

**Coursework**

**Faculty:** Mathematics and Informatics

**Specialty:** Informatics

**Discipline:** Databases

**Produced: Nikola** Atanasov Hadzhiev

**Subject:** Book Library Database

**Lecturer in discipline**:

Assc. Tsvetanka Lyubomirova Georgieva-Trifonova MD

**Faculty No:**  2009012027

Content

[1. Description of the subject area 2](#_Toc104328910)

[2. ER model based on 3](#_Toc104328911)

[3 . Database relation model 4](#_Toc104328912)

[4 . Realization of the relationship model of the database 5](#_Toc104328913)

[5. Create queries 6](#_Toc104328914)

[6. Create the necessary : 7](#_Toc104328915)

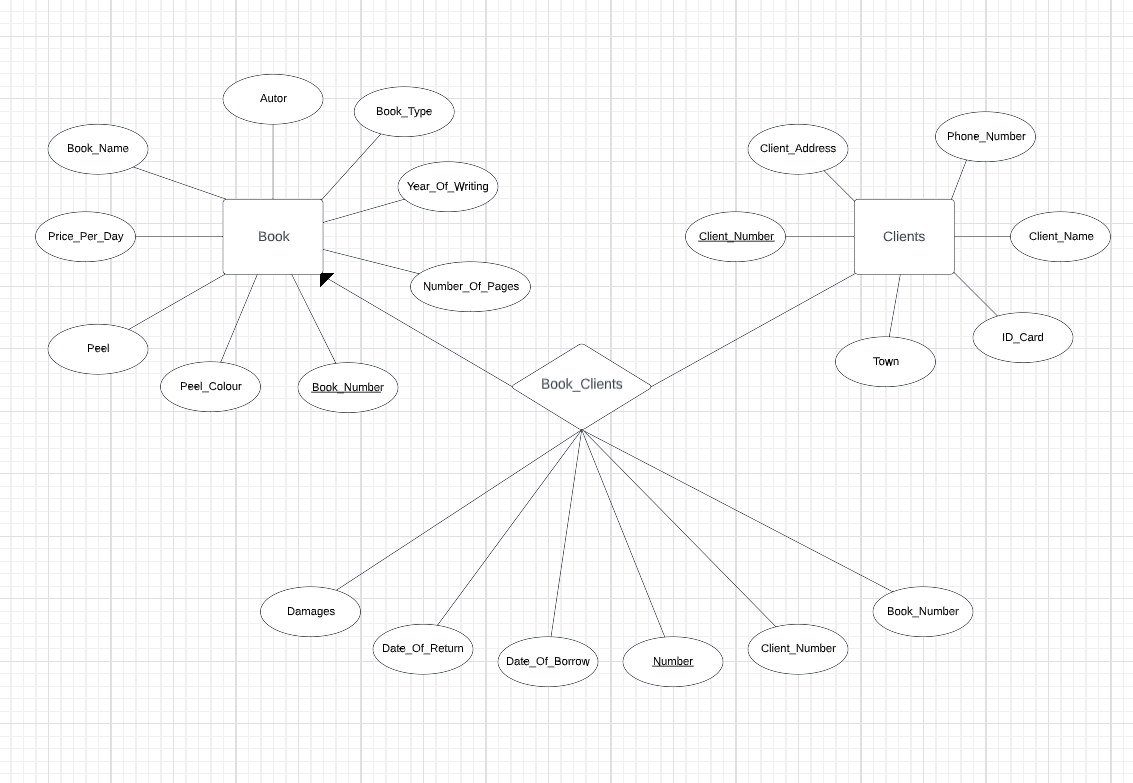
[7. Future work 8](#_Toc104328916)

## 1. Description of the subject matter

Introducing an SQL database into the library activity provides easier book store management and faster access to the necessary information. Working with readers is facilitated and accelerated. The database will assist and facilitate staff in searching, finding and entering certain information.

The database stores information about a library of books (name, author, book type, book year, number of pages, rental price, cover type and color), reader's card (name and surname of the particular person, personal identification number, city, address and phone number).

## 2. ER model based on



The objects in the database are: book, clients, and book\_clients.

The *book object* has *Book\_Number* attributes – primary key, *Book\_Name* – book name, *Autor* – a tuesday, *Book\_Type* – book type, *Year\_Of\_Writing* – year of writing, *Number\_Of\_Pages* – number of pages,  *Price\_Per\_Day* – price per day, *Peel* – cover*, Peel\_Color* – cover color.

The *client object* has *Client\_Number* attributes – primary key, *Clients\_Name* – *client name*, ID\_Card – personal number, *Town* – city, *Client\_Address* – address, *Phone\_Number* – phone number.

Thereare *book\_clients number attributes* –  *primary* key, *Book\_Number* – primary key, *Client\_Number* – primary key, *Date\_Of\_Borrow* – pickup date, *Date\_Of\_Return* – return date, *Damages* – damage.

The  *book* and *clients* are connected via a "many-to-one" link  *via a book\_clients* link (one book can be occupied by one user and one user may have occupied several bookies).

## 3 . Relational model of the database

The database consists of 3 tables: *Books*, *Clients* and *Book\_clients*.

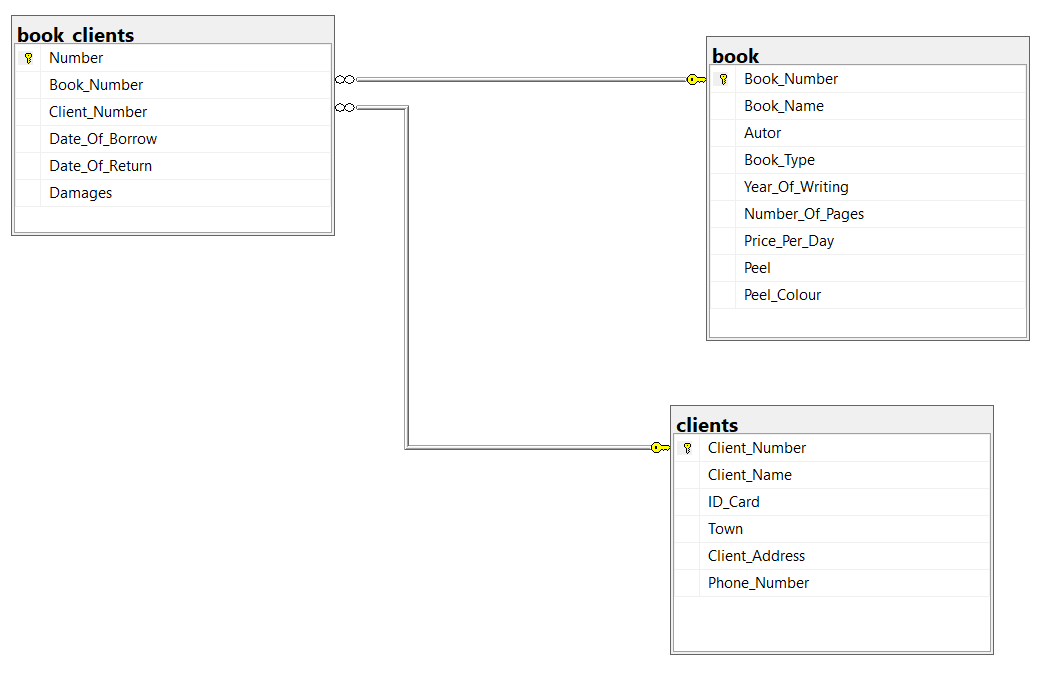
The  *Books table* consists of 9 columns: *Book\_Number*–  *primary key of type INT*, *Book\_Name* of  *type VARCHAR(30), Author of type VARCHAR(30)*, *Book\_Type of type VARCHAR(30), Year\_Of\_Writing* of  *type INT* *, Number\_Of\_Pages int type, Price\_Per\_Day of type MONEY, Peel of type VARCHAR(20)*  *and Peel\_Colour of type VARCHAR(20)*.

The  *Clients* table consists of 5 columns: *Client\_Number* – primary key o t *type INT*, *Client\_Name of type VARCHAR(80)*, *ID\_Card of type INT, Town of type VARCHAR(40), Client\_Address* of type *varchar(80)* and *Phone\_Number* of type *VARCHAR(50)*.

The Book\_ *Clients table consists*  *of 6* columns: *Number* –  *PRIMARY KEY OF TYPE INT*, *Book\_Number*  *of type INT, Client\_Number of type INT, Date\_Of\_Borrow of type DATE, Date\_Of\_Return of type DATE* and *Damages* of type  *VARCHAR(50).*

A link was *made*  *between Book\_Clients* books tables using the *primary key Book\_Number*  *the Book table and* the *column Book\_Number* of table *Book\_Clients.*

A link was *made*  *between the Clients Book\_Clients* tables using the primary  *key Client\_Number* the *Clients table* and the *Client\_Number* column in table *Book\_Clients.*



## 4 . Realization of the relationship model of the database

-- Createa book (Books)

CREATE TABLE book (

Book\_Number int PRIMARY KEY IDENTITY (1,1),

Book\_Name varchar (30) NOT NULL,

Autor varchar (30) NOT NULL,

Book\_Type varchar (30) NOT NULL,

Year\_Of\_Writing INT NOT NULL,

Number\_Of\_Pages INT NOT NULL,

Price\_Per\_Day MONEY NOT NULL,

Peel varchar (20) NOT NULL,

Peel\_Colour varchar (20) NOT NULL

)

-- Filledin on the data "book" table

INSERT INTO book (Book\_Name,Autor,Book\_Type,Year\_Of\_Writing,Number\_Of\_Pages,Price\_Per\_Day,Peel,Peel\_Colour)

VALUES ("Under theIgo," "Ivan Vazov," "novel," 1894, 431, 3, "hard cover," "red"),

("1885, 128, 5, "hard cover", "green"),

("Tobacco", "Dimitar Dimov", "novel",1951, 616, 2, "soft cover","green"),

("The Iron Lamp" , "Dimitar Talev" , "roman",1954, 338, 3, "soft cover","green"),

("The Prespa Bells," "Dimitar Talev," "Roman," 1954, 486, 4, "trda cover," "yellow"),

('Bai Gaño', 'Aleko Konstantinov', 'narrative', 1895, 176, 4, 'soft cover', 'red'),

("To Chicago and Back," "Aleko Konstantinov," "novel," 1894, 120, 2, "soft cover," "red")

-- Create the"clients" table

CREATE TABLE clients (

Client\_Number int PRIMARY KEY IDENTITY (1,1),

Client\_Name varchar (80) NOT NULL,

ID\_Card INT NOT NULL,

Town varchar (40) NOT NULL,

Client\_Address varchar (80) NOT NULL,

Phone\_Number varchar (50) NOT NULL

)

-- Filledin on the table "clients" with data

INSERT INTO clients (Client\_Name, ID\_Card, Town, Client\_Address, Phone\_Number)

VALUES ("Stefan Marinov", 582179286, "Varna", "Brothers Miladinovi Str." No1","0895/465-345"),

("Stefani Todorova", 986729123, "Plovdiv", "Street". 9,'0893/312-561',

("Petar Ivanov", 771354361, "Veliko Tarnovo", "Blvd." 22 Nikola Gabrovski Str., '0894/111-222'),

("Nikola Zhekov", 954897653, "Varna", "Ul." 5,'0899/465-345',

("Ivan Valentinov", 911852041, "Sofia", "Blvd." 1,'0892/483-859',

("Asparus Nicodemov", 801857093, "Plovdiv", "Ul." 77 Vasil Levski Str., '0878/345-999'),

("Ivaylo Shopov", 756349719, "Sofia", "Ul." 4,0894/532-112')

-- Creating the "book\_clients" (Book Clients)

CREATE TABLE book\_clients(

Number int PRIMARY KEY IDENTITY(1,1),

Book\_Number INT NOT NULL,

Client\_Number INT NOT NULL,

Date\_Of\_Borrow DATE NOT NULL,

Date\_Of\_Return DATE NOT NULL,

Damages varchar(50) NOT NULL

)

-- Filledin on the table "book\_clients" with data

INSERT INTO book\_clients (Book\_Number,Client\_Number,Date\_Of\_Borrow,Date\_Of\_Return,Damages)

VALUES (1,1, '2022-03-21', '2022-03-23', 'no damage'),

(7,2,'2022-03-25','2022-04-02','no damage'),

(4,5,'2022-02-26','2022-03-02','no damage'),

(3,7,'2022-03-20','2022-03-22','there are damage'),

(2,4,'2022-03-01','2022-03-02','no damage'),

(5,3,'2022-03-25','2022-03-02','there are damage'),

(6,6,'2022-03-25','2022-03-02','there are damage')

## 5. Create queries

-- Query that displays the order by age and removes recurrences

SELECT DISTINCT Year\_Of\_Writing

FROM book

-- Query that displays the information from the three columns arranged by number

SELECT Client\_Number, ID\_Card, Town

FROM clients

ORDER BY Client\_Number

-- A query that displays the names of customers who start with the letter "I" and their phone number

SELECT Client\_Name, Phone\_Number

FROM clients

WHERE Client\_Name LIKE 'And%'

-- A request that displays all the information about the customers and books that have been taken

SELECT book.\*, clients.\*

FROM clients, book\_clients, book

WHERE book. Book\_Number = book\_clients. Book\_Number AND /connection between

clients. Client\_Number = book\_clients. Client\_Number primary keys/

-- A request that shows the name and phone of customers who have taken a book more expensive than 3 BGN per day

SELECT Client\_Name, Phone\_Number

FROM clients, book\_clients, book

WHERE book. Book\_Number = book\_clients. Book\_Number AND

clients. Client\_Number = book\_clients. Client\_Number AND

Price\_Per\_Day > 3

-- Query that groups books by author and creates a new column showing the number of books by that author in the table

SELECT Autor, COUNT(\*) AS [Number of books] FROM book

GROUP BY Autor

HAVING Autor = 'Ivan Vazov';

## 6. Establishment of the necessary:

- views:

-- Create a view that displays the names of all books written after 1900.

CREATE VIEW [BooksAfter1900] AS

SELECT Book\_Name, Year\_Of\_Writing

FROM book

WHERE Year\_Of\_Writing > 1900;

-check

SELECT\* FROM BooksAfter1900

- stored procedures and/or user defined functions:

-- Create a procedure that displays all books that, for a particular type of

CREATE PROCEDURE SelectAllBook @Book\_Type varchar(30)

AS

SELECT \* FROM book WHERE Book\_Type = @Book\_Type

GO

-check

EXEC SelectAllBook @Book\_Type = 'novel';

- triggers:

-- Trigger that disables the deletion of customers from the table containing user data

CREATE TRIGGER No\_DeleteClients

ON clients

INSTEAD OF DELETE

AS

IF @@ROWCOUNT = 0 RETURN

RAISERROR('Deleting rows in the table is not allowed.' , 8, 1);

-check

DELETE clients

WHERE Client\_Number = 1

## 7. Future work

Future improvements would also include implementing a book-buying option. The possibility of home delivery can be added. The database could be used as a basis for the development of a web application for the sale of books in electronic form.