INFO 605: Database Management Systems

Instructor: Prof. Stephen Frein

Spring 2022

Library Database Management System

Group Members:

Aditi Salunkhe

Natasha Lalwani

06/05/2022

TABLE OF CONTENTS

Project Description

The Library Management System is a database that stores information about the books in the library, their authors, library users to whom books are provided, library staff, and so on. Manually organising this is quite difficult and maintaining all of this data is a difficult task. The organisation of any library has gotten considerably easier owing to technological advancements.

The objective of this project is to provide a generalised database management system that can be implemented by any library. Our proposed Library Management System is created with the intention to ease organisation of information. It minimises management's workload by reducing the amount of manual work required.

We first created the ERD (Entity Relational Diagram), following which we translated that into a relational model. That was then implemented using SQL via Oracle using Data Definition Language (DDL) and Data Manipulation Language (DML).

Requirements

Our proposed library database management system aims to diminish the burden on managing members, books, equipment, etc. effectively. This data set is also optimised to store and retrieve information about members(student/ staff), various books, late payments, booked rooms, etc. For the convenience of any reviewer, the vital aspects of the design requirement are listed below.

Member: This is the super class that consists of attributes; memberID (this is the primary key), firstName, lastName, address, dateOfBirth, numberOfBooks. The subclasses student and staff inherit from the

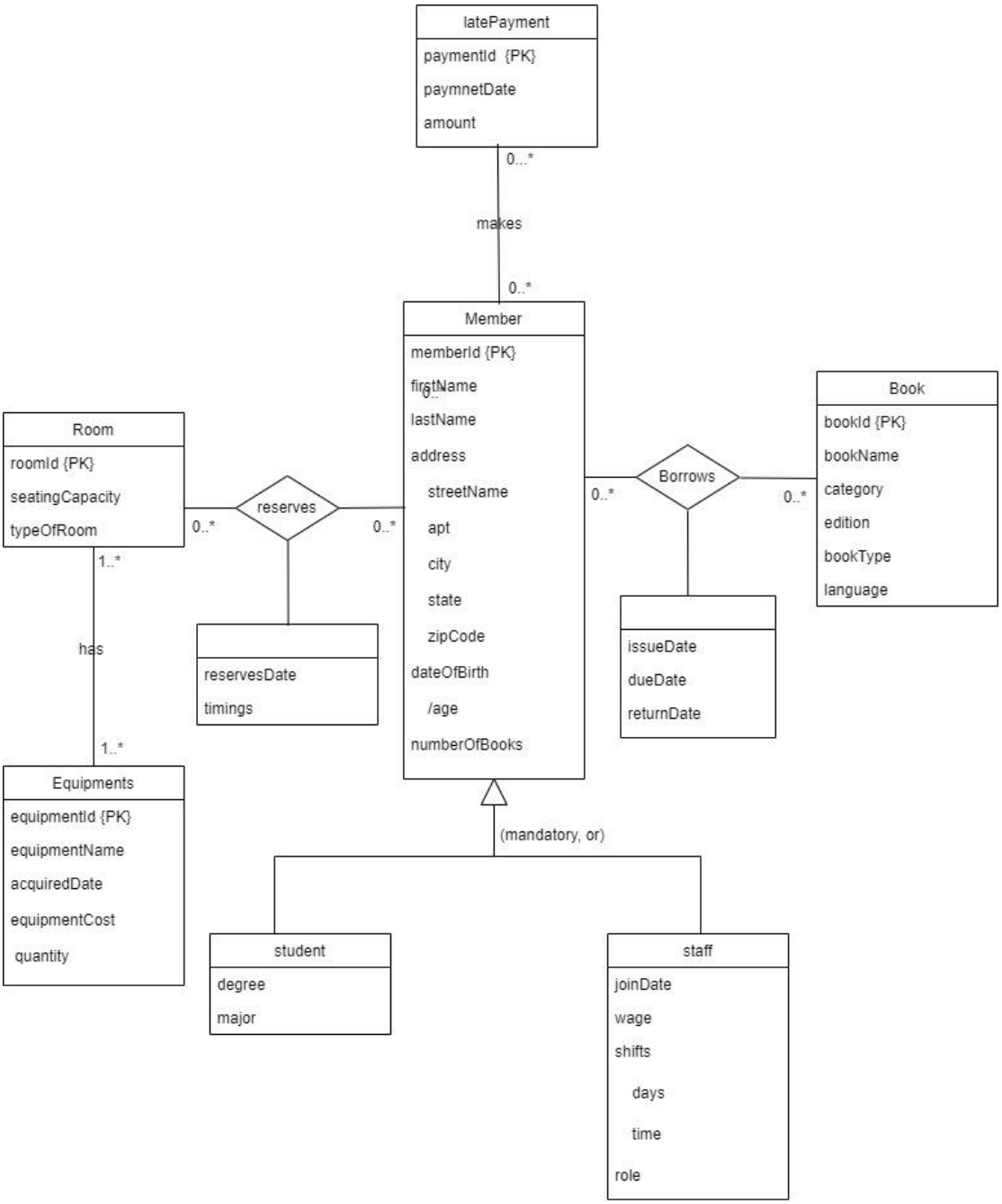
Member class. Multiple members can borrow multiple books, book multiple rooms and make multiple late payments.

Book: It consists of attributes like bookId (this is the primary key), bookName, category, edition, bookType and language. We track the issueDate, dueDate and returnDate for each book. It can be borrowed by multiple members.

Room: It consists of attributes like roomId (this is the primary key), seatingCapacity and typeOfRoom. We also keep track of timings of booking a room and the reserveDate. It can be reserved by multiple members.

Equipments: It consists of attributes like equipmentId (this is the primary key), equipmentName, acquiredDate, equipmentCost and quantity. A room has at least 1. It can also have multiple equipments.

Entity Relationship Diagram



Database Schema

Room (roomId, seatingCapacity, typeOfRoom)

Equipment (equipmentId, equipmentName, acquireDate, equipmentCost, quantity)

Has (room, equipment)

Reserves (roomId, memberId, reservesDate, timings)

Member (memberId, firstName, lastName, addressStreetName, addressApt, addressState, addressZipCode, dateOfBirth, numberOfBooks)

Student (memberId, degree, major)

Staff (memberId, joinDate, wage, shiftDay, shiftTime, role)

Book (bookId, bookName, category, edition, bookType, language)

Borrow (memberId, bookId, issueDate, dueDate, returnDate)

LatePayment (paymentId, paymentDate, amount)

Makes (memberId, paymentId)

Data Dictionary

Please see the excel file name “Data Dictionary” in the zip folder to see the complete data dictionary.

Data Definition Language (DDL) Commands

```
CREATE TABLE MEMBER
(
    memberId VARCHAR2(10) CONSTRAINT member_pk PRIMARY KEY,
    firstName VARCHAR(20) CONSTRAINT member_fname NOT NULL,
    lastName VARCHAR(20) CONSTRAINT member_lname NOT NULL,
    address VARCHAR(50),
    dateOfBirth DATE,
    numberOfBooks NUMBER(1)
);

CREATE TABLE STUDENT
(
    memberId VARCHAR2(10) CONSTRAINT member_fk REFERENCES MEMBER(memberId),
    degree CHAR(20),
    major VARCHAR(20)
);

CREATE TABLE STAFF
(
    memberID VARCHAR2(10) CONSTRAINT member_fk_staff REFERENCES MEMBER(memberId),
    wage NUMBER(7,2),
    shifts VARCHAR2(10) CONSTRAINT staff_shifts NOT NULL,
    role VARCHAR2(10) CONSTRAINT staff_role NOT NULL
);

CREATE TABLE ROOM
(
    roomId VARCHAR2(10) CONSTRAINT room_pk PRIMARY KEY,
    seatingCapacity NUMBER(2),
    typeOfRoom VARCHAR2(10) CONSTRAINT room_type NOT NULL
);

```

```
32
33  CREATE TABLE RESERVES
34  (
35      roomId VARCHAR2(10) CONSTRAINT room_fk_reserves REFERENCES ROOM(roomId),
36      memberID VARCHAR2(10) CONSTRAINT member_fk_reserves REFERENCES MEMBER(memberId),
37      reservesDate DATE CONSTRAINT reserves_date NOT NULL,
38      timings VARCHAR2(10) CONSTRAINT reserves_time NOT NULL
39  );
40
41  CREATE TABLE BOOK
42  (
43      bookId VARCHAR2(10) CONSTRAINT book_pk PRIMARY KEY,
44      bookName VARCHAR2(20) CONSTRAINT book_name NOT NULL,
45      category VARCHAR2(20),
46      edition VARCHAR2(20),
47      bookType VARCHAR2(10),
48      language VARCHAR2(10)
49  );
50
51  CREATE TABLE BORROW
52  (
53      memberId VARCHAR2(10) CONSTRAINT member_fk_borrow REFERENCES MEMBER(memberId),
54      bookId VARCHAR2(10) CONSTRAINT book_fk_borrow REFERENCES BOOK(bookId),
55      issueDate DATE CONSTRAINT borrow_issue NOT NULL,
56      dueDate DATE CONSTRAINT borrow_due NOT NULL,
57      returnDate DATE CONSTRAINT borrow_return NOT NULL
58  );
59
60  CREATE TABLE LATE_PAYMENT
61  (
62      paymentId VARCHAR2(10) CONSTRAINT payment_pk PRIMARY KEY,
63      paymentDate DATE CONSTRAINT payment_date NOT NULL,
64      amount NUMBER(5,2) CONSTRAINT payment_amount NOT NULL
65  );
```

```
66
67  CREATE TABLE MAKES
68  (
69      paymentId VARCHAR2(10) CONSTRAINT payment_fk_makes REFERENCES LATE_PAYMENT(paymentId),
70      memberId VARCHAR2(10) CONSTRAINT member_fk_makes REFERENCES MEMBER(memberId)
71  );
72
73  CREATE TABLE EQUIPMENT
74  (
75      equipmentId VARCHAR2(10) CONSTRAINT equipment_pk PRIMARY KEY,
76      equipmentName VARCHAR2(20) CONSTRAINT equipment_name NOT NULL,
77      acquireDate DATE,
78      equipmentCost NUMBER(7,2),
79      quantity NUMBER(2)
80  );
81
82  CREATE TABLE HAS
83  (
84      roomId VARCHAR2(10) CONSTRAINT room_fk_has REFERENCES ROOM(roomId),
85      equipmentId VARCHAR2(10) CONSTRAINT equipment_fk_has REFERENCES EQUIPMENT(equipmentId)
86  );
```

```
CREATE TABLE MEMBER
(
   memberId VARCHAR2(10) CONSTRAINT member_pk PRIMARY KEY,
   firstName VARCHAR(20) CONSTRAINT member_fname NOT NULL,
   lastName VARCHAR(20) CONSTRAINT member_lname NOT NULL,
   address VARCHAR(50),
   dateOfBirth DATE,
   numberOfBooks NUMBER(1)
);
```

```
CREATE TABLE STUDENT
(
   memberId VARCHAR2(10) CONSTRAINT member_fk REFERENCES
MEMBER(memberId),
   degree CHAR(20),
   major VARCHAR(20)
);
```

```
CREATE TABLE STAFF
(
   memberID VARCHAR2(10) CONSTRAINT member_fk_staff REFERENCES
MEMBER(memberId),
   wage NUMBER(7,2),
   shifts VARCHAR2(10) CONSTRAINT staff_shifts NOT NULL,
   role VARCHAR2(10) CONSTRAINT staff_role NOT NULL
);
```

```
CREATE TABLE ROOM
(
   roomId VARCHAR2(10) CONSTRAINT room_pk PRIMARY KEY,
   seatingCapacity NUMBER(2),
   typeOfRoom VARCHAR2(10) CONSTRAINT room_type NOT NULL
);
```

```
CREATE TABLE RESERVES
(
   roomId VARCHAR2(10) CONSTRAINT room_fk_reserves REFERENCES
ROOM(roomId),
   memberID VARCHAR2(10) CONSTRAINT member_fk_reserves REFERENCES
MEMBER(memberId),
   reservesDate DATE CONSTRAINT reserves_date NOT NULL,
   timings VARCHAR2(10) CONSTRAINT reserve_time NOT NULL
);
```

```
CREATE TABLE BOOK
(
   bookId VARCHAR2(10) CONSTRAINT book_pk PRIMARY KEY,
   bookName VARCHAR2(20) CONSTRAINT book_name NOT NULL,
   category VARCHAR2(20),
   edition VARCHAR2(20),
   bookType VARCHAR2(10),
   language VARCHAR2(10)
);
```

```
CREATE TABLE BORROW
(
   memberId VARCHAR2(10) CONSTRAINT member_fk_borrow REFERENCES
MEMBER(memberId),
   bookId VARCHAR2(10) CONSTRAINT book_fk_borrow REFERENCES
BOOK(bookId),
   issueDate DATE CONSTRAINT borrow_issue NOT NULL,
   dueDate DATE CONSTRAINT borrow_due NOT NULL,
   returnDate DATE CONSTRAINT borrow_return NOT NULL
);
```

```
CREATE TABLE LATE_PAYMENT
(
   paymentId VARCHAR2(10) CONSTRAINT payment_pk PRIMARY KEY,
   paymentDate DATE CONSTRAINT payment_date NOT NULL,
   amount NUMBER(5,2) CONSTRAINT payment_amount NOT NULL
);
```

```
CREATE TABLE MAKES
(
   paymentId VARCHAR2(10) CONSTRAINT payment_fk_makes REFERENCES
LATE_PAYMENT(paymentId),
   memberId VARCHAR2(10) CONSTRAINT member_fk_makes REFERENCES
MEMBER(memberId)
);
```

```
CREATE TABLE EQUIPMENT
(
   equipmentId VARCHAR2(10) CONSTRAINT equipment_pk PRIMARY KEY,
   equipmentName VARCHAR2(20) CONSTRAINT equipment_name NOT NULL,
   acquireDate DATE,
   equipmentCost NUMBER(7,2),
   quantity NUMBER(2)
);
```

```
CREATE TABLE HAS
(
   roomId VARCHAR2(10) CONSTRAINT room_fk_has REFERENCES
ROOM(roomId),
   equipmentId VARCHAR2(10) CONSTRAINT equipment_fk_has REFERENCES
EQUIPMENT(equipmentId)
);
```

Data Manipulation Language (DML)

```
87
88  INSERT INTO MEMBER VALUES ('14525049', 'Aditi', 'Salunkhe', '500 N, 39th Street, Philadelphia, PA 19104', TO_DATE('1997/11/24', 'yyyy/mm/dd'), 3);
89  INSERT INTO MEMBER VALUES('14463926', 'Natasha', 'Lalwani', '3414 Baring St, Apt 2F,  Philadelphia, PA 19104', TO_DATE('1998/06/09' , 'YYYY/MM/DD'), 3);
90  INSERT INTO MEMBER VALUES('14476512', 'Janam', 'Patel', '4020 Baring St, Apt 210,  Philadelphia, PA 19104', TO_DATE('1999/01/18' , 'YYYY/MM/DD'), 5);
91  INSERT INTO MEMBER VALUES('14498734', 'Suyog', 'Narkhede', '3836 Haverford, Apt 134,  Philadelphia, PA 19104', TO_DATE('1997/10/26' , 'YYYY/MM/DD'), 5);
92  INSERT INTO MEMBER VALUES('14401267', 'Oshin', 'Kushwaha', '314 N Powelton, Apt 2A,  Philadelphia, PA 19104', TO_DATE('1995/06/14' , 'YYYY/MM/DD'), 3);
93  INSERT INTO MEMBER VALUES('14424681', 'Atharv', 'Joshi', '3734 Brandywine, Philadelphia, PA 19104', TO_DATE('1997/12/23', 'YYYY/MM/DD'), 5);
94
95  SELECT * FROM MEMBER;
```

Script Output | Query Result

SQL | All Rows Fetched: 6 in 0.019 seconds

	MEMBERID	FIRSTNAME	LASTNAME	ADDRESS	DATEOFBIRTH	NUMBEROFBOOKS
1	14525049	Aditi	Salunkhe	500 N, 39th Street, Philadelphia, PA 19104	24-11-97	3
2	14463926	Natasha	Lalwani	3414 Baring St, Apt 2F, Philadelphia, PA 19104	09-06-98	3
3	14476512	Janam	Patel	4020 Baring St, Apt 210, Philadelphia, PA 19104	18-01-99	5
4	14498734	Suyog	Narkhede	3836 Haverford, Apt 134, Philadelphia, PA 19104	26-10-97	5
5	14401267	Oshin	Kushwaha	314 N Powelton, Apt 2A, Philadelphia, PA 19104	14-06-95	3
6	14424681	Atharv	Joshi	3734 Brandywine, Philadelphia, PA 19104	23-12-97	5

INSERT INTO MEMBER VALUES ('14525049', 'Aditi', 'Salunkhe', '500 N, 39th Street, Philadelphia, PA 19104', TO_DATE('1997/11/24', 'yyyy/mm/dd'), 3);

INSERT INTO MEMBER VALUES('14463926', 'Natasha', 'Lalwani', '3414 Baring St, Apt 2F, Philadelphia, PA 19104', TO_DATE('1998/06/09' , 'YYYY/MM/DD'), 3);

INSERT INTO MEMBER VALUES('14476512', 'Janam', 'Patel', '4020 Baring St, Apt 210, Philadelphia, PA 19104', TO_DATE('1999/01/18' , 'YYYY/MM/DD'), 5);

INSERT INTO MEMBER VALUES('14498734', 'Suyog', 'Narkhede', '3836 Haverford, Apt 134, Philadelphia, PA 19104', TO_DATE('1997/10/26' , 'YYYY/MM/DD'), 5);

INSERT INTO MEMBER VALUES('14401267', 'Oshin', 'Kushwaha', '314 N Powelton, Apt 2A, Philadelphia, PA 19104', TO_DATE('1995/06/14' , 'YYYY/MM/DD'), 3);

INSERT INTO MEMBER VALUES('14424681', 'Atharv', 'Joshi', '3734 Brandywine, Philadelphia, PA 19104', TO_DATE('1997/12/23', 'YYYY/MM/DD'), 5);

```
96
97  INSERT INTO STUDENT VALUES('14525049', 'Masters', 'Information System');
98  INSERT INTO STUDENT VALUES('14463926', 'Masters', 'AIML');
99  INSERT INTO STUDENT VALUES('14401267', 'Bacherlors', 'Business Analytics');
100
101 SELECT * FROM STUDENT;
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 3 in 0.01 seconds

	MEMBERID	DEGREE	MAJOR
1	14525049	Masters	Information System
2	14463926	Masters	AIML
3	14401267	Bachelors	Business Analytics

```
INSERT INTO STUDENT VALUES('14525049', 'Masters', 'Information System');
INSERT INTO STUDENT VALUES('14463926', 'Masters', 'AIML');
INSERT INTO STUDENT VALUES('14401267', 'Bachelors', 'Business Analytics');
```

```
102
103 INSERT INTO STAFF VALUES('14476512', 3450.67, 'Morning', 'Librarian');
104 INSERT INTO STAFF VALUES('14498734', 1200.95, 'Afternoon', 'Assistant');
105 INSERT INTO STAFF VALUES('14424681', 2400.75, 'Evening', 'Technician');
106
107 SELECT * FROM STAFF;
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 3 in 0.868 seconds

	MEMBERID	WAGE	SHIFTS	ROLE
1	14476512	3450.67	Morning	Librarian
2	14498734	1200.95	Afternoon	Assistant
3	14424681	2400.75	Evening	Technician

```
INSERT INTO STAFF VALUES('14476512', 3450.67, 'Morning', 'Librarian');
INSERT INTO STAFF VALUES('14498734', 1200.95, 'Afternoon', 'Assistant');
INSERT INTO STAFF VALUES('14424681', 2400.75, 'Evening', 'Technician');
```

```
108
109 INSERT INTO ROOM VALUES('2468', 10, 'Conference');
110 INSERT INTO ROOM VALUES('1234', 15, 'Conference');
111 INSERT INTO ROOM VALUES('5678', 1, 'Work Pod');
112 INSERT INTO ROOM VALUES('1213', 50, 'Seminar');
113 INSERT INTO ROOM VALUES('1415', 25, 'Lecture');
114 INSERT INTO ROOM VALUES('9101', 2, 'Work Pod');
115
116 SELECT * FROM ROOM;
```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 6 in 0.011 seconds

	ROOMID	SEATINGCAPACITY	TYPEOFROOM
1	2468	10	Conference
2	1234	15	Conference
3	5678	1	Work Pod
4	1213	50	Seminar
5	1415	25	Lecture
6	9101	2	Work Pod

```
INSERT INTO ROOM VALUES('2468', 10, 'Conference');
INSERT INTO ROOM VALUES('1234', 15, 'Conference');
INSERT INTO ROOM VALUES('5678', 1, 'Work Pod');
INSERT INTO ROOM VALUES('1213', 50, 'Seminar');
INSERT INTO ROOM VALUES('1415', 25, 'Lecture');
INSERT INTO ROOM VALUES('9101', 2, 'Work Pod');
```

```
117
118 INSERT INTO RESERVES VALUES('2468', '14525049', TO_DATE('2022/01/15', 'YYYY/MM/DD'), '10AM-12PM');
119 INSERT INTO RESERVES VALUES('5678', '14476512', TO_DATE('2022/05/02', 'YYYY/MM/DD'), '12PM-4PM');
120 INSERT INTO RESERVES VALUES('1213', '14401267', TO_DATE('2022/06/08', 'YYYY/MM/DD'), '5PM-6PM');
121
122 SELECT * FROM RESERVES;
```

Script Output | Query Result

SQL | All Rows Fetched: 3 in 0.015 seconds

	ROOMID	MEMBERID	RESERVESDATE	TIMINGS
1	2468	14525049	15-01-22	10AM-12PM
2	5678	14476512	02-05-22	12PM-4PM
3	1213	14401267	08-06-22	5PM-6PM

```
INSERT INTO RESERVES VALUES('2468', '14525049', TO_DATE('2022/01/15',
'YYYY/MM/DD'), '10AM-12PM');

INSERT INTO RESERVES VALUES('5678', '14476512', TO_DATE('2022/05/02',
'YYYY/MM/DD'), '12PM-4PM');

INSERT INTO RESERVES VALUES('1213', '14401267', TO_DATE('2022/06/08',
```

'YYYY/MM/DD'), '5PM-6PM');

```
123
124  INSERT INTO BOOK VALUES('123', 'Gone Girl', 'Thriller','3', 'Audiobook', 'English');
125  INSERT INTO BOOK VALUES('456', 'Forbes', 'Business', '1', 'Magazine', 'English');
126  INSERT INTO BOOK VALUES('789', 'Don Quixote', 'Parody', '2', 'Paperback','Spanish');
127  INSERT INTO BOOK VALUES('001', 'Nirmala', 'Drama','2' , 'Hardcover','Hindi');
128  INSERT INTO BOOK VALUES('333', 'Divergent', 'Science Fiction', '3', 'Audiobook','English');
129  INSERT INTO BOOK VALUES('646', 'Mala', 'Romantic', '1', 'Paperback','Urdu');
130
131  SELECT * FROM BOOK;
```

Script Output × Query Result 1 ×

SQL | All Rows Fetched: 6 in 0.013 seconds

	BOOKID	BOOKNAME	CATEGORY	EDITION	BOOKTYPE	LANGUAGE
1	123	Gone Girl	Thriller	3	Audiobook	English
2	456	Forbes	Business	1	Magazine	English
3	789	Don Quixote	Parody	2	Paperback	Spanish
4	001	Nirmala	Drama	2	Hardcover	Hindi
5	333	Divergent	Science Fiction	3	Audiobook	English
6	646	Mala	Romantic	1	Paperback	Urdu

INSERT INTO BOOK VALUES('123', 'Gone Girl', 'Thriller','3', 'Audiobook', 'English');
INSERT INTO BOOK VALUES('456', 'Forbes', 'Business', '1', 'Magazine', 'English');
INSERT INTO BOOK VALUES('789', 'Don Quixote', 'Parody', '2', 'Paperback','Spanish');
INSERT INTO BOOK VALUES('001', 'Nirmala', 'Drama','2' , 'Hardcover','Hindi');
INSERT INTO BOOK VALUES('333', 'Divergent', 'Science Fiction', '3', 'Audiobook','English');
INSERT INTO BOOK VALUES('646', 'Mala', 'Romantic', '1', 'Paperback','Urdu');

```
132
133  INSERT INTO BORROW VALUES('14525049', '456', TO_DATE('2022/06/03', 'YYYY/MM/DD'), TO_DATE('2022/06/17', 'YYYY/MM/DD'),  TO_DATE('2022/06/21', 'YYYY/MM/DD'));
134  INSERT INTO BORROW VALUES('14463926', '001', TO_DATE('2022/04/25', 'YYYY/MM/DD'), TO_DATE('2022/05/16', 'YYYY/MM/DD'),  TO_DATE('2022/05/13', 'YYYY/MM/DD'));
135  INSERT INTO BORROW VALUES('14424681', '333', TO_DATE('2021/10/26', 'YYYY/MM/DD'), TO_DATE('2021/11/15', 'YYYY/MM/DD'),  TO_DATE('2022/11/30', 'YYYY/MM/DD'));
136  INSERT INTO BORROW VALUES('14498734', '123', TO_DATE('2020/01/13', 'YYYY/MM/DD'), TO_DATE('2020/02/07', 'YYYY/MM/DD'),  TO_DATE('2020/02/05', 'YYYY/MM/DD'));
137  INSERT INTO BORROW VALUES('14476512', '646', TO_DATE('2019/03/09', 'YYYY/MM/DD'), TO_DATE('2019/03/30', 'YYYY/MM/DD'),  TO_DATE('2019/04/02', 'YYYY/MM/DD'));
138
139  SELECT * FROM BORROW;
```

Script Output × Query Result ×

SQL | All Rows Fetched: 5 in 0.013 seconds

	MEMBERID	BOOKID	ISSUEDATE	DUEDATE	RETURNDATE
1	14525049	456	03-06-22	17-06-22	21-06-22
2	14463926	001	25-04-22	16-05-22	13-05-22
3	14424681	333	26-10-21	15-11-21	30-11-22
4	14498734	123	13-01-20	07-02-20	05-02-20
5	14476512	646	09-03-19	30-03-19	02-04-19

INSERT INTO BORROW VALUES('14525049', '456', TO_DATE('2022/06/03', 'YYYY/MM/DD'), TO_DATE('2022/06/17', 'YYYY/MM/DD'), TO_DATE('2022/06/21', 'YYYY/MM/DD'));

INSERT INTO BORROW VALUES('14463926', '001', TO_DATE('2022/04/25',

'YYYY/MM/DD'), TO_DATE('2022/05/16', 'YYYY/MM/DD'), TO_DATE('2022/05/13', 'YYYY/MM/DD'));

INSERT INTO BORROW VALUES('14424681', '333', TO_DATE('2021/10/26', 'YYYY/MM/DD'), TO_DATE('2021/11/15', 'YYYY/MM/DD'), TO_DATE('2022/11/30', 'YYYY/MM/DD'));

INSERT INTO BORROW VALUES('14498734', '123', TO_DATE('2020/01/13', 'YYYY/MM/DD'), TO_DATE('2020/02/07', 'YYYY/MM/DD'), TO_DATE('2020/02/05', 'YYYY/MM/DD'));

INSERT INTO BORROW VALUES('14476512', '646', TO_DATE('2019/03/09', 'YYYY/MM/DD'), TO_DATE('2019/03/30', 'YYYY/MM/DD'), TO_DATE('2019/04/02', 'YYYY/MM/DD'));

```
140
141  INSERT INTO LATE_PAYMENT VALUES('5641', TO_DATE('2022/06/21', 'YYYY/MM/DD'),45.95);
142  INSERT INTO LATE_PAYMENT VALUES('1029', TO_DATE('2022/06/30', 'YYYY/MM/DD'),950.00);
143  INSERT INTO LATE_PAYMENT VALUES('3847', TO_DATE('2019/04/03', 'YYYY/MM/DD'),30.50);
144
145  SELECT * FROM LATE_PAYMENT;
```

Script Output × | Query Result ×

SQL | All Rows Fetched: 3 in 0.016 seconds

	PAYMENTID	PAYMENTDATE	AMOUNT
1	5641	21-06-22	45.95
2	1029	30-06-22	950
3	3847	03-04-19	30.5

INSERT INTO LATE_PAYMENT VALUES('5641', TO_DATE('2022/06/21', 'YYYY/MM/DD'),45.95);

INSERT INTO LATE_PAYMENT VALUES('1029', TO_DATE('2022/06/30', 'YYYY/MM/DD'),950.00);

INSERT INTO LATE_PAYMENT VALUES('3847', TO_DATE('2019/04/03', 'YYYY/MM/DD'),30.50);

```
146
147  INSERT INTO MAKES VALUES('5641','14525049');
148  INSERT INTO MAKES VALUES('1029','14424681');
149  INSERT INTO MAKES VALUES('3847','14476512');
150
151  SELECT * FROM MAKES;
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 3 in 0.008 seconds

	PAYMENTID	MEMBERID
1	5641	14525049
2	1029	14424681
3	3847	14476512

```
INSERT INTO MAKES VALUES('5641','14525049');
INSERT INTO MAKES VALUES('1029','14424681');
INSERT INTO MAKES VALUES('3847','14476512');
```

```
159
160  SELECT * FROM EQUIPMENT;
161
162  INSERT INTO HAS VALUES('2468','22');
163  INSERT INTO HAS VALUES('1234','33');
164  INSERT INTO HAS VALUES('5678',
165  INSERT INTO HAS VALUES('1213',
166  INSERT INTO HAS VALUES('1415',
167  INSERT INTO HAS VALUES('9101',
```

Script Output | Query Result 2

SQL | All Rows Fetched: 6 in 0.009 seconds

	EQUIPMENTID	EQUIPMENTNAME	ACQUIREDATE	EQUIPMENTCOST	QUANTITY
1	22	Printer	14-02-22	500	1
2	33	Scanner	10-03-21	350	1
3	44	Chair	03-11-19	4000	20
4	55	Desk	24-09-15	2100	30
5	66	Marker	13-04-22	100	4
6	77	White Board	05-05-18	12	1

```
INSERT INTO EQUIPMENT VALUES('22','Printer', TO_DATE('2022/02/14',
'YYYY/MM/DD'),500.00, 1);

INSERT INTO EQUIPMENT VALUES('33','Scanner', TO_DATE('2021/03/10',
'YYYY/MM/DD'), 350.00, 1);
```

```
INSERT INTO EQUIPMENT VALUES('44','Chair', TO_DATE('2019/11/03',
'YYYY/MM/DD'),4000.00,20);

INSERT INTO EQUIPMENT VALUES('55','Desk', TO_DATE('2015/09/24',
'YYYY/MM/DD'), 2100.00, 30);

INSERT INTO EQUIPMENT VALUES('66','Marker', TO_DATE('2022/04/13',
'YYYY/MM/DD'),100.00,4);

INSERT INTO EQUIPMENT VALUES('77', 'White Board', TO_DATE('2018/05/05',
'YYYY/MM/DD'),12.00,1);
```

```
161
162  INSERT INTO HAS VALUES('2468','22');
163  INSERT INTO HAS VALUES('1234','33');
164  INSERT INTO HAS VALUES('5678','66');
165  INSERT INTO HAS VALUES('1213','44');
166  INSERT INTO HAS VALUES('1415','55');
167  INSERT INTO HAS VALUES('9101','77');
168
169  SELECT * FROM HAS;
```

Script Output | Query Result

SQL | All Rows Fetched: 6 in 0.009 seconds

	ROOMID	EQUIPMENTID
1	2468	22
2	1234	33
3	5678	66
4	1213	44
5	1415	55
6	9101	77

```
INSERT INTO HAS VALUES('2468','22');
INSERT INTO HAS VALUES('1234','33');
INSERT INTO HAS VALUES('5678','66');
INSERT INTO HAS VALUES('1213','44');
INSERT INTO HAS VALUES('1415','55');
INSERT INTO HAS VALUES('9101','77');
```

QUERIES

1) Updated Member's Address

```
UPDATE MEMBER
SET address = '744 Broad Street, Philadelphia, PA 19104'
WHERE memberId = '14525049';
```

	MEMBERID	FIRSTNAME	LASTNAME	ADDRESS	DATEOFBIRTH	NUMBEROFBOOKS
1	14525049	Aditi	Salunkhe	500 N, 39th Street, Philadelphia, PA 19104	24-11-97	3

	MEMBERID	FIRSTNAME	LASTNAME	ADDRESS	DATEOFBIRTH	NUMBEROFBOOKS
1	14525049	Aditi	Salunkhe	744 Broad Street, Philadelphia, PA 19104	24-11-97	3

2) Viewing Maximum Salary, Minimum Salary and Average Salary

```
SELECT
MAX(wage) AS MAXWAGE,
MIN(wage) AS MINWAGE,
AVG(wage) AS AVGWAGE
FROM STAFF;
```

```
177
178 SELECT
179 MAX(wage) AS MAXWAGE,
180 MIN(wage) AS MINWAGE,
181 AVG(wage) AS AVGWAGE
182 FROM STAFF;
```

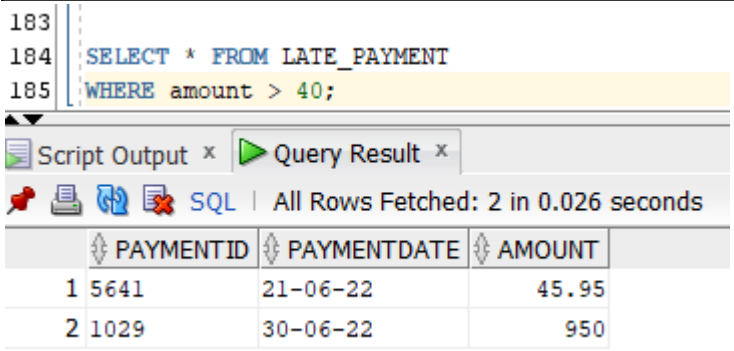
Script Output x | Query Result x

SQL | All Rows Fetched: 1 in 0.01 seconds

	MAXWAGE	MINWAGE	AVGWAGE
1	3450.67	1200.95	2350.79

3) Viewing Late Payment Fields where Amount is greater than 40

```
SELECT * FROM LATE_PAYMENT
WHERE amount > 40;
```



```
183
184  SELECT * FROM LATE_PAYMENT
185  WHERE amount > 40;
```

Script Output | Query Result | SQL | All Rows Fetched: 2 in 0.026 seconds

	PAYMENTID	PAYMENTDATE	AMOUNT
1	5641	21-06-22	45.95
2	1029	30-06-22	950

4) Updated Quantity of a specific equipment

```
UPDATE EQUIPMENT
SET quantity = quantity + 3
WHERE equipmentId = '22';
```

	EQUIPMENTID	EQUIPMENTNAME	ACQUIREDATE	EQUIPMENTCOST	QUANTITY
1	22	Printer	14-02-22	500	1

	EQUIPMENTID	EQUIPMENTNAME	ACQUIREDATE	EQUIPMENTCOST	QUANTITY
1	22	Printer	14-02-22	500	4

5) Altered Data Type of Column

```
## Initially it was VARCHAR2. We changed it to Number.

ALTER TABLE MEMBER
MODIFY (numberOfBooks NUMBER(1));
```

6) Updated Role of Staff Member

```
UPDATE STAFF
SET role = 'Manager'
WHERE memberId = '14498734';
```

```
14498734    1200.95 Afternoon Assistant
```

	MEMBERID	WAGE	SHIFTS	ROLE
1	14498734	1200.95	Afternoon	Manager

PROJECT SUMMARY

Over the course of the project, we have successfully created a library database management system that captures information about members, books, room, payments, equipment and other miscellaneous for the sake of organising pertinent information into a retrievable system. This was done in order to achieve the objective of storing information on the complete library organisation. To achieve that aim, we took the database schema we had previously established and implemented it in SQL to verify it fulfilled the demands indicated in our project requirements. This approach was effective in achieving the goals we set for data storage and handling data gracefully in a variety of contexts.

Various tables based on the entities generated during the design and outline stages earlier in the process were created in order to correctly store data. Data was then input into a SQL database to hold information about the library database in order to put it into practice. Data integrity had to be considered to guarantee that the data submitted was not compromised; otherwise, the database would be unable to hold data as planned, potentially causing problems. To that

purpose, a number of limitations were implemented to ensure data integrity. Constraints like 'NOT NULL', different key constraints, and others fall under this category.

Despite its time-based scope constraints, this database is completely capable of reducing the stress of membership management and successfully optimising monitoring data throughout this library's whole organisation. To summarise, this system is completely completed in its present iteration and fits the organisation's requirements as described in the outline proposal.