name]

    join: users {...}

    sql_always_where: ${orders.retailer} = 'Fashion.ly' ;;

    }
```

B.
```
explore: orders {

    fields: [product.name]

    join: product {...}

    join: users {...}

    sql_always_where: ${orders.retailer} = 'Fashion.ly' ;;

    }
```

C.
```
explore: users {

join: product {

fields: [product.name]

}

join: orders {...}

always_filter: {

filters: {

fields: orders.retailer

value: "Fashion.ly"

}}}
```

D.
```
explore: users {

join: product {

fields: [product.name]

}

join: orders {...}

sql_always_where: ${orders.retailer} = 'Fashion.ly' ;;

}
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: D