

common

NETHERLANDS

DB2 op IBM i - Ervaringen in de praktijk

[rvhelvoirt@common.nl](mailto:rvhelvoirt@common.nl)

# Agenda

- IBM i Access Client Solutions
- DB2 for i – Services
- Praktijkvoorbeelden:
  - Check inactieve gebruikers
  - PTF overview
  - Inventariseren “deleted” records



# IBM i Access Client Solutions

- Beschikbaar sinds juli 2012
- Laatste update januari 2017 versie 1.1.6.2
- Download locatie:

<http://www-03.ibm.com/systems/power/software/i/access/solutions.html>



IBM Industries & solutions Services Products Support & downloads My

IT infrastructure > Power Systems > Software > IBM i >

## IBM i Access

Overview Client Solutions Windows Linux Web Mo

Overview Support Resources

IBM i Access Client Solutions is the newest member of the IBM i Access family. It provides a Java based, platform-independent interface that runs on most operating systems that support Java, including Linux, Mac, and Windows™.

IBM i Access Client Solutions consolidates the most commonly used tasks for managing your IBM i into one simplified location. The latest version of IBM i Access Client Solutions is available to customers with an IBM i software maintenance contract.

[→ Download IBM i Access Client Solutions base package](#)

[→ QuickStartGuide](#)

[→ GettingStarted](#)

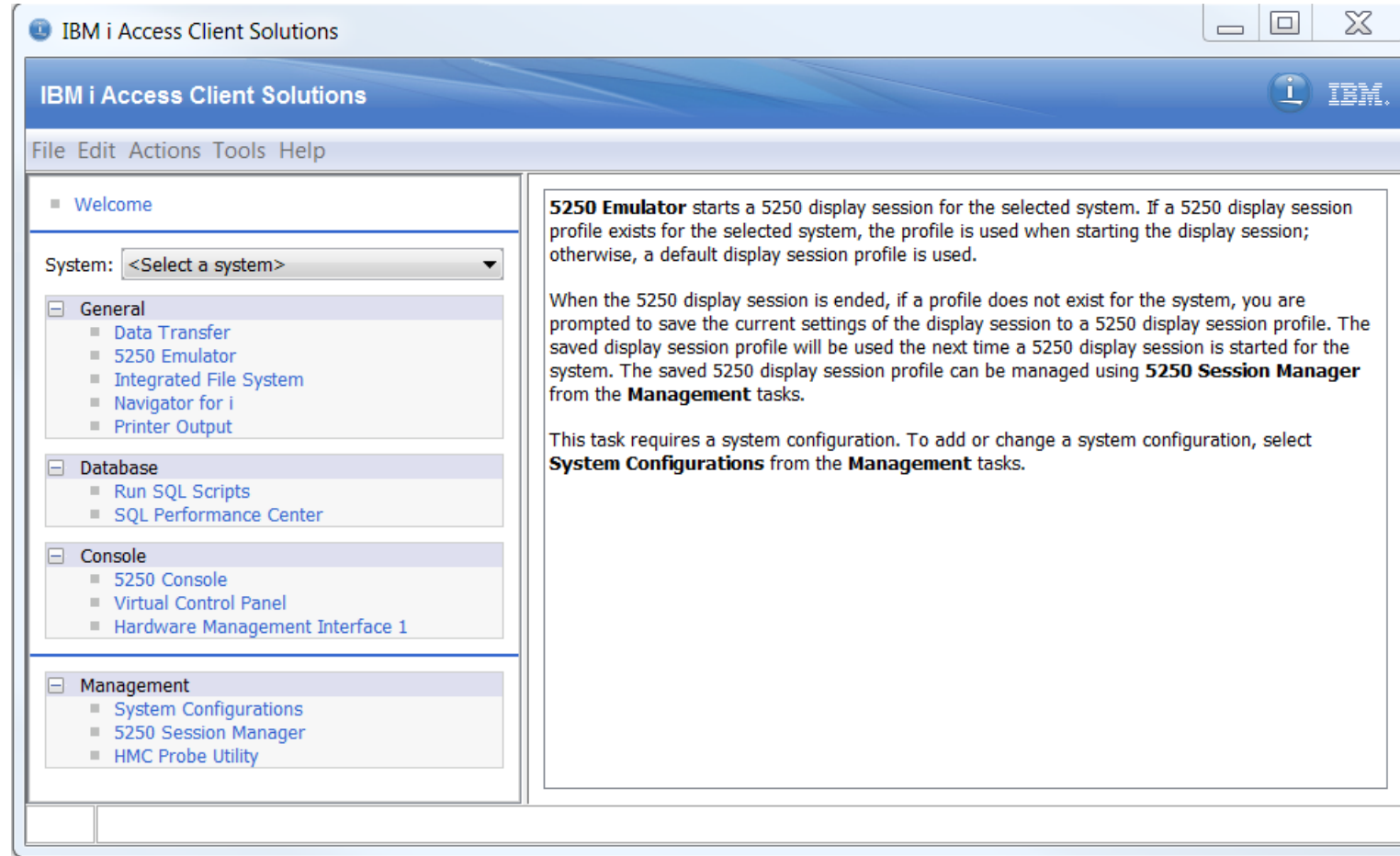
[↓ Updates](#)

### Access videos

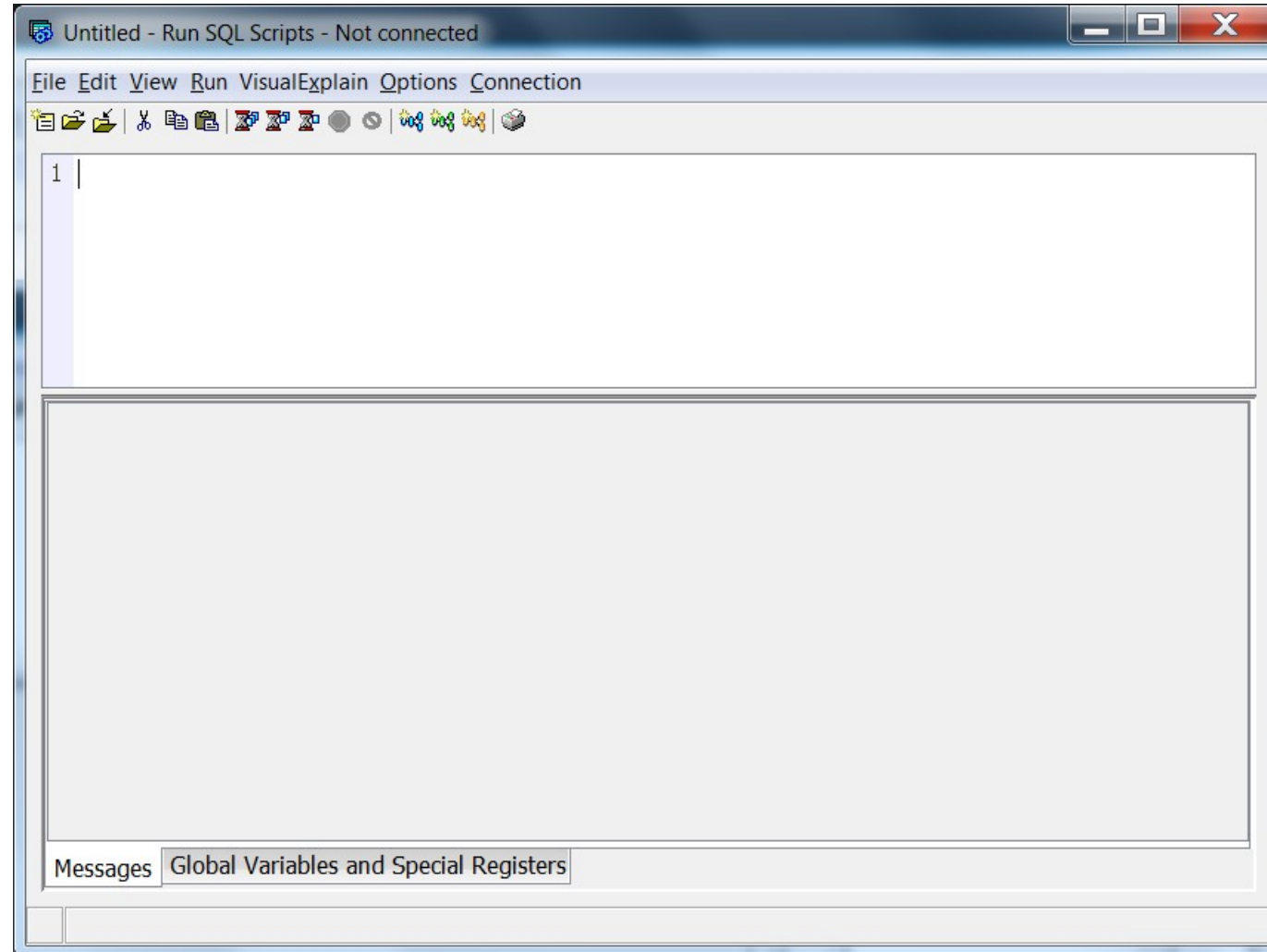
[Introduction to IBM i Access Client Solutions \(00:01:17\)](#)

[All IBM i Access videos](#)

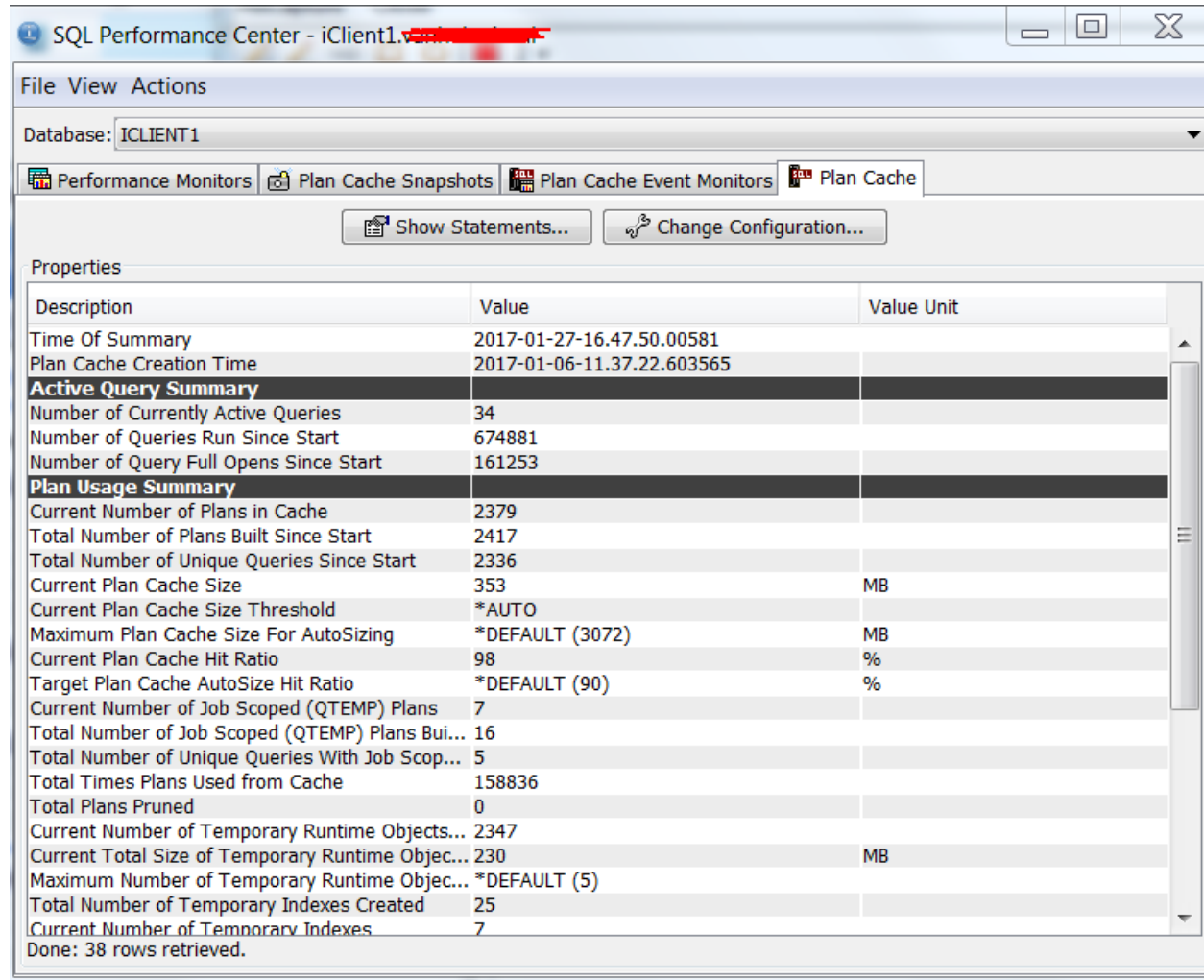
# IBM i Access Client Solutions



# IBM i Access Client Solutions



# IBM i Access Client Solutions



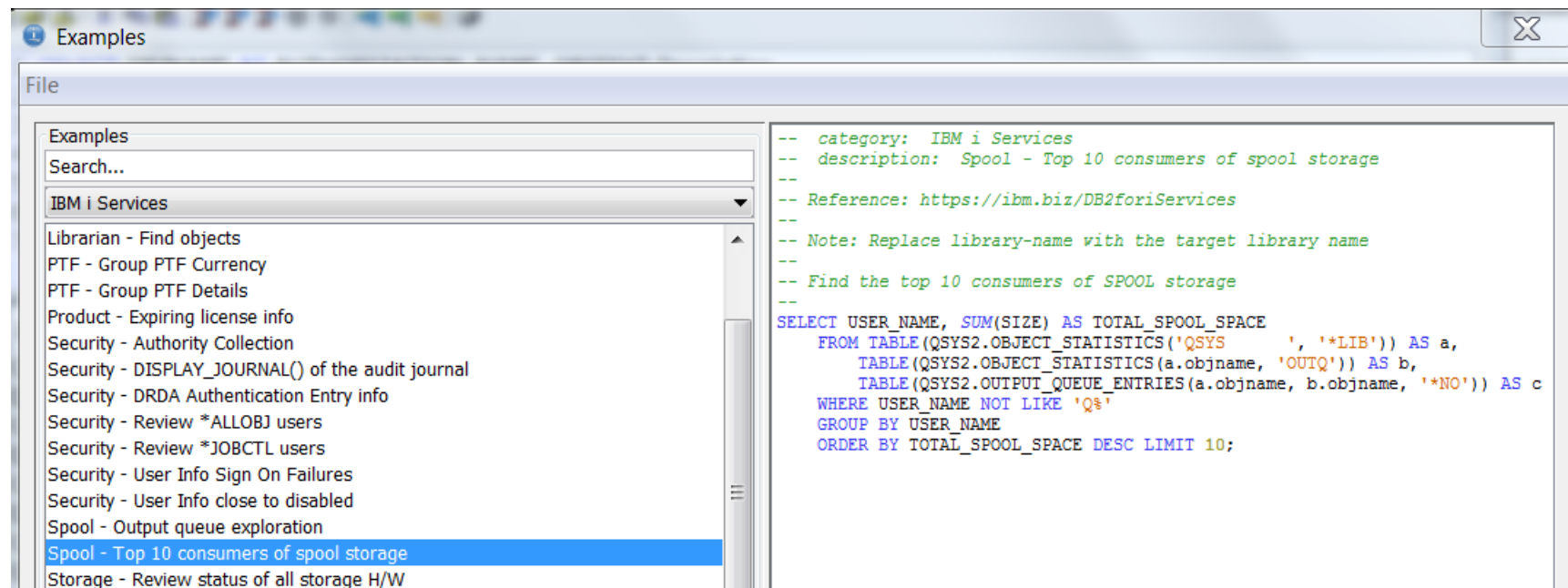
The screenshot displays the 'SQL Performance Center - iClient1' application window. The 'Database' dropdown is set to 'ICLIENT1'. The 'Plan Cache' tab is selected, showing a table of properties. The table has three columns: 'Description', 'Value', and 'Value Unit'. The data is organized into sections: 'Active Query Summary' and 'Plan Usage Summary'. The 'Done: 38 rows retrieved.' message is visible at the bottom of the table.

Description	Value	Value Unit
Time Of Summary	2017-01-27-16.47.50.00581	
Plan Cache Creation Time	2017-01-06-11.37.22.603565	
<b>Active Query Summary</b>		
Number of Currently Active Queries	34	
Number of Queries Run Since Start	674881	
Number of Query Full Opens Since Start	161253	
<b>Plan Usage Summary</b>		
Current Number of Plans in Cache	2379	
Total Number of Plans Built Since Start	2417	
Total Number of Unique Queries Since Start	2336	
Current Plan Cache Size	353	MB
Current Plan Cache Size Threshold	*AUTO	
Maximum Plan Cache Size For AutoSizing	*DEFAULT (3072)	MB
Current Plan Cache Hit Ratio	98	%
Target Plan Cache AutoSize Hit Ratio	*DEFAULT (90)	%
Current Number of Job Scoped (QTEMP) Plans	7	
Total Number of Job Scoped (QTEMP) Plans Bui...	16	
Total Number of Unique Queries With Job Scop...	5	
Total Times Plans Used from Cache	158836	
Total Plans Pruned	0	
Current Number of Temporary Runtime Objects...	2347	
Current Total Size of Temporary Runtime Objec...	230	MB
Maximum Number of Temporary Runtime Objec...	*DEFAULT (5)	
Total Number of Temporary Indexes Created	25	
Current Number of Temporary Indexes	7	

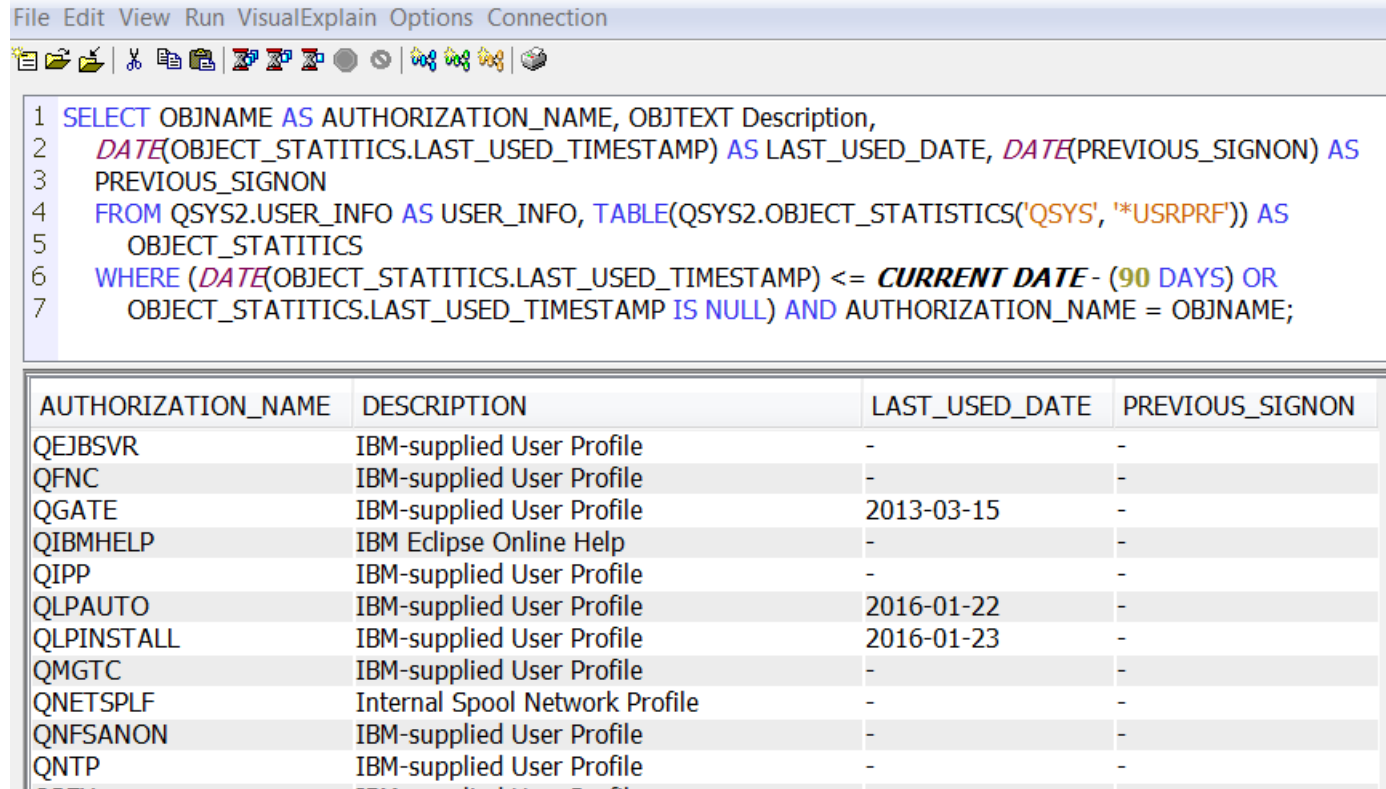
Done: 38 rows retrieved.

# DB2 for i – Services

- SQL is de standaard makkelijker dan API's
- [DB2 for i – Services documentation link](#)
- DB2 for i – Services examples: Menu => Edit => Insert from Examples



# Praktijkvoorbeeld - Check inactieve gebruikers



The screenshot shows a database query tool interface. At the top, there is a menu bar with 'File Edit View Run Visual Explain Options Connection' and a toolbar with various icons. Below the toolbar, an SQL query is displayed in a text area, numbered 1 through 7. The query selects user information from a table, filtering for users who have not signed on in the last 90 days or whose last used timestamp is null, and where the authorization name matches the object name.

```
1 SELECT OBJNAME AS AUTHORIZATION_NAME, OBJTEXT Description,  
2     DATE(OBJECT_STATITICS.LAST_USED_TIMESTAMP) AS LAST_USED_DATE, DATE(PREVIOUS_SIGNON) AS  
3     PREVIOUS_SIGNON  
4 FROM QSYS2.USER_INFO AS USER_INFO, TABLE(QSYS2.OBJECT_STATISTICS('QSYS', '*USRPRF')) AS  
5     OBJECT_STATITICS  
6 WHERE (DATE(OBJECT_STATITICS.LAST_USED_TIMESTAMP) <= CURRENT DATE - (90 DAYS) OR  
7     OBJECT_STATITICS.LAST_USED_TIMESTAMP IS NULL) AND AUTHORIZATION_NAME = OBJNAME;
```

Below the query, a table displays the results of the query. The table has four columns: AUTHORIZATION\_NAME, DESCRIPTION, LAST\_USED\_DATE, and PREVIOUS\_SIGNON. The results list various user profiles and their last used dates.

AUTHORIZATION_NAME	DESCRIPTION	LAST_USED_DATE	PREVIOUS_SIGNON
QEJBSVR	IBM-supplied User Profile	-	-
QFNC	IBM-supplied User Profile	-	-
QGATE	IBM-supplied User Profile	2013-03-15	-
QIBMHELP	IBM Eclipse Online Help	-	-
QIPP	IBM-supplied User Profile	-	-
QLPAUTO	IBM-supplied User Profile	2016-01-22	-
QLPINSTALL	IBM-supplied User Profile	2016-01-23	-
QMGTC	IBM-supplied User Profile	-	-
QNETSPLF	Internal Spool Network Profile	-	-
QNFSANON	IBM-supplied User Profile	-	-
QNTP	IBM-supplied User Profile	-	-



# Praktijkvoorbeeld - PTF overview

```

16 --
17 -- If using IBM i 7.2
18 --
19 SELECT *
20 FROM SYSTOOLS.GROUP_PTF_CURRENCY
21 WHERE PTF_GROUP_RELEASE = 'R720'
22 ORDER BY PTF_GROUP_LEVEL_AVAILABLE - PTF_GROUP_LEVEL_INSTALLED DESC;
23

```

PTF_GROUP_CURRENCY	PTF_GROUP_ID	PTF_GROUP_TITLE	PTF_GROUP_LEVEL_INSTALLED	PTF_GROUP_LEVEL_AVAILABLE	PTF_GROUP_LAST_UPDATED_BY_IBM
UPDATE AVAILABLE	SF99719	SF99719 720 Group Hiper	77	82	01/24/2017
UPDATE AVAILABLE	SF99775	SF99775 720 Hardware and Related PTFs	19	22	01/23/2017
UPDATE AVAILABLE	SF99223	SF99223 720 IBM Open Source Solutions for i	1	3	01/19/2017
UPDATE AVAILABLE	SF99713	SF99713 720 IBM HTTP Server for i	17	19	01/23/2017
UPDATE AVAILABLE	SF99718	SF99718 720 Group Security	39	41	12/13/2016
UPDATE AVAILABLE	SF99251	SF99251 720 Content Manager OnDemand f...	3	4	01/09/2017
UPDATE AVAILABLE	SF99334	SF99334 720 DB2 Web Query for i V2.2.0	1	2	01/10/2017
UPDATE AVAILABLE	SF99481	SF99481 720 WebSphere App Server V8.5	9	10	12/22/2016
UPDATE AVAILABLE	SF99658	SF99658 720 DB2 Web Query for i V2.1.1	4	5	12/29/2016
UPDATE AVAILABLE	SF99702	SF99702 720 DB2 for IBM i	14	15	01/19/2017
UPDATE AVAILABLE	SF99715	SF99715 720 Backup Recovery Solutions	28	29	12/22/2016
UPDATE AVAILABLE	SF99747	SF99747 720 DB2 Web Query for i V2.1.0	15	16	12/29/2016
UPDATE AVAILABLE	SF99759	SF99759 720 IBM MQ for IBM i - v7.1.0/v8.0.0	7	8	01/19/2017
INSTALLED LEVEL IS CU...	SF99480	720 WebSphere App Server V8.0	6	6	02/01/2016
INSTALLED LEVEL IS CU...	SF99714	SF99714 720 Performance Tools	4	4	05/17/2016
INSTALLED LEVEL IS CU...	SF99716	SF99716 720 Java	11	11	09/05/2016
INSTALLED LEVEL IS CU...	SF99717	SF99717 720 Technology Refresh	5	5	11/10/2016
INSTALLED LEVEL IS CU...	SF99720	Current Cumulative PTF Media Documentation	16306	16306	11/10/2016

# Praktijkvoorbeeld - Inventariseren “deleted” records

Gebaseerd op het artikel op IBM i Developerworks

“Are deleted rows wasting resources on your IBM i system?”

A task for database engineers on IBM DB2 for i

Figure 1 – Example output from step 1 query

FILENAME	MEMBER	LIBRARY	Number Non- Deleted Rows	Number Deleted Rows	Reusing Deleted Rows	Percent Deleted	SIZE	Deleted Space
DELTEST	DELTEST	MCKINLEY2	3,700,000	4,300,000	Y	53	392,011,776	210,706,329
ITEM_FACT	ITEM_FACT	PURGETEST	3,568,632	2,432,583	Y	40	1,612,763,136	653,730,984
QAPZREQ	QAPZREQ	QUSRSYS	44,236	121,803	N	73	16,781,312	12,310,446
QAPZSYM	QAPZSYM	QUSRSYS	43,824	118,801	N	73	50,335,744	36,771,324
SELECT1P	TEST1	MCKINLEY2	550,000	50,000	Y	8	7,221,248	601,770
QAPZGRP	QAPZGRP	QUSRSYS	8,212	2,389	N	22	3,690,496	831,675
QAPZPTF	QAPZPTF	QUSRSYS	11,308	361	N	3	1,581,056	48,912
QAPMCCCNAT	QAPMCCCNAT	QUSRSYS	37	299	Y	88	217,088	193,182
CLASSDBMON	CLASSDBMON	DENTON	430,959	256	Y	0	1,122,041,856	666,124
KENTJOB	KENTJOB	DENTON	271	126	Y	31	3,309,568	1,050,391
QZG0000777	KENTJOB	DBQTEAM03	271	126	Y	31	3,145,728	998,392
QZG0000777	KENTJOB	DBQTEAM04	271	126	Y	31	3,145,728	998,392
QZG0000777	KENTJOB	DBQTEAM05	271	126	Y	31	3,145,728	998,392

# Praktijkvoorbeeld - Inventariseren “deleted” records

```
File Edit View Run VisualExplain Options Connection
1  /*****
2  /* Create the DB2 tables with physical files to investigate the number of deleted rows */
3  /*****
4
5  CL:DSPFD FILE(*ALLUSR/*ALL) TYPE(*MBR) OUTPUT(*OUTFILE) FILEATR(*PF)
6  OUTFILE(QRPLOBJ/DSPFD_MBR) OUTMBR(*FIRST *ADD);
7
8  CL:DSPFD FILE(*ALLUSR/*ALL) TYPE(*ATR) OUTPUT(*OUTFILE) FILEATR(*PF)
9  OUTFILE(QRPLOBJ/DSPFD_ATR) OUTMBR(*FIRST *ADD);
10
11 /*****
12 /* Create indexes over these two files to improve the performance of your analysis */
13 /*****
14
15 CREATE INDEX QRPLOBJ.DSPFD_ATR_IX ON QRPLOBJ.DSPFD_ATR(PHFILE);
16 CREATE INDEX QRPLOBJ.DSPFD_MBR_IX
17   ON QRPLOBJ.DSPFD_MBR(MBFTYP, MBFILE, MBNDTR);
18
19 /*****
20 /* Stored procedure dumps SQL plan cache entries to a table */
21 /*****
22
23 CALL QSYS2.DUMP_PLAN_CACHE('QRPLOBJ', 'PCSS');
24
```

# Praktijkvoorbeeld - Inventariseren “deleted” records

```
25 /*****  
26 /* Step 1 Identify tables that have a large number of deleted rows */  
27 /*****  
28 SELECT DISTINCT F.MBFILE AS FILENAME, F.MBNAME AS Member, MBLIB AS LIBRARY, MBNRCD AS  
29 "Number Non-Deleted Rows", MBNDTR AS "Number Deleted Rows", PHRUSE AS "Reusing Deleted Rows",  
30 BIGINT(mbndtr /  
31 (  
32 CASE  
33   WHEN mbnrcd + mbndtr = 0  
34     THEN CAST(1 AS BIGINT)  
35     ELSE CAST(mbnrcd + mbndtr AS BIGINT)  
36 END)  
37   * 100) AS "Percent Deleted",  
38   CASE  
39     WHEN MBDSSZ > 0  
40       THEN CAST(MBDSSZ AS BIGINT)  
41       ELSE CAST(MBDSZ2 AS BIGINT)  
42 END AS SIZE, BIGINT(CASE WHEN MBDSSZ > 0 THEN CAST(MBDSSZ AS BIGINT) ELSE  
43   CAST(MBDSZ2 AS BIGINT) END * (CAST(mbndtr AS BIGINT) /  
44   (  
45 CASE  
46   WHEN mbnrcd + mbndtr = 0  
47     THEN CAST(1 AS BIGINT)  
48     ELSE CAST(mbnrcd + mbndtr AS BIGINT)  
49 END)  
50 )) AS "Deleted Space" FROM QRPLOBJ.DSPFD_MBR AS F JOIN QRPLOBJ.DSPFD_ATR AS A ON F.MBFILE =  
51   A.PHFILE WHERE MBFTYP = 'P' AND MBNDTR > 10000 ORDER BY MBNDTR DESC  
52   FETCH FIRST 25 ROWS ONLY;  
53  
54 .....
```


## Praktijkvoorbeeld - Inventariseren “deleted” records

```
53
54 /*****
55 /* Step 2 Look for SQL statements that are wasting resources */
56 /*****
57 WITH bigdel
58   AS (SELECT DISTINCT mbfile AS FILE, mbname AS Member, mblib AS Library, mbnrcd AS
59        NumberRecords, mbdtr AS NumberDeletedRecords, phruse AS ReuseDeleted
60        FROM qrplobj.DSPFD_MBR F
61        JOIN qrplobj.DSPFD_ATR a ON f.MBFILE = a.PHFILE
62        WHERE mbftyp = 'P' AND mbdtr > 10000)
63 -- Join table scan info from the plan cache
64 SELECT QQTln
65 AS LIBrary, QQTfn
66 AS "Table Name", D.REUSEDELETED
67 AS "Reusing Deleted Rows", MAX(QQTOTR)
68 AS "Number Non-Deleted Rows", MAX(d.numberDeletedRecords)
69 AS "Number Deleted Rows", SUM(d.numberDeletedRecords)
70 AS "Total Deleted Rows Scanned", COUNT(*)
71 AS TotalScans FROM QRPLOBJ.PCSS M JOIN bigdel d ON d.library = m.qqtln AND d.file = m.qqtfn
72 AND d.Member = m.qqtmn WHERE qqrid = 3000 AND QQC11 <> 'Y' GROUP BY qqtln, qqtfn,
73 d.reusedeleted ORDER BY SUM(d.numberDeletedRecords) DESC FETCH FIRST 25 ROWS ONLY
74 OPTIMIZE FOR ALL ROWS;
75
```

## Praktijkvoorbeeld - Inventariseren “deleted” records

```
76 /*****  
77 /* Step 3 Look at highly accessed tables */  
78 /*****  
79 WITH bigdel  
80 AS (SELECT mbfile, mblib, F.MBNAME AS Member, mbftyp, MBNRCD AS NumberRecords, MBNDTR AS  
81     numberDeletedRecords, PHRUSE AS REUSEDELETED, MBDSSZ AS MBRSIZE  
82     FROM QRPLOBJ.DSPFD_MBR F  
83     JOIN QRPLOBJ.DSPFD_ATR a ON f.MBFILE = a.PHFILE  
84     WHERE MBFTYP = 'P' AND mbndtr > 10000)  
85 -- Join Index builds to the tables with large number of deleted rows  
86 SELECT QQTln  
87 AS Library, QQTfn  
88 AS "Table Name", D.REUSEDELETED  
89 AS "Reusing Deleted Rows", MAX(QQTOTR)  
90 AS "Non-Deleted Rows", MAX(d.numberDeletedRecords)  
91 AS "Deleted Rows", SUM(d.numberDeletedRecords)  
92 AS "Total Deleted Rows Scanned", SUM(M.QQRIDX)  
93 AS "Total index entries created", QQIDX  
94 AS Index_Advised_Columns, SUM(CASE WHEN QQC16 = 'N' THEN qqi6 ELSE 0 END)  
95 AS total_keys_built, SUM(CASE WHEN QQC16 = 'N' THEN 1 ELSE 0 END) as indexCreated, SUM(CASE  
96     WHEN QQC16 = 'Y' THEN 1 ELSE 0 END) as indexreused, COUNT(*)  
97 AS TotalIXsCreated FROM QRPLOBJ.PCSS M JOIN bigdel d ON d.mblib = m.qqtln AND d.mbfile =  
98     m.qqtfn AND d.Member = m.qqtmn WHERE qqrid = 3002 GROUP BY qqtln, qqtfn, qqtmn,  
99     d.REUSEDELETED, m.QQRCOD, QQIDX ORDER BY SUM(d.numberDeletedRecords) DESC FETCH FIRST 25  
100 ROWS ONLY OPTIMIZE FOR ALL ROWS;  
101
```

# Praktijkvoorbeeld - Inventariseren “deleted” records

```
102 /*****  
103 /* Step 4  Fix the problematic tables */  
104 /*****  
105 /* Look at highly accessed tables */  
106 SELECT table_schema, TABLE_NAME, Data_size, Number_Deleted_Rows, Logical_Reads, Physical_reads,  
107    Sequential_reads, Random_reads  
108    FROM qsys2.systablestat  
109    ORDER BY Logical_reads DESC  
110    FETCH FIRST 25 ROWS ONLY
```

## Praktijkvoorbeeld - Inventariseren “deleted” records

TABLE_SCHEMA	TABLE_NAME	DATA_SIZE	NUMBER_DELETED_ROWS	LOGICAL_READS	PHYSICAL_READS	SEQUENTIAL_READS	RANDOM_READS
QUSRBRM	QA1ALI2	22297612288	3260176	394471138	4411768	394471138	0
QSYS2	SYSLIMTBL	15347712	2844	189214240	30945	189214240	0
QUSRISJ	QAIJS4CM	15106048	2265	154733512	47681	0	0
QPFRDATA	QAPMSMSYS	12242944	0	118544901	2643	118544901	0
PCSECDTA	TRAPOD	2168590336	0	82922315	189218	81077346	226558
QUSRBRM	QA1ADI2	1595006976	224709	56149005	1029078	30265173	25883832
QNAVSRV	QINAVMNTRG	688128	953	34524196	85	34352403	171793
QUSRSYS	QA1PJ1	794624	0	31211691	802	0	0
QPFRDATA	QAPMJOBMI	10907090944	0	28207647	371367	18448384	9759263
QUSRBRM	QA1AOR	126455808	253	20726266	39276	20726266	0
QUSRBRM	QA1AOD	488808448	125100	19627078	245486	1926920	17657375
QPFRDATA	QAPMJOBOS	2802761728	0	19296761	10324	81920	19214841
QPFRDATA	QAPMSMSYS	737280	0	14797850	120	14797850	0
QUSRSYS	QAPZGRP	1290240	797	12024795	896	0	0
QUSRBRM	QA1AMB	336707584	71522	10138672	55778	10138672	0
QUSRSYS	QA1PSBS	12288	0	9978840	0	0	0
QUSRBRM	QA1AHS	15458304	1797	9879320	35445	4735868	2243340
QUSRSYS	QA1PONE	23068672	0	8839073	35573	684871	0
PR_MAILRCV	VHA_DISKUSG	6385664	0	6113930	2564	6113930	0
QPFRHIST	QAPMHDJOBM	253849600	308557	5686783	552	0	5686783
QUSRBRM	QA1AOQ	169930752	152950	5491411	31631	5491411	0
QRPLOBJ	QAEZD0001D	679936	0	4295211	3	4086420	208791
PR_MAILRCV	VHA_CPUUSG	4284416	0	4095444	1984	4095444	0
OPFRHIST	OAPMHDJOB0	144764928	309133	3339756	425	0	3339756



# Praktijkvoorbeeld - Inventariseren "deleted" records

Reorganize DB\_TEST."BLOB Example" - Localhost(192.168.1.10)

Reorganize DB\_TEST."BLOB Example" - Localhost(192.168.1.10)

**Reorganize the table**

By compressing out deleted rows

Preserve arrival row sequence

By selected index

[Empty]

Table partition: First partition

Reorganize starting at: First row

Allow reorganization to be suspended

Allow users to access the table during reorganization (Online)

Allow changes to the table during the reorganization

Access paths: Maintain during

Number of tasks: Use current setting

Show Command OK Cancel

```
File Edit View Run Visual Explain Options Connection
1 /* Reorganize the table */
2 CL:RGZPFM FILE(DB_TEST / BLOB0001) MBR(* FIRST) KEYFILE(* RPLDLTRCD) ALWCANCEL(* YES) LOCK(*
3 SHRUPD) RBDACCPH(* NO) FROMRCD(* START);
```

