Alibaba Cloud MaxCompute

Management

Issue: 20190906

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. **You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.**
2. **No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.**
3. **The content of this document may be changed due to product version upgrades , adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.**
4. **This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults " and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity , applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use**

or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified , reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates . The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).
6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	 Note: You can use Ctrl + A to select all files.
>	Multi-level menu cascade.	Settings > Network > Set network type
Bold	It is used for buttons, menus , page names, and other UI elements.	Click OK.
`Courier font`	It is used for commands.	Run the `cd /d C:/windows` command to enter the Windows system folder.
Italics	It is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` *[-all|-t]*

Style	Description	Example
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich {stand | slave}`

Contents

1 Security management

1.1 Security model

This topic describes the security model of MaxCompute and that of DataWorks. The security model of MaxCompute can be used by MaxCompute project owners and security administrators for better overall O&M and regular security operations. To ensure better data security, we recommend that you read about the security model before you configure any security functions on Alibaba Cloud.

A security model can be configured for MaxC ompute and DataWorks. When you interwork MaxCompute with DataWorks but the security model of DataWorks does not meet your service security requirements, you need to use the security models of both MaxCompute and DataWorks combined together.

MaxCompute security model

Benefits

MaxCompute supports multi-tenant data security, which has the following benefits:

- User authentication

  MaxCompute supports two account systems: the Alibaba Cloud account system and RAM user system. Note that MaxCompute recognizes RAM users but cannot recognize RAM permissions. That is, you can add RAM users under your Alibaba Cloud account to a MaxCompute project. However, MaxCompute does not consider the RAM permission definitions when it verifies the permissions of RAM users.
- User management

  User management operations such as adding and removing users and granting permissions to users are supported for MaxCompute projects. You can manage permissions by using roles. For each project, an admin role is provided

automatically. Next, you can grant permissions by using access control lists (ACLs) or by setting policies.

ACLs are similar to the `GRANT` and `REVOKE` statements defined in SQL-92. You can use simple statements to grant or revoke permissions for objects in your workspace. An example is as follows:

```
grant actions on object to subject ;
```

- LabelSecurity

  `LabelSecurity` is a workspace-level mandatory access control (MAC) policy that enables workspace administrators to control user access to column-level sensitive data more flexibly.

- Resource sharing across projects based on package

  You can share data and resources, such as tables and functions, among workspaces by using packages. For these operations, you only need to manage the users in your project.

- Data protection of projects

  Multi-tenant data security meets customer requirements on not allowing user data to be transmitted outside workspaces.

Permissions, roles, and labels

The security system provided by MaxCompute includes a variety of policies. Permissions are granted by the application of different policies, and help maintain fine-grained authorization. The following describes an example of how to grant the permission on an L4 table to a user to illustrate how permissions are granted by the use of policies:

1. If no permissions have been granted to the user and the user does not belong to the project, add the user to the project. The user does not have any permissions before they are added to the project.
2. Grant operation permissions to the user. For details, see Authorization.
   a. Grant a specific operation permission to the user.
   b. Grant an ACL to a role and then to the user. If a resource does not have a label, the user has obtained the permission on the resource.

3. If the user manages resources that have labels, such as datasheets and packages with datasheets, grant label permissions to the user. Four types of label permissions are provided:

   a. Permissions on fields in a datasheet
   b. Permissions on a datasheet (This type of permission is not supported currently.)
   c. Permissions on a package
   d. Permissions on a user (Label permissions cannot be granted to a role.)

The following figure shows how permissions are granted by means of fine-grained authorization and access control.

DataProtection and packages

`DataProtection` prevents data from leaking from a project. After `DataProtection` is enabled, data can be exchanged only between projects that are in the same trusted project group. If two projects are not in the same trusted project group, you need to grant permissions on resources in one project to users in the other project by using a package. For more information, see Data protection of projects.

You can group some resources, such as commonly used tables and user-defined functions (UDFs), into a package, and then grant the permissions on this package to another project.

In some scenarios, `ProjectProtection` allows you to configure exception policies specific to application IP addresses and Alibaba Cloud accounts, so that data can be exchanged if needed.

DataWorks security model

DataWorks supports the access of multiple users to shared data sources to help develop data analytics applications. Its security model ensures the following requirements:

- Isolation of data among organizations.
- Security of data development during extract, transform, and load (ETL) processes. Specifically, it helps limit changes to production tasks, manage which members can edit and debug code, and manage which members can publish production tasks.

- Permissions can be granted on MaxCompute resources (such as tables, functions, and instances) even though MaxCompute provides its own security model that does not include the permissions for such processes as ETL.

Authentication and interoperation with RAM is supported. Specially, you can use your Alibaba Cloud account to create and activate a DataWorks project, and then authorize RAM users under your Alibaba Cloud account the permissions to operate resources in DataWorks.

Using the same account to create all your projects could comprise an organization. To avoid doing so, you can configure dependencies among tasks from different projects. The data of various tasks from projects created by using different accounts is isolated.

To ensure better security, DataWorks distinguishes between development projects and production projects by services to isolate task development and debug from stable production. You can use roles to specify which members can develop and debug tasks and which members can operate and maintain production tasks.

For permissions on MaxCompute resources, while a MaxCompute project is created , roles are created in the project based on roles in DataWorks and permissions are granted to these roles in the project.

1.2 MaxCompute permissions and DataWorks permissions

This topic describes how permissions can be authorized when MaxCompute interoperates with DataWorks and the limitations of using the permissions of only one service. When you use the security model of MaxCompute to control permissions, project members can perform authorized operations on any interfaces in DataWorks. However, when you use DataWorks to assign roles to users, the permissions of project members on MaxCompute resources are more limited.

Project relationship

If you log on to the console from the MaxCompute or DataWorks official website, you can create one of the following two types of projects:

- Projects in simple mode: In simple mode, a DataWorks workspace is associated with a MaxCompute project. A number of roles are created in the MaxCompute project. For details about the role permissions, see Member roles and permissions in this topic.

- Projects in standard mode: In standard mode, a DataWorks workspace is associated with a MaxCompute development project and a MaxCompute production project. A number of roles are created in each MaxCompute project. For details about the role permissions, see Member roles and permissions in this topic.

Account authentication

In a DataWorks project, an Alibaba Cloud account must be the owner of the project. That is, a RAM user account cannot be the owner. In a MaxCompute project, an Alibaba Cloud account can be the owner or a user. When you add members by using the project member management system of DataWorks, you can add only the RAM users under your Alibaba Cloud account. In MaxCompute, however, you can add other Alibaba Cloud accounts by running the `add user xxx;` command on the command line interface (CLI).

Member roles and permissions

Project members require permissions on MaxCompute resources during extract, transform, and load (ETL) operations. Therefore, the roles for DataWorks projects are also created for MaxCompute projects. For more information, see User management. In addition to the project owner role, the admin role is also provided by MaxCompute. The following table describes the interoperation between MaxCompute role permissions and DataWorks role permissions.

MaxCompute role	MaxCompute permission	DataWorks role	DataWorks permission
project owner	This role has permissions to operate on all MaxCompute projects.	None	None

MaxCompute role	MaxCompute permission	DataWorks role	DataWorks permission
admin	When you create a project, the system creates an admin role for it and grant the following permissions to the role: accessing all objects in the project, managing users or roles, and granting permissions to users or roles. Unlike a project owner, an admin role cannot grant admin role permissions to users, set security policies for workspaces, or change the authentication models of workspaces. The permissions of an admin role cannot be changed. The project owner role can assign an admin role to a user so that the user is authorized with security management.	None	None

MaxCompute role	MaxCompute permission	DataWorks role	DataWorks permission
role_project_admin	This role has all permissions on projects, tables, functions, resources, instances, jobs, and packages.	Administrator	This role is the administrator of a workspace. It can manage the basic properties, data sources, compute engine configurations, and project members in the workspace. It also can assign `administrator`, `development`, `O&M`, `deploy`, and `visitor` roles to project members.
role_project_dev	This role has permissions to operate on projects, functions, resources, instances, jobs, packages, and tables.	Development	A user with this role can create workflows, script files, resources, and user-defined functions (UDFs), create or delete tables, and create packages. However, this role does not have the permission to publish.
role_project_pe	This role has permissions to operate on projects, functions, resources, instances, and jobs. It also has read permissions for packages and also read and describe permissions for tables.	O&M	A user with this role has publish and online O&M permissions, which are granted by the project administrator. This role does not have the permission to develop data.
role_project_deploy	This role does not have any permissions by default.	Deploy	This role is similar to the `O&M` role, except that a user with the `deploy` role does not have the online O&M permission.
role_project_guest	This role does not have any permissions by default.	Visitor	A user with this role can only view data, but cannot edit workflows or code.

MaxCompute role	MaxCompute permission	DataWorks role	DataWorks permission
role_Project_security	This role does not have any permissions by default.	Security Administrator	This role is only used to configure sensitivity rules and audit data risks in Data Security Guard.

Note:

According to the preceding table, the mapping between DataWorks roles and MaxCompute permissions is fixed. After a user is assigned a DataWorks role and obtains the permissions of the MaxCompute role that is associated with this DataWorks role, if you assign other MaxCompute permissions to the user on the CLI, the user's permissions in MaxCompute become inconsistent with those in DataWorks.

Users and permissions

After a DataWorks workspace is associated with one MaxCompute project, you can specify whether members of another DataWorks workspace have permissions to operate on this MaxCompute project. Specifically, you can choose Workspace Manage from the main menu. On the page that is displayed, you can then choose `Workspace Management` from the left pane. Last, on the Settings page that is displayed, set the `AccessKey ID` parameter.

`AccessKey ID` has two values: `Personal Account` and `Compute Engine Designated Account`. The following figure shows the interoperation between users and permissions.

In standard mode, a DataWorks workspace is associated with a MaxCompute development project and a MaxCompute production project. Members of other DataWorks workspaces can be granted the permissions of the roles assigned to this MaxCompute development project. However, they cannot be granted the permissions of the roles assigned to this MaxCompute production project. To execute a MaxCompute task, you need to publish it to the production project, and then submit it to MaxCompute as the owner.

1.3 Manage users, roles, and permissions

This topic describes how to manage users, roles, and permissions for MaxCompute and DataWorks. We recommend that you read this topic before you configure User management.

Manage users

You can manage users by either adding users or by deleting or locking accounts that do not have owners, remain inactive, or that are owned by employees that have resigned. Furthermore, you can control the permissions of the `Project Manager` and `O & M` roles in DataWorks.

Note:
If you create an account in DataWorks, a default role in MaxCompute is assigned to this account.

Item	MaxCompute	DataWorks
Role	`project owner` or `admin`	`Project Manager`
View details	1. Run the `list users ;` command to view the users in the current project. 2. Run the `show grants for < username >;` command to view the permissions of a specified user.	Log on to the DataWorks console, and navigate to the management page for a workspace. In the left-side navigation pane, choose User Management. Then, view the members and roles in the workspace, and check the validity of the permissions of each member.

Item	MaxCompute	DataWorks
Grant permissions	Run the `add user < username >;` command to add a specified user to the current project. MaxCompute project members can be operated only in combination with object, role, and label permissions. You need to check whether MaxCompute project members have object, role, or label permissions and need to delete these permissions when required. Note: MaxCompute project members are not associated with DataWorks. DataWorks allows you only to add Alibaba Cloud accounts and RAM user accounts.	Log on to the DataWorks console, and navigate to the management page for a workspace. In the left-side navigation pane, choose User Management. Then, add members, and assign roles to the members. · You can add only a RAM user account under the owner of a workspace to be a member for this workspace. · After you add a member and assign a role to this member, this member is granted the default permissions of this role. For more information, see MaxCompute permissions and DataWorks permissions.

Item	MaxCompute	DataWorks	
Roll back settings	Run the `remove user < username >;` command to delete a specified user from the current project.	Delete the permissions of members or roles from DataWorks. After doing so, the system deletes the corresponding users and roles from MaxCompute.	

Manage roles

Roles are managed differently in MaxCompute and DataWorks. In MaxCompute, you can manage roles by creating roles, by granting permissions to these roles, and by deleting necessary accounts, resources, or permissions from roles. In DataWorks, you can assign roles by either changing the roles of members or controlling the assignment of the `Project Manager` and `O & M` roles.

Note:

After a MaxCompute project is created, it is assigned with both the default `admin` role in MaxCompute and another role in DataWorks. For more information, see MaxCompute permissions and DataWorks permissions.

	MaxCompute	DataWorks role
Role	`project owner` or `admin`	`Project Manager`

	MaxCompute	DataWorks role
View details	Run the `list roles ;` command to view all the roles in the current project. Run the `describe role <role_name>;` command to view the permissions of a specified role. Run the `show grants for <username>;` command to view the role assigned to a specified user. Note: The users to whom a specified role is assigned are invisible.	Log on to the DataWorks console, and navigate to the management page for a workspace. In the left-side navigation pane, choose User Management. Then, view the members to which each role is assigned.

	MaxCompute	DataWorks role
Grant permissions	In addition to the default roles provided by MaxCompute, you can define custom role permissions and assign the custom roles to users. The overall process is as follows: 1. Run the `create role <role_name>;` command to create a role. 2. Run the `grant actions on object to <role_name>;` command to grant permissions to a role. 3. Run the `GRANT <roleName> TO <full_usern ame>;` command to assign a role to a user. Log on to the DataWorks console, and navigate to the management page for a workspace. In the left-side navigation pane, choose MaxCompute Management. In the middle pane, choose Custom User Roles, Then, define roles, and assign the roles to members.  Note: The roles that are created by using commands are not displayed.	For DataWorks, the roles that are created by running commands are not displayed. Instead, you need to log on to the DataWorks console, and navigate to the management page for a workspace. Then, in the left-side navigation pane, choose MaxCompute Management. In the middle pane, choose Custom User Roles, define roles, and assign the roles to members.

	MaxCompute	DataWorks role
Roll back settings	1. Run the `REVOKE < roleName > FROM < full_usern ame >;` command to delete a user from a role. 2. Run the `revoke < privList > on < objType > < objName > from role < rolename >;` command to revoke the permissions that are granted to a role. 3. Run the `DROP ROLE < roleName >;` command to delete a role.	In DataWorks, roles cannot be deleted. You can only delete a member from a role.

Grant permissions by using ACLs

You can revoke unnecessary object permissions, which involve various objects and operation types that must be verified before the permissions on them are revoked.

Item	Description
Role	`project owner` or `admin`
View details	· Run the `show grants for < username >;` command to view the permissions of a specified user. · Run the `show grants ;` command to view the permissions of the current user. · Run the `show acl for < objectName > [ on type < objectType >];` command to view the permissions to operate a specified object. · Run the `show acl for alipaydw . alipaydw_f or_alisec_ app on type package ;` command to view the permissions that are included in a specified package.

Item	Description
Grant permissions	Run the `grant actions on object to subject ;` command to grant the permissions to operate a specified object. Subject, object, and action types are expressed as follows: · Subject types: `user full_usern ame , role role_name` · Object types: `project project_na me , table schema_nam e , instance inst_name , function func_name , resource res_name` · Action types: `action_ite m1 , action_ite m2 , ...` For more information about subjects, objects, and actions, see Authorization.
Roll back settings	Run the `revoke actions on object from subject ;` command to revoke the permissions to operate a specified object.

Grant permissions by using packages

You can grant the permissions to operate the projects that have ProjectProtection enabled but are not in the same trusted project group. The packages for these projects and the permissions included in these packages must be properly planned. No packages can remain inactive.

Item	Description
Role	`project owner`

Item	Description
View details	· View the packages for the current project and the permissions included in these packages. - To view the packages that are created and installed in the project, run the `show packages ;` command. - To view details about the packages, run the `describe package < pkgname >;` command. · Run the `show acl for < project_name . package_name >` `on type package ;` command to view which packages are granted to users in the current project.

Item	Description
Grant permissions	As a user who creates packages, you can: · Run the `create package <pkgname>;` command to create a package. · Run the `add project_object to package package_name [ with privileges privileges ];` command to add shared resources to a package. `project_object` is expressed as follows: `table table_name , instance inst_name , function func_name , resource res_name .` · Run the `allow project <prjname> to install package <pkgname> [ using label <number>];` command to grant the package for one project to another project. As a user who installs packages, you can: · Run the `install package <pkgname>;` command to install a package. · Grant a package to a user or role. When you do so, you are not allowed to specify a label, whereas a user with the `project owner` or `admin` role can. - To grant a package to a user, run the `grant actions on package <pkgName> to user <username>;` command. - To grant a package to a role, run the `grant actions on package <pkgName> to role <role_name>;` command. For more information about action types, see Authorization. In most cases, to enable a user to access the resources in a package, you only need to grant the read permission for the package to the user.

Item	Description
Roll back settings	· Run the `disallow project < prjname > to install package < pkgname >;` command to revoke the permissions that enable a specified project to operate a package. · Run the `delete package < pkgname >;` command to delete a package. · Run the `remove project_ob ject from package package_na me ;` command to remove shared resources from a package. `project_ob ject` is expressed as follows: `table table_name , instance inst_name , function func_name , resource res_name .` · Revoke a package for a user or role: - To revoke a package for a user, run the `revoke actions on package < pkgName > from user < username >;` command. - To revoke a package for a role, run the `revoke actions on package < pkgName > from role < role_name >;` command.

Grant permissions by using labels

You can grant labels for fields, tables, and packages, which fall into one to four levels of labels, in MaxCompute.

Item	Description
Role	`project owner`

Item	Description
View details	· Run the `SHOW LABEL [< level >] GRANTS [ FOR USER < username >];` command to check which sensitive data sets are accessible to a user. - If `[FOR USER <username>]` is not specified, the sensitive data sets accessible to the current user are displayed. - If `<level>` is not specified, labels of all levels that are granted to the specified user are displayed. - If `<level>` is specified, only the labels of the specified level that are granted to the specified user are displayed. · Run the `SHOW LABEL [< level >] GRANTS ON TABLE < tablename >;` command to check which users can access a specified table that contains sensitive data. · Run the `SHOW LABEL [< level >] GRANTS ON TABLE < tablename > FOR USER < username >;` command to view all the column-level labels that a user has on a data table. For more information, see Column-level access control.

Item	Description

Grant permissions	· Run the following command to grant a label for a table or field to a user:
	`GRANT LABEL < number > ON TABLE < tablename >[( column_lis t )] TO USER < username > [ WITH EXP < days >];`
	The default value of `days` in `[ WITH EXP < days >]` is `180 .`
	For example, if you run the `GRANT LABEL 2 ON TABLE t1 TO USER alice WITH EXP 1 ;` command, the user named `alice` is explicitly granted the permissions to access data, whose sensitivity levels are 2 or lower, in the `t1` table. Furthermore, the permissions remain valid for one day.
	· Run the following command to grant a label for a project to a user:
	`SET LABEL < number > TO USER < username >;`
	· Run the following command as the creator of a package to grant the permissions for sensitive resources in the package to a user who installs the package:
	`ALLOW PROJECT < prjName > TO INSTALL PACKAGE < pkgName > [ USING LABEL < number >];`
	· Grant a package to a user or role. When you do so, you are not allowed to specify a label.
	- To grant a package to a user, run the `grant actions on package < pkgName > to user < username >;` command.
	- To grant a package to a role, run the `grant actions on package < pkgName > to role < role_name >;` command.

Item	Description
Roll back settings	· Revoke the label for a table or field from a user. - To revoke permissions, run the `REVOKE LABEL ON TABLE < tablename >[( column_lis t )] FROM USER < username >;` command. - To delete expired permissions, run the `CLEAR EXPIRED GRANTS ;` command. For example, to revoke the permissions that enable the user named `alice` to access sensitive data in the `t1` table, run the `REVOKE LABEL ON TABLE t1 FROM USER alice ;` command. · Run the `SET LABEL < number > TO USER < username >;` command to change the level of label that a user has for a project. The default level of label is 0. · Run the following command to change the level of label that enables a user, who installs a package, to access the sensitive resources in the package: `ALLOW PROJECT < prjName > TO INSTALL PACKAGE < pkgName > [ USING LABEL < number >];` The default level of label is 0. · Revoke the permissions of a user or role. - To revoke the permissions of a user, run the `revoke actions on package < pkgName > from user < username >;` command. - To revoke the permissions of a role, run the `revoke actions on package < pkgName > from role < role_name >;` command.

1.4 Configure security functions

This topic describes how to enable, set, and disable specific security functions of MaxCompute and DataWorks for improved security purposes. For more information,

see the documents Security configurations, Data protection of projects, and Column-level access control.

Enable ProjectProtection

ProjectProtection helps to limit the transmission of data from workspaces. It does this by prohibiting data from being downloaded in batches to personal computers. We recommend that you enable this function. It is disabled by default. For more information, see Security configurations.

Item	Description
Role	`project owner`
View the function status	Run the `show SecurityConfiguration;` command to check whether `ProjectProtection` is set to `true`.

Item	Description
Set the function	Use one of the following two methods to enable ProjectProtection: · Log on to the DataWorks console, navigate to MaxCompute Management, and enable `Protect workspace data` in Basic Settings. · On the command line interface (CLI) of MaxCompute, run the `SET ProjectProtection = true [WITH EXCEPTION <policyFile>];` command. If some Alibaba Cloud accounts or private accounts require the permissions to transmit data from workspaces after ProjectProtection is enabled, you can set exception policies (namely, enable the whitelist function). We recommend that you configure exception policies if: · You want to specify the number of Alibaba Cloud accounts that can transmit data from workspaces or a number of IP addresses from which data can be transmitted. · You want to specify the number of tables that private accounts can download. Add trusted projects: If you add project A as a trusted project of project B, data can be exchanged between them. · To view all the trusted projects of the current project, run the `list trustedprojects;` command. · To add a trusted project to the current project, run the `add trustedproject <projectname>;` command. · To remove a trusted project from the current project, run the `remove trustedproject <projectname>;` command. Note: If project C requires data from project D but it is not a trusted project of project D, you need to grant permissions to project C by using a package. For more information, see Package

Item	Description
Roll back settings	To disable ProjectProtection for the current project, run the `SET ProjectPro tection = false ;` command. To remove a trusted project from the current project, run the `remove trustedpro ject < projectnam e >;` command.

Enable LabelSecurity

LabelSecurity is a type of mandatory access control (MAC) for workspaces. It helps workspace owners to manage user access to column-level security-sensitive data more flexibly. This can allow you to keep fields in your tables more secure. We recommend that you enable LabelSecurity, which is disabled by default. For more information, see Column-level access control.

Item	Description
Role	`project owner`
View the function status	Run the `show SecurityCo nfiguratio n ;` command to check whether `LabelSecur ity` is set to `true .`
Set the function	Run the `Set LabelSecur ity = true ;` command to enable LabelSecurity.
Roll back settings	Run the `Set LabelSecur ity = false ;` command to disable LabelSecurity. Before you disable LabelSecurity for a project, check whether any other projects depend on it and whether the labels for tables in this project are also granted to the other projects.

Set labels for fields

We recommend that you set labels for tables, which may be divided into different levels of labels according to data sensitivity in MaxCompute.

Item	Description
View the function status	Use one of the following two methods to view the labels for fields in a MaxCompute table: · Run the `DESCRIBE < tablename >;` command. · Log on to the DataWorks console, navigate to Data Management, and view details about fields in the table.

Item	Description
Set the function	Use one of the following two methods to set labels for fields: · Method 1 (recommended) Log on to the DataWorks console, and navigate to Data Management. You can set labels for fields in new and existing tables. Note: The label attributes are visible in Data Management only when `LabelSecurity` is set to `true`. · Method 2 Run the `SET LABEL <number> TO TABLE tablename [( column_list )];` command. Examples: - To set the label for the *t1* table to `1`, run the `SET LABEL 1 TO TABLE t1;` command. - To set the labels for the `mobile` and `addr` columns in the *t1* table to `2`, run the `SET LABEL 2 TO TABLE t1 ( mobile , addr );` command. - To set the label for the *t1* table to `3`, run the `SET LABEL 3 TO TABLE t1;` command. After you do so, the labels for the `mobile` and `addr` columns remain `2`. Note: After you enable automatic labels by using the command line interface (CLI), the labels of fields in Data Management cannot be updated according to the automatically allocated labels. Therefore, we recommend that you set labels for fields by using DataWorks.

Item	Description
Roll back settings	Return fields to their original labels. Note: When you reconfigure labels for fields to make the fields more secure, the original permissions owned by packages, production accounts, and private accounts are no longer valid. To mitigate the impact, you must notify the involved users before the reconfiguration.

Enable IP address whitelist

We recommend that you enable an IP address whitelist to specify the IP addresses from which users can access a specified project. Each of these IP addresses correspond to pages in the console or SDKs. For more information, see Console and SDK.

Note:

- The whitelist takes effect on all the accounts (including `project owner`) of the project.
- The whitelist does not take effect on the servers that run DataWorks. If your server runs DataWorks, you can submit MaxCompute tasks by using DataWorks even though the IP address of your server is not included in the whitelist.

Item	Description
Role	`project owner`
View the function status	Run the `setproject;` command on the DataWorks console. In the command output, check whether information follows `odps.security.ip.whitelist=`. If no information follows `odps.security.ip.whitelist=`, the whitelist is disabled.

Item	Description
Set the function	Note: Before you enable the whitelist, you must add the IP address of your computer to the whitelist. Otherwise, you will not be able to operate the project after the whitelist takes effect. Run the following command on the client: `setproject odps . security . ip . whitelist = xxx . xxx . xxx . xxx , xxx . xxx . x . x / xx , xxx . xxx . xxx . xxx - xxx . xxx . xxx . xxx ;` The whitelist supports IPv6 addresses, and the IP addresses in the whitelist can be expressed in one of the following three ways: · IP addresses, for example, `101 . 132 . 236 . 134` and `FE80 : 0202 : B3FF : FE1E : 8329` · Subnet masks, for example, `100 . 116 . 0 . 0 / 16` and `FE80 : 0101 : 4567 : F456 : 0202 : B3FF : 1111 : 1111 / 126` · Network segments, for example, `101 . 132 . 236 . 134 - 101 . 132 . 236 . 144` and `FE80 : 0101 : 4567 : F456 : 0202 : B3FF : FE1E : 8330 - FE80 : 0101 : 4567 : F456 : 0202 : B3FF : FE1E : 8331` The whitelist takes effect 5 minutes after you set it. If you want to manage permissions at finer levels, you can grant permissions by using policies.
Rollback settings	Delete the information following `setproject odps . security . ip . whitelist =:` `setproject odps . security . ip . whitelist =;` After you do so, the whitelist is disabled for the project.

Disable SELECT result download in DataWorks

Disable `SELECT result download` in DataWorks. We recommend that you disable `SELECT result download`. When you analyze data by using DataWorks, the data analysis results are displayed in IDE and can be downloaded. After `ProjectProtection` is set to `true`, you only need to have the permissions to read tables in the project for you to be able to select and download data analysis results in Data Analytics in the DataWorks console.

Item	Description
Role	`Project Manager` in DataWorks
View the function status	Log on to the DataWorks console, navigate to Workspace Settings, and check whether `SELECT result download` is enabled.
Set the function	Log on to the DataWorks console, navigate to Workspace Settings, and disable `SELECT result download`.
Roll back settings	Log on to the DataWorks console, navigate to Workspace Settings, and enable `SELECT result download`.

Promote security management by using other cloud services

You may use other cloud services while using MaxCompute. Therefore, you can promote the security management of MaxCompute by using the other associated cloud services. For example, when you use MaxCompute on the DataWorks console, you need to use RAM user accounts to add members to projects. The following describes how to promote security management on RAM user accounts.

MaxCompute supports two account systems: the Alibaba Cloud account system and the RAM user account system. MaxCompute can identify RAM users but cannot identify their permissions, which enables you to add any RAM users under the `project owner` account of a project to this project. When MaxCompute authenticates these RAM users, it does not verify their permissions. Therefore, you only need to promote security management on logons of RAM users.

Set password policies for RAM users

If you allow RAM users to change their passwords, you need to specify strong password policies such as the password length, whether characters other than letters are required, and the intervals at which RAM users change their passwords.

Set logon address masks for RAM users

You can set log address masks to specify from which IP addresses RAM users can log on the DataWorks console.

Revoke the permissions that RAM users no longer require

When some permissions of a RAM user are not longer used because the user's position changes, you need to revoke these permissions promptly.

2 Configure security features

2.1 Target users

This article is intended for MaxCompute project owners, administrators, and users interested in the MaxCompute multi-tenant data security system.

The MaxCompute multi-tenant data security system includes:

- User authentication.
- User and authorization management of projects.
- Sharing of resources across projects.
- Data protection of projects.

2.2 Quick Start

2.2.1 Use case: Add users and grant permissions

Description:

Jack is the project administrator of a project prj1. A new team member named Alice , who already has an Alibaba Cloud account as alice@aliyun.com, applies to join the prj1project. Alice requests the following permissions: view table lists, submit jobs, and create tables.

Solution:

As a project administrator, Jack performs the following procedure to add Alice as the user and grant her permissions to view table lists, submit jobs, and create tables:

```
use prj1 ;
add user aliyun $ alice @ aliyun . com ; -- Add the user
grant List , CreateTabl e , CreateInst ance on project
prj1 to user aliyun $ alice @ aliyun . com ; -- Authorize
the user by using the GRANT statement
```

2.2.2 Use case: Add users and grant permissions using ACL

This article shows you how to add a project role and authorize it through ACLs.

Description:

Jack is the project administrator of a project prj1. The three new data auditors, Alice , Bob, and Charlie, are added to the project team. They all need to apply for the following permissions: view table lists, submit jobs, and read the table userprofile.

Solution:

As a project administrator, Jack can perform authorization by using the object-based ACL Authorization.

Jack must perform the following procedure:

```
use prj1 ;
add user aliyun $ alice @ aliyun . com ; -- Add the user
add user aliyun $ bob @ aliyun . com ;
add user aliyun $ charlie @ aliyun . com ;
create role tableviewe r ; -- Create a role
grant List , CreateInst ance on project prj1 to
role tableviewe r ; -- Grant permission s to the role
grant Describe , Select on table userprofil e to
role tableviewe r ;
grant tableviewe r to aliyun $ alice @ aliyun . com ; --
Grant the tableviewe r role to the user
grant tableviewe r to aliyun $ bob @ aliyun . com ;
grant tableviewe r to aliyun $ charlie @ aliyun . com ;
```

2.2.3 Use case: Project data protection

Description:

Jack is the project administrator of a project prj1. The project involves a large volume of sensitive data including user IDs, shopping records along with the data mining algorithms with proprietary intellectual property rights. Jack wants to protect the sensitive data and algorithms and allow only project users to access the data within the project. He also wants to make sure that data flows within the project only.

Solution:

To protect the project data, Jack must perform these steps:

```
use prj1 ;
set ProjectPro tection = true ; -- Enable the project
data protection mechanism
```

Once the project data protection is enabled, data within the project cannot be transferred out of the project. All the data flows only within the project.

If users want to export data tables out of the project, an approval of the project administrator is needed. Here, MaxCompute provides the TrustedProject configuration to support external data export from the protected project. In this case,

configure project prj2 as a trusted project of prj1 and enable data flow from prj1 to prj2 through the following command:

```
use   prj1 ;
add   trustedpro  ject   prj2 ;
```

2.3 Manage users and permissions

2.3.1 Authenticate users

MaxCompute supports the Alibaba Cloud account system and the RAM account system.

 Note:
MaxCompute recognizes the RAM account system but cannot recognize the RAM permission system. As a user, you can add any of your RAM sub-accounts to a MaxCompute project. However, MaxCompute skips the RAM permission definitions when it verifies the permissions of the RAM sub-account.

By default, the MaxCompute project only recognizes the Alibaba Cloud account system. You can view the account system supported by this project by running `list accountpro viders ;`.

Typically, only Alibaba Cloud accounts are displayed. To add the RAM account system, run the `add accountpro vider ram ;` command. After the RAM account system is added, run `list accountpro viders ;` to make sure it has been successfully added to the supported account systems.

Apply for an Alibaba Cloud account

If you do not have an Alibaba Cloud account, visit here to apply for one.

 Note:
A valid email address is needed, when you apply for an Alibaba Cloud account. Because this email address is used as the account name after registration. For example, Alice can use her email address alice@aliyun.com to register an Alibaba Cloud account. Her account name will be alice@aliyun.com after Alibaba Cloud account registration.

Apply for AccessKey

Click here to create or manage your AccessKey list after you register an Alibaba Cloud account.

An AccessKey consists of the AccessKeyID and AccessKeySecret. The AccessKeyID is used to retrieve the AccessKey, and the AccessKeySecret is used to sign the computing messages. You must secure your AccessKey for further use. If you need to update an AccessKey, create a new AccessKey and disable the existing one.

Log on to MaxCompute with an Alibaba Cloud account

Configure the AccessKey in the configuration file *conf / odps_confi g . ini* before you use odpscmd to log on. See the following example:

```
project_na me = myproject
access_id =< Input the AccessKeyI D here , excluding the
angle brackets >
access_key =< Input the AccessKey here , excluding the
angle brackets >
end_point = http :// service . odps . aliyun - inc . com / api
```

Note:

To enable or disable an AccessKey on the Alibaba Cloud website, wait for at least 15 minutes after the operation is complete.

2.3.2 Manage users

Any user, except the project owner, must be added to the MaxCompute project and granted the corresponding permissions to manage data, jobs, resources, and functions in MaxCompute. This article describes how a project owner can add, authorize, and remove other users, including RAM sub-accounts to MaxCompute.

If you are a project owner, we recommend that you read this article carefully. If you are a typical user, we recommend that you submit an application to the project owner to be added to the corresponding project. We recommend all users to read the subsequent sections.

All the operations mentioned in this article are executed on the console. For Linux, run `./ bin / odpscmd` and for Windows, run `./ bin / odpscmd . bat .`

Add a user

In this example, the project owner, Alice, wants to authorize another user, therefore she must add the user to the project first. Only a user who has been added to the project can be authorized.

The command to add a user is as follows:

```
add   user
```

The <username> of an Alibaba Cloud account is a valid email address registered with Alibaba Cloud, or a RAM sub-account of an Alibaba Cloud account that runs the command. For example:

```
add   user   ALIYUN $ odps_test_  user @ aliyun . com ;
add   user   RAM $ ram_test_u  ser ;
```

Assume that the Alibaba Cloud account of Alice is alice@aliyun.com. When Alice runs these statements, the following results are returned by running the `list  users ;` command:

```
RAM $ alice @ aliyun . com : ram_test_u  ser
ALIYUN $ odps_test_  user @ aliyun . com
```

This indicates that the Alibaba Cloud account odps_test_user@aliyun.com and the sub-account ram_test_user created by Alice using RAM have been added to the project.

Add a RAM sub-account

The two ways to add a RAM sub-account are as follows:

- By using DataWorks, for more information, see Prepare a RAM account.
- By using MaxCompute client commands as described in this document.

> Note:
>
> - MaxCompute only allows a primary account to add its own RAM sub-accounts to a project. RAM sub-accounts of other Alibaba Cloud accounts are not allowed. Therefore, you can skip to specify the name of the primary account before the RAM sub-accounts when `add  user`. MaxCompute determines by default that the account which runs the command is the corresponding sub-account.

- MaxCompute only recognizes the RAM account system and does not recognize the RAM permission system. Users can add any of their RAM sub-accounts to a MaxCompute project, but MaxCompute does not consider the permission limits in RAM when performing permission verification of RAM sub-accounts.

By default, MaxCompute project only recognizes Alibaba Cloud account systems. To view the supported account systems use the `list accountpro viders ;` command. Typically, only the ALIYUN account is visible, for example:

```
odps @ ****> list accountpro viders ;
ALIYUN
```

Note:

Only the project owner has the permission to perform operations related to `accountpro viders` .

As shown in the preceding command, you can only see the `ALIYUN` account system. If you want to add RAM accounts support, run the `add accountpro vider ram ; as follows : ram ;` as follows:

```
odps @ odps_pd_in ter > add accountpro vider ram ;
OK
```

The user will still not be able to operate MaxCompute successfully. This is because, the user must be granted certain permissions to operate MaxCompute within the permissive limits. For more information, see Authorization.

User Authorization

Once the user is added, the project owner or project administrator must authorize the user. The user can perform the operations only after obtaining the permissions.

MaxCompute provides ACL authorization, cross-project resource sharing, and project resource protection. The following are two common scenarios, for more information, see ACL Authorization .

Scenario 1

In the following scenario, Jack is the administrator of the project prj1. A new project team member Alice (Alibaba Cloud account: alice@aliyun.com) applies to join the project prj1, and for permission to view table lists, submit jobs, and create tables.

The admin or the project owner can run the following command on the client:

```
use prj1 ; -- Open the project prj1
add user aliyun $ alice @ aliyun . com ; -- Add the user
grant List , CreateTabl e , CreateInst ance on project
prj1 to user aliyun $ alice @ aliyun . com ; -- Authorize the
  user
```

Scenario 2

In the following scenario, assume Alibaba Cloud account user (bob@aliyun.com) has been added to a project ($user_project_name), and must be granted permission to create tables, obtain table information, and run functions.

The admin or the project owner can run the following command on the client:

```
grant CreateTabl e on PROJECT $ user_proje ct_name to
USER ALIYUN $ bob @ aliyun . com ;
 -- Grant CreateTabl e permission on project “$ user_proje
ct_name ” to bob @ aliyun . com
grant Describe on Table $ user_table _name to USER
ALIYUN $ bob @ aliyun . com ;
 -- Grant Describe permission on table “$ user_table _name
” to bob @ aliyun . com
grant Execute on Function $ user_funct ion_name to USER
ALIYUN $ bob @ aliyun . com ;
 -- Grant Run permission on function "$ user_funct ion_name
" to bob @ aliyun . com
```

Authorize RAM Sub-account

To check accounts support, run `list accountpro viders ;` command as follows:

```
odps @ ****> list accountpro viders ;
ALIYUN , RAM
```

In this project, RAM accounts are also supported. You can add a RAM sub-account to this project and grant `Describe` permission on the tables. For example:

```
odps @ ****> add user ram $ bob @ aliyun . com : Alice ;
OK : DisplayNam e = RAM $ bob @ aliyun . com : Alice
odps @ ****> grant Describe on table src to user ram $
bob @ aliyun . com : Alice ;
OK
```

After running these commands, *Alice* account, which is a RAM sub-account of *bob@aliyun.com* , can logon to MaxCompute with the AccessKeyID and AccessKeyS ecret, and run `desc` on the table *src*.

 **Note:**

- For more information about how to create a RAM sub-account `AccessKeyI D` and `AccessKeyS ecret`, see RCreate a RAM user.
- For more information about how to add or remove users on MaxCompute, see the corresponding content of this article.
- For more information about authorizing a user, see Authorization.

Remove a User

When a user leaves the project team, Alice must remove the user from the project. Once removed from the project, the user no longer has any access permission to the project resources.

The command to remove a user from a project is as follows:

```
remove  user
```

Note:

- A user removed from a project immediately loses an authority to access resources of the project.
- Revoke all the roles of the user, before removing a user whom the roles are assigned. For more information about roles, see Role Management.
- After a user is removed, allACL Authorization data related to the user is retained. After a user is added to a project again, the ACL Authorization of this user is enabled again.
- MaxCompute does not support complete removal of a user and all permission data from a project.

To remove corresponding users, Alice can run the following commands:

```
remove  user  ALIYUN $ odps_test_  user @ aliyun . com ;
remove  user  RAM $ ram_test_u  ser ;
```

To make sure the users are removed, run the following command:

```
LIST  USERS ;
```

If those two accounts are no longer listed after running the command, it indicates that the accounts have been removed from the project.

Remove a RAM Sub-account

Similarly, RAM sub-account can be removed by using the `remove user` command. For example:

```
odps @ ****> revoke describe on table src from user ram
$ bob @ aliyun . com : Alice ;
OK
-- Revoke Alice sub - account permission s
odps @ ****> remove user ram $ bob @ aliyun . com : Alice ;
Confirm to " remove user ram $ bob @ aliyun . com : Alice ;" (
yes / no )? yes
OK
-- Remove sub - account
```

If you are the project owner, you can also remove the RAM account system from the current project by `remove accountpro vider` as follows:

```
odps @ ****> remove accountpro vider ram ;
Confirm to " remove accountpro vider ram ;" ( yes / no )?
yes
OK
odps @ ****> list accountpro viders ;
ALIYUN
```

2.3.3 Manage roles

A role is a defined set of access permissions. It assigns the same set of permissions to a group of users. Role-based authorization greatly simplifies the authorization process and reduces the authorization management cost. It must be used with priority.

When a project is created, an admin role is automatically created with a definite privilege authorized to the role, including access to all objects within the project, management of users and roles, and authorization to users and roles. In comparison to a project owner, the admin role cannot assign admin permission to any user, set the project security configuration, or change the authentication model for the project. Permissions of the admin role cannot be modified.

Role management related commands include the following:

```
    create role < rolename > -- Create a role
    drop role < rolename > -- Delete a role
    grant < rolename > to < username > -- Grant a role to
a user
    revoke < rolename > from < username > -- Revoke a role
from a user
```

```
```

Note:

- One role can be assigned to multiple users at the same time, and one user can be assigned multiple roles.
- For more information about the mapping between the roles in DataWorks and in MaxCompute, and the platform permissions of these roles, see the project member management module in Project Management.

Create a role

To create a role, use the following command :

```
CREATE ROLE ;
```

Example:

To create a role player, enter the following command on the client:

```
create role player ;
```

Note:
The role permissions you create can view the specified user permissions through Permission check.

Add a user to the role

To add a user to the role, use the following command:

```
GRANT < roleName > TO < full_usern ame > ;
```

Example:

To assign user bob@aliyun.com the player role, enter the following command on the console:

```
grant player to bob @ aliyun . com ;
```

Authorize role

The authorization statement for the role is similar to the authorization for the user. For more information, see User authorization.

Note:

After role authorization is complete, all users under this role have the same permissions.

Example:

Jack is the administrator of project prj1. Three new data auditors, Alice, Bob, and Charlie, are added to the project team. They must apply for the following permissions : view the table lists, submit the jobs, and read the table userprofile.

In this scenario, the project administrator can perform authorization by using the object-based ACL Authorization.

The commands are as follows:

```
use prj1 ;
add user aliyun $ alice @ aliyun . com ; -- Add the user
add user aliyun $ alice @ aliyun . com ; -- Add the user
add user aliyun $ charlie @ aliyun . com ;
create role tableviewe r ; -- Create a role
grant List , CreateInst ance on project prj1 to
role tableviewe r ; -- Grant permission s to the role
grant Describe , Select on table userprofil e to
role tableviewe r ;
grant tableviewe r to aliyun $ alice @ aliyun . com ; --
Grant the tableviewe r role to the user
grant tableviewe r to aliyun $ bob @ aliyun . com ;
grant tableviewe r to aliyun $ charlie @ aliyun . com ;
```

Revoke the role from the user

To revoke the role from the user, use the following command:

```
REVOKE < roleName > FROM < full_usern ame >;
```

Example:

To remove the user bob@aliyun.com from the player role, use the following command on the client:

```
revoke player from bob @ aliyun . com ;
```

Delete a Role

To delete a role, use the following command:

```
DROP ROLE < roleName >;
```

Example:

To delete the role of the player, use the following command:

```
drop   role   player ;
```

Note:
When a role is deleted a role, MaxCompute checks whether other users are in this role. If yes, this role cannot be deleted. The role can be successfully deleted only when all users in the role are revoked from this role.

2.3.4 Authorize users

Authorization allows a user to perform operations including read, write, and view on tables, tasks, resources, and other objects of the MaxCompute. After the user is added, the project owner or the project administrator must authorize the user. The user can perform operations only after obtaining the permission.

MaxCompute provides Access Control List (ACL) authorization, cross-project resource sharing, and project resource protection. Authorization typically includes three elements: subject, object, and action. In MaxCompute, the subject refers to a user or a role and the object refers to various types of objects in a project.

ACL authorization includes following MaxCompute objects: Project,Table ,Function , Resource, and Instance. Operations are related to specific object types, therefore different types of objects support different types of actions.

MaxCompute projects support the following object types and actions:

Object	Action	Description
Project	Read	View project information (excluding any project objects), such as the creation time.
Project	Write	Update project information (excluding any project objects), such as comments.
Project	List	View the list of all types of objects in the project.
Project	CreateTable	Create a table in the project.
Project	CreateInstance	Create an instance in the project.
Project	CreateFunction	Create a function in the project.
Project	CreateResource	Create a resource in the project.
Project	All	Grant all of the preceding permissions.
Table	Describe	Read the metadata of the table.

Object	Action	Description
Table	Select	Read the table data.
Table	Alter	Change the metadata of the table and add or delete a partition.
Table	Update	Overwrite or add table data.
Table	Drop	Delete a table.
Table	All	Grant all the preceding permissions.
Function	Read	Read and run permissions.
Function	Write	Update.
Function	Delete	Delete.
Function	Run	Run.
Function	All	Grant all the preceding permissions.
Resource	Read	Read.
Resource	Write	Update.
Resource	Delete	Delete.
Resource	All	Grant all the preceding permissions.
Instance	Read	Read.
Instance	Write	Update.
Instance	All	Grant all the preceding permissions.

Note:

- The CreateTable action for the objects of Project type must work with the CreateInstance permission for the Project object. The Select, Alter, Update, and Drop actions for the objects of Table type must work with the CreateInstance permission for the Project object.
- If the CreateInstance permission is not granted, the corresponding operations cannot be performed even though the mentioned permissions are granted. This is related to the internal implementation of MaxCompute. The Select permission for Table type objects must work with the CreateInstance permission. While performing cross-project operation, such as selecting the table of project B in the project A, you must have the project A CreateInstance and the project B Table select permissions.

- After a user or role is added, you must grant permissions to the user or role . MaxCompute authorization is an object-based authorization method. The permission data authorized by ACL is considered as a type of sub-resource of the object. Authorization can be performed only if the object exists. When the object is deleted, the authorized permission data is automatically deleted.

- SQL92 Authorization

  MaxCompute supports authorization using the syntax similar to the GRANT and REVOKE commands defined by SQL92. It grants or revokes permissions to/from the existing project object through simple authorization statements. The authorization syntax is as follows:

```
grant actions on object to subject
revoke actions on object from subject
actions ::= action_ite m1 , action_ite m2 , ...
object ::= project project_na me | table schema_nam e
|
          instance inst_name | function func_name |
           resource res_name
subject ::= user full_usern ame | role role_name
```

  Users familiar with GRANT and REVOKE commands defined by SQL92 or with Oracle database security management can identify that the ACL authorization syntax of MaxCompute does not support [WITH GRANT OPTION] authorization parameters. For example, when User A authorizes User B to access an object, User B cannot grant the permission to User C. In this scenario, all permissions can be granted by one of the following three roles:

  - Project owner
  - Project administrator
  - Object creator

- Use example of ACL authorization

  In the following scenario, the Alibaba Cloud account user alice@aliyun.com is a newly added member to the project test_project_a, and Allen is a RAM-sub account added to bob@aliyun.com. In test_project_a, they both must submit jobs, create tables, and view existing objects in the project.

  The project administrator bob performs the following authorization operations:

```
use test_proje ct_a ;
add user aliyun $ alice @ aliyun . com ;
add user ram $ bob @ aliyun . com : Allen ;
create role worker ;
```

```
    grant   worker   TO   aliyun $ alice @ aliyun . com ;
    grant   worker   TO   ram $ bob @ aliyun . com : Allen ;
    grant   CreateInst  ance ,  CreateReso  urce ,  CreateFunc
tion ,  CreateTabl  e ,  List   ON   PROJECT   test_proje  ct_a
TO   ROLE   worker ;
```

- Cross-project Table/Resource/Function sharing

  Following the preceding example, aliyun$alice@aliyun.com and ram$bob@aliyun.com:Allen have certain permissions in test_project_a. These two users must query table prj_b_test_table in test_project_b, and use test_project_b. UDF prj_b_test_udf.

  The project administrator performs the following authorization operations for test_project_b:

```
    use   test_proje  ct_b ; -- Open   the   project
    add   user   aliyun $ alice @ aliyun . com ; -- Add   the
user
    add   user   ram $ bob @ aliyun . com : Allen ; -- Add   th
RAM   sub - account
    create   role   prj_a_work  er ; -- Create   a   role
    grant   prj_a_work  er   TO   aliyun $ alice @ aliyun . com ;
-- Grant   the   role
    grant   prj_a_work  er   TO   ram $ bob @ aliyun . com : Alice
; -- Grant   the   role
    grant   Describe  ,  Select   ON   TABLE   prj_b_test  _table
TO   ROLE   prj_a_work  er ; -- Authorize   the   role
    grant   Read   ON   Function   prj_b_test  _udf   TO   ROLE
prj_a_work  er ; -- Authorize   the   role
    grant   Read   ON   Resource   prj_b_test  _udf_resou  rce   TO
  ROLE   prj_a_work  er ; -- Authorize   the   role
-- After   authorizat  ion ,  the   two   users   query   table
and   use   udf   in   test_proje  ct_a   as   follows :
use   test_proje  ct_a ;
select   test_proje  ct_b : prj_b_test  _udf ( arg0 ,  arg1 )  as
res   from   test_proje  ct_b . prj_b_test  _table ;
```

Note:

If UDF is created in test_project_a, then only Resource authorization is required. Use the following code:

```
create   function   function_n  ame   as  ' com . aliyun . odps .
compiler . udf . PlaybackJs  onShrinkUd  f '  using  ' test_proje
ct_b / resources / odps - compiler - playback . jar ' - f ;.
```

2.3.5 Check permissions

MaxCompute provides the ability to view multiple permissions, including the permissions of certain users or roles, and authorization lists of specified objects.

MaxCompute uses the markup characters A, C, D, and G when showing the permissions of users or roles. The meanings of these markup characters are as follows:

- A: Access allowed.
- D: Access denied.
- C: Access granted with conditions. It appears only in a policy authorization system.
- G: Access granted with conditions. Permission can be granted to objects.

An example of viewing permissions is as follows:

```
    odps @ test_proje  ct >  show   grants   for   aliyun $ odpstest1
@ aliyun . com ;
   [ roles ]
    dev
    Authorizat  ion   Type :  ACL
   [ role / dev ]
    A   projects / test_proje  ct / tables / t1 :  Select
   [ user / odpstest1 @ aliyun . com ]
    A   projects / test_proje  ct :  CreateTabl  e  |  CreateInst
ance  |  CreateFunc  tion  |  List
    A   projects / test_proje  ct / tables / t1 :  Describe  |
Select
    Authorizat  ion   Type :  Policy
   [ role / dev ]
    AC   projects / test_proje  ct / tables / test_ *:  Describe
    DC   projects / test_proje  ct / tables / alifinance  _ *:
Select
   [ user / odpstest1 @ aliyun . com ]
    A   projects / test_proje  ct :  Create * |  List
    AC   projects / test_proje  ct / tables / alipay_ *:  Describe
|  Select
    Authorizat  ion   Type :  ObjectCrea  tor
    AG   projects / test_proje  ct / tables / t6 :  All
    AG   projects / test_proje  ct / tables / t7 :  All
```

Note:

Currently, desc role only displays ACL information of project and table authorization types, while ACL of other objects (function, resource, instance, job) does not support display.

View permissions of a specified user

```
    show   grants ; -- View   permission  s   of   the   current
user .
    show   grants   for  < username >; -- View   access   permission
s   of   a   specified   user .  The   operation   can   be
executed   by   project   owners   and   administra  tors .
```

Example:

To view the user Alibaba Cloud account bob@aliyun.com permissions in the current project, run the following command on the client:

```
show grants for ALIYUN $ bob @ aliyun . com ;
```

To view RAM sub-account permissions:

```
show grants for RAM $ account : sub - account ;
```

Example:

```
show grants for RAM $ bob @ aliyun . com : Alice ;
```

View permissions of a specified role:

```
describe role -- View access permission s granted to a
  specified role
```

Note:
In the public cloud environment, description role currently only displays ACL information of the object authorization type of project and table, while ACL information of other objects (such as function, resource, instance, job) is not displayed.

View the authorization list of a specified object:

```
show acl for < objectName > [ on type < objectType >];--
View the user and role authorizat ion list of a
specified object
```

Note:
When *[on type <objectType>]* is excluded, the default type is Table.

2.4 Column-level access control

Label-based security (LabelSecurity) is a required MaxCompute Access Control (MAC) policy at the project space level. It allows project administrators to control the user access to column-level sensitive data with improved flexibility.

Difference between MAC and DAC in MaxCompute

In MaxCompute, MAC is independent of Discretionary Access Control (DAC). Two examples are provided to illustrate the differences between MAC and DAC.

To drive a vehicle, you must first have to apply and acquire a valid driver's license, similarly, a user who wants to read data in a MaxCompute project must first apply for the SELECT permission. The permission application is within the scope of DAC.

Because the country with a high accident rate, drunk driving is strictly restricted. To curb this, all drivers are required to have a driver’ s license and must not drink and drive. Likewise, in MaxCompute, reading highly sensitive data is analogous to the law against drunk driving. The read prohibition is within the scope of MAC.

Data sensitivity classification

LabelSecurity assigns security levels to data and the users who access the data. In the government and financial sectors, data sensitivity is usually classified into four levels: 0 (Unclassified), 1 (Confidential), 2 (Sensitive), and 3 (Highly Sensitive). MaxCompute adopts such classification. Project owners must define standards for data sensitivity classification and access level classification. The default access level of all users is 0, and the default sensitivity level of data is 0.

LabelSecurity supports data sensitivity classification at the column level. Administrators can set sensitivity labels for all the columns of a table. A table may have columns of different sensitivity levels.

Administrators can also set sensitivity labels for views. A view and its base table have independent sensitivity labels. The default sensitivity level of a new view is 0.

Default security policies of LabelSecurity

LabelSecurity applies the following default security policies to the data and users assigned with sensitivity or security labels:

- No-ReadUp: A user is not allowed to read data with a sensitivity level higher than the user level unless the user is explicitly authorized.
- Trusted-User: A user is allowed to write data of all sensitivity levels. The default sensitivity level of new data is 0 (unclassified).

 Note:

- In some traditional MAC systems, other complex security policies are applied to prohibit unauthorized data distribution in a project. For example, the No-WriteDown policy prohibits users from writing data with a sensitivity level not higher than the user level. By default, MaxCompute does not support No-

WriteDown, considering the costs involved in managing the data sensitivity levels of project administrators. The effect of No-WriteDown can be attained by modifying the project security settings (`Set ObjectCrea torHasGran tPermissio n = false`).

- To prohibit data flowing among different projects, you can set the projects to the protected state (ProjectProtection). With the setting, users can only access the data within their projects. This prevents data transfer or data sharing outside the project.

By default, projects disable LabelSecurity. The project owners can enable it as required.

After LabelSecurity is enabled, the default security policies are executed. When a user accesses a data table, the user must have the SELECT permission and the access level required for sensitive data reading. Compliance with LabelSecurity is a required but not the sufficient condition for passing CheckPermission.

LabelSecurity operations

- Enable or disable LabelSecurity

```
 Set  LabelSecur ity = true | false ;
  -- Enables  or  disables  LabelSecur ity .  The  default
value  is  false .
  -- LabelSecur ity  can  be  enabled  or  disabled  only
  by  the  project  owner .  Other  operations  can  be
performed  by  the  project  administra tor .
```

- Set security labels for users

```
 SET  LABEL  < number >  TO  USER  < username >;--  Value
range  of  " number ": [ 0 ,  9 ].  This  operation  can  be
  performed  only  by  the  project  owner  or  administra
tor .
- Example :
 ADD  USER  aliyun $ yunma @ aliyun . com ; -- Adds  a  user
with  the  default  security  label  0 .
 ADD  USER  ram $ yunma @ aliyun . com : Allen ; -- Adds  user
Allen ,  which  is  a  RAM  subaccount  of  yunma @ aliyun .
com .
 SET  LABEL  3  TO  USER  aliyun $ yunma @ aliyun . com ;
  -- Sets  the  security  label  of  yunma  to  3  to
allow  this  user  to  access  only  the  data  with  a
sensitivit y  level  not  higher  than  3 .
 SET  LABEL  1  TO  USER  ram $ yunma @ aliyun . com : Allen ;
```

```
  -- Sets the security label of subaccount Allen to
  1 to allow this user to access only the data
with a sensitivit y level not higher than 1 .
```

- Set sensitivity labels for data

```
 SET LABEL < number > TO TABLE tablename ( column_lis t );
-- Value range of " number ": [ 0 , 9 ]. This operation
  can be performed only by the project owner or
administra tor .
- Example :
 SET LABEL 1 TO TABLE t1 ; -- Sets the sensitivit y
  label of table t1 to 1 .
 SET LABEL 2 TO TABLE t1 ( mobile , addr ); -- Sets
the sensitivit y labels of the " mobile " and " addr "
columns of table t1 to 2 .
 SET LABEL 3 TO TABLE t1 ; -- Sets the sensitivit y
  label of table t1 to 3 . The sensitivit y labels
  of the " mobile " and " addr " columns are still 2 .
```

Note:

The sensitivity labels explicitly set for the columns overwrite the sensitivity label set for the table, without considering the label setting order and the sensitivity level.

- Explicitly authorize lower-level users to access specific data tables with a high sensitivity level

```
-- Grant permission s :
 GRANT LABEL < number > ON TABLE < tablename >[( column_lis
t )] TO USER < username > [ WITH EXP < days >]; -- The
default validity period is 180 days .
-- Revoke the permission s :
 REVOKE LABEL ON TABLE < tablename >[( column_lis t )]
FROM USER < username >;
-- Clear the expired permission s :
 CLEAR EXPIRED GRANTS ;
- Example :
 GRANT LABEL 2 ON TABLE t1 TO USER ram $ yunma @
aliyun . com : Allen WITH EXP 1 ; -- Explicitly authorizes
  Allen to access the data of table t1 with a
sensitivit y level not higher than 2 for a period
  of 1 day .
 GRANT LABEL 3 ON TABLE t1 ( col1 , col2 ) TO USER
ram $ yunma @ aliyun . com : Allen WITH EXP 1 ; -- Explicitly
  authorizes Allen to access the data in col1 and
  col2 of table t1 with a sensitivit y level not
higher than 3 for a period of 1 day .
 REVOKE LABEL ON TABLE t1 FROM USER ram $ yunma @
aliyun . com : Allen ; -- Revokes the permission of Allen
to access the sensitive data in table t1 .
```

Note:

Once the label-authorized permission of a user to access a table is revoked, the permission to access the table fields of the same user is also revoked.

- List the sensitive data sets that a user can access

```
SHOW LABEL [< level >] GRANTS [ FOR USER < username >];
  -- When [ FOR USER < username >] is unspecifie d , the
 system lists the sensitive data sets that the
current user can access .
  -- When < level > is unspecifie d , the system lists
the permission s granted by all label levels . When
 < level > is specified , the system lists only the
permission s granted by a specific label level .
```

- List the users who can access a specific table containing sensitive data

```
SHOW LABEL [< level >] GRANTS ON TABLE < tablename >;
 -- Displays the label - authorized permission s on the
 specified table .
```

- List the label-authorized permissions of a user at all levels to access a data table

```
SHOW LABEL [< level >] GRANTS ON TABLE < tablename > FOR
 USER < username >;
  -- Displays the label - authorized permission s of
the specified user to access the columns of a
specific table .
```

- List the sensitivity levels of all the columns of a table

```
DESCRIBE < tablename >;
```

- Control the access level of a package installer regarding the sensitive resources of the package

```
 ALLOW PROJECT < prjName > TO INSTALL PACKAGE < pkgName >
[ USING LABEL < number >];
  -- The package creator grants an access level to
the package installer regarding the sensitive resources
  of the package .
```

Note:

- When `[ USING LABEL < number >]` is unspecified, the default access level is 0. The package installer can only access non-sensitive data.
- When accessing to sensitive data across projects, the access level defined by this command applies to all the users in the project of the package installer.

LabelSecurity use cases

- **Prohibit all the users in a project except the project administrator from reading some sensitive columns of a table**

  Description:

  user_profile is a table with sensitive data in a project. It has 100 columns, five of which contain sensitive data: id_card, credit_card, mobile, user_addr, and birthday. DAC grants all users the SELECT permission on this table. The project owner wants to prohibit all the project users except the project administrator from reading the sensitive columns of the table.

  To achieve this purpose, the project owner can perform the following operations:

```
 set  LabelSecur  ity = true ;
  -- Enables  LabelSecur  ity .
 set  label  2  to  table  user_profi  le ( mobile ,
user_addr ,  birthday );
  -- Sets  the  sensitivit  y  level  of  the  specified
columns  to  2 .
 set  label  3  to  table  user_profi  le ( id_card ,
credit_car  d );
  -- Sets  the  sensitivit  y  level  of  the  specified
columns  to  3 .
```

  Note:
  After the preceding operations, non-administrator users cannot access the data in the five columns. To access the sensitive data for business purposes, the user must be authorized by the project owner or administrator.

  Solution:

  Alice is a member of the project. For official purposes, she wants to apply for access to the data in the mobile column of table user_profile for a period of one week. To authorize Alice, the project administrator can perform the following operation:

```
GRANT  LABEL  2  ON  TABLE  user_profi  le  TO  USER
ALIYUN $ alice @ aliyun . com  WITH  EXP  7 ;
```

  Note:
  Mobile, user_addr, and birthday column contain data with a sensitivity level of 2. Birthday. After authorization, Alice can access the data in these three columns.

The authorization causes the issue of excessive permission grants. This issue can be avoided if the project administrator sets the sensitive columns properly.

- Prohibit the project users with access to sensitive data from copying and distributi ng the sensitive data within the project without authorization

  Description:

  In the preceding use case, Alice is granted the access permission on the data with a sensitivity level of 2 for official purposes. The project administrator worries that Alice may copy that data from table user_profile to table user_profile_copy created by her and grants Bob the access permission on user_profile_copy. The project administrator needs a method to restrict Alice's actions.

  Solution:

  Considering security usability and management costs, LabelSecurity adopts the default security policy that allows for WriteDown. Users can write data to the columns with a sensitivity level not higher than the user level. MaxCompute cannot address the preceding requirement of the project administrator. However, the project administrator can restrict the discretionary authorization behavior of Alice by allowing her to only access the data she created, but disallowing her to grant the data access permission to other users. The procedure is as follows:

```
SET ObjectCrea torHasAcce ssPermissi on = true ;
 -- Allows the object creator to operate objects .
SET ObjectCrea torHasGran tPermissio n = false ;
 -- Prohibits the object creator from granting the
object access permission to other users .
```

2.5 Resource share across project space

2.5.1 Resource sharing across projects based on package

Assume that you are the project owner or administrator (admin role) of a few projects. One of your primary accounts has multiple projects, wherein the project prj1 has some resources (including tables, resources, and custom functions) that can be shared with other projects. However, adding users of other projects to prj1 and granting permissions to them one by one is complicated, and adding the users who are irrelevant but are added to the prj1 project (if they exist) complicates the project management.This section describes cross-project resource sharing.

If resources must be controlled by the user in a fine-grained manner, and the user who applies for the control permission is a member of the business project team, we recommend using the Project user and authorization management feature.

Package is used for sharing data and resources across projects. It solves the problem of cross-project user authorization.

Use package to solve the following problems effectively:

If members of the Alifinance project want to access data in the Alipay project, the administrator of the Alipay project must perform tedious authentication operations : First, add users in the Alifinance project to the Alipay project, and then perform general authentications on the newly added users, respectively.

Actually, the administrator of the Alipay project does not want to authenticate and manage all users in the Alifiance project. Instead, the administrator expects more efficient feature for autonomous authentication controls over permissive objects.

After Package is used, the administrator of the Alipay project can perform packaging authorization on the objects to be used by the Alifinance project (that is, create a Package), and then permit the Alifinance project to install the Package. After the Alifinance project's administrator installs the Package, the administrator can determine whether to grant permissions of the Package to the users of the Alifinance project as required.

2.5.2 Package usage method

This article introduces you to the operations involved in the project space Package creator and Package consumer.

Package usage method

The use of package involves two subjects: the package creator and the package user.

- The package creator provides the resources to be shared and the permissions to access it. It also allows the package user to install and use it.
- The package user uses the package. After the package is published, the user can directly access the resource across projects.

The following is a description of the operations involved with the package creator and package user.

Package creator

- Create package

```
create  package  < pkgname >;
```

Note:

- Only the project owner has the permission to create a package.
- The name of the package cannot exceed 128 characters.

- Add a resource to be shared to the package

```
    Add  project_ob  ject  to  package  package_na  me  [ with
  privileges ] --  add  objects  to  package
    Remove  project_ob  ject  from  package  package_na  me ;
--  remove  object  from  package
    project_ob  ject  ::=  table  table_name  |
                  instance  inst_name  |
                  function  func_name  |
                  resource  res_name
    privileges  ::=  action_ite  m1 ,  action_ite  m2 , ...
```

Additional considerations

- Currently, supported types of objects exclude projects. Therefore, you cannot use a package to create objects in other projects.
- When you add resources to a project, ensure that the entered object names do not contain the prefix of the project name. For example, if you want to add a table named `table_test` to a package in project `prj1` , the table name in the `ADD` statement cannot be `prj1 . table_test` . Enter `table_test` as the table name in the statement.
- The objects themselves and the permission to perform operations on them are added to the package at the same time. When not passed (with privileges) even specifying an action permission, the default is read-only, that is, read/describe /select. The object and its permissions are treated as a whole and cannot be updated once added. If necessary, you can only delete and re-add.
- When an object is added to a package, it is not packaged as a snapshot, so subsequent object data changes, and access to the object through package authorization is also the current data of the object.

- Allow other projects to use a package

```
allow project < prjName > to install package < pkgName > [
using label < num >]
```

- Revoke other projects' permission to use a package

```
disallow project < prjName > to install package < pkgName
>
```

- Drop a package

```
Delete package < pkgname >;
```

- View the list of packages already created and installed

```
Show packages ;
```

- View package details

```
Describe package < pkgname >;
```

Package users

- Install package

```
Install package < pkgname >;
```

For package installation, the pkgName format is: <projectName>.<packageName>.

> **Note:**
> Only the project owner has permissions to perform this operation.

- Uninstalling package

```
Uninstall package < pkgname >;
```

For package installation, the pkgName format is:
<projectName>.<packageName>.`< projectNam e >.< packageNam e >`

- View a package

```
    Show packages ;
    View the list of packages already created and
installed
    Describe package < pkgname >;
    View details of package
```

- Client project grants access to package to other members or role of this project

  The installed package is an independent type of MaxCompute object. To access resources in a package (resources shared with you by other projects), you must have the permission to read package.

  If you do not have the Read permission, you must apply to the project owner or admin for the permission. The project owner or admin can grant permissions through ACL authorization or policy authorization.

  Authorize package to user or role:

```
grant actions on package < pkgName > to user < username
>;
grant actions on package < pkgName > to role < role_name
>;
```

Note:
After authorization, user has access to the object in that package only in this project.

  For example, the following ACL authorization allows the cloud account user odps_test@aliyun.com to access resources in the package:

```
use prj2 ;
install package prj1 . testpkg ;
grant read on package prj1 . testpackag e to user
aliyun $ odps_test @ aliyun . com ;
```

  Or allow all members of role role_dev to access resources in package:

```
use prj2 ;
install package prj1 . testpkg ;
grant read on package prj1 . testpackag e to role
role_dev ;
```

Example

Jack is the administrator of prj1. John is the administrator of prj2. To address some business needs, Jack wants to share some resources of prj1 (such as datamining.jar and sampletable) to John's prj2. If prj2 user Bob must access these resources, the prj2 administrator can self-authorize Bob through ACL administrator or policy authorization without Jack’s involvement.

Procedure:

1. Prj1 administrator Jack creates resources package in prj1.

```
    Use   prj1 ;
    Create   package   datamicing ; --  creating   a   package
    Add   Resource   dating .  jar   to   package   dating ;- add
resource   to   package
    Add   Table   sampletabl  e   to   package   dating ; --
adding   table   to   package
    Allow   project   prm9   to   install   package   dating ; --
sharing   package   to   Project   Space   prm9
```

2. Prj2 administrator Bob installs a package in prj2.

```
    use   prj2 ;
    install   package   prj1 . datamining ; --  installs   a
package
    describe   package   prj1 . datamining ; --  view   a   list
of   resources   in   the   package
```

3. Bob self-authorizes the package.

```
    use   prj2 ;
    grant   Read   on   package   prj1 . datamining   to   user
aliyun $ bob @ aliyun . com ; --  authorizat  ion   of   Bob   to
use   package   via   ACL
```

2.6 Security configurations

MaxCompute is a multi-tenant data processing platform. Distinct tenants have distinct data security requirements. Therefore, MaxCompute provides project-level security configurations to comply with the unique requirements of individual tenants . Project owners can customize their external account support and authentication models.

MaxCompute provides multiple methods of orthogonal authorization, including Access Control List (ACL) authorization and implicit authorization. An object creator is automatically granted the object access permission. Not all users need these security features. Users can properly configure the project authentication model based on their service security requirements and usage patterns.

```
    show   SecurityCo  nfiguratio  n
     -- View   the   project   security   configurat  ion .
    set   CheckPermi  ssionUsing  ACL = true / false
     -- Enable / Disable   the   ACL   authorizat  ion   mechanism .
The   default   value   is   true .
    set   ObjectCrea  torHasAcce  ssPermissi  on = true / false
     -- Enable / Disable   automatic   access   permission   granting
  to   object   creators .  The   default   value   is   true .
    set   ObjectCrea  torHasGran  tPermissio  n = true / false -* +
     -- Enable / Disable   automatic   authorizat  ion   permission
  granting   to   object   creators .  The   default   value   is
true .
```

```
    set   ProjectPro  tection = true / false
       -- Enable / Disable   project   data   protection   to
enable / disable   data   transfer   from   the   project .
```

Note:

You can also complete the security configuration of a project in a visualized technique using DataWorks.

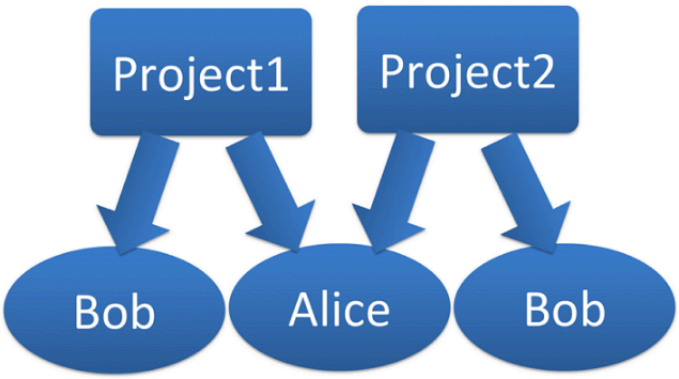
2.7 Data protection of projects

Background and motivation

Some companies (including financial institutions, military enterprises and so on) are extremely sensitive to data security. Hence, to secure the data, additional security measures are taken, that include not allowing employees to carry USB storage devices or personal hard disks to work; or most of the times the USB ports are disabled. Employees are not allowed to work from home. All these measures are taken to secure the sensitive data.

As a MaxCompute Project Space Administrator, do you have similar security requirements, where users are not allowed to move data out of the project space?

For example, the owner of Project Space prj1 may encounter a situation that prj1's user Alice will transfer the data to prj2, only because she has access to prj2.



More specifically, assume that Alice has been granted access to myprj, which is the Select permission for table1, and then she is also granted create table permission by the administrator of prj2.

By these permissions, Alice is able to transfer the data to prj2 in any of the following ways:

- Submit SQL:

```
create table prj2 . table2 as select * from myprj .
table1 ;
```

- Write MapReduce to read myprj.table1 and write to the prj2.table2.

If the data in your project space is sensitive, you will be restricted to share data out of your project. MaxCompute can resolve issues pertaining to data protection and the aforementioned operations as well.

Data protection feature

MaxCompute provides a project space protection feature that helps to resolve issues mentioned earlier. As a user, set the project as follows:

```
    set projectPro tection = true
     -- Set project protection rule : data can only
flow and cannot flow out
```

When project protection is set up, the data flow in your project space is controlled , "Data can only flow and cannot flow out ". That is, both of these actions will fail because they are against the project protection rule.

By default, ProjectProtection cannot be set and its value is false.

Also, users authorized to access multiple projects can freely use cross-project data access operations to share or transfer project data. If users are highly sensitive to project data security, the administrator must define a ProjectProtection feature likewise.

Data outflow method after enabling data protection

After setting ProjectProtection in the user's project, the user may soon make requests such as Alice applies to the user for exporting the data of a table out of the user’ s project.

Moreover, user review confirms that this table does not contain sensitive data. In order not to affect Alice's normal business requirements, MaxCompute provides two data export methods to the user after setting ProjectProtection.

- Set TrustedProject

  In case, the current project space is protected, and if you set the target space for the data inflows to the trustedproject for the current space. Then, the data flow to the target project space will not be considered a violation of the project protection

rule. If multiple project spaces are set to trustedproject between two and one another, so these project spaces form a trustedproject.

Group; the data can flow within the project group, but restricted to be shared out of the project group.

Use the following command to manage the TrustedProject:

```
    list  trustedpro  jects ;
     --  View  All  trustedpro  jects  in  the  current
project
    add  trustedpro  ject  < projectnam  e >;
     --  Add  a  trustdproj  ect  to  the  current  project
    remove  trustedpro  ject  < projectnam  e >;
     --  Remove  a  trustdproj  ect  from  the  current
project
```

- Resource sharing and data protection

  In MaxCompute, the package-based resource sharing feature and the project protection data protection feature are orthogonal, but they are similar to each other in terms of functions.

  MaxCompute rules give priority to resource sharing over data protection. Therefore, if a data object allows access by users from other projects through resource sharing, the ProjectProtection rules will not apply to this data object.

Best practices

To prevent data outflow from the project, after setting `ProjectPro tection = true`, check the following settings:

- Make sure the trustedproject is not added. If set, you must assess possible risks;
- Make sure that package data is not used for sharing. If set, make sure that no sensitive data exists in the package.

2.8 Security command list

2.8.1 Security configuration of a project

This article introduces you to the concept of authentication configuration and data protection in some project space security configurations.

Authentication configuration

Statement	Description
show SecurityConfiguration	View the security configuration of the project.
set CheckPermissionUsingACL=true/false	Enable/Disable the ACL-based authorization.
set CheckPermissionUsingPolicy=true/false	Enable/Disable the policy authorization.
set ObjectCreatorHasAccessPermission=true/false	Grant/Revoke default access permissions to/from object creators.
set ObjectCreatorHasGrantPermission=true/false	Grant/Revoke default authorization permissions to/from object creators.

Data protection

Statement	Description
set ProjectProtection=false	Disable data protection.
list TrustedProjects	View the list of trusted projects.
add TrustedProject <projectName> `< projectNam e >`	Add a trusted project.
remove TrustedProject `< projectNam e >`	Remove a trusted project.

2.8.2 Manage permissions

This article introduces you to the related concepts of user management, role management, ACL authorization, and permission review in project space rights management.

Manage users

Statement	Description
list users	View all users added to the project.
add user <username> `< username >`	Add a user.
remove user <username> `< username >`	Remove the user.

Manage roles

Statement	Description
list roles	View all created roles.
create role <rolename> `< rolename >`	Create a role.
drop role <rolename> `< rolename >`	Delete a role.
grant `< rolelist >` to `< username >`	Assign one or multiple roles to the user.
revoke `< rolelist >` from `< username >`	Revoke a role from the user.

ACL Authorization

Statement	Description
grant `< privList >` on `< objType >` `< objName >` to user `< username >`	Authorize a user.
grant `< privList >` on `< objType >` `< objName >` to role `< rolename >`	Authorize a role.
revoke `< privList >` on `< objType >` `< objName >` from user `< username >`	Revoke user authorization.
revoke `< privList >` on `< objType >` `< objName >` from role `< rolename >`	Revoke role authorization.

Permission review

Statement	Description
whoami	View current user information.

Statement	Description
show grants [for < username >] [on type < objectType >]	View user role and permissions.
show acl for < objectName > [on type < objectType >]	View specific object authorization information.
describe role < roleName >	View role authorization information and role assignments.

2.8.3 Package-based resource sharing

This article gives you a description of resource sharing statements based on Package.

Share resources

Statement	Description
Create package <pkgname> < pkgName >	Create a package.
Delete package <pkgname> < pkgName >	Delete a package.
add < objType >< objName > to package < pkgName > [with privileges privs]	Add resources to be shared to a package.
remove < objType >< objName > from package < pkgName >	Remove shared resources from a package .
allow project < prjName > to install package < pkgName > [using label < num >]	Allow a project to use a user package.
disallow project < prjName > to install package < pkgName >	Disallow a project from using a user package.

Use Resources

Statement	Description:
Install package <pkgname> < pkgName >	Install a package.
uninstall package < pkgName >	Uninstall a package.

View a package

Statement	Description:
show packages	List all created and installed packages.
describe package < pkgName >	View details of a package.

 **Note:**

If you execute a production plan authorization related request in DataWorks:

1. **Project owner is executed by temporary query and cannot be submitted to the production environment for execution.Because the production environment is executed by the production account, which has no authorized authority.**
2. **Add the `use < production project >;` statement before the query and submit it with the command. Because DataWorks data development defaults the current project is the development project ending in _dev. When executing authorization commands from the command line, ask project owner to execute the below command first:**

```
use project_na me ;
```

3 Security management use cases

3.1 Create a project

This topic uses two basic services as examples to describe how to create and manage a project. Before you create and manage a project, we recommend that you read Security management and Target users to learn about the security models of MaxCompute and DataWorks.

Create an ETL project

Scenario

In this scenario, multiple users work together as members in an extract, transform , and load (ETL) project. This project involves development, debug, and publish procedures.

Benefits

- DataWorks enables multiple users to work together in one project.
- DataWorks provides basic roles such as `Project Manager`, `Developmen t`, `O & M`, `Deployment`, and `Visitor`, which can be assigned to members to help divide responsibilities.
- DataWorks enables you to create and distinguish between development and production projects. This helps to manage the permissions to view production data and ensures that each project goes through development, debug, and publish procedures.

Procedure

1. Create a project.

   For details about how to create the project, see Create a workspace. The following figure shows the parameter settings for the project.

   - If you set Mode to Developmen t and Production Environmen ts , one DataWorks workspace is bound to two MaxCompute projects: one development project and one production project.
   - The Identity to Access MaxCompute for the development project is Private Account . The project members use their private accounts to compile and debug code.
   - The Identity to Access MaxCompute for the production project is Workspace Owner . This is to ensure that the production project runs smoothly and securely and to limit the permissions of the project members to submit jobs, delete production tables, and modify project data.
2. Add members to the development project.

   Add members to the development project and assign roles to the members in DataWorks. The system automatically assigns roles to RAM users in the development project. The following are the roles available:

   - Project Manager

     A user with the Project Manager role in DataWorks has all the permissions of the Developmen t and O & M roles and can operate the project such as adding members, deleting members, and assign custom resource groups

to roles. This user is also assigned the `role_proje ct_admin` role in MaxCompute.

- `Developmen t`

  A user with the `Developmen t` role in DataWorks can design UIs for compiling code and maintain workflows in Data Analytics. This user is also assigned the `role_proje ct_dev` role in MaxCompute.

- `O & M`

  A user with the `O & M` role in DataWorks can manage all tasks in Maintenance Center. In MaxCompute, this user is also assigned the `role_proje ct_pe` role.

- `Deployment`

  A user with the `Deployment` role in DataWorks can review code and decide whether to submit the code to users with the `O & M` role. This user is also assigned the `role_proje ct_deploy` role in MaxCompute.

- `Visitor`

  A user with the `Visitor` role in DataWorks can only view workflows and code in Data Analytics. In MaxCompute, this user is also assigned the `role_proje ct_guest` role.

- `Safety Manager`

  A user with the `Safety Manager` role in DataWorks has only the Data Security Guard permission. In MaxCompute, this user is also assigned the `role_proje ct_securit y` role.

3. Run a task for debugging code.

   Log on to the DataWorks console as a member with the `Developmen t` role. Then navigate to Data Analytics and debug your code. If required, you can apply for the permissions for production tables in Data Analytics.

4. Publish the task to the production project.

   Package the task, and ask a user with the `O & M` role to review your code. You need to personally notify this user of the code review request. After reviewing your code, this user packages the task and publishes it to the production project only upon approval. For more information, see Publish a task.

5. Test the production task.

   After your task is published to the production project, navigate to Maintenance Center and test your task as a member with the `Developmen t` role. If the task is executed, view logs to check whether the task execution is successful. Furthermore, you can view the result tables in Data Analytics to check whether output data is properly generated. By default, private accounts do not have the permissions for the tables that are generated in the production project. If your private account requires the permissions, you can navigate to Data Management to apply for them.

Note:

- DataWorks enables multiple users to compile code in Data Analytics. All the members in the development project can view the code. Some members can even edit the code after they obtain the edit permission. As a result, some crucial, security-sensitive code has the potential risk of being leaked. We recommend that you group confidential tasks and data into a separate project, on which only the specified users can operate.
- In the production project, only the `project owner` account has the permissions to create tables, functions, and resources in MaxCompute. As a result, you may find that you create a table but the table owner is not your private account, or that you do not have the permissions to view the tables that you create.
- The development and production projects share one `project owner` account. Do not publish a task to the production project, read and write the production tables into the development project, and then obtain production data from the development project.

Create a project in Single Environment mode

Scenario

This project provides a limited number of services, for which the same roles are used . No new services will be added to the project in the future. For example, a carrier only wants to obtain data for analysis and does not need to compile code. In this example, the carrier requires only the query and download services for obtaining data from other projects.

Prerequisites

- The owner of this project is the same as the owner of the development or production project from which data is to be obtained.
- The `Identity to Access MaxCompute` for this project is set to `Private Account`, so that each member can use their private accounts to query and download data.
- Permissions are properly defined for the default role that is assigned to each member of this project in DataWorks after the `Identity to Access MaxCompute` is set to `Private Account`. This is to enable each member to have only the permissions to operate their own tables.

Procedure

1. Create a project.

   For details about how to create the project, see Create a workspace. The following figure shows the parameter settings for this project.

2. Create MaxCompute custom roles and grant permissions to them by using the `project owner` account.

   For more information, see Client.

```
create role custom_dev ;-- Create a custom role .
grant List , CreateInst ance , CreateTabl e , CreateFunc
tion , CreateReso urce on project prj_name to role
custom_dev ;-- Grant permission s to the custom role .
```

3. Enable `Allow object creators to access objects` for the project in MaxCompute by using the `project owner` account.

```
set ObjectCrea torHasAcce ssPermissi on = true ; -- This
parameter is set to true by default . To view the
  parameter setting , run the following command :
show SecurityCo nfiguratio n ;
```

   Alternatively, navigate to MaxCompute Management, and enable Allow object creators to access objects in Basic Settings.

4. Add members to the project.

   Add RAM users as members in DataWorks. For example, after you add a member with the `Developmen t` role in DataWorks, this member is assigned the `role_proje ct_dev` role in MaxCompute. To view the members in the project,

run the `show grants for ram$Alibaba Cloud Account:RAMUser;` command by using the `project owner` account.

5. Modify the permissions of new members in MaxCompute by using the `project owner` account.

```
revoke role_proje ct_dev from ram$Alibaba Cloud
Account:RAM User; -- Remove a new member from its
default role.
grant custom_dev to ram$Alibaba Cloud Account:RAM
User; -- Assign a custom role to a new member.
```

Note:

- If you assign a member with its default role in DataWorks again after you remove this member from its default role, the `role_proje ct_dev` role in MaxCompute is also assigned to this member.
- Each member can view only their own tables (objects). However, each member can view their own tasks in addition to the tasks that are created by other members.
- The members in this project can query the tables from other projects only after they apply for the permissions in Data Management in DataWorks. Alternatively, you can add these tables to a package, install the package in this project, and then grant the package to the members. For more information, see Manage users, roles, and permissions.

3.2 Grant packages

This topic describes how to grant packages to service analysis personnel, so that these personnel can be granted the corresponding permissions to operate on tables of multiple production projects all at once.

Scenario

Service analysis personnel require to view production tables, but often may not have the corresponding permissions. In such scenarios, you can create packages for multiple projects separately and add the tables that can allow service analysis personnel to view the packages. Specifically, you can create an independent analysis project. Then, install the packages in the analysis project, and grant the packages to service analysis personnel. This method can reduce the cost of management

because service analysis personnel do not need to be added to all production projects . Service analysis personnel can view only the tables specified in the packages that are installed in the analysis project.

Procedure

1. Create packages in production projects.

```
CREATEPACK AGE PACKAGE_NA ME ;
For example :
CREATEPACK AGE prj_prod2b i ;
```

2. Add resources to be shared to the packages in the production projects.

```
ADD table TO PACKAGE [ Package name ];
For example :
ADD table adl_test_t able TO PACKAGE prj_prod2b i ;
```

3. Create an independent analysis project.

```
ALLOW PROJECT [ Project in which packages can be
installed ] TO INSTALL PACKAGE [ Package name ];
For example :
ALLOW PRJ_BI TO INSTALL PACKAGE prj_prod2b i ;
```

4. Install the packages in the analysis project.

```
INSTALLPAC KAGE [ Applicatio n name ].[ Package name ];
For example :
INSTALLPAC KAGE prj_prod . prj_prod2b i ;
```

5. Grant the packages to specified users.

```
Grant the package to a user :
GRANTreado npackage prj_prod2b i TOUSER [ Cloud account ];
Grant the package to a role :
GRANTreado npackage prj_prod2b i TOROLE [ Role name ];
```

3.3 Check data security

This topic describes how to check the security of your data and the adjustments that you can make for better data security.

Background information

Often when a project is initially created, its users and permissions may be loosely managed so to expedite the project progress. However, as the project matures, data security becomes an increasingly important aspect of the management of the project . To ensure better data security, we recommend that you check the security of your data and thereafter formulate a data security plan accordingly.

Methods

1. Calculate the number of accounts in your DataWorks projects and in MaxCompute projects. Also, make sure that each member or user has only one RAM user account so that the operations performed by each member or user can be tracked and managed more easily.
2. Calculate the number of accounts that have been discarded and the permissions of these accounts.

   If a RAM user account has a role in a MaxCompute or DataWorks project, the account must be unbound from the role and then deleted from its workspace. If you do not do so, the account is displayed as `p4_xxxxxxx xxxxxxxxxx xxx`, which means that the account cannot be removed from the workspace (even though the workspace stills runs properly).

   If the RAM user account of a member or user changes due to role changes, the account and its permissions must be recycled. We recommend that, after you survey account usage and notify the involved users, you delete or recycle short-term accounts and the accounts that remain inactive for an extended period of time.
3. Survey and analyze the data retrieval and computing tasks (most of which are SQL tasks) that are submitted by RAM user accounts within the last three months. Specifically, identify which accounts submit the most tasks and analyze the tasks submitted by specific accounts.

   For example, the account owned by a member occupies a position in an algorithm development project, and this member executes more SQL tasks for querying and writing tables than it executes algorithm tasks and MapReduce tasks. Based on this fact, the system preferentially calls SQL to process data for this member.

   In another example, an account submits a large number of tasks. After a thorough survey and analysis, the user who owns this account is found to be designing an application with the Software Development Kit (SDK). Multiple users can use this user's Access Key (AK) to query data by using this application. However, such behavior is forbidden.
4. Calculate the number of tasks for downloading data from each project, and plan the projects from which data can be downloaded.

Adjustments

- Allocate accounts properly

  Each member or user can have only one RAM user account, which is properly allocated. For example, the account is allocated based on service groups such as management, data integration, data model, algorithm, analysis, O&M, and security groups.

  Members and users are granted data access permissions according to their groups and roles, and their accounts cannot be shared. This is to avoid data security risks that may be incurred by improperly managed permissions.

- Manage the flow of data

  The permissions of members or users to export data from projects must be overseen and managed. For example, you can restrict the flow of data to only specified projects or locations. We recommend that you restrict the unlimited flow of data among projects because it may interrupt the Alibaba Cloud data architecture and cause data leakage.

- Limit data exporting

  Roles must be divided and bound to service groups properly, so that only users in specified groups can export data as files. Data is no longer in your control once it is exported as files from MaxCompute.

3.4 Manage permissions by row

This topic describes how to manage permissions by row. This can allow you to enable specific users to only view specific data.

Example scenario

Project A has a table named `table_orde r` . This table contains information about the transaction orders of all merchants. Each merchant can view only their own transaction orders.

Solutions

The `table_orde r` table contains merchant IDs, based on which the system can filter transaction orders. To enable each merchant to view their own transaction orders, the system must be able to manage permissions on the row

level. MaxCompute provides the following two solutions to row-level permission management:

- Solution 1: Create an independent downstream table for each merchant in the *table_orde r* table and grant the permissions for the independent table to the corresponding merchant. In this solution, duplicate data may be stored. Therefore, when the *table_orde r* table is updated, its downstream tables must also be updated to ensure data consistency.
- Solution 2: Create an independent downstream view for each merchant in the *table_orde r* table and grant the permissions for the view to the corresponding merchant.The second solution is superior to the first solution in the regard that it does not incur duplicate data, therefore we recommend that you use the second solution.

To use the second solution, take these steps:

1. Create a view for each merchant in project A.

```
CREATE VIEW < viewname > as select * from table_orde r
WHERE sellerid =' xxxx ';
```

2. Create a package for each view in Project A and share the resources in this package to grant the merchant the viewing permissions for these resources.

```
create package < packagenam e >;
add table < viewname > to package < packagenam e >;
allow project < Projectnam e_seller > to install package
< packagenam e >;
```

3. Allow each merchant to be able to use their view.

```
-- All commands are run the project for the merchant
.
install package < ProjectA >.< packagenam e >;
grant read on package < ProjectA >.< packagenam e > to
user < username >;-- The username is the account that
requests to query a view in the project .
```

Note:

You can also grant the select and describe permissions for a view to the corresponding merchant by using an ACL as follows:

```
grant select , describe on table < viewname > to user <
username >;
```

4 MaxCompute Manager

When you start MaxCompute pre-payment, you will encounter one common problem : you have purchased 150 CUs, however, many of your tasks in pre-paid projects may still have to queue up for a long time. Administrators or operations want to know which tasks have occupied resources, so as to control their tasks properly, such as adjusting the scheduling time according to the corresponding business priority of tasks.

MaxCompute Manager provides pre-payment computing resource monitoring and management. Currently, MaxCompute Manager mainly provides three functions: system status monitoring, resource group allocation, and task monitoring. See the DataWorks document MaxCompute Manager for detailed instructions.

 Note:

MaxCompute Manager prerequisite:

- You should already have purchased MaxCompute subscribe CU resources and a quantity of 60 CUs or more. You can only take complete advantage of computing resources and MaxCompute Manager when you have sufficient CUs.