# SQL Server

## Schema Pin

SQL Server database having “select” permission on sys schema shows schema name under schema pin

Command to select database

USE [TestDb]

Command to grant select access permission to TestDb Database

grant select ON database::[TestDb] TO [rajeev]

Command to grant select access permission on sys schema

grant select ON schema::[sys] TO [rajeev]



Command to select database

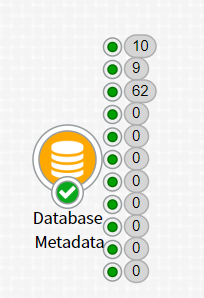
USE [TestDb]

Command to remove select access permission on sys schema

Deny SELECT ON schema::[sys] TO [rajeev]

Command to remove select access permission on dbo schema

Deny SELECT ON schema::[dbo] TO [rajeev]



After removing select permission of “sys” schema and “dbo” schema database, count of schema pin has decreased from 75 to 62

Note: Schemas which are coming are sql server default schemas not user created database schema

## Catalog Pin

1. SQL Server database having “select” permission on database shows database name under catalog pin.

Command to select database

USE [TestDb]

Command to grant select access permission to TestDb Database

grant select ON database::[TestDb] TO [rajeev]



**Note**: By default, it shows all other default system database available on the logged in user database.

## Tables/Views Pin

1. SQL Server database having “select” permission on “sys” schema and its database schema shows table/views and its columns

Command to select database

USE [TestDb]

Command to grant select access permission on **dbo schema**

grant select ON schema:[dbo] TO [rajeev]

Command to grant select access permission on sys schema

grant select ON schema::[sys] TO [rajeev]



1. SQL Server database having no “select” permission on “sys” schema and its database schema, then it will not show table/views and its columns for selected database schema

Command to revoke select permission

DENY SELECT ON schema::[dbo] TO [rajeev]

DENY SELECT ON schema::[sys] TO [rajeev]



Note: After denying the permission on “dbo” and “sys” schema, Table and Table Columns are not getting displayed for schema “sys” and “dbo”. Table pin count has decreased from 2488 to 2448

## Table / View Columns pin

Table / View column will be displayed to the user if user has “select” permission on the database. It requires same permission as of table pin

Command to grant select permission

grant SELECT ON schema::[sys] TO [rajeev]

grant SELECT ON schema::[dbo] TO [rajeev]

## Procedure and its Parameter Pin

1. SQL Server database having “execute” rights on “test” database will show procedure and procedure parameters in their respective pins

Command to grant execute permission on schema

grant execute ON schema::[dbo] TO [rajeev]

Command to grant execute permission on database

grant exec ON database::[TestDb] TO [rajeev]



1. SQL Server database having no “execute” rights will not show procedure and procedure parameters in their respective pins

Command to deny execute permission on schema dbo

Deny execute ON schema:: [dbo] TO [rajeev]

Command to deny execute permission on database

deny exec ON database::[TestDb] TO [rajeev]



Note: Procedure which are coming are sql server default system procedure not user created database procedure

## Function and its Parameter Pin

1. SQL Server database having “execute” rights on database will show function and function parameters in their respective pins

Command to grant execute permission on schema

grant execute ON schema:[dbo] TO [rajeev]

Command to grant execute permission on database

grant exec ON database::[TestDb] TO [rajeev]



1. SQL Server database having no “execute” rights will not show function and function parameters in their respective pins.

Command to deny execute permission on schema dbo

Deny execute ON schema:: [dbo] TO [rajeev]

Command to deny execute permission on database

deny exec ON database::[TestDb] TO [rajeev]



Note: Function which are coming are sql server default system procedure not user created database function

## Trigger Pin

SQL Server database having “select” permission on sys schema and user database will show triggers under trigger pin

Command to grant select access permission to TestDb Database

grant select ON database::[TestDb] TO [rajeev]

Command to grant select access permission on sys schema

grant select ON schema::[sys] TO [rajeev]



SQL Server database having no “select” permission on “sys” schema and its database schema, then it will not show triggers under trigger pin

Command to select database

USE [TestDb]

Command to remove select access permission on sys schema

Deny SELECT ON schema::[sys] TO [rajeev]

Command to remove select access permission on dbo schema

Deny SELECT ON schema::[dbo] TO [rajeev]

## Object Definition Pin

In this pin data is getting fetched from “sys” schema. So if we have “select” permission on the user database and “sys” schema, then data will be displayed.

## Database Metadata Node SQL

Please find below database objects getting used for Database MetaData Node pin

|  |  |  |  |
| --- | --- | --- | --- |
| **Pin Type** | **Dependent Objects** | **Permissions** | **Useful Resource** |
| Schema Pin | Retrieves the schema names that are available in the current database.  It is returned from view **sys.schemas** | Requires SELECT permission on the sys schema | https://docs.microsoft.com/en-us/sql/connect/jdbc/reference/getschemas-method?view=sql-server-ver15 |
| Catalog Pin | It is fetched from below query  SELECT name AS TABLE\_CAT FROM sys.databases order by name | Requires SELECT permission on the user database | https://docs.microsoft.com/en-us/sql/connect/jdbc/reference/getcatalogs-method-sqlserverdatabasemetadata?view=sql-server-ver15 |
| Table/View Pin | Procedure sp\_tables : Returns a list of objects that can be queried in the current environment. This means any table or view, except synonym objects. | Requires SELECT permission on the user database | https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-tables-transact-sql?view=sql-server-ver15 |
| Table/View Column pin | Procedure sp\_columns\_100,sp\_columns : Returns column information for the specified objects that can be queried in the current environment. | Requires SELECT permission on the user database | https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-columns-transact-sql?view=sql-server-ver15 |
| Procedure Pin | It fetches list of procedure from sp\_stored\_procedures.This procedures depends on below table  spt\_all\_procedures objects numbered\_procedures$ system\_objects | Requires execute permission on the database. | https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-stored-procedures-transact-sql?view=sql-server-ver15 |
| Procedure Param Pin | Procedure sp\_sproc\_columns,sp\_sproc\_columns\_100 : Returns column information for a stored procedure or user-defined function in the current environment. | Requires execute permission on the database. | https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-sproc-columns-transact-sql?view=sql-server-ver15 |
| Function | It fetches list of procedure from sp\_stored\_procedures.This procedures depends on below table  spt\_all\_procedures objects numbered\_procedures$ system\_objects | Requires execute permission on the database. | https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-stored-procedures-transact-sql?view=sql-server-ver15 |
| Function Param Pin | Procedure sp\_sproc\_columns,sp\_sproc\_columns\_100 : Returns column information for a stored procedure or user-defined function in the current environment. | Requires execute permission on the database. | https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-sproc-columns-transact-sql?view=sql-server-ver15 |
| Trigger | It returns list of triggers from below sql query  select db\_name() as "Catalog", schema\_name(ob.schema\_id) as "Schema", tg.name as "Trigger", OBJECT\_NAME(tg.PARENT\_ID) table\_name from sys.triggers tg,sys.objects ob where tg.parent\_id = ob.object\_id and tg.name like ? and schema\_name(ob.schema\_id) = ? and db\_name() = ? | Requires SELECT permission on the sys schema and user database | Query for trigger pin is written in perDBConfiguration.props |
| **Object Definition Pin** | | | |
| ObjectType | Query | Tables |  |
| view | SELECT DB\_NAME() as "Catalog", SCHEMA\_NAME(schema\_id) as "Schema", 'VIEW' as "ObjectType", name as "ObjectName", CONVERT(NVARCHAR(MAX), OBJECT\_DEFINITION(object\_id)) as "ObjectCode", '0' as "Notes" from sys.views where name like ? and SCHEMA\_NAME(schema\_id) = ? and DB\_NAME() = ? | sys.views |  |
| Procedure | select DB\_NAME() as "Catalog", SCHEMA\_NAME(o.schema\_id) as "Schema", 'PROCEDURE' as "ObjectType", o.name as "ObjectName", convert(nvarchar(MAX), OBJECT\_DEFINITION(o.object\_id)) as "ObjectCode", '0' as "Notes" from sys.objects o WHERE o.type IN ('P', 'PC') and o.name like ? and SCHEMA\_NAME(o.schema\_id) = ? and DB\_NAME() = ? | sys.objects |  |
| Function | select DB\_NAME() as "Catalog", SCHEMA\_NAME(o.schema\_id) as "Schema", 'FUNCTION' as "ObjectType", o.name as "ObjectName", convert(nvarchar(MAX), OBJECT\_DEFINITION(o.object\_id)) as "ObjectCode", '0' as "Notes" from sys.objects o WHERE o.type IN ('FN', 'TF', 'IF', 'AF', 'FT', 'IS', 'FS') and o.name like ? and SCHEMA\_NAME(o.schema\_id) = ? and DB\_NAME() = ? | sys.objects |  |
| Trigger | select s.catalog\_name as "Catalog", a.tg\_schema as "Schema", a.ObjectType, a.ObjectName, a.ObjectCode, '0' as "Notes" from (select schema\_name(ob.schema\_id) tg\_schema, 'TRIGGER' as "ObjectType", OBJECT\_NAME(tg.PARENT\_ID) as "ObjectName", object\_definition(tg.object\_id) as "ObjectCode" from sys.triggers tg,sys.objects ob where tg.parent\_id = ob.object\_id ) a, information\_schema.schemata s where a.tg\_schema = s.schema\_name and ObjectName like ? and a.tg\_schema = ? and s.catalog\_name = ? | sys.triggers |  |

## Commands to grant permission

1. Logged in user should be either DBA or have a db\_owner role of the database
2. Command to create user on database

USE TestDb

CREATE LOGIN [dberman] WITH PASSWORD='dberman', DEFAULT\_DATABASE=[TestDb]

CREATE USER [dberman] FOR LOGIN [dberman]

**Note**:

TestDb is a user database on which user wants to create user

dberman is a new login / user name that needs to be created on TestDb database

1. Execute either of the below commands to grant permissions to run Database Metadata Node

grant VIEW DEFINITION ON database::[TestDb] TO [dberman]

**OR**

grant select, execute, ALTER ON database::[TestDb] TO [dberman]

## Database inbuilt schemas

1. db\_accessadmin
2. db\_backupoperator
3. db\_datareader
4. db\_datawriter
5. db\_ddladmin
6. db\_denydatareader
7. db\_denydatawriter
8. db\_owner
9. db\_securityadmin
10. guest
11. INFORMATION\_SCHEMA
12. sys

Query to view schemas on SQL Server

SELECT s.name AS schema\_name, s.schema\_id, u.name AS schema\_owner

FROM sys.schemas s INNER JOIN sys.sysusers u ON u.uid = s.principal\_id ORDER BY s.name

## Database inbuilt catalog

1. master
2. tempdb
3. model
4. msdb
5. rdsadmin

Query to view database on SQL Server

SELECT name, database\_id, create\_date

FROM sys.databases ;

Note: “rdsadmin” is a sql server AWS default database

## Function and its Parameter Pin

1. MySQL database having “execute” rights on “test” database will show function and function parameters in their respective pins

Command

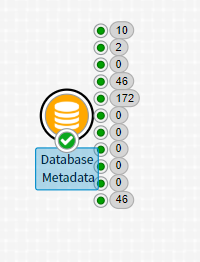
GRANT execute ON test. \* to 'rajeev'@'%'



1. MySQL database having no “execute” rights will not show function and function parameters in their respective pins.

Command

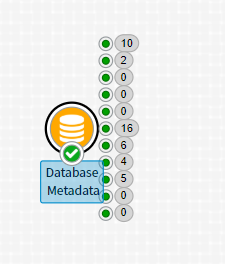
REVOKE execute ON test. \* FROM 'rajeev'@'%'



1. MySQL database having no “SELECT” rights on “mysql.proc” but “EXECUTE” rights shows function in “function” pin but will not show function in object definition as it has not select rights

Command to grant permission to see function in function pin

GRANT execute ON test. \* to 'rajeev'@'%'



To view Object definition also run below

Command to see function definition in object definition pin

GRANT select ON mysql. \* to 'rajeev'@'%'

## Useful Commands

1. Command to select database

USE [TestDb]

1. Command to create User with login name “rajeev” and user name “rajeev”

USE [TestDb]

GO

CREATE LOGIN [rajeev] WITH PASSWORD=N'rajeev', DEFAULT\_DATABASE=[TestDb], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=OFF

GO

USE [msdb]

GO

CREATE USER [rajeev] FOR LOGIN [rajeev]

1. Command to grant select access permission to TestDb Database

grant select ON database::[TestDb] TO [rajeev]

1. Command to grant select access permission on sys schema

grant select ON schema::[sys] TO [rajeev]

1. Command to grant execute permission on schema

grant execute ON schema:[dbo] TO [rajeev]

1. Command to grant execute permission on database

grant exec ON database::[TestDb] TO [rajeev]

1. Command to deny select permission on schema

DENY SELECT ON schema:[dbo] TO [rajeev]

1. Command to deny execute permission on database

deny exec ON database::[TestDb] TO [rajeev]

1. Command to grant view definition on a schema

grant VIEW DEFINITION ON schema::[dbo] TO [rajeev]

1. Command to drop user name and login

DROP USER rajeev

DROP login rajeev

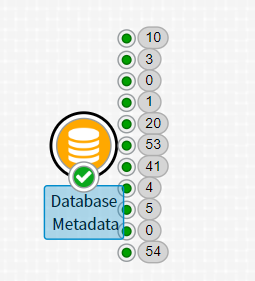
# MySQL

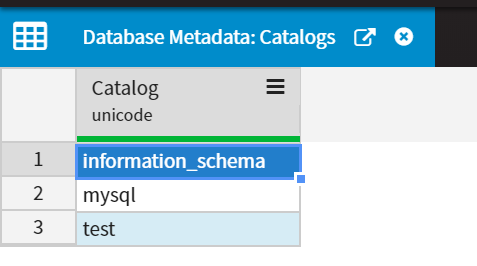
## Catalog Pin

1. MySQL database having “select” permission on database shows database name under catalog pin.

Command to grant select permission

grant select ON `mysql`.\* to 'rajeev'@'%'

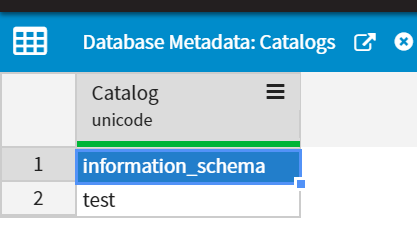




1. If we remove the select permission on “mysql” database , then it will not be displayed under catalog

Command to revoke select permission

revoke select ON `mysql`.\* from 'rajeev'@'%'



## Tables/Views Pin

1. MySQL database having “select” permission on “test” database shows table/views and its columns

Command to grant select permission

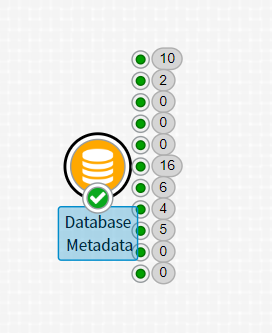
GRANT SELECT ON test. \* to 'rajeev'@'%'



1. MySQL database having no “select” permission on “test” database shows 0 table/views and its columns

Command to revoke select permission

REVOKE select ON test.\* FROM 'rajeev'@'%''



## Table / View Columns pin

Table / View column will be displayed to the user if user has “select” permission on the database. It requires same permission as of table

Command to grant select permission

GRANT SELECT ON test. \* to 'rajeev'@'%'

## Procedure and its Parameter Pin

1. MySQL database having “execute” rights on “test” database will show procedure and procedure parameters in their respective pins

Command to grant execute permission

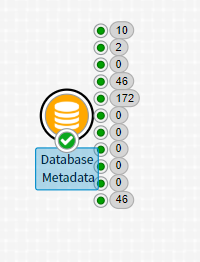
GRANT execute ON test. \* to 'rajeev'@'%'



1. MySQL database having no “execute” rights will not show procedure and procedure parameters in their respective pins

Command

REVOKE execute ON test. \* FROM 'rajeev'@'%'

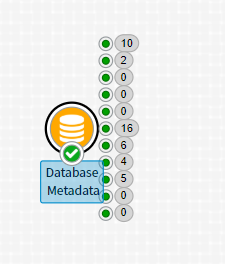


1. MySQL database having no “SELECT” rights on “mysql.proc” but “EXECUTE” rights shows procedure in “procedure” pin but will not show procedure in object definition as it has not select rights

Command

REVOKE select ON mysql.\* FROM 'rajeev'@'%'

GRANT execute ON test.\* to 'rajeev'@'%'



To view Object definition also run below

Command

GRANT select ON mysql. \* to 'rajeev'@'%'

## Object Definition Pin

In this pin data is getting fetched from “information\_schema” and “mysql.proc”. So if we have “select” permission on the user database and “mysql.proc”, then data will be displayed

## Database Metadata Node SQL

Please find below database objects getting used for Database MetaData Node pin

|  |  |  |  |
| --- | --- | --- | --- |
| **Pin Type** | **Dependent Objects** | **Permissions** | **Useful Resource** |
| Schema Pin | Retrieves the schema names that are available in the current database.  It is returned from view **sys.schemas** | Requires SELECT permission on the sys schema | https://docs.microsoft.com/en-us/sql/connect/jdbc/reference/getschemas-method?view=sql-server-ver15 |
| Catalog Pin | It is fetched from below query  SHOW DATABASES | Requires SELECT permission on the Database | https://docs.microsoft.com/en-us/sql/connect/jdbc/reference/getcatalogs-method-sqlserverdatabasemetadata?view=sql-server-ver15 |
| Table/View Pin | Returns list of Table / View from below SQL Query  SHOW TABLES FROM <database name> SHOW FULL TABLES FROM <database name> | Requires SELECT permission on the database | https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-tables-transact-sql?view=sql-server-ver15 |
| Table/View Column pin | Returns list of Table / View columns from below table  INFORMATION\_SCHEMA.COLUMNS  Below query is also used to get Table / view column for a specific table SHOW COLUMNS FROM <databaseName>.<schemaName>.<table\_name> | Requires SELECT permission on the database | https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-columns-transact-sql?view=sql-server-ver15 |
| Procedure Pin | It fetches list of procedure from  SELECT name, type, comment FROM mysql.proc | Requires SELECT and execute permission on the user database . | https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-stored-procedures-transact-sql?view=sql-server-ver15 |
| Procedure Param Pin | It fetches list of procedure parameters by executing below command on procedure name obtained from mysql.proc   SHOW CREATE PROCEDURE <function\_name obtained from mysql.proc> | Requires SELECT and execute permission on the user database . | https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-sproc-columns-transact-sql?view=sql-server-ver15 |
| Function | It fetches list of procedure from  SELECT name, type, comment FROM mysql.proc | Requires SELECT and execute permission on the user database . | https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-stored-procedures-transact-sql?view=sql-server-ver15 |
| Function Param Pin | It fetches list of procedure parameters by executing below command on procedure name obtained from mysql.proc   SHOW CREATE FUNCTION <procedure\_name obtained from mysql.proc> | Requires SELECT and execute permission on the user database . | https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-sproc-columns-transact-sql?view=sql-server-ver15 |
| Trigger | It returns list of triggers from below sql query  select trigger\_catalog as "Catalog", trigger\_schema as "Schema", trigger\_name as "Trigger" from information\_schema.triggers where trigger\_name like ? | Requires SELECT permission on the sys schema | Query for trigger pin is written in perDBConfiguration.props |
|  | | | |
| **Object Definition Pin** | | | |
|  |  |  |  |
| **ObjectType** | **Query** | **Tables** |  |
| View | select table\_schema as "Catalog", '' as "Schema", "VIEW" as "ObjectType", ltrim(table\_name) as "ObjectName", ltrim(view\_definition) as "ObjectCode", '0' as "Notes" from information\_schema.views where table\_name like ? and table\_schema = ? | information\_schema.views |  |
| Procedure | select db as "Catalog", '' as "Schema", type as "ObjectType", name as "ObjectName", concat('CREATE PROCEDURE ', name, '(', param\_list, ')', body) as "ObjectCode", '0' as "Notes" FROM mysql.proc where type='PROCEDURE' and name like ? and db = ? | mysql.proc |  |
| Function | select db as "Catalog", '' as "Schema", type as "ObjectType", name as "ObjectName", concat('CREATE FUNCTION ', name, '(', param\_list, ') RETURNS ', returns, ' ', body) as "ObjectCode", '0' as "Notes" FROM mysql.proc where type='FUNCTION' and name like ? and db = ? | mysql.proc |  |
| Trigger | select trigger\_schema as "Catalog", '' as "Schema", "TRIGGER" as "ObjectType", trigger\_name as "ObjectName", concat('CREATE TRIGGER ', trigger\_name, ' ', ACTION\_TIMING, ' ', EVENT\_MANIPULATION, ' ON ', EVENT\_OBJECT\_TABLE, ' FOR EACH ROW ', ACTION\_STATEMENT) as "ObjectCode", '0' as "Notes" from information\_schema.triggers where trigger\_name like ? and trigger\_schema = ? | information\_schema.triggers |  |

## Commands to grant permission

1. Logged in user should be either DBA or have a admin role of the database
2. Command to create user on database

CREATE USER 'dberman'@'%' IDENTIFIED BY 'dberman';

**Note**:

dberman is a new login / user name that needs to be created on database

1. Execute below commands to grant permissions to run Database Metadata Node

GRANT SELECT, EXECUTE, trigger, show view ON test.\* to 'dberman'@'%'

GRANT SELECT ON mysql.proc to 'dberman'@'%'

**Note**:

“test” is a user database on which user wants to create user

dberman is a database user name to which permission needs to be granted

## Database inbuilt schemas

MySQL does not support schema

## Database inbuilt catalog

1. information\_schema
2. mysql
3. performance\_schema
4. sys

## Useful Commands

1. Command to create User

CREATE USER 'rajeev'@'%' IDENTIFIED BY 'rajeev';

1. Command to grant permission to newly created user

GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, PROCESS, REFERENCES, INDEX, ALTER, SHOW DATABASES, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION SLAVE, REPLICATION CLIENT, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, TRIGGER ON test.\* TO 'rajeev'@'%' WITH GRANT OPTION;

1. Command to grant select permission

Grant SELECT ON test.\* to 'rajeev'@'%'

1. Command to revoke select permission from a database

REVOKE select ON test.\* FROM 'rajeev'@'%'

1. Command to grant execute permission to view/execute procedures

GRANT execute ON test. \* to 'rajeev'@'%'

1. Command to grant select permission to see procedures / function definition

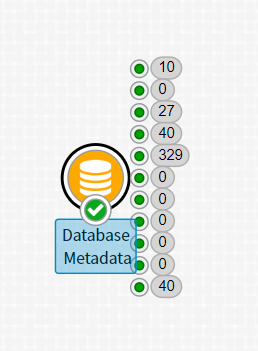
GRANT select ON mysql.proc to 'rajeev'@'%'

# Oracle

## Schema Pin

Oracle database users having “select” permission on “all\_users” table shows all schema present in database under schema pin

Since “all\_users” is a system table, so user has default access on it.



## Catalog Pin

1. Oracle database have no catalog pin.

## Tables/Views Pin

1. Oracle database users having “select” permission on the tables will able to see table/views and its columns

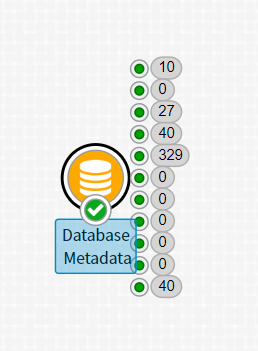
Command to grant select access permission to a user

grant select any table to rajeev

Command to grant select access permission to a specific table to a user

grant select on laetestuser.LOWER to rajeev

Note: Here “laetestuser”, “rajeev” is user / schema of oracle database and “LOWER” is a table available in laetestuser schema.



1. Oracle database users having no “select” permission on the table will not be showed table/views and its columns

Command to deny select access permission to a user

revoke select any table from rajeev

Command to deny select access permission to a specific table to a user

revoke select on laetestuser.LOWER from rajeev

Note: Here “laetestuser”, “rajeev” is user / schema of oracle database and “LOWER” is a table available in laetestuser schema.



## Table / View Columns pin

Table / View column will be displayed to the user if user has “select” permission on the tables. It requires same permission as of table

Command to grant select access permission to a user

grant select any table to rajeev

Command to grant select access permission to a specific table to a user

grant select on laetestuser.LOWER to rajeev

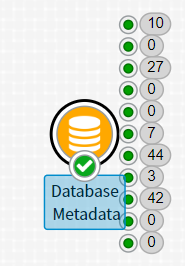
Note: Here “laetestuser”, “rajeev” is user / schema of oracle database and “LOWER” is a table available in laetestuser schema.

## Procedure and its Parameter Pin

1. Oracle database having “execute” rights on the procedures will show procedure and procedure parameters in their respective pins

Command to grant execute permission on schema

grant execute any procedure to rajeev



1. Oracle database having no “execute” rights will not show procedure and procedure parameters in their respective pins

Command to deny execute permission on schema

revoke execute any procedure from rajeev

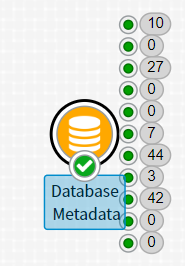


## Function and its Parameter Pin

1. Oracle database having “execute” rights on database will show function and function parameters in their respective pins

Command to grant execute permission on schema

grant execute any procedure to rajeev



1. Oracle database having no “execute” rights will not show function and function parameters in their respective pins.

Command to deny execute permission on schema

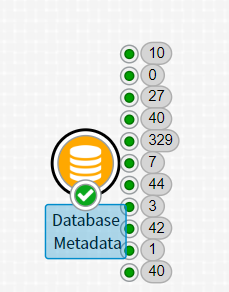
revoke execute any procedure from rajeev



## Trigger Pin

Oracle database having “select” permission on the table “all\_triggers” will show all triggers under trigger pin

Since “all\_triggers” is a system table, so user has default access on it.



## Object Definition Pin

In this pin data is displayed if user has “select” and “execute” permission on the schema for which data needs to be displayed.

User will see object of its own schema and other user schema if User has a role SELECT\_CATALOG\_ROLE

grant SELECT\_CATALOG\_ROLE to rajeev

## Database Metadata Node SQL

Please find below database objects getting used for Database MetaData Node pin

|  |  |  |
| --- | --- | --- |
| **Pin Type** | **Dependent Objects** | **Permissions** |
| Schema Pin | Retrieves the schema names that are available in the current database.  It is returned from below sql query  SELECT username AS table\_schem,null as table\_catalog FROM all\_users ORDER BY table\_schem | Requires SELECT permission on the all\_users table |
| Catalog Pin | In Oracle no catalog |  |
| Table/View Pin | Returns list of Table / View from join of below tables  all\_objects , all\_tab\_comments,all\_synonyms | Requires SELECT permission on the tables of a schema |
| Table/View Column pin | Returns list of Table / View columns from below table  all\_tab\_cols / all\_tab\_columns | Requires SELECT permission on the tables of a schema |
| Procedure Pin | It fetches list of procedure from below tables all\_objects , all\_procedures , all\_arguments | Requires SELECT and execute permission on the user database . |
| Procedure Param Pin | It fetches list of procedure parameters by join of below table  all\_arguments arg, all\_procedures proc | Requires SELECT and execute permission on the user database . |
| Function | It fetches list of function from below tables all\_objects, all\_arguments | Requires SELECT and execute permission on the user database . |
| Function Param Pin | It fetches list of function parameters from below tables  all\_arguments arg, all\_procedures proc | Requires SELECT and execute permission on the user database . |
| Trigger | It returns list of triggers from below sql query  select '' as "Catalog", owner as "Schema", 'TRIGGER' as "ObjectType", trigger\_name "ObjectName", case when DBMS\_LOB.GETLENGTH(DBMS\_METADATA.GET\_DDL('TRIGGER', TRIGGER\_NAME, owner)) <= 4000 then to\_char(DBMS\_METADATA.GET\_DDL('TRIGGER', TRIGGER\_NAME, owner)) else '' end as "ObjectCode", case when DBMS\_LOB.GETLENGTH(DBMS\_METADATA.GET\_DDL('TRIGGER', TRIGGER\_NAME, owner)) <= 4000 then '0' else '1' end as "Notes" from all\_triggers a where trigger\_name like ? and a.owner = ? | Requires SELECT permission on all\_triggers table |
| **Object Definition Pin** | | |
|  |  |  |
| **ObjectType** | **Query** | **Tables** |
| View | select '' as "Catalog", a.owner as "Schema", a.object\_type as "ObjectType", a.object\_name as "ObjectName", DBMS\_METADATA.get\_ddl (a.object\_type, a.object\_name, a.owner) as "ObjectCode", '0' as "Notes" from all\_objects a where object\_type = 'VIEW' and a.object\_name like ? and a.owner = ? | all\_objects |
| Procedure | select '' as "Catalog", a.owner as "Schema", a.object\_type as "ObjectType", a.object\_name as "ObjectName", DBMS\_METADATA.get\_ddl (a.object\_type, a.object\_name, a.owner) as "ObjectCode", '0' as "Notes" from all\_objects a where object\_type = 'PROCEDURE' and a.object\_name like ? and a.owner = ? | all\_objects |
| Function | select '' as "Catalog", a.owner as "Schema", a.object\_type as "ObjectType", a.object\_name as "ObjectName", DBMS\_METADATA.get\_ddl (a.object\_type, a.object\_name, a.owner) as "ObjectCode", '0' as "Notes" from all\_objects a where object\_type = 'FUNCTION' and a.object\_name like ? and a.owner = ? | all\_objects |
| Trigger | select '' as "Catalog", owner as "Schema", 'TRIGGER' as "ObjectType", trigger\_name "ObjectName", case when DBMS\_LOB.GETLENGTH(DBMS\_METADATA.GET\_DDL('TRIGGER', TRIGGER\_NAME, owner)) <= 4000 then to\_char(DBMS\_METADATA.GET\_DDL('TRIGGER', TRIGGER\_NAME, owner)) else '' end as "ObjectCode", case when DBMS\_LOB.GETLENGTH(DBMS\_METADATA.GET\_DDL('TRIGGER', TRIGGER\_NAME, owner)) <= 4000 then '0' else '1' end as "Notes" from all\_triggers a where trigger\_name like ? and a.owner = ? | all\_triggers |

## Commands to grant permission

1. Loggin to same schema /user where permission needs to be granted to other user
2. Logged in user should be either DBA or have a admin role of the database
3. Command to create user on database

CREATE USER dberman IDENTIFIED BY dberman;

GRANT CREATE SESSION TO dberman;

**Note**:

dberman is a new login / user name that needs to be created on database

1. Execute below commands to grant permissions to run Database Metadata Node

grant select any table to dberman

grant execute any procedure to dberman

grant SELECT\_CATALOG\_ROLE to dberman

grant CREATE ANY TRIGGER to dberman

**Note**:

1. dberman is a database user name to which permission needs to be granted
2. ‘grant CREATE ANY TRIGGER to dberman’ is required to be executed only if user want to see other user triggers. If not executed, then user can its own schema trigger. For reference visit below link

<https://docs.oracle.com/cd/B19306_01/server.102/b14237/statviews_2107.htm#i1592586>

## Database inbuilt schemas

1. SYSTEM
2. XDB
3. CTXSYS
4. ANONYMOUS
5. DIP
6. SYS
7. DBSNMP
8. RDSADMIN
9. OUTLN

In oracle, users are schema. So it can differ based on oracle version

## Database inbuilt catalog

Oracle does not support catalog.

## Useful Commands

1. Command to create user in oracle database

CREATE USER rajeev IDENTIFIED BY rajeev;

GRANT CREATE SESSION TO rajeev;

1. Command to grant select access permission to a table

grant select on laetestuser.LOWER to rajeev

Note: Here “laetestuser”, “rajeev” is user / schema of oracle database and “LOWER” is a table available in laetestuser schema.

1. Command to grant select access permission to any table

grant select any table to rajeev

1. Command to grant execute permission on any procedure and function

grant execute any procedure to rajeev

1. Command to deny select access permission to any table

revoke select any table from rajeev

1. Command to deny select permission on any procedure / function

revoke execute any procedure from Rajeev

# PostgreSQL

## Schema Pin

PostgreSQL database displays all the schemas available in the database

.



## Catalog Pin

1. PostgreSQL database shows same database name to which user has connected.

## Tables/Views Pin

1. PostgreSQL database users having database connection permission will be able to see all table/views and its columns



## Table / View Columns pin

Table / View column will be displayed to the user if user has database connection permission.

## Procedure and its Parameter Pin

All the procedure and procedure parameters are displayed in Function and Function parameter pin



## Function and its Parameter Pin

1. PostgreSQL database users having database connection permission will be able to see function and function parameters in their respective pins

## Trigger Pin

PostgreSQL database users having “Trigger” permission on the tables of schema will show all triggers under trigger pin

Commands to grant trigger permission

GRANT TRIGGER ON ALL TABLES IN SCHEMA public TO dberman

Note:

Here dberman is a user of database

Public is a schema



## Object Definition Pin

In this pin procedure / function / view is displayed to the user if user has “ownership” of the object. Trigger is displayed if user has grant permission on all the tables of a schema for which data needs to be displayed. Perform below commands

1. Create a role “owner\_role” by running below command

create role owner\_role

1. Grant role “owner\_role” to the user by running below command

grant owner\_role to dberman

1. Execute below function and run it

CREATE OR REPLACE FUNCTION grantOwnerShip(dbCatalog varchar , dbSchema varchar)

RETURNS VOID AS $$

DECLARE

rel RECORD;

BEGIN

FOR rel IN

SELECT table\_name FROM information\_schema.views where table\_catalog = dbCatalog and table\_schema=dbSchema

LOOP

EXECUTE ' alter view '|| rel.table\_name || ' owner to owner\_role ';

END LOOP;

FOR rel IN

SELECT routine\_name FROM information\_schema.routines,pg\_proc pgp WHERE routine\_definition IS NOT NULL and routine\_type in ( 'FUNCTION','PROCEDURE') AND pgp.proname = routine\_name and routine\_schema=dbSchema AND specific\_catalog=dbCatalog

LOOP

EXECUTE ' alter ROUTINE '|| rel.routine\_name || ' owner to owner\_role ';

END LOOP;

END;

$$ LANGUAGE plpgsql;

1. Run this function by executing below command

select grantOwnerShip('lavastorm','public')

Note :

lavastorm is the database

public is a schema

## Database Metadata Node SQL

Please find below database objects getting used for Database Metadata Node pin

|  |  |  |
| --- | --- | --- |
| **Pin Type** | **Dependent Objects** | **Permissions** |
| Schema Pin | Retrieves the schema names that are available in the current database.  It is returned from below sql query  SELECT nspname AS TABLE\_SCHEM, NULL AS TABLE\_CATALOG FROM pg\_catalog.pg\_namespace   WHERE nspname <> 'pg\_toast' | Users having database login access |
| Catalog Pin | It belongs to connection to which we connected. Only 1 catalog | Users having database login access |
| Table/View Pin | Returns list of Table / View from join of below tables  pg\_catalog.pg\_namespace n, pg\_catalog.pg\_class , pg\_catalog.pg\_description , pg\_catalog.pg\_class | Users having database login access |
| Table/View Column pin | Returns list of Table / View columns from join of below table  pg\_catalog.pg\_namespace n, pg\_catalog.pg\_class , pg\_catalog.pg\_description , pg\_catalog.pg\_attrdef , pg\_catalog.pg\_type , pg\_catalog.pg\_attribute | Users having database login access |
| Procedure Pin | It fetches list of procedure from join of below tables pg\_catalog.pg\_namespace n, pg\_catalog.pg\_class , pg\_catalog.pg\_description , pg\_catalog.pg\_proc | Users having database login access |
| Procedure Param Pin | It fetches list of procedure parameters by join of below table  pg\_catalog.pg\_proc p, pg\_catalog.pg\_namespace n, pg\_catalog.pg\_type | Users having database login access |
| Function | It fetches list of function from join of below tables pg\_catalog.pg\_namespace n, pg\_catalog.pg\_description , pg\_catalog.pg\_proc | Users having database login access |
| Function Param Pin | It fetches list of function parameters from below tables  pg\_catalog.pg\_proc p, pg\_catalog.pg\_namespace n, pg\_catalog.pg\_type | Users having database login access |
| Trigger | It returns list of triggers from below sql query  select a.trigger\_catalog as "Catalog", a.trigger\_schema as "Schema", 'TRIGGER' as "ObjectType", a.trigger\_name as "ObjectName", pg\_get\_triggerdef(oid) as "ObjectCode", '0' as "Notes" from information\_schema.triggers a, pg\_trigger pgt where a.action\_statement is not null and pgt.tgname = a.trigger\_name and a.trigger\_name like ? and a.trigger\_schema = ? and a.trigger\_catalog = ? | Users having “Trigger” permission on the tables of schema |
|  |  |  |
|  | **Object Definition Pin** |  |
|  |  |  |
| **ObjectType** | **Query** | **Tables** |
| View | select table\_catalog as "Catalog", table\_schema as "Schema", 'VIEW' as "ObjectType", table\_name as "ObjectName", 'CREATE OR REPLACE VIEW ' || table\_schema || '.' || table\_name || ' AS ' || view\_definition as "ObjectCode", '0' as "Notes" from information\_schema.views where view\_definition is not null and table\_name like ? and table\_schema = ? and table\_catalog = ? | information\_schema.views |
| Procedure | select routine\_catalog as "Catalog", routine\_schema as "Schema", routine\_type as "ObjectType", routine\_name as "ObjectName", (select pg\_get\_functiondef(pgp.oid)) as "ObjectCode", '0' as "Notes" from information\_schema.routines, pg\_proc pgp where routine\_definition is not null and routine\_type='PROCEDURE' and pgp.proname = routine\_name and routine\_name like ? and routine\_schema = ? and routine\_catalog = ? | information\_schema.routines, pg\_proc |
| Function | select routine\_catalog as "Catalog", routine\_schema as "Schema", routine\_type as "ObjectType", routine\_name as "ObjectName", (select pg\_get\_functiondef(pgp.oid)) as "ObjectCode", '0' as "Notes" from information\_schema.routines, pg\_proc pgp where routine\_definition is not null and routine\_type='FUNCTION' and pgp.proname = routine\_name and routine\_name like ? and routine\_schema = ? and routine\_catalog = ? | information\_schema.routines, pg\_proc |
| Trigger | select a.trigger\_catalog as "Catalog", a.trigger\_schema as "Schema", 'TRIGGER' as "ObjectType", a.trigger\_name as "ObjectName", pg\_get\_triggerdef(oid) as "ObjectCode", '0' as "Notes" from information\_schema.triggers a, pg\_trigger pgt where a.action\_statement is not null and pgt.tgname = a.trigger\_name and a.trigger\_name like ? and a.trigger\_schema = ? and a.trigger\_catalog = ? | information\_schema.triggers a, pg\_trigger |

## Database inbuilt schemas

1. information\_schema
2. pg\_catalog

## Database inbuilt catalog

It doesn’t have predefined catalog.

## Commands to grant permission

1. Logged in user should be either DBA or have a admin role of the database
2. Command to create user on database

CREATE USER dberman WITH ENCRYPTED PASSWORD ' dberman ';

**Note**:

dberman is a new login / user name that needs to be created on database

1. Execute below commands to grant permissions to run Database Metadata Node

GRANT TRIGGER ON ALL TABLES IN SCHEMA PUBLIC to dberman

## Useful Commands

1. Command to create user in PostgreSQL database

CREATE USER dberman WITH ENCRYPTED PASSWORD ' dberman ';

1. Command to grant select access permission on all table

GRANT SELECT ON ALL TABLES IN SCHEMA public TO dberman

Note: Here “dberman” is user of PostgreSQL database and “public” is a schema.

1. Command to grant select access permission to all table

GRANT SELECT ON ALL TABLES IN SCHEMA public TO dberman

1. Command to grant execute permission on all function

grant execute on all functions in schema public to dberman

1. Command to create role

create role owner\_role

1. Command to grant role to a user

GRANT owner\_role TO dberman

# DB2

## Database inbuilt schemas

1. NULLID
2. SQLJ
3. SYSCAT
4. SYSFUN
5. SYSIBM
6. SYSIBMADM
7. SYSIBMINTERNAL
8. SYSIBMTS
9. SYSPROC
10. SYSPUBLIC
11. SYSSTAT
12. SYSTOOLS

## Database inbuilt catalog

It doesn’t have support catalog.

## Commands to grant permission

It does not require any special permission

# Db2 Z / OS

## Database inbuilt schemas

1. ADB
2. CLARKG
3. DSN8110
4. DSN81110
5. DSN811SA
6. DSN8BQRY
7. DSNRGCOL
8. IBMUSER
9. Q
10. RUNSTATS
11. SYSIBM
12. SYSIBMTS
13. SYSTOOLS

For reference, check point number 3

<https://www.ibm.com/support/knowledgecenter/en/SSTQBD_12.0.5/com.ibm.zsys.rdt.tools.user.guide.doc/topics/api_zos_db2.html>

## Database inbuilt catalog

It doesn’t have support catalog.

## Commands to grant permission

It does not require any special permission

# Sybase

## Database inbuilt schemas

Users and system roles represents schema in Sybase. Please find below schema list

1. ha\_role
2. sso\_role
3. mon\_role
4. messaging\_role
5. sa\_role
6. dbo
7. Schema
8. js\_admin\_role
9. js\_user\_role
10. Schema
11. guest
12. oper\_role

## Database inbuilt catalog

1. master
2. model
3. sybsystemprocs
4. sybsystemdb
5. tempdb
6. sybsecurity
7. sybmgmtdb

## Commands to grant permission

1. Logged in user should be either DBA or have a admin role of the database
2. Command to create user and login on database

**use** master

sp\_addlogin dberman, dberman

sp\_adduser dberman

1. Command to grant select and execute permission on “master” database

select 'grant select, execute on ' + name + ' to dberman' from sysobjects where type in ('U','P')

1. Command to create user on “sybmgmtdb” database and provide select execute permission on the database objects

**use** sybmgmtdb

sp\_adduser dberman

**select** 'grant select, execute on ' + name + ' to dberman' **from** sysobjects **where** **type** **in** ('U','P')

1. Command to create user on “model” database and provide select execute permission on the database objects

**use** model

sp\_adduser dberman

**select** 'grant select, execute on ' + name + ' to dberman' **from** sysobjects **where** **type** **in** ('U','P')

1. Command to create user on “sybsystemdb” database and provide select execute permission on the database objects

**use** sybsystemdb

sp\_adduser dberman

**select** 'grant select, execute on ' + name + ' to dberman' **from** sysobjects **where** **type** **in** ('U','P')

1. Command to create user on “sybsystemprocs” database and provide select execute permission on the database objects

**use** sybsystemprocs

sp\_adduser dberman

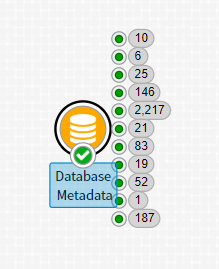
**select** 'grant select, execute on ' + name + ' to dberman' **from** sysobjects **where** **type** **in** ('U','P')

**Note**:

master is a user database on which user wants to create user

sybmgmtdb, model, sybsystemdb, sybsystemprocs is a system database of sysbase.

dberman is a new login / user name that needs to be created on master database



# Snowflake

## Database inbuilt schemas

|  |  |
| --- | --- |
| **Schema** | **Catalog** |
| INFORMATION\_SCHEMA | User Database specified in connection to metatdata |
| PUBLIC | User Database specified in connection to metatdata |
| INFORMATION\_SCHEMA | UTIL\_DB |
| PUBLIC | UTIL\_DB |
| TPCDS\_SF100TCL | SNOWFLAKE\_SAMPLE\_DATA |
| TPCDS\_SF10TCL | SNOWFLAKE\_SAMPLE\_DATA |
| TPCH\_SF10 | SNOWFLAKE\_SAMPLE\_DATA |
| WEATHER | SNOWFLAKE\_SAMPLE\_DATA |
| INFORMATION\_SCHEMA | SNOWFLAKE\_SAMPLE\_DATA |
| PUBLIC | SNOWFLAKE\_SAMPLE\_DATA |
| TPCH\_SF001 | SNOWFLAKE\_SAMPLE\_DATA |
| TPCH\_SF100 | SNOWFLAKE\_SAMPLE\_DATA |
| TPCH\_SF10000 | SNOWFLAKE\_SAMPLE\_DATA |
| TPCH\_SF1 | SNOWFLAKE\_SAMPLE\_DATA |
| TPCH\_SF1000 | SNOWFLAKE\_SAMPLE\_DATA |

## Database inbuilt catalog

1. DEMO\_DB
2. UTIL\_DB
3. SNOWFLAKE\_SAMPLE\_DATA

## Commands to grant permission

1. Logged in user should login through sysadmin role or have a role with permission to create user, role and grant permission
2. Command to create a user

CREATE USER dberman password=' dberman' default\_role = metadata\_access\_role default\_warehouse='COMPUTE\_WH'

Note:

dberman is the new user that we are going to create

metadata\_access\_role is the role having ownership permission to all user defined database object

'COMPUTE\_WH' is the default warehouse of the database. Please change it as per customer database warehouse. Follow below command to get default warehouse

SELECT CURRENT\_WAREHOUSE()

If the customer has a role which has ownership of all the user defined database objects, then we need to only assign new user to this role and we don’t need to perform below steps i.e. step 3 and step 4. But if there is no any role having ownership of all the user defined database objects then follow below steps i.e. step 3 and step 4

1. Command to create a role

CREATE ROLE metadata\_access\_role

Grant role metadata\_access\_role to user dberman

1. Commands to grant permission on tables, view, procedures and functions
   1. **GRANT** **OWNERSHIP** **ON** **ALL** procedures IN **DATABASE** DEMO\_DB **TO** **ROLE** metadata\_access\_role;
   2. **GRANT** **OWNERSHIP** **ON** **ALL** **views** IN **DATABASE** DEMO\_DB **TO** **ROLE** metadata\_access\_role;
   3. **GRANT** **OWNERSHIP** **ON** **ALL** **FUNCTIONS** IN **DATABASE** DEMO\_DB **TO** **ROLE** metadata\_access\_role;
   4. **GRANT** **SELECT** **ON** **ALL** **tables** IN **DATABASE** DEMO\_DB **TO** **ROLE** metadata\_access\_role
   5. grant usage on warehouse COMPUTE\_WH to role metadata\_access\_role;
   6. GRANT **USAGE** **ON** **SCHEMA** TestCatalog.TESTSCHEMA **TO** **ROLE** metadata\_access\_role;

Note:

* + - * + 'COMPUTE\_WH' is the default warehouse of the database. Please change it as per customer database warehouse
        + If GRANT OWNERSHIP is throwing error specifying that it is used for another role /user. If it is ok to remove the reference, then append “REVOKE CURRENT GRANTS” to GRANT OWNERSHIP as per below

GRANT OWNERSHIP ON ALL procedures IN DATABASE DEMO\_DB TO ROLE metadata\_access\_role **REVOKE CURRENT GRANTS**

* + - * + TestCatalog.TESTSCHEMA is a schema in “TestCatalog” database. We need to grant usage permission to user schema if the schema is not “public”

1. Commands to grant permission to all objects that will be created in future

**grant** **OWNERSHIP** **on** future **tables** **in** **database** DEMO\_DB **TO** **ROLE** metadata\_access\_role;

**grant** **OWNERSHIP** **on** future **VIEWS** **in** **database** DEMO\_DB **TO** **ROLE** metadata\_access\_role;

**grant** **OWNERSHIP** **on** future PROCEDURES **in** **database** DEMO\_DB **TO** **ROLE** metadata\_access\_role;

**grant** **OWNERSHIP** **on** future **FUNCTIONS** **in** **database** DEMO\_DB **TO** **ROLE** metadata\_access\_role;

**grant** **usage** **on** future **schemas** **in** **database** DEMO\_DB **to** **role** metadata\_access\_role;

## Useful Commands

1. Command to view definition of a database object

**Select** get\_ddl('view','emp\_vw')

1. Command to see permission granted to a user

**SHOW GRANTS TO USER infogix**

1. Command to see permission granted to a role

**SHOW GRANTS TO ROLE metadata\_access\_role**

1. Command to see all users created in the database

**SHOW USERS**

1. Commands to set a warehouse to a user

**ALTER USER dberman SET DEFAULT\_WAREHOUSE = COMPUTE\_WH**

1. Commands to assign default role to a user

**ALTER USER dberman set default\_role=** **metadata\_access\_role**

# Redshift

## Database inbuilt schemas

1. information\_schema
2. pg\_catalog
3. pg\_internal
4. public

## Database inbuilt catalog

No default inbuilt catalog.

## Commands to grant permission

1. Logged in user should be Admin or have permission to create user and grant permission
2. Command to create user password

select md5('dberman@123' || ' dberman ');

In the output of this query append ‘md5’, so password will be ‘md5’ concatenated with output of the above query.

Note:

dberman is the new user that we are going to create with password 'dberman@123'.

1. Command to create user

create user dberman password 'md546f892995c7400d12875dbfc3debd4b8';

Note:

Password of the user is md5 hash string generated at step 2.

1. Commands to grant permission to access Database Metadata node

grant select on all tables in schema public to dberman

grant execute on all procedures in schema public to dberman

grant execute on all functions in schema public to dberman

# HANA

## Database inbuilt schemas

|  |
| --- |
| 1. \_SYS\_TASK |
| 1. \_SYS\_TELEMETRY |
| 1. \_SYS\_STATISTICS |
| 1. \_SYS\_WORKLOAD\_REPLAY |
| 1. \_SYS\_SECURITY |
| 1. \_SYS\_REPO |
| 1. SAP\_XS\_USAGE |
| 1. \_SYS\_BIC |
| 1. \_SYS\_AUDIT |
| 1. SAP\_XS\_LM |
| 1. \_SYS\_EPM |
| 1. \_SYS\_AFL |
| 1. \_SYS\_BI |
| 1. \_SYS\_SQL\_ANALYZER |
| 1. SYS |
| 1. \_SYS\_RT |
| 1. SAP\_XS\_LM\_PE |
| 1. SAP\_REST\_API |
| 1. \_SYS\_XS |
| 1. \_SYS\_DATA\_ANONYMIZATION |
| 1. HANA\_XS\_BASE |
| 1. UIS |
| 1. \_SYS\_PLAN\_STABILITY 2. SYSTEM: SYSTEM is an inbuilt user / schema. In system schema, we can create database objects |

## Database inbuilt catalog

No default inbuilt catalog.

## Commands to grant permission

1. Logged in user should be Admin or have permission to create user and grant permission
2. Command to create new user

CREATE USER harvestortestuser PASSWORD Harvestortestuser01 NO FORCE\_FIRST\_PASSWORD\_CHANGE;

Note:

harvestortestuser is the new user that we are going to create with password ‘Harvestortestuser01'.

1. Commands to grant permission to access Database Metadata node

Option 1

1. Create a role “TestCatalogRole”

**CREATE** **ROLE** TestCatalogRole

1. Grant “TestCatalogRole” to user harvestertestuser

**GRANT** TestCatalogRole **TO** harvestertestuser;

1. Grant below permission to role “TestCatalogRole” to allow user to see Metadata Definition node

**GRANT** **CATALOG** **READ** **TO** TestCatalogRole

Option 2

Grant in built role “SAP\_INTERNAL\_HANA\_SUPPORT” to user harvestertestuser

**GRANT** SAP\_INTERNAL\_HANA\_SUPPORT **TO** harvestertestuser

## SAP HANA Administration console

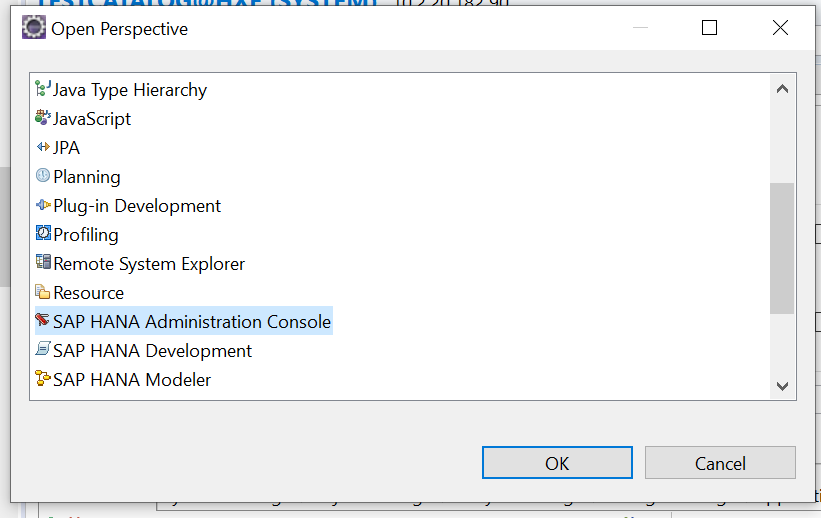
1. Install HANA Plugin into eclipse from below URL

[*https://tools.hana.ondemand.com/neon*](https://tools.hana.ondemand.com/neon)

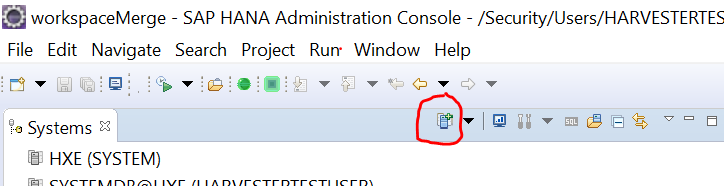
*For help, refer below youtube link for installing HANA plugin in eclipse*

<https://www.youtube.com/watch?v=lnVvmXs8x6Q>

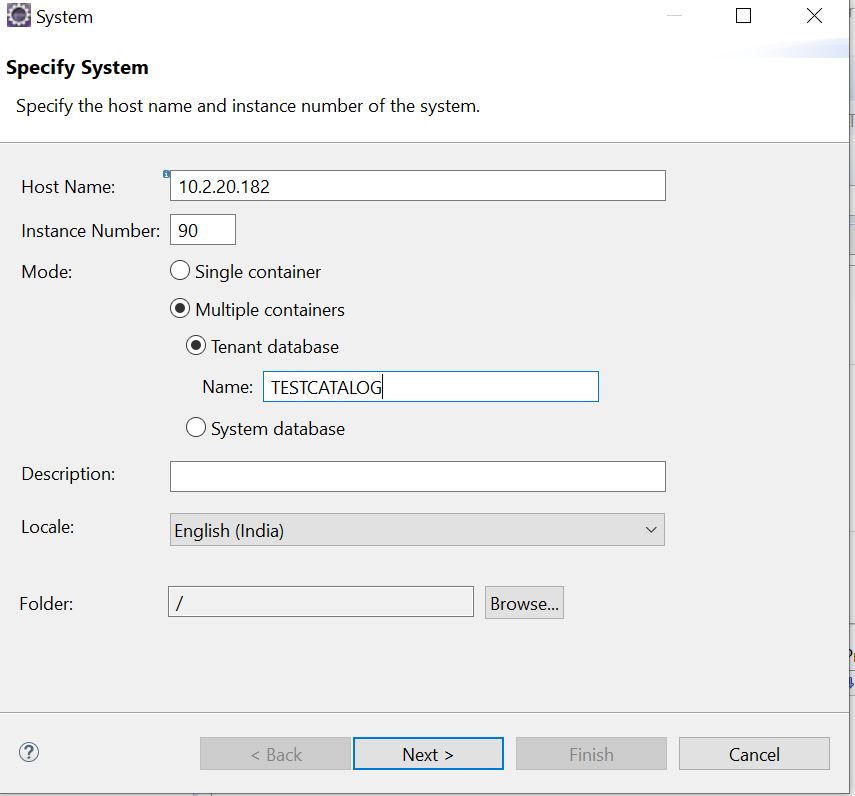
1. Open Perspective ‘SAP HANA Administration Console’



1. Create a database connection by clicking on “Add System” icon



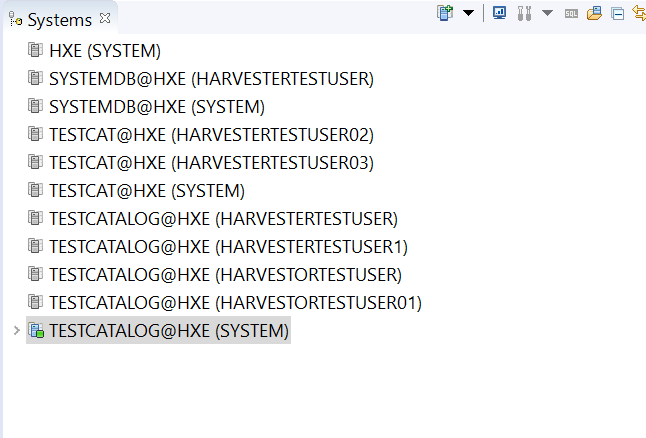
1. Enter the details as per below screenshot



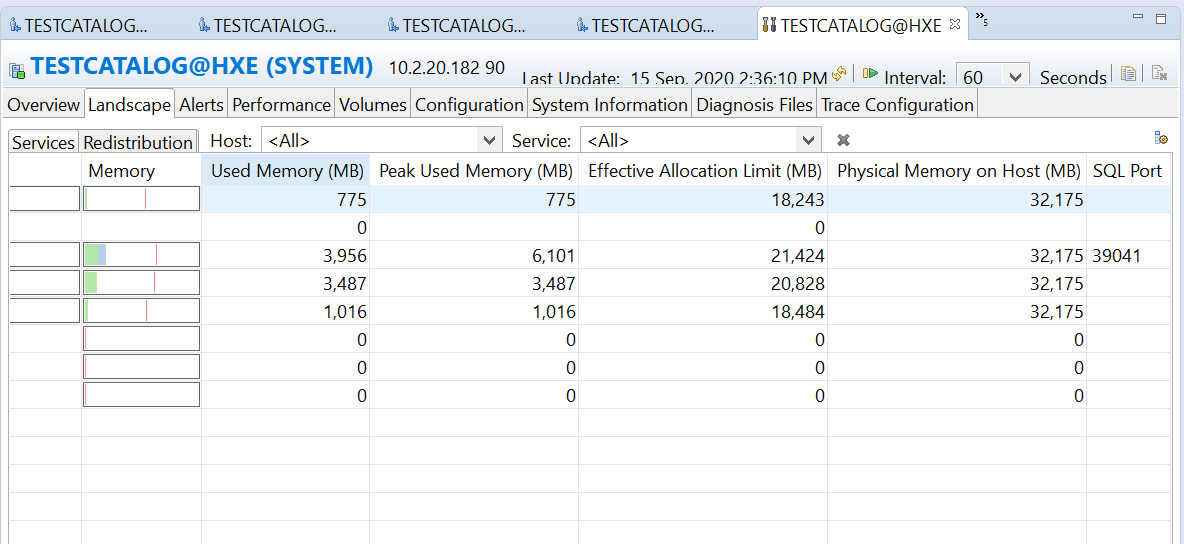
1. Enter user name as “system” and password as “Infogix@1240” and database connection will be completed

## Steps to find SQL port of a database

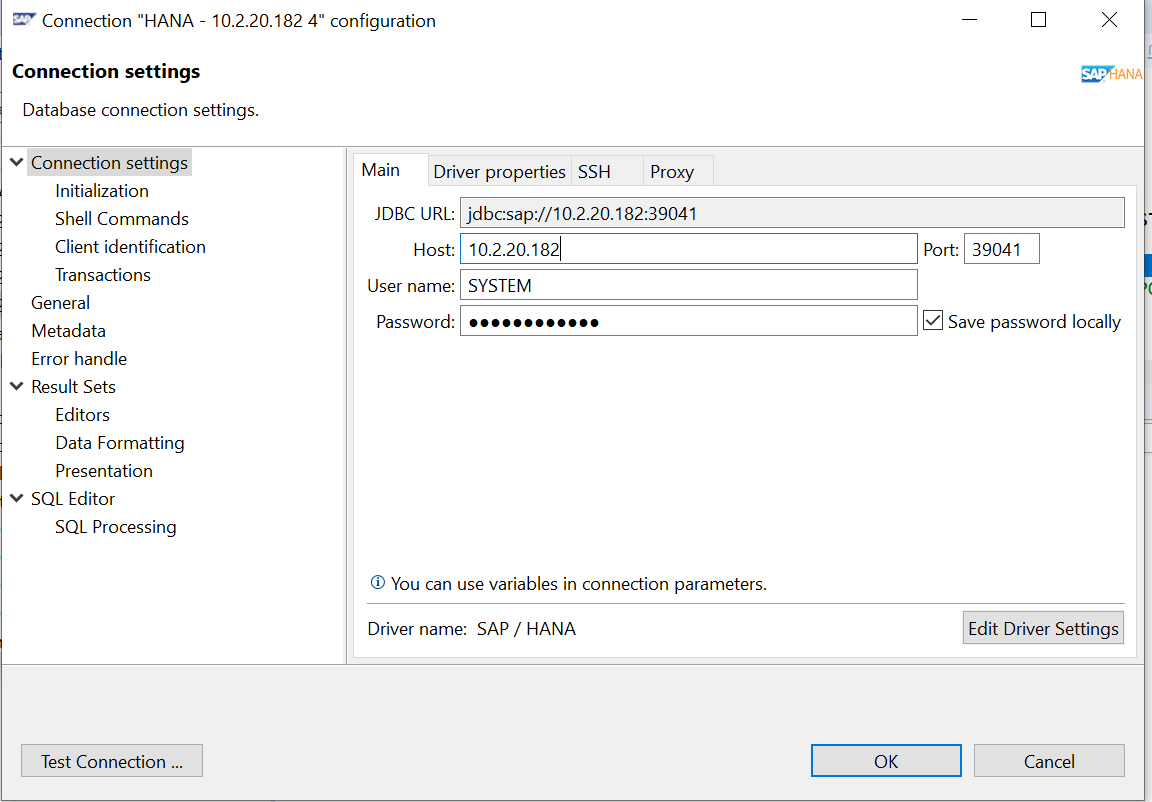
1. Click on Database. As per below screenshot, click on “TestCatalog” database



1. Click on “Landscape” tab. In “SQL Port” column, port of the database will be displayed. In this case port for “TestCatalog” database is 39041



1. Use this sql port to make database connection. Please find sample connection configuration in Dbeaver tool



# HIVE

## Database inbuilt schemas

It has no system defined schemas

## Database inbuilt catalog

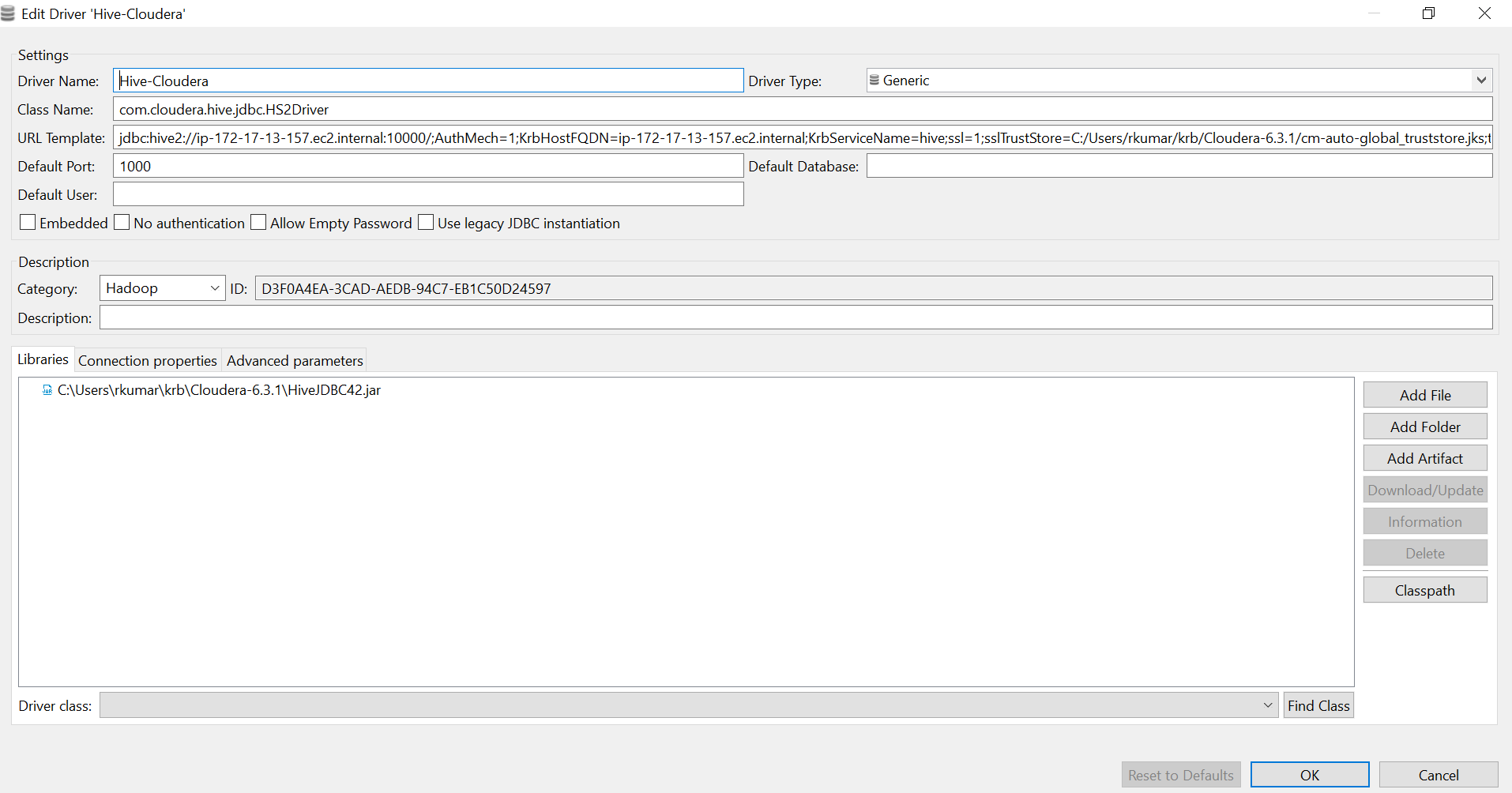
It doesn’t have support catalog.

## Commands to grant permission

It does not require any special permission

## Database Connection Information

DB URL: jdbc:hive2://ip-172-17-13-157.ec2.internal:10000/;AuthMech=1;KrbHostFQDN=ip-172-17-13-157.ec2.internal;KrbServiceName=hive;ssl=1;sslTrustStore=C:/Users/rkumar/krb/Cloudera-6.3.1/cm-auto-global\_truststore.jks;trustStorePwd=changeit



Schema Created: TestSchema and TestSchemaOther

User: harvestertestuser

Password: analyze

Note: Hive EC2 machine was not accepting password as “harvestertestuser” , so provided the password analyze

Connection to HIVE through "Dbeaver Commmunity" tool

* For accessing HIVE database in windows operating system machine, User needs to install "Dbeaver Commmunity" tool with version higher than or equal to 7
* For windows operating system, User needs to put below mapping in "C:\Windows\System32\drivers\etc\hosts" file

172.17.13.157 ip-172-17-13-157.ec2.internal

* Add below configuration in "dbeaver.ini" file present inside "Dbeaver Commmunity" installed folder

-Djavax.security.auth.useSubjectCredsOnly=false

-Dsun.security.krb5.debug=true

-Djava.security.krb5.debug=true

-Djava.security.krb5.conf=C:/Users/rkumar/krb/Cloudera-6.3.1/krb5.conf

-Djava.security.auth.login.config=C:/Users/rkumar/krb/Cloudera-6.3.1/gss-jaas.conf

-Djavax.net.ssl.trustStore=C:/Users/rkumar/krb/Cloudera-6.3.1/cm-auto-global\_truststore.jks

-Dcom.sun.jndi.ldap.object.disableEndpointIdentification=true

-Djavax.net.debug=all

Note:

* "C:/Users/rkumar/krb/Cloudera-6.3.1" is a user local machine folder where we have placed file "krb5.conf" , "gss-jaas.conf" , "cm-auto-global\_truststore.jks"
* In "gss-jaas.conf" file , replace path mentioned for file "sagacity.keytab" with path present in local machine carrying file "sagacity.keytab".

# Teradata

## Database inbuilt schemas

DBC, LockLogShredder, SQLJ, SYSJDBC, SYSBAR, SYSLIB, SYSSPATIAL, SYSUDTLIB, SYSUIF, Sys\_Calendar, SysAdmin, SystemFe, TDMaps, TDQCD, TDStats, TD\_SERVER\_DB, TD\_SYSFNLIB, TD\_SYSXML, tdwm

## Database inbuilt catalog

It doesn’t have support catalog.

## Commands to grant permission

It does not require any special permission

Note: In Object Definition, Procedure body is not displayed. It’s a known limitation.

## Database Connection Information

DB URL: teradata.eng.infogix.com

User Created: harvestertestuser and harvestertestuserother

Below are the user details

* User: harvestertestuser

Password: harvestertestuser

* User: harvestertestuserother

Password: harvestertestuserother

Note: In Teradata, user behaves same as schema. We have created the objects in these users

Below user is a test user which can be used to run Database Metadata node. It has no objects and no special access permission

* User: harvestertestuserAnalyze

Password: harvestertestuserother