```
CREATE SEQUENCE SEQ_type;
CREATE SEQUENCE SEQ_project;
CREATE SEQUENCE SEQ_process;
CREATE SEQUENCE SEQ_activity;
CREATE SEQUENCE SEQ_task;
CREATE SEQUENCE SEQ_rol;
CREATE SEQUENCE SEQ_participant;
CREATE SEQUENCE SEQ_inproduct;
CREATE SEQUENCE SEQ_outproduct;
CREATE SEQUENCE SEQ_tecnic;
CREATE SEQUENCE SEQ_profile;

CREATE TABLE Type
(
	pk	integer PRIMARY KEY DEFAULT nextval('SEQ_type'),
	name	varchar(255) NOT NULL,
	descr	text
);


CREATE TABLE Project
(
	pk	integer PRIMARY KEY DEFAULT nextval('SEQ_project'),
	fktype	integer NOT NULL REFERENCES Type(pk),
	name	varchar(255) NOT NULL,
	descr	text
);


CREATE TABLE Proccess
(
	pk	integer PRIMARY KEY DEFAULT nextval('SEQ_process'),
	name	varchar(255) NOT NULL,
	descr	text
);


CREATE TABLE Activity
(
	pk	integer PRIMARY KEY DEFAULT nextval('SEQ_activity'),
	fkpro	integer NOT NULL REFERENCES Proccess(pk),
	name	varchar(255) NOT NULL,
	descr	text
);
```

```
CREATE TABLE Task
(
	pk	integer PRIMARY KEY DEFAULT nextval('SEQ_task'),
	fkact	integer NOT NULL REFERENCES Activity(pk),
	name	varchar(255) NOT NULL,
	descr	text
);


CREATE TABLE OutProduct
(
	pk	integer PRIMARY KEY DEFAULT nextval('SEQ_outproduct'),
	name	varchar(255) NOT NULL,
	descr	text,
	example varchar(255)

);


CREATE TABLE InProduct
(
	pk	integer PRIMARY KEY DEFAULT nextval('SEQ_inproduct'),
	name	varchar(255) NOT NULL,
	descr	text,
	example varchar(255)

);


CREATE TABLE Tecnic
(
	pk	integer PRIMARY KEY DEFAULT nextval('SEQ_tecnic'),
	name	varchar(255) NOT NULL,
	descr	text,
	example varchar(255)

);


CREATE TABLE Profile
(
   pk	integer PRIMARY KEY DEFAULT nextval('SEQ_profile'),
   name	varchar(255) NOT NULL,
```

```
    descr text
);


CREATE TABLE Rol
(
      pk      integer PRIMARY KEY DEFAULT nextval('SEQ_rol'),
      name    varchar(255) NOT NULL,
      fkprofile integer NOT NULL REFERENCES Profile(pk),
      descr   text
);



CREATE TABLE Person

(

  pk       integer PRIMARY KEY DEFAULT nextval('SEQ_Participant'),
  name      varchar(255) NOT NULL,

  surname   varchar(255) NOT NULL,

  bdate     date,

  photo     varchar(255),

  remarks   text
,
  uid       integer NOT NULL
);


CREATE TABLE estado
(
      pk      serial  PRIMARY KEY,
      estado  varchar(255) NOT NULL
);


CREATE TABLE REL_taop
(
      pk      serial  PRIMARY KEY,
      fktask  integer NOT NULL REFERENCES Task(pk),
      fkop    integer NOT NULL REFERENCES OutProduct(pk),
```

```
	UNIQUE  (fktask, fkop)
);


CREATE TABLE REL_tain
(
	pk	serial  PRIMARY KEY,
	fktask  integer NOT NULL REFERENCES Task(pk),
	fkip    integer NOT NULL REFERENCES InProduct(pk),
	UNIQUE  (fktask, fkip)
);


CREATE TABLE REL_paropro
(
	pk	serial  PRIMARY KEY,
	fkpar   integer NOT NULL REFERENCES Person(pk) ON DELETE CASCADE,
	fkpro   integer NOT NULL REFERENCES Project(pk) ON DELETE CASCADE,
	fkrol   integer NOT NULL REFERENCES Rol(pk) ON DELETE CASCADE,
	UNIQUE  (fkpar,fkpro,fkrol)
);


CREATE TABLE REL_protaro
(
	pk	serial PRIMARY KEY,
	fkpro   integer NOT NULL REFERENCES Project(pk),
	fktask  integer NOT NULL REFERENCES Task(pk),
	fkrol   integer REFERENCES Rol(pk),
	estado  integer REFERENCES estado(pk),
	p_out   varchar(255),
	fecha   date,
	f_fin   date,
	notas   text,
	UNIQUE  (fkpro,fktask,fkrol)
);

CREATE TABLE REL_tyta
(
	pk	serial PRIMARY KEY,
	fktyp   integer NOT NULL REFERENCES Type(pk) ON DELETE CASCADE,
	fktask  integer NOT NULL REFERENCES Task(pk) ON DELETE CASCADE,
	UNIQUE  (fktyp,fktask)
);
```

```
CREATE TABLE REL_tate
(
	pk	serial PRIMARY KEY,
	fktask	integer NOT NULL REFERENCES Task(pk),
	fktec	integer NOT NULL REFERENCES Tecnic(pk),
	UNIQUE	(fktask,fktec)
);


CREATE TABLE REL_rota
(
	pk	serial PRIMARY KEY,
	fktask	integer NOT NULL REFERENCES Task(pk),
	fkrol	integer NOT NULL REFERENCES Rol(pk),
	UNIQUE	(fktask,fkrol)
);




CREATE TABLE Telephone

(

  pk	serial PRIMARY KEY,

  fkperson	integer NOT NULL REFERENCES Person(pk) ON DELETE CASCADE,

  telephone varchar(15) NOT NULL,

  descr	varchar(255) NOT NULL

);




CREATE TABLE Address
(

  pk	serial PRIMARY KEY,

  fkperson	integer NOT NULL REFERENCES Person(pk) ON DELETE CASCADE,
```

```
  address   varchar(255) NOT NULL,

  descr     varchar(255) NOT NULL

);



CREATE TABLE Email
(

  pk        serial PRIMARY KEY,

  fkperson  integer NOT NULL REFERENCES Person(pk) ON DELETE CASCADE,

  email     varchar(255) NOT NULL,

  descr     varchar(255) NOT NULL

);



CREATE TABLE Remarks
(

  pk        serial PRIMARY KEY,

  fkperson  integer NOT NULL REFERENCES Person(pk) ON DELETE CASCADE,

  descr     varchar(255) NOT NULL,

  remarks   text

);


CREATE TABLE Message
(
  pk      serial PRIMARY KEY,
  fromuid integer NOT NULL,
  touid   integer NOT NULL,
  tstamp  timestamp NOT NULL,
```

```
  subject varchar(255) NOT NULL,
  body    text,
  deleted char NOT NULL DEFAULT 'N'
);

CREATE TABLE todo
(
  pk  serial PRIMARY KEY,
  uid integer NOT NULL,
  title varchar(255) NOT NULL,
  tstamp timestamp,
  endt   timestamp,
  priority integer NOT NULL DEFAULT '3',
  remarks  text,
  file     varchar
);


CREATE VIEW VIEW_prorolusu AS
      SELECT REL_paropro.fkpro AS fkpro, REL_paropro.fkrol AS fkrol, Person.uid AS uid
   FROM
      REL_paropro, Person
   WHERE
      REL_paropro.fkpar=Person.pk;


CREATE VIEW VIEW_proacttas AS
  SELECT task.pk AS pk,
    proccess.name as pname, activity.name as aname, task.name AS tname
  FROM
     proccess, activity, task
  WHERE
     activity.fkpro=proccess.pk
    AND
     task.fkact=activity.pk;


CREATE VIEW VIEW_prorolper AS
  SELECT
    rel_protaro.fkpro as fkpro, Project.name as project, Rol.name as rol, task.name as task, estado.estado as estado,
    rel_protaro.p_out as p_out, rel_protaro.fecha as fecha,
    rel_protaro.f_fin as f_fin, rel_protaro.notas as notas, person.uid AS uid
  FROM
     REL_protaro left outer join rel_paropro on (rel_protaro.fkpro=rel_paropro.fkpro
     and rel_protaro.fkrol=rel_paropro.fkrol) left outer join Person on rel_paropro.fkpar=person.pk,
```

```
      Project, Rol, Task, estado
   WHERE
      rel_protaro.fkpro=project.pk AND rel_protaro.fkrol=rol.pk
      AND rel_protaro.fktask=task.pk and rel_protaro.estado=estado.pk;



CREATE VIEW VIEW_tastec AS
   SELECT
     Task.pk as pk, Task.name as tname, proccess.name as pname, activity.name as aname, Tecnic.pk AS fktec, Tecnic.name AS tecname, Tecnic.descr AS descr,Tecnic.example AS example
   FROM
      proccess, activity, Task, Tecnic
   WHERE
      activity.fkpro=proccess.pk
     AND
      task.fkact=activity.pk
     AND
      Task.pk=REL_tate.fktask
     AND
      Tecnic.pk=REL_tate.fktec;

CREATE VIEW VIEW_tasrol AS
   SELECT
     Task.pk as pk, Task.name as tname, proccess.name as pname, activity.name as aname, Rol.pk AS fkrol, Rol.name AS rname
   FROM
      proccess, activity, Task, Rol
   WHERE
      activity.fkpro=proccess.pk
     AND
      task.fkact=activity.pk
     AND
      Task.pk=REL_rota.fktask
     AND
      Rol.pk=REL_rota.fkrol;

CREATE VIEW VIEW_tasip AS
   SELECT task.pk AS pk,
     task.name AS tname, proccess.name as pname, activity.name as aname,InProduct.pk as fkip, InProduct.name AS ipname, InProduct.descr AS descr, InProduct.example AS example
   FROM
      proccess, activity, task, InProduct
   WHERE
       activity.fkpro=proccess.pk
     AND
      task.fkact=activity.pk
     AND
```

```
        Task.pk=REL_tain.fktask
       AND
        InProduct.pk=REL_tain.fkip;

CREATE VIEW VIEW_tasop AS
  SELECT task.pk AS pk,
    task.name AS tname, proccess.name as pname, activity.name as aname,OutProduct.pk AS fkop, OutProduct.name AS opname, OutProduct.descr AS descr, OutProduct.example AS example
  FROM
     proccess, activity, task, OutProduct
  WHERE
      activity.fkpro=proccess.pk
     AND
      task.fkact=activity.pk
     AND
      Task.pk=REL_taop.fktask
     AND
      OutProduct.pk=REL_taop.fkop;

CREATE VIEW VIEW_tec AS
  SELECT
     Tecnic.pk AS fktec, Tecnic.name AS tecname, Tecnic.descr AS descr,Tecnic.example AS example
  FROM
     Tecnic;

CREATE VIEW VIEW_ip AS
  SELECT
      InProduct.pk as fkip, InProduct.name AS ipname, InProduct.descr AS descr, InProduct.example AS example
  FROM
     InProduct;

CREATE VIEW VIEW_op AS
  SELECT
      OutProduct.pk AS fkop, OutProduct.name AS opname, OutProduct.descr AS descr, OutProduct.example AS example
  FROM
      OutProduct;



CREATE OR REPLACE FUNCTION func_protaro_ins () RETURNS trigger AS'
   BEGIN

      INSERT INTO REL_protaro (fkpro, fktask, fkrol) SELECT NEW.pk as fkpro, fktask, fkrol FROM VIEW_tytaro WHERE fkpro=NEW.fktype;
      UPDATE REL_protaro SET estado=1 WHERE fkpro=NEW.pk;
      RETURN NEW;
   END;
```

```
'LANGUAGE 'plpgsql';


CREATE OR REPLACE FUNCTION func_protaro_upd () RETURNS trigger AS'
   BEGIN

      DELETE FROM REL_protaro WHERE fkpro = OLD.pk;

      INSERT INTO REL_protaro (fkpro, fktask, fkrol) SELECT NEW.pk as fkpro, fktask, fkrol FROM VIEW_tytaro WHERE fkpro=NEW.fktype;
      RETURN NEW;

   END;
'LANGUAGE 'plpgsql';


CREATE OR REPLACE FUNCTION func_protaro_DEL () RETURNS trigger AS'
   BEGIN

      DELETE FROM REL_protaro WHERE fkpro = OLD.pk;
      RETURN OLD;

   END;
'LANGUAGE 'plpgsql';


CREATE TRIGGER tri_protaro AFTER  INSERT
   ON project FOR EACH ROW
   EXECUTE PROCEDURE  func_protaro_ins ();


CREATE TRIGGER tri_protaro_del BEFORE DELETE
   ON project FOR EACH ROW
   EXECUTE PROCEDURE  func_protaro_DEL ();


CREATE TRIGGER tri_protaro_upd AFTER UPDATE
   ON project FOR EACH ROW
   EXECUTE PROCEDURE  func_protaro_upd ();


CREATE VIEW VIEW_tytaro AS
      SELECT REL_tyta.fktyp AS fkpro, REL_tyta.fktask AS fktask, REL_rota.fkrol AS fkrol
   FROM
      REL_tyta, REL_rota
   WHERE
```

```
REL_tyta.fktask=REL_rota.fktask;
```