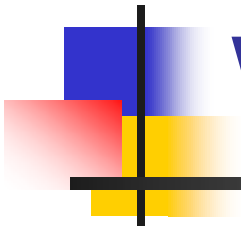


SAS vs DB2 Functionality – Who Does What Where ?!

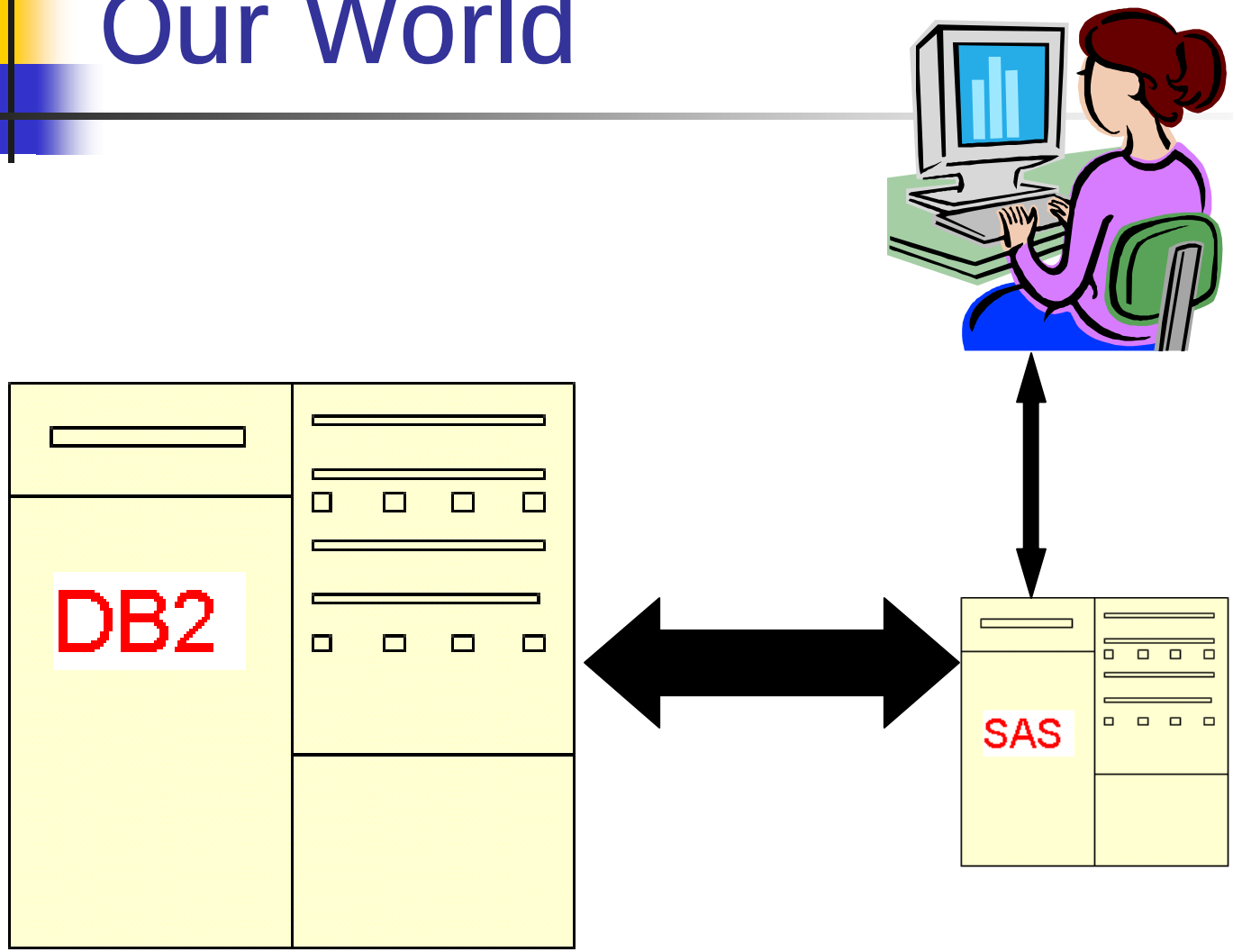
Harry Droogendyk
Stratia Consulting Inc.

BIG Mountain of Data

Move the Mountain ?

Move the Mountain !

Our World



Expensive Data Pulls

```
create table visa_bal as
     select * from connection to db2 (
       select acct_id, client_product_ds,
              current_balance_am
         from edw.visa_acct
        where effective_dt = '2009-10-31'
          and lifecycle_cd in ( 114,116,117 )
     );
```

- 5,000,000 rows come through the pipe to SAS

Expensive Data Pulls

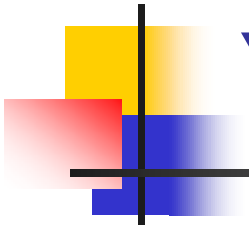
```
proc summary data = visa_bal;
  class client_product_ds;
  var current_balance_am;
  output out = visa_bal_sum sum=;
run;
```

- 18 rows in summary data set

Efficient Data Pulls

```
select      client_product_ds,
            sum(current_balance_am)  as
                 current_balance_am
  from      edw.visa_acct
 where      effective_dt = '2009-10-31'
   and      lifecycle_cd in ( 114,116,117 )
 group by client_product_ds
 order by client_product_ds
```

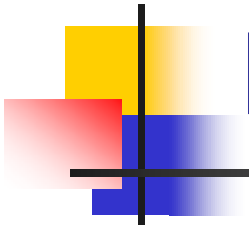
- let DB2 do the heavy lifting
  - query optimizer
- 18 rows through the pipe to SAS



Yo-Yo Programming

- up, down, up, down

- DB2 query into SAS
- SAS data step to massage contents
- Bump data back to DB2
- Use in DB2
- Back down to SAS....



No Mo' Yo –Yo Programming

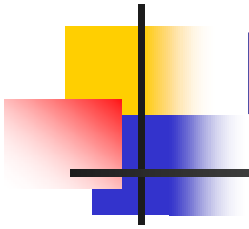
- use DB2 "WITH" to create temporary tables on the fly
  - exist only for duration of query
- allows division of tasks
- complex joins can be simplified
  - left, right, inner, full
  - how do I combine them ?!
- can lower query cost

No Mo' Yo –Yo Programming

```
create table visa_bal_sum as
  select * from connection to db2 (
      with intermediate_acct as (
     select a.acct_id, a.acct_type,
            b.current_balance_am
       from edw.acct      a    left join
            edw.acct_bal b
         on a.acct_id = b.acct_id        )

        select a.*, c.cust_id
         from intermediate_acct a …
```

SAS/Access - DB2

- allows direct access to back-end DB
- no DB-specific SQL required
- SAS/Access handles interface
- libname with DB2 engine
- two ways
  - explicitly coded
  - available through EG



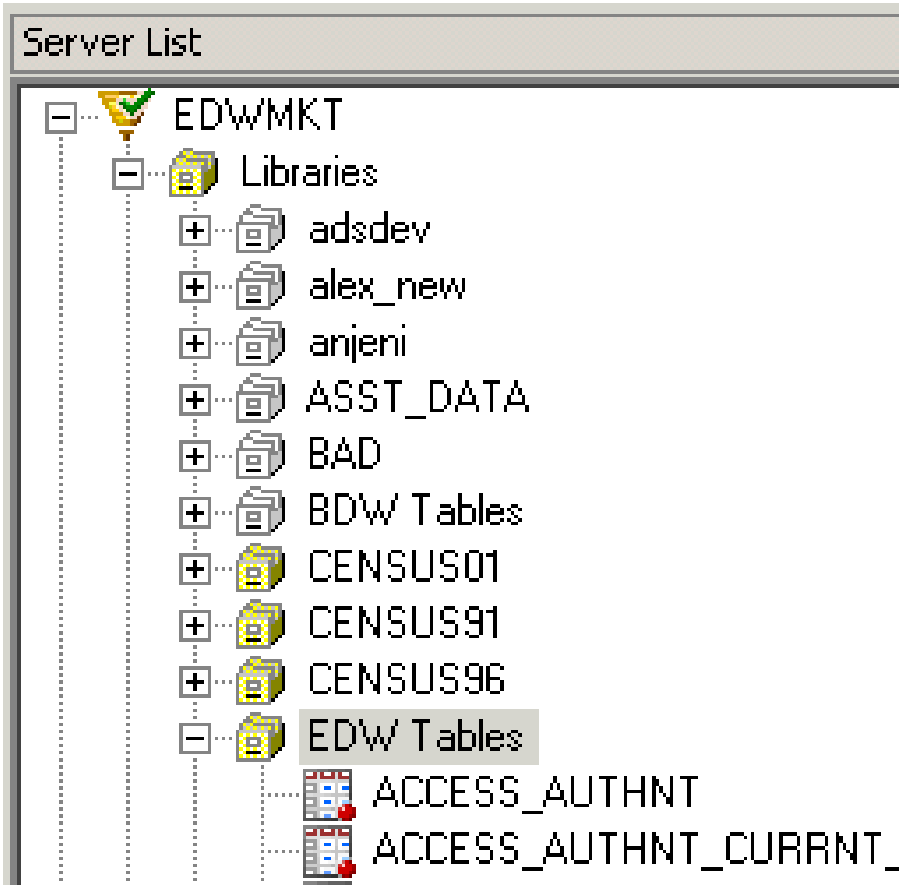
DB2 Library

- coded explicitly

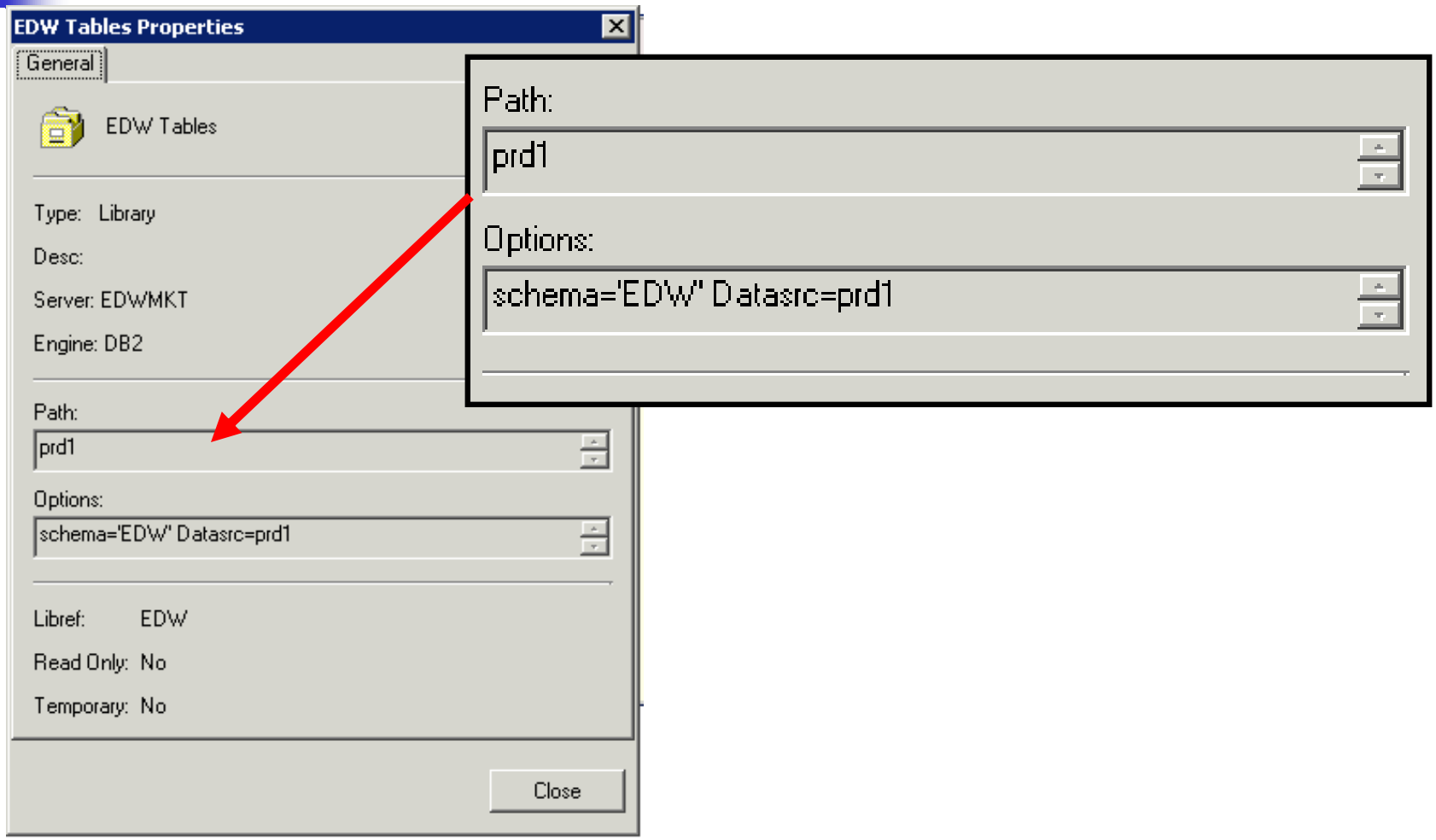
```
libname edw db2 database=prd1
                schema=edw;
proc print
     data = edw.ua_campaigns;
run;
```

DB2 Library

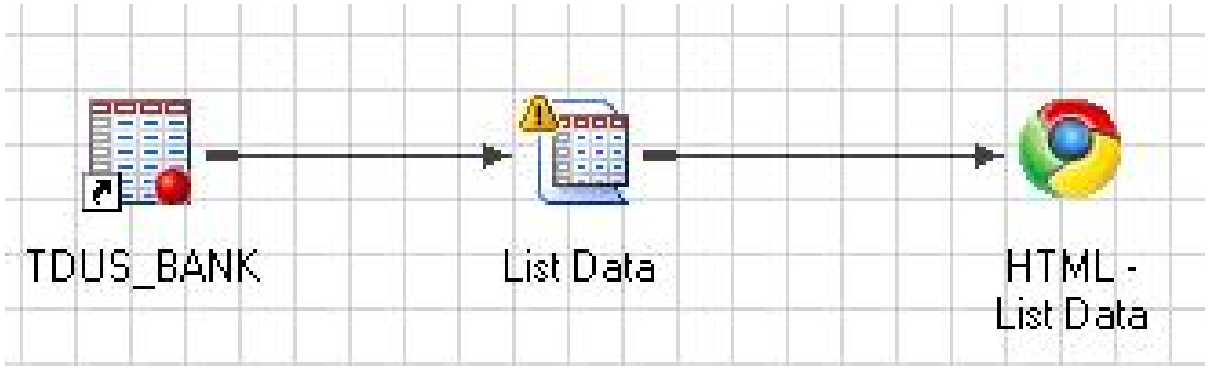
- EG

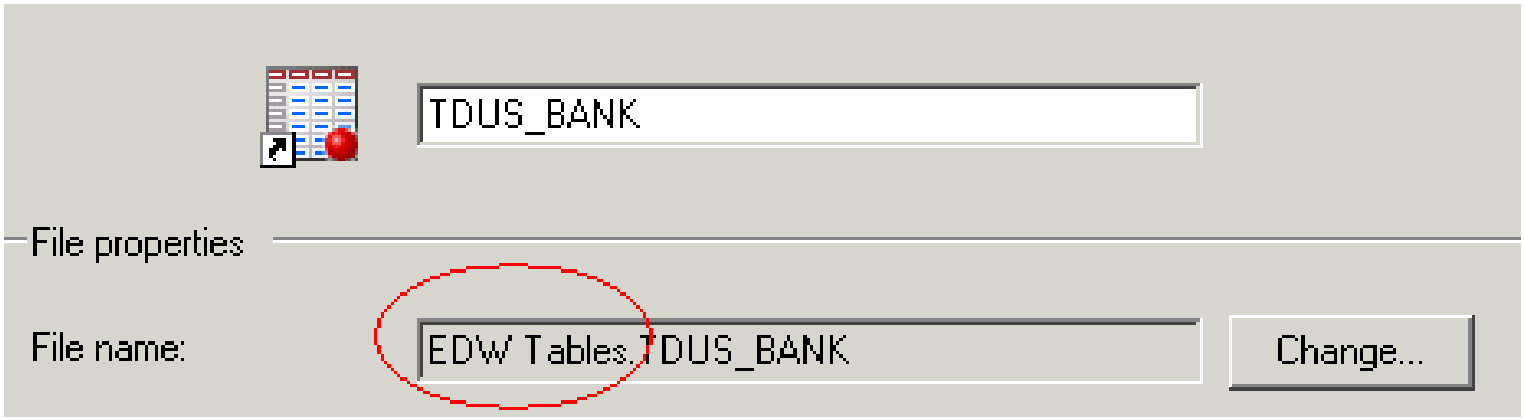


DB2 Library


EDW Tables Properties
General
EDW Tables
Type: Library
Desc:
Server: EDWMKT
Engine: DB2
Path:
prd1
Options:
schema='EDW' Datasrc=prd1
Libref: EDW
Read Only: No
Temporary: No
Close
Path:
prd1
Options:
schema='EDW' Datasrc=prd1

DB2 Library


TDUS_BANK
List Data
HTML -
List Data


TDUS_BANK
File properties
File name:
EDW Tables.TDUS_BANK
Change...

DB2 Troubles

- DB2 libref is h-a-n-d-y !
- no messy pass-thru syntax
- usable in familiar SAS PROC steps

```
proc print data = EDW.UA_CAMPAIGN;
  var campaign_id name desc
      campaigncode createdate;
   where datepart(createdate) >
         intnx('month',today(),-6,'b') ;
run;
```

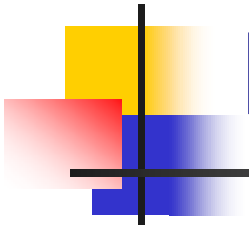
DB2 Troubles

- datepart & intnx SAS functions
- debugging option

```
options sastrace=',,,d' sastraceloc=saslog
                                nostsuffix;
```

- log snippet

```
SAS_SQL:  Unable to convert the query to a DBMS
   specific SQL statement due to an error.
ACCESS ENGINE:  SQL statement was not passed to
   the DBMS, SAS will do the processing.
```



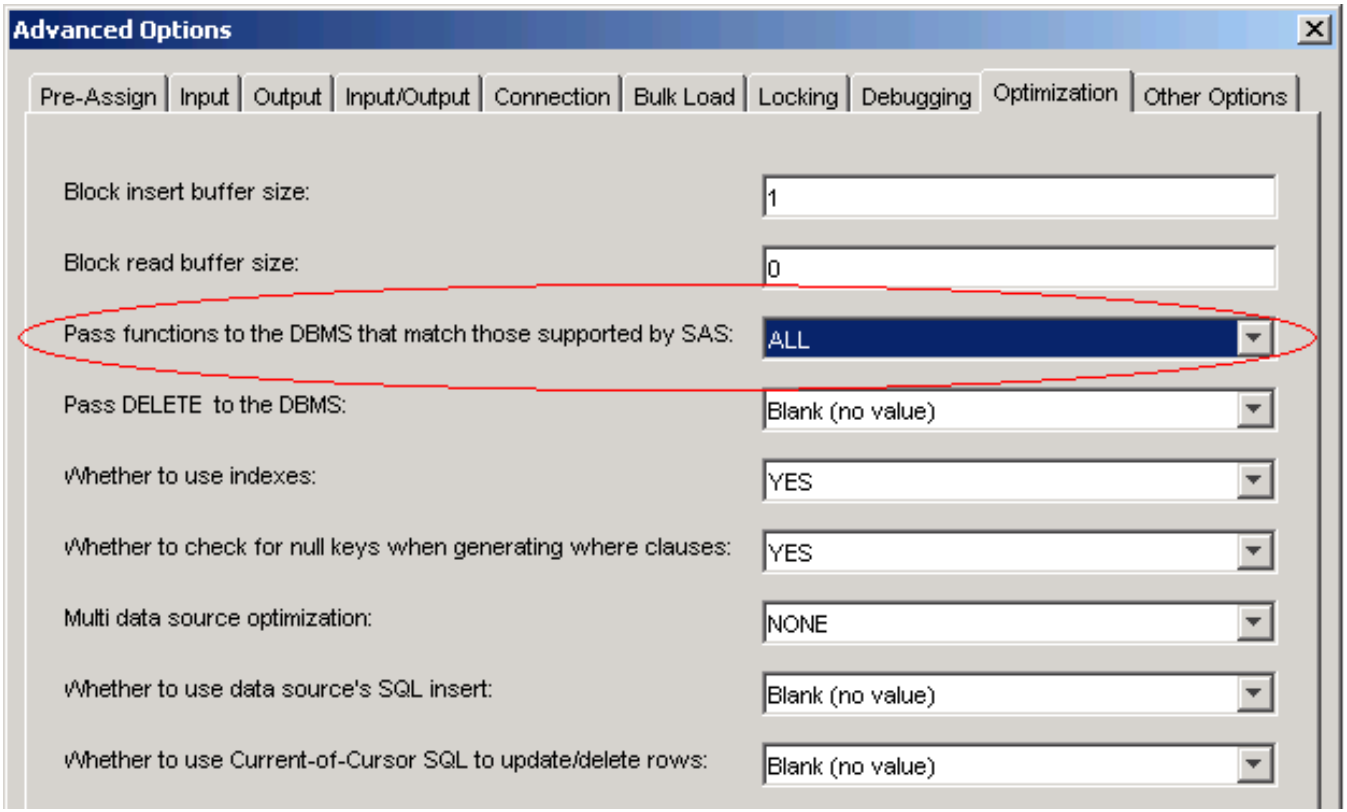
DB2 Untroubled

- DB2 libname option
  - sql_functions = all
- pass all *possible* functions through to DB

DATE TODAY QTR COMPRESS SUBSTR
DATEPART DAY SECOND INDEX TRANWRD
DATETIME HOUR WEEKDAY LENGTH TRIMN
TIME MINUTE YEAR REPEAT MOD
TIMEPART MONTH BYTE SOUNDEX

DB2 Untroubled

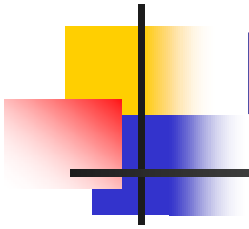
- EG and sql_functions = all



DB2 Troubles, kinda

- sometimes SAS is *really* smart
- WHERE clause:

```
where datepart(createdate) >
      intnx('month',today(),-6,'b')
                and
  scan(desc,1,' ') = 'Customers';
```



DB2 Troubles, kinda

- LOG snippet

```
NOTE: There were 12 observations read from the
  data set EDW.UA_CAMPAIGN.
WHERE SCAN(DESC, 1, ' ')='Customers';

NOTE: There were 2 observations read from the
  data set WORK.SORTTEMPTABLESORTED.
```

DB2 Library and KEEP

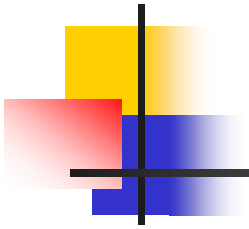
```
proc summary data = edw.visa_acct
  ( keep = client_product_ds
           current_balance_am );

  where    effective_dt = '31Oct2009'd
    and    lifecycle_cd in ( 114,116,117 );

  class client_product_ds;
  var current_balance_am;
  output out = visa_bal_sum sum=;
run;
```

SAS vs DB2 Functions

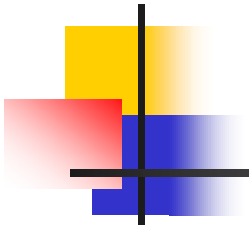
- INTNX
  - advance date intervals
- PUT / INPUT
  - convert data types
- NOTDIGIT
  - find non-numeric data

- SAS can’t convert – use pass-thru

INTNX in DB2



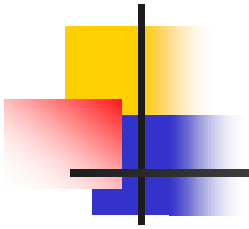
- rich date functions in SAS
- not as varied in DB2
- some things are easier in DB2
  - current date + 3 YEARS + 2 MONTHS + 15 DAYS
- mimic INTNX - use combinations of
  - DAY()
  - +/- *n* months



INTNX in DB2

- advance *n* months, same day

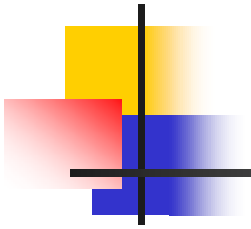
```
%macro intnx_month_same_day (d,sign,months);

      &d &sign &months month

%mend ;
```



INTNX in DB2

- advance *n* months, end of month

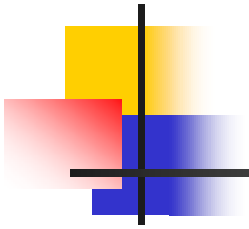
```
%macro intnx_month_end (d,sign,months);

  &d - (day (&d)) days &sign &months month

%mend ;
```



INTNX in DB2

- advance *n* months, beginning of month

```
%macro intnx_month_beginning (d,sign,months);

  &d - (day (&d)-1) days &sign &months month

%mend ;
```



INTNX in DB2

```
select * from connection to db2 (
      select test_dt
       , %intnx_month_beginning(test_dt,-,0)
                        as first_day_this_month
       , %intnx_month_beginning(test_dt,-,1)
                        as first_day_last_month
       , %intnx_month_beginning(test_dt,+,1)
                        as first_day_next_month
        from test_date
        );
```

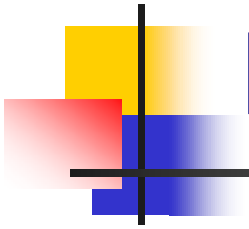
INTNX in DB2

```
select * from connection to db2 (
      select test_dt
      , %intnx_month_end(test_dt,+,0)
                as prev_last_day
      , %intnx_month_end(test_dt,+,3)
                as next3_last_day
      , %intnx_month_same_day(test_dt,-,3)
                as same_day_month3
        from test_date
        );
```

INTNX in DB2

TEST_DT	FIRST_DAY_ THIS_MONTH	FIRST_DAY_ LAST_MONTH	FIRST_DAY_ NEXT_MONTH
29FEB2008	01FEB2008	01JAN2008	01MAR2008
14SEP2009	01SEP2009	01AUG2009	01OCT2009
31AUG2009	01AUG2009	01JUL2009	01SEP2009

TEST_DT	PREV_LAST_ DAY	NEXT3_LAST_ DAY	SAME_DAY_ MONTH3
29FEB2008	31JAN2008	30APR2008	29NOV2007
14SEP2009	31AUG2009	30NOV2009	14JUN2009
31AUG2009	31JUL2009	31OCT2009	31MAY2009



DB2 Date Functions

- Bunch of DB2 date functions

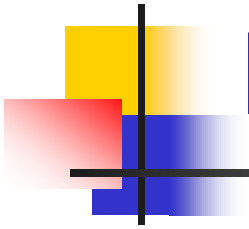
current date	- today
last_day	- last day of month
next_day	- first weekday
dayofweek	
etc...	

Non-Numeric & Type Changes

- credit bureau data
  - store flags and numerics in same field
  - how to parse in DB2?

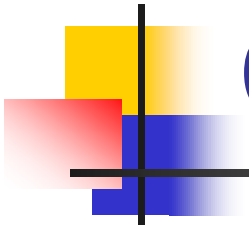
Resultset # 1 | Messages

	REGLAR_OPE	REGLAR_OI	REGLAR_OPE
1	I	I	I
2	03	0335	0002
3	W	W	W
4	02	0135	0000
5	W	W	W
6	I	I	I
7	W	W	W
8	01	0050	0000
9	W	W	W
10	I	I	I



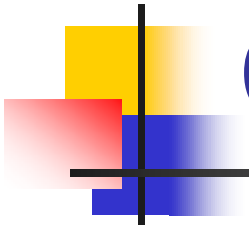
Non-Numeric & Type Changes

```
%macro numeric(f,mult);

  case when
    translate(&f,'','0123456789') = ''
      then cast(&f as integer) * &mult
      else 0
  end

%mend numeric;
```



Conclusion

- do stuff where it makes sense
  - summarize, subset, sort in DB2
- no more yo-yo
- not all SAS functions are portable into DB2
- sql_functions=all
- use pass-thru when it makes sense



Contact

Harry Droogendyk, SAS Consultant

harry@stratia.ca

Phone: 905-512-3827

Web: www.stratia.ca/papers