

Why We Replaced DB2® Software with SAS® software as a Relational DBMS for a 30-Gigabyte User Information System

Stephen C. Scott, Systems Seminar Consultants

ABSTRACT:

Every company has a need for a user information system where data can be presented for analysis. This system was originally designed as a normalized relational DB2 system to contain data from various databases. The users for this system are 100 analysts from departments throughout the corporation.

However, in tests comparing access and CPU time, SAS software so outperformed the DB2 tables that the company elected to change from a relational database management system (DBMS) using DB2 software to one that uses SAS software. When tested against DB2 in accessing millions of rows of data, SAS software provided an 80 percent reduction in CPU time.

This presentation describes the test plan and results of the comparison between DB2 software and SAS software as a relational DBMS. In addition, this presentation shows how we designed a user model DBMS with SAS software that contains one percent of production and, thus, allows quick testing of complex SAS programs.

INTRODUCTION:

The database contains information on the individual customer's policies and their monthly historical activity. The relational database looks as illustrated (Table 1). It is a read only database that is updated monthly. The plan for the database will be for db100 to contain data for 5 million individuals although to begin with we will have closer to 2 million. Monthly history-activity will occur for 3 million although we will start at 1 million.

HISTORY:

Beginning in 1990 our client began studying and planning a major information repository - a user information system - decision support system - that would contain and present the corporations data in a relational data base system. Phase 1 involved the selection of a RDBMS - and after comparing and testing DB2 and Oracle® the selection committee chose DB2 because of its performance and robustness when handling large amounts of data.

By January, 1991 a project team had reviewed all of the corporate IMS® databases and VSAM® files and

selected data elements based on interviews with potential users from departments throughout the company. These data elements were tentatively arranged in about 20 tables. The pilot phase of the project was completed and the decision was made to proceed with DB2 as the engine and SAS software as the interface.

In February 1991 the company brought in DB2 consultants from a major consulting corporation to begin work of a logical model and then proceed to the physical model and the writing of the extracts that would load the database from the source IMS and VSAM databases. The logical model was approved by the technical team and the user team by mid March 1991.

By June 1991 the DB2 consultants and corporate staff had completed the extracts and constructed the model database to be used for user acceptance testing and stress testing to see the impact to resource utilization. Work was proceeding on the monthly update programs that would extract data at month end to load new information into the RDBMS.

July 1991: Systems Seminar Consultants is called in to present a 1 day seminar on SAS/ACCESS® to DB2 to the corporate staff and the DB2 consultants.

August 1991: Systems Seminar Consultants is brought in to help finalize and implement the system, specifically to finalize the SAS system's production environment - to identify the most effective and efficient method to interface SAS and DB2 to process user queries.

OVERVIEW:

□ TYPE OF PROCESSING

The usage of the product will be for global analysis - generally selecting groups of the population and comparing their history over time to other groups of people. Therefore large processing is expected instead of searching for an individual record - therefore indexing and design will be for processing 25-100 percent of the population.

□ DATA LOADED

At this point the database design and the programs to extract and load the data into DB2 tables have been completed. We have loaded 5% of production data into the databases in order to complete a test.

□ CPU USAGE - GOAL and REQUIREMENT

Our purpose includes keeping total CPU usage for this new system at near and not over 9 percent of the CPU. The CPU/system is an IBM 3090 mainframe.

□ NUMBER AND TYPE OF USERS

There will be 100 users with 30 users per day and of these 20 will be considered heavy users. Thus we expect that there will constantly be 5 users accessing large amounts of the data. Our model test will have to account for this - allowing reasonable access without going over 9 percent of the CPU.

□ TRAINING END USERS

To train the end users, System Seminar Consultants customized a three-day intermediate course in base SAS software. All exercises were designed and written with the help of end users and accessed the testing data base.

□ TEST PREPARATION

The DB2 database is already loaded. Therefore we secure the same number of disk packs and load SAS data sets from DB2. We now have two environments for testing:

1. DB2 Tables: Access vis SAS/ACCESS to DB2. For verification we will also run all programs directly against DB2 with SQL language.

2. SAS tables: Access via SAS language or SAS SQL Procedure.

□ TEST ENVIRONMENT CONSIDERATIONS:

When testing large systems it may be necessary to secure disk packs for this type of parallel test. For example each system could require four 3380 or 3390 type disk packs. Therefore for testing we had to design a test that would fit our environment and be accurate in size. From our research on potential usage we could adequately test with a subset of the production tables since the typical query would be accessing only 25% of the existing tables.

□ SAS OPTIONS for Measurement:

All programs were run with STIMER, FULLSTATS, and MEMRPT turned on. This gave us our information for analysis.

□ OTHER MEASUREMENTS:

At the same time Capacity Management was monitoring the CPU, DB2 and SAS impact on system performance during the tests.

RELATIONAL DATABASES:

(See Tables 1, 2, 3)

The normalized relational system contains 17 types of tables. They are presented in the table shown here. There are common keys for 15 of the tables, two

miscellaneous tables have keys that are found in the tables that they have arrows pointing to. Essentially the tables can be seen in three groups:

Group 1: DB100.SAS100 - contains all keys and the most sought after information or basic information that is common to all of its members. DB110.SAS110 - DB190.SAS190 contain related subsets of information that would normally be joined with PS100 depending upon the purpose of the data gathering query.

Group 2: DB300.SAS300, DB310.SAS310, etc.: These are tables that are joined to for specific reporting or analysis purposes. They contain information that may be common to thousands of members of the other tables, or provide specialized access to key data.

Group 3: DB200B00.SAS200 - DB200B60.SAS200 - Historical tables with the monthly information on all active members found in DB100. The B00 is symbolic for Back 0 months. Thus, B12 would be back 12 months and B48 would be back 48 months. These are generation data groups and a new member is created every month. DB210 - DB240 are related subsets that would normally be joined with DB200. They have the same historical structure.

Types:

1. PRELIMINARY TEST (5-10% of production) 6/1991-10/91

This database is in *Table 3*. It was designed for testing the original design.

2. PRODUCTION - 10/1991-Present

This database is in *Table 2*. It is the complete system as of 1992.

3. PRODUCTION TEST - 10/91-Present

This database is in *Table 2*. It contains 1% of production and is used for testing SAS code since all the databases have the same tables.

TESTING ENVIRONMENT:

Summary of Preliminary Data Base: (See *Table 3*)

The preliminary testing data base contained approximately 5% of production.

SAS language accessing SAS data sets provided 90% to 98% reduction in CPU time when compared to SAS/ACCESS to DB2 tables or compared to direct SQL extracts from DB2 tables.

A 75% to 88% reduction on ELAPSED time was achieved when accessing SAS tables with SAS instead of accessing DB2 tables.

Preliminary Testing: SAS language vs PROC SQL vs SAS/ACCESS to DB2
(See Table 4 and Table 5)

Three sample problems were designed. They were then written in three different ways:

1. SAS with MERGE against SAS TABLES.
2. SAS with PROC SQL against SAS TABLES.
3. SAS/ACCESS against DB2 tables.

Problem A:

Method: Join DB200 and DB210 by key1, key2, key3, key4.

Result: 895,196 observations, 67 variables, 404 length, 367 megabytes size.

Problem B:

Method: Join DB200 (where conditions are meant) with DB100 by key3 key4.

Result: 86,924 observations, 15 columns.

Problem C:

Method: Join selected rows and columns of DB100 with matching DB200 by key3 and key4.

Result: 1,258,131 observations (rows), 15 variables (columns).

Preliminary Conclusion:

For this application:

1. Both methods utilizing SAS tables are faster. SAS code offers CPU time reduction of 55% to 94%. SAS PROC SQL offers CPU reductions of 39% to 76% compared to the method using DB2 tables.
2. The method using SAS code accessing SAS tables offers 27% -75% CPU time reduction compared to SAS PROC SQL accessing SAS tables.
3. Therefore, because of our goal to reduce CPU time, we selected direct SAS code over PROC SQL for our next level of testing.

Final Testing:

□ TEST PROGRAMS:

From interviews with the future users of the system and with their help we wrote 5 equivalent programs

that would be run against each environment. (See Appendix.)

□ TEST METHOD:

Method 1: Single threaded batch.

The five model queries were run individually against DB2 and SAS and statistics were gathered for comparison.. When these were run it was the best case scene in which no other program was running against the tables.

Method 2: One Hour of Reality Stress Test.

The programs were submitted during equivalent hours of the day for SAS and for DB2.

Programs identified as 1,2,4 were submitted on the hour and programs 3,5 were submitted on the half hour.

Summary of Single Threaded Batch Test Results:
(See Table 6)

The results show a 62%-88% reduction in CPU time by accessing SAS data sets compared to accessing DB2 tables.

The overall comparison shows Batch1-Batch5 completing in 1287 CPU sec. for DB2 tables compared to 166 CPU sec. for SAS data sets. This is a 87% reduction in CPU time overall.

Summary of One-Hour Test:
(See Table 7)

Method: Jobs 1,2,4 were submitted on the hour. Jobs 3,5 were submitted on the half hour. No other jobs were running accessing the disk packs or data. This introduced the factor that more than 1 job will be running against the data at the same time.

Result:

SAS/DB2: 3 jobs cancelled after 1 hour (not completed)
2 jobs completed

SAS/SAS: 5 jobs completed in under 40 minutes.

Conclusion: The jobs run against SAS tables had clearly superior performance over the jobs run against DB2 tables. All 5 jobs completed when accessing SAS tables. Against DB2 tables three jobs were cancelled after 1 hour and on review were nowhere near complete.

The difference between the two is very significant.

From this test we would conclude that all data should be made available in SAS data sets instead of DB2 tables. In a production environment, given the concern about CPU usage, the environment

consisting of SAS data sets is clearly the best choice. Furthermore with the users of the system needing good turnaround the use of SAS tables again is the best choice.

PRODUCTION ENVIRONMENT:

How Do Users Access the System:

We constructed two CLISTS that contain the SAS CLIST and access to all the user SAS data sets.

1. PSAS607P Accesses all SAS production tables
2. PSAS607T Accesses all SAS test tables

The users thus have simple JCL that allows them to run a job to test their code and then by simply changing the "T" to a "P" run the same job against their production system.

Sample JCL and Program:

```
//JOBNAME JOB (0100,00), 'NAME', MSGCLASS=T,  
// CLASS=M, NOTIFY=USER, REGION=4046K  
//*ROUTE PRINT YOURPRINTER  
//STEP1 EXEC PSAS607T,  
// PARM=('PAGESIZE=54 LINESIZE=132  
// STIMER FULLSTATS MEMRPT')  
//SYSIN DD *
```

```
DATA TEMP100  
  SET DB100 SAS100;  
  where KEY1 = 'XXXXXXXXX',  
run;  
proc print;  
  title 'DEMONSTRATION OF PROC';  
run;
```

Creating a Production Test System Database: **(See Table 3)**

Because of the large size of the production database and the global nature of the average inquiry into the system we elected to create a Production Test Database.

Using the SAS software's RANUNI function we selected a random sample of 1% of the keys of the customers in the DB100 database. Then from the 15 main tables of the production database we selected all observations and their variables for these keys.

Two tables that have different keys were brought into the system entirely. This is necessary since they are needed by any subset of the production database.

This random 1% of our 3 million keys resulted in about 30,000 observations for our main test database.

For our History databases this would mean up to 30,000 per month times 48 or 60 months.

This allowed enough data for analysts to run their data selections against the test database and then run their PROC TABULATE or PROC SUMMARY or SAS/GRAPH® reports. The results were similar to the results from production.

The database based on the 1% random selection allows the 100 analysts to quickly test their SAS code and their reports, then make corrections or changes, and resubmit until they have everything ready for their production run.

Updating test system:

The production test system is updated every month with all changes that occurred to those keys records. Also 1% of all new customers are selected and added to the test database.

Summary of Effect of Coding Style on CPU Time: **(See Table 8)** **(See Appendix Batch 5 for example)**

The results definitely show that within SAS itself benefits can be achieved through improvements in coding styles to achieve greater efficiency. These improvements range from 15% to 34% in our five test programs.

Besides showing the effect of coding style on CPU time this table shows there is a strong correlation between the reduction in CPU time as observed in the TEST suite and the PRODUCTION suite. This is apparent when looking at the percent reduction in CPU time.

Thus in our Test system it is easy to test the effect of coding changes on CPU time and accurately predict the effect on Production CPU time.

Summary of Indexing: **(See Table 9)**

All of the SAS data sets are non indexed. We rarely have to search for a unique customer record and expected the index to slow processing. However, when we placed an index on the data set for the unique key, we achieved CPU time reduction on global as well as unique calls.

From the brief view, it is apparent that a project studying indexes on all of these data sets is warranted as significant savings are possible in CPU time.

Indexing brought 99.9% CPU time reduction when searching on the key in the 2.8 million row data set. Indexing, contrary to expectations, also improved our

processing 3% when we read all records without using any keys.

Summary of V6.06/V6.07:
(See Table 10)

The CPU time reduction of 18% in our User Information System from the new version was encouraging enough that our group of 100 analysts immediately switched from V6.06 to V6.07.

CONCLUSION:

As of May 1993 the SAS end-user RDMS has been in production for 19 months. Over 100 business analysts have been trained in accessing the data. The system meets all the criteria of Capacity Management and continues to use under 9% of the CPU.

The SAS production test database has been very well received and is used extensively. The SAS production RDMS provides information never before available to the analysts and is considered invaluable to the corporation.

Now that hundreds of ad hoc programs are being run monthly, the next stage of the project will be to analyze all usage and productionize some of the programs and develop SAS/AF[®], SAS/EIS[®] front ends for the major end users.

TABLE 1

PHYSICAL DATA MODEL OF RELATIONAL DATABASE

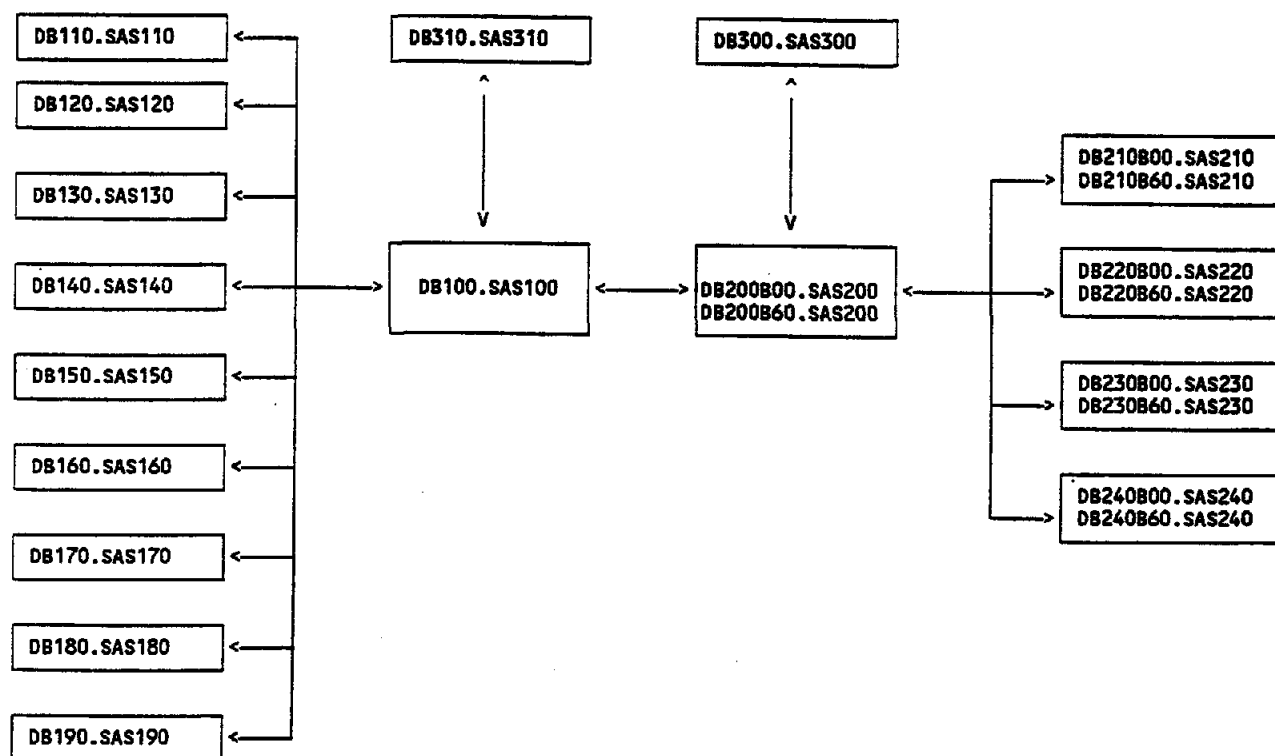


TABLE 2

SAS PRODUCTION DATABASE									
SAS DATABASE	ROWS (OBS.)	COL'S (VAR.S)	OBS LENGTH	PAGE SIZE	# PAGES	BYTES (in millions)	MAX OBS/PG	TEST ROWS	TEST PAGES
db100	2,859,555	66	276	23040	36,198	834.00	080	25,423	307
db110	315,489	04	028	23040	549	12.65	575	2,840	4
db120	967,764	10	062	23040	3,122	71.93	310	8,060	22
db130	1,112,187	15	093	23040	5,079	117.02	219	10,218	42
db140	399,256	16	065	23040	1,340	30.87	298	3,279	10
db150	20,545	09	062	23040	67	1.54	310	217	1
db160	64,276	07	032	23040	123	2.83	523	817	1
db170	119,063	17	030	23040	218	5.02	547	716	2
db180	37,282	06	046	23040	95	2.19		68	1
db190	1,069,649	25	158	23040	7,924	182.57	135	9,764	68
db300	1,540	11	060	23040	5	0.12	319	1,540	5
db310	4,000	72	394	23040	91	2.10	056	5,065	44
db200(1mo)	1,431,834	43	144	23040	9,741	224.43	147	11,513	73
db210(1mo)	835,574	31	138	23040	2,211	50.94	153	6,843	42
db220(1mo)	283,557	44	177	23040	1,283	29.56	121	2,131	17
db230(1mo)	1,148,277	30	078	23040	4,504	103.77	255	9,382	33
db240(1mo)	508,000	34	154	23040	2,188	50.41	138	4,324	30
db200(48mo)						10,772.64			
db210(48mo)						2,445.12			
db220(48mo)						1,418.88			
db230(48mo)						4,980.96			
db240(48mo)						2,419.68			

TABLE 3

PRELIMINARY AND FINAL TESTING DATABASE										
RDB	VARS	OBS	CPU (SEC) DB2	CPU (SEC) SAS/ ACCESS TO DB2	CPU (SEC) SAS	% CPU TIME REDUCED - SAS		ELAPSED TIME (MIN) SAS/ ACCESS TO DB2	ELAPSED TIME (MIN) SAS	% ELAPSED TIME REDUCED (SAS VS. SAS/ ACCESS TO DB2)
						VS. DB2	VS. SAS/ ACCESS TO DB2			
db100	57	132,825	62	94	02	97%	98%	4:32	:32	88%
db110	4	26,367		7				:23		
db120	4	8,984		3				:08		
db130	15	61,827		23				1:03		
db140	16	31,445	12	12	00.5	96%	96%	:32	:06	81%
db150	9	13,121		5				:14		
db160	7	1,175		.5				:02		
db170	17	2,994		1	00.1			:04	:03	
db190	25	45,275		17	01			:44	:40	
db200	42	1,372,065	626	797	29	95%	96%	32:09	7:57	75%
db210	31	966,388		468				21:02		
db220	44	405,677		276				12:29		
db230	30	966,388		562				24:18		
db240	34	670,517		301				12:09		
db300	11	88		.2				:03		
db310	72	4084		5				:15		

Note: SAS Version 6.06

TABLE 4

PRELIMINARY TESTING: COMPARE SAS MERGE, SAS PROC SQL, SAS/ACCESS TO DB2							
METHOD	TYPE TABLE	CPU (SEC) PROBLEM A	% CPU TIME REDUCTION PROBLEM A	CPU (SEC) PROBLEM B	% CPU TIME REDUCTION PROBLEM B	CPU (SEC) PROBLEM C	% CPU TIME REDUCTION PROBLEM C
SAS	SAS	80	94%	44	55%	113	88%
SAS PROC SQL	SAS	318	71%	60	39%	227	76%
SAS/ACCESS TO DB2	DB2	1080		98		952	

Note: SAS V6.06

TABLE 5

PRELIMINARY TESTING: COMPARE SAS MERGE, SAS PROC SQL							
METHOD	TYPE TABLE	CPU (SEC) PROBLEM A	% CPU TIME REDUCTION PROBLEM A	CPU (SEC) PROBLEM B	% CPU TIME REDUCTION PROBLEM B	CPU (SEC) PROBLEM C	% CPU TIME REDUCTION PROBLEM C
SAS	SAS	80	75%	44	27%	113	50%
SAS PROC SQL	SAS	318		60		227	

Note: SAS V6.06

TABLE 6

FINAL TEST: SINGLE THREADED BATCH TEST RESULTS					
SAS PROGRAM	TABLE TYPE	EXCP	CPU (sec)	% CPU TIME REDUCED	ELAPSED (sec)
Batch1	db2	8976	876	88.7%	1567
	sas	6580	99		1112
Batch2	db2	964	161	88.8%	321
	sas	768	18		142
Batch3	db2	1519	195	85.1%	395
	sas	2149	29		302
Batch4	db2	342	18	66.7%	108
	sas	333	6		67
Batch5	db2	422	37	62.2%	163
	sas	428	14		194
Total	db2	12323	1287	87.1%	2554
	sas	10258	166		1817

Note: SAS V6.06

TABLE 7

FINAL TEST: ONE HOUR BATCH TEST - MULTITHREADED								
SAS PROGRAM	TABLE TYPE	START	STOP	DURATION (min)	CPU TIME (min)	STATUS	EXCP	SWAPS
Batch1	db2	10:02	11:01	>58:41	>1:29	cancelled	309	10
	sas	11:03	11:28	25:25	1:30	completed	10,855	1
Batch2	db2	10:03	11:01	>58:15	>1:26	cancelled	639	6
	sas	11:04	11:13	9:25	:56	completed	5,923	4
Batch3	db2	10:34	11:01	>27:38	>1:17	cancelled	620	8
	sas	11:29	11:41	12:11	1:18	completed	9,646	2
Batch4	db2	10:04	10:19	15:53	:21	completed	649	1,213
	sas	11:04	11:05	1:09	:06	completed		
Batch5	db2	10:34	10:43	9:06	:41	completed	792	6
	sas	11:30	11:37	7:24	:34	completed	5,705	2
Total	db2	10:00	11:01	>2:49:33	>5:12	cancelled 60min	3,009	33
	sas	11:03	11:37	55:33	4:34	completed 34min	33,343	9

Note: SAS V6.06

TABLE 8

PRODUCTION BENCHMARKS: EFFECT OF CODING STYLE ON CPU TIME										
SAS DATABASE/ STYLE OF CODE	BATCH 1 CPU (SEC)	% CPU TIME REDUCED	BATCH 2 CPU (SEC)	% CPU TIME REDUCED	BATCH 3 CPU (SEC)	% CPU TIME REDUCED	BATCH 4 CPU (SEC)	% CPU TIME REDUCED	BATCH 5 CPU (SEC)	% CPU TIME REDUCED
PROD LONG PROD EFFIC	139 112	19.5%	194 165	15%	348 252	28%	47 35	26%	67 54	20%
TEST LONG TEST EFFIC	1.6 1.3	18.7%	2.2 1.8	18%	3.8 2.5	34%	.81 .65	20%	1.07 .82	23%

Note: Test data 11/92 SAS V6.07

Note: These programs were run against Production Data Base in Figure 1.

TABLE 9

PRODUCTION BENCHMARK: COMPARING INDEXING WITH NON INDEXING IN V607								
DATABASE	VERSION	SET TYPE	OBSERVATION	VARS	BYTES IN MILLIONS	CPU TIME SEC NO INDEX	CPU TIME SEC INDEX	% CPU TIME REDUCTION
DB100	PROD	ALL	2,882,209	66	800	58	56	3.4%
DB100	PROD	KEY	1	66	0	19	.03	99.9%
DB100	TEST	ALL	25,523	66	8	.59	.53	10.2%
DD100	TEST	KEY	1	66	0	.19	.03	15.8%

Note: December 1992

Note: SAS V6.07

TABLE 10

PRODUCTION BENCHMARK: COMPARE SAS V606/SAS V607 - CPU TIME						
	OBSERVATION	VARS	APPROX TOTAL BYTES (MILLIONS)	CPU SAS 606	CPU SAS 607	% CPU TIME REDUCED
SET PROD DB100	2,713,349	66	800	66	54	18%
SET TEST DB100	24,904	66	8	.69	.57	17%

Note: June 1992

APPENDIX:

FINAL TEST PROGRAMS: (SAS Versions)

The names of the variables and the databases have been changed from the actual programs - all else is the same. The programs are purposefully longer than they could have been so each step could be analyzed and so that SAS and DB2 coding has exactly the same steps.

BATCH 1

```
DATA temp200;
    set db200b00.sas200(keep=key1 key2 key3 var4 var5 var6 var7 var8 var9 var10);
run;
proc sort data=temp200; by key1 key2 key3;
run;
data temp100;
    set db100.sas100(keep=key1 key2 var11 var12 var13 var14 var15);
```

```
run;
proc sort data=temp100; by key1 key2;
run;
data tempmrg;
    merge temp100(in=a)
          temp200(in=b);
    by key1 key2;
    if A AND B;
```

BATCH 2

```
Data Temp200;
    set Db200B00.SAS200(keep=key1 key2 key3 key4 key5 key6 key7 key8 var9
    var10 var11 var12 var13);
    where key1 in('x01','x02','x03','x04','x05','x06','x07');
```

```
run;
proc sort data=temp200; by key2 key3;
run;
data temp100;
    set db100.sas100(keep=key2 key3 var4x var5x var6x var7x);
```

```
run;
proc sort data=temp100; by key2 key3;
run;
data temp170;
    set db170.sas170;
```

```
run;
proc sort data=temp170; by key2 key3;
run;
```

```
data tempabc;
    merge temp100(in=a)
          temp170(in=b)
          temp200(in=c);
    by key2 key3;
    if a and b and c;
```

BATCH 3

```
data temp200;
    set db200b00.sas200(keep=key1 key2 key3 key4 key5 key6 key7 key8 var9 var10
    var11 var12 var13);
    where key1 in('x01','x02','x03','x04','x05','x06','x07') and key2 = '01jun91'd;
```

```
run;
proc sort data=temp200; by key3 key4;
run;
```

```
data temp220;
    set db220b00.SAS220(KEEP=KEY1 key2 key3 key4 key5 key6 bvar7 bvar8 bvar9
    bvar10 bvar11 bvar12 bvar13);
    where key1 in ('x01','x02','x03','x04','x05','x06','x07') and key2 = '01jun91'd;
```

```
run;
proc sort data=temp220; by key3 key4;
run;
```

```
data temp100;
    set db100.sas100(keep=key3 key4 key5 key6 cvar8 cvar9 cvar10);
```

```
run;
proc sort data=temp100; by key3 key4;
run;
```

```
data temp190;
    set db190.sas190;
```

```
run;
proc sort data=temp190; by key3 key4;
run;
```

```
data tempinfo;
    merge temp100(in=a)
          temp190(in=b)
          temp200(in=c)
          temp220(in=d);
```

```
by key3 key4;
if a and b and c and d;
```

```
run;
```

BATCH 4

```
data temp100;
    set db100.sas100(keep=key3 key4 key5 key6 var7 var8 var9);
    where key5 = 'xxxxx';
```

```
run;
proc sort data=temp100; by key3 key4;
```

```
run;
```

```
data temp140;
    set db140.sas140;
```

```
run;
proc sort data=temp140; by key3 key4;
```

```
run;
```

```
data tempcccc;
    merge temp100(in=a)
          temp140(in=b);
    by key3 key4;
    if a and b;
```

```
run;
```

BATCH 5

```
data temp200;
    set db200b00.sas200(keep=key1 key2 key3 key4 key5 key6 key7 key8 var9 var10
    var11 var12 var13);
    where key1 in('x01')
    and key2 = '01mar91'd;
```

```
run;
proc sort data=temp200; by key3 key4;
```

```
run;
```

```
data temp100;
    set db100.sas100(keep=key3 key4 key9 key10 bvar1 bvar2 bvar3);
    where key10 = 'xxxx';
```

```
run;
proc sort data=temp100; by key3 key4;
```

```
run;
```

```
data temp140;
    set db140.sas140;
```

```
run;
```

```
proc sort data=temp140; by key3 key4;
```

```
run;
```

```
data tempcccc;
    merge temp100(in=a)
          temp140(in=b)
          temp200(in=c);
    by key3 key4;
    if a and b and c;
```

```
run;
```

Short Version BATCH 5

```
data temp200;
    merge db200b00.sas200(in=a
    keep=key1 key2 key3 key4 key5 key6 key7 key8 var9 var10 var11 var12 var13
    where=(key1 in('x01')
    and key2='01mar91'd)))
    db100.sas100(in=b keep=key3 key4 key9 key10 bvar1 bvar2 bvar3
    where=(key10='xxxx'))
    db140.sas140(in=c);
    by key3 key4;
    if a and b and c;
```

```
run;
```

SAS, SAS/ACCESS, SAS/AF, SAS/EIS, and SAS/GRAPH 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. Oracle is a registered trademark or trademark of Oracle Corporation. Other brand and product names are registered trademarks or trademarks of their respective companies.

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

Stephen Scott Work: 608/222-7081
Systems Seminar Consultants FAX: 608/222-7310
5011 Monona Drive
Madison, WI 53716