

# Looker LookML Developer Practice Test (Sample)

## Study Guide



**Everything you need from our exam experts!**

**Copyright © 2025 by Examzify - A Kaluba Technologies Inc. product.**

**ALL RIGHTS RESERVED.**

**No part of this book may be reproduced or transferred in any form or by any means, graphic, electronic, or mechanical, including photocopying, recording, web distribution, taping, or by any information storage retrieval system, without the written permission of the author.**

**Notice: Examzify makes every reasonable effort to obtain from reliable sources accurate, complete, and timely information about this product.**

**SAMPLE**

## **Questions**

- 1. True or False: The max\_cache\_age parameter causes PDTs to rebuild.**
  - A. True**
  - B. False**
  - C. Conditionally true**
  - D. Not applicable**
- 2. True or False: You cannot add results from SQL Runner to a project.**
  - A. True**
  - B. False**
  - C. Rarely true**
  - D. Depends on the project setup**
- 3. True or False: An Explore can be used to create a query, then get the SQL command for that query for use in SQL Runner.**
  - A. True**
  - B. False**
  - C. Only for advanced users**
  - D. Scenario-dependent**
- 4. What happens if a datagroup is triggered?**
  - A. The derived table is deleted**
  - B. The query is re-run**
  - C. The persistence method changes**
  - D. The data is archived**
- 5. True or False: If your Looker admin has enabled the SQL Runner Vis Labs feature, you cannot create visualizations directly in SQL Runner.**
  - A. True**
  - B. False**
  - C. Depends on user privileges**
  - D. Not applicable**

6. How can you create a custom duration field for a created dimension group of type: time?
- A. Use standard SQL commands
  - B. Specify a ::string reference type
  - C. Specify a reference type for the dimensions in sql\_start and sql\_end
  - D. Define the duration in terms of weeks
7. True or False: The `${derived_table_or_view_name.SQL_TABLE_NAME}` syntax is not supported with a datagroup's sql\_trigger parameter.
- A. True
  - B. False
  - C. Only in certain contexts
  - D. Only for derived tables
8. What does a model file specify in Looker?
- A. A database connection and set of Explores
  - B. A detailed configuration for user permissions
  - C. A template for dashboard creation
  - D. A definition of SQL query limits
9. Which command is used to create a new LookML project in Looker?
- A. create project
  - B. new project
  - C. init project
  - D. add project
10. What is the purpose of a project manifest file in Looker?
- A. To manage user permissions
  - B. To define project-level settings and integrations
  - C. To create data models only
  - D. To compile LookML into reports

## **Answers**

SAMPLE

- 1. B**
- 2. B**
- 3. A**
- 4. B**
- 5. B**
- 6. C**
- 7. A**
- 8. A**
- 9. C**
- 10. B**

SAMPLE

## **Explanations**

SAMPLE



**1. True or False: The max\_cache\_age parameter causes PDTs to rebuild.**

**A. True**

**B. False**

**C. Conditionally true**

**D. Not applicable**

The max\_cache\_age parameter does not directly cause Persistent Derived Tables (PDTs) to rebuild. Instead, it is used to define the maximum age of cached results before they are considered stale. When a PDT is executed, if the cached results are still valid within the max\_cache\_age timeframe, the existing results are returned, and there is no need for a rebuild. If the cached results exceed the specified max\_cache\_age, then Looker will determine that a rebuild is necessary and will execute the underlying query to refresh the data. Understanding this distinction is crucial, as it highlights the role of max\_cache\_age in managing the freshness of data in PDTs without directly triggering their creation or rebuilding processes. This means that while the parameter can affect when a PDT is rebuilt based on the age of the cached data, the primary function is about cache management rather than an action that initiates a rebuild.

**2. True or False: You cannot add results from SQL Runner to a project.**

**A. True**

**B. False**

**C. Rarely true**

**D. Depends on the project setup**

The correct answer is that you can indeed add results from SQL Runner to a project, which makes the statement false. SQL Runner is a tool in Looker that allows users to execute raw SQL queries against their database and see the results immediately. This capability is especially useful for quick ad hoc analysis or checking data directly in the database. Once you have executed a query in SQL Runner and are satisfied with the results, you have the option to create a derived table or view that can then be incorporated into your Looker project. This allows the output from your SQL queries to be transformed into reusable components within LookML, such as views or explores, thereby enhancing your Looker project with valuable insights derived directly from SQL queries. This process of integrating SQL Runner results into a Looker project underscores its flexibility and the seamless way in which SQL can intersect with LookML development. By leveraging the results from SQL Runner, users can advance their data analysis capabilities and enrich the overall data modeling experience in Looker.

**3. True or False: An Explore can be used to create a query, then get the SQL command for that query for use in SQL Runner.**

**A. True**

**B. False**

**C. Only for advanced users**

**D. Scenario-dependent**

An Explore in Looker serves as a bridge between the user interface and the underlying data model, allowing users to create queries dynamically based on the dimensions and measures defined in the LookML. When a user builds a query through an Explore, they can generate the corresponding SQL that Looker would send to the database. This SQL can then be accessed and utilized in SQL Runner, which is a feature that allows users to run custom SQL queries directly against the database. The ability to obtain the SQL command from an Explore is particularly useful for users who need to understand how their selections translate into SQL or who want to take that SQL for further use or optimization outside of Looker. Thus, the statement is true, as Explorers are designed to facilitate this type of interaction with the data.

**4. What happens if a datagroup is triggered?**

**A. The derived table is deleted**

**B. The query is re-run**

**C. The persistence method changes**

**D. The data is archived**

When a datagroup is triggered, it initiates the re-evaluation of derived tables that depend on the datagroup. This means any query that utilizes a derived table associated with the particular datagroup will be re-run to ensure that the data is up to date. A datagroup is essentially used to manage the refresh logic of derived tables and helps in controlling when the underlying data is fetched again. It is important to understand that this mechanism helps maintain data accuracy and consistency in reports generated by Looker, as outdated data could lead to incorrect insights. By re-running the query upon triggering the datagroup, Looker ensures the information retrieved reflects the most current state of the data. The other options do not accurately describe the function of datagroups: derived tables are not deleted, persistence methods remain the same unless explicitly changed, and data is not archived as part of the datagroup trigger process.

**5. True or False: If your Looker admin has enabled the SQL Runner Vis Labs feature, you cannot create visualizations directly in SQL Runner.**

**A. True**

**B. False**

**C. Depends on user privileges**

**D. Not applicable**

The assertion that you cannot create visualizations directly in SQL Runner, even with the SQL Runner Vis Labs feature enabled, is indeed false. When the SQL Runner Vis Labs feature is turned on, it allows users to create and explore visualizations directly from their SQL queries. This means that users can leverage the power of SQL to extract data and simultaneously visualize that data within the SQL Runner interface. The ability to create visualizations facilitates a more interactive data analysis experience, allowing users to see the results of their SQL queries in various formats, such as charts and graphs. This capability enhances the overall utility of SQL Runner, making it a more powerful tool for analysts who want to quickly visualize their query results without having to move that data into another application or tool for visualization. Therefore, the correct answer highlights the ability to create visualizations within SQL Runner, thus dispelling any misconceptions regarding its functionality with the Vis Labs feature enabled.

**6. How can you create a custom duration field for a created dimension group of type: time?**

**A. Use standard SQL commands**

**B. Specify a ::string reference type**

**C. Specify a reference type for the dimensions in sql\_start and sql\_end**

**D. Define the duration in terms of weeks**

To create a custom duration field for a created dimension group of type time, specifying a reference type for the dimensions in sql\_start and sql\_end is the most appropriate approach. By defining these two dimensions, you establish the beginning and ending points of the time frame you want to measure the duration for. This allows Looker to calculate the difference between the two timestamps effectively. When you set the reference types correctly for both sql\_start and sql\_end, Looker understands how to compute the duration by comparing these timestamps. This mechanism is essential for creating accurate time-based calculations, enabling users to derive insights related to time metrics, such as the length of an event or the duration between two dates. In contrast, other approaches like using standard SQL commands may not integrate seamlessly within Looker's LookML structure. Similarly, specifying a ::string reference type is not suitable for time calculations, as it would not allow Looker to perform the necessary date arithmetic. Defining the duration in terms of weeks, while potentially useful in specific contexts, does not encompass the broader and more flexible requirement of specifying the start and end references necessary for general duration calculations. Hence, focusing on sql\_start and sql\_end with appropriate reference types is key to achieving a precise and functional custom duration field.

**7. True or False: The `${derived_table_or_view_name.SQL_TABLE_NAME}` syntax is not supported with a datagroup's `sql_trigger` parameter.**

**A. True**

**B. False**

**C. Only in certain contexts**

**D. Only for derived tables**

The statement is true because the `${derived_table_or_view_name.SQL_TABLE_NAME}` syntax, which is used to reference the underlying SQL table or view name of a derived table, is indeed not supported in the context of a datagroup's `sql_trigger` parameter. Datagroups are designed to manage data freshness and caching by tracking when data is updated. They typically rely on simpler SQL expressions or raw table names rather than the dynamic references that are used with derived tables. Since the `sql_trigger` parameter requires a specific and static reference, using a syntax that pulls in dynamic context does not work in this scenario. This reinforces the need to understand the limitations of using derived table specifics when setting up datagroups in LookML.

**8. What does a model file specify in Looker?**

**A. A database connection and set of Explores**

**B. A detailed configuration for user permissions**

**C. A template for dashboard creation**

**D. A definition of SQL query limits**

A model file in Looker plays a crucial role in defining how data is structured and accessed. It specifies the database connection, which allows Looker to know where to pull the data from. This connection is essential for establishing the relationship between Looker and the underlying database. In addition to defining the connection, the model file includes a set of Explores. Explores are a key component in Looker, enabling users to query the data easily and access various dimensions and measures defined in the model. By organizing the data into Explores, users can interact with it more intuitively, generating reports and analyses based on the established relationships and logic in the model. This answer stands out as the most accurate reflection of what a model file encompasses within Looker, differentiating it from aspects such as user permissions, dashboard templates, or SQL query limits, which are handled in other configurations or structural files within Looker.

**9. Which command is used to create a new LookML project in Looker?**

- A. create project**
- B. new project**
- C. init project**
- D. add project**

The command used to create a new LookML project in Looker is "init project." This command is essential for setting up a new project environment, allowing developers to specify the project's name and establish the base structure needed for LookML development. By using "init project," Looker's interface prepares the foundational components for managing data models, views, and explores related to the project. Using "init project" ensures that all required configurations and settings are properly initialized, which is critical for a smooth workflow in developing and deploying LookML models. This sets the stage for building a data exploration interface that aligns with the data requirements of your organization.

**10. What is the purpose of a project manifest file in Looker?**

- A. To manage user permissions**
- B. To define project-level settings and integrations**
- C. To create data models only**
- D. To compile LookML into reports**

The project manifest file in Looker serves a crucial role in defining project-level settings and integrations. This file, typically named `manifest.lkml`, outlines various configurations that govern how the Looker project operates. It includes settings for the project's versioning, integration with other systems, and configurations for LookML modeling and extensions. Using the manifest file allows teams to manage and maintain their Looker development projects more effectively. By specifying integration points and project settings, the manifest ensures that developers can work consistently while collaborating on models and dashboards. This organizational structure is essential for scaling Looker projects, particularly in environments where multiple stakeholders are involved. In contrast, managing user permissions is handled separately through user roles and permissions settings. Creating data models is a function of the LookML files themselves, not the manifest. Compiling LookML into reports is part of Looker's backend processing but is not directly related to the purpose of the project manifest file. Thus, the project manifest file's prime objective is centered on project-level settings and integrations, which is why defining those attributes correctly is so vital for effective Looker project management.