SAS Querying Guidelines

## Keywords, Symbols, and Operators

Keywords should be uppercase. Examples include: DATA, PROC SQL, RUN, SELECT, DISTINCT, TOP, JOIN, ON, IN, NOT

In general, the AS keyword should be included, and un-needed keywords and symbols should be excluded. Examples include: OUTER

Parenthesis should be excluded from table joins and always included when using OR.

Operators should be surrounded by spaces. Not-equal should be denoted with <> rather than != in SAS. Use != for Not-equal in SQL queries.

## Table and Column Aliases

When possible, use an acronym for the table aliased. (e.g. INNER JOIN LN10\_LON **LN10**; LEFT JOIN LN16\_LON\_DLQ\_HST **LN16**) When not possible, use PascalCase for table aliases.

Avoid using single quote aliasing, and bracket aliasing (where possible):  
SELECT

SUM(LA\_CUR\_PRI) AS Balance, --preferred

SUM(LA\_CUR\_PRI) 'Balance', --avoid

SUM(LA\_CUR\_PRI) [Balance], --avoid

SUM(LA\_CUR\_PRI) [Total Balance] --ok

FROM

CDW..LN10\_LON

WHERE

BF\_SSN = '000000000'

## System Functions

System functions should be uppercase. User-defined functions should be PascalCase. Examples include: ISNULL, COALESCE, SUM, DATEADD

## Data Sets

Data set names should follow table naming conventions.

## Object References

Query object references (column and table names) should match the case of the database design. Since table names and attributes are uppercase in the data model, everything should be uppercase.

Always include the table alias along with the column name. (e.g SELECT **LN10**.BF\_SSN, **LN10**.LN\_SEQ FROM LN10\_LON **LN10**)

## Magic Numbers

Magic numbers should be avoided but AES uses them so when referring to a specific value, include a comment that indicates the meaning behind the value. (e.g. WHERE LC\_STA\_LON16 = 1 **-- Active**)

## Query Optimization

Use most modern syntax if feasible. I.E. RAISERROR VS TRY / CATCH

Where possible, use the entire primary key of a table in the join/search criteria of a query to ensure proper key usage by the sql query optimizer.

DECLARE @Table TABLE() - cant index. Don’t use for larger data sets than ~1k records as it is stored in memory.  
  
CREATE TABLE #Table () – can index. Preferred for larger data sets as the data is stored on tempdb rather than memory.

Consider SARGability (search argument ability) when writing queries such that indexes can be used on tables. Prefer inclusive lists to exclusive (IN over NOT IN), make sure all manipulation is being done on the right hand side of a comparison, and avoid it on the left hand side where feasible I.E.

SomeVariable < DATEADD(DAY,1,GETDATE()) vs DATEADD(DAY,1,GETDATE()) > SomeVariable

-refer to <https://www.sqlshack.com/how-to-use-sargable-expressions-in-t-sql-queries-performance-advantages-and-examples/>

When joining tables, the current table should come first in the join criteria.

CDW..LN10\_LON LN10

INNER JOIN CDW..LN90\_FIN\_ATY LN90

ON LN90.BF\_SSN = LN10.BF\_SSN --Correct order

AND LN10.LN\_SEQ = LN90.LN\_SEQ --avoid reversing table order

## Query Formatting

When formatting a query, readability by the team is a high priority. Generally this means indenting and putting each item on its own line when a statement has multiple arguments. If a data step has multiple arguments, each item must be indented and placed on its own line. Single argument data steps may be written as a single line.

In a comma delimited select statement, the commas should be at the end of the line, rather than the start of the line.

Parenthesis can be flush with the join, or tabbed in 1 level, according to personal preference.

INNER JOIN

(

)

INNER JOIN

(

)

## Example Data Step

DATA LEGEND.SingleArg; SET SingleArg; RUN;

DATA LEGEND.MultiArg;

SET MultiArg;

WHERE

VALUE = “Active” /\*Gets only active records\*/

;

RUN;

Example SAS Query

PROC SQL;

CONNECT TO DB2 (DATABASE=&DB);

CREATE TABLE INITIAL\_POPULATION AS

SELECT

DB.\*

FROM

POPULATION\_IN PIN

INNER JOIN CONNECTION TO DB2

(

SELECT DISTINCT

PD10.DF\_SPE\_ACC\_ID,

RTRIM(PD10.DM\_PRS\_1) || ' ' || RTRIM(PD10.DM\_PRS\_LST) AS BRW\_NAME,

PD10.DM\_PRS\_1,

PD10.DM\_PRS\_LST,

PD30.DX\_STR\_ADR\_1,

PD30.DX\_STR\_ADR\_2,

PD30.DM\_CT,

PD30.DC\_DOM\_ST,

PD30.DF\_ZIP\_CDE,

PD30.DM\_FGN\_CNY,

PD32H.DX\_ADR\_EML AS HOME\_EML,

AY10.LN\_ATY\_SEQ,

PD10.DF\_PRS\_ID

FROM

PKUB.PD10\_PRS\_NME PD10

INNER JOIN PKUB.AY10\_BR\_LON\_ATY AY10

ON PD10.DF\_PRS\_ID = AY10.BF\_SSN

INNER JOIN PKUB.PD30\_PRS\_ADR PD30

ON PD10.DF\_PRS\_ID = PD30.DF\_PRS\_ID

AND PD30.DI\_VLD\_ADR <> 'Y'

AND PD30.DC\_ADR = 'L'

LEFT JOIN PKUB.PD32\_PRS\_ADR\_EML PD32H

ON PD10.DF\_PRS\_ID = PD32H.DF\_PRS\_ID

AND PD32H.DC\_ADR\_EML = 'H'

AND PD32H.DI\_VLD\_ADR\_EML = 'Y'

AND PD32H.DC\_STA\_PD32 = 'A'

WHERE

DAYS(AY10.LD\_ATY\_REQ\_RCV) = DAYS(CURRENT\_DATE) - 1

FOR READ ONLY WITH UR

) DB

ON DB.BF\_SSN = PIN.DF\_PRS\_ID

;

DISCONNECT FROM DB2;

QUIT;

/\*export to Excel spreadsheet\*/

PROC EXPORT

DATA = DCR\_DATA

OUTFILE = "Q:\Support Services\Test Files\DCR AI-12"

DBMS = EXCEL

REPLACE;

SHEET = "A";

RUN;

DATA DCR; SET DUSTER.DCR; RUN;

DATA DCR (DROP = BR\_DATE\_OF\_BIRTH LD\_BIL\_DU\_LON);

SET DCR;

BR\_DATE\_OF\_BIRTH = PUT(BR\_DATE\_OF\_BIRTH,MMDDYY10.);

IF WEEKDAY(LD\_BIL\_DU\_LON) = 1 THEN EFF\_DATE = '';

ELSE IF WEEKDAY(LD\_BIL\_DU\_LON) = 7 THEN

DO;

EFF\_DATE = PUT(LD\_BIL\_DU\_LON,MMDDYY10.) + 1;

APP\_DATE = PUT(LD\_BIL\_DU\_LON,MMDDYY10.) + 2;

END;

ELSE EFF\_DATE = PUT(LD\_BIL\_DU\_LON,MMDDYY10.);

RUN;

PROC PRINT NOOBS SPLIT='/' DATA=DEMO WIDTH=UNIFORM WIDTH=MIN LABEL;

FORMAT

DF\_SPE\_ACC\_ID $10.

LN\_SEQ BEST2.

LD\_LON\_GTR MMDDYY10.

;

VAR

DF\_SPE\_ACC\_ID

LN\_SEQ

LD\_LON\_GTR

;

LABEL

DF\_SPE\_ACC\_ID = 'Account Number'

LN\_SEQ = 'Loan Sequence Number'

LD\_LON\_GTR = 'Guarantee Date'

;

RUN;

PROC PRINT NOOBS SPLIT = '/' DATA = DEMO WIDTH = UNIFORM WIDTH = MIN LABEL;

PROC PRINT

NOOBS SPLIT = '/'

DATA = DEMO

WIDTH = UNIFORM

WIDTH = MIN

LABEL;

FORMAT

DM\_PRS\_1 DF\_SPE\_ACC\_ID $10.

LN\_SEQ BEST2.

DD\_BRT LD\_LON\_GTR MMDDYY10.

;

FORMAT

DM\_PRS\_1

DF\_SPE\_ACC\_ID $10.

LN\_SEQ BEST2.

DD\_BRT

LD\_LON\_GTR MMDDYY10.

;

FORMAT

DM\_PRS\_1 $10.

DF\_SPE\_ACC\_ID $10.

LN\_SEQ BEST2.

DD\_BRT MMDDYY10.

LD\_LON\_GTR MMDDYY10.

;

PROC PRINTTO PRINT=REPORT2 NEW; RUN;

PROC PRINTTO

PRINT = REPORT2

NEW;

RUN;

OPTIONS

ORIENTATION = LANDSCAPE

PS = 39

LS = 127

;

OPTIONS ORIENTATION=LANDSCAPE PS=39 LS=127;

FILE 'T:\SAS\CSV OUTPUT.txt' DELIMITER=',' DSD DROPOVER LRECL=32767;

FILE

'T:\SAS\CSV OUTPUT.txt'

DELIMITER = ','

DSD

DROPOVER

LRECL = 32767

;

Example SQL Query

SELECT

E.employee\_id,

E.first\_name AS FirstName,

E.last\_name AS LastName,

E.email,

EDC.department\_count AS [Department Count],

ISNULL(PS.parking\_stall, 'N/A') AS [Parking Stall]

FROM

EMPLOYEES E

INNER JOIN

(

SELECT

ED.employee\_id,

COUNT(ED.employee\_id) AS department\_count

FROM

EMPLOYEE\_DEPARTMENTS ED

WHERE

ED.department\_id != 3 --Internship

GROUP BY

ED.employee\_id

HAVING

COUNT(ED.employee\_id) > 1

) EDC

ON EDC.employee\_id = E.employee\_id

INNER JOIN EMPLOYEE\_DEPARTMENTS ED

ON ED.employee\_id = E.employee\_id

AND ED.active = 1 --current employee

AND ED.updated\_at > DATEADD(mm, 1, GETDATE())

INNER JOIN DEPARTMENTS D

ON D.department\_id = ED.department\_id

LEFT JOIN PARKING\_STALLS PS

ON PS.parking\_stall\_id = E.parking\_stall\_id

WHERE

E.email like '%@admin.com' --Always include administrators

OR

(

E.active = 1 --current employee

AND E.email NOT LIKE '%@test.com' --exclude test employees

)

ORDER BY

E.last\_name,

E.first\_name