SAP HANA System Tables and Monitoring Views Reference

SAP HANA Appliance Software SPS 04

Target Audience

- Consultants
- Administrators
- SAP Hardware Partner
- Others

Copyright

in Motion Limited.

Google App Engine, Google Apps, Google Checkout, Google Data API, Google Maps, Google Mobile Ads, Google Mobile Updater, Google Mobile, Google Store, Google Sync, Google Updater, Google Voice, Google Mail, Gmail, YouTube, Dalvik and Android are trademarks or registered trademarks of Google Inc.

INTERMEC is a registered trademark of Intermec Technologies Corporation.

Wi-Fi is a registered trademark of Wi-Fi Alliance.

Bluetooth is a registered trademark of Bluetooth SIG Inc.

Motorola is a registered trademark of Motorola Trademark Holdings LLC.

Computop is a registered trademark of Computop Wirtschaftsinformatik GmbH.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, SAP HANA, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.

Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase Inc. Sybase is an SAP company.

Crossgate, m@gic EDDY, B2B 360°, and B2B 360° Services are registered trademarks of Crossgate AG in Germany and other countries. Crossgate is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

Table of Content

SAP HANA System Tables and Monitoring Views Reference

About this Document

This guide describes the system tables and monitoring views available in the SAP HANA database.

System tables and monitoring views allow you to query for various information about the system state using SQL commands. The results appear as tables.

Each table contains the following columns:

- **Column name:** Contains the name of the table column name
- **Data type:** Describes what the data type is (integer, character, boolean and so on). For a full explanation of data types see the SAP HANA SQL Reference.
- **Unit:** If there is a relevant measurement unit for the column (millisecond, byte and so on) it is displayed here
- **Description:** Describes the column

The tables are listed in alphabetical order. System tables can be identified as they have no prefix whereas the monitoring views are prefixed with **M_**. Some other abbreviations are used in the table names and they are explained here:

- **CS** Column Store
- **RS** Row store
- **CE** Calculation Engine
- **XS** XSEngine

SAP HANA Guides

Latest documentation for SAP HANA:

https://service.sap.com/hana*
http://help.sap.com/hana_appliance

AUDIT_POLICIES

Short description

Defined audit policies

Structure

Column name	Data type	Unit	Description
AUDIT_POLICY_NAME	NVARCHAR(256)		Audit policy name
AUDIT_POLICY_OID	BIGINT		Object ID of the audit policy
EVENT_STATUS	VARCHAR(5000)		Status of events to be audited: 'SUCCESSFUL EVENTS', 'UNSUCCESSFUL EVENTS', 'ALL EVENTS'
EVENT_LEVEL	VARCHAR(5000)		Level of events to be audited: 'EMERGENCY', 'CRITICAL', 'ALERT', 'WARNING', 'INFO'
IS_AUDIT_POLICY_ACTIVE	VARCHAR(5000)		Audit policy is active: 'TRUE', 'FALSE'
EVENT_ACTION	VARCHAR(5000)		Action to be audited, for example: 'SELECT', 'GRANT PRIVILEGE', ...
USER_NAME	NVARCHAR(5000)		User whose actions are to be audited
OBJECT_TYPE	VARCHAR(5000)		Type of object to be audited
OBJECT_SCHEMA	VARCHAR(5000)		Schema of object to be audited
OBJECT_NAME	VARCHAR(5000)		Name of object to be audited

AUTHORIZATION_GRAPH

Short description

Represents authorization dependencies of complex database objects

Full description

The WHERE clause is mandatory when you SELECT from AUTHORIZATION_GRAPH. This view only returns data if you specify 'ROOT_DEPENDENT_TYPE_ID = ...AND ROOT_DEPENDENT_OBJECT_ID=...' as part of the where clause.

Structure

Column name	Data type	Unit	Description
DEPENDENT_TYPE_ID	INTEGER		Object type ID of the dependent object
DEPENDENT_TYPE_ID_NAME	NVARCHAR(5000)		Object type name of the dependent object
DEPENDENT_SUBTYPE_ID	SMALLINT		Object subtype ID of the dependent object
DEPENDENT_SUBTYPE_ID_NAME	NVARCHAR(256)		Object subtype name of the dependent object
DEPENDENT_OBJECT_ID	BIGINT		Object ID of the dependent object
DEPENDENT_SCHEMA_NAME	NVARCHAR(256)		Schema name the dependent object belongs to
DEPENDENT_OBJECT_NAME	NVARCHAR(256)		Object name of the dependent object
DEPENDENT_OWNER_OID	BIGINT		ID of the user owning the dependent object
DEPENDENT_OWNER_NAME	NVARCHAR(5000)		Name of the user owning the dependent object
DEPENDENT_READONLY	INTEGER		Readonly property of the dependent object
UNDERLYING_TYPE_ID	INTEGER		Object type ID of the underlying object
UNDERLYING_TYPE_ID_NAME	NVARCHAR(5000)		Object type name of the underlying object
UNDERLYING_SUBTYPE_ID	SMALLINT		Object subtype ID of the underlying object
UNDERLYING_SUBTYPE_ID_NAME			
	NVARCHAR(256)		Object subtype name of the underlying object
UNDERLYING_OBJECT_ID	BIGINT		Object ID of the underlying object
UNDERLYING_SCHEMA_NAME	NVARCHAR(256)		Schema name the underlying object belongs to
UNDERLYING_OBJECT_NAME	NVARCHAR(256)		Object name of the underlying object
UNDERLYING_OWNER_OID	BIGINT		ID of the user owning the underlying object
UNDERLYING_OWNER_NAME	NVARCHAR(5000)		Name of the user owning the underlying object
UNDERLYING_READONLY	INTEGER		Readonly property of the underlying object
DEPENDENCY_USER_OID	BIGINT		ID of the user required to have a certain privilege on the underlying object in order to validate the dependent object
DEPENDENCY_USER_NAME	NVARCHAR(256)		Name of the user required to have a certain privilege on the underlying object in order to validate the dependent object

DEPENDENCY_TYPE	INTEGER		Type of dependency determines the validation semantics for dependencies sharing the same dependent objects
PRIVILEGE_ID	BIGINT		ID of the privilege the dependency user is required to have on the underlying object
PRIVILEGE_NAME	NVARCHAR(256)		Name of the privilege the dependency user is required to have on the underlying object
IS_VALID	TINYINT		Current state of the dependency. It is set by the object validation logic and reflects missing privileges or invalidated objects

CONSTRAINTS

Short description

Constraints defined for tables

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
TABLE_NAME	NVARCHAR(256)		Table name
COLUMN_NAME	NVARCHAR(256)		Column name
POSITION	SMALLINT		Column number in table definition
CONSTRAINT_NAME	NVARCHAR(256)		Constraint name
IS_PRIMARY_KEY	VARCHAR(5000)		Constraint is primary key constraint: 'TRUE', 'FALSE'
IS_UNIQUE_KEY	VARCHAR(5000)		Constraint is unique constraint: 'TRUE', 'FALSE'

CS_BO_VIEWS

Short description

Business object views for column store join views

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
VIEW_NAME	NVARCHAR(256)		Join view name
BO_VIEW_NAME	NVARCHAR(256)		Business object view name
ANCHOR_SCHEMA_NAME	NVARCHAR(256)		Anchor schema name
ANCHOR_TABLE_NAME	NVARCHAR(256)		Anchor table name

CS_FREESTYLE_COLUMNS

Short description

Freestyle search columns for column store join views

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
VIEW_NAME	NVARCHAR(256)		Join view name
COLUMN_NAME	NVARCHAR(256)		Freestyle column name

CS_JOIN_CONDITIONS

Short description

Join conditions defined for column store join views

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
VIEW_NAME	NVARCHAR(256)		Join view name
JOIN_CONDITION_NAME	NVARCHAR(256)		Join condition name
JOIN_ORDER	BIGINT		Join order number
TABLE_SCHEMA_NAME1	NVARCHAR(256)		Schema name of column 1
TABLE_NAME1	NVARCHAR(256)		Table name of column 1
COLUMN_NAME1	NVARCHAR(256)		Name of column 1
TABLE_SCHEMA_NAME2	NVARCHAR(256)		Schema name of column 2
TABLE_NAME2	NVARCHAR(256)		Table name of column 2
COLUMN_NAME2	NVARCHAR(256)		Name of column 2
CONSTRAINTS	NVARCHAR(256)		Join constraint name
JOIN_TYPE	VARCHAR(5000)		Join type: 'inner', 'left', 'right', 'full'
CARDINALITY	VARCHAR(5000)		Join cardinality: '1:1', '1:n', 'n:1', 'n:n'

CS_JOIN_CONSTRAINTS

Short description

Join constraints defined for column store join views

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
VIEW_NAME	NVARCHAR(256)		Join view name
JOIN_CONSTRAINT_NAME	NVARCHAR(256)		Join constraint name
CONSTRAINT_TYPE	VARCHAR(5000)		Type of constraint owner
LOCATION	NVARCHAR(256)		Name of constraint owner
TABLE_SCHEMA_NAME	NVARCHAR(256)		Schema name of table
TABLE_NAME	NVARCHAR(256)		Table name of column
COLUMN_NAME	NVARCHAR(256)		Column name
OPERATOR	NVARCHAR(256)		Operator
VALUE	NVARCHAR(5000)		Value

CS_JOIN_PATHS

Short description

Join paths defined for column store join views

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
VIEW_NAME	NVARCHAR(256)		Join view name
JOIN_PATH_NAME	NVARCHAR(256)		Join path name
JOIN_CONDITIONS	NVARCHAR(5000)		Comma separated list of join conditions
JOIN_CONSTRAINTS	NVARCHAR(5000)		Comma separated list of join constraints

CS_JOIN_TABLES

Short description

Physical tables referred by column store join views

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
VIEW_NAME	NVARCHAR(256)		Join view name
TABLE_SCHEMA_NAME	NVARCHAR(256)		Schema name of physical table name
TABLE_NAME	NVARCHAR(256)		Physical table name
TABLE_TYPE	VARCHAR(5000)		Type of table

CS_KEY_FIGURES

Short description

Key figures defined for column store join views

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
VIEW_NAME	NVARCHAR(256)		Join view name
KEY_FIGURE_NAME	NVARCHAR(256)		Key figure name
DEFAULT_AGGREGATION_TYPE	NVARCHAR(5000)		Aggregation type like: 'sum', 'count', 'min', ...
DESCRIPTION	NVARCHAR(5000)		Description
UNIT_CONVERSION_NAME	NVARCHAR(256)		Name of unit conversion
TABLE_SCHEMA_NAME	NVARCHAR(256)		Schema name of table
TABLE_NAME	NVARCHAR(256)		Table name of column
COLUMN_NAME	NVARCHAR(256)		Column name
EXPRESSION_FLAGS	VARCHAR(5000)		Expression flags
EXPRESSION	NVARCHAR(256)		Expression

CS_VIEW_COLUMNS

Short description

Columns defined for column store join views

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
VIEW_NAME	NVARCHAR(256)		Join view name
VIEW_COLUMN_NAME	NVARCHAR(256)		View column name
TABLE_SCHEMA_NAME	NVARCHAR(256)		Schema name of table
TABLE_NAME	NVARCHAR(256)		Table name of column
COLUMN_NAME	NVARCHAR(256)		Column name
JOIN_PATH	NVARCHAR(256)		Join path name
CS_TYPE	VARCHAR(5000)		Column store data type
EXPRESSION_FLAGS	VARCHAR(5000)		Expression flags
EXPRESSION	NVARCHAR(256)		Expression

DATA_TYPES

Short description

Available SQL data types

Structure

Column name	Data type	Unit	Description
TYPE_ID	SMALLINT		Data type ID
TYPE_NAME	VARCHAR(16)		Data type name
COLUMN_SIZE	INTEGER		Maximum size of the data type that this system can support
LITERAL_PREFIX	CHAR(1)		ODBC 2.0 SQLGetTypeInfo Varchar. Character or characters used to prefix a literal, for example, a single quotation mark (') for character data types or 0x for binary data types; NULL is returned for data types where a literal prefix is not applicable
LITERAL_SUFFIX	CHAR(1)		ODBC 2.0 SQLGetTypeInfo Varchar. Character or characters used to terminate a literal, for example, a single quotation mark (') for character data types; NULL is returned for data types where a literal suffix is not applicable
CREATE_PARAMS	VARCHAR(16)		ODBC 2.0 SQLGetTypeInfo Varchar. A list of keywords, separated by commas, corresponding to each parameter that the application may specify in parentheses when using the name that is returned in the TYPE_NAME field
NULLABLE	TINYINT		Specifies whether the data type can accept null or not
CASE_SENSITIVE	TINYINT		ODBC 2.0 SQLGetTypeInfo smallint. Whether a character data type is case-sensitive in collations and comparisons
SEARCHABLE	TINYINT		How the data type can be used in the WHERE clause
UNSIGNED_ATTRIBUTE	TINYINT		Signed or unsigned
FIXED_PREC_SCALE	TINYINT		Whether the data type has predefined fixed precision and scale (ODBC)
AUTO_UNIQUE_VALUE	TINYINT		Whether the data type is autoincrementing (ODBC)
LOCAL_TYPE_NAME	VARCHAR(16)		ODBC 2.0 SQLGetTypeInfo varchar. Localized version of the data source-dependent name of the data type
MINIMUM_SCALE	SMALLINT		ODBC 2.0 SQLGetTypeInfo smallint. The minimum scale of the data type on the data source
MAXIMUM_SCALE	SMALLINT		ODBC 2.0 SQLGetTypeInfo smallint. The maximum scale of the data type on the data source
SQL_DATA_TYPE	SMALLINT		ODBC 3.0 SQLGetTypeInfo smallint. The value of the SQL data type as it appears in the SQL_DESC_TYPE field of the descriptor. This column is the same as the DATATYPE column, except for interval and datetime data types
SQL_DATETIME_SUB	SMALLINT		ODBC 3.0 SQLGetTypeInfo smallint. When the value of SQL_DATATYPE is SQL_DATETIME or SQL_INTERVAL, this column contains the datetime/interval subcode
NUM_PREC_RADIX	INTEGER		ODBC 3.0 SQLGetTypeInfo integer. In case of an approximate

			numeric data type, value is 2 to indicate that COLUMN_SIZE specifies the number of bits. For exact numeric data types value 10 indicates that COLUMN_SIZE specifies the number of decimal digits
INTERVAL_PRECISION	SMALLINT		ODBC 3.0 SQLGetTypeInfo smallint. If the data type is an interval data type, then this column contains the value of the interval leading precision. Otherwise, this column is NULL

EFFECTIVE_PRIVILEGES

Short description

Privileges of the current user

Full description

The WHERE clause is mandatory when you SELECT from EFFECTIVE_PRIVILEGES. This view only returns data if you specify 'USER_NAME =...' as part of the where clause.

Structure

Column name	Data type	Unit	Description
USER_NAME	NVARCHAR(256)		Name of the user for whom effective privileges are shown
GRANTEE	NVARCHAR(256)		User or role that has the privilege
GRANTEE_TYPE	VARCHAR(5000)		'USER' or 'ROLE'
GRANTOR	NVARCHAR(256)		User or role that provided the privilege
GRANTOR_TYPE	VARCHAR(5000)		'USER' or 'ROLE'
OBJECT_TYPE	NVARCHAR(256)		Type of the granted object like: 'TABLE', 'SCHEMA', ...
SCHEMA_NAME	NVARCHAR(256)		Schema name the object belongs to
OBJECT_NAME	NVARCHAR(5000)		Object name of granted object
COLUMN_NAME	NVARCHAR(5000)		Column name
PRIVILEGE	NVARCHAR(256)		Privilege granted
IS_GRANTABLE	VARCHAR(5000)		Privilege was granted 'WITH GRANT OPTION', 'WITH ADMIN OPTION': 'TRUE', 'FALSE'
IS_VALID	VARCHAR(5000)		Privilege is valid or it became invalid because of implicit revoking: 'TRUE', 'FALSE'

EXPLAIN_PLAN_TABLE

Short description

SQL query plan explanation result

Structure

Column name	Data type	Unit	Description
STATEMENT_NAME	VARCHAR(256)		The string specified as STATEMENT_NAME on executing EXPLAIN PLAN command. This is used to distinguish plans from each other when there are multiple plans in EXPLAIN_PLAN_TABLE view
OPERATOR_NAME	VARCHAR(5000)		Name of an operator. Details are described in the following section
OPERATOR_DETAILS	NCLOB		Details of an operator. Predicates and expressions used by the operator are shown here
EXECUTION_ENGINE	VARCHAR(256)		Execution engine where the plan operator is executed: COLUMN or ROW
SCHEMA_NAME	NVARCHAR(256)		Schema name the dependent object belongs to
TABLE_NAME	NVARCHAR(256)		Name of database tables and views accessed by an operator
TABLE_TYPE	VARCHAR(5000)		Table stored type: COLUMN or ROW
TABLE_SIZE	DOUBLE		Estimated input row count of an operator. This is available only for operators accessing tables and views directly
OUTPUT_SIZE	DOUBLE		Estimated output row count of an operator
SUBTREE_COST	DOUBLE		Estimated cost of executing the subtree starting from an operator
OPERATOR_ID	INTEGER		Unique ID of an operator. IDs are integers starting from 1
PARENT_OPERATOR_ID	INTEGER		OPERATOR_ID of the parent of an operator. The shape of a query plan is a tree and the topology of the tree can be reconstructed using OPERATOR_ID and PARENT_OPERATOR_ID. PARENT_OPERATOR_ID of the root operator is shown as NULL
LEVEL	INTEGER		Level from the root operator. Level of the root operator is 1, level of a child of the root operator is 2 and so on. This can be utilized for output indentation
POSITION	INTEGER		Position in the parent operator. Position of the first child is 1, position of the second child is 2 and so on
HOST	VARCHAR(256)		Host where the plan operator is generated
PORT	INTEGER		Port where the plan operator is generated
TIMESTAMP	TIMESTAMP		Date and time when the EXPLAIN PLAN command was executed
CONNECTION_ID	INTEGER		Connection ID

FULLTEXT_INDEXES

Short description

Fulltext indexes on table columns

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
TABLE_NAME	NVARCHAR(256)		Table name
TABLE_OID	BIGINT		Object ID of the table
INDEX_NAME	NVARCHAR(256)		Name of the fulltext index
INDEX_OID	BIGINT		Object ID of the fulltext index
LANGUAGE_COLUMN	NVARCHAR(256)		Column name of the indexed table containing language indicators for the indexed documents
MIME_TYPE_COLUMN	NVARCHAR(256)		Column name of the indexed table containing mimetype indicators for the indexed documents
LANGUAGE_DETECTION	NVARCHAR(256)		Set of languages to be considered for automatic language detection
FAST_PREPROCESS	VARCHAR(5000)		If set to true, preprocessing will be done with reduced functionality (e.g. linguistic search is not available), which will speed up the indexing
FUZZY_SEARCH_INDEX	VARCHAR(5000)		If set to true, fuzzy search is performed with an additional index (faster search, but higher memory consumption)
SEARCH_ONLY	VARCHAR(5000)		If set to true, the text-attributes will not store any DAFs, i.e. it is not possible to retrieve the HTML-converted or original data from the text-attribute. If the text-attribute is a shadow attribute (i.e. created via CREATE FULLTEXT INDEX), the source-attribute (containing the original data) is not affected by this flag
IS_EXPLICIT	VARCHAR(5000)		If set to true, the index is an explicit fulltext index, i.e. created by CREATE FULLTEXT INDEX
FLUSH_AFTER_DOCUMENTS	INTEGER		Used to store change-tracking behaviour of the fulltext index
FLUSH_EVERY_MINUTES	INTEGER		Used to store change-tracking behaviour of the fulltext index
CONFIGURATION	NVARCHAR(256)		An explicit config-file can be specified, for example, to enable named entity extraction
INTERNAL_COLUMN_NAME	NVARCHAR(512)		Name of the internal text-attribute, only set if fulltext index was not created using TEXT-datatype
PHRASE_INDEX_RATIO	REAL		Float between 0.0 and 1.0 which specifies how much memory the phrase index may use (as percentage of the memory size of the fulltext index)

FUNCTIONS

Short description

Available functions

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name of the function
FUNCTION_NAME	NVARCHAR(256)		Name of the function
FUNCTION_OID	BIGINT		Object ID of the function
INPUT_PARAMETER_COUNT	INTEGER		Input type parameter count
RETURN_VALUE_COUNT	INTEGER		Return value type parameter count
IS_UNICODE	VARCHAR(5000)		Specifies whether the function contains Unicode or not: 'TRUE', 'FALSE'
DEFINITION	NCLOB		Query string of the function
FUNCTION_TYPE	VARCHAR(5000)		Type of the function

FUNCTION_PARAMETERS

Short description

Parameters of functions

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name of the function
FUNCTION_NAME	NVARCHAR(256)		Name of the function
FUNCTION_OID	BIGINT		Object ID of the function
PARAMETER_NAME	NVARCHAR(256)		Parameter name
DATA_TYPE_ID	INTEGER		Data type ID
DATA_TYPE_NAME	VARCHAR(5000)		Data type name
LENGTH	INTEGER		Parameter length
SCALE	INTEGER		Scale of the parameter
POSITION	INTEGER		Ordinal position of the parameter
TABLE_TYPE_SCHEMA	VARCHAR(5000)		Schema name of table type if DATA_TYPE_NAME is TABLE_TYPE
TABLE_TYPE_NAME	VARCHAR(5000)		Name of table type if DATA_TYPE_NAME is TABLE_TYPE
PARAMETER_TYPE	VARCHAR(5000)		Parameter mode: IN, OUT, INOUT
HAS_DEFAULT_VALUE	VARCHAR(5000)		Specifies whether the parameter has default value or not: 'TRUE', 'FALSE'
IS_NULLABLE	VARCHAR(5000)		Specifies whether the parameter accepts null value: 'TRUE', 'FALSE'

GRANTED_PRIVILEGES

Short description

Privileges granted to users and roles

Structure

Column name	Data type	Unit	Description
GRANTEE	NVARCHAR(256)		User or role the privilege is granted to
GRANTEE_TYPE	VARCHAR(5000)		'USER' or 'ROLE'
GRANTOR	NVARCHAR(256)		User who granted the privilege
OBJECT_TYPE	NVARCHAR(256)		Type of the granted object like: 'TABLE', 'SCHEMA', ...
SCHEMA_NAME	NVARCHAR(256)		Schema name the object belongs to
OBJECT_NAME	NVARCHAR(5000)		Object name of granted object
COLUMN_NAME	NVARCHAR(5000)		Column name
PRIVILEGE	NVARCHAR(256)		Privilege granted
IS_GRANTABLE	VARCHAR(5000)		Privilege was granted 'WITH GRANT OPTION', 'WITH ADMIN OPTION': 'TRUE', 'FALSE'
IS_VALID	VARCHAR(5000)		Privilege is valid or it became invalid because of implicit revoking: 'TRUE', 'FALSE'

GRANTED_ROLES

Short description

Roles granted to users or other roles

Structure

Column name	Data type	Unit	Description
GRANTEE	NVARCHAR(256)		User or role the role is granted to
GRANTEE_TYPE	VARCHAR(5000)		'USER' or 'ROLE'
ROLE_NAME	NVARCHAR(256)		Name of the granted role
GRANTOR	NVARCHAR(256)		User who granted the role
IS_GRANTABLE	VARCHAR(5000)		Role was granted 'WITH ADMIN OPTION': 'TRUE', 'FALSE'

INDEXES

Short description

Indexes on tables

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
TABLE_NAME	NVARCHAR(256)		Table name
TABLE_OID	BIGINT		Object ID of the table
INDEX_NAME	NVARCHAR(256)		Index name
INDEX_OID	BIGINT		Object ID of the index
INDEX_TYPE	VARCHAR(5000)		Type of RS-index: 'BTREE', 'BTREE_UNIQUE', 'CPBTREE', 'CPBTREE_UNIQUE'; Type of CS-index: 'INVERTED', 'INVERTED_UNIQUE', 'CONCAT_INVERTED', 'CONCAT_INVERTED_UNIQUE'
CONSTRAINT	VARCHAR(5000)		Constraint type of the index: 'UNIQUE', 'NOT_NULL_UNIQUE', 'PRIMARY_KEY'
KEY_LENGTH	SMALLINT		Length of the key
BTREE_FILL_FACTOR	SMALLINT		B-tree fill factor (50~100)
BTREE_SPLIT_TYPE	TINYINT		B-tree split type: 'SPLIT_PERCENT', 'SPLIT_INVOKER'
BTREE_SPLIT_POSITION	TINYINT		B-tree split position (0~100)

INDEX_COLUMNS

Short description

Columns of indexes

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
TABLE_NAME	NVARCHAR(256)		Table name
TABLE_OID	BIGINT		Object ID of the table
INDEX_NAME	NVARCHAR(256)		Index name
INDEX_OID	BIGINT		Object ID of the index
CONSTRAINT	VARCHAR(5000)		Constraint type of the index: UNIQUE, NOT_NULL_UNIQUE, PRIMARY_KEY
COLUMN_NAME	NVARCHAR(256)		Name of the indexed column
POSITION	SMALLINT		Ordinal position of the indexed column
ASCENDING_ORDER	VARCHAR(5000)		Specifies whether the indexed column is ascending order or not: 'TRUE', 'FALSE'

INVALID_CONNECT_ATTEMPTS

Short description

Number of invalid connect attempts for a user between two successful connects

Structure

Column name	Data type	Unit	Description
USER_NAME	NVARCHAR(256)		Name of the user
SUCCESSFUL_CONNECT_TIME	TIMESTAMP		Time of the valid connect attempt
INVALID_CONNECT_ATTEMPTS	BIGINT		Number of invalid connect attempts for this user between last successful connect and given time

LAST_USED_PASSWORDS

Short description

Recently used passwords for each user

Structure

Column name	Data type	Unit	Description
USER_NAME	NVARCHAR(256)		Name of the user
PASSWORD_CHANGE_TIME	TIMESTAMP		Time this password was overwritten by the next one
PASSWORD	NVARCHAR(256)		Hash value of password

M_ATTACHED_STORAGES

Short description

Information about currently attached devices

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host
STORAGE_ID	INTEGER		Storage ID
PATH	VARCHAR(512)		Path
KEY	VARCHAR(256)		Key (from storage provider)
VALUE	VARCHAR(2000)		Value (from storage provider)

M_BACKUP_CATALOG

Short description

Common data for all backup catalog entries

Structure

Column name	Data type	Unit	Description
ENTRY_ID	BIGINT	Counter	Unique ID of a backup catalog entry
ENTRY_TYPE_NAME	VARCHAR(64)		Classification of backup catalog entries. The following types are supported: 'complete data backup', 'log backup', 'log missing'
BACKUP_ID	BIGINT	Counter	Unique ID of a data backup or a log backup respectively. All backup files of a single data backup share the same BACKUP_ID
SYS_START_TIME	TIMESTAMP		Start time given in server local time
UTC_START_TIME	TIMESTAMP		Start time given in UTC
SYS_END_TIME	TIMESTAMP		Stop time given in server local time
UTC_END_TIME	TIMESTAMP		Stop time given in UTC
STATE_NAME	VARCHAR(64)		Result of corresponding action: 'successful', 'failed', 'running', 'cancel pending', 'canceled'
COMMENT	VARCHAR(256)		Additional information
MESSAGE	VARCHAR(512)		Additional information

M_BACKUP_CATALOG_FILES

Short description

Location information of all backup catalog entries

Structure

Column name	Data type	Unit	Description
ENTRY_ID	BIGINT		Unique ID of a backup catalog entry
BACKUP_ID	BIGINT		Unique ID of a data backup and log backup respectively. All backup files of a single data backup share the same BACKUP_ID
SOURCE_ID	BIGINT		Persistence volume ID
SOURCE_TYPE_NAME	VARCHAR(64)		Type of persistence to be backed up: 'volume', 'topology'
HOST	VARCHAR(64)		Host name
SERVICE_TYPE_NAME	VARCHAR(64)		Type of database service: 'indexserver', 'nameserver', 'statisticsserver'
REDO_LOG_POSITION	BIGINT		In case of a data backup this value specifies the log position that must be processed next when a log recovery is requested after restoring the data backup
FIRST_REDO_LOG_POSITION	BIGINT		In case of a log backup this value specifies the log position of the oldest log entry contained in the backup
LAST_REDO_LOG_POSITION	BIGINT		In case of a log backup this value specifies the log position of the youngest log entry contained in the backup
BACKUP_SIZE	BIGINT		Specifies the size of the backup
DESTINATION_PATH	VARCHAR(512)		Data or log backup was written to this location

M_BLOCKED_TRANSACTIONS

Short description

Transaction list waiting for locks

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
BLOCKED_TRANSACTION_ID	INTEGER		Transaction ID of the transaction waiting for a lock
BLOCKED_UPDATE_TRANSACTION_ID			
	BIGINT		Transaction ID of the blocked update transaction
LOCK_OWNER_TRANSACTION_ID	INTEGER		Transaction ID of the transaction holding the lock
LOCK_OWNER_UPDATE_TRANSACTION_ID			
	BIGINT		Transaction ID of the update transaction holding the lock
BLOCKED_TIME	TIMESTAMP		Blocked timestamp
WAITING_RECORD_ID	VARCHAR(256)		ID of the record on which the lock is currently placed
WAITING_SCHEMA_NAME	VARCHAR(32)		Name of the schema on which the lock is currently placed
WAITING_TABLE_NAME	VARCHAR(32)		Name of the table on which the lock is currently placed
LOCK_TYPE	VARCHAR(32)		Lock type: 'RECORD', 'TABLE', 'METADATA'
LOCK_MODE	VARCHAR(32)		Lock mode: 'SHARED', 'EXCLUSIVE', 'INTENTIONAL EXCLUSIVE'

M_CACHES

Short description

Shows aggregated information on caches

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_CACHES_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_CACHES_RESET
```

or the management console command:

```
monitor reset -n M_CACHES_RESET
```

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
CACHE_ID	VARCHAR(128)		The unique identifier for the cache
TOTAL_SIZE	BIGINT	Byte	The maximum available memory budget in bytes available for the cache instance
USED_SIZE	BIGINT	Byte	The memory in bytes used by the cache instance
ENTRY_COUNT	BIGINT	Counter	The number of entries in the cache instance
INSERT_COUNT	BIGINT	Counter	The number of insertions into the cache instance
INVALIDATE_COUNT	BIGINT	Counter	The number of invalidations in the cache instance
HIT_COUNT	BIGINT	Counter	The number of cache hits for the cache instance
MISS_COUNT	BIGINT	Counter	The number of cache misses for the cache instance

M_CACHES_RESET

Short description

Shows aggregated information on caches (since last reset)

This view contains values accumulated since the last reset of the main view **M_CACHES**. Please refer to **M_CACHES** for information about the structure and use of this view.

In addition to the members mentioned in **M_CACHES**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_CATALOG_MEMORY

Short description

Memory usage information by catalog manager

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
CATEGORY	VARCHAR(128)		Category of catalog data
ALLOCATION_COUNT	BIGINT		The number of allocated entries
ALLOCATED_FIXED_PART_SIZE	BIGINT	Byte	Allocated memory size for fixed-size part
USED_FIXED_PART_SIZE	BIGINT	Byte	Used memory size for fixed-size part
ALLOCATED_VARIABLE_PART_SIZE			
	BIGINT	Byte	Allocated memory size for variable-size part
USED_VARIABLE_PART_SIZE	BIGINT	Byte	Used memory size for variable-size part

M_CE_CALCSCENARIOS

Short description

All available CalculationScenarios

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
SCENARIO_NAME	NVARCHAR(256)		Scenario name
IS_PERSISTENT	VARCHAR(6)	Boolean	Indicates if calculation scenario is persistent or transient
FLAGS	BIGINT		Calculation scenario flags
CREATE_TIME	TIMESTAMP		Creation time
MEMORY_SIZE	BIGINT	Byte	Memory size of loaded calculation scenario model
JSON	NCLOB		JSON string representing the calculation scenario

M_CE_CALCVIEW_DEPENDENCIES

Short description

All views that are referencing a CalculationScenario

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
SCHEMA_NAME	NVARCHAR(256)		Schema name of column view
VIEW_NAME	NVARCHAR(256)		Column view name
CALCNODE_NAME	NVARCHAR(256)		Referenced calculation node name
SCENARIO_NAME	NVARCHAR(256)		Scenario name

M_CE_DEBUG_INFOS

Short description

Debug information after execution of a CalculationScenario

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
SCENARIO_NAME	NVARCHAR(256)		Scenario name
NODE_NAME	NVARCHAR(256)		Calculation node name
STATEMENT_ID	BIGINT		Statement ID
DEBUG_TYPE_MASK	BIGINT		Bit mask indicating type of debug information
TABLE_NAME	NVARCHAR(256)		Table name (temporay intermediate result table)
COLUMN_NAMES	CLOB		Column names
TABLE_SIZE	BIGINT	Byte	Table size (temporay intermediate result table)
EXECUTION_TIMESTAMP	TIMESTAMP		Execution timestamp

M_CE_DEBUG_JSONS

Short description

All available JSONS (original, instantiated, optimized) of a scenario for a concrete query

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
SCENARIO_NAME	NVARCHAR(256)		Scenario name
STATEMENT_ID	BIGINT		Statement ID
TYPE	NVARCHAR(256)		JSON type (original, instantiated, optimized, extrace)
MODEL_JSON	NCLOB		JSON string representing the calculation scenario
EXECUTION_TIMESTAMP	TIMESTAMP		Execution timestamp

M_CE_DEBUG_NODE_MAPPING

Short description

Node mapping between CalculationNodes and RuntimeNodes after execution

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
STATEMENT_ID	BIGINT		Statement ID
SCENARIO_NAME	NVARCHAR(256)		Scenario name
NODE_NAME	NVARCHAR(256)		Calculation node name
NODE_TYPE	NVARCHAR(256)		Calculation node type (ROOT, TEMPLATE, CALC_DS)
SUCC_SCENARIO_NAME	NVARCHAR(256)		Successor scenario name
SUCC_NODE_NAME	NVARCHAR(256)		Successor calculation node name
RUNTIME_NODE_NAME	NVARCHAR(256)		Runtime calculation node name
RUNTIME_NODE_JSON	NCLOB		JSON string representing the calculation node
EXECUTION_TIMESTAMP	TIMESTAMP		Execution timestamp

M_CE_PLE_CALCSCENARIOS

Short description

All available CalculationScenarios created by the PlanningEngine

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
SCHEMA_NAME	NVARCHAR(256)		Schema name of calculation scenario
SCENARIO_NAME	NVARCHAR(256)		Calculation scenario name

M_CLIENT_VERSIONS

Short description

Versions of all supported client applications

Full description

This view shows versions of all supported client applications.

Structure

Column name	Data type	Unit	Description
CLIENT_TYPE	VARCHAR(256)		Client, e.g. 'ABAP_FDA', 'BI Modeler'
CLIENT_RELEASE_ID	SMALLINT		Technical release ID
CLIENT_RELEASE_DESC	VARCHAR(256)		Human readable release description
MIN_VERSION	SMALLINT		Lowest supported protocol version
MAX_VERSION	SMALLINT		Highest supported protocol version

M_COMPACTION_THREAD

Short description

Compaction thread statistics

Full description

The compaction thread automatically calls compaction of memory (including resource container shrink) in case memory gets low, but is not already exhausted. It also executes the compaction requests triggered by other processes (inter-process memory management). Some information about the compaction requests and the compacted sizes is shown in this view.

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
NUM_COMPACTION_COLLISIONS	BIGINT	Counter	Count of memory compaction collisions (other thread currently in compaction and compaction was not forced)
NUM_COMPACTIONS	BIGINT	Counter	Count number of compaction requests
LAST_SIZE_COMPACTION_REQUESTS			
	BIGINT	Byte	Compaction request in bytes (last)
MAX_SIZE_COMPACTION_REQUESTS			
	BIGINT	Byte	Compaction request in bytes (max)
MIN_SIZE_COMPACTION_REQUESTS			
	BIGINT	Byte	Compaction request in bytes (min)
SUM_SIZE_COMPACTION_REQUESTS			
	BIGINT	Byte	Compaction request in bytes (total)
AVG_SIZE_COMPACTION_REQUESTS			
	BIGINT	Byte	Compaction request in bytes (avg)
LAST_SIZE_FREEABLE	BIGINT	Byte	Bytes compactible at compaction call (last)
MAX_SIZE_FREEABLE	BIGINT	Byte	Bytes compactible at compaction call (max)
MIN_SIZE_FREEABLE	BIGINT	Byte	Bytes compactible at compaction call (min)
SUM_SIZE_FREEABLE	BIGINT	Byte	Bytes compactible at compaction call (total)
AVG_SIZE_FREEABLE	BIGINT	Byte	Bytes compactible at compaction call (avg)
LAST_SIZE_FREED_BY_GARBAGE_COLLECTION			
	BIGINT	Byte	Bytes compacted by memory garbage collection (defragmentation) (last)
MAX_SIZE_FREED_BY_GARBAGE_COLLECTION			

	BIGINT	Byte	Bytes compacted by memory garbage collection (defragmentation) (max)
MIN_SIZE_FREED_BY_GARBAGE_COLLECTION			
	BIGINT	Byte	Bytes compacted by memory garbage collection (defragmentation) (min)
SUM_SIZE_FREED_BY_GARBAGE_COLLECTION			
	BIGINT	Byte	Bytes compacted by memory garbage collection (defragmentation) (total)
AVG_SIZE_FREED_BY_GARBAGE_COLLECTION			
	BIGINT	Byte	Bytes compacted by memory garbage collection (defragmentation) (avg)
LAST_COMPACTION_RESULT	BIGINT	Byte	Compaction result in bytes (difference of allocated bytes before and after compaction, difference may be influenced by other factors than compaction) (last)
MAX_COMPACTION_RESULT	BIGINT	Byte	Compaction result in bytes (difference of allocated bytes before and after compaction, difference may be influenced by other factors than compaction) (max)
MIN_COMPACTION_RESULT	BIGINT	Byte	Compaction result in bytes (difference of allocated bytes before and after compaction, difference may be influenced by other factors than compaction) (min)
SUM_COMPACTION_RESULT	BIGINT	Byte	Compaction result in bytes (difference of allocated bytes before and after compaction, difference may be influenced by other factors than compaction) (total)
AVG_COMPACTION_RESULT	BIGINT	Byte	Compaction result in bytes (difference of allocated bytes before and after compaction, difference may be influenced by other factors than compaction) (avg)

M_CONDITIONAL_VARIABLES

Short description

Semaphore statistics

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_CONDITIONAL_VARIABLES_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_CONDITIONAL_VARIABLES_RESET
```

or the management console command:

```
monitor reset -n M_CONDITIONAL_VARIABLES_RESET
```

Full description

This view contains information about single conditional variable objects or groups of conditional variable objects. It does not contain information about all conditional variables.

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
STATISTICS_NAME	VARCHAR(128)		Statistics object name
STATISTICS_ID	BIGINT		Statistics object unique ID
WAIT_COUNT	BIGINT	Counter	Count of wait calls
BLOCKING_COUNT	BIGINT	Counter	Count of blocking wait calls
TIMEOUT_COUNT	BIGINT	Counter	Count of timeouts
WAIT_RATE	DOUBLE	Percent	Wait rate
LAST_BLOCKING_TIME	BIGINT	Microsecond	Time of blocking wait calls (last)
MAX_BLOCKING_TIME	BIGINT	Microsecond	Time of blocking wait calls (max)
MIN_BLOCKING_TIME	BIGINT	Microsecond	Time of blocking wait calls (min)
SUM_BLOCKING_TIME	BIGINT	Microsecond	Time of blocking wait calls (total)
AVG_BLOCKING_TIME	BIGINT	Microsecond	Time of blocking wait calls (avg)
CREATE_COUNT	BIGINT	Counter	Count of semaphore creation (for shared statistics only)
DESTROY_COUNT	BIGINT	Counter	Count of semaphore destruction (for shared statistics only)

M_CONDITIONAL_VARIABLES_RESET

Short description

Semaphore statistics (since last reset)

This view contains values accumulated since the last reset of the main view **M_CONDITIONAL_VARIABLES**. Please refer to **M_CONDITIONAL_VARIABLES** for information about the structure and use of this view.

In addition to the members mentioned in **M_CONDITIONAL_VARIABLES**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_CONNECTIONS

Short description

Detailed information on connections between a client and database. Information includes connection status, client information, connection type, resource utilization

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
CONNECTION_ID	INTEGER		Connection ID
TRANSACTION_ID	INTEGER		Transaction ID
START_TIME	TIMESTAMP		Connected Time
IDLE_TIME	BIGINT	Millisecond	Time that the connection is unused and idle
CONNECTION_STATUS	VARCHAR(128)		Connection Status: RUNNING or IDLE
CLIENT_HOST	VARCHAR(64)		Host name of client machine
CLIENT_IP	VARCHAR(16)		IP of client machine
CLIENT_PID	BIGINT		Client Process ID
USER_NAME	NVARCHAR(256)		User name
CONNECTION_TYPE	VARCHAR(128)		Connection type: Remote, Local, History (remote), History (local)
OWN	VARCHAR(6)	Boolean	Own connection: TRUE if own connection, FALSE if not
IS_HISTORY_SAVED	VARCHAR(6)	Boolean	No longer vaid field - will be deprecated soon
MEMORY_SIZE_PER_CONNECTION			
	BIGINT	Byte	Allocated memory size per connection
AUTO_COMMIT	VARCHAR(6)	Boolean	Commit mode of the current transaction: TRUE if the current connection is in auto-commit mode, FALSE otherwise
LAST_ACTION	VARCHAR(128)		The last action done by the current connection: ExecuteGroup, CommitTrans, AbortTrans, PrepareStatement, CloseStatement, ExecutePrepared, ExecuteStatement, FetchCursor, CloseCursor, LobGetPiece, LogPutPiece, LobFind, Authenticate, Connect, Disconnect, ExecQidItab, CursorFetchItab, InsertIncompleteItab, AbapStream, TxStartXA, TxJoinXA
CURRENT_STATEMENT_ID	VARCHAR(256)		Current statement ID
CURRENT_OPERATOR_NAME	VARCHAR(64)		Current operator name

FETCHED_RECORD_COUNT	BIGINT		Fetched record count
SENT_MESSAGE_SIZE	BIGINT	Byte	Total size of messages sent by the current connection
SENT_MESSAGE_COUNT	BIGINT		Total message count sent by the current connection
RECEIVED_MESSAGE_SIZE	BIGINT	Byte	Total size of messages received by the current connection
RECEIVED_MESSAGE_COUNT	BIGINT		Total message count received by the current connection
CREATOR_THREAD_ID	BIGINT		Thread ID who created the current connection
CREATED_BY	VARCHAR(256)		Engine component that created the connections: Session, Planning, Repository, CalcEngine, Authentication, Table Exporter, Loader, LLVM, JSVM, IMS Search API, OLAP Engine, Mergedog, Ping Status, Name Server, Queue Server, SQL Stored Procedure, Authorization, TrexViaDbsl from ABAP, HybridTable Reorganizer, Session external
IS_ENCRYPTED	VARCHAR(6)	Boolean	Encrypted: TRUE if the secure communication is enabled (SSL enabled), FALSE, otherwise
END_TIME	TIMESTAMP		The time when the connection is closed for history connections
PARENT_CONNECTION_ID	INTEGER		Parent connection ID
CLIENT_DISTRIBUTION_MODE	VARCHAR(128)		Client distribution mode of the current connection (i.e. [distribution] client_distribution_mode)
LOGICAL_CONNECTION_ID	INTEGER		Logical connection ID in statement routing

M_CONNECTION_STATISTICS

Short description

Detailed statistics on each connection between an application and database

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
CONNECTION_ID	INTEGER		Connection ID
SELECT_EXECUTION_COUNT	BIGINT		Total number of select statement executions
SELECT_TOTAL_EXECUTION_TIME			
	BIGINT	Microsecond	Total execution time of select statement execution
SELECT_AVG_EXECUTION_TIME	REAL	Microsecond	Average execution time of select statement execution
SELECT_MAX_EXECUTION_TIME	BIGINT	Microsecond	Maximum execution time of select statement execution
SELECT_FOR_UPDATE_COUNT	BIGINT		Total number of select for update exeution
SELECT_FOR_UPDATE_TOTAL_EXECUTION_TIME			
	BIGINT	Microsecond	Total execution time of select for update execution
SELECT_FOR_UPDATE_AVG_EXECUTION_TIME			
	REAL	Microsecond	Average execution time of select for update execution
SELECT_FOR_UPDATE_MAX_EXECUTION_TIME			
	BIGINT	Microsecond	Maximum execution time of select for update execution
SELECT_FOR_UPDATE_LOCK_WAIT_COUNT			
	BIGINT		Number of lock waits during select for update
SELECT_FOR_UPDATE_TOTAL_LOCK_WAIT_TIME			
	BIGINT	Microsecond	Total lock wait time during select for update
SELECT_FOR_UPDATE_AVG_LOCK_WAIT_TIME			
	REAL	Microsecond	Average lock wait time during select for update
SELECT_FOR_UPDATE_MAX_LOCK_WAIT_TIME			
	BIGINT	Microsecond	Maximum lock wait time uring select for update
UPDATE_COUNT	BIGINT		Total number of update statement executions

UPDATE_TOTAL_EXECUTION_TIME			
	BIGINT	Microsecond	Total execution time of update statement executions
UPDATE_AVG_EXECUTION_TIME	REAL	Microsecond	Average execution time of update statement executions
UPDATE_MAX_EXECUTION_TIME	BIGINT	Microsecond	Maximum execution time of update statement executions
UPDATE_LOCK_WAIT_COUNT	BIGINT		Number of lock waits during update statement executions
UPDATE_TOTAL_LOCK_WAIT_TIME			
	BIGINT	Microsecond	Total lock wait execution time during update statement executions
UPDATE_AVG_LOCK_WAIT_TIME	REAL	Microsecond	Average lock wait time during update statement exectuions
UPDATE_MAX_LOCK_WAIT_TIME	BIGINT	Microsecond	Maximum lock wait time during update statement executions
READ_ONLY_TRANSACTION_COUNT			
	BIGINT	Microsecond	Number of read only transactions
READ_ONLY_TRANSACTION_TOTAL_EXECUTION_TIME			
	BIGINT	Microsecond	Total execution time of read only transactions
READ_ONLY_TRANSACTION_AVG_EXECUTION_TIME			
	REAL	Microsecond	Average execution time of read only transactions
READ_ONLY_TRANSACTION_MAX_EXECUTION_TIME			
	BIGINT	Microsecond	Maximum execution time of read only transactions
UPDATE_TRANSACTION_COUNT	BIGINT		Number of update transactions
UPDATE_TRANSACTION_TOTAL_EXECUTION_TIME			
	BIGINT	Microsecond	Total execution time of update transactions
UPDATE_TRANSACTION_AVG_EXECUTION_TIME			
	REAL	Microsecond	Average execution time of update transactions
UPDATE_TRANSACTION_MAX_EXECUTION_TIME			
	BIGINT	Microsecond	Maximum execution time of update transactions
ROLLBACK_COUNT	BIGINT		Number of rollbacked transactions
ROLLBACK_TOTAL_EXECUTION_TIME			
	BIGINT	Microsecond	Total execution time of rollbacks
ROLLBACK_AVG_EXECUTION_TIME			
	REAL	Microsecond	Average execution time of rollbacks
ROLLBACK_MAX_EXECUTION_TIME			
	BIGINT	Microsecond	Maximum execution time of rollbacks

OTHERS_COUNT	BIGINT		Number of other statement executions including data definition statements and data control statements
OTHERS_TOTAL_EXECUTION_TIME			
	BIGINT	Microsecond	Total execution time of other statements
OTHERS_AVG_EXECUTION_TIME	REAL	Microsecond	Average execution time of other statements
OTHERS_MAX_EXECUTION_TIME	BIGINT	Microsecond	Maximum execution time of other statements
OTHERS_LOCK_WAIT_COUNT	BIGINT		Total lock wait count of other statements
OTHERS_TOTAL_LOCK_WAIT_TIME			
	BIGINT	Microsecond	Total lock wait time of other statements
OTHERS_AVG_LOCK_WAIT_TIME	REAL	Microsecond	Average lock wait time of other statements
OTHERS_MAX_LOCK_WAIT_TIME	BIGINT	Microsecond	Maximum lock wait time of other statements
LAST_EXECUTED_TIME	TIMESTAMP		Last execution timestamp with this connection
AVG_EXECUTION_MEMORY_SIZE	BIGINT		Average memory size used during each execution
MAX_EXECUTION_MEMORY_SIZE	BIGINT		Maximum memory size used during each execution
MIN_EXECUTION_MEMORY_SIZE	BIGINT		Minimum memory size used during each execution
TOTAL_EXECUTION_MEMORY_SIZE			
	BIGINT		Sum of memory size used during each execution
AVG_PREPARATION_TIME	BIGINT		Average time of statement preparation
MAX_PREPARATION_TIME	BIGINT		Maximum time of statement preparation
MIN_PREPARATION_TIME	BIGINT		Minimim time of statement preparation
TOTAL_PREPARATION_TIME	BIGINT		Total time of statement preparation
TOTAL_PREPARATION_COUNT	BIGINT		Total count of statement preparation
EXECUTION_COUNT_BY_ROUTING			
	BIGINT		Execution count by the client routed connection in statement routing
END_TIME	TIMESTAMP		The time when the connection is closed for history connections

M_CONTAINER_DIRECTORY

Short description

ContainerDirectory statistics

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
CNT_CREATE	BIGINT	Counter	Count of creates
CNT_CREATE_ROLLBACK	BIGINT	Counter	Count of rolled back creates
CNT_REMOVE	BIGINT	Counter	Count of removes
CNT_REMOVE_FAIL	BIGINT	Counter	Count of failed removes
CNT_REMOVE_ROLLBACK	BIGINT	Counter	Count of rolled back removes
CNT_MOVE	BIGINT	Counter	Count of moves
CNT_GET_PHYSICALSIZE	BIGINT	Counter	Count of getPhysicalSize
CNT_GET	BIGINT	Counter	Count of retrieved containers
CNT_GET_FAIL	BIGINT	Counter	Count of failed gets
CNT_BEGIN	BIGINT	Counter	Count of used iterators
CNT_ITERATED	BIGINT	Counter	Count of iterated containers
CNT_CACHEHIT	BIGINT	Counter	Count of cache hits
CNT_CACHEMISS	BIGINT	Counter	Count of cache misses
CNT_CONTAINERS	BIGINT	Counter	Count of existing containers

M_CONTAINER_NAME_DIRECTORY

Short description

ContainerNameDirectory statistics

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
CNT_CREATE	BIGINT	Counter	Count of creates
CNT_CREATE_FAIL	BIGINT	Counter	Count of failed creates
CNT_INITIAL_CREATE	BIGINT	Counter	Count of creates on load
CNT_INITIAL_SKIP	BIGINT	Counter	Count of skips on load
CNT_REMOVE	BIGINT	Counter	Count of removes
CNT_REMOVE_FAIL	BIGINT	Counter	Count of failed removes
CNT_REMOVE_ALL	BIGINT	Counter	Count of removeAll
CNT_RENAME	BIGINT	Counter	Count of renames
CNT_RENAME_FAIL	BIGINT	Counter	Count of failed renames
CNT_EXISTS	BIGINT	Counter	Count of checked containers
CNT_GET	BIGINT	Counter	Count of retrieved containers
CNT_GET_FAIL	BIGINT	Counter	Count of failed gets
CNT_BEGIN	BIGINT	Counter	Count of used iterators
CNT_ITERATE	BIGINT	Counter	Count of iterated containers
CNT_CONTAINERS	BIGINT	Counter	Count of existing containers

M_CONTEXT_MEMORY

Short description

Memory allocator statistics

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_CONTEXT_MEMORY_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_CONTEXT_MEMORY_RESET
```

or the management console command:

```
monitor reset -n M_CONTEXT_MEMORY_RESET
```

Full description

This view contains information about memory consumption grouped by connections and/or users. It does not contain information about all memory, only on memory, which can be uniquely associated with either a connection, a statement or a user.

To see information about allocated memory by component, use **M_HEAP_MEMORY**.

See also:

M_HEAP_MEMORY

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
STATISTICS_ID	BIGINT		Statistics object unique ID
CATEGORY	VARCHAR(128)		Allocator name
DEPTH	BIGINT		Depth
INCLUSIVE_SIZE_IN_USE	BIGINT	Byte	Current size of this allocator, including suballocators
INCLUSIVE_COUNT_IN_USE	BIGINT	Counter	Number of blocks currently in use, including suballocators
INCLUSIVE_ALLOCATED_SIZE	BIGINT	Byte	Total allocated size in this allocator and suballocators
INCLUSIVE_DEALLOCATED_SIZE			
	BIGINT	Byte	Total deallocated size in this allocator and suballocators
INCLUSIVE_ALLOCATED_COUNT	BIGINT	Counter	Count of allocations, including suballocators
INCLUSIVE_DEALLOCATED_COUNT			

	BIGINT	Counter	Count of deallocations, including suballocators
INCLUSIVE_MAX_SINGLE_ALLOCATION_SIZE			
	BIGINT	Byte	Maximum ever allocated block size in this allocator and suballocators
INCLUSIVE_PEAK_ALLOCATION_SIZE			
	BIGINT	Byte	Maximum size of this allocator and suballocators (estimate)
EXCLUSIVE_SIZE_IN_USE	BIGINT	Byte	Current size of this allocator
EXCLUSIVE_COUNT_IN_USE	BIGINT	Counter	Number of blocks currently in use
EXCLUSIVE_ALLOCATED_SIZE	BIGINT	Byte	Total allocated size in this allocator
EXCLUSIVE_DEALLOCATED_SIZE			
	BIGINT	Byte	Total deallocated size in this allocator
EXCLUSIVE_ALLOCATED_COUNT	BIGINT	Counter	Count of allocations
EXCLUSIVE_DEALLOCATED_COUNT			
	BIGINT	Counter	Count of deallocations
EXCLUSIVE_MAX_SINGLE_ALLOCATION_SIZE			
	BIGINT	Byte	Maximum ever allocated block size in this allocator
EXCLUSIVE_PEAK_ALLOCATION_SIZE			
	BIGINT	Byte	Maximum size of this allocator (estimate)
EXCLUSIVE_ALLOC_ERRORS	BIGINT	Counter	Count of allocation errors
MALLOC_PROXY_CACHE_MISSES	BIGINT	Counter	Count of malloc proxy cache misses
FLAGS	VARCHAR(64)		Allocator flags

M_CONTEXT_MEMORY_RESET

Short description

Memory allocator statistics (since last reset)

This view contains values accumulated since the last reset of the main view **M_CONTEXT_MEMORY**. Please refer to **M_CONTEXT_MEMORY** for information about the structure and use of this view.

In addition to the members mentioned in **M_CONTEXT_MEMORY**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_CONVERTER_STATISTICS

Short description

Converter statistics

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_CONVERTER_STATISTICS_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_CONVERTER_STATISTICS_RESET
```

or the management console command:

```
monitor reset -n M_CONVERTER_STATISTICS_RESET
```

Full description

This view contains information about the converter, which administers logical page numbers and maps them to physical pages within the DataVolume(s).

See also:

M_PAGEACCESS_STATISTICS, **M_DATA_VOLUME_PAGE_STATISTICS**

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
TYPE	VARCHAR(32)		Type of converter
MAX_LEVEL	BIGINT	Counter	Maximum level, e.g. level of root page
MAX_PAGENUMBER	BIGINT	HEXID	Maximum page number
ALLOCATED_PAGE_COUNT	BIGINT	Counter	Number of currently allocated pages
ALLOCATED_PAGE_SIZE	BIGINT	Byte	Total size [bytes] of currently allocated pages
MAX_ALLOCATED_PAGE_COUNT	BIGINT	Counter	Maximum number of allocated pages
MAX_ALLOCATED_PAGE_SIZE	BIGINT	Byte	Maximum size [bytes] of allocated pages
ALLOCATE_PAGE_COUNT	BIGINT	Counter	Number of page allocations
ALLOCATE_OR_GET_STATIC_PAGE_COUNT			
	BIGINT	Counter	Number of page allocations or retrievals during static phase
DEALLOCATE_PAGE_COUNT	BIGINT	Counter	Number of page deallocations
ASSIGN_PHYSICAL_PAGE_COUNT			
	BIGINT	Counter	Number of physical page assignments

UNASSIGN_PHYSICAL_PAGE_COUNT			
	BIGINT	Counter	Number of physical page unassignments
UNASSIGN_PHYSICAL_PAGE_COUNT_DURING_DROP_SNAPSHOT			
	BIGINT	Counter	Number of physical page unassignments during drop snapshot
CREATE_SNAPSHOT_COUNT	BIGINT	Counter	Number of snapshots created
DROP_SNAPSHOT_COUNT	BIGINT	Counter	Number of snapshots dropped
WRITE_CONVERTERPAGE_LEVEL0_COUNT			
	BIGINT	Counter	Number of level 0 converter pages written to disk
WRITE_CONVERTERPAGE_LEVEL1_COUNT			
	BIGINT	Counter	Number of level 1 converter pages written to disk
WRITE_CONVERTERPAGE_LEVEL2_COUNT			
	BIGINT	Counter	Number of level 2 converter pages written to disk
WRITE_CONVERTERPAGE_LEVEL3_COUNT			
	BIGINT	Counter	Number of level 3 converter pages written to disk
WRITE_CONVERTERPAGE_LEVEL4_COUNT			
	BIGINT	Counter	Number of level 4 converter pages written to disk

M_CONVERTER_STATISTICS_RESET

Short description

Converter statistics (since last reset)

This view contains values accumulated since the last reset of the main view **M_CONVERTER_STATISTICS**. Please refer to **M_CONVERTER_STATISTICS** for information about the structure and use of this view.

In addition to the members mentioned in **M_CONVERTER_STATISTICS**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_CS_COLUMNS

Short description

Runtime information of columns of column tables

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
SCHEMA_NAME	NVARCHAR(256)		Schema name
TABLE_NAME	NVARCHAR(256)		Table name
COLUMN_NAME	NVARCHAR(256)		Column name
PART_ID	INTEGER		Partition ID. 0 for non-partitioned tables and 1 through # of partitions for partitioned tables
MEMORY_SIZE_IN_TOTAL	BIGINT	Byte	Sum of MEMORY_SIZE_IN_MAIN and MEMORY_SIZE_IN_DELTA columns
MEMORY_SIZE_IN_MAIN	BIGINT	Byte	Current memory consumption in main; 0 if not loaded
MEMORY_SIZE_IN_DELTA	BIGINT	Byte	Current memory consumption in delta; 0 if not loaded
UNCOMPRESSED_SIZE	BIGINT	Byte	Estimated size assuming that column is not compressed
COMPRESSION_RATIO_IN_PERCENTAGE			
	DOUBLE	Percent	MEMORY_SIZE_IN_MAIN / UNCOMPRESSED_SIZE * 100
COUNT	BIGINT		Record count; 0 if not loaded
DISTINCT_COUNT	BIGINT		Distinct count of values; 0 if not loaded
COMPRESSION_TYPE	VARCHAR(16)		Type of column compression: can be SPARSE, PREFIXED, CLUSTERED, INDIRECT, RLE or DEFAULT if column is only dictionary coded. Columns in M_CS_COLUMNS show the runtime value, which can be changed during the runtime
INDEX_TYPE	VARCHAR(16)		Type of inverted index; can be FULL, SIGNATURE or NONE for no index
INDEX_LOADED	VARCHAR(16)		Load status of inverted index; can be NOT_SUPPORTED, UNLOADED, NOW_LOADING or LOADED
IMPLEMENTATION_FLAGS	BIGINT		Column internal implementation spec summary
LOADED	VARCHAR(6)	Boolean	Flag to indicate that column is loaded into memory

M_CS_PARTITIONS

Short description

Partition information of column tables

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema Name
TABLE_NAME	NVARCHAR(256)		Table Name
PART_ID	INTEGER		Part ID. 0 for non-partitioned tables and 1 through number of parts for partitioned tables
PARTITION	INTEGER		Partition ID. 0 for non-partitioned tables and 1 through number of partitions for partitioned tables
SUBPARTITION	INTEGER		Subpartition ID. 0 for non-partitioned tables or tables without multi-level partitioning and 1 through number of subpartitions for partitioned table with multi-level partitioniong
RANGE	VARCHAR(64)		Range definition of partition/subpartition for range-partitioned tables, empty otherwise

M_CS_TABLES

Short description

Runtime data of column tables

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
SCHEMA_NAME	NVARCHAR(256)		Schema name
TABLE_NAME	NVARCHAR(256)		Table name
PART_ID	INTEGER		Partition ID. 0 for non-partitioned tables and 1 through number of partitions for partitioned tables
MEMORY_SIZE_IN_TOTAL	BIGINT	Byte	Total memory size is the sum of memory size in main, delta, and history parts
MEMORY_SIZE_IN_MAIN	BIGINT	Byte	Current memory consumption in main; this value varies depending on the number of attributes actually loaded
MEMORY_SIZE_IN_DELTA	BIGINT	Byte	Current memory consumption in delta
MEMORY_SIZE_IN_HISTORY_MAIN			
	BIGINT	Byte	Current memory consumption in history-main; 0 for normal non-history tables
MEMORY_SIZE_IN_HISTORY_DELTA			
	BIGINT	Byte	Current memory consumption in history-delta; 0 for normal non-history tables
ESTIMATED_MAX_MEMORY_SIZE_IN_TOTAL			
	BIGINT	Byte	Estimated maximum memory consumption in total for table fully loaded
RECORD_COUNT	BIGINT	Byte	Record count
RAW_RECORD_COUNT_IN_MAIN	BIGINT		Current number of entries in table main part, this value differs from number of visible table main rows because there are entries of modified rows marked as invalidated
RAW_RECORD_COUNT_IN_DELTA	BIGINT		Current number of entries in table delta part, this value differs from number of visible table delta rows because there are additional entries such as deleted rows or updated rows; can contain deleted records
RAW_RECORD_COUNT_IN_HISTORY_MAIN			
	BIGINT		Raw record count in history-main
RAW_RECORD_COUNT_IN_HISTORY_DELTA			
	BIGINT		Raw record count in history-delta

LAST_COMPRESSED_RECORD_COUNT			
	BIGINT		The number of entries in main during last compression run
MAX_UDIV	BIGINT		Maximum table row number, this number is pure technical and only internally used
MAX_MERGE_CID	BIGINT		Maximum commit-ID of transactions for which changes were already merged to table main part
IS_DELTA2_ACTIVE	VARCHAR(6)	Boolean	Flag to indicate that a second delta is used, during a table delta merge updates and inserts are stored to a second delta because the delta one is locked
IS_DELTA_LOADED	VARCHAR(6)	Boolean	Flag to indicate that delta part of table is loaded
IS_LOG_DELTA	VARCHAR(6)	Boolean	Flag to indicate that currently delta log is written
PERSISTENT_MERGE	VARCHAR(6)	Boolean	Flag to indicate that new main part will be written to disk during table delta merge unless requested differently
CREATE_TIME	TIMESTAMP		Created time
MODIFY_TIME	TIMESTAMP		Modified time
LAST_MERGE_TIME	TIMESTAMP		Point in time, Unix time format, last time the table delta part was merged into main part
LAST_REPLAY_LOG_TIME	TIMESTAMP		Point in time, Unix time format, last time the table log was replayed
LOADED	VARCHAR(10)		Flat to show how many columns of table are loaded in memory (NO, PARTIALLY, and FULL); see M_CS_COLUMNS for each column
READ_COUNT	BIGINT		Number of read accesses on the table or partition
WRITE_COUNT	BIGINT		Number of write accesses on the table or partition
MERGE_COUNT	BIGINT		Number of delta merges done on the table or partition

M_DATABASE

Short description

Database information

Structure

Column name	Data type	Unit	Description
DATABASE_NAME	VARCHAR(256)		SAPSYSTEMNAME
HOST	VARCHAR(64)		SAPGLOBALHOST
START_TIME	TIMESTAMP		Start time
VERSION	VARCHAR(32)		Version: major.minor.patch.build

M_DATABASE_HISTORY

Short description

Installation version history

Structure

Column name	Data type	Unit	Description
INSTALL_TIME	TIMESTAMP		Install/first start time
VERSION	VARCHAR(32)		Version: major.minor.patch.build

M_DATA_VOLUMES

Short description

DataVolume statistics

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
FILE_NAME	VARCHAR(512)		Filename of DataVolume
FILE_ID	BIGINT		FileID of DataVolume
SIZE	BIGINT	Byte	Size of DataVolume
MAX_SIZE	BIGINT	Byte	Max. size of DataVolume

M_DATA_VOLUME_PAGE_STATISTICS

Short description

Page usage statistics on data volumes

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_DATA_VOLUME_PAGE_STATISTICS_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_DATA_VOLUME_PAGE_STATISTICS_RESET
```

or the management console command:

```
monitor reset -n M_DATA_VOLUME_PAGE_STATISTICS_RESET
```

Full description

This view cotains information about the number and distribution of free, used and shadow pages inside DataVolumes. INITIAL_BLOCK_COUNT is the number of pages the database was started with. TOTAL_*_COUNT give the numbers of blocks allocated, freed or set to status FreeAfterSavepoint since the start of the database. Columns SUPERBLOCK_COUNT, USED_BLOCK_COUNT and SHADOW_BLOCK_COUNT contain the number of (super)blocks currently in use by the database. FILL_RATIO gives the ratio of minimum number of needed superblocks versus the actual number of used superblocks. Note that unused Superblocks are not part of this formula (see **M_DATA_VOLUME_SUPERBLOCK_STATISTICS**).

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
DATA_VOLUME_NAME	VARCHAR(512)		DataVolume name
PAGE_SIZECLASS	VARCHAR(16)		Page size class
PAGE_SIZE	BIGINT	Byte	Page size
SUPERBLOCK_SIZE	BIGINT	Byte	Superblock size
INITIAL_BLOCK_COUNT	BIGINT	Counter	Count of init pages
TOTAL_ALLOCATE_BLOCK_COUNT			
	BIGINT	Counter	Number of single and group allocated blocks
TOTAL_SET_BLOCK_FREE_COUNT			
	BIGINT	Counter	Number of single and group freed blocks
TOTAL_SET_BLOCK_FREE_AFTER_SAVEPOINT_COUNT			
	BIGINT	Counter	Number of single and group freed-after-savepoint blocks
SUPERBLOCK_COUNT	BIGINT	Counter	Number of used Superblocks

USED_BLOCK_COUNT	BIGINT	Counter	Count of used blocks
SHADOW_BLOCK_COUNT	BIGINT	Counter	Count of shadow blocks
FILL_RATIO	DOUBLE	Percent	Fill ratio

M_DATA_VOLUME_PAGE_STATISTICS_RESET

Short description

FreeBlockManager SizeClass statistics (since last reset)

This view contains values accumulated since the last reset of the main view **M_DATA_VOLUME_PAGE_STATISTICS**. Please refer to **M_DATA_VOLUME_PAGE_STATISTICS** for information about the structure and use of this view.

In addition to the members mentioned in **M_DATA_VOLUME_PAGE_STATISTICS**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_DATA_VOLUME_SUPERBLOCK_STATISTICS

Short description

FreeBlockManager Superblock statistics

Full description

This view shows information about the number and distribution of Superblocks inside DataVolumes. SUPERBLOCK_COUNT is the total number of Superblocks and USED_SUPERBLOCK_COUNT is the number of Superblocks currenbtly occupied by at least one used or shadow page. FILL_RATIO gives the ratio of the number of used Superblocks versus the total number of Superblocks. Note that the fill ratio of the Superblocks themselves is not part of this formula.

See also:

M_DATA_VOLUME_PAGE_STATISTICS

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
DATA_VOLUME_NAME	VARCHAR(512)		DataVolume name
SUPERBLOCK_SIZE	BIGINT	Byte	Superblock size
USED_SUPERBLOCK_COUNT	BIGINT	Counter	Count of used Superblocks
SUPERBLOCK_COUNT	BIGINT	Counter	Count of Superblocks
FILL_RATIO	DOUBLE	Percent	Fill ratio

M_DELTA_MERGE_STATISTICS

Short description

Table delta merge statistics

Full description

This view shows table delta merge statistics. It lists the table delta merge events since the indexserver was last restarted. Table delta merges, optimize compression runs and application merge hints are listed separately.

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
TYPE	VARCHAR(7)		MERGE (table delta merge), HINT (application merge hint), SPARSE (optimize compression)
SCHEMA_NAME	NVARCHAR(256)		Schema name
TABLE_NAME	NVARCHAR(256)		Table name
PART_ID	INTEGER		Partition number (0 for non-partitioned tables and 1 through number of partitions for partitioned tables
HISTORY	VARCHAR(6)	Boolean	Flag to indicate history part was merged
MEMORY_MERGE	VARCHAR(6)	Boolean	Flag to indicate table was merged in memory only
PASSPORT	VARCHAR(256)		External identifier for table merge called by an application
START_TIME	TIMESTAMP		Execution start time
EXECUTION_TIME	BIGINT	Millisecond	Execution duration
MOTIVATION	VARCHAR(8)		HARD, SMART, AUTO, FORCE
SUCCESS	VARCHAR(6)	Boolean	Call success flag; depends on field TYPE; HINT: application merge hint was accepted/rejected; MERGE/SPARSE: delta merge/optimize compression was completed with or without success
MERGED_DELTA_RECORDS	INTEGER	Counter	Number of documents in delta
LAST_ERROR	INTEGER		Error code of the last error that occurred. Can be used to explain why a merge did not succeed. See M_ERROR_CODES for error code descriptions

M_DISKS

Short description

Disk configuration and utilization of the host machine

Structure

Column name	Data type	Unit	Description
DISK_ID	INTEGER		Disk ID
DEVICE_ID	BIGINT		Device ID from OS
HOST	VARCHAR(64)		Host name. Only set if disk is only used by one host
PATH	VARCHAR(512)		Path
SUBPATH	VARCHAR(512)		Subpath if volume is on a separate disk
USAGE_TYPE	VARCHAR(128)		Usage Type: LOG,DATA,TRACE
TOTAL_SIZE	BIGINT	Byte	Volume Usage
USED_SIZE	BIGINT	Byte	Volume Size

M_ERROR_CODES

Short description

Error codes with descriptions

Structure

Column name	Data type	Unit	Description
CODE	INTEGER		Error Code
CODE_STRING	VARCHAR(256)		Error Code String
DESCRIPTION	VARCHAR(2000)		Error Description

M_EVENTS

Short description

Internal events

Full description

Important events (for example, DiskFull) reported by the database are shown in this view. The State of an event can be either NEW or HANDLED. CREATE TIME shows the time the event was created (and reported) and HANDLE time shows the time the event was handled (that is, cleared). Handled events are removed periodically by the StatisticsServer

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host
PORT	INTEGER		Port
TYPE	VARCHAR(256)		Type of event
ID	BIGINT		ID of event
INFOTEXT	VARCHAR(2000)		Additional info freetext
CREATE_TIME	TIMESTAMP		Time event was created
UPDATE_TIME	TIMESTAMP		Time event was updated
HANDLE_TIME	TIMESTAMP		Time event was handled
STATE	VARCHAR(256)		State of event
ACKNOWLEDGED	VARCHAR(6)		Event acknowledged
FAILED_HANDLES	BIGINT		Number of failed handle attempts

M_EXPENSIVE_STATEMENTS

Short description

All statements with duration longer than a threshold

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
CONNECTION_ID	BIGINT		Connection ID
TRANSACTION_ID	BIGINT		Transaction ID
STATEMENT_ID	NVARCHAR(256)		Statement ID
DB_USER	NVARCHAR(256)		User name
APP_USER	NVARCHAR(256)		Application user name
START_TIME	TIMESTAMP		Statement start time
DURATION_MICROSEC	BIGINT	Microsecond	Statement duration
OBJECT_NAME	NVARCHAR(5000)		Related objects
OPERATION	NVARCHAR(5000)		Operation
RECORDS	BIGINT		Number of records
STATEMENT_STRING	NCLOB		Statement string
PARAMETERS	NVARCHAR(5000)		Parameters
ERROR_CODE	INTEGER		Error code
ERROR_TEXT	NVARCHAR(5000)		Error message
LOCK_WAIT_COUNT	INTEGER		Accumulated lock wait count
LOCK_WAIT_DURATION	BIGINT		Accumulated lock wait duration
ALLOC_MEM_SIZE_ROWSTORE	BIGINT	Byte	Allocated memory size for row store
ALLOC_MEM_SIZE_COLSTORE	BIGINT	Byte	Allocated memory size for column store

M_EXPORT_BINARY_STATUS

Short description

Export status information for current session

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
CONNECTION_ID	INTEGER		Connection ID to distinguish data of each import session (-1 for Python)
SCHEMA_NAME	NVARCHAR(256)		Schema of table or view being imported
TABLE_NAME	NVARCHAR(256)		Name of table or view being imported
INDEX_TYPE	VARCHAR(128)		Index type: 'UNKNOWN', 'PHYSICAL', 'OLAP', 'JOIN', 'HIERARCHY', 'CALCULATION', ...
STATUS	VARCHAR(128)		Import status. 'queued', 'working', 'skipped', 'done', 'failed'
ERROR	VARCHAR(128)		Empty if import was successful, error text otherwise

M_EXTRACTORS

Short description

Direct extractor connection (DXC) status information

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Name of the schema where the DSO tables are generated
TABLE_NAME	NVARCHAR(256)		Name of the DSO specific status table
CLIENT	NVARCHAR(3)		The ID of the client
STATUS	VARCHAR(5)		General status overview saying 'OK' or 'ERROR'
REQUEST_ID	INTEGER		SID of the request used for activation
ERROR_MESSAGE	NVARCHAR(5000)		The message in case of an error
PHASE	VARCHAR(20)		The phase of the last running operation
TIMESTAMP	TIMESTAMP		Timestamp of the operation phase

M_FEATURES

Short description

All supported features

Structure

Column name	Data type	Unit	Description
COMPONENT_NAME	VARCHAR(16)		Name of the component owning the feature
FEATURE_NAME	VARCHAR(32)		Name of the feature
FEATURE_VERSION	BIGINT		Version of the feature

M_FULLTEXT_QUEUES

Short description

Fulltext index queue status

Structure

Column name	Data type	Unit	Description
INDEX_OID	BIGINT		OID of the corresponding fulltext index
SCHEMA_NAME	NVARCHAR(256)		Schema of the data table
TABLE_NAME	NVARCHAR(256)		Name of the data table
PART_ID	INTEGER		Part number if split, 0 otherwise
COLUMN_NAME	NVARCHAR(256)		Name of the document column
STATUS	NVARCHAR(128)		Queue status (ACTIVE, SUSPENDED)
TOTAL_DOCUMENT_COUNT	BIGINT		Total number of documents in the original table
INDEXED_DOCUMENT_COUNT	BIGINT		Number of successfully indexed documents
QUEUE_DOCUMENT_COUNT	BIGINT		Number of documents currently in the queue
ERROR_DOCUMENT_COUNT	BIGINT		Number of failed documents
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port

M_GARBAGE_COLLECTION_STATISTICS

Short description

Garbage collection/history manager statistics

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_GARBAGE_COLLECTION_STATISTICS_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_GARBAGE_COLLECTION_STATISTICS_RESET
```

or the management console command:

```
monitor reset -n M_GARBAGE_COLLECTION_STATISTICS_RESET
```

Full description

This view shows various statistics about garbage collection jobs.

Garbage collection is used to remove old versions of data objects from the system. Afterwards no transaction can reference them. References to such objects are kept in history (cleanup) files, which are processed by the garbage collector.

Normally, the system runs in statement consistency isolation level, that is, a consistent view is acquired when an SQL command starts. Alternatively, the system can be run in transaction consistency isolation level, where the consistent view is acquired when the transaction starts and is held until the transaction terminates (snapshot isolation, similar to SERIALIZABLE isolation level). Each consistent view contains a reference to a minReadTID Transaction ID (TID), which is the minimum TID, from which the changes are "seen" by the transaction. Global minReadTID (minimum of minReadTIDs of all consistent views) is used by the garbage collector to determine which history files can be cleaned.

In the case that history files accumulate and MIN_READ_TID doesn't advance, there is a good chance that some transaction holds its consistent view for very long time (for example, a long-running transaction of a forgotten read transaction in a GUI). You can look into **M_TRANSACTIONS** to see information about each running transaction.

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
HISTORY_COUNT	INTEGER	Counter	Current count of history files in GC
WAITER_COUNT	INTEGER	Counter	Current count of GC waiters
MIN_READ_TID	BIGINT	TID	Last known minimum TID of a reading transaction at EOT

LAST_STARTED_TID	BIGINT	TID	TID of the last started GC job
FIRST_WAITING_TID	BIGINT	TID	TID of the first waiting GC job
ENTERS	BIGINT	Counter	Count of history files which entered the queue
STARTED_JOBS	BIGINT	Counter	Count of garbage collection jobs started
PROCESSED_JOBS	BIGINT	Counter	Count of garbage collection files processed
QUEUE_LOADS	BIGINT	Counter	Count of all GC queue loads
QUEUE_LOADS_NONEMPTY	BIGINT	Counter	Count of GC queue loads which found some elements
QUEUE_EMPTY_COUNT	BIGINT	Counter	Count of queue empty states after GC finished
SAVEPOINTS	BIGINT	Counter	Count of savepoints
LAST_HISTORY_SIZE_AT_SVP	BIGINT	Counter	Count of history files present at savepoint (last)
MAX_HISTORY_SIZE_AT_SVP	BIGINT	Counter	Count of history files present at savepoint (max)
MIN_HISTORY_SIZE_AT_SVP	BIGINT	Counter	Count of history files present at savepoint (min)
SUM_HISTORY_SIZE_AT_SVP	BIGINT	Counter	Count of history files present at savepoint (total)
AVG_HISTORY_SIZE_AT_SVP	DOUBLE	Rate	Count of history files present at savepoint (avg)
EMPTY_HISTORY_AT_SVP_COUNT			
	BIGINT	Counter	Count of savepoints finding empty queue

M_GARBAGE_COLLECTION_STATISTICS_RESET

Short description

Garbage collection/history manager statistics (since last reset)

This view contains values accumulated since the last reset of the main view **M_GARBAGE_COLLECTION_STATISTICS**. Please refer to **M_GARBAGE_COLLECTION_STATISTICS** for information about the structure and use of this view.

In addition to the members mentioned in **M_GARBAGE_COLLECTION_STATISTICS**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_HEAP_MEMORY

Short description

Memory allocator statistics

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_HEAP_MEMORY_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_HEAP_MEMORY_RESET
```

or the management console command:

```
monitor reset -n M_HEAP_MEMORY_RESET
```

Full description

This view contains information about memory consumption of various components in the system. Parallel to heap memory, you can also query memory consumption by connection/statement/user using **M_CONTEXT_MEMORY**.

The overhead of allocators is not considered here.

See also:

M_CONTEXT_MEMORY, **M_MEMORY**

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
STATISTICS_ID	BIGINT		Statistics object unique ID
CATEGORY	VARCHAR(128)		Allocator name
DEPTH	BIGINT		Depth
INCLUSIVE_SIZE_IN_USE	BIGINT	Byte	Current size of this allocator, including suballocators
INCLUSIVE_COUNT_IN_USE	BIGINT	Counter	Number of blocks currently in use, including suballocators
INCLUSIVE_ALLOCATED_SIZE	BIGINT	Byte	Total allocated size in this allocator and suballocators
INCLUSIVE_DEALLOCATED_SIZE			
	BIGINT	Byte	Total deallocated size in this allocator and suballocators
INCLUSIVE_ALLOCATED_COUNT	BIGINT	Counter	Count of allocations, including suballocators
INCLUSIVE_DEALLOCATED_COUNT			

	BIGINT	Counter	Count of deallocations, including suballocators
INCLUSIVE_MAX_SINGLE_ALLOCATION_SIZE			
	BIGINT	Byte	Maximum ever allocated block size in this allocator and suballocators
INCLUSIVE_PEAK_ALLOCATION_SIZE			
	BIGINT	Byte	Maximum size of this allocator and suballocators (estimate)
EXCLUSIVE_SIZE_IN_USE	BIGINT	Byte	Current size of this allocator
EXCLUSIVE_COUNT_IN_USE	BIGINT	Counter	Number of blocks currently in use
EXCLUSIVE_ALLOCATED_SIZE	BIGINT	Byte	Total allocated size in this allocator
EXCLUSIVE_DEALLOCATED_SIZE			
	BIGINT	Byte	Total deallocated size in this allocator
EXCLUSIVE_ALLOCATED_COUNT	BIGINT	Counter	Count of allocations
EXCLUSIVE_DEALLOCATED_COUNT			
	BIGINT	Counter	Count of deallocations
EXCLUSIVE_MAX_SINGLE_ALLOCATION_SIZE			
	BIGINT	Byte	Maximum ever allocated block size in this allocator
EXCLUSIVE_PEAK_ALLOCATION_SIZE			
	BIGINT	Byte	Maximum size of this allocator (estimate)
EXCLUSIVE_ALLOC_ERRORS	BIGINT	Counter	Count of allocation errors
MALLOC_PROXY_CACHE_MISSES	BIGINT	Counter	Count of malloc proxy cache misses
FLAGS	VARCHAR(64)		Allocator flags

M_HEAP_MEMORY_RESET

Short description

Memory allocator statistics (since last reset)

This view contains values accumulated since the last reset of the main view **M_HEAP_MEMORY**. Please refer to **M_HEAP_MEMORY** for information about the structure and use of this view.

In addition to the members mentioned in **M_HEAP_MEMORY**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_HISTORY_INDEX_LAST_COMMIT_ID

Short description

Last commit ID of history index for each session

Structure

Column name	Data type	Unit	Description
SESSION_ID	BIGINT		Session ID
SESSION_OWNER_VOLUME_ID	BIGINT		Volume ID of this session
LAST_COMMIT_ID	BIGINT		Last commit ID of history index in this session

M_HOST_INFORMATION

Short description

Host information such as machine, OS configuration

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
KEY	VARCHAR(32)		Key
VALUE	VARCHAR(256)		Value

M_HOST_RESOURCE_UTILIZATION

Short description

Host resource utilization, CPU time is in milliseconds and added across all cores since system start

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
FREE_PHYSICAL_MEMORY	BIGINT	Byte	Free physical memory on the host
USED_PHYSICAL_MEMORY	BIGINT	Byte	Used physical memory on the host
FREE_SWAP_SPACE	BIGINT	Byte	Free swap memory on the host
USED_SWAP_SPACE	BIGINT	Byte	Used swap memory on the host
TOTAL_CPU_USER_TIME	BIGINT	Millisecond	CPU time spent in user mode
TOTAL_CPU_SYSTEM_TIME	BIGINT	Millisecond	CPU time spent in kernel mode
TOTAL_CPU_WIO_TIME	BIGINT	Millisecond	CPU time spent in wait IO (Linux only, Windows always 0)
TOTAL_CPU_IDLE_TIME	BIGINT	Millisecond	CPU idle time
SYS_TIMESTAMP	TIMESTAMP		Host timestamp in local time zone
UTC_TIMESTAMP	TIMESTAMP		Host timestamp in UTC

M_IMPORT_BINARY_STATUS

Short description

Import status information for current session

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
CONNECTION_ID	INTEGER		Connection ID to distinguish data of each import session (-1 for Python)
SCHEMA_NAME	NVARCHAR(256)		Schema of table or view being imported
TABLE_NAME	NVARCHAR(256)		Name of table or view being imported
INDEX_TYPE	VARCHAR(128)		Index type: 'UNKNOWN', 'PHYSICAL', 'OLAP', 'JOIN', 'HIERARCHY', 'CALCULATION', ...
STATUS	VARCHAR(128)		Import status. 'queued', 'working', 'skipped', 'done', 'failed'
ERROR	VARCHAR(128)		Empty if import was successful, error text otherwise

M_INIFILES

Short description

All configuration files

Full description

Change Inifiles Calls

For the SYSTEM layer (you cannot set the DEFAULT layer):

```
ALTER SYSTEM INIFILE('filename', layer) SET ('section1', 'key1') = 'value1', ('secti
on2', 'key2') = 'value2';
```

For the TENANT and HOST layers:

```
ALTER SYSTEM INIFILE('filename', 'layer', 'layername') SET ('section1', 'key1') = 'v
alue1', ('section2', 'key2') = 'value2';
```

To remove a section or key:

```
ALTER SYSTEM INIFILE('filename', 'layer'[, 'layername']) REMOVE ('section1', 'key1')
, ('section2', 'key2'), ('section3') with reconfigure;
```

With reconfigure:

```
ALTER SYSTEM INIFILE ..... [ WITH RECONFIGURE ] ;
```

The services to be reconfigured are defined in inifiles.ini

Structure

Column name	Data type	Unit	Description
FILE_NAME	VARCHAR(256)		Configuration file name
DEFAULT_LAYER	VARCHAR(6)	Boolean	TRUE if file has configuration on this layer (Always TRUE)
SYSTEM_LAYER	VARCHAR(6)	Boolean	TRUE if file has configuration on this layer
TENANT_LAYER	VARCHAR(6)	Boolean	TRUE if file has configuration on this layer
HOST_LAYER	VARCHAR(6)	Boolean	TRUE if file has configuration on this layer

M_INIFILE_CONTENTS

Short description

Configuration information from inifiles

Structure

Column name	Data type	Unit	Description
FILE_NAME	VARCHAR(256)		Configuration file name
LAYER_NAME	VARCHAR(256)		Configuration layer: DEFAULT, SYSTEM, TENANT, HOST
TENANT_NAME	VARCHAR(256)		Tenant name, if layer is TENANT
HOST	VARCHAR(64)		Host name, if layer is HOST
SECTION	VARCHAR(128)		Configuration section name
KEY	VARCHAR(128)		Configuration key name
VALUE	VARCHAR(2000)		Configuration value

M_JOB_PROGRESS

Short description

Current long running system operations

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
SCHEMA_NAME	NVARCHAR(256)		Schema of the object
OBJECT_NAME	NVARCHAR(256)		Name of the object
JOB_NAME	NVARCHAR(128)		Type of the operation
CONNECTION_ID	INTEGER		Connection that triggered the operation
START_TIME	TIMESTAMP		Start time of the operation
CURRENT_PROGRESS	INTEGER		Steps of the operation that are already finished
MAX_PROGRESS	INTEGER		Maximum steps of the operation

M_LANDSCAPE_HOST_CONFIGURATION

Short description

Host roles in a distributed landscape

Full description

The view is read-only. To change the configuration use SAP HANA studio, which uses this stored procedure:

```
UPDATE_LANDSCAPE_CONFIGURATION(ACTION in VARCHAR(80), DATA in NCLOB) -> resultset (N
AME,VALUE)
```

HOST_ACTIVE

Value	Description
YES	All services are active
STARTING	Some service are active, host is starting
STOPPING	Some services are active, host is stopping
NO	No services are active

HOST_STATUS

Value	Description
OK	Landscape operational, host actual role equals configured role
IGNORE	Landscape operational, host configured as standby is available, but not used
INFO	Landscape operational, host actual role is different from configured role
WARNING	Landscape not operational, host should become operational after startup/failover
ERROR	Landscape not operational, host is missing

FAILOVER_STATUS

Value	Description
	No failover pending/active
waiting	Failure detected. Waiting for Restart to prevent unneccessary failover
failover to 'host'	Failover to 'host' active.
failback to 'host'	Failback to worker 'host' active. Happens when a standby host was assigned and is stopped. There is not automatic failback while the standby host is assigned, because this would cause some downtime
failed	Failover not possible, e.g. no more standby hosts available. See nameserver.trc for details

NAMESERVER_CONFIG_ROLE

During installation up to 3 hosts are automatically configured as MASTER candidate. All others are SLAVE.

It is recommended to specifiy these 3 hosts in the connection url (see SAP HANA Administration Guide, chapter 'Configuring clients for Failover Support), because at least one of these hosts will be active.

NAMESERVER_ACTUAL_ROLE
Exactly one of the master candidates is MASTER, all other are SLAVE.

The actual master nameserver and master indexserver are both on the same host.

INDEXSERVER_CONFIG_ROLE
During installation with 'hdbaddhost' the host is configured as WORKER or STANDBY

INDEXSERVER_ACTUAL_ROLE
Exactly one host is MASTER, the others are SLAVE or STANDBY.

During normal operation, when all hosts are available, a configured WORKER has an actual role of MASTER or SLAVE and a configured STANDBY has an actual role of STANDBY. After failover the actual roles switch and the HOST_STATUS changes from OK+IGNORE to 2*INFO.

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
HOST_ACTIVE	VARCHAR(128)		Host active Status (Summary of Active values of all services on that host)
HOST_STATUS	VARCHAR(128)		Host status
FAILOVER_STATUS	VARCHAR(128)		Failover status
FAILOVER_GROUP	VARCHAR(256)		Failover group
NAMESERVER_CONFIG_ROLE	VARCHAR(32)		Nameserver configured role: MASTER, SLAVE
NAMESERVER_ACTUAL_ROLE	VARCHAR(32)		Nameserver actual role: MASTER, SLAVE
INDEXSERVER_CONFIG_ROLE	VARCHAR(32)		Indexserver configured role: WORKER, STANDBY
INDEXSERVER_ACTUAL_ROLE	VARCHAR(32)		Indexserver actual role: MASTER, SLAVE, STANDBY
STORAGE_PARTITION	INTEGER		Storage Partition ~= subpath used by that host

M_LICENSE

Short description

Information on the currently valid license (if any) installed on this system

Structure

Column name	Data type	Unit	Description
HARDWARE_KEY	VARCHAR(256)		Hardware key of this SAP HANA installation
SYSTEM_ID	VARCHAR(256)		System identifier (SID) of this SAP HANA database
INSTALL_NO	VARCHAR(256)		Installation number of this SAP HANA database
SYSTEM_NO	VARCHAR(256)		System number of this SAP HANA database
PRODUCT_NAME	VARCHAR(256)		Name of the licensed software product, e.g. SAP HANA
PRODUCT_LIMIT	BIGINT		Licensed amount of main memory as specified by the license
PRODUCT_USAGE	BIGINT		Currently utilized amount of main memory
START_DATE	TIMESTAMP		Start date of the validity period of the license
EXPIRATION_DATE	TIMESTAMP		Expiration date of the validity period of the license
LAST_SUCCESSFUL_CHECK	TIMESTAMP		Latest date on which the license was successfully checked and found valid
PERMANENT	VARCHAR(6)		TRUE if the license is a permanent one, FALSE if the license is a temporary one
VALID	VARCHAR(6)		TRUE if the license is currently valid, otherwise FALSE
ENFORCED	VARCHAR(6)		TRUE if the license is a memory-enforced license, otherwise FALSE
LOCKED_DOWN	VARCHAR(6)		TRUE if the system is locked down due to license status, otherwise FALSE
MEASUREMENT_XML	CLOB		Measurement log produced for this licensed system with current memory consumption

M_LIVECACHE_CONTAINER_STATISTICS

Short description

LiveCache container statistics

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_LIVECACHE_CONTAINER_STATISTICS_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_LIVECACHE_CONTAINER_STATISTICS_RESE
T
```

or the management console command:

```
monitor reset -n M_LIVECACHE_CONTAINER_STATISTICS_RESET
```

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
CONTAINER_ID	BIGINT		Container ID
OMS_CLASS_ID	INTEGER		OMS class ID
OMS_CLASS_NAME	VARCHAR(64)		OMS class name
OMS_SCHEMA_HANDLE	INTEGER		OMS schema handle
OMS_CONTAINER_NO	INTEGER		OMS container number
CLASS_SIZE	BIGINT	Byte	Object class size
KEY_SIZE	INTEGER	Byte	Object key size
KEY_PARTITION_COUNT	INTEGER		Key partition count
KEY_PARTITION_PREFIX_SIZE	INTEGER	Byte	Key partition prefix length
CACHED_KEYS	VARCHAR(6)		Cached keys active
COPY_ON_UPDATE	VARCHAR(6)		Copy on update active
OBJECT_COUNT	BIGINT	Counter	Object count
OBJECT_SIZE_SUM	BIGINT	Byte	Object size sum
HISTORY_OBJECT_COUNT	BIGINT	Counter	History (mark deleted) object count
HISTORY_SIZE_SUM	BIGINT	Byte	History data size sum
PAGE_COUNT	BIGINT	Counter	Page count
PAGE_SIZE_SUM	BIGINT	Byte	Page size sum
OBJECT_CREATE_COUNT	BIGINT	Counter	Object create count since restart
OBJECT_UPDATE_COUNT	BIGINT	Counter	Object update count since restart
OBJECT_DELETE_COUNT	BIGINT	Counter	Object delete count since restart
CREATE_TIME	TIMESTAMP		Create timestamp

M_LIVECACHE_CONTAINER_STATISTICS_RESET

Short description

LiveCache container statistics (since last reset)

This view contains values accumulated since the last reset of the main view **M_LIVECACHE_CONTAINER_STATISTICS**. Please refer to **M_LIVECACHE_CONTAINER_STATISTICS** for information about the structure and use of this view.

In addition to the members mentioned in **M_LIVECACHE_CONTAINER_STATISTICS**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_LIVECACHE_LOCKS

Short description

Detailed information on the Object Management System (OMS) locks

Full description

The liveCache uses its own lock-manager for object locks, container locks, and schema locks. This view shows information about all the locks, which are currently kept in this lock manager.

Attention:

Exclusive locks on objects are in most cases not managed by this lock manager, but the necessary lock-information is stored in the respective object header.
This view can only be used if liveCache is enabled.

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
ID	VARCHAR(20)		ID of the lock (OID, container ID, or schema ID dependent on the lock class)
CLASS	VARCHAR(20)		The class that the lock belongs to
GRANTED_MODE	VARCHAR(20)		The mode that the lock has currently granted to some request(s)
MODE	VARCHAR(20)		The mode that the request is requesting on the lock
TYPE	VARCHAR(10)		The type specifies when the lock can be removed (either at transaction-end (EOT) or if not visible anymore (Consistent))
STATE	VARCHAR(10)		State of this lock request
TID	BIGINT	TID	Transaction ID belonging to this lock request
TIMEOUT	BIGINT		Remaining timeout, if timeout is specified

M_LIVECACHE_LOCK_STATISTICS

Short description

LiveCache lock statistics

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_LIVECACHE_LOCK_STATISTICS_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_LIVECACHE_LOCK_STATISTICS_RESET
```

or the management console command:

```
monitor reset -n M_LIVECACHE_LOCK_STATISTICS_RESET
```

Full description

The liveCache uses its own lock-manager for object locks, container locks, and schema locks. This view shows the accumulated statistics of the lock requests to this lock-manager, which have been executed since restart.

Attention:

This view is only usable if liveCache is enabled.

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
CONTAINER_EXCLUSIVE_COUNT	BIGINT	Counter	Number of exclusive lock requests on containers
CONTAINER_SHARED_COUNT	BIGINT	Counter	Number of shared lock requests on containers
CONTAINER_COLLISION_COUNT	BIGINT	Counter	Number of lock collisions on containers
CONTAINER_TIMEOUT_COUNT	BIGINT	Counter	Number of lock timeouts on containers
SCHEMA_EXCLUSIVE_COUNT	BIGINT	Counter	Number of exclusive lock requests on schemas
SCHEMA_SHARED_COUNT	BIGINT	Counter	Number of shared lock requests on schemas
SCHEMA_COLLISION_COUNT	BIGINT	Counter	Number of lock collisions on schemas
SCHEMA_TIMEOUT_COUNT	BIGINT	Counter	Number of lock timeouts on schemas
OBJECT_EXCLUSIVE_COUNT	BIGINT	Counter	Number of exclusive lock requests on objects
OBJECT_SHARED_COUNT	BIGINT	Counter	Number of shared lock requests on objects
OBJECT_COLLISION_COUNT	BIGINT	Counter	Number of lock collisions on objects
OBJECT_TIMEOUT_COUNT	BIGINT	Counter	Number of timeouts on objects
COMMITTED_REQUESTS_COUNT	BIGINT	Counter	Number of committed requests

M_LIVECACHE_LOCK_STATISTICS_RESET

Short description

LiveCache lock statistics (since last reset)

This view contains values accumulated since the last reset of the main view **M_LIVECACHE_LOCK_STATISTICS**. Please refer to **M_LIVECACHE_LOCK_STATISTICS** for information about the structure and use of this view.

In addition to the members mentioned in **M_LIVECACHE_LOCK_STATISTICS**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_LIVECACHE_OMS_VERSIONS

Short description

Detailed information on the OMS versions that currently exists

Full description

Status information is shown for each OMS version that currently exists.

Attention:
This view can only be used if liveCache is enabled.

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VERSION_ID	VARCHAR(23)		ID of the OMS version
CREATE_DATE	VARCHAR(29)		Timestamp of the creation of the OMS version
LAST_OPEN_DATE	VARCHAR(29)		Timestamp, when the OMS version was opened last
IS_OPEN	VARCHAR(2)		Flag indicating whether the version is currently open
HEAP_USAGE	BIGINT		Memory that is currently used by the version
VERSION_DESCRIPTION	NVARCHAR(513)		Description

M_LIVECACHE_PROCEDURE_STATISTICS

Short description

LiveCache procedure statistics

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_LIVECACHE_PROCEDURE_STATISTICS_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_LIVECACHE_PROCEDURE_STATISTICS_RESE
T
```

or the management console command:

```
monitor reset -n M_LIVECACHE_PROCEDURE_STATISTICS_RESET
```

Full description

For each liveCache procedure, which has been called already since the last restart, statistics are shown.

Attention:

This view can only be used if liveCache is enabled.

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
OBJECT_NAME	VARCHAR(128)		Object name
METHOD_NAME	VARCHAR(128)		Method name
CALL_COUNT	BIGINT	Counter	Number of calls of the procedure
SUM_RUN_TIME	BIGINT	Microsecond	Sum runtime of the procedure
MAX_RUN_TIME	BIGINT	Microsecond	Max runtime of the procedure
MIN_RUN_TIME	BIGINT	Microsecond	Min runtime of the procedure
AVERAGE_RUN_TIME	BIGINT	Microsecond	Average runtime of the procedure
DEREF_COUNT	BIGINT	Counter	Number of OID derefs
DEREF_BASE_COUNT	BIGINT	Counter	Number of OID derefs against basis
DEREF_BASE_IN_VERSION_COUNT			
	BIGINT	Counter	Number of OID derefs against basis from within an OMS version
DEREF_KEYED_OBJECT_COUNT	BIGINT	Counter	Number of key derefs
DEREF_KEYED_OBJECT_BASE_COUNT			
	BIGINT	Counter	Number of key derefs against basis
DEREF_KEYED_OBJECT_BASE_IN_VERSION_COUNT			

	BIGINT	Counter	Number of key derefs against basis from within an OMS version
ITER_BASE_COUNT	BIGINT	Counter	Number of objects read via an OID-iterator from the basis
ITER_KEYED_OBJECT_BASE_COUNT			
	BIGINT	Counter	Number of objects read via a key-iterator from the basis
ITER_BASE_IN_VERSION_COUNT			
	BIGINT	Counter	Number of objects read via an OID-iterator from the basis from within an OMS version
ITER_IN_VERSION_COUNT	BIGINT	Counter	Number of objects, created in an OMS-version, read via an OID-iterator
ITER_KEYED_OBJECT_BASE_IN_VERSION_COUNT			
	BIGINT	Counter	Number of objects read via an OID-iterator from the basis from within an OMS version
ITER_KEYED_OBJECT_IN_VERSION_COUNT			
	BIGINT	Counter	Number of objects, created in an OMS version, read via a key-iterator
DEREF_VAR_OBJECT_COUNT	BIGINT	Counter	Number of derefs of var objects
DEREF_VAR_OBJECT_BASE_COUNT			
	BIGINT	Counter	Number of derefs of var objects against basis
DEREF_VAR_OBJECT_BASE_SIZE			
	BIGINT	Byte	Accumulated size of var objects read from the basis
DEREF_VAR_OBJECT_BASE_IN_VERSION_COUNT			
	BIGINT	Counter	Number of derefs of var objects against basis from within an OMS version
DEREF_VAR_OBJECT_BASE_IN_VERSION_SIZE			
	BIGINT	Byte	Accumulated size of var objects read from the basis from within an OMS version
NEW_OBJECT_COUNT	BIGINT	Counter	Number of newly created standard objects
NEW_KEYED_OBJECT_COUNT	BIGINT	Counter	Number of newly created keyed objects
NEW_VAR_OBJECT_COUNT	BIGINT	Counter	Number of newly created var objects
NEW_OBJECT_IN_VERSION_COUNT			
	BIGINT	Counter	Number of newly created standard objects in an OMS version
NEW_KEYED_OBJECT_IN_VERSION_COUNT			
	BIGINT	Counter	Number of newly created keyed objects in an OMS version
NEW_VAR_OBJECT_IN_VERSION_COUNT			
	BIGINT	Counter	Number of newly created var objects in an OMS version

STORE_COUNT	BIGINT	Counter	Number of store calls on standard objects
STORE_IN_VERSION_COUNT	BIGINT	Counter	Number of store calls on standard objects in an OMS versino
STORE_KEYED_OBJECT_COUNT	BIGINT	Counter	Number of store calls on keyed objects
STORE_KEYED_OBJECT_IN_VERSION_COUNT			
	BIGINT	Counter	Number of store calls on keyed objects in an OMS version
STORE_VAR_OBJECT_COUNT	BIGINT	Counter	Number of store calls on var objects
STORE_VAR_OBJECT_SIZE	BIGINT	Byte	Accumulated size of store calls on var objects
STORE_VAR_OBJECT_IN_VERSION_COUNT			
	BIGINT	Counter	Number of store calls on var objects in an OMS version
STORE_VAR_OBJECT_IN_VERSION_SIZE			
	BIGINT	Byte	Accumulated size of store calls on var objects in an OMS version
STORE_BASE_COUNT	BIGINT	Counter	Number of updates of standard objects in the basis
STORE_KEYED_OBJECT_BASE_COUNT			
	BIGINT	Counter	Number of updates of keyed objects in the basis
STORE_VAR_OBJECT_BASE_COUNT			
	BIGINT	Counter	Number of updates of var objects in the basis
STORE_VAR_OBJECT_BASE_SIZE			
	BIGINT	Byte	Accumulated sizes of updated objects in the basis
DELETE_COUNT	BIGINT	Counter	Number of delete calls on standard objects
DELETE_KEYED_OBJECT_COUNT	BIGINT	Counter	Number of delete calls on keyed objects
DELETE_VAR_OBJECT_COUNT	BIGINT	Counter	Number of delete calls on var objects
DELETE_IN_VERSION_COUNT	BIGINT	Counter	Number of delete calls on standard objects in an OMS version
DELETE_KEYED_OBJECT_IN_VERSION_COUNT			
	BIGINT	Counter	Number of delete calls on keyed objects in an OMS version
DELETE_VAR_OBJECT_IN_VERSION_COUNT			
	BIGINT	Counter	Number of delete calls on var objects in an OMS version
DELETE_BASE_COUNT	BIGINT	Counter	Number of deleted standard objects in the basis
DELETE_KEYED_OBJECT_BASE_COUNT			
	BIGINT	Counter	Number of deleted keyed objects in the basis

DELETE_VAR_OBJECT_BASE_COUNT			
	BIGINT	Counter	Number of deleted var objects in the basis
LOCK_EXCLUSIVE_BASE_COUNT	BIGINT	Counter	Number of exclusive lock requests on standard objects in the basis
LOCK_SHARE_BASE_COUNT	BIGINT	Counter	Number of share lock reqeusts on standard objects in the basis
LOCK_EXCLUSIVE_KEYED_OBJECT_BASE_COUNT			
	BIGINT	Counter	Number of exclusive lock requests on keyed objects in the basis
LOCK_SHARE_KEYED_OBJECT_BASE_COUNT			
	BIGINT	Counter	Number of share lock reqeusts on keyed objects in the basis
LOCK_EXCLUSIVE_VAR_OBJECT_BASE_COUNT			
	BIGINT	Counter	Number of exclusive lock requests on var objects in the basis
LOCK_SHARE_VAR_OBJECT_BASE_COUNT			
	BIGINT	Counter	Number of share lock reqeusts on var objects in the basis
RELEASE_CALLED_COUNT	BIGINT	Counter	Number of OMS release calls
RELEASE_EXECUTED_COUNT	BIGINT	Counter	Number of released standard objects
RELEASE_EXECUTED_KEYED_OBJECT_COUNT			
	BIGINT	Counter	Number of releases keyed objects
RELEASE_EXECUTED_VAR_OBJECT_COUNT			
	BIGINT	Counter	Number of releases var objects
HISTORY_HOP_COUNT	BIGINT	Counter	Number of hops in the history chain during deref
ITER_HISTORY_HOP_COUNT	BIGINT	Counter	Number of hops in the history chain during iteration
EXCEPTION_COUNT	BIGINT	Counter	Number of dbp-exceptions thrown
OUT_OF_DATE_EXCEPTION_COUNT			
	BIGINT	Counter	Number of out-of-date-exceptions thrown
OUT_OF_MEMORY_EXCEPTION_COUNT			
	BIGINT	Counter	Number of out-of-memory-exceptions thrown
TIMEOUT_EXCEPTION_COUNT	BIGINT	Counter	Number of timeout-exceptions thrown
OMS_TERMINATE_COUNT	BIGINT	Counter	Number of OMS terminate calls
SUBTRANSACTION_ROLLBACK_COUNT			
	BIGINT	Counter	Number of rolled back sub-transactions
SUBTRANSACTION_COMMIT_COUNT			
	BIGINT	Counter	Number of committed sub-transactions
MAX_SUBTRANSACTION_LEVEL	BIGINT	Counter	Maximal subtransaction level
NEW_CONSISTENT_VIEW_COUNT	BIGINT	Counter	Number of new-consistent-view calls with objects

AVERAGE_NEW_CONSISTENT_VIEW_WAIT_TIME			
	BIGINT	Second	Average wait time of new-consistent-view calls with objects
NEW_CONSISTENT_VIEW_MAX_WAIT_TIME			
	BIGINT	Second	Maximal wait time of new-consistent-view calls with objects
KEY_CACHE_HIT_COUNT	BIGINT	Counter	Number of cache hits in the key-cache
KEY_MISS_CACHE_HIT_COUNT	BIGINT	Counter	Number of cache hits in the key-miss-cache
DEREF_VERSION_KEYED_OBJECT_COUNT			
	BIGINT	Counter	Number of key-derefs in an OMS version on objects created in this version
OMS_REHASH_COUNT	BIGINT	Counter	Number of rehashes of the OMS-object-hash
AVERAGE_HASH_CHAIN_SEARCH_LENGTH			
	BIGINT	Microsecond	Average search length on the hash-chains of the OMS-object-hash
MAX_HASH_CHAIN_LENGTH	BIGINT	Counter	Maximum length of a hash-chain of the OMS-object-hash
VERSION_CREATE_COUNT	BIGINT	Counter	Number of created OMS versions
VERSION_OPEN_COUNT	BIGINT	Counter	Number of open-version calls
VERSION_CLOSE_COUNT	BIGINT	Counter	Number of close-version calls
VERSION_DROP_COUNT	BIGINT	Counter	Number of dropped OMS versions
USER_ALLOC_COUNT	BIGINT	Counter	Number of allocations via a user allocator
USER_MAX_CHUNK_ALLOCATED_SIZE			
	BIGINT	Byte	Max size of an chunk allocated via a user allocator
USER_MIN_CHUNK_ALLOCATED_SIZE			
	BIGINT	Byte	Min size of an chunk allocated via a user allocator
USER_AVERAGE_CHUNK_ALLOCATED_SIZE			
	BIGINT	Microsecond	Average size of an chunk allocated via a user allocator
USER_DELETE_COUNT	BIGINT	Counter	Number of deallocations via a user allocator
USER_MAX_CHUNK_DELETED_SIZE			
	BIGINT	Byte	Max size of an chunk deallocated via a user allocator
USER_MIN_CHUNK_DELETED_SIZE			
	BIGINT	Byte	Min size of an chunk deallocated via a user allocator
USER_AVERAGE_CHUNK_DELETED_SIZE			
	BIGINT	Microsecond	Average size of an chunk deallocated via a user allocator

USER_DELTA_MAX_SIZE	BIGINT	Byte	Maximal difference between allocation and deallocation on a user allocator during the execution of a method
OMS_ALLOC_COUNT	BIGINT	Counter	Number of allocations via an OMS internal allocator
OMS_MAX_CHUNK_ALLOCATED_SIZE			
	BIGINT	Byte	Max size of an chunk allocated via an OMS internal allocator
OMS_MIN_CHUNK_ALLOCATED_SIZE			
	BIGINT	Byte	Min size of an chunk allocated via an OMS internal allocator
OMS_AVERAGE_CHUNK_ALLOCATED_SIZE			
	BIGINT	Microsecond	Average size of an chunk allocated via an OMS internal allocator
OMS_DELETE_COUNT	BIGINT	Counter	Number of deallocations via an OMS internal allocator
OMS_MAX_CHUNK_DELETED_SIZE			
	BIGINT	Byte	Max size of an chunk deallocated via an OMS internal allocator
OMS_MIN_CHUNK_DELETED_SIZE			
	BIGINT	Byte	Min size of an chunk deallocated via an OMS internal allocator
OMS_AVERAGE_CHUNK_DELETED_SIZE			
	BIGINT	Microsecond	Average size of an chunk deallocated via an OMS internal allocator
OMS_DELTA_MAX_SIZE	BIGINT	Byte	Maximum difference between allocation and deallocation on an OMS internal allocator during the execution of a method
STREAM_COMMUNICATION_TIME	BIGINT	Millisecond	Average stream communication time
STREAM_READ_COUNT	BIGINT	Counter	Number of communications for reading ABAP tables
STREAM_WRITE_COUNT	BIGINT	Counter	Number of communications for writing ABAP tables
STREAM_READ_ROW_COUNT	BIGINT	Counter	Number of rows read from ABAP tables
STREAM_WRITE_ROW_COUNT	BIGINT	Counter	Number of rows written to ABAP tables

M_LIVECACHE_PROCEDURE_STATISTICS_RESET

Short description

LiveCache procedure statistics (since last reset)

This view contains values accumulated since the last reset of the main view **M_LIVECACHE_PROCEDURE_STATISTICS**. Please refer to **M_LIVECACHE_PROCEDURE_STATISTICS** for information about the structure and use of this view.

In addition to the members mentioned in **M_LIVECACHE_PROCEDURE_STATISTICS**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_LIVECACHE_SCHEMA_STATISTICS

Short description

LiveCache schema statistics

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_LIVECACHE_SCHEMA_STATISTICS_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_LIVECACHE_SCHEMA_STATISTICS_RESET
```

or the management console command:

```
monitor reset -n M_LIVECACHE_SCHEMA_STATISTICS_RESET
```

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
OMS_SCHEMA_HANDLE	INTEGER		OMS schema ID
OMS_SCHEMA_NAME	NVARCHAR(96)		OMS schema name
CREATE_TIME	TIMESTAMP		Create timestamp

M_LIVECACHE_SCHEMA_STATISTICS_RESET

Short description

LiveCache schema statistics (since last reset)

This view contains values accumulated since the last reset of the main view **M_LIVECACHE_SCHEMA_STATISTICS**. Please refer to **M_LIVECACHE_SCHEMA_STATISTICS** for information about the structure and use of this view.

In addition to the members mentioned in **M_LIVECACHE_SCHEMA_STATISTICS**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_LOCK_WAITS_STATISTICS

Short description

Accumulated lock wait count and duration for record lock, table lock and metadata lock for all available services

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
LOCK_TYPE	VARCHAR(32)		Lock type: RECORD, TABLE, METADATA
TOTAL_LOCK_WAITS	BIGINT		Total lock wait count
TOTAL_LOCK_WAIT_TIME	BIGINT		Total lock wait duration

M_LOG_BUFFERS

Short description

Log buffer statistics

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_LOG_BUFFERS_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_LOG_BUFFERS_RESET
```

or the management console command:

```
monitor reset -n M_LOG_BUFFERS_RESET
```

Full description

The current configuration of in-memory log buffers is shown in the BUFFER_SIZE and BUFFER_COUNT columns. This defines how much log information can be collected transiently in memory, before the log queue becomes full.

Counters for buffer switches indicate performance of the in-memory log buffers. Normally, buffer switching happens without any waits. In the case of buffer full, however, a wait is necessary. Then, SWITCH_WAIT_COUNT is counted up, otherwise SWITCH_NOWAIT_COUNT is counted up. If the wait ratio is higher than one percent, this indicates a possible misconfiguration of the system. In this case:

- check if regular peaks exceed current log buffer configuraion and if so, increase log buffer size and/or count
- check if the I/O subsystem is performing poorly (see also **M_VOLUME_IO_STATISTICS**).

Due to the lock-free nature of the algorithm used, some race conditions can happen. These are properly detected and resolved. Additionally, a count of such races is recorded in SWITCH_OPEN_COUNT. Normally, the ratio of races to buffer switches should also be under one percent even for high workloads.

See also:

M_LOG_PARTITIONS, **M_LOG_SEGMENTS**, **M_VOLUME_IO_STATISTICS**

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
LOG_MODE	VARCHAR(16)		Log mode
BUFFER_SIZE	BIGINT	Kilobyte	Size of one log buffer in memory
BUFFER_COUNT	BIGINT	Counter	Count of log buffers in memory
SEGMENT_SIZE	BIGINT	Megabyte	Maximum size of one log segment

BACKUP_ENABLED	VARCHAR(6)		Log segment backup enabled flag (disabled on log backup history broken)
BACKUP_TIMEOUT	BIGINT	Second	Log segment backup timeout
SWITCH_NOWAIT_COUNT	BIGINT	Counter	Count of buffer switches without blocking on buffer semaphore
SWITCH_WAIT_COUNT	BIGINT	Counter	Count of buffer switches with blocking on buffer semaphore
SWITCH_OPEN_COUNT	BIGINT	Counter	Count of ignored still-open buffer switches (resolved races)

M_LOG_BUFFERS_RESET

Short description

Log buffer statistics (since last reset)

This view contains values accumulated since the last reset of the main view **M_LOG_BUFFERS**. Please refer to **M_LOG_BUFFERS** for information about the structure and use of this view.

In addition to the members mentioned in **M_LOG_BUFFERS**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_LOG_PARTITIONS

Short description

Log partition statistics

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_LOG_PARTITIONS_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_LOG_PARTITIONS_RESET
```

or the management console command:

```
monitor reset -n M_LOG_PARTITIONS_RESET
```

Full description

This view collects various performance statistics for each log partition. The collected statistics can be used to optimize workload.

See also:

M_LOG_BUFFERS, **M_LOG_SEGMENTS**, **M_VOLUME_IO_STATISTICS**

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
PARTITION_ID	BIGINT		Log partition ID
PATH	VARCHAR(512)		Log partition root path
LAST_BUFFER_PREPARE_SIZE	BIGINT	Byte	Size of log buffer at prepare time (actual log data size) (last)
MAX_BUFFER_PREPARE_SIZE	BIGINT	Byte	Size of log buffer at prepare time (actual log data size) (max)
MIN_BUFFER_PREPARE_SIZE	BIGINT	Byte	Size of log buffer at prepare time (actual log data size) (min)
SUM_BUFFER_PREPARE_SIZE	BIGINT	Byte	Size of log buffer at prepare time (actual log data size) (total)
AVG_BUFFER_PREPARE_SIZE	BIGINT	Byte	Size of log buffer at prepare time (actual log data size) (avg)
LAST_BUFFER_OVERHEAD_SIZE	BIGINT	Byte	Size of log buffer alignment overhead at I/O time (last)
MAX_BUFFER_OVERHEAD_SIZE	BIGINT	Byte	Size of log buffer alignment overhead at I/O time (max)
MIN_BUFFER_OVERHEAD_SIZE	BIGINT	Byte	Size of log buffer alignment overhead at

			I/O time (min)
SUM_BUFFER_OVERHEAD_SIZE	BIGINT	Byte	Size of log buffer alignment overhead at I/O time (total)
AVG_BUFFER_OVERHEAD_SIZE	BIGINT	Byte	Size of log buffer alignment overhead at I/O time (avg)
LAST_BUFFER_IO_SIZE	BIGINT	Byte	Size of log buffer at I/O time (actual data plus alignment overhead) (last)
MAX_BUFFER_IO_SIZE	BIGINT	Byte	Size of log buffer at I/O time (actual data plus alignment overhead) (max)
MIN_BUFFER_IO_SIZE	BIGINT	Byte	Size of log buffer at I/O time (actual data plus alignment overhead) (min)
SUM_BUFFER_IO_SIZE	BIGINT	Byte	Size of log buffer at I/O time (actual data plus alignment overhead) (total)
AVG_BUFFER_IO_SIZE	BIGINT	Byte	Size of log buffer at I/O time (actual data plus alignment overhead) (avg)
LAST_GROUP_COMMIT_FREQUENCY			
	BIGINT	Counter	Group commit frequency (callback count per buffer with sync callback) (last)
MAX_GROUP_COMMIT_FREQUENCY			
	BIGINT	Counter	Group commit frequency (callback count per buffer with sync callback) (max)
MIN_GROUP_COMMIT_FREQUENCY			
	BIGINT	Counter	Group commit frequency (callback count per buffer with sync callback) (min)
SUM_GROUP_COMMIT_FREQUENCY			
	BIGINT	Counter	Group commit frequency (callback count per buffer with sync callback) (total)
AVG_GROUP_COMMIT_FREQUENCY			
	BIGINT	Counter	Group commit frequency (callback count per buffer with sync callback) (avg)
LAST_CALLBACK_TIME	BIGINT	Microsecond	Callback time per buffer with sync callback (last)
MAX_CALLBACK_TIME	BIGINT	Microsecond	Callback time per buffer with sync callback (max)
MIN_CALLBACK_TIME	BIGINT	Microsecond	Callback time per buffer with sync callback (min)
SUM_CALLBACK_TIME	BIGINT	Microsecond	Callback time per buffer with sync callback (total)
AVG_CALLBACK_TIME	BIGINT	Microsecond	Callback time per buffer with sync callback (avg)
PREPARED_BUFFERS	BIGINT	Counter	Count of buffers prepared for I/O
WRITTEN_BUFFERS	BIGINT	Counter	Count of completed and written buffers
WRITTEN_BUFFERS_OOO	BIGINT	Counter	Count of buffers written out-of-order
NEW_SEGMENT_REQUEST_COUNT	BIGINT	Counter	Count of new log segment requests
FREE_SEGMENTS	BIGINT	Counter	Count of currently free log segments

IN_BACKUP_SEGMENTS	BIGINT	Counter	Count of currently in-backup log segments
IN_BACKUP_TRUNCATED_SEGMENTS			
	BIGINT	Counter	Count of currently in-backup truncated log segments
BACKED_UP_SEGMENTS	BIGINT	Counter	Count of log segments backed up so far
TOTAL_SEGMENTS	BIGINT	Counter	Total log segment count in the partition
RECOVERY_SEGMENTS_IN_LOAD	BIGINT	Counter	Count of segments in-load during recovery
RECOVERY_SEGMENTS_WAITING_FOR_LOAD			
	BIGINT	Counter	Count of segments waiting for load during recovery
RECOVERY_SEGMENTS_IN_PROCESS			
	BIGINT	Counter	Count of segments in process during recovery
RECOVERY_SEGMENTS_PROCESSED			
	BIGINT	Counter	Count of segments processed during recovery

M_LOG_PARTITIONS_RESET

Short description

Log partition statistics (since last reset)

This view contains values accumulated since the last reset of the main view **M_LOG_PARTITIONS**. Please refer to **M_LOG_PARTITIONS** for information about the structure and use of this view.

In addition to the members mentioned in **M_LOG_PARTITIONS**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_LOG_SEGMENTS

Short description

Log segment statistics

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_LOG_SEGMENTS_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_LOG_SEGMENTS_RESET
```

or the management console command:

```
monitor reset -n M_LOG_SEGMENTS_RESET
```

Full description

This view describes each allocated log segment and shows its current state and log position range, which is currently contained in this segment.

The following log segment states exist:

- **Formatting** - The log segment is being formatted and not yet used.
- **Preallocated** - The log segment has been preallocated, but never used.
- **Writing** - The log segment is currently being written.
- **Closed** - The log segment is closed, not backed up and is still required for restart.
- **Truncated** - The log segment is not required for restart, but has not been backed up.
- **BackedUp** - The log segment has been backed up, but is still required for restart.
- **Free** - The log segment has been backed up, it is not required for restart and can be reused.

In case you are using log_mode = **legacy**, you must execute a data backup in order to release log files for reuse (log segments will stay in the state `Closed` or `BackedUp` until the backup is done). After a data backup, empty log files can be freed explicitly after backup using the RECLAIM LOG SQL command. This command can also be used in other log modes after a savepoint to release empty log files freed by the savepoint.

See also:

M_LOG_BUFFERS, **M_LOG_PARTITIONS**, **M_VOLUME_IO_STATISTICS**

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
PARTITION_ID	BIGINT		Log partition ID
SEGMENT_ID	BIGINT		Log segment ID within partition

FILE_NAME	VARCHAR(512)		Log segment file name
FILE_OFFSET	BIGINT	Byte	Start position of log segment in the file
STATE	VARCHAR(16)		Log segment state
MIN_POSITION	BIGINT		First position contained in this log segment
MAX_POSITION	BIGINT		Position behind the last log record in this log segment (closed log segments only)
HOLE_POSITION	BIGINT		Start position of the log hole before this log segment (equal to min position if no hole)
USED_SIZE	BIGINT	Byte	Used log segment size in bytes
TOTAL_SIZE	BIGINT	Byte	Total log segment size in bytes
IN_BACKUP	VARCHAR(6)		Flag for log segment in backup

M_LOG_SEGMENTS_RESET

Short description

Log segment statistics (since last reset)

This view contains values accumulated since the last reset of the main view **M_LOG_SEGMENTS**. Please refer to **M_LOG_SEGMENTS** for information about the structure and use of this view.

In addition to the members mentioned in **M_LOG_SEGMENTS**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_MEMORY

Short description

Memory configuration and usage status by SAP HANA indexserver

Full description

This view contains information about current memory consumption. The following describes the contents of the individual lines of the monitor view (`VALUE`), as identified by `NAME`:

PROCESS_CODE_SIZE
Size of shared libraries/DLLs

PROCESS_STACK_SIZE
Size of thread stacks

PROCESS_MEMORY_SIZE
Virtual size of the process

PROCESS_RESIDENT_SIZE
Resident size of the process

SYSTEM_MEMORY_SIZE
Size of physical RAM (same as statistics counter `GlobalMemory.Memory.ram_size`)

SYSTEM_MEMORY_FREE_SIZE
Size of free RAM according to operating system

SHARED_MEMORY_ALLOCATED_SIZE
Size of allocated memory in row store: shared memory segments + MemMonitor Footprint

SHARED_MEMORY_USED_SIZE
Size of used memory in row store: shared memory segments + MemMonitor Used

SHARED_MEMORY_FREE_SIZE
Size of free space in row store: MemMonitor Free

PROCESS_ALLOCATION_LIMIT
Process allocation limit, usually 90% of physical RAM (same as statistics counter `GlobalMemory.Memory.process_allocation_limit`)

EFFECTIVE_PROCESS_ALLOCATION_LIMIT
Current limit respecting current usage of other processes

HEAP_MEMORY_ALLOCATED_SIZE
Allocated size by allocators (including overhead and free lists) and thread callstacks (same as statistics counter `GlobalMemory.Memory.allocated_size`)

HEAP_MEMORY_USED_SIZE

Size used by clients of allocators (new, malloc etc.) (same as statistics counter `Memory.Usage.INCLUSIVE_SIZE_IN_USE` and same as `INCLUSIVE_SIZE_IN_USE` of category "/" in view **M_HEAP_MEMORY**)

HEAP_MEMORY_FREE_SIZE
= `HEAP_MEMORY_ALLOCATED_SIZE - HEAP_MEMORY_USED_SIZE`

HEAP_MEMORY_ROOT_ALLOCATED_SIZE
= `HEAP_MEMORY_USED_SIZE`

HEAP_MEMORY_ROOT_FREE_SIZE
Size of big block free list (same as statistics counter `GlobalMemory.Memory.bigblock_free_size`)

COMPACTORS_SIZE
Size of caches (ResourceContainer) memory management may shrink in case of low memory

COMPACTORS_FREEABLE_SIZE
Shrinkable size of caches

TOTAL_MEMORY_SIZE_IN_USE
= `PROCESS_CODE_SIZE + SHARED_MEMORY_ALLOCATED_SIZE + HEAP_MEMORY_ROOT_ALLOCATED_SIZE`

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
NAME	VARCHAR(256)		Type of memory
VALUE	BIGINT	Byte	Size of memory

M_MEMORY_OBJECTS

Short description

Memory object statistics

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_MEMORY_OBJECTS_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_MEMORY_OBJECTS_RESET
```

or the management console command:

```
monitor reset -n M_MEMORY_OBJECTS_RESET
```

Full description

This view provides information about the number and size of resources currently in the resource container and about the throughput of the resource container. One row in this view represents one resource type (the type is specified by the resource statistic). The resource statistics are kept in a tree data structue and this view represents this tree in a human-readable form. Each value (except for HOST, PORT, VOLUME_ID, STATISTICS_NAME) is the AGGREGATED value of the subtree (including the current node).

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
TYPE	VARCHAR(128)		Object (statistic) type
OBJECT_COUNT	BIGINT	Counter	Number of objects currently in memory object container
OBJECT_SIZE	BIGINT	Byte	Size of objects currently in memory object container
NON_SWAPPABLE_SIZE	BIGINT	Byte	Size of non swappable objects currently in memory object container
SWAPPABLE_SIZE	BIGINT	Byte	Size of swappable objects currently in memory object container
PUT_COUNT	BIGINT	Counter	Total number of put objects
PUT_SIZE	BIGINT	Byte	Total size of put objects
EVICT_COUNT	BIGINT	Counter	Total number of evicted objects
EVICT_SIZE	BIGINT	Byte	Total size of evicted objects
TEMP_EVICT_COUNT	BIGINT	Counter	Total number of temp-evicted objects
TEMP_EVICT_SIZE	BIGINT	Byte	Total size of temp-evicted objects
RESIZE_COUNT	BIGINT	Counter	Total number of resizes
RESIZE_DELTA_SIZE	BIGINT	Byte	Total size delta changed by resize of objects

SHRINK_COUNT	BIGINT	Counter	Total number of shrunk objects
SHRINK_SIZE	BIGINT	Byte	Total size of shrunk objects
FAILED_SHRINK_COUNT	BIGINT	Counter	Total number of objects shrink failed to remove
FAILED_SHRINK_SIZE	BIGINT	Byte	Total size of objects shrink failed to remove

M_MEMORY_OBJECTS_RESET

Short description

Memory object statistics (since last reset)

This view contains values accumulated since the last reset of the main view **M_MEMORY_OBJECTS**. Please refer to **M_MEMORY_OBJECTS** for information about the structure and use of this view.

In addition to the members mentioned in **M_MEMORY_OBJECTS**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_MEMORY_OBJECT_DISPOSITIONS

Short description

Disposition specific memory object statistics. The statistics are calculated and reading them may take a while.

Full description

The number and the size of resources in the resource container are shown depending on their specific disposition (whether the memory objects are short, mid, long-term or non-swappable). For each type of resource (specified by the resource statistic), which is currently in the resource container, one row is added to this view. The tree structure of the resource statistics is not considered here, therefore the values are not aggregated.

Caution: Reading this view may take some time as the entire resource container must be traversed to generate this view.

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
TYPE	VARCHAR(128)		Object (statistic) type)
TEMPORARY_OBJECT_COUNT	BIGINT	Counter	Number of temporary objects
INTERNAL_SHORT_TERM_OBJECT_COUNT			
	BIGINT	Counter	Number of internal short term objects
SHORT_TERM_OBJECT_COUNT	BIGINT	Counter	Number of short term objects
MID_TERM_OBJECT_COUNT	BIGINT	Counter	Number of mid term objects
LONG_TERM_OBJECT_COUNT	BIGINT	Counter	Number of long term objects
NON_SWAPPABLE_OBJECT_COUNT			
	BIGINT	Counter	Number of non swappable objects
TEMPORARY_OBJECT_SIZE	BIGINT	Byte	Size of temporary objects
INTERNAL_SHORT_TERM_OBJECT_SIZE			
	BIGINT	Byte	Size of internal short term objects
SHORT_TERM_OBJECT_SIZE	BIGINT	Byte	Size of short term objects
MID_TERM_OBJECT_SIZE	BIGINT	Byte	Size of mid term objects
LONG_TERM_OBJECT_SIZE	BIGINT	Byte	Size of long term objects
NON_SWAPPABLE_OBJECT_SIZE	BIGINT	Byte	Size of non swappable objects

M_MERGED_TRACES

Short description

Contains the merged content of all trace files

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host on which server process ran
PORT	BIGINT		Communication port of server process
USER_NAME	NVARCHAR(256)		Database user
APPLICATION_USER_NAME	NVARCHAR(256)		Application user
SERVICE_NAME	VARCHAR(256)		Name of server process
CONNECTION_ID	INTEGER		Connection ID
TRANSACTION_ID	BIGINT		Transaction ID
THREAD_ID	BIGINT		ID of thread that wrote trace entry
TIMESTAMP	TIMESTAMP		Time when trace entry was written
TRACE_LEVEL	VARCHAR(8)		Trace level
COMPONENT	VARCHAR(32)		Trace component
SOURCE_FILE_NAME	VARCHAR(512)		Source file which contains trace
SOURCE_FILE_LINE	BIGINT		Source file line
TRACE_TEXT	NCLOB		Traced text
TRACE_FILE_NAME	VARCHAR(256)		Name of trace file containing trace entry
TRACE_FILE_LINE	INTEGER		Trace file line
PASSPORT_ROOT_CONTEXT_ID	VARCHAR(32)	Guid	SAP EPP Passport GUID identifying source of request
PASSPORT_TRANSACTION_ID	VARCHAR(32)		SAP EPP Passport GUID identifying business transaction
PASSPORT_CONNECTION_ID	VARCHAR(32)	Guid	SAP EPP Passport GUID identifying connection
PASSPORT_CONNECTION_COUNTER			
	BIGINT		SAP EPP Passport connection counter
PASSPORT_COMPONENT_NAME	NVARCHAR(32)		SAP EPP Passport component name of initial/root context
PASSPORT_ACTION	NVARCHAR(40)		SAP EPP Passport action

M_MONITORS

Short description

Available monitoring views

Full description

This view can be used in conjunction with **M_MONITOR_COLUMNS** to get information about existing monitoring views.

See also:

M_MONITOR_COLUMNS

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(5000)		Schema name
VIEW_NAME	NVARCHAR(5000)		Name of monitoring view
DESCRIPTION	NVARCHAR(5000)		Short description of the monitoring view
RESETTABLE	VARCHAR(5000)		Set to TRUE, if values of this monitor can be reset

M_MONITOR_COLUMNS

Short description

All the columns in the monitoring views

Full description

This view can be used in conjunction with **M_MONITORS** to get information about existing monitoring views.

See also:

M_MONITORS

Structure

Column name	Data type	Unit	Description
VIEW_NAME	NVARCHAR(256)		Name of monitoring view
VIEW_COLUMN_NAME	NVARCHAR(256)		Name of monitoring view column
DATA_TYPE_ID	SMALLINT		Column data type ID
DATA_TYPE_NAME	VARCHAR(16)		Column data type name
POSITION	INTEGER		Ordinal position of the column in a record
DEFAULT_VALUE	NVARCHAR(5000)		Default value
UNIT	VARCHAR(128)		Unit for the value
COLLATION	NVARCHAR(256)		Collation
LENGTH	DECIMAL(34,-1)		Length
SCALE	INTEGER		Scale
IS_NULLABLE	VARCHAR(5000)		TRUE, if NULL value is allowed
DESCRIPTION	NVARCHAR(5000)		Short description of the monitoring view column

M_MUTEXES

Short description

Mutex statistics

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_MUTEXES_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_MUTEXES_RESET
```

or the management console command:

```
monitor reset -n M_MUTEXES_RESET
```

Full description

This view contains information about single mutex (mutual exclusion) objects or groups of mutex objects. It does not contain information about all mutex objects. Information like LOCK_COUNT, WAIT_COUNT and WAIT_TIMES can be used to anaylse performance bottlenecks. Using OWNER_ID you can find possible deadlocks.

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
STATISTICS_NAME	VARCHAR(128)		Statistics object name
STATISTICS_ID	BIGINT		Statistics object unique ID
LOCK_COUNT	BIGINT	Counter	Count of lock calls
WAIT_COUNT	BIGINT	Counter	Count of blocking lock calls
SPURIOUS_WAKEUP_COUNT	BIGINT	Counter	Count of spurious wakeups (collisions on futex)
COLLISION_RATE	DOUBLE	Percent	Collision rate in percent
OWNER_ID	BIGINT		Context ID of the owner context
LAST_WAIT_TIME	BIGINT	Microsecond	Time of blocking lock calls (last)
MAX_WAIT_TIME	BIGINT	Microsecond	Time of blocking lock calls (max)
MIN_WAIT_TIME	BIGINT	Microsecond	Time of blocking lock calls (min)
SUM_WAIT_TIME	BIGINT	Microsecond	Time of blocking lock calls (total)
AVG_WAIT_TIME	BIGINT	Microsecond	Time of blocking lock calls (avg)
CREATE_COUNT	BIGINT	Counter	Count of mutex creation (for shared statistics only)
DESTROY_COUNT	BIGINT	Counter	Count of mutex destruction (for shared statistics only)

M_MUTEXES_RESET

Short description

Mutex statistics (since last reset)

This view contains values accumulated since the last reset of the main view **M_MUTEXES**. Please refer to **M_MUTEXES** for information about the structure and use of this view.

In addition to the members mentioned in **M_MUTEXES**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_MVCC_TABLES

Short description

Statistics of row-store Multiversion Concurrency Control (MVCC) manager

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
NAME	VARCHAR(256)		Name column
VALUE	VARCHAR(128)		Value column

M_PAGEACCESS_STATISTICS

Short description

PageAccess statistics

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_PAGEACCESS_STATISTICS_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_PAGEACCESS_STATISTICS_RESET
```

or the management console command:

```
monitor reset -n M_PAGEACCESS_STATISTICS_RESET
```

Full description

This view contains information about pages accessed. TYPE specifies the pageaccess. The *_COUNT values are the number of the respective operations on the page access. CHUNK_SIZE is the number of pages accessed by one operation (CHUNK_SIZE is 1 except for RowStorePageAccess where blocks of pages are accessed. For example, LOAD_COUNT=4 and CHUNK_SIZE=1024 means that 2*1024=4096 pages had been loaded). On allocation / deallocation the corresponding Converter is accessed.

See also:

M_CONVERTER_STATISTICS

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
TYPE	VARCHAR(32)		Type of pageaccess
CHUNK_SIZE	INTEGER	Counter	Chunk size
ALLOCATE_COUNT	BIGINT	Counter	Number of allocations
GET_COUNT	BIGINT	Counter	Number of gets
LOAD_COUNT	BIGINT	Counter	Number of synchronous loads
TRIGGER_LOAD_COUNT	BIGINT	Counter	Number of asynchronous loads
DEALLOCATE_COUNT	BIGINT	Counter	Number of deallocations

M_PAGEACCESS_STATISTICS_RESET

Short description

PageAccess statistics (since last reset)

This view contains values accumulated since the last reset of the main view **M_PAGEACCESS_STATISTICS**. Please refer to **M_PAGEACCESS_STATISTICS** for information about the structure and use of this view.

In addition to the members mentioned in **M_PAGEACCESS_STATISTICS**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_PASSWORD_POLICY

Short description

Configuration values concerning password policy

Structure

Column name	Data type	Unit	Description
PROPERTY	VARCHAR(128)		Configuration property
VALUE	VARCHAR(5000)		Value

M_PERFTRACE

Short description

Current PerfTrace state

Full description

The performance trace (PerfTrace) displays detailed information about query execution.

Performance Trace calls

Start Performance Trace. It can be restricted to a specific SQL and application user. Optionally plan execution details can be recorded. If duration is specified it automatically stops after ... seconds.

```
ALTER SYSTEM START PERFTRACE [USER name] [APPLICATIONUSER name] [PLAN_EXECUTION] [FU
NCTION_PROFILER] [DURATION seconds]
```

Stop Performance Trace

```
ALTER SYSTEM STOP PERFTRACE
```

Collect Performance Trace data and save into a file. The file is located on the server in the trace directory. If no file name is given 'perftrace.tpt' is used.

```
ALTER SYSTEM SAVE PERFTRACE [INTO FILE 'filename']
```

Structure

Column name	Data type	Unit	Description
STATUS	VARCHAR(256)		Status: STOPPED, STARTED, SAVING
START_TIME	TIMESTAMP		Start time
STOP_TIME	TIMESTAMP		Stop time
FILE_SIZE	BIGINT	Byte	Size of collected trace files. Only valid if stopped and not saved
REMAINING_SECONDS	INTEGER	Second	Remaining seconds until automatic stop
USER_NAME	NVARCHAR(256)		SQL user name filter
APPLICATION_USER_NAME	NVARCHAR(256)		Application user name filter
PLAN_EXECUTION	VARCHAR(6)	Boolean	Flag if plan execution details are recorded
FUNCTION_PROFILER	VARCHAR(6)	Boolean	Flag if function profiler details are recorded

M_PERSISTENCE_MANAGERS

Short description

Persistence manager statistics

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_PERSISTENCE_MANAGERS_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_PERSISTENCE_MANAGERS_RESET
```

or the management console command:

```
monitor reset -n M_PERSISTENCE_MANAGERS_RESET
```

Full description

Persistence manager is the module responsible for low-level operations on the persistent data structures. This view shows various statistics counters, which are used to measure the performance of those operations.

See also:

M_GARBAGE_COLLECTION_STATISTICS

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
MAX_TID	BIGINT	TID	Maximum known TID
CCH_COUNT	BIGINT	Counter	Count of normal consistent changes (terminated sessions)
MASS_CCH_COUNT	BIGINT	Counter	Count of consistent changes for mass operations (terminated sessions)
LAST_CCH_TIME	BIGINT	Microsecond	Time spent in normal consistent changes (terminated sessions) (last)
MAX_CCH_TIME	BIGINT	Microsecond	Time spent in normal consistent changes (terminated sessions) (max)
MIN_CCH_TIME	BIGINT	Microsecond	Time spent in normal consistent changes (terminated sessions) (min)
SUM_CCH_TIME	BIGINT	Microsecond	Time spent in normal consistent changes (terminated sessions) (total)
AVG_CCH_TIME	BIGINT	Microsecond	Time spent in normal consistent changes (terminated sessions) (avg)
LAST_MASS_CCH_TIME	BIGINT	Microsecond	Time spent in consistent changes for mass

			operations (terminated sessions) (last)
MAX_MASS_CCH_TIME	BIGINT	Microsecond	Time spent in consistent changes for mass operations (terminated sessions) (max)
MIN_MASS_CCH_TIME	BIGINT	Microsecond	Time spent in consistent changes for mass operations (terminated sessions) (min)
SUM_MASS_CCH_TIME	BIGINT	Microsecond	Time spent in consistent changes for mass operations (terminated sessions) (total)
AVG_MASS_CCH_TIME	BIGINT	Microsecond	Time spent in consistent changes for mass operations (terminated sessions) (avg)
PREPARE_COMMIT_POS	BIGINT		Newest known log position of prepare commit (slave only)
MASTER_COMMIT_POS	BIGINT		Newest known log position of commit record on transaction master from the point of view of this node
INDOUBT_WAITERS	BIGINT		Count of in-doubt waiters for currently-running COMMIT (master only)
INDOUBT_RESTART_COUNT	BIGINT	Counter	Count of open in-doubt transactions before restart (master only)
INDOUBT_ONLINE_COUNT	BIGINT	Counter	Count of open in-doubt transactions since last restart (master only)
SAVEPOINT_CONFIG_FREQUENCY			
	BIGINT	Second	Configured savepoint frequency
SAVEPOINT_ACTIVE_FREQUENCY			
	BIGINT	Second	Currently active savepoint frequency

M_PERSISTENCE_MANAGERS_RESET

Short description

Persistence manager statistics (since last reset)

This view contains values accumulated since the last reset of the main view **M_PERSISTENCE_MANAGERS**. Please refer to **M_PERSISTENCE_MANAGERS** for information about the structure and use of this view.

In addition to the members mentioned in **M_PERSISTENCE_MANAGERS**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_PREPARED_STATEMENTS

Short description

Prepared statements list

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
CONNECTION_ID	INTEGER		Connection ID
STATEMENT_ID	VARCHAR(256)		Prepared statement ID
START_MVCC_TIMESTAMP	BIGINT		Internal MVCC timestamp of the transaction start time
COMPILED_TIME	TIMESTAMP		Compilation timestamp of the statement
STATEMENT_STATUS	VARCHAR(128)		Status of SQL statement
STATEMENT_STRING	NCLOB		SQL statement
ALLOCATED_MEMORY_SIZE	BIGINT	Byte	Allocated memory size
USED_MEMORY_SIZE	BIGINT	Byte	Used memory size
PLAN_ID	BIGINT		Logical plan ID
LAST_EXECUTED_TIME	TIMESTAMP		Last execution timestamp of the statement
RECOMPILE_COUNT	BIGINT		Recompile count
EXECUTION_COUNT	BIGINT		Count of executions
AVG_EXECUTION_TIME	BIGINT		Average time of statement execution
MAX_EXECUTION_TIME	BIGINT		Maximum time of statement execution
MIN_EXECUTION_TIME	BIGINT		Minimum time of statement execution
TOTAL_EXECUTION_TIME	BIGINT		Sum of time of statement execution
AVG_CURSOR_DURATION	BIGINT		Average time of statement execution including communicate time with clients
MAX_CURSOR_DURATION	BIGINT		Maximum time of statement execution including communicate time with clients
MIN_CURSOR_DURATION	BIGINT		Minimum time of statement execution including communicate time with clients
TOTAL_CURSOR_DURATION	BIGINT		Sum of time of statement execution including communicate time with clients
AVG_EXECUTION_MEMORY_SIZE	BIGINT		Average memory size used during each execution
MAX_EXECUTION_MEMORY_SIZE	BIGINT		Maximum memory size used during each execution
MIN_EXECUTION_MEMORY_SIZE	BIGINT		Minimum memory size used during each execution
TOTAL_EXECUTION_MEMORY_SIZE			
	BIGINT		Sum of memory size used during each execution
AVG_LOCKWAIT_TIME	BIGINT		Average lock wait time for the statement

MAX_LOCKWAIT_TIME	BIGINT		Maximum lock wait time for the statement
MIN_LOCKWAIT_TIME	BIGINT		Minimum lock wait time for the statement
TOTAL_LOCKWAIT_COUNT	BIGINT		Total lock wait count for the statement
TOTAL_LOCKWAIT_TIME	BIGINT		Accumulated lock wait time for the statement
AVG_PREPARATION_TIME	BIGINT		Average time of statement preparation
MAX_PREPARATION_TIME	BIGINT		Maximum time of statement preparation
MIN_PREPARATION_TIME	BIGINT		Minimum time of statement preparation
TOTAL_PREPARATION_TIME	BIGINT		Total time of statement preparation
TOTAL_PREPARATION_COUNT	BIGINT		Total count of statement preparation
HAS_HOLDABLE_CURSOR	VARCHAR(6)		Holdable cursor existence
PARENT_STATEMENT_ID	VARCHAR(256)		Parent prepared statement ID

M_READWRITELOCKS

Short description

Read/Write lock statistics

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_READWRITELOCKS_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_READWRITELOCKS_RESET
```

or the management console command:

```
monitor reset -n M_READWRITELOCKS_RESET
```

Full description

This view contains information about single reader/writer lock objects or groups of reader/writer lock objects. It does not contain information about all reader/writer lock. Information like LOCK_COUNT, WAIT_COUNT and WAIT_TIMES can be used to anaylse performance bottlenecks.

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
STATISTICS_NAME	VARCHAR(128)		Statistics object name
STATISTICS_ID	BIGINT		Statistics object unique ID
OWNER_ID	BIGINT		Context ID of the owner context (for exclusive/intent locks)
EXCLUSIVE_LOCK_COUNT	BIGINT	Counter	Count of exclusive lock calls
EXCLUSIVE_WAIT_COUNT	BIGINT	Counter	Count of blocking exclusive lock calls
EXCLUSIVE_CAS_COLLISION_COUNT			
	BIGINT	Counter	Collision count on atomic operation on exclusive lock
EXCLUSIVE_COLLISION_RATE	DOUBLE	Percent	Collision rate on exclusive lock in percent
LAST_EXCLUSIVE_WAIT_TIME	BIGINT	Microsecond	Time of blocking exclusive lock calls (last)
MAX_EXCLUSIVE_WAIT_TIME	BIGINT	Microsecond	Time of blocking exclusive lock calls (max)
MIN_EXCLUSIVE_WAIT_TIME	BIGINT	Microsecond	Time of blocking exclusive lock calls (min)
SUM_EXCLUSIVE_WAIT_TIME	BIGINT	Microsecond	Time of blocking exclusive lock calls (total)
AVG_EXCLUSIVE_WAIT_TIME	BIGINT	Microsecond	Time of blocking exclusive lock calls (avg)
INTENT_LOCK_COUNT	BIGINT	Counter	Count of intent lock calls
INTENT_WAIT_COUNT	BIGINT	Counter	Count of blocking intent lock calls

INTENT_CAS_COLLISION_COUNT			
	BIGINT	Counter	Collision count on atomic operation on intent lock
INTENT_TIMEOUT_COUNT	BIGINT	Counter	Count of timed out intent lock calls
INTENT_COLLISION_RATE	DOUBLE	Percent	Collision rate on intent lock in percent
LAST_INTENT_WAIT_TIME	BIGINT	Microsecond	Time of blocking intent lock calls (last)
MAX_INTENT_WAIT_TIME	BIGINT	Microsecond	Time of blocking intent lock calls (max)
MIN_INTENT_WAIT_TIME	BIGINT	Microsecond	Time of blocking intent lock calls (min)
SUM_INTENT_WAIT_TIME	BIGINT	Microsecond	Time of blocking intent lock calls (total)
AVG_INTENT_WAIT_TIME	BIGINT	Microsecond	Time of blocking intent lock calls (avg)
SHARED_LOCK_COUNT	BIGINT	Counter	Count of shared lock calls
SHARED_WAIT_COUNT	BIGINT	Counter	Count of blocking shared lock calls
SHARED_CAS_COLLISION_COUNT			
	BIGINT	Counter	Collision count on atomic operation on shared lock
SHARED_TIMEOUT_COUNT	BIGINT	Counter	Count of timed out shared lock calls
SHARED_COLLISION_RATE	DOUBLE	Percent	Collision rate on shared lock in percent
LAST_SHARED_WAIT_TIME	BIGINT	Microsecond	Time of blocking shared lock calls (last)
MAX_SHARED_WAIT_TIME	BIGINT	Microsecond	Time of blocking shared lock calls (max)
MIN_SHARED_WAIT_TIME	BIGINT	Microsecond	Time of blocking shared lock calls (min)
SUM_SHARED_WAIT_TIME	BIGINT	Microsecond	Time of blocking shared lock calls (total)
AVG_SHARED_WAIT_TIME	BIGINT	Microsecond	Time of blocking shared lock calls (avg)
COLLISION_RATE	DOUBLE	Percent	Global collision rate
CREATE_COUNT	BIGINT	Counter	Count of read/write lock creation (for shared statistics only)
DESTROY_COUNT	BIGINT	Counter	Count of read/write lock destruction (for shared statistics only)

M_READWRITELOCKS_RESET

Short description

Read/Write lock statistics (since last reset)

This view contains values accumulated since the last reset of the main view **M_READWRITELOCKS**. Please refer to **M_READWRITELOCKS** for information about the structure and use of this view.

In addition to the members mentioned in **M_READWRITELOCKS**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_RECORD_LOCKS

Short description

Record lock status

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
LOCK_OWNER_TRANSACTION_ID	INTEGER		Transaction ID of the lock owner
ACQUIRED_TIME	TIMESTAMP		Lock acquisition time
RECORD_ID	VARCHAR(256)		Record ID
SCHEMA_NAME	NVARCHAR(256)		Schema name
TABLE_NAME	NVARCHAR(256)		Table name
LOCK_MODE	VARCHAR(32)		Lock mode:I (Exclusive Lock)

M_RS_INDEXES

Short description

Statistics of B-tree and CPB-tree indexes

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
SCHEMA_NAME	NVARCHAR(256)		Schema name
TABLE_NAME	NVARCHAR(256)		Table name
INDEX_NAME	NVARCHAR(256)		Index name
KEY_TYPE	VARCHAR(128)		Key type (data type or composite)
INDEX_STATUS	VARCHAR(128)		Index status: 'VALID', 'INVALID', 'UNUSABLE'
TREE_HEIGHT	BIGINT		B-tree level
LEAF_NODE_COUNT	BIGINT		Number of leaf nodes
NONLEAF_NODE_COUNT	BIGINT		Number of intermediate nodes
NODE_SIZE	BIGINT	Byte	Node size
FANOUT	DOUBLE		Index fan-out
BULKLOAD_FACTOR	DOUBLE		Fill factor of creating or recovering indexes
INDEX_SIZE	BIGINT	Byte	Index size in memory = number of nodes * node size
ENTRY_COUNT	BIGINT		Number of indexed records
FIXED_LEAF_NODE_COUNT	BIGINT		Number of fixed leaf nodes. 'FIXED' means that the key lengths of all entries in the node have the same value
FIXED_NONLEAF_NODE_COUNT	BIGINT		Number of fixed non-leaf nodes
AVG_LEAF_OFFSET_SIZE	DOUBLE		Average offset size of leaf nodes
AVG_NONLEAF_OFFSET_SIZE	DOUBLE		Average offset size of non-leaf nodes
AVG_LEAF_POINTER_SIZE	DOUBLE		Average pointer size of leaf nodes
AVG_NONLEAF_POINTER_SIZE	DOUBLE		Average pointer size of non-leaf nodes
LEAF_PARTIAL_KEY_SIZE	BIGINT		Partial key length of leaf node
NONLEAF_PARTIAL_KEY_SIZE	BIGINT		Partial key length of non-leaf node
INDEX_UTILIZATION	DOUBLE		Index utilization (num_entries / max_entries)
IS_UNIQUE	VARCHAR(6)		Information whether the index is unique: 'TRUE', 'FALSE'
SEARCH_COUNT	BIGINT		Number of search operations on this CPB-tree index since monitoring level was changed to 2
INSERT_COUNT	BIGINT		Number of insert operations on this CPB-tree index since monitoring level was changed to 2

REMOVE_COUNT	BIGINT		Number of remove operations on this CPB-tree index since monitoring level was changed to 2
FULL_KEY_REFERENCE_COUNT	BIGINT		Number of references of full keys on this CPB-tree index since monitoring level was changed to 2. Full key references occur when keys cannot be resolved by partial keys in CPB-tree nodes
DISTINCT_KEY_COUNT	BIGINT		Number of distinct keys
KEY_COUNT	BIGINT		Number of keys
ELIMINATED_DUPLICATE_LEAF_NODE_COUNT			
	BIGINT		Number of leaf nodes whose duplicate keys are eliminated
UNUSED_LEAF_SLOTS_PER_NODE			
	DOUBLE		Number of unused slots per leaf node

M_RS_TABLES

Short description

Information on row tables: detailed table sizes and record count

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
SCHEMA_NAME	NVARCHAR(256)		Schema name
TABLE_NAME	NVARCHAR(256)		Table name
RECORD_COUNT	BIGINT		Number of records in this table
ALLOCATED_FIXED_PART_SIZE	BIGINT	Byte	Allocated memory size for fixed-size part
FIXED_PAGE_HEADER_SIZE	BIGINT	Byte	Allocated and used memory size for page headers of fixed-size part
FIXED_PAGE_FRAGMENT_SIZE	BIGINT	Byte	Fragmented memory size of fixed-size part
USED_FIXED_PART_SIZE	BIGINT	Byte	Used memory size for fixed-size part
FIXED_PART_FRAGMENT_SIZE	BIGINT	Byte	Fragmented memory size of used fixed-size part
FIXED_PART_FREE_SIZE	BIGINT	Byte	Free memory size in the pages of fixed-size part
ALLOCATED_VARIABLE_PART_SIZE			
	BIGINT	Byte	Allocated memory size for variable-size part
USED_VARIABLE_PART_SIZE	BIGINT	Byte	Used memory size for variable-size part
VARIABLE_PART_FRAGMENT_SIZE			
	BIGINT	Byte	Fragmented memory size of used variable-size part
LOAD_STATUS	NVARCHAR(256)		Loading state of table: 'LOADED', 'NOW_LOADING', 'PREPARING_UNLOAD', 'NOW_UNLOADING', 'UNLOADED', 'NOT_SUPPORTED'

M_SAVEPOINT_STATISTICS

Short description

Savepoint statistics

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_SAVEPOINT_STATISTICS_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_SAVEPOINT_STATISTICS_RESET
```

or the management console command:

```
monitor reset -n M_SAVEPOINT_STATISTICS_RESET
```

Full description

This view shows information about executed savepoints.

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
SAVEPOINTS	BIGINT	Counter	Count of executed savepoints
LAST_SAVEPOINT_TIME	TIMESTAMP		Last savepoint time (wall clock)
REQUESTED_FREQUENCY	BIGINT	Second	Currently active requested savepoint frequency
LAST_FREQUENCY	BIGINT	Second	Actual savepoint frequency (time between last two savepoints) (last)
AVG_FREQUENCY	BIGINT	Second	Actual savepoint frequency (time between last two savepoints) (avg)
LAST_DURATION	BIGINT	Microsecond	Time spent in savepoint (last)
AVG_DURATION	BIGINT	Microsecond	Time spent in savepoint (avg)
LAST_FLUSHED_CONV_PAGES	BIGINT	Counter	Count of asynchronously flushed converter pages (last)
AVG_FLUSHED_CONV_PAGES	DOUBLE	Rate	Count of asynchronously flushed converter pages (avg)
LAST_FLUSHED_PAGES	BIGINT	Counter	Count of asynchronously flushed logical pages (last)
AVG_FLUSHED_PAGES	DOUBLE	Rate	Count of asynchronously flushed logical pages (avg)
LAST_FLUSHED_CONV_PAGES_IN_CS			
	BIGINT	Counter	Count of converter pages flushed in critical section (last)

AVG_FLUSHED_CONV_PAGES_IN_CS			
	DOUBLE	Rate	Count of converter pages flushed in critical section (avg)
LAST_FLUSHED_PAGES_IN_CS	BIGINT	Counter	Count of logical pages flushed in critical section (last)
AVG_FLUSHED_PAGES_IN_CS	DOUBLE	Rate	Count of logical pages flushed in critical section (avg)
LAST_FLUSHED_ROWSTORE_CONV_PAGES			
	BIGINT	Counter	Count of absolute converter pages flushed (last)
AVG_FLUSHED_ROWSTORE_CONV_PAGES			
	DOUBLE	Rate	Count of absolute converter pages flushed (avg)
LAST_FLUSHED_ROWSTORE_PAGES			
	BIGINT	Counter	Count of absolute logical pages flushed (last)
AVG_FLUSHED_ROWSTORE_PAGES			
	DOUBLE	Rate	Count of absolute logical pages flushed (avg)
LAST_RTT_SIZE	BIGINT	Counter	Total size of rollback transaction table at last savepoint (might be less than sum of components due to duplicate TIDs)

M_SAVEPOINT_STATISTICS_RESET

Short description

Savepoint statistics (since last reset)

This view contains values accumulated since the last reset of the main view **M_SAVEPOINT_STATISTICS**. Please refer to **M_SAVEPOINT_STATISTICS** for information about the structure and use of this view.

In addition to the members mentioned in **M_SAVEPOINT_STATISTICS**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_SEMAPHORES

Short description

Semaphore statistics

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_SEMAPHORES_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_SEMAPHORES_RESET
```

or the management console command:

```
monitor reset -n M_SEMAPHORES_RESET
```

Full description

This view contains information about single semaphore objects or groups of semaphore objects. It does not contain information about all semaphores.

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
STATISTICS_NAME	VARCHAR(128)		Statistics object name
STATISTICS_ID	BIGINT		Statistics object unique ID
WAIT_COUNT	BIGINT	Counter	Count of wait calls
BLOCKING_COUNT	BIGINT	Counter	Count of blocking wait calls
TIMEOUT_COUNT	BIGINT	Counter	Count of timeouts
WAIT_RATE	DOUBLE	Percent	Wait rate
LAST_BLOCKING_TIME	BIGINT	Microsecond	Time of blocking wait calls (last)
MAX_BLOCKING_TIME	BIGINT	Microsecond	Time of blocking wait calls (max)
MIN_BLOCKING_TIME	BIGINT	Microsecond	Time of blocking wait calls (min)
SUM_BLOCKING_TIME	BIGINT	Microsecond	Time of blocking wait calls (total)
AVG_BLOCKING_TIME	BIGINT	Microsecond	Time of blocking wait calls (avg)
CREATE_COUNT	BIGINT	Counter	Count of semaphore creation (for shared statistics only)
DESTROY_COUNT	BIGINT	Counter	Count of semaphore destruction (for shared statistics only)

M_SEMAPHORES_RESET

Short description

Semaphore statistics (since last reset)

This view contains values accumulated since the last reset of the main view **M_SEMAPHORES**. Please refer to **M_SEMAPHORES** for information about the structure and use of this view.

In addition to the members mentioned in **M_SEMAPHORES**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_SERVICES

Short description

Status of all services

Full description

Service Calls:

Stop a service

```
ALTER SYSTEM STOP SERVICE (host,port)
```

Kill a service

```
ALTER SYSTEM KILL SERVICE (host,port) [WITH COREFILE]
```

Reconfigure a service

```
ALTER SYSTEM RECONFIGURE SERVICE ('service', 'host', port)
```

Depending on service, host, port, different services can be reconfigured:

Service	Host	Port	Description
n.a.	not empty	not 0	Reconfigure single service on host y with port z
not empty	not empty	0	Reconfigure all services of type x on host y
not empty	empty	0	Reconfigure all services of type x
empty	empty	0	Reconfigure all services

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
SERVICE_NAME	VARCHAR(256)		Service name. See M_SERVICE_TYPES for all known service names
PROCESS_ID	INTEGER		Process ID
DETAIL	VARCHAR(128)		Only for display. Use COORDINATOR_TYPE to test service role
ACTIVE_STATUS	VARCHAR(128)		NO, YES, UNKNOWN, STARTING, STOPPING
SQL_PORT	INTEGER		SQL port
COORDINATOR_TYPE	VARCHAR(128)		Coordinator type in distributed landscape: MASTER, SLAVE, STANDBY, NONE

M_SERVICE_MEMORY

Short description

Detailed information on memory utilization by services

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
SERVICE_NAME	VARCHAR(256)		Service name
PROCESS_ID	INTEGER		Process ID
LOGICAL_MEMORY_SIZE	BIGINT	Byte	Process virtual memory size
PHYSICAL_MEMORY_SIZE	BIGINT	Byte	Process physical memory size
CODE_SIZE	BIGINT	Byte	Process code size
STACK_SIZE	BIGINT	Byte	Process stack size
HEAP_MEMORY_ALLOCATED_SIZE			
	BIGINT	Byte	Process heap memory allocated size
HEAP_MEMORY_USED_SIZE	BIGINT	Byte	Process heap memory used
SHARED_MEMORY_ALLOCATED_SIZE			
	BIGINT	Byte	Process shared memory allocated (row store only)
SHARED_MEMORY_USED_SIZE	BIGINT	Byte	Process shared memory used (row store only)
COMPACTORS_ALLOCATED_SIZE	BIGINT	Byte	Total memory allocated by compactors
COMPACTORS_FREEABLE_SIZE	BIGINT	Byte	Memory which can be freed on demand

M_SERVICE_NETWORK_IO

Short description

Service network I/O statistics

Structure

Column name	Data type	Unit	Description
SENDER_HOST	VARCHAR(64)		Host name of the sending service
SENDER_PORT	INTEGER		Port that the sending service listens on
RECEIVER_HOST	VARCHAR(64)		Host name of the receiving service
RECEIVER_PORT	INTEGER		Port that the receiving service listens on
SEND_SIZE	BIGINT	Byte	Number of bytes sent
RECEIVE_SIZE	BIGINT	Byte	Number of bytes received
SEND_DURATION	BIGINT	Microsecond	Time spent sending
RECEIVE_DURATION	BIGINT	Microsecond	Time spent receiving
REQUEST_COUNT	BIGINT	Counter	Number of requests handled

M_SERVICE_STATISTICS

Short description

Statistics on active services

Full description

ACTIVE

Value	Description
NO	Service not started.
STARTING	Service is starting. This state might occur for several minutes on first startup or recovery.
YES	Service started and ready for requests.
STOPPING	Service is stopping.
UNKNOWN	Initial state after starting a landscape or when state not known (e.g after crash). If the service does not start within a minute this state changes to NO.

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
SERVICE_NAME	VARCHAR(256)		Service name
PROCESS_ID	INTEGER		Process ID
DETAIL	VARCHAR(128)		Detail information, similar to COORDINATOR_TYPE in M_SERVICES
ACTIVE_STATUS	VARCHAR(128)		NO, YES, UNKNOWN, STARTING, STOPPING
START_TIME	TIMESTAMP		Process start time
PROCESS_CPU	SMALLINT	Percent	CPU usage of current process
PROCESS_CPU_TIME	BIGINT	Millisecond	CPU usage of current process since start
TOTAL_CPU	SMALLINT	Percent	CPU usage of all processes
PROCESS_MEMORY	BIGINT		Process logical memory usage
PROCESS_PHYSICAL_MEMORY	BIGINT		Process physical memory usage
TOTAL_MEMORY	BIGINT		Host physical+swap memory usage
AVAILABLE_MEMORY	BIGINT		Host physical+swap memory size
PHYSICAL_MEMORY	BIGINT		Host physical memory size
REQUESTS_PER_SEC	DOUBLE	Rate	Requests per second. Average over last 1000 requests
RESPONSE_TIME	INTEGER	Millisecond	Request response time. Average over last 1000 requests
FINISHED_NON_INTERNAL_REQUEST_COUNT			

	BIGINT	Counter	Finished requests
ALL_FINISHED_REQUEST_COUNT			
	BIGINT	Counter	Finished requests including internal requests
ACTIVE_REQUEST_COUNT	INTEGER		Number of active requests
PENDING_REQUEST_COUNT	INTEGER		Number of pending requests
ACTIVE_THREAD_COUNT	INTEGER		Number of active threads
THREAD_COUNT	INTEGER		Number of total threads
OPEN_FILE_COUNT	INTEGER		Number of open files

M_SERVICE_THREADS

Short description

Detailed information on threads created by services

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
SERVICE_NAME	VARCHAR(256)		Service name. See M_SERVICE_TYPES for all known service names
HIERARCHY	VARCHAR(128)		Thread grouping information. Filled with Connection-ID/(internal)Transaction-ID
CONNECTION_ID	INTEGER		Connection ID
THREAD_ID	BIGINT		Thread ID
THREAD_TYPE	VARCHAR(128)		Thread Type
THREAD_METHOD	VARCHAR(256)		Thread method
THREAD_DETAIL	NCLOB		Thread detail
DURATION	BIGINT	Millisecond	Thread Duration
CALLER	VARCHAR(256)		The service which called this thread
CALLING	VARCHAR(256)		The service which the thread calls
USER_NAME	NVARCHAR(256)		SQL user name
APPLICATION_USER_NAME	NVARCHAR(256)		Application user name
SYSTEM_CPU_TIME	BIGINT	Microsecond	System CPU time
USER_CPU_TIME	BIGINT	Microsecond	User CPU time

M_SERVICE_THREAD_CALLSTACKS

Short description

Stack frame information for service threads

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
SERVICE_NAME	VARCHAR(256)		Service name. See M_SERVICE_TYPES for all known service names
THREAD_ID	BIGINT		Thread ID
FRAME_LEVEL	INTEGER		Level of stack frame
FRAME_NAME	VARCHAR(256)		Name of stack frame (function, file, library, ...)

M_SERVICE_TRACES

Short description

Configured trace components for each service type

Structure

Column name	Data type	Unit	Description
SERVICE_NAME	VARCHAR(256)		Service name
COMPONENT_NAME	VARCHAR(32)		Trace component name

M_SERVICE_TYPES

Short description

Service types

Structure

Column name	Data type	Unit	Description
SERVICE_NAME	VARCHAR(256)		Service name
INIFILE	VARCHAR(256)		Configuration file name of service
HAS_DETAIL	VARCHAR(6)	Boolean	Service shows details in M_SERVICE_STATISTICS

M_SESSION_CONTEXT

Short description

Session variables for each connection

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
CONNECTION_ID	INTEGER		Connection ID
KEY	VARCHAR(32)		Key name of a session context variable
VALUE	VARCHAR(512)		Value of a session context variable

M_SHARED_MEMORY

Short description

Shared memory usage information by SAP HANA indexserver

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
CATEGORY	VARCHAR(128)		Module name
ALLOCATED_SIZE	BIGINT		Allocated shared memory size on the module
USED_SIZE	BIGINT		Used shared memory size on the module
FREE_SIZE	BIGINT		Free shared memory size on the module

M_SNAPSHOTS

Short description

Existing snapshots

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
ID	BIGINT		Snapshot ID
TIMESTAMP	TIMESTAMP		Creation time
FOR_BACKUP	VARCHAR(6)		Created for backup
ANCHOR	BIGINT	HEXID	Anchor

M_SQL_PLAN_CACHE

Short description

Statistics of an individual execution plan

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_SQL_PLAN_CACHE_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_SQL_PLAN_CACHE_RESET
```

or the management console command:

```
monitor reset -n M_SQL_PLAN_CACHE_RESET
```

Full description

- GLOBAL: A plan always can be shared with any other connections concurrently. No restriction.
- SESSION EXCLUSIVE GLOBAL: If a connection uses a plan, the plan cannot be shared with any other connections concurrently, but after the connection releases the plan, it can be acquired by another connection.
- SESSION LOCAL: A plan is never shared. This sharing type is more restricted than SESSION EXCLUSIVE GLOBAL. If a connection generates a plan, the plan belongs to the connection. This plan cannot be shared with any other connection and it cannot be also shared with any other cursor even if in the same connection.
- GLOBAL: -1 (always).
- SESSION EXCLUSIVE GLOBAL: -1 or owner connection ID with positive value (can be changed).
- SESSION LOCAL: owner connection ID with positive value (never changed).

The M_SQL_PLAN_CACHE view shows statistics for an individual plan but not all plans. It shows whether a specified plan runs longer than expected or which part of execution is dominant. For each cached plan, this view delivers statistics from execution on distributed configuration as well as technical details such as related object IDs, updated objects and so on.

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
STATEMENT_STRING	NCLOB		Statement string
USER_NAME	NVARCHAR(256)		User name who prepared the plan
SCHEMA_NAME	NVARCHAR(256)		Schema name that the SQL plan belongs to. SQL plans are generated in each schema even though the statement string is the same since the query optimizer statistics might be different

IS_VALID	VARCHAR(6)		TRUE if the plan is currently valid. A plan is invalidated whenever its corresponding schema objects, such as table and view, are changed. Invalidated plans are recompiled if the same statement is executed, otherwise they are evicted by another plan when newly compiled
IS_INTERNAL	VARCHAR(6)		TRUE if the plan is executed from a database internal connection. FALSE if it is executed from a remote connection
IS_DISTRIBUTED_EXECUTION	VARCHAR(6)		TRUE if tables are located in multi nodes, FALSE otherwise
ABAP_VARCHAR_MODE	VARCHAR(6)		TRUE if ABAP VARCHAR mode enabled, FALSE otherwise. ABAP application developers concern this mode, which indicates null terminated string or not
ACCESSED_OBJECTS	VARCHAR(2000)		Relevant table object ID list for the plan
TABLE_LOCATIONS	VARCHAR(2000)		Relevant table locations for the plan
TABLE_TYPES	VARCHAR(128)		Specification whether the plan refers Column store only, Row store only or mixed, displayed as one of three possible values 'ROW', 'COLUMN', 'ROW, COLUMN'
PLAN_SHARING_TYPE	VARCHAR(128)	The	Plan sharing type: 'GLOBAL', 'SESSION EXCLUSIVE GLOBAL', 'SESSION LOCAL'
OWNER_CONNECTION_ID	BIGINT	A	Connection owning the plan
PLAN_ID	BIGINT		Logical plan ID which is a non-negative value
PLAN_MEMORY_SIZE	BIGINT	Byte	Memory size used by the plan
REFERENCE_COUNT	BIGINT	Counter	Number of statements using the plan. When the count reaches zero, it can be evicted by the victim selection policy of plan cache
PARAMETER_COUNT	BIGINT	Counter	Number of parameters to be assigned for the execution
UPDATED_TABLE_OID	BIGINT		Object ID of the updated table for the plan
EXECUTION_COUNT	BIGINT	Counter	Accumulated count of plan execution
EXECUTION_COUNT_BY_ROUTING			
	BIGINT	Counter	Accumulated count of plan execution by client-routed connection in statement routing. This column shows how many times the statement is executed in routed connection. The routed connection means a physical connection in a part of the logical session
TOTAL_CURSOR_DURATION	BIGINT	Microsecond	Sum of time of plan execution including communication time with clients
AVG_CURSOR_DURATION	BIGINT	Microsecond	Average time of plan execution including communicate time with clients

MIN_CURSOR_DURATION	BIGINT	Microsecond	Minimum time of plan execution including communicate time with clients
MAX_CURSOR_DURATION	BIGINT	Microsecond	Maximum time of plan execution including communicate time with clients
TOTAL_EXECUTION_TIME	BIGINT	Microsecond	Sum of time of plan executions
AVG_EXECUTION_TIME	BIGINT	Microsecond	Average time of plan execution
MIN_EXECUTION_TIME	BIGINT	Microsecond	Minimum time of plan execution
MAX_EXECUTION_TIME	BIGINT	Microsecond	Maximum time of plan execution
TOTAL_EXECUTION_OPEN_TIME	BIGINT	Microsecond	Sum of time of establishing result sets
AVG_EXECUTION_OPEN_TIME	BIGINT	Microsecond	Average time for cursor open
MIN_EXECUTION_OPEN_TIME	BIGINT	Microsecond	Minimum time for cursor close
MAX_EXECUTION_OPEN_TIME	BIGINT	Microsecond	Maximum time for cursor close
TOTAL_EXECUTION_FETCH_TIME			
	BIGINT	Microsecond	Sum of time for transferring rows
AVG_EXECUTION_FETCH_TIME	BIGINT	Microsecond	Average time for cursor fetch
MIN_EXECUTION_FETCH_TIME	BIGINT	Microsecond	Minimum time for cursor fetch
MAX_EXECUTION_FETCH_TIME	BIGINT	Microsecond	Maximum time for cursor fetch
TOTAL_EXECUTION_CLOSE_TIME			
	BIGINT	Microsecond	Sum of time for cleanup of result sets
AVG_EXECUTION_CLOSE_TIME	BIGINT	Microsecond	Average time for cursor close
MIN_EXECUTION_CLOSE_TIME	BIGINT	Microsecond	Minimum time for cursor close
MAX_EXECUTION_CLOSE_TIME	BIGINT	Microsecond	Maximum time for cursor close
TOTAL_METADATA_CACHE_MISS_COUNT			
	BIGINT	Counter	Accumulated count of metadata cache misses during plan preparation and plan execution. When a metadata object is first requested to be accessed in a slave server instance, it gets the object from a master server instance via network communication, if the object does not exist or is out of date.
PREPARATION_COUNT	BIGINT	Counter	Number of plan preparations
TOTAL_PREPARATION_TIME	BIGINT	Microsecond	Sum of time of plan preparation
AVG_PREPARATION_TIME	BIGINT	Microsecond	Average time of plan preparation
MIN_PREPARATION_TIME	BIGINT	Microsecond	Minimum time of plan preparation
MAX_PREPARATION_TIME	BIGINT	Microsecond	Maximum time of plan preparation
TOTAL_RESULT_RECORD_COUNT	BIGINT	Counter	Accumulated number of records during plan execution.
TOTAL_LOCK_WAIT_COUNT	BIGINT	Counter	Accumulated lock wait count for the plan
TOTAL_LOCK_WAIT_DURATION	BIGINT	Microsecond	Accumulated lock wait duration for the plan
LAST_EXECUTION_TIMESTAMP	TIMESTAMP		Last execution timestamp
LAST_PREPARATION_TIMESTAMP			

	TIMESTAMP		Last preparation timestamp

M_SQL_PLAN_CACHE_OVERVIEW

Short description

Overall statistics of evicted and cached plans

Full description

M_SQL_PLAN_CACHE_OVERVIEW shows the overall statistics of evicted and cached plans. It shows information such as how many times plan eviction occurred and how long it takes to execute currently cached plans. In addition to statistics, the current status and flags of the SQL plan cache are shown in this view.

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
STATISTICS_COLLECTION_ENABLED			
	VARCHAR(6)		TRUE if runtime statistics are collected ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini','system') SET('sql', 'plan_cache_statistics_enabled') = <'True' or 'False'> WITH RECONFIGURE
PLAN_CACHE_CAPACITY	BIGINT	Byte	Maximum SQL Plan Cache size ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini','system') SET('sql', 'plan_cache_size') = '268435456' WITH RECONFIGURE
EVICTED_PLAN_COUNT	BIGINT		Number of evicted plans from SQL Plan Cache
EVICTED_PLAN_PREPARATION_COUNT			
	BIGINT		Total plan preparation count for evicted plans
EVICTED_PLAN_EXECUTION_COUNT			
	BIGINT		Total plan execution count for evicted plans
EVICTED_PLAN_PREPARATION_TIME			
	BIGINT	Microsecond	Total duration for plan preparation for all evicted plans
EVICTED_PLAN_CURSOR_DURATION			
	BIGINT	Microsecond	Total cursor duration for evicted plans
EVICTED_PLAN_TOTAL_EXECUTION_TIME			
	BIGINT	Microsecond	Total execution time for evicted plans
EVICTED_PLAN_SIZE	BIGINT	Byte	Accumulated total size of evicted plans
CACHED_PLAN_COUNT	BIGINT		Total cached plan count in SQL Plan Cache
CACHED_PLAN_PREPARATION_COUNT			
	BIGINT		Total plan preparation count for cached plans
CACHED_PLAN_EXECUTION_COUNT			

	BIGINT		Total execution count for cached plans
CACHED_PLAN_PREPARATION_TIME			
	BIGINT	Microsecond	Total plan preparation duration for cached plans
CACHED_PLAN_CURSOR_DURATION			
	BIGINT	Microsecond	Total cursor duration for cached plans
CACHED_PLAN_TOTAL_EXECUTION_TIME			
	BIGINT	Microsecond	Total execution time for cached plans
CACHED_PLAN_SIZE	BIGINT	Byte	Total size of SQL Plan Cache
PLAN_CACHE_ENABLED	TINYINT		TRUE if SQL Plan Cache is turned on ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini','system') SET('sql', 'plan_cache_enabled') = <'True' or 'False'> WITH RECONFIGURE
CLEAR_TIMESTAMP	TIMESTAMP		Timestamp when SQL Plan Cache is cleared

M_SQL_PLAN_CACHE_RESET

Short description

Statistics of an individual execution plan (since last reset)

This view contains values accumulated since the last reset of the main view **M_SQL_PLAN_CACHE**. Please refer to **M_SQL_PLAN_CACHE** for information about the structure and use of this view.

In addition to the members mentioned in **M_SQL_PLAN_CACHE**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_SYSTEM_LIMITS

Short description

System limits information

Structure

Column name	Data type	Unit	Description
CATEGORY	VARCHAR(128)		Category of system limit is adaptable
NAME	NVARCHAR(256)		Name of system limit
VALUE	VARCHAR(256)		Value of system limit
TYPE	VARCHAR(128)		Type of value
UNIT	VARCHAR(256)		Unit of value
COMMENT	VARCHAR(2000)		Additional comment for system limits

M_SYSTEM_OVERVIEW

Short description

Overview of system status including important resource usage information and alerts

Structure

Column name	Data type	Unit	Description
SECTION	VARCHAR(16)		Section name
NAME	VARCHAR(32)		Key name in section
STATUS	VARCHAR(8)		Status value: OK, ERROR, WARNING or empty for info items
VALUE	VARCHAR(256)		Key value in section

M_TABLES

Short description

Information on row and column tables

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
TABLE_NAME	NVARCHAR(256)		Table name
RECORD_COUNT	BIGINT		Number of records in this table
TABLE_SIZE	BIGINT		Allocated memory size for fixed-size and variable-size part
IS_COLUMN_TABLE	VARCHAR(5000)		Specifies whether the table is a column table: 'TRUE', 'FALSE'
TABLE_TYPE	VARCHAR(5000)		Specifies type of the table: 'ROW', 'COLUMN', 'HYBRID'

M_TABLE_LOB_FILES

Short description

All LOB files that belong to a table

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
SCHEMA_NAME	NVARCHAR(256)		Schema name
TABLE_NAME	NVARCHAR(256)		Table name
TABLE_OID	BIGINT		Table OID, same as owner_oid if the table is found; 0 otherwise
CONTAINER_ID	BIGINT		Container ID
PHYSICAL_SIZE	BIGINT	Byte	File size
BINARY_SIZE	BIGINT	Byte	BLOB size
CHARACTER_SIZE	INTEGER		Number of characters in NCLOB
PAGE_COUNT	BIGINT		Number of pages used for a BLOB file
REF_COUNT	BIGINT		Current Refcount of a BLOB file

M_TABLE_LOCATIONS

Short description

Tables and their logical location. Physical locations are shown in M_TABLE_PERSISTENCE_LOCATIONS

Full description

To move tables to other locations use:

```
ALTER TABLE name MOVE TO LOCATION 'host:port'
```

To move parts of a split table to other locations use:

```
ALTER TABLE name MOVE PART i TO LOCATION 'host:port'
```

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
TABLE_NAME	NVARCHAR(256)		Table name
PART_ID	INTEGER		Part ID of split table
HOST	VARCHAR(64)		Host where table data is located. Empty for Views
PORT	INTEGER		Port where table data is located. 0 for Views
LOCATION	VARCHAR(5000)		Host and port where the table data is located

M_TABLE_LOCKS

Short description

Status of currently acquired locks on tables with detailed information such as lock acquisition time, lock mode

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
LOCK_OWNER_TRANSACTION_ID	INTEGER		Transaction ID of the lock owner
ACQUIRED_TIME	TIMESTAMP		Lock acquisition time
SCHEMA_NAME	NVARCHAR(256)		Schema name
TABLE_NAME	NVARCHAR(256)		Table name
OBJECT_TYPE	VARCHAR(32)		Object type
LOCK_MODE	VARCHAR(32)		Lock mode: I (Exclusive Lock), IX (Intentional Exclusive Lock)

M_TABLE_PERSISTENCE_LOCATIONS

Short description

Column store tables and their physical data locations

Full description

This view shows information on which node contains the persistence parts of a table. This will include the assigned node as visible from the view M_TABLE_LOCATIONS but will also include other nodes that still contains some persistence of the table. This could occur if a table is moved and not yet merged. In this case the old node will still contain some persistence content for the table beside the currently assigned node.

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
TABLE_NAME	NVARCHAR(256)		Table name
PART_ID	INTEGER		Part ID of split table
IS_HISTORY	VARCHAR(6)		Is this the history part of a table
PERSISTENCE_HOST	VARCHAR(64)		Host where table data is located
PERSISTENCE_PORT	INTEGER		Port where table data is located

M_TABLE_PERSISTENCE_STATISTICS

Short description

Persistence virtual file summary statistics for tables

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema Name
TABLE_NAME	NVARCHAR(256)		Table Name
DISK_SIZE	BIGINT		Total disk size of all table parts
PAGE_COUNT	BIGINT		Total number of pages of all table parts
BYTES_WRITTEN	BIGINT		Number of bytes written to the table
BYTES_APPENDED	BIGINT		Number of bytes appended to the table
BYTES_READ	BIGINT		Number of bytes read from the table
BYTESTREAM_WRITTEN	BIGINT		Number of bytes written to the table via streaming interface
APPEND_COUNT	BIGINT		Number of times the table was appended to
WRITE_COUNT	BIGINT		Number of times the table was written to
OPTIMIZE_COUNT	BIGINT		Number of times the table was access optimized
READ_COUNT	BIGINT		Number of times the table was read from
TRUNCATE_COUNT	BIGINT		Number of times the table was truncated
COPY_COUNT	BIGINT		Number of times the table was copied

M_TABLE_VIRTUAL_FILES

Short description

All virtual files that belong to a table

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
SCHEMA_NAME	NVARCHAR(256)		Schema name
TABLE_NAME	NVARCHAR(256)		Table name
PART_ID	INTEGER		Part ID of split table
IS_HISTORY	VARCHAR(6)		Is this the history part of a table
CONTAINER_ID	BIGINT		Container ID
NAMESPACE	VARCHAR(512)		File path
NAME	VARCHAR(512)		File name
PHYSICAL_SIZE	BIGINT	Byte	File size
PAGE_COUNT	BIGINT		Unused

M_TEMPORARY_TABLES

Short description

Temporary tables

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
TABLE_NAME	NVARCHAR(256)		Table name
TABLE_OID	BIGINT		Object ID of the table
COMMENTS	NVARCHAR(5000)		Comment on the table
FIXED_PART_SIZE	SMALLINT	Byte	Fixed-size part of record
IS_LOGGED	NVARCHAR(10)		Specifies whether logging is enabled for the table: 'TRUE', 'FALSE'
IS_SYSTEM_TABLE	NVARCHAR(10)		Specifies whether the table is a system table: 'TRUE', 'FALSE'
IS_COLUMN_TABLE	NVARCHAR(10)		Specifies whether the table is a column table: 'TRUE', 'FALSE'
IS_INSERT_ONLY	NVARCHAR(10)		Specifies whether the table is an 'insert only' table: 'TRUE', 'FALSE'
IS_TENANT_SHARED_DATA	NVARCHAR(10)		Specifies whether the table can be shared among other instances: 'TRUE', 'FALSE'
IS_TENANT_SHARED_METADATA	NVARCHAR(10)		Specifies whether the table is a global table: 'TRUE', 'FALSE'
SESSION_TYPE	NVARCHAR(20)		Session type: 'NONE', 'SIMPLE', 'HISTORY'. For 'HISTORY', time travel is possible
IS_TEMPORARY	NVARCHAR(10)		Specifies whether the table is a temporary table: 'TRUE', 'FALSE'
TEMPORARY_TABLE_TYPE	NVARCHAR(10)		Temporary table type
IS_USER_DEFINED_TYPE	NVARCHAR(10)		'TRUE' if user defined table type, 'FALSE' otherwise
PRELOAD	NVARCHAR(10)		'TRUE' if the table uses preloading, else 'FALSE'. Only valid for column store tables

M_TEMPORARY_TABLE_COLUMNS

Short description

Columns of temporary tables

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
TABLE_NAME	NVARCHAR(256)		Table name
TABLE_OID	BIGINT		Object ID of the table
COLUMN_NAME	NVARCHAR(256)		Name of the column
POSITION	INTEGER		Ordinal position of the table column
DATA_TYPE_ID	SMALLINT		SQL data type ID of the column
DATA_TYPE_NAME	VARCHAR(16)		SQL data type name of the column
OFFSET	SMALLINT	Byte	Offset of the column in the record
LENGTH	DECIMAL(34,-1)		# of chars for char types; # of max digits for numeric types; # of chars for datetime types; # of bytes for LOB types
SCALE	INTEGER		Numeric types: the maximum number of digits to the right of the decimal point; time, timestamp: the decimal digits are defined as the number of digits to the right of the decimal point in the second's component of the data
IS_NULLABLE	VARCHAR(5000)		Specifies whether the column is allowed to accept a null value: 'TRUE', 'FALSE'
DEFAULT_VALUE	NVARCHAR(5000)		Default value of the column
COLLATION	NVARCHAR(5000)		Collation of the column
COMMENTS	NVARCHAR(5000)		Comments on the column
MAX_VALUE	NVARCHAR(5000)		Maximum value
MIN_VALUE	NVARCHAR(5000)		Minimum value
CS_DATA_TYPE_ID	INTEGER		Column store data type ID
CS_DATA_TYPE_NAME	VARCHAR(5000)		Column store data type name
DDIC_DATA_TYPE_ID	INTEGER		DDIC data type ID
DDIC_DATA_TYPE_NAME	VARCHAR(5000)		DDIC data type name
COMPRESSION_TYPE	VARCHAR(5000)		Type of compression: 'DEFAULT', 'PREFIXED', 'SPARSE', 'CLUSTERED', 'INDIRECT', 'RLE'
INDEX_TYPE	VARCHAR(5000)		Type of index: 'NONE', 'FULL', 'SIGNATURE'
COLUMN_ID	BIGINT		ID of the column
PRELOAD	VARCHAR(5000)		Specifies if column is preloaded: 'TRUE', 'FALSE'

M_TEMPORARY_VIEWS

Short description

Temporary views

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
VIEW_NAME	NVARCHAR(256)		View name
VIEW_OID	BIGINT		Object ID of the view
IS_UNICODE	NVARCHAR(10)		Specifies whether the query string contains Unicode: 'TRUE', 'FALSE'
IS_READ_ONLY	NVARCHAR(10)		Specifies whether this view is a read-only or an updatable view: 'TRUE', 'FALSE'
HAS_CHECK_OPTION	NVARCHAR(10)		Specifies whether this view has an updatable view condition: 'TRUE', 'FALSE'
HAS_COLUMN_ALIASE	NVARCHAR(10)		Specifies whether the view has a columns alias : 'TRUE', 'FALSE'
DEFINITION	NCLOB		Definition of the view
COMMENTS	NVARCHAR(5000)		Description on the view
IS_COLUMN_VIEW	NVARCHAR(10)		Specifies whether this view is a column view or not: 'TRUE', 'FALSE'
VIEW_TYPE	NVARCHAR(20)		Type of view : 'ROW', 'OLAP', 'JOIN', 'HIERARCHY', 'CALC'
IS_TENANT_SHARED	NVARCHAR(10)		Specifies whether the view metadata can be shared across tenants: 'TRUE', 'FALSE'

M_TEMPORARY_VIEW_COLUMNS

Short description

Columns of temporary tables

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
VIEW_NAME	NVARCHAR(256)		View name
VIEW_OID	BIGINT		Object ID of the view
COLUMN_NAME	NVARCHAR(256)		View column name
POSITION	INTEGER		Ordinal position of the view column
DATA_TYPE_ID	SMALLINT		SQL data type ID of the column
DATA_TYPE_NAME	VARCHAR(16)		SQL data type name of the column
OFFSET	SMALLINT		Offset of the column in a record
LENGTH	DECIMAL(34,-1)		# of chars for char types; # of max digits for numeric types; # of chars for datetime types; # of bytes for LOB types
SCALE	INTEGER		Numeric types: the maximum number of digits to the right of the decimal point; time, timestamp: the decimal digits are defined as the number of digits to the right of the decimal point in the second's component of the data
IS_NULLABLE	VARCHAR(5000)		Specifies whether the column is allowed to accept null value: 'TRUE', 'FALSE'
DEFAULT_VALUE	NVARCHAR(5000)		Default value
COLLATION	NVARCHAR(5000)		Collation
COMMENTS	NVARCHAR(5000)		Description for this column
MAX_VALUE	NVARCHAR(5000)		Maximum value
MIN_VALUE	NVARCHAR(5000)		Minimum value
CS_DATA_TYPE_ID	INTEGER		Column store data type ID
CS_DATA_TYPE_NAME	VARCHAR(5000)		Column store data type name
DDIC_DATA_TYPE_ID	INTEGER		DDIC data type ID
DDIC_DATA_TYPE_NAME	VARCHAR(5000)		DDIC data type name
COMPRESSION_TYPE	VARCHAR(5000)		Type of compression: 'DEFAULT', 'PREFIXED', 'SPARSE', 'CLUSTERED', 'INDIRECT', 'RLE'
INDEX_TYPE	VARCHAR(5000)		Type of index: 'NONE', 'FULL', 'SIGNATURE'
COLUMN_ID	BIGINT		ID of the column
PRELOAD	VARCHAR(5000)		Specifies if column is preloaded: 'TRUE', 'FALSE'

M_TENANTS

Short description

Available tenant information

Structure

Column name	Data type	Unit	Description
TENANT_NAME	VARCHAR(256)		Tenant name
CREATE_TIME	TIMESTAMP		Creation time
TENANT_STATUS	VARCHAR(128)		Suspend status: SUSPENDED, ACTIVE, PENDING
HOST	VARCHAR(64)		Tenant service host name
PORT	INTEGER		Tenant service internal port
SQL_PORT	INTEGER		Tenant service SQL port

M_TEXT_ANALYSIS_LANGUAGES

Short description

List of supported languages

Structure

Column name	Data type	Unit	Description
LANGUAGE_NAME	VARCHAR(256)		Language name
LANGUAGE_CODE	VARCHAR(2)		Language ISO 639-1 code

M_TEXT_ANALYSIS_MIME_TYPES

Short description

List of supported mime types

Structure

Column name	Data type	Unit	Description
MIME_TYPE_NAME	VARCHAR(256)		Name of the mime type
MIME_TYPE_DESCRIPTION	NVARCHAR(256)		Description of the mime type

M_TOPOLOGY_TREE

Short description

SAP HANA nameserver topology content

Full description

This monitoring view only returns data if you specify PATH=... . Use '/' as root path. For deeper paths use PATH+'/'+NAME. If NAME contains '/' you have to enclose it in CHR(1), For example

```
SELECT * FROM M_TOPOLOGY_TREE WHERE PATH='/index/'||CHR(1)||'SYSTEM:A/B'||CHR(1)||'#
15046'
```

To change topology tree values use ALTER SYSTEM INIFILE ('topology.ini','system') set (PATH,NAME)=VALUE
To delete values use ALTER SYSTEM INIFILE('topology.ini','system') remove (PATH,NAME)

Structure

Column name	Data type	Unit	Description
PATH	VARCHAR(512)		Path to Key
NAME	VARCHAR(256)		Key name
VALUE	VARCHAR(256)		Key value
LEAF	VARCHAR(6)	Boolean	Leaf flag

M_TRACEFILES

Short description

All trace files

Full description

With the "CLEAR" command, all files that were opened by a service will be removed or reset to size 0. On a distributed system, this command will clear all traces on all hosts.

```
ALTER SYSTEM CLEAR TRACES ('ALERT','CLIENT','CRASHDUMP,'*','INDEXSERVER',...,'DAEMO
N' );
```

It can clear different types of files:

Name	Files
ALERT	<service>alert....trc
CLIENT	localclient_....trc
CRASHDUMP	*.crashdump....trc
*	open *.trc files of all active services
INDEXSERVER,NAMESERVER,...,DAEMON	open *.trc files of a single service type

This command will delete files on host 'HOST_NAME'. When a service has a trace file open, then it cannot be deleted, so a "CLEAR" command should be used in that case.

```
ALTER SYSTEM DELETE TRACES ('HOST_NAME', 'FILENAME1', 'FILENAME2' ... );
```

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
FILE_NAME	VARCHAR(256)		File name
FILE_SIZE	BIGINT	Byte	File size
FILE_MTIME	TIMESTAMP		File date

M_TRACEFILE_CONTENTS

Short description

SAP HANA information from trace files

Full description

This view only returns data if you specify 'HOST=... and FILE_NAME=...' as part of the where clause. Use HOST, FILE_NAME as returned by the view M_TRACEFILES. This prevents unintentional materialization of all trace files.

The optional where clause 'OFFSET<,<=,=,>,>= value' is efficently handled by this view. With 'OFFSET > -value' you can read from the end of the file, without having to know the file size in advance. With OFFSET <> -value you can read from the start and end of a file. Do not use the equivalent OFFSET < value OR OFFSET >-value, because this is very inefficient for large files and returns duplicates for small files where the file size is < 2*value.

Trace files typically contain ASCII or CESU-8, but can also contain binary data. To support all kinds of data, each byte from the trace file is encoded as one NVARCHAR with values in the range 0 to 255. To recode as CESU-8 you have to use code like in this Python example:

```
cursor.execute(" select CONTENT from M_TRACEFILE_CONTENTS where HOST='...' and FILE_
NAME='...' ")
filedata=''
for row in cursor.fetchall():
    filedata += row[1].encode('latin-1') # reinterpret as bytearray
displaydata = filedata.decode('utf-8','replace') # do not use 'strict' error handlin
g
```

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
FILE_NAME	VARCHAR(256)		File name
OFFSET	BIGINT	Byte	File offset
CONTENT	NVARCHAR(1000)		File content at offset

M_TRANSACTIONS

Short description

All transactions created by users or database

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
CONNECTION_ID	INTEGER		Connection ID
TRANSACTION_ID	INTEGER		Transaction ID
VOLUME_ID	INTEGER		Volume ID where the transaction has started
TRANSACTION_SEQUENCE	INTEGER		Transaction sequence number in a transaction
UPDATE_TRANSACTION_ID	BIGINT		Update transaction ID
TRANSACTION_STATUS	VARCHAR(128)		Transaction status; INACTIVE, ACTIVE, PRECOMMITTED, ABORTING, PARTIAL_ABORTING, ACTIVE_PREPARE_COMMIT
TRANSACTION_TYPE	VARCHAR(128)		Transaction type; USER, VERSION GARBAGE COLLECTION, DDL VERSION GARBAGE COLLECTION, INTERNAL, EXTERNAL
START_TIME	TIMESTAMP		Start time
END_TIME	TIMESTAMP		End time
START_MVCC_TIMESTAMP	BIGINT		Internal MVCC timestamp of the transaction start time
END_MVCC_TIMESTAMP	BIGINT		Internal MVCC timestamp of the transaction end time
EXECUTED_STATEMENT_COUNT	INTEGER		Number of executed statements in the transaction
CREATED_VERSION_COUNT	INTEGER		Number of created versions in the transaction
ALLOCATED_VERSION_SIZE	BIGINT	Byte	Allocated version size in the transaction
ACQUIRED_LOCK_COUNT	INTEGER		Number of acquired locks in the transaction
LOCK_WAIT_COUNT	INTEGER		Number of lock waits in the transaction
LOCK_WAIT_TIME	DOUBLE	Second	Accumulated lock wait time in the transaction
LOG_SIZE	BIGINT	Byte	Generated log amount in the transaction
CURRENT_STATEMENT_ID	VARCHAR(256)		Current statement ID
CURRENT_STATEMENT_SEQUENCE			
	INTEGER		Sequence number of the currently executed statement
ALLOCATED_MEMORY_SIZE	BIGINT	Byte	Memory size occupied by the transaction
ACQUIRED_METALOCK_INDEX	INTEGER		Index of acquired metalock
LOG_PARTITION_ID	SMALLINT		Log partition ID of the transaction

REDO_LOG_AMOUNT	BIGINT	Byte	Size of redo log amount generated by the transaction
UNDO_LOG_AMOUNT	BIGINT	Byte	Size of undo log amount generated by the transaction
MIN_MVCC_SNAPSHOT_TIMESTAMP			
	BIGINT		Logical timestamp of the first executed statement in the transaction
LAST_COMMIT_ID	BIGINT		Last commit ID of the transaction
ACTIVE_STATEMENT_COUNT	INTEGER		Number of opened cursors in the transaction
ISOLATION_LEVEL	VARCHAR(128)		Isolation level of each transaction: READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SERIALIZABLE
LOG_FLUSH_ENABLED	VARCHAR(6)		TRUE if logging is enabled for the transaction, FALSE otherwise
LOGGING_ENABLED	VARCHAR(6)		TRUE if logging is enabled for the transaction, FALSE otherwise

M_UNDO_CLEANUP_FILES

Short description

Information about undo files and cleanup files

Full description

Each undo or cleanup file in the system is represented by one row in this view. Undo files contain information needed for transaction rollback. These files are removed on transaction end. If data is deleted but must still be accessible because of MVCC isolation, the corresponding information is written to cleanup files. On transaction end cleanup files are passed to history management. Garbage collection uses the cleanup files to finally remove data. Undo files and cleanup files may be cached and reused because of performance issues. Cached files get assigned the type "FREE". Undo files for row store get the type "EXTERNALUNDO".

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
TYPE	VARCHAR(16)		File type (UNDO, CLEANUP, EXTERNALUNDO, FREE)
TID	BIGINT	TID	Transaction ID
PAGE_COUNT	BIGINT	Counter	Page count
RAW_SIZE	BIGINT	Byte	Raw size
CLEANUP_MARK	BIGINT		Cleanup position mark
NESTED_SESSION_ID	INTEGER		Nested session ID
NESTED_SESSION_PARENT_ID	INTEGER		Nested session parent ID
DEPENDENT_INDEX	INTEGER		Dependent index during redo
INDOUBT_FLAG	VARCHAR(6)		Indoubt flag for distributed transaction
TENTATIVE_PRECOMMIT_POSITION			
	BIGINT		Tentative precommit position

M_VERSION_MEMORY

Short description

Memory usage of row-store Multiversion Concurrency Control (MVCC) manager

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
ALLOCATED_MEMORY_SIZE	BIGINT	Byte	Size of memory allocated for row-store version space
USED_MEMORY_SIZE	BIGINT	Byte	Size of memory actually used by row-store versions
RECLAIMED_VERSION_SIZE	BIGINT	Byte	Size of memory reclaimed by row-store version garbage collection
FREE_MEMORY_SIZE	BIGINT	Byte	Size of memory freed and reusable in row-store version space

M_VOLUMES

Short description

Volumes used by SAP HANA servers

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
SERVICE_NAME	VARCHAR(256)		Service name
VOLUME_ID	INTEGER		Volume ID
SUBPATH	VARCHAR(512)		Subpath appended to M_DISK.PATH

M_VOLUME_FILES

Short description

Information about volume files

Full description

Information about files in the volume directories. All files in these directories will be shown, but only registered files (that is, files currently used by the database) have a file type. `TOTAL_SIZE` is the size as reported by the file system.

The meaning of `USED_SIZE` depends on the file type:

- **DATA:** Size of used and shadow pages in this data volume file
- **LOG:** Used size equals `TOTAL_SIZE`
- **TRACE:** Used size is zero for unused trace files and equals `TOTAL_SIZE` for used trace files

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Volume ID
FILE_TYPE	VARCHAR(16)		Type of file (DATA, LOG, TRACER, ...)
FILE_NAME	VARCHAR(256)		File name
USED_SIZE	BIGINT		Size of used data within file
TOTAL_SIZE	BIGINT		Total file size

M_VOLUME_IO_PERFORMANCE_STATISTICS

Short description

File performance statistics

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_VOLUME_IO_PERFORMANCE_STATISTICS_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_VOLUME_IO_PERFORMANCE_STATISTICS_RESET
```

or the management console command:

```
monitor reset -n M_VOLUME_IO_PERFORMANCE_STATISTICS_RESET
```

Full description

This view shows I/O performance for various buffer sizes. Each buffer size is a max. value. That is, buffer size 4k actually means <= 4k and buffer size 16k means 4k < buffer size <= 16k. The I/O time is the total time from starting a request until it finishes, including enqueue time. All measured times are timespans between enqueueing and finishing a request. This means they contain wait times if the request cannot be started immediately. Enqueue time is measured separately to see how many requests are executed asynchronously and how many are synchronous or blocking. For overall I/O performance, see `TOTAL_READ/WRITE_SIZE` and `TOTAL_IO_TIME` in the monitoring view **M_VOLUME_IO_STATISTICS**.

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
PATH	VARCHAR(512)		Filesystem path
TYPE	VARCHAR(16)		Type of contained files
MAX_IO_BUFFER	BIGINT	Byte	Max. I/O buffer size
READ_SYNC	BIGINT	Counter	Count of synchronous reads
WRITE_SYNC	BIGINT	Counter	Count of synchronous writes
READ_REQUESTS	BIGINT	Counter	Count of read requests
WRITE_REQUESTS	BIGINT	Counter	Count of write requests
READ_COMPLETIONS	BIGINT	Counter	Count of read completions
WRITE_COMPLETIONS	BIGINT	Counter	Count of write completions
FULL_RETRY_READS	BIGINT	Counter	Count of full retry reads
FULL_RETRY_WRITES	BIGINT	Counter	Count of full retry writes
SHORT_READS	BIGINT	Counter	Count of short reads

SHORT_WRITES	BIGINT	Counter	Count of short writes
DELAYED_READ_REQUESTS	BIGINT	Counter	Count of delayed read requests
DELAYED_WRITE_REQUESTS	BIGINT	Counter	Count of delayed write requests
DEQUEUED_DELAYED_REQUESTS	BIGINT	Counter	Count of dequeued delayed requests
RESUBMITTED_DELAYED_REQUESTS			
	BIGINT	Counter	Count of resubmitted delayed requests
RESUBMITTED_DELAYED_REQUESTS_DELAYED_AGAIN			
	BIGINT	Counter	Count of resubmitted delayed requests delayed
ITEMS_IN_DELAY_QUEUE	BIGINT	Counter	Count of items in delay queue
MAX_ITEMS_IN_DELAY_QUEUE	BIGINT	Counter	Maximum size of delay queue
LAST_READ_SYNC_TIME	BIGINT	Microsecond	Time for synchronous reads (last)
MAX_READ_SYNC_TIME	BIGINT	Microsecond	Time for synchronous reads (max)
MIN_READ_SYNC_TIME	BIGINT	Microsecond	Time for synchronous reads (min)
SUM_READ_SYNC_TIME	BIGINT	Microsecond	Time for synchronous reads (total)
AVG_READ_SYNC_TIME	BIGINT	Microsecond	Time for synchronous reads (avg)
LAST_WRITE_SYNC_TIME	BIGINT	Microsecond	Time for synchronous writes (last)
MAX_WRITE_SYNC_TIME	BIGINT	Microsecond	Time for synchronous writes (max)
MIN_WRITE_SYNC_TIME	BIGINT	Microsecond	Time for synchronous writes (min)
SUM_WRITE_SYNC_TIME	BIGINT	Microsecond	Time for synchronous writes (total)
AVG_WRITE_SYNC_TIME	BIGINT	Microsecond	Time for synchronous writes (avg)
LAST_READ_TIME	BIGINT	Microsecond	Time for read events (last)
MAX_READ_TIME	BIGINT	Microsecond	Time for read events (max)
MIN_READ_TIME	BIGINT	Microsecond	Time for read events (min)
SUM_READ_TIME	BIGINT	Microsecond	Time for read events (total)
AVG_READ_TIME	BIGINT	Microsecond	Time for read events (avg)
LAST_WRITE_TIME	BIGINT	Microsecond	Time for write events (last)
MAX_WRITE_TIME	BIGINT	Microsecond	Time for write events (max)
MIN_WRITE_TIME	BIGINT	Microsecond	Time for write events (min)
SUM_WRITE_TIME	BIGINT	Microsecond	Time for write events (total)
AVG_WRITE_TIME	BIGINT	Microsecond	Time for write events (avg)
LAST_READ_ENQUEUE_TIME	BIGINT	Microsecond	Time for enqueuing read I/O events (last)
MAX_READ_ENQUEUE_TIME	BIGINT	Microsecond	Time for enqueuing read I/O events (max)
MIN_READ_ENQUEUE_TIME	BIGINT	Microsecond	Time for enqueuing read I/O events (min)
SUM_READ_ENQUEUE_TIME	BIGINT	Microsecond	Time for enqueuing read I/O events (total)
AVG_READ_ENQUEUE_TIME	BIGINT	Microsecond	Time for enqueuing read I/O events (avg)
LAST_WRITE_ENQUEUE_TIME	BIGINT	Microsecond	Time for enqueuing write I/O events (last)
MAX_WRITE_ENQUEUE_TIME	BIGINT	Microsecond	Time for enqueuing write I/O events (max)
MIN_WRITE_ENQUEUE_TIME	BIGINT	Microsecond	Time for enqueuing write I/O events (min)
SUM_WRITE_ENQUEUE_TIME	BIGINT	Microsecond	Time for enqueuing write I/O events (total)

AVG_WRITE_ENQUEUE_TIME	BIGINT	Microsecond	Time for enqueuing write I/O events (avg)
LAST_READ_SIZE	BIGINT	Byte	Size of read data (last)
MAX_READ_SIZE	BIGINT	Byte	Size of read data (max)
MIN_READ_SIZE	BIGINT	Byte	Size of read data (min)
SUM_READ_SIZE	BIGINT	Byte	Size of read data (total)
AVG_READ_SIZE	BIGINT	Byte	Size of read data (avg)
LAST_WRITE_SIZE	BIGINT	Byte	Size of written data (last)
MAX_WRITE_SIZE	BIGINT	Byte	Size of written data (max)
MIN_WRITE_SIZE	BIGINT	Byte	Size of written data (min)
SUM_WRITE_SIZE	BIGINT	Byte	Size of written data (total)
AVG_WRITE_SIZE	BIGINT	Byte	Size of written data (avg)
LAST_READ_SYNC_SIZE	BIGINT	Byte	Size of synchronously read data (last)
MAX_READ_SYNC_SIZE	BIGINT	Byte	Size of synchronously read data (max)
MIN_READ_SYNC_SIZE	BIGINT	Byte	Size of synchronously read data (min)
SUM_READ_SYNC_SIZE	BIGINT	Byte	Size of synchronously read data (total)
AVG_READ_SYNC_SIZE	BIGINT	Byte	Size of synchronously read data (avg)
LAST_WRITE_SYNC_SIZE	BIGINT	Byte	Size of synchronously written data (last)
MAX_WRITE_SYNC_SIZE	BIGINT	Byte	Size of synchronously written data (max)
MIN_WRITE_SYNC_SIZE	BIGINT	Byte	Size of synchronously written data (min)
SUM_WRITE_SYNC_SIZE	BIGINT	Byte	Size of synchronously written data (total)
AVG_WRITE_SYNC_SIZE	BIGINT	Byte	Size of synchronously written data (avg)

M_VOLUME_IO_PERFORMANCE_STATISTICS_RESET

Short description

File performance statistics (since last reset)

This view contains values accumulated since the last reset of the main view **M_VOLUME_IO_PERFORMANCE_STATISTICS**. Please refer to **M_VOLUME_IO_PERFORMANCE_STATISTICS** for information about the structure and use of this view.

In addition to the members mentioned in **M_VOLUME_IO_PERFORMANCE_STATISTICS**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_VOLUME_IO_STATISTICS

Short description

File access statistics

Note:

This view has a resettable counterpart. This means, you can also see the values since the last reset in the dependent view **M_VOLUME_IO_STATISTICS_RESET**. To reset the values, use the SQL command:

```
ALTER SYSTEM RESET MONITORING VIEW SYS.M_VOLUME_IO_STATISTICS_RESET
```

or the management console command:

```
monitor reset -n M_VOLUME_IO_STATISTICS_RESET
```

Full description

This view shows information about basic I/O operations on I/O subsystems (that is, paths). Please refer to **M_VOLUME_IO_PERFORMANCE_STATISTICS** for detailed information about read/write performance on various buffer sizes.

TOTAL_READ/WRITE_SIZE is aggregated from M_VOLUME_IO_PERFORMANCE_STATISTICS. Note that the column TOTAL_IO_TIME is only an approximation of real I/O times and should not be used to compute exact throughput ratios.

Resetting this view implicitly resets the dependent child view M_VOLUME_IO_PERFORMANCE_STATISTICS.

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
VOLUME_ID	INTEGER		Persistence Volume ID
PATH	VARCHAR(512)		Filesystem path
TYPE	VARCHAR(16)		Type of contained files
OPEN_CALL_COUNT	BIGINT	Counter	Count of open calls
CLOSE_CALL_COUNT	BIGINT	Counter	Count of close calls
EXISTS_CALL_COUNT	BIGINT	Counter	Count of exists calls
REMOVE_CALL_COUNT	BIGINT	Counter	Count of remove calls
RESIZE_CALL_COUNT	BIGINT	Counter	Count of resize calls
SYNC_CALL_COUNT	BIGINT	Counter	Count of sync calls
ENQUEUED_WRITE_REQUESTS	BIGINT	Counter	Count of enqueued write requests
EOF_READS	BIGINT	Counter	Count of EOF reads

TOTAL_FULL_RETRY_READS	BIGINT	Counter	Count of full retry reads
TOTAL_FULL_RETRY_WRITES	BIGINT	Counter	Count of full retry writes
TOTAL_SHORT_READS	BIGINT	Counter	Count of short reads
TOTAL_SHORT_WRITES	BIGINT	Counter	Count of short writes
TOTAL_READ_SIZE	BIGINT	Byte	Size of read data
TOTAL_WRITE_SIZE	BIGINT	Byte	Size of written data
TOTAL_IO_TIME	BIGINT	Microsecond	Total I/O time
TOTAL_READ_TIME	BIGINT	Microsecond	Total read time
TOTAL_WRITE_TIME	BIGINT	Microsecond	Total write time

M_VOLUME_IO_STATISTICS_RESET

Short description

File access statistics (since last reset)

This view contains values accumulated since the last reset of the main view **M_VOLUME_IO_STATISTICS**. Please refer to **M_VOLUME_IO_STATISTICS** for information about the structure and use of this view.

In addition to the members mentioned in **M_VOLUME_IO_STATISTICS**, this view also contains a timestamp field **RESET_TIME**, which indicates the last time the data was reset.

M_VOLUME_SIZES

Short description

Volume sizes used by SAP HANA servers

Structure

Column name	Data type	Unit	Description
VOLUME_ID	INTEGER		Volume ID. See M_VOLUMES
DISK_ID	INTEGER		Disk ID. See M_DISKS
DATA_SIZE	BIGINT	Byte	Data area size
LOG_SIZE	BIGINT	Byte	Log area size

M_WORKLOAD

Short description

Database workload collected every minute

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		Host name
PORT	INTEGER		Internal port
EXECUTION_COUNT	BIGINT		Total count of all executed statements for data manipulation, data definition, and system control
COMPILATION_COUNT	BIGINT		Number of statement preparation
UPDATE_TRANSACTION_COUNT	BIGINT		Number of update transactions
COMMIT_COUNT	BIGINT		Number of transaction commits
ROLLBACK_COUNT	BIGINT		Number of transaction rollbacks
CURRENT_EXECUTION_RATE	DOUBLE		Current statement execution count per minute
PEAK_EXECUTION_RATE	DOUBLE		Peak statement execution count per minute
CURRENT_COMPILATION_RATE	DOUBLE		Current statement preparation count per minute
PEAK_COMPILATION_RATE	DOUBLE		Peak statement preparation count per minute
CURRENT_UPDATE_TRANSACTION_RATE			
	DOUBLE		Current update transaction count per minute
PEAK_UPDATE_TRANSACTION_RATE			
	DOUBLE		Peak update transaction count per minute
CURRENT_TRANSACTION_RATE	DOUBLE		Current transaction count per minute
PEAK_TRANSACTION_RATE	DOUBLE		Peak transaction count per minute
CURRENT_COMMIT_RATE	DOUBLE		Total number of commits per minute
PEAK_COMMIT_RATE	DOUBLE		Peak commit counts per minute
CURRENT_ROLLBACK_RATE	DOUBLE		Total number of rollbacks per minute
PEAK_ROLLBACK_RATE	DOUBLE		Peak rollback count per minute
CURRENT_MEMORY_USAGE_RATE	DOUBLE		Total size of used memory per minute
PEAK_MEMORY_USAGE_RATE	DOUBLE		Peak size of used memory per minute

M_XS_APPLICATIONS

Short description

Applications deployed and running in the XSEngine

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		
PORT	INTEGER		
APP_NAME	VARCHAR(512)		Technical name of the application
APP_ALIAS	VARCHAR(512)		Alias name of the application, used to have more readable URIs
IS_STATEFUL	INTEGER		Whether the application keeps a session state between requests

M_XS_APP_ISSUES

Short description

Issues related with applications in the XSEngine

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR(64)		
PORT	INTEGER		
APP_NAME	VARCHAR(512)		Technical name of the application
ERROR_MESSAGE	VARCHAR(1024)		Textual description of the issue

OBJECTS

Short description

Available objects

Structure

Column name	Data type	Unit	Description
OBJECT_CATEGORY	VARCHAR(5000)		Object category of the object (default = null)
SCHEMA_NAME	NVARCHAR(256)		Schema name of the object
OBJECT_NAME	NVARCHAR(5000)		Object name
OBJECT_TYPE	VARCHAR(5000)		Object type
OBJECT_OID	BIGINT		Object ID

OBJECT_DEPENDENCIES

Short description

Dependencies between objects, for example, views which refer to a specific table

Structure

Column name	Data type	Unit	Description
BASE_SCHEMA_NAME	NVARCHAR(256)		Schema name of the base object
BASE_OBJECT_NAME	NVARCHAR(5000)		Object name of the base object
BASE_OBJECT_TYPE	VARCHAR(5000)		Type of the base object
DEPENDENT_SCHEMA_NAME	NVARCHAR(256)		Schema name of the dependent object
DEPENDENT_OBJECT_NAME	NVARCHAR(5000)		Object name of the dependent object
DEPENDENT_OBJECT_TYPE	VARCHAR(5000)		Type of the base dependent
DEPENDENCY_TYPE	INTEGER		Type of dependency between base and dependent object

OWNERSHIP

Short description

Ownership of available objects when users create any object in other users' schemas

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(5000)		Schema name of the object
OWNER_NAME	NVARCHAR(256)		Name of the object owner
OBJECT_NAME	NVARCHAR(256)		Object name
OBJECT_TYPE	NVARCHAR(5000)		Object type
OBJECT_OID	BIGINT		Object ID

PROCEDURES

Short description

Available stored procedures

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name of the stored procedure
PROCEDURE_NAME	NVARCHAR(256)		Name of the stored procedure
PROCEDURE_OID	BIGINT		Object ID of the stored procedure
INPUT_PARAMETER_COUNT	INTEGER		Input type parameter count
OUTPUT_PARAMETER_COUNT	INTEGER		Output type parameter count
INOUT_PARAMETER_COUNT	INTEGER		In-out type parameter count
RESULT_SET_COUNT	INTEGER		Result set count
IS_UNICODE	VARCHAR(5000)		Specifies whether the stored procedure contains Unicode or not: 'TRUE', 'FALSE'
DEFINITION	NCLOB		Query string of the stored procedure
PROCEDURE_TYPE	VARCHAR(5000)		Type of the stored procedure
READ_ONLY	VARCHAR(5000)		Specifies whether the procedure is read-only or not: 'TRUE', 'FALSE'

PROCEDURE_OBJECTS

Short description

Contains results of the system procedure GET_PROCEDURE_OBJECTS

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(128)		Schema name of the stored procedure
PROCEDURE_NAME	NVARCHAR(128)		Procedure name of the stored procedure
OBJECT_SCHEMA	NVARCHAR(128)		Schema name of the object in the stored procedure code
OBJECT_NAME	NVARCHAR(128)		Object name of the object in the stored procedure code
OBJECT_TYPE_ID	INTEGER		Type ID of the object in the stored procedure code
OBJECT_TYPE	VARCHAR(5000)		Type of the object in the stored procedure code
START_POSITION	INTEGER		Start position of the object in the stored procedure code
END_POSITION	INTEGER		End position of the object in the stored procedure code

PROCEDURE_PARAMETERS

Short description

Parameters of stored procedures

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name of the stored procedure
PROCEDURE_NAME	NVARCHAR(256)		Name of the stored procedure
PROCEDURE_OID	BIGINT		Object ID of the stored procedure
PARAMETER_NAME	NVARCHAR(256)		Parameter name
DATA_TYPE_ID	SMALLINT		Data type ID
DATA_TYPE_NAME	VARCHAR(5000)		Data type name
LENGTH	INTEGER		Parameter length
SCALE	INTEGER		Scale of the parameter
POSITION	INTEGER		Ordinal position of the parameter
TABLE_TYPE_SCHEMA	NVARCHAR(256)		Schema name of table type if DATA_TYPE_NAME is TABLE_TYPE
TABLE_TYPE_NAME	NVARCHAR(256)		Name of table type if DATA_TYPE_NAME is TABLE_TYPE
PARAMETER_TYPE	VARCHAR(5000)		Parameter mode: 'IN', 'OUT', 'INOUT'
HAS_DEFAULT_VALUE	VARCHAR(5000)		Specifies whether the parameter has a default value or not: 'TRUE', 'FALSE'
IS_NULLABLE	VARCHAR(5000)		Specifies whether the parameter accepts a null value: 'TRUE', 'FALSE'

QUERY_PLANS

Short description

Plans how to handle query execution

Structure

Column name	Data type	Unit	Description
PLAN_ID	VARCHAR(256)		Plan ID
OPERATOR_NAME	VARCHAR(5000)		Operator name
OPERATOR_DETAILS	NCLOB		Detailed information on operators in the query plan
OBJECT_NAME	NVARCHAR(256)		Object name
SUBTREE_COST	DOUBLE		Subtree cost
INPUT_CARDINALITY	DOUBLE		Input cardinality
OUTPUT_CARDINALITY	DOUBLE		Output cardinality
OPERATOR_ID	INTEGER		Operator ID
PARENT_OPERATOR_ID	INTEGER		Parent operator ID
LEVEL	INTEGER		Level
POSITION	INTEGER		Position
TIMESTAMP	TIMESTAMP		Timestamp
HOST	VARCHAR(256)		Host where the plan operator is executed
PORT	INTEGER		Port where the plan operator is executed

ROLES

Short description

Shows available roles

Structure

Column name	Data type	Unit	Description
ROLE_NAME	NVARCHAR(256)		Role name
ROLE_MODE	VARCHAR(5000)		Mode of the role: 'LOCAL', 'GLOBAL'
GLOBAL_IDENTITY	NVARCHAR(256)		Identity specified for role with ROLE_MODE GLOBAL
CREATOR	NVARCHAR(256)		Name of user who created the role
CREATE_TIME	TIMESTAMP		Time the role was created

SCHEMAS

Short description

Shows available schemas

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
SCHEMA_OWNER	NVARCHAR(256)		Schema owner
HAS_PRIVILEGES	VARCHAR(5000)		Shows if user is schema owner or has any privilege for the schema or any object within the schema: 'TRUE', 'FALSE'

SEQUENCES

Short description

Available sequences

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name of the sequence
SEQUENCE_NAME	NVARCHAR(256)		Name of the sequence
SEQUENCE_OID	BIGINT		Object ID of the sequence
START_NUMBER	BIGINT		Start number
MIN_VALUE	BIGINT		Minimum value of the sequence
MAX_VALUE	BIGINT		Maximum value of the sequence
INCREMENT_BY	BIGINT		Incremental value
IS_CYCLED	VARCHAR(5000)		Specifies whether the sequence will start with MIN_VALUE after having reached MAX_VALUE in case of INCREMENT_BY greater than 0 or start with MAX_VALUE after having reached MIN_VALUE in case of INCREMENT_BY less than 0: 'TRUE', 'FALSE'
RESET_BY_QUERY	NCLOB		Reset by query string for the sequence

SQLSCRIPT_TRACE

Short description

Contains the names of the local temporary tables created by the SQLScript TRACE operator

Structure

Column name	Data type	Unit	Description
TRACE_TIMESTAMP	TIMESTAMP		Timestamp generated when a TRACE is performed
TRANSACTION_ID	INTEGER		Transaction object ID
STATEMENT_ID	BIGINT		Logical ID of the outmost user statement, which triggers the procedure execution (direct or indirect)
PROCEDURE_OID	BIGINT		The procedure object ID
VARIABLE_NAME	NVARCHAR(256)		The internal variable name used for the variable to which the TRACE operator is assigned
TABLE_NAME	NVARCHAR(256)		The name of the local temporary table containing the table result

STATISTICS

Short description

Stores a histogram of tables. The histogram can be used by query optimizer to calculate size of an intermediate result

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
TABLE_NAME	NVARCHAR(256)		Table name
COLUMN_NAME	NVARCHAR(256)		Column name
POSITION	SMALLINT		Ordinal position of the column in the table
MINVALUE	DOUBLE		Min value of the column. The value is manipulated into the Double type value
MINVALUE_STR	NVARCHAR(256)		String representation of the column
MAXVALUE	DOUBLE		Max value of the column. The value is manipulated into the Double type value
MAXVALUE_STR	NVARCHAR(256)		String representation of the column
COUNT	DOUBLE		Number of not null value of the column
DCOUNT	DOUBLE		Number of distinct value of the column
NULLCOUNT	DOUBLE		Number of null values of the column

STRUCTURED_PRIVILEGES

Short description

Available structured privileges

Structure

Column name	Data type	Unit	Description
STRUCTURED_PRIVILEGE_NAME	NVARCHAR(256)		Name of the structured privilege
RESTRICTION_TYPE	NVARCHAR(256)		Type of the restriction: 'CUBERESTRICTION', 'ACTIVITYRESTRICTION', 'VALIDITYRESTRICTION', 'DIMENSIONRESTRICTION'
DIMENSION_ATTRIBUTE	NVARCHAR(256)		Name of the dimension attribute
FILTER_ID	BIGINT		Number of filter to be able to combine all operators/ operands belonging to one filter
NEGATED	VARCHAR(5000)		Operator was negated in filter; 'TRUE', 'FALSE'
OPERATOR	VARCHAR(5000)		Type of operator: 'CONTAINS PATTERN', 'BETWEEN', 'EQUAL', 'IN', 'LESS THAN', 'LESS EQUAL', 'GREATER THAN', 'GREATER EQUAL'
OPERAND_ORDER	INTEGER		Sequence of operands per filter ID
OPERAND	NVARCHAR(5000)		Value the operator will be compared to

SYNONYMS

Short description

Available synonyms

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name of the synonym
SYNONYM_NAME	NVARCHAR(256)		Name of the synonym
SYNONYM_OID	BIGINT		Object ID of the synonym
OBJECT_SCHEMA	NVARCHAR(256)		Schema name of the referenced object
OBJECT_NAME	NVARCHAR(256)		Name of the referenced object
OBJECT_TYPE	VARCHAR(5000)		Type of the referenced object
IS_COLUMN_OBJECT	VARCHAR(5000)		Specifies whether this view is a column object or not: 'TRUE', 'FALSE'

TABLES

Short description

Available tables

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
TABLE_NAME	NVARCHAR(256)		Table name
TABLE_OID	BIGINT		Object ID of the table
COMMENTS	NVARCHAR(5000)		Table description
FIXED_PART_SIZE	SMALLINT		Fixed part size of the table
IS_LOGGED	VARCHAR(5000)		Specifies whether logging is on for the table: 'TRUE', 'FALSE'
IS_SYSTEM_TABLE	VARCHAR(5000)		Specifies whether the table is a system table: 'TRUE', 'FALSE'
IS_COLUMN_TABLE	VARCHAR(5000)		Specifies whether the table is a column table: 'TRUE', 'FALSE'
TABLE_TYPE	VARCHAR(5000)		Specifies type of the table: 'ROW', 'COLUMN', 'HYBRID'
IS_INSERT_ONLY	VARCHAR(5000)		Specifies whether the table is an 'insert only' table: 'TRUE', 'FALSE'
IS_TENANT_SHARED_DATA	VARCHAR(5000)		Specifies whether the table can be shared among other instances: 'TRUE', 'FALSE'
IS_TENANT_SHARED_METADATA	VARCHAR(5000)		Specifies whether the table is a global table: 'TRUE', 'FALSE'
SESSION_TYPE	VARCHAR(5000)		Session type: 'NONE', 'SIMPLE', 'HISTORY'. For 'HISTORY', time travel is possible
IS_TEMPORARY	VARCHAR(5000)		Specifies whether the table is a temporary table: 'TRUE', 'FALSE'
TEMPORARY_TABLE_TYPE	VARCHAR(5000)		Temporary table type
IS_USER_DEFINED_TYPE	VARCHAR(5000)		'TRUE' if user defined table type, 'FALSE' otherwise
HAS_PRIMARY_KEY	VARCHAR(5000)		'TRUE' if the table has a primary key, else 'FALSE'. Only valid for column store tables
PARTITION_SPEC	NCLOB		Detail specification of table partitioning. Only valid for column store tables
USES_EXTKEY	VARCHAR(5000)		'TRUE' if the table uses an external key, else 'FALSE'. Only valid for column store tables
AUTO_MERGE_ON	VARCHAR(5000)		'TRUE' if automatic delta table merge is activated for the table, else 'FALSE'. Only valid for column store tables
USES_DIMFN_CACHE	VARCHAR(5000)		'TRUE' if DimFunctionCache feature is used by the

			table, else 'FALSE'. Only valid for column store tables
IS_PUBLIC	VARCHAR(5000)		'TRUE' if the table is public, else 'FALSE'. Only valid for column store tables
AUTO_OPTIMIZE_COMPRESSION_ON			
	VARCHAR(5000)		
COMPRESSED_EXTKEY	VARCHAR(5000)		'TRUE' if the table uses a compressed external key, else 'FALSE'. Only valid for column store tables
HAS_TEXT_FIELDS	VARCHAR(5000)		'TRUE' if the table has at least one column of type TEXT, else 'FALSE'. Only valid for column store tables
USES_QUEUE_TABLE	VARCHAR(5000)		'TRUE' if the table uses a queue table, else 'FALSE'. Only valid for column store tables
IS_PRELOAD	VARCHAR(5000)		'TRUE' if the table uses preloading, else 'FALSE'. Only valid for column store tables
IS_PARTIAL_PRELOAD	VARCHAR(5000)		'TRUE' if the table uses partial preloading, else 'FALSE'. Only valid for column store tables

TABLE_COLUMNS

Short description

Available table columns

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
TABLE_NAME	NVARCHAR(256)		Table name
TABLE_OID	BIGINT		Object ID of the table
COLUMN_NAME	NVARCHAR(256)		Name of the column
POSITION	INTEGER		Ordinal position of the column in a record
DATA_TYPE_ID	SMALLINT		SQL data type ID of the column
DATA_TYPE_NAME	VARCHAR(16)		SQL data type name of the column
OFFSET	SMALLINT		Offset of the column in the record
LENGTH	DECIMAL(34,-1)		Number of chars for char types, number of max digits for numeric types; number of chars for datetime types, number of bytes for LOB types
SCALE	INTEGER		Numeric types: the maximum number of digits to the right of the decimal point; time, timestamp: the decimal digits are defined as the number of digits to the right of the decimal point in the second's component of the data
IS_NULLABLE	VARCHAR(5000)		Specifies whether the column is allowed to accept null value: 'TRUE', 'FALSE'
DEFAULT_VALUE	NVARCHAR(5000)		Default value of the column
COLLATION	NVARCHAR(256)		Collation used for this column
COMMENTS	NVARCHAR(5000)		Description for this column
MAX_VALUE	NVARCHAR(5000)		Maximum value
MIN_VALUE	NVARCHAR(5000)		Minimum value
CS_DATA_TYPE_ID	INTEGER		Column store data type ID
CS_DATA_TYPE_NAME	VARCHAR(5000)		Column store data type name
DDIC_DATA_TYPE_ID	INTEGER		DDIC data type ID
DDIC_DATA_TYPE_NAME	VARCHAR(5000)		DDIC data type name
COMPRESSION_TYPE	VARCHAR(5000)		Type of compression: 'DEFAULT', 'PREFIXED', 'SPARSE', 'CLUSTERED', 'INDIRECT', 'RLE'. Columns in TABLE_COLUMNS show the default value which can be specified during table creation. They can also be updated using an 'ALTER TABLE' statement.
INDEX_TYPE	VARCHAR(5000)		Type of index: 'NONE', 'FULL', 'SIGNATURE'
COLUMN_ID	BIGINT		ID of the column
PRELOAD	VARCHAR(5000)		Specifies if column is preloaded: 'TRUE', 'FALSE'

TABLE_COLUMNS_ODBC

Short description

Available table columns

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
TABLE_NAME	NVARCHAR(256)		Table name
TABLE_OID	BIGINT		Object ID of the table
COLUMN_NAME	NVARCHAR(256)		Name of the column
DATA_TYPE_ID	SMALLINT		SQL data type ID of the column
DATA_TYPE_NAME	VARCHAR(16)		SQL data type name of the column
COLUMN_SIZE	DECIMAL(34,-1)		Total number of characters required to display the value when it is converted to characters
BUFFER_LENGTH	INTEGER		Length in bytes required to transfer the value
DECIMAL_DIGITS	INTEGER		Total number of significant digits to the right of the decimal point
NUM_PREC_RADIX	INTEGER		For numeric data types either 10 or 2. Specifies how to interpret columns COLUMN_SIZE and DECIMAL_DIGITS
NULLABLE	VARCHAR(5000)		Specifies whether the column is allowed to accept null value: 'TRUE', 'FALSE'
COLUMN_DEF	NVARCHAR(5000)		Default value of the column
SQL_DATA_TYPE	SMALLINT		SQL data type ID of the column
SQL_DATETIME_SUB	SMALLINT		Subtype code for datetime data types
CHAR_OCTET_LENGTH	INTEGER		Maximum length in bytes of a character or binary data type column
POSITION	INTEGER		Ordinal position of the column in a record
COMMENTS	NVARCHAR(5000)		Description for this column

TRANSACTION_HISTORY

Short description

Committed transactions and their users

Structure

Column name	Data type	Unit	Description
COMMIT_TIME	TIMESTAMP		Timestamp
COMMIT_ID	BIGINT		ID of the committed transaction
APP_USER	NVARCHAR(256)		User that used the transaction

TRIGGERS

Short description

Triggers defined for tables

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name of the trigger
TRIGGER_NAME	NVARCHAR(256)		Trigger name
TRIGGER_OID	BIGINT		Object ID of the trigger
OWNER_NAME	NVARCHAR(256)		Name of the trigger owner
OWNER_OID	BIGINT		Object ID of the trigger owner
SUBJECT_TABLE_SCHEMA	NVARCHAR(256)		Schema name of the table, the trigger is defined for
SUBJECT_TABLE_NAME	NVARCHAR(256)		Table name of the table, the trigger is defined for
TRIGGER_ACTION_TIME	VARCHAR(5000)		Time the trigger is executed: 'BEFORE', 'AFTER', 'INSTEAD OF' the specified event
TRIGGER_EVENT	VARCHAR(5000)		Event, the trigger is defined for: 'INSERT', 'UPDATE', 'DELETE'
TRIGGERED_ACTION_LEVEL	VARCHAR(5000)		Level of the event, where the triggered action will happen: 'ROW', 'STATEMENT'
DEFINITION	NCLOB		Query string of the trigger

USERS

Short description

All users

Structure

Column name	Data type	Unit	Description
USER_NAME	NVARCHAR(256)		Name of the user
USER_MODE	VARCHAR(5000)		Mode of the user: 'LOCAL', 'GLOBAL', 'EXTERNAL'
EXTERNAL_IDENTITY	NVARCHAR(256)		External identity of the user
CREATOR	NVARCHAR(256)		Creator of the user
CREATE_TIME	TIMESTAMP		Creation time
LAST_SUCCESSFUL_CONNECT	TIMESTAMP		Time of the last successful connect of the user
LAST_INVALID_CONNECT_ATTEMPT			
	TIMESTAMP		Time of the last invalid connect attempt
INVALID_CONNECT_ATTEMPTS	INTEGER		Number of invalid connect attempts since last successful connect
ADMIN_GIVEN_PASSWORD	VARCHAR(5000)		Specifies if the password was given by administrator or by the user himself: 'TRUE', 'FALSE'
PASSWORD_CHANGE_TIME	TIMESTAMP		Time of last password change, NULL, if password-lifetime will not be checked for this user
PASSWORD_CHANGE_NEEDED	VARCHAR(5000)		Specifies if user is forced to change his password himself: 'TRUE', 'FALSE'
USER_DEACTIVATED	VARCHAR(5000)		Specifies if user is dectivated: 'TRUE', 'FALSE'
DEACTIVATION_TIME	TIMESTAMP		Time given with explicit deactivation command for this user

USER_PARAMETERS

Short description

All parameters and their values, which have been assigned to users in the system (using CREATE USER ... SET PARAMETER or ALTER USER ... SET PARAMETER)

Structure

Column name	Data type	Unit	Description
USER_NAME	NVARCHAR(256)		Name of the user
PARAMETER	NVARCHAR(256)		Parameter name: CLIENT
VALUE	NVARCHAR(256)		Parameter value

VIEWS

Short description

Available views

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
VIEW_NAME	NVARCHAR(256)		View name
VIEW_OID	BIGINT		Object ID of the view
IS_UNICODE	VARCHAR(5000)		Specifies whether the query string contains Unicode: 'TRUE', 'FALSE'
IS_READ_ONLY	VARCHAR(5000)		Specifies whether this view is read-only or updatable: 'TRUE', 'FALSE'
HAS_CHECK_OPTION	VARCHAR(5000)		Specifies whether this view has an updatable view condition: 'TRUE', 'FALSE'
HAS_COLUMN_ALIASES	VARCHAR(5000)		Specifies whether the view has a columns alias : 'TRUE', 'FALSE'
DEFINITION	NCLOB		View definition
COMMENTS	NVARCHAR(5000)		View description
IS_COLUMN_VIEW	VARCHAR(5000)		Specifies whether this view is a column view or not: 'TRUE', 'FALSE'
VIEW_TYPE	VARCHAR(5000)		Type of view : 'ROW', 'OLAP', 'JOIN', 'HIERARCHY', 'CALC'
IS_TENANT_SHARED_METADATA	VARCHAR(5000)		

VIEW_COLUMNS

Short description

Available view columns

Structure

Column name	Data type	Unit	Description
SCHEMA_NAME	NVARCHAR(256)		Schema name
VIEW_NAME	NVARCHAR(256)		View name
VIEW_OID	BIGINT		Object ID of the view
COLUMN_NAME	NVARCHAR(256)		View column name
POSITION	INTEGER		Ordinal position of the view column
DATA_TYPE_ID	SMALLINT		Data type ID
DATA_TYPE_NAME	VARCHAR(16)		Data type name
OFFSET	SMALLINT		Offset of the column in a record
LENGTH	DECIMAL(34,-1)		Number of chars for character types; max number of digits for numeric types; number of chars for datetime types; number of bytes for LOB types
SCALE	INTEGER		Numeric types: the maximum number of digits to the right of the decimal point; time, timestamp: the decimal digits are defined as the number of digits to the right of the decimal point in the second's component of the data
IS_NULLABLE	VARCHAR(5000)		Specifies whether the column is allowed to accept null value: 'TRUE', 'FALSE'
DEFAULT_VALUE	NVARCHAR(5000)		Default value
COLLATION	NVARCHAR(256)		Collation
COMMENTS	NVARCHAR(5000)		Description for this column
MAX_VALUE	NVARCHAR(5000)		Maximum value
MIN_VALUE	NVARCHAR(5000)		Minimum value
CS_DATA_TYPE_ID	INTEGER		Column store data type ID
CS_DATA_TYPE_NAME	VARCHAR(5000)		Column store data type name
DDIC_DATA_TYPE_ID	INTEGER		DDIC data type ID
DDIC_DATA_TYPE_NAME	VARCHAR(5000)		DDIC data type name
COMPRESSION_TYPE	VARCHAR(5000)		Type of compression: 'DEFAULT', 'PREFIXED', 'SPARSE', 'CLUSTERED', 'INDIRECT', 'RLE'
INDEX_TYPE	VARCHAR(5000)		Type of index: 'NONE', 'FULL', 'SIGNATURE'
COLUMN_ID	BIGINT		ID of the column
PRELOAD	VARCHAR(5000)		Specifies if column is preloaded: 'TRUE', 'FALSE'

Statistics Server Tables

The following tables are from the _SYS_STATISTICS schema.

GLOBAL_COLUMN_TABLES_SIZE

Short description

Information on all column tables

Structure

Column name	Data type	Unit	Description
SNAPSHOT_ID	TIMESTAMP		ID of the snaphot
DATA_SEQ	INTEGER		Data seqence ID of the statistics
SERVER_TIMESTAMP	TIMESTAMP		Server timestamp
LAST_COMPRESSED_RECORD_COUNT	BIGINT		The number of entries in main during last compression run
MEMORY_SIZE_IN_DELTA	BIGINT		Current memory consumption in delta
MEMORY_SIZE_IN_HISTORY_DELTA	BIGINT		Current memory consumption in history-delta
MEMORY_SIZE_IN_HISTORY_MAIN	BIGINT		Current memory consumption in history-main
MEMORY_SIZE_IN_MAIN	BIGINT		Current memory consumption in main
MEMORY_SIZE_IN_TOTAL	BIGINT		Total memory size is the sum of memory size in main, delta, and history parts
PART_COUNT	BIGINT		Count of partitions of a table
RAW_RECORD_COUNT_IN_DELTA	BIGINT		Raw record count in delta
RAW_RECORD_COUNT_IN_HISTORY_DELTA	BIGINT		Raw record count in history-delta
RAW_RECORD_COUNT_IN_HISTORY_MAIN	BIGINT		Raw record count in main
RAW_RECORD_COUNT_IN_MAIN	BIGINT		Raw record count in history-main
RECORD_COUNT	BIGINT		Count of records
SCHEMA_NAME	NVARCHAR	256	Schema name
TABLE_NAME	NVARCHAR	256	Table name
ESTIMATED_MAX_MEMORY_SIZE_IN_TOTAL	BIGINT		Estimated maximum memory consumption in total for table fully loaded
TABLE_COLUMN_COUNT	BIGINT		Count of table columns

GLOBAL_INTERNAL_DISKFULL_EVENTS

Short description

Global Internal disk full events

Structure

Column name	Data type	Unit	Description
SNAPSHOT_ID	TIMESTAMP		ID of the snaphot
DATA_SEQ	INTEGER		Data seqence ID of the statistics
SERVER_TIMESTAMP	TIMESTAMP		Server timestamp
CREATE_TIME	TIMESTAMP		Time event was created
HANDLE_TIME	TIMESTAMP		Time event was handled
HOST	VARCHAR	64	Host name
ID	BIGINT		ID of event
INFO	VARCHAR	2000	Additional information
PORT	INTEGER		Internal port
STATE	VARCHAR	256	State of event
TYPE	VARCHAR	256	Type of event

GLOBAL_INTERNAL_EVENTS

Short description

Global internal events

Structure

Column name	Data type	Unit	Description
SNAPSHOT_ID	TIMESTAMP		ID of the snaphot
DATA_SEQ	INTEGER		Data seqence ID of the statistics
SERVER_TIMESTAMP	TIMESTAMP		Server timestamp
CREATE_TIME	TIMESTAMP		Time event was created
HANDLE_TIME	TIMESTAMP		Time event was handled
HOST	VARCHAR	64	Host name
ID	BIGINT		ID of event
INFO	VARCHAR	2000	Additional information
PORT	INTEGER		Internal port
STATE	VARCHAR	256	State of event
TYPE	VARCHAR	256	Type of event

GLOBAL_PERSISTENCE_STATISTICS

Short description

Information on all available persistence

Structure

Column name	Data type	Unit	Description
SNAPSHOT_ID	TIMESTAMP		ID of the snaphot
DATA_SEQ	INTEGER		Data seqence ID of the statistics
SERVER_TIMESTAMP	TIMESTAMP		Server timestamp
TOTAL_DISKS_SIZE	BIGINT		Total disk size available
TOTAL_DISKS_USED_SIZE	BIGINT		Total disk size used
TOTAL_VOLUMES_DATA_SIZE	BIGINT		Total log area size
TOTAL_VOLUMES_LOG_SIZE	BIGINT		Total data area size

HOST_BLOCKED_TRANSACTIONS

Short description

Information on currently blocked transactions per host

Structure

Column name	Data type	Unit	Description
SNAPSHOT_ID	TIMESTAMP		ID of the snaphot
SERVER_TIMESTAMP	TIMESTAMP		Server timestamp
BLOCKED_TIME	TIMESTAMP		Blocked timestamp
BLOCKED_TRANSACTION_ID	INTEGER		Transaction ID of the transaction waiting for a lock
BLOCKED_UPDATE_TRANSACTION_ID	BIGINT		Transaction ID of the blocked update transaction
HOST	VARCHAR	64	Host name
LOCK_MODE	VARCHAR	32	Lock mode: 'SHARED'/'EXCLUSIVE'/'INTENTIONAL EXCLUSIVE'
LOCK_OWNER_TRANSACTION_ID	INTEGER		Transaction ID of the transaction holding the lock
LOCK_OWNER_UPDATE_TRANSACTION_ID	BIGINT		Transaction ID of the update transaction holding the lock
LOCK_TYPE	VARCHAR	32	Lock type: 'RECORD'/'TABLE'/'METADATA'
PORT	INTEGER		Internal port
WAITING_RECORD_ID	VARCHAR	256	ID of the record on which the lock is currently placed
WAITING_SCHEMA_NAME	VARCHAR	32	Name of the schema on which the lock is currently placed
WAITING_TABLE_NAME	VARCHAR	32	Name of the table on which the lock is currently placed

HOST_DATA_VOLUME_PAGE_STATISTICS

Short description

Data volume page information per host

Structure

Column name	Data type	Unit	Description
SNAPSHOT_ID	TIMESTAMP		ID of the snaphot
SERVER_TIMESTAMP	TIMESTAMP		Server timestamp
FILE_NAME	VARCHAR	256	File Name
FILL_RATIO	DOUBLE	15	Fill ratio
HOST	VARCHAR	64	Host name
INITIAL_BLOCK_COUNT	BIGINT		Count of init pages
PAGE_SIZE	INTEGER		Page size
PAGE_SIZE_CLASS	VARCHAR	32	Page size class
PORT	INTEGER		Internal port
SHADOW_BLOCK_COUNT	BIGINT		Count of shadow blocks
SUPERBLOCK_COUNT	BIGINT		Number of used Superblocks
SUPERBLOCK_SIZE	INTEGER		Superblock size
TOTAL_ALLOCATE_BLOCK_COUNT	BIGINT		Number of single and group allocated blocks
TOTAL_SET_BLOCK_FREE_AFTER_SAVEPOINT_COUNT	BIGINT		Number of single and group freed-after-savepoint blocks
TOTAL_SET_BLOCK_FREE_COUNT	BIGINT		Number of single and group freed blocks
USED_BLOCK_COUNT	BIGINT		Count of used blocks
VOLUME_ID	INTEGER		Persistence Volume ID

HOST_DATA_VOLUME_SUPERBLOCK_STATISTICS

Short description

Data volume superblock information per host

Structure

Column name	Data type	Unit	Description
SNAPSHOT_ID	TIMESTAMP		ID of the snaphot
SERVER_TIMESTAMP	TIMESTAMP		Server timestamp
DATA_VOLUME_NAME	VARCHAR	256	DataVolume name
FILL_RATIO	DOUBLE	15	Fill ratio
HOST	VARCHAR	64	Host name
PORT	INTEGER		Internal port
SUPERBLOCK_COUNT	BIGINT		Count of Superblocks
SUPERBLOCK_SIZE	BIGINT		Superblock size
USED_SUPERBLOCK_COUNT	BIGINT		Count of used Superblocks
VOLUME_ID	INTEGER		Persistence Volume ID

HOST_DELTA_MERGE_STATISTICS

Short description

Table delta merge statistics per host

Structure

Column name	Data type	Unit	Description
SNAPSHOT_ID	TIMESTAMP		ID of the snaphot
SERVER_TIMESTAMP	TIMESTAMP		Server timestamp
EXECUTION_TIME	BIGINT		Execution duration
HOST	VARCHAR	64	Host name
MEMORY_MERGE	VARCHAR	6	Flag to indicate table was merged in memory only
MERGED_DELTA_RECORDS	INTEGER		Number of documents in delta
MOTIVATION	VARCHAR	8	HARD, SMART, AUTO, FORCE
PART_ID	INTEGER		Partition number (0 for non-partitioned tables and 1 through number of partitions for partitioned tables
PASSPORT	VARCHAR	256	External identifier for table merge called by an application
PORT	INTEGER		Internal port
SCHEMA_NAME	VARCHAR	256	Schema name
START_TIME	TIMESTAMP		Execution start time
SUCCESS	VARCHAR	6	Call success flag; depends on field TYPE; HINT: application merge hint was accepted/rejected; MERGE/SPARSE: delta merge/optimize compression was completed with or without success
TABLE_NAME	VARCHAR	256	Table name
TYPE	VARCHAR	7	MERGE (table delta merge), HINT (application merge hint), SPARSE (optimize compression)

HOST_HEAP_ALLOCATORS

Short description

Memory allocator statistics per host

Structure

Column name	Data type	Unit	Description
SNAPSHOT_ID	TIMESTAMP		ID of the snaphot
SERVER_TIMESTAMP	TIMESTAMP		Server timestamp
CATEGORY	VARCHAR	128	Allocator name
DEPTH	BIGINT		Depth
EXCLUSIVE_ALLOC_ERRORS	BIGINT		Count of allocation errors
EXCLUSIVE_ALLOCATED_COUNT	BIGINT		Count of allocations
EXCLUSIVE_ALLOCATED_SIZE	BIGINT		Total allocated size in this allocator
EXCLUSIVE_COUNT_IN_USE	BIGINT		Number of blocks currently in use
EXCLUSIVE_DEALLOCATED_COUNT	BIGINT		Count of deallocations
EXCLUSIVE_DEALLOCATED_SIZE	BIGINT		Total deallocated size in this allocator
EXCLUSIVE_MAX_SINGLE_ALLOCATION_SIZE	BIGINT		Maximum ever allocated block size in this allocator
EXCLUSIVE_PEAK_ALLOCATION_SIZE	BIGINT		Maximum size of this allocator (estimate)
EXCLUSIVE_SIZE_IN_USE	BIGINT		Current size of this allocator
FLAGS	VARCHAR	64	Allocator flags
HOST	VARCHAR	64	Host name
INCLUSIVE_ALLOCATED_COUNT	BIGINT		Count of allocations, including suballocators
INCLUSIVE_ALLOCATED_SIZE	BIGINT		Total allocated size in this allocator and suballocators
INCLUSIVE_COUNT_IN_USE	BIGINT		Number of blocks currently in use, including suballocators
INCLUSIVE_DEALLOCATED_COUNT	BIGINT		Count of deallocations, including suballocators
INCLUSIVE_DEALLOCATED_SIZE	BIGINT		Total deallocated size in this allocator and suballocators
INCLUSIVE_MAX_SINGLE_ALLOCATION_SIZE	BIGINT		Maximum ever allocated block size in this allocator and suballocators
INCLUSIVE_PEAK_ALLOCATION_SIZE	BIGINT		Maximum size of this allocator and suballocators (estimate)
INCLUSIVE_SIZE_IN_USE	BIGINT		Current size of this allocator, including suballocators
MALLOC_PROXY_CACHE_MISSES	BIGINT		Count of malloc proxy cache misses
PORT	INTEGER		Internal port
STATISTICS_ID	BIGINT		Statistics object unique ID
VOLUME_ID	INTEGER		Persistence Volume ID

HOST_LONG_RUNNING_STATEMENTS

Short description

Information on long running system operations per host

Structure

Column name	Data type	Unit	Description
SNAPSHOT_ID	TIMESTAMP		ID of the snaphot
SERVER_TIMESTAMP	TIMESTAMP		Server timestamp
APPLICATION_USER_NAME	NVARCHAR	256	Application user
AUTO_COMMIT	VARCHAR	6	Commit mode of the current transaction: TRUE if the current connection is in auto-commit mode, FALSE otherwise
CLIENT_HOST	VARCHAR	64	Host name of client machine
CLIENT_IP	VARCHAR	16	IP of client machine
CLIENT_PID	BIGINT		Client Process ID
CONNECTION_ID	INTEGER		Connection ID
CONNECTION_START_TIME	TIMESTAMP		Connection start timestamp
DURATION	BIGINT		Connectino duration
HOST	VARCHAR	64	Host name
PORT	INTEGER		Internal port
THREAD_ID	BIGINT		Thread ID
TRANSACTION_ID	INTEGER		Transaction ID
USER_NAME	NVARCHAR	256	Name of user who triggered the statement

HOST_ONE_DAY_FILE_COUNT

Short description

Information on trace files accumulated per host

Structure

Column name	Data type	Unit	Description
SNAPSHOT_ID	TIMESTAMP		ID of the snaphot
SERVER_TIMESTAMP	TIMESTAMP		Server timestamp
FILE_TYPE	VARCHAR	128	File type
HOST	VARCHAR	64	Host name
NUM_FILES	BIGINT		Number of trace files
SUM_SIZE	BIGINT		Total size of trace files

HOST_RESOURCE_UTILIZATION_STATISTICS

Short description

Host resource utilization

Structure

Column name	Data type	Unit	Description
SNAPSHOT_ID	TIMESTAMP		ID of the snaphot
SERVER_TIMESTAMP	TIMESTAMP		Server timestamp
FREE_PHYSICAL_MEMORY	BIGINT		Free physical memory on the host
FREE_SWAP_SPACE	BIGINT		Free swap memory on the host
HOST	VARCHAR	64	Host name
TOTAL_CPU_IDLE_TIME	BIGINT		CPU idle time
TOTAL_CPU_SYSTEM_TIME	BIGINT		CPU time spent in kernel mode
TOTAL_CPU_USER_TIME	BIGINT		CPU time spent in user mode
TOTAL_CPU_WIO_TIME	BIGINT		CPU time spent in wait IO (Linux only, Windows always 0)
USED_PHYSICAL_MEMORY	BIGINT		Used physical memory on the host
USED_SWAP_SPACE	BIGINT		Used swap memory on the host

HOST_SERVICE_MEMORY

Short description

Service memory usage per host

Structure

SNAPSHOT_ID	TIMESTAMP		ID of the snaphot
SERVER_TIMESTAMP	TIMESTAMP		Server timestamp
CODE_SIZE	BIGINT		Process code size
COMPACTORS_ALLOCATED_SIZE	BIGINT		Total memory allocated by compactors
COMPACTORS_FREEABLE_SIZE	BIGINT		Memory which can be freed on demand
HEAP_MEMORY_ALLOCATED_SIZE	BIGINT		Process heap memory allocated size
HEAP_MEMORY_USED_SIZE	BIGINT		Process heap memory used
HOST	VARCHAR	64	Host name
LOGICAL_MEMORY_SIZE	BIGINT		Process virtual memory size
PHYSICAL_MEMORY_SIZE	BIGINT		Process physical memory size
PORT	INTEGER		Internal port
PROCESS_ID	BIGINT		Process ID
SERVICE_NAME	VARCHAR	256	Service name
SHARED_MEMORY_ALLOCATED_SIZE	BIGINT		Process shared memory allocated (row store only)
SHARED_MEMORY_USED_SIZE	BIGINT		Process shared memory used (row store only)
STACK_SIZE	BIGINT		Process stack size

HOST_SERVICE_STATISTICS

Short description

Memory utilization by services per host

Structure

Column name	Data type	Unit	Description
SNAPSHOT_ID	TIMESTAMP		ID of the snaphot
SERVER_TIMESTAMP	TIMESTAMP		Server timestamp
ACTIVE_REQUEST_COUNT	INTEGER		Number of active requests
ACTIVE_STATUS	VARCHAR	128	NO, YES, UNKNOWN, STARTING, STOPPING
ACTIVE_THREAD_COUNT	INTEGER		Number of active threads
ALL_FINISHED_REQUEST_COUNT	BIGINT		Finished requests including internal requests
AVAILABLE_MEMORY	BIGINT		Host physical+swap memory size
DETAIL	VARCHAR	128	Detail information, similar to COORDINATOR_TYPE in M_SERVICES
FINISHED_NON_INTERNAL_REQUEST_COUNT	BIGINT		Finished requests
HOST	VARCHAR	64	Host name
OPEN_FILE_COUNT	INTEGER		Number of open files
PENDING_REQUEST_COUNT	INTEGER		Number of pending requests
PHYSICAL_MEMORY	BIGINT		Host physical memory size
PORT	INTEGER		Internal port
PROCESS_CPU_TIME	BIGINT		CPU usage of current process since start
PROCESS_ID	BIGINT		Process ID
PROCESS_MEMORY	BIGINT		Process logical memory usage
PROCESS_PHYSICAL_MEMORY	BIGINT		Process physical memory usage
REQUESTS_PER_SEC	DOUBLE	15	Requests per second. Average over last 1000 requests
RESPONSE_TIME	INTEGER		Request response time. Average over last 1000 requests
SERVICE_NAME	VARCHAR	256	Service name
START_TIME	TIMESTANP		Process start time
THREAD_COUNT	INTEGER		Number of total threads
TOTAL_MEMORY	BIGINT		CPU usage of all threads

HOST_VOLUME_FILES

Short description

Information about volume files per host

Structure

Column name	Data type	Unit	Description
SNAPSHOT_ID	TIMESTAMP		ID of the snaphot
SERVER_TIMESTAMP	TIMESTAMP		Server timestamp
FILE_NAME	VARCHAR	256	File name
FILE_TYPE	VARCHAR	128	Type of file (DATA, LOG, TRACER, ...)
HOST	VARCHAR	64	Host name
PORT	INTEGER		Internal port
TOTAL_SIZE	BIGINT		Total file size
USED_SIZE	BIGINT		Size of used data within file
VOLUME_ID	INTEGER		Volume ID

HOST_VOLUME_IO_STATISTICS

Short description

File access statistics per host

Structure

Column name	Data type	Unit	Description
SNAPSHOT_ID	TIMESTAMP		ID of the snaphot
SERVER_TIMESTAMP	TIMESTAMP		Server timestamp
CLOSE_CALL_COUNT	BIGINT		Count of close calls
ENQUEUED_WRITE_REQUESTS	BIGINT		Count of enqueued write requests
EOF_READS	BIGINT		Count of EOF reads
EXISTS_CALL_COUNT	BIGINT		Count of exists calls
HOST	VARCHAR	64	Host name
OPEN_CALL_COUNT	BIGINT		Count of open calls
PATH	VARCHAR	256	Filesystem path
PORT	INTEGER		Internal port
REMOVE_CALL_COUNT	BIGINT		Count of remove calls
RESIZE_CALL_COUNT	BIGINT		Count of resize calls
SYNC_CALL_COUNT	BIGINT		Count of sync calls
TOTAL_FULL_RETRY_READS	BIGINT		Count of full retry reads
TOTAL_FULL_RETRY_WRITES	BIGINT		Count of full retry writes
TOTAL_IO_TIME	BIGINT		Total I/O time
TOTAL_READ_SIZE	BIGINT		Size of read data
TOTAL_SHORT_READS	BIGINT		Count of short reads
TOTAL_SHORT_WRITES	BIGINT		Count of short writes
TOTAL_WRITE_SIZE	BIGINT		Size of written data
TYPE	VARCHAR	16	Type of contained files
VOLUME_ID	INTEGER		Persistence Volume ID

STATISTICS_ALERTS

Short description

Information on alerts and their status

Structure

Column name	Data type	Unit	Description
SNAPSHOT_ID	TIMESTAMP		ID of the snaphot
HOST	VARCHAR	64	Host name
SQL_PORT	INTEGER		SQL port
ALERT_ID	INTEGER		Alert ID
ALERT_SEQ	INTEGER		Alert Sequence
ALERT_TIMESTAMP	TIMESTAMP		Timestamp of alert
ALERT_RATING	TINYINT	3	Alert rating (1-5)
ALERT_DETAILS	NVARCHAR	1024	Information on the alert
EMAIL_STATE	TINYINT	3	Email sent or not; values 1 or 0
PROCESS_STATE	TINYINT	3	State of the process
PROCESS_TIME	TIMESTAMP		Timestamp of the process
PROCESS_AGENT	NVARCHAR	256	User owing the process

STATISTICS_ALERT_INFORMATION

Short description

Information on alerts, possible resolutions and polling intervals

Structure

Column name	Data type	Unit	Description
ALERT_ID	INTEGER		Alert ID
ALERT_NAME	NVARCHAR	64	Name of the Alert
ALERT_DESCRIPTION	NVARCHAR	1024	Description of the alert
ALERT_USERACTION	NVARCHAR	1024	Hint to user on actions to take in response to the alert
INTERVAL	NVARCHAR	128	Interval between polling

STATISTICS_ALERT_LAST_CHECK_INFORMATION

Short description

Information from the last alert poll

Structure

Column name	Data type	Unit	Description
SNAPSHOT_ID	TIMESTAMP		ID of the snaphot
HOST	VARCHAR	64	Host name
SQL_PORT	INTEGER		SQL port
ALERT_ID	INTEGER		Alert ID
ALERT_SEQ	INTEGER		Alert Sequence
ALERT_LAST_CHECK_TIMESTAMP	TIMESTAMP		Timestamp of last alert check
ALERT_LAST_CHECK_RATING	TINYINT	3	Rating of last alert check
ALERT_LAST_CHECK_EMAIL	TINYINT	3	Email sent on last alert check

STATISTICS_INTERVAL_INFORMATION

Short description

Information on polling interval durations and last start times

Structure

Column name	Data type	Unit	Description
NAME	NVARCHAR	128	Name of interval
DURATION	INTEGER		Duration of internal in seconds
START_TIME	TIMESTAMP		Start time of interval

STATISTICS_LASTVALUES

Short description

Information on the last recorded status per process

Structure

Column name	Data type	Unit	Description
HOST	VARCHAR	64	Host name
PORT	INTEGER		Internal port
NAME	NVARCHAR	128	Process name
INDEX	NVARCHAR	128	
REACHED_AT	TIMESTAMP		Timestamp of last value
VALUE	NVARCHAR	1024	

STATISTICS_STATE

Short description

Information on current state of statistics process

Structure

Column name	Data type	Unit	Description
STATE	NVARCHAR	64	State of the process (Starting, Stopping, Running, Stopped)
REACHED_AT	TIMESTAMP		Time the state was reached

STATISTICS_VERSION

Short description

Information on version of the statistics

Structure

Column name	Data type	Unit	Description
VERSION	TINYINT	3	Statistics version
SINCE	TIMESTAMP		Active since, timestamp