

Adatbázisrendszerek I. Assignment

Orosz Péter - WO02D7

November 27, 2022

Contents

1	Introduction	1
2	ER Diagram	2
3	RS Diagram	3
4	SQL	4
4.1	Database Creation	4
4.2	Fill Database	6
4.3	Alter and Update Commands	6
4.4	Select Commands	6

1 Introduction

For my assignment i have chosen to create a library management system for a university. The database will contain the staff members with the following attributes:

- Staff ID (Primary Key)
- Name (First Name, Last Name)
- Phone Number
- Birthday (Year, Month, Day)
- Age (Derived)

The staff will have a mandatory user name and password which they can use to log into the software that handles all the books in the library. They will be able to process reserves and returns in this software.

Every staff member will handle multiple students with the following attributes:

- Student ID (Primary Key)
- Name (First Name, Last Name)
- Phone Number (Multiple)
- Email (Multiple)
- Address (Postal Code, City, Street)
- Birthday (Year, Month, Day)
- Age (Derived)

One student can reserve multiple books at the same time. The reserve and return dates will have to be stored in an archive, so staff can check who reserved the book in the past.

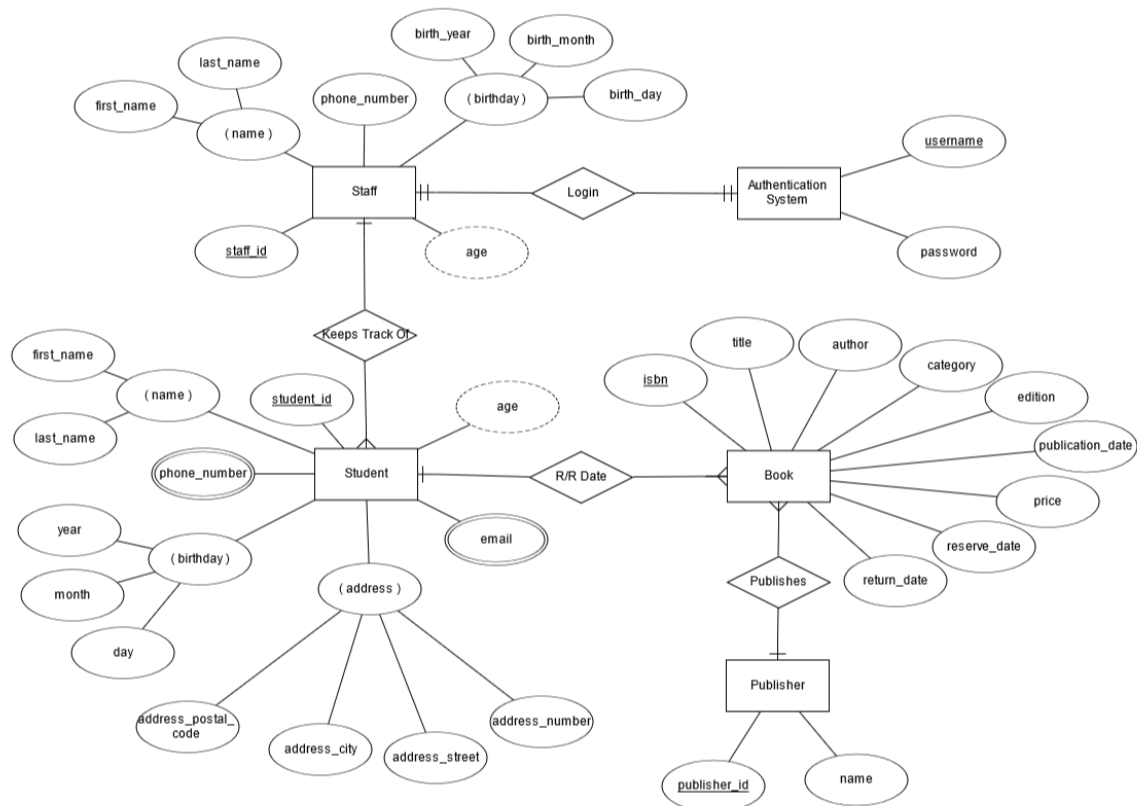
A books attributes:

- ISBN (Primary Key)
- Author
- Title
- Category
- Price
- Edition

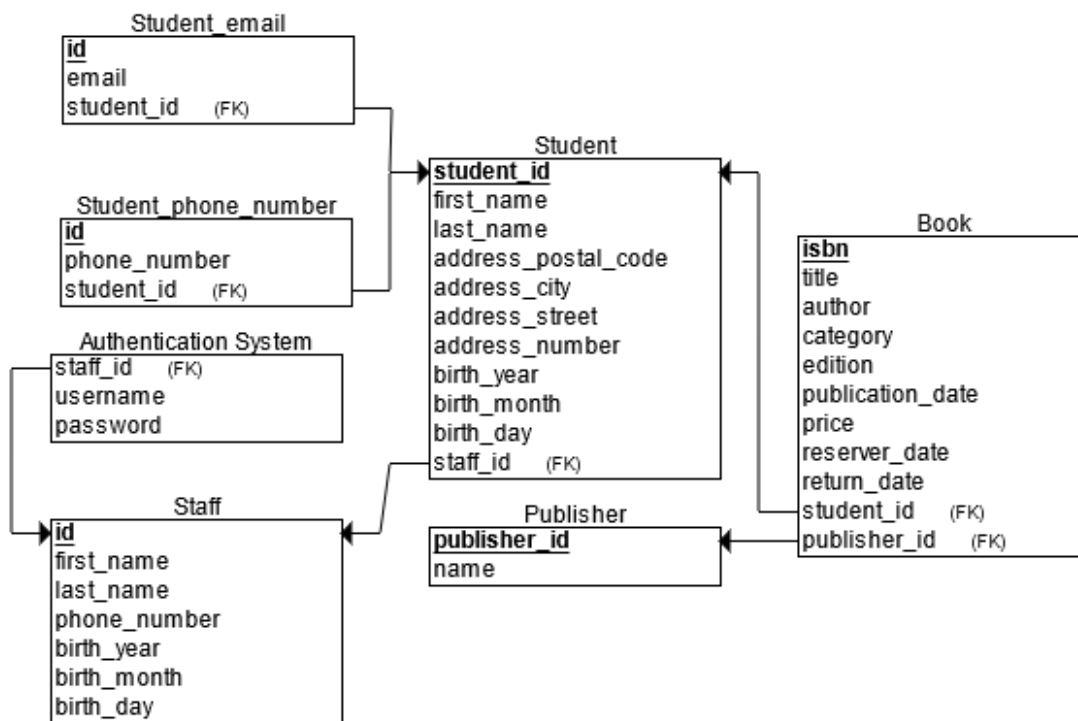
The books publishers attributes:

- Publisher ID (Primary Key)
- Year of Publication
- Name

2 ER Diagram



3 RS Diagram



4 SQL

The creation of the database was executed in Oracle.

4.1 Database Creation

```
CREATE TABLE Staff(  
    id NUMBER(8,0) NOT NULL PRIMARY KEY,  
    first_name VARCHAR(255) NOT NULL,  
    last_name VARCHAR(255) NOT NULL,  
    phone_number VARCHAR(255) NOT NULL,  
    birth_year NUMBER(4,0) NOT NULL,  
    birth_month NUMBER(2,0) NOT NULL,  
    birth_day NUMBER(2,0) NOT NULL  
);  
  
CREATE TABLE Student(  
    id NUMBER(8,0) NOT NULL PRIMARY KEY,  
    first_name VARCHAR(255) NOT NULL,  
    last_name VARCHAR(255) NOT NULL,  
    address_postal_code VARCHAR(255) NOT NULL,  
    address_city VARCHAR(255) NOT NULL,  
    address_street VARCHAR(255) NOT NULL,  
    address_number VARCHAR(255) NOT NULL,  
    birth_year NUMBER(4,0) NOT NULL,  
    birth_month NUMBER(2,0) NOT NULL,  
    birth_day NUMBER(2,0) NOT NULL,  
    staff_id NUMBER(8,0) NOT NULL,  
    FOREIGN KEY (staff_id) REFERENCES Staff(id)  
);  
  
CREATE TABLE Publisher(  
    id NUMBER(1,0) NOT NULL PRIMARY KEY,  
    name VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE Authentication_System(  
    staff_id NUMBER(8,0) NOT NULL PRIMARY KEY,  
    username VARCHAR(255) NOT NULL,  
    password VARCHAR(255) NOT NULL,  
    FOREIGN KEY(staff_id) REFERENCES Staff(id)  
);  
  
CREATE TABLE Student_email(  
    id NUMBER(8,0) NOT NULL PRIMARY KEY,  
    student_id NUMBER(8,0) NOT NULL,  
    email VARCHAR(255),  
    FOREIGN KEY(student_id) REFERENCES Student(id)  
);  
  
CREATE TABLE Student_phone_number(  
    id NUMBER(8,0) NOT NULL PRIMARY KEY,  
    student_id NUMBER(8,0) NOT NULL,  
    phone_number VARCHAR(255) NOT NULL,  
    FOREIGN KEY(student_id) REFERENCES Student(id)  
);  
  
CREATE TABLE Book(  
    isbn VARCHAR(255) NOT NULL PRIMARY KEY,  
    title VARCHAR(255) NOT NULL,  
    author VARCHAR(255) NOT NULL,  
    category VARCHAR(255) NOT NULL,  
    edition VARCHAR(255) NOT NULL,  
    publication_year NUMBER(4,0) NOT NULL,  
    price INT NOT NULL,  
    student_id NUMBER(8,0) NOT NULL,  
    publisher_id NUMBER(1,0) NOT NULL,  
    FOREIGN KEY(student_id) REFERENCES Student(id),  
    FOREIGN KEY(publisher_id) REFERENCES Publisher(id)  
);
```

4.2 Fill Database

I filled the tables with the INSERT command like this:

```
INSERT INTO table VALUES(id, ...);
```

Some examples from the staff database:

```
INSERT INTO Staff VALUES(98017604, 'Axel', 'Nieri', '615-258-7302', 1968, 04, 16);
INSERT INTO Staff VALUES(73267121, 'Red', 'Lowe', '715-993-3449', 1982, 04, 27);
INSERT INTO Staff VALUES(26991745, 'Jelena', 'Kearney', '606-857-8284', 1953, 10, 01);
INSERT INTO Staff VALUES(37065937, 'Jockel', 'Lu', '321-967-3130', 1979, 12, 30);
INSERT INTO Staff VALUES(25187152, 'Sverrir', 'Gerhard', '303-892-6655', 1980, 11, 22);
```

The full list of commands used to fill up the database can found in the seperate .sql file.

4.3 Alter and Update Commands

Updates the wrong book title in the book table

```
UPDATE Book SET Book.title = 'Defender of dawn' WHERE Book.title = 'Fefender of dawn';
```

Update the wrong book category in the book table

```
UPDATE Book SET category='Adventure' WHERE category='Adcenture';
```

Adds the non mandatory foundation date for each publisher

```
ALTER TABLE Publisher ADD founding_date date;
```

4.4 Select Commands

Which students reserved the book titled: Game of wind?

```
SELECT Student.first_name, Student.last_name FROM Student JOIN Book
ON Student.id = Book.student_id WHERE Book.title = 'Game of wind';
```

How many students does each staff member handle?

```
SELECT Staff.id, COUNT(Student.id) FROM Staff JOIN Student
ON Staff.id = Student.staff_id GROUP BY Staff.id;
```

Distinct books in library and how many of each?

```
Select Book.title, COUNT(Book.title) FROM Book GROUP BY Book.title;
```

Not reserved books.

```
Select Book.isbn, Book.title FROM Book WHERE Book.reserve_date IS NULL;
```


How many books are reserved by each student.

```
SELECT Book.student_id, COUNT(*) "Number of Books" FROM Book GROUP BY Book.student_id;
```

Number of books of each publisher.

```
SELECT Publisher.name, COUNT(Book.publisher_id) FROM Publisher JOIN Book  
ON Publisher.id = Book.publisher_id GROUP BY Publisher.name;
```

Not reserved books with adventure category

```
SELECT Book.isbn, Book.title FROM Book WHERE  
Book.reserve_date IS NULL AND Book.category = 'Adventure';
```

Students older than 18

```
SELECT Student.first_name, Student.last_name, Student.birth_year  
FROM Student WHERE Student.birth_year < 2004
```