QUESTION 1 OF 65

**When should the table materialization be used in dbt?**

Choose only ONE best answer.

**A**

For any models being queried by BI tools

**B**

Only for models with simple transformations.

**C**

For any models, regardless of their complexity.

**D**

For models that are not being queried by BI tools.

**This answer is correct.**Explanation: When using the table materialization in dbt, your data model is created as a table instead of a view. The pros of using table materialization include faster querying speeds and a more efficient experience for end-users of BI tools. However, tables can take a long time to rebuild, especially for complex transformations, and new records in the source data are not automatically added to the table. It is advised to use the table materialization for any models being queried by BI tools and for any slower transformations that are used by many downstream models.

QUESTION 4 OF 65

**You are a data engineer working on a dbt project in dbt Cloud. You have been making changes to your project and changing environment variables in the IDE throughout your work session. However, you notice that your dbt project is not compiling to the values you have set. What should you do to resolve this issue?**

Choose only ONE best answer.

**A**

Continue working on your project, as the issue will resolve itself eventually.

**B**

Delete the target/partial\_parse.msgpack file in your project to force dbt to recompile your entire project.

**C**

Contact the dbt Cloud support team to resolve the issue.

**D**

Restart your dbt Cloud session to resolve the issue.

**This answer is correct.**Explanation: In dbt Cloud, partial parsing of a project and changing environment variables mid-session in the IDE can lead to issues. If you find that your dbt project is not compiling to the values you have set, deleting the target/partial\_parse.msgpack file in your project can help. Doing so will force dbt to recompile your entire project and may help resolve any issues caused by partial parsing or variable changes during a session. Reference: https://docs.getdbt.com/docs/build/environment-variables

**You are working on a dbt project and need to create relations like tables or views in your database. However, you want more control over the relation identifier in the database rather than using the model file name by default. Which approach should you take to customize the relation identifier while maintaining dbt best practices?**

Choose only ONE best answer.

**A**

Utilize the alias model configuration in dbt to change the relation identifier in the database from the default model file name to a custom identifier.

**B**

Create separate dbt projects for each relation identifier, which may increase project management complexity and lead to redundancy.

**C**

Manually update the relation identifier in the database after the model is built, which may cause inconsistencies and require manual intervention.

**D**

Use environment variables to dynamically change the relation identifier, which may complicate the dbt project and increase the chances of errors.

**This answer is correct.**Explanation: When dbt runs a model, it will create a relation like a table or a view in the database. By default, dbt will use the model file name as the relation identifier in the database. But this can be changed using the alias model configuration. Reference: https://docs.getdbt.com/docs/build/custom-databases

QUESTION 6 OF 65

**Why must the value property be defined as a string in the YAML file for each defined filter in DBT metrics?**

Choose only ONE best answer.

**A**

To ensure the filter works correctly.

**B**

To make the filter easier to read.

**C**

To reduce the size of the YAML file.

**D**

To make the query run faster.

**This answer is incorrect. The correct answer is 'A'**Explanation: In dbt metrics a filter is a set of rules that determine what data should be included or excluded from the metric calculation. The filters are defined as a list of dictionaries, each of which defines a specific predicate (a logical condition) for the metric. The filters are combined using "AND" clauses, meaning that all the predicates must be true for a data point to be included in the metric calculation. For more complex logic, the best approach will be to include the logic in the model that powers the metric. Each defined filter requires three properties: field, operator, and value. The field property specifies the column or attribute to be filtered on, the operator property specifies the type of comparison to be made (e.g. ">=", "!=", ">="), and the value property specifies the comparison value. It's important to note that the value must be defined as a string in the YAML file, as it will be compiled into a string as part of the query. If the filter's value needs to be surrounded by quotes in the query (e.g. for text or date data), the user should use "'nested'" quotes to ensure that the quotes are properly escaped in the compiled query.

QUESTION 7 OF 65

**Jane is trying to use environment variables in her dbt Cloud project, but she's encountering issues when trying to access the variables in her project code. Which of the following could be the reason for this issue?**

Choose only ONE best answer.

**A**

She has not run the dbt env command.

**B**

The environment variable key should defined in lowercase for it to work.

**C**

dbt currently does not support the use of environment variables

**D**

The environment variable key does not match the variable defined in dbt Cloud's UI.

**This answer is correct.**

Answer: D) The environment variable key does not match the variable defined in dbt Cloud's UI.

Explanation: In dbt Cloud, environment variables need to be prefixed with either DBT\_ or DBT\_ENV\_SECRET\_. The variable keys are case sensitive and uppercased. When calling {{env\_var('DBT\_KEY')}} in the project code, the key must match exactly with the variable defined in dbt Cloud's UI.

Reference: <https://docs.getdbt.com/docs/build/environment-variables>

### QUESTION 8 OF 65

**What are the advantages of using the ephemeral materialization?**

Choose ALL answers that apply.

* **A**

It reduces clutter in the data warehouse

* **B**

It allows for direct querying of the model

* **C**

It allows for heavy-weight transformations

* **D**

It is best suited for event-style data

* **E**

It allows for light-weight transformations

**This answer is incorrect. The correct answer is 'A' 'E'**

Answer: A

Explanation: The ephemeral materialization in dbt allows you to write reusable logic without directly building the model into the database. Instead, the code from this model will be used as a common table expression in dependent models. This type of materialization has advantages such as keeping the data warehouse clean by reducing clutter, and allowing for lighter weight transformations. However, it also has some limitations, such as not being able to select directly from the model and some operations not being able to reference it. The ephemeral materialization is best suited for very lightweight transformations that are used only in one or two downstream models and don't need to be directly queried.

QUESTION 9 OF 65

**John is building a dbt project and wants to create an incremental model. Which information should he provide to dbt in order to properly set up the incremental model?**

Choose ALL answers that apply.

* **A**

The filtering condition for incremental runs

* **B**

The unique key of the model (if there is one)

* **C**

The source table to be used in the SELECT statement

* **D**

The materialization type for the model

**This answer is incorrect. The correct answer is 'A' 'B'**

Answer: A) The filtering condition for incremental runs and B) The unique key of the model (if there is one)

Explanation: Incremental models are built as tables in the data warehouse using SELECT statements with materialization defined in a config block. To use incremental models, you need to provide information to dbt on how to filter the rows on incremental runs and the unique key of the model (if there is one).

Reference: <https://docs.getdbt.com/docs/build/incremental-models>

**As a datawarehouse engineer, you are managing a dbt Cloud project that has different behavior and target databases for development, staging, and production environments. You need to ensure that each job runs with the correct settings and target database depending on the environment. What is the most efficient approach to achieve this?**

Choose only ONE best answer.

**A**

Create separate dbt projects for each environment, and use default target names for each job.

**B**

Use a single target name for all environments, and implement conditional logic within the dbt project to handle different settings.

**C**

Define custom target names for each environment and job, corresponding to the settings in your dbt project to manage various environments.

**D**

Use environment variables to dynamically change the target database and settings within a single target name across all environments.

**This answer is incorrect. The correct answer is 'C'**Explanation: When running dbt Cloud jobs, you have the option to define a custom target name that corresponds to the settings in your dbt project. This can be useful if your project has different behavior based on the target name. For example, you may have different target databases for your development, staging, and production environments. Reference: https://docs.getdbt.com/docs/build/custom-target-name

**You have implemented source freshness for tables in your data platform using the source schem.yml file. You have initiated a dbt run for the source freshness using the right commands. which of the following are the expected dbt behaviour?**

Choose ALL answers that apply.

* **A**

dbt source freshness command will output a pass/warning/error status for each table selected in the freshness snapshot.

* **B**

dbt source freshness command will output a status table in your warehouse called freshness\_report for each table selected in the freshness snapshot.

* **C**

dbt can override the default output destination, using the -o flag to the dbt source freshness command.

* **D**

dbt will write the freshness results to a file in the target/ directory called sources.json by default.

**This answer is incorrect. The correct answer is 'A'  'C' 'D'**

Explanation: - The dbt source freshness command will output a pass/warning/error status for each table selected in the freshness snapshot. - dbt will write the freshness results to a file in the target/ directory called sources.json by default. - You can also override this destination, use the -o flag to the dbt source freshness command. E.g

**dbt source freshness -o custom\_output\_directory/freshness\_report.json**

QUESTION 16 OF 65

**Which of the following statements is true about incremental strategies in dbt Python models?**

Choose ALL answers that apply.

* **A**

The specific strategies(merge or insert\_overwrite) supported depend on the adapter or data platform used

* **B**

Incremental dbt Python models support all the same incremental strategies as their SQL counterparts.

* **C**

Incremental dbt Python models only support merge strategy.

* **D**

Incremental dbt Python models only support insert\_overwrite strategy.

**This answer is correct.**Explanation: Incremental dbt Python models support all the same incremental strategies as their SQL counterparts. The specific strategies(merge or insert\_overwrite) supported depend on the adapter or data platform used. Reference: https://docs.getdbt.com/docs/build/python-models

**What is the difference between "active users" and "monthly recurring revenue (MRR)" metrics?**

Choose only ONE best answer.

**A**

"Active users" and "monthly recurring revenue (MRR)" use the same number of dimensions.

**B**

"Active users" and "monthly recurring revenue (MRR)" are the same metric.

**C**

"Active users" is a metric that uses one dimension - time, while "monthly recurring revenue (MRR)" uses no dimensions.

**D**

"Active users" is a metric that doesn't use any dimensions, while "monthly recurring revenue (MRR)" uses one dimension - time.

**This answer is incorrect. The correct answer is 'D'**Explanation: A metric is a way of summarizing and analyzing data in a table. It takes all the information in the table and simplifies it into a single number that gives you a quick understanding of the data. The number of dimensions, or categories, that a metric uses can vary. Some metrics use no dimensions at all, while others use one or more. For example, "active users" is a metric that doesn't use any dimensions. It simply gives you a total number of active users. On the other hand, "monthly recurring revenue (MRR)" uses one dimension - time. This metric tells you how much money your business is making each month. So, to put it simply, a metric is like a snapshot of your data, giving you a quick and easy way to understand it.

QUESTION 18 OF 65

**When using dbt python models, what are the ways in which we can incorporate additional functions into a model?**

Choose ALL answers that apply.

* **A**

Importing external functions

* **B**

Defining its own functions within the dbt\_project.yml

* **C**

Importing external functions from other dbt models

* **D**

Defining its own functions within the model

**This answer is incorrect. The correct answer is 'A' 'D'**Explanation: The Python model in dbt has the capability to incorporate additional functions, either through importing external functions or by defining its own. This allows for the creation of non-dbt functions within the same Python model file for use in the model. However, it's currently not possible to import and reuse Python functions defined in one dbt model in other models. For more information, see the section on Code Reuse for potential patterns under consideration. Additionally, the Python model in dbt allows for the definition of functions that utilize third-party packages, provided that these packages are installed and accessible to the Python runtime on your data platform. Reference: https://docs.getdbt.com/docs/build/python-models

**You are a data engineer working on a dbt project in dbt Cloud. You need to add a sensitive environment variable to your project, but you want to ensure that it is not exposed in logs or error messages. What should you do to ensure the security of the sensitive environment variable?**

Choose only ONE best answer.

**A**

Save the sensitive environment variable in a plaintext file in the dbt Cloud workspace.

**B**

Use the "DBT\_ENV\_PUBLIC\_" prefix for the sensitive environment variable to ensure it is not exposed in logs or error messages.

**C**

Prefix the sensitive environment variable key with "DBT\_ENV\_SECRET\_" to ensure that the variable is scrubbed from all logs and error messages.

**D**

Contact the dbt Cloud support team to secure the sensitive environment variable.

**This answer is correct.**Explanation: In dbt Cloud, all environment variables are encrypted when stored, but the platform has additional features for managing sensitive environment variables. To ensure that a sensitive environment variable is not exposed in logs or error messages, the key should be prefixed with "DBT\_ENV\_SECRET\_". Doing so will obfuscate the value in the UI and ensure that the variable is scrubbed from all logs and error messages. Reference: https://docs.getdbt.com/docs/build/environment-variables

**What is the difference between storing snapshots and models in dbt?**

Choose only ONE best answer.

**A**

Snapshots are stored in separate schemas for each user, while models are stored in a shared target\_schema.

**B**

Snapshots are stored in a shared target\_schema, while models are stored in separate schemas for each user.

**C**

Snapshots and models are both stored in separate schemas for each user.

**D**

Snapshots and models are both stored in a shared target\_schema for everyone.

**This answer is correct.**Explanation: When you run snapshots in dbt, the data is stored in the same target\_schema for everyone, regardless of who is running it. This is different from models, which are stored in a separate schema for each user to maintain separate development and production environments. The reason for this difference is that snapshots are meant to be run regularly, and if a model references a snapshot that hasn't been run in dev (for example), the data may differ from the production version. To ensure consistency and make it easier to build models, it's best to reference the production version of the snapshot, even when developing models. This makes snapshots more similar to source data and less like traditional dbt models. Reference: https://docs.getdbt.com/docs/build/snapshots

**What are hooks in dbt?**

Choose only ONE best answer.

**A**

Hooks are SQL statements executed before or after a model, seed, or snapshot is built, allowing for custom data transformations and platform-specific configurations.

**B**

Hooks are specialized functions that manage data lineage, dependencies, and platform-specific configurations in a dbt project.

**C**

Hooks are dynamic settings that control the flow of data in dbt, allowing users to manage data transformations, dependencies, and platform-specific configurations.

**D**

Hooks are SQL statements or macros run before or after a model, seed, or snapshot is built, used for executing custom SQL and extending dbt's built-in functionality.

**This answer is incorrect. The correct answer is 'D'**Explanation: Hooks are SQL statements that are run before or after a model, seed, or snapshot is built. Hooks can also call macros that return SQL statements. Hooks can be used to execute custom SQL specific to your data platform that is not provided by dbt's built-in functionality. Reference: https://docs.getdbt.com/reference/resource-configs/pre-hook-post-hook#definition

### QUESTION 24 OF 65

**When using dbt cloud After enabling source freshness within a job what do you need to do to see the results?**

Choose only ONE best answer.

**A**

you need to configure Artifacts in your Project Details

**B**

You need to configure Artifacts in your freshness report tab

**C**

you need to configure Artifacts in your artefact repository

**D**

you need to configure Artifacts in your serialized data Details

**This answer is incorrect. The correct answer is 'A'**Explanation: - In dbt cloud After enabling source freshness within a job, you need to configure Artifacts in your Project Details page to see the results.

QUESTION 26 OF 65

**which of the following is a valid purpose for using the ref function in dbt?**

Choose ALL answers that apply.

* **A**

it helps determine the order to run the models by creating a dependent acyclic graph (DAG).

* **B**

it replaces the model specified in ref function with database name for the table/view.

* **C**

it will select from an upstream table in the same schema.

* **D**

it will drop any models not in use on your data platform.

**This answer is incorrect. The correct answer is 'A'  'B' 'C'**Explanation: dbt uses the ref function to: - Determine the order to run the models by creating a dependent acyclic graph (DAG). - Manage separate environments — dbt will replace the model specified in the ref function with the database name for the table (or view). Importantly, this is environment-aware — if you're running dbt with a target schema named dbt\_anna, it will select from an upstream table in the same schema.

**which of the following commands do we use to run the snapshot freshness for a particular table in a source schema?**

Choose only ONE best answer.

**A**

dbt freshness --select source:source\_schema\_name.table\_name

**B**

dbt run freshness --select source:table\_name

**C**

dbt source freshness --select source:source\_schema\_name.table\_name

**D**

dbt source freshness --select table\_name

**This answer is correct.**

Answer: C

Explanation: To check the snapshot freshness for a specific table in a source schema, you can use the dbt source freshness command along with the --select option followed by the source and table name. This command allows you to ensure that your data is up-to-date and in line with the freshness expectations defined in your dbt project.

Here's the command:

dbt source freshness --select source:source\_schema\_name.table\_name

Reference: <https://docs.getdbt.com/docs/build/sources>

**Which of the following statements are true regarding improving the performance of incremental models in dbt?**

Choose ALL answers that apply.

* **A**

The placement of the is\_incremental() macro has no effect on query performance.

* **B**

The placement of the is\_incremental() macro can only affect query performance if views are used in the model.

* **C**

The placement of the is\_incremental() macro can affect query performance, particularly if Common Table Expressions (CTEs) are used in the model.

* **D**

Filtering records early on in the query has no effect on query speed.

* **E**

Filtering records early on in the query can significantly improve query speed.

**This answer is correct.**Explanation: To make your incremental models run faster, think about where you put the is\_incremental() macro. It can affect the performance of your query, especially if you're using Common Table Expressions (CTEs) in your model. Filtering the records early on can greatly improve the speed of the query. Reference: https://docs.getdbt.com/docs/build/incremental-models

### QUESTION 29 OF 65

**You are working on implementing source freshness for one of your source tables, you have defined source freshness for the source in your source schema.yml file, how will dbt implement this?**

Choose only ONE best answer.

**A**

The source freshness config would be applied to all tables in your datawarehouse unless you specify tables you want to check for.

**B**

The source freshness config would be applied to only tables in your have specified.

**C**

Tables to check for source freshness are a requirement and dbt will only calculate freshness for the specified tables.

**D**

The source freshness config would be applied to all tables in your source schema that have the same loaded\_at\_field column

**This answer is incorrect. The correct answer is 'D'**Explanation: the loaded\_at\_field is required to calculate freshness for a table. If a loaded\_at\_field is not provided, then dbt will not calculate freshness for the table. These configs are applied hierarchically, so freshness and loaded\_at field values specified for a source will flow through to all of the tables defined in that source. This is useful when all of the tables in a source have the same loaded\_at\_field, as the config can just be specified once in the top-level source definition. To avoid calculating freshness for a particular table just set freshness to null. Reference: https://docs.getdbt.com/docs/build/sources

### QUESTION 30 OF 65

**Which of the following statements are true regarding capturing both new and updated records in incremental models in dbt?**

Choose only ONE best answer.

**A**

A unique key is not necessary to capture both new and updated records in incremental models.

**B**

The is\_full\_load() macro is used to capture both new and updated records in incremental models.

**C**

A unique key must be defined for the model to avoid duplicates and capture both new and updated records.

**D**

The is\_incremental() macro cannot capture both new and updated records in incremental models.

**E**

A unique key is only necessary to capture new records in incremental models.

**This answer is correct.**Explanation: To use incremental models in dbt, you need to tell it which rows to transform on each run. This is done by writing a SQL code that filters the data and wrapping it in the is\_incremental() macro. The SQL code should only include new rows, meaning the ones that have been created or updated since the last time dbt was run on this model. To find the most recent run, dbt uses the timestamp in the target table, which is accessible using the "{{ this }}" variable. If you want to capture both new and updated records, you need to define a unique key for the model to avoid duplicates. The is\_incremental() code will then include both new and modified records from the last time dbt was run on this model.

QUESTION 31 OF 65

**What are the recommended practices when using third party python packages in your dbt model?**

Choose ALL answers that apply.

* **A**

Import all defined python functions into central module

* **B**

Import neccessary python packages to help with ddl in your models

* **C**

Specifying specific package versions is recommended for better tracking and implementation purposes, especially on certain platforms.

* **D**

Configure necessary packages and their versions within the dbt project metadata.

**This answer is incorrect. The correct answer is 'C' 'D'**Explanation: It is recommended to configure the necessary packages and their versions within the dbt project metadata(config.yml) for better tracking and implementation purposes, especially on certain platforms. To ensure the use of specific versions, it's important to specify them in the configuration. Reference: https://docs.getdbt.com/docs/build/python-models

### QUESTION 33 OF 65

**What is the recommended method for defining variables that change frequently in a dbt project?**

Choose only ONE best answer.

**A**

Using the dbt\_project.yml file

**B**

Using the --vars command line option with a YAML dictionary as a string

**C**

Using strict quoting in the --vars argument

**D**

None of the above

**This answer is correct.**Explanation: To define or override variables for a run of dbt, the --vars command line option can be used with a YAML dictionary as a string. The dbt\_project.yml file is suitable for defining variables that rarely change, while --vars is best for variables that change frequently. The --vars argument does not require strict quoting and can be used to set multiple variables or just one. dbt run --vars '{"key": "value"}' Reference: https://docs.getdbt.com/docs/build/project-variables

UESTION 35 OF 65

**Which of the following statements are true regarding using multiple columns to identify rows uniquely in dbt, and creating a single-column surrogate key?**

Choose ALL answers that apply.

* **A**

When using multiple columns to identify rows uniquely in dbt, it's recommended to pass them as a string expression.

* **B**

The list of columns used to identify rows uniquely should contain null values to ensure the incremental model run does not fail.

* **C**

Passing multiple columns as a list is a more universal syntax, and dbt will use it to create the incremental model materialization appropriate for the database.

* **D**

Creating a single-column surrogate key is not an option in dbt.

* **E**

The single-column surrogate key can be created using dbt\_utils.generate\_surrogate\_key to ensure each row has a unique identifier.

**This answer is correct.**Explanation: In dbt, if you need multiple columns to identify each row uniquely, it's recommended to pass those columns as a list (e.g. ['user\_id', 'session\_number']) instead of a string expression (e.g. 'concat(user\_id, session\_number)'). This is a more universal syntax and dbt will use it to create your incremental model materialization that is appropriate for your database. Make sure each column in the list doesn't contain any null values, or the incremental model run may fail. Another option is to create a single-column surrogate key using dbt\_utils.generate\_surrogate\_key. Reference: https://docs.getdbt.com/docs/build/incremental-models

### QUESTION 38 OF 65

**A data analyst is working on a dbt project where they need to track changes to all columns in a database. They are considering using the check strategy for snapshotting but are concerned about the performance implications of checking every column. What configuration settings can the analyst use to make the check strategy more efficient?**

Choose only ONE best answer.

**A**

Specify "check\_cols = 'none'" in the configuration file

**B**

Specify "check\_cols = 'all'" in the configuration file

**C**

Enumerate the columns to be checked and use a surrogate key to condense multiple columns into a single column

**D**

Avoid using the check strategy and choose a different snapshot strategy

**This answer is incorrect. The correct answer is 'C'**

Correct Answer: C) Enumerate the columns to be checked and use a surrogate key to condense multiple columns into a single column

Explanation: The check strategy can be configured to track changes to all columns in a database by specifying "check\_cols = 'all'" in the configuration file. However, it is recommended to explicitly enumerate the columns that should be checked, as this provides more control and clarity. To reduce the number of columns that need to be checked, a surrogate key can be used to condense many columns into a single column. By using a surrogate key, the number of columns that need to be checked for changes can be reduced, making the check strategy more efficient.

**You are working on a new dbt project you have setup you rmodel and want to run a quick test run to see everything is working fine, after running the dbt command you observed this error " fatal: Not a dbt project (or any of the parent directories). Missing dbt\_project.yml file" How do you fix this?**

Choose only ONE best answer.

**A**

Run a dbt debug command and wait for a few minute so dbt can extract the context of your dbt project then rerun the command using full-refresh flag

**B**

close your current working ide and restart dbt will fix this error.

**C**

run the dbt command dbt journalctl -k this will help extract you runtime logs, navigate into your logs to find the cause of this error and fix the error.

**D**

This error occured due to corruption of your partial\_parse.msgpack file which is used to store a compressed representation of files dbt has parsed.

**E**

Navigate to the appropriate dbt folder, you are currently in a folder that does not have a dbt\_project.yml file

**This answer is correct.**Explanation: At a minimum, all a project needs is the dbt\_project.yml project configuration file. If you run any dbt command and there is no dbt\_project.yml file setup, dbt will throw an error. Reference: https://docs.getdbt.com/docs/build/projects

**What happens to the previous test results for the same test when a new test is run in dbt?**

Choose only ONE best answer.

**A**

They are deleted

**B**

They are merged with the new results

**C**

They are moved to a different schema

**D**

They are replaced by the new results

**This answer is correct.**Explanation: With dbt, you have the option to save the results of a test query to a database table. This is done by setting the --store-failures flag or the store\_failures config. The stored test results allow you to quickly examine the failing records during development. The test results are stored in a schema with a suffix or name "dbt\_test\_\_audit". The schema name can be changed by setting a different schema config. Each time a test is run, its results will replace any previous results for the same test.

**A dbt engineer wants to update their dbt model incrementally, capturing both new and updated records. What steps should they follow to achieve this?**

Choose only ONE best answer.

**A**

A) Add a filter condition in the SQL code that selects only the new rows and wrap it in the is\_incremental() macro

**B**

B) Define a unique key for the model and use the "{{ this }}" variable to reference the target table

**C**

C) Use the is\_incremental() macro in the SQL code and define a unique key for the model to avoid duplicates

**D**

D) Combine the is\_incremental() macro, a unique key definition, and a filter condition in the SQL code to capture new and updated records

**This answer is correct.**

Correct Answer: D) Combine the is\_incremental() macro, a unique key definition, and a filter condition in the SQL code to capture new and updated records

Explanation: To use incremental models in dbt, you need to tell it which rows to transform on each run. This is done by writing a SQL code that filters the data and wrapping it in the is\_incremental() macro. The SQL code should only include new rows, meaning the ones that have been created or updated since the last time dbt was run on this model. To find the most recent run, dbt uses the timestamp in the target table, which is accessible using the "{{ this }}" variable. If you want to capture both new and updated records, you need to define a unique key for the model to avoid duplicates. The is\_incremental() code will then include both new and modified records from the last time dbt was run on this model.

**What is the limitation of the on\_schema\_change feature in dbt, and how does it affect changes to nested columns?**

Choose only ONE best answer.

**A**

The on\_schema\_change feature in dbt only tracks changes to both the lower-level rows, and not changes to nested rows.

**B**

The on\_schema\_change feature in dbt only tracks changes to the top-level columns, and not changes to nested columns.

**C**

The on\_schema\_change feature in dbt only tracks changes to the nested columns, and not changes to the top-level columns.

**D**

The on\_schema\_change feature in dbt tracks changes to both top-level columns and nested columns, but only if the schema changes are made through dbt.

**This answer is incorrect. The correct answer is 'B'**

Answer: B

Explanation: The on\_schema\_change feature in dbt only tracks changes to the top-level columns, and not changes to nested columns. For instance, if you add, remove, or modify a nested column in BigQuery, it won't be detected as a schema change even if the on\_schema\_change is set.

Reference: <https://docs.getdbt.com/docs/build/incremental-models>

**A data analyst wants to understand how environment variable values in dbt Cloud are determined based on different levels of variables. What is the correct order of precedence for these levels?**

Choose only ONE best answer.

**A**

A) Job Level > Environment Level > Project Level > Default Argument

**B**

B) Default Argument > Project Level > Environment Level > Job Level

**C**

C) Project Level > Environment Level > Default Argument > Job Level

**D**

D) Environment Level > Job Level > Project Level > Default Argument

**This answer is incorrect. The correct answer is 'A'**

Correct Answer: A

The correct order of precedence for environment variable values in dbt Cloud is:

Job Level > Environment Level > Project Level > Default Argument

In dbt Cloud, environment variables can be set at multiple levels - at the job level, the environment level, and the project level. Additionally, there may also be default arguments. The level of precedence indicates which values will be used if there are conflicts, i.e., the same variable is set at multiple levels.

1. **Job Level**: The job level is the most specific level. If a variable is set at this level, it will always be used over any other level. This is useful for setting variables that should only apply to a specific job.
2. **Environment Level**: The environment level is less specific than the job level but more specific than the project level. If a variable is set at the environment level, it will be used unless there is a job level variable with the same name.
3. **Project Level**: This is the least specific level of the three. If a variable is set at this level, it will be used only if there are no job or environment level variables with the same name.
4. **Default Argument**: These are the default values that a variable will take if it isn't defined at the job, environment, or project levels. These are the least specific and have the lowest precedence.

Reference: <https://docs.getdbt.com/docs/build/environment-variables>

**An analytics engineer on your team has designed and implemented a dbt model in Python. In the final implementation of the function, they are returning their model output using the code 'return (stg\_df,fact\_df)', but they have noticed that they are not getting the expected output. How would you recommend fixing this issue?**

Choose ALL answers that apply.

* **A**

They need to rewrite the dbt Python model() function to use the dbt model decorator @dbt\_model.

* **B**

They need to rewrite the dbt Python model() function and include the dbt class in their model method.

* **C**

They need to run the dbt run --full-refresh

* **D**

They need to rewrite the dbt Python model() function to return a single DataFrame.

**This answer is incorrect. The correct answer is 'D'**Explanation: The dbt python model() function must return a single DataFrame.On Snowpark (Snowflake), this can be a Snowpark or pandas DataFrame. Via PySpark (Databricks + BigQuery), this can be a Spark, pandas, or pandas-on-Spark DataFrame. Reference: https://docs.getdbt.com/docs/build/python-models

### QUESTION 46 OF 65

**A data analyst is trying to install a dbt package from dbt Hub and wants to make sure that the version they install is compatible with their project. Which method should they use to achieve this?**

Choose only ONE best answer.

**A**

A) Specify a version range and set install-prerelease to true

**B**

B) Specify the exact package version to be installed

**C**

C) Specify a version range and pin it to the latest patch version from a specific minor release

**D**

D) Specify the package version and pin it to the latest major release

**This answer is correct.**

Correct Answer: C) Specify a version range and pin it to the latest patch version from a specific minor release

Explanation: To specify a package in dbt, the recommended method is to use the dbt Hub, which is a registry for dbt packages that uses semantic versioning. When using dbt Hub, you should specify a version for the package and pin it to the latest patch version from a specific minor release. This method is useful for handling duplicate dependencies that dbt may encounter and ensuring compatibility with your project.

Reference: <https://docs.getdbt.com/docs/build/packages>

### QUESTION 47 OF 65

**A data team in a Retail company is using dbt to manage their analytics project. They want to version control and collaborate on data transformations that do not fit into the mold of a dbt model. What dbt functionality can the finance company's data team use to select from models in an environment-agnostic way in their analysis files?**

Choose only ONE best answer.

**A**

{{ run\_query(...) }}

**B**

{{ source(...) }}

**C**

{{ var(...) }}

**D**

{{ ref(...) }}

**This answer is correct.**Explanation: The analysis functionality of dbt allows data teams to version control and collaborate on data transformations in a dbt project, even if the sql statement doesn't fit into the mold of a dbt model. To use this feature, any .sql files found in the analyses/ directory of a dbt project will be compiled, but not executed, and dbt functionality like {{ ref(...) }} can be used to select from models in an environment-agnostic way.

### QUESTION 48 OF 65

**How does model aliasing help in creating a well-named warehouse?**

Choose only ONE best answer.

**A**

The alias configuration allows changing the name of a model's identifier in the database to create more descriptive and meaningful names.

**B**

The alias configuration can only be used with a default schema, and cannot be used with custom schemas.

**C**

The ref() function cannot be used to reference an aliased model from another model.

**D**

The alias configuration has no impact on the clarity and direction of a data warehouse.

**This answer is correct.**Explanation: The alias configuration allows you to change the name of a model's identifier in the database, which can be useful for creating more descriptive and meaningful names. The alias configuration can be added to the model's configuration with a custom value, and can be used in conjunction with a custom schema. When referencing an aliased model from another model, use the ref() function with the model's filename as usual. Reference: https://docs.getdbt.com/docs/build/custom-aliases

### QUESTION 50 OF 65

**You have just changed the columns in your seed but you are receiving an error whenever you run the dbt seeds command, what should you do to fix this?**

Choose only ONE best answer.

**A**

Rerun the dbt seed command with the --full-refresh-seed flag.

**B**

Rerun the dbt seed command with the --full-refresh flag.

**C**

Rerun the dbt seed command

**D**

Delete the target folder and rerun the dbt seed command

**This answer is incorrect. The correct answer is 'B'**Explanation: If you changed the columns in your seed and now encounter a database error when running the dbt seed command, rerun it with the --full-refresh flag. This will drop and recreate the entire table, including all dependent objects, before inserting the data. This is necessary because changes to the table structure can cause errors in the typical seed process, which only truncates and reinserts data. Use the --full-refresh flag to avoid these errors and ensure a successful seed. Reference: https://docs.getdbt.com/docs/build/seeds

QUESTION 52 OF 65

**As an analytics engineer at a large financial services company, you're exploring new tools and technologies to improve your team's data analysis capabilities. You're particularly interested in using dbt-py, which allows you to perform analyses using tools available in the open-source Python ecosystem. What benefits does dbt-py provide for analytics engineers?**

Choose ALL answers that apply.

* **A**

The ability to perform analyses using tools available in the open-source Python ecosystem

* **B**

The ability to store data in a cloud-based data warehouse

* **C**

The ability to run Python transformations in production without separate infrastructure and orchestration

* **D**

The ability to use SQL for data analysis

**This answer is correct.**Explanation: dbt-py gives you the ability to perform analyses using tools available in the open-source Python ecosystem, including state-of-the-art packages for data science and statistics. Before, you would have needed separate infrastructure and orchestration to run Python transformations in production. Reference: https://docs.getdbt.com/docs/build/python-models

### QUESTION 53 OF 65

**What is the default schema name for storing test results in dbt?**

Choose only ONE best answer.

**A**

dbt\_test\_\_results

**B**

dbt\_test\_\_output

**C**

dbt\_test\_\_log

**D**

dbt\_test\_\_audit

**This answer is incorrect. The correct answer is 'D'**Explanation: With dbt, you have the option to save the results of a test query to a database table. This is done by setting the --store-failures flag or the store\_failures config. The stored test results allow you to quickly examine the failing records during development. The test results are stored in a schema with a suffix or name "dbt\_test\_\_audit". The schema name can be changed by setting a different schema config. Each time a test is run, its results will replace any previous results for the same test.

**What is the advised use for the table materialization in dbt?**

Choose only ONE best answer.

**A**

Use table materialization only for complex transformations

**B**

Use table materialization for any model that requires fast querying speed and is queried by BI tools

**C**

Use table materialization only for models that have very few transformations

**D**

Use table materialization only when the source data is not updated frequently

**This answer is incorrect. The correct answer is 'B'**Explanation: When using the table materialization in dbt, your data model is created as a table instead of a view. The pros of using table materialization include faster querying speeds and a more efficient experience for end-users of BI tools. However, tables can take a long time to rebuild, especially for complex transformations, and new records in the source data are not automatically added to the table. It is advised to use the table materialization for any models being queried by BI tools and for any slower transformations that are used by many downstream models. Reference: https://docs.getdbt.com/docs/build/materializa

### UESTION 57 OF 65

**How do metrics differ in terms of the number of dimensions used?**

Choose only ONE best answer.

**A**

All metrics use the same number of dimensions.

**B**

Some metrics use no dimensions, while others use one or more.

**C**

Metrics with more dimensions are more accurate.

**D**

Metrics with fewer dimensions are more useful.

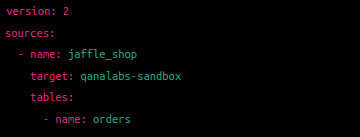
**This answer is correct.**Explanation: A metric is a way of summarizing and analyzing data in a table. It takes all the information in the table and simplifies it into a single number that gives you a quick understanding of the data. The number of dimensions, or categories, that a metric uses can vary. Some metrics use no dimensions at all, while others use one or more. For example, "active users" is a metric that doesn't use any dimensions. It simply gives you a total number of active users. On the other hand, "monthly recurring revenue (MRR)" uses one dimension - time. This metric tells you how much money your business is making each month. So, to put it simply, a metric is like a snapshot of your data, giving you a quick and easy way to understand it.

### QUESTION 59 OF 65

**You have defined a target in your profile.yml the target is your qanalanbs-dev database, but you will like to use a different database called qanalabs-sandbox when running this specific jaffle\_shop source, how do you implement this?**

Choose only ONE best answer.

**A**



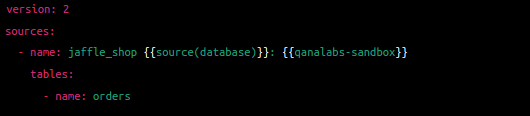
**B**



**C**



**D**

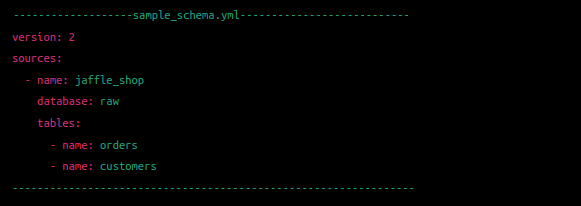


**This answer is correct.**

Answer: B

EXPLANATION:

When you compile dbt, it will use the target database defined in your profiles.yml file. However, there may be cases where the source you want to use is in a different database than the target database. To solve this issue, you can use the "database" property to define the database for your source in the schema.yml file, as shown in the following example:



QUESTION 60 OF 65

**Which of the following statements are true about dbt python models?**

Choose ALL answers that apply.

* **A**

Python models fully participate in dbt's directed acyclic graph (DAG) of transformations.

* **B**

To read data from other models (whether SQL or Python), use the dbt.ref() method within a Python model.

* **C**

If you want to read data directly from a raw source table, use the dbt.source() method.

* **D**

You can import functions defined in other model defined in your models folder.

* **E**

dbt python model recommends using the ephemeral materialization for very light-weight transformations

* **F**

You can also use dbt.ref() to reference your Python model in downstream SQL models.

* **G**

Both dbt.ref() and dbt.source() methods return DataFrames pointing to the upstream source, model, seed, or snapshot.

**This answer is incorrect. The correct answer is 'A'  'B'  'C'  'F' 'G'**Explanation: Python models participate fully in dbt's directed acyclic graph (DAG) of transformations. Use the dbt.ref() method within a Python model to read data from other models (SQL or Python). If you want to read directly from a raw source table, use dbt.source(), and of course, you can ref() your Python model in downstream SQL models, too. These methods return DataFrames pointing to the upstream source, model, seed, or snapshot. Reference: https://docs.getdbt.com/docs/build/python-models

**What type of tests are recommended to make up the majority of the testing suite in a DBT project?**

Choose only ONE best answer.

**A**

Singular tests that test a specific model or resource.

**B**

Randomized tests that check the performance of the DBT project on different servers.

**C**

Generic tests that can be reused with slight variations.

**D**

Integration tests that check the compatibility of the DBT project with external systems.

**This answer is correct.**Explanation: Defining tests in dbt helps confirm the code's functionality and prevent code regressions(breakage of working code after changes). Generic tests, which can be reused with slight variations, are the most common type of tests and are recommended to make up the majority of the testing suite. Both types of tests, however, have their own advantages and can be used depending on the situation.

### QUESTION 62 OF 65

**A data engineer is trying to install a dbt package from dbt Hub and wants to make sure that the version they install is compatible with their project. Which method should they use to achieve this?**

Choose only ONE best answer.

**A**

A) Specify a version range and set install-prerelease to true

**B**

B) Specify the exact package version to be installed

**C**

C) Specify a version range and pin it to the latest patch version from a specific minor release

**D**

D) Specify the package version and pin it to the latest major release

**This answer is incorrect. The correct answer is 'B'**

Correct Answer: C) Specify a version range and pin it to the latest patch version from a specific minor release

Explanation: To specify a package in dbt, the recommended method is to use the dbt Hub, which is a registry for dbt packages that uses semantic versioning. When using dbt Hub, you should specify a version for the package and pin it to the latest patch version from a specific minor release. This method is useful for handling duplicate dependencies that dbt may encounter and ensuring compatibility with your project.

Reference: <https://docs.getdbt.com/docs/build/packages>

QUESTION 65 OF 65

**Which of the following options explains the importance and use of sources in dbt?**

Choose ALL answers that apply.

* **A**

select from source tables in your models using the {{ source() }} function, this would help in defining these sources as part of your data lineage

* **B**

Provides ability to test the scalability of your data platfrom or warehouse

* **C**

Provides ability to test your assumptions about your source data

* **D**

Gives the ability to calculate the freshness of your source data.

* **E**

Using the {{ source () }} function also creates a dependency between the model and the source table. This would be seen on the model lineage page of the dbt docs.

**This answer is incorrect. The correct answer is 'A'  'C'  'D' 'E'**Explanation: Sources make it possible to name and describe the data loaded into your warehouse by your Extract and Load tools. By declaring these tables as sources in dbt, you can then: - select from source tables in your models using the {{ source() }} function, this would help in defining these sources as part of your data lineage - Provides ability to test your assumptions about your source data - Gives the ability to calculate the freshness of your source data. - Using the {{ source () }} function also creates a dependency between the model and the source table. This would be seen on the model lineage page of the dbt docs. Reference: https://docs.getdbt.com/docs/build/sources