ALIBABA CLOUD

Alibaba Cloud

Hologres

Quick Start

Document Version: 20220125

Alibaba Cloud

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 because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should 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 this document based on the "status quo", "being defective", and "existing functions" of its products and services. 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 take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, 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 contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, 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 this document 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 directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
Danger	A danger notice 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.
Warning	A warning notice 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 restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	**Notice:** If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	**Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Procedure to use Hologres

This topic describes how to use Hologres.

Hologres is a real-time data warehousing service designed for enterprise-level analysis and service integration. Hologres provides real-time data writes and updates, as well as queries of petabytes of data.

If you are new to Hologres, we recommend that you read the following topics:

- What is Hologres?: provides an overview of Hologres and describes its features and use scenarios.
- Billing methods: introduces the pricing and billing methods of Hologres.

The following flowchart shows the procedure to use Hologres.

1. Create an Alibaba Cloud account: Before you use Hologres, you must have an Alibaba Cloud account ready. This topic describes how to create an Alibaba Cloud account and create an AccessKey pair for the account.
2. Purchase a Hologres instance: After you have created an Alibaba Cloud account, you must purchase a Hologres instance. This topic describes how to use your Alibaba Cloud account to purchase a Hologres instance.
3. Create a database: After you have purchased a Hologres instance, you must create a database. This topic describes how to create a database in the Hologres console or from the PostgreSQL client.
4. Connect to the development tool: After you have created a database, you must connect to the development tool to develop and manage the database. This topic describes how to develop and manage the database in the HoloWeb console.
5. Query data in Hologres: This topic describes how to import and query data in foreign tables and internal tables in Hologres.

2.Create an Alibaba Cloud account

This topic describes how to create an Alibaba Cloud account and create an AccessKey pair for the account.

Procedure

1. Create an Alibaba Cloud account

   Go to the international site (alibabacloud.com) and click **Free Account** in the upper-right corner. On the page that appears, enter the required information to create an Alibaba Cloud account.

   > **Notice**
   > - The Alibaba Cloud account owner has full operational control over all purchased resources. To keep your account safe, we recommend that you regularly update your password and do not share your account with others.
   > - You can purchase and connect to a Hologres instance only by using your Alibaba Cloud account.

2. Complete real-name verification.

   Go to the Real-name Registration page in the Account Management console and complete the verification.

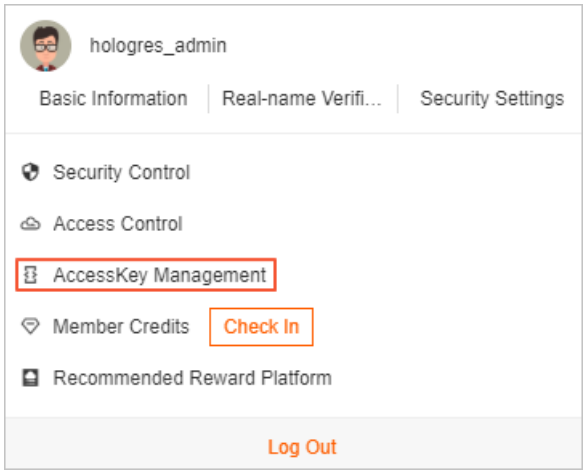
   > **Note** You must complete real-name verification before you purchase and use Alibaba Cloud services.

3. Create an AccessKey pair.

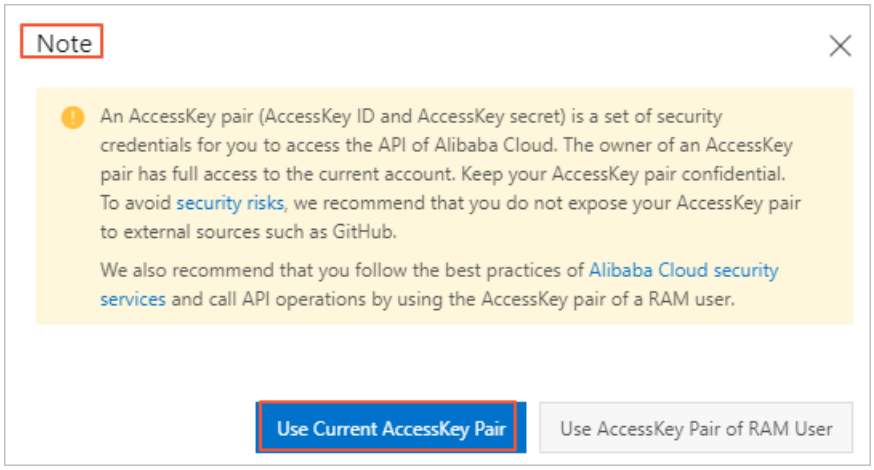
   To analyze data by using Hologres, you must create an AccessKey pair for your Alibaba Cloud account. Unlike the username and password that you use to log on to the Alibaba Cloud Management Console, an AccessKey pair is used to connect Hologres to development or BI tools. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. To create an AccessKey pair, perform the following steps:

   > **Notice** After an AccessKey pair is created for your Alibaba Cloud account, keep the AccessKey ID and AccessKey secret safe. If the AccessKey ID or AccessKey secret is or may be disclosed, change the AccessKey pair in a timely manner.

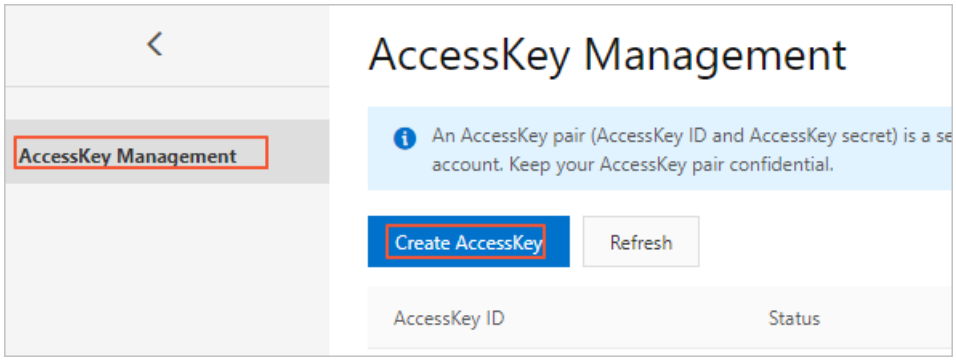
i. Log on to the Hologres console. Move the pointer over your profile picture in the upper-right corner and click **AccessKey Management**.



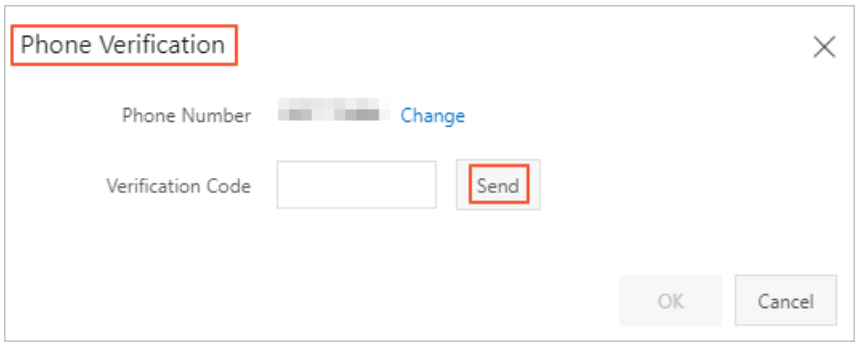
ii. In the **Note** message, click **Use Current AccessKey Pair**.



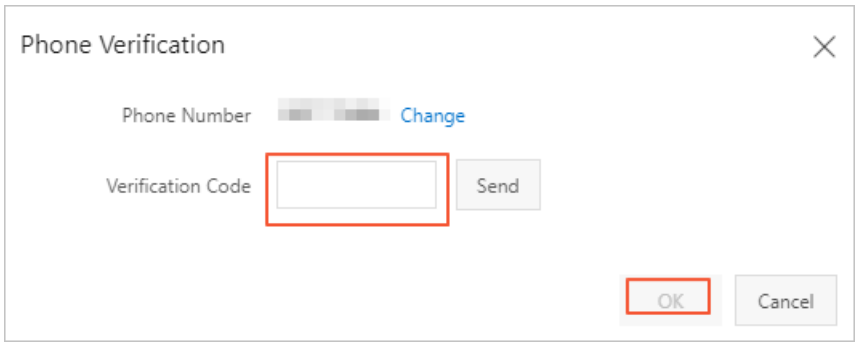
iii. On the **AccessKey Management** page, click **Create AccessKey**.



iv. In the **Phone Verification** dialog box, click **Send**.



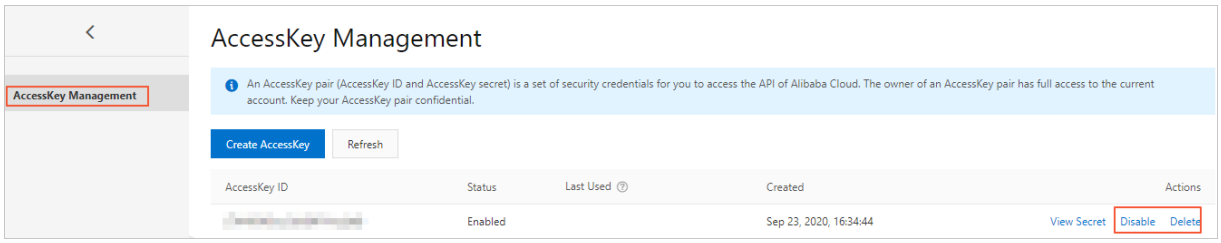
v. Enter the verification code and click **OK**.



vi. In the **Create AccessKey** message, view the AccessKey pair that you created.

Result

After you close the Create AccessKey message, view the status of the AccessKey pair on the **AccessKey Management** page. You can also click **Disable** or **Delete** in the Actions column to disable or delete the AccessKey pair.



Note If you disable the AccessKey pair, all the services that use the AccessKey pair fail to be run and report errors. Therefore, after you change your AccessKey pair, you must check the running status of services that use the AccessKey pair and recover the services in a timely manner.

3.Purchase a Hologres instance

Before you use Hologres, you must purchase a Hologres instance. This topic describes how to purchase a Hologres instance by using an Alibaba Cloud account.

Prerequisites

- An Alibaba Cloud account is created.
- Real-name verification is complete.

Context

By default, the system sets the Alibaba Cloud account that is used to purchase an instance as a superuser of the instance. The superuser has all permissions on the instance.

A RAM user must be authorized by the Alibaba Cloud account before the RAM user can purchase instances. For more information, see Grant permissions on Hologres to RAM users. A RAM user can follow the same procedure as an Alibaba Cloud account to purchase an instance.

Procedure

1. Go to the Hologres product page.
2. Click **Buy Now**.
3. Select a billing method and set the parameters.
4. Click **Buy Now**.
5. On the **Confirm Order** page, verify the information about the purchased instance, such as the billing method, instance name, resources, and region. Read and agree to the terms of service by selecting the check box.
6. Click **Pay**.
7. Complete the payment on the **Purchase** page.

   After an instance is purchased, go to the Hologres console to view the instance information. For more information about how to use the Hologres console, see Overview.

   Hologres is compatible with PostgreSQL. You can connect to a Hologres instance from the PostgreSQL client for data analytics. You can also use extract, transform, load (ETL) or business intelligence (BI) tools to connect to a Hologres instance.

   - For more information about the PostgreSQL client, see PSQL quick start.
   - For more information about common development tools, see Overview.

4.Create a database

This topic describes how to create a database in the Hologres console or from the PostgreSQL client.

Prerequisites

A Hologres instance is purchased. For more information, see Purchase a Hologres instance.

Context

After you purchase a Hologres instance, a database named **postgres** is automatically created. This database is allocated with a few resources and is used only for management purposes. Therefore, we recommend that you create another database to process business data.

Only a superuser or a user granted the database creation permission can create a database.

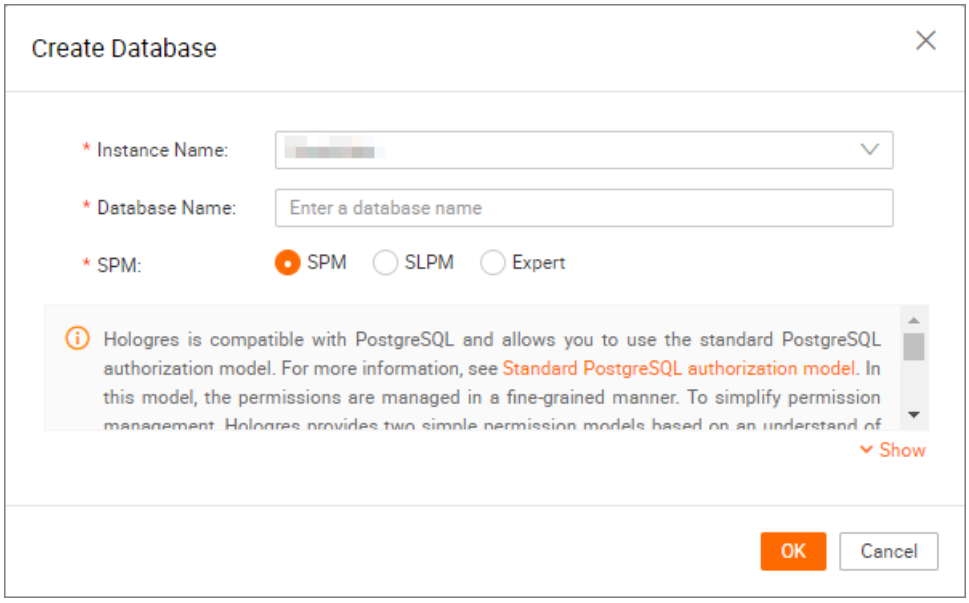
Create a database in the Hologres console

1. Log on to the Hologres console. In the left-side navigation pane, click **Instances**.
2. On the **Hologres Instances** page, find the instance for which you want to create a database and click the instance name.

   You can also find the instance for which you want to create a database and click **Manage** in the **Actions** column to go to the instance details page.
3. In the left-side pane of the instance details page, click **Databases**.
4. On the **Database Authorization** page, click **Create Database** in the upper-right corner.
5. In the **Create Database** dialog box, select the name of the instance for which you want to create the database from the **Instance Name** drop-down list, enter a database name in the **Database Name** field, and then specify a permission model based on your business needs by setting the **SPM** parameter. We recommend that you select **SPM**.

Permission model	Description
SPM	If the simple permission model (SPM) is used, permissions are granted at the database level. The following user groups are provided: admin, developer, writer, and viewer. You need only to use a few functions to manage the permissions on the objects in a database in a convenient and secure way. For more information, see Overview.
SLPM	If the schema-level permission model (SLPM) is used, permissions are granted at the schema level. The following user groups are provided: <db>.admin, <db>.<schema>.developer, <db>.<schema>.writer, and <db>.<schema>.viewer. Compared with the SPM, the SLPM manages permissions in a more fine-grained manner. For more information, see Overview.

Permission model	Description
Standard PostgreSQL authorization model	Hologres allows you to use the standard PostgreSQL authorization model. For more information, see Standard PostgreSQL authorization model.



6. Click **OK**.

   You can view the created database on the **Database Authorization** page.

Create a database from the PostgreSQL client

1. Connect to the Hologres instance for which you want to create a database from the PostgreSQL client. For more information, see Connect to a Hologres instance from the PostgreSQL client.
2. Execute the CREATE Database statement to create a database. The following code provides the syntax and a sample statement:

```
CREATE Database NewDatabaseName;
CREATE Database test; // Create a database named test.
```

3. Run the `\l` command to view the databases in the instance.
4. Run the `\c NewDatabaseName` command to connect to the newly created database. When you run this command, replace *NewDatabaseName* with the name of the created database.

What's next

Execute standard PostgreSQL statements to analyze data from the PostgreSQL client. For example, you can execute SQL statements to import MaxCompute data to Hologres. For more information, see Import data from MaxCompute to Hologres by executing SQL statements.

You can also use HoloWeb to analyze data. For more information, see HoloWeb quick start.

5.HoloWeb quick start

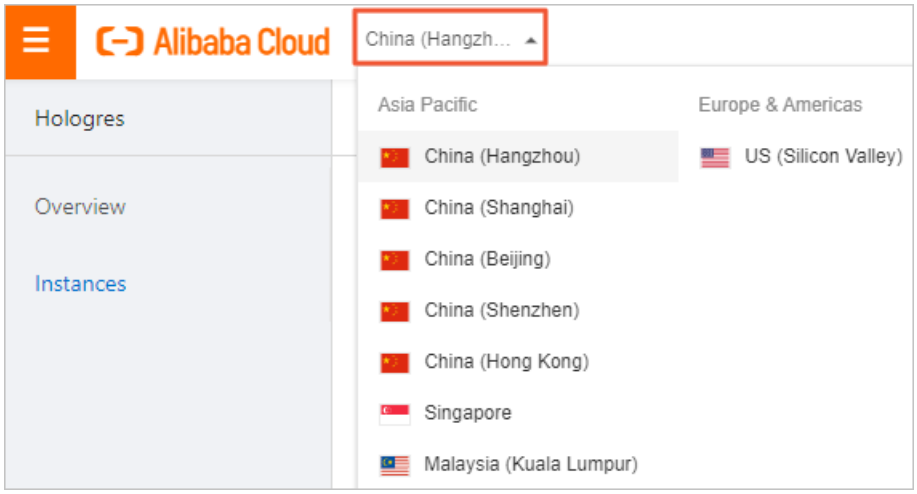
HoloWeb is a comprehensive platform that is built based on Hologres. HoloWeb allows you to analyze data and manage databases in a visualized manner. This topic describes the basic operations that you can perform in the HoloWeb console.

Prerequisites

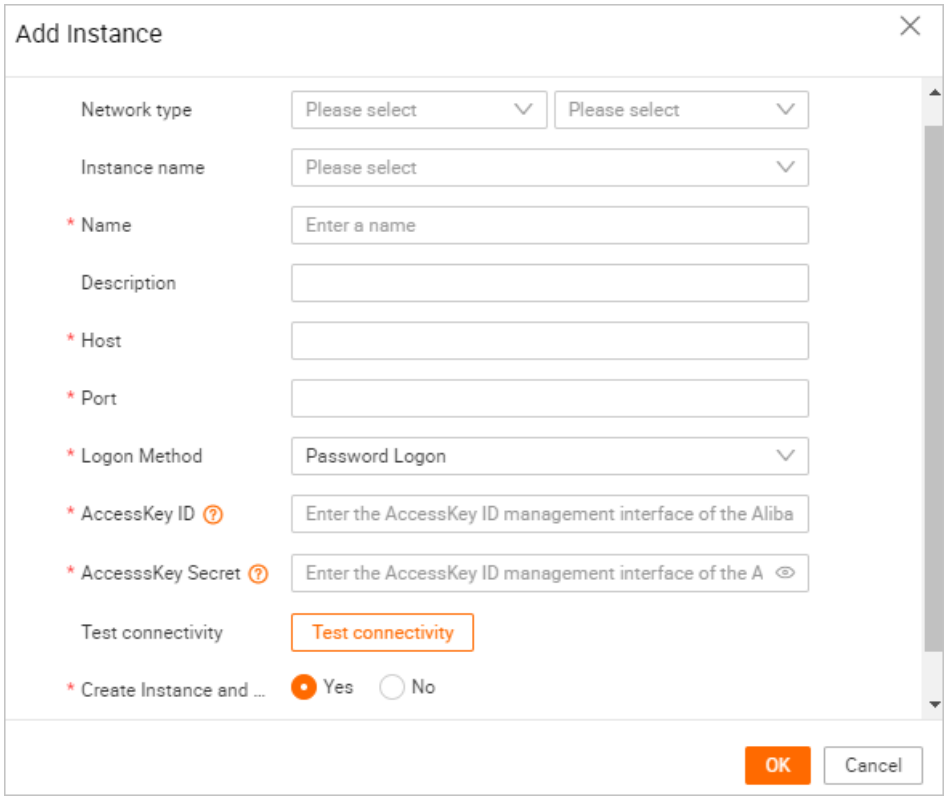
- An Alibaba Cloud account is created.
- The real-name verification is complete.
- A Hologres instance is purchased. For more information, see Purchase a Hologres instance.

Procedure

1. Log on to the Hologres console.
2. In the top navigation bar, select a region from the drop-down list.



3. On the **Hologres Instances** page, click **Go to HoloWeb** to go to the HoloWeb console.
4. Connect to a Hologres instance.
   i. On the **Metadata Management** tab, click **Add Instance**.
   ii. In the **Add Instance** dialog box, set the parameters as required and click **OK**.



Parameter	Description	Required
Network type	■ **public network**: Supported regions include China (Shanghai), China (Shenzhen), China (Beijing), China (Hangzhou), Singapore (Singapore), China (Hong Kong), Malaysia (Kuala Lumpur), and US (Silicon Valley). The icon indicates that the network type of the instance is public network. ■ **VPC**: You can select only the region from which you log on to the HoloWeb console. The icon indicates that the network type of the instance is VPC. This instance cannot be edited or removed.	No
Instance name	The Hologres instance that is created by using the current Alibaba Cloud account.	No
Name	If you have set the **Instance name** parameter, the name of the specified instance is automatically entered in the **Name** field. You can also enter a custom connection name.	Yes
Description	The description of the connection.	No

Parameter	Description	Required
Host	The endpoint of the Hologres instance. You can view the endpoint of the Hologres instance on the **Configurations** tab of the instance details page in the Hologres console. If you have set the **Instance name** parameter, the endpoint of the specified instance is automatically entered in the **Host** field. You can also enter the endpoint of the Hologres instance.	Yes
Port	The port number of the Hologres instance. You can view the port number of the Hologres instance on the **Configurations** tab of the instance details page in the Hologres console. If you have set the **Instance name** parameter, the port number of the specified instance is automatically entered in the **Port** field. You can also enter the port number of the Hologres instance.	Yes
Logon Method	■ **Password-free Logon**: You can directly log on to the instance by using the current Alibaba Cloud account, without the need to enter the AccessKey ID and AccessKey secret. ■ **Password Logon**: You can enter the AccessKey ID and AccessKey secret of the current or another Alibaba Cloud account to log on to the instance.	Yes
AccessKey ID	This parameter is displayed only when you set the Logon Method parameter to *Password Logon*. The AccessKey ID of your Alibaba Cloud account. You can obtain the AccessKey ID from the Security Management page.	No
AccessKey Secret	This parameter is displayed only when you set the Logon Method parameter to *Password Logon*. The AccessKey secret of your Alibaba Cloud account. You can obtain the AccessKey secret from the Security Management page.	No

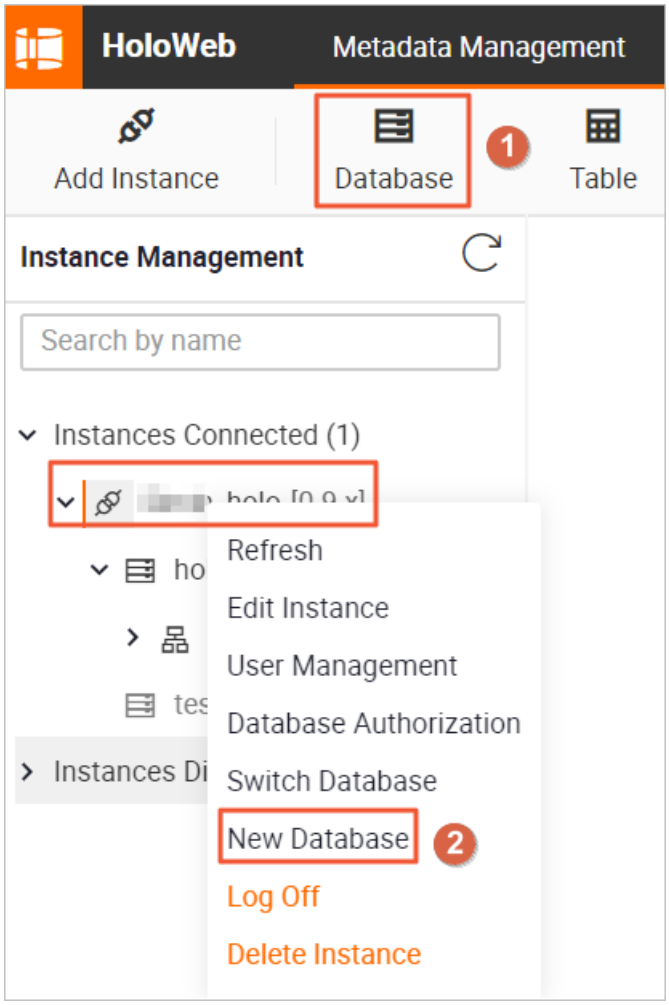
Parameter	Description	Required
Test Connectivity	Tests whether the instance can be connected to. ■ If the instance can be connected to, the message **Test passed** is displayed. ■ If the instance cannot be connected to, the message **Test failed** is displayed.	No
Create Instance and Log On	Specifies whether to log on to the instance. ■ **Yes**: logs on to the instance. The connected instance is displayed in the left-side Instances Connected list. ■ **No**: does not log on to the instance. The disconnected instance is displayed in the left-side Instances Disconnected list.	Yes

5. Create a database.

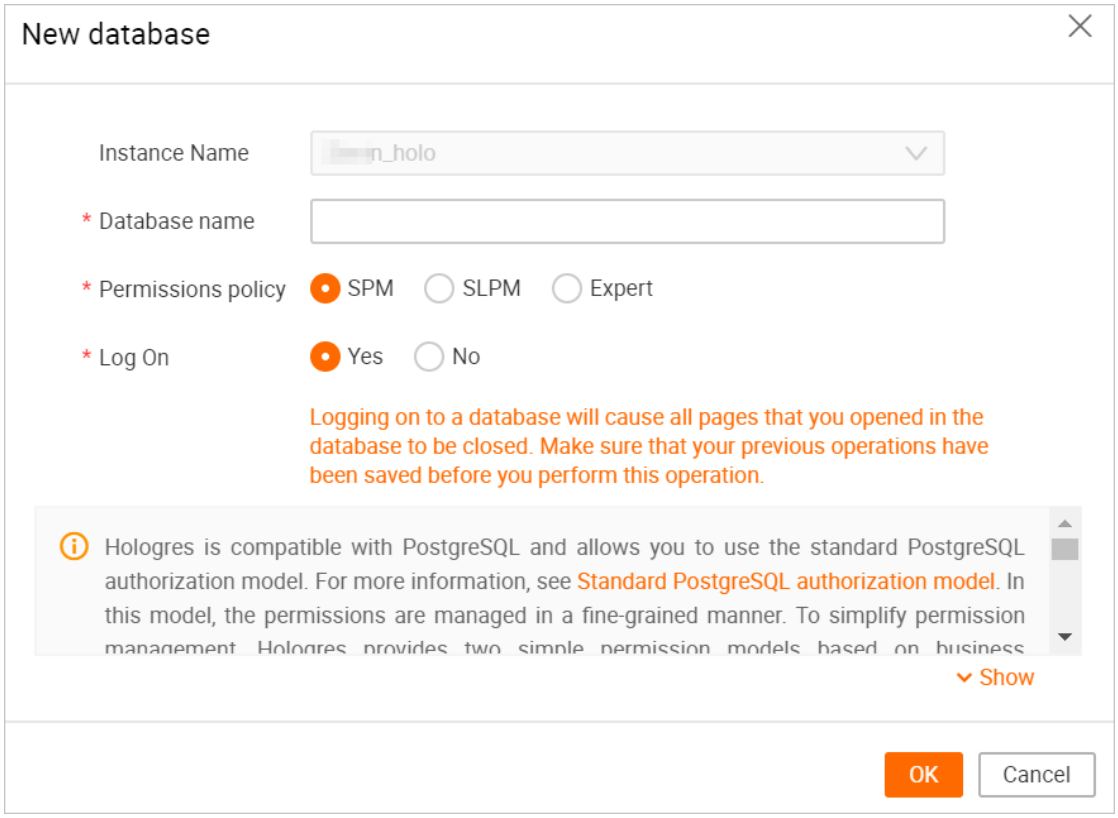
   After you purchase a Hologres instance, a database named **postgres** is automatically created. This database is allocated with a few resources and is used only for management purposes. Therefore, we recommend that you create another database to process business data. If you have created a database, skip this step and proceed to the next step.

   i. On the **Metadata Management** tab, click **Database**.

      You can also find the instance for which you want to create a database in the **Instances Connected** list on the **Metadata Management** tab. Right-click the instance and select **New Database**.



ii. In the **New database** dialog box, set the parameters as required and click **OK**.



Parameter	Description
Instance Name	The name of the instance in which the database is to be created. By default, the name of the current instance is entered.
Database Name	Enter a database name. **Note** The database name must be unique.
Permissions Policy	Select a permission model for the database based on your business requirements. For more information about permission models, see: ■ SPM ■ SLPM ■ Standard PostgreSQL authorization model
Log On	■ Yes: logs on to the database. After the database is created, you can directly use the database. ■ No: does not log on to the database. Before you use the created database, you must log on to the database.

6. Create a query task.

   After you connect HoloWeb to a Hologres instance, you can use standard PostgreSQL statements

to analyze data on the **SQL Editor** tab. To create a query task, perform the following steps:

i. On the **SQL Editor** tab, click **Ad-hoc Query**.

You can also right-click **My SQL query** in the left-side navigation pane and select **New SQL query**. For more information, see Manage an SQL query task.

ii. On the **Ad-hoc Query** tab, select an instance from the **Instance** drop-down list and a database from the **Database** drop-down list, enter the following sample statements in the SQL editor, and then click **Run**.

```
BEGIN;
CREATE TABLE nation (
 n_nationkey bigint NOT NULL,
 n_name text NOT NULL,
 n_regionkey bigint NOT NULL,
 n_comment text NOT NULL,
PRIMARY KEY (n_nationkey)
);
CALL SET_TABLE_PROPERTY('nation', 'bitmap_columns', 'n_nationkey,n_name,n_regionkey
');
CALL SET_TABLE_PROPERTY('nation', 'dictionary_encoding_columns', 'n_name,n_comment'
);
CALL SET_TABLE_PROPERTY('nation', 'time_to_live_in_seconds', '31536000');
COMMIT;
INSERT INTO nation VALUES
(11,'zRAQ', 4,'nic deposits boost atop the quickly final requests? quickly regula')
,
(22,'RUSSIA', 3  ,'requests against the platelets use never according to the quickl
y regular pint'),
(2,'BRAZIL',  1 ,'y alongside of the pending deposits. carefully special packages a
re about the ironic forges. slyly special '),
(5,'ETHIOPIA',  0 ,'ven packages wake quickly. regu'),
(9,'INDONESIA', 2  ,'slyly express asymptotes. regular deposits haggle slyly. caref
ully ironic hockey players sleep blithely. carefull'),
(14,'KENYA',  0  ,'pending excuses haggle furiously deposits. pending, express pint
o beans wake fluffily past t'),
(3,'CANADA',  1 ,'eas hang ironic, silent packages. slyly regular packages are furi
ously over the tithes. fluffily bold'),
(4,'EGYPT', 4 ,'y above the carefully unusual theodolites. final dugouts are quickl
y across the furiously regular d'),
(7,'GERMANY', 3 ,'l platelets. regular accounts x-ray: unusual, regular acco'),
(20 ,'SAUDI ARABIA',  4 ,'ts. silent requests haggle. closely express packages slee
p across the blithely');
SELECT * FROM nation;
```

> **Note** The names of tables and fields in SQL statements are not case-sensitive. To reference a table with an exactly matched name, you must enclose the name of the table in double quotation marks (") in SQL statements.

iii. View the execution result.

You can view the execution result of the statements on the **Run Log** tab.

```
CALL SET_TABLE_PROPERTY('nation', 'bitmap_columns', 'n_nationkey,n_name,n_regionkey');
CALL SET_TABLE_PROPERTY('nation', 'dictionary_encoding_columns', 'n_name,n_comment');
CALL SET_TABLE_PROPERTY('nation', 'time_to_live_in_seconds', '31536000');
COMMIT;

INSERT INTO nation VALUES
(11,'zRAQ', 4,'nic deposits boost atop the quickly final requests? quickly regula'),
(22,'RUSSIA', 3  ,'requests against the platelets use never according to the quickly regular pint'),
(2,'BRAZIL',  1 ,'y alongside of the pending deposits. carefully special packages are about the ironic forges. slyly special '),
(5,'ETHIOPIA',  0 ,'ven packages wake quickly. regu'),
(9,'INDONESIA', 2  ,'slyly express asymptotes. regular deposits haggle slyly. carefully ironic hockey players sleep blithely. carefull'),
(14,'KENYA',  0  ,'pending excuses haggle furiously deposits. pending, express pinto beans wake fluffily past t'),
(3,'CANADA',  1 ,'eas hang ironic, silent packages. slyly regular packages are furiously over the tithes. fluffily bold'),
(4,'EGYPT', 4 ,'y above the carefully unusual theodolites. final dugouts are quickly across the furiously regular d'),
(7,'GERMANY', 3 ,'l platelets. regular accounts x-ray: unusual, regular acco'),
(20 ,'SAUDI ARABIA',  4 ,'ts. silent requests haggle. closely express packages sleep across the blithely');

SELECT * FROM nation;
```

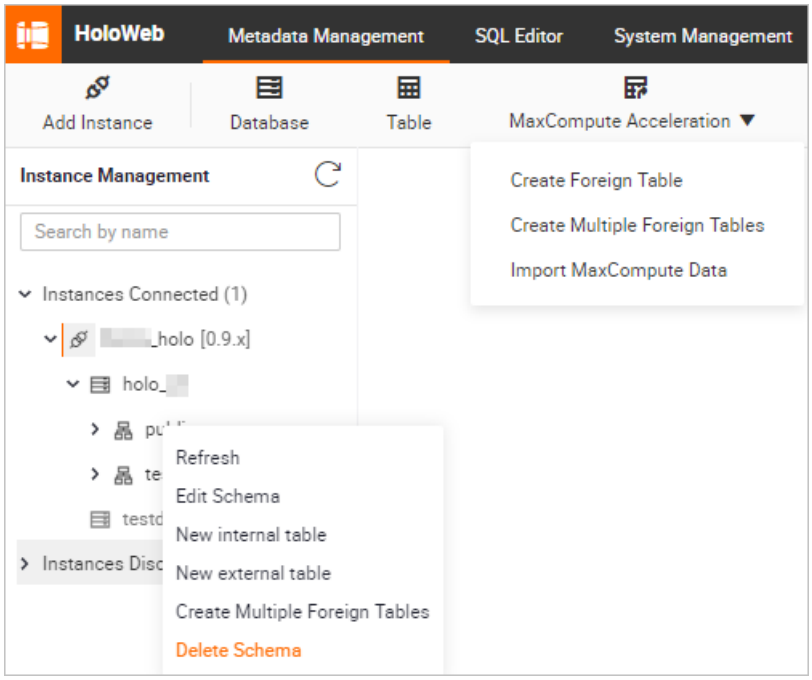
Run Log

	A	B	C	D
1	n_nationkey	n_name	n_regionkey	n_comment
2	11	zRAQ	4	nic deposits boost ato
3	3	CANADA	1	eas hang ironic, silent
4	7	GERMANY	3	l platelets. regular acc
5	20	SAUDI ARABIA	4	ts. silent requests hag
6	2	BRAZIL	1	y alongside of the pen
7	5	ETHIOPIA	0	ven packages wake qu
8	9	INDONESIA	2	slyly express asymptot
9	14	KENYA	0	pending excuses hagg
10	4	EGYPT	4	y above the carefully u
11	22	RUSSIA	3	requests against the pl

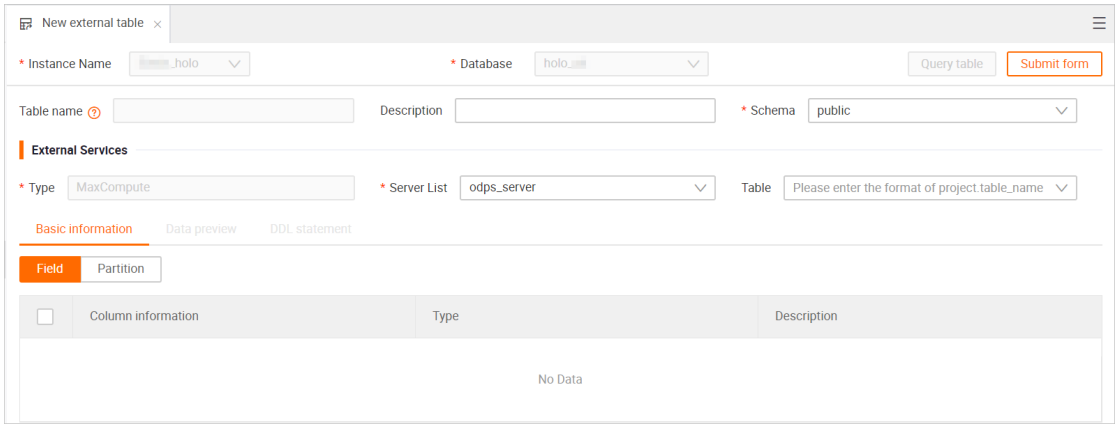
7. Create a foreign table.

Hologres is seamlessly integrated with MaxCompute at the underlying layer. You can create foreign tables in Hologres to accelerate queries of MaxCompute data. To create a foreign table in the HoloWeb console, perform the following steps:

i. On the **Metadata Management** tab, choose **MaxCompute Acceleration > Create Foreign Table**.



ii. On the **New external table** tab, set the parameters as required and click **Submit form**.



Parameter	Description
Instance Name	The name of the current instance.
Database	The name of the current database.
Table name	The name of the Hologres foreign table. After you enter the name of the source MaxCompute table, the name of the foreign table must be the same as that of the source MaxCompute table. In this case, you cannot modify the name of the foreign table. To rename the foreign table, you must create the foreign table, find the foreign table in the left-side Instances Connected list, and then right-click the foreign table.
Description	The description of the Hologres foreign table.
Schema	The name of the schema. You can select the default schema **public** or a custom schema.
Type	The service type of the source table. The default value is MaxCompute, which cannot be changed.
Server List	You can select the **odps_server** server that is created at the underlying layer of Hologres. For more information, see postgres_fdw.

Parameter	Description
Table	The name of the source MaxCompute table to be mapped, including the name of the MaxCompute project. Format: **project.table_name**. **Note** ■ You cannot query data in a source table that is stored in a region different from that of the selected Hologres instance. ■ After you enter the name of the source MaxCompute table, all the fields of the source table are displayed. By default, the created foreign table contains all the fields of the source table. If you need to create a foreign table that contains only specific fields of the source table, use SQL statements to create the foreign table. For more information, see CREATE FOREIGN TABLE.

Note When you create a foreign table to synchronize data from a MaxCompute table, the table comments and column comments of the MaxCompute table are synchronized to the foreign table.

iii. After you create the foreign table, find the instance where the foreign table resides from the **Instances Connected** list in the left-side navigation pane on the **Metadata Management** tab. Double-click the foreign table. On that tab that appears, click **Data preview** to preview the data in the MaxCompute table for which queries are accelerated.

6.Query data in Hologres

This topic describes how to query data in foreign tables and internal tables in Hologres. This helps you experience the performance of data queries in Hologres.

Prerequisites

- A Hologres instance is purchased. For more information, see Purchase a Hologres instance.
- A RAM user that you want to use to perform operations is granted the required permissions. For more information, see Grant permissions to a RAM user.

Context

Hologres can give quick response when you query data. This topic shows you how to create a database, create foreign tables and internal tables, and then import data to the internal tables in Hologres. Then, this topic shows you how to query data in the foreign tables and internal tables in Hologres. This helps you experience data queries in Hologres.

Procedure

1. Step 1: Create a database

   This step shows you how to create a database in Hologres. You can use the database to store the data of your Hologres instance for later queries.

2. Step 2: Create tables

   This step shows you how to create tables in the created database. You can use the tables to store sample data. Hologres allows you to create foreign tables and internal tables. These two types of tables have the following characteristics:

   - Foreign tables map fields in external data sources and do not store data in Hologres.
   - Internal tables store data in Hologres.

3. Step 3: Import sample data

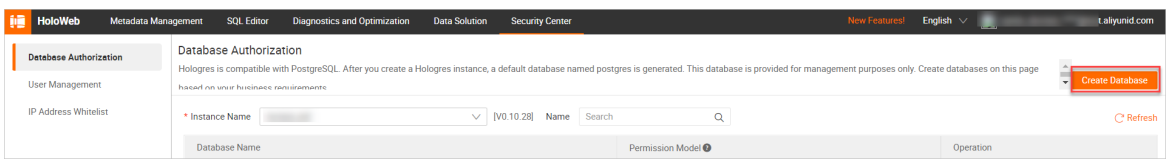
   This step shows you how to import data to internal tables in Hologres for later queries. In this example, the data of the TPC Benchmark™ H (TPC-H) benchmark test is used.

4. Step 4: Query data in tables

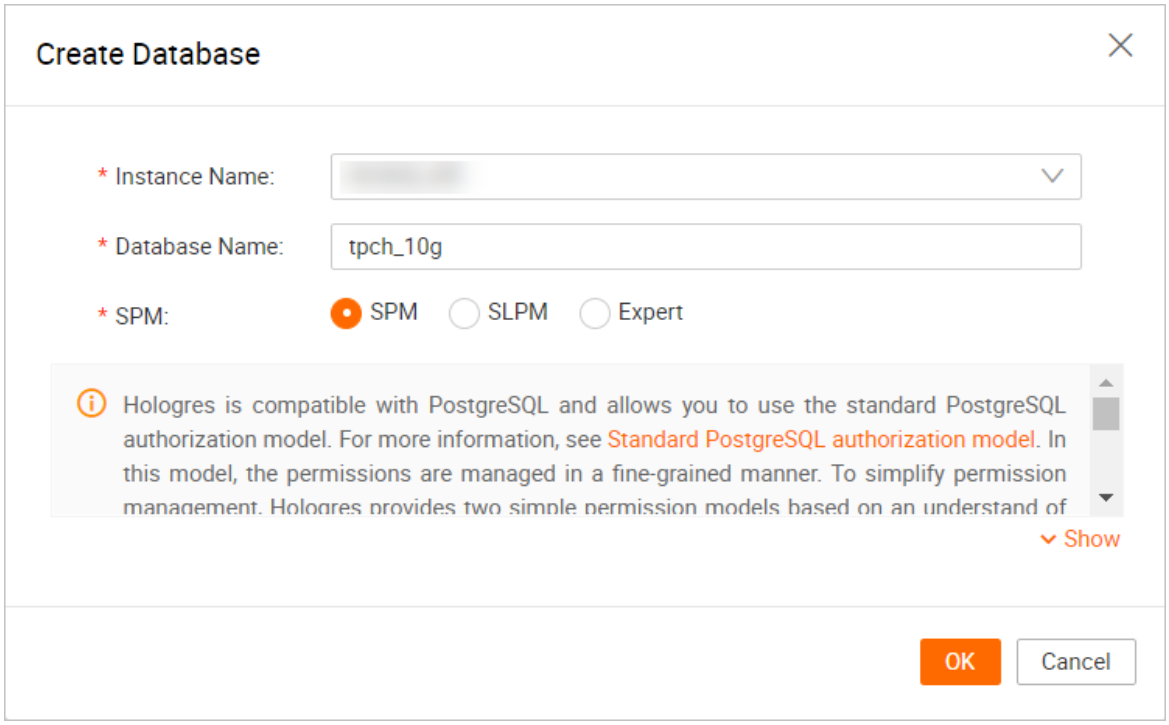
   This step shows you how to query data in tables. In this example, the 22 TPC-H query statements are used to help you experience the timeliness and stability of data queries in Hologres.

Step 1: Create a database

1. Log on to the Hologres console. In the left-side navigation pane, click **Instances**.
2. On the **Instances** page, click the name of your Hologres instance.
3. In the left-side pane of the instance details page, click **Databases**.
4. On the **Database Authorization** page, click **Create Database** in the upper-right corner.



5. In the **Create Database** dialog box, set the parameters that are described in the following table.

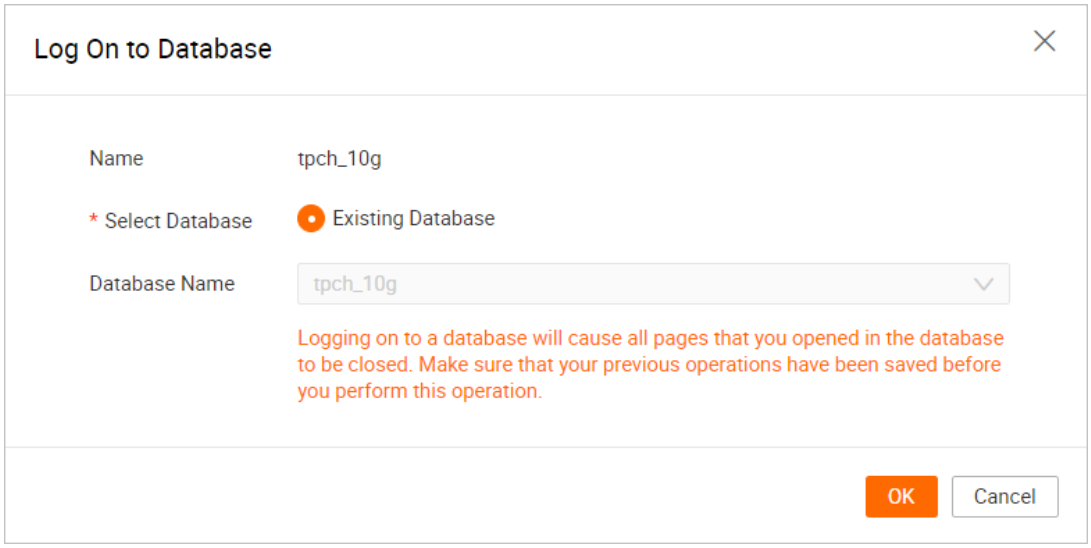


Parameter	Description
Instance Name	The name of the Hologres instance in which you want to create the database. By default, the name of the connected instance is displayed. You can also select another Hologres instance from the drop-down list.
Database Name	The name of the database. In this example, enter tpch_10g. **Note** The database name must be unique.
SPM	The permission model of the database. Valid values: ○ SPM: the simple permission model (SPM). If you select SPM, permissions are granted at the database level. The following user groups are provided: admin, developer, writer, and viewer. You need to use only a few functions to manage the permissions on the objects in a database in a convenient and secure way. ○ SLPM: the schema-level permission model (SLPM). If you select SLPM, permissions are granted at the schema level. The following user groups are provided: <db>.admin, <db>.<schema>.developer, <db>.<schema>.writer, and <db>.<schema>.viewer. Compared with the SPM, the SLPM manages permissions in a finer-grained manner. ○ Expert: the standard PostgreSQL authorization model. Hologres is compatible with PostgreSQL. If you select Expert, you can use the standard PostgreSQL authorization model.

Step 2: Create tables

After the database is created, you can create tables in the database. You can create foreign tables or internal tables based on where your data is stored.

- **Create foreign tables.**
  i. Log on to the database.
     a. On the **Database Authorization** page of the HoloWeb console, click **Metadata Management** in the top navigation bar.
     b. On the **Metadata Management** tab, find the created database in the left-side navigation pane and double-click the database name. In the dialog box that appears, click **OK**.



  ii. Create foreign tables.
     a. Go to the **SQL Editor** tab. Click the **Ad-hoc Query** icon in the upper-left corner.

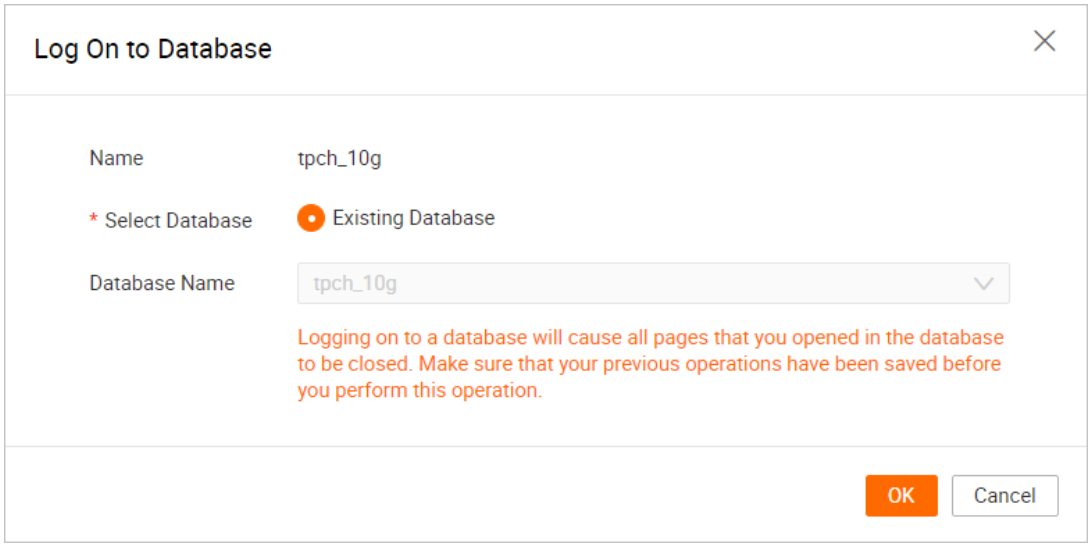
b. On the **Ad-hoc Query** tab, select an instance from the **Instance** drop-down list and a database from the **Database** drop-down list, enter the following sample statements in the SQL editor, and then click **Run**.

The following SQL statements are used to create foreign tables for later queries. The foreign tables map fields in the source tables such as the odps_customer_10g and odps_lineitem_10g tables in the MaxCompute project MAXCOMPUTE_PUBLIC_DATA in the public dataset.

```
DROP FOREIGN TABLE IF EXISTS odps_customer_10g;
DROP FOREIGN TABLE IF EXISTS odps_lineitem_10g;
DROP FOREIGN TABLE IF EXISTS odps_nation_10g;
DROP FOREIGN TABLE IF EXISTS odps_orders_10g;
DROP FOREIGN TABLE IF EXISTS odps_part_10g;
DROP FOREIGN TABLE IF EXISTS odps_partsupp_10g;
DROP FOREIGN TABLE IF EXISTS odps_region_10g;
DROP FOREIGN TABLE IF EXISTS odps_supplier_10g;
IMPORT FOREIGN SCHEMA MAXCOMPUTE_PUBLIC_DATA LIMIT to
(
    odps_customer_10g,
    odps_lineitem_10g,
    odps_nation_10g,
    odps_orders_10g,
    odps_part_10g,
    odps_partsupp_10g,
    odps_region_10g,
    odps_supplier_10g
)
FROM SERVER odps_server INTO public OPTIONS(if_table_exist 'error',if_unsupported
_type 'error');
```

- **Create internal tables.**
  i. Log on to the database.
    a. On the **Database Authorization** page of the HoloWeb console, click **Metadata Management** in the top navigation bar.

b. On the **Metadata Management** tab, find the created database in the left-side navigation pane and double-click the database name. In the dialog box that appears, click **OK**.



ii. Create internal tables.

a. Go to the **SQL Editor** tab. Click the **Ad-hoc Query** icon in the upper-left corner.

b. On the **Ad-hoc Query** tab, select an instance from the **Instance** drop-down list and a database from the **Database** drop-down list, enter the following sample statements in the SQL editor, and then click **Run**.

The following SQL statements are used to create internal tables named LINEITEM, ORDERS, PARTSUPP, PART, CUSTOMER, SUPPLIER, NATION, and REGION. The internal tables are used to store data for later queries.

```
DROP TABLE IF EXISTS LINEITEM;
BEGIN;
CREATE TABLE LINEITEM
(
    L_ORDERKEY      BIGINT      NOT NULL,
    L_PARTKEY       INT         NOT NULL,
    L_SUPPKEY       INT         NOT NULL,
    L_LINENUMBER    INT         NOT NULL,
    L_QUANTITY      DECIMAL(15,2) NOT NULL,
    L_EXTENDEDPRICE DECIMAL(15,2) NOT NULL,
    L_DISCOUNT      DECIMAL(15,2) NOT NULL,
    L_TAX           DECIMAL(15,2) NOT NULL,
    L_RETURNFLAG    TEXT        NOT NULL,
    L_LINESTATUS    TEXT        NOT NULL,
    L_SHIPDATE      TIMESTAMPTZ NOT NULL,
    L_COMMITDATE    TIMESTAMPTZ NOT NULL,
    L_RECEIPTDATE   TIMESTAMPTZ NOT NULL,
    L_SHIPINSTRUCT  TEXT        NOT NULL,
    L_SHIPMODE      TEXT        NOT NULL,
    L_COMMENT       TEXT        NOT NULL,
    PRIMARY KEY (L_ORDERKEY,L_LINENUMBER)
);
CALL set_table_property('LINEITEM', 'clustering_key', 'L_SHIPDATE,L_ORDERKEY');
CALL set_table_property('LINEITEM', 'segment_key', 'L_SHIPDATE');
CALL set_table_property('LINEITEM', 'distribution_key', 'L_ORDERKEY');
```

```
CALL set_table_property('LINEITEM', 'bitmap_columns', 'L_ORDERKEY,L_PARTKEY,L_SUP
PKEY,L_LINENUMBER,L_RETURNFLAG,L_LINESTATUS,L_SHIPINSTRUCT,L_SHIPMODE,L_COMMENT')
;
CALL set_table_property('LINEITEM', 'dictionary_encoding_columns', 'L_RETURNFLAG,
L_LINESTATUS,L_SHIPINSTRUCT,L_SHIPMODE,L_COMMENT');
CALL set_table_property('LINEITEM', 'time_to_live_in_seconds', '31536000');
COMMIT;
DROP TABLE IF EXISTS ORDERS;
BEGIN;
CREATE TABLE ORDERS
(
    O_ORDERKEY      BIGINT         NOT NULL PRIMARY KEY,
    O_CUSTKEY       INT         NOT NULL,
    O_ORDERSTATUS   TEXT        NOT NULL,
    O_TOTALPRICE    DECIMAL(15,2) NOT NULL,
    O_ORDERDATE     timestamptz NOT NULL,
    O_ORDERPRIORITY TEXT        NOT NULL,
    O_CLERK         TEXT        NOT NULL,
    O_SHIPPRIORITY  INT         NOT NULL,
    O_COMMENT       TEXT        NOT NULL
);
CALL set_table_property('ORDERS', 'segment_key', 'O_ORDERDATE');
CALL set_table_property('ORDERS', 'distribution_key', 'O_ORDERKEY');
CALL set_table_property('ORDERS', 'bitmap_columns', 'O_ORDERKEY,O_CUSTKEY,O_ORDER
STATUS,O_ORDERPRIORITY,O_CLERK,O_SHIPPRIORITY,O_COMMENT');
CALL set_table_property('ORDERS', 'dictionary_encoding_columns', 'O_ORDERSTATUS,O
_ORDERPRIORITY,O_CLERK,O_COMMENT');
CALL set_table_property('ORDERS', 'time_to_live_in_seconds', '31536000');
COMMIT;
DROP TABLE IF EXISTS PARTSUPP;
BEGIN;
CREATE TABLE PARTSUPP
(
    PS_PARTKEY    INT    NOT NULL,
    PS_SUPPKEY    INT    NOT NULL,
    PS_AVAILQTY   INT    NOT NULL,
    PS_SUPPLYCOST DECIMAL(15,2) NOT NULL,
    PS_COMMENT    TEXT   NOT NULL,
    PRIMARY KEY(PS_PARTKEY,PS_SUPPKEY)
);
CALL set_table_property('PARTSUPP', 'distribution_key', 'PS_PARTKEY');
CALL set_table_property('PARTSUPP', 'colocate_with', 'LINEITEM');
CALL set_table_property('PARTSUPP', 'bitmap_columns', 'PS_PARTKEY,PS_SUPPKEY,PS_A
VAILQTY,PS_COMMENT');
CALL set_table_property('PARTSUPP', 'dictionary_encoding_columns', 'PS_COMMENT');
CALL set_table_property('PARTSUPP', 'time_to_live_in_seconds', '31536000');
COMMIT;
DROP TABLE IF EXISTS PART;
BEGIN;
CREATE TABLE PART
(
    P_PARTKEY     INT    NOT NULL PRIMARY KEY,
    P_NAME        TEXT   NOT NULL,
    P_MFGR        TEXT   NOT NULL,
```

```
    P_BRAND       TEXT   NOT NULL,
    P_TYPE        TEXT   NOT NULL,
    P_SIZE        INT    NOT NULL,
    P_CONTAINER   TEXT   NOT NULL,
    P_RETAILPRICE DECIMAL(15,2) NOT NULL,
    P_COMMENT     TEXT   NOT NULL
);
CALL set_table_property('PART', 'distribution_key', 'P_PARTKEY');
CALL set_table_property('PART', 'bitmap_columns', 'P_PARTKEY,P_SIZE,P_NAME,P_MFGR
,P_BRAND,P_TYPE,P_CONTAINER,P_COMMENT');
CALL set_table_property('PART', 'dictionary_encoding_columns', 'P_NAME,P_MFGR,P_B
RAND,P_TYPE,P_CONTAINER,P_COMMENT');
CALL set_table_property('PART', 'time_to_live_in_seconds', '31536000');
COMMIT;
DROP TABLE IF EXISTS CUSTOMER;
BEGIN;
CREATE TABLE CUSTOMER
(
    C_CUSTKEY    INT    NOT NULL PRIMARY KEY,
    C_NAME       TEXT   NOT NULL,
    C_ADDRESS    TEXT   NOT NULL,
    C_NATIONKEY  INT    NOT NULL,
    C_PHONE      TEXT   NOT NULL,
    C_ACCTBAL    DECIMAL(15,2) NOT NULL,
    C_MKTSEGMENT TEXT   NOT NULL,
    C_COMMENT    TEXT   NOT NULL
);
CALL set_table_property('CUSTOMER', 'distribution_key', 'C_CUSTKEY');
CALL set_table_property('CUSTOMER', 'bitmap_columns', 'C_CUSTKEY,C_NATIONKEY,C_NA
ME,C_ADDRESS,C_PHONE,C_MKTSEGMENT,C_COMMENT');
CALL set_table_property('CUSTOMER', 'dictionary_encoding_columns', 'C_NAME,C_ADDR
ESS,C_PHONE,C_MKTSEGMENT,C_COMMENT');
CALL set_table_property('CUSTOMER', 'time_to_live_in_seconds', '31536000');
COMMIT;
DROP TABLE IF EXISTS SUPPLIER;
BEGIN;
CREATE TABLE SUPPLIER
(
    S_SUPPKEY   INT    NOT NULL PRIMARY KEY,
    S_NAME      TEXT   NOT NULL,
    S_ADDRESS   TEXT   NOT NULL,
    S_NATIONKEY INT    NOT NULL,
    S_PHONE     TEXT   NOT NULL,
    S_ACCTBAL   DECIMAL(15,2) NOT NULL,
    S_COMMENT   TEXT   NOT NULL
);
CALL set_table_property('SUPPLIER', 'distribution_key', 'S_SUPPKEY');
CALL set_table_property('SUPPLIER', 'bitmap_columns', 'S_SUPPKEY,S_NAME,S_ADDRESS
,S_NATIONKEY,S_PHONE,S_COMMENT');
CALL set_table_property('SUPPLIER', 'dictionary_encoding_columns', 'S_NAME,S_ADDR
ESS,S_PHONE,S_COMMENT');
CALL set_table_property('SUPPLIER', 'time_to_live_in_seconds', '31536000');
COMMIT;
DROP TABLE IF EXISTS NATION;
```

```
BEGIN;
CREATE TABLE NATION(
  N_NATIONKEY INT NOT NULL PRIMARY KEY,
  N_NAME text NOT NULL,
  N_REGIONKEY INT NOT NULL,
  N_COMMENT text NOT NULL
);
CALL set_table_property('NATION', 'distribution_key', 'N_NATIONKEY');
CALL set_table_property('NATION', 'bitmap_columns', 'N_NATIONKEY,N_NAME,N_REGIONK
EY,N_COMMENT');
CALL set_table_property('NATION', 'dictionary_encoding_columns', 'N_NAME,N_COMMEN
T');
CALL set_table_property('NATION', 'time_to_live_in_seconds', '31536000');
COMMIT;
DROP TABLE IF EXISTS REGION;
BEGIN;
CREATE TABLE REGION
(
    R_REGIONKEY INT  NOT NULL PRIMARY KEY,
    R_NAME      TEXT NOT NULL,
    R_COMMENT   TEXT
);
CALL set_table_property('REGION', 'distribution_key', 'R_REGIONKEY');
CALL set_table_property('REGION', 'bitmap_columns', 'R_REGIONKEY,R_NAME,R_COMMENT
');
CALL set_table_property('REGION', 'dictionary_encoding_columns', 'R_NAME,R_COMMEN
T');
CALL set_table_property('REGION', 'time_to_live_in_seconds', '31536000');
COMMIT;
```

Step 3: Import sample data

After the internal tables are created, you can perform the following steps to import data to the internal tables in Hologres.

> **Note** Foreign tables map fields in external data sources and do not store data in Hologres. You can use foreign tables in Hologres to query the data that is stored in the MaxCompute project MAXCOMPUTE_PUBLIC_DATA in the public dataset.

1. Go to the **SQL Editor** tab. Click the **Ad-hoc Query** icon in the upper-left corner.
2. On the **Ad-hoc Query** tab, select an instance from the **Instance** drop-down list and a database from the **Database** drop-down list, enter the following sample statements in the SQL editor, and then click **Run**.

   The following SQL statements are used to import data for later queries. The data is imported from tables such as the public.odps_customer_10g and public.odps_lineitem_10g tables in the MaxCompute project MAXCOMPUTE_PUBLIC_DATA in the public dataset to the internal tables with corresponding names in Hologres.

```
INSERT INTO public.customer SELECT * FROM public.odps_customer_10g ;
INSERT INTO public.lineitem SELECT * FROM public.odps_lineitem_10g ;
INSERT INTO public.nation SELECT * FROM public.odps_nation_10g ;
INSERT INTO public.orders SELECT * FROM public.odps_orders_10g ;
INSERT INTO public.part SELECT * FROM public.odps_part_10g ;
INSERT INTO public.partsupp SELECT * FROM public.odps_partsupp_10g ;
INSERT INTO public.region SELECT * FROM public.odps_region_10g ;
INSERT INTO public.supplier SELECT * FROM public.odps_supplier_10g ;
vacuum  nation;
vacuum  region;
vacuum  supplier;
vacuum  customer;
vacuum  part;
vacuum  partsupp;
vacuum  orders;
vacuum  lineitem;
analyze nation;
analyze region;
analyze lineitem;
analyze orders;
analyze customer;
analyze part;
analyze partsupp;
analyze supplier;
analyze lineitem (l_orderkey,l_partkey,l_suppkey);
analyze orders (o_custkey);
analyze partsupp(ps_partkey,ps_suppkey);
```

Step 4: Query data in tables

1. Go to the **SQL Editor** tab. Click the **Ad-hoc Query** icon in the upper-left corner.
2. On the **Ad-hoc Query** tab, select an instance from the **Instance** drop-down list and a database from the **Database** drop-down list, enter the following sample statements in the SQL editor, and then click **Run**.

> **Note** The following SQL statements are used to query data in internal tables. To query data in foreign tables, replace the table name to be queried with the name of the foreign table that you want to query.

The following table provides the links to the 22 TPC-H query statements. To view a specific query statement, you can click the corresponding link in the table.

Item	Query statement			
22 TPC-H query statements	Q1	Q2	Q3	Q4
	Q5	Q6	Q7	Q8
	Q9	Q10	Q11	Q12
	Q13	Q14	Q15	Q16

Item	Query statement			
	Q17	Q18	Q19	Q20
	Q21	Q22	-	-

- Q1

```
select
        l_returnflag,
        l_linestatus,
        sum(l_quantity) as sum_qty,
        sum(l_extendedprice) as sum_base_price,
        sum(l_extendedprice * (1 - l_discount)) as sum_disc_price,
        sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) as sum_charge,
        avg(l_quantity) as avg_qty,
        avg(l_extendedprice) as avg_price,
        avg(l_discount) as avg_disc,
        count(*) as count_order
from
        lineitem
where
        l_shipdate <= date '1998-12-01' - interval '120' day
group by
        l_returnflag,
        l_linestatus
order by
        l_returnflag,
        l_linestatus;
```

- Q2

```
select
        s_acctbal,
        s_name,
        n_name,
        p_partkey,
        p_mfgr,
        s_address,
        s_phone,
        s_comment
from
        part,
        supplier,
        partsupp,
        nation,
        region
where
        p_partkey = ps_partkey
        and s_suppkey = ps_suppkey
        and p_size = 48
        and p_type like '%STEEL'
        and s_nationkey = n_nationkey
        and n_regionkey = r_regionkey
        and r_name = 'EUROPE'
        and ps_supplycost = (
                select
                        min(ps_supplycost)
                from
                        partsupp,
                        supplier,
                        nation,
                        region
                where
                        p_partkey = ps_partkey
                        and s_suppkey = ps_suppkey
                        and s_nationkey = n_nationkey
                        and n_regionkey = r_regionkey
                        and r_name = 'EUROPE'
        )
order by
        s_acctbal desc,
        n_name,
        s_name,
        p_partkey
limit 100;
```

- Q3

```
select
        l_orderkey,
        sum(l_extendedprice * (1 - l_discount)) as revenue,
        o_orderdate,
        o_shippriority
from
        customer,
        orders,
        lineitem
where
        c_mktsegment = 'MACHINERY'
        and c_custkey = o_custkey
        and l_orderkey = o_orderkey
        and o_orderdate < date '1995-03-23'
        and l_shipdate > date '1995-03-23'
group by
        l_orderkey,
        o_orderdate,
        o_shippriority
order by
        revenue desc,
        o_orderdate
limit 10;
```

- Q4

```
select
        o_orderpriority,
        count(*) as order_count
from
        orders
where
        o_orderdate >= date '1996-07-01'
        and o_orderdate < date '1996-07-01' + interval '3' month
        and exists (
                select
                        *
                from
                        lineitem
                where
                        l_orderkey = o_orderkey
                        and l_commitdate < l_receiptdate
        )
group by
        o_orderpriority
order by
        o_orderpriority;
```

- Q5

```
select
        n_name,
        sum(l_extendedprice * (1 - l_discount)) as revenue
from
        customer,
        orders,
        lineitem,
        supplier,
        nation,
        region
where
        c_custkey = o_custkey
        and l_orderkey = o_orderkey
        and l_suppkey = s_suppkey
        and c_nationkey = s_nationkey
        and s_nationkey = n_nationkey
        and n_regionkey = r_regionkey
        and r_name = 'EUROPE'
        and o_orderdate >= date '1996-01-01'
        and o_orderdate < date '1996-01-01' + interval '1' year
group by
        n_name
order by
        revenue desc;
```

- Q6

```
select
        sum(l_extendedprice * l_discount) as revenue
from
        lineitem
where
        l_shipdate >= date '1996-01-01'
        and l_shipdate < date '1996-01-01' + interval '1' year
        and l_discount between 0.02 - 0.01 and 0.02 + 0.01
        and l_quantity < 24;
```

- Q7

```
set hg_experimental_enable_double_equivalent=on;
select
        supp_nation,
        cust_nation,
        l_year,
        sum(volume) as revenue
from
        (
                select
                        n1.n_name as supp_nation,
                        n2.n_name as cust_nation,
                        extract(year from l_shipdate) as l_year,
                        l_extendedprice * (1 - l_discount) as volume
                from
                        supplier,
                        lineitem,
                        orders,
                        customer,
                        nation n1,
                        nation n2
                where
                        s_suppkey = l_suppkey
                        and o_orderkey = l_orderkey
                        and c_custkey = o_custkey
                        and s_nationkey = n1.n_nationkey
                        and c_nationkey = n2.n_nationkey
                        and (
                                (n1.n_name = 'CANADA' and n2.n_name = 'BRAZIL')
                                or (n1.n_name = 'BRAZIL' and n2.n_name = 'CANADA')
                        )
                        and l_shipdate between date '1995-01-01' and date '1996-12-31
'
        ) as shipping
group by
        supp_nation,
        cust_nation,
        l_year
order by
        supp_nation,
        cust_nation,
        l_year;
```

- Q8

```
set hg_experimental_enable_double_equivalent=on;
select
        o_year,
        sum(case
                when nation = 'BRAZIL' then volume
                else 0
        end) / sum(volume) as mkt_share
from
        (
                select
                        extract(year from o_orderdate) as o_year,
                        l_extendedprice * (1 - l_discount) as volume,
                        n2.n_name as nation
                from
                        part,
                        supplier,
                        lineitem,
                        orders,
                        customer,
                        nation n1,
                        nation n2,
                        region
                where
                        p_partkey = l_partkey
                        and s_suppkey = l_suppkey
                        and l_orderkey = o_orderkey
                        and o_custkey = c_custkey
                        and c_nationkey = n1.n_nationkey
                        and n1.n_regionkey = r_regionkey
                        and r_name = 'AMERICA'
                        and s_nationkey = n2.n_nationkey
                        and o_orderdate between date '1995-01-01' and date '1996-12-3
1'
                        and p_type = 'LARGE ANODIZED COPPER'
        ) as all_nations
group by
        o_year
order by
        o_year;
```

- Q9

```
set hg_experimental_enable_double_equivalent=on;
select
        nation,
        o_year,
        sum(amount) as sum_profit
from
        (
                select
                        n_name as nation,
                        extract(year from o_orderdate) as o_year,
                        l_extendedprice * (1 - l_discount) - ps_supplycost * l_quanti
ty as amount
                from
                        part,
                        supplier,
                        lineitem,
                        partsupp,
                        orders,
                        nation
                where
                        s_suppkey = l_suppkey
                        and ps_suppkey = l_suppkey
                        and ps_partkey = l_partkey
                        and p_partkey = l_partkey
                        and o_orderkey = l_orderkey
                        and s_nationkey = n_nationkey
                        and p_name like '%maroon%'
        ) as profit
group by
        nation,
        o_year
order by
        nation,
        o_year desc;
```

- Q10

```
select
        c_custkey,
        c_name,
        sum(l_extendedprice * (1 - l_discount)) as revenue,
        c_acctbal,
        n_name,
        c_address,
        c_phone,
        c_comment
from
        customer,
        orders,
        lineitem,
        nation
where
        c_custkey = o_custkey
        and l_orderkey = o_orderkey
        and o_orderdate >= date '1993-02-01'
        and o_orderdate < date '1993-02-01' + interval '3' month
        and l_returnflag = 'R'
        and c_nationkey = n_nationkey
group by
        c_custkey,
        c_name,
        c_acctbal,
        c_phone,
        n_name,
        c_address,
        c_comment
order by
        revenue desc
limit 20;
```

- Q11

```
select
        ps_partkey,
        sum(ps_supplycost * ps_availqty) as value
from
        partsupp,
        supplier,
        nation
where
        ps_suppkey = s_suppkey
        and s_nationkey = n_nationkey
        and n_name = 'EGYPT'
group by
        ps_partkey having
                sum(ps_supplycost * ps_availqty) > (
                        select
                                sum(ps_supplycost * ps_availqty) * 0.0001000000
                        from
                                partsupp,
                                supplier,
                                nation
                        where
                                ps_suppkey = s_suppkey
                                and s_nationkey = n_nationkey
                                and n_name = 'EGYPT'
                )
order by
        value desc;
```

- Q12

```
select
        l_shipmode,
        sum(case
                when o_orderpriority = '1-URGENT'
                        or o_orderpriority = '2-HIGH'
                        then 1
                else 0
        end) as high_line_count,
        sum(case
                when o_orderpriority <> '1-URGENT'
                        and o_orderpriority <> '2-HIGH'
                        then 1
                else 0
        end) as low_line_count
from
        orders,
        lineitem
where
        o_orderkey = l_orderkey
        and l_shipmode in ('FOB', 'AIR')
        and l_commitdate < l_receiptdate
        and l_shipdate < l_commitdate
        and l_receiptdate >= date '1997-01-01'
        and l_receiptdate < date '1997-01-01' + interval '1' year
group by
        l_shipmode
order by
        l_shipmode;
```

- Q13

```
select
        c_count,
        count(*) as custdist
from
        (
                select
                        c_custkey,
                        count(o_orderkey) as c_count
                from
                        customer left outer join orders on
                                c_custkey = o_custkey
                                and o_comment not like '%special%deposits%'
                group by
                        c_custkey
        ) c_orders
group by
        c_count
order by
        custdist desc,
        c_count desc;
```

- Q14

```
select
        100.00 * sum(case
                when p_type like 'PROMO%'
                        then l_extendedprice * (1 - l_discount)
                else 0
        end) / sum(l_extendedprice * (1 - l_discount)) as promo_revenue
from
        lineitem,
        part
where
        l_partkey = p_partkey
        and l_shipdate >= date '1997-06-01'
        and l_shipdate < date '1997-06-01' + interval '1' month;
```

- Q15

```
with revenue0(SUPPLIER_NO, TOTAL_REVENUE)  as
  (
  select
    l_suppkey,
    sum(l_extendedprice * (1 - l_discount))
  from
    lineitem
  where
    l_shipdate >= date '1995-02-01'
    and l_shipdate < date '1995-02-01' + interval '3' month
  group by
    l_suppkey
  )
select
  s_suppkey,
  s_name,
  s_address,
  s_phone,
  total_revenue
from
  supplier,
  revenue0
where
  s_suppkey = supplier_no
  and total_revenue = (
    select
      max(total_revenue)
    from
      revenue0
  )
order by
  s_suppkey;
```

- Q16

```
select
        p_brand,
        p_type,
        p_size,
        count(distinct ps_suppkey) as supplier_cnt
from
        partsupp,
        part
where
        p_partkey = ps_partkey
        and p_brand <> 'Brand#45'
        and p_type not like 'SMALL ANODIZED%'
        and p_size in (47, 15, 37, 30, 46, 16, 18, 6)
        and ps_suppkey not in (
                select
                        s_suppkey
                from
                        supplier
                where
                        s_comment like '%Customer%Complaints%'
        )
group by
        p_brand,
        p_type,
        p_size
order by
        supplier_cnt desc,
        p_brand,
        p_type,
        p_size;
```

- Q17

```
select
        sum(l_extendedprice) / 7.0 as avg_yearly
from
        lineitem,
        part
where
        p_partkey = l_partkey
        and p_brand = 'Brand#51'
        and p_container = 'WRAP PACK'
        and l_quantity < (
                select
                        0.2 * avg(l_quantity)
                from
                        lineitem
                where
                        l_partkey = p_partkey
        );
```

- Q18

```
select
        c_name,
        c_custkey,
        o_orderkey,
        o_orderdate,
        o_totalprice,
        sum(l_quantity)
from
        customer,
        orders,
        lineitem
where
        o_orderkey in (
                select
                        l_orderkey
                from
                        lineitem
                group by
                        l_orderkey having
                                sum(l_quantity) > 312
        )
        and c_custkey = o_custkey
        and o_orderkey = l_orderkey
group by
        c_name,
        c_custkey,
        o_orderkey,
        o_orderdate,
        o_totalprice
order by
        o_totalprice desc,
        o_orderdate
limit 100;
```

- Q19

```
select
        sum(l_extendedprice* (1 - l_discount)) as revenue
from
        lineitem,
        part
where
        (
                p_partkey = l_partkey
                and p_brand = 'Brand#52'
                and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM PKG')
                and l_quantity >= 3 and l_quantity <= 3 + 10
                and p_size between 1 and 5
                and l_shipmode in ('AIR', 'AIR REG')
                and l_shipinstruct = 'DELIVER IN PERSON'
        )
        or
        (
                p_partkey = l_partkey
                and p_brand = 'Brand#43'
                and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED PACK')
                and l_quantity >= 12 and l_quantity <= 12 + 10
                and p_size between 1 and 10
                and l_shipmode in ('AIR', 'AIR REG')
                and l_shipinstruct = 'DELIVER IN PERSON'
        )
        or
        (
                p_partkey = l_partkey
                and p_brand = 'Brand#52'
                and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG PKG')
                and l_quantity >= 21 and l_quantity <= 21 + 10
                and p_size between 1 and 15
                and l_shipmode in ('AIR', 'AIR REG')
                and l_shipinstruct = 'DELIVER IN PERSON'
        );
```

- Q20

```
select
        s_name,
        s_address
from
        supplier,
        nation
where
        s_suppkey in (
                select
                        ps_suppkey
                from
                        partsupp
                where
                        ps_partkey in (
                                select
                                        p_partkey
                                from
                                        part
                                where
                                        p_name like 'drab%'
                        )
                        and ps_availqty > (
                                select
                                        0.5 * sum(l_quantity)
                                from
                                        lineitem
                                where
                                        l_partkey = ps_partkey
                                        and l_suppkey = ps_suppkey
                                        and l_shipdate >= date '1996-01-01'
                                        and l_shipdate < date '1996-01-01' + interval
'1' year
                        )
        )
        and s_nationkey = n_nationkey
        and n_name = 'KENYA'
order by
        s_name;
```

- Q21

```
select
        s_name,
        count(*) as numwait
from
        supplier,
        lineitem l1,
        orders,
        nation
where
        s_suppkey = l1.l_suppkey
        and o_orderkey = l1.l_orderkey
        and o_orderstatus = 'F'
        and l1.l_receiptdate > l1.l_commitdate
        and exists (
                select
                        *
                from
                        lineitem l2
                where
                        l2.l_orderkey = l1.l_orderkey
                        and l2.l_suppkey <> l1.l_suppkey
        )
        and not exists (
                select
                        *
                from
                        lineitem l3
                where
                        l3.l_orderkey = l1.l_orderkey
                        and l3.l_suppkey <> l1.l_suppkey
                        and l3.l_receiptdate > l3.l_commitdate
        )
        and s_nationkey = n_nationkey
        and n_name = 'PERU'
group by
        s_name
order by
        numwait desc,
        s_name
limit 100;
```

- Q22

```
select
        cntrycode,
        count(*) as numcust,
        sum(c_acctbal) as totacctbal
from
        (
                select
                        substring(c_phone from 1 for 2) as cntrycode,
                        c_acctbal
                from
                        customer
                where
                        substring(c_phone from 1 for 2) in
                                ('24', '32', '17', '18', '12', '14', '22')
                        and c_acctbal > (
                                select
                                        avg(c_acctbal)
                                from
                                        customer
                                where
                                        c_acctbal > 0.00
                                        and substring(c_phone from 1 for 2) in
                                                ('24', '32', '17', '18', '12', '14',
'22')
                        )
                        and not exists (
                                select
                                        *
                                from
                                        orders
                                where
                                        o_custkey = c_custkey
                        )
        ) as custsale
group by
        cntrycode
order by
        cntrycode;
```

7.Grant permissions to a RAM user

This topic shows you how to use your Alibaba Cloud account to authorize a RAM user to connect to and use Hologres.

Prerequisites

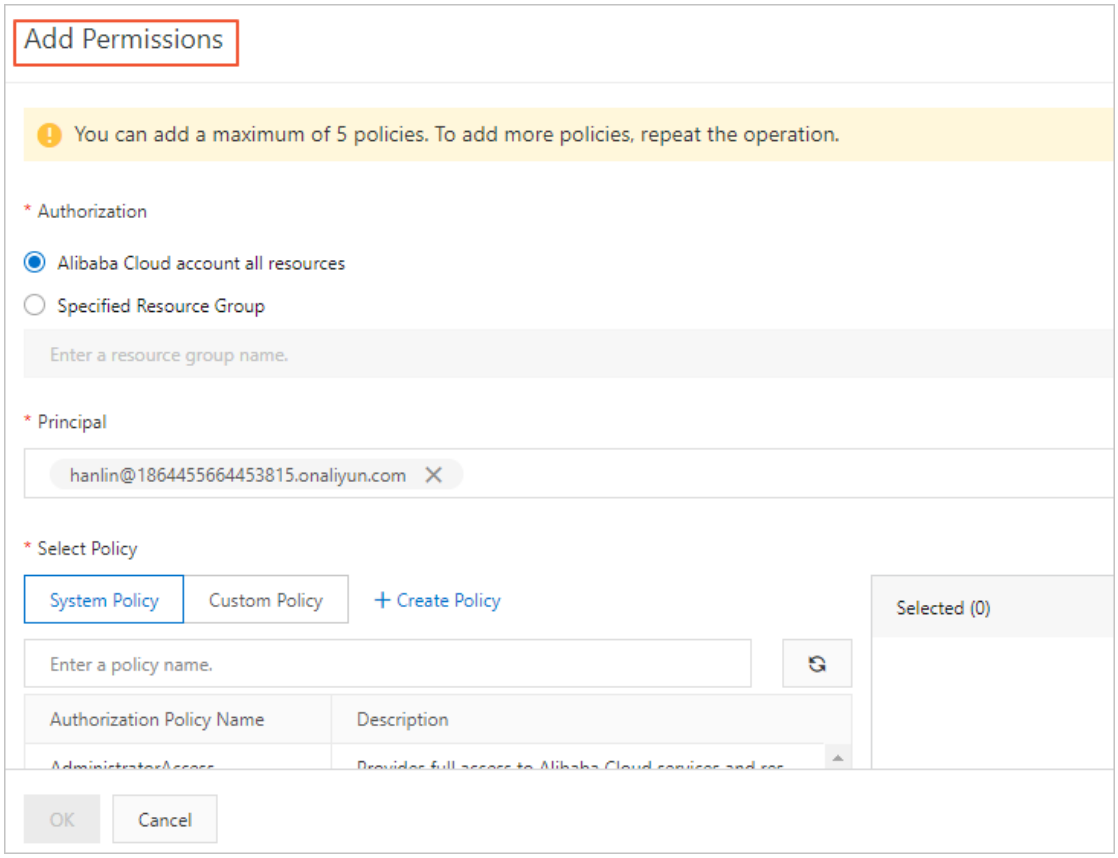
- A RAM user is created. For more information, see Create a RAM user.
- An AccessKey pair is created for the RAM user. For more information, see Create an AccessKey pair for a RAM user.

Grant Hologres permissions to a RAM user

After you grant relevant Hologres permissions to a RAM user in the Resource Access Management (RAM) console by using your Alibaba Cloud account, you can log on to the Hologres console and view, purchase, or delete instances as the RAM user. You can log on to the RAM console, find a RAM user, and then attach policies to the RAM user. If you need to grant the RAM user all permissions to view instance information in the Hologres console, attach the **AliyunHologresFullAccess** and **AliyunRAMReadOnlyAccess** policies.

1. Log on to the RAM console by using your Alibaba Cloud account.
2. Select the RAM user to which you want to grant permissions.
    i. In the left-side navigation pane, click **Users** under **Identities**.
    ii. On the **Users** page, find the RAM user to which you want to grant permissions and click **Add Permissions** in the **Actions** column.
3. Grant permissions to the RAM user.

i. In the **Add Permissions** panel, set the parameters as required.



Parameter	Description
Authorization	Valid values: ■ Alibaba Cloud account all resources ■ Specified Resource Group
Principal	The RAM user to which you want to grant permissions.
Select Policy	Valid values: ■ System Policy ■ Custom Policy **Note** ■ You can create custom policies based on your business needs. ■ You can attach a maximum of five policies at a time. To attach more policies, perform the operation multiple times.

You can select **System Policy** or **Custom Policy** based on the following descriptions:

- **System Policy**

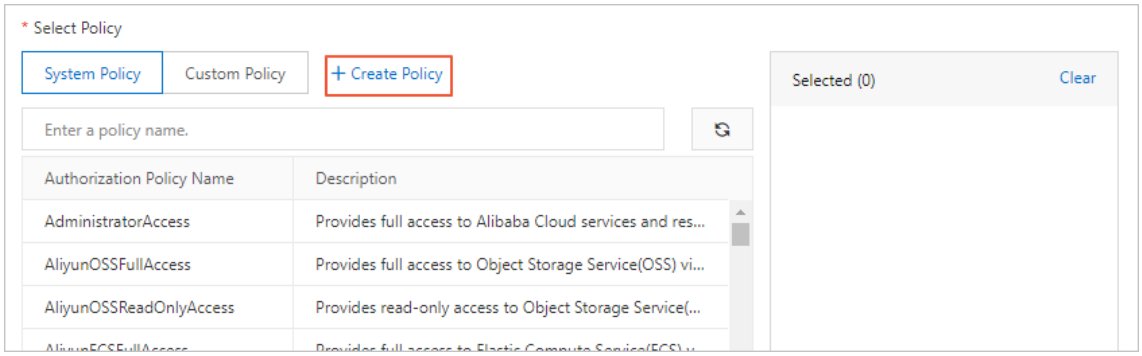
  The following table describes the system policies that you can use to grant permissions on Hologres. If you attach all of these system policies to the RAM user, the RAM user is authorized to perform all operations in the Hologres console.

Policy	Description
AliyunHologresFullAccess	Grants full access permissions on Hologres. If you attach this policy to the RAM user, the RAM user can view the information about all instances and purchase instances in the Hologres console. **Note** To view user information on the **Users** tab of an instance details page in the Hologres console, you must attach the **AliyunRAMReadOnlyAccess** policy to the RAM user.
AliyunBSSOrderAccess	Grants permissions to view, pay for, and cancel orders in the Billing Management console. If you attach this policy to the RAM user, the RAM user can upgrade or downgrade instance specifications and renew instances in the Hologres console.
AliyunRAMReadOnlyAccess	Grants read-only permissions on RAM. If you attach this policy to the RAM user, the RAM user can view the information about the current users, groups, and permissions on the **Users** tab of an instance details page in the Hologres console.
AliyunHologresReadOnlyAccess	Grants read-only permissions on Hologres. If you attach this policy to the RAM user, the RAM user can view the information about all instances but cannot manage the instances in the Hologres console. For example, the RAM user cannot modify the network configurations of instances.

> **Note**
>
> - If you use a RAM user to purchase an instance, the RAM user and the Alibaba Cloud account are superusers by default.
> - If you use an Alibaba Cloud account to purchase an instance, only the Alibaba Cloud account can use the instance by default. RAM users must be authorized by the Alibaba Cloud account before they can use the instance.

- **Custom Policy**

  You can click **Create Policy** to create a custom policy based on your business needs.



On the **Create Custom Policy** page, you can set the configuration mode to **Script**. Then, edit the script of the policy.

Sample statements:

```
{
    "Statement": [
        {   // Grant a RAM user the permissions to perform all operations. After the permissions are granted, the other permissions are not required.
            "Effect": "Allow",
            "Action": "hologram:*", // Indicates that the RAM user has the permissions to perform all operations.
            "Resource": "acs:hologram:*:<Alibaba Cloud account ID>:instance/*" // Indicates that the RAM user has access to instances in all regions.
        },
        {   // Grant a RAM user the permissions to purchase or renew instances.
            "Effect": "Allow",
            "Action": "hologram:*",
            "Resource": "acs:hologram:cn-<region>:<Alibaba Cloud account ID>:instance/*"
        },
        {   // Grant a RAM user the permissions to delete instances.
            "Effect": "Allow",
            "Action": "hologram:DeleteInstance",
            "Resource": "acs:hologram:cn-<region>:<Alibaba Cloud account ID>:instance/*"
        },
        {   // Grant a RAM user the permissions to purchase instances. The RAM user can purchase instances only after the permissions are granted.
            "Effect": "Allow",
            "Action": "bss:PayOrder",
            "Resource": "acs:hologram:cn-<region>:<Alibaba Cloud account ID>:instance
```

```
            "Resource": "acs:hologram:cn-<region>:<Alibaba Cloud account ID>:instance
/*"
        },
        {   // Grant a RAM user the permissions to view instance details.
            "Effect": "Allow",
            "Action": "hologram:DescribeInstance",
            "Resource": "acs:hologram:cn-<region>:<Alibaba Cloud account ID>:instance
/*"
        },
        {   // Grant a RAM user the permissions to view the instance list.
            "Effect": "Allow",
            "Action": "hologram:ListInstances",
            "Resource": "acs:hologram:cn-<region>:<Alibaba Cloud account ID>:instance
/*"
        },
        {   // Grant a RAM user the permissions to suspend instances.
            "Effect": "Allow",
            "Action": "hologram:StopInstance",
            "Resource": "acs:hologram:cn-<region>:<Alibaba Cloud account ID>:instance
/*"
        },
        {   // Grant a RAM user the permissions to resume instances.
            "Effect": "Allow",
            "Action": "hologram:ResumeInstance",
            "Resource": "acs:hologram:cn-<region>:<Alibaba Cloud account ID>:instance
/*"
        },
        {   // Grant a RAM user the permissions to view the monitoring metrics of ins
tances.
            "Effect": "Allow",
            "Action": "hologram:GetInstanceMetrics",
            "Resource": "acs:hologram:cn-<region>:<Alibaba Cloud account ID>:instance
/*"
        },
        {   // Grant a RAM user the permissions to modify the network configurations
of instances.
            "Effect": "Allow",
            "Action": "hologram:ModifyInstanceNetworkType",
            "Resource": "acs:hologram:cn-<region>:<Alibaba Cloud account ID>:instance
/*"
        }
    ],
    "Version": "1"
}
```

The following table describes the parameters in the syntax.

Parameter	Description
<region>	The region of the Hologres instance. Example: **beijing**.
<Alibaba Cloud account ID>	The ID of your Alibaba Cloud account.

Parameter	Description
*	The IDs of all Hologres instances within your Alibaba Cloud account. You can also replace the asterisk (*) with the ID of a specific Hologres instance.

Sample statement:

```
acs:hologram:cn-beijing:4322xxxxx:instance/hhhgggxxxx
```

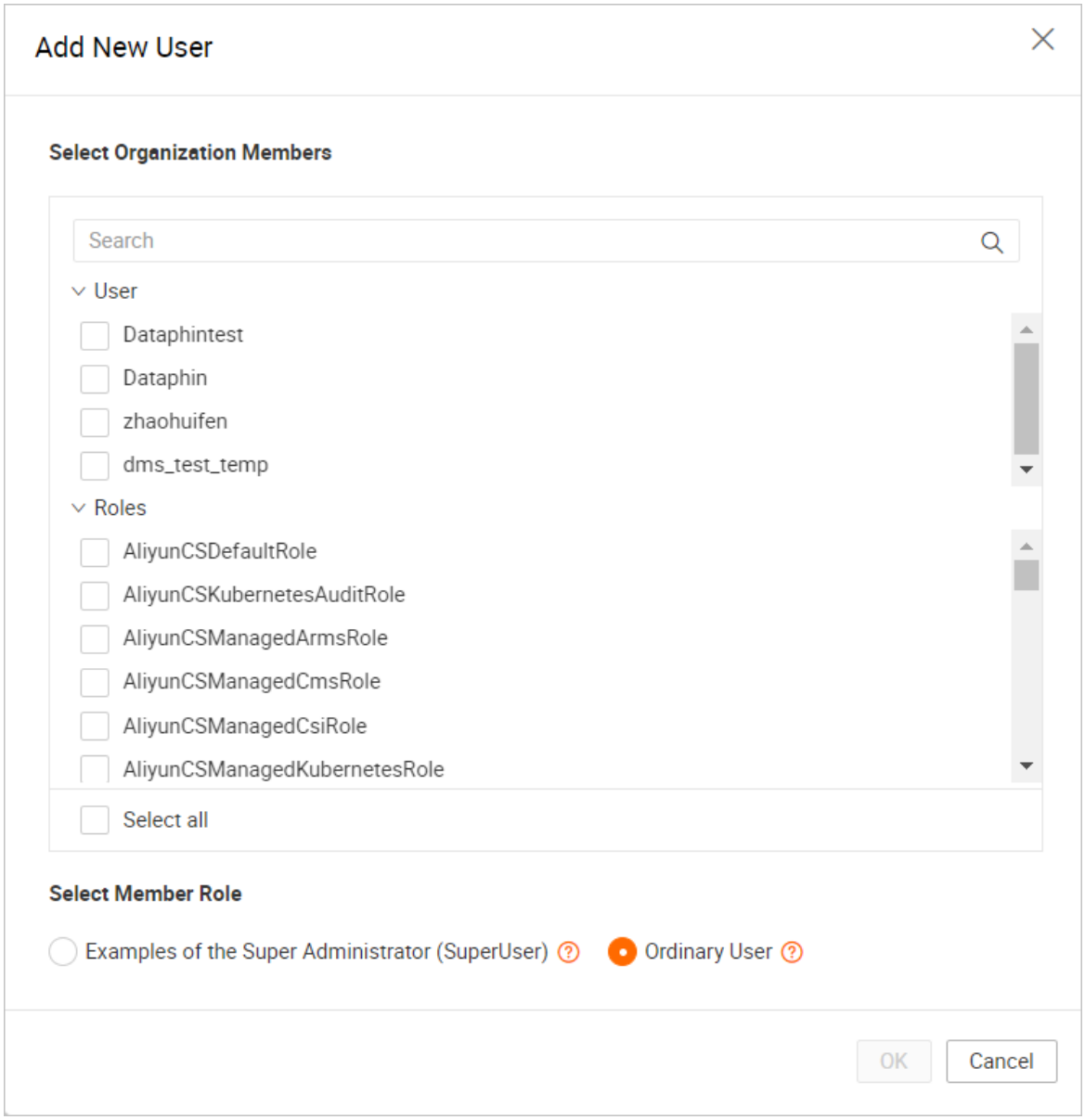
4. Click **OK**.

Grant the development permissions on a Hologres instance to a RAM user

Before you can perform data analytics operations on a Hologres instance as a RAM user, you must use your Alibaba Cloud account to grant the development permissions on the Hologres instance to the RAM user. You can log on to the Hologres console, go to the HoloWeb console, add a user on the User Management page, and then grant permissions to the user. This section describes how to use the simple permission model (SPM) to grant the development permissions on a Hologres instance to a RAM user.

> **Note** You can execute SQL statements to grant permissions to a RAM user by using different permission models. For more information, see the following topics:
>
> - Use the SPM
> - Standard PostgreSQL authorization model

1. Log on to the Alibaba Cloud international site (alibabacloud.com) by using your Alibaba Cloud account.
2. Go to the Hologres console. Click the name of the instance that you want to manage. The instance details page appears.
3. In the left-side pane of the instance details page, click **Users**.
4. On the **User Management** page, click **Add New User**.
5. In the **Add New User** dialog box, set the parameters that are described in the following table.

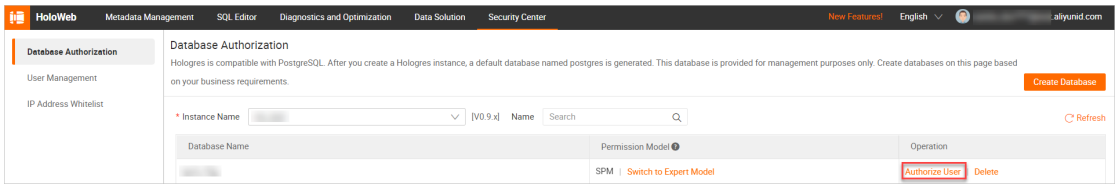


Parameter	Description
Select Organization Members	The RAM user that you want to add to the instance.
Select Member Role	The role to be assigned to the RAM user. Valid values: ○ **Examples of the Super Administrator (SuperUser)**: A superuser has all permissions on the instance. ○ **Ordinary User**: By default, a regular user has no permissions on the instance. A regular user can log on to a Hologres instance and perform allowed data analytics operations only after the regular user is granted the required development permissions.

6. (Optional)If the RAM user is assigned the regular user role, perform the following steps to grant the required permissions to the RAM user:

i. In the left-side pane of the instance details page, click **Databases**.

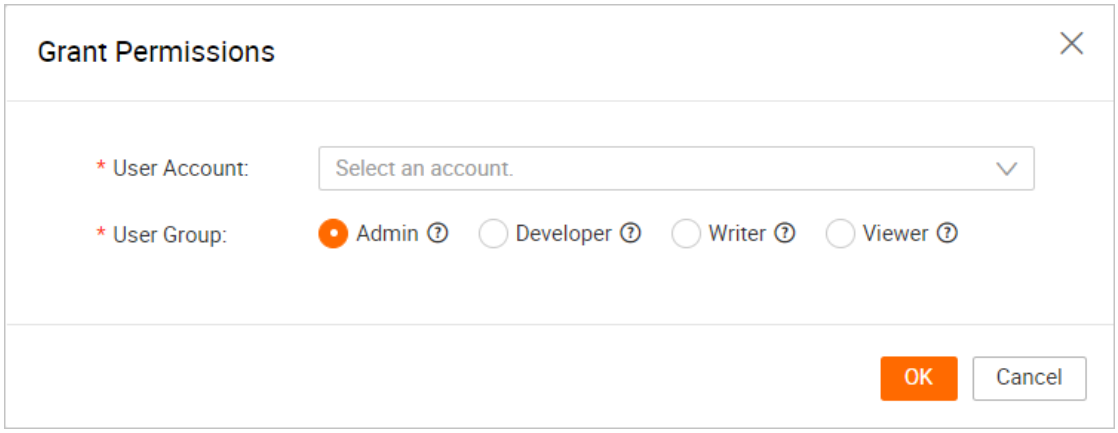
ii. On the **Database Authorization** page, find the database that you want to manage and click **Authorize User** in the Operation column.



> **Note** If no database is created in the Hologres instance, click **Create Database** in the upper-right corner to create a database.

iii. On the permission management page of the database, click **Grant Permissions**.

iv. In the **Grant Permissions** dialog box, set the parameters that are described in the following table.



Parameter	Description
User Account	The RAM user to which you want to grant permissions.
User Group	■ Admin: Users in this group are the owners of the current database and are authorized to manage the database and users in the four user groups. ■ Developer: Users in this group are authorized to read and write data in the current database, and create, delete, or modify objects in the database by executing DDL statements. ■ Writer: Users in this group are authorized to read and write data in the current database. ■ Viewer: Users in this group are authorized to read data in the current database.

v. Click **OK**.

7. Click **OK**.

What's next

After you grant the RAM user the required permissions, you can connect to the instance that you want to manage and perform data analytics operations on the instance as the RAM user. You can use HoloWeb to perform data analytics operations in the Hologres console. For more information, see HoloWeb quick start.