

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.pass2lead.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 LookML developer builds a view that contains sensitive information. Only members of the Management group should have access to the view. The developer needs to restrict the view from appearing in the field picker for any Explore where it might be joined for users outside of the Management group.

Which LookML parameter should the developer use to meet this requirement?

- A. access_grant
- B. always_filter
- C. access_filter
- D. sql_always_where

Correct Answer: A

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 retail company wants to limit who can see the financial information in their reports to executives and store managers. The LookML Developer creates a user attribute called leadership with the value "000" for executives and "999" for store managers. The developer creates three access grant objects and one dimension: How should the developer ensure that only authorized users see the data in the Total Revenue dimension?

```
access_grant: can_view_financial_data_corporate{
user_attribute: leadership
allowed_values: ["000"]
}
access_grant: can_view_financial_data_store_managers{
user_attribute: leadership
allowed_values: ["999"]
}
access_grant: can_view_financial_data_{
user_attribute: leadership
allowed_values: ["000", "999"]
}
dimension: total_revenue
...
required_access_grants: [_____]
...
}
```

- A. required_access_grants: [can_view_financial_data]
- B. required_access_grants: [leadership]
- C. required_access_grants: ["000", "999"]
- D. required_access_grants: [total_revenue]

Correct Answer: A

QUESTION 4

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 5

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 6

The developer has moved the `orders` Explore (shown below) from `model_a` to `model_b`, where both models are in the same project, and all users have access to both models.

```
Connection: "demo" include: ".view" explore: orders {}
```

What will happen after making this change?

- A. Dashboard tiles and Looks will be automatically pointed to the `orders` Explore in `model_b`.
- B. Dashboard tiles and Looks will redirect to the new database connection.
- C. Dashboard tiles and Looks that rely on this Explore will be deleted.

D. Dashboard tiles and Looks that rely on this Explore will return an error.

Correct Answer: C

QUESTION 7

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.DERIVED_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

QUESTION 8

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;"