```
CREATE TABLE users
(
            username varchar primary key CHECK(char_length(username) between 3
and 20),
            password varchar NOT NULL CHECK(char_length(password) between 5 and
15),
            email varchar NOT NULL UNIQUE CHECK (char_length(email) between 4 and
30),
            firstName varchar NOT NULL CHECK(char_length(firstName) between 2 and
15),
            lastName varchar NOT NULL CHECK(char_length(lastName) between 2 and
20),
            address varchar NOT NULL CHECK(char_length(address) between 5 and
60),
            type varchar NOT NULL CHECK(type = 'admin' or type = 'user')
)
;
select * from users;
CREATE TABLE item
(
            id serial PRIMARY KEY,
            category varchar NOT NULL,
            name varchar NOT NULL,
            description varchar(60),
            price decimal NOT NULL CHECK(price > 0),
            avgRev decimal CHECK(avgRev between 1 and 5)
)
;

CREATE TABLE keyword
(
            itemID int NOT NULL,
            keyword varchar(20) NOT NULL
)
;

CREATE TABLE review
(
            itemId int NOT NULL,
            username varchar NOT NULL,
            rating smallint NOT NULL CHECK(rating between 1 and 5)
)
;

CREATE TABLE orders
(
            orderID serial PRIMARY KEY,
            username varchar NOT NULL,
            itemListId int NOT NULL,
            date timestamp NOT NULL,
            status varchar NOT NULL CHECK(status in('Awaiting process',
'Delivered', 'Cancelled'))
)
;

CREATE TABLE itemList
(
            listID SERIAL,
            itemID int NOT NULL
```

```sql
)
;

alter table itemList
           add foreign key (itemID)
           references item (id)
           on delete restrict
           on update restrict
;

alter table keyword
           add foreign key (itemID)
           references item (id)
           on delete restrict
           on update restrict
;

alter table orders
           add foreign key (itemListId)
           references itemlist (listid)
           on delete restrict
;

alter table review
           add foreign key (username)
           references users (username)
           on delete restrict
           on update restrict
;

alter table review
           add foreign key (itemId)
           references item (id)
           on delete restrict
           on update restrict
;

alter table orders
           add foreign key (username)
           references users (username)
           on delete restrict
           on update restrict
;

--Trigger--

create function avgRating()
returns trigger as
           $$
declare

  itemIDvar review.itemId%type;

begin

 if TG_OP = 'DELETE' then
            itemIDvar := old.itemId;
 else
            itemIDvar := new.itemId;
 end if
 ;

 update item
   set  avgRev = (
                        select avg(rating)
                                                                   from review
                                                                  where itemId =
itemIDvar
                    )
  where id = itemIDvar
  ;

  if TG_OP = 'DELETE' then
            return old;
 else
            return new;
 end if;
end;
$$ language plpgsql
;



create trigger reviewTrig
 after insert or delete or update on review
 for each row
  execute procedure avgRating()
  ;
```