**--Create the tables with constraints**

**CREATE TABLE public.category (**

**id SERIAL PRIMARY KEY,**

**photo CHARACTER(128) NOT NULL,**

**name CHARACTER(32) NOT NULL,**

**description CHARACTER(256) NOT NULL**

**);**

**CREATE TABLE public.user (**

**id SERIAL PRIMARY KEY,**

**username CHARACTER(32) UNIQUE NOT NULL,**

**password CHARACTER(128) NOT NULL,**

**email CHARACTER(128),**

**phone CHARACTER(8),**

**name CHARACTER(64) NOT NULL,**

**bio CHARACTER(1000),**

**created\_time TIMESTAMP WITH TIME ZONE NOT NULL,**

**last\_logged\_in TIMESTAMP WITH TIME ZONE,**

**role CHARACTER(32) NOT NULL,**

**CHECK(role = 'admin' OR role='user')**

**);**

**-- make trigger to remove expired tokens**

**CREATE TABLE public.user\_auth\_tokens (**

**id SERIAL PRIMARY KEY,**

**selector CHARACTER(12) UNIQUE,**

**token CHARACTER(64),**

**userid INTEGER REFERENCES public.user(id) ON DELETE CASCADE NOT NULL,**

**expires TIMESTAMP WITH TIME ZONE NOT NULL**

**);**

**CREATE TABLE public.task (**

**id SERIAL PRIMARY KEY,**

**name CHARACTER(128) NOT NULL,**

**description CHARACTER(1024) NOT NULL,**

**postal\_code INTEGER NOT NULL,**

**location CHARACTER(128) NOT NULL,**

**task\_start\_time TIMESTAMP WITH TIME ZONE NOT NULL,**

**task\_end\_time TIMESTAMP WITH TIME ZONE NOT NULL,**

**listing\_price MONEY,**

**created\_time TIMESTAMP WITH TIME ZONE NOT NULL,**

**updated\_time TIMESTAMP WITH TIME ZONE,**

**status CHARACTER(32) NOT NULL,**

**bid\_picked BOOLEAN NOT NULL,**

**category\_id INTEGER REFERENCES public.category(id) ON DELETE CASCADE NOT NULL,**

**creator\_id INTEGER REFERENCES public.user(id) ON DELETE CASCADE NOT NULL,**

**CHECK(status = 'pending' OR status='completed'),**

**CHECK (task\_end\_time >= task\_start\_time),**

**CHECK (updated\_time >= created\_time)**

**);**

**CREATE TABLE public.bid (**

**id SERIAL PRIMARY KEY,**

**amount MONEY NOT NULL,**

**bid\_time TIMESTAMP WITH TIME ZONE NOT NULL,**

**selected BOOLEAN NOT NULL DEFAULT FALSE,**

**user\_id INTEGER REFERENCES public.user(id) ON DELETE CASCADE NOT NULL,**

**task\_id INTEGER REFERENCES public.task(id) ON DELETE CASCADE NOT NULL**

**);**

**CREATE TABLE public.comment (**

**id SERIAL PRIMARY KEY,**

**comment CHARACTER(128) NOT NULL,**

**created\_time TIMESTAMP NOT NULL,**

**user\_id INTEGER REFERENCES public.user(id) ON DELETE CASCADE NOT NULL,**

**task\_id INTEGER REFERENCES public.task(id) ON DELETE CASCADE NOT NULL**

**);**

**-- function for generating random strings**

**CREATE OR REPLACE FUNCTION random\_string(length INTEGER) returns TEXT AS**

**$$**

**DECLARE**

**chars TEXT[] := '{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z}';**

**result TEXT := '';**

**i INTEGER := 0;**

**BEGIN**

**IF length < 0 THEN**

**RAISE EXCEPTION 'Given length cannot be less than 0';**

**END IF;**

**FOR i IN 1..length LOOP**

**result := result || chars[1+RANDOM()\*(array\_length(chars, 1)-1)];**

**END LOOP;**

**RETURN result;**

**END;**

**$$ LANGUAGE plpgsql;**