

# 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.leads4pass.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

Users have built a popular dashboard and want to have change management built in for any edits made to the dashboard. The developer sets up version control for the model on which the dashboard is based.

What should the developer build to meet the business requirement?

- A. A native derived table based on the dashboard.
- B. A dashboard LookML file included in the project.
- C. A link to the dashboard included in the project.
- D. An Explore LookML file based on the dashboard.

Correct Answer: B

---

## QUESTION 2

The `daily_forecast` Explore used by the sales team needs to be cached for 24 hours. All other Explores used by the sales team need to be cached for one hour.

What is a scalable way to configure this caching logic?

- A. Define two datagroups for the model. Apply `persist_with` at the model level with the datagroup for 1 hour caching, and apply `persist_with` to `daily_forecast` with the datagroup for 24-hour caching.
- B. Define `max_cache_age` on `daily_forecast` Explores of 24 hours. Define `max_cache_age` on all other Explores for one hour.
- C. Define two datagroups for the model. Create a persistent derived table (PDT) for the `daily_forecast` Explore, and apply `datagroup_trigger` to it using the datagroup for 24-hour caching.
- D. Define for the model one datagroup that caches for 1 hour. Create a persistent derived table (PDT) for the `daily_forecast` Explore, and apply `sql_trigger_value` to it selecting the current date.

Correct Answer: A

---

## 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 user needs to create a report that shows a count of all orders and of orders over \$100. Which solution should the developer implement to meet these requirements?

A. An always\_filter parameter

B. A front-end filter in the Explore

C. A sql\_always\_where parameter

D. A filtered measure

Correct Answer: B

---

#### 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

[Latest LOOKML-DEVELOPER Dumps](#)

[LOOKML-DEVELOPER PDF Dumps](#)

[LOOKML-DEVELOPER Practice Test](#)