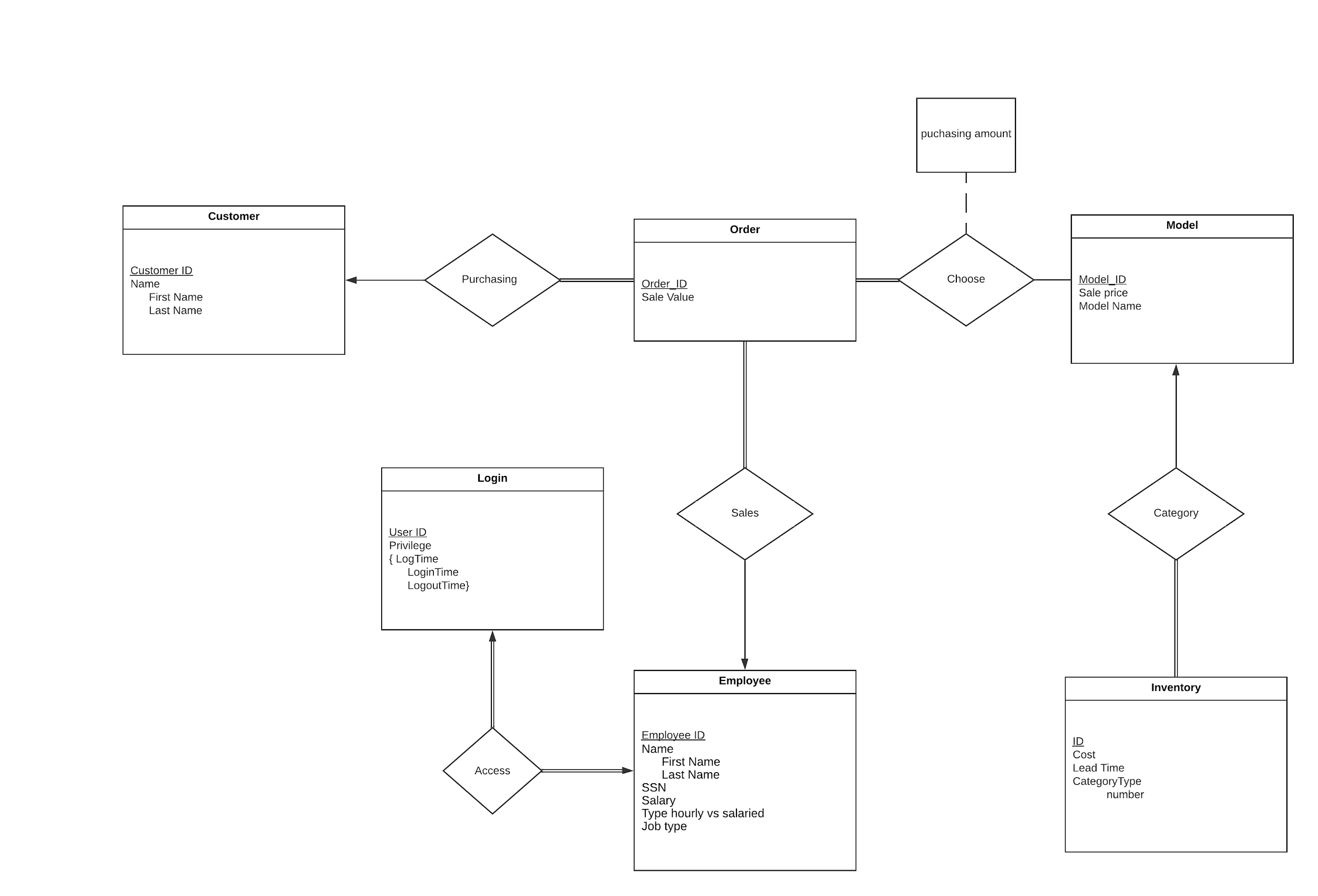
ER-Model Diagram

Team Number: 16

Team Members: Ning Zhang, Yinting Hui, Zhanglijun Shao



Assumptions:

1. Assuming a production can only produce one model therefore CategoryType is not multivalued
2. CategoryType is a composite attribute including number and Model\_ID as a foreign key reference to Model.
3. LogTime is a composite attribute including LoginTime and LogoutTime.
4. The attributes (LoginTime and LogoutTime) in ‘LogTime’ should be multivalued.
5. For entities (Login and Employee) they are one to one but at the same time they should be full participation to each other, where every employee must have one user ID. When the user ID is then removed, it must also remove the related Employee from the table.
6. In the relation ‘Choose’, there should be an attribute ‘purchasing amount’ to indicate how many items there will be in one order.

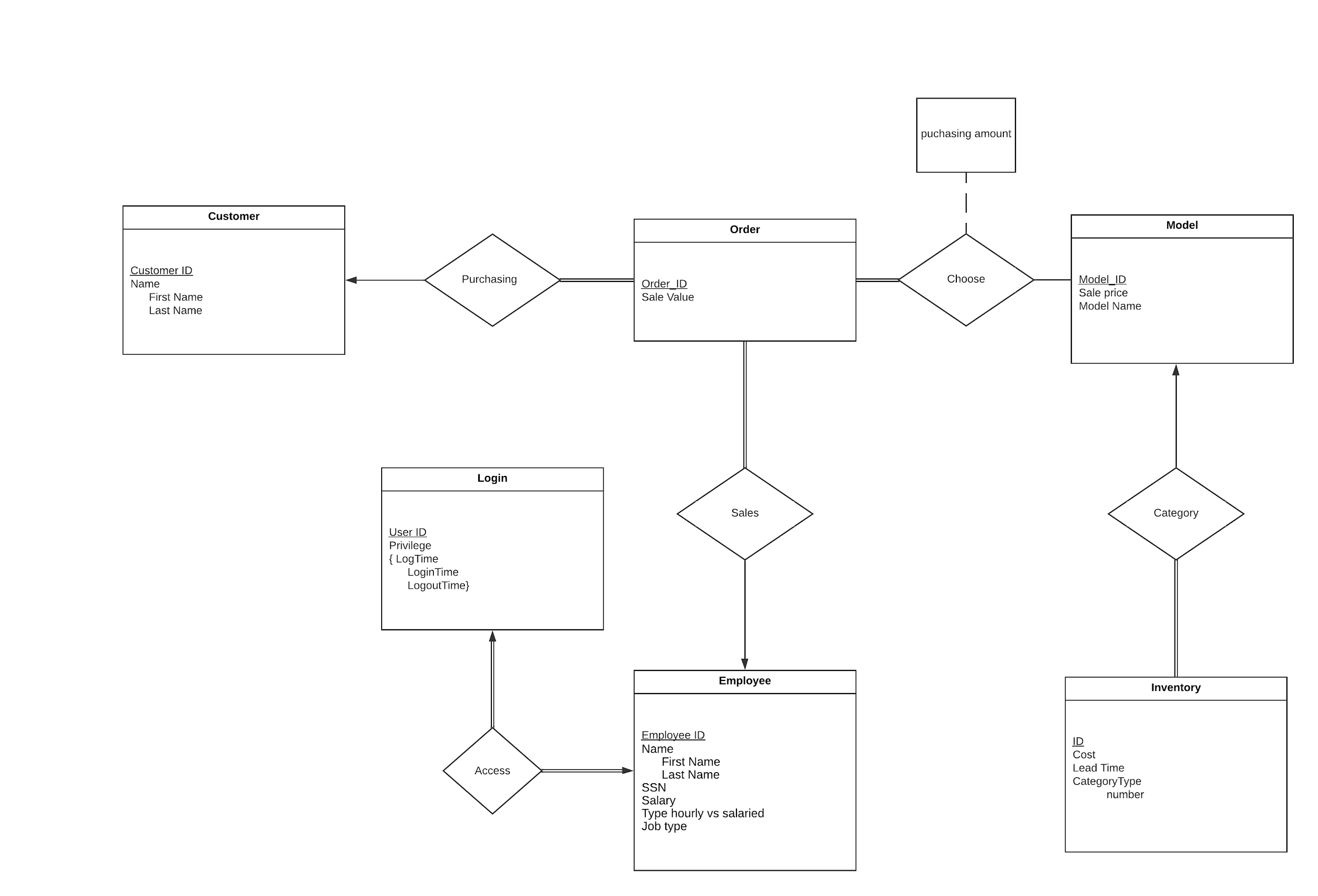
Contributions: all members of the team were involved in the discussion and the creation of this ER model diagram.

Project Part2

Team Number: 16

Team Members: Ning Zhang, Yinting Hui, Zhanglijun Shao

1. ER-diagram



1. Schemas

Inventory(ID, cost, Lead\_time, number, model\_ID ) model\_ID as foreign key from Model (rule1, rule4)

Model(model\_ID, model\_name, sales\_price)(rule1, rule4)

EmployeeLogin(employee\_ID, user\_ID, privilege, first\_name, last\_name, ssn, salary, salary\_type, job\_type)(rule1, rule 3, rule 4)

Customer(customer\_ID, first\_name, last\_name)(rule1, rule4)

Orders(order\_ID, sale\_value, employee\_ID, customer\_ID) employee\_ID as foreign key from EmpolyeeLogin. customer\_ID as foreign key from Customer (rule1, rule4)

Choose(model\_ID, order\_ID, purchasing\_amount) model\_ID as foreign key to Model, Order\_ID as foreign key to Order. (rule5)

Session( employee\_ID, login\_time, logout\_time), employee\_ID as foreign key from EomplyeeLogin (rule7)

1. Script with Postgresql

**create** **table** Model(model\_ID **char**(10) **not** **null**, model\_name **varchar**(20), sales\_price **numeric**(20, 2), **primary** **key**(model\_ID));

**create** **table** Customer(customer\_ID **char**(10) **not** **null**, first\_name **varchar**(20), last\_name **varchar**(20), **primary** **key**(customer\_ID));

**create** **table** EmployeeLogin(employee\_ID **char**(10) **not** **null**, user\_ID **char**(10) **not** **null**, privilege **varchar**(20), first\_name **varchar**(20), last\_name **varchar**(20), ssn **char**(10), salary **numeric**(20, 2) **check**(salary>0), salary\_type **varchar**(20), job\_type **varchar**(20) **check**(job\_type **in** ('HR','Sale','Engineer')), **primary** **key**(employee\_ID));

**create** **table** Inventory(ID **char**(10) **not** **null**, **cost** **numeric**(20, 2) **check**(**cost**>0), lead\_time **int** **check** (lead\_time>0), model\_ID **char**(10), **number** **int** **check**(**number** > -1),**primary** **key**(ID), **foreign** **key**(model\_ID) **references** Model);

**create** **table** Orders(order\_ID **char**(10) **not** **null**, sale\_value **numeric** (20, 2) **check**(sale\_value>0), employee\_ID **char**(10), customer\_ID **char**(10), **year** **int** **check**(**year** >0), **primary** **key**(order\_ID), **foreign** **key** (employee\_ID) **references** EmployeeLogin, **foreign** **key** (customer\_ID) **references** Customer);

**create** **table** Choose(model\_ID **char**(10) **not** **null**, order\_ID **char**(10) **not** **null**, purchasing\_amount **int** **check**(purchasing\_amount>0), **primary** **key**(model\_ID, order\_ID), **foreign** **key** (model\_ID) **references** Model, **foreign** **key** (order\_ID) **references** Orders);

**create** **table** Sessions(employee\_ID **char**(10) **not** **null**, login\_time **timestamp** **not** **null**, logout\_time **timestamp** **not** **null**, **primary** **key**(employee\_ID, login\_time, logout\_time), **foreign** **key** (employee\_ID) **references** EmployeeLogin);

**create** **view** TotalSaleV **as** (**select** **sum**(sale\_value) **from** Orders **group** **by** **year**);

**create** **view** orderInventory **as** (**select** choose.model\_ID, order\_id, purchasing\_amount, **number** **from** (choose **natural** **join** inventory));

**create** **view** modelReport **as** (**select** model\_ID, **year**,**sum**(purchasing\_amount) **from** Choose **natural** **join** Orders **group** **by** model\_ID, **year**);

**create** **view** InventoryExpense **as** (**select** **year**, **sum**(**cost** \* purchasing\_amount) **as** Total\_Cost **from** (inventory **natural** **join** Choose **natural** **join** Orders) **group** **by** **year**);

**create** **view** EmployeeExpense **as** (**select** **sum**(salary) **as** Total\_Salary **from** employeelogin);

**create** **view** EarnedSale **as** (**select** **year**, **sum**(sales\_price \* purchasing\_amount) **as** Total\_Earn **from** (orders **natural** **join** model **natural** **join** choose) **group** **by** **year**);

**create** **view** totalExpense **as** (**select** **year**, Total\_Salary + Total\_Cost **as** Total\_Expense **from** (InventoryExpense **cross** **join** EmployeeExpense));

**create** **view** TotalRevenue **as** (**select** **year**, Total\_Earn - Total\_Expense **as** Revenue **from** (EarnedSale **natural** **join** totalExpense));

**create** **view** BuisnessReport **as** (**select** **year**, Total\_Cost, Total\_Salary, Total\_Expense **from** ((totalExpense **natural** **join** InventoryExpense)**cross** **join** EmployeeExpense));

**create** **role** admin **INHERIT** LOGIN;

**create** **role** Sale **INHERIT** LOGIN;

**create** **role** HR **INHERIT** LOGIN;

**create** **role** Engineer **INHERIT** LOGIN;

**create** **index** ModelID\_index **on** Model(model\_ID);

**create** **index** CustomerID\_index **on** Customer(customer\_ID);

**create** **index** EmployeeLoginID\_index **on** EmployeeLogin(employee\_ID);

**create** **index** InventoryID\_index **on** Inventory(ID);

**create** **index** OrdersID\_index **on** Orders(order\_ID);

**create** **index** ChooseID\_index **on** Choose(model\_ID, order\_ID);

**create** **index** SessionsID\_index **on** Sessions(employee\_ID, login\_time, logout\_time);

**create** **view** jobtype **as** (**select** user\_ID, privilege **from** EmployeeLogin);

**grant** **create** **on** **database** postgres **to** **admin**;

**grant** **all** **privileges** **on** Model **to** **admin** **with** **grant** **option**;

**grant** **all** **privileges** **on** Customer **to** **admin** **with** **grant** **option**;

**grant** **all** **privileges** **on** employeelogin **to** **admin** **with** **grant** **option**;

**grant** **all** **privileges** **on** Inventory **to** **admin** **with** **grant** **option**;

**grant** **all** **privileges** **on** Orders **to** **admin** **with** **grant** **option**;

**grant** **all** **privileges** **on** Choose **to** **admin** **with** **grant** **option**;

**grant** **all** **privileges** **on** Sessions **to** **admin** **with** **grant** **option**;

**grant** **all** **privileges** **on** jobtype **to** **admin** **with** **grant** **option**;

**grant** **all** **privileges** **on** TotalSaleV **to** **admin** **with** **grant** **option**;

**grant** **all** **privileges** **on** orderInventory **to** **admin** **with** **grant** **option**;

**grant** **all** **privileges** **on** modelReport **to** **admin** **with** **grant** **option**;

**grant** **all** **privileges** **on** InventoryExpense **to** **admin** **with** **grant** **option**;

**grant** **all** **privileges** **on** EmployeeExpense **to** **admin** **with** **grant** **option**;

**grant** **all** **privileges** **on** EarnedSale **to** **admin** **with** **grant** **option**;

**grant** **all** **privileges** **on** totalExpense **to** **admin** **with** **grant** **option**;

**grant** **all** **privileges** **on** TotalRevenue **to** **admin** **with** **grant** **option**;

**grant** **all** **privileges** **on** BuisnessReport **to** **admin** **with** **grant** **option**;

**grant** **select** **on** jobtype **to** sale;

**GRANT** **UPDATE** **ON** customer **TO** sale;

**GRANT** **SELECT** **ON** customer **TO** sale;

**GRANT** **INSERT** **ON** Orders **TO** sale;

**GRANT** **SELECT** **ON** Orders **TO** sale;

**GRANT** **select** **ON** Inventory **TO** sale;

**GRANT** **update** **ON** Inventory **TO** sale;

**GRANT** **select** **ON** Choose **TO** sale;

**GRANT** **insert** **ON** Choose **TO** sale;

**grant** **select** **on** jobtype **to** Engineer;

**grant** **update** **on** Model **to** Engineer;

**grant** **select** **on** Model **to** Engineer;

**grant** **update** **on** Inventory **to** Engineer;

**grant** **select** **on** Inventory **to** Engineer;

**create** **view** enginView **as**(**select** first\_name, last\_name, job\_type **from** Employeelogin);

**grant** **select** **on** enginView **to** Engineer;

**grant** **select** **on** jobtype **to** HR;

**grant** **update** **on** Employeelogin **to** HR;

**grant** **select** **on** Employeelogin **to** HR;

**create** **view** emploSales **as** (**select** Employeelogin.employee\_id, first\_name, last\_name, ssn, salary, salary\_type, job\_type, **sum**(sale\_value) **from** Employeelogin **natural** **left** **outer** **join** Orders **group** **by** Employeelogin.employee\_id);

**grant** **select** **on** emploSales **to** HR;

**insert** **into** employeelogin **values**('e000000001', 'u000000001','Admin', 'Ning', 'Zhang','123457890','10000000','salaried', 'Engineer');

**create** **user** u000000001 **with** **password** '1' **inherit** login;

**grant** **admin** **to** u000000001;

**insert** **into** employeelogin **values**('e000000002', 'u000000002','HR', 'Bob', 'Green','123457891','10000000','salaried', 'HR');

**create** **user** u000000002 **with** **password** '1' **inherit** login;

**grant** HR **to** u000000002;

**insert** **into** employeelogin **values**('e000000003', 'u000000003','Sale', 'Ann', 'Allcock','123457892','10000000','salaried', 'Sale');

**create** **user** u000000003 **with** **password** '1' **inherit** login;

**grant** Sale **to** u000000003;

**insert** **into** employeelogin **values**('e000000004', 'u000000004','Engineer', 'Bill', 'Gates','123457893','100000000','salaried', 'Engineer');

**create** **user** u000000004 **with** **password** '1' **inherit** login;

**grant** Engineer **to** u000000004;

**insert** **into** employeelogin **values**('e000000005', 'u000000005','Sale', 'Newton', 'Willson','123457894','10000000','salaried', 'Sale');

**create** **user** u000000005 **with** **password** '1'**inherit** login;

**grant** Sale **to** u000000005;

**insert** **into** model **values**('m000000001','Model\_S', 100000);

**insert** **into** model **values**('m000000002','Model\_E', 50000);

**insert** **into** model **values**('m000000003','Model\_X', 120000);

**insert** **into** model **values**('m000000004','Model\_Y', 150000);

**insert** **into** customer **values**('c000000001','Dan', 'Ford');

**insert** **into** customer **values**('c000000002','Hanna', 'White');

**insert** **into** inventory **values**('i000000001',20000,30,'m000000001',30);

**insert** **into** inventory **values**('i000000002',10000,30,'m000000002',30);

**insert** **into** inventory **values**('i000000003',60000,30,'m000000003',30);

**insert** **into** inventory **values**('i000000004',100000,30,'m000000004',30);

**insert** **into** orders **values**('o000000001',500000,'e000000003','c000000001',2018);

**insert** **into** orders **values**('o000000002',3200000,'e000000005','c000000002',2019);

**insert** **into** choose **values**('m000000001','o000000001',5);

**insert** **into** choose **values**('m000000002','o000000002',10);

**insert** **into** choose **values**('m000000003','o000000002',10);

**insert** **into** choose **values**('m000000004','o000000002',10);

**insert** **into** sessions **values**('e000000001', '2020-04-29 19:00:00', '2020-04-29 23:00:00');

**insert** **into** sessions **values**('e000000002', '2020-05-29 19:00:00', '2020-05-29 23:00:00');

Contributions: all members of the team were involved in the discussion and the creation of the project part 2.

Project Part3

Team Number: 16

Team Members: Ning Zhang, Yinting Hui, Zhanglijun Shao

1. Login UI:

UserID: admin

Password: 1

Try to connect the database with userID and password from login UI.

If connecting successfully, go to User Operation UI; Otherwise, return UI “wrong username or password”

Please input your userID and password!

UserID: “369”

Password: “1”

Connect the database with username and password.

Return true and checking the EmployeeLogin table to get the job type. Goto to the User Operation UI which belongs to a specific job type.

1. User Operation UI

There are four user UI : admin, HR, Sale, Engineer.

1. Admin UI:

Please select the operation number:

1. Create a new employee (create a new userID and password and insert the userID and the new employee tuple to EmployeeLogin table )

UI: Please input the new employee’s userID: “qeae”

Please input the password:”369”

Please input the insert SQL about the information of this new employee.(read the employee information by userID and put them )

1. Set up tables: Please input your SQL.
2. Grant access, Please input your SQL.
3. Access and create the business reporting(access and create a yearly view for the total order number, order value), Please input your SQL.
4. Business Analytic

Total revenue from sale, associated employee and customer

Customer model bought and quantity to make prediction and understand trending

For each order, the associated parts and available inventor

Expense report, employee showing salary, bonus expense and part cost

1. Logout; goto login UI
2. HR UI:

System.***out***.println("HR Operations:");

System.***out***.println("1.View Employee");

System.***out***.println("2.Search in Employee"); //firstname, job type, loginID

System.***out***.println("3.Update Employee"); //salary, privilege, job type (search by login id)

System.***out***.println("4.View employee and associated sales number");

System.***out***.println("5.Please input your SQL operation:");

System.***out***.println("6.Logout");

System.***out***.println("Please select a HR operations(for exmaple: 1):");

Search in Employee UI:

System.***out***.println("Search Operations:");

System.***out***.println("1.Search by EmployeeID");

System.***out***.println("2.Search by jobtype");

System.***out***.println("3.Search by firstname");

System.***out***.println("4.Return to EmployeeUI");

1. Sale UI:

Salesman Operations:

1. View and update customer

2. Create an Order

3. Access sales reports

4. Goto loginUI;

Please select your operation(for example: a):

1: Please input your SQL operation about View and update customer:

1. Engineer UI:

1. System.***out***.println("Engineer Operations:");

System.***out***.println("1.View Model");

System.***out***.println("2. Search in Model”);

System.***out***.println("2.Update Model");

System.***out***.println("3.View Inventory");

System.***out***.println("2. Search in Inventory”);

System.***out***.println("4.Update Inventory");

System.***out***.println("5.View Employee Information");

System.***out***.println("5.Please input your SQL operation:");

System.***out***.println("6.Goto Login UI");

System.***out***.println("Please Select Your Operation(for exmaple: 1):");

Search in Model UI:

System.***out***.println("1. Search by ModelID”);

System.***out***.println("2. Search by ModelName”);

System.***out***.println("3. Return to EgineerUI”);

Search in Inventory UI:

System.***out***.println("1. Search by ModelID”);

System.***out***.println("2. Search by ModelName”);

System.***out***.println("3. Return to EgineerUI”);

Update Model UI:

System.***out***.println("1. Update Saleprice by ModeID”);

System.***out***.println("3. Return to EgineerUI”);

Update InventoryUI

System.***out***.println("1. Update Number by ID”);

System.***out***.println("1. Update Model\_ID by ID”);

System.***out***.println("3. Return to EgineerUI”);

How to create another database in postgres and delete a database in postgres?

1. Create another database

create database datebasename;

Using \l to check if it is created or not;

Using \c to connect the database;

1. create a super user to database;

Create user username with password ‘ooo’;

Grant all privileges on database databasename to username;

Alter role username with password ‘00’; change password;

1. Delete the database

Drop database databasename;

But you need to connect to another database;