This sample chapter provides an overview of the role creation and deletion process and then reviews several options for managing privilege assignments to roles. It also covers how roles can be assigned to other roles.

“Database Roles”

Contents
Index
The Author

Jonathan Haun
SAP HANA 2.0 Security Guide
608 Pages, 2020, $89.95
www.sap-press.com/4982
Chapter 5  
Database Roles

In this chapter, we’ll present a complete overview of standard roles, including the processes used to provision roles, maintain roles, and grant privileges to roles.

Roles are objects that aid in the creation of a security model. The bulk of the security model development process involves defining roles, assigning privileges, and assigning roles to user accounts. In this chapter, we’ll define standard database roles, which are created using the `CREATE ROLE` SQL syntax. Based on best practices, standard database roles shouldn’t be used in SAP HANA; we’ll discuss why shortly. However, in some instances, you’ll need to use standard database roles. In this chapter, we’ll start with an overview of the role creation and deletion process, then review several options for managing privilege assignments to roles. Finally, we’ll explore how roles can be assigned to other roles.

5.1 What Are Roles?

A role is essentially an object in the SAP HANA system that contains a collection of granted privileges. A role can be granted to a user, causing the user to inherit all the privileges assigned to the role, and privileges can be granted to a role directly. In these instances, you’ll use `GRANT` statements to assign individual privileges to the role. However, existing roles can also be granted to a role. In these cases, the role will contain its privileges plus any privileges in the attached role. We often refer to these kinds of roles as nested roles because the parent role’s privileges are extended to the child role.

Roles aid in the creation of a security model because they allow administrators to create reusable collections of privileges. Without roles, administrators would need to grant individual permissions to each individual user account. If the SAP HANA environment had hundreds of users, each with a diverse level of access, administrators
would find it difficult to manage SAP HANA security. With roles, the security administrator should only be concerned with granting roles, not permissions, to users. Because roles are reusable, a single role can be granted to hundreds of users. This methodology helps the administrator in several ways: First, any given user account would only need a few roles granted. Second, updates to a role are automatically cascaded to each user assigned the role. Third, roles support the creation of a consistently implemented security model. Therefore, roles are fundamentally important to the creation of a security model; in fact, you might argue that roles are the key objects we produce when creating a security model.

The SAP HANA system is delivered with several base roles that are created by the system during installation. Most experts agree that you should avoid assigning these roles to users because they often grant a high level of access to the SAP HANA system. Administrators can review the privileges granted to these roles to determine privileges assigned to specific activities within SAP HANA. As a rule of thumb, new custom roles should be created based on the requirements of each organization.

Table 5.1 shows an overview of a few example standard database roles delivered with an SAP HANA system.

<table>
<thead>
<tr>
<th>Role Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTENT_ADMIN</td>
<td>In general, this role includes all the privileges assigned to the MODELING role, plus a few extra. However, this role allows these privileges to be granted to other users. This role also allows a user to import and export content within the SAP HANA package repository and provides full access to both imported and native development artifacts. This role has high privileges and should not be granted to an actual user. This role is best used as a template to help an administrator understand the various rights and privileges required by an SAP HANA modeler and security administrator.</td>
</tr>
<tr>
<td>PUBLIC</td>
<td>This role is the default role assigned to all standard users within SAP HANA and includes a variety of SELECT and EXECUTE object privileges on tables, views, and procedures found in the SYS schema. Many of the stored procedures in the SYS schema will be used to create database objects or list SAP HANA content. Without access to these procedures, tables, or views, a user would have extremely limited rights within SAP HANA and SAP HANA administration tools. Additional privileges can be both granted and revoked from the PUBLIC role. This flexibility allows administrators to establish a baseline set of privileges for all standard users.</td>
</tr>
<tr>
<td>MONITORING</td>
<td>This role contains all the privileges necessary to query system metadata and system state tables. As its name implies, this role grants the privileges needed to monitor the system. This role grants the DBA_COCKPIT role, grants the catalog read system privilege, and enables the use of SELECT on the <em>SYS</em> STATISTICS schema.</td>
</tr>
<tr>
<td>RESTRICTED_USER_ JDBC_ACCESS</td>
<td>This role provides the absolute minimum privileges necessary to connect to SAP HANA using Java Database Connectivity (JDBC). This role is often granted to restricted users so they can connect to SAP HANA via JDBC.</td>
</tr>
<tr>
<td>RESTRICTED_USER_ ODBC_ACCESS</td>
<td>This role provides the absolute minimum privileges necessary to connect to SAP HANA using Open Database Connectivity (ODBC). This role is often granted to restricted users so they can connect to SAP HANA via ODBC.</td>
</tr>
</tbody>
</table>

Table 5.1 Default Standard Database Roles Delivered with SAP HANA (Cont.)
Two main issues with the usage of standard database roles often arise when developing a security model: The first issue is an organization’s ability to maintain a consistent security model between environments. This issue is universal for all current versions of SAP HANA. The second issue is an organization’s ability to manage the user accounts that grant privileges to other users and roles. This issue is only a problem prior to SAP HANA 2.0, but let’s look at these differences in more detail:

### Consistency Issues

Most organizations will maintain multiple related SAP HANA systems to facilitate lifecycle management practices. One characteristic of a lifecycle management practice is that it’s a process in which content is developed in a particular environment and then promoted first to a test environment, followed by a production environment. The process isn’t limited only to SAP HANA application development artifacts; you can also include the development lifecycle of a security model in this process. In particular, a security model can be created in a development environment and then promoted to test and production environments for the purposes of validation. This practice is important because each environment requires security, and you’ll need to test your security changes outside of a production environment.

When you use standard database roles, the design-time scripts used to create roles, grant privileges, and revoke privileges must be maintained outside of the SAP HANA system in the form of text files containing SQL scripts. To promote these roles and their privileges, you must execute identical SQL scripts in each environment. In addition, you must also manually preserve each version of the role within SQL scripts in instances in which you’ll need to revert a change. Because the design-time versions of roles are decoupled from the SAP HANA system, management of this process is manual and prone to inconsistencies between environments. Fortunately, SAP HANA offers an alternative to standard database roles, called repository-based roles, which aren’t subject to this issue. We’ll discuss repository roles in more detail in the next chapter.

### Grantor Issues

In SAP HANA 1.0 and before SAP HANA 2.0 SPS 03, standard database roles were also subject to grantor issues. When a user account grants privileges to another user or role, the first user becomes the grantor, and only the grantor can revoke a role granted to the second user. Starting with SAP HANA 2.0 SPS 03, any user with the ROLE ADMIN system privilege can grant and revoke roles regardless of the original grantor. Another grantee issue common with all SAP HANA versions is the grantor deletion issue. As discussed in Chapter 4, technical user accounts should be created for each security administrator to avoid the cascade deletion issue common to deleting user accounts. Deleting user accounts that have granted privileges will revoke any privileges that they have granted to a standard role. Therefore, the use of standard roles in a security model will require an extra level of discipline because you must make sure that all security changes are implemented with technical user accounts and that these accounts are never deleted. This requirement can make the security model management process more difficult than usual. Again, fortunately, repository-based roles aren’t subject to this issue (see Chapter 6 for more information).

### 5.2 Creating and Managing Roles

At a high level, three different ways exist for creating and managing roles within SAP HANA: via SQL statements based on the CREATE ROLE syntax, in the SAP HANA cockpit, or in the SAP HANA Web-based Development Workbench security manager. In this section, we’ll discuss the three different options used to provision and manage standard roles.

#### 5.2.1 Creating and Deleting Roles with SQL Statements

You can create and drop roles with SQL statements. Creating global roles using SQL statements involves issuing a CREATE ROLE statement. Users must have the ROLE ADMIN system privilege to create and delete standard roles. To create a standard role, use the following syntax:

```
CREATE ROLE ROLENAME
```

Replace `ROLENAME` with the desired role name. Role names can contain most characters, including spaces and special characters. For example, you can create a role called “My Role Has a Long Name?” with the following SQL statement:
CREATE ROLE "My Role Has a Long Name?";

As of SAP HANA SPS 11, roles can also be created with a schema prefix. The name of a given role must be unique within the global system namespace or schema-specific namespace, but the same role name can exist in the global and schema namespaces simultaneously. Technically, the schema prefix differentiates the roles within the system, even when their display names appear to overlap. Schema-prefixed roles will also be deleted from the system if the associated schema is deleted from the system.

To create a role with a schema name prefix, issue the following two SQL statements:

```
CREATE ROLE <MY_SCHEMA_NAME>.<ROLENAME>;
CREATE ROLE "MySchema"."DATABASE ADMIN USER 2200";
```

Replace `<MY_SCHEMA_NAME>` with the name of an existing schema.

You can also use the DROP ROLE SQL syntax to delete a role, as follows:

```
DROP ROLE <ROLENAME>;
DROP ROLE <MY_SCHEMA_NAME>.<ROLENAME>;
```

Replace `<ROLENAME>` with the name of an existing standard role and replace `<MY_SCHEMA_NAME>` with the name of an existing schema.

When a role is dropped from the system, this role will be revoked from any user or role that was previously granted the deleted role, as will all the privileges associated with the role. In general, you should avoid deleting roles from the system when they're currently associated with users or other roles.

Starting with SAP HANA SPS 12, a new system view was created in the SYS schema to help identify the user accounts associated with a specified role. The name of the view is EFFECTIVE_ROLE_GRANTEES. The following SQL statement queries this view:

```
SELECT * FROM EFFECTIVE_ROLE_GRANTEES WHERE ROLE_NAME = <ROLE_NAME>;
```

Replace `<ROLE_NAME>` with the name of a role defined within the system. The view must be filtered based on the ROLE_NAME column. Therefore, the WHERE clause in this SQL statement is mandatory.

Starting with SAP HANA 2.0 SPS O3, a role can also be created without automatically granting the role to the creator. Security administrators must add the additional syntax `NO GRANT TO CREATOR` when issuing SQL to create the role. The following SQL statement provides an example of this syntax:

```
CREATE ROLE <MY_SCHEMA_NAME>.<ROLENAME> NO GRANT TO CREATOR;
```
Privileges, Is part of Roles, Application Privileges, Package Privileges, and Privileges on users. Click each tab to reveal new options to grant or revoke the privileges applicable to the tab name.

To delete a role with the SAP HANA cockpit, click on the role on the left side of Manage Role window. The role details window for the selected role will open. In this window, click the Delete button on the bottom left, and a new Confirmation popup window will appear. Confirm that you want to delete the role by clicking the OK button.

When a role is deleted from the system, the role will be revoked from any user or role that was previously granted the deleted role, as will all the privileges associated with the role. In general, you should avoid deleting roles from the system when they’re currently associated with a user. Query the system view EFFECTIVE_ROLE_GRANTEES to identify any users or roles currently assigned a role.

5.2.3 Creating and Deleting Roles with the SAP HANA Web-Based Development Workbench

The SAP HANA Web-Based Development Workbench security manager, hosted within the SAP HANA extended application services, classic model (SAP HANA XS) engine (XS engine), provides a graphical user interface (GUI) to create and delete roles. You can access the SAP HANA Web-Based Development Workbench via a supported Internet browser. Use the following URLs customized to match the details of your environment:

- http://<sap_hana_host>:80<instance_number>/sap/hana/ide/security

Replace <sap_hana_host> with the host name of the SAP HANA system in your environment. Replace <instance_number> with the two-digit instance number corresponding to your SAP HANA system. For secure access, customize the following URLs:

- https://<sap_hana_host>:43<instance_number>/sap/hana/ide/security

To use the SAP HANA Web-Based Development Workbench security manager, the user account first will need to be granted one of the roles listed in Table 5.2. Users only need one of the two roles listed to access the security manager.

### Table 5.2 Roles Required for the SAP HANA Web-Based Development Workbench Security Manager

<table>
<thead>
<tr>
<th>Role Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sap.hana.ide.roles::SecurityAdmin</td>
<td>Within the SAP HANA Web-Based Development Workbench, a user will only be able to access the security manager, which provides interfaces to create and manage users and roles.</td>
</tr>
<tr>
<td>sap.hana.ide.roles::Developer</td>
<td>This role grants a higher level of privileges for users that need full access to SAP HANA Web-Based Development Workbench. This role grants access to the editor, catalog, and security interfaces and is considered the parent to the sap.hana.ide.roles::SecurityAdmin role.</td>
</tr>
</tbody>
</table>

### Additional Privileges to Use Security Manager

Note that the roles listed in Table 5.2 simply allow the granted user access to the SAP HANA Web-Based Development Workbench security manager. To manage a security model, additional privileges and roles will need to be granted. For example, the user will need the ROLE_ADMIN system privilege to create and drop roles. To grant privileges to a user or role, the account accessing the security manager will also require privileges to grant the privileges to other users or roles.

To create a role within the SAP HANA Web-Based Development Workbench security manager interface, expand the Security ▶ Roles nodes located on the left. Right-click the Roles node and choose New Role. A New Role tab will open on the right, as shown in Figure 5.2.

To specify the role name, enter the desired name in the mandatory Role Name field. At the bottom of the New Role tab, you’ll see seven tabs: Granted Roles, Part of Roles, System Privileges, Object Privileges, Analytic Privileges, Package Privileges, and Application Privileges. Click each tab to reveal new options to grant or revoke privileges applicable to the tab name.

To activate a new role, click the disk icon located in the icon bar. Clicking the icon will save and activate the changes. Note that the activation status window is located at the bottom of the interface window and has a black background. If any errors are encountered during activation, red text will appear indicating the issue.
5.3 Granting and Revoking Privileges

As discussed in Chapter 4, we don’t recommend assigning individual privileges to individual user accounts as the basis for a security model. Roles should be created, and privileges should then be assigned to roles. Once a role is configured with the applicable privileges, the role should then be granted to one or more user accounts.

In the following sections, we’ll look at the different methodologies for granting privileges to roles and then show you the steps for granting privileges to roles with SQL, the SAP HANA cockpit, and the SAP HANA Web-Based Development Workbench.

5.3.1 Methodologies for Granting Privileges to Roles

Using a role-based security model methodology is generally more scalable in environments with a large number of user accounts and a diverse mixture of access requirements. Roles should be created based on common access requirements and common user functions. For example, you might create a role named PUBLIC_DATA_ACCESS. Such a role can grant basic select access to database tables, column views, and standard views that are deemed accessible by all defined user accounts.

You can create another role named SECURE_DATA_ACCESS that grants limited access to tables, column views, and views that contain confidential data. Such a role might also incorporate a structured privilege that filters the results based on a defined level of data access.

Generally, we recommend that two broad categories of roles be created. The first category involves roles that accommodate both basic catalog object and schema-level access. These roles focus on providing select and execute access to specific catalog objects or perhaps an entire schema. For example, you might create a role named DATA_MART_READ_ONLY. This role will provide select access to all objects in the schema named DATA_MART. Often, each schema will have its own dedicated set of roles, with each role providing a different level of access to the schema. For example, you can define the following roles in which each role grants a higher level of privileges to the schema name used in the prefix:

- DATA_MART_READ_ONLY
- DATA_MART_POWER_USER
- DATA_MART_SEC_ADMIN
- DATA_MART_OWNER

The second category involves roles based on functional activities. For example, developers could be granted privileges to work in the SAP HANA package repository. A role named ECORP_INFORMATION_MODELER might be created for users that will create information views. Security administrators would be granted privileges to manage security. For example, the ECORP_SECURITY_ADMIN role would be created for individuals that plan to manage security. Finally, the BACKUP_MONITOR and BACKUP_ADMIN roles could be created and granted to individuals that manage backups.

These categories provide a nice delineation, but some overlap is likely between the types of privileges you grant within either type of role. However, you should still try to avoid granting the same privileges to multiple roles. One easy way to manage this
separation is by granting roles to roles. SAP HANA supports granting privileges to roles and granting existing roles to other roles. Nesting roles within roles, when implemented properly, can greatly reduce maintenance on a security model. For example, you can grant the role DM_READ_ONLY to the ECORP_INFORMATION.MODELER role. In this way, you can ensure that developers also have SELECT access to the data mart schema even if you don’t explicitly grant those privileges in the ECORP_INFORMATION.MODELER role, ensuring that the information modeler has a basic level of access to the data mart schema.

Figure 5.3 shows how you can nest roles to simplify the security model design.

The ECORP_INFORMATION.MODELER parent role has been granted the DM_READ_ONLY, STG_READ_ONLY, and PACKAGE_ECORP_ROOT child roles to provide access to the schemas and packages used by the information modeler. Figure 5.3 only shows a few example privileges granted to the roles, shown to the right of each role name. In a true security model, you’d expect each role to be granted several different privileges.

5.3.2 Granting and Revoking Privileges with SQL

Using the SQL console and SQL statements, you can both grant privileges to and revoke privileges from roles. When granting privileges to specific objects, you’ll need to first recognize the object type. Specifically, you’ll need to use a different syntax for repository-based objects than for standard catalog objects. Repository-based objects are owned by the _SYS_REPO system account. With this account, you have no way to delegate grant and revoke privileges to other users in the system. For example, you can’t log on as _SYS_REPO and grant SELECT to another user while including the WITH ADMIN OPTION or WITH GRANT OPTION statement options. Because of this limitation, you must use a series of system-delivered stored procedures to grant privileges for repository-based objects owned by _SYS_REPO. These stored procedures are activated in the _SYS_REPO schema and can be called directly, assuming you have the EXECUTE privilege for the procedures. Standard catalog objects can be granted or revoked using the standard SQL GRANT or REVOKE statements. Let’s look at a series of SQL statements used to grant privileges based on the object and privilege type.

System Privileges

You can grant system privileges to roles using the GRANT SQL syntax. The following three SQL statements are separated by concluding semicolons:

- `GRANT <SYSTEM_PRIVILEGE> TO <ROLENAME>;`
- `GRANT <SYSTEM_PRIVILEGE> TO <ROLENAME> WITH ADMIN OPTION;`
- `GRANT DATA ADMIN, EXPORT, IMPORT TO ROLE_SEC_ADMIN WITH ADMIN OPTION;`

The first statement provides the general syntax, with variables indicated by text within angle brackets (`<>`). The variables must be replaced with items unique to your environment. The `<SYSTEM_PRIVILEGES>` variable can be replaced with a comma-separated list of system privileges, as indicated in the third SQL example. The `<ROLENAME>` variable will be replaced with the name of an activated role within the system.

The second SQL statement includes the WITH ADMIN OPTION at the end of the statement. This option allows the grantee of the role to subsequently grant the same privilege to another user or role. This capability is typically reserved for user accounts or roles that manage security models. The final SQL statement provides a full example in which you’ll grant system privileges to the role ROLE_SEC_ADMIN. This statement includes the WITH ADMIN OPTION, which allows the grantee to grant the same privileges to other users or roles.
You can also revoke system privileges using the `REVOKE` SQL syntax, as follows:

```
REVOKE <SYSTEM_PRIVILEGE> FROM <ROLENAME>;
REVOKE DATA ADMIN, EXPORT, IMPORT FROM ROLE_SEC_ADMIN;
```

The first statement provides the general syntax, and the second SQL statement provides a full example in which the listed system privileges are removed from the role ROLE_SEC_ADMIN.

**Note**

As we continue through this section, the same rules will apply to the each of the examples: The first and/or second example is a generic template that contains variables within angle brackets that need to be replaced with items unique to your environment, and the final SQL statement will provide a full example.

### Catalog Schema Privileges

If a schema is owned by a standard database user, you can use the syntax shown in Listing 5.1 to grant one or more privileges to a role.

```
GRANT <SCHEMA_PRIVILEGE> ON SCHEMA <SCHEMA_NAME> TO <ROLENAME>;
GRANT <SCHEMA_PRIVILEGE> ON SCHEMA <SCHEMA_NAME> TO <ROLENAME> WITH GRANT OPTION;
GRANT SELECT, EXECUTE ON SCHEMA "MySchema" TO MYSCHEMA_READONLY_ACCESS;
```

**Listing 5.1 Three Example SQL Statements Granting Privileges to a Role**

The `<SCHEMA_PRIVILEGES>` variable can be replaced with a comma-separated list of schema object privileges, as indicated in the example.

To remove the privileges, use the following syntax:

```
REVOKE <SCHEMA_PRIVILEGE> ON SCHEMA <SCHEMA_NAME> FROM <ROLENAME>;
REVOKE SELECT, EXECUTE ON SCHEMA "MySchema" FROM MYSCHEMA_READONLY_ACCESS;
```

**Note**

You can also revoke system privileges using the `REVOKE` SQL syntax, as follows:

```
REVOKE <SYSTEM_PRIVILEGE> FROM <ROLENAME>;
REVOKE DATA ADMIN, EXPORT, IMPORT FROM ROLE_SEC_ADMIN;
```

The first statement provides the general syntax, and the second SQL statement provides a full example in which the listed system privileges are removed from the role ROLE_SEC_ADMIN.

### Repository Schema Privileges

When a schema is created in the SAP HANA development repository, that schema is owned by the system user _SYS_REPO; these schemas are repository schemas. In these cases, you can’t use the standard `GRANT` and `REVOKE` SQL syntax to manage permissions for a given role. Instead, you must use a specific stored procedure to grant (and another to revoke) repository schema privileges.

To grant repository schema privileges, you’ll need to execute a stored procedure in the _SYS_REPO schema named `GRANT_SCHEMA_PRIVILEGE_ON_ACTIVATED_CONTENT`. To use this procedure, the security administrator must have the `EXECUTE` object privilege on the stored procedure. This procedure has three input parameters: The first parameter can accommodate a comma-separated array of schema privileges, the second parameter can accommodate a single schema name, and the third parameter can accommodate a role name.

In the following example, you’ll need to replace `<OBJECT_PRIVILEGES>` with one or more schema privileges; replace `<SCHEMA_NAME>` with the case-sensitive name of the schema, enclosed in double quotes if lowercase or mixed case; and replace `<ROLE-NAME>` with the name of a role defined in the SAP HANA system. To execute a stored procedure, use the SQL `CALL` function followed by the name of the stored procedure, as in the following statement:

```
CALL "_SYS_REPO"."GRANT_SCHEMA_PRIVILEGE_ON_ACTIVATED_CONTENT" (<OBJECT_PRIVILEGES>, <SCHEMA_NAME>, <ROLENAME>);
```

The following SQL statement provides a full example in which multiple schema privileges are granted to the MYSCHEMA_ADMIN role within the MySchema schema:

```
CALL "_SYS_REPO"."GRANT_SCHEMA_PRIVILEGE_ON_ACTIVATED_CONTENT" ('SELECT, EXECUTE, UPDATE, CREATE ANY', 'MySchema', 'MYSCHEMA_ADMIN');
```

**Listing 5.2 Two Example SQL Statements Granting and Then Revoking Repository-Owned Schema Privileges to a Database Role**

To revoke repository schema privileges, you’ll need to execute a stored procedure in the _SYS_REPO schema named `REVOKE_SCHEMA_PRIVILEGE_ON_ACTIVATED_CONTENT`. This procedure has three input parameters, which are the same as for the previous procedure used to grant the privileges, as shown in Listing 5.2.

```
CALL "_SYS_REPO"."REVOKE_SCHEMA_PRIVILEGE_ON_ACTIVATED_CONTENT" (<OBJECT_PRIVILEGES>, <SCHEMA_NAME>, <ROLENAME>);
CALL "_SYS_REPO"."REVOKE_SCHEMA_PRIVILEGE_ON_ACTIVATED_CONTENT" ('SELECT, EXECUTE, DROP, CREATE ANY', 'MySchema', 'MYSCHEMA_ADMIN');
```

**Repository Schema Privileges**

When a schema is created in the SAP HANA development repository, that schema is owned by the system user _SYS_REPO; these schemas are repository schemas. In these cases, you can’t use the standard `GRANT` and `REVOKE` SQL syntax to manage permissions for a given role. Instead, you must use a specific stored procedure to grant (and another to revoke) repository schema privileges.

To grant repository schema privileges, you’ll need to execute a stored procedure in the _SYS_REPO schema named `GRANT_SCHEMA_PRIVILEGE_ON_ACTIVATED_CONTENT`. To use this procedure, the security administrator must have the `EXECUTE` object privilege on the stored procedure. This procedure has three input parameters: The first parameter can accommodate a comma-separated array of schema privileges, the second parameter can accommodate a single schema name, and the third parameter can accommodate a role name.

In the following example, you’ll need to replace `<OBJECT_PRIVILEGES>` with one or more schema privileges; replace `<SCHEMA_NAME>` with the case-sensitive name of the schema, enclosed in double quotes if lowercase or mixed case; and replace `<ROLE-NAME>` with the name of a role defined in the SAP HANA system. To execute a stored procedure, use the SQL `CALL` function followed by the name of the stored procedure, as in the following statement:

```
CALL "_SYS_REPO"."GRANT_SCHEMA_PRIVILEGE_ON_ACTIVATED_CONTENT" (<OBJECT_PRIVILEGES>, <SCHEMA_NAME>, <ROLENAME>);
```

The following SQL statement provides a full example in which multiple schema privileges are granted to the MYSCHEMA_ADMIN role within the MySchema schema:

```
CALL "_SYS_REPO"."GRANT_SCHEMA_PRIVILEGE_ON_ACTIVATED_CONTENT" ('SELECT, EXECUTE, UPDATE, CREATE ANY', 'MySchema', 'MYSCHEMA_ADMIN');
```

**Listing 5.2 Two Example SQL Statements Granting and Then Revoking Repository-Owned Schema Privileges to a Database Role**

To revoke repository schema privileges, you’ll need to execute a stored procedure in the _SYS_REPO schema named `REVOKE_SCHEMA_PRIVILEGE_ON_ACTIVATED_CONTENT`. This procedure has three input parameters, which are the same as for the previous procedure used to grant the privileges, as shown in Listing 5.2.

```
CALL "_SYS_REPO"."REVOKE_SCHEMA_PRIVILEGE_ON_ACTIVATED_CONTENT" (<OBJECT_PRIVILEGES>, <SCHEMA_NAME>, <ROLENAME>);
CALL "_SYS_REPO"."REVOKE_SCHEMA_PRIVILEGE_ON_ACTIVATED_CONTENT" ('SELECT, EXECUTE, DROP, CREATE ANY', 'MySchema', 'MYSCHEMA_ADMIN');
```
Catalog Object Privileges

If a catalog object is owned by a standard database user, you can use the following SQL syntax to grant catalog object privileges to a role. The `<OBJECT_PRIVILEGE>` variable can be replaced with a comma-separated array of privileges. The `WITH GRANT OPTION`, when specified, allows the grantee the ability to subsequently grant the same privileges to other users in the system. The example shown in Listing 5.3 assumes that the schema is not a repository-based schema.

```sql
GRANT <OBJECT_PRIVILEGE> ON <OBJECT_NAME> TO <ROLENAME>;
GRANT <OBJECT_PRIVILEGE> ON <OBJECT_NAME> TO <ROLENAME> WITH GRANT OPTION;
GRANT SELECT, INSERT, UPDATE, DELETE ON "MySchema"."MyTable" TO MYSCHEMA_ADMIN;
```

Listing 5.3 Three Example SQL Statements Granting Object Privileges to a Database Role

To remove privileges, use the following syntax:

```sql
REVOKE <OBJECT_PRIVILEGE> ON <OBJECT_NAME> FROM <ROLENAME>;
REVOKE SELECT, INSERT, UPDATE, DELETE ON "MySchema"."MyTable" FROM MYSCHEMA_ADMIN;
```

Repository Object Privileges

When a catalog object is created in the SAP HANA development repository, that object is owned by the system user _SYS_REPO. We call these repository catalog objects. In these cases, you can’t use the standard GRANT and REVOKE SQL syntax to manage permissions on a given user account. Instead, you must use a specific stored procedure to grant (and another to revoke) repository catalog object privileges.

To grant repository catalog object privileges, you’ll need to execute a stored procedure in the _SYS_REPO schema, named GRANT_PRIVILEGE_ON_ACTIVATED_CONTENT. To use this procedure, the security administrator must have the object privilege EXECUTE on the stored procedure. This procedure has three input parameters: The first parameter can accommodate a comma-separated array of schema privileges, the second parameter can accommodate a single schema and object name, and the third parameter can accommodate a role name.

In the example shown in Listing 5.4, you’ll need to replace `<OBJECT_PRIVILEGES>` with one or more schema privileges; replace `<OBJECT_NAME>` with the case-sensitive and fully qualified name of your catalog object, enclosed in double quotes if the name is lowercase or mixed case; and replace `<ROLENAME>` with the name of a role defined in the SAP HANA system.

```sql
CALL "_SYS_REPO"."GRANT_PRIVILEGE_ON_ACTIVATED_CONTENT"('SELECT, EXECUTE', 'StagingMart'.'e-corp.sales.tables::STG_CUSTOMERS', 'STAGING_READ_ONLY');
CALL "_SYS_REPO"."REVOKE_PRIVILEGE_ON_ACTIVATED_CONTENT"('SELECT, EXECUTE', 'StagingMart'.'e-corp.sales.tables::STG_CUSTOMERS', 'STAGING_READ_ONLY');
```

Listing 5.4 Two Example SQL Statements Granting a Repository-Owned Object's Object Privileges to a Database Role

To revoke repository catalog object privileges, you’ll need to execute a stored procedure in the _SYS_REPO schema named REVOKE_PRIVILEGE_ON_ACTIVATED_CONTENT. This procedure has three input parameters, which are identical to those used in the previous procedure to grant the privileges. Use the syntax shown in Listing 5.5 to call the stored procedure equipped to revoke catalog object privileges previously granted.

```sql
CALL "_SYS_REPO"."REVOKE_PRIVILEGE_ON_ACTIVATED_CONTENT"('SELECT, EXECUTE', 'StagingMart'.'e-corp.sales.tables::STG_CUSTOMERS', 'STAGING_READ_ONLY');
```

Listing 5.5 Two Example SQL Statements Revoking a Repository-Owned Object's Object Privileges to a Database Role

Structured Privileges

Structured privileges are a special type of object used to force the system to provide additional access restrictions against a column view or catalog view. In Chapter 2, we introduced you to analytic privileges, which are used with SAP HANA information views to provide data access to an activated information view. Technically, analytic privileges are a type of structured privilege, but most GUIs refer to them as analytic privileges. Repository-based analytic privileges are owned and managed by the system account _SYS_REPO. However, structured privileges can be owned by any user with system privileges.

If a structured privilege is owned by a standard database user, you can use the GRANT STRUCTURED_PRIVILEGE SQL syntax to grant such privileges to a user account. To grant a structured privilege, you must own the structured privilege because no WITH GRANT OPTION is available to allow another user to grant this privilege. Note that this syntax won’t work on repository-based analytic privileges. In the following example, replace the `<STRUCTURED_PRIVILEGE_NAME>` variable with the name of the structured privilege:
Now, we'll walk you through the process of creating a catalog view and a structured privilege and the process of granting the privilege to a user. Start by using the SQL code shown in Listing 5.6 to create a standard catalog view.

```
CREATE VIEW "MySchema"."MYVIEW" ( "MYKEYCOLUMN", "STATE", "CITY", "POSTALCODE" ) AS select T0."MYKEYCOLUMN", T0."STATE", T0."CITY", T0."POSTALCODE"
FROM "SYSTEM"."MYTABLENAME" T0
WITH STRUCTURED PRIVILEGE CHECK;
```

### Listing 5.6 CREATE VIEW SQL Code

You’ll then create a structured privilege referencing the view, which will apply a filter on the column named STATE. The following SQL statement will create a structured privilege that will filter the STATE to the value TN; after the privilege name, use the FOR SELECT syntax to indicate that the privilege should be applied when querying the view:

```
CREATE STRUCTURED PRIVILEGE "MyStructuredPriv" FOR SELECT ON "MySchema"."MYVIEW" WHERE "STATE" = 'TN';
```

You can then grant the structured privilege to the user JONATHAN, for example. Once granted, the user will only see value from the view where "STATE" = 'TN':

```
GRANT STRUCTURED PRIVILEGE "MyStructuredPriv" TO ROLE_STATE_TN;
```

To revoke the structured privilege, use the REVOKE STRUCTURED PRIVILEGE SQL syntax:

```
REVOKE STRUCTURED PRIVILEGE "MyStructuredPriv" FROM ROLE_STATE_TN;
```

To allow other users or roles to grant the structured privilege, you can include the WITH GRANT OPTION. For example, to allow the role ROLE_SEC_ADMIN the ability to grant the structured privilege to other users, you must include the WITH GRANT OPTION, as in the following statement:

```
GRANT STRUCTURED PRIVILEGE "MyStructuredPriv" TO ROLE_SEC_ADMIN WITH GRANT OPTION;
```

### Remote Sources

To grant access to a remote source or an SAP HANA smart data access federated database connection, issue a SQL statement as follows:

```
GRANT <SOURCE_PRIVILEGE> ON REMOTE SOURCE <SOURCE_NAME> TO <ROLENAME>;
```

The `<SOURCE_PRIVILEGE>` variable can be replaced with a comma-separated array of privileges specific to remote source objects, and the `<SOURCE_NAME>` variable can be replaced with the name of the remote source.

To remove remote source privileges, use the REVOKE SQL statement. The following statement will remove the privileges specified in the `<SOURCE_PRIVILEGES>` variable array:

```
REVOKE <SOURCE_PRIVILEGE> ON REMOTE SOURCE <SOURCE_NAME> FROM <ROLENAME>;
```

### Repository Analytic Privileges

When you create a standard analytic privilege, that privilege is owned by the system user _SYS_REPO. Analytic privileges are a type of structured privilege, but you can’t use the standard GRANT and REVOKE SQL syntax to assign analytic privileges to a user. Instead, you must use a specific stored procedure to grant (and another to revoke) analytic privileges for users.

To grant an analytic privilege to a user, you must execute the stored procedure `GRANT_ACTIVATED_ANALYTIC_PRIVILEGE`, stored in the _SYS_REPO schema. To use this procedure, the security administrator must have the EXECUTE object privilege on the stored procedure. The procedure provides two input parameters. The first parameter should contain the name of the analytic privilege, and the second input parameter will contain the name of the user, as shown in Listing 5.7.

```
CALL "_SYS_REPO"."GRANT_ACTIVATED_ANALYTICAL_PRIVILEGE" (<ANALLYTIC_PRIVILEGE_NAME>,'<ROLENAME>');
```

### Listing 5.7 Two Example SQL Statements Granting Analytic Privileges to a Database Role
To revoke an analytic privilege from a user, you must execute the stored procedure `REVOKE_ACTIVATED_ANALYTIC_PRIVILEGE`, stored in the `_SYS_REPO` schema. This stored procedure uses the same two input parameters, as shown in Listing 5.8.

```sql
CALL "SYS_REPO"."REVOKE_ACTIVATED_ANALYTICAL_PRIVILEGE"
("<ANALYTIC_PRIVILEGE_NAME>','<ROLENAME>");
CALL "SYS_REPO"."REVOKE_ACTIVATED_ANALYTICAL_PRIVILEGE"
(""e-corp.sales.security/ap_sales_reporting_region"','ROLE_REGION' );
```

**Listing 5.8 Two Example SQL Statements Revoking Analytic Privileges to a Database Role**

**Application Privileges**

When you create an application privilege, it’s owned by the system user `_SYS_REPO`. To grant or revoke application privileges, you must use a specific stored procedure, as follows:

```sql
CALL "SYS_REPO"."GRANT_APPLICATION_PRIVILEGE"
("<APPLICATION_PRIVILEGE_NAME>','<ROLENAME>");
CALL "SYS_REPO"."GRANT_APPLICATION_PRIVILEGE"
(""sap.hana.admin::Administrator"','ROLE_APP');
```

Use the following syntax to revoke an application privilege:

```sql
CALL "SYS_REPO"."REVOKE_APPLICATION_PRIVILEGE"
("<APPLICATION_PRIVILEGE_NAME>','<ROLENAME>");
CALL "SYS_REPO"."REVOKE_APPLICATION_PRIVILEGE"
(""sap.hana.admin::Administrator"','ROLE_APP');
```

**Package Privileges**

To grant access to a package in the SAP HANA repository, issue a SQL statement like the one shown in Listing 5.9.

```sql
GRANT <PACKAGE_PRIVILEGE> ON <"package.subpackage"> TO <ROLENAME> WITH GRANT OPTION;
GRANT REPO.READ, REPO.EDIT_NATIVE_OBJECTS ON "e-corp" TO ROLE_ECORP_PKG_ROOT_ADMIN;
```

**Listing 5.9 Two Example SQL Statements Granting Package Privileges to a Database Role**

The `<PACKAGE_PRIVILEGE>` variable can be replaced with a comma-separated array of package privileges, and the `<package.subpackage>` variable can be replaced with the case-sensitive name of the package and subpackage hierarchy. Each node of the package hierarchy should be separated by a period. The WITH GRANT OPTION allows the grantee to grant the same privileges to other users.

To remove package privileges, use the `REVOKE` SQL statement. The following example SQL statement will remove the privileges specified in the `<PACKAGE_PRIVILEGE>` variable array:

```sql
REVOKE <PACKAGE_PRIVILEGE> ON <"package.subpackage"> FROM <ROLENAME>;
REVOKE REPO.EDIT_NATIVE_OBJECTS ON "e-corp" FROM ROLE_ECORP_PKG_ROOT_ADMIN;
```

**Full SQL Syntax**


Search for the sections pertaining to the GRANT, REVOKE, CREATE, and DROP ROLE statements.

### 5.3.3 Granting and Revoking Privileges with the SAP HANA Cockpit

As an alternative to using SQL statements, you can use the SAP HANA cockpit GUI action called Manage Roles. This interface allows you to both grant and revoke privileges for a given standard role. Most users will find using the SAP HANA cockpit easier than memorizing and entering in SQL statements.

To access the Manage Roles area, start at the SAP HANA cockpit Home screen, locate an SAP HANA system in the Resource Directory. Connect to the desired tenant database or SYSTEMDB, which will take you to the System Overview window. Using the Filter by Area dropdown list, choose the option Security. Click the Manage Roles link in the User & Role Management tile, which will launch the Manage Roles window where you’ll see a list of roles on the left and selected roles properties on the right.

The role list provides a search filter bar at the top of the list that you can use to filter the list of roles. Select a role to reveal its assigned privileges in the Role Details pane to the right.

Figure 5.4 shows the Role Details interface within the SAP HANA cockpit for the TEST_ROLE role. In the window, you’ll see tabs for each privilege category, such as the Roles, System Privileges, Object Privileges, Analytic Privileges, Is Part of Roles, Package
5 Database Roles

Privileges, and Privileges on Users tabs. Each tab is an interface via which specific types of privileges can be granted or revoked. In this section, we’ll look at the System Privileges, Object Privileges, Analytic Privileges, and Package Privileges tabs in detail. The Roles and Is Part of Roles tabs will be discussed in detail in Section 5.4.2.

Figure 5.4 Managing Privileges Assigned to Role in the SAP HANA Cockpit

System Privileges

Let’s first review the workflows necessary to grant and revoke system privileges. The second tab in the Role Details window is named System Privileges. You’ll use this tab to list, grant, or revoke system privileges for a selected role.

Privileges listed in this interface are privileges that are already granted to the role. If the list is blank, nothing is currently granted. To grant a system privilege, click the Edit button located on the far-right of the interface near the Search field. Once you click the Edit button, new buttons labeled Save, Cancel, Add, and Remove will appear. Click the Add button to open the Select System Privileges popup window. Within this window, you’ll see a search bar and a list of system privileges each with checkboxes to their left. Use the search bar to filter the list based on the entered text. Then, scroll through the list and select the checkbox beside each system privilege you want to grant. With one or more system privileges selected, click the OK button to add the desired items to the user. Your selected system privileges should now appear in the System Privileges section, as shown in Figure 5.5.

To assign the Grantable to Others privilege, locate a system privilege in the list. To the right, move the slider button to Yes to allow the user to grant the same privilege to another user or role. Repeat this process for each system privilege in the list. Once you’ve added all the desired system privileges to the list, click the Save button to grant the privileges.

Figure 5.5 List of Selected System Privileges

To revoke a system privilege, start by clicking the Edit button again, select the checkbox beside each system privilege in the list, and click the Remove button to revoke the privileges. Repeat the process for each system privilege you want to revoke. Once you’ve removed all the desired system privileges, click the Save button. The removed privileges will be revoked, and the user object will be updated.

During the process of adding or removing, you can click the Cancel button to discard any unsaved changes. Clicking the Cancel button will deactivate the Save, Add, Cancel, and Remove buttons and return you to a read-only list of currently granted system privileges.

Object Privileges

The third tab in Role Details window is named Object Privileges. The process for granting or revoking object privileges is slightly different than the process used for system privileges in the previous section. Like before, privileges listed in this interface are those privileges already granted to the role. If the list is blank, nothing is currently granted. To grant an object privilege, click Edit button located on the far-right side of the interface just to the right of the Search field. Once you click the Edit button to enable edit mode, new buttons labeled Save, Cancel, Add Object, Change Privileges,
and Remove Object will appear. Click the Add Object button to open the Select Objects popup window. Within this window, you'll see a search bar and a list of catalog objects.

Use the search bar to filter the list based on the entered text. The search allows for multiple words allowing you to enter the object name and type. For example, you can enter the text "SYS_BIC schema" to filter objects with that name and of a catalog object type. Scroll through the list and highlight the catalog object you want to assign privileges to. Figure 5.6 shows an example of how you can use multiple words to filter to just the SYS_BIC schema.

Figure 5.6 Filtering the List of Object Privileges When Searching for a Catalog Object

Once you click the desired catalog object, the Add Object with Privileges window will appear. In this window, you'll see a list of possible object privileges that can be assigned. Select the checkbox beside each privilege you want to grant. In addition, for each privilege, you can use the Yes/No slider under Grantable to Others to allow the privilege to be granted to other users. Figure 5.7 shows an example of the interface used to assign object privileges and the Yes/No slider. Click the OK button located at the bottom right of the interface to continue.

Figure 5.7 Selecting and Granting Object Privileges for a Given Catalog Object

Note that once the Save button is clicked, you'll be returned to the list of privileges and be required to click the Edit button again to make further changes.

To change assigned object privileges, while in edit mode, click an assigned object and its assigned privilege from the list and then click the Change Privileges button to open a new popup window called Change Privileges of concatenated with the name of the object. A list of assigned and unassigned privileges for the object will appear. Select the checkbox to grant the privilege or deselect the checkbox to revoke the privilege. Use the Yes/No slider to manage the grantable to others privilege. Click the OK button at the bottom right of the window to apply your changes. Again, to make the changes active, click the Save button or continue making other changes if necessary until you're ready to activate all changes.

Analytic Privileges

The fourth tab in the Role Details interface is Analytic Privileges. The process for granting or revoking analytic privileges is like the process used for system privileges discussed in this section.
The analytic privileges listed in this interface are those that are already granted to the role. If the list is blank, nothing is currently granted. To grant an analytic privilege, click the Edit button located on the far-right side of the interface and just to the right of the Search field to enable edit mode. Once you click the Edit button, new buttons labeled Cancel, Add, and Remove will appear. Click the Add button to open the Select Analytic Privileges popup window. Within this window, you'll see a search bar and a list of runtime analytic privileges each with a checkbox. Use the search bar to filter the list based on the entered text. Then, scroll through the list and select the checkbox beside the analytic privilege you want to grant. With one or more analytic privileges checked, click the OK button to add the desired items to the user. Your selected analytic privileges should now appear in the list.

To assign the Grantable to Others privilege, while in edit mode, locate an analytic privilege in the list. To the right, you’ll see a Yes/No slider that you’ll move to Yes, thus allowing the user to grant the same privilege to another user or role. Note that the slider will not be enabled if your current logon user account does not have the privileges necessary to grant Grantable to Others. Repeat this process for each analytic privilege in the list. Once you’ve added all the desired analytic privileges to the list, click the Save button to grant the privileges.

To revoke an analytic privilege, start by clicking the Edit button again to enable edit mode, then select the checkbox beside each analytic privilege in the list, and click the Remove button to revoke the privileges. Repeat the process for each analytic privilege you want to revoke. Once you’ve removed all the desired analytic privileges, click the Save button. The removed privileges will be revoked, and the user object will be updated.

During the process of adding or removing privileges, you can click the Cancel button to discard any unsaved changes. Clicking the Cancel button will deactivate the Add, Cancel, and Remove buttons and return you to a read-only list of currently granted analytic privileges.

Package Privileges

The seventh tab in Assign Privileges window is named Package Privileges. The process for granting or revoking package privileges like the process used for object privileges. Packages and privileges listed in this interface are those that are already granted to the role. If the list is blank, nothing is currently granted. To grant a package privilege, click the Edit button located on the far-right side of the interface just to the right of the search field. Once you click the Edit button to enable edit mode, new buttons labeled Cancel, Add Package, Change Privileges, and Remove Package will appear. Click the Add Package button to open the Select Objects popup window. Within this window, you’ll see a search bar and a list of packages. Use the search bar to filter the list based on the entered text. Scroll through the list and highlight the package you want to assign privileges to.

Once you click the desired package, a new window titled Add Package with Privileges will appear. In this window, you’ll see a list of possible package privileges that can be assigned. Select the checkbox beside each privilege you want to grant. In addition, for each privilege, you can use the Yes/No slider under Grantable to Others to allow the privilege to be granted to other users. Click the OK button located at the bottom right of the interface to continue.

You should now see the selected packages and the privileges you assigned listed in the Package Privileges tab.

To revoke a package privilege, click an assigned package and its assigned privilege from the list and then click the Remove Package button to revoke the privilege. To make the changes active, click the Save button or continue making other changes if necessary. Note that, once the Save button is clicked, you’ll be returned to the list of privileges and be required to click the Edit button again to make further changes.

To change assigned object privileges, while in edit mode, click an assigned object and its assigned privilege from the list and then click the Change Privileges button to open a new popup window titled Change Privileges of concatenated with the name of the package. A list of assigned and unassigned privileges for the package will appear. Select the relevant checkbox to grant the privilege or deselect the checkbox to revoke the privilege. Use the Yes/No slider to manage the grantable to others privilege. Click the OK button at the bottom right of the window to apply your changes. Again, to make the changes active, click the Save button or continue making other changes if necessary until you’re ready to activate all changes.

5.3.4 Granting and Revoking Privileges with the SAP HANA Web-Based Development Workbench

The SAP HANA Web-Based Development Workbench security manager, hosted within the XS engine, provides an interface to create users, create roles, or manage security. You can access SAP HANA Web-Based Development Workbench via a supported Internet browser. The following two URLs can be customized to match the details of your environment:
In the second example, replace `<sap_hana_host>` with the host name of the SAP HANA system in your environment. Replace `<instance_number>` with the two-digit instance corresponding to your SAP HANA system.

For secure access, the following examples should help you construct the correct URL:

- http://<sap_hana_host>:43<instance_number>/sap/hana/ide/security
- https://<sap_hana_host>: 43<instance_number>/sap/hana/ide/security

Once you have access to the SAP HANA Web-Based Development Workbench, expand the Security node on the right side of the window. Expand Roles to reveal a list of roles. Click a role to open the management interface for the given role. With the selected Role window open, you can begin granting and revoking privileges. Just below the role name, you’ll see the Granted Roles, Part of Roles, System Privileges, Object Privileges, Analytic Privileges, Package Privileges, and Application Privileges (not visible in screenshot) tabs. Figure 5.8 shows an example of this role interface where you can assign privileges to a role.

![Figure 5.8](image-url) Layout of the Role Management Interface Where You Can Assign Privileges to a Role

In the absence of the SAP HANA cockpit, you can use this GUI to manage security for standard catalog objects and _SYS_REPO-owned catalog objects that were created at design time.

While working in this interface, if any activation issues arise, warning messages will appear in the status pane at the bottom of the Role Editor window. The status pane has a dark background, and error messages will appear in red text. Errors are generated only during an attempt to activate or save changes.

**System Privileges**

To grant or revoke system privileges within a role, locate the System Privileges tab. To add a system privilege, click the plus sign (+) to open the Find System Privileges tab. You can use the search box to filter the list based on the entered text. Select one or more privileges from the list and click OK to continue. Your selected system privileges will now be listed in the tab. To assign the Grantable to Other Users and Roles, select the privilege and locate the checkbox to the right in the Details for area to the right of each selected system privilege as needed. Figure 5.8 also shows an example where system privileges have been granted to the role TEST_ROLE.

To revoke a system privilege, select the privilege from the list and click the red X. To save your changes, press `Ctrl+S`. Alternatively, click the save icon (floppy disk) at the top left to save and activate the privilege. Look for activation errors just below the privileges tab in the section with a dark background. Errors will be listed in red text. White text will indicate that the change was successful.

**Object Privileges**

The fourth tab in the user role editor interface is called Object Privileges. You’ll use this tab to grant or revoke object privileges. To add an object privilege, click the plus sign (+) and search for the desired catalog objects. A search dialog will appear, allowing you to search for an object. Unlike in the SAP HANA cockpit, the SAP HANA Web-Based Development Workbench also provides additional search checkbox options based on the object type, which may help you pinpoint specific objects by type. Select the desired objects and click OK. A list of selected objects will appear in the tab, as shown in Figure 5.9. To the right of each object is a list of applicable privileges that can be granted using checkboxes. Checked items will be granted and unchecked boxes will be revoked when you save the changes. The Grantable to Others option is also available using the Yes and No radio buttons.

To revoke individual privileges previously granted, select the object in the left pane and then deselect any existing privileges in the pane to the right. To revoke all privileges for an object privilege, select the object from the list and click the red X. This action will remove the object and revoke all currently granted privileges. To save your changes, press `Ctrl+S` or click the disk icon to save and activate the privilege. Look for activation errors just below the privileges tab in the section with a dark background.
5 Database Roles

background. Errors will be listed in red text. White text will indicate that the change was successful.

Analytic Privileges
The fifth tab in the role editor interface is called Analytic Privileges. To add an analytic privilege to the list, click the plus sign (+) and search for the desired analytic privilege. To revoke an analytic privilege, select the privilege from the list and click the red X to remove the analytic privilege. To save your changes, press (Ctrl)+(S) or click the disk icon to save and activate the privilege.

Package Privileges
The sixth tab in the role editor interface is called Package Privileges. To add a package and configure its privileges, click the plus sign (+) and search for the desired package hierarchy level. The Find Package search interface window will appear allowing you to search for and select one or more package levels. Select the packages you want to secure and click the OK button to return to the previous screen where you can grant the desired package privileges. Figure 5.10 shows an example of the Package Privileges interface where you can assign packages and grant privileges within a role.

To revoke a package privilege, selected the package hierarchy level from the list and click the red X to remove it. To save your changes, press (Ctrl)+(S) or click the disk icon to save and activate the privilege.

Application Privileges
The seventh tab in the role editor interface is called Application Privileges (see Figure 5.11). To add an application privilege to the list, click + and search for the desired application privilege. To revoke an application privilege, select the privilege from the list and click the red X to remove the application privilege. To save your changes, press (Ctrl)+(S) or click the disk icon to save and activate the privilege.

Figure 5.9 Working with a Role's Object Privileges in the SAP HANA Web-Based Development Workbench Security Manager

Figure 5.10 Working with Package Privileges in the SAP HANA Web-Based Development Workbench Security Interface

Figure 5.11 Working with Application Privileges in the SAP HANA Web-Based Development Workbench Security Interface
5.4 Managing Nested Roles

You’ll use three basic interfaces to grant and revoke nested roles or roles granted to other roles: the SQL console to issue SQL statements, the SAP HANA cockpit, or the SAP HANA Web-Based Development Workbench. We’ll discuss these options next.

5.4.1 Granting and Revoking Roles with SQL

When granting roles to another role, you must first determine the role type. If the role was created by a standard user account or if the role was provisioned by the system during installation, we refer to the role as a standard role. These roles often are created using the CREATE ROLE SQL syntax. Standard roles can be granted and revoked with the GRANT and REVOKE SQL statements.

Alternatively, if the role was created as a development artifact and owned by the system account _SYS_REPO, we refer to the role as a repository role. We’ll discuss the process for nesting repository roles with a GUI in Chapter 6. However, you can grant a repository role to a standard role using SQL, which we’ll discuss in this section.

Granting a standard role to a different standard role with SQL is performed using a syntax similar to the syntax used when granting privileges. To grant a role to another role, you must use a GRANT SQL statement, as follows:

- GRANT <EXISTING_ROLENNAME> TO <TARGET_ROLENAME>;
- GRANT <EXISTING_ROLENAME> TO <TARGET_ROLENAME> WITH ADMIN OPTION;
- GRANT MODELING, DBA_COCKPIT, CONTENT_ADMIN TO MY_ROLE_NAME;

The <EXISTING_ROLENAME> variable can be replaced with a comma-separated list of existing roles, and the <TARGET_ROLENAME> variable can be replaced with the name of the target role. The WITH ADMIN OPTION, when included, allows the existing role to be granted to other users or roles.

To revoke a role using SQL, you must issue a REVOKE SQL statement, as follows:

- REVOKE <EXISTING_ROLENAME> FROM <TARGET_ROLENAME>;
- REVOKE MODELING, DBA_COCKPIT, CONTENT_ADMIN FROM MY_ROLE_NAME;

You might need to grant a repository-based role to a standard role using SQL. This action will be permanent if the repository role is not deleted at any point in the future.

Granting Standard Roles to Repository-Based Roles at Runtime

Keep in mind that you cannot permanently grant a standard role to a repository-based role using SQL, but technically the SQL will execute. For that granting action to be permanent, you must update the design-time version of the repository role and then save and activate the design-time script. This process might sound confusing, but remember that repository roles exist in both design time and runtime. Sometimes, you can manipulate runtime security for a repository object, but these settings will be lost when the system reboots.

When a role is created in the SAP HANA development repository, that role is owned by the system user _SYS_REPO. These roles are repository roles. You can’t use the standard GRANT and REVOKE SQL syntax to assign repository roles to a standard role; instead, you must use a specific stored procedure to grant repository roles to standard roles and use a different stored procedure to revoke repository roles from standard roles.

The first stored procedure is stored in the _SYS_REPO schema and is named GRANT_ACTIVATED_ROLE. To use this procedure, the security administrator must have the EXECUTE object privilege on the stored procedure. The procedure has two input parameters: The first input parameter requires the name of one repository role and doesn’t support entering multiple repository roles; the second input parameter requires the name of an existing standard role. The following SQL statement uses variables for the repository role name and standard role parameters:

- CALL ".SYS_REPO"."GRANT_ACTIVATED_ROLE"('REPOSITORY_ROLE_NAME', 'STANDARD_ROLE');

The following example demonstrates the execution of the procedure with sample parameters:

- CALL _SYS_REPO"."GRANT_ACTIVATED_ROLE"('sap.hana.ide.roles::Developer', 'TEST_ROLE');

The second stored procedure is stored in the _SYS_REPO schema and is named REVOKE_ACTIVATED_ROLE. To use this procedure, the security administrator must have the EXECUTE object privilege for the stored procedure. The procedure has two
input parameters: The first input parameter requires the name of one repository role and doesn’t support entering multiple repository roles; the second input parameter requires the name of a standard role. The following SQL statement uses variables for the repository role name and standard role parameters:

```sql
CALL "_SYS_REPO"."REVOKE_ACTIVATED_ROLE"('<REPOSITORY_ROLE_NAME>', '<ROLE_NAME>');
```

The following statement demonstrates the execution of the procedure with sample parameters:

```sql
CALL "_SYS_REPO"."REVOKE_ACTIVATED_ROLE"('sap.hana.ide.roles::Developer','TEST_ROLE');
```

### 5.4 Managing Nested Roles

See a search bar and a list of roles with checkboxes to their left. Use the search bar to filter the list. Then, scroll through the list and select the checkboxes beside the roles you want to nest. With one or more roles selected, click the OK button to add the desired items to the role. Your selected roles should now appear in the Roles section, as shown in Figure 5.12.

![Figure 5.12 List of Selected Roles](image)

To assign the Grantable to Others privilege, locate a role in the list. To the right, move the slider button to Yes to allow the user to grant the same role to another user or role. Repeat this process for each role in the list. Once you’ve added all the desired roles to the list, click the Save button to grant the roles.

### Repository Roles in the SAP HANA Cockpit

Based on our testing, you can grant a repository role to another role using the SAP HANA cockpit. However, we couldn’t revoke that role in the same way. Therefore, in our experience, the SAP HANA cockpit does not fully support repository-based roles. When working with repository-based roles, use the SAP HANA Web-Based Development Workbench security manager.

To revoke a role, start by clicking the Edit button again, then select the checkbox beside each role in the list, and click the Remove button to revoke the role. Repeat the process for each role you want to revoke. Once you’ve removed all the desired roles, click the Save button. The removed roles will be revoked, and the role object will be updated.
During the process of adding or removing roles, you can use the Cancel button to discard any unsaved changes. Clicking the Cancel button will deactivate the Save, Add, Cancel, and Remove buttons and return you to a read-only list of currently granted roles.

### 5.4.3 Granting and Revoking Roles with the SAP HANA Web-Based Development Workbench

As mentioned earlier, the SAP HANA Web-Based Development Workbench security manager, hosted within the XS engine, provides an interface to create users, create roles, or manage security. In summary, you can access the SAP HANA Web-Based Development Workbench security manager by updating the following two URLs to match the details of your environment:

- http://<sap_hana_host>:80<instance_number>/sap/hana/ide/security
- https://<sap_hana_host>:443<instance_number>/sap/hana/ide/security

Once you have access to the SAP HANA Web-Based Development Workbench security manager, expand the Security node on the right side of the window. Expand Roles to reveal a list of roles. Double-click a role to open the management interface for the given role. With the selected role management window open, you can begin granting and revoking roles using the Granted Roles tab.

To grant or revoke role within the selected role, locate the Granted Roles tab. To add a role, click the plus sign (+) to open the Find Role popup window. You can use the search box to filter the list. Select one or more roles from the list and click OK to continue. Your selected roles will now be listed in the Granted Roles tab. To assign the Grantable to Other Users and Roles, select the role and select the checkbox in the Details for area to the right of each selected role as needed.

To revoke a role, select the role from the list and click the red X. To save your changes, press 
Alternatively, click the save icon (floppy disk) at the top left to save and activate the role. Look for activation errors just below the Granted Roles tab in the section with a dark background. Errors will be listed in red text. White text will indicate that the change was successful.

### 5.5 Mapping LDAP Groups to Roles

Starting with SAP HANA 2.0 SPS 03, Lightweight Directory Access Protocol (LDAP)-based users can now be provisioned within SAP HANA, and their LDAP-managed passwords can be used for authentication. In addition to authentication, starting

with SAP HANA 2.0 SPS 00, you can also map SAP HANA database roles to LDAP groups to simplify authorization. Effectively, SAP HANA will look up a user’s group memberships in the LDAP directory. If the LDAP group is mapped to an SAP HANA role, the user will be granted the SAP HANA database role and its privileges. This check is performed each time a user is authenticated. Before you can use this functionality, you’ll need to set up and configure an LDAP provider within the SAP HANA system. We’ll discuss this setup in more detail in Chapter 12. In this section, we’ll review the process for mapping LDAP groups to roles using both SQL statements and the SAP HANA cockpit.

#### 5.5.1 Mapping Roles with SQL

You can use SQL to create an SAP HANA database role and map that role to an LDAP group in a single step. You can also modify an existing role and add an LDAP group. (Note that you can’t map an LDAP group to a repository role at this time.) Let’s first review the SQL required to create a role and map it to an LDAP group in a single step. The following SQL statement will create the SAP HANA role ECORP_SECURITY_ADMINS and map it to the LDAP group HANA Security Team:

```
CREATE ROLE ECORP_SECURITY_ADMINS LDAP GROUP 'CN=HANA_SECURITY_TEAM,CN=Users,DC=ecorp,DC=com';
```

To modify an existing role and add map and LDAP group, you would run the following SQL statement to add the LDAP group HANA Security Team to the SAP HANA role ECORP_SECURITY_ADMINS:

```
ALTER ROLE ECORP_SECURITY_ADMINS ADD LDAP GROUP 'CN=HANA_SECURITY_TEAM,CN=Users,DC=ecorp,DC=com';
```

To remove the LDAP group mapping from a role, run the following SQL statement:

```
ALTER ROLE ECORP_SECURITY_ADMINS DROP LDAP GROUP 'CN=HANA_SECURITY_TEAM,CN=Users,DC=ecorp,DC=com';
```

In the next section, we’ll review the process for mapping LDAP groups to SAP HANA roles using the SAP HANA cockpit.

#### 5.5.2 Mapping Roles with the SAP HANA Cockpit

As an alternative to using SQL statements, you can use the SAP HANA cockpit GUI action called Manage Roles to map an LDAP group to a role. This interface allows you...
to both add or remove LDAP groups from a standard role. Most users will find using
the SAP HANA cockpit easier than memorizing and entering in SQL statements.

To access the Manage Roles area, start at the SAP HANA cockpit Home screen and
locate an SAP HANA system in the Resource Directory. Connect to the desired tenant
database or SYSTEMDB, which will take you to the System Overview window. Using
the Filter by Area dropdown list, choose the option Security. Click the Manage Roles
link in the User & Role Management tile. This action will launch the Manage Roles
window where you’ll see a list of roles on the left and selected roles properties on the
right. The role list provides a search filter bar located at the top of the list that you can
use to filter the list of roles. Select a role to reveal its assigned privileges in the Role
Details pane to the right.

In the Role Details pane, locate the Edit button to the right of the role name, as shown
in Figure 5.13. The role edit pane will appear allowing you to modify basic information
for the role such as its Comments, Schema, and LDAP Groups.

In the LDAP Groups section, you’ll see the option to enable LDAP group mappings
titled Assign LDAP Groups. Change the radio button option to Yes to enable additional
LDAP group mapping options. In the LDAP Group Name field, click the Add
LDAP Groups button and then type the name of the LDAP group using the LDAP nam-
ing convention, as shown in Figure 5.14.

Figure 5.14 Configuring LDAP Group Mappings Using the SAP HANA Cockpit

To remove the mapping, click the remove icon (X in a circle) to the right of the listed LDAP group and save to
remove the mapping.

You now understand how you can use SQL and the SAP HANA cockpit to manage
LDAP group mappings to roles.

5.6 Summary

The process of creating and managing roles can be accomplished using SQL state-
ments, the SAP HANA cockpit, or the SAP HANA Web-Based Development Work-
bench security manager hosted in the XS engine. You can also use all three interfaces
to grant and revoke both privileges and roles with some limitations with the SAP
HANA cockpit and repository-based roles. In the next chapter, we’ll will explore
repository-based roles, which are an alternative to standard roles and which in most
cases provide advantages over standard roles.
## Contents

Preface ........................................................................................................................ 19  
Introduction ................................................................................................................... 27  

### 1 Managing Security with the SAP HANA Cockpit  

1.1 What Is the SAP HANA Cockpit? ................................................................................... 40  
1.1.1 SAP HANA Cockpit Architecture ...................................................................... 40  
1.1.2 Getting Started with the SAP HANA Cockpit .............................................. 41  
1.1.3 Navigating SAP HANA Cockpit ......................................................................... 51  
1.2 Security Areas in SAP HANA Cockpit ....................................................................... 54  
1.2.1 User & Role Management Area ....................................................................... 55  
1.2.2 Data Encryption .................................................................................................... 56  
1.2.3 Authentication ...................................................................................................... 57  
1.2.4 Security Related Links ......................................................................................... 58  
1.2.5 Anonymization Report ........................................................................................ 59  
1.2.6 Auditing .................................................................................................................. 60  
1.3 SAP HANA Database Explorer and SQL Console .................................................... 61  
1.4 Summary ....................................................................................................................... 64  

### 2 Introduction to SAP HANA Privileges  

2.1 Privileges within SAP HANA ......................................................................................... 66  
2.1.1 System Privileges .................................................................................................. 66  
2.1.2 Object Privileges ................................................................................................... 67  
2.1.3 Analytic Privileges ................................................................................................ 69  
2.1.4 Package Privileges ................................................................................................ 71  
2.1.5 Application Privileges .......................................................................................... 72  
2.2 Privilege Validation and Assignment .......................................................................... 72  
2.2.1 Assigning Privileges ............................................................................................. 72  
2.2.2 Validating Privileges ............................................................................................ 73  
2.3 Summary ....................................................................................................................... 76
3 Catalog Objects

3.1 What Are SAP HANA Catalog Objects? .............................................................. 77
3.2 Creating and Managing Native Catalog Objects ............................................. 79
  3.2.1 Creating Schemas ...................................................................................... 80
  3.2.2 Creating Catalog Tables ............................................................................ 82
  3.2.3 Creating Other Catalog Objects ................................................................. 83
3.3 Creating and Managing Repository Catalog Objects ...................................... 84
  3.3.1 Creating Repository Schemas ..................................................................... 85
  3.3.2 Creating Repository Tables ....................................................................... 87
3.4 Deploying Repository Objects .......................................................................... 90
3.5 Case Study ....................................................................................................... 96
3.6 Summary ......................................................................................................... 101

4 User Accounts .................................................................................................... 103

4.1 What Are User Accounts? .............................................................................. 103
  4.1.1 Standard User Accounts .......................................................................... 104
  4.1.2 Technical User Accounts ......................................................................... 105
  4.1.3 Restricted User Accounts .......................................................................... 106
  4.1.4 LDAP User Accounts ................................................................................ 109
4.2 Creating and Managing User Accounts ........................................................... 110
  4.2.1 Creating and Managing Users with SQL Statements .............................. 110
  4.2.2 Creating and Managing Users in the SAP HANA Cockpit ....................... 112
  4.2.3 Creating and Managing Users with the SAP HANA Web-Based Development Workbench ................................................................. 114
  4.2.4 User Account System Views ..................................................................... 117
  4.2.5 Deleting User Accounts .......................................................................... 120
4.3 Granting and Revoking Privileges .................................................................. 123
  4.3.1 Granting and Revoking Privileges with SQL .......................................... 123
  4.3.2 Granting and Revoking Privileges with the SAP HANA Web-Based Development Workbench ................................................................. 131
  4.3.3 Granting and Revoking Privileges with the SAP HANA Web-Based Development Workbench ................................................................. 138
  4.3.4 Effective Privileges System View ............................................................... 143

5 Database Roles .................................................................................................... 161

5.1 What Are Roles? ............................................................................................. 161
5.2 Creating and Managing Roles ......................................................................... 165
  5.2.1 Creating and Deleting Roles with SQL Statements ............................... 165
  5.2.2 Creating and Deleting Roles with the SAP HANA Cockpit ...................... 167
  5.2.3 Creating and Deleting Roles with the SAP HANA Web-Based Development Workbench ................................................................. 168
5.3 Granting and Revoking Privileges ................................................................. 170
  5.3.1 Methodologies for Granting Privileges to Roles ..................................... 171
  5.3.2 Granting and Revoking Privileges with SQL .......................................... 173
  5.3.3 Granting and Revoking Privileges with the SAP HANA Cockpit .......... 181
  5.3.4 Granting and Revoking Privileges with the SAP HANA Web-Based Development Workbench ................................................................. 187
5.4 Managing Nested Roles .................................................................................. 192
  5.4.1 Granting and Revoking Roles with SQL ................................................ 192
  5.4.2 Granting and Revoking Roles with the SAP HANA Cockpit ................. 194
  5.4.3 Granting and Revoking Roles with the SAP HANA Web-Based Development Workbench ................................................................. 196
5.5 Mapping LDAP Groups to Roles .......................................................... 196
5.5.1 Mapping Roles with SQL .............................................................. 197
5.5.2 Mapping Roles with the SAP HANA Cockpit ............................. 197
5.6 Summary .......................................................................................... 199

6 Repository Roles .................................................................................. 201

6.1 What Are Repository Roles? .............................................................. 201
6.1.1 User Account _SYS_REPO and Repository Roles ....................... 202
6.1.2 Grantors and Privileges ............................................................... 204
6.1.3 Grantors and Roles ................................................................. 205
6.1.4 Why Use Repository Roles? ....................................................... 205
6.2 Managing Repository Roles with Design-Time Scripts .................... 207
6.2.1 Creating a Repository Package ................................................. 208
6.2.2 Creating Repository Roles within a Package .................... 210
6.2.3 Defining the Role Name Tag ........................................... 211
6.2.4 Extending Roles ................................................................. 211
6.2.5 Assigning Privileges ............................................................ 212
6.2.6 Save and Activate ............................................................... 213
6.2.7 Runtime Repository Roles .................................................... 213
6.3 Granting and Revoking Privileges in Design-Time Scripts .......... 213
6.3.1 System Privileges ................................................................. 214
6.3.2 Schema Privileges .............................................................. 215
6.3.3 Object Privileges ................................................................. 216
6.3.4 Structured Privileges .......................................................... 218
6.3.5 Remote Source Privileges .................................................... 219
6.3.6 Analytic Privileges .............................................................. 219
6.3.7 Application Privileges ......................................................... 220
6.3.8 Package Privileges .............................................................. 220
6.4 Managing Repository Roles with the SAP HANA Web-Based Development Workbench ................................. 221
6.4.1 Accessing and Navigating the SAP HANA Web-Based Development Workbench Editor .................................................. 221
6.4.2 System Privileges ................................................................. 223
6.4.3 Object Privileges ................................................................. 224
6.4.4 Analytic Privileges .............................................................. 227
8 Object Privileges 269

8.1 What Are Object Privileges? .............................................................. 269
  8.1.1 Catalog Object Privileges .............................................................. 270
  8.1.2 Security Considerations for Catalog Objects .................................. 276

8.2 Granting Object Privileges with SQL ............................................. 280
  8.2.1 Securing Schemas with SQL ......................................................... 280
  8.2.2 Securing Individual Catalog Objects with SQL ............................... 283

8.3 Granting Object Privileges with the SAP HANA Cockpit .................. 285

8.4 Granting Object Privileges with the SAP HANA Web-Based Development Workbench ........................................ 287

8.5 Granting Object Privileges with Repository Roles ............................. 289
  8.5.1 Script-Based Repository Roles ....................................................... 289
  8.5.2 SAP HANA Web-Based Development Workbench GUI ................... 292

8.6 Case Study: Updating Repository Roles to Access Information Views .... 295
  8.6.1 Consumer .................................................................................... 296
  8.6.2 Power User .................................................................................. 297
  8.6.3 Developer ................................................................................... 298

8.7 Summary ......................................................................................... 299

9 Package Privileges 301

9.1 What Is the SAP HANA Development Repository? ......................... 301
  9.1.1 Structure of the Development Repository ..................................... 302
  9.1.2 Creating Packages and Subpackages .......................................... 303
  9.1.3 Overview of Delivery Units .......................................................... 303

9.2 What Are Package Privileges? .......................................................... 304

9.3 Granting Package Privileges ............................................................. 307
  9.3.1 Granting Package Privileges with SQL ........................................... 307
  9.3.2 Granting Package Privileges with the SAP HANA Cockpit ................ 307
  9.3.3 Granting Package Privileges with the SAP HANA Web-Based Development Workbench ........................................ 309
  9.3.4 Granting Package Privileges within Repository-Based Roles ............ 311

9.4 Case Study: Preventing Content Developers from Elevating Their Privileges ................................................................. 315
  9.4.1 Assessing the Current Configuration ............................................. 315
  9.4.2 Recommendations ..................................................................... 316

9.5 Summary ......................................................................................... 319

10 Analytic Privileges 321

10.1 What Are SAP HANA Information Views? .................................... 322
  10.1.1 Attribute Views .......................................................................... 322
  10.1.2 Analytic Views .......................................................................... 323
  10.1.3 Calculation Views ...................................................................... 324

10.2 What Are Analytic Privileges? ....................................................... 324
  10.2.1 XML-Based Analytic Privileges ................................................... 325
  10.2.2 SQL-Based Analytic Privileges .................................................... 328

10.3 _SYS_BI_CP_ALL: A System-Generated Analytic Privilege ................ 330

10.4 Managing Static Analytic Privileges ............................................... 331
  10.4.1 Creating Static XML-Based Analytic Privileges ............................. 331
  10.4.2 Creating Static SQL-Based Analytic Privileges ............................. 335

10.5 Managing Dynamic Analytic Privileges ......................................... 336
  10.5.1 Dynamic XML-Based Analytic Privileges .................................... 337
  10.5.2 Dynamic SQL-Based Analytic Privileges .................................... 339

10.6 Managing Dynamic Expression-Based SQL Analytic Privileges ...... 346

10.7 Troubleshooting Effective Analytic Privileges and Filter Conditions .... 350

10.8 Granting Analytic Privileges ........................................................... 351
  10.8.1 Granting Analytic Privileges with SQL ........................................ 351
  10.8.2 Granting Analytic Privileges with the SAP HANA Cockpit ................. 352
  10.8.3 Granting Analytic Privileges with the SAP HANA Web-Based Development Workbench ........................................ 354
  10.8.4 Granting Analytic Privileges with Repository Roles ......................... 355

10.9 Summary ......................................................................................... 359
11 Application Privileges 361

11.1 What Are Application Privileges? ................................................................. 361
11.2 Creating Application Privileges ...................................................................... 362
11.3 Granting Application Privileges ...................................................................... 364
  11.3.1 Granting Application Privileges with SQL ................................................. 364
  11.3.2 Granting Application Privileges with the SAP HANA Cockpit ................. 365
  11.3.3 Granting Application Privileges with the SAP HANA Web-Based
       Development Workbench Security Manager ............................................... 367
  11.3.4 Granting Application Privileges within Repository Roles ....................... 368
11.4 Privileges on Users ....................................................................................... 372
  11.4.1 Granting Privileges on Users with the SAP HANA Cockpit .................... 373
  11.4.2 Granting Privileges on Users with SQL .................................................... 374
11.5 Summary ....................................................................................................... 375

12 Authentication 377

12.1 SAP HANA Internal Authentication Mechanism .......................................... 378
  12.1.1 Protecting SAP HANA Passwords with Encryption ............................... 379
  12.1.2 Configuring the Internal Authentication Password Policy ...................... 379
  12.1.3 Managing Password Policy Settings with SQL ...................................... 386
  12.1.4 Managing Password Policy Settings in GUIs ........................................ 387
12.2 SAP HANA and LDAP Authentication ............................................................ 393
12.3 Supported Third-Party Authentication Providers .......................................... 396
  12.3.1 Kerberos Authentication ......................................................................... 396
  12.3.2 SAML Authentication ............................................................................ 399
  12.3.3 X.509 Authentication ............................................................................ 402
  12.3.4 SAP Logon Tickets ................................................................................ 403
  12.3.5 SAP Assertion Tickets .......................................................................... 405
  12.3.6 JWT Identity Providers .......................................................................... 406
12.4 Case Study: Adding SAML Identity User Accounts ....................................... 406
12.5 Summary ....................................................................................................... 409

13 Certificate Management and Encryption 411

13.1 SSL Certificates ............................................................................................ 411
  13.1.1 In-Database Certificate Management ...................................................... 412
  13.1.2 External SAP HANA PSE File and Certificate Management ................. 419
13.2 Client Encryption Settings ............................................................................. 423
  13.2.1 SAP HANA Studio .................................................................................. 423
  13.2.2 XS Engine Web-Based Applications ....................................................... 425
  13.2.3 JDBC and ODBC Drivers ...................................................................... 427
  13.2.4 SAP HANA Cockpit .............................................................................. 430
13.3 Encrypting Data ............................................................................................ 431
  13.3.1 Server-Side Data Encryption ................................................................. 432
  13.3.2 Managing Root Keys within the SSFS ............................................... 433
  13.3.3 Encrypting the Data Volume ............................................................... 437
  13.3.4 Encrypting the Log Volume ............................................................... 438
  13.3.5 Encryption the Backup Media ............................................................ 439
13.4 Summary ....................................................................................................... 440

14 Security Lifecycle Management 441

14.1 Maintaining a Consistent Security Model .................................................... 441
  14.1.1 Best Practices ....................................................................................... 442
  14.1.2 Testing Security Model Changes ......................................................... 444
  14.1.3 Keeping Repository Roles in Sync ....................................................... 446
14.2 Creating Delivery Units for Security-Related Packages ............................... 448
  14.2.1 Creating a Delivery Unit with SAP HANA Studio ............................... 449
  14.2.2 Creating a Delivery Unit with SAP HANA Application Lifecycle
       Management ................................................................................................. 452
  14.2.3 Importing and Exporting Delivery Units with SAP HANA
       Application Lifecycle Management ............................................................. 456
14.3 Transporting Security Packages to Other SAP HANA Systems .................... 457
  14.3.1 Transporting a Delivery Unit with SAP HANA Application Lifecycle
       Management ................................................................................................. 457
  14.3.2 Exporting a Delivery Unit to a File ....................................................... 462
14.3.3 Importing a Delivery Unit from a File ............................ 464
14.4 Additional Options in SAP HANA Application Lifecycle Management ...... 466
14.4.1 Change Recording .......................................................... 467
14.4.2 Using the Change and Transport System ......................... 467
14.5 Summary .............................................................................. 468

15 Auditing .............................................................................. 469

15.1 Why Do You Need Auditing? .................................................. 469
15.2 Configuring Auditing .............................................................. 471
  15.2.1 Enable Auditing with the SAP HANA Cockpit ................. 471
  15.2.2 Audit Log Targets and Options in the SAP HANA Cockpit .... 472
  15.2.3 Viewing Audit Logs in the SAP HANA Cockpit ................. 475
  15.2.4 Enabling Auditing with the SAP HANA Web-Based Development Workbench 476
  15.2.5 Enabling Auditing with SQL ............................................. 477
15.3 Creating Audit Policies .......................................................... 479
  15.3.1 Components of the Audit Policy ....................................... 480
  15.3.2 Managing Policies with the SAP HANA Web-Based Development Workbench 486
  15.3.3 Managing Audit Policies with SQL ................................. 488
  15.3.4 Creating Policies with the SAP HANA Cockpit .................. 490
15.4 Querying Audit Data ............................................................. 494
15.5 Case Study: Defining Audit Policies ........................................ 496
  15.5.1 Proactive Event Monitoring ............................................. 496
  15.5.2 Audit Reporting ............................................................. 496
  15.5.3 Authentication Auditing .................................................. 497
  15.5.4 Unauthorized Action Auditing .......................................... 498
  15.5.5 System Change Auditing ................................................. 499
  15.5.6 Security Management Task Auditing .............................. 499
  15.5.7 Super User Event Auditing .............................................. 502
15.6 Summary .............................................................................. 503

16 Security Tracing and Troubleshooting .................................... 505

16.1 Authorization Tracing .......................................................... 505
  16.1.1 Enabling Tracing with the SAP HANA Cockpit ................. 506
  16.1.2 Enabling Tracing with SQL ............................................. 509
  16.1.3 Viewing the Trace File in the SAP HANA Cockpit .......... 511
16.2 Querying the System to Review Effective Privileges ................. 513
  16.2.1 Granted Privileges ......................................................... 513
  16.2.2 Granted Roles ............................................................... 514
  16.2.3 Accessible Views ........................................................... 515
  16.2.4 Effective Privilege Grantees ............................................ 516
  16.2.5 Effective Structured Privileges ....................................... 517
  16.2.6 Effective Privileges ....................................................... 519
  16.2.7 Effective Role Grantees .................................................. 520
  16.2.8 Effective Roles .............................................................. 521
16.3 Case Study: Identifying Deficiencies in Information View Access ........ 522
  16.3.1 Troubleshooting the Problem ................................. 522
  16.3.2 Reviewing the Results ............................................... 523
  16.3.3 Reviewing the Solution ............................................. 524
16.4 Summary .............................................................................. 524

17 Security Recommendations .................................................. 525

17.1 Password Authentication Settings ....................................... 525
  17.1.1 Standard User Password Policies .................................. 525
  17.1.2 Service Accounts .......................................................... 528
17.2 Encryption Settings ............................................................. 529
17.3 Identifying Users with Elevated Privileges .............................. 529
  17.3.1 System Privileges .......................................................... 530
  17.3.2 Root Package Privileges ................................................ 532
  17.3.3 Bypass Analytic Privileges ............................................ 534
  17.3.4 Default Standard Roles ................................................. 536
  17.3.5 WITH GRANT or WITH ADMIN .................................... 537
  17.3.6 Trace, Dump File, and Debug Access ............................ 538
Index

_SYS_BIC schema .............................................. 535
_SYS_BIC schema .............................................. 535
_SYS_REPO ...... 77, 85, 105, 173, 202, 204, 244, 278, 280, 291, 530
_SYS_REPO repository ................................. 545
_SYS_REPO schema ............................................. 352
hdbrule file .................................................... 576
hdbrule file .................................................... 576
hdbrule repository artifact .................................. 289
hdbschema file .................................................. 86

A

ABAP Workbench ............................................. 467
Accessible views ............................................. 515
Active Directory ............................................. 394, 397
Adapter framework ........................................... 246
Alert levels ......................................................... 474
Analytic privileges ......................................... 69, 136, 141–142, 185, 190–191, 218–219, 227–228, 244, 321, 358, 483, 501, 570
bypassing .......................................................... 534
configure ......................................................... 345
create ............................................................... 331
default ............................................................. 330
filters ................................................................. 326
grant ................................................................. 351
management window ........................................... 332
overview ........................................................... 324
repository roles ............................................... 355
restrictions ......................................................... 346
revoke ......................................................... 186, 353
SAP HANA Web-Based Development
Workbench ......................................................... 354
SQL ................................................................. 354
_SYS_BIC_CP_ALL ........................................... 361, 372
_Application identifier ........................................ 551
_Application privilege objects ........ 72, 130, 180, 220, 230, 361, 372
add ................................................................. 371
creation ........................................................... 362
creation guide .................................................... 364
grant ................................................................. 364, 366
remove ............................................................. 372
revoke ......................................................... 186, 353
SAP HANA Web-Based Development
Workbench ......................................................... 367
SQL ................................................................. 364
subpackage node ................................................ 364
within repository roles ...................................... 368
Application programming interfaces
(APIs) ................................................................. 427
Application roles ............................................... 553
Application-access file ...................................... 363
Application-privileges file .................................... 362
Audit policies ....................................................... 479
Audit trail target .................................................. 485
components ..................................................... 480
creation ........................................................... 488, 490
define new ......................................................... 487, 490
management ..................................................... 486, 488
mandatory ......................................................... 479

597
<table>
<thead>
<tr>
<th>Index</th>
<th>598</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audit policies (Cont.)</td>
<td>490</td>
</tr>
<tr>
<td>SQL</td>
<td>489</td>
</tr>
<tr>
<td>target objects</td>
<td>485, 487</td>
</tr>
<tr>
<td>target user</td>
<td>485</td>
</tr>
<tr>
<td>view</td>
<td>495</td>
</tr>
<tr>
<td>Audit reporting</td>
<td>496</td>
</tr>
<tr>
<td>Audit rule</td>
<td>474</td>
</tr>
<tr>
<td>Audit trail</td>
<td>471, 478</td>
</tr>
<tr>
<td>Auditing</td>
<td>60, 243, 448, 469, 506</td>
</tr>
<tr>
<td>action status</td>
<td>484</td>
</tr>
<tr>
<td>all actions</td>
<td>480</td>
</tr>
<tr>
<td>backup deletion</td>
<td>481</td>
</tr>
<tr>
<td>benefits</td>
<td>469</td>
</tr>
<tr>
<td>case study</td>
<td>496</td>
</tr>
<tr>
<td>certificate and PSE store</td>
<td>481</td>
</tr>
<tr>
<td>configuration</td>
<td>471, 481</td>
</tr>
<tr>
<td>data</td>
<td>494</td>
</tr>
<tr>
<td>data definition</td>
<td>481</td>
</tr>
<tr>
<td>data provisioning</td>
<td>481</td>
</tr>
<tr>
<td>data query and manipulation</td>
<td>482</td>
</tr>
<tr>
<td>data volume encryption</td>
<td>482</td>
</tr>
<tr>
<td>enable</td>
<td>471</td>
</tr>
<tr>
<td>granting and revoking authorizations</td>
<td>482</td>
</tr>
<tr>
<td>level target trails</td>
<td>474</td>
</tr>
<tr>
<td>license deletion</td>
<td>482</td>
</tr>
<tr>
<td>license installation</td>
<td>482</td>
</tr>
<tr>
<td>monitoring</td>
<td>482</td>
</tr>
<tr>
<td>regulatory compliance</td>
<td>470</td>
</tr>
<tr>
<td>repository content operations</td>
<td>483</td>
</tr>
<tr>
<td>SAP HANA cockpit</td>
<td>471</td>
</tr>
<tr>
<td>SAP HANA dynamic tiering</td>
<td>483</td>
</tr>
<tr>
<td>SAP HANA Web-Based Development</td>
<td>476</td>
</tr>
<tr>
<td>Workbench</td>
<td>476</td>
</tr>
<tr>
<td>session management and system configuration</td>
<td>483</td>
</tr>
<tr>
<td>SQL</td>
<td>477</td>
</tr>
<tr>
<td>structured privileges management</td>
<td>483</td>
</tr>
<tr>
<td>targets</td>
<td>472</td>
</tr>
<tr>
<td>unauthorized actions</td>
<td>498</td>
</tr>
<tr>
<td>user account management</td>
<td>489</td>
</tr>
<tr>
<td>user and role management</td>
<td>484</td>
</tr>
<tr>
<td>Authentication</td>
<td>57, 103, 114, 377, 412, 420</td>
</tr>
<tr>
<td>auditing</td>
<td>497</td>
</tr>
<tr>
<td>case study</td>
<td>406</td>
</tr>
<tr>
<td>Authentication (Cont.)</td>
<td>481</td>
</tr>
<tr>
<td>Authorization component</td>
<td>505</td>
</tr>
<tr>
<td>Authorization trace</td>
<td>505, 508</td>
</tr>
<tr>
<td>configuration</td>
<td>509</td>
</tr>
<tr>
<td>Authorization validation</td>
<td>75</td>
</tr>
<tr>
<td>Authorizations</td>
<td>103, 377</td>
</tr>
<tr>
<td>catalog object-level</td>
<td>278</td>
</tr>
<tr>
<td>grantable to others</td>
<td>277</td>
</tr>
<tr>
<td>local v. LDAP</td>
<td>113</td>
</tr>
<tr>
<td>schema-level</td>
<td>278</td>
</tr>
</tbody>
</table>

| B                                                                     | 247 |
| Backup admin system privilege                                        | 247 |
| Basis administrator                                                  | 82 |
| Business Function Library (BFL)                                      | 29 |

| C                                                                     | 251 |
| Calculation scenarios                                                | 69, 324, 585 |
| Calculation views                                                    | 120 |
| CASCADE option                                                       | 96, 150, 234, 262, 295, 315, 406, 496, 522 |
| Case studies                                                         | 68 |
| Catalog object access                                                | 171 |
| Catalog object privileges                                            | 68, 126, 176, 217, 270, 295–296 |
| Catalog object types                                                 | 62, 559 |
| Catalog objects                                                      | 67–68, 77, 89, 184, 188, 269, 278, 291, 517, 563 |
| case study                                                           | 96 |
| create and maintain                                                  | 79 |
| creation                                                             | 83 |
| runtime                                                              | 290 |
| security                                                             | 289 |
| security                                                             | 276 |
| Catalog read system privilege                                        | 252 |
| catalog schema privileges                                            | 124, 174 |
| Catalog schemas                                                      | 81–82, 271, 290 |
| privileges                                                           | 271 |
| Catalog sequence                                                     | 276 |
| Catalog stored procedures                                           | 274, 279 |
| privileges                                                           | 275 |
| Catalog table objects                                                | 568 |

| Catalog tables                                                      | 272, 284, 348 |
| creation                                                            | 82 |
| grant/revoke privileges                                             | 283 |
| privileges                                                           | 273 |
| Catalog views                                                       | 83, 178, 273, 278 |
| privileges                                                           | 274 |
| Certificate admin                                                    | 244 |
| Certificate collection                                              | 58, 412, 415 |
| Certificate keys                                                    | 420 |
| Certificate management                                              | 411 |
| external                                                             | 419 |
| in-database                                                         | 412 |
| Certificate store                                                   | 413 |
| Certificates                                                        | 58, 412 |
| collection purpose                                                  | 419 |
| collections                                                         | 415 |
| configuration                                                       | 423 |
| ID                                                                   | 414 |
| import                                                               | 414 |
| validate                                                             | 431 |
| Change and Transport System (CTS)                                   | 467 |
| Change recording                                                     | 467 |
| Change tracking                                                      | 467 |
| Child objects                                                        | 68 |
| referenced                                                           | 74 |
| Client certificate                                                   | 403 |
| Client-side encryption                                              | 481 |
| Cloud Foundry                                                       | 543 |
| Column store table                                                  | 83, 89 |
| Compile server                                                       | 34 |
| Consumer role                                                        | 296 |
| Content vendor                                                       | 91 |
| change                                                               | 92 |
| create                                                               | 92 |
| CONTENT_ADMIN role                                                  | 536 |
| Core data services (CDS)                                            | 88, 272, 556 |
| document table                                                       | 100 |
| entity                                                               | 88 |
| Client                                                              | 29 |
| Create scenario system privilege                                    | 89 |
| Create schema                                                       | 66 |
| CREATE statement                                                    | 202 |
| Create users                                                        | 157 |
| CSV file                                                            | 473, 478 |
| CSV file import                                                     | 153 |

| D                                                                     | 248, 530 |
| Data admin system privilege                                          | 248, 530 |
| role                                                                 | 266 |
| Data Definition Language (DDL)                                       | 248, 270, 530 |
| Data encryption                                                      | 56, 435 |
| Data Manipulation Language (DML)                                     | 249, 270, 340 |
| Data provisioning agent framework                                    | 247 |
| Data source name (DSN)                                               | 428 |
| Data volume encryption                                               | 457, 482 |
| SAP HANA Studio                                                      | 438 |
| SQL                                                                  | 437 |
| Database accounts                                                    | 541 |
| Database forensics                                                   | 471 |
| Database object dependencies                                         | 56 |
| Database objects                                                     | 67, 77, 206 |
| Database roles                                                        | 161 |
| consistency                                                          | 164 |
| Change recording                                                     | 156 |
| standard                                                             | 162 |
| Database schemas                                                     | 560 |
| Database store                                                       | 413 |
| Database users                                                       | 561 |
| Database views                                                       | 70 |
| DBA_COCKPIT role                                                     | 163 |
| Default roles                                                        | 536 |
| Definer mode                                                         | 75, 279 |
| Delivery units                                                       | 91–92, 206, 242, 301, 443, 454 |
| actions                                                              | 93 |
| assign packages                                                       | 94, 455 |
| create                                                               | 92–93, 454 |
| creation                                                             | 448 |
| export options                                                       | 65, 464 |
| export                                                                | 463 |
| export to file                                                       | 462 |
| import                                                                | 95 |
| import from file                                                     | 464 |
| import status                                                        | 462 |
| import/export                                                        | 456 |
| management window                                                    | 450 |
| overview                                                             | 303 |
| packages                                                             | 451 |
| privileges                                                           | 304 |
| remove packages                                                      | 456 |
This sample chapter provides an overview of the role creation and deletion process and then reviews several options for managing privilege assignments to roles. It also covers how roles can be assigned to other roles.

"Database Roles"

Contents

Index

The Author

Jonathan Haun

SAP HANA 2.0 Security Guide
608 Pages, 2020, $89.95

www.sap-press.com/4982
Jonathan Haun is a senior director at Enowa LLC where he is responsible for managing the business intelligence and technology practices. Over the past several years, he has helped several clients implement solutions using SAP HANA. In addition to being certified in multiple SAP BusinessObjects BI tools, he also holds the certifications of SAP Certified Application Associate—SAP HANA 1.0, SAP Certified Technology Associate—SAP HANA 1.0, and SAP Certified Technology Specialist—SAP HANA Installations.

Jonathan has worked in the field of business intelligence and database administration for more than 19 years. During this time, he has gained invaluable experience helping customers implement solutions using the tools from the SAP BusinessObjects BI and SAP HANA product lines. Before working as a full-time SAP consultant, he worked in a variety of information technology management and administrative roles. His combination of experience and wealth of technical knowledge make him an ideal source of information pertaining to SAP BusinessObjects BI and SAP HANA.

We hope you have enjoyed this reading sample. You may recommend or pass it on to others, but only in its entirety, including all pages. This reading sample and all its parts are protected by copyright law. All usage and exploitation rights are reserved by the author and the publisher.