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.pass4itsure.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

A developer needs to model out LookML to convert existing reports into Looker. The existing reports are:

Report 1: A report with order and order_items data, which finds the order with the largest total value of the order_item prices. Report 2: A report with order and order_items data, which finds the order with the largest total number of

products ordered.

Report 3: A report with data on every product, whether or not it has been ordered.

Each database table used is updated in real time as orders are made.

How should the developer construct an Explore using the order_items view as the base view?

A. Create one persistent derived table to calculate Report 1, create one persistent derived table to calculate Report 2, and join in the products view with a full_outer join.

B. Create one persistent derived table to calculate Reports 1 and 2, and join in the products view with a full_outer join.

C. Create one ephemeral derived table to calculate Report 1, create one ephemeral derived table to calculate Report 2, and join in the products view with a left_outer join.

D. Create one ephemeral derived table to calculate Reports 1 and 2, and join in the products view with a full_outer join.

Correct Answer: A

QUESTION 2

Users must be able to click on the Country field in their Explore and be redirected to another Explore that shows all countries compared.

Which parameter should be added to the country dimension to create a connection to this other associated Explore?

A. url_encode

B. drill_fields

C. tags

D. link

Correct Answer: D

QUESTION 3

A developer is building an e-commerce Explore with the following datasets: orders and users. The business user needs to be able to answer questions about sellers and buyers within the same Explore. Each order in the orders table reports a buyer and seller ID. The users table has the detailed information about the individual buyer and seller.

How should the Explore be defined to meet this requirement?

○ A.
```
explore: orders

join: buyers {

view_name: users

sql_on: ${orders.buyer_id} = ${buyers.id} ;;

relationship: many_to_one

}

join: sellers {

view_name: users

sql_on: ${orders.seller_id} = ${sellers.id} ;;

relationship: many_to_one

}
```

○ B.
```
explore: orders

join: users {

sql_on: ${orders.buyer_id} = ${users.id} AND ${orders.seller_id} = ${users.id} ;;

A relationship: many_to_one

}
```

C.
```
explore: orders

join: buyers {

from: users

sql_on: ${orders.buyer_id} = ${buyers.id} ;;

relationship: many_to_one

}

join: sellers {

from: users

sql_on: ${orders.seller_id} = ${sellers.id} ;;

relationship: many_to_one

}
```

D.
```
explore: orders

join: users {

sql_on: ${orders.buyer_id} = ${users.id} OR ${orders.seller_id} = ${users.id} ;;

relationship: many_to_one

}
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: B

QUESTION 4

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

QUESTION 5

A LookML developer has written the following persistent derived table. It references orders_rollup, another persistent derived table that also rebuilds with the same SQL trigger value.

```
view: user_facts {

derived_table: {

sql_trigger_value: SELECT "current date function";;

sql: SELECT col1, col2, col3

FROM ${orders_rollup.SQL_TABLE_NAME} ;;

}
}
```

Which change is needed to guarantee that user_facts will always rebuild with the latest data from orders_rollup?

A. Change the sql_trigger_value parameter of user_facts to select the current date plus one hour, so it triggers an hour after orders_rollup.

B. Change the orders_rollup view reference to ${orders_rollup.DERVIED_TABLE_NAME}

C. Change the sql_trigger_value parameter for both persistent derived tables to a datagroup_trigger parameter, and set them to use the same datagroup.

D. Change the orders_rollup view reference to the literal table name from the database\\'s scratch schema.

Correct Answer: C