

SAS® vs DB2® as a Relational DBMS for End Users: Three Corporations with Three Different Solutions

Stephen C. Scott, Scott Consulting Services, Inc.

ABSTRACT:

Three corporations with different sizes and types of Relational DBMS are reviewed. All three started with DB2 as the DBMS and SAS as the analysis and reporting system.

Corp. A sought to build a information warehouse and to store its data in DB2 tables - but in the process discovered that SAS software as a DBMS reduced CPU times by over 86%. Because of the size of this database - 30 gigabytes - stored on 17 3380 triples - the potential saving were in the millions of dollars.

Corp. B purchased a DB2 based Human Resource and Payroll System. They also built a DB2 corporate reporting database based on the production system and updated daily - and a SAS reporting database from which the weekly/supplemental/monthly/quarterly/annual reports are produced. By reporting from SAS instead of DB2 or Flat File they reduced potential CPU costs by over 90%.

Corp. C had a small DB2 database containing charges for computer services - software, hardware, cpu time, programming etc. - that is used to produce monthly billings to internal customers. After seeing the results of studying this small database its owner said - 'it is obvious that if any table is to be read more than once - it should be placed in SAS data sets instead of DB2 tables'.

INTRODUCTION:

This paper is based on the included tables and each section will refer to its tables. These 3 different database applications were selected to compare the use of DB2 and SAS as the database because each database is of much different size and use.

- Corp. A - 30,000 megabytes - daily analysis by business analysts.
- Corp. B - 280 megabytes - weekly/monthly reporting, adhoc reports.
- Corp. C - 3 megabytes - monthly billing to departments.

CORPORATION A: (TABLES 1, 2, 3, 4, 5)

□ DB2 EATS MAINFRAME

Corp. A had designed a relational information repository. The extracts from it's legacy systems were to be loaded monthly into a carefully designed DBMS where analysts could study the data in order to make improved business decisions. At the final stages of testing the system in DB2 - with 5% of the data loaded -it was projected by capacity management that the system would take 100% of the CPU! Since the project was supposed to take under 10% of the CPU - something needed to be done.

□ SAS RESCUES MAINFRAME

I was asked to set up tests and perhaps design an extract from the DB2 system into SAS data sets of the most used elements - to see if the goal of 10% could be reached. As a first step I took all the DB2 tables and placed them as they were into SAS data sets. To my surprise - and to the corporations - there was a 80% - 99% reduction in CPU time accessing SAS data sets. Therefore - since the original design of the tables was what was really wanted - it was not necessary to extract from DB2 to SAS. Instead the corporate team elected to change from a information repository based on DB2 tables to one based on SAS data sets.

The system was brought into production on schedule in Oct. 1991 and as of January 1993 continued to use under 9% of the cpu - with over 100 business analysts trained in its usage - and at any moment of time 4 end-users querying the system.

For a full exploration of this system please see the proceedings of SUGI 18 where my paper on it is published, or write to me for a copy of the paper: "Why We Replaced DB2 Software with SAS software as a Relational DBMS for a 30-Gigabyte User Information System." by Stephen C. Scott.

CORPORATION B: (TABLES 1, 2, 6, 7, 8, 9, 10, 11)

Oct. 1993 was to be the date of production for Corp. B's new Human Resource system that was based on DB2 production tables.

The team I was working with in July 1993 was to design and produce a large number of reports that were produced in the previous system. We had planned to use the Corp. Rept. DBMS that was loaded every morning from production DB2 into its own DB2 tables.

□ SAS DATASETS FROM FLAT FILE

When the production DB2 system does its processing it produces a 'flat file' containing all the tables with the rows and columns that are being processed. This 'flat file' then updates DB2. Since this file contains all the information that we are reporting on - we decided to do our daily, weekly, monthly, quarterly, annual reporting directly from this file.

However, instead of have 200 reports all reading this file which would be costly in CPU time (see table 7) we decided to read the flat file once into SAS data sets - each SAS data set would correspond to a similar DB2 table - and would contain the exact production data that we needed.

□ SAS LIBRARY AUDIT SYSTEM

By designing the SAS library as a Generation Data Group we would also have an 'audit system' that would allow us to 'proof' any past reports since it would contain an 'image in time' of the production DB2 at the time of our report. The system is processed up to 3 times per week so by having 15 generations of the SAS library containing the tables - we are able to check on any questions regarding the data in our reports - even though a question may come up a week later and the data would have changed in the database itself - the SAS library would still be available with the exact image of the database as it was at that time!

□ CPU TIME REDUCED 95%

Therefore we keep three types of SAS libraries - Weekly (15 generations), Monthly (13 generations) and Adhoc (several generations). Each library contains SAS data sets with names similar to the DB2 production tables and data elements with the same columns names as used in the production package since they were 8 characters or less. By doing our reporting from SAS data sets instead of the 'flat file' our potential CPU time was reduced by over 95%!

□ DB2 FOR ADHOC QUERIES

The DB2 based Corp. Rept. DB is used for adhoc queries. I wrote a group of 'Proc Compares' that compare the production DB2 system with the Corp. Rept. DB system. When they are run early each day both systems should be in sync - thus allowing the end-users to query the Corp. Rept. DB for most questions. The end users have Platinum software queries they can run against the reporting DB2 system. I also built SAS/ACCESS views of every DB2 table so that the end users can also write SAS queries against the DB2 system using the 8 character names that they prefer instead of having to use the long DB2 names which are more difficult to type.

TESTS: Corp. B allowed me to set up and run tests of its DB2 based system and its SAS based system. The tests are presented in Tables 6-11. It is interesting that again it is shown that accessing SAS data sets provides a 90% reduction or more in CPU time.

Since most of the heavy reporting is done against SAS data sets the company realizes these savings. The adhoc queries that go against the DB2 corp. Rept. DB - if indexed - should be reasonably efficient - and even though SAS data sets could provide much savings in CPU time - the fact is that not much CPU time is used anyway so that the difference may not be readily apparent. To the wealthy there may be little difference between \$20 and \$.50.

CORPORATION C: (TABLES 1, 2, 12, 13, 14, 15)

Corp. C is a division of a government agency which sends monthly billings and detail reports to the internal customers of the agency. I became involved when they requested that I write a production SAS program to provide the detail reports - backing up a summarized report that was produced by a Cobol/DB2 program feeding a flat file into a reporting SAS program. They also requested that the Cobol be replaced by a SAS program.

□ SAS/ACCESS VIEWS FOR End USERS

My first step was to build SAS/ACCESS View descriptors that were stored in a SAS library - this allowed the end user to write Proc SQL code using the View name and the SAS column names.

□ COBOL/DB2 VS SAS/ACCESS to DB2

Table 13 shows how a 1744 line COBOL/DB2 program was replaced by a SAS Proc SQL program using Views that is 37 lines long.

For an unknown reason we could not get the Cobol program to balance correctly. Instead of wasting time attempting to debug a 1700 line program I simply rewrote its function using SAS Proc SQL in 37 lines. The new program balanced exactly. This was accomplished within half a day (writing, testing, preparing for production). (Table 15)

Needless to point out that the COBOL/DB2 program took much longer than that to produce. Note that both programs have similar CPU time since the time comes from reading DB2: 1.98 for the COBOL/DB2 program and 1.92 for the SAS/Proc SQL program.

In the final version I combined the SAS extract and the SAS reporting programs into 1 unit. The final result being one program with 334 lines. Compared to two programs combined for 2130 lines.

CONCLUSION:

1. SAS data sets are the preferred method of storing data vs DB2 or flat files.

Discussion: All of my research shows greatly reduced CPU time when accessing SAS data sets,

A. Corp. A has 91% improvement.

B. Corp. B showed 97% improvement.

C. Corp. C showed 93% improvement.

2. If data is stored in a 'flat file' or DB2 and you are accessing it more than once, then extract into a SAS data set and access the data from there.

Discussion:

A. Corp A extracts the data from its legacy systems into SAS data sets.

B. Corp B the processing flat file is extracted to SAS data sets.

3. If you are using a SAS WHERE statement to read small subsets of data based on indexed columns, the DBMS should have similar CPU times as SAS data sets with indexes. (TABLE 8)

Discussion:

A. Corp. A when selecting 1 record on an indexed column the DBMS could achieve the same CPU time as the indexed SAS data set.

B. Corp. B when selecting a small subset on an indexed column the DBMS can achieve the same CPU time as the indexed SAS data set.

4. Advantages of SAS/ACCESS View Descriptors:

Discussion:

A. Multiple WHERE conditions in your SAS programs can be passed to the DBMS.

B. Corp. B as shown in Table 8 when using where statements with the view the where is passed to the DBMS. Of course the 'where' also is optimized for SAS data sets.

C. SAS variable names and formats that are consistent can be used.

D. Corp. B has 8 character SAS variable names that are the same as the names in the production DB2 package. End users stated that they found these much easier to use than the long and overly wordy DB2 names assigned by committees to fit old Cobol technologies.

5. Disadvantage of SAS/ACCESS View Descriptors:

Discussion:

A. SELECT, VAR, DROP, and KEEP statements are generally processed by the SAS system, not by the DBMS.

B. Corp. C, Table 12, illustrates that the 'keep' is not passed to the DB2 DBMS. There is little change in CPU time whether 1 or 13 variables are kept.

Therefore if there are many variables that are not used then a view with only the variables most often used will improve performance. SAS to SAS tables, of course, always pass the select or keep.

6. SAS/ACCESS views with Proc SQL provide a highly improved programming environment over COBOL/DB2 programs. (TABLE 13)

Discussion:

A. Corp. C replaced a 1700 line COBOL/DB2 program with a 37 line Proc Sql program that was written, tested, approved and placed into production within 1/2 a day.

B. The same proc sql program that accesses DB2 by a change in JCL can access SAS data sets - with reduced CPU charges.

REFERENCES:

Scott, Stephen C. (1993), "Why We Replaced DB2 Software with SAS Software as a Relational DBMS for a 30-Gigabyte User Information System," Proceedings of the 18th Annual SAS Group International Conference, 18, 187-196.

SAS Institute, Inc. (1993), SAS Technical Report P-242, SAS Software: Changes and Enhancements, Release 6.08, p. 53-59, Cary, N.C. SAS Institute, Inc.

CONTACT:

The author will be glad to answer questions and accept suggestions at the following address:

Stephen Scott Phone: 608/276-7843
 Scott Consulting Services, Inc.
 346 Kent Ln. #302
 Madison, WI 53713

ACKNOWLEDGMENTS:

SAS, SAS/ACCESS, SAS/DB2 are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. * indicates USA registration. DB2, IMS, and VSAM are registered trademarks or trademarks of International Business Machines Corporation. Other brand and product names are registered trademarks or trademarks of their respective companies.

TABLE 1

CORP A, B, C DATABASES			
CORP	DBMS	DESCRIPTION	MEGABYTES
A	SAS	Information Warehouse Database	30,000
A	SAS	Test Database	250
B	DB2	Corporate Reporting Databases	288
B	SAS	Weekly Corp Rept DB Avg	60
B	SAS	Monthly Corp Rept DB Avg	283
C	DB2	Monthly Database	3

TABLE 2

CORP A, B, C				
CORP	Hardware	O/S	DB2	SAS
A	Amdahl 5995 - 1400A	MVS	2.3	6.07
B	IBM Model 9021	MVS	2.3	6.08
C	Hitachi Model 9021	MVS	2.3	6.07

TABLE 3 Corporation A

Table	Var's	Obs	CPU/SEC DB2	CPU/SEC SAS	% CPU Reduction
DB100	57	132,625	94	0.982	99.0
DB140	16	31,445	12	0.5	95.8
DB170	17	2,994	1	0.1	90.0
DB190	25	45,275	17	1	94.1
DB200	42	1,372,065	797	29	96.4
Totals			921	32.6	96.5

TABLE 4 Corporation A

	CPU/SE C DB2	CPU/SE C SAS	% CPU Reduction
Program 1	876	99	88.7
Program 2	161	18	88.8
Program 3	195	29	85.1
Program 4	18	6	66.7
Program 5	37	14	62.2
Total	1287	166	87.1

TABLE 5 Corporation A

Program	Var's Result	Obs	Megabytes	CPU/SE C DB2	CPU/SE C SAS	% CPU Reduction
A	67	895,198	367	1080	80	94.0
B	15	86,924		98	44	55.0
C	15	1,258,131		952	113	88.0
Total				2130	237	89.0

TABLE 6 Corporation B

Reading Tables							
Tables Reporting Database	Rows Obs	Col's Var's	Obs Length	Total Megabytes	CPU/Sec Read DB2	CPU/Sec Read SAS	Percent Savings CPU/Sec
B5	9,593	18	290	2.82	2.46	.14	92.9
Q3	7,407	29	160	1.2	2.57	.19	92.6
Q3 Prod	7,405	84	419	3.2	4.57	.26	94.3
Q3	7,987	29	160	1.3	2.89	.08	97.2
Q0	40,557	87	826	34.0	44.05	.76	98.3
R5	7,448	25	196	1.49	3.57	.08	97.8
S1	7,588	6	57	.44	1.93	.02	99.0
T1	24,840	21	153	3.84	8.83	.19	97.9
U1	145,908	23	175	25.71	58.79	1.11	98.1
U4	24,341	22	193	4.73	8.12	.20	97.5
U5	29,540	30	233	6.94	13.78	.27	98.0
V1	34,628	26	242	8.40	13.49	.31	97.7
V2	32,444	32	247	8.10	15.59	.29	98.1
Total				102	173.83	3.87	96.88
Cost @ \$.013/sec					\$22.60	\$0.50	

TABLE 7 Corporation B

Monthly/Weekly Process					
Table	OBS/ ROWS	VARs/ COLS	CPU/SEC READ FLAT FILE*	CPU/SEC READ SAS	% CPU REDUCTION
E4Q0	5293	156	6.60	.19	98.9
E4Q1	5293	107	3.78	.11	97.1
E4R5	5293	47	3.5	.07	98.0
E4S1	5293	77	3.68	.14	96.2
E4T1	11929	31	3.4	.12	96.5
E4U1	102576	43	14.46	1.01	93.0
E4U4	20869	30	3.92	.19	95.2
E4U5	20869	43	5.28	.22	95.8
E4V1	24619	33	5.17	.24	95.4
E4V2	24619	54	5.94	.28	95.3
TOTAL			55.73	2.57	96.1

Note: Flat File is VB with record lengths 172-1070, contains 247,427 records in August '93, 380,458 records in December '93.

TABLE 8

Corporation B

Test 1: Different WHERE Clauses on Q0 D.B.					
SELECT WHERE	Q0 Return Rows	KEEP Columns	Table Types		% CPU Reduction
			SAS CPU	DB2 CPU	
ALL	40,557	87	.76	42.65	98.2
ALL	40,557	1	.28	41.44	99.3
Middle Key	4	1	.19	.09	+(53)
Mid Key Range	2008	1	.22	2.20	90.0
Col Value M	24,614	87	.58	26.43	97.8
Col Value F	15,893	87	.45	17.70	97.5
Col Value COM	12,848	87	.40	14.89	97.3
Col Value SAL	7,840	87	.32	9.75	96.7
Col Date Range	8,578	87	.40	11.10	96.4
TOTAL			3.600	166.250	98.000

Note: ACCESS does pass up 'WHERE', ACCESS does not pass up 'KEEP'

TABLE 9 CORPORATION B

Test 2: Run a 'PROC FREQ' on a field in U1 using data stored in DB2 or in SAS	
DB2	59.58 CPU SEC
SAS	.9 CPU SEC
% CPU Reduction	99%

TABLE 10 CORPORATION B

Type of Select		
	U1 Table* CPU SEC	R5 Table** CPU SEC
PROC DB2 EXT	55:74	6.03
PROC SQL 'PASSTHRU'	55:34	6.03
PROC SQL 'ACCESS VIEW'	57:33	6.03
SAS DATASET	1:31	.12
% CPU Reduction	98%	98%

*V1 has 164,639 rows and 21 columns

**R5 has 14,219 rows and 23 columns

TABLE 11

Corporation B

SAS Reporting System Tables (15 Generations)		
Daily SAS Library Database Generation	Device 3390 Tracks	Size In Megabytes
0	313	17.73
-1	150	8.49
-2	175	9.91
-3	1035	58.65
-4	150	8.49
-5	153	8.67
-6	321	18.19
-7	150	8.50
-8	155	8.78
-9	1030	58.36
-10	150	8.50
-11	318	18.02
-12	149	8.44
-13	151	8.56
-14	1614	91.46
Total	6014	340.78

SAS Library (13 Generations)		
Annual/Monthly	Device 3390Tracks	Megabytes
0	4513	255.72
-1	2982	168.97
-2	5093	288.59
Total	12,588.00	713.28
Projected Totals	52,000 Tracks	3,563 Megabytes

ADHOC SAS Library (up to 9 Generations)		
Annual/Monthly	Device 3390Tracks	Megabytes
0	3016	170.90
-1	3042	172.37
Total	6,058	343.27
Projected Totals	27,000 Tracks	1,019 Megabytes

TABLE 12

Corporation C

Table	Row Obs	Cols Vars	Record Length	Approx Megabytes	CPU SEC DB2	CPU SEC SAS	% CPU Reduction
BILLNUM	6613	5	42	0.278	0.81	0.08	91%
CLASS	22	4	51	0.001	0.05	0.01	80%
CUST	173	10	158	0.027	0.07	0.01	86%
DEBCRD	192	8	87	0.013	0.07	0.01	86%
FUNDPKT	1193	9	61	0.073	0.21	0.02	91%
ROLLUP	12275	13	142	1.743	1.78	0.10	95%
TOTALS					2.99	.23	93%
Cost @ \$0.85/CPU SEC					\$2.54	\$0.19	

TABLE 13

Corporation C

SAS/ACCESS vs SAS/ACCESS PASS-THRU					
VIEW	OBS	VAR	PASS-THRU CPU/SEC DB2	ACCESS/VIEW CPU/SEC DB2	ACCESS/VIEW CPU/SEC DB2 KEEP=1
VIEWALL	12275	13	2.04	1.81	1.78
VIEW 5	12275	5	1.61	1.38	1.37
VIEW 1	12275	1	1.32	1.11	1.09

TABLE 14

Corporation C

COBOL/DB2 vs SAS/PROC SQL					
Method Type	Program	Purpose	Lines Source	Pages	CPU
1	COBOL/DB2	Extract	1744	67	1.98
1	SAS (Prog 2)	Report	386	7	1.63
Total			2130	74	3.61
2	SAS (Prog 1)	PROC SQL	37	2	1.92
2	SAS (Prog 2)	Report	386	7	1.63
Total			423	9	3.55
3	SAS (1&2)	SQL & Report	334	7	3.02

Sample SQL: Returns 1978 rows, 14 columns
CPU Time DB2: 2.14
CPU Time SAS: .67
Percent Reduction: 69%

```
PROC SQL;

CREATE TABLE REPORT AS
SELECT E.FUNDPKT,
       E.ACCSTRNG,
       E.SPLITPCT,
       D.CLASS,
       R.BILLNUM,
       R.BILLMO,
       R.BILLYR,
       R.CID,
       R.DEBCRD,
       R.BILLAMT,
       C.ITCID,
       C.CUSTOMER,
       C.CUSTNAME,
       L.ITPERCNT
FROM TEST.FUNDPKT E,
     TEST.DEBCRD D,
     TEST.ROLLUP R,
     TEST.CUST C,
     TEST.BILLNUM B,
     TEST.CLASS L

WHERE
    R.BILLMO = '07'
AND R.BILLYR = '94'
AND E.FUNDPKT = B.FUNDPKT
AND B.BILLNUM = R.BILLNUM
AND R.DEBCRD = D.DEBCRD
AND D.CLASS = L.CLASS
AND R.CID = C.CID;
```