Exam	**LookML Developer**
Title	**Looker LookML Developer Exam**
Product Type	**50 Q&A with explanations**

QUESTION 1
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
}
```

Answer: C
Section: (none)
Explanation
Explanation/Reference:

Reference: https://hevodata.com/learn/understanding-looker-ml/

Data Analytics and Business Intelligence (BI) are some of the few technologies that companies are incorporating to better understand their customers. Many tools are built in this field to help companies gain valuable customer insights. One such tool is Looker. Looker is a Business Intelligence software and Big Data Analytics platform that helps companies explore, analyze and share business insights easily in real-time. Looker uses **Looker ML** (LookML) to describe the relationships between various attributes in a database.

Looker ML is the language Looker uses to describe the dimensions and relationships between various entities in an SQL database. Understanding Looker ML plays a pivotal role in determining the relationships between different database entities and also to construct the correct SQL queries.

QUESTION 2
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
Section: (none)
Explanation
Explanation/Reference:

QUESTION 3
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.

Answer: A
Section: (none)

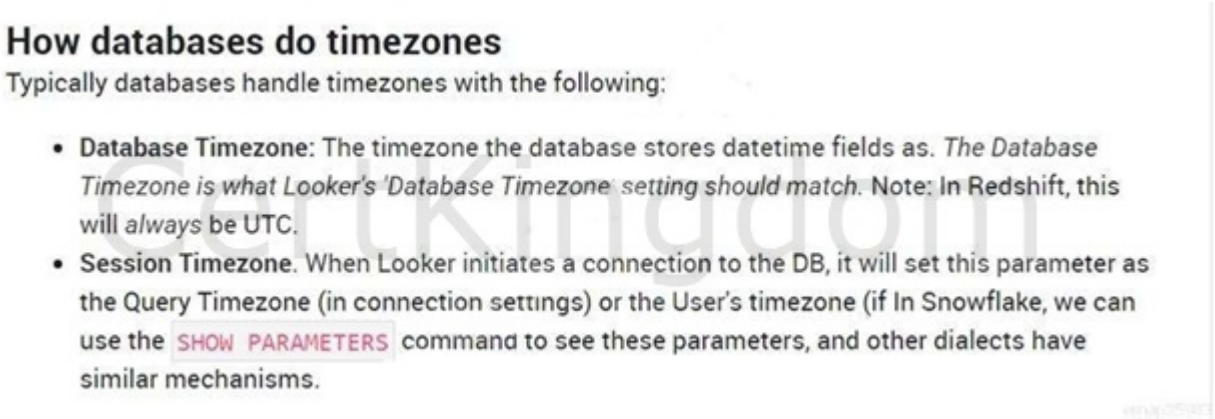
Explanation
Explanation/Reference:
Reference: https://docs.looker.com/data-modeling/learning-lookml/sql-runner

QUESTION 4
A developer has User Specific Time Zones enabled for a Looker instance, but wants to ensure that queries run in Looker are as performant as they can be. The developer wants to add a datatype: date parameter to all dimension_group definitions without time data in a table-based view, so that time conversions don?t occur for these fields.
How can the developer determine to which fields this parameter should be applied through SQL Runner?

A. Open the Explore query in SQL Runner and validate whether removing the conversion from date fields changes the results.
B. Open the Explore query in SQL Runner to determine which fields are converted.
C. Use the CAST function in SQL Runner to ensure that all underlying fields are dates and conversions are not applied.
D. Use the Describe feature in SQL Runner to determine which fields include time data.

Answer: C
Section: (none)
Explanation
Explanation/Reference:
Reference: https://community.looker.com/technical-tips-tricks-1021/how-looker-does-timezones-and-how-totroubleshoot-

them-25477



QUESTION 5
A LookML developer creates a new model and a test dashboard from the model. The developer shares the link to the new dashboard with users, but the users report that all they see is the ?Model Not Found? error.
What is a possible cause of this issue?

A. The developer has not pushed the new model to Production Mode.
B. The developer has not added users to the new model set.
C. The users do not have permission to access this dashboard.
D. The new model is missing an Explore definition.

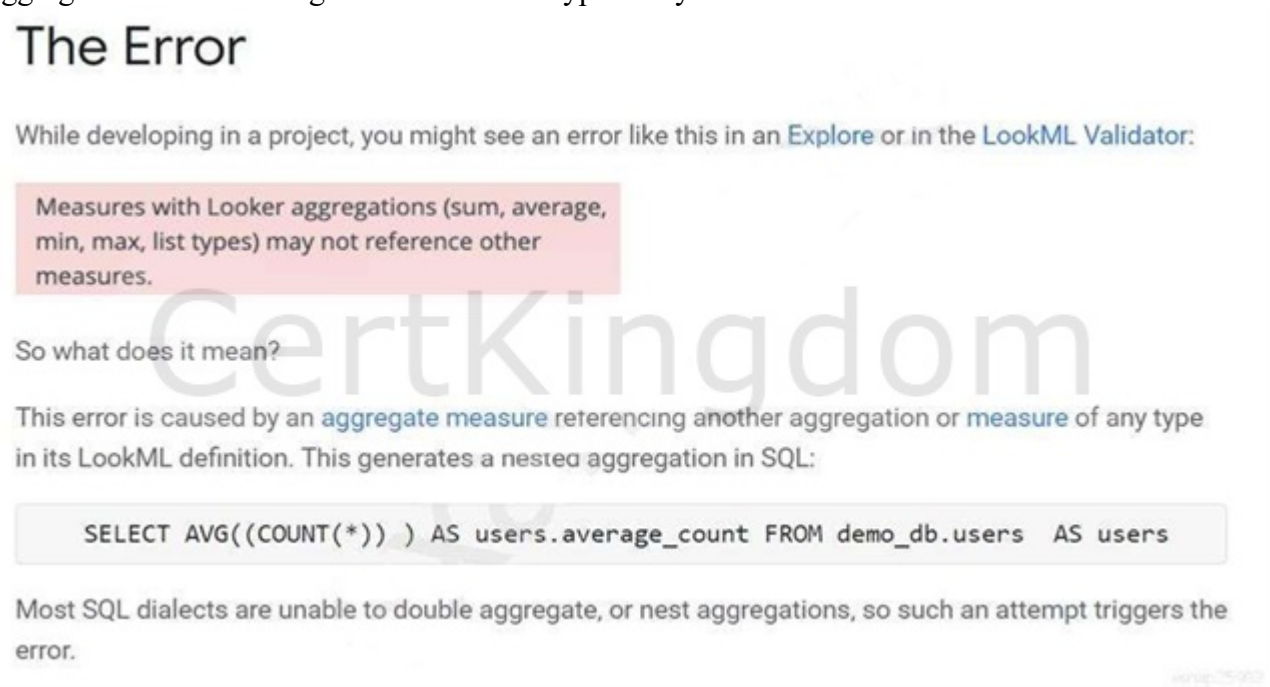
Answer: B
Section: (none)
Explanation
Explanation/Reference:

QUESTION 6
After running the LookML Validator, a developer sees the following error message in the Looker development environment:
?Measures with Looker aggregations (sum, average, min, max, list types) may not reference other measures?.
What could be causing this error?

A. A measure of type: count has a sql parameter defined.
B. A measure of type: sum adds up other measures in the sql parameter.
C. A measure of type: sum has a SUM function written in the sql parameter.
D. A measure of type: number has a SUM function written in the sql parameter.

Answer: A
Section: (none)
Explanation
Explanation/Reference:
Reference: https://help.looker.com/hc/en-us/articles/360038371614--Error-Measures-with-Looker-

Aggregations-Sum-Average-Min-Max-List-Types-May-Not-Reference-Other-Measures

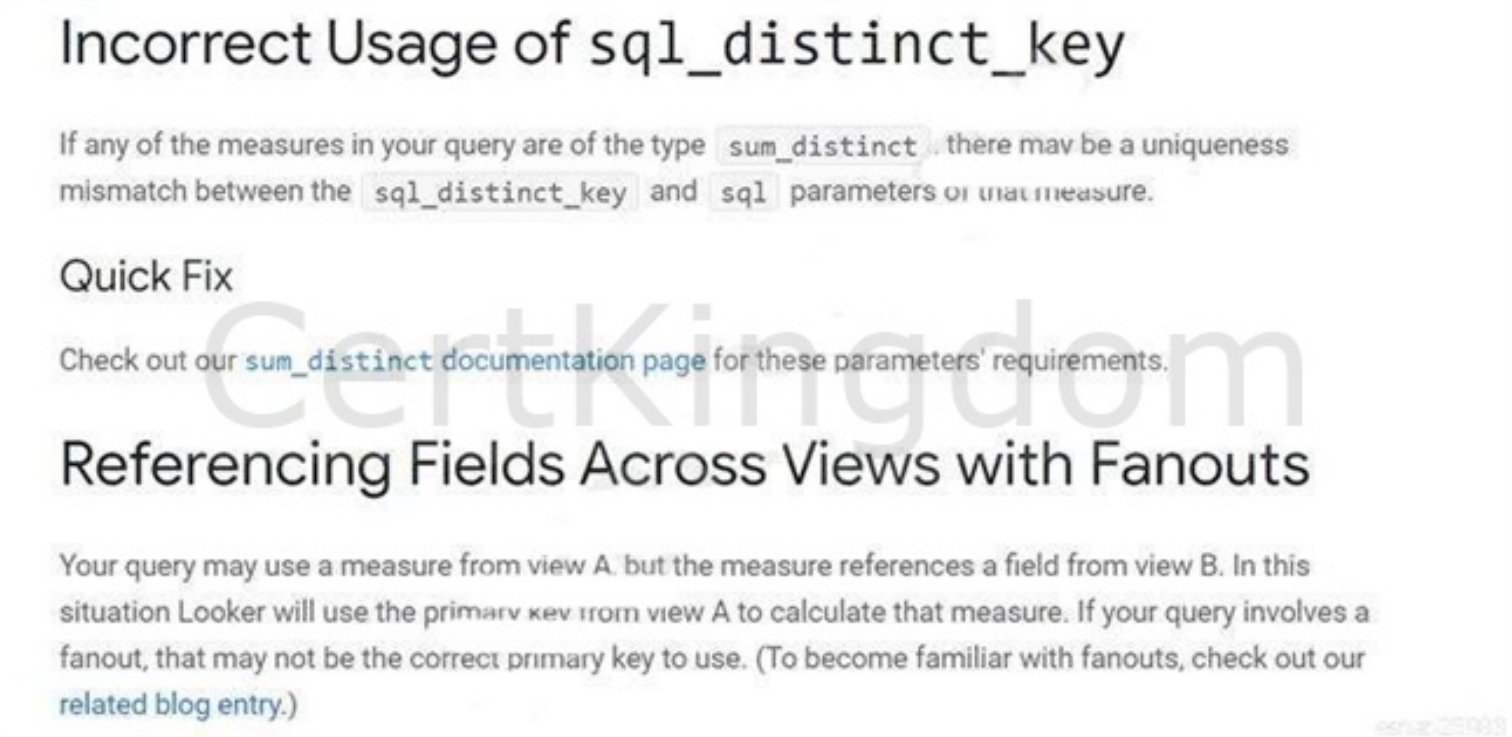


QUESTION 7
A user is seeing an error in the Explore that indicates the primary key defined for a one-million-row table is not unique.
How can the developer use SQL Runner to troubleshoot quickly?

A. Create a query that selects all the fields from the table, and sort by primary key.
B. Create a query that selects the primary key from the base view, and look for duplicates.
C. Create a query that counts how many occurrences of the primary key value are in the base view, and sort by count.
D. Create a query that concatenates two columns to create a compound primary key.

Answer: D
Section: (none)
Explanation
Explanation/Reference:
Reference: https://help.looker.com/hc/en-us/articles/360037732513-Error-Non-Unique-value-primary-key-orsql-

distinct-key-value-overflow-or-collision-when-computing-sum

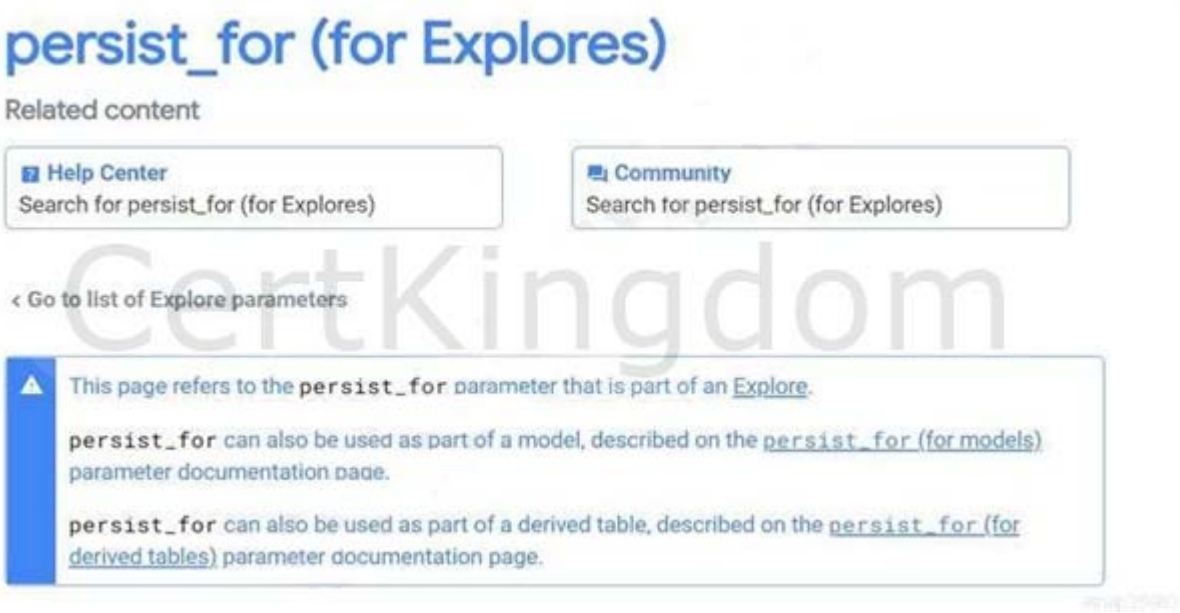


QUESTION 8
The daily_forecast Explore used by the sales team needs to be cached for 24 hours. All other Explores used by the sales team need to be cached for one hour.
What is a scalable way to configure this caching logic?

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.

Answer: A
Section: (none)
Explanation
Explanation/Reference:

Reference: https://docs.looker.com/reference/explore-params/persist_for-for-explore



QUESTION 9

A LookML developer has written the following persistent derived table. It references orders_rollup, another persistent derived table that also rebuilds with the same SQL trigger value.

```
view: user_facts {

derived_table: {

sql_trigger_value: SELECT "current date function";;

sql: SELECT col1, col2, col3

FROM ${orders_rollup.SQL_TABLE_NAME} ;;

}
}
```


Which change is needed to guarantee that user_facts will always rebuild with the latest data from orders_rollup?

A. Change the sql_trigger_value parameter of user_facts to select the current date plus one hour, so it triggers an hour after orders_rollup.
B. Change the orders_rollup view reference to ${orders_rollup.DERVIED_TABLE_NAME}
C. Change the sql_trigger_value parameter for both persistent derived tables to a datagroup_trigger parameter, and set them to use the same datagroup.
D. Change the orders_rollup view reference to the literal table name from the database?s scratch schema.

Answer: C
Section: (none)
Explanation
Explanation/Reference:

Reference: https://docs.looker.com/reference/view-params/sql_trigger_value

QUESTION 10

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:

```
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: [_____]

...
}
```


How should the developer ensure that only authorized users see the data in the Total Revenue dimension?

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]

Answer: A
Section: (none)
Explanation
Explanation/Reference:
Reference: https://community.looker.com/technical-tips-tricks-1021/how-to-use-and-troubleshoot-accessgrants-

field-permissions-23986

SYNTAX

Access grants are built in the model file.
You can define the permission like this:

```
access_grant: privileged_departments {

user_attribute: department

allowed_values: ["hr", "finance"]

}
```

Allowed_values use OR logic - any of the values allow access.

Access grants are applied at the explore, join, view or field level. You can call the access grant with the required_access_grants parameter. Ex:

```
view: view_name {

required_access_grants: [privileged_departments]
```

QUESTION 11
A user needs to create a report that shows a count of all orders and of orders over $100.
Which solution should the developer implement to meet these requirements?

A. An always_filter parameter
B. A front-end filter in the Explore
C. A sql_always_where parameter
D. A filtered measure

Answer: B
Section: (none)
Explanation
Explanation/Reference:

QUESTION 12
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
Section: (none)

Explanation
Explanation/Reference:

QUESTION 13
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.

Answer: B
Section: (none)
Explanation
Explanation/Reference:
Reference: https://docs.looker.com/dashboards/creating-lookml-dashboards

QUESTION 14
Only users with department attributes of Finance and Executive should be able to access the revenue view. Only users with the value of Executive for the department user attribute should be able to view the total_revenue field.
Given the code snippet below:

```
explore: financial_data {
  view_name: base_table

  join: revenue {▭}
}

view: revenue {
  measure: total_revenue
}

access_grant: grant_a
  user_attribute: department
  allowed_values: ["executive"]
}

access_grant: grant_b {
  user_attribute: department
  allowed_values: ["finance", "executive"]
}
```


How should the required access grants be structured to set up this system of access?

A. required_access_grants: [grant_b] in the revenue view, required_access_grants: [grant_a] in the total_revenue field
B. required_access_grants: [grant_a] in the revenue view, required_access_grants: [grant_a, grant_b] in the total_revenue field
C. required_access_grants: [grant_b] in the financial_data Explore, required_access_grants: [grant_a] in the total_revenue field
D. required_access_grants: [grant_a, grant_b] in the revenue view, required_access_grants: [grant_a] in the total_revenue field

Answer: B
Section: (none)
Explanation
Explanation/Reference:

QUESTION 15
The code below shows a view order_items with its measures total_revenue and user_count

```
view: order_items {

measure: total_revenue {

type: sum

sql: ${TABLE}.sale_price ;;

}

measure: user_count {

type: count_distinct

sql: ${users.id} ;;

}
}
```

Which code correctly represents a new measure that calculates average revenue per user?

A.

```
measure: average_revenue_per_user {

type: number

sql: ${total_revenue}/${user_count} ;;

}
```

B.

```
measure: average_revenue_per_user {

type: average

sql: ${total_revenue}/${user_count} ;;

}
```

C.

```
measure: average_revenue_per_user {

type: number

sql: ${total_revenue}/${users.id};;

}
```

D.

```
measure: average_revenue_per_user {

type: average sql: ${total_revenue}/${users.id};;

}
```

Answer: C
Section: (none)
Explanation
Explanation/Reference:

QUESTION 16
Users report that the main dashboard has been slow to show results.
Which two options should the developer evaluate to improve dashboard performance? (?hoose 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
Section: (none)
Explanation
Explanation/Reference:
Reference: https://help.looker.com/hc/en-us/articles/360038233334-Considerations-When-Building-Performant-

Looker-Dashboards

One of the best ways to empower end users to explore data is by providing them with curated views by building effective Looker dashboards. If you want to create a great performance experience for your users, consider the tips in this article as you design your dashboards.

Looker Dashboards load in the browser. To build for optimal performance, keep these facts in mind:

The most important element of dashboard performance is the underlying SQL query performance. Each dashboard element, when not returned from cache, runs a SQL query that takes time to execute on the underlying database. See the **Optimize Query Performance** section of the Best Practice: Optimize Looker Performance Help Center article for more details regarding building performant queries.

Some components are more memory intensive than they are SQL related — these can cause slow performance in dashboards:

QUESTION 17
The developer has moved the orders Explore (shown below) from model_a to model_b, where both models are in the same project, and all users have access to both models.
Connection: ?demo?
include: ?.view?
explore: orders {}
What will happen after making this change?

A. Dashboard tiles and Looks will be automatically pointed to the orders Explore in model_b.
B. Dashboard tiles and Looks will redirect to the new database connection.
C. Dashboard tiles and Looks that rely on this Explore will be deleted.
D. Dashboard tiles and Looks that rely on this Explore will return an error.

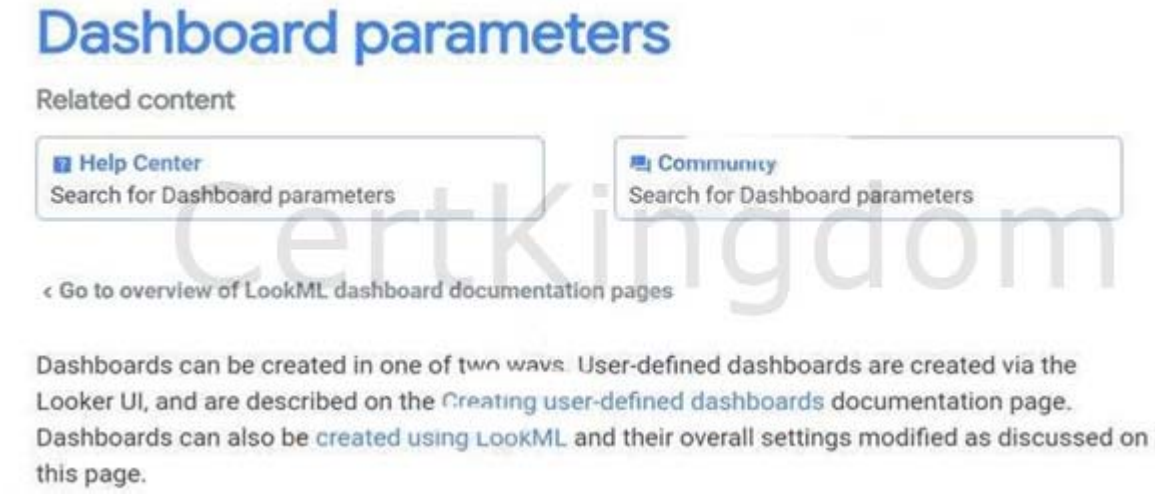
Answer: C
Section: (none)
Explanation
Explanation/Reference:

QUESTION 18
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
Section: (none)
Explanation
Explanation/Reference:

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



QUESTION 19

A developer has created a persistent derived table that tracks new or updated orders and they want to cache the results. The cache should be refreshed whenever some new order is available on the underlying datasource table my tablename or at least every 24 hours.

Which datagroup definition will refresh the cache as expected?

A.

```
datagroup: my_datagroup {

sql_trigger: SELECT current_date FROM my_tablename ;;

max_cache_age: "24 hours"

}
```

B.

```
datagroup: my_datagroup {

sql_trigger: SELECT max(order_id) FROM my_tablename ;;

max_cache_age: "24 hours"

}
```

C.

```
datagroup: my_datagroup {

sql_trigger: SELECT max(current_date) FROM my_tablename ;;

max_cache_age: "1 day"

}
```

D.

```
datagroup: my_datagroup {

sql_trigger: SELECT max(order_id) FROM my_tablename ;;

max_cache_age: "1 day"

}
```

Answer: A
Section: (none)
Explanation
Explanation/Reference:

QUESTION 20
A developer is building an e-commerce Explore with the following datasets: orders and users. The business user needs to be able to answer questions about sellers and buyers within the same Explore. Each order in the orders table reports a buyer and seller ID. The users table has the detailed information about the individual buyer and seller.
How should the Explore be defined to meet this requirement?
A.

```
explore: orders

join: buyers {

view_name: users

sql_on: ${orders.buyer_id} = ${buyers.id} ;;

relationship: many_to_one

}

join: sellers {

view_name: users

sql_on: ${orders.seller_id} = ${sellers.id} ;;

relationship: many_to_one

}
```

B.

```
explore: orders

join: users {

sql_on: ${orders.buyer_id} = ${users.id} AND ${orders.seller_id

A relationship: many_to_one

}
```

C.

```
explore: orders

join: buyers {

from: users

sql_on: ${orders.buyer_id} = ${buyers.id} ;;

relationship: many_to_one

}

join: sellers {

from: users

sql_on: ${orders.seller_id} = ${sellers.id} ;;

relationsh. : many_to_one

}
```

D.

```
explore: orders

join: users {

sql_on: ${orders.buyer_id} = ${users.id} OR ${orders.seller_id}

relationship: many_to_one

}
```

Answer: B
Section: (none)

Explanation
Explanation/Reference:

QUESTION 21
A LookML developer creates an Explore that joins two views. The base view has information about users? interactions with the support team. The joined view contains data about the users. The support team using this Explore feels overwhelmed by the amount of data this Explore shows them and decides to just look at open tickets.
What should the developer add to the Explore in the model to achieve these requirements?

A. A filtered measure
B. The hidden parameter
C. The sql_always_where parameter
D. A relationship definition

Answer: D
Section: (none)
Explanation
Explanation/Reference:

QUESTION 22
A developer would like to add a new dimension of type: yesno for the enabled column in their users table. The column is of type: string in the database and returns yes and no values.
How should the developer define the yesno dimension?
A.

```
dimension: is_enabled {

type: yesno

sql: $(TABLE).enabled IS NOT NULL ;;

}
```

B.

```
dimension: is_enabled {

type: yesno

sql: CASE WHEN $(TABLE).enabled = ""yes"" then ""Yes"" ELSE ""N
END;;

}
```

C.

```
dimension: is_enabled {

type: yesno

sql: $(TABLE).enabled ;;

}
```

D.

```
dimension: is_enabled {

type: yesno

sql: $(TABLE).enabled = ""yes"" ;;

}
```

Answer: A
Section: (none)
Explanation
Explanation/Reference:

QUESTION 23
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.

Answer: B
Section: (none)

Explanation
Explanation/Reference:

QUESTION 24
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

Answer: D
Section: (none)
Explanation
Explanation/Reference:
Reference: https://docs.looker.com/data-modeling/getting-started/version-control-and-deploying-changes

QUESTION 25
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. Writhe the query form 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
Section: (none)
Explanation
Explanation/Reference:

QUESTION 26
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'] }}
}
```

Answer: B
Section: (none)
Explanation
Explanation/Reference:

QUESTION 27
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.

Answer: B
Section: (none)
Explanation
Explanation/Reference:

QUESTION 28
A LookML developer is creating a new view with 20 dimensions in Development Mode using the Create View from Table functionality. Now users want the developer to add only four dimensions from this new view to an existing Explore.
What can the developer add to the Explore to limit the number of fields from the view that are accessible to the user in the Explore?

A. Set definition
B. Join condition
C. Fields parameter
D. Hidden parameter

Answer: B
Section: (none)
Explanation
Explanation/Reference:
Reference: https://docs.looker.com/reference/view-params/view

QUESTION 29
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
Section: (none)
Explanation
Explanation/Reference:

QUESTION 30
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.

Answer: C
Section: (none)
Explanation
Explanation/Reference:

QUESTION 31
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.

Answer: D
Section: (none)
Explanation
Explanation/Reference:

QUESTION 32
The developer is creating an Explore that includes the product users, and orders views that will meet the following guidelines.
Joins between the orders and users views should not incur high performance costs.

Users of this Explore will primarily be looking at data from the orders view.
Users of this Explore should only be able to see orders from the retailer ?Fashion.ly?.
The only field the users need from the products view is product.name.
Which LookML should the developer use?
A.

```
explore: orders {

join: product {

fields: [product.name]

join: users {...}

sql_always_where: ${orders.retailer} = 'Fashion.ly' ;;

}
```

B.

```
explore: orders {

fields: [product.name]

join: product {...}

join: users {...}

sql_always_where: ${orders.retailer} = 'Fashion.ly' ;;

}
```

C.

```
explore: users {

join: product {

fields: [product.name]

}

join: orders {...}

always_filter: {

filters: {

fields: orders.retailer

value: "Fashion.ly"

}}}
```

D.

```
explore: users {

join: product {

fields: [product.name]

}

join: orders {...}

sql_always_where: ${orders.retailer} = 'Fashion.ly' ;;

}
```

Answer: D

Section: (none)
Explanation
Explanation/Reference:

QUESTION 33
Users must be able to click on the Country field in their Explore and be redirected to another Explore that shows all countries compared.
Which parameter should be added to the country dimension to create a connection to this other associated

Explore?

A. url_encode
B. drill_fields
C. tags
D. link

Answer: D
Section: (none)
Explanation
Explanation/Reference:
Reference: https://docs.looker.com/setup-and-management/connecting-to-db

QUESTION 34
A developer commits changes after adding LookML for a new measure. Upon pulling from production, the developer notices the following lines in the LookML:

```
<<<<<< HEAD
measure: metric_b {
type: average
sql: ${item.price} ;;
}
======
dimension: metric_a {
type: number
sql: ${item.price} ;;
}
>>>>>> branch 'master'
```

A. Remove ?<<<<<< HEAD?, ?======?, and ?>>>>>> branch ?master??
B. Remove ?<<<<<< HEAD?, ?======?, and everything following ?======?
C. Remove everything between ?<<<<<< HEAD? and ?======?, and ?>>>>>> branch ?master??
D. Remove everything between ?<<<<<< HEAD? and ?>>>>>> branch ?master??

Answer: D
Section: (none)

Explanation
Explanation/Reference:

QUESTION 35
A developer needs to model out LookML to convert existing reports into Looker. The existing reports are:
Report 1: A report with order and order_items data, which finds the order with the largest total value of the order_item prices.
Report 2: A report with order and order_items data, which finds the order with the largest total number of products ordered.
Report 3: A report with data on every product, whether or not it has been ordered.
Each database table used is updated in real time as orders are made.
How should the developer construct an Explore using the order_items view as the base view?

A. Create one persistent derived table to calculate Report 1, create one persistent derived table to calculate Report 2, and join in the products view with a full_outer join.

B. Create one persistent derived table to calculate
Reports 1 and 2, and join in the products view with a full_outer join.

C. Create one ephemeral derived table to calculate Report 1, create one ephemeral derived table to calculate Report 2, and join in the products view with a left_outer join.

D. Create one ephemeral derived table to calculate
Reports 1 and 2, and join in the products view with a full_outer join.

Answer: A
Section: (none)
Explanation
Explanation/Reference:

QUESTION 36
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
Section: (none)
Explanation
Explanation/Reference:

QUESTION 37
Users report that every time they change the filter on their Explore, the filters take a very long time to populate.
How can the developer improve the filtering experience with this Explore?

A. Limit the filter suggestions using the suggestions parameter.
B. Add an always_filter parameter to restrict the filter suggestions.
C. Use an access_filter parameter to automatically apply filters.

D. Add persistence to the base view of the Explore.

Answer: A
Section: (none)
Explanation
Explanation/Reference:

QUESTION 38
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.

Answer: D
Section: (none)
Explanation
Explanation/Reference:

QUESTION 39
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.

Answer: A
Section: (none)
Explanation
Explanation/Reference:

Reference: https://help.looker.com/hc/en-us/articles/360023586293-Error-Unknown-or-Inaccessible-Field

QUESTION 40

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} ;;
}
```

Answer: B

Section: (none)
Explanation
Explanation/Reference:

QUESTION 41
A LookML Developer is working with denormalized tables and needs to create a measure adding up the Order Shipping column in the table below:

Order Item ID	Order ID	Order Shipping
1	1	10.00
2	1	10.00
3	2	20.00
4	2	20.00
5	2	20.00

A.

```
measure: total_shipping {
type: sum
sql: ${order_shipping} ;;
}
```

B.

```
measure: total_shipping {
type: sum_distinct
sql: ${order_shipping} ;;
}
```

C.

```
measure: total_shipping {
type: sum_distinct
sql_distinct_key: ${order_id} ;;
sql: ${order_shipping} ;;
}
```

D.

```
measure: total_shipping {
type: sum
sql_distinct_key: ${order_id} ;;
 sql: ${order_shipping} ;;
}
```

Answer: A
Section: (none)
Explanation
Explanation/Reference:

QUESTION 42
A LookML developer finishes some LookML work and commits changes in their personal development branch.

The developer is asked to Pull and Merge Other Changes.
What does this indicate?

A. A change has been deployed to a shared branch.
B. A change has been committed in another developer?s personal branch.
C. A change has been committed in another shared branch.
D. A change has been deployed to production.

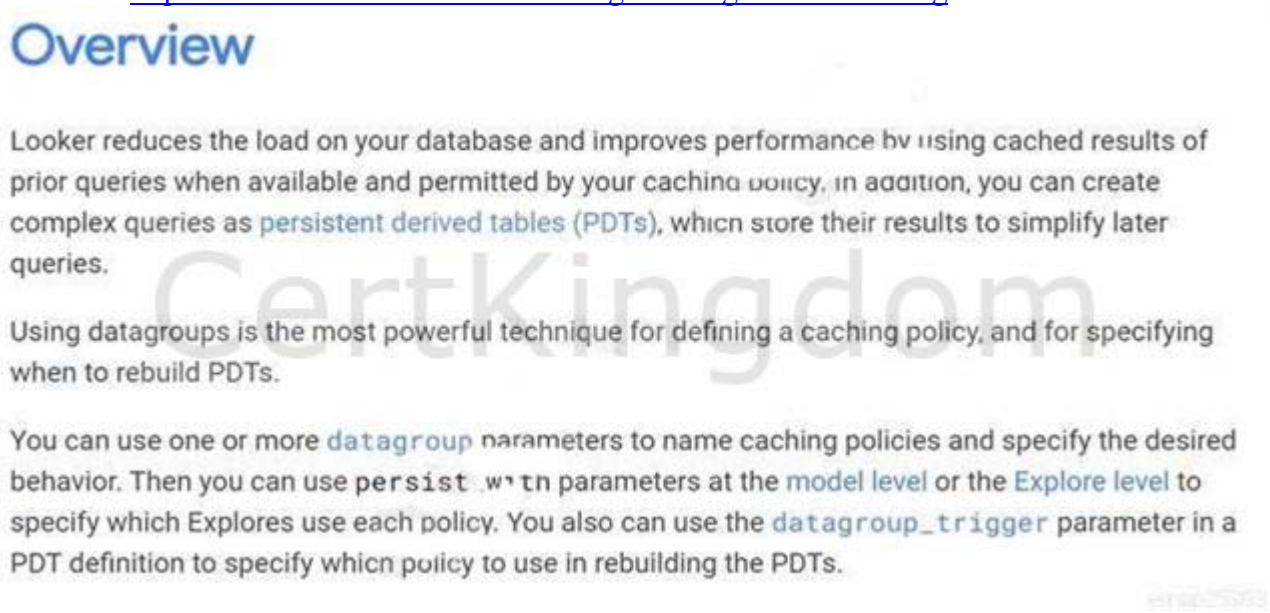
Answer: B
Section: (none)
Explanation
Explanation/Reference:
Reference: https://docs.looker.com/data-modeling/getting-started/version-control-and-deploying-changes

QUESTION 43
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
Section: (none)
Explanation
Explanation/Reference:

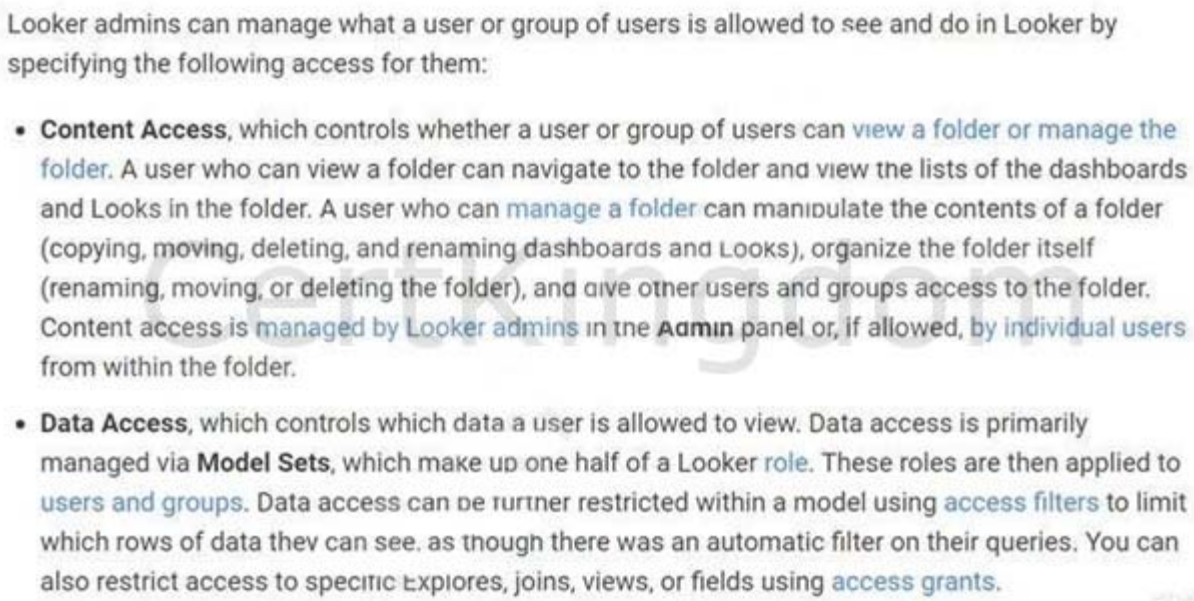
Reference: https://docs.looker.com/data-modeling/learning-lookml/caching



QUESTION 44
A LookML developer builds a view that contains sensitive information. Only members of the Management group should have access to the view. The developer needs to restrict the view from appearing in the field picker for any Explore where it might be joined for users outside of the Management group.
Which LookML parameter should the developer use to meet this requirement?

A. access_grant
B. always_filter
C. access_filter
D. sql_always_where

Answer: A
Section: (none)
Explanation
Explanation/Reference:
Reference: https://docs.looker.com/admin-options/tutorials/permissions



QUESTION 45
A developer defines the following measure in the order_items view:

```
measure: total_gross_margin {

  type: sum

  sql: ${sale_price} - ${inventory_items.cost} ;;

}
```

The code must validate without errors.
Which action should the developer take?

A. Join order_items and inventory_items in a derived table.
B. Join order_items and inventory_items in the same Explore.
C. Copy the cost definition from inventory_items to the order_items view file.
D. Add the following to the order_items view file: include: ?inventory_items,view.lkml?

Answer: A
Section: (none)
Explanation
Explanation/Reference:

QUESTION 46
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} ;;
}
```

Answer: A
Section: (none)
Explanation
Explanation/Reference:

Reference: https://docs.looker.com/reference/field-params/dimension_group

Usage

```
view: view_name {
  dimension_group: field_name { … }
}
```

Hierarchy

View File

↳ view

↳ dimension_group

Accepts

A Looker identifier (to serve as the first part of the name for each dimension created by the dimension group)

Special Rules

- The type can be time or duration
- Typically used with a timeframes or intervals parameter
- Use a datatype parameter if the dimension

QUESTION 47

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};;

}
}
```


Answer: A,C
Section: (none)
Explanation
Explanation/Reference:

Reference: https://docs.looker.com/data-modeling/learning-lookml/advanced-lookml-concepts

QUESTION 48
A user reports that, when a date dimension is filtered to ?before now? results are returned that consistently include tomorrow. Dimension fill has been ruled out as a cause of the issue.
Which LookML parameter should be used to resolve this issue?

A. Week_start_day
B. Convert_tz
C. Datatype
D. Fiscal_month_offset

Answer: D
Section: (none)
Explanation
Explanation/Reference:

QUESTION 49
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.

Answer: A
Section: (none)
Explanation
Explanation/Reference:

QUESTION 50
Two developers are working on adding a new view to a project. Once both developers have finished their work in the view, the changes will be pushed to production.
Where should the developers write the LookML for this view?

A. In the master branch, with both users writing to the branch
B. In one user?s personal branch, with both users writing to the branch
C. In a new shared branch created from the master branch
D. In each of their personal branches, with each user writing code separately

Answer: C
Section: (none)
Explanation
Explanation/Reference:
Explanation/Reference: