> **同济大学大数据管理系统实验报告01**

《MySQL关系数据库使用》

> - 学院：电子信息工程学院

> - 专业：数据科学与大数据专业
> - 学号：1851804
> - 姓名：苗成林
> - 指导教师：李文根
> - 时间：2021.10.26

实验目的及要求：

1. 安装MySQL关系数据库
2. 基于模式图创建数据库
3. 执行基本的SQL语句

实验原理：

实体-联系模型

三个基本概念：实体集、联系集和属性

参与

实体集之间的关联：实体集E1,E2,E3,...,EnE1,E2,E3,...,En参与联系集R

联系实例

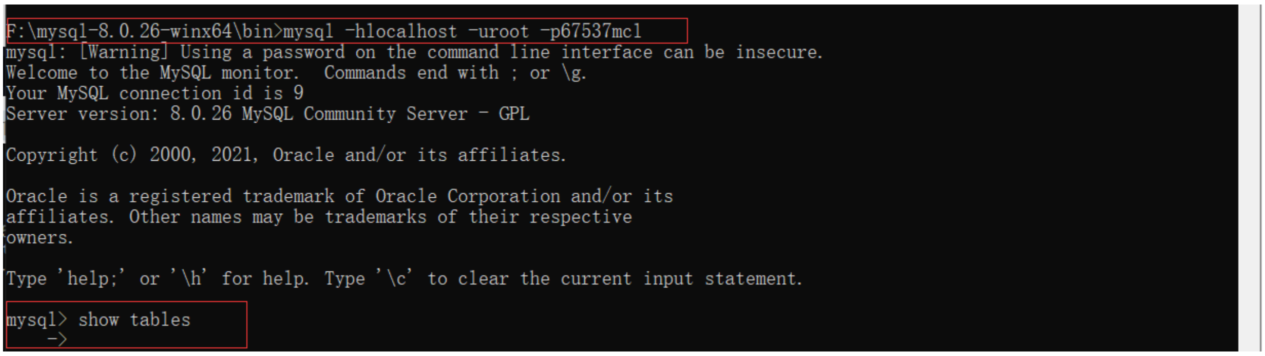
在锁建模的显式世界中命名实体间的一个关联，比如一个教师ID位9527的instructor实体Katz和一个学生ID为12345的student实体Shankar参与到advisor的一个联系实例中。这一联系实例表示在大学中教师Katz指导学生Shankar

描述属性

联系集的描述属性，比如实体集instructor和student之间的联系集advisor。我们可以将属性date与该联系关联起来，称为具有描述属性date的联系集advisor
给定一个联系集中的一个联系实例必须是由其参与实体唯一标识的，而不必使用其描述属性

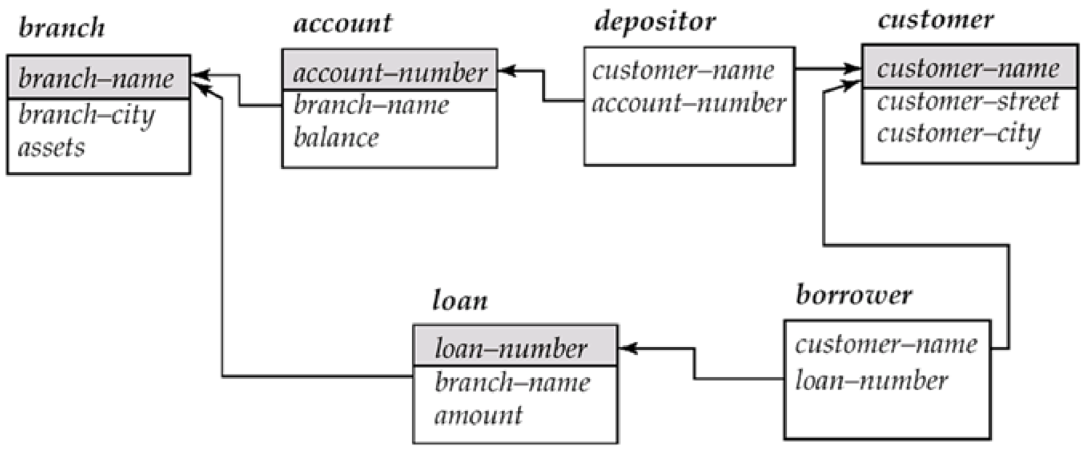
实验过程：

启动MySQL



```
F:\mysql-8.0.26-winx64\bin>mysql -hlocalhost -uroot -p67537mcl
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.26 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show tables
    ->
```

银行数据库模式图



创建数据库及相应数据表

branch

```
CREATE TABLE IF NOT EXISTS branch(
    branch_name VARCHAR(50) PRIMARY KEY,
    branch_city enum("SHANGHAI","NEWYORK","BEIJING","LOS ANGELES") NOT NULL DEFAULT"SHANGHAI",
    assets BIGINT(20) NOT NULL
)ENGINE=INNODB CHARSET=UTF8;
```

account

```
CREATE TABLE IF NOT EXISTS account(
account_number INT(20) PRIMARY KEY AUTO_INCREMENT NOT NULL,
branch_name VARCHAR(50),
FOREIGN KEY(branch_name) REFERENCES branch(branch_name),
balance BIGINT(20)
)ENGINE=INNODB CHARSET=UTF8;
```

loan

```
CREATE TABLE IF NOT EXISTS loan(
loan_number INT(10) PRIMARY KEY AUTO_INCREMENT NOT NULL,
branch_name VARCHAR(50),
FOREIGN KEY(branch_name) REFERENCES branch(branch_name),
amount BIGINT(20)
)ENGINE=INNODB CHARSET=UTF8;
```

depositor

```
CREATE TABLE IF NOT EXISTS depositor(
customer_name VARCHAR(20),
account_number INT(20) AUTO_INCREMENT NOT NULL,
FOREIGN KEY(account_number) REFERENCES account(account_number)
)ENGINE=INNODB CHARSET=UTF8;
```

customer

```sql
CREATE TABLE IF NOT EXISTS customer(
customer_name VARCHAR(20) PRIMARY KEY NOT NULL,
customer_street VARCHAR(50),
customer_city VARCHAR(50)
)ENGINE=INNODB CHARSET=UTF8;
```

borrower

```sql
CREATE TABLE IF NOT EXISTS borrower(
customer_name VARCHAR(20) NOT NULL,
FOREIGN KEY(customer_name) REFERENCES customer(customer_name),
loan_number INT(10) AUTO_INCREMENT NOT NULL,
FOREIGN KEY(loan_number) REFERENCES loan(loan_number)
)ENGINE=INNODB CHARSET=UTF8;
```

指定主码和外码依赖

```sql
#指定从depositor到customer的外码依赖
#其他外码依赖在创建表时已指定
ALTER TABLE depositor
ADD FOREIGN KEY(customer_name) REFERENCES customer(customer_name);
```

向表中插入样例数据

branch

```sql
INSERT INTO branch(branch_name,branch_city,assets)
VALUES("NY0001","NEWYORK",54000000),
("NY0011","NEWYORK",564000000),
("NY0511","NEWYORK",1114000000),
("SH000411","SHANGHAI",54000000),
("SH08411","SHANGHAI",54000550000),
("SH00011","SHANGHAI",54010330000),
("BJ9999999","BEIJING",54000550000),
("BJ0899411","BEIJING",54000550000),
("LA020411","LOS ANGELES",54000550000);
```

account

```sql
INSERT INTO account(account_number,branch_name,balance)
VALUES(800921,"NY0011",250000000),
(800922,"SH00011",25000),
(800923,"SH000411",25000),
(800924,"NY0511",10000000),
(800925,"SH00011",500000),
(800926,"BJ0899411",54000),
(100921,"SH08411",25000000),
(100922,"SH00011",25000),
(100923,"SH000411",25000),
(110921,"LA020411",10000000),
(110922,"BJ9999999",500000),
(110923,"BJ0899411",54000),
(188921,"LA020411",100010000),
(171021,"NY0011",250000000),
(171022,"SH00011",25000),
(171023,"SH000411",25000),
(171024,"NY0511",10000000),
(171025,"SH00011",500000),
(171026,"LA020411",10000);
```

loan

```
INSERT INTO loan(loan_number,branch_name,amount)
VALUES(99800921,"NY0011",250000000),
(99800922,"SH00011",25000),
(99800923,"SH000411",25000),
(99800924,"NY0511",10000000),
(99800925,"SH00011",500000),
(99800926,"BJ0899411",54000),
(99100921,"SH08411",25000000),
(99100922,"SH00011",25000),
(99100923,"SH000411",25000),
(99110921,"LA020411",10000000),
(99110922,"BJ9999999",500000),
(99110923,"BJ0899411",54000),
(99188921,"LA020411",100010000),
(99171021,"NY0011",250000000),
(99171022,"SH00011",25000),
(99171023,"SH000411",25000),
(99171024,"NY0511",10000000),
(99171025,"SH00011",500000),
(99171026,"LA020411",10000);
```

depositor

```
INSERT INTO depositor(customer_name,account_number)
VALUES("Smith",800921),
("Smith",800922),
("Smith",800923),
("Smith",800924),
("Smith",800925),
("Smith",800926),
("Liu",100921),
("Liu",100922),
("Liu",100923),
("Li",110921),
("Li",110922),
("Li",110923),
("Miao",188921),
("Su",171021),
("Su",171022),
("Su",171023),
("Su",171024),
("Su",171025),
("Su",171026);
```

customer

```
INSERT INTO customer(customer_name,customer_street,customer_city)
VALUES("Smith","Harrison","NEWYORK"),
("Liu","Harrison","NEWYORK"),
("Li","Queen","NEWYORK"),
("Miao","Jiading","SHANGHAI"),
("Su","Jingan","SHANGHAI"),
("Yang","Jiading","SHANGHAI"),
("Zoey","Jingan","SHANGHAI"),
("Xue","Fengcheng","BEIJING"),
("Han","Chaoyang","BEIJING"),
("William","Central","London"),
("Peck","Central","LOS ANGELES");
```

borrower

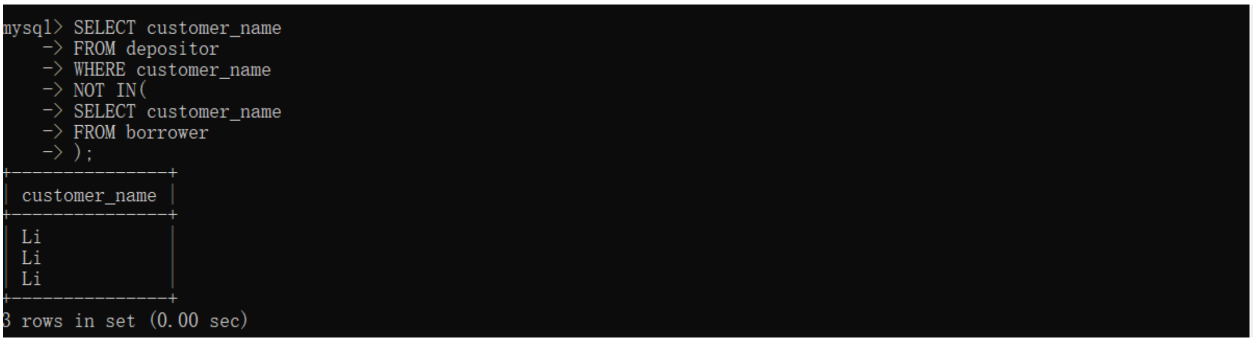
```
INSERT INTO borrower(customer_name,loan_number)
VALUES("Smith",99800921),
("Smith",99800922),
("Smith",99800923),
("Yang",99800924),
("Yang",99800925),
("Yang",99800926),
("Liu",99100921),
("Liu",99100922),
("Liu",99100923),
("Zoey",99110921),
("Zoey",99110922),
("Zoey",99110923),
("Miao",99188921),
("Han",99171021),
("William",99171022),
("Peck",99171023),
("Su",99171024),
("Su",99171025),
("Su",99171026);
```

使用SQL完成作业查询

找出银行中所有有账户但无贷款的客户

```
SELECT customer_name
FROM depositor
WHERE customer_name
NOT IN(
SELECT customer_name
FROM borrower
);
```

查询结果

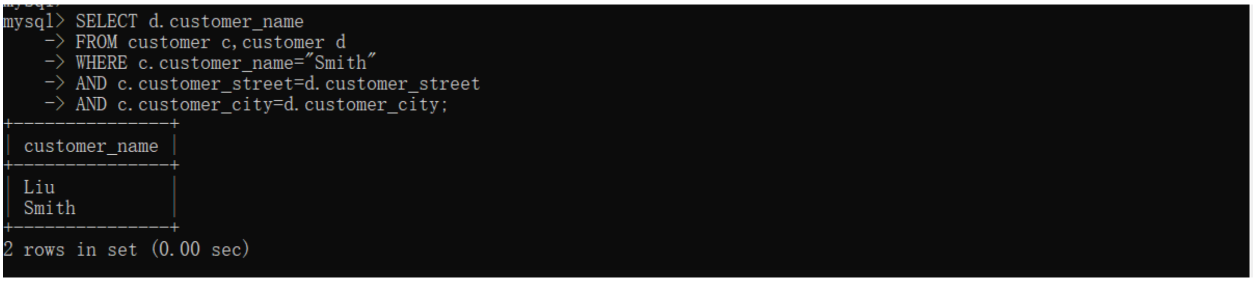


```
mysql> SELECT customer_name
    -> FROM depositor
    -> WHERE customer_name
    -> NOT IN(
    -> SELECT customer_name
    -> FROM borrower
    -> );
+---------------+
| customer_name |
+---------------+
| Li            |
| Li            |
| Li            |
+---------------+
3 rows in set (0.00 sec)
```

找出与“Smith”居住在同一城市、同一街道的所有客户的名字

```
SELECT d.customer_name
FROM customer c,customer d
WHERE c.customer_name="Smith"
AND c.customer_street=d.customer_street
AND c.customer_city=d.customer_city;
```

查询结果

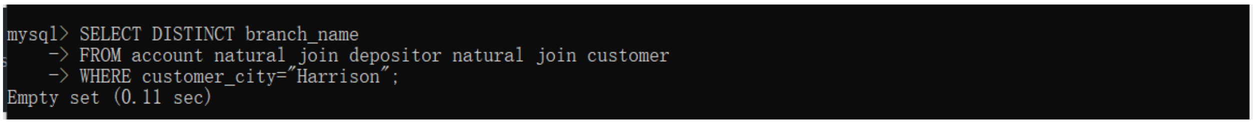


```
mysql> SELECT d.customer_name
    -> FROM customer c,customer d
    -> WHERE c.customer_name="Smith"
    -> AND c.customer_street=d.customer_street
    -> AND c.customer_city=d.customer_city;
+---------------+
| customer_name |
+---------------+
| Liu           |
| Smith         |
+---------------+
2 rows in set (0.00 sec)
```

找出所有支行的名称，在这些支行中都有居住在“Harrion”的客户所开设的账户

```
SELECT DISTINCT branch_name
FROM account natural join depositor natural join customer
WHERE customer_city="Harrison";
```

查询结果

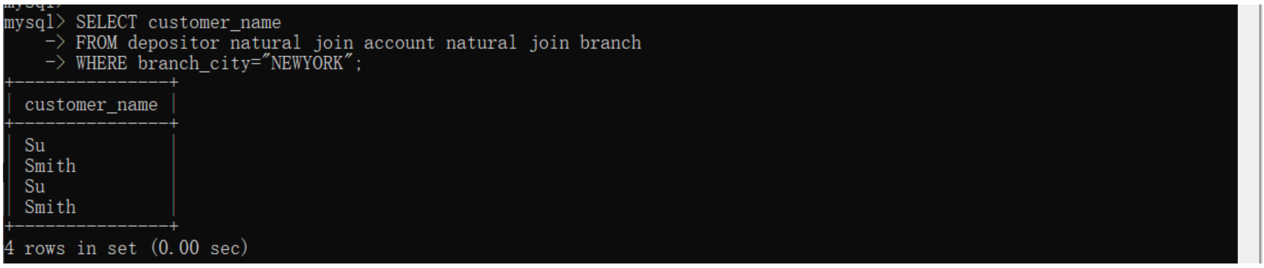


```
mysql> SELECT DISTINCT branch_name
    -> FROM account natural join depositor natural join customer
    -> WHERE customer_city="Harrison";
Empty set (0.11 sec)
```

找出在“NEWYORK”的所有支行都有账户的所有客户

```sql
SELECT customer_name
FROM depositor natural join account natural join branch
WHERE branch_city="NEWYORK";
```

查询结果

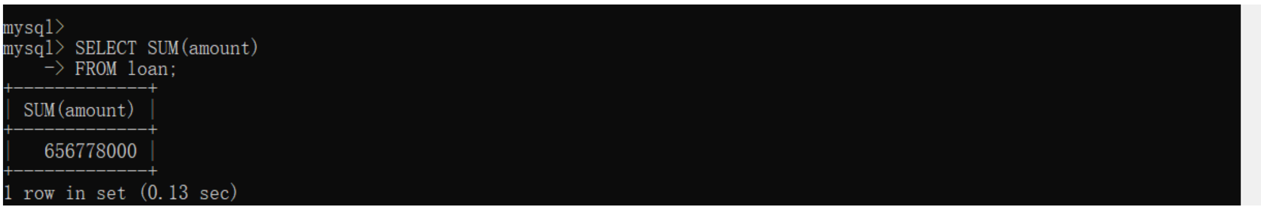


```
mysql>
mysql> SELECT customer_name
    -> FROM depositor natural join account natural join branch
    -> WHERE branch_city="NEWYORK";
+---------------+
| customer_name |
+---------------+
| Su            |
| Smith         |
| Su            |
| Smith         |
+---------------+
4 rows in set (0.00 sec)
```

找出银行的所有贷款额的总和

```sql
SELECT SUM(amount)
FROM loan;
```

查询结果

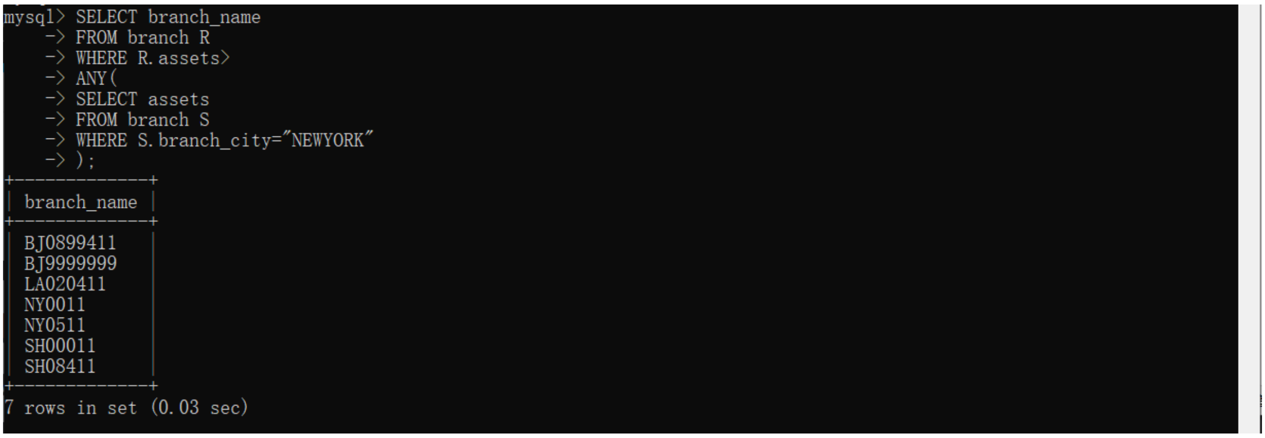


```
mysql>
mysql> SELECT SUM(amount)
    -> FROM loan;
+-------------+
| SUM(amount) |
+-------------+
|   656778000 |
+-------------+
1 row in set (0.13 sec)
```

找出总资产至少比位于Brooklyn的某一家支行要多的所有支行的名字

```sql
SELECT branch_name
FROM branch R
WHERE R.assets>
ANY(
SELECT assets
FROM branch S
WHERE S.branch_city="NEWYORK"
);
```

查询结果

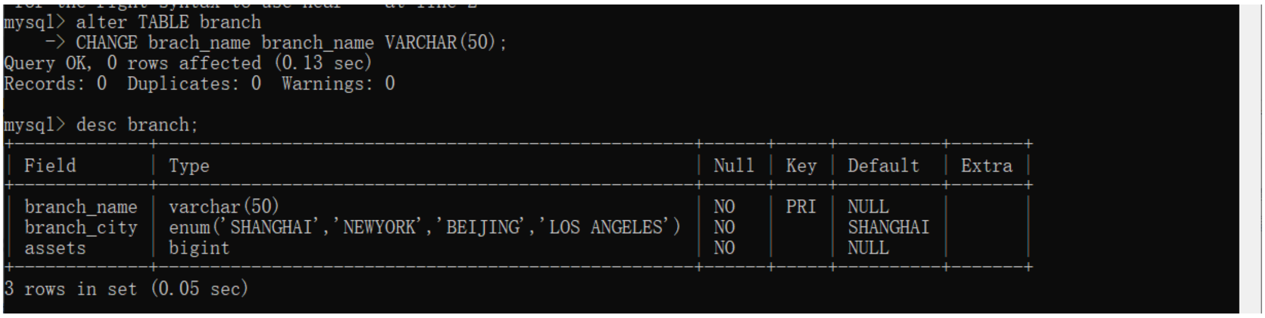


```
mysql> SELECT branch_name
    -> FROM branch R
    -> WHERE R.assets>
    -> ANY(
    -> SELECT assets
    -> FROM branch S
    -> WHERE S.branch_city="NEWYORK"
    -> );
+-------------+
| branch_name |
+-------------+
| BJ0899411   |
| BJ9999999   |
| LA020411    |
| NY0011      |
| NY0511      |
| SH00011     |
| SH08411     |
+-------------+
7 rows in set (0.03 sec)
```

问题总结：

修改查看数据库属性

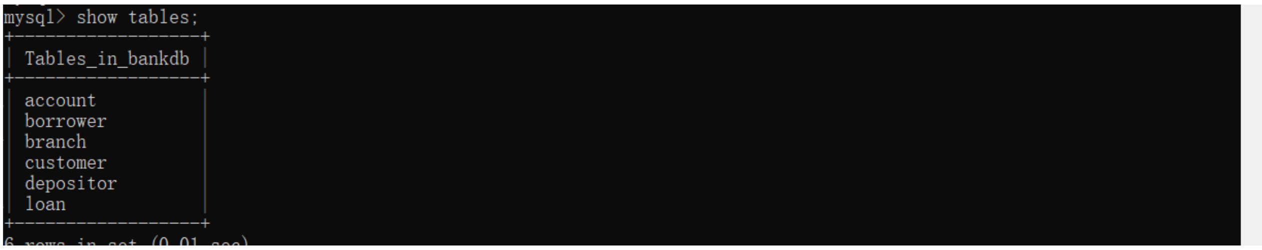
```
ALTER TABLE branch
CHANGE brach_name branch_name VARCHAR(50);
```



```
mysql> alter TABLE branch
    -> CHANGE brach_name branch_name VARCHAR(50);
Query OK, 0 rows affected (0.13 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc branch;
+-------------+---------------------------------------------------+------+-----+----------+-------+
| Field       | Type                                              | Null | Key | Default  | Extra |
+-------------+---------------------------------------------------+------+-----+----------+-------+
| branch_name | varchar(50)                                       | NO   | PRI | NULL     |       |
| branch_city | enum('SHANGHAI','NEWYORK','BEIJING','LOS ANGELES') | NO   |     | SHANGHAI |       |
| assets      | bigint                                            | NO   |     | NULL     |       |
+-------------+---------------------------------------------------+------+-----+----------+-------+
3 rows in set (0.05 sec)
```

查看数据表

```
SHOW tables;
```



```
mysql> show tables;
+------------------+
| Tables_in_bankdb |
+------------------+
| account          |
| borrower         |
| branch           |
| customer         |
| depositor        |
| loan             |
+------------------+
6 rows in set (0.01 sec)
```