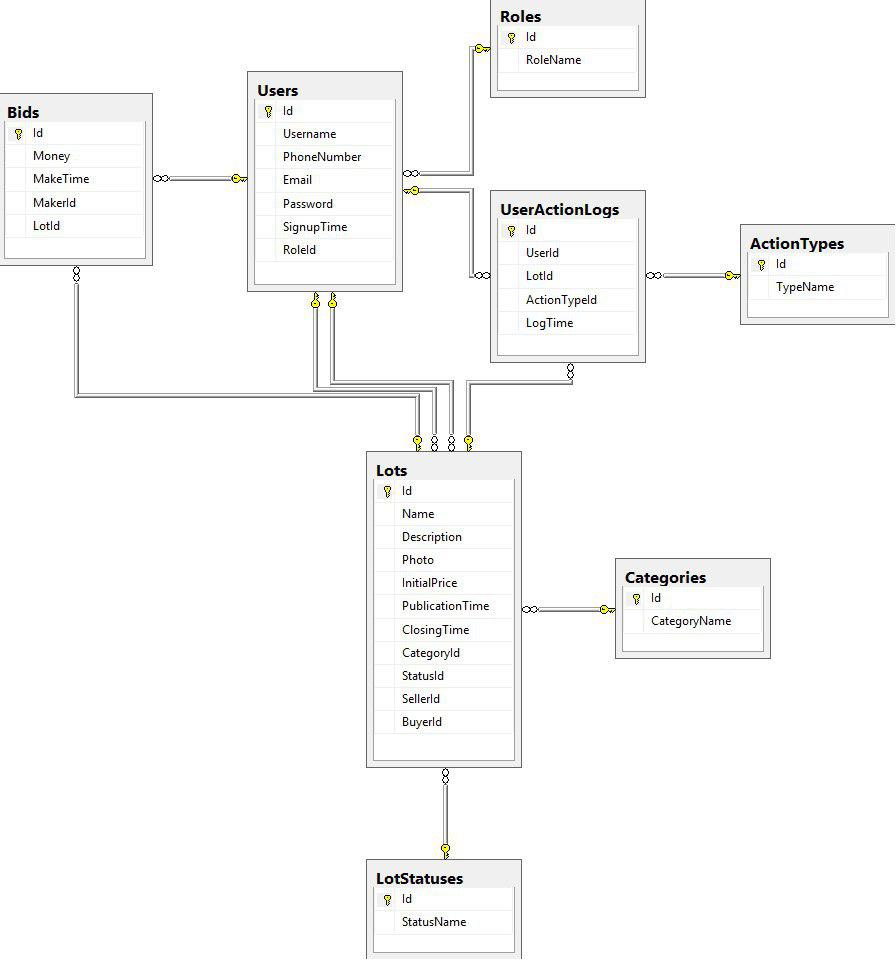


***Database:***



procedure remove\_signature(current\_user\_id in integer, signature\_id in integer)

is

creator\_id integer;

begin

select c.creator\_id into creator\_id from contracts c, signatures s where s.contract\_id = c.id and s.id = signature\_id;

if not is\_user\_admin(current\_user\_id) and creator\_id <> current\_user\_id then

raise\_application\_error(-20005, 'Not admin');

end if;

delete from signatures where id=signature\_id;

end remove\_signature;

procedure remove\_contract(current\_user\_id in integer, contract\_id in integer)

is

begin

if not is\_user\_admin(current\_user\_id) then

raise\_application\_error(-20005, 'Not admin');

end if;

delete from contracts where id=contract\_id;

end remove\_contract;

procedure remove\_status(current\_user\_id in integer, status\_id in integer)

is

begin

if not is\_user\_admin(current\_user\_id) then

raise\_application\_error(-20005, 'Not admin');

end if;

delete from statuses where id=status\_id;

end remove\_status;

procedure remove\_currency(current\_user\_id in integer, currency\_id in integer)

is

begin

if not is\_user\_admin(current\_user\_id) then

raise\_application\_error(-20005, 'Not admin');

end if;

delete from currencies where id=currency\_id;

end remove\_currency;

function is\_user\_admin(id in integer) return boolean

is

admin\_count integer;

begin

select count(\*) into admin\_count from users u where u.id=id and is\_admin='Y';

if admin\_count = 0 then

return false;

end if;

return true;

end is\_user\_admin;

create table statuses

(

id integer generated always as identity primary key,

name nvarchar2(20) not null,

constraint unique\_status\_name unique(name)

);

create table parties (

id integer primary key,

full\_name nvarchar2(150) not null,

phone nvarchar2(20) not null,

address nvarchar2(100) not null,

company\_phone\_fax nvarchar2(20) null,

company\_name nvarchar2(250) null,

company\_address nvarchar2(100) null,

UPN integer null

);

create table users (

id integer primary key,

username nvarchar2(50) not null,

party\_id integer null,

is\_admin char(1) default('N'),

email nvarchar2(50) not null,

phone\_number nvarchar2(20) not null,

password\_hash nvarchar2(50) not null,

constraint fk\_user\_party foreign key (party\_id) references parties(id) on delete set null,

constraint unique\_name unique(username)

);

create table currencies (

id integer generated always as identity primary key,

short\_name nvarchar2(5) not null,

full\_name nvarchar2(20) not null,

to\_USD\_factor decimal(38, 25)

);

create table contracts (

id integer generated always as identity primary key,

creator\_id integer not null,

text blob not null,

stake integer not null,

currency\_id integer not null,

status\_id integer not null,

start\_time timestamp not null,

expiry\_time date not null,

constraint fk\_contract\_currency foreign key (currency\_id) references currencies(id),

constraint fk\_contract\_status foreign key (status\_id) references statuses(id),

constraint fk\_contract\_user foreign key (creator\_id) references users(id)

);

create table signatures (

id integer generated always as identity primary key,

party\_id integer not null,

contract\_id integer not null,

signed char(1) default('N'),

constraint fk\_signature\_party foreign key (party\_id) references parties(id),

constraint fk\_signature\_contract foreign key (contract\_id) references contracts(id) on delete cascade,

constraint unique\_signature unique(party\_id, contract\_id)

);

create table action\_types (

id integer generated always as identity primary key,

type\_name nvarchar2(30) not null,

constraint unique\_type\_name unique(type\_name)

);

create table logs (

id integer generated always as identity primary key,

who\_id integer null,

when date not null,

action\_type\_id integer not null,

contract\_id integer null,

user\_id integer null,

constraint fk\_log\_actor foreign key (who\_id) references users(id) on delete cascade,

constraint fk\_log\_action\_type foreign key (action\_type\_id) references action\_types(id),

constraint fk\_log\_user foreign key (user\_id) references users(id) on delete cascade,

constraint fk\_log\_contract foreign key (contract\_id) references contracts(id) on delete cascade

);

create sequence user\_pk start with 1;

create sequence party\_pk start with 1;