

DUMPSBOSS.

Google LookML Developer

Google Google-LookML-Developer

Version Demo

Total Demo Questions: 10

Total Premium Questions: 50

Buy Premium PDF

<https://dumpsboss.co>

support@dumpsboss.co

support@dumpsboss.co

dumpsboss.co

QUESTION NO: 1

A developer has a persistent derived table view called `user_facts` that contains aggregated data for each user. The developer needs to query the data from this table in another derived table view called `user_region_facts`.

Which strategy should the developer use to write the query for `user_region_facts` that will leverage the existing derived table?

- A. Use `$(user_facts.SQL_TABLE_NAME)` to reference the `user_facts` derived table.
- B. Copy the name of the database table in the scratch schema for the `user_facts` derived table.
- C. Write the query from `user_facts` into a common table expression (`WITH user_facts AS...`).
- D. Write a subquery in the `FROM` clause and alias with `$(user_facts)`.

ANSWER: C

QUESTION NO: 2

Business users report that an ephemeral derived table tile on the dashboard is slow.

Information about the dashboard includes:

The dashboard filter is linked to the user attributes.

This tile usually takes approximately 5 minutes to complete running.

Which solution should be used to improve the dashboard load time?

- A. Use a conditional `WHERE` clause for Development Mode.
- B. Build a user attribute filter into the Explore.
- C. Use `index_distribution_key` or `sort_key` for this derived table.
- D. Persist the derived table.

ANSWER: D

Explanation:

Reference: <https://docs.looker.com/reference/dashboard-reference>

QUESTION NO: 3

After running the Content Validator, a developer can see the error “Unknown field”.

Which two changes could cause this issue? (Choose two.)

- A. View name was changed from users to customers.
- B. Field type was changed from number to string.
- C. Model name was changed from e_commerce to reporting.
- D. Explore label was changed from users to customers.
- E. Field name was changed from id to user_id.

ANSWER: B E

QUESTION NO: 4

Business users report that they are unable to build useful queries because the list of fields in the Explore is too long to find what they need.

Which three LookML options should a developer use to curate the business user’s experience? (Choose three.)

- A. Add a description parameter to each field with context so that users can search key terms.
- B. Create a separate project for each business unit containing only the fields that the unit needs.
- C. Add a group_label parameter to relevant fields to organize them into logical categories.
- D. Use the hidden parameter to remove irrelevant fields from the Explore.
- E. Use a derived table to show only the relevant fields.

ANSWER: A C E

QUESTION NO: 5

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.

ANSWER: B C

QUESTION NO: 6

A developer creates a derived table and wants to add persistence to it. Because the table is not used on a frequent basis, the developer wants the table to be cached for 12 hours, but only when a user has queried it.

Which persistence parameter should be added to the derived table's definition in order to satisfy this use case?

- A. persist_with: "12 hours"
- B. datagroup: 12_hours { max_cache_age: "12 hours" }
- C. persist_for: "12 hours"
- D. sql_trigger_value: SELECT FLOOR{UNIX_TIMESTAMP} / {6*60*60} ;;

ANSWER: A

Explanation:

Reference: <https://docs.looker.com/data-modeling/learning-lookml/caching>

QUESTION NO: 7

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};;
```

```
sql: ${sale_price};;
```

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

ANSWER: A C

Explanation:

Reference:<https://docs.looker.com/data-modeling/learning-lookml/advanced-lookmlconcepts>

QUESTION NO: 8

Users report that the main dashboard has been slow to show results.

Which two options should the developer evaluate to improve dashboard performance?

(Choose two.)

- A. Number of databases used by dashboard elements
- B. Number of queries used by the dashboard
- C. Ratio of visualizations to text tiles
- D. Format used to deliver these reports
- E. Amount of data rendered for each query

ANSWER: B C

Explanation:

Reference:<https://help.looker.com/hc/en-us/articles/360038233334-Considerations-When->

QUESTION NO: 9

A developer wants to create a new Explore based on the order_items view. The developer creates an Explore in the ecommerce model file with the following definition:

```
explore: order_items {}
```

After saving and validations, the developer receives this LookML validator error:

Inaccessible view "inventory_items", "inventory_items" is not accessible in explore "order_items". Check for typos and missing joins in explore "order_items".

What caused this error to appear?

- A. A field in the order_items view references a field in the inventory_items view.
- B. A field in the inventory_items view references a field in the order_items view.
- C. There is an Explore named inventory_items which references the order_items view.
- D. There is another Explore named order_items which references the inventory_items view.

ANSWER: A

QUESTION NO: 10

Users viewing an Explore should be able to view rows of data only where the value of the product.brand column matches the value of the user's company user attribute.

Which access filter should the developer use to meet this requirement?

A)

```
access_filter: {  
  field: company  
  user_attribute: ${product.brand}  
}
```

B)

```
access_filter: {  
  field: product.brand  
  user_attribute: company  
}
```

C)

```
access_filter: {  
  field: user.company  
  user_attribute: brand  
}
```

D)

```
access_filter: {  
  field: product.brand  
  user_attribute: {{ _user_attributes['company'] }}  
}
```

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

ANSWER: B