



VCE & PDF

PassApply.com

<https://www.passapply.com/lookml-developer.html>

2024 Latest passapply LOOKML-DEVELOPER PDF and VCE dumps

Download

# 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.passapply.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 wants to calculate the ratio of total sales from the orders view and total users from the users view.

Which two methods can be used to create a measure that meets these requirements? (Choose two.)



A.

```
view: users{
  measure: total_users{
    type: count
  }
  measure: total_sales_per_user {
    type: sum
    sql: 1.0*${orders.total_sales}/${total_users};;
    value_format_name: usd
  }
}

view: orders{
  dimension: sale_price{
    type: number
    sql: ${TABLE}.sale_price;;
  }
  measure: total_sales{
    type: sum
    sql: ${sale_price};;
  }
}
```



```
❑ B. view: users{
  measure: total_users{
    type: count
  }
  measure: total_sales_per_user {
    type: number
    sql: 1.0*${orders.total_sales}/${total_users};;
    value_format_name: usd
  }
}
view: orders{
  dimension: sale_price{
    type: number
    sql: ${TABLE}.sale_price;;
  }
  measure: total_sales{
    type: sum
    sql: ${sale_price};;
  }
}
```

 C.

```
view: users{  
  
  measure: total_users{  
  
    type: count  
  
  }  
}  
  
view: orders{  
  
  dimension: sale_price{  
  
    type: number  
  
    sql: ${TABLE}.sale_price;;  
  
  }  
  
  measure: total_sales{  
  
    type: sum  
  
    sql: ${sale_price};;  
  
  }  
  
  measure: total_sales_per_user {  
  
    type: number  
  
    sql: 1.0*${total_sales}/users.${total_users};;  
  
    value_format_name: usd  
  
  }  
}
```



D.

```
view: users{

measure: total_users{

type: count

}

}

view: orders{

dimension: sale_price{

type: number

sql: ${TABLE}.sale_price;;

}

measure: total_sales{

type: sum

sql: ${sale_price};;

}

measure: total_sales_per_user {

type: number

sql: 1.0*${total_sales}/${users.total_users};;

value_format_name: usd

}

}
```



```
❑ E. view: users{

  measure: total_users{

    type: count

  }

  measure: total_sales_per_user {

    type: number

    sql: 1.0*${total_sales}/${total_users};;

    value_format_name: usd

  }

}

view: orders{

  dimension: sale_price{

    type: number

    sql: ${TABLE}.sale_price;;

  }

  measure: total_sales{

    type: sum

    sql: ${sale_price};;
```



- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

Correct Answer: AC

### QUESTION 2

A developer wants to create a measure that shows the item count broken out by category. When a second, more granular dimension is added to the same query, the count broken out by category should still represent the original aggregation and be duplicated on each line. The business wants this “count” in “category” available in the Explore section without any additional work done by the end user. For example:

The Count column represents the count for each combination of Category and Item.

The Count in Category column represents the count for each Category only.

Category	Item	Count	Count in Category
Fruit	Watermelon	3	10
Fruit	Apple	4	10
Fruit	Orange	3	10
Clothes	Pants	2	5
Clothes	Shirt	3	5

How can the developer address this need with a LookML object?

- A. Create a measure filtered on Category, and make the filter value controlled by a parameter.
- B. Calculate the measure using a derived table, and then join that derived table back into the Explore.
- C. Create a measure with type: sum\_over\_dimension, and make the dimension value controlled by a parameter.
- D. Calculate the overall count using table calculations in the Explore.

Correct Answer: B

### QUESTION 3





A LookML developer has a transactions view with several measures that each perform complex calculations involving multiple fields. The LookML developer creates an Explore based on the transactions view. The product team wants to perform further functions on these measures, such as SUM, AVG, MIN, MAX, and RANK. The team wants these further functions to be performed at different levels of detail: weekly, monthly, and yearly.

How can the LookML developer model these requirements and minimize the amount of code rewriting?

- A. Add measures to the transactions view of type: number to apply the required functions.
- B. Change the existing measures in the transactions view to dimensions, and add measures of the different required types.
- C. Create a constant for each measure so it can be reused across other areas of the LookML project.
- D. Create native derived tables using transactions as the explore\_source.

Correct Answer: D

#### 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

After validating LookML code, a developer receives the following error message:

“Unknown or Inaccessible Field users.name”

What is causing this error?

- A. There is a missing join.
- B. The field is set to “hidden”.
- C. The join relationship is incorrect.



D. The field uses incorrect SQL syntax.

Correct Answer: A

[Latest LOOKML-DEVELOPER Dumps](#)

[LOOKML-DEVELOPER VCE Dumps](#)

[LOOKML-DEVELOPER Practice Test](#)