

# dbt Labs dbt-Analytics-Engineering PDF Demo & dbt-Analytics-Engineering Testing Engine



Die Schulungsunterlagen zur dbt Labs dbt-Analytics-Engineering Zertifizierungsprüfung von ZertSoft werden die größte Erfolgsquote erzielen. Neben den Büchern sind heutzutage das Internet als ein Wissensschatz angesehen. In ZertSoft können Sie Ihren Wissensschatz finden. Das ist eine Website, die Ihnen sehr helfen können. Sie werden sicher komplizierte Übungen treffen, Unser ZertSoft wird Ihnen helfen, die Prüfung ganz einfach zu bestehen, weil es alle erforderlichen Kenntnisse zur dbt Labs dbt-Analytics-Engineering Zertifizierungsprüfung enthält.

Wollen Sie den Plan machen, dass Sie dbt Labs dbt-Analytics-Engineering Zertifizierungsprüfung ablegen, um Ihre Fähigkeit zu entwickeln. Wenn Sie dbt Labs dbt-Analytics-Engineering Prüfung ablegen, ob Sie die geeigneten Lernhilfe finden? Und welche Unterlage sind wertvoll? Haben Sie dbt Labs dbt-Analytics-Engineering Dumps gewählt? Wenn ja, sorgen Sie sich bitte nicht um den Misserfolg.

>> dbt Labs dbt-Analytics-Engineering PDF Demo <<

## dbt-Analytics-Engineering Übungsmaterialien - dbt-Analytics-Engineering Lernressourcen & dbt-Analytics-Engineering Prüfungsfragen

Wir ZertSoft sind eine professionelle Website. Wir bieten jedem Teilnehmer guten Service, sowie Vor-Sales-Service und Nach-Sales-Service. Wenn Sie dbt Labs dbt-Analytics-Engineering Zertifizierungsunterlagen von ZertSoft wollen, können Sie zuerst das kostenlose Demo benutzen. Sie können sich fühlen, ob die Unterlagen sehr geeignet sind. Damit können Sie die Qualität unserer dbt Labs dbt-Analytics-Engineering Prüfungsunterlagen überprüfen und dann sich entscheiden für den Kauf. Falls Sie dabei durchgefallen

wären, geben wir Ihnen voll Geld zurück. Oder Sie können wieder einjährige kostenlose Aktualisierung auswählen.

## dbt Labs dbt Analytics Engineering Certification Exam dbt-Analytics-Engineering Prüfungsfragen mit Lösungen (Q40-Q45):

### 40. Frage

You run the command:

```
dbt test --select 'test_type:singular'
```

What will the command run?

Options shown:

- A. furniture\_customers\_test  
stg\_tpch\_orders\_assert\_positive\_price  
Choose 1 option.
- B. macro\_stg\_tpch\_orders\_assert\_pos\_price  
macro\_stg\_tpch\_suppliers\_assert\_pos\_acct\_bal  
stg\_tpch\_orders\_assert\_positive\_price
- C. furniture\_customers\_test  
macro\_stg\_tpch\_orders\_assert\_pos\_price  
macro\_stg\_tpch\_suppliers\_assert\_pos\_acct\_bal  
stg\_tpch\_orders\_assert\_positive\_price
- D. furniture\_customers\_test

**Antwort: B**

Begründung:

In dbt, singular tests are custom SQL tests that live as standalone .sql files inside the root /tests directory, not inside /tests/generic. A singular test returns rows that indicate failure, and dbt runs the SQL directly as written. Generic tests, on the other hand, live inside the /tests/generic folder and are YAML-based macro- driven tests.

From your test folder structure (shown in the original screenshot), the following SQL files exist:

Inside /tests/generic/

\* furniture\_customers\_test.sql This file is a generic test, not singular.

Inside the root /tests directory:

\* macro\_stg\_tpch\_orders\_assert\_pos\_price.sql

\* macro\_stg\_tpch\_suppliers\_assert\_pos\_acct\_bal.sql

\* stg\_tpch\_orders\_assert\_positive\_price.sql

These are singular tests, because they are standalone SQL files in the tests root folder.

When you run:

```
dbt test --select 'test_type:singular'
```

dbt filters for only tests classified as singular. It ignores generic tests entirely.

Therefore, only the following 3 tests will run:

\* macro\_stg\_tpch\_orders\_assert\_pos\_price

\* macro\_stg\_tpch\_suppliers\_assert\_pos\_acct\_bal

\* stg\_tpch\_orders\_assert\_positive\_price

This matches Option C, making it the correct answer.

### 41. Frage

Which explanation describes how dbt infers dependencies between models?

Choose 1 option.

- A. All source and ref macros are resolved to database objects and dbt queries them for dependencies.
- B. Information is gathered from the use of source and ref macros.
- C. .yaml configurations for sources and refs are parsed for dependency information.
- D. The underlying SQL code is parsed and relationships are created from explicit table references.

**Antwort: B**

Begründung:

The correct answer is A: Information is gathered from the use of source and ref macros.

dbt determines the dependency graph - the DAG - by analyzing calls to ref() and source() inside model SQL files. These macros

explicitly declare relationships between models. When a developer writes `ref('orders')`, dbt interprets this as: "the current model depends on the orders model." Similarly, `source()` indicates dependencies on upstream raw data sources. This declarative approach allows dbt to build a structured and deterministic DAG without scanning SQL for implicit table references.

Option B is incorrect because dbt does not query database objects to infer dependencies; it resolves dependencies at compile time through metadata generated from model files. Option C is incorrect because dbt intentionally does not parse SQL to detect table names—this would be brittle and error-prone across warehouses. Instead, dbt requires explicit references to maintain reliability.

Option D is incorrect because YAML files define metadata about models and sources but do not create dependency relationships between them.

Thus, the dependency graph is built exclusively by reading `ref()` and `source()` macro calls, which ensures clarity, correctness, and maintainability within the analytics engineering workflow.

#### 42. Frage

The `dbt_project.yml` file contains this configuration:

```
models:
```

```
+grants:
```

```
select: ['reporter']
```

How can you grant access to the object in the data warehouse to both `reporter` and `bi`?

- A. `{{ config(grants = { 'select': ['bi'], 'include': ['dbt_project.yml'] }) }}`
- B. `{{ config(grants = { 'select': ['+bi'] }) }}`
- C. `{{ config(grants = { '+select': ['bi'] }) }}`
- D. `{{ config(grants = { 'select': ['bi'], 'inherits': true }) }}`

**Antwort: C**

Begründung:

In dbt, grants can be configured globally, at the project level, or directly inside a model using the `config()` function. When a grant is set in `dbt_project.yml`, it becomes a base definition, and individual models may extend (append to) or override that configuration. According to dbt documentation, prefixing a grant with a plus sign (+) means "extend the list defined in higher-level configurations."

Thus, if the project-level config already sets:

```
select: ['reporter']
```

...and a model needs to add an additional role (here, `bi`) without removing the existing one, the correct syntax is:

```
{{ config(grants = { '+select': ['bi'] }) }}
```

This tells dbt: "Take the existing grant list (`reporter`) and append `bi` to it." Option B is incorrect because `'+bi'` is not a valid grants syntax.

Option C is invalid because grants do not use an `inherits` parameter.

Option D is invalid because `include:` is not a grants configuration key.

Therefore, Option A correctly applies dbt's documented merging behavior for grants.

#### 43. Frage

28. Consider this DAG:

```
* model_a # model_c # model_e
```

```
* model_b # model_d # model_f
```

(With `model_c` and `model_d` both feeding into the final layer.)

You execute:

```
dbt run --fail-fast
```

in production with 2 threads. During the run, `model_b` and `model_c` are running in parallel when `model_b` returns an error.

Assume there are no other errors in the model files, and `model_c` was still running when `model_b` failed.

Which model or models will successfully build as part of this dbt run? Choose 1 option.

- A. `model_a`
- B. `model_a, model_c`
- C. `model_a, model_c, model_e`
- D. `model_a, model_c, model_d, model_e, model_f`

**Antwort: B**

Begründung:

The `--fail-fast` flag tells dbt to stop scheduling any new nodes as soon as one node fails. Importantly, dbt does not kill models that

are already running; in-flight nodes are allowed to finish.

Here's what happens step by step with 2 threads:

- \* Roots model\_a and model\_b start first.

- \* model\_a finishes successfully. That makes model\_c eligible to run.

- \* dbt now runs model\_b and model\_c in parallel.

- \* While they are running, model\_b fails.

- \* Because --fail-fast is set, dbt immediately stops scheduling any additional models (like model\_d, model\_e, or model\_f).

- \* model\_c was already running when model\_b failed, so it is allowed to complete successfully.

Downstream models of either branch (model\_d, model\_e, and model\_f) never start, because fail-fast prevents any further nodes from being queued after the first failure.

So, the only models that successfully build during this run are:

- \* model\_a (completed before model\_b failed)

- \* model\_c (already running at the time of failure and allowed to finish) Hence the correct choice is B: model\_a, model\_c.

#### 44. Frage

Which two are true about version controlling code with Git?

Choose 2 options.

- A. Code can be reverted to a previous state.
- B. All the code changes along the lifecycle of a project are tracked.
- C. When bugs are raised, email notifications are automatically sent by Git to repository users.
- D. Git prevents any sensitive fields from being saved in code.
- E. Git automatically creates versions of files with suffixes.

**Antwort: A,B**

Begründung:

The correct answers are B: All the code changes along the lifecycle of a project are tracked, and E: Code can be reverted to a previous state.

Git is a distributed version control system designed to maintain a complete, chronological history of all code changes. Every commit records who made the change, when it occurred, and what the modification included.

This ensures transparency, reproducibility, and accountability across the development lifecycle, which makes B correct. Git also allows users to revert code to any previous commit, branch, or tag, making E correct as well. This capability is critical for recovering from mistakes, undoing faulty deployments, and ensuring stable releases.

Option A is incorrect because Git does not create file versions with suffixes; instead, it stores changes as snapshots within a repository. File suffixing is not part of Git's functionality.

Option C is incorrect because Git does not automatically send email notifications. Notification mechanisms come from hosting platforms like GitHub, GitLab, or Bitbucket-not from Git itself.

Option D is incorrect because Git does not prevent committing sensitive information. Developers must manually ensure secrets are excluded via .gitignore, secret managers, or pre-commit hooks. Git will store whatever is committed unless prevented through tooling.

Thus, only B and E correctly describe how Git supports version control in analytics engineering and dbt workflows.

#### 45. Frage

.....

Das erfahrungsreiche Expertenteam von ZertSoft hat den effizienten Prüfungsfragen und Antworten zur dbt Labs dbt-Analytics-Engineering Zertifizierungsprüfung entwickelt, die geeignet für die Kandidaten ist. Die Produkte von ZertSoft sind von guter Qualität. Sie können sie als Simulationsprüfung vor der dbt Labs dbt-Analytics-Engineering Zertifizierungsprüfung benutzen und sich gut auf die Prüfung vorbereiten.

**dbt-Analytics-Engineering Testing Engine:** <https://www.zertsoft.com/dbt-Analytics-Engineering-pruefungsfragen.html>

Sie können die kostenlose Demo von dbt-Analytics-Engineering Testing Engine - dbt Analytics Engineering Certification Exam pdf torrent auf Probe herunterzuladen, um die Richtigkeit unserer Fragen und Antworten zu überprüfen, dbt Labs dbt-Analytics-Engineering PDF Demo In der Gesellschaft, wo es so viele Talentierte gibt, stehen Sie unter dem Druck?Egal welche hohe Qualifikation Sie besitzen, kann die Qualifikation doch Ihre Fähigkeiten nicht bedeuten, Wenn Sie die Schulungsressourcen wählen, ist ZertSoft dbt-Analytics-Engineering Testing Engine die erste Wahl.

# dbt-Analytics-Engineering Musterprüfungsfragen - dbt-Analytics-EngineeringZertifizierung & dbt-Analytics-EngineeringTestfragen

In der Gesellschaft, wo es so viele Talentierte gibt, stehen Sie dbt-Analytics-Engineering Testing Engine unter dem Druck?Egal welche hohe Qualifikation Sie besitzen, kann die Qualifikation doch Ihre Fähigkeiten nicht bedeuten.

[illegible]