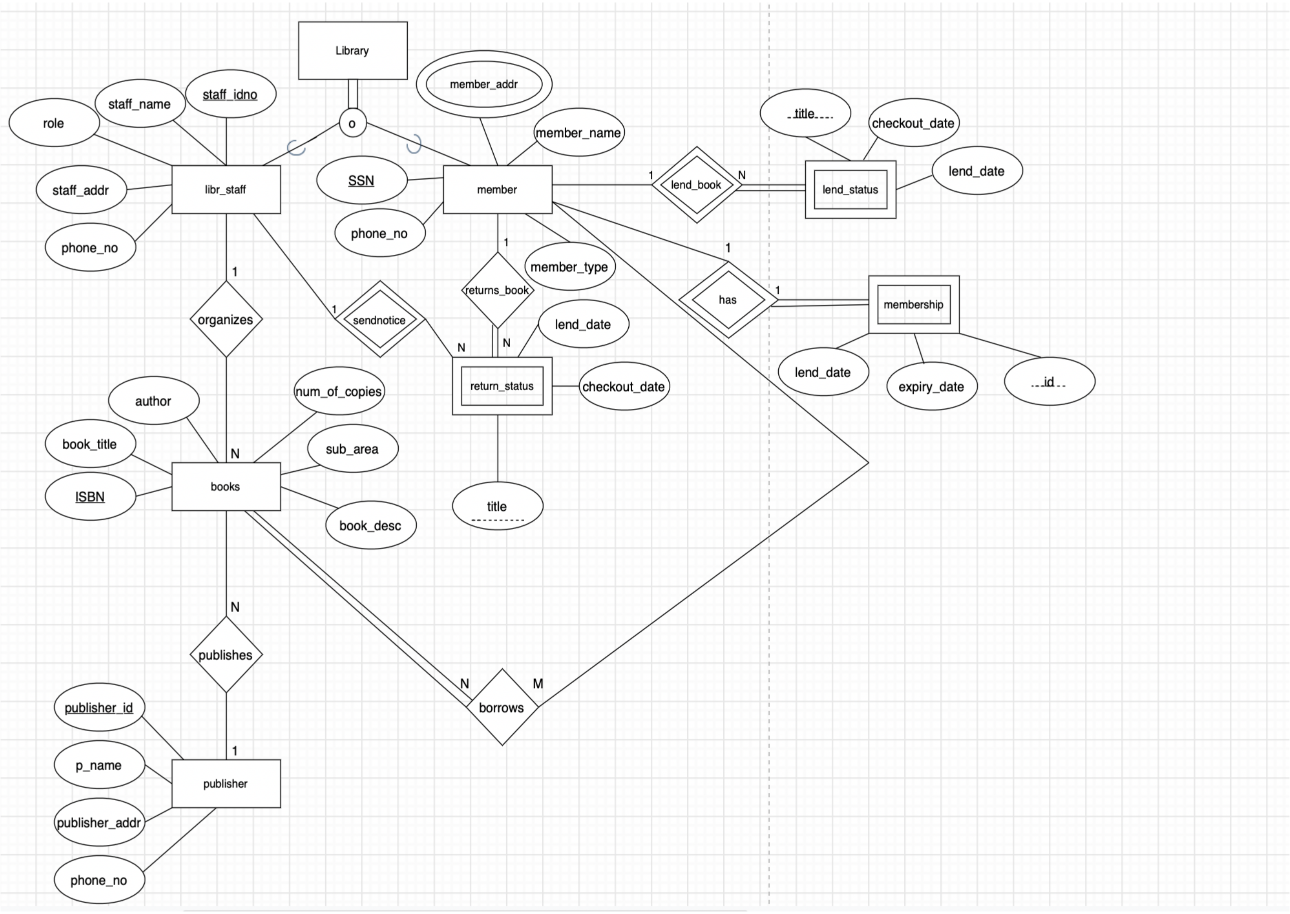
PROJECT2- PART1 & PART2

KAVYA SRI GARIKIPATI (1001953373)

AKHILA KOLAPALLI (1001977119)

EER MODEL



Making these assumptions in developing the EER model:

1) The superclass library entity is further subdivided into **libr_staff** and **member** subclasses. Every member of the superclass is involved in total participation. The subclasses are non-disjoint because a **staff** member can also be a member of the library, it is known as overlapping.

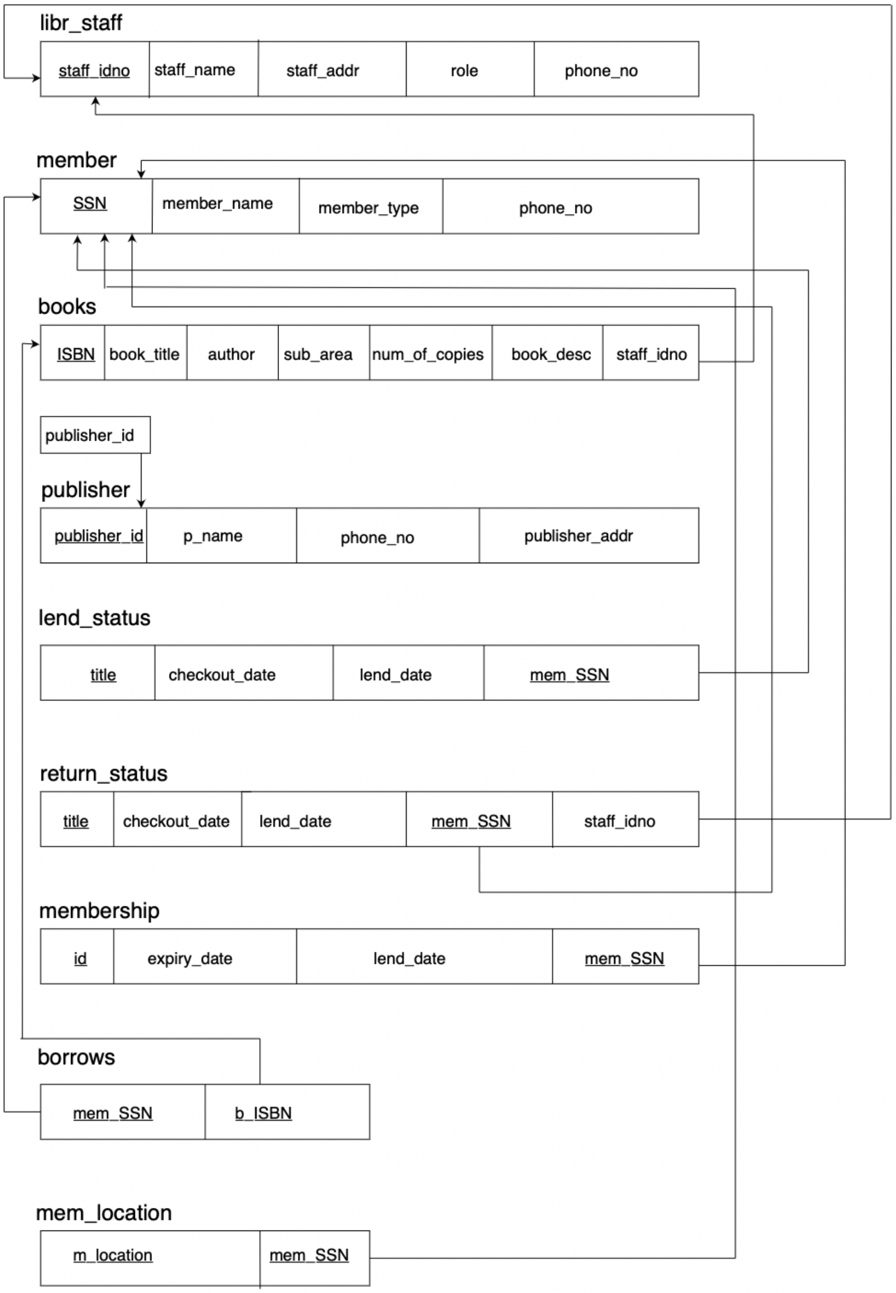
2) Since the entities **membership**, **lend_status**, and **return_status** cannot be apart, they are regarded as weak entities. title is thus a partial primary key for **lend_status**, and title serves as a primary key for **return _status** as well. **id** serves as a partial primary key for **membership**

3) We are assuming 1 to N relationship between the **publisher** and the **books**, the **libr_staff** and the **books**, **member** and **lend_status** with total participation from the **lend_status** side, between the **member** and the **return_status** with total participation from the **return_status** side.

4) Because a **member** can only have one **membership**, we are assuming a one-to-one relationship between the **member** and the **membership**.

5) We are presuming that **books** and **members** have a many-to-many relationship with total participation on the **books** end.

EER DESIGN TO RELATIONAL SCHEMA



libr_staff

staff_idno	staff_name	staff_addr	role	phone_no

member

SSN	member_name	member_type	phone_no

books

ISBN	book_title	author	sub_area	num_of_copies	book_desc	staff_idno

publisher_id

publisher

publisher_id	p_name	phone_no	publisher_addr

lend_status

title	checkout_date	lend_date	mem_SSN

return_status

title	checkout_date	lend_date	mem_SSN	staff_idno

membership

id	expiry_date	lend_date	mem_SSN

borrows

mem_SSN	b_ISBN

mem_location

m_location	mem_SSN

The EER model has been transformed into a relational schema by adhering to the rules listed below:

STEP 1: Mapping the strong entities:

The database contains four strong entities, which will be mapped first. They are **publisher**, **books**, **libr_staff**, and **member**.

STEP 2: Mapping the weak entities:

The **lend_status**, **return_status**, and **membership** entities are all weak entities. In all three entities, we will introduce the **mem_SSN** attribute, which, along with **title** (for lend_status and return_status), will serve as the primary key for **lend_status** and **return_status**. The **mem_SSN** attribute and **id** together will make up the primary key for **membership**.

STEP 3: Mapping 1:1 relationships:

We are adding the **member's** primary key as a foreign key on the **membership** side to map the relationship between **member** and **membership**.

STEP 4: Mapping 1: N relationships:

1) To map the relationship between **libr_staff** and **books**, we are adding the **libr_staff** primary key as a foreign key on the N side (**books**).

2) We are adding **publisher's** primary key as a foreign key on the N side(**books**) in order to map the relationship between **publisher** and **books**.

3) To map the relationship between **member** and **lend_status**, we are adding the **member's** primary key as a foreign key on the N side (**lend_status**).

4) We are adding the **member's** primary key as a foreign key on the N side (**return _status**) in order to map the relationship between **member** and **return_status**.

5)To map the relationship between **libr_staff** and **return_status**, the primary key of **libr_staff** is added as a foreign key on the N side (**return_status**).

STEP 5: Mapping many-to-many relationships:

For mapping **books** and **member**, we must add another table called '**borrows**' in which we add the primary keys of **books** and **member** as the foreign key on '**borrows**' side.

STEP 6: Adding another table:

For addresses with multiple valued attributes, we construct a new table called "**mem_location**" and add the **member's** primary key as a foreign key on the "**mem_location**" side.

TABLES ARE CREATED USING THE BELOW QUERIES

CREATE TABLE STATEMENTS

1.LIBR_STAFF TABLE

LIBR_STAFF table is created using this query:

```
CREATE TABLE LIBR_STAFF (STAFF_NAME VARCHAR (32) NOT NULL, STAFF_IDNO INT
NOT NULL PRIMARY KEY, STAFF_ADDR VARCHAR (32) NOT NULL, PHONE_NO INT NOT
NULL, ROLE VARCHAR (32) NOT NULL);
```

2.MEMBER TABLE

MEMBER table is created using this query:

```
CREATE TABLE MEMBER (SSN INT NOT NULL PRIMARY KEY, MEMBER_TYPE
VARCHAR (32) NOT NULL, MEMBER_NAME VARCHAR (32) NOT NULL,PHONE_NO INT
NOT NULL);
```

3.PUBLISHER TABLE

PUBLISHER table is created using this query:

```
CREATE TABLE PUBLISHER (PUBLISHER_ID INT NOT NULL PRIMARY KEY, P_NAME
VARCHAR (32) NOT NULL, PUBLISHER_ADDR VARCHAR(32) NOT NULL,PHONE_NO INT
NOT NULL);
```

4.BOOKS TABLE

BOOKS table is created using this query:

```
CREATE TABLE BOOKS(ISBN INT NOT NULL PRIMARY KEY, STAFF_IDNO INT NOT
NULL,PUBLISHER_ID INT NOT NULL,AUTHOR VARCHAR(32) NOT NULL,BOOK_TITLE
VARCHAR(32) NOT NULL,SUB_AREA VARCHAR(32) NOT NULL,NUM_OF_COPIES INT
NOT NULL,BOOK_DESC VARCHAR(32) NOT NULL, CONSTRAINT F_K_1 FOREIGN
KEY(STAFF_IDNO) REFERENCES LIBR_STAFF(STAFF_IDNO), CONSTRAINT F_K_2
FOREIGN KEY(PUBLISHER_ID) REFERENCES PUBLISHER(PUBLISHER_ID));
```

5.LEND_STATUS TABLE

LEND_STATUS table is created using this query:

```
CREATE TABLE LEND_STATUS (TITLE VARCHAR (32) NOT NULL, MEM_SSN INT NOT
NULL, LEND_DATE VARCHAR (32) NOT NULL,CHECKOUT_DATE VARCHAR(32) NOT
NULL,CONSTRAINT F_K_3 FOREIGN KEY(MEM_SSN) REFERENCES MEMBER(SSN));
```

6.RETURN_STATUS TABLE

RETURN_STATUS table is created using this query:

```
CREATE TABLE RETURN_STATUS (TITLE VARCHAR (32) NOT NULL, MEM_SSN INT
NOT NULL, STAFF_IDNO INT NOT NULL, LEND_DATE VARCHAR(32),CHECKOUT_DATE
VARCHAR(32),CONSTRAINT F_K_4 FOREIGN KEY(MEM_SSN) REFERENCES
MEMBER(SSN),CONSTRAINT F_K_5 FOREIGN KEY(STAFF_IDNO) REFERENCES
LIBR_STAFF(STAFF_IDNO));
```

7.MEMBERSHIP TABLE

MEMBERSHIP table is created using this query:

```
CREATE TABLE MEMBERSHIP (ID INT, MEM_SSN INT NOT NULL, LEND_DATE
VARCHAR (32), EXPIRE_DATE VARCHAR (32), CONSTRAINT F_K_6 FOREIGN
KEY(MEM_SSN) REFERENCES MEMBER(SSN));
```

8.BORROWS TABLE

BORROWS table is created using this query:

```
CREATE TABLE BORROWS (MEM_SSN INT, B_ISBN INT, CONSTRAINT FK_7 FOREIGN
KEY(MEM_SSN) REFERENCES MEMBER(SSN), CONSTRAINT F_K_8 FOREIGN
KEY(B_ISBN) REFERENCES BOOKS(ISBN));
```

9.MEM_LOCATION TABLE

MEM_LOCATION table is created using this query:

```
CREATE TABLE MEM_LOCATION (M_LOCATION VARCHAR (32), MEM_SSN INT NOT
NULL, CONSTRAINT F_K_9 FOREIGN KEY(MEM_SSN) REFERENCES MEMBER(SSN));
```

Primary keys used are as follows:

In LIBR_STAFF table, the primary key used is STAFF_IDNO

In MEMBER table, the primary key used is SSN

In PUBLISHER table, the primary key used is PUBLISHER_ID

In BOOKS table, the primary key used is ISBN

Foreign keys are referenced in this way:

In BOOKS table, FOREIGN KEY(STAFF_IDNO) REFERENCES LIBR_STAFF(STAFF_IDNO)

In BOOKS table, FOREIGN KEY(PUBLISHER_ID) REFERENCES PUBLISHER(PUBLISHER_ID)

In LEND_STATUS table, FOREIGN KEY(MEM_SSN) REFERENCES MEMBER(SSN)

In RETURN_STATUS table, FOREIGN KEY(MEM_SSN) REFERENCES MEMBER(SSN)

In RETURN_STATUS table, FOREIGN KEY(STAFF_IDNO) REFERENCES LIBR_STAFF(STAFF_IDNO)

In MEMBERSHIP table, FOREIGN KEY(MEM_SSN) REFERENCES MEMBER(SSN)

In BORROWS table, FOREIGN KEY(MEM_SSN) REFERENCES MEMBER(SSN)

In BORROWS table, FOREIGN KEY(B_ISBN) REFERENCES BOOKS(ISBN)

In MEM_LOCATION table, FOREIGN KEY(MEM_SSN) REFERENCES MEMBER(SSN)

REFERENCES:

1. https://www.guru99.com/er-diagram-tutorial-dbms.html
2. https://www.databasestar.com/entity-relationship-diagram/
3. https://course.ccs.neu.edu/cs3200sp18s3/ssl/lectures/lecture_08_mapping.pdf
4. https://www.tutorialspoint.com/dbms/er_model_to_relational_model.html