

HP Vertica Analytics Platform 6.1.x

New Features

Doc Revision 3

Copyright© 2006-2013 Hewlett-Packard

Date of Publication: Monday, October 28, 2013



Contents

Syntax Conventions	4
---------------------------	----------

New Features and Changes in HP Vertica 6.1.x	5
---	----------

New Features and Changes in HP Vertica 6.1 SP3 (6.1.3)	5
Improved Performance When a Node Fails	5
New MC System-Level Charts	5
Hadoop Connector Supports MapR	5
New Features and Changes in HP Vertica 6.1 SP2 (6.1.2)	5
Enterprise Edition and Community Edition License Changes	6
HP Vertica End User License Agreement (EULA)	6
R Language Pack Improvements	6
Management Console Enhancements in 6.1.2	7
Using MC with the Community Edition	9
Community Edition Support for an AWS Cluster	9
New and Modified System Tables in 6.1.2	9
New Features and Changes in HP Vertica 6.1 SP1 (6.1.1)	10
Improved Diagnostics Collection	10
New Configuration Parameters	10
New SEARCH_PATH User Setting	11
Changes in the HP Vertica Catalog Directory	11
Statistics Gathering When Node is Down	11
MC Database Naming Requirement Changes	12
Grant/Revoke Privileges on all Objects Within a Schema	12
Backup and Restore Changes	12
Query Run-Time Priority Changes	14
Improved Performance When Inserting Data into Segmented Projections	15
Improved Performance for COMMIT Operations	15
COPY FROM VERTICA and EXPORT TO VERTICA Improvements	15
DROP_PARTITION Updates	16
New EXPORT_STATISTICS Argument	16
PURGE_TABLE Behavior Change	16
ALTERTABLE Updates	16
New System Table Column Identifiers	17
Monitoring Real-time Data in Query Plan Paths	17
New and Modified System Tables in 6.1.1	18
New Features and Changes in HP Vertica 6.1.0	20
EXPORT_OBJECTS and EXPORT_CATALOG Changes	20
Database Designer Improvements	20
New Hardware Verification Tests	22
Management Console Enhancements in 6.1.0	22
HP Vertica 6.1.x AMI on AWS	23
Privileges Required For Database Operations	24
Authenticating Users With Ident	24
New LDAP Client Authentication Options	25
Reduced Network Latency for Locking and Commits	25
Storage Location Updates	25
Storage Location Functions	26

CREATE TABLE LIKE Statement	26
New INTERSECT and EXCEPT Clauses for Set Operations	27
SQL WITH Clause Support	28
Improved Statistics Collection.....	28
Outer Join Improvements for Large Inputs	28
Analytic Functions Using PARTITION BY Run in Parallel	28
Subquery DELETE/UPDATE Enhancements	29
CLEAR_PROFILING Function Change	29
Improved Performance for the UNION Clause	30
User Defined Function Binary Compatibility	30
ADO.NET Connection Pooling.....	30
HP Vertica Hadoop Connector Version for Hadoop 2.0/CDH4.....	30
New Hadoop Distributed File System Connector	31
New and Modified System Tables in 6.1.0	31

Deprecated Functionality	32
---------------------------------	-----------

Retired Functionality Table	33
------------------------------------	-----------

Copyright Notice	36
-------------------------	-----------

Syntax Conventions

The following are the syntax conventions used in the HP Vertica documentation.

Syntax Convention	Description
Text without brackets/braces	Indicates content you type, as shown.
< <i>Text inside angle brackets</i> >	Represents a placeholder for which you must supply a value. The variable is usually shown in italics. See <i>Placeholders</i> below.
[Text inside brackets]	Indicates optional items; for example, CREATE TABLE [<i>schema_name</i>]. <i>table_name</i> The brackets indicate that the <i>schema_name</i> is optional. Do not type the square brackets.
{ Text inside braces }	Indicates a set of options from which you choose one; for example: QUOTES { ON OFF } indicates that exactly one of ON or OFF must be provided. You do not type the braces: QUOTES ON
Backslash \	Represents a continuation character used to indicate text that is too long to fit on a single line.
Ellipses ...	Indicate a repetition of the previous parameter. For example, option[,...] means that you can enter multiple, comma-separated options. Showing ellipses in code examples might also mean that part of the text has been omitted for readability, such as in multi-row result sets.
Indentation	Is an attempt to maximize readability; SQL is a free-form language.
<i>Placeholders</i>	Represent items that must be replaced with appropriate identifiers or expressions and are usually shown in italics.
Vertical bar	Is a separator for mutually exclusive items. For example: [ASC DESC] Choose one or neither. You do not type the square brackets.

New Features and Changes in HP Vertica 6.1.x

This section describes new features and changes that have been added across the HP Vertica 6.1.x release line.

For a list of known and resolved issues, see the HP Vertica Analytics Platform Version 6.1 Release Notes, available at <http://www.vertica.com/documentation> (<http://www.vertica.com/documentation>)

New Features and Changes in HP Vertica 6.1 SP3 (6.1.3)

Read the topics in this section for information about new and changed functionality in HP Vertica 6.1 SP3 (6.1.3).

Improved Performance When a Node Fails

HP Vertica 6.1 SP3 provides performance optimization when cluster nodes fail by distributing the work of the down nodes uniformly among available nodes throughout the cluster.

New MC System-Level Charts

Management Console 6.1 SP3 introduces new system-level activity charts to help identify issues related to resources used on individual cluster nodes, such as system memory and bottlenecks. The User Sessions chart includes updates that improve the visibility of overlap and data presentation. See the following topics in the Administrator's Guide:

- Monitoring System Memory Usage
- Monitoring System Bottlenecks
- Monitoring User Sessions

Hadoop Connector Supports MapR

The Hadoop Connector now works with MapR. See the **Supported Platforms** (<http://myvertica.vertica.com>) document for specific details of which version of MapR have been tested with the HP Vertica Hadoop Connector.

New Features and Changes in HP Vertica 6.1 SP2 (6.1.2)

Read the topics in this section for information about new and changed functionality in HP Vertica 6.1 SP2 (6.1.2).

Enterprise Edition and Community Edition License Changes

HP Vertica 6.1.2 introduces a licensing change on the HP Vertica server. Beginning with this release, all HP Vertica server RPMs are installed with a Community Edition license already applied. This simplifies installation and upgrade for Community Edition users.

Enterprise Edition customers must apply their Enterprise license upon installation to use the system without Community Edition limits.

HP Vertica End User License Agreement (EULA)

The first time you log in to the Administration Tools or Management Console after you install or upgrade HP Vertica to 6.1.2, you must accept the HP Vertica End-User License Agreement (EULA) before you can proceed.

Each time you use the Administration Tools or Management Console to upgrade your license key, you must also accept the EULA.

R Language Pack Improvements

The HP Vertica 6.1.2 R Language Pack uses R 3.0. If you installed additional R libraries in your 6.1.0 or 6.1.1 R Language Pack, then see [Installing/Upgrading the R Language Pack for HP Vertica](#) for details on upgrading your manually-installed packages to work with R 3.0. The R packages included in the 6.1.2 R Language Pack have been updated to work with R 3.0.

In 6.1.0, HP Vertica added the following enhancements for R support:

- Improved installation process
- Volatility and strictness support
- Polymorphic support
- Parameter support

See the Programmer's Guide section [Developing a User Defined Function in R](#) for complete details.

Improved installation process

The R Language Pack is now installed using an RPM instead of manually building and installing sources. You must install the R Language Pack RPM on all nodes in the cluster.

Volatility and strictness

UDx's developed in R can now use the same Volatility and Strictness settings that are available in the C++ UDx API.

Polymorphic support

R UDX's now support polymorphism. Polymorphic functions in R can be designed to accept any number and type of argument that the user of the function supplies.

Parameter Support

R UDX's can now use parameters. Parameters are passed to the R function with the *USING PARAMETERS* keyword followed by a series of key-value pairs.

Management Console Enhancements in 6.1.2

In HP Vertica 6.1.2 (SP2), Management Console has been completely redesigned to improve performance and stability, reduce disk space requirements, and to provide enhanced monitoring capabilities. MC is also now available for those of you running the *HP Vertica Community Edition* (page [9](#)).

Dedicated MC Server Requirement Lifted

HP Vertica no longer requires that you allocate a dedicated server for MC operations. You can install MC on any node in the HP Vertica cluster or (as in previous MC releases) on a dedicated server outside the HP Vertica cluster.

Installing/Upgrading MC 6.1.2

If you are moving from MC 6.1.1 to MC 6.1.2, you can now install MC on an HP Vertica cluster node, which requires a fresh install. HP does not provide scripts to migrate data from your existing server to the cluster node. See *Installing and Configuring Management Console* in the *Installation Guide*.

Embedded Database Removed from MC

The embedded `VERTICA` database has been removed from MC. Instead of querying a separate, internal database, MC now directly queries Data Collector tables on the MC-monitored databases themselves. This change significantly decreases MC disk requirements and can improve system performance by improving scalability between the agents and MC under heavy loads. See *Management Console Architecture* in the *Concepts Guide*.

Changes to the MC installation path

MC 6.1.2 is now installed on `/opt/vconsole`, instead of `/opt/vertica`. Separate storage locations for MC and HP Vertica prevent directory ownership/permission issues between Linux administrative users when you install MC on an HP Vertica node.

HP Vertica License Requirements on MC

The following license changes were made in HP Vertica 6.1.2.

- License requirements after you install/upgrade HP Vertica

The first time you log in to Management Console after you install or upgrade HP Vertica, you must accept the HP Vertica End-User License Agreement (EULA) in order to proceed.

- License requirements when you install HP Vertica through the MC interface
If you use the MC Cluster Installation Wizard to create a Community Edition cluster, you can leave the HP Vertica license key blank. Omitting the license key specifies that you're installing the HP Vertica Community Edition with a 1 TB, 3-node limit. You specify a license for Enterprise Edition installations only.
- Updating your HP Vertica license through the MC interface
If you upload a new license key through the MC Settings page (for example, if you upgrade from Community Edition to Enterprise Edition), you must accept the HP Vertica EULA after you click Apply in order to proceed.

Each time you upgrade your license key through MC, you will also have to accept the EULA.

Cluster Discovery

When you install MC 6.1.2 on a cluster node, MC now automatically discovers the cluster, which appears on the Databases and Clusters page. If you install MC on a server outside the HP Vertica cluster, you must manually import the cluster. See *Managing Database Clusters on MC in the Administrator's Guide*.

MC-created Database Passwords

MC now allows you to create passwordless HP Vertica database through the MC interface. Previously, a password was required.

New MC Monitoring Charts

New and enhanced graph-based charts can help you more effectively monitor MC-managed database clusters:

- On the MC **Overview** page, the former Cluster subsection has been renamed CPU/Memory Usage with the following modifications:
 - The cluster information has been moved within the Database subsection panel
 - A new CPU/Memory Usage subsection presents graph-based data that includes overall average CPU and memory consumption, as well CPU/memory usage by-node over the last hour
- The **License** subsection has been moved from the Overview page to a new tab at the bottom of the page, which directs you to the Settings > License page.
- The **Activity** page now lets you toggle between query concurrency, internal sessions, and user sessions. Each chart also shows corresponding average resource usage (CPU/memory, networking, and disk storage) based on maximum rated bandwidth.
- The **Node Details** page now lets you view average CPU, memory, disk I/O percent usage, network consumption in kilobytes, and the percentage of disk storage the running queries have been using.

See *Monitoring HP Vertica Using MC in the Administrator's Guide*.

MC Support for an AWS Cluster

If your Amazon Web Services (AWS) cluster is running HP Vertica Release 6.1.2 or later, you can install and run <Management Console> to monitor and manage your database.

If you are creating a new HP Vertica cluster on AWS, build your new cluster using a 6.1.2 or later Amazon Machine Image (AMI).

This release of MC on AWS includes restrictions:

- You cannot create a cluster on AWS using the MC.
- You cannot import a cluster into AWS using the MC.
- You cannot monitor an AWS cluster using MC on a node that is outside of your AWS cluster. You must install MC on an instance within the AWS cluster itself.

Note: Each version of HP Vertica Management Console is compatible only with the matching version of the HP Vertica server. Version numbers must match to three digits; for example, HP Vertica 6.1.2 server is supported with HP Vertica 6.1.2 MC only. This is a general MC requirement and is not specific to MC on AWS.

Using MC with the Community Edition

In HP Vertica 6.1.2, Community Edition customers can install and use Management Console to create, import, manage, and monitor a database cluster with the 1 TB/3-node limit. See Management Console in the Concepts Guide to learn about MC.

Community Edition Support for an AWS Cluster

If your Amazon Web Services (AWS) cluster is running HP Vertica Release 6.1.2 or later, you can use the expanded HP Vertica Community Edition, which is the free version of the HP Vertica Analytics Platform. The free version remains limited to three nodes and up to 1 TB of data. Refer to HP Vertica Community Edition for more information on Community Edition.

New and Modified System Tables in 6.1.2

This section lists new and modified system tables introduced in HP Vertica 6.1.2. See the SQL Reference Manual for details.

Enhanced in V_MONITOR schema

The SESSIONS table's CLIENT_LABEL column now outputs an MC value when there is a client connection to an MC-managed database for that USER_NAME.

New Features and Changes in HP Vertica 6.1 SP1 (6.1.1)

Read the topics in this section for information about new and changed functionality in HP Vertica 6.1 SP1 (6.1.1).

Improved Diagnostics Collection

A new diagnostics-collection tool introduced in 6.1 SP1 (Version 6.1.1), called Scrutinize, combines and extends the functionality of pre-6.1 diagnostics utilities and lets you send information about your HP Vertica environment to your technical support contact faster than ever before.

Scrutinize provides the following benefits:

- Gathers more complete diagnostics information, whether the database is down or up
- Generates a faster, more lightweight collection with minimal impact on a running database
- Is more configurable than the pre-6.1 diagnostics tools:
 - Lets you use arguments to collect information related to specific or immediate needs
 - Simplifies any further, detailed investigation that your HP Vertica support contact might request
- Can automatically upload files to HP Vertica support if you supply an upload argument to the script

For more information, see Collecting Diagnostics (scrutinize Command) in the Administrator's Guide.

Note: Although the HP Vertica diagnostics collection tools (`diagnostics` and `collect_diag_dump.sh` scripts) are still available, the new `scrutinize` script provides much more flexibility and functionality. HP recommends that you use `scrutinize`.

New Configuration Parameters

This section lists any new configuration parameters. See the Configuration Parameters section in the Administrator's Guide for details.

CopyFromVerticaWithIdentity

HP Vertica 6.1 SP1 (Version 6.1.1) adds the `CopyFromVerticaWithIdentity` configuration parameter to change the default behavior for the `COPY FROM VERTICA` and `EXPORT TO VERTICA` statements. These statements now load data into an Identity or Auto-increment column when you specify the Identity columns to load. Previously, `IDENTITY` columns were silently ignored if they existed in the source table, and were missing in the destination table.

See ***COPY FROM VERTICA and EXPORT TO VERTICA*** (page [15](#)).

New SEARCH_PATH User Setting

The default search path that HP Vertica uses to find tables and User Defined Function (UDF)s referenced without a schema can be set in 6.1 SP1 (Version 6.1.1) on a per-user basis. The `CREATE USER` and `ALTER USER` statements have a new `SEARCH_PATH` argument to set the list of schemas HP Vertica searches when it find an unqualified reference to a table or UDFs. See *Setting Search Paths in the Administrator's Guide* for more information.

Changes in the HP Vertica Catalog Directory

In 6.1 SP1 (Version 6.1.1), the following files in the catalog directory have been enhanced to make it easier to trace operations, as well as quickly determine the initiator node of a particular operation:

- `vertica.log` includes the current transaction ID, in hexadecimal format, after the thread name
- `ErrorReport.txt` also includes the transaction ID (also in hex format), if known

For additional information about files that reside in the catalog directory, see *Understanding the Catalog Directory in the Administrator's Guide*.

Statistics Gathering When Node is Down

In previous releases, you could not gather statistics when a cluster node was down. In 6.1 SP1 (Version 6.1.1) you can now run the `ANALYZE_STATISTICS()` and `ANALYZE_HISTOGRAM()` functions, as well as gather rowcount statistics by running the `DO_TM_TASK('analyze_row_count')` utility. HP Vertica gathers statistics for all nodes and returns the correct minimum/maximum values when the down node comes back up.

MC Database Naming Requirement Changes

In previous Management Console releases, a database that you created through the MC interface could have a name between 3-25 alphanumeric characters. In HP Vertica 6.1 SP1 (Version 6.1.1), database names for both MC and the Administration Tools must conform to the following naming requirements:

- Between 1-30 characters
- Begin with a letter
- Follow initial letter with any combination of:
 - letters (upper and lowercase)
 - numbers
 - underscores

Grant/Revoke Privileges on all Objects Within a Schema

Instead of having to grant privileges to users or roles one object at a time, in 6.1 SP1 (Version 6.1.1) you can grant or revoke privileges to or from *all* tables or sequences or functions within one or more schemas to a user/role in a single statement.

The new syntax is:

```
... ON ALL { FUNCTIONS | SEQUENCES | TABLES } IN SCHEMA
```

See the following statements in the SQL Reference Manual for details:

- GRANT (User Defined Extension)
- GRANT (Sequence)
- GRANT (Table)
- REVOKE (User Defined Extension)
- REVOKE (Sequence)
- REVOKE (Table)

Backup and Restore Changes

These new features and changes are available as of HP Vertica 6.1 SP1 (Version 6.1.1).

New Progress Indicator Bar

When using the `vbr.py` utility to create a snapshot, or use `copycluster`, `vbr` displays a textual progress indicator as follows:

```
> vbr.py --task backup --config-file myconfig.ini
```

```
Copying...
[=====] 100%
.
.
.
```

New vbr Option to List Backups

You can now list existing backups associated with a configuration file, using the `vbr.py` task option `--listbackup`. For more information about `vbr.py` parameters, see [VBR Utility Reference](#).

New vbr --config-file question for multiple database choice

The `vbr.py` utility now detects multiple databases exist if they are defined in the `admintools.conf` file. When using the tool to create the required parameters for the backup configuration file, the `vbr.py` lists the databases it finds so you can choose one to backup from the following message:

```
Choose from these databases: HDFSCON,hdfscon1,HADOOP (HDFSCON):
```

New Column to View Object-level Snapshots

The `DATABASE_BACKUPS` table has a new column that lists all existing object-level snapshots.

New Backup Hosts SSH Port Parameter

Internally, the `vbr.py` utility uses the default SSH port 22. You can now override this setting by manually adding the `ssh_port_backup` parameter to the Transmission section of the backup configuration file. Specify the SSH port that all backup hosts are using. For more information, see [Configuring Backup Hosts and VBR Configuration File Reference](#).

NOTE: Changing the SSH port value is supported only when using the `backup` and `restore` tasks. Using a non-default SSH port with the `copycluster` task is not supported.

Compatibility Between 5.x and 6.x Backups

If you have upgraded from HP Vertica 5.x to 6.x, backups created with 5.x are incompatible with 6.x. Once you have created new backups with the 6.x `vbr.py` utility, consider removing any created with the 5.x version.

--dry-run Parameter

The `vbr.py` utility `--dry-run` parameter has been removed from the documentation.

HP Vertica 6.1 SP1 (Version 6.1.1) adds a new library to support parallel loading of delimited files. This section refers to the functionality generically as *Pload*.

The Pload feature is ideal for loading large data files (10s of GBs), to significantly reduce file load time by dividing file parsing tasks across each core on the server where the data file resides. For instance, given a 12GB file to load, performance could be 3 - 5 times faster than loading the file without Pload.

You can specify the file division size (in bytes) by supplying an integer value to the `pload chunk_size` parameter.

After installing the library, you can use `COPY` (not `COPY LOCAL`) with the new option, `WITH SOURCE PloadDelimitedSource`.

For more information about using Pload, see [Using the Parallel Load Library in the Administrator's Guide](#).

Query Run-Time Priority Changes

HP Vertica 6.1 SP1 (Version 6.1.1) introduces a number of changes to simplify the process used to change the run-time priority of a query.

New Function: `CHANGE_CURRENT_STATEMENT_RUNTIME_PRIORITY`

HP Vertica 6.1 SP1 (Version 6.1.1) simplifies the process for changing the priority of a running query. This release introduces a new function that allows you to change the priority of a running query without specifying the statement ID. Using the `CHANGE_CURRENT_STATEMENT_RUNTIME_PRIORITY` function, you can change the priority of the query that is currently running within a transaction.

NULL Values Now Accepted for `CHANGE_RUNTIME_PRIORITY` Statement IDs

The `CHANGE_RUNTIME_PRIORITY` function (introduced in 6.0) now accepts NULL values for the statement ID. When you specify NULL for the statement ID, HP Vertica changes the run-time priority of the currently running query in the session. While you can continue to use this function in the short term, you should begin using the new `CHANGE_CURRENT_STATEMENT_RUNTIME_PRIORITY` function instead. `CHANGE_RUNTIME_PRIORITY` will be deprecated in a future release.

Changing the Run-Time Priority of a Query That Has Not Yet Begun Executing Results In An Error

In previous releases, HP Vertica did not prevent you from trying to change the run-time priority of a query before it had started (for example, a query waiting for a table lock to clear.) In previous releases, when you tried to change the run-time priority of such queries, HP Vertica did not display an error.

In 6.1 SP1 (Version 6.1.1), HP Vertica now issues an error if you try to change the run-time priority of queries that have not begun executing.

See Also

- [Changing Run-Time Priority of a Running Query](#)
- `CHANGE_CURRENT_STATEMENT_RUNTIME_PRIORITY`
- `CHANGE_RUNTIME_PRIORITY`

Improved Performance When Inserting Data into Segmented Projections

In 6.1 SP1 (Version 6.1.1), HP Vertica achieves improved performance through better parallelism when inserting data into a segmented projection. However, sometimes this performance increase comes at the cost of using more resources.

This new behavior is on by default and only affects systems that have local segments enabled. This feature can be turned off by setting the configuration parameter `ParallelizeLocalSegmentLoad` to `false`.

Improved Performance for COMMIT Operations

HP Vertica 6.1 SP1 (Version 6.1.1) includes improved performance of the COMMIT statement.

COPY FROM VERTICA and EXPORT TO VERTICA Improvements

In HP Vertica 6.1 SP1 (Version 6.1.1), the COPY FROM VERTICA and EXPORT TO VERTICA statements support loading data into an Identity or Auto-increment column when you specify an Identity column to load in both source and destination tables. Previously, IDENTITY columns were silently ignored if they existed in the source table, and were missing in the destination table. Loading into an Identity column does not increment the value automatically, and requires invoking ALTER SEQUENCE.

In this release, failure to list which Identity columns to load can cause an error, because Identity columns are no longer ignored and may be missing in the destination table.

If the source table includes an IDENTITY column and you do not specify the column in both source and destination, the source table will be identified as having an additional column (the IDENTITY column).

Depending what behavior you require, you can:

- Update existing queries and scripts to either insert into the destination's IDENTITY column, or omit specifying the IDENTITY column in the source table.
- Globally disable the new behavior by setting the configuration parameter to zero (0) as follows:

```
SELECT SET_CONFIG_PARAMETER('CopyFromVerticaWithIdentity', 0);
```

For more information, see *Moving Data Between HP Vertica Databases* in the Administrator's Guide.

DROP_PARTITION Updates

In HP Vertica 6.1 SP1 (Version 6.1.1), the DROP_PARTITION function has been changed to return a `Partition does not exist` message, if the partition parameter does not equate to an existing partition.

For more information, see:

- DROP_PARTITION in the SQL Reference Manual
- Dropping Partitions in the Administrator's Guide

New EXPORT_STATISTICS Argument

Previously, the EXPORT_STATISTICS() function took one argument that exported all database statistics to an XML file. In HP Vertica 6.1.1, a new parameter lets you export more specific database objects, which you can use if you notice that statistics in the database do not match the statistics of data. See EXPORT_STATISTICS in the SQL Reference Manual.

PURGE_TABLE Behavior Change

Using the PURGE_TABLE() function to purge a temporary table previously returned a misleading error message. In HP Vertica 6.1 SP1 (Version 6.1.1), attempting to purge a temporary table now returns the correct error:

```
ERROR 0: Purge is not allowed on temporary tables
```

ALTER TABLE Updates

Earlier releases required two separate statements to add a table column with a NOT NULL clause:

```
alter table mytable add column int newcol;  
alter table mytable alter column newcol set default 5 not null;
```

HP Vertica 6.1 SP1 (Version 6.1.1) lets you do this in one statement:

```
alter table mytable add column b int default 5 not null;
```

For more information, see:

ALTER TABLE in the SQL Reference Manual

Adding Constraints in the Administrator's Guide

In previous versions, COPY (or COPY LOCAL) only populated EXCEPTIONS and REJECTED DATA files if the COPY statement succeeded.

This behavior has changed in HP Vertica 6.1 SP1 (Version 6.1.1). Using the `EXCEPTIONS` and `REJECTED DATA` parameters to specify one or both file locations causes `COPY` to populate the files, even if the `COPY` transaction fails.

NOTE: `COPY` does not leave empty files at the location you specify if no exceptions or rejections occur.

For more information, see `COPY` and `COPY LOCAL` in the SQL Reference Manual.

Also, see Bulk Loading in the Administrator's Guide

New System Table Column Identifiers

In HP Vertica 6.1 SP1 (Version 6.1.1), several system tables contain a new column identifier, which makes it easier to join related tables.

These identifiers are synthetic `VARCHAR` columns composed of the the `TABLE_ID` column and the column position in the `SELECT` list. Column output will resemble `'45035996273933982-3'`.

For details, see the following topics in the SQL Reference Manual:

- `V_CATALOG.COLUMNS.COLUMN_ID`
- `V_CATALOG.PROJECTION_COLUMNS.TABLE_COLUMN_ID`
- `V_CATALOG.SYSTEM_COLUMNS.COLUMN_ID`
- `V_CATALOG.VIEW_COLUMNS.COLUMN_ID`
- `V_MONITOR.COLUMN_STORAGE.ANCHOR_TABLE_COLUMN_ID`

See Also

See *New and Modified System Tables in 6.1.0* (page [31](#)) in this guide for information about other system table enhancements and additions.

Monitoring Real-time Data in Query Plan Paths

In HP Vertica 6.1 SP1 (Version 6.1.1), you can now monitor the real-time flow of data and the time and resources consumed for each path in each query plan.

See the topics in the following books:

- `QUERY_PLAN_PROFILES` in the SQL Reference Manual
- Profiling query plan profiles in the Administrator's Guide

New and Modified System Tables in 6.1.1

This section lists new and modified system tables introduced in HP Vertica 6.1.1. See the SQL Reference Manual for details.

Note: Many of table updates in this release provide new column identifiers to make it easier to join related tables. See *New System Table Column Identifiers* (page [17](#)) for more information.

New and enhanced in V_CATALOG schema

- ALL_TABLES has a new TABLE_ID column.
- COLUMNS has a new COLUMN_ID column.
- CONSTRAINT_COLUMNS has the following new columns:
 - CONSTRAINT_ID
 - TABLE_ID
 - REFERENCE_TABLE_ID
- EPOCHS provides the date and time of the close and the corresponding epoch number of the closed epoch.
- GRANTS has the following new columns:
 - GRANT_ID
 - GRANTEE_ID
 - GRANTOR_ID
 - OBJECT_ID
 - OBJECT_TYPE
- NODES.EXPORT_ADDRESS column now shows an IP value instead of an object ID
- PROJECTION_COLUMNS has the following new columns:
 - PROJECTION_ID
 - TABLE_ID
 - TABLE_COLUMN_ID
- QUERY_PLAN_PROFILES provides detailed execution status for queries that are currently running in the system. Output from the table shows the real-time flow of data and the time and resources consumed for each path in each query plan.
- RESOURCE_POOL_DEFAULTS has a new POOL_ID column.
- RESOURCE_POOLS has a new POOL_ID column.
- ROLES has a new ROLE_ID column.
- STORAGE_LOCATIONS has a new LOCATION_ID column.
- SYSTEM_COLUMNS has a new COLUMN_ID column.
- SYSTEM_TABLES has the following new columns:
 - TABLE_ID
 - TABLE_SCHEMA_ID

- USERS has a new VARCHAR column, SEARCH_PATH, which shows the default schema search path for the user. See Setting Schema Search Paths in the Administrator's Guide for more information.
- VIEW_COLUMNS has a new COLUMN_ID column.
- VIEWS has the following new columns:
 - TABLE_ID
 - CREATE_TIME

New and enhanced in V_MONITOR schema

- COLUMN_STORAGE has the following new columns:
 - PROJECTION_ID
 - COLUMN_ID
 - ANCHOR_TABLE_ID
 - ANCHOR_TABLE_COLUMN_ID
 - ANCHOR_TABLE_COLUMN_NAME
- CRITICAL_NODES has a new NODE_ID column.
- DATABASE_BACKUPS has a new OBJECTS column.
- LOAD_STREAMS has a new TABLE_ID column.
- NODE_STATES has a new NODE_ID column.
- PARTITION_REORGANIZE_ERRORS has a new PROJECTION_ID column.
- PROJECTION_REFRESHES has a new PROJECTION_ID column.
- PROJECTION_STORAGE has a new PROJECTION_ID column.
- QUERY_EVENTS has the following new columns:
 - OPERATOR_NAME
 - PATH_ID
- RESOURCE_ACQUISITIONS has the following new columns:
 - POOL_ID
 - REQUEST_TYPE
- RESOURCE_REJECTION_DETAILS has a new POOL_ID column.
- RESOURCE_REJECTIONS has a new POOL_ID column.
- STORAGE_LOCATIONS has a new LOCATION_ID column
- STRATA has the following new columns:
 - PROJECTION_ID
 - MERGING_STRATA_COUNT
- STRATA_STRUCTURES has a new PROJECTION_ID column.
- SYSTEM_SESSIONS has a new STATEMENT_ID column.
- UDX_FENCED_PROCESSES has a new column named LANGUAGE that details the language of the fenced mode UDX that is running in a side process.
- USER_SESSIONS has the following new columns:

- TRANSACTION_ID
- STATEMENT_ID

New Features and Changes in HP Vertica 6.1.0

Read the topics in this section for information about new and changed functionality in HP Vertica version 6.1.0.

EXPORT_OBJECTS and EXPORT_CATALOG Changes

In previous releases, using the `EXPORT_OBJECTS` and `EXPORT_CATALOG` functions did not produce SQL scripts that used the original projection definitions. For example, if a table had two projections that were buddies of each other, running `export_objects` on the table resulted in this sort of projection definition, even if the original definition included a `KSAFE` clause:

```
create projection p_b0 ... segmented by ... all nodes offset 0;
create projection p_b1 ... segmented by ... all nodes offset 1;
```

In HP Vertica 6.1.0, running either the `EXPORT_OBJECTS` or `EXPORT_CATALOG` functions now always attempt to create scripts that include the original projection clauses, including `KSAFE` if it was used, or `offset` if it was not, producing a statement like this:

```
create projection p ... segmented by ... all nodes ksafe 1;
```

EXPORT_OBJECTS and Storage Policies

The new storage location capability to create labeled locations for use in storage policies saves the policy as a database object. In some cases, the objects are associated with tables. However, using `EXPORT_OBJECTS()` for a table does not include its storage policy, because the policy relates to a storage location, which may not be relevant when exporting a table definition for use elsewhere.

For more information, see *Working With Storage Locations* in the *Administrator's Guide*.

Database Designer Improvements

HP Vertica 6.1.0 offers improvements to the design, performance, and usability of Database Designer (DBD), speeding up design and deployment operations with minimal impact to design quality.

Performance

The following improvements have been made to the Database Designer during the query optimization and encoding (storage) optimization phases

- During the sort order enumeration phase, DBD now completes candidate projection cleanup more efficiently, improving database performance.
- During the encoding optimization phase, the Database Designer:

- Uses local temporary tables to determine the best encoding; during this phase, DBD also takes fewer locks than in previous releases.
- Reduces the encoding sample row count and storage band count.
- Optimizes storage for wide tables using fewer resources, which allows for the design process to complete faster.
- Improves encoding heuristics; for example, choosing the best encoding type for a particular data type or skipping columns with only one encoding option.

Note: DBD no longer tries RLE as an encoding choice for PK projection columns.

Design

As part of comprehensive design deployment, the Database Designer now refreshes projections in parallel, rather than in serial mode. Deployment is not limited by the concurrency setting of the refresh pool, and designs can be deployed more efficiently and quickly. During this operation, the refresh pool settings are cached and then automatically set to the maximum allowed settings by the system, in order to let the design process run quickly. When the design completes, the refresh resource pool settings automatically return to their pre-deployment value. There's no action required on your part to benefit from this enhanced design.

During an incremental design, the deployment process respects the concurrency setting of the refresh pool, which allows for improved performance, but deployment does not implicitly adjust the concurrency at will.

Usability

The Database Designer introduces the following support for input queries files:

- Can reference system tables, without creating design output for system tables; however, each input query must reference at least one catalog system table.
- Accepts and executes SET SEARCH_PATH statement in the input queries file. The default schema can change midway through a queries file, and there is no limit to the number of times the path changes in a queries input file.
- If an input query has a constant predicate on a Primary Key column, DBD now proposes that projections be segmented on the PK column. These benefits are extended to high cardinality, non-Primary Key columns and nonessential nodes are ignored when estimating cost.

SQL support

The REFRESH() function now attempts to refresh all the tables provided as arguments in parallel. Such calls are part of the Database Designer deployment (and deployment script).

Additionally, the V_CATALOG.PROJECTIONS system table contains a new IS_SUPER_PROJECTION column, which returns a Boolean value, indicating *true* if the projection is a superprojection.

New Hardware Verification Tests

HP Vertica 6.1.0 introduces new platform verification tests to aid in performance troubleshooting. The new tests are:

- `vcpuperf` - a CPU performance test used to verify CPU performance on a host.
- `vioperf` - an Input/Output test used to verify the speed and consistency of your hard drives.
- `vnetperf` - a Network test used to test the latency and throughput of your network between hosts.

The 6.1.0 installer uses a subset of these tests to verify basic performance before HP Vertica is installed. However, you can manually run the tests on one or more nodes to verify performance or help pinpoint an issue during performance troubleshooting.

Complete details are available in the Installation Guide in the section Validation scripts.

Management Console Enhancements in 6.1.0

HP Vertica 6.1.0 introduced significant improvements to the Management Console user experience.

Using MC to Create an HP Vertica Cluster

MC now offers a Create Cluster operation, which lets you create a database cluster on any number of hosts that do not have HP Vertica already installed. The Create Cluster operation copies the HP Vertica RPM (or DEB) file to the hosts, installs HP Vertica, then organizes the nodes into a cluster. See [Creating an HP Vertica Cluster Using MC](#) in the Installation Guide for more information.

Managing Multiple Database Clusters on MC

Previously MC supported one database per cluster. In 6.1.0, you can create or import one or more databases for each MC-managed HP Vertica cluster.

HP Vertica still supports only one running database. See [Managing Database Clusters on MC](#) in the Administrator's Guide for details.

MC Access to Non-administrator users

Previous releases of Management Console supported one superuser who managed all MC-related operations through the MC interface. In this release, you can create additional MC users and grant them access to the MC interface and/or MC-managed databases by assigning them a role.

MC roles are specific to the MC interface in that they govern what MC users can see and do on MC when they log in. MC roles provide flexibility by allowing MC and database user accounts to be isolated and/or shared. For example, if multiple MC users all need the same level of access to an MC-managed database, an MC administrator can map those users to a single HP Vertica database user account.

See the following topics in the Administrator's Guide for details:

- About MC Users
- About MC Privileges and Roles

Authenticating MC User Access

Administrators can decide how MC will authenticate users who log in to MC through their browser. You can choose your organization's LDAP repository or have the MC authenticate users through username/password combination. See *Creating an MC user* in the Administrator's Guide.

Auditing User Activity on MC

You can monitor the MC-related activities users performed on the MC interface, such as starting/stopping a database or adding/removing/replacing cluster nodes. This information is available on a new **Audit Log** page. You can also export audit data to a file. See *Monitoring MC User Activity* in the Administrator's Guide.

Viewing Cluster Metrics on MC

New Key Performance Indicators let you view the health of an MC-managed HP Vertica database cluster. See *Monitoring Cluster Performance* in the Administrator's Guide.

Monitoring Database Activity Through Event Flags

Two new flags appear on the MC Activity page to help you quickly identify when a Workload Analyzer (WLA) or rebalance operation (RBL) occurred. See *Monitoring Query and Session Activity* in the Administrator's Guide.

HP Vertica 6.1.x AMI on AWS

Beginning with Release 6.1.0, users must run HP Vertica on Amazon Web Services (AWS) using an HP Vertica pre-configured Amazon Machine Image (AMI). The pre-configured AMI is pre-tested, saves users steps which would otherwise be needed for determining the size of and configuring an optimal AMI, and is recommended and fully supported by HP Vertica. If you were previously using a custom AMI, you should migrate to this new HP Vertica AMI.

Notes:

- HP Vertica does not recommend upgrading your HP Vertica version on your custom AMI. Migrate to the officially-supported HP Vertica 6.1 AMI.
- The HP Vertica-supplied AMI installs HP Vertica Enterprise Edition; the HP Vertica Community Edition is not supported on AWS at this time.
- Management Console is not supported in an HP Vertica 6.1.0 AWS configuration.

Prepare for creating an HP Vertica cluster on AWS by configuring components using the AWS interface. For information on each component and the order in which to configure them, see the guide, *Using HP Vertica 6.1 on Amazon Web Services*.

Privileges Required For Database Operations

HP Vertica 6.1.0 provides enhancements to its privilege model.

Tables

Operation	Grantor/Permissions
CREATE EXTERNAL TABLE AS COPY	CREATE and USAGE privileges on schema, and READ privileges on USER-accessible storage location from which external is copied, or on which it is created.

Storage locations

Users and roles without superuser privileges can run the COPY statement after a superuser creates a storage location with the USER keyword, indicating the area is accessible to non-DBAdmin users. Each user/role that needs access to the storage location must be granted READ and/or WRITE access to the area.

Privilege	Description
READ	Allows the user to copy data from files in the storage location into a table, including external tables.
WRITE	Allows the user to copy data to the specific storage location. Users with WRITE privileges can also save COPY statement exceptions and rejected data files to the specified storage location.

Authenticating Users With Ident

HP Vertica 6.1.0 allows you to configure HP Vertica client authentication to query an Ident server to allow system users password-less access to the database.

To configure Ident authentication, do one of the following:

- Edit the `ClientAuthentication` record in the `vertica.conf` file using Administration Tools.
- Set the `ClientAuthentication` parameter using the `SET_CONFIG_PARAMETER` function.

Warning: Ident responses can be easily spoofed by untrusted servers. Ident authentication should only take place on local connections, where the Ident server is installed on the same computer as the HP Vertica database server.

For detailed information, see Configuring Ident authentication in the Administrator's Guide.

New LDAP Client Authentication Options

HP Vertica 6.1.0 provides three new options for configuring LDAP as the authentication method for validating user name/password pairs.

New authentication modes

You can now configure two LDAP authentication modes for client authentication:

- **Bind:** The server binds to the full distinguished name (DN).
- **Bind and search:** The server binds to an LDAP directory and using the provided parameters, searches the directory to find the full distinguished name (DN) of a user.

Specifying multiple LDAP URLs

The `ClientAuthentication` record in the `vertica.conf` file can now contain multiple LDAP URLs. The following record instructs the LDAP server to search the entire directory (`basedn=dc=example,dc=com`) for a DN with an OU (office unit) attribute that matches `Sales`. If the search returns no results or otherwise fails, the LDAP server searches for a DN with the OU attribute that matches `Marketing`:

```
ClientAuthentication = host all 10.0.0.0/8 ldap
"ldap://ldap.example.com/search;basedn=dc=example,dc=com;OU=Sales"
"ldap://ldap.example.com/search;basedn=dc=example,dc=com;OU=Marketing"
```

To use LDAP for client authentication:

- You must be connected to one or more preconfigured LDAP servers.
- The LDAP directory must contain a record for each client you need to authenticate.

For detailed information about using LDAP authentication, see [Configuring LDAP authentication](#).

Reduced Network Latency for Locking and Commits

In version 6.1.0, the number of messages sent over the network to coordinate locking and commits between database nodes have been reduced. This change helps reduce the network traffic between the nodes in the HP Vertica cluster and increases the performance of these operations.

Storage Location Updates

You can now add labels to a new or existing storage location. Labeled locations can then be used in one or more storage policies. Each storage policy associates a database object with its default storage location. There are several new functions to support these updates, listed in **[Storage Location Functions](#)** (page [26](#)).

In addition to adding labels to storage locations, the existing `ADD_LOCATION` function now lets you add a storage location to all cluster nodes. Previous releases required that you add a storage location to each node individually. In this release, using an empty string (' ') for the `node` parameter adds a location to all cluster nodes. Omitting the function's `node` parameter creates the storage location on the initiator node.

For a description of using these capabilities, see *Working With Storage Locations* in the *Administrator's Guide*, and the *Storage Management Functions* in the *SQL Reference Manual*.

Storage Location Functions

To support new storage location capabilities, HP Vertica 6.1.0 adds or changes these functions:

Function	Description
<code>ADD_LOCATION</code>	Changed. Includes a new parameter, <i>storage_label</i> . Use the new parameter to identify storage locations and create object storage policies. This change is backward-compatible will not affect existing use of the function, since the new parameter is optional.
<code>ALTER_LOCATION_LABEL</code>	New. Adds a label to an existing storage location. This function can also change or remove an existing storage location label.
<code>SET_OBJECT_STORAGE_POLICY</code>	New. Creates a new storage policy, or changes an existing one by associating a database object with a labeled storage location.
<code>CLEAR_OBJECT_STORAGE_POLICY</code>	New. Clears an existing object storage policy.

See *Storage Location Updates* (page [25](#)).

For a description of using the new functions, see *Working With Storage Locations* in the *Administrator's Guide*. Also, see *Storage Management Functions* in the *SQL Reference Manual*.

CREATE TABLE LIKE Statement

HP Vertica 6.1.0 adds new functionality to the `CREATE TABLE` statement with the `LIKE` clause. You can optionally copy the original table's projections, too. You can use this statement as part of the procedure to archive partitions, and then drop them. This procedure is described in *Moving Partitions* in the *Administrator's Guide*.

Note: You cannot create a new table from a temporary table using the `CREATE TABLE...LIKE` statement.

Replicated Storage Policies

When using CREATE TABLE with its LIKE *existing_table* option, an associated storage policy for the table was only replicated when you also used the optional WITH PROJECTIONS clause. In HP Vertica 6.1 SP1 (Version 6.1.1), a storage policy for the existing table is always replicated.

For more information, see:

- CREATE TABLE in the SQL Reference Manual
- Creating a Table Like Another in the Administrator's Guide

New INTERSECT and EXCEPT Clauses for Set Operations

HP Vertica 6.1.0 supports the use of the set operators INTERSECT and EXCEPT, in addition to UNION [ALL], which were supported in the previous release.

The following query uses INTERSECT to return the names of male customers who live in Denver, Colorado from the VMart example database:

```
=> SELECT customer_name FROM customer_dimension
      WHERE customer_state = 'CO'
      INTERSECT
      (SELECT customer_name FROM customer_dimension
       WHERE customer_gender = 'Male')
      INTERSECT
      (SELECT customer_name FROM customer_dimension
       WHERE customer_city = 'Denver');
```

The following query uses EXCEPT to search the VMart example database and return information about all Connecticut-based customers who bought items through stores and whose purchases amounted to more than \$500, except for those customers who paid cash:

```
=> SELECT customer_key, customer_name FROM public.customer_dimension
      WHERE customer_key IN (SELECT customer_key FROM store.store_sales_fact
                            WHERE sales_dollar_amount > 500)
      EXCEPT
      (SELECT customer_key FROM store.store_sales_fact
       WHERE tender_type = 'Cash')
      AND customer_state = 'CT';
```

Note: MINUS is an alias for EXCEPT.

SQL WITH Clause Support

HP Vertica 6.1.0 supports SQL WITH clauses for use in a primary SELECT statement. Each WITH clause includes at least one SELECT statement, but not INSERT, DELETE, and UPDATE statements. Using WITH clauses can simplify complex queries and improve overall performance.

The total number of WITH clauses evaluated in a primary query is undefined.

For more information, see:

- WITH Clause in the SQL Reference Manual
- WITH Subqueries in SELECT in the Programmer's Guide

Improved Statistics Collection

A new path in EXPLAIN <query> output, called PREDICATE VALUE OUT-OF-RANGE, lets you know when a predicate falls outside the histogram range. Previously this path was named STALE STATISTICS.

Outer Join Improvements for Large Inputs

Previously, HP Vertica planned both right outer joins and left outer joins so that the preserved relation (tb2 in the example below) was treated as if it were the smaller relation.

```
(tb1 RIGHT OUTER JOIN tb2)
(tb2 LEFT OUTER JOIN tb1)
```

In certain cases, this behavior could lead to suboptimal plans, such as building a very large hash table from tb2, even when tb1 was much smaller.

In 6.1.0, the optimizer considers the size of both inputs when optimizing outer joins, just as it always has for inner joins. This enhancement can improve query performance in cases where the preserved relation—such as the inner side of a left outer join—is very large compared to the non-preserved side (the outer side of a left outer join).

Analytic Functions Using PARTITION BY Run in Parallel

Analytic functions using the `partition by` clause automatically run in parallel when possible to improve query performance.

Subquery DELETE/UPDATE Enhancements

The performance of DML (UPDATE/DELETE) statements with subqueries has been improved. To be eligible for DELETE optimization, all target table columns referenced in a DELETE or UPDATE statement's WHERE clause must be in the projection definition.

For example, the following simple schema has two tables and three projections:

```
CREATE TABLE tb1 (a INT, b INT, c INT, d INT);
CREATE TABLE tb2 (g INT, h INT, i INT, j INT);
```

The first projection references all columns in tb1 and sorts on column a:

```
CREATE PROJECTION tb1_p AS SELECT a, b, c, d FROM tb1 ORDER BY a;
```

The buddy projection references and sorts on column a in tb1:

```
CREATE PROJECTION tb1_p_2 AS SELECT a FROM tb1 ORDER BY a;
```

This projection references all columns in tb2 and sorts on column i:

```
CREATE PROJECTION tb2_p AS SELECT g, h, i, j FROM tb2 ORDER BY i;
```

Consider the following DML statement, which references tb1.a in its WHERE clause. Since both projections on tb1 contain column a, both are eligible for the optimized DELETE:

```
DELETE FROM tb1 WHERE tb1.a IN (SELECT tb2.i FROM tb2);
```

Restrictions

Optimized DELETES are not supported under the following conditions:

- With pre-join projections on nodes that are down
- With replicated and pre-join projections if subqueries reference the target table. For example, the following syntax is not supported:

```
DELETE FROM tb1 WHERE tb1.a IN (SELECT e FROM tb2, tb2 WHERE tb2.e =
tb1.e);
```

- With subqueries that do not return multiple rows. For example, the following syntax is not supported:

```
DELETE FROM tb1 WHERE tb1.a = (SELECT k from tb2);
```

CLEAR_PROFILING Function Change

The CLEAR_PROFILING() function now takes a single argument, 'type_of_profiling'. The 'scope' argument is deprecated and is no longer necessary.

If you run CLEAR_PROFILING with the 'scope' argument, HP Vertica might generate a warning, such as in the following example:

```
=> SELECT clear_profiling('ee', 'local');
WARNING 4246: Only GLOBAL scope is supported for clearing EE profiles
```

```
clear_profiling
```

```
-----  
Did not clear any profiling data  
(1 row)
```

Note: Until the functionality for 'scope' is removed entirely, HP Vertica may or may not process the second argument without error.

Improved Performance for the UNION Clause

The performance of UNION and UNION ALL queries has been improved by more fully distributing the work across the entire cluster.

User Defined Function Binary Compatibility

In previous versions, any upgrade to the HP Vertica server would require you to recompile your User Defined Functions (UDFs) libraries to be compatible with the new version. You also had to drop all UDFs and libraries from your catalog and then recreate them.

In Version 6.1.0, compiled UDFs libraries remain compatible between most minor HP Vertica server revisions. You will only need to recompile your UDF libraries if there is a revision to the HP Vertica SDK, which only takes place between major versions (for example, between 6.0 and 6.1). See Updating UDF Libraries in the Programmer's Guide for more information.

In addition, a new statement named ALTER LIBRARY lets you update a library without having to drop all of the functions that reference the library, making upgrades easier. See ALTER LIBRARY in the SQL Reference Manual for more information.

ADO.NET Connection Pooling

HP Vertica 6.1.0 supports connection pooling for ADO.NET connections. Details are available in the Programmer's Guide section ADO.NET connection properties.

HP Vertica Hadoop Connector Version for Hadoop 2.0/CDH4

Changes to Hadoop 2.0.0 (used by Cloudera Distribution of Hadoop version 4) have broken compatibility with earlier versions of the HP Vertica Hadoop Connector. A new version of the Connector has been compiled that is compatible with the new version of Hadoop.

See the *myVertica portal* <http://my.vertica.com/> for the new Connector installation package.

New Hadoop Distributed File System Connector

Version 6.1.0 introduces a new Hadoop Distributed File System (HDFS) Connector that lets you load files directly from a Hadoop cluster's file system into your HP Vertica database. Like the previously-released Hadoop Connector, the HDFS Connector is an additional package available for download from the *myVertica portal* <http://my.vertica.com/>.

See Using the Hadoop Distributed File System Connector in the Programmer's Guide for instructions on installing and using the HDFS Connector.

New and Modified System Tables in 6.1.0

This section lists new and modified system tables introduced in HP Vertica 6.1.0. See the SQL Reference Manual for details.

New and enhanced in V_CATALOG schema

- PROJECTIONS has a new Boolean column, `IS_SUPER_PROJECTION`, where true output indicates the projection is a superprojection.
- STORAGE_LOCATIONS supplies information about existing storage locations, their labels, and other information.

New and enhanced in V_MONITOR schema

- EXECUTION_ENGINE_PROFILES added the following `COUNTER_NAME` values:
 - Start time—Time (timestamp) when HP Vertica started to process the operation
 - End time—Time (timestamp) when HP Vertica stopped processing the operation
- PARTITIONS has a new `VARCHAR location_label` column, indicating the user-labeled location that the partition uses as its default storage location.
- STORAGE_CONTAINERS has a new `VARCHAR location_label` column. This new column indicates any storage locations with a user-defined label. Such labels typically indicate the characteristics of the location, such as `SSD` or `ARCHIVE`.
- STORAGE_POLICIES includes information about the current storage policies defined for your site.
- STORAGE_TIERS provides data totals about labeled locations across the cluster.

Deprecated Functionality

In this release, the following HP Vertica functionality has been deprecated:

- Volatility and NULL behavior parameters of CREATE FUNCTION statement
- CLEAR_PROFILING() 'scope' argument
- USER_TRANSFORMS system table
- IMPLEMENT_TEMP_DESIGN() function

See ***Retired Functionality Table*** (page [33](#)) for additional information.

Retired Functionality Table

The functionality listed in the following table has been made obsolete, deprecated, or removed from HP Vertica:

Functionality	Component	Obsolete Version	Deprecated Version	Removed Version
VT_ tables	Server	4.0	4.1	5.0
EpochAdvancementMode	Server	4.0	4.1	5.0
MANAGED load (server keyword and related client parameter)	Server, clients	4.1	5.0	
getNumAcceptedRows getNumRejectedRows	ODBC, JDBC clients	4.1	5.0	
use35CopyParameters	ODBC, JDBC clients	4.1	4.1	5.0
BatchAutoComplete	All clients	4.1	4.1	5.0
ReportParamSuccess	All clients	4.1	4.1	5.0
EnableStrataBasedMrgOutPolicy	Server	4.1	4.1	5.0
MergeOutPolicySizeList	Server	4.1	4.1	5.0
LCOPY (see Note section below table)	Server, clients	—	4.1 (Client) 5.1 (Server)	5.1 (Client)
backup.sh	Server	6.0		
restore.sh	Server	6.0		
copy_vertica_database.sh	Server	6.0		
Ganglia	Server	—	6.0	
Volatility and NULL behavior parameters of CREATE FUNCTION	Server	6.0	6.1	
USER_TRANSFORMS system table	Server	6.0	6.1	
RESOURCE_ACQUISITIONS_HISTORY system table	Server	—	—	6.0

New Features

<p>Query Repository, which includes: <code>SYS_DBA.QUERY_REPO</code> table</p> <p>Functions:</p> <ul style="list-style-type: none"> ▪ <code>CLEAR_QUERY_REPOSITORY()</code> ▪ <code>SAVE_QUERY_REPOSITORY()</code> <p>Configuration parameters:</p> <ul style="list-style-type: none"> ▪ <code>CleanQueryRepoInterval</code> ▪ <code>QueryRepoMemoryLimit</code> ▪ <code>QueryRepoRetentionTime</code> ▪ <code>QueryRepositoryEnabled</code> ▪ <code>SaveQueryRepoInterval</code> ▪ <code>QueryRepoSchemaName</code> ▪ <code>QueryRepoTableName</code> <p>See Notes section below table.</p>	Server	—	—	6.0
<code>IMPLEMENT_TEMP_DESIGN()</code>	Server, clients	—	6.1	
scope parameter of <code>CLEAR_PROFILING</code>	Server	—	6.1	

Terminology

- **Obsolete.** HP Vertica processes obsolete functionality the same way as regular functionality (without warnings, interruptions, error, and so on), but users are notified that the functionality will be deprecated in a future release.
- **Deprecated.** HP Vertica accepts the request to execute deprecated functionality, but the functionality result in a user notification, typically a warning message or error message. HP Vertica may process the functionality, or not; the behavior of the deprecated feature will be documented.
- **Removed.** The retired functionality is removed, and HP Vertica will no longer recognize that functionality as supported.

Notes

- **LCOPY:** Supported by the 5.1 server to maintain backwards compatibility with the 4.1 client drivers.
- **Query Repository:** You can still monitor query workloads with the following system tables:
 - `QUERY_PROFILES`
 - `SESSION_PROFILES`
 - `EXECUTION_ENGINE_PROFILES`

Version 6.0 introduced more robust, stable workload-related system tables, particularly:

- `QUERY_REQUESTS`
- `QUERY_EVENTS`
- `RESOURCE_ACQUISITIONS`
- The `RESOURCE_ACQUISITIONS` system table captures historical information.

Copyright Notice

Copyright© 2006-2013 Hewlett-Packard, and its licensors. All rights reserved.

Hewlett-Packard 150 CambridgePark Drive Cambridge, MA 02140 Phone: +1 617 386 4400 E-Mail: info@vertica.com Web site: http://www.vertica.com (http://www.vertica.com)

The software described in this copyright notice is furnished under a license and may be used or copied only in accordance with the terms of such license. Hewlett-Packard software contains proprietary information, as well as trade secrets of Hewlett-Packard, and is protected under international copyright law. Reproduction, adaptation, or translation, in whole or in part, by any means — graphic, electronic or mechanical, including photocopying, recording, taping, or storage in an information retrieval system — of any part of this work covered by copyright is prohibited without prior written permission of the copyright owner, except as allowed under the copyright laws.

This product or products depicted herein may be protected by one or more U.S. or international patents or pending patents.

Trademarks

HP Vertica™, the HP Vertica Analytics Platform™, and FlexStore™ are trademarks of Hewlett-Packard.

Adobe®, Acrobat®, and Acrobat® Reader® are registered trademarks of Adobe Systems Incorporated.

AMD™ is a trademark of Advanced Micro Devices, Inc., in the United States and other countries.

DataDirect® and DataDirect Connect® are registered trademarks of Progress Software Corporation in the U.S. and other countries.

Fedora™ is a trademark of Red Hat, Inc.

Intel® is a registered trademark of Intel.

Linux® is a registered trademark of Linus Torvalds.

Microsoft® is a registered trademark of Microsoft Corporation.

Novell® is a registered trademark and SUSE™ is a trademark of Novell, Inc., in the United States and other countries.

Oracle® is a registered trademark of Oracle Corporation.

Red Hat® is a registered trademark of Red Hat, Inc.

VMware® is a registered trademark or trademark of VMware, Inc., in the United States and/or other jurisdictions.

Other products mentioned may be trademarks or registered trademarks of their respective companies.

Information on third-party software used in HP Vertica, including details on open-source software, is available in the guide [Third-Party Software Acknowledgements](#).