CSCI 3901 Assignment 5

Anqi Chen, B00838586

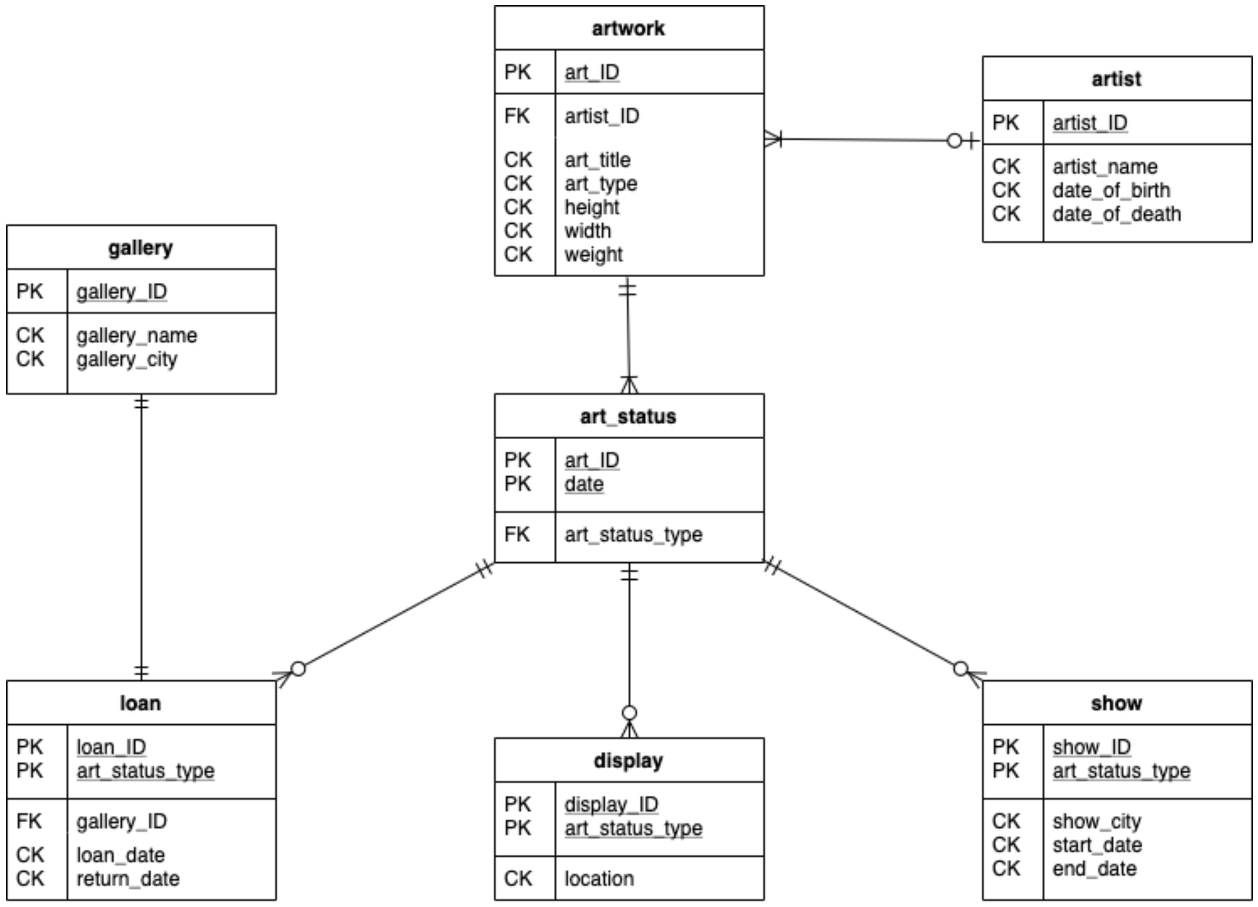
Problem 1

Goal

Design a database schema.

Solution

The entity relation graph for an art museum is designed like this:

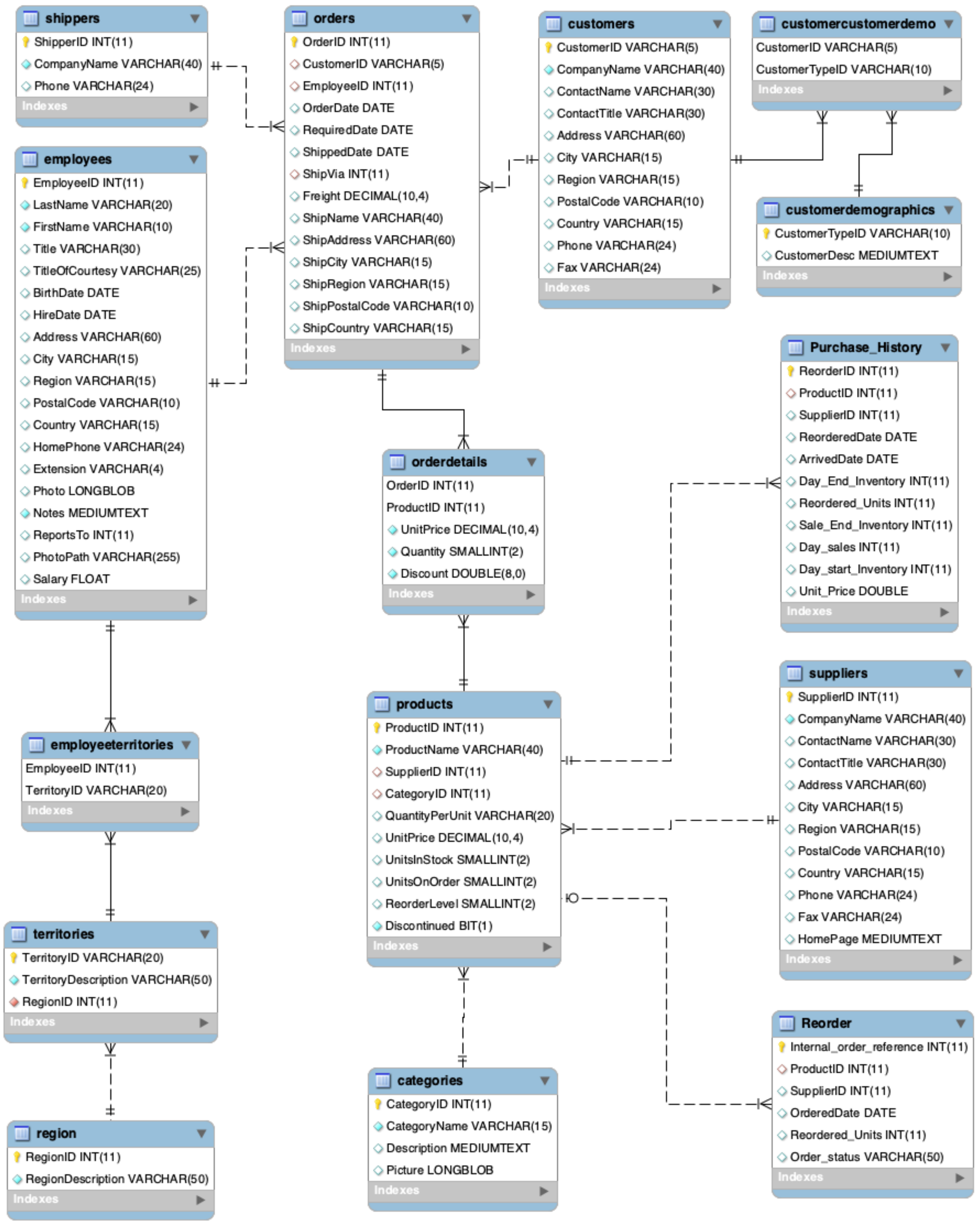


Problem 2

Goal

Make some meaningful changes to a database.

The ER diagram for the new database



The SQL statements to update the database

```
Create table Reorder (Internal_order_reference int (11) not null auto_increment,
        ProductID int (11) default null,
        SupplierID int (11) default null,
        OrderedDate date default null,
        Reordered_Units int (11) default null,
        Order_status varchar (50) default null,
        primary key (Internal_order_reference),
        foreign key (ProductID) references products (ProductID))


Create table Purchase_History (ReorderID int (11) not null auto_increment,
        ProductID int (11) default null,
        SupplierID int (11) default null,
        ReorderedDate date default null,
        ArrivedDate date default null,
        Day_End_Inventory int (11) default null,
        Reordered_Units int (11) default null,
        Sale_End_Inventory int (11) default null,
        Day_sales int (11) default null,
        Day_start_Inventory int (11) default null,
        Unit_Price double default null,
        primary key (ReorderID),
        foreign key (ProductID) references products (ProductID))
```

Structure and flow

MainConnect.java

1. Access the database

   Load the database driver; Create a Connection to connect the database with the logging information; Create a Statement object of the connection; Choose the database to use.

2. Main UI for the program
   Prompt the user to enter proper commends via the console table. It is more accessible for users and testers to interact with the program in a unified format.

MyIdentity.java

Use a properties structure to hide the information from other users. Set the database name that we want to access. Set the username and password for logging into the database.

InventoryControl.java

This class is simply constructing an interface for Ship_order, Issue_reorders and Receive_order methods.

Inventory.java

This is the most important class which implements the three core methods. When calling the Ship_order method, which means that the order is now shipped away, the shipped date in table orders will be set as the current date, also the units in stock will be cut down. When calling the Issue_reorders method, the internal order reference is created and for each order the useful information is inserted into table Reorder. Also, the method will return the number of suppliers from whom we will be placing an order. When calling the Receive_reorder method, the Order_status in Reorder table will be updated, then the number of stock units will also be updated by using the units quantity information from Reorder table.

PurchaseHistory.java

Using data from products, orders and orderdetails to build a history of purchases. To estimate the product cost, I use the sum of UnitPrice from orderdetails and the standard 15% markup principle as product cost = (cost / 1.15). Also, the sale_end_inv, daysale and day_start_inv are calculated in separate cases.

OrderException.java

The Ship_order and Receive_order methods can throw “OrderException”. The OrderException class allows the user to retrieve information on what went wrong with a getMessage() method and to retrieve a reference integer for the order in question with a getReference() method.

Assumptions

- The internal order reference number will be created like this: date + productid, for example, the identifier for an order is 2020040627 means that the product is reordered on 2020/04/06 and its id is 27.

- When it comes to establishing the item cost in the purchase order, assume that our company has a standard 15% markup on the price, so use the price in the last sale of the day to estimate the item cost.

- The reorder level is set to be compared with ¼ of the units in stock.

Limitations

- The way of setting reorder level is now hard-coded, so it is difficult to have frequent changes as market conditions change;
- The separate methods should be called in different session, not within the same connection to database.

Testing

After running the program, write some sql statements to check for date in related tables: Purchase_History, ReOrder, orders, orderdetails and products.

References

MySQL Java tutorial—MySQL programming in Java with JDBC. (n.d.). Retrieved April 7, 2020, from http://zetcode.com/db/mysqljava/

Soam, T. (2018, February 3). *Create ER Diagram of a Database in MySQL Workbench*. Medium. https://medium.com/@tushar0618/how-to-create-er-diagram-of-a-database-in-mysql-workbench-209fbf63fd03