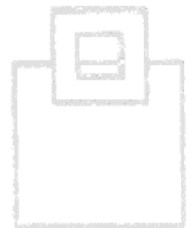
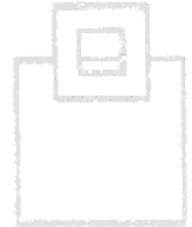


---

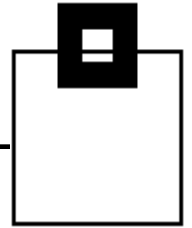
## Utility Support

**Roy Boxwell**  
***SOFTWARE ENGINEERING GmbH***  
**&**  
***SEGUS Inc***

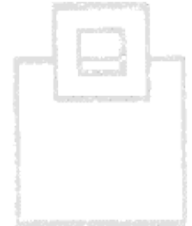
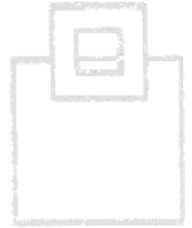


# AGENDA

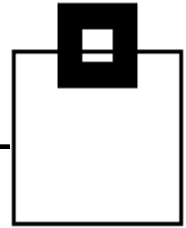
---



1. WLX – How it works
2. Quiet Times – Is a table in use?
3. Reorg Suppression and Detection
4. Cluster Index – Review the choice
5. Utility review – What ran when and did what?
6. Q&A Session

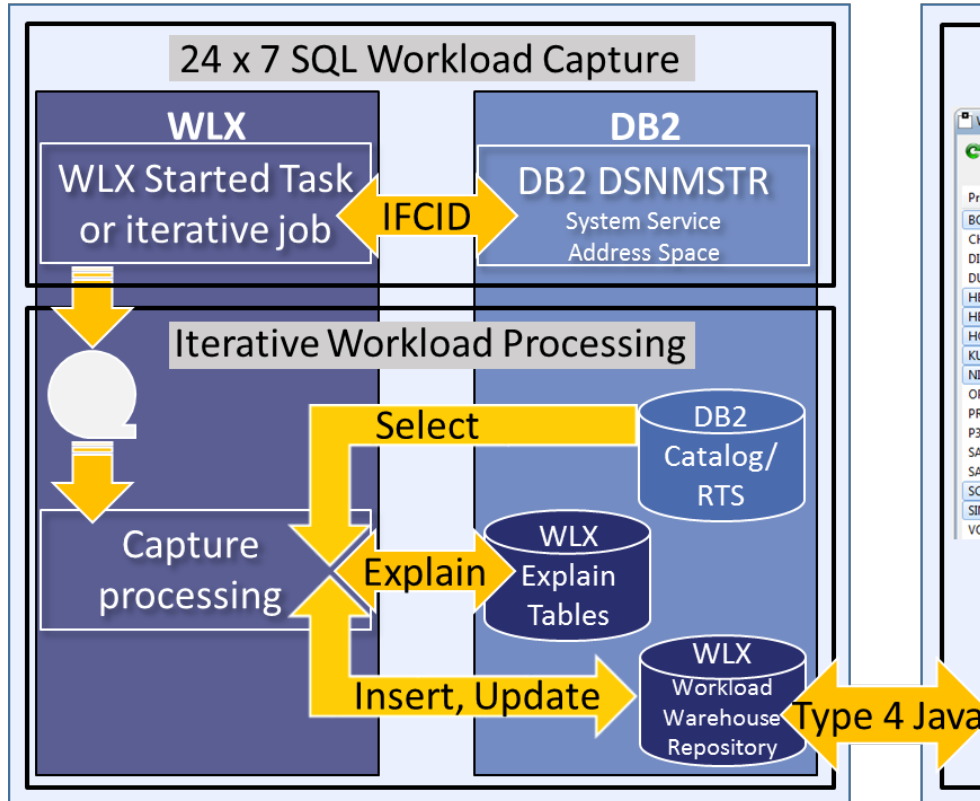


# WLX - How it works

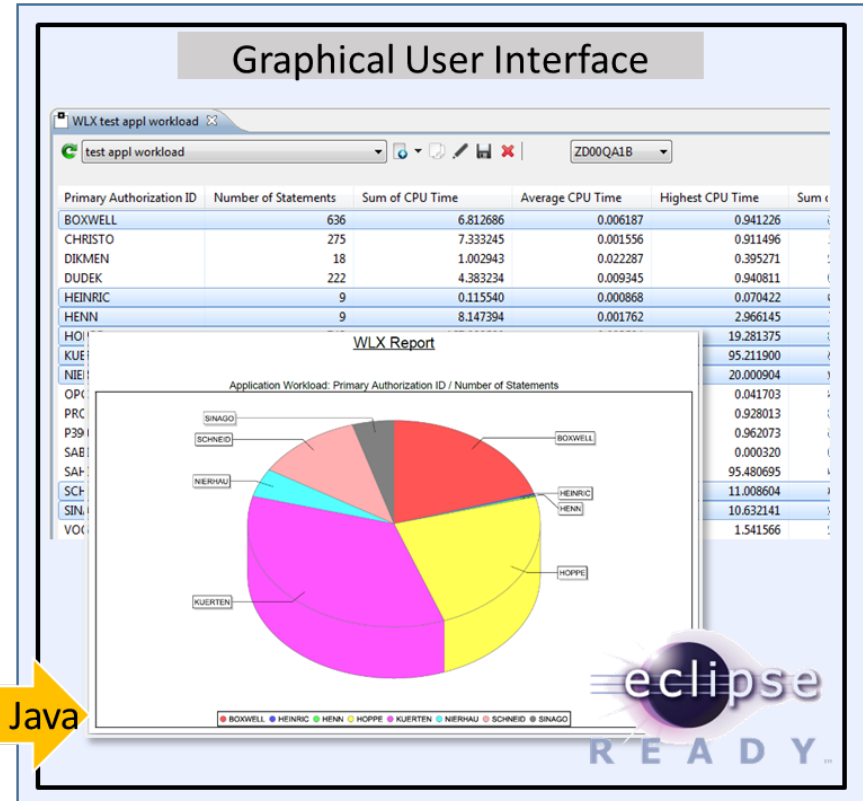


Captures the hard-to-get SQLs,  
even the ones that disappear ...

Mainframe Engine

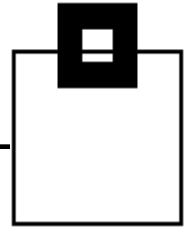


Workstation Engine

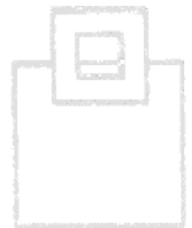
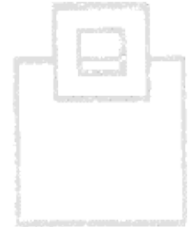
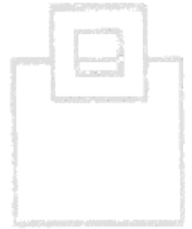


# AGENDA

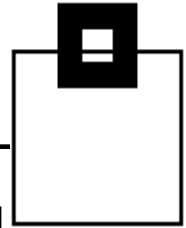
---



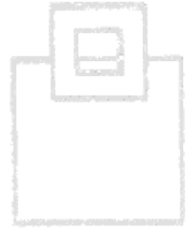
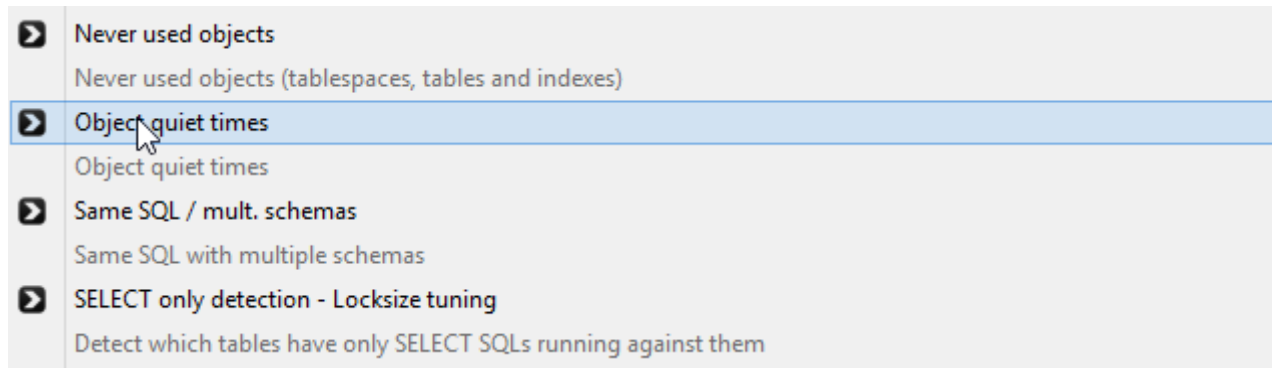
1. WLX – How it works
2. Quiet Times – Is a table in use?
3. Reorg Suppression and Detection
4. Cluster Index – Review the choice
5. Utility review – What ran when and did what?
6. Q&A Session



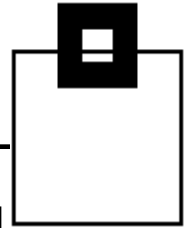
# Quiet Times



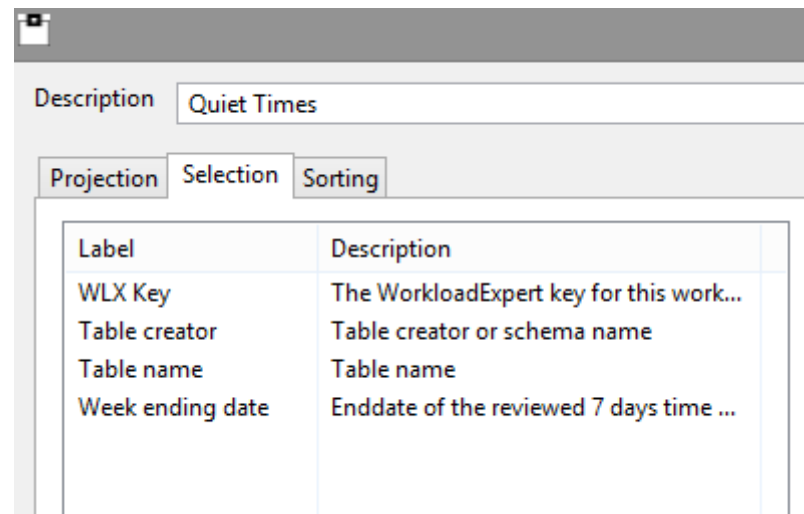
Question: Do you know how often tables are “in use” and so cannot easily be ALTERed or REORGed?



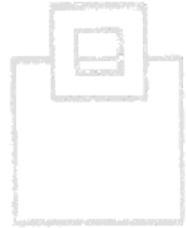
# Quiet Times



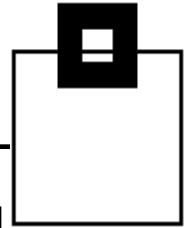
Question: Do you know how often tables are “in use” and so cannot easily be ALTERed or REORGed?



Label	Description
WLX Key	The WorkloadExpert key for this work...
Table creator	Table creator or schema name
Table name	Table name
Week ending date	Enddate of the reviewed 7 days time ...

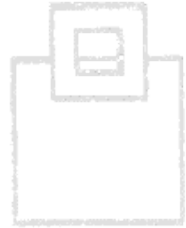


# Quiet Times



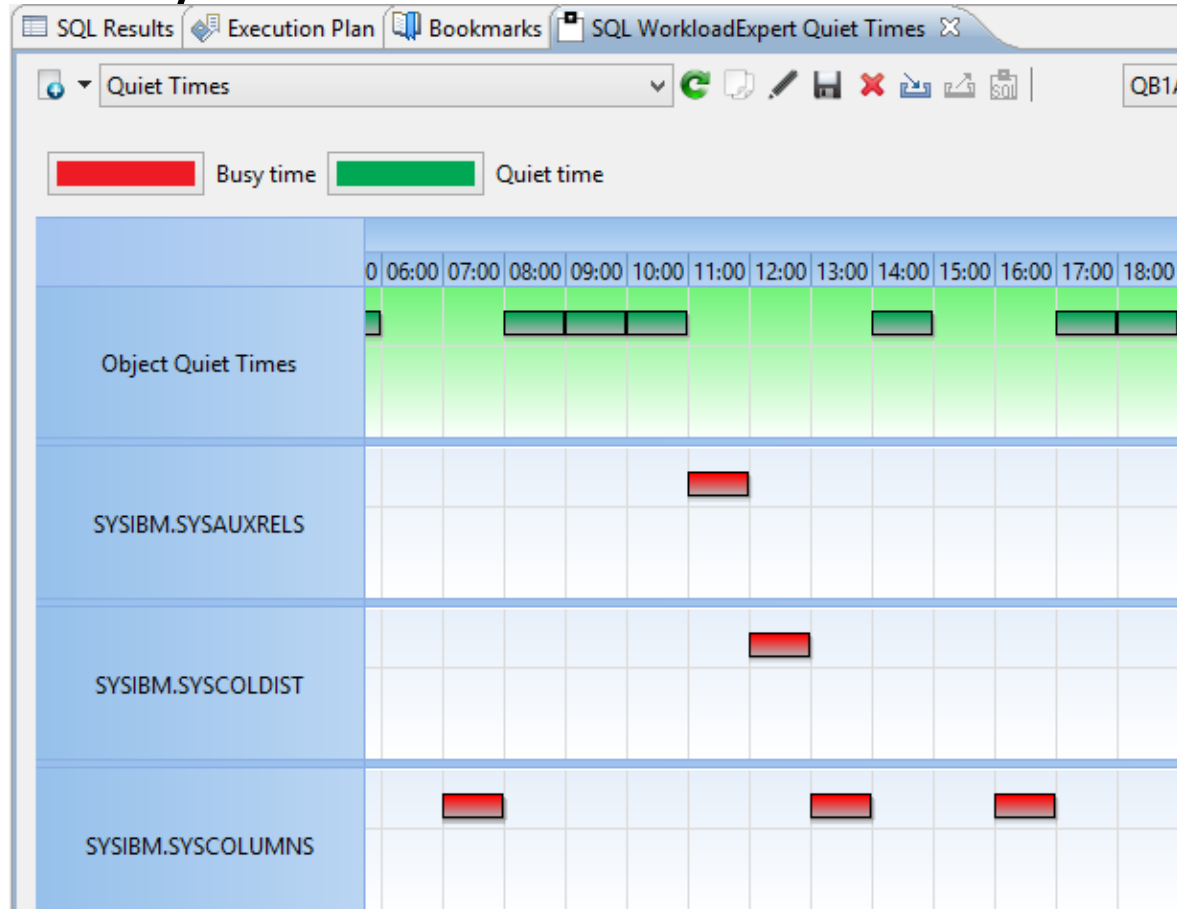
Question: Do you know how often tables are “in use” and so cannot easily be ALTERed or REORGed?

Label	Operator	Value	Description
WLX Key	=	newest	The WorkloadExpert key for this wor...
Table creator	=	SYSIBM	Table creator or schema name
Table name	LIKE	SYS%	Table name
Week ending date	=	2015-03-03-14...	Enddate of the reviewed 7 days time...



# Quiet Times

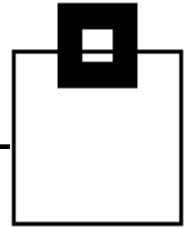
Question: Do you know how often tables are "in use" and so cannot easily be ALTERed or REORGed?



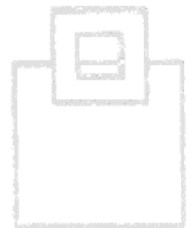
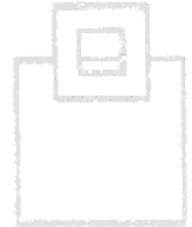


# AGENDA

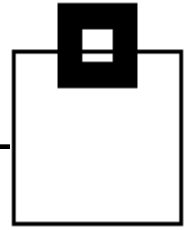
---



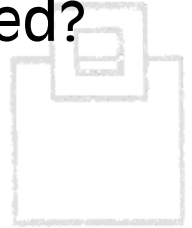
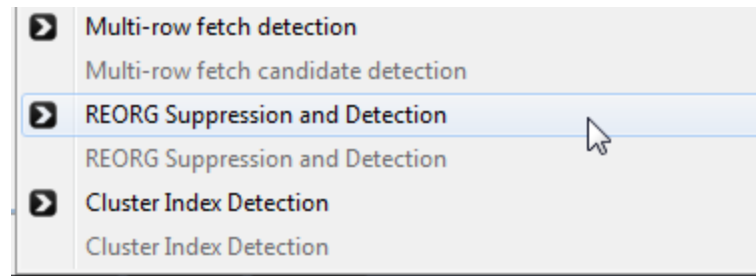
1. WLX – How it works
2. Quiet Times – Is a table in use?
- 3. Reorg Suppression and Detection**
4. Cluster Index – Review the choice
5. Utility review – What ran when and did what?
6. Q&A Session



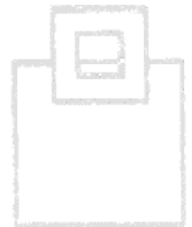
# Reorg Suppression and Detection



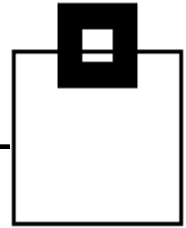
Now we will look at a nice way to review your REORG strategy. Do you *\*know\** if the REORGs you are faithfully doing every day, week, or month are really actually needed? Or are you doing them “Just for fun?”



First select the REORG Suppression and Detection use case

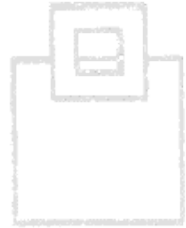


# Reorg Suppression and Detection



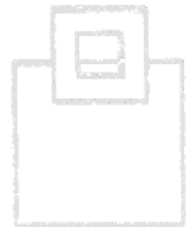
First select one or more objects from the Object View or directly in the Use case (like here):

Label	Operator	Value	Description
Database name	=	MVNXTEST	Database name
Tablespace name	=	MVNXS80	tablespace name
Object type	=	T	T=Table, I=Index, M=Materialized q...



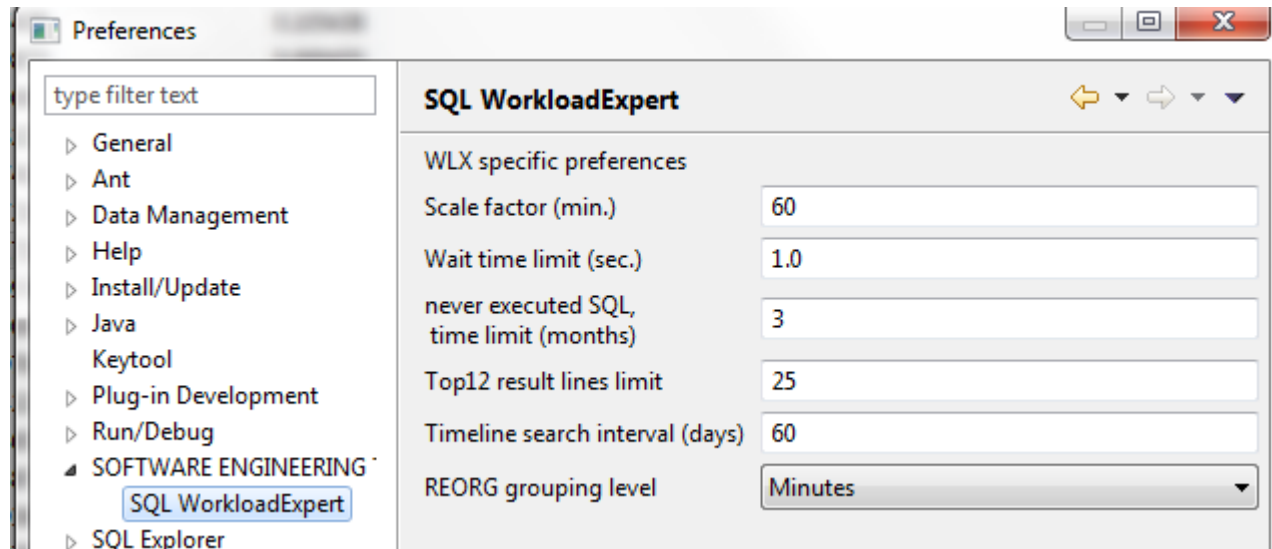
Then select the “columns of interest”:

Label	Description
Timestamp	Statement insert/update or REORG t...
Executions ...	The number of Executions
Elapsed Time ...	The elapsed time in microseconds
Average elapse...	Average elapsed time per hour
CPU Time ...	The CPU time in microseconds
Average CPU ti...	Average CPU time per hour

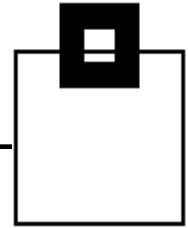


# Reorg Suppression and Detection

One important point is that this Use Case gets data from multiple sources (E.g. REORGs from the SYSCOPY and/or the WLX Utility Table) and merges the result into sensible looking tabular data. To get the time axis you can decide which “level” of time to group to – Days (Default), Hours, or Minutes:



# Reorg Suppression and Detection

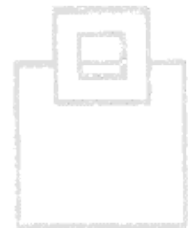
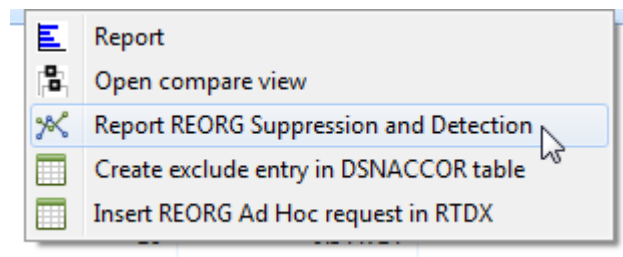


Then we see the tabular result first (Here grouped at Minutes):

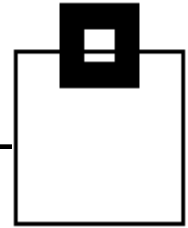
The screenshot shows the SQL WorkloadExpert interface with a table of REORG statistics. The table has the following columns: REORG, Timestamp, Executions, Elapsed Time, Average elapsed time per hour, CPU Time, and Average CPU time per hour. The data is grouped by REORG and timestamp.

REORG	Timestamp	Executions	Elapsed Time	Average elapsed time per hour	CPU Time	Average CPU time per hour
	2015-03-06-08.37.00.000000	8,932	3.005453	0.000336	1.972212	0.000220
REORG	2015-03-06-09.01.00.000000	0	0.000000	0.000000	0.000000	0.000000
	2015-03-06-09.59.00.000000	136	0.312686	0.002299	0.061538	0.000452
	2015-03-06-10.01.00.000000	1	0.038081	0.038081	0.036476	0.036476
	2015-03-06-10.04.00.000000	1	0.843276	0.843276	0.654787	0.654787
	2015-03-06-10.10.00.000000	1	0.004053	0.004053	0.003721	0.003721
REORG	2015-03-06-10.15.00.000000	0	0.000000	0.000000	0.000000	0.000000
	2015-03-06-11.08.00.000000	3.285	76.223428	0.023203	65.314139	0.019882

Right-click and select the special report:

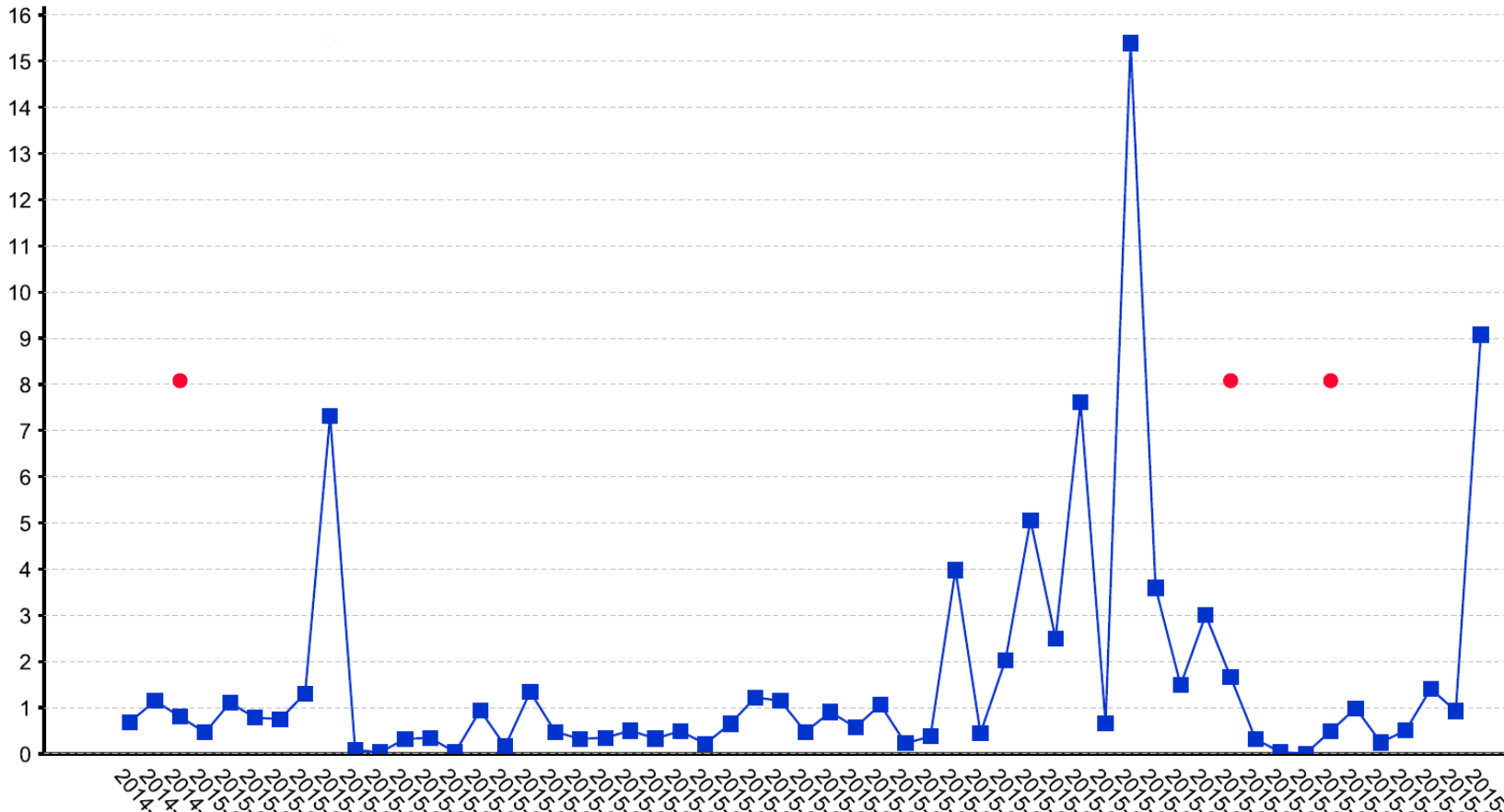


# Reorg Suppression and Detection

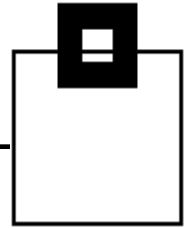


Now we see the various reports, first is elapsed:  
WLX Report

REORG Suppression and Detection



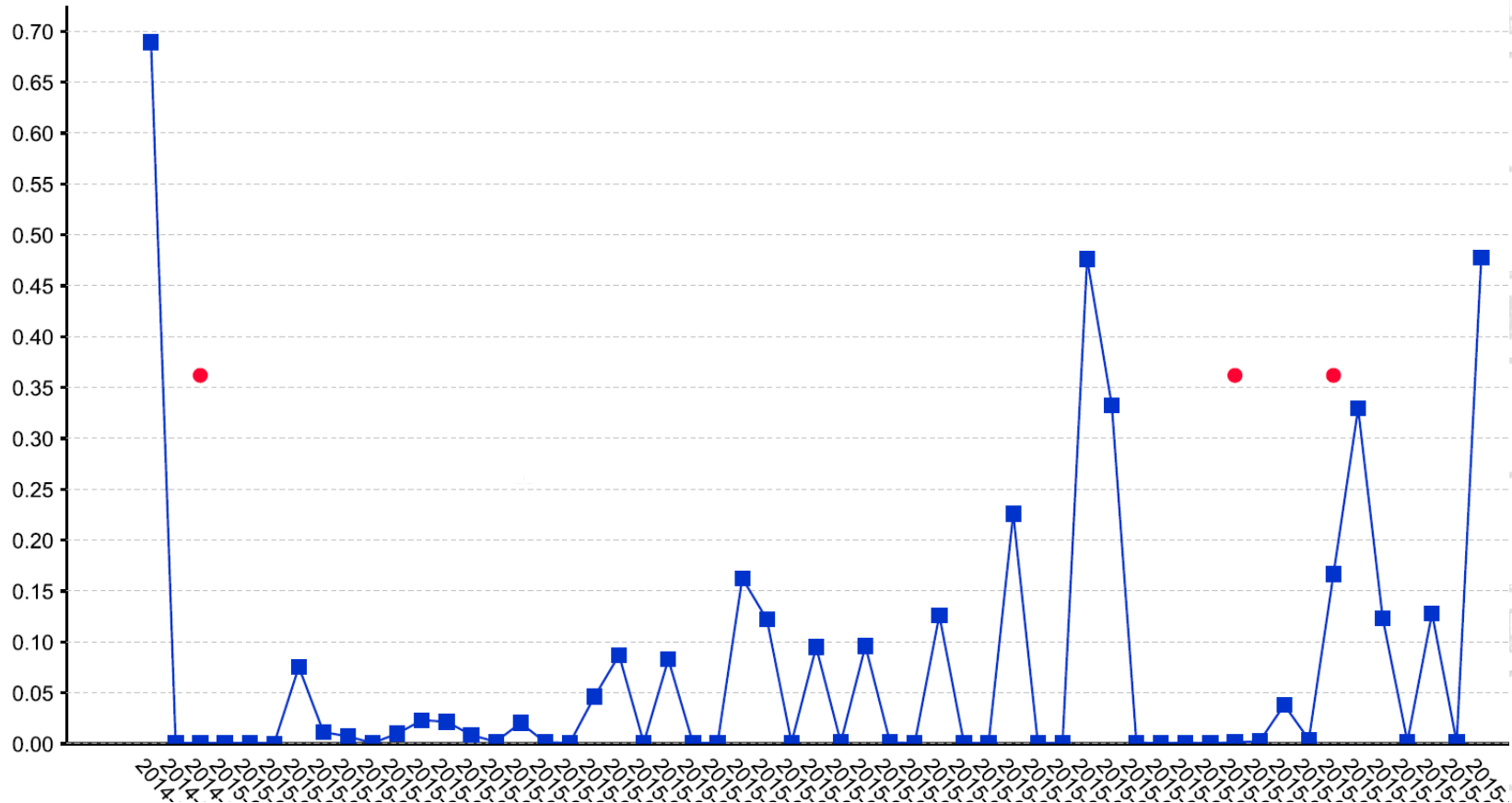
# Reorg Suppression and Detection



Then CPU:

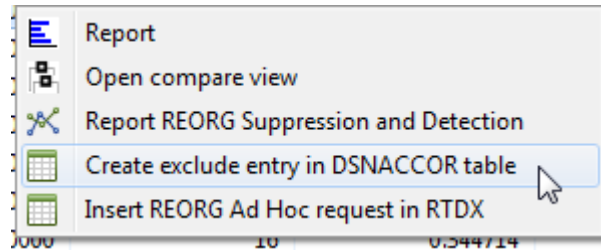
## WLX Report

REORG Suppression and Detection



# Reorg Suppression and Detection

Now back on the tabular data you can do a right-click and see the other choices you have:

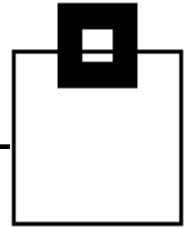


The Create exclude entry does an insert into the DSNACC.EXCEPT\_TBL table which is generally used to restrict Utilities.

The Insert REORG entry is for our own  RealTime DBAExpert utility generator to generate a REORG the next time it runs.



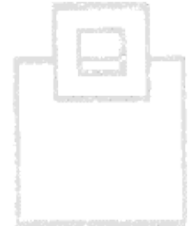
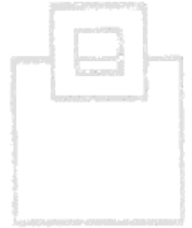
# Reorg Suppression and Detection



Having done the Create exclude entry you can start

RealTime DBAExpert and navigate down to RESTRICTIONS:

```
RealTimeMaintain ----- Restrictions Selection -----  
Command ==> _____ DB2: QA1B  
  
Select one of the following. Then press ENTER.  
  
1. EXCLUSIONS          - View summary of all excluded objects  
2. VOLATILE CONTROL      - View and edit volatile table list  
3. WLM CONTROL           - Limit utility runs based on system load  
  
X. EXIT                  - Terminate RESTRICTIONS
```

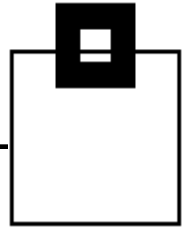


# Reorg Suppression and Detection

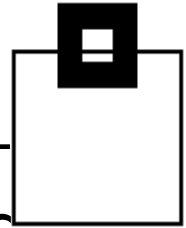
Selecting option 1 for EXCLUSIONS then shows:

```
RealTimeMaintain ----- Exclusions ----- Row 1 of 12
Command ==> _____ Scroll ==> CSR
                                     DB2: QA1B
Primary cmd: L(ocate DATABASE), S(ort), P(rint)
Line      cmd: S(elect), Z(oom Index)

                                     .... FOR UTILITY ....
      DATABASE SPACE      PART I | EXCLUDED BY | R R C | M M | C D Q S
                          X |            | O S O | O S | K S U T
-----|-----|-----|-----|-----|-----|-----|-----|
- *          *            | THRESHOLD  | Y Y | Y   |       Y Y
- ACCEL      DSNLOBT4     | THRESHOLD  | Y   |     |       |
- ANKEXPL1   ANKEX4       | THRESHOLD  | Y   |     |       |
- BIGINSDB   BEGINSTS    | THRESHOLD  | Y   |     |       |
- DBNIGHT*   TSNIGHT*    | EXEC MODE: ID=n | Y Y Y |     |       |
- DBNONP*    TSNONP*     | EXEC MODE: ID=* | Y Y Y |     |       |
- DBPEAK*    TSPEAK*     | EXEC MODE: ID=p | Y Y Y |     |       |
- DBWEND*    TSWEND*     | EXEC MODE: ID=w | Y Y Y |     |       |
- DB2CAT     *            | THRESHOLD  | Y Y | Y   |       |
- DSN8*      *            | THRESHOLD  | Y Y Y | Y Y | Y Y Y Y
- MIVP*      *            | THRESHOLD  |     | Y   |       |
- ROYSDB     ROYSTS      | DSNACCOR EXCEPTION TBL | Y   |     |       |
-----|-----|-----|-----|-----|-----|-----|
```



# Reorg Suppression and Detection



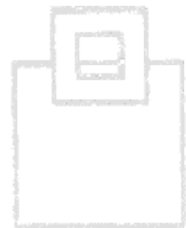
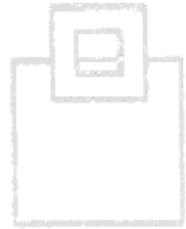
Using the line command S to select the DSNACCOR values then shows:

```
RealTimeMaintain ----- Exclusions ----- Row 1 of 12
Command ==> Scroll ==> CSR
DB2: QA1B

Primary cmd: L(ocate DATABASE), S(ort), P(rint)
Line
----- DSNACCOR Exception Table ----- Row 1 of 1
Command ==> Scroll ==> CSR
DB2: QA1B
Primary cmd: L(ocate DBNAME), P(rint), S(ort)

  DBNAME   NAME      QUERYTYPE
-----
ROYSDB    ROYSTS    REORG from WLX.....

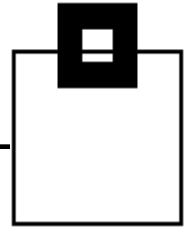
DAT
---
*
ACC
ANK
BIG
DBN
DBN
DBP
DBW
DB2
DSN
MIV
S  ROY
---
```



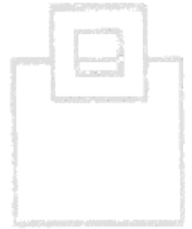
So you can see from where this request came!

# AGENDA

---

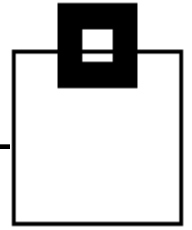


1. WLX – How it works
2. Quiet Times – Is a table in use?
3. Reorg Suppression and Detection
4. **Cluster Index – Review the choice**
5. Utility review – What ran when and did what?
6. Q&A Session



# Cluster Index – Review the choice

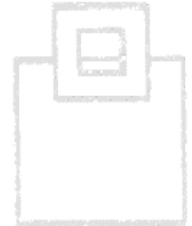
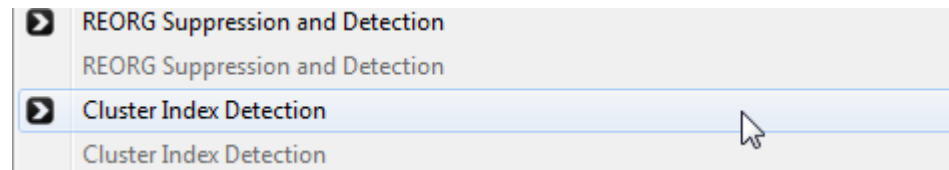
---



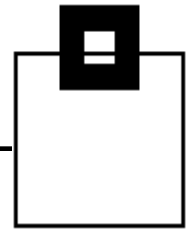
Clustering indexes are a great idea for SQL performance. The only problem was always

“Which Index should get the CLUSTER keyword?”

Historically it was always the first index. Nowadays this decision must be checked as it could quite well be that the design was sub-optimal:

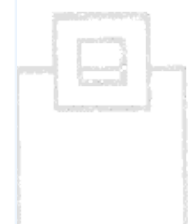
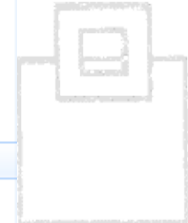


# Cluster Index – Review the choice

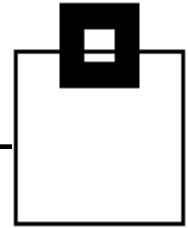


After selecting the time ranges of interest you then get an Index overview for the workload:

Table creator	Table name	Executions	Rows processed per execution	Index creator	Index name
PROD	PLAN_TABLE	7	7,965		
PTFADMIN	LYDIATAB	0	0	PTFADMIN	PTFTIN08
PTFADMIN	LYDIATAB	0	0	PTFADMIN	PTFTIN09
PTFADMIN	LYDIATAB	3	3	PTFADMIN	PTFTIN10
PTFADMIN	MEMBER	11	8		
PTFADMIN	MEMBER	46	1	PTFADMIN	PTFTIN06
PTFADMIN	MEMBER	82	1	PTFADMIN	PTFTIN07
PTFADMIN	MEMBER	3	0	PTFADMIN	PTFTIN07
PTFADMIN	MEMBER	9	0	PTFADMIN	PTFTIN07
PTFADMIN	MEMBER	3	1	PTFADMIN	PTFTIN07
PTFADMIN	PTF	5	1		
PTFADMIN	PTF	4	0	PTFADMIN	PTFTIN02
PTFADMIN	PTF	1	1	PTFADMIN	PTFTIN02
PTFADMIN	PTF	2	0	PTFADMIN	PTFTIN03
PTFADMIN	PTF	100	1	PTFADMIN	PTFTIN04
PTFADMIN	PTF	17	0	PTFADMIN	PTFTIN04
PTFADMIN	PTF	118	0	PTFADMIN	PTFTIN04
PTFADMIN	PTF	86	0	PTFADMIN	PTFTIN04



# Cluster Index – Review the choice



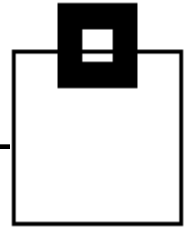
After selecting the time ranges of interest you then get an Index overview for the workload:

Used	Access type	Index only	Non-matching index scan	Number of entries	OBID	Clustering	Clustered	Clusterratio	GETPAGES per execution
D	R	N	-	6	0	-	-	0	16,103
-	-	-	-	0	43	Y	Y	100	0
-	-	-	-	0	45	N	Y	100	0
D	I	Y	N	3	47	N	Y	100	2
D	R	N	-	9	0	-	-	0	4,518
D	I	Y	N	46	29	Y	Y	99	3
D	I	N	N	29	31	N	Y	99	567
D	I	N	Y	3	31	N	Y	99	176
D	I	Y	N	4	31	N	Y	99	15
D	N	N	N	1	31	N	Y	99	27
D	R	N	-	3	0	-	-	0	92
D	I	N	N	4	19	Y	Y	100	132
D	I	Y	N	1	19	Y	Y	100	8
D	I	N	N	2	21	N	Y	100	8
D	I	N	N	7	23	N	Y	100	3
D	I	N	Y	10	23	N	Y	100	22
D	I	Y	N	14	23	N	Y	100	2
D	I	Y	Y	28	23	N	Y	100	22

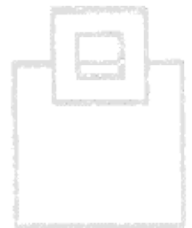
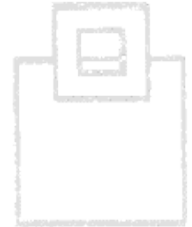
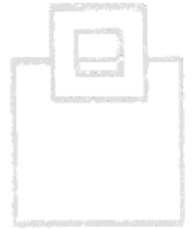


# AGENDA

---



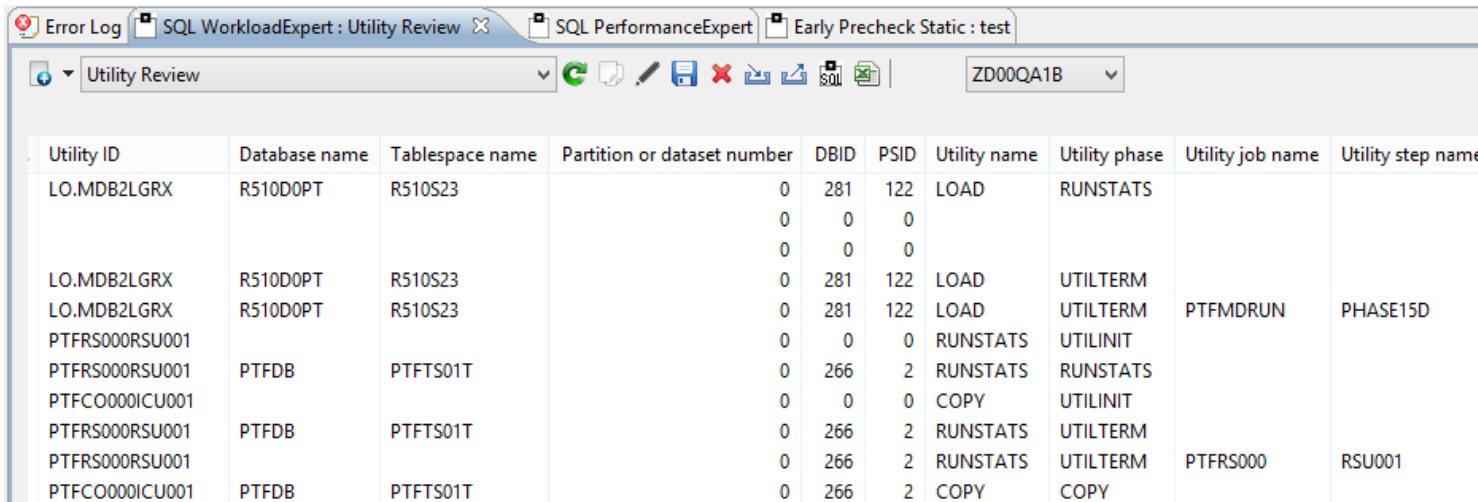
1. WLX – How it works
2. Quiet Times – Is a table in use?
3. Reorg Suppression and Detection
4. Cluster Index – Review the choice
5. **Utility review – What ran when and did what?**
6. Q&A Session





# Utility Review

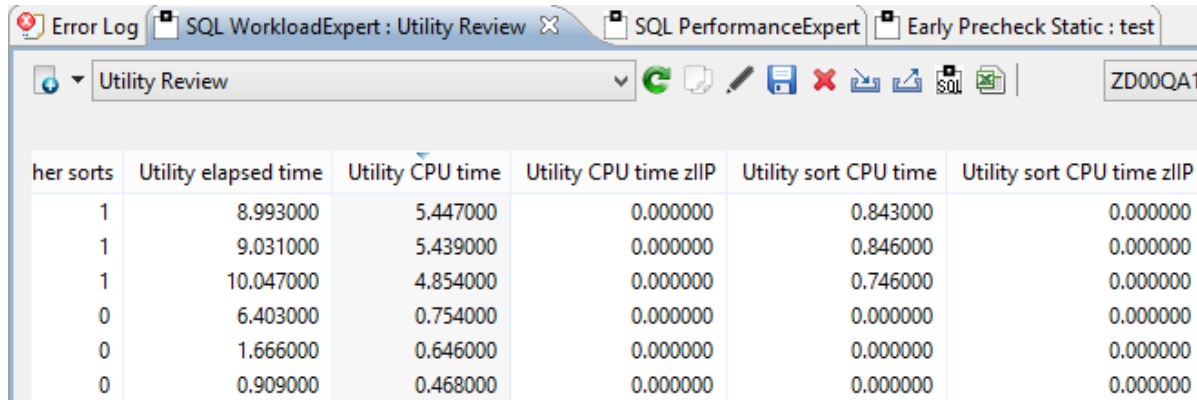
Ever wondered how many Utilities are running? Ever wondered how much CPU, IO or Elapsed they are using?



Utility ID	Database name	Tablespace name	Partition or dataset number	DBID	PSID	Utility name	Utility phase	Utility job name	Utility step name
LO.MDB2LGRX	R510D0PT	R510S23		0	281	122	LOAD	RUNSTATS	
				0	0	0			
				0	0	0			
LO.MDB2LGRX	R510D0PT	R510S23		0	281	122	LOAD	UTILTERM	
LO.MDB2LGRX	R510D0PT	R510S23		0	281	122	LOAD	UTILTERM	PTFMDRUN
PTFRS000RSU001				0	0	0	RUNSTATS	UTILINIT	
PTFRS000RSU001	PTFDB	PTFTS01T		0	266	2	RUNSTATS	RUNSTATS	
PTFCO000ICU001				0	0	0	COPY	UTILINIT	
PTFRS000RSU001	PTFDB	PTFTS01T		0	266	2	RUNSTATS	UTILTERM	
PTFRS000RSU001				0	266	2	RUNSTATS	UTILTERM	PTFRS000
PTFCO000ICU001	PTFDB	PTFTS01T		0	266	2	COPY	COPY	RSU001

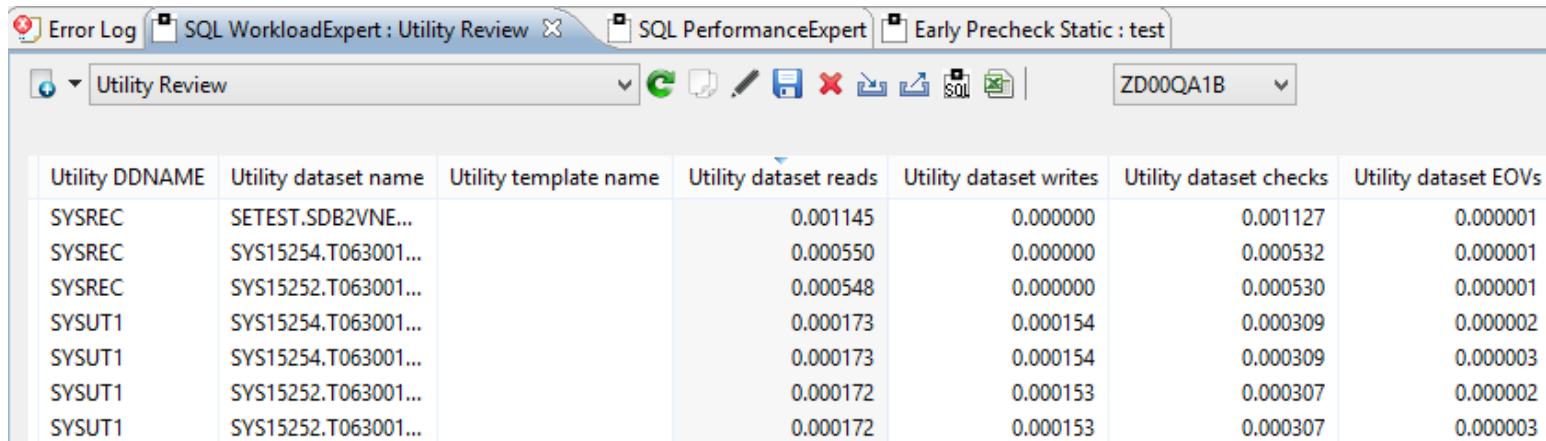
# Utility Review

Now you know!



The screenshot shows the 'Utility Review' window in SQL WorkloadExpert. The table displays performance metrics for various utility jobs. The columns are: 'her sorts', 'Utility elapsed time', 'Utility CPU time', 'Utility CPU time zIIP', 'Utility sort CPU time', and 'Utility sort CPU time zIIP'.

her sorts	Utility elapsed time	Utility CPU time	Utility CPU time zIIP	Utility sort CPU time	Utility sort CPU time zIIP
1	8.993000	5.447000	0.000000	0.843000	0.000000
1	9.031000	5.439000	0.000000	0.846000	0.000000
1	10.047000	4.854000	0.000000	0.746000	0.000000
0	6.403000	0.754000	0.000000	0.000000	0.000000
0	1.666000	0.646000	0.000000	0.000000	0.000000
0	0.909000	0.468000	0.000000	0.000000	0.000000

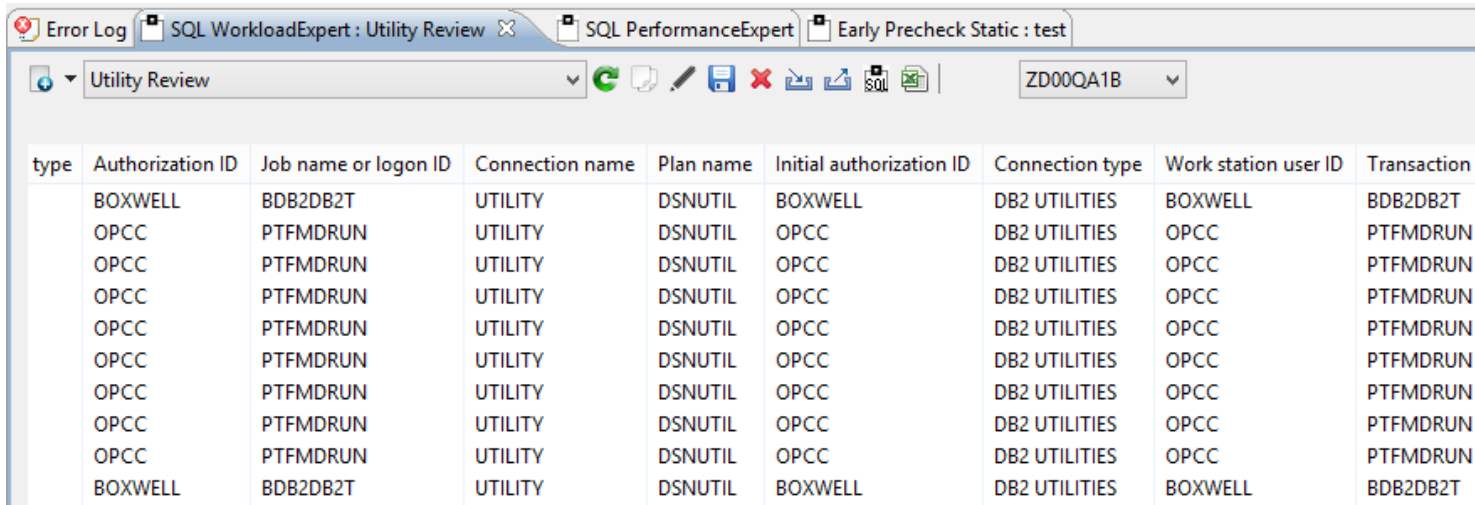


The screenshot shows the 'Utility Review' window in SQL WorkloadExpert, displaying a table of utility dataset statistics. The columns are: 'Utility DDNAME', 'Utility dataset name', 'Utility template name', 'Utility dataset reads', 'Utility dataset writes', 'Utility dataset checks', and 'Utility dataset EOVS'.

Utility DDNAME	Utility dataset name	Utility template name	Utility dataset reads	Utility dataset writes	Utility dataset checks	Utility dataset EOVS
SYSREC	SETEST.SDB2VNE...		0.001145	0.000000	0.001127	0.000001
SYSREC	SYS15254.T063001...		0.000550	0.000000	0.000532	0.000001
SYSREC	SYS15252.T063001...		0.000548	0.000000	0.000530	0.000001
SYSUT1	SYS15254.T063001...		0.000173	0.000154	0.000309	0.000002
SYSUT1	SYS15254.T063001...		0.000173	0.000154	0.000309	0.000003
SYSUT1	SYS15252.T063001...		0.000172	0.000153	0.000307	0.000002
SYSUT1	SYS15252.T063001...		0.000172	0.000153	0.000307	0.000003

# Utility Review

Including all the data about Who did it:

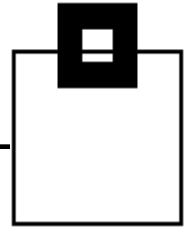


The screenshot shows the 'Utility Review' window in SQL WorkloadExpert. The window title bar includes 'Error Log', 'SQL WorkloadExpert : Utility Review', 'SQL PerformanceExpert', and 'Early Precheck Static : test'. The main area displays a table with the following columns: type, Authorization ID, Job name or logon ID, Connection name, Plan name, Initial authorization ID, Connection type, Work station user ID, and Transaction. The table contains 10 rows of data.

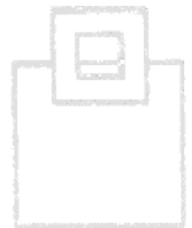
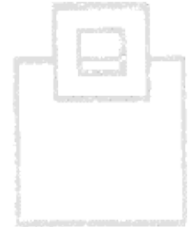
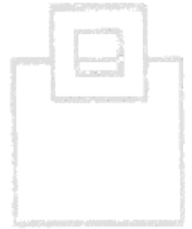
type	Authorization ID	Job name or logon ID	Connection name	Plan name	Initial authorization ID	Connection type	Work station user ID	Transaction
	BOXWELL	BDB2DB2T	UTILITY	DSNUTIL	BOXWELL	DB2 UTILITIES	BOXWELL	BDB2DB2T
	OPCC	PTFMDRUN	UTILITY	DSNUTIL	OPCC	DB2 UTILITIES	OPCC	PTFMDRUN
	OPCC	PTFMDRUN	UTILITY	DSNUTIL	OPCC	DB2 UTILITIES	OPCC	PTFMDRUN
	OPCC	PTFMDRUN	UTILITY	DSNUTIL	OPCC	DB2 UTILITIES	OPCC	PTFMDRUN
	OPCC	PTFMDRUN	UTILITY	DSNUTIL	OPCC	DB2 UTILITIES	OPCC	PTFMDRUN
	OPCC	PTFMDRUN	UTILITY	DSNUTIL	OPCC	DB2 UTILITIES	OPCC	PTFMDRUN
	OPCC	PTFMDRUN	UTILITY	DSNUTIL	OPCC	DB2 UTILITIES	OPCC	PTFMDRUN
	OPCC	PTFMDRUN	UTILITY	DSNUTIL	OPCC	DB2 UTILITIES	OPCC	PTFMDRUN
	OPCC	PTFMDRUN	UTILITY	DSNUTIL	OPCC	DB2 UTILITIES	OPCC	PTFMDRUN
	BOXWELL	BDB2DB2T	UTILITY	DSNUTIL	BOXWELL	DB2 UTILITIES	BOXWELL	BDB2DB2T

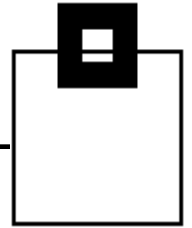
# AGENDA

---

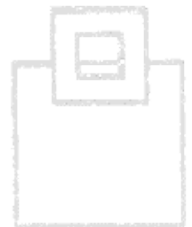
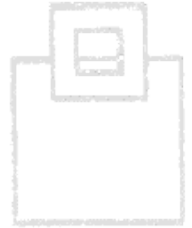


1. WLX – How it works
2. Quiet Times – Is a table in use?
3. Reorg Suppression and Detection
4. Cluster Index – Review the choice
5. Utility review – What ran when and did what?
6. Q&A Session





Thank you for listening!



If you want more info then please subscribe to my newsletter by going to [www.seg.de](http://www.seg.de) or [www.segus.com](http://www.segus.com)

---

Roy Boxwell  
SOFTWARE ENGINEERING GmbH  
&  
SEGUS Inc

*R.Boxwell@seg.de*

