

## <PostgreSQL - cs421> resetDB

```
/*
 * Initialize the Database.
 * Warning - upon execution, this script will drop all tables in the database,
 * then recreate them with the default values.
 */

-- Cascade to avoid integrity constraints from preventing successful DROP.
drop table if exists client cascade;
drop table if exists businessowner cascade;
drop table if exists developer cascade;
drop table if exists employee cascade;
drop table if exists person cascade;
drop table if exists clientcompany cascade;
drop table if exists team cascade;
drop table if exists project cascade;
drop table if exists projectassignment cascade;
drop table if exists ticket cascade;
drop table if exists lowticket cascade;
drop table if exists highticket cascade;
drop table if exists sprint cascade;
drop table if exists ticketsprintassignment cascade;
drop table if exists clientstakeholder cascade;
drop table if exists projectbowner cascade;
drop table if exists ticketdeveloperassignment cascade;
drop table if exists accountablebowner cascade;

-- Generic Person entity
create table person(
    pid integer,
    name varchar(50),
    phonenumber char(15),
    email varchar(100),
    primary key (pid)
);

-- Team. Used to store name of team, and enforce various integrity
constraints.
create table team (
    teamname varchar(50),
    primary key (teamname)
);
```

<PostgreSQL - cs421> resetDB

```
-- Employee. Extends Person.
create table employee(
    pid integer,
    teamname varchar(50),
    title varchar(50),
    primary key (pid),
    foreign key (pid) references person(pid),
    foreign key (teamname) references team(teamname)
);

-- Business owner. Extends Employee.
create table businessowner(
    pid integer,
    primary key (pid),
    foreign key (pid) references employee(pid)
);

-- Developer. Extends Employee.
create table developer(
    pid integer,
    primary key (pid),
    foreign key (pid) references employee(pid)
);

-- Client company
create table clientcompany(
    ccid integer,
    companyname varchar(50),
    pointbudget integer,
    primary key (ccid)
);

-- Client. Extends person, references Client Company
create table client(
    pid integer,
    ccid integer not null,
    primary key (pid),
    foreign key (pid) references person(pid),
    foreign key (ccid) references clientcompany(ccid)
```

<PostgreSQL - cs421> resetDB

);

-- Project. References Client Company.

```
create table project(  
    projid integer,  
    ccid integer not null, -- ID of client company which initialized project  
    description text,  
    datecreated date,  
    name varchar(50),  
    primary key (projid),  
    foreign key (ccid) references clientcompany(ccid)  
);
```

-- Project assignment. Relationship between Project and Employee

```
create table projectassignment(  
    projid integer,  
    pid integer,  
    role varchar(50),  
    primary key (projid, pid),  
    foreign key (projid) references project(projid),  
    foreign key (pid) references employee(pid)  
);
```

-- Generic Ticket.

```
create table ticket (  
    tid integer,  
    urgency smallint, -- note: urgencies are restricted between 1 and 5.  
    datecreated date,  
    status varchar(50),  
    description text,  
    primary key (tid)  
);
```

-- High level ticket. Extends ticket. References Project (key & participation constraint), references client (key & participation constraint)

```
create table highticket (  
    tid integer,  
    projid integer not null,  
    clientcreatorid integer not null,  
    primary key (tid),
```

<PostgreSQL - cs421> resetDB

```
foreign key (tid) references ticket(tid),
foreign key (projid) references project(projid),
foreign key (clientcreatorid) references client(pid)
);

-- Low level ticket. Extends ticket. References Business owner (key and
participation constraint)
create table lowticket (
    tid integer,
    bownercreatorid integer not null,
    highticketid integer not null, --High level ticket this low-level ticket
was partitioned from
    pointestimate smallint,
    primary key (tid),
    foreign key (tid) references ticket(tid),
    foreign key (highticketid) references highticket(tid),
    foreign key (bownercreatorid) references businessowner(pid)
);

-- sprint. Weak entity - depends on team. Identified by sprint number, and
creator team.
create table sprint(
    sprintnumber smallint,
    teamname varchar(50),
    startdate date,
    enddate date,
    pointgoal smallint,
    primary key (sprintnumber, teamname),
    foreign key (teamname) references team(teamname)
);

-- Assignmet of low level ticket to a sprint. many-many relationship.
create table ticketsprintassignment(
    tid integer,
    sprintnumber smallint,
    teamname varchar(50),
    datescheduled date,
    primary key (tid, sprintnumber, teamname),
    foreign key (tid) references lowticket(tid),
    foreign key (sprintnumber, teamname) references sprint(sprintnumber,
teamname)
```

<PostgreSQL - cs421> resetDB

);

-- Assignment of low level ticket to a developer. One-many relationship.

```
create table ticketdeveloperassignment(  
    tid integer,  
    devid integer, --id of developer to whom ticket is assigned  
    dateassigned date,  
    primary key (tid),  
    foreign key (tid) references lowticket(tid)  
);
```

```
create table clientstakeholder(  
    projid integer,  
    pid integer,  
    role varchar(50),  
    primary key (projid,pid),  
    foreign key (projid) references project(projid),  
    foreign key (pid) references client(pid)  
);
```

-- Keep track of which business owner is accountable for which project

```
create table accountableowner(  
    projid integer,  
    pid integer,  
    primary key (projid, pid),  
    foreign key (projid) references project(projid),  
    foreign key (pid) references businessowner(pid)  
);
```