VAN HELVOIRT AUTOMATISERING

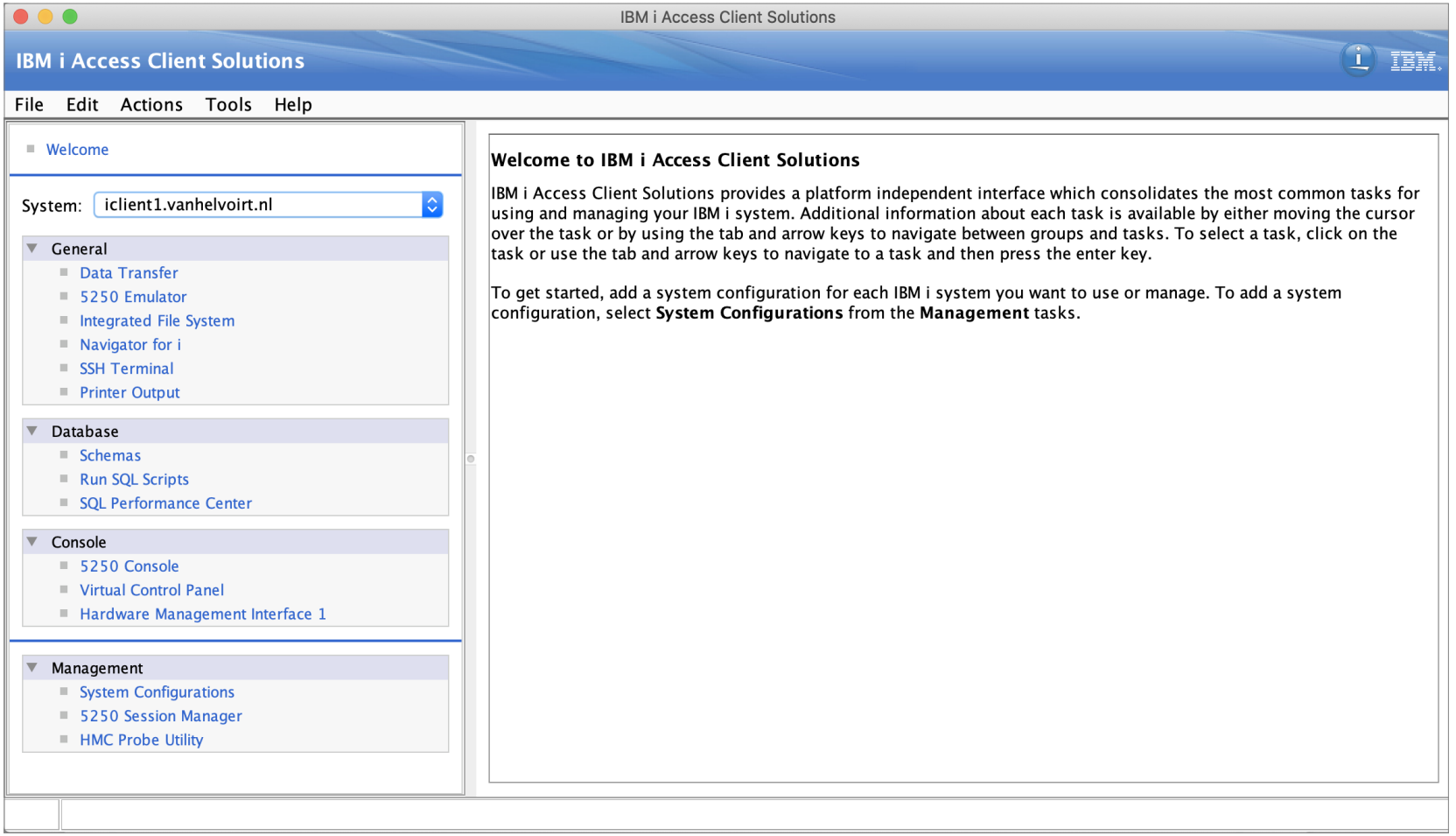
IBM i Access Client Solutions
SQL Performance Center
10 June 2020

Rudi van Helvoirt
rvanhelvoirt@vanhelvoirt.nl

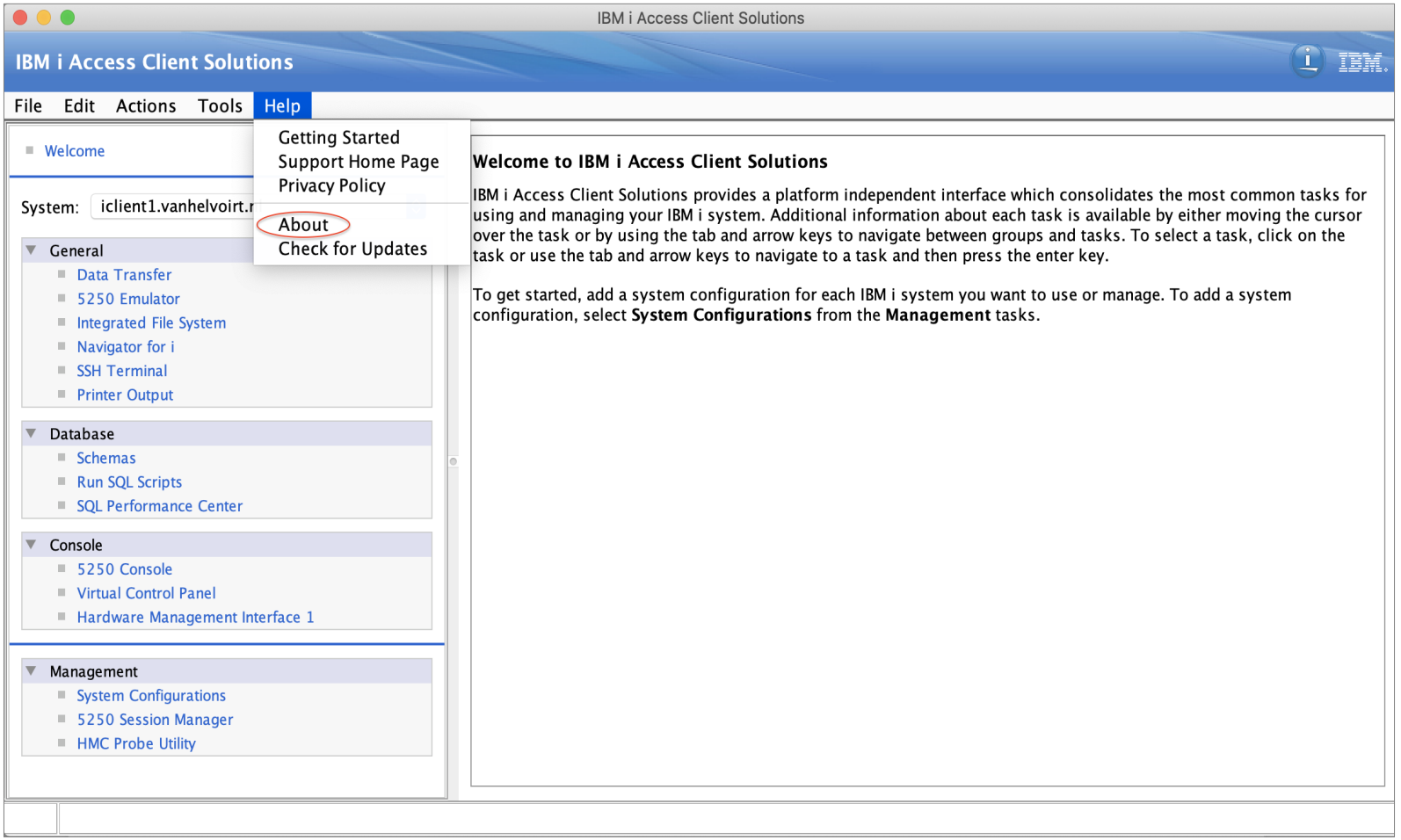
Agenda

- IBM i Access Client Solutions
- How to stay current with ACS the easy way
- SQL Performance Center
- Miscellaneous

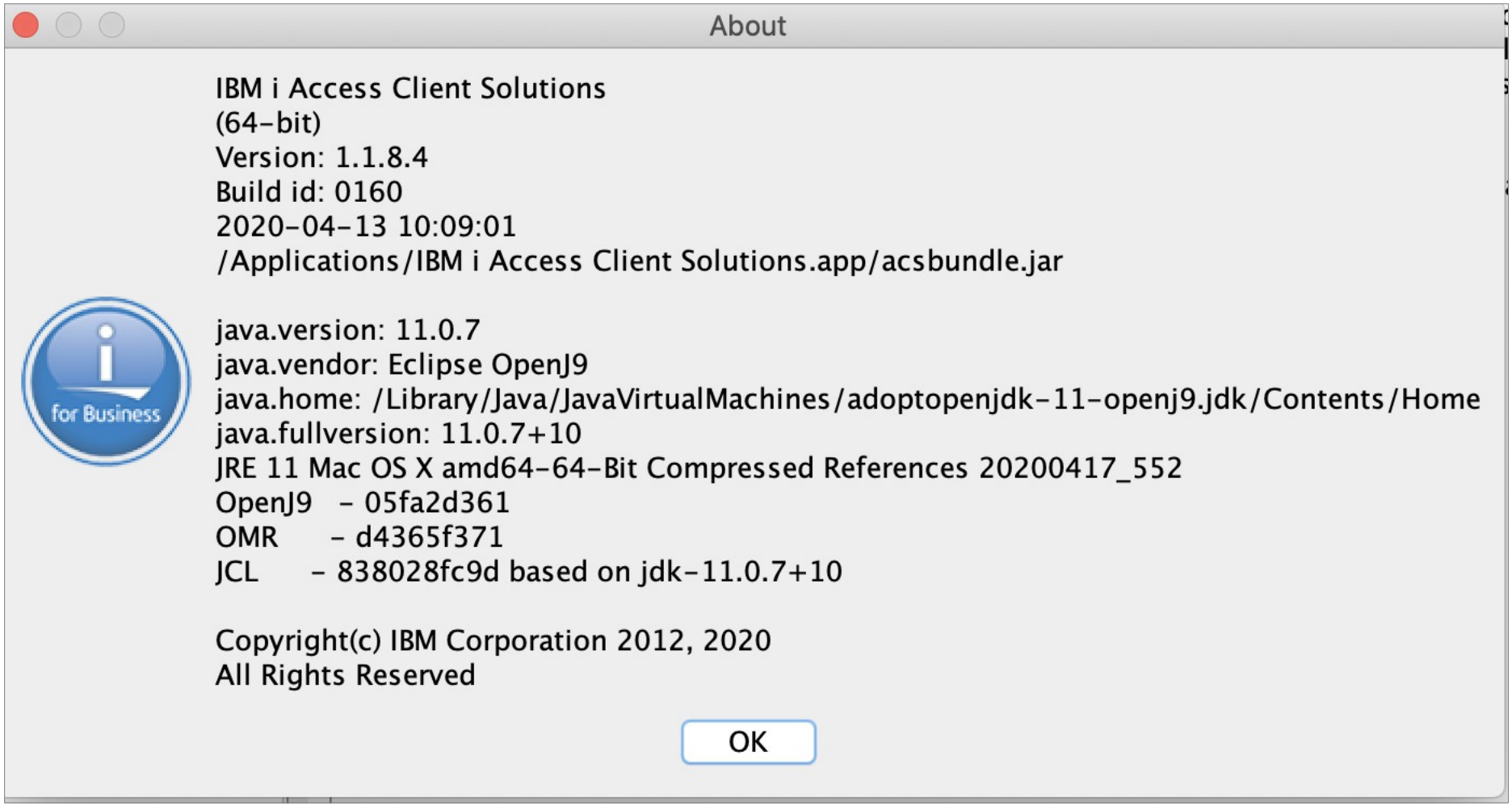
IBM i Access Client Solutions



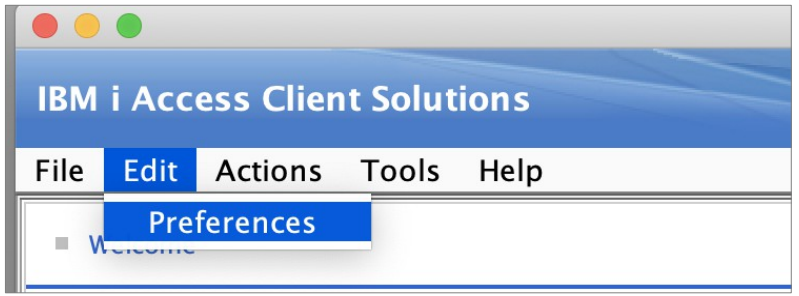
IBM i Access Client Solutions

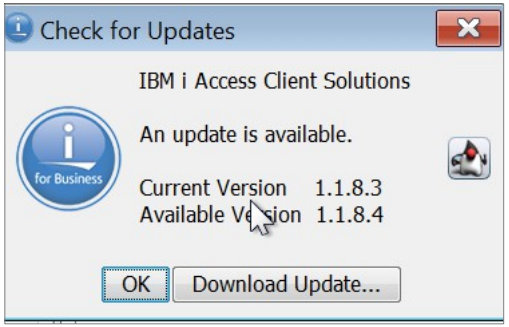


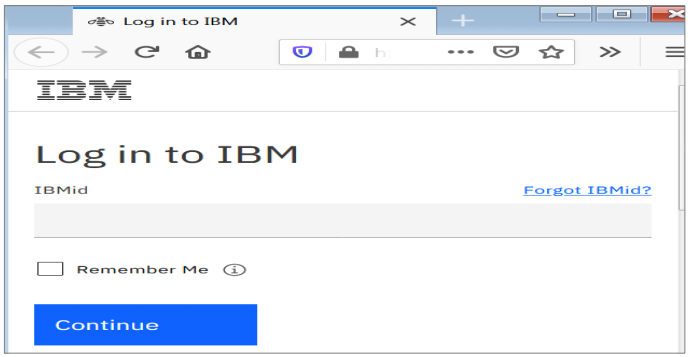
IBM i Access Client Solutions

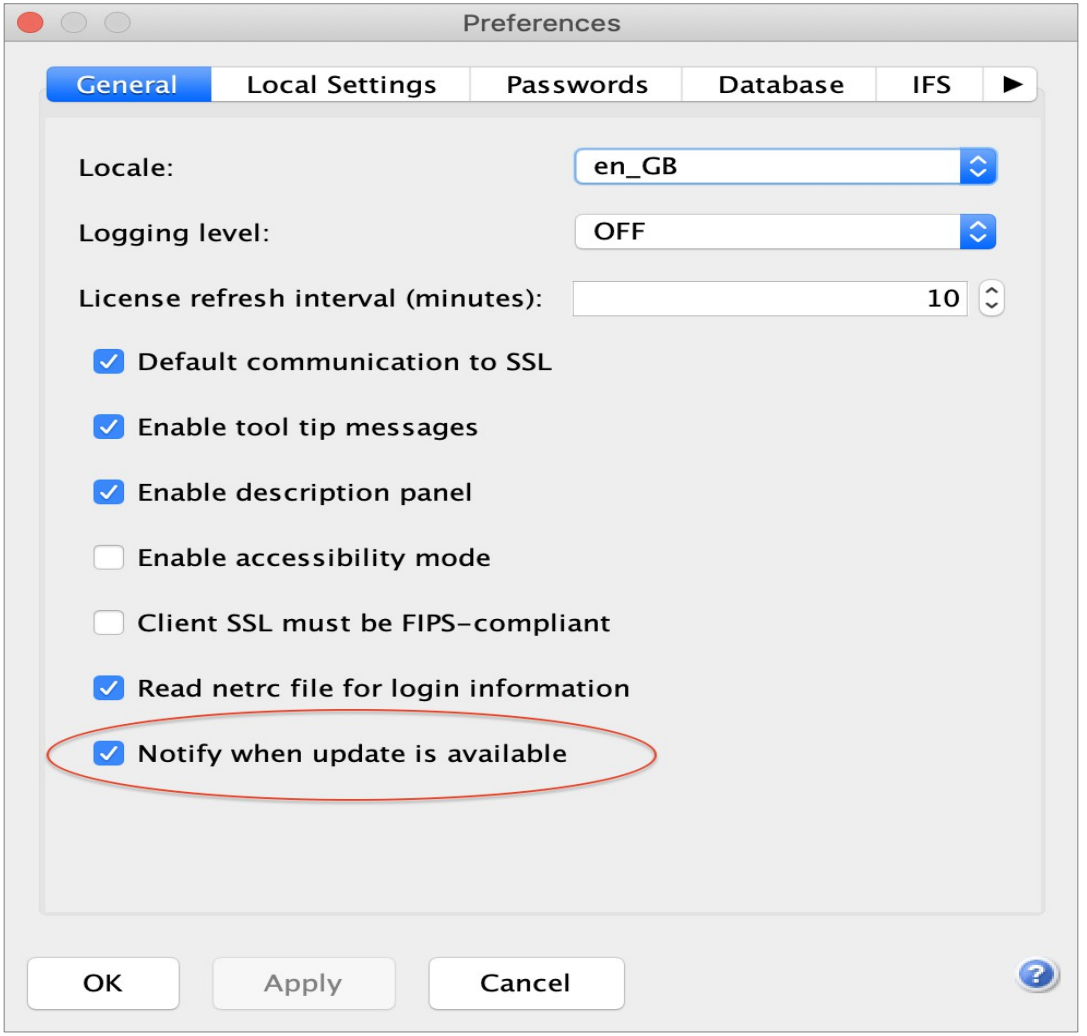


IBM i Access Client Solutions



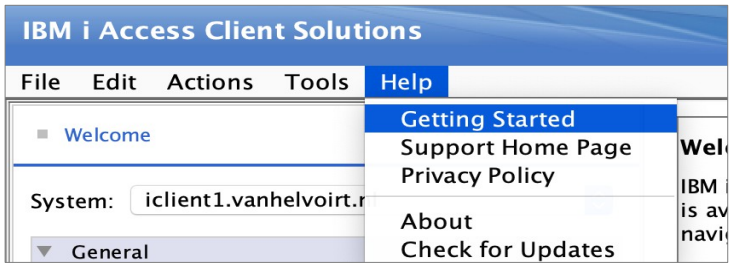






VAN HELVOIRT AUTOMATISERING

IBM i Access Client Solutions



5733XJ1 IBM i Access Client Solutions - GettingStarted

The content of this document was last updated on: April 8, 2020

0.5 QuickStartGuide

The QuickStartGuide contains instructions for deploying the product that will work for most users runn
single user, or for a multi-user PC, see the QuickStartGuide in the Documentation folder.

Additional details about the product, other deployment options and customizing the product are include

Contents:

IBM i Access Client Solutions

5.2 Setting up an IBM i Update Location

Administrators can use an IBM i system as their central location for doing installs and applying updates. Regardless of how you initially deploy the product to your users, you can use a central location on an IBM i so that your users can apply an update with a click of a button. Here are the steps:

- Decide what IBM i system will be used as the central location.
- Extract the latest product zip archive (IBMiAccess_v1r1.zip) to any place in the IBM i Integrated File System (IFS).
- Set the following properties in the AcsConfig.properties file:

```
com.ibm.iaccess.CheckUpdateSystem=system_name                    (see Note 1 below)
com.ibm.iaccess.CheckUpdatePath=/path_where_zip_was_extracted    (see Note 2 below)
```

Setting the com.ibm.iaccess.CheckUpdateSystem property causes Help->Check for Updates to behave differently. Instead of checking an external web location for the availability of an updated version, the above IBM i location will be checked for updated product files. An update will be detected when the timestamp of any required product file changes. The timestamp will be displayed on the Help->Check for Updates panel with an option to install the update.

The user will need valid credentials to the system specified for com.ibm.iaccess.CheckUpdateSystem. If their credentials have not already been cached from a previous connection to this system, they will be prompted to provide valid credentials. Failing to provide valid credentials will result in updates not being detected.

Users that have an IBM i Update Location configured and have selected the option "Notify when update is available" under Edit->Preferences, will not be prompted to provide credentials. They will only get notified of an available update if they have successfully connected to the com.ibm.iaccess.CheckUpdateSystem since the last time they logged into their PC.

Note:

1. This feature is only enabled when using version 1.1.8.3 or higher.
2. If com.ibm.iaccess.CheckUpdatePath is not set, the default location is assumed to be:
   /QIBM/ProdData/Access/ACS/Base
   Note: IBM now has release-specific PTFs that extract the contents of the product zip archive (IBMiAccess_v1r1.zip) to this location (see section 5.3). Administrators may extract IBMiAccess_v1r1.zip to this location themselves as long as they are aware that if the PTF is applied, this location is updated.
3. The AcsConfig.properties file along side acsbundle.jar gets propagated to the local PC during the initial deployment, but does not get updated during a product update. To leverage this feature, you will need to update AcsConfig.properties on the user's PC prior to selecting Help->Check for Updates.

Special note for administrators that maintain a customized AcsConfig.properties file within acsbundle.jar:
This feature downloads the acsbundle.jar that exists on the IBM i at the specified location. No special processing is done to maintain a customized version of the AcsConfig.properties file during an update. If you have previously customized the AcsConfig.properties file inside acsbundle.jar, you will need to make sure the new version of acsbundle.jar contains your desired customizations before making it available for download.

IBM i Access Client Solutions

5.3 Optional IBM i PTFs

For Administrators that want to maintain a centralized location on an IBM i for their users to install and update the product, release-specific PTFs are available that will provide the extracted contents of IBMiAccess_v1r1.zip at:

/QIBM/ProdData/Access/ACS/Base

The initial PTFs are:

```
V7R4M0  SI71900
V7R3M0  SI71934
V7R2M0  SI71935
```

These PTFs will be superseded for each product update and will normally be available within 2 weeks of the availability of the product update.

ACS PTF

Circumvention

PTFs Available

R720 SI73105 1000

R730 SI73104 1000

R740 SI73103 1000

Affected Modules

Modified date:
30 April 2020

Subscribe to this APAR

By subscribing, you receive periodic emails alerting you to the status of the

Contact and feedback

IBM i Access Client Solutions

SF99662 740 IBM HTTP Server for i - level 5

Preventive Service Planning

Abstract

SF99662: SF99662 740 IBM HTTP Server for i

Content

Preventive Service Planning -PSP

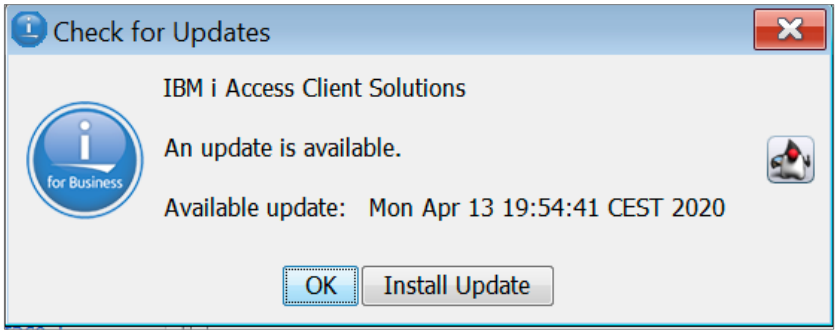
SF99662 740 IBM HTTP Server for i

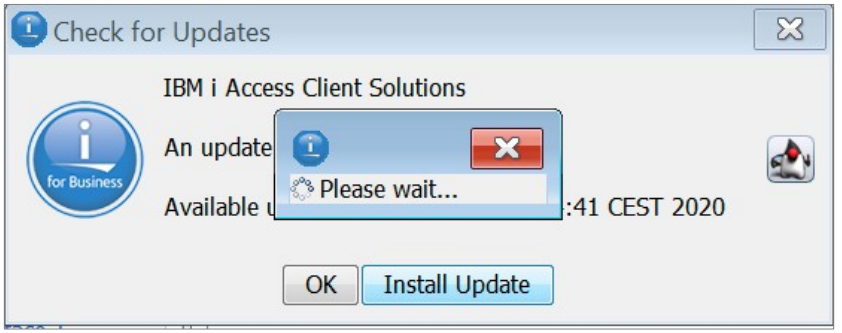
Release -- R740

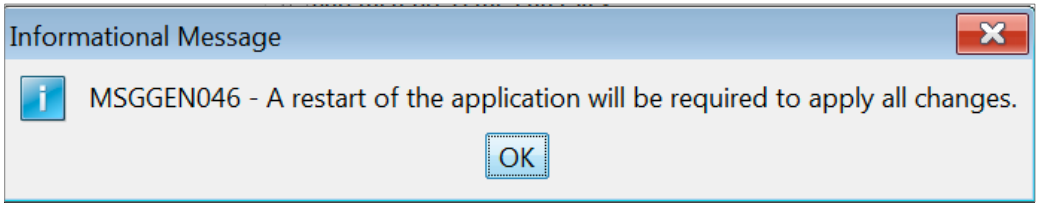
```
SF99662:  740 IBM HTTP Server for i
PTF Group Level:  5
Last Updated:  4/29/20
How to Display:  WRKPTFGRP SF99662
Description:  Description: IBM HTTP Server for i 7.4 PTF Group.
```

PTF NUMBER	DATE ADDED	APAR	LICENSED PROGRAM	CUMUL/ PACKA(
SI70669	04/29/20	SE71837	5770DG1	9304
SI73086	04/29/20	SE73244	5770DG1	1000
SI71936	04/29/20	SE72766	5770SS1	0121
SI72356	04/29/20	SE73103	5770SS1	0121
SI73087	04/29/20	SE73245	5770SS1	1000
SI73103	04/29/20	SE73495	5770SS1	1000
SI72594	04/11/20	SE73240	5770DG1	1000
SI72280	04/11/20	SE72750	5770DG1	1000
SI72301	04/11/20	SE73064	5770DG1	1000
SI71970	04/11/20	SE72790	5770DG1	0121
SI72517	04/11/20	SE73186	5770DG1	1000
SI72017	04/11/20	SE72720	5770SS1	1000
SI72016	04/11/20	SE72845	5770SS1	1000
SI72298	04/11/20	SE73063	5770SS1	1000
SI72070	04/11/20	SE72898	5770SS1	1000
SI72059	04/11/20	SE72885	5770SS1	1000
SI72347	04/11/20	SE73109	5770SS1	1000
SI71619	01/13/20	SE72428	5770DG1	0121
SI71907	01/13/20	SE72632	5770DG1	0121

IBM i Access Client Solutions







IBM i Access Client Solutions

When using the IBM i Update Location:

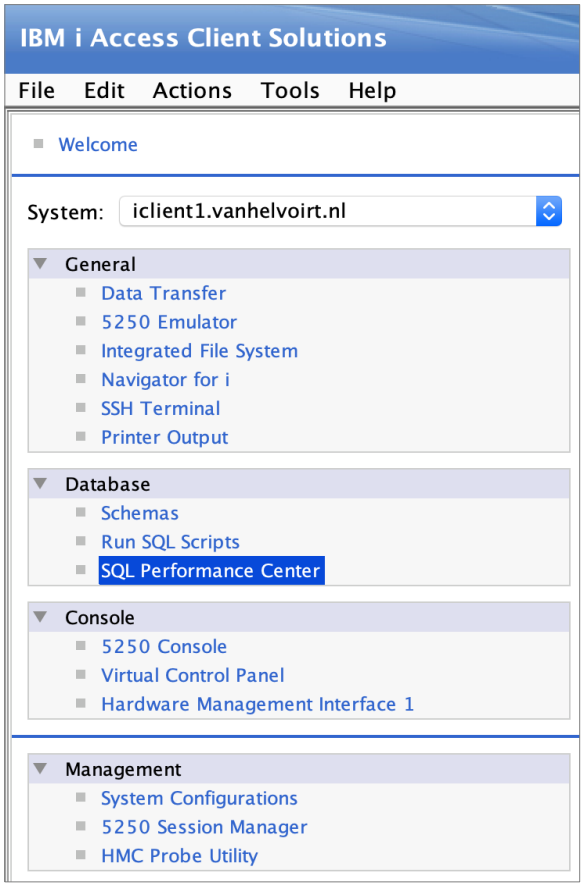
“com.ibm.iaccess.CheckUpdateSystem=system_name”

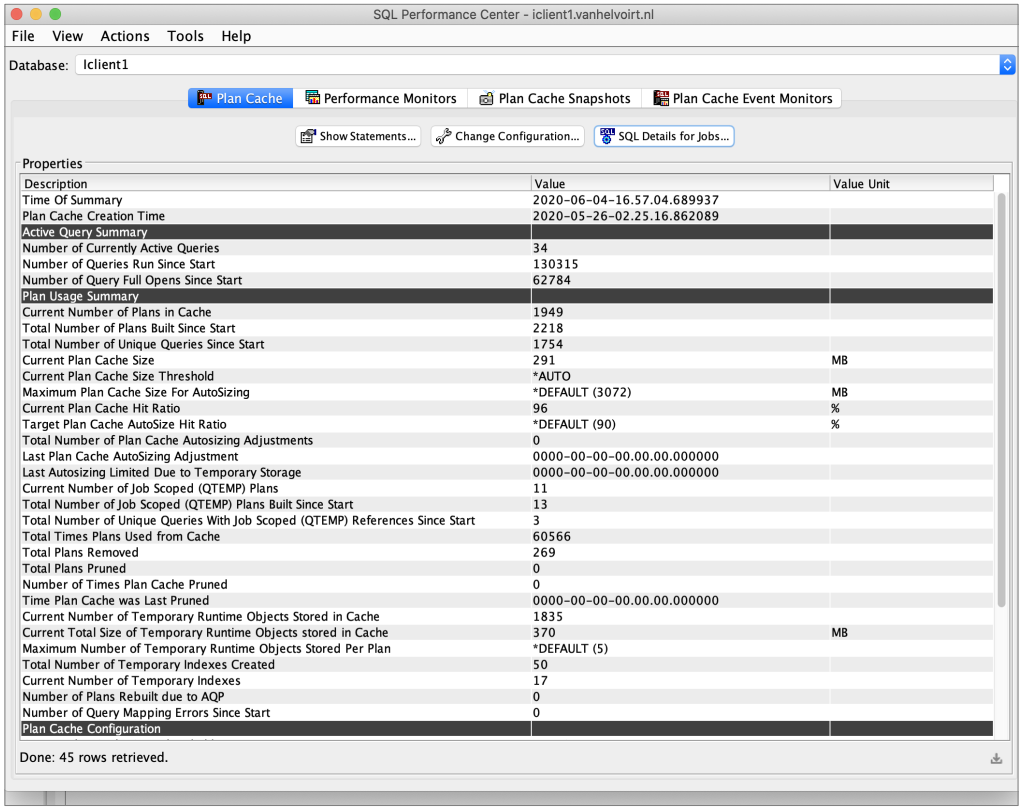
You need to be signed on when starting IBM i ACS

or

You need to use the option “Check for Updates”
(which will ask you to signon)

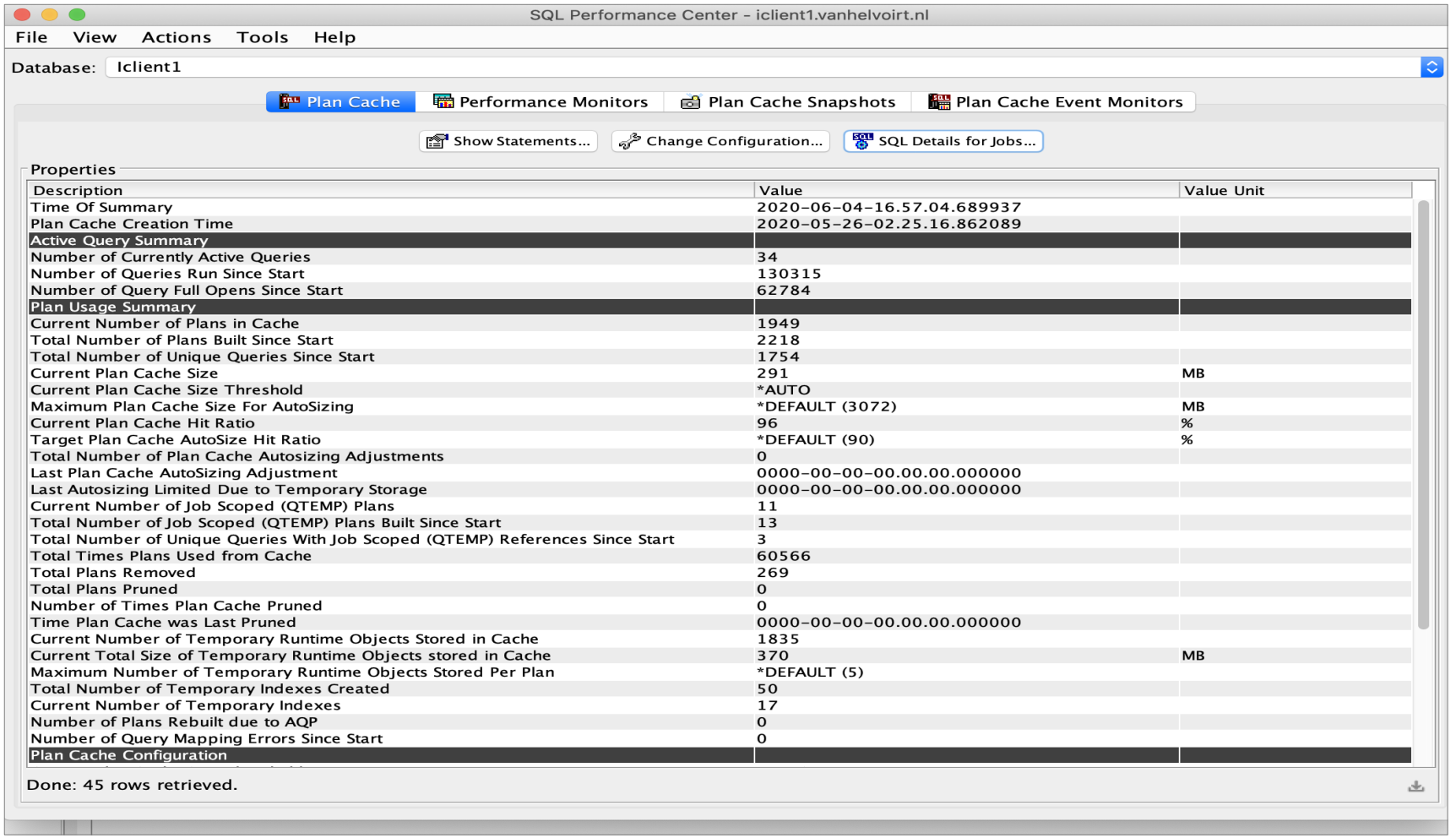
SQL Performance Center



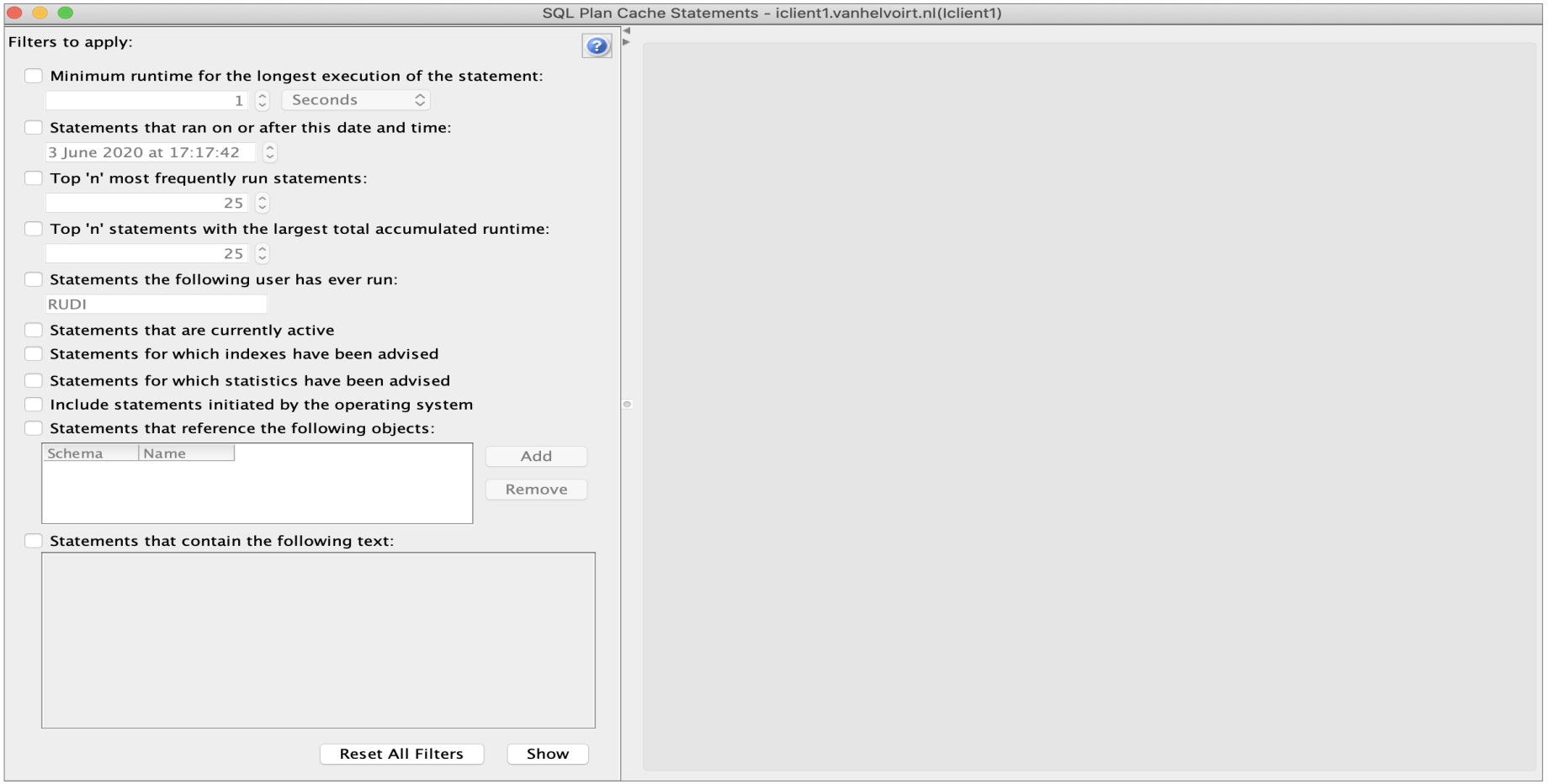


VAN HELVOIRT AUTOMATISERING

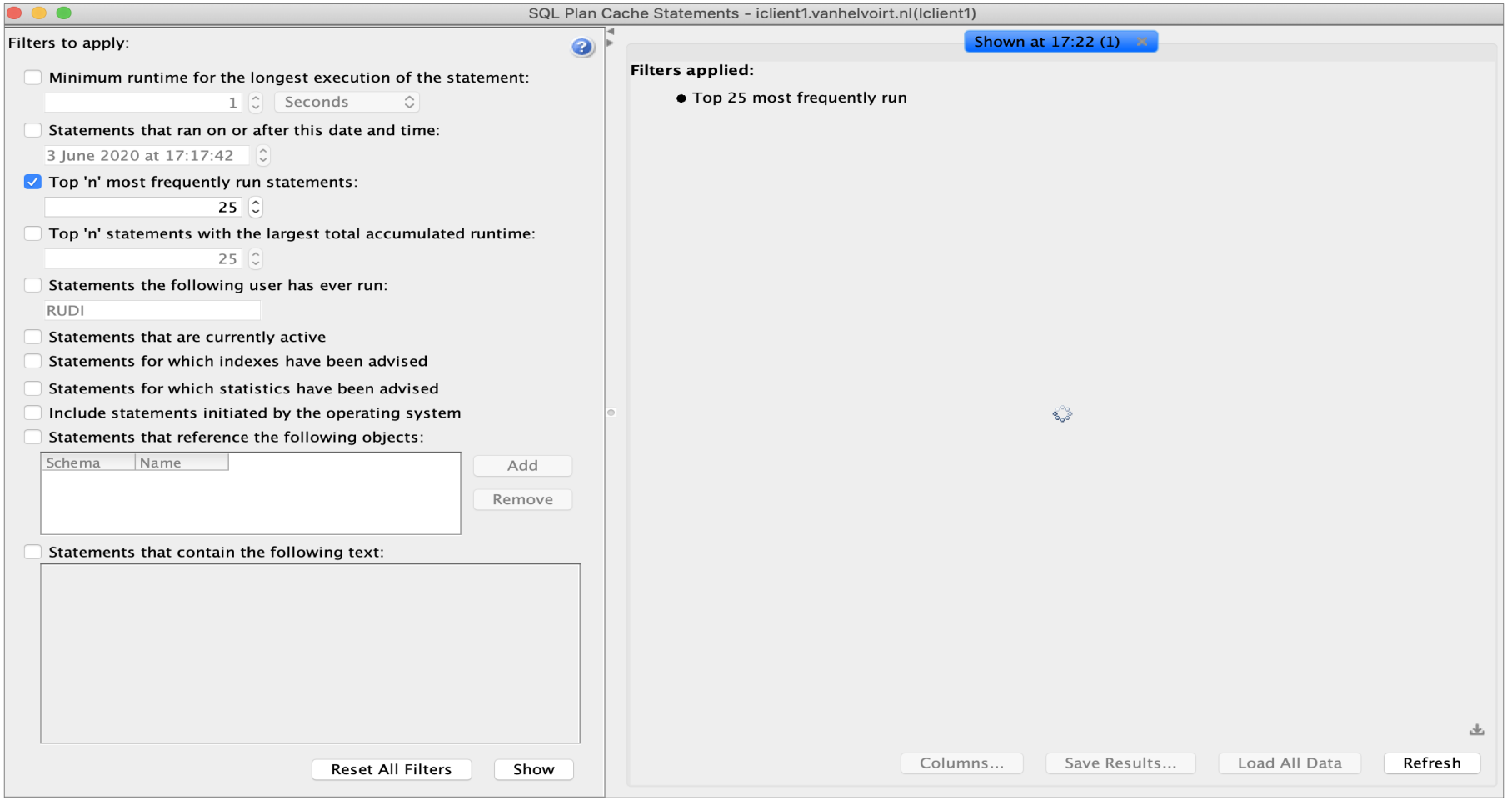
SQL Performance Center



Plan Cache – Show Statements...

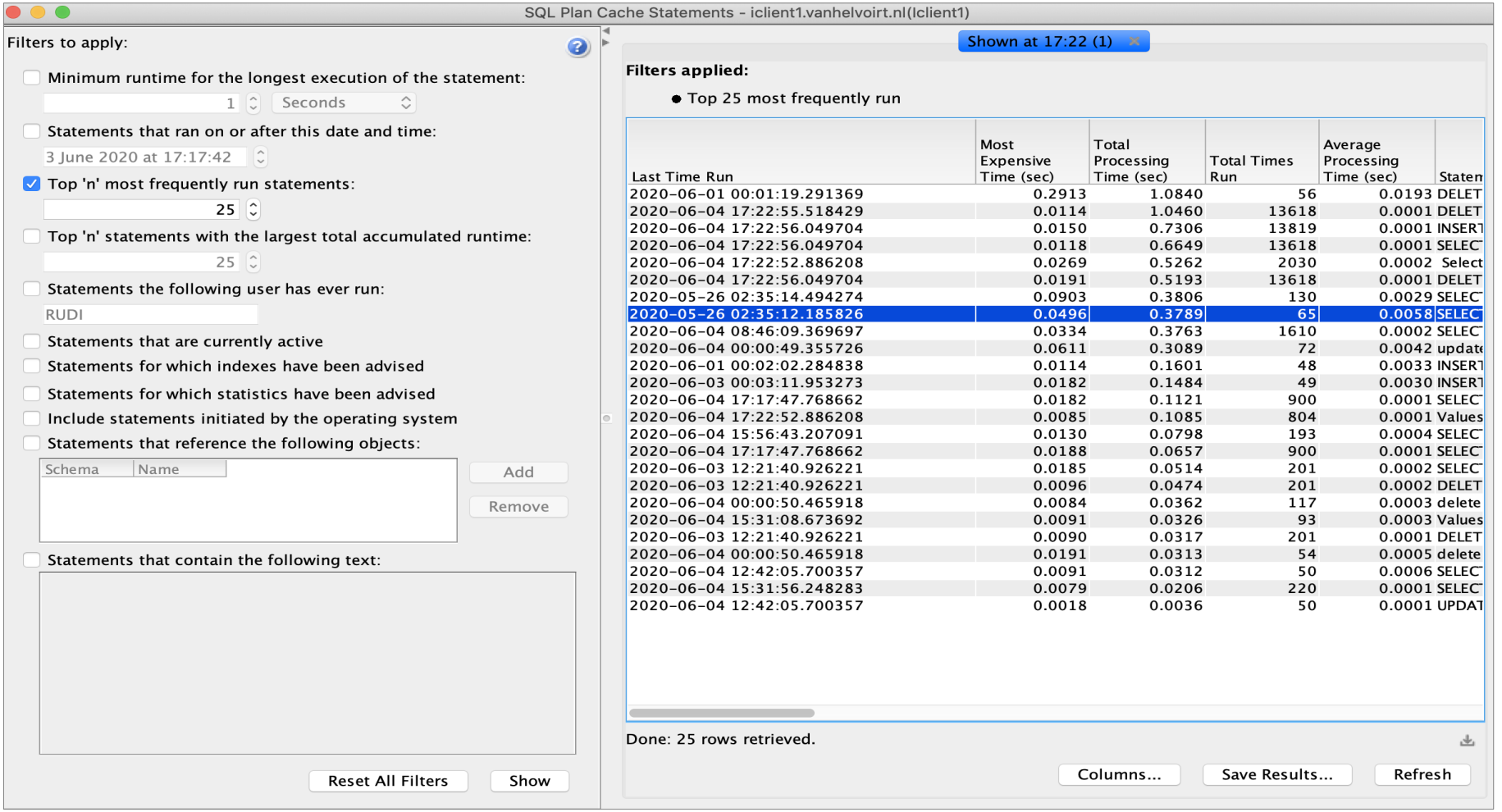


Plan Cache – Show Statements...

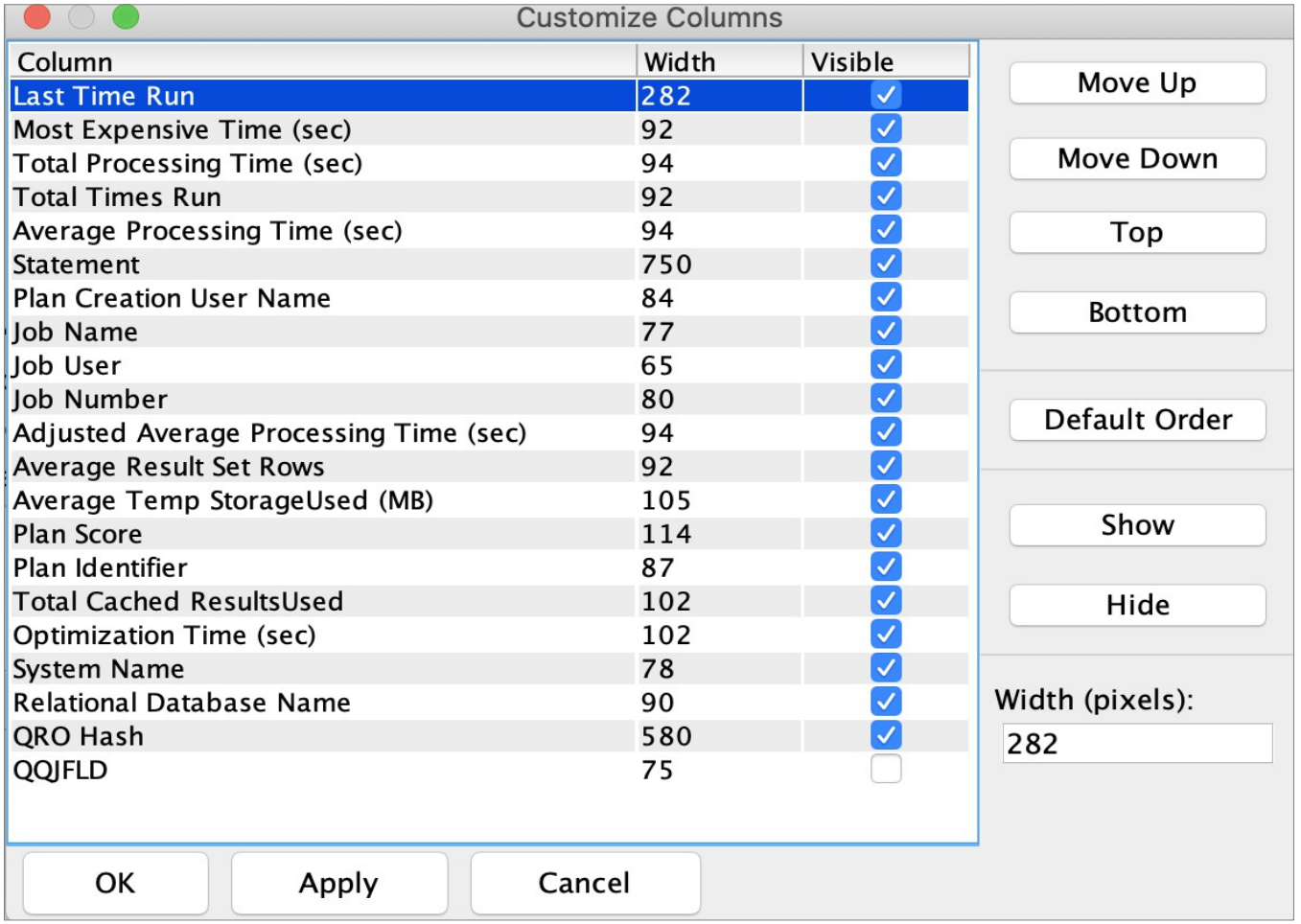


VAN HELVOIRT AUTOMATISERING

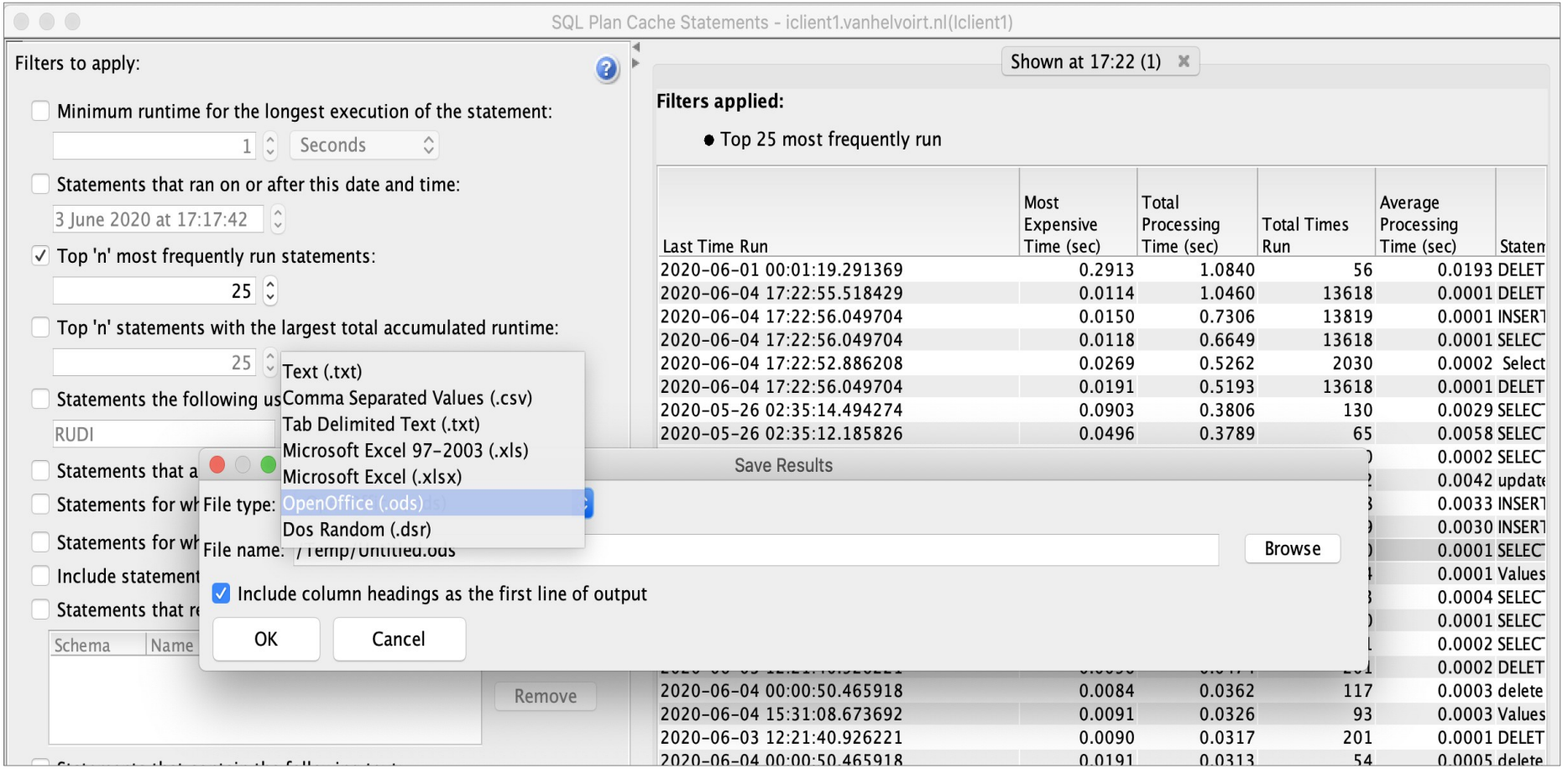
Plan Cache – Show Statements...



Plan Cache – Show Statements...

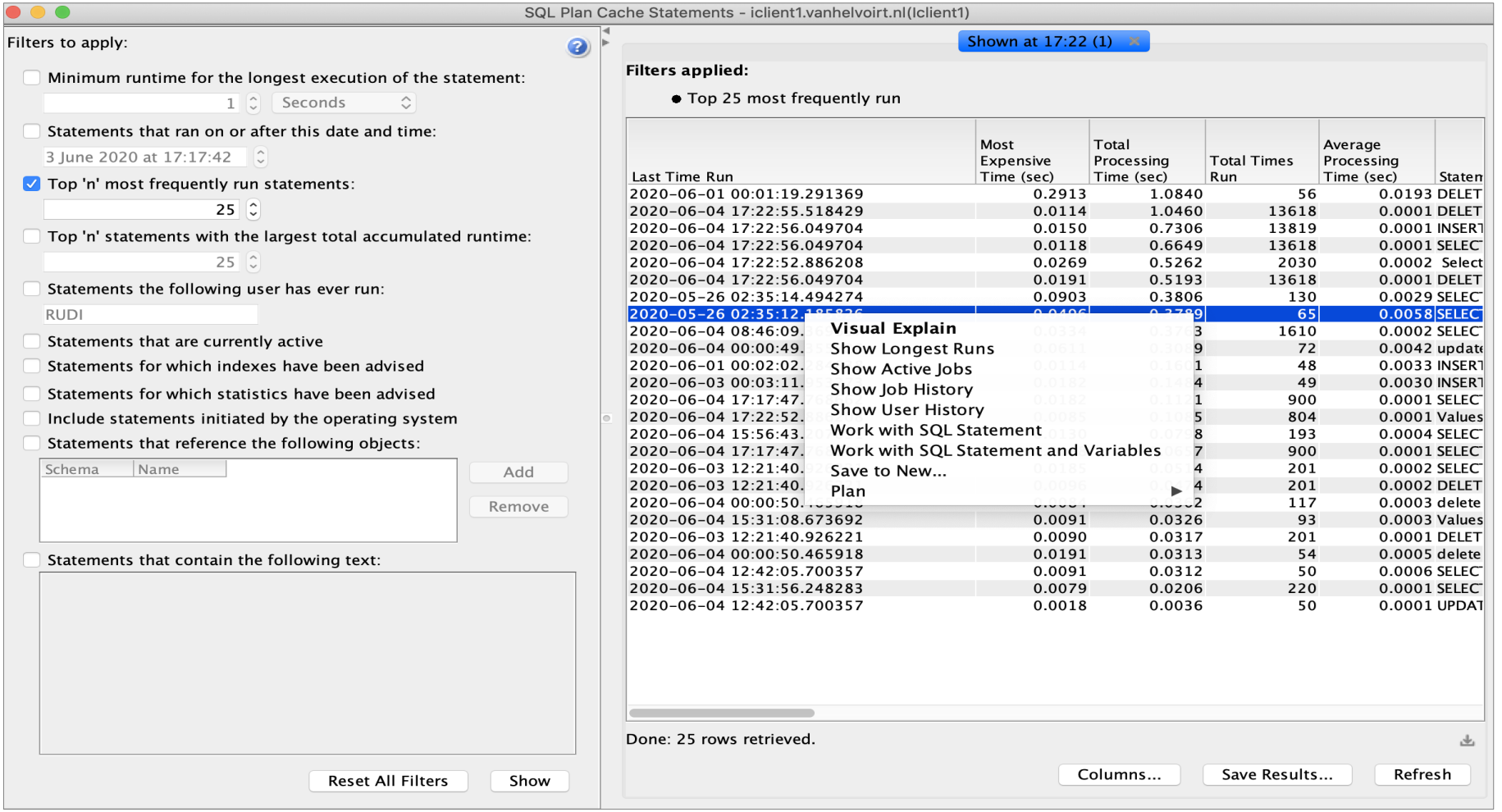


Plan Cache – Show Statements...

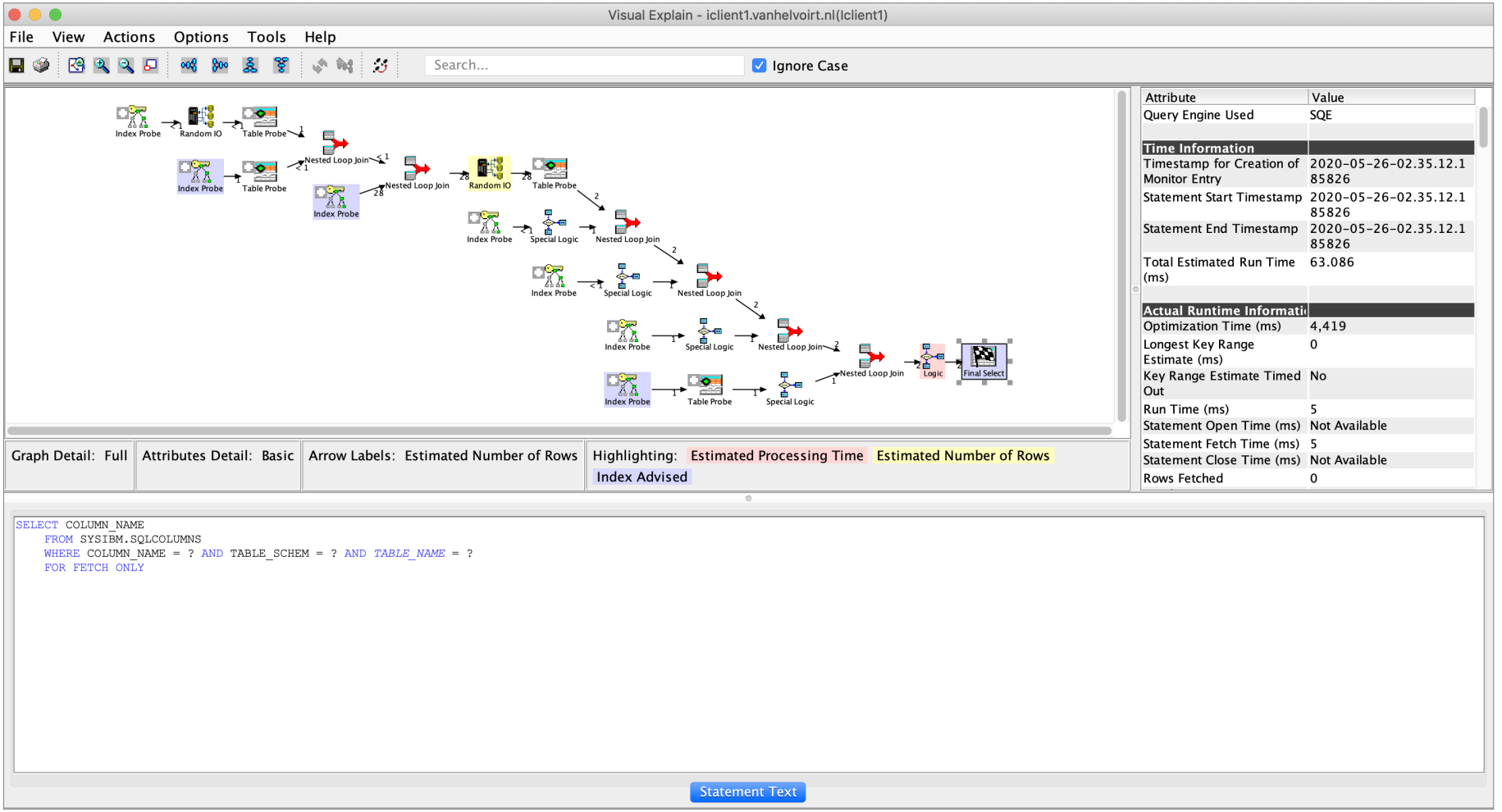


VAN HELVOIRT AUTOMATISERING

Plan Cache – Show Statements...

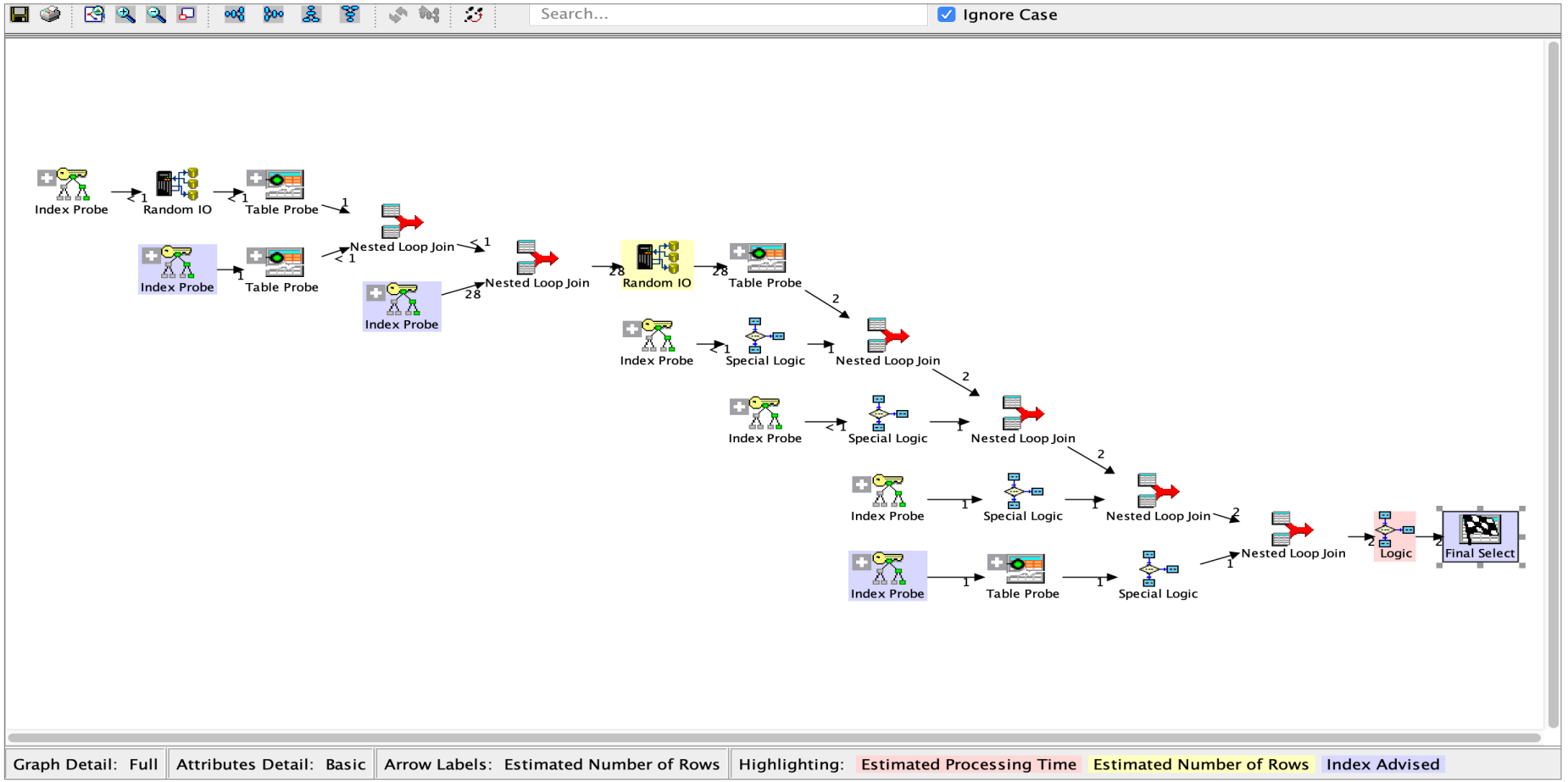


Plan Cache – Show Statements...



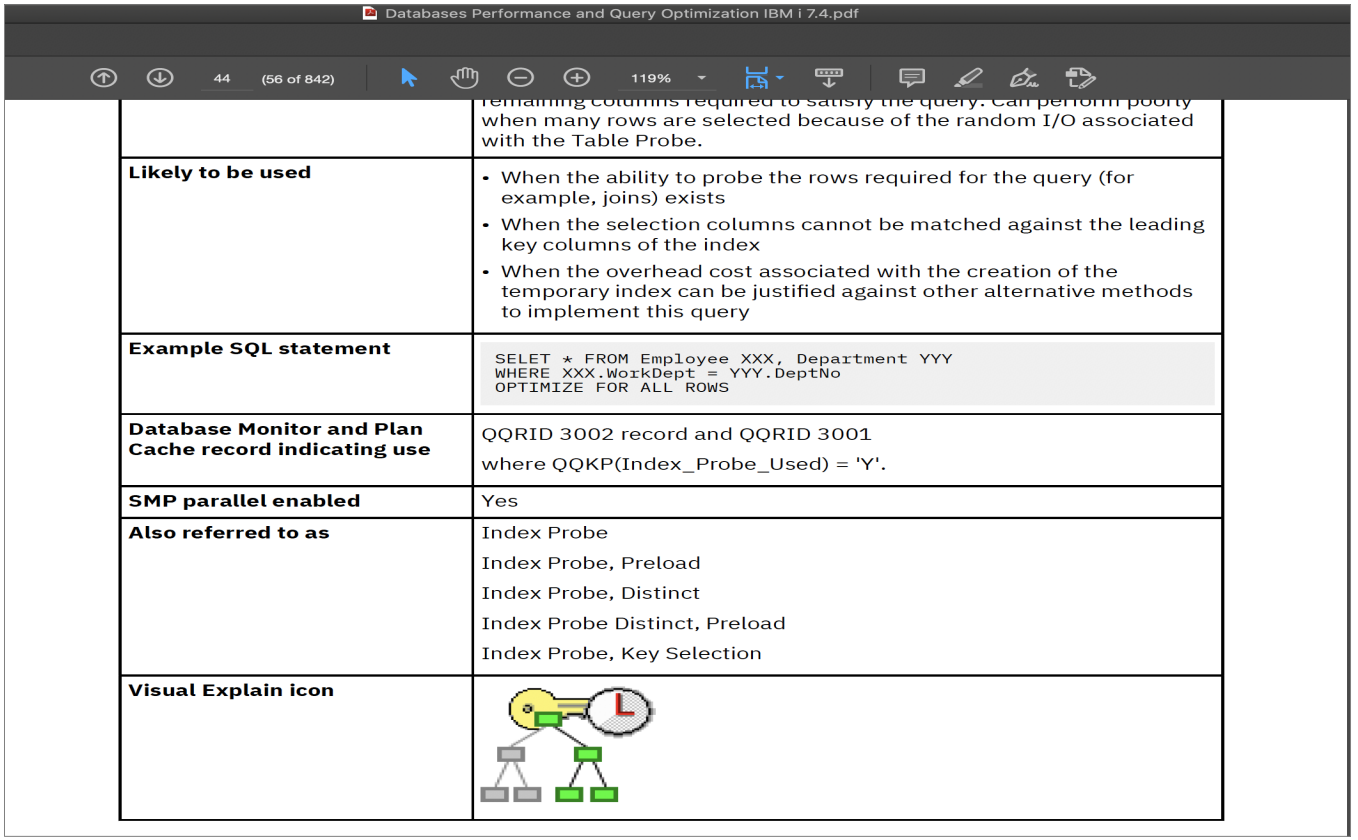
VAN HELVOIRT AUTOMATISERING

Plan Cache – Show Statements...



Plan Cache – Show Statements...

IBM i 7.4 Database Performance and Query Optimization



	when many rows are selected because of the random I/O associated with the Table Probe.
Likely to be used	• When the ability to probe the rows required for the query (for example, joins) exists • When the selection columns cannot be matched against the leading key columns of the index • When the overhead cost associated with the creation of the temporary index can be justified against other alternative methods to implement this query

Example SQL statement	`SELET * FROM Employee XXX, Department YYY`
	`WHERE XXX.WorkDept = YYY.DeptNo`
	`OPTIMIZE FOR ALL ROWS`

Database Monitor and Plan Cache record indicating use	QQRID 3002 record and QQRID 3001 where QQKP(Index_Probe_Used) = 'Y'.
SMP parallel enabled	Yes
Also referred to as	Index Probe Index Probe, Preload Index Probe, Distinct Index Probe Distinct, Preload Index Probe, Key Selection
Visual Explain icon	

Plan Cache – Show Statements...

SMP parallel enabled Yes

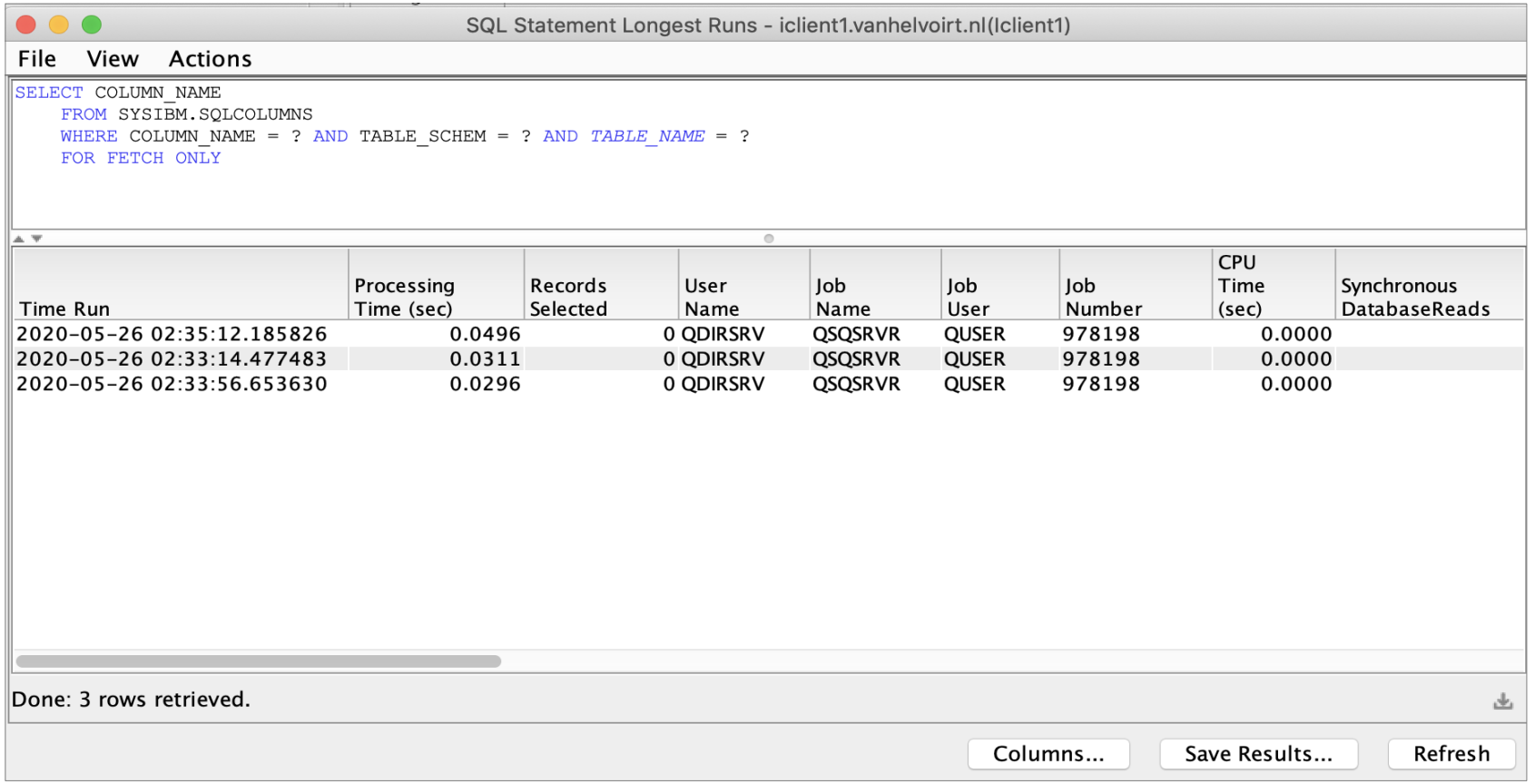
License Program:

5770-SS1 DB2 Symmetric MultiProcessing

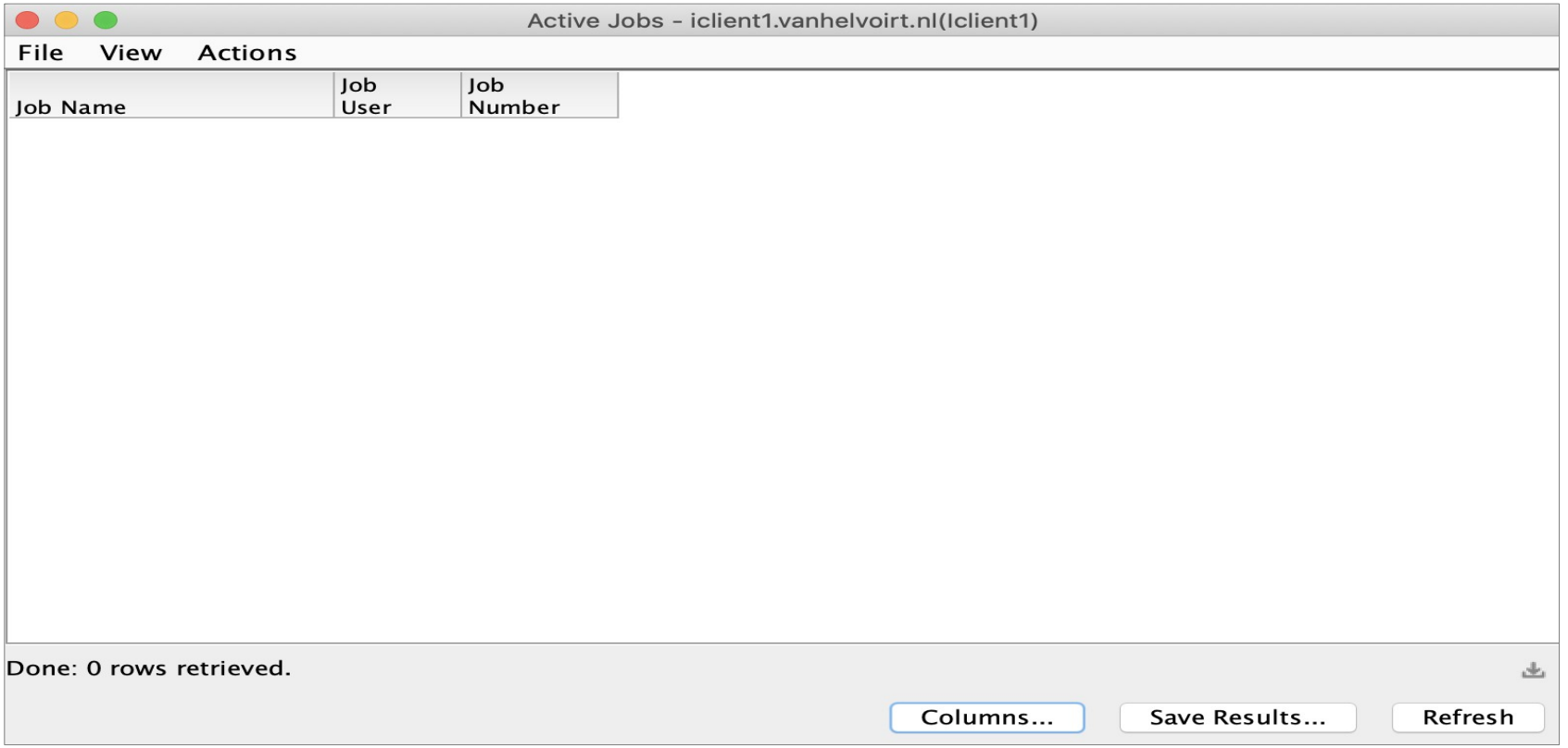
Introduction to DB2 Symmetric Multiprocessing for IBM i

Process your DB2 for i indexes in parallel

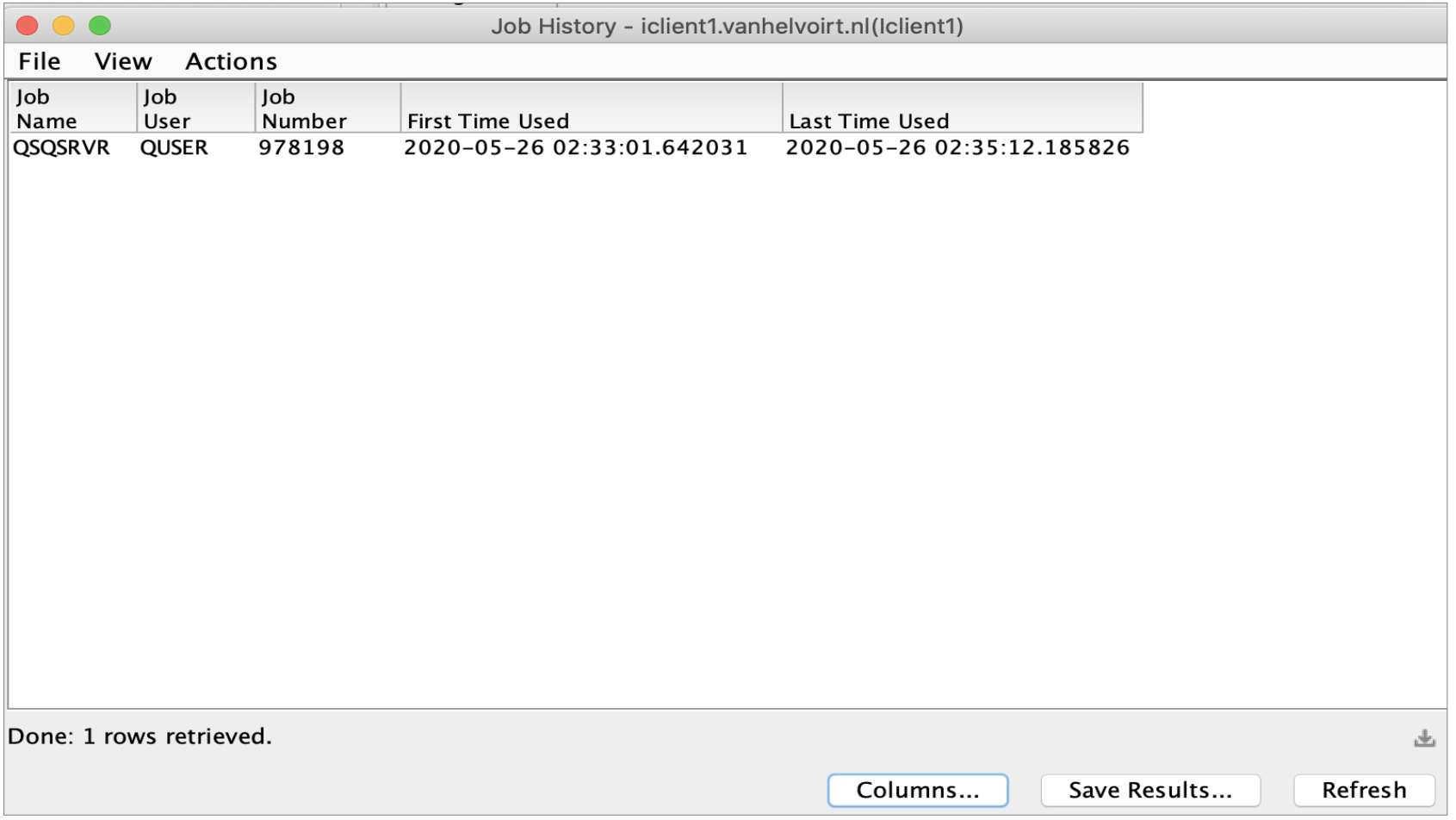
Plan Cache – Show Statements...



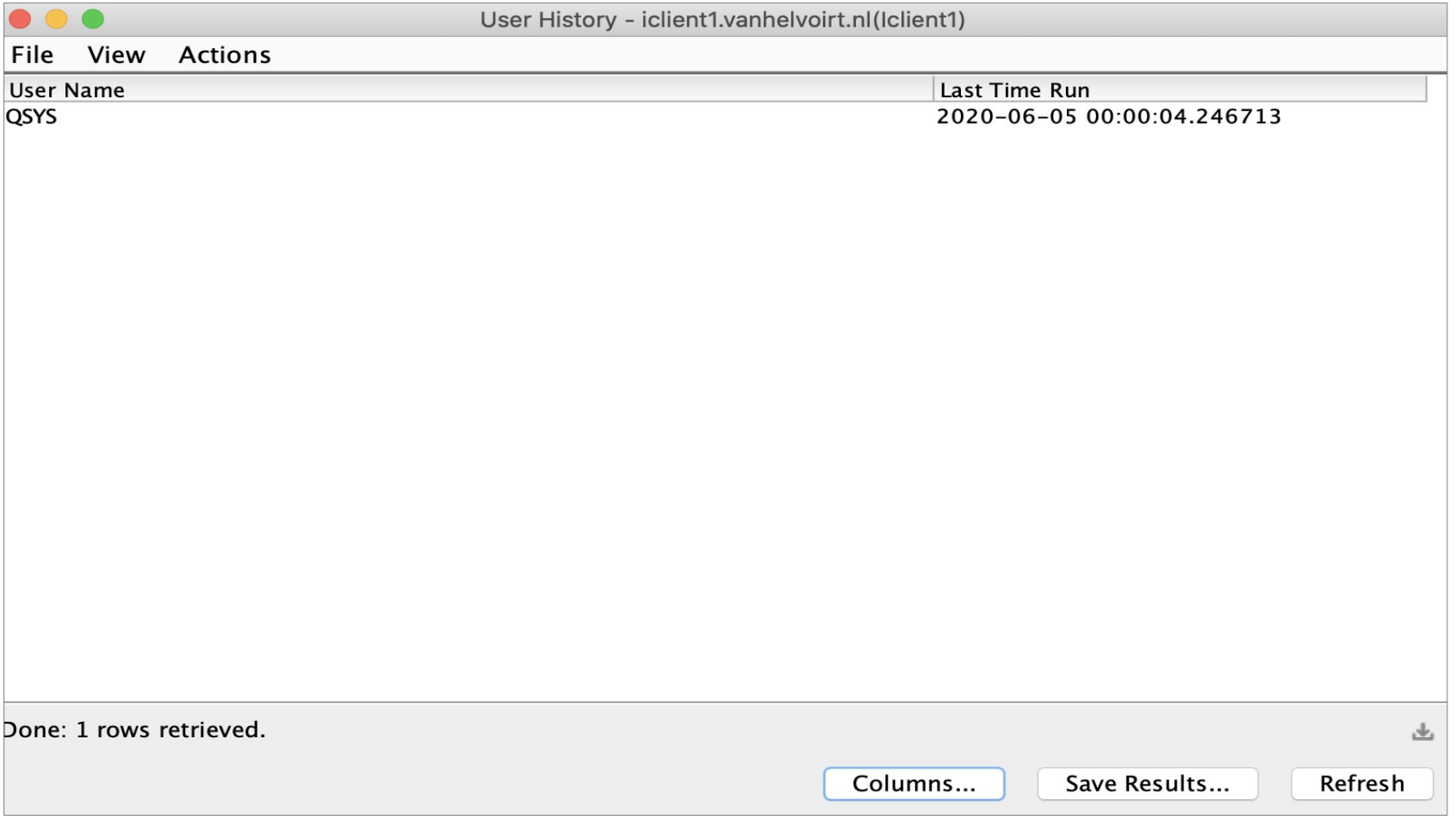
Plan Cache – Show Statements...



Plan Cache – Show Statements...

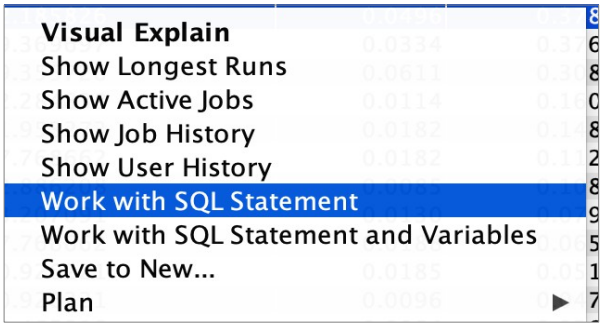


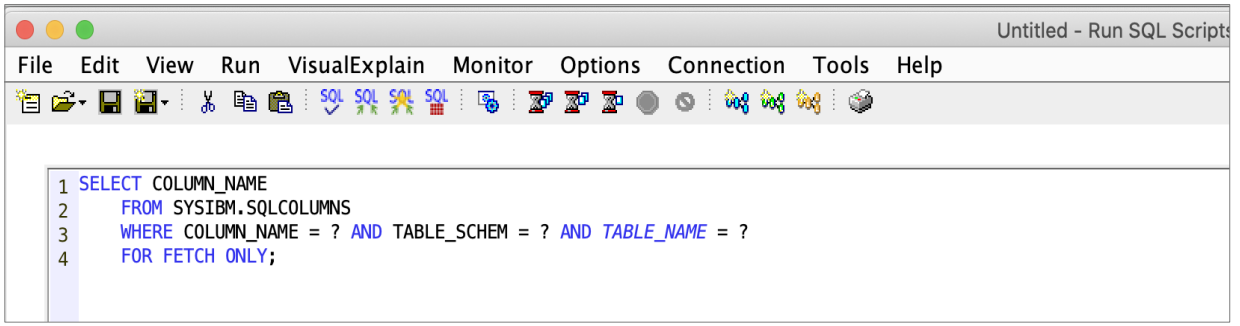
Plan Cache – Show Statements...

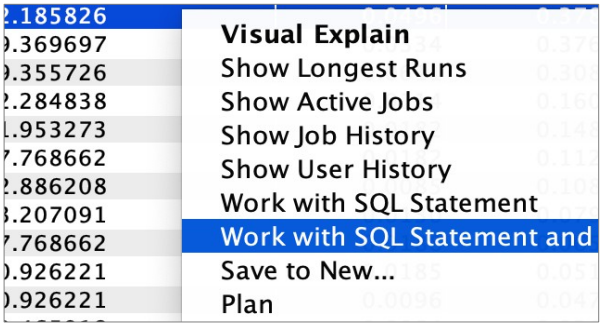


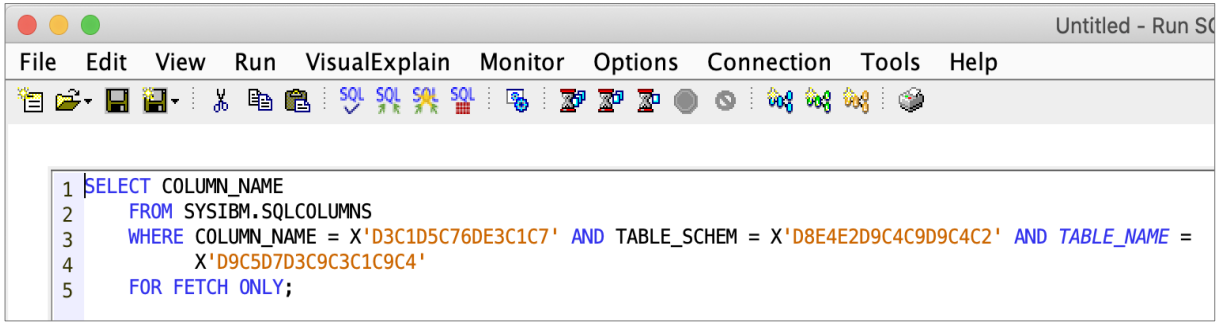
VAN HELVOIRT AUTOMATISERING

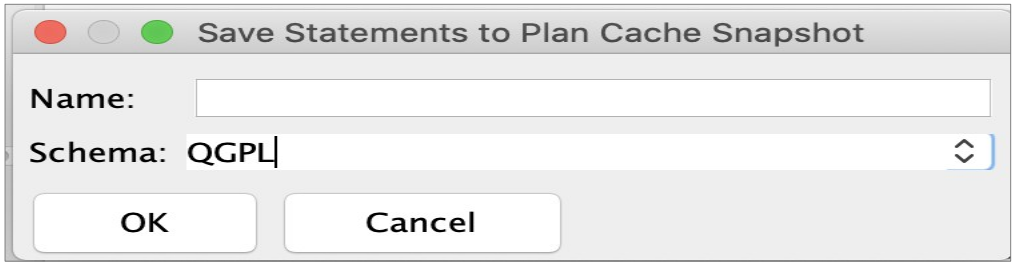
Plan Cache – Show Statements...



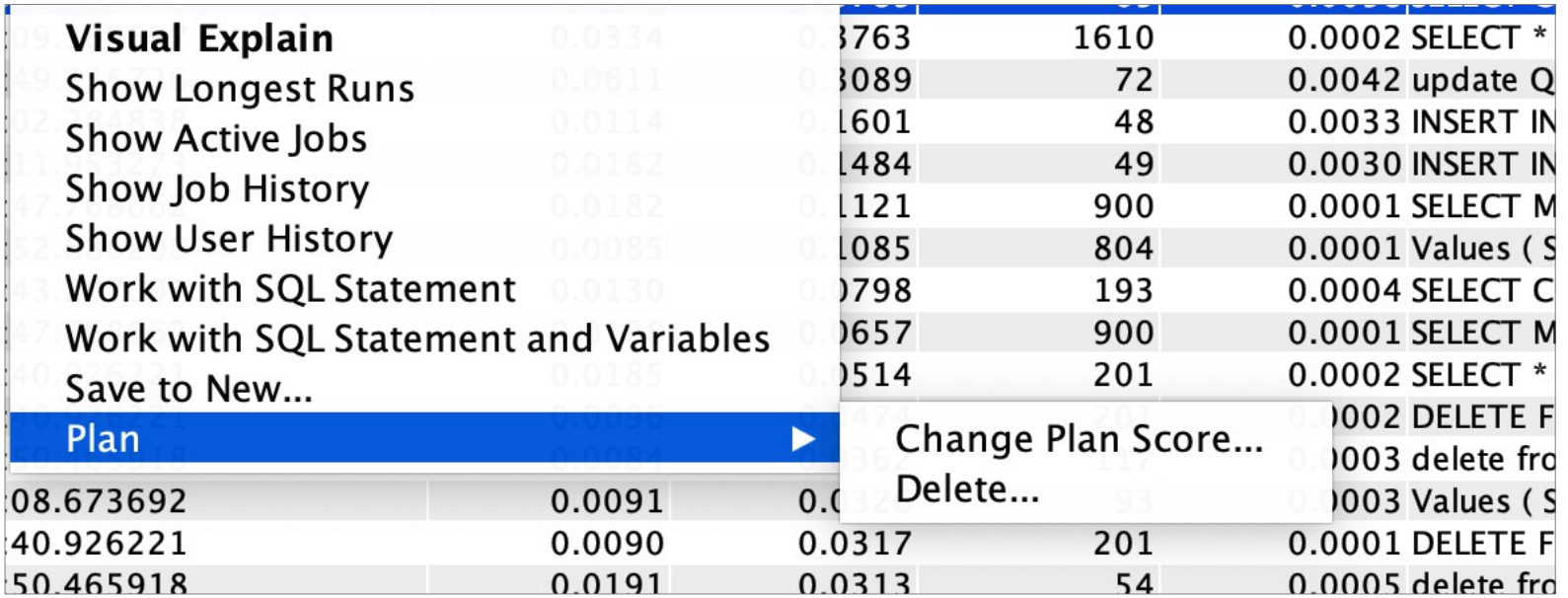


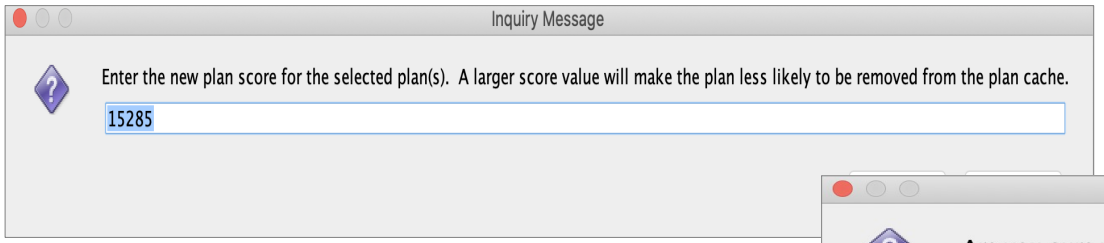


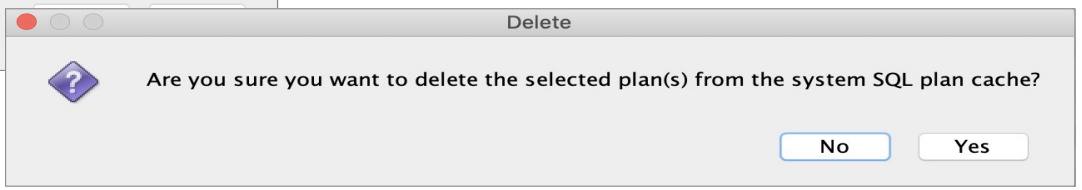




Plan Cache – Show Statements...

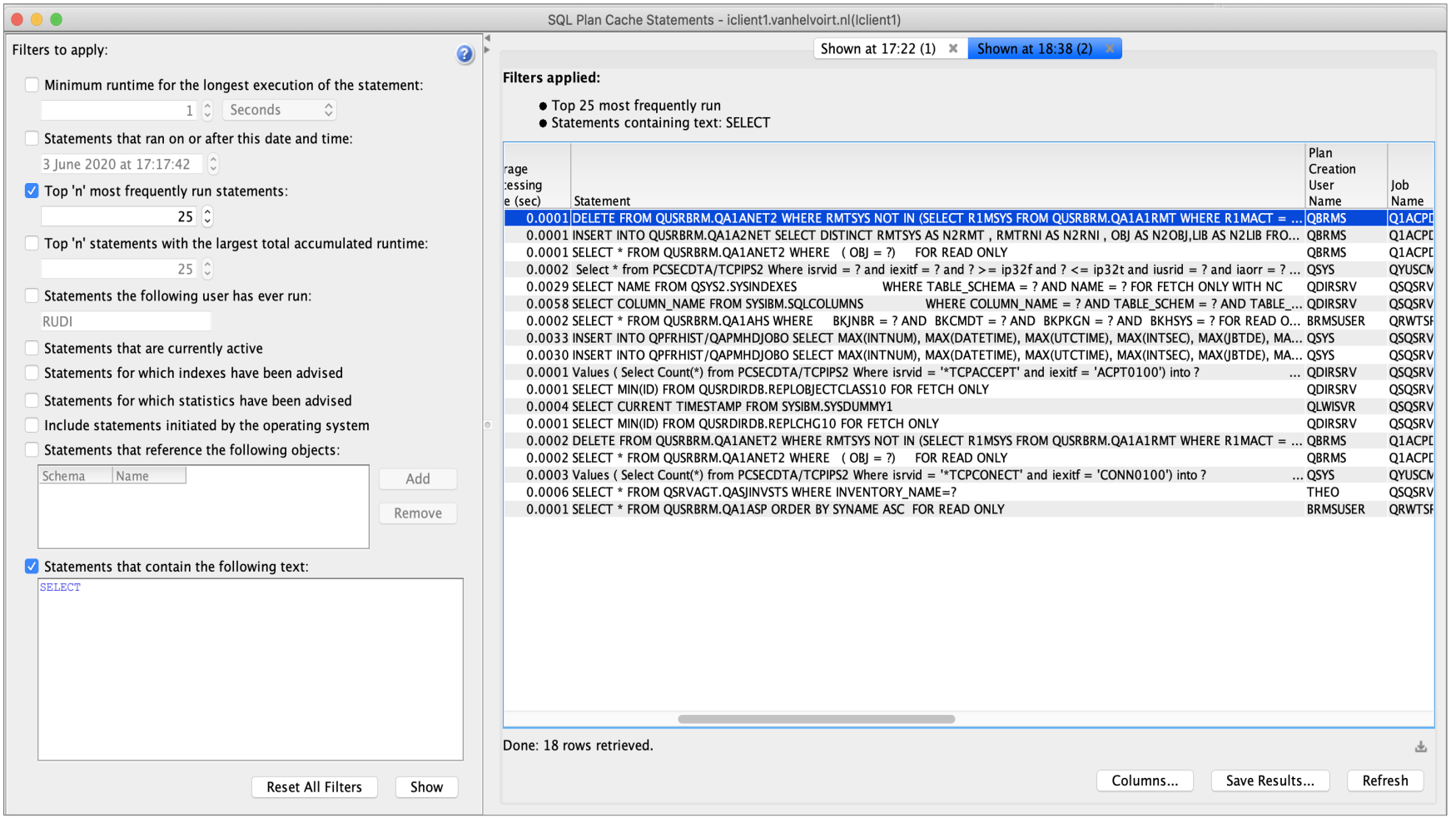






VAN HELVOIRT AUTOMATISERING

Plan Cache – Show Statements...



Plan Cache – Autosizing

Plan Cache | Performance Monitors | Plan Cache Snapshots | Plan Cache Event Monitors

Show Statements... | Change Configuration... | SQL Details for Jobs...

Properties

Description	Value	Value Unit
Time Of Summary	2020-06-04-17.44.43.906932	
Plan Cache Creation Time	2020-03-21-14.51.03.324595	
Active Query Summary		
Number of Currently Active Queries	17294	
Number of Queries Run Since Start	659742527	
Number of Query Full Opens Since Start	9095712	
Plan Usage Summary		
Current Number of Plans in Cache	5984	
Total Number of Plans Built Since Start	708121	
Total Number of Unique Queries Since Start	181490	
Current Plan Cache Size	508	MB
Current Plan Cache Size Threshold	*AUTO	
Maximum Plan Cache Size For AutoSizing	*DEFAULT (9216)	MB
Current Plan Cache Hit Ratio	92	%
Target Plan Cache AutoSize Hit Ratio	*DEFAULT (90)	%
Total Number of Plan Cache Autosizing Adjustments	0	
Last Plan Cache AutoSizing Adjustment	2020-03-22-21.19.00.118709	
Last Autosizing Limited Due to Temporary Storage	2020-06-04-17.24.11.919093	
Current Number of Job Scoped (QTEMP) Plans	3	
Total Number of Job Scoped (QTEMP) Plans Built Since Start	16895	
Total Number of Unique Queries With Job Scoped (QTEMP) References Since Start	49	
Total Times Plans Used from Cache	8387724	
Total Plans Removed	48744	
Total Plans Pruned	653441	
Number of Times Plan Cache Pruned	2551	
Time Plan Cache was Last Pruned	2020-06-04-17.24.11.919080	
Current Number of Temporary Runtime Objects Stored in Cache	8294	
Current Total Size of Temporary Runtime Objects stored in Cache	282	MB
Maximum Number of Temporary Runtime Objects Stored Per Plan	*DEFAULT (5)	
Total Number of Temporary Indexes Created	1222	
Current Number of Temporary Indexes	79	
Number of Plans Rebuilt due to AQP	15	
Number of Query Mapping Errors Since Start	12	

Plan Cache – Autosizing

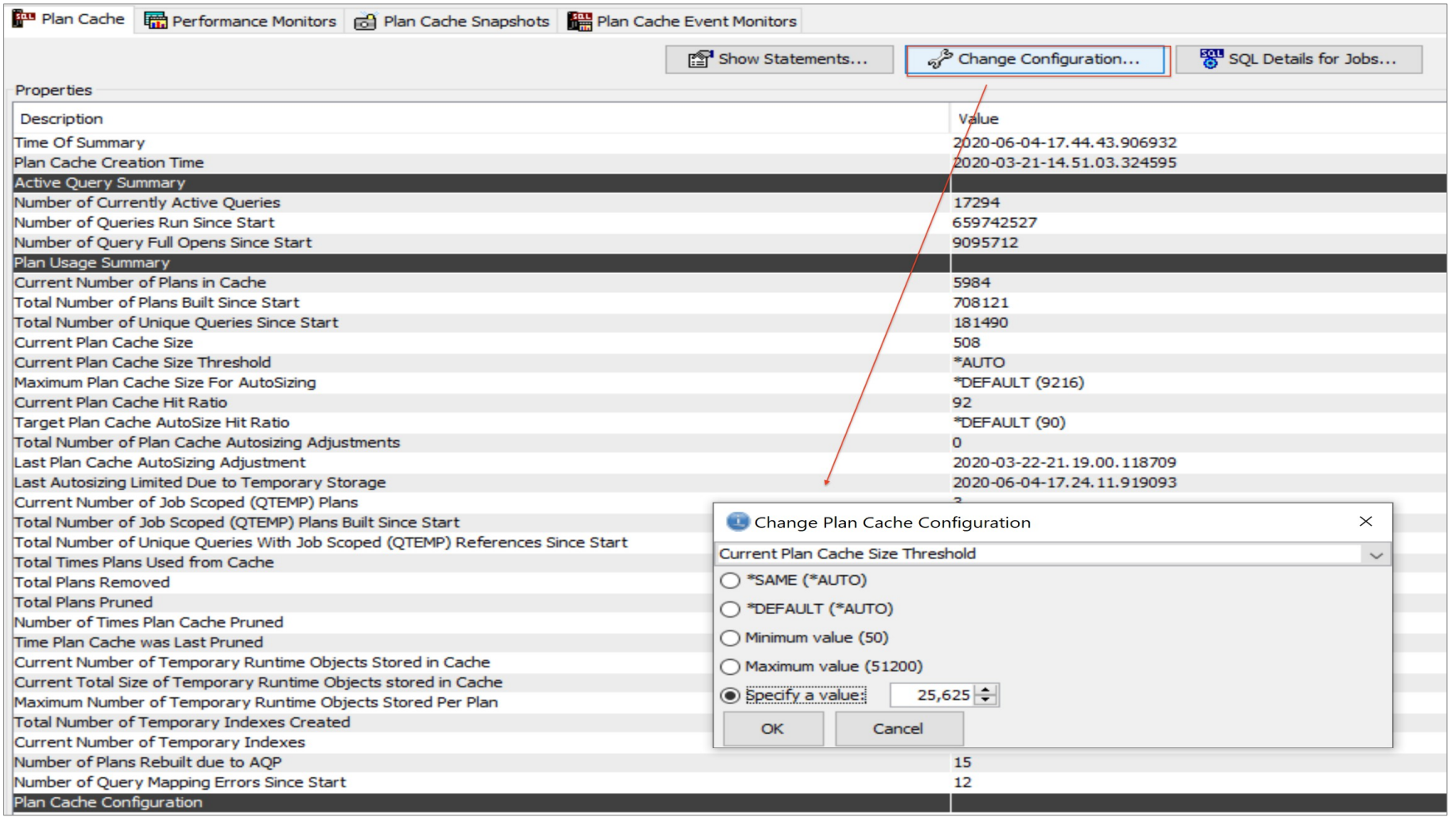
Each plan cache entry may also have query runtime objects associated with it. These runtime objects are the real executable objects and temporary storage containers (hash tables, sorts, temporary indexes, and so on) used to run the query. Although these objects are not included the in the plan cache size calculation, they may indirectly affect and be affected by the plan cache size. This is because, in addition to honoring the determined plan cache size, the system also seeks to keep the total temporary storage usage for inactive, cached plans within an internally determined threshold. Unlike the plan cache size calculation, this temporary storage calculation considers both the cached plans and the associated runtime objects. If this temporary storage calculation exceeds a system determined percentage of the system auxiliary storage pool (ASP) or if the system storage lower limit (defined by QSTGLOWLMT) is surpassed, the system considers the plan cache to be using excessive temporary storage.

The 7 % rule

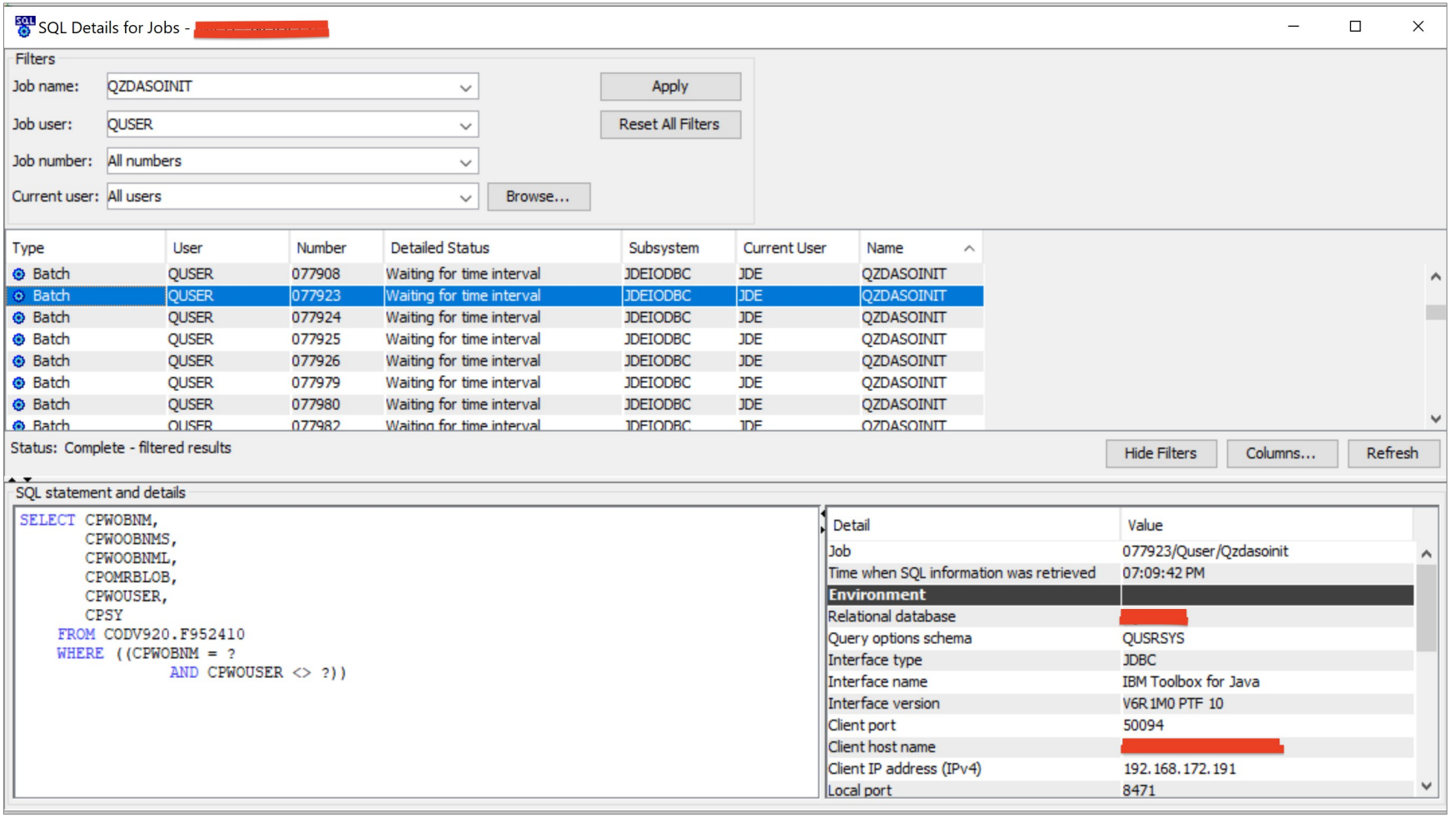
```
1 -- category: Systems Management
2 -- description: How close are we to hitting the 7% rule?
3 SELECT
4     dec(
5         DECFLOAT(current_temporary_storage, 34) / (100 - DECFLOAT(system_asp_used, 34)) / DECFLOAT(system_asp_storage, 34) * 10000,
6         3, 1) AS CURRENT_TEMPORARY_PERCENTAGE_USED
7     FROM qsys2.system_status_info;
```

CURRENT_TEMPORARY_PERCENTAGE_USED
15.2

Plan Cache – Change Configuration...



Plan Cache – SQL Details for Jobs...



Performance Monitors

Monitoring your queries using the Database Monitor

Start Database Monitor (STRDBMON) command gathers information about a query in real time and stores this information in an output table. This information can help you determine whether your system and your queries are performing well, or whether they need fine-tuning. Database monitors can generate significant CPU and disk storage overhead when in use.

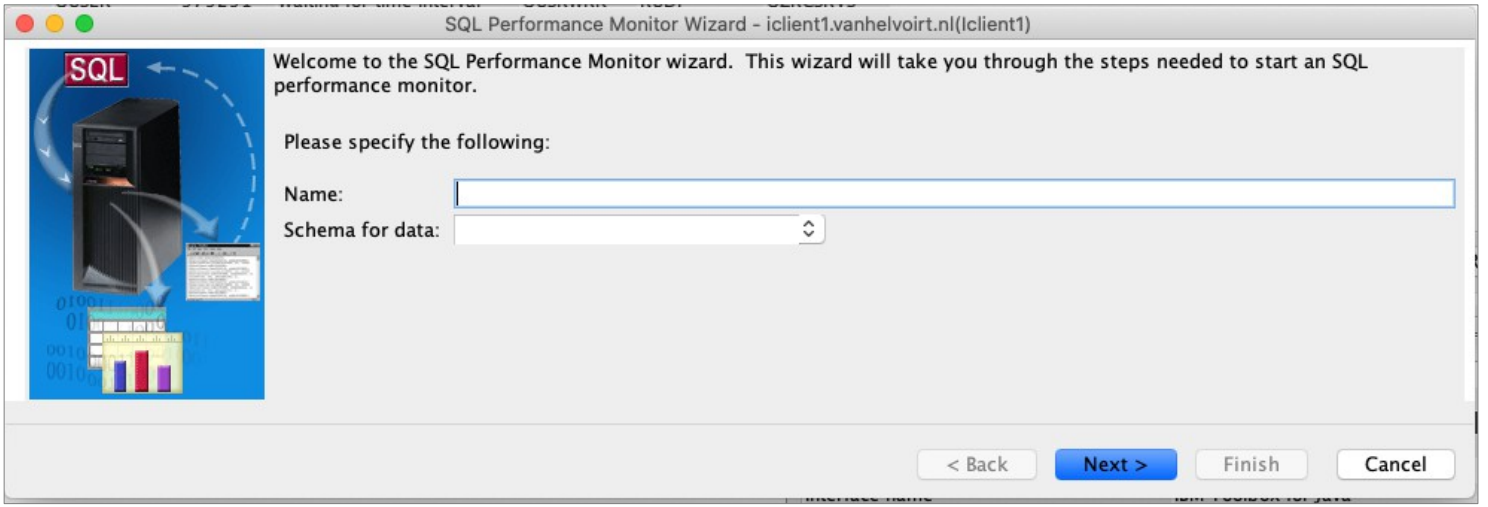
You can gather performance information for a specific query, for every query on the system, or for a group of queries on the system. When a job is monitored by multiple monitors, each monitor is logging rows to a different output table. You can identify rows in the output database table by its unique identification number.

What kinds of statistics you can gather

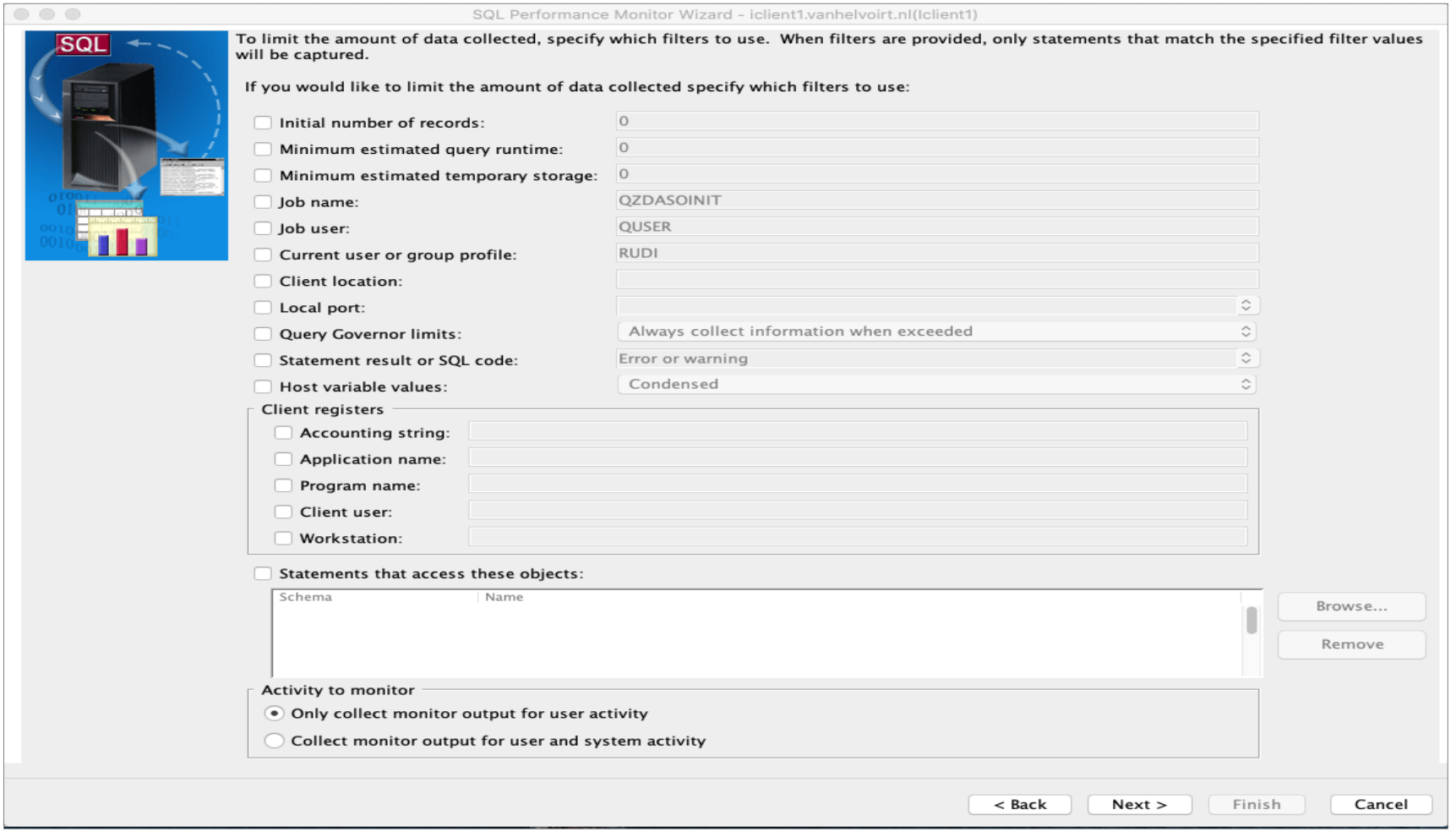
The database monitor provides the same information that is provided with the query optimizer debug messages (**Start Debug (STRDBG)**) and the **Print SQL information (PRTSQLINF)** command. The following is a sampling of the additional information that is gathered by the database monitors:

- System and job name
- SQL statement and subselect number
- Start and end timestamp
- Estimated processing time
- Total rows in table queried
- Number of rows selected
- Estimated number of rows selected
- Estimated number of joined rows
- Key columns for advised index
- Total optimization time
- Join type and method
- ODP implementation

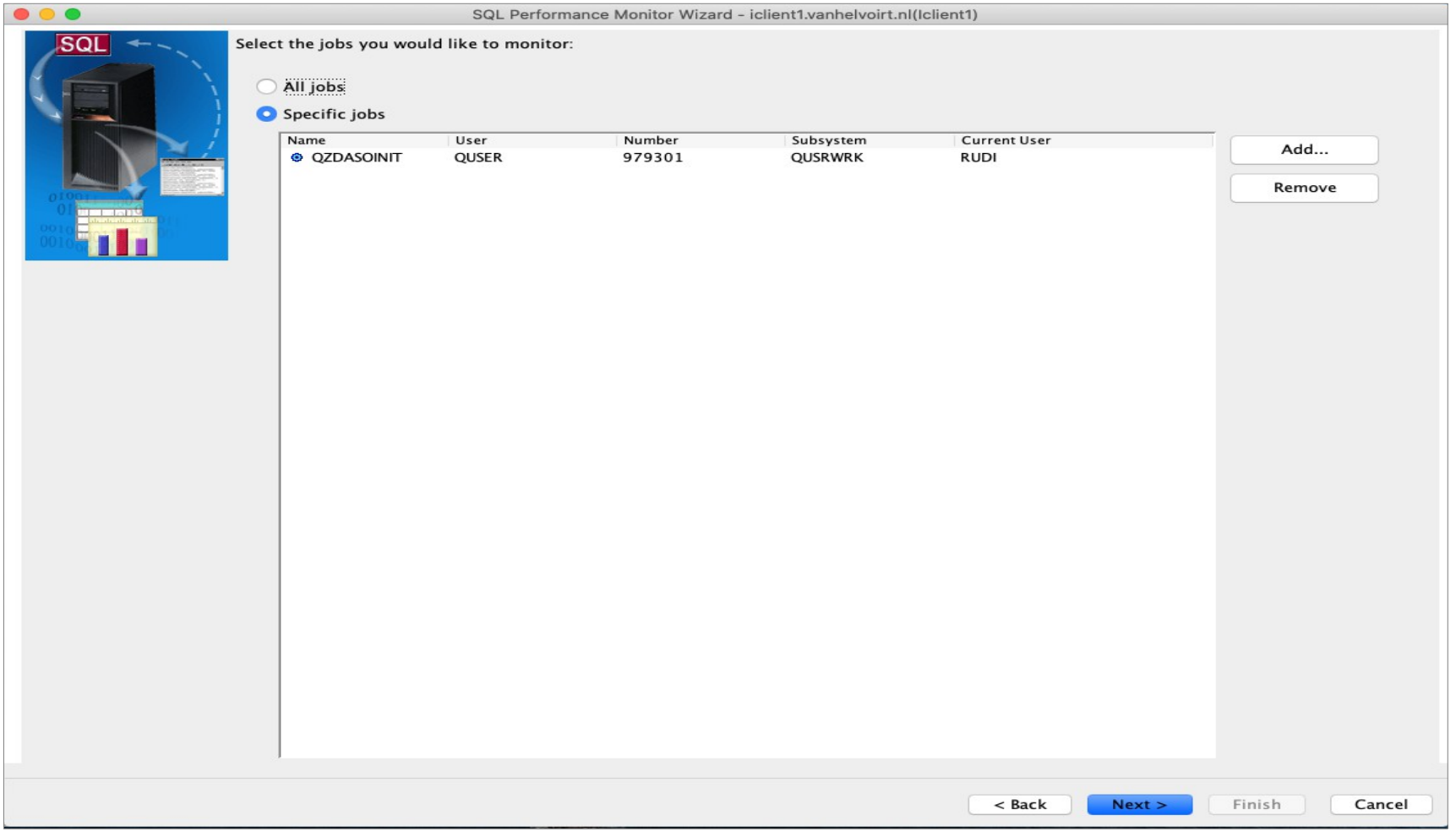
Performance Monitors



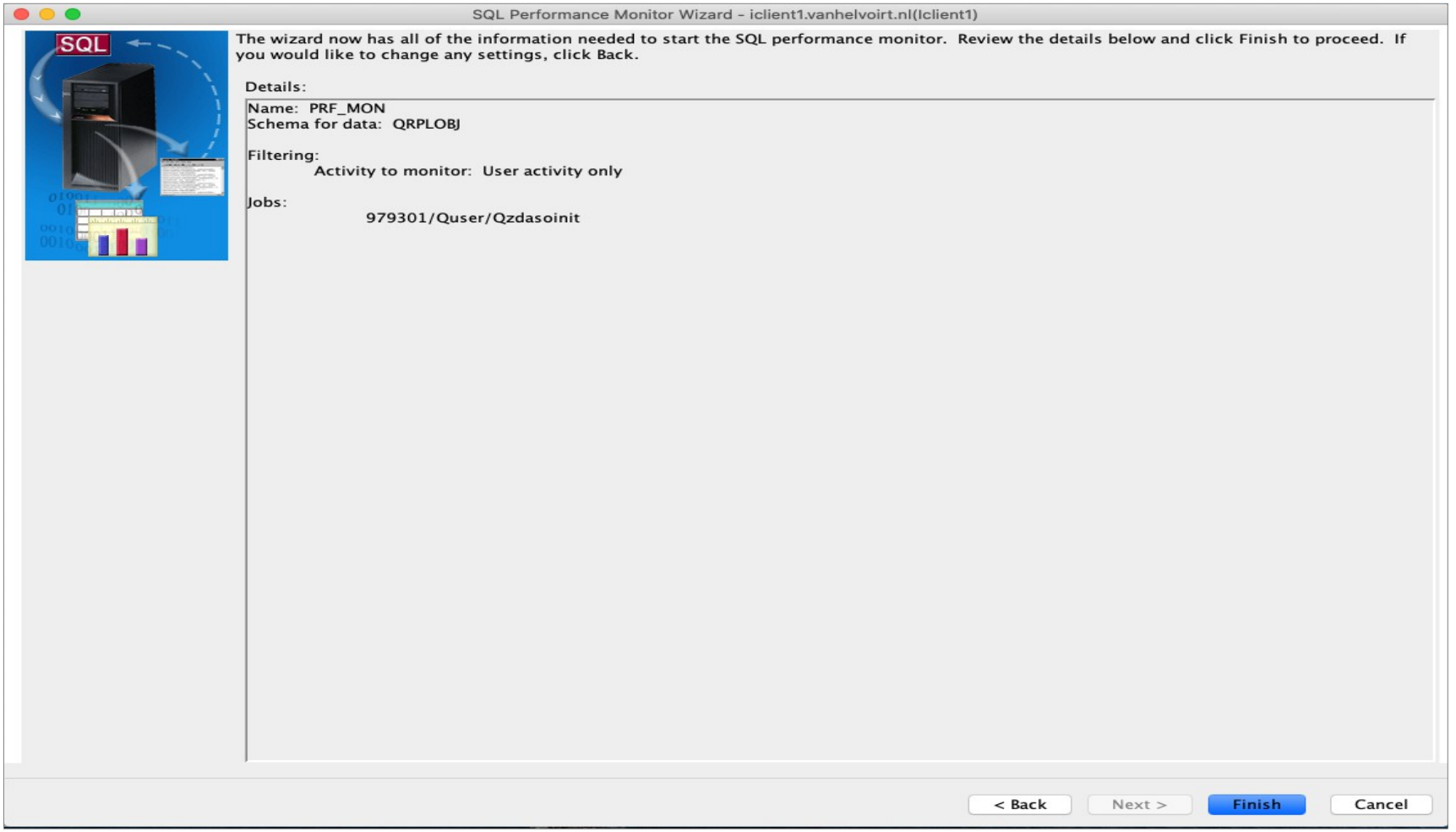
Performance Monitors



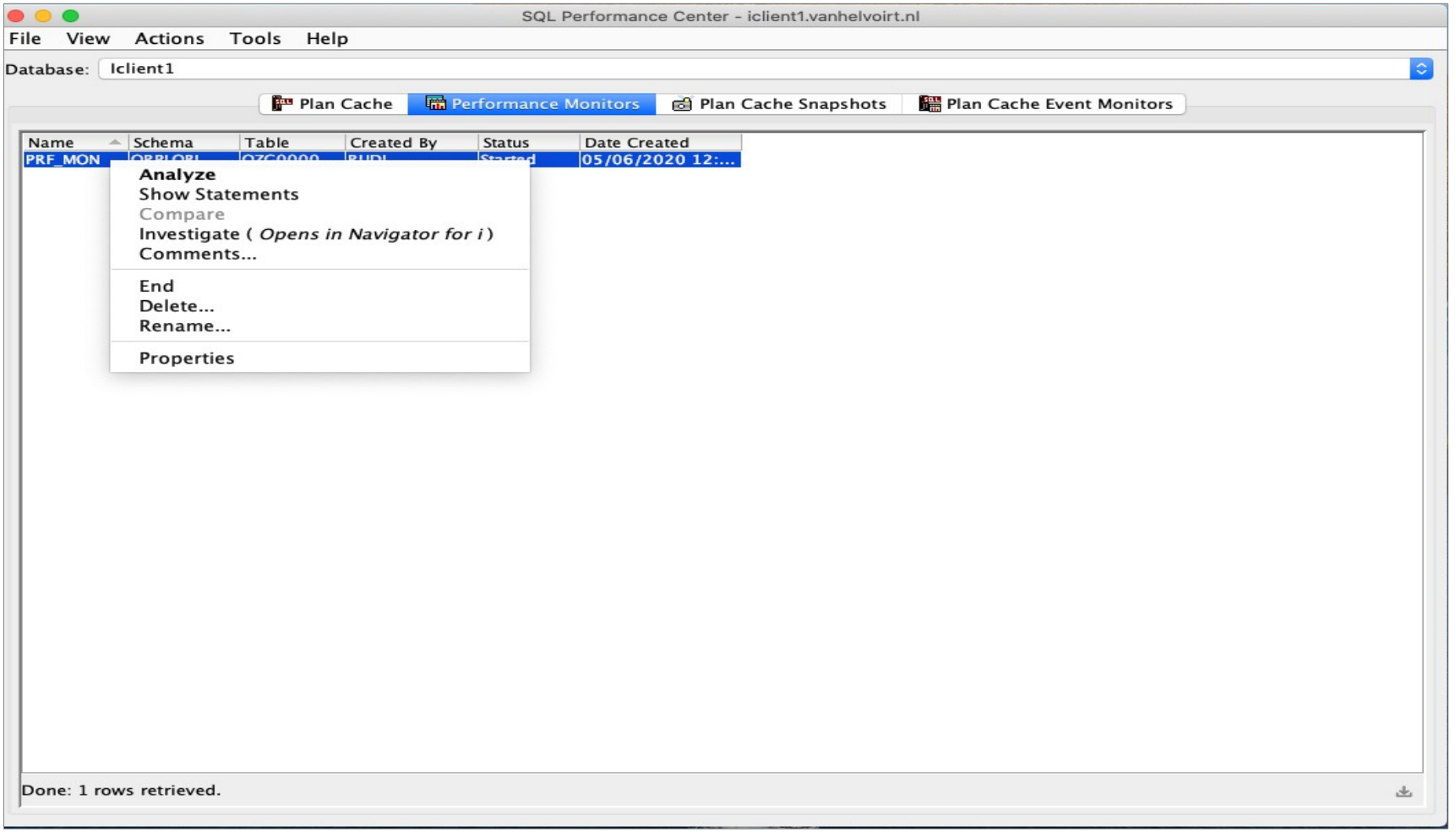
Performance Monitors



Performance Monitors



Performance Monitors

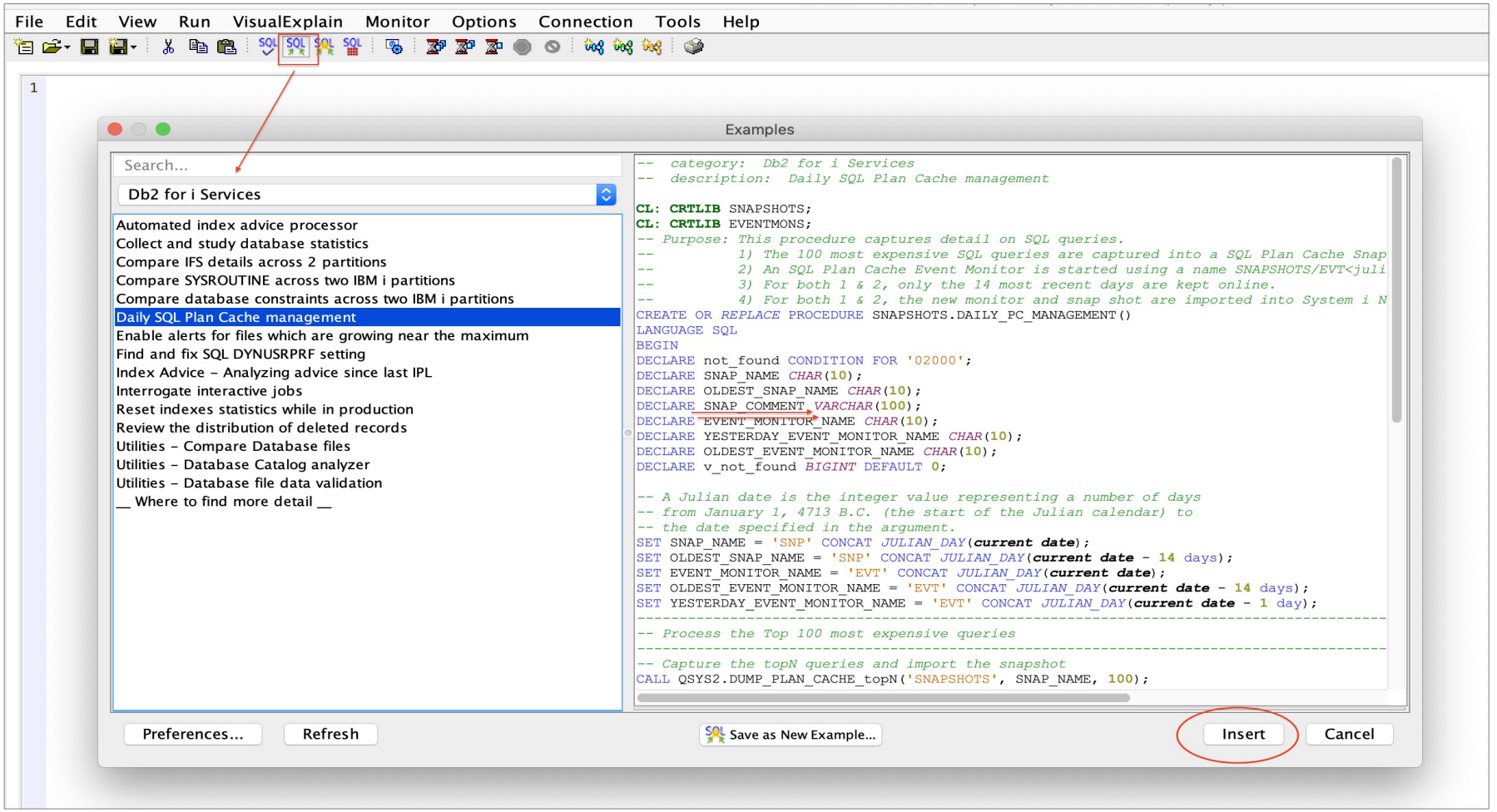


Plan Cache Snapshots & Plan Cache Event Monitors

The SQL plan cache event monitor captures monitor records of plans as they are removed from the plan cache. The event monitor is useful for ensuring that all performance information potentially available in the cache is captured even if plans are removed from the cache. Combining the event monitor output with a plan cache snapshot provides a composite view of the cache from when the event monitor was started until the snapshot is taken.

VAN HELVOIRT AUTOMATISERING

Plan Cache Snapshots & Plan Cache Event Monitors



Plan Cache Snapshots & Plan Cache Event Monitors

File View Actions Tools Help

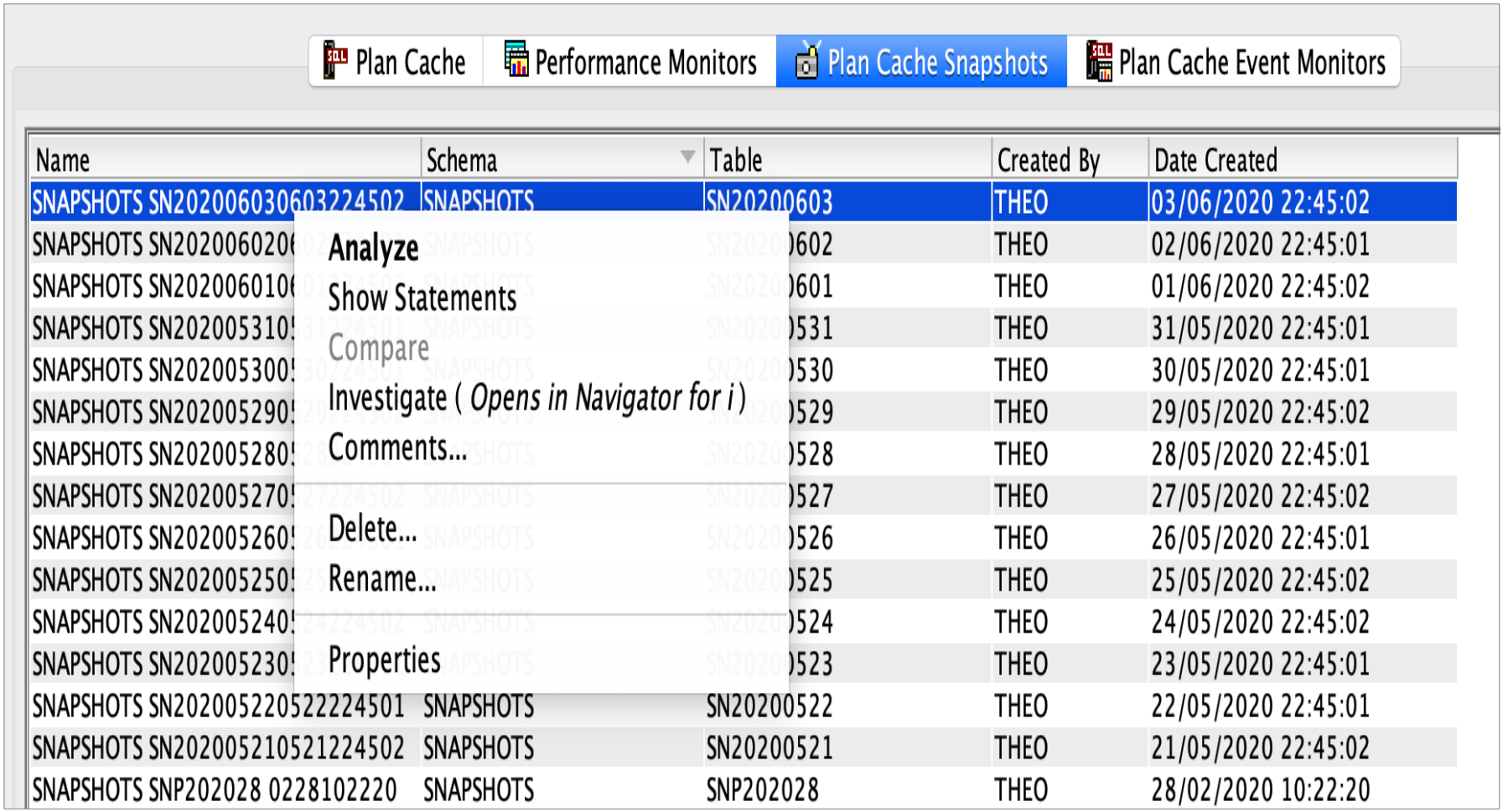
Database: Geeltje

Plan Cache | Performance Monitors | Plan Cache Snapshots | Plan Cache Event Monitors

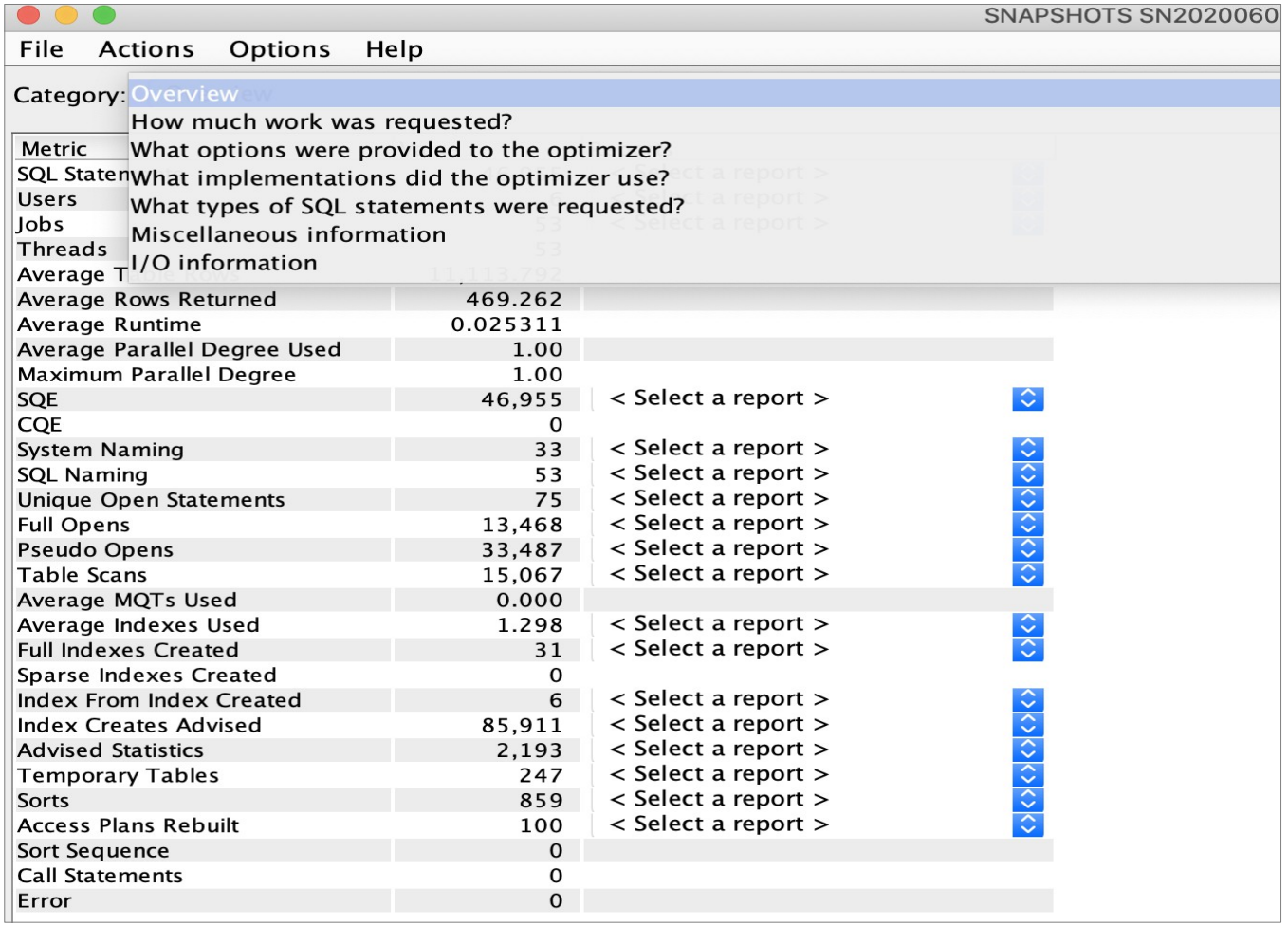
Name	Schema	Table	Created By	Date Created
SNAPSHOTS SN202006030603224502	SNAPSHOTS	SN20200603	THEO	03/06/2020 22:45:02
SNAPSHOTS SN202006020602224501	SNAPSHOTS	SN20200602	THEO	02/06/2020 22:45:01
SNAPSHOTS SN202006010601224502	SNAPSHOTS	SN20200601	THEO	01/06/2020 22:45:02
SNAPSHOTS SN202005310531224501	SNAPSHOTS	SN20200531	THEO	31/05/2020 22:45:01
SNAPSHOTS SN202005300530224501	SNAPSHOTS	SN20200530	THEO	30/05/2020 22:45:01
SNAPSHOTS SN202005290529224502	SNAPSHOTS	SN20200529	THEO	29/05/2020 22:45:02
SNAPSHOTS SN202005280528224501	SNAPSHOTS	SN20200528	THEO	28/05/2020 22:45:01
SNAPSHOTS SN202005270527224502	SNAPSHOTS	SN20200527	THEO	27/05/2020 22:45:02
SNAPSHOTS SN202005260526224501	SNAPSHOTS	SN20200526	THEO	26/05/2020 22:45:01
SNAPSHOTS SN202005250525224502	SNAPSHOTS	SN20200525	THEO	25/05/2020 22:45:02
SNAPSHOTS SN202005240524224502	SNAPSHOTS	SN20200524	THEO	24/05/2020 22:45:02
SNAPSHOTS SN202005230523224501	SNAPSHOTS	SN20200523	THEO	23/05/2020 22:45:01
SNAPSHOTS SN202005220522224501	SNAPSHOTS	SN20200522	THEO	22/05/2020 22:45:01
SNAPSHOTS SN202005210521224502	SNAPSHOTS	SN20200521	THEO	21/05/2020 22:45:02
SNAPSHOTS SNP202028 0228102220	SNAPSHOTS	SNP202028	THEO	28/02/2020 10:22:20

VAN HELVOIRT AUTOMATISERING

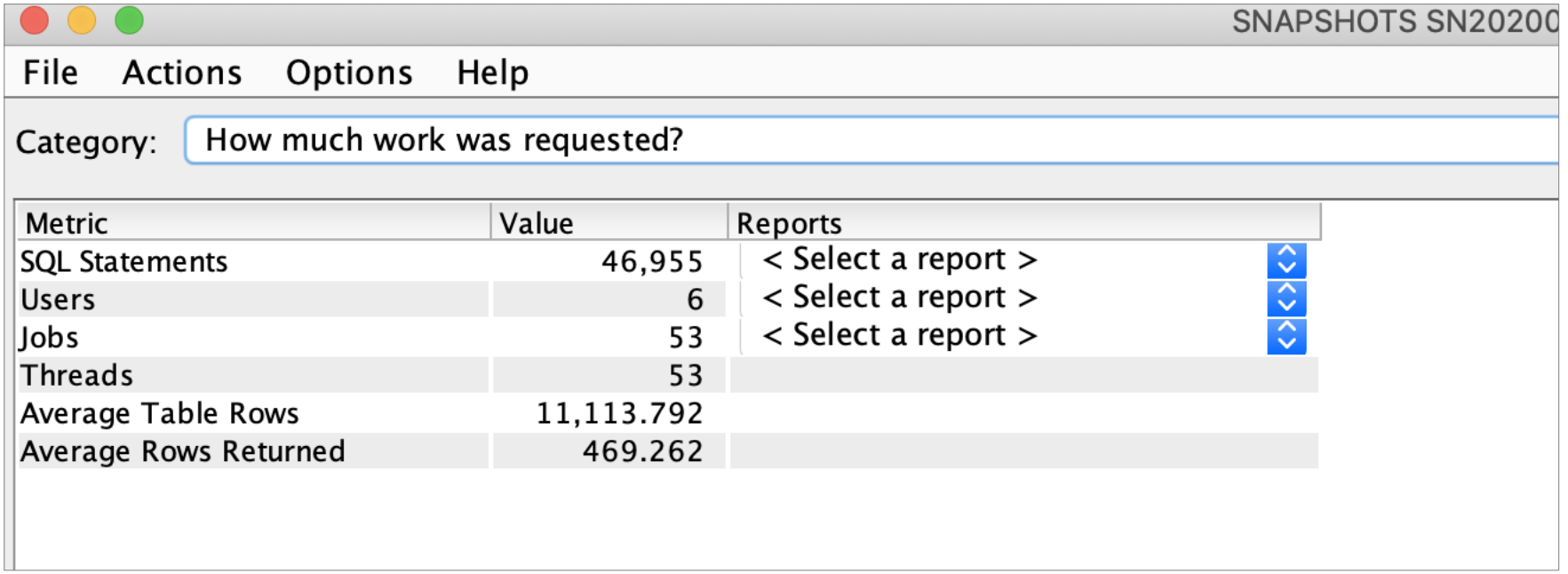
Plan Cache Snapshots



Plan Cache Snapshots – Analyze

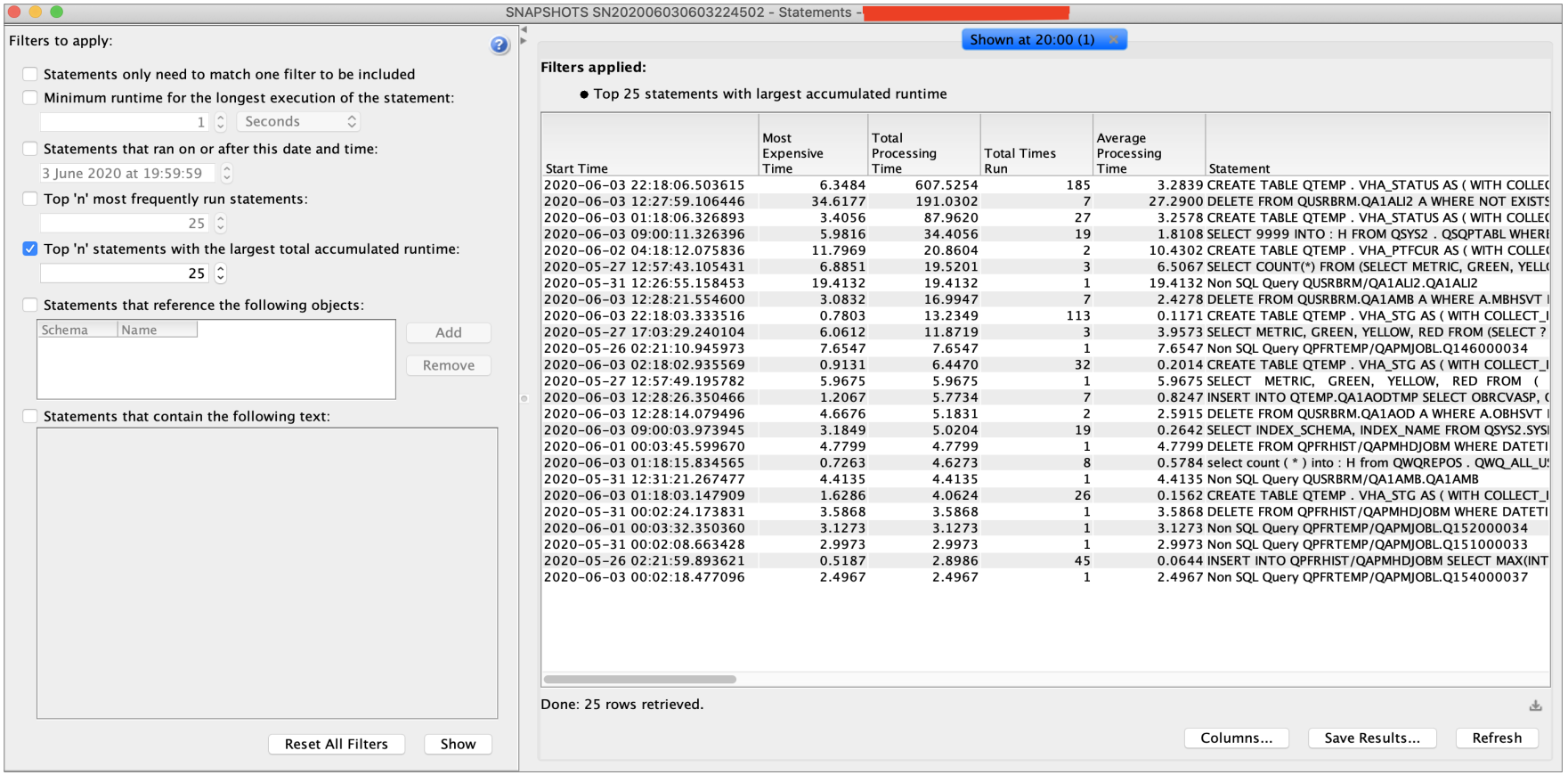


Plan Cache Snapshots – Analyze



VAN HELVOIRT AUTOMATISERING

Plan Cache Snapshots – Show Statements...

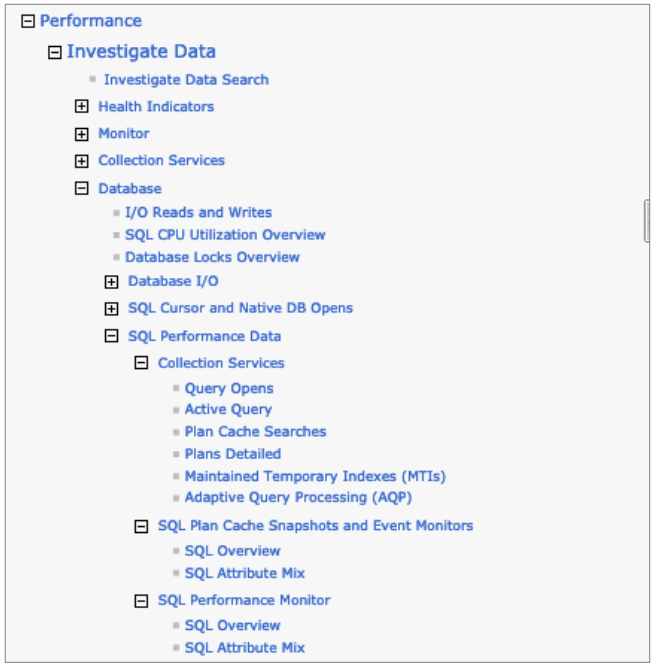


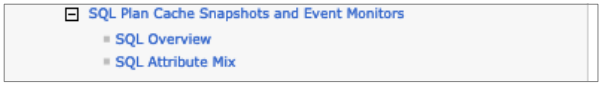
Miscellaneous

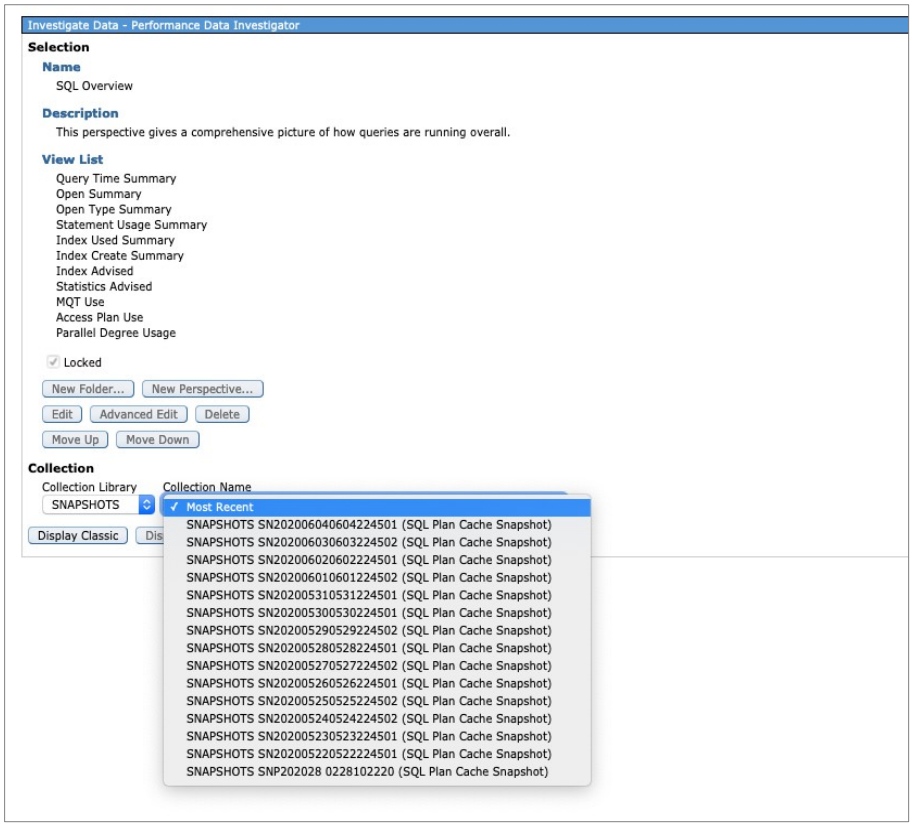
- Navigator for i – Performance Data Investigator
- IBM i ACS – Schemas
- Run SQL Scripts...
- Index Advisor/Visual Explain

VAN HELVOIRT AUTOMATISERING

Performance Data Investigator

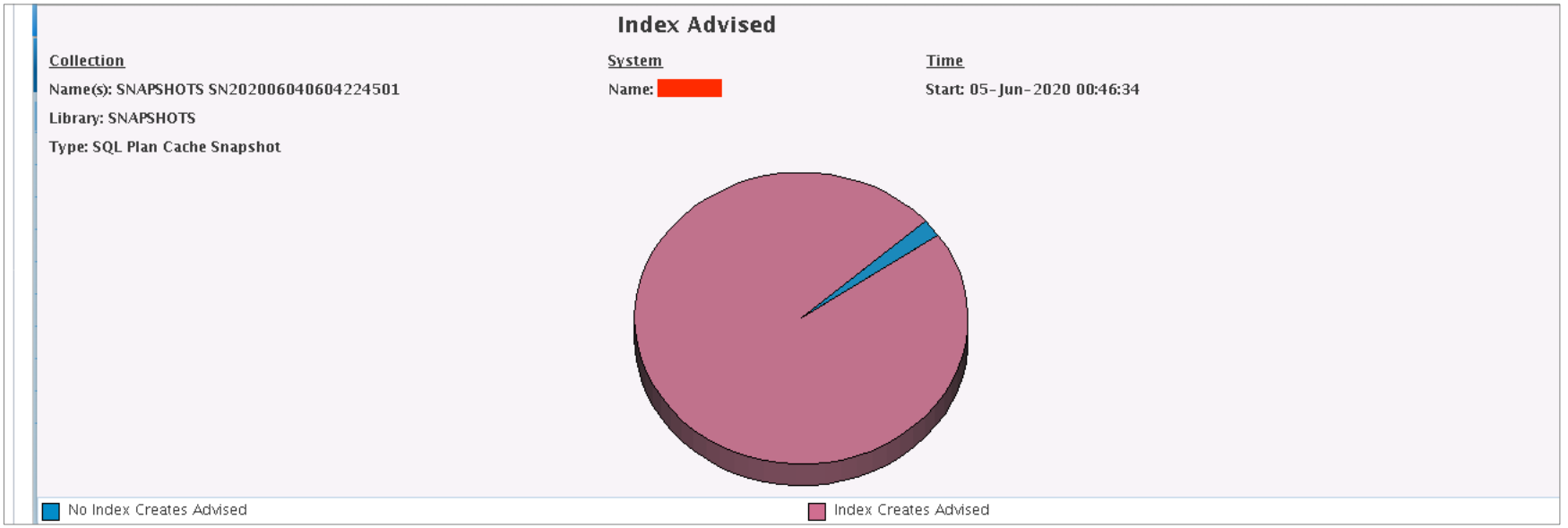




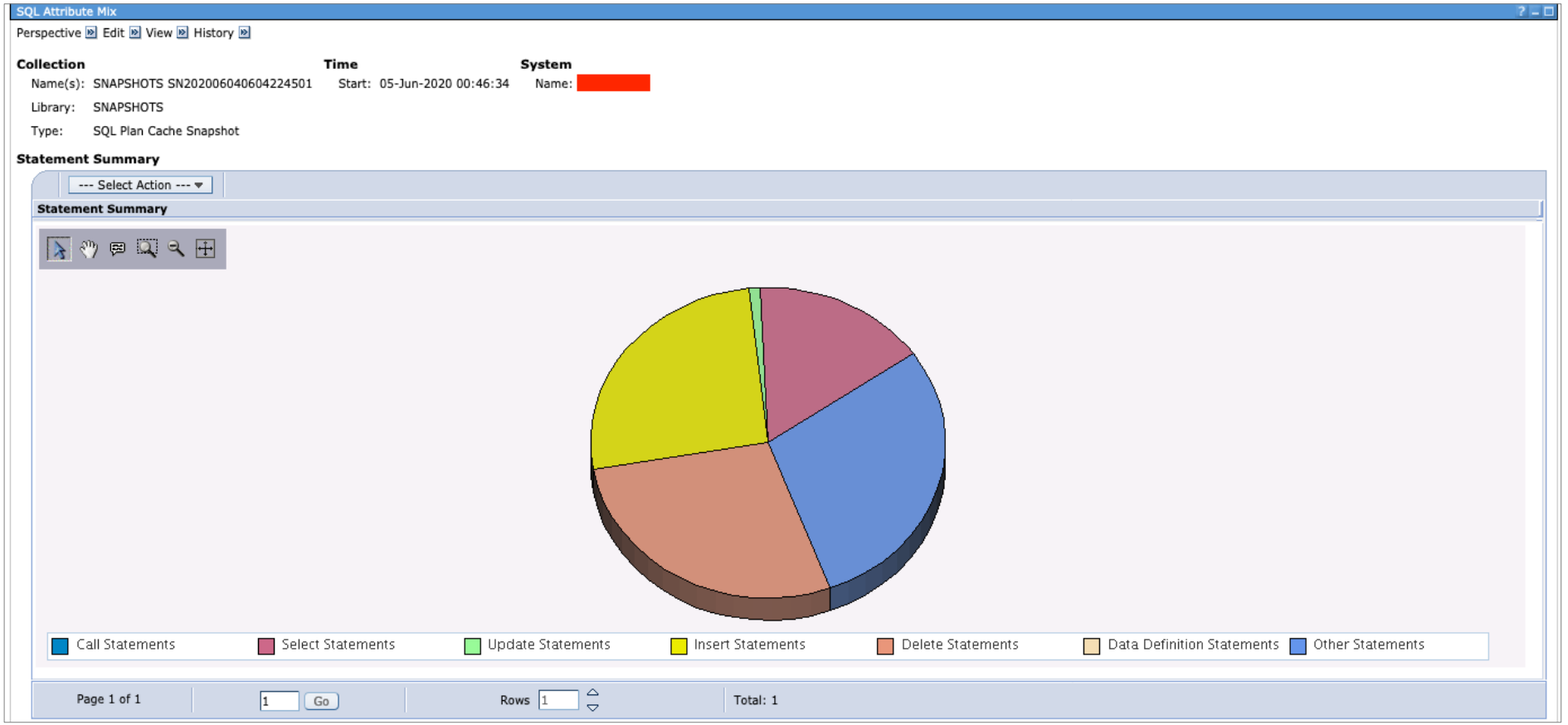


Performance Data Investigator

SQL Overview

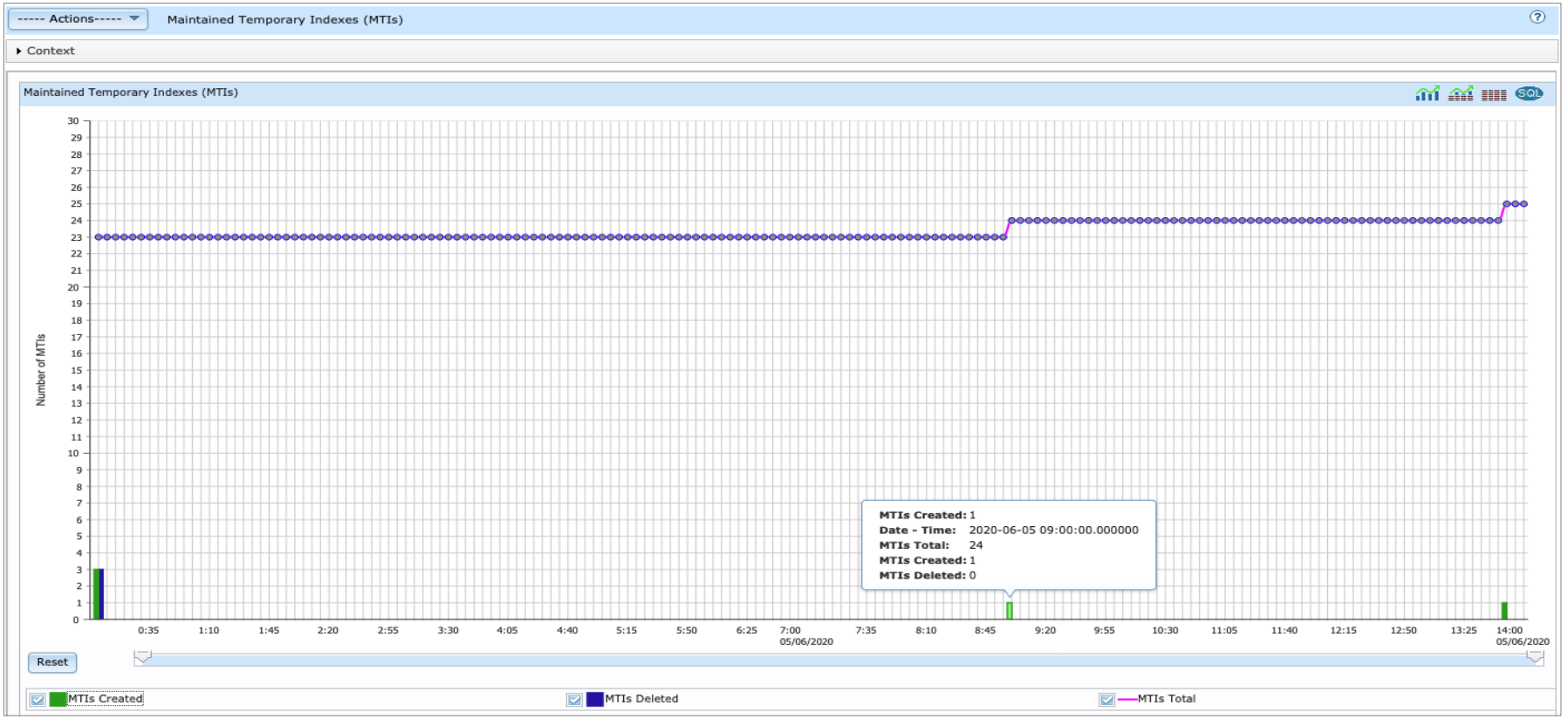


Performance Data Investigator



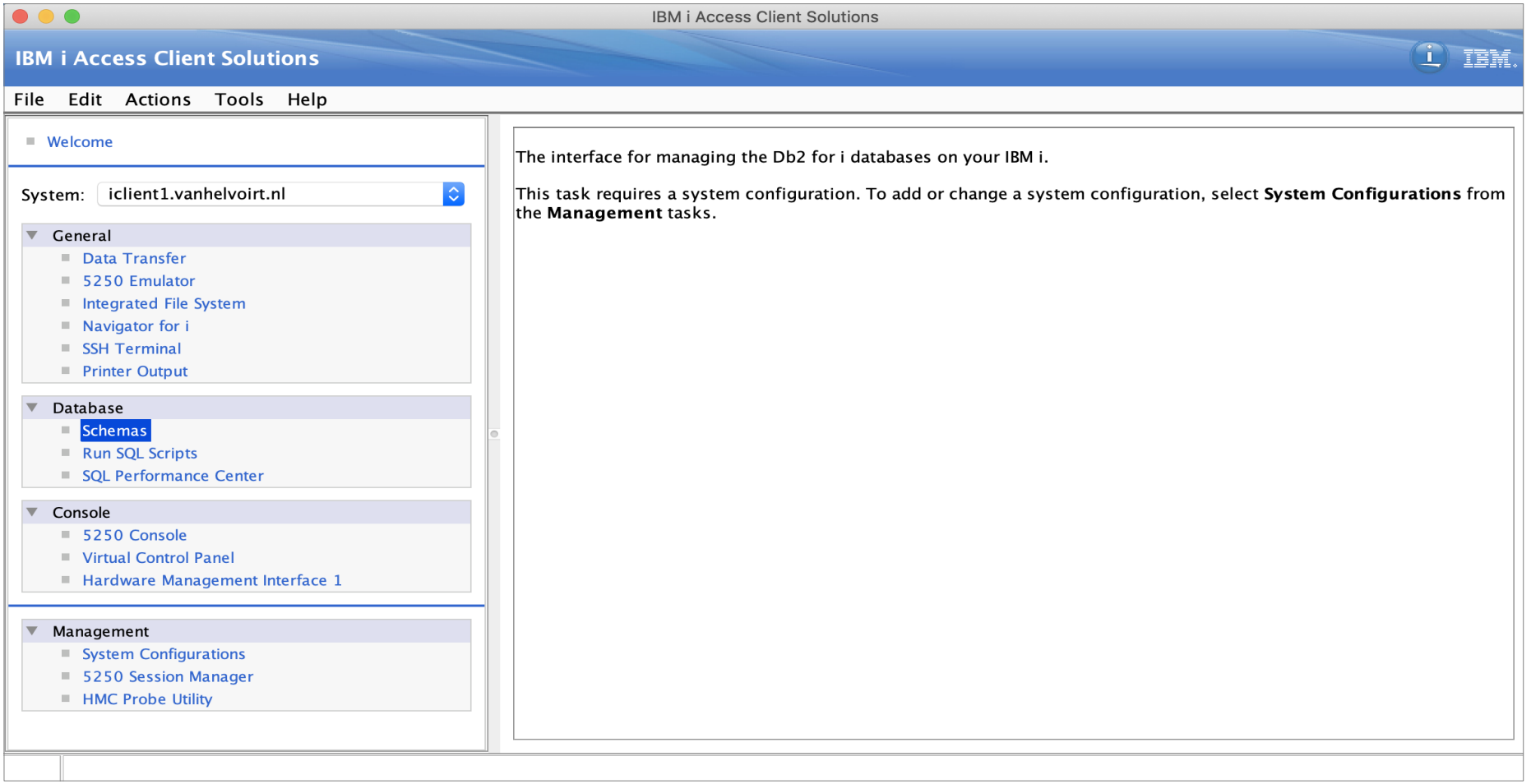
VAN HELVOIRT AUTOMATISERING

Performance Data Investigator

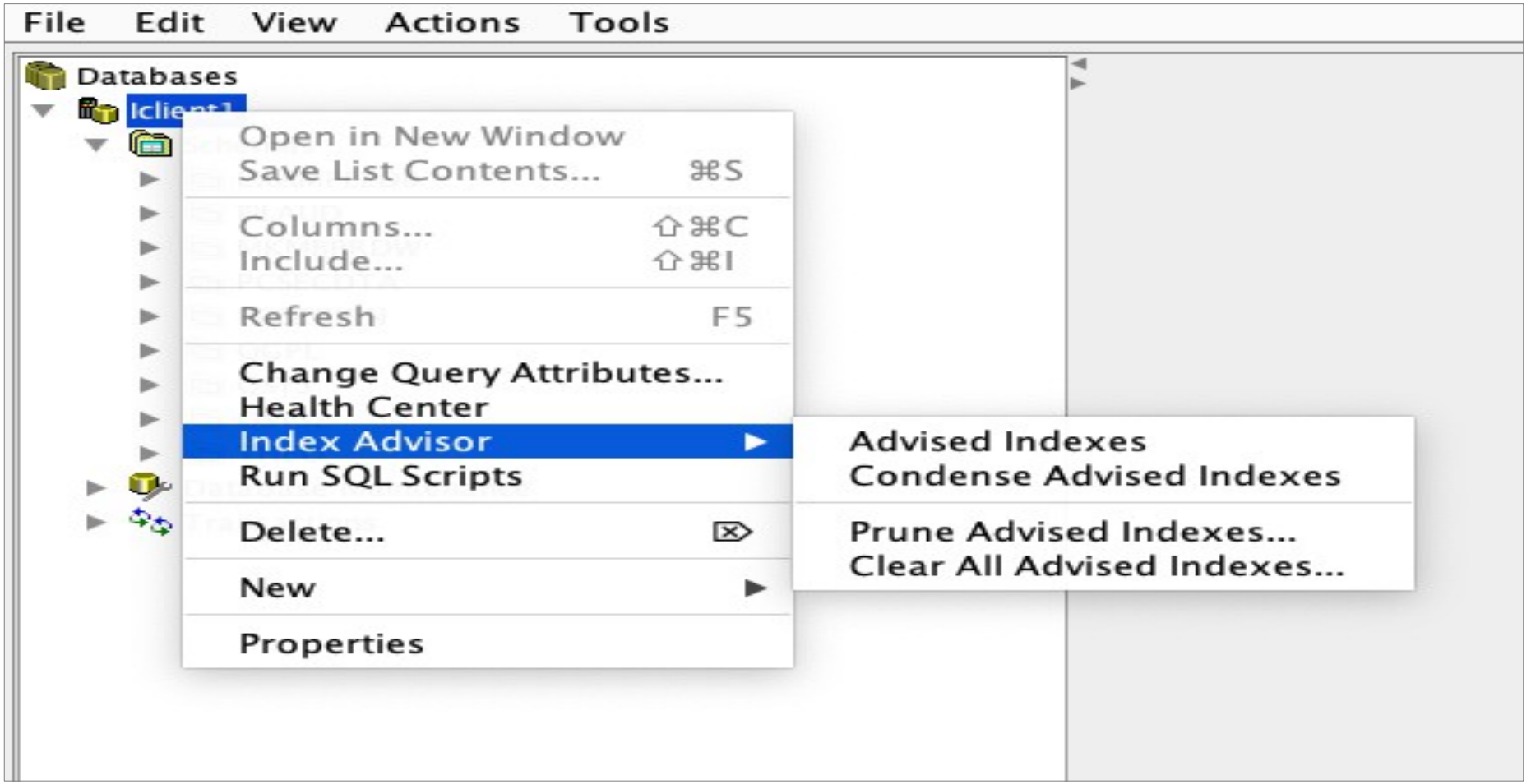


Manage your Maintained Temporary Indexes (MTIs)

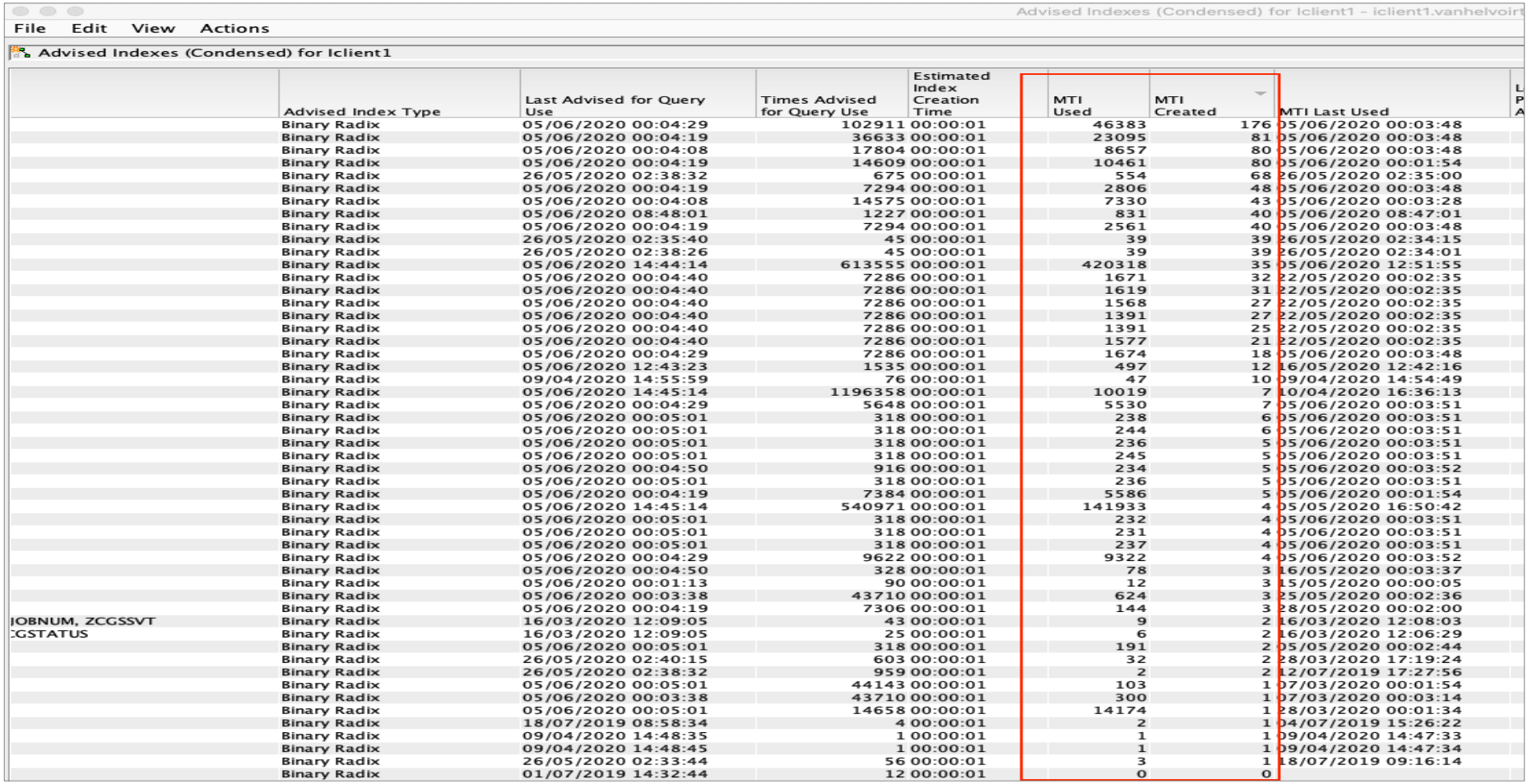
Schemas



Schemas



Schemas



Advised Indexes (Condensed) for Iclient1 – iclient1.vanhelvoirt

File Edit View Actions

Advised Indexes (Condensed) for Iclient1

	Advised Index Type	Last Advised for Query Use	Times Advised for Query Use	Estimated Index Creation Time	MTI Used	MTI Created	MTI Last Used	L P A
	Binary Radix	05/06/2020 00:04:29	102911	00:00:01	46383	176	05/06/2020 00:03:48	
	Binary Radix	05/06/2020 00:04:19	36633	00:00:01	23095	81	05/06/2020 00:03:48	
	Binary Radix	05/06/2020 00:04:08	17804	00:00:01	8657	80	05/06/2020 00:03:48	
	Binary Radix	05/06/2020 00:04:19	14609	00:00:01	10461	80	05/06/2020 00:01:54	
	Binary Radix	26/05/2020 02:38:32	675	00:00:01	554	68	26/05/2020 02:35:00	
	Binary Radix	05/06/2020 00:04:19	7294	00:00:01	2806	48	05/06/2020 00:03:48	
	Binary Radix	05/06/2020 00:04:08	14575	00:00:01	7330	43	05/06/2020 00:03:28	
	Binary Radix	05/06/2020 08:48:01	1227	00:00:01	831	40	05/06/2020 08:47:01	
	Binary Radix	05/06/2020 00:04:19	7294	00:00:01	2561	40	05/06/2020 00:03:48	
	Binary Radix	26/05/2020 02:35:40	45	00:00:01	39	39	26/05/2020 02:34:15	
	Binary Radix	26/05/2020 02:38:26	45	00:00:01	39	39	26/05/2020 02:34:01	
	Binary Radix	05/06/2020 14:44:14	613555	00:00:01	420318	35	05/06/2020 12:51:55	
	Binary Radix	05/06/2020 00:04:40	7286	00:00:01	1671	32	22/05/2020 00:02:35	
	Binary Radix	05/06/2020 00:04:40	7286	00:00:01	1619	31	22/05/2020 00:02:35	
	Binary Radix	05/06/2020 00:04:40	7286	00:00:01	1568	27	22/05/2020 00:02:35	
	Binary Radix	05/06/2020 00:04:40	7286	00:00:01	1391	27	22/05/2020 00:02:35	
	Binary Radix	05/06/2020 00:04:40	7286	00:00:01	1391	25	22/05/2020 00:02:35	
	Binary Radix	05/06/2020 00:04:40	7286	00:00:01	1577	21	22/05/2020 00:02:35	
	Binary Radix	05/06/2020 00:04:29	7286	00:00:01	1674	18	05/06/2020 00:03:48	
	Binary Radix	05/06/2020 12:43:23	1535	00:00:01	497	12	16/05/2020 12:42:16	
	Binary Radix	09/04/2020 14:55:59	76	00:00:01	47	10	09/04/2020 14:54:49	
	Binary Radix	05/06/2020 14:45:14	1196358	00:00:01	10019	7	10/04/2020 16:36:13	
	Binary Radix	05/06/2020 00:04:29	5648	00:00:01	5530	7	05/06/2020 00:03:51	
	Binary Radix	05/06/2020 00:05:01	318	00:00:01	238	6	05/06/2020 00:03:51	
	Binary Radix	05/06/2020 00:05:01	318	00:00:01	244	6	05/06/2020 00:03:51	
	Binary Radix	05/06/2020 00:05:01	318	00:00:01	236	5	05/06/2020 00:03:51	
	Binary Radix	05/06/2020 00:05:01	318	00:00:01	245	5	05/06/2020 00:03:51	
	Binary Radix	05/06/2020 00:04:50	916	00:00:01	234	5	05/06/2020 00:03:52	
	Binary Radix	05/06/2020 00:05:01	318	00:00:01	236	5	05/06/2020 00:03:51	
	Binary Radix	05/06/2020 00:04:19	7384	00:00:01	5586	5	05/06/2020 00:01:54	
	Binary Radix	05/06/2020 14:45:14	540971	00:00:01	141933	4	05/05/2020 16:50:42	
	Binary Radix	05/06/2020 00:05:01	318	00:00:01	232	4	05/06/2020 00:03:51	
	Binary Radix	05/06/2020 00:05:01	318	00:00:01	231	4	05/06/2020 00:03:51	
	Binary Radix	05/06/2020 00:05:01	318	00:00:01	237	4	05/06/2020 00:03:51	
	Binary Radix	05/06/2020 00:04:29	9622	00:00:01	9322	4	05/06/2020 00:03:52	
	Binary Radix	05/06/2020 00:04:50	328	00:00:01	78	3	16/05/2020 00:03:37	
	Binary Radix	05/06/2020 00:01:13	90	00:00:01	12	3	15/05/2020 00:00:05	
	Binary Radix	05/06/2020 00:03:38	43710	00:00:01	624	3	25/05/2020 00:02:36	
	Binary Radix	05/06/2020 00:04:19	7306	00:00:01	144	3	28/05/2020 00:02:00	
IOBNUM, ZCGSSVT	Binary Radix	16/03/2020 12:09:05	43	00:00:01	9	2	16/03/2020 12:08:03	
CGSTATUS	Binary Radix	16/03/2020 12:09:05	25	00:00:01	6	2	16/03/2020 12:06:29	
	Binary Radix	05/06/2020 00:05:01	318	00:00:01	191	2	05/05/2020 00:02:44	
	Binary Radix	26/05/2020 02:40:15	603	00:00:01	32	2	28/03/2020 17:19:24	
	Binary Radix	26/05/2020 02:38:32	959	00:00:01	2	2	12/07/2019 17:27:56	
	Binary Radix	05/06/2020 00:05:01	44143	00:00:01	103	1	07/03/2020 00:01:54	
	Binary Radix	05/06/2020 00:03:38	43710	00:00:01	300	1	07/03/2020 00:03:14	
	Binary Radix	05/06/2020 00:05:01	14658	00:00:01	14174	1	28/03/2020 00:01:34	
	Binary Radix	18/07/2019 08:58:34	4	00:00:01	2	1	04/07/2019 15:26:22	
	Binary Radix	09/04/2020 14:48:35	1	00:00:01	1	1	09/04/2020 14:47:33	
	Binary Radix	09/04/2020 14:48:45	1	00:00:01	1	1	09/04/2020 14:47:34	
	Binary Radix	26/05/2020 02:33:44	56	00:00:01	3	1	18/07/2019 09:16:14	
	Binary Radix	01/07/2019 14:32:44	12	00:00:01	0	0		

VAN HELVOIRT AUTOMATISERING

Schemas

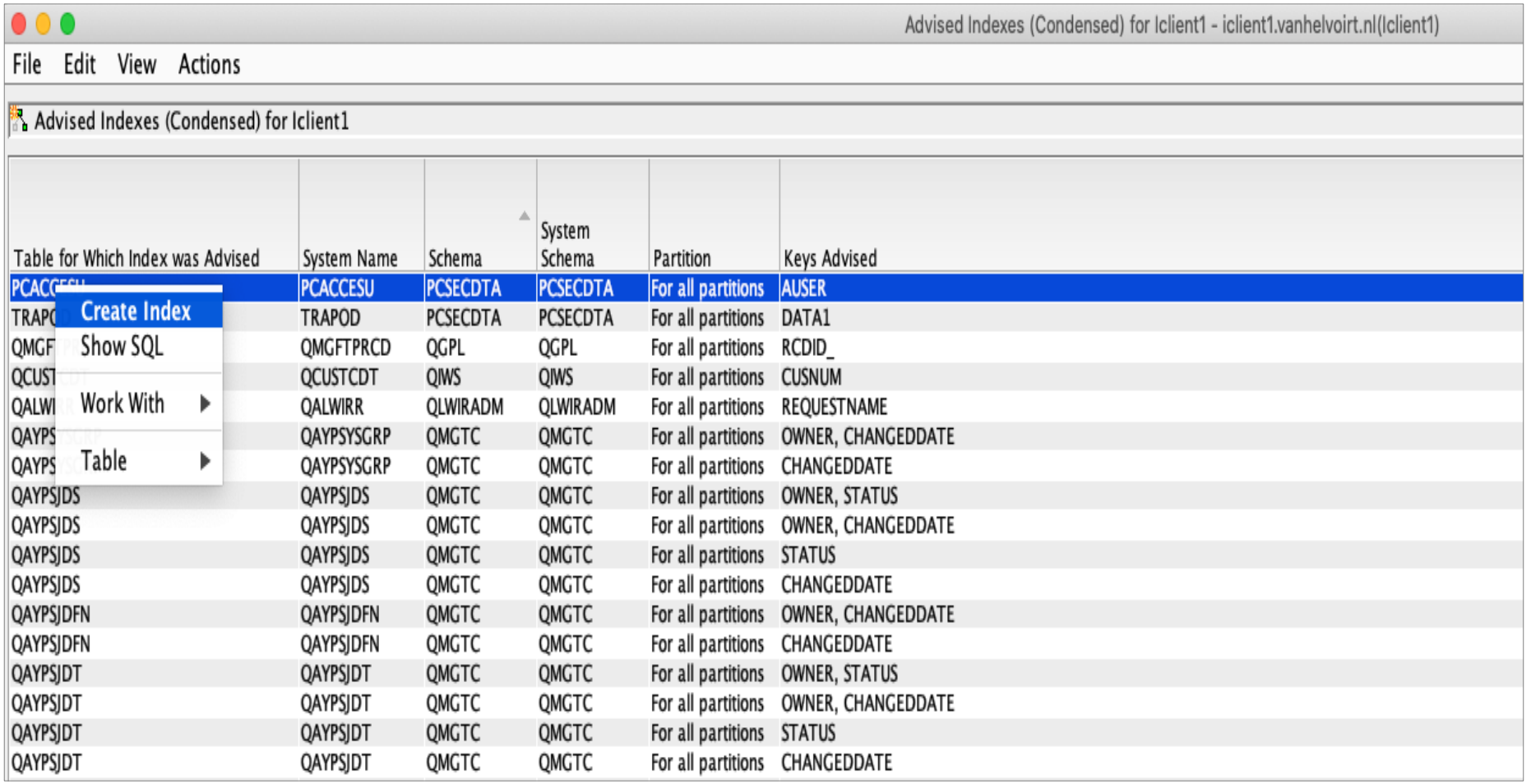


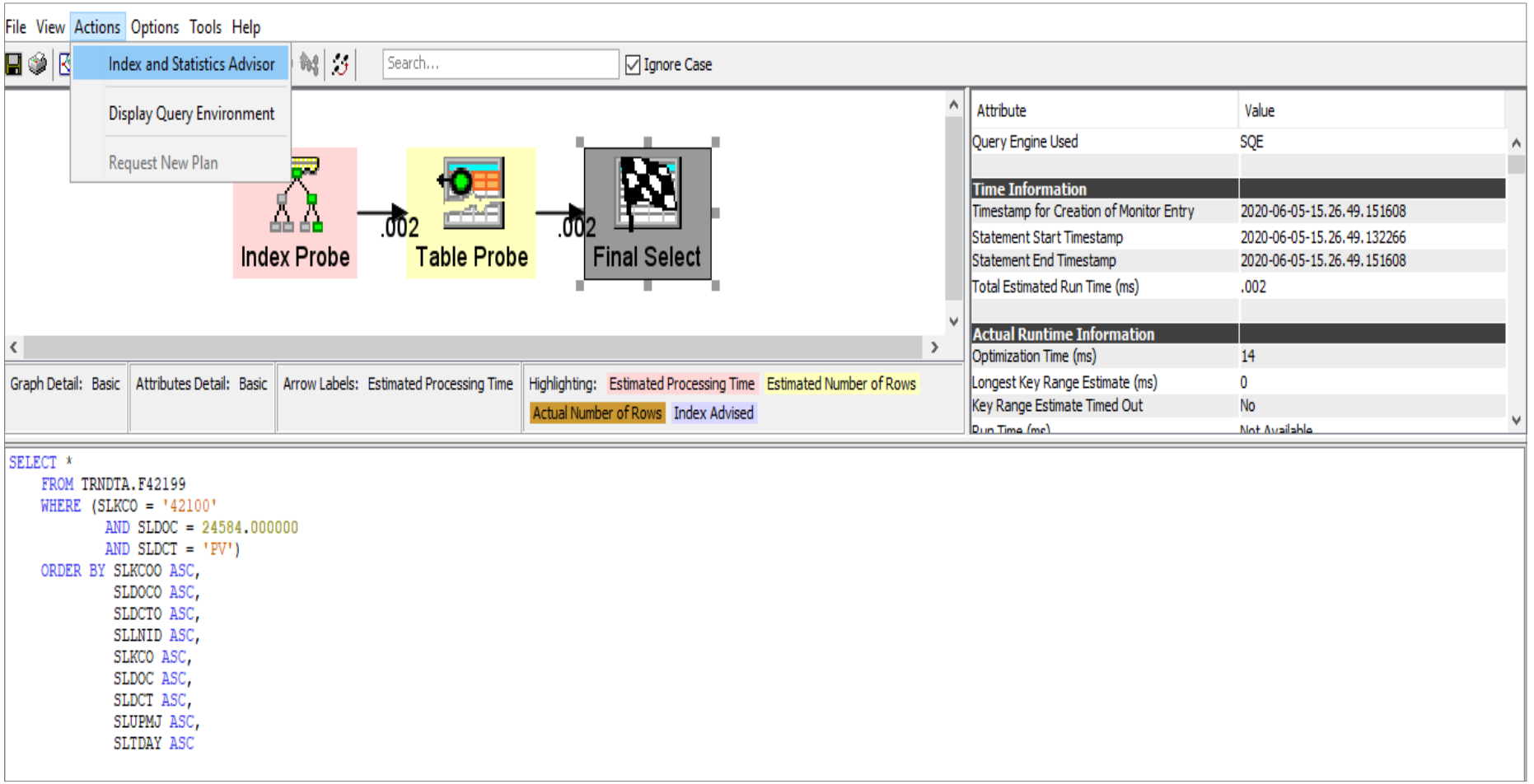
Table for Which Index was Advised	System Name	Schema	System Schema	Partition	Keys Advised
PCACCESU	PCACCESU	PCSECDTA	PCSECDTA	For all partitions	AUSER
TRAPOD	TRAPOD	PCSECDTA	PCSECDTA	For all partitions	DATA1
QMGFTPRCD	QMGFTPRCD	QGPL	QGPL	For all partitions	RCDID_
QCUSTCDT	QCUSTCDT	QIWS	QIWS	For all partitions	CUSNUM
QALWIRR	QALWIRR	QLWIRADM	QLWIRADM	For all partitions	REQUESTNAME
QAYPSYSGRP	QAYPSYSGRP	QMGTC	QMGTC	For all partitions	OWNER, CHANGEDDATE
QAYPSYSGRP	QAYPSYSGRP	QMGTC	QMGTC	For all partitions	CHANGEDDATE
QAYPSJDS	QAYPSJDS	QMGTC	QMGTC	For all partitions	OWNER, STATUS
QAYPSJDS	QAYPSJDS	QMGTC	QMGTC	For all partitions	OWNER, CHANGEDDATE
QAYPSJDS	QAYPSJDS	QMGTC	QMGTC	For all partitions	STATUS
QAYPSJDS	QAYPSJDS	QMGTC	QMGTC	For all partitions	CHANGEDDATE
QAYPSJDFN	QAYPSJDFN	QMGTC	QMGTC	For all partitions	OWNER, CHANGEDDATE
QAYPSJDFN	QAYPSJDFN	QMGTC	QMGTC	For all partitions	CHANGEDDATE
QAYPSJDT	QAYPSJDT	QMGTC	QMGTC	For all partitions	OWNER, STATUS
QAYPSJDT	QAYPSJDT	QMGTC	QMGTC	For all partitions	OWNER, CHANGEDDATE
QAYPSJDT	QAYPSJDT	QMGTC	QMGTC	For all partitions	STATUS
QAYPSJDT	QAYPSJDT	QMGTC	QMGTC	For all partitions	CHANGEDDATE

Run SQL Scripts...

```
1  SELECT *
2      FROM TRNDTA.F42199
3      WHERE (SLKCO = '42100'
4             AND SLDOC = 24584.000000
5             AND SLDCT = 'PV')
6      ORDER BY SLKCOO ASC,
7               SLDOCO ASC,
8               SLDCTO ASC,
9               SLLNID ASC,
10              SLKCO ASC,
11              SLDOC ASC,
12              SLDCT ASC,
13              SLUPMJ ASC,
14              SLTDAY ASC
```

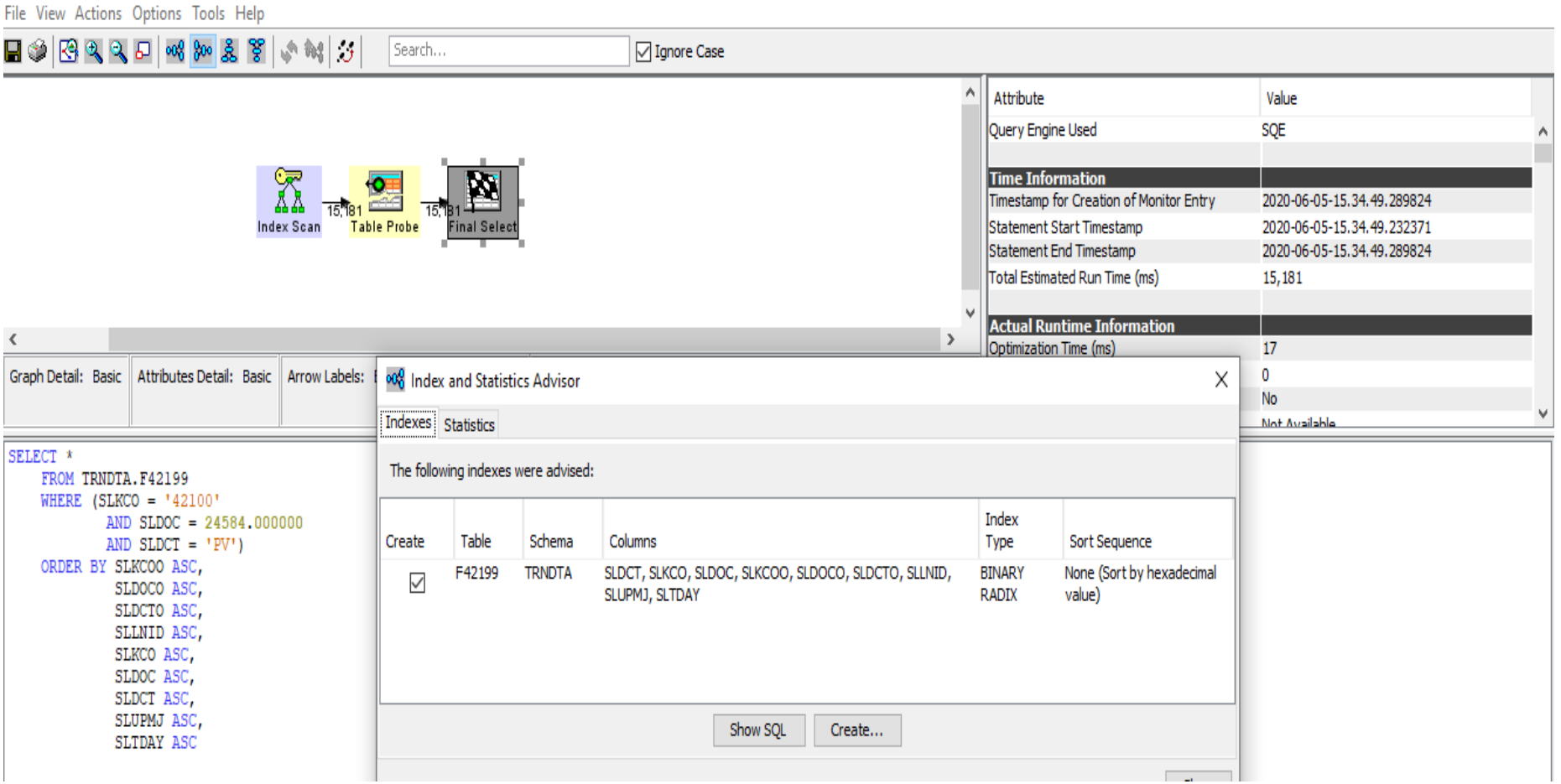
```
Connected to relational database [redacted] HA - 110897/QUSER/QZDASOINIT using JDBC configuration 'Default'.


[ 06/05/2020, 02:58:31 PM ]  Run Selected...
SELECT * FROM TRNDTA.F42199 WHERE (SLKCO = '42100' AND SLDOC = 24584.000000 AND SLDCT = 'PV') ORDER BY SLKCOO ASC, SLDOCO ASC, SLDCTO ASC, SLLNID ASC, SLKCO ASC, SLDOC ASC, SLDCT ASC, SLUPMJ ASC, SLTDAY ASC
Statement ran successfully (28,895 ms = 28.895 sec)
```

Visual Explain

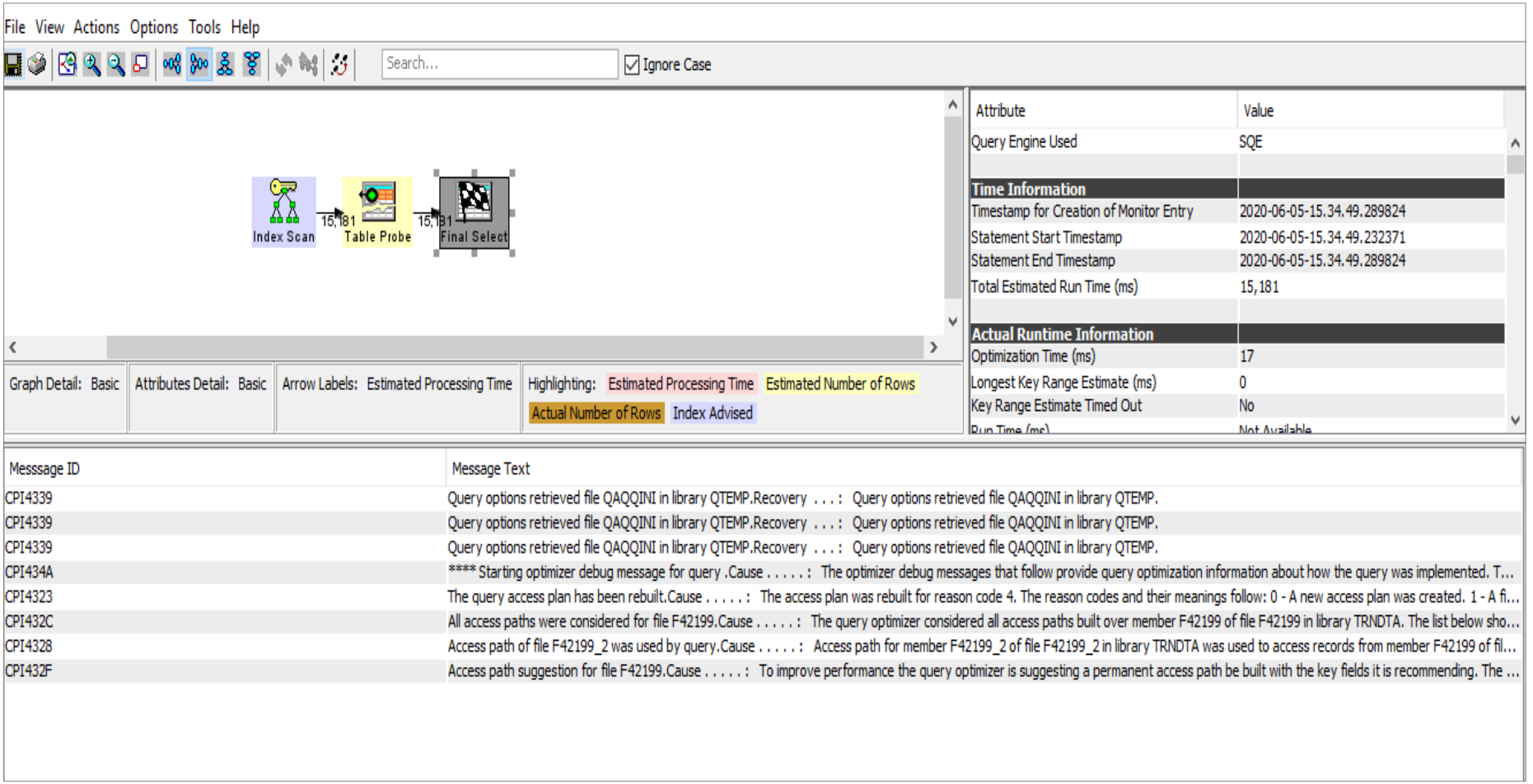



```
SELECT *
    FROM TRNDTA.F42199
    WHERE (SLKCO = '42100'
           AND SLDOC = 24584.000000
           AND SLDCT = 'PV')
    ORDER BY SLKCOO ASC,
             SLDOCO ASC,
             SLDCTO ASC,
             SLLNID ASC,
             SLKCO ASC,
             SLDOC ASC,
             SLDCT ASC,
             SLUPMJ ASC,
             SLTDAY ASC
```

Visual Explain



Schemas



Schemas

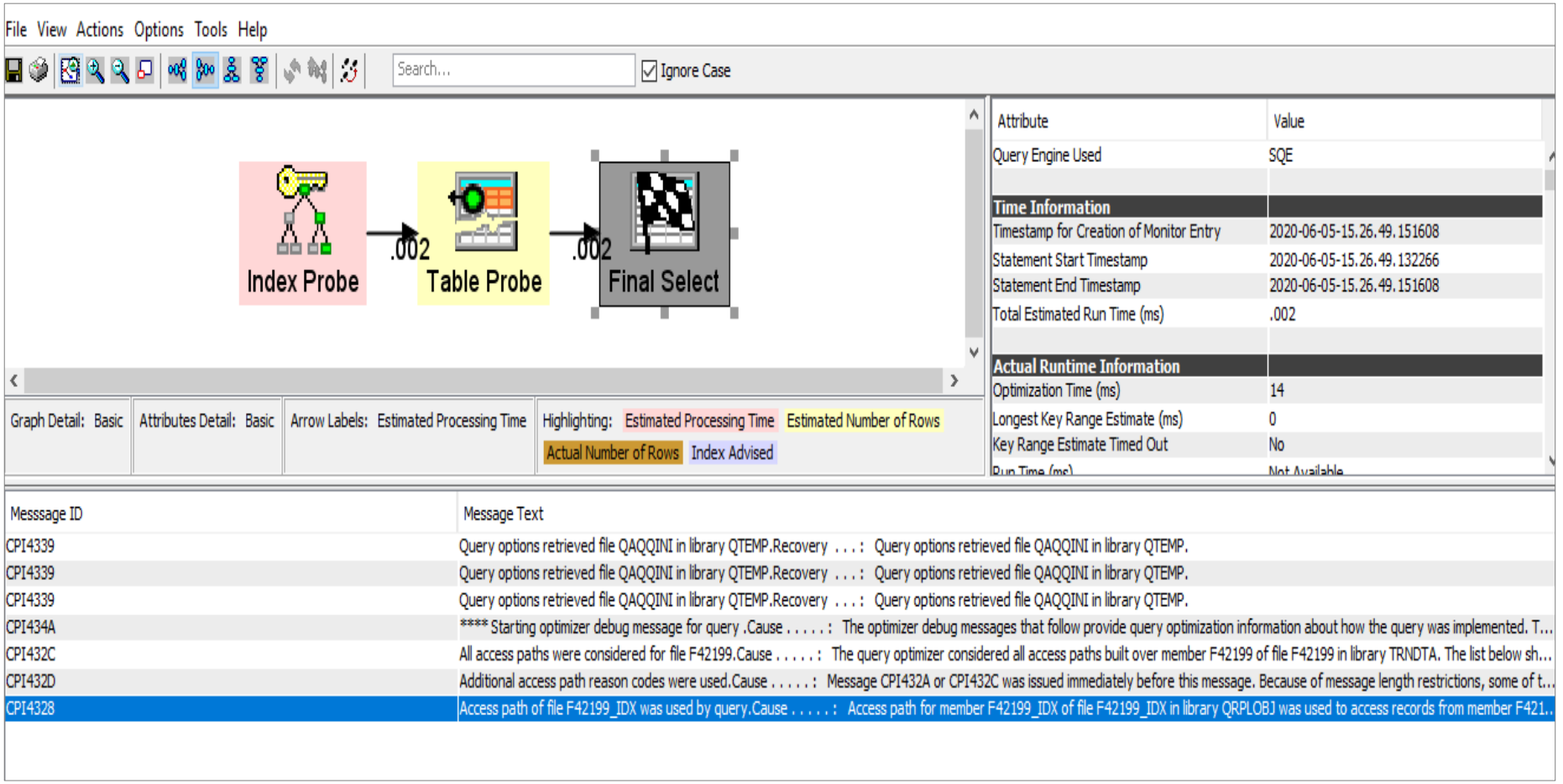
```
1  /*
2  Creating TRNDTA.F42199_IDX [Index]
3  When creating this index the database connection should have a sort sequence of *HEX.
4   */
5  CREATE INDEX QRPLOBJ.F42199_IDX
6      ON TRNDTA.F42199 (SLDCT ASC, SLKCO ASC, SLDOC ASC, SLKCOO ASC, SLDOCO ASC, SLDCTO ASC, SLLNID ASC, SLUPMJ ASC, SLTDAY ASC) UNIT
7          ANY KEEP IN MEMORY NO;
8
9  /* Setting label text for TRNDTA.F42199_IDX */
10 LABEL ON INDEX QRPLOBJ.F42199_IDX IS 'Index generated from Index Advisor';
11
12 ;
```

Schemas

```
1  SELECT *
2      FROM TRNDTA.F42199
3      WHERE (SLKCO = '42100'
4              AND SLDOC = 24584.000000
5              AND SLDCT = 'PV')
6      ORDER BY SLKCOO ASC,
7               SLDOCO ASC,
8               SLDCTO ASC,
9               SLLNID ASC,
10              SLKCO ASC,
11              SLDOC ASC,
12              SLDCT ASC,
13              SLUPMJ ASC,
14              SLTDAY ASC
```

```
Connected to relational database [redacted]s HA - 110897/QUSER/QZDASOINIT using JDBC configuration 'Default'.

[ 06/05/2020, 02:58:31 PM ]  Run Selected...
SELECT * FROM TRNDTA.F42199 WHERE (SLKCO = '42100' AND SLDOC = 24584.000000 AND SLDCT = 'PV') ORDER BY SLKCOO ASC, SLDOCO ASC, SLDCTO ASC, SLLNID ASC, SL
Statement ran successfully   (28,895 ms  =  28.895 sec)


[ 06/05/2020, 02:59:25 PM ]  Explain...
SELECT * FROM TRNDTA.F42199 WHERE (SLKCO = '42100' AND SLDOC = 24584.000000 AND SLDCT = 'PV') ORDER BY SLKCOO ASC, SLDOCO ASC, SLDCTO ASC, SLLNID ASC, SL

[ 06/05/2020, 03:06:27 PM ]  Explain...
SELECT * FROM TRNDTA.F42199 WHERE (SLKCO = '42100' AND SLDOC = 24584.000000 AND SLDCT = 'PV') ORDER BY SLKCOO ASC, SLDOCO ASC, SLDCTO ASC, SLLNID ASC, SL

[ 06/05/2020, 03:17:21 PM ]  Run Selected...
SELECT * FROM TRNDTA.F42199 WHERE (SLKCO = '42100' AND SLDOC = 24584.000000 AND SLDCT = 'PV') ORDER BY SLKCOO ASC, SLDOCO ASC, SLDCTO ASC, SLLNID ASC, SL
Statement ran successfully  (25,142 ms  =  25.142 sec)


[ 06/05/2020, 03:20:24 PM ]  Run Selected...
SELECT * FROM TRNDTA.F42199 WHERE (SLKCO = '42100' AND SLDOC = 24584.000000 AND SLDCT = 'PV') ORDER BY SLKCOO ASC, SLDOCO ASC, SLDCTO ASC, SLLNID ASC, SL
Statement ran successfully  (152 ms)
```

Visual Explain



Questions?

Thanks