University Library Management
May 23, 2023


Faculty of Computers and Artificial Intelligence

# IS211 Database Systems I - 2023

## University Library Management System


| Name | ID |
|---|---|
| Aya Ali Hassan Ali | 20210083 |
| Khadija Ayman Elsayed Eshra | 20210540 |
| Sara Tamer Bihery | 20210155 |
| Sara Walid Farouk Mohamed | 20210495 |
| Sohaila Abdelazim Khalifa | 20210492 |


Section: 29,30

Eng. Nancy Saeed

# Table of contents

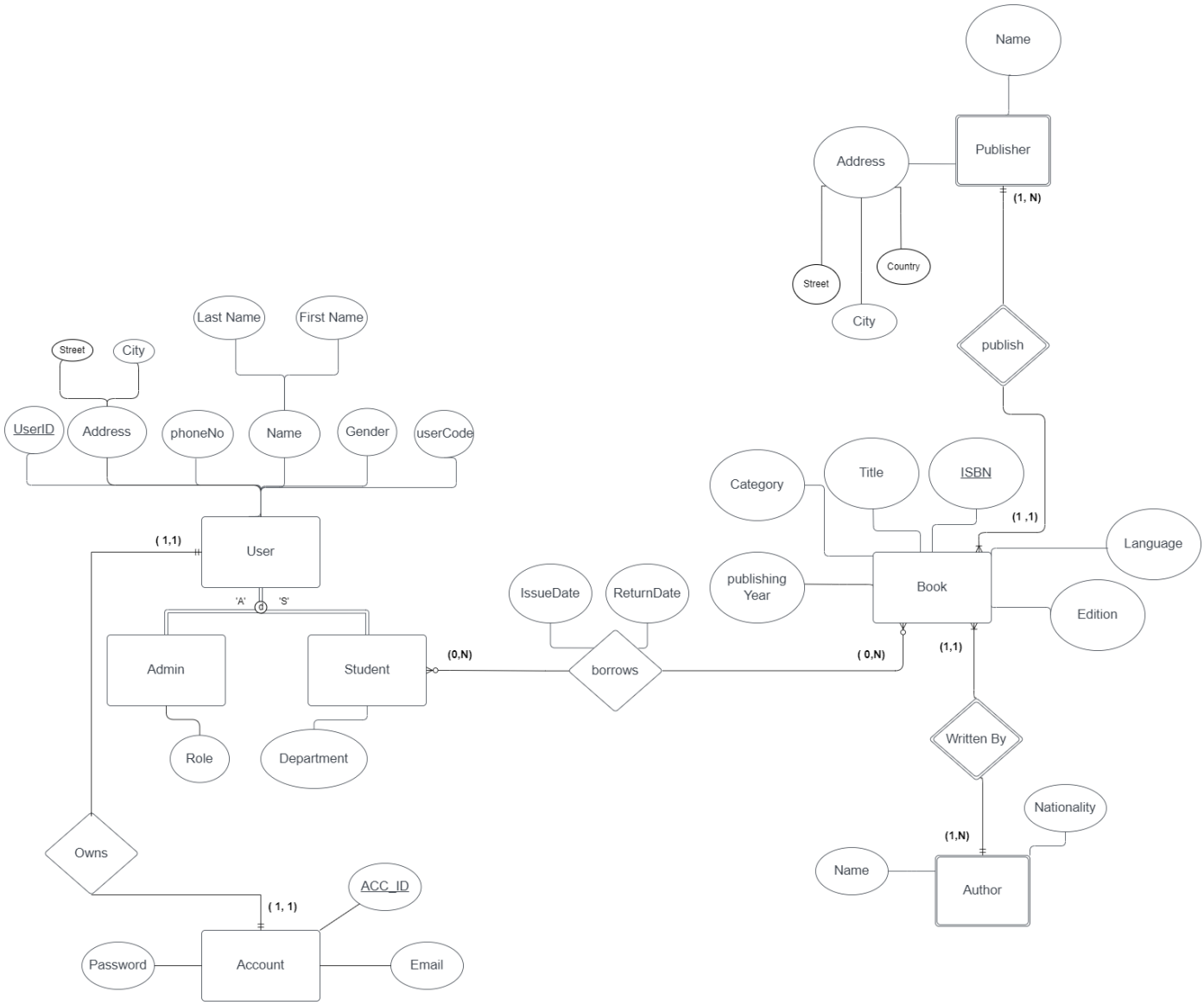
# Introduction

In this system we are trying to simulate a University Library System by providing a system database with different functionalities such as add, delete, update and display information, about the system entities: admin, student, book. Also, the user can search for any book that satisfy certain criteria, such as search for certain author to see all his books. The system also generates reports when used by an admin to view several statistics such as the number of students using the system, number of the books available in the library. Moreover, it shows the number of students in each department and number of books grouped and ordered by publishing year. Our project provides these functions in friendly interface which allows the user to act easily with.

# Assumptions

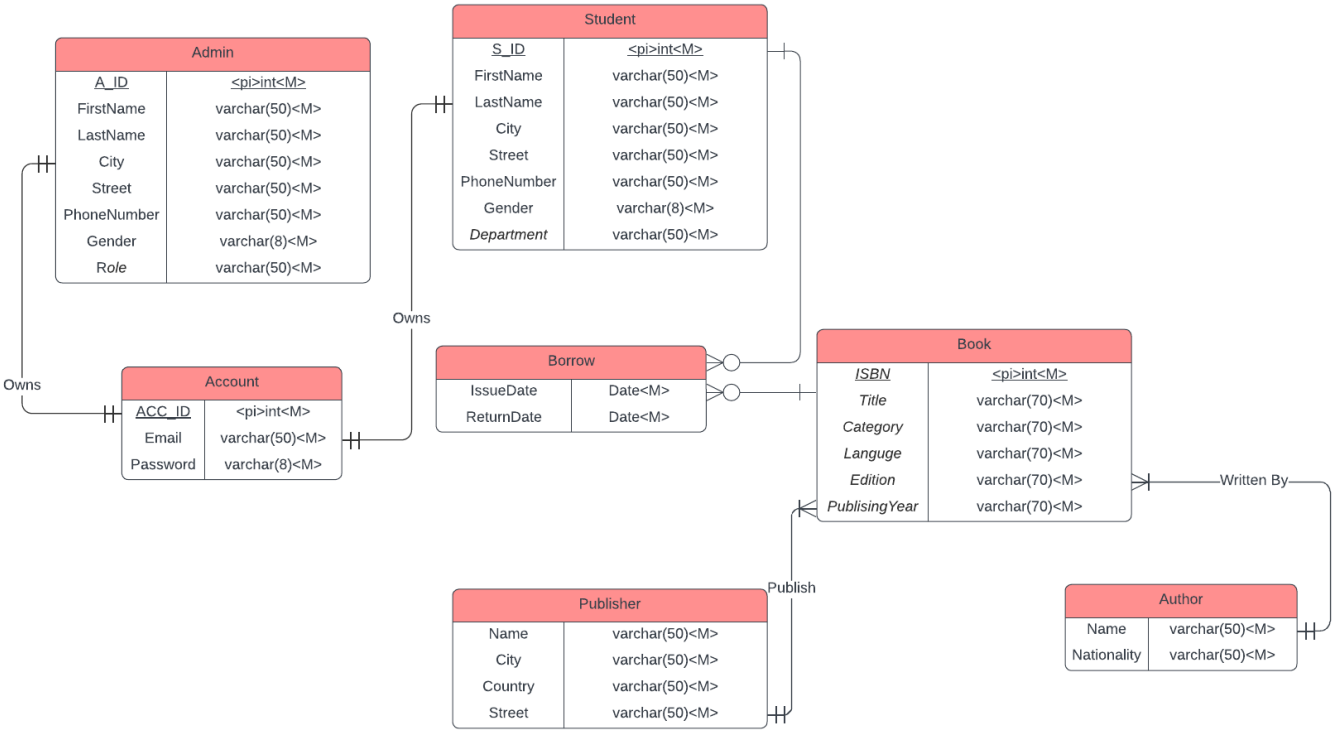
1. A Book is written by only one author.
2. An author writes many books.
3. A Book is published by only one publisher.
4. A Publisher can publish many books.
5. An Admin can own only one Account.
6. A Student can own only one Account.
7. Student can borrow many Books.
8. A Book could be borrowed by many Students.
9. Student & Admin login with their ID provided by University.

# ERD

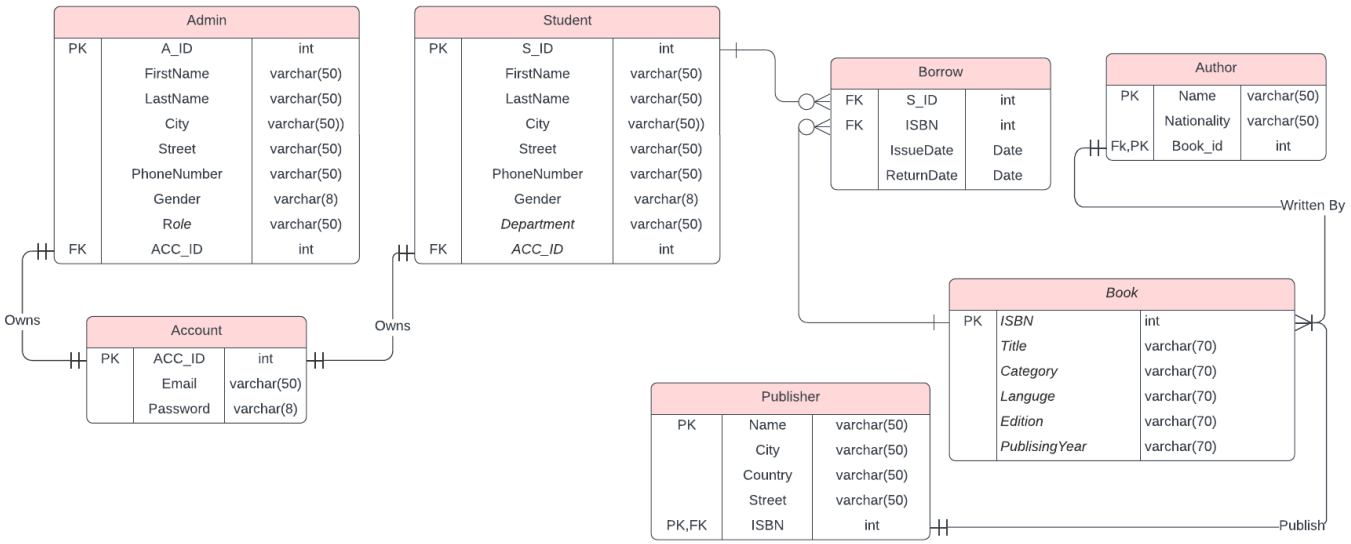



For better Visualization please refer to:
https://shorturl.at/pHU04

# Conceptual Diagram




# Physical Diagram




For better Visualization please refer to:

https://shorturl.at/kWZ47

# SQL Queries

Attached as: UniveristyLibarary.sql

UniveristyLibarary.sql

```
CREATE TABLE Account (
  ACC_ID int AUTO_INCREMENT primary key,
  Email varchar(50),
  Password varchar(8)
);
CREATE TABLE Admin (
  A_ID int,
  FirstName varchar(50),
  LastName varchar(50),
  City varchar(50),
  Street varchar(50),
  PhoneNumber varchar(50),
  Gender varchar(8),
  Role varchar(50),
  ACC_ID int,
  PRIMARY KEY (A_ID),
  CONSTRAINT FK_Account
  FOREIGN KEY (ACC_ID)
  REFERENCES Account(ACC_ID)
```

```
  ON DELETE CASCADE
  ON UPDATE CASCADE
);
CREATE TABLE Book (
  ISBN int,
  Title varchar(70),
  Category varchar(70),
  Languge  varchar(70),
  Edition varchar(70),
  PublisingYear varchar(70),
  PRIMARY KEY (ISBN)
);
CREATE TABLE Student (
  S_ID int,
  FirstName varchar(50),
  LastName varchar(50),
  City varchar(50),
  Street varchar(50),
  PhoneNumber varchar(50),
  Gender varchar(8),
  Department varchar(50),
  ACC_ID int,
  PRIMARY KEY (S_ID),
  CONSTRAINT FK_SAccount
  FOREIGN KEY (ACC_ID)
  REFERENCES Account(ACC_ID)
```

```
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

CREATE TABLE Publisher (
  Name varchar(50),
  City varchar(50),
  Country varchar(50),
  Street varchar(50),
  ISBN int,
CONSTRAINT P_Book
  FOREIGN KEY (ISBN)
  REFERENCES Book(ISBN)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  PRIMARY KEY (Name, ISBN)
);

CREATE TABLE Author (
  Name varchar(50),
  Nationality varchar(50),
  ISBN int,
CONSTRAINT A_Book
  FOREIGN KEY (ISBN)
  REFERENCES Book(ISBN)
    ON DELETE CASCADE
```

```
    ON UPDATE CASCADE,
  PRIMARY KEY (Name, ISBN)
);
CREATE TABLE Borrow (
  S_ID int,
  ISBN int,
  IssueDate Date,
  ReturnDate Date,
  CONSTRAINT S_Borrow
  FOREIGN KEY (S_ID)
 REFERENCES Student(S_ID)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT B_Borrow
  FOREIGN KEY (ISBN)
  REFERENCES Book(ISBN)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  PRIMARY KEY (S_ID, ISBN)
);
```

# SQL quarries to test the program:

File has attached as: SQL insert.txt

SQL insert.txt

# System requirements & Functions Proposed

1. 2 insert statements on 2 different tables:
    - Insert new student in Student table.
    - Insert new book in Book Table.
    - Insert new admin in Admin Table.
    - Insert new Account in Account Table.
    - Insert new Author in Author Table.
2. 2 delete statements on 2 different tables (with conditions).
    - Delete student from Student table.
    - Delete book from Book table. (Using on delete cascade)
3. 2 update statements on 2 different tables (with condition).
    - Update Student data.
    - Update Admin data.
    - Update book information.
4. Select data from any table(s) of the database.
    - Select books and display them.
5. Select data that involves more than one table of the database (using joins).
    - Displays the author with all books he wrote.
    - When someone makes login need to check two different tables to know whose account belongs to.
6. Meaningful report – by using aggregate function.
    1. Number of Books.
    2. Number of Admins.
    3. Number of Students.
    4. Number of Students in each department.
    5. Number of Books published in each year.

# Bonus

- Included Weak Entity.
- Generated 1 meaningful report.
- Implemented GUI.

# Tools

- Drawio: draw.io (drawio.com)
- Lucid: https: Documents (lucid.app)
- My SQL
- IntelliJ idea (java)

# Program reference

File has attached as: 20210492_20210083_20210540_20210155_20210495.file

20210492_20210083_20210540
_20210155_20210495