



# LOOKML-DEVELOPER<sup>Q&As</sup>

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

-  **Instant Download** After Purchase
-  **100% Money Back** Guarantee
-  **365 Days** Free Update
-  **800,000+** Satisfied Customers





### QUESTION 1

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

---

## QUESTION 2

A developer is defining the users table within a view file in Looker. The users table will be available as an individual Explore and it will also be joined into other Explores, such as the products Explore. The developer needs to limit the fields visible in the products Explore without affecting the visibility of the fields in the users Explore.

How should the developer meet this requirement?

A. Use the fields parameter at the join level for the products Explore to specify which fields should be included and leave the users Explore as is.

B. Create duplicate dimensions and measures, one for the users Explore and one for the products Explore, and use the hidden parameter to modify the visibility of the fields.

C. Create two view files for the users table. One view file will have all possible fields for the users Explore, and the other will have only the fields required for the products Explore.

D. Use the hidden parameter in the users view file for the fields that should not come over to the products Explore and leave the users Explore as is.

Correct Answer: A

---

## QUESTION 3

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 4

A developer commits changes after adding LookML for a new measure. Upon pulling from production, the developer notices the following lines in the LookML:

```
<<<<<<< HEAD
measure: metric_b {
  type: average
  sql: ${item.price} ;;
}
=====
dimension: metric_a {
  type: number
  sql: ${item.price} ;;
}
>>>>>> branch 'master'
```

A. Remove “andlt;> branch ‘master’”

B. Remove “andlt;”