SAS and SQL databases
Understanding and tuning queries and settings
Under the covers
The query undergoes multiple steps

**SAS code**
- Implicit or Explicit passthrough.
- Handwritten or generated by a task.

**SAS/Access engine**
- Re-expresses implicit passthrough as native SQL for the database

**ODBC Driver**
- Attempts to optimise the query for the database

**Database**
- Attempts to optimise the received query

**Is the code efficient?**

**Are the options correct?**

**Are the options correct?**

**Is it the right driver?**

**Are indexes set appropriately?**
What about metadata libraries?
Metadata adds another layer of abstraction

Using Base v9 Engine
Libname mktg v9 "C:\marketing";

Using Meta engine
Libname demo meta library="Marketing data";
Retrieve assignment information and construct libname statement

<table>
<thead>
<tr>
<th>Library</th>
<th>Libname</th>
<th>Engine</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing data</td>
<td>Mktg</td>
<td>V9</td>
<td>C:\marketing</td>
</tr>
</tbody>
</table>

Libname mktg v9 "C:\marketing";

Assign library

V9 engine

Data
Under the covers
There is a nesting of metadata leading to the data

SAS Server metadata
Points to the host running the database server

SAS Connection metadata
Points to the server metadata.
Indicates the ODBC DSN or connection string, and authentication method

SAS Library metadata
Points to the SAS Server which will execute the libname statement
Points to the server and connection metadata objects
Is used to generate the libname statement

SAS Table metadata
A metadata item for each database table registered to this connection

ODBC DSN
Points to the database schema

RDBMS

RDBMS tables
Assignment

Initialisation delays

• Libraries are often pre-assigned as a convenience feature
  • Making multiple connections will slow down server initialisation. Consider setting deferred connections (defer=yes) Defer= can be set on the Connection tab

• Engine options
  • Native
    - When using the native engine no SAS metadata security is applied
  • Meta
    - When using the META engine, metadata security is applied
  • External
    - Uses an autoexec to define the connection
    - Metadata does not generate a connection
Library

Advanced options - Connection

Type of connection: SHAREDREAD
Connection group name: 
User-defined connection initialization command: 
User-defined connection termination command: 
User-defined library initialization command: 
User-defined library termination command: 
Whether to defer a connection until needed: No
Whether to automatically drop utility connections: No
Login timeout, in seconds: 
Library

Advanced options - Bulkload
### Advanced Options - Optimisation

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block insert buffer size:</td>
<td>1</td>
</tr>
<tr>
<td>Block read buffer size:</td>
<td>0</td>
</tr>
<tr>
<td>Pass functions to the DBMS that match those supported by SAS:</td>
<td></td>
</tr>
<tr>
<td>Pass DELETE to the DBMS:</td>
<td>Blank (no value)</td>
</tr>
<tr>
<td>Whether to use indexes:</td>
<td>Yes</td>
</tr>
<tr>
<td>Whether to check for null keys when generating WHERE clauses:</td>
<td>Yes</td>
</tr>
<tr>
<td>Multi data source optimization:</td>
<td>NONE</td>
</tr>
<tr>
<td>Whether to use data source's SQL insert:</td>
<td>Blank (no value)</td>
</tr>
<tr>
<td>Whether to use Current-of-Cursor SQL to update/delete rows:</td>
<td>Blank (no value)</td>
</tr>
<tr>
<td>Whether to create a spool file for two-pass processing:</td>
<td>Yes</td>
</tr>
<tr>
<td>Threaded DBMS access:</td>
<td>(THREADED_APPS, 3)</td>
</tr>
</tbody>
</table>
Optimization Tab - ODBC Libraries
options to optimize the access to the database table

• Block insert buffer size
  • specifies the number of rows in a single Insert operation. Corresponds to the INSERTBUFF= option in the LIBNAME statement.

• Block read buffer size
  • specifies the number of rows of DBMS data to read into the buffer. Corresponds to the READBUFF= option in the LIBNAME statement.

• Pass functions to the DBMS that match those supported by SAS
  • specifies whether functions are passed to ODBC that are normally not passed. Use of this option can cause unexpected results. Corresponds to the SQL_FUNCTIONS= option in the SAS/ACCESS LIBNAME statement.

• Pass DELETE to the DBMS
  • specifies that an SQL delete statement is passed directly to the DBMS for processing. Selecting this option improves performance, because SAS does not have to read the entire result set and delete one row at a time. Corresponds to the DIRECT_EXE= option in the SAS/ACCESS LIBNAME statement.
Efficient SAS/SQL code
Efficient SAS/SQL code

- Optimisation has four competing factors
  - CPU
  - Memory
  - I/O (disk and network)
  - Disk space
Efficient SAS/SQL code

Overview

• Basic principles:
  • Don't do more work than you need to
  • Optimise for your environment
  • Go for the quick wins first
    - Jobs which take the most time
    - Jobs which are run most often
  • Benchmark after each change
Efficient SAS/SQL code

Overview

Data Integration Studio produces detailed information on resources used and time taken by each step.

In base SAS and SAS Enterprise Guide use **options fullstimer** to get more information on the log.
Efficient SAS/SQL code

Overview

- NOTE: The data set WORK.SAMPLE2 has 317223 observations and 27 variables.
- NOTE: DATA statement used (Total process time):
  - real time 2.10 seconds
  - cpu time 0.28 seconds

- NOTE: The data set WORK.SAMPLE2 has 317223 observations and 27 variables.
- NOTE: DATA statement used (Total process time):
  - real time 0.27 seconds
  - cpu time 0.28 seconds

- NOTE: The data set WORK.SAMPLE2 has 317223 observations and 27 variables.
- NOTE: DATA statement used (Total process time):
  - real time 0.27 seconds
  - cpu time 0.28 seconds

On the second and third runs the input data set was cached so the runtime was much lower. On a larger data set this effect will be reduced.
Proc SQL options

- **_TREE _METHOD**
  - Displays a diagram of the SQL query plan
  - Useful in optimising a query

- **BUFFERSIZE / UBUFSIZE=1000000**
  - Specifies the number of rows SAS reads in one IO
  - Ideally, should fit the smallest table in a query entirely into memory

- **Readbuff=** and **Writebuff=**
libname demo odbc (dsn=demo);
proc sql;
  select *
  from demo.customers
  where upcase(LN) ="SMITH";
quit;

proc sql;
  connect to odbc (dsn=demo);
  select *
  from connection to odbc
    (select *
     from customers
     where upper(LN)='SMITH'
    )
  ;
quit
Implicit SQL Passthrough
Implicit SQL before and after passthrough

```
proc sql;
  select *
  from demo.customers
  where upcase(LN) = "SMITH";
quit;
```

```
select *
from customers
where upper(LN) = 'SMITH'
```

Use the SASTRACE option to see the code passed through by the library engine

```
options sastrace=',,d' sastraceloc=saslog nostsuffix;
```
Passthrough on "Non - SQL" SAS code

- Any data step or proc step in SAS will attempt to pass through as much code as possible
  - Where clauses
  - Calculations
  - Functions
  - Summarisation
  - Sorting
- For example a `proc sort` may be automatically re-expressed as SQL
  ```sql
  select * from..
  order by..
  ```
ODBC Driver considerations
ODBC Driver

- The correct ODBC driver can make a 10x difference in performance
- The tuning options will vary with each driver
- The Database Administrator should be involved in tuning, to show the code which is actually running and advise on tuning.
Database considerations
Indexes
Efficient subsetting of large data

- Index tables which are:
  - Large
  - Read multiple times between updates
  - Usually sampled or subset

- Index variables which are:
  - Used in filters
  - Have high cardinality

Use the MSGLEVEL option to see the information on index usage

```options msglevel=i;```
Sensible starting values for SAS options

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>bufno</td>
<td>128 then tune up or down</td>
</tr>
<tr>
<td>bufsize</td>
<td>0 (zero) or 128k</td>
</tr>
<tr>
<td>memsize</td>
<td>Depends on concurrency, data size, physical memory. ~ 4G?</td>
</tr>
<tr>
<td>sortsize</td>
<td>1G, or 75% of memsize - whichever is largest</td>
</tr>
<tr>
<td>maxmemquery</td>
<td>0 (zero)</td>
</tr>
</tbody>
</table>
THANK YOU
Wrap-up

• Ideas for future sessions, SAS Forum UK, any speaker volunteers
• Re-vamped Customer Loyalty website
• Any other business

Thank you for joining us today, we appreciate your time and participation