** SOME QUERIES MAY BE MODIFIED BECAUSE OF VARIABLES INPUTS

Book Queries

// Selects all books with filters

1. Select * from book

where title LIKE %title% and author LIKE %author% and publisher LIKE %publisher% and genre LIKE %genre% ;

// Inserts a new book

2. INSERT INTO Book (isbn, title, author, publisher,

genre) Values (isbn , title , author , publisher , genre)

// Updates a book

3. UPDATE Book

Set title= title , author= author ,

publisher= publisher , genre= genre

WHERE isbn = isbn ;

// Gets book based off isbn

4. SELECT * FROM Book

WHERE isbn = isbn ;

// Deletes a book

5. DELETE FROM Book

WHERE isbn= isbn ;

// Gets Genre Count Group By

6. SELECT genre, count(*) AS count FROM Book

GROUP BY genre;

Branch and Room Queries

1.SELECT * FROM LibraryBranch;

2.SELECT * FROM LibraryBranch where branchNum = branchNum;

3.SELECT * FROM Room where branchNum = branchNum;

4.SELECT * FROM Room where branchNum = branchNum and roomnumber = roomnumber;

Employee Queries

1.SELECT eid, eEmail, sin, ename, eaddress, ephonenumber, branchnum, adminStatus FROM employee;

2.INSERT INTO employee(eEmail,SIN,ename,eAddress,ePhoneNumber,branchNum, adminStatus, password) VALUES (email , sin ,

name , address , phoneNum , branch , admin , password);

3.UPDATE employee SET eEmail= email , eAddress= address , ePhoneNumber= phoneNum , password= password

WHERE eid= eid ;

4.INSERT INTO employeeworkedfor(eid, branchNum, fromDate, toDate, fromTime, toTime) VALUES(

eid , branch , DATE_FORMAT(curdate(), %m/%d/%y), present,0:00, present);

5.UPDATE employee SET eEmail= email , eAddress= address , ePhoneNumber= phoneNum , password= password

WHERE eid= eid ;

6.SELECT eid, eEmail, sin, ename, eaddress, ephonenumber, branchnum, adminStatus FROM employee WHERE eid= String(eid) ;

7.SELECT EXISTS(SELECT 1 FROM employee WHERE eEmail= email AND password= password);

8.SELECT eid FROM employee WHERE eEmail= email;

9.SELECT employee.eid, employee.ename, employeeworkedfor.branchNum, fromDate, toDate

FROM employee

INNER JOIN employeeworkedfor ON employee.eid = employeeworkedfor.eid;

10. SELECT employee.eid, employee.ename, employeeworkedfor.branchNum, fromDate, toDate

FROM employee

INNER JOIN employeeworkedfor ON employee.eid =

employeeworkedfor.eid WHERE employee.eid= id ;

Event Queries

1. SELECT * FROM event ORDER BY STR_TO_DATE(fromDate, %m/%d/%Y), STR_TO_DATE(fromTime, %H:%i);

2. SELECT * FROM event WHERE CURDATE() <= STR_TO_DATE(toDate, %m/%d/%Y)

ORDER BY STR_TO_DATE(fromDate, %m/%d/%Y), STR_TO_DATE(fromTime, %H:%i);

3. INSERT INTO event(name,branchNum,fromTime,fromDate,toTime,toDate) VALUES (name , branchNum ,

fromTime , fromDate , toTime , toDate);

4. DELETE FROM event WHERE eventid= eventid ;

Member Queries

1.SELECT accountID,phoneNum,email,name,fines FROM members

2.SELECT accountID,phoneNum,email,name,fines FROM members where members.accountid = id;

3.insert into members (phoneNum,email,name,fines,password) values

(phoneNum , email , name ,0, password);

4. update members set phoneNum = phoneNum

, fines = fines;

, password = password

where accountid = id ;

5. SELECT EXISTS(SELECT 1 FROM members WHERE email= email AND password= password);

Review and Rating Queries

1. SELECT b.isbn,b.title, avg(r.rating) as average FROM

book b, review r

where b.isbn = r.isbn GROUP BY b.isbn order by average DESC;

2.SELECT b.isbn,b.title, avg(r.rating) as average

FROM book b, review r

where b.isbn = r.isbn GROUP BY b.isbn order by average ASC

3.select AVG(rating) AS rating from review where isbn = isbn

4.select b.title, r.rating,r. review, m.name

from review r, book b, members m

where b.isbn = isbn and r.isbn = isbn and r.accountID = m.accountI

5. **insert into review (accountID,isbn,rating,review)**

values accountID isbn rating review ;

Rentals Queries

1.select * from rental where status = 0 and accountID = id;

2.insert into rental
(status,bookid,accountID,fromTime,toTime,fromDate,toDate,returnTime,returnDate)

values (0 bookid accountid fromTime toTime fromDate toDate ,null,null

3.update rental set status = 1, returnTime = returnTime ,returnDate = returnDate where status = 0 and bookid = bookid ;

Schedules Queries

1. SELECT * FROM schedule;

2.INSERT INTO schedules value id , roomName ,

fromTime , toTime , fromDate , toDate ;

3.SELECT * FROM schedules where roomName = roomname ;

Division Query

1.

```
SELECT DISTINCT branchNum
  FROM branchlist AS PS1
  WHERE NOT EXISTS
     (SELECT *
       FROM isbnlist
      WHERE NOT EXISTS
          (SELECT *
            FROM branchlist AS PS2
           WHERE (PS1.branchNum = PS2.branchNum)
             AND (PS2.isbn = isbnlist.isbn)));
```