BRM Database Migration Progress Notes

LAST UPDATED: ARMarshall, ARM LLC - 20230131

GENERAL COMMENTS ABOUT THE UPGRADE SCRIPT(S)

```
PON_CONCAT calls might be better formed using LISTAGG available in Oracle 12C+

see this>   [Oracle LISTAGG documentation]
(https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions089.htm#SQLRF3003
0 "Oracle LISTAGG documentation")


<span style="font-weight:500; font-size:12px;">
         <p> SELECT LISTAGG('T.'|| DICT.COLUMN_NAME ,',')
           WITHIN GROUP (ORDER BY DICT.COLUMN_ID) "DELIM_COL_LIST"
                                                 FROM USER_TAB_COLS DICT
                                                 GROUP BY DICT.TABLE_NAME
                                                 HAVING DICT.TABLE_NAME = 'BRIDGE'
                                                 ORDER BY DICT.TABLE_NAME
                                                 </p>
</span>
```

TO EXPORT THE DATABASE

```
        C:\oracle\instantclient_21_7>expdp
kdot_blp/Einstein345@localhost:1521/xepdb1 directory=KDOT_DP_DIR
dumpfile=KDOT_BLP20221220SQL.dmp schemas=('KDOT_BLP')
logfile=KDOT_BLP20221220SQL.log
```

TO SEE THE DDL FROM DATAPUMP DMP FILE

Burleson Consulting says: **Show SQL From Dump FIle**

GOTCHAS

> > > ***A COMMON PROBLEM WITH ALL THESE SCRIPTS IS THAT THE SQL DDL and DML CODE ASSUMES THE PERSON RUNNING THEM HAS DBA PRIVILEGES AND CAN ISSUE THE ALTER SYSTEM COMMAND. CONSEQUENTLY, ANY ALTER SYSTEM STATEMENTS HAVE TO BE COMMENTED OUT BEFORE RUNNING ANY OF THEM***

PROGRESS NOTES

2022-08-17

-- ARMarshall, ARM LLC 20220817 - named the block where the globals are set to SET_GLOBALS_FOR_RUN
-- ARMarshall, ARM LLC 20220817 - added PURGE to all DROP TABLE instances, including those constructed in loops.

-- ARMarshall, ARM LLC 20220817 - all these calls to PON_CONCAT use the old CURSOR loop way - probably should use 12C LISTAGG or possibly XMLAGG functions if the string result is > 4000 characters -- leaving them alone.

```
            ...
            ...

            SELECT PON_CONCAT(CURSOR(SELECT 'T.' || DICT.COL_NAME || ' = S.' ||
DICT.COL_NAME
                                   FROM PON_DICT DICT
                                   WHERE DICT.TABLE_NAME = X.TABLE_NAME
                                   AND ACTIVE_PK = 1
                                   ORDER BY DICT.COL_ORDER)
                             ,' AND ')
            INTO V_MATCH_COLS
            FROM DUAL;
            ...
            ...
```

2022-08-18

-- 20220818 - ARMarshall, ARM LLC - initial review, added globals to preserve datadict and paramtrs tables for KDOT LBIS 9 (BrM 5.3) database
-- 20220818 - ARMarshall, ARM LLC - corrected minor typos as encountered

---

2022-12-20

-- 202221220 - code example for BLP tables that are backed up SEPARATELY in addition to whatever the main script does

These tables are:

- COPTIONS
- DATADICT
- PON_COPTIONS
- PARAMTRS
- KDOTBLP_ATTRIBUTE_DESCRIPTOR

```
        BEGIN
            -- if the setting is true, make a backup of the table before doing
anything else!
                IF GET_VAR('SAVEAGENCYSYSTABLES') = 1 THEN
                    DECLARE

                    V_X VARCHAR2(64):=RAWTOHEX(SYS_GUID());

                    BEGIN

                        BEGIN
                            EXECUTE IMMEDIATE q'[DROP TABLE PARAMTRS]'||V_X || q'[
PURGE]';
                        EXCEPTION
                                WHEN OTHERS THEN NULL;  -- do not report if the
table does not exist
                        END;

                        BEGIN

                            EXECUTE IMMEDIATE q'[CREATE TABLE PARAMTRS]'||V_X|| q'[
AS SELECT parms.* FROM PARAMTRS psrc ORDER BY parms.TABLE_NAME, parms.FIELD_NAME,
parms.PARMVALUE]' ;

                        EXCEPTION
                            WHEN OTHERS THEN RAISE;
                        END;
                    END;
                END IF;


        // do stuff here..
        ...
        ...
        ...



        EXCEPTION
            WHEN OTHERS THEN RAISE;
            END;
        /
```

-- 20221220 - added option to print cursor detail for debugging purposes which should be set to 0 normally

-- IN PROCEDURE REV and anywhere else, examine this to determine if SQL strings or other materials should be echoed to the log -- set it at the top in this case to TRUE so it prints

```
         INSERT INTO PON_GLOB_VAR (VARI, VAR_VALUE) VALUES ('PRINT_DETAILS', 1);
```

-- where REV is invoked*

```
        IF GET_VAR('PRINT_DETAILS') = 1 THEN
            REV(V_CUR,1);
        ELSE
            REV(V_CUR);
        END IF;
```

2022-12-21

-- ARMarshall, ARM LLC 20221221 - Status - table VALIDATION_WARNINGS is missing, - tables PON_APP_ROLES and PON_PROGRAM have an extra column - table USERINSP is nowhere to be found but there is an old one we can rename and quantify.

after imports, it may be necessary to recompile the KDOT_BLP schema wholesale, like this (connected with DBA privileges e.g. SYS):

```
            SQL>CONNECT SYS/xxxxx@localhost:1521/xepdb1 as SYSDBA;
            SQL>EXECUTE UTL_RECOMP.RECOMP_PARALLEL(4,'KDOT_BLP');
```

2022-12-27

Created a script to generate table ***VALIDATION_WARNING***,

Create statement is derived from this section of the script:

```
            /*
            Insert into PON_DICT
(TABLE_NAME,PK,REQUIRED,COL_ORDER,COL_NAME,DATA_TYPE,LENGTH,SCALE,DEF_VALUE,ID,FOR
CE_DEF) values
('VALIDATION_WARNING',3,1,1,'VALIDATION_WARNING_GD','VARCHAR',32,null,'REPLACE(NEW
ID(), ''-'','''')',0,1);
            Insert into PON_DICT
(TABLE_NAME,PK,REQUIRED,COL_ORDER,COL_NAME,DATA_TYPE,LENGTH,SCALE,DEF_VALUE,ID,FOR
CE_DEF) values
('VALIDATION_WARNING',0,0,2,'TABLE_NAME','VARCHAR',100,null,null,0,1);
            Insert into PON_DICT
(TABLE_NAME,PK,REQUIRED,COL_ORDER,COL_NAME,DATA_TYPE,LENGTH,SCALE,DEF_VALUE,ID,FOR
CE_DEF) values
('VALIDATION_WARNING',0,0,3,'COL_NAME','VARCHAR',100,null,null,0,1);
            Insert into PON_DICT
(TABLE_NAME,PK,REQUIRED,COL_ORDER,COL_NAME,DATA_TYPE,LENGTH,SCALE,DEF_VALUE,ID,FOR
CE_DEF) values ('VALIDATION_WARNING',0,0,4,'ROW_PK','VARCHAR',32,null,null,0,1);
            Insert into PON_DICT
```

```
(TABLE_NAME,PK,REQUIRED,COL_ORDER,COL_NAME,DATA_TYPE,LENGTH,SCALE,DEF_VALUE,ID,FOR
CE_DEF) values
('VALIDATION_WARNING',0,0,5,'INPUT_VALUE','VARCHAR',3999,null,null,0,1);
            Insert into PON_DICT
(TABLE_NAME,PK,REQUIRED,COL_ORDER,COL_NAME,DATA_TYPE,LENGTH,SCALE,DEF_VALUE,ID,FOR
CE_DEF) values
('VALIDATION_WARNING',0,0,6,'PON_APP_USERS_GD','VARCHAR',32,null,null,0,1);
            Insert into PON_DICT
(TABLE_NAME,PK,REQUIRED,COL_ORDER,COL_NAME,DATA_TYPE,LENGTH,SCALE,DEF_VALUE,ID,FOR
CE_DEF) values
('VALIDATION_WARNING',0,0,7,'VALIDATION_DATE','DATETIME',null,null,null,0,1);
            Insert into PON_DICT
(TABLE_NAME,PK,REQUIRED,COL_ORDER,COL_NAME,DATA_TYPE,LENGTH,SCALE,DEF_VALUE,ID,FOR
CE_DEF) values
('VALIDATION_WARNING',0,0,8,'VALIDATION_SOURCE_GENERIC','VARCHAR',100,null,null,0,
1);
            Insert into PON_DICT
(TABLE_NAME,PK,REQUIRED,COL_ORDER,COL_NAME,DATA_TYPE,LENGTH,SCALE,DEF_VALUE,ID,FOR
CE_DEF) values
('VALIDATION_WARNING',0,0,9,'VALIDATION_SOURCE_SPECIFIC','VARCHAR',1000,null,null,
0,1);
            Insert into PON_DICT
(TABLE_NAME,PK,REQUIRED,COL_ORDER,COL_NAME,DATA_TYPE,LENGTH,SCALE,DEF_VALUE,ID,FOR
CE_DEF) values
('VALIDATION_WARNING',0,0,10,'VALIDATION_MESSAGE','VARCHAR',3999,null,null,0,1);
            */
```

which translates to this:

```
            DROP TABLE VALIDATION_WARNING;

            CREATE TABLE VALIDATION_WARNING
            (
            VALIDATION_WARNING_GD VARCHAR2(32) NOT NULL,
            TABLE_NAME VARCHAR2(100) NOT NULL,
            COL_NAME VARCHAR2(100) NOT NULL,
            ROW_PK VARCHAR2(32) NOT NULL,
            INPUT_VALUE VARCHAR2(3999 CHAR) NOT NULL,
            PON_APP_USERS_GD VARCHAR2(32) NOT NULL,
            VALIDATION_DATE DATE NOT NULL,
            VALIDATION_SOURCE_GENERIC VARCHAR2(100) NOT NULL,
            VALIDATION_SOURCE_SPECIFIC VARCHAR2(1000) NOT NULL,
            VALIDATION_MESSAGE VARCHAR2(3999 CHAR) NOT NULL);
```

and these constraints on table ***VALIDATION_WARNING*** are also taken fronm the upgrade script:

```
            ALTER TABLE VALIDATION_WARNING ADD CONSTRAINT PK_VALIDATION_WARNING
PRIMARY KEY ("VALIDATION_WARNING_GD") using index tablespace PONT_TBL;
```

```
            ALTER TABLE VALIDATION_WARNING ADD CONSTRAINT
FK_VALIDATION_WARNING_USERS FOREIGN KEY (PON_APP_USERS_GD) REFERENCES
PON_APP_USERS (PON_APP_USERS_GD) ON DELETE CASCADE;
```

-- added the table PON_MOBILE_ERRORS, using these statenents from the script

```
            Insert into PON_DICT
(TABLE_NAME,PK,REQUIRED,COL_ORDER,COL_NAME,DATA_TYPE,LENGTH,SCALE,DEF_VALUE,ID,FOR
CE_DEF) values
('PON_MOBILE_ERRORS',3,1,1,'PON_MOBILE_ERRORS_GD','VARCHAR',32,null,'REPLACE(NEWID
(), ''-'','''')',0,1);
            Insert into PON_DICT
(TABLE_NAME,PK,REQUIRED,COL_ORDER,COL_NAME,DATA_TYPE,LENGTH,SCALE,DEF_VALUE,ID,FOR
CE_DEF) values
('PON_MOBILE_ERRORS',0,1,2,'PON_APP_USERS_GD','VARCHAR',32,null,null,0,1);
            Insert into PON_DICT
(TABLE_NAME,PK,REQUIRED,COL_ORDER,COL_NAME,DATA_TYPE,LENGTH,SCALE,DEF_VALUE,ID,FOR
CE_DEF) values
('PON_MOBILE_ERRORS',0,1,3,'DATE_REPORTED','DATETIME',null,null,null,0,1);
            Insert into PON_DICT
(TABLE_NAME,PK,REQUIRED,COL_ORDER,COL_NAME,DATA_TYPE,LENGTH,SCALE,DEF_VALUE,ID,FOR
CE_DEF) values ('PON_MOBILE_ERRORS',0,1,4,'DATA','VARCHAR',4000,null,null,0,1);
            Insert into PON_DICT
(TABLE_NAME,PK,REQUIRED,COL_ORDER,COL_NAME,DATA_TYPE,LENGTH,SCALE,DEF_VALUE,ID,FOR
CE_DEF) values
('PON_MOBILE_ERRORS',0,1,5,'ORDER_NUM','SMALLINT',null,null,null,0,1);
```

and the create statement looks like:

```
            CONNECT KDOT_BLP/********************@localhost:1521/xepdb1 AS NORMAL

            SET SERVEROUTPUT ON
            spool C:\git\repos\KDOT-LBIS-BRM-
MIGRATION\stages\6.0\out\Create_Table_PON_MOBILE_ERRORS-20221227T1646.log

            drop table PON_MOBILE_ERRORS purge;

            create table PON_MOBILE_ERRORS
            ( PON_MOBILE_ERRORS_GD VARCHAR2(32) DEFAULT ('REPLACE(NEWID(), ''-
'')')  NOT NULL,
            PON_APP_USERS_GD VARCHAR2(32) NOT NULL ,
            DATE_REPORTED DATE DEFAULT (SYSDATE)  NOT NULL,
            DATA VARCHAR2(4000 CHAR),
            ORDER_NUM INTEGER
            );
            ALTER TABLE PON_MOBILE_ERRORS ADD CONSTRAINT PK_PON_MOBILE_ERRORS
PRIMARY KEY ("PON_MOBILE_ERRORS_GD") using index tablespace PONT_TBL;
```

- Added the column ACTIVE_DIRECTORY_ROLE (CHAR(1)) to PON_APP_USERS_ROLES
- Recreated USERINSP and USERBRDG and USERRWAY as

```
        SELECT * FROM KDOTBLP_BRIDGE, KDOTBLP_INSPECTIONS, KDOTBLP_ROADWAYS,
```

this is done to ensure USERSTRUNIT had an entry for every bridge **(SQL manipulation)**
Added KDOT agency tables KDOTBLP_BRIDGE, KDOTBLP_INSPECTIONS, KDOTBLP_ROADWAYS, KDOTBLP_LOAD_RATINGS
to the upgrade script near line

- Added a script GenUserTables to make dummy copies of USERBRGD and USER INSP so the upgrade script won't choke Discovered some orphan records in KDOTBLP_INSPECTIONS that needed to be cleaned out

```
            -- fixup for KDOTBLP_INSPECTIONS - these FK were not being enforced as of 12/27/2022 - must be to avoid ORPHANS in KDOTBLP_INSPECTIONS when a parent INSPEVNT record is removed
            alter table KDOTBLP_INSPECTIONS
            drop constraint FK_KDOT_INSP_INSPEVNT_GD;

            alter table KDOTBLP_INSPECTIONS
            add constraint FK_KDOT_INSP_INSPEVNT_GD foreign key (INSPEVNT_GD)
            references INSPEVNT (INSPEVNT_GD) on delete cascade
            enable
            novalidate;
```

The bridges involved (and the inspections INSPKEY values along with the # of duplicates) were

BRKEY	INSPKEY	MODTIME	HITS
000000000630360	GPIO	2/10/2020 12:27:00 PM	2
000000000630370	VAXU	2/10/2020 12:27:00 PM	2
000000000890625	RHOA	2/17/2022 08:39:42 AM	5
414301052550076	CXJU	8/27/2021 11:49:30 AM	2
414301052550076	QLEB	2/10/2020 12:27:00 PM	2
427950895521H9C	TECQ	2/17/2022 10:33:42 AM	4
42795089552D10C	VYOV	2/17/2022 10:37:16 AM	2
430400876401010	BCAF	3/6/2020 10:55:29 AM	2
430400876401010	PRRL	2/17/2022 09:05:49 AM	3
530400870000098	RMUN	2/17/2022 09:44:25 AM	3

BRKEY	INSPKEY	MODTIME	HITS
530400870000098	UCPG	3/6/2020 01:10:21 PM	2
5304008700MCR10	EBVY	3/6/2020 02:14:01 PM	2
5304008700MCR10	EUPQ	6/28/2022 08:54:16 AM	2
5304008700MCR20	IYUL	6/28/2022 09:02:01 AM	2
5304008700MCR20	UATA	3/9/2020 12:16:44 PM	2
5304008700MCR30	IMKC	8/16/2021 09:47:27 AM	3
5304008700MCR40	NIIM	8/16/2021 09:48:23 AM	4
5304008700MCR40	NIIM	8/16/2021 09:48:23 AM	4

As of 20221227, there were no orphan records in USERRWAY or USERSTRUNIT

(later update work revealed that Referential Integrit Cascading Deletes betwen main tables and agency tables was hosed up by the upgrade scripts)

2022-12-28

-- Made sure the BrM-specific table PON_PROGRAM has the column INCLUDE_WORK_CANDIDATES CHAR(1) DEFAULT ('T') - apparently not in KDOT_BLP PRODUCTION -- Working on script to fix INSPUSRGUID where either null or not a GUID already. WIP

2023-01-02

-- Added dedicated procedure to create and drop KDOTBLP Portal tables, registered in table KDOTBLP_PORTAL_TABLES, and incorporated the same proc into the upgrade script itself.
An optional setting in the upgrade script will kill these temp tables at the end of the upgrade process - the default is 0 to NOT drop them.

Looks like:

```
                CREATE OR REPLACE PROCEDURE BACKUP_KDOTBLP_PORTAL_TABLES AS

                -- THIS PROCEDURE MAKES A BACKUP WITH A UNIQUE NAME  ENDING IN _KT
                -- FOR  ANY TABLES REGISTERED IN TABLE KDOTBLP_PORTAL_TABLES
                -- ALSO INCORPORATED DIRECTLY IN OracleBRM6.sql

                -- ARMarshall, ARM_LLC - 20230102 - created (and in
OracleBRM6.sql)

                TYPE R_CURSOR IS REF CURSOR;
                CUR R_CURSOR;

                V_Q      VARCHAR2(4000);
                V_SUFFIX VARCHAR2(36) := q'[_]' || HEXTORAW(SYS_GUID()) ||
q'[_KT]';
                V_TN     VARCHAR2(128);

                BEGIN

                -- commented out for stand-alone
                --IF GET_VAR('SAVEAGENCYSYSTABLES') = 1 THEN
                OPEN CUR FOR
                    SELECT TABLE_NAME FROM KDOTBLP_PORTAL_TABLES ORDER BY
PROCESSING_ORDER;
                LOOP
                    FETCH CUR
                    INTO V_TN;
                    EXIT WHEN CUR%NOTFOUND;

                    DBMS_OUTPUT.PUT_LINE(q'[Archiving KDOT Portal Table ]' || V_TN
||
                                        q'[ TO ]' || V_TN || V_SUFFIX ||
q'[...]');

                    V_Q := TRIM(q'[CREATE TABLE ]' || TRIM(V_TN || V_SUFFIX) ||
                                q'[ AS SELECT * FROM ]' || TRIM(V_TN));
                    BEGIN
                    EXECUTE IMMEDIATE V_Q;
                    DBMS_OUTPUT.PUT_LINE(q'[Table ]' || V_TN || q'[ archived
OK]');
                    EXCEPTION
                    WHEN OTHERS THEN
                        DBMS_OUTPUT.PUT_LINE(q'[ERROR ARCHIVING TABLE: ]' || V_Q
||
                                            q'[: ]' || SQLERRM);
                    END;

                END LOOP;
                CLOSE CUR;
                -- commented out for stand-alone
                -- END IF;
                EXCEPTION
                WHEN OTHERS THEN
                    BEGIN
                    RAISE_APPLICATION_ERROR(-20000, SQLERRM);
                    END;
                END;
```

and the drop procedure looks like:

```

            CREATE OR REPLACE PROCEDURE DROP_KDOTBLP_BACKUP_TABLES AS
            -- THIS PROCEDURE DROPS ANY TABLES WITH NAMES ENDING IN _KT
            -- this proc is **_intended to be run standalone_** at any time to
drop and purge (permanently remove) **_all_** tables with name extension _KT
            -- left over from an upgrade script run
            -- usage (from SQL*PLUS or PL/SQL Command Window):
            -- SQL>EXEC DROP_KDOTBLP_BACKUP_TABLES;
            -- SQL>/
            -- EXTRACTED FROM OracleBRM6.sql
            -- ARMarshall, ARM_LLC - 20230102 - created (and in OracleBRM6.sql)
            --DECLARE

                v_TN1 VARCHAR2(30);
                v_q VARCHAR2(4000 CHAR);
                CURSOR curColData IS
                        SELECT ut.TABLE_NAME
                        FROM USER_TABLES ut
                        JOIN KDOTBLP_PORTAL_TABLES kt
                        ON  ut.TABLE_NAME LIKE kt.TABLE_NAME || '%_KT'
                        ORDER BY ut.TABLE_NAME;
            BEGIN
                --IF  GET_VAR('KDOT_TABLE_CLEANUP') = '1' THEN
                    /** H.1.1 | Start a loop through the cursor to create a
dynamic drop process  **/
                    OPEN curColData;
                    FETCH curColData INTO v_TN1;
                    WHILE (curColData%FOUND)
                    LOOP
                        BEGIN

                        DBMS_OUTPUT.PUT_LINE('Cleaning up KDOTBLP PORTAL TABLES
(_KT) tables -' || V_TN1);

                            v_q := 'DROP TABLE ' || V_TN1 || ' PURGEX';
                            execute immediate v_q;
                        EXCEPTION WHEN OTHERS THEN
                            DBMS_OUTPUT.PUT_LINE('ERROR F1: Dropping KDOTBLP
PORTAL TABLES (_KT) Table [' || v_q || ']; ' || SQLERRM);
                        END;
                        FETCH curColData INTO v_TN1;
                    END LOOP;
                    CLOSE curColData;
            -- END IF;
            --INSERT INTO PROFILING(PLACE, END_TIME) VALUES ('DROP TEMP KDOTBLP
PORTAL TABLES (_KT)',  CURRENT_TIMESTAMP);

            EXCEPTION

            WHEN OTHERS THEN
                BEGIN
                RAISE_APPLICATION_ERROR(-20000, SQLERRM);

                END;
            END;
```

2023-01-02

-- A very similar procedure based on DROP_KDOTBLP_BACKUP_TABLES named DROP_BRM_T_BACKUP_TABLES was created that is ***intended to be run standalone*** to drop and purge (permanently remove) ***all*** _T tables

```
        /*
        to see the tables that will be deleted:
        SELECT ut.TABLE_NAME,pt.table_name
                    FROM USER_TABLES ut
                    JOIN PON_TABLE pt
                    ON  ut.TABLE_NAME LIKE pt.TABLE_NAME || '%_T'
                    ORDER BY ut.TABLE_NAME;
        */

    -- usage (from SQL*PLUS or PL/SQL Command Window):
    -- SQL>EXEC DROP_BRM_T_BACKUP_TABLES;
    -- SQL>/
```

-- Everywhere I could find it, changed Oracle Object names to 128 bytes per the new Oracle 12C object name length limit

For example, ***TABLE_NAME VARCHAR2(128 BYTE) NOT NULL***,:

```
    v_q := 'CREATE TABLE PON_TABLE (
    TABLE_NAME VARCHAR2(128 BYTE)  NOT NULL,
    SHORT_NAME VARCHAR(20),
    RANK NUMBER(10,0)  NOT NULL,
    CONSTRAINT PON_TABLE_PK PRIMARY KEY (TABLE_NAME))';
```

and if there was a string collection variable of VARCHAR2(2000) or similar, changed to VARCHAR2(32767) which is permitted in code (not in table column sizes)
For example,

```
    CREATE TABLE VALIDATION_WARNING
    (
    VALIDATION_WARNING_GD VARCHAR2(32) NOT NULL,
    TABLE_NAME VARCHAR2(128 BYTE) NOT NULL,
    COL_NAME VARCHAR2(128 BYTE) NOT NULL,
    ROW_PK VARCHAR2(32) NOT NULL,
    INPUT_VALUE VARCHAR2(3999 CHAR) NOT NULL,
    PON_APP_USERS_GD VARCHAR2(32) NOT NULL,
    VALIDATION_DATE DATE NOT NULL,
    VALIDATION_SOURCE_GENERIC VARCHAR2(128 BYTE) NOT NULL,
    VALIDATION_SOURCE_SPECIFIC VARCHAR2(1000) NOT NULL,
    VALIDATION_MESSAGE VARCHAR2(3999 CHAR) NOT NULL);
```

-- Script modified to use procedure backup_KDOTBLP_PORTAL_TABLES to backup separately every table listed in KDOTBLP_PORTAL_TABLES with unique name and suffix _KT, added global var KDOT_TABLE_CLEANUP to kill the tables after the run, defaults to 0');
-- Fixed apparent typo IS_K1EY to IS_KEY in the inserts to DATADICT_T near line 6080');

-- moved the ***prerequisite scripts*** to ***C:/git/repos/KDOT-LBIS-BRM-MIGRATION/files/scripts/***

-- a bunch of calls to SUBSTR were assuming object names were limited to 30, so they used SUBSTR(TABLE_NAME, 1, 28) || '_T' for example -- all of these (hopefully) were found and upgraded. Also affected temporary tables like PON_FK and PON_TABLE where the object name lengths were 30..

```
            DBMS_OUTPUT.PUT_LINE('20230103 - ARMarshall, ARM LLC - EXAMPLE SNIPPET (near line 4860:');
            DBMS_OUTPUT.PUT_LINE(q['>>>>>>>>>>>>>>> -- Oracle object name used to be 30]');
            DBMS_OUTPUT.PUT_LINE(q['>>>>>>>>>>>>>>> -- the SUBSTR function here has been changed to permit up to 126 chars
             of the real name to be used for the _T table ]');
            DBMS_OUTPUT.PUT_LINE(q['>>>>>>>>>>>>>>> -- given that the suffix _T takes up 2 chars.]');
            DBMS_OUTPUT.PUT_LINE(q[' WITH T(TABLE_NAME) AS (SELECT DISTINCT SUBSTR(TABLE_NAME, 1, 126) || '_T' FROM PON_DICT  ] etc.');
```

- table PON_NAV_CONTROL has no parent record for PON_NAV_CONTROL_GD=C01F1D4AE11B4ABFBC3A837674D30DAD , so any attempt to create a FK from PON_APP_CONTROL_SECURITY etc. will fail. DELETED entries in PON_APP_CONTROL_SECURITY

```
            SELECT * FROM PON_NAV_CONTROL PNC WHERE PNC.PON_NAV_CONTROL_GD='C01F1D4AE11B4ABFBC3A837674D30DAD';

            DELETE PON_APP_CONTROL_SECURITY
            WHERE PON_NAV_CONTROLGD='C01F1D4AE11B4ABFBC3A837674D30DAD';
```

2023-01-04

-- create process and procedures to shrink local database from max to workable size. hese instructions assume you created 2 tablespaces PONT_TBL and KDOT_BLP and assigned some data files to these.

To create a tablespace (your directory location will be different) - this uses 3 4M dbf files. The names and extensions are up to you....:

```
create tablespace KDOT_BLP  datafile 'D:\oradata\orcl\df1.dbf' size 4m , 'D:\oradata\orcl\df2.dbf' size 4m,  'D:\oradata\orcl\df3.dbf' size 4m;
```

To add a file to a tablespace (your directory location will be different)

```
        ALTER TABLESPACE  KDOT_BLP   ADD DATAFILE ' D:\oradata\orcl\df4.dbf' SIZE
1000M
```

To see what you have , run the script ***Script_Check_Tablespace_Size.pdc*** from a command window e.g. SQL*PLUS and something like this should show up, where the directory locations will probably be entirely different:

```
        SQL>

        TABLESPACE_NAME                         TOTAL  AVAILABLE       USED    PCTUSED


        ------------------------------ ---------- ---------- ---------- ----------
        PONT_TBL                                 3072    3068.88       3.12        0.1

        FILE_NAME
BYTES/1024/1024


        --------------------------------------------------------------------------
------ ---------------
        C:\PRJ\KDOT\ORA\DATA\PONT_TBL_DATA1.DBF
1024
        C:\PRJ\KDOT\ORA\DATA\PONT_TBL_DATA2.DBF
1024
        C:\PRJ\KDOT\ORA\DATA\PONT_TBL_DATA3.DBF
1024

        SQL>

        TABLESPACE_NAME                         TOTAL  AVAILABLE       USED    PCTUSED
        ------------------------------ ---------- ---------- ---------- ----------
        PONT_TBL                                 3072    3068.88       3.12        0.1

        FILE_NAME
BYTES/1024/1024
        --------------------------------------------------------------------------
------ ---------------
        C:\PRJ\KDOT\ORA\DATA\PONT_TBL_DATA1.DBF
1024
        C:\PRJ\KDOT\ORA\DATA\PONT_TBL_DATA2.DBF
1024
        C:\PRJ\KDOT\ORA\DATA\PONT_TBL_DATA3.DBF
1024

        SQL>

        TABLESPACE_NAME                         TOTAL  AVAILABLE       USED    PCTUSED
        ------------------------------ ---------- ---------- ---------- ----------
        KDOT_BLP                            8604.9375    6222.81  2382.1275      27.68
        FILE_NAME
BYTES/1024/1024
        --------------------------------------------------------------------------
------ ---------------
        C:\PRJ\KDOT\ORA\DATA\KDOT_BLP_DATA1.DBF
3131.25
        C:\PRJ\KDOT\ORA\DATA\KDOT_BLP_DATA2.DBF
2807.75
        C:\PRJ\KDOT\ORA\DATA\KDOT_BLP_DATA3.DBF
2665.9375

        SQL>
```

Process to squish database:

1.) Run the sql script ***Script_Show_Total_Database_Size_GB.sql*** - it shows the total GB and other stuff

2.) Delete cruft (like we discussed earlier e.g. BRIDGE_AR INSPEVNT_AR, ARC_****, *_T, *_tmp etc. etc.) and remember to PURGE the recycle bin after dropping the cruft. I actually deleted all the materialized views as well to save space.

3.) Run the attached script ***Script_Create_KDOTBLP_PORTAL_TABLES_Entries.sql*** to create the table KDOTBLP_PORTAL_TABLES and add the tables that you may care about. Add or remove table namess as you see fit - these are the ones that will be shrunk. They will be processed in the order shown. You only have to create this table once....

4.) Create the shrink procedure by running ***Script_Create_Replace_SHRINK_KDOTBLP_TABLES.pdc***

5.) Run the shrink procedure... load the file ***Script_Execute_SHRINK_KDOTBLP_TABLES.pdc*** and run it

6.) Check the work by repeating step 1.

and also make sure ***no objects exist in a dead tablespace***
e.g. PONT_TBL for a LOCAL development database.

2023-01-04

(CONTINUED)

-- changed tablespace sizes etc. to make sure the Express db does not exceed 12G overall - it was causing script errors.
-- created mismatch check script
Script_Show_USERBRDG_and_USERINSP_And_USRSTRUNIT_NonMatches_Parent.sql

2023-01-05

-- some code near 35960 of the BrM5.3 Oracle script is trying to enable a NOT NULL constraint when it is already set to NOT NULL and ENABLED. ***Ignorable***

```
            -- Enable disabled NOT NULL constraints.
            -- If successful this will prevent "ORA-01442: column to be modified to NOT NULL is already NOT NULL" when making columns required in the next block. 2.
            -- Will fail with "ORA-02293: cannot validate" if there is bad data.
            SELECT 'ALTER TABLE ' || D.TABLE_NAME || ' ENABLE CONSTRAINT ' || CONS.CONSTRAINT_NAME
                FROM PON_DICT D
                JOIN USER_TAB_COLUMNS C ON C.TABLE_NAME = D.TABLE_NAME AND C.COLUMN_NAME = D.COL_NAME
                JOIN USER_CONSTRAINTS CONS ON CONS.TABLE_NAME = D.TABLE_NAME
                JOIN USER_CONS_COLUMNS CONS_COLS ON CONS_COLS.CONSTRAINT_NAME = CONS.CONSTRAINT_NAME AND CONS_COLS.COLUMN_NAME = D.COL_NAME
                WHERE D.REQUIRED = 1 AND
                    D.PK <> 1 AND
                    C.NULLABLE = 'Y' AND
                    D.COL_NAME NOT LIKE '%CAPTION_ID' AND
                    CONS.STATUS = 'DISABLED' AND
                    CONS_COLS.POSITION IS NULL AND
                    INSTR(F_READ_SEARCH_CONDITION(CONS.CONSTRAINT_NAME), 'IS NOT NULL') > 0
                ORDER BY D.TABLE_NAME, D.COL_NAME;
```

and this

```

                --  Make some non PK columns required.
                DECLARE V_CUR SYS_REFCURSOR;
                BEGIN
                    DBMS_OUTPUT.PUT_LINE('Make non PK GUID columns required.');

                    OPEN V_CUR FOR
                    SELECT 'ALTER TABLE ' || D.TABLE_NAME || ' MODIFY ' || D.COL_NAME || ' ' || ' NOT NULL'
                    FROM PON_DICT D
                    JOIN USER_TAB_COLUMNS C ON C.TABLE_NAME = D.TABLE_NAME AND C.COLUMN_NAME = D.COL_NAME
                    WHERE D.REQUIRED = 1 AND D.PK <> 1 AND C.NULLABLE = 'Y' AND D.COL_NAME NOT LIKE '%CAPTION_ID'
                    ORDER BY D.TABLE_NAME, D.COL_NAME;

                    IF GET_VAR('PRINT_DETAILS') = 1 THEN
                            REV(V_CUR,1);
                        ELSE
                            REV(V_CUR);
                        END IF;
                    CLOSE V_CUR;

                END;
                /

```

-- 20230105 - ARMarshall,ARM LLC -added/restored NOT NULL constraint to USERRWAY.ON_UNDER and USRSTRUNIT.STRUNITKEY

-- The following constraints remained disabled after a run. All of these are GUID columns. This is near line 35983 ...

- ALTER TABLE PON_APP_GROUPS ENABLE CONSTRAINT SYS_C0031990
- ALTER TABLE PON_APP_PERMISSIONS ENABLE CONSTRAINT SYS_C0031586
- ALTER TABLE PON_APP_ROLES_PERMISSIONS ENABLE CONSTRAINT SYS_C0031146
- ALTER TABLE PON_APP_ROLES_PERMISSIONS ENABLE CONSTRAINT SYS_C0031145
- ALTER TABLE PON_APP_USERS_GROUPS ENABLE CONSTRAINT SYS_C0031184
- ALTER TABLE PON_APP_USERS_GROUPS ENABLE CONSTRAINT SYS_C0031183
- ALTER TABLE PON_APP_USERS_ROLES ENABLE CONSTRAINT SYS_C0032018
- ALTER TABLE PON_APP_USERS_ROLES ENABLE CONSTRAINT SYS_C0032017

-- 20230105 - ARMarshall,ARM LLC - removed function named "f_is_active(pBridgeGroup IN VARCHAR2)" - bad name. Fixed and capitalized. See script **Script_Create_Replace_F_IS_ACTIVE_BRIDGEGROUP.fnc_**
-- 20230105 - ARMarshall,ARM LLC - addded post step to restore a couple missing indexes on KDOTBLP_BRIDGE - see script ***Script_Restore_Missing_KDOT_Objects.pdc*** where all such restores are done.

-- 20230105 - ARMarshall,ARM LLC - addded script to remove records with no parents and create child records for KDOTBLP_BRIDGE, KDOTBLP_INSPECTIONS, USERSTRUNIT

2023-01-07

The following two commands should be issued before attempting an impdb run

```
            SQL>connect sys@xe as sysdba;
            SQL<create or replace directory KDOT_DP_DIR as
'C:\temp\Ora\DataPump\KDOT'; --<<< Windows file system location >>>
```

The connection string for IMPDP should be as sys and look like

```
            SYS>CONNECT sys/********@localhost:1521/xepdb1 AS SYSDBA
```

ROLE BRMADMIN must exist in the database in advance

The user KDOT_BLP should be DROPPED before first attempt to import. Can exist afterwards.
The user KDOT_BLP can be granted SYSDBA, ***optionally***
The user KDOT_BLP may be granted exp_full_database and imp_full_database ***optionally***.

2023-01-09

Here is how the user (schema) KDOT_BLP was set:

```
        -- Create the user
        create user KDOT_BLP
        default tablespace KDOT_BLP
        temporary tablespace TEMP
        profile DEFAULT
        password expire;
        -- Grant/Revoke role privileges
        grant connect to KDOT_BLP;
        grant plustrace to KDOT_BLP;
        grant resource to KDOT_BLP;
        -- Grant/Revoke system privileges
        grant alter session to KDOT_BLP;
        grant create any job to KDOT_BLP;
        grant create any materialized view to KDOT_BLP;
        grant create any synonym to KDOT_BLP;
        grant create database link to KDOT_BLP;
        grant create external job to KDOT_BLP;
        grant create job to KDOT_BLP;
        grant create materialized view to KDOT_BLP;
        grant create public synonym to KDOT_BLP;
        grant create synonym to KDOT_BLP;
        grant create table to KDOT_BLP;
        grant create view to KDOT_BLP;
        grant debug connect session to KDOT_BLP;
        grant drop any synonym to KDOT_BLP;
        grant export full database to KDOT_BLP;
        grant import full database to KDOT_BLP;
        grant manage scheduler to KDOT_BLP;
        grant select any dictionary to KDOT_BLP;
        grant unlimited tablespace to KDOT_BLP;
```

-- In Visual Studio Code, you can find all statements in the output log like **CREATE UNIQUE INDEX "KDOT_BLP"**. using a REGEX like this:

```
        ^(CREATE UNIQUE INDEX "KDOT_BLP"\.).+$
```

which helps finding errors reported in the log.

- 5.2 upgrade scripts ***521_March302015_PontisOracle52.sql*** and ***BrM_5_3\521_April2015_PontisOracle52.sql*** only vary by the release id in table DBDESCRP - did not use

20220109

Testing with BrM 53

```
Error#1:
reference to an unknown column ORA-00904: "grps_name":  when enclosed in quotes
like this it is matched by case.

<span style="color:green;font-weight:500;font-size:12px;font-style: italic;">
System.Web.HttpUnhandledException (0x80004005): Exception of type
'System.Web.HttpUnhandledException' was thrown. System.Exception:
BridgeRepository::GetBridgeList: Unable To Retrieve Bridge List  System.Exception:
AdoRepository::ExecuteReader Error executing command: SELECT * FROM (SELECT
"_sd_".*, ROWNUM "_#_" FROM (SELECT DISTINCT(pon_bridge_grps.pon_bridge_grps_gd)
AS "key_", pon_bridge_grps.grps_name AS "Bridge Groups", (select
COUNT(DISTINCT(bridge_gd))from pon_grps_road r where r.pon_bridge_grps_gd =
pon_bridge_grps.pon_bridge_grps_gd) "Bridge_Count", (select COUNT(bridge_gd)from
pon_grps_road r where r.pon_bridge_grps_gd = pon_bridge_grps.pon_bridge_grps_gd)
"Roadway_Count",pon_bridge_grps.description as "Description" from pon_bridge_grps
left outer join pon_grps_road on pon_grps_road.pon_bridge_grps_gd =
pon_bridge_grps.pon_bridge_grps_gd left outer join bridge on bridge.bridge_gd =
pon_grps_road.bridge_gd WHERE 1=1 order by "grps_name" ASC) "_sd_" WHERE ROWNUM <=
1) "_sd2_" WHERE "_sd2_"."_#_" > 0 order by ROWNUM
Oracle.ManagedDataAccess.Client.OracleException: ORA-00904: "grps_name": invalid
identifierat
OracleInternal.ServiceObjects.OracleCommandImpl.VerifyExecution(OracleConnectionIm
pl connectionImpl, Int32& cursorId, Boolean bThrowArrayBindRelatedErrors,
OracleException& exceptionForArrayBindDML, Boolean& hasMoreRowsInDB, Boolean
bFirstIterationDone)at
OracleInternal.ServiceObjects.OracleCommandImpl.ExecuteReader(String commandText,
OracleParameterCollection paramColl, CommandType commandType, OracleConnectionImpl
connectionImpl, OracleDataReaderImpl& rdrImpl, Int32 longFetchSize, Int64
clientInitialLOBFS, OracleDependencyImpl orclDependencyImpl, Int64[]
scnForExecution, Int64[]& scnFromExecution, OracleParameterCollection&
bindByPositionParamColl, Boolean& bBindParamPresent, Int64& internalInitialLOBFS,
OracleException& exceptionForArrayBindDML, Boolean isDescribeOnly, Boolean
isFromEF)at Oracle.ManagedDataAccess.Client.OracleCommand.ExecuteReader(Boolean
requery, Boolean fillRequest, CommandBehavior behavior)at
Oracle.ManagedDataAccess.Client.OracleCommand.ExecuteDbDataReader(CommandBehavior
behavior)at
BRIDGEWare.Pontis.DataAccess.Repositories.AdoRepository.ExecuteReader(String
commandText, Dictionary`2 queryParameters, Nullable`1 expectedRowCountHint, List`1
columnFilter, Nullable`1 timeout, Boolean forceSemiColonStrip) in
E:\Agents\PRG01\_work\33\s\DataAccess\Repositories\AdoRepository.cs:line 91 ---
End of inner exception stack trace --- at
BRIDGEWare.Pontis.DataAccess.Repositories.AdoRepository.ExecuteReader(String
commandText, Dictionary`2 queryParameters, Nullable`1 expectedRowCountHint, List`1
columnFilter, Nullable`1 timeout, Boolean forceSemiColonStrip) in
E:\Agents\PRG01\_work\33\s\DataAccess\Repositories\AdoRepository.cs:line 130at
BRIDGEWare.Pontis.DataAccess.Repositories.BridgeRepository.GetBridgeList(GridSearc
hCriteria criteria, Int32 currentPage, Int32 rowsPerPage) in
E:\Agents\PRG01\_work\33\s\DataAccess\Repositories\BridgeRepository.cs:line 136  -
- End of inner exception stack trace --- at
BRIDGEWare.Pontis.DataAccess.Repositories.BridgeRepository.GetBridgeList(GridSearc
hCriteria criteria, Int32 currentPage, Int32 rowsPerPage) in
E:\Agents\PRG01\_work\33\s\DataAccess\Repositories\BridgeRepository.cs:line 155at
PontisModules_Bridge_UiDTCBagGrid.get_GridList()at
PontisModules_Bridge_UiDTCBagGrid.get_GridDataSource()at
ContextGridControl.SetGridDataSource(String sortColumnName, Boolean filterResults,
Boolean showAllWhenUnfiltered)at PontisModules_Bridge_UiDTCBagGrid.BindData()at
PontisModules_Bridge_UiDTCBagGrid.Page_Load(Object sender, EventArgs e)at
System.Web.UI.Control.OnLoad(EventArgs e)at
System.Web.UI.Control.LoadRecursive()at System.Web.UI.Control.LoadRecursive()at
System.Web.UI.Control.LoadRecursive()at System.Web.UI.Control.LoadRecursive()at
System.Web.UI.Control.LoadRecursive()at System.Web.UI.Control.LoadRecursive()at
System.Web.UI.Control.LoadRecursive()at System.Web.UI.Control.LoadRecursive()at
System.Web.UI.Control.LoadRecursive()at System.Web.UI.Control.LoadRecursive()at
System.Web.UI.Control.LoadRecursive()at System.Web.UI.Control.LoadRecursive()at
System.Web.UI.Control.LoadRecursive()at System.Web.UI.Control.LoadRecursive()at
System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint,
Boolean includeStagesAfterAsyncPoint)at System.Web.UI.Page.HandleError(Exception
e)at System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint,
Boolean includeStagesAfterAsyncPoint)at System.Web.UI.Page.ProcessRequest(Boolean
includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint)at
System.Web.UI.Page.ProcessRequest()at
System.Web.UI.Page.ProcessRequest(HttpContext context)at
ASP.bridges_bridgeanalysisgrplist_aspx.ProcessRequest(HttpContext context) in
c:\Users\armpl\AppData\Local\Temp\Temporary ASP.NET
Files\root\8d3e4b96\eea78b4\App_Web_f5frnmw3.0.cs:line 0at
System.Web.HttpApplication.CallHandlerExecutionStep.System.Web.HttpApplication.IEx
ecutionStep.Execute()at System.Web.HttpApplication.ExecuteStepImpl(IExecutionStep
step)at System.Web.HttpApplication.ExecuteStep(IExecutionStep step, Boolean&
completedSynchronously)</span>
```

```
            SELECT *
            FROM (SELECT "_sd_".*, ROWNUM "_#_"
                    FROM (SELECT DISTINCT (pon_bridge_grps.pon_bridge_grps_gd) AS
"key_",
                                            pon_bridge_grps.grps_name AS "Bridge
Groups",
                                            (select COUNT(DISTINCT(bridge_gd))
                                            from pon_grps_road r
                                            where r.pon_bridge_grps_gd =

pon_bridge_grps.pon_bridge_grps_gd) "Bridge_Count",
                                            (select COUNT(bridge_gd)
                                            from pon_grps_road r
                                            where r.pon_bridge_grps_gd =

pon_bridge_grps.pon_bridge_grps_gd) "Roadway_Count",
                                            pon_bridge_grps.description as
"Description"
                            from pon_bridge_grps
                            left outer join pon_grps_road
                                on pon_grps_road.pon_bridge_grps_gd =
                                pon_bridge_grps.pon_bridge_grps_gd
                            left outer join bridge
                                on bridge.bridge_gd = pon_grps_road.bridge_gd
```

```
                        WHERE 1 = 1
                        order by "GRPS_NAME" ASC) "_sd_"
                WHERE ROWNUM <= 1) "_sd2_"
        WHERE "_sd2_"."_#_" > 0
        order by ROWNUM
```

-- need some aliases (synonyms) -- these synonym names are magic strings that the portal automatically replaces in all occurrences.

```
                Connected to Oracle Database 21c Express Edition Release
21.0.0.0.0
                Connected as KDOT_BLP@localhost:1521/xepdb1

                                SQL> CREATE PUBLIC SYNONYM XTRNBRDGTABLE FOR
KDOTBLP_BRIDGE;
                                SQL> CREATE PUBLIC SYNONYM XTRNINSPTABLE FOR
KDOTBLP_INSPECTIONS;
                                SQL> CREATE PUBLIC SYNONYM XTRNRWAYTABLE FOR
USERRWAY;
                                SQL> CREATE PUBLIC SYNONYM XTRNSTRUNITTABLE FOR
USERSTRUNIT;

                Synonym created
```

-- some filter SQL needs repair e.g. in this case, the alias b was not used where it needed to be: -- this is current layout Bridges w/ No Inspections
-- used to say ....

```
               ... ON inspevnt.bridge_gd = bridge.bridge_gd
```

but that makes the ADO database layer unhappy so must be consistent

```
              ... ON inspevnt.bridge_gd = b.bridge_gd

        SELECT DISTINCT (b.bridge_gd) "key_",
                        b.bridge_id "Bridge ID",
                        b.district "District",
                        b.county "County",
                        b.facility "Facility Carried",
                        b.featint "Feature Intersected",
                        b.owner "Own",
                        b.custodian "Maint",
                        b.yearbuilt "Built",
                        b.brkey "brkey"
        from bridge b
        left outer join (select *
                            from roadway
```

```
                                where roadway.ON_UNDER = '1'
                                or (ROADWAY.bridge_gd not in
                                    (select bridge_gd
                                        from ROADWAY
                                        where roadway.ON_UNDER = '1') and
                                    (roadway.on_under = '2' or roadway.on_under =
'A'))) roadway
                on roadway.bridge_gd = b.bridge_gd
            left outer join XTRNBRDGTABLE
                on XTRNBRDGTABLE.bridge_gd = b.bridge_gd
            left outer join XTRNRWAYTABLE
                on roadway.roadway_gd = XTRNRWAYTABLE.roadway_gd
            left outer join (Select *
                                from (select inspevnt.*,
                                            row_number() over(partition by
bridge_gd order by inspdate, createdatetime desc) rn
                                        from inspevnt) inspevnt2
                                where rn = 1) inspevnt
                ON inspevnt.bridge_gd = b.bridge_gd
            left outer join XTRNINSPTABLE
                on XTRNINSPTABLE.inspevnt_gd = inspevnt.inspevnt_gd
            left outer join structure_unit
                on structure_unit.bridge_gd = b.bridge_gd
            left outer join XTRNSTRUNITTABLE
                on XTRNSTRUNITTABLE.structure_unit_gd =
                structure_unit.structure_unit_gd
            where b.bridge_gd not in (select i2.bridge_gd from inspevnt i2)
```

and this won't work

```
                      SELECT "sqlfilter_".*
                    FROM (SELECT (b.bridge_gd) AS "Key_",
```

has to be "**key_**"

```
                      SELECT "sqlfilter_".*
                    FROM (SELECT (b.bridge_gd) AS "key_",
```

These difficulties necessitate minor but pervasive code changes to make every layout say

```
                ...bridge_gd as "key_"
```

and ***never make BRKEY the key_***

```
                ...brkey as "key_"
```

and update the C# and JS throughout to use that convention. There are definitely exceptions as of 20230109

20230110

BrM 5.3 can run against the database migrated to 6.0 at least as far as seeing the desktop.

20230117

---

added script to quantify PON_FILTERS with layouts conforming to BRM requirements

- uses a REGEX to check the layouts for bare minimum compliance.
- created a SQL script to try and exercise every layout - check compliance and whether it returns any rows
- created this Oracle REGEX to check layouts:

```
select pf.filterkey, pf.name ,nvl(1,0) as compliant,pf.sql_filter, pf.rowid
  FROM PON_FILTERS pf
 WHERE  pf.filterkey>=15000 and pf.accessfilter=0 and pf.shared=1
and  ( regexp_like(lower(trim(pf.sql_filter)),
                   q'[^.*SELECT.*(b\.bridge_gd +as +key_).*(b\.brkey).*FROM( *| *
* *)(bridge +b).*(ORDER +BY +(1|b\.brkey))*.*((/* *#END# *\*/)*).*\Z]' ,'inm')
);
```

- and this C# function

```


        /// <summary>
        ///     Check the desktop list for obvious missing elements
        ///     Ignores any case-sensitivity in the SQL fragments
        /// </summary>
        /// <param name="sqlString"></param>
        /// <param name="errFatal">True if the error is deemed fatal</param>
        /// <param name="errsOut">List of error messages generated by the
validation</param>
        /// <param name="throwError">Raise an exception if a fatal error is
encountered in validation</param>
        /// <returns></returns>
        public static bool ValidateDesktopListSqlRegex(this string sqlString, out
bool errFatal, out string errsOut,
            bool throwError = false)
        {
            var errMsg = new StringBuilder().AppendLine("");
            errFatal = false; // be optimistic
            try
            {
                var regexOpts = new RegexOptions();

                var errCount = 0;
                var warnCount = 0;
                var infoCount = 0;

                regexOpts = RegexOptions.IgnoreCase | RegexOptions.Singleline |
RegexOptions.CultureInvariant;
                //1 see if the list contains FROM BRIDGE b in any way shape or
form
                var regex = new Regex(@"^.*?([ ]*FROM[ ]*BRIDGE[ ]*b[ ]*).*?$",
regexOpts);
                var result = regex.IsMatch(sqlString); // either t or f on first
call

                if (!result)
                {
                    errCount++;
                    errMsg.AppendLine(
                        $@"Check #1 - CHECK SQL - the sql does not contain any
version of {"FROM BRIDGE b"} - (regex: {regex}) - this SQL fragment Is usually
required for all layout list SQL");
                }

                //2 - must contain SELECT b.BRIDGE_GD as key_ in some way spaces
do not matter
                // this is a BrM requirement
                // ARMarshall, ARM_LLC - 20230111 - changed to bridge_gd from
brkey
                regex = new Regex("^(.*?)(SELECT[ ]+b\\.bridge_gd[ ]+(?:as[ ]+?)*?
key_[ ]*,)(.*?)$", regexOpts);
                result = regex.IsMatch(sqlString);

                if (!result)
                {
                    errFatal = true;
                    errCount++;
                    errMsg.AppendLine(
                        $@"Check #2 - ERROR - the sql does not contain any version
of {"SELECT b.bridge_gd as key_"} - (regex: {regex}) - this SQL fragment is
required for all layout list SQL and BrM");
                }



                //3 - should contain b.bridge_gd, maybe with more columns before
and after it..
                regex = new Regex("^.*?[ ]*([,][ ]*?b\\.bridge_gd)[ ]*[,]*[ ]*.*?
$", regexOpts);
                result = regex.IsMatch(sqlString);

                if (!result)
                {
                    warnCount++;
                    errMsg.AppendLine(
                        $@"Check #3 -  CHECK SQL - the layout sql must contain the
fragment {"bridge_gd"} - (regex: {regex}) -this column was not found in the sql
string and is required for all layout list SQL.");
                }


                //4 - should contain b.bridge_gd, maybe with more columns before
and after it..
                regex = new Regex("^.*?[ ]*([,][ ]*?b\\.brkey)[ ]*[,]*[ ]*.*?$",
regexOpts);
                result = regex.IsMatch(sqlString);

                if (!result)
                {
                    warnCount++;
                    errMsg.AppendLine(
                        $@"Check #4 -  CHECK SQL - the layout sql must contain the
fragment {"b.brkey"} - (regex: {regex}) -this column was not found in the sql
string and is required for all layout list SQL.");
                }


                // log this before throwing the exception
                if (!result) // something bad happened
                {
                    if (errFatal)
                        Logger.Instance.Error(errMsg);
                    else
                        Logger.Instance.Warn(errMsg);
                }

                if (errCount > 0 || warnCount > 0 || infoCount > 0)
                {
                    errMsg.AppendLine("");
                    errMsg.AppendLine(
                        $"Summary: {errCount} errors, {warnCount} warnings, and
{infoCount} info messages");
                }

                errsOut = errCount > 0 || warnCount > 0 ? errMsg.ToString() :
Empty;

                if (throwError && errFatal)
                    throw new ArgumentException(
                        $"Desktop list sql: {sqlString.FormatSqlString()}
{Environment.NewLine} Errors:{Environment.NewLine}{errMsg}");

                //  possibly passed, or it failed so there are >0 errors in the
errCount
                return errCount == 0;
            }
            catch (RegexMatchTimeoutException rte)
            {
                rte.Log(rte.GatherExceptionData());
                throw;
            }
            catch (Exception ex)
            {
                errMsg.AppendLine("");
                errMsg.AppendLine("Layout Sql:");
                errMsg.Append($"{sqlString.FormatSqlString()}");
                errMsg.AppendLine("");
                errMsg.Append(ex.GatherExceptionData());
                errMsg.AppendLine("");

                ex.Log(errMsg.ToString());

                throw;
            }
        }
```

where the first regex is:

```
^.*?([ ]*FROM[ ]*BRIDGE[ ]*b[ ]*).*?$
```

which verifies that the filter SQL contains the **FROM** clause:

```
FROM BRIDGE b
```

and the second regex is:

```
^(.*?)(SELECT[ ]+b\\.bridge_gd[ ]+(?:as[ ]+?)*?key_[ ]*,)(.*?)$
```

which verifies that the SQL contains the expression:

```
 b.bridge_gd as key_
```

20230118

Updated BrM code throughout to assume BRIDGE_GD as KEY_ convention (see above)

20230123

Added 2 functions to ESOADEV

- FUNCTION **F_IS_ACTIVE_BRIDGEGROUP**(pBridgeGroup IN VARCHAR2)
- FUNCTION **f_Get_Pon_App_Users_Gd_For_Userkey**(The_Userkey Pon_App_Users.Userkey%TYPE)

replaced 3 views (updated)

- **V_BRIDGES_WITH_FC_INSPECTIONS**
- **V_BRIDGES_WITH_UW_INSPECTIONS**
- **V_BRIDGES_WITH_OS_INSPECTIONS**

added tables to ESOADEV

- added **KDOTBLP_PORTAL_TABLES** to itemize tables that the upgrade script should backup before doing anything else
- added table **VALIDATION_WARNINGS** to create a table missing from upgrade scripts but apparently required by BrM
- added table **PON_MOBILE_ERRORS** to create a table missing from upgrade scripts but apparently required by BrM

20230125

Ran the upgrade script against ESOADEV. Minor twiddles and it finished successfully. Found duplicate KDOTBLP_INSPECTIONS records - same BRKEY/INSPKEY - created scripts to find and remove these

20230125

Ran script to clean orphaned KDOTBLP_INSPECTIONS with no parent INSPEVNT record Ran script to clean duplicate KDOTBLP_INSPECTIONS (Re)-Ran the upgrade script against ESOADEV.

- Minor twiddles and it finished successfully. **Testing**
- My testing with BrM 5.3 and 6.0 has been ***very cursory***, but I can connect to them, run validation, calculate sufficiency rating, select / unselect etc. etc. No attempts at formulating a future bridge program for the BLP as yet (!!!)
- I do have some apparently bad news. BrM uses Telerik (apparently) and uses the old version that we have determined is vulnerable. ***The dependency on Telerik may not exist in future versions - don't know yet.***
- Also, there is cruft lying around in the web.config which suggests that they did not thoroughly check it between 5.3 and 6.0 but I suspect the differences are not profound.
- I ran into a problem trying to tweak **PARAMTRS** table entries for the KDOTBLP_INSPECTIONS special inspection type SPINSPTYPE. I have reported it to Mayvue. It is issue #BRMSD-6826 https://bridgeware.atlassian.net/servicedesk/customer/portal/1/BRMSD-6826

Augmented and ran script Fixup_Missing_INSPEVNT_User_Identifiers.pdc to fix inspevnt records where user identifiers are null e.g. USERKEY Updated .gitignore to exclude unwanted files fro version control Reset web.config for both BrM 5.3 and 6.0 to use the file names BrM5_3.log and BrM6_0.log respectively for the log4net log files

20230126

Ran Oracle 6.1 upgrade script after patching the BrM exe.

- It has an error near line 212 - extraneous space before the slash at the end of the ISNUMERIC create script

```
-- ARMarshall, ARM_LLC -20230126 - removed extraneous space before the command
slash at the bottom of the create script for this function ISNUMERIC
--Above function (IS_INTEGER) returns false for 0, so creating a new one that
actually treats it as a number
CREATE OR REPLACE FUNCTION ISNUMERIC(p_string IN VARCHAR2)
   RETURN INT
IS
   v_new_num NUMBER;
BEGIN
   v_new_num := TO_NUMBER(p_string);
   RETURN 1;
EXCEPTION
WHEN VALUE_ERROR THEN
   RETURN 0;
END ISNUMERIC;
/
-- ARMarshall, ARM_LLC -20230126 - fixed the slash above to remove space
```

- Validation and sufficiency rating worked after upgrade
- About box does not show the right version though - review of DLL properties shows

```
BRIDGEWare.Pontis.www, Version=6.1.0.0, Culture=neutral, PublicKeyToken=null
```

Ran scripts for 6.2, 6.3, 6.4, 6.5, 6.6 - BrM and Portal both work with the rudimentary testing I performed

- It is important to note the the Portal BrM DLLs are from BrM 5.3, and were NOT updated. This is ***presumably OK***, given that we want the Portal to behave as it always has.
- Several weird things happening attempting to run BrM 6.4 where MULTIMEDIA was added to the application. There seeems to be a configuration error and maybe MULTIMEDIA directory permissions issues as well.

BrM through release 6.5 apparently uses an internal browser tool/control that is essentially an ancient version of Internet Explorer. This browser control does not support mapping, so the mapping does not work

BrM6.6 does not permit logging in as pontis/pontis. I can login as amyjcoon with her password fine. That is amazing.

20230130

- the upgrade script destroyed some KDOTBLP foreign key references from KDOTBLP_BRIDGE to BRIDGE and KDOTBLP_INSPECTIONS to INSPEVNT.

```
-- Create/Recreate primary, unique and foreign key constraints
alter table KDOTBLP_BRIDGE
  add constraint FK_KDOT_BRIDGE_TO_BRIDGE foreign key (BRIDGE_GD)
  references bridge (BRIDGE_GD) on delete cascade;
```

and

```
-- Create/Recreate primary, unique and foreign key constraints
alter table KDOTBLP_INSPECTIONS
  add constraint FK_KDOBTLP_INSPS_TO_INSPEVNT foreign key (INSPEVNT_GD)
  references inspevnt (INSPEVNT_GD) on delete cascade;
```