Online Shopping & Logistics Services

Basic Information

Course: CS6360.003
Team Number: 1
Team Member: Wenbo Du, wxd200001

Read Me

In addition to this report, the folder also contains these items:

- create-database.sql stores all the program of this project.
- ER Diagram.drawio is a document used for designing ER model, and ER Diagram.png is its generated picture.
- Relational Schema.drawio is used for designing relational model, and Relational Schema.png is its generated picture.
- Step8, Step8-1 and Step8-1 are the screenshots for APEX.

Data Requirements

This is a database for online shopping and logistics company services system, which aims to provide an efficient management platform for e-commerce. The system includes all links of online shopping, from the user's purchase in the store to the management of logistics company.
The requirements of database system are shown below:
First, the system has 6 kinds of entities, which are user, shop, item, group, bill and logistic company.

- Each user has a unique id number, which is also the key attribute. Besides, the user should set its name, gender and the country it belongs to, submitting mobile number and address. One user is allowed to have multiple addresses, and passwords are also required.
- Entity in shop set also has a unique id number (the key attribute), its name, location and country. The shop may have multiple locations. In addition, the shop must have business license to prove it is qualified for business, and there is a credit rating whose default value is 0 for each shop to show the quality of items in it.
- Group is composed of multiple users, and users are able to exchange information and communicate with each other in a group. Groups are uniquely identified by their id number, and each of them contains group name, description, announcement, topics it covers and creation data. Topics may have multiple options.
- Item is also uniquely identified by their id number. Name, price, description, remaining quantity, popularity (the default value is 0), category and the data when it appears in store are also recorded. Category is a multivalued attribute. When a customer buys a product, the remaining quantity is reduced by one.

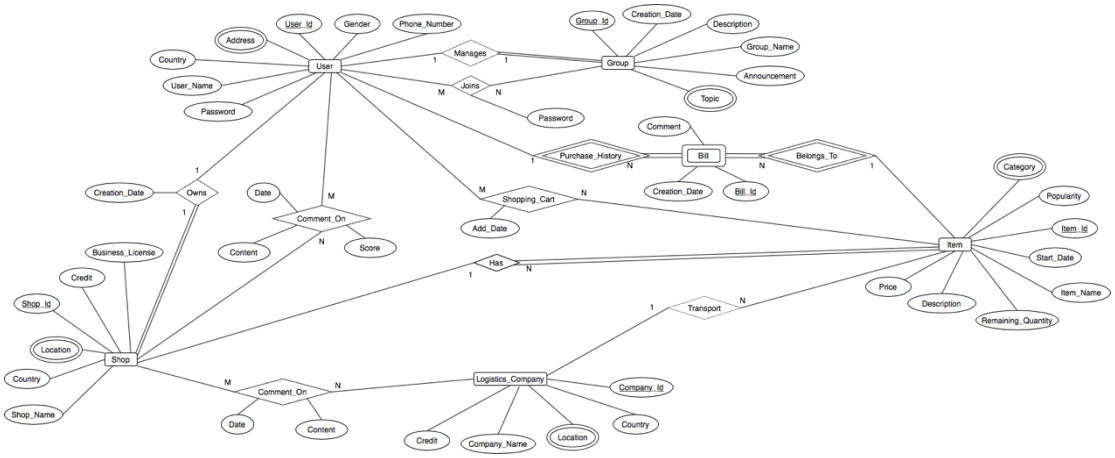
- Every time a user purchases an item, a bill is generated, so it is a weak entity. Each bill has its unique id number, and the data it was created, and user comments are also included.
- The logistics company is responsible for transporting every item, whose key attribute is its company id. And it also has attribute: company name, location, country and credit whose default value is 0. Location is a multivalued attribute.

Then, there are some relationships among these entities.

- In order to facilitate management, each user is allowed to have no more than one shop, and each shop must have its owner. This relation should indicate the creation data.
- Each shop can have multiple items, and each item must belong to a shop.
- Users have their own shopping cart to add some items they want, and the date adding the item is required.
- Each group must have a manager, and one user can only manage one group.
- Users can join multiple groups according to their interest.
- Users can evaluate the store, including the specific content, the score in the range of 0 to 10 (the default value is 10) and the data of making the comment.
- Shop can also evaluate the logistics company, which also include the specific content and the data of the comment.
- One logistics company can transport many different items.
- Each bill must connect with its buyer.
- Each bill must have the information of its item.

ER Diagram

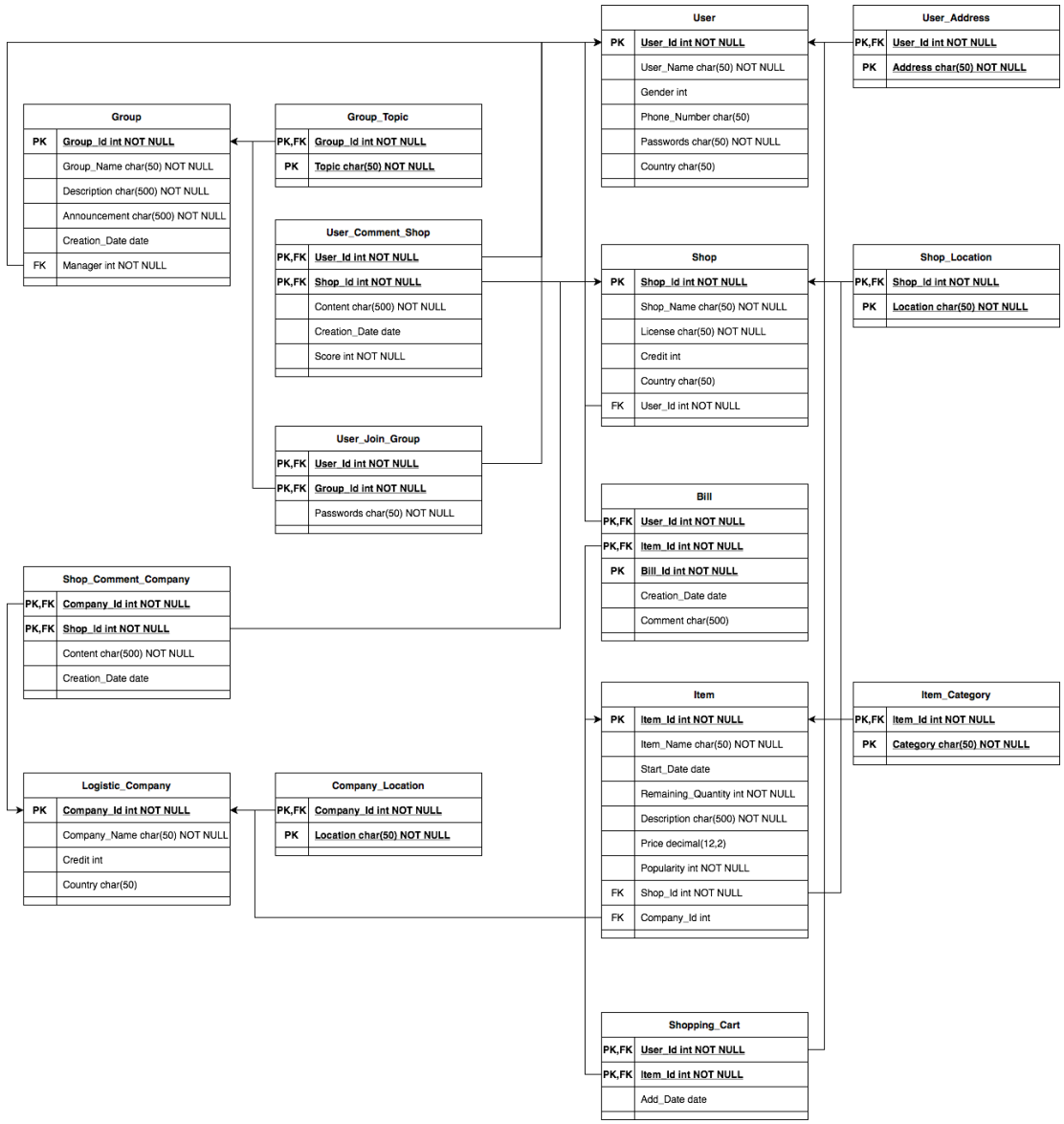
The ER diagram is shown below:



The source file is ER Diagram.drawio.

Relational Schema

The corresponding relational schema is shown below:



The source file is Relational Schema.drawio. The schema has already in 3rd normal forms.

Procedures & Triggers

The sql program are in create-database.sql file.

Procedures & Triggers

Then two stored procedures are designed:
The first one is about calculating the popularity of each item. The specific calculation method is as follows: when a user has bought this item, the value of its popularity

increases by 2, and when a user has added it into the shopping cart, the value of its popularity increases by 1. Thus, the total sum of these two categories represents the popularity degree. The program is as follows:

```
/* Procedures */
/* Once a user bought an item, the popularity of this item will increase by 2, and when the item was added
into a shopping cart, its popularity will increase by 1 */
create or replace view Bill_Count
as select Item_Id, count(*) as billitem FROM Bill group by Bill.Item_Id;

create or replace PROCEDURE Popularity_Calculation_Bill AS
CURSOR Item_CURSOR IS
SELECT Item_Id, count(*) as billitem FROM Bill group by Bill.Item_Id;
thisItem Bill_Count%ROWTYPE;
BEGIN
OPEN Item_CURSOR;
LOOP
  FETCH Item_CURSOR INTO thisItem;
  EXIT WHEN (Item_CURSOR%NOTFOUND);
  UPDATE Item SET Popularity = Popularity + thisItem.billitem * 2
  WHERE Item.Item_Id = thisItem.Item_Id;
END LOOP;
CLOSE Item_CURSOR;
END;

create or replace view Cart_Count
as select Item_Id, count(*) as cartitem FROM Shopping_Cart group by Shopping_Cart.Item_Id;

create or replace PROCEDURE Popularity_Calculation_Cart AS
CURSOR Item_CURSOR IS
SELECT Item_Id, count(*) as cartitem FROM Shopping_Cart group by Shopping_Cart.Item_Id;
thisItem Cart_Count%ROWTYPE;
BEGIN
OPEN Item_CURSOR;
LOOP
  FETCH Item_CURSOR INTO thisItem;
  EXIT WHEN (Item_CURSOR%NOTFOUND);
  UPDATE Item SET Popularity = Popularity + thisItem.cartitem
  WHERE Item.Item_Id = thisItem.Item_Id;
END LOOP;
CLOSE Item_CURSOR;
END;
```

And the second one is used to calculate the credit of shops. The calculation method is as follows: first collect all users' scores of this store which is in the relation named User_Comment_Shop. Then calculate the average of these scores. The program is shown below:

```
/* The store's credit is the average of all user reviews */
create or replace view Shop_Avg_Credit
as select Shop_Id, AVG(User_Comment_Shop.Score) as Avg_Credit FROM User_Comment_Shop group by User_Comment_Shop.Shop_Id;

create or replace PROCEDURE Credit_Calculation AS
CURSOR Shop_Credit IS
SELECT Shop_Id, AVG(User_Comment_Shop.Score) as Avg_Credit FROM User_Comment_Shop group by User_Comment_Shop.Shop_Id;
thisCredit Shop_Avg_Credit%ROWTYPE;
BEGIN
OPEN Shop_Credit;
LOOP
  FETCH Shop_Credit INTO thisCredit;
  EXIT WHEN (Shop_Credit%NOTFOUND);
  UPDATE Shop SET Credit = thisCredit.Avg_Credit
  WHERE Shop.Shop_Id = thisCredit.Shop_Id;
END LOOP;
CLOSE Shop_Credit;
END;
```

Besides, the system also has two triggers:

The first trigger is the constraint about score of the store made by user. Before updating or inserting a comment from a user, the system will check if its score is in the range of 0 to 10. If not, the system will raise an error. The program is:

```
/* Before adding or updating the score of shop made by user, system will check if the score is in the range of 0 to 10 */
create or replace TRIGGER Check_Score
AFTER INSERT OR UPDATE OF Score ON User_Comment_Shop
FOR EACH ROW
DECLARE
    New_Score number;
BEGIN
    New_Score := :NEW.Score;
    if New_Score < 0 then
        Raise_Application_Error(-20000, 'Score is too small');
    end if;
    if New_Score > 10 then
        Raise_Application_Error(-20000, 'Score is too big');
    end if;
END;
```

The second one is about the bill. Before creating a new bill, the system will check if the remaining quantity of the corresponding item is larger than 0. If not, this means the transaction cannot be completed. Otherwise, the bill will be created successfully, and the remaining quantity of this item will be reduced by 1. The program is:

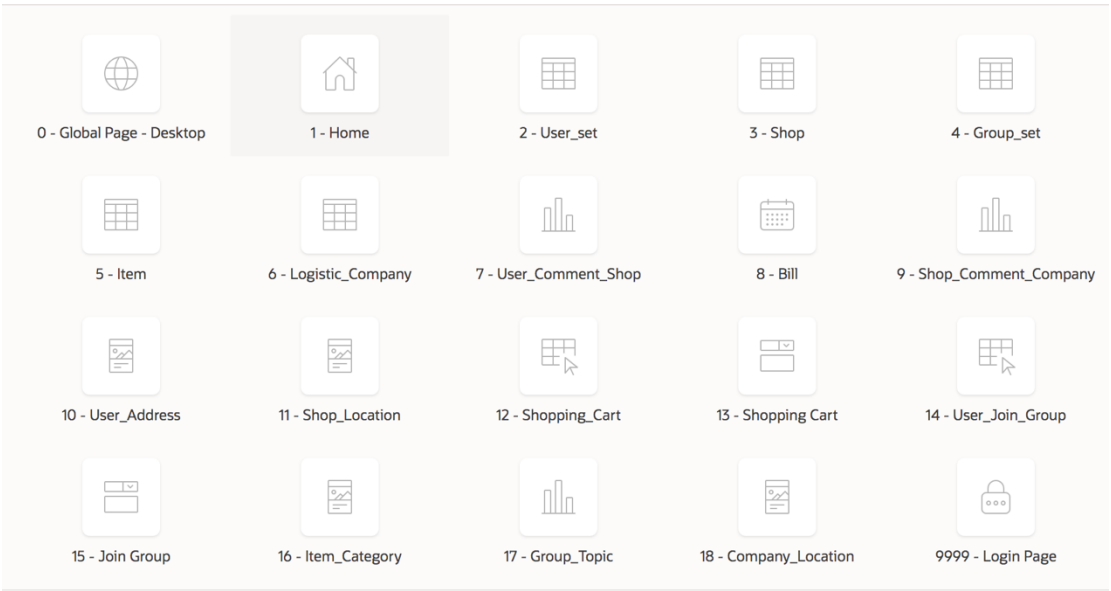
```
/* Before creating a bill, system will check if the remaining quantity of the corresponding item is large than 0 */
create or replace TRIGGER Check_Quantity
BEFORE INSERT ON Bill
FOR EACH ROW
DECLARE
    Item_Quantity number;
BEGIN
    select Item.Remaining_Quantity into Item_Quantity from Item where Item.Item_Id = :NEW.Item_Id;
    if Item_Quantity < 1 then
        Raise_Application_Error(-20000, 'There is not enough stock');
    end if;
    if Item_Quantity >= 1 then
        UPDATE Item SET Remaining_Quantity = Remaining_Quantity - 1
        WHERE Item.Item_Id = :NEW.Item_Id;
    end if;
END;
```

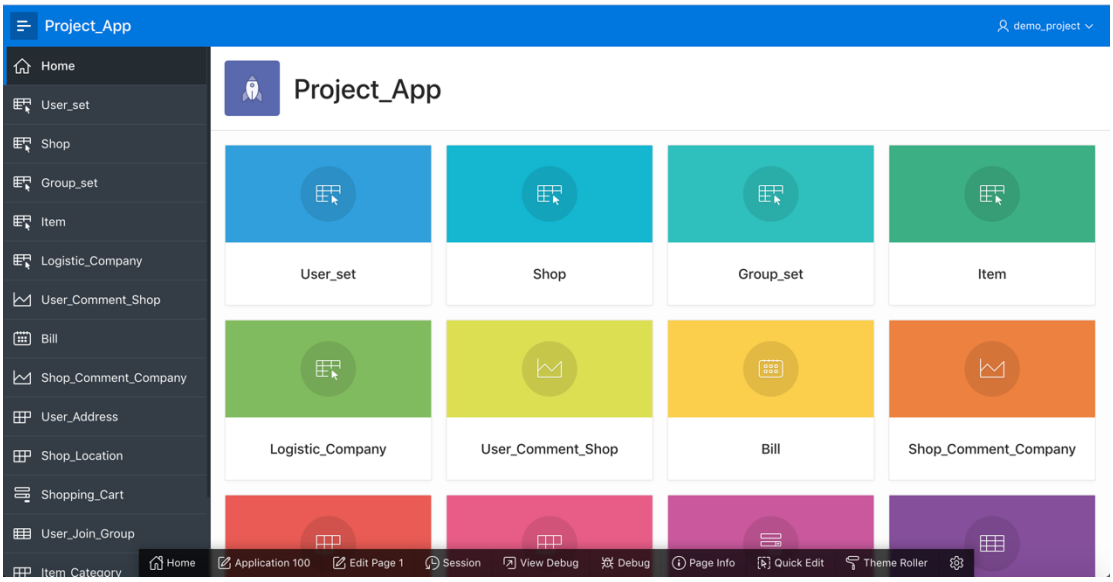
APEX Application

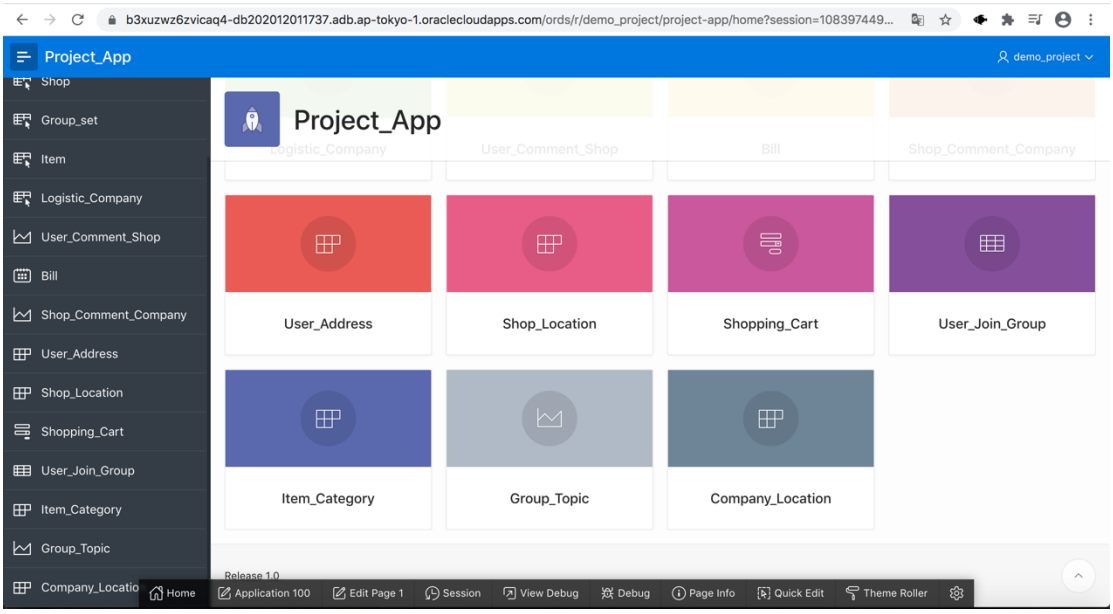
First, we create a workspace named Demo_Project, then importing the existing tables in it.

```
/* Import the tables into APEX */
CREATE TABLE DEMO_Project.User_set as (select * from ADMIN.User_set);
CREATE TABLE DEMO_Project.User_Address as (select * from ADMIN.User_Address);
CREATE TABLE DEMO_Project.Shop as (select * from ADMIN.Shop);
CREATE TABLE DEMO_Project.Shop_Location as (select * from ADMIN.Shop_Location);
CREATE TABLE DEMO_Project.Item as (select * from ADMIN.Item);
CREATE TABLE DEMO_Project.Item_Category as (select * from ADMIN.Item_Category);
CREATE TABLE DEMO_Project.Group_set as (select * from ADMIN.Group_set);
CREATE TABLE DEMO_Project.Group_Topic as (select * from ADMIN.Group_Topic);
CREATE TABLE DEMO_Project.Logistic_Company as (select * from ADMIN.Logistic_Company);
CREATE TABLE DEMO_Project.Company_Location as (select * from ADMIN.Company_Location);
CREATE TABLE DEMO_Project.Bill as (select * from ADMIN.Bill);
CREATE TABLE DEMO_Project.User_Comment_Shop as (select * from ADMIN.User_Comment_Shop);
CREATE TABLE DEMO_Project.Shop_Comment_Company as (select * from ADMIN.Shop_Comment_Company);
CREATE TABLE DEMO_Project.Shopping_Cart as (select * from ADMIN.Shopping_Cart);
CREATE TABLE DEMO_Project.User_Join_Group as (select * from ADMIN.User_Join_Group);
```

We design pages for each relation in relational schema, including interactive grid, calendar, interactive report, card and chart.
The screenshots are:


0 - Global Page - Desktop
1 - Home
2 - User_set
3 - Shop
4 - Group_set
5 - Item
6 - Logistic_Company
7 - User_Comment_Shop
8 - Bill
9 - Shop_Comment_Company
10 - User_Address
11 - Shop_Location
12 - Shopping_Cart
13 - Shopping Cart
14 - User_Join_Group
15 - Join Group
16 - Item_Category
17 - Group_Topic
18 - Company_Location
9999 - Login Page


Project_App
demo_project
Home
User_set
Shop
Group_set
Item
Logistic_Company
User_Comment_Shop
Bill
Shop_Comment_Company
User_Address
Shop_Location
Shopping_Cart
User_Join_Group
Item_Category
Home
Application 100
Edit Page 1
Session
View Debug
Debug
Page Info
Quick Edit
Theme Roller


b3xuzwz6zvicaq4-db202012011737.adb.ap-tokyo-1.oraclecloudapps.com/ords/r/demo_project/project-app/home?session=108397449...
Project_App
demo_project
Shop
Group_set
Item
Logistic_Company
User_Comment_Shop
Bill
Shop_Comment_Company
User_Address
Shop_Location
Shopping_Cart
User_Join_Group
Item_Category
Group_Topic
Company_Locatio
Release 1.0
Home
Application 100
Edit Page 1
Session
View Debug
Debug
Page Info
Quick Edit
Theme Roller