数据库实验报告

年级	13 级	专业（方向）	移动信息工程
学号	13354485	姓名	朱琳
电话	13726231932	Email	280273861@qq.com

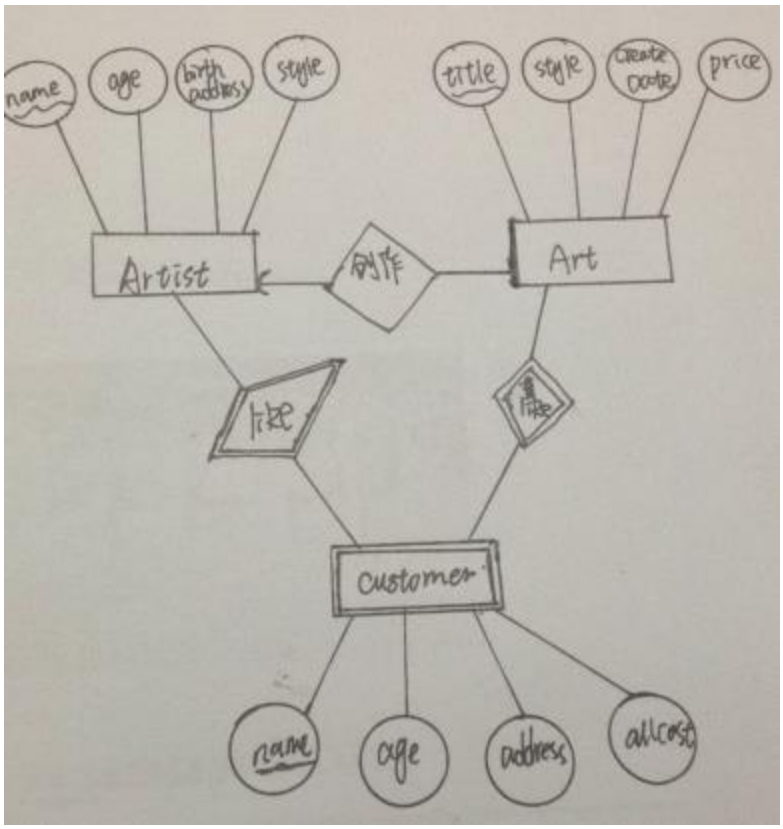
一. 实验题目

尽管你一直想成为艺术家，但最终成为了一个数据库领域的专家，可是心中的艺术梦没有破灭，于是你决定成立一个数据库公司"Artbase"为艺术馆制作一个产品。产品的核心是一个数据库，需要储存的信息有艺术家、作品、顾客，对于风格类似的作品需要将其分成组，例如肖像、风景、19 世纪的作品等。当然每一组都有一个描述标题。对于每位顾客都需要保存他们的一些基本信息（年龄、住所）、他们在艺术馆花费的总和以及喜爱的艺术家或者作品组。对于艺术家的信息，除了基本信息之外（姓名、年龄、出生地），还需要储存他们的艺术风格。而对于作品，需要储存的信息有标题（唯一）、艺术风格、创作年份、和价格。

（1）根据题目信息描绘出 ER 图。

（2）根据 ER 图创建一个"Artbase"数据库，并建表，对每张表进行增删查改操作，注意添加相应的依赖关系。

二.ER 图



三 思路

在这里我将艺术家和作品作为两个实体集，两者之间存在一个关系集合 made，即艺术家创作了艺术品。艺术家的属性是姓名，年龄，出生地，作品风格，其中姓名作为主键。作品的属性为标题，风格，创作日期，价格，其中标题为主键。而顾客作为一个弱实体，其依附于作品和艺术家，可以理解为没有市场就没有顾客。

四. 主要代码

```
CREATE TABLE Artist
(
    name varchar(20),
    age int,
    birthAddress varchar(20),
    style varchar(20),
    primary key(name)
);
CREATE TABLE Art
(
    title   varchar(20),
    style   varchar(20),
    createDate  varchar(20),
    price  int,
    primary key(title)
);

CREATE TABLE Customer
(
    name varchar(20),
    address varchar(20),
    age int,
    allCost int,
    title varchar(20),
    favorit_name varchar(20),
    primary key(title,favorit_name,name),
    foreign key(title) references Art(title),
    foreign key(favorit_name) references Artist(name)
);
```

```
CREATE TABLE Made
(
    name varchar(20),
    title   varchar(20),
    primary key(title),
    foreign key(name) references Artist(name),
    foreign key(title) references Art(title)
);

use Artbase;
insert into Artist values("John",30,"China","portrait");
insert into Artist values("Bob",80,"Lendon","portrait");
insert into Artist values("Ann",25,"China","scenery");

insert into Art values("Queen","portrait","20101009","1000");
insert into Art values("King","portrait","20080108","200000");
insert into Art values("Linda","portrait","20100126","5000");
insert into Art values("Beijing","scenery","20150108","2000");
insert into Art values("Zhuhai","scenery","20150101","2000");

insert into Made values("John","Queen");
insert into Made values("John","King");
insert into Made values("Bob","Linda");
insert into Made values("Ann","Beijing");
insert into Made values("Ann","Zhuhai");

insert into Customer values("Hello","China",20,8880,"Beijing","Ann");
insert into Customer values("Kitty","Zhuhai",50,100000,"Zhuhai","Ann");
```

【说明】我在创建完数据库之后对每一个表格添加了一定的元素。这就是初始的数据。

五 实验结果

（1）我将 art.sql 文件导入 mysql 之后表格情况：

```
mysql> select * from Art;
+---------+---------+------------+--------+
| title   | style   | createDate | price  |
+---------+---------+------------+--------+
| Beijing | scenery | 20150108   |   2000 |
| King    | portrait | 20080108   | 200000 |
| Linda   | portrait | 20100126   |   5000 |
| Queen   | portrait | 20101009   |   1000 |
| Zhuhai  | scenery | 20150101   |   2000 |
+---------+---------+------------+--------+
5 rows in set (0.00 sec)

mysql> select * from Artist;
+------+------+--------------+----------+
| name | age  | birthAddress | style    |
+------+------+--------------+----------+
| Ann  |   25 | China        | scenery  |
| Bob  |   80 | Lendon       | portrait |
| John |   30 | China        | portrait |
+------+------+--------------+----------+
3 rows in set (0.00 sec)

mysql> select * from Made;
+------+---------+
| name | title   |
+------+---------+
| Ann  | Beijing |
| Ann  | Zhuhai  |
| Bob  | Linda   |
| John | King    |
| John | Queen   |
+------+---------+
5 rows in set (0.00 sec)

mysql> select * from Customer;
+-------+---------+------+---------+---------+---------------+
| name  | address | age  | allCost | title   | favorit_name  |
+-------+---------+------+---------+---------+---------------+
| Hello | China   |   20 |    8880 | Beijing | Ann           |
| Kitty | Zhuhai  |   50 |  100000 | Zhuhai  | Ann           |
+-------+---------+------+---------+---------+---------------+
```

【分析】可以看到我在 art.sql 中所有的增加表格元素语句都表现在了数据库中。

（2）对 Artist 进行增加操作

```
mysql> select * from Artist;
+-------+------+---------------+----------+
| name  | age  | birthAddress  | style    |
+-------+------+---------------+----------+
| Ann   |   25 | China         | scenery  |
| Bob   |   80 | Lendon        | portrait |
| Bobby |   18 | HongKong      | scenery  |
| John  |   30 | China         | portrait |
+-------+------+---------------+----------+
4 rows in set (0.00 sec)

mysql> insert into Artist(name,style) value("Monkey","scenery");
Query OK, 1 row affected (0.13 sec)

mysql> select * from Artist;
+--------+------+---------------+----------+
| name   | age  | birthAddress  | style    |
+--------+------+---------------+----------+
| Ann    |   25 | China         | scenery  |
| Bob    |   80 | Lendon        | portrait |
| Bobby  |   18 | HongKong      | scenery  |
| John   |   30 | China         | portrait |
| Monkey | NULL | NULL          | scenery  |
+--------+------+---------------+----------+
5 rows in set (0.00 sec)
```

（3）对 Art 和 made 表格进行添加。注意只有先添加 Art 和 Artist 之后才能添加 made 将两者联系起来

```
mysql> insert into Art(title,style,price) value("Hometwon","scenery",100000);
Query OK, 1 row affected (0.08 sec)

mysql> select * from Art;
+----------+----------+------------+--------+
| title    | style    | createDate | price  |
+----------+----------+------------+--------+
| Beijing  | scenery  | 20150108   |   2000 |
| Hometwon | scenery  | NULL       | 100000 |
| King     | portrait | 20080108   | 200000 |
| Linda    | portrait | 20100126   |   5000 |
| Queen    | portrait | 20101009   |   1000 |
| Zhuhai   | scenery  | 20150101   |   2000 |
+----------+----------+------------+--------+
6 rows in set (0.00 sec)

mysql> insert into Made value("Bobby","Hometwon");
Query OK, 1 row affected (0.05 sec)

mysql> select * from Made;
+-------+----------+
| name  | title    |
+-------+----------+
| Ann   | Beijing  |
| Ann   | Zhuhai   |
| Bob   | Linda    |
| Bobby | Hometwon |
| John  | King     |
| John  | Queen    |
+-------+----------+
6 rows in set (0.00 sec)
```

（4）对 Artist 表格进行删除操作&对 Artist 表格中的 name=Bobby 选项的年龄进行更新

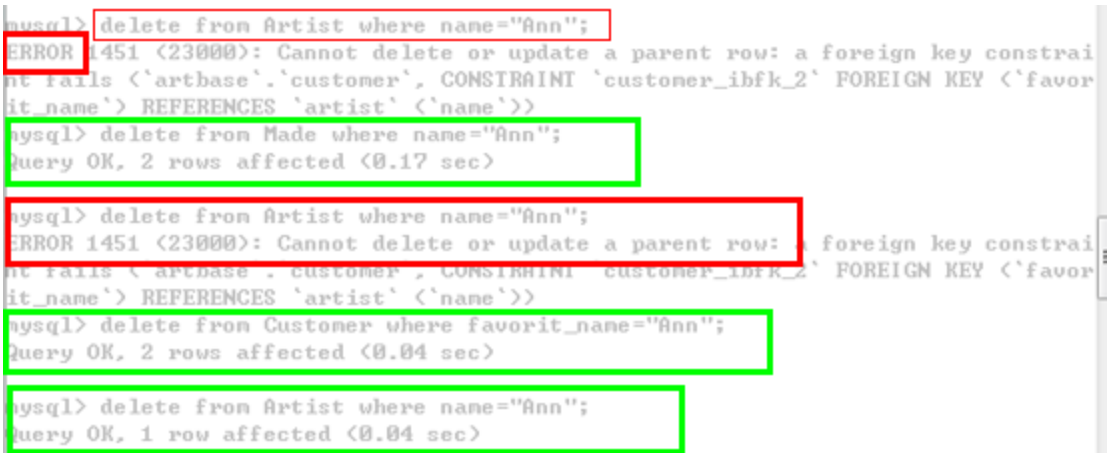
```
mysql> delete from Artist where name="Monkey";
Query OK, 1 row affected (0.25 sec)

mysql> select * from Artist;
+-------+------+--------------+----------+
| name  | age  | birthAddress | style    |
+-------+------+--------------+----------+
| Ann   |   25 | China        | scenery  |
| Bob   |   80 | Lendon       | portrait |
| Bobby |   18 | HongKong     | scenery  |
| John  |   30 | China        | portrait |
+-------+------+--------------+----------+
4 rows in set (0.00 sec)

mysql> update Artist set age="20" where name="Bobby";
Query OK, 1 row affected (0.21 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from Artist;
+-------+------+--------------+----------+
| name  | age  | birthAddress | style    |
+-------+------+--------------+----------+
| Ann   |   25 | China        | scenery  |
| Bob   |   80 | Lendon       | portrait |
| Bobby |   20 | HongKong     | scenery  |
| John  |   30 | China        | portrait |
+-------+------+--------------+----------+
4 rows in set (0.00 sec)
```

（5）删除操作



【分析】如图，红色部分是出错的地方，原因就是我没有先删去联系，再去删掉实体集，由此可证明我们必须先删掉两个实体集之间的联系，所有绿色框出的部分就是正确的指令，即先删掉 made，再删掉 Customer 中的相关信息（因为 Customer 作为一个弱实体是被 Art 和 Artist 所联系的），最后再去删掉 Art 和 Artist 中的相关信息，这样的指令才是合法指令

六 实验感想

1.本次实验总体感觉还是比较好实现的，主要是学习了外键的使用。开始不知道什么样的才能作为外键，写完之后发现导入 mysql 的时候还是不能够导入。通过对 TA 代码的观察和查资料，发现只有在别的实体集中作为主键的东西才能作为联系表和弱实体的外键。

2.将自己写的.sql 语句进行导入 mysql 的步骤：

①打开 cmd，将当前目录 cd 到.sql 所在的目录

②打开 mysql，创建 dbname。

③在命令行输入 mysql -h localhost -u root -p **dbname < xxx.sql，** 注意将黑体部分替换掉

3. 这次出乎意料在命令行出现了一些问题。有些是自己写错了，而有的时候他会告诉我 update 失败。后来发现是不能重复导入的，即使第一次导入失败。所以需要在 mysql 中将创建的 database 删掉再重建。

4.这是第一次写数据库的实验，我对一些指令还不是很熟悉，通过不断的练习，现在可以实现一些基本的操作，心里还是很开心。