Quiz answers with explanations

QUESTION

Which of the following is a correct Jinja expression?

- **A: {{ my_variable }}**
- B: { my_variable } • C: [my_variable] • D: (my_variable)

CORRECT ANSWER: **A**

EXPLANATION

Explanation: In Jinja, expressions are denoted by double curly braces {{ }}. This syntax is used to output strings, reference variables, and call macros. The correct way to reference a variable in Jinja is to use the double curly braces, followed by the name of the variable.

Reference: https://docs.getdbt.com/docs/build/jinja-macros

QUESTION

What are dbt seeds?

- A: SQL scripts used to migrate data from one database to another • B: Files containing raw data that have not yet been transformed
- **C: CSV files with static data that can be loaded into your data platform with dbt**
- D: A set of predefined macros used for common transformations

CORRECT ANSWER: **C**

EXPLANATION

Explanation: dbt seeds are CSV files with static data that can be loaded into your data platform with dbt. They are used for providing initial data or reference data that does not change frequently, such as a list of countries or product categories.

Option A is not true as dbt seeds are not SQL scripts used for database migration.

Option B is not true as dbt seeds are not raw data but rather preprocessed data.

Option D is not accurate as macros are not used in dbt seeds.

Reference: https://docs.getdbt.com/docs/build/projects

QUESTION

You are working on a dbt project and have created a seed file to provide static reference data for your models. Which of the following statements accurately describes the benefit of storing seeds in your dbt repository?

- A: Seeds can be used to store dynamic data that changes frequently and requires frequent updates to your models.
- B: Seeds can be easily loaded into your data warehouse without requiring the use of any SQL or ETL tools.
- **C: Seeds are stored as CSV files in your dbt repository, which means they are version controlled and code reviewable.**
- D: Seeds can be used to store large amounts of unstructured data that cannot be easily loaded into your data warehouse.

CORRECT ANSWER: **C**

EXPLANATION

Explanation: Seeds are stored as CSV files in your dbt repository, which means they are version controlled and code reviewable. This makes it easy to track changes to the seed data over time and ensure that the data is consistent across different versions of your dbt project.

Option A is incorrect because seeds are not best suited for dynamic data that changes frequently, but rather for static data that changes infrequently.

Option B is incorrect because seeds still require the use of the dbt seed command to load the data into your data warehouse, and may require additional SQL or ETL tools to transform the data.

Option D is incorrect because seeds are not best suited for storing large amounts of unstructured data, but rather for storing structured data in CSV format.

Reference: https://docs.getdbt.com/docs/build/seeds

QUESTION

What are the prerequisites for dbt Python models?

- A: Using an adapter for a data platform that supports SQL.
- B: Using a data platform that supports Python.
- **C: Using an adapter for a data platform that supports a fully featured Python runtime.**
- D: Using a data platform that supports a fully featured Python runtime.

CORRECT ANSWER: **C**

EXPLANATION

Explanation: The prerequisites for dbt Python models include using an adapter for a data platform that supports a fully featured Python runtime. This means that the adapter must allow the execution of arbitrary Python code, rather than just simple calculations or transformations.

Option A is incorrect because SQL support is not required for dbt Python models specifically.

Option B is incorrect because it is not enough for the data platform to support Python; it must support a fully featured Python runtime.

Option D is incorrect because it is not enough for the data platform to support a fully featured Python runtime; you must also use an adapter that allows the execution of arbitrary Python code.

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

QUESTION

What is the purpose of the profiles.yml file in dbt?

- A: To store sensitive credentials
- **B: To connect to the data warehouse**
- C: To avoid checking sensitive credentials into version control
- D: To store project details

CORRECT ANSWER: **B**

EXPLANATION

Explanation: The Purpose Of The Profiles.Yml File In Dbt Is To Connect To The Data Warehouse. It Contains All The Details Needed To Connect To The Warehouse, Such As Credentials And Connection Details. The File Is Generally Located Outside Of The Dbt Project To Avoid Sensitive Credentials Being Checked Into Version Control.

Reference: Https://Docs.Getdbt.Com/Docs/Get-Started/Connection-Profiless

QUESTION

As a data developer working with dbt, what happens when you execute the "dbt run" command, and how does dbt build the model data warehouse?

- A: The "dbt run" command creates a temporary view in the source database and pulls data into the target data warehouse.
- B: The "dbt run" command triggers a Python script that runs a series of SQL queries on the source database to extract and transform data.
- **C: The "dbt run" command builds the model data warehouse by wrapping it in a "create view as" or "create table as" statement.**
- D: The "dbt run" command builds the model data warehouse by creating a temporary table in the target database and loading data into it.

CORRECT ANSWER: **C**

EXPLANATION

Explanation: When you execute the "dbt run" command, dbt builds the model data warehouse by wrapping it in a "create view as" or "create table as" statement. This creates a view or table in the target data warehouse that contains the results of the SQL statement defined in the model.

Option A is incorrect because dbt does not create a temporary view in the source database or pull data into the target data warehouse when running "dbt run".

Option B is incorrect because "dbt run" does not trigger a Python script or run a series of SQL queries to extract and transform data from the source database.

Option D is incorrect because "dbt run" does not create a temporary table in the target database and load data into it.

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

QUESTION

What are the four materializations that ship with dbt?

- A: The four materializations that ship with dbt are view, table, incremental, and ephemeral. There are no additional options for creating custom materializations.
- **B: The four materializations that ship with dbt are view, table, incremental, and ephemeral. You can also create your own custom materializations using advanced features of dbt.**
- C: The four materializations that ship with dbt are view, table, incremental, and ephemeral. You can also create your own custom materializations using SQL.
- D: The four materializations that ship with dbt are view, table, incremental, and ephemeral. You can also create your own custom materializations by requesting them from dbt support.

CORRECT ANSWER: **B**

EXPLANATION

The Four Materializations That Ship With Dbt Are View, Table, Incremental, And Ephemeral. You Can Also Create Your Own Custom Materializations Using Advanced Features Of Dbt.

Dbt Ships With Four Materializations: View, Table, Incremental And Ephemeral. Check Out The Documentation On Materializations For More Information On Each Of These Options. You Can Also Create Your Own Custom Materializations, If Required However This Is An Advanced Feature Of Dbt.

Reference: Https://Docs.Getdbt.Com/Docs/Get-Started/Getting-Started-Dbt-Core

QUESTION

In the following dbt code, what is causing the error "doc block not found"? Example; "' version: 2 models: name: the model name description: '{{ doc("doc_block_name") }}'

- A: The "doc" macros has not been installed run “DBT devs”
- B: The model name is incorrect
- **C: The doc block name is incorrect**
- D: The quotation marks around the doc block name are missing
- E: The "doc" macros is not properly defined

CORRECT ANSWER: **C**

EXPLANATION

Reference: Https://Docs.Getdbt.Com/Docs/Get-Started/Getting-Started/Building-Your-First-Project/Test-And-Document-Your-Project

QUESTION

You are reviewing the schema.yml file for a dbt project and notice that a source table has been defined, but no column descriptions have been provided. What is the benefit of adding descriptions to the source table columns?

- A: It will improve the performance of the queries that reference the source.
- B: It will allow you to run tests on the source table, to ensure the quality of the data.
- C: It will help you test your assumptions about the source data.
- **D: It will provide additional metadata about the source table, which will be rendered on the documentation site.**

CORRECT ANSWER: **D**

EXPLANATION

Explanation: When you define a source table in the schema.yml file, you can add descriptions to the columns of the table. These descriptions will be rendered on the documentation site, and will provide additional metadata about the source table.

Option D is the correct benefit of adding descriptions to the source table columns. Option A, B, and C are incorrect because they describe benefits that are not related to the documentation site.

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

QUESTION

Which two strategies are used by dbt to determine if a row in a database has been altered?

- A: Hash and Map
- B: Check and Transform
- C: Snapshot and Check

- **D: Timestamp and Check**

CORRECT ANSWER: **D**

EXPLANATION

Explanation: Snapshot "strategies" are used by dbt to determine if a row in a database has been altered. Timestamp and Check are two such strategies. The timestamp strategy involves comparing a row's updated_at timestamp to the previous snapshot's updated_at timestamp to determine if the row has been altered.

The check strategy involves comparing a row's hash value to the previous snapshot's hash value to determine if the row has been altered. By using these strategies, dbt can create a historical record of changes to data in a database, making it easier to track the evolution of data over time.

Reference: https://docs.getdbt.com/docs/build/snapshots

QUESTION

What is the purpose of the logs file in the dbt project folder?

- A: To show the select statements that are running during dbt runs
- B: To track the python logging happening within the project
- C: To monitor the performance of dbt Core
- **D: All of the above**

CORRECT ANSWER: **D**

EXPLANATION

Explanation: The Logs File In The Dbt Project Folder Is Used To See How Dbt Core Logs All Of The Action Happening Within Your Project. It Shows The Select Statements That Are Running And The Python Logging Happening When Dbt Runs.

Reference: Https://Docs.Getdbt.Com/Docs/Get-Started/Getting-Started-Dbt-Core

QUESTION

How can increasing the number of threads in dbt minimize the run time of your project?

- A: By creating a directed acyclic graph (DAG) of links between models.
- **B: By Increasing the number of paths through the graph dbt may work on at once**
- C: By increasing the number of models dbt starts building
- D: By building models in a sequential order

CORRECT ANSWER: **B**

EXPLANATION

Explanation:

The Actual Number Of Models dbt Can Work On Will Likely Be Constrained By The Available Paths Through The Dependency Graph.

There'S No Set Limit Of The Maximum Number Of Threads You Can Set – While Increasing The Number Of Threads Generally Decreases Execution Time, There Are A Number Of Things To Consider: Increasing The Number Of Threads Increases The Load On Your Warehouse, Which May Impact Other Tools In Your Data Stack.

For Example, If Your Bi Tool Uses The Same Compute Resources As Dbt, Their Queries May Get Queued During A Dbt Run. The Number Of Concurrent Queries Your Database Will Allow You To Run May Be
A Limiting Factor In How Many Models Can Be Actively Built

Some Models May Queue While Waiting For An Available Query Slot. Generally The Optimal Number Of Threads Depends On Your Data Warehouse And Its Configuration. It'S Best To Test Different Values To Find The Best Number Of Threads For Your Project.

dbt Recommend Setting This To 4 To Start With. You Can Use A Different Number Of Threads Than The Value Defined In Your Target By Using The --Threads Option When Executing A Dbt Command.

Reference: Https://Docs.Getdbt.Com/Docs/Get-Started/Connection-Profiles

QUESTION

What happens when dbt is unable to find a definition for a variable?

- A: A runtime error is raised.
- **B: A compilation error is raised.**
- C: The default value for the variable is used.
- D: The variable is ignored.

CORRECT ANSWER: **B**

EXPLANATION

Explanation: In dbt, if dbt is unable to find a definition for a variable, a compilation error is raised. Therefore, option B is the correct answer.

Option A is incorrect because a runtime error is not raised when dbt is unable to find a definition for a variable.

Option C is incorrect because if dbt is unable to find a definition for a variable, the default value for the variable is not used.

Option D is incorrect because if dbt is unable to find a definition for a variable, the variable is not ignored.

Reference: https://docs.getdbt.com/docs/build/project-variables

QUESTION

You are a data scientist working on a data pipeline and you want to use incremental models in dbt. What information do you need to provide to dbt to use incremental models?

- A: The model name and the name of the database adapter being used.
- **B: The unique key of the model and how to filter the rows on incremental runs.**
- C: The path to the SQL file containing the select statement.
- D: The SQL query used to create the source data.

CORRECT ANSWER: **B**

EXPLANATION

Explanation: To use incremental models in dbt, 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). This is done by configuring the incremental strategy in the dbt project file, using the unique_key and strategy keys in the incremental block.

The unique_key defines the columns that make up the unique key of the model, while the strategy defines how to filter the incoming data to only include new rows.

The strategy can be one of "timestamp", "check", "delete", "merge", "insert_overwrite", or "full_refresh".

By configuring these options, dbt knows how to incrementally update the target table, improving the performance and efficiency of updating data.

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

QUESTION

Which of the following is true about hooks in dbt for seeds?

- A: Hooks cannot be used with seeds.
- B: Only pre-hooks and post-hooks can be used with seeds.
- **C: All types of hooks can be used with seeds.**
- D: Only on-run-start hooks and on-run-end hooks can be used with seeds.

CORRECT ANSWER: **C**

EXPLANATION

Explanation: In dbt, all types of hooks can be used with seeds, including pre-hooks, post-hooks, on-run-start hooks, and on-run-end hooks. These hooks can be configured in the dbt_project.yml file, and are useful for executing custom code before or after running seeds or performing other actions. Using hooks can help automate and streamline data workflows.

Option A is incorrect as hooks can be used with seeds.

Option B and D are incorrect because all types of hooks can be used with seeds, not just pre-hooks and post-hooks or on-run-start hooks and on-run-end hooks.

Reference: https://docs.getdbt.com/docs/build/seeds

QUESTION

What is the difference between global variables and package-specific variables in dbt projects?

- A: Global variables are defined in the dbt_project.yml file, while package-specific variables are defined in individual package files
- **B: Global variables can be accessed by all models and packages in the project, while package-specific variables are only accessible by models and packages in**

the same package

- C: Global variables can only be set on the command line, while package-specific variables can only be set in the dbt_project.yml file
- D: There is no difference between global variables and package-specific variables in dbt projects

CORRECT ANSWER: **B**

EXPLANATION

The correct answer is 'B' Answer: B) Global variables can be accessed by all models and packages in the project, while package-specific variables are only accessible by models and packages in the same package

Explanation: The main difference between global variables and package-specific variables in dbt projects is that global variables can be accessed by all models and packages in the project, while package-specific variables are only accessible by models and packages in the same package. When defining variables in the dbt_project.yml file, you can use the vars config to define global variables that can be accessed by all models, analyses, tests, and hooks in your project. When defining variables in a package file, you can use the vars config to define variables that are only accessible by models, analyses, tests, and hooks in the same package. This can be useful for encapsulating variables that are only relevant to a specific package.

Reference: https://docs.getdbt.com/docs/build/project-variables

QUESTION

Which of the following actions can be taken to debug a database error caused by a bad or wrong SQL query?

- **A: Use the error message and the compiled SQL to debug the error**
- B: Modify any models downstream of the affected model
- C: Ignore the error and continue running the program
- D: Restart the database server to help dbt self correct

CORRECT ANSWER: **A**

EXPLANATION

When Running A Bad Or Wrong Sql Query, Generally There Will Be A Database Error, Dbt Will Return This Exact Error And Any Models Downstream Of This Model Will Also Be

Skipped. To Debug This Error We Can Use The Error Message And The Compiled Sql To Debug Any Errors.

Reference: Https://Docs.Getdbt.Com/Docs/Get-Started/Getting-Started/Building-Your-First-Project/Build-Your-First-Models

QUESTION

You are using dbt Core or dbt Cloud to run your project, and you want to ensure that all of your defined models are running correctly and that your tests are passing. What command should you use?

- A: dbt run
- **B: dbt build**
- C: dbt test
- D: dbt deploy

CORRECT ANSWER: **B**

EXPLANATION

Explanation: The Dbt Build Command Is Used To Build And Test Resources Such As Models, Seeds, Snapshots, And Tests. This Ensures That All Of Your Defined Models Are Running Correctly And That Your Tests Are Passing. The Dbt Run Command Is Used To Run The Models You Defined In Your Project, While The Dbt Test Command Is Used To Execute The Tests You Defined For Your Project. The Dbt Deploy Command Is Not A Valid Command In Dbt.

Reference: Https://Docs.Getdbt.Com/Docs/Get-Started/Run-Your-Dbt-Projects

QUESTION

What is the difference between defining variables in the dbt_project.yml file and using the --vars command line option?

- A: The dbt_project.yml file is suitable for frequently changing variables, while --vars is best for variables that rarely change
- B: The dbt_project.yml file requires strict quoting, while --vars does not

- **C: The dbt_project.yml file can only be used to define global variables, while --vars can be used to set run-specific variables**
- D: There is no difference between defining variables in the dbt_project.yml file and using the --vars command line option

CORRECT ANSWER: **C**

EXPLANATION

Answer: C) The dbt_project.yml file can only be used to define global variables, while --vars can be used to set run-specific variables

Explanation: The main difference between defining variables in the dbt_project.yml file and using the --vars command line option is that the dbt_project.yml file can only be used to define global variables, while --vars can be used to set run-specific variables. When defining variables in the dbt_project.yml file, you can only define global variables that are accessible by all models, analyses, tests, and hooks in your project.

When using the --vars command line option, you can set or override variables that are defined in the dbt_project.yml file or in package-specific files, and you can do so on a per-run basis. This can be useful for setting variables that are specific to a particular run or environment.

Reference: https://docs.getdbt.com/docs/build/project-variables

QUESTION

You are working on testing a specific model in dbt. How can you use the --select flag or -s flag to only test one model at a time?

- A: dbt test --select model_name||
- B: dbt test model_name --select
- C: dbt test model_name -s
- D: dbt test -s model_name

CORRECT ANSWER: **A, D**

EXPLANATION

Explanation:

To Test One Model At A Time. We Can Also Implement This Using The --Select Flag (Or -S Flag), Followed By The Name Of The Model. Dbt Test --Select Model_Name Or Dbt Test -S

Model_Name

Reference: Https://Docs.Getdbt.Com/Docs/Get-Started/Getting-Started-Dbt-Core

QUESTION

What is the naming convention used by dbt to generate the schema name for a model that belongs to a custom schema?

- **A: target_schema_name_custom_schema_name**
- B: custom_schema_name_target_schema_name
- C: custom_schema_name/model_name
- D: model_name/custom_schema_name

CORRECT ANSWER: **A**

EXPLANATION

Explanation: In dbt, when you specify a custom schema for a model, dbt generates the schema name for the model by concatenating the custom schema name with the target schema name, as in "target_schema_name_custom_schema_name". For example, if the target schema is "analytics" and the custom schema is "marketing", the schema name for a model named "customer_segmentation" would be "analytics_marketing_customer_segmentation".

Reference: https://docs.getdbt.com/docs/build/custom-schemas

QUESTION

A company is using dbt snapshots to track changes in a table, but they are getting errors when running queries. What could be the issue?

- A: The snapshot strategy is set to "check".
- B: The "updated_at" column is a boolean value.
- C: The "updated_at" column is not a timestamp.
- D: None of the above

CORRECT ANSWER: **B, C**

EXPLANATION

If queries are failing when using dbt snapshots, the "updated_at" column may not be a timestamp. This column is used to track changes to the data, so it must be a timestamp in order for the queries to work correctly. If it is not a timestamp, the company should change it to a timestamp or choose a different strategy for tracking changes.

Reference: https://docs.getdbt.com/docs/build/snapshots

QUESTION

Which of the following is NOT part of the out-of-the-box context supported by the dbt class?

- A: dbt.ref()
- B: dbt.this()
- C: dbt.source()
- **D: dbt.is_target()**
- E: dbt.is_incremental()

CORRECT ANSWER: **D**

EXPLANATION

Explanation: The out-of-the-box context supported by the dbt class includes dbt.ref(), dbt.this(), and dbt.source(). However, dbt.is_target() is not part of the supported context.

Option A is incorrect because dbt.ref() is part of the supported context, and it is used to return DataFrames by referencing the locations of other resources in dbt.

Option B is incorrect because dbt.this() is part of the supported context, and it is used to access the database location of the current model.

Option C is incorrect because dbt.source() is part of the supported context, and it is used to return DataFrames by referencing the locations of sources in dbt.

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

QUESTION

You have recently made some changes to the columns in your seed file and are now encountering a database error when running the dbt seed command. Which of the following is the recommended way to resolve this issue?

- A: Manually drop the table from the database and recreate it with the new structure, then re-run the dbt seed command.
- B: Rerun the dbt seed command with the --truncate flag to remove the existing data and insert the updated data.
- **C: Rerun the dbt seed command with the --full-refresh flag to drop and recreate the entire table, including all dependent objects, before inserting the data.**
- D: A simple action of closing the application and then reopen it again would fix this

CORRECT ANSWER: **C**

EXPLANATION

Explanation: If you encounter a database error after changing the columns in your seed file, the recommended way to resolve this issue is to rerun the dbt seed command with the --full-refresh flag. This will drop and recreate the entire table, including all dependent objects, before inserting the data. 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.

Option A is incorrect because manually dropping and recreating the table is a time-consuming and error-prone process that is best avoided.

Option B is incorrect because the --truncate flag only removes the existing data and does not address changes to the table structure.

Option D This is not IT Crowd Mayytte, wont work!

Reference: https://docs.getdbt.com/docs/build/seeds

QUESTION

What is the purpose of the version: 2 at the start of a model or source yml file?

- A: To indicate the file type
- B: To ensure compatibility
- **C: To improve flexibility and adaptability of the structure**
- D: To ensure the structure is more complex

CORRECT ANSWER: **C**

EXPLANATION

Explanation: The Version 2 At The Start Of A Model Or Source Yml File "Allowed The Structure To Be More Extensible." This Means That The Version 2 Structure Is More Flexible And Adaptable, Which Is Consistent With The Definition Of "Extensible." - Model And Source Yml Files Always Start With `Version: 2` Because It Allowed The Structure To Be More Extensible.

Reference: Https://Docs.Getdbt.Com/Docs/Get-Started/Getting-Started-Dbt-Core

QUESTION

You are a data analytics engineer working on a dbt project. You have a model that involves complex transformations and is built on top of other views. Which materialization should you use for this model?

- A: Incremental
- **B: Table**
- C: View
- D: Seed

CORRECT ANSWER: **B**

EXPLANATION

Explanation: In dbt, when a model is set to use the table materialization, it is built as a table on the first run and then incrementally updated on each subsequent run. The advantage of this method is that it can be faster to query tables that involve complex transformations or are built on top of other views. Therefore, for a model that involves complex transformations and is built on top of other views, you should use the table materialization. It is recommended to only switch to another materialization if performance problems are noticed.

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

QUESTION

A data team is seeking to implement a dbt snapshot strategy for monitoring changes in their customer data. Specifically, they need a snapshot strategy that can detect changes based on

a column indicating the time when a row was last updated. What snapshot strategy would be suitable for this purpose?

- A: Check strategy
- B: Surrogate key strategy
- **C: Timestamp strategy**
- D: None of the above

CORRECT ANSWER: **C**

EXPLANATION

The `timestamp` strategy uses an `updated_at` field to determine if a row has changed. If the configured `updated_at` column for a row is more recent than the last time the snapshot ran, then dbt will invalidate the old record and record the new one. If the timestamps are unchanged, then dbt will not take any action.

Reference: https://docs.getdbt.com/docs/build/snapshots

QUESTION

When using the "--full-refresh" flag to refresh an incremental model in dbt, what additional steps are recommended?

- A: Nothing else is needed
- **B: Rebuild any downstream models by including the "+" symbol at the end of the model name when running the command.**
- C: Use the "--incremental-refresh" flag to update downstream models automatically.
- D: Manually update any downstream models after running the refresh command.

CORRECT ANSWER: **B**

EXPLANATION

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

QUESTION

What is the correct way to find the SQL that dbt ran in dbt CLI?

- A: Within the test output, click on the failed test, and then select "Details"
- B: Navigate to the target/run/ directory for all run queries
- C: Navigate to the target/compiled/ directory for all compiled queries
- D: Copy the SQL into a query editor

CORRECT ANSWER: **B, C**

EXPLANATION

To access the SQL queries that dbt ran, you can navigate to the "target/run/" directory to view all run queries, or to the "target/compiled/" directory to view all compiled queries.

Reference: https://docs.getdbt.com/docs/getting-started/dbt-core/getting-started/

QUESTION

In dbt, what is the recommended approach to enforce uniqueness and not-null constraints on columns in a model?

- A: Use DDL to enforce these constraints in the database.
- B: Use the constraint configuration block in the dbt_project.yml file to specify these constraints.
- C: Use the unique and not_null configuration blocks in the model file to specify these constraints.
- **D: Copy the SQL into a query editor**

CORRECT ANSWER: **D**

EXPLANATION

Explanation: In dbt, the recommended approach to enforce uniqueness and not-null constraints on columns in a model is to use dbt's testing functionality. You can define tests in a separate directory from your models, and use the unique and not_null test functions to check that your assertions about your model hold true.

Option A is incorrect because using DDL to enforce these constraints can limit the portability and flexibility of your dbt models.

Option B is incorrect because the constraint configuration block in the dbt_project.yml file is not used to specify column constraints.

Option C is incorrect because the unique and not_null configuration blocks are not used to specify column constraints in the model file.

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

QUESTION

Which of the following is NOT a feature of dbt?

- A: Wrapping select queries in a statement to create a new relation(DDL and DML)
- B: Configuring materializations for each model you create
- C: Handling boilerplate code to materialize queries as relations
- **D: Creating features for Machine learning models**

CORRECT ANSWER: **D**

EXPLANATION

A Dbt Feature Is Handle Boilerplate Code To Materialize Queries As Relations. Description.

For Each Model You Create, You Can Easily Configure A Materialization. A Materialization Represents A Build Strategy For Your Select Query – The Code Behind A Materialization Is Robust, Boilerplate Sql That Wraps Your Select Query In A Statement To Create A New, Or Update An Existing, Relation.

Reference: Https://Docs.Getdbt.Com/Docs/Introduction

QUESTION

Suppose you are troubleshooting a dbt model that is not performing optimally, and you need to inspect the SQL queries that dbt is running to identify the problem. Where can you locate this information?

- A: Within the run output, click on a model name and select "Details" in dbt Cloud
- B: The target/compiled/ directory for compiled select statements in dbt CLI
- C: The target/run/ directory for compiled create statements in dbt CLI
- D: The dbt/var/log/syslog which contains messages from dbt and its various packages.
- E: The logs/dbt.log file for verbose logging in dbt CLI

CORRECT ANSWER: **A, B, C, E**

EXPLANATION

Explanation: The SQL that dbt is executing can be accessed via several methods such as checking the run output by selecting the "Details" option after clicking on a model name in dbt Cloud, reviewing the compiled select statements located in the target/compiled/ directory in dbt CLI, viewing the compiled create statements in the target/run/ directory in dbt CLI, and checking the logs/dbt.log file for verbose logging in dbt CLI.

To review the SQL that dbt is running, you can refer to the following resources:

- dbt Cloud: Check the run output, click on a model name, and select "Details"
- dbt CLI:
  - target/compiled/ directory for compiled select statements
  - target/run/ directory for compiled create statements
  - logs/dbt.log file for verbose logging.

Reference: https://docs.getdbt.com/docs/getting-started/dbt-core/getting-started/

QUESTION

When using sources in dbt, which is the best practice for ensuring data integrity?

- A: Running data integrity tests against the data warehouse
- B: Running data integrity tests against individual models
- C: Ignoring data integrity tests altogether
- **D: Running data integrity tests against the sources**
- E: Performing manual checks on the data

CORRECT ANSWER: **D**

EXPLANATION

In Dbt When Using Sources It Is Best Adviced To Perform And Running Data Intergrity Tests Against The Sources Rather Than Against Individual Models

Reference: Https://Docs.Getdbt.Com/Docs/Get-Started/Getting-Started/Building-Your-First-Project/Test-And-Document-Your-Project

QUESTION

What are the three ways to design tests for dbt models?

- A: Use the out of the box tests dbt ships with
- B: Create your own custom tests
- C: Use open-sourced custom schema tests
- D: Use dbt's built-in SQL queries
- E: Use open-source SQL scripts

CORRECT ANSWER: **A, B, C**

EXPLANATION

Explanation:

There Are Three Ways To Design Tests For Our Dbt Models.

1. Use The Out Of The Box Tests Dbt Ships With
2. Create Your Own Custom Tests
3. Use Open-Sourced Custom Schema Tests Out Of The Box, Dbt Ships With The Following Tests: Unique Not_Null Accepted_Values Relationships (I.E. Referential Integrity)

Reference: Https://Docs.Getdbt.Com/Docs/Get-Started/Getting-Started-Dbt-Core

QUESTION

You are working on a project that requires you to add the source freshness config to dbt. Which two blocks of information do you need to provide in the source schema.yml file to accomplish this?

- A: target property and snapshot_method(warn_after & error_after)
- B: query_method(warn_after & error_after) and loaded_at_field
- **C: freshness property(warn_after & error_after) and loaded_at_field**
- D: target property and loaded_at_field(warn_after & error_after)

CORRECT ANSWER: **C**

EXPLANATION

Explanation: To add the source freshness config to dbt, you need to provide two blocks of information to the source schema.yml: freshness property(warn_after & error_after) and loaded_at_field. The freshness property defines the maximum age of the data you want to consider when running the snapshot. The loaded_at_field specifies the name of the column in the table that tracks the last time the table was updated.

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

QUESTION

In a dbt Python model, where is the Python code executed?

- A: On the local machine where dbt is installed.
- B: Remotely on the data platform where the model is defined.
- **C: Remotely on the data platform where the model is executed.**
- D: On a separate compute cluster.

CORRECT ANSWER: **C**

EXPLANATION

Explanation: In a dbt Python model, the Python code is executed remotely on the data platform where the model is executed. This means that none of the Python code is run by dbt locally; all of it is executed on the data platform.

Option A is incorrect because none of the Python code is run locally by dbt.

Option B is incorrect because the Python code is not necessarily executed on the same data platform where the model is defined; it is executed wherever the model is executed.

Option D is incorrect because there is no requirement that the Python code be executed on a separate compute cluster; it could be executed on the same compute resources as the data platform.

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

QUESTION

What is the purpose of the dbt.config() method in a Python model in dbt?

- A: To define the data transformations to be applied to the input data.
- **B: To set configurations within the Python model.**
- C: To define the schema of the output data.
- D: To define the data lineage of the model.

CORRECT ANSWER: **B**

EXPLANATION

Explanation: According to the given scenario, the purpose of the dbt.config() method in a Python model in dbt is to set configurations within the Python model. Therefore, option B is correct, while the other options are incorrect.

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

QUESTION

Which of the following is the recommended way to test and document seeds in dbt?

- A: By writing a separate documentation file and storing it in the same directory as the seed file.
- B: By writing test cases for the seed in the model file where the seed data is being used.
- **C: By using a schema file and nesting the configurations under a seeds key.**
- D: By writing comments in the seed file describing the data and its structure.

CORRECT ANSWER: **C**

EXPLANATION

Explanation: To test and document seeds, the recommended way in dbt is to use a schema file and nest the configurations under a seeds key just like for models.

Reference: https://docs.getdbt.com/docs/build/seeds

QUESTION

What is the equivalent macro to dbt.config() for setting configurations in .sql model files in dbt?

- A: {{ setup() }}
- B: {{ define() }}
- **C: {{ config() }}**
- D: {{ dbt.run() }}

CORRECT ANSWER: **C**

EXPLANATION

Explanation: The {{ config() }} macro is the equivalent macro to dbt.config() for setting configurations in .sql model files in dbt. Therefore, option C is correct, while the other options are incorrect.

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

QUESTION

Which of the following best describes the capabilities of DBT?

- A: DBT helps you write boilerplate DML and DDL by managing transactions, dropping tables, and managing schema changes.
- B: DBT allows you to write business logic with just a SQL select statement, or a Python DataFrame, that returns the dataset you need, and dbt takes care of materialization.
- C: DBT is a tool for managing transactions, dropping tables, and managing schema changes.
- **D: All of the above.**

CORRECT ANSWER: **D**

EXPLANATION

Dbt Helps Avoid Writing Boilerplate Dml And Ddl By Managing Transactions, Dropping Tables, And Managing Schema Changes. Write Business Logic With Just A Sql Select Statement, Or A Python Dataframe, That Returns The Dataset You Need, And Dbt Takes Care Of Materialization.

Reference: Https://Docs.Getdbt.Com/Docs/Introduction

QUESTION

As a Datawarehouse Engineer you are utilizing dbt to generate a target schema for your data, and you want to verify that the schema is created accurately without spending additional time on manual creation. What steps can you take to achieve this goal?

- A: You must create the target schema manually before running dbt.

- B: Run dbt without checking if the schema already exists, and hope that it creates the schema correctly.
- **C: Run dbt it will check if the schema already exists when it runs, and create it if it doesn't.**
- D: Create the target schema using a different tool and then use dbt to run the query.

CORRECT ANSWER: **C**

EXPLANATION

Explanation: dbt checks whether the target schema already exists during its execution and generates the schema automatically if it does not exist. Hence, option C is the correct answer.

Reference: https://docs.getdbt.com/docs/getting-started/connection-profiles

QUESTION

You are a data scientist working with dbt and want to optimize the performance of your incremental models. Where should you put the is_incremental() macro to improve the speed of the query?

- A: At the end of the SQL code in the config macros.
- B: In a CTE in the SQL code.
- **C: At the beginning of the SQL code.**

CORRECT ANSWER: **C**

EXPLANATION

The correct answer is 'C' Answer: C. At the beginning of the SQL code in the config macros

QUESTION

You are a analytics engineer working with dbt. What are the default actions taken by dbt when creating models, and what options are available for modifying these actions?

- A: By default, dbt builds models as tables, uses the schema defined in the source database, and generates a unique name for each table. These behaviors can be

changed by specifying the desired schema and table name in the model definition.

- **B: By default, dbt builds models as views, uses the schema defined in the target database, and uses the file name as the view or table name in the database. These behaviors can be changed by using configurations in the model file and dbt_project.yml file.**
- C: By default, dbt builds models as tables, uses the schema defined in the target database, and generates a unique name for each table. These behaviors can be changed by specifying the desired schema and table name in the model definition.
- D: By default, dbt builds models as views, uses the schema defined in the source database, and uses the file name as the view or table name in the database. These behaviors cannot be changed.

CORRECT ANSWER: **B**

EXPLANATION

By default, dbt builds models as views, uses the schema defined in the target database, and uses the file name as the view or table name in the database. These behaviors can be changed by using configurations in the model file and dbt_project.yml file.

QUESTION

You are working on a data team that utilizes dbt to manage data transformations. You have created a model file with a specific filename, but you want to use a different name for the model identifier on your data platform.

- A: schema
- B: source
- **C: alias**
- D: target

CORRECT ANSWER: **C**

EXPLANATION

Explanation: To accomplish this, you need to modify a configuration parameter in your dbt project. Specifically, you can use the `alias` parameter in the model definition to specify a different name for the model. The `alias` parameter is utilized to give the model an alternate name that can be used in the data platform as a model identifier. By default, the filename of the model file is used as the model identifier.

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

QUESTION

Where can you configure many dbt Python models at once in dbt?

- A: In the .py files for each Python model.
- **B: In a dedicated .yml file within the models/ directory.**
- C: In a SQL file within the models/ directory.
- D: In dbt_project.yml.

CORRECT ANSWER: **B**

EXPLANATION

Explanation: Many dbt Python models can be configured at once in dbt_project.yml. Therefore, option D is correct, while the other options are incorrect.

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

QUESTION

Which combination of dbt resource types are not supported for Python in dbt?

- A: Models and sources
- **B: Tests and macros**
- C: Models and tests
- D: Snapshots and sources

CORRECT ANSWER: **B**

EXPLANATION

Explanation: Python is not supported for non-model resource types like tests and macros in dbt. Python can only be used for writing dbt models.

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

QUESTION

You have set the incremental_strategy to "append" in the dbt_project.yml file for all models in your project, but when running the dbt command, the new data is not being appended to the target table. What could be the cause of this issue?

- **A: The individual model settings are overriding the project-level settings.**
- B: The SQL used in the model is incorrect.
- C: The target table does not exist in the database.
- D: There is an issue with the data being updated.

CORRECT ANSWER: **A**

EXPLANATION

Explanation: The `incremental_strategy` option can be defined in the `dbt_project.yml` file to set the incremental strategy for all models in a dbt project. If the new data is not being appended to the target table, the most likely reason is that the individual model settings are superseding the project-level settings. This may occur if an individual model's `incremental_strategy` is configured differently, which takes precedence over the project-level settings. To avoid this problem, it is critical to ensure that the individual model settings align with the project-level settings.

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

QUESTION

In dbt, when a model description requires more than one sentence, which of the following methods should be used to ensure that the documentation is reusable across the dbt project?

- A: Use dbt docs generate command to write the description in a separate markdown file and call the macros in the schema yaml file of the model
- B: Create a doc macros and use this to write the description in a separate markdown file and call the macros in the schema yaml file of the model
- **C: Use the doc block macros to write the description in a separate markdown file and call the macros in the schema yaml file of the model**
- D: All of the above

CORRECT ANSWER: **C**

EXPLANATION

This answer is correct.Answer: C. Use The Doc Block Macros To Write The Description In A Separate Markdown File And Call The Macros In The Schema Yaml File Of The Model.

Reference: Https://Docs.Getdbt.Com/Docs/Get-Started/Getting-Started/Building-Your-First-Project/Test-And-Document-Your-Project

QUESTION

John is currently working on a dbt project and has included pre- and post-hooks in the configuration block of his model. Additionally, he added pre-hooks to the `dbt_project.yml` *file. What will be the behavior of these defined hooks?*

- A: Hooks defined in the configuration block of model will run after hooks defined in the dbt_project.yml file||
- B: Hooks defined in the dbt_project.yml file override hooks defined in the configuration block||
- C: Hooks from dependent packages will be run before hooks in the active package.||
- D: Hooks defined in both files are executed cumulatively in the order they were defined||

CORRECT ANSWER: **A, C, D**

EXPLANATION

Explanation:

Hooks are cumulative

If you define hooks in both your dbt_project.yml and in the config block of a model, both sets of hooks will be applied to your model.

Execution ordering

If multiple instances of any hooks are defined, dbt will run each hook using the following ordering:

Hooks from dependent packages will be run before hooks in the active package.
Hooks defined within the model itself will be run after hooks defined in dbt_project.yml.
Hooks within a given context will be run in the order in which they are defined.

Reference: https://docs.getdbt.com/reference/resource-configs/pre-hook-post-hook

QUESTION

Which of the following statements about dbt is true?

- A: All objects in dbt are saved in the dbt object class.
- B: dbt does not have the ability to create tables.
- **C: All objects in dbt are created as views by default**
- D: The default materialization type in dbt cannot be changed.

CORRECT ANSWER: **C**

EXPLANATION

By Default, Everything In Dbt Gets Created As A View.

Reference: Https://Docs.Getdbt.Com/Docs/Get-Started/Getting-Started/Building-Your-First-Project/Build-Your-First-Models

QUESTION

Which of the following is true about the schema where data is stored when snapshots are run in dbt?

- A: The data is stored in a separate schema for each user to maintain separate development and production environments.
- B: The data is stored in a separate schema for each snapshot run, to ensure consistency and easier model building.
- **C: The data is stored in the same target_schema for everyone, regardless of who is running it.**
- D: The data is stored in a separate schema for each project, to avoid conflicts between different projects.

CORRECT ANSWER: **C**

EXPLANATION

Explanation: In dbt, when you run snapshots, the data is saved in the same `target_schema` for all users, regardless of who executes it. This is distinct from models, which are saved in a separate schema for each user to maintain distinct development and production environments. The reason for this discrepancy is that snapshots are intended to be executed regularly, and to ensure consistency and make model

construction easier, it is best to reference the production version of the snapshot even when constructing models.

Option A is incorrect because models are kept in a separate schema for each user, not snapshots.

Option B is incorrect because data is not saved in a separate schema for each run of a snapshot.

Option D is incorrect because data is not stored in a separate schema for each project, but rather in the same `target_schema` for all users.

Reference: https://docs.getdbt.com/docs/build/snapshots

QUESTION

When a test fails in dbt, what options are available for determining the path to the file (model) that failed when using the dbt CLI?

- A: Navigating to the test output and clicking on the failed test
- B: Copying the SQL code into a new Condition
- C: Navigating to the target/compiled/schema_tests directory for all compiled test queries
- **D: error message returned by dbt**

CORRECT ANSWER: **D**

EXPLANATION

Explanation: When a test fails in dbt, the error message displayed in the console output contains information about the path to the file (model) that failed. By examining the error message, you can identify the name of the file and the line number where the test failed. This information allows you to locate and debug the problematic code within the file.

Reference: https://docs.getdbt.com/docs/getting-started/building-your-first-project/test-and-document-your-project/

QUESTION

Mary is a data engineer working on a dbt project. She wants to create a custom solution to address a specific problem area. What is the benefit of using dbt packages to accomplish this goal?

- **A: By using packages, she can develop her solution more efficiently by leveraging existing models and macros from other projects.**
- B: Using packages would limit her creativity in developing a custom solution to the specific problem area.
- C: She can develop her solution more efficiently by leveraging existing models and macros from the same project.

CORRECT ANSWER: **A**

EXPLANATION

The correct answer is 'A' Answer: A

Explanation: In 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. By leveraging packages in your dbt projects, you can more efficiently develop analytics solutions that solve specific problem areas. This can be particularly helpful when you need to solve a problem area that has already been addressed in another project, as you can leverage existing models and macros to create your custom solution.

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

QUESTION

As a data engineer working with dbt, what is a model, and where are models defined in a dbt project?

- A: A SQL model is a file that defines a target database and schema, and contains SQL scripts that transform data. Models are defined in the project.yml configuration file.
- B: A SQL model is a subdirectory within the models directory that contains multiple .sql files. Models are defined in the .yml file that defines the target database and schema.
- **C: A SQL model is a select statement that is defined in a .sql file within the models directory. Each .sql file contains one model / select statement.**
- D: A SQL model is a Python script that extracts data from a source system and loads it into a target database. Models are defined in the .yml file that specifies the ETL process.

CORRECT ANSWER: **C**

EXPLANATION

Explanation: A SQL model is a single select statement that is defined in a .sql file within the models directory. Each .sql file contains one model / select statement. The model name is inherited from the filename. Models can be nested in subdirectories within the models directory.

Option A is incorrect because a SQL model is not a file that defines a target database and schema, but rather a select statement that transforms data within a defined context.

Option B is incorrect because a SQL model is not a subdirectory within the models directory, but rather a single .sql file that contains one model / select statement.

Option D is incorrect because a SQL model is not a Python script, and models do not define ETL processes.

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

QUESTION

What type of test does dbt out of the box test for referential integrity?

- A: Unique
- B: Not_null
- C: Accepted_values
- **D: Relationships**

CORRECT ANSWER: **D**

EXPLANATION

Explanation:

There Are Three Ways To Design Tests For Our Dbt Models.

1. Use The Out Of The Box Tests Dbt Ships With
2. Create Your Own Custom Tests
3. Use Open-Sourced Custom Schema Tests Out Of The Box, Dbt Ships With The Following Tests: Unique Not_Null Accepted_Values Relationships (I.E. Referential Integrity)

Reference: Https://Docs.Getdbt.Com/Docs/Get-Started/Getting-Started-Dbt-Core

QUESTION

When using dbt, which of the following types of tests would you expect to be able to run out-of-the-box to ensure data integrity?

- A: Data duplication
- B: Null values in specific columns
- C: Conformance to predefined values
- D: Referential integrity between tables

CORRECT ANSWER: **A, B, C, D**

EXPLANATION

Out Of The Box, Dbt Ships With 1) Unique Tests 2) Not_Null Test 3) Accepted_Values Test 4) Relationships I.E Referential Integrit.

Reference: Https://Docs.Getdbt.Com/Docs/Get-Started/Getting-Started/Building-Your-First-Project/Test-And-Document-Your-Project

QUESTION

You have just set up a dbt project and are now ready to build your models. What command should you use to begin building your models?

- A: dbt build
- B: dbt create
- **C: dbt run**
- D: dbt compile

CORRECT ANSWER: **C**

EXPLANATION

The correct answer is 'C' Answer: C. Dbt Run

Reference: Https://Docs.Getdbt.Com/Docs/Get-Started/Getting-Started-Dbt-Core

QUESTION

When should an incremental model be used in dbt?

- A: When the source data tables have a small number of rows and the transformations are computationally expensive.
- B: When the source data tables have a large number of rows and the transformations are simple.
- **C: When the source data tables have a large number of rows and the transformations are computationally expensive.**
- D: When the source data tables have a small number of rows and the transformations are simple.

CORRECT ANSWER: **C**

EXPLANATION

Explanation: An incremental model should be used in dbt when the source data tables have a large number of rows and the transformations are computationally expensive. Incremental models allow you to only process new or updated rows from the source tables, reducing the amount of data that needs to be processed and improving performance.

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

QUESTION

What is the impact of using version 2 in the dbt model and source yml files?

- A: It enables the yaml structure to be more extensible.
- B: It allows for easy implementation of new structures for these yaml files in the future.
- C: It ensures compatibility with future versions of the dbt model.
- D: It limits the use of the dbt model and source yml files to only version 2.

CORRECT ANSWER: **A, B, C, D**

EXPLANATION

Explanation: The version key in the dbt model and source YAML files is a required key that specifies the version of the YAML file. Currently, version 2 is the only supported version for these YAML files.

Using version 2 in the dbt model and source YAML files has several implications, including:

A. Enabling the YAML structure to be more extensibleB. Allowing the use of new features and functionalities introduced in dbt v0.18.0 and aboveC. Enabling the definition of more advanced data tests and the specification of custom schemas for modelsD. Allowing the definition of custom sources in a dbt project

In summary, upgrading to version 2 of the dbt model and source YAML files is necessary to take advantage of the latest features and functionalities of dbt and to allow for future extensibility of the YAML structure.

Reference: https://docs.getdbt.com/docs/getting-started/building-your-first-project/test-and-document-your-project/

QUESTION

You are a data analyst working for a social media company. The company has recently acquired a new set of data for their advertising platform. The data includes information about users, their demographics, and their interactions with advertisements. The data is stored in a SQL database and you will be using a tool called dbt to perform data modeling and testing. You have been tasked with ensuring the data from the new set meets the company's requirements for accuracy, completeness, and compliance with data privacy regulations. Choose the appropriate tests that can be performed using dbt core tests on the data from the new set, to ensure the data meets the company's requirements:

- A: Test for unique values in the user identification numbers.||
- B: Test for null values in the demographic data.||
- C: Test for accepted values in the user age data, within a specified range (e.g. 13-120)||
- D: Test for referential integrity between the user and advertisement interaction tables, ensuring that every advertisement interaction has a corresponding user.||
- E: Test that the data is compliant with data privacy regulations, such as ensuring that users' personal information is encrypted and not accessible to unauthorized parties.
- F: Test for the validity of the user location data by cross-referencing with a geolocation database
- G: Regex tests for Pattern matching.

CORRECT ANSWER: **A, B, C, D**

EXPLANATION

Explanation: Generally, we always have assumptions about our source data, and it is a best practice to test these assumptions. These tests can be performed on the source data based on SQL data modeling principles using a tool called dbt.

The following tests are available in dbt:

- Unique: This test ensures that all values in a column are unique.
- Not_Null: This test ensures that no required columns have null values.
- Accepted_Values: This test checks whether the values in a column match an expected list of values.
- Relationships: This test checks the referential integrity of foreign keys in the data.

Performing these tests in dbt will ensure that the source data meets the required standards for accuracy, completeness, and compliance with data privacy regulations.

For Options E, F, and G, it is worth noting that while each of these tests could be important, they are not part of the core tests that dbt ships with. Therefore, they may not be necessary for this specific scenario.

Reference: https://docs.getdbt.com/docs/getting-started/building-your-first-project/test-and-document-your-project/

QUESTION

You want to run the models downstream of a source table in your dbt project. Which command should you use?

- A: dbt run --select source:jaffle_shop.orders
- B: dbt run --select source:jaffle_shop
- C: dbt run --select *
- **D: dbt run --select source:jaffle_shop.orders+**

CORRECT ANSWER: **D**

EXPLANATION

Explanation: To run the models downstream of a source table in your dbt project, you should use the command 'dbt run --select source:jaffle_shop.orders+'. The plus sign (+) at the end of the selector tells dbt to run all models downstream of the specified source table. Option A is incorrect because it only runs the specified source table and not the models downstream of it. Option B is incorrect because it runs all the models downstream of all the tables in the specified source. Option C is incorrect because it runs all the models in the project, including models not downstream of the specified source table.

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

QUESTION

When migrating from a legacy system to dbt, what is the recommended practice for querying the data warehouse?

- A: Use raw database tables
- B: Use stored procedures
- **C: Use dbt sources**
- D: Use SQL views

CORRECT ANSWER: **C**

EXPLANATION

Reference: Https://Docs.Getdbt.Com/Docs/Get-Started/Learning-More/Refactoring-Legacy-Sql

QUESTION

How would you use the --select flag to execute the required set of models and their associated tests?

- A: `dbt run --select tag:customer_analytics`*
- B: `dbt run --select tag:customer_analytics --select +`*
- **C: *`dbt run --select tag:customer_analytics+`***
- D: `dbt run --select tag:customer_analytics+ && dbt test`*

CORRECT ANSWER: **C**

EXPLANATION

Correct Answer: C. **`dbt run --select tag:customer_analytics+`**

Explanation: In this scenario, you want to execute a specific set of models related to customer analytics and include their dependencies. Option C correctly uses the **`--select`** flag with the tag **`customer_analytics`** followed by a **`+`** graph operator. The **`+`** graph operator indicates that you want to include all dependencies of the models tagged with **`customer_analytics`**. This command will also execute any tests associated with the selected models, as tests can be selected "indirectly" via their parents.

QUESTION

The marketing team later requests that you exclude a specific model, ***`customer_lifetime_value`****, while still executing the other customer analytics models and their dependencies. How would you modify your dbt command to accommodate this request?*

- **A: * *`dbt run --select tag:customer_analytics+ --select -customer_lifetime_value`****
- B: *`dbt run --select tag:customer_analytics+ --exclude customer_lifetime_value`**
- C: *`dbt run --select tag:customer_analytics+ --select +customer_lifetime_value`**
- D: *`dbt run --select tag:customer_analytics+ -customer_lifetime_value`**

CORRECT ANSWER: **A**

EXPLANATION

Correct Answer: A. **`dbt run --select tag:customer_analytics+ --select -customer_lifetime_value`**

Explanation: Option A is the correct answer as it uses the **`--select`** flag with the tag **`customer_analytics`** followed by a **`+`** graph operator, which selects all models with the **`customer_analytics`** tag and their dependencies. Additionally, it uses another **`--select`** flag with a **`-`** set operator followed by the model name **`customer_lifetime_value`** to exclude that specific model from execution. This command will still execute the other customer analytics models and their dependencies, while also executing any associated tests.

QUESTION

You have been asked to run a specific set of models related to the daily financial reports, but you need to exclude models tagged as "nightly" and those with a materialized configuration

set to "table."What command would you use to run the models contained in the "daily_reports" directory, excluding those with the "nightly" tag and materialized as "table"?

- **A: * `dbt run --select path:models/daily_reports --exclude tag:nightly,config.materialized:table` ***
- B: `dbt run --select path:models/daily_reports --select -tag:nightly,config.materialized:table` *
- C: `dbt run --select path:models/daily_reports,tag:nightly,config.materialized:table` *
- D: `dbt run --select path:models/daily_reports --select -tag:nightly --select -config.materialized:table` *

CORRECT ANSWER: **A**

EXPLANATION

Correct Answer: A. **`dbt run --select path:models/daily_reports --exclude tag:nightly,config.materialized:table`**

Explanation: In this case, you want to execute models contained in the "daily_reports" directory, excluding those with the "nightly" tag and materialized as "table." Option A correctly uses the **`--select`** flag with the path **`models/daily_reports`** and the **`--exclude`** flag with the exclusion criteria, separated by commas. This command will run the desired models while excluding those with the "nightly" tag and materialized as "table."

QUESTION

What is the command used to specify a target other than the default when using DBT?

- **A: --target**
- B: prod"
- C: dev"
- D: default"
- E: default"

CORRECT ANSWER: **A**

EXPLANATION

Correct Answer: A. "--Target"

Explanation: "Users Can Also Use The "--Target" Option When Issuing A Dbt Command To Use A Target Other Than The Default."

Dbt Supports Multiple Targets Within A Profile, Which Encourages The Use Of Separate Development And Production Environments. When Developing In Dbt Locally It Is Always Good Practice To Use The Dev Target Usually Set As The Default. A Separate "Prod" Target Can Also Be Created For Production Environments. Users Can Also Use The "--Target" Option When Issuing A Dbt Command To Use A Target Other Than The Default.

Reference: Https://Docs.Getdbt.Com/Docs/Get-Started/Connection-Profiless

QUESTION

In a dbt Python model, what is the concept of "lazy evaluation"?

- A: The ability to preview data using methods like .show() or .head() in development.
- B: The ability to execute Python code remotely on a data platform.
- C: The ability to create a series of meaningful transformations using CTEs.
- **D: The output of each dataframe operation not being immediately calculated, but only computed when explicitly asked for.**
- E: The output of each dataframe operation not being immediately calculated, but only computed when explicitly asked for.

CORRECT ANSWER: **D**

EXPLANATION

Answer: D

Explanation:

In a dbt Python model, the concept of "lazy evaluation" refers to the output of each dataframe operation not being immediately calculated. Instead, the operations are only computed when you explicitly ask for the final result of the data. In development, you can preview the data using methods like .show() or .head(). When you run a Python model, the full result of the final DataFrame will be saved as a table in your data warehouse.

Option A is incorrect because previewing data in development is not the same as "lazy evaluation".

Option B is incorrect because remote execution is not necessarily related to "lazy evaluation".

Option C is incorrect because CTEs are a separate concept from "lazy evaluation".

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

QUESTION

How can you specify column types in dbt?

- A: Use the type configuration block in the dbt_project.yml file to specify the column types for each model.
- B: Use the dtype function to specify column types in the SQL select statement.
- **C: Cast the column to the correct type in your model using SQL syntax.**
- D: Use the column_type configuration block in the model file to specify the column types for each column.
- E: Use the column_type configuration block in the model file to specify the column types for each column.

CORRECT ANSWER: **C**

EXPLANATION

Answer: C

Explanation:

In dbt, you can specify column types by casting the column to the correct type in your model using SQL syntax. For example, if you want to cast a column called age to an integer type, you could write:

```
select
cast(age as integer) as age
from my_table
```

This will ensure that the age column is treated as an integer by downstream tools and models.

Option A is incorrect because the type configuration block in the dbt_project.yml file is used to specify the types of sources and targets, not the types of columns in a model.

Option B is incorrect because there is no dtype function in dbt.

Option D is incorrect because there is no column_type configuration block in dbt.

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

QUESTION

You are running a model in dbt and encounter an error in the SQL code. How will dbt handle this error and how should you debug it?

- A: dbt will stop running the model and any models downstream of this model will also be stopped. No debugging information will be provided.
- B: dbt will continue running the model and any models downstream of this model will also be ran. No debugging information will be provided.
- C: dbt will print the error message on the screen and continue running the model and any models downstream of this model.
- **D: dbt will return the error that the database or data warehouse returns and any models downstream of this model will also be skipped. Use the error message and the compiled SQL to debug any errors.**
- E: dbt will return the error that the database or data warehouse returns and any models downstream of this model will also be skipped. Use the error message and the compiled SQL to debug any errors.

CORRECT ANSWER: **D**

EXPLANATION

Answer: D

Explanation: When Dbt Encounters An Error In The Sql Code, It Will Return The Error That The Database Or Data Warehouse Returns And Any Models Downstream Of This Model Will Also Be Skipped.

The dbt docs Suggests Using The Error Message And The Compiled Sql To Debug Any Errors. Therefore, Option A Is The Correct Answer.

Reference: Https://Docs.Getdbt.Com/Docs/Get-Started/Getting-Started-Dbt-Core

QUESTION

You are a data platform engineer working with a Python model that requires incremental updates. What materialization options do you have for Python models in dbt?

- A: View and incremental
- B: Table and view
- **C: Table and incremental**
- D: Ephemeral and table
- E: Ephemeral and table

CORRECT ANSWER: **C**

EXPLANATION

Answer: C. Table and incremental

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. Therefore, for a Python model that requires incremental updates, you can use either the table or incremental materialization. The incoming data must be filtered to only include new rows. 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

Imagine you are a data engineer working for a company that deals with massive amounts of data. You have been tasked with optimizing the data transformation process to reduce the time it takes to process the data.You have been using dbt as your primary tool for data transformation, but you have noticed that the process is taking too long when working with large data volumes. You have heard about a feature in dbt called "incremental_predicates" that can help improve performance in such scenarios. How does this dbt feature does this?

- A: The "incremental_predicates" feature in dbt optimizes the incremental build process for large data volumes by using Tree based algorithms to determine changed data and update only those records, resulting in improved performance and reduced processing time for data transformations.
- **B: The "incremental_predicates" feature in dbt optimizes the incremental build process for large data volumes by using specified SQL expressions to determine changed data and update only those records, resulting in improved performance and reduced processing time for data transformations.**
- C: The "incremental_predicates" feature in dbt optimizes the incremental build process for large data volumes by using specified Jinja control structures to determine changed data and update only those records, resulting in improved performance and reduced processing time for data transformations.
- D: dbt has no incremental_predicates feature, dbt uses pre-hook and post-hooks to determine changed data and update only those records, resulting in improved performance and reduced processing time for data transformations.
- E: dbt has no incremental_predicates feature, dbt uses pre-hook and post-hooks to determine changed data and update only those records, resulting in improved performance and reduced processing time for data transformations.

CORRECT ANSWER: **B**

EXPLANATION

Answer: B)
Explanation: The "incremental_predicates" feature in dbt is used to improve performance when working with large data volumes. It allows for a list of SQL expressions to be specified that will be used to optimize the incremental build process. By specifying these expressions, dbt can better determine which data has changed and only update those records instead of reprocessing the entire dataset. This can significantly improve performance and reduce the time it takes to run data transformations on large datasets.

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

QUESTION

You have an incremental model in dbt with the default setting "on_schema_change: ignore", but when running the dbt command, the new column you added is not appearing in the target table. What could be the cause of this issue?

- A: The database does not support adding new columns in this way.
- B: The on_schema_change parameter is not set correctly in the configuration.
- **C: A full-refresh of both the incremental model and any related models has not been executed.**
- D: There is an issue with the SQL used in the model.
- E: There is an issue with the SQL used in the model.

CORRECT ANSWER: **C**

EXPLANATION

Correct answer: c) A full-refresh of both the incremental model and any related models has not been executed.

Explanation:
When using an incremental model in dbt with the "on_schema_change: ignore" setting, any changes made to the columns in the incremental model will not be reflected in the target table until a full-refresh of both the incremental model and any related models is executed. This means that if you add a column to the incremental model, it will not appear in the target table until a full-refresh is executed. Therefore, the most likely cause of this issue is that a full-refresh of both the incremental model and any related models has not been executed.

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

QUESTION

How do exposures help data consumers understand and use the outputs of a dbt project?

- **A: By providing a dedicated page in the auto-generated documentation site with context relevant to the outputs.**
- B: By creating a separate schema for each user to maintain separate development and production environments.
- C: By enabling you to test and run resources that feed into the exposure.
- D: By defining and describing the upstream use of the project.

- E: By defining and describing the upstream use of the project.

CORRECT ANSWER: **A**

EXPLANATION

Answer: A

Explanation:

Exposures in dbt help data consumers understand and use the outputs of a project by providing a dedicated page in the auto-generated documentation site with context relevant to the outputs. This allows data consumers to easily access information about the outputs and how they can be used in downstream applications, dashboards, or data science pipelines.

Option B is incorrect because exposures do not create a separate schema for each user.

Option C is incorrect because testing and running resources that feed into an exposure is only one of the benefits of using exposures, and does not directly help data consumers understand and use the outputs.

Option D is incorrect because exposures define and describe the downstream use of the project, not the upstream use.

Reference: https://docs.getdbt.com/docs/build/exposures

QUESTION

Select the best reason for using dbt sources rather than referencing raw database tables:

- **A: It allows for easy dependency tracing**
- B: It improves performance of SQL queries
- C: It allows for version control of data sources
- D: It improves data visualization capabilities
- E: It improves data visualization capabilities

CORRECT ANSWER: **A**

EXPLANATION

Answer: A

Implement Dbt Sources Rather Than Referencing Raw Database Tables:

During Legacy To Dbt Migration Once We Can Confirm That Our Sql Code Has Worked And Materialized To The Data Warehouse Or Platform, Its Time To Convert All The Dataware House Sources To Dbt Sources.

The Recommended Practice For Querying The Data Warehouse From Dbt Is To Use Dbt Sources, This Allows Us To Call The Same Table In Multiple Places Using The {{ Src('My_Source', 'My_Table') }} Rather Than My_Database.My_Schema.My_Table.

This Is A Best Practice Because;

Source Freshness Reporting: Using Sources Unlocks The Ability To Run Source Freshness Reporting To Make Sure Your Raw Data Isn'T Stale.

Easy Dependency Tracing: If You'Re Migrating Multiple Stored Procedures Into Dbt, With Sources You Can See Which Queries Depend On The Same Raw Tables.

This Allows You To Consolidate Modeling Work On Those Base Tables, Rather Than Calling Them Separately In Multiple Places.

Build The Habit Of Analytics-As-Code: Sources Are An Easy Way To Get Your Feet Wet Using Config Files To Define Aspects Of Your Transformation Pipeline. With A Few Lines Of Code In A Schema .Yml File In Your Dbt Project'S /Models Subfolder, You Can Now Version Control How Your Data Sources (Snowplow, Shopify, Etc) Map To Actual Database Tables.

Reference: Https://Docs.Getdbt.Com/Docs/Get-Started/Learning-More/Refactoring-Legacy-Sql

QUESTION

What are the limitations of the dbt.config() method in terms of the types of arguments that can be passed to it?

- A: It can accept any type of data structure or function.
- B: It can only accept strings and numbers.
- **C: It can only accept basic data types such as strings, booleans, and numbers.**
- D: It can accept any data type as long as it is defined in a YAML file.
- E: It can accept any data type as long as it is defined in a YAML file.

CORRECT ANSWER: **C**

EXPLANATION

Answer: C

Explanation: The dbt.config() method can only accept basic data types such as strings, booleans, and numbers as arguments. Therefore, option C is correct, while the other options are incorrect.

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

QUESTION

A healthcare company wants to keep a historical record of changes to their patient data, such as changes to their medical history and medication information. which feature of dbt is the best for this scenario?

- A: dbt historical cannot be used for tracking changes in patient data as it violates HIPAA regulations.
- **B: dbt snapshots can be used to keep a historical record of changes to patient data, including medical history and medication information.**
- C: dbt data lineage can only be used for tracking changes to dimension tables in data warehousing and cannot be used in healthcare settings.
- D: dbt incremental can be used for tracking changes to patient data but only if the patient has given their explicit consent.
- E: dbt incremental can be used for tracking changes to patient data but only if the patient has given their explicit consent.

CORRECT ANSWER: **B**

EXPLANATION

Answer: B.

dbt snapshots can be used to keep a historical record of changes to patient data, including medical history and medication information. By using snapshots, the healthcare company can implement type-2 Slowly Changing Dimensions (SCDs) to track changes to rows in mutable source tables over time, making it easier to track the evolution of patient data over time.

Reference: https://docs.getdbt.com/docs/build/snapshots