|  |  |
| --- | --- |
| **Date:** | **Nov 09, 2011** |
| **Remedy Request ID:** | **Not Applicable** |
| **Production Build ID:** | **P.1.114.0** |
| Build Description : |  |

**Section 1 –**

|  |  |  |  |
| --- | --- | --- | --- |
| Type of Build : | * Hot Fix | * Full |  |
| Database Team Responsible Party: | SACWIS DBA’s | | |

**Section 2 – Technical Solution**

|  |  |
| --- | --- |
| Location of Solution Documentation: | R:\sacwis\Database\DBAWorkArea\DBDevelopment\P.1.114.0 |
| Solution Documentation Folder Name: | Prod\Docs |
| Software/SQL File Name: | sacwis\_prod\_db\_p.1.114. |

|  |  |
| --- | --- |
| List new table(s)/View/MView be added: |  |
| List new column(s) be added / modified: | **Add 1 New Table(s)**  CREATE TABLE pdp\_incr\_ids  (person\_id NUMBER NOT NULL,  CONSTRAINT pdp\_incr\_ids\_p PRIMARY KEY (person\_id))  **Added 7 Columns to 7 Existing Table(s)**  ALTER TABLE intake\_participant\_acv ADD (prenatal\_drug\_abuse\_code VARCHAR2(30))  ALTER TABLE intake ADD (ai\_completion\_date DATE, ai\_completion\_action\_code VARCHAR2(30))  ALTER TABLE intake\_participant ADD (prefix\_code VARCHAR2(30))  ALTER TABLE allegation ADD (incident\_date DATE)  ALTER TABLE report\_template ADD (private\_agency\_viewable NUMBER DEFAULT 0 NOT NULL)  ALTER TABLE hmo\_data ADD odate DATE  ALTER TABLE hmo\_data\_header ADD odate DATE  **Modified 13 Existing Table(s)**  ALTER TABLE icpc\_icama MODIFY (supervising\_agency\_name VARCHAR2 (100))  ALTER TABLE pdp\_aka MODIFY (person\_id NUMBER)  ALTER TABLE pdp\_aka MOVE STORAGE (FREELIST GROUPS 3)  ALTER TABLE pdp\_aka NOPARALLEL  ALTER TABLE pdp\_event\_log  MODIFY (pdp\_event\_log\_id NUMBER)  MONITORING  DROP TABLE pdp\_incr\_aka CASCADE CONSTRAINTS;  CREATE TABLE pdp\_incr\_aka  (person\_id NUMBER NOT NULL,  first\_name VARCHAR2(35),  last\_name VARCHAR2(35))  DROP TABLE pdp\_incr\_match\_keys CASCADE CONSTRAINTS;  CREATE TABLE pdp\_incr\_match\_keys  (match\_key\_type VARCHAR2(1) NOT NULL,  match\_key\_value VARCHAR2(20) NOT NULL,  person\_id NUMBER NOT NULL,  CONSTRAINT pdp\_incr\_match\_keys\_p PRIMARY KEY (match\_key\_type, match\_key\_value, person\_id))  ORGANIZATION INDEX  TABLESPACE person\_t  PARTITION BY RANGE (match\_key\_type)  (PARTITION p\_ckey1 VALUES LESS THAN ('2'),  PARTITION p\_ckey2 VALUES LESS THAN ('3'),  PARTITION p\_firstname\_dm VALUES LESS THAN ('G'),  PARTITION p\_lastname\_dm VALUES LESS THAN ('M'),  PARTITION p\_firstname VALUES LESS THAN ('g'),  PARTITION p\_lastname VALUES LESS THAN ('m'),  PARTITION p\_ssn VALUES LESS THAN ('t'),  PARTITION p\_default VALUES LESS THAN (MAXVALUE)  DROP TABLE pdp\_incr\_person CASCADE CONSTRAINTS;  CREATE TABLE pdp\_incr\_person  (person\_id NUMBER NOT NULL,  first\_name VARCHAR2 (35),  last\_name VARCHAR2 (35),  ssn VARCHAR2 (9),  birth\_date DATE,  gender\_code VARCHAR2 (1),  race\_code VARCHAR2 (3),  created\_by NUMBER,  created\_date DATE,  owning\_agency\_id NUMBER,  CONSTRAINT pdp\_incr\_person\_p PRIMARY KEY (person\_id))  ALTER TABLE pdp\_key\_matches MODIFY (person\_id1 NUMBER, person\_id2 NUMBER)  ALTER TABLE pdp\_key\_matches MOVE STORAGE (FREELIST GROUPS 3)  ALTER TABLE pdp\_match\_keys MODIFY (person\_id NUMBER);  ALTER TABLE pdp\_match\_keys LOGGING;  ALTER TABLE pdp\_match\_keys MOVE PARTITION p\_ckey1 STORAGE (FREELIST GROUPS 3);  ALTER TABLE pdp\_match\_keys MOVE PARTITION p\_ckey2 STORAGE (FREELIST GROUPS 3);  ALTER TABLE pdp\_match\_keys MOVE PARTITION p\_ssn STORAGE (FREELIST GROUPS 3);  ALTER TABLE pdp\_match\_keys MOVE PARTITION p\_firstname STORAGE (FREELIST GROUPS 3);  ALTER TABLE pdp\_match\_keys MOVE PARTITION p\_firstname\_dm STORAGE (FREELIST GROUPS 3);  ALTER TABLE pdp\_match\_keys MOVE PARTITION p\_lastname STORAGE (FREELIST GROUPS 3);  ALTER TABLE pdp\_match\_keys MOVE PARTITION p\_lastname\_dm STORAGE (FREELIST GROUPS 3);  ALTER TABLE pdp\_match\_keys MOVE PARTITION p\_default STORAGE (FREELIST GROUPS 3)  ALTER TABLE pdp\_message\_log  MODIFY (pdp\_event\_log\_id NUMBER,  pdp\_message\_log\_id NUMBER)  ALTER TABLE pdp\_person  MODIFY (person\_id NUMBER,  created\_by NUMBER NOT NULL,  created\_date NOT NULL,  owning\_agency\_id NUMBER)  ALTER TABLE pdp\_race\_map NOMONITORING  ALTER TABLE pdp\_results1 MONITORING;  ALTER TABLE pdp\_results1 MOVE STORAGE (FREELIST GROUPS 3)  ALTER TABLE pdp\_results2 MONITORING;  ALTER TABLE pdp\_results2 MOVE STORAGE (FREELIST GROUPS 3) |
| List existing table(s) to be deleted: | **Droped 2 Table(s)**  DROP TABLE pdp\_owning\_agency CASCADE CONSTRAINTS  DROP TABLE pdp\_sql\_log CASCADE CONSTRAINTS |
| List existing column(s) to be deleted: | **Deleted Columns from 2 Existing Table(s)**  ALTER TABLE reporter  DROP (src\_of\_knldg\_observed\_flag,  src\_of\_knldg\_other\_party\_flag,  observerd\_or\_told\_date,  other\_party\_name\_contact,  addrd\_con\_parent\_ct\_flag,  addrd\_con\_parent\_ct\_desc,  last\_rptr\_contact\_with\_acvs,  oths\_name\_contact\_knldg  )  ALTER TABLE intake DROP (post\_decision\_narrative\_text) |

|  |  |
| --- | --- |
| Application Ref Data / Seed Data | **Some Ref & Seed Data Updates and Modifies:**  **Adds:**  **REF\_DATA**  INSERT INTO ref\_data  (ref\_data\_id, parent\_ref\_data\_code, domain\_code, ref\_data\_code, short\_desc,  long\_desc, sort\_value, inactive\_flag, created\_by, created\_date, modified\_by, modified\_date, ref\_data\_config\_id,  group\_code  )  VALUES (ref\_data\_id\_seq.NEXTVAL, NULL, 'SunsetActivity', 'CASEPLANAPPROVALDATE', 'Case Plan Approval Date',  'Case Plan Approval Date', 1, 0, 999, SYSDATE, 999, SYSDATE, 8024,  '00'  )  INSERT INTO ref\_data  (ref\_data\_id, parent\_ref\_data\_code, domain\_code, ref\_data\_code, short\_desc,  long\_desc, sort\_value, inactive\_flag, created\_by, created\_date, modified\_by, modified\_date, ref\_data\_config\_id,  group\_code  )  VALUES (ref\_data\_id\_seq.NEXTVAL, NULL, 'SunsetActivity', 'CASEPLANCOURTSTATUSDATE', 'Case Plan Court Status Date',  'Case Plan Court Status Date', 1, 0, 999, SYSDATE, 999, SYSDATE, 8024,  '00'  )  INSERT INTO ref\_data (ref\_data\_id,  parent\_ref\_data\_code,  domain\_code,  ref\_data\_code,  short\_desc,  long\_desc,  sort\_value,  inactive\_flag,  created\_by,  created\_date,  modified\_by,  modified\_date,  ref\_data\_config\_id,  group\_code)  VALUES (ref\_data\_id\_seq.NEXTVAL,  NULL,  'Relationship',  'LEGALFATHER',  'Legal Father',  'Legal Father',  1,  0,  pkAuditUser.GetBuildUser('1.114'),  SYSDATE,  pkAuditUser.GetBuildUser('1.114'),  SYSDATE,  (SELECT ref\_data\_config\_id  FROM ref\_data\_config  WHERE category\_name = 'relationshipType'),  '00')  INSERT INTO ref\_data (ref\_data\_id,  parent\_ref\_data\_code,  domain\_code,  ref\_data\_code,  short\_desc,  long\_desc,  sort\_value,  inactive\_flag,  created\_by,  created\_date,  modified\_by,  modified\_date,  ref\_data\_config\_id,  group\_code)  VALUES (ref\_data\_id\_seq.NEXTVAL,  NULL,  'Relationship',  'LEGALSON',  'Legal Son',  'Legal Son',  1,  0,  pkAuditUser.GetBuildUser('1.114'),  SYSDATE,  pkAuditUser.GetBuildUser('1.114'),  SYSDATE,  (SELECT ref\_data\_config\_id  FROM ref\_data\_config  WHERE category\_name = 'relationshipType'),  '00')  INSERT INTO ref\_data (ref\_data\_id,  parent\_ref\_data\_code,  domain\_code,  ref\_data\_code,  short\_desc,  long\_desc,  sort\_value,  inactive\_flag,  created\_by,  created\_date,  modified\_by,  modified\_date,  ref\_data\_config\_id,  group\_code)  VALUES (ref\_data\_id\_seq.NEXTVAL,  NULL,  'Relationship',  'LEGALDAUGHTER',  'Legal Daughter',  'Legal Daughter',  1,  0,  pkAuditUser.GetBuildUser('1.114'),  SYSDATE,  pkAuditUser.GetBuildUser('1.114'),  SYSDATE,  (SELECT ref\_data\_config\_id  FROM ref\_data\_config  WHERE category\_name = 'relationshipType'),  '00')  INSERT INTO ref\_data (ref\_data\_id,  parent\_ref\_data\_code,  domain\_code,  ref\_data\_code,  short\_desc,  long\_desc,  sort\_value,  inactive\_flag,  created\_by,  created\_date,  modified\_by,  modified\_date,  ref\_data\_config\_id,  group\_code)  VALUES (ref\_data\_id\_seq.NEXTVAL,  NULL,  'Relationship',  'LEGALCHILD',  'Legal Child',  'Legal Child',  1,  0,  pkAuditUser.GetBuildUser('1.114'),  SYSDATE,  pkAuditUser.GetBuildUser('1.114'),  SYSDATE,  (SELECT ref\_data\_config\_id  FROM ref\_data\_config  WHERE category\_name = 'relationshipType'),  '00')  NSERT INTO ref\_data  (ref\_data\_id, parent\_ref\_data\_code, domain\_code, ref\_data\_code, short\_desc,  long\_desc, sort\_value, inactive\_flag, created\_by, created\_date, modified\_by,  modified\_date, ref\_data\_config\_id, group\_code  )  VALUES (ref\_data\_id\_seq.NEXTVAL, NULL, 'AICompletionActionCode', 'Family Assessment', 'Family Assessment',  'Family Assessment', 1, 0, pkaudituser.getbuilduser ('1.114'), SYSDATE, pkaudituser.getbuilduser ('1.114'),  SYSDATE, (SELECT ref\_data\_config\_id  FROM ref\_data\_config  WHERE category\_name = 'AICompletionActionCode'), '00'  )  INSERT INTO ref\_data  (ref\_data\_id, parent\_ref\_data\_code, domain\_code, ref\_data\_code, short\_desc,  long\_desc, sort\_value, inactive\_flag, created\_by, created\_date, modified\_by,  modified\_date, ref\_data\_config\_id, group\_code  )  VALUES (ref\_data\_id\_seq.NEXTVAL, NULL, 'AICompletionActionCode', 'Ongoing Assessment', 'Ongoing Assessment',  'Ongoing Assessment', 2, 0, pkaudituser.getbuilduser ('1.114'), SYSDATE, pkaudituser.getbuilduser ('1.114'),  SYSDATE, (SELECT ref\_data\_config\_id  FROM ref\_data\_config  WHERE category\_name = 'AICompletionActionCode'), '00'  )  INSERT INTO ref\_data  (ref\_data\_id, parent\_ref\_data\_code, domain\_code, ref\_data\_code, short\_desc,  long\_desc, sort\_value, inactive\_flag, created\_by, created\_date, modified\_by,  modified\_date, ref\_data\_config\_id, group\_code  )  VALUES (ref\_data\_id\_seq.NEXTVAL, NULL, 'AICompletionActionCode', 'Specialized Assessment', 'Specialized Assessment',  'Specialized Assessment', 3, 0, pkaudituser.getbuilduser ('1.114'), SYSDATE, pkaudituser.getbuilduser ('1.114'),  SYSDATE, (SELECT ref\_data\_config\_id  FROM ref\_data\_config  WHERE category\_name = 'AICompletionActionCode'), '00'  )  INSERT INTO ref\_data  (ref\_data\_id, parent\_ref\_data\_code, domain\_code, ref\_data\_code, short\_desc,  long\_desc, sort\_value, inactive\_flag, created\_by, created\_date, modified\_by,  modified\_date, ref\_data\_config\_id, group\_code  )  VALUES (ref\_data\_id\_seq.NEXTVAL, NULL, 'AICompletionActionCode', 'Justification/Waiver', 'Justification/Waiver',  'Justification/Waiver', 4, 0, pkaudituser.getbuilduser ('1.114'), SYSDATE, pkaudituser.getbuilduser ('1.114'),  SYSDATE, (SELECT ref\_data\_config\_id  FROM ref\_data\_config  WHERE category\_name = 'AICompletionActionCode'), '00'  )  INSERT INTO ref\_data  (ref\_data\_id, parent\_ref\_data\_code, domain\_code, ref\_data\_code, short\_desc, long\_desc, sort\_value, inactive\_flag,  created\_by, created\_date, modified\_by, modified\_date,  ref\_data\_config\_id, group\_code  )  VALUES (ref\_data\_id\_seq.NEXTVAL, NULL, 'AICompletionActionCode', 'Disposition', 'Disposition', 'Disposition', 5, 0,  pkaudituser.getbuilduser ('1.114'), SYSDATE, pkaudituser.getbuilduser ('1.114'), SYSDATE,  (SELECT ref\_data\_config\_id  FROM ref\_data\_config  WHERE category\_name = 'AICompletionActionCode'), '00'  )  INSERT INTO ref\_data  (ref\_data\_id, parent\_ref\_data\_code, domain\_code, ref\_data\_code, short\_desc,  long\_desc, sort\_value, inactive\_flag, created\_by, created\_date, modified\_by,  modified\_date, ref\_data\_config\_id, group\_code  )  VALUES (ref\_data\_id\_seq.NEXTVAL, NULL, 'AICompletionActionCode', 'Case Activity Log', 'Case Activity Log',  'Case Activity Log', 6, 0, pkaudituser.getbuilduser ('1.114'), SYSDATE, pkaudituser.getbuilduser ('1.114'),  SYSDATE, (SELECT ref\_data\_config\_id  FROM ref\_data\_config  WHERE category\_name = 'AICompletionActionCode'), '00'  )  INSERT INTO ref\_data  (ref\_data\_id, parent\_ref\_data\_code, domain\_code, ref\_data\_code, short\_desc, long\_desc, sort\_value, inactive\_flag,  created\_by, created\_date, modified\_by, modified\_date,  ref\_data\_config\_id, group\_code  )  VALUES (ref\_data\_id\_seq.NEXTVAL, NULL, 'UnitName', '#', 'Number', 'Number', 1, 0,  pkaudituser.getbuilduser ('1.114'), SYSDATE, pkaudituser.getbuilduser ('1.114'), SYSDATE,  (SELECT ref\_data\_config\_id  FROM ref\_data\_config  WHERE category\_name = 'unitNameType'), '00'  )  INSERT INTO ref\_data  (ref\_data\_id, parent\_ref\_data\_code, domain\_code, ref\_data\_code,  short\_desc,  long\_desc,  sort\_value, inactive\_flag, created\_by, created\_date, modified\_by, modified\_date,  ref\_data\_config\_id, group\_code  )  VALUES (ref\_data\_id\_seq.NEXTVAL, NULL, 'PrenatalDrugAbuseCode', 'AFFECTEDBY',  'An infant identified as affected by illegal substance abuse...',  'An infant identified as affected by illegal substance abuse/ having withdrawal symptoms resulting from prenatal drug exposure.',  1, 0, pkaudituser.getbuilduser ('1.114'), SYSDATE, pkaudituser.getbuilduser ('1.114'), SYSDATE,  (SELECT ref\_data\_config\_id  FROM ref\_data\_config  WHERE category\_name = 'PrenatalDrugAbuseCode'), '00'  )  INSERT INTO ref\_data  (ref\_data\_id, parent\_ref\_data\_code, domain\_code, ref\_data\_code,  short\_desc,  long\_desc, sort\_value, inactive\_flag, created\_by, created\_date,  modified\_by, modified\_date, ref\_data\_config\_id, group\_code  )  VALUES (ref\_data\_id\_seq.NEXTVAL, NULL, 'PrenatalDrugAbuseCode', 'FETALALCOHOL',  'An infant diagnosed with Fetal Alcohol Spectrum Disorder.',  'An infant diagnosed with Fetal Alcohol Spectrum Disorder.', 2, 0, pkaudituser.getbuilduser ('1.114'), SYSDATE,  pkaudituser.getbuilduser ('1.114'), SYSDATE, (SELECT ref\_data\_config\_id  FROM ref\_data\_config  WHERE category\_name = 'PrenatalDrugAbuseCode'), '00'  )  INSERT INTO ref\_data  (ref\_data\_id, parent\_ref\_data\_code, domain\_code, ref\_data\_code,  short\_desc,  long\_desc,  sort\_value, inactive\_flag, created\_by, created\_date, modified\_by, modified\_date,  ref\_data\_config\_id, group\_code  )  VALUES (ref\_data\_id\_seq.NEXTVAL, NULL, 'PrenatalDrugAbuseCode', 'BOTH',  'An infant identified as affected by ... AND diagnosed with Fetal Alcohol Spectrum Disorder.',  'An infant identified as affected by illegal substance abuse/ having withdrawal symptoms resulting from prenatal drug exposure ' ||  'AND diagnosed with Fetal Alcohol Spectrum Disorder.',  3, 0, pkaudituser.getbuilduser ('1.114'), SYSDATE, pkaudituser.getbuilduser ('1.114'), SYSDATE,  (SELECT ref\_data\_config\_id  FROM ref\_data\_config  WHERE category\_name = 'PrenatalDrugAbuseCode'), '00'  )  INSERT INTO ref\_data (ref\_data\_id,  parent\_ref\_data\_code,  domain\_code,  ref\_data\_code,  short\_desc,  long\_desc,  sort\_value,  inactive\_flag,  created\_by,  created\_date,  modified\_by,  modified\_date,  ref\_data\_config\_id,  group\_code)  VALUES (ref\_data\_id\_seq.NEXTVAL,  NULL,  'HarmDescription',  'INFAFFECTED',  'Infant Affected by Illegal Substance Abuse',  'Infant Affected by Illegal Substance Abuse',  1,  0,  pkAuditUser.GetBuildUser('1.114'),  SYSDATE,  pkAuditUser.GetBuildUser('1.114'),  SYSDATE,  (SELECT ref\_data\_config\_id  FROM ref\_data\_config  WHERE category\_name = 'HarmDescription'),  '00')  INSERT INTO ref\_data (ref\_data\_id,  parent\_ref\_data\_code,  domain\_code,  ref\_data\_code,  short\_desc,  long\_desc,  sort\_value,  inactive\_flag,  created\_by,  created\_date,  modified\_by,  modified\_date,  ref\_data\_config\_id,  group\_code)  VALUES (ref\_data\_id\_seq.NEXTVAL,  NULL,  'HarmDescription',  'WITHDRAWAL',  'Withdrawal Symptoms due to Prenatal Drug Exposure',  'Withdrawal Symptoms due to Prenatal Drug Exposure',  1,  0,  pkAuditUser.GetBuildUser('1.114'),  SYSDATE,  pkAuditUser.GetBuildUser('1.114'),  SYSDATE,  (SELECT ref\_data\_config\_id  FROM ref\_data\_config  WHERE category\_name = 'HarmDescription'),  '00')  **REF\_DATA\_CONFIG**  INSERT INTO ref\_data\_config  (ref\_data\_config\_id, category\_name, domain\_code, parent\_domain\_code, has\_child\_flag,  sql\_statement,  class\_text, category\_desc, state\_exclusive\_flag, created\_by, created\_date,  modified\_by, modified\_date  )  VALUES (ref\_data\_config\_id\_seq.NEXTVAL, 'AICompletionActionCode', 'AICompletionActionCode', NULL, NULL,  'SELECT ref\_data\_code AS "lookup\_code", ' ||  'short\_desc AS "short\_description", ' ||  'long\_desc AS "long\_description" ' ||  'FROM ref\_data ' ||  'WHERE DOMAIN\_CODE = ''AICompletionActionCode''',  'us.oh.state.odjfs.sacwis.framework.rda.ReferenceVO', NULL, 0, pkaudituser.getbuilduser ('1.114'), SYSDATE,  pkaudituser.getbuilduser ('1.114'), SYSDATE  )  INSERT INTO ref\_data\_config  (ref\_data\_config\_id, category\_name, domain\_code, parent\_domain\_code, has\_child\_flag,  sql\_statement,  class\_text, category\_desc, state\_exclusive\_flag, created\_by, created\_date,  modified\_by, modified\_date  )  VALUES (ref\_data\_config\_id\_seq.NEXTVAL, 'PrenatalDrugAbuseCode', 'PrenatalDrugAbuseCode', NULL, NULL,  'SELECT ref\_data\_code AS "lookup\_code", ' ||  'short\_desc AS "short\_description", ' ||  'long\_desc AS "long\_description" ' ||  'FROM ref\_data ' ||  'WHERE DOMAIN\_CODE = ''PrenatalDrugAbuseCode''',  'us.oh.state.odjfs.sacwis.framework.rda.ReferenceVO', NULL, 0, pkaudituser.getbuilduser ('1.114'), SYSDATE,  pkaudituser.getbuilduser ('1.114'), SYSDATE    **ROLE**  INSERT INTO role  (role\_id, role\_name, role\_code, workitem\_assignable\_flag, state\_only\_role\_flag, public\_agency\_only\_role\_flag,  created\_by, created\_date, modified\_by, modified\_date  )  VALUES (role\_id\_seq.NEXTVAL, 'Case Reviewer', 'CASEREVIEWER', 1, 0, 1,  0, SYSDATE, 0, SYSDATE  )  **SECURITY\_PROFILE**  INSERT INTO security\_profile  (security\_profile\_id, role\_id, business\_function\_id,  profile\_name, profile\_desc, state\_exclusive\_flag, public\_only\_flag, created\_by, created\_date, modified\_by,  modified\_date, asgn\_rqd\_read\_prv\_exmpt\_flag, asgn\_rqd\_updt\_prv\_exmpt\_flag, asgn\_rqd\_read\_pub\_exmpt\_flag,  asgn\_rqd\_updt\_pub\_exmpt\_flag  )  VALUES (security\_profile\_id\_seq.NEXTVAL, (SELECT role\_id  FROM role  WHERE role\_code = 'CASEREVIEWER'), (SELECT business\_function\_id  FROM business\_function  WHERE business\_function\_code = 'CM41'),  'Case Review/SAR - Case Reviewer', 'Case Review/SAR - Case Reviewer', 0, 0, 100, SYSDATE, 100,  SYSDATE, 0, 0, 1,  1  )  INSERT INTO security\_profile  VALUES (security\_profile\_id\_seq.NEXTVAL, 10003, business\_function\_id\_seq.CURRVAL, 'Admin Reports (Private User)',  'Private Agency Worker access to Manage Standard Reports (CF17a)', 0, 0, 100, SYSDATE, 100, SYSDATE, 0, 0, 0, 0)  **ACTION\_ROLE**  INSERT INTO action\_role  (uc\_action\_id,  role\_id, action\_role\_id  )  VALUES ((SELECT uc\_action\_id  FROM uc\_action  WHERE uc\_action\_code = 'CM41\_MAINTAIN\_CASE\_REVIEW'  AND action\_mode = 'UPDATE'  AND business\_function\_id = (SELECT business\_function\_id  FROM business\_function  WHERE business\_function\_code = 'CM41')),  (SELECT role\_id  FROM role  WHERE role\_code = 'CASEREVIEWER'), action\_role\_id\_seq.NEXTVAL  );  INSERT INTO action\_role  (uc\_action\_id,  role\_id, action\_role\_id  )  VALUES ((SELECT uc\_action\_id  FROM uc\_action  WHERE uc\_action\_code = 'CM41\_VIEW\_CASE\_REVIEW'  AND action\_mode = 'VIEW'  AND business\_function\_id = (SELECT business\_function\_id  FROM business\_function  WHERE business\_function\_code = 'CM41')),  (SELECT role\_id  FROM role  WHERE role\_code = 'CASEREVIEWER'), action\_role\_id\_seq.NEXTVAL  )  **PROFILE\_RESOURCE**  INSERT INTO profile\_resource  SELECT profile\_resource\_id\_seq.NEXTVAL, NULL, business\_function\_id, menu\_id, 100, SYSDATE, 100, SYSDATE  FROM menu m, business\_function bf  WHERE LOWER (menu\_item) LIKE 'home%' AND bf.business\_function\_code = 'CF36'  INSERT INTO profile\_resource  (profile\_resource\_id, business\_function\_id, menu\_id, created\_by, created\_date, modified\_by, modified\_date)  SELECT profile\_resource\_id\_seq.NEXTVAL, business\_function\_id\_seq.CURRVAL, menu\_id, 100, SYSDATE, 100, SYSDATE  FROM menu  WHERE menu\_item IN ('Administration', 'Administration.Reports')  **BUSINESS\_FUNCTION**  INSERT INTO business\_function  SELECT business\_function\_id\_seq.NEXTVAL, 'Admin Reports (Private)', 'CF17a', business\_function\_desc,  business\_function\_cat\_code, assign\_rqd\_read\_priv\_flag, assign\_rqd\_update\_priv\_flag, assign\_rqd\_read\_pub\_flag,  assign\_rqd\_update\_pub\_flag, status\_code, 100, SYSDATE, 100, SYSDATE  FROM business\_function  WHERE business\_function\_code = 'CF17'  **REF\_DATA\_CONFIG**  **UC\_ACTION**  **MENU**  **REPORT\_TEMPLATE**  **Changes:**  **MENU**  UPDATE menu  SET menu\_item = 'reportSearchInit.do'  WHERE menu\_item = 'Administration.Reports'  **REF\_DATA**  UPDATE ref\_data  SET inactive\_flag = 1,  modified\_by = 999,  modified\_date = SYSDATE  WHERE ref\_data\_code = 'DATEOFAGENCYWORKERSIGNATUREONC'  UPDATE ref\_data  SET ref\_data\_code = 'Aly',  modified\_by = pkaudituser.getbuilduser ('1.114'),  modified\_date = SYSDATE  WHERE domain\_code = 'StreetSuffix' AND ref\_data\_code = 'AL  UPDATE ref\_data  SET ref\_data\_code = 'Cmns',  modified\_by = pkaudituser.getbuilduser ('1.114'),  modified\_date = SYSDATE  WHERE domain\_code = 'StreetSuffix' AND ref\_data\_code = 'CMN'  UPDATE ref\_data  SET ref\_data\_code = 'Po Box',  modified\_by = pkaudituser.getbuilduser ('1.114'),  modified\_date = SYSDATE  WHERE domain\_code = 'UnitName' AND ref\_data\_code = 'POBOX'  UPDATE ref\_data  SET short\_desc = 'Positive Toxicology of Infant/Child',  long\_desc = 'Positive Toxicology of Infant/Child',  modified\_by = pkAuditUser.GetBuildUser('1.114'),  modified\_date = SYSDATE  WHERE domain\_code = 'HarmDescription'  AND ref\_data\_code = 'POSTOXOFCHILD'  **BUSINESS\_FUNCTION**  UPDATE business\_function bf  SET assign\_rqd\_update\_pub\_flag = 1  WHERE bf.business\_function\_code = 'CM41'  **REPORT\_TEMPLATE**  UPDATE report\_template  SET private\_agency\_viewable = 1,  modified\_by = 999,  modified\_date = SYSDATE  WHERE report\_template\_code IN ('Rpt313', 'Rpt337', 'Rpt150')  **MENU**  UPDATE menu  SET menu\_item = 'reportSearchInit.do'  WHERE menu\_item = 'Administration.Reports'  **GROUP\_CATEGORY**  **ROLE / SECURITY\_PROFILE**  **UC\_ACTION**  **ACTION\_ROLE**  **SECURITY\_PROFILE**  **PROFILE\_RESOURCE**  **GROUP\_CAT\_CEIL\_TYP\_LINK**  **TICKLER\_TEMPLATE**   1. **Deletes:** |
| Batch Ref Data / Seed Data Update |  |
| Materialized View |  |
| Sequence | **Added New Sequence(s) Overall:**  **Modified 0 Sequence(s) Overall:** |

|  |  |
| --- | --- |
| Index | **Add 1 New Index(es) Overall:**  CREATE INDEX pdp\_event\_log\_n01 ON pdp\_event\_log (created\_date)  **Modified Index(es) Overall:**  **Dropped Index(es) Overall:** |
| Constraint | **Added New and Modified 1 Existing Constraint(s):**  **Modified 1 Constraint(s) Overall:**  ALTER TABLE intake\_participant  MODIFY unknown\_flag CONSTRAINT intake\_part\_unknown\_flag\_nn NOT NULL  **Dropped 0 Constraint(s) Overall:** |
|  |  |
| Function Updated | **Added New & Modified 1 Existing Function(s) & Dropped 0 Functions Overall:**  **New Function(s):**      **Updated 1 Existing Function(s):**   1. GETCASETRIGGERACTIVITYINFO   **Dropped 0 Unused Function(s) :** |
| Package(s) Updated | **Added 2 New & Modified 5 Package(s) & Dropped 0 Packages Overall:**  **2 New Package(s):**   1. FM25a\_MEDICAID\_HMO\_MITS 2. PKGINTAKECOMPLETIONSTATE   **Updated 5 Existing Package(s):**   1. PKG\_NON\_ODJFS\_PRVDR\_MERGE 2. PKG\_DUP\_PERSON\_MATCH 3. FM25a\_MITS\_HMO\_ERRORS 4. FM25a\_MEDICAID\_HMO\_DATA\_LOADER 5. DSBRS\_WARRANT\_EFT\_TO\_AUDITOR   **Dropped 0 Unused Package(s) :** |
| Procedure to be compiled | **Added 1 New & Modified Existing Procedure(s) & Dropped 0 Existing Overall:**   1. SPRECORDAICOMPLETION   **Updated Existing Procedure:**  **Dropped 0 Unused Procedure(s) :** |
|  |  |

|  |  |
| --- | --- |
| Triggers to be updated and enabled | **Added 5 New and Modified 32 Existing Trigger(s) Overall:**  **Added 5 New Trigger(s):**   1. INTAKE\_COMPL\_CODE\_VALIDATE 2. INTAKE\_PART\_PREFIX\_VALIDATE 3. INTAKE\_PART\_PRENATAL\_VALIDATE 4. PARTY\_ADDRESS\_PRIMARY\_VALIDATE 5. AFTER\_RTH\_CHANGES   **Modified 32 Existing Trigger(s):**   1. ALLEGATION\_AUDR 2. ICPC\_ICAMA\_AUDR 3. INTAKE\_AUDR 4. INTAKE\_PARTICIPANT\_AUDR 5. INTAKE\_PARTICIPANT\_ACV\_AUDR 6. REPORTER\_AUDR 7. REPORT\_TEMPLATE\_AUDR 8. HMO\_DATA\_INSRT 9. BEFORE\_RTH\_CHANGES 10. RTH\_APPROVE\_FA 11. RTH\_APPROVE\_OA 12. RTH\_APPROVE\_SPA 13. RTH\_APPROVE\_JUST 14. RTH\_REMOVE\_APPROVAL\_FA 15. RTH\_REMOVE\_APPROVAL\_OA 16. RTH\_REMOVE\_APPROVAL\_SPA 17. RTH\_REMOVE\_APPROVAL\_JUST 18. RTH\_UPDATE\_APPROVAL\_FA 19. RTH\_UPDATE\_APPROVAL\_OA 20. RTH\_UPDATE\_APPROVAL\_SPA 21. RTH\_UPDATE\_APPROVAL\_JUST 22. BEFORE\_AI\_LINK\_CHANGES 23. DELETE\_AI\_INTAKE\_LINK 24. BEFORE\_JUSTIFICATION\_CHANGES 25. AFTER\_JUSTIFICATION\_CHANGES 26. DELETE\_JUSTIFICATION 27. UPDATE\_JUSTIFICATION 28. BEFORE\_JUST\_TYPE\_CHANGES 29. AFTER\_JUST\_TYPE\_CHANGES 30. INSERT\_JUSTIFICATION\_TYPE 31. DELETE\_JUSTIFICATION\_TYPE 32. UPDATE\_JUSTIFICATION\_TYPE |
| Script Updated | UPDATE intake\_participant SET unknown\_flag = 0 WHERE unknown\_flag IS NULL  UPDATE address  SET street\_suffix\_code = 'Aly'  WHERE street\_suffix\_code = 'AL'  UPDATE address  SET street\_suffix\_code = 'Cmns'  WHERE street\_suffix\_code = 'CMN'  UPDATE address  SET unit\_name\_code = 'Po Box'  WHERE unit\_name\_code = ANY ('POBOX', 'PO BOX', 'PO Box', 'Box')  CREATE TABLE temp\_datafix\_14021\_p  PCTFREE 0  TABLESPACE sacwis\_t AS  SELECT DISTINCT pr.provider\_id  FROM provider pr, provider\_type pt, agency ag  WHERE pr.provider\_category\_code = 'HOME'  AND pt.provider\_id = pr.provider\_id  AND pt.end\_date IS NULL  AND pt.agency\_id = ag.agency\_id  AND ag.public\_agency\_flag = 0;  ALTER TABLE temp\_datafix\_14021\_p ADD (has\_public\_type VARCHAR2(1) NULL);  ALTER TABLE temp\_datafix\_14021\_p ADD (has\_current\_assignments VARCHAR2(1) NULL);  ALTER TABLE temp\_datafix\_14021\_p ADD (workload\_item\_id NUMBER NULL);  UPDATE temp\_datafix\_14021\_p tdf  SET tdf.has\_public\_type = 'Y'  WHERE EXISTS (SELECT 1  FROM provider\_type pt, agency ag  WHERE pt.provider\_id = tdf.provider\_id  AND pt.end\_date IS NULL  AND pt.agency\_id = ag.agency\_id  AND ag.public\_agency\_flag = 1);  UPDATE temp\_datafix\_14021\_p tdl  SET tdl.workload\_item\_id = (SELECT workload\_item\_id  FROM workload\_item  WHERE work\_item\_id = tdl.provider\_id  AND work\_item\_type\_code = 'PROVIDER');  UPDATE temp\_datafix\_14021\_p tdf  SET tdf.has\_current\_assignments = 'Y'  WHERE tdf.workload\_item\_id IS NOT NULL  AND EXISTS (SELECT 1  FROM work\_assignment  WHERE workload\_item\_id = tdf.workload\_item\_id  AND end\_date IS NULL);  CREATE TABLE temp\_datafix\_14021\_work  PCTFREE 0  TABLESPACE sacwis\_t AS  SELECT \*  FROM work\_assignment  WHERE workload\_item\_id IN  (SELECT workload\_item\_id  FROM temp\_datafix\_14021\_p  WHERE has\_public\_type IS NULL  AND has\_current\_assignments = 'Y')  AND end\_date IS NULL;  UPDATE work\_assignment  SET end\_date = TRUNC (SYSDATE),  modified\_by = 999,  modified\_date = SYSDATE  WHERE workload\_item\_id IN (SELECT workload\_item\_id  FROM temp\_datafix\_14021\_p  WHERE has\_public\_type IS NULL  AND has\_current\_assignments = 'Y')  AND end\_date IS NULL  CREATE TABLE sacwis\_fix.defect\_13662\_sum\_wksht  PCTFREE 0  TABLESPACE sacwis\_t AS  (SELECT \* FROM inc\_res\_summary\_wksht WHERE 1=2);  --DROP TABLE sacwis\_fix.defect\_13662\_wksht\_detail;  CREATE TABLE sacwis\_fix.defect\_13662\_wksht\_detail  PCTFREE 0  TABLESPACE sacwis\_t AS  (SELECT \* FROM inc\_res\_wksht\_detail WHERE 1=2);  DECLARE  CURSOR mainquery IS  SELECT eligible\_member\_id --,count(distinct det.INC\_RES\_SUMMARY\_WKSHT\_ID)  FROM inc\_res\_summary\_wksht inc INNER JOIN inc\_res\_wksht\_detail det  ON det.inc\_res\_summary\_wksht\_id = inc.inc\_res\_summary\_wksht\_id  GROUP BY inc.eligible\_member\_id  HAVING COUNT (DISTINCT det.inc\_res\_summary\_wksht\_id) > 1;  CURSOR subquery (eligiblememberid NUMBER) IS  SELECT eligible\_member\_id, total\_resource\_verif\_code, tot\_income\_verif\_code, inc.created\_by AS wksht\_created\_by,  inc.created\_date AS wksht\_created\_date, inc.modified\_by AS wksht\_modified\_by,  inc.modified\_date AS wksht\_modified\_date, inc.reimburseability\_member\_id, comment\_text, det.\*  FROM inc\_res\_summary\_wksht inc INNER JOIN inc\_res\_wksht\_detail det  ON det.inc\_res\_summary\_wksht\_id = inc.inc\_res\_summary\_wksht\_id  WHERE inc.eligible\_member\_id = eligiblememberid  ORDER BY det.inc\_res\_summary\_wksht\_id DESC;  CURSOR mainquery2 IS  SELECT inc.reimburseability\_member\_id, COUNT (DISTINCT det.inc\_res\_summary\_wksht\_id)  FROM inc\_res\_summary\_wksht inc INNER JOIN inc\_res\_wksht\_detail det  ON det.inc\_res\_summary\_wksht\_id = inc.inc\_res\_summary\_wksht\_id  GROUP BY inc.reimburseability\_member\_id  HAVING COUNT (DISTINCT det.inc\_res\_summary\_wksht\_id) > 1;  CURSOR subquery2 (reimburseabilitymemberid NUMBER) IS  SELECT eligible\_member\_id, total\_resource\_verif\_code, tot\_income\_verif\_code, inc.created\_by AS wksht\_created\_by,  inc.created\_date AS wksht\_created\_date, inc.modified\_by AS wksht\_modified\_by,  inc.modified\_date AS wksht\_modified\_date, inc.reimburseability\_member\_id, comment\_text, det.\*  FROM inc\_res\_summary\_wksht inc INNER JOIN inc\_res\_wksht\_detail det  ON det.inc\_res\_summary\_wksht\_id = inc.inc\_res\_summary\_wksht\_id  WHERE inc.reimburseability\_member\_id = reimburseabilitymemberid  ORDER BY det.inc\_res\_summary\_wksht\_id DESC;  worksheetid NUMBER := 0;  firstrecordfound VARCHAR (2) := 'N';  previouswkshtid NUMBER := 0;  BEGIN  FOR mainqueryresults IN mainquery LOOP --Main Loop  firstrecordfound := 'N';  FOR subqueryresults IN subquery (mainqueryresults.eligible\_member\_id) LOOP  IF (firstrecordfound = 'N') THEN  worksheetid := subqueryresults.inc\_res\_summary\_wksht\_id;  firstrecordfound := 'Y';  ELSIF (worksheetid <> subqueryresults.inc\_res\_summary\_wksht\_id) THEN  IF (previouswkshtid <> subqueryresults.inc\_res\_summary\_wksht\_id) THEN  INSERT INTO sacwis\_fix.defect\_13662\_sum\_wksht  VALUES (subqueryresults.eligible\_member\_id, subqueryresults.inc\_res\_summary\_wksht\_id,  subqueryresults.total\_resource\_verif\_code, subqueryresults.tot\_income\_verif\_code,  subqueryresults.wksht\_created\_by, subqueryresults.wksht\_created\_date,  subqueryresults.wksht\_modified\_by, subqueryresults.wksht\_modified\_date,  subqueryresults.reimburseability\_member\_id, subqueryresults.comment\_text);  END IF;  INSERT INTO sacwis\_fix.defect\_13662\_wksht\_detail  VALUES (subqueryresults.inc\_res\_summary\_wksht\_id, subqueryresults.inc\_res\_wksht\_detail\_id,  subqueryresults.finance\_cat\_code, subqueryresults.finance\_type\_code, subqueryresults.finance\_amt,  subqueryresults.disregard\_amt, subqueryresults.expense\_amt, subqueryresults.do\_not\_include\_flag,  subqueryresults.total\_countable\_amt, subqueryresults.created\_by, subqueryresults.created\_date,  subqueryresults.modified\_by, subqueryresults.modified\_date, subqueryresults.employment\_history\_id,  subqueryresults.finance\_type\_id);  DELETE FROM inc\_res\_summary\_wksht  WHERE inc\_res\_summary\_wksht\_id = subqueryresults.inc\_res\_summary\_wksht\_id;  DELETE FROM inc\_res\_wksht\_detail  WHERE inc\_res\_wksht\_detail\_id = subqueryresults.inc\_res\_wksht\_detail\_id;  previouswkshtid := subqueryresults.inc\_res\_summary\_wksht\_id;  COMMIT;  END IF;  END LOOP;  END LOOP; --END MAIN LOOP  --REIMBURSEABILITY RECORDS  FOR mainqueryresults2 IN mainquery2 LOOP --Main2 Loop  firstrecordfound := 'N';  previouswkshtid := 0;  FOR subqueryresults2 IN subquery2 (mainqueryresults2.reimburseability\_member\_id) LOOP  IF (firstrecordfound = 'N') THEN  worksheetid := subqueryresults2.inc\_res\_summary\_wksht\_id;  firstrecordfound := 'Y';  ELSIF (worksheetid <> subqueryresults2.inc\_res\_summary\_wksht\_id) THEN  --  IF (previouswkshtid <> subqueryresults2.inc\_res\_summary\_wksht\_id) THEN  INSERT INTO sacwis\_fix.defect\_13662\_sum\_wksht  VALUES (subqueryresults2.eligible\_member\_id, subqueryresults2.inc\_res\_summary\_wksht\_id,  subqueryresults2.total\_resource\_verif\_code, subqueryresults2.tot\_income\_verif\_code,  subqueryresults2.wksht\_created\_by, subqueryresults2.wksht\_created\_date,  subqueryresults2.wksht\_modified\_by, subqueryresults2.wksht\_modified\_date,  subqueryresults2.reimburseability\_member\_id, subqueryresults2.comment\_text);  END IF;  INSERT INTO sacwis\_fix.defect\_13662\_wksht\_detail  VALUES (subqueryresults2.inc\_res\_summary\_wksht\_id, subqueryresults2.inc\_res\_wksht\_detail\_id,  subqueryresults2.finance\_cat\_code, subqueryresults2.finance\_type\_code,  subqueryresults2.finance\_amt, subqueryresults2.disregard\_amt, subqueryresults2.expense\_amt,  subqueryresults2.do\_not\_include\_flag, subqueryresults2.total\_countable\_amt,  subqueryresults2.created\_by, subqueryresults2.created\_date, subqueryresults2.modified\_by,  subqueryresults2.modified\_date, subqueryresults2.employment\_history\_id,  subqueryresults2.finance\_type\_id);  DELETE FROM inc\_res\_summary\_wksht  WHERE inc\_res\_summary\_wksht\_id = subqueryresults2.inc\_res\_summary\_wksht\_id;  DELETE FROM inc\_res\_wksht\_detail  WHERE inc\_res\_wksht\_detail\_id = subqueryresults2.inc\_res\_wksht\_detail\_id;  previouswkshtid := subqueryresults2.inc\_res\_summary\_wksht\_id;  COMMIT;  END IF;  END LOOP;  END LOOP --END MAIN2 LOOP  COMMIT;  END  CREATE TABLE sacwis\_fix.defect\_13225\_data (  tablename VARCHAR2 (30),  ID NUMBER,  moddified\_date DATE,  modified\_by NUMBER,  inactive\_flag NUMBER  )  PCTFREE 0  TABLESPACE sacwis\_t;  DECLARE  sequenceid NUMBER;  refdataconfigid NUMBER;  refdataid NUMBER;  domaincode VARCHAR2 (30) := 'UnearnedIncome';  CURSOR c (id1 IN NUMBER) IS  SELECT \*  FROM ref\_data  WHERE ref\_data\_id = id1;  BEGIN  --Make Security Survivors Benefits inactive  SELECT ref\_data\_id  INTO refdataid  FROM ref\_data  WHERE domain\_code = domaincode AND ref\_data\_code = 'SclScrtySrvivr';  FOR r IN c (refdataid) LOOP  INSERT INTO sacwis\_fix.defect\_13225\_data  VALUES ('UPDATEREFDATA', refdataid, r.modified\_date, r.modified\_by, r.inactive\_flag);  END LOOP;  UPDATE ref\_data rd  SET rd.inactive\_flag = 1,  rd.modified\_by = 999,  rd.modified\_date = SYSDATE  WHERE rd.ref\_data\_id = refdataid;  --Add new values to REF\_DATA  SELECT ref\_data\_id\_seq.NEXTVAL  INTO sequenceid  FROM fast\_dual;  INSERT INTO sacwis\_fix.defect\_13225\_data  VALUES ('REFDATA', sequenceid, NULL, NULL, NULL);  SELECT ref\_data\_config\_id  INTO refdataconfigid  FROM ref\_data\_config rdc  WHERE rdc.domain\_code = domaincode;  INSERT INTO ref\_data rd  (ref\_data\_id, parent\_ref\_data\_code, domain\_code, ref\_data\_code, short\_desc,  long\_desc, sort\_value, inactive\_flag, created\_by, created\_date, modified\_by, modified\_date,  ref\_data\_config\_id, group\_code  )  VALUES (sequenceid, NULL, domaincode, 'Ssdi', 'SSDI- Social Security Disability Insurance',  'SSDI- Social Security Disability Insurance', 1, 0, 100, SYSDATE, 100, SYSDATE,  refdataconfigid, '00'  );  SELECT ref\_data\_id\_seq.NEXTVAL  INTO sequenceid  FROM fast\_dual;  INSERT INTO sacwis\_fix.defect\_13225\_data  VALUES ('REFDATA', sequenceid, NULL, NULL, NULL);  INSERT INTO ref\_data rd  (ref\_data\_id, parent\_ref\_data\_code, domain\_code, ref\_data\_code, short\_desc,  long\_desc, sort\_value, inactive\_flag, created\_by, created\_date, modified\_by, modified\_date,  ref\_data\_config\_id, group\_code  )  VALUES (sequenceid, NULL, domaincode, 'Rsdi', 'RSDI-Retirement, Survivors and Disability Insurance',  'RSDI-Retirement, Survivors and Disability Insurance', 1, 0, 100, SYSDATE, 100, SYSDATE,  refdataconfigid, '00'  );  --Insert values into FINANCE\_ELIG\_CALC for new values to be included in the income/resources worksheet  SELECT finance\_elig\_calc\_id\_seq.NEXTVAL  INTO sequenceid  FROM fast\_dual;  INSERT INTO sacwis\_fix.defect\_13225\_data  VALUES ('FINANCEELIGCALC', sequenceid, NULL, NULL, NULL);  INSERT INTO finance\_elig\_calc fec  (finance\_elig\_calc\_id, finance\_cat\_code, finance\_type\_code, eligibility\_calc\_flag, created\_by, created\_date,  modified\_by, modified\_date  )  VALUES (sequenceid, 'Unearned', 'Ssdi', 1, 100, SYSDATE,  100, SYSDATE  );  SELECT finance\_elig\_calc\_id\_seq.NEXTVAL  INTO sequenceid  FROM fast\_dual;  INSERT INTO sacwis\_fix.defect\_13225\_data  VALUES ('FINANCEELIGCALC', sequenceid, NULL, NULL, NULL);  INSERT INTO finance\_elig\_calc fec  (finance\_elig\_calc\_id, finance\_cat\_code, finance\_type\_code, eligibility\_calc\_flag, created\_by, created\_date,  modified\_by, modified\_date  )  VALUES (sequenceid, 'Unearned', 'Rsdi', 1, 100, SYSDATE,  100, SYSDATE  );  COMMIT;  END  CREATE TABLE datafix\_13918\_intake  PCTFREE 0  STORAGE (FREELIST GROUPS 3)  TABLESPACE intake\_t  AS SELECT /\*+ PARALLEL (i, 4, 1) \*/  i.intake\_id,  i.ai\_completion\_date,  i.ai\_completion\_action\_code,  (SELECT LEAST (COUNT(\*), 1)  FROM intake\_type it  WHERE it.intake\_id = i.intake\_id  AND i.category\_code = 'FINS'  AND it.intake\_type\_code = 'OOHPERP'  ) AS stranger\_danger\_flag,  NVL (i.is\_disp\_complete, 0) AS is\_disp\_complete,  i.disp\_date,  i.case\_disposition,  DECODE (LEAST (i.created\_by, 99), 99, 'SACWIS', 'CONVERSION') AS intake\_data\_source,  (SELECT MIN (rth.status\_date) AS approval\_date  FROM ai\_intake\_link aiil  INNER JOIN routed\_task\_history rth ON rth.task\_id = aiil.family\_assessment\_id  AND rth.task\_type\_code = 'FA'  AND rth.current\_status\_flag = 1  AND rth.status\_date IS NOT NULL  AND rth.task\_action\_code = 'AF'  AND rth.task\_status\_code = 'A'  WHERE aiil.intake\_id = i.intake\_id  ) AS fa\_approval\_date,  (SELECT MIN (rth.status\_date) AS approval\_date  FROM ai\_intake\_link aiil  INNER JOIN routed\_task\_history rth ON rth.task\_id = aiil.ongoing\_assessment\_id  AND rth.task\_type\_code = 'OA'  AND rth.current\_status\_flag = 1  AND rth.status\_date IS NOT NULL  AND rth.task\_action\_code = 'AF'  AND rth.task\_status\_code = 'A'  WHERE aiil.intake\_id = i.intake\_id  ) AS oa\_approval\_date,  (SELECT MIN (rth.status\_date) AS approval\_date  FROM ai\_intake\_link aiil  INNER JOIN routed\_task\_history rth ON rth.task\_id = aiil.spec\_assessment\_id  AND rth.task\_type\_code = 'SPA'  AND rth.current\_status\_flag = 1  AND rth.status\_date IS NOT NULL  AND rth.task\_action\_code = 'AF'  AND rth.task\_status\_code = 'A'  WHERE aiil.intake\_id = i.intake\_id  ) AS spec\_approval\_date,  (SELECT MIN (rth.status\_date) AS approval\_date  FROM justification j  INNER JOIN justification\_type jt ON jt.justification\_id = j.justification\_id  INNER JOIN routed\_task\_history rth ON rth.task\_id = j.justification\_id  AND rth.task\_type\_code = 'JUST'  AND rth.current\_status\_flag = 1  AND rth.status\_date IS NOT NULL  AND rth.task\_action\_code = 'AF'  AND rth.task\_status\_code = 'A'  WHERE j.intake\_id = i.intake\_id  AND jt.justification\_type\_code = ANY ('WAIVECOMPFAMASS', 'WAIVECOMPONGOINGASS', 'WAIVECOMPSPECASS')  ) AS just\_approval\_date,  (SELECT MIN (cal.activity\_date)  FROM case\_act\_log\_intake\_link calil  INNER JOIN case\_activity\_log cal ON cal.activity\_log\_id = calil.activity\_log\_id  INNER JOIN case\_activity\_narrative can ON can.activity\_log\_id = cal.activity\_log\_id  WHERE calil.intake\_id = i.intake\_id  AND NVL (start\_time, cal.activity\_date) >= i.decision\_date  AND TO\_CHAR (DBMS\_LOB.SUBSTR (can.activity\_note, 40, 1)) = ANY ('From event I04 Investigation Completed', 'From event I04')  AND cal.activity\_state\_code = 'COMPLETED'  ) AS completed\_date,  i.modified\_by,  i.modified\_date  FROM intake i  WHERE ( i.category\_code = ANY ('CANRPT', 'DEPENRPT')  OR ( i.category\_code = 'FINS'  AND EXISTS (SELECT 'T'  FROM intake\_type it  WHERE it.intake\_id = i.intake\_id  AND it.intake\_type\_code = 'OOHPERP'  )  )  )  AND i.screening\_decision\_code = ANY ('SCREENED IN', 'SCREENED IN AR')  AND ( i.ai\_completion\_date IS NULL  OR i.ai\_completion\_action\_code IS NULL);  -- Analyze table  --BEGIN  -- dbms\_stats.gather\_table\_stats  -- (ownname => 'SACWIS',  -- tabname => 'DATAFIX\_13918\_INTAKE',  -- estimate\_percent => dbms\_stats.auto\_sample\_size,  -- block\_sample => TRUE,  -- method\_opt => 'FOR ALL COLUMNS SIZE SKEWONLY',  -- degree => dbms\_stats.default\_degree,  -- cascade => TRUE);  --END;  --/  /\*  +----------------------------------------------------------------+  | Create a unique audit user for this datafix. |  +----------------------------------------------------------------+  \*/  EXECUTE pkAuditUser.CreateDataFixAudit('13918');  /\*  +----------------------------------------------------------------+  | Correct intake completion information. |  +----------------------------------------------------------------+  \*/  DECLARE  vRowLimit CONSTANT PLS\_INTEGER := 2000;  vFetchCnt PLS\_INTEGER := 0; -- Current row fetch counter  vTotalFetchCnt PLS\_INTEGER := 0; -- Total rows fetched  vUpdPtr PLS\_INTEGER := 0; -- Pointer into output buffer array  vUpdRowCnt PLS\_INTEGER := 0; -- Updated row counter  --  CURSOR curIntake IS  SELECT intake\_id,  ai\_completion\_date,  ai\_completion\_action\_code,  stranger\_danger\_flag,  is\_disp\_complete,  disp\_date,  case\_disposition,  intake\_data\_source,  fa\_approval\_date,  oa\_approval\_date,  spec\_approval\_date,  just\_approval\_date,  completed\_date,  modified\_by,  modified\_date  FROM datafix\_13918\_intake;  --  -- Input data storage array  TYPE tIntake IS TABLE OF curIntake%ROWTYPE;  rIntake tIntake := tIntake (); -- instantiate input buffer (array of records)  --  -- Intermediate variables  vAiCompletionDate intake.ai\_completion\_date%TYPE;  vAiCompletionSource intake.ai\_completion\_action\_code%TYPE;  --  -- Output data storage arrays  TYPE tCmplDateArr IS TABLE OF intake.ai\_completion\_date%TYPE;  TYPE tCmplActCdArr IS TABLE OF intake.ai\_completion\_action\_code%TYPE;  TYPE tIntakeIdArr IS TABLE OF intake.intake\_id%TYPE;  aAiCompletionDate tCmplDateArr := tCmplDateArr ();  aAiCompletionSource tCmplActCdArr := tCmplActCdArr ();  aIntakeId tIntakeIdArr := tIntakeIdArr ();  --  BEGIN  -- Allocate array memory  aIntakeId.EXTEND (vRowLimit);  aAiCompletionDate.EXTEND (vRowLimit);  aAiCompletionSource.EXTEND (vRowLimit);  --  OPEN curIntake;  --  <<lp1>>  LOOP  --  FETCH curIntake BULK COLLECT INTO rIntake LIMIT vRowLimit;  --  vFetchCnt := curIntake%ROWCOUNT - vTotalFetchCnt;  --  EXIT WHEN vFetchCnt <= 0;  --  vTotalFetchCnt := vTotalFetchCnt + vFetchCnt;  --  <<lp2>>  FOR i IN 1 .. vFetchCnt LOOP  vAiCompletionDate := NULL;  vAiCompletionSource := NULL;  --  IF rIntake (i).fa\_approval\_date IS NOT NULL THEN  vAiCompletionDate := rIntake (i).fa\_approval\_date;  vAiCompletionSource := 'Family Assessment';  END IF;  --  IF rIntake (i).oa\_approval\_date IS NOT NULL THEN  IF vAiCompletionDate IS NULL THEN  vAiCompletionDate := rIntake (i).oa\_approval\_date;  vAiCompletionSource := 'Ongoing Assessment';  --  ELSIF rIntake (i).oa\_approval\_date < vAiCompletionDate THEN  vAiCompletionDate := rIntake (i).oa\_approval\_date;  vAiCompletionSource := 'Ongoing Assessment';  END IF;  END IF;  --  IF rIntake (i).spec\_approval\_date IS NOT NULL THEN  IF vAiCompletionDate IS NULL THEN  vAiCompletionDate := rIntake (i).spec\_approval\_date;  vAiCompletionSource := 'Specialized Assessment';  --  ELSIF rIntake (i).spec\_approval\_date < vAiCompletionDate THEN  vAiCompletionDate := rIntake (i).spec\_approval\_date;  vAiCompletionSource := 'Specialized Assessment';  END IF;  END IF;  --  IF vAiCompletionDate IS NULL THEN  IF rIntake (i).just\_approval\_date IS NOT NULL THEN  IF rIntake (i).stranger\_danger\_flag = 1 AND rIntake (i).disp\_date IS NOT NULL THEN  vAiCompletionDate := rIntake (i).disp\_date;  vAiCompletionSource := 'Disposition';  --  ELSE  vAiCompletionDate := rIntake (i).just\_approval\_date;  vAiCompletionSource := 'Justification/Waiver';  END IF;  --  ELSE  IF rIntake (i).intake\_data\_source = 'CONVERSION' AND rIntake (i).completed\_date IS NOT NULL THEN  vAiCompletionDate := rIntake (i).completed\_date;  vAiCompletionSource := 'Case Activity Log';  --  ELSIF rIntake (i).disp\_date IS NOT NULL THEN  vAiCompletionDate := rIntake (i).disp\_date;  vAiCompletionSource := 'Disposition';  END IF;  END IF;  END IF;  --  IF vAiCompletionDate IS NOT NULL AND vAiCompletionSource IS NOT NULL THEN  -- Buffer data for bulk update  vUpdPtr := vUpdPtr + 1;  aAiCompletionDate (vUpdPtr) := vAiCompletionDate;  aAiCompletionSource (vUpdPtr) := vAiCompletionSource;  aIntakeId (vUpdPtr) := rIntake (i).intake\_id;  END IF;  END LOOP; /\* lp2 \*/  --  IF vUpdPtr > 0 THEN  --  FORALL j IN 1 .. vUpdPtr  UPDATE intake  SET ai\_completion\_date = aAiCompletionDate (j),  ai\_completion\_action\_code = aAiCompletionSource (j)  WHERE intake\_id = aIntakeId (j);  --  vUpdRowCnt := vUpdRowCnt + SQL%ROWCOUNT;  vUpdPtr := 0; -- Reset output buffer pointer  --  COMMIT;  --  END IF;  END LOOP; /\* lp1 \*/  --  CLOSE curIntake;  --  -- Free output buffer memory  aAiCompletionDate.TRIM (vRowLimit);  aAiCompletionSource.TRIM (vRowLimit);  aIntakeId.TRIM (vRowLimit);  --  dbms\_output.put\_line (TO\_CHAR (vUpdRowCnt) || ' intakes updated');  --  EXCEPTION  WHEN OTHERS THEN  ROLLBACK;  dbms\_output.put\_line ('Program error when updating intake');  dbms\_output.put\_line (SQLERRM);  RAISE\_APPLICATION\_ERROR(-20099, 'Program error when updating intake', TRUE);  END;  /  COMMIT |
| Audit Tables Updated | New Tables and Modified Tables  Reference Above for Table and Column Adds and Modifications |
| List of reports deployed : | Reference Bill Ennis |
| List of reports deactivated : | Reference Bill Ennis |

**Section 3 – Testing Plan/Results**

3.1 Test Plan

|  |  |
| --- | --- |
| Identify the database used to test the software/SQL: | Flair , system test |
| How many days removed from production is the data in this database? | n/a |
| Name of person(s) completing the test in this database: | Stephen Fibelkorn |
| Identify the database used for user acceptance test: | UAT |
| How many days removed from production is the data in this database? | Start of build Cycle (32 Days) |
| Name of person(s) completing the user acceptance test in this database: | Iroabuchi Arum |

**Section 4 – Production Execution**

* 1. Execution Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identify whether a full or partial backup of the production database is needed, or indicate that no backup is needed. | Full | **Successful RMAN backup Needed** | Partial |  |
| Explanation: | | | |
| Identify when the backup will be taken: | Date: 11/09/2011  Time: 7.00 pm | | | |
| Identify who will apply the software/SQL: | ODJFS DBA | | | |

* 1. Execution Results

|  |  |
| --- | --- |
| Location of results documentation: | Results to be e-mailed to the SACWIS team. |