6QUESTION 1 OF 65

**What are the advantages of Python models over slq models in dbt?**

Choose ALL answers that apply.

* **A**

Wider range of syntax options compared to SQL.

* **B**

faster to run and require less resources compared to SQL models.

* **C**

Access to the dbt context using jinja

* **D**

Ability to incorporate additional functions and third-party packages.

**This answer is incorrect. The correct answer is 'A' 'D'**Explanation: Python models in dbt offer advantages compared to SQL models but also have some drawbacks. Python models are slower to run and require more resources, but can be faster and cheaper from a human perspective when deploying via dbt with unified lineage, testing, and documentation. Python offers a wider range of syntax options compared to SQL, but the vast choices can be overwhelming. New data warehouse features may offer better mechanisms for deploying Python transformations. As a rule of thumb, if a transformation can be written equally well in SQL or Python, SQL is preferred as it's more accessible and easier to write performant code, but If there's a transformation you can't write in SQL, or where ten lines of elegant and well-annotated Python could save you 1000 lines of hard-to-read Jinja-SQL, Python is the way to go. Reference: https://docs.getdbt.com/docs/build/python-models

### QUESTION 3 OF 65

**You are a Analytics engineer working on a dbt project and need to customize the aliases for specific models in your database. The goal is to have more control over the model's identifiers and maintain a consistent naming convention. Which approach should you take to achieve this?**

Choose only ONE best answer.

**A**

Modify the model's filename directly in the database, as it is the default alias name, but this method may lead to inconsistencies in the naming convention.

**B**

Implement the generate\_alias\_name macro, which accepts two arguments (custom alias and node), to override the default alias generation process that works similarly to the generate\_schema\_name macro, providing greater control over model identifiers.

**C**

Use the generate\_schema\_name macro with the custom alias supplied in the model configuration and the node for which the custom alias is being generated, leveraging the existing macro but requiring additional customization to target aliases specifically.

**D**

Create a separate dbt project for each model with different alias requirements, using environment variables to control the model's identifier in the database, which may increase project management complexity.

**This answer is correct.**

Explanation: The alias generated for a model in dbt can be customized using the alias configuration parameter, which changes the name of a model's identifier in the database. The default name for the alias is the model's filename. However, the generate\_alias\_name macro can be overridden in a dbt project to change how dbt generates the alias. This macro works similarly to the generate\_schema\_name macro.

The macro can be created with the same name to override dbt's alias name generation. The macro accepts two arguments: the custom alias supplied in the model configuration and the node that a custom alias is being generated for.

The default implementation of generate\_alias\_name uses the supplied alias config (if present) as the model alias, otherwise falling back to the model name.

QUESTION 31 OF 65

**In the context of using incremental models in dbt, what is the purpose of the "unique\_key" and how can it be defined in the model's configuration block?**

Choose ALL answers that apply.

* **A**

The "unique\_key" helps in updating existing rows and ignoring duplicates by allowing new information to replace current information for an existing unique key; it can be defined as a single column name or a list of column names in the configuration block.

* **B**

The "unique\_key" helps in filtering out new rows from the source data, improving the performance of incremental models; it can be defined as a single column name or a list of column names in the configuration block.

* **C**

The "unique\_key" helps in identifying columns that should be indexed to improve query performance in the target table; it can be defined as a single column name or a list of column names in the configuration block.

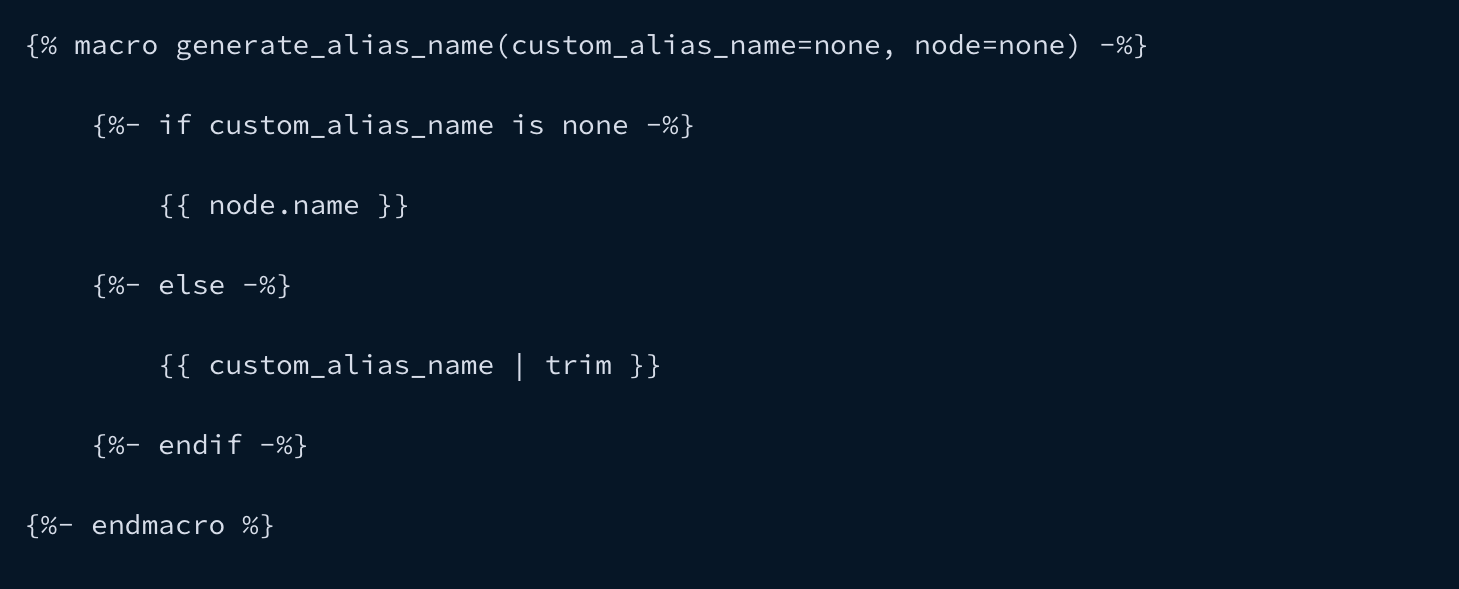
* **D**

The "unique\_key" helps in partitioning the target table to improve storage and retrieval efficiency; it can be defined as a single column name or a list of column names in the configuration block.

**This answer is correct.**

Answer: A,B

Explanation: The "unique\_key" in dbt helps in updating existing rows instead of just adding new ones. It allows new information to replace the current information if it arrives for an existing unique key, and ignores duplicate rows. The unique key can be a single column or a combination of columns and must not contain any nulls. If a unique key is not specified, dbt will add all returned rows from the model's SQL without considering duplicates. The unique key can be defined as a single column name or a list of column names in the model's configuration block.



Reference: <https://docs.getdbt.com/docs/build/custom-aliases>

]

QUESTION 4 OF 65

**What is the purpose of the on\_schema\_change parameter in dbt configuration, and what are the available options?**

Choose ALL answers that apply.

* **A**

It is used to specify how dbt should handle data changes in the source database, with the following options: ignore, fail, append\_new\_columns, and sync\_all\_columns.

* **B**

It is used to specify how dbt should handle schema changes, with the following options: ignore, fail, append\_new\_columns, and sync\_all\_columns.

* **C**

It is used to specify how dbt should handle query execution errors, with the following options: ignore, fail, append\_new\_columns, and sync\_all\_columns.

* **D**

It is used to specify how dbt should handle security configuration, with the following options: ignore, fail, append\_new\_columns, and sync\_all\_columns.

**This answer is correct.**

Correct Answer: B) It is used to specify how dbt should handle schema changes, with the following options: ignore, fail, append\_new\_columns, and sync\_all\_columns.

Explanation: The on\_schema\_change parameter in your configuration is used to specify how dbt should handle schema changes, with the following options: ignore: the default behavior, which does not handle schema changes fail: triggers an error message when the source and target schemas diverge append\_new\_columns: appends new columns to the existing table, but does not remove any missing columns sync\_all\_columns: adds new columns and removes missing columns, including data type changes. Note: none of the on\_schema\_change options backfill values for newly added columns, so manual updates or a full-refresh may be necessary.

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

### QUESTION 5 OF 65

**A data engineer wants to avoid calculating freshness for a specific table in a source schema while using dbt. What should they do to achieve this?**

Choose only ONE best answer.

**A**

Remove the loaded\_at\_field from the table configuration

**B**

Set the freshness for the table to a very high value

**C**

Set the freshness for the table to null

**D**

Exclude the table from the source definition

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

Correct Answer: C) Set the freshness for the table to null

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 7 OF 65

**A data engineer is using incremental models in dbt and notices that their unique key is not working as expected, causing errors during the incremental model run. What could be the possible reason and solution to this issue?**

Choose only ONE best answer.

**A**

The unique\_key is not defined in the model config

**B**

The unique\_key has multiple rows in either the existing table or the new data

**C**

The unique\_key is not used in the SQL code for the incremental model

**D**

The unique\_key is not supported by the database

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

Correct Answer: B) The unique\_key has multiple rows in either the existing table or the new data

Explanation: If a unique\_key is not working as expected and causing errors during the incremental model run, it is likely that the unique\_key has multiple rows in either the existing table or the new data. This violates the uniqueness constraint of the key and causes the run to fail. To resolve this issue, the data engineer should ensure that the unique\_key is truly unique in both the existing table and the new data, and that the unique\_key is properly defined in the model configuration.

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

QUESTION 5 OF 65

**Which of the following accurately describe the concept of metrics in dbt?**

Choose ALL answers that apply.

* **A**

Metrics are defined as a specific type of node in a directed acyclic graph (DAG).

* **B**

Metric definitions are expressed in YAML files, which are easy to understand and organize.

* **C**

DBT helps ensure the accuracy and consistency of metrics by encoding business logic into version-controlled code.

* **D**

Metrics can be exposed to downstream tools, which promotes efficient and precise reporting.

**This answer is correct.**Explanation: DBT supports defining metrics as a specific type of node in a directed acyclic graph (DAG). These metric definitions are expressed in YAML files, making them easily understandable and organized. By encoding business logic into version-controlled code, DBT helps ensure the accuracy and consistency of your metrics. Additionally, these metrics can be exposed to downstream tools, promoting efficient and precise reporting.

**You are a Analytics engineer working on a dbt project with multiple models that need to be logically grouped based on business units and control their visibility for end users. You also want to build models in a schema other than the target schema. What approach should you take to achieve these goals?**

Choose only ONE best answer.

**A**

Manually move the models to different schemas in the database after they are built, which may lead to inconsistencies and requires manual intervention.

**B**

Use a single schema for all models and implement conditional logic within the dbt project to handle visibility and grouping, which may complicate the project.

**C**

Specify custom schemas for models using the schema configuration key, applying it to specific models with a config block or to subdirectories by specifying the key in the dbt\_project.yml file, allowing logical grouping and visibility control.

**D**

Modify the target schema directly in the database to include the custom schema information, which may cause issues with database structure and naming conventions.

**This answer is correct.**Explanation: In dbt, you can specify a custom schema for a model using the schema configuration key. This allows you to build the model in a schema other than your target schema. To specify a custom schema for a specific model, you can use a config block within the model file itself. To apply it to a subdirectory of models, you can specify the schema configuration key in your dbt\_project.yml file for that subdirectory. This helps you logically group models and control their visibility, for example, grouping models based on the business unit that uses them, or hiding intermediate models in a staging schema and presenting only the ones that should be queried by end users in an analytics schema. It is important to note that by default, dbt will generate the schema name for a model by concatenating the custom schema to the target schema. Reference: https://docs.getdbt.com/docs/build/custom-schemas

### QUESTION 8 OF 65

**Which of the following is untrue about using dbt packages in your project?**

Choose only ONE best answer.

**A**

dbt packages are separate projects that include models and macros.

**B**

Adding a package to your project makes its models and macros part of your project.

**C**

Models in the package will be materialized when you run your project.

**D**

You can reference models from the package in your own models using the ref function.

**E**

You can use macros from the package in your own project.

**F**

Models in the package can only be referenced and will not be materialized when you run your project.

**G**

Leveraging packages in your dbt projects can help you more efficiently develop analytics solutions that solve specific problem areas.

**This answer is correct.**Explanation: dbt packages are separate projects with models and macros that address specific problem areas. By adding a package to your project, the models and macros included in the package become part of your own project. This means that models in the package will be materialized when you run your project, and you can use the ref function in your own models to reference models from the package. Additionally, you can use macros from the package in your own project. By leveraging packages in your dbt projects, you can more efficiently develop analytics solutions that solve specific problem areas. Reference: https://docs.getdbt.com/docs/build/packages

QUESTION 12 OF 65

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

Choose ALL answers that apply.

* **A**

Incremental models cannot update the target table with new data.

* **B**

Incremental model transforms new rows that have been created since the last time dbt ran are transformed.

* **C**

Incremental model updates rows that have been modified since the last time dbt ran are transformed.

* **D**

Incremental models transform all rows of source data each time they are run.

* **E**

Incremental models are not effective for updating data in your data warehouse incrementally.

**This answer is incorrect. The correct answer is 'B' 'C'**Explanation: Incremental models in dbt are built as tables in your data warehouse and they allow you to update the target table with only new or updated data from your source data. This means that on each run, only the rows that have been created or updated since the last time dbt ran are transformed, and inserted into the target table. The first time an incremental model is run, the target table is built by transforming all rows of source data. The subsequent runs will only transform the filtered rows, improving the performance of your transformations and reducing compute costs. This approach is effective for updating data in your data warehouse incrementally, thus improving its performance and efficiency. Reference: https://docs.getdbt.com/docs/build/incremental-models

### QUESTION 13 OF 65

**A company is implementing snapshot tables in dbt to track changes in employee data over time. They want to ensure that their snapshot data is accurate and provides a solid foundation for downstream analysis and processing. Why is it important to avoid including joins in snapshot queries in dbt?**

Choose only ONE best answer.

**A**

Joining tables can cause query performance issues

**B**

Joining tables can complicate the process of building a reliable "updated\_at" timestamp

**C**

Joining tables can cause conflicts with other models in the project

**D**

Joining tables can result in data duplication and inconsistencies

**This answer is incorrect. The correct answer is 'B'**Explanation: When creating snapshots in dbt, it is important to avoid including joins in your snapshot query. Joining tables in a snapshot can complicate the process of building a reliable "updated\_at" timestamp. To mitigate this issue, it is recommended to snapshot each table separately and then join them in downstream models. This approach helps to ensure that the snapshot data is accurate and provides a solid foundation for further analysis and processing. Reference: https://docs.getdbt.com/docs/build/snapshots

### QUESTION 18 OF 65

**As a data engineer at a growing e-commerce company, you've been tasked with building a new data pipeline using dbt. One of the requirements for the pipeline is to ensure that all columns have a default value. What is the recommended dbt approach to ensuring dbt uses the default value?**

Choose only ONE best answer.

**A**

Use the SQL "COALESCE" function to set "last\_updated" to the current timestamp if it is null

**B**

Add a "default" parameter to the column definition in the dbt model file

**C**

Use the SQL "IFNULL" function to check if "last\_updated" is null and set it to the current timestamp if it is

**D**

None of the above

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

Explanation: In dbt rather than creating default values for a column, use SQL to express defaults (e.g. coalesce(updated\_at, current\_timestamp()) as updated\_at)

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

### QUESTION 23 OF 65

**As a certified dbt developer, you are working with dbt models and have accidentally created two models with the same identifier in the database. This issue could lead to unexpected outputs and confusion for data consumers in your warehouse. How should you resolve this situation to maintain clarity and direction?**

Choose only ONE best answer.

**A**

Ignore the issue, as dbt automatically manages duplicate identifiers and prioritizes the latest model.

**B**

Manually delete one of the models from the database, which may cause loss of important data or functionality.

**C**

Follow the error message provided by dbt that checks for ambiguous model names and aliases, and change the schema or alias configurations of the resources as suggested.

**D**

Merge the two models with the same identifier into a single model, which may complicate the data processing and increase the chances of errors in the output.

**This answer is correct.**Explanation: When using aliases for dbt models, it's possible to accidentally create models with the same identifier in the database. If this happens, dbt would create only one of the views, and the output may not be what you expect. To prevent this, dbt checks for ambiguous model names and aliases and presents an error message with suggestions to change the schema or alias configurations of the resources. To work around this issue, you can configure a custom schema for one of the models. It's important to avoid ambiguous names to ensure clarity and direction for consumers of data in your warehouse. Reference: https://docs.getdbt.com/docs/build/custom-aliases

### QUESTION 24 OF 65

**What is the purpose of incremental models in dbt, and how do they improve the performance and efficiency of data transformations in your data warehouse?**

Choose only ONE best answer.

**A**

Incremental models update the target table with all source data rows, improving performance by reducing the number of transformations required.

**B**

Incremental models update the target table with only new or updated data from the source, reducing transformation time and compute costs by transforming only the filtered rows on subsequent runs.

**C**

Incremental models store source data in a temporary table, reducing storage costs and improving performance by reusing the temporary table across different runs.

**D**

Incremental models store target table data in a cache, improving performance by reducing the time required to retrieve data from the warehouse.

**This answer is correct.**

Answer:

B) Incremental models update the target table with only new or updated data from the source, reducing transformation time and compute costs by transforming only the filtered rows on subsequent runs.

Explanation: Incremental models in dbt are built as tables in your data warehouse and they allow you to update the target table with only new or updated data from your source data. This means that on each run, only the rows that have been created or updated since the last time dbt ran are transformed, and inserted into the target table. The first time an incremental model is run, the target table is built by transforming all rows of source data. The subsequent runs will only transform the filtered rows, improving the performance of your transformations and reducing compute costs. This approach is effective for updating data in your data warehouse incrementally, thus improving its performance and efficiency. Reference: <https://docs.getdbt.com/docs/build/incremental-models>

### QUESTION 25 OF 65

**What is the purpose of the "dbt\_valid\_from" column in a snapshot table in dbt?**

Choose only ONE best answer.

**A**

It represents the timestamp when a snapshot row was last updated

**B**

It indicates which columns should be included in the snapshot

**C**

It is used to specify the schema where the snapshot table should be stored

**D**

It represents the timestamp when a snapshot row was first inserted.

**This answer is incorrect. The correct answer is 'D'**Explanation: The "dbt\_valid\_from" column in a snapshot represents the timestamp when a snapshot row was first inserted. This information can be used to order the different "versions" of a record and provide a clear understanding of the history of the data. The "dbt\_valid\_from" column is a valuable tool for tracking the evolution of a record and can be used to help answer questions about the data over time. Reference: https://docs.getdbt.com/docs/build/snapshots

### QUESTION 28 OF 65

**A data team in a finance 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 functionality of dbt allows the finance company's data team to version control and collaborate on data transformations that do not fit into the mold of a dbt model?**

Choose only ONE best answer.

**A**

The analysis functionality of dbt

**B**

The modeling functionality of dbt

**C**

The documentation functionality of dbt

**D**

The test functionality of dbt

**E**

The deployment functionality of dbt

**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. Reference: https://docs.getdbt.com/docs/build/analyses

QUESTION 29 OF 65

**What is the purpose of the ephemeral materialization in dbt?**

Choose ALL answers that apply.

* **A**

To directly build a model into the database

* **B**

To write reusable logic without building the model into the database

* **C**

To allow for faster querying speeds

* **D**

To prevent data clutter in the data warehouse

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

Answer: B, D

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

**What are the advantages of using table materialization in dbt?**

Choose only ONE best answer.

**A**

Slower querying speeds and a less efficient experience for end-users of BI tools.

**B**

Faster querying speeds and a more efficient experience for end-users of BI tools.

**C**

Slower querying speeds and a more efficient experience for end-users of BI tools.

**D**

Faster querying speeds and a less efficient experience for end-users of BI tools.

**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.

QUESTION 34 OF 65

**The "is\_incremental()" macro in dbt determines whether a model is running in incremental mode or not, what are the conditions that must be met for the is\_incremental() macro to return true?**

Choose ALL answers that apply.

* **A**

The model does not have any dependencies

* **B**

The destination table already exists.

* **C**

dbt is not running in full-refresh mode.

* **D**

The model is set to use incremental materialization.

**This answer is incorrect. The correct answer is 'B'  'C' 'D'**Explanation: The "is\_incremental()" macro in dbt determines whether a model is running in incremental mode or not. It returns "True" if the destination table already exists, dbt is not running in full-refresh mode, and the model is set to use incremental materialization. The SQL used in the model should be able to run successfully under either scenario. Reference: https://docs.getdbt.com/docs/build/incremental-models

**In a company's data analytics project, they are using dbt to manage their data transformation pipelines. The team wants to define metrics that are consistent, version-controlled, and easily exposed to downstream tools. Which feature of dbt should they utilize to achieve this goal?**

Choose only ONE best answer.

**A**

Defining metrics using dbt's run operations in a separate SQL file.

**B**

Defining metrics as a specific type of node in a directed acyclic graph (DAG) using YAML files.

**C**

Defining metrics as a JSON object within the dbt\_project.yml file.

**D**

Defining metrics using custom Jinja templates in a separate SQL file.

**This answer is correct.**

Answer: B) Defining metrics as a specific type of node in a directed acyclic graph (DAG) using YAML files.

Explanation: DBT supports defining metrics as a specific type of node in a directed acyclic graph (DAG). These metric definitions are expressed in YAML files, making them easily understandable and organized. By encoding business logic into version-controlled code, DBT helps ensure the accuracy and consistency of your metrics. Additionally, these metrics can be exposed to downstream tools, promoting efficient and precise reporting.

QUESTION 38 OF 65

**What is true about the incremental models in Python?**

Choose ALL answers that apply.

* **A**

They support incremental strategies as SQL models

* **B**

They support only one incremental strategy

* **C**

The incremental strategies they support depend on the database adapter being used

* **D**

They do not require filtering of incoming data to only include new rows

**This answer is correct.**Explanation: In dbt, Python models have two materialization options: table and incremental. Incremental models in Python support the same incremental strategies as SQL models, but the specific strategies depend on the database adapter being used. It is not possible to use view or ephemeral materialization for Python models, nor can Python be used for non-model resources like tests and snapshots. For incremental models, the incoming data must be filtered to only include new rows. Note that the insert\_overwrite strategy for incremental models is not yet supported for BigQuery/Dataproc, but the merge incremental strategy is supported. Reference: https://docs.getdbt.com/docs/build/materializations

### QUESTION 39 OF 65

**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. It's now time to check the results of your run. How do you do this?**

Choose only ONE best answer.

**A**

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

**B**

dbt source freshness command will print a status table in your cli showing freshness report for each table selected in the freshness snapshot.

**C**

dbt source freshness command will output a freshness report for each table selected in the freshness snapshot.

**D**

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

**This answer is correct.**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. Reference: https://docs.getdbt.com/docs/build/sources

### QUESTION 41 OF 65

**A data engineer is considering using the "Incremental" materialization type in their dbt project. Which of the following statements best describes the use case and recommendation for using incremental materialization?**

Choose only ONE best answer.

**A**

Incremental materialization is best suited for event-style data and is recommended as the starting point for all dbt projects.

**B**

Incremental materialization is best suited for event-style data and should be used only when dbt runs are becoming too slow.

**C**

Incremental materialization is best suited for dimension tables and is recommended as the starting point for all dbt projects.

**D**

Incremental materialization is best suited for dimension tables and should be used only when dbt runs are becoming too slow.

**This answer is correct.**

Answer:B) Incremental materialization is best suited for event-style data and should be used only when dbt runs are becoming too slow.

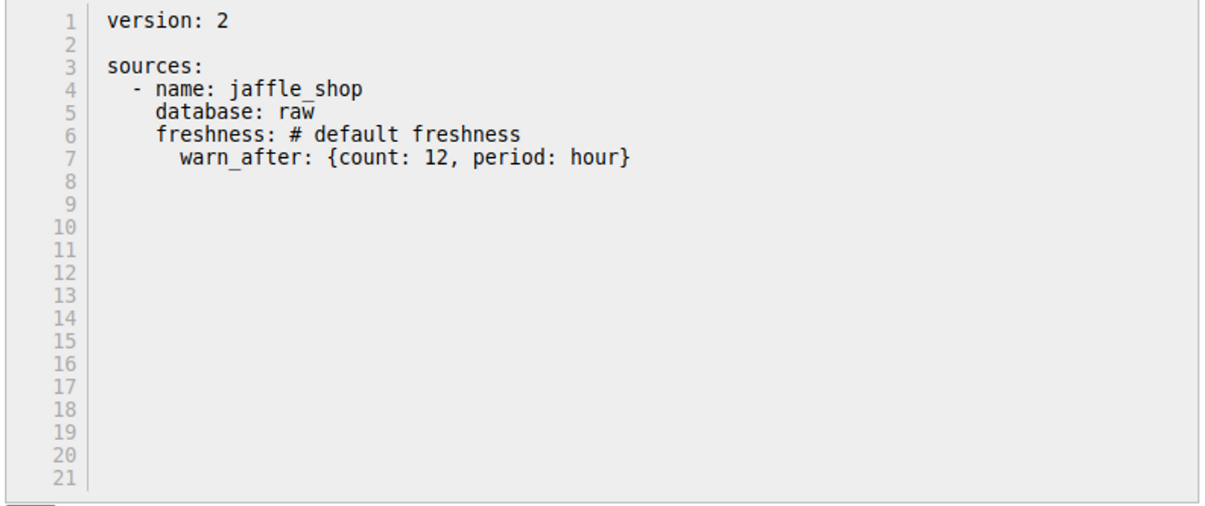
Explanation: The "Incremental" materialization type in dbt allows for records to be inserted or updated in a table since the last time dbt was run, reducing the build time. This type of materialization is best suited for event-style data. However, it requires extra configuration and is considered an advanced usage of dbt, so it is advised to use this type only when dbt runs are becoming too slow and not as a starting point. Reference: <https://docs.getdbt.com/docs/build/materializations>

QUESTION 42 OF 65

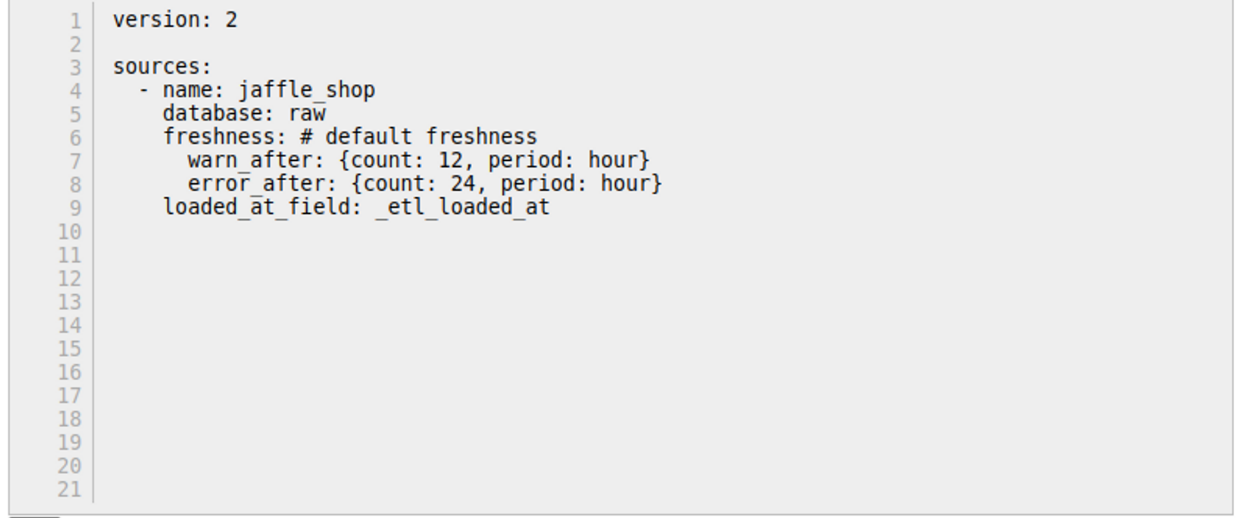
**When implementing a freshness block property your colleague asks you the dbt requirements for ensuring dbt calculates the freshness of your data sources, which of the following schema options would work as expected.**

Choose ALL answers that apply.

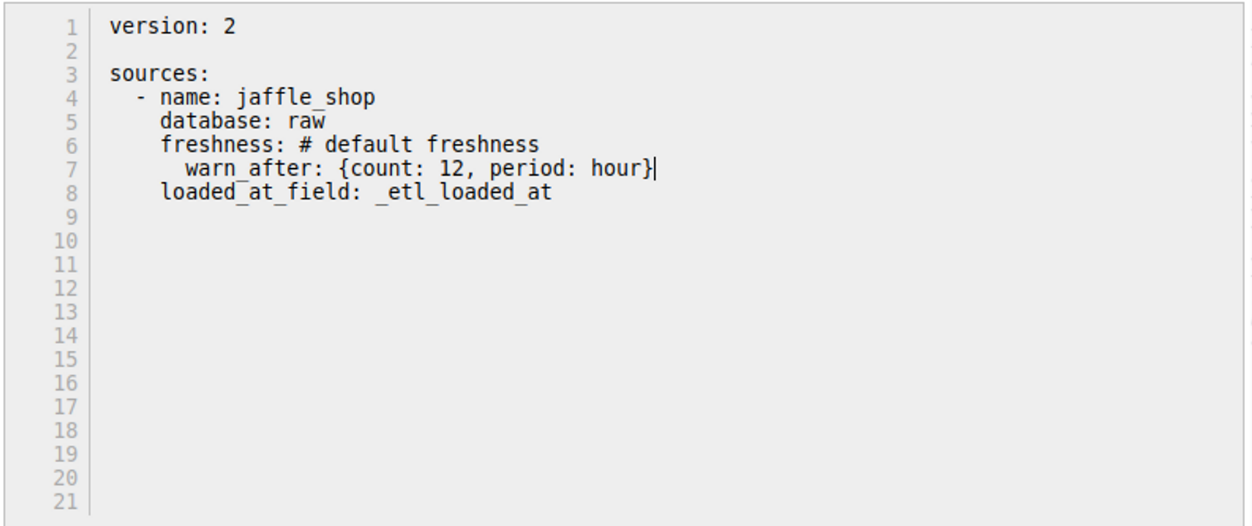
* **A**



* **B**



* **C**



* **D**



**This answer is correct.**

Answer: B,C

To configure sources to snapshot freshness information, add a freshness block to your source and loaded\_at\_field to your table declaration:

version: 2

sources:

- name: jaffle\_shop

database: raw

freshness: # default freshness

warn\_after: {count: 12, period: hour}

error\_after: {count: 24, period: hour}

loaded\_at\_field: \_etl\_loaded\_at

tables:

- name: orders

freshness: # make this a little more strict

warn\_after: {count: 6, period: hour}

error\_after: {count: 12, period: hour}

- name: customers # this will use the freshness defined above

- name: product\_skus

freshness: null # do not check freshness for this table

In the freshness block, one or both of warn\_after and error\_after can be provided. If neither is provided, then dbt will not calculate freshness snapshots for the tables in this source.

Additionally, 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.

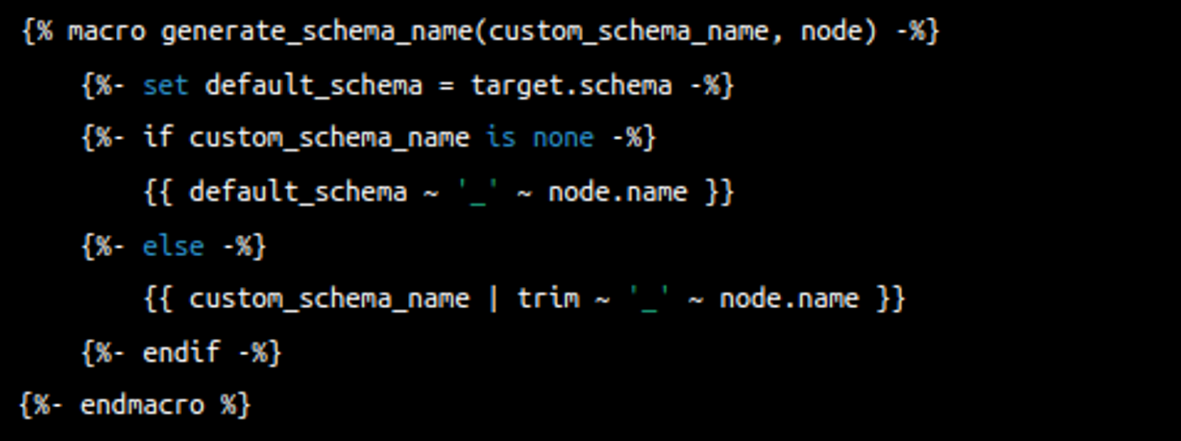
Reference: <https://docs.getdbt.com/docs/build/sources#declaring-source-freshness>

### QUESTION 43 OF 65

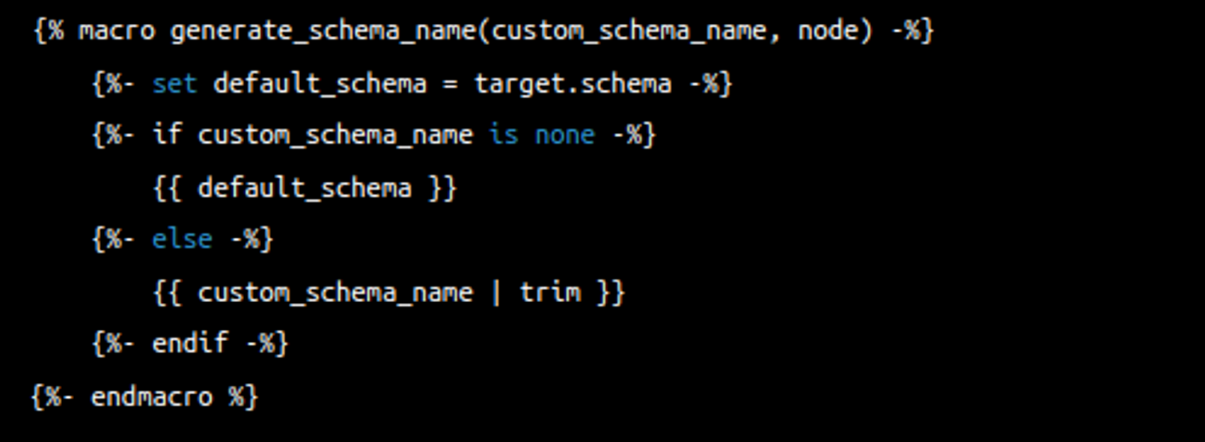
**You are an Analytics Engineer working on a dbt project and have implemented custom schemas. Upon running your dbt models, you notice that it concatenates the custom schema name with the target schema name. According to the dbt documentation, dbt uses the generate\_schema\_name macro to determine the schema where a model should be built, and by default, this macro concatenates the custom schema name with the target schema name. You want to customize the generate\_schema\_name macro to create a macro that returns either the custom schema name (if provided) or the default target schema without concatenating them. Choose the correct macro implementation from the options below:**

Choose only ONE best answer.

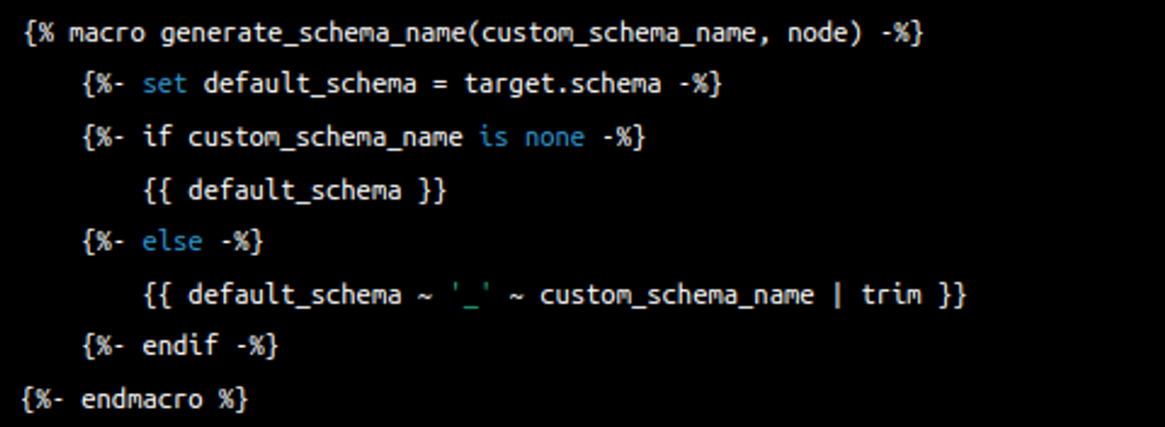
**A**



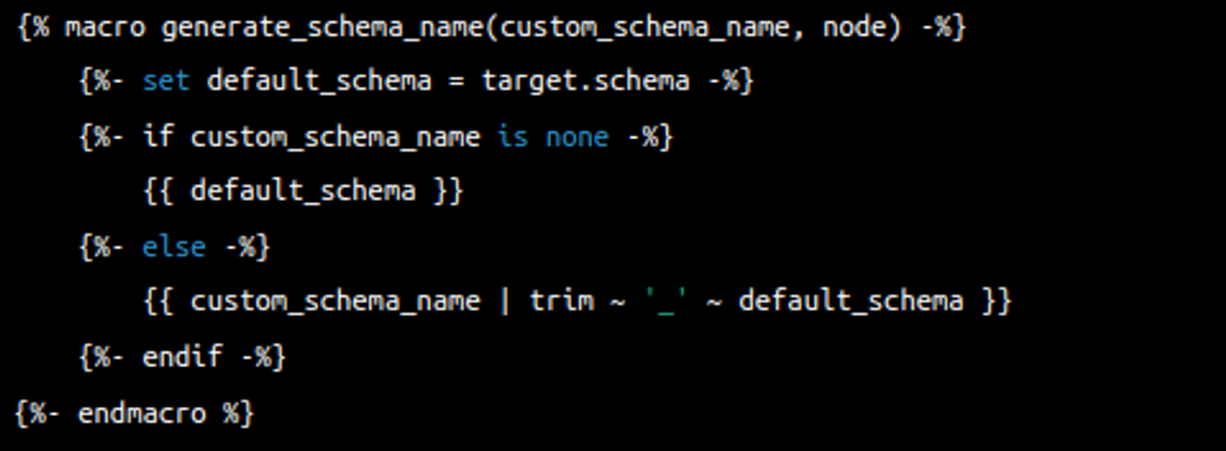
**B**



**C**



**D**



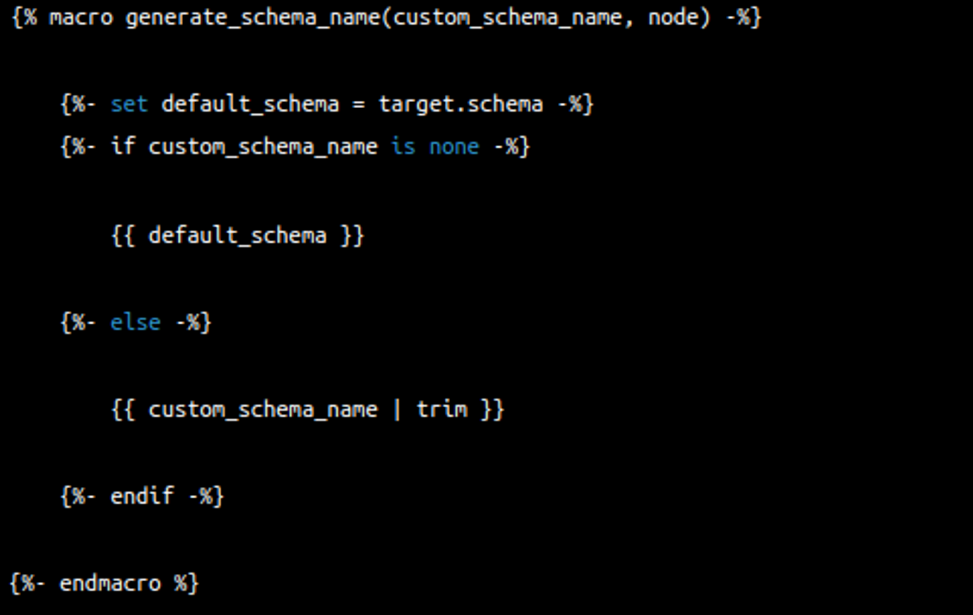
**This answer is correct.**

Answer: B

Explanation:

dbt uses a macro called **generate\_schema\_name** to determine the name of the schema that a model should be built in. By default, this macro concatenates the custom schema name with the target schema name. If no custom schema name is specified, the model is built in the target schema.

However, the following custom implementation of the **generate\_schema\_name** macro doesn't concatenate the custom schema name with the target schema name:



This custom macro returns either the custom schema name (if provided) or the default target schema. If you want to change the way dbt generates schema names, you can create your own macro named **generate\_schema\_name** in your dbt project. dbt will always use the macro in your dbt project instead of the default macro.

NOTE: If a package installed in your project includes its own **generate\_schema\_name** macro, dbt will ignore it, giving priority to the custom macro defined in your dbt project.

### QUESTION 44 OF 65

**You are a data engineer working on a dbt project that has a package installed with its own implementation of the generate\_schema\_name macro. Your goal is to create a custom implementation of the generate\_schema\_name macro in your dbt project that returns either the custom schema name (if provided) or the default target schema without concatenating them. What should you do to ensure your custom implementation takes precedence over the package's implementation?**

Choose only ONE best answer.

**A**

Modify the package's implementation of the generate\_schema\_name macro to match your custom implementation.

**B**

Remove the package's implementation of the generate\_schema\_name macro and create a custom implementation in your dbt project.

**C**

Create a custom implementation of the generate\_schema\_name macro in your dbt project without making any changes to the package.

**D**

Define an environment variable to control which implementation of the generate\_schema\_name macro should be used in your dbt project.

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

Answer: c.  Create a custom implementation of the generate\_schema\_name macro in your dbt project without making any changes to the package.

Explanation:

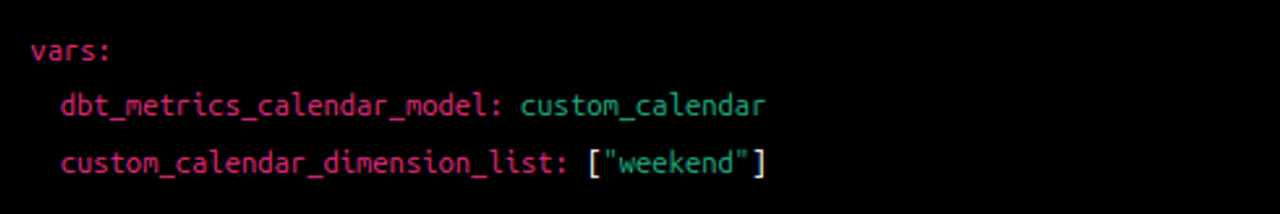
dbt uses a macro called **generate\_schema\_name** to determine the name of the schema that a model should be built in. By default, this macro concatenates the custom schema name with the target schema name. If no custom schema name is specified, the model is built in the target schema.

**NOTE: If a package installed in your project includes its own generate\_schema\_name macro, dbt will ignore it, giving priority to the custom macro defined in your dbt project.**

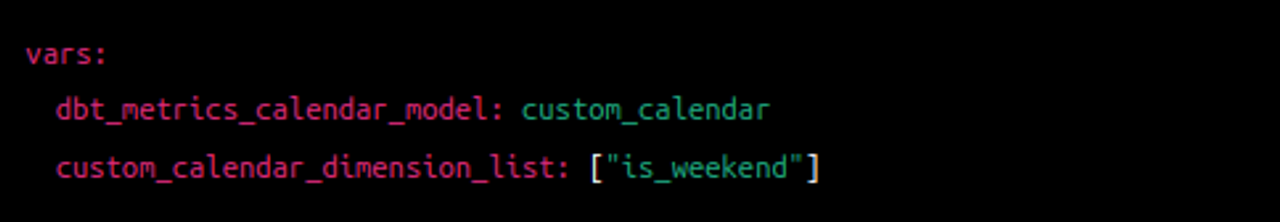
**An engineer is troubleshooting an issue with their dbt metrics package and wants to use a custom calendar table with an additional "is\_weekend" column. Which code implementation should be added to the "dbt\_project.yml" file to correctly set up the custom calendar? B**

Choose only ONE best answer.

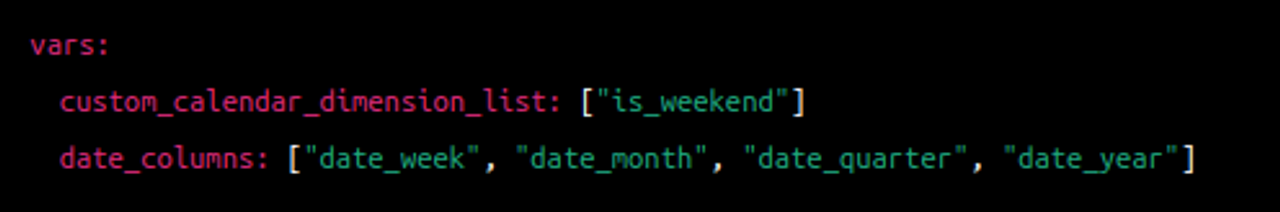
**A**



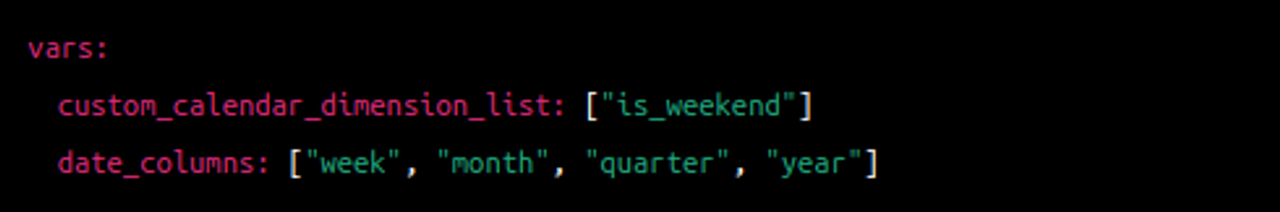
**B**



**C**



**D**



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

Answer: B

Explanation:

The dbt metrics package comes with a default basic calendar table for dbt runs. However, if you wish to use a custom calendar table, there are certain requirements that need to be met:

1. The custom calendar table must have a "date\_day" column.
2. It should also include columns for week, month, quarter, and year (or equivalent) with the "date\_" prefix.
3. Any additional date columns in the custom calendar table must have the "date\_" prefix as well.

To use the custom calendar table, you need to update the "dbt\_project.yml" file. Specifically, you should change the value of the "dbt\_metrics\_calendar\_model" variable to point to your custom calendar model.

Additionally, the text provides an example of how to add a custom column to the calendar dimension list. In this case, a custom column named "is\_weekend" is added. To achieve this, you would include the following code in the "dbt\_project.yml" file:

vars:

custom\_calendar\_dimension\_list: ["is\_weekend"]

This code snippet defines a variable called "custom\_calendar\_dimension\_list" and assigns it a list containing the "is\_weekend" custom column. This will be used by the dbt metrics package in conjunction with the custom calendar table.

**What are the requirements for a custom calendar table in the dbt metrics package?**

Choose only ONE best answer.

**A**

The custom calendar table must have a "day\_date" column and columns for week, month, quarter, and year without any prefix.

**B**

The custom calendar table must have a "date\_day" column and columns for week, month, quarter, and year with the "date\_" prefix.

**C**

The custom calendar table must have a "date\_day" column and columns for day, week, month, and year without any prefix.

**D**

The custom calendar table must have a "day\_date" column and columns for day, week, month, and year with the "date\_" prefix.

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

Answer: B

Explanation:

The dbt metrics package comes with a default basic calendar table for dbt runs. However, if you wish to use a custom calendar table, there are certain requirements that need to be met:

1. The custom calendar table must have a "date\_day" column.
2. It should also include columns for week, month, quarter, and year (or equivalent) with the "date\_" prefix.
3. Any additional date columns in the custom calendar table must have the "date\_" prefix as well.

To use the custom calendar table, you need to update the "dbt\_project.yml" file. Specifically, you should change the value of the "dbt\_metrics\_calendar\_model" variable to point to your custom calendar model.

Additionally, the text provides an example of how to add a custom column to the calendar dimension list. In this case, a custom column named "is\_weekend" is added. To achieve this, you would include the following code in the "dbt\_project.yml" file:

vars:

custom\_calendar\_dimension\_list: ["is\_weekend"]

This code snippet defines a variable called "custom\_calendar\_dimension\_list" and assigns it a list containing the "is\_weekend" custom column. This will be used by the dbt metrics package in conjunction with the custom calendar table.

QUESTION 47 OF 65

**Which variable in the "dbt\_project.yml" file should be updated to point to your custom calendar model?**

Choose only ONE best answer.

**A**

calendar\_model

**B**

custom\_calendar\_model

**C**

dbt\_calendar\_model

**D**

dbt\_metrics\_calendar\_model

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

Correct answer: D

Explanation:

The dbt metrics package comes with a default basic calendar table for dbt runs. However, if you wish to use a custom calendar table, there are certain requirements that need to be met:

1. The custom calendar table must have a "date\_day" column.
2. It should also include columns for week, month, quarter, and year (or equivalent) with the "date\_" prefix.
3. Any additional date columns in the custom calendar table must have the "date\_" prefix as well.

To use the custom calendar table, you need to update the "dbt\_project.yml" file. Specifically, you should change the value of the "dbt\_metrics\_calendar\_model" variable to point to your custom calendar model.

Additionally, the text provides an example of how to add a custom column to the calendar dimension list. In this case, a custom column named "is\_weekend" is added. To achieve this, you would include the following code in the "dbt\_project.yml" file:

vars:

custom\_calendar\_dimension\_list: ["is\_weekend"]

This code snippet defines a variable called "custom\_calendar\_dimension\_list" and assigns it a list containing the "is\_weekend" custom column. This will be used by the dbt metrics package in conjunction with the custom calendar table.

QUESTION 48 OF 65

**In the context of the dbt metrics package, what prefix should be used for any additional date columns in a custom calendar table?**

Choose only ONE best answer.

**A**

date\_

**B**

custom\_

**C**

additional\_

**D**

extra\_

**This answer is correct.**

Correct answer: A

Explanation:

The dbt metrics package comes with a default basic calendar table for dbt runs. However, if you wish to use a custom calendar table, there are certain requirements that need to be met:

1. The custom calendar table must have a "date\_day" column.
2. It should also include columns for week, month, quarter, and year (or equivalent) with the "date\_" prefix.
3. Any additional date columns in the custom calendar table must have the "date\_" prefix as well.

### QUESTION 50 OF 65

**Before using the macros to query a metrics, which model do you need to run?**

Choose only ONE best answer.

**A**

dbt\_metrics\_custom\_calendar

**B**

dbt\_metrics\_default\_calendar

**C**

dbt\_metrics\_initial\_setup

**D**

dbt\_metrics\_grain\_calendar

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

Answer: B

Explanation:

The dbt\_metrics package enables previewing and verifying metrics within dbt before deployment. Install the package by adding it to your packages.yml file and running dbt deps. Running the dbt\_metrics\_default\_calendar model, required for the macros. To calculate metrics, use the metrics.calculate macro with specified metrics, grain, and dimensions.

### QUESTION 53 OF 65

**You are working on a dbt project that involves data processing on Databricks and GCP. Which of the following frameworks is used by Databricks and GCP's Dataproc for Python processing in dbt?**

Choose only ONE best answer.

**A**

Snowpark

**B**

Presto

**C**

PySpark

**D**

Spark

**This answer is correct.**

Answer: C

Explanation: PySpark is the processing framework used by Databricks and GCP's Dataproc for Python processing in dbt. Snowflake uses its own framework, Snowpark, which is similar to PySpark. Presto and Spark are other processing frameworks that can be used with various data platforms, but are not used by Databricks or GCP's Dataproc for Python processing in dbt.

Option A is incorrect because Snowpark is the framework used by Snowflake, not Databricks or GCP's Dataproc, for Python processing in dbt.

Option B is incorrect because Presto is a processing framework that can be used with various data platforms, but is not used by Databricks or GCP's Dataproc for Python processing in dbt.

Option D is incorrect because Spark is a processing framework that can be used with various data platforms, but is not used by Databricks or GCP's Dataproc for Python processing in dbt.

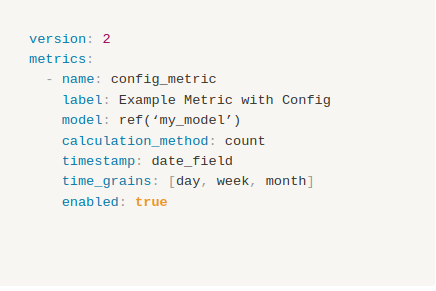
Reference: https://docs.getdbt.com/docs/build/python-models

### QUESTION 55 OF 65

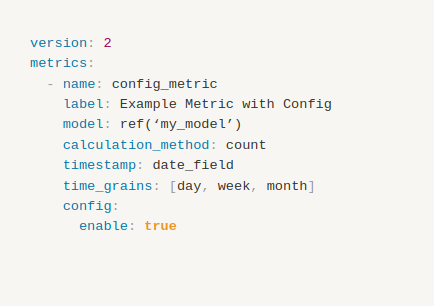
**Examine the following code snippets. Which one correctly specifies a metric configuration in the models/metrics.yml file?**

Choose only ONE best answer.

**A**



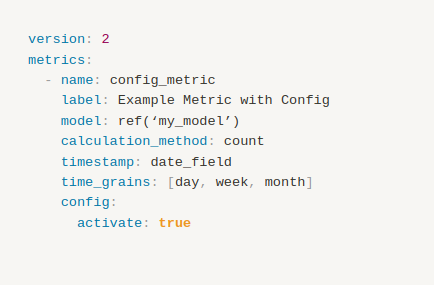
**B**



**C**



**D**



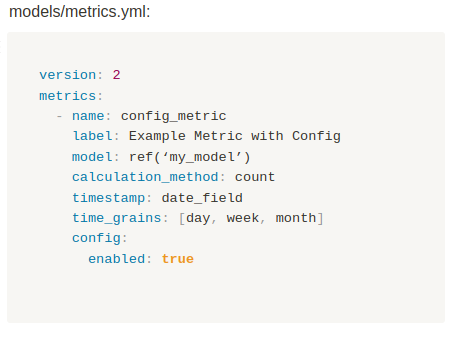
**This answer is correct.**

Answer: C

Explanation dbt metrics support specific configurations, such as "enabled" and "treat\_null\_values\_as\_zero." The "enabled" configuration allows users to activate or deactivate a metric, while the "treat\_null\_values\_as\_zero" configuration specifies how metrics with no observations are handled.

Metric configurations can be defined in the metric YAML file or the dbt\_project.yml file for multiple metrics. The example YAML files for both configuration methods are provided below.



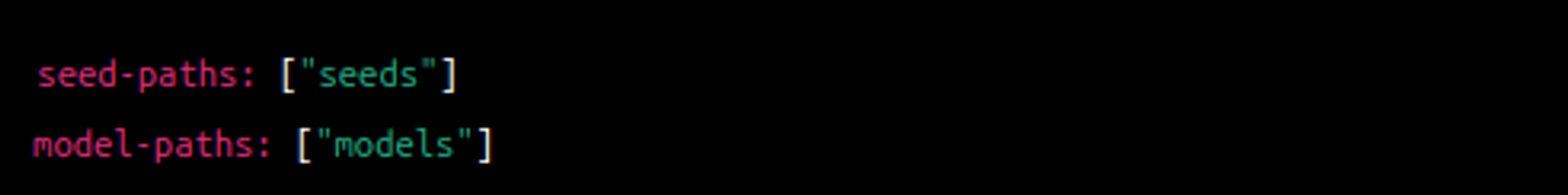


### QUESTION 56 OF 65

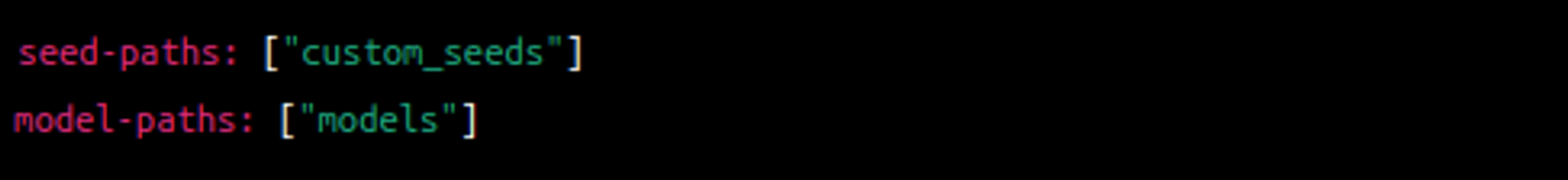
**You are working on a dbt project and want to troubleshoot an issue with the seed files. Currently, the seed files are not being recognized by dbt, and you are trying to find the reason. You have been given the following dbt\_project.yml file configurations to analyze, which of this would cause an error in dbt?**

Choose only ONE best answer.

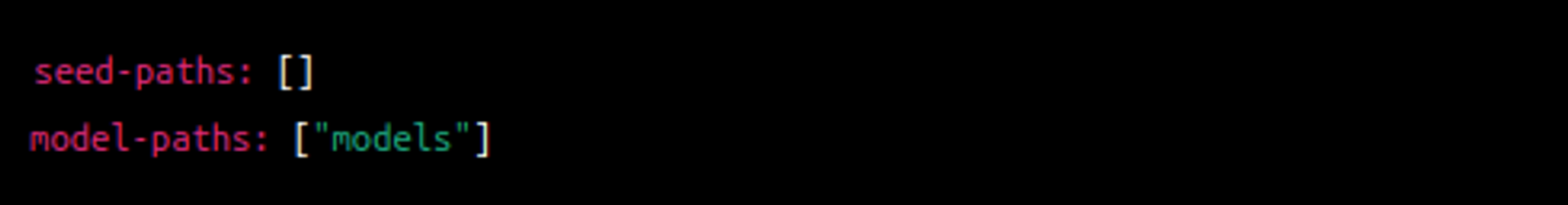
**A**



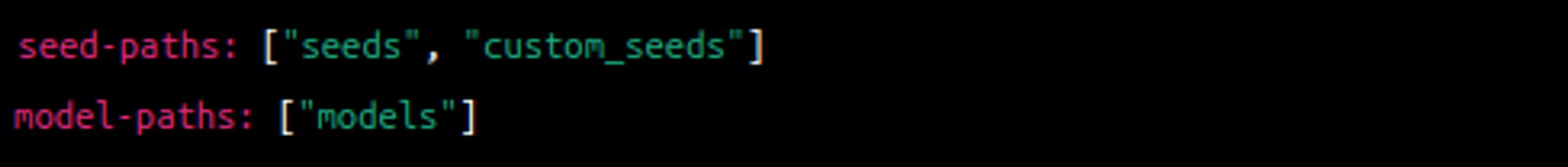
**B**



**C**



**D**

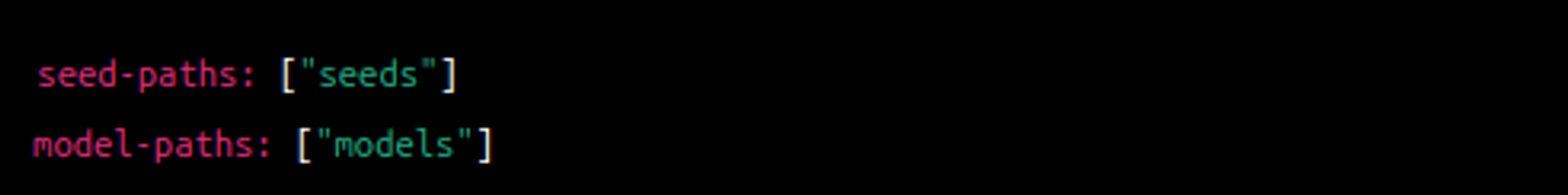


**This answer is correct.**

Answer: C)

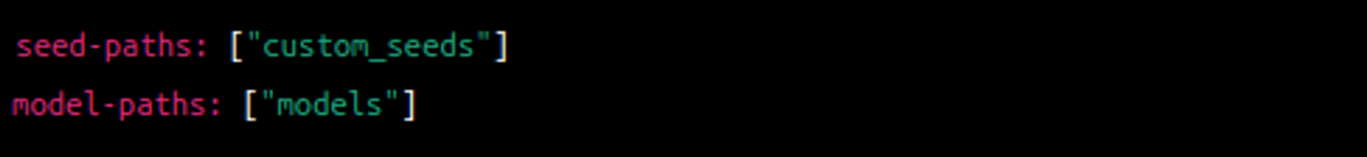
Explanation:

Option A:



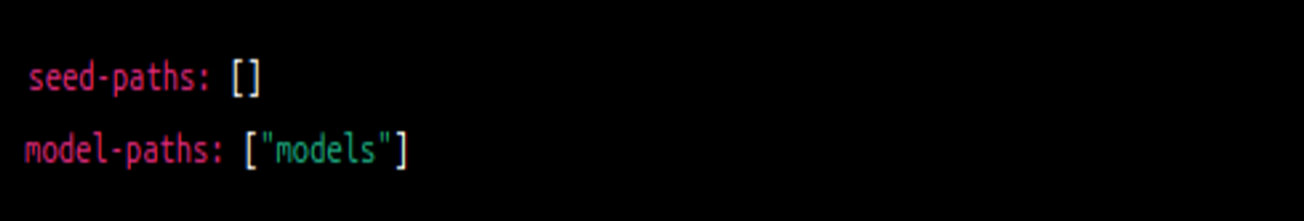
Explanation: In this configuration, dbt will recognize seed files located in the "seeds" directory and model files located in the "models" directory. This is the default configuration and should work properly.

Option B:



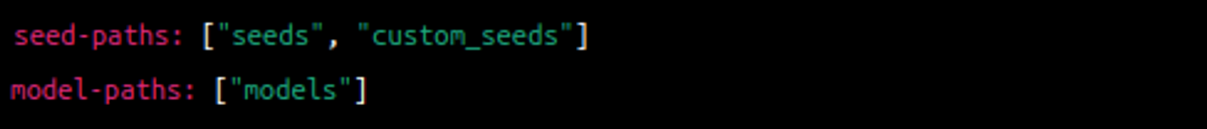
Explanation: In this configuration, dbt will recognize seed files located in the "custom\_seeds" directory instead of the default "seeds" directory. Model files are still located in the "models" directory. This configuration will work if the seed files are placed in the "custom\_seeds" directory.

Option C:



Explanation: In this configuration, no directories are specified for seed files, which means dbt will not recognize any seed files. Model files are located in the "models" directory. This configuration is the cause of the issue, as dbt does not know where to find the seed files.

Option D:



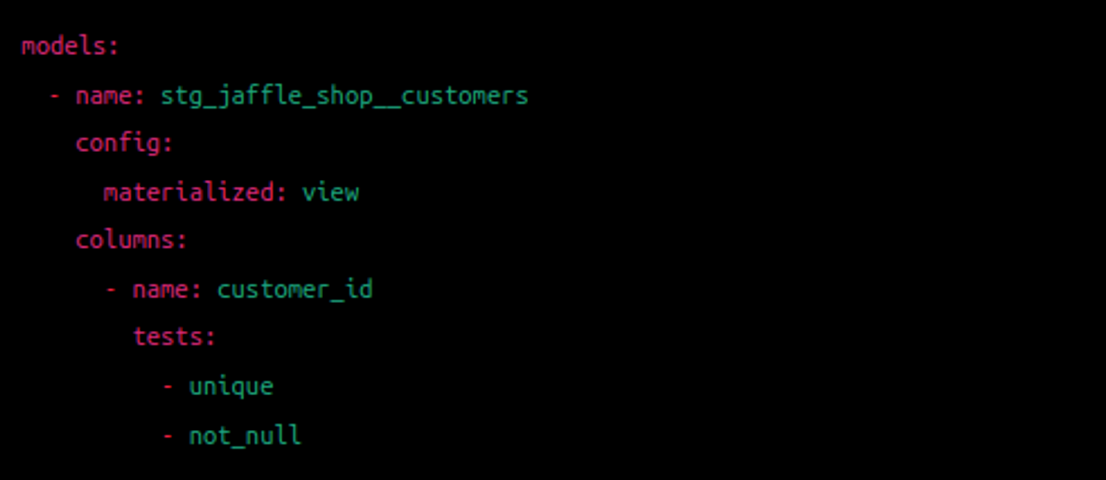
Explanation: In this configuration, dbt will recognize seed files located in either the "seeds" directory or the "custom\_seeds" directory. Model files are located in the "models" directory. This configuration will work if the seed files are placed in either the "seeds" or "custom\_seeds" directory.

### QUESTION 57 OF 65

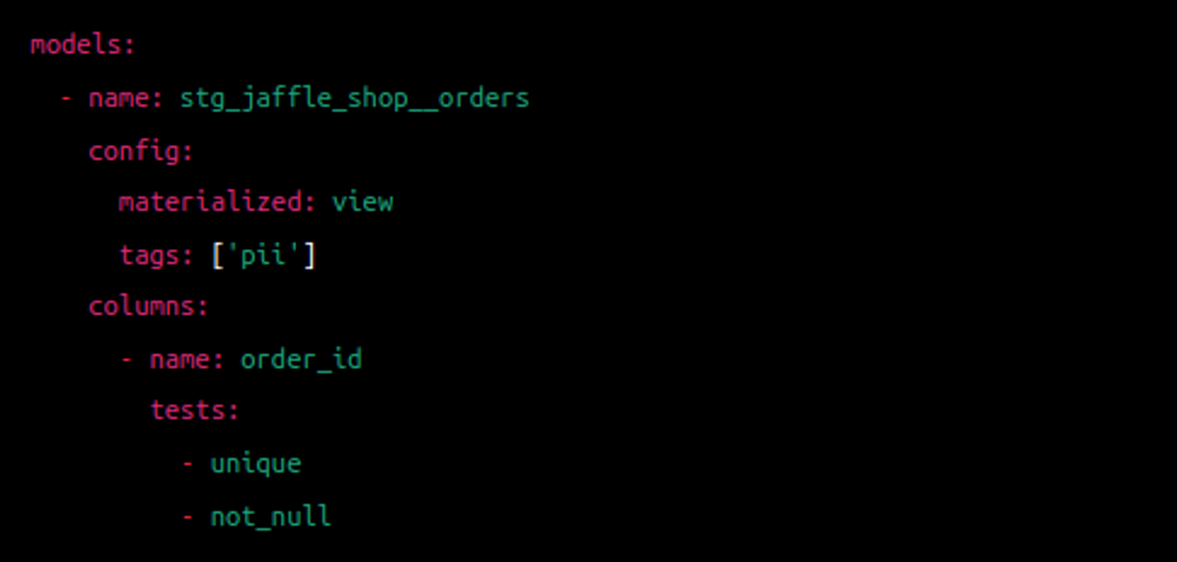
**Identify the incorrect schema.yml configuration in a dbt project.**

Choose only ONE best answer.

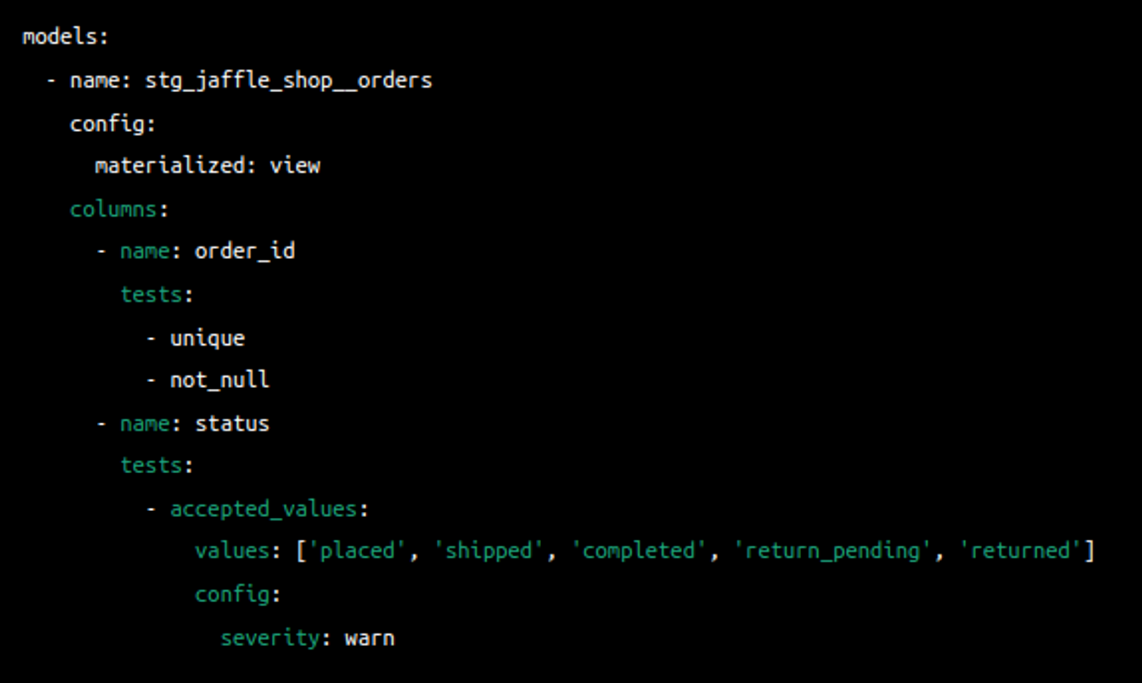
**A**



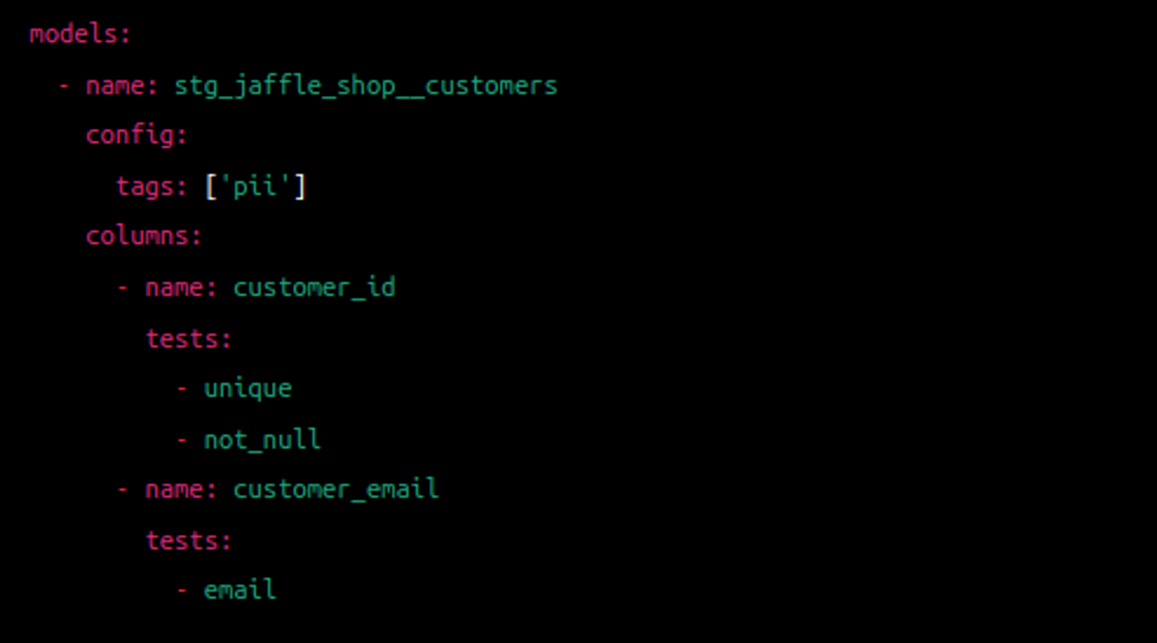
**B**



**C**



**D**



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

Answer: B

Explanation:

Option A is a correct configuration, as it sets the materialization to 'view' for the stg\_jaffle\_shop\_\_customers model and defines tests for the customer\_id column.

Option B is incorrect because it sets both materialization and tags within the same config block. Tags should be defined in a separate config block.

Option C is a correct configuration, as it sets the materialization to 'view' for the stg\_jaffle\_shop\_\_orders model and defines tests for the order\_id and status columns, with the severity set to 'warn' for the accepted\_values test.

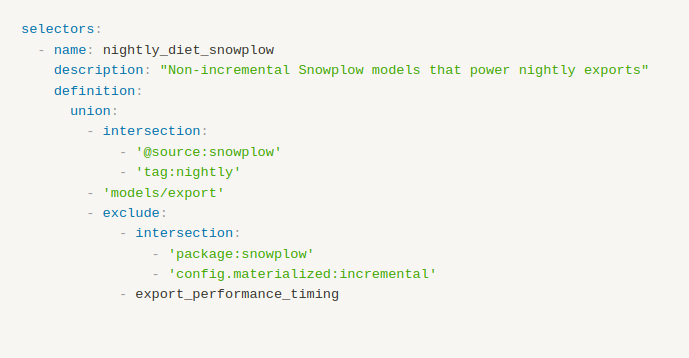
Option D is a correct configuration, as it defines tags for the stg\_jaffle\_shop\_\_customers model and tests for the customer\_id and customer\_email columns. Note that the email test is not provided in the original schema.yml, but it is a valid test to include.

### QUESTION 58 OF 65

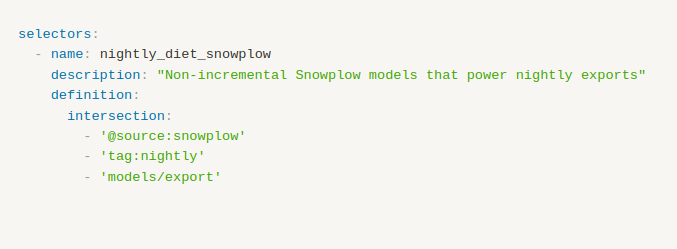
**You are a data engineer working on a dbt project. You want to select only the non-incremental Snowplow models that power nightly exports using YAML selectors. Which of the following selectors would achieve this goal?**

Choose only ONE best answer.

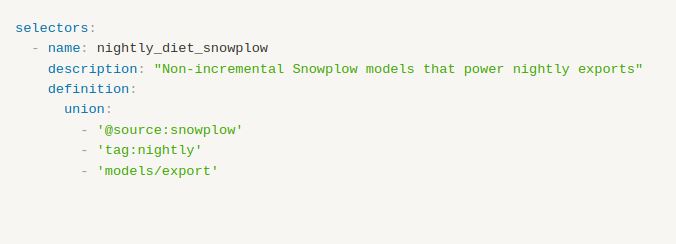
**A**



**B**



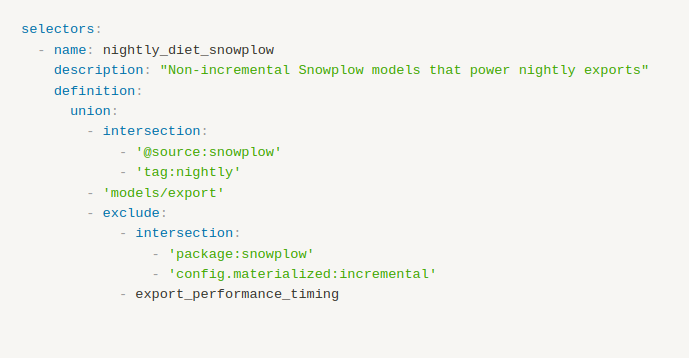
**C**



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

Answer: A

Explanation:  
  
Selector A does the following:  
- Selects all models with the Snowplow source and the nightly tag using the intersection of '@source:snowplow' and 'tag:nightly'.  
- Selects all models in the 'models/export' directory.  
- Unites the results from steps 1 and 2 using the 'union' clause.  
- Excludes all incremental models in the Snowplow package using the 'exclude' clause with the intersection of 'package:snowplow' and 'config.materialized:incremental'.  
- Excludes the 'export\_performance\_timing' model.



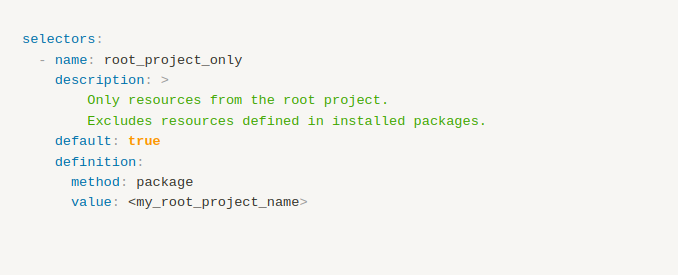
Reference: <https://docs.getdbt.com/reference/node-selection/yaml-selectors>

### QUESTION 59 OF 65

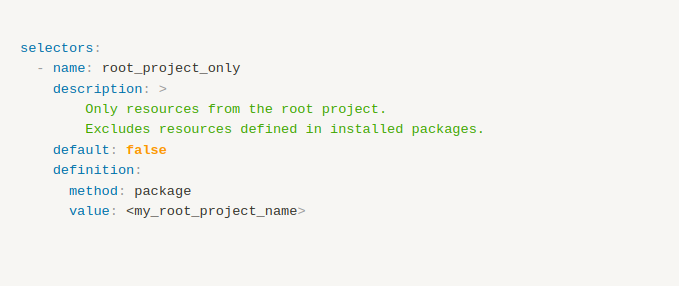
**You are working on a dbt project and want to create a default selector that only selects resources defined in your root project. Which of the following YAML configurations achieves this goal?**

Choose only ONE best answer.

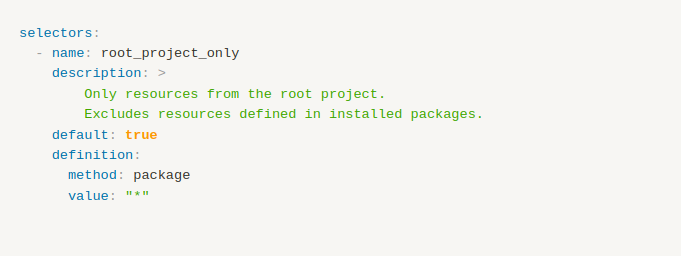
**A**



**B**



**C**



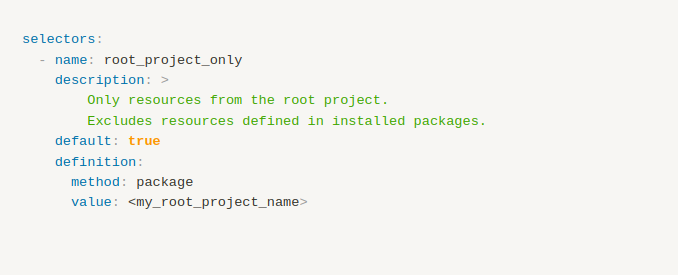
**This answer is incorrect. The correct answer is 'A'**

Answer: A

Explanation: Option A correctly sets the default property to true and specifies the package method with the root project name as the value. This configuration will only select resources defined in the root project when no other selection criteria are defined.

This configuration sets the selector as the default one (default: true) and specifies that it should only include resources from the root project by using the method: package and value: <my\_root\_project\_name> options. Replace <my\_root\_project\_name> with the actual name of your root project.

The package method is used to select models defined within the root project or an installed dbt package.



Reference: <https://docs.getdbt.com/reference/node-selection/yaml-selectors>