# Vertica Studio
## User Guide

**Release 1.00**
**February 2017**

***Shaik Abdul Wahab***

# Preface

## Purpose

This book provides information about Vertica Studio, which is a designed to work with Vertica Database.

Vertica Studio provides an easy-to-use Excel-based graphical interface to the Vertica Database for performing a database administration tasks on the Vertica Database.

## Audience

This book is intended for use by:

- System administrators
- Database administrators
- Technical personnel using Vertica Studio

## Supported Releases

This book supports the following releases:

- Vertica Database 6.0
- Vertica Database 7.0
- Vertica Database 7.1
- Vertica Database 7.2
- Vertica Database 8.0

## Prerequisites

The following prerequisite knowledge is required for this product:

- Computer technology and terminology
- Vertica Database
- Connectivity software, such as ODBC
- Microsoft Windows operating system
- Microsoft Excel Application

# Table of Contents


**Preface** ..... **2**
Purpose ..... 2
Audience ..... 2
Supported Releases ..... 2
Prerequisites ..... 2
**CHAPTER 1** ..... **7**
**Introduction** ..... **7**
Overview ..... 7
Installation ..... 7
**CHAPTER 2** ..... **8**
**Getting Started with Vertica Studio** ..... **8**
Overview ..... 8
Login Window ..... 9
Tab Details ..... 9
Instructions ..... 10
Troubleshooting ..... 13
**CHAPTER 3** ..... **14**
**Configuring Vertica Studio** ..... **14**
Database Access Prerequisites ..... 14
Define a Data Source ..... 14
Save Password in Data Source ..... 15
**CHAPTER 4** ..... **16**
**Performing Maintenance Tasks** ..... **16**
Create/Drop Schemas ..... 16
*Procedure* ..... *16*
*Privileges* ..... *16*
*Schema Attributes and Options* ..... *16*
Create/Alter/Modify User Accounts ..... 17
*Procedure* ..... *17*
*Privileges* ..... *17*
*User Attributes and Options* ..... *17*
Create/Alter/Modify Resource Pools ..... 20
*Procedure* ..... *20*
*Privileges* ..... *21*
*Resource Pools Attributes and Options* ..... *21*
Analyze/Drop Statistics ..... 28
*Procedure* ..... *28*

*Analyze Statistics Attributes and Options* ... *28*
Refresh Projections ... 29
*Procedure* ... *29*
*Refresh Projection Attributes and Options* ... *30*
Close Session ... 30
*Procedure* ... *30*
*Privileges* ... *30*
*Close Session Attributes and Options* ... *30*
Interrupt Statement ... 31
*Procedure* ... *31*
*Privileges* ... *31*
*Notes* ... *31*
*Interrupt Statement Attributes and Options* ... *31*
Configuration Parameters ... 31
*Procedure* ... *31*
*Privileges* ... *32*
*Set & Get Configuration Parameter Attributes and Options* ... *32*
Run CLOSE_ALL_SESSIONS ... 32
*Procedure* ... *32*
*Privileges* ... *33*
Run MAKE_AHM_NOW ... 33
*Procedure* ... *33*
*Privileges* ... *33*
*Run MAKE_AHM_NOW () Attributes and Options* ... *33*
Run DO_TM_TASK ... 33
*Procedure* ... *33*
*Privileges* ... *33*
*Run DO_TM_TASK () Attributes and Options* ... *34*
Generate Epoch Report ... 35
*Procedure* ... *35*
*GET_AHM_EPOCH* ... *35*
*GET_CURRENT_EPOCH* ... *35*
*GET_CURRENT_EPOCH* ... *35*
*LGE-AHM DIFFERENCE* ... *35*
*Sample Report:* ... *35*
**CHAPTER 5** ... **36**
**Setting Access Rights** ... **36**
Create / Drop / Alter Rename Role ... 36
*Create Role* ... *36*
*Procedure* ... *36*
*Privileges* ... *36*
*Drop Role* ... *36*

*Procedure* ..... *36*
*Privileges* ..... *36*
*Alter Role Rename* ..... *37*
*Procedure* ..... *37*
*Privileges* ..... *37*
*Create/Drop/Alter Rename Role Attributes and Options* ..... *37*
Assign Role to Users / Revoke Role from Users ..... 37
*Procedure* ..... *37*
*Assign Role to Users/Revoke Role from Users - Attributes and Options* ..... *38*
Grant or Revoke Object Rights ..... 38
*Procedure* ..... *38*
*Grant/Revoke - Attributes and Options* ..... *39*
Role Access Rights ..... 39
*Procedure* ..... *39*
*Role_AccessRights - Attributes and Options* ..... *40*
**CHAPTER 6** ..... **41**
**Working with Queries** ..... **41**
Export_Objects ..... 41
*Procedure* ..... *41*
*Privileges* ..... *41*
*Export_Objects/Export_Tables Attributes and Options* ..... *41*
1to1_View_Generation ..... 42
*Procedure* ..... *42*
*1to1 View Generation Attributes and Options* ..... *42*
Export ResultSet ..... 43
*Procedure* ..... *43*
Execute Query ..... 43
*Procedure* ..... *43*
**CHAPTER 7** ..... **44**
**Performing Monitoring Tasks** ..... **44**
Long Running Queries ..... 44
*Procedure* ..... *44*
Resource Consuming Queries ..... 45
*Procedure* ..... *45*
**CHAPTER 8** ..... **46**
**BACKUP TABLES** ..... **46**
Create Backup Table – Empty Table ..... 46
*Procedure* ..... *46*
*Table Counts* ..... *46*
*Create Backup Table Attributes and Options* ..... *46*
Create Backup Table – INSERT-SELECT ..... 47
*Procedure* ..... *47*

*Table Counts* ..... *47*
*Create Backup Table Attributes and Options* ..... *47*
Backup Table – COPY_PARTITIONS_TO_TABLE ..... 48
*Procedure* ..... *48*
*Privileges* ..... *48*
*Table Restrictions* ..... *48*
*Table Counts* ..... *49*
*Create Backup Table Attributes and Options* ..... *49*
Backup Table – COPY_TABLE ..... 49
*Procedure* ..... *50*
*Privileges* ..... *50*
*Table Restrictions* ..... *50*
*Table Counts* ..... *50*
*Create Backup Table Attributes and Options* ..... *51*

# CHAPTER 1
# Introduction

The following topics provide an introduction to Vertica Studio:

- Overview
- Installation

## Overview

For an overview of the capabilities of Vertica Studio and an introduction to the graphical interface, see Chapter 2: “Getting Started with Vertica Studio.”

To get familiar with the Vertica Studio interface, see the following chapters:

| Task | Location |
|---|---|
| Set up the initial installation of Vertica Studio and Database Access Prerequisites | Chapter 3: “Configuring Vertica Studio” |
| Work with Schemas, Users, Resource Pools, Statistics, Refresh Projections, Sessions, Configuration Parameters | Chapter 4: “Performing Maintenance Tasks” |
| Grant or revoke access rights for various objects | Chapter 5: “Setting Access Rights” |
| Work with SQL queries | Chapter 6: “Working With Queries” |
| Generating Long Running and Resource Consuming Queries | Chapter 7: "Performing Monitoring Tasks |
| Creating Backup Tables | Chapter 8: "Backup Tables" |

## Installation

No Installation required for Vertica Studio. Just copy the file and save in your system.

# CHAPTER 2
# Getting Started with Vertica Studio

The following topics provide the basic information to get started using Vertica Studio:

- Overview
- Login Window
- Tab Details
- Instructions
- Troubleshooting

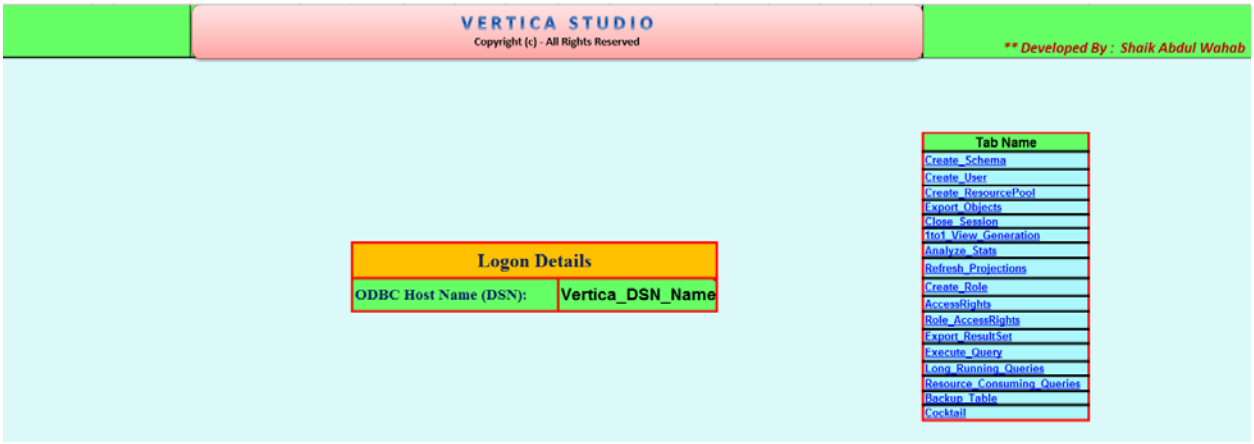
## Overview

Vertica Studio provides a comprehensive Excel-based graphical interface to the Vertica Database for performing a database administration tasks on the Vertica Database.

The following functions can be performed:

- Create and Drop Schemas
- Create, Alter and Drop Users
- Create, Alter and Drop Resource Pools
- Export Objects definition and Export Table definition
- Close Session and Interrupt Statement
- 1 to 1 View Generation
- Analyze and Drop Statistics
- Refresh Projections in foreground
- Create, Alter, Drop Roles and Assign/Revoke Role to Users
- Grant and Revoke Access Rights
- Create and Drop Read/DML/DDL Roles and assign respective Access Rights to Role
- Export Resultset
- Run an SQL query
- Generate Long Running Queries and Close the Selected Sessions
- Generate Resource Consuming Queries and Close the Selected Sessions
- Create Backup Empty and Populated tables with multiple approaches
- Set and Get the Configuration Parameter Values
- Close All Sessions
- Run MAKE_AHM_NOW
- Run START_REFRESH & CANCEL_REFRESH which runs on background.
- Run DO_TM_TASK
- Generate Epoch Report which gives AHM Epoch/Current Epoch/Last Good Epoch

# Login Window




Fill the ODBC host name (DSN Name) of the Vertica system you will be running. **You have to save the password in the ODBC DSN you use, otherwise it won't work.**

# Tab Details

| Tab Name |
|---|
| Create_Schema |
| Create_User |
| Create_ResourcePool |
| Export_Objects |
| Close_Session |
| 1to1_View_Generation |
| Analyze_Stats |
| Refresh_Projections |
| Create_Role |
| AccessRights |
| Role_AccessRights |
| Export_ResultSet |
| Execute_Query |
| Long_Running_Queries |
| Resource_Consuming_Queries |
| Backup_Table |
| Cocktail |

# Instructions

| Tab Name | Operations | Instructions |
|---|---|---|
| Logon | Logon | Fill the ODBC host name (DSN Name) of the Vertica system you will be running. **You have to save the password in the ODBC DSN you use, otherwise it won't work.** |
| Create_Schema | Create Schema | Fill all the required fields before creating Schema. |
| | Modify Schema | Fill all the required fields before modifying Schema. |
| | Drop Schema | Fill the Schema Name to drop the Schema. |
| | Clear Contents | Clear all the data which you filled in sheet. |
| Create_User | Create User | Fill all the required fields before creating User. |
| | Alter User | Fill all the required fields before modifying User. |
| | Drop User | Fill the User Name to drop the User. |
| Create_ResourcePool | Create Resource Pool | Fill all the required fields before creating Resource Pool. |
| | Alter Resource Pool | Fill all the required fields before modifying Resource Pool |
| | Drop Resource Pool | Fill the User Name to drop the Resource Pool. |
| Export_Objects | Export Objects | Fill all the required fields before running the Export Objects. |
| | Export Table | Fill all the required fields before running the Export Tables. |
| Close_Session | Close Session | Fill all the required fields before running the Close Session. |
| | Interrupt Statement | Fill all the required fields before running the Interrupt Statement. |
| 1to1_View_Generation | Generate 1:1 View | Fill all the required fields to Generate 1:1 view. If Target View Name is not provided then it will take Source Table Name as Target View Name. |
| Analyze_Stats | Analyze Stats | Fill all the required fields to Collect the Stats. |
| | Drop Stats | Fill all the required fields to drop the Stats. |

| | | |
|---|---|---|
| Refresh_Projections | Refresh Projections | Fill all the required fields to refresh the Projections. |
| Create_Role | Create Role | Fill all the required fields before creating Role. For Create Role User Name is not required |
| | Drop Role | Fill all the required fields before dropping Role. For Drop Role User Name is not required |
| | Revoke Role from Users | Fill all the required fields before Assign Role to Users. For Assign Role to Users, User Name and Role Name are required |
| | Assign Role to Users | Fill all the required fields before Revoke Role from Users. For Revoke Role from Users, Role Name and User Name are required |
| AccessRights | Grant | Fill all the required fields before Granting AccessRights to Schema/User/Role. With this we can grant the Access Right to Schema or User or Role. We can grant access right at Schema Level or Table Level. By default its Schema level. |
| | Revoke | Fill all the required fields before Revoking AccessRights to Schema/User/Role. With this we can revoke the Access Right to Schema or User or Role. We can revoke access right at Schema Level or Table Level. By default its Schema level. |
| Role_AccessRights | Create Role & Assign Respective Access Rights to Role | With this you can create Read Role, DML Role & DDL Role for the respective Schema. This will assign respective privileges to the respective roles. For Read Role assigned USAGE ON SCHEMA & SELECT ON ALL TABLES IN SCHEMA Privileges. For DML role assigned USAGE ON SCHEMA & SELECT, INSERT, UPDATE, DELETE, REFERENCES ON ALL TABLES IN SCHEMA privileges. For DDL Role assigned ALL ON SCHEMA & SELECT, INSERT, UPDATE, DELETE, REFERENCES ON ALL TABLES IN SCHEMA privileges. |
| | Drop Role | Fill all the required fields before dropping Role. It will drop all the roles which is filled in the sheet. |

| | | |
|---|---|---|
| Export_ResultSet | Export ResultSet | Export the resultset for the user given query to an excel file with a user provided report name. You will get the file path where it got exported. In this you have feasibility to limit the number of rows to export. Maximum 65534 rows you can export as this it the limit of excel sheet. |
| Execute_Query | Execute Query | With this you can execute normal queries. For Queries, Which is selecting the data, select the Query Type as "SELECT DATA". For all other queries select query type as "Other Query" |
| Long_Running_Queries | Long Running Queries | Here we can generate the Long Running Query details in the system. Long Running Query Duration Default Value is 30 Minutes. Its editable. You can define this time period. By default it will show top 10 long running queries. This field also editable. You can define the number of rows which we want to display.<br>**Close_Session:** In this sheet itself you can close the selected sessions which we want to close. |
| Resource_Consuming_Queries | Resource Consuming Queries | Here we can generate the Resource Consuming Query details in the system. By default it will show top 10 resource consuming queries. This field is editable. You can define the number of rows which we want to display.<br>**Close_Session:** In this sheet itself you can close the selected sessions which we want to close. |
| Backup_Table | Backup Table - Empty Table | Create the Empty Backup Table |
| | Backup Table - INSERT-SELECT | Create the Backup Table and Insert the data into backup table from source table. |
| | Backup Table - COPY_PARTITIONS_TO_TABLE | COPY_PARTITIONS_TO_TABLE function will create the Backup table and insert the data into the backup table. For this, Source Table needs to be Partitioned. Other wise it will fail. |

| | | |
|---|---|---|
| | Backup Table - COPY_TABLE | COPY_TABLE function will create the Backup table and insert the data into the backup table. This functionality will work only Vertica version 8.0.0 onwards. |
| Cocktail | Configuration Parameter - Set & Get the Values | We can set & get the configuration parameter values. |
| | Close All Sessions | We can close all sessions from here. |
| | MAKE_AHM_NOW | We can run MAKE_AHM_NOW here with/without parameter value "TRUE". |
| | START_REFRESH & CANCEL_REFRESH | We can run START_REFRESH from here. Start refresh will run in Background. We also run CANCEL_REFRESH from here. |
| | DO_TM_TASK | We can run DO_TM_TASK with all types of Tasks. |
| | Epoch Report | Epoch Report will generate the AHM EPOCH, CURRENT EPOCH, LAST GOOD EPOCH & LGE-AHM Difference Values from the system.. |

# Troubleshooting

For any issues with this tool Vertica Studio please contact:

**Name:** **Shaik Abdul Wahab**

**E-mail:** wahabworld@gmail.com

**Linked In:** https://in.linkedin.com/in/abdulwahabshaik

# CHAPTER 3
# Configuring Vertica Studio

Before using Vertica Studio:

1. Ensure the proper access privileges are set
2. Define the data sources to connect to
3. Save Password in DSN to connect Vertica Studio

The following topics describe these procedures in detail. If Vertica Studio is being configured, complete below steps

- Database Access Prerequisites
- Define a Data Source
- Save Password in Data Source

## Database Access Prerequisites

The following access privileges are required to use the Vertica Studio functions:

- To use the viewing functions, Select access privileges are needed for the System Tables/Views of the Vertica Database.

- To use the Copy, Drop, Create or Grant tools, the corresponding privilege is needed on the table or schema that is being modified or created.

## Define a Data Source

Define a data source for each Vertica Database prior to connecting with ODBC. Use the ODBC Data Source Administrator to create ODBC data sources and to configure the drivers.

Use the following procedure to define a data source.

**To define a data source**

1. Click the Windows Start button and type “ODBC Administrator”
   The ODBC Data Source Administrator dialog box appears and displays the User DSN tab by default.
2. Click the Drivers tab, and ensure the required ODBC driver is installed on your system.
3. Click the System DSN tab or User DSN tab, and then click the Add button.
   The Create New Data Source dialog box appears.
4. Select the Vertica ODBC driver, and then click Finish.

The ODBC Driver Setup for Vertica Database dialog box appears.

5. Enter the following fields.
   - DSN Name - Name for the data source.
     Type a unique description such as Payroll or Accounts Payable.
   - [Optional] Description - Descriptive text about this data source.
   - Server - Name or IP address of the server of your Vertica Database to connect to.
   - Port – Port number (By default port value 5433)
   - Database – Database Name of the server of your Vertica Database to connect to.
   - User Name - User name to use to log on to the Vertica Database.
   - Password - Password for the user name.
   - [Optional] Prompt for Missing Password – Select the checkbox if required
   - [Optional] Use Windows Authentication – Select the checkbox if required
6. Click OK twice.

## Save Password in Data Source

Save the User password in Password field as its mandatory for Vertica Studio.

# CHAPTER 4
# Performing Maintenance Tasks

Vertica Studio is a powerful tool for administering Vertica database. The maintenance tasks include the following:

- Create/Drop Schemas
- Create/Alter/Modify User Accounts
- Create/Alter/Modify Resource Pools
- Analyze/Drop Statistics
- Refresh Projections
- Close Session
- Interrupt Statement
- Configuration Parameters
- Run CLOSE_ALL_SESSIONS
- Run MAKE_AHM_NOW
- Run DO_TM_TASK
- Generate Epoch Report

## Create / Drop Schemas

Defines a new schema.

### Procedure

Use the following procedure to create or modify a Schema in Create_Schema tab.

**Create Schema**

1. Specify the Schema Name
2. [Optional] Specify the User Name for Authorization.
3. Click Create Schema

**Drop Schema**

1. Specify the Schema Name
2. Click Drop Schema

### Privileges

To create a schema, the user must either be a super user or have CREATE privilege for the database.

### Schema Attributes and Options

Table 1: Create Schema and Modify Schema fields Description

| Field | Description |
|---|---|
| ***Schema Name*** | Specifies the name of the schema to create. |
| **AUTHORIZATION *user-name*** | Assigns ownership of the schema to a user. If a user name is not provided, the user who creates the schema is assigned ownership. Only a Super user is allowed to create a schema that is owned by a different user. |

# Create / Alter / Modify User Accounts

Adds a name to the list of authorized database users.

## Procedure

Use the following procedure to create or modify a User in Create_User tab.

**Create User**

1. Specify the User Name
2. Define the attributes and options as indicated in Table 2.
3. Click Create User

**Alter User**

1. Specify the User Name
2. Define the attributes and options as indicated in Table 2.
3. Click Alter User

**Drop User**

1. Specify the User Name
2. Click Drop User

## Privileges

Must be a super user to create a user.

## User Attributes and Options

Table 2: Create User and Modify User fields Description

| Field | Description |
|---|---|
| **Name** | Specifies the name of the user to create; names that contain special characters must be double-quoted.<br><br>**Tip:** HP Vertica database user names are logically separate from user names of the operating system in which the server runs. If all the users of a particular server also have accounts on the server's machine, it makes sense to assign database user names that match their operating system user names. However, a server that accepts remote connections could have many database users who have no local operating system account, and in such cases there need be no connection between database user names and OS user names. |
| **ACCOUNT LOCK \| UNLOCK** | Locks or unlocks the user's access to the database. UNLOCK is the default for new users, so the keyword is optional. You'll most commonly use UNLOCK with the ALTER USER command. Specifying LOCK prevents a new user from logging in, which might be useful if you want to create an account for users who don't need access yet.<br><br>**Tip:** A super user can automate account locking by setting a maximum number of failed login attempts through the CREATE PROFILE statement. |
| **IDENTIFIED BY *'password'*** | Sets the new user's password. Supplying an empty string for password creates a user without a password, as does omitting the IDENTIFIED BY 'password' clause. If a user does not have a password, he or she will not be prompted for one when connecting.<br><br>Providing a password using the IDENTIFIED BY clause requires that the given password conform to the password complexity policy set by the user's profile. User profiles are either specified with the PROFILE parameter, or associated with a default profile if a super user omits the PROFILE parameter.<br><br>See Password Guidelines and Creating a Database Name and Password for password policies. |

| | |
|---|---|
| **PASSWORD EXPIRE** | Expires the user's password immediately. The user will be forced to change the password when he or she next logs in. The grace period setting (if any) in the user's profile is overridden.<br><br>**Note:** PASSWORD EXPIRE has no effect when using external password authentication methods such as LDAP or Kerberos. |
| **MEMORYCAP '*memory-limit*' \| NONE** | Limits the amount of memory that the user's requests can use. This value is a number representing the amount of space, followed by a unit (for example, '10G'). The unit can be one of the following:<br><br>• % percentage of total memory available to the Resource Manager. (In this case value for the size must be 0-100)<br><br>  • K— Kilobytes<br>  • M—Megabytes<br>  • G—Gigabytes<br>  • T—Terabytes<br><br>Setting this value to NONE means the user's sessions have no limits on memory use. This is the default value. |
| **PROFILE *profile* \| DEFAULT** | Assigns the user to the profile named profile. Profiles set the user's password policy. See Profiles in the Administrator's Guide for details. Using the value DEFAULT here assigns the user to the default profile. If this parameter is omitted, the user is assigned to the default profile. |
| **RESOURCE POOL *pool-name*** | Sets the name of the resource pool from which to request the user's resources. This command creates a usage grant for the user on the resource pool unless the resource pool is publicly usable. |
| **RUNTIMECAP '*time-limit*' \| NONE** | Sets the maximum amount of time any of the user's queries can execute. Time-limit is an interval, such as '1 minute' or '100 seconds' (see Interval Values for details). The maximum duration allowed is one year. Setting this value to NONE means there is no time limit on the user's queries. |

| | |
|---|---|
| | It is possible for a query to run longer than the user RUNTIMECAP if the super user sets the session RUNTIMECAP at a greater value. Non-super users can only set the session RUNTIMECAP at a value lower than their own RUNTIMECAP.<br><br>If the super user sets a session RUNTIMECAP and a RUNTIMECAP is set for the resource pool, HP Vertica uses the shorter of these two values, since the super user session value overrides the user value.<br><br>If a non-super user sets a session RUNTIMECAP and a RUNTIMECAP is set for the resource pool, HP Vertica uses the shortest of: user RUNTIMECAP, session RUNTIMECAP, and resource pool RUNTIMECAP. |
| **TEMPSPACECAP '*space-limit*' \| NONE** | Limits the amount of temporary file storage the user's requests can use. This parameter's value has the same format as the MEMORYCAP value. |
| **SEARCH_PATH *schema*[, *schema2*,...] \| DEFAULT** | Sets the user's default search path that tells HP Vertica which schemas to search for unqualified references to tables and UDFs. See Setting Search Paths in the Administrator's Guide for an explanation of the schema search path. The DEFAULT keyword sets the search path to:<br><br>"$user", public, v_catalog, v_monitor, v_internal |

# Create / Alter / Modify Resource Pools

Creates a resource pool.

## Procedure

Use the following procedure to create/alter/drop a Resource Pool in Create_Resourcepool tab.

**Create Resource Pool**

1. Specify the Resource Pool Name
2. Define the attributes and options as indicated in Table 3.
3. Click Create Resource Pool

**Alter Resource Pool**

1. Specify the Resource Pool Name
2. Define the attributes and options as indicated in Table 3.
3. Click Alter Resource Pool

**Drop Resource Pool**

1. Specify the Resource Pool Name
2. Click Drop Resource Pool.

## Privileges

Must be a super user to create a Resource Pool.

## Resource Pools Attributes and Options

Table 3: Create Resource Pool and Modify Resource Pool fields Description

| Field | Description |
|---|---|
| ***pool-name*** | Specifies the name of the resource pool to create. Name follows the conventions described in Identifiers. |
| **MEMORYSIZE '*sizeUnits*'** | Amount of memory allocated to the resource pool. Memory allocated to this pool is per node and not across the whole cluster.<br>**Default:** 0% (No memory allocation; must borrow from the GENERAL pool.<br>**Valid values:**<br>• % percentage of total memory available to the Resource Manager. (In this case, size must be 0-100.).<br>  • K—Kilobytes<br>  • M—Megabytes<br>  • G—Gigabytes<br>  • T—Terabytes<br>See also MAXMEMORYSIZE parameter. |
| **MAXMEMORYSIZE'*sizeUnits*' \| NONE** | Maximum size the resource pool could grow by borrowing memory from the GENERAL pool. See Built-In Pools for a discussion on how resource pools interact with the GENERAL pool. |

| | |
|---|---|
| | **Default:** Unlimited; pool can borrow as much memory from the GENERAL pool as is available.<br>**Valid values:**<br>• % percentage of total memory available to the Resource Manager.<br>  • K—Kilobytes<br>  • M—Megabytes<br>  • G—Gigabytes<br>  • T—Terabytes<br>The MAXMEMORYSIZE parameter refers to the maximum memory borrowed by this pool per node and not across the whole cluster.<br>If MAXMEMORYSIZENONE is specified, there is no upper limit. |
| **EXECUTIONPARALLELISM** | Limits the number of threads used to process any single query issued in this resource pool.<br><br>When set to default (AUTO), HP Vertica sets this value based on the number of cores, available memory, and amount of data in the system. Unless memory is limited, or the amount of data is very small, HP Vertica sets this value to the number of cores on the node.<br><br>Reducing this value increases the throughput of short queries issued in the pool, especially if the queries are executed concurrently.<br><br>If you choose to set this parameter manually, set it to a value between 1 and the number of cores. |
| **PRIORITY** | Integer that represents priority of queries in this pool, when they compete for resources in the GENERAL pool. Higher numbers denote higher priority.<br><br>**Default:** 0 |

| | |
|---|---|
| | • **Valid values:** Administrator-created resource pools: -100 to 100, and HOLD<br>• SYSQUERY, RECOVERY, and TM built-in pools: -110 to 110<br>When set to HOLD, pool priority is -999 and query remains queued until QUEUETIMEOUT is reached. |
| **RUNTIMEPRIORITY** | Determines the amount of run-time resources (CPU, I/O bandwidth) the Resource Manager should dedicate to queries already running in the resource pool.<br>**Default:** MEDIUM<br>**Valid values:**<br>• HIGH<br>• MEDIUM<br>• LOW<br>Queries with a HIGH run-time priority receive more CPU and I/O resources than those with a MEDIUM or LOW run-time priority.<br>**Note:** If you are upgrading from an HP Vertica version prior to 6.0, RUNTIMEPRIORITY is set to LOW. If upgrading from version 6.0 or higher, RUNTIMEPRIORITY retains its value. |
| **RUNTIMEPRIORITYTHRESHOLD** | [Default 2]<br>Specifies a time limit (in seconds) by which a query must finish before the Resource Manager assigns to it the RUNTIMEPRIORITY of the resource pool. All queries begin running at a HIGH priority. When a query's duration exceeds this threshold, it is assigned the RUNTIMEPRIORITY of the resource pool. |

| | |
|---|---|
| | **Note:** If upgrading from an HP Vertica version prior to 6.0, RUNTIMEPRIORITYTHRESHOLD gets set to 0. If upgrading from version 6.0 or higher, RUNTIMEPRIORITYTHRESHOLD retains its value. |
| **QUEUETIMEOUT** | Integer value, in seconds, indicating the maximum time the request can wait for resources to become available before rejection.<br><br>If set to NONE, the request can be queued for an unlimited amount of time. |
| **PLANNEDCONCURRENCY** | Integer that represents the preferred number of concurrently executing queries in the resource pool. When possible, query resource budgets are limited to allow this level of concurrent execution.<br><br>When set to default (AUTO), this value is calculated automatically at query runtime. HP Vertica sets this parameter to the lower of these two calculations:<br><br>• Number of logical cores<br><br>• Memory divided by 2GB<br><br>HP Vertica advises changing this value only after evaluating performance over a period of time.<br><br>**Notes:**<br><br>• When set to AUTO, HP Vertica will never set it to a value less than 4.<br><br>• This is a cluster-wide maximum and not a per-node limit.<br><br>• For clusters where the number of logical cores differs on different nodes, AUTO can apply differently on each node. Distributed queries run like the minimal effective |

| | |
|---|---|
| | planned concurrency. Single node queries run with the planned concurrency of the initiator. |
| **MAXCONCURRENCY** | Integer that represents the maximum number of concurrent execution slots available to the resource pool, across the cluster.<br><br>**Default:** unlimited<br><br>• Specifying NONE indicates no limit.<br><br>• This value is a cluster-wide maximum and not a per-node limit. |
| **RUNTIMECAP** | Sets the maximum amount of time any query on the pool has to execute.<br><br>**Default:** NONE<br><br>**Valid values:** Interval, such as 1 minute or 100 seconds, not to exceed one year. (See Interval Values for details).<br><br>• Setting this value to NONE specifies that there is no time limit on queries running on the pool.<br><br>• If the user or session also has a RUNTIMECAP, the shorter limit applies |
| **SINGLEINITIATOR** | Included only for backwards compatibility.<br><br>Default: false Do not change the value. |
| **CPUAFFINITYSET** | The set of CPUs on which queries associated with this pool are executed. For this setting, CPU numbering is defined by the number of |

| | |
|---|---|
| | CPUs in the system based on a 0 index. HP Vertica is limited to 1024 CPUs per node.<br><br>**Valid values:**<br><br>• '*cpuIndex*'—Index of a specific CPU on which to run the queries for this pool. Queries in this pool are not run on any other CPUs besides the CPU defined.<br><br>• '*cpuIndex list*'—List of CPUs on which to run the queries for this pool. CPU indexes can be comma-separated for non-continuous indexes, or use the '-' character for continuous CPU indexes. Queries in this pool are not run on any other CPUs besides the CPUs defined. The cpuIndex must be unique across all resource pools if using a CPUAFFINITYMODE of exclusive.<br><br>• '*integer percentage*'—Percentage of all available CPUs to use for this query. You must define this setting in whole percentages. HP Vertica rounds the percentages down to account for whole CPU units.<br><br>• NONE—Set no CPU affinity for this resource pool. The queries associated with this pool are executed on any CPU.<br><br>• DEFAULT—Same as NONE.<br><br>**Considerations:**<br><br>• Can only be used with user-defined resource pools.<br><br>• CPU affinity settings apply to all nodes in the cluster. CPU counts must be identical on all nodes in the cluster. |

| | |
|---|---|
| | • If you change the CPUAFFINITYSET from the default value, you must also specify the CPUAFFINITYMODE at the same time. |
| CPUAFFINITYMODE | The mode in which CPU affinity operates for this resource pool. Can be one of:<br>• SHARED—Queries run in this pool can only run on CPUs defined in CPUAFFINITYSET. Other HP Vertica resource pools can run queries that utilize the same CPUs.<br>• EXCLUSIVE—The CPUs defined in CPUAFFINITYSET are exclusively assigned to this resource pool. Other HP Vertica resource pools are unable to use the CPUs. If CPUAFFINITYSET is set as a percentage, then that percentage of CPU resources available to HP Vertica is assigned solely for this resource pool.<br>• ANY—Queries can be run on any CPU. If CPUAFFINITYSET is set to a non-default value, and you set CPUAFFINITYMODE to ANY, CPUAFFINITYSET is removed (set to NONE) by HP Vertica, since the ANY mode is only valid for the NONE set.<br>• DEFAULT—Same as ANY.<br>CPU affinity settings apply to all nodes in the cluster. CPU counts must be identical on all nodes in the cluster. |
| CASCADE TO | Indicates a secondary resource pool. Queries that exceed the RUNTIMECAP of their assigned resource pool will be executed on the secondary pool. |

| | |
|---|---|
| | Use CASCADE TO DEFAULT to remove any cascading functionality.<br><br>You cannot use CASCADE TO create a resource pool loop. |

# Analyze / Drop Statistics

Collects and aggregates data samples and storage information from all nodes that store projections associated with the specified table or column.

If the function returns successfully, HP Vertica writes the statistics to the catalog. The query optimizer uses this collected data to recommend the best possible plan to execute a query. Without analyzing table statistics, the query optimizer assumes uniform distribution of data values and equal storage usage for all projections.

## Procedure

Use the following procedure to Analyze/Drop Stats in Analyze_Stats tab.

**Analyze Statistics**

1. Specify the Schema Name
2. [Optional] Specify the Table Name & Column Name
3. [Optional] Specify the percentage of data to read from desk.
4. Click Analyze_Stats

**Drop Statistics**

1. Specify the Schema Name
2. [Optional] Specify the Table Name & Column Name
3. Click Drop Stats.

## Analyze Statistics Attributes and Options

Table 4: Analyze Statistics and Drop Statistics fields Description

| Field | Description |
|---|---|
| **Schema Name** | [Optional] Specifies the schema name.<br><br>You must be connected to the database that you specify. You cannot make changes to objects in other databases. |

| | |
|---|---|
| | You can qualify database objects as explicitly as required. For example, use a table and column (mytable.column1), a schema, table, and column (myschema.mytable.column1), and, as full qualification, a database, schema, table, and column (mydb.myschema.mytable.column1). |
| ***table*** | Specifies the name of the table and collects statistics for all projections of that table. If you are using more than one schema, specify the schema that contains the projection, as noted in the [[*db-name*.]*schema*.] entry. |
| **[*.column-name*]** | [Optional] Specifies the name of a single column, typically a predicate column. Using this option with a table specification lets you collect statistics for only that column.<br><br>If you alter a table to add or drop a column, or add a new column to a table and populate its contents with either default or other values, Hewlett-Packard recommends calling this function on the new table column to get the most current statistics. |
| ***percent*** | [Optional] Specifies what percentage of data to read from disk (not the amount of data to analyze). Specify a float from 1–100, such as 33.3. By default, the function reads 10% of the table data from disk. |

# Refresh Projections

Performs a synchronous, optionally-targeted refresh of a specified table's projections.

Information about a refresh operation—whether successful or unsuccessful—is maintained in the PROJECTION_REFRESHES system table until either the CLEAR_PROJECTION_REFRESHES() function is executed or the storage quota for the table is exceeded.
The PROJECTION_REFRESHES.IS_EXECUTING column returns a Boolean value that indicates whether the refresh is currently running (t) or occurred in the past (f).

## Procedure

Use the following procedure to Refresh Projections in Refresh_Projections tab.

**Refresh Projections**

1. Specify the Schema Name
2. Specify the Table Name
3. Click Refresh_Projections

## Refresh Projection Attributes and Options

Table 5: Refresh Projections fields Description

| Field | Description |
|---|---|
| ***Schema Name*** | [Optional] Specifies the database name and optional schema name, for example, mydb.myschema. The database name identifies objects that are not unique within the current search path. (See Setting Search Paths.) |
| ***table_name*** | Specifies the name of a specific table containing the projections to be refreshed. The REFRESH () function attempts to refresh all the tables provided as arguments in parallel. Such calls will be part of the Database Designer deployment (and deployment script).<br><br>When using more than one schema, specify the schema that contains the table, as noted above. |

# Close Session

Interrupts the specified external session, rolls back the current transaction, if any, and closes the socket.

## Procedure

Use the following procedure to close the Session in Close_Session tab.

**Refresh Projections**

1. Specify the Session_ID
2. Click Close Session

## Privileges

None; however, a non-super user can only close his or her own session.

## Close Session Attributes and Options

Table 5: Close Session fields Description

| Field | Description |
|---|---|
| ***sessionid*** | A string that specifies the session to close. This identifier is unique within the cluster at any point in time but can be reused when the session closes. |

# Interrupt Statement

Interrupts the specified statement (within an external session), rolls back the current transaction, and writes a success or failure message to the log file.

## Procedure

Use the following procedure to interrupt the Statement in Close_Session tab.

**Refresh Projections**

1. Specify the Session_ID
2. Specify the Statement_ID
3. Click Interrupt Statement

## Privileges

Must be a Super User

## Notes

- Only statements run by external sessions can be interrupted.
- Sessions can be interrupted during statement execution.
- If the *statement_id* is valid, the statement is interruptible. The command is successfully sent and returns a success message. Otherwise the system returns an error.

## Interrupt Statement Attributes and Options

Table 7: Interrupt Statement fields Description

| Field | Description |
|---|---|
| ***session_id*** | Specifies the session to interrupt. This identifier is unique within the cluster at any point in time. |
| ***statement_id*** | Specifies the statement to interrupt |

# Configuration Parameters

Use SET_CONFIG_PARAMETER to specify the value of a configuration parameter. Because HP Vertica is designed to operate with minimal configuration changes, use this capability sparingly. Carefully follow any documented guidelines for the parameter you want to configure.

## Procedure

Use the following procedure to set the Configuration Parameter value in Cocktail tab.

Set the Configuration Value functionality will work only after Vertica Version 7.2.0 onwards in this tool.

**To Set the Configuration Parameter Value**

1. Specify the Configuration Parameter Name
2. Specify the Configuration Parameter Value
3. Click Set Value

**To Get the Configuration Parameter Value**

1. Specify the Configuration Parameter Name
2. Click Get Value

## Privileges

You must be a super user to alter the database. HP Vertica displays an error message if you try to set or clear a parameter value at an invalid level.

## Set & Get Configuration Parameter Attributes and Options

Table 8: Get & Set and Configuration Parameter fields Description

| Field | Description |
|---|---|
| ***parameter*** | Specifies the name of the parameter value to set.<br>See Configuration Parameters in the Administrator's Guide for a list of supported parameters, their purposes, and usage examples. |
| ***value*** | Specifies the value of the supplied parameter argument. Syntax for this argument varies depending upon the parameter and its expected data type. For strings, enclose the argument in single quotes; integer arguments can be unquoted.<br><br>If value is specified as NULL, the parameter is cleared at that level. |

# Run CLOSE_ALL_SESSIONS

Closes all external sessions except the one issuing the CLOSE_ALL_SESSIONS functions.

## Procedure

Use the following procedure to close all the sessions in Cocktail tab.

**To Close All the Session**

1. Click Run CLOSE_ALL_SESSIONS()

## Privileges

None; however, a non-super user can only close his or her own session.

# Run MAKE_AHM_NOW

Sets the Ancient History Mark (AHM) to the greatest allowable value, and lets you drop any projections that existed before the issue occurred.

## Procedure

Use the following procedure to run make_ahm_now() in Cocktail tab.

**To Close All the Session**

1. Select “True” when nodes are down. If all nodes are up please do not select anything or select “False”
2. Click Run MAKE_AHM_NOW()

## Privileges

Must be a super user

## Run MAKE_AHM_NOW () Attributes and Options

Table 9: Run MAKE_AHM_NOW () fields Description

| Field | Description |
|---|---|
| ***true*** | [Optional] Allows AHM to advance when nodes are down. **Note:** If the AHM is advanced after the last good epoch of the failed nodes, those nodes must recover all data from scratch. Use with care. |

# Run DO_TM_TASK

Runs a Tuple Mover operation on one or more projections defined on the specified table.

## Procedure

Use the following procedure to run do_tm_task () in Cocktail tab.

**To Close All the Session**

1. Select the tuple mover operations from list box
2. Specify the schema_name.table_name/schema_name.projection_name
3. Click Run DO_TM_TASK()

## Privileges

- Any INSERT/UPDATE/DELETE privilege on table

- USAGE privileges on schema

## Run DO_TM_TASK () Attributes and Options

Table 10: Run DO_TM_TASK () fields Description

| Field | Description |
|---|---|
| ***task*** | Specifies one of the following tuple mover operations:<br>• 'moveout'—Moves out all projections on the specified table (if a particular projection is not specified) from WOS to ROS.<br>• 'mergeout'—Consolidates ROS containers and purges deleted records.<br>• 'analyze_row_count'—Automatically collects the number of rows in a projection at a time interval specified by the AnalyzeRowCountInterval configuration parameter. DO_TM_TASK aggregates row counts calculated during loads.<br>**Note:** You cannot analyze the row count of external tables with this option. You must use ANALYZE_EXTERNAL_ROW_COUNT. |
| ***[[db-name.]schema.]*** | [Optional] Specifies the schema name. (See Setting Search Paths.)<br>You must be connected to the database that you specify. You cannot make changes to objects in other databases.<br>You can qualify database objects as explicitly as required. For example, use a table and column (mytable.column1), a schema, table, and column (myschema.mytable.column1), and, as full qualification, a database, schema, table, and column (mydb.myschema.mytable.column1). |
| ***table*** | Runs a tuple mover operation for all projections within the specified table. When using more than one schema, specify the schema that contains the table with the projections you want to affect, as noted above. |
| ***projection*** | If *projection* is not passed as an argument, all projections in the system are used. If *projection* is specified, DO_TM_TASK looks for a projection of that name and, if found, uses it; if a named |

| | |
|---|---|
| | projection is not found, the function looks for a table with that name and, if found, moves out all projections on that table. |

# Generate Epoch Report

We will get the following values when we run the Epoch Report

## Procedure

Use the following procedure to run the Epoch Report in Cocktail tab.

**To Run Epoch Report**

1. Click Epoch Report
2. Will get the following values.

### GET_AHM_EPOCH

Returns the number of the epoch in which the Ancient History Mark is located. Data deleted up to and including the AHM epoch can be purged from physical storage.
**Syntax:** GET_AHM_EPOCH ()

### GET_CURRENT_EPOCH

Returns the current epoch number during which data is written by the COPY, INSERT, UPDATE, and DELETE operations.
**Syntax:** GET_CURRENT_EPOCH ()

### GET_CURRENT_EPOCH

A term used in manual recovery, LGE (Last Good Epoch) refers to the most recent epoch that can be recovered.
**Syntax:** GET_LAST_GOOD_EPOCH ()

### LGE-AHM DIFFERENCE

This column will give the value of difference between LGE & AHM
(GET_LAST_GOOD_EPOCH () - GET_AHM_EPOCH ())

## Sample Report:

| EPOCH DIFFERENCE REPORT - Will give the EPOCH values in the System | | | | |
|---|---|---|---|---|
| AHM EPOCH | CURRENT EPOCH | LAST GOOD EPOCH | LGE-AHM Difference | |
| 51 | 52 | 51 | 0 | Epoch Report |

# CHAPTER 5
# Setting Access Rights

The following topics provide ways of setting access rights for the database.

- Create / Drop / Alter Rename Role
- Assign Role to Users / Revoke Role from Users
- Grant or Revoke Object Rights
- Role Access Rights

## Create / Drop / Alter Rename Role

### Create Role

Creates a new, empty role. You must then add permissions to the role using one of the GRANT statements.

### Procedure

**Create Role**

Use the following procedure to Create Role in Create_Role tab.

1. Specify the Role Name
2. Click Create Role

### Privileges

Must be a super user to create a role.

### Drop Role

Removes a role from the database. Only the database super user can drop a role.

### Procedure

**Drop Role**

Use the following procedure to Drop Role in Create_Role tab.

1. Specify the Role Name
2. Click Drop Role

### Privileges

Must be a super user to create a role.

## Alter Role Rename

Renaming an existing role.

## Procedure

**Alter Role**

Use the following procedure to Alter Role in Create_Role tab.

1. Specify the Role Name
2. Specify the New Role Name
3. Click Alter Role

## Privileges

Must be a super user to create a role.

## Create / Drop / Alter Rename Role Attributes and Options

Table 11: Create/Drop/Alter Rename Role fields Description

| Field | Description |
|---|---|
| ***Role*** | The name of the role |
| ***new_name (for Alter Rename Role)*** | The new name for the role. |

# Assign Role to Users / Revoke Role from Users

Adds a predefined role to users. Granting a role does not activate the role automatically; the user must enable it using the SET ROLE command.

## Procedure

**Assign Role to User**

Use the following procedure to assign the Role to User in Create_Role tab.

1. Specify the Role Name
2. Specify the User Name.
3. Click Assign Role to User

**Revoke Role from User**

Use the following procedure to revoke the Role from User in Create_Role tab.

1. Specify the Role Name
2. Specify the User Name.
3. Click Revoke Role from User

### Assign Role to Users/Revoke Role from Users - Attributes and Options

Table 12: Assign Role to Users/Revoke Role from Users fields Description

| Field | Description |
|---|---|
| ***Role Name*** | Specifies the Role Name which you want to assign to users |
| **User Name** | Specify the User Name |

# Grant or Revoke Object Rights

GRANT statements, described in this section, allow you to grant privileges on database objects to specific users:

Revoke statements allow you to revoke privileges on database objects for specified users:

## Procedure

**Grant**

Use the following procedure to Grant the access rights in AccessRights tab.

1. Specify the User/Role Name
2. Specify the Schema Name.
3. Specify the Table Name. For All Tables access type “ALL TABLES”
4. Specify the Access Right
5. [Optional] Specify the With Grant Option (for Grant)
6. Click Grant

**Revoke**

Use the following procedure to revoke the access rights in AccessRights tab.

1. Specify the User/Role Name
2. Specify the Schema Name.
3. Specify the Table Name. For All Tables access type “ALL TABLES”
4. Specify the Access Right
5. [Optional] Specify the Cascade Option (for Revoke)
6. Click Revoke

## Grant / Revoke - Attributes and Options

Table 13: Grant/Revoke fields Description

| Field | Description |
|---|---|
| **User Name / Role Name** | Specifies the User / Role name |
| **Schema name** | Specifies the schema name |
| **table name** | Specifies the table on which to grant the privileges. |
| **Access right** | Specify the Access Right |
| **With Grant Option (For Grant)** | This Option is for Grant Statements. This will Allows the user to grant the same privileges to other users. |
| **Cascade (For Revoke)** | Revokes the privilege from the specified user or role and then from others. Once a user or role has been granted a privilege, the user can grant that privilege to other users and roles. The CASCADE keyword first revokes the privilege from the initial user or role, and then from other grantees extended the privilege. |

# Role Access Rights

This will create Read Role, DML Role & DDL Role for the respective Schema.

This will assign respective privileges to the respective roles.

- For Read Role it will assign USAGE ON SCHEMA & SELECT ON ALL TABLES IN SCHEMA Privileges.
- For DML role assigned USAGE ON SCHEMA & SELECT, INSERT, UPDATE, DELETE, REFERENCES ON ALL TABLES IN SCHEMA privileges.
- For DDL Role assigned ALL ON SCHEMA & SELECT, INSERT, UPDATE, DELETE, REFERENCES ON ALL TABLES IN SCHEMA privileges.

## Procedure

**Create Role & Assign Respective Access Rights**

Use the following procedure to Create Role & Assign Respective Access Rights to Role in Role_AccessRights tab.

1. Specify the Schema Name
2. Specify the Read Role Name.
3. Specify the DML Role Name
4. Specify the DDL Role Name
5. Click “Create Role & Assign Respective Access Rights to Role”

Giving 3 role name are not mandatory. You can give any role which you want create and assign the respective access rights.

**Drop Role**

Use the following procedure to Drop the roles in Role_AccessRights tab.

1. Specify the Role Name (Read Role / DML Role / DDL Role)
2. Click Drop Role

## Role_AccessRights - Attributes and Options

Table 14: Role_AccessRights fields Description

| Field | Description |
|---|---|
| **Schema name** | Specifies the schema name |
| **Read Role** | Specify the Read Role name which you want give only Read permissions. |
| **DML Role** | Specify the DML Role name which you want give DML permissions. |
| **DDL Role** | Specify the DDL Role name which you want give DDL permissions. |

# CHAPTER 6
# Working with Queries

Vertica Studio provides an easy way to get the object definitions, generate 1to1 views, export results sets and run SQL queries on the database.

Read the following topics to learn more:

- Export_Objects
- 1to1_View_Generation
- Export ResultSet
- Execute Query

## Export_Objects

Use EXPORT_OBJECTS function to recreate the exported objects.

### Procedure

**Export Objects**

Use the following procedure to export the objects definition in Export_Objects tab.

1. Specify the Schema Name
2. Specify the Table/Projection/View Name.
3. Click Export Objects

**Export Tables**

Use the following procedure to export the table definition in Export_Objects tab.

1. Specify the Schema Name
2. Specify the Table Name.
3. Click Export Tables

### Privileges

None. However:

- EXPORT_OBJECTS exports only the objects visible to the user.
- Only a super user can export output to a file.

### Export_Objects/Export_Tables Attributes and Options

Table 15: Export_Objects and Export_Tables fields Description

| Field | Description |
|---|---|
| ***Schema Name*** | Specifies the name of the schema to create. |
| **Table/Projection/View Name** | Specify the Names of the Table or Projection or View |

# 1to1_View_Generation

It will generate the 1 to 1 View Definitions for the given tables

## Procedure

**1to1 View Generation**

Use the following procedure to generate the 1 to 1 View Definition in 1to1_View_Generation tab.

1. Specify the Source Schema Name
2. Specify the Source Table Name.
3. Specify the Target Schema Name
4. [Optional] Specify the Target View Name. If no name given here then then it will create the view with Source Table Name
5. Click Generate 1:1 View

## 1to1 View Generation Attributes and Options

Table 16: 1to1 View Generation fields Description

| Field | Description |
|---|---|
| ***Source Schema Name*** | Specifies the name of a Source Schema Name |
| ***Source Table Name*** | Specifies the Source Table Name |
| ***Target Schema Name*** | Specifies the name of a Target Schema Name where you want to generate 1to1 View |
| ***Target View Name*** | [Optional] Specifies the Target View Name which you want to create |

# Export ResultSet

Export the ResultSet for the given query.

## Procedure

**Export ResultSet**

Use the following procedure to export the ResultSet for the given query in Export_ResultSet tab.

1. Specify the Query
2. [Optional] Specify the Default Schema
3. [Optional] Specify the Report Name
4. [Optional] Specify the Worksheet Name
5. [Optional] Specify the Maximum Number of Answer Set Rows to Export
6. Click Export ResultSet
7. Report Path column will give you the path where the ResultSet saved after Step 6.

# Execute Query

Execute the given query and display the result set in the same sheet

## Procedure

**Execute Query**

Use the following procedure to execute the given Query in Execute_Query tab.

1. Specify the Query (in cell C2)
2. Specify the Query Type (SELECT DATA / OTHER QUERY) in Cell A2. Default Value is "SELECT DATA"
3. [Optional] Specify the Number of Rows Required (LIMIT) in Cell J2. Default value is 100 Rows.
4. Click Execute Query or Refresh. (Both the buttons will have same functionality)
5. Cell I1 will give you the Time of Execution if the query successful. If query returns error then this cell display the error message.
6. Display the result set from row 3 onwards. 3rd will display the column names and from 4th row onwards it will display the data.
7. Clear Content button will clear all the Input / Result Set cells.

# CHAPTER 7
# Performing Monitoring Tasks

Vertica Studio provides an easy way to Monitoring the Long Running queries and Resource Consuming Queries on the database. It also provide facility to close the sessions which you select.

Read the following topics to learn more:

- Long Running Queries
- Resource Consuming Queries

## Long Running Queries

This sheet will generate the Long Running Queries from the system.

## Procedure

**Generate Long Running Queries**

Use the following procedure to generate the Long Running Queries in Long_Running_Queries tab.

1. [Optional] Specify the Query Duration in Minutes (in cell C2). Default Value is 30 Minutes
2. [Optional] Specify the Number of Rows Required (LIMIT) in Cell E2. Default value is 10 Rows.
3. Click Generate Long Running Queries or Refresh. (Both the buttons will have same functionality)
4. Cell F2 will give you the Time of Execution.
5. Display the result set from row 3 onwards. 3rd will display the column names and from 4th row onwards it will display the data.
6. Clear Content button will clear all the Input / Result Set cells.

**Close Selected Long Running Sessions**

Use the following procedure to close the Long Running Queries in Long_Running_Queries tab.

1. In Column I, Check Boxes will be available for first 20 Rows. You can select the Check Box from the respective rows where the session id is available which you want to close.
2. You can select the check boxes from multiple rows. Once you select the check box, it will display TRUE value in the respective cell.
3. After selecting the Check Boxes, Click the “Close Sleeted Session” Button.
4. This will close all the selected sessions and display the status message in Column J.

5. To confirm this action, you can again Generate Long Running Queries. In the Result set, it won't show the sessions which you closed.
6. Need to be very careful when selecting the sessions to close. Please select the correct sessions.

# Resource Consuming Queries

This sheet will generate the Resource Consuming Queries from the system.

## Procedure

**Generate Resource Consuming Queries**

Use the following procedure to generate the Resource Consuming Queries in Resource_Consuming_Queries tab.

1. [Optional] Specify the Number of Rows Required (LIMIT) in Cell C2. Default value is 10 Rows.
2. Click Generate Resource Consuming Queries or Refresh. (Both the buttons will have same functionality)
3. Cell D2 will give you the Time of Execution.
4. Display the result set from row 3 onwards. 3rd will display the column names and from 4th row onwards it will display the data.
5. Clear Content button will clear all the Input / Result Set cells.

**Close Selected Resource Consuming Sessions**

Use the following procedure to close the Resource Consuming Queries in Long_Running_Queries tab.

1. In Column I, Check Boxes will be available for first 20 Rows. You can select the Check Box from the respective rows where the session id is available which you want to close.
2. You can select the check boxes from multiple rows. Once you select the check box, it will display TRUE value in the respective cell.
3. After selecting the Check Boxes, Click the "Close Sleeted Session" Button.
4. This will close all the selected sessions and display the status message in Column J.
5. To confirm this action, you can again Generate Resource Consuming Queries. In the Resultset, it won't show the sessions which you closed.
6. Need to be very careful when selecting the sessions to close. Please select the correct sessions.

# CHAPTER 8
# BACKUP TABLES

Vertica Studio provides an easy way to taking the Backups of the table with different approaches.

Read the following topics to learn more:

- Create Backup Table – Empty Table
- Create Backup Table – INSERT-SELECT
- Create Backup Table – COPY_PARTITIONS_TO_TABLE
- Create Backup Table – COPY_TABLE

## Create Backup Table – Empty Table

Create a new table based on an existing table using the CREATE TABLE statement with the LIKE *existing_table* clause, including the projections of the existing table. Creating a new table with the LIKE option replicates the table definition and any storage policy associated with the existing table. The statement does not copy any data.

### Procedure

Use the following procedure to create the Empty Table in Backup_Table tab.

**To Create Empty Backup table**

1. Specify the Source Schema Name
2. Specify the Source Table Name
3. Specify the Target Schema Name
4. Specify the Target Table Name
5. Click Backup Table – Empty Table

### Table Counts

- Source Table Count Column will give the Number of records from the Source Table
- Target Table Count Column will give the Number of records from the Target Table, in this case its zero records.

## Create Backup Table Attributes and Options

Table 17: Create Backup Table fields Description

| Field | Description |
|---|---|
| ***Source Schema Name*** | Specifies the name of a Source Schema Name |

| | |
|---|---|
| ***Source Table Name*** | Specifies the Source Table Name |
| ***Target Schema Name*** | Specifies the name of a Target Schema Name where you want to create Backup Table |
| ***Target Table Name*** | Specifies the Target Table Name which you want to create as a Backup |

# Create Backup Table – INSERT-SELECT

Create a new table based on an existing table using the CREATE TABLE statement with the LIKE *existing_table* clause, including the projections of the existing table. Creating a new table with the LIKE option replicates the table definition and any storage policy associated with the existing table.

After creating the Target table, it will insert the data into target table from Source Table using INSERT-SELECT operation.

## Procedure

Use the following procedure to create the Backup Table in Backup_Table tab.

**To Create Empty Backup table**

1. Specify the Source Schema Name
2. Specify the Source Table Name
3. Specify the Target Schema Name
4. Specify the Target Table Name
5. Click Backup Table INSERT-SELECT

## Table Counts

- Source Table Count Column will give the Number of records from the Source Table
- Target Table Count Column will give the Number of records from the Target Table, after inserting the data from Source Table.

## Create Backup Table Attributes and Options

Table 18: Create Backup Table – INSERT SELECT fields Description

| Field | Description |
|---|---|
| ***Source Schema Name*** | Specifies the name of a Source Schema Name |
| ***Source Table Name*** | Specifies the Source Table Name |
| ***Target Schema Name*** | Specifies the name of a Target Schema Name where you want to create Backup Table |
| ***Target Table Name*** | Specifies the Target Table Name which you want to create as a Backup |

# Backup Table – COPY_PARTITIONS_TO_TABLE

For this operation, Source Table must be Partitioned table, otherwise it will throw the error and will come out.

Copies partitions from one table to another. This lightweight partition copy increases performance by initially sharing the same storage between two tables. After the copy operation is complete, the tables are independent of each other. Users can perform operations on one table without impacting the other. These operations can increase the overall storage required for both tables.

## Procedure

Use the following procedure to create the Backup Table in Backup_Table tab.

**To Create Empty Backup table**

1. Specify the Source Schema Name
2. Specify the Source Table Name
3. Specify the Target Schema Name
4. Specify the Target Table Name
5. Click Backup Table COPY_PARTITIONS_TO_TABLE

## Privileges

- Ownership or USAGE privileges on the source table.
- CREATE privileges on the target table, if COPY_PARTITIONS_TO_TABLE creates it.

## Table Restrictions

The following restrictions apply to the source and target tables:

- The target table cannot be part of a pre-join.
- If the source and target partitions are in different storage tiers, Vertica returns a warning but the operation proceeds. The partitions remain in their existing storage tier.
- The following tables cannot be used as sources or targets:
  - Temporary tables
  - Virtual tables
  - System tables
  - External tables

## Table Counts

- Source Table Count Column will give the Number of records from the Source Table
- Target Table Count Column will give the Number of records from the Target Table, after inserting the data from Source Table.

## Create Backup Table Attributes and Options

Table 19: Create Backup Table - COPY_PARTITIONS_TO_TABLE fields Description

| Field | Description |
|---|---|
| ***Source Schema Name*** | Specifies the name of a Source Schema Name |
| ***Source Table Name*** | Specifies the Source Table Name |
| ***Target Schema Name*** | Specifies the name of a Target Schema Name where you want to create Backup Table |
| ***Target Table Name*** | The target table of the partitions to copy. If the table does not exist, Vertica creates a table from the source table's definition, by calling CREATE TABLE with LIKE and INCLUDING PROJECTIONS clause. The new table inherits ownership from the source table. |

# Backup Table – COPY_TABLE

Copies one table to another. This lightweight, in-memory copy increases performance by initially sharing the same storage between two tables. The copied table includes copies of any explicitly created projections from the source table. Once copied, the source and copy tables are independent of each other. Users can perform operations on one table without impacting the other. These operations can increase the overall storage required for both tables.

Creating multiple, concurrent copies of the same table may cause some of the copies to fail. To ensure success, copy tables sequentially.

## Procedure

Use the following procedure to create the Backup Table in Backup_Table tab.

**To Create Empty Backup table**

1. Specify the Source Schema Name
2. Specify the Source Table Name
3. Specify the Target Schema Name
4. Specify the Target Table Name
5. Click Backup Table COPY TABLE

## Privileges

- User must have INSERT, UPDATE, DELETE and SELECT privileges on the source table.
- User must have CREATE privileges on the target table and target table schema if COPY_TABLE is creating a new table.
- User must have INSERT privileges on the target table if COPY_TABLE is adding to an existing table.

## Table Restrictions

The following restrictions apply to the source and target tables:

- The target table cannot be part of a pre-join.
- If the source and target partitions are in different storage tiers, Vertica returns a warning but the operation proceeds. The partitions remain in their existing storage tier.
- If the source table contains a sequence, Vertica converts the sequence to an integer before copying it to the target table. If the target table contains auto increment, identity, or named sequence columns, Vertica cancels the copy and displays an error message.
- The following tables cannot be used as sources or targets:
  - Temporary tables
  - Virtual tables
  - System tables
  - External tables

## Table Counts

- Source Table Count Column will give the Number of records from the Source Table
- Target Table Count Column will give the Number of records from the Target Table, after inserting the data from Source Table.

## Limitations

- This functionality will work only from Vertica 8.0.0 onwards.

## Create Backup Table Attributes and Options

Table 20: Create Backup Table-COPY_TABLE fields Description

| Field | Description |
|---|---|
| ***Source Schema Name*** | Specifies the name of a Source Schema Name |
| ***Source Table Name*** | The source table to copy. Vertica copies all data from this table to the target table. |
| ***Target Schema Name*** | Specifies the name of a Target Schema Name where you want to create Backup Table |
| ***Target Table Name*** | The target table of the source table. If the target table already exists, Vertica appends the source to the existing table.<br><br>If the table does not exist, Vertica creates a table from the source table's definition, by calling CREATE TABLE with LIKE and INCLUDING PROJECTIONS clause. The new table inherits ownership from the source table. For details, see Replicating a Table. |