

1 根据需求编写SQL语句。

MOBIKE单车租赁App中，用户使用手机号码作为用户名注册一个用户账号，注册时还需要指定一个用户密码，新注册的用户默认账号余额为0，此后充值后，账号余额可以增加。系统中还需要记录所有的单车信息，单车编号，和每小时租金（不足1元按1元算）。用户开启某辆单车后开始计时，骑行结束后，需要记录此次骑行开始和结束时间，并根据车辆每小时租金，从账户余额中扣除此次产生的费用。

根据上述描述，创建系统必须的用户表、单车信息表，以及记录租车信息的表；并插入一定量的模拟数据。

参考答案

一个用户可以租用多次单车，一辆单车又可以被多个用户先后租用，所以用户和单车间属于典型的“多对多关联”，需要建立中间表，才能记录用户对单车的租用信息。数据库和表结构的创建代码如下：

```
01.  SET NAMES UTF8;
02.  DROP DATABASE IF EXISTS mobike;
03.  CREATE DATABASE mobike CHARSET=UTF8;
04.  USE mobike;
05.
06.  /**用户信息表**/
07.  CREATE TABLE mb_user(
08.      uid INT PRIMARY KEY AUTO_INCREMENT,
09.      phone VARCHAR(32),
10.      upwd VARCHAR(32),
11.      balance FLOAT(10, 2)
12.  );
13.  INSERT INTO mb_user VALUES
14.      (10, '13501234567', '111111', 100),
15.      (20, '13501234568', '888888', 50);
16.
17.  /**单车信息表**/
18.  CREATE TABLE mb_bike(
19.      bid INT PRIMARY KEY AUTO_INCREMENT,    #车辆编号
20.      charge FLOAT(4, 2)                      #每小时租金
21.  );
22.  INSERT INTO mb_bike VALUES
23.      (101, 2.0),
24.      (102, 2.0),
```

```

25.     (201, 1.0),
26.     (202, 1.0);
27.
28.     /**车辆租用记录表*/
29.     CREATE TABLE mb_rent(
30.         rid INT PRIMARY KEY AUTO_INCREMENT,
31.         userId INT,                #用户编号
32.         bikeId INT,                #单车编号
33.         startTime BIGINT,          #租车开始时间, 毫秒为单位
34.         endTime BIGINT             #租车结束时间, 毫秒为单位
35.     );
36.     INSERT INTO mb_rent VALUES
37.     (NULL, 10, 2, 1489248609723, 1489368609723),
38.     (NULL, 10, 4, 1489346809723, 1489488809723);

```

2 根据需求编写SQL语句

参考答案

```

01.     SELECT phone, bid, startTime, endTime, ceil((endTime-startTime)/3600000)
02.     FROM mb_user u INNER JOIN mb_bike b INNER JOIN mb_rent r
03.     ON u.uid=r.userId AND b.bid=r.bikeId
04.     WHERE u.uid=10;
05.
06.
07.     #定义存储过程
08.     DELIMITER //
09.     DROP PROCEDURE IF EXISTS checkLogin //
10.     CREATE PROCEDURE checkLogin(
11.         user_phone VARCHAR(32),
12.         user_pwd VARCHAR(32),
13.         OUT result INT
14.     )
15.     BEGIN
16.         DECLARE c INT;
17.         SELECT COUNT(*) INTO c FROM mb_user
18.         WHERE phone=user_phone AND upwd=user_pwd;
19.         IF c=1 THEN
20.             SET result=200;

```

```
21.     ELSE
22.         SET result=400;
23.     END IF;
24. END //
25. DELIMITER ;
26.
27. #测试调用存储过程
28. CALL checkLogin('13501234567', '111111', @num);
29. SELECT @num;
30.
31.
32. #定义存储过程
33. DELIMITER //
34. DROP PROCEDURE IF EXISTS saveRent //
35. CREATE PROCEDURE saveRent (
36.     user_phone VARCHAR(32),
37.     bike_id INT,
38.     start_time BIGINT,
39.     end_time BIGINT,
40.     OUT rent_id INT
41. )
42. BEGIN
43.     DECLARE user_id INT;           #用户编号
44.     DECLARE unit_price FLOAT(4,2); #指定单车的租金单价
45.     DECLARE sum_price FLOAT(10,2); #此次租用的总金额
46.
47.     #根据电话号码查询用户编号
48.     SELECT uid INTO user_id FROM mb_user
49.     WHERE phone=user_phone;
50.
51.     #保存租车记录
52.     INSERT INTO mb_rent VALUES
53.     (NULL, user_id, bike_id, start_time, end_time);
54.     SELECT last_insert_id() INTO rent_id;
55.
56.     #计算此次费用
57.     SELECT charge INTO unit_price
58.     FROM mb_bike
59.     WHERE bid=bike_id;
60.
61.     SET sum_price=CEIL((end_time-start_time)/3600000);
```

```

62.      #修改账户余额，扣除此次费用
63.      UPDATE mb_user SET balance=balance-sum_price
64.      WHERE uid=user_id;
65.  END //
66.  DELIMITER ;
67.
68.  #测试调用存储过程
69.  CALL saveRent('13501234567', 101, 1489248609723, 1489368609723, @rid);
70.  SELECT * FROM mb_rent WHERE rid = @rid;

```

3 使用第三方的mysql模块调用上题中第3小题中的存储过程。

参考答案

由于Node.js官网并未提供对MySQL数据库的支持模块，必须使用NPM工具从www.npmjs.org上下载mysql模块，在命令行中当前项目所在目录下执行：

```
01.  npm i mysql
```

然后调用mysql模块提供 api，访问MySQL数据库：

```

01.  const mysql = require('mysql');
02.
03.  //为了提高创建连接的效率，使用数据库连接池
04.  var pool = mysql.createPool({
05.    host: '127.0.0.1',
06.    user: 'root',
07.    password: '',
08.    database: 'mobike',
09.    port: 3306,
10.    multipleStatements: true,
11.    connectionLimit: 10
12.  });
13.  //从连接池中获取数据库连接对象
14.  pool.getConnection(function (err, conn) {
15.    conn.query(
16.      'CALL saveRent(?, ?, ?, ?, @rid); SELECT @rid AS rid',
17.      ['13501234567', 101, 1489248609723, 1489258609723],

```

```
18.         function (err, result) {
19.             console.log('租车记录编号为: '+result[1][0].rid);
20.             conn.release();
21.         })
22.     });
```