



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

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



**QUESTION 1**

A user reports an error message on an Explore: “Non-unique value/primary key (or sql_distinct_key), value overflow or collision when computing sum”.

What should the LookML developer check for in the joined views of the Explore?

- A. The sum measure used is defined correctly.
- B. A unique primary key is defined in each view.
- C. Symmetric_aggregates: no is not present in the Explore definition.
- D. No concatenated primary keys are used.

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

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

A user reports the following SQL error when selecting the discounted sale price field:



ERROR: column "order_items.sale_price"; must appear in the GROUP BY clause or be used in an aggregate function.

The developer checks the field definition and finds it to be:

```
measure: discounted_sale_price {  
  
type: number  
  
sql: ${sale_price} * 0.8 ;;  
  
}
```

The developer also finds the LookML definition of the sale_price field to be:

```
dimension: sale_price {  
  
type: number  
  
sql: ${TABLE}.sale_price ;;  
  
}
```

What is the likely cause of the SQL error?

- A. The discounted_sale_price field should have a group_by: yes parameter.
- B. The sale_price field should be defined as a measure of type: number, not as a dimension.
- C. The underlying database table does not have a field called sale_price.
- D. The discounted_sale_price field should be defined as a dimension of type: number, not as a measure.

Correct Answer: B

QUESTION 5

A developer has the dimensions enrollment_month and graduation_month already defined in the view. Both were created as part of dimension_groups of type: time. The developer need to use these two dimensions in the sql_start and sql_end parameters of a dimension group of type: duration.

Which LookML should be used to calculate the number of month and years between enrollment month and graduation month?



- A.
- ```
dimension_group: enrolled{
 type: duration
 intervals: [month, year]

 sql_start: ${enrollment_raw} ;;
 sql_end: $(graduation_raw) ;;
}
```
- B.
- ```
dimension_group: enrolled{
  type: duration
  intervals: [month, year]

  sql_start: ${enrollment} ;;
  sql_end: $(graduation) ;;
}
```
- C.
- ```
dimension_group: enrolled{
 type: duration
 intervals: [month, year]

 sql_start: ${enrollment_day} ;;
 sql_end: $(graduation_day) ;;
}
```
- D.
- ```
dimension_group: enrolled{
  type: duration
  intervals: [month, year]

  sql_start: ${enrollment_month} ;;
  sql_end: $(graduation_month) ;;
}
```



A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: A

[LOOKML-DEVELOPER
Practice Test](#)

[LOOKML-DEVELOPER
Study Guide](#)

[LOOKML-DEVELOPER
Exam Questions](#)