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.geekcert.com/lookml-developer.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Google Official Exam Center

QUESTION 1

A user reports that a query run against the orders Explore takes a long time to run. The query includes only fields from the users view. Data for both views is updated in real time. The developer runs the following query in SQL Runner and quickly receives results:

SELECT * FROM users.

What should the developer do to improve the performance of the query in the Explore?

A. Create an Explore with users as the base table.

B. Create a persistent derived table from the user\\'s query.

C. Create an ephemeral derived table from the user\\'s query.

D. Add persist_for: "24 hours" to the orders Explore.

Correct Answer: A

QUESTION 2

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 3

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 4

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 5

A developer needs to implement three persistent derived tables (PDTs) as described below.

The PDTs need to be refreshed after the daily ETL pipeline adds incremental loads to the underlying

tables.

Each PDT is built off of one underlying table in the database (one PDT per table).

The underlying tables for each PDT are updated one after the other, and a new row is added to an ETL log

table each time a table is updated.

Due to the unpredictable nature of the ETL pipeline, each PDT does not refresh at the same time from day

to day.

Each PDT takes over an hour to build, and to save on compute costs each PDT should only be refreshed

once per day.

How can the developer set up the PDTs according to these requirements?

A. Create one datagroup tied to all three PDTs that runs when the total row count across all three tables changes.

B. Create one datagroup tied to all three PDTs that parameterizes the view name for each PDT in the SQL trigger condition.

C. Create three separate datagroups tied to three PDTs that run when each corresponding table\\'s row count changes.

D. Create three separate datagroups tied to three PDTs that run when a new row is added to the ETL log table.

Correct Answer: D