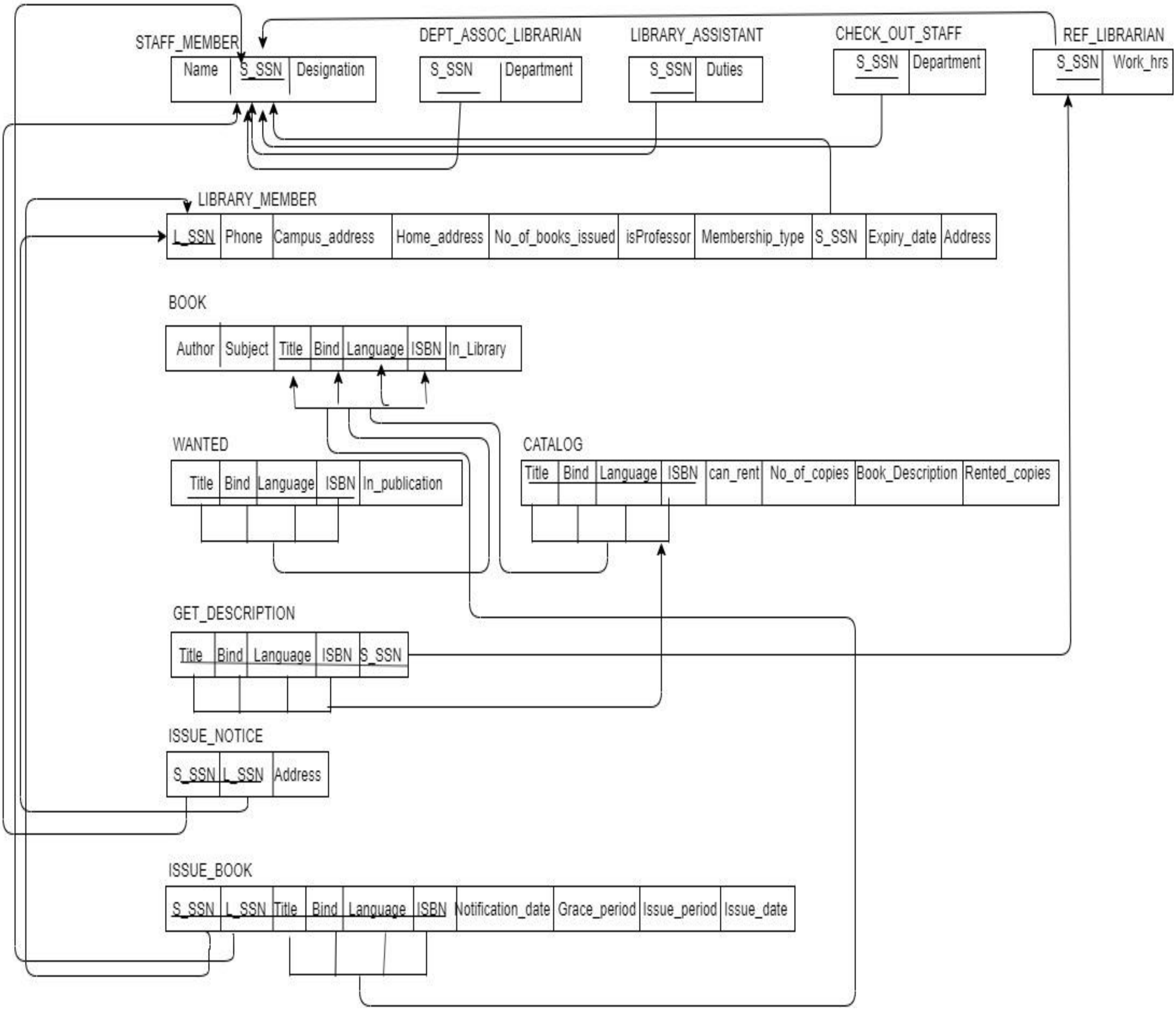
# RELATIONAL MAPPING REPORT

A **relational schema** for a **database** is an outline of how data is organized. It can be a graphic illustration or another kind of chart used by programmers to understand how each table is laid out, including the columns and the types of data they hold and how tables connect.

Following is the relational database schema for the LIBRARY database:



The following steps are followed while converting the schema to the relational database:

- For each regular (strong) entity types STAFF_MEMBER, LIBRARY_MEMBER and BOOK in the ER schema, create a relation R that includes all the *simple* attributes of the entities and choose one of the key attributes as primary key for R.

- For the BOOK entity type since the chosen primary key is *composite*, the set of simple attributes that form it will together form the primary key of R.
- For each regular binary 1:N relationship type R such as ISSUE_CARD, we identified the relation S that represent the participating entity type at the N-side of the relationship type and included as foreign key in S the primary key of the relation T that represents the other entity type participating in R. Also, we included any simple attributes of the relationship in the relational table.
- For each regular binary M:N relationship type R such as ISSUE_CARD and GET_DESCRIPTION, we created a new relation S to represent R. We also include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; their combination will form the primary key of S. Also, we included any simple attributes of the relationship in the relational table.
- For a ternary relationship type R such as ISSUE_BOOK, where n=3, we created a new relationship relation S to represent R. And also included as foreign key attributes in S the primary keys of the relations that represent the participating entity types.
- For each specialization of type STAFF_MEMBER, BOOK and CATALOG we created new relations for each subclass and include a foreign key as the primary key referencing the superclass which it is specialized from and also the simple attributes of the specialization type.

Following is the create table commands for the LIBRARY DATABASE:

```
CREATE TABLE STAFF_MEMBER
(
NAME VARCHAR(20),
S_SSN VARCHAR(15) PRIMARY KEY,
DESIGNATION VARCHAR(20)
);

CREATE TABLE LIBRARY_MEMBER
(
PHONE_NO VARCHAR(15),
L_SSN VARCHAR(15) PRIMARY KEY,
CAMPUS_ADDRESS VARCHAR(50),
HOME_ADDRESS VARCHAR(50),
NO_OF_BOOKS_ISSUED INT(5),
ISPROFESSOR BOOLEAN,
MEMBERSHIP_TYPE VARCHAR(10),
S_SSN VARCHAR(15),
CARD_EXPIRY_DATE DATE,
CARD_ADDRESS VARCHAR(20),
FOREIGN KEY(S_SSN) REFERENCES STAFF_MEMBER(S_SSN)
);

CREATE TABLE BOOK
(
AUTHOR VARCHAR(20),
SUBJECT VARCHAR(20),
TITLE VARCHAR(20),
BIND VARCHAR(20),
LANGUAGE VARCHAR(15),
ISBN VARCHAR(20),
IN_LIBRARY BOOLEAN,
PRIMARY KEY (TITLE,BIND,LANGUAGE,ISBN)
);

CREATE TABLE ISSUE_NOTICE
(
S_SSN VARCHAR(15),
L_SSN VARCHAR(15),
ADDRESS VARCHAR(20),
FOREIGN KEY(S_SSN) REFERENCES STAFF_MEMBER(S_SSN),
FOREIGN KEY(L_SSN) REFERENCES LIBRARY_MEMBER(L_SSN),
PRIMARY KEY(S_SSN,L_SSN)
);

CREATE TABLE ISSUE_BOOK
(
TITLE VARCHAR(20),
BIND VARCHAR(20),
LANGUAGE VARCHAR(15),
ISBN VARCHAR(20),
S_SSN VARCHAR(15),
L_SSN VARCHAR(15),
NOTIFICATION_DATE DATE,
GRACE_PERIOD VARCHAR(10),
ISSUE_PERIOD VARCHAR(10),
ISSUE_DATE DATE,
FOREIGN KEY(S_SSN) REFERENCES STAFF_MEMBER(S_SSN),
FOREIGN KEY(L_SSN) REFERENCES LIBRARY_MEMBER(L_SSN),
FOREIGN KEY(TITLE) REFERENCES BOOK(TITLE),
FOREIGN KEY(BIND) REFERENCES BOOK(BIND),
FOREIGN KEY(LANGUAGE) REFERENCES BOOK(LANGUAGE),
FOREIGN KEY(ISBN) REFERENCES BOOK(ISBN),
PRIMARY KEY(TITLE,BIND,LANGUAGE,ISBN,S_SSN,L_SSN)
);

CREATE TABLE DEPT_ASSOC_LIBRARIAN
(
S_SSN VARCHAR(10) PRIMARY KEY,
DEPT VARCHAR(10),
FOREIGN KEY(S_SSN) REFERENCES STAFF_MEMBER(S_SSN)
);

CREATE TABLE LIBRARY_ASSISTANT
(
S_SSN VARCHAR(10) PRIMARY KEY,
DUTIES VARCHAR(10),
FOREIGN KEY(S_SSN) REFERENCES STAFF_MEMBER(S_SSN)
);

CREATE TABLE REFERENCE_LIBRARIAN
(
S_SSN VARCHAR(10) PRIMARY KEY,
DEPT VARCHAR(10),
FOREIGN KEY(S_SSN) REFERENCES STAFF_MEMBER(S_SSN)
);

CREATE TABLE CHECK_OUT_STAFF
(
S_SSN VARCHAR(10) PRIMARY KEY,
WRK_HRS VARCHAR(10),
FOREIGN KEY(S_SSN) REFERENCES STAFF_MEMBER(S_SSN)
);

CREATE TABLE WANTED
(
TITLE VARCHAR(20),
BIND VARCHAR(20),
LANGUAGE VARCHAR(15),
ISBN VARCHAR(20),
IN_PUBLICATION VARCHAR(10),
FOREIGN KEY(TITLE) REFERENCES BOOK(TITLE),
FOREIGN KEY(BIND) REFERENCES BOOK(BIND),
FOREIGN KEY(LANGUAGE) REFERENCES BOOK(LANGUAGE),
FOREIGN KEY(ISBN) REFERENCES BOOK(ISBN),
PRIMARY KEY (TITLE,BIND,LANGUAGE,ISBN)
);

CREATE TABLE CATALOG
(
TITLE VARCHAR(20),
BIND VARCHAR(20),
LANGUAGE VARCHAR(15),
ISBN VARCHAR(20),
BOOK_DESCRIPTION VARCHAR(10),
NO_OF_COPIES INT(5),
CAN_RENT BOOLEAN,
TYPE_OF_BOOK VARCHAR(10),
RENTED_COPIES INT(5),
FOREIGN KEY(TITLE) REFERENCES BOOK(TITLE),
FOREIGN KEY(BIND) REFERENCES BOOK(BIND),
FOREIGN KEY(LANGUAGE) REFERENCES BOOK(LANGUAGE),
FOREIGN KEY(ISBN) REFERENCES BOOK(ISBN),
PRIMARY KEY (TITLE,BIND,LANGUAGE,ISBN)
);

CREATE TABLE GET_DESCRIPTION
(
TITLE VARCHAR(20),
BIND VARCHAR(20),
LANGUAGE VARCHAR(15),
ISBN VARCHAR(20),
S_SSN VARCHAR(10),
FOREIGN KEY(TITLE) REFERENCES CATALOG(TITLE),
FOREIGN KEY(BIND) REFERENCES CATALOG(BIND),
FOREIGN KEY(LANGUAGE) REFERENCES CATALOG(LANGUAGE),
FOREIGN KEY(ISBN) REFERENCES CATALOG(ISBN),
FOREIGN KEY(S_SSN) REFERENCES REFERENCE_LIBRARIAN(S_SSN),
PRIMARY KEY (TITLE,BIND,LANGUAGE,ISBN,S_SSN)
);
```