SDD-SQL DATABASE DEVELOPER

VERSION 1.5

SDD – SQL Database Developer Guide

Contents

1. Introduction

The use of a graphical interface is the best way to access database information for an ad-hoc query. All relational databases have a way to offer this type of access using fancy interfaces. The SDD-SQL Database Developer is a web interface to provide a simple way for DMSII users to submit SQL ANSI queries to retrieve information.

The basic features present in the SDD are:

- Web interface to submit SQL statements to DMSII databases with the result displayed in a grid
- Use an Oracle database to store parameters and activity
- Results can be stored in CSV files
- Log of all queries submitted by the users
- Function to store queries to be used anytime
- Profiles to restrict access to databases. Users can only access databases assigned to them.
- Inquiry and Inquiry/Update access
- Use Unisys-MCPSQL or Unisys-JDBC
- Resource control: timeout for online queries, the maximum number of records returned, only one online query in execution per user, queries in background mode to be executed asynchronously
- Background query module to control the asynchronous queries
- Menu for support users to check the executions
- Control the external access to DMSII databases from only one location: the application server

There are two ways to connect to a DMSII database to submit SQL statements: JDBC and MCPSQL interfaces. The SDD application can use both access types. It is also possible to create a SQL statement using joins between two different DMSII databases or between a DMSII dataset and Oracle or SQL Server tables.

The application stores all configuration definitions, user activities, and requests in an Oracle database. The application was designed to allow one database for different environments, like DEVELOPMENT, ACCEPTANCE, and PRODUCTION.

The runtime environment needs a web server like Tomcat. This web server can be installed on an application server running Linux or Windows. Only the application server must connect to the mainframe using a specific TCPIP port. Because of this, a firewall rule can be defined allowing connections to the mainframe from the application server only. That way, no other server or personal computer could access the mainframe to submit JDBC or MCPSQL SQL statements.

The application has a login page to authorize the user's access. Four types of users are available: administrator, support, query, and basic user. Each type of user has access to specific functions, grouped in menus. More than one type can be selected for a user and the menus will be visible depending on the user's user type. A superuser is a user that has all user types.

The Admin module is available to define infrastructure components like environments, hostnames, databases, and users.

The Support module is responsible for monitoring the use of the application by other users. Users with this type enabled can verify the currently connected users, the queries in progress, databases connections opened, kill database connections, the log messages, and background query status.

The User module contains functions related to SQL statements such as sending queries to be processed online and in the background, saving statements for later use, checking the background queries status, and downloading the results.

The query type must be assigned to a user to allow sending SQL statements. The databases the user can access must be configured. This configuration is based on roles that contain a list of databases, the access mode (inquiry or update), and resource limits. When a user is assigned to a role, all databases present on this role can be used by the user. A user can be assigned to more than one role and the allowed databases are the list of databases included in all of the roles associated with the user.

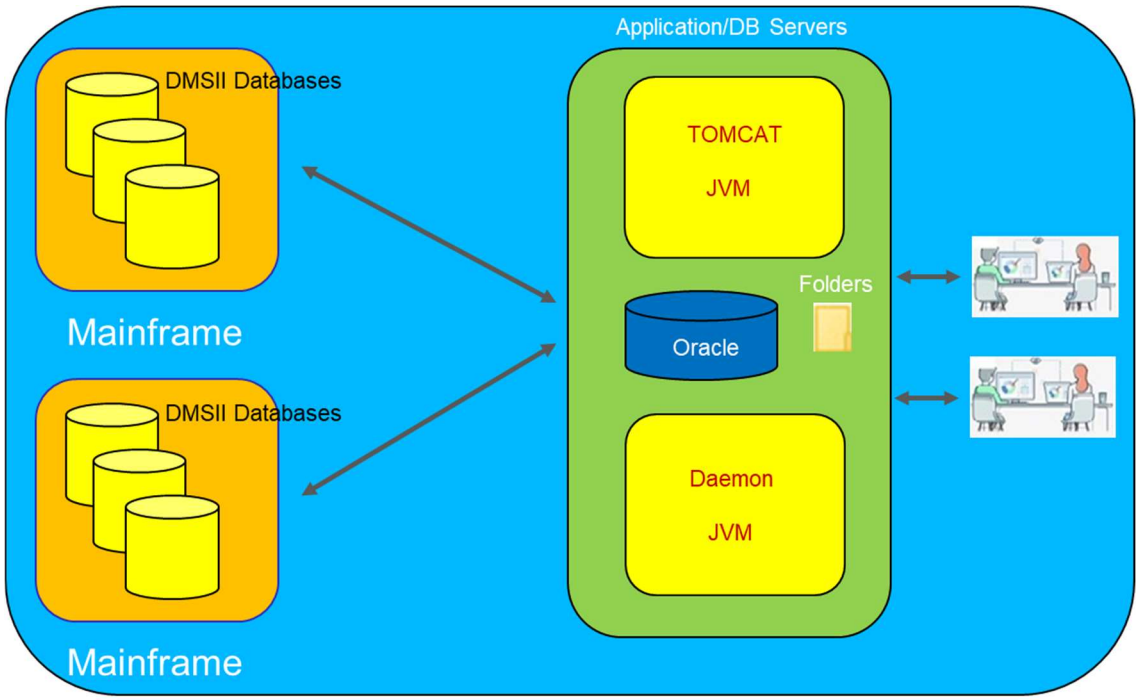
A background query function is available to avoid having to wait connected for the result of a long query. The user can select to run the query in background mode, check the progress of the execution on a query page, and, when finished, get the result in a CSV file. It is possible to define a timeout for a background query, to control the resources used.

A connection pool controls the DMSII connections. The maximum number of opened DMSII connections is defined in a parameter page per mainframe server. This defines the concurrent SQL statements that can be processed in parallel at a time on each server. An error message is sent if all connections are in use. In this case, the user must wait and try to submit the SQL statement later.

To control the mainframe resources used, the user can only submit one query at a time. When a second query is submitted while the previous one is already running, a message is sent and the execution is not allowed.

The basic diagram of the solution is shown below. SDD works with the concept of environments. One environment, such as DEVELOPMENT or PRODUCTION, consists of one or more mainframes. The databases defined in that environment are present in the mainframes defined in it.

The basic architecture to configure SDD is shown below.



The configuration can use one application server and one database server or having both services in only one server. The Tomcat web server is responsible for the interface with the users and the daemon is responsible for running background queries and some administrative tasks.

The image below shows the SDD login page. The user must inform a valid username and a password. The pre-defined username is ***ADMIN*** and the password is ***admin***. This is the initial user that must be used to configure the entire environment.

After a successful login, the application shows a home page when the functions can be selected using a menu bar. The upper right corner of any page has the name of the current user and which databases are selected at the moment if any.

The image below shows the SDD home menu.

The image below shows the connected user “ADMIN” and the selected database BDCNS on the PROD server.

All the functions available are grouped in 4 modules and will be described in the following sections. The web application has the 3 modules. The fourth module is the Daemon module which is a batch java class, responsible for the control and the execution of background queries and some administrative tasks.

2. Administrator Module

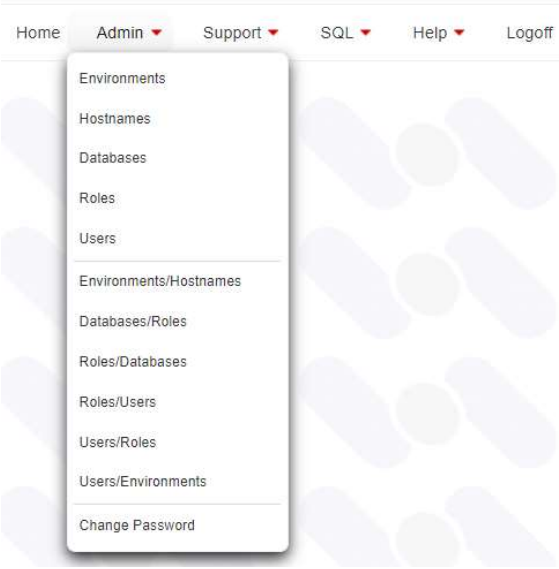
This module has the objective to define global execution parameters, manage environments, hostnames, databases, roles, resource limits, and set roles for users. SQL tables store the information that will be used when a user connects to the SQL module.

The role definition, that limits the users activity, can contain:

- List of databases allowed
- Number of returned records allowed per query
- Inquiry/Update
- Connection time limit

The link between roles and users defines what a user can do. According to the rules, the users' queries are limited only to use the databases that were authorized. Before sending SQL statements to the mainframe using a web interface, all rules are used to validate the action.

The available options can be viewed and updated using the corresponding menu item ("Admin").



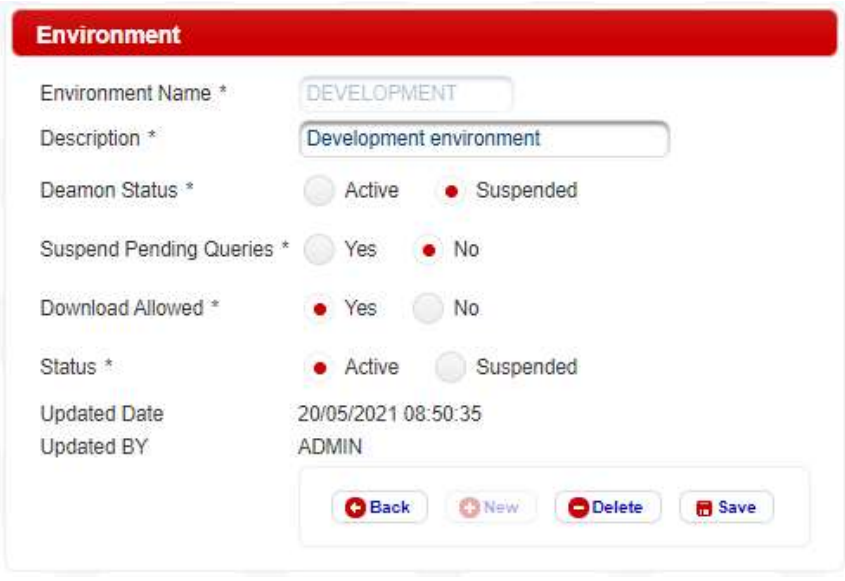
In general, environments, hostnames, databases, roles, and users have a query page to list all existing definitions in a grid. There are two buttons to go to the detail page: one to select a specific definition to be changed and another button to create a new definition ("Add New" button).

The functions available in this module are:

2.1 Environments

This option is used to maintain the environment definitions. The selection of the menu option "Environments" shows a grid on a page with all current environments defined.

Pressing the button  calls the definition page where it is possible to create a new environment. To see all fields, change the environment attributes or delete the environment, the button  must be pressed. This action shows the definition page.



The following image has the attributes available to define or update an environment.

Environment Name: the name of the environment. The name defined here should be used to set the SWENV environment variable used by the web application(Tomcat) and by the daemon program.

Description: the description of the environment

Background timeout: This attribute define the maximum time in minutes that one background query should last. After reaching this elapsed time, the background execution is aborted.

Download Allowed: This attribute defines if the grid resulting from the query execution can be downloaded as an XLS or CSV file. If the selection Is “No”, then the record can only be visualized on the grid, but the download buttons will be disabled.

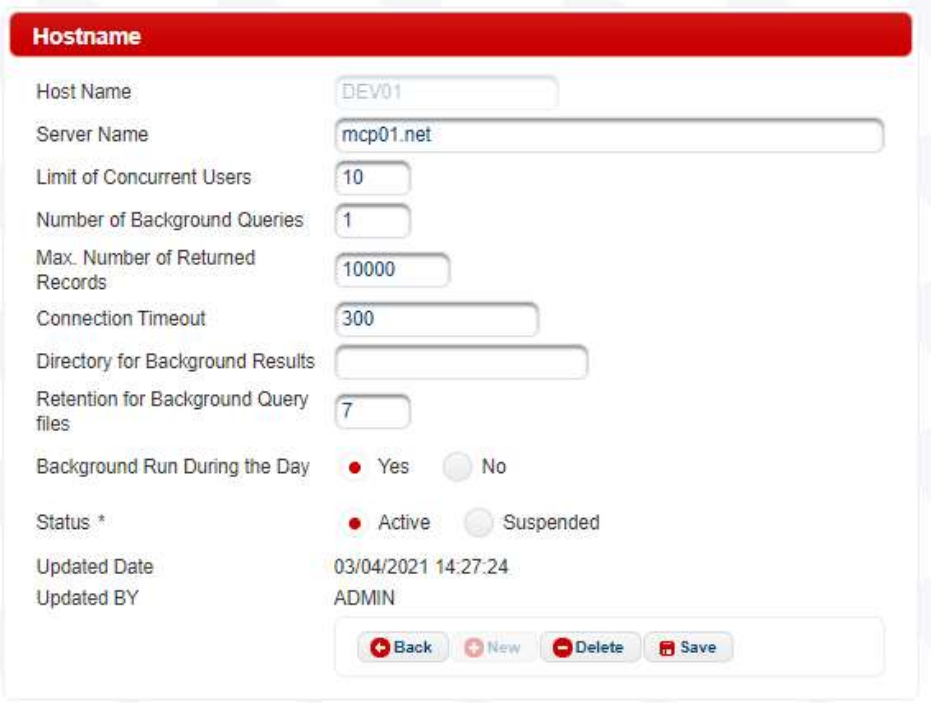
Status: Status of the environment. The value Active means that all the databases of hostnames that belong to this environment are ready to receive SQL statements to process. The value of Suspended means that no queries are allowed using databases of hostnames that belong to this environment.

2.2 Hostnames

This option is used to maintain the hostnames definitions. The selection of the menu option “Hostnames” shows a grid on a page with all current environments defined.

Pressing the button 

calls the definition page where it is possible to create a new hostname. To see all fields, change the hostname attributes or delete the hostname, the button  must be pressed. This action shows the definition page.



The following image has the attributes available to define or update an environment.

Host Name: the name of the host. The host is used to be assigned to an environment. All databases running on this hostname can be used by users granted to the environment.

Server Name: define the server name known by the network. If the hostname is defined with the same name as the server name, this field can be left blank. It's possible to inform the IP address instead of the server name in this field.

Number of Background Queries: define the maximum number of background queries that can be executed in parallel. The higher this number, the greatest the CPU utilization.

Max. Number of Returned Records: define the maximum number an online query can return. This number is the maximum number of records present in the grid of the Query page. This number will be used with the same attribute present on the Role and User record. The value for a user will be the smaller of them.

Connection timeout: define the maximum time in minutes that can be used by an online query. After that number of minutes, the online query is automatically aborted and a message will be sent to the user.

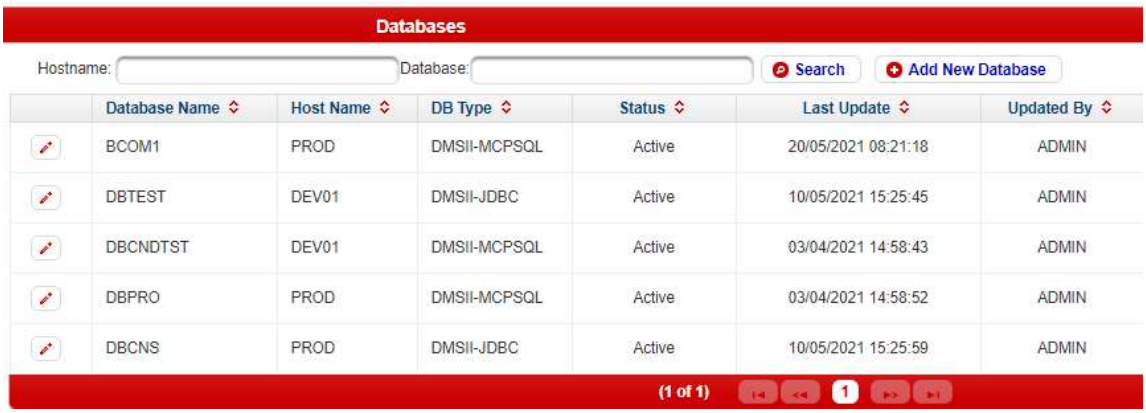
Directory for Background Queries: define the folder the CSV files created by a background query will be saved.

Retention for Background Files: define how many days files created by background queries can be kept. The daemon process is responsible for processing this feature.

Status: Status of the hostname. The value Active means that all the databases of these hostnames are ready to process SQL statements. The value of Suspended means that no queries are allowed using databases of this hostname.

2.3 Databases

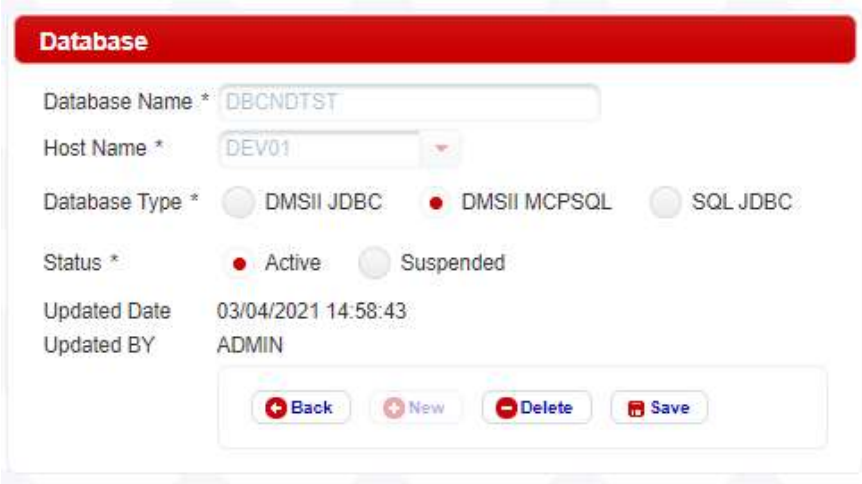
This option is used to maintain the database definitions. The selection of the menu option "Databases" shows a grid on a page with all current databases defined.



The page has two filter fields to create a more specific grid result. You can use the character % to filter part of the database or hostname field.

Pressing the button 

calls the definition page where it is possible to create a new database. To see all fields, change the database attributes or delete the database, the button  must be pressed. This action shows the definition page.



Database: the name of the database. If using the MCPSQL interface, this name must be the name defined in the CONFIG file. If using the JDBC interface, this must be a valid DMSII database in the mainframe defined in the field **Hostname**. The database is used to be assigned to a role. When a role is assigned to a user, all databases can be accessed by that user.

Hostname: define the hostname where this database is present.

Database Type: define the database type available: two options for DMSII databases (DMSII JDBC or DMSII MCPSQL) and one option for Oracle and SQL Server databases (SQL JDBC).

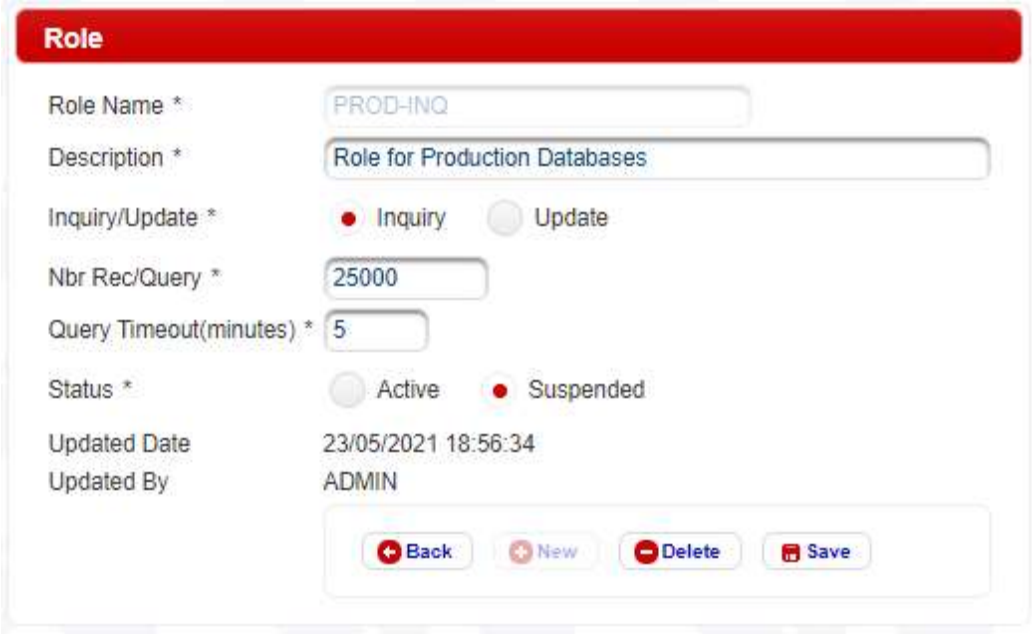
Status: Status of the database. The value Active means that this database is ready to execute SQL statements. The value of Suspended means that no queries are allowed using this database.

2.4 Roles

This option is used to maintain the role definitions. The selection of the menu option “Roles” shows a grid on a page with all current roles defined.

Pressing the button 

calls the definition page where it is possible to create a new role. To see all fields, change the role attributes or delete the role, the button  must be pressed. This action shows the definition page.



Role Name: the name of the role.

Description: This field define a description for the role

Inquiry/Update: This option defines the access type of the databases included. There are two options:

> ***Inquiry****: the access to all databases belonging to a role will be in inquiry mode. No updates will be allowed.*
>
> ***Update****: the access to all databases belonging to a role will be in inquiry and update mode.*

Nbr.Rec/Query: this field specifies the maximum number of records that can be returned to the browser when an online query is run. In case of the number of records returned is exceeded, the message **"Maximum number of records exceeded"** will be shown and the resulting grid will be

displayed with the limited number of records. This parameter is the same for all databases assigned to this role.

Seconds per Query: This field defines the maximum number of seconds a query can use. When the elapsed time for an online query exceeds the number of seconds defined in this field, the query is automatically aborted. This parameter is the same for all databases belonging to a role.

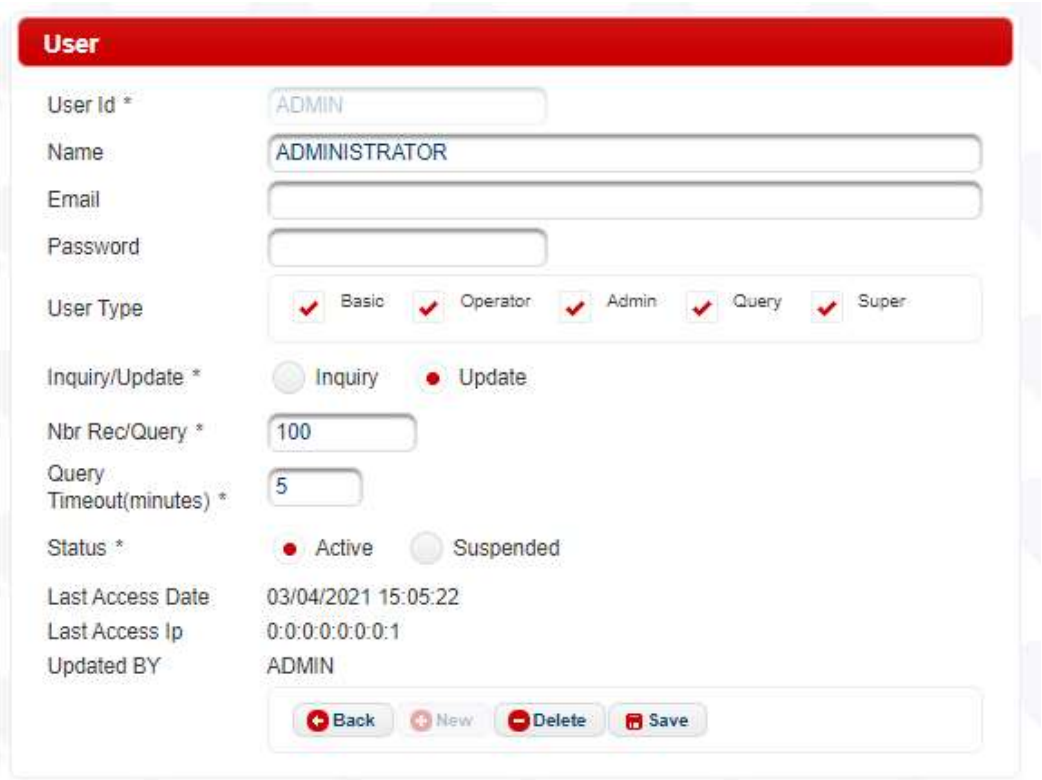
Status: This field defines the current role status. If “Suspended” is set, the role will no longer be used to define the user access. If the value is changed to “Active” again, all databases assigned to the role can be accessed by users.

2.5 Users

This option is used to maintain the user definitions. To access the applications by a user, the corresponding user record must be created. The user type defines the menu options that will be available to the user. The roles where the user was assigned contain the databases that can be accessed.

The selection of the menu option “Users” shows a grid on a page with all current users defined.

Pressing the button  calls the definition page where it is possible to create a new user. To see all fields, change the user attributes or delete the user, the button  must be pressed. This action shows the definition page.



The field “User Id” is the key for the user and should be the corporate key. This will be used to validate the user when logged in in Windows using a production or development corporate key.

The fields available to configure a user are:

User Id: the user identification

Name: The user’s name

Email: The user’s email

User type: This option define which functions the user can use. It is possible to define more than one type for a user. There are four options:

- ***Basic**: the user can access the functions to create and submit queries to the DMSII databases that he has to access.*
- ***Admin:** the user can access the "Admin" and "Support" menu.*
- ***Support:** the user can access the "Support" menu to access information about active sessions, active connections and the current SQL statements in progress, and the status of background queries.*
- ***Query:** the user can access the "Support" menu to access information about active sessions, active connections and the current SQL statements in progress, and the status of background queries.*
- ***Super:** this user has access to any function in the application.*

Inquiry/Update: specify if the user can access the allowed databases in inquiry mode only or inquiry and update mode.

Nbr Rec/Query: this field specifies the maximum number of records that can be returned to the browser when a query is run. When the number of returned records exceeds this maximum number allowed, the message **"Maximum number of records exceeded"** will be displayed and the resulting grid will be displayed with the limited number of records.

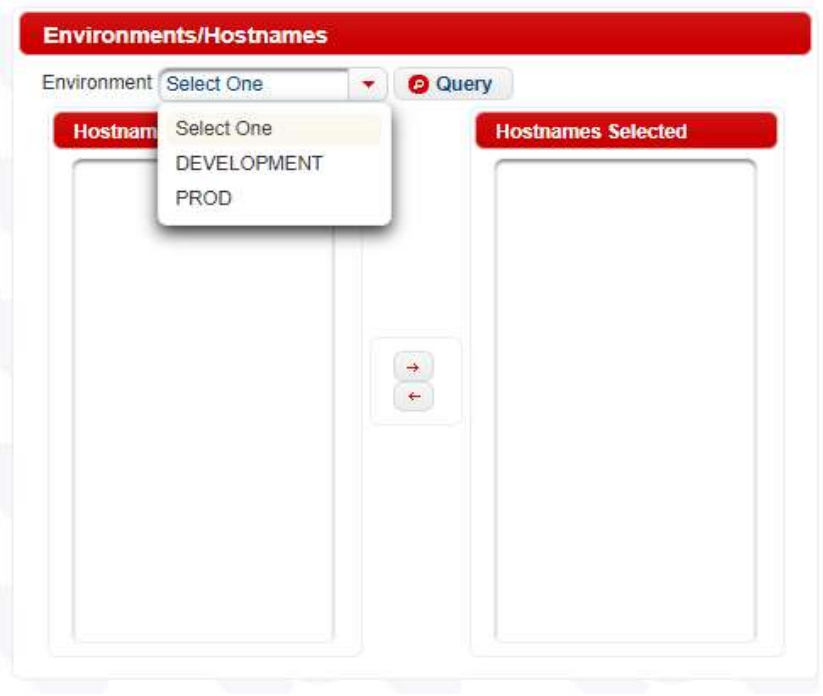
Seconds per Query: This field defines the maximum number of seconds a query can use before it will be automatically terminated. This restriction is used only for online queries and is will not be considered for background queries.

Status: This field defines the current user status. If "Suspended" is set, the user has no longer access to the application.

2.6 Environments/Hostnames

This menu option is used to control the hostnames assigned to an environment.

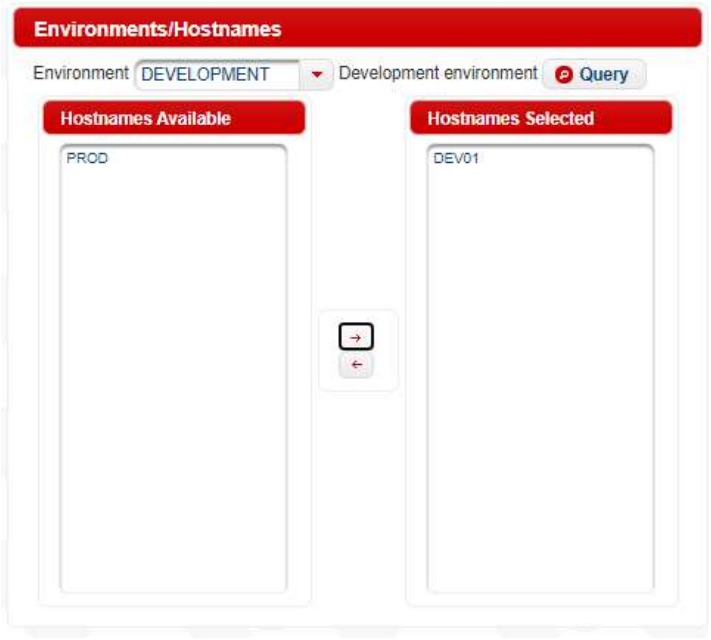
Selecting this option, the page below is displayed.



There is a combo box with all existing environments and two list boxes: the available hostnames, on the left side, and hostnames already associated with the selected environment, on the right side.

Selecting one of the available environments, a query is performed and the two list boxes are updated.

To insert or delete on hostname from a specific environment, just click on the hostname and press the corresponding button to move the hostname out of the list box and include it in the other list box.



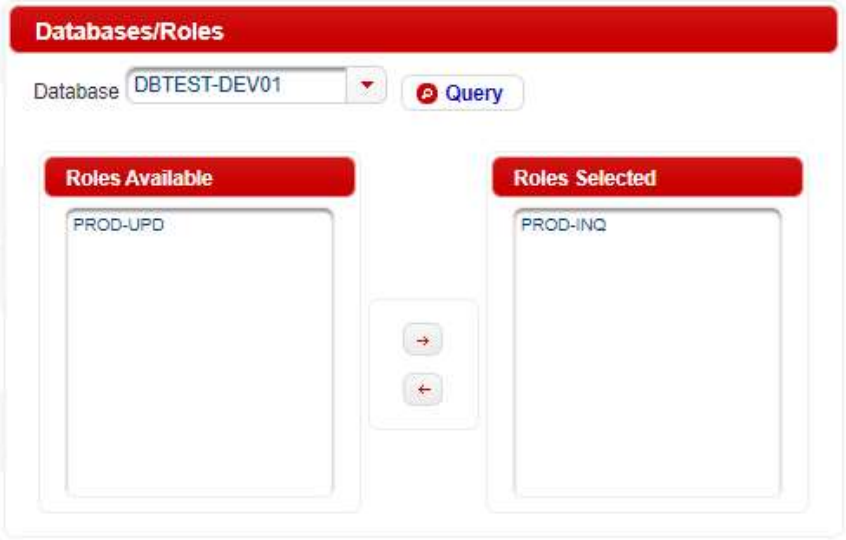
2.7 Databases/Roles

When selecting the “Databases/Roles” menu option, the following page is displayed.

All existing databases are available in a list and it is possible to select one database to query. After that, it is possible to include or exclude a role, if necessary.

All the available roles are listed on the left side and roles already associated with the database/hostname on the right side.

To insert or delete a role from a specific database/hostname, just click on the desired role and press the corresponding button to move the role out of the current list box and include it in the other list box.



The same database can be present in different roles.

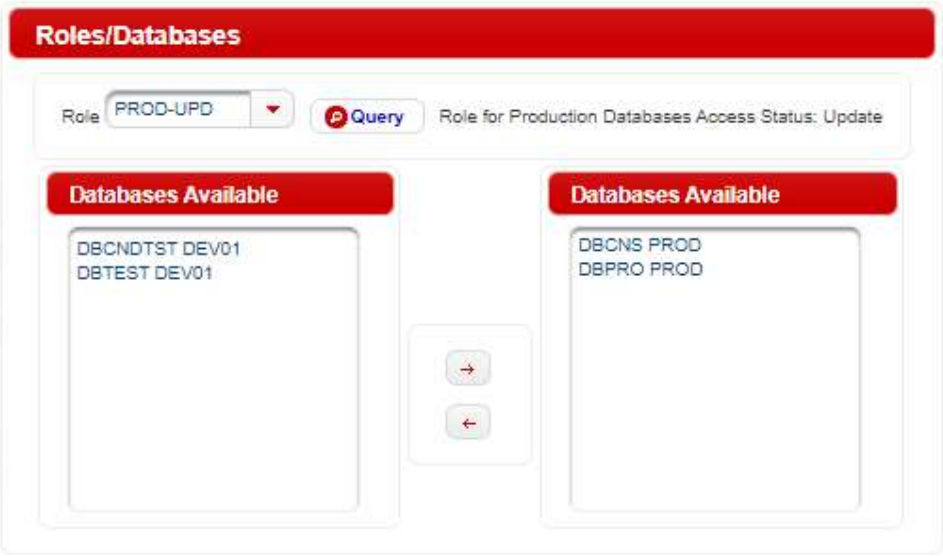
2.8 Roles/Databases

When selecting the “Roles/Databases” menu option, the following page is displayed. This is another way to manage roles and databases.

All existing roles are available in a list and it is possible to select one role to query. After that, it is possible to include or exclude a database, if necessary.

All the available databases are listed on the left side and databases are already associated with the role on the right side.

To insert or delete the database from a specific role, just click on the database and press the corresponding button to move the database out of the current list box and include it in the other list box.



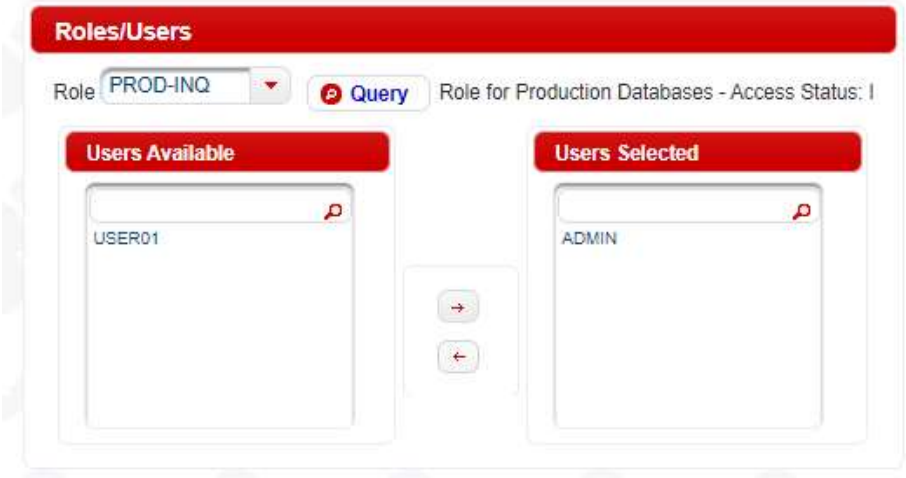
2.9 Roles/Users

When selecting the “Roles/Users” menu option, the following page is displayed.

All existing roles are available in a list and it is possible to select one role to query. After that, it is possible to include or exclude users, if necessary.

All the available users are listed on the left side and users are already associated with the user on the right side.

To insert or delete one user from a specific role, just click on the user and press the corresponding button to move the user out of the current list box and include it in the other list box.



The databases that a user can access are those that are present in any role associated with the user. The inquiry/update mode for each database is defined depending on the role definition. If at least one role has the UPDATE option set for a database, the user will have that privilege for that database.

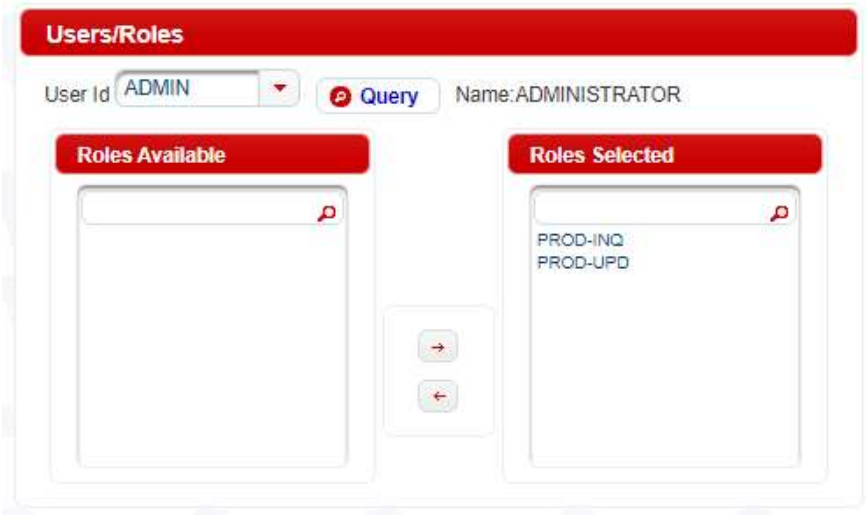
2.10 Users/Roles

When selecting the “Users/Roles” menu option, the following page is displayed. This is another way to manage users and roles.

All existing users are available in a list and it is possible to select one user to query. After that, it is possible to include or exclude a role, if necessary.

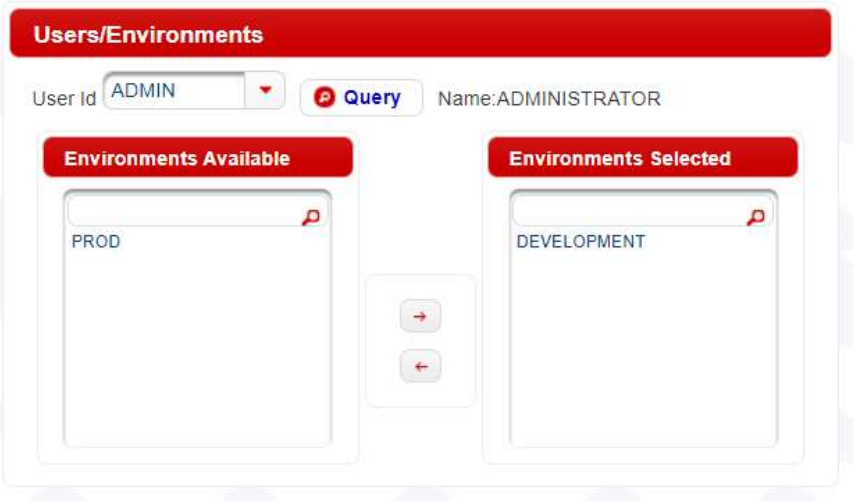
All the available roles are listed on the left side and roles already associated with the user on the right side.

To insert or delete one role from a specific user, just click on the role and press the corresponding button to move the role out of the current list box and include it in the other list box.



2.11 Users/Environments

The function Users/Environments is used when the SDD application will be installed in a mode when no environment is set in the SWENV environment variable. When this variable is not defined, the login page will have an additional combo box for the user to select the environment he wants to work with. The environments present in this combo are defined in this function. See the web application installation later in this document.



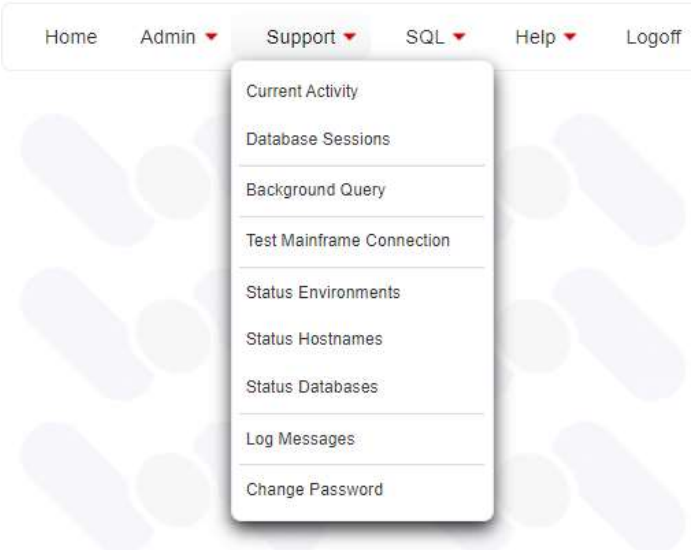
When selecting a user, all environments that are already assigned to the user are listed on the right list box and all available environments are listed on the left list box.

Selecting the environment and using the arrow buttons it is possible to move the environments from one side to another.

The user will be able to use any of the environments he has granted.

3 Support module

In the support module, it is possible to manage database connections, enable and disable environments, hostnames, and databases, check the connected users, and access the log messages. Background queries management can be done using this menu option. A function to check the connection between the application server and the mainframes is also available.



3.1 Current Activity

When selecting this option, a page listing all the connected sessions is displayed. Each SESSIONID is listed. When a user is executing a query using the Query page, the database name, SQL statement, and the time of the beginning of the execution are displayed. The number of SQL statements issued and the number of records returned is displayed for each sessionId.

3.2 Database Sessions

A list of all the opened database connections can be queried using this menu option. The Host Name and Database Name are displayed. The “Time Status” column contains information about when the last status change occurred.

If the Session-Id, User Id, and SQL Text columns are blank, the “Time Status” column indicates the idle time for that connection.

When a user is executing a query using the Query page, the Session ID, SQL statement, and User Id are displayed. In this case, the “Time Status” column indicates since when the query was started.

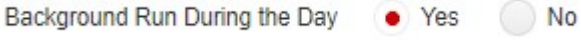
The first column has a button to kill the session in any state, idle or busy. If a busy connection id is killed, the query in progress is ended with an error message.

3.3 Background Queries

The background query is a process where one SQL statement can be executed later, in a background mode. This is important to have long queries executed overnight when the activity has almost no online transactions.

There is a daemon that is running all the time and is responsible for some functions. One of that is checking periodically if any user submits a background query and if that background query can be started.

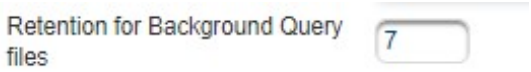
The parameter page for each hostname has a field that defines if a background query can be processed all day or only overnight.



When this option is set with Yes, the daemon can start a background query at any time. If set to No, the background queries will only start between 18h00 and 06h00.

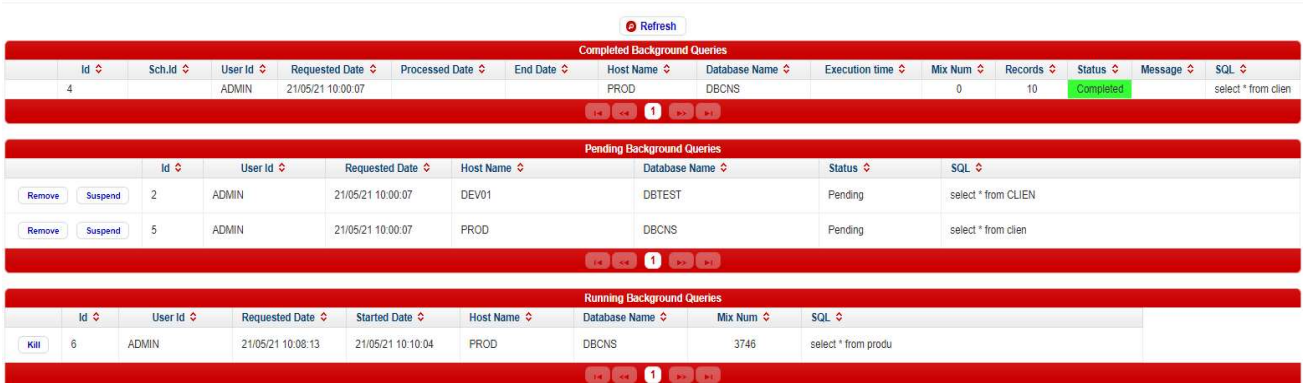
The result of a background query is a CSV file. This file is stored in a folder defined on the hostname page.

The retention period for the files created by a background query is defined on the hostname page. The daemon is responsible for checking all files present on the result folder and removing those who have expired the retention period.



The following image shows the background query page, where the user can see their requests and download the CSV file created during the execution of the submitted query.

The page is divided into three parts: completed requests, pending requests, and running requests.



The three grids show queries already processed, pending, or running, depending on the current status of the request. If the connected user has the SUPPORT or SUPER status, the

background queries from all users are shown. Otherwise, only the background queries from the current user are displayed.

The first column of the “Completed Background queries” grid is a button to download the CSV file. When this button is pressed the “Save As” window is displayed and it is possible to select a local folder to save the file. The download button is available only for queries submitted by the current user and if the file is present on disk. Otherwise, the column will be blank.

The first column of the “Pending Background queries” grid is a button to cancel the scheduled query. The owner of the request and any user with the SUPPORT type can cancel a background query. The record status is changed to “C” and the information is visible on the “Completed Background queries” grid, with a message text indicating who canceled the query.

The first column of the “Running Background queries” grid is a button to kill the execution. The SQL statement is aborted and a message is associated with the request indicating that the request was aborted. The request is moved to the Completed Background queries” grid.

When clicking on the kill button the action resulted in a change in the status field of the request to “K”. The effective abort procedure is performed by the daemon, who started the request. A thread is running on daemon(**CheckQueries**) to find requests that the status field is marked to be killed. The connection is closed by the daemon and the message and status fields are updated to reflect the kill situation.

3.4 Test Mainframe Connection

The support has an option to verify if the connection between the application server and the mainframes of a specific environment. The 

button will perform the test for each hostname checked in the grid.

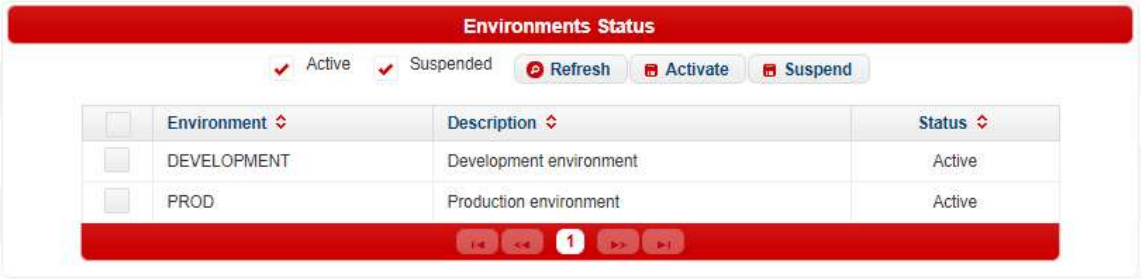
In the end, a message will be displayed indicating the result of the test and which hostname is not currently available.

3.5 Status Environments

This function allows support users to check the current status of environments and suspend activity in a specific environment, if necessary.

After selecting the **Status Environment** menu option, a page is displayed with all existing environments in a grid.

The query can be restricted just to one status(Active or Suspended) or list all environments using the appropriate checkbox available above the grid.



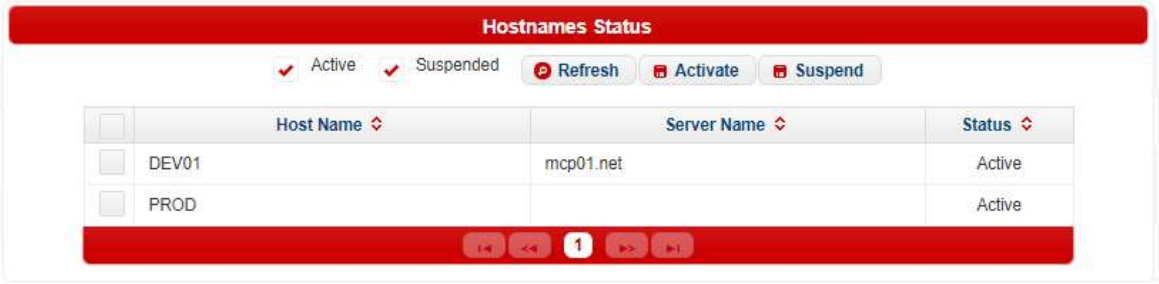
Select the environments to be changed and press the buttons Activate or Suspend to enable or disable any activity in that environment. An appropriate message will be sent.

3.6 Status Hostnames

This function allows support users to check the current status of the hostnames and suspend the activity in a specific hostname, if necessary.

After selecting the **Status Hostnames** menu option, a page is displayed with all existing hostnames in a grid.

The query can be restricted just to one status(Active or Suspended) or list all hostnames using the appropriate checkbox available above the grid.



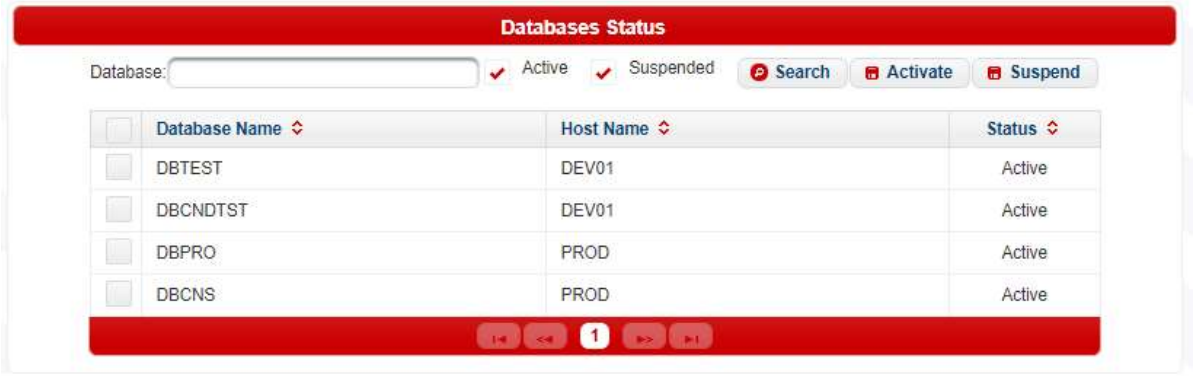
Select the hostnames to be changed and press the buttons Activate or Suspend to enable or disable any activity in that hostname. An appropriate message will be sent.

3.7 Status Databases

This function allows support users to check the current status of the databases and suspend the activity in a specific database, if necessary.

After selecting the **Status Databases** menu option, a page is displayed with all existing databases in a grid.

The query can be restricted just to one status(Active or Suspended) or list all databases using the appropriate checkbox available above the grid. There is a field to filter the database name. It is possible to use the character % to perform a “like” query.

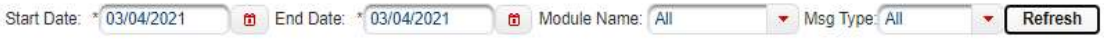


Select the hostnames to be changed and press the buttons Activate or Suspend to enable or disable any activity in that hostname. An appropriate message will be sent.

3.8 Log Messages

This page shows the messages generated while using the application. The message date/time, message type, module that generated the message, and the text are displayed when an interval date is specified.

The query uses the fields in the select area below to create the grid.



The **Start Date** and **End Date** columns define the period of the log records to be displayed. If the columns **Module Name** and **Msg Type** have the default value **All**, all records from that period will be listed. If a specific module or type is defined, the query will be filtered.

There are three message types:

SUCCESS: This message type indicates that the text is related to a successful operation. Not all successful operations are logged but all meaningful are recorded.

WARNING: This message type is related to texts generated by the application after situations identified as temporary.

ERROR: this message type is related to operations that were not completed by the application due to security, database, or software error.

There are five modules:

ADMIN: this module is related to admin operations, such as definitions of hostnames, databases, and users.

BACKGROUND: this module represents the background query actions.

DAEMON: this module is related to functions performed by the daemon process.

SUPPORT: this module is associated with support actions, such as deleting a database session or aborting a background query.

SECURITY: this module is associated with functions that involve connections

SQL: this module is associated with functions that involve executing DMSII SQL statements.

Log Messages

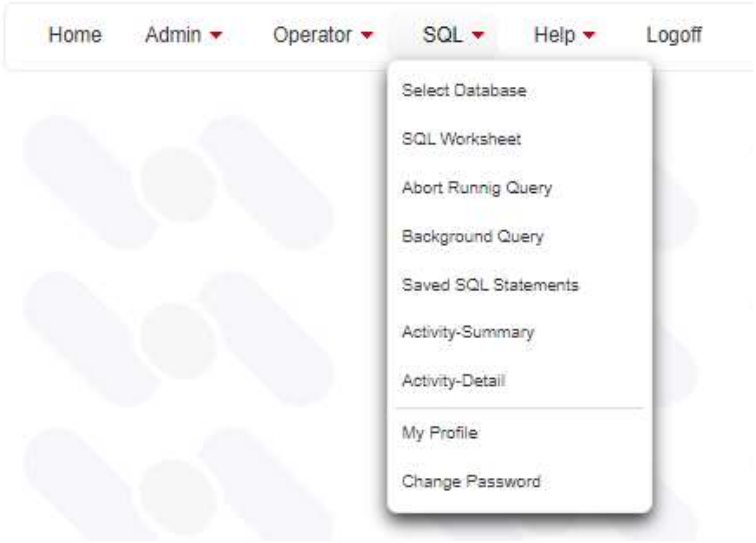
Id	Date	Msg Type	Module	Text
175	03/04/2021 17:50:53	INFORMATION	SECURITY	User USER01 connected(Ip=0:0:0:0:0:0:0:1, SessionId=5883B3739A65
174	03/04/2021 17:50:47	SUCCESS	SECURITY	ADMIN stored the user record USER01
173	03/04/2021 17:26:47	SUCCESS	ADMIN	User USER01 has been added to profile PROD-INQ list by user ADMIN
172	03/04/2021 17:08:33	SUCCESS	ADMIN	Profile PROD-INQ has been added from database DBCNS-PROD list by
171	03/04/2021 17:03:03	SUCCESS	ADMIN	Hostname DEV01 has been added to environment DEVELOPMENT list
170	03/04/2021 17:02:54	SUCCESS	ADMIN	Hostname ABRU0111 has been removed from environment DEVELOPM
169	03/04/2021 17:00:08	INFORMATION	SECURITY	User ADMIN connected(Ip=0:0:0:0:0:0:0:1, SessionId=8D00D177E7472
168	03/04/2021 15:42:22	SUCCESS	ADMIN	Profile PROD-UPD created by user ADMIN
167	03/04/2021 15:42:03	SUCCESS	ADMIN	Profile PROD-INQ created by user ADMIN
166	03/04/2021 15:05:23	INFORMATION	SECURITY	User ADMIN connected(Ip=0:0:0:0:0:0:0:1, SessionId=621708F1AA2BD
165	03/04/2021 15:03:38	INFORMATION	SECURITY	User ADMIN connected(Ip=0:0:0:0:0:0:0:1, SessionId=C8E7294556BE3
164	03/04/2021 14:59:06	SUCCESS	ADMIN	Database DBCNS created by user ADMIN
163	03/04/2021 14:58:52	SUCCESS	ADMIN	Database DBPRO created by user ADMIN
162	03/04/2021 14:58:43	SUCCESS	ADMIN	Database DBCNDTST created by user ADMIN
161	03/04/2021 14:58:02	SUCCESS	ADMIN	Database DBTEST created by user ADMIN
160	03/04/2021 14:27:24	SUCCESS	ADMIN	Hostname DEV01 stored by user ADMIN
159	03/04/2021 14:26:31	SUCCESS	ADMIN	Hostname PROD created by user ADMIN
158	03/04/2021 14:15:32	SUCCESS	ADMIN	Hostname PROD deleted by user ADMIN
157	03/04/2021 14:15:21	SUCCESS	ADMIN	Hostname PROD created by user ADMIN
156	03/04/2021 14:15:02	INFORMATION	SECURITY	User ADMIN connected(Ip=0:0:0:0:0:0:0:1, SessionId=F991373DB8FA1

1 2 3 4 50

4 SQL Module

This module is used by basic users and is responsible for submitting SQL statements and functions available around the SQL statements submitted.

The menu option is shown below.



4.1 Select Database

The Select Database option allows the user to select one of the available databases. The list of databases depends on the roles assigned to the user and the databases included in these roles. Environments, Hostnames, and Databases suspended will be considered to include a database on the **Select Database** page.

The user must select the desired database from the list and the **SQL Worksheet** page will be displayed. The current database selected is displayed in the top right corner of the page. Only the databases from available servers are listed.

If the resource is not available on the mainframe a message will be displayed after the selection indicating the problem.

4.2 SQL Worksheet

This option allows the user to submit SQL statements to a DMSII database previously selected. The resulting records are shown in a grid and, depending on the number of records returned, a paginator can be used to see all the records, page by page.

As an option, the grid content can be exported to a CSV or Excel file.

The number of returned records depends on the attribute maximum number of records returned presently on hostname, role, and user definition. A smaller number of them will be used.

It will be possible to define a query to be run in background mode, avoiding waiting for the result with the session connected.

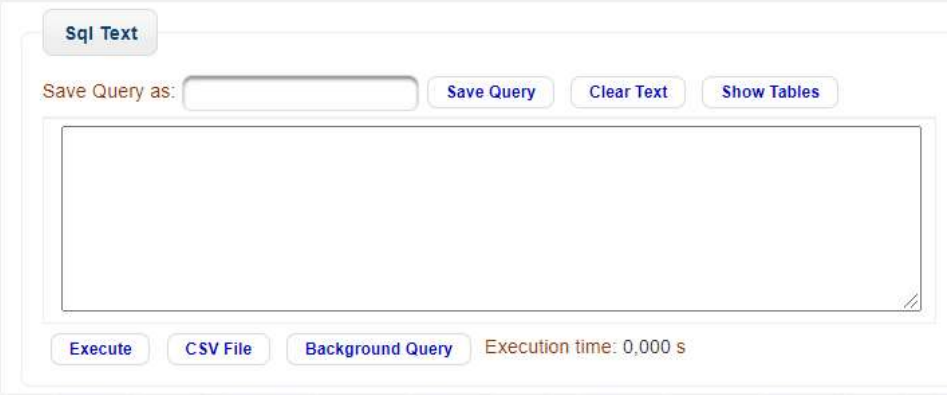
All the queries created by users can be saved to a SQL table and be restored any time the user wants. There are two options to do this. In the SQL Worksheet page or using a specific menu option. When saving SQL statements, it can be defined as “public” and be available to any user with the same role. A public SQL statement can only be updated by the owner.

Using the Admin, it is possible to control the number of simultaneous queries and include some restrictions to force users to design optimized queries that return few records and then submit a background query to get the full result.

There is a limitation of one execution at a time, even if the user uses different browsers to have different session ids. The session control module has information about all running queries and blocks any attempt by the user to send a second query when there is an execution query is in progress.

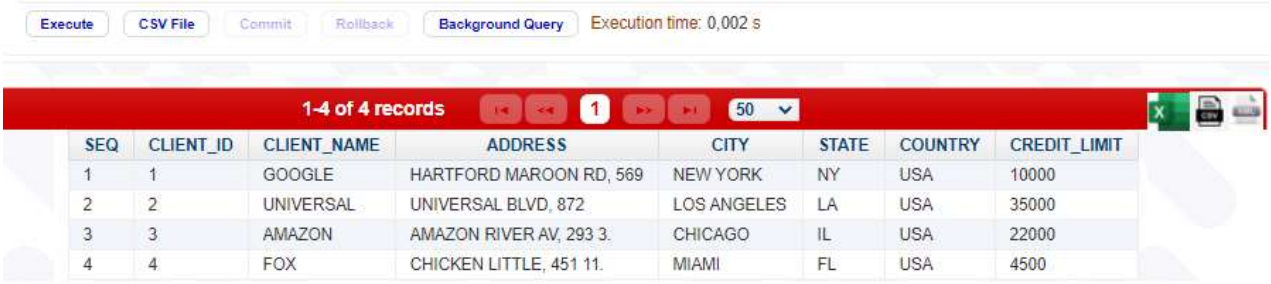
If an attempt is made to send a second query by the same user, an error message will be displayed.

When the SQL Worksheet option is selected, the following page is displayed, and it is possible to define a SQL statement and send it to a selected database.



After submitting a query, the resulting records will be displayed in a grid. That grid will display, by default, 50 records per page. The grid header has a paginator component that allows the user to navigate through the resulting records.

The image below contains a typical grid header.

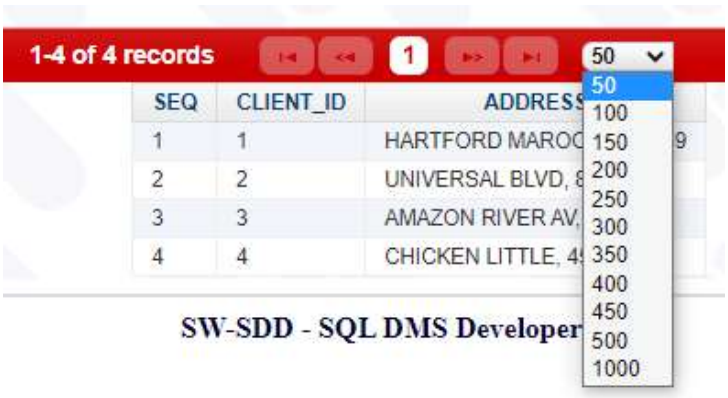


SEQ	CLIENT_ID	CLIENT_NAME	ADDRESS	CITY	STATE	COUNTRY	CREDIT_LIMIT
1	1	GOOGLE	HARTFORD MAROON RD, 569	NEW YORK	NY	USA	10000
2	2	UNIVERSAL	UNIVERSAL BLVD, 872	LOS ANGELES	LA	USA	35000
3	3	AMAZON	AMAZON RIVER AV, 293 3.	CHICAGO	IL	USA	22000
4	4	FOX	CHICKEN LITTLE, 451 11.	MIAMI	FL	USA	4500

Update SQL statements can be submitted only for users and profiles with UPDATE set to true. When an update statement is processed, a transaction state started and the corresponding message is sent to the user. The transaction state can be finished using COMMIT or ROLLBACK statements.

A transaction state can be inactive for 2 minutes. After that, SDD will automatically abort the transaction and the locked records will be free.

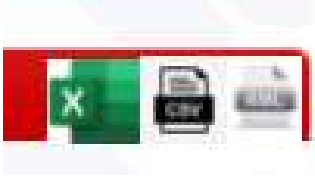
The footer has a combo box with the default value of 50 records per page. It is possible to change this number, selecting a different value:



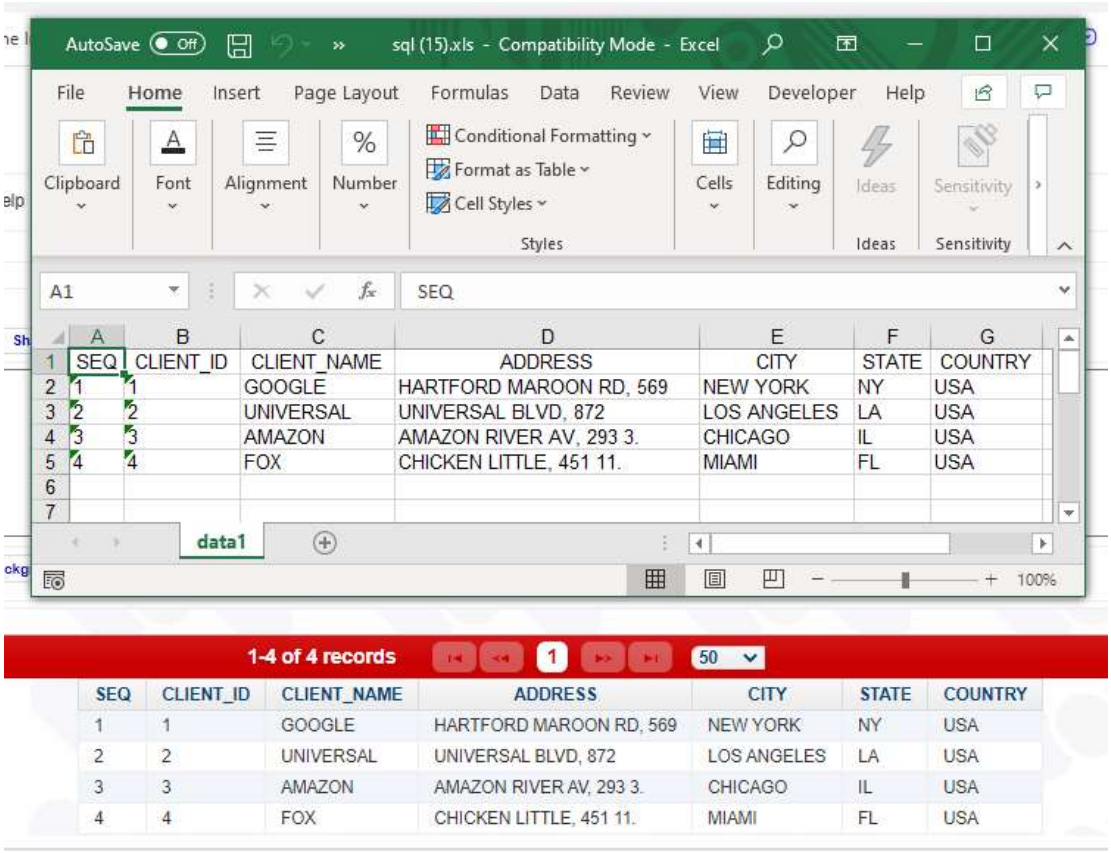
The grid will be resized according to the new number of records selected.

To select a new page it is possible to click over the page number or select the arrows to move to the next, previous, first on the last page. The information on the left side of the paginator component shows the number of records available, in this example, 4 records.

On the right side of the paginator, there are three icons to download the grid information. The options available are Excel, CVS, or XML files.



Clicking on the desired type of file, a window will appear to allow the user to define the destination folder.



	A	B	C	D	E	F	G
1	SEQ	CLIENT_ID	CLIENT_NAME	ADDRESS	CITY	STATE	COUNTRY
2	1	1	GOOGLE	HARTFORD MAROON RD, 569	NEW YORK	NY	USA
3	2	2	UNIVERSAL	UNIVERSAL BLVD, 872	LOS ANGELES	LA	USA
4	3	3	AMAZON	AMAZON RIVER AV, 293 3.	CHICAGO	IL	USA
5	4	4	FOX	CHICKEN LITTLE, 451 11.	MIAMI	FL	USA
6							
7							

SEQ	CLIENT_ID	CLIENT_NAME	ADDRESS	CITY	STATE	COUNTRY
1	1	GOOGLE	HARTFORD MAROON RD, 569	NEW YORK	NY	USA
2	2	UNIVERSAL	UNIVERSAL BLVD, 872	LOS ANGELES	LA	USA
3	3	AMAZON	AMAZON RIVER AV, 293 3.	CHICAGO	IL	USA
4	4	FOX	CHICKEN LITTLE, 451 11.	MIAMI	FL	USA

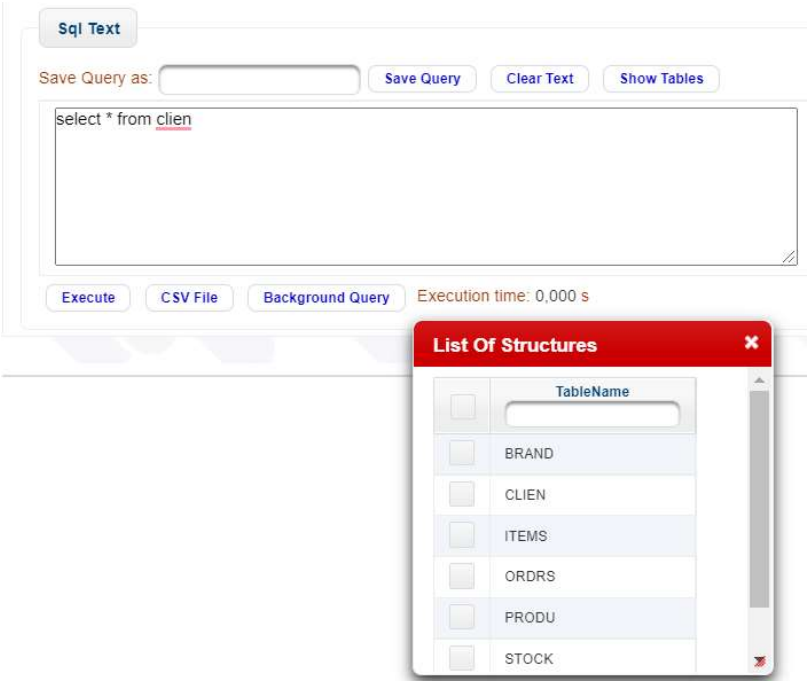
The field “SQL Text” is used to define the query. The SQL syntax must follow the SQL ANSI92. If you want to create a query with joins between DMSII databases or between a DMSII database and an Oracle or SQL Server database, a specific syntax must be used. This syntax will be described later in this document.

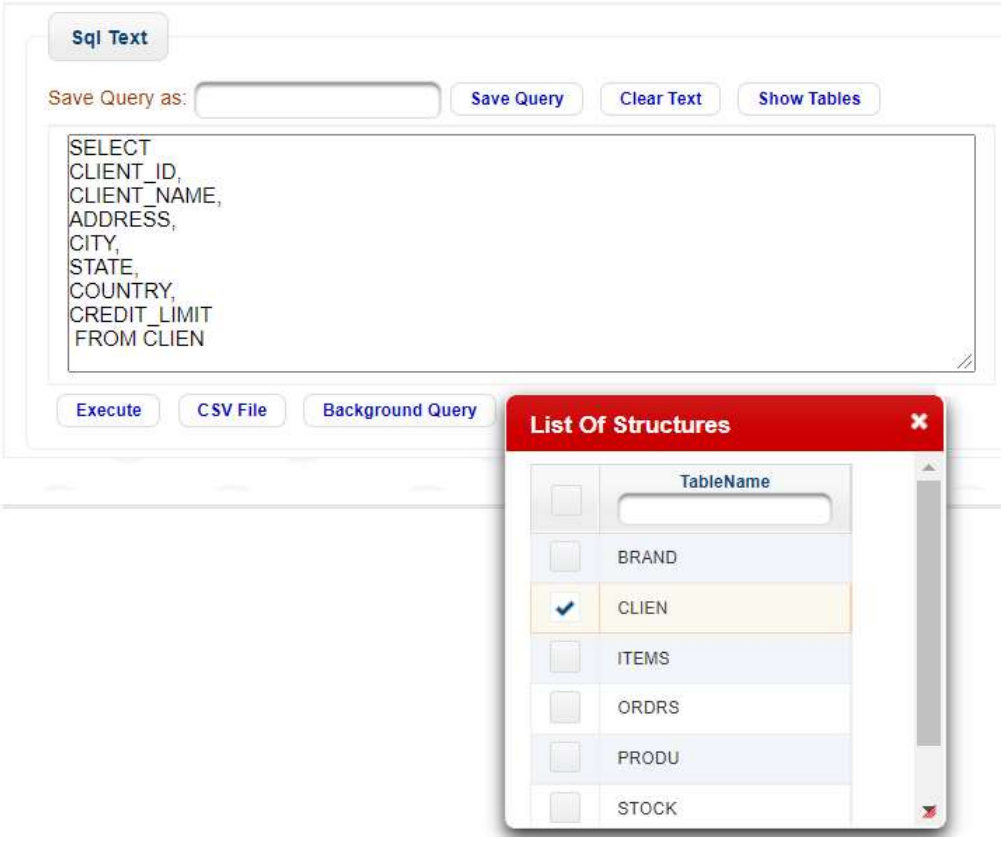
Above the SQL text box, there is a way to save the query you created in the SQL text box. You can use the text box “Save Query as:” to define the name for the query and press the “Save Query” button. This action stores the query assigned to this query name and the saved query can be accessed using the “SQL Statements Catalog” option in the SQL menu. If you forgot to define a query name, a message will be displayed:

The "**Clear text**" button clears the SQL text box.

The "**Show Tables**" button opens a window with all the existing tables. The image below shows an example of this window. This option is available only for connections using MCPSQL.

It is possible to create a query using this window. When double-clicking in the cell containing a table name, a SELECT SQL statement is automatically created and the text is stored in the "SQL Text" field. All existing columns are included and the statement can be edited, if desired, before submitting the query.

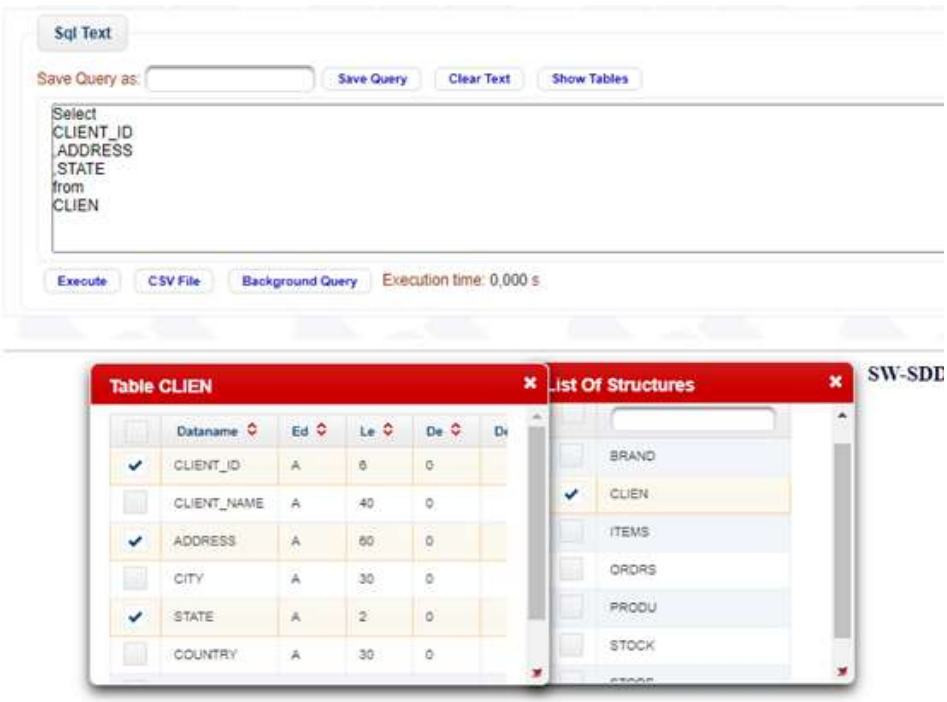




With a click in the select column, the first column of the tables window, a second window is opened and all the table fields are displayed. It is possible the create a query specifying only the desired columns.

This new window has the column name, edit, length, and the number of decimals.

Each newly selected field is included in the SQL statement. Deselecting a field removes the field from the query.

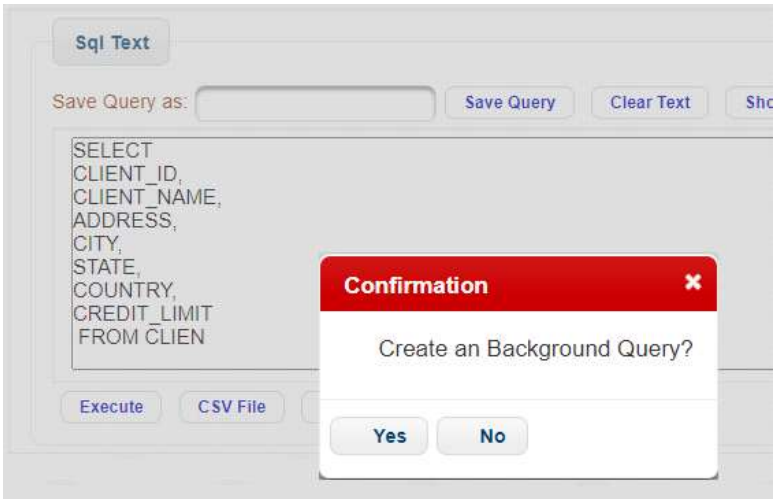


The “Execute” button is used to submit a query to a DMSII database. The records retrieved will be displayed in a grid and the “Execution Time” field is updated with the query time.

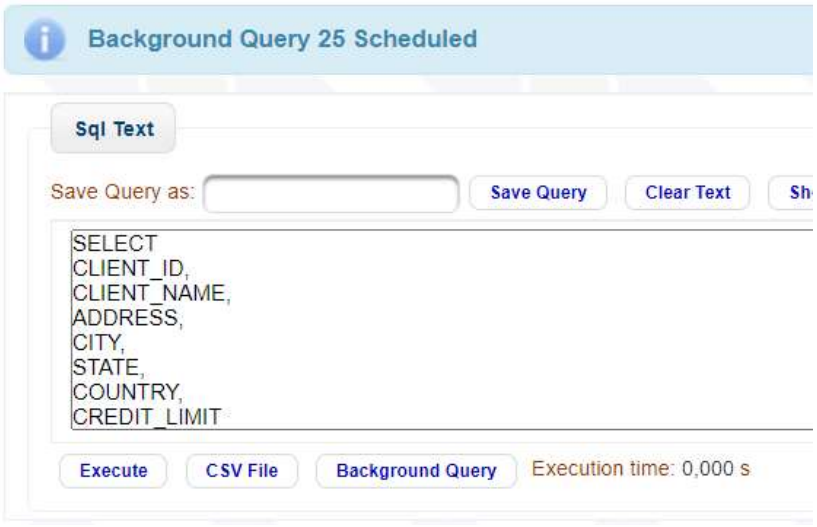
The CSV button is an option to execute a query and create a CSV file without return the records in a grid. The file download will be started after the end of the query.

The “Background Query” button is used when the user wants to submit the SQL in background mode. Queries in background mode are processed in a batch way by the daemon and when the resources are available. The result of a Background Query is a CSV file that contains the records returned by the SQL statement.

A confirmation box is sent before storing the request.



The confirmation page informs the Background Query number generated for this execution. The “**Background Query**” option in the SQL menu can be used to check the execution and to download the CSV file created. The effective execution of this query is made by the daemon process. This process must be in execution to have the background queries processed.



4.3 Abort Running Queries

This function can use used if a running query must be killed. As the SQL Worksheet page is in use, a new tab should be used and the Abort Running Query should be selected.

The current query in progress is displayed and a button is available to kill the query. A message will be sent to the SQL Worksheet page indicating that the query was aborted.

4.4 Background Queries

This function was described in topic 3.3. The difference when a user accesses this function is that a download button will be available in the first column in the Complete Background Queries grid if the resulting CSV file is present in the result folder. Pressing the download button, the file can be copied to the user's machine.

A periodic routine is performed by the daemon program to verify if the files created by a background query must be removed due to the expiration of the retention days.

4.5 Saved SQL Statements

This page returns all the saved SQL statements created by the user or created by other users but defined as public. An Admin user can see all existing SQL statements but only their statements can be changed.

The first button (select button) will be enabled if the current database, selected using the Select Database page, is the same as the stored query. When this button is enabled, the user can press it to navigate to the SQL Worksheet page and the SQL text will appear in the textbox, ready to be executed. Public queries from other users cannot be changed and this button will always appear disabled.

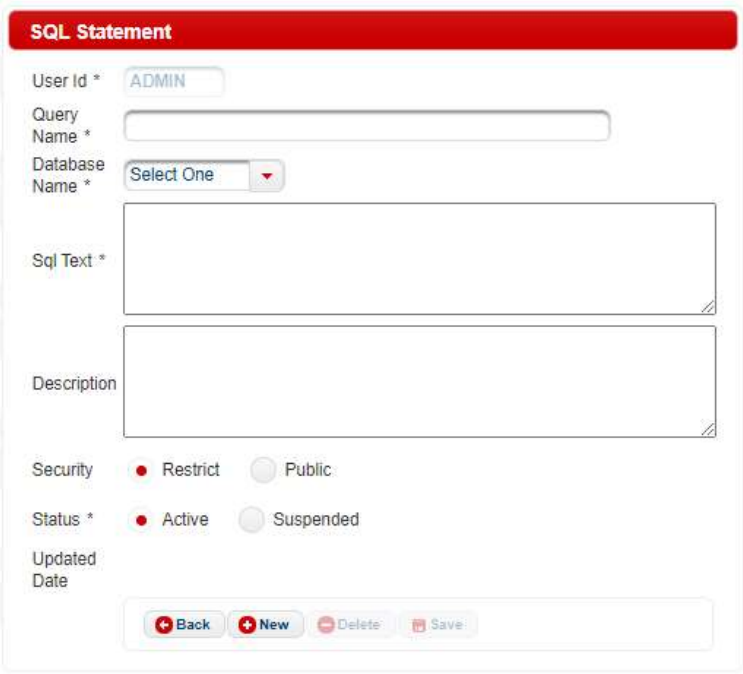
The second button (update button)  is enabled when the query belongs to the current user. Only the owner can update his SQL statement.

The user can filter the records in the grid using the SQL Statement Filter input text and pressing the Query button. The text must use the like character %. If no filter is used, all SQL Statements created by the user or created by other users with the secure mode public are listed in the grid.

The resulting page is displayed below.

To create a new SQL statement, the button 

should be pressed. This action displays the definition page to create a new statement. The query is stored in the application database and can be retrieved at any time.



The available attributes are:

User ID: The field “User Id” is disabled. It is not possible to create a query associated with another user. It is used only as query information.

Query Name: this field defines the name of the query. This name will be used to select the defined SQL statement.

Database Name: This field is to assign the database related to this SQL statement. A combo is available with all databases defined for the environment.

SQL Text: this field stores the SQL statement. No syntax is validated and the user needs to test the query first using the SQL Worksheet page.

Security: this field defines the security mode for the query. The options are:

> ***Restrict:*** *Only the creator can access the content of this query.*
>
> ***Public:*** *Any user can have access to the SQL statement. This field controls the records displayed in the query grid of the queries page.*

Status: This field defines the current SQL Statement status. If “Suspended” is set, this query will not be available to other users, if public, and cannot be selected in the grid to navigate to the SQL Worksheet page.

To update an existing SQL Statement, the button should be pressed on the SQL Statements query page. The definition page will be displayed with the fields **Query Name** and **Database Name** disabled. The user can change the desired information and press the save button. To delete the SQL statement, the delete button should be pressed.

4.6 SQL Activities-Summary

The SQL ACrivities-Summary page is used to query the user activity per day basis. The user can specify a start date and all activities will be displayed in descending order. If the logged-in user has the type of Admin set, it is possible to see the activities from all users. A combo box will be present and can be used to select a specific user.

The resulting query has these fields:

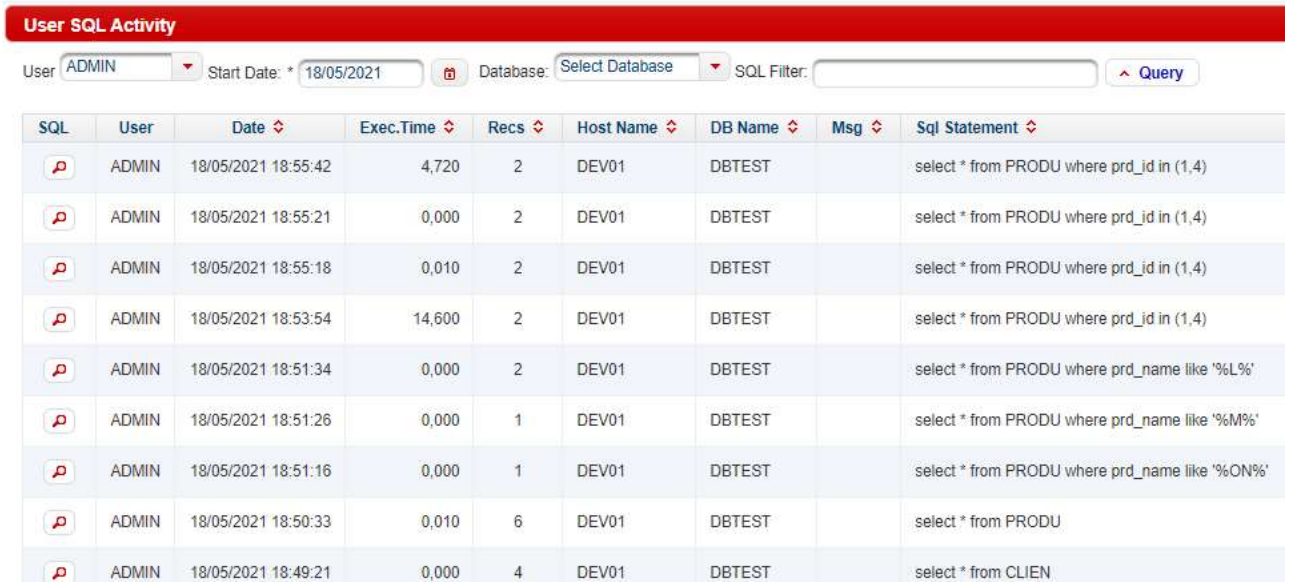
No.Execs: number of SQL statements submitted on the date

Total Time: This column shows the total number of seconds for process all the SQL statements listed in column N.Execs

Recs: this column shows the total number of records returned by the queries submitted

Clicking on the icon present in the first column of the grid displays a page that lists all SQL statements submitted by the user at that date(SQL Activities-Detail page).

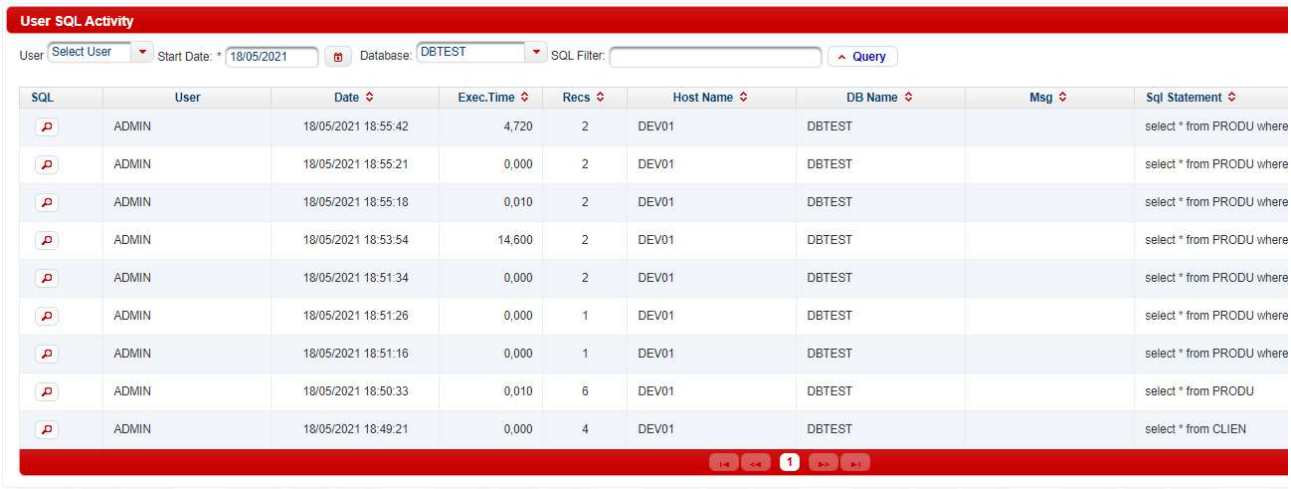
If this page is selected by a basic user, only that user's activities are listed.



The first column has a button to navigate to the SQL Worksheet page coping the corresponding SQL statement to the SQL text box. This button is enabled for each line assigned to the same database that is currently selected. Using this feature the user can reuse a previous statement easily.

4.7 SQL Activities-Details

This page shows the SQL statements submitted on a given date. The user combo box appears only for users of type **Support**. This type of user can view the activities of any other user. If the userbox is not informed, activity related to all users will be displayed.



SQL	User	Date	Exec.Time	Recs	Host Name	DB Name	Msg	Sql Statement
	ADMIN	18/05/2021 18:55:42	4,720	2	DEV01	DBTEST		select * from PRODU where
	ADMIN	18/05/2021 18:55:21	0,000	2	DEV01	DBTEST		select * from PRODU where
	ADMIN	18/05/2021 18:55:18	0,010	2	DEV01	DBTEST		select * from PRODU where
	ADMIN	18/05/2021 18:53:54	14,600	2	DEV01	DBTEST		select * from PRODU where
	ADMIN	18/05/2021 18:51:34	0,000	2	DEV01	DBTEST		select * from PRODU where
	ADMIN	18/05/2021 18:51:26	0,000	1	DEV01	DBTEST		select * from PRODU where
	ADMIN	18/05/2021 18:51:16	0,000	1	DEV01	DBTEST		select * from PRODU where
	ADMIN	18/05/2021 18:50:33	0,010	6	DEV01	DBTEST		select * from PRODU
	ADMIN	18/05/2021 18:49:21	0,000	4	DEV01	DBTEST		select * from CLIEN

The resulting query has these fields:

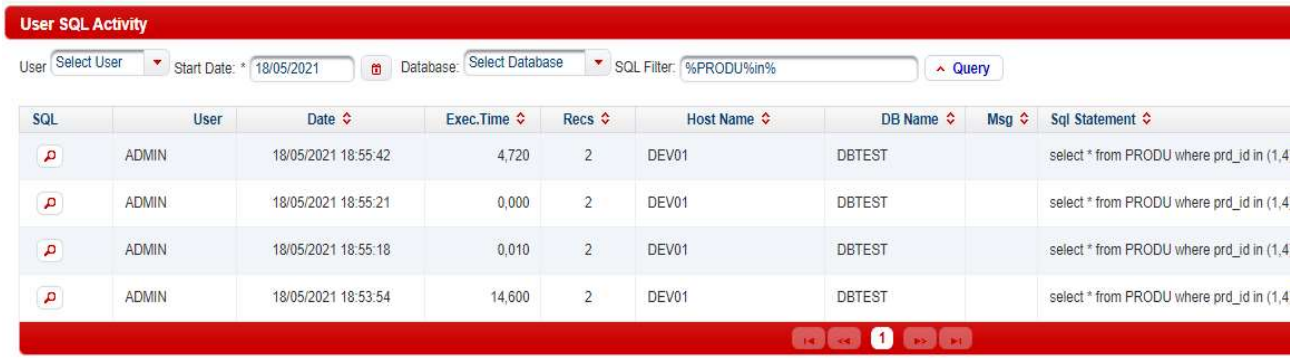
Total Time: This column shows the total number of seconds for process all the SQL statements listed in column N.Execs

Recs: this column shows the total number of records returned by the queries submitted

Msg: Any error message that occurred when the SQL statement was processed as listed in this field

SQL Statement: this column shows the SQL statement processed

It is possible to use the filter field to select the SQL statement that had the text. The filter field will be used in a LIKE statement to return only records that contain the selected text.



If this page is selected by a basic user, only that user's activities are listed.

4.8 My Profile

This function shows the user information, including the type of user, limits, and all roles and databases that can be selected.

4.9 Change Password

This menu option allows users to change their passwords. The current password must be informed and the new password must be informed twice.

After updating the password, the home page will be displayed with the following message:

5 Daemon Module

This module is responsible for some administrative tasks and manipulating queries that users set to run in a background mode.

A shell script or bat file must be used to run this process and keep it running. If the daemon is not in execution, no background queries will be started.

The retention period for CSV files created by background queries is managed by the daemon. The number of days the file is available is defined on the Hostname page and can be different per hostname.

Periodically, the daemon verifies if any background query was submitted by the users and, depending on the limits defined for the mainframe/database selected, the query is started.

The result of a query will be stored by the background module in a CSV file. The file will be stored in the directory specified on the Hostname page. The users can query any time about the status of their request and, when finished, download the file created.

The maximum number of threads can be defined using a parameter for running the daemon. This represents the maximum parallel runs for all mainframe servers. During the daemon execution. after reaching the limit of parallel runs defined, the next background query request will wait until one of the running background queries ends. If no parameter is assigned, the default value is 4 simultaneous queries.

One log file is created and contains all messages generated by the daemon during its process.

All the log messages created by the daemon program regarding the process of one background query are stored in the message table. These messages can be queried using the Log Messages page. The module name associated with these daemon messages is BACKGROUND.

All additional messages generated by the daemon are also recorded in the message table. These messages be queries using the Log Messages page and are identified by the module name DAEMON.

The log file created with the messages generated by the daemon always starts with the daemon version. The lines below show a typical daemon start messages:

```
27-03-2021 11:21:14: ***********************************
27-03-2021 11:21:14: *       SDD Daemon started          *
27-03-2021 11:21:14: *          Version 21.7.1           *
27-03-2021 11:21:14: ***********************************
```

6 Installation

The current version of SDD uses a web application, a batch process(daemon), and an Oracle database. All files needed for the installation process can be downloaded from Github.

The SQL Database Developer works using Unisys MCPSQL or JDBC. One of these software must be installed on the mainframe and the databases must be configured in their config files.

- ## Creating the schema

  The first step to install the application is creating the database schema. SDD uses an Oracle database to store configuration settings and activity. The application can use any Oracle database version, including Oracle Express.

  You need to create a specific username to install the database objects. This username and password will be used to update the sdd.properties later.

  When connected using the username created, you can execute the script SDD.sql. This script creates all database objects and inserts an initial record in the USERS table related to the user **ADMIN** and password **admin**. This user can be used to connect to the Web application with administrator privilege and configure the setup of the environment.

  The script SDD_drop.sql can be used to drop all objects created.

- ## Installing the Web application

  The war file contains an embedded Tomcat. The command line to start the application is:

  ```
  java -jar Sdd.war
  ```

  Some additional information is needed to inform the application of some configurations.

Spring properties

Spring properties configuration can be stored in any folder. The sdd.properties sample file is available on GitHub. Parameters must be changed according to the environment that is being configured, like the parameters to access the Oracle database, such as:

spring.datasource.url, spring.datasource.username, spring.datasource.password: to define the database connection to access the drs tables;

server.port: to defined the desired port for the application;

server.address: can be used to define the IP address of the server where DRS is running;

logging.path, server.error.path: to specify the location of log files;

To inform the application which configuration file to use, a java argument must be defined to when the application is started:

```
java -Dspring.config.location=<folder name>\sdd.properties Sdd.war
```

SWDIR_SETUP

This variable must contain the folder where the configuration files will be stored.

If you are using the Unisys JDBC to connect to DMSII databases, you need to configure one property file per database. The property file name uses the same name you created using the Databases page.

The database name defined on the SDD application can be different from the existing DMSII database on the mainframe. The property file has a Connect String and that is the place you assign the existing DMSII database name. For example, you can define a database TEST using the Databases page and create a property file **TEST.properties** with a connect string assigned to the TESTDB:

```
connect_string=jdbc:unisys:mcpsql:Unisys.DMSII:resource=DBTST;host=192.168.16.5;port=2012
```

If you are using the Unisys MCPSQL to connect to DMSII databases, you need to configure the parameters in a file called **mcpsql.properties**.

The sample properties files are present in Github.

configDir

This property can be used when executing the java with -D option. The use of this variable is an alternative to SWDIR_SETUP.

```
java -DconfigDir=<folder name>
    -Dspring.config.location=<folder name>\sdd.properties Sdd.war
```

SWENV

This variable defines the environment to which the instance will refer. The oracle schema is multi-environment, and it is possible to assign users to different environments. The databases included in the profiles assigned to a user and present on mainframes of a specific environment will be available.

Using this variable, it is possible to install different tomcat servers for different environments. The tomcat servers can be configured using different ports on the same server or installed in different servers. The environment name created using the Environments page must match the value defined in this variable.

When the corresponding URL is accessed and the login page is sent, it refers to a specific environment defined in SWENV, displayed on the login page. In the example below, the environment is DEVELOPMENT.

If SWENV is not defined, the login page will have an additional combo box when the user must select one of the authorized environments he can work. The function Users/Environments is used to define the environments one user can access. is With this configuration, a Tomcat instance can work with any environment. The login page sent when the SWENV is not defined is shown below:

Environment

This property can be used when executing the java with -D option. The use of this variable is an alternative to SWENV.

```
java -DconfigDir=<folder name>  -DEnvironment=DEVELOPMENT
    -Dspring.config.location=<folder name>\sdd.properties Sdd.war
```

- **Installing the Daemon**

The daemon process is a jar file that has the function to run the background processes and to perform some administrative tasks.

The sddDaemon has two parameters:

<environment>: define the environment the execution will be assigned

<maximum number of parallel runs>: this optional parameter define the maximum number of parallel runs that will be possible

GitHub contains a folder called sddDaemon that includes the jar file, a bat file, and a shell script.

To install this component, copy the folder sddDaemon and paste it in the machine where the daemon will be executed. The log folder must be present to store the log files. Update the script with the new location.

Create a scheduled task on Windows or an entry in the Linux crontab file to run this jar permanently.