

Db2 13 for z/OS

SQL Data Insights User Guide



Notes

Before using this information and the product it supports, be sure to read the general information under "Notices" at the end of this information.

Subsequent editions of this PDF will not be delivered in IBM Publications Center. Always download the latest edition from IBM Documentation.

2022-06-16 edition

This edition applies to Db2[®] 13 for z/OS[®] (product number 5698-DB2[®]), Db2 13 for z/OS Value Unit Edition (product number 5698-DBV), and to any subsequent releases until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Specific changes are indicated by a vertical bar to the left of a change. A vertical bar to the left of a figure caption indicates that the figure has changed. Editorial changes that have no technical significance are not noted.

© **Copyright International Business Machines Corporation 2022, 2022.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

- About this information..... V**
 - Who should read this information.....v
 - Terminology and citations.....v
 - Accessibility features for Db2 for z/OS.....vi
 - How to send your comments about Db2 for z/OS documentation.....vi

- Chapter 1. SQL Data Insights..... 1**

- Chapter 2. Preparing SQL Data Insights installation.....3**
 - Configuring setup user ID for SQL Data Insights..... 5
 - Configuring user authentication for SQL Data Insights.....8
 - Configuring Db2 for SQL Data Insights..... 9
 - Db2 catalog tables for SQL Data Insights..... 10

- Chapter 3. Installing and configuring SQL Data Insights.....19**

- Chapter 4. Verifying the installation and configuration of SQL Data Insights..... 23**

- Chapter 5. Administering SQL Data Insights..... 25**
 - Modifying your SQL Data Insights settings..... 25
 - Creating a started task for the SQL Data Insights application..... 26
 - Creating started tasks for the Spark cluster..... 27

- Chapter 6. Creating a connection to Db2.....29**

- Chapter 7. Adding an AI object.....31**

- Chapter 8. Enabling AI query..... 33**

- Chapter 9. Viewing an AI object model..... 35**

- Chapter 10. Running an AI query.....37**

- Chapter 11. Analyzing data..... 39**

- Information resources for Db2 for z/OS and related products.....41**

- Notices.....43**
 - Trademarks..... 44
 - Terms and conditions for product documentation..... 44
 - Privacy policy considerations..... 45

- Glossary.....47**

- Index..... 49**

About this information

Throughout this information, "Db2" means "Db2 13 for z/OS". References to other Db2 products use complete names or specific abbreviations.

Important: To find the most up to date content for SQL data insights, always use the online product documentation in [IBM Documentation](#) or download the latest PDF file from [PDF format manuals for Db2 13 for z/OS](#).

For more about how to use this information, see ["About this information"](#) in the online product documentation.

Who should read this information

This information is for data scientists and data engineers who want to enable and run AI queries against Db2 data to extract hidden patterns and derive business insights.

This information is also for Db2 application architects and developers who are familiar with Structured Query Language (SQL) and who want to develop AI-based applications that can quickly analyze complex Db2 data for explainable insights in real time.

Terminology and citations

When referring to a Db2 product other than Db2 for z/OS, this information uses the product's full name to avoid ambiguity.

The following terms are used as indicated:

Db2

Represents either the Db2 licensed program or a particular Db2 subsystem.

IBM OMEGAMON® for Db2 Performance Expert on z/OS

Refers to any of the following products:

- IBM® IBM OMEGAMON for Db2 Performance Expert on z/OS
- IBM Db2 Performance Monitor on z/OS
- IBM Db2 Performance Expert for Multiplatforms and Workgroups
- IBM Db2 Buffer Pool Analyzer for z/OS

C, C++, and C language

Represent the C or C++ programming language.

CICS®

Represents CICS Transaction Server for z/OS.

IMS

Represents the IMS Database Manager or IMS Transaction Manager.

MVS™

Represents the MVS element of the z/OS operating system, which is equivalent to the Base Control Program (BCP) component of the z/OS operating system.

RACF®

Represents the functions that are provided by the RACF component of the z/OS Security Server.

Accessibility features for Db2 for z/OS

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use information technology products successfully.

Accessibility features

The following list includes the major accessibility features in z/OS products, including Db2 for z/OS. These features support:

- Keyboard-only operation.
- Interfaces that are commonly used by screen readers and screen magnifiers.
- Customization of display attributes such as color, contrast, and font size

Tip: IBM Documentation (which includes information for Db2 for z/OS) and its related publications are accessibility-enabled for the IBM Home Page Reader. You can operate all features using the keyboard instead of the mouse.

Keyboard navigation

For information about navigating the Db2 for z/OS ISPF panels using TSO/E or ISPF, refer to the *z/OS TSO/E Primer*, the *z/OS TSO/E User's Guide*, and the *z/OS ISPF User's Guide*. These guides describe how to navigate each interface, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

Related accessibility information

IBM and accessibility

See the *IBM Accessibility Center* at <http://www.ibm.com/able> for more information about the commitment that IBM has to accessibility.

How to send your comments about Db2 for z/OS documentation

Your feedback helps IBM to provide quality documentation.

Send any comments about Db2 for z/OS and related product documentation by email to db2zinfo@us.ibm.com.

To help us respond to your comment, include the following information in your email:

- The product name and version
- The address (URL) of the page, for comments about online documentation
- The book name and publication date, for comments about PDF manuals
- The topic or section title
- The specific text that you are commenting about and your comment

Chapter 1. SQL Data Insights

SQL Data Insights (SQL DI) is an AI-powered feature of Db2 for z/OS. SQL DI combines deep learning with advanced IBM Z technologies to infuse the Db2 engine with SQL-based semantic queries on tables and views.

SQL DI leverages deep learning in AI, exploits new IBM Z processors, and extends the standard SQL language to enhance the traditional data processing in a relational database. It extrapolates unsupervised learning to train neural network models for discovering, matching, and grouping records with similarities, dis-similarities, and clusters in Db2 data. For example, by learning from a large amount of training data, SQL DI can infer hidden “look-like” relationships across two different records that are traditionally not considered an exact match. This industry-first ability of inferring hidden information in a relational database enables SQL DI to detect and extract the semantic similarities between the two records.

SQL DI is seamlessly integrated into your Db2 for z/OS environment. The feature is comprised of a data and query engine, a set of machine learning libraries, and a machine learning web application. The IBM Z Deep Neural Network Library (zDNN) stack resides natively in z/OS, which enables SQL DI to take full advantage of IBM Z processors. The data and query engine is built into Db2 for z/OS and provides services for processing data and semantic queries. The core of this processing engine is the new AI_SIMILARITY, AI_SEMANTIC_CLUSTER, and AI_ANALOGY scalar functions. You can use these built-in functions in SQL statements to ask semantic questions about your data. Leveraging the analytic framework of IBM Watson[®] Machine Learning for z/OS and the runtime engines of z/OS Spark, the SQL DI web application provides the interface for you to create AI objects, enable AI queries, train models, and visualize query results.

With SQL DI, you can easily uncover the hidden relationships across tables and views in your Db2 data, quickly gain actionable insights for solving your business problems as they arise, and confidently arrive at better business decisions. You can achieve all these without the costly effort of moving the data across platforms, procuring any machine learning infrastructure, acquiring advanced AI skills, and analyzing expansive data.

Related information

[Db2 built-in scalar functions](#)

Chapter 2. Preparing SQL Data Insights installation

SQL Data Insights (SQL DI) has specific system, network, user access, and security requirements. Plan ahead and prepare your environment for SQL DI installation by satisfying all the requirements.

Hardware and software requirements

The SQL DI feature requires the following hardware, software, and integrated development tools. If you decide to enable the feature, make sure that you meet all the prerequisites. Consider planning the system requirements for both your Db2 system and the SQL DI feature together, particularly if you decide to install the SQL DI feature on the same LPAR where your Db2 system runs.

- IBM z16, z15™, z14, z13®, or zEnterprise® EC12 system.

For best performance, consider running SQL DI on the latest models of Z.

- IBM z/OS 2.5 or 2.4.

Verify that data set SYS1.SIEALNKE and CEE.SCEERUN2 are APF authorized and accessible by Db2. See [z/OS 2.5 program directory](#) or [z/OS 2.4 program directory](#) for instructions.

- IBM Z Deep Neural Network Library (zDNN), including the IBM Z AI Optimization Library and the IBM Z AI Data Embedding Library:
 - For z/OS 2.5 with APARs OA62901, OA62902, and OA62903.
 - For z/OS 2.4 with APARs OA62849, OA62886, and OA62887.
- z/OS Supervisor with APAR OA62728 for both z/OS 2.5 and 2.4.
- IBM OpenBLAS with APARs PH45672 (for z/OS 2.5), PH45663 (for z/OS 2.4), and PH44479 (for both z/OS 2.5 and 2.4).
- IBM Db2 13 for z/OS (5698-DB2 or 5698-DBV).
- z/OS OpenSSH. See [z/OS OpenSSH](#) for instructions.
- IBM 64-bit SDK for z/OS Java™ Technology Edition Version 8 SR7 FP6 or later.

System capacity requirements

System capacity for SQL DI varies based on several key workload factors, including the size of source data, the number of unique values, and the data type of selected columns. As the number of rows and columns increase, more CPU, memory, and storage are required for enabling and running AI queries. The number of unique column values and the size of an AI object model correspond proportionally. The more distinct column values there are, the larger the resulting object model becomes and the more system resources are needed for training the model.

Take for example the system resource usage for processing a small SQL DI AI object. The AI object is 2.2 GB in size with 26 columns and 10 million rows. While 14 columns are of the SQL DI numeric data type, the remaining 12 columns are of the categorical type. It requires 8 threads on 10 CPUs, up to 17 GB of memory, and 20 GB file system storage to enable the object for AI query while achieving adequate performance goals. The total of 4 million unique values contributes to the final size of the resulting model, which requires 13 GB of disk space in the Db2 storage group.

Network requirements

SQL DI requires dedicated networks and ports for communications across systems and services. Some of the ports are predefined while others can be user-defined. You must configure the following ports in your firewall before the SQL DI installation.

System or service	Port number	Outbound system	Inbound system	Default address space
SQL DI UI	15001 or user-defined	Your network	z/OS system	SQLDAPPS
z/OS Spark master	7077 or user-defined	z/OS system	z/OS Spark system	SQLDSPKM
z/OS Spark master REST API	6066 or user-defined	z/OS system	z/OS Spark system	SQLDSPKM
z/OS Spark master UI	8080 or user-defined	Your network	z/OS Spark system	SQLDSPKM
z/OS Spark worker	System-assigned* or user-defined	z/OS system	z/OS Spark system	SQLDSPKW
z/OS Spark worker UI	8081 or user-defined	Your network	z/OS Spark system	SQLDSPKW
z/OS Spark driver	System-assigned* or user-defined	z/OS system	z/OS Spark system	SQLDSPKD
z/OS Spark block manager	System-assigned* or user-defined	z/OS system	z/OS Spark system	SQLDSPKX
z/OS driver-specific port for Spark block manager	System-assigned* or user-defined	z/OS system	z/OS Spark system	SQLDSPKD

Notes:

* If you manage port assignments and access in your sysplex, do not use system-assigned port numbers for Spark worker, Spark driver, Spark block manager, or z/OS driver-specific port for Spark block manager. Also, a Spark cluster requires a port range, instead of a single port, at runtime. The actual range depends on the specified maximum number of retries for binding to a port.

MVS resource workload requirements

When you enable an AI object for AI query, SQL DI creates and trains a model for the object. Model training can consume all the resources available for your OMVS subsystem. Consider defining your SQL DI workload in z/OS Workload Manager (WLM) and assign a service class for Spark associated with this workload. For the service class, specify the default qualifier names SQLD% and SQLDAPPS with your performance goals and resource requirements for the workload. Also, consider assigning the service class for your SQL DI workload a lower priority than for your Db2 workloads. See [z/OS workload management for Apache Spark](#) for more information.

Db2 configuration requirements

To enable SQL Data Insights (SQL DI), you must customize and submit the DSNTIJAI job to create the required database and pseudo-catalog tables in Db2 for z/OS. See [“Configuring Db2 for SQL Data Insights” on page 9](#) for instructions.

Setup user ID requirements

The installation and configuration of SQL Data Insights (SQL DI) requires that you have sufficient privileges to access your z/OS system, allocate system resources, and customize system environment variables. Consider creating a multipurpose SQL DI setup user ID, grant it required permissions, and

customize your z/OS environment for it. See [“Configuring setup user ID for SQL Data Insights”](#) on page 5 for instructions.

User authentication requirements

SQL Data Insights (SQL DI) uses SSL to secure network communications and RACF to authenticate users. You must configure a RACF keyring and an associated keystore (JCERACFKS) to manage your SSL certificates and SQL DI user authentication. See [“Configuring user authentication for SQL Data Insights”](#) on page 8 for instructions.

Browser requirements

SQL DI features a web-based user interface (UI). Make sure that you run the UI with the following standard or desktop version of Mozilla Firefox and Google Chrome:

- Firefox version 54 or higher
- Chrome version 60 or higher.

Related tasks

[“Installing and configuring SQL Data Insights”](#) on page 19

The installation and configuration of SQL Data Insights (SQL DI) involve a script-driven sequence of interactive tasks. Make sure that you follow the step-by-step instruction and successfully complete each task.

[“Verifying the installation and configuration of SQL Data Insights”](#) on page 23

Before you and your business start to use SQL Data Insights (SQL DI), complete a quick procedure to verify that the feature is properly installed and configured.

Configuring setup user ID for SQL Data Insights

The installation and configuration of SQL Data Insights (SQL DI) requires that you have sufficient privileges to access your z/OS system, allocate system resources, and customize system environment variables. Consider creating a multipurpose SQL DI setup user ID, grant it required permissions, and customize your z/OS environment for it.

Procedure

1. If you have not done so, create a multipurpose `<sqldi_setup_userid>`, which you will use to install, configure, and run your SQL DI.

You can create the required `<sqldi_setup_userid>` in different ways. For example, you can customize and run the following sample JCL job to create the ID:

```
//CREATE JOB (0),SQLDI RACF',CLASS=A,REGION=0M,
//MSGCLASS=H,NOTIFY=&SYSUID
//*-----*/
//RACF      EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
ADDGROUP SQLDIGRP OMVS(GID(<group-identifier>)) OWNER(SYS1)
ADDUSER  <sqldi_setup_userid> DFLTGRP(SQLDIGRP) OMVS(UID(<user-identifier>)) -
HOME(/u/<sqldi_setup_userid>) -
PROGRAM($SQLDI_INSTALL_DIR/tools/bin/bash) -
NAME('SQLDI ID') PASSWORD(<password>) NOIDCARD
/*
```

where

- `<sqldi_setup_userid>` is the user ID that you will use to configure and run your SQL DI.
- `SQLDIGRP` is a RACF group that you will use to associate SQL DI users and manage their access.
- `<group-identifier>` is the identifier for `SQLDIGRP`.

- `<user-identifier>` is the identifier for `<sqldi_setup_userid>`. Do not use UID 0 for `<sqldi_setup_userid>`.
 - `$SQLDI_INSTALL_DIR` is the directory where SQL DI is installed. The default is `/usr/lpp/IBM/db2sqldi/v1r1`.
2. Allocate a minimum of 500 MB disk space to the home directory for `<sqldi_setup_userid>`.
 3. Create a `SQLDI_HOME` directory where SQL DI will store all the configuration, customization, and log files as well as runtime data.
 - a. Create the `SQLDI_HOME` directory. Make sure that `SQLDI_HOME` is mounted to a zFS file system with at least 100 GB storage available.

Tip: Avoid creating or configuring the `SQLDI_HOME` directory with automount management. Automount might unmount a directory if it is not referenced for a period of time. Any unplanned unmount of the `SQLDI_HOME` directory will cause SQL DI to fail.
 - b. If you use another user ID to create the `SQLDI_HOME` directory, make sure to change the directory owner to `<sqldi_setup_userid>` by issuing the following command:

```
chown -R sqldi_setup_userid:SQLDIGRP SQLDI_HOME/
```

- c. To allocate zFS data sets for `SQLDI_HOME` that are larger than 100 GB, make sure that you specify the DFSMS data class with extended format and extended addressability.
4. Configure your z/OS UNIX shell environment for `<sqldi_setup_userid>`
 - a. Copy the `$SQLDI_INSTALL_DIR/templates/profile.template` directory into `$HOME/.profile` for `<sqldi_setup_userid>`.
 - b. Customize the following environment variables in the profile template:
 - Set `$JAVA_HOME` to the directory of your IBM Java 8 SR7 installation.
 - Set `$SQLDI_INSTALL_DIR` to the directory where your SQL DI is installed. The default is `/usr/lpp/IBM/db2sqldi/v1r1`.
 - Set `$ZADE_INSTALL_DIR` to the directory where the IBM Z AI Data Embedding Library is located on your z/OS system. The default is `/usr/lpp/IBM/aie/zade`.
 - Set `$BLAS_INSTALL_DIR` to the directory where the IBM OpenBLAS is located on your z/OS system. The default is `/usr/lpp/cbclib`.
 - c. Verify that the `PATH` environment variable is correctly set in the profile template as shown in the following example:

```
PATH=/bin:
PATH=$SQLDI_INSTALL_DIR/sql-data-insights/bin:$PATH
PATH=$SQLDI_INSTALL_DIR/tools/bin:$PATH
PATH=$PATH:$JAVA_HOME/bin
PATH=$ZADE_INSTALL_DIR/zade/bin:$PATH
export PATH=$PATH
```

Where `/tools/bin` is home to `bash`, `OpenSSL`, and other tools.

5. Configure `<sqldi_setup_userid>` access to your z/OS UNIX shell environment.

`<sqldi_setup_userid>` must have the following permissions to install, configure, and run your SQL DI:

 - Permission to read and write to the `SQLDI_HOME` directory.
 - Permission to read and execute to the `$SQLDI_INSTALL_DIR` directory used by the SMP/E installation.
 - `$JAVA_HOME/bin` defined in the `$PATH` environment variable in the user's profile.
 - `$IBM_JAVA_OPTIONS` environment variable set to `-Dfile.encoding=UTF-8` in the user's profile.
 - `$_BPXK_AUTOCVT` environment variable set to `ON` in the user's profile.

- READ access to the RACF BPX .JOBNAME facility class so that SQL DI default address space names can take effect and that you can assign default job names with the SQLDI prefix to SQL DI started services.
 - READ access to resources CSFDSG, CSFDSV, CSFEDH, CSFIQA, CSFIQF, CSFOWH, CSFPKG, CSFPKI, CSFPKX, CSFRNG, and CSFRNGL for ICSF services in the CSFSERV class if your system is CryptoCard-enabled.
6. Update system resource settings, including CPUTIMEMAX, MEMLIMIT, and ASSIZEMAX values in the OMVS segment of the RACF profile for `<sqldi_setup_userid>`.

If needed, issue the **ALTUSER** command to update the CPUTIMEMAX, MEMLIMIT, and ASSIZEMAX settings as shown in the following example:

```
ALTUSER <sqldi_setup_userid> OMVS(ASSIZEMAX(address-space-size)
    MEMLIMIT(nonshared-memory-size) CPUTIMEMAX(cpu-time))
```

SQL DI requires sufficient system memory to function properly. You can use the MEMLIMIT and ASSIZEMAX parameters to control the amount of memory for the address space started by `<sqldi_setup_userid>`. At the minimum, set MEMLIMIT initially to 32 GB or greater and ASSIZEMAX to 1 GB.

SQL DI also requires sufficient system CPU to run unimpeded. Consider setting the CPUTIMEMAX parameter to unlimited to ensure uninterrupted operations.

You can issue the **ulimit** command in a z/OS UNIX shell session to verify CPUTIMEMAX, MEMLIMIT, and ASSIZEMAX settings. The command returns a message that is similar to the following example:

```
/bin/ulimit -a
core file          8192b
cpu time           unlimited
data size         unlimited
file size         unlimited
stack size        unlimited
file descriptors  520000
address space     1048576k
memory above bar  24576m
```

Where

- "cpu time" is the value of the CPUTIMEMAX parameter.
- "address space" is the value of the ASSIZEMAX parameter.
- "memory above bar" is the value of the MEMLIMIT parameter.

See [ALTUSER \(Alter user profile\)](#) and [ulimit](#) for more information.

7. Verify that the required Java is installed on the z/OS system where you will install the SQL DI and available to `<sqldi_setup_userid>`.

Related tasks

[“Installing and configuring SQL Data Insights” on page 19](#)

The installation and configuration of SQL Data Insights (SQL DI) involve a script-driven sequence of interactive tasks. Make sure that you follow the step-by-step instruction and successfully complete each task.

[“Configuring user authentication for SQL Data Insights” on page 8](#)

SQL Data Insights (SQL DI) uses SSL to secure network communications and RACF to authenticate users. You must configure a RACF keyring and an associated keystore (JCERACFKS) to manage your SSL certificates and SQL DI user authentication.

Configuring user authentication for SQL Data Insights

SQL Data Insights (SQL DI) uses SSL to secure network communications and RACF to authenticate users. You must configure a RACF keyring and an associated keystore (JCERACFKS) to manage your SSL certificates and SQL DI user authentication.

Before you begin

A RACF keyring is a set of digital certificates, private keys, and key mappings that defines your network trust policy, and a RACF keystore (JCERACFKS) collects and manages all the keyrings. To configure a RACF keystore, make sure that you grant your user ID with the RACF `SPECIAL` authority or sufficient authority as described in [RACDCERT](#) command.

Procedure

1. Create a keyring by issuing the following RACF command:

```
RACDCERT ADDRING(SQLDIRG) ID(SQLDIID)
```

Where SQLDIID is the owner of the RACF keyring.

2. Generate a CA (certificate authority) certificate by issuing the following command:

```
RACDCERT GENCERT CERTAUTH +  
SUBJECTSDN( +  
  CN('PLEXE2') +  
  C('US') +  
  SP('CA') +  
  L('SAN JOSE') +  
  O('IBM') +  
  OU('SQLDI') +  
) +  
ALTNAME( +  
  EMAIL('user1@ibm.com') +  
) +  
WITHLABEL('SQLDICert') +  
NOTAFTER(DATE(2030/01/01))
```

If you decide to use an existing CA-signed certificate used by your business, make sure that you add and import the root CA certificate to RACF. See instructions in [RACDCERT](#) command for using the **RACDCERT ADD** and **RACDCERT IMPORT** commands.

3. Generate and sign a user certificate for `<sqldi_setup_userid>` by issuing the following command:

```
RACDCERT GENCERT ID(SQLDIID) +  
SUBJECTSDN( +  
  CN('PLEXE2') +  
  C('US') +  
  SP('CA') +  
  L('SAN JOSE') +  
  O('IBM') +  
  OU('SQLDI') +  
) +  
ALTNAME( +  
  EMAIL('user1@ibm.com') +  
) +  
WITHLABEL('SQLDICert_SQLDIID') +  
SIGNWITH(CERTAUTH LABEL('SQLDICert')) +  
NOTAFTER(DATE(2022/01/01))
```

Where SQLDIID is the owner of the RACF keyring.

4. Connect the user certificate and the CA certificate to the keyring you created and add usage options by issuing the following commands:

```
RACDCERT ID(SQLDIID) CONNECT(CERTAUTH LABEL('SQLDICert') +
RING(SQLDIRG))
```

```
RACDCERT ID(SQLDIID) CONNECT(ID(SQLDIID) LABEL('SQLDICert_SQLDIID') +
RING(SQLDIRG) USAGE(PERSONAL))
```

5. Grant `<sqldi_setup_userid>` permission to access the keyring and the CA certificate.

`<sqldi_setup_userid>` must have the READ or UPDATE authority to the IRR.DIGTCERT.LISTRING resource in the FACILITY class. While the READ access enables the listing of your own keyring, the UPDATE access enables the listing of keyrings owned by others.

Issue the following commands:

```
RDEFINE FACILITY IRR.DIGTCERT.LIST UACC(NONE)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(<sqldi_setup_userid>) ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

`<sqldi_setup_userid>` must also have the READ or UPDATE authority to the `<ringOwner>.<ringName>.LST` resource in the RDATALIB class. While the READ access enables the retrieval of your private keys, the UPDATE access enables the retrieval of keys by others.

Issue the following commands:

```
RDEFINE RDATALIB SQLDIID.SQLDIRG.LST UACC(NONE)
SETROPTS CLASSACT(RDATALIB) RACLIST(RDATALIB)
SETROPTS CLASSACT(RDATALIB)
PERMIT SQLDIID.SQLDIRG.LST CLASS(RDATALIB) ID(<sqldi_setup_userid>) ACCESS(READ)
SETROPTS RACLIST(RDATALIB) REFRESH
```

See “[Configuring setup user ID for SQL Data Insights](#)” on page 5 for the full list of the privileges required for `<sqldi_setup_userid>`.

Related tasks

“[Installing and configuring SQL Data Insights](#)” on page 19

The installation and configuration of SQL Data Insights (SQL DI) involve a script-driven sequence of interactive tasks. Make sure that you follow the step-by-step instruction and successfully complete each task.

Related information

[RACDCERT command](#)

Configuring Db2 for SQL Data Insights

To enable SQL Data Insights (SQL DI), you must customize and submit the DSNTIJAI job to create the required database and pseudo-catalog tables in Db2 for z/OS.

Before you begin

The DSNUTILU stored procedure must be configured in Db2. See [DSNUTILU stored procedure](#).

Note: DSNTIJAI uses program DSNTIAD, the package of which must be bound with APPLCOMPAT V12R1M500 or higher in order to run.

Procedure

1. Copy and customize the DSNTIJAI sample job member in the Db2 SDSNSAMP library according to your needs.

When you create STOGROUP DSNAIDSG in the DSNTIJAI sample job, use a catalog alias for the VCAT option. Make sure that the alias is assigned to SMS-managed data sets that have allocation for extended format and extended addressability. The allocation will accommodate models that are larger than 4 GB in size.

2. Submit DSNTIJAI.

Note: The last step, DSNTIAI4, of the DSNTIJAI job grants the necessary database permissions to SQL DI users. You must repeat step DSNTIAI4 for each user.

3. Review the load template and add permissions.

This step involves loading a Db2 table. SQL DI provides a template with the utility control statements for the zLoad process.

Follow the instructions described in step “3” on page 25 of “[Modifying your SQL Data Insights settings](#)” on page 25 to review the load template file. Ensure that SQL DI users have access to the data sets referred to in the template, and that adequate temporary space has been allocated, based on the size of your source table.

Note: The default high-level qualifier (HLQ) specified in the TEMPLATE statements in the load template is the z/OS job name indicated by the "&JO" variable. Review whether it is appropriate for SQL DI users to run the zLOAD utility in your environment.

Related tasks

[“Installing and configuring SQL Data Insights” on page 19](#)

The installation and configuration of SQL Data Insights (SQL DI) involve a script-driven sequence of interactive tasks. Make sure that you follow the step-by-step instruction and successfully complete each task.

[“Modifying your SQL Data Insights settings” on page 25](#)

During the installation and configuration, default values are automatically set to some of your SQL Data Insights (SQL DI) parameters, including the minimum amount of memory to run Spark jobs and the maximum number of rows to load for AI queries. Depending on the size of your data and the need of your workload, you can modify the default settings on the Settings page of the SQL DI user interface.

Db2 catalog tables for SQL Data Insights

When you run the sample DSNTIJAI job to configure your Db2 system for SQL DI, the job creates a set of pseudo-catalog tables that are used to record and store metadata for AI objects, object models, and tables.

The sample DSNTIJAI job creates the following Db2 pseudo-catalog tables, table spaces, and indexes for SQL DI:

Table 1. Db2 pseudo catalog tables, table spaces, and indexes for SQL DI

Table name	Description	Table space	Indexes	Index field
SYSAIDB.SYSAIOBJETS	Contains a row for each Db2 table or view you select for an SQL DI AI object	SYSTSAIO	SYSAIOBJECTSIX1 SYSAIOBJECTSIX2	OBJECT_ID SCHEMA, NAME
SYSAIDB.SYSAICONFIGURATIONS	Contains a row for each column that you select for a column configuration	SYSTSAIC	SYSAICONFIGURATION_SIX1	CONFIGURATION_ID
SYSAIDB.SYSAICOLUMNCONFIG	Contains a row for each column and related attributes within a column configuration	SYSTSAID	SYSAICOLUMNCONFIGIX1	CONFIGURATION_ID, COLUMN_NAME, COLUMN_AISQL_TYPE

Table 1. Db2 pseudo catalog tables, table spaces, and indexes for SQL DI (continued)

Table name	Description	Table space	Indexes	Index field
<u>S</u> SYS AIDB.SYSAIMODEL	Contains a row for each AI object model and related table and state information	SYSTSAIM	SYSAIMODELSIX1	MODEL_ID
<u>NCENTERS</u> SYS AIDB.SYSAICOLUM	Contains a row for each column centroid for a trained model	SYSTSAIE	SYSAICOLUMNCENTER SIX2	MODEL_ID , COLUMN_ NAME, CENTROID
<u>NGJOBS</u> SYS AIDB.SYSAITRAINI	Contains a row for each training job that you initiate and job status information	SYSTSAIT	SYSAITRAININGJOB SIX1 SIX2	TRAINING_ JOB_ID OBJECT_I D, CONFIGU RATION_I D, MODEL_ID

SYSAIDB.SYSAIOBJECTS

The SYSAIDB.SYSAIOBJECTS table contains a row for each Db2 table or view you select for an SQL DI AI object.

Column name	Data type	Description	Usage
OBJECT_ID	BIGINT NOT NULL	A unique identifier for the AI object.	
OBJECT_NAME	VARCHAR(32)	A user-defined name for the AI object.	
OBJECT_TYPE	CHAR(1)	An identifier that identifies a Db2 table or view: T Specifies a Db2 table V Specifies a Db2 view	
SCHEMA	VARCHAR(128) NOT NULL	The schema of the AI object.	
NAME	VARCHAR(128) NOT NULL	The name of the AI object.	

Column name	Data type	Description	Usage
STATUS	VARCHAR(16) NOT NULL	The status of the AI query enabling process: Enabled Indicates that the AI object is enabled with AI query and that the row for the AI object model is populated. Disabled Indicates that the AI object is not enabled with AI query and that the row for the AI object model is not populated. Training Indicates that the AI object is being enabled with AI query and that the row for the AI object model is being updated. Failed Indicates that the AI query enabling process for the AI object failed.	
CONFIGURATION_ID	BIGINT	The identifier for the configuration used for the active model. A null value indicates that there is no active configuration yet.	
MODEL_ID	BIGINT	The identifier for the active model. A null value indicates that there is no active model table created yet.	
CREATED_BY	VARCHAR(32) (With SESSION_USER as default)	The SQLID of the user to which the object is registered.	
CREATED_DATE	VARCHAR(32) (With CURRENT TIMESTAMP as default)	The timestamp when the object was registered.	
LAST_UPDATED_BY	VARCHAR(32) (With SESSION_USER as default)	The SQLID of the user who last updated the object.	
LAST_UPDATED_DATE	TIMESTAMP (With ROW CHANGE TIMESTAMP as default)	The timestamp when the object was last updated.	
DESCRIPTION	VARCHAR(256)	A user-specified description of the object.	

SYSAIDB.SYSAICONFIGURATIONS

The SYSAIDB.SYSAICONFIGURATIONS table contains a row for each column that you select for a column configuration.

Column name	Data type	Description	Usage
CONFIGURATION_ID	BIGINT NOT NULL	A unique identifier for this configuration.	
NAME	VARCHAR(32)	A user-defined name for the configuration.	

Column name	Data type	Description	Usage
OBJECT_ID	BIGINT NOT NULL	An identifier of the object for which this configuration is created.	
RETRAIN_INTERVAL	INTEGER	The interval at which retraining occurs.	
KEEP_ROWIDENTIFIER_KEY	CHAR(1) NOT NULL	An indicator for the presence of the row identifier key in a model: Y Indicates that the row identifier key is kept in the model. N Indicates that the row identifier key is not kept in the model.	
NEGLECT_VALUES	VARCHAR(1024)	A semicolon-separated string of values to be treated as null in the model.	
CREATED_BY	VARCHAR(32) (With SESSION_USER as default)	The SQLID of the user to which the object is registered.	
CREATED_DATE	VARCHAR(32) (With CURRENT_TIMESTAMP as default)	The timestamp when the object was registered.	
LAST_UPDATED_BY	VARCHAR(32) (With SESSION_USER as default)	The SQLID of the user who last updated the object.	
LAST_UPDATED_DATE	TIMESTAMP (With ROW CHANGE_TIMESTAMP as default)	The timestamp when the object was last updated.	

SYSAIDB.SYSAICOLUMNCONFIG

The SYSAIDB.SYSAICOLUMNCONFIG table contains a row for each column and related attributes within a column configuration.

Column name	Data type	Description	Usage
CONFIGURATION_ID	BIGINT NOT NULL	A unique identifier for the column configuration.	
COLUMN_NAME	VARCHAR(128) NOT NULL	The name of the column in the column configuration.	

Column name	Data type	Description	Usage
COLUMN_AISQL_TYPE	CHAR(1) NOT NULL	A SQL DI data type that you assign to a column in the column configuration: K Indicates that a column is assigned the key data type. C Indicates that a column is assigned the categorical data type. N Indicates that a column is assigned the numeric data type. I Indicates that a column is not assigned a data type. U Indicates that a column is assigned an unsupported data type	
COLUMN_PRIORITY	CHAR(1)	(Reserved) The processing priority that you assign to a column in the column configuration: H Indicates an high priority. M Indicates a medium priority. L Indicates a low priority.	
NEGLECT_VALUES	VARCHAR(1024)	A semicolon-separated string of values to be treated as null in the model.	

SYSAIDB.SYSAIMODELS

The SYSAIDB.SYSAIMODELS table contains a row for each AI object model and related table and state information.

Column name	Data type	Description	Usage
MODEL_ID	BIGINT NOT NULL	A unique identifier for the model.	
NAME	VARCHAR(32)	A user-defined name for the model.	
OBJECT_ID	BIGINT NOT NULL	An identifier of the object for which this configuration is created.	
CONFIGURATION_ID	BIGINT NOT NULL	A unique identifier for the configuration that is used to create this model.	
VECTOR_TABLE_CREATOR	VARCHAR(128)	The name of the user who created the vector table.	
VECTOR_TABLE_NAME	VARCHAR(128) NOT NULL	The name of the vector table.	

Column name	Data type	Description	Usage
VECTOR_TABLE_STATUS	CHAR(2) NOT NULL	The status of the vector table. I Indicates that the table is initialized for the current process. L Indicates that the table is loading. A Indicates that the table is available for use. E Indicates that the table is in error state.	
VECTOR_TABLE_DBID	SMALLINT NOT NULL	The internal identifier of the vector table database.	
VECTOR_TABLE_OBID	SMALLINT NOT NULL	The internal identifier of the vector table.	
VECTOR_TABLE_IXDBID	SMALLINT NOT NULL	The internal identifier of the vector table index database.	
VECTOR_TABLE_IXOBID	SMALLINT NOT NULL	The internal identifier of the vector table index.	
VECTOR_TABLE_VERSION	SMALLINT NOT NULL	The internal format number of the vector table.	
METRICS	CLOB(8K)	A JSON object to store metrics about the model for display in the user interface.	
INTERPRETABILITY_OCCURENCE_STRUCT	BLOB(2G)	Reserved.	
CREATED_BY	VARCHAR(32) (With SESSION_USER as default)	The SQLID of the user who created the model.	
CREATED_DATE	TIMESTAMP (With CURRENT_TIMESTAMP as default)	The timestamp when the model was created.	
LAST_UPDATED_BY	VARCHAR(32) (With SESSION_USER as default)	The SQLID of the user who last updated the model.	
LAST_UPDATED_DATE	TIMESTAMP	The timestamp when the model was last updated.	
MODEL_ROWID	ROWID NOT NULL	A rowid column to support a LOB table.	

SYSAIDB.SYSAICOLUMNCENTERS

The SYSAIDB.SYSAICOLUMNCENTERS table contains a row for each column centroid for a trained model.

Column name	Data type	Description	Usage
MODEL_ID	BIGINT NOT NULL	The unique identifier of the model to which the centroid belongs.	
COLUMN_NAME	VARCHAR(128) NOT NULL	The name of the column to which the centroid belongs.	
CLUSTER_MIN	FLOAT NOT NULL	The numeric center of a cluster.	
LABEL	VARCHAR(5) NOT NULL	The label of the vector corresponding to the cluster.	

SYSAIDB.SYSAITRAININGJOBS

The SYSAIDB.SYSAITRAININGJOBS table contains a row for each training job that you initiate and job status information.

Column name	Data type	Description	Usage
TRAINING_JOB_ID	BIGINT NOT NULL	A unique identifier for the model training job.	
OBJECT_ID	BIGINT NOT NULL	The identifier for the object for which the model is being trained.	
CONFIGURATION_ID	BIGINT NOT NULL	The identifier for the configuration that is used for the model training.	
MODEL_ID	BIGINT NOT NULL	The identifier for the model that is created as a result of training.	
STATUS	CHAR(2) NOT NULL	The status of the model training for the object: I Indicates that the training process is being initialized. L Indicating that the data is being loaded for the training job. P Indicates that the data is being processed. T Indicates that the training is started. C Indicates that the training process is completed. F Indicates that the training process failed.	
PROGRESS	SMALLINT NOT NULL	The percentage of the training process completed.	
RESOURCE	VARCHAR(512) NOT NULL	A JSON object that describes the resources allocated to the training job.	
MESSAGES	CLOB(8K)	The output of the training job.	

Column name	Data type	Description	Usage
START_TIME	TIMESTAMP NOT NULL	The start time of the training job.	
END_TIME	TIMESTAMP	The end time of the training job. A null value indicates that the training job has not yet completed.	
CREATED_BY	VARCHAR(32) (With SESSION_USER as default)	The SQLID of the user who initiated the training job.	
CREATED_DATE	TIMESTAMP (With CURRENT TIMESTAMP as default)	The timestamp when the training job started.	
LAST_UPDATED_BY	VARCHAR(32) (With SESSION_USER as default)	The SQLID of the user who last updated the training job.	
LAST_UPDATED_DATE	TIMESTAMP	The timestamp when the training job was last updated.	

Related tasks

“Configuring Db2 for SQL Data Insights” on page 9

To enable SQL Data Insights (SQL DI), you must customize and submit the DSNTIJAI job to create the required database and pseudo-catalog tables in Db2 for z/OS.

Chapter 3. Installing and configuring SQL Data Insights

The installation and configuration of SQL Data Insights (SQL DI) involve a script-driven sequence of interactive tasks. Make sure that you follow the step-by-step instruction and successfully complete each task.

Before you begin

Make sure that you have completed the following tasks before starting the SQL DI installation:

- Obtain the SMP/E image and program directory for SQL DI.
- Obtain the latest maintenance packages if available.
- Transfer the SMP/E image, program directory, and maintenance packages onto the Z system where the SQL DI will be installed.

Procedure

1. Locate the SQL DI SMP/E image and program directory on your Z system.
2. Follow the instructions in the program directory to install the SQL DI and apply all available maintenance packages.

The SMP/E program installs SQL DI in the `$SQLDI_INSTALL_DIR` directory. The default `$SQLDI_INSTALL_DIR` directory is `/usr/lpp/IBM/db2sqldi/v1r1`.
3. Verify that SMP/E was successfully executed for both the SQL DI image and the maintenance packages.
4. Create a separate `SQLDI_HOME` directory, with a minimum of 100 GB free disk space, to store the SQL DI configuration and log files.
5. Install the SQL DI web application.
 - a) In a bash session, change to the `$SQLDI_INSTALL_DIR/bin` directory.
 - b) Execute the installation script by issuing the following command:

```
./sqldi.sh create
```

- c) For each prompt, respond by entering requested information or accepting the default.
 - Enter the `SQLDI_HOME` directory where your SQL DI configuration and log files will be stored.
 - Enter the IP address or hostname for your SQL DI application.
 - Enter the port number for your SQL DI application or press **Enter** to use the default port of 15001.
 - Enter your keystore information.

SQL DI requires one of the following keystore types:

1. JCERACFKS (for managing RACF certificates and keys)
2. JCECCARACFKS (for managing RACF certificates and keys and exploiting hardware cryptography)

Select your keystore type and then enter the keyring name, the keyring owner, and the label of your SSL certificate.

- Enter the IP address or hostname of your Spark master.
- Enter the port number of your Spark master or press **Enter** to use the default port of 7077.
- Enter the port number of your Spark master REST API or press **Enter** to use the default port of 6066.

- Enter the port number of your Spark web UI or press **Enter** to use the default port of 8080.
- Enter the port number of your Spark worker or press **Enter** to use a system-assigned port.
- Enter the port number of your Spark worker web UI or press **Enter** to use the default port of 8081.
- Enter the port number of your Spark driver or press **Enter** to use a system-assigned port.
- Enter the port number of your Spark block manager or press **Enter** to use a system-assigned port.
- Enter the driver-specific port for the Spark block manager to listen on or press **Enter** to use a system-assigned port as the default.
- Enter the maximum number of retries when binding to a port or press **Enter** to use the default number of 16.

The installation process continues. You will see a message similar to the following example when it completes:

```
Congratulations! You have successfully installed SQL Data Insights.
```

- Confirm or decline to start your SQL DI automatically. If you confirm, the current command process will start SQL DI automatically. If you decline, continue to the next step to start SQL DI manually.

6. Start your SQL DI by issuing the following command:

```
./sqldi.sh start
```

Your SQL DI is successfully started if you see a message similar to the following example:

```
SQL Data Insights will use SQLDI_HOME to store configuration files and logs.
Bash version is 4.3
Starting SQL Data Insights ...
Reading configurations ...
Generating required configuration files ...
Launching SQL Data Insights ...

.....

SQL Data Insights is successfully started.

You can access it at https://<SQLDI-IPAddress>:<SQLDI-PortNumber>
```

Where *SQLDI-IPAddress* and *SQLDI-PortNumber* are either the IP address and port number that you entered or the defaults you accepted earlier. Make note of this URL and distribute it to your users.

7. Verify that the SQL DI user interface (UI) is up and running.

Sign in the UI at the following address with a valid RACF user ID that belongs to the SQLDIGRP group:

```
https://<SQLDI-IPAddress>:<SQLDI-PortNumber>
```

The SQL DI UI supports the standard or desktop version of Mozilla Firefox and Google Chrome. See Chapter 2, “Preparing SQL Data Insights installation,” on page 3 for details.

Tip: When the installation and configuration process completes successfully, the `sqldi.sh` script appends a set of command aliases to your `$HOME/.profile`. After you execute a **source** `$HOME/.profile` command, you can use the aliases to manage the lifecycle of SQL DI application and related Spark processes as follows:

- **start_sqldi** used for starting the SQL DI application.
- **stop_sqldi** used for stopping the SQL DI application.
- **start_spark** used for starting the embedded Spark cluster.
- **stop_spark** used for stopping the embedded Spark cluster.

Related tasks

[“Verifying the installation and configuration of SQL Data Insights” on page 23](#)

Before you and your business start to use SQL Data Insights (SQL DI), complete a quick procedure to verify that the feature is properly installed and configured.

Related reference

[“Preparing SQL Data Insights installation” on page 3](#)

SQL Data Insights (SQL DI) has specific system, network, user access, and security requirements. Plan ahead and prepare your environment for SQL DI installation by satisfying all the requirements.

Chapter 4. Verifying the installation and configuration of SQL Data Insights

Before you and your business start to use SQL Data Insights (SQL DI), complete a quick procedure to verify that the feature is properly installed and configured.

Before you begin

- Complete all the pre-installation tasks as described in [Chapter 2, “Preparing SQL Data Insights installation,”](#) on page 3.
- Complete the installation and configuration of SQL DI as described in [Chapter 3, “Installing and configuring SQL Data Insights,”](#) on page 19.

Procedure

Your SQL DI is properly installed and configured and ready for use if you can successfully complete the following sequence of tasks.

1. Customize and run the DSNTIJAV sample job in the Db2 SDSNSAMP data set.
The JCL job creates the sample DSNAIDB.CHURN AI object.
2. Create a connection to the Db2 system where the DSNAIDB.CHURN object is stored, as described in [Chapter 6, “Creating a connection to Db2,”](#) on page 29.
3. Enable DSNAIDB.CHURN for AI query as described in [Chapter 8, “Enabling AI query,”](#) on page 33.
 - a) For column configuration, assign SQL DI key data type to the CustomerID and retain the pre-assigned SQL DI data types for all other columns. You don't need to set column filter values.
 - b) Click **Enable AI query** to start the enablement process.

When the AI query enabling process completes successfully, the status of the DSNAIDB.CHURN object is changed to Enabled.
4. Run AI query on DSNAIDB.CHURN as described in [Chapter 10, “Running an AI query,”](#) on page 37.
 - a) Enter the following statement in the SQL editor:

```
SELECT AI_SIMILARITY(X.customerID, '3668-QPYBK') AS SimilarityScore, X.*
FROM DSNAIDB.CHURN X
WHERE X.customerID<>'3668-QPYBK'
ORDER BY SimilarityScore DESC
FETCH FIRST 10 ROWS ONLY
```

The purpose of this SQL statement is to identify top 10 customers who share similar characteristics with customer with ID 3668-QPYBK at a banking service. Customer 3668-QPYBK closed all accounts and left the service. The DSNAIDB.CHURN object and this sample query are intended to identify other customers who might also churn so that the service can act on this insight to mitigate the risk and try to retain those potential churners.

- b) Click **Run** to start the query.

As specified, the query displays 10 rows of the result set in the **Results** section.

Chapter 5. Administering SQL Data Insights

After SQL Data Insights (SQL DI) is up and running, it is important to keep it that way. You can adjust your SQL DI settings based on your workload. You can also manage the SQL DI application and the embedded Spark cluster with z/OS started tasks.

SQL Data Insights supports the standard or desktop version of Mozilla Firefox and Google Chrome. See Chapter 2, “Preparing SQL Data Insights installation,” on page 3 for details.

Related tasks

“Installing and configuring SQL Data Insights” on page 19

The installation and configuration of SQL Data Insights (SQL DI) involve a script-driven sequence of interactive tasks. Make sure that you follow the step-by-step instruction and successfully complete each task.

Related reference

“Preparing SQL Data Insights installation” on page 3

SQL Data Insights (SQL DI) has specific system, network, user access, and security requirements. Plan ahead and prepare your environment for SQL DI installation by satisfying all the requirements.

Modifying your SQL Data Insights settings


During the installation and configuration, default values are automatically set to some of your SQL Data Insights (SQL DI) parameters, including the minimum amount of memory to run Spark jobs and the maximum number of rows to load for AI queries. Depending on the size of your data and the need of your workload, you can modify the default settings on the Settings page of the SQL DI user interface.

Procedure

1. Sign in your SQL DI user interface with a valid RACF user ID (associated with the SQLDIGRP group) on the LPAR where your SQL DI is installed:

```
https://<SQLDI-IPAddress>:<SQLDI-PortNumber>
```

The SQL DI UI supports the standard or desktop version of Mozilla Firefox and Google Chrome. See Chapter 2, “Preparing SQL Data Insights installation,” on page 3 for details.

2. On the upper right corner of the SQL DI framework, click the  (gear) icon to open the Settings page.
3. On the Settings page, review and adjust the settings for the following parameters:
 - For Spark, specify the number of cores and the amount of memory for your Spark driver and executor. SQL DI uses Spark jobs to transform the data in the tables or views in your AI object. Increase the number of cores and the amount of memory to improve performance if the size of the data is large.
 - For CPU threads, specify the number of CPU threads for data preprocessing and model training. Increase the number of CPU threads to reduce the model training time.
 - For Db2 load utility, use the default LOAD utility control statement or customize it for loading trained models. SQL DI uses the ZLOAD command to upload the model training results to your Db2 system. Verify that the Db2 user ID has the permissions to run the ZLOAD command and to access the data sets referenced in the load utility control statement. See “Configuring Db2 for SQL Data Insights” on page 9 for details.

Customize the LOAD control statement based on your Db2 settings. For a very large data, increase the space allocation and specify a data class with the extended addressability attribute, allowing the data sets to grow in size beyond 4 GB.

- For AI query, specify the maximum number of rows to load for a query result set. By default, SQL DI loads up to 1000 rows of a query result set. You can change the default value if needed.

Creating a started task for the SQL Data Insights application

After you have successfully installed and configured SQL Data Insights (SQL DI), consider running the application as a z/OS started task. You can quickly create a started task by customizing the SQLDAPPS sample JCL job.

Before you begin

- Plan, install, and configure SQL DI as described in Chapter 2, “Preparing SQL Data Insights installation,” on page 3 and Chapter 3, “Installing and configuring SQL Data Insights,” on page 19.

Procedure

1. Locate the following files in the `$SQLDI_INSTALL_DIR/templates/started-task-samples` directory on the z/OS system where your SQL DI runs:
 - SQLDAPPS (sample JCL job)
 - SQLDSTRT-STDPARM.template (sample MVS data set content)
 - SQLDSTOP-STDPARM.template (sample MVS data set content)
 - stdenvs-STDENV.template (sample z/OS Unix text file content)
2. Copy SQLDAPPS into a data set in your PROCLIB concatenation, such as SYS1.PROCLIB, and customize them based on your system environment.
3. Define a new MVS data set to be used by ddname STDPARM.

- a) Create member SQLDSTRT for SQLDAPPS by copying the following lines from SQLDSTRT-STDPARM.template:

```
PGM /usr/lpp/IBM/db2sqldi/v1r1/tools/bin/bash
/usr/lpp/IBM/db2sqldi/v1r1/sql-data-insights/bin/sqldi.sh start
```

- b) Create member SQLDSTOP for SQLDAPPS by copying the following lines from SQLDSTOP-STDPARM.template:

```
PGM /usr/lpp/IBM/db2sqldi/v1r1/tools/bin/bash
/usr/lpp/IBM/db2sqldi/v1r1/sql-data-insights/bin/sqldi.sh stop
```

- c) If necessary, replace the default `/usr/lpp/IBM/db2sqldi/v1r1` segment in each directory path with `$SQLDI_INSTALL_DIR` where your SQL DI is installed.
4. Create a new z/OS Unix text file `stdenvs` to be used by ddname STDENV.

You can use the same STDENV file for all your SQL DI started tasks.

- a) Copy the following lines from `stdenvs-STDENV.template`:

```
_BPXK_AUTOCVT=ON
SQLDI_INSTALL_DIR=/usr/lpp/IBM/db2sqldi/v1r1
ZADE_INSTALL_DIR=/usr/lpp/IBM/aie/zade
BLAS_INSTALL_DIR=/usr/lpp/cbclib
SQLDI_HOME=/path/to/sqldi-home
SPARK_HOME=/usr/lpp/IBM/db2sqldi/v1r1/spark24x
SPARK_CONF_DIR=/path/to/sqldi-home/spark/conf
JAVA_HOME=/java8_64/J8.0_64
PATH=/usr/lpp/IBM/aie/zade/bin:/usr/lpp/IBM/db2sqldi/v1r1/tools/bin:/java8_64/J8.0_64/bin:/bin
IBM_JAVA_OPTIONS="-Dfile.encoding=UTF-8"
_ENCODE_FILE_NEW=ISO8859-1
_ENCODE_FILE_EXISTING=UNTAGGED
_CEE_RUNOPTS="FILETAG(AUTOCVT,AUTOTAG) POSIX(ON)"
```

- b) If necessary, replace the default `/usr/lpp/IBM/db2sqldi/v1r1` segment in each directory path with `$SQLDI_INSTALL_DIR` where your SQL DI is installed.
- c) If necessary, replace the default `/usr/lpp/IBM/aie/zade` segment in each directory path with `$ZADE_INSTALL_DIR` where the IBM Z AI Data Embedding Library is installed.

- d) If necessary, replace the default `/usr/lpp/cbclib` segment in each directory path with `$BLAS_INSTALL_DIR` where the IBM OpenBLAS is installed.
 - e) Set all environment variables based on your z/OS system environment and your SQL DI installation. See [“Configuring setup user ID for SQL Data Insights”](#) on page 5 for instructions.
5. Define a RACF profile for the new SQLDAPPS started task and assign `<sqldi_setup_userid>` as the owner by issuing the following commands:

```
RDEFINE STARTED SQLDAPPS.* STDATA(USER(<sqldi_setup_userid>) GROUP(SQLDIGRP))
SETROPTS RACLIST(STARTED) REFRESH
```

6. Run the SQLDAPPS started task to start the SQL DI application as shown in the following example:

```
/S SQLDAPPS
```

7. If necessary, run the SQLDAPPS started task to stop the UI services by issuing the following command:

```
/S SQLDAPPS,OPTION='SQLDSTOP'
```

The `OPTION` value is case-sensitive. Make sure that you issue the command in your SDSF system command extension to retain the lower case of your input.

Related tasks

[“Creating started tasks for the Spark cluster”](#) on page 27

The SQL Data Insights (SQL DI) application is powered by an embedded Spark cluster. After you have successfully installed SQL DI, consider managing the cluster by creating and running z/OS started tasks. You can quickly create the started tasks for the Spark master and worker by customizing the SQLDSPKM and SQLDSPKW sample JCL jobs.

Creating started tasks for the Spark cluster

The SQL Data Insights (SQL DI) application is powered by an embedded Spark cluster. After you have successfully installed SQL DI, consider managing the cluster by creating and running z/OS started tasks. You can quickly create the started tasks for the Spark master and worker by customizing the SQLDSPKM and SQLDSPKW sample JCL jobs.

Before you begin

- Plan, install, and configure SQL DI as described in [Chapter 2, “Preparing SQL Data Insights installation,”](#) on page 3 and [Chapter 3, “Installing and configuring SQL Data Insights,”](#) on page 19.

Procedure

1. Navigate to the `$SQLDI_INSTALL_DIR/templates/started-task-samples` directory on the z/OS system where your SQL DI runs.
2. Copy the SQLDSPKM and SQLDSPKW sample JCL files into a data set in your PROCLIB concatenation, such as `SYS1.PROCLIB`.
3. Follow the instructions in the sample procedures to customize the environment variables based on your system environment.

For example, set `$SPARK_CONF_DIR` to `SQLDI_HOME/spark/conf`.

4. Copy the `spark-zos-started-tasks.sh.template` file to the `SQLDI_HOME/spark/conf` directory by issuing the following command:

```
cp $SQLDI_INSTALL_DIR/templates/started-task-samples/spark-zos-started-tasks.sh.template
SQLDI_HOME/spark/conf/spark-zos-started-tasks.sh
```

5. Update the `spark-zos-started-tasks.sh` script in the `SQLDI_HOME/spark/conf` directory as shown in the following example:

```

# Java environment variable - REQUIRED
# Default: /usr/lpp/java/J8.0_64
export JAVA_HOME=<PATH_TO_JAVA_HOME>

# SQL DI installation directory - REQUIRED
# Default: /usr/lpp/IBM/db2sqldi/v1r1
export SQLDI_INSTALL_DIR=<PATH_TO_SQLDI_INSTALL_DIR>

# IBM Z AI Data Embedding Library installation directory - REQUIRED
# Default: /usr/lpp/IBM/aie/zade
export ZADE_INSTALL_DIR=<PATH_TO_ZADE_INSTALL_DIR>

# OpenBLAS installation directory - REQUIRED
# Default: /usr/lpp/cbclib
export BLAS_INSTALL_DIR=<PATH_TO_BLAS_INSTALL_DIR>

```

6. Define a RACF profile for the new SQLDSPKM and SQLDSPKW started tasks and assign `<sqldi_setup_userid>` as the owner by issuing the following commands:

```

RDEFINE STARTED SQLDSPKM.* STDATA(USER(<sqldi_setup_userid>) GROUP(SQLDIGRP))
RDEFINE STARTED SQLDSPKW.* STDATA(USER(<sqldi_setup_userid>) GROUP(SQLDIGRP))
SETROPTS RACLIST(STARTED) REFRESH

```

7. Start the SQLDSPKM and SQLDSPKW started tasks by issuing the following MVS commands without any parameter:

```

start SQLDSPKM
start SQLDSPKW

```

To run the Spark started tasks manually, make sure that you start SQLDSPKM before SQLDSPKW. If you automate the run, you can start them in parallel in which the processes triggered by SQLDSPKW will start right after those by SQLDSPKM.

8. If necessary, stop the SQLDSPKM and SQLDSPKW started tasks by issuing the following MVS commands without any parameter:

```

stop SQLDSPKM
stop SQLDSPKW

```

See [Stopping z/OS started tasks](#) for more information about stopping Spark started tasks.

Related tasks

[“Installing and configuring SQL Data Insights” on page 19](#)

The installation and configuration of SQL Data Insights (SQL DI) involve a script-driven sequence of interactive tasks. Make sure that you follow the step-by-step instruction and successfully complete each task.

Related reference

[“Preparing SQL Data Insights installation” on page 3](#)

SQL Data Insights (SQL DI) has specific system, network, user access, and security requirements. Plan ahead and prepare your environment for SQL DI installation by satisfying all the requirements.

Chapter 6. Creating a connection to Db2

To access data and enable AI query, your SQL Data Insights (SQL DI) must be connected to your Db2 system or data sharing group. You can create and activate a required JDBC connection on the Connections page of the SQL DI user interface.

Procedure

1. Sign in your SQL DI user interface with a valid RACF user ID (associated with the SQLDIGRP group) on the LPAR where your SQL DI is installed:

```
https://<SQLDI-IPAddress>:<SQLDI-PortNumber>
```

The SQL DI UI supports the standard or desktop version of Mozilla Firefox and Google Chrome. See [Chapter 2, “Preparing SQL Data Insights installation,”](#) on page 3 for details.

2. On the Connections page, click **Add connection**.
3. On the Add connection page, specify a name and details for the connection, including the hostname or IP address, port number, location, JDBC properties (optional), and special registers (optional) of your Db2 system or data sharing group.
4. Optionally, check **Port enabled for SSL connections** and enter the SSL certificate content in the input field.

If your Db2 system or data sharing group uses SSL for network communications, you must configure the new connection with the required SSL certificate information. You can provide your SSL certificate information in one of the following ways:

- Option 1: Check the box for **Port enabled for SSL connections** and provide the required SSL certificate information. Your SSL certificate must use Base64 ASCII encoding in PEM (.pem) format. Make sure that the certificate content is bound by the -----BEGIN CERTIFICATE----- header and the -----END CERTIFICATE----- footer.
- Option 2: Specify the following value in the JDBC properties field in the previous step:


```
sslConnection=true;sslCertLocation=<path_to_trusted_certificate>
```


Where *<path_to_trusted_certificate>* is the full path and the file name of your SSL certificate on the system where your SQL DI runs. The certificate must use Base64 ASCII or binary encoding in ARM (.arm), PEM (.pem), CERT (.cert), CRT (.crt), or DER (.der) format. If the certificate file uses Base64 ASCII encoding, make sure that the certificate content is bound by the plain-text -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- lines.

5. Enter your Db2 username and password.

Make sure that your username or ID has sufficient privileges to access the Db2 system or the Db2 data sharing group.

6. Click **Add** to create the new connection.
7. Back on the Connections page, verify that the new connection shows up.
8. Optionally, activate the new connection.

A new connection is activated (green-checked) when it is created. If necessary, you can deactivate the connection by selecting **Disconnect** from the  action menu of the connection. To re-activate the connection:

- a) Click the action menu  and select **Connect**.
- b) Enter your Db2 username and password.
- c) Click **Connect** to activate the connection.

The connection is successfully activated when it's green-checked.

If necessary, you can deactivate, edit, or remove the connection.

Chapter 7. Adding an AI object


SQL Data Insights (SQL DI) enables AI functions on a Db2 object, such as a user table or view, by creating a corresponding AI object and enabling it for AI query. The AI objects are associated with a specific connection. You can add and manage an AI object for a connection on the AI objects page of the SQL DI user interface.

Procedure

1. Sign in your SQL DI user interface with a valid RACF user ID (associated with the SQLDIGRP group) on the LPAR where your SQL DI is installed:

`https://<SQLDI-IPAddress>:<SQLDI-PortNumber>`

The SQL DI UI supports the standard or desktop version of Mozilla Firefox and Google Chrome. See [Chapter 2, “Preparing SQL Data Insights installation,” on page 3](#) for details.

2. On the Connections page, select a connection and click the  action menu.
3. Select **List AI objects** to open the AI objects page for the connection.
4. On the AI objects page, click **Add object**.
5. On the Add object page, choose one or more schemas to display all associated Db2 objects.
6. Select one or more Db2 objects.
7. Click **Add object** to add selected objects.

If you select only one Db2 object to add and want to enable it for AI query, click **Enable AI query** to accomplish both in a single step. See [Chapter 8, “Enabling AI query,” on page 33](#) for instructions on column configurations.

8. Back on the AI objects page, verify that the newly added objects show up and their statuses are Created.

Chapter 8. Enabling AI query



You can enable an AI object for AI query when or after the object is created. Enabling AI query requires column configuration and model training. You can enable an AI object for AI query on the AI objects page of the SQL Data Insights (SQL DI) user interface.

Procedure

1. Sign in your SQL DI user interface with a valid RACF user ID (associated with the SQLDIGRP group) on the LPAR where your SQL DI is installed:

```
https://<SQLDI-IPAddress>:<SQLDI-PortNumber>
```

The SQL DI UI supports the standard or desktop version of Mozilla Firefox and Google Chrome. See [Chapter 2, “Preparing SQL Data Insights installation,” on page 3](#) for details.

2. On the Connections page, select a connection and click the  action menu.
3. Select List AI objects to open the AI objects page for the connection.
4. Select an AI object and from the  action menu, select Enable AI query.
5. On the Enable AI query page, select, configure, and filter the columns that you want to include for your AI queries.
 - a) For Column configuration, select one or more columns and assign each one a SQL DI data type. SQL DI uses your column configuration to create and train a machine learning model for the object.

SQL DI supports the following data types:

- **Categorical:** The SQL DI categorical data type is used for columns with discrete values, each of which is its own entity. Type `categorical` is common in columns of many SQL data types. Columns with character or datetime SQL data types, such as CHAR, VARCHAR, DATE, TIME, TIMESTAMP, and TIMESTAMP WITH TIMEZONE, are a SQL DI categorical type, and so are columns with numeric values representing social security or ID numbers.
- **Numeric:** The SQL DI numeric data type is used for columns with continuous values. Columns with numeric SQL data types, such as SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, FLOAT, and DECFLOAT, are a SQL DI numeric type. SQL DI uses clustering to group numeric values that are close together during the AI query enablement process.
- **Key:** The SQL DI key data type is used to indicate that a column represents an entire row. A customer ID column is a SQL DI key type. When processing an AI query that includes a column with the key type, SQL DI evaluates the affected rows in their entirety and effectively compares all the values in one row to those in another, not just the values in the column of the key type.

Tip:

- While you can specify SQL DI type `categorical` or `numeric` to as many columns as you want, you can assign type `key` to only one column.
- You can treat a column with numeric SQL data types as a SQL DI `categorical` type if the column contains 10 or fewer distinct values. A grade column in a class schedule table and an interest rate in a bank loan table are good examples. If the grade column has only 6-10 unique values, set the column as a SQL DI `categorical` column.

Optionally, you can **import** a column configuration from a JSON file. The `.json` file can contain the column configuration of an AI object that is already enabled for AI query. Make sure that the schema in the imported column configuration file is consistent with that defined in the AI object you currently select.

- b) Click **Next** to continue.
- c) Optionally, for Column filter, enter values to filter and exclude records from selected columns.

In the `Global filter values` field, enter a value that applies to one or more columns, which is considered a global filter value. Any record with the matching value will be excluded from the column configuration. You can add multiple global filter values, but you must enter them one by one.

In the `Column-specific filter values` field for each row, enter one or more values, separated by semicolon, that you want to exclude only from that column.


Suppose that many columns of your AI object table, including column `PAYMENTMETHOD`, contain the values of `N/A`, `n/a`, `na`, and `NR`. If you want to exclude all records with any of the matching values from your table, simply enter `n/a` in the `Global filter values` field and click **Add**. Then, repeat the process to add `na` and `nr` as the second and third global filter values.


Let's say that column `PAYMENTMETHOD` contains the values of `invalid` and `EMPTY`. If you want to exclude any record with one of matching values from the column, simply enter `invalid;empty` as the column-specific filter values.

When you run an AI query on this AI object table, the column configuration with both the global and column-specific filter values are applied. As a result, all records that match the value of `N/A`, `n/a`, `na`, or `NR` will be excluded from all columns. In addition, all records that match the value of `invalid` or `EMPTY` will be excluded from the `PAYMENTMETHOD` column.

6. Click **Enable** to start the model training in the background.

If you create your AI objects one at a time, you have the option to enable the object for AI query during the creation process. On the `Add object` page, click **Enable AI query** to add the object and enable it for AI query in a single step. See [Chapter 7, "Adding an AI object," on page 31](#) for more information.

SQL DI starts the enabling process in the background. The entire process may take some time to complete depending on the size of your table or view and the number of selected columns. You can monitor the progress by refreshing the page and then clicking the  details arrow to the left of the object name.

The AI query enabling process completes successfully when the object's status is changed to `Enabled` with a green check mark. The `Enabled` status indicates that the model for the object is successfully created and trained. If needed, export the column configuration of this object for future use by clicking `Export column configuration` from the  action menu of the object. The column configuration is saved into a `.json` file.

If the status is `Failed` with a red triangle, repeat steps ["4" on page 33](#) - ["6" on page 34](#) to restart the enabling process. Make sure that you review your column configuration and eliminate any error.

If needed, you can disable the object for AI query. Afterward, the status of the object is changed to `Disabled`.

If an object is never initiated for AI query enablement, its status remains `Created`.

Chapter 9. Viewing an AI object model

When an AI object is enabled for AI query, SQL Data Insights (SQL DI) creates and trains a model for the object. You can view the model on the `Model details` page of the SQL DI user interface.

Procedure

1. Sign in your SQL DI user interface with a valid RACF user ID (associated with the SQLDIGRP group) on the LPAR where your SQL DI is installed:

`https://<SQLDI-IPAddress>:<SQLDI-PortNumber>`

The SQL DI UI supports the standard or desktop version of Mozilla Firefox and Google Chrome. See [Chapter 2, “Preparing SQL Data Insights installation,” on page 3](#) for details.

2. On the `Connections` page, select a connection and click the `⋮` action menu.
3. Select `List AI objects` to open the `AI objects` page for the connection.
4. On the `AI objects` page, select an object and click the `⋮` action menu.
5. Select `View model` to open the `Model details` page.

You can toggle between `Training history` and `Cluster center` tabs to view the details of an object model.

Chapter 10. Running an AI query


After an AI object is enabled for AI query, you can run queries on the object on the AI objects page of the SQL Data Insights (SQL DI) user interface.

Procedure

1. Sign in your SQL DI user interface with a valid RACF user ID (associated with the SQLDIGRP group) on the LPAR where your SQL DI is installed:

`https://<SQLDI-IPAddress>:<SQLDI-PortNumber>`

The SQL DI UI supports the standard or desktop version of Mozilla Firefox and Google Chrome. See [Chapter 2, “Preparing SQL Data Insights installation,” on page 3](#) for details.

2. On the Connections page, select a connection and click the  action menu.
3. Select List AI objects to open the AI objects page for the connection.
4. Click **Run query** to open the Run query page with the query editor.
5. On the Run query page, select the type of query that you want to run.

You can select one of the following types of query based on your business use case:

- *Semantic similarity*: A similarity query identifies groups of similar records or entities in records. Consider selecting semantic similarity if your query intends to identify the similarities of customer characteristics and behaviors in industries, such as commerce, finance, and insurance.
- *Semantic dissimilarity*: A dissimilarity query finds the outliers from the norm in records. Consider selecting semantic dissimilarity if your query intends to detect operational anomalies, fraudulent activities, and other patterns of deviation.
- *Semantic clustering*: A clustering query forms a cluster of entities in records and evaluates whether or not an additional entity belongs in the cluster. Consider selecting semantic clustering if your query intends to examine similarities or dissimilarities across multiple entities in a broader context.
- *Semantic analogy*: An analogy query determines if the relationship between two entities applies to that of a second pair of entities. Consider selecting semantic analogy if your query intends to discover your customers' preference for a specific product and the degree of their affinity for other products.

Depending on the query type that you select, SQL DI processes your query by using the AI_SIMILARITY, AI_SEMANTIC_CLUSTER, or AI_ANALOGY scalar function. See [Db2 built-in scalar functions](#) for details about the built-in AI scalar functions.

6. For the query editor, select a tab and enter your SQL statement or use the default statement.

SQL DI retains and caches the SQL statement on each tab of the editor. If needed, click **Add SQL +** to add a new tab or click **X** to remove a tab from the editor.

You can open up to 10 tabs of the query editor. When the limit is reached, you must close some existing tabs in order to open new ones. When you close a tab, the SQL statement on the tab and in the cache is deleted.

7. Click **Run** to run the query or **Clear** to edit your SQL statement.
 - You can run multiple queries specified on multiple tabs at the same time.
 - By default, SQL DI displays 50 rows for a query result set on this page. If you want to see fewer than 50 rows, you can specify the `fetch * rows` option in your SQL statement. You have the option to export the displayed rows into a CSV file.
 - SQL DI loads the remaining rows of a query result set in the backend. You have the option to download the remaining rows or the entire set of your query result. The default value for the maximum number of loaded rows is 1000. If you want more rows loaded, change the default value on the Settings page as described in [“Modifying your SQL Data Insights settings” on page 25](#).

- An AI query returns an SQL null value if it includes functions with arguments of null, filtered, or unseen values. Filtered values result from the application of the global or column-specific value filters you set in your column configuration for AI query. In the AI model, they are represented with the DB2_GENERATED_EMPTY string. Unseen values are those that are not present in the AI object when it's enabled for AI query. If your query includes arguments with null, filtered, or unseen values, SQL DI does not compute any result and thus returns an SQL null value.
- It is a best practice to specify constant or unchanging values for up to three clustering-arguments for the AI_SEMANTIC_CLUSTER option. If the arguments to AI_SEMANTIC_CLUSTER are different for every row, SQL DI scores each of the unique values against a potentially different cluster, making the query results difficult to understand.

Related information

[Db2 built-in scalar functions](#)

Chapter 11. Analyzing data



When training the model for an AI object, SQL Data Insights (SQL DI) collects key data statistics and renders them into metric scores for the model. The visualized scores can help you understand the results of AI queries on the object. You can view the data statistics and model scores on the Analyze data page of the SQL DI user interface.

Procedure

1. Sign in your SQL DI user interface with a valid RACF user ID (associated with the SQLDIGRP group) on the LPAR where your SQL DI is installed:


`https://<SQLDI-IPAddress>:<SQLDI-PortNumber>`

The SQL DI UI supports the standard or desktop version of Mozilla Firefox and Google Chrome. See [Chapter 2, “Preparing SQL Data Insights installation,”](#) on page 3 for details.

2. On the Connections page, select a connection and click the  action menu.
3. Select List AI objects to open the AI objects page for the connection.
4. Select an AI object and from the  action menu, select Analyze data.
5. On the Analyze data page, toggle between the Object details, Data statistics, and Column influence tabs.
 - The Object details tab displays the column configuration information, including the name, Db2 data type, and SQL DI data type of a column.
 - The Data statistics tab displays the column value distribution information, including the most common value, the number of most common value, the number of unique value, standard deviation, mean, max, and min values of a column.
 - The Column influence tab displays the scores from the "influence" and "discriminator" metrics. The influence and discriminatory scores are specific to the column names used in the associated relational table and capture the properties of the trained model. The metrics use model-specific statistics to generate query-agnostic interpretation of your AI query results. You can use the metric scores to understand the contributions of individual columns to the training of the model, identify and correct skews in the data, and determine the key influencers on the query results.

The influence metrics indicates the influence of a particular column on the training of a model. The influence score for every column is computed as the ratio of NULL and user-specified empty values, such as NA or Not Recorded, to the total number of values (that matches the total number of rows in a table). Empty values are not used in model training. The fewer the empty values a column has, the higher its influence score becomes, and vice versa.

The discriminatory metrics captures the value distribution of each column in the associated table. The discriminatory score measures the ability of a column (the values in a column) to distinguish its co-occurring entries in rows. If a column has many repeated values, the column can be considered skewed towards those values. This translates into a relatively low discriminatory score, indicating possible skews in the data. On the contrary, if a column has many unique values, the discriminatory score for the column tends to be high. So, the more the unique values a column has, the higher its discriminatory score becomes. The unique primary key column contains unique values only, and its discriminatory score is the highest.

6. Click the  icon to reload the page to display data updates.

Information resources for Db2 for z/OS and related products

You can find the online product documentation for Db2 for z/OS in IBM Documentation.

IBM Documentation is the home of all online product documentation for Db2 for z/OS and related products, including PDF format manuals.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785 US*

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing Legal and Intellectual Property Law IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Director of Licensing IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785 US*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as shown below:

© (your company name) (year).

Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. (enter the year or years).

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)[®] are trademarks or registered marks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at: <http://www.ibm.com/legal/copytrade.shtml>.

Linux[®] is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions:

Applicability: These terms and conditions are in addition to any terms of use for the IBM website.

Personal use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights: Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at <http://www.ibm.com/privacy> and IBM’s Online Privacy Statement at <http://www.ibm.com/privacy/details> the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at <http://www.ibm.com/software/info/product-privacy>.

Glossary

The glossary is available in IBM Documentation

For definitions of Db2 for z/OS terms, see .

Index

A

- accessibility
 - keyboard [vi](#)
 - shortcut keys [vi](#)
- adding
 - AI object [31](#)
 - object [31](#)
- administering SQL Data Insights [25](#)
- AI object
 - adding [31](#)
 - AI query [33](#), [39](#)
- AI query
 - enabling [33](#), [39](#)
 - running [37](#)

C

- configuration
 - verifying [23](#)
- configuring
 - Db2
 - configuring [9](#)
 - keyring [8](#)
 - keystore [8](#)
 - setup user ID [5](#)
 - SQL Data Insights
 - configuring [9](#)
 - SSL certificate [8](#)
 - user authentication [8](#)
 - z/OS system environment
 - [5](#)
- connection
 - creating [29](#)
- creating
 - connection [29](#)
 - started task
 - Spark [27](#)
 - SQL DI [26](#)

D

- disability [vi](#)

E

- enabling
 - AI object [33](#), [39](#)
 - AI query [33](#), [39](#)

H

- hardware requirements
 - planning [3](#)

I

- index
 - SQL DI table [10](#)
- installation
 - verifying [23](#)
- installing
 - SQL Data Insights [19](#)

K

- keyring
 - configuring [8](#)
- keyring-based keystore
 - configuring [8](#)
- keystore
 - configuring [8](#)

L

- links
 - non-IBM Web sites
 - [44](#)

M

- model
 - viewing [35](#)
- modifying
 - settings [25](#)
 - SQL DI [25](#)

N

- network requirements
 - planning [3](#)

O

- object
 - adding [31](#)
- overview
 - SQL Data Insights [1](#)

P

- planning
 - hardware [3](#)
 - installation [3](#)
 - network [3](#)
 - software [3](#)
 - system capacity [3](#)
- port
 - planning [3](#)

R

RACF

- keyring [8](#)
- keystore [8](#)

running

- AI query [37](#)
- started task
 - Spark [27](#)
 - SQL DI [26](#)

S

settings

- modifying [25](#)

setup user ID

- configuring [5](#)

shortcut keys

- keyboard [vi](#)

software requirements

- planning [3](#)

Spark

- started task [27](#)

SQL Data Insights

- administering [25](#)
- configuring [19](#)
- installing [19](#)
- overview [1](#)

SQL DI

- started task [26](#)

SQL DI table

- index [10](#)
- table space [10](#)

SSL certificate

- configuring [8](#)

started task

- creating [26, 27](#)
- running [26, 27](#)
- Spark [27](#)
- SQL DI [26](#)

SYSAIDB.SYSAICOLUMNCENTERSSQL DI table [10](#)

SYSAIDB.SYSAICOLUMNCONFIGSQL DI table [10](#)

SYSAIDB.SYSAICONFIGURATIONSSQL DI table [10](#)

SYSAIDB.SYSAIMODELSSQL DI table [10](#)

SYSAIDB.SYSAIOBJECTSSQL DI table [10](#)

SYSAIDB.SYSAITRAININGJOBSSQL DI table [10](#)

system capacity

- planning [3](#)

T

table spaceSQL DI table [10](#)

U

user authentication

- configuring [8](#)

V

verifying

- configuration [23](#)
- installation [23](#)

viewing

AI query [35](#)

model [35](#)



Product Number: 5698-DB2
5698-DBV