

DBT LABS ANALYTICS ENGINEER CERTIFICATION PRACTICE TEST (SAMPLE) STUDY GUIDE



Prepare with structure. Pass with confidence



Copyright © 2026 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 accurate, complete, and timely information about this product from reliable sources.

SAMPLE

Table of Contents

Copyright	1
Table of Contents	2
Introduction	3
How to Use This Guide	4
Questions	5
Answers	8
Explanations	10
Next Steps	15

SAMPLE

Introduction

Preparing for a certification exam can feel overwhelming, but with the right tools, it becomes an opportunity to build confidence, sharpen your skills, and move one step closer to your goals. At Examzify, we believe that effective exam preparation isn't just about memorization, it's about understanding the material, identifying knowledge gaps, and building the test-taking strategies that lead to success.

This guide was designed to help you do exactly that.

Whether you're preparing for a licensing exam, professional certification, or entry-level qualification, this book offers structured practice to reinforce key concepts. You'll find a wide range of multiple-choice questions, each followed by clear explanations to help you understand not just the right answer, but why it's correct.

The content in this guide is based on real-world exam objectives and aligned with the types of questions and topics commonly found on official tests. It's ideal for learners who want to:

- Practice answering questions under realistic conditions,
- Improve accuracy and speed,
- Review explanations to strengthen weak areas, and
- Approach the exam with greater confidence.

We recommend using this book not as a stand-alone study tool, but alongside other resources like flashcards, textbooks, or hands-on training. For best results, we recommend working through each question, reflecting on the explanation provided, and revisiting the topics that challenge you most.

Remember: successful test preparation isn't about getting every question right the first time, it's about learning from your mistakes and improving over time. Stay focused, trust the process, and know that every page you turn brings you closer to success.

Let's begin.

How to Use This Guide

This guide is designed to help you study more effectively and approach your exam with confidence. Whether you're reviewing for the first time or doing a final refresh, here's how to get the most out of your Examzify study guide:

1. Start with a Diagnostic Review

Skim through the questions to get a sense of what you know and what you need to focus on. Your goal is to identify knowledge gaps early.

2. Study in Short, Focused Sessions

Break your study time into manageable blocks (e.g. 30 – 45 minutes). Review a handful of questions, reflect on the explanations.

3. Learn from the Explanations

After answering a question, always read the explanation, even if you got it right. It reinforces key points, corrects misunderstandings, and teaches subtle distinctions between similar answers.

4. Track Your Progress

Use bookmarks or notes (if reading digitally) to mark difficult questions. Revisit these regularly and track improvements over time.

5. Simulate the Real Exam

Once you're comfortable, try taking a full set of questions without pausing. Set a timer and simulate test-day conditions to build confidence and time management skills.

6. Repeat and Review

Don't just study once, repetition builds retention. Re-attempt questions after a few days and revisit explanations to reinforce learning. Pair this guide with other Examzify tools like flashcards, and digital practice tests to strengthen your preparation across formats.

There's no single right way to study, but consistent, thoughtful effort always wins. Use this guide flexibly, adapt the tips above to fit your pace and learning style. You've got this!

Questions

SAMPLE

1. What does the `schema.yml` file do in a dbt project?

- A. It stores raw data before transformation
- B. It defines metadata for models and documentation
- C. It holds user access permissions for projects
- D. It specifies data sources for database connections

2. What is a 'dbt artifact'?

- A. A platform for creating dbt dashboards
- B. A repository for storing dbt packages or dependencies for versioning
- C. A section of the dbt documentation
- D. A command for optimizing dbt projects

3. What are custom tests in dbt?

- A. Pre-built tests provided by dbt
- B. User-defined checks for data integrity
- C. Tests that run automatically without user input
- D. Comparative tests between different models

4. How can you specify a materialization strategy in a dbt model?

- A. By adding the strategy in the models directory
- B. Through the config block in a model file
- C. By adjusting the dbt project's YAML file
- D. Through the dbt profile settings

5. How do you enforce data freshness in dbt?

- A. By scheduling regular runs in a production environment
- B. By implementing tests that verify timestamps and other freshness metrics
- C. By archiving old data regularly
- D. By creating new tables for fresh data daily

6. Which command is used to build your dbt project?

- A. dbt start
- B. dbt build
- C. dbt develop
- D. dbt run

7. What happens when you set --threads to a number greater than your database allows?

- A. All threads will be ignored
- B. Only the allowed number of threads will run
- C. Execution will fail immediately
- D. Performance will improve beyond limits

8. What is the primary function of the 'dbt compile' command?

- A. To run all models in the project
- B. To generate compiled SQL files without executing them
- C. To delete unnecessary files in the project
- D. To create new dbt models

9. What is a 'dbt profile'?

- A. A configuration file that defines how dbt connects to a target database
- B. A type of database schema used in dbt
- C. A documentation file format used in dbt projects
- D. A visualization tool for dbt outputs

10. What is the primary function of singular tests in dbt?

- A. To assess the freshness of sources
- B. To define accepted values
- C. To write SQL that returns failing records
- D. To validate data types across models

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. What does the `schema.yml` file do in a dbt project?

- A. It stores raw data before transformation
- B. It defines metadata for models and documentation**
- C. It holds user access permissions for projects
- D. It specifies data sources for database connections

The `schema.yml` file in a dbt project plays a crucial role in defining metadata for models and documentation. This file is used to organize and describe the various models within the project, including information such as descriptions, column-level documentation, tests, and relationships between models. By providing this structure, it enhances the readability and usability of the data models, making it easier for team members and stakeholders to understand their purpose and usage. Furthermore, the schema file allows analytics engineers to add tests that validate data integrity and quality, ensuring that the models produce reliable outputs. Documentation generated from the `schema.yml` can be leveraged to create a clear and comprehensive understanding of the data assets, which is invaluable for knowledge sharing and onboarding processes. In contrast, the other choices don't accurately represent the function of the `schema.yml` file. For example, while user access permissions are important in database management, they are not defined in the `schema.yml`. Similarly, raw data storage and data source specifications are handled through different configurations within dbt and do not belong in the schema file.

2. What is a 'dbt artifact'?

- A. A platform for creating dbt dashboards
- B. A repository for storing dbt packages or dependencies for versioning**
- C. A section of the dbt documentation
- D. A command for optimizing dbt projects

A 'dbt artifact' serves as a repository for storing dbt packages or dependencies, aiding in versioning. This is crucial for managing the various components that a dbt project may rely upon, allowing teams to maintain consistency and traceability in their analytics workflows. By utilizing an artifact, users benefit from a structured approach to handle package versions, enabling them to easily update or revert to previous package states as needed. This enhances collaboration across teams and ensures that everyone is using the same dependencies, which is vital for reproducibility in analytics projects. The alternative options describe other aspects of dbt or analytics workflows but do not accurately represent the purpose of a dbt artifact.

3. What are custom tests in dbt?

- A. Pre-built tests provided by dbt
- B. User-defined checks for data integrity**
- C. Tests that run automatically without user input
- D. Comparative tests between different models

Custom tests in dbt refer to user-defined checks for data integrity. These tests allow analysts and engineers to create specific assertions to ensure that the data meets certain conditions or business rules. By defining custom tests, users can validate data consistency, accuracy, and completeness tailored to their specific requirements, which enhances trust in the data being analyzed. For instance, a custom test might check that no values in a particular column are null, or that certain relationships between data points hold true across datasets. This flexibility enables users to implement checks that can capture unique scenarios relevant to their business logic or data model. The other options do not accurately capture the essence of custom tests. Pre-built tests may serve as a foundation but do not offer the customization feature that allows users to tailor their checks. Tests that run automatically could imply a lack of user-defined parameters, which is contrary to the nature of custom tests that rely on specific, user-defined criteria. Lastly, comparative tests typically involve analyzing relationships between models, which is not the primary focus of what custom tests represent in dbt.

4. How can you specify a materialization strategy in a dbt model?

- A. By adding the strategy in the models directory
- B. Through the config block in a model file**
- C. By adjusting the dbt project's YAML file
- D. Through the dbt profile settings

To specify a materialization strategy in a dbt model, it is done through the config block in a model file. In dbt, each model can have its own configuration settings, which include the materialization strategy. This is achieved by using the `{{ config() }}` function within the model file, where you can set parameters such as `materialized`, defining whether the model should be a table, view, incremental, or ephemeral. This approach provides granular control at the model level, allowing you to define how each specific model should be built and behave. It also enhances the clarity and maintainability of your project, as all configurations related to a model are neatly encapsulated within the model file itself. Other methods mentioned involve broader configuration settings or different contexts, which do not apply directly to an individual model's materialization. The models directory is merely the location of model files and does not dictate specific behaviors. The dbt project's YAML file is used for project-wide configurations but does not customize individual model materializations. dbt profile settings are concerned with connection settings for your data warehouse and do not influence how models are materialized.

5. How do you enforce data freshness in dbt?

- A. By scheduling regular runs in a production environment
- B. By implementing tests that verify timestamps and other freshness metrics**
- C. By archiving old data regularly
- D. By creating new tables for fresh data daily

Enforcing data freshness in dbt primarily involves implementing tests that verify timestamps and other freshness metrics. This practice is crucial because it ensures that the data being used in analyses is current and accurately reflects the most recent state of the source data. With freshness tests, you can define specific criteria that dictate how old the data can be before it is considered stale. These tests check for the recency of timestamps or other indicators of data age, allowing you to quickly identify and address any issues with outdated information. When discrepancies arise, these tests can trigger notifications or alert data teams to investigate, ensuring that the analytics produced from the dbt models are based on reliable and timely data. In contrast, while scheduling regular runs in a production environment is a common practice in dbt, it does not inherently measure or enforce data quality regarding freshness. Regular runs are important for workflow efficiency but do not provide mechanisms to ensure that the data meets freshness criteria. Archiving old data regularly addresses data management concerns but does not directly relate to the freshness of the data being analyzed. This practice is more about data lifecycle management rather than ensuring that active datasets are up-to-date. Creating new tables for fresh data daily may contribute to ensuring that fresh datasets are available, but this approach could lead to

6. Which command is used to build your dbt project?

- A. dbt start
- B. dbt build
- C. dbt develop
- D. dbt run**

The command used to build your dbt project is "dbt build." This command encompasses a series of steps that not only run your models but also handle tests, snapshots, and seeds in one action. When you execute this command, it ensures that the models specified in your project are built according to the logic defined in their SQL files and also confirms that these models meet any defined tests. While "dbt run" does execute the SQL models defined in your project, it does not include the additional capabilities that "dbt build" offers, such as executing tests and processing snapshots and seeds. Using "dbt build" is essential for a comprehensive approach to building your project, as it allows you to confirm the integrity and quality of your data as it flows through your transformations. Thus, "dbt build" is the correct command to fully prepare and deploy your dbt project, ensuring that all aspects of the transformation process are accounted for in one go.

7. What happens when you set --threads to a number greater than your database allows?

- A. All threads will be ignored
- B. Only the allowed number of threads will run**
- C. Execution will fail immediately
- D. Performance will improve beyond limits

Setting the --threads option to a number greater than what your database allows means that only the maximum number of threads permitted by the database will be utilized during execution. The system effectively caps the number of concurrent processes to adhere to the limits set by the database settings. This ensures that the database does not become overwhelmed and can handle the queries efficiently within its operational constraints. The other options suggest unrealistic scenarios. Ignoring all threads would render the command ineffective, while immediate execution failure implies a lack of any possible threading operation. The idea of improved performance beyond the database's limits contradicts the fundamental principles of concurrent processing, where exceeding capacity can lead to resource contention rather than enhanced throughput. Therefore, only the allowed number of threads will run, aligning with the database's configured limits.

8. What is the primary function of the 'dbt compile' command?

- A. To run all models in the project
- B. To generate compiled SQL files without executing them**
- C. To delete unnecessary files in the project
- D. To create new dbt models

The primary function of the 'dbt compile' command is to generate compiled SQL files without executing them. When this command is executed, dbt processes the project's models, macros, and configurations, converting them into SQL statements that can later be run against the database. This allows users to review the generated SQL to verify that it matches their expectations before actually executing any operations against the database. This step is particularly crucial during the development process, as it helps identify syntax errors or logical problems in SQL generation without the risk of running potentially faulty SQL against a live database. The compiled SQL files are not executed; they are simply created, enabling a clear understanding of what will happen if the models are run. Focusing on this function helps practitioners ensure that their dbt models are correctly configured and ready for execution when they choose to do so.

9. What is a 'dbt profile'?

- A. A configuration file that defines how dbt connects to a target database**
- B. A type of database schema used in dbt
- C. A documentation file format used in dbt projects
- D. A visualization tool for dbt outputs

A 'dbt profile' is a configuration file that is essential for establishing the connection between dbt and the target database. It contains all the necessary information, such as the database credentials, connection details, and specific configurations needed to facilitate dbt's interaction with the database environment where transformations will take place. This setup allows dbt to understand where to send requests and pull or push data, making it a fundamental component of any dbt project. Other options, while relevant to dbt in different contexts, do not accurately describe the primary function of a dbt profile. The concept of database schemas or types is related to how data is organized within the database rather than connection specifics. Documentation formats serve to clarify and elaborate on the project's elements, but they do not pertain to connection configurations. Visualization tools may be utilized for analyzing or displaying outputs from dbt runs, but they do not involve the connection settings necessary for dbt to function correctly.

10. What is the primary function of singular tests in dbt?

- A. To assess the freshness of sources
- B. To define accepted values
- C. To write SQL that returns failing records**
- D. To validate data types across models

The primary function of singular tests in dbt is to write SQL that returns failing records. Singular tests are designed to validate a specific condition or set of conditions in your data. When a singular test is executed, it runs the SQL logic you've defined, and if that logic identifies any records that do not meet the specified criteria, those records are returned as the failing cases. This type of testing allows you to enforce quality checks on your data. For example, you can use singular tests to ensure that certain fields are not null, or that the values in a column fall within a specified range. The key aspect is that the SQL you write must clearly reflect the conditions you want to validate, and failing records will be those that do not conform to those conditions. In contrast, the other options refer to different functionalities within dbt's testing framework. Assessing the freshness of sources relates to ensuring that data inputs are up to date. Defining accepted values pertains to constraints or enumerations that limit what can be entered into a field. Validating data types across models typically involves checks on the structure and type consistency of data, which is not the primary role of singular tests. Therefore, the focus of singular tests specifically on querying for failing records sets it apart.

Next Steps

Congratulations on reaching the final section of this guide. You've taken a meaningful step toward passing your certification exam and advancing your career.

As you continue preparing, remember that consistent practice, review, and self-reflection are key to success. Make time to revisit difficult topics, simulate exam conditions, and track your progress along the way.

If you need help, have suggestions, or want to share feedback, we'd love to hear from you. Reach out to our team at hello@examzify.com.

Or visit your dedicated course page for more study tools and resources:

<https://dbtlabsanalyticsengr.examzify.com>

We wish you the very best on your exam journey. You've got this!

SAMPLE