```
create table address
(addressID INT not null  auto_increment,
state char(20),
city char(20),
street char(40),
apartment char(20),
zipcode char(5),
primary key(addressID))


create table user
(userID INT not null auto_increment,
userType enum( bcustomer, hcustomer, salesperson, storemanager, regionmanager),
userTypeID b_customerID/h_customerID/salespersonID      how to insert trigger here?
username char(20),
password char(20),
primary key(userID),
foreign key(b_customerID, h_customeID, salespersonID)
references bcustomer, hcustomer, salesperson)


create table bcustomer
(b_customer INT not null auto_increment,
first_name char(10),
last_name char(10),
business category char(20),
company_ GAI REAL,
addressID INT,
primary key(b_customerID),
foreign key(addressID)
references addressID,
check (company_GAL >=0))


create table hcustomer
(h_customer INT not null auto_increment,
first_name char(10),
last_name char(10),
gender enum(male, female),
birth_date date,
income real,
marriage_status enum(single, married, divorced, widowed),
addressID addressID,
primary key(h_customerID),
```

```
foreign key(addressID)
references address,
check(income>=0))


create table salesperson
(salespersonID INT not null auto_increment
first_name char(10),
last_name char(10),
gender enum(male, female),
email char(20),
birth_date date,
job_title enum( store_manager, region_manager, direct_salesperson),
salary real,
addressID addressID,
primary key(salespersonID),
foreign key( addressID)
references address,
check(salary>=0))


create table customer
(customerID INT not null auto_increment,
customerType bcustomer, hcustomer,
typeID b_customerID/ h_customerID,
primary key(customerID),
foreign key (b_customerID, h_customerID)
references bcustomer, hcustomer)


create table store
storeID INT not null auto_increment,
addressID INT,
regionID INT,
salespersonID INT
primary key(storeID),
foreign key(salespersonID,regionID)
references salesperson, region,
check('store_manager'=
            (select job_title
             from salesperson
             where salesperson.salespersonID=store. salespersonID)))
```

```
create table category
(categoryID INT not null auto_increment,
name char(20),
primary key(categoryID))


create table brand
(brandID INT not null auto_increment,
name char(20),
primary key(brandID))


create table store_direct_sale
(salespersonID INT,
storeID INT,
primary key(salespersonID, storeID),
foreign key(storeID)
references store,
check('direct_salesperson'=
            (select job_title
             from store
             where store_direct_sale.storeID= store.storeID)))


create table transaction
(transactionID INT not null auto_increment,
date date,
customerID INT,
primary key(transactionID),
foreign key(customerID)
references customer)


create table product
(productID INT not null auto_increment,
name char(20),
cost real,
categoryID INT,
brandID INT,
primary key(productID),
foreign key(categoryID, brandID)
references category, brand,
check(cost>0))
```

```
create table storage
(productID INT,
storeID INT,
amount INT,
primary key(productID, storeID),
foreign key(productID, storeID)
reference product, store,
check(amount>=0) )


create table region
(regionID INT not null auto_increment,
name char(20)
salespersonID salespersonID(job_title = 'region_manager'),
primary key(regionID),
foreign key(salespersonID)
references store_direct_sale,
check('region_manager'=
            (select job_title
             from store_direct_sale
             where store_direct_sale.salespersonID= region.salespersonID)))


create table transaction_instance
(transactionID INT,
productionID INT,
ranking enum(1,2,3,4,5),
primary key(transactionID),
foreign key(transactionID, productID)
references transaction, product)


create table promotion
(promotionID INT not null auto_increment,
start_date date,
end_date date,
discount real,
productID productID,
primary key(promotionID),
foreign key(productID)
references product,
check(discount<100 and discount>0))
```