4th Year CSE (2018/2019)
Database Systems

Library Database

Names:

فادي جورج عطا الله سليمان	43831
ديفيد نادر يعقوب جرجس	43779
جورج محسن رفعت فوزي	43768
كيرلس شريف هنري تادرس	43840

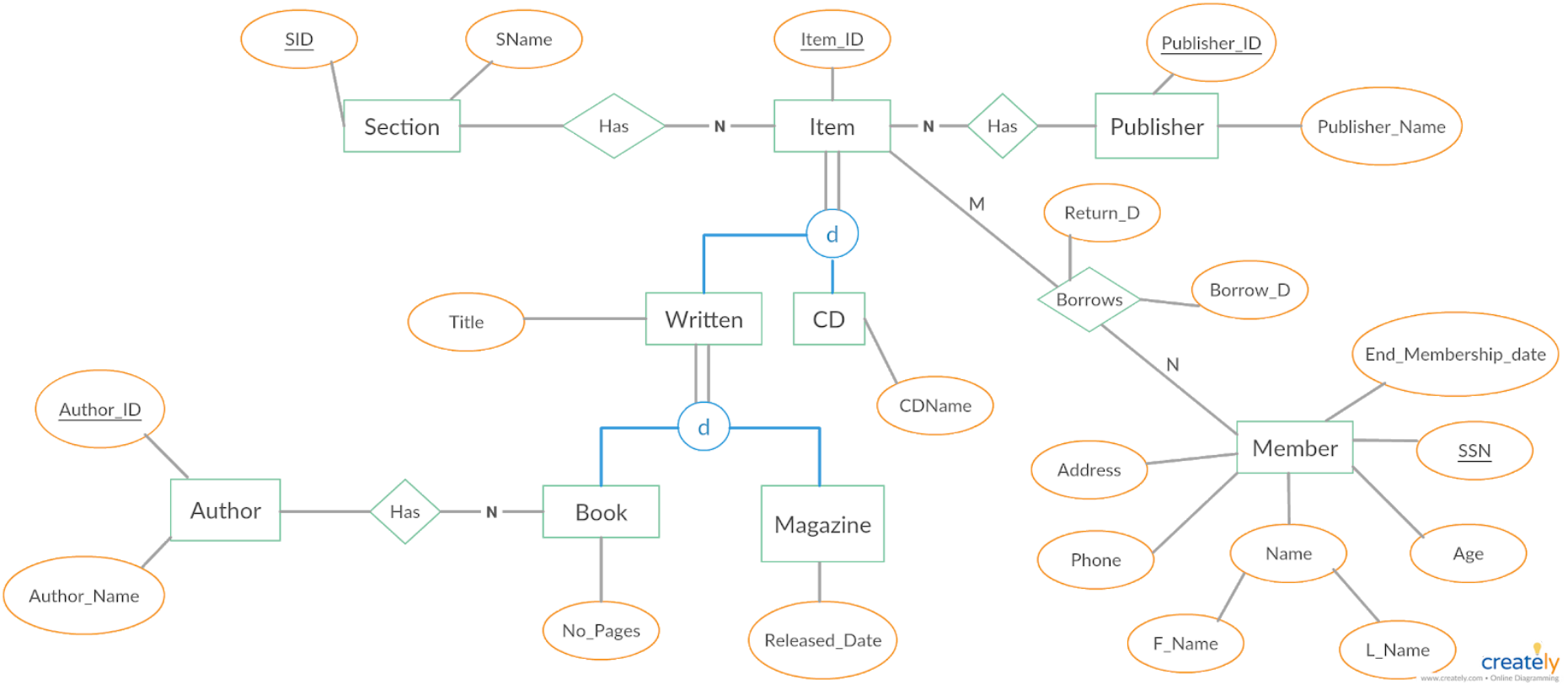
Description

- ❏ We Have designed a database for a library.
- ❏ Any Member in this library can borrow any of its items.
- ❏ Library Items are: CDs, Magazines and Books.
- ❏ Membership of the Library members has an end date.
- ❏ This Library stores in its database the list of authors of all its books,because many members choose the book to read according to its author.
- ❏ When a member borrows an item, we record the *borrow_date*, when they return the item, we record the *return_date*.
- ❏ The Library also store the publisher in its database.
- ❏ The Library divides its items into sections (e.g. Literature, Engineering, etc…) in order to allow members to find their interest in an easy way.

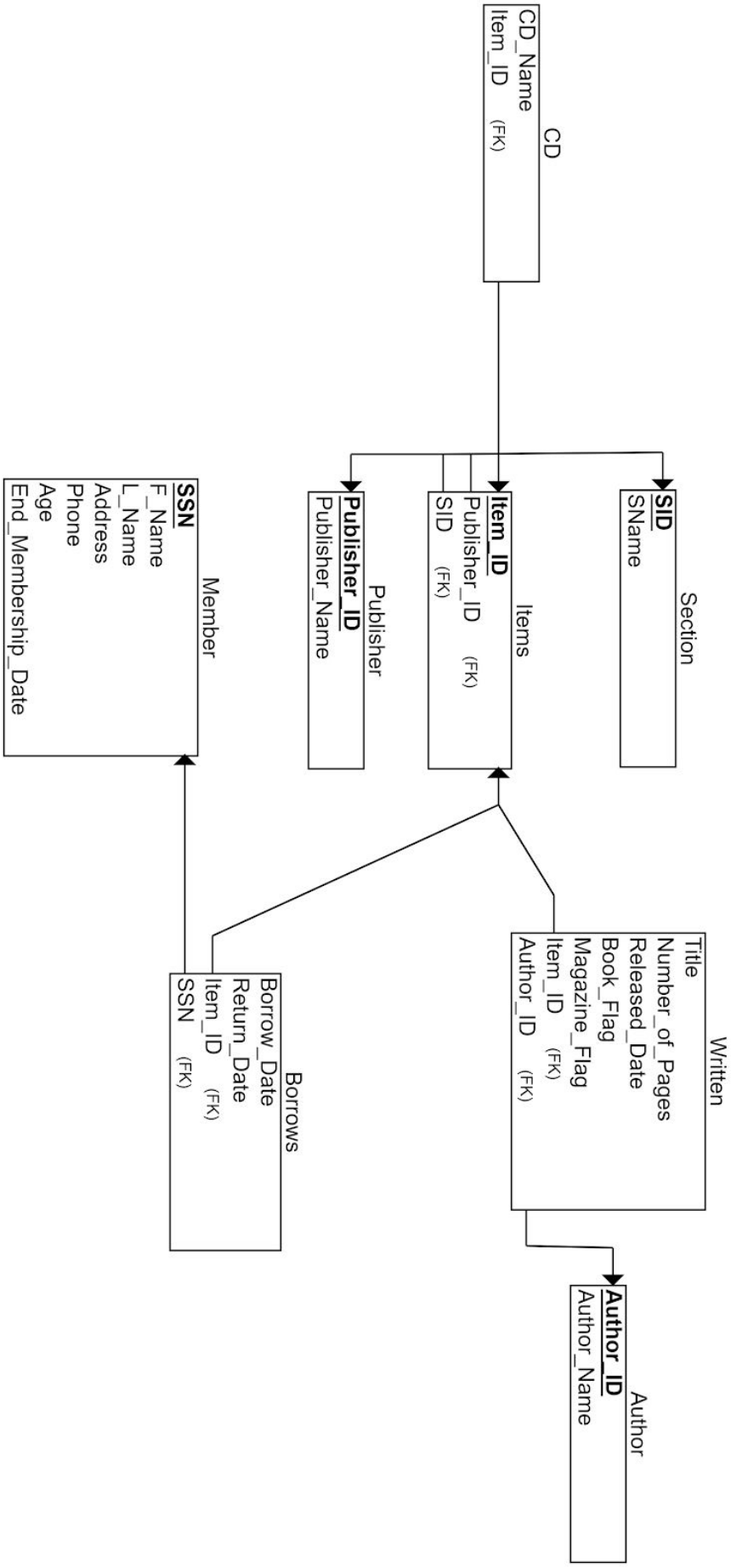
Assumptions:

- Each book is written by only one author.
- The library doesn’t have more than one copy of the same item.
- As long as the return date of an item is NULL, this item is borrowed by the member and he hasn’t returned it yet.

EER Diagram


SID
SName
Section
Has
N
Item
Item_ID
N
Has
Publisher
Publisher_ID
Publisher_Name
M
Return_D
Borrows
Borrow_D
N
End_Membership_date
Member
SSN
Address
Phone
Name
Age
F_Name
L_Name
d
Title
Written
CD
CDName
d
Author_ID
Author
Author_Name
Has
N
Book
No_Pages
Magazine
Released_Date
creately
www.creately.com • Online Diagramming

Relational Schema



Sample of Tables Creation

```
CREATE DATABASE BookShop
use BookShop

CREATE TABLE Publisher (
	ID 			INT 		NOT NULL,
	Publisher_Name 	VARCHAR(100) NOT NULL,
	PRIMARY KEY (ID)
);

CREATE TABLE Item(
	ID 		INT NOT NULL,
	Publisher_ID INT,
	Section_ID 	INT,
	FOREIGN KEY (Publisher_ID) REFERENCES Publisher (ID)
		ON UPDATE CASCADE
		ON DELETE RESTRICT,
	PRIMARY KEY (ID),
	FOREIGN KEY (Section_ID) REFERENCES Section (ID)
		ON UPDATE CASCADE
		ON DELETE RESTRICT
);

CREATE TABLE Borrows (
	SSN 		INT 	NOT NULL,
	Item_ID 	INT 	NOT NULL,
	Borrow_Date 	DATE 	NOT NULL,
	Return_Date 	DATE,
	PRIMARY KEY (SSN, Item_ID),
	FOREIGN KEY (SSN) REFERENCES Member (SSN)
		ON DELETE RESTRICT
		ON UPDATE CASCADE,
	FOREIGN KEY (Item_ID) REFERENCES Item (ID)
		ON DELETE RESTRICT
		ON UPDATE CASCADE,
);
```

```
CREATE TABLE Section (
ID INT NOT NULL,
SName VARCHAR(50) NOT NULL,
PRIMARY KEY (ID)
);


CREATE TABLE Publisher (
ID INT NOT NULL,
Publisher_Name VARCHAR(100) NOT NULL,
PRIMARY KEY (ID)
);
```

```
CREATE TABLE Member
(
    SSN INT NOT NULL,
    First_Name VARCHAR(50) NOT NULL,
    Last_Name VARCHAR(50),
    Phone VARCHAR(20),
    End_Membership_Date DATE,
    Age INT,
    Adr VARCHAR(120),
    PRIMARY KEY (SSN)
);
```

Sample of Data Insertion

```
INSERT INTO `Written`
(Item_ID,Title,Magazine_Flag,Released_Date,Book_Flag,Author_ID,NumberOfPages)
VALUES (1,'Gray''s Anatomy',NULL,NULL,1,5,1200);

INSERT INTO `Written`
(Item_ID,Title,Magazine_Flag,Released_Date,Book_Flag,Author_ID,NumberOfPages)
VALUES (2,'Calculus',NULL,NULL,1,6,900);

INSERT INTO `Section` (ID,SName) VALUES (1,'Engineering');
INSERT INTO `Section` (ID,SName) VALUES (2,'Medicine');

INSERT INTO `Member` (SSN,First_Name,Last_Name,Phone,End_Membership_Date,Age,Adr)
VALUES (111,'Amira','Nagy','01001234567','5/1/2019',26,'8 Main St.');

INSERT INTO `Item` (ID,Publisher_ID,Section_ID) VALUES (1,3,2);
INSERT INTO `Item` (ID,Publisher_ID,Section_ID) VALUES (2,2,1);
```

```
INSERT INTO `Borrows` (SSN,Item_ID,Borrow_Date,Return_Date) VALUES (111,1,'2018-12-01',NULL);
INSERT INTO `Borrows` (SSN,Item_ID,Borrow_Date,Return_Date) VALUES (111,6,'2018-07-05','2018-07-19');
INSERT INTO `Borrows` (SSN,Item_ID,Borrow_Date,Return_Date) VALUES (222,2,'2018-04-01','2018-04-10');
INSERT INTO `Borrows` (SSN,Item_ID,Borrow_Date,Return_Date) VALUES (222,3,'2018-11-25',NULL);
INSERT INTO `Borrows` (SSN,Item_ID,Borrow_Date,Return_Date) VALUES (333,6,'2018-11-1','2018-11-16');
INSERT INTO `Borrows` (SSN,Item_ID,Borrow_Date,Return_Date) VALUES (333,5,'2018-11-16','2018-11-26');
INSERT INTO `Borrows` (SSN,Item_ID,Borrow_Date,Return_Date) VALUES (333,4,'2018-26-11','2018-12-05');
INSERT INTO `Author` (ID,Author_Name) VALUES (2,'Taha Hussein');
INSERT INTO `Author` (ID,Author_Name) VALUES (3,'William Shakespeare');
INSERT INTO `Author` (ID,Author_Name) VALUES (4,'Charles Dickens');
INSERT INTO `Author` (ID,Author_Name) VALUES (5,'Henry Gray');
INSERT INTO `Author` (ID,Author_Name) VALUES (6,'James Stewart');
INSERT INTO `Author` (ID,Author_Name) VALUES (7,'Ramez Elmasri');
COMMIT;
```

	ID	Author_Name
1	2	Taha Hussein
2	3	William Shakespeare
3	4	Charles Dickens
4	5	Henry Gray
5	6	James Stewart
6	7	Ramez Elmasri

Author

	ID	Publisher_Name
1	1	Al Ahram
2	2	O'Reilly
3	3	Pearson
4	4	Penguin
5	5	Nahdet Masr
6	6	Disney

Publisher

	Item_ID	Title	Magazine_Flag	Released_Date	Book_Flag	Author_ID	NumberOfPages
1	1	Gray's Anatomy	*NULL*	*NULL*	1	5	1200
2	2	Calculus	*NULL*	*NULL*	1	6	900
3	3	Database Systems	*NULL*	*NULL*	1	7	1000
4	4	الأيام	*NULL*	*NULL*	1	2	600
5	5	Hamlet	*NULL*	*NULL*	1	3	550
6	6	Oliver Twist	*NULL*	*NULL*	1	4	500
7	7	میکی	1	2018-12-13	*NULL*	*NULL*	25
8	8	میکی	1	2018-12-20	*NULL*	*NULL*	25

Written

	SSN	First_Name	Last_Name	Phone	End_Membership_Date	Age	Adr
1	111	Amira	Nagy	01001234567	5/1/2019	26	8 Main St.
2	222	Adel	Zaki	01221234567	4/2/2019	22	13 Main Sq.
3	333	Ramy	Fouad	01991234567	29/12/2018	17	3 Green St.
4	444	Rania	Ramzy	01001234567	21/3/2019	12	9 Blue St.

Member

	ID	SName
1	1	Engineering
2	2	Medicine
3	3	Literature
4	4	Comics & Cartoons

Section

	Item_ID	CD_Name
1	9	Oliver Twist - The movie
2	10	Tom and Jerry

CD

	SSN	Item_ID	Borrow_Date	Return_Date
1	111	1	2018-12-01	*NULL*
2	111	6	2018-07-05	2018-07-19
3	222	2	2018-04-01	2018-04-10
4	222	3	2018-11-25	*NULL*
5	333	6	2018-11-1	2018-11-16
6	333	5	2018-11-16	2018-11-26
7	333	4	2018-26-11	2018-12-05

Borrows

	ID	Publisher_ID	Section_ID
1	1	3	2
2	2	2	1
3	3	2	1
4	4	1	3
5	5	4	3
6	6	4	3
7	7	5	4
8	8	5	4
9	9	6	4
10	10	6	4

Item

Reports

1. Get all books, and the corresponding author, publisher and section

```
select  title, author_name, sname, publisher_name
from    item
        join written    on item.id = written.item_Id
        join author     on author.id = written.author_id
        join section    on section.id = item.section_id
        join publisher  on publisher.id = item.publisher_id
```

	Title	Author_Name	SName	Publisher_Name
1	Gray's Anatomy	Henry Gray	Medicine	Pearson
2	Calculus	James Stewart	Engineering	O'Reilly
3	Database Systems	Ramez Elmasri	Engineering	O'Reilly
4	الأيام	Taha Hussein	Literature	Al Ahram
5	Hamlet	William Shakespeare	Literature	Penguin
6	Oliver Twist	Charles Dickens	Literature	Penguin

---

2. Get total number of books in each section

```
select      sname, count(*) AS "Number of books"
from        item
            join written on item.id = written.item_id
            join section on item.section_id = section.id
where       written.book_flag IS NOT NULL
group by    sname
```

	SName	Number of books
1	Engineering	2
2	Literature	3
3	Medicine	1

---

3. Get list of unreturned books, and the member who borrowed them

```
select  title, borrow_date, first_name, last_name, phone
from    item
        join written on (item.id = written.item_id)
        join borrows on (item.id = borrows.item_id)
        join member  on (member.ssn = borrows.ssn)
where   written.book_flag is not null and
        borrows.return_date is null
```

	Title	Borrow_Date	First_Name	Last_Name	Phone
1	Gray's Anatomy	2018-12-01	Amira	Nagy	01001234567
2	Database Systems	2018-11-25	Adel	Zaki	01221234567

---

4. Get members who never borrowed a book

```
select  *
from    member
where   not exists (select  *
                    from    borrows
                    where   member.ssn = borrows.ssn)
```

	SSN	First_Name	Last_Name	Phone	End_Membership_Date	Age	Adr
1	444	Rania	Ramzy	01001234567	21/3/2019	12	9 Blue St.

---

5. Get sections that has less than 2 books

```
select      sname, count(*)
from        section
            join item    on item.section_id = section.id
            join written on item.id = written.item_id
where       written.book_flag is not NULL
group by    sname
having      count(*) < 2
order by    count(*)
```

	SName	count(*)
1	Medicine	1

---

6. Get total number of borrows for each member (including members who never borrowed a book)

```
select      first_name, last_name, count(borrows.ssn) as "Number of borrows"
from        member
            left outer join borrows on member.ssn = borrows.ssn
group by    member.ssn
order by    "Number of borrows" DESC
```

	First_Name	Last_Name	Number of borrows
1	Ramy	Fouad	3
2	Amira	Nagy	2
3	Adel	Zaki	2
4	Rania	Ramzy	0