Romancing Your Data:
The Getting-to-Know-You Phase

Carole Jesse

Sr SAS Analyst

PRIME THERAPEUTICS®

Eagan, MN

Overview

Motivation

Oracle® Database Architecture and Data Dictionary

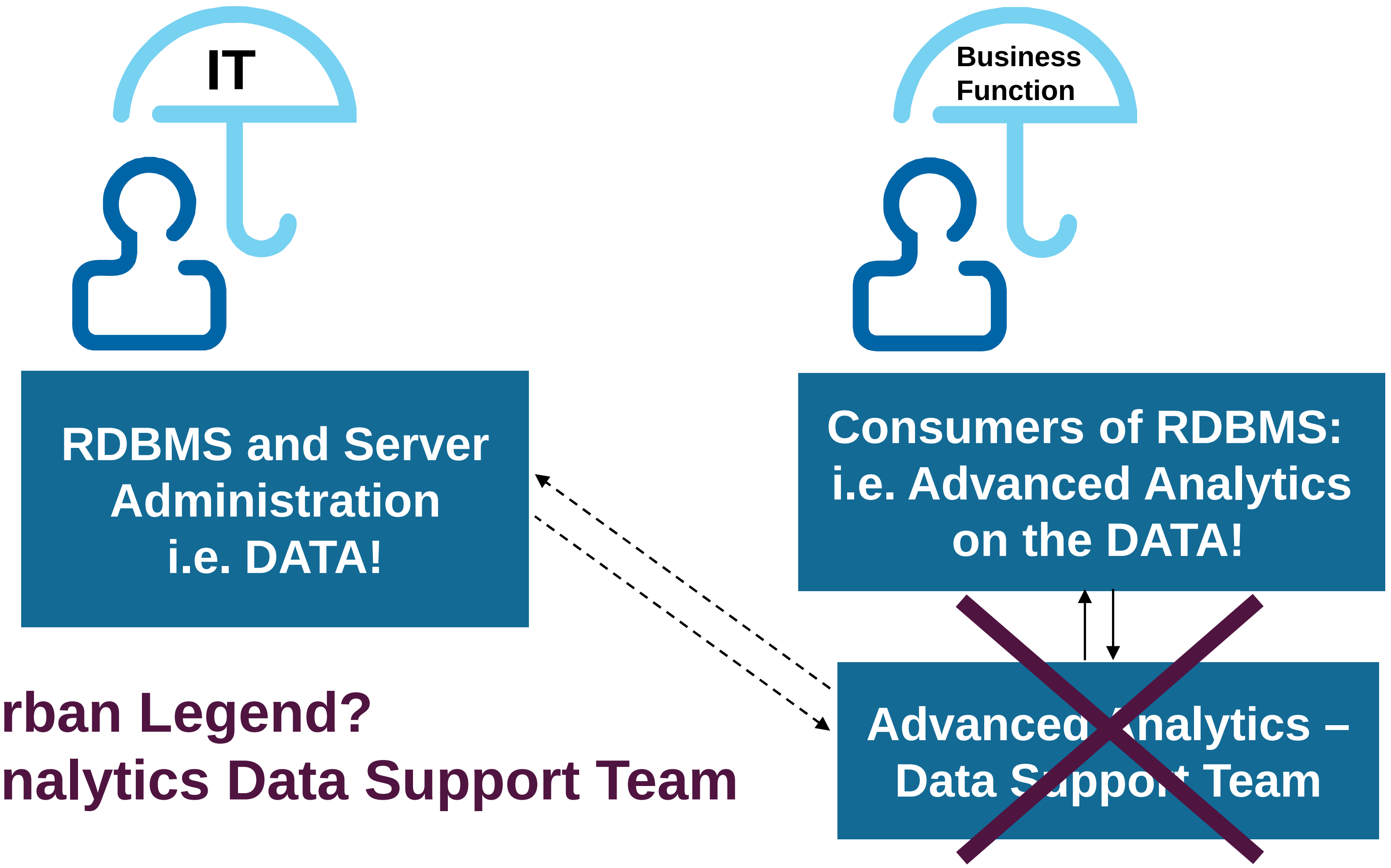
Review of SAS/ACCESS® Interface - Query Types

5 Base SAS® Scripts

- `1_Code_SysAllViews3Fam.sas`
- `4_Code_SysAllIndColumns.sas`

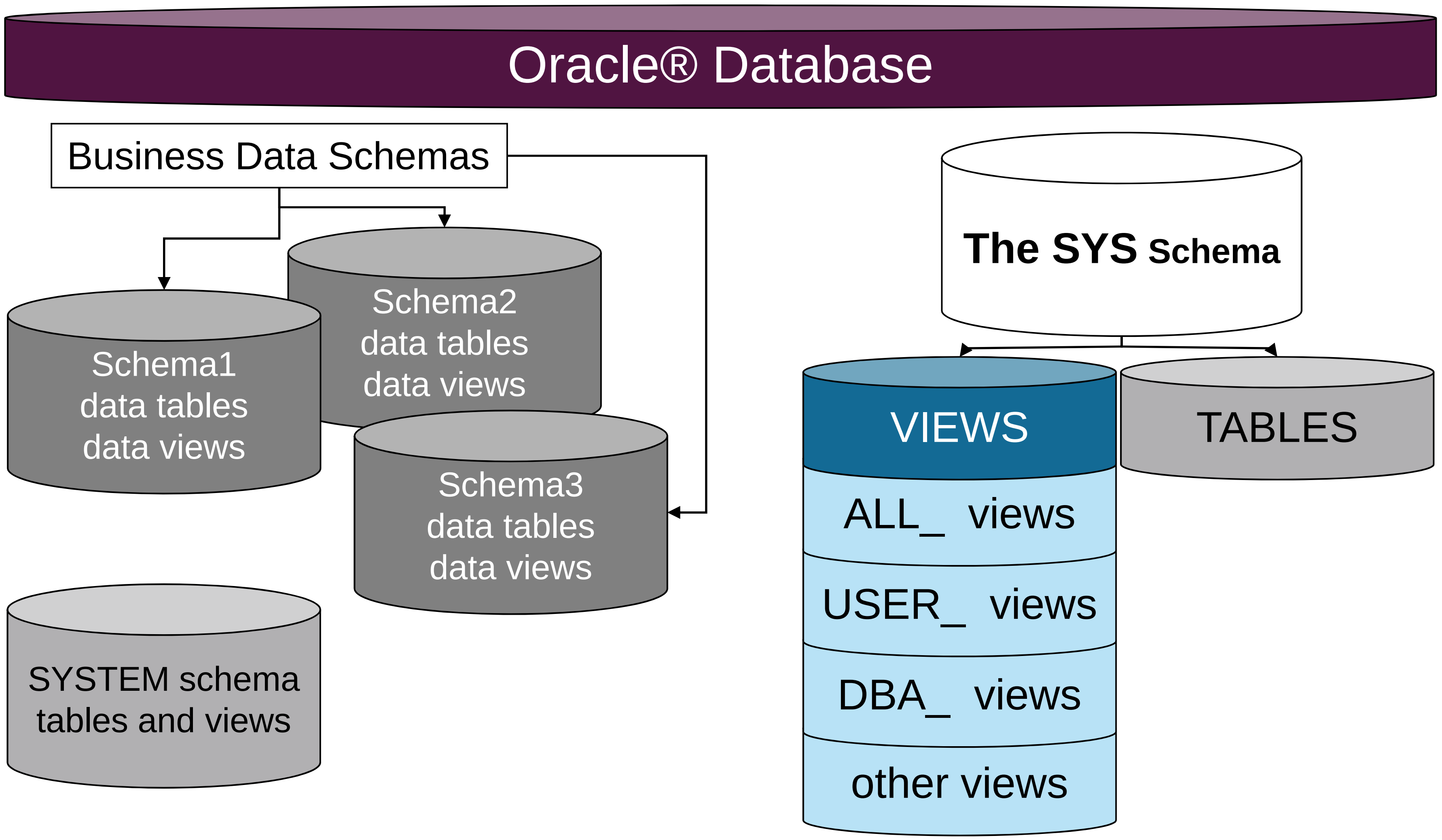
Conclusions & Questions

Motivation


IT
Business Function
RDBMS and Server Administration i.e. DATA!
Consumers of RDBMS: i.e. Advanced Analytics on the DATA!
Advanced Analytics – Data Support Team

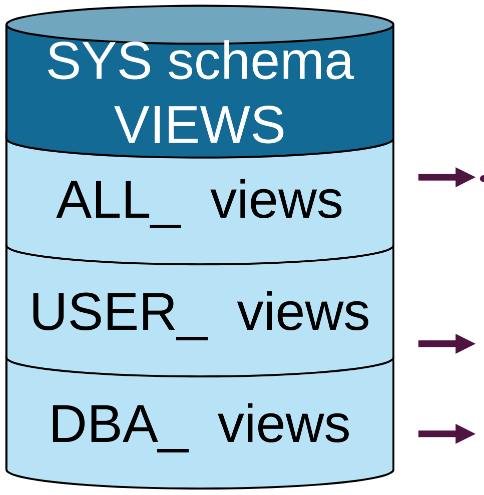
Urban Legend?
Analytics Data Support Team

Oracle® Database Architecture


Oracle® Database
Business Data Schemas
Schema1 data tables data views
Schema2 data tables data views
Schema3 data tables data views
SYSTEM schema tables and views
The SYS Schema
VIEWS
TABLES
ALL_ views
USER_ views
DBA_ views
other views

Data Dictionary: Where and Why

5 Base SAS® Scripts that work with six of the SYS Views in the Oracle Data Dictionary:



1: SYS.all_views
2: SYS.all_tables
4: SYS.all_ind_columns
5: SYS.all_constraints, SYS.all_cons_columns
3: SYS.user_role_privs
3: SYS.dba_role_privs

SAS/ACCESS® Interface, Query Types: LIBNAME

Generic Syntax of the LIBNAME statement to an Oracle schema:

LIBNAME *libref* **oracle USER=**’*ORACLE-user-name*’ **PASSWORD=**’*ORACLE-password*’ **PATH=**”*ORACLE-database-specification*” **SCHEMA=***schema-name*;

```
/* LIBNAME query in SAS/ACCESS Interface */
LIBNAME  db1_sch1  oracle  USER='cjesse'
PASSWORD='MyPW'  PATH="db1.server1.com"  SCHEMA=sch1;

PROC SQL;
 create table WORK.TBL1 as
 select col1, col2, col3
 from db1_sch1.TBL1;
QUIT;

DATA WORK.TBL1;
  set db1_sch1.TBL1;
  keep col1 col2 col3;
RUN;
```

SAS/ACCESS® Interface, Query Types: Pass-through Facility

Generic Syntax of the CONNECT statement to an Oracle database within PROC SQL:

```
CONNECT TO oracle <AS Alias>
        (USER='ORACLE-user-name'  PASSWORD='ORACLE-password'
         PATH="ORACLE-database-specification");
```

```
/*Pass-Through Facility query in SAS/ACCESS Interface*/

PROC SQL;
CONNECT TO oracle AS db1 (USER='cjesse' PASSWORD='MyPW'
PATH="db1.server1.com");
  CREATE table WORK.TBL1 as
  SELECT *                                 SAS SQL, SELECT
  FROM connection to db1                   connection based on CONNECT statement
   (SELECT col1, col2, col3 from sch1.tbl1)
  ;                                        Oracle pass-through SQL, SELECT
  DISCONNECT FROM db1;
QUIT;
```

Typical Layout of   #_Code.sas

```
ODS HTML body="&unixpath.<filename>.html";
  Title1 "<Text>"; Title2 "<Text>";
    PROC SQL;
    CONNECT to oracle as &ODBshrt.
           ( path="&ODBlong" &ODBcred. );
    SELECT
    < SAS SQL >
    FROM connection to &ODBshrt.
      (
       SELECT
       < Oracle pass-through SQL >
      )
    < SAS SQL > ;
    DISCONNECT FROM &ODBshrt.;
    QUIT;
  Title1; Title2;
ODS HTML close;
```

$\Big\}$ The Meat of the Query: the PROC SQL select

PROC SQL select: 1_Code_SysAllViews3Fam.sas

```
SELECT
SCANQ(VIEW_NAME,1,"_") as  FAMILY,
*
FROM connection to &ODBshrt.
   (
    SELECT
    VIEW_NAME
    FROM SYS.all_views
    WHERE OWNER='SYS'
   )
WHERE SCANQ(VIEW_NAME,1,"_")
      in ('ALL','USER','DBA')
ORDER FAMILY, VIEW_NAME
;
```

SAS SQL

Oracle pass-through SQL (native SQL)

SAS SQL

SYS.all_views
1_Code_SysAllViews3Fam.sas, Part 1 Results

SYS.all_views: 11 columns

View CONTAINS:
Information on all the Views in the database, including those in SYS, as well as the Business data schemas.

Most important columns:
- OWNER (schema name)
- VIEW_NAME

Breakdown of SYS.ALL_VIEWS in ALL_, USER_, DBA_
For ORACLE database database1.server2.com

FAMILY	VIEW_NAME
ALL	ALL_ALL_TABLES
ALL	ALL_APPLY
ALL	ALL_APPLY_CONFLICT_COLUMNS
. . .	. . .
ALL	ALL_WARNING_SETTINGS
DBA	DBA_AUTO_SEGADV_CTL
DBA	DBA_AUTO_SEGADV_SUMMARY
DBA	DBA_DATA_FILES
. . .	. . .
DBA	DBA_TABLESPACES
USER	USER_ADVISOR_ACTIONS
USER	USER_ADVISOR_DIRECTIVES
USER	USER_ADVISOR_FINDINGS
. . .	. . .
USER	USER_WARNING_SETTINGS

PROC SQL select: 4_Code_SysAllIndColumns.sas

```
SELECT
*
FROM connection to &ODBshrt.
   (
    SELECT
    INDEX_NAME,
    COLUMN_POSITION,
    COLUMN_NAME
    FROM SYS.all_ind_columns
    WHERE TABLE_OWNER=&OWNlong. and
          TABLE_NAME=&TBL.
    ORDER by INDEX_NAME,
             COLUMN_POSITION
   )
;
```

SAS SQL

Oracle pass-through SQL
(native SQL)

SYS.all_ind_columns
4_Code_SysAllIndColumns.sas, Results for TABLE11

SYS.all_ind_columns: 9 columns

View CONTAINS:
Information related to how Tables are indexed.

Most important columns:

- INDEX_NAME
- COLUMN_POSITION
- COLUMN_NAME

Indexes on Schema 'OWNER5', Table 'TABLE11' in Database: database1.server2.com

INDEX_NAME	COLUMN_POSITION	COLUMN_NAME
TABLE11_PK	1	FIPS_STATE_CODE
TABLE11_PK	2	FIPS_COUNTY_CODE

SYS.all_ind_columns
4_Code_SysAllIndColumns.sas, Results for TABLE236

Indexes on Schema 'OWNER5', Table 'TABLE236' in Database: database1.server2.com

INDEX_NAME	COLUMN_POSITION	COLUMN_NAME
NI1_TABLE236	1	BATCH_DATE
NI1_TABLE236	2	ACCOUNT_NUMBER
NI1_TABLE236	3	BAL_PRIN
NI1_TABLE236	4	LATE_FEE_UNCOLL
NI2_TABLE236	1	BATCH_DATE
NI2_TABLE236	2	ACCOUNT_NUMBER
NI2_TABLE236	3	STRAT_COLLECTIONS
NI2_TABLE236	4	LOAN_STATUS_CODE
NI2_TABLE236	5	COLLECTION_RESPONSE_CODE
NI3_TABLE236	1	BATCH_DATE
NI3_TABLE236	2	ACCOUNT_NUMBER
NI3_TABLE236	3	LOAN_STATUS_CODE
NI3_TABLE236	4	CREDIT_MAX
NI3_TABLE236	5	BLOCK_NUMBER
NI3_TABLE236	6	INVESTOR_NUMBER

Example 1: In Practice

Pass-through SQL to TABLE236 BEFORE SYS.all_ind_columns

```
(
 SELECT ACCOUNT_NUMBER,
        BAL_PRIN
 FROM OWNER5.TABLE236
 WHERE
 580 < FICO_SCORE_CURR
 and
 FICO_SCORE_CURR < 620
 and
 BATCH_DATE = '28-FEB-2010'
)
```

Pass-through SQL to TABLE236 AFTER SYS.all_ind_columns

```
(
 SELECT ACCOUNT_NUMBER,
        BAL_PRIN
 FROM OWNER5.TABLE236
 WHERE
 BATCH_DATE = '28-FEB-2010'
 and
 580 < FICO_SCORE_CURR
 and
 FICO_SCORE_CURR < 620
)
```

Using the knowledge about table indexing (BATCH_DATE) for the variable order in 'where' logic on the Oracle side yields a 21-31% reduction in run time!

Example 2: In Practice, Gnarley LIBNAME SQL

```
LIBNAME clmcom oracle DSN=ENT_PROD SCHEMA=STGINT READBUFF=200 &db_cred.;

PROC sql;
create table procsqltest2 as
SELECT *,
FROM clmcom.CLM_STG

WHERE
  CLM_FILL_DT_YR='2010' and
  DW_FINAL_CLM_STAT_CD='P' and
  CARR_ID in (&carrid.) and
  CARR_ID not in
  ('CARVE','HI8002','HI8032','IL8052','IL8054') and
  ACCT_ID not in
  ('5MXDISC','5DISCMX','BLUMNRC','BLUMNRG','BLUMNRP') and
  trim(CARR_ID)||trim(substr(ACCT_ID,1,3)) ne 'PGIGNCRV' and
  substr(GRP_PLAN_CD,1,4) ne 'MNF6' and
  trim(substr(CARR_ID,1,2))||substr(GRP_PLAN_CD,3,2) ne 'HMD0' and
  substr(GRP_PLAN_CD,1,6) ne 'NE5100' and
  substr(GRP_PLAN_CD,1,7) ne 'NEC5100'

ORDER BY CARR_ID, ACCT_ID, GRP_ID
;
QUIT;
```

2.5 - 3 hours to return 600k records from 410 million

Example 2: In Practice, Gnarley LIBNAME SQL

```
LIBNAME clmcom oracle DSN=ENT_PROD SCHEMA=STGINT READBUFF=200 &db_cred.;

PROC sql;
create table procsqltest2 as
SELECT *,
FROM clmcom.CLM_STG

WHERE
  CLM_FILL_DT_YR='2010' and
  DW_FINAL_CLM_STAT_CD='P' and
  CARR_ID in (&carrid.) and
  CARR_ID not in
  ('CARVE','HI8002','HI8032','IL8052','IL8054') and
  ACCT_ID not in
  ('5MXDISC','5DISCMX','BLUMNRC','BLUMNRG','BLUMNRP') and
  trim(CARR_ID)||trim(substr(ACCT_ID,1,3)) ne 'PGIGNCRV' and
  substr(GRP_PLAN_CD,1,4) ne 'MNF6' and
  trim(substr(CARR_ID,1,2))||substr(GRP_PLAN_CD,3,2) ne 'HMD0' and
  substr(GRP_PLAN_CD,1,6) ne 'NE5100' and
  substr(GRP_PLAN_CD,1,7) ne 'NEC5100'

ORDER BY CARR_ID, ACCT_ID, GRP_ID
;
QUIT;
```

SYS.all_ind_columns helps ID indexes and non-indexes

Example 2: In Practice, Pass-through SQL

```
PROC sql;
CONNECT to oracle as ENT_PROD (dsn=ENT_PROD &db_cred.);
CREATE table procsqltest as
/* start of SAS select */
SELECT *,
FROM connection to ENT_PROD
/* pass-through logic to Oracle server */
  (SELECT *
    FROM STGINT.CLM_STG
          WHERE
       CLM_FILL_DT_YR='2010' and
       DW_FINAL_CLM_STAT_CD='P' and
       CARR_ID in (&carrid.) and
       CARR_ID not in
         ('CARVE','HI8002','HI8032','IL8052','IL8054') and
       ACCT_ID not in
         ('5MXDISC','5DISCMX','BLUMNRC','BLUMNRG','BLUMNRP')
  )
/* back to SAS */
WHERE
  CATS(CARR ID, substr(ACCT ID,1,3)) NE 'PGIGNCRV' and
  substr(GRP_PLAN_CD,1,4) NE 'MNF6' and
  CATS(substr(CARR_ID,1,2),substr(GRP_PLAN_CD,3,2)) NE 'HMD0' and
  substr(GRP_PLAN_CD,1,6) NE 'NE5100' and
  substr(GRP_PLAN_CD,1,7) NE 'NEC5100'
ORDER BY CARR_ID, ACCT_ID, GRP_ID
;
QUIT;
```

10-12 minutes to return 600k records from 410 million

Conclusions and Contact Information

Utilizing SAS/Access Interface products to explore the RDBMS Data Dictionary is the Getting-to-Know-You Phase of Romancing Your Data!

Carole Jesse

E-mail:	carole.jesse@primetherapeutics.com
LinkedIn:	http://www.LinkedIn.com/in/CaroleJesse
SASCommunity:	http://www.sascommunity.org/wiki/User:CaroleJesse
Twitter:	http://www.twitter.com/CaroleJesse