dbt Analytics Engineering Certification Exam Study Guide

dbt Labs

How to use this study guide

This is the official study guide for the **dbt Analytics Engineering Certification Exam** from the team at dbt Labs. While the guide suggests a sequence of courses and reading material, we recommend using it to supplement (rather than substitute) real-world use and experience with dbt.

The **exam overview** will provide high-level details on the format of the exam. We recommend being mindful of the number of questions and time constraints.

The **topic outline** will provide a clear list of the topics that are assessed on the exam. dbt subject matter experts used this topic outline to write and meticulously review all of the exam items that you will find on the exam.

The **sample exam questions** provide examples of some of the formats to expect for the exam. The types of questions you can expect on the exam include

- Multiple-choice
- Fill-in-the-blank
- Matching
- Hotspot
- Build list
- Discrete Option Multiple Choice (DOMC)

The **learning path** will walk you through a suggested series of courses, readings, and documentation. We will also provide some guidance for the types of experience to build while using dbt on a real-world project.

Finally, the **additional resources** section will provide additional places to continue your learning.

—Amy Chen, Kyle Coapman, and the dbt Labs Team

Exam Overview

The **dbt Analytics Engineering Certification Exam** is designed to evaluate your ability to

- build, test, and maintain models to make data accessible to others
- use dbt to apply engineering principles to analytics infrastructure

We recommend that you have at least SQL proficiency and have had 6+ months of experience working in dbt (Core or Cloud) before attempting the exam.

Logistics

Please see **the exam website** for details on scheduling, retake policy, and other exam policies.

Topic Outline

The **dbt Analytics Engineering Certification Exam** has been designed to assess the following topics and sub-topics.

Topic 1: Developing dbt models

- Identifying and verifying any raw object dependencies
- Understanding core dbt materializations
- Conceptualizing modularity and how to incorporate DRY principles
- Converting business logic into performant SQL queries
- Using commands such as run, test, docs and seed
- Creating a logical flow of models and building clean DAGs
- Defining configurations in dbt_project.yml
- Configuring sources in dbt
- Using dbt Packages

Topic 2: Debugging data modeling errors

- Understanding logged error messages
- Troubleshooting using compiled code
- Troubleshooting .yml compilation errors
- Distinguishing between a pure SQL and a dbt issue that presents itself as a SQL issue
- Developing and implementing a fix and testing it prior to merging

Topic 3: Monitoring data pipelines

- Understanding and testing the warehouse-level implications of a model run failing at different points in the DAG
- Understanding the general landscape of tooling

Topic 4: Implementing dbt tests

- Using generic, singular and custom tests on a wide variety of models and sources
- Understanding assumptions specific to the datasets being generated in models and to the raw data in the warehouse
- Implementing various testing steps in the workflow
- Ensuring data is being piped into the warehouse and validating accuracy against baselines

Topic 5: Deploying dbt jobs

- Understanding the differences between deployment and development environments
- Configuring development and deployment environments
- Configuring the appropriate tasks, settings and triggers for the job
- Understanding how a dbt job utilizes an environment in order to build database objects and artifacts
- Using dbt commands to execute specific models

Topic 6: Creating and Maintaining dbt documentation

- Updating dbt docs
- Implementing source, table, and column descriptions in .yml files
- Using dbt commands to generate a documentation site
- Using macros to show model and data lineage on the DAG

Topic 7: Promoting code through version control

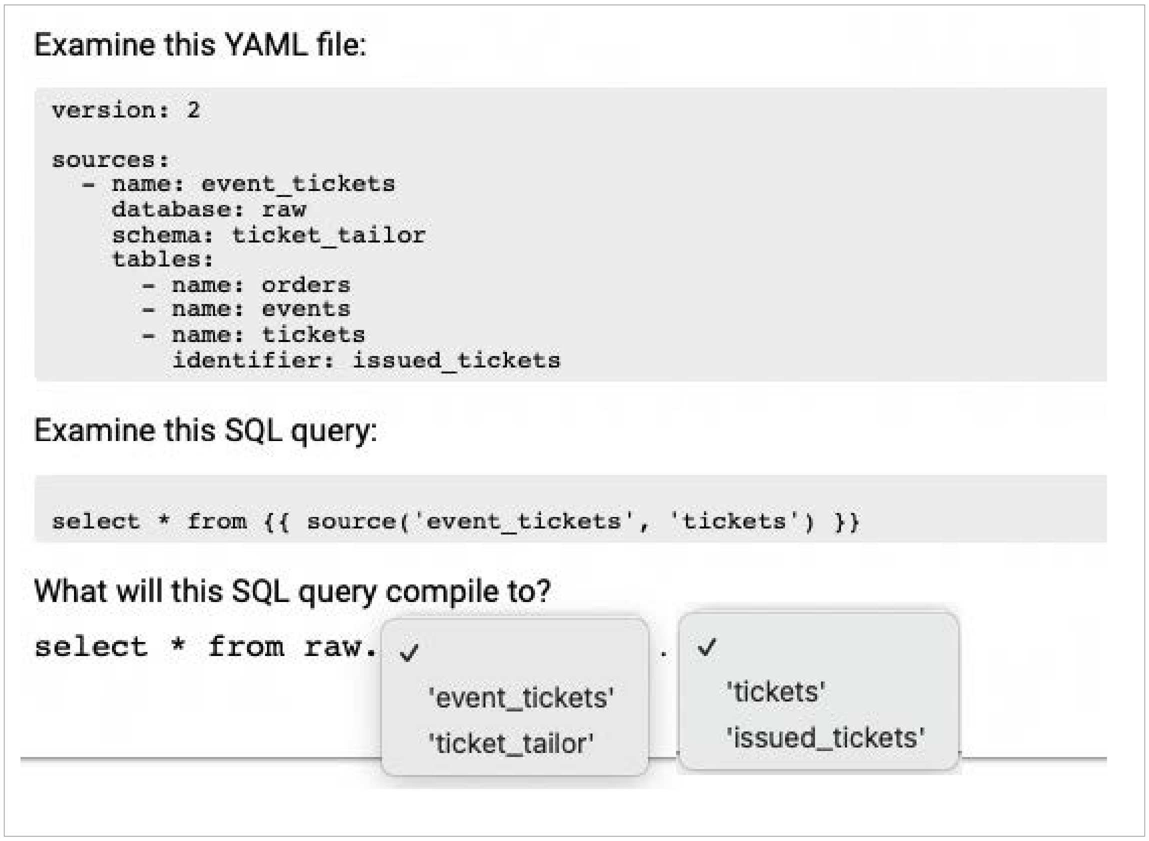
- Understanding concepts and working with Git branches and functionalities
- Creating clean commits and pull requests
- Merging code to the main branch

Topic 8: Establishing environments in data warehouse for dbt

- Understanding environment’s connections
- Understanding the differences between production data, development data, and raw data

Sample exam questions

Sample Exam Question 1



Examine this YAML file:

```
version: 2

sources:
  - name: event_tickets
    database: raw
    schema: ticket_tailor
    tables:
      - name: orders
      - name: events
      - name: tickets
        identifier: issued_tickets
```

Examine this SQL query:

```
select * from {{ source('event_tickets', 'tickets') }}
```

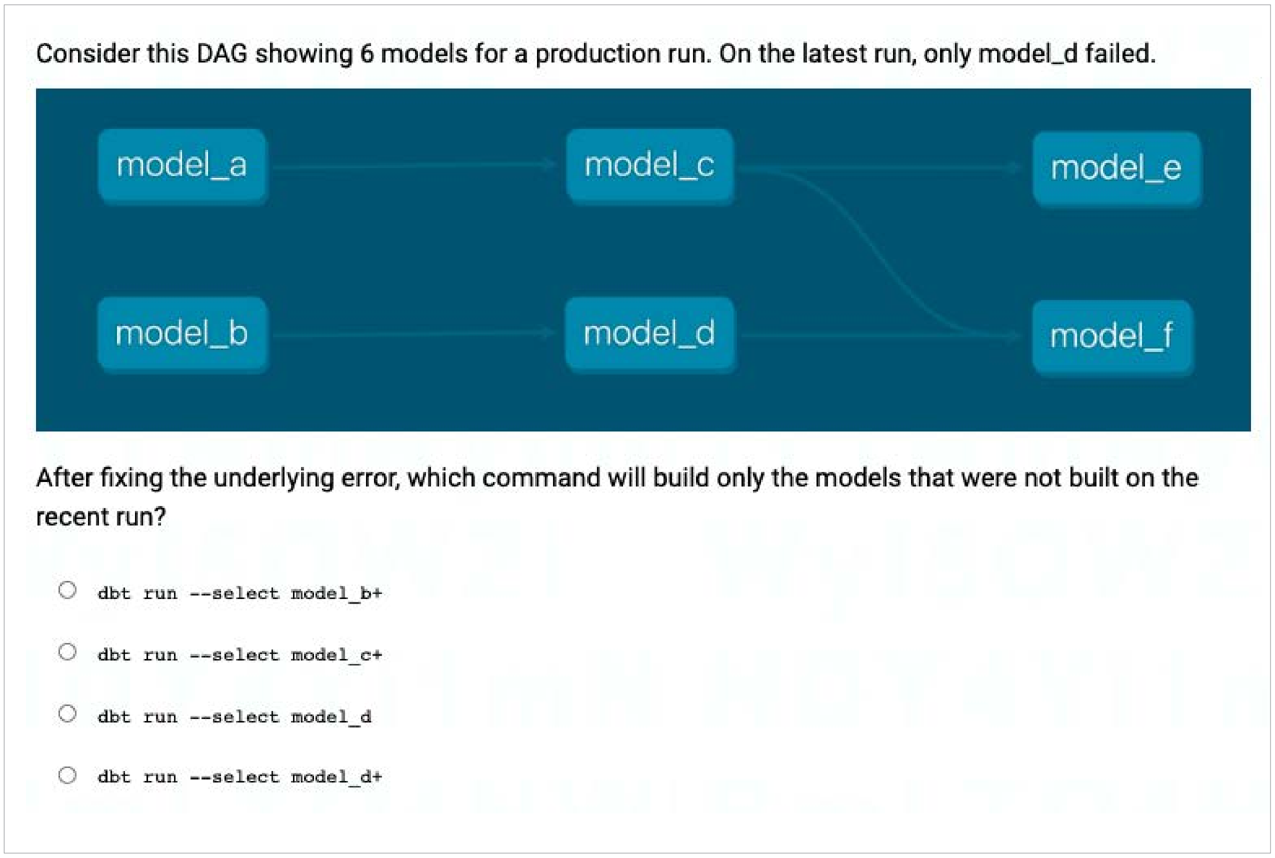
What will this SQL query compile to?

select * from raw. ✓ 'event_tickets' / 'ticket_tailor' . ✓ 'tickets' / 'issued_tickets'

Explanation: In this question, you are presented with a YAML file showing a source definition. The answer choices test your knowledge of source configurations. In the first drop-down, you're presented with event_ticket and ticket_tailor as options. While dbt does use the source name: field by default, if you define an explicit schema in the file, it will use that value instead. That is why ticket_tailor is the correct answer.

In the second drop-down, you're presented with tickets and issued_tickets as possible answer choices. The identifier parameter lets you specify the name of the table as it is named in the database. That is why issued_tickets is the correct choice here.

Sample Exam Question 2



Consider this DAG showing 6 models for a production run. On the latest run, only model_d failed.

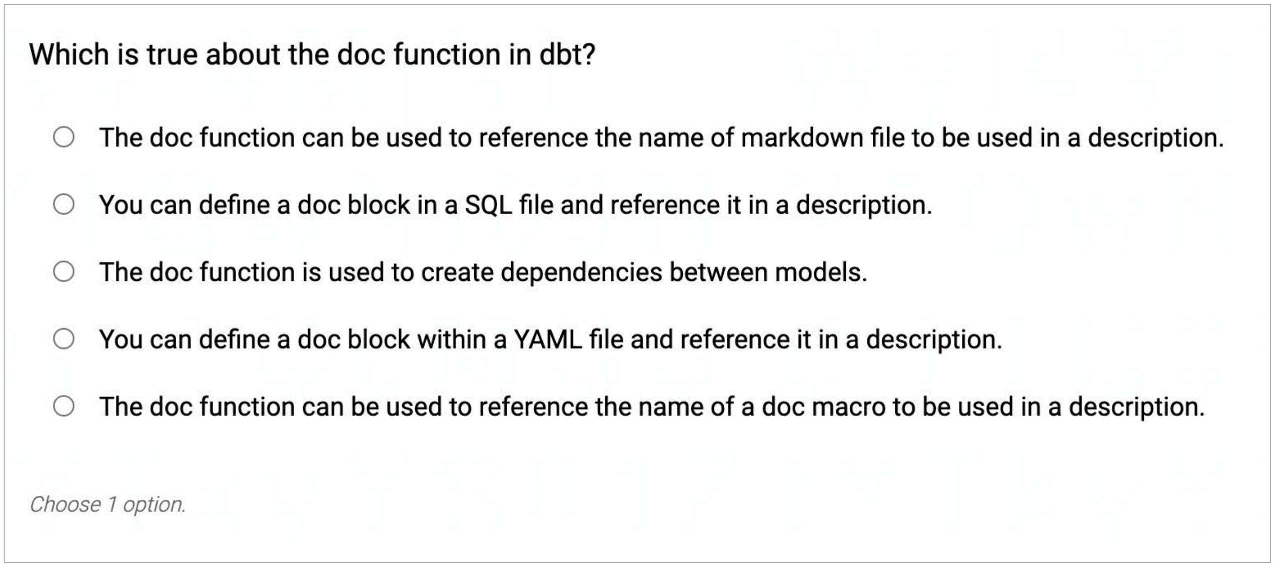
$$\begin{tikzcd}
\text{model\_a} \arrow[r] & \text{model\_c} \arrow[r] \arrow[dr] & \text{model\_e} \\
\text{model\_b} \arrow[r] & \text{model\_d} \arrow[r] & \text{model\_f}
\end{tikzcd}$$

After fixing the underlying error, which command will build only the models that were not built on the recent run?

```
○ dbt run --select model_b+

○ dbt run --select model_c+

○ dbt run --select model_d

○ dbt run --select model_d+
```

Explanation: `dbt run --select model_d+` is the correct answer because model_d was the model that failed in the previous run *and* its downstream models would have been skipped previously. By selecting only `model_d` and its downstream dependencies (children), we are able to rebuild only the models that failed in the last run.

Sample Exam Question 3



Which is true about the doc function in dbt?

- ○ The doc function can be used to reference the name of markdown file to be used in a description.
- ○ You can define a doc block in a SQL file and reference it in a description.
- ○ The doc function is used to create dependencies between models.
- ○ You can define a doc block within a YAML file and reference it in a description.
- ○ The doc function can be used to reference the name of a doc macro to be used in a description.

Choose 1 option.

Explanation: When referencing a doc block in a YAML file for the description of a model, it is important to reference the name of the actual macro rather than the name of the markdown file.

So the correct answer is “The doc function can be used to reference the name of a doc macro to be used in a description.”

Learning Path:

This is not the **only** way to prepare for the exam, but just one recommended path for someone new to dbt to prepare for the exam. Each checkpoint provides a logical set of courses to complete, readings and documentation to digest, and real-world experience to seek out. We recommend this order, but things can be reorganized based on your learning preferences.

Checkpoint 0 - Prerequisites

dbt is a tool that brings together several different technical skills in one place. We recommend starting this path after you've developed foundational git and SQL skills.

For SQL, the exam expects familiarity with **joins, aggregations, common table expressions (CTEs), and window functions.**

For git, the exam expects familiarity with **branching strategies (including development vs main branches), basic git commands, and pull requests.**

Checkpoint 1 - Build a Foundation

Resources:

- Courses:
  - dbt Fundamentals
- Readings:
  - dbt viewpoint
- Documentation:
  - Source properties
  - Node selection syntax
  - dbt_project.yml
  - General resource properties
- Experience
  - Creating a dbt project from scratch to deployment
  - Debugging errors
- Commands
  - `dbt compile`
  - `dbt run`
  - `dbt source freshness`
  - `dbt test`
  - `dbt docs generate`
  - `dbt build`
  - `dbt run-operation`

Checkpoint 2 - Modularity and Refactoring

Resources:

- Courses:
  - Refactoring SQL for Modularity
- Readings:
  - How we structure our dbt projects
  - Your Essential dbt Project Checklist
- Documentation:
  - Refactoring legacy SQL to dbt
- Experience:
  - Refactoring SQL for performance and clarity

Checkpoint 3 - Doing More with dbt

- Courses:
  - Jinja, Macros, and Packages
  - Advanced Materializations
  - Analyses and Seeds
- Documentation:
  - Exposures
  - Env_var
  - Target
  - Schema
  - Database
- Experience
  - Utilizing packages and macros in a dbt project
  - Implementing all core materializations into a dbt project
  - Implementing seeds
- Commands:
  - dbt snapshot
  - dbt seed

Checkpoint 4 - Deployment and Testing

Resources:

- Courses:
    - Advanced Testing
    - Advanced Deployment
- Readings:
    - The exact grant statements we use in a dbt project
    - The exact GitHub pull request template we use at dbt Labs
    - How to review an analytics pull request
    - How we configure Snowflake
- Documentation:
    - Tags
    - Hooks & Operations
    - Custom Schema
    - Threads
- Experience
    - Defining environments in your data platform
    - Defining environments in dbt
    - Promoting code through git including use of multiple branches, pull requests
    - Troubleshooting errors in production runs
    - Defining dbt jobs for optimal performance

Additional Resources:

- dbt Slack
    - #dbt-certification
    - #learn-on-demand
    - #advice-dbt-for-beginners
    - #advice-dbt-for-power-users
    - #dbt-deployment-and-orchestration

Want to use email? Contact certification@dbtlabs.com