LIBRARY DATABASE DESIGN PROJECT

Name
Ala Salah Morshed Abdo

Index:

Part 1: Requirements Analysis

Designing the library management system at Taiz University requires organizing book categories into multiple sections. Each section is identified by a unique identifier and also has a distinct name. Each section contains a collection of books, where each book is identified by its own identifier. Each book has the following properties: book title, book identifier, and publication date. To track the number of available copies of each book, a registry is created to record the number of available copies. The author of multiple books is determined by their name and also has nationality and date of birth.

These books are published by a publishing house or publisher that includes an identifier, name, email, and address (city, street).

Each section has a group of employees who work on organizing the borrowing operations. Each employee has the following properties: employee type, first name, middle name, last name.

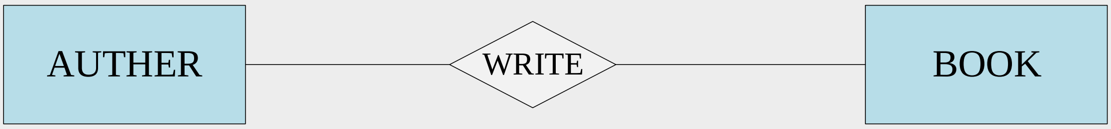
Readers borrow from the library and have the following properties: identifier, first name, middle name, last name, phone number, email, and address. Readers can request multiple borrowings, and each borrowing has its own invoice that includes the identifier, borrowing date, return date.

Part 2:

Conceptual Design

1.
    a. Entities : Author ,Book
    b. Relationships :

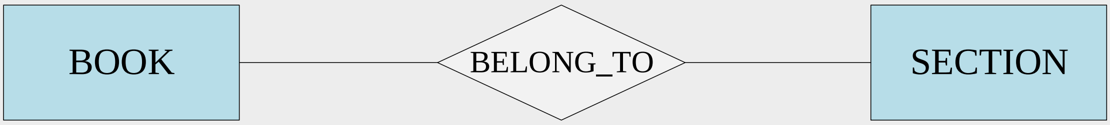
    One Author writes many books(1:M)



2.

a. Entities : Book,Section
b. Relationships :

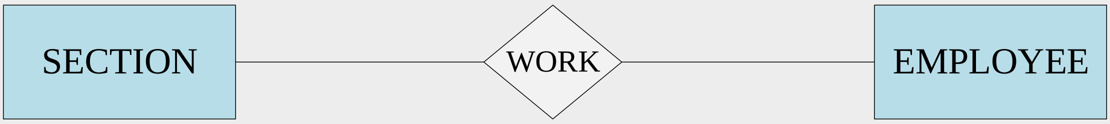
Many Books belongs to a section (M:1)



3.

a. Entities : Employee, Section
b. Relationships:

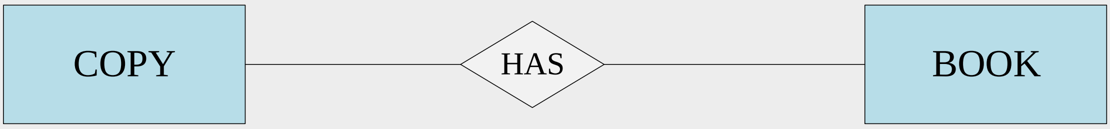
Many Employees works in a section (M:1)



4.

a. Entities : Book ,Copies
b. Relationships :

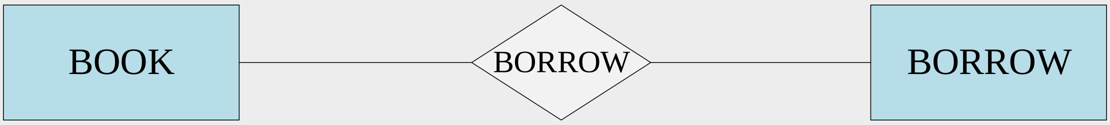
One Book has a copies (1:M)



5.

a. Entities : Borrowing ,Book
b. Relationships :

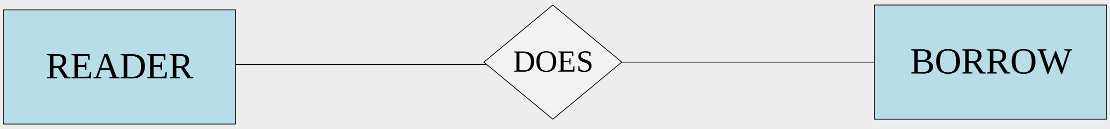
One Borrowing borrows many book (1:1)



6.

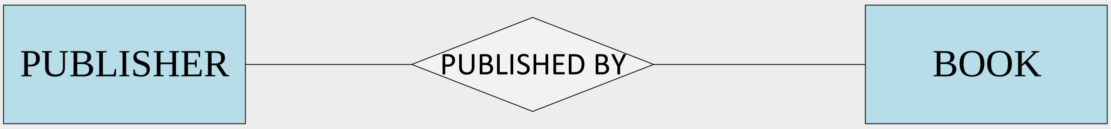
a. Entities : Borrowing ,Book
b. Relationships:

One Borrowing borrows a book (1:1)

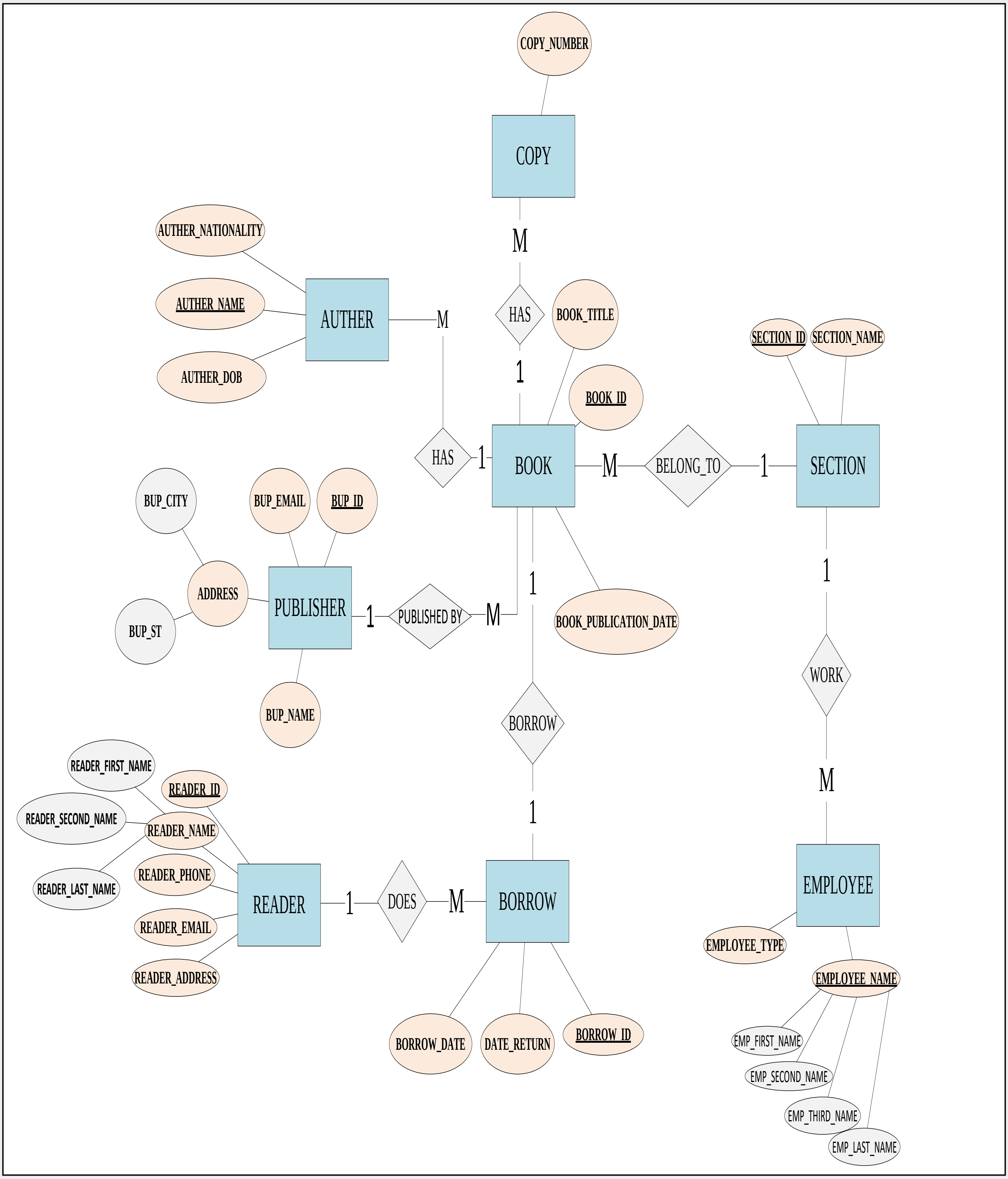


7-

a. Entities: BOOK, BUPLISHER
b. Relationships:
c. One published by many book(1:M)



ER Design


COPY_NUMBER
COPY
M
HAS
1
AUTHER_NATIONALITY
AUTHER_NAME
AUTHER_DOB
AUTHER
M
BOOK_TITLE
BOOK_ID
HAS
1
BOOK
M
BELONG_TO
1
SECTION_ID
SECTION_NAME
SECTION
1
WORK
M
EMPLOYEE
EMPLOYEE_TYPE
EMPLOYEE_NAME
EMP_FIRST_NAME
EMP_SECOND_NAME
EMP_THIRD_NAME
EMP_LAST_NAME
BUP_CITY
BUP_EMAIL
BUP_ID
ADDRESS
BUP_ST
PUBLISHER
1
PUBLISHED BY
M
BUP_NAME
1
BOOK_PUBLICATION_DATE
BORROW
1
READER_FIRST_NAME
READER_ID
READER_SECOND_NAME
READER_NAME
READER_LAST_NAME
READER_PHONE
READER_EMAIL
READER_ADDRESS
READER
1
DOES
M
BORROW
BORROW_DATE
DATE_RETURN
BORROW_ID

Part 3:

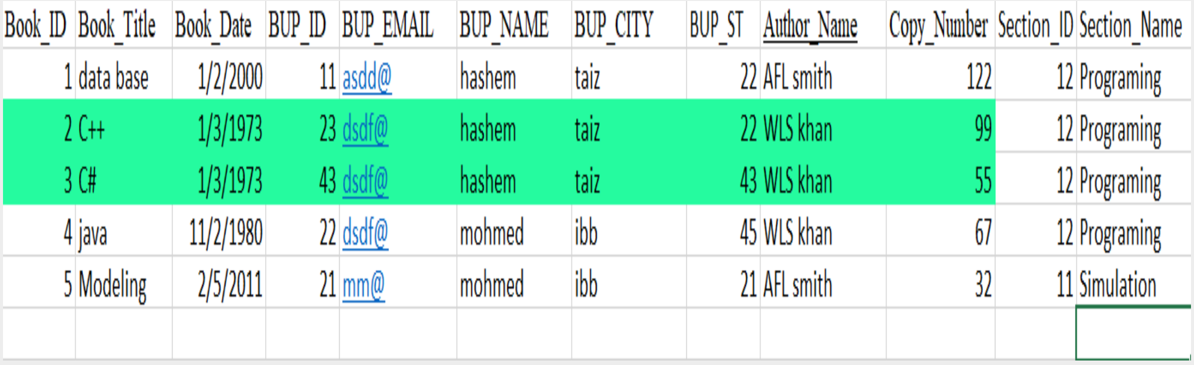
Detailed Design

Fill the following table for above ER diagram:

Entities	Attributes	Relationship	Primary Key	Foreign Key
BOOK	Book_ID Bup_id Book_Title Book_Date Section_ID	Is borrowed by a Reader	Book_ID	Section Bup_id
BUPLISHER	BUP_ID BUP_EMAIL BUP_NAME BUP_CITY BUP_ST	Buplishes many books	ID	
READER	Reader_ID Reader_First_Name Reader_Second_Name Reader_Last_Name Reader_Phone_Nmber Reader_Email; Reader_Address	Borrows book	Reader_ID	
AUTHOR	Author_Name Nation Dob Book_ID	Writes book	Author_Name	Book_ID
EMPLOEE	Emp_First_Name Emp_Second_Name Emp_Third_Name Emp_Last_Name Section_ID Emp_Type	Works in section	Emp_First_Name, Emp_Second_Name, Emp_Third_Name, Emp_Last_Name	Section_ID
BORROW	Borrow_ID Reader_ID Book_ID Borrow_Date Borrow_Return_Date	Borrowing book	Borrow_ID	Reader_ID Book_ID
COPY	Book-ID Copy_Number	Has copies of books		Book-ID
SECTION	Section_ID Section_Name	Has books	Section_ID	

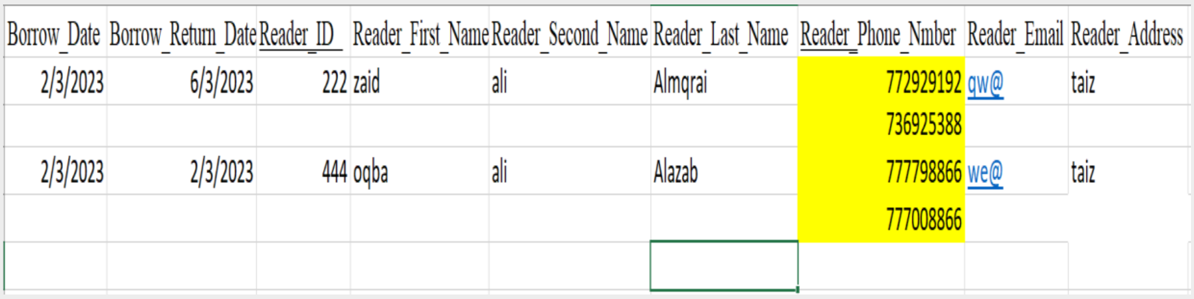
Normalization:

Book table



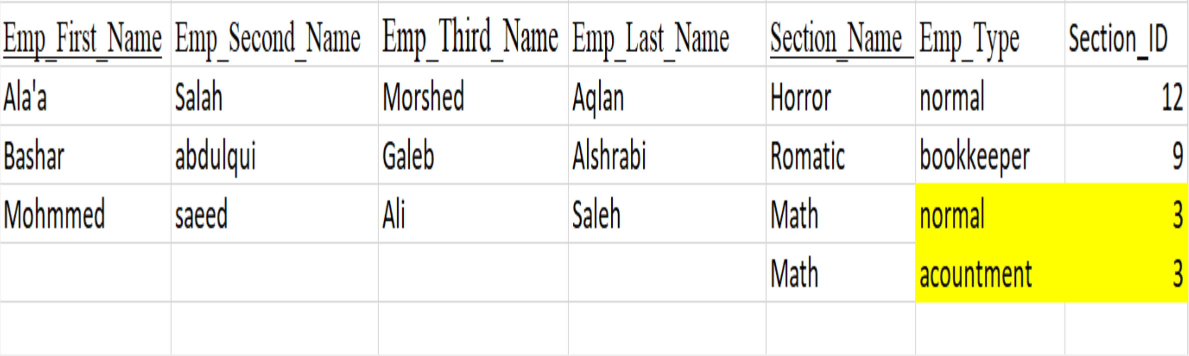
Book_ID	Book_Title	Book_Date	BUP_ID	BUP_EMAIL	BUP_NAME	BUP_CITY	BUP_ST	Author_Name	Copy_Number	Section_ID	Section_Name
1	data base	1/2/2000	11	asdd@	hashem	taiz	22	AFL smith	122	12	Programing
2	C++	1/3/1973	23	dsdf@	hashem	taiz	22	WLS khan	99	12	Programing
3	C#	1/3/1973	43	dsdf@	hashem	taiz	43	WLS khan	55	12	Programing
4	java	11/2/1980	22	dsdf@	mohmed	ibb	45	WLS khan	67	12	Programing
5	Modeling	2/5/2011	21	mm@	mohmed	ibb	21	AFL smith	32	11	Simulation

Borrow table



Borrow_Date	Borrow_Return_Date	Reader_ID	Reader_First_Name	Reader_Second_Name	Reader_Last_Name	Reader_Phone_Nmber	Reader_Email	Reader_Address
2/3/2023	6/3/2023	222	zaid	ali	Almqrai	772929192	qw@	taiz
						736925388		
2/3/2023	2/3/2023	444	oqba	ali	Alazab	777798866	we@	taiz
						777008866		

Employee table

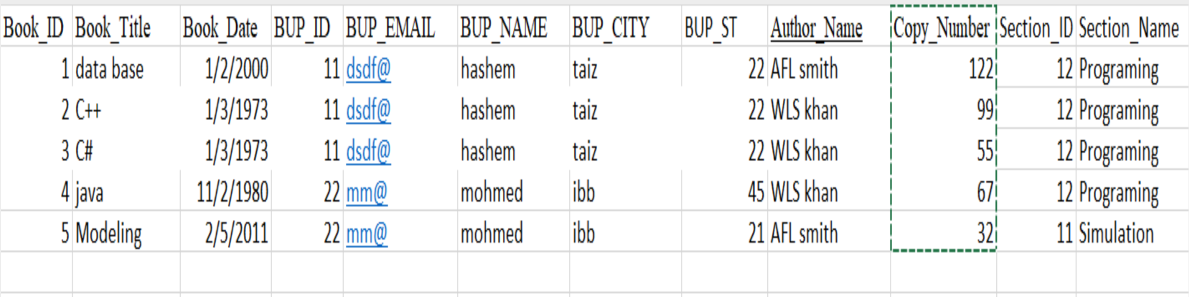


Emp_First_Name	Emp_Second_Name	Emp_Third_Name	Emp_Last_Name	Section_Name	Emp_Type	Section_ID
Ala'a	Salah	Morshed	Aqlan	Horror	normal	12
Bashar	abdulqui	Galeb	Alshrabi	Romatic	bookkeeper	9
Mohmmed	saeed	Ali	Saleh	Math	normal	3
				Math	acountment	3

The first normalization for the three tables

To remove the repeating data and drop the anomalies insertion,deletion and updating

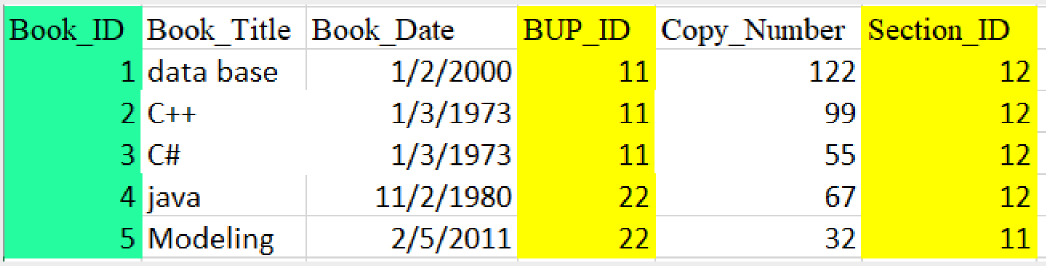
For the book table



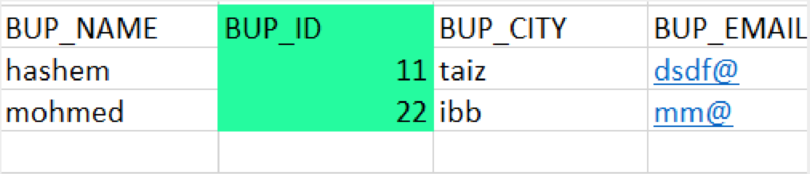
Book_ID	Book_Title	Book_Date	BUP_ID	BUP_EMAIL	BUP_NAME	BUP_CITY	BUP_ST	Author_Name	Copy_Number	Section_ID	Section_Name
1	data base	1/2/2000	11	dsdf@	hashem	taiz	22	AFL smith	122	12	Programing
2	C++	1/3/1973	11	dsdf@	hashem	taiz	22	WLS khan	99	12	Programing
3	C#	1/3/1973	11	dsdf@	hashem	taiz	22	WLS khan	55	12	Programing
4	java	11/2/1980	22	mm@	mohmed	ibb	45	WLS khan	67	12	Programing
5	Modeling	2/5/2011	22	mm@	mohmed	ibb	21	AFL smith	32	11	Simulation

repeating in the records so we remove this repeating by applying the first and the second normalization in one step 1NF ,2NF,3NF

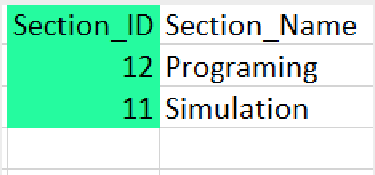
To remove the partial dependency separate the publisher and the auther and section from the book



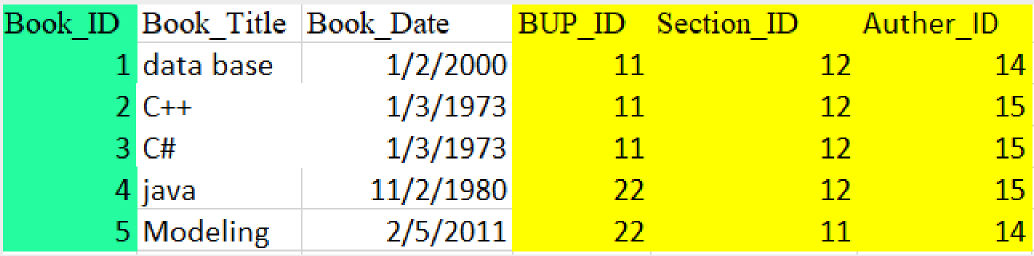
Book_ID	Book_Title	Book_Date	BUP_ID	Copy_Number	Section_ID
1	data base	1/2/2000	11	122	12
2	C++	1/3/1973	11	99	12
3	C#	1/3/1973	11	55	12
4	java	11/2/1980	22	67	12
5	Modeling	2/5/2011	22	32	11



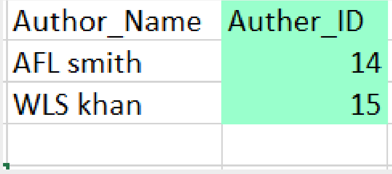
BUP_NAME	BUP_ID	BUP_CITY	BUP_EMAIL
hashem	11	taiz	dsdf@
mohmed	22	ibb	mm@



Section_ID	Section_Name
12	Programing
11	Simulation

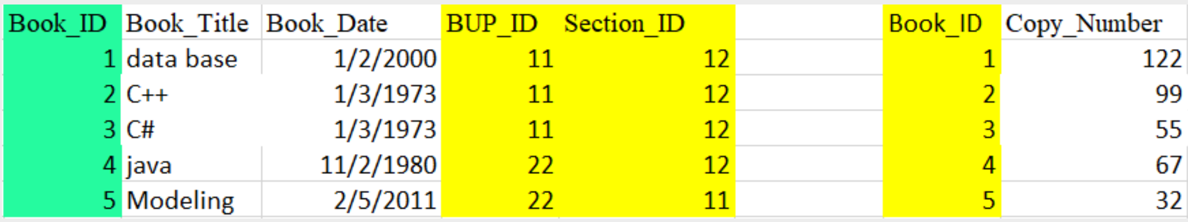


Book_ID	Book_Title	Book_Date	BUP_ID	Section_ID	Auther_ID
1	data base	1/2/2000	11	12	14
2	C++	1/3/1973	11	12	15
3	C#	1/3/1973	11	12	15
4	java	11/2/1980	22	12	15
5	Modeling	2/5/2011	22	11	14



Author_Name	Auther_ID
AFL smith	14
WLS khan	15

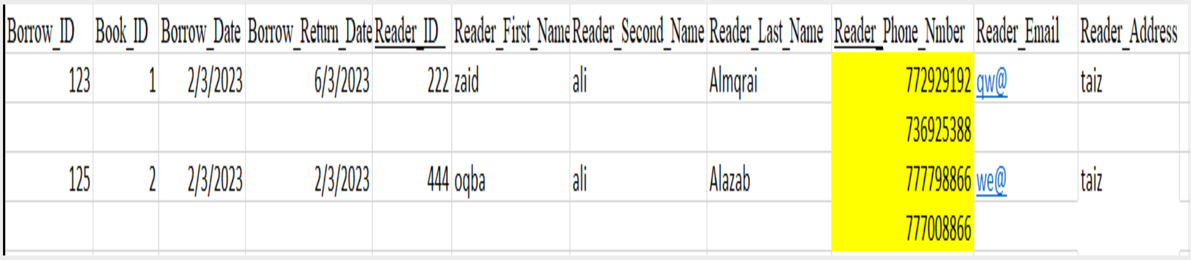
To remove the transitive dependency separate te copies from the book



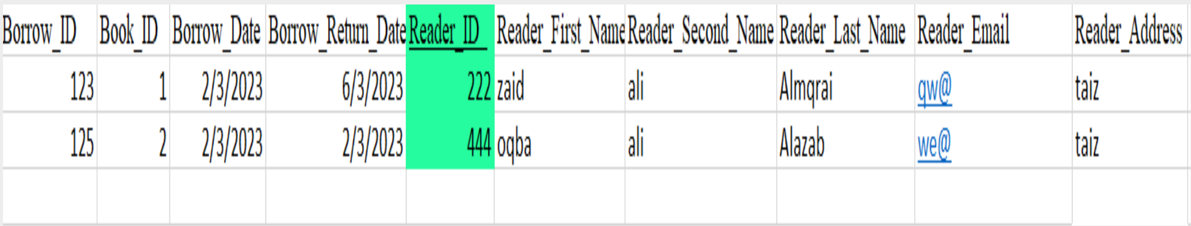
Book_ID	Book_Title	Book_Date	BUP_ID	Section_ID		Book_ID	Copy_Number
1	data base	1/2/2000	11	12		1	122
2	C++	1/3/1973	11	12		2	99
3	C#	1/3/1973	11	12		3	55
4	java	11/2/1980	22	12		4	67
5	Modeling	2/5/2011	22	11		5	32

For the borrow table

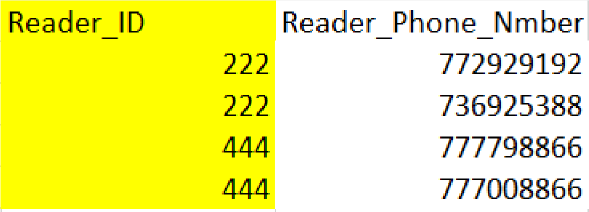
The first normalization 1FN to remove the repeating values



Borrow_ID	Book_ID	Borrow_Date	Borrow_Return_Date	Reader_ID	Reader_First_Name	Reader_Second_Name	Reader_Last_Name	Reader_Phone_Nmber	Reader_Email	Reader_Address
123	1	2/3/2023	6/3/2023	222	zaid	ali	Almqrai	772929192	qw@	taiz
								736925388		
125	2	2/3/2023	2/3/2023	444	oqba	ali	Alazab	777798866	we@	taiz
								777008866		



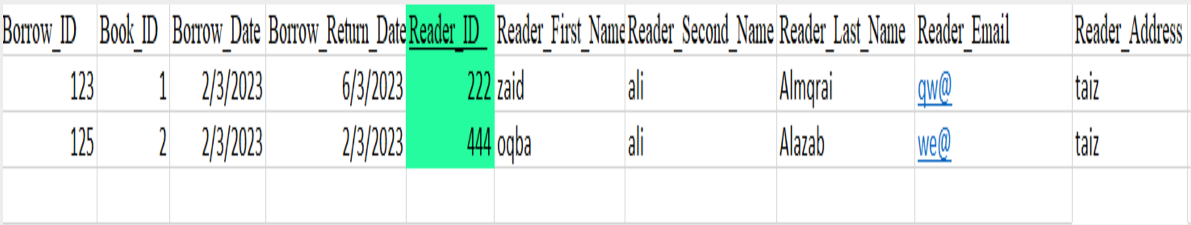
Borrow_ID	Book_ID	Borrow_Date	Borrow_Return_Date	Reader_ID	Reader_First_Name	Reader_Second_Name	Reader_Last_Name	Reader_Email	Reader_Address
123	1	2/3/2023	6/3/2023	222	zaid	ali	Almqrai	qw@	taiz
125	2	2/3/2023	2/3/2023	444	oqba	ali	Alazab	we@	taiz



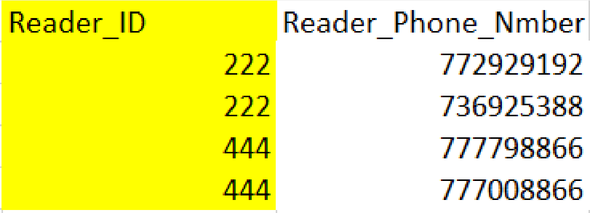
Reader_ID	Reader_Phone_Nmber
222	772929192
222	736925388
444	777798866
444	777008866

The second normalization

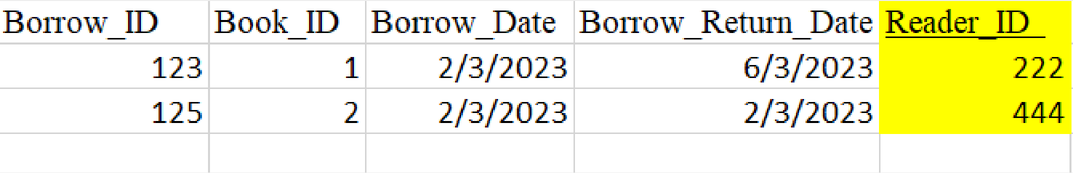
To remove the partial dependency



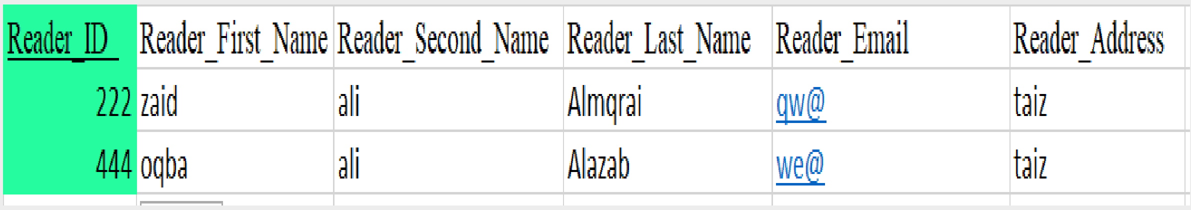
Borrow_ID	Book_ID	Borrow_Date	Borrow_Return_Date	Reader_ID	Reader_First_Name	Reader_Second_Name	Reader_Last_Name	Reader_Email	Reader_Address
123	1	2/3/2023	6/3/2023	222	zaid	ali	Almqrai	qw@	taiz
125	2	2/3/2023	2/3/2023	444	oqba	ali	Alazab	we@	taiz



Reader_ID	Reader_Phone_Nmber
222	772929192
222	736925388
444	777798866
444	777008866

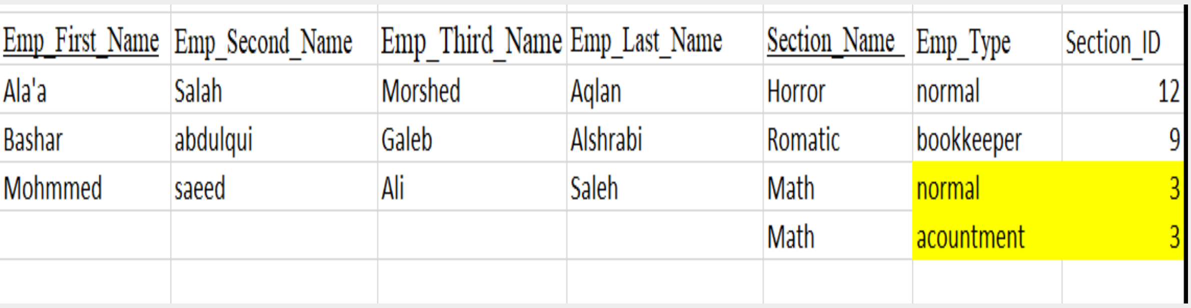


Borrow_ID	Book_ID	Borrow_Date	Borrow_Return_Date	Reader_ID
123	1	2/3/2023	6/3/2023	222
125	2	2/3/2023	2/3/2023	444



Reader_ID	Reader_First_Name	Reader_Second_Name	Reader_Last_Name	Reader_Email	Reader_Address
222	zaid	ali	Almqrai	qw@	taiz
444	oqba	ali	Alazab	we@	taiz

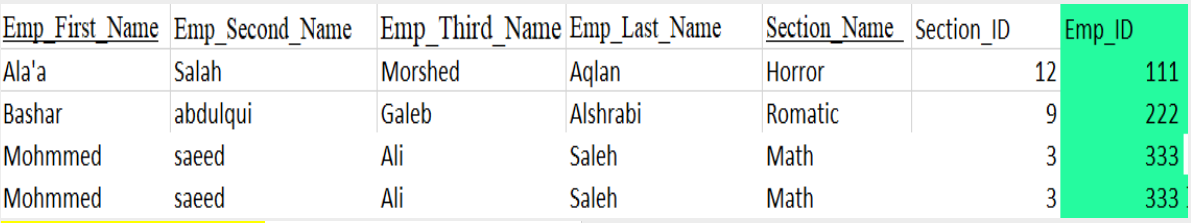
For the Employee table



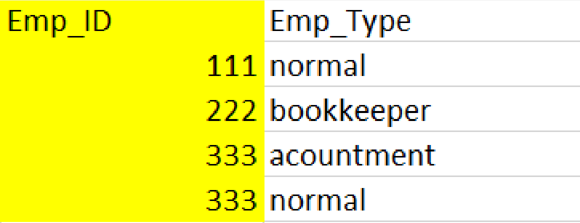
Emp_First_Name	Emp_Second_Name	Emp_Third_Name	Emp_Last_Name	Section_Name	Emp_Type	Section_ID
Ala'a	Salah	Morshed	Aqlan	Horror	normal	12
Bashar	abdulqui	Galeb	Alshrabi	Romatic	bookkeeper	9
Mohmmed	saeed	Ali	Saleh	Math	normal	3
				Math	acountment	3

The first 1NF

Removing the repeating data



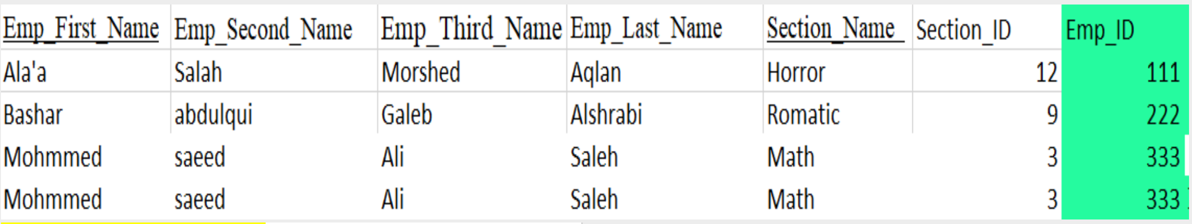
Emp_First_Name	Emp_Second_Name	Emp_Third_Name	Emp_Last_Name	Section_Name	Section_ID	Emp_ID
Ala'a	Salah	Morshed	Aqlan	Horror	12	111
Bashar	abdulqui	Galeb	Alshrabi	Romatic	9	222
Mohmmed	saeed	Ali	Saleh	Math	3	333
Mohmmed	saeed	Ali	Saleh	Math	3	333



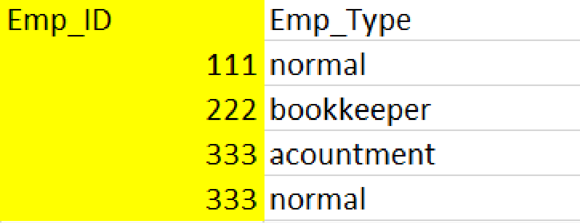
Emp_ID	Emp_Type
111	normal
222	bookkeeper
333	acountment
333	normal

The second 2NF

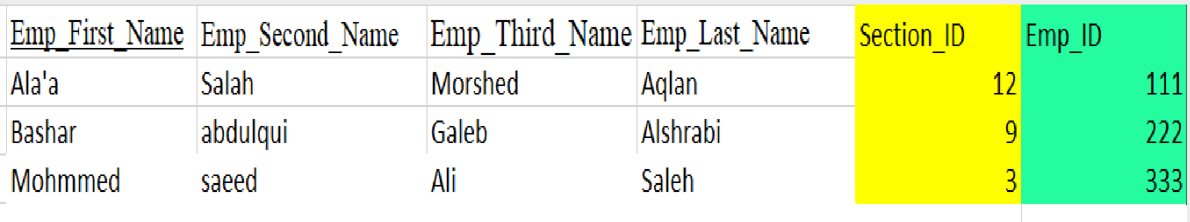
Removing the partial dependency



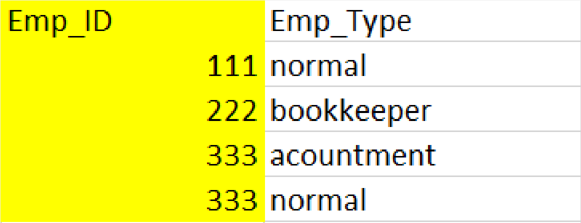
Emp_First_Name	Emp_Second_Name	Emp_Third_Name	Emp_Last_Name	Section_Name	Section_ID	Emp_ID
Ala'a	Salah	Morshed	Aqlan	Horror	12	111
Bashar	abdulqui	Galeb	Alshrabi	Romatic	9	222
Mohmmed	saeed	Ali	Saleh	Math	3	333
Mohmmed	saeed	Ali	Saleh	Math	3	333



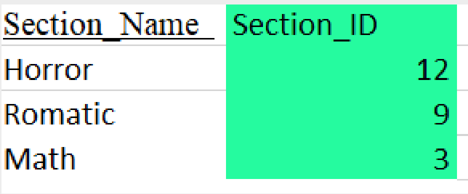
Emp_ID	Emp_Type
111	normal
222	bookkeeper
333	acountment
333	normal



Emp_First_Name	Emp_Second_Name	Emp_Third_Name	Emp_Last_Name	Section_ID	Emp_ID
Ala'a	Salah	Morshed	Aqlan	12	111
Bashar	abdulqui	Galeb	Alshrabi	9	222
Mohmmed	saeed	Ali	Saleh	3	333



Emp_ID	Emp_Type
111	normal
222	bookkeeper
333	acountment
333	normal



Section_Name	Section_ID
Horror	12
Romatic	9
Math	3

The 3NF : to remove the transitive dependency

No tarnsitive dependency here

Part 4: Physical Model (Mapping)

Entity Type: **BUPLISHER**		
Attribute	**Data Type**	**Required/Optional**
BUP_ID	Number (4)	Required
BUP_EMAIL	Varchar2 (30)	Required
BUP_CITY	Varchar2(20)	Required
BUP_ST	Varchar2(15)	Required
BUP_NAME	Varchar2(20)	Required

Entity Type: **SECTION**		
Attributes	**Data Type**	**Required/Optional**
Section_ID	Number (4)	Required
Section_Name	Varchar2(12)	Required

Entity Type: **READER**		
Attributes	**Data Type**	**Required/Optional**
Attributes	**Data Type**	**Required/Optional**
Reader_ID	Number (4)	Required
Reader_First_Name	Varchar2(10)	Required
Reader_Second_Name	Varchar2(10)	Required
Reader_Last_Name	Varchar2(10)	Required
Phone_Nmber	Number (9)	Required
Email	Date	Required

Entity Type: **BOOK**		
Attributes	**Data Type**	**Required/Optional**
Book_ID	Number (4)	Required
Bup_ID	Number(4)	Required
Book_Title	Varchar2(20)	Required
Date	Date	Required
Section_id	Number (4)	Required

Entity Type: **AUTHOR**		
Attributes	**Data Type**	**Required/Optional**
Author_Name	Varchar2(18)	Required
Nation	Varchar2(15)	Required
DOB	Date	Required
Book_ID	Number(4)	Required

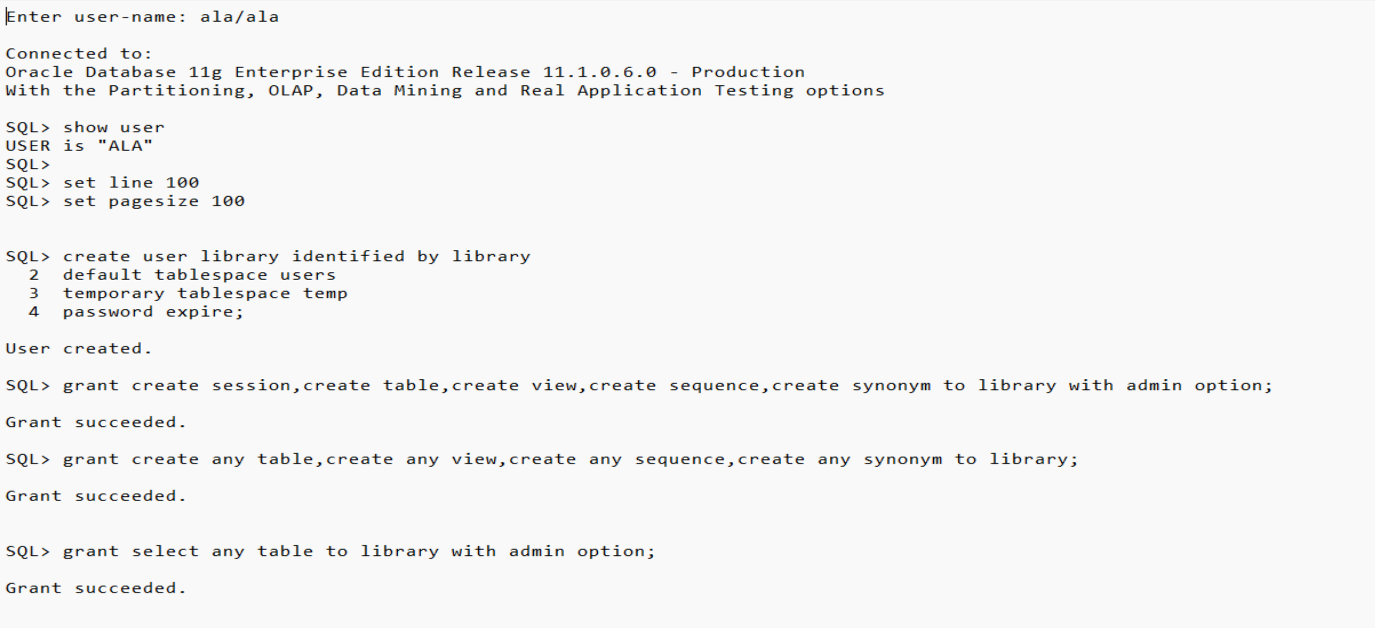
Entity Type: **COPY**		
Attributes	**Data Type**	**Required/Optional**
Book_ID	Number (4)	Required
Copy_Number	Number (4)	Required

Entity Type: **BORROW**		
Attributes	**Data Type**	**Required/Optional**
Borrow_ID	Number (4)	Required
Reader_ID	Number (4)	Required
Book_ID	Number (4)	Required
Date_Borrow	Date	Required
Date_Return	Date	Required

Entity Type: **EMPLOYEE**		
Attributes	**Data Type**	**Required/Optional**
Emp_First_Name	Varchar2(18)	Required
Emp_Second_Name	Varchar2(18)	Required
Emp_Third_Name	Varchar2(18)	Required
Emp_Last_Name	Varchar2(18)	Required
Section_ID	Number (4)	Required
Emp_Type	Varchar2(10)	Required

Part 5: Implementation of Physical Model

Note: Implement into any DBMS and share screenshots

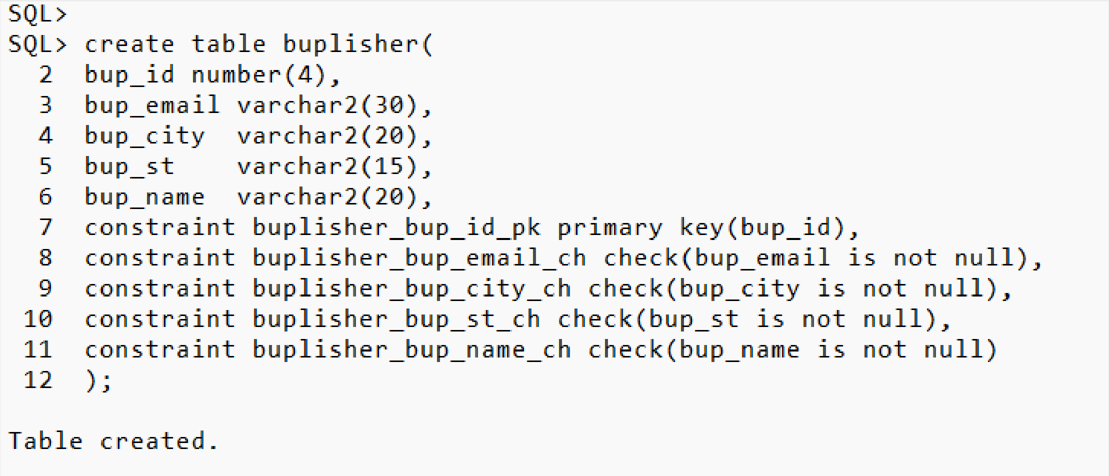


```
Enter user-name: ala/ala

Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> show user
USER is "ALA"
SQL>
SQL> set line 100
SQL> set pagesize 100


SQL> create user library identified by library
  2  default tablespace users
  3  temporary tablespace temp
  4  password expire;

User created.

SQL> grant create session,create table,create view,create sequence,create synonym to library with admin option;

Grant succeeded.

SQL> grant create any table,create any view,create any sequence,create any synonym to library;

Grant succeeded.


SQL> grant select any table to library with admin option;

Grant succeeded.
```

- Construct CREATE statement for each table

Create statements for BUPLISHER table:

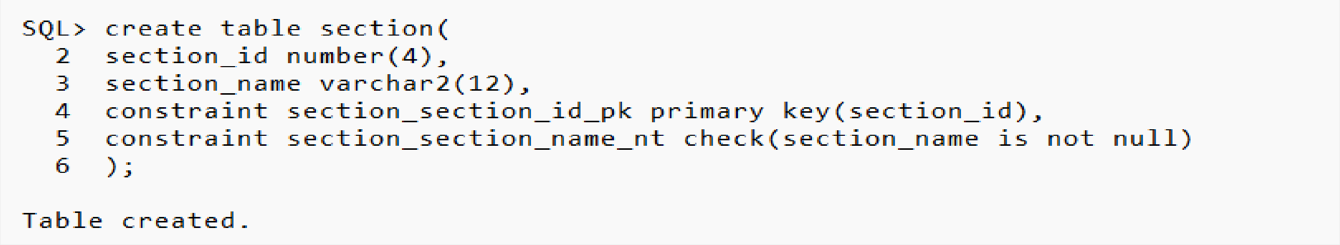
```
create table buplisher(
bup_id number(4),
bup_email varchar2(30),
bup_city  varchar2(20),
bup_st    varchar2(15),
bup_name  varchar2(20),
constraint buplisher_bup_id_pk primary key(bup_id),
constraint buplisher_bup_email_ch check(bup_email is not null),
constraint buplisher_bup_city_ch check(bup_city is not null),
constraint buplisher_bup_st_ch check(bup_st is not null),
constraint buplisher_bup_name_ch check(bup_name is not null)
);
```



```
SQL>
SQL> create table buplisher(
  2  bup_id number(4),
  3  bup_email varchar2(30),
  4  bup_city  varchar2(20),
  5  bup_st    varchar2(15),
  6  bup_name  varchar2(20),
  7  constraint buplisher_bup_id_pk primary key(bup_id),
  8  constraint buplisher_bup_email_ch check(bup_email is not null),
  9  constraint buplisher_bup_city_ch check(bup_city is not null),
 10  constraint buplisher_bup_st_ch check(bup_st is not null),
 11  constraint buplisher_bup_name_ch check(bup_name is not null)
 12  );

Table created.
```

Create statements for SECTION table:

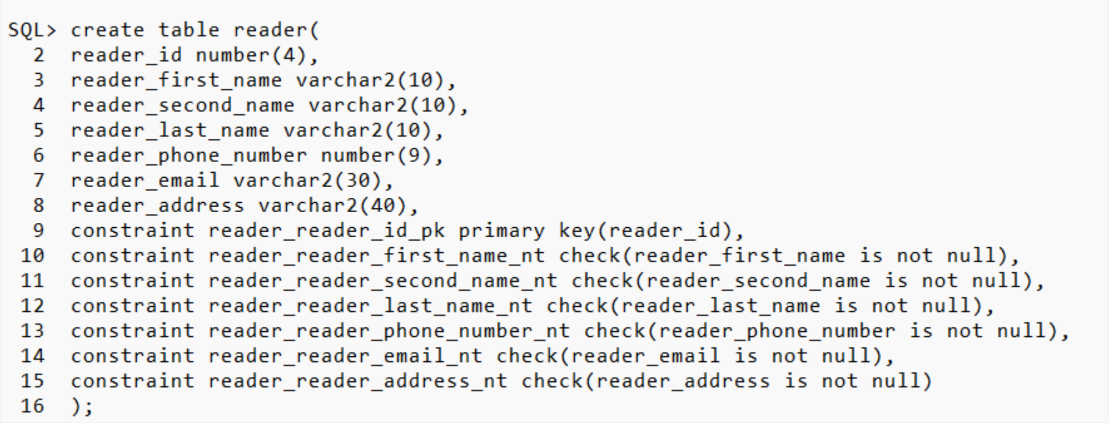
```
create table section(
create table section(
section_id number(4),
section_name varchar2(12),
constraint section_section_id_pk primary key(section_id),
constraint section_section_name_nt check(section_name is not null)
);
```



```
SQL> create table section(
  2  section_id number(4),
  3  section_name varchar2(12),
  4  constraint section_section_id_pk primary key(section_id),
  5  constraint section_section_name_nt check(section_name is not null)
  6  );

Table created.
```

Create statements for READER table:

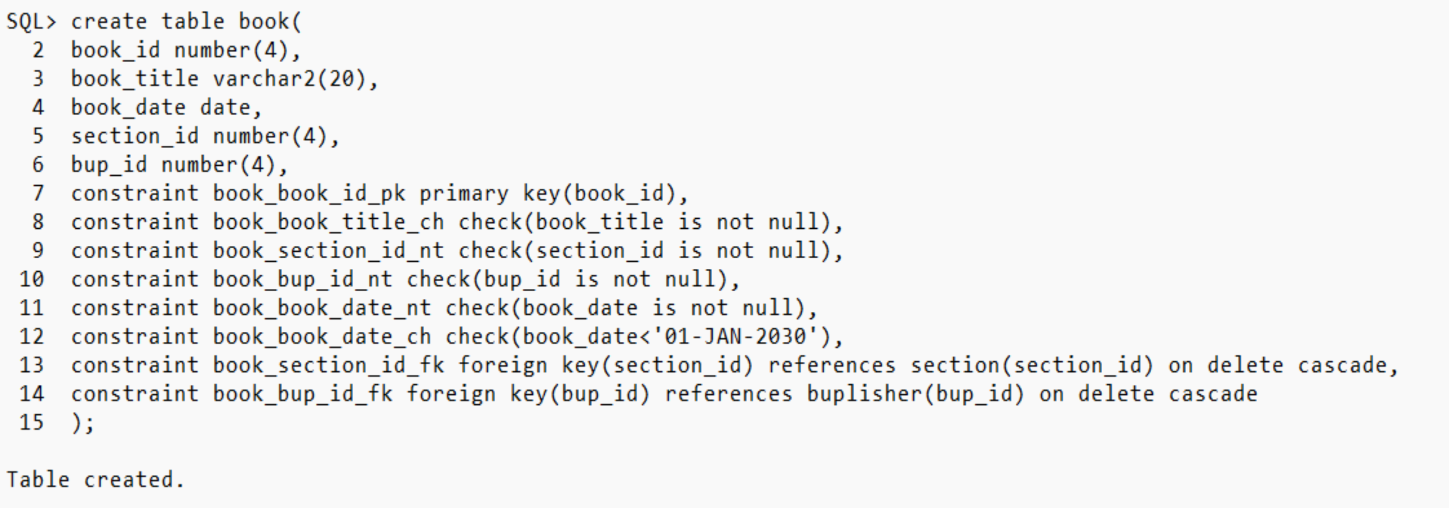
```
create table reader(
reader_id number(4),
reader_first_name varchar2(10),
reader_second_name varchar2(10),
reader_last_name varchar2(10),
reader_phone_number number(9),
reader_email varchar2(30),
reader_address varchar2(40),
constraint reader_reader_id_pk primary key(reader_id),
constraint reader_reader_first_name_nt check(reader_first_name is not null),
constraint reader_reader_second_name_nt check(reader_second_name is not null),
constraint reader_reader_last_name_nt check(reader_last_name is not null),
constraint reader_reader_phone_number_nt check(reader_phone_number is not null),
constraint reader_reader_email_nt check(reader_email is not null),
constraint reader_reader_address_nt check(reader_address is not null)
);
```



```
SQL> create table reader(
  2  reader_id number(4),
  3  reader_first_name varchar2(10),
  4  reader_second_name varchar2(10),
  5  reader_last_name varchar2(10),
  6  reader_phone_number number(9),
  7  reader_email varchar2(30),
  8  reader_address varchar2(40),
  9  constraint reader_reader_id_pk primary key(reader_id),
 10  constraint reader_reader_first_name_nt check(reader_first_name is not null),
 11  constraint reader_reader_second_name_nt check(reader_second_name is not null),
 12  constraint reader_reader_last_name_nt check(reader_last_name is not null),
 13  constraint reader_reader_phone_number_nt check(reader_phone_number is not null),
 14  constraint reader_reader_email_nt check(reader_email is not null),
 15  constraint reader_reader_address_nt check(reader_address is not null)
 16  );
```

Create statements for BOOK table:

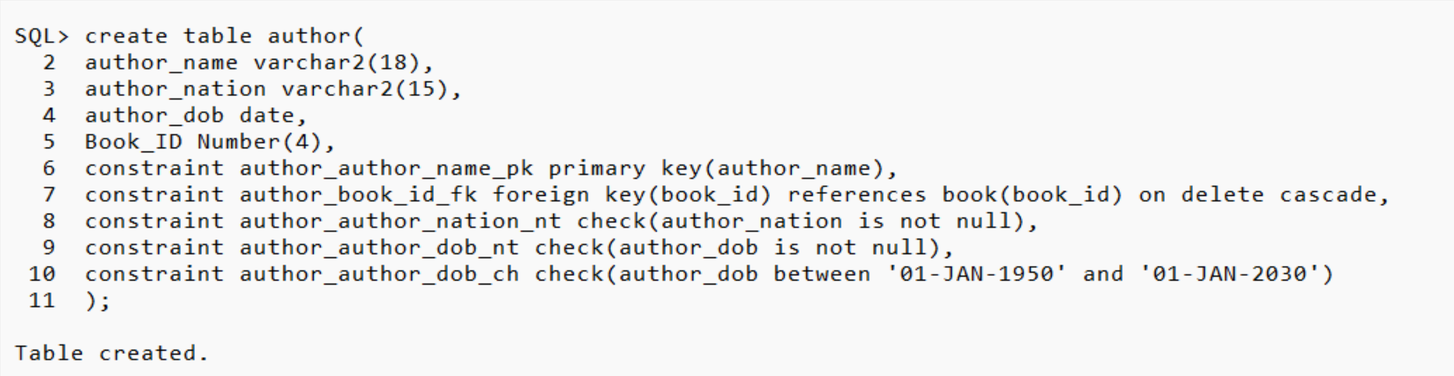
```
create table book(
book_id number(4),
book_title varchar2(20),
book_date date,
section_id number(4),
bup_id number(4),
constraint book_book_id_pk primary key(book_id),
constraint book_book_title_ch check(book_title is not null),
constraint book_section_id_nt check(section_id is not null),
constraint book_bup_id_nt check(bup_id is not null),
constraint book_book_date_nt check(book_date is not null),
constraint book_book_date_ch check(book_date<'01-JAN-2030'),
constraint book_section_id_fk foreign key(section_id) references section(section_id) on delete
cascade,
constraint book_bup_id_fk foreign key(bup_id) references buplisher(bup_id) on delete cascade
);
```



```
SQL> create table book(
book_id number(4),
book_title varchar2(20),
book_date date,
section_id number(4),
bup_id number(4),
constraint book_book_id_pk primary key(book_id),
constraint book_book_title_ch check(book_title is not null),
constraint book_section_id_nt check(section_id is not null),
constraint book_bup_id_nt check(bup_id is not null),
constraint book_book_date_nt check(book_date is not null),
constraint book_book_date_ch check(book_date<'01-JAN-2030'),
constraint book_section_id_fk foreign key(section_id) references section(section_id) on delete cascade,
constraint book_bup_id_fk foreign key(bup_id) references buplisher(bup_id) on delete cascade
);

Table created.
```

Create statements for AUTHOR table:

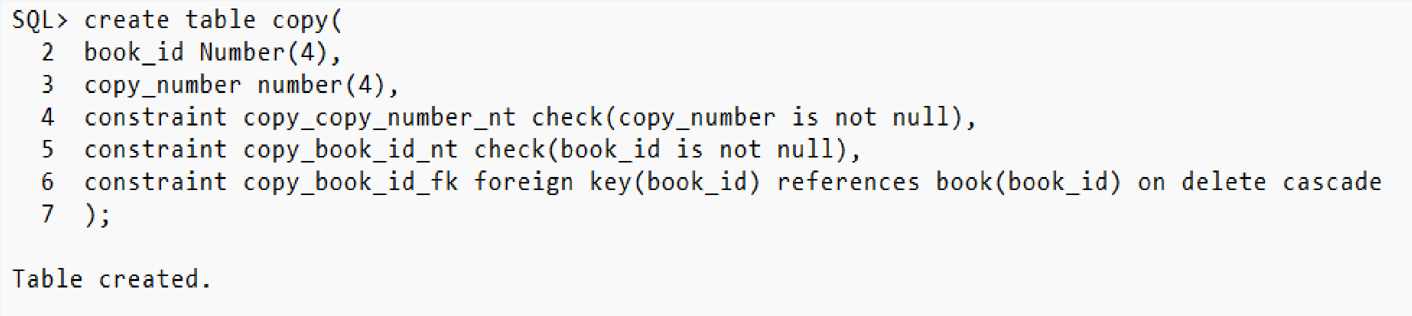
```
create table author(
author_name varchar2(18),
author_nation varchar2(15),
author_dob date,
Book_ID Number(4),
constraint author_author_name_pk primary key(author_name),
constraint author_book_id_fk foreign key(book_id) references book(book_id) on delete cascade,
constraint author_author_nation_nt check(author_nation is not null),
constraint author_author_dob_nt check(author_dob is not null),
constraint author_author_dob_ch check(author_dob between '01-JAN-1950' and '01-JAN-2030')
);
```



```
SQL> create table author(
  2  author_name varchar2(18),
  3  author_nation varchar2(15),
  4  author_dob date,
  5  Book_ID Number(4),
  6  constraint author_author_name_pk primary key(author_name),
  7  constraint author_book_id_fk foreign key(book_id) references book(book_id) on delete cascade,
  8  constraint author_author_nation_nt check(author_nation is not null),
  9  constraint author_author_dob_nt check(author_dob is not null),
 10  constraint author_author_dob_ch check(author_dob between '01-JAN-1950' and '01-JAN-2030')
 11  );

Table created.
```

Create statements for COPY table:

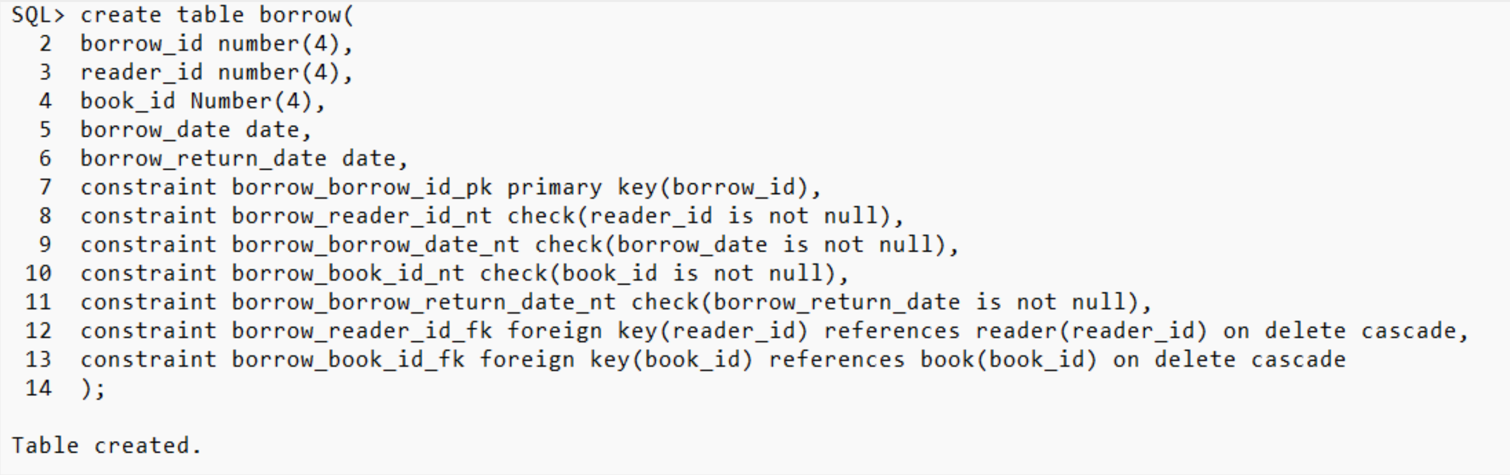
```
create table copy(
book_id Number(4),
copy_number number(4),
constraint copy_copy_number_nt check(copy_number is not null),
constraint copy_book_id_nt check(book_id is not null),
constraint copy_book_id_fk foreign key(book_id) references book(book_id) on delete cascade
);
```



```
SQL> create table copy(
book_id Number(4),
copy_number number(4),
constraint copy_copy_number_nt check(copy_number is not null),
constraint copy_book_id_nt check(book_id is not null),
constraint copy_book_id_fk foreign key(book_id) references book(book_id) on delete cascade
);

Table created.
```

Create statements for BORROW table:

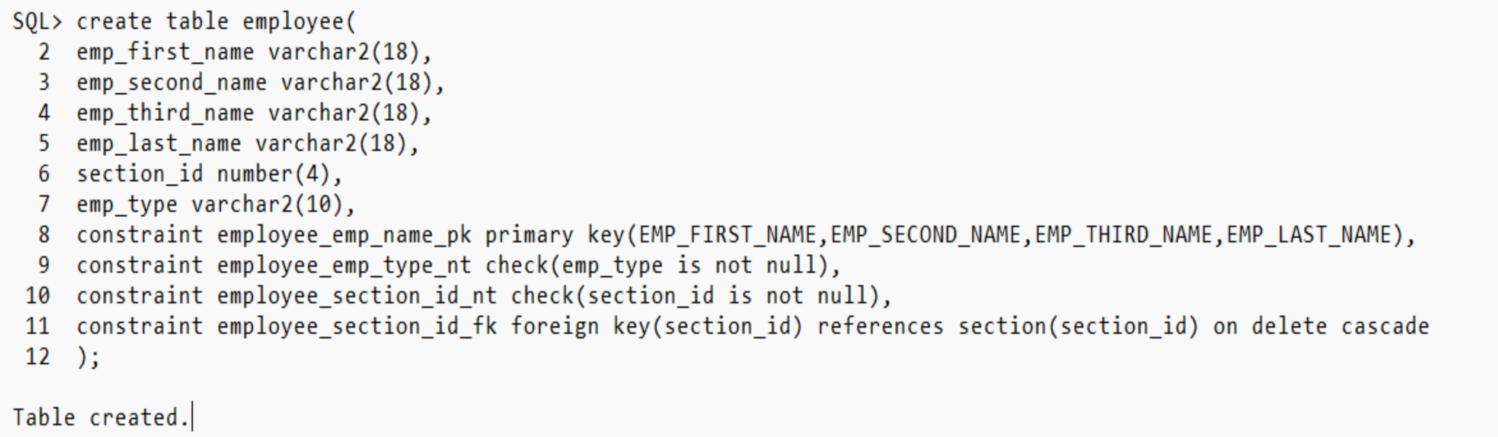
```
create table borrow(
borrow_id number(4),
reader_id number(4),
book_id Number(4),
borrow_date date,
borrow_return_date date,
constraint borrow_borrow_id_pk primary key(borrow_id),
constraint borrow_reader_id_nt check(reader_id is not null),
constraint borrow_borrow_date_nt check(borrow_date is not null),
constraint borrow_book_id_nt check(book_id is not null),
constraint borrow_borrow_return_date_nt check(borrow_return_date is not null),
constraint borrow_reader_id_fk foreign key(reader_id) references reader(reader_id) on delete cascade,
constraint borrow_book_id_fk foreign key(book_id) references book(book_id) on delete cascade
);
```



```
SQL> create table borrow(
borrow_id number(4),
reader_id number(4),
book_id Number(4),
borrow_date date,
borrow_return_date date,
constraint borrow_borrow_id_pk primary key(borrow_id),
constraint borrow_reader_id_nt check(reader_id is not null),
constraint borrow_borrow_date_nt check(borrow_date is not null),
constraint borrow_book_id_nt check(book_id is not null),
constraint borrow_borrow_return_date_nt check(borrow_return_date is not null),
constraint borrow_reader_id_fk foreign key(reader_id) references reader(reader_id) on delete cascade,
constraint borrow_book_id_fk foreign key(book_id) references book(book_id) on delete cascade
);

Table created.
```

Create statements for EMPLOYEE table:

```
create table employee(
emp_first_name varchar2(18),
emp_second_name varchar2(18),
emp_third_name varchar2(18),
emp_last_name varchar2(18),
section_id number(4),
emp_type varchar2(10),
constraint employee_emp_name_pk primary key(EMP_FIRST_NAME,EMP_SECOND_NAME,EMP_THIRD_NAME,EMP_LAST_NAME),
constraint employee_emp_type_nt check(emp_type is not null),
constraint employee_section_id_nt check(section_id is not null),
constraint employee_section_id_fk foreign key(section_id) references section(section_id) on delete cascade
);
```



```
SQL> create table employee(
  2  emp_first_name varchar2(18),
  3  emp_second_name varchar2(18),
  4  emp_third_name varchar2(18),
  5  emp_last_name varchar2(18),
  6  section_id number(4),
  7  emp_type varchar2(10),
  8  constraint employee_emp_name_pk primary key(EMP_FIRST_NAME,EMP_SECOND_NAME,EMP_THIRD_NAME,EMP_LAST_NAME),
  9  constraint employee_emp_type_nt check(emp_type is not null),
 10  constraint employee_section_id_nt check(section_id is not null),
 11  constraint employee_section_id_fk foreign key(section_id) references section(section_id) on delete cascade
 12  );

Table created.
```

Part 6: Design queries

Populate the tables with dummy values. Design as many queries as possible?

To inquire about the names of the authors and the addresses of their books:

```
Select p.bup_name , b.book_title
From buplisher p , book b
Where p.bup_id=b.bup_id;
```

To inquire about the author's name through the book's title:

```
Select p.pub_name
From buplisher p , book b
Where p.bup_id=b.bup_id and b.book_title='TROY_STORY';
```

To inquire about the title of the book through the name of its author:

```
select b.BOOK_TITLE
from book b ,   buplisher p
where p.bup_id=b.bup_id and p.bup_NAME='ZAID';
```

Inquire about the list of books, the names of their authors, and the number of copies available from these books:

```
select b.BOOK_TITLE , p.bup_name , c.copy_number
from book b,copy c , buplisher p
where b.BOOK_id=c.BOOK_id and p.bup_id=b.bup_id ;
```

Add new book to the library:

```
insert into book values(6,'TROY_STORY','13-NOV-2023',4,30);
```

Add a new reader to the database:

```
insert into reader
values(1,'AROWA','MALEK','ALI',730808090,'arowa21@gmail.com','TAIZ');
```

Perform a loan for a specific book for a specific user:

```
insert into borrow values(1,1,10,'05-NOV-2023','12-NOV-2023');
```

Delete a book from the database:

```
delete from book
where book_id=(select book_id from book where book_title='TROY_STORY');
```