

Automated Generation of SAS® Access Views from Oracle Tables

Xinsheng Lin, IMS America, Plymouth Meeting, PA
Sheng Luo, Providian Corporation, Frazer, PA

Abstract

The authors have been involved in creating large SAS Access Views from Oracle database (about 30 tables and hundreds of columns) while the Oracle tables are still in development stage. That has required constant change in the SAS Access View as result of constant change in the Oracle tables, including the number and names of tables and the number, names and formats of the data columns. Heavy typing is involved. Coding errors occur often. It generally takes several rounds of coding and checking to make SAS codes correct. This means headaches for programmers.

To reduce manual labor and increase productivity, authors have created a SQL script and a SAS program to automatedally generate SAS Access Views from Oracle tables, as well as the documentation of the views. The purpose of this paper is to describe the automated process, hoping that this process could benefit other SAS View developers working with Oracle, Sybase or other relational databases.

Problems

The primary author has been involved in creating large SAS Access Views from her company's Oracle database of medical research, when the Oracle database is still under development. The number and names of Oracle tables and the number, names and formats of data columns keep changing constantly. As a result, frequent change of the SAS codes for Access View generation is required. Because the numbers of tables and data columns are large, heavy typing is seemingly unavoidable. In order to eliminate heavy typing and to increase productivity in SAS coding, authors have generated a simple process that can create SAS Access Views and their documentation from Oracle tables quickly and accurately.

The process involved 3 steps. The first step is to use the SQL script to generate name lists of Oracle

tables and data columns. The next step is to use the SAS program created by authors to read in Oracle name lists, to produce another SAS program for SAS View generation and to generate the documentation of the Access Views. The last step is to execute the SAS program obtained from step 2 to produce SAS Access Views.

The next section will demonstrate the Oracle SQL script and the SAS program, and provide more detailed discussion of this automated process.

Automated Process

In IMS, the Oracle database and SAS Access Views resides on UNIX system. The SQL script and the SAS program described in this section reflect this feature. However other users can easily change these codes to adapt their own system environment. In addition, the example shown here only involves seven tables and small number of columns, in order to save space in this paper. The process, however, gives more advantage when dealing with large numbers of tables and columns.

The first step of the process is to execute the Oracle SQL script named 'oracle_tables.sql' as shown in Box 1 below.

Box 1: Oracle SQL Script

```
rem oracle_tables.sql
rem to create and run table.lst

set feedback off
set heading off

spool table.lst
select 'desc '||Table_Name||';'
  from all_tables
  where owner='XINSHL';
spool off

spool variable.lst
start table.lst
spool off
```

The execution of this SQL script will result in 2 files. One file, named 'table.lst', contains names of all the tables as shown in Box 2.

Box 2: Contents of the File 'table.lst'

```
desc DIAGNOSIS;  
desc DOCTOR;  
desc EMR_OPR_DATE;  
desc MASTER_DIAGNOSIS;  
desc MASTER_DRUG;  
desc PATIENT;  
desc THERAPY;
```

Another file, named 'variable.lst', contains the names and format of data elements for each table, separated from another by the headings and dashed lines(see Appedix 1). Even though the 'variable.lst' does not contain any table names, the sequential order of the tables in 'variable.lst' follows exactly with that in the 'tables.lst'.

After name lists of the Oracle tables are created, the second step is to execute the SAS program, named 'autoview.sas', as presented in Appendix 2.

This program has 4 major modules. The first module starts by reading in Oracle column names and their formats from the 'variable.lst'. Since the Oracle column list is in variable length, MISCOVER option and ':' input modifier are used to read in 'variable.lst'. The module then assign table number for each column name, and create SAS variable name for each Oracle column name. The rule of naming SAS variables is to take first 8 bytes of Oracle column names. However if it results in duplication of SAS variable names, the first 7 bytes of Oracle column name plus a distinct number at the 8th byte will be assigned to the SAS variable name. The second module read in Oracle table list in the same fashion as in Oracle column list, assign table number for each table, and create SAS member name for each Oracle table name. The rule of naming SAS member names is the same as that of naming SAS variables. The third module is to combines SAS member names and variable names together, then to output a SAS program named 'sascode.sas'. The 'sascode.sas' is exactly the program that we need for SAS Access Views generation. The last module simply applies PROC PRINT to output the documentation for the SAS

Views. Samples of 'sascode.sas' and 'autoview.log' are displayed in Appendices 3 and 4.

Finally, to create SAS Access Views, you just need to run 'sascode.sas' program. The execution of the 'sascode.sas' can be done by either including 'sascode.sas' at the end of 'autoview.sas' or running it separately.

REFERENCES

SAS Institute Inc.(1995), SAS/ACCESS Software for Relational Databases, Reference Version 6, First Edition, Cary, NC:SAS Institute Inc.

SAS Institute Inc. (1990), SAS Language, Reference Version 6, First Edition, Cary, NC:SAS Institute Inc.

George Koch and Kevin Loney (1995), ORACLE: The Complete Reference, Third Edition, Berkeley, CA: Osborne McGraw-Hill.

SAS is a registered trademark or trademark of the SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

Author Contact

Xinsheng Lin
IMS America LTD
660 W. Germantown Pike
Plymouth Meeting, PA 19462-0905
Phone: (610) 832-5532
Email: xinshl@imsint.com

Sheng Luo
Providian Corporation
20 Moores Road
Frazer, PA 19355
Phone: (610) 648-5065
Fax: (610) 648-5614
Email: sheng_luo@providian.com

Appendix 1: Contents of the File 'variable.lst'

Name	Null?	Type
-----	-----	-----
PATIENT_ID	NOT NULL	NUMBER(5)
DOCTOR_ID	NOT NULL	NUMBER(4)
DIAGNOSIS_DATE		DATE
ICD9_CODE	NOT NULL	VARCHAR2(5)
Name	Null?	Type
-----	-----	-----
DOCTOR_ID	NOT NULL	NUMBER(4)
SPECIALTY_CODE		VARCHAR2(3)
DOC_BIRTH_DATE		DATE
DOC_GENDER		VARCHAR2(1)
Name	Null?	Type
-----	-----	-----
AS_OF_DATE		DATE
Name	Null?	Type
-----	-----	-----
ICD9_CODE	NOT NULL	VARCHAR2(5)
DIAGNOSIS_DESCRIPTION		VARCHAR2(50)
ACUTE_CHRONIC_INDICATOR		VARCHAR2(1)
Name	Null?	Type
-----	-----	-----
DRUG_ID	NOT NULL	NUMBER(10)
DRUG_DESCRIPTION		VARCHAR2(30)
DRUG_AVERAGE_COST		NUMBER(7,2)
Name	Null?	Type
-----	-----	-----
PATIENT_ID	NOT NULL	NUMBER(5)
OWNING_DOCTOR_ID	NOT NULL	NUMBER(4)
PT_BIRTH_DATE		DATE
PT_GENDER		VARCHAR2(1)
MARITAL_STATUS		VARCHAR2(1)
INSURANCE_NAME		VARCHAR2(20)
Name	Null?	Type
-----	-----	-----
PATIENT_ID	NOT NULL	NUMBER(5)
DOCTOR_ID	NOT NULL	NUMBER(4)
THERAPY_DATE		DATE
DRUG_ID	NOT NULL	NUMBER(10)
NUMBER_OF_REFILLS		NUMBER(3)
DRUG_QUANTITY		NUMBER(3)

Appendix 2: SAS Program 'autoview.sas' for Generation of Access Views

```

****MODULE ONE: SAS VARIABLE NAMES****;
DATA VARIABLE;
INFILE '/fs11.1/xinshl/sql/paper/variable.lst' MISSOVER;
INPUT @2  ONAME $30.
      @43 DATATYPE :$30.;
IF SUBSTR(ONAME,1,4)= '----' THEN TABLENUM+1;
SASVAR=SUBSTR(COMPRESS(ONAME,'_'),1,8);
IF SUBSTR(ONAME,1,4) NOT IN('NAME','----',' ') THEN OUTPUT;
RUN;
PROC SORT;
BY TABLENUM SASVAR;
RUN;
DATA VARIABLE;

```

```

SET VARIABLE;
BY TABLENUM SASVAR;
IF FIRST.SASVAR THEN ID=0;
IF FIRST.SASVAR AND LAST.SASVAR THEN RETURN;
ELSE DO;
    ID+1;
    SUBSTR(SASVAR,8,1)=LEFT(ID);
    SASVAR=COMPRESS(SASVAR);
END;
RUN;

****MODULE TWO: SAS MEMBER NAMES****;
DATA TABLE;
INFILE '/fs11.1/xinshl/sql/paper/table.lst' MISSOEVER;
INPUT @6 OTABLE :$30.;
IF OTABLE NE ' ' THEN DO;
OTABLE=COMPRESS(OTABLE,','');
TABLENUM+1;
SASFILE=SUBSTR((COMPRESS(OTABLE,'_')),1,8);
OUTPUT;
END;
RUN;
PROC SORT;
BY SASFILE TABLENUM;
RUN;
DATA TABLE;
SET TABLE;
BY SASFILE;
IF FIRST.SASFILE THEN ID=0;
IF FIRST.SASFILE AND LAST.SASFILE THEN RETURN;
ELSE DO;
    ID+1;
    SUBSTR(SASFILE,8,1)=LEFT(ID);
    SASFILE=COMPRESS(SASFILE);
END;
RUN;

****MODULE THREE: SAS CODE GENERATION****;
DATA Q;
MERGE TABLE VARIABLE;
BY TABLENUM;
KEEP TABLENUM OTABLE SASVAR SASFILE ONAME DATATYPE;
RUN;
DATA Q;
LENGTH FMT $ 8;
SET Q;
FILE '/fs11.1/xinshl/sascode.sas';
BY TABLENUM;
IF _N_=1 THEN
    PUT @1 "LIBNAME ADLIB ' /users/ddd05_1/xinshl/ad/home lib'; RUN;" /
    @1 "LIBNAME VLIB ' /home/users/ddd05_1/xinshl/vlib'; RUN;" /
    @1 "%LET MPATH=PATH="@DEVL";" / ;
IF FIRST.TABLENUM THEN DO;
    PUT / @1 '***** TABLE: ' TABLENUM ' *****;' /
    @1 'PROC ACCESS DBMS=ORACLE;' /
    @8 'CREATE ADLIB.' SASFILE +(-1) '.ACCESS;' /
    @8 'USER=XINSHL;' /
    @8 'ORAPW=XINSHL1;' /
    @8 '&MPATH;' /
    @8 'TABLE=' @14 OTABLE ';' /
    @8 'ASSIGN=YES;' ;
END;
IF INDEX(DATATYPE,'DATE')>0 THEN FMT='DATE9.';
ELSE IF INDEX(DATATYPE,'NUMBER')>0 AND INDEX(DATATYPE,',')=0 THEN DO;
    FMT=SUBSTR(DATATYPE,INDEX(DATATYPE,'')+1);

```

```

FMT=TRIM(TRANSLATE(FMT,'.' , ')') || '|' ;
END;
ELSE IF INDEX(DATATYPE,'NUMBER')>0 AND INDEX(DATATYPE,',')>0 THEN DO;
FMT=SUBSTR(DATATYPE,INDEX(DATATYPE,'(')+1);
FMT=TRANSLATE(FMT,'.' , ')');
END;
ELSE DO;
FMT=SUBSTR(DATATYPE, INDEX(DATATYPE, '('));
FMT=TRANSLATE(FMT,'$.','('));
END;
PUT @1 'RENAME' @8 ONAME '=' SASVAR ';';
IF SUBSTR(FMT,1,1) NE '$' THEN PUT @1 'FORMAT' @8 ONAME FMT;
IF LAST.TABLENUM THEN
PUT @8 'LIST ALL;' //
@8 'CREATE VLIB.' SASVAR +(-1) '.VIEW;' /
@8 'SELECT ALL;' /
@8 'LIST VIEW;' /
@1 'RUN;';
FMT=COMPRESS(FMT,',');
RUN;

*****MODULE FOUR: PRITING DOCUMENTATIONS****;
OPTIONS LS=95 PS=75 NOCENTER PAGENO=1 NODATE;
PROC SORT DATA=Q;
BY OTABLE;
RUN;
PROC PRINT DATA=Q SPLIT='*' NOOBS UNIFORM;
VAR ONAME DATATYPE SASVAR FMT;
BY OTABLE SASFILE;
LABEL
ONAME = 'COLUMN*NAME'
OTABLE = 'ORACLE TABLE NAME'
DATATYPE = 'DATA*TYPE'
SASFILE = ' SAS MEMBER NAME'
SASVAR = 'VARIABLE*NAME'
FMT = 'SAS*DATA*FORMAT';
TITLE1 'SAS ACCESS VIEWS';
TITLE2 "REPORT DATE: &SYSDATE";
RUN;

```

Appendix 3: Sample Listing of 'sascode.sas', Result of Running 'autoview.sas'

```

LIBNAME ADLIB '/home/users/ddd05_1/xinshl/adlib'; RUN;
LIBNAME VLIB '/home/users/ddd05_1/xinshl/vlib'; RUN;
%LET MPATH=PATH="@DEVL";

***** TABLE: 1 *****;
PROC ACCESS DBMS=ORACLE;
CREATE ADLIB.DIAGNOSI.ACCESS;
USER=XINSHL;
ORAPW=XINSHL1;
&MPATH;
TABLE=DIAGNOSIS ;
ASSIGN=YES;
RENAME DIAGNOSIS_DATE =DIAGNOSI ;
FORMAT DIAGNOSIS_DATE DATE9.;
RENAME DOCTOR_ID =DOCTORID ;
FORMAT DOCTOR_ID 4.;
RENAME ICD9_CODE =ICD9CODE ;
RENAME PATIENT_ID =PATIENTI ;
FORMAT PATIENT_ID 5.;
LIST ALL;

```

```

CREATE VLIB.PATIENTI.VIEW;
SELECT ALL;
LIST VIEW;
RUN;

***** TABLE: 2 *****;
PROC ACCESS DBMS=ORACLE;
CREATE ADLIB.DOCTOR.ACCESS;
USER=XINSHL;
ORAPW=XINSHL1;
&MPATH;
TABLE=DOCTOR ;
ASSIGN=YES;
RENAME DOC_BIRTH_DATE =DOCBIRTH ;
FORMAT DOC_BIRTH_DATE DATE9.;
RENAME DOC_GENDER =DOCGENDE ;
RENAME DOCTOR_ID =DOCTORID ;
FORMAT DOCTOR_ID 4.;
RENAME SPECIALTY_CODE =SPECIALT ;
LIST ALL;

CREATE VLIB.SPECIALT.VIEW;
SELECT ALL;
LIST VIEW;
RUN;

```

Appendix 4: Sample Listing of 'autoview.lst', Result of Running 'autoview.sas'

```

SAS ACCESS VIEWS
1
REPORT DATE: 28AUG96

ORACLE TABLE NAME=DIAGNOSIS      SAS MEMBER NAME=DIAGNOSI

COLUMN          DATA          VARIABLE      SAS
NAME            TYPE            NAME          DATA
              FORMAT

DIAGNOSIS_DATE  DATE            DIAGNOSI     DATE9.
DOCTOR_ID       NUMBER(4)       DOCTORID     4.
ICD9_CODE       VARCHAR2(5)     ICD9CODE     $5.
PATIENT_ID      NUMBER(5)       PATIENTI     5.

ORACLE TABLE NAME=DOCTOR        SAS MEMBER NAME=DOCTOR

COLUMN          DATA          VARIABLE      SAS
NAME            TYPE            NAME          DATA
              FORMAT

DOC_BIRTH_DATE  DATE            DOCBIRTH     DATE9.
DOC_GENDER      VARCHAR2(1)     DOCGENDE     $1.
DOCTOR_ID       NUMBER(4)       DOCTORID     4.
SPECIALTY_CODE  VARCHAR2(3)     SPECIALT     $3.

```