INFORMATION_SCHEMA - Table Functions (Account level)

Name	Data Retention	Notes
AUTOMATIC_CLUSTERING_HISTORY (requires ACCOUNTADMIN role/MONITOR USAGE privilege)	14 days	Used for querying the **Automatic clustering history** for given tables with a specified data range **[**i.e. **NVL(**Date_Range_Start, **Date_Range_End - Interval '12 hours')** & **NVL(**Date_Range_End, **Current_Date)**, both endpoints are included, if start & end date are not specified then **default** is **last 12 hours] .** Information returned(only for ACCOUNTADMIN & role with MONITOR USAGE privilege) includes **credits consumed**, **bytes updated**, and **rows updated** each time a table is reclustered. **Columns:** START_TIME, END_TIME, TABLE_NAME, CREDITS_USED(No of credits billed), NUM_BYTES_RECLUSTERED(No of bytes reclustered) & NUM_ROWS_RECLUSTERED.
AUTO_REFRESH_REGISTRATION_HISTORY (requires ACCOUNTADMIN role/MONITOR USAGE privilege)	14 days	Used to query the **history of data files registered in the metadata of specified objects and the credits billed for these operations**. The table function returns the billing history within a specified date range **[**i.e. **NVL(**Date_Range_Start, **Date_Range_End - Interval '10 mins')** & **NVL(**Date_Range_End, **Current_Date), the default is to show the previous 10 minutes of the billing history]** for the entire Snowflake account. **Columns:** START_TIME, END_TIME, OBJECT_NAME, OBJECT_TYPE, CREDITS_USED(No of credits billed), FILES_REGISTERED (No of files registered during START_TIME and END_TIME window).
COPY_HISTORY (requires MONITOR or USAGE privilege)	14 days	- Used to query Snowflake **data loading history** along various dimensions within the last 14 days. - Returns history for **COPY INTO <table>** statements & continuous data loading using **Snowpipe**. - Avoids the **10,000** row limitation of the **LOAD_HISTORY** view. - Dropping/Recreating table removes historical data for bulk data loads (i.e. COPY INTO <table>). - Dropping/Recreating pipe object removes historical data for Snowpipe data loads. **Columns:** FILE_NAME(Source file with relative path), STAGE_LOCATION(Stage name where source file is located), LAST_LOAD_TIME, STATUS (Loaded, Load in progress, Load failed, Partially loaded or Load skipped), ROW_COUNT (No of rows loaded from source file), ROW_PARSED(No of rows parsed, Null if STATUS is Load in progress), FILE_SIZE, FIRST_ERROR_MESSAGE, ERROR_COUNT, ERROR_LIMIT, TABLE_CATALOG_NAME, PIPE_CATALOG_NAME, TABLE_NAME, PIPE_NAME.
DATA_TRANSFER_HISTORY (requires ACCOUNTADMIN role/MONITOR USAGE privilege)	14 days	Used to query the history of data transferred from Snowflake tables into a different cloud storage provider’s network (i.e. from Snowflake on AWS, GCP or Azure into the other cloud provider’s network) and/or geographical region within a specified date range. **Columns:** START_TIME, END_TIME, SOURCE_CLOUD (Data transfer originator), SOURCE_REGION, TARGET_CLOUD, TARGET_REGION, BYTES_TRANSFERRED, TRANSFER_TYPE (i.e. Operation that cause transfer, example- Copy, External Function, Replication, External Access)
DATABASE_REFRESH_HISTORY (require ACCOUNTADMIN role)	14 days	Returns the **refresh history** for a secodary database within 14 days.
DATABASE_REPLICATION_USAGE_HISTORY (requires ACCOUNTADMIN role/MONITOR USAGE privilege)	14 days	Used to query the **replication history for a specified database within a specified date range**. The information returned by the function includes the **database name, credits consumed**, and **bytes transferred** for replication.
REPLICATION_GROUP_REFRESH_HISTORY	14 days	Returns the **replication history** for a **secondary replication or failover group** within the last 14 days in the current account. **Input Param:** Name of Secondary group enclosed in single quotes.
REPLICATION_GROUP_USAGE_HISTORY (requires ACCOUNTADMIN role/MONITOR USAGE privilege)	14 days	Returns the **replication usage history** for **secondary replication or failover groups** within the last 14 days in the current account. **Key Columns:** CREDITS_USED (credit billed for replication), BYTES_TRANSFERRED etc.
REPLICATION_GROUP_REFRESH_PROGRESS	14 days	Returns the replication or failover group refresh activity (as JSON object) for the most recent refresh if it occurred within the last 14 days in the current account only.

REPLICATION_GROUP_REFRESH_PROGRESS_BY_JOB	14 days	Returns replication or failover group refresh activity (as JSON object) within the last 14 days in the current account only.
REPLICATION_USAGE_HISTORY	14 days	This function is **deprecated**, instead use **DATABASE_REPLICATION_USAGE_HISTORY**
EXTERNAL_FUNCTIONS_HISTORY (requires ACCOUNTADMIN role/MONITOR USAGE privilege)	14 days	Retrieves history of external functions called by Snowflake for the entire Snowflake account.Return results only for activity within the last 14 days.
MATERIALIZED_VIEW_REFRESH_HISTORY (requires ACCOUNTADMIN role/MONITOR USAGE privilege)	14 days	This table function is used for querying the materialized views refresh history for a specified materialized view within a specified date range. **Note:** The history is displayed in increments of 1 hour.
DATABASE_REFRESH_PROGRESS (require ACCOUNTADMIN role)	14 days	Returns a JSON object indicating the **current refresh status** for secodary database **by NAME**
DATABASE_REFRESH_PROGRESS_BY_JOB (require ACCOUNTADMIN role)	14 days	Returns a JSON object indicating the **current refresh status** for secodary database **by JOB**
WAREHOUSE_LOAD_HISTORY	14 days	Returns warehouse activity history (defined as the "query load") for a single warehouse within a specified date range, within the last 14 days. If the range falls outside the last 15 days, an error is returned. **Note #1:** To get results from this function, one of the following roles or privileges are required: *1) The ACCOUNTADMIN role can get results from this function as it has all of the global account permissions.* *2) A role with the MONITOR USAGE global privilege on the ACCOUNT can query this function for any warehouses in the account.* *3) A role with the MONITOR privilege on the WAREHOUSE can query this function for the warehouse it has permissions on.* *4) A role with the OWNERSHIP privilege on the WAREHOUSE has all permissions on the warehouse including MONITOR.* **Note #2:** If the selected period is less than 8 hours, load is shown in **5-second intervals**; otherwise, **5-minute intervals** are used. **Note #3:** For the output columns of this function, the query load value(AVG_QUEUE_LOAD) is the ratio of the total execution time (in seconds) of all queries in a specific state in an interval by the total time (in seconds) for that interval. **For example**, if 276 seconds was the total time for 4 queries in a 5 minute (300 second) interval, then the query load value is 276 / 300 = 0.92.
NOTIFICATION_HISTORY (Role - ACCOUNTADMIN, Ownership/Usage on Integration)	14 days	Used to query the history of notifications within last 14 days sent through Snowflake. These notifications include: Notifications from tasks, Notifications from Snowpipe and Email notifications.
PIPE_USAGE_HISTORY (Role required - ACCOUNTADMIN, MONITORY USAGE)	14 days	Used to query the history of data loaded into Snowflake tables using Snowpipe within a specified date range within the last 14 days. The function returns the **history of data loaded** and **credits billed** for your entire Snowflake account. **Key Columns:** START_TIME, END_TIME, PIPE_NAME, CREDITS_USED (No of credit billed for Snowpipe data loads), BYTES_INSERTED (No of bytes loaded due time window), FILES_INSERTED
VALIDATE_PIPE_LOAD	14 days	Used to validate data files processed by Snowpipe within a specified time range of 14days. The function returns details about any errors encountered during an attempted data load into Snowflake tables. **Key Columns:** ERROR(First error in Source file), FILE, LINE(No of line in source file where error encountered), CHARACTER (Postition of character where error was encountered, BYTES_OFFSET, CATEGORY, CODE, SQL_STATE, COLUMN_NAME, ROW_NUMBER, ROW_START_TIME, REJECTED_RECORD.

SEARCH_OPTIMIZATION_HISTORY	14 days	Used for querying the search optimization service maintenance history for a specified table within a specified date range of 14 days. The information returned by the function includes the table name and credits consumed each time a search optimization maintenance operation occurred. **Key Columns:** START_TIME, END_TIME, CREDITS_USED (No of credits billed for search index maintenance during START_TIME & END_TIME window), TABLE_NAME. The **search optimization service** can significantly improve the **performance** of **selective point lookup queries** that returns only one or a small number of distinct rows and Substring and regular expression searches (e.g. LIKE, RLIKE etc.), Queries on fields in VARIANT, OBJECT and ARRAY (semi-structured) columns that use EQUALITY/IN predicates, Predicate that checks NULL value or use ARRAY_CONTAINS etc.
SERVERLESS_TASK_HISTORY (Role required - ACCOUNTADMIN & Monitor USAGE)	14 days	Used for querying the serverless task usage history within last 14 days. The information returned by the function includes the task name and credits consumed by runs of each task.
STAGE_DIRECTORY_FILE_REGISTRATION_HISTORY (Role - OWNERSHIP priv on Stage, USAGE priv on Schema)	14 days	Used to query information about the metadata history of last 14 days for a directory table, including: 1) Files added or removed automatically as part of a metadata refresh. 2) Any errors found when refreshing the metadata. **Key Columns:** JOB_CREATED_TIME, FILE_NAME, OPERATION_STATUS (i.e. REGISTERED_NEW, REGISTERED_UPDATE, REGISTER_SKIPPED, REGISTER_FAILED, UNREGISTERED or UNREGISTER_FAILED), MESSAGE(Msg accompanying operation status),FILE_SIZE(Size of file added to directory table), LAST_MODIFIED
QUERY_ACCELERATION_HISTORY	14 days	Used for querying the query acceleration service history within a specified date range for last 14 days. The information returned includes the credits used for the query acceleration service at the warehouse level for a given time frame. **Columns:** START_TIME,END_TIME,CREDITS_USED,WAREHOUSE_NAME,NUM_FILES_SCANNED, NUM_BYTES_SCANNED.
QUERY_HISTORY / QUERY_HISTORY_BY_SESSION/USER/WAREHOUSE	7 days	Used to query Snowflake query history along various dimensions (time range, session, user, warehouse etc.) within last 7 days. 1) **QUERY_HISTORY** returns queries within a specified time range. 2) **QUERY_HISTORY_BY_SESSION** returns queries within a specified session and time range. 3) **QUERY_HISTORY_BY_USER** returns queries submitted by a specified user within a specified time range. 4) **QUERY_HISTORY_BY_WAREHOUSE** returns queries executed by a specified warehouse within a specified time range. **Note :** The **ACCOUNT_USAGE.QUERY_HISTORY** view can be used to query Snowflake query history by various dimensions (time range, session, user, warehouse, etc.) within the **last 365 days (1 year)**. The "Query_History" view is available in both the ACCOUNT_USAGE and READER_ACCOUNT_USAGE schemas.
LOGIN_HISTORY / LOGIN_HISTORY_BY_USER	7 days	**LOGIN_HISTORY** returns login events within a specified time range of last 7 days. **LOGIN_HISTORY_BY_USER** returns login events of a specified user within a specified time range of last 7 days.
TASK_HISTORY (Role/Priviliege requied - ACCOUNTADMIN, Global MONITOR EXECUTION privilege, OWNERSHIP privilege)	7 days	Used to query the history of task usage for the entire Snowflake account or a specified task within a specified date range of last 7 days.
REST_EVENT_HISTORY	7 days	Returns a list of SCIM REST API requests made to Snowflake over a specified time interval of last 7 days.
EXTERNAL_TABLE_FILE_REGISTRATION_HISTORY	30 days	Used to query information about the metadata history for an external table within last 30 days. The info includes: 1) Files added or removed automatically as part of a metadata refresh. 2) Any errors found when refreshing the metadata. **Input Parameter:** TABLE_NAME, START_TIME (Default 30 days) - Optional parameter
WAREHOUSE_METERING_HISTORY (Role required - ACCOUNTADMIN, MONITOR USAGE priv)	6 months	Used in queries to return the **HOURLY credit usage for a single warehouse** (**or all the warehouses** in your **account**) within a specified date range of **last 6 months**. **Key Columns:** CREDITS_USED (No of **credits billed** in specific hour), CREDITS_USED_COMPUTE (**No of credits used** in specifc hour), CREDITS_USED_CLOUD_SERVICES (**No of credits used for Cloud services** in hour).

DATABASE_**STORAGE_USAGE_HISTORY** (Role required - ACCOUNTADMIN, MONITOR USAGE priv)	6 months	Used to query the **average daily storage usage, in bytes**, **for a single database (or all the databases** in your **account**) within a specified date range of **last 6 months**. The results include: -> **All data stored in tables and materialized views** in the database(s). -> **All historical data maintained in Fail-safe** for the database(s). **<u>Columns:</u>** USAGE_DATE, DATABASE_NAME, AVERAGE_DATABASE_BYTES (No of bytes of database storage used), AVERAGE_FAILSAFE_BYTES (No of bytes of Fail-safe storage used).
STAGE_**STORAGE_USAGE_HISTORY** (Role required - ACCOUNTADMIN, MONITOR USAGE priv)	6 months	Used to query the **average daily data storage usage, in bytes**, for **ALL** the Snowflake **STAGES** in your account within a specified date range of last 6 months. The output will include storage for: **-> Named internal stages.** **-> Default staging areas (for tables and users).** **<u>Columns:</u>** USAGE_DATE, AVERAGE_STAGE_BYTES (No of bytes of Stage storage used).
COMPLETE_TASK_GRAPHS	60 minutes	Returns the **Graph Run Details** that are executed successfully, failed, or were cancelled in the past 60 minutes for a specific root task or entire Snowflake Account. A graph is currently defined as a single scheduled task or a DAG of tasks i.e. series of tasks composed of a scheduled root task and one or more dependent tasks (i.e. tasks that have one or more defined predecessor tasks). Here, root task refers to either the single scheduled task or the root task in a DAG. **Note 1:** To retrieve the details for graph runs that are currently executing, or are next scheduled to run within the next 8 days, query the **CURRENT_TASK_GRAPHS** table function. **Note 2:** The MAXIMUM number of rows returned by the function is limited to 1000 by default. To change the number of rows returned, modify the RESULT_LIMIT argument value.
CURRENT_TASK_GRAPHS	N/A	Returns the **status of a graph run that is currently executing or scheduled for execution** within next 8 days.
EXTERNAL_TABLE_FILES	N/A	Used to query information about the **staged data files included in the metadata for a specified external table**. This function cannot retrieve metadata about staged data files until the external table is refreshed (i.e. synched) to include the data files in its metadata. **<u>Key Columns:</u>** FILE_NAME(Name of source file & relative path to the staged file), FILE_SIZE, REGISTERED_ON(Timestamp when the file metadata was added to an external table i.e. when the external table metadata was refreshed with the file details).
POLICY_REFERENCES	N/A	Returns a row for each object that has the specified policy assigned to the object or returns a row for each policy assigned to the specified object. **Example #1:** For Network Policy objects only => POLICY_REFRENCES (POLICY_NAME => '< >', POLICY_KIND = '**NETWORK POLICY**'. **Example #2:** For other policy objects => POLICY_REFRENCES (POLICY_NAME => 'TEST') **Example #3:** For all policy objects => POLICY_REFRENCES(REF_ENTITY_NAME => ' ', REF_ENTITY_DOMAIN => ' ') *REF_ENTITY_NAME* -> The name of the object, such as name of TABLE, VIEW, EXTERNAL TABLE or USERs on which the policy is set. *REF_ENTITY_DOMAIN* -> The object type on which the policy is set such as ACCOUNT, INTEGRATION, TABLE, TAG, USER & VIEW. Snowflake supports the following policies when specifying the policy name as an argument: **AUTHENTICATION_POLICY, MASKING_POLICY, NETWORK_POLICY, NETWORK POLICY, PACKAGES POLICY, PASSWORD POLICY, ROW ACCESS POLICY, SESSION POLICY**
TASK_DEPENDENTS (Role required - OWNERSHIP/MONITOR/OPERATE privilege on a task)	N/A	This table **function returns the list of child tasks for a given root task in a DAG of tasks.** **<u>Input Arguments:</u>** TASK NAME, RECURSIVE - TRUE (Default, all recursive child tasks), FALSE (only direct child tasks). **<u>Key Columns:</u>** SCHEDULE (Task Schedule. Displays NULL if no schedule is defined), PREDECESSORS (JSON array for all predecessors tasks), STATE ('Started' or 'Suspended'), DEFINITION (SQL statements executed when task runs).