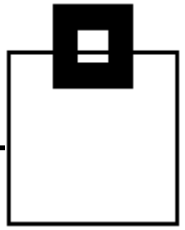
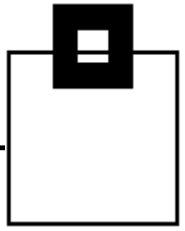


Where's the BiF??

Roy Boxwell,
Senior Software Architect,
Software Engineering GmbH

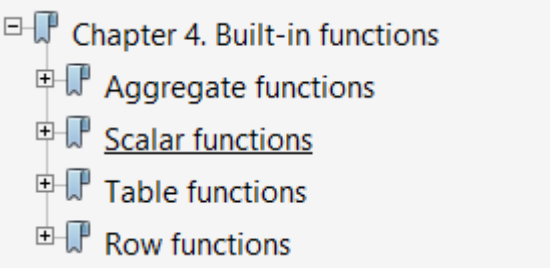
Agenda



- What's a BiF?
- Where's the BiF?
- Customer Story
- Where's the BiF? – Wrap up
- Q&A

What's a BiF?

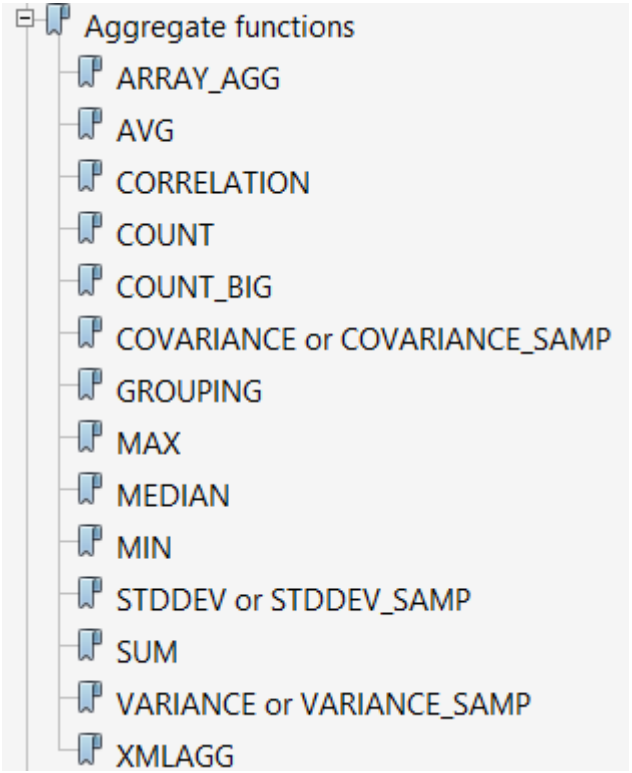
A BiF is a Built-In Function such as CHAR, DECIMAL, etc.
They exist in four different flavors:



There are hundreds of them these days and every new release of DB2 brings more. In the last few DB2 releases, IBM has changed a few to make DB2 more compatible with SQL standards.

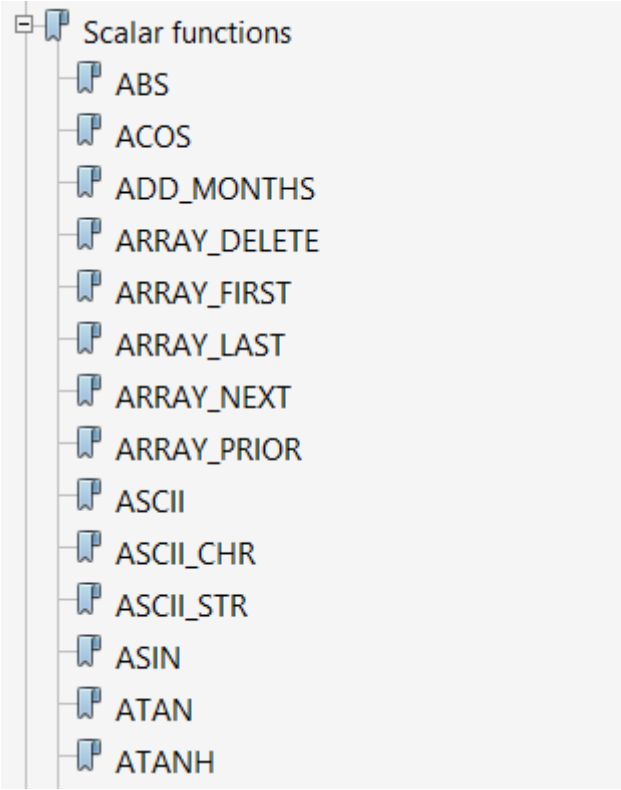
What's a BiF?

Aggregate functions we all know - but mostly we use just AVG, COUNT, MAX, MIN, and SUM:



What's a BiF?

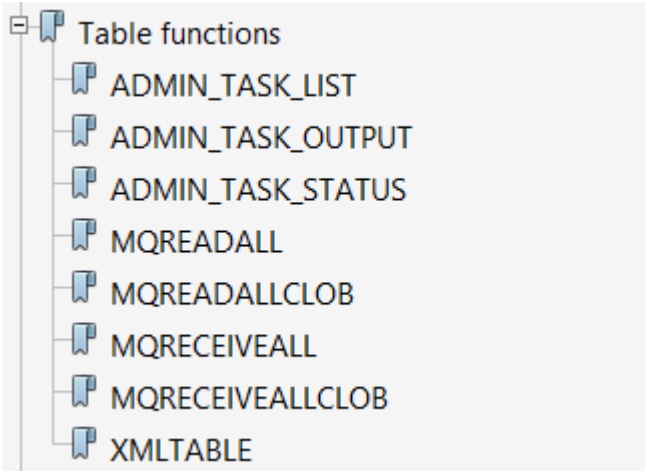
Scalar functions are the ones we really use all the time.
Here are the first few in DB2 11:



There are 184 of these…

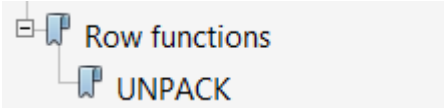
What's a BiF?

Table functions are a bit exotic and not used that often:



What's a BiF?

Row functions are brand-new, in fact there is only one at this time! Quite why it gets the plural is a puzzle!



What's a BiF?

There are lots of examples of these in the DB2 Documentation and also in my newsletters, at least all new functions from DB2 V8 and up, so you can have a play in SPUFI to see what all these things can do.

So, now we know what a BiF is, Where's the BiF?

All these functions are very, very handy and useful tools that we use all the time, the problem is that the output of these BiFs sometimes changes with a DB2 release…

Where's the BiF?

The one that started the ball rolling was the change of decimal data within a CHAR function.

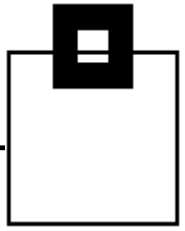
Your SQL looked like:

```
SELECT CHAR( DECCOL1 ) FROM SYSIBM.SYSDUMMY1;
```

You used to get `' 000.1'` and suddenly you got `'.1'`

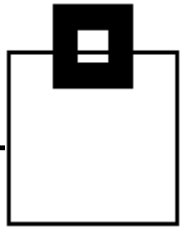
This caused a few German people to jump up and down so quick-as-a-flash IBM came up with the [in]famous BIF_COMPATIBILITY flag in ZPARMs…

Where‘s the BiF? BIF_COMPATIBILITY



IBM introduced BIF_COMPATIBILITY with values V9 for old format CHAR and CURRENT for new format.

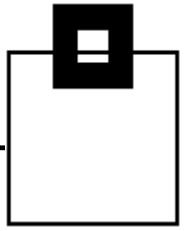
Where's the BiF? BIF_COMPATIBILITY



IBM introduced BIF_COMPATIBILITY with values V9 for old format CHAR and CURRENT for new format.

Then they discovered VARCHAR…so BIF_COMPATIBILITY got a new value: V9_DECIMAL_VARCHAR which included CHAR.

Where's the BiF? BIF_COMPATIBILITY

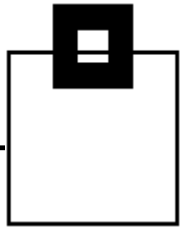


IBM introduced BIF_COMPATIBILITY with values V9 for old format CHAR and CURRENT for new format.

Then they discovered VARCHAR…so BIF_COMPATIBILITY got a new value: V9_DECIMAL_VARCHAR which included CHAR.

Then they discovered TIMESTAMPS…so now if BIF_COMPATIBILITY was set to V9 or V9_DECIMAL_VARCHAR illegal TIMESTAMP formats were "accepted".

Where's the BiF? BIF_COMPATIBILITY



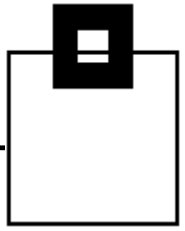
IBM introduced BIF_COMPATIBILITY with values V9 for old format CHAR and CURRENT for new format.

Then they discovered VARCHAR…so BIF_COMPATIBILITY got a new value: V9_DECIMAL_VARCHAR which included CHAR.

Then they discovered TIMESTAMPS…so now if BIF_COMPATIBILITY was set to V9 or V9_DECIMAL_VARCHAR illegal TIMESTAMP formats were "accepted".

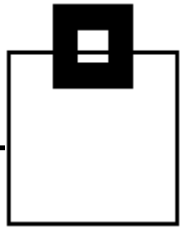
Then they discovered TRIM…so BIF_COMPATIBILITY got a new value: V9_TRIM for LTRIM, RTRIM, and STRIP which also included all of the above…

Where's the BiF? Further Fixes



Two new SCHEMAs were then created to help sort out the mess: SYSCOMPAT_V9 and SYSCURRENT (Just for CHAR, VARCHAR, and all the TRIM stuff. They must be added before SYSIBM in the path of course!)

Where's the BiF? Further Fixes



Two new SCHEMAs were then created to help sort out the mess: SYSCOMPAT_V9 and SYSCURRENT (Just for CHAR, VARCHAR, and all the TRIM stuff. They must be added before SYSIBM in the path of course!)

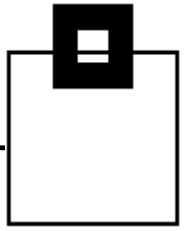
DB2 11: APPLCOMPAT – New ZPARM introduced.

V10R1 after migration

V11R1 after install

If set to V10R1 then DB2 10 NFM behavior is guaranteed for the next two releases. So in DB2 13 it will all go horribly wrong…That's about seven years from now!

Where's the BiF? Further Fixes



Two new SCHEMAs were then created to help sort out the mess: SYSCOMPAT_V9 and SYSCURRENT (Just for CHAR, VARCHAR, and all the TRIM stuff. They must be added before SYSIBM in the path of course!)

DB2 11: APPLCOMPAT – New ZPARM introduced.

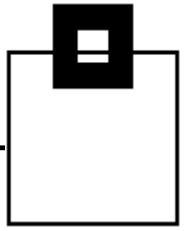
V10R1 after migration

V11R1 after install

If set to V10R1 then DB2 10 NFM behavior is guaranteed for the next two releases. So in DB2 13 it will all go horribly wrong…That's about seven years from now!

DB2 11: Two new BiF's - CHAR9 and VARCHAR9.

Where's the BiF? Monitoring IFCIDs

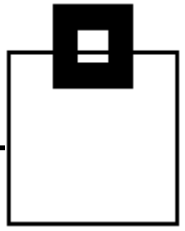


This was, to quote Monty Python, "Getting Silly"

So then the IFCID 366 was born. This IFCID is output whenever DB2 detects a *possible* change of behavior from the current release to the next release.

Within this IFCID are a bunch of fields (for the full details please look in your <db2hlq>.SDSNMACS(DSNDQW05) member) including a QW0366FN field that is a number (ICI – Incompatible Change Indicator, you just gotta love another TLA from IBM) that started at 1 and went to 3 with the initial version. A "1" meant a CHAR decimal problem, a "2" meant a VARCHAR decimal problem, and a "3" meant a TIMESTAMP problem.

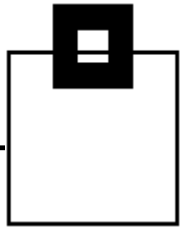
Where's the BiF? Monitoring IFCIDs



Now it has 10 numbers for DB2 10 and 17(!) for DB2 11.

People also started getting lots of 366's and so the 376 was born, but only for DB2 11. This is an aggregated version of the 366.

Where's the BiF? Monitoring IFCIDs

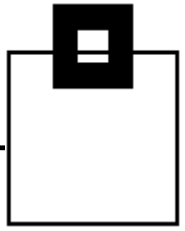


Now it has 10 numbers for DB2 10 and 17(!) for DB2 11.

People also started getting lots of 366's and so the 376 was born, but only for DB2 11. This is an aggregated version of the 366.

This was TMI (Too Much Information) as most shops were attempting to use SMF and filling up packs of disks with millions of records that contained Statement Ids of dynamic SQL that had long disappeared from the DSC.

Where's the BiF? Monitoring IFCIDs



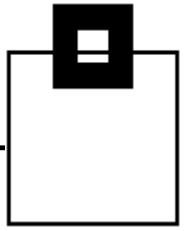
Now it has 10 numbers for DB2 10 and 17(!) for DB2 11.

People also started getting lots of 366's and so the 376 was born, but only for DB2 11. This is an aggregated version of the 366.

This was TMI (Too Much Information) as most shops were attempting to use SMF and filling up packs of disks with millions of records that contained Statement Ids of dynamic SQL that had long disappeared from the DSC.

This is "sub-optimal"… (or, in English, a problem!)

Customer Example



Temporary solution for the problem until the affected applications are identified:

```
ZPARM BIF_COMPATIBILITY=V9_DECIMAL_VARCHAR
```


But this just “hides” the problem of course

Let’s have a quick look at a real customer and his problems…

Methodology at customer‘s site

- To get all relevant applications (daily, weekly etc.) since mid of April 2014 a performance trace with IFCID 366 was run in production

```
-STA TRACE(P) CLASS(32) IFCID(366) DEST(SMF) SCOPE(GROUP)
```

- For packages with static SQL additionally IFCID 63 was activated

- Identified were 18 programs:

1	Cobol Program	(Static SQL)
1	C++ Program	(Dynamic SQL) 3rd party
4	Java Programs	(Dynamic SQL)
12	High Performance Unloads	(Dynamic SQL)

Evaluation of trace records

- Evaluation of SMF records with the help of
  OMEGAMON XE FOR DB2 PERFORMANCE EXPERT

- BMC Mainview also offers reporting capabilities described in the „Performance Reporter User Guide“

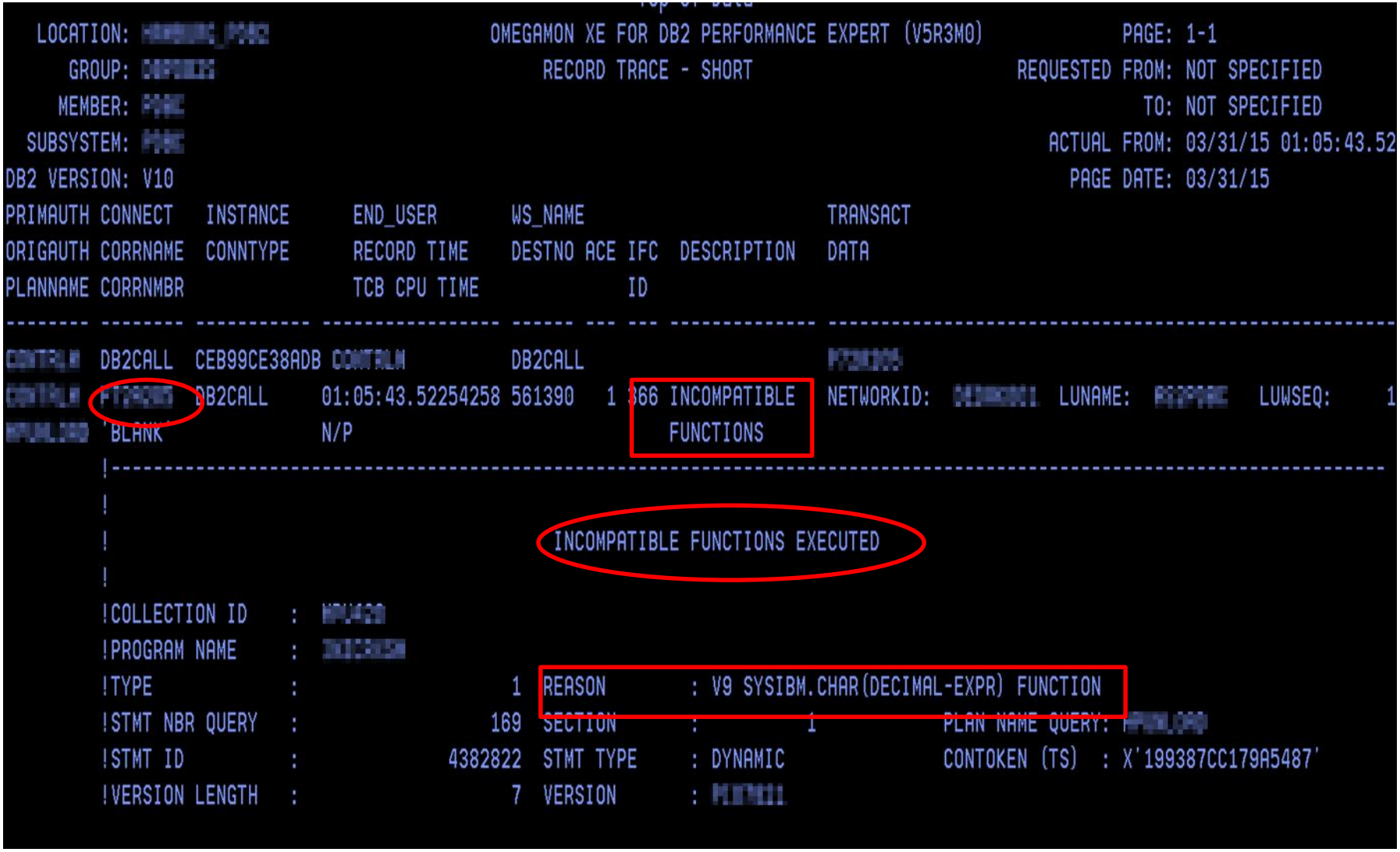
  The BMC Datacollector has to be active.

  Attention:

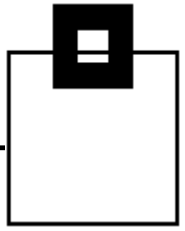
  Depending on the frequency of program calls space requirements for the SMF records will increase drastically!

JCL of the evaluation job (OMEGAMON)

```
//[redacted]  EXEC PGM=DB2PM
//STEPLIB  DD   DISP=SHR,DSN=[redacted]
//         DD   DISP=SHR,DSN=[redacted]
//INPUTDD  DD   DISP=SHR,DSN=[redacted]
//JOBSUMDD DD   SYSOUT=*
//RTTRCDD1 DD   SYSOUT=*
//SYSIN    DD   *
  GLOBAL
                       TIMEZONE(-2)
                       EXCLUDE(
                               PRIMAUTH   (
                                          [redacted]*))
                       INCLUDE(
                               IFCID  (366)
                               GROUP  ([redacted])
                               )
  RECTRACE
              TRACE
                       LEVEL(SHORT)
                       DDNAME(RTTRCDD1)
  EXEC
```

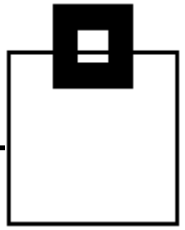

LOCATION:
OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)
PAGE: 1-1
GROUP:
RECORD TRACE - SHORT
REQUESTED FROM: NOT SPECIFIED
MEMBER:
TO: NOT SPECIFIED
SUBSYSTEM:
ACTUAL FROM: 03/31/15 01:05:43.52
DB2 VERSION: V10
PAGE DATE: 03/31/15
PRIMAUTH CONNECT INSTANCE END_USER WS_NAME TRANSACT
ORIGAUTH CORRNAME CONNTYPE RECORD TIME DESTNO ACE IFC DESCRIPTION DATA
PLANNAME CORRNMBR TCB CPU TIME ID
DB2CALL CEB99CE38ADB
DB2CALL
DB2CALL 01:05:43.52254258 561390 1 366 INCOMPATIBLE FUNCTIONS
NETWORKID:
LUNAME:
LUWSEQ: 1
'BLANK' N/P
INCOMPATIBLE FUNCTIONS EXECUTED
!COLLECTION ID :
!PROGRAM NAME :
!TYPE : 1
!STMT NBR QUERY : 169
!STMT ID : 4382822
!VERSION LENGTH : 7
REASON : V9 SYSIBM.CHAR(DECIMAL-EXPR) FUNCTION
SECTION : 1
PLAN NAME QUERY:
STMT TYPE : DYNAMIC
CONTOKEN (TS) : X'199387CC179A5487'
VERSION :

SET CURRENT PATH = SYSCOMPAT_V9
BIF_COMPATIBILITY=V9_DECIMAL_VARCHAR



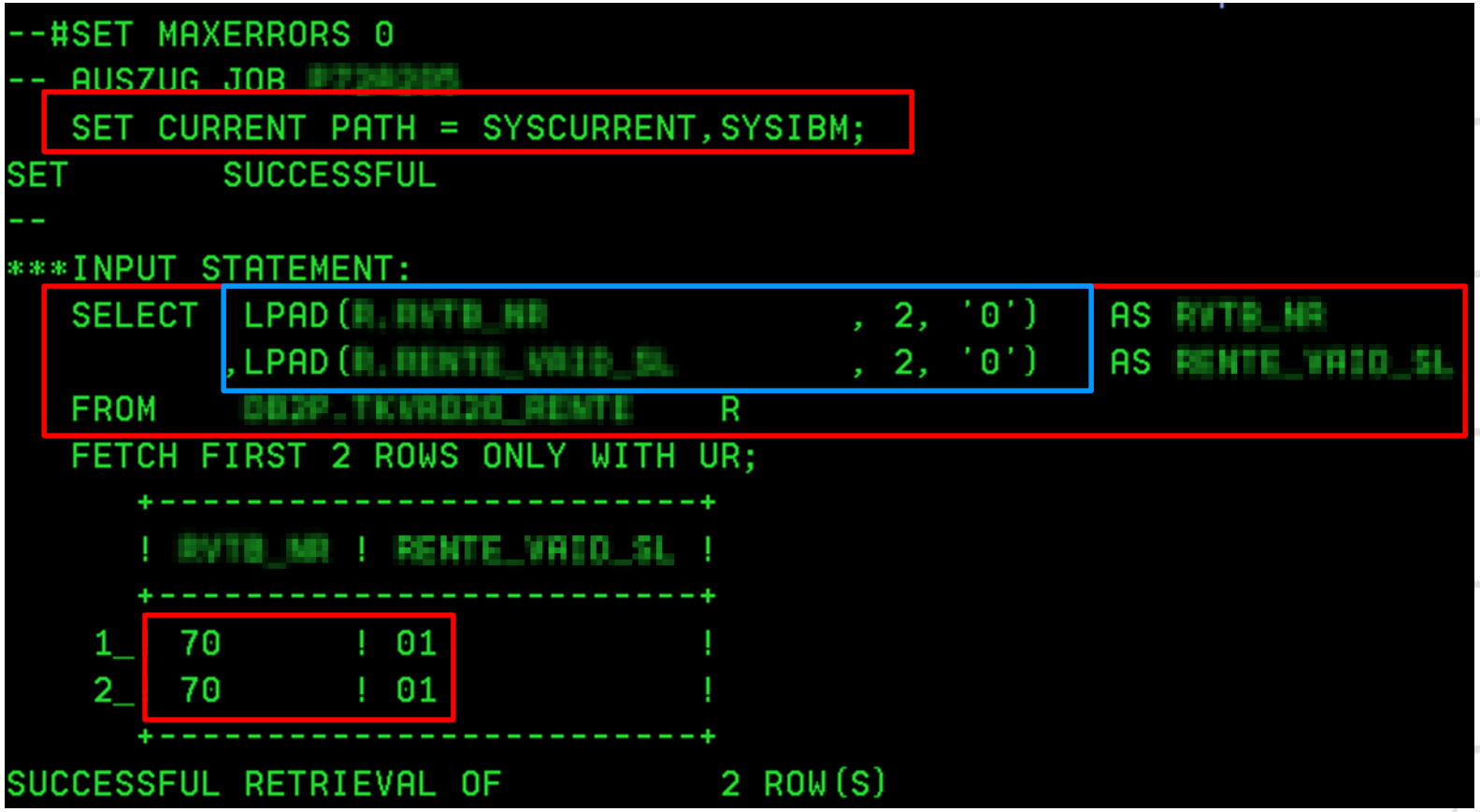
```
-- AUSZUG JOB PT2A2D5
   SET CURRENT PATH = SYSCOMPAT_V9;
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000,  SUCCESSFUL EXECUTION
SET       SUCCESSFUL
--
***INPUT STATEMENT:
   SELECT  SUBSTR(CHAR(R.AVTB_NR)       , 2, 2)     AS AVTB_NR
          ,SUBSTR(CHAR(R.RENTE_VAID_SL), 2, 2)     AS RENTE_VAID_SL
   FROM    DB3P.TEVA020_RENTE    R
   FETCH FIRST 2 ROWS ONLY WITH UR;
      +-------------------------+
      ! AVTB_NR ! RENTE_VAID_SL !
      +-------------------------+
    1_! 70      ! 01            !
    2_! 70      ! 01            !
      +-------------------------+
SUCCESSFUL RETRIEVAL OF          2 ROW(S)
```

SET CURRENT PATH = SYSCURRENT,SYSIBM BIF_COMPATIBILITY=V9_DECIMAL_VARCHAR



```
-- AUSZUG JOB [illegible]
   SET CURRENT PATH = SYSCURRENT,SYSIBM;
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000,  SUCCESSFUL EXECUTION
SET      SUCCESSFUL
--
***INPUT STATEMENT:
   SELECT  SUBSTR(CHAR(A.[illegible])      , 2, 2)     AS [illegible]
          ,SUBSTR(CHAR(A.[illegible]), 2, 2)     AS [illegible]
   FROM    [illegible]    R
   FETCH FIRST 2 ROWS ONLY WITH UR;
      +-------------------------+
      ! [illegible] ! [illegible] !
      +-------------------------+
    1_! 0       !                 !
    2_! 0       !                 !
      +-------------------------+
SUCCESSFUL RETRIEVAL OF          2 ROW(S)
```

Modified SQL Statement
BIF_COMPATIBILITY=V9_DECIMAL_VARCHAR

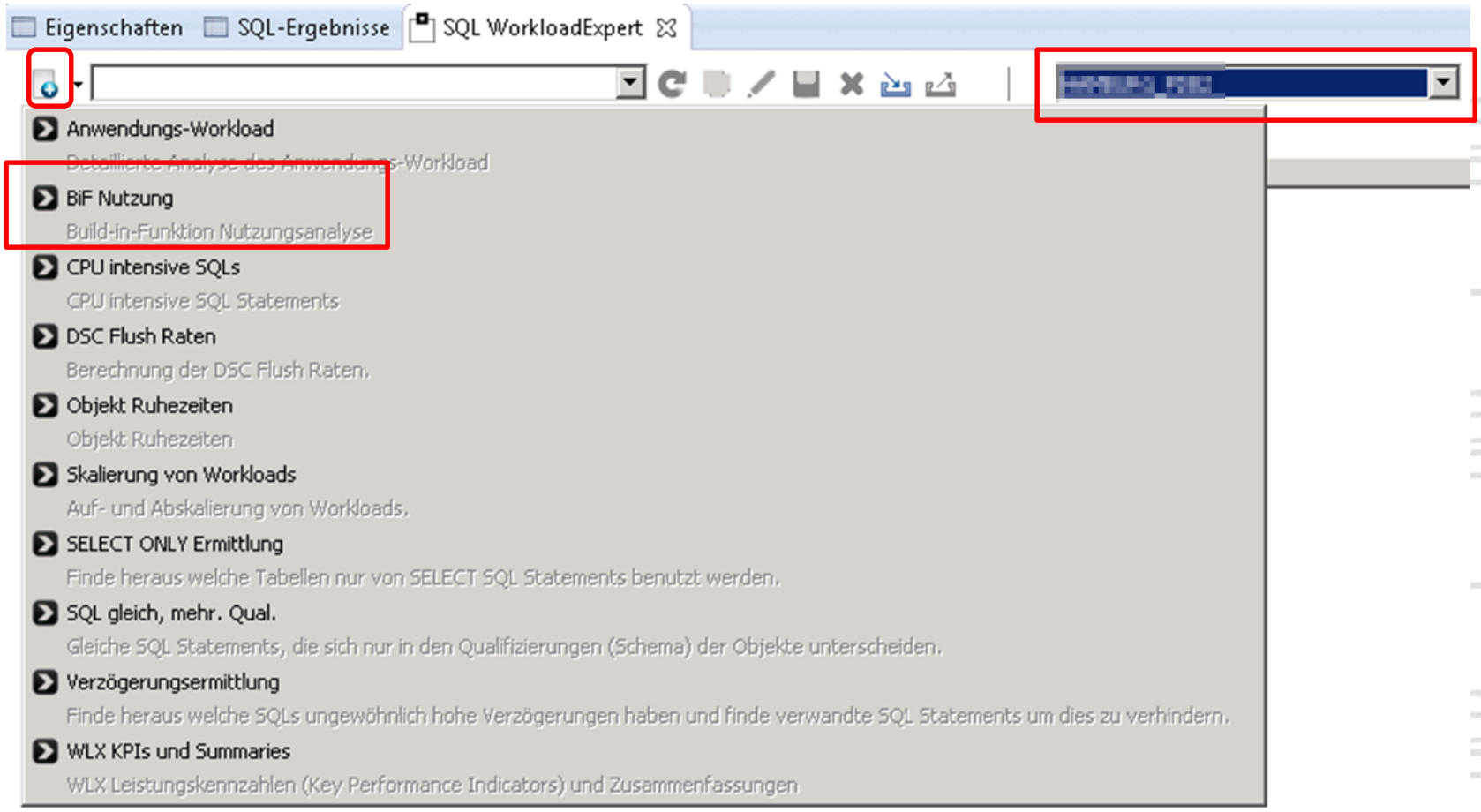


```
--#SET MAXERRORS 0
-- AUSZUG JOB
   SET CURRENT PATH = SYSCURRENT,SYSIBM;
SET        SUCCESSFUL
--
***INPUT STATEMENT:
   SELECT  LPAD(A.BVTB_NR                    , 2, '0')   AS BVTB_NR
          ,LPAD(A.RENTE_VAID_SL              , 2, '0')   AS RENTE_VAID_SL
   FROM     DB2P.TKVA030_RENTE     R
   FETCH FIRST 2 ROWS ONLY WITH UR;
      +-------------------------+
      ! BVTB_NR ! RENTE_VAID_SL !
      +-------------------------+
    1_  70       ! 01            !
    2_  70       ! 01            !
      +-------------------------+
SUCCESSFUL RETRIEVAL OF           2 ROW(S)
```

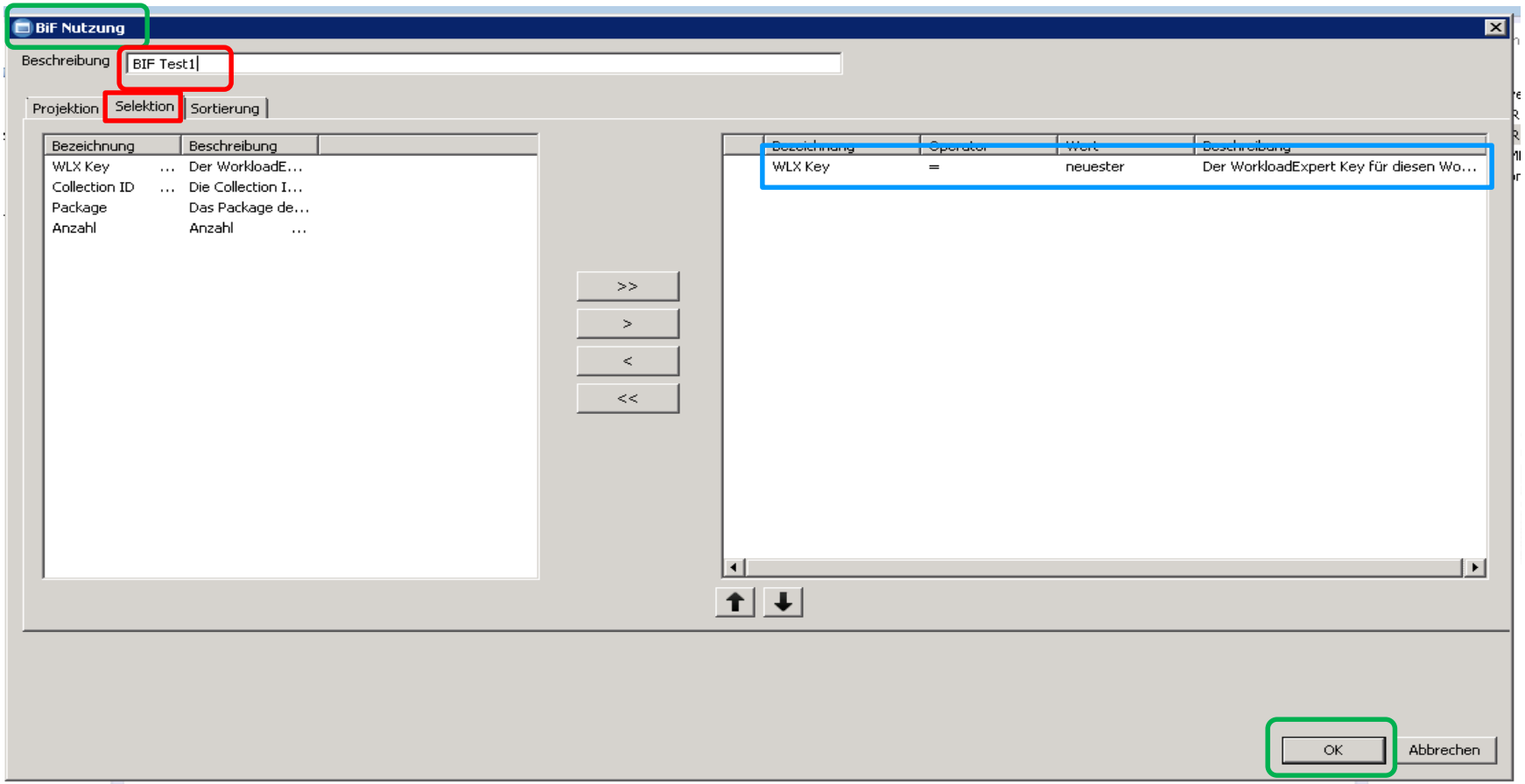
SQL WorkloadExpert (WLX) for DB2 z/OS

- The SQL WorkloadExpert architecture comes in two parts:
  - Mainframe DB2 Workload collector and processing engine
  - Workstation analysis and reporting engine
- DSC and SSC were selected and the data is stored in WLX tables
- Processing and display of the data in Datastudio 4.1.1 or Eclipse on the workstation

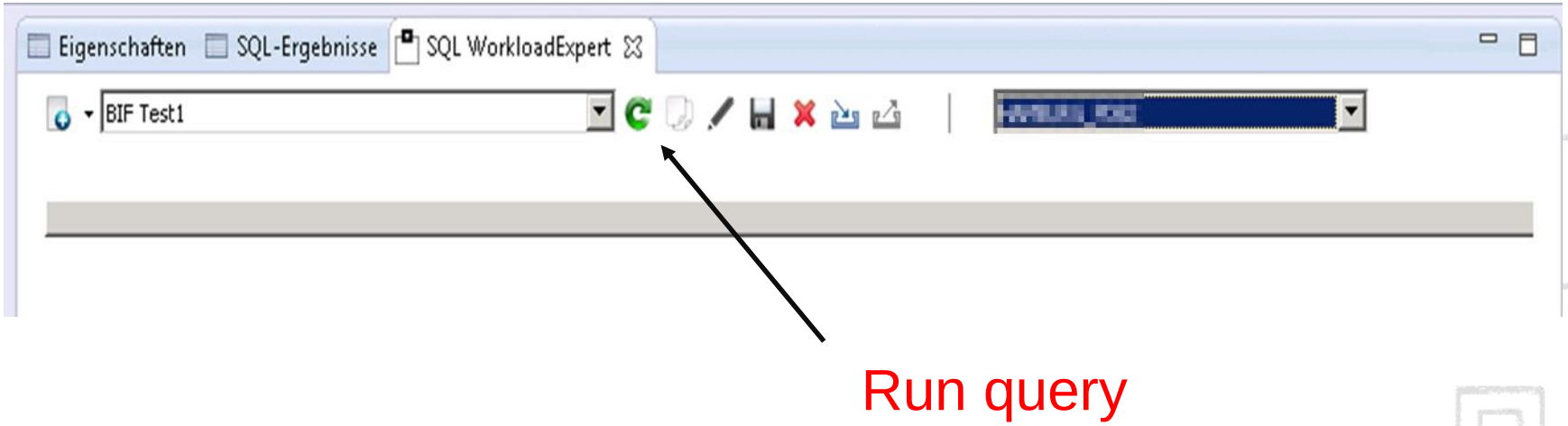
Evaluation with SQL WorkloadExpert (1)



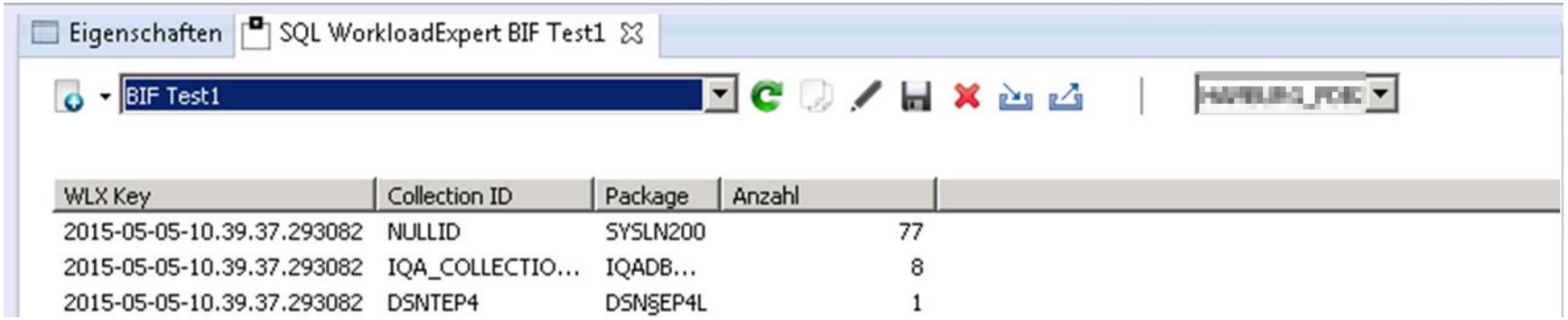
Evaluation with SQL WorkloadExpert (2)



Evaluation with SQL WorkloadExpert (3)

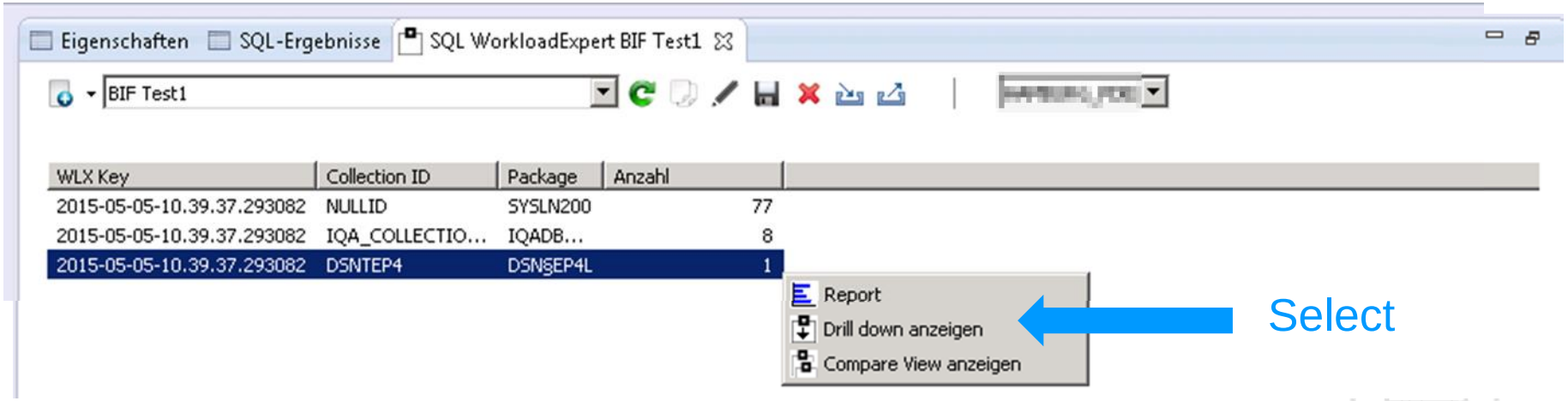


Evaluation with SQL WorkloadExpert (4)

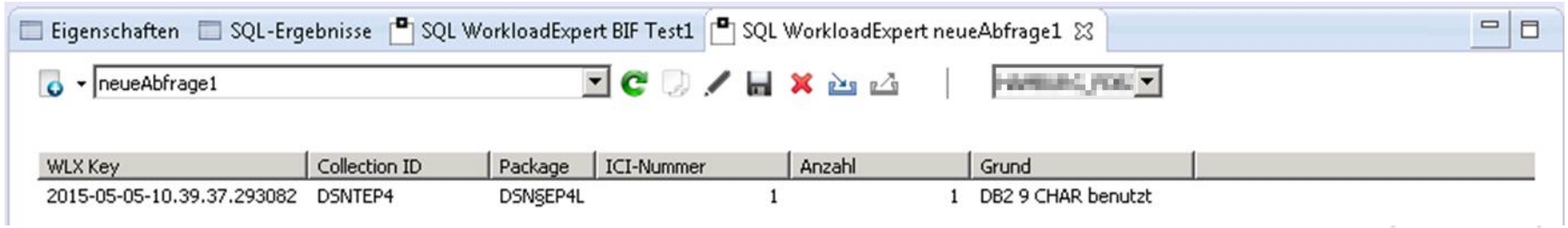


WLX Key	Collection ID	Package	Anzahl
2015-05-05-10.39.37.293082	NULLID	SYSLN200	77
2015-05-05-10.39.37.293082	IQA_COLLECTIO...	IQADB...	8
2015-05-05-10.39.37.293082	DSNTEP4	DSNSEP4L	1

Evaluation with SQL WorkloadExpert (5)

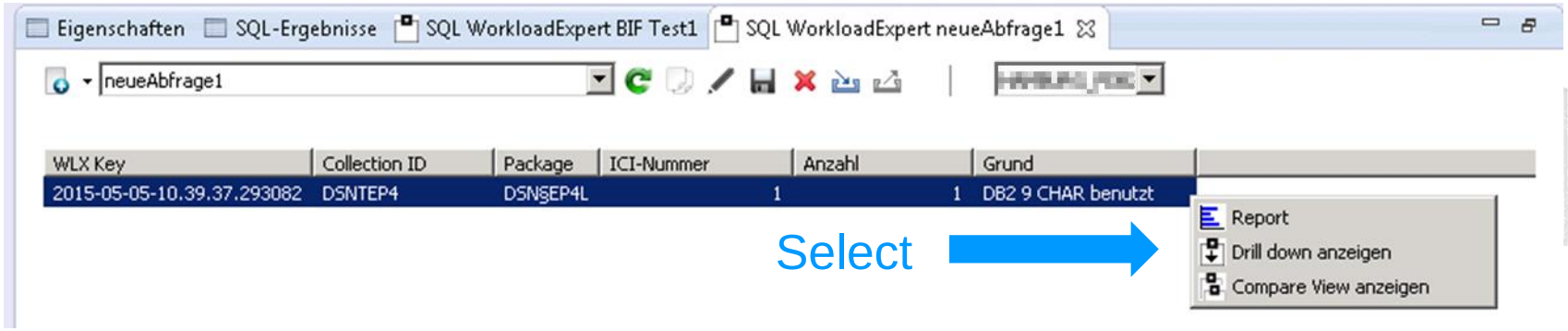


Evaluation with SQL WorkloadExpert (6)

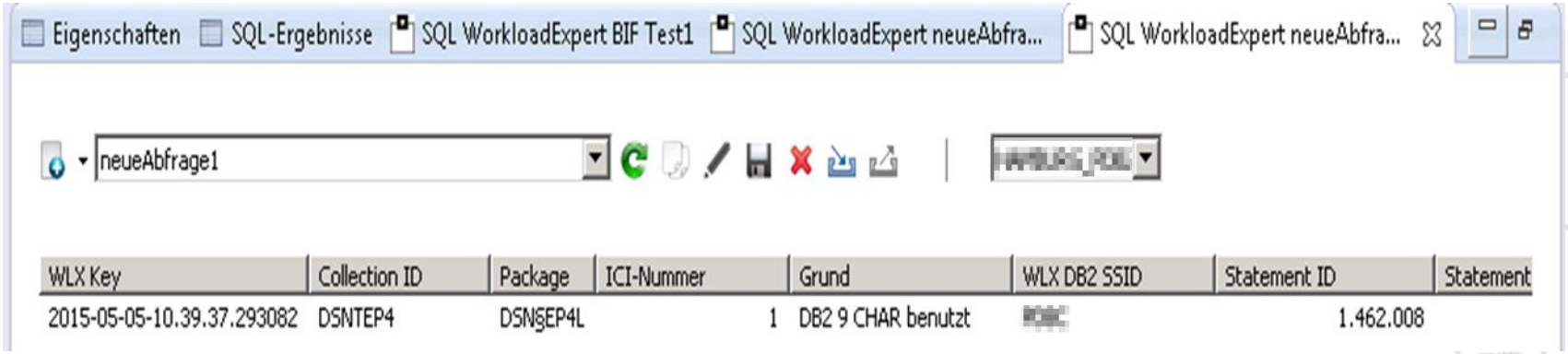


WLX Key	Collection ID	Package	ICI-Nummer	Anzahl	Grund
2015-05-05-10.39.37.293082	DSNTEP4	DSNSEP4L	1	1	DB2 9 CHAR benutzt

Evaluation with SQL WorkloadExpert (7)

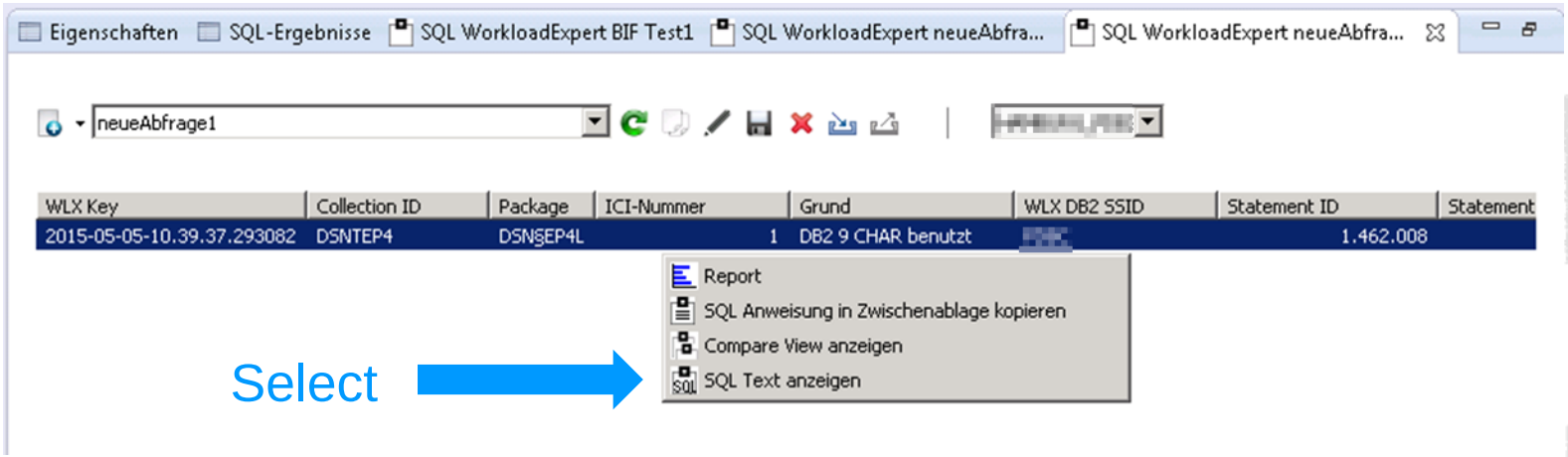


Evaluation with SQL WorkloadExpert (8)

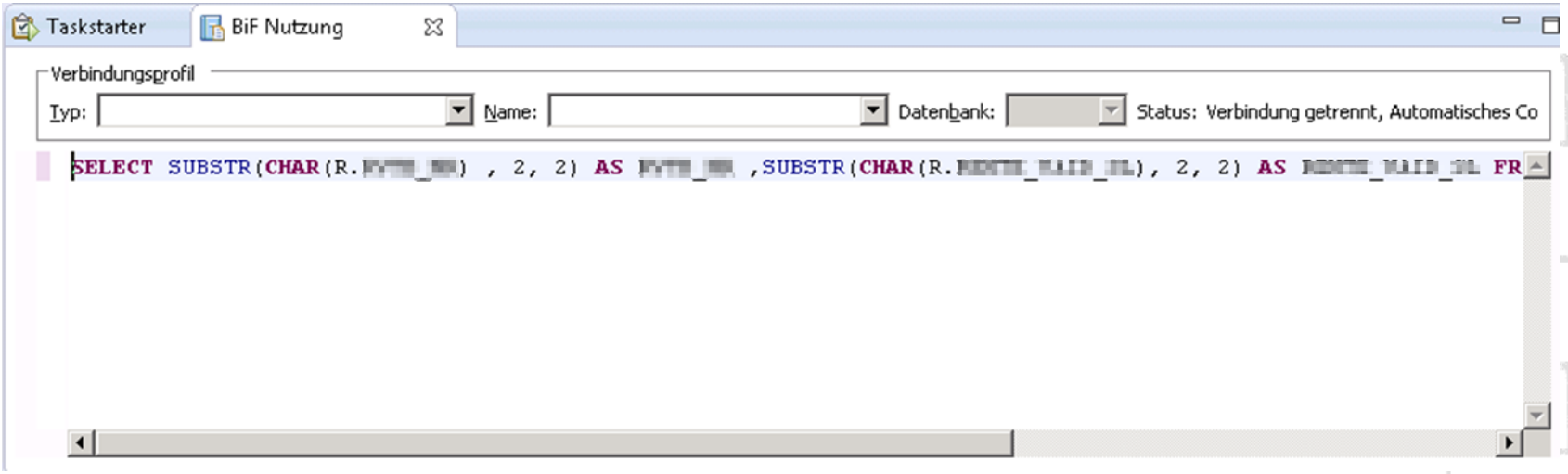


WLX Key	Collection ID	Package	ICI-Nummer	Grund	WLX DB2 SSID	Statement ID	Statement
2015-05-05-10.39.37.293082	DSNTEP4	DSNSEP4L	1	DB2 9 CHAR benutzt	[redacted]	1.462.008	

Evaluation with SQL WorkloadExpert (9)

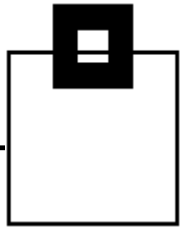


Evaluation with SQL WorkloadExpert (10)




```
SELECT   SUBSTR(CHAR(R.[illegible])        , 2, 2)      AS [illegible]
       ,SUBSTR(CHAR(R.[illegible]), 2, 2)      AS [illegible]
FROM     [illegible]     R
FETCH FIRST 2 ROWS ONLY WITH UR;
```

Customers Goal

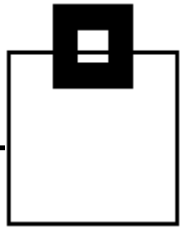


To already redress in DB2 10 all incompatibilities before the migration to DB2 11 and to set the ZPARM BIF_COMPATIBILITY, still in DB2 10, to be CURRENT.

What we have learnt

- In contrast to the previously performed examinations using Omegamon, with WLX you need not collect SMF data for more than 12 months before the migration and analyze them.
- With WLX you do not even need SMF data. The WLX straight-forward approach saves enormous time, manpower, and storage.
- “Where is the BiF?” Is answered directly and the SQL is categorized and shown – Even for dynamic SQL you get the entire SQL text displayed.

Where's the BiF?



As you have seen BiFs can cause you problems if you are *not* aware of their usage:

- Start a project plan *now* for finding the BiF
- Action and Approve any and all SQL Changes
- Store all results for comparison next time…
- Stay aware and ready!

Where's the BiF?

You do not want your next migration looking like this:

Where's the BiF?