**DAISEY DBA TASKS**

**ATS DBA**

***June 2018***

**Contents**

1. **Vacuum Databases**
2. **Monitor Extracts Run Durations**
3. **Refresh Users Passwords**
4. **Appendix A: Bloat Script**
5. **Appendix B: Extracts Script**
6. **Vacuum Databases**

The four databases Prod2 Reporting, Prod2 Transactional, Fix2 Reporting, and Fix2 Transactional are regularly checked for bloat. Production databases are checked weekly and Fix database are checked as needed. The script used for all four is given in Appendix A.

Auto vacuum is running for all databases.

If it is determined that the bloat or performance issues justify a full vacuum, arrangements are made with CPPR to run a full vacuum. This process can use one of the two scripts below.

Once you run a query from below, copy and paste the results in a query widow for execution.

Single line execution enables you to vacuum individual tables. You get a report on each line that has been executed.

--DAISEY

--All tables listed

SELECT 'vacuum full verbose analyze ' || tablename ||';' FROM pg\_catalog.pg\_tables where schemaname='kehs';

This will vacuum the entire database. You will get a report at the end of the execution.

--Entire database

VACUUM FULL VERBOSE ANALYZE

1. **Monitor Extracts Run Durations**

The script in Appendix B is scheduled to run several times each day and email the durations to the DBA Team. You can also execute it manually when needed, and also change the record count at the end of the script.

This is an example of a message, indicating that the duration was 20 seconds

pexport.sql:337: NOTICE: DAISEY Reporting Durations in Fix2 for:2018-06-06 12:35:01

pexport.sql:337: NOTICE: Duration hh:mm:ss (expected run time around 20 sec):00:00:20 DO

Currently, 1000 records are selected for benchmark comparisons. Currently, the typical duration for Prod2 Reporting is 40 seconds and for Fix2 Reporting it is 20 seconds.

If the durations are longer, it is necessary to check the database. It might be necessary to run a full vacuum.

1. **Refresh Users Passwords**

User passwords are to be refreshed every six months. Not all roles are included. Service/group roles are excluded.

The primary purpose is to refresh production passwords, both for the reporting and transactional databases. Currently, a user will have the same password for both databases.

Passwords are given to CPPR and individuals at ATS via Skype for Business.

We currently generate randomly passwords consisting of 14 characters. We avoid # since there have been some issues with the Java application.

Login as postgres to perform these tasks.

The command to change password for an existing user is

ALTER ROLE username WITH PASSWORD '##############’

For non-existing users it is necessary to create a user and then grant permissions by inheritance (here we show granting reader permissions) as follows

CREATE ROLE username with login;

GRANT kehs\_reader TO username GRANTED BY postgres;

1. **Appendix A: Bloat Script**

SELECT current\_database(), schemaname, tblname, bs\*tblpages AS real\_size, (tblpages-est\_tblpages)\*bs AS extra\_size, CASE WHEN tblpages - est\_tblpages > 0 THEN 100 \* (tblpages - est\_tblpages)/tblpages::float ELSE 0 END AS extra\_ratio, fillfactor, (tblpages-est\_tblpages\_ff)\*bs AS bloat\_size, CASE WHEN tblpages - est\_tblpages\_ff > 0 THEN 100 \* (tblpages - est\_tblpages\_ff)/tblpages::float ELSE 0 END AS bloat\_ratio, is\_na -- , (pst).free\_percent + (pst).dead\_tuple\_percent AS real\_fragFROM ( SELECT ceil( reltuples / ( (bs-page\_hdr)/tpl\_size ) ) + ceil( toasttuples / 4 ) AS est\_tblpages, ceil( reltuples / ( (bs-page\_hdr)\*fillfactor/(tpl\_size\*100) ) ) + ceil( toasttuples / 4 ) AS est\_tblpages\_ff, tblpages, fillfactor, bs, tblid, schemaname, tblname, heappages, toastpages, is\_na -- , stattuple.pgstattuple(tblid) AS pst FROM ( SELECT ( 4 + tpl\_hdr\_size + tpl\_data\_size + (2\*ma) - CASE WHEN tpl\_hdr\_size%ma = 0 THEN ma ELSE tpl\_hdr\_size%ma END - CASE WHEN ceil(tpl\_data\_size)::int%ma = 0 THEN ma ELSE ceil(tpl\_data\_size)::int%ma END ) AS tpl\_size, bs - page\_hdr AS size\_per\_block, (heappages + toastpages) AS tblpages, heappages, toastpages, reltuples, toasttuples, bs, page\_hdr, tblid, schemaname, tblname, fillfactor, is\_na FROM ( SELECT tbl.oid AS tblid, ns.nspname AS schemaname, tbl.relname AS tblname, tbl.reltuples, tbl.relpages AS heappages, coalesce(toast.relpages, 0) AS toastpages, coalesce(toast.reltuples, 0) AS toasttuples, coalesce(substring( array\_to\_string(tbl.reloptions, ' ') FROM '%fillfactor=#"\_\_#"%' FOR '#')::smallint, 100) AS fillfactor, current\_setting('block\_size')::numeric AS bs, CASE WHEN version()~'mingw32' OR version()~'64-bit|x86\_64|ppc64|ia64|amd64' THEN 8 ELSE 4 END AS ma, 24 AS page\_hdr, 23 + CASE WHEN MAX(coalesce(null\_frac,0)) > 0 THEN ( 7 + count(\*) ) / 8 ELSE 0::int END + CASE WHEN tbl.relhasoids THEN 4 ELSE 0 END AS tpl\_hdr\_size, sum( (1-coalesce(s.null\_frac, 0)) \* coalesce(s.avg\_width, 1024) ) AS tpl\_data\_size, bool\_or(att.atttypid = 'pg\_catalog.name'::regtype) AS is\_na FROM pg\_attribute AS att left JOIN pg\_class AS tbl ON att.attrelid = tbl.oid left JOIN pg\_namespace AS ns ON ns.oid = tbl.relnamespace left JOIN pg\_stats AS s ON s.schemaname=ns.nspname AND s.tablename = tbl.relname AND s.inherited=false AND s.attname=att.attname LEFT JOIN pg\_class AS toast ON tbl.reltoastrelid = toast.oid WHERE att.attnum > 0 AND NOT att.attisdropped AND tbl.relkind = 'r' GROUP BY 1,2,3,4,5,6,7,8,9,10, tbl.relhasoids ORDER BY tblname--2,3 ) AS s ) AS s2 ) AS s3 where schemaname='kehs' order by real\_size

1. **Appendix B: Extracts Script**

--Extracts Run Durations

--You may change the limit (1000) at the bottom.

--The scheduled validation uses 1000.

--Run in DAISEY Reporting Fix2 and Prod2.

SELECT CAST("clients"."Client Type" AS TEXT) AS "Client Type",

CAST("clients"."DAISEY ID (Profile)" AS TEXT) AS "DAISEY ID (Profile)",

"program visit forms"."DAISEY ID (Program Visit)" AS "DAISEY ID (Program Visit)",

"service forms"."DAISEY ID (Service)" AS "DAISEY ID (Service)",

"program visit forms"."Date of Activity (Program Visit)" AS "Date of Activity (Program Visit)",

"service forms"."Date of Activity (Service)" AS "Date of Activity (Service)",

CAST("clients"."First Name" AS TEXT) AS "First Name",

"program visit forms"."Form Data ID (Program Visit)" AS "Form Data ID (Program Visit)",

"service forms"."Form Data ID (Service)" AS "Form Data ID (Service)",

"clients"."Form ID (Profile)" AS "Form ID (Profile) (clients)",

"forms"."Form ID (Profile)" AS "Form ID (Profile)",

"program visit forms"."Form ID (Program Visit)" AS "Form ID (Program Visit) (Custom SQL Query)",

"forms"."Form ID (Program Visit)" AS "Form ID (Program Visit)",

"service forms"."Form ID (Service)" AS "Form ID (Service) (Custom SQL Query)",

"forms"."Form ID (Service)" AS "Form ID (Service)",

"clients"."Grantee ID" AS "Grantee ID",

"clients"."Grantee" AS "Grantee",

CAST("clients"."Last Name" AS TEXT) AS "Last Name",

"clients"."Module ID" AS "Module ID",

"clients"."Module" AS "Module",

1 AS "Number of Records",

"clients"."Organization ID" AS "Organization ID",

"clients"."Organization" AS "Organization",

CAST("forms"."Program ID (Program Visit)" AS TEXT) AS "Program ID (Program Visit) (forms)",

CAST("program visit forms"."Program ID (Program Visit)" AS TEXT) AS "Program ID (Program Visit)",

CAST("forms"."Program Name" AS TEXT) AS "Program Name",

CAST("forms"."Program Visit Form Type" AS TEXT) AS "Program Visit Form Type",

CAST("forms"."Service Form Abbreviation" AS TEXT) AS "Service Form Abbreviation",

CAST("forms"."Service Form Name" AS TEXT) AS "Service Form Name",

"clients"."System ID (Profile)" AS "System ID (Profile)",

"program visit forms"."System ID (Program Visit)" AS "System ID (Program Visit)",

"service forms"."System ID (Service)" AS "System ID (Service)"

FROM (

-- File: kdhe\_forms\_custom.sql

-- Date: 2017-09-22

-- Purpose: A junction query to join program visit forms to service forms

-- Forms: The table was generated from the following query.

-- WITH views AS (SELECT (regexp\_matches(definition, '\\(fd\\.form\_id = (\\d\*)\\)'))[1]::INTEGER AS "form\_id", schemaname || '.' || viewname AS "viewname" FROM pg\_catalog.pg\_views WHERE schemaname = 'kehs'), forms AS (SELECT DISTINCT form\_id, form\_name FROM kehs.form) SELECT forms.form\_id, forms.form\_name, views.viewname FROM forms LEFT JOIN views USING (form\_id) WHERE (form\_name ~ 'Service Form' OR form\_name ~ 'KDHE.\*Profile' OR form\_name ~ 'KDHE Program Visit Form') AND NOT (form\_name ~ 'Test' OR form\_name ~ 'Discharge') ORDER BY form\_id;

-- form\_id | form\_name | viewname

-- ---------+------------------------------------+----------------------------------------------

-- 28 | KDHE Adult Profile | kehs.kdhe\_adult\_profile\_view

-- 29 | KDHE Child Profile | kehs.kdhe\_child\_profile\_view

-- 30 | KDHE Program Visit Form - Adult | kehs.kdhe\_program\_visit\_form\_adult\_view

-- 31 | KDHE Program Visit Form - Child | kehs.kdhe\_program\_visit\_form\_child\_view

-- 32 | Family Planning Service Form | kehs.family\_planning\_service\_form\_view

-- 33 | Maternal Child Health Service Form | kehs.maternal\_child\_health\_service\_form\_view

-- 34 | TPTCM Service Form | kehs.tptcm\_service\_form\_view

-- 35 | Becoming a Mom\u00ae Service Form | kehs.becoming\_a\_mom\_service\_form\_view

-- 62 | PMI Service Form | kehs.pmi\_service\_form\_view

--

-- Usage: Use as the base of the join, called "forms".

-- - LEFT JOIN <kdhe\_clients\_custom.sql> clients ON

-- - forms."Form ID (Profile)" = clients."Form ID"

-- - LEFT JOIN <kdhe\_program\_visit\_forms\_custom.sql> "program visit forms" ON

-- - forms."Form ID (Program Visit)" = "program visit forms"."Form ID"

-- - forms."Program ID (Program Visit)" = "program visit forms"."Program ID (Program Visit)"

-- - LEFT JOIN <kdhe\_service\_forms\_custom.sql> "service forms"

-- - forms."Form ID (Service)" = "service forms"."Form ID"

--

-- - Perform the following additional joins between "program visit forms" and "service forms".

-- - "program visit forms" LEFT JOIN "service forms" ON

-- - "program visit forms"."System ID" = "service forms"."System ID"

-- - "program visit forms"."Date of Activity" = "service forms"."Date of Activity"

--

-- - Filter the results by:

-- - "Module" = 'KDHE'

-- - "program visit forms"."Form Data ID (Program Visit)" IS NOT NULL

--

WITH forms(

-- Related to Program Visit Forms

"Form ID (Profile)"

,"Form ID (Program Visit)" -- Join to a Program Visit form

,"Program Visit Form Type" -- Caregiver or Child

,"Program ID (Program Visit)" -- Join to a Program Visit form; Code from Q611 or Q1992

,"Program Name" -- Text from Q611 or Q1992

-- Related to Service Forms

,"Form ID (Service)" -- Join to a Service Form

,"Service Form Abbreviation"

,"Service Form Name"

) AS (VALUES

-- Q611: 1,Becoming a Mom|2,Family Planning|4,Maternal Child Health (MCH/M&I)|5,Pregnancy Maintenance (PMI)|6,Teen Pregnancy (TPTCM)

-- Q1992: Maternal Child Health (MCH/M&I)

----------------------------------------------------------------------------------------------------------------------------------------------

-- Form ID (Profile)

-- Form ID (Program Visit) Form ID (Service)

-- Program Visit Form Type Service Form Abbreviation

-- Program ID (Program Visit) Program Name Service Form Name

----------------------------------------------------------------------------------------------------------------------------------------------

(28, 30, 'Caregiver', '1' , 'Becoming a Mom\u00ae' , 35, 'BaM' , 'Becoming a Mom\u00ae Service Form' ),

(28, 30, 'Caregiver', '2' , 'Family Planning' , 32, 'FP' , 'Family Planning Service Form' ),

(28, 30, 'Caregiver', '4' , 'Maternal Child Health (MCH/M&I)', 33, 'MCH' , 'Maternal Child Health Service Form'),

(28, 30, 'Caregiver', '5' , 'Pregnancy Maintenance (PMI)' , 62, 'PMI' , 'PMI Service Form' ),

(28, 30, 'Caregiver', '6' , 'Teen Pregnancy (TPTCM)' , 34, 'TPTCM', 'TPTCM Service Form' ),

(29, 31, 'Child' , 'Maternal Child Health (MCH/M&I)', 'Maternal Child Health (MCH/M&I)', 33, 'MCH' , 'Maternal Child Health Service Form')

)

SELECT \* FROM forms

) "forms"

INNER JOIN (

-- File: kdhe\_clients\_custom.sql

-- Date: 2017-10-02

-- Purpose: To include all adults and children in one list

-- Forms: The table was generated from the following query.

-- WITH views AS (SELECT (regexp\_matches(definition, '\\(fd\\.form\_id = (\\d\*)\\)'))[1]::INTEGER AS "form\_id", schemaname || '.' || viewname AS "viewname" FROM pg\_catalog.pg\_views WHERE schemaname = 'kehs'), forms AS (SELECT DISTINCT form\_id, form\_name FROM kehs.form) SELECT forms.form\_id, forms.form\_name, views.viewname FROM forms LEFT JOIN views USING (form\_id) WHERE form\_name ~ 'KDHE.\*Profile' AND NOT (form\_name ~ 'Test') ORDER BY form\_id;

-- form\_id | form\_name | viewname

-- ---------+--------------------+------------------------------

-- 28 | KDHE Adult Profile | kehs.kdhe\_adult\_profile\_view

-- 29 | KDHE Child Profile | kehs.kdhe\_child\_profile\_view

--

-- Usage: Join on "System ID (Profile)" (preferably) or "DAISEY ID (Profile)" to "program visit forms" and "service forms"

-- - <kdhe\_program\_visit\_forms\_service\_forms\_custom.sql> "forms" LEFT JOIN <kdhe\_clients\_custom.sql> "clients"

-- ON "forms"."Form ID (Profile)" = "clients"."Form ID (Profile)"

-- Uncomment the following to test:

-- SELECT DISTINCT "System ID (Profile)" IS NULL, "DAISEY ID (Profile)" IS NULL, "Form ID (Profile)", "Client Type" FROM (

SELECT

module\_id AS "Module ID"

,module\_name AS "Module"

,grantee\_organization\_id AS "Grantee ID"

,grantee\_organization\_name AS "Grantee"

,organization\_id AS "Organization ID"

,organization\_name AS "Organization"

,form\_id AS "Form ID (Profile)"

,'Caregiver' AS "Client Type"

,form\_data\_id AS "System ID (Profile)"

,que1\_caregiver\_id\_494 AS "DAISEY ID (Profile)"

,que4\_first\_name\_234 AS "First Name"

,que5\_last\_name\_235 AS "Last Name"

FROM kehs.kdhe\_adult\_profile\_view

WHERE module\_id = 5 -- KDHE

UNION ALL

SELECT

module\_id AS "Module ID"

,module\_name AS "Module"

,grantee\_organization\_id AS "Grantee ID"

,grantee\_organization\_name AS "Grantee"

,organization\_id AS "Organization ID"

,organization\_name AS "Organization"

,form\_id AS "Form ID (Profile)"

,'Child' AS "Client Type"

,form\_data\_id AS "System ID (Profile)"

,que1\_child\_id\_495 AS "DAISEY ID (Profile)"

,que3\_first\_name\_216 AS "First Name"

,que4\_last\_name\_217 AS "Last Name"

FROM kehs.kdhe\_child\_profile\_view

WHERE module\_id = 5 -- KDHE

-- Uncomment the following to test:

-- ) dummy;

) "clients" ON ("forms"."Form ID (Profile)" = "clients"."Form ID (Profile)")

INNER JOIN (

-- File: kdhe\_program\_visit\_forms\_custom.sql

-- Date: 2017-10-02

-- Purpose: To include both adult and child service forms in one list

-- Forms: The table is generated from the following query.

-- WITH views AS (SELECT (regexp\_matches(definition, '\\(fd\\.form\_id = (\\d\*)\\)'))[1]::INTEGER AS "form\_id", schemaname || '.' || viewname AS "viewname" FROM pg\_catalog.pg\_views WHERE schemaname = 'kehs'), forms AS (SELECT DISTINCT form\_id, form\_name FROM kehs.form) SELECT forms.form\_id, forms.form\_name, views.viewname FROM forms LEFT JOIN views USING (form\_id) WHERE form\_name ~ 'KDHE Program Visit Form' ORDER BY form\_id;

-- form\_id | form\_name | viewname

-- ---------+---------------------------------+-----------------------------------------

-- 30 | KDHE Program Visit Form - Adult | kehs.kdhe\_program\_visit\_form\_adult\_view

-- 31 | KDHE Program Visit Form - Child | kehs.kdhe\_program\_visit\_form\_child\_view

--

-- Usage: - Join on "System ID (Program Visit)" (preferably) or "DAISEY ID (Program Visit)" to any service or profile forms.

-- - <kdhe\_program\_visit\_forms\_service\_forms\_custom.sql> "forms" LEFT JOIN <kdhe\_program\_visit\_forms\_custom.sql> "program visit forms"

-- ON "forms"."Form ID (Program Visit)" = "program visit forms"."Form ID (Program Visit)"

-- AND "forms"."Program ID" = "program visit forms"."Program ID (Program Visit)"

-- - <kdhe\_clients\_custom.sql> "clients" LEFT JOIN <kdhe\_program\_visit\_forms\_custom.sql> "program visit forms"

-- ON "clients"."System ID (Profile)" = "program visit forms"."System ID (Program Visit)"

--

-- Uncomment the following to test:

-- SELECT DISTINCT "Form ID (Program Visit)", "Program ID (Program Visit)", "System ID (Program Visit)" IS NULL, "DAISEY ID (Program Visit)" IS NULL, CASE WHEN "System ID (Program Visit)" IS NULL THEN "Form Data ID (Program Visit)" END FROM (

--

SELECT

form\_id AS "Form ID (Program Visit)"

,form\_data\_id AS "Form Data ID (Program Visit)"

,que2\_date\_of\_activity\_277 AS "Date of Activity (Program Visit)"

,que11\_program\_611 AS "Program ID (Program Visit)"

,CASE WHEN que1\_which\_caregiver\_was\_invol\_274 = ''

THEN NULL::BIGINT

ELSE que1\_which\_caregiver\_was\_invol\_274::BIGINT

END AS "System ID (Program Visit)"

,caregiver\_daisey\_id AS "DAISEY ID (Program Visit)"

FROM kehs.kdhe\_program\_visit\_form\_adult\_view

WHERE module\_id = 5 -- KDHE

UNION ALL

SELECT

form\_id AS "Form ID (Program Visit)"

,form\_data\_id AS "Form Data ID (Program Visit)"

,que2\_date\_of\_activity\_277 AS "Date of Activity (Program Visit)"

,que11\_program\_1992 AS "Program ID (Program Visit)"

,CASE WHEN que1\_which\_child\_was\_involved\_273 = ''

THEN NULL::BIGINT

ELSE que1\_which\_child\_was\_involved\_273::BIGINT

END AS "System ID (Program Visit)"

,child\_daisey\_id AS "DAISEY ID (Program Visit)"

FROM kehs.kdhe\_program\_visit\_form\_child\_view

WHERE module\_id = 5 -- KDHE

-- Uncomment the following to test:

-- ) dummy;

) "program visit forms" ON (("forms"."Form ID (Program Visit)" = "program visit forms"."Form ID (Program Visit)") AND ("forms"."Program ID (Program Visit)" = CAST("program visit forms"."Program ID (Program Visit)" AS TEXT)) AND ("clients"."System ID (Profile)" = "program visit forms"."System ID (Program Visit)"))

LEFT JOIN (

-- File: kdhe\_service\_forms\_custom.sql

-- Date: 2017-11-06

-- Purpose: To include all service forms in one list

-- Forms:

-- form\_id | form\_name | viewname

-- ---------+------------------------------------+----------------------------------------------

-- 32 | Family Planning Service Form | kehs.family\_planning\_service\_form\_view

-- 33 | Maternal Child Health Service Form | kehs.maternal\_child\_health\_service\_form\_view

-- 34 | TPTCM Service Form | kehs.tptcm\_service\_form\_view

-- 35 | Becoming a Mom\u00ae Service Form | kehs.becoming\_a\_mom\_service\_form\_view

-- 62 | PMI Service Form | kehs.pmi\_service\_form\_view

--

-- Usage: Join to

-- - kdhe\_program\_visit\_forms\_service\_forms\_custom ON

-- - "Form ID"

-- - kdhe\_program\_visit\_forms\_custom ON

-- - "System ID (Service)" (preferably) or "DAISEY ID (Service)"

-- - "Date of Activity (Service)"

--

-- Uncomment the following to test:

-- SELECT DISTINCT "Form ID (Service)", "System ID (Service)" IS NULL AS "Null System ID", "DAISEY ID (Service)" IS NULL AS "Null DAISEY ID", "Date of Activity (Service)" IS NULL AS "Null DoA", CASE WHEN "System ID (Service)" IS NULL THEN "Form Data ID (Service)" END AS "Form Data ID for Null System ID" FROM (

-- Family Planning

SELECT

form\_id AS "Form ID (Service)"

,form\_data\_id AS "Form Data ID (Service)"

,CASE WHEN que1\_which\_caregiver\_was\_invol\_274 = ''

THEN NULL::BIGINT

ELSE que1\_which\_caregiver\_was\_invol\_274::BIGINT

END AS "System ID (Service)"

,caregiver\_daisey\_id AS "DAISEY ID (Service)"

,que2\_date\_of\_activity\_277 AS "Date of Activity (Service)"

FROM kehs.family\_planning\_service\_form\_view

WHERE module\_id = 5 -- KDHE

UNION ALL

-- MCH

-- This one branches based off of que1\_visit\_for\_caregiveradult\_1184: 1,Caregiver/Adult|2,Child

SELECT

form\_id AS "Form ID (Service)"

,form\_data\_id AS "Form Data ID (Service)"

,CASE que1\_visit\_for\_caregiveradult\_1184

-- Caregiver/Adult

WHEN '1' THEN CASE WHEN que2\_which\_caregiver\_was\_invol\_274 = ''

THEN NULL::BIGINT

ELSE que2\_which\_caregiver\_was\_invol\_274::BIGINT

END

-- Child

WHEN '2' THEN CASE WHEN que4\_which\_child\_was\_involved\_273 = ''

THEN NULL::BIGINT

ELSE que4\_which\_child\_was\_involved\_273::BIGINT

END

END AS "System ID (Service)"

,CASE que1\_visit\_for\_caregiveradult\_1184

-- Caregiver/Adult

WHEN '1' THEN caregiver\_daisey\_id

-- Child

WHEN '2' THEN child\_daisey\_id

END AS "DAISEY ID (Service)"

,que5\_date\_of\_activity\_277 AS "Date of Activity (Service)"

FROM kehs.maternal\_child\_health\_service\_form\_view

WHERE module\_id = 5 -- KDHE

UNION ALL

-- TPTCM

SELECT

form\_id AS "Form ID (Service)"

,form\_data\_id AS "Form Data ID (Service)"

,CASE WHEN que1\_which\_caregiver\_was\_invol\_274 = ''

THEN NULL::BIGINT

ELSE que1\_which\_caregiver\_was\_invol\_274::BIGINT

END AS "System ID (Service)"

,caregiver\_daisey\_id AS "DAISEY ID (Service)"

,que2\_date\_of\_activity\_277 AS "Date of Activity (Service)"

FROM kehs.tptcm\_service\_form\_view

WHERE module\_id = 5 -- KDHE

UNION ALL

-- Becoming a Mom

SELECT

form\_id AS "Form ID (Service)"

,form\_data\_id AS "Form Data ID (Service)"

,CASE WHEN que1\_which\_caregiver\_was\_invol\_274 = ''

THEN NULL::BIGINT

ELSE que1\_which\_caregiver\_was\_invol\_274::BIGINT

END AS "System ID (Service)"

,caregiver\_daisey\_id AS "DAISEY ID (Service)"

,que2\_date\_of\_activity\_277 AS "Date of Activity (Service)"

FROM kehs.becoming\_a\_mom\_service\_form\_view

WHERE module\_id = 5 -- KDHE

UNION ALL

-- PMI

SELECT

form\_id AS "Form ID (Service)"

,form\_data\_id AS "Form Data ID (Service)"

,CASE WHEN que1\_which\_caregiver\_was\_invol\_274 = ''

THEN NULL::BIGINT

ELSE que1\_which\_caregiver\_was\_invol\_274::BIGINT

END AS "System ID (Service)"

,caregiver\_daisey\_id AS "DAISEY ID (Service)"

,que2\_date\_of\_activity\_277 AS "Date of Activity (Service)"

FROM kehs.pmi\_service\_form\_view

WHERE module\_id = 5 -- KDHE

-- Uncomment the following to test:

-- ) dummy;

) "service forms" ON (("forms"."Form ID (Service)" = "service forms"."Form ID (Service)") AND

("program visit forms"."Date of Activity (Program Visit)" = "service forms"."Date of Activity (Service)") AND

("program visit forms"."System ID (Program Visit)" = "service forms"."System ID (Service)"))

limit 1000