TERADATA AND SAS CODE BACKUP

```
   /** Most likely scenario based on provided data single digit month, day or
hour only **/
     /* 7 scenarios '8/1/2017 1:02:09 PM' '8/11/2017 1:02:09 PM' '8/1/2017
11:02:09 PM' '10/1/2017 1:02:09 PM' '8/11/2017 11:02:09 PM' '10/1/2017
11:02:09 PM' '10/14/2017 1:02:09 PM'
      */
    SELECT
    '10/12/2017 1:02:09 PM' AS YourDate,
    CAST(CASE
        WHEN YourDate LIKE '_/_/____
_:%'  THEN '0'||SUBSTRING(YourDate FROM 1 FOR 2)
|| '0' || SUBSTRING(YourDate FROM 3 FOR 2) || SUBSTRING(YourDateFROM 5 FOR 5)
|| '0' || SUBSTRING(YourDate FROM 10)
        WHEN YourDate LIKE '_/__/____
_:%' THEN '0'||SUBSTRING(YourDate FROM 1 FOR 2)
|| SUBSTRING(YourDate FROM 3 FOR 8) || '0' || SUBSTRING(YourDateFROM 11)
        WHEN YourDate LIKE '_/_/____
__:%' THEN '0' || SUBSTRING(YourDate FROM 1 FOR 2)
|| '0' || SUBSTRING(YourDate FROM 3)
        WHEN YourDate LIKE '_/__/____
__:%' THEN '0'||SUBSTRING(YourDate FROM 1 FOR 2)
|| SUBSTRING(YourDate FROM 3)
        WHEN YourDate LIKE '__/_/____
__:%' THEN SUBSTRING(YourDate FROM 1 FOR 3)
|| '0' || SUBSTRING(YourDate FROM 4)
        WHEN YourDate LIKE '__/__/____
_:%'THEN SUBSTRING(YourDate FROM 1 FOR 11)
|| '0' || SUBSTRING(YourDate FROM 12)
        WHEN YourDate LIKE '__/_/____
_:%' THEN SUBSTRING(YourDate FROM 1 FOR 3)
|| '0' || SUBSTRING(YourDate FROM 4 FOR 7)
|| '0' || SUBSTRING(YourDateFROM 11)

        ELSE YourDate
     END AS TIMESTAMP(0) FORMAT 'mm/dd/YYYYBhh:mi:ssBt');
```

```
CREATE VIEW HC_PRD_D_RDDL_INVITRO_0_1_0_0_0_0_0_0.AAA_D_CALIBRATION_DETAIL AS
    SELECT
    ID,
    ACTIONOPERATIONTASK_ID,
    NODE_ID,
    CREATED,
    DETAILS,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'SLOPE=\-
?\d+(\,?\d*)*\.?\d*',1,1),'SLOPE=','') AS FLOAT)  AS SLOPE,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'INTERCEPT=\-
?\d+(\,?\d*)*\.?\d*',1,1),'INTERCEPT=','') AS FLOAT)  AS INTERCEPT,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'CalibratorName=\w+',1,1),'CalibratorName=','') AS VARCHAR(50))  AS CALIBRATOR_NAME,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'CalibratorID=\w+',1,1),'CalibratorID=','') AS VARCHAR(50))  AS CALIBRATOR_ID,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'ReagentProductCode=\w+',1,1),'ReagentProductCode=','') AS VARCHAR(50)) AS REAGENT_PRODUCT_CODE,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'MethodName=\w+',1,1),'MethodName=','') AS VARCHAR(50))  AS METHOD_NAME,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-Concentration=\w+',1,1),'L1-Concentration=','') AS VARCHAR(50))  AS L1_Concentration,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-Sample_ID=\w+',1,1),'L1-Sample_ID=','') AS VARCHAR(50))  AS L1_Sample_ID,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-Result=\w+',1,1),'L1-Result=','') AS VARCHAR(50))  AS L1_Result,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-RLUMean=\w+',1,1),'L1-RLUMean=','') AS VARCHAR(50))  AS L1_RLUMean,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-RLUReplicate1=\w+',1,1),'L1-RLUReplicate1=','') AS VARCHAR(50))  AS L1_RLUReplicate1,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-RLUReplicate2=\w+',1,1),'L1-RLUReplicate2=','') AS VARCHAR(50))  AS L1_RLUReplicate2,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-RLUReplicate3=\w+',1,1),'L1-RLUReplicate3=','') AS VARCHAR(50))  AS L1_RLUReplicate3,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-Flags=\w+',1,1),'L1-Flags=','') AS VARCHAR(50))  AS L1_Flags,

    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-Concentration=\w+',1,1),'L2-Concentration=','') AS VARCHAR(50))  AS L2_Concentration,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-Sample_ID=\w+',1,1),'L2-Sample_ID=','') AS VARCHAR(50))  AS L2_Sample_ID,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-Result=\w+',1,1),'L2-Result=','') AS VARCHAR(50))  AS L2_Result,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-RLUMean=\w+',1,1),'L2-RLUMean=','') AS VARCHAR(50))  AS L2_RLUMean,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-RLUReplicate1=\w+',1,1),'L2-RLUReplicate1=','') AS VARCHAR(50))  AS L2_RLUReplicate1,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-RLUReplicate2=\w+',1,1),'L2-RLUReplicate2=','') AS VARCHAR(50))  AS L2_RLUReplicate2,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-RLUReplicate3=\w+',1,1),'L2-RLUReplicate3=','') AS VARCHAR(50))  AS L2_RLUReplicate3,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-Flags=\w+',1,1),'L2-Flags=','') AS VARCHAR(50))  AS L2_Flags,

    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Units=\w+',1,1),'Units=','') AS VARCHAR(50))  AS Units,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'CheckSum=\w+',1,1),'CheckSum=','') AS VARCHAR(50))  AS Check_Sum,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'RackID=\w+',1,1),'RackID=','') AS VARCHAR(50))  AS Rack_ID,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'OrderDateTime=\d+\/\d+\/\d+
\d+\:\d+\:\d+ \w+',1,1),'OrderDateTime=','') AS VARCHAR(50)
) ASORDERDATETIME_TXT,

   CAST(CASE
        WHEN ORDERDATETIME_TXT LIKE '_/_/____
_:%'  THEN '0'||SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 2)
|| '0' || SUBSTRING(ORDERDATETIME_TXT FROM 3 FOR2)
|| SUBSTRING(ORDERDATETIME_TXT FROM 5 FOR 5)
|| '0' || SUBSTRING(ORDERDATETIME_TXT FROM 10)
        WHEN ORDERDATETIME_TXT LIKE '_/__/____
_:%' THEN '0'||SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 2)
|| SUBSTRING(ORDERDATETIME_TXT FROM 3 FOR 8)
||'0' || SUBSTRING(ORDERDATETIME_TXT FROM 11)
        WHEN ORDERDATETIME_TXT LIKE '_/_/____
__:%' THEN '0' || SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 2)
|| '0' || SUBSTRING(ORDERDATETIME_TXT FROM 3)
        WHEN ORDERDATETIME_TXT LIKE '_/__/____
__:%' THEN '0'||SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 2)
|| SUBSTRING(ORDERDATETIME_TXT FROM 3)
        WHEN ORDERDATETIME_TXT LIKE '__/_/____
__:%' THEN SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 3)
|| '0' || SUBSTRING(ORDERDATETIME_TXT FROM 4)
        WHEN ORDERDATETIME_TXT LIKE '__/__/____
_:%'THEN SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 11)
|| '0' || SUBSTRING(ORDERDATETIME_TXT FROM 12)
        WHEN ORDERDATETIME_TXT LIKE '__/_/____
_:%' THEN SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 3)
|| '0' || SUBSTRING(ORDERDATETIME_TXT FROM 4 FOR 7)
||'0' || SUBSTRING(ORDERDATETIME_TXT FROM 11)

        ELSE ORDERDATETIME_TXT
     END AS TIMESTAMP(0) FORMAT 'mm/dd/YYYYBhh:mi:ssBt') AS ORDER_DATETIME,


    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'SampleStatus=\w+',1,1),'Sample
Status=','') AS VARCHAR(50))  AS Sample_Status,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'UserNameSystemName=\w+',1,1),'
UserNameSystemName=','') AS VARCHAR(50))  AS User_Name_System_Name,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'ReagentName=\w+',1,1),'Reagent
Name=','') AS VARCHAR(50))  AS Reagent_Name,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'ReagentPackID=\w+',1,1),'Reage
ntPackID=','') AS VARCHAR(50))  AS Reagent_Pack_ID,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'AspirationDate=\d+\/\d+\/\d+
\d+\:\d+\:\d+
\w+',1,1),'AspirationDate=','') AS VARCHAR(50)) ASASPIRATIONDATE_TXT,

       CAST(CASE
        WHEN ASPIRATIONDATE_TXT LIKE '_/_/____
_:%'  THEN '0'||SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 2)
|| '0' || SUBSTRING(ASPIRATIONDATE_TXT FROM 3FOR 2)
|| SUBSTRING(ASPIRATIONDATE_TXT FROM 5 FOR 5)
|| '0' || SUBSTRING(ASPIRATIONDATE_TXT FROM 10)
        WHEN ASPIRATIONDATE_TXT LIKE '_/__/____
_:%' THEN '0'||SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 2)
|| SUBSTRING(ASPIRATIONDATE_TXT FROM 3 FOR 8)
|| '0' || SUBSTRING(ASPIRATIONDATE_TXT FROM 11)
```

```
        WHEN ASPIRATIONDATE_TXT LIKE '_/_/____
__:%' THEN '0' || SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 2)
|| '0' || SUBSTRING(ASPIRATIONDATE_TXT FROM3)
        WHEN ASPIRATIONDATE_TXT LIKE '_/__/____
__:%' THEN '0'||SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 2)
|| SUBSTRING(ASPIRATIONDATE_TXT FROM 3)
        WHEN ASPIRATIONDATE_TXT LIKE '__/_/____
__:%' THEN SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 3)
|| '0' || SUBSTRING(ASPIRATIONDATE_TXT FROM 4)
        WHEN ASPIRATIONDATE_TXT LIKE '__/__/____
_:%'THEN SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 11)
|| '0' || SUBSTRING(ASPIRATIONDATE_TXT FROM 12)
        WHEN ASPIRATIONDATE_TXT LIKE '__/_/____
_:%' THEN SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 3)
|| '0' || SUBSTRING(ASPIRATIONDATE_TXT FROM 4 FOR 7)
|| '0' || SUBSTRING(ASPIRATIONDATE_TXT FROM 11)

        ELSE ASPIRATIONDATE_TXT
     END AS TIMESTAMP(0) FORMAT 'mm/dd/YYYYBhh:mi:ssBt') AS ASPIRATION_DATETI
ME,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'DBSynchID=\w+',1,1),'DBSynchID
=','') AS VARCHAR(50))  AS DBSynch_ID

    FROM "HC_PRD_D_ACLO_BAS_0_14_0_0_0_0_0_0"."D_CALIBRATION";
```

```
/*** first create detail table then create ext table ***/

CREATE VIEW HC_PRD_D_RDDL_INVITRO_0_1_0_0_0_0_0_0.AAA_D_CALIBRATION_DETAIL AS
    SELECT

    ACTIONOPERATIONTASK_ID,
    ASSAYLOT,
    ASSAYNAME,
    BAS_LOAD_DTTM,
    CALIBRATORLOT,
    ----CITY,
   ------ COUNTRY_NAME,
    CREATED,
    CREATED_DATE,
   ------ CUSTOMER_NAME,
    DATETIMEUTC,
    DETAILS,


    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'SLOPE=\-
?\d+(\,?\d*)*\.?\d*',1,1),'SLOPE=','') AS FLOAT)  AS SLOPE,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'INTERCEPT=\-
?\d+(\,?\d*)*\.?\d*',1,1),'INTERCEPT=','') AS FLOAT)  AS INTERCEPT,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'CalibratorName=\w+',1,1),'Cali
bratorName=','') AS VARCHAR(50))  AS CALIBRATORNAME,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'CalibratorID=\w+',1,1),'Calibr
atorID=','') AS VARCHAR(50))  AS CALIBRATORID,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'ReagentProductCode=\w+',1,1),'
ReagentProductCode=','') AS VARCHAR(50)) AS REAGENTPRODUCTCODE,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'MethodName=\w+',1,1),'MethodNa
me=','') AS VARCHAR(50))  AS METHODNAME,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-
Concentration=\w+',1,1),'L1-Concentration=','') AS VARCHAR(50))  AS
L1_CONCENTRATION,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-Sample_ID=\w+',1,1),'L1-
Sample_ID=','') AS VARCHAR(50))  AS L1_SAMPLE_ID,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-Result=\w+',1,1),'L1-
Result=','') AS VARCHAR(50))  AS L1_RESULT,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-RLUMean=\w+',1,1),'L1-
RLUMean=','') AS VARCHAR(50))  AS L1_RLUMEAN,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-
RLUReplicate1=\w+',1,1),'L1-RLUReplicate1=','') AS VARCHAR(50))  AS
L1_RLUREPLICATE1,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-
RLUReplicate2=\w+',1,1),'L1-RLUReplicate2=','') AS VARCHAR(50))  AS
L1_RLUREPLICATE2,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-
RLUReplicate3=\w+',1,1),'L1-RLUReplicate3=','') AS VARCHAR(50))  AS
L1_RLUREPLICATE3,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-Flags=\w+',1,1),'L1-
Flags=','') AS VARCHAR(50))  AS L1_FLAGS,

    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-
Concentration=\w+',1,1),'L2-Concentration=','') AS VARCHAR(50))  AS
L2_CONCENTRATION,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-Sample_ID=\w+',1,1),'L2-
Sample_ID=','') AS VARCHAR(50))  AS L2_SAMPLE_ID,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-Result=\w+',1,1),'L2-
Result=','') AS VARCHAR(50))  AS L2_RESULT,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-RLUMean=\w+',1,1),'L2-
RLUMean=','') AS VARCHAR(50))  AS L2_RLUMEAN,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-
RLUReplicate1=\w+',1,1),'L2-RLUReplicate1=','') AS VARCHAR(50))  AS
L2_RLUREPLICATE1,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-
RLUReplicate2=\w+',1,1),'L2-RLUReplicate2=','') AS VARCHAR(50))  AS
L2_RLUREPLICATE2,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-
RLUReplicate3=\w+',1,1),'L2-RLUReplicate3=','') AS VARCHAR(50))  AS
L2_RLUREPLICATE3,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-Flags=\w+',1,1),'L2-
Flags=','') AS VARCHAR(50))  AS L2_FLAGS,

    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Units=\w+',1,1),'Units=','')
AS VARCHAR(50))  AS UNITS,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'CheckSum=\w+',1,1),'CheckSum='
,'') AS VARCHAR(50))  AS CHECKSUM,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'RackID=\w+',1,1),'RackID=','')
AS VARCHAR(50))  AS RACKID,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'OrderDateTime=\d+\/\d+\/\d+
\d+\:\d+\:\d+ \w+',1,1),'OrderDateTime=','') AS VARCHAR(50) ) AS
ORDERDATETIME_TXT,

   CAST(CASE
        WHEN ORDERDATETIME_TXT LIKE '_/_/____ _:%'  THEN
'0'||SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 2) || '0' ||
SUBSTRING(ORDERDATETIME_TXT FROM 3 FOR 2) || SUBSTRING(ORDERDATETIME_TXT FROM
5 FOR 5) || '0' || SUBSTRING(ORDERDATETIME_TXT FROM 10)
```

```
        WHEN ORDERDATETIME_TXT LIKE '_/__/____ _:%' THEN
'0'||SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 2) || SUBSTRING(ORDERDATETIME_TXT
FROM 3 FOR 8) || '0' || SUBSTRING(ORDERDATETIME_TXT FROM 11)
        WHEN ORDERDATETIME_TXT LIKE '_/_/____ __:%' THEN '0' ||
SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 2) || '0' ||
SUBSTRING(ORDERDATETIME_TXT FROM 3)
        WHEN ORDERDATETIME_TXT LIKE '_/__/____ __:%' THEN
'0'||SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 2) || SUBSTRING(ORDERDATETIME_TXT
FROM 3)
        WHEN ORDERDATETIME_TXT LIKE '__/_/____ __:%' THEN
SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 3) || '0' ||
SUBSTRING(ORDERDATETIME_TXT FROM 4)
        WHEN ORDERDATETIME_TXT LIKE '__/__/____ _:%'THEN
SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 11) || '0' ||
SUBSTRING(ORDERDATETIME_TXT FROM 12)
        WHEN ORDERDATETIME_TXT LIKE '__/_/____ _:%' THEN
SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 3) || '0' ||
SUBSTRING(ORDERDATETIME_TXT FROM 4 FOR 7) || '0' ||
SUBSTRING(ORDERDATETIME_TXT FROM 11)

        ELSE ORDERDATETIME_TXT
     END AS TIMESTAMP(0) FORMAT 'mm/dd/YYYYBhh:mi:ssBt') AS ORDERDATETIME,


    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'SampleStatus=\w+',1,1),'Sample
Status=','') AS VARCHAR(50))  AS SAMPLESTATUS,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'UserNameSystemName=\w+',1,1),'
UserNameSystemName=','') AS VARCHAR(50))  AS USERNAMESYSTEMNAME,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'ReagentName=\w+',1,1),'Reagent
Name=','') AS VARCHAR(50))  AS REAGENTNAME,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'ReagentPackID=\w+',1,1),'Reage
ntPackID=','') AS VARCHAR(50))  AS REAGENTPACKID,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'AspirationDate=\d+\/\d+\/\d+
\d+\:\d+\:\d+ \w+',1,1),'AspirationDate=','') AS VARCHAR(50)) AS
ASPIRATIONDATE_TXT,

       CAST(CASE
        WHEN ASPIRATIONDATE_TXT LIKE '_/_/____ _:%'  THEN
'0'||SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 2) || '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 3 FOR 2) || SUBSTRING(ASPIRATIONDATE_TXT
FROM 5 FOR 5) || '0' || SUBSTRING(ASPIRATIONDATE_TXT FROM 10)
        WHEN ASPIRATIONDATE_TXT LIKE '_/__/____ _:%' THEN
'0'||SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 2) ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 3 FOR 8) || '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 11)
        WHEN ASPIRATIONDATE_TXT LIKE '_/_/____ __:%' THEN '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 2) || '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 3)
        WHEN ASPIRATIONDATE_TXT LIKE '_/__/____ __:%' THEN
'0'||SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 2) ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 3)
        WHEN ASPIRATIONDATE_TXT LIKE '__/_/____ __:%' THEN
SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 3) || '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 4)
        WHEN ASPIRATIONDATE_TXT LIKE '__/__/____ _:%'THEN
SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 11) || '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 12)
```

```
        WHEN ASPIRATIONDATE_TXT LIKE '__/_/____ _:%' THEN
SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 3) || '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 4 FOR 7) || '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 11)

        ELSE ASPIRATIONDATE_TXT
     END AS TIMESTAMP(0) FORMAT 'mm/dd/YYYYBhh:mi:ssBt') AS ASPIRATIONDATE,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'DBSynchID=\w+',1,1),'DBSynchID
=','') AS VARCHAR(50))  AS DBSYNCHID,

    EXPIRATIONDATETIMEUTC,
    ID,
  -----  MATERIAL_NUMBER,
    NODE_ID,
  -----  NODE_NAME,
   ----- PRODUCT_SECTION_NAME,
  -------- SERIAL_NUMBER,
    STATUS

    FROM "HC_PRD_D_ACLO_BAS_0_14_0_0_0_0_0_0"."D_CALIBRATION";


    CREATE VIEW
"HC_PRD_D_RDDL_INVITRO_0_1_0_0_0_0_0_0"."AAA_D_CALIBRATION_EXT" AS
    SELECT D_CALIBRATION."ACTIONOPERATIONTASK_ID",
        D_CALIBRATION."ASSAYLOT",
        D_CALIBRATION."ASSAYNAME",
        NODE."BAS_LOAD_DTTM",
        D_CALIBRATION."CALIBRATORLOT",
        NODE."CITY",
        COUNTRY."COUNTRY_NAME",
        D_CALIBRATION."CREATED",
        D_CALIBRATION."CREATED_DATE",
        NULL AS "CUSTOMER_NAME",
        COALESCE(CAST(SUBSTR(D_CALIBRATION."DATETIMEUTC",1,20) AS
TIMESTAMP(6) FORMAT 'YY-MM-DD:HH:MI:SS'), NULL) AS "DATETIMEUTC",
        D_CALIBRATION."DETAILS",
        COALESCE(CAST(SUBSTR(D_CALIBRATION."EXPIRATIONDATETIMEUTC",1,20) AS
TIMESTAMP(6) FORMAT 'YY-MM-DD:HH:MI:SS'), NULL) AS "EXPIRATIONDATETIMEUTC",
        D_CALIBRATION."ID",
        NODE."MATERIAL_NUMBER",
        NODE."NODE_ID",
        NODE."NODE_NAME",
        PRODUCT_SECTION."PRODUCT_SECTION_NAME",
        NODE."SERIAL_NUMBER",
        D_CALIBRATION."STATUS",
        D_CALIBRATION."SLOPE",
        D_CALIBRATION."INTERCEPT",
        D_CALIBRATION."CALIBRATORNAME",
        D_CALIBRATION."CALIBRATORID",
        D_CALIBRATION."REAGENTPRODUCTCODE",
        D_CALIBRATION."METHODNAME",
        D_CALIBRATION."L1_CONCENTRATION",
        D_CALIBRATION."L1_SAMPLE_ID",
        D_CALIBRATION."L1_RESULT",
        D_CALIBRATION."L1_RLUMEAN",
        D_CALIBRATION."L1_RLUREPLICATE2",
        D_CALIBRATION."L1_RLUREPLICATE3",
        D_CALIBRATION."L1_FLAGS",
        D_CALIBRATION."L2_CONCENTRATION",
        D_CALIBRATION."L2_SAMPLE_ID",
        D_CALIBRATION."L2_RESULT",
        D_CALIBRATION."L2_RLUMEAN",
        D_CALIBRATION."L2_RLUREPLICATE1",
        D_CALIBRATION."L2_RLUREPLICATE2",
        D_CALIBRATION."L2_RLUREPLICATE3",
        D_CALIBRATION."L2_FLAGS",
        D_CALIBRATION."UNITS",
        D_CALIBRATION."CHECKSUM",
        D_CALIBRATION."RACKID",
        D_CALIBRATION."ORDERDATETIME",
        D_CALIBRATION."SAMPLESTATUS",
        D_CALIBRATION."USERNAMESYSTEMNAME",
        D_CALIBRATION."REAGENTNAME",
        D_CALIBRATION."REAGENTPACKID",
        D_CALIBRATION."ASPIRATIONDATE",
        D_CALIBRATION."DBSYNCHID"
        FROM
"HC_PRD_D_RDDL_INVITRO_0_1_0_0_0_0_0_0"."AAA_D_CALIBRATION_DETAIL" AS
D_CALIBRATION LEFT JOIN "HC_PRD_D_ACLO_BAS_0_14_0_0_0_0_0_0"."NODE" AS NODE
ON ( D_CALIBRATION."NODE_ID" = NODE."NODE_ID" ) LEFT JOIN
"HC_PRD_D_ACLO_BAS_0_13_0_0_0_0_0_0"."PRODUCT_SECTION" AS PRODUCT_SECTION ON
( NODE."PRODUCT_SECTION_ID" = PRODUCT_SECTION."PRODUCT_SECTION_ID" ) LEFT
JOIN "HC_PRD_D_ACLO_BAS_0_14_0_0_0_0_0_0"."COUNTRY_NODE" AS COUNTRY ON (
NODE."COUNTRY_ID" = COUNTRY."COUNTRY_ID" AND NODE."NODE_ID" =
COUNTRY."NODE_ID" )
        WHERE EXTRACT(YEAR
        FROM D_CALIBRATION."CREATED") >= 2016
            AND  NODE."MATERIAL_NUMBER" IN
('11065004','11065006','11065464','11066000','11066001','11067000','11068008'
,'11069001','11069004','11069018','11069020');
```

```
     /******  DROP table
"HC_PRD_D_RDDL_INVITRO_0_1_0_0_0_0_0_0"."AAA_D_CALIBRATION_EXT_v2" ****/
   create table
"HC_PRD_D_RDDL_INVITRO_0_1_0_0_0_0_0_0"."AAA_D_CALIBRATION_EXT_v2" as
      (select
         D_CALIBRATION."ACTIONOPERATIONTASK_ID",
         D_CALIBRATION."ASSAYLOT",
         D_CALIBRATION."ASSAYNAME",
         NODE."BAS_LOAD_DTTM",
         D_CALIBRATION."CALIBRATORLOT",
         NODE."CITY",
         COUNTRY."COUNTRY_NAME",
         D_CALIBRATION."CREATED",
         D_CALIBRATION."CREATED_DATE",
         null as "CUSTOMER_NAME",
         coalesce(CAST(substr(D_CALIBRATION."DATETIMEUTC",1,20) AS
TIMESTAMP(6) FORMAT 'YY-MM-DD:HH:MI:SS'), null)  as "DATETIMEUTC",
         D_CALIBRATION."DETAILS",
         coalesce(CAST(substr(D_CALIBRATION."EXPIRATIONDATETIMEUTC",1,20) AS
TIMESTAMP(6) FORMAT 'YY-MM-DD:HH:MI:SS'), null) as "EXPIRATIONDATETIMEUTC",
         D_CALIBRATION."ID",
         NODE."MATERIAL_NUMBER",
         NODE."NODE_ID",
         NODE."NODE_NAME",
         PRODUCT_SECTION."PRODUCT_SECTION_NAME",
         NODE."SERIAL_NUMBER",
         D_CALIBRATION."STATUS",

    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'SLOPE=\s?\-
?\d+\.?\d*',1,1),'SLOPE=','') AS FLOAT) AS SLOPE,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'INTERCEPT=\s?\-
?\d+\.?\d*',1,1),'INTERCEPT=','') AS FLOAT) AS INTERCEPT,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Correlation Coeff=\s?\-
?\d+\.?\d*',1,1),'Correlation Coeff=','') AS FLOAT) AS Correlation_Coeff,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'OPERATOR_ID=\w+',1,1),'OPERATO
R_ID=','') AS VARCHAR(20)) AS OPERATOR_ID,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Calibration
ID=\d+',1,1),'Calibration ID=','') AS VARCHAR(5))  AS CALIBRATION_ID,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Calibration
Method=\w+',1,1),'Calibration Method=','') AS VARCHAR(50))  AS
CALIBRATION_METHOD,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Percent Deviation=\s?\-
?\d+\.?\d*',1,1),'Percent Deviation=','') AS VARCHAR(50))  AS
Percent_Deviation,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Criteria=\s?\-
?\d+\.?\d*',1,1),'Criteria=','') AS VARCHAR(50))  AS Criteria,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Prereaction Limit=\s?\-
?\d+\.?\d*',1,1),'Prereaction Limit=','') AS VARCHAR(50))  AS
Prereaction_Limit,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Calibration
Type=\w+(\s?\w*)*',1,1),'Calibration Type=','') AS VARCHAR(50))  AS
Calibration_Type,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Server=\w+(\s?\w*)*',1,1),'Ser
ver=','') AS VARCHAR(50))  AS Server,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'C0=\s?\-
?\d+\.?\d*',1,1),'C0=','') AS FLOAT) AS C0,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'C1=\s?\-
?\d+\.?\d*',1,1),'C1=','') AS FLOAT) AS C1,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'C2=\s?\-
?\d+\.?\d*',1,1),'C2=','') AS FLOAT) AS C2,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'C3=\s?\-
?\d+\.?\d*',1,1),'C3=','') AS FLOAT) AS C3,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'C4=\s?\-
?\d+\.?\d*',1,1),'C4=','') AS FLOAT) AS C4,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'C5=\s?\-
?\d+\.?\d*',1,1),'C5=','') AS FLOAT) AS C5,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'C6=\s?\-
?\d+\.?\d*',1,1),'C6=','') AS FLOAT) AS C6,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'C7=\s?\-
?\d+\.?\d*',1,1),'C7=','') AS FLOAT) AS C7,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'C8=\s?\-
?\d+\.?\d*',1,1),'C8=','') AS FLOAT) AS C8,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'C9=\s?\-
?\d+\.?\d*',1,1),'C9=','') AS FLOAT) AS C9,


    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'User Code=\w+',1,1),'User
Code=','') AS VARCHAR(50)) AS USER_CODE,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'R1 Pack=\w+',1,1),'R1
Pack=','') AS VARCHAR(50))  AS R1_PACK,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'R2 Pack=\w+',1,1),'R2
Pack=','') AS VARCHAR(50))  AS R2_PACK,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'RBL Precision=\w+',1,1),'RBL
Precision=','') AS VARCHAR(50))  AS RBL_Precision,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Value Of RBL=\w+',1,1),'Value
Of RBL=','') AS VARCHAR(50))  AS Value_Of_RBL,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Sample Type=\w+',1,1),'Sample
Type=','') AS VARCHAR(50))  AS SAMPLE_TYPE,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'FORMULA_NUMBER=\d+',1,1),'FORM
ULA_NUMBER=','') AS VARCHAR(4)) AS FORMULA_NUMBER,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'CalLot_R1Lot_R2Lot=\w+(\s?\w*)
*',1,1),'CalLot_R1Lot_R2Lot=','') AS VARCHAR(200))  AS CalLot_R1Lot_R2Lot,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'BLKFV_BLKMean_ABS-RB=(\s?\-
?\d+\.?\d*)+',1,1),'BLKFV_BLKMean_ABS-RB=','') AS VARCHAR(200))  AS
BLKFV_BLKMean_ABS_RB,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'STD1FV_STD1Mean_ABS-RB=(\s?\-
?\d+\.?\d*)+',1,1),'STD1FV_STD1Mean_ABS-RB=','') AS VARCHAR(200))  AS
STD1FV_STD1Mean_ABS_RB,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'STD2FV_STD2Mean_ABS-RB=(\s?\-
?\d+\.?\d*)+',1,1),'STD2FV_STD2Mean_ABS-RB=','') AS VARCHAR(200))  AS
STD2FV_STD2Mean_ABS_RB,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'STD3FV_STD3Mean_ABS-RB=(\s?\-
?\d+\.?\d*)+',1,1),'STD3FV_STD3Mean_ABS-RB=','') AS VARCHAR(200))  AS
STD3FV_STD3Mean_ABS_RB,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'STD4FV_STD4Mean_ABS-RB=(\s?\-
?\d+\.?\d*)+',1,1),'STD4FV_STD4Mean_ABS-RB=','') AS VARCHAR(200))  AS
STD4FV_STD4Mean_ABS_RB,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'STD5FV_STD5Mean_ABS-RB=(\s?\-
?\d+\.?\d*)+',1,1),'STD5FV_STD5Mean_ABS-RB=','') AS VARCHAR(200))  AS
STD5FV_STD5Mean_ABS_RB,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'RBL_Check_Results=\s?\w+(\s?\w
+|\s?\-+)*',1,1),'RBL_Check_Results=','') AS VARCHAR(50)) AS
RBL_Check_Results,

    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'ALLERGEN_CODE=\w+',1,1),'ALLER
GEN_CODE=','') AS VARCHAR(20)) AS ALLERGEN_CODE,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'ALLERGEN_LOT=\w+',1,1),'ALLERG
EN_LOT=','') AS VARCHAR(20)) AS ALLERGEN_LOT,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Standard A
Lot=\w+',1,1),'Standard A Lot=','') AS VARCHAR(50)) AS STDA_LOT,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Standard B
Lot=\w+',1,1),'Standard B Lot=','') AS VARCHAR(50)) AS STDB_LOT,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Flush Lot=\w+',1,1),'Flush
Lot=','') AS VARCHAR(50)) AS FLUSH_LOT,
    CASE WHEN REGEXP_INSTR(DETAILS,'Salt Soln Lot=') > 0 AND
REGEXP_INSTR(DETAILS,'(..?\/..?\/.. ..\:..)') > 0 THEN
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Salt Soln
Lot=\s?..?\/..?\/.. ..\:..',1,1),'Salt Soln Lot=','') AS VARCHAR(50))
         WHEN REGEXP_INSTR(DETAILS,'Salt Soln Lot=') > 0
THEN  CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Salt Soln
Lot=\w+',1,1),'Salt Soln Lot=','') AS VARCHAR(50)) END AS Salt_Soln_Lot,

    CASE WHEN REGEXP_INSTR(DETAILS,'Diluent Lot=\s?..\:..') > 0  THEN
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Diluent
Lot=\s?..\:..',1,1),'Diluent Lot=','') AS VARCHAR(50))
```

```
         WHEN REGEXP_INSTR(DETAILS,'Diluent Lot=') > 0
THEN  CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Diluent
Lot=\w+',1,1),'Diluent Lot=','') AS VARCHAR(50)) END AS Diluent_Lot,
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Na Slope=\s?\-
?\d+\.?\d*',1,1),'Na Slope=','') AS VARCHAR(50))  AS Na_Slope,
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'K Slope=\s?\-
?\d+\.?\d*',1,1),'K Slope=','') AS VARCHAR(50))  AS K_Slope,
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Cl Slope=\s?\-
?\d+\.?\d*',1,1),'Cl Slope=','') AS VARCHAR(50))  AS Cl_Slope,
     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Air Detect=\s?\-
?\d+\.?\d*',1,1),'Air Detect=','') AS VARCHAR(50))  AS Air_Detect,
     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Liquid=\s?\-
?\d+\.?\d*',1,1),'Liquid=','') AS VARCHAR(50))  AS Liquid,
     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'CALR_SAMP_ID=\w+',1,1),'CALR_
SAMP_ID=','') AS VARCHAR(50))  AS CALR_SAMP_ID,
     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'TEST_UNITS=\s?\%?\s?\w+',1,1)
,'TEST_UNITS=','') AS VARCHAR(50))  AS TEST_UNITS,
     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'CURVE_SLOPE=\s?\-
?\d+\.?\d*',1,1),'CURVE_SLOPE=','') AS VARCHAR(50))  AS CURVE_SLOPE,
     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'CURVE_INTERCEPT=\s?\-
?\d+\.?\d*',1,1),'CURVE_INTERCEPT=','') AS VARCHAR(50))  AS CURVE_INTERCEPT,
     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'CALR_REPL_MEAN=\s?\-
?\d+\.?\d*',1,1),'CALR_REPL_MEAN=','') AS VARCHAR(50))  AS CALR_REPL_MEAN,
     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'CALR_RSLT_CONC=\s?\-
?\d+\.?\d*',1,1),'CALR_RSLT_CONC=','') AS VARCHAR(50))  AS CALR_RSLT_CONC,
     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'CURVE_CAL_RATIO=\s?\-
?\d+\.?\d*',1,1),'CURVE_CAL_RATIO=','') AS VARCHAR(50))  AS CURVE_CAL_RATIO,
     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'CURVE_DEVIATION=\s?\-
?\d+\.?\d*',1,1),'CURVE_DEVIATION=','') AS VARCHAR(50))  AS CURVE_DEVIATION,
     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'CURVE_RATIO_MIN_SD=\s?\-
?\d+\.?\d*',1,1),'CURVE_RATIO_MIN_SD=','') AS VARCHAR(50))  AS
CURVE_RATIO_MIN_SD,

     -----
CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'ISE_Type_CalLot_Exp=\w+(\s?\w*\s?\
.?\.*)*',1,1),'ISE_Type_CalLot_Exp=','') AS VARCHAR(200))  AS
ISE_Type_CalLot_Exp,
     CASE WHEN DETAILS LIKE '%ISE_Type_CalLot_Exp=%' AND DETAILS
LIKE  '%__/__/__%' THEN
          CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'ISE_Type_CalLot_Exp=\w+(
\/?\s*\w*|\s*\w*\/?\w*\/?\w*)*',1,1),'ISE_Type_CalLot_Exp=','') AS
VARCHAR(200))
     ELSE
          CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'ISE_Type_CalLot_Exp=\w+(
\s?\w*\s?\.?\.*)*',1,1),'ISE_Type_CalLot_Exp=','') AS VARCHAR(200)) END  AS
ISE_Type_CalLot_Exp,

    ----
RTRIM(CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'ISE_ELEC_LOT_Exp_Install=\w+
(\s*\w*|\s*\-?\d*\.?\d*)*',1,1),'ISE_ELEC_LOT_Exp_Install=','') AS
VARCHAR(50)))  AS ISE_ELEC_LOT_Exp_Install,
     CASE WHEN DETAILS LIKE '%ISE_ELEC_LOT_Exp_Install=%' AND DETAILS
LIKE  '%__/__/__%' THEN
          CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'ISE_ELEC_LOT_Exp_Install
=\w+(\/?\s*\w*|\s*\w*\/?\w*\/?\w*)*',1,1),'ISE_ELEC_LOT_Exp_Install=','') AS
VARCHAR(200))
     ELSE
```

```
          CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'ISE_ELEC_LOT_Exp_Install
=\w+(\s?\w*\s?\.?\.*)*',1,1),'ISE_ELEC_LOT_Exp_Install=','') AS VARCHAR(200))
END  AS ISE_ELEC_LOT_Exp_Install,

   ----
-  RTRIM(CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'REF_ELEC_LOT_Exp_Install=
\w+(\s*\w*|\s*\-?\d*\.?\d*)*',1,1),'REF_ELEC_LOT_Exp_Install=','') AS
VARCHAR(50))) AS REF_ELEC_LOT_Exp_Install,
        CASE WHEN DETAILS LIKE '%REF_ELEC_LOT_Exp_Install=%' AND DETAILS
LIKE  '%__/__/__%' THEN
          CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'REF_ELEC_LOT_Exp_Install
=\w+(\/?\s*\w*|\s*\w*\/?\w*\/?\w*)*',1,1),'REF_ELEC_LOT_Exp_Install=','') AS
VARCHAR(200))
     ELSE
          CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'REF_ELEC_LOT_Exp_Install
=\w+(\s?\w*\s?\.?\.*)*',1,1),'REF_ELEC_LOT_Exp_Install=','') AS VARCHAR(200))
END  AS REF_ELEC_LOT_Exp_Install,

     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'ISE_ELEC_HSTD_HBuff_LSTD_LBuf
f_Slope_Dil=\w+(\s*\-
?\w*\.?\w*)*',1,1),'ISE_ELEC_HSTD_HBuff_LSTD_LBuff_Slope_Dil=','') AS
VARCHAR(200))  AS ISE_ELEC_HSTD_HBuff_LSTD_LBuff_Slope_Dil,
     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'TH1_HSTD_HBuff_LSTD_LBuff=\w+
(\s*\-?\w*\.?\w*)*',1,1),'TH1_HSTD_HBuff_LSTD_LBuff=','') AS
VARCHAR(200))  AS TH1_HSTD_HBuff_LSTD_LBuff,
     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'TH2_HSTD_HBuff_LSTD_LBuff=\w+
(\s*\-?\w*\.?\w*)*',1,1),'TH2_HSTD_HBuff_LSTD_LBuff=','') AS
VARCHAR(200))  AS TH2_HSTD_HBuff_LSTD_LBuff,
     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'CL_BIAS_HSTD=\s*\-
?\w+\.?\w*',1,1),'CL_BIAS_HSTD=','') AS VARCHAR(200))  AS CL_BIAS_HSTD,
     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'REF_ELEC_HSTD= \s*\-
?\w+\.?\w*',1,1),'REF_ELEC_HSTD= ','') AS VARCHAR(200))  AS REF_ELEC_HSTD,

     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'CalibratorName=\w+',1,1),'Cal
ibratorName=','') AS VARCHAR(50)) AS CALIBRATOR_NAME,
     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'CalibratorID=\w+',1,1),'Calib
ratorID=','') AS VARCHAR(50)) AS CALIBRATOR_ID,
     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'ReagentProductCode=\w+',1,1),
'ReagentProductCode=','') AS VARCHAR(50)) AS REAGENT_PRODUCT_CODE,
     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'MethodName=\w+',1,1),'MethodN
ame=','') AS VARCHAR(50)) AS METHOD_NAME,
     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-
Concentration=\w+(\.?\w*|\,?\w*)*',1,1),'L1-Concentration=','') AS
VARCHAR(50))  AS L1_CONCENTRATION,
     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-
Sample_ID=\:\%\w+',1,1),'L1-Sample_ID=','') AS VARCHAR(50))  AS L1_SAMPLE_ID,

    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-
Result=\w+(\.?\w*)',1,1),'L1-Result=','') AS VARCHAR(50))  AS L1_RESULT,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-
RLUMean=\w+(\.?\w*)',1,1),'L1-RLUMean=','') AS VARCHAR(50))  AS L1_RLUMEAN,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-
RLUReplicate1=\w+(\.?\w*)',1,1),'L1-RLUReplicate1=','') AS VARCHAR(50))  AS
L1_RLUREPLICATE1,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-
RLUReplicate2=\w+(\.?\w*)',1,1),'L1-RLUReplicate2=','') AS VARCHAR(50))  AS
L1_RLUREPLICATE2,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-
RLUReplicate3=\w+(\.?\w*)',1,1),'L1-RLUReplicate3=','') AS VARCHAR(50))  AS
L1_RLUREPLICATE3,

    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L1-Flags=\w+',1,1),'L1-
Flags=','') AS VARCHAR(50))  AS L1_FLAGS,

     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-
Concentration=\w+(\.?\w*|\,?\w*)*',1,1),'L2-Concentration=','') AS
VARCHAR(50))  AS L2_CONCENTRATION,
     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-
Sample_ID=\:\%\w+',1,1),'L2-Sample_ID=','') AS VARCHAR(50))  AS L2_SAMPLE_ID,

    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-
Result=\w+(\.?\w*)',1,1),'L2-Result=','') AS VARCHAR(50))  AS L2_RESULT,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-
RLUMean=\w+(\.?\w*)',1,1),'L2-RLUMean=','') AS VARCHAR(50))  AS L2_RLUMEAN,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-
RLUReplicate1=\w+(\.?\w*)',1,1),'L2-RLUReplicate1=','') AS VARCHAR(50))  AS
L2_RLUREPLICATE1,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-
RLUReplicate2=\w+(\.?\w*)',1,1),'L2-RLUReplicate2=','') AS VARCHAR(50))  AS
L2_RLUREPLICATE2,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-
RLUReplicate3=\w+(\.?\w*)',1,1),'L2-RLUReplicate3=','') AS VARCHAR(50))  AS
L2_RLUREPLICATE3,

    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'L2-Flags=\w+',1,1),'L2-
Flags=','') AS VARCHAR(50))  AS L2_FLAGS,

     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'Units=\w+(\/\w+)*',1,1),'Unit
s=','') AS VARCHAR(50))  AS UNITS,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'CheckSum=\w+',1,1),'CheckSum='
,'') AS VARCHAR(50))  AS CHECKSUM,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'RackID=\w+',1,1),'RackID=','')
AS VARCHAR(50))  AS RACKID,
        CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'OrderDateTime=\d+\/\d+\/\d
+ \d+\:\d+\:\d+ \w+',1,1),'OrderDateTime=','') AS VARCHAR(50) ) AS
ORDERDATETIME_TXT,

   CAST(CASE
        WHEN ORDERDATETIME_TXT LIKE '_/_/____ _:%'  THEN
'0'||SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 2) || '0' ||
SUBSTRING(ORDERDATETIME_TXT FROM 3 FOR 2) || SUBSTRING(ORDERDATETIME_TXT FROM
5 FOR 5) || '0' || SUBSTRING(ORDERDATETIME_TXT FROM 10)
        WHEN ORDERDATETIME_TXT LIKE '_/__/____ _:%' THEN
'0'||SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 2) || SUBSTRING(ORDERDATETIME_TXT
FROM 3 FOR 8) || '0' || SUBSTRING(ORDERDATETIME_TXT FROM 11)
        WHEN ORDERDATETIME_TXT LIKE '_/_/____ __:%' THEN '0' ||
SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 2) || '0' ||
SUBSTRING(ORDERDATETIME_TXT FROM 3)
        WHEN ORDERDATETIME_TXT LIKE '_/__/____ __:%' THEN
'0'||SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 2) || SUBSTRING(ORDERDATETIME_TXT
FROM 3)
        WHEN ORDERDATETIME_TXT LIKE '__/_/____ __:%' THEN
SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 3) || '0' ||
SUBSTRING(ORDERDATETIME_TXT FROM 4)
```

```
        WHEN ORDERDATETIME_TXT LIKE '__/__/____ _:%'THEN
SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 11) || '0' ||
SUBSTRING(ORDERDATETIME_TXT FROM 12)
        WHEN ORDERDATETIME_TXT LIKE '__/_/____ _:%' THEN
SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 3) || '0' ||
SUBSTRING(ORDERDATETIME_TXT FROM 4 FOR 7) || '0' ||
SUBSTRING(ORDERDATETIME_TXT FROM 11)

        ELSE ORDERDATETIME_TXT
     END AS TIMESTAMP(0) FORMAT 'mm/dd/YYYYBhh:mi:ssBt') AS ORDERDATETIME,


    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'SampleStatus=\w+',1,1),'Sample
Status=','') AS VARCHAR(50))  AS SAMPLESTATUS,
     CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'UserNameSystemName=\w+(\-
?\w*)*',1,1),'UserNameSystemName=','') AS VARCHAR(50))  AS
USERNAMESYSTEMNAME,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'ReagentName=\w+',1,1),'Reagent
Name=','') AS VARCHAR(50))  AS REAGENTNAME,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'ReagentPackID=\w+',1,1),'Reage
ntPackID=','') AS VARCHAR(50))  AS REAGENTPACKID,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'AspirationDate=\d+\/\d+\/\d+
\d+\:\d+\:\d+ \w+',1,1),'AspirationDate=','') AS VARCHAR(50)) AS
ASPIRATIONDATE_TXT,

       CAST(CASE
        WHEN ASPIRATIONDATE_TXT LIKE '_/_/____ _:%'  THEN
'0'||SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 2) || '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 3 FOR 2) || SUBSTRING(ASPIRATIONDATE_TXT
FROM 5 FOR 5) || '0' || SUBSTRING(ASPIRATIONDATE_TXT FROM 10)
        WHEN ASPIRATIONDATE_TXT LIKE '_/__/____ _:%' THEN
'0'||SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 2) ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 3 FOR 8) || '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 11)
        WHEN ASPIRATIONDATE_TXT LIKE '_/_/____ __:%' THEN '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 2) || '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 3)
        WHEN ASPIRATIONDATE_TXT LIKE '_/__/____ __:%' THEN
'0'||SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 2) ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 3)
        WHEN ASPIRATIONDATE_TXT LIKE '__/_/____ __:%' THEN
SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 3) || '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 4)
        WHEN ASPIRATIONDATE_TXT LIKE '__/__/____ _:%'THEN
SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 11) || '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 12)
        WHEN ASPIRATIONDATE_TXT LIKE '__/_/____ _:%' THEN
SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 3) || '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 4 FOR 7) || '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 11)

        ELSE ASPIRATIONDATE_TXT
     END AS TIMESTAMP(0) FORMAT 'mm/dd/YYYYBhh:mi:ssBt') AS ASPIRATIONDATE,
    CAST(REGEXP_REPLACE(REGEXP_SUBSTR(DETAILS,'DBSynchID=\w+',1,1),'DBSynchID
=','') AS VARCHAR(50))  AS DBSYNCHID
      from
```

```
         "HC_PRD_D_RDDL_INVITRO_0_1_0_0_0_0_0_0"."AAA_D_CALIBRATION_DETAIL"
as D_CALIBRATION left join
         "HC_PRD_D_ACLO_BAS_0_14_0_0_0_0_0_0"."NODE" as NODE
            on
            (
               D_CALIBRATION."NODE_ID" = NODE."NODE_ID"
            ) left join
         "HC_PRD_D_ACLO_BAS_0_13_0_0_0_0_0_0"."PRODUCT_SECTION" as
PRODUCT_SECTION
            on
            (
               NODE."PRODUCT_SECTION_ID" =
PRODUCT_SECTION."PRODUCT_SECTION_ID"
            ) left join
         "HC_PRD_D_ACLO_BAS_0_14_0_0_0_0_0_0"."COUNTRY_NODE" as COUNTRY
            on
            (
               NODE."COUNTRY_ID" = COUNTRY."COUNTRY_ID"
               and NODE."NODE_ID" = COUNTRY."NODE_ID"
            )
      where
         EXTRACT(YEAR FROM D_CALIBRATION."CREATED") >= 2016
         and NODE."MATERIAL_NUMBER" IN
('11065004','11065006','11065464','11066000','11066001','11067000','11068008'
,'11069001','11069004','11069018','11069020')) with data;
```

```
proc sql;
   connect to TERADATA
   (
       DBSLICEPARM=(THREADED_APPS,4)
SERVER=BASDW  AUTHDOMAIN="TeradataAuthPRD"
   );
  /* CREATE THE CALENDAR TABLE */
   execute
   (
      CREATE VOLATILE TABLE CALENDAR_TABLE AS
      (SELECT
       CALENDAR_DATE
       FROM SYS_CALENDAR.Calendar
       WHERE SYS_CALENDAR.Calendar.calendar_date BETWEEN  '2016-04-18' AND
CURRENT_DATE
    /*WHERE SYS_CALENDAR.Calendar.calendar_date BETWEEN
ADD_MONTHS(CURRENT_DATE,-13) AND CURRENT_DATE */
) WITH DATA
ON COMMIT PRESERVE ROWS

   ) by TERADATA;
   execute (commit) by TERADATA;
```

```
proc sql;
   connect to TERADATA
   (
       DBSLICEPARM=(THREADED_APPS,4) SERVER=BASDW
AUTHDOMAIN="TeradataAuthPRD"
   );
   execute
   (
      create view "HC_PRD_D_RDDL_INVITRO_0_1_0_0_0_0_0_0"."AAA_D_CALIBRATION"
as
      select
         D_CALIBRATION."ACTIONOPERATIONTASK_ID",
         D_CALIBRATION."ASSAYLOT",
         D_CALIBRATION."ASSAYNAME",
         NODE."BAS_LOAD_DTTM",
         D_CALIBRATION."CALIBRATORLOT",
         NODE."CITY",
         COUNTRY."COUNTRY_NAME",
         D_CALIBRATION."CREATED",
         D_CALIBRATION."CREATED_DATE",
         NODE."ACCOUNT_NAME" as "CUSTOMER_NAME",
         coalesce(CAST(substr(D_CALIBRATION."DATETIMEUTC",1,20) AS
TIMESTAMP(6) FORMAT 'YY-MM-DD:HH:MI:SS'), null) as "DATETIMEUTC",
         D_CALIBRATION."DETAILS",
         coalesce(CAST(substr(D_CALIBRATION."EXPIRATIONDATETIMEUTC",1,20) AS
TIMESTAMP(6) FORMAT 'YY-MM-DD:HH:MI:SS'), null) as "EXPIRATIONDATETIMEUTC",
         D_CALIBRATION."ID",
         NODE."MATERIAL_NUMBER",
         NODE."NODE_ID",
         NODE."NODE_NAME",
         PRODUCT_SECTION."PRODUCT_SECTION_NAME",
         NODE."SERIAL_NUMBER",
         D_CALIBRATION."STATUS",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'SLOPE=\s?
\-?\d+\.?\d*',1,1),'SLOPE=','') AS FLOAT) as "SLOPE",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'INTERCEPT
=\s?\-?\d+\.?\d*',1,1),'INTERCEPT=','') AS FLOAT) as "INTERCEPT",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Correlati
on Coeff=\s?\-?\d+\.?\d*',1,1),'Correlation Coeff=','') AS FLOAT) as
"CORRELATION_COEFF",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'OPERATOR_
ID=\w+',1,1),'OPERATOR_ID=','') AS VARCHAR(20)) as "OPERATOR_ID",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Calibrati
on ID=\d+',1,1),'Calibration ID=','') AS VARCHAR(5)) as "CALIBRATION_ID",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Calibrati
on Method=\w+',1,1),'Calibration Method=','') AS VARCHAR(50)) as
"CALIBRATION_METHOD",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Percent
Deviation=\s?\-?\d+\.?\d*',1,1),'Percent Deviation=','') AS VARCHAR(50)) as
"PERCENT_DEVIATION",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Criteria=
\s?\-?\d+\.?\d*',1,1),'Criteria=','') AS VARCHAR(50)) as "CRITERIA",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Prereacti
on Limit=\s?\-?\d+\.?\d*',1,1),'Prereaction Limit=','') AS VARCHAR(50)) as
"PREREACTION_LIMIT",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Calibrati
on Type=\w+(\s?\w*)*',1,1),'Calibration Type=','') AS VARCHAR(50)) as
"CALIBRATION_TYPE",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Server=\w
+(\s?\w*)*',1,1),'Server=','') AS VARCHAR(50)) as "SERVER",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'C0=\s?\-
?\d+\.?\d*',1,1),'C0=','') AS FLOAT) as "C0",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'C1=\s?\-
?\d+\.?\d*',1,1),'C1=','') AS FLOAT) as "C1",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'C2=\s?\-
?\d+\.?\d*',1,1),'C2=','') AS FLOAT) as "C2",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'C3=\s?\-
?\d+\.?\d*',1,1),'C3=','') AS FLOAT) as "C3",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'C4=\s?\-
?\d+\.?\d*',1,1),'C4=','') AS FLOAT) as "C4",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'C5=\s?\-
?\d+\.?\d*',1,1),'C5=','') AS FLOAT) as "C5",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'C6=\s?\-
?\d+\.?\d*',1,1),'C6=','') AS FLOAT) as "C6",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'C7=\s?\-
?\d+\.?\d*',1,1),'C7=','') AS FLOAT) as "C7",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'C8=\s?\-
?\d+\.?\d*',1,1),'C8=','') AS FLOAT) as "C8",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'C9=\s?\-
?\d+\.?\d*',1,1),'C9=','') AS FLOAT) as "C9",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'User
Code=\w+',1,1),'User Code=','') AS VARCHAR(50)) as "USER_CODE",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'R1
Pack=\w+',1,1),'R1 Pack=','') AS VARCHAR(50)) as "R1_PACK",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'R2
Pack=\w+',1,1),'R2 Pack=','') AS VARCHAR(50)) as "R2_PACK",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'RBL
Precision=\w+',1,1),'RBL Precision=','') AS VARCHAR(50)) as "RBL_PRECISION",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Value Of
RBL=\w+',1,1),'Value Of RBL=','') AS VARCHAR(50)) as "VALUE_OF_RBL",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Sample
Type=\w+',1,1),'Sample Type=','') AS VARCHAR(50)) as "SAMPLE_TYPE",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'FORMULA_N
UMBER=\d+',1,1),'FORMULA_NUMBER=','') AS VARCHAR(4)) as "FORMULA_NUMBER",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'CalLot_R1
Lot_R2Lot=\w+(\s?\w*)*',1,1),'CalLot_R1Lot_R2Lot=','') AS VARCHAR(200)) as
"CALLOT_R1LOT_R2LOT",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'BLKFV_BLK
Mean_ABS-RB=(\s?\-?\d+\.?\d*)+',1,1),'BLKFV_BLKMean_ABS-RB=','') AS
VARCHAR(200)) as "BLKFV_BLKMEAN_ABS_RB",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR
         (D_CALIBRATION."DETAILS",'STD1FV_STD1Mean_ABS-RB=(\s?\-
?\d+\.?\d*)+',1,1),'STD1FV_STD1Mean_ABS-RB=','') AS VARCHAR(200)) as
"STD1FV_STDMEAN_ABS_RB",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR
         (D_CALIBRATION."DETAILS",'STD2FV_STD2Mean_ABS-RB=(\s?\-
?\d+\.?\d*)+',1,1),'STD2FV_STD2Mean_ABS-RB=','') AS VARCHAR(200)) as
"STD2FV_STDMEAN_ABS_RB",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR
         (D_CALIBRATION."DETAILS",'STD3FV_STD3Mean_ABS-RB=(\s?\-
?\d+\.?\d*)+',1,1),'STD3FV_STD3Mean_ABS-RB=','') AS VARCHAR(200)) as
"STD3FV_STDMEAN_ABS_RB",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR
         (D_CALIBRATION."DETAILS",'STD4FV_STD4Mean_ABS-RB=(\s?\-
?\d+\.?\d*)+',1,1),'STD4FV_STD4Mean_ABS-RB=','') AS VARCHAR(200)) as
"STD4FV_STDMEAN_ABS_RB",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR
         (D_CALIBRATION."DETAILS",'STD5FV_STD5Mean_ABS-RB=(\s?\-
?\d+\.?\d*)+',1,1),'STD5FV_STD5Mean_ABS-RB=','') AS VARCHAR(200)) as
"STD5FV_STDMEAN_ABS_RB",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'RBL_Check
_Results=\s?\w+(\s?\w+|\s?\-+)*',1,1),'RBL_Check_Results=','') AS
VARCHAR(50)) as "RBL_CHECK_RESULTS",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'ALLERGEN_
CODE=\w+',1,1),'ALLERGEN_CODE=','') AS VARCHAR(20)) as "ALLERGEN_CODE",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'ALLERGEN_
LOT=\w+',1,1),'ALLERGEN_LOT=','') AS VARCHAR(20)) as "ALLERGEN_LOT",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Standard
A Lot=\w+',1,1),'Standard A Lot=','') AS VARCHAR(50)) as "STDA_LOT",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Standard
B Lot=\w+',1,1),'Standard B Lot=','') AS VARCHAR(50)) as "STDB_LOT",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Flush
Lot=\w+',1,1),'Flush Lot=','') AS VARCHAR(50)) as "FLUSH_LOT",
         CASE WHEN REGEXP_INSTR(D_CALIBRATION."DETAILS",'Salt Soln Lot=') > 0
AND REGEXP_INSTR(D_CALIBRATION."DETAILS",'(..?\/..?\/.. ..\:..)') > 0 THEN
                  CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'
Salt Soln Lot=\s?..?\/..?\/.. ..\:..',1,1),'Salt Soln Lot=','') AS
VARCHAR(50))
                  WHEN REGEXP_INSTR(D_CALIBRATION."DETAILS",'Salt Soln Lot=')
> 0 THEN  CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Salt
Soln Lot=\w+',1,1),'Salt Soln Lot=','') AS VARCHAR(50)) END as
"SALT_SOLN_LOT",
         CASE WHEN REGEXP_INSTR(D_CALIBRATION."DETAILS",'Diluent
Lot=\s?..\:..') > 0  THEN
                  CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'
Diluent Lot=\s?..\:..',1,1),'Diluent Lot=','') AS VARCHAR(50))
                  WHEN REGEXP_INSTR(D_CALIBRATION."DETAILS",'Diluent Lot=') >
0 THEN  CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Diluent
Lot=\w+',1,1),'Diluent Lot=','') AS VARCHAR(50)) END as "DILUENT_LOT",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Na
Slope=\s?\-?\d+\.?\d*',1,1),'Na Slope=','') AS VARCHAR(50)) as "NA_SLOPE",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'K
Slope=\s?\-?\d+\.?\d*',1,1),'K Slope=','') AS VARCHAR(50)) as "K_SLOPE",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Cl
Slope=\s?\-?\d+\.?\d*',1,1),'Cl Slope=','') AS VARCHAR(50)) as "CL_SLOPE",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Air
Detect=\s?\-?\d+\.?\d*',1,1),'Air Detect=','') AS VARCHAR(50)) as
"AIR_DETECT",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Liquid=\s
?\-?\d+\.?\d*',1,1),'Liquid=','') AS VARCHAR(50)) as "LIQUID",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'CALR_SAMP
_ID=\w+',1,1),'CALR_SAMP_ID=','') AS VARCHAR(50)) as "CALR_SAMP_ID",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'TEST_UNIT
S=\s?\%?\s?\w+',1,1),'TEST_UNITS=','') AS VARCHAR(50)) as "TEST_UNITS",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'CURVE_SLO
PE=\s?\-?\d+\.?\d*',1,1),'CURVE_SLOPE=','') AS VARCHAR(50)) as "CURVE_SLOPE",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'CURVE_INT
ERCEPT=\s?\-?\d+\.?\d*',1,1),'CURVE_INTERCEPT=','') AS VARCHAR(50)) as
"CURVE_INTERCEPT",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'CALR_REPL
_MEAN=\s?\-?\d+\.?\d*',1,1),'CALR_REPL_MEAN=','') AS VARCHAR(50)) as
"CALR_REPL_MEAN",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'CALR_RSLT
_CONC=\s?\-?\d+\.?\d*',1,1),'CALR_RSLT_CONC=','') AS VARCHAR(50)) as
"CALR_RSLT_CONC",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'CURVE_CAL
_RATIO=\s?\-?\d+\.?\d*',1,1),'CURVE_CAL_RATIO=','') AS VARCHAR(50)) as
"CURVE_CAL_RATIO",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'CURVE_DEV
IATION=\s?\-?\d+\.?\d*',1,1),'CURVE_DEVIATION=','') AS VARCHAR(50)) as
"CURVE_DEVIATION",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'CURVE_RAT
IO_MIN_SD=\s?\-?\d+\.?\d*',1,1),'CURVE_RATIO_MIN_SD=','') AS VARCHAR(50)) as
"CURVE_RATIO_MIN_SD",
         CASE WHEN D_CALIBRATION."DETAILS" LIKE '%ISE_Type_CalLot_Exp=%' AND
D_CALIBRATION."DETAILS" LIKE  '%__/__/__%' THEN
                   CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",
'ISE_Type_CalLot_Exp=\w+(\/?\s*\w*|\s*\w*\/?\w*\/?\w*)*',1,1),'ISE_Type_CalLo
t_Exp=','') AS VARCHAR(200))
              ELSE
                   CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",
'ISE_Type_CalLot_Exp=\w+(\s?\w*\s?\.?\.*)*',1,1),'ISE_Type_CalLot_Exp=','')
AS VARCHAR(200)) END as "ISE_TYPE_CALLOT_EXP",
         CASE WHEN D_CALIBRATION."DETAILS" LIKE '%ISE_ELEC_LOT_Exp_Install=%'
AND D_CALIBRATION."DETAILS" LIKE  '%__/__/__%' THEN
                   CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",
'ISE_ELEC_LOT_Exp_Install=\w+(\/?\s*\w*|\s*\w*\/?\w*\/?\w*)*',1,1),'ISE_ELEC_
LOT_Exp_Install=','') AS VARCHAR(200))
              ELSE
                   CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",
'ISE_ELEC_LOT_Exp_Install=\w+(\s?\w*\s?\.?\.*)*',1,1),'ISE_ELEC_LOT_Exp_Insta
ll=','') AS VARCHAR(200)) END as "ISE_ELEC_LOT_EXP_INSTALL",
         CASE WHEN D_CALIBRATION."DETAILS" LIKE '%REF_ELEC_LOT_Exp_Install=%'
AND D_CALIBRATION."DETAILS" LIKE  '%__/__/__%' THEN
                   CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",
'REF_ELEC_LOT_Exp_Install=\w+(\/?\s*\w*|\s*\w*\/?\w*\/?\w*)*',1,1),'REF_ELEC_
LOT_Exp_Install=','') AS VARCHAR(200))
              ELSE
                   CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",
'REF_ELEC_LOT_Exp_Install=\w+(\s?\w*\s?\.?\.*)*',1,1),'REF_ELEC_LOT_Exp_Insta
ll=','') AS VARCHAR(200)) END as "REF_ELEC_LOT_EXP_INSTALL",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR
         (D_CALIBRATION."DETAILS",'ISE_ELEC_HSTD_HBuff_LSTD_LBuff_Slope_Dil=\
w+(\s*\-?\w*\.?\w*)*',1,1),'ISE_ELEC_HSTD_HBuff_LSTD_LBuff_Slope_Dil=','') AS
VARCHAR(200)) as "ISE_ELEC_HSTD_HBUFF_LSTD_LBUFF_S",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR
         (D_CALIBRATION."DETAILS",'TH1_HSTD_HBuff_LSTD_LBuff=\w+(\s*\-
?\w*\.?\w*)*',1,1),'TH1_HSTD_HBuff_LSTD_LBuff=','') AS VARCHAR(200)) as
"TH1_HSTD_HBuff_LSTD_LBuff",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR
         (D_CALIBRATION."DETAILS",'TH2_HSTD_HBuff_LSTD_LBuff=\w+(\s*\-
?\w*\.?\w*)*',1,1),'TH2_HSTD_HBuff_LSTD_LBuff=','') AS VARCHAR(200)) as
"TH2_HSTD_HBUFF_LSTD_LBUFF",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'CL_BIAS_H
STD=\s*\-?\w+\.?\w*',1,1),'CL_BIAS_HSTD=','') AS VARCHAR(200)) as
"CL_BIAS_HSTD",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'REF_ELEC_
HSTD= \s*\-?\w+\.?\w*',1,1),'REF_ELEC_HSTD= ','') AS VARCHAR(200)) as
"REF_ELEC_HSTD",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Calibrato
rName=\w+',1,1),'CalibratorName=','') AS VARCHAR(50)) as "CALIBRATOR_NAME",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Calibrato
rID=\w+',1,1),'CalibratorID=','') AS VARCHAR(50)) as "CALIBRATOR_ID",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'ReagentPr
oductCode=\w+',1,1),'ReagentProductCode=','') AS VARCHAR(50)) as
"REAGENT_PRODUCT_CODE",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'MethodNam
e=\w+',1,1),'MethodName=','') AS VARCHAR(50)) as "METHOD_NAME",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'L1-
Sample_ID=\:\%\w+',1,1),'L1-Sample_ID=','') AS VARCHAR(50)) as
"L1_SAMPLE_ID",
         CASE WHEN REGEXP_INSTR(D_CALIBRATION."DETAILS",'L1-
Concentration=\d+(\.?\d*|\,?\d*)*',1,1,0) > 0
                       THEN
CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'L1-
Concentration=\d+(\.?\d*|\,?\d*)*',1,1),'L1-Concentration=','') AS FLOAT)
                       ELSE NULL END as "L1_CONCENTRATION",
         CASE WHEN REGEXP_INSTR(D_CALIBRATION."DETAILS",'L1-
Result=\d+(\.?\d*)',1,1,0) > 0
                       THEN
CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'L1-
Result=\d+(\.?\d*)',1,1),'L1-Result=','') AS FLOAT)
                       ELSE NULL END as "L1_RESULT",
         CASE WHEN REGEXP_INSTR(D_CALIBRATION."DETAILS",'L1-
RLUMean=\d+(\.?\d*)',1,1,0) > 0
                       THEN
CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'L1-
RLUMean=\d+(\.?\d*)',1,1),'L1-RLUMean=','') AS FLOAT)
                       ELSE NULL END as "L1_RLUMEAN",
         CASE WHEN REGEXP_INSTR(D_CALIBRATION."DETAILS",'L1-
RLUReplicate1=\d+(\.?\d*)',1,1,0) > 0
                       THEN
CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'L1-
RLUReplicate1=\d+(\.?\d*)',1,1),'L1-RLUReplicate1=','') AS FLOAT)
                       ELSE NULL END as "L1_RLUREPLICATE1",
         CASE WHEN REGEXP_INSTR(D_CALIBRATION."DETAILS",'L1-
RLUReplicate2=\d+(\.?\d*)',1,1,0) > 0
                       THEN
CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'L1-
RLUReplicate2=\d+(\.?\d*)',1,1),'L1-RLUReplicate2=','') AS FLOAT)
                       ELSE NULL END as "L1_RLUREPLICATE2",
         CASE WHEN REGEXP_INSTR(D_CALIBRATION."DETAILS",'L1-
RLUReplicate3=\d+(\.?\d*)',1,1,0) > 0
                       THEN
CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'L1-
RLUReplicate3=\d+(\.?\d*)',1,1),'L1-RLUReplicate3=','') AS FLOAT)
                       ELSE NULL END as "L1_RLUREPLICATE3",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'L1-
Flags=\w+',1,1),'L1-Flags=','') AS VARCHAR(50)) as "L1_FLAGS",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'L2-
Sample_ID=\:\%\w+',1,1),'L2-Sample_ID=','') AS VARCHAR(50)) as
"L2_SAMPLE_ID",
         CASE WHEN REGEXP_INSTR(D_CALIBRATION."DETAILS",'L2-
Concentration=\d+(\.?\d*|\,?\d*)*',1,1,0) > 0
                       THEN
CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'L2-
Concentration=\d+(\.?\d*|\,?\d*)*',1,1),'L2-Concentration=','') AS FLOAT)
                       ELSE NULL END as "L2_CONCENTRATION",
         CASE WHEN REGEXP_INSTR(D_CALIBRATION."DETAILS",'L2-
Result=\d+(\.?\d*)',1,1,0) > 0
                       THEN
CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'L2-
Result=\d+(\.?\d*)',1,1),'L2-Result=','') AS FLOAT)
                       ELSE NULL END as "L2_RESULT",
         CASE WHEN REGEXP_INSTR(D_CALIBRATION."DETAILS",'L2-
RLUMean=\d+(\.?\d*)',1,1,0) > 0
                       THEN
CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'L2-
RLUMean=\d+(\.?\d*)',1,1),'L2-RLUMean=','') AS FLOAT)
                       ELSE NULL END as "L2_RLUMEAN",
         CASE WHEN REGEXP_INSTR(D_CALIBRATION."DETAILS",'L2-
RLUReplicate1=\d+(\.?\d*)',1,1,0) > 0
                       THEN
CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'L2-
RLUReplicate1=\d+(\.?\d*)',1,1),'L2-RLUReplicate1=','') AS FLOAT)
                       ELSE NULL END as "L2_RLUREPLICATE1",
         CASE WHEN REGEXP_INSTR(D_CALIBRATION."DETAILS",'L2-
RLUReplicate2=\d+(\.?\d*)',1,1,0) > 0
                       THEN
CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'L2-
RLUReplicate2=\d+(\.?\d*)',1,1),'L2-RLUReplicate2=','') AS FLOAT)
                       ELSE NULL END as "L2_RLUREPLICATE2",
         CASE WHEN REGEXP_INSTR(D_CALIBRATION."DETAILS",'L2-
RLUReplicate3=\d+(\.?\d*)',1,1,0) > 0
                       THEN
CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'L2-
RLUReplicate3=\d+(\.?\d*)',1,1),'L2-RLUReplicate3=','') AS FLOAT)
                       ELSE NULL END as "L2_RLUREPLICATE3",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'L2-
Flags=\w+',1,1),'L2-Flags=','') AS VARCHAR(50)) as "L2_FLAGS",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Units=\w+
(\/\w+)*',1,1),'Units=','') AS VARCHAR(50)) as "UNITS",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'CheckSum=
\w+',1,1),'CheckSum=','') AS VARCHAR(50)) as "CHECKSUM",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'RackID=\w
+',1,1),'RackID=','') AS VARCHAR(50)) as "RACKID",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'OrderDate
Time=\d+\/\d+\/\d+ \d+\:\d+\:\d+ \w+',1,1),'OrderDateTime=','') AS
VARCHAR(50)) as "ORDERDATETIME_TXT",
         CAST(CASE
                 WHEN ORDERDATETIME_TXT LIKE '_/_/____ _:%'  THEN
'0'||SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 2) || '0' ||
SUBSTRING(ORDERDATETIME_TXT FROM 3 FOR 2) || SUBSTRING(ORDERDATETIME_TXT FROM
5 FOR 5) || '0' || SUBSTRING(ORDERDATETIME_TXT FROM 10)
                 WHEN ORDERDATETIME_TXT LIKE '_/__/____ _:%' THEN
'0'||SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 2) || SUBSTRING(ORDERDATETIME_TXT
FROM 3 FOR 8) || '0' || SUBSTRING(ORDERDATETIME_TXT FROM 11)
                 WHEN ORDERDATETIME_TXT LIKE '_/_/____ __:%' THEN '0' ||
SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 2) || '0' ||
SUBSTRING(ORDERDATETIME_TXT FROM 3)
                 WHEN ORDERDATETIME_TXT LIKE '_/__/____ __:%' THEN
'0'||SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 2) || SUBSTRING(ORDERDATETIME_TXT
FROM 3)
```

```
                 WHEN ORDERDATETIME_TXT LIKE '__/_/____ __:%' THEN
SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 3) || '0' ||
SUBSTRING(ORDERDATETIME_TXT FROM 4)
                 WHEN ORDERDATETIME_TXT LIKE '__/__/____ _:%'THEN
SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 11) || '0' ||
SUBSTRING(ORDERDATETIME_TXT FROM 12)
                 WHEN ORDERDATETIME_TXT LIKE '__/_/____ _:%' THEN
SUBSTRING(ORDERDATETIME_TXT FROM 1 FOR 3) || '0' ||
SUBSTRING(ORDERDATETIME_TXT FROM 4 FOR 7) || '0' ||
SUBSTRING(ORDERDATETIME_TXT FROM 11)

                 ELSE ORDERDATETIME_TXT
              END AS TIMESTAMP(0) FORMAT 'mm/dd/YYYYBhh:mi:ssBt') as
"ORDERDATETIME",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'SampleSta
tus=\w+',1,1),'SampleStatus=','') AS VARCHAR(50)) as "SAMPLESTATUS",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'UserNameS
ystemName=\w+(\-?\w*)*',1,1),'UserNameSystemName=','') AS VARCHAR(50)) as
"USERNAMESYSTEMNAME",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'ReagentNa
me=\w+',1,1),'ReagentName=','') AS VARCHAR(50)) as "REAGENTNAME",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'ReagentPa
ckID=\w+',1,1),'ReagentPackID=','') AS VARCHAR(50)) as "REAGENTPACKID",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'Aspiratio
nDate=\d+\/\d+\/\d+ \d+\:\d+\:\d+ \w+',1,1),'AspirationDate=','') AS
VARCHAR(50)) as "ASPIRATIONDATE_TXT",
         CAST(CASE
                 WHEN ASPIRATIONDATE_TXT LIKE '_/_/____ _:%'  THEN
'0'||SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 2) || '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 3 FOR 2) || SUBSTRING(ASPIRATIONDATE_TXT
FROM 5 FOR 5) || '0' || SUBSTRING(ASPIRATIONDATE_TXT FROM 10)
                 WHEN ASPIRATIONDATE_TXT LIKE '_/__/____ _:%' THEN
'0'||SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 2) ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 3 FOR 8) || '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 11)
                 WHEN ASPIRATIONDATE_TXT LIKE '_/_/____ __:%' THEN '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 2) || '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 3)
                 WHEN ASPIRATIONDATE_TXT LIKE '_/__/____ __:%' THEN
'0'||SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 2) ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 3)
                 WHEN ASPIRATIONDATE_TXT LIKE '__/_/____ __:%' THEN
SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 3) || '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 4)
                 WHEN ASPIRATIONDATE_TXT LIKE '__/__/____ _:%'THEN
SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 11) || '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 12)
                 WHEN ASPIRATIONDATE_TXT LIKE '__/_/____ _:%' THEN
SUBSTRING(ASPIRATIONDATE_TXT FROM 1 FOR 3) || '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 4 FOR 7) || '0' ||
SUBSTRING(ASPIRATIONDATE_TXT FROM 11)

                 ELSE ASPIRATIONDATE_TXT
              END AS TIMESTAMP(0) FORMAT 'mm/dd/YYYYBhh:mi:ssBt') as
"ASPIRATIONDATE",
         CAST(REGEXP_REPLACE(REGEXP_SUBSTR(D_CALIBRATION."DETAILS",'DBSynchID
=\w+',1,1),'DBSynchID=','') AS VARCHAR(50)) as "DBSYNCHID"
```

```
      from
         "HC_PRD_D_ACLO_BAS_0_14_0_0_0_0_0_0"."D_CALIBRATION" as
D_CALIBRATION left join
         "HC_PRD_D_ACLO_BAS_0_14_0_0_0_0_0_0"."NODE" as NODE
            on
            (
               D_CALIBRATION."NODE_ID" = NODE."NODE_ID"
            ) left join
         "HC_PRD_D_ACLO_BAS_0_13_0_0_0_0_0_0"."PRODUCT_SECTION" as
PRODUCT_SECTION
            on
            (
               NODE."PRODUCT_SECTION_ID" =
PRODUCT_SECTION."PRODUCT_SECTION_ID"
            ) left join
         "HC_PRD_D_ACLO_BAS_0_14_0_0_0_0_0_0"."COUNTRY_NODE" as COUNTRY
            on
            (
               NODE."COUNTRY_ID" = COUNTRY."COUNTRY_ID"
               and NODE."NODE_ID" = COUNTRY."NODE_ID"
            )
      where
         EXTRACT(YEAR FROM D_CALIBRATION."CREATED") >= 2016
         and NODE."MATERIAL_NUMBER" IN
('11065004','11065006','11065464','11066000','11066001','11067000','11068008'
,'11069001','11069004','11069018','11069020')

   ) by TERADATA;

   %rcSet(&sqlrc);

   execute (commit) by TERADATA;
   disconnect from TERADATA;
quit;
```

```
/** SDTB_DI **/
LIBNAME GSMS_T1 ORACLE  PATH=SCPROD2  SCHEMA=HPSC6  AUTHDOMAIN="
OracleAuth_SCPROD2" ;
LIBNAME WORK1 BASE "/sas/sasdata/BAS_Platform/030_Business/120_I
NVITRO/010_Data_Integration/Work/data/sandbox";
/* Access the data for INVITRO - LASR  */
LIBNAME INVLR SASIOLA  TAG=INVITRO  PORT=10027 SIGNER="https://s
asprod.healthcare.siemens.com:443/SASLASRAuthorization" HOST="sh
cffmva1p.shcbas.lokal" ;

LIBNAME GSMS_T1 ORACLE  PATH=SCPROD2  SCHEMA=HPSC6  AUTHDOMAIN="
OracleAuth_SCPROD2" ;

LIBNAME textdata BASE "/sas/sasdata/_SHARED/data/PublicDataProvi
der";
LIBNAME GSMS_V1 ORACLE  PATH=SCPROD2  SCHEMA=SHDWDB4QCA01  AUTHD
OMAIN="OracleAuth_SCPROD2" ;
```

```
options OBS=max;

libname myWork '
/sas/saswork_das/SAS_work8ED500007F84_shcsbiapp1vp/';



PROC SQL;
*DROP TABLE PROBSUMMARYM1;
CREATE TABLE HSC_PROBSUMMARYM1 AS
SELECT
BRIEF_DESCRIPTION     ,
CATEGORY   ,
CAUSE_CODE ,
CLOSE_TIME ,
COST_CENTRE     ,
CUSTOMER_NO     ,
HC_CSE_EMAIL    ,
HC_PROBLEM_ISSUE_TYPE ,
HC_PROBLEM_NUMBER     ,
INCIDENT_ID     ,
LOCATION   ,
NUMBERPRGN ,
ONSITE_TO_REPAIR ,
OPEN_TIME  ,
REFERENCE_NO    ,
RESOLUTION_CODE ,
SERIAL_NO_ ,
SMED_CITY  ,
SMED_COUNTRY    ,
SMED_CUSTOMER_NAME1   ,
SMED_HOTSITE_FLAG     ,
SMED_HOTSITE_NUMBER

FROM
GSMS_T1.PROBSUMMARYM1
WHERE SMED_COUNTRY IN ('US','UK','FR','DE','SP')
AND COST_CENTRE LIKE 'DX%'
AND DATEPART(OPEN_TIME) >= '01OCT2016'd;
QUIT;

LIBNAME INVLR SASIOLA  TAG=INVITRO  PORT=10027 SIGNER="https://s
asprod.healthcare.siemens.com:443/SASLASRAuthorization" HOST="sh
cffmva1p.shcbas.lokal" ;
LIBNAME WORK1 BASE "/sas/sasdata/BAS_Platform/030_Business/120_I
NVITRO/010_Data_Integration/Work/data/sandbox";
```

```
%let tablename=HSC_PROBSUMMARYM1;
proc metalib;
  omr (library="INVITRO - LASR" );
  SELECT("&tablename.");
  update_rule=(delete);
  report;
run;

PROC SQL;
DROP TABLE INVLR.HSC_PROBSUMMARYM1;
QUIT;


PROC APPEND BASE=INVLR.HSC_PROBSUMMARYM1 DATA=HSC_PROBSUMMARYM1;
 RUN;

/*** TABLE LOADED INTO LASER ***/
```

SAS Code

```
libname oraprod oracle user=reporter orapw=r3p0rt schema=cntl
path="@NEXUS";
libname db2prod db2 database=ldb2prod schema=CNTL user=HCSTGPRD
password=HCSTGP;
libname db2sas db2 database=ldb2prod schema=SAS user=HCSTGPRD
password=HCSTGP;
libname db2CTL db2 database=ldb2prod schema=CTL1 user=HCSTGPRD
password=HCSTGP;
libname etltest oracle user=dw_owner orapw=abc123
schema=dw_owner path="@RUMBA";
libname hcprod1 oracle user=dw_owner orapw=redarmy
path="@HCPROD1";

%let db=ORAPROD;
%let db2=db2PROD;
%let dbSAS=DB2SAS;
%let dbCTL=DB2CTL;
%let dbetl=etltest;
%let dbhc=hcprod1;



%let root=F:\SAS;

%let data=E:\USERS\DWADMIN\SASPROD\;
```

```
****************************************************************
*********;
*  USING DBSOUR PROGRAM TO CREATE A
SINGLE   ;
*  VISITS
TABLE                              ;
*  INFORMATION FOR ORACLE
DATAMART           ;
*--------------------------------------------
;
* SETUP TO RUN IN PRODUCTION
MODE            ;
*--------------------------------------------;
OPTIONS SYMBOLGEN MACROGEN MPRINT OBS=10000;
DATA
_NULL_;

 CALL
SYMPUT("CFYR",PUT(YEAR(TODAY()),4.));

 CALL
SYMPUT("CYR",SUBSTR(PUT(YEAR(TODAY()),4.),3,2));

 CALL
SYMPUT("CMTH",PUT(MONTH(TODAY()),2.));

 CALL SYMPUT("PFYR",PUT(YEAR(TODAY())-
1,4.));
 CALL SYMPUT("PYR",SUBSTR(PUT(YEAR(TODAY())-
1,$4.),3,2));
RUN;

%MACRO
RPTRUN;



PROC
SQL;



CREATE TABLE delete
AS
```

```
(SELECT DELETE_PATIENT_ID AS
PAT_ID,
        DELETE_PROVIDER_ID AS
STAFF_ID,
        DELETE_VISIT_DATE AS
VDATE,
        DELETE_TWICE_DAILY AS
TDSEQNO
 FROM
&DBCTL..APVE_delete

 %IF &CMTH < 9
%THEN
 WHERE YEAR(delete_visit_date) in (&cfyr, &pfyr)
AND;
 %ELSE

 WHERE YEAR(delete_visit_date) in (&cfyr)
AND;
       delete_pay =
'Y')
 ORDER BY PAT_ID, STAFF_ID, VDATE, TDSEQNO;
QUIT;
PROC SORT DATA=DELETE;
BY PAT_ID STAFF_ID VDATE TDSEQNO;
RUN;

PROC SQL;


CREATE TABLE UNBWO
AS

(SELECT
CASE_NUM,
        WRITEOFF_PROVIDER AS
STAFF_ID,
        WRITEOFF_VISIT_DTE AS
VDATE,
        WRITEOFF_TWICE_DAILY AS
TDSEQNO,
        WRITEOFF_AMOUNT AS
UNBWO
 FROM
&DBHC..UNBILL_WRITE_OFF
```

```
 %IF &CMTH < 9
%THEN
 WHERE YEAR(WRITEOFF_VISIT_DTE) in (&cfyr, &pfyr)
AND;
 %ELSE

 WHERE YEAR(WRITEOFF_VISIT_DTE) in (&cfyr)
AND;
       SUBSTR(WRITEOFF_PROVIDER,1,1) ^=
'A');


CREATE TABLE FIS
AS

(SELECT
CASE_NUM,
        INSUR_COUNT AS
FISNUM,
        INSUR_CODE AS
PAYOR
FROM &DB..TPCLN_INS_FISC
WHERE FISNUM < 6);
%*WHERE INSUR_COUNT
<6);

CREATE TABLE ARMS
AS

(SELECT PAYOR_CDE AS
PAYOR,
        PAYOR_CAT AS
PAYCAT
 FROM
&DB..TPARMS_PAYOR);


 CREATE TABLE VHS
AS

(SELECT
STAFF_ID,
        pat_id,

        CHGE_AMT AS
CHARGE,
        CHGE_STTS AS
STATUS,
        NO_OF_HOURS AS
HOURS,
        VISIT_DTE AS
VDATE,
        VISIT_ENTRY_DTE AS
EDATE,
        VISIT_PURP_CDE AS
VPURP,
        CASE_NO AS
CASE_NUM,

        PAYOR_CODE AS
PAY0,
        TWICE_DAILY_SEQ_NO AS
TDSEQNO,
        CHG_PAT_BILL_COMP AS
AGCY_CDE,
        CHG_PAT_COST_CTR AS
PAT_CTR,
        CHG_STAFF_COST_CTR AS
STF_CTR,
        INV_NO,

        BILLED_REIMB_AMT
AS  BILLWO,
        COB_SPLIT_IND AS
COB,
        DEN_CDE,

        VNS_RSN_CDE AS
RSN_CDE,
        HOUR(CHG_CLINICAL_TIME) AS
HR,
        MINUTE(CHG_CLINICAL_TIME) AS
MIN
 FROM
&DB2..TPCHG_CHARGE

 %IF &CMTH < 9
%THEN
 WHERE YEAR(VISIT_DTE) in (&cfyr, &pfyr)
AND;
 %ELSE
 WHERE YEAR(VISIT_DTE) in (&cfyr)
AND;
       CURRENT_ROW_IND='0')
 order by pat_id, staff_id, vdate, tdseqno;


 QUIT;




data vhs; merge vhs(in=x)
delete(in=y);
 by pat_id staff_id vdate
tdseqno;
 if
x;

 if y then status =
'DEP';
run;


PROC SUMMARY DATA=UNBWO
NWAY;
 CLASS CASE_NUM STAFF_ID VDATE
TDSEQNO;
 VAR
UNBWO;

 OUTPUT OUT=UNBWOS
SUM=;
RUN;



PROC SORT
DATA=FIS;
 BY CASE_NUM
FISNUM;
 RUN;



DATA
TFIS;
```

```
ARRAY PAY(5) $ PAY1-
PAY5('   ','   ','   ','   ','   ');
 SET
FIS;

 BY CASE_NUM
FISNUM;
 IF FIRST.CASE_NUM THEN
DO;
    DO I=1 TO
5;
    PAY(I)='   ';

    END;

 END;

 PAY(FISNUM)=PAYOR;

 IF LAST.CASE_NUM THEN
OUTPUT;
RUN;



PROC DATASETS
LIB=WORK;
 DELETE
FIS;
 RUN;



PROC
SQL;

 CREATE TABLE vst
AS
 SELECT
V.CASE_NUM,
 V.STAFF_ID,

 V.CHARGE,

 V.STATUS,
 V.HOURS,

 V.VDATE,

 V.EDATE,

 V.VPURP,

 V.PAY0,

 V.TDSEQNO,

 V.AGCY_CDE,

 V.PAT_CTR,

 V.STF_CTR,

 V.INV_NO,

 V.BILLWO,

 V.COB,

 V.DEN_CDE,

 V.RSN_CDE,

 F.UNBWO,

 V.HR,

 V.MIN

FROM VHS V LEFT JOIN UNBWOS
F
ON V.CASE_NUM=F.CASE_NUM
AND
   V.STAFF_ID=F.STAFF_ID
AND
   V.VDATE=F.VDATE
AND
   V.TDSEQNO=F.TDSEQNO;
 CREATE TABLE vst
AS
 SELECT
F.PAY1,
 F.PAY2,

 F.PAY3,

 F.PAY4,

 F.PAY5,

 V.CASE_NUM,

 V.STAFF_ID,

 V.CHARGE,

 V.STATUS,

 V.HOURS,

 V.VDATE,

 V.EDATE,

 V.VPURP,

 V.PAY0,

 V.TDSEQNO,

 V.AGCY_CDE,

 V.PAT_CTR,

 V.STF_CTR,

 V.INV_NO,

 V.BILLWO,

 V.COB,

 V.DEN_CDE,
 V.RSN_CDE,

 V.UNBWO,

 V.HR,

 V.MIN

FROM vst V LEFT JOIN TFIS
F
ON
V.CASE_NUM=F.CASE_NUM;

QUIT;



PROC DATASETS
LIB=WORK;
 DELETE TFIS VHS UNBWO
UNBWOS;
RUN;



PROC
SQL;

 CREATE TABLE vst
AS
 SELECT
V.PAY1,
 V.PAY2,

 V.PAY3,

 V.PAY4,

 V.PAY5,

 V.CASE_NUM,

 V.STAFF_ID,

 V.CHARGE,
 V.STATUS,

 V.HOURS,

 V.VDATE,

 V.EDATE,

 V.VPURP,

 V.PAY0,

 V.TDSEQNO,

 V.AGCY_CDE,

 V.PAT_CTR,

 V.STF_CTR,

 V.INV_NO,

 V.BILLWO,

 V.COB,

 V.DEN_CDE,

 V.RSN_CDE,

 V.UNBWO,

 V.HR,

 V.MIN,

 A.PAYCAT

FROM vst V LEFT JOIN ARMS
A
ON
V.PAY0=A.PAYOR;

 CREATE TABLE vst
AS
 SELECT
V.PAY1,
 V.PAY2,
 V.PAY3,
 V.PAY4,
 V.PAY5,
 V.CASE_NUM,
 V.STAFF_ID,
 V.CHARGE,
 V.STATUS,
 V.HOURS,
 V.VDATE,
 V.EDATE,
 V.VPURP,
 V.PAY0,
 V.TDSEQNO,
 V.AGCY_CDE,
 V.PAT_CTR,
 V.STF_CTR,
 V.INV_NO,
 V.BILLWO,
 V.COB,
 V.DEN_CDE,
 V.RSN_CDE,
 V.UNBWO,
 V.HR,

 V.MIN,

 V.PAYCAT,

 C.MRN,

 C.ADMISSION_DATE AS
ADATE,

 C.AGEGRP,

 C.BOROUGH AS
BORO,

 C.COC,

 C.DISCHARGE_DATE AS
DDATE,

 C.DIAGNOSIS_CODE AS
DIAG,

 C.DIAGRP,

 C.DISGRP,

 C.DMS_SCORE AS
DMS,

 C.HEALTH_AREA AS
HA,

 C.ICD9_NORMAL  AS
ICD9_NR1,

 C.ICD9_NORMAL2 AS
ICD9_NR2,

 C.ICD9_NORMAL3 AS
ICD9_NR3,

 C.ICD9_NORMAL4 AS
ICD9_NR4,
 C.ICD9_NORMAL5 AS
ICD9_NR5,

 C.ICD9_NORMAL6 AS
ICD9_NR6,

 C.ICD9_NORMAL7 AS
ICD9_NR7,

 C.LANGRP,

 C.LIVES_WITH AS
LIVWITH,

 C.PAYTYP,

 C.PROGRAM AS
PGM,

 C.PRDPGM,

 C.RACE,

 C.REFERRAL_ID AS
REFERRAL,

 C.REFERRAL_STATUS_CODE AS
RSCR,

 C.GENDER AS
SEX,

 C.TEAM,

 C.REFERRAL_TYPE AS
TYPE,

 C.ZIPCODE

FROM &dbetl..case_facts
C,
     vst
V
WHERE
C.CASE_NUM=V.CASE_NUM;
```

```
QUIT;



PROC DATASETS
LIB=WORK;
 DELETE
ARMS;
RUN;




*--------------------------------------------
;
*  CREATION OF NEW SINGLE
SOURCE             ;
*  USING DATA FROM DB2
DIRECTLY              ;
*  LOAD DB2 TABLES FROM SAS
DATASET          ;
*           WORK.VST95     1.                ;

*--------------------------------------------
;
DATA
VST;

 LENGTH BU $2.
;
SET
VST;

*----------- THIS IS THE NETWORK OR NASSAU PROBLEM ----------
;
*----------- USING '9' FOR NETWORK CASES --------------------
;
IF AGCY_CDE='NET' THEN
BORO='9';


/*

IF PAYCAT NOT IN
('AAA','BCA','COL','COM','DSS','FRE','HIP','MCP','OTH',
                  'PRI','VEN','UUU','APA','VCP') THEN
DO;
```

```
                  */

IF PAYCAT = '' THEN
DO;
 PAYCAT='UUU';

 PAY0='UUU'; PAY1='UUU';
PAY2='UUU';
 PAY3='UUU'; PAY4='UUU';
PAY5='UUU';
END;



 IF PAY0='000' THEN
DO;
    PAY0='JLR';

    PAYCAT='COM';

 END;

*----------------------------------------------------------
;
*     CONTINUE TO CREATE BUSINESS
UNIT                    ;
*   01=LTC, 02=ACUTE CARE, 03=MANAGED CARE, 04=VNS CHOICE
;
*----------------------------------------------------------
;
 IF PGM IN:('V') THEN BU='04'; /* VNS CHOICE
*/
 ELSE IF PGM IN:('L','H','C','E') OR PGM='PEI' OR
PGM='GCM'
                 OR
PGM='PRC'
         THEN BU='01';   /* LONG TERM CARE
*/
 ELSE IF
BORO='9'
         THEN BU='03';   /* MANAGED CARE
*/
 ELSE IF PAY0
IN:('DSS','AAA','AAB','J13','S','APA','APB')
         THEN BU='02';   /* ACUTE CARE
*/
```

```
 ELSE IF PAY0 NOT
IN:('DSS','AAA','AAB','J13','S','U','C','APA','APB')
         THEN BU='03';   /* MANAGED CARE
*/
 ELSE IF (PAY0 =:'C' OR PAY0
=:'U')
        AND (PAY1 =:'C' OR PAY1 =:'  ' OR PAY1
=:'U')
        AND (PAY2 =:'C' OR PAY2 =:'  ' OR PAY2
=:'U')
        AND (PAY3 =:'C' OR PAY3 =:'  ' OR PAY3
=:'U')
        AND (PAY4 =:'C' OR PAY4 =:'  ' OR PAY4
=:'U')
        AND (PAY5 =:'C' OR PAY5 =:'  ' OR PAY5
=:'U')
         THEN BU='02';   /* ACUTE CARE
*/
 ELSE IF (PAY0 =:'C' OR PAY0 =:'U')
AND
         PAY1
IN:('DSS','AAA','AAB','J13','S','APA','APB')
         THEN BU='02';   /* ACUTE CARE
*/
 ELSE IF (PAY0 =:'C' OR PAY0 =:'U')
AND
         PAY1 NOT
IN:('DSS','AAA','AAB','J13','S','U','C','APA','APB')
         THEN BU='03';   /* MANAGED CARE
*/
 ELSE IF (PAY1 =:'C' OR PAY1 =:'U')
AND
         PAY2
IN:('DSS','AAA','AAB','J13','S','APA','APB')
         THEN BU='02';   /* ACUTE CARE
*/
 ELSE IF (PAY1 =:'C' OR PAY1 =:'U')
AND
         PAY2 NOT
IN:('DSS','AAA','AAB','J13','S','U','C','APA','APB')
         THEN BU='03';   /* MANAGED CARE
*/
 ELSE IF (PAY2 =:'C' OR PAY2 =:'U')
AND
         PAY3
IN:('DSS','AAA','AAB','J13','S','APA','APB')
```

```
         THEN BU='02';   /* ACUTE CARE
*/
 ELSE IF (PAY2 =:'C' OR PAY2 =:'U')
AND
         PAY3 NOT
IN:('DSS','AAA','AAB','J13','S','U','C','APA','APB')
         THEN BU='03';   /* MANAGED CARE
*/
 ELSE IF (PAY3 =:'C' OR PAY3 =:'U')
AND
         PAY4
IN:('DSS','AAA','AAB','J13','S','APA','APB')
         THEN BU='02';   /* ACUTE CARE
*/
 ELSE IF (PAY3 =:'C' OR PAY3 =:'U')
AND
         PAY4 NOT
IN:('DSS','AAA','AAB','J13','S','U','C','APA','APB')
         THEN BU='03';   /* ;MANAGED CARE
*/
 ELSE IF (PAY4 =:'C' OR PAY4 =:'U')
AND
         PAY5
IN:('DSS','AAA','AAB','J13','S','APA','APB')
         THEN BU='02';   /* ACUTE CARE
*/
 ELSE IF (PAY4 =:'C' OR PAY4 =:'U')
AND
         PAY5 NOT
IN:('DSS','AAA','AAB','J13','S','U','C','APA','APB')
         THEN BU='03';   /* MANAGED CARE
*/
 ELSE BU='02' ;          /* ACUTE CARE
*/


 *  CREATE REAL HOURS FROM CHARGE TABLE
HOURS                    *;
 NHOURS=HR+(MIN/60);

 /*

 IF NHOURS = 0 and substr(staff_id,1,1) = 'V' THEN
DO;
  V1=ROUND(MOD((HOURS*10),10),1);
  V11=INT(HOURS);

  IF V1=0 THEN
V2=.00;
  ELSE IF V1=1
THEN
  V2=.25;

  ELSE IF V1=2
THEN
  V2=.50;

  ELSE IF V1=3
THEN
  V2=.75;

  NHOURS=V11+V2;

 END;

 */



 *------------ creating STAFPER and skill dimension ----------
;
  SKILL =
SUBSTR(STAFF_ID,1,1);
  staffper =
SUBSTR(STAFF_ID,3,1);


  IF SKILL IN ('R','B','L','O','F','H','P','I','S') THEN
DO;
   /* Therapists
*/
   IF SKILL IN ('P','I','S') THEN
DO;
    IF SUBSTR(STAFF_ID,2,1) = '*' AND staffper not in ('C','P')
then
     staffper =
'*';
    IF SUBSTR(STAFF_ID,2,1) = '*' AND staffper in ('C','P')
then
     staffper =
'A';
```

```
    IF SUBSTR(STAFF_ID,2,1) ^= '*' AND staffper not in ('C','P')
then
     staffper =
'0';
    IF SUBSTR(STAFF_ID,2,1) ^= '*' AND staffper in ('C','P')
then
     staffper =
'C';
   END;



   /* Nursing
*/
   ELSE
DO;
    IF staffper NOT IN ('C','P') THEN staffper =
'0';
    ELSE staffper =
'C';
   END;



  END;

  ELSE staffper =
'/';
 *------------ unkown for company,charge_status --------------
;
  if AGCY_CDE=' ' then
AGCY_CDE='UUU';
  if status=' ' then
status='UUU';
 *------------ create Visit Date hirearchy -----------
;
 year =
year(vdate);
 month =
month(vdate);
 *------------ initial unbill write off --------
;
 if unbwo = . then unbwo =
0;
 *------------ GET RID OF SOME JUNK ------------
;
```

```
 DROP HOURS PAY1 PAY2 PAY3 PAY4 PAY5
;
 RUN;


 **filename vstcyr 'c:\----;
data
_null_;

set  VST(where=(year(vdate)=&cfyr))             end=EFIEOD;

%let _EFIERR_ =
0;
%let _EFIREC_ =
0;
file vstcyr delimiter=',' DSD
LRECL=310;
   format BU $2.
;
   format CASE_NUM 11.
;
   format STAFF_ID $5.
;
   format CHARGE 9.2
;
   format STATUS $3.
;
   format VDATE MMDDYY10.
;
   format EDATE MMDDYY10.
;
   format VPURP $2.
;
   format PAY0 $3.
;
   format TDSEQNO 6.
;
   format AGCY_CDE $3.
;
   format PAT_CTR $5.
;
   format STF_CTR $5.
;
   format PAYCAT $3.
;
   format ADATE MMDDYY10.
;
```

```
   format AGEGRP best12.
;
   format BORO $1.
;
   format COC $5.
;
   format DDATE MMDDYY10.
;
   format DIAG best12.
;
   format DIAGRP best12.
;
   format DISGRP best12.
;
   format HA $4.
;
   format ICD9_NR1 $5.
;
   format ICD9_NR2 $5.
;
   format ICD9_NR3 $5.
;
   format ICD9_NR4 $5.
;
   format ICD9_NR5 $5.
;
   format ICD9_NR6 $5.
;
   format ICD9_NR7 $5.
;
   format LANGRP best12.
;
   format LIVWITH $1.
;
   format PAYTYP best12.
;
   format PGM $3.
;
   format PRDPGM $3.
;
   format RACE $1.
;
   format REFERRAL $6.
;
   format RSCR $1.
;
```

```
   format SEX $1.
;
   format TEAM $2.
;
   format TYPE $1.
;
   format ZIPCODE best12.
;
   format NHOURS best12.
;
   format INV_NO $8.
;
   format SKILL $5.
;
   format staffper $5.
;
   format year best12.
;
   format month best12.
;
   format COB $1.
;
   format DMS best12.
;
   format DEN_CDE $5.
;
   format RSN_CDE $2.
;
if _n_ = 1
then
 do;

   put

   'BU'

   ','

   'CASE_NUM'

   ','

   'STAFF_ID'

   ','
```

```
'CHARGE'

','

'STATUS'

','

'VDATE'

','

'EDATE'

','

'VPURP'

','

'PAY0'
','
'TDSEQNO'

','

'AGCY_CDE'

','

'PAT_CTR'

','

'STF_CTR'

','

'PAYCAT'

','

'MRN'

','
```

```
'ADATE'

','

'AGEGRP'

','

'BORO'

','

'COC'

','

'DDATE'

','

'DIAG'

','

'DIAGRP'

','

'DISGRP'

','

'HA'

','

'ICD9_NR1'

','

'ICD9_NR2'

','

'ICD9_NR3'
```

```
','

'ICD9_NR4'

','

'ICD9_NR5'

','

'LANGRP'

','

'LIVWITH'

','

'PAYTYP'

','

'PGM'

','

'PRDPGM'

','

'RACE'

','

'REFERRAL'

','

'RSCR'

','

'SEX'

','
```

```
'TEAM'

','

'TYPE'

','

'ZIPCODE'

','

'NHOURS'

','

'INV_NO'

','

'SKILL'

','

'staffper'

','

'year'

','

'month'

','

'COB'

','

'ICD9_NR6'

','

'ICD9_NR7'
```

```
   ','
   'DMS'
   ','
   'DEN_CDE'
   ','
   'RSN_CDE'
   ;
 end;
 do;
   EFIOUT +
1;
   put BU $
@;
   put CASE_NUM
@;
   put STAFF_ID $
@;
   put CHARGE
@;
   put STATUS $
@;
   put VDATE
@;
   put EDATE
@;
   put VPURP $
@;
   put PAY0 $
@;
   put TDSEQNO
@;
   put AGCY_CDE $
@;
   put PAT_CTR $
@;
   put STF_CTR $
@;
```

```
   put PAYCAT $
@;
   put MRN
@;
   put ADATE
@;
   put AGEGRP
@;
   put BORO $
@;
   put COC $
@;
   put DDATE
@;
   put DIAG
@;
   put DIAGRP
@;
   put DISGRP
@;
   put HA $
@;
   put ICD9_NR1 $
@;
   put ICD9_NR2 $
@;
   put ICD9_NR3 $
@;
   put ICD9_NR4 $
@;
   put ICD9_NR5 $
@;
   put LANGRP
@;
   put LIVWITH $
@;
   put PAYTYP
@;
   put PGM $
@;
   put PRDPGM $
@;
   put RACE $
@;
   put REFERRAL $
@;
```

```
   put RSCR $
@;
   put SEX $
@;
   put TEAM $
@;
   put TYPE $
@;
   put ZIPCODE
@;
   put NHOURS
@;
   put INV_NO $
@;
   put SKILL $
@;
   put staffper $
@;
   put year
@;
   put month
@;
   put COB
@;
   put ICD9_NR6 $
@;
   put ICD9_NR7 $
@;
   put DMS
@;
   put DEN_CDE $
@;
   put RSN_CDE
$;
   ;

 end;

if _ERROR_ then call
symput('_EFIERR_',1);
If EFIEOD
then
   call
symput('_EFIREC_',EFIOUT);
run;
```

```
data
_null_;

set  VST(where=(year(vdate)=&pfyr))             end=EFIEOD;

%let _EFIERR_ =
0;
%let _EFIREC_ =
0;
file vstpyr delimiter=',' DSD
LRECL=310;
   format BU $2.
;
   format CASE_NUM 11.
;
   format STAFF_ID $5.
;
   format CHARGE 9.2
;
   format STATUS $3.
;
   format VDATE MMDDYY10.
;
   format EDATE MMDDYY10.
;
   format VPURP $2.
;
   format PAY0 $3.
;
   format TDSEQNO 6.
;
   format AGCY_CDE $3.
;
   format PAT_CTR $5.
;
   format STF_CTR $5.
;
   format PAYCAT $3.
;
   format MRN 11.
;
   format ADATE MMDDYY10.
;
   format AGEGRP best12.
;
```

```
   format BORO $1.
;
   format COC $5.
;
   format DDATE MMDDYY10.
;
   format DIAG best12.
;
   format DIAGRP best12.
;
   format DISGRP best12.
;
   format HA $4.
;
   format ICD9_NR1 $5.
;
   format ICD9_NR2 $5.
;
   format ICD9_NR3 $5.
;
   format ICD9_NR4 $5.
;
   format ICD9_NR5 $5.
;
   format LANGRP best12.
;
   format LIVWITH $1.
;
   format PAYTYP best12.
;
   format PGM $3.
;
   format PRDPGM $3.
;
   format RACE $1.
;
   format REFERRAL $6.
;
   format RSCR $1.
;
   format SEX $1.
;
   format TEAM $2.
;
   format TYPE $1.
;
```

```
   format ZIPCODE best12.
;
   format NHOURS best12.
;
   format INV_NO $8.
;
   format SKILL $5.
;
   format staffper $5.
;
   format year best12.
;
   format month best12.
;
   format COB $1.
;
   format ICD9_NR6 $5.
;
   format ICD9_NR7 $5.
;
   format DMS best12.
;
   format DEN_CDE $5.
;
   format RSN_CDE $2.
;
if _n_ = 1
then
 do;

   put

   'BU'

   ','

   'CASE_NUM'

   ','

   'STAFF_ID'

   ','

   'CHARGE'
```

```
','

'STATUS'

','

'VDATE'

','

'EDATE'

','
'VPURP'
','

'PAY0'

','

'TDSEQNO'

','

'AGCY_CDE'

','

'PAT_CTR'

','

'STF_CTR'

','

'PAYCAT'

','

'MRN'

','

'ADATE'
```

```
','

'AGEGRP'

','

'BORO'

','

'COC'

','

'DDATE'

','

'DIAG'

','

'DIAGRP'

','

'DISGRP'

','

'HA'

','

'ICD9_NR1'

','

'ICD9_NR2'

','

'ICD9_NR3'

','
```

```
   'ICD9_NR4'

   ','

   'ICD9_NR5'

   ','

   'LANGRP'

   ','

   'LIVWITH'

   ','

   'PAYTYP'

   ','

   'PGM'

   ','

   'PRDPGM'

   ','

 %
```

SAS MAINFRAME JCL

```
//TAPPLBTB JOB
(00,21000),'D.D',CLASS=C,MSGCLASS=T,
//            NOTIFY=&SYSUID,REGION=9M
RESTART=STEP020R
//**-DESCRIPTION: -------------------- -------------------------------
**
//** MAPS GHI COUNTY CODE TO COUNTY NAME IN A FORMAT FILE ------------
**
//**---------------------------------- -------------------------------
**
//CA07RMS  EXEC
UCC11RMS,TYPRUN=F
//**-----------------------------------------------------------------
**
//STEP010N EXEC
SAS
//SYSOUT   DD
```

```
SYSOUT=*
//SYSPRINT DD
SYSOUT=*                                         COPIES=2
//WORK     DD
UNIT=SYSDA,SPACE=(CYL,(250,250),RLSE)
//INFLEA   DD
DSN=PCMK.B4.ZIPCNTY.MASTER.PS,DISP=SHR,BUFNO=100
//OUTFLEA  DD
DSN=TCMK.B4.TAPPLBT4.ID6274.ZIPCNTY.PS,
//            UNIT=SYSDA,SPACE=(CYL,(50,10),RLSE),BUFNO=50,

//            DISP=(,CATLG,DELETE)

//*           DCB=(RECFM=FB,LRECL=200),DISP=(,CATLG,DELETE)

//SYSIN    DD
*
   DATA
WORK.A;
   INFILE INFLEA
END=LAST;
   RETAIN FMTNAME
'$ZIPCNTY';
   INPUT @001
START   $05.
         @007
CNTY    $02.
         @012
NAME    $18.;
   LABEL=CNTY||'
'||NAME;
   OUTPUT;

   IF LAST THEN DO;
START='OTHER';
                    LABEL='-- UNKNOW COUNTY'; OUTPUT;
END;
   RUN;

   PROC SORT DATA=WORK.A
NODUPKEY;
   BY
START;
   RUN;

   PROC FORMAT CNTLIN=WORK.A LIBRARY=OUTFLEA;
```

```
options noxwait xsync;
*PATH = '\\fpsm0102\is\operations\sas download files\USR6102A.XLS';
X 'DEL &EXCEL.USR6102A.XLS';

PROC DBLOAD DBMS=XLS DATA=USR6102A;
*PATH = '\\fpsm0102\is\operations\sas download files\USR6102A.XLS';
PATH = '&EXCEL.USR6102A.XLS';
LABEL;
PUTNAMES=YES;
LIMIT=0;
RESET ALL;
LOAD;
RUN;
*/
PROC EXPORT data=USR6102A
OUTFILE="&excel.USR6102A.CSV" DBMS=DLM REPLACE;
DELIMITER=',';
RUN;
```

SAS ODBC CONNECITIVITY TO ORACLE

```
/*** successful connection to Queens ***/
proc sql;
connect to ODBC(dsn=Queens user=sterlinb password=Oracle200);
create table Queens1 as
select * from connection to ODBC
(select * from ud_master.proc
where rownum < 10;);
disconnect from odbc;
quit;

-------------------------
%let uid=;
%let pwd=;
*****/
%let dsn=p5555;
%let db=FINANCE_PDE_WORKDB;

libname tera TERADATA dbprompt=no database=FINANCE_PDE_WORKDB
tdpid=prod user=&uid password=&pwd;

proc sql;
connect to odbc(dsn = &dsn uid = &uid password = &pwd );
Select * from connection to odbc
();
```

```
QUIT;
```

SQL SERVER FUNCTION

```
/*** THIS FUNCTION Returns a table of image id's, pathways and fully
qualified image files and number of expected images ***/
CREATE FUNCTION ImageReturn(@IMAGE1 VARCHAR(50))
returns @Images TABLE(Directory VARCHAR(50),Image_id VARCHAR(50), FullP
ath VARCHAR(50),NumberOfImages INT)
as begin
--- @Images TABLE(Directory VARCHAR(50),Image_id VARCHAR(50), FullPath
VARCHAR(50),NumberOfImages INT);
DECLARE
/* build the directory */
@DIR1a char(1), @DIR1b char(1),
@DIR2a char(1), @DIR2b char(1),
@DIR3a char(1), @DIR3b char(1),
@Directory VARCHAR(50),
@FullPath VARCHAR(50),
@Type CHAR(3),
@i INT,
@Next INT,
@LEN INT,
--- @IMAGE1 VARCHAR(50),
@NbrImages INT,
@NUMBER1 INT,
@NextImageNum INT,
@NextImage VARCHAR(50);
/*-- begin processing -*/
--- (used for testing) SET @NbrImages = 9999;
--- (used for testing) set @IMAGE1 = 'QNW-0000809994'; /*input image
number note all 10 digits are the number for incrementing*/
--- (used for testing) set @Type = 'JPG';
/*-- USE MED_MAN_ELMDB --*/
/*- are any images in image table? if not send back with message of
'Image Not Found', else if 1 image send back single image info -*/
select
@NbrImages = s.number_of_images,
@Type = i.type
from
MED_MAN_ELMDB.dbo.images i join MED_MAN_ELMDB.dbo.studies s
on i.study_id = s.study_id
and i.patient_id = s.patient_id
where i.image_id = @IMAGE1;
if(@Type = 'JPG') set @Type = 'jpg';
if(@Type = 'TIF') set @Type = 'tif';
/*- if no images found exit with message in output -*/
if(@Type is null) begin goto NoneFound end;
set @DIR1a = SUBSTRING(@IMAGE1,5,1);
set @DIR1b = SUBSTRING(@IMAGE1,6,1);
set @DIR2a = SUBSTRING(@IMAGE1,7,1);
set @DIR2b = SUBSTRING(@IMAGE1,8,1);
set @DIR3a = SUBSTRING(@IMAGE1,9,1);
set @DIR3b = SUBSTRING(@IMAGE1,10,1);
set @Directory = '/dataextracts/images/' + @DIR1a + @DIR1b + '/' + @DIR
3a + @DIR3b + '/' + @DIR2a + @DIR2b + '/';
set @FullPath = @Directory + @IMAGE1 + '.' + @Type;
/*- grab first image and put into return table -*/
INSERT INTO @Images VALUES (@Directory,@IMAGE1,@FullPath,@NbrImages);
if(@NbrImages = 1) goto OnlyOne; /*- if only one image then we are
done. Exit program -*/
set @DIR1a = NULL;
set @DIR1b = NULL;
set @DIR2a = NULL;
set @DIR2b = NULL;
set @DIR3a = NULL;
set @DIR3b = NULL;
set @Directory = NULL;
set @FullPath = NULL;
/*- then grab the rest of the images -*/
set @NUMBER1 = SUBSTRING(@IMAGE1,5,10);
set @NextImageNum = (SUBSTRING(@IMAGE1,5,10) + 4);
/*-there is no 0000 only 0001 -*/
--- if(@NextImageNum % 10 = 0) set @NextImageNum = @NextImageNum + 1;
/*- This is the next image after the initial image -*/
/*- begin while loop processing for all image -*/
set @i = 1;
set @Next = @NextImageNum;
while @i < @NbrImages
begin
/*-there is no 0000 only 0001 -*/
/* if substring(@IMAGE1,11,4) + @NbrImages) > 10000 then doing a
rollover into next directory */
----if(@Next % 10 = 0 AND (substring(@IMAGE1,11,4) + @NbrImages) >
10000) set @Next = @Next + 1;
set @LEN = LEN(CAST(@Next AS VARCHAR(11)));
if(@LEN = 1)
set @NextImage = 'QNW-000000000' + CAST(@Next as varchar(10));
if(@LEN = 2)
set @NextImage = 'QNW-00000000' + CAST(@Next as varchar(10));
if(@LEN = 3)
set @NextImage = 'QNW-0000000' + CAST(@Next as varchar(10));
if(@LEN = 4)
set @NextImage = 'QNW-000000' + CAST(@Next as varchar(10));
if(@LEN = 5)
set @NextImage = 'QNW-00000' + CAST(@Next as varchar(10));
if(@LEN = 6)
set @NextImage = 'QNW-0000' + CAST(@Next as varchar(10));
if(@LEN = 7)
set @NextImage = 'QNW-000' + CAST(@Next as varchar(10));
if(@LEN = 8)
set @NextImage = 'QNW-00' + CAST(@Next as varchar(10));
if(@LEN = 9)
set @NextImage = 'QNW-0' + CAST(@Next as varchar(10));
if(@LEN = 10)
set @NextImage = 'QNW-' + CAST(@Next as varchar(10));
/* create directory */
set @DIR1a = SUBSTRING(@NextImage,5,1);
set @DIR1b = SUBSTRING(@NextImage,6,1);
set @DIR2a = SUBSTRING(@NextImage,7,1);
set @DIR2b = SUBSTRING(@NextImage,8,1);
set @DIR3a = SUBSTRING(@NextImage,9,1);
set @DIR3b = SUBSTRING(@NextImage,10,1);
set @Directory = '/dataextracts/images/' + @DIR1a + @DIR1b + '/' + @DIR
3a + @DIR3b + '/' + @DIR2a + @DIR2b + '/';
set @FullPath = @Directory + @NextImage + '.' + @Type;
INSERT INTO @Images VALUES (@Directory,@NextImage,@FullPath,@NbrImages)
;
set @Next = @Next + 1;
set @i = @i + 1;
end
goto done;
/*-if image is not found in image table then exit function with message
-*/
NoneFound:
INSERT INTO @Images VALUES ('Image Not Found',@IMAGE1,'Image Not
Found',0);
/*- If only one image then exit. No further processing required -*/
OnlyOne:
done:
RETURN;
END
GO
---- drop function ImageReturn;
--- SELECT * FROM ImageReturn(@IMAGE1 VARCHAR(50);
-- SELECT * FROM ImageReturn('QNW-0000019997');
--- SELECT * FROM ImageReturn('QNW-0000079845');
--- select * from ImageReturn('QNW-0000019470');
---- select * from ImageReturn('QNW-0000099994');
--- select * from ImageReturn('QNW-0000169986');
--- select * from ImageReturn('QNW-0100219981')
---- SELECT * FROM ImageReturn('QNW-0000009129');
/**
select top 10000
*
from
dbo.images i join dbo.studies s
on i.study_id = s.study_id
and i.patient_id = s.patient_id
where number_of_images > 45
and image_id >= 'QNW-0000019520'
order by image_id
select
*
from
dbo.images i join dbo.studies s
on i.study_id = s.study_id
and i.patient_id = s.patient_id
where
image_id >= 'QNW-0000009129'
order by image_id;
select top 10000
*
from
dbo.images i join dbo.studies s
on i.study_id = s.study_id
and i.patient_id = s.patient_id
where number_of_images > 40
```

```
and substring(image_id,11,4) >= '9990'
order by image_id
*/
```

Loading information into webpage from SQL Server @ NYC HHC

Running a stored procedure in JAVA

```
----------------------------------------------------
   try{

        // execute SP that requires two string arguments
         PreparedStatement ps = conn.prepareStatement("exec
storedProcedure ?,?");
         ps.setEscapeProcessing(true);

         ps.setString(1, "str1"); //set the parameters to be
run
         ps.setString(2, "str2");

         ResultSet rs = ps.executeQuery();

         while (rs.next()) {
             String strVal = rs.getString(1);
             // do stuff    } //end while

       /* If you get multiple results back from you SP, you can
iterate through them using:

         ps.getMoreResults();
         rs = ps.getResultSet();

       while (rs.next()) {
           String strVal = rs.getString(1);
           String  relatedConcept = rs.getString(1);
           int   shortestDistance = rs.getInt(2);
           float weightedDistance = rs.getFloat(3);
            ...
      }  */

   rs.close();
   ps.close();
   } //end try
   catch (Exception e) {
       e.printStackTrace();
   }//end catch
```

```
    }//end main method
};//end java class
```

```
>>> Bonnie Sterling 7/29/2014 2:25 PM >>>
TESTING STORED PROCEDURES IN ORACLE FOR LOADING WEB APPLICATIONS

SET SERVEROUTPUT ON SIZE 1000000;

Set Echo On

create or replace
PROCEDURE   IMM_TEST (
   PRC          OUT SYS_REFCURSOR,
   P_EMPI       IN     VARCHAR2,
   P_NETWORK    IN     VARCHAR2,
   P_FACILITY   IN     VARCHAR2,
   P_Mrn        In     Varchar2,
   P_type      IN     varchar2)
IS
BEGIN
   OPEN PRC FOR
      SELECT *
        FROM DATACONVERSION.IMMUNIZATION tbl
       WHERE tbl.EMPI = P_EMPI
        AND tbl.NETWORK = P_NETWORK
        AND tbl.FACILITY_NAME = P_FACILITY
        And Tbl.Medical_Record_Number = P_Mrn
        And Tbl.Immunization_Type = P_Type
        order by tbl.empi,tbl.immunization_date;
END IMM_TEST;

SET SERVEROUTPUT ON SIZE 1000000;

VARIABLE X refcursor

Exec Imm_Test(:X,'25812311','Queens Health','Queens
Hospital','4002842','DTP');
print X

EXEC IMM_TEST(:X,'25812311','Queens Health','Elmhurst
Hospital','4002848','MENINGITIS');
print X
```