COMP 2016

PROJECT Report

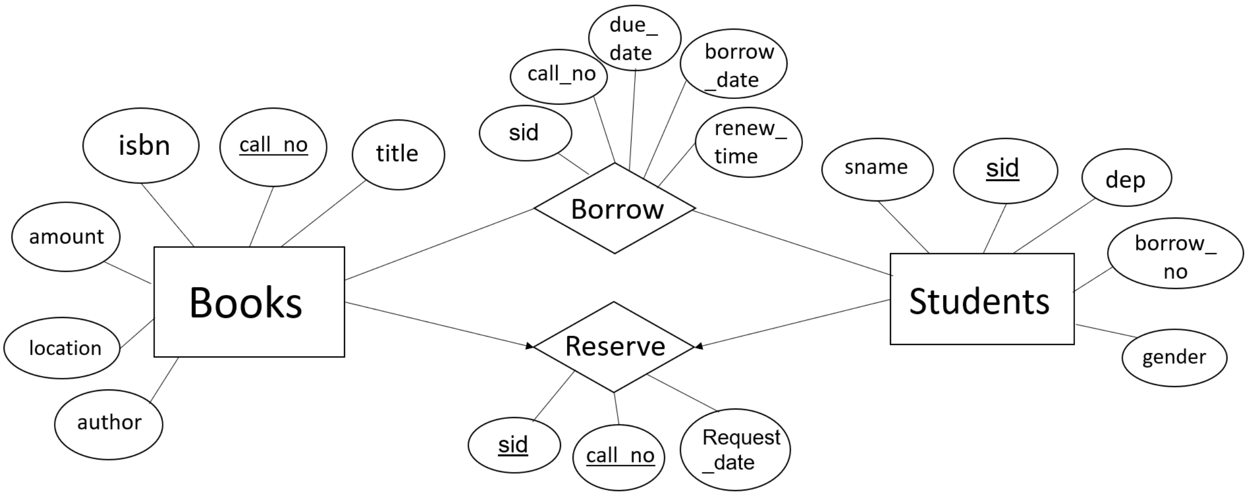
Group 8

Group Members:
Cheung Ling Fung (21210950)
Chim Hoi Yan (20226942)
Chung King To (20223382)
Chung King To (20223382)

Contens

2.ER diagram

a. This is an ER diagram for a university library management system.



Explanation:

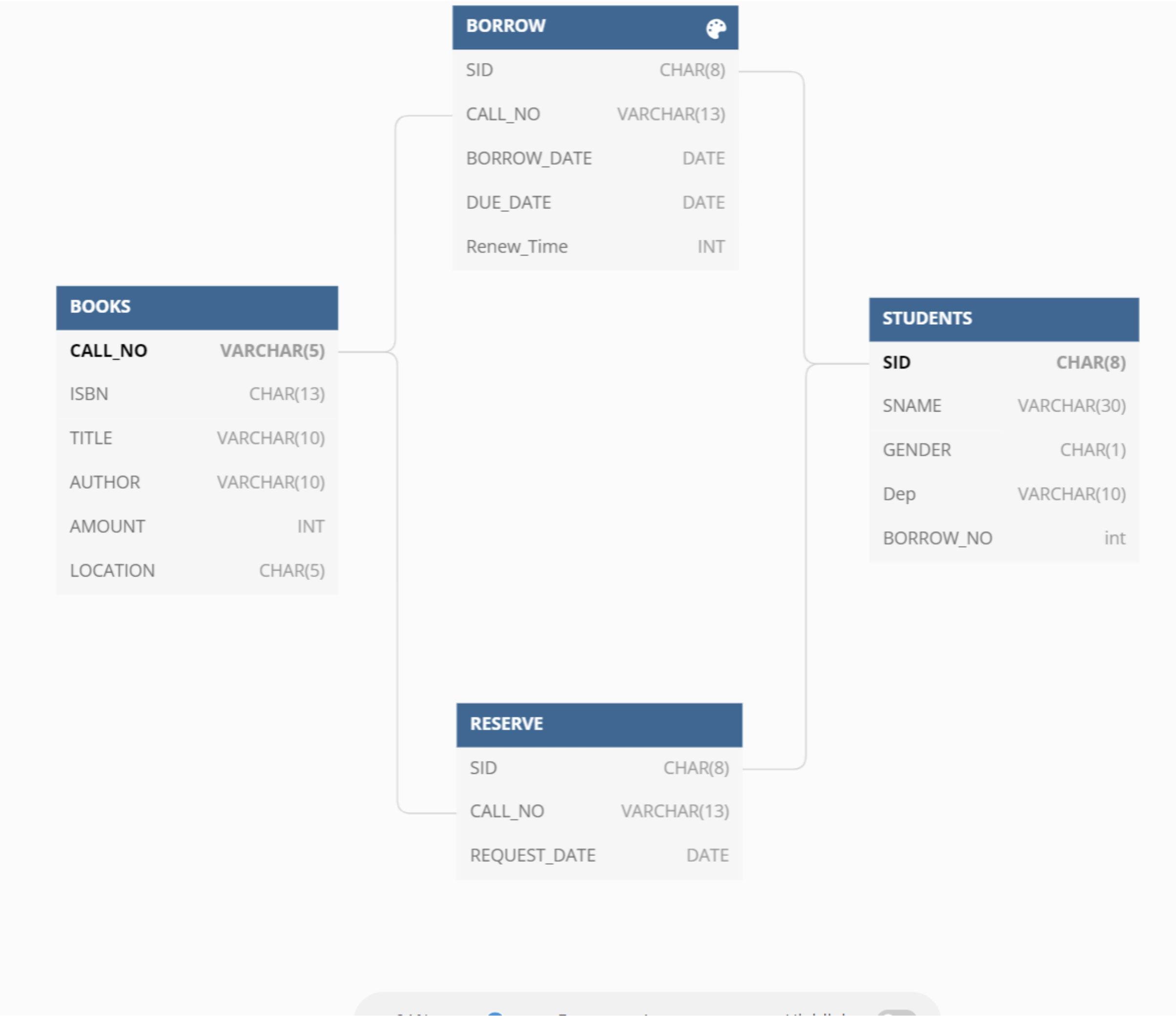
For each student, the database stores the name of the student, his/her student number, major, gender and the number of borrowed books from him/her.
For each book, the database stores its ISBN, call number, title, author, location and amount.

Students can borrow none, one or more than one book. After borrowing successfully, the Borrow Table saves the student number of the borrower, the call number of the borrowed book, the renewtime for the student to renew the book and the borrow and due date of the book.
Also, the amount of books will be decreased by 1 and the borrowed number of students will be increased by 1. The relationship between Students Table and Books Table in borrow is many-to-many.

A student can reserve at most one book only. After reserving successfully, the Reserve Table saves the student number of the reserver, the call number of the borrowed book and the request date of the reserver to the book. The relationship between Students Table and Books Table in reserve is one-to-one.

In return, it deletes the record from borrow, decreases the borrow number of the student by 1 and increases the amount of the book by 1. In renew, it checks the attributes from borrow and reserve for validations and updates the due date of the book in the Borrow Table.

3.Table schemes



BORROW	
SID	CHAR(8)
CALL_NO	VARCHAR(13)
BORROW_DATE	DATE
DUE_DATE	DATE
Renew_Time	INT

BOOKS	
CALL_NO	VARCHAR(5)
ISBN	CHAR(13)
TITLE	VARCHAR(10)
AUTHOR	VARCHAR(10)
AMOUNT	INT
LOCATION	CHAR(5)

STUDENTS	
SID	CHAR(8)
SNAME	VARCHAR(30)
GENDER	CHAR(1)
Dep	VARCHAR(10)
BORROW_NO	int

RESERVE	
SID	CHAR(8)
CALL_NO	VARCHAR(13)
REQUEST_DATE	DATE

4. comment of triggers and SQL commands

BORROW_AMO TRIGGER

```
 create or replace trigger BORROW_AMO
 after insert on BORROW
 For each row
 Begin
 UPDATE BOOKS SET AMOUNT = AMOUNT -1
 where call_no = :New.call_no;
 UPDATE Students SET Borrow_no = Borrow_no +1
 where SID = :NEW.SID;
 END;
/
```

Create a trigger BORROW_AMO before insert on BORROW

Every time a user inserts a new data, the Amount in BOOKS will be updated to Amount+1, which will check CALL_NO.
At the same time, the Borrow_no in BOOKS will be updated to Borrow_no + 1, which will check SID.

INCREASE_BOOK_AMOUT TRIGGER

```
CREATE OR REPLACE TRIGGER increase_book_amount
AFTER DELETE ON BORROW
FOR EACH ROW
DECLARE
cnt INTEGER;
BEGIN
SELECT Amount INTO cnt
FROM BOOKS
WHERE CALL_NO  = :OLD.CALL_NO;
cnt := cnt + 1;
UPDATE BOOKS
SET Amount = cnt
 WHERE CALL_NO = :old.CALL_NO;
END;
/
```

Create a trigger increase_book_amout after delete on BORROW
Temporarily store the Amount data of BOOKS in cnt, which will check CALL_NO

When the data is found, cnt = cnt+1 is updated, and then update the Amount to cnt.This step also check the CALL_NO
At the same time, the Borrow_no in BOOKS will be updated to Borrow_no + 1, which will check SID.

The SQL will first delete all the table and trigger which needs to be use in the project

Drop table

```
 DROP TABLE STUDENTS CASCADE CONSTRAINT;
 DROP TABLE BOOKS CASCADE CONSTRAINT;
 DROP TABLE BORROW CASCADE CONSTRAINT;
 DROP TABLE RESERVE CASCADE CONSTRAINT;
 DROP TRIGGER BORROW_AMO;
 DROP TRIGGER increase_book_amount;
```

Create Table Books

```
CREATE TABLE BOOKS(
 CALL_NO VARCHAR(5),
 ISBN    CHAR(13) NOT NULL,
 TITLE   VARCHAR(10),
 AUTHOR VARCHAR(10),
 AMOUNT INT,
 LOCATION    CHAR(5),
 PRIMARY KEY(CALL_NO));
```

Create Table Students

```
 CREATE TABLE STUDENTS(
 SID     CHAR(8) NOT NULL,
 SNAME VARCHAR(30),
 GENDER  CHAR(1),
 Dep VARCHAR(10),
 BORROW_NO int,
 PRIMARY KEY(SID));
```

Create Table Borrow

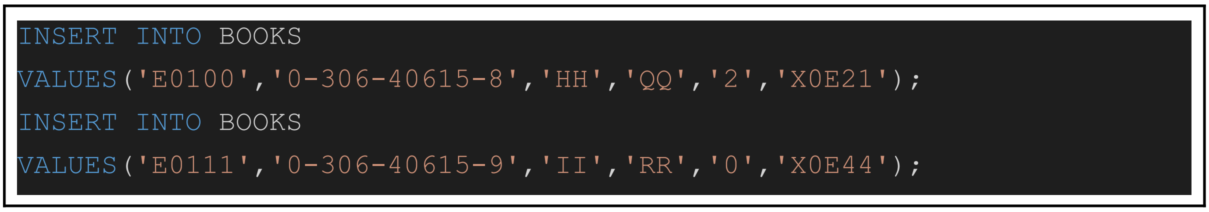
```
CREATE TABLE BORROW(
 SID CHAR(8) NOT NULL,
 CALL_NO VARCHAR(13),
 BORROW_DATE     DATE,
 DUE_DATE    DATE,
 Renew_Time INT,
 PRIMARY KEY (SID, CALL_NO),
 FOREIGN KEY (SID) REFERENCES STUDENTS(SID),
 FOREIGN KEY (CALL_NO) REFERENCES BOOKS(CALL_NO));
```

Create Table Reserve

```
 CREATE TABLE RESERVE
 (SID CHAR(8) NOT NULL,
 CALL_NO VARCHAR(13),
 REQUEST_DATE     DATE,
 PRIMARY KEY (SID, CALL_NO),
 FOREIGN KEY (SID) REFERENCES STUDENTS(SID),
 FOREIGN KEY (CALL_NO) REFERENCES BOOKS(CALL_NO));
```

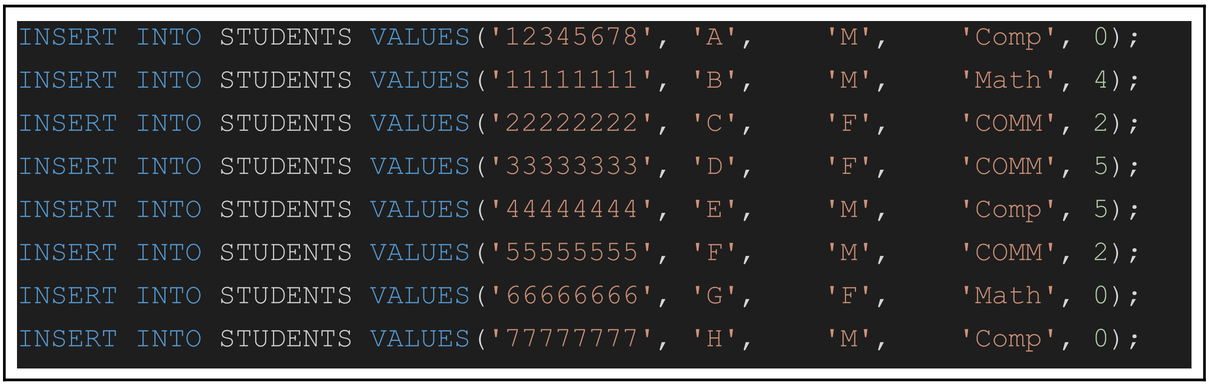
Insert test data of Books

```
INSERT INTO BOOKS
VALUES('A0000','0-306-40615-1','AA','XX','0','S1E01');
INSERT INTO BOOKS
VALUES('B0000','0-306-40615-2','BB','YY','0','S2E02');
INSERT INTO BOOKS
VALUES('C1111','0-306-40615-3','CC','ZZ','2','D1E11');
INSERT INTO BOOKS
VALUES('B0001','0-306-40615-4','DD','UU','2','G1E00');
INSERT INTO BOOKS
VALUES('A1111','0-306-40615-5','EE','VV','2','B1E00');
INSERT INTO BOOKS
VALUES('D0101','0-306-40615-6','FF','WW','1','B2E11');
INSERT INTO BOOKS
VALUES('E0000','0-306-40615-7','GG','PP','0','X0E22');
```



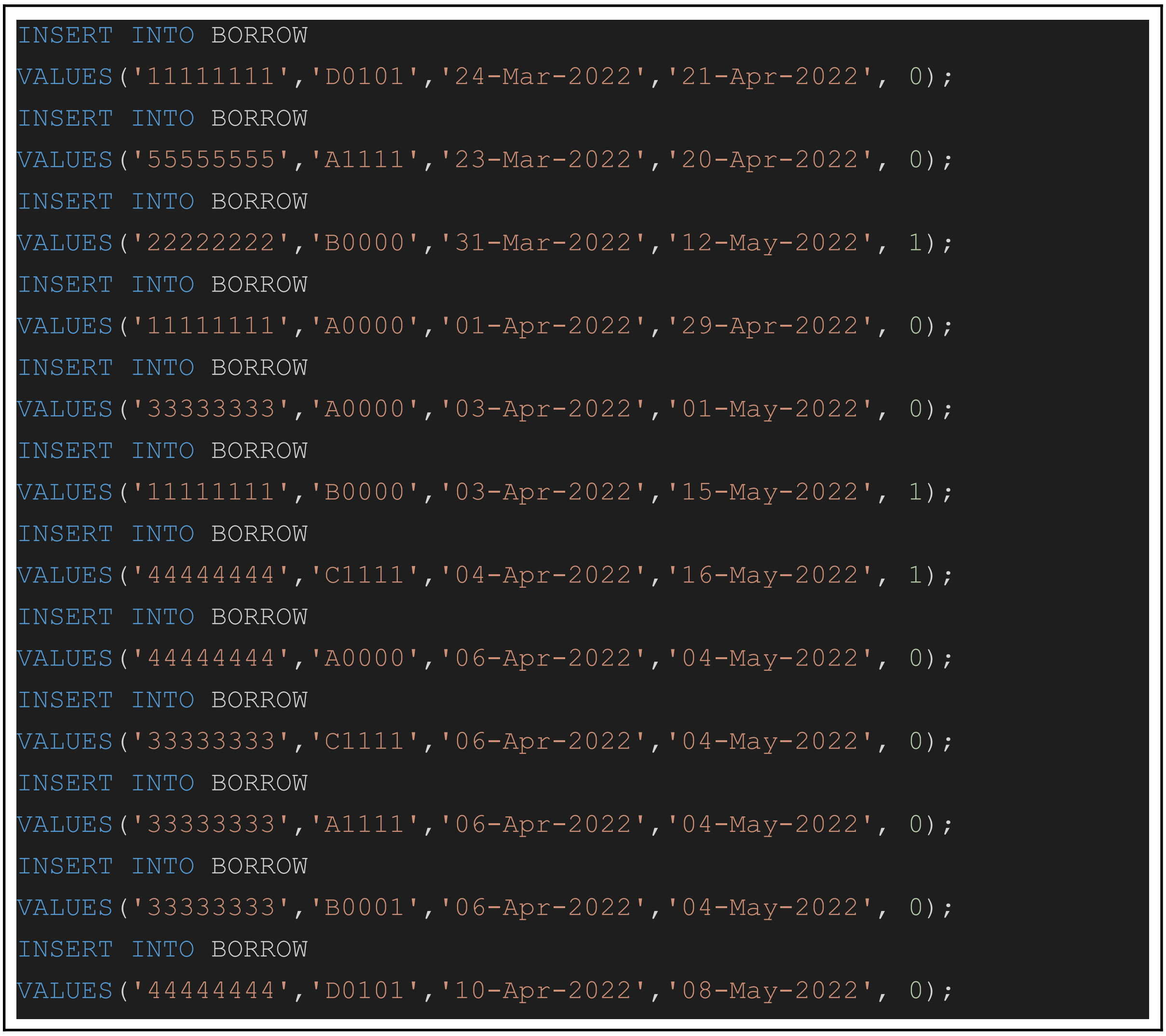
```
INSERT INTO BOOKS
VALUES('E0100','0-306-40615-8','HH','QQ','2','X0E21');
INSERT INTO BOOKS
VALUES('E0111','0-306-40615-9','II','RR','0','X0E44');
```

Insert test data for Students

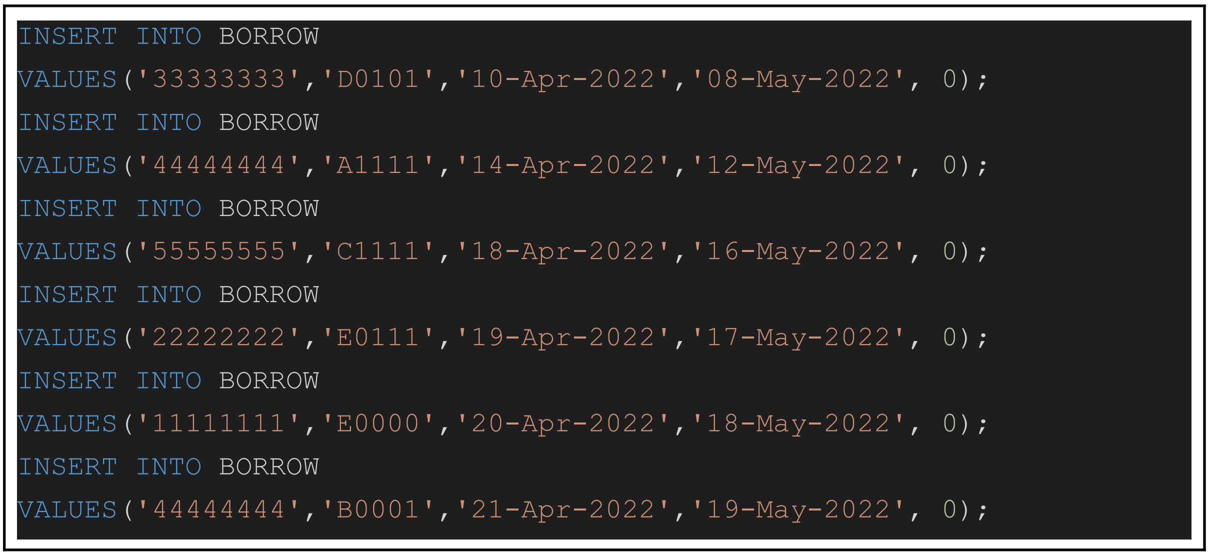


```
INSERT INTO STUDENTS VALUES('12345678', 'A',     'M',    'Comp', 0);
INSERT INTO STUDENTS VALUES('11111111', 'B',     'M',    'Math', 4);
INSERT INTO STUDENTS VALUES('22222222', 'C',     'F',    'COMM', 2);
INSERT INTO STUDENTS VALUES('33333333', 'D',     'F',    'COMM', 5);
INSERT INTO STUDENTS VALUES('44444444', 'E',     'M',    'Comp', 5);
INSERT INTO STUDENTS VALUES('55555555', 'F',     'M',    'COMM', 2);
INSERT INTO STUDENTS VALUES('66666666', 'G',     'F',    'Math', 0);
INSERT INTO STUDENTS VALUES('77777777', 'H',     'M',    'Comp', 0);
```

Insert test data for Borrow

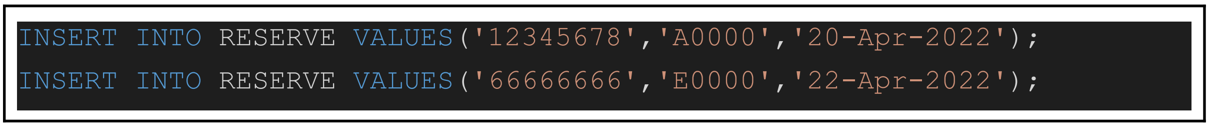


```
INSERT INTO BORROW
VALUES('11111111','D0101','24-Mar-2022','21-Apr-2022', 0);
INSERT INTO BORROW
VALUES('55555555','A1111','23-Mar-2022','20-Apr-2022', 0);
INSERT INTO BORROW
VALUES('22222222','B0000','31-Mar-2022','12-May-2022', 1);
INSERT INTO BORROW
VALUES('11111111','A0000','01-Apr-2022','29-Apr-2022', 0);
INSERT INTO BORROW
VALUES('33333333','A0000','03-Apr-2022','01-May-2022', 0);
INSERT INTO BORROW
VALUES('11111111','B0000','03-Apr-2022','15-May-2022', 1);
INSERT INTO BORROW
VALUES('44444444','C1111','04-Apr-2022','16-May-2022', 1);
INSERT INTO BORROW
VALUES('44444444','A0000','06-Apr-2022','04-May-2022', 0);
INSERT INTO BORROW
VALUES('33333333','C1111','06-Apr-2022','04-May-2022', 0);
INSERT INTO BORROW
VALUES('33333333','A1111','06-Apr-2022','04-May-2022', 0);
INSERT INTO BORROW
VALUES('33333333','B0001','06-Apr-2022','04-May-2022', 0);
INSERT INTO BORROW
VALUES('44444444','D0101','10-Apr-2022','08-May-2022', 0);
```



```
INSERT INTO BORROW
VALUES('33333333','D0101','10-Apr-2022','08-May-2022', 0);
INSERT INTO BORROW
VALUES('44444444','A1111','14-Apr-2022','12-May-2022', 0);
INSERT INTO BORROW
VALUES('55555555','C1111','18-Apr-2022','16-May-2022', 0);
INSERT INTO BORROW
VALUES('22222222','E0111','19-Apr-2022','17-May-2022', 0);
INSERT INTO BORROW
VALUES('11111111','E0000','20-Apr-2022','18-May-2022', 0);
INSERT INTO BORROW
VALUES('44444444','B0001','21-Apr-2022','19-May-2022', 0);
```

Insert test data for RESERVE:



```
INSERT INTO RESERVE VALUES('12345678','A0000','20-Apr-2022');
INSERT INTO RESERVE VALUES('66666666','E0000','22-Apr-2022');
```