SIGNATURE ASSIGNMENT-

Milestone B

Individual work

Zewei Qian

ITC6000-21373

# **Introduction**:

"FIFA 19" is a football sports game produced and published by Electronic Arts. I retrieve the data from Kaggle. This dataset contains not only features of all the football players in FIFA, including all the player attributes, ability level, and images. But also, the attributes of players play style statistics like Dribbling, Aggression, GK Skills, etc.

The original dataset has 89 columns and my project is to do the football game analysis and make a database to contain all the football club information and player attributes. My database should allow users to do the advanced search base on a player name, club name, overall performance score, and best player at different positions. So database user can create their dream team.

# **Database System Analysis and Design：**

## User Profiles：

## (1) Management of **player** information, management of Nationality, Photo, Club, Age, Wage, Salary, etc. of current players

## (2) management of **club** information. management of each player information and position at this club and the **club** information such as club name and logo image

## (3) **Attributes** tables contains information about the player attributes and play styles and performance level

## (4) **POSITION** table contains all of the positions in the football game and all the players that plays this position in this game

## 

Business Rules：

1. Each player only have one attribute that indicates his playing style and overall performance
2. Each club have multiple players in a different position
3. Each player have a player ID
4. Each club have a club ID and club image
5. Database and do the player ranking base on the attribute score
6. Database and do the ranking of different position
7. Attributes and be updated by the player current performance
8. Each player should have a club name and position
9. Each attribute should have one overall performance score
10. Database allows ranking search based on overall performance score

分割线

规则补充参考：

同一个订单可以在不同的store下，所以订单表里面 order id 和store id 是主键 。

由于原始数据中

|  |
| --- |
| Benefit per order |
| Sales per customer |
| Order Profit Per Order |

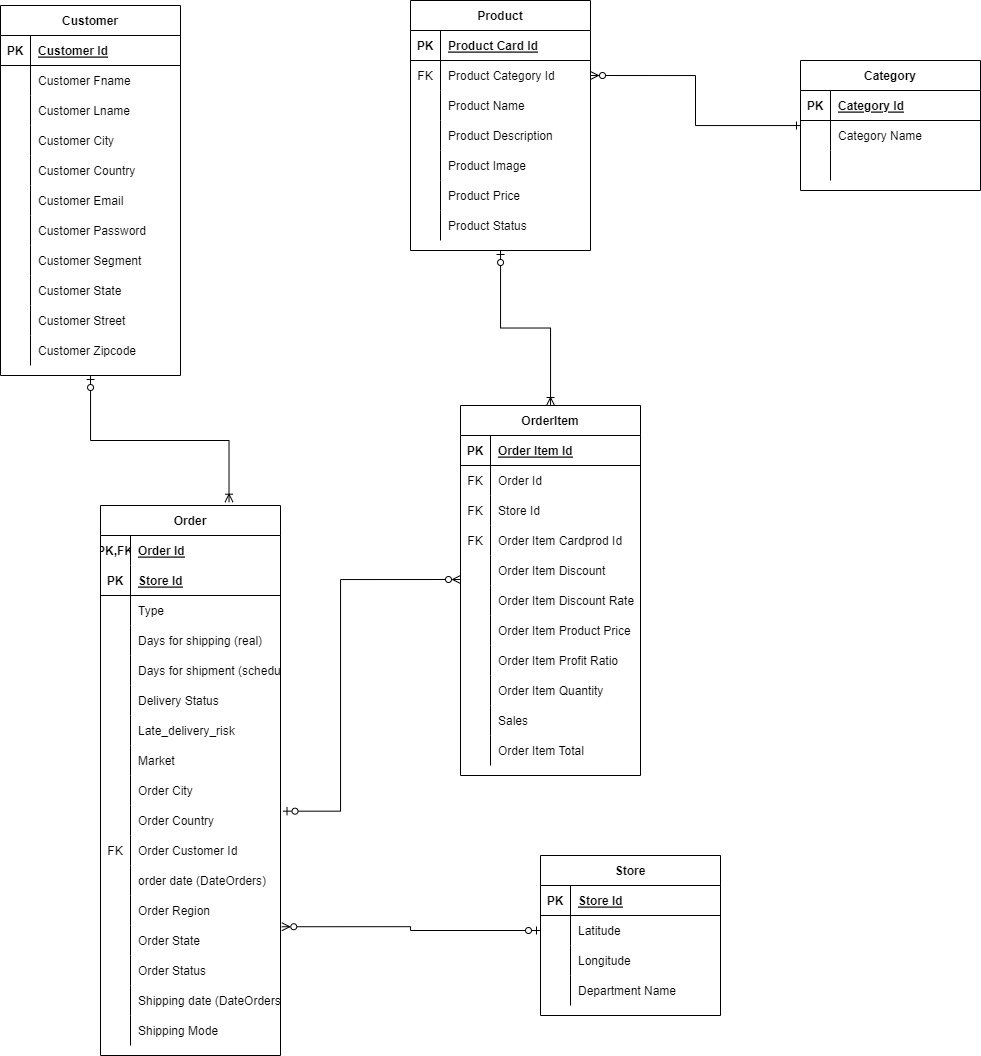
这三个属性时统计性质的，不是transaction的数据，因此不考虑使用在数据库实现中。

其他规则描述erd 的图就行

# Milestone -B: Database Design, Implementation, and Basic Queries

## Database Design：ERD

Based on the dataset, I design the ERD:



## Implement the finalized design in the SQLite

Firstly, I use DB browser to load the dataset into sqlite db.



Then, definition tables based on the ERD.

create table Customer (

"Customer Id" integer primary key,

"Customer Fname" text,

"Customer Lname" text,

"Customer City" text,

"Customer Country" text,

"Customer Email" text,

"Customer Password" text,

"Customer Segment" text,

"Customer State" text,

"Customer Street" text,

"Customer Zipcode" text

);

create table Category (

"Category Id" integer primary key,

"Category Name" text

);

create table Product(

"Product Card Id" integer primary key,

"Product Category Id" integer,

"Product Description" text,

"Product Image" text,

"Product Name" text ,

"Product Price" float,

"Product Status" integer,

foreign key ("Product Category Id") references Category("Category Id")

);

create table Store(

"Store Id" integer primary key autoincrement,

Latitude float,

Longitude float,

"Department Id" integer,

"Department Name" text

);

create table "Order"(

"Order Id" integer,

"Type" text,

"Days for shipping (real)" integer,

"Days for shipment (scheduled)" integer,

"Delivery Status" text,

"Late\_delivery\_risk" integer,

"Market" text,

"Order City" text,

"Order Country" text,

"Order Customer Id" integer,

"order date (DateOrders)" date,

"Order Region" text,

"Order State" text,

"Order Status" text,

"Shipping date (DateOrders)" date,

"Shipping Mode" text,

"Store Id" integer,

primary key ("Order Id","Store Id"),

foreign key ("Store Id") references Store("Store Id"),

foreign key ("Order Customer Id") references Customer("Customer Id")

);

create table OrderItem(

"Order Item Id" integer primary key,

"Order Id" integer,

"Store Id" integer,

"Order Item Cardprod Id" integer,

"Order Item Discount" float,

"Order Item Discount Rate" float,

"Order Item Product Price" float,

"Order Item Profit Ratio" float,

"Order Item Quantity" integer,

"Sales" float,

"Order Item Total" float,

foreign key ("Order Id","Store Id") references "Order"("Order Id","Store Id"),

foreign key ("Order Item Cardprod Id") references Product("Product Card Id")

);

## Populate the database

insert into Customer

select distinct

"Customer Id",

"Customer Fname",

"Customer Lname",

"Customer City",

"Customer Country",

"Customer Email",

"Customer Password",

"Customer Segment",

"Customer State",

"Customer Street",

"Customer Zipcode"

from DataCoSupplyChainDataset;

insert into Category

select distinct

"Category Id",

"Category Name"

from DataCoSupplyChainDataset;

insert into Product

select distinct

"Product Card Id",

"Product Category Id",

"Product Description",

"Product Image",

"Product Name",

"Product Price",

"Product Status"

from DataCoSupplyChainDataset;

insert into Store(Latitude,

Longitude,

"Department Id",

"Department Name")

select distinct

Latitude,

Longitude,

"Department Id",

"Department Name"

from DataCoSupplyChainDataset;

insert into "Order"

select distinct

"Order Id",

"Type",

"Days for shipping (real)",

"Days for shipment (scheduled)",

"Delivery Status",

"Late\_delivery\_risk",

"Market",

"Order City",

"Order Country",

"Order Customer Id",

"order date (DateOrders)",

"Order Region",

"Order State",

"Order Status",

"Shipping date (DateOrders)",

"Shipping Mode",

Store."Store Id"

from DataCoSupplyChainDataset d

join Store on d.Latitude=Store.Latitude and d.Longitude=Store.Longitude and d."Department Id"=Store."Department Id";

insert into OrderItem

select distinct

"Order Item Id",

"Order Id",

Store."Store Id",

"Order Item Cardprod Id",

"Order Item Discount",

"Order Item Discount Rate",

"Order Item Product Price",

"Order Item Profit Ratio",

"Order Item Quantity",

"Sales",

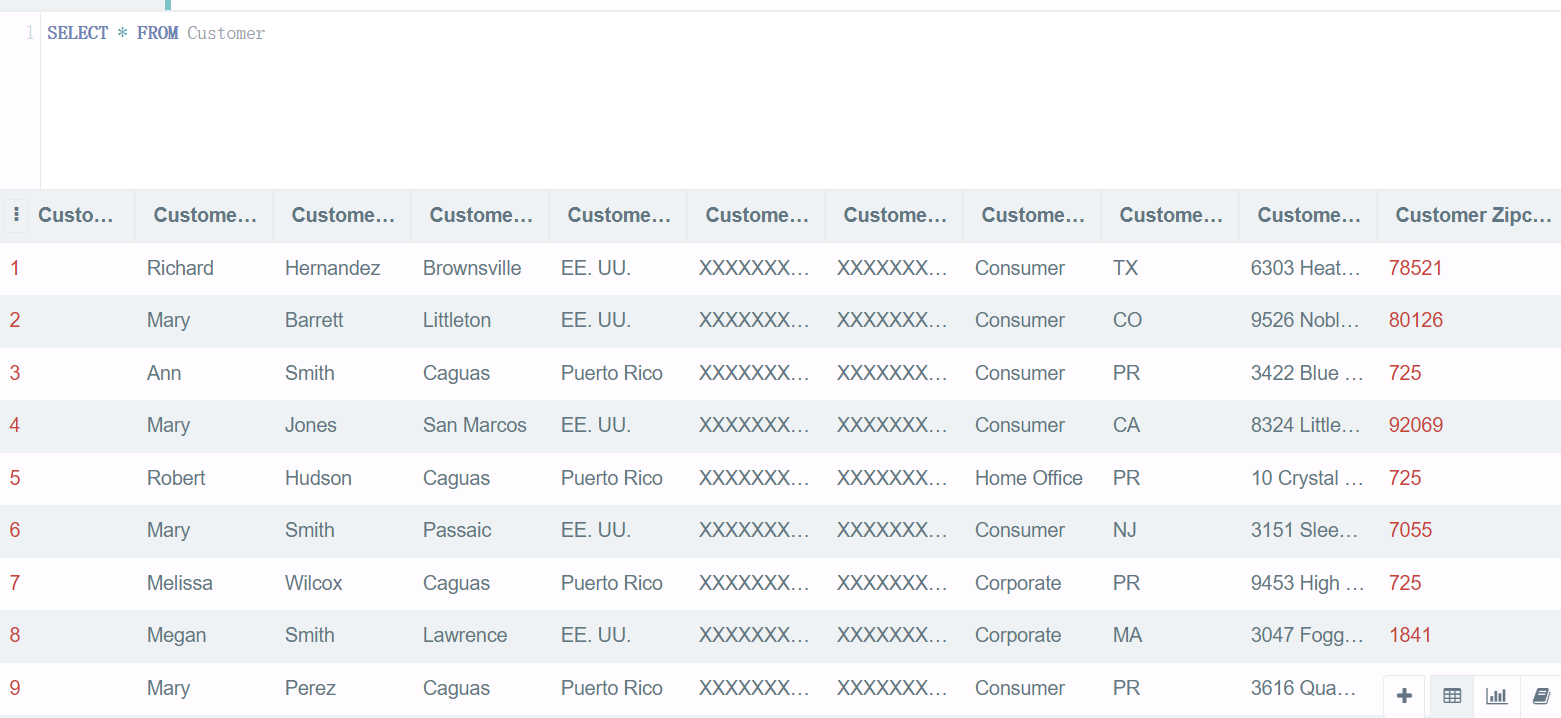
"Order Item Total"

from DataCoSupplyChainDataset d

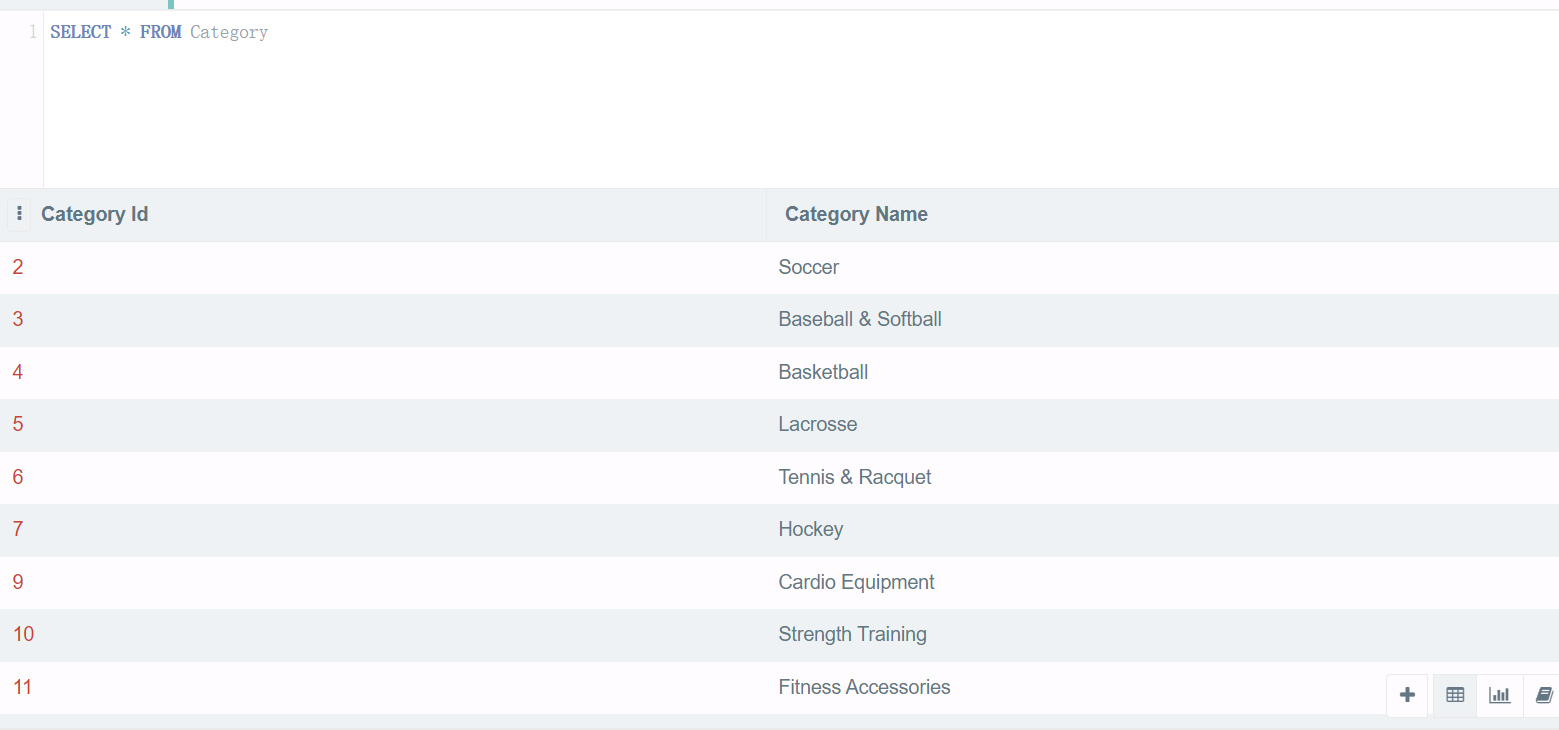
join Store on d.Latitude=Store.Latitude and d.Longitude=Store.Longitude and d."Department Id"=Store."Department Id";

## Basic Queries

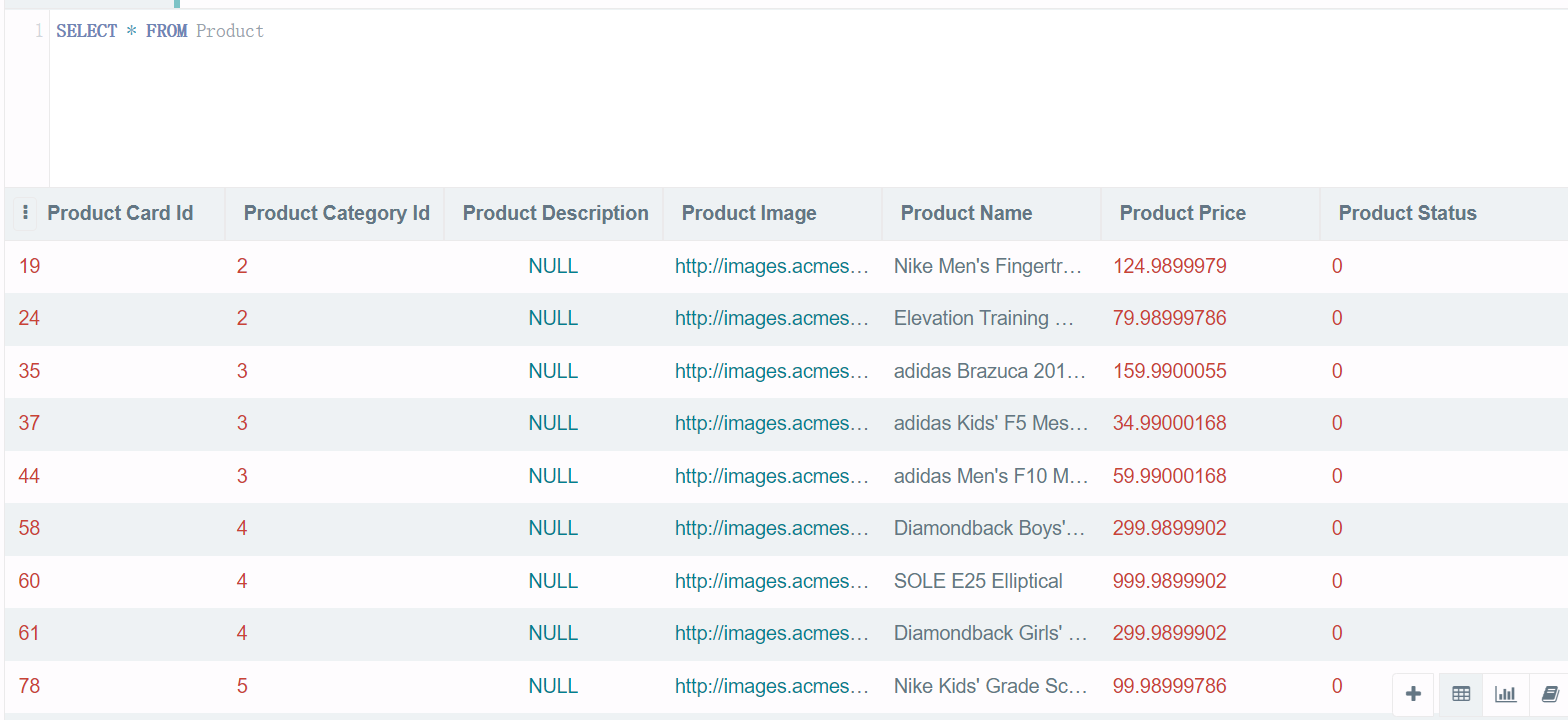
select \* from Customer;



select \* from Category;



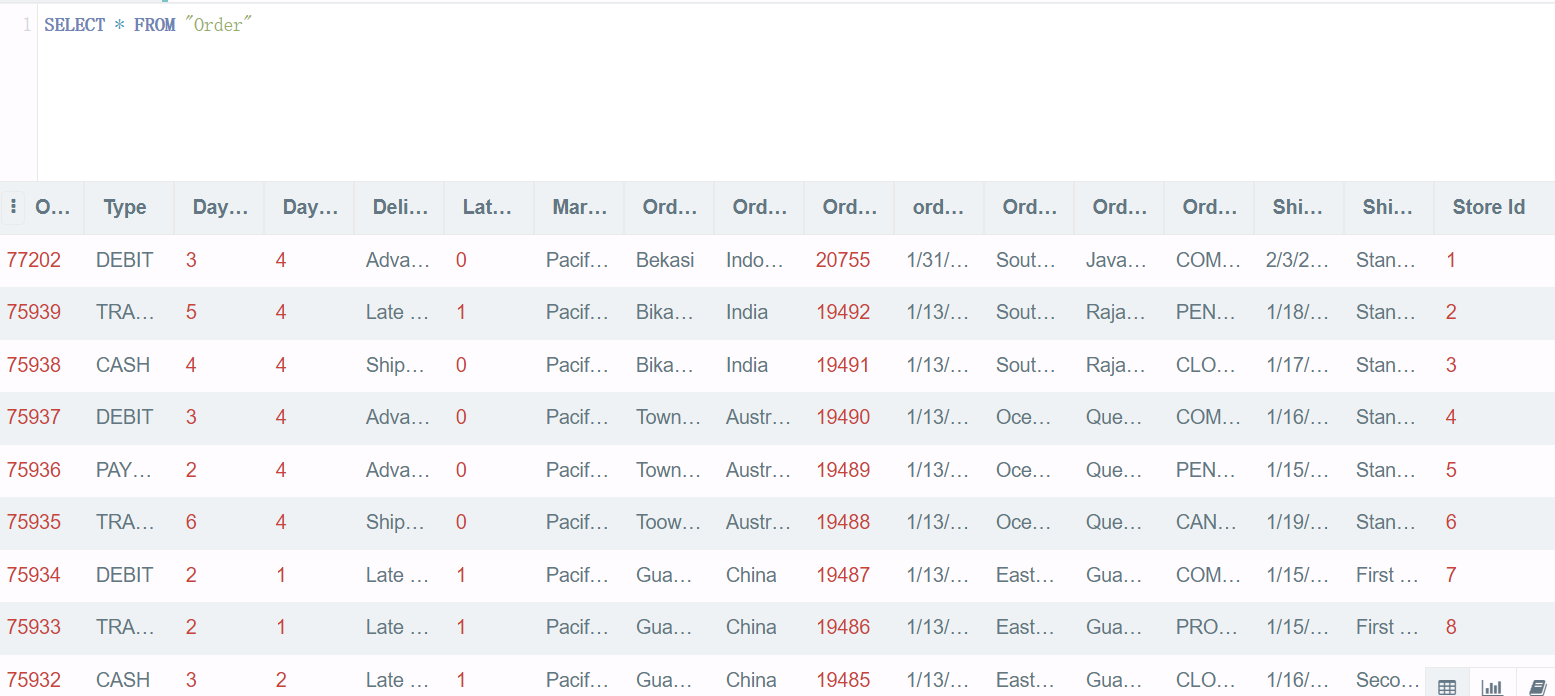
select \* from Product;



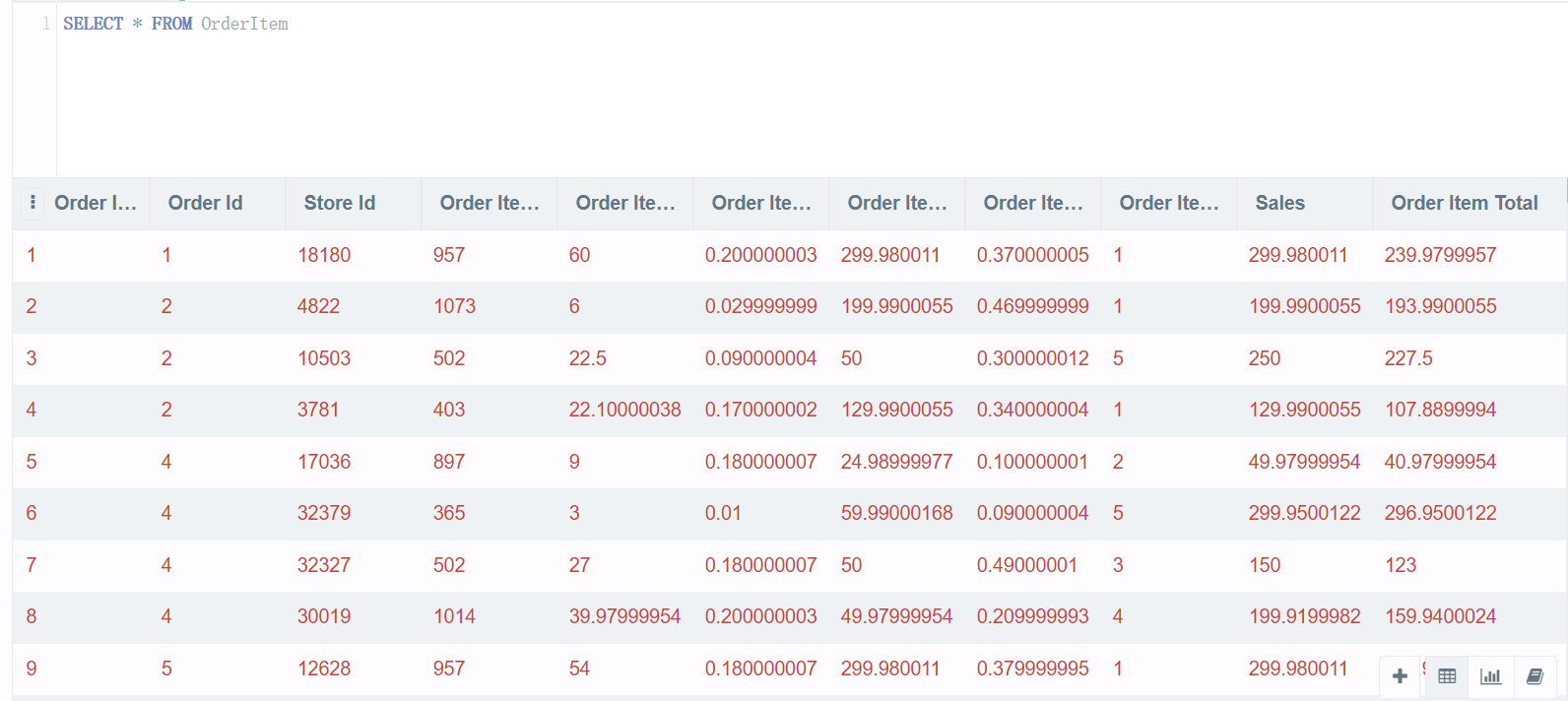
select \* from Store;



select \* from "Order";



select \* from OrderItem;



-- 查询每个category 有分别有多少Product 被售卖（显示Customers更关注哪种产品）, display category 名称，产品数量，一个产品id 仅被计算一次， 并将结果由大到小排列。

select c."Category Name", count(distinct p."Product Card Id") as ProductNum

from Category c

join Product p on c."Category Id"=p."Product Category Id"

group by c."Category Name"

order by ProductNum desc;



-- 查询不同地区的客户购买产品总金额，按照产品金额降序排列

select c."Customer State", sum(oi."Order Item Total") as TotalAmount

from Customer c

join "Order" o on c."Customer Id"=o."Order Customer Id"

join OrderItem oi on o."Order Id"=oi."Order Id" and o."Store Id"=oi."Store Id"

group by c."Customer State"

order by TotalAmount desc;



-- 查询Order is delivered to 哪些Market的数量？即query 所有Market的Order的数量

select o.Market, count(o."Order Id") as OrderCount

from "Order" o

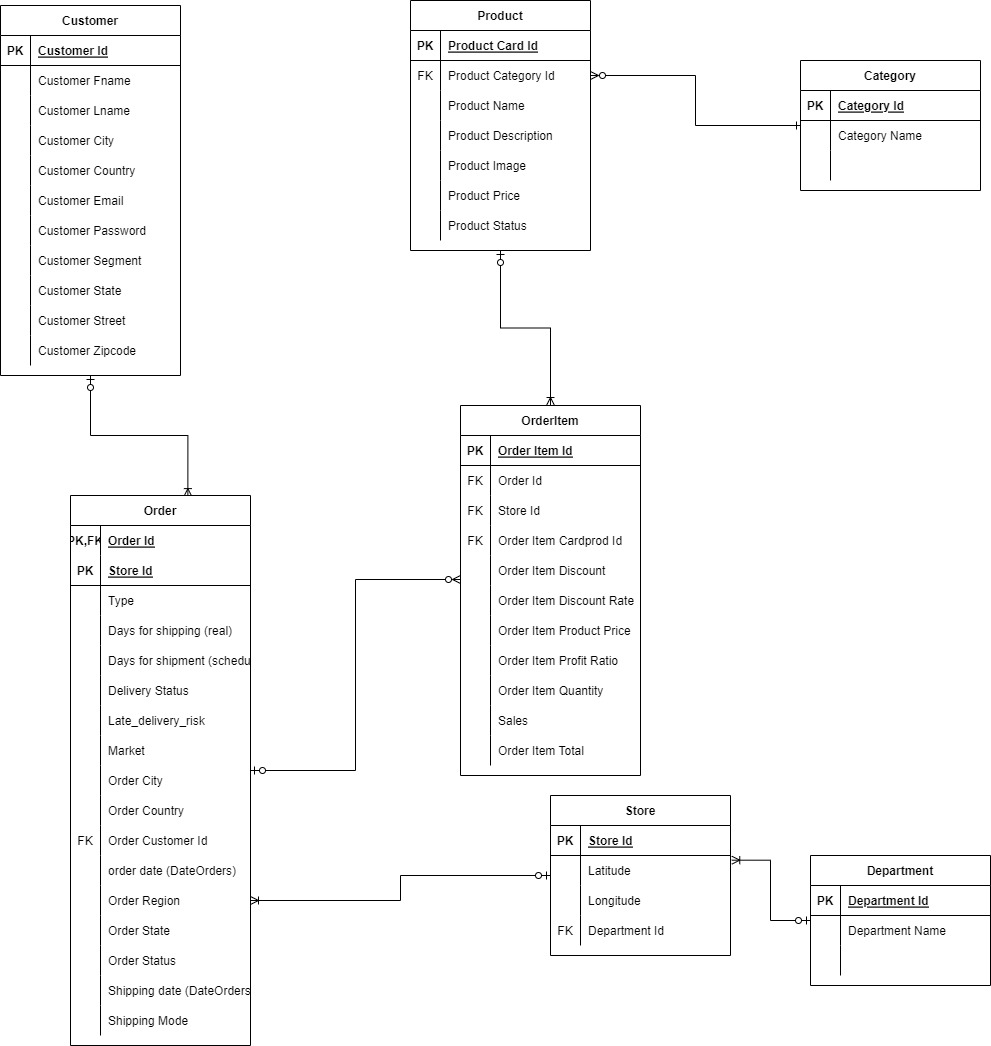
group by o.Market

order by OrderCount desc;



# Milestone -C: Database Queries (Intermediate SQL queries)

## Revise the database design (ERD)



## Revise and execute the SQL script for data population to insert full data previously collected or at least ensure that sufficiently large number of observations are inserted to implement advance queries

跟B相比增加了 check约束，另外，基于修改的ERD，新建了Department表

-- tables definition

create table Customer (

"Customer Id" integer primary key,

"Customer Fname" text,

"Customer Lname" text,

"Customer City" text,

"Customer Country" text,

"Customer Email" text,

"Customer Password" text,

"Customer Segment" text check ("Customer Segment" in ('Consumer','Home Office','Corporate')),

"Customer State" text,

"Customer Street" text,

"Customer Zipcode" text

);

create table Category (

"Category Id" integer primary key,

"Category Name" text

);

create table Product(

"Product Card Id" integer primary key,

"Product Category Id" integer,

"Product Description" text,

"Product Image" text,

"Product Name" text ,

"Product Price" float,

"Product Status" integer check("Product Status" in (0,1)),

foreign key ("Product Category Id") references Category("Category Id")

);

create table Department(

"Department Id" integer primary key,

"Department Name" text

);

create table Store(

"Store Id" integer primary key autoincrement,

Latitude float,

Longitude float,

"Department Id" integer,

foreign key ("Department Id") references Department("Department Id")

);

create table "Order"(

"Order Id" integer,

"Type" text,

"Days for shipping (real)" integer,

"Days for shipment (scheduled)" integer,

"Delivery Status" text check ("Delivery Status" in ('Advance shipping','Late delivery','Shipping on time','Shipping canceled')),

"Late\_delivery\_risk" integer check ("Late\_delivery\_risk" in (0,1)),

"Market" text check ("Market" in ('Pacific Asia','USCA','Africa','Europe','LATAM')),

"Order City" text,

"Order Country" text,

"Order Customer Id" integer,

"order date (DateOrders)" date,

"Order Region" text,

"Order State" text,

"Order Status" text check ("Order Status" in ('COMPLETE','PENDING','CLOSED','PENDING\_PAYMENT',

'CANCELED','PROCESSING','SUSPECTED\_FRAUD','ON\_HOLD','PAYMENT\_REVIEW')),

"Shipping date (DateOrders)" date,

"Shipping Mode" text check ("Shipping Mode" in ('Standard Class','First Class','Second Class','Same Day')),

"Store Id" integer,

primary key ("Order Id","Store Id"),

foreign key ("Store Id") references Store("Store Id"),

foreign key ("Order Customer Id") references Customer("Customer Id")

);

create table OrderItem(

"Order Item Id" integer primary key,

"Order Id" integer,

"Store Id" integer,

"Order Item Cardprod Id" integer,

"Order Item Discount" float,

"Order Item Discount Rate" float,

"Order Item Product Price" float,

"Order Item Profit Ratio" float,

"Order Item Quantity" integer,

"Sales" float,

"Order Item Total" float,

foreign key ("Order Id","Store Id") references "Order"("Order Id","Store Id"),

foreign key ("Order Item Cardprod Id") references Product("Product Card Id")

);

-- populate data into tables

insert into Customer

select distinct

"Customer Id",

"Customer Fname",

"Customer Lname",

"Customer City",

"Customer Country",

"Customer Email",

"Customer Password",

"Customer Segment",

"Customer State",

"Customer Street",

"Customer Zipcode"

from DataCoSupplyChainDataset;

insert into Category

select distinct

"Category Id",

"Category Name"

from DataCoSupplyChainDataset;

insert into Product

select distinct

"Product Card Id",

"Product Category Id",

"Product Description",

"Product Image",

"Product Name",

"Product Price",

"Product Status"

from DataCoSupplyChainDataset;

insert into Department

select distinct "Department Id", "Department Name"

from DataCoSupplyChainDataset;

insert into Store(Latitude,

Longitude,

"Department Id")

select distinct

Latitude,

Longitude,

"Department Id"

from DataCoSupplyChainDataset;

insert into "Order"

select distinct

"Order Id",

"Type",

"Days for shipping (real)",

"Days for shipment (scheduled)",

"Delivery Status",

"Late\_delivery\_risk",

"Market",

"Order City",

"Order Country",

"Order Customer Id",

"order date (DateOrders)",

"Order Region",

"Order State",

"Order Status",

"Shipping date (DateOrders)",

"Shipping Mode",

Store."Store Id"

from DataCoSupplyChainDataset d

join Store on d.Latitude=Store.Latitude and d.Longitude=Store.Longitude and d."Department Id"=Store."Department Id";

insert into OrderItem

select distinct

"Order Item Id",

"Order Id",

Store."Store Id",

"Order Item Cardprod Id",

"Order Item Discount",

"Order Item Discount Rate",

"Order Item Product Price",

"Order Item Profit Ratio",

"Order Item Quantity",

"Sales",

"Order Item Total"

from DataCoSupplyChainDataset d

join Store on d.Latitude=Store.Latitude and d.Longitude=Store.Longitude and d."Department Id"=Store."Department Id";

select \* from Customer;

select \* from Category;

select \* from Product;

select \* from Department;

select \* from Store;

select \* from "Order";

select \* from OrderItem;

## Develop SQL query solutions for the use-cases you identified for this phase.

a. Write the necessary SQL script using the R/RStudio or DB Browser.

b. These queries may require joining multiple tables, transformation of data columns, conditional queries, use of regular expression, and union of multiple queries.

c. Evaluate the queries to ensure it produces desired output and generate the HTML output.

d. Write your inferences of the query results.

e. The DB Browser users should save the SQL script as a separate file for submission and paste screen shots of result window to report.

-- 创建视图 -- query 所有普通客户的订单信息，包含客户first name, last name, 客户所在国家、城市，下订单日期，订单delivery 日期，订单购买的product 数量， 订单总金额

create view customer\_orders\_view

as

select c."Customer Fname", c."Customer Lname", c."Customer Country", c."Customer City",

o."order date (DateOrders)", o."Shipping date (DateOrders)",

sum(oi."Order Item Quantity") as TotalItemQuantity,

sum(oi."Order Item Total") as TotalOrderAmount

from Customer c

join "Order" o on c."Customer Id"=o."Order Customer Id"

join OrderItem oi on o."Order Id"=oi."Order Id" and o."Store Id"=oi."Store Id"

where c."Customer Segment"='Consumer'

group by

c."Customer Fname", c."Customer Lname", c."Customer Country", c."Customer City",

o."order date (DateOrders)", o."Shipping date (DateOrders)";





-- 查询购买客户是'Home Office'或者'Corporate' 的销往亚太地区的产品名称,将产品名称按照字母顺序排列

select distinct p."Product Name" as ProductName

from Customer c join "Order" o on o."Order Customer Id"=c."Customer Id"

join OrderItem oi on o."Order Id"=oi."Order Id" and o."Store Id"=oi."Store Id"

join Product p on p."Product Card Id"=oi."Order Item Cardprod Id"

where c."Customer Segment"='Home Office' and o.Market like '%Pacific Asia%'

union

select distinct p."Product Name"as ProductName

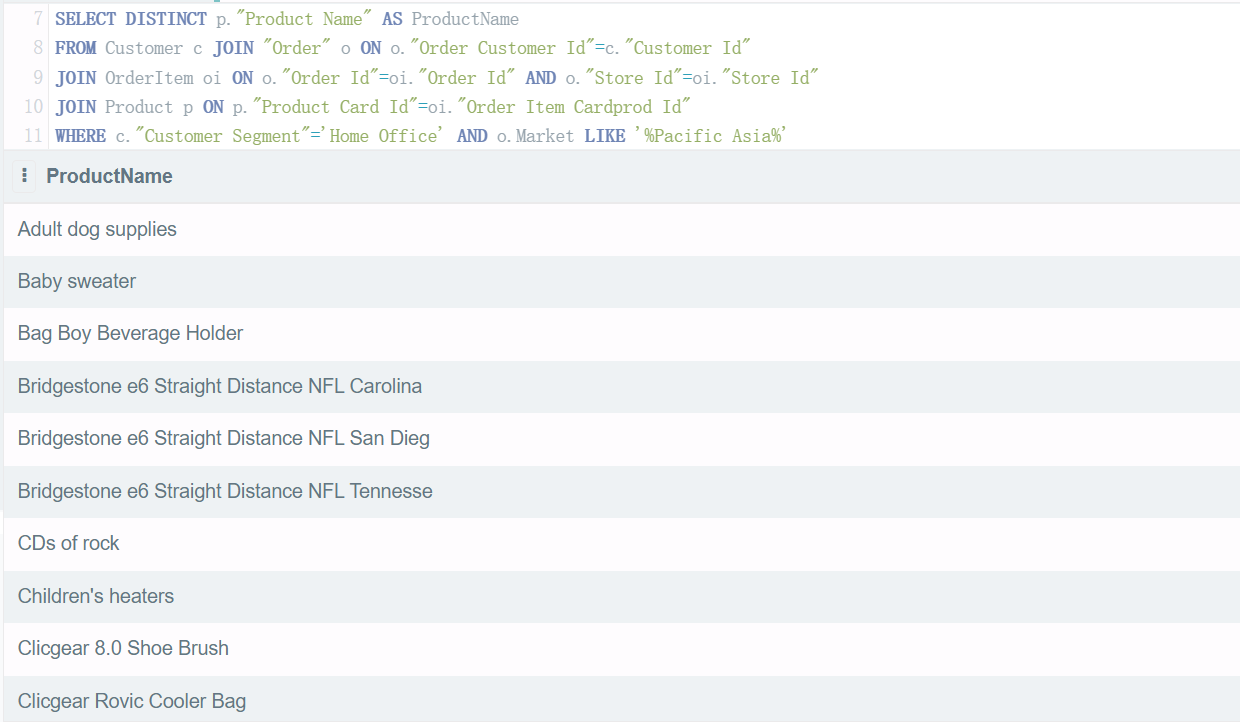
from Customer c join "Order" o on o."Order Customer Id"=c."Customer Id"

join OrderItem oi on o."Order Id"=oi."Order Id" and o."Store Id"=oi."Store Id"

join Product p on p."Product Card Id"=oi."Order Item Cardprod Id"

where c."Customer Segment"='Corporate' and o.Market like '%Pacific Asia%'

order by ProductName;



# Individu al Signature Project Report and Presentation

## Revise the SQL queries

参见B 、C的sql queries

做ppt的时候把前面的描述文字重写，把图贴上去

这块到时候也把前面的内容再贴进来

## Develop SQL query solutions for the newly identified use-cases

新增的sql queries

-- with clause: 查询

-- which department 销往不同国家的销售总额最多？

with t as

(

select d."Department Name", o."Order Country", sum(oi."Order Item Total") as TotalAmount

from Department d join Store s on s."Department Id"=d."Department Id"

join "Order" o

on o."Store Id"=s."Store Id"

join OrderItem oi on o."Order Id"=oi."Order Id" and o."Store Id"=oi."Store Id"

group by d."Department Name", o."Order Country"

order by TotalAmount desc

)

select \* from t limit 1;

