



Supercharge Your SAS/ACCESS Queries One Option at a Time

Super Demo - 314

#SASGF

Supercharge Your SAS/ACCESS® Queries One Option at a Time

"This program should run **much faster**! What can I do to make it run like a Ferrari?"

We've all been there; it is painful. Fortunately, SAS/ACCESS® software provides the tools you need to make your database queries run fast. This presentation details a **handful of tricks** used by the SAS/ACCESS masters to make data access sing. You will be able to go home and **impress all your friends**.

A HANDFUL of Tricks

Write Speed

- **INSERTBUFF=**

```
LIBNAME snowslow ODBC DSN='snowflake_DSN'  
SCHEMA=SGFDEMO USER=jbailey PW=somePW  
;
```

```
DATA snowslow.testtab;  
    SET work.testtab;  
RUN;
```

A HANDFUL of Tricks

Write Speed

- **INSERTBUFF=**

```
LIBNAME snowfast ODBC DSN='snowflake_DSN'  
SCHEMA=SGFDEMO USER=jbailey PW=somePW  
INSERTBUFF=32767;
```

```
DATA snowfast.testtab;  
    SET work.testtab;  
RUN;
```

A HANDFUL of Tricks

Write Speed

R U Committed?

- INSERTBUFF=
- **DBCOMMIT=**
- **AUTOCOMMIT=**

```
LIBNAME snowfast ODBC DSN='snowflake_DSN'  
SCHEMA=SGFDEMO USER=jbailey PW=somePW  
INSERTBUFF=32767  
DBCOMMIT=32767 AUTOCOMMIT=no;
```

```
DATA snowfast.testtab;  
    SET work.testtab;  
RUN;
```

A HANDFUL of Tricks

Write Speed

R U Committed?

Read Speed

- INSERTBUFF=
- DBCOMMIT=
- AUTOCOMMIT=
- **READBUFF=**

```
LIBNAME snowfast ODBC DSN='snowflake_DSN'  
SCHEMA=SGFDEMO USER=jbailey PW=somePW  
INSERTBUFF=32767 READBUFF=32767  
DBCOMMIT=32767 AUTOCOMMIT=no;
```

```
DATA snowfast.testtab;  
    SET work.testtab;  
RUN;
```

A HANDFUL of Tricks

Write Speed

R U Committed?

Read Speed

What's the Query?

```
OPTIONS SASTRACE=',,,d' SASTRACELOC=saslog  
NOSTSUFFIX;
```

A HANDFUL of Tricks

Write Speed

R U Committed?

Read Speed

What's the Query?

```

01 explain select * from eecdata.order_fact a, eecdata.order_fact b;
02
03 1) First, we lock a distinct eecdata."pseudo table" for read on a
04 RowHash to prevent global deadlock for eecdata.a.
05 2) Next, we lock eecdata.a for read.
06 3) We execute the following steps in parallel.
07 1) We do an all-AMPs RETRIEVE step from eecdata.a by way of an
08 all-rows scan with no residual conditions into Spool 2
09 (all_amps), which is built locally on the AMPs. The input
10 table will not be cached in memory, but it is eligible for
11 synchronized scanning. The result spool file will not be
12 cached in memory. The size of Spool 2 is estimated with low
13 confidence to be 798,865,024 rows (190,928,740,736 bytes).
14 The estimated time for this step is 8 hours and 14 minutes.
15 2) We do an all-AMPs RETRIEVE step from eecdata.b by way of an
16 all-rows scan with no residual conditions into Spool 3
17 (all_amps), which is duplicated on all AMPs. The input table
18 will not be cached in memory, but it is eligible for
  
```


A HANDFUL of Tricks

Write Speed

R U Committed?

Read Speed

What's the Query?

```

01 explain select * from eecdata.order_fact a, eecdata.order_fact b;
02
03 all-rows scan, which is joined to Spool 1 (Last Use) by a
04 all-rows scan. Spool 2 and Spool 3 are joined using a product
05 join, with a join condition of ("(1=1)"). The result goes into
06 Spool 1 (all_amps), which is built locally on the AMPs. The
07 result spool file will not be cached in memory. The size of Spool
08 1 is estimated with low confidence (estimated size is 6,095,345,663
09 bytes). The estimated time for this step is 6,095,345,663 hours and 28
10 minutes.
11
12 5) Finally, we send out an END TRANSACTION step to all AMPs involved
13 in processing the request.
14
15 -> The contents of Spool 1 are sent back to the user as the result of
16 statement 1. The total estimated time is 6,095,345,705 hours and
17 25 minutes.

```

A HANDFUL of Tricks

Write Speed

R U Committed?

Read Speed

What's the Query?

DBIDIRECTEXEC

- Push CTAS, UPDATE, INSERT, and DELETE into the DBMS

```
OPTIONS DBIDIRECTEXEC; /* Turns it ON */
```

```
OPTIONS NOBIDIRECTEXEC; /* Turns it OFF */
```

```
LIBNAME myTera TERADATA ...;
```

```
PROC SQL;
```

```
    CREATE TABLE myTera.tab AS
```

```
        SELECT * FROM myTera.tab1;
```

```
QUIT;
```

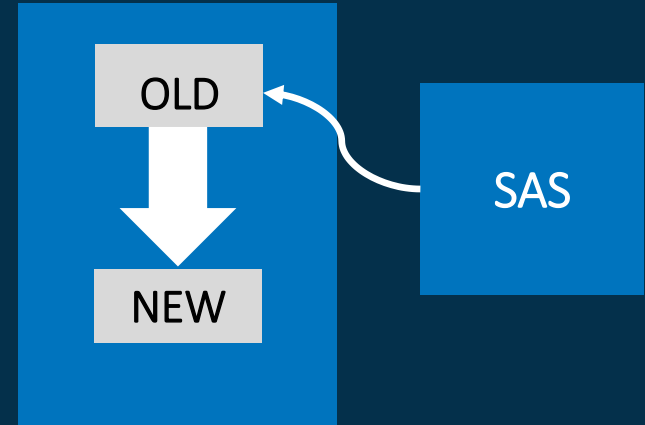
“Beware of the Illusion of Success™ ...

It is possible for your code to work, and not work, at the same time

```
OPTIONS SASTRACE=',,,d' SASTRACELOC=saslog NOSTSUFFIX;  
OPTIONS SQL_IP_TRACE=note MSGLEVEL=i;  
  
OPTIONS DBIDIRECTEXEC;
```

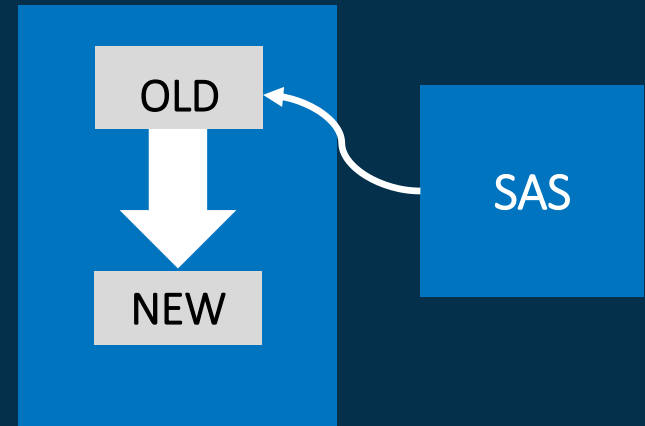
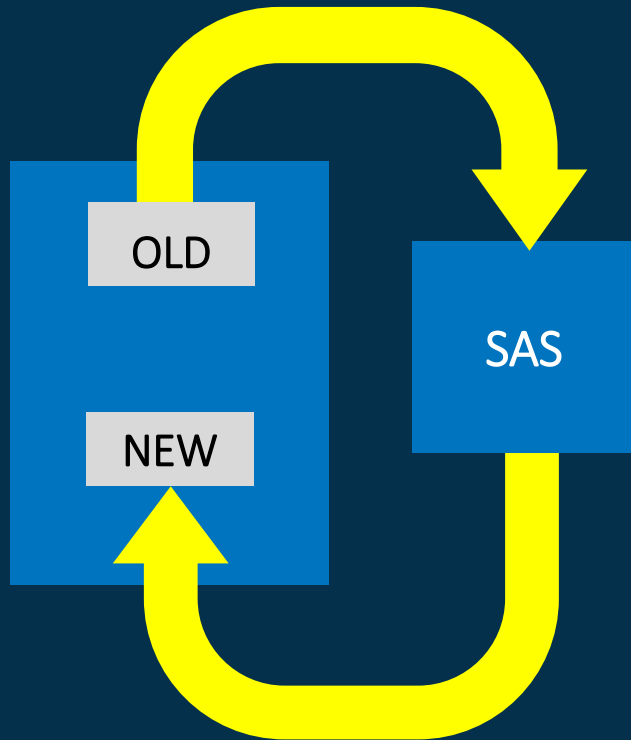
SUCCESS on Parade

DBIDIRECTEXEC



The ILLUSION of SUCCESS on Parade

NODBIDIRECTEXEC DBIDIRECTEXEC



A HANDFUL of Tricks

Write Speed

R U Committed?

Read Speed

What's the Query?

DBIDIRECTEXEC

- **Amazon Redshift**
- Aster
- DB2 under UNIX and PC Hosts
- DB2 under z/OS
- Greenplum,
- Hadoop
- HAWQ
- Impala
- Informix,
- Microsoft SQL Server
- MySQL
- Netezza
- ODBC
- OLE DB
- Oracle
- PostgreSQL
- SAP ASE
- SAP HANA
- SAP IQ
- Teradata
- Vertica

Why is Amazon Redshift **GREEN**
and the others **RED?**

The End!

<https://www.linkedin.com/in/jeffreydbailey>

<https://github.com/Jeff-Bailey>