SAP HANA Developer Guide

Generated on: 2024-11-26 03:44:09 GMT+0000

SAP HANA Platform | 2.0 SPS 08

PUBLIC

Original content: https://help.sap.com/docs/SAP_HANA_PLATFORM/52715f71adba4aaeb480d946c742d1f6?locale=en-US&state=PRODUCTION&version=2.0.08

Warning

This document has been generated from the SAP Help Portal and is an incomplete version of the official SAP product documentation. The information included in custom documentation may not reflect the arrangement of topics in the SAP Help Portal, and may be missing important aspects and/or correlations to other topics. For this reason, it is not for productive use.

For more information, please visit the https://help.sap.com/docs/disclaimer.

Privileges

Several privilege types are used in SAP HANA (system, object, analytic, package, and application).

Privilege Type	Applicable To	Target User	Description
System privilege 9.1	System, database	Administrators, developers 9.1	System privileges control general system activities. They are mainly used for administrative purposes, such as creating schemas, creating and changing users and roles, monitoring and tracing. 9.1 System privileges are also used to authorize basic repository operations. System privileges granted to users in a particular tenant database authorize operations in that database only. The only exception is the system privileges DATABASE ADMIN, DATABASE STOP, DATABASE START, and DATABASE AUDIT ADMIN. These system privileges can only be granted to users of the system database. They authorize the execution of operations on individual tenant databases. For example, a user with DATABASE ADMIN can create and drop tenant databases, change the database-specific properties in configuration (*.ini) files, and perform database-specific backups.
Object privilege	Database objects (schemas, tables, views, procedures and so on)	End users, technical users	Object privileges are used to allow access to and modification of database objects, such as tables and views. Depending on the object type, different actions can be authorized (for example, SELECT, CREATE ANY, ALTER, DROP). Schema privileges are object privileges that are used to allow access to and modification of schemas and the objects that they contain. Source privileges are object privileges that are used to restrict access to and modification of remote data sources, which are connected through SAP HANA smart data access. Object privileges granted to users in a particular database authorize access to and modification of database objects in that database only. That is, unless cross-database access has been enabled for the user. This is made possible through the association of the requesting user with a remote identity on the remote database. For more information, see *Cross-Database Authorization in Tenant Databases* in the *SAP HANA Security Guide*.
Analytic privilege	Analytic views	End users	Analytic privileges are used to allow read access to data in SAP HANA information models (that is, analytic views, attribute views, and calculation views) depending on certain values or combinations of values. Analytic privileges are evaluated during query processing.

Privilege Type	Applicable To	Target User	Description
			Analytic privileges granted to users in a particular database authorize access to information models in that database only.
Package privilege	Packages in the classic repository of the SAP HANA database	Application and content developers working in the classic SAP HANA repository	Package privileges are used to allow access to and the ability to work in packages in the classic repository of the SAP HANA database. Packages contain design time versions of various objects, such as analytic views, attribute views, calculation views, and analytic privileges. Package privileges granted to users in a particular database authorize access to and the ability to work in packages in the repository of that database only. **ⓘ Note** With SAP HANA XS advanced, source code and web content are not versioned and stored in the SAP HANA database, so package privileges are not used in this context. For more information, see *Authorization in SAP HANA XS Advanced*.
Package privilege	Packages in the classic repository of the SAP HANA database	Application and content developers working in the classic SAP HANA repository	Package privileges are not relevant in the SAP HANA service for SAP BTP context as the SAP HANA repository is not supported.
Application privilege	SAP HANA XS classic applications	Application end users, technical users (for SQL connection configurations)	Developers of SAP HANA XS classic applications can create application privileges to authorize user and client access to their application. They apply in addition to other privileges, for example, object privileges on tables. Application privileges can be granted directly to users or roles in runtime in the SAP HANA studio. However, it is recommended that you grant application privileges to roles created in the repository in design time. **ⓘ Note** With SAP HANA XS advanced, application privileges are not used. Application-level authorization is implemented using OAuth and authorization scopes and attributes. For more information, see *Authorization in SAP HANA XS Advanced*.
Application privilege	SAP HANA XS classic applications	Application end users, technical users (for SQL connection configurations)	Application privileges are not relevant in the SAP HANA service for SAP BTP context as SAP HANA XS classic is not supported.

ⓘ Note

There are no HDI or XS advanced equivalents in the SAP HANA authorization concept for package privileges on repository packages and applications privileges on SAP HANA XS classic applications. For more information about the authorization

concept of XS advanced, see the *SAP HANA Security Guide*.

Privileges on Users

An additional privilege type, privileges on users, can be granted to users. Privileges on users are SQL privileges that users can grant on their user. ATTACH DEBUGGER is the only privilege that can be granted on a user.

For example, User A can grant User B the privilege ATTACH DEBUGGER to allow User B debug SQLScript code in User A's session. User A is only user who can grant this privilege. Note that User B also needs the object privilege DEBUG on the relevant SQLScript procedure.

For more information, see the section on debugging procedures in the *SAP HANA Developer Guide*.

Related Information

GRANT
Cross-Database Authorization in Tenant Databases
Authorization in SAP HANA XS Advanced
Debug an External Session
Recommendations for Database Users, Roles, and Privileges

System Privileges

System privileges control general system activities.

System privileges are mainly used to authorize users to perform administrative actions, including:

- Creating and deleting schemas
- Managing users and roles
- Performing data backups
- Monitoring and tracing
- Managing licenses

System privileges are also used to authorize basic repository operations, for example:

- Importing and exporting content
- Maintaining delivery units (DU)

System privileges granted to users in a particular database authorize operations in that database only. The only exception is the system privileges DATABASE ADMIN, DATABASE STOP, and DATABASE START, DATABASE AUDIT ADMIN. These system privileges can only be granted to users of the system database. They authorize the execution of operations on individual tenant databases. For example, a user with DATABASE ADMIN can create and drop tenant databases, change the database-specific properties in configuration (*.ini) files, and perform database-specific or full-system data backups.

ℹ Note

System privileges should always be assigned with caution.

Related Information

System Privileges (Reference)

SAP Note 2950209

System Privileges (Reference)

System privileges control general system activities.

General System Privileges

System privileges restrict administrative tasks. The following table describes the supported system privileges in an SAP HANA database.

System Privilege	Description
ADAPTER ADMIN	Controls the execution of the following adapter-related statements: CREATE ADAPTER, DROP ADAPTER, and ALTER ADAPTER. It also allows access to the ADAPTERS and ADAPTER_LOCATIONS system views.
AGENT ADMIN	Controls the execution of the following agent-related statements: CREATE AGENT, DROP AGENT, and ALTER AGENT. It also allows access to the AGENTS and ADAPTER_LOCATIONS system views.
ALTER CLIENTSIDE ENCRYPTION KEYPAIR	Authorizes a user to add a new version of a client-side encryption key pair (CKP), or to drop all older versions of the CKP.
AUDIT ADMIN	Controls the execution of the following auditing-related statements: CREATE AUDIT POLICY, DROP AUDIT POLICY, and ALTER AUDIT POLICY, as well as changes to the auditing configuration. It also allows access to the AUDIT_LOG, XSA_AUDIT_LOG, and ALL_AUDIT_LOG system views.
AUDIT OPERATOR	Authorizes the execution of the following statement: ALTER SYSTEM CLEAR AUDIT LOG. It also allows access to the AUDIT_LOG system view.
AUDIT READ	Authorizes read-only access to the rows of the AUDIT_LOG, XSA_AUDIT_LOG, and ALL_AUDIT_LOG system views.
BACKUP ADMIN	Authorizes BACKUP and RECOVERY statements for defining and initiating backup and recovery procedures. It also authorizes changing system configuration options with respect to backup and recovery.
BACKUP OPERATOR	Authorizes the BACKUP statement to initiate a backup.
CATALOG READ	Authorizes unfiltered access to the data in the system views that a user has already been granted the SELECT privilege on. Normally, the content of these views is filtered based on the privileges of the user. CATALOG READ does not allow a user to view system views on which they have not been granted the SELECT privilege.
CERTIFICATE ADMIN	Authorizes the changing of certificates and certificate collections that are stored in the database.
CLIENT PARAMETER ADMIN	Authorizes a user to override the value of the CLIENT parameter for a database connection or to overwrite the value of the $$client$$ parameter in an SQL query.

System Privilege	Description
CREATE CLIENTSIDE ENCRYPTION KEYPAIR	Authorizes a user to create client-side encryption key pairs.
CREATE REMOTE SOURCE	Authorizes the creation of remote data sources by using the CREATE REMOTE SOURCE statement.
CREATE SCENARIO	Controls the creation of calculation scenarios and cubes (calculation database).
CREATE SCHEMA	Authorizes the creation of database schemas using the CREATE SCHEMA statement.
CREATE STRUCTURED PRIVILEGE	Authorizes the creation of structured (analytic privileges). Only the owner of the privilege can further grant or revoke that privilege to other users or roles.
CREDENTIAL ADMIN	Authorizes the use of the statements CREATE CREDENTIAL, ALTER CREDENTIAL, and DROP CREDENTIAL.
DATA ADMIN	Authorizes reading all data in the system views. It also enables execution of Data Definition Language (DDL) statements in the SAP HANA database. A user with this privilege cannot select or change data in stored tables for which they do not have access privileges, but they can drop tables or modify table definitions.
DATABASE ADMIN	Authorizes all statements related to tenant databases, such as CREATE, DROP, ALTER, RENAME, BACKUP, and RECOVERY.
DATABASE START	
DATABASE STOP	Authorizes a user to stop any database in the system and to select from the M_DATABASES view.Authorizes a user to start any database in the system and to select from the M_DATABASES view.
DROP CLIENTSIDE ENCRYPTION KEYPAIR	Authorizes a user to start any database in the system and toAuthorizes a user to drop other users' client-side encryption key pairs.
ENCRYPTION ROOT KEY ADMIN	Authorizes all statements related to management of root keys: Allows access to the system views pertaining to encryption (for example, ENCRYPTION_ROOT_KEYS, M_ENCRYPTION_OVERVIEW, M_PERSISTENCE_ENCRYPTION_STATUS, M_PERSISTENCE_ENCRYPTION_KEYS, and so on).
EXPORT	Authorizes EXPORT to a file on the SAP HANA server. The user must also have the SELECT privilege on the source tables to be exported.
EXTENDED STORAGE ADMIN	Authorizes the management of SAP HANA dynamic tiering and the creation of extended storage.
IMPORT	Authorizes the import activity in the database using the IMPORT statements. Additional privileges may also be required to be able to execute an IMPORT. See the IMPORT statement for more information.
INIFILE ADMIN	Authorizes making changes to system settings.

System Privilege	Description
LDAP ADMIN	Authorizes the use of the CREATE | ALTER | DROP | VALIDATE LDAP PROVIDER statements.
LICENSE ADMIN	Authorizes the use of the SET SYSTEM LICENSE statement to install a new license.
LOG ADMIN	Authorizes the use of the ALTER SYSTEM LOGGING [ON | OFF] statements to enable or disable the log flush mechanism.
MONITOR ADMIN	Authorizes the use of the ALTER SYSTEM statements for events.
OPTIMIZER ADMIN	Authorizes the use of the ALTER SYSTEM statements concerning SQL PLAN CACHE and ALTER SYSTEM UPDATE STATISTICS statements, which influence the behavior of the query optimizer.
PARTITION ADMIN	Authorizes the use of all non-destructive partitioning operations when altering a table.
RESOURCE ADMIN	Authorizes statements concerning system resources (for example, the ALTER SYSTEM RECLAIM DATAVOLUME and ALTER SYSTEM RESET MONITORING VIEW statements). It also authorizes use of the Kernel Profiler statements, and many of the statements available in the Management Console.
ROLE ADMIN	Authorizes the creation and deletion of roles by using the CREATE ROLE and DROP ROLE statements. It also authorizes the granting and revoking of roles by using the GRANT and REVOKE statements. Activated repository roles, meaning roles whose creator is the predefined user _SYS_REPO, can neither be granted to other roles or users nor dropped directly. Not even users with the ROLE ADMIN privilege can do so. Check the documentation concerning activated objects.
SAVEPOINT ADMIN	Authorizes the execution of a savepoint using the ALTER SYSTEM SAVEPOINT statement.
SCENARIO ADMIN	Authorizes all calculation scenario-related activities (including creation).
SERVICE ADMIN	Authorizes the ALTER SYSTEM [START|CANCEL|RECONFIGURE] statements for administering system services of the database.
SESSION ADMIN	Authorizes the ALTER SYSTEM commands concerning sessions to stop or disconnect a user session or to change session variables.
SSL ADMIN	Authorizes the use of the SET...PURPOSE SSL statement. It also allows access to the PSES system view.
STRUCTUREDPRIVILEGE ADMIN	Authorizes the creation, reactivation, and dropping of structured (analytic) privileges.
SYSTEM REPLICATION ADMIN	Authorizes the use of ALTER SYSTEM statements related to system replication.
TABLE ADMIN	Authorizes the LOAD, UNLOAD and MERGE DELTA statements for tables and table partitions, as well as the ALTER TABLE statement for those clauses that do change the structure of the table and do not allow access to table data either explicitly or implicitly, for

System Privilege	Description
	example: LOB REORGANIZE, CLEAR COLUMN JOIN DATA STATISTICS, and PRELOAD.
TRACE ADMIN	Authorizes the use of the ALTER SYSTEM statements related to database tracing (including the Kernel Profiler feature) and the changing of trace system settings.
TRUST ADMIN	Authorizes the use of statements to update the trust store.
USER ADMIN	Authorizes the creation and modification of users by using the CREATE | ALTER | DROP USER statements.
VERSION ADMIN	Authorizes the use of the ALTER SYSTEM RECLAIM VERSION SPACE statement of the multi-version concurrency control (MVCC) feature.
WORKLOAD ADMIN	Authorizes execution of the workload class and mapping statements (for example, CREATE | ALTER | DROP WORKLOAD CLASS, and CREATE | ALTER | DROP WORKLOAD MAPPING).
WORKLOAD ANALYZE ADMIN	Used by the Analyze Workload, Capture Workload, and Replay Workload applications when performing workload analysis.
WORKLOAD CAPTURE ADMIN	Authorizes access to the monitoring view M_WORKLOAD_CAPTURES to see the current status of capturing and captured workloads, as well of execution of actions with the WORKLOAD_CAPTURE procedure.
WORKLOAD REPLAY ADMIN	Authorizes access to the monitoring views M_WORKLOAD_REPLAY_PREPROCESSES and M_WORKLOAD_REPLAYS to see current status of preprocessing, preprocessed, replaying, and replayed workloads, as well as the execution of actions with the WORKLOAD_REPLAY procedure.
<identifier>.<identifier>	Components of the SAP HANA database can create new system privileges. These privileges use the component-name as the first identifier of the system privilege and the component-privilege-name as the second identifier.

Repository System Privileges

ℹ Note

The following privileges authorize actions on individual packages in the SAP HANA repository, used in the SAP HANA Extended Services (SAP HANA XS) classic development model. With SAP HANA XS advanced, source code and web content are no longer versioned and stored in the repository of the SAP HANA database.

System Privilege	Description
REPO.EXPORT	Authorizes the export of delivery units for example
REPO.IMPORT	Authorizes the import of transport archives
REPO.MAINTAIN_DELIVERY_UNITS	Authorizes the maintenance of delivery units (DU, DU vendor and system vendor must be the same

System Privilege	Description
REPO.WORK_IN_FOREIGN_WORKSPACE	Authorizes work in a foreign inactive workspace
REPO.CONFIGURE	Authorize work with SAP HANA Change Recording, which is part of SAP HANA Application Lifecycle Management
REPO.MODIFY_CHANGE	
REPO.MODIFY_OWN_CONTRIBUTION	
REPO.MODIFY_FOREIGN_CONTRIBUTION	

Related Information

GRANT
Developer Authorization in the Repository

Object Privileges

Object privileges are SQL privileges that are used to allow access to and modification of database objects.

For each SQL statement type (for example, SELECT, UPDATE, or CALL), a corresponding object privilege exists. If a user wants to execute a particular statement on a simple database object (for example, a table), he or she must have the corresponding object privilege for either the actual object itself, or the schema in which the object is located. This is because the schema is an object type that contains other objects. A user who has object privileges for a schema automatically has the same privileges for all objects currently in the schema and any objects created there in the future.

Object privileges are not only grantable for database catalog objects such as tables, views and procedures. Object privileges can also be granted for non-catalog objects such as development objects in the repository of the SAP HANA database.

Initially, the owner of an object and the owner of the schema in which the object is located are the only users who can access the object and grant object privileges on it to other users.

An object can therefore be accessed only by the following users:

- The owner of the object
- The owner of the schema in which the object is located
- Users to whom the owner of the object has granted privileges
- Users to whom the owner of the parent schema has granted privileges

⚠ Caution

The database owner concept stipulates that when a database user is deleted, all objects created by that user and privileges granted to others by that user are also deleted. If the owner of a schema is deleted, all objects in the schema are also deleted even if they are owned by a different user. All privileges on these objects are also deleted.

ℹ Note

The owner of a table can change its ownership with the ALTER TABLE SQL statement. In this case, the new owner becomes the grantor of all privileges on the table granted by the original owner. The original owner is also automatically granted all privileges for the table with the new owner as grantor. This ensures that the original owner can continue to work with the table as before.

Authorization Check on Objects with Dependencies

The authorization check for objects defined on other objects (that is, stored procedures and views) is more complex. In order to be able to access an object with dependencies, both of the following conditions must be met:

- The user trying to access the object must have the relevant object privilege on the object as described above.
- The user who created the object must have the required privilege on all underlying objects **and** be authorized to grant this privilege to others.

If this second condition is not met, only the owner of the object can access it. He cannot grant privileges on it to any other user. This cannot be circumvented by granting privileges on the parent schema instead. Even if a user has privileges on the schema, he will still not be able to access the object.

ℹ Note

This applies to procedures created in DEFINER mode only. This means that the authorization check is run against the privileges of the user who created the object, not the user accessing the object. For procedures created in INVOKER mode, the authorization check is run against the privileges of the accessing user. In this case, the user must have privileges not only on the object itself but on all objects that it uses.

→ Tip

The SAP HANA studio provides a graphical feature, the authorization dependency viewer, to help troubleshoot authorization errors for object types that typically have complex dependency structures: stored procedures and calculation views.

Related Information

GRANT Statement (Access Control)
ALTER TABLE Statement (Data Definition)
Resolve Errors Using the Authorization Dependency Viewer
Object Privileges (Reference)
Cross-Database Authorization in Tenant Databases

Object Privileges (Reference)

Object privileges are used to allow access to and modification of database objects, such as tables and views.

The following table describes the supported object privileges in an SAP HANA database.

Object Privilege	Command Types	Applies to	Privilege Description
ALL PRIVILEGES	DDL & DML	• Schemas • Tables • Views	This privilege is a collection of all Data Definition Language (DDL) and Data Manipulation Language (DML) privileges that the grantor currently possesses and is allowed to grant further. The privilege it grants is specific to the particular object being acted upon. This privilege collection is dynamically evaluated for the given grantor and object.
ALTER	DDL	• Schemas • Tables • Views • Functions/procedures	Authorizes the ALTER statement for the object.
AGENT MESSAGING	DDL	• Schemas • Tables • Views • Functions/procedures	Authorizes the user with which the agent communicates with the data provisioning server using HTTP protocol.
ATTACH DEBUGGER	DDL	• User	Authorizes debugging across different user sessions. For example, userA can grant ATTACH DEBUGGER to userB to allow userB to debug a procedure in userA's session (userB still needs DEBUG privilege on the procedure, however).
CREATE ANY	DDL	• Schemas • Tables • Views • Sequences • Functions/procedures • Remote sources • Graph workspaces • Triggers	Authorizes all CREATE statements for the object.
CREATE OBJECT STRUCTURED PRIVILEGE	DDL	• Schemas • Views	Authorizes creation of structured privileges on the object even if the user does not have the CREATE STRUCTURED PRIVILEGE.

Object Privilege	Command Types	Applies to	Privilege Description
CREATE REMOTE SUBSCRIPTION	DDL	• Remote Source	Authorizes the creation of remote subscriptions executed on this source entry.
CREATE VIRTUAL FUNCTION	DDL	• Remote sources	Authorizes creation of virtual functions (the REFERENCES privilege is also required).
CREATE VIRTUAL PROCEDURE	DDL	• Remote sources	Authorizes creation of virtual procedure to create and run procedures on a remote source.
CREATE VIRTUAL PACKAGE	DDL	• Schemas	Authorizes creation of virtual packages that can be run on remote sources.
CREATE VIRTUAL TABLE	DDL	• Remote sources	Authorizes the creation of proxy tables pointing to remote tables from the source entry.
CREATE TEMPORARY TABLE	DDL	• Schemas	Authorizes the creation of a temporary local table, which can be used as input for procedures, even if the user does not have the CREATE ANY privilege for the schema.
DEBUG	DML	• Schemas • Calculation Views • Functions/procedures	Authorizes debug functionality for the procedure or calculation view or for the procedures and calculation views of a schema.
DEBUG MODIFY	DDL	• Functions/procedures	For internal use only.
DELETE	DML	• Schemas • Tables • Views • Functions/procedures	Authorizes the DELETE and TRUNCATE statements for the object. While DELETE applies to views, it only applies to updatable views (that is, views that do not use a join, do not contain a UNION, and do not use aggregation).
DROP	DDL	• Schemas • Tables • Views • Sequences • Functions/procedures • Remote sources • Graph workspaces	Authorizes the DROP statements for the object.

Object Privilege	Command Types	Applies to	Privilege Description
EXECUTE	DML	• Schemas • Functions/procedures	Authorizes the execution of a SQLScript function or a database procedure by using the CALLS or CALL statement respectively. It also allows a user to execute a virtual function.
INDEX	DDL	• Schemas • Tables	Authorizes the creation, modification, or dropping of indexes for the object.
INSERT	DML	• Schemas • Tables • Views	Authorizes the INSERT statement for the object. The INSERT and UPDATE privilege are both required on the object to allow the REPLACE and UPSERT statements to be used. While INSERT applies to views, it only applies to updatable views (views that do not use a join, do not contain a UNION, and do not use aggregation).
PROCESS REMOTE SUBSCRIPTION EXCEPTION	DDL	• Remote Source • Remote Subscription	Authorizes processing exceptions on this source entry.
LINKED DATABASE	DDL	• Remote source	Authorizes execution of linked database queries.
REFERENCES	DDL	• Schemas • Tables	Authorizes the usage of all tables in this schema or this table in a foreign key definition, or the usage of a personal security environment (PSE). It also allows a user to reference a virtual function package.
REMOTE TABLE ADMIN	DDL	• Remote sources	Authorizes the creation of tables on a remote source object.
SELECT	DML	• Schemas • Tables • Views • Sequences • Graph workspaces	Authorizes the SELECT statement for the object or the usage of a sequence. When selection from system-versioned tables, users must have SELECT on both the table and its associated history table.
SELECT CDS METADATA	DML	• Schemas • Tables	Authorizes access to CDS metadata from the catalog.
SELECT METADATA	DML	• Schemas	Authorizes access to the complete metadata of all

Object Privilege	Command Types	Applies to	Privilege Description
		• Tables	objects in a schema (including procedure and view definitions), including objects that may be located in other schemas.
SQLSCRIPT LOGGING	DML	• Function • Library • Procedure • Schema	Authorizes the collection of logs for a SQLScript object.
TRIGGER	DDL	• Schemas • Tables	Authorizes the CREATE/ALTER/DROP/ENABLE and DISABLE TRIGGER statements for the specified table or the tables in the specified schema.
UNMASKED	DML	• Schemas • Views • Tables	Authorizes access to masked data in user-defined views and tables. This privilege is required to view the original data in views and tables that are defined by using the WITH MASK clause.
UPDATE	DML	• Schemas • Tables • Views	While UPDATE applies to views, it only applies to updatable views (views that do not use a join, do not contain a UNION, and do not use aggregation).
USAGE	DDL	• Client Side Encryption Coulmn Key	Authorizes client side encryption keys.
USERGROUP OPERATOR	DML	• User groups	Authorizes a user to change the settings for a user group, and to add and remove users to/from a user group. Users with the USERGROUP OPERATOR privilege can also create and drop users, but only within the user group they have the USERGROUP OPERATOR privilege on (CREATE USER *<user_name>* SET USERGROUP *<usergroup_name>*). A user can have the USERGROUP OPERATOR privilege on more than one user group, and a user group can have more than one user with the USERGROUP OPERATOR privilege on it.

Object Privilege	Command Types	Applies to	Privilege Description
			When granting USERGROUP OPERATOR to a user group, you must include the keyword USERGROUP before the name of the user group (for example: `GRANT USERGROUP OPERATOR ON USERGROUP <usergroup> TO <grantee>`). This is slightly differenct syntax than granting USERGROUP OPERATOR to a user.
<identifier>.<identifier>	DDL		Components of the SAP HANA database can create new object privileges. These privileges use the component-name as first identifier of the system privilege and the component-privilege-name as the second identifier.

Related Information

GRANT

Analytic Privileges

Analytic privileges grant different users access to different portions of data in the same view based on their business role. Within the definition of an analytic privilege, the conditions that control which data users see is either contained in an XML document or defined using SQL.

Row-Level Access Control

Standard object privileges (`SELECT`, `ALTER`, `DROP`, and so on) implement coarse-grained authorization at object level only. Users either have access to an object, such as a table, view or procedure, or they don't. While this is often sufficient, there are cases when access to data in an object depends on certain values or combinations of values. Analytic privileges are used in the SAP HANA database to provide such fine-grained control at row level of which data individual users can see within the same view.

Example

Sales data for all regions are contained within one analytic view. However, regional sales managers should only see the data for their region. In this case, an analytic privilege could be modeled so that they can all query the view, but only the data that each user is authorized to see is returned.

Creating Analytic Privileges

Although analytic privileges can be created directly as catalog objects in runtime, we recommend creating them as design-time objects that become catalog objects on deployment (database artifact with file suffix `.hdbanalyticprivilege`).

In an SAP HANA XS classic environment, analytic privileges are created in the built-in repository of the SAP HANA database using either the SAP HANA Web Workbench or the SAP HANA studio. In an SAP HANA XS advanced environment, they are created using

the SAP Web IDE and deployed using the SAP HANA deployment infrastructure (SAP HANA DI).

ℹ Note

HDI supports only analytic privileges deployed using SQL (see below). Furthermore, due to the container-based model of HDI, where each container corresponds to a database schema, analytic privileges created in HDI are schema specific.

XML- Versus SQL-Based Analytic Privileges

Before you implement row-level authorization using analytic privileges, you need to decide which type of analytic privilege is suitable for your scenario. In general, SQL-based analytic privileges allow you to more easily formulate complex filter conditions using sub-queries that might be cumbersome to model using XML-based analytic privileges.

→ Recommendation

SAP recommends the use of SQL-based analytic privileges. Using the **SAP HANA Modeler** perspective of the SAP HANA studio, you can migrate XML-based analytic privileges to SQL-based analytic privileges. For more information, see the *SAP HANA Modeling Guide for SAP HANA Studio*.

ℹ Note

As objects created in the repository, XML-based analytic privileges are deprecated as of SAP HANA 2.0 SPS 02. For more information, see SAP Note 2465027.

The following are the main differences between XML-based and SQL-based analytic privileges:

Feature	SQL-Based Analytic Privileges	XML-Based Analytic Privileges
Control of read-only access to SAP HANA information models: • Attribute views • Analytic views • Calculation views	Yes	Yes
Control of read-only access to SQL views	Yes	No
Control of read-only access to database tables	No	No
Design-time modeling using the SAP HANA Web-based Workbench or the **SAP HANA Modeler** perspective of the SAP HANA studio **ℹ Note** This corresponds to development in an SAP HANA XS classic environment using the SAP HANA repository.	Yes	Yes
Design-time modeling using the SAP Web IDE for SAP HANA **ℹ Note** This corresponds to development in an SAP HANA XS advanced environment using HDI.	Yes	No
Transportable	Yes	Yes
HDI support	Yes	No

Feature	SQL-Based Analytic Privileges	XML-Based Analytic Privileges
Complex filtering	Yes	No

Enabling an Authorization Check Based on Analytic Privileges

All column views modeled and activated in the SAP HANA modeler and the SAP HANA Web-based Development Workbench automatically enforce an authorization check based on analytic privileges. XML-based analytic privileges are selected by default, but you can switch to SQL-based analytic privileges.

Column views created using SQL must be explicitly registered for such a check by passing the relevant parameter:

- `REGISTERVIEWFORAPCHECK` for a check based on XML-based analytic privileges
- `STRUCTURED PRIVILEGE CHECK` for a check based on SQL-based analytic privileges

SQL views must always be explicitly registered for an authorization check based on analytic privileges by passing the `STRUCTURED PRIVILEGE CHECK` parameter.

ℹ Note

It is not possible to enforce an authorization check on the same view using both XML-based and SQL-based analytic privileges. However, it is possible to build views with different authorization checks on each other.

Related Information

Create Static SQL Analytic Privileges (SAP Web IDE for SAP HANA)
Create Dynamic SQL Analytic Privileges (SAP Web IDE for SAP HANA)
Create Analytic Privileges Using SQL Expressions (SAP Web IDE for SAP HANA)
Create Classical XML-Based Analytic Privileges (SAP HANA Web Workbench)
Create Static SQL Analytic Privileges (SAP HANA Web Workbench)
Create Classical XML-based Analytic Privileges (SAP HANA Studio)
Create SQL Analytic Privileges (SAP HANA Studio)
Convert Classical XML-based Analytic Privileges to SQL-based Analytic Privileges (SAP HANA Studio)
SAP Note 2465027

Package Privileges

Package privileges authorize actions on individual packages in the SAP HANA repository.

Privileges granted on a repository package are implicitly assigned to the design-time objects in the package, as well as to all sub-packages. Users are only allowed to maintain objects in a repository package if they have the necessary privileges for the package in which they want to perform an operation, for example to read or write to an object in that package. To be able perform operations in all packages, a user must have privileges on the root package `.REPO_PACKAGE_ROOT`.

If the user authorization check establishes that a user does not have the necessary privileges to perform the requested operation in a specific package, the authorization check is repeated on the parent package and recursively up the package hierarchy to the root level of the repository. If the user does not have the necessary privileges for any of the packages in the hierarchy chain, the authorization check fails and the user is not permitted to perform the requested operation.

In the context of repository package authorizations, there is a distinction between native packages and imported packages.

- Native package

  A package that is created in the current system and expected to be edited in the current system. Changes to packages or to objects the packages contain must be performed in the original development system where they were created and transported into subsequent systems. The content of native packages are regularly edited by developers.

- Imported package

  A package that is created in a remote system and imported into the current system. Imported packages should not usually be modified, except when replaced by new imports during an update. Otherwise, imported packages or their contents should only be modified in exceptional cases, for example, to carry out emergency repairs.

ℹ Note

The SAP HANA administrator can grant the following package privileges to an SAP HANA user: edit, activate, and maintain.

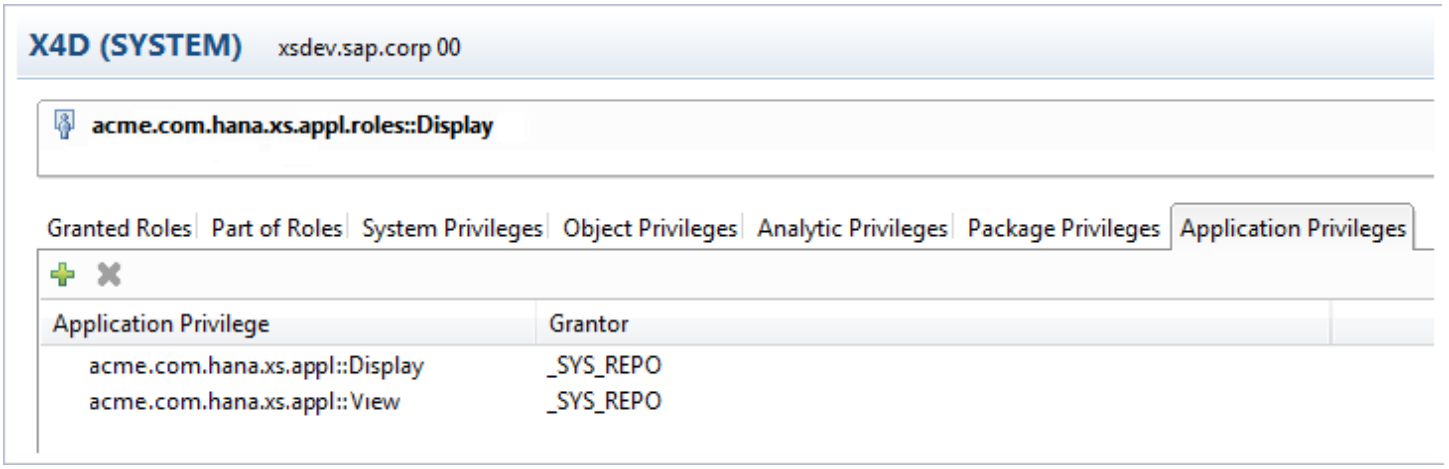
Related Information

Package Privilege Options

Application Privileges

In SAP HANA Extended Application Services (SAP HANA XS), application privileges define the authorization level required for access to an SAP HANA XS application, for example, to start the application or view particular functions and screens.

Application privileges can be assigned to an individual user or to a group of users, for example, in a user **role**. The user role can also be used to assign system, object, package, and analytic privileges, as illustrated in the following graphic. You can use application privileges to provide different levels of access to the same application, for example, to provide advanced maintenance functions for administrators and view-only capabilities to normal users.



Application Privileges for Users and User Roles

If you want to define application-specific privileges, you need to understand and maintain the relevant sections in the following design-time artifacts:

- Application-privileges file (`.xsprivileges`)
- Application-access file (`.xsaccess`)
- Role-definition file (*`<RoleName>`*`.hdbrole`)

Application privileges can be assigned to users individually or by means of a user **role**, for example, with the "*application privilege*" keyword in a role-definition file (*`<RoleName>`*`.hdbrole`) as illustrated in the following code. You store the roles as design-time artifacts within the application package structure they are intended for, for example, `acme.com.hana.xs.app1.roles`.

```
role acme.com.hana.xs.app1.roles::Display
{
    application privilege: acme.com.hana.xs.appl::Display;
    application privilege: acme.com.hana.xs.appl::View;

    catalog schema "ACME_XS_APP1": SELECT;

    package acme.com.hana.xs.app1: REPO.READ;
    package ".REPO_PACKAGE_ROOT" : REPO.READ;

    catalog sql object "_SYS_REPO"."PRODUCTS": SELECT;
    catalog sql object "_SYS_REPO"."PRODUCT_INSTANCES": SELECT;
    catalog sql object "_SYS_REPO"."DELIVERY_UNITS": SELECT;
    catalog sql object "_SYS_REPO"."PACKAGE_CATALOG": SELECT;
    catalog sql object "ACME_XS_APPL"."acme.com.hana.xs.appl.db::SYSTEM_STATE": SELECT, INSERT, UPD
}
```

The application privileges referenced in the role definition (for example, `Display` and `View`) are actually defined in an application-specific `.xsprivileges` file, as illustrated in the following example, which also contains entries for additional privileges that are not explained here.

ℹ Note

The `.xsprivileges` file must reside in the package of the application to which the privileges apply.

The package where the `.xsprivileges` resides defines the scope of the application privileges; the privileges specified in the `.xsprivileges` file can only be used in the package where the `.xsprivileges` resides (or any sub-packages). This is checked during activation of the `.xsaccess` file and at runtime in the by the XS JavaScript API *$.session.(has|assert)AppPrivilege()*.

```
{
  "privileges" : [
   { "name" : "View", "description" : "View Product Details" },
   { "name" : "Configure", "description" : "Configure Product Details" },
   { "name" : "Display", "description" : "View Transport Details" },
   { "name" : "Administrator", "description" : "Configure/Run Everything" },
   { "name" : "ExecuteTransport", "description" : "Run Transports"},
   { "name" : "Transport", "description" : "Transports"}
  ]
}
```

The privileges are **authorized** for use with an application by inserting the *authorization* keyword into the corresponding `.xsaccess` file, as illustrated in the following example. Like the `.xsprivileges` file, the `.xsaccess` file must reside either in the root package of the application to which the privilege authorizations apply or the specific subpackage which requires the specified authorizations.

ℹ Note

If a privilege is inserted into the `.xsaccess` file as an authorization requirement, a user must have this privilege to access the application package where the `.xsaccess` file resides. If there is more than one privilege, the user must have at least one of these privileges to access the content of the package.

```
{
    "prevent_xsrf": true,
    "exposed": true,
    "authentication": {
        "method": "Form"
    },
    "authorization": [
       "acme.com.hana.xs.appl::Display",
       "acme.com.hana.xs.appl::Transport"
```

```
        ]
}
```

Related Information

[Custom Role for Developers]
[Creating the Application Descriptors]