I'm not a robot

×Sorry to interruptCSS Error You might encounter a number of different error messages during your coding process. While sometimes annoying, these error messages are intended to help you identify and fix issues with your programming with helpful descriptions, fulfilling a crucial role in your debugging workflow. However, some error messages can be confusing due to their ambiguous text and a lack of information on the possible sources, like Snowflake’s syntax error. 😕 In this article, you'll learn what an error in Snowflake is, what happens when this error occurs in Snowflake, some possible sources of this error, and how to find and fix it. What is the syntax error? An unexpected '' syntax error in Snowflake simply means that your SQL compiler has hit an obstacle in parsing your code. Specifically, while processing said code, there was an unexpected end of file (EOF), your code deviated from the standard syntax, and the compiler posits there is something missing or incomplete. This error can be due to any single statement in your blocks of code, and your SQL compiler will give you positional information on the flawed statement if it can. When debugging this error, you'll notice incomplete parameters, functions, or loop statements, causing the termination of your process. Generally, your compiler is programmed to expect certain sequences in code, like closing brackets and statements. When certain processes are left hanging without any more code to complete them, the compiler publishes an unexpected '' syntax error. Solving the '' syntax error in Snowflake The unexpected '' syntax error can occur from most SQL commands run in Snowflake. The following tutorial goes through the process of creating a data set in Snowflake and explores examples of this error in SQL queries. Prerequisites To follow along with this article, you'll need the following: ❄ A Snowflake account. ❄ A local installation of SnowSQL, a CLI tool that allows Windows, macOS, and Linux users to connect their systems to their Snowflake instances. For this article, you'll use the gearbox_data.csv file from this GitHub repository. The data set is a sample of a Kaggle project that simulated the functioning of two mechanical teeth (healthy and broken) across different load variations. Download the data set using the following CLI command: curl --output gearbox_data.csv Connecting SnowSQL to Snowflake This process allows you to set up communication between your CLI and your Snowflake instance. You can then use SnowSQL CLI commands to interact with Snowflake and its resources. Connect to your Snowflake instance using the following command in your CLI: snowsql -a -u Then, input your Snowflake password to finish logging in to the Snowflake CLI. If you're using a new account, your Snowflake instance will be bare, with no Snowflake warehouse, database, or schema resources set up: Snowflake CLI But don’t worry, we’ll walk through creating those resources using the relevant SnowSQL commands. 🤦‍♂ Utilizing SnowSQL for Your SQL Commands After connecting SnowSQL to your Snowflake instance, you can utilize the tool to execute the necessary queries and SQL operations. Your first task is to create a database, schema, and warehouse where you can run queries for your resources. Create a database with the following SQL command: create or replace database sf_errors; This creates a database named sf_errors, automatically creates a public schema, and sets the new database as the default one for your current session. You can check the database and schema name you are using with the following SQL command: select current_database(), current_schema(); Create a warehouse, defining the resources to be used for your queries, with the following SQL command: create or replace warehouse sf_errors_wh with warehouse_size='X-SMALL' auto_suspend = 180 auto_resume = true initially_suspended=true; With the database, schema, and warehouse created, all that is needed of you is to port in data tables to utilize the resources you've established in SQL queries and commands. You'll be using the gearbox_data.csv file you downloaded earlier. To input this data set into your Snowflake account, first create an empty table with the necessary structure and data types. You can do so, using the downloaded data set's columns, with the following SQL command: create or replace table sf_table (a1 double, a2 double, a3 double, a4 double, load int, failure int); Your next step is to populate the empty table you created with the CSV file's data. You can do this from the SnowSQL CLI with the PUT command. Unexpected syntax errors and fixes Up until this point, we've been running SQL commands that have been parsed by the SQL compiler and executed once validated. This validation process is basically the compiler running through checks based on the SQL syntax, the defined variables, and the SQL keywords used. You'll now send in a command that the compiler recognizes as incomplete according to its standard syntax. The PUT command — as a particular syntax that also depends on the system environment the SnowSQL command — is running in (Linux/Windows). Generally, the PUT command requires the directory of the file you want to upload and the location in Snowflake where the data will be uploaded. Try out this command with SnowSQL: put file: /* Example PUT file: error As you can see, running this command will generate an unexpected '' syntax error, since the compiler expects all the required parameters for the PUT command (directory of the data, directory in Snowflake) immediately after it starts parsing the command. Note: You might need to use Ctrl + Enter rather than Enter alone to run this command, as by default, SnowSQL requires a closing ; before executing commands. You can fix this error by adding in the missing required parameter: put file: /* Example PUT file: Fixed query @~ in SnowSQL points to the current Snowflake directory or internal stage the user is currently working on. Think of it as a pointer to the current working directory (i.e., pwd). The unexpected '' error can occur with all commands that are missing their required syntax. For instance, the earlier create table command can generate an error if the closing bracket and semicolon are left out: create or replace table sf_table_error (a1 double, a2 double, a3 double, a4 double, load int, failure int Create table error There are a number of ways your various commands can be left insufficient and parsed as incomplete by the SQL compiler. It's always important to confirm the syntax of the commands you are using. If you encounter an unexpected '' syntax error, go through your syntax line by line and ensure that you've taken care of all the required elements in the command. All commands can be affected by CTEs, stored procedures, pipes, block scripts, and even SELECT statements. Putting your skills to the test Error messages are helpful tools for your debugging processes, as they tell you exactly what obstacles were encountered in your code and where to find them. Sometimes, they’re direct pointers to the issues that plague your system. But sometimes, they’re just confusing. The unexpected '' syntax error, just like any other error, helps you create the best-standardized code. At its base, this error is concerned with the syntax of your code and whether it's up to par with the programmed syntax encoded in the compiler. In this article, you learned more about the unexpected '' syntax error, its causes, and how to debug and fix it. Want to learn more SQL tips and tricks? Check out our free SQL workshop with Ergest Xhelbati, author of Minimum Viable SQL Patterns. Check it out! Want to start syncing your data from Snowflake to all your business tools? Book a demo with a Census product specialists to learn how we work for your specific operational analytics needs. ×Sorry to interruptCSS Error At Orchestra we’re focused on making data engineers’ lives easier by building an innovative consolidated orchestration and observability platform. The advantage of having a single control plane is that architecturally, you as a data team aren’t paying 50 different vendors for 50 different compute clusters, all of which cost time and money to maintain.For too long, having a slick UI for rapidly building pipelines, end-to-end lineage, and visibility of exactly what’s going on in your data estate have been features of a data platform reserved only for large companies with big budgets. You can get started with Orchestra for free and can see it in Action hereWant to see how Orchestra is saving Data Teams hours building and maintaining Data Products? Check us out below.IntroductionWhen working with Snowflake, data engineers and developers often encounter statement error codes. These error codes are designed to help identify issues in SQL statements, query execution, and data integration processes. In this guide, we’ll break down the most common Snowflake statement error codes, provide explanations, and offer troubleshooting tips to help you resolve them quickly and efficiently.Whether you're managing a large-scale data pipeline or executing complex SQL queries, understanding Snowflake's error codes is essential for maintaining smooth workflows.1. What Are Snowflake Statement Error Codes?Snowflake statement error codes are specific numerical values assigned to errors that occur when executing queries or commands. Each code corresponds to a particular issue, making it easier to diagnose and fix problems without extensive troubleshooting. These errors can range from syntax mistakes to data integrity issues or platform constraints.Snowflake categorizes these errors into the following types:Syntax Errors: Caused by incorrect SQL syntax.Permission Errors: Triggered when you lack the necessary access rights.Resource Errors: Related to performance and resource limitations.Data Errors: Issues with data formatting, types, or constraints.2. Common Snowflake Statement Error Codes and How to Resolve ThemHere are some of the most common error codes Snowflake users encounter and guidance on how to resolve them:Error Code 0001: SQL Compilation ErrorDescription: This error occurs when there is a problem compiling the SQL query.Common Causes:Incorrect SQL syntaxMissing or undefined tables, columns, or functionsHow to Resolve:Review the SQL query for syntax errors, such as missing commas, parentheses, or keywords.Ensure all tables, columns, and functions used in the query are defined and accessible.Use Snowflake's DESCRIBE command to verify table and column names.Error Code 2001: Access DeniedDescription: You do not have the necessary permissions to execute the query or access the data.Common Causes:Lack of sufficient role or access privilegesAttempting to perform an operation on a restricted table or schemaHow to Resolve:Check the roles and privileges associated with your user account using the SHOW GRANTS command.Request necessary permissions from your Snowflake administrator or adjust your role to ensure you have access.Error Code 3013: Out of Virtual Warehouse ResourcesDescription: This error indicates that there are not enough resources in the virtual warehouse to execute the query.Common Causes:Virtual warehouse has reached capacity.Query requires more resources than the current virtual warehouse can provide.How to Resolve:Resize the virtual warehouse to a larger size to allocate more resources.Optimize your SQL queries to minimize resource usage (e.g., by avoiding cross joins or using better indexing strategies).Use Snowflake’s QUERY_HISTORY function to track which queries consume the most resources.Error Code 4000: Data Integrity ViolationDescription: A data integrity error occurs when there is an issue with the structure or content of the data.Common Causes:Violating a primary key or foreign key constraintInserting data that does not meet column constraints (e.g., null values in non-nullable fields)How to Resolve:Check the integrity constraints of the affected tables using SHOW CREATE TABLE.Validate the data being inserted or updated to ensure it adheres to the required structure and constraints.Remove or modify problematic data and attempt the query again.Error Code 5009: Query TimeoutDescription: The query has taken too long to execute and was automatically terminated.Common Causes:Long-running queries that exceed the set query timeout limit.Poorly optimized queries, especially those involving large datasets or complex joins.How to Resolve:Optimize your query by revisiting the structure, ensuring indexes are used efficiently.Break down large queries into smaller, more manageable steps.Increase the timeout setting if the query requires more processing time.3. Best Practices for Avoiding Snowflake Statement ErrorsWhile encountering statement error codes is part of working with Snowflake, following these best practices can minimize their occurrence and make troubleshooting easier:1. Write Clear and Efficient SQLBreak complex queries into smaller, manageable subqueries.Avoid unnecessary joins, subqueries, and calculations.Ensure that SQL syntax is correct by leveraging Snowflake’s built-in SQL editors and query planners.2. Monitor and Manage Resource UsageUse Snowflake's performance monitoring tools to track resource usage and query performance.Scale your virtual warehouses based on workload needs to prevent resource exhaustion errors.Implement query governance policies to avoid long-running queries that consume excessive resources.3. Manage Permissions EffectivelyRegularly audit user permissions and roles to ensure only authorized personnel have access to specific data and commands.Use role-based access control (RBAC) to limit unnecessary permissions that may cause access-related errors.4. Validate Data Before QueriesUse constraints and validation rules to ensure data integrity before executing queries.Utilize Snowflake’s data quality features to catch issues like duplicate records or incorrect data formats.4. How Snowflake Error Logs and Documentation Can HelpSnowflake provides detailed error logs that you can access through the Snowflake web interface or programmatically. These logs give additional insights into why a query failed and provide links to Snowflake documentation to guide troubleshooting. You can also check Snowflake’s official documentation for a complete list of error codes and their resolutions.To further streamline the process of resolving errors, Snowflake’s QUERY_HISTORY function can help identify problematic queries and track their performance over time.ConclusionSnowflake statement error codes provide critical insights into what went wrong when executing SQL queries. By understanding these error codes and knowing how to troubleshoot them, you can optimize your workflows and ensure smooth data operations within your Snowflake environment.Keep this guide handy as a quick reference for resolving common Snowflake error codes and ensuring that your queries run efficiently.Find out more about OrchestraOrchestra is a low-code data orchestration and data observability platform. It’s also a build any DAG you can build in Airflow using orchestra. You can use it to solve many use-cases and solutions like swiftly building data products, preserving data quality, and even data governance. Our docs are here, but why not also check out our integrations - we manage these so you can get started with your pipelines instantly. We also have a blog, written by the Orchestra team + guest writers, and some whitepapers for more in-depth reads.How to check errors in Snowflake?To check errors in Snowflake without writing any code, you can utilize the features provided in the Snowflake web interface and the system tables that offer insights into query execution and errors. Here's how you can do it:1. Use the Snowflake Web Interface (Classic Console or Snowsight)Access the History TabLog into the Snowflake web interface.Navigate to the "History" section:In the Classic Console, click on the "History" tab located at the top.In Snowsight (the new web interface), select "Activity" and then "Query History" from the sidebar.Review Query Execution HistoryView All Executed Queries: The history section displays all queries that have been executed, along with their status (success or failure), execution time, and other details.Identify Failed Queries:Look for queries marked with a status indicating failure (e.g., a red "Error" icon or message).Failed queries are usually highlighted or have a distinct status indicator.Examine Error DetailsClick on a Failed Query: Selecting a failed query will open detailed information about that query.Read the Error Message:An error message will be displayed explaining why the query failed.Error messages in Snowflake are descriptive and often provide guidance on what went wrong.2. Understand Common Error TypesSyntax Errors: Occur when there is a mistake in the SQL query structure. The error message will point out the unexpected token or keyword.Permission Issues: Happen when the user does not have the necessary privileges to execute a query or access certain data.Resource Limits: Errors related to exceeding resource quotas, such as storage limits or query timeouts.Data Load Errors: When loading data, errors can occur due to format mismatches or data inconsistencies.3. Utilize the ACCOUNT_USAGE ViewsSnowflake provides a set of account usage views that contain historical data about queries, including errors.Accessing ACCOUNT_USAGE ViewsNavigate to the Database Tab: In the web interface, go to the "Databases" section.Select the SNOWFLAKE Database: This is a shared database provided by Snowflake.Choose the ACCOUNT_USAGE Schema: It contains various views related to account usage.Review Relevant ViewsQUERY_HISTORY View:Contains information about all queries executed in your account.Includes columns like ERROR_CODE and ERROR_MESSAGE for failed queries.COPY_HISTORY View:Provides details about data loading operations.Useful for identifying errors during data import.Analyzing the DataFilter the Views:Use the interface's filtering options to focus on queries with errors.Look for entries where the ERROR_CODE is not null.4. Set Up Alerts and NotificationsWhile this involves some configuration, you can set up alerts to notify you when errors occur.Use the Alerts FeatureAccess the Alerts Section: In the web interface, navigate to where alerts and notifications are managed.Create a New Alert:Define the conditions under which the alert should be triggered (e.g., when a query fails).Specify how you want to be notified (email, dashboard notification, etc.).5. Consult the Documentation and SupportSnowflake Documentation: Offers comprehensive guides and explanations of error messages and troubleshooting steps.Community Forums and Support:Engage with the Snowflake community to find solutions to common errors.Contact Snowflake support if you encounter persistent or unclear issues.SummaryBy utilizing the Snowflake web interface's built-in tools, you can effectively monitor and investigate errors without writing any code. Regularly reviewing the query history and understanding error messages will help you maintain smooth operations and quickly address any issues that arise.What is error code 394304 in Snowflake?Error Code 394304 in Snowflake: Numeric Value Not RecognizedDescriptionError code 394304 in Snowflake signifies that a numeric value in your data is not recognized or cannot be converted into a valid number during data loading or processing. This typically occurs when the data contains non-numeric characters in a field expected to hold numeric values.Common CausesNon-Numeric Characters in Numeric Fields:The data contains letters, symbols, or special characters in columns designated for numbers.Example: Trying to load '12a3' into an INTEGER column.Incorrect Data Formatting:Numeric fields include formatting like commas, currency symbols, or parentheses that are not automatically parsed.Example: '$1,234' or '(567)' in a numeric column.Whitespace or Empty Strings:Fields contain spaces or empty strings instead of NULL or valid numeric values.Data Type Mismatch:Attempting to insert or cast string values into numeric columns without proper conversion.How to Resolve the ErrorReview Your Data:Inspect the data source for any non-numeric values in numeric fields.Look for unexpected characters or formatting issues.Cleanse the Data:Remove or correct invalid characters or formats in the source data.Use data preprocessing tools to sanitize the data before loading.Adjust Data Loading Settings:Use Data Transformation Functions:Apply functions to strip out unwanted characters during the load process.Set the NULL_IF Parameter:Specify strings that should be treated as NULLs (e.g., NULL_IF = ('', 'N/A')).Handle Errors Appropriately:Use the ON_ERROR option to decide how to handle problematic rows (e.g., ON_ERROR = 'CONTINUE' to skip errors).Validate Data Types:Ensure the target columns in Snowflake have the correct data types matching your data.Convert or cast data types explicitly if necessary.Consult the Error Message:Read the full error message accompanying the error code for specific details about the problematic value.Preventive MeasuresData Validation Prior to Loading:Implement checks to validate data types and formats before loading into Snowflake.Consistent Data Formatting:Standardize data formats in the source system to match the expected formats in Snowflake.Use Safe Data Loading Practices:Load data in smaller batches to isolate and identify errors more effectively.Example ScenarioSuppose you're trying to load a CSV file into Snowflake, and one of the numeric columns contains the value 'N/A' or '—'. Snowflake expects a numeric value but encounters a string, resulting in error code 394304.Resolution Steps:Modify the Source Data:Replace 'N/A' or '—' with NULL or a valid numeric value in the CSV file.Use the NULL_IF Option:Set NULL_IF = ('N/A', '—') in your data loading command so that these strings are treated as NULL.SummaryError code 394304 occurs when Snowflake cannot interpret a value as a valid number during data operations. By cleansing your data, adjusting loading parameters, and ensuring data types are consistent, you can resolve this error and facilitate smooth data loading and processing in Snowflake.Additional Tips:Check for Hidden Characters:Sometimes non-visible characters like tabs or carriage returns can cause issues.Leverage Snowflake Documentation:Consult Snowflake's official documentation for detailed guidance on data loading and error handling.Reach Out for Support:If the error persists after troubleshooting, consider contacting Snowflake support for assistance. ×Sorry to interruptCSS Error You can’t perform that action at this time.