PROMIS  
DB DEPLOYMENT DOCUMENT

Must take Backup of existing code for Every objects before Migrate changes.

1. Overview of changes in DB

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SCHEMA | TYPE | NAME | NEW/ MODIFIED | JIRA |
| IPMS\_DATA | TABLE | COLLABORATION\_DETAILS | NEW | PROMIS- 602 |
| LICENSE\_DETAIL\_TYPE | NEW | PROMIS- 602 |
| LICENSE\_DETAILS | NEW | PROMIS- 602 |
| PARTNER | NEW | PROMIS- 602 |
| PROJECT\_RELATED\_DEV | NEW | PROMIS- 604 |
| PROJECT\_RELATED\_MSP | NEW | PROMIS- 604 |
| PROJECT\_RELATED\_PREDECESSOR | NEW | PROMIS- 604 |
| PROJECT\_RELATED\_SUCCESSOR | NEW | PROMIS- 604 |
| BUDGET\_STATUS | NEW | PROMIS- 604 |
| PROJECT\_ROADMAP | NEW | PROMIS- 604 |
| SAMD\_STATUS | NEW | PROMIS- 604 |
| PHASE | MODIFIED | PROMIS- 604 |
| PROGRAM | MODIFIED | PROMIS- 604 |
| PROJECT | MODIFIED | PROMIS- 604 |
| SAMD\_STATUS | MODIFIED | PROMIS- 604 |
| BUDGET\_STATUS | MODIFIED | PROMIS- 604 |
| PROJECT\_AREA | MODIFIED | PROMIS- 604 |
| HELP\_BUNDLE | MODIFIED | PROMIS- 604 |
| MILESTONE | MODIFIED | PROMIS- 604 |
| MILESTONE\_TYPE | MODIFIED | PROMIS- 604 |
| PROJECT\_AREA\_MILESTONE | MODIFIED | PROMIS- 604 |
| TRIGGER | PROJECT\_COLLABORATION\_TR | NEW | PROMIS- 602 |
| PROJECT\_LICENSE\_TR | NEW | PROMIS- 602 |
| PROJECT\_RELATED\_DEV\_TR | NEW | PROMIS- 604 |
| PROJECT\_RELATED\_DEV\_DEL\_TR | NEW | PROMIS-604 |
| PROJECT\_RELATED\_MSP\_TR | NEW | PROMIS- 604 |
| PROJECT\_CODE\_CHANGED\_TR | MODIFIED | PROMIS- 604 |
| PROJECT\_RELATED\_PRED\_TR | NEW | PROMIS- 604 |
| PROJECT\_RELATED\_SUCC\_TR | NEW | PROMIS- 604 |
| PROJECT\_TR | MODIFIED | PROMIS- 604 |
| PROJECT\_ROADMAP\_TR | NEW | PROMIS- 604 |
| TEAM\_ROLE\_TR | MODIFIED | PROMIS- 604 |
| SEQUENCE | COLLABORATION\_ID\_SEQ | NEW | PROMIS- 602 |
| LICENSE\_ID\_SEQ | NEW | PROMIS- 602 |
| PRJ\_REL\_DEV\_ID\_SEQ | NEW | PROMIS- 604 |
| PRJ\_REL\_MSP\_ID\_SEQ | NEW | PROMIS- 604 |
| PRJ\_REL\_PRED\_ID\_SEQ | NEW | PROMIS- 604 |
| PRJ\_REL\_SUCC\_ID\_SEQ | NEW | PROMIS- 604 |
| ROADMAP\_ID\_SEQ | NEW | PROMIS- 604 |
| IPMS\_DATA | VIEW | SOPHIA\_PROJECT\_ACTIVITIES\_VW | MODIFIED | PROMIS-605 |
|  |  | PRJ\_AUDIT\_MILESTONES\_SRC\_VW | MODIFIED | PROMIS-604 |
| IPMS\_REPO | MATERIALIZED VIEW | COLLABORATION\_DETAILS\_DIM | NEW | PROMIS- 602 |
| LICENSE\_DETAILS\_DIM | NEW | PROMIS- 602 |
| PROJECT\_ROADMAP\_DIM | NEW | PROMIS- 604 |
| PROJECT\_REL\_MSP\_DEV\_DIM | NEW | PROMIS- 604 |
| PROJECT\_REL\_PREDECESSOR\_DIM | NEW | PROMIS- 604 |
| PROJECT\_REL\_SUCCESSOR\_DIM | NEW | PROMIS- 604 |
| PROJECT\_ROADMAP\_DIM | NEW | PROMIS- 604 |

1. Changes in DB

|  |
| --- |
| IPMS\_DATA |
| COLLABORATION\_DETAILS |
| CREATE SEQUENCE "IPMS\_DATA"."COLLABORATION\_ID\_SEQ" MINVALUE 1 MAXVALUE 999999999 INCREMENT BY 1 START WITH 2385 CACHE 20 NOORDER CYCLE NOKEEP GLOBAL ; |
| CREATE TABLE "IPMS\_DATA"."COLLABORATION\_DETAILS"  ( "ID" NUMBER(10,0),  "PROJECT\_ID" NVARCHAR2(20),  "CREATE\_USER\_ID" NVARCHAR2(20),  "UPDATE\_USER\_ID" NVARCHAR2(20),  "CREATE\_DATE" DATE DEFAULT SYSDATE,  "UPDATE\_DATE" DATE DEFAULT SYSDATE,  "PARTNER\_CODE" NVARCHAR2(100),  "COLLABORATION\_TYPE\_CODE" NVARCHAR2(100),  "START\_PHASE\_CODE" NVARCHAR2(100),  "END\_PHASE\_CODE" NVARCHAR2(100),  "COMMENT" NVARCHAR2(500)  ) SEGMENT CREATION IMMEDIATE  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255  NOCOMPRESS LOGGING  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ; |
| CREATE UNIQUE INDEX "IPMS\_DATA"."SYS\_C00266651" ON "IPMS\_DATA"."COLLABORATION\_DETAILS" ("ID")  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ; |
| ALTER TABLE "IPMS\_DATA"."COLLABORATION\_DETAILS" MODIFY ("ID" NOT NULL ENABLE);  ALTER TABLE "IPMS\_DATA"."COLLABORATION\_DETAILS" ADD PRIMARY KEY ("ID")  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ENABLE; |
| CREATE OR REPLACE EDITIONABLE TRIGGER "IPMS\_DATA"."PROJECT\_COLLABORATION\_TR"  before insert or update on collaboration\_details for each row  begin  if inserting then  if :new.id is null then  select collaboration\_id\_seq.nextval into :new.id from dual;  end if;  :new.create\_user\_id := nvl(:new.create\_user\_id,'TRIGGER');  :new.update\_user\_id := :new.create\_user\_id;  end if;  if updating then  :new.update\_date := sysdate;  :new.update\_user\_id := nvl(:new.update\_user\_id,'TRIGGER');  end if;  end;  /  ALTER TRIGGER "IPMS\_DATA"."PROJECT\_COLLABORATION\_TR" ENABLE; |
| LICENSE\_DETAIL\_TYPE |
| CREATE TABLE "IPMS\_DATA"."LICENSE\_DETAIL\_TYPE"  ( "CODE" NVARCHAR2(10) NOT NULL ENABLE,  "NAME" NVARCHAR2(100) NOT NULL ENABLE,  "IS\_ACTIVE" NUMBER(1,0) DEFAULT 1 NOT NULL ENABLE,  "DESCRIPTION" NVARCHAR2(500),  CHECK (is\_active in (0,1)) ENABLE,  PRIMARY KEY ("CODE")  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ENABLE  ) SEGMENT CREATION IMMEDIATE  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255  NOCOMPRESS LOGGING  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ; |
| LICENSE\_DETAILS |
| CREATE TABLE "IPMS\_DATA"."LICENSE\_DETAILS"  ( "ID" NUMBER(10,0) NOT NULL ENABLE,  "PROJECT\_ID" NVARCHAR2(20),  "CREATE\_USER\_ID" NVARCHAR2(20),  "UPDATE\_USER\_ID" NVARCHAR2(20),  "CREATE\_DATE" DATE DEFAULT SYSDATE,  "UPDATE\_DATE" DATE DEFAULT SYSDATE,  "LICENSOR\_CODE" NVARCHAR2(100),  "LICENSEE\_CODE" NVARCHAR2(100),  "LICENSE\_TYPE\_CODE" NVARCHAR2(100),  "LICENSE\_AGREEMENT\_DATE" DATE,  "LICENSE\_COMMENT" NVARCHAR2(500),  "PREDECESSOR\_ID" CHAR(30 BYTE),  PRIMARY KEY ("ID")  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ENABLE  ) SEGMENT CREATION IMMEDIATE  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255  NOCOMPRESS LOGGING  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ; |
| CREATE SEQUENCE "IPMS\_DATA"."LICENSE\_ID\_SEQ" MINVALUE 1 MAXVALUE 9999999999999999999 INCREMENT BY 1 START WITH 1421 CACHE 20 NOORDER CYCLE NOKEEP GLOBAL ; |
| CREATE OR REPLACE EDITIONABLE TRIGGER "IPMS\_DATA"."PROJECT\_LICENSE\_TR"  for insert or update on license\_details  compound trigger  before each row is  begin  if inserting then  if :new.id is null then  select license\_id\_seq.nextval into :new.id from dual;  end if;  -- if :new.predecessor\_id is null then  -- project\_license\_pkg.update\_predecessor(:new.project\_id);  -- end if;  :new.create\_user\_id := nvl(:new.create\_user\_id,'TRIGGER');  :new.update\_user\_id := :new.create\_user\_id;  end if;  if updating then  -- if :new.predecessor\_id is null then  -- project\_license\_pkg.update\_predecessor(:new.project\_id);  -- end if;  :new.update\_date := sysdate;  :new.update\_user\_id := nvl(:new.update\_user\_id,'TRIGGER');  end if;  end before each row;  end;  /  ALTER TRIGGER "IPMS\_DATA"."PROJECT\_LICENSE\_TR" ENABLE; |
| PARTNER |
| CREATE TABLE "IPMS\_DATA"."PARTNER"  ( "CODE" NVARCHAR2(10) NOT NULL ENABLE,  "NAME" NVARCHAR2(100) NOT NULL ENABLE,  "IS\_ACTIVE" NUMBER(1,0) DEFAULT 1 NOT NULL ENABLE,  "DESCRIPTION" NVARCHAR2(500),  CHECK (is\_active in (0,1)) ENABLE,  PRIMARY KEY ("CODE")  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ENABLE  ) SEGMENT CREATION IMMEDIATE  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255  NOCOMPRESS LOGGING  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ; |
| PROJECT\_RELATED\_DEV |
| CREATE SEQUENCE "IPMS\_DATA"."PRJ\_REL\_DEV\_ID\_SEQ" MINVALUE 1 MAXVALUE 9999999999999999999 INCREMENT BY 1 START WITH 401 CACHE 20 NOORDER CYCLE NOKEEP GLOBAL ; |
| CREATE TABLE "IPMS\_DATA"."PROJECT\_RELATED\_DEV"  ( "ID" NUMBER,  "PROJECT\_ID" NVARCHAR2(20) NOT NULL ENABLE,  "DEV\_PROJECT\_ID" VARCHAR2(10 BYTE) CONSTRAINT "PRJ\_REL\_DEV\_CODE\_CNN" NOT NULL ENABLE,  "CREATE\_DATE" DATE CONSTRAINT "PRJ\_REL\_DEV\_CREATE\_DATE\_CNN" NOT NULL ENABLE,  "CREATE\_USER\_ID" VARCHAR2(20 BYTE) CONSTRAINT "PRJ\_REL\_DEV\_CREATE\_BY\_CNN" NOT NULL ENABLE,  "UPDATE\_DATE" DATE CONSTRAINT "PRJ\_REL\_DEV\_UPDATE\_DATE\_CNN" NOT NULL ENABLE,  "UPDATE\_USER\_ID" VARCHAR2(20 BYTE) CONSTRAINT "PRJ\_REL\_DEV\_UPDATE\_BY\_CNN" NOT NULL ENABLE,  CONSTRAINT "PRJ\_REL\_DEV\_PK" PRIMARY KEY ("ID")  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ENABLE,  CONSTRAINT "PRJ\_REL\_DEV\_PROJECT\_ID\_FK" FOREIGN KEY ("PROJECT\_ID")  REFERENCES "IPMS\_DATA"."PROJECT" ("ID") ON DELETE CASCADE ENABLE  ) SEGMENT CREATION IMMEDIATE  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255  NOCOMPRESS LOGGING  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ;  COMMENT ON COLUMN "IPMS\_DATA"."PROJECT\_RELATED\_DEV"."ID" IS 'Autogenerated surogate ID.';  COMMENT ON COLUMN "IPMS\_DATA"."PROJECT\_RELATED\_DEV"."PROJECT\_ID" IS 'The project for which related dev project is being selected.';  COMMENT ON COLUMN "IPMS\_DATA"."PROJECT\_RELATED\_DEV"."DEV\_PROJECT\_ID" IS 'Selected relevant Dev project.';  COMMENT ON TABLE "IPMS\_DATA"."PROJECT\_RELATED\_DEV" IS 'Table stores all related dev project ids for selected project. Table was created while implementing requirement: PROMIS-604';  CREATE INDEX "IPMS\_DATA"."PRJ\_REL\_DEV\_PROJECT\_ID\_FKI" ON "IPMS\_DATA"."PROJECT\_RELATED\_DEV" ("PROJECT\_ID")  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ; |
| CREATE OR REPLACE EDITIONABLE TRIGGER "IPMS\_DATA"."PROJECT\_RELATED\_DEV\_TR"  for insert or update on PROJECT\_RELATED\_DEV  compound trigger  before each row is  begin  if inserting then  if :new.id is null then  select PRJ\_REL\_DEV\_ID\_SEQ.nextval into :new.id from dual;  end if;  :new.create\_date := sysdate;  :new.create\_user\_id := nvl(:new.create\_user\_id,'TRIGGER');  :new.update\_user\_id := :new.create\_user\_id;  end if;  if updating then  :new.update\_user\_id := nvl(:new.update\_user\_id,'TRIGGER');  end if;  :new.update\_date := sysdate;  end before each row;  after each row is  begin  if inserting then  insert into project\_related\_msp(project\_id, rel\_msp\_project\_id)  values (:new.dev\_project\_id , :new.project\_id);  end if;  -- if deleting then  -- delete from project\_related\_msp prm where prm.project\_id = :old.dev\_project\_id;  -- end if;  end after each row;  end;  /  ALTER TRIGGER "IPMS\_DATA"."PROJECT\_RELATED\_DEV\_TR" ENABLE;  CREATE OR REPLACE EDITIONABLE TRIGGER "IPMS\_DATA"."PROJECT\_RELATED\_DEV\_DEL\_TR"  before delete on PROJECT\_RELATED\_DEV for each row  begin  delete from project\_related\_msp prm where prm.project\_id = :old.dev\_project\_id;  end; |
| PROJECT\_RELATED\_MSP |
| CREATE SEQUENCE "IPMS\_DATA"."PRJ\_REL\_MSP\_ID\_SEQ" MINVALUE 1 MAXVALUE 9999999999999999999 INCREMENT BY 1 START WITH 461 CACHE 20 NOORDER CYCLE NOKEEP GLOBAL ; |
| CREATE TABLE "IPMS\_DATA"."PROJECT\_RELATED\_MSP"  ( "ID" NUMBER,  "PROJECT\_ID" NVARCHAR2(20) NOT NULL ENABLE,  "REL\_MSP\_PROJECT\_ID" VARCHAR2(30 BYTE) CONSTRAINT "PRJ\_REL\_MSP\_ID\_CNN" NOT NULL ENABLE,  "CREATE\_DATE" DATE CONSTRAINT "PRJ\_REL\_MSP\_CREATE\_DATE\_CNN" NOT NULL ENABLE,  "CREATE\_USER\_ID" VARCHAR2(20 BYTE) CONSTRAINT "PRJ\_MSP\_CREATE\_BY\_CNN" NOT NULL ENABLE,  "UPDATE\_DATE" DATE CONSTRAINT "PRJ\_REL\_MSP\_UPDATE\_DATE\_CNN" NOT NULL ENABLE,  "UPDATE\_USER\_ID" VARCHAR2(20 BYTE) CONSTRAINT "PRJ\_REL\_MSP\_UPDATE\_BY\_CNN" NOT NULL ENABLE,  CONSTRAINT "PRJ\_REL\_MSP\_PK" PRIMARY KEY ("ID")  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ENABLE  ) SEGMENT CREATION IMMEDIATE  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255  NOCOMPRESS LOGGING  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ;  CREATE INDEX "IPMS\_DATA"."PRJ\_REL\_MSP\_PROJECT\_CODE\_FKI" ON "IPMS\_DATA"."PROJECT\_RELATED\_MSP" ("PROJECT\_ID")  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ; |
| CREATE OR REPLACE EDITIONABLE TRIGGER "IPMS\_DATA"."PROJECT\_RELATED\_MSP\_TR"  before insert or update on PROJECT\_RELATED\_MSP  for each row  begin  if inserting then  if :new.id is null then  select PRJ\_REL\_MSP\_ID\_SEQ.nextval into :new.id from dual;  end if;  :new.create\_date := sysdate;  :new.create\_user\_id := nvl(:new.create\_user\_id,'TRIGGER');  :new.update\_user\_id := :new.create\_user\_id;  end if;  if updating then  :new.update\_user\_id := nvl(:new.update\_user\_id,'TRIGGER');  end if;  :new.update\_date := sysdate;  end;  /  ALTER TRIGGER "IPMS\_DATA"."PROJECT\_RELATED\_MSP\_TR" ENABLE; |
| PROJECT\_RELATED\_PREDECESSOR |
| CREATE SEQUENCE "IPMS\_DATA"."PRJ\_REL\_PRED\_ID\_SEQ" MINVALUE 1 MAXVALUE 9999999999999999999 INCREMENT BY 1 START WITH 341 CACHE 20 NOORDER CYCLE NOKEEP GLOBAL ; |
| CREATE TABLE "IPMS\_DATA"."PROJECT\_RELATED\_PREDECESSOR"  ( "ID" NUMBER,  "PROJECT\_ID" NVARCHAR2(20) NOT NULL ENABLE,  "REL\_PREDECESSOR\_PROJECT\_ID" VARCHAR2(30 BYTE) CONSTRAINT "PRJ\_REL\_PRED\_CODE\_CNN" NOT NULL ENABLE,  "CREATE\_DATE" DATE CONSTRAINT "PRJ\_REL\_PRED\_CREATE\_DATE\_CNN" NOT NULL ENABLE,  "CREATE\_USER\_ID" VARCHAR2(20 BYTE) CONSTRAINT "PRJ\_PRED\_CREATE\_BY\_CNN" NOT NULL ENABLE,  "UPDATE\_DATE" DATE CONSTRAINT "PRJ\_REL\_PRED\_UPDATE\_DATE\_CNN" NOT NULL ENABLE,  "UPDATE\_USER\_ID" VARCHAR2(20 BYTE) CONSTRAINT "PRJ\_REL\_PRED\_UPDATE\_BY\_CNN" NOT NULL ENABLE,  CONSTRAINT "PRJ\_REL\_PRED\_PK" PRIMARY KEY ("ID")  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ENABLE,  CONSTRAINT "PRJ\_REL\_PRED\_PROJECT\_ID\_FK" FOREIGN KEY ("PROJECT\_ID")  REFERENCES "IPMS\_DATA"."PROJECT" ("ID") ON DELETE CASCADE ENABLE  ) SEGMENT CREATION IMMEDIATE  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255  NOCOMPRESS LOGGING  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ;  CREATE INDEX "IPMS\_DATA"."PRJ\_REL\_PRED\_PROJECT\_ID\_FKI" ON "IPMS\_DATA"."PROJECT\_RELATED\_PREDECESSOR" ("PROJECT\_ID")  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ; |
| CREATE OR REPLACE EDITIONABLE TRIGGER "IPMS\_DATA"."PROJECT\_RELATED\_PRED\_TR"  before insert or update on PROJECT\_RELATED\_PREDECESSOR  for each row  begin  if inserting then  if :new.id is null then  select PRJ\_REL\_PRED\_ID\_SEQ.nextval into :new.id from dual;  end if;  :new.create\_date := sysdate;  :new.create\_user\_id := nvl(:new.create\_user\_id,'TRIGGER');  :new.update\_user\_id := :new.create\_user\_id;  end if;  if updating then  :new.update\_user\_id := nvl(:new.update\_user\_id,'TRIGGER');  end if;  :new.update\_date := sysdate;  end;  /  ALTER TRIGGER "IPMS\_DATA"."PROJECT\_RELATED\_PRED\_TR" ENABLE; |
| PROJECT\_RELATED\_SUCCESSOR |
| CREATE SEQUENCE "IPMS\_DATA"."PRJ\_REL\_SUCC\_ID\_SEQ" MINVALUE 1 MAXVALUE 9999999999999999999 INCREMENT BY 1 START WITH 321 CACHE 20 NOORDER CYCLE NOKEEP GLOBAL ; |
| CREATE TABLE "IPMS\_DATA"."PROJECT\_RELATED\_SUCCESSOR"  ( "ID" NUMBER,  "PROJECT\_ID" NVARCHAR2(20) NOT NULL ENABLE,  "SUCCESSOR\_PROJECT\_ID" VARCHAR2(30 BYTE) CONSTRAINT "PRJ\_REL\_SUCC\_CODE\_CNN" NOT NULL ENABLE,  "CREATE\_DATE" DATE CONSTRAINT "PRJ\_REL\_SUCC\_CREATE\_DATE\_CNN" NOT NULL ENABLE,  "CREATE\_USER\_ID" VARCHAR2(20 BYTE) CONSTRAINT "PRJ\_SUCC\_DEV\_CREATE\_BY\_CNN" NOT NULL ENABLE,  "UPDATE\_DATE" DATE CONSTRAINT "PRJ\_REL\_SUCC\_UPDATE\_DATE\_CNN" NOT NULL ENABLE,  "UPDATE\_USER\_ID" VARCHAR2(20 BYTE) CONSTRAINT "PRJ\_REL\_SUCC\_UPDATE\_BY\_CNN" NOT NULL ENABLE,  CONSTRAINT "PRJ\_REL\_SUCC\_PK" PRIMARY KEY ("ID")  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ENABLE,  CONSTRAINT "PRJ\_REL\_SUCC\_PROJECT\_ID\_FK" FOREIGN KEY ("PROJECT\_ID")  REFERENCES "IPMS\_DATA"."PROJECT" ("ID") ON DELETE CASCADE ENABLE  ) SEGMENT CREATION IMMEDIATE  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255  NOCOMPRESS LOGGING  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ;  CREATE INDEX "IPMS\_DATA"."PRJ\_REL\_SUCC\_PROJECT\_ID\_FKI" ON "IPMS\_DATA"."PROJECT\_RELATED\_SUCCESSOR" ("PROJECT\_ID")  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ; |
| CREATE OR REPLACE EDITIONABLE TRIGGER "IPMS\_DATA"."PROJECT\_RELATED\_SUCC\_TR"  before insert or update on PROJECT\_RELATED\_SUCCESSOR  for each row  begin  if inserting then  if :new.id is null then  select PRJ\_REL\_SUCC\_ID\_SEQ.nextval into :new.id from dual;  end if;  :new.create\_date := sysdate;  :new.create\_user\_id := nvl(:new.create\_user\_id,'TRIGGER');  :new.update\_user\_id := :new.create\_user\_id;  end if;  if updating then  :new.update\_user\_id := nvl(:new.update\_user\_id,'TRIGGER');  end if;  :new.update\_date := sysdate;  end;  /  ALTER TRIGGER "IPMS\_DATA"."PROJECT\_RELATED\_SUCC\_TR" ENABLE; |
| PROJECT\_ROADMAP |
| CREATE SEQUENCE "IPMS\_DATA"."ROADMAP\_ID\_SEQ" MINVALUE 1 MAXVALUE 999999999 INCREMENT BY 1 START WITH 2365 CACHE 20 NOORDER CYCLE NOKEEP GLOBAL ; |
| CREATE TABLE "IPMS\_DATA"."PROJECT\_ROADMAP"  ( "ID" NUMBER(10,0) NOT NULL ENABLE,  "PROJECT\_ID" NVARCHAR2(20),  "DATE" DATE,  "TOPIC" NVARCHAR2(100),  "DESCRIPTION" NVARCHAR2(500),  "CREATE\_USER\_ID" NVARCHAR2(20),  "UPDATE\_USER\_ID" NVARCHAR2(20),  "CREATE\_DATE" DATE DEFAULT SYSDATE,  "UPDATE\_DATE" DATE DEFAULT SYSDATE,  CONSTRAINT "PROJECT\_ROADMAP\_PK" PRIMARY KEY ("ID")  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ENABLE  ) SEGMENT CREATION IMMEDIATE  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255  NOCOMPRESS LOGGING  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ;  CREATE INDEX "IPMS\_DATA"."PROJECT\_ID\_FKI" ON "IPMS\_DATA"."PROJECT\_ROADMAP" ("PROJECT\_ID")  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ; |
| CREATE OR REPLACE EDITIONABLE TRIGGER "IPMS\_DATA"."PROJECT\_ROADMAP\_TR"  before insert or update on project\_roadmap for each row  begin  if inserting then  if :new.id is null then  select roadmap\_id\_seq.nextval into :new.id from dual;  end if;  :new.create\_user\_id := nvl(:new.create\_user\_id,'TRIGGER');  :new.update\_user\_id := :new.create\_user\_id;  end if;  if updating then  :new.update\_date := sysdate;  :new.update\_user\_id := nvl(:new.update\_user\_id,'TRIGGER');  end if;  end;  /  ALTER TRIGGER "IPMS\_DATA"."PROJECT\_ROADMAP\_TR" ENABLE; |
| SAMD\_STATUS |
| CREATE TABLE "IPMS\_DATA"."SAMD\_STATUS"  ( "CODE" NVARCHAR2(10) NOT NULL ENABLE,  "NAME" NVARCHAR2(100) NOT NULL ENABLE,  "IS\_ACTIVE" NUMBER(1,0) DEFAULT 1 NOT NULL ENABLE,  "DESCRIPTION" NVARCHAR2(500),  CHECK (is\_active in (0,1)) ENABLE,  PRIMARY KEY ("CODE")  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ENABLE  ) SEGMENT CREATION IMMEDIATE  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255  NOCOMPRESS LOGGING  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ; |
| BUDGET\_STATUS |
| CREATE TABLE "IPMS\_DATA"."BUDGET\_STATUS"  ( "CODE" NVARCHAR2(10) NOT NULL ENABLE,  "NAME" NVARCHAR2(100) NOT NULL ENABLE,  "IS\_ACTIVE" NUMBER(1,0) DEFAULT 1 NOT NULL ENABLE,  "DESCRIPTION" NVARCHAR2(500),  CHECK (is\_active in (0,1)) ENABLE,  PRIMARY KEY ("CODE")  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ENABLE  ) SEGMENT CREATION IMMEDIATE  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255  NOCOMPRESS LOGGING  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)  TABLESPACE "USERS" ; |
| PHASE |
| INSERT INTO "IPMS\_DATA"."PHASE" (CODE, NAME, IS\_ACTIVE, ORDERING) VALUES ('SAMD.1', 'Ideation', '1', 150);  INSERT INTO "IPMS\_DATA"."PHASE" (CODE, NAME, IS\_ACTIVE, ORDERING) VALUES ('SAMD.2', 'Feasibility/Concept Finalization', '1', 160);  INSERT INTO "IPMS\_DATA"."PHASE" (CODE, NAME, IS\_ACTIVE, ORDERING) VALUES ('SAMD.3', 'Product Development', '1', 170);  INSERT INTO "IPMS\_DATA"."PHASE" (CODE, NAME, IS\_ACTIVE, ORDERING) VALUES ('SAMD.4', 'Product Deployment & Localization', '1', 180);  INSERT INTO "IPMS\_DATA"."PHASE" (CODE, NAME, IS\_ACTIVE, ORDERING) VALUES ('SAMD.5', 'Life Cycle Management', '1', 190); |
| BUDGET\_STATUS |
| INSERT INTO "IPMS\_DATA"."BUDGET\_STATUS" (CODE, NAME, IS\_ACTIVE) VALUES ('1', 'On track', '1');  INSERT INTO "IPMS\_DATA"."BUDGET\_STATUS" (CODE, NAME, IS\_ACTIVE) VALUES ('2', 'At risk', '1');  INSERT INTO "IPMS\_DATA"."BUDGET\_STATUS" (CODE, NAME, IS\_ACTIVE) VALUES ('3', 'Off truck', '1'); |
| SAMD\_STATUS |
| INSERT INTO "IPMS\_DATA"."SAMD\_STATUS" (CODE, NAME, IS\_ACTIVE) VALUES ('1', 'On track', '1');  INSERT INTO "IPMS\_DATA"."SAMD\_STATUS" (CODE, NAME, IS\_ACTIVE) VALUES ('2', 'At risk', '1');  INSERT INTO "IPMS\_DATA"."SAMD\_STATUS" (CODE, NAME) VALUES ('3', 'Off truck'); |
| MILESTONE |
| INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS0', 'MS0', 'DEC-SAMD', '1', '680');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS1', 'MS1', 'DEC-SAMD', '1', '690');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS2', 'MS2', 'DEC-SAMD', '1', '700');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3a', 'MS3a', 'DEC-SAMD', '1', '710');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3b', 'MS3b', 'DEC-SAMD', '1', '720');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3c', 'MS3c', 'DEC-SAMD', '1', '730');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3d', 'MS3d', 'DEC-SAMD', '1', '740');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3e', 'MS3e', 'DEC-SAMD', '1', '750');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3f', 'MS3f', 'DEC-SAMD', '1', '760');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3g', 'MS3g', 'DEC-SAMD', '1', '770');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3h', 'MS3h', 'DEC-SAMD', '1', '780');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3i', 'MS3i', 'DEC-SAMD', '1', '790');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3j', 'MS3j', 'DEC-SAMD', '1', '800');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3k', 'MS3k', 'DEC-SAMD', '1', '810');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3m', 'MS3m', 'DEC-SAMD', '1', '820');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3n', 'MS3n', 'DEC-SAMD', '1', '830');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3o', 'MS3o', 'DEC-SAMD', '1', '840');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3p', 'MS3p', 'DEC-SAMD', '1', '850');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3q', 'MS3q', 'DEC-SAMD', '1', '860');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3r', 'MS3r', 'DEC-SAMD', '1', '870');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3s', 'MS3s', 'DEC-SAMD', '1', '880');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3t', 'MS3t', 'DEC-SAMD', '1', '890');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3u', 'MS3u', 'DEC-SAMD', '1', '900');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3v', 'MS3v', 'DEC-SAMD', '1', '910');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3w', 'MS3w', 'DEC-SAMD', '1', '920');  INSERT INTO "IPMS\_DATA"."MILESTONE" (CODE, NAME, TYPE\_CODE, IS\_ACTIVE, ORDERING) VALUES ('MS3x', 'MS3x', 'DEC-SAMD', '1', '930'); |
| PROJECT\_AREA |
| INSERT INTO "IPMS\_DATA"."PROJECT\_AREA" (CODE, NAME, IS\_RUNNING\_IMPORT) VALUES ('SAMD', 'SaMD', 'SAMD'); |
| HELP\_BUNDLE |
| Insert into HELP\_BUNDLE (CODE,NAME,DEFINITION,URL) values ('PROJECT\_SAMD\_OVERVIEW\_CHARACTERISTICS','SaMD Project -> Overview -> Characteristics','Help','http://sp-coll-bhc.bayer-ag.com/sites/221953/ProMIS\_UserGuide/Under%20Construction.aspx');  Insert into HELP\_BUNDLE (CODE,NAME,DEFINITION,URL) values ('PROJECT\_SAMD\_OVERVIEW\_PLANS','SaMD Project -> Overview -> Plan Versions','Help','http://sp-coll-bhc.bayer-ag.com/sites/221953/ProMIS\_UserGuide/Under%20Construction.aspx'); |
| MILESTONE\_TYPE |
| Insert into IPMS\_DATA.MILESTONE\_TYPE (CODE,NAME,IS\_ACTIVE) values ('DEC-SAMD','SaMD Decision',1); |
| PROJECT\_AREA\_MILESTONE |
| Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS0');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS1');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS2');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3a');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3b');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3c');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3d');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3e');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3f');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3g');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3h');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3i');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3j');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3k');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3l');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3m');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3n');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3o');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3p');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3q');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3r');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3s');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3t');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3u');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3v');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3w');  Insert into IPMS\_DATA.PROJECT\_AREA\_MILESTONE (AREA\_CODE,MILESTONE\_CODE) values ('SAMD','MS3x'); |
| PROJECT\_TR |
| CREATE OR REPLACE EDITIONABLE TRIGGER IPMS\_DATA.PROJECT\_TR  for insert or update or delete on project  compound trigger  before each row is  begin  if inserting then  if :new.id is null then  select :new.program\_id||'-'||project\_id\_seq.nextval into :new.id from dual;  end if;  :new.create\_date := sysdate;  :new.create\_user\_id := nvl(:new.create\_user\_id,'TRIGGER');  if :new.phase\_code is null then :new.phase\_code := '9'; end if;  end if;  if inserting or updating then  select decode(:new.details\_progress, :old.details\_progress, :old.progress\_date, sysdate)  into :new.progress\_date  from dual  ;  if :new.area\_code = 'D1' then--PROMIS-747  if :new.is\_active = 1 then  :new.state\_code:='1';--=Going  elsif :new.is\_active = 0 and :new.termination\_date is not null then--BAY\_PROMIS-115  :new.state\_code := '5'; --5=Terminated  else  :new.state\_code:= null;  end if;  end if;  if inserting or updating then  if :old.area\_code in('SAMD') then  select decode(:new.details\_risks, :old.details\_risks, :old.progress\_date, sysdate)--PROMIS 604  into :new.progress\_date  from dual;  end if;  end if;  if :new.area\_code in ('PRD-MNT','RS','LG','LO','D3-TR','CO','D1') and :new.pidt\_release\_date is null then  --jira:PROMIS-68  --jira:PROMIS-442  :new.pidt\_release\_date:=sysdate;  end if;  end if;  if updating then  :new.update\_date := sysdate;  :new.update\_user\_id := nvl(:new.update\_user\_id,'TRIGGER');  if :new.state\_code <> '5' then --5=Terminated  :new.termination\_code := null;  :new.termination\_date := null;  :new.termination\_reason := null;  end if;  end if;  end before each row;    after each row is  begin  if inserting and :new.area\_code in ('PRE-POT','POST-POT') then--PROMIS-730,costs\_probability applies only for DEV projects!  insert into costs\_probability(project\_id,scope\_code,phase\_code,probability)  select :new.id,scope\_code,phase\_code,probability  from costs\_probability  where scope\_code='INT'  and project\_id is null  and sbe\_code is null  and substance\_type\_code is null;  end if;  if inserting and :new.area\_code in ('PRE-POT', 'POST-POT') and :new.predecessor\_project\_id is not null --PROMIS-602  then  declare countVar NUMBER(10);  begin  select count(\*) into countVar from license\_details where project\_id=:new.predecessor\_project\_id;  if countVar > 0 then  UPDATE ipms\_data.license\_details SET predecessor\_id=:new.id  WHERE project\_id=:new.predecessor\_project\_id;  declare  cursor license\_rec IS select licensor\_code,licensee\_code,license\_type\_code,license\_agreement\_date,license\_comment  FROM ipms\_data.license\_details WHERE project\_id = :new.predecessor\_project\_id;  newId nvarchar2(200);  newPredId nvarchar2(200);  p\_licensor NVARCHAR2(100);  p\_licensee NVARCHAR2(100);  p\_license\_type NVARCHAR2(100);  p\_license\_agreement\_date DATE;  p\_license\_comment NVARCHAR2(500);  p\_id2 NUMBER(10);  begin  open license\_rec;  loop  fetch license\_rec into p\_licensor, p\_licensee, p\_license\_type, p\_license\_agreement\_date,p\_license\_comment;  EXIT when license\_rec%notfound;  newId := :new.id;--project id  newPredId := :new.predecessor\_project\_id;-- predecessor\_project\_id  SELECT  license\_id\_seq.NEXTVAL  INTO p\_id2  FROM  dual;    INSERT INTO ipms\_data.license\_details (  id,  project\_id,  create\_user\_id,  update\_user\_id,  create\_date,  update\_date,  licensor\_code,  licensee\_code,  license\_type\_code,  license\_agreement\_date,  license\_comment,  predecessor\_id  ) VALUES (  p\_id2,  newId,  'PROC',  'PROC',  TO\_DATE(SYSDATE,'DD-MM-RR'),  TO\_DATE(SYSDATE,'DD-MM-RR'),  p\_licensor,  p\_licensee,  p\_license\_type,  p\_license\_agreement\_date,  p\_license\_comment,  NULL  );  --call (ipms\_data.project\_license\_pkg.update\_license\_detail(newId, newPredId));  end loop;  close license\_rec;  end;  end if;  end;  end if;  end after each row;  end;  /  ALTER TRIGGER "IPMS\_DATA"."PROJECT\_TR" ENABLE;  CREATE OR REPLACE EDITIONABLE TRIGGER "IPMS\_DATA"."PROJECT\_CODE\_CHANGED\_TR"  after update on project  for each row-- This trigger should only fire when an existing code changes  WHEN (old.code is not null and old.code != new.code) begin  if :new.qplan\_release\_date is not null then  notify\_pkg.project\_code\_changed(:new.id, :old.code, :new.code);  end if;  end;  /  ALTER TRIGGER "IPMS\_DATA"."PROJECT\_CODE\_CHANGED\_TR" ENABLE; |
| ALTER TABLE project  ADD (  DETAILS\_OBJECTIVE NVARCHAR2(2000 CHAR),  DETAILS\_RATIONALE NVARCHAR2(2000 CHAR),  DETAILS\_SCOPE NVARCHAR2(2000 CHAR),  DETAILS\_INTENDED\_USE NVARCHAR2(2000 CHAR),  DETAILS\_BENEFITS NVARCHAR2(2000 CHAR),  DETAILS\_EXECUTIVE\_SUMMARY NVARCHAR2(2000 CHAR),  DETAILS\_RISKS NVARCHAR2(2000 CHAR),  DETAILS\_HIGHLIGHTS NVARCHAR2(2000 CHAR),  DETAILS\_ACTIVITIES\_EVENTS NVARCHAR2(2000 CHAR),  DETAILS\_BUDGET NVARCHAR2(2000 CHAR),  DETAILS\_SAMD\_STATUS NVARCHAR2(100 CHAR),  DETAILS\_BUDGET\_STATUS NVARCHAR2(100 CHAR)  ); |
| TEAM\_ROLE\_TR |
| CREATE OR REPLACE EDITIONABLE TRIGGER "IPMS\_DATA"."TEAM\_ROLE\_TR"  before insert or update on team\_role  for each row  begin  --PROMISIII-337  --- it is still not clear what are default values (when UI does not provide explicit select,  --- but it is less risky to allow at least for DEV type of projects, the rest could be -disabled  :new.is\_dev\_relevant := nvl(:new.is\_dev\_relevant,1);  :new.is\_prdmnt\_relevant := nvl(:new.is\_prdmnt\_relevant,0);  :new.is\_d2prj\_relevant := nvl(:new.is\_d2prj\_relevant,0);  -- PROMIS 604  :new.is\_samd\_relevant := nvl(:new.is\_samd\_relevant,0);  end;  /  ALTER TRIGGER "IPMS\_DATA"."TEAM\_ROLE\_TR" ENABLE; |
| GRANT STMT |
| grant select on COLLABORATION\_DETAILS to ipms\_repo;  grant select on LICENSE\_DETAILS to ipms\_repo;  grant select on PROJECT\_RELATED\_PREDECESSOR to ipms\_repo;  grant select on PROJECT\_RELATED\_DEV to ipms\_repo;  grant select on PROJECT\_RELATED\_MSP to ipms\_repo;  grant select on PROJECT\_RELATED\_SUCCESSOR to ipms\_repo;  grant select on PROJECT\_ROADMAP to ipms\_repo; |
| IPMS\_REPO |
| COLLABORATION\_DETAILS\_DIM |
| -- Unable to render MATERIALIZED VIEW DDL for object IPMS\_REPO.COLLABORATION\_DETAILS\_DIM with DBMS\_METADATA attempting internal generator.  CREATE MATERIALIZED VIEW COLLABORATION\_DETAILS\_DIM  LOGGING  TABLESPACE USERS  PCTFREE 10  INITRANS 1  STORAGE  (  INITIAL 65536  NEXT 1048576  MINEXTENTS 1  MAXEXTENTS UNLIMITED  BUFFER\_POOL DEFAULT  )  NOCOMPRESS  NO INMEMORY  NOCACHE  NOPARALLEL  USING INDEX  REFRESH ON DEMAND  FORCE  USING DEFAULT LOCAL ROLLBACK SEGMENT  DISABLE QUERY REWRITE AS  SELECT  CD.ID,  CD.PROJECT\_ID,  PR.CODE,  (decode(CD.PARTNER\_CODE,PT.CODE,PT.NAME,null))PARTNER,  CD.PARTNER\_CODE,  (decode(CD.COLLABORATION\_TYPE\_CODE,CT.CODE,CT.NAME,null))COLLABORATION\_TYPE,  CD.COLLABORATION\_TYPE\_CODE,  decode(CD.START\_PHASE\_CODE, PH.CODE,PH.NAME ,null) START\_PHASE,  CD.START\_PHASE\_CODE,  (decode(CD.END\_PHASE\_CODE,PH2.CODE,PH2.NAME ,null))END\_PHASE,  CD.END\_PHASE\_CODE,  CD."COMMENT"  FROM IPMS\_DATA.COLLABORATION\_DETAILS CD, IPMS\_DATA.PROJECT PR, IPMS\_DATA.COLLABORATION\_TYPE CT, IPMS\_DATA.PARTNER PT, IPMS\_DATA.PHASE PH,  IPMS\_DATA.PHASE PH2  WHERE CD.PROJECT\_ID = PR.ID  and PT.CODE (+)= CD.PARTNER\_CODE  and CT.CODE (+)= CD.COLLABORATION\_TYPE\_CODE  and CD.START\_PHASE\_CODE = PH.CODE (+)  and CD.END\_PHASE\_CODE = PH2.CODE(+);  COMMENT ON MATERIALIZED VIEW COLLABORATION\_DETAILS\_DIM IS 'snapshot table for snapshot IPMS\_REPO.COLLABORATION\_DETAILS\_DIM'; |
| LICENSE\_DETAILS\_DIM |
| CREATE MATERIALIZED VIEW LICENSE\_DETAILS\_DIM  LOGGING  TABLESPACE USERS  PCTFREE 10  INITRANS 1  STORAGE  (  INITIAL 65536  NEXT 1048576  MINEXTENTS 1  MAXEXTENTS UNLIMITED  BUFFER\_POOL DEFAULT  )  NOCOMPRESS  NO INMEMORY  NOCACHE  NOPARALLEL  USING INDEX  REFRESH ON DEMAND  FORCE  USING DEFAULT LOCAL ROLLBACK SEGMENT  DISABLE QUERY REWRITE AS  select  LD.id,  LD.PROJECT\_ID,  PR.CODE,  (DECODE(LD.LICENSOR\_CODE,PT.CODE,PT.NAME,null)) LICENSOR,  LD.LICENSOR\_CODE,  (DECODE(LD.LICENSEE\_CODE,PT2.CODE,PT2.NAME,null)) LICENSEE,  LD.LICENSEE\_CODE,  (DECODE(LD.LICENSE\_TYPE\_CODE,LT.CODE,LT.NAME,null)) LICENSE\_TYPE,  LD.LICENSE\_TYPE\_CODE,  LD.LICENSE\_AGREEMENT\_DATE,  LD.LICENSE\_COMMENT,  LD.PREDECESSOR\_ID  from IPMS\_DATA.LICENSE\_DETAILS LD  left join IPMS\_DATA.PARTNER PT on PT.CODE = LD.LICENSOR\_CODE  left join IPMS\_DATA.PARTNER PT2 on PT2.CODE = LD.LICENSEE\_CODE  left join IPMS\_DATA.project PR on PR.id = LD.PROJECT\_ID  left join IPMS\_DATA.LICENSE\_DETAIL\_TYPE LT on LT.CODE = LD.LICENSE\_TYPE\_CODE;  COMMENT ON MATERIALIZED VIEW LICENSE\_DETAILS\_DIM IS 'snapshot table for snapshot IPMS\_REPO.LICENSE\_DETAILS\_DIM'; |
| PROJECT\_REL\_MSP\_DEV\_DIM |
| CREATE MATERIALIZED VIEW PROJECT\_REL\_MSP\_DEV\_DIM  LOGGING  TABLESPACE USERS  PCTFREE 10  INITRANS 1  STORAGE  (  INITIAL 65536  NEXT 1048576  MINEXTENTS 1  MAXEXTENTS UNLIMITED  BUFFER\_POOL DEFAULT  )  NOCOMPRESS  NO INMEMORY  NOCACHE  NOPARALLEL  USING INDEX  REFRESH ON DEMAND  FORCE  USING DEFAULT LOCAL ROLLBACK SEGMENT  DISABLE QUERY REWRITE AS  select  PRD.ID,  PRD.PROJECT\_ID MSP\_PROJECT\_ID,  PR.CODE MSP\_CODE,  PRD.DEV\_PROJECT\_ID,  PR1.CODE DEV\_PROJECT\_CODE  from ipms\_data.PROJECT\_RELATED\_DEV PRD  left join ipms\_data.project PR on PRD.PROJECT\_ID=PR.ID  left join ipms\_data.project PR1 on PR1.ID=PRD.DEV\_PROJECT\_ID;  COMMENT ON MATERIALIZED VIEW PROJECT\_REL\_MSP\_DEV\_DIM IS 'snapshot table for snapshot IPMS\_REPO.PROJECT\_REL\_MSP\_DEV\_DIM'; |
| PROJECT\_REL\_PREDECESSOR\_DIM |
| CREATE MATERIALIZED VIEW PROJECT\_REL\_PREDECESSOR\_DIM  LOGGING  TABLESPACE USERS  PCTFREE 10  INITRANS 1  STORAGE  (  INITIAL 65536  NEXT 1048576  MINEXTENTS 1  MAXEXTENTS UNLIMITED  BUFFER\_POOL DEFAULT  )  NOCOMPRESS  NO INMEMORY  NOCACHE  NOPARALLEL  USING INDEX  REFRESH ON DEMAND  FORCE  USING DEFAULT LOCAL ROLLBACK SEGMENT  DISABLE QUERY REWRITE AS  select  PRP.ID,  PRP.PROJECT\_ID,  PR.CODE,  PRP.REL\_PREDECESSOR\_PROJECT\_ID,  PR1.CODE PREDECESSOR\_PROJECT\_CODE  from ipms\_data.PROJECT\_RELATED\_PREDECESSOR PRP  left join ipms\_data.project PR on PRP.PROJECT\_ID=PR.ID  left join ipms\_data.project PR1 on PR1.ID=PRP.REL\_PREDECESSOR\_PROJECT\_ID;  COMMENT ON MATERIALIZED VIEW PROJECT\_REL\_PREDECESSOR\_DIM IS 'snapshot table for snapshot IPMS\_REPO.PROJECT\_REL\_PREDECESSOR\_DIM'; |
| PROJECT\_REL\_SUCCESSOR\_DIM |
| CREATE MATERIALIZED VIEW PROJECT\_REL\_SUCCESSOR\_DIM  LOGGING  TABLESPACE USERS  PCTFREE 10  INITRANS 1  STORAGE  (  INITIAL 65536  NEXT 1048576  MINEXTENTS 1  MAXEXTENTS UNLIMITED  BUFFER\_POOL DEFAULT  )  NOCOMPRESS  NO INMEMORY  NOCACHE  NOPARALLEL  USING INDEX  REFRESH ON DEMAND  FORCE  USING DEFAULT LOCAL ROLLBACK SEGMENT  DISABLE QUERY REWRITE AS  select  PRS.ID,  PRS.PROJECT\_ID,  PR.CODE,  PRS.SUCCESSOR\_PROJECT\_ID,  PR1.CODE SUCCESSOR\_PROJECT\_CODE  from ipms\_data.PROJECT\_RELATED\_SUCCESSOR PRS  left join ipms\_data.project PR on PRS.PROJECT\_ID=PR.ID  left join ipms\_data.project PR1 on PR1.ID=PRS.SUCCESSOR\_PROJECT\_ID;  COMMENT ON MATERIALIZED VIEW PROJECT\_REL\_SUCCESSOR\_DIM IS 'snapshot table for snapshot IPMS\_REPO.PROJECT\_REL\_SUCCESSOR\_DIM'; |
| PROJECT\_ROADMAP\_DIM |
| CREATE MATERIALIZED VIEW PROJECT\_ROADMAP\_DIM  LOGGING  TABLESPACE USERS  PCTFREE 10  INITRANS 1  STORAGE  (  INITIAL 65536  NEXT 1048576  MINEXTENTS 1  MAXEXTENTS UNLIMITED  BUFFER\_POOL DEFAULT  )  NOCOMPRESS  NO INMEMORY  NOCACHE  NOPARALLEL  USING INDEX  REFRESH ON DEMAND  FORCE  USING DEFAULT LOCAL ROLLBACK SEGMENT  DISABLE QUERY REWRITE AS  select  PRD.ID,  PRD.PROJECT\_ID,  PR.CODE,  PRD."DATE",  PRD.TOPIC,  PRD.DESCRIPTION  from ipms\_data.PROJECT\_ROADMAP PRD, IPMS\_DATA.PROJECT PR  WHERE PRD.PROJECT\_ID = PR.ID;  COMMENT ON MATERIALIZED VIEW PROJECT\_ROADMAP\_DIM IS 'snapshot table for snapshot IPMS\_REPO.PROJECT\_ROADMAP\_DIM' |
| PROJECT\_DIM |
| DROP MATERIALIZED VIEW PROJECT\_DIM;  CREATE MATERIALIZED VIEW PROJECT\_DIM  LOGGING  TABLESPACE USERS  PCTFREE 10  INITRANS 1  STORAGE  (  INITIAL 65536  NEXT 1048576  MINEXTENTS 1  MAXEXTENTS UNLIMITED  BUFFER\_POOL DEFAULT  )  NOCOMPRESS  NO INMEMORY  NOCACHE  NOPARALLEL  USING INDEX  REFRESH ON DEMAND  FORCE  USING DEFAULT LOCAL ROLLBACK SEGMENT  DISABLE QUERY REWRITE AS  select  p.id,  p.program\_id,  p.code,  p.name,  p.abbreviation as short\_name,  p.state\_code as project\_state\_code,  pstate.name as project\_state,  p.is\_active,  p.is\_lead,  p.development\_phase\_code as le\_overview\_code,  devphase.name as le\_overview,  p.sbe\_code,  sbe.name as sbe,  p.pidt\_bu\_code as gbu\_code,  gbu.name as gbu,  p.ta\_code as therapeutic\_area\_code,  pta.name as therapeutic\_area,  p.area\_code as project\_area\_code,  area.name as project\_area,  p.phase\_code,  phase.name as phase,  p.priority\_code,  prio.name as priority,  cast(category.code as nvarchar2(20)) as lc\_type\_code,  category.name as lc\_type,  cast(subgroup.code as nvarchar2(20)) as subgroup\_code,  subgroup.name as subgroup\_name,  mg.code as maingroup\_code,  mg.name as maingroup\_name,  p.substance\_type\_code,  substance\_type.name as substance\_type,  p.source\_code,  source.name as source,  p.tc\_code as target\_class\_code,  ptc.name as target\_class,  p.to\_code as target\_origin\_code,  pto.name as target\_origin,  p.enpv,  p.npv,  p.peak\_sales,  p.is\_portfolio as portfolio\_project,  p.is\_hpr as hpr,  p.is\_regional,  p.is\_orphan\_drug,  --p.is\_direct\_phase3,  --p.is\_combined\_phase2,  p.is\_collaboration as collaboration,  p.is\_pip as pip\_project,  p.pip\_date,  p.pdco\_date,  p.is\_pdco\_positive,  p.is\_waiver,  p.pip\_activities,  substr(termination.name,1,100) as termination\_name,  p.termination\_code,  p.termination\_date,  p.termination\_reason,  pts\_preclinical.probability as pts\_preclinical,  pts\_phase1.probability as pts\_phase1,  pts\_phase2.probability as pts\_phase2,  pts\_phase3.probability as pts\_phase3,  pts\_submission.probability as pts\_submission,  round(100 \* nvl(pts\_preclinical.probability / 100, 0) \* nvl(pts\_phase1.probability / 100, 0) \* nvl(pts\_phase2.probability / 100, 0) \* nvl(pts\_phase3.probability / 100, 0) \* nvl(pts\_submission.probability / 100, 0)) as pts\_total,  p.review\_date as review\_meeting\_date,  p.details\_partner,  p.details\_progress,  p.progress\_date,  p.details\_criteria,  p.details\_mesh,  p.details\_indication,  p.details\_product,  p.details\_goal,  p.details\_action,  p.details\_competition,  p.details\_patent,  p.details\_sales,  p.details\_pmo,  p.details\_comment,  --p.details\_modality,  ipms\_Data.configuration\_pkg.get\_names\_for\_codes(p.details\_modality) details\_modality,  combination.combination\_code,  combination.combination\_name as combination\_id,  pbay.name as preferred\_bay\_no,  p.id||'-RAW' as p6\_raw\_plan,  p.id||'-CUR' as p6\_current\_plan,  p.id||'-APR' as p6\_approved\_plan,  p.planning\_enabled,  p.ipowner\_code as ip\_owner\_code,  ipowner.name as ip\_owner,  p.d1\_decision\_date as d1\_milestone\_for\_d1\_prj,  pgfd1.goal\_factor as d1\_goal\_factor\_for\_d1\_prj,  p.start\_hts\_date as start\_hts\_milestone,  p.lsa\_date,  pgft.goal\_factor as termination\_goal\_factor,  cast(decode(p.area\_code,'D1',null,p.d2\_compound) as nvarchar2(80)) d2\_compound,  p.succ\_project\_id as successor\_project\_id,  p.predecessor\_project\_id,  p.d3transition\_project\_id,  p.lo\_review\_meeting\_date,  p.lo\_review\_result\_code,  p.trg as therapeutic\_research\_group,  p.target\_gene\_code,  p.er as exploratory\_research,  p.general\_project\_frame,  p.phase\_estimated\_code,  phase\_estimated.name as phase\_estimated,  --p.d2\_planned\_date,  --p.d2\_achieved\_date,  p.ptr\_for\_d2\_code,  p.ptr\_for\_d3\_code,  p.previous\_names,  p.is\_device\_project as device\_project,  p.bpi\_no,  p.project\_device\_type\_code,  pdt.name as project\_device\_type\_name,  substr(p.regulatory\_code,1,instr(p.regulatory\_code||';',';')-1) as regulatory\_code,  rt.name as regulatory\_name,  p.regulatory\_other,  p.device\_phase\_code,  dp.name device\_phase\_name,  p.DETAILS\_OBJECTIVE DET\_OBJECTIVE\_MSP,--promis 604  p.DETAILS\_RATIONALE DET\_RATIONALE\_MSP,  p.DETAILS\_SCOPE DET\_SCOPE\_MSP,  p.DETAILS\_INTENDED\_USE DET\_INTENDED\_USE\_MSP,  p.DETAILS\_BENEFITS DET\_BENEFITS\_MSP,  p.DETAILS\_EXECUTIVE\_SUMMARY DET\_EXECUTIVE\_SUMMARY\_MSP,  p.DETAILS\_RISKS DET\_RISKS\_MSP,  p.DETAILS\_HIGHLIGHTS DET\_HIGHLIGHTS\_MSP,  p.DETAILS\_ACTIVITIES\_EVENTS DET\_ACTIVITIES\_EVENTS\_MSP,  p.DETAILS\_BUDGET DET\_BUDGET\_MSP,  ss.NAME DET\_SAMD\_STATUS\_MSP,  p.DETAILS\_SAMD\_STATUS DET\_SAMD\_STATUS\_CODE\_MSP,  bs.NAME DET\_BUDGET\_STATUS\_MSP,  p.DETAILS\_BUDGET\_STATUS DET\_BUDGET\_STATUS\_CODE\_MSP  from ipms\_data.project p  left join ipms\_data.budget\_status bs on bs.CODE = p.DETAILS\_BUDGET\_STATUS  left join ipms\_data.samd\_status ss on ss.CODE = p.DETAILS\_SAMD\_STATUS  left join ipms\_data.combination\_vw combination on combination.project\_id = p.id  left join ipms\_data.bay\_number pbay on pbay.code = p.bay\_code  left join ipms\_data.project\_state pstate on pstate.code = p.state\_code  left join ipms\_data.display\_state displaystate on displaystate.code = p.display\_state\_code  left join ipms\_data.development\_phase devphase on devphase.code = p.development\_phase\_code  left join ipms\_data.strategic\_business\_entity sbe on sbe.code = p.sbe\_code  left join ipms\_data.global\_business\_unit gbu on gbu.code = p.pidt\_bu\_code  left join ipms\_data.therapeutic\_area pta on pta.code = p.ta\_code  left join ipms\_data.project\_area area on area.code = p.area\_code  left join ipms\_data.phase phase on phase.code = p.phase\_code  left join ipms\_data.priority prio on prio.code = p.priority\_code  left join ipms\_data.project\_category category on (category.code = p.category\_code and category.is\_promis=1)  left join ipms\_data.project\_category subgroup on (subgroup.code = p.project\_group\_code)  left join ipms\_data.maingroup mg on (subgroup.maingroup\_code = mg.code)  left join ipms\_data.substance\_type substance\_type on substance\_type.code = p.substance\_type\_code  left join ipms\_data.project\_source source on source.code = p.source\_code  left join ipms\_data.target\_class ptc on ptc.code = p.tc\_code  left join ipms\_data.target\_origin pto on pto.code = p.to\_code  left join --ipms\_data.termination\_reason termination on termination.code = p.termination\_code  (--PROMIS-748  select tr.code, decode(trg.name,null,null,trg.name||' - ')||tr.name name  from ipms\_data.termination\_reason tr  left join ipms\_data.termination\_reason trg on (tr.ref\_reason\_code=trg.code)  ) termination on (termination.code = p.termination\_code)  left join ipms\_data.phase\_estimated phase\_estimated on phase\_estimated.code = p.phase\_estimated\_code  left join ipms\_data.costs\_probability pts\_preclinical on pts\_preclinical.project\_id = p.id and pts\_preclinical.phase\_code = '1' and pts\_preclinical.scope\_code = 'INT'  left join ipms\_data.costs\_probability pts\_phase1 on pts\_phase1.project\_id = p.id and pts\_phase1.phase\_code = '2' and pts\_phase1.scope\_code = 'INT'  left join ipms\_data.costs\_probability pts\_phase2 on pts\_phase2.project\_id = p.id and pts\_phase2.phase\_code = '34' and pts\_phase2.scope\_code = 'INT'  left join ipms\_data.costs\_probability pts\_phase3 on pts\_phase3.project\_id = p.id and pts\_phase3.phase\_code = '5' and pts\_phase3.scope\_code = 'INT'  left join ipms\_data.costs\_probability pts\_submission on pts\_submission.project\_id = p.id and pts\_submission.phase\_code = '6' and pts\_submission.scope\_code = 'INT'  left join ipms\_data.ipowner ipowner on ipowner.code = p.ipowner\_code  left join ipms\_data.project\_goal\_factor pgft on (p.id = pgft.project\_id) and pgft.milestone\_code='Termn'  left join ipms\_data.project\_goal\_factor pgfd1 on (p.id = pgfd1.project\_id) and pgfd1.milestone\_code='D1' and p.area\_code='D1'  left join ipms\_data.project\_device\_type pdt on (pdt.code = p.project\_device\_type\_code)  left join ipms\_data.regulatory\_type rt on (rt.code = substr(p.regulatory\_code,1,instr(p.regulatory\_code||';',';')-1))  left join ipms\_data.device\_phase dp on (dp.code = p.device\_phase\_code);  COMMENT ON MATERIALIZED VIEW PROJECT\_DIM IS 'snapshot table for snapshot IPMS\_REPO.PROJECT\_DIM'; |
| SOPHIA\_PROJECT\_ACTIVITIES\_VW |
| CREATE OR REPLACE FORCE EDITIONABLE VIEW IPMS\_DATA.SOPHIA\_PROJECT\_ACTIVITIES\_VW (ARTIFICIAL\_ID, PROJECT\_ID, PROJECT\_ACTIVITY\_ID, PROJECT\_ACTIVITY\_NAME, STUDY\_ID, PLAN\_START\_DATE, PLAN\_FINISH\_DATE, ACT\_START\_DATE, ACT\_FINISH\_DATE, WBS\_CATEGORY) AS  select  prj.id||':'||pa.project\_activity\_id||':'||pa.study\_id||':'||mls.wbs\_category artificial\_id,  prj.id as project\_id,  pa.project\_activity\_id,  nvl2(pa.study\_id,pa.study\_id||': ','')||pa.project\_activity\_name as project\_activity\_name,  pa.study\_id,  pa.plan\_start\_date,  pa.plan\_finish\_date,  pa.act\_start\_date,  pa.act\_finish\_date,  mls.wbs\_category  from project\_activities@sophia\_db pa  join project prj on to\_char(pa.project\_id) = to\_char(prj.code)  left join milestone mls on mls.code = pa.decision\_start and mls.is\_active = 1  where pa.decision\_start not in ('D1','D2')  ; |
| PRJ\_AUDIT\_MILESTONES\_SRC\_VW |
| CREATE OR REPLACE FORCE EDITIONABLE VIEW "IPMS\_DATA"."PRJ\_AUDIT\_MILESTONES\_SRC\_VW" ("AUDIT\_ID", "PRJ\_ID", "PARENT\_ID", "MS\_TYPE\_CODE", "MS\_CODE", "MS\_NAME", "MS\_PLAN", "MS\_ACHIEVED") AS  select  audit\_id,  prj\_id,  parent\_id,  ms\_type\_code,  extractvalue(ms\_xml, '/MILESTONE/@CODE') ms\_code,  extractvalue(ms\_xml, '/MILESTONE/@NAME') ms\_name,  extractvalue(ms\_xml, '/MILESTONE/@PLAN') ms\_plan,  extractvalue(ms\_xml, '/MILESTONE/@ACHIEVED') ms\_achieved  from (  select 'DEC' ms\_type\_code,  pa.id audit\_id,  pa.project\_id prj\_id,  pa.parent\_id,  ms.column\_value ms\_xml  from project\_audit pa, xmltable('/PROJECT/DECISIONS/MILESTONE' passing pa.details) ms  union all -- Promis 604  select 'DEC-SAMD' ms\_type\_code,  pa.id audit\_id,  pa.project\_id prj\_id,  pa.parent\_id,  ms.column\_value ms\_xml  from project\_audit pa, xmltable('/PROJECT/DECISIONS/MILESTONE' passing pa.details) ms  union all  select 'DEV',  pa.id,  pa.project\_id,  pa.parent\_id,  ms.column\_value  from project\_audit pa, xmltable('/PROJECT/DEVELOPMENT/MILESTONE' passing pa.details) ms  union all  select 'REG',  pa.id,  pa.project\_id,  pa.parent\_id,  ms.column\_value  from project\_audit pa, xmltable('/PROJECT/REGIONAL/MILESTONE' passing pa.details) ms  );  GRANT SELECT ON "IPMS\_DATA"."PRJ\_AUDIT\_MILESTONES\_SRC\_VW" TO "PMO\_READ";  GRANT SELECT ON "IPMS\_DATA"."PRJ\_AUDIT\_MILESTONES\_SRC\_VW" TO "IPMS\_REPO"; |

1. Procedure changes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Package Name | Procedure/Function | Type | Line No | Changes | Remarks |
| audit\_pkg | audit\_project\_changes | Procedure | 22 | and ml.type\_code IN('DEC','DEC-SAMD')-- PROMIS- 604 | amend where clause |
|  | release\_all\_changes\_to\_qplan | Procedure | 150 & 151 | and (area\_code !='D1' --PROMIS-691  or area\_code !='SAMD') --PROMIS 604 | amend where clause |
|  | pidt\_audit\_job | Procedure | 285 | and prj.area\_code in ('PRE-POT','POST-POT','D2-PRJ', 'SAMD') --PROMIS 604 | amend where clause |
|  |  |  | 303 | and area\_code in ('PRE-POT','POST-POT','D2-PRJ', 'SAMD') --PROMIS 604 | amend where clause |
|  |  |  | 323 | and area\_code in ('PRE-POT','POST-POT','D2-PRJ', 'SAMD')--PROMIS 604 | amend where clause |
|  |  |  | 378 | and area\_code in ('PRE-POT','POST-POT','D2-PRJ', 'SAMD') --PROMIS 604 | amend where clause |
|  |  |  | 397 | and prj.area\_code in ('PRE-POT','POST-POT','D2-PRJ', 'SAMD'); --PROMIS 604 | amend where clause |
| Generic\_timeline\_pkg | is\_generic\_timeline\_possible | Function | 83 | v\_project\_rec.area\_code != 'SAMD'-- PROMIS 604 | amend if condition |
|  | calculate\_generic\_timeline | Procedure | 184 | or v\_project\_rec.area\_code = 'SAMD' --PROMIS 604 | amend if condition |
|  | calculate\_generic\_timelines | Procedure | 473 | or p.area\_code = 'SAMD'--PROMIS 604 | amend where clause |
| Import\_pkg | launch\_all | Procedure | 1241 | order by decode(prj.area\_code,'PRE-POT','A1','POST-POT','A2','PRD-MNT','A3','D2-PRJ','A4','D3-TR','A5', 'SAMD','A6', prj.area\_code), prj.id--PROMIS 604 | amend order by clause |
|  |  |  | 1321 | where nvl(prj.area\_code,'#####') NOT in ('PRE-POT','POST-POT','D2-PRJ','D1','SAMD')--PROMIS 604 | amend where clause |
|  | submit\_masterdata | Function | 2154 | if v\_project.area\_code in ('PRE-POT','POST-POT','D2-PRJ', 'SAMD') then-- PROMIS 604 | amend if condition |
| mesaage\_pkg | send\_users | Procedure | 361 | v\_nextprojecttype := 'SAMD'; | change if block |
|  |  |  | 362 | elsif v\_projecttype = 'SAMD' then --PROMIS 604" | add else if condition |
| plan\_assumptions\_pkg | starts\_bpm | Procedure | 94 | where code = 'RESERVED-PT-D2-PRJ' or code = 'RESERVED-PT-SAMD';--PROMIS 604 | amend where clause |
| project\_pkg | xml | Function | 402 | decode(area\_code,'D2-PRJ',get\_element('templateCode','D2TMPL'),'SAMD',get\_element('templateCode','SAMD\_TMPL'),'')|| -- PROMIS 604 | add template |
|  | adjust | Procedure | 440 | if v\_area\_code !='SAMD' then ---- PROMIS 604 | add if condition |
|  |  |  | 486 | end if; -- PROMIS 604 | end if condition |
| Qplan\_pkg | get\_ProjectFinishedDate\_value | Procedure | 423 | and prj.area\_code !='SAMD' --- Promis 604 | add where clause |
| Estimates\_pkg | setup | procedure | 1141 to 1150 | /\*JIRA: PROMIS-605 Strt Deactiating Cost split in case the project is late/early development and there is poc milestone\*/  --nvl(dev.code, prj.development\_phase\_code) as development\_phase\_code,  --decode(dev.code, '0005', poc.milestone\_date, leproc.start\_date) as start\_date,  --decode(dev.code, '0002', poc.milestone\_date, leproc.finish\_date) as finish\_date,    prj.development\_phase\_code as development\_phase\_code,  leproc.start\_date as start\_date,  leproc.finish\_date as finish\_date,  /\*JIRA: PROMIS-605 end Deactiating Cost split in case the project is late/early development and there is poc milestone\*/ | Amend merge latest\_estimate statement |