KPI Metrics Data Dictionary

An Open Source Asset for use with TIBCO® Data Virtualization

Project Name	AS Assets KPI Metrics
Document Location	This document is only valid on the day it was printed. The source of the document will be found in the ASAssets_KPI folder (https://github.com/TIBCOSoftware)
Purpose	Self-paced instructional

www.tibco.com

Global Headquarters
3303 Hillview Avenue
Palo Alto, CA 94304

Tel: +1 650-846-1000
+1 800-420-8450
Fax: +1 650-846-1005

Revision History

Version	Date	Author	Comments
1.0	Jan 20 2020	Mike Tinius	Initial revision
1.2	Feb 25 2020	Mike Tinius	Added views vJobDeleteCheck, vMetadataSysAllResources. Added annotation to vMetadataDatasource[Arch],vMetadataResource[Arch], vMetadataResourceColumns[Arch]. Added layertype to vMetadataPrivilege[Arch]. Added message to vPolicyAssignment[Arch]
1.3	Apr 6 2020	Mike Tinius	Added views reportMetadataAllCount[Arch], metrics_requests_groupby_date, metrics_requests_groupby_nodehost_nodeport, metrics_resources_usage_groupby_date, metrics_resources_usage_groupby_nodehost_nodeport, metrics_sessions_groupby_date, metrics_sessions_groupby_nodehost_nodeport
1.4 – 2020.202	May 1 2020	Mike Tinius	Removed Archive views by adding partitioning to hold current and history in the same table.
1.5 – 2020.203	Jun 24 2020	Mike Tinius	Added “partition” column to each view that has been modified to use “daily” or “interval” partitioning.
1.6 – 2020.300	Aug 17 2020	Mike Tinius	Added notificationstatus to vEventRegLog
1.7 – 2021.101	Mar 11 2021	Mike Tinius	Added metrics collection roll-up section.
1.8 – 2022.100	Feb 23 2022	Mike Tinius	Added metrics_requests_hist_sqlplan and metrics_all_kpimetrics_table_counts
1.9 – 2022.201	May 25 2022	Mike Tinius	Removed references to LDAP which has been removed from KPImetrics. Added vSysCluster and vSysNodes.

Table of Contents

1 Introduction

Purpose

The purpose of this document is to provide a data dictionary for AS Assets KPI Metrics.

Audience

This document is intended to provide guidance for the following users:

- Data Virtualization Administrators – provides a guide for installation.
- Architects – provides the KPImetrics architecture.
- Data professionals – provides background on the published views and usage.
- Operations users – provides insight into triggers and procedures that are executed.
- Project Managers – provides general information on KPImetrics.

References

Product references are shown below. Any references to CIS or DV refer to the current TIBCO® Data Virtualization.

- TIBCO® Data Virtualization was formerly known as
    - Cisco Data Virtualization (DV)
    - Composite Information Server (CIS)

Overview

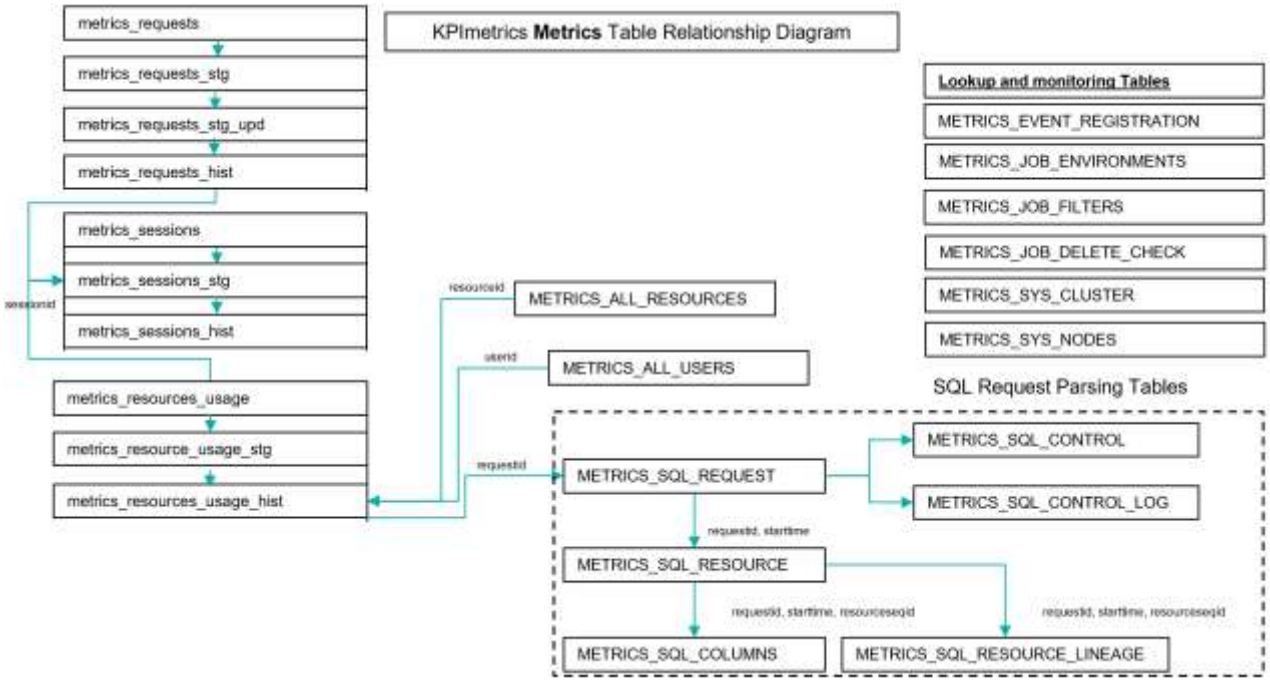
Please review the document "**KPImetrics Overview.pdf**".

2 Table Relationship Diagrams

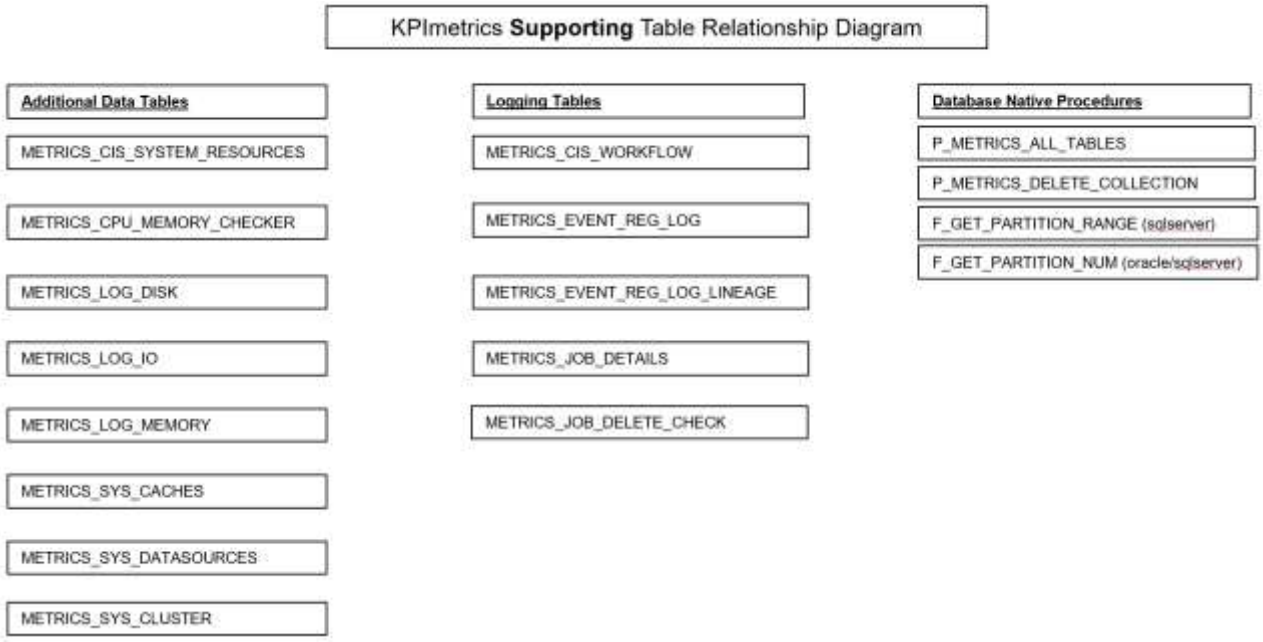
The following is the table relationship diagram for KPI metrics:

- The diagrams contain the following information:
  - KPImetrics **Metrics** table relationship diagram
    - KPImetrics **Supporting** table relationship diagram
    - Metrics Partitioning Strategy (w/leap year)
    - Metrics Partitioning Strategy (w/no leap year)
  - KPImetrics **Metadata** table relationship diagram
    - Metadata Partitioning Strategy (w/leap year)
    - Metadata Partitioning Strategy (w/no leap year)

KPImetrics "Metrics" Table Relationship Diagram



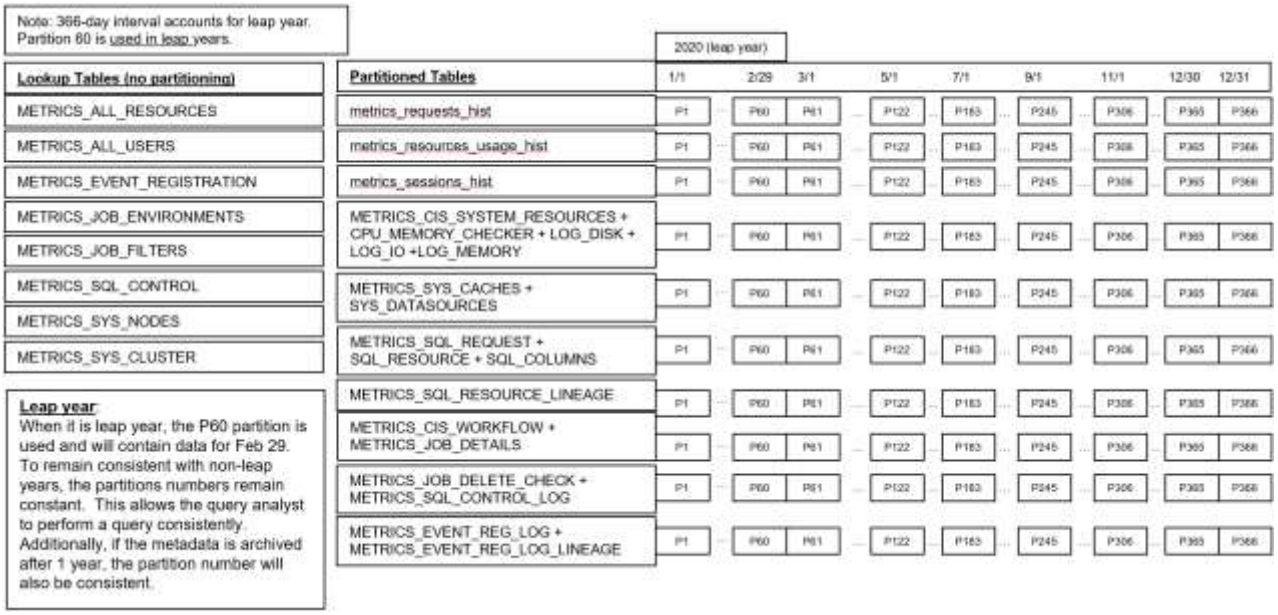
KPImetrics "Metrics" Supporting Table Relationship Diagram



KPI Metrics "Metrics" Tables Partitioning Strategy (w/leap year)

Leap year:

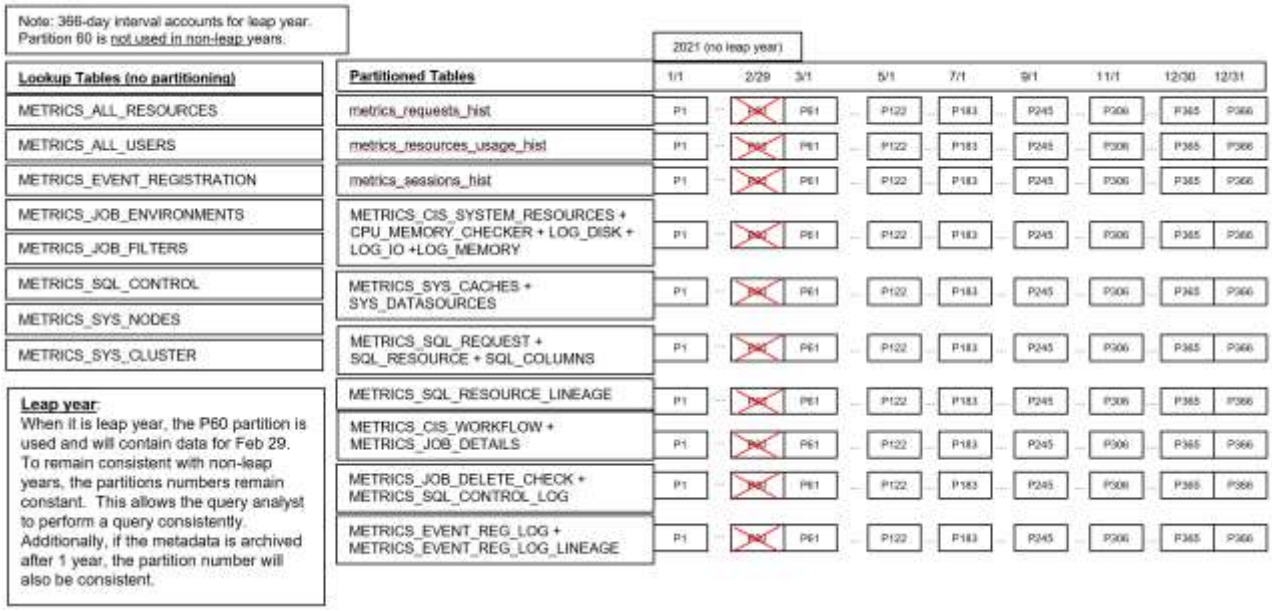
When it is leap year, the P60 partition is used and will contain data for Feb 29. To remain consistent with non-leap years, the partitions numbers remain constant. This allows the query analyst to perform a query consistently. Additionally, if the metadata is archived after 1 year, the partition number will also be consistent.



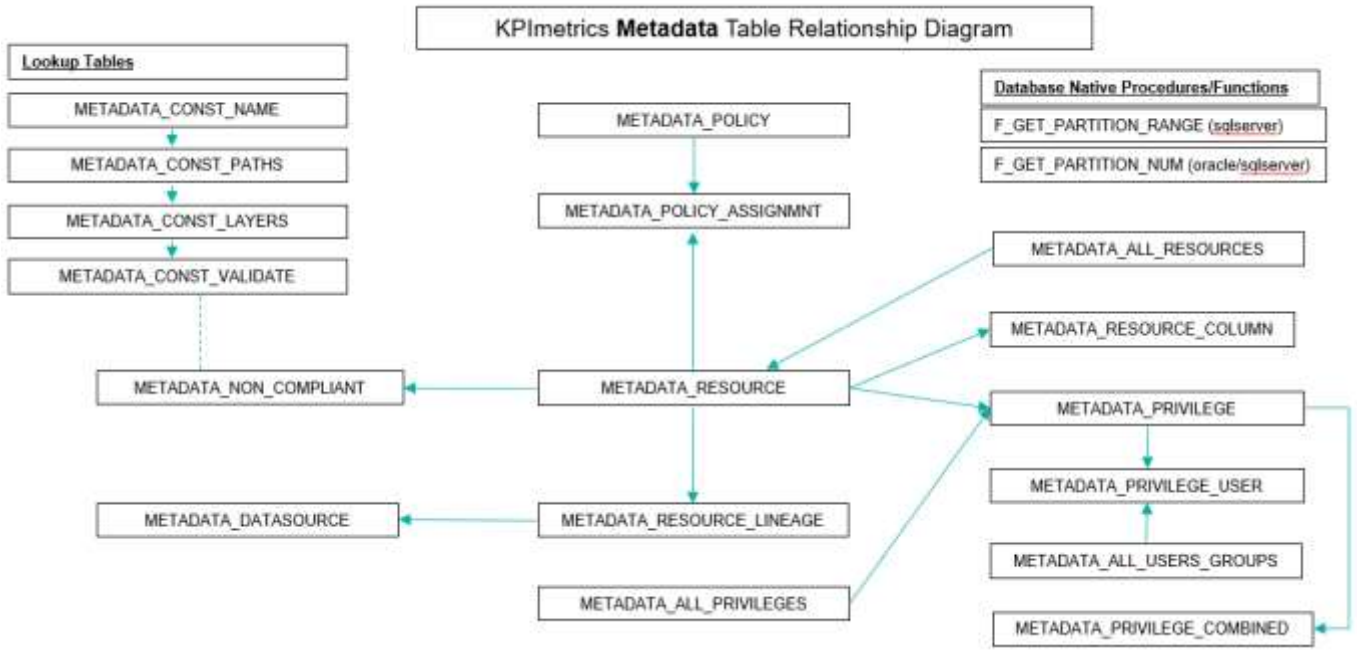
KPI Metrics "Metrics" Tables Partitioning Strategy (w/no leap year)

No Leap year:

When it is not leap year, the P60 partition is not used. To remain consistent with leap years, the partitions numbers remain constant. This allows the query analyst to perform a query consistently. Additionally, if the metadata is archived after 1 year, the partition number will also be consistent.



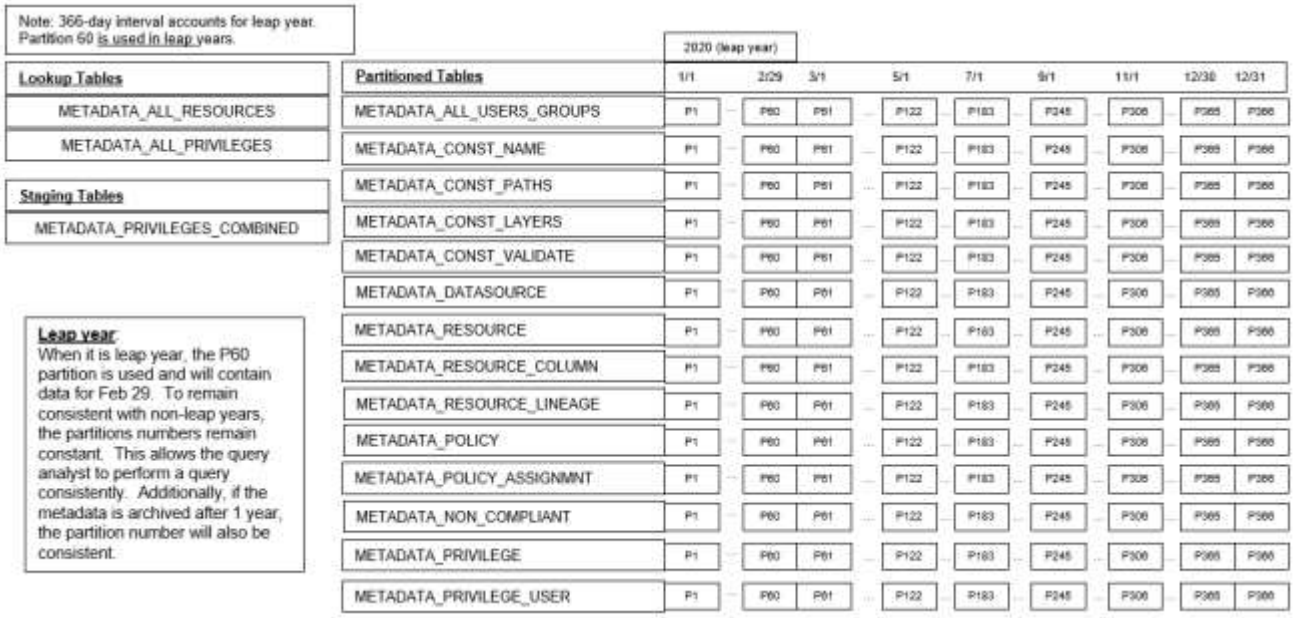
KPImetrics "Metadata" Table Relationship Diagram



KPI Metrics "Metadata" Tables Partitioning Strategy (w/leap year)

Leap year:

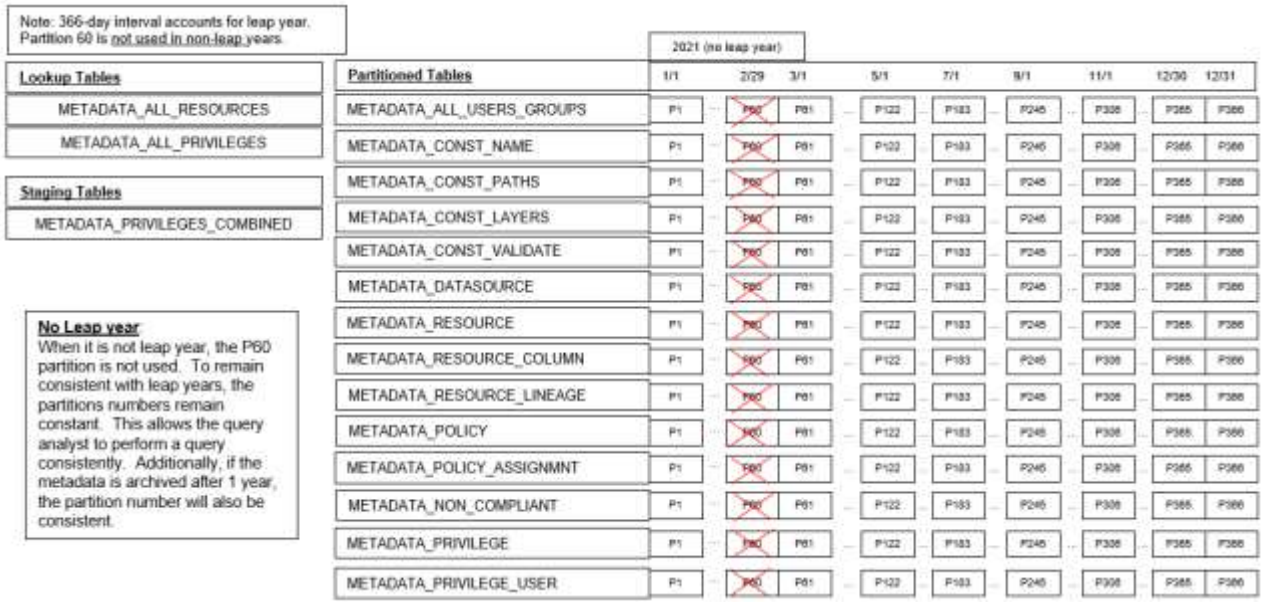
When it is leap year, the P60 partition is used and will contain data for Feb 29. To remain consistent with non-leap years, the partitions numbers remain constant. This allows the query analyst to perform a query consistently. Additionally, if the metadata is archived after 1 year, the partition number will also be consistent.



KPI Metrics "Metadata" Tables Partitioning Strategy (w/no leap year)

No Leap year:

When it is not leap year, the P60 partition is not used. To remain consistent with leap years, the partitions numbers remain constant. This allows the query analyst to perform a query consistently. Additionally, if the metadata is archived after 1 year, the partition number will also be consistent.



3 KPImetrics Data Dictionary

Table Definitions

This section provides information on the table definitions. This section provides table information regarding the ASAssets published database and KPImetrics catalog which is found at this location: /services/databases/ASAssets/KPImetrics

Column Definitions

This section provides information on the column definitions. This section provides column information regarding the ASAssets published database and KPImetrics catalog which is found at this location: /services/databases/ASAssets/KPImetrics. Any column with the [k] designation indicates that the field is part of a key which makes a row in that table unique. Any column with the [fk] designation indicates that the field is a foreign key to another table.

metrics_history – History Definitions

metrics_requests_hist (History) Table

Historical metrics requests table. Derived from metrics_requests_hist and metrics_resources_usage_hist. Expanded with user information and resourcekind and dataservicename.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
cluster	VARCHAR(255)	The name of the cluster or null if no cluster.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
parentid [fk]	DECIMAL(19,0)	A foreign key back to the requestid within this table. Contains -1 when no parent reference.
sessionid [fk]	DECIMAL(19,0)	A unique session id and foreign key to metrics_sessions_hist.
requesttype	VARCHAR(255)	The type of request that is stored in the description field. “SQL”=select statement, “SQL Script”=call statement executed, “XSLT, Basic, or Streaming Transformation”, “Java Procedure (built-in)”, etc.
description	CLOB	The actual client request or internal DV request.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
endtime	TIMESTAMP	The timestamp of when the request completed. E.g. 2020-01-06 10:19:36.899
totalduration	DECIMAL(19,0)	The total duration of the request measured in milli-seconds. Divide by 1000 to get minutes.
serverduration	DECIMAL(19,0)	The duration of the request within the server measured in milli-seconds. Divide by 1000 to get minutes.
rowsAffected	DECIMAL(19,0)	The number of rows affected or returned. The value may be -1.
maxmemory	DECIMAL(19,0)	The maximum amount of memory used by the request.
maxdisk	DECIMAL(19,0)	The maximum amount of disk used by the request. It is not good for a request to used disk.
message	CLOB	The exception message if a request was not successful.

status	VARCHAR(20)	The status of the request [SUCCESS, FAILED].
resourcekind	VARCHAR(20)	The type of resource that executed [system, user defined].
dataservicename	VARCHAR(255)	The published data service name.
userid	DECIMAL(19,0)	The DV userid from the system.ALL_USERS tables.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
group	VARCHAR(255)	The group of the user who connected.

metrics_requests_hist_sqlplan (History) Table

Historical metrics requests SQL Plan table. Derived from metrics_requests_hist and metrics_resources_usage_hist. Provides details about whether the SQL Plan is being pushed down to the database or not.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
planstatus	VARCHAR(20)	A foreign key back to the requestid within this table. Contains -1 when no parent reference.
message	VARCHAR(4000)	The exception message if a request was not successful.
rootnodetitle	VARCHAR(255)	The root node title is derived from the source query resource names with a "-" separator if there is more than one resource involved in the query but not to exceed 255 characters.
ispusheddown	VARCHAR(5)	true|false – specifies whether the query plan is pushed down to the database or not.
formattedsqlplan	CLOB	The formatted SQL Plan is generated and stored similar to the way it would be seen in TDV Studio by a developer.
description	CLOB	The actual client request or internal TDV request.

metrics_resources_usage_hist (History) Table

Historical metrics resources usage table. Derived from metrics_resources_usage_hist.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
cluster	VARCHAR(255)	The name of the cluster or null if no cluster.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
sessionid [fk]	DECIMAL(19,0)	A unique session id and foreign key to metrics_sessions_hist.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
group	VARCHAR(255)	The group of the user who connected.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
parentid [fk]	DECIMAL(19,0)	A foreign key back to the requestid within this table. Contains -1 when no parent reference.
datasourcepath	VARCHAR(255)	The DV path to the published datasource/dataservice [/services/databases/<name>, /services/webservices/<path|name>]

datasourcetype	VARCHAR(255)	The type of published datasource [VirtualRelational | VirtualWsdl | CompositeWebService].
resourcepath	VARCHAR(4000)	The DV path to the resource.
resourcetype	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
resourceguid	VARCHAR(40)	The guid of the resource path. Eg. cff5fe78-c3c5-4a81-b134-d497cea5351e
resourcekind	VARCHAR(20)	The type of resource that executed [system, user defined].
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
endtime	TIMESTAMP	The timestamp of when the request completed. E.g. 2020-01-06 10:19:36.899
resourceorigin	VARCHAR(30)	The original DV system table in which the resource originated from. [ALL_TABLES| ALL_PROCEDURES|ALL_WSDL_OPERATIONS|ALL_RESOURCES| ALL_COLUMNS|ALL_PARAMETERS]
resourceid [k]	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in "resourceorigin".
dataservicename	VARCHAR(255)	The published data service name.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
parentname	VARCHAR(255)	The parent name of the resource. A.k.a. schema name if from published database.
grandparentname	VARCHAR(255)	The grand-parent name of the resource. A.k.a. catalog name if from published database.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct ALL_RESOURCES: null
parentpath	VARCHAR(4000)	The parent path of the resource.
userid	DECIMAL(19,0)	The DV userid from the system.ALL_USERS tables.

metrics_sessions_hist (History) Table

Historical metrics sessions table. Derived from metrics_sessions_hist and metrics_resources_usage_hist. Expanded with user information if found.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
cluster	VARCHAR(255)	The name of the cluster or null if no cluster.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
sessionid [k]	DECIMAL(19,0)	A unique session id and primary identifier of this table.
sessiontype	VARCHAR(40)	Designates the type session connection [JDBC, HTTP, TASK, etc.].
clienthost	VARCHAR(255)	Designates the client host name that connected to DV.
type [k]	VARCHAR(20)	Indicates whether the session is of type 'begin' or 'end'.
logintime [k]	TIMESTAMP	The timestamp of when the session started.

logouttime	TIMESTAMP	The timestamp of when the session ended
status	VARCHAR(20)	The status of the session [ACTIVE, CLOSING].
totalduration	DECIMAL(19,0)	The total session duration in milli-seconds. Only on row of type='end' where status='CLOSING'.
totalRequests	DECIMAL(19,0)	The total number of requests [metrics_requests_hist] executed for this session.
bytestoclient	DECIMAL(19,0)	The total number of bytes sent to the client host during this session.
bytesfromclient	DECIMAL(19,0)	The total number of bytes received from the client host during this session.
userid	DECIMAL(19,0)	The DV userid from the system.ALL_USERS tables.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
group	VARCHAR(255)	The group of the user who connected.

metrics history roll-up tables

Resource (Resource Type)	Description
Metrics Roll-up Counts	
metrics_all_kpimetrics_table_counts (TABLE)	Provides a rollup of all counts by nodehost and nodeport for all metrics tables.
metrics_all_min_max_starttime_count (TABLE)	Provides a rollup of the min/max starttime/logintime, min/max requestid/sessionid and the total count of rows for each of the 6 metrics collection and historical tables.
metrics_history_tables_row_distribution (TABLE)	Provides a partition distribution for each partition month for each metrics history table.
metrics_history_other_tables_row_distribution (TABLE)	Provides a partition distribution for each partition month for each metrics non-history tables.
Metrics Requests History Roll-up	
metrics_requests_hist_groupby_date (TABLE)	Provides a rollup row count of the metrics_requests_hist table grouped by date.
metrics_requests_hist_groupby_date_month (TABLE)	Provides a rollup row count of the metrics_requests_hist table grouped by month.
metrics_requests_hist_groupby_date_month_nodehost_nodeport (TABLE)	Provides a rollup row count of the metrics_requests_hist table grouped by month, nodehost and nodeport.
metrics_requests_hist_groupby_date_nodehost_nodeport (TABLE)	Provides a rollup row count of the metrics_requests_hist table grouped by date, nodehost and nodeport.
metrics_requests_hist_groupby_date_user_domain (TABLE)	Provides a rollup row count of the metrics_requests_hist table grouped by date, user and domain.
metrics_requests_hist_groupby_date_user_domain_resourcekind (TABLE)	Provides a rollup row count of the metrics_requests_hist table grouped by date, user, domain and resourcekind [system, user defined].
metrics_requests_hist_groupby_date_user_domain_resourcekind_dataservicename (TABLE)	Provides a rollup row count of the metrics_requests_hist table grouped by date, user, domain and resourcekind [system, user defined] and dataservicename.
metrics_requests_hist_groupby_nodehost_nodeport (TABLE)	Provides a rollup row count of the metrics_requests_hist table grouped by nodehost and nodeport.
metrics_requests_hist_groupby_user_domain_nodehost_nodeport (TABLE)	Provides a rollup row count of the metrics_requests_hist table grouped by user, domain, nodehost and nodeport.
metrics_requests_hist_groupby_user_domain_resourcekind_dataservicename (TABLE)	Provides a rollup row count of the metrics_requests_hist table grouped by user, domain, resourcekind [system, user defined] and dataservicename.
metrics_requests_hist_groupby_user_domain_resourcekind_dataservicename_nodehost_nodeport (TABLE)	Provides a rollup row count of the metrics_requests_hist table grouped by user, domain, resourcekind [system, user defined], dataservicename, nodehost and nodeport.
Metrics Resources Usage History Roll-Up	

metrics_resources_usage_hist_groupby_date (TABLE)	Provides a rollup row count of the metrics_resources_usage_hist table grouped by date.
metrics_resources_usage_hist_groupby_date_month (TABLE)	Provides a rollup row count of the metrics_resources_usage_hist table grouped by month.
metrics_resources_usage_hist_groupby_date_month_nodehost_nodeport (TABLE)	Provides a rollup row count of the metrics_resources_usage_hist table grouped by month, nodehost and nodeport.
metrics_resources_usage_hist_groupby_date_nodehost_nodeport (TABLE)	Provides a rollup row count of the metrics_resources_usage_hist table grouped by date, nodehost and nodeport.
metrics_resources_usage_hist_groupby_date_user_domain_resourcekind (TABLE)	Provides a rollup row count of the metrics_resources_usage_hist table grouped by date, user, domain and resourcekind [system, user defined].
metrics_resources_usage_hist_groupby_nodehost_nodeport (TABLE)	Provides a rollup row count of the metrics_resources_usage_hist table grouped by nodehost and nodeport.
metrics_resources_usage_hist_groupby_user_domain_resourcekind (TABLE)	Provides a rollup row count of the metrics_resources_usage_hist table grouped by user, domain and resourcekind [system, user defined].
metrics_resources_usage_hist_groupby_user_domain_resourcekind_dataservicename	Provides a rollup row count of the metrics_resources_usage_hist table grouped by user, domain, resourcekind [system, user defined] and dataservicename.
Metrics Sessions History Roll-up	
metrics_sessions_hist_groupby_clienthost_nodehost_nodeport (TABLE)	Provides a rollup row count of the metrics_sessions_hist table grouped by clienthost, nodehost and nodeport.
metrics_sessions_hist_groupby_date (TABLE)	Provides a rollup row count of the metrics_sessions_hist table grouped by date.
metrics_sessions_hist_groupby_date_nodehost_nodeport (TABLE)	Provides a rollup row count of the metrics_sessions_hist table grouped by date, nodehost and nodeport.
metrics_sessions_hist_groupby_date_type (TABLE)	Provides a rollup row count of the metrics_sessions_hist table grouped by date and type='end'
metrics_sessions_hist_groupby_month (TABLE)	Provides a rollup row count of the metrics_sessions_hist table grouped by month.
metrics_sessions_hist_groupby_month_nodehost_nodeport (TABLE)	Provides a rollup row count of the metrics_sessions_hist table grouped by month, nodehost and nodeport.
metrics_sessions_hist_groupby_month_type (TABLE)	Provides a rollup row count of the metrics_sessions_hist table grouped by month and type='end'
metrics_sessions_hist_groupby_nodehost_nodeport (TABLE)	Provides a rollup row count of the metrics_sessions_hist table grouped by nodehost and nodeport.
metrics_sessions_hist_groupby_sessiontype (TABLE)	Provides a rollup row count of the metrics_sessions_hist table grouped by session type [TASK,INTERNAL,JDBC, etc].

metrics_count – Count Definitions

metrics_tables_row_distribution Table

Provides a metrics table row distribution for each table and partition.

Column Name	Column Type	Definition
table_name	BIGINT	The name of the table
partition_name	VARCHAR(255)	The name of the partition. Also known as the partition window.
boundary_definition	VARCHAR(255)	The date of the boundary high value.
range_type	TIMESTAMP	The range associated with the boundary: LEFT or RIGHT. If LEFT then rows are contained to the LEFT of the boundary definition.
partition_number	INTEGER	The partition number is associated with an interval integer representing the day of the year and is a value between 1 and 366. 366 days in a year accounts for leap year.

num_rows	INTEGER	The number of rows in the partition.

metrics_all_kpimetrics_table_countsTable

Provides a metadata table row distribution for each table and partition.

Column Name	Column Type	Definition
table_name	BIGINT	The name of the table
type	VARCHAR(255)	The description specifying the row type of GROUPBY or TOTAL.
countformatted	VARCHAR(255)	A formatted count with commas.
min_date	VARCHAR(255)	The minimum timestamp found in the table.
max_date	VARCHAR(255)	The maximum timestamp found in the table.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

metrics_collection – Collection Definitions

metrics_requests (Collection) Table

Native DV out-of-the-box metrics requests collection table. Derived from metrics_requests.

Column Name	Column Type	Definition
cluster	VARCHAR(255)	The name of the cluster or null if no cluster.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
parentid [fk]	DECIMAL(19,0)	A foreign key back to the requestid within this table. Contains -1 when no parent reference.
sessionid [fk]	DECIMAL(19,0)	A unique session id and foreign key to metrics_sessions.
requesttype	VARCHAR(255)	The type of request that is stored in the description field. “SQL”=select statement, “SQL Script”=call statement executed, “XSLT, Basic, or Streaming Transformation”, “Java Procedure (built-in)”, etc.
description	CLOB	The actual client request or internal DV request.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
endtime	TIMESTAMP	The timestamp of when the request completed. E.g. 2020-01-06 10:19:36.899
totalduration	DECIMAL(19,0)	The total duration of the request measured in milli-seconds. Divide by 1000 to get minutes.
serverduration	DECIMAL(19,0)	The duration of the request within the server measured in milli-seconds. Divide by 1000 to get minutes.
rowsAffected	DECIMAL(19,0)	The number of rows affected or returned. The value may be -1.
maxmemory	DECIMAL(19,0)	The maximum amount of memory used by the request.
maxdisk	DECIMAL(19,0)	The maximum amount of disk used by the request. It is not good for a request to used disk.
message	CLOB	The exception message if a request was not successful.
status	VARCHAR(20)	The status of the request [SUCCESS, FAILED].

metrics_resources_usage (Collection) Table

Native DV out-of-the-box metrics resources usage collection table. Derived from metrics_resources_usage.

Column Name	Column Type	Definition
cluster	VARCHAR(255)	The name of the cluster or null if no cluster.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
sessionid [fk]	DECIMAL(19,0)	A unique session id and foreign key to metrics_sessions.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
group	VARCHAR(255)	The group of the user who connected.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
parentid [fk]	DECIMAL(19,0)	A foreign key back to the requestid within this table. Contains -1 when no parent reference.
datasourcepath	VARCHAR(255)	The DV path to the published datasource/dataservice [/services/databases/<name>, /services/webservices/<path|name>]
datasourcetype	VARCHAR(255)	The type of published datasource [VirtualRelational | VirtualWsdl | CompositeWebService].
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
resourceguid	VARCHAR(40)	The guid of the resource path. Eg. cff5fe78-c3c5-4a81-b134-d497cea5351e
resourcekind	VARCHAR(20)	The type of resource that executed [system, user defined].
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
endtime	TIMESTAMP	The timestamp of when the request completed. E.g. 2020-01-06 10:19:36.899

metrics_sessions (Collection) Table

Native DV out-of-the-box metrics sessions collection table. Derived from metrics_sessions.

Column Name	Column Type	Definition
cluster	VARCHAR(255)	The name of the cluster or null if no cluster.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
sessionid [k]	DECIMAL(19,0)	A unique session id and primary identifier of this table.
sessiontype	VARCHAR(40)	Designates the type session connection [JDBC, HTTP, TASK, INTERNAL, etc.].
clienthost	VARCHAR(255)	Designates the client host name that connected to DV.
type [k]	VARCHAR(20)	Indicates whether the session is of type ‘begin’ or ‘end’.
logintime [k]	TIMESTAMP	The timestamp of when the session started.
logouttime	TIMESTAMP	The timestamp of when the session ended
status	VARCHAR(20)	The status of the session [ACTIVE, CLOSING].
totalduration	DECIMAL(19,0)	The total session duration in milli-seconds. Only on row of type=‘end’ where status=‘CLOSING’. Divide by 1000 to get minutes.
totalRequests	DECIMAL(19,0)	The total number of requests [metrics_requests_hist] executed for this session.
bytestoclient	DECIMAL(19,0)	The total number of bytes sent to the client host during this session.
bytesfromclient	DECIMAL(19,0)	The total number of bytes received from the client host during this session.

metrics collection roll-up tables

Resource (Resource Type)	Description
Metrics Requests Collection Roll-up	

metrics_requests_groupby_date (TABLE)	Rollup of metrics_requests collection data grouped by date. This can be useful to validate that metrics data is being written to the tables and how many records exist per day. Derived from metrics_requests
metrics_requests_groupby_nodehost_nodeport (TABLE)	Rollup of metrics_requests collection grouped by hostname and port. This can be usefule to validate that metrics is data is being written by each node in the cluster. It can also be used to detect data from other nodes that do not exist indicating that the database is being shared by another environment. Derived from metrics_requests.
Metrics Resources Usage Collection Roll-Up	
metrics_resources_usage_groupby_date (TABLE)	Provides a rollup row count of the metrics_resources_usage table grouped by date. This can be useful to validate that metrics data is being written to the tables and how many records exist per day. Derived from metrics_requests.
metrics_resources_usage_groupby_nodehost_nodeport (TABLE)	Provides a rollup row count of the metrics_resources_usage table grouped by nodehost and nodeport. This can be usefule to validate that metrics is data is being written by each node in the cluster. It can also be used to detect data from other nodes that do not exist indicating that the database is being shared by another environment. Derived from metrics_requests.
metrics_resources_usage_groupby_resourcekind (TABLE)	Provides a rollup row count of the metrics_resources_usage table grouped by resourcekind [system, user defined].
metrics_resources_usage_groupby_user_domain	Provides a rollup row count of the metrics_resources_usage table grouped by user and domain.
Metrics Sessions Collection Roll-up	
metrics_sessions_groupby_clienthost_nodehost_nodeport (TABLE)	Provides a rollup row count of the metrics_sessions table grouped by clienthost, nodehost and nodeport.
metrics_sessions_groupby_date (TABLE)	Provides a rollup row count of the metrics_sessions table grouped by date.
metrics_sessions_groupby_date_type (TABLE)	Provides a rollup row count of the metrics_sessions table grouped by date and type='end'. This can be useful to validate that metrics data is being written to the tables and how many records exist per day. Derived from metrics_requests.
metrics_sessions_groupby_month (TABLE)	Provides a rollup row count of the metrics_sessions table grouped by month and type='end'.
metrics_sessions_groupby_month_type (TABLE)	Provides a rollup row count of the metrics_sessions table grouped by month and type='end'.
metrics_sessions_groupby_nodehost_nodeport (TABLE)	Provides a rollup row count of the metrics_sessions table grouped by nodehost and nodeport. This can be usefule to validate that metrics is data is being written by each node in the cluster. It can also be used to detect data from other nodes that do not exist indicating that the database is being shared by another environment. Derived from metrics_requests.
metrics_sessions_groupby_sessiontype (TABLE)	Provides a rollup row count of the metrics_sessions table grouped by session type [TASK,INTERNAL,JDBC, etc] and type='end'.

AllCustomReports Definitions

AccessByUserOvertime [RT] Table

AccessByUserOvertimeRT – real time
Real-time report of the most active resources by a user over time. Sorted resource count, user and date. The algorithm for this report is shown below. The group by is the key to this report. The key grouping is on the user first and then the resource.
SELECT

"user", "domain", requestdate, resourcepath, resourcename, resourcetype, parentpath, resourcekind, datasourcetype, dataservicename, categoryname, nodehost, nodeport, COUNT(resourceid) countname,
CAST(ROUND(MONTHS_BETWEEN(CURRENT_DATE, requestdat),2) AS DECIMAL(19,2)) requestdatemonths,
CAST(DAYS_BETWEEN(CURRENT_DATE, requestdate) AS DECIMAL(19,0)) requestdatedays
FROM
/shared/ASAssets/KPImetrics/Business/Logical/resourceUsage/vResourceUsage
GROUP BY "user", "domain", requestdate, resourceid, resourcepath, resourcename, resourcetype, parentpath, resourcekind, datasourcetype, dataservicename, categoryname, nodehost, nodeport
Derived from the following tables: vResourceCountUsersDate (group by clause) ? vResourceUsage ? metrics_resources_usage_hist {pm}

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in “resourceorigin”.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
parentpath	VARCHAR(4000)	The parent path of the resource.
dataservicename	VARCHAR(255)	The published data service name.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct ALL_RESOURCES: null
countname	INTEGER	The number of resources for a given resource name.
requestdate	DATE	The date [YYYY-MM-DD] of the request.
requestdatemonths	DECIMAL(19,2)	The number of months difference between the request date and the current date.
requestdatedays	DECIMAL(19,0)	The number of days difference between the request date and the current date.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

ActiveResourcesOverPeriodOfTime [RT] Table

ActiveResourcesOverPeriodOfTimeRT – real time

Real-time report of the most active resources over time. Sorted by resource count and date.
SELECT
requestdate, resourcepath, resourcename, resourcetype, parentpath, resourcekind, datasourcetype, dataservicename, categoryname, nodehost, nodeport,
COUNT(resourceid) countname,
CAST(ROUND(MONTHS_BETWEEN(CURRENT_DATE, requestdate),2) AS DECIMAL(19,2)) requestdatemonths,
CAST(DAYS_BETWEEN(CURRENT_DATE, requestdate) AS DECIMAL(19,0)) requestdatedays
FROM
/shared/ASAssets/KPImetrics/Business/Logical/resourceUsage/vResourceUsage
GROUP BY requestdate, resourceid, resourcepath, resourcename, resourcetype, parentpath, resourcekind, datasourcetype, dataservicename, categoryname, nodehost, nodeport
Derived from the following tables: vResourceCountDate (group by clause) → vResourceUsage → metrics_resources_usage_hist {pm}

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in "resourceorigin".
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
parentpath	VARCHAR(4000)	The parent path of the resource.
dataservicename	VARCHAR(255)	The published data service name.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct ALL_RESOURCES: null
countname	INTEGER	The number of resources for a given resource name.
requestdate	DATE	The date [YYYY-MM-DD] of the request.
requestdatemonths	DECIMAL(19,2)	The number of months difference between the request date and the current date.
requestdatedays	DECIMAL(19,0)	The number of days difference between the request date and the current date.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

ResourceAccessByUsers [RT] Table

ResourceAccessByUsersRT – real time

Real-time report of the most used resources by a user with no time period. Derived from the following tables: vResourceCountUsers (group by clause) → vResourceUsage → metrics_resources_usage_hist {pm}

Column Name	Column Type	Definition
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in "resourceorigin".
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
parentpath	VARCHAR(4000)	The parent path of the resource.
dataservicename	VARCHAR(255)	The published data service name.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct ALL_RESOURCES: null
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
totalusagecount	BIGINT	The count of (resourceid) grouped by "user", "domain", resourceid, resourcepath, resourcename, resourcetype, parentpath, resourcekind, datasourcetype, dataservicename, categoryname, nodehost, nodeport
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

ResourceCount_Details [RT] Table

ResourceCount_DetailsRT – real time

Real-time detail report of resource count by date. Derived from the following tables: vResourceUsage → metrics_resources_usage_hist {pm}

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate	TIMESTAMP	The timestamp of when the record was inserted or loaded.
month	VARCHAR(255)	A month string indicating the month of the request.
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in "resourceorigin".

resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
parentpath	VARCHAR(4000)	The parent path of the resource.
dataservicename	VARCHAR(255)	The published data service name.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct ALL_RESOURCES: null
countname	INTEGER	The number of resources for a given resource name.
fromdate	DATE	The from date range.
todate	DATE	The to date range.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

ResourceCount_Total [RT] Table

ResourceCount_TotalRT – real time Real-time roll-up report of resource count by month. Derived from the following tables: vResourceUsage → metrics_resources_usage_hist {pm}

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate	TIMESTAMP	The timestamp of when the record was inserted or loaded.
month	VARCHAR(255)	A month string indicating the month of the request.
fromdate	DATE	The from date range.
todate	DATE	The to date range.
resourcecount	INTEGER	The total number of resources matching the criteria.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

SystemCPUandMemoryStatus Table

Report of system CPU and memory utilization and DV memory over time. Derived from the following tables: vSystemResources → METRICS_CIS_SYSTEM_RESOURCES{pm}

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.

eventtime	TIMESTAMP	The timestamp of when the row was recorded in the metrics tables.
avgmembytes	DECIMAL(19,0)	The average amount of memory in bytes currently being used.
avgmanagedmembytes	DECIMAL(19,0)	The average managed memory bytes currently in DV.
avgmemmax	DECIMAL(19,0)	The average maximum memory in bytes configured in DV.
avgmanagedmemmax	DECIMAL(19,0)	The average managed maximum memory in bytes configured in DV.
cpuutilization	DECIMAL(24,4)	The system CPU utilization (used) percent.
sysavailablememory	DECIMAL(24,4)	The system memory used in MB. This will always be showing high since DV takes the configured amount of memory at startup time. This is only a good guage of potential other processes running on the same server as DV if a large fluctuation is detected.
sysusedmemory	DECIMAL(24,4)	The system memory available in MB. Counterpart to the above statement.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vEventRequestSqlColumns Table

Report of the columns accessed by a SQL request query. Derived from the following tables: [METRICS_SQL_COLUMNS{pm}, METRICS_SQL_RESOURCE{pm}, RequestExpanded --> [metrics_requests_hist{pm}, metrics_sessions_hist{pm}]]

Column Name	Column Type	Definition
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
requestdatemonths	DECIMAL(19,2)	The number of months difference between the request date and the current date.
requestdatedays	DECIMAL(19,0)	The number of days difference between the request date and the current date.
resourceseqid	DECIMAL(19,0)	The resource sequence id is a foreign key back to vMetricsSqlResource.
resourceorder	DECIMAL(10,0)	The resource order.
columnposition	NUMERIC(10,2)	The ordinal position of the column and derivied colunm. 1.0=1st column, 1.1=1st column of a derived field 1.2=2nd colunm of dervied field.
columnorigtype	VARCHAR(20)	Column origination type determines the type of SQL the column came from. CALL, PRIMARY, COMPLEX, SUBQUERY
columnalias	VARCHAR(255)	The logical column (alias) name from the SQL statement. If not present then "columnName" is the alias.
columnname	VARCHAR(255)	The projected column name from the SQL statement. If * used, then same as actual column name
columnactual	VARCHAR(255)	The physical/source column name from the metadata table.
columntype	VARCHAR(4000)	The type of column.
columndirection	VARCHAR(10)	The direction of the column. For TABLE types it is always OUT. For PROCEDURE types it may be IN, OUT, INOUT, RESULT, RETURN.
columnderived	VARCHAR(10)	Determine whether this column is derived or not [true|false].
columnexpression	CLOB	The complete column expression.
columnfunctionlist	VARCHAR(4000)	A list of functions and occurrences that are used in the column expression. UPPER[3] would indicate that UPPER was used 3 times in the column expression.
columnnamelist	VARCHAR(4000)	A list of column names that make up the column expression
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vEventRequestSqlResources Table

Report of the resources accessed by a SQL request query. Derived from the following tables: [METRICS_SQL_REQUEST{pm}, METRICS_SQL_RESOURCE{pm}, RequestExpanded → [metrics_requests_hist{pm}, metrics_sessions_hist{pm}]]

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
endtime	TIMESTAMP	The timestamp of when the request completed. E.g. 2020-01-06 10:19:36.899
requestdatemonths	DECIMAL(19,2)	The number of months difference between the request date and the current date.
requestdatedays	DECIMAL(19,0)	The number of days difference between the request date and the current date.
totalduration	DECIMAL(19,0)	The total duration of the request measured in milli-seconds. Divide by 1000 to get minutes.
totaldurationdisplay	VARCHAR(25)	The total duration formatted for display using the format 0 00:00:00.000.
status	VARCHAR(25)	The status of the request [SUCCESS, FAILED].
message	VARCHAR(4000)	The exception message if a request was not successful.
kpistatus	VARCHAR(25)	The KPImetrics SQL request processing status [SUCCESS, FAIL]
kpimessage	VARCHAR(4000)	An exception message that could occur during the processing of the SQL or null if no exception.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
sessionid [fk]	DECIMAL(19,0)	A unique session id and foreign key to metrics_sessions_hist.
clienthost	VARCHAR(255)	Designates the client host name that connected to DV.
bytesfromclient	DECIMAL(19,0)	The total number of bytes received from the client host during this session.
bytestoclient	DECIMAL(19,0)	The total number of bytes sent to the client host during this session.
rowsAffected	DECIMAL(19,0)	The number of rows affected or returned. The value may be -1.
sessiontype	VARCHAR(40)	Designates the type session connection [JDBC, HTTP, TASK, etc.].
sqlinvocationtype	VARCHAR(20)	The SQL invocation type indicates the type of request [SELECT, CALL, INSERT, UPDATE, DELETE, PROCEDURE, WITH].
description	CLOB	The actual client request or internal DV request.
resourceseqid	DECIMAL(19,0)	The resource sequence id is a foreign key back to vMetricsSqlResource.
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in “resourceorigin”.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
frompath	VARCHAR(4000)	The from path is the database URL such as catalog.schema.table.
fromalias	VARCHAR(255)	The from path alias if applicable.
dataservicename	VARCHAR(255)	The published data service name.
catalogname	VARCHAR(255)	The name of the published catalog or null if not applicable.
schemaname	VARCHAR(255)	The name of the published schema or null if not applicable.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vResourceUsage Table

A raw report of all user defined resources. Derived from the following tables: metrics_resources_usage_hist {pm}

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
cluster	VARCHAR(255)	The name of the cluster or null if no cluster.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
parentid [fk]	DECIMAL(19,0)	A foreign key back to the requestid within this table. Contains -1 when no parent reference.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
sessionid [fk]	DECIMAL(19,0)	A unique session id and foreign key to metrics_sessions_hist.
requestdate	DATE	The date [YYYY-MM-DD] of the request.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
endtime	TIMESTAMP	The timestamp of when the request completed. E.g. 2020-01-06 10:19:36.899
requestdatemonths	DECIMAL(19,2)	The number of months difference between the request date and the current date.
requestdatedays	DECIMAL(19,0)	The number of days difference between the request date and the current date.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct ALL_RESOURCES: null
dataservicename	VARCHAR(255)	The published data service name.
datasourcepath	VARCHAR(255)	The DV path to the published datasource/dataservice [/services/databases/<name>, /services/webservices/<path|name>]
datasourcetype	VARCHAR(255)	The type of published datasource [VirtualRelational | VirtualWsdl | CompositeWebService].
resourcekind	VARCHAR(20)	The type of resource that executed [system, user defined].
resourceguid	VARCHAR(40)	The guid of the resource path. Eg. cff5fe78-c3c5-4a81-b134-d497cea5351e
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in “resourceorigin”.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
parentname	VARCHAR(255)	The parent name of the resource. A.k.a. schema name if from published database.
grandparentname	VARCHAR(255)	The grand-parent name of the resource. A.k.a. catalog name if from published database.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
parentpath	VARCHAR(4000)	The parent path of the resource.
userid	DECIMAL(19,0)	The DV userid from the system.ALL_USERS tables.
user	VARCHAR(255)	The user name who connected for this session.

domain	VARCHAR(255)	The domain of the user who connected.
group	VARCHAR(255)	The group of the user who connected.

vResourceUsagePublished Table

A raw report of published user defined resources accessed over time where the resourcekind='user defined' and resourcetype='LINK'. Derived from the following tables: vResourceUsage → metrics_resources_usage_hist {pm}

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
requestdate	DATE	The date [YYYY-MM-DD] of the request.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
parentid [fk]	DECIMAL(19,0)	A foreign key back to the requestid within this table. Contains -1 when no parent reference.
userid	DECIMAL(19,0)	The DV userid from the system.ALL_USERS tables.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in "resourceorigin".
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
parentpath	VARCHAR(4000)	The parent path of the resource.
dataservicename	VARCHAR(255)	The published data service name.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct ALL_RESOURCES: null
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

cache Definitions

vCache Table

Report of the KPImetrics cache_status table. Displays all records in the table. Derived from METRICS_SYS_CACHES{pm}.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
eventtime	TIMESTAMP	The timestamp of when the row was recorded in system.SYS_CACHE.
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in "resourceorigin".
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
parentpath	VARCHAR(4000)	The parent path of the resource.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
ownerid	DECIMAL(19,0)	The resource owner unique id.
owner	VARCHAR(60)	The resource owner name.
status	VARCHAR(24)	The status of the SYS_CACHE [NOT LOADED, DISABLED, UP]
Variant [k]	VARCHAR(255)	The variant provides values for a procedure cache.
lastrefreshend	TIMESTAMP	The timestamp the cache was last refreshed.
lastsuccessend	TIMESTAMP	The timestamp of the last successful refresh.
lastfailend	TIMESTAMP	The timestamp of the last refresh failure.
lastaccess	TIMESTAMP	The timestamp of the last cache access.
lastsuccessduration	DECIMAL(19,0)	The duration of the last cache refresh.
lastfailduration	DECIMAL(19,0)	The duration of the last cache refresh failure.
numsuccess	DECIMAL(19,0)	The number of cache refresh successes.
numfail	DECIMAL(19,0)	The number of cache refresh failures.
numaccess	DECIMAL(19,0)	The number of accesses.
storageused	DECIMAL(19,0)	The amount of storage used in bytes.
message	CLOB	The exception message if a request was not successful.
initialtime	TIMESTAMP	The timestamp of the initial time the cache was refreshed.
nexttime	TIMESTAMP	The timestamp of the next time the cache will be refreshed.
frequency	VARCHAR(255)	The frequency of cache refresh such as [Every day at time].
currentrefreshstart	TIMESTAMP	The timestamp of the current refresh start.
currentduration	DECIMAL(19,0)	The duration of the current refresh.
currentstorage	DECIMAL(19,0)	The amount of storage in bytes of the current refresh.
currentcause	VARCHAR(20)	The current cause/exception of the refresh.

vCacheActive Table

Report of the KPImetrics cache_status table. Displays all "**active**" records in the table no matter if the cache is up or down or has a configuration error. Derived from vCache →METRICS_SYS_CACHES{pm}.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
eventtime	TIMESTAMP	The timestamp of when the row was recorded in system.SYS_CACHE.
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in "resourceorigin".
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
parentpath	VARCHAR(4000)	The parent path of the resource.

resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
ownerid	DECIMAL(19,0)	The resource owner unique id.
owner	VARCHAR(255)	The resource owner name.
status	VARCHAR(24)	The status of the SYS_CACHE [NOT LOADED, DISABLED, UP]
Variant [k]	VARCHAR(255)	The variant provides values for a procedure cache.
lastrefreshend	TIMESTAMP	The timestamp the cache was last refreshed.
lastsuccessend	TIMESTAMP	The timestamp of the last successful refresh.
lastfailend	TIMESTAMP	The timestamp of the last refresh failure.
lastaccess	TIMESTAMP	The timestamp of the last cache access.
lastsuccessduration	DECIMAL(19,0)	The duration of the last cache refresh.
lastfailduration	DECIMAL(19,0)	The duration of the last cache refresh failure.
numsuccess	DECIMAL(19,0)	The number of cache refresh successes.
numfail	DECIMAL(19,0)	The number of cache refresh failures.
numaccess	DECIMAL(19,0)	The number of accesses.
storageused	DECIMAL(19,0)	The amount of storage used in bytes.
message	CLOB	The exception message if a request was not successful.
initialtime	TIMESTAMP	The timestamp of the initial time the cache was refreshed.
nexttime	TIMESTAMP	The timestamp of the next time the cache will be refreshed.
frequency	VARCHAR(255)	The frequency of cache refresh such as [Every day at time].
currentrefreshstart	TIMESTAMP	The timestamp of the current refresh start.
currentduration	DECIMAL(19,0)	The duration of the current refresh.
currentstorage	DECIMAL(19,0)	The amount of storage in bytes of the current refresh.
currentcause	VARCHAR(20)	The current cause/exception of the refresh.

vCacheDisabled Table

Report of the KPImetrics cache_status table. Displays all "**disabled**" records in the table. Derived from vCache →METRICS_SYS_CACHES{pm}.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
eventtime	TIMESTAMP	The timestamp of when the row was recorded in system.SYS_CACHE.
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in "resourceorigin".
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
parentpath	VARCHAR(4000)	The parent path of the resource.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
ownerid	DECIMAL(19,0)	The resource owner unique id.
owner	VARCHAR(255)	The resource owner name.
status	VARCHAR(24)	The status of the SYS_CACHE [NOT LOADED, DISABLED, UP]
Variant [k]	VARCHAR(255)	The variant provides values for a procedure cache.
lastrefreshend	TIMESTAMP	The timestamp the cache was last refreshed.
lastsuccessend	TIMESTAMP	The timestamp of the last successful refresh.
lastfailend	TIMESTAMP	The timestamp of the last refresh failure.
lastaccess	TIMESTAMP	The timestamp of the last cache access.

lastsuccessduration	DECIMAL(19,0)	The duration of the last cache refresh.
lastfailduration	DECIMAL(19,0)	The duration of the last cache refresh failure.
numsuccess	DECIMAL(19,0)	The number of cache refresh successes.
numfail	DECIMAL(19,0)	The number of cache refresh failures.
numaccess	DECIMAL(19,0)	The number of accesses.
storageused	DECIMAL(19,0)	The amount of storage used in bytes.
message	CLOB	The exception message if a request was not successful.
initialtime	TIMESTAMP	The timestamp of the initial time the cache was refreshed.
nexttime	TIMESTAMP	The timestamp of the next time the cache will be refreshed.
frequency	VARCHAR(255)	The frequency of cache refresh such as [Every day at time].
currentrefreshstart	TIMESTAMP	The timestamp of the current refresh start.
currentduration	DECIMAL(19,0)	The duration of the current refresh.
currentstorage	DECIMAL(19,0)	The amount of storage in bytes of the current refresh.
currentcause	VARCHAR(20)	The current cause/exception of the refresh.

vCacheIssues Table

Report of the KPImetrics cache_status table. Displays all records with "**issues**" in the table that have an error state such as DOWN, CONFIG ERROR and NOT LOADED. Derived from vCache →METRICS_SYS_CACHES{pm}.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
eventtime	TIMESTAMP	The timestamp of when the row was recorded in system.SYS_CACHE.
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in "resourceorigin".
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
parentpath	VARCHAR(4000)	The parent path of the resource.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
ownerid	DECIMAL(19,0)	The resource owner unique id.
owner	VARCHAR(255)	The resource owner name.
status	VARCHAR(24)	The status of the SYS_CACHE [NOT LOADED, DISABLED, UP]
Variant [k]	VARCHAR(255)	The variant provides values for a procedure cache.
lastrefreshend	TIMESTAMP	The timestamp the cache was last refreshed.
lastsuccessend	TIMESTAMP	The timestamp of the last successful refresh.
lastfailend	TIMESTAMP	The timestamp of the last refresh failure.
lastaccess	TIMESTAMP	The timestamp of the last cache access.
lastsuccessduration	DECIMAL(19,0)	The duration of the last cache refresh.
lastfailduration	DECIMAL(19,0)	The duration of the last cache refresh failure.
numsuccess	DECIMAL(19,0)	The number of cache refresh successes.
numfail	DECIMAL(19,0)	The number of cache refresh failures.
numaccess	DECIMAL(19,0)	The number of accesses.
storageused	DECIMAL(19,0)	The amount of storage used in bytes.
message	CLOB	The exception message if a request was not successful.
initialtime	TIMESTAMP	The timestamp of the initial time the cache was refreshed.

nexttime	TIMESTAMP	The timestamp of the next time the cache will be refreshed.
frequency	VARCHAR(255)	The frequency of cache refresh such as [Every day at time].
currentrefreshstart	TIMESTAMP	The timestamp of the current refresh start.
currentduration	DECIMAL(19,0)	The duration of the current refresh.
currentstorage	DECIMAL(19,0)	The amount of storage in bytes of the current refresh.
currentcause	VARCHAR(20)	The current cause/exception of the refresh.

vCacheSchedule Table

Report of the KPImetrics cache_status table. Displays all active records in the table with a cache schedule and potential cache schedule dependency. It is ordered by their next schedule refresh time and dependencies upon other cached resources. Derived from vCache →METRICS_SYS_CACHES{pm}. Executes getCacheScheduleDependenciesProc() to get dependent cache resources and used cache resources.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
frequency	VARCHAR(255)	The frequency of cache refresh such as [Every day at time].
nexttime	TIMESTAMP	The timestamp of the next time the cache will be refreshed.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in "resourceorigin".
ownerid	DECIMAL(19,0)	The resource owner unique id.
owner	VARCHAR(255)	The resource owner name.
status	VARCHAR(20)	The status of the SYS_CACHE [NOT LOADED, DISABLED, UP]
Variant [k]	VARCHAR(255)	The variant provides values for a procedure cache.
lastrefreshend	TIMESTAMP	The timestamp the cache was last refreshed.
lastsuccessend	TIMESTAMP	The timestamp of the last successful refresh.
lastfailend	TIMESTAMP	The timestamp of the last refresh failure.
lastaccess	TIMESTAMP	The timestamp of the last cache access.
lastsuccessduration	DECIMAL(19,0)	The duration of the last cache refresh.
lastfailduration	DECIMAL(19,0)	The duration of the last cache refresh failure.
numsuccess	DECIMAL(19,0)	The number of cache refresh successes.
numfail	DECIMAL(19,0)	The number of cache refresh failures.
numaccess	DECIMAL(19,0)	The number of accesses.
storageused	DECIMAL(19,0)	The amount of storage used in bytes.
message	CLOB	The exception message if a request was not successful.
initialtime	TIMESTAMP	The timestamp of the initial time the cache was refreshed.
currentrefreshstart	TIMESTAMP	The timestamp of the current refresh start.
currentduration	DECIMAL(19,0)	The duration of the current refresh.
currentstorage	DECIMAL(19,0)	The amount of storage in bytes of the current refresh.
currentcause	VARCHAR(20)	The current cause/exception of the refresh.

dependentcacheresources	VARCHAR(2147483647)	A list of dependent cache resources or null.
usedcacheresources	VARCHAR(2147483647)	A list of used cache resources or null.

configurations Definitions

pMetricsEventRegistrationList Procedure

Provides the ability to list the subscriptions for the metrics event registration programmatically.

Column Name	Column Type	Definition
debug	CHAR(1)	IN – Y=debug on. N=debug off.
requesterEmail	VARCHAR(255)	IN – the requestor's email to use to search for event subscriptions.
subscriberEmail	VARCHAR(255)	OUT - Subscriber email. The email alias where alerts will be sent.
groupName	VARCHAR(255)	OUT - The CIS group name to monitor.
environment	VARCHAR(255)	OUT - The CIS environment nickname to monitor.
eventType	VARCHAR(255)	OUT - The KPImetrics event type to monitor.
requesterEmail	VARCHAR(255)	OUT - Primary requester contact. The person requesting the subscription.
requesterFirstName	VARCHAR(255)	OUT - Primary requester contact first name.
requesterLastName	VARCHAR(255)	OUT - Primary requester contact last name.
excludeText	VARCHAR(4000)	OUT - Pipe separated list of text phrases that would appear in the SYS_REQUESTS SQL description field and signify exclusion of this event. If the text is found in the SQL description.

pMetricsEventRegistrationSubscribe Procedure

Provides the ability to subscribe to a metrics event registration programmatically.

Column Name	Column Type	Definition
debug	CHAR(1)	IN - Y=debug on, N=debug off
requesterEmail	VARCHAR(255)	IN - Primary requester contact. The person requesting the subscription.
subscriberEmail	VARCHAR(255)	IN - Subscriber email. The email alias where alerts will be sent.
groupName	VARCHAR(255)	IN - The CIS group name to monitor.
environment	VARCHAR(255)	IN - The CIS environment nickname to monitor.
eventType	VARCHAR(255)	IN - The KPImetrics event type to monitor.
excludeText	VARCHAR(4000)	IN - Pipe separated list of text phrases that would appear in the SYS_REQUESTS SQL description field and signify exclusion of this event. If the text is found in the SQL description.

pMetricsEventRegistrationUnsubscribe Procedure

Provides the ability to unsubscribe to a metrics event registration programmatically.

Column Name	Column Type	Definition
debug	CHAR(1)	IN - Y=debug on, N=debug off
requesterEmail	VARCHAR(255)	IN - Primary requester contact. The person requesting the subscription.
subscriberEmail	VARCHAR(255)	IN - Subscriber email. The email alias where alerts will be sent.
groupName	VARCHAR(255)	IN - The CIS group name to monitor.
environment	VARCHAR(255)	IN - The CIS environment nickname to monitor.

eventType	VARCHAR(255)	IN - The KPImetrics event type to monitor.

metadata Definitions

metadata_tables_row_distribution Table

Provides a metadata table row distribution for each table and partition.

Column Name	Column Type	Definition
table_name	BIGINT	The name of the table
partition_name	VARCHAR(255)	The name of the partition. Also known as the partition window.
boundary_definition	VARCHAR(255)	The date of the boundary high value.
range_type	TIMESTAMP	The range associated with the boundary: LEFT or RIGHT. If LEFT then rows are contained to the LEFT of the boundary definition.
partition_number	INTEGER	The partition number is associated with an interval integer representing the day of the year and is a value between 1 and 366. 366 days in a year accounts for leap year.
num_rows	INTEGER	The number of rows in the partition.

reportMetadataAllCount Table

Report on how many rows exist in each table. Count various types including the following:

- project – Count the rows in each table for each project found in METADATA_CONST_NAME and nodehost and nodeport
  - GROUP BY loaddate, projectnameid, projectname, nodehost, nodeport
- subtotal – Count the subtotal of rows for each nodehost and nodeport.
  - GROUP BY loaddate, nodehost, nodeport
- total – Count the total rows in each table.

When this view is invoked externally, the invoking report should sort by the following:
ORDER BY viewname, loaddate DESC, counttype, nodehost, nodeport, projectnameid

Column Name	Column Type	Definition
cnt	BIGINT	The record count for the counttype.
counttype	VARCHAR(8)	The type of count including [project, subtotal, total] project - Count the rows in each table for each project found in METADATA_CONST_NAME and nodehost and nodeport. GROUP BY loaddate, projectnameid, projectname, nodehost, nodeport subtotal - Count the subtotal of rows for each nodehost and nodeport. GROUP BY loaddate, nodehost, nodeport total - Count the total rows in each table.
viewname	VARCHAR(24)	The name of the view metadata view where the count was taken.
loaddate	TIMESTAMP	The timestamp date the when the table was loaded.
projectnameid	DECIMAL(19,0)	The project name id is a foreign key to the vMetadataConstName table.
projectname	VARCHAR(50)	A unique project name.
nodehost	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

reportMetadataDatasource Table

Report on what datasource a view is using. Lineage from the view to the datasource. This can be joined with vMetadataDatasource to get additional datasource information. It may also be joined with vMetadataResource to get additional resource information.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate [k]	TIMESTAMP	The timestamp date the when the table was loaded.
projectnameid [fk]	DECIMAL(19,0)	The project name id is a foreign key to the vMetadataConstName table.
projectname [k]	VARCHAR(50)	A unique project name.
environmentname	VARCHAR(255)	The environment name from commonValues.cisServerNickname.
layertype	VARCHAR(50)	A custom tag specifying the name of the layer. [Published_DS, Application_views, Business_Logical, Physical_Formatting, Physical_Metadata, etc.]
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in “resourceorigin”.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
dataservicename	VARCHAR(255)	The published data service name.
catalogname	VARCHAR(255)	The name of the published catalog or null if not applicable.
schemaname	VARCHAR(255)	The name of the published schema or null if not applicable.
datasourceid	DECIMAL(19,0)	The unique datasource id which comes from system.ALL_RESOURCES.
datasourceadaptertype	VARCHAR(50)	The adapter type of the datasource such as Oracle, PostgresSQL, SqlServer, Composite, Rest, Wsdl, File, Netezza, MsExcel(POI), Ldap, MySql, XmlFile, Jdbc, DB2, etc.
datasourcepath	VARCHAR(255)	The DV path to the published datasource/dataservice [/services/databases/<name>, /services/webservices/<path|name>]
annotation	CLOB	The datasource annotation.

reportMetadataNonCompliantColumns Table

Report of the non-compliant columns as configured by METADATA_CONST_VALIDATE.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate [k]	TIMESTAMP	The timestamp date the when the table was loaded.
projectnameid [fk]	DECIMAL(19,0)	The project name id is a foreign key to the vMetadataConstName table.
projectname [k]	VARCHAR(50)	A unique project name.
environmentname	VARCHAR(255)	The environment name from commonValues.cisServerNickname.
layertype	VARCHAR(50)	A custom tag specifying the name of the layer. [Published_DS, Application_views, Business_Logical, Physical_Formatting, Physical_Metadata, etc.]
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in “resourceorigin”.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
noncompliantreason	VARCHAR(50)	The non-compliant reason such as [MISSING_COLUMN]
noncompliantdesc	VARCHAR(4000)	The non-compliant description such as the column name when MISSING_COLUMN.

lastmodifieddate	TIMESTAMP	The last modified date (timestamp).
lastmodifiedusername	VARCHAR(50)	The last modified user name.
lastmodifieduserdomain	VARCHAR(50)	The last modified user domain.
lastmodifieduserid [fk]	INTEGER	The last modified user id which is a foreign key to system.ALL_USERS.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

reportMetadataNonCompliantLayers Table

Report of the non-compliant columns as configured by METADATA_CONST_VALIDATE.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate [k]	TIMESTAMP	The timestamp date the when the table was loaded.
projectnameid [fk]	DECIMAL(19,0)	The project name id is a foreign key to the vMetadataConstName table.
projectname [k]	VARCHAR(50)	A unique project name.
environmentname	VARCHAR(255)	The environment name from commonValues.cisServerNickname.
layertype	VARCHAR(50)	A custom tag specifying the name of the layer. [Published_DS, Application_views, Business_Logical, Physical_Formatting, Physical_Metadata, etc.]
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in “resourceorigin”.
lineageorder	INTEGER	The order from top to bottom in the layer hierarchy with respect to how the views traverse through the lineage.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
noncompliantreason	VARCHAR(50)	The non-compliant reason such as [NON_COMPLIANT_PATH].
noncompliantdesc	VARCHAR(4000)	The non-compliant description such as the resource path if NON_COMPLIANT_PATH FROM_LAYER_TYPE[Business_Business] DOES NOT INVOKE TO_LAYER_TYPE[Business_Business,Business_Logical] ACTUAL TO_LAYER_TYPE=[Physical_Formatting] TO_RESOURCE_ID=[971661] TO_RESOURCE_TYPE:PATH= [TABLE:/shared/labCommon/Physical/Formatting/ds_orders/tutorial/orderdetails]
fromlayertype	VARCHAR(50)	A custom tag specifying the name of the “from” layer. [Published_DS, Application_views, Business_Logical, Physical_Formatting, Physical_Metadata, etc.]
fromresourceid	DECIMAL(19,0)	The unique “from” resource id which comes from the DV system tables listed in “resourceorigin”.
fromresourcepath	VARCHAR(4000)	The “from” resource path.
fromresourcetype	VARCHAR(40)	The “from” resource type.
fromresourcesubtype	VARCHAR(40)	The DV resource subtype. The further qualifies the precise type of resource. E.g. NONE, SQL_TABLE (view), DATABASE_TABLE (datasource table), SQL_SCRIPT_PROCEDURE (DV procedure), OPERATION_PROCEDURE (web service procedure), plus many more.
tolayertype	VARCHAR(50)	A custom tag specifying the name of the “to” layer. [Published_DS, Application_views, Business_Logical, Physical_Formatting, Physical_Metadata, etc.]
toresourceid	DECIMAL(19,0)	The unique “to” resource id which comes from the DV system tables listed in “resourceorigin”.
toresourcepath	VARCHAR(4000)	The “to” resource path.

toresourcetype	VARCHAR(40)	The “to” resource type.
toresourcesubtype	VARCHAR(40)	The DV resource subtype. The further qualifies the precise type of resource. E.g. NONE, SQL_TABLE (view), DATABASE_TABLE (datasource table), SQL_SCRIPT_PROCEDURE (DV procedure), OPERATION_PROCEDURE (web service procedure), plus many more.
lastmodifieddate	TIMESTAMP	The last modified date (timestamp).
lastmodifiedusername	VARCHAR(50)	The last modified user name.
lastmodifieduserdomain	VARCHAR(50)	The last modified user domain.
lastmodifieduserid	INTEGER	The last modified user id which is a foreign key to system.ALL_USERS.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

reportMetadataPrivilegeUsers Table

Provides a listing of all group and user privileges for each resource for a given project path.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate [k]	TIMESTAMP	The timestamp date the when the table was loaded.
projectnameid [fk]	DECIMAL(19,0)	The project name id is a foreign key to the vMetadataConstName table.
layertype	VARCHAR(50)	A custom tag specifying the name of the layer. [Published_DS, Application_views, Business_Logical, Physical_Formatting, Physical_Metadata, etc.]
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in “resourceorigin”.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
nametype	VARCHAR(5)	The name type specified whether name is a USER or GROUP.
nameid	DECIMAL(19,0)	The id of the name or group.
name	VARCHAR(50)	The name of the user or group for the privilege.
domainname	VARCHAR(50)	The domain name specifies which domain the name belongs to.
actualprivileges	VARCHAR(60)	The actual privileges [READ|WRITE|EXECUTE|SELECT|UPDATE|INSERT|DELETE|GRANT]
combinedprivileges	VARCHAR(60)	The combined privileges [READ|WRITE|EXECUTE|SELECT|UPDATE|INSERT|DELETE|GRANT]
inheritedprivileges	VARCHAR(60)	The inherited privileges [READ|WRITE|EXECUTE|SELECT|UPDATE|INSERT|DELETE|GRANT]
userpk	DECIMAL(19,0)	The unique sequence id
userid	DECIMAL(19,0)	The user id from system.ALL_USERS
username	VARCHAR(255)	The name of the user.
userdomain	VARCHAR(255)	The domain of the user.
explicitrights	VARCHAR(255)	The studio rights that the user has.
effectiverights	VARCHAR(255)	The effective rights are merged from effective and inherited.
inheritedrights	VARCHAR(255)	The inherited rights that the user has based on membership in groups.
annotation	VARCHAR(1024)	The users annotation.
islocked	VARCHAR(10)	Indicates whether the user is locked or not,
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

reportNumResourcesByLayer Table

Report of the number of resources in each layer.

Column Name	Column Type	Definition
loaddate [k]	TIMESTAMP	The timestamp date the when the table was loaded.
projectnameid [fk]	DECIMAL(19,0)	The project name id is a foreign key to the vMetadataConstName table.
projectname [k]	VARCHAR(50)	A unique project name.
environmentname	VARCHAR(255)	The environment name from commonValues.cisServerNickname.
layertype	VARCHAR(50)	A custom tag specifying the name of the layer. [Published_DS, Application_views, Business_Logical, Physical_Formatting, Physical_Metadata, etc.]
catalogname	VARCHAR(255)	The name of the published catalog or null if not applicable.
schemaname	VARCHAR(255)	The name of the published schema or null if not applicable.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
resourceCount	BIGINT	The number of resources at the specified layer.

reportResourceColumns Table

Report of all the resources and their columns.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate [k]	TIMESTAMP	The timestamp date the when the table was loaded.
projectnameid [fk]	DECIMAL(19,0)	The project name id is a foreign key to the vMetadataConstName table.
projectname [k]	VARCHAR(50)	A unique project name.
environmentname	VARCHAR(255)	The environment name from commonValues.cisServerNickname.
layertype	VARCHAR(50)	A custom tag specifying the name of the layer. [Published_DS, Application_views, Business_Logical, Physical_Formatting, Physical_Metadata, etc.]
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in “resourceorigin”.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
resourcesubtype	VARCHAR(40)	The DV resource subtype. The further qualifies the precise type of resource. E.g. NONE, SQL_TABLE (view), DATABASE_TABLE (datasource table), SQL_SCRIPT_PROCEDURE (DV procedure), OPERATION_PROCEDURE (web service procedure), plus many more.
dataservicename	VARCHAR(255)	The published data service name.
catalogname	VARCHAR(255)	The name of the published catalog or null if not applicable.
schemaname	VARCHAR(255)	The name of the published schema or null if not applicable.
columnname	VARCHAR(255)	The projected column name from the SQL statement. If * used, then same as actual column name
columntype	VARCHAR(40)	The type of column.
columndirection	VARCHAR(10)	The direction of the column. For TABLE types it is always OUT. For PROCEDURE types it may be IN, OUT, INOUT, RESULT, RETURN.
columnpos	INTEGER	The ordinal position of the column within the resource.
annotation	CLOB	The column annotation.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').

nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

reportResourceDatasourceLineage Table

Report of resources and their datasource lineage associations. A resource may have 0 or many data sources associated with it.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate [k]	TIMESTAMP	The timestamp date the when the table was loaded.
projectnameid [fk]	DECIMAL(19,0)	The project name id is a foreign key to the vMetadataConstName table.
layertype	VARCHAR(50)	A custom tag specifying the name of the layer. [Published_DS, Application_views, Business_Logical, Physical_Formatting, Physical_Metadata, etc.]
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in “resourceorigin”.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
resourcesubtype	VARCHAR(40)	The DV resource subtype. The further qualifies the precise type of resource. E.g. NONE, SQL_TABLE (view), DATABASE_TABLE (datasource table), SQL_SCRIPT_PROCEDURE (DV procedure), OPERATION_PROCEDURE (web service procedure), plus many more.
resourcelinktype	VARCHAR(40)	The type of resource the published “LINK” is pointing to such as [TABLE, PROCEDURE].
dataservicename	VARCHAR(255)	The published data service name.
catalogname	VARCHAR(255)	The name of the published catalog or null if not applicable.
schemaname	VARCHAR(255)	The name of the published schema or null if not applicable.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
datasourceid	DECIMAL(19,0)	The unique datasource id which comes from system.ALL_RESOURCES.
datasourcepath	VARCHAR(255)	The DV path to the published datasource/dataservice [/services/databases/<name>, /services/webservices/<path|name>]
datasourcetype	VARCHAR(255)	The type of published datasource [VirtualRelational | VirtualWsdl | CompositeWebService].
datasourcesubtype	VARCHAR(40)	The datasource subtype which further qualifies what type of datasource it is. E.g. RELATIONAL_DATA_SOURCE, XML_FILE_DATA_SOURCE.
datasourceadaptertype	VARCHAR(50)	The adapter type of the datasource such as Oracle, PostgresSQL, SqlServer, Composite, Rest, Wsdl, File, Netezza, MsExcel(POI), Ldap, MySql, XmlFile, Jdbc, DB2, etc.
datasourcehostname	VARCHAR(255)	The connection host name or ip address.
datasourceport	INTEGER	The connection port.
datasourcedatabasename	VARCHAR(100)	The connection database name.
datasourcelogin	VARCHAR(255)	The connection login name.
datasourceurl	VARCHAR(500)	The connection URL. URL if relational.
datasourcepatternstr	VARCHAR(500)	The connection URL pattern string.
datasourceroot	VARCHAR(500)	The connection root. Root if file.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetadataAllPrivileges Table

This view provides the latest cache of the DV system ALL_PRIVILEGES. This view provides a performance enhancement to Cache_METADATA_TABLES which needs to access this

information more than once during its processing. It is only refreshed when Cache_METADATA_TABLES is executed via the trigger or manually.

Column Name	Column Type	Definition
loaddate	TIMESTAMP	The timestamp of when the record was inserted or loaded.
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in “resourceorigin”.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK. Also maybe TABLE or PRODEDURE.
nametype	VARCHAR(5)	The name type is either USER or GROUP and defines the type of privilege.
nameid	DECIMAL(19,0)	The name id which can be a user or group id.
name	VARCHAR(255)	The name is either a user or group name.
ownerdomain	VARCHAR(255)	The domain of the user or group name.
privilege	INTEGER	A bitmask representation of the privilege. NONE=0, READ=1, WRITE=2, EXECUTE=4, SELECT=8, UPDATE=16, INSERT=32, DELETE=64 and GRANT=128. The value can be 0-255 which represents all combinations of privileges.
actualprivilege	VARCHAR(255)	The actual privilege from the privilege bitmask represented in format of [READ WRITE EXECUTE SELECT UPDATE INSERT DELETE GRANT]
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetadataAllResources Table

This view provides the latest cache of the DV system ALL_RESOURCES, ALL_COLUMNS, ALL_PROCEDURES, ALL_WSDL_OPERATIONS, ALL_PARAMETERS and joined with ALL_USERS. This view provides a performance enhancement to Cache_METADATA_TABLES which needs to access this information more than once during its processing. It is only refreshed when Cache_METADATA_TABLES is executed via the trigger or manually.

Column Name	Column Type	Definition
loaddate	TIMESTAMP	The timestamp of when the record was inserted or loaded.
resourceorigin	VARCHAR(30)	The original DV system table in which the resource originated from. [ALL_TABLES| ALL_PROCEDURES|ALL_WSDL_OPERATIONS|ALL_RESOURCES| ALL_COLUMNS|ALL_PARAMETERS]
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in “resourceorigin”.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
parentpath	VARCHAR(4000)	The parent path of the resource.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK. Also maybe TABLE or PRODEDURE.
resourcesubtype	VARCHAR(40)	The DV resource subtype. The further qualifies the precise type of resource. E.g. NONE, SQL_TABLE (view), DATABASE_TABLE (datasource table), SQL_SCRIPT_PROCEDURE (DV procedure), OPERATION_PROCEDURE (web service procedure), plus many more.
resourcelinktype	VARCHAR(40)	The type of resource the published “LINK” is pointing to such as [TABLE, PROCEDURE].
ownerid	DECIMAL(19,0)	The resource owner unique id.
ownername	VARCHAR(60)	The resource owner name.

ownerdomain	VARCHAR(255)	The domain of the ownername
guid	VARCHAR(1024)	The guid of the resource path. Eg. cff5fe78-c3c5-4a81-b134-d497cea5351e
datasourcetype	VARCHAR(255)	The type of published datasource [VirtualRelational | VirtualWsdl | CompositeWebService].
dataservicename	VARCHAR(255)	The published data service name.
catalogname	VARCHAR(255)	The name of the published catalog or null if not applicable.
schemaname	VARCHAR(255)	The name of the published schema or null if not applicable.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct ALL_RESOURCES: null
columnid	DECIMAL(19,0)	The DV column unique resource id.
columnname	VARCHAR(255)	The name of the column.
columntype	VARCHAR(255)	The column type base such as varchar, numeric, decimal.
columnsize	DECIMAL(19,0)	The column type size such as 255 if varchar(255) or 19 if decimal(19,2).
columnscale	DECIMAL(19,0)	The column type scale such as 2 if decimal(19,2).
columnorder	DECIMAL(9,0)	The order in which the column appears in the resource.
columndirection	DECIMAL(9,0)	The column direction such [4=OUT [table or scalar procedure, 1=IN, 3=CURSOR OUT]
annotation	CLOB	The resource annotation.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetadataAllUsersGroups Table

This view provides the latest cache of the DV system ALL_DOMAINS joined with ALL_USERS to provide a representation of user, groups and domains. This view is joined with vMetadataPrivilegeUser to associate a privilege with a user., This view provides a performance enhancement to Cache_METADATA_TABLES which needs to access this information more than once during its processing. It is only refreshed when Cache_METADATA_TABLES is executed via the trigger or manually.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate	TIMESTAMP	The timestamp of when the record was inserted or loaded.
userpk	DECIMAL(19,0)	The unique sequence id
userid	DECIMAL(19,0)	The user id from system.ALL_USERS
username	VARCHAR(255)	The name of the user.
userdomain	VARCHAR(255)	The domain of the user.

explicitrights	VARCHAR(255)	The studio rights that the user has.
effectiverights	VARCHAR(255)	The effective rights are merged from effective and inherited.
inheritedrights	VARCHAR(255)	The inherited rights that the user has based on membership in groups.
annotation	VARCHAR(1024)	The users annotation.
islocked	VARCHAR(10)	Indicates whether the user is locked or not,
groupid	DECIMAL(19,0)	The group id of the group that the user belongs to.
groupname	VARCHAR(255)	The name of the group that the user belongs to.
groupdomain	VARCHAR(255)	The domain of the group tha the user belongs to.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetadataConstLayers Table

Provides a listing of all of the metadata layer types for each project path.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate [k]	TIMESTAMP	The timestamp date the when the table was loaded.
projectnameid [fk]	DECIMAL(19,0)	The project name id is a foreign key to the vMetadataConstName table.
projectname [k]	VARCHAR(50)	A unique project name.
projectpath	VARCHAR(4000)	The DV resource path for the project name.
layertype	VARCHAR(50)	A custom tag specifying the name of the layer. [Published_DS, Application_views, Business_Logical, Physical_Formatting, Physical_Metadata, etc.]
parentpath	VARCHAR(4000)	The parent path of the resource.
generatelineage	CHAR(1)	Y=Generate lineage for this layer path. N=Do not generate lineage for this layer path.
exclusionlist	CLOB	A comma-separated list of paths or partial paths ending in a / that are to be excluded from the lineage generation. If a comma exists within a path then escape the comma with "_002C". e.g. /shared/my,path1/path2/ --> /shared/my_002Cpath1/path2/
assignprivileges	VARCHAR(25)	ASSIGN_PRVILEGES: Provides the rules for assigning privileges on a per layer basis. • NO_PRIVILEGES - Do not assign any privileges for this layer • ACTUAL_NO_USERS - Assign actual privileges but do not invoke the getResourcePrivileges() api to get COMBINED or INHERITED. Do not retrieve users associated with groups. • ACTUAL_WITH_USERS - [DEFAULT] Assign actual privileges but do not invoke the getResourcePrivileges() api to get COMBINED or INHERITED. Retrieve all users associated with a GROUP privilege. • COMBINED_NO_USERS - Invoke the getResourcePrivileges() api to get COMBINED and INHERITED privileges. Do not retrieve users associated with groups. Invoking the api will slow down the processing considerably. • COMBINED_WITH_USERS - Invoke the getResourcePrivileges() api to get COMBINED and INHERITED privileges. Retrieve all users associated with a GROUP privilege. Invoking the api will slow down the processing considerably.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetadataConstName Table

Provides a listing of all configured metadata constant name projects. This is the main driver table. It contains the project name, project name id, environment name, execute flag and archive flag.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate [k]	TIMESTAMP	The timestamp date the when the table was loaded.
projectnameid [k]	DECIMAL(19,0)	A unique project name id.
projectname [k]	VARCHAR(50)	A unique project name.
environmentname	VARCHAR(255)	The environment name from commonValues.cisServerNickname.
executeflag	CHAR(1)	The execute flag indicates if this project gets executed. Y=execute this row. N=do not execute when triggered.
projectdesc	VARCHAR(255)	A description of the project path.
executestatus	VARCHAR(4000)	The execution status. [IN_PROGRESS, SUCCESS, exception message].
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetadataConstPaths Table

Provides a listing of all configured metadata constant name projects. This table controls the paths that are connected with the vMetadataConstName view.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate [k]	TIMESTAMP	The timestamp date the when the table was loaded.
projectnameid [fk]	DECIMAL(19,0)	The project name id is a foreign key to the vMetadataConstName table.
projectname [k]	VARCHAR(50)	A unique project name.
projectpath	VARCHAR(4000)	The DV resource path for the project name.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetadataConstValidate Table

Provides a listing of all of the metadata layer validations for each project path.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate [k]	TIMESTAMP	The timestamp date the when the table was loaded.
projectnameid [fk]	DECIMAL(19,0)	The project name id is a foreign key to the vMetadataConstName table.
projectname [k]	VARCHAR(50)	A unique project name.
projectpath	VARCHAR(4000)	The DV resource path for the project name.
layertype	VARCHAR(50)	A custom tag specifying the name of the layer. [Published_DS, Application_views, Business_Logical, Physical_Formatting, Physical_Metadata, etc.]
ruletype	VARCHAR(50)	Valid values=[ENFORCE_LAYER|ENFORCE_COLUMN]
ruledesc	VARCHAR(4000)	Enforce the rule type. When RULE_TYPE=ENFORCE_COLUMN

		• Enforces which columns must be present in all of the views for a given layer type. Comma-separated list of case-sensative column names. When RULE_TYPE=ENFORCE_LAYER • Enforces which source layer resource can invoke which target layer resource. Comma-separated list of valid LAYER_TYPES. • If a resource can invoke another resource in the same layer then add its own layer to the list.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetadataDatasource Table

Provides a complete reference on the datasources that exist within the project paths referenced by vMetadataConstName.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate [k]	TIMESTAMP	The timestamp date the when the table was loaded.
projectnameid [fk]	DECIMAL(19,0)	The project name id is a foreign key to the vMetadataConstName table.
datasourceid	DECIMAL(19,0)	The unique datasource id which comes from system.ALL_RESOURCES.
datasourcepath	VARCHAR(255)	The DV path to the published datasource/dataservice [/services/databases/<name>, /services/webservices/<path|name>]
datasourcetype	VARCHAR(255)	The type of published datasource [VirtualRelational | VirtualWsdl | CompositeWebService].
datasourcesubtype	VARCHAR(40)	The datasource subtype which further qualifies what type of datasource it is. E.g. RELATIONAL_DATA_SOURCE, XML_FILE_DATA_SOURCE.
datasourceadaptertype	VARCHAR(50)	The adapter type of the datasource such as Oracle, PostgresSQL, SqlServer, Composite, Rest, Wsdl, File, Netezza, MsExcel(POI), Ldap, MySql, XmlFile, Jdbc, DB2, etc.
datasourcehostname	VARCHAR(255)	The connection host name or ip address.
datasourceport	INTEGER	The connection port.
datasourcedatabasename	VARCHAR(100)	The connection database name.
datasourcelogin	VARCHAR(255)	The connection login name.
datasourceurl	VARCHAR(500)	The connection URL. URL if relational.
datasourcepatternstr	VARCHAR(500)	The connection URL pattern string.
datasourceroot	VARCHAR(500)	The connection root. Root if file.
ownername	VARCHAR(50)	The resource owner name.
ownerdomain	VARCHAR(50)	The resource owner domain.
ownerid	INTEGER	The resource owner unique id.
creationdate	TIMESTAMP	The resource creation date (timestamp).
creatorusername	VARCHAR(50)	The resource creation user name.
creatoruserdomain	VARCHAR(50)	The resource creation user domain.
creatoruserid	INTEGER	The resource creation user id which is a foreign key to system.ALL_USERS.
lastmodifieddate	TIMESTAMP	The last modified date (timestamp).
lastmodifiedusername	VARCHAR(50)	The last modified user name.
lastmodifieduserdomain	VARCHAR(50)	The last modified user domain.
lastmodifieduserid	INTEGER	The last modified user id which is a foreign key to system.ALL_USERS.
annotation	CLOB	The datasource annotation.

nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetadataNonCompliant Table

Provides information on all non-compliant resources for all layers.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate [k]	TIMESTAMP	The timestamp date the when the table was loaded.
projectnameid [fk]	DECIMAL(19,0)	The project name id is a foreign key to the vMetadataConstName table.
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in "resourceorigin".
lineageorder	INTEGER	The order from top to bottom in the layer hierarchy with respect to how the views traverse through the lineage.
noncompliantreason	VARCHAR(50)	The non-compliant reason such as [MISSING_COLUMN, NON_COMPLIANT_PATH].
noncompliantdesc	VARCHAR(4000)	The non-compliant description such as the column name when MISSING_COLUMN or the resource path if NON_COMPLIANT_PATH FROM_LAYER_TYPE[Business_Business] DOES NOT INVOKE TO_LAYER_TYPE[Business_Business,Business_Logical] ACTUAL TO_LAYER_TYPE=[Physical_Formatting] TO_RESOURCE_ID=[971661] TO_RESOURCE_TYPE:PATH= [TABLE:/shared/labCommon/Physical/Formatting/ds_orders/tutorial/orderdetails]
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetadataPolicy Table

Provides information on all policies for RBS [rule-based security] and CBS [column-based security] rows for a given project path.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate [k]	TIMESTAMP	The timestamp date the when the table was loaded.
projectnameid [fk]	DECIMAL(19,0)	The project name id is a foreign key to the vMetadataConstName table.
policyid [k]	DECIMAL(19,0)	The unique policy id.
policytype	VARCHAR(10)	The type of policy may be RBS (row-based security) or CBS (column-based security).
resourcepath [k]	VARCHAR(4000)	The DV path to the resource policy.
resourcetype [k]	VARCHAR(40)	The DV resource type. [PROCEDURE]
resourcesubtype	VARCHAR(40)	The policy subtype. [cbs_policy, rbs_policy]
enabled	VARCHAR(5)	The policy is enabled (true) or not enabled (false).
form	VARCHAR(50)	Refers to the form if created with "free-form".
defaultrulefilter	VARCHAR(50)	The default rule may be [No Rows, All Rows, Predicate, or Procedure].
defaultruledata	VARCHAR(1000)	The default rule data may refer to the data or a path.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetadataPolicyAssignmnt Table

Provides the policy assignments for each policy.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate [k]	TIMESTAMP	The timestamp date the when the table was loaded.
projectnameid [fk]	DECIMAL(19,0)	The project name id is a foreign key to the vMetadataConstName table.
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in “resourceorigin”.
policyid	DECIMAL(19,0)	The unique policy id.
policytype	VARCHAR(10)	The type of policy may be RBS (row-based security) or CBS (column-based security).
columnname	VARCHAR(255)	The name of the column associated with the policy.
message	VARCHAR(4000)	Provides a way of logging a message. For example, if the policy assignment path no longer exists, then a message would be logged here.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty(‘SERVER_HOSTNAME’).
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty(‘SERVER_JDBC_PORT’)-1.

vMetadataPrivilege Table

Provides a listing of all privileges for each resource for a given project path. The privilege may be for a nametype=[GROUP|USER]. The privilege

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate [k]	TIMESTAMP	The timestamp date the when the table was loaded.
Projectnameid [fk]	DECIMAL(19,0)	The project name id is a foreign key to the vMetadataConstName table.
privilegeid [k]	DECIMAL(19,0)	The privilege id generated sequence.
layertype	VARCHAR(50)	A custom tag specifying the name of the layer. [Published_DS, Application_views, Business_Logical, Physical_Formatting, Physical_Metadata, etc.]
resourceid [k]	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in “resourceorigin”.
resourcepath	VARCHAR(4000)	The DV path to the resource.
resourcetype	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
nametype [k]	VARCHAR(5)	The name type specified whether name is a USER or GROUP.
nameid	DECIMAL(19,0)	The id of the name or group.
name [k]	VARCHAR(50)	The name of the user or group for the privilege.
domainname [k]	VARCHAR(50)	The domain name specifies which domain the name belongs to.
actualprivileges	VARCHAR(60)	The actual privileges [READ|WRITE|EXECUTE|SELECT|UPDATE|INSERT|DELETE|GRANT]
combinedprivileges	VARCHAR(60)	The combined privileges [READ|WRITE|EXECUTE|SELECT|UPDATE|INSERT|DELETE|GRANT]
inheritedprivileges	VARCHAR(60)	The inherited privileges [READ|WRITE|EXECUTE|SELECT|UPDATE|INSERT|DELETE|GRANT]
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetadataPrivilegeUser Table

This view provides a way to associate privileges with users. It is the many-to-many table that gets joined with vMetadataPrivilege and vMetadataAllUsersGroups.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate [k]	TIMESTAMP	The timestamp of when the record was inserted or loaded.
projectnameid [fk]	DECIMAL(19,0)	The project name id is a foreign key to the vMetadataConstName table.
userpk [k]	DECIMAL(19,0)	The user sequence primary key from vMetadataAllUsersGroups.
privilegeid [k]	DECIMAL(19,0)	The privilege id from vMetadataPrivilege
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetadataResource Table

Provides a listing of all of the metadata resources for each project path. This is the core table that provides information about a resource.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate [k]	TIMESTAMP	The timestamp date the when the table was loaded.
projectnameid [fk]	DECIMAL(19,0)	The project name id is a foreign key to the vMetadataConstName table.
layertype	VARCHAR(50)	A custom tag specifying the name of the layer. [Published_DS, Application_views, Business_Logical, Physical_Formatting, Physical_Metadata, etc.]
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in “resourceorigin”.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
resourcesubtype	VARCHAR(40)	The DV resource subtype. The further qualifies the precise type of resource. E.g. NONE, SQL_TABLE (view), DATABASE_TABLE (datasource table), SQL_SCRIPT_PROCEDURE (DV procedure), OPERATION_PROCEDURE (web service procedure), plus many more.
resourcelinktype	VARCHAR(40)	The type of resource the published “LINK” is pointing to such as [TABLE, PROCEDURE].
dataservicename	VARCHAR(255)	The published data service name.
catalogname	VARCHAR(255)	The name of the published catalog or null if not applicable.
schemaname	VARCHAR(255)	The name of the published schema or null if not applicable.
compliant	SMALLINT	The resource is compliant when it meets the layer and column validations. 1=compliant. 0=non-compliant.
ownername	VARCHAR(50)	The resource owner name.
ownerdomain	VARCHAR(50)	The resource owner domain.
ownerid	INTEGER	The resource owner unique id.
creationdate	TIMESTAMP	The resource creation date (timestamp).
creatorusername	VARCHAR(50)	The resource creation user name.
creatoruserdomain	VARCHAR(50)	The resource creation user domain.
creatoruserid	INTEGER	The resource creation user id which is a foreign key to system.ALL_USERS.

lastmodifieddate	TIMESTAMP	The last modified date (timestamp).
lastmodifiedusername	VARCHAR(50)	The last modified user name.
lastmodifieduserdomain	VARCHAR(50)	The last modified user domain.
lastmodifieduserid	INTEGER	The last modified user id which is a foreign key to system.ALL_USERS.
annotation	CLOB	The resource annotation.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetadataResourceColumn Table

Provides a listing of all of the metadata resource columns for each resource and project path. This is the core table that provides information about columns and their position.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate [k]	TIMESTAMP	The timestamp date the when the table was loaded.
projectnameid [fk]	DECIMAL(19,0)	The project name id is a foreign key to the vMetadataConstName table.
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in “resourceorigin”.
columnname	VARCHAR(255)	The name of the column.
columntype	VARCHAR(40)	The type of column.
columndirection	VARCHAR(10)	The direction of the column. For TABLE types it is always OUT. For PROCEDURE types it may be IN, OUT, INOUT, RESULT, RETURN.
columnpos	INTEGER	The ordinal position of the column within the resource.
annotation	CLOB	The column annotation.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetadataResourceLineage Table

Provides a listing of the lineage for each resource at each layer. Lineage will be different based on which layer is referenced. Each record also references the datasource that is used.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate [k]	TIMESTAMP	The timestamp date the when the table was loaded.
projectnameid [fk]	DECIMAL(19,0)	The project name id is a foreign key to the vMetadataConstName table.
layertype	VARCHAR(50)	A custom tag specifying the name of the layer. [Published_DS, Application_views, Business_Logical, Physical_Formatting, Physical_Metadata, etc.]
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in “resourceorigin”.
lineageorder	INTEGER	The order from top to bottom in the layer hierarchy with respect to how the views traverse through the lineage.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
resourcesubtype	VARCHAR(40)	The DV resource subtype. The further qualifies the precise type of resource. E.g. NONE, SQL_TABLE (view), DATABASE_TABLE (datasource table),

		SQL_SCRIPT_PROCEDURE (DV procedure), OPERATION_PROCEDURE (web service procedure), plus many more.
resourcelinktype	VARCHAR(40)	The type of resource the published "LINK" is pointing to such as [TABLE, PROCEDURE].
datasourceid	DECIMAL(19,0)	The unique datasource id which comes from system.ALL_RESOURCES.
fromlayertype	VARCHAR(50)	A custom tag specifying the name of the "from" layer. [Published_DS, Application_views, Business_Logical, Physical_Formatting, Physical_Metadata, etc.]
fromresourceid	DECIMAL(19,0)	The unique "from" resource id which comes from the DV system tables listed in "resourceorigin".
fromresourcepath	VARCHAR(4000)	The "from" resource path.
fromresourcetype	VARCHAR(40)	The "from" resource type.
fromresourcesubtype	VARCHAR(40)	The DV resource subtype. The further qualifies the precise type of resource. E.g. NONE, SQL_TABLE (view), DATABASE_TABLE (datasource table), SQL_SCRIPT_PROCEDURE (DV procedure), OPERATION_PROCEDURE (web service procedure), plus many more.
tolayertype	VARCHAR(50)	A custom tag specifying the name of the "to" layer. [Published_DS, Application_views, Business_Logical, Physical_Formatting, Physical_Metadata, etc.]
toresourceid	DECIMAL(19,0)	The unique "to" resource id which comes from the DV system tables listed in "resourceorigin".
toresourcepath	VARCHAR(4000)	The "to" resource path.
toresourcetype	VARCHAR(40)	The "to" resource type.
toresourcesubtype	VARCHAR(40)	The DV resource subtype. The further qualifies the precise type of resource. E.g. NONE, SQL_TABLE (view), DATABASE_TABLE (datasource table), SQL_SCRIPT_PROCEDURE (DV procedure), OPERATION_PROCEDURE (web service procedure), plus many more.
message	VARCHAR(4000)	The exception message if a request was not successful.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

requests Definitions

vEventRegLog Table

Derived from METRICS_EVENT_REG_LOG. Details about the events that have occurred and the emails that have been sent out. An event is only logged if an email is sent. Events are registered in the METRICS_EVENT_REGISRATION table and include: LONG_RUNNING, EXCEEDED_MEMORY, INACTIVITY, WORKFLOW_FAILURE, DBMS_SCHEDULER_ERROR, and PURGE_HISTORY.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
eventlogid [k]	DECIMAL(19,0)	A generated unique sequence id for this nodehost and nodeport.
eventtime	TIMESTAMP	The timestamp of when the the event occurred in DV and was recorded in METRICS_EVENT_REG_LOG.
eventtype	VARCHAR(255)	The event type describes what type of event occurred: LONG_RUNNING – A long running process was triggered but only if DV is configured for 'Request Run Time'. EXCEEDED_MEMORY – A running process has exceeded the allowed memory setting as configured by the administrator. 'Available Managed Memory' and 'Maximum Memory per Request'

		INACTIVITY – No metrics table inserts have occurred consistently over a period of time. WORKFLOW_FAILURE – A KPImetrics triggered process has thrown an exception. DBMS_SCHEDULER_ERROR – A KPImetrics database-oriented process has thrown an exception. PURGE_HISTORY – A KPImetrics purge history process has thrown an exception.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
sessionid [fk]	DECIMAL(19,0)	A unique session id and foreign key to metrics_sessions_hist.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
datasourceid	DECIMAL(19,0)	The unique datasource id which comes from system.ALL_RESOURCES.
datasourcename	VARCHAR(255)	The datasource name.
totalduration	DECIMAL(19,0)	The total duration of the request measured in milli-seconds. Divide by 1000 to get minutes.
totaldurationformatted	VARCHAR(20)	A formatted total duration for display purposes.
maxmemory	DECIMAL(19,0)	The maximum memory in bytes.
maxusedmemory	DECIMAL(19,0)	The maximum memory used by the request in bytes.
currentmemory	DECIMAL(19,0)	The current memory in bytes.
currentusedmemory	DECIMAL(19,0)	The current memory used by the request in bytes.
requesttype	VARCHAR(255)	The type of request that is stored in the description field. “SQL”=select statement, “SQL Script”=call statement executed, “XSLT, Basic, or Streaming Transformation”, “Java Procedure (built-in)”, etc.
description	CLOB	The actual client request or internal DV request.
ownerid	DECIMAL(19,0)	The resource owner unique id.
owner	VARCHAR(255)	The resource owner name.
emailsendto	VARCHAR(1000)	Derived from the email subscription or default configured sendto.
emailsubject	VARCHAR(1000)	Concatenated value of server environment, hostname and port.
emailmessage	CLOB	The email message that was sent to the event requestor.
notificationstatus	VARCHAR(255)	Indicates that a notification was emailed. NOTIFY_SUCCESS, NOTIFY_ERROR, NOTIFY_REQUIRED.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vEventRegLogLineage Table

Derived from METRICS_EVENT_REG_LOG_LINEAGE. Details about the data source lineage for a SQL description (request) that results from an event registered by METRICS_EVENT_REGISTRATION. Events that log data source lineage include: LONG_RUNNING and EXCEEDED_MEMORY because they have an associated SQL description that gets parsed in order to determine the lineage.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
eventlineageid [k]	DECIMAL(19,0)	A generated unique sequence id for this nodehost and nodeport.
eventlogid [fk]	DECIMAL(19,0)	A foreign key to the vEentRegLog view.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
datasourcepath	VARCHAR(255)	The DV path to the published datasource/dataservice [/services/databases/<name>, /services/webservices/<path|name>]

datasourcetype	VARCHAR(255)	The type of published datasource [VirtualRelational | VirtualWsdl | CompositeWebService].
datasourcesubtype	VARCHAR(255)	The datasource subtype which further qualifies what type of datasource it is. E.g. RELATIONAL_DATA_SOURCE, XML_FILE_DATA_SOURCE.
connhostname	VARCHAR(4000)	The connection host name or ip address.
connport	INTEGER	The connection port.
connloginname	VARCHAR(255)	The connection login name.
conndatabasename	VARCHAR(255)	The connection database name.
conndatabasetype	VARCHAR(255)	The connection database type such as Oracle, PostgresSQL, SqlServer, Composite, Rest, Wsdl, File, Netezza, MsExcel(POI), Ldap, MySql, XmlFile, Jdbc, DB2, etc.
connurlorroot	VARCHAR(4000)	The connection URL or root. URL if relational. Root if file.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vEventRequestSqlColumns Table

Details about the column projection list from the metrics_request description (SQL). These columns were parsed. Derived from the following tables: METRICS_SQL_REQUEST, METRICS_SQL_RESOURCE, METRICS_SQL_COLUMNS, metrics_requests_hist and metrics_sessions_hist

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
requestdatemonths	DECIMAL(19,2)	The number of months difference between the request date and the current date.
requestdatedays	DECIMAL(19,0)	The number of days difference between the request date and the current date.
status	VARCHAR(20)	The status of the request [SUCCESS, FAILED].
sqlmessage	VARCHAR(4000)	Provides a general informational message originating from vMetricsSqlRequest.
message	CLOB	The exception message if a request was not successful.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
clienthost	VARCHAR(255)	Designates the client host name that connected to DV.
type	VARCHAR(20)	The type of request [begin, end, null].
bytesfromclient	DECIMAL(19,0)	The total number of bytes received from the client host during this session.
bytestoclient	DECIMAL(19,0)	The total number of bytes sent to the client host during this session.
rowsAffected	DECIMAL(19,0)	The number of rows affected or returned. The value may be -1.
sessiontype	VARCHAR(40)	Designates the type session connection [JDBC, HTTP, TASK, etc.].
sqlinvocationtype	VARCHAR(20)	The SQL invocation type indicates the type of request [SELECT, CALL, INSERT, UPDATE, DELETE, PROCEDURE, WITH].
kpirequestidmatch	DECIMAL(19,0)	The requestid of a match to a similar request templated so that the SQL template can be copied instead of processed from the start.
kpidescriptiontemplate	VARCHAR(4000)	The template of the actual client request or internal DV request. All literal values have been removed in order to normalize the SQL so it can be used to compare with other requests on a generic basis. The contents are 4000 characters or less which means that queries can be pushed down to the database.

kpidescriptiontemplateclob	CLOB	The template of the actual client request or internal DV request. All literal values have been removed in order to normalize the SQL so it can be used to compare with other requests on a generic basis. The contents are greater than 4000 characters.
description	CLOB	The actual client request or internal DV request.
resourceseqid	DECIMAL(19,0)	The resource sequence id is a foreign key back to vMetricsSqlResource.
resourceorder	DECIMAL(10,0)	The resource order.
columnposition	NUMERIC(10,2)	The ordinal position of the column and derivied colunm. 1.0=1st column, 1.1=1st column of a derived field 1.2=2nd colunm of dervied field.
columnorigtype	VARCHAR(20)	Column origination type determines the type of SQL the column came from. CALL, PRIMARY, COMPLEX, SUBQUERY
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in "resourceorigin".
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
frompath	VARCHAR(4000)	The from path is the database URL such as catalog.schema.table.
fromalias	VARCHAR(255)	The from path alias if applicable.
dataservicename	VARCHAR(255)	The published data service name.
catalogname	VARCHAR(255)	The name of the published catalog or null if not applicable.
schemaname	VARCHAR(255)	The name of the published schema or null if not applicable.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
columnalias	VARCHAR(255)	The logical column (alias) name from the SQL statement. If not present then "columnName" is the alias.
columnname	VARCHAR(255)	The projected column name from the SQL statement. If * used, then same as actual column name
columnactual	VARCHAR(255)	The physical/source column name from the metadata table.
columntype	VARCHAR(4000)	The type of column.
columndirection	VARCHAR(10)	The direction of the column. For TABLE types it is always OUT. For PROCEDURE types it may be IN, OUT, INOUT, RESULT, RETURN.
columnderived	VARCHAR(10)	Determine whether this column is derived or not [true|false].
columnexpression	CLOB	The complete column expression.
columnfunctionlist	VARCHAR(4000)	A list of functions and occurrences that are used in the column expression. UPPER[3] would indicate that UPPER was used 3 times in the column expression.
columnnamelist	VARCHAR(4000)	A list of column names that make up the column expression
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vEventRequestSqlResources Table

Details about the resources used from the metrics_request description (SQL). These resources were parsed. Derived from the following tables: METRICS_SQL_REQUEST, METRICS_SQL_RESOURCE, metrics_requests_hist and metrics_sessions_hist

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
endtime	TIMESTAMP	The timestamp of when the request completed. E.g. 2020-01-06 10:19:36.899

requestdatemonths	DECIMAL(19,2)	The number of months difference between the request date and the current date.
requestdatedays	DECIMAL(19,0)	The number of days difference between the request date and the current date.
totalduration	DECIMAL(19,0)	The total duration of the request measured in milli-seconds. Divide by 1000 to get minutes.
totaldurationdisplay	VARCHAR(25)	The total duration formatted for display using the format 0 00:00:00.000.
status	VARCHAR(20)	The status of the request [SUCCESS, FAILED].
message	VARCHAR(4000)	The exception message if a request was not successful.
kpistatus	VARCHAR(25)	The KPImetrics SQL request processing status [SUCCESS, FAIL]
kpiprocesstime	TIMESTAMP	The time the SQL request processing started.
kpidurationseconds	DECIMAL(19,3)	The number of seconds and fraction of seconds [5.102] the processing took.
kpidurationinterval	VARCHAR(25)	A formatted duration interval of how long the processing took [0 00:00:02].
kpimessage	VARCHAR(4000)	An exception message that could occur during the processing of the SQL or null if no exception.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
sessionid [fk]	DECIMAL(19,0)	A unique session id and foreign key to metrics_sessions_hist.
clienthost	VARCHAR(255)	Designates the client host name that connected to DV.
type	VARCHAR(20)	Indicates whether the session is of type ‘begin’ or ‘end’ or null.
bytesfromclient	DECIMAL(19,0)	The total number of bytes received from the client host during this session.
bytestoclient	DECIMAL(19,0)	The total number of bytes sent to the client host during this session.
rowsAffected	DECIMAL(19,0)	The number of rows affected or returned. The value may be -1.
sessiontype	VARCHAR(40)	Designates the type session connection [JDBC, HTTP, TASK, etc.].
sqlinvocationtype	VARCHAR(20)	The SQL invocation type indicates the type of request [SELECT, CALL, INSERT, UPDATE, DELETE, PROCEDURE, WITH].
kpirequestidmatch	DECIMAL(19,0)	The requestid of a match to a similar request templated so that the SQL template can be copied instead of processed from the start.
kpidescriptiontemplate	VARCHAR(4000)	The template of the actual client request or internal DV request. All literal values have been removed in order to normalize the SQL so it can be used to compare with other requests on a generic basis. The contents are 4000 characters or less which means that queries can be pushed down to the database.
kpidescriptiontemplateclob	CLOB	The template of the actual client request or internal DV request. All literal values have been removed in order to normalize the SQL so it can be used to compare with other requests on a generic basis. The contents are greater than 4000 characters.
description	CLOB	The actual client request or internal DV request.
resourceseqid	DECIMAL(19,0)	The resource sequence id is a foreign key back to vMetricsSqlResource.
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in “resourceorigin”.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
frompath	VARCHAR(4000)	The from path is the database URL such as catalog.schema.table.
fromalias	VARCHAR(255)	The from path alias if applicable.
dataservicename	VARCHAR(255)	The published data service name.
catalogname	VARCHAR(255)	The name of the published catalog or null if not applicable.
schemaname	VARCHAR(255)	The name of the published schema or null if not applicable.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vEventRequestSqlResourcesAllErrors Table

Details about errors produced during parsing of the metrics_request description (SQL). This can be used to improve the SQL Parser code implemented by KPImetrics. These resources were parsed. Derived from the following tables: METRICS_SQL_REQUEST, METRICS_SQL_RESOURCE, metrics_requests_hist and metrics_sessions_hist

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
endtime	TIMESTAMP	The timestamp of when the request completed. E.g. 2020-01-06 10:19:36.899
requestdatemonths	DECIMAL(19,2)	The number of months difference between the request date and the current date.
requestdatedays	DECIMAL(19,0)	The number of days difference between the request date and the current date.
totalduration	DECIMAL(19,0)	The total duration of the request measured in milli-seconds. Divide by 1000 to get minutes.
totaldurationdisplay	VARCHAR(25)	The total duration formatted for display using the format 0 00:00:00.000.
status	VARCHAR(20)	The status of the request [SUCCESS, FAILED].
message	VARCHAR(4000)	The exception message if a request was not successful.
kpistatus	VARCHAR(25)	The KPImetrics SQL request processing status [SUCCESS, FAIL]
kpiprocesstime	TIMESTAMP	The time the SQL request processing started.
kpidurationseconds	DECIMAL(19,3)	The number of seconds and fraction of seconds [5.102] the processing took.
kpidurationinterval	VARCHAR(25)	A formatted duration interval of how long the processing took [0 00:00:02].
kpimessage	VARCHAR(4000)	An exception message that could occur during the processing of the SQL or null if no exception.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
sessionid [fk]	DECIMAL(19,0)	A unique session id and foreign key to metrics_sessions_hist.
clienthost	VARCHAR(255)	Designates the client host name that connected to DV.
type	VARCHAR(20)	Indicates whether the session is of type ‘begin’ or ‘end’ or null.
bytesfromclient	DECIMAL(19,0)	The total number of bytes received from the client host during this session.
bytestoclient	DECIMAL(19,0)	The total number of bytes sent to the client host during this session.
rowsAffected	DECIMAL(19,0)	The number of rows affected or returned. The value may be -1.
sessiontype	VARCHAR(40)	Designates the type session connection [JDBC, HTTP, TASK, etc.].
sqlinvocationtype	VARCHAR(20)	The SQL invocation type indicates the type of request [SELECT, CALL, INSERT, UPDATE, DELETE, PROCEDURE, WITH].
kpirequestidmatch	DECIMAL(19,0)	The requestid of a match to a similar request templated so that the SQL template can be copied instead of processed from the start.
kpidescriptiontemplate	VARCHAR(4000)	The template of the actual client request or internal DV request. All literal values have been removed in order to normalize the SQL so it can be used to compare with other requests on a generic basis. The contents are 4000 characters or less which means that queries can be pushed down to the database.
kpidescriptiontemplateclob	CLOB	The template of the actual client request or internal DV request. All literal values have been removed in order to normalize the SQL so it can be used to compare with other requests on a generic basis. The contents are greater than 4000 characters.
description	CLOB	The actual client request or internal DV request.
resourceseqid	DECIMAL(19,0)	The resource sequence id is a foreign key back to vMetricsSqlResource.

resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in "resourceorigin".
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
frompath	VARCHAR(4000)	The from path is the database URL such as catalog.schema.table.
fromalias	VARCHAR(255)	The from path alias if applicable.
dataservicename	VARCHAR(255)	The published data service name.
catalogname	VARCHAR(255)	The name of the published catalog or null if not applicable.
schemaname	VARCHAR(255)	The name of the published schema or null if not applicable.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vEventRequestSqlResourcesCount Table

Returns a count of the number of unique requestid rows parsed for each metrics_request description (SQL). These resources were parsed. Derived from the following tables: METRICS_SQL_REQUEST, METRICS_SQL_RESOURCE, metrics_requests_hist and metrics_sessions_hist

Column Name	Column Type	Definition
sqlresourcecount	BIGINT	The total number of rows for the counttype.
counttype	VARCHAR(24)	The count type: TOTAL ROWS – the total number of rows for all categories. NUM TEMPLATE 4K ROWS – the number of rows in the varchar(4000) column. NUM TEMPLATE CLOB ROWS – the number of rows in the clob column. BOTH TEMPLATES NULL ROWS – the number of rows where both templates are null.

vExceededMemoryPercentRequests Table

Details on exceede memory queries that are occurring at the time this resource is executed. Derived from the system catalog table SYS_REQUESTS.

Column Name	Column Type	Definition
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
parentid [fk]	DECIMAL(19,0)	A foreign key back to the requestid within this table. Contains -1 when no parent reference.
sessionid [fk]	DECIMAL(19,0)	A unique session id and foreign key to metrics_sessions_hist.
transactionid	BIGINT	A reference to the DV transaction id.
ownerid	DECIMAL(19,0)	The resource owner unique id.
owner	VARCHAR(255)	The resource owner name.
domainname	VARCHAR(255)	The owner domain.
requesttype	VARCHAR(255)	The type of request that is stored in the description field. "SQL"=select statement, "SQL Script"=call statement executed, "XSLT, Basic, or Streaming Transformation", "Java Procedure (built-in)", etc.
status	VARCHAR(20)	The status of the request [READY, RUNNING, STARTED].
description	CLOB	The actual client request or internal DV request.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867

endtime	TIMESTAMP	The timestamp of when the request completed. E.g. 2020-01-06 10:19:36.899
totalduration	BIGINT	The total duration of the request measured in milli-seconds. Divide by 1000 to get minutes.
totaldurationformatted	VARCHAR(20)	A formatted total duration for display purposes.
serverrduration	BIGINT	The server duration of the request measured in milli-seconds. Divide by 1000 to get minutes. How much time the request spent in the DV server being processed.
rowsAffected	BIGINT	The number of rows affected or returned. The value may be -1.
maxmemory	BIGINT	The maximum amount of memory used by the request.
maxdisk	BIGINT	The maximum amount of disk used by the request. It is not good for a request to used disk.
currentmemory	BIGINT	The number of bytes of current memory used by the request.
currentdisk	BIGINT	The number of bytes of disk used by the request.
message	VARCHAR(65535)	The exception message if a request was not successful.
maxusedmemory	BIGINT	The maximum number of memory bytes used by the request.
currentusedmemory	BIGINT	The current number of memory bytes used by the request.
cisattrmanagedmax	BIGINT	Comes from a server attribute defined by commonValues.cisAttrManagedMemoryMax
cisattrequestquotapct	DECIMAL(10,6)	Comes from a server attribute defined by commonValues.cisAttrRequestQuotaPercent
maxmemallowedperrequest	BIGINT	The maximum memory in bytes allowed for a request as determined by a DV configuration parameter set by the admin.
datasourceid	INTEGER	The unique datasource id which comes from system.ALL_RESOURCES.
datasourcename	VARCHAR(255)	The datasource name.
datasourceparentpath	VARCHAR(4000)	The datasource parent path.

vGetSystemInformation Table

Returns nodehost and nodeport for a DV server. Derived from /System/Helpers/pGetSystemInformation() which in turn invokes /lib/util/GetProperty('SERVER_HOSTNAME') and /lib/util/GetProperty('SERVER_JDBC_PORT')

Column Name	Column Type	Definition
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vLongRunningRequests Table

Details on long running queries that are occurring at the time this resource is executed. Derived from the system catalog table SYS_REQUESTS.

Column Name	Column Type	Definition
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
parentid [fk]	DECIMAL(19,0)	A foreign key back to the requestid within this table. Contains -1 when no parent reference.
sessionid [fk]	DECIMAL(19,0)	A unique session id and foreign key to metrics_sessions_hist.
transactionid	BIGINT	A reference to the DV transaction id.
ownerid	DECIMAL(19,0)	The resource owner unique id.
owner	VARCHAR(255)	The resource owner name.
domainname	VARCHAR(255)	The owner domain.

requesttype	VARCHAR(255)	The type of request that is stored in the description field. “SQL”=select statement, “SQL Script”=call statement executed, “XSLT, Basic, or Streaming Transformation”, “Java Procedure (built-in)”, etc.
status	VARCHAR(20)	The status of the request [READY, RUNNING, STARTED].
description	CLOB	The actual client request or internal DV request.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
endtime	TIMESTAMP	The timestamp of when the request completed. E.g. 2020-01-06 10:19:36.899
totalduration	DECIMAL(19,0)	The total duration of the request measured in milli-seconds. Divide by 1000 to get minutes.
totaldurationformatted	VARCHAR(20)	A formatted total duration for display purposes.
serverrduration	BIGINT	The server duration of the request measured in milli-seconds. Divide by 1000 to get minutes. How much time the request spent in the DV server being processed.
rowsAffected	BIGINT	The number of rows affected or returned. The value may be -1.
maxmemory	BIGINT	The maximum amount of memory used by the request.
maxdisk	BIGINT	The maximum amount of disk used by the request. It is not good for a request to used disk.
currentmemory	BIGINT	The number of bytes of current memory used by the request.
currentdisk	BIGINT	The number of bytes of disk used by the request.
message	VARCHAR(65535)	The exception message if a request was not successful.
maxusedmemory	BIGINT	The maximum number of memory bytes used by the request.
currentusedmemory	BIGINT	The current number of memory bytes used by the request.
cisrequestrunminutes	INTEGER	The number of minutes the request has been running.
datasourceid	INTEGER	The unique datasource id which comes from system.ALL_RESOURCES.
datasourcename	VARCHAR(255)	The datasource name.
datasourceparentpath	VARCHAR(4000)	The datasource parent path.

vMetricsSqlColumns Table

Details of the parsed SQL columns for a user defined request. Derived from METRICS_SQL_COLUMNS with no other join.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
resourceseqid	DECIMAL(19,0)	The resource sequence id is a foreign key back to vMetricsSqlResource.
resourceorder	DECIMAL(10,0)	The order of the table or procedure resource within the context of the query.
columnorigtype	VARCHAR(20)	Column origination type determines the type of SQL the column came from. CALL, PRIMARY, COMPLEX, SUBQUERY
columnposition	NUMERIC(10,2)	The ordinal position of the column and derivied colunm. 1.0=1st column, 1.1=1st column of a derived field 1.2=2nd colunm of dervied field.
columnalias	VARCHAR(255)	The logical column (alias) name from the SQL statement. If not present then "columnName" is the alias.
columnname	VARCHAR(255)	The projected column name from the SQL statement. If * used, then same as actual column name
columnactual	VARCHAR(255)	The physical/source column name from the metadata table.
columntype	VARCHAR(4000)	The type of column.
columndirection	VARCHAR(10)	The direction of the column. For TABLE types it is always OUT. For PROCEDURE types it may be IN, OUT, INOUT, RESULT, RETURN.

columnderived	VARCHAR(10)	Determine whether this column is derived or not [true|false].
columnexpression	CLOB	The complete column expression.
columnfunctionlist	VARCHAR(4000)	A list of functions and occurrences that are used in the column expression. UPPER[3] would indicate that UPPER was used 3 times in the column expression.
columnnamelist	VARCHAR(4000)	A list of column names that make up the column expression.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetricsSqlRequest Table

Details of the parsed SQL for a user defined request. Derived from METRICS_SQL_REQUEST with no other join. Note the SQL Template is either stored in KPI_DESCRIPTION_TEMPLATE VARCHAR(4000) when the SQL statement is <= 4000 characters or KPI_DESCRIPTION_TEMPLATE_CLOB [CLOB/TEXT] when it is greater than 4000. The vast majority of SQL will be less than 4000. Since the odds are in favor of smaller SQL statements, an index can be placed on KPI_DESCRIPTION_TEMPLATE allowing faster access. However, if it is determined that the application has both then the user interface must account for the CLOB field in its queries.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
endtime	TIMESTAMP	The timestamp of when the request completed. E.g. 2020-01-06 10:19:36.899
totaldurationdisplay	VARCHAR(25)	The total duration formatted for display using the format 0 00:00:00.000.
dataservicename	VARCHAR(255)	The published data service name.
status	VARCHAR(20)	The status of the request [SUCCESS, FAILED].
message	VARCHAR(4000)	The exception message if a request was not successful.
kpiprocesstime	TIMESTAMP	The time the SQL request processing started.
kpistatus	VARCHAR(25)	The KPImetrics SQL request processing status [SUCCESS, FAIL]
kpimessage	VARCHAR(4000)	An exception message that could occur during the processing of the SQL or null if no exception.
kpidurationseconds	DECIMAL(19,3)	The number of seconds and fraction of seconds [5.102] the processing took.
kpidurationinterval	VARCHAR(25)	A formatted duration interval of how long the processing took [0 00:00:02].
kpirequestidmatch	DECIMAL(19,0)	The requestid of a match to a similar request templated so that the SQL template can be copied instead of processed from the start.
kpisqlinvocationtype	VARCHAR(20)	The SQL invocation type indicates the type of request [SELECT, CALL, INSERT, UPDATE, DELETE, PROCEDURE, WITH].
kpidescriptiontemplate	VARCHAR(4000)	The template of the actual client request or internal DV request. All literal values have been removed in order to normalize the SQL so it can be used to compare with other requests on a generic basis. The contents are 4000 characters or less which means that queries can be pushed down to the database.
kpidescriptiontemplateclob	CLOB	The template of the actual client request or internal DV request. All literal values have been removed in order to normalize the SQL so it can be used to compare with other requests on a generic basis. The contents are greater than 4000 characters.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

processednodehost	VARCHAR(255)	The hostname (nodehost) of the DV server that is performing the processing. Due to locking issues [mainly] with SQL Server, the architecture dictates that only one node performs the processing for all the other nodes.
processednodeport	INTEGER	The port (nodeport) of the DV server that is performing the processing.

vMetricsSqlRequestLineage Table

Details of the parsed SQL for a user defined request with the addition of the data source resource lineage. A request may contain many published resources. Each published resource may result in many datas ource resources. Therefore, the result may contain 0 to many resources and 0 to many data source resources.

REQUEST → RESOURCES → DATASOURCE LINEAGE

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
dataservicename	VARCHAR(255)	The published data service name.
kpidescriptiontemplate	VARCHAR(4000)	The template of the actual client request or internal DV request. All literal values have been removed in order to normalize the SQL so it can be used to compare with other requests on a generic basis. The contents are 4000 characters or less which means that queries can be pushed down to the database.
kpidescriptiontemplateclob	CLOB	The template of the actual client request or internal DV request. All literal values have been removed in order to normalize the SQL so it can be used to compare with other requests on a generic basis. The contents are greater than 4000 characters.
resourceseqid [k]	DECIMAL(19,0)	A generated unique id for this row.
resourcename_pub	VARCHAR(255)	The published name for the resource associated with “resourcepath”.
resourcepath_pub	VARCHAR(4000)	The published resource path for the resource associated with “resourcepath”.
resourcetype_pub	VARCHAR(255)	The published resource type for the resource associated with “resourcepath”.
catalogname	VARCHAR(255)	The name of the published catalog or null if not applicable.
schemaname	VARCHAR(255)	The name of the published schema or null if not applicable.
id	DECIMAL(19,0)	The resource id of the published resource.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
resourcesubtype	VARCHAR(40)	The DV resource subtype. The further qualifies the precise type of resource. E.g. NONE, SQL_TABLE (view), DATABASE_TABLE (datasource table), SQL_SCRIPT_PROCEDURE (DV procedure), OPERATION_PROCEDURE (web service procedure), plus many more.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
datasourcename	VARCHAR(255)	The datasource name.
datasourcetype	VARCHAR(255)	The type of published datasource [VirtualRelational | VirtualWsdl | CompositeWebService].
datasourcesubtype	VARCHAR(255)	The datasource subtype which further qualifies what type of datasource it is. E.g. RELATIONAL_DATA_SOURCE, XML_FILE_DATA_SOURCE.
datasourcepath	VARCHAR(255)	The DV path to the published datasource/dataservice [/services/databases/<name>, /services/webservices/<path|name>]
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').

nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetricsSqlRequestUniqueSqlTemplates Table

A unique grouping of parsed SQL templates derived from parsing the request "description". Since the description is stored as a VARCHAR(4000) field it allows push down on queries. Data is only stored in this field when <= 4000 characters. Derived from METRICS_SQL_REQUEST. KPI_DESCRIPTION_TEMPLATE.

Column Name	Column Type	Definition
numDescriptions	BIGINT	The number of related varchar(4000) descriptions. Use this number to determine if searching kpidescriptiontemplate is required.
dataservicename	VARCHAR(255)	The published data service name.
kpisqlinvocationtype	VARCHAR(20)	The SQL invocation type indicates the type of request [SELECT, CALL, INSERT, UPDATE, DELETE, PROCEDURE, WITH].
kpidescriptiontemplate	VARCHAR(4000)	The template of the actual client request or internal DV request. All literal values have been removed in order to normalize the SQL so it can be used to compare with other requests on a generic basis. The contents are 4000 characters or less which means that queries can be pushed down to the database.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetricsSqlRequestUniqueSqlTemplatesByUser Table

A unique grouping of parsed SQL templates sorted by user and derived from parsing the request "description". Since the description is stored as a VARCHAR(4000) field it allows push down on queries. Data is only stored in this field when <= 4000 characters. Derived from METRICS_SQL_REQUEST. KPI_DESCRIPTION_TEMPLATE and metrics_resources_usage_hist.

Column Name	Column Type	Definition
numDescriptions	BIGINT	The number of related varchar(4000) descriptions. Use this number to determine if searching kpidescriptiontemplate is required.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
dataservicename	VARCHAR(255)	The published data service name.
kpisqlinvocationtype	VARCHAR(20)	The SQL invocation type indicates the type of request [SELECT, CALL, INSERT, UPDATE, DELETE, PROCEDURE, WITH].
kpidescriptiontemplate	VARCHAR(4000)	The template of the actual client request or internal DV request. All literal values have been removed in order to normalize the SQL so it can be used to compare with other requests on a generic basis. The contents are 4000 characters or less which means that queries can be pushed down to the database.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetricsSqlRequestUniqueSqlTemplatesByUserByDate Table

A unique grouping of parsed SQL templates sorted by user and request date and derived from parsing the request "description". Since the description is stored as a VARCHAR(4000) field it allows push down on queries. Data is only stored in this field when <= 4000 characters. Derived

from METRICS_SQL_REQUEST. KPI_DESCRIPTION_TEMPLATE and metrics_resources_usage_hist.

Column Name	Column Type	Definition
numDescriptions	BIGINT	The number of related varchar(4000) descriptions. Use this number to determine if searching kpidescriptiontemplate is required.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
requestdate	DATE	The date [YYYY-MM-DD] of the request.
requestdatemonths	DECIMAL(19,2)	The number of months difference between the request date and the current date.
requestdatedays	DECIMAL(19,0)	The number of days difference between the request date and the current date.
dataservicename	VARCHAR(255)	The published data service name.
kpisqlinvocationtype	VARCHAR(20)	The SQL invocation type indicates the type of request [SELECT, CALL, INSERT, UPDATE, DELETE, PROCEDURE, WITH].
kpidescriptiontemplate	VARCHAR(4000)	The template of the actual client request or internal DV request. All literal values have been removed in order to normalize the SQL so it can be used to compare with other requests on a generic basis. The contents are 4000 characters or less which means that queries can be pushed down to the database.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetricsSqlRequestUniqueSqlTemplatesClob Table

A unique grouping of parsed SQL templates derived from parsing the request “description”. Warning. Since the description is stored as a CLOB field and push down on CLOB is not supported, this may cause high memory usage in CIS. Derived from METRICS_SQL_REQUEST. KPI_DESCRIPTION_TEMPLATE_CLOB.

Column Name	Column Type	Definition
numDescriptions	BIGINT	The number of related varchar(4000) descriptions. Use this number to determine if searching kpidescriptiontemplate is required.
connected	VARCHAR(255)	The published data service name.
kpisqlinvocationtype	VARCHAR(20)	The SQL invocation type indicates the type of request [SELECT, CALL, INSERT, UPDATE, DELETE, PROCEDURE, WITH].
kpidescriptiontemplateclob	CLOB	The template of the actual client request or internal DV request. All literal values have been removed in order to normalize the SQL so it can be used to compare with other requests on a generic basis. The contents are greater than 4000 characters.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetricsSqlRequestUniqueSqlTemplatesClobByUser Table

A unique grouping of parsed SQL templates sorted by user and derived from parsing the request “description”. Warning. Since the description is stored as a CLOB field and push down on CLOB is not supported, this may cause high memory usage in CIS. Derived from METRICS_SQL_REQUEST. KPI_DESCRIPTION_TEMPLATE_CLOB and metrics_resources_usage_hist.

Column Name	Column Type	Definition

numDescriptions	BIGINT	The number of related varchar(4000) descriptions. Use this number to determine if searching kpidescriptiontemplate is required.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
dataservicename	VARCHAR(255)	The published data service name.
kpisqlinvocationtype	VARCHAR(20)	The SQL invocation type indicates the type of request [SELECT, CALL, INSERT, UPDATE, DELETE, PROCEDURE, WITH].
kpidescriptiontemplateclob	CLOB	The template of the actual client request or internal DV request. All literal values have been removed in order to normalize the SQL so it can be used to compare with other requests on a generic basis. The contents are greater than 4000 characters.
Nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
Nodeport [k]	INTEGER	The http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetricsSqlRequestUniqueSqlTemplatesClobByUserByDate Table

A unique grouping of parsed SQL templates sorted by user and request date and derived from parsing the request "description". Warning. Since the description is stored as a CLOB field and push down on CLOB is not supported, this may cause high memory usage in CIS. Derived from METRICS_SQL_REQUEST. KPI_DESCRIPTION_TEMPLATE_CLOB and metrics_resources_usage_hist.

Column Name	Column Type	Definition
numDescriptions	BIGINT	The number of related varchar(4000) descriptions. Use this number to determine if searching kpidescriptiontemplate is required.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
requestdate	DATE	The date [YYYY-MM-DD] of the request.
requestdatemonths	DECIMAL(19,2)	The number of months difference between the request date and the current date.
requestdatedays	DECIMAL(19,0)	The number of days difference between the request date and the current date.
dataservicename	VARCHAR(255)	The published data service name.
kpisqlinvocationtype	VARCHAR(20)	The SQL invocation type indicates the type of request [SELECT, CALL, INSERT, UPDATE, DELETE, PROCEDURE, WITH].
kpidescriptiontemplateclob	CLOB	The template of the actual client request or internal DV request. All literal values have been removed in order to normalize the SQL so it can be used to compare with other requests on a generic basis. The contents are greater than 4000 characters.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetricsSqlResource Table

Details of the parsed SQL resources for a user defined request. Derived from METRICS_SQL_RESOURCE with no other joins.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
resourceseqid	DECIMAL(19,0)	The resource sequence id.

resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in “resourceorigin”.
resourceorigin	VARCHAR(30)	The original DV system table in which the resource originated from. [ALL_TABLES| ALL_PROCEDURES|ALL_WSDL_OPERATIONS|ALL_RESOURCES| ALL_COLUMNS|ALL_PARAMETERS]
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
sqlinvocationtype	VARCHAR(255)	The SQL invocation type indicates the type of request [SELECT, CALL, INSERT, UPDATE, DELETE, PROCEDURE, WITH].
frompath	VARCHAR(4000)	The from path is the database URL such as catalog.schema.table.
fromalias	VARCHAR(255)	The from path alias if applicable.
dataservicename	VARCHAR(255)	The published data service name.
catalogname	VARCHAR(255)	The name of the published catalog or null if not applicable.
schemaname	VARCHAR(255)	The name of the published schema or null if not applicable.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
messagetype	VARCHAR(10)	Typically null but if no lineage is found then [NO_LINEAGE].
message	VARCHAR(4000)	The informational message if no lineage was found such as [NO DATA SOURCE LINEAGE GENERATED].
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
processednodehost	VARCHAR(255)	The hostname (nodehost) of the DV server that is performing the processing. Due to locking issues [mainly] with SQL Server, the architecture dictates that only one node performs the processing for all the other nodes.
processednodeport	INTEGER	The port (nodeport) of the DV server that is performing the processing.

vMetricsSqlResourceLineage Table

Details of the parsed SQL resources data source lineage for a user defined request. Derived from METRICS_SQL_RESOURCE_LINEAGE with no other joins.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
id	DECIMAL(19,0)	A unique generated sequence id.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
resourceseqid	DECIMAL(19,0)	This field gets selected from vMetricsSqlResource table.
status	VARCHAR(20)	C=copied from existing database records. L=lineage generated from getResourceLineageDirectRecursive()
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in “resourceorigin”.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
resourcesubtype	VARCHAR(40)	The DV resource subtype. The further qualifies the precise type of resource. E.g. NONE, SQL_TABLE (view), DATABASE_TABLE (datasource table), SQL_SCRIPT_PROCEDURE (DV procedure), OPERATION_PROCEDURE (web service procedure), plus many more.
datasourceid	DECIMAL(19,0)	The unique datasource id which comes from system.ALL_RESOURCES.
datasourcename	VARCHAR(255)	The datasource name.

datasourcepath	VARCHAR(255)	The DV path to the published datasource/dataservice [/services/databases/<name>, /services/webservices/<path|name>]
datasourcesubtype	VARCHAR(255)	The datasource subtype which further qualifies what type of datasource it is. E.g. RELATIONAL_DATA_SOURCE, XML_FILE_DATA_SOURCE.
datasourcetype	VARCHAR(255)	The type of published datasource [VirtualRelational | VirtualWsdl | CompositeWebService].
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vMetricsSqlResourceLineageCountReport Table

Provides a count of METRICS_SQL_RESOURCE and METRICS_SQL_RESOURCE_LINEAGE tables under different scenarios:

TOTAL METRICS_SQL_RESOURCE COUNT

TOTAL METRICS_SQL_RESOURCE COUNT AVAILABLE TO PROCESS

TOTAL METRICS_SQL_RESOURCE NOT PROCESSED

TOTAL METRICS_SQL_RESOURCE NOT PROCESSED BY NODE

TOTAL METRICS_SQL_RESOURCE_LINEAGE PROCESSED

TOTAL METRICS_SQL_RESOURCE_LINEAGE PROCESSED BY nodehost

TOTAL METRICS_SQL_RESOURCE_LINEAGE PROCESSED BY STATUS=C=[C]opied lineage

TOTAL METRICS_SQL_RESOURCE_LINEAGE PROCESSED BY STATUS=L=[L]ineage generated

TOTAL METRICS_SQL_RESOURCE_LINEAGE PROCESSED BY nodehost, status=C=[C]opied lineage

TOTAL METRICS_SQL_RESOURCE_LINEAGE PROCESSED BY nodehost, status=L=[L]ineage generated

TOTAL METRICS_SQL_RESOURCE_LINEAGE PROCESSED BY nodehost, starttime, status=C=[C]opied lineage

TOTAL METRICS_SQL_RESOURCE_LINEAGE PROCESSED BY nodehost, starttime, status=L=[L]ineage generated

Column Name	Column Type	Definition
SECTION	CHAR(3)	A section label used for sorting purposes: 1.1, 1.2, 1.3, 2.1, 2.2, 3.1, 3.2 etc.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
statusdesc	VARCHAR(310)	A description of the count.
cnt	BIGINT	A count of rows.

vPublishedResourcePerRequest Table

Details on published requests correlated with user information. Derived from metrics_resources_usage_hist, metrics_sessions_hist, metrics_requests_hist.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
sessionid [fk]	DECIMAL(19,0)	A unique session id and foreign key to metrics_sessions_hist.
userid	DECIMAL(19,0)	The DV userid from the system.ALL_USERS tables.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.

type	VARCHAR(20)	Indicates whether the session is of type 'begin' or 'end' or null.
description	CLOB	The actual client request or internal DV request.
dataservicename	VARCHAR(255)	The published data service name.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct ALL_RESOURCES: null
rowsAffected	DECIMAL(19,0)	The number of rows affected or returned. The value may be -1.
bytesfromclient	DECIMAL(19,0)	The total number of bytes received from the client host during this session.
bytestoclient	DECIMAL(19,0)	The total number of bytes sent to the client host during this session.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
endtime	TIMESTAMP	The timestamp of when the request completed. E.g. 2020-01-06 10:19:36.899
totalduration	DECIMAL(19,0)	The total duration of the request measured in milli-seconds. Divide by 1000 to get minutes.
serverduration	DECIMAL(19,0)	The duration of the request within the server measured in milli-seconds. Divide by 1000 to get minutes.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vRequestDurationSqlTemplates Table

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
cluster	VARCHAR(255)	The name of the cluster or null if no cluster.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
parentid [fk]	DECIMAL(19,0)	A foreign key back to the requestid within this table. Contains -1 when no parent reference.
sessionid [fk]	DECIMAL(19,0)	A unique session id and foreign key to metrics_sessions_hist.
requesttype	VARCHAR(255)	The type of request that is stored in the description field. "SQL"=select statement, "SQL Script"=call statement executed, "XSLT, Basic, or Streaming Transformation", "Java Procedure (built-in)", etc.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867

endtime	TIMESTAMP	The timestamp of when the request completed. E.g. 2020-01-06 10:19:36.899
totalduration	DECIMAL(19,0)	The total duration of the request measured in milli-seconds. Divide by 1000 to get minutes.
totaldurationStr	VARCHAR(255)	The total duration formatted string such as 0 01:02:4.032
totaldurationDays	DECIMAL(12,4)	The total duration number of days.
totaldurationMinutes	DECIMAL(12,4)	The total duration number of minutes.
totaldurationHours	DECIMAL(12,4)	The total duration number of hours.
totaldurationSeconds	DECIMAL(12,4)	The total duration number of seconds incluing fraction of a second such as 4.032.
serverduration	DECIMAL(19,0)	The duration of the request within the server measured in milli-seconds. Divide by 1000 to get minutes.
rowsAffected	DECIMAL(19,0)	The number of rows affected or returned. The value may be -1.
maxmemory	DECIMAL(19,0)	The maximum amount of memory used by the request.
maxdisk	DECIMAL(19,0)	The maximum amount of disk used by the request. It is not good for a request to used disk.
message	CLOB	The exception message if a request was not successful.
status	VARCHAR(20)	The status of the request [SUCCESS, FAILED].
resourcekind	VARCHAR(20)	The type of resource that executed [system, user defined].
dataservicename	VARCHAR(255)	The published data service name.
description	CLOB	The actual client request or internal DV request.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
kpistatus	VARCHAR(25)	The KPImetrics SQL request processing status [SUCCESS, FAIL]
kpidescriptiontemplate	VARCHAR(4000)	The template of the actual client request or internal DV request. All literal values have been removed in order to normalize the SQL so it can be used to compare with other requests on a generic basis. The contents are 4000 characters or less which means that queries can be pushed down to the database.
numDescriptions	BIGINT	The number of related varchar(4000) descriptions. Use this number to determine if searching kpidescriptiontemplate is required.
kpidescriptiontemplateclob	CLOB	The processed SQL template where all literal values have been removed and the string is greater than 4000 characters. Does not allow for database push-down for searching. Strings are brought into DV for searching.
numDescriptionsClob	BIGINT	The number of related CLOB descriptions. Use this number to determine if searching numDescriptionsClob is required.

vRequestExpandedAll Table

Details of a request expanded with user information. All records are displayed. Derived from metrics_requests_hist and metrics_sessions_hist.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
cluster	VARCHAR(255)	The name of the cluster or null if no cluster.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
parentid [fk]	DECIMAL(19,0)	A foreign key back to the requestid within this table. Contains -1 when no parent reference.
status	VARCHAR(20)	The status of the request [SUCCESS, FAILED].

dataservicename	VARCHAR(255)	The published data service name.
resourcekind	VARCHAR(20)	The type of resource that executed [system, user defined].
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
endtime	TIMESTAMP	The timestamp of when the request completed. E.g. 2020-01-06 10:19:36.899
totalduration	DECIMAL(19,0)	The total duration of the request measured in milli-seconds. Divide by 1000 to get minutes.
serverduration	DECIMAL(19,0)	The duration of the request within the server measured in milli-seconds. Divide by 1000 to get minutes.
maxmemory	DECIMAL(19,0)	The maximum amount of memory used by the request.
maxdisk	DECIMAL(19,0)	The maximum amount of disk used by the request. It is not good for a request to used disk.
message	CLOB	The exception message if a request was not successful.
rowsAffected	DECIMAL(19,0)	The number of rows affected or returned. The value may be -1.
requesttype	VARCHAR(255)	The type of request that is stored in the description field. "SQL"=select statement, "SQL Script"=call statement executed, "XSLT, Basic, or Streaming Transformation", "Java Procedure (built-in)", etc.
description	CLOB	The actual client request or internal DV request.
sessionid [fk]	DECIMAL(19,0)	A unique session id and foreign key to metrics_sessions_hist.
type [k]	VARCHAR(20)	Indicates whether the session is of type 'begin' or 'end'.
clienthost	VARCHAR(255)	Designates the client host name that connected to DV.
sessiontype	VARCHAR(40)	Designates the type session connection [JDBC, HTTP, TASK, etc.].
bytesfromclient	DECIMAL(19,0)	The total number of bytes received from the client host during this session.
bytestoclient	DECIMAL(19,0)	The total number of bytes sent to the client host during this session.
userid	DECIMAL(19,0)	The DV userid from the system.ALL_USERS tables.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.

vRequestExpandedUD Table

Details of a request expanded with user information. Only records of resourcekind='user defined' are displayed. Derived from metrics_requests_hist and metrics_sessions_hist.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
cluster	VARCHAR(255)	The name of the cluster or null if no cluster.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
parentid [fk]	DECIMAL(19,0)	A foreign key back to the requestid within this table. Contains -1 when no parent reference.
status	VARCHAR(20)	The status of the request [SUCCESS, FAILED].
dataservicename	VARCHAR(255)	The published data service name.
resourcekind	VARCHAR(20)	The type of resource that executed [system, user defined].
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
endtime	TIMESTAMP	The timestamp of when the request completed. E.g. 2020-01-06 10:19:36.899
requestdatemonths	DECIMAL(19,2)	The number of months difference between the request date and the current date.
requestdatedays	DECIMAL(19,0)	The number of days difference between the request date and the current date.
totalduration	DECIMAL(19,0)	The total duration of the request measured in milli-seconds. Divide by 1000 to get minutes.

serverduration	DECIMAL(19,0)	The duration of the request within the server measured in milli-seconds. Divide by 1000 to get minutes.
maxmemory	DECIMAL(19,0)	The maximum amount of memory used by the request.
maxdisk	DECIMAL(19,0)	The maximum amount of disk used by the request. It is not good for a request to used disk.
message	CLOB	The exception message if a request was not successful.
rowsAffected	DECIMAL(19,0)	The number of rows affected or returned. The value may be -1.
requesttype	VARCHAR(255)	The type of request that is stored in the description field. "SQL"=select statement, "SQL Script"=call statement executed, "XSLT, Basic, or Streaming Transformation", "Java Procedure (built-in)", etc.
description	CLOB	The actual client request or internal DV request.
sessionid [fk]	DECIMAL(19,0)	A unique session id and foreign key to metrics_sessions_hist.
type [k]	VARCHAR(20)	Indicates whether the session is of type 'begin' or 'end'.
clienthost	VARCHAR(255)	Designates the client host name that connected to DV.
sessiontype	VARCHAR(40)	Designates the type session connection [JDBC, HTTP, TASK, etc.].
bytesfromclient	DECIMAL(19,0)	The total number of bytes received from the client host during this session.
bytestoclient	DECIMAL(19,0)	The total number of bytes sent to the client host during this session.
userid	DECIMAL(19,0)	The DV userid from the system.ALL_USERS tables.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.

vRequestDurationSqlTemplates Table

This view provides a way to view each request with additional information such as user, duration brokend down into a formatted string, days, minutes, hours, seconds and a SQL template that can be used to find other requests that are like this request. A SQL template is a normalized string of the actual requests with all literal values removed.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
cluster	VARCHAR(255)	The name of the cluster or null if no cluster.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
parentid [fk]	DECIMAL(19,0)	A foreign key back to the requestid within this table. Contains -1 when no parent reference.
sessionid [fk]	DECIMAL(19,0)	A unique session id and foreign key to metrics_sessions_hist.
requesttype	VARCHAR(255)	The type of request that is stored in the description field. "SQL"=select statement, "SQL Script"=call statement executed, "XSLT, Basic, or Streaming Transformation", "Java Procedure (built-in)", etc.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
endtime	TIMESTAMP	The timestamp of when the request completed. E.g. 2020-01-06 10:19:36.899
totalduration	DECIMAL(19,0)	The total duration of the request measured in milli-seconds. Divide by 1000 to get minutes.
totaldurationStr	VARCHAR(255)	The total duration formatted string such as 0 01:02:4.032
totaldurationDays	DECIMAL(12,4)	The total duration number of days.
totaldurationMinutes	DECIMAL(12,4)	The total duration number of minutes.

totaldurationHours	DECIMAL(12,4)	The total duration number of hours.
totaldurationSeconds	DECIMAL(12,4)	The total duration number of seconds incluing fraction of a second such as 4.032.
serverduration	DECIMAL(19,0)	The duration of the request within the server measured in milli-seconds. Divide by 1000 to get minutes.
rowsAffected	DECIMAL(19,0)	The number of rows affected or returned. The value may be -1.
maxmemory	DECIMAL(19,0)	The maximum amount of memory used by the request.
maxdisk	DECIMAL(19,0)	The maximum amount of disk used by the request. It is not good for a request to used disk.
message	CLOB	The exception message if a request was not successful.
status	VARCHAR(20)	The status of the user requests [SUCCESS, FAILED].
resourcekind	VARCHAR(20)	The type of resource that executed [system, user defined].
dataservicename	VARCHAR(255)	The published data service name.
description	CLOB	The actual client request or internal DV request.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
kpistatus	VARCHAR(25)	The KPImetrics SQL request processing status [SUCCESS, FAIL]
kpidescriptiontemplate	VARCHAR(4000)	The processed SQL template where all literal values have been removed and the string is less than or equal to 4000 characters. Allows for database push-down for searching. This is more efficient than CLOB for searching.
numDescriptions	BIGINT	The number of related varchar(4000) descriptions. Use this number to determine if searching kpidescriptiontemplate is required.
kpidescriptiontemplateclob	CLOB	The processed SQL template where all literal values have been removed and the string is greater than 4000 characters. Does not allow for database push-down for searching. Strings are brought into DV for searching.
numDescriptionsClob	BIGINT	The number of related CLOB descriptions. Use this number to determine if searching numDescriptionsClob is required.

vRequestsCountsByUser Table

A count of requests by user and date along with the following per request: avg rows, avg bytes from client, avg bytes to client, min total duration, max total duration, min server duration and max server duration.

Column Name	Column Type	Definition
userid	DECIMAL(19,0)	The DV userid from the system.ALL_USERS tables.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
dataservicename	VARCHAR(255)	The published data service name.
requestdate	DATE	The date [YYYY-MM-DD] of the request.
requestcount	BIGINT	The number of requests that match the group by criteria. GROUP BY userid, "user", "domain", dataservicename, CAST(starttime AS DATE), nodehost, nodeport
avgrowsaffected	DECIMAL(19,0)	The average number of rows affected or returned. The value may be -1.
avgbytesfromclient	DECIMAL(19,0)	The average number of bytes received from the client host during this session.
avgbytestoclient	DECIMAL(19,0)	The average number of bytes sent to the client host during this session.
mintotalduration	DECIMAL(19,0)	The minimum total duration.
maxtotalduration	DECIMAL(19,0)	The maximum total duration.
minServerDuration	DECIMAL(19,0)	The minimum duration of the request within the server measured in milli-seconds. Divide by 1000 to get minutes.

maxServerDuration	DECIMAL(19,0)	The maximum duration of the request within the server measured in milli-seconds. Divide by 1000 to get minutes.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vSessions Table

Details for user sessions. Equivalent to metrics_sessions_hist.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
cluster	VARCHAR(255)	The name of the cluster or null if no cluster.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
sessionid [fk]	DECIMAL(19,0)	A unique session id and foreign key to metrics_sessions_hist.
sessiontype	VARCHAR(40)	Designates the type session connection [JDBC, HTTP, TASK, etc.].
clienthost	VARCHAR(255)	Designates the client host name that connected to DV.
type [k]	VARCHAR(20)	Indicates whether the session is of type ‘begin’ or ‘end’.
logindate	DATE	The date [YYYY-MM-DD] of when the session began. Derived from logintime.
logoutdate	DATE	The date [YYYY-MM-DD] of when the session ended. Derived from logouttime.
logintime [k]	TIMESTAMP	The timestamp of when the session started.
logouttime	TIMESTAMP	The timestamp of when the session ended
status	VARCHAR(20)	The status of the session [ACTIVE, CLOSING].
totalduration	DECIMAL(19,0)	The total duration of the request measured in milli-seconds. Divide by 1000 to get minutes.
totalRequests	DECIMAL(19,0)	The total number of requests [metrics_requests_hist] executed for this session.
bytestoclient	DECIMAL(19,0)	The total number of bytes sent to the client host during this session.
bytesfromclient	DECIMAL(19,0)	The total number of bytes received from the client host during this session.
userid	DECIMAL(19,0)	The DV userid from the system.ALL_USERS tables.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.

vSessionvUserRequests Table

Details on requests generated by each user session.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
cluster	VARCHAR(255)	The name of the cluster or null if no cluster.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
logindate	DATE	The date [YYYY-MM-DD] of when the session began. Derived from logintime.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
clienthost	VARCHAR(255)	Designates the client host name that connected to DV.
sessiontype	VARCHAR(40)	Designates the type session connection [JDBC, HTTP, TASK, etc.].

sessioncount	BIGINT	The count of (sessionid) from metrics_sessions_hist where type='end' and grouped by cluster, nodehost, nodeport, logintime, user, domain, clienthost, sessiontype
requestcount	BIGINT	The count of (requestid) from metrics_requests_hist where type='end' and grouped by cluster, nodehost, nodeport, logintime, user, domain, clienthost, sessiontype
totalduration	DECIMAL(25,0)	The total duration of the request measured in milli-seconds. Divide by 1000 to get minutes.
totalbytestoclient	DECIMAL(25,0)	The total number of bytes sent to the client host during this session.
totalbytesfromclient	DECIMAL(25,0)	The total number of bytes received from the client host during this session.

resource Definitions

vAllResources_GroupBy_NodehostNodeport Table

Report that provides a count by nodehost and nodeport for the resources that exist in METRICS_ALL_RESOURCES

Column Name	Column Type	Definition
nodehost	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
cnt	BIGINT	The row count for the nodehost and nodeport.

vAllResources Table

Report of all resources currently in the DV repository. Derived from METRICS_ALL_RESOURCES and is loaded full each time based on a trigger. The old copy is deleted.

Column Name	Column Type	Definition
loaddate	TIMESTAMP	The timestamp of when the record was inserted or loaded.
resourceorigin	VARCHAR(30)	The original DV system table in which the resource originated from. [ALL_TABLES| ALL_PROCEDURES|ALL_WSDL_OPERATIONS|ALL_RESOURCES| ALL_COLUMNS|ALL_PARAMETERS]
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in "resourceorigin".
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
parentpath	VARCHAR(4000)	The parent path of the resource.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK. Also maybe TABLE or PRODEDURE.
resourcesubtype	VARCHAR(40)	The DV resource subtype. The further qualifies the precise type of resource. E.g. NONE, SQL_TABLE (view), DATABASE_TABLE (datasource table), SQL_SCRIPT_PROCEDURE (DV procedure), OPERATION_PROCEDURE (web service procedure), plus many more.
resourcelinktype	VARCHAR(40)	The type of resource the published "LINK" is pointing to such as [TABLE, PROCEDURE].
ownerid	DECIMAL(19,0)	The resource owner unique id.
ownername	VARCHAR(60)	The resource owner name.
ownerdomain	VARCHAR(255)	The domain of the ownername
guid	VARCHAR(1024)	The guid of the resource path. Eg. cff5fe78-c3c5-4a81-b134-d497cea5351e
datasourcetype	VARCHAR(255)	The type of published datasource [VirtualRelational | VirtualWsdl | CompositeWebService].

dataservicename	VARCHAR(255)	The published data service name.
catalogname	VARCHAR(255)	The name of the published catalog or null if not applicable.
schemaname	VARCHAR(255)	The name of the published schema or null if not applicable.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct ALL_RESOURCES: null
columnid	DECIMAL(19,0)	The DV column unique resource id.
columnname	VARCHAR(255)	The name of the column.
columntype	VARCHAR(255)	The column type base such as varchar, numeric, decimal.
columnsize	DECIMAL(19,0)	The column type size such as 255 if varchar(255) or 19 if decimal(19,2).
columnscale	DECIMAL(19,0)	The column type scale such as 2 if decimal(19,2).
columnorder	DECIMAL(9,0)	The order in which the column appears in the resource.
columndirection	DECIMAL(9,0)	The column direction such [4=OUT [table or scalar procedure, 1=IN, 3=CURSOR OUT]
annotation	CLOB	The resource annotation.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vResourceCount Table

Report of the count of resources where resourcekind=’user defined’. Derived from metrics_resources_usage_hist.

Column Name	Column Type	Definition
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
parentpath	VARCHAR(4000)	The parent path of the resource.
datasourcetype	VARCHAR(255)	The type of published datasource [VirtualRelational | VirtualWsdl | CompositeWebService].
dataservicename	VARCHAR(255)	The published data service name.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null

		ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct ALL_RESOURCES: null
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
countname	INTEGER	The number of resources for a given resource name.

vResourceCountDate Table

Report of the count of resources by date where resourcekind='user defined'. Derived from metrics_resources_usage_hist.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
requestdate	DATE	The date [YYYY-MM-DD] of the request.
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in "resourceorigin".
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
parentpath	VARCHAR(4000)	The parent path of the resource.
resourcekind	VARCHAR(20)	The type of resource that executed [system, user defined].
datasourcetype	VARCHAR(255)	The type of published datasource [VirtualRelational | VirtualWsdl | CompositeWebService].
dataservicename	VARCHAR(255)	The published data service name.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct ALL_RESOURCES: null
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
countname	INTEGER	The number of resources for a given resource name.
requestdatemonths	DECIMAL(19,2)	The number of months difference between the request date and the current date.
requestdatedays	DECIMAL(19,0)	The number of days difference between the request date and the current date.

vResourceCountUsers Table

Report of the count of resources by user where resourcekind='user defined'. Derived from metrics_resources_usage_hist.

Column Name	Column Type	Definition
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in "resourceorigin".
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
parentpath	VARCHAR(4000)	The parent path of the resource.
resourcekind	VARCHAR(20)	The type of resource that executed [system, user defined].
datasourcetype	VARCHAR(255)	The type of published datasource [VirtualRelational | VirtualWsdl | CompositeWebService].
dataservicename	VARCHAR(255)	The published data service name.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct ALL_RESOURCES: null
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
countname	INTEGER	The number of resources for a given resource name.

vResourceCountUsersDate Table

Report of the count of resources by user and date where resourcekind='user defined'. Derived from metrics_resources_usage_hist.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
requestdate	DATE	The date [YYYY-MM-DD] of the request.
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in "resourceorigin".
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.

parentpath	VARCHAR(4000)	The parent path of the resource.
resourcekind	VARCHAR(20)	The type of resource that executed [system, user defined].
datasourcetype	VARCHAR(255)	The type of published datasource [VirtualRelational | VirtualWsdl | CompositeWebService].
dataservicename	VARCHAR(255)	The published data service name.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct ALL_RESOURCES: null
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
countname	INTEGER	The number of resources for a given resource name.
requestdatemonths	DECIMAL(19,2)	The number of months difference between the request date and the current date.
requestdatedays	DECIMAL(19,0)	The number of days difference between the request date and the current date.

vResourceDistinctPublishedDatabases Table

This table returns a list of all resources published under a database on the DV server derived from metrics_resources_usage_hist.

Column Name	Column Type	Definition
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
datasourcetype	VARCHAR(255)	The type of published datasource [VirtualRelational | VirtualWsdl | CompositeWebService].
dataservicename	VARCHAR(255)	The published data service name.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct ALL_RESOURCES: null

vResourceDistinctPublishedResources Table

This table returns a list of all distinct resources published under a database on the DV server derived from metrics_resources_usage_hist.

Column Name	Column Type	Definition
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
datasourcetype	VARCHAR(255)	The type of published datasource [VirtualRelational | VirtualWsdl | CompositeWebService].
dataservicename	VARCHAR(255)	The published data service name.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct ALL_RESOURCES: null

vResourceDistinctPublishedWebServices Table

This table returns a list of all resources published as a web service operation on the DV server derived from metrics_resources_usage_hist.

Column Name	Column Type	Definition
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
datasourcetype	VARCHAR(255)	The type of published datasource [VirtualRelational | VirtualWsdl | CompositeWebService].
dataservicename	VARCHAR(255)	The published data service name.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct

		ALL_RESOURCES: null

vResourceDistinctResources Table

This table returns a list of all distinct resources on the DV server derived from metrics_resources_usage_hist.

Column Name	Column Type	Definition
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
datasourcetype	VARCHAR(255)	The type of published datasource [VirtualRelational | VirtualWsdl | CompositeWebService].
dataservicename	VARCHAR(255)	The published data service name.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct ALL_RESOURCES: null

vResourcesPublishedNotUsed Table

Report of all published resources present on the DV server that have not been used derived from METRICS_ALL_RESOURCES, metrics_resources_usage_hist.

Column Name	Column Type	Definition
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
parentpath	VARCHAR(4000)	The parent path of the resource.
dataservicename	VARCHAR(255)	The published data service name.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources

		e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct ALL_RESOURCES: null

vResourceUsageAll Table

Report of all resources historically where resourcekind='user defined' and 'system'. Derived from METRICS_ALL_RESOURCES, metrics_resources_usage_hist.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
cluster	VARCHAR(255)	The name of the cluster or null if no cluster.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
parentid [fk]	DECIMAL(19,0)	A foreign key back to the requestid within this table. Contains -1 when no parent reference.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
sessionid [fk]	DECIMAL(19,0)	A unique session id and foreign key to metrics_sessions_hist.
requestdate	DATE	The date [YYYY-MM-DD] of the request.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
endtime	TIMESTAMP	The timestamp of when the request completed. E.g. 2020-01-06 10:19:36.899
requestdatemonths	DECIMAL(19,2)	The number of months difference between the request date and the current date.
requestdatedays	DECIMAL(19,0)	The number of days difference between the request date and the current date.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct ALL_RESOURCES: null
dataservicename	VARCHAR(255)	The published data service name.
datasourcepath	VARCHAR(255)	The DV path to the published datasource/dataservice [/services/databases/<name>, /services/webservices/<path|name>]
datasourcetype	VARCHAR(255)	The type of published datasource [VirtualRelational | VirtualWsdl | CompositeWebService].
resourcekind	VARCHAR(20)	The type of resource that executed [system, user defined].
resourceguid	VARCHAR(40)	The guid of the resource path. Eg. cff5fe78-c3c5-4a81-b134-d497cea5351e
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in "resourceorigin".
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
parentname	VARCHAR(255)	The parent name of the resource. A.k.a. schema name if from published database.
grandparentname	VARCHAR(255)	The grand-parent name of the resource. A.k.a. catalog name if from published database.

resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
parentpath	VARCHAR(4000)	The parent path of the resource.
userid	DECIMAL(19,0)	The DV userid from the system.ALL_USERS tables.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
group	VARCHAR(255)	The group of the user who connected.

vResourceUsageUD Table

Report of all resources where resourcekind='user defined'. Derived from metrics_resources_usage_hist.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
cluster	VARCHAR(255)	The name of the cluster or null if no cluster.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
parentid [fk]	DECIMAL(19,0)	A foreign key back to the requestid within this table. Contains -1 when no parent reference.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
sessionid [fk]	DECIMAL(19,0)	A unique session id and foreign key to metrics_sessions_hist.
requestdate	DATE	The date [YYYY-MM-DD] of the request.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
endtime	TIMESTAMP	The timestamp of when the request completed. E.g. 2020-01-06 10:19:36.899
requestdatemonths	DECIMAL(19,2)	The number of months difference between the request date and the current date.
requestdatedays	DECIMAL(19,0)	The number of days difference between the request date and the current date.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct ALL_RESOURCES: null
dataservicename	VARCHAR(255)	The published data service name.
datasourcepath	VARCHAR(255)	The DV path to the published datasource/dataservice [/services/databases/<name>, /services/webservices/<path|name>]
datasourcetype	VARCHAR(255)	The type of published datasource [VirtualRelational | VirtualWsdl | CompositeWebService].
resourcekind	VARCHAR(20)	The type of resource that executed [system, user defined].
resourceguid	VARCHAR(40)	The guid of the resource path. Eg. cff5fe78-c3c5-4a81-b134-d497cea5351e
resourceid	DECIMAL(19,0)	The unique resource id which comes from the DV system tables listed in "resourceorigin".
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.

resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
parentname	VARCHAR(255)	The parent name of the resource. A.k.a. schema name if from published database.
grandparentname	VARCHAR(255)	The grand-parent name of the resource. A.k.a. catalog name if from published database.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
parentpath	VARCHAR(4000)	The parent path of the resource.
userid	DECIMAL(19,0)	The DV userid from the system.ALL_USERS tables.
user	VARCHAR(255)	The user name who connected for this session.
domain	VARCHAR(255)	The domain of the user who connected.
group	VARCHAR(255)	The group of the user who connected.

resourceDataCount Definitions

getResourceDataCount Procedure

This procedure returns a list of the top N most frequently accessed resources for the specified data range. Each row includes a count of the number of rows of data each resource contains.

Column Name	Column Type	Definition
topN	INTEGER	IN – The top number of records to return.
fromdate	DATE	IN - The from date range.
todate	DATE	IN - The to date range.
resourcepath [k]	VARCHAR(4000)	OUT - The DV path to the resource.
resourcename	VARCHAR(255)	OUT - The name of the resource. A.k.a. table or procedure name.
resourcetype [k]	VARCHAR(40)	OUT - The DV resource type. If published, then it will be LINK.
parentpath	VARCHAR(4000)	OUT - The parent path of the resource.
dataservicename	VARCHAR(255)	OUT - The published data service name.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct ALL_RESOURCES: null
resourceusagecount	BIGINT	OUT – The number of resources.
datarowcount	BIGINT	OUT – The data count for the resource.

resourceMetadata Definitions

vResourceListAllPublishedResources Table

This table returns a list of all published resources present on the DV server metadata catalog derived from the cached METRICS_ALL_RESOURCES.

Column Name	Column Type	Definition
resourcename	VARCHAR(255)	The name of the resource. A.k.a. table or procedure name.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
publishedtype	VARCHAR(11)	The type of datasource that the published resource belongs to [DATABASE|WEB_SERVICE]
datasourcetype	VARCHAR(255)	The type of published datasource [VirtualRelational | VirtualWsdl | CompositeWebService].
dataservicename	VARCHAR(255)	The published data service name.
categoryname	VARCHAR(255)	The category name of the resource. A normalized name from either the resource name, parent name or grand-parent name. ALL_TABLES: TABLE_NAME or SCHEMA_NAME if not null ALL_COLUMNS: TABLE_NAME or SCHEMA_NAME if not null ALL_PROCEDURES: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_PARAMETERS: PROCEDURE_NAME or SCHEMA_NAME if not null ALL_WSDL_OPERATIONS: If legacy web service then it is the service name otherwise it is the parent folder name of the web service. If no parent folder name then it is the web service dataservice name. e.g. Legacy: /services/webservices/system/admin/**resource**/operations/getUsedResources e.g. Non-Legacy: /services/webservices/GrandParent/**Parent**/CustomerWS/LookupProduct e.g. Non-Legacy: /services/webservices/**CustomerWS**/LookupProduct ALL_RESOURCES: null

systemUsage Definitions

vCpuMemUtilization Table

Details on system CPU and memory utilization. Derived from METRICS_CPU_MEMORY_CHECKER.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
currenttimestamp	TIMESTAMP	The current event timestamp.
cpuusedpercent	DECIMAL(10,2)	The system CPU utilization (used) percent.
memoryusedmb	DECIMAL(10,2)	The system memory used in MB. This will always be showing high since DV takes the configured amount of memory at startup time. This is only a good guage of potential other processes running on the same server as DV if a large fluctuation is detected.
memoryavailmb	DECIMAL(10,2)	The system memory available in MB. Counterpart to the above statement.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vDatasourceConnectionChanges Table

Details on data sources connection changes. Derived from METRICS_SYS_DATASOURCES which was cached from /services/databases/system/SYS_DATASOURCES.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.

historyid [pk]	DECIMAL(19,0)	A unique generated id for this row.
previoushistoryid [fk]	DECIMAL(19,0)	A foreign key back to the same table and the historyid which connects one record to another record ina sort of a lineage of history which allows the viewer to place the events in order.
iscurrent	INTEGER	Indicates the current datasource history record. 1=current. 0=not current.
isdeleted	INTEGER	Indicates whether the datasource has been deleted since the last time the Cache_SYS_DATASOURCES was executed. 1=deleted. 0=not deleted.
eventtime	TIMESTAMP	The timestamp of when the row was recorded to the table.
sourceid	DECIMAL(19,0)	The unique DV datasource id.
sourcename	VARCHAR(255)	The DV datasource name.
sourcetype	VARCHAR(60)	The adapter type of the datasource such as Oracle, PostgresSQL, SqlServer, Composite, Rest, Wsdl, File, Netezza, MsExcel(POI), Ldap, MySql, XmlFile, Jdbc, DB2, etc.
sourcecategory	VARCHAR(60)	The category of the datasource such as RELATIONAL, FILE, SERVICE, etc.
ownerid	DECIMAL(19,0)	The resource owner unique id.
owner	VARCHAR(255)	The resource owner name.
parentpath	VARCHAR(4000)	The parent path of the resource.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
status	VARCHAR(20)	The status of the datasource [UP, DOWN, DISABLED].
statuschangeflag	DECIMAL(19,0)	The status change flag indicates if the datasource changed from the previous time. NULL if first entry. 0=no change, 1=changed since last record.
connectionchangeflag	DECIMAL(19,0)	The connection change flag indicates if the datasource connection information changed since the previous time. NULL if first entry. 0=no change. 1=changed since last record.
connectionchangelist	VARCHAR(255)	The connection change list provides information on what changed. STATUS_CHANGE - The status of the datasource has changed such as UP to DOWN. CONN_HOST_NAME - The connection host name changed since the last record. CONN_PORT - The connection port changed since the last record. CONN_LOGIN_NAME - The connection login name changed since the last record. CONN_LOGIN_VALUE - The connection login value changed since the last record. CONN_DATABASE_NAME - The connection database name changed since the last record. CONN_DATABASE_TYPE - The connection dataase type changed since the last record. CONN_URL_OR_ROOT - The connection URL or root changed since the last record. CONN_CACHE_STATUS - The connection cache status resource path changed since the last record. CONN_CACHE_TRACKING - The connection cache tracking resource path changed since the last record. CONN_CREATED_BY - The connection created by name changed since the last record. CONN_CREATED_DATE - The connection created date changed since the last record. CONN_MODIFIED_DATE - The connection modified date changed since the last record. CONN_MODIFIED_BY - The connection modified by name changed since the last record.
numrequests	DECIMAL(19,0)	The numer of requests that have been executed during the datasource lifetime.
activerequests	DECIMAL(19,0)	The number of active requests being executed at this point in time (eventtime).
maxconn	DECIMAL(19,0)	The maximum configured connections on the advanced tab.
numcurrentconn	DECIMAL(19,0)	The number of current connections at this point in time (eventtime).
numinuseconn	DECIMAL(19,0)	The number of “in use” connections at this point in time (eventtime).
numlogins	DECIMAL(19,0)	The number of logins during the datasource lifetime.
numlogouts	DECIMAL(19,0)	The number of logouts during the datasource lifetime.
bytesto	DECIMAL(19,0)	The number of estimated bytes sent into the datasource.

bytesfrom	DECIMAL(19,0)	The numer of estimated bytes sent from the datasource.
message	VARCHAR(4000)	The exception message if a request was not successful.
connhostname	VARCHAR(4000)	The connection host name or ip address.
connport	INTEGER	The connection port.
connloginname	VARCHAR(255)	The connection login name.
connloginvalue	VARCHAR(255)	The connection login value.
conndatabasename	VARCHAR(255)	The connection database name.
conndatabasetype	VARCHAR(255)	The connection database type such as Oracle, PostgresSQL, SqlServer, Composite, Rest, Wsdl, File, Netezza, MsExcel(POI), Ldap, MySql, XmlFile, Jdbc, DB2, etc.
connurlorroot	VARCHAR(4000)	The connection URL or root. URL if relational. Root if file.
conncachestatus	VARCHAR(4000)	The connection cache status table path or null if not applicable.
conncachetracking	VARCHAR(4000)	The connection cache tracking table path or null if not applicable.
conncreatedby	VARCHAR(255)	The connection created by user name.
conncreateddate	TIMESTAMP	The connection created date (timestamp).
connmodifiedby	VARCHAR(255)	The connection modified by user name.
connmodifieddate	TIMESTAMP	The connection modified date (timestamp).
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vDatasourceCurrentStatusChanges Table

Details on “current” data source status changes. Derived from METRICS_SYS_DATASOURCES WHERE ISCURRENT=1 which was cached from /services/databases/system/SYS_DATASOURCES.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
iscurrent	INTEGER	Indicates the current datasource history record. 1=current. 0=not current.
sourceid	DECIMAL(19,0)	The unique DV datasource id.
sourcename	VARCHAR(255)	The DV datasource name.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
historyid [pk]	DECIMAL(19,0)	A unique generated id for this row.
previoushistoryid [fk]	DECIMAL(19,0)	A foreign key back to the same table and the historyid which connects one record to another record ina sort of a lineage of history which allows the viewer to place the events in order.
new_status	VARCHAR(20)	The new status change of the datasource [UP, DOWN, DISABLED].
previous_status	VARCHAR(20)	The previous status of the datasource [UP, DOWN, DISABLED].
new_eventtime	TIMESTAMP	The new timestamp of when the row was recorded to the table.
previous_eventtime	TIMESTAMP	The previous timestamp of when the row was recorded to the table.
elapsed_time_minutes	DECIMAL(18,2)	The elapsed time in minutes between the previous and new eventtime.
status_change	SMALLINT	The changed status of the datasource [UP, DOWN, DISABLED].
new_message	VARCHAR(4000)	The new exception message or null if not applicable.
previous_message	VARCHAR(4000)	The previous exception message or null if not applicable.

connectionchangeflag	DECIMAL(19,0)	The connection change flag indicates if the datasource connection information changed since the previous time. NULL if first entry. 0=no change. 1=changed since last record.
new_connectionchangelist	VARCHAR(255)	The "new" connection change list provides information on what changed. STATUS_CHANGE - The status of the datasource has changed such as UP to DOWN. CONN_HOST_NAME - The connection host name changed since the last record. CONN_PORT - The connection port changed since the last record. CONN_LOGIN_NAME - The connection login name changed since the last record. CONN_LOGIN_VALUE - The connection login value changed since the last record. CONN_DATABASE_NAME - The connection database name changed since the last record. CONN_DATABASE_TYPE - The connection dataase type changed since the last record. CONN_URL_OR_ROOT - The connection URL or root changed since the last record. CONN_CACHE_STATUS - The connection cache status resource path changed since the last record. CONN_CACHE_TRACKING - The connection cache tracking resource path changed since the last record. CONN_CREATED_BY - The connection created by name changed since the last record. CONN_CREATED_DATE - The connection created date changed since the last record. CONN_MODIFIED_DATE - The connection modified date changed since the last record. CONN_MODIFIED_BY - The connection modified by name changed since the last record.
previous_connectionchangelist	VARCHAR(255)	The "previous" connection change list provides information on what happened previously. STATUS_CHANGE - The status of the datasource has changed such as UP to DOWN. CONN_HOST_NAME - The connection host name changed since the last record. CONN_PORT - The connection port changed since the last record. CONN_LOGIN_NAME - The connection login name changed since the last record. CONN_LOGIN_VALUE - The connection login value changed since the last record. CONN_DATABASE_NAME - The connection database name changed since the last record. CONN_DATABASE_TYPE - The connection dataase type changed since the last record. CONN_URL_OR_ROOT - The connection URL or root changed since the last record. CONN_CACHE_STATUS - The connection cache status resource path changed since the last record. CONN_CACHE_TRACKING - The connection cache tracking resource path changed since the last record. CONN_CREATED_BY - The connection created by name changed since the last record. CONN_CREATED_DATE - The connection created date changed since the last record. CONN_MODIFIED_DATE - The connection modified date changed since the last record. CONN_MODIFIED_BY - The connection modified by name changed since the last record.

vDatasourceStatusChanges Table

Details on "all" data source status changes. Derived from METRICS_SYS_DATASOURCES which was cached from /services/databases/system/SYS_DATASOURCES.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
iscurrent	INTEGER	Indicates the current datasource history record. 1=current. 0=not current.
sourceid	DECIMAL(19,0)	The unique DV datasource id.
sourcename	VARCHAR(255)	The DV datasource name.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.

historyid [pk]	DECIMAL(19,0)	A unique generated id for this row.
previoushistoryid [fk]	DECIMAL(19,0)	A foreign key back to the same table and the historyid which connects one record to another record ina sort of a lineage of history which allows the viewer to place the events in order.
new_status	VARCHAR(20)	The “new” changed status of the datasource [UP, DOWN, DISABLED].
previous_status	VARCHAR(20)	The “previous” changed status of the datasource [UP, DOWN, DISABLED].
new_eventtime	TIMESTAMP	The new timestamp of when the row was recorded to the table.
previous_eventtime	TIMESTAMP	The previous timestamp of when the row was recorded to the table.
elapsed_time_minutes	DECIMAL(18,2)	The total elapsed time in minutes.
status_change	SMALLINT	The changed status of the datasource [UP, DOWN, DISABLED].
new_message	VARCHAR(4000)	The new exception message or null if not applicable.
previous_message	VARCHAR(4000)	The previous exception message or null if not applicable.
connectionchangeflag	DECIMAL(19,0)	The connection change flag indicates if the datasource connection information changed since the previous time. NULL if first entry. 0=no change. 1=changed since last record.
new_connectionchangelist	VARCHAR(255)	The “new” connection change list provides information on what changed. STATUS_CHANGE - The status of the datasource has changed such as UP to DOWN. CONN_HOST_NAME - The connection host name changed since the last record. CONN_PORT - The connection port changed since the last record. CONN_LOGIN_NAME - The connection login name changed since the last record. CONN_LOGIN_VALUE - The connection login value changed since the last record. CONN_DATABASE_NAME - The connection database name changed since the last record. CONN_DATABASE_TYPE - The connection dataase type changed since the last record. CONN_URL_OR_ROOT - The connection URL or root changed since the last record. CONN_CACHE_STATUS - The connection cache status resource path changed since the last record. CONN_CACHE_TRACKING - The connection cache tracking resource path changed since the last record. CONN_CREATED_BY - The connection created by name changed since the last record. CONN_CREATED_DATE - The connection created date changed since the last record. CONN_MODIFIED_DATE - The connection modified date changed since the last record. CONN_MODIFIED_BY - The connection modified by name changed since the last record.
previous_connectionchangelist	VARCHAR(255)	The “previous” connection change list provides information on what happened previously. STATUS_CHANGE - The status of the datasource has changed such as UP to DOWN. CONN_HOST_NAME - The connection host name changed since the last record. CONN_PORT - The connection port changed since the last record. CONN_LOGIN_NAME - The connection login name changed since the last record. CONN_LOGIN_VALUE - The connection login value changed since the last record. CONN_DATABASE_NAME - The connection database name changed since the last record. CONN_DATABASE_TYPE - The connection dataase type changed since the last record. CONN_URL_OR_ROOT - The connection URL or root changed since the last record. CONN_CACHE_STATUS - The connection cache status resource path changed since the last record. CONN_CACHE_TRACKING - The connection cache tracking resource path changed since the last record. CONN_CREATED_BY - The connection created by name changed since the last record. CONN_CREATED_DATE - The connection created date changed since the last record. CONN_MODIFIED_DATE - The connection modified date changed since the last record. CONN_MODIFIED_BY - The connection modified by name changed since the last record.

vDatasourceUsage Table

Details on data sources usage. Derived from METRICS_SYS_DATASOURCES which was cached from /services/databases/system/SYS_DATASOURCES.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
historyid [pk]	DECIMAL(19,0)	A unique generated id for this row.
previoushistoryid [fk]	DECIMAL(19,0)	A foreign key back to the same table and the historyid which connects one record to another record ina sort of a lineage of history which allows the viewer to place the events in order.
iscurrent	INTEGER	Indicates the current datasource history record. 1=current. 0=not current.
isdeleted	INTEGER	Indicates whether the datasource has been deleted since the last time the Cache_SYS_DATASOURCES was executed. 1=deleted. 0=not deleted.
eventtime	TIMESTAMP	The timestamp of when the row was recorded to the table.
sourceid	DECIMAL(19,0)	The unique DV datasource id.
sourcename	VARCHAR(255)	The DV datasource name.
sourcetype	VARCHAR(60)	The adapter type of the datasource such as Oracle, PostgresSQL, SqlServer, Composite, Rest, Wsdl, File, Netezza, MsExcel(POI), Ldap, MySql, XmlFile, Jdbc, DB2, etc.
sourcecategory	VARCHAR(60)	The category of the datasource such as RELATIONAL, FILE, SERVICE, etc.
ownerid	DECIMAL(19,0)	The resource owner unique id.
owner	VARCHAR(255)	The resource owner name.
parentpath	VARCHAR(4000)	The parent path of the resource.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
status	VARCHAR(20)	The status of the datasource [UP, DOWN, DISABLED].
statuschangeflag	DECIMAL(19,0)	The status change flag indicates if the datasource changed from the previous time. NULL if first entry. 0=no change, 1=changed since last record.
connectionchangeflag	DECIMAL(19,0)	The connection change flag indicates if the datasource connection information changed since the previous time. NULL if first entry. 0=no change. 1=changed since last record.
connectionchangelist	VARCHAR(255)	The connection change list provides information on what changed. STATUS_CHANGE - The status of the datasource has changed such as UP to DOWN. CONN_HOST_NAME - The connection host name changed since the last record. CONN_PORT - The connection port changed since the last record. CONN_LOGIN_NAME - The connection login name changed since the last record. CONN_LOGIN_VALUE - The connection login value changed since the last record. CONN_DATABASE_NAME - The connection database name changed since the last record. CONN_DATABASE_TYPE - The connection dataase type changed since the last record. CONN_URL_OR_ROOT - The connection URL or root changed since the last record. CONN_CACHE_STATUS - The connection cache status resource path changed since the last record. CONN_CACHE_TRACKING - The connection cache tracking resource path changed since the last record. CONN_CREATED_BY - The connection created by name changed since the last record. CONN_CREATED_DATE - The connection created date changed since the last record. CONN_MODIFIED_DATE - The connection modified date changed since the last record. CONN_MODIFIED_BY - The connection modified by name changed since the last record.
numrequests	DECIMAL(19,0)	The numer of requests that have been executed during the datasource lifetime.
activerequests	DECIMAL(19,0)	The number of active requests being executed at this point in time (eventtime).
maxconn	DECIMAL(19,0)	The maximum configured connections on the advanced tab.
numcurrentconn	DECIMAL(19,0)	The number of current connections at this point in time (eventtime).

numinuseconn	DECIMAL(19,0)	The number of "in use" connections at this point in time (eventtime).
numlogins	DECIMAL(19,0)	The number of logins during the datasource lifetime.
numlogouts	DECIMAL(19,0)	The number of logouts during the datasource lifetime.
bytesto	DECIMAL(19,0)	The number of estimated bytes sent into the datasource.
bytesfrom	DECIMAL(19,0)	The numer of estimated bytes sent from the datasource.
message	VARCHAR(4000)	The exception message if a request was not successful.
connhostname	VARCHAR(4000)	The connection host name or ip address.
connport	INTEGER	The connection port.
connloginname	VARCHAR(255)	The connection login name.
connloginvalue	VARCHAR(255)	The connection login value.
conndatabasename	VARCHAR(255)	The connection database name.
conndatabasetype	VARCHAR(255)	The connection database type such as Oracle, PostgresSQL, SqlServer, Composite, Rest, Wsdl, File, Netezza, MsExcel(POI), Ldap, MySql, XmlFile, Jdbc, DB2, etc.
connurlorroot	VARCHAR(4000)	The connection URL or root. URL if relational. Root if file.
conncachestatus	VARCHAR(4000)	The connection cache status table path or null if not applicable.
conncachetracking	VARCHAR(4000)	The connection cache tracking table path or null if not applicable.
conncreatedby	VARCHAR(255)	The connection created by user name.
conncreateddate	TIMESTAMP	The connection created date (timestamp).
connmodifiedby	VARCHAR(255)	The connection modified by user name.
connmodifieddate	TIMESTAMP	The connection modified date (timestamp).
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vDatasourceUsageCurrent Table

Details on current data sources usage. Derived from METRICS_SYS_DATASOURCES which was cached from /services/databases/system/SYS_DATASOURCES

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
historyid [pk]	DECIMAL(19,0)	A unique generated id for this row.
previoushistoryid [fk]	DECIMAL(19,0)	A foreign key back to the same table and the historyid which connects one record to another record ina sort of a lineage of history which allows the viewer to place the events in order.
iscurrent	INTEGER	Indicates the current datasource history record. 1=current. 0=not current.
isdeleted	INTEGER	Indicates whether the datasource has been deleted since the last time the Cache_SYS_DATASOURCES was executed. 1=deleted. 0=not deleted.
eventtime	TIMESTAMP	The timestamp of when the row was recorded to the table.
sourceid	DECIMAL(19,0)	The unique DV datasource id.
sourcename	VARCHAR(255)	The DV datasource name.
sourcetype	VARCHAR(60)	The adapter type of the datasource such as Oracle, PostgresSQL, SqlServer, Composite, Rest, Wsdl, File, Netezza, MsExcel(POI), Ldap, MySql, XmlFile, Jdbc, DB2, etc.
sourcecategory	VARCHAR(60)	The category of the datasource such as RELATIONAL, FILE, SERVICE, etc.
ownerid	DECIMAL(19,0)	The resource owner unique id.
owner	VARCHAR(255)	The resource owner name.
parentpath	VARCHAR(4000)	The parent path of the resource.
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.

status	VARCHAR(20)	The status of the datasource [UP, DOWN, DISABLED].
statuschangeflag	DECIMAL(19,0)	The status change flag indicates if the datasource changed from the previous time. NULL if first entry. 0=no change, 1=changed since last record.
connectionchangeflag	DECIMAL(19,0)	The connection change flag indicates if the datasource connection information changed since the previous time. NULL if first entry. 0=no change. 1=changed since last record.
connectionchangelist	VARCHAR(255)	The connection change list provides information on what changed. STATUS_CHANGE - The status of the datasource has changed such as UP to DOWN. CONN_HOST_NAME - The connection host name changed since the last record. CONN_PORT - The connection port changed since the last record. CONN_LOGIN_NAME - The connection login name changed since the last record. CONN_LOGIN_VALUE - The connection login value changed since the last record. CONN_DATABASE_NAME - The connection database name changed since the last record. CONN_DATABASE_TYPE - The connection dataase type changed since the last record. CONN_URL_OR_ROOT - The connection URL or root changed since the last record. CONN_CACHE_STATUS - The connection cache status resource path changed since the last record. CONN_CACHE_TRACKING - The connection cache tracking resource path changed since the last record. CONN_CREATED_BY - The connection created by name changed since the last record. CONN_CREATED_DATE - The connection created date changed since the last record. CONN_MODIFIED_DATE - The connection modified date changed since the last record. CONN_MODIFIED_BY - The connection modified by name changed since the last record.
numrequests	DECIMAL(19,0)	The numer of requests that have been executed during the datasource lifetime.
activerequests	DECIMAL(19,0)	The number of active requests being executed at this point in time (eventtime).
maxconn	DECIMAL(19,0)	The maximum configured connections on the advanced tab.
numcurrentconn	DECIMAL(19,0)	The number of current connections at this point in time (eventtime).
numinuseconn	DECIMAL(19,0)	The number of “in use” connections at this point in time (eventtime).
numlogins	DECIMAL(19,0)	The number of logins during the datasource lifetime.
numlogouts	DECIMAL(19,0)	The number of logouts during the datasource lifetime.
bytesto	DECIMAL(19,0)	The number of estimated bytes sent into the datasource.
bytesfrom	DECIMAL(19,0)	The numer of estimated bytes sent from the datasource.
message	VARCHAR(4000)	The exception message if a request was not successful.
connhostname	VARCHAR(4000)	The connection host name or ip address.
connport	INTEGER	The connection port.
connloginname	VARCHAR(255)	The connection login name.
connloginvalue	VARCHAR(255)	The connection login value.
conndatabasename	VARCHAR(255)	The connection database name.
conndatabasetype	VARCHAR(255)	The connection database type such as Oracle, PostgresSQL, SqlServer, Composite, Rest, Wsdl, File, Netezza, MsExcel(POI), Ldap, MySql, XmlFile, Jdbc, DB2, etc.
connurlorroot	VARCHAR(4000)	The connection URL or root. URL if relational. Root if file.
conncachestatus	VARCHAR(4000)	The connection cache status table path or null if not applicable.
conncachetracking	VARCHAR(4000)	The connection cache tracking table path or null if not applicable.
conncreatedby	VARCHAR(255)	The connection created by user name.
conncreateddate	TIMESTAMP	The connection created date (timestamp).
connmodifiedby	VARCHAR(255)	The connection modified by user name.
connmodifieddate	TIMESTAMP	The connection modified date (timestamp).

nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vLogDisk Table

This table returns DV server disk events derived from the cached METRICS_LOG_DISK which was cached from /services/databases/system/LOG_DISK. The information includes configured disk size/used, temporary disk size/used and log disk size/used.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
eventtime	TIMESTAMP	The timestamp of when the row was recorded in system.LOG_DISK.
confdisksize	DECIMAL(19,0)	The configured disk size on the DV server.
confdiskused	DECIMAL(19,0)	The configured disk used on the DV server.
conffreepercent	DECIMAL(19,3)	The configured free percent on the DV server.
tmpdisksize	DECIMAL(19,0)	The temporary disk size on the DV server.
tmpdiskused	DECIMAL(19,0)	The temporary disk used on the DV server.
tmpfreepercent	DECIMAL(19,3)	The temporary free percent on the DV server.
logdisksize	DECIMAL(19,0)	The log disk size on the DV server.
logdiskused	DECIMAL(19,0)	The log disk used on the DV server.
logfreepercent	DECIMAL(19,3)	The log free percent on the DV server.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vLogIO Table

This table returns DV server disk events derived from the cached METRICS_LOG_IO which was cached from /services/databases/system/LOG_IO. The information includes bytes from clients, bytes to clients, bytes from data sources and bytes to data sources.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
eventtime	TIMESTAMP	The timestamp of when the row was recorded in system.LOG_IO.
fromclients	DECIMAL(19,0)	The number of bytes coming from clients into DV over the IO channel.
toclients	DECIMAL(19,0)	The number of bytes going to clients from DV over the IO channel.
fromdatasources	DECIMAL(19,0)	The number of bytes coming from datasources to DV over the IO channel.
todatasources	DECIMAL(19,0)	The number of bytes going to datasources from DV over the IO channel.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vLogMemory Table

This table returns DV server disk events derived from the cached METRICS_LOG_MEMORY which was cached from /services/databases/system/LOG_MEMORY. The information memory bytes, memory max, managed bytes and managed max which can also be found on the Studio tab and memory tab for a graphical display.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
eventtime	TIMESTAMP	The timestamp of when the row was recorded in system.LOG_MEMORY.
memorybytes	DECIMAL(19,0)	The amount of memory in bytes currently being used.
memorymax	DECIMAL(19,0)	The maximum memory in bytes configured in DV.
managedbytes	DECIMAL(19,0)	The managed memory bytes currently in DV.
managedmax	DECIMAL(19,0)	The managed maximum memory in bytes configured in DV.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

VSySCluster Table

This table returns DV SYS_CLUSTER information for all nodes in the cluster. It has no rows for single-node non-cluster tables. It is not partitioned.

Column Name	Column Type	Definition
loadtime	TIMESTAMP	The timestamp of when the row was recorded in system.
nodehost	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
iscluster	VARCHAR(1)	The value 'Y' to indicate that this is a cluster.
displayName	VARCHAR(255)	The name of the cluster.
serverid	VARCHAR(255)	The TDV server id.
servername	VARCHAR(255)	The TDV server name.
serverhost	VARCHAR(255)	The member hostname of a node in the cluster.
serverport	DECIMAL(9,0)	The member port of a node in the cluster.
status	VARCHAR(255)	The status of the cluster node. [OPERATIONAL, DISCONNECTED, CONNECTED READY]
timekeeper	VARCHAR(1)	The value indicates whether this node is the timekeeper or not. Y or N
connecttime	TIMESTAMP	The timsteamp of when the node connected to the cluster.
lastcontact	TIMESTAMP	The timestamp of when the node was last contacted.
numcontancts	DECIMAL(24,0)	The number of contacts that have been made.
bytesto	DECIMAL(24,0)	The number of bytes going into the node.
bytesfrom	DECIMAL(24,0)	The number of bytes going out of the node.
message	CLOB	A message of what has happened with this node.

vSysNodes Table

This table returns DV nodes and the node suffix which is used to create and manage the METRICS_ALL_RESOURCES_N##_T01/02 and METRICS_ALL_USERS_N##_T01/02 tables. Each node in the cluster is assigned 2 cache tables to insert data into in a round-robin basis when their trigger is executed. This allows each node to individually manage their ALL_RESOURCES and ALL_USERS more efficiently in that a table of data is always available for use and they can truncate their table when the alternate table is fully cached. This is a similar concept to TDV caching but this concept allows the tables to run on all nodes at the same time to capture that nodes metadata. This table is not partitioned.

Column Name	Column Type	Definition
loadtime	TIMESTAMP	The timestamp of when the row was recorded in system.LOG_MEMORY.
nodehost	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
nodesuffix	VARCHAR(255)	The node suffix [N01] indicates the node number that is assigned to this particular node which will be used to insert ALL_RESOURCES and ALL_USERS into its corresponding set of tables.

vSystemResources Table

Details on system resource usage. Derived from METRICS_CIS_SYSTEM_RESOURCES. This table provides a correlation between vLogMemory, vLogDisk, vLogIO and vCpuMemUtilization.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
eventtime	TIMESTAMP	The timestamp of when the row was recorded in the metrics tables.
avgmembytes	DECIMAL(19,0)	The average amount of memory in bytes currently being used.
avgmemmax	DECIMAL(19,0)	The average maximum memory in bytes configured in DV.
avgmanagedmembytes	DECIMAL(19,0)	The average managed memory bytes currently in DV.
avgmanagedmemmax	DECIMAL(19,0)	The average managed maximum memory in bytes configured in DV.
iofromclients	DECIMAL(19,0)	The number of bytes coming from clients into DV over the IO channel.
iotoclients	DECIMAL(19,0)	The number of bytes going to clients from DV over the IO channel.
iofromdatasources	DECIMAL(19,0)	The number of bytes coming from datasources to DV over the IO channel.
iotodatasources	DECIMAL(19,0)	The number of bytes going to datasources from DV over the IO channel.
confdisksize	DECIMAL(19,0)	The configured disk size on the DV server.
confdiskused	DECIMAL(19,0)	The configured disk used on the DV server.
tmpdisksize	DECIMAL(19,0)	The temporary disk size on the DV server.
tmpdiskused	DECIMAL(19,0)	The temporary disk used on the DV server.
logdisksize	DECIMAL(19,0)	The log disk size on the DV server.
logdiskused	DECIMAL(19,0)	The log disk used on the DV server.
cpuutilization	DECIMAL(24,4)	The system CPU utilization (used) percent.
sysavailablememory	DECIMAL(24,4)	The system memory used in MB. This will always be showing high since DV takes the configured amount of memory at startup time. This is only a good guage of potential other processes running on the same server as DV if a large fluctuation is detected.
sysusedmemory	DECIMAL(24,4)	The system memory available in MB. Counterpart to the above statement.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

users Definitions

vAllUsers Table

Details on all current users derived from METRICS_ALL_USERS. It is loaded fully each time based on the trigger. The old copy is deleted.

Column Name	Column Type	Definition
loaddate	TIMESTAMP	The timestamp of when the record was inserted or loaded.

userid	DECIMAL(19,0)	The DV userid from the system.ALL_USERS tables.
user	VARCHAR(255)	The user name who connected for this session.
domainid	DECIMAL(19,0)	The unique domain id.
domain	VARCHAR(255)	The domain of the user who connected.
annotation	CLOB	The annotation of the user.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

workflow Definitions

vCISWorkflow Table

Report to view the status of the KPImetrics triggers derived from METRICS_CIS_WORKFLOW.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
workflowid [k]	DECIMAL(19,0)	The unique id for this record.
workflowname	VARCHAR(255)	The workflow name.
workflowstatus	VARCHAR(10)	The workflow status [S=success, F=fail, I=in progress].
workflowstarttime	TIMESTAMP	The workflow start time.
workflowendtime	TIMESTAMP	The workflow end time.
resourcename	VARCHAR(255)	The resource name that started the workflow.
message	VARCHAR(4000)	An informational message about the workflow status.
errormessage	CLOB	The exception message if a request was not successful
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
workflowstarttimestr	VARCHAR(30)	The workflow start time in a string.

vCISWorkflowStatus Table

Report to view the status of the KPImetrics triggers sort by workflowstatus ASC and derived from METRICS_CIS_WORKFLOW. Failure message come first. Workflowstatus=F, I, S. F=Fail, I=In process, S=Success. This view has an ORDER BY of workflowstatus ASC.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
workflowid [k]	DECIMAL(19,0)	The unique id for this record.
workflowname	VARCHAR(255)	The workflow name.
workflowstatus	VARCHAR(10)	The workflow status [S=success, F=fail, I=in progress].
workflowstarttime	TIMESTAMP	The workflow start time.
workflowendtime	TIMESTAMP	The workflow end time.
resourcename	VARCHAR(255)	The resource name that started the workflow.
message	VARCHAR(4000)	An informational message about the workflow status.
errormessage	CLOB	The exception message if a request was not successful
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').

nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
workflowstarttimestr	VARCHAR(30)	The workflow start time in a string.

vEventRegistration Table

Report to view the list of email event registration subscriptions that are currently configured. Derived from METRICS_EVENT_REGISTRATION.

Column Name	Column Type	Definition
subscriberemail	VARCHAR(255)	The email of the person subscribing to this event.
groupname	VARCHAR(255)	The DV or LDAP group the person belongs to.
environmenttype	VARCHAR(255)	The environment name from commonValues.cisServerNickname or ALL for all environments.
envtype	VARCHAR(255)	The event type [LONG_RUNNING, EXCEEDED_MEMORY, INACTIVITY, WORKFLOW_FAILURE, DBMS_SCHEDULER_ERROR, PURGE_HISTORY]
requesteremail	VARCHAR(255)	The emails of the person requesting this event.
requesterfirstname	VARCHAR(255)	The first name of the person requesting this event.
requesterlastname	VARCHAR(255)	The last name of the person requesting this event.
Excludetext	VARCHAR(4000)	A comma separate list of text that when found will signal exclusion and the email will not be sent. This is a way of filtering out emails based on text.

vJobDeleteCheck Table

Report to view the status of the METRICS_JOB_DELETE_CHECK DBMS table. This table provides information about TDV metrics delete statement activity when trigger "kpimetricsTrig_35_DBMSDeleteCheck" is turned on. It will reveal when the delete statement was executed for each node and how many rows were deleted from the collection tables.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
loaddate	TIMESTAMP	The timestamp of when the data was loaded.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
status	VARCHAR(20)	The status of the request (delete statement). E.g. SUCCESS, FAILED, TERMINATED
requesttype	VARCHAR(20)	The request type is SQL.
sqlcommand	VARCHAR(20)	The SQL command is DELETE.
resourcename	VARCHAR(255)	The metrics collection table resource name being deleted. E.g. metrics_requests, metrics_resources_usage or metrics_sessions.
description	CLOB	The SQL delete statement.
starttime	TIMESTAMP	The timestamp when the delete statement started.
endtime	TIMESTAMP	The timestamp of when the delete statement ended.
totalduration	INTEGER	The total duration of the delete statement in milliseconds.
rowsaffected	INTEGER	The number of rows deleted.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vJobDetails Table

Report to view the status of the METRICS_JOB_DETAILS DBMS Scheduler table. Provides a sorted status by most recent rows first. The information informs the user of the status of the data transfer from the collection tables to the history tables for each of the 3 metrics tables.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
reqstartdate	TIMESTAMP	The timestamp of when the request is started.
reqenddate	TIMESTAMP	The timestamp of when the request ends.
jobtablename	VARCHAR(40)	The job table name which identies the type of processing that is taking place. DBMS_SCHEDULER – An overall record that indicates processing for the metrics collection and history tables. metrics_sessions – process rows. metrics_resources_usage – process rows. metrics_requests – process rows. metrics_sessions_hist – process rows only if commonValues.partitionNumber=0. metrics_resources_usage_hist – process rows only if commonValues.partitionNumber=0. metrics_requests_hist – process rows only if commonValues.partitionNumber=0. DELETE_COLLECTION – An overall record that indicates processing for the metrics collection tables. metrics_resources_usage – delete rows. metrics_requests – delete rows. REBUILD_INDEXES – An overall record that indicates the various metrics tables indexes are being rebuilt or reorganized. This usually happens once a week. PARTITION_MANAGEMENT_ADD – An overall record indicating the status of the partition add command. PARTITION_MANAGEMENT_DROP – An overall record indicating the status of the partition drop command.
correlationid	DECIMAL(19,0)	A generated unique sequence that correlates different records together in a batch.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
processednodehost	VARCHAR(255)	The hostname (nodehost) of the DV server that is performing the processing. Due to locking issues [mainly] with SQL Server, the architecture dictates that only one node performs the processing for all the other nodes.
processednodeport	INTEGER	The port (nodeport) of the DV server that is performing the processing.
processedstartdate	TIMESTAMP	The timestamp of when the DV process started performing the processing. This timestamp will remain the same across all rows with the same correlationid.
insertnumrows	DECIMAL(19,0)	For DBMS_SCHEDULER, the number of rows inserted.
deletenumrows	DECIMAL(19,0)	For DBMS_SCHEDULER and DELETE_COLLECTION, the number of rows deleted.
updatenumrows	DECIMAL(19,0)	For DBMS_SCHEDULER, the number of rows updated.
notinsertednumrows	DECIMAL(19,0)	For DBMS_SCHEDULER, the number of rows not inserted.
maxstarttime	TIMESTAMP	The maximum starttime [metrics_requests, metrics_resources_usage] or logintime [metrics_sessions] found in the specified table being processed.
maxid	DECIMAL(19,0)	The maximum requestid [metrics_requests, metrics_resources_usage] or sessionid [metrics_sessions] found in the specified table being processed.
status	VARCHAR(100)	The job details status [COMPLETE=the table record, NO RECORDS FOUND=table record, SUCCESS=the overall record, FAILURE=the overall record.]

		Intermediate status for each stage in the workflow process: DELETE COMPLETE <table name> TABLE STATISTICS COMPLETE <table name> DELETE STAGE COMPLETE INSERT STAGE COMPLETE <table name> UPDATE COMPLETE INSERT HISTORY COMPLETE
currentoperation	VARCHAR(150)	The current operation being executed.
insertduration	VARCHAR(30)	The duration of the insert.
deleteduration	VARCHAR(30)	The duration of the delete.
updateduration	VARCHAR(30)	The duration of the update.
totalduration	VARCHAR(30)	The total duration of the request measured in milli-seconds. Divide by 1000 to get minutes.
additionalinfo	VARCHAR(4000)	The additional info provide informational message. For example, if batching is being used it might look like this: [Total resourcekind-user defined Update Batches(50000)=1] [Total resourcekind-system Update Batches(50000)=0] [Total Insert Batches(50000)=1] [Total Delete Batches(50000)=1]
errormessage	CLOB	The exception message if a request was not successful

vJobDetailsReport Table

Report to summarize the results of METRICS_JOB_DETAILS grouped by the job table name, node host and node port. It provides various averages, min and max of duration and number of rows. It provides summation of rows inserted, updated, deleted and not inserted.

Column Name	Column Type	Definition
jobtablename	VARCHAR(40)	The job table name which identies the type of processing that is taking place. DBMS_SCHEDULER – An overall record that indicates processing for the metrics collection and history tables. metrics_sessions – process rows. metrics_resources_usage – process rows. metrics_requests – process rows. metrics_sessions_hist – process rows only if commonValues.partitionNumber=0. metrics_resources_usage_hist – process rows only if commonValues.partitionNumber=0. metrics_requests_hist – process rows only if commonValues.partitionNumber=0. DELETE_COLLECTION – An overall record that indicates processing for the metrics collection tables. metrics_resources_usage – delete rows. metrics_requests – delete rows. REBUILD_INDEXES – An overall record that indicates the various metrics tables indexes are being rebuilt or reorganized. This usually happens once a week. PARTITION_MANAGEMENT_ADD – An overall record indicating the status of the partition add command. PARTITION_MANAGEMENT_DROP – An overall record indicating the status of the partition drop command.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

insert_avg	INTERVAL DAY TO SECOND	The average insert duration.
delete_avg	INTERVAL DAY TO SECOND	The average delete duration.
update_avg	INTERVAL DAY TO SECOND	The average update duration.
total_avg	INTERVAL DAY TO SECOND	The average total time.
total_min	INTERVAL DAY TO SECOND	The minimum total minutes across all operatons.
total_max	INTERVAL DAY TO SECOND	The maximum total minutes across all operatons.
insertrows_avg	DECIMAL(19,0)	The average number of rows inserted.
deleterows_avg	DECIMAL(19,0)	The average number of rows deleted.
updaterows_avg	DECIMAL(19,0)	The average number of rows updated.
notinsertrows_avg	DECIMAL(19,0)	The average number of rows not inserted.
insertrows_sum	DECIMAL(25,0)	The total number of rows inserted.
deleterows_sum	DECIMAL(25,0)	The total number of rows deleted.
updaterows_sum	DECIMAL(25,0)	The total number of rows updated.
notinsertrows_sum	DECIMAL(25,0)	The total number of rows not inserted.

vJobDetailsStatusSummary Table

Column Name	Column Type	Definition
jobtablename	VARCHAR(40)	The job table name which identies the type of processing that is taking place. DBMS_SCHEDULER – An overall record that indicates processing for the metrics collection and history tables. metrics_sessions – process rows. metrics_resources_usage – process rows. metrics_requests – process rows. metrics_sessions_hist – process rows only if commonValues.partitionNumber=0. metrics_resources_usage_hist – process rows only if commonValues.partitionNumber=0. metrics_requests_hist – process rows only if commonValues.partitionNumber=0. DELETE_COLLECTION – An overall record that indicates processing for the metrics collection tables. metrics_resources_usage – delete rows. metrics_requests – delete rows. REBUILD_INDEXES – An overall record that indicates the various metrics tables indexes are being rebuilt or reorganized. This usually happens once a week. PARTITION_MANAGEMENT_ADD – An overall record indicating the status of the partition add command. PARTITION_MANAGEMENT_DROP – An overall record indicating the status of the partition drop command.
status	VARCHAR(100)	The job details status [COMPLETE=the table record, NO RECORDS FOUND=table record, SUCCESS=the overall record, FAILURE=the overall record.] Intermediate status for each stage in the workflow process: DELETE COMPLETE <table name> TABLE STATISTICS COMPLETE <table name>

		DELETE STAGE COMPLETE INSERT STAGE COMPLETE <table name> UPDATE COMPLETE INSERT HISTORY COMPLETE
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
cnt	BIGINT	The count of records grouped by the columns in this view.

vJobEnvironments Table

Report to view the list of valid environments derived from METRICS_JOB_ENVIRONMENTS

Column Name	Column Type	Definition
envtype	VARCHAR(255)	The environment type describes the type of environment such as DEV, UAT, TEST, PROD.

vJobFilters Table

Report to view the list of job filters that are currently configured derived METRICS_JOB_FILTERS.

Column Name	Column Type	Definition
envtype	VARCHAR(255)	The environment type describes the type of environment such as DEV, UAT, TEST, PROD.
tablename	VARCHAR(255)	Currently always set to metrics_resources_usage.
user	VARCHAR(255)	The DV user to be filtered out.
domain	VARCHAR(255)	The DV user domain that the user belongs to.
resourcekind	VARCHAR(20)	The type of resource that executed [system, user defined].
resourcepath [k]	VARCHAR(4000)	The DV path to the resource.
resourcetype [k]	VARCHAR(40)	The DV resource type. If published, then it will be LINK.
pathtype	VARCHAR(10)	The pathtype is either FULL=full DV path or PARTIAL=partial wild card path (resourcepath must contain a % at the end of the path indicating a wild card)

vSqlControl Table

Derived from METRICS_SQL_CONTROL. Normally has no rows unless Cache_METRICS_SQL_REQUEST_EXEC is executing. If it contains rows then they indicate that a particular node is performing the pre-processing for the METRICS_SQL_REQUEST table. Only only one node at a time can perform this processing due to the fact that any single node can process another nodes SQL requests in order to achieve parallel processing. This table acts as the control table to allow each node to select their requestids that they will process. This is why this part of the process is single-threaded and must be controlled.

Column Name	Column Type	Definition
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.

vSqlControlLog Table

Derived from METRICS_SQL_CONTROL_LOG. Contains a log of the Gatekeeper code block within Cache_METRICS_SQL_REQUEST_EXEC when debugGatekeeper = '1'. During normal operation, this will be turned off so that no rows are produced. It is only useful for debugging purposes to insure that in a clustered environment, the nodes are taking their turn initializing their own set of rows when doing parallel processing on one of the node's data.

Column Name	Column Type	Definition
partition [k]	SMALLINT	The partition number is based on the day of the year. There can be 366 partitions which accounts for leap year. This interval partition will act like a sliding window of 1 year.
controllogid	DECIMAL(19,0)	A unique generated sequence.
starttime [k]	TIMESTAMP	The timestamp of when the request was started. E.g. 2020-01-06 10:19:36.867
codeblock	VARCHAR(255)	The block of code being processed to be used for debug if needed. 1.00 Gatekeeper:[Rows exist to be processed.] Count= 1.01 Gatekeeper:[Detect processing. Count= 1.02 Gatekeeper:[Successfully inserted control row.] 1.03 Gatekeeper:[Waiting for access.] 1.04 Gatekeeper:[Construct the SQL statement.] 1.05 Gatekeeper:[Perform pre-populate (insert rows).] 1.06 Gatekeeper:[Replace ASCII Characaters > 255]: 1.07 Gatekeeper:[Insert successful. Row= 1.08 Gatekeeper:[No insert was performed.] performInsert= 1.09 Gatekeeper:[No insert was performed.] performInsert= 1.10 Gatekeeper:[Total insert duration= 1.11 Gatekeeper:[No rows to process.] 1.12 Gatekeeper:[Delete control row] 1.13 Gatekeeper:[Pre-processing timeout: Waiting for another node to finish.]
trynum	DECIMAL(9,0)	The number of tries.
processtime	TIMESTAMP	The timestamp of when the processing started.
requestid [k]	DECIMAL(19,0)	The unique request id that is the primary key of this table.
nodehost [k]	VARCHAR(255)	The host name which comes from GetProperty('SERVER_HOSTNAME').
nodeport [k]	INTEGER	The Http port (e.g. 9400) which comes from GetProperty('SERVER_JDBC_PORT')-1.
processednodehost	VARCHAR(255)	The hostname (nodehost) of the DV server that is performing the processing. Due to locking issues [mainly] with SQL Server, the architecture dictates that only one node performs the processing for all the other nodes.
processednodeport	DECIMAL(9,0)	The port (nodeport) of the DV server that is performing the processing.

4 Release Notes

This section provides information on what has changed in this release.

Added or Modified in this Release
This section provides bullet points on what has been added or changed in this release.

Release 2022Q100 [Feb 10 2022]

- Added metrics_all_kpimetrics_table_counts and metrics_requests_hist_sqlplan.
- Refactored the published metrics schema into metrics_collection, metrics_count and metrics_history to make it easier to find view.

Release 2020Q300 [Aug 17 2020]

- Modified notificationstatus to vEventRegLog.

Release 2020Q203 [Jun 24 2020]

- Implemented daily (interval) Table Partitioning scheme for all tables. Changed monthly to daily for metrics history tables.

Release 2020Q202 [May 1 2020]

- Removed all [Arch] tables as partitioning for each table has been added so that the current and historical data will reside in the same table in an interval of 366 days.

Release 2020Q201 [Apr 6 2020]

- Added metadata. reportMetadataAllCount[Arch]
- Added metrics_requests_groupby_date, metrics_requests_groupby_nodehost_nodeport
- Added metrics_resources_usage_groupby_date, metrics_resources_usage_groupby_nodehost_nodeport
- Added metrics_sessions_groupby_date, metrics_sessions_groupby_nodehost_nodeport

Release 2020Q101 [Feb 18 2020]

- Added metadata.reportMetadataPrivilegeUsers[Arch]
- Added metadata.vMetadataAllPrivileges
- Added metadata.vMetadataAllResources
- Added metadata.vMetadataAllUsersGroups[Arch]
- Added metadata.vMetadataPrivilegeUser[Arch]
- Removed vAllResourcesMax
- Renamed vAllResourcesHist to vAllResources
    - Added resourcelinktype, ownerdomain, annotation

    - Renamed owner to ownername, loadtime to loaddate
    - Modified types for guid, resourcetype, resourcesubtype
- Removed vAllUsersMax
- Renamed vAllUsersHist to vAllUsers
- Change all references of loadtime to loaddate.
- vMetadataPrivilege[Arch]
    - Dropped username column
    - Added privilegeid, layertype, nameid

Release 2020Q100 [Jan 14 2020]

- Initial release of the KPImetrics Data Dictionary