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

QUESTION 1

A developer wants to create a measure that shows the item count broken out by category. When a second,

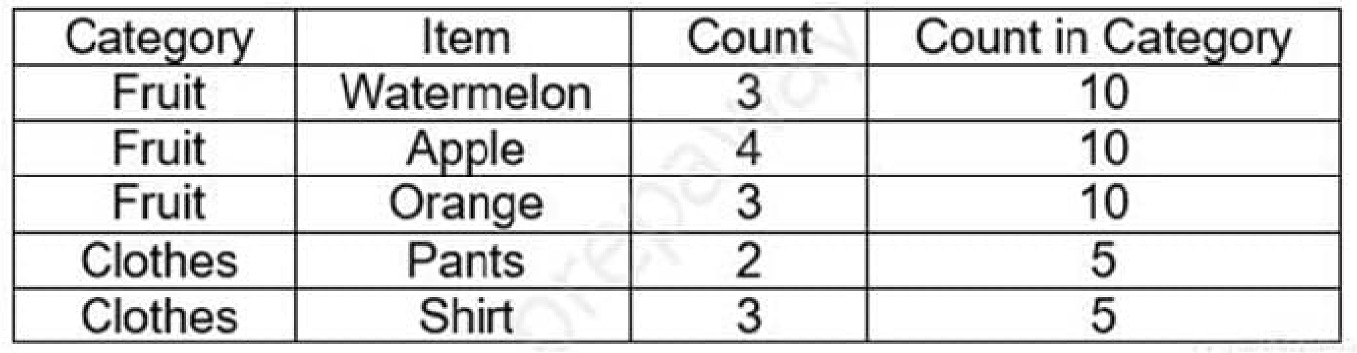
more granular dimension is added to the same query, the count broken out by category should still

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.

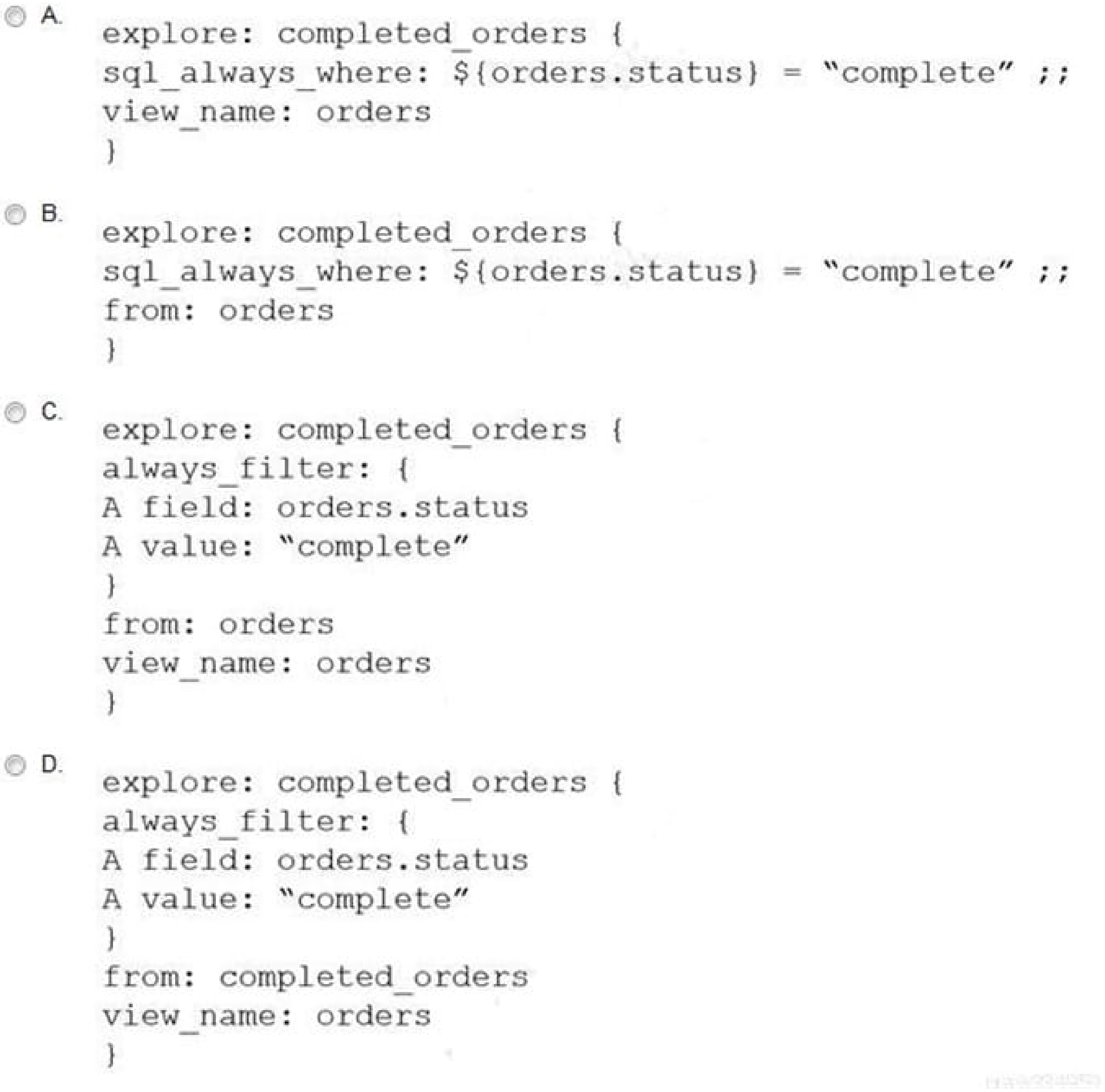
D. Calculate the overall count using table calculations in the Explore.

Correct Answer: B

QUESTION 2

A developer needs to add an Explore built off of the orders view, which surfaces only completed orders. An orders Explore exists that contains all order information. Fields from the orders view are also referenced in other existing views such as ${orders.fieldname}.

How should developer define a new Explore for completed orders and keep all field references working correctly?



○ A.

```
explore: completed_orders {
sql_always_where: ${orders.status} = "complete" ;;
view_name: orders
}
```

○ B.

```
explore: completed_orders {
sql_always_where: ${orders.status} = "complete" ;;
from: orders
}
```

○ C.

```
explore: completed_orders {
always_filter: {
A field: orders.status
A value: "complete"
}
from: orders
view_name: orders
}
```

○ D.

```
explore: completed_orders {
always_filter: {
A field: orders.status
A value: "complete"
}
from: completed_orders
view_name: orders
}
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: C

QUESTION 3

A developer is connecting a LookML project to a remote Git repository. The developer wants to track which users are committing code changes, creating pull requests, or deploying to production when the different Git commands are initiated from within Looker.

Which type of Git connection should be utilized to meet this business requirement?

A. A bare Git repository

B. Multiple account HTTPS

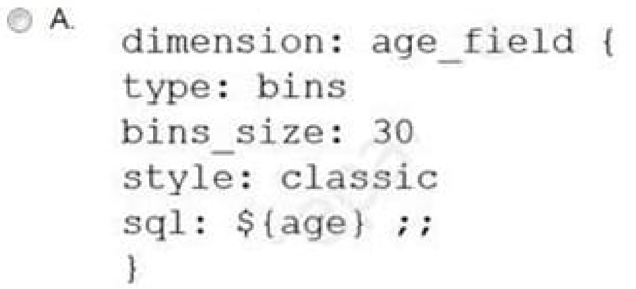
C. Single account HTTPS

D. SSH

Correct Answer: D

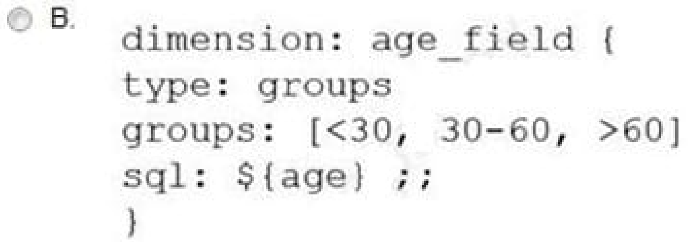
QUESTION 4

A developer needs to build a new dimension that offers an age-based cohort representation of users. Which LookML code should the developer use to meet the requirement?



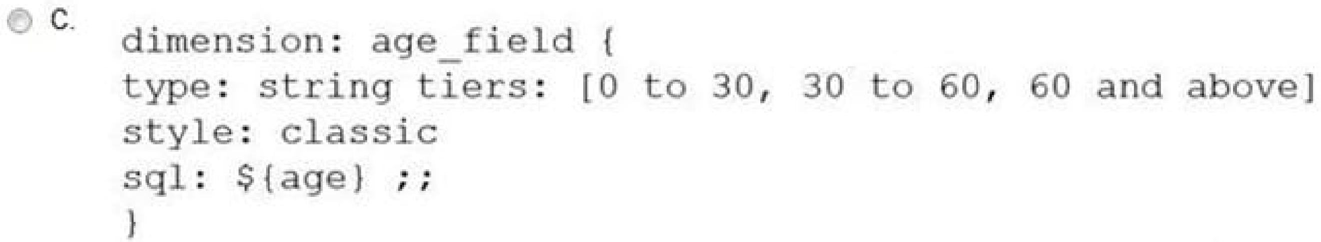
○ A.

```
dimension: age_field {
type: bins
bins_size: 30
style: classic
sql: ${age} ;;
}
```



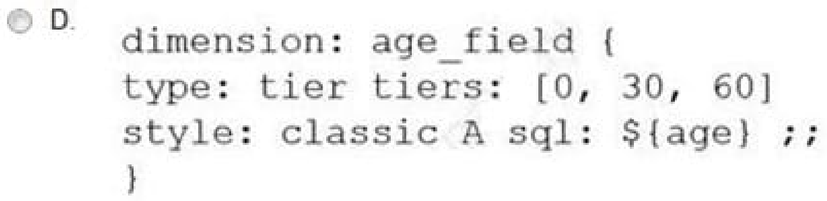
○ B.

```
dimension: age_field {
type: groups
groups: [<30, 30-60, >60]
sql: ${age} ;;
}
```



○ C.

```
dimension: age_field {
type: string tiers: [0 to 30, 30 to 60, 60 and above]
style: classic
sql: ${age} ;;
}
```



○ D.

```
dimension: age_field {
type: tier tiers: [0, 30, 60]
style: classic A sql: ${age} ;;
}
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: B

QUESTION 5

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