

Тема „База от данни за книжарница“
Изготвил: Акага Павлова, ФН: 45750

1. Обхват на модела. Дефиниране на задачата

Информационна система, съхраняваща данните за книгите в една книжарница.

Всяка книга има име, година на издаване, уникален идентификационен номер и номер на издателската къща, от която е била издадена. Всяка такава има жанр, за всеки, от който се пази уникално име, като една книга може да попада в повече от една жанрова категория и във всеки жанр може да се включват 0, 1 или повече книги.

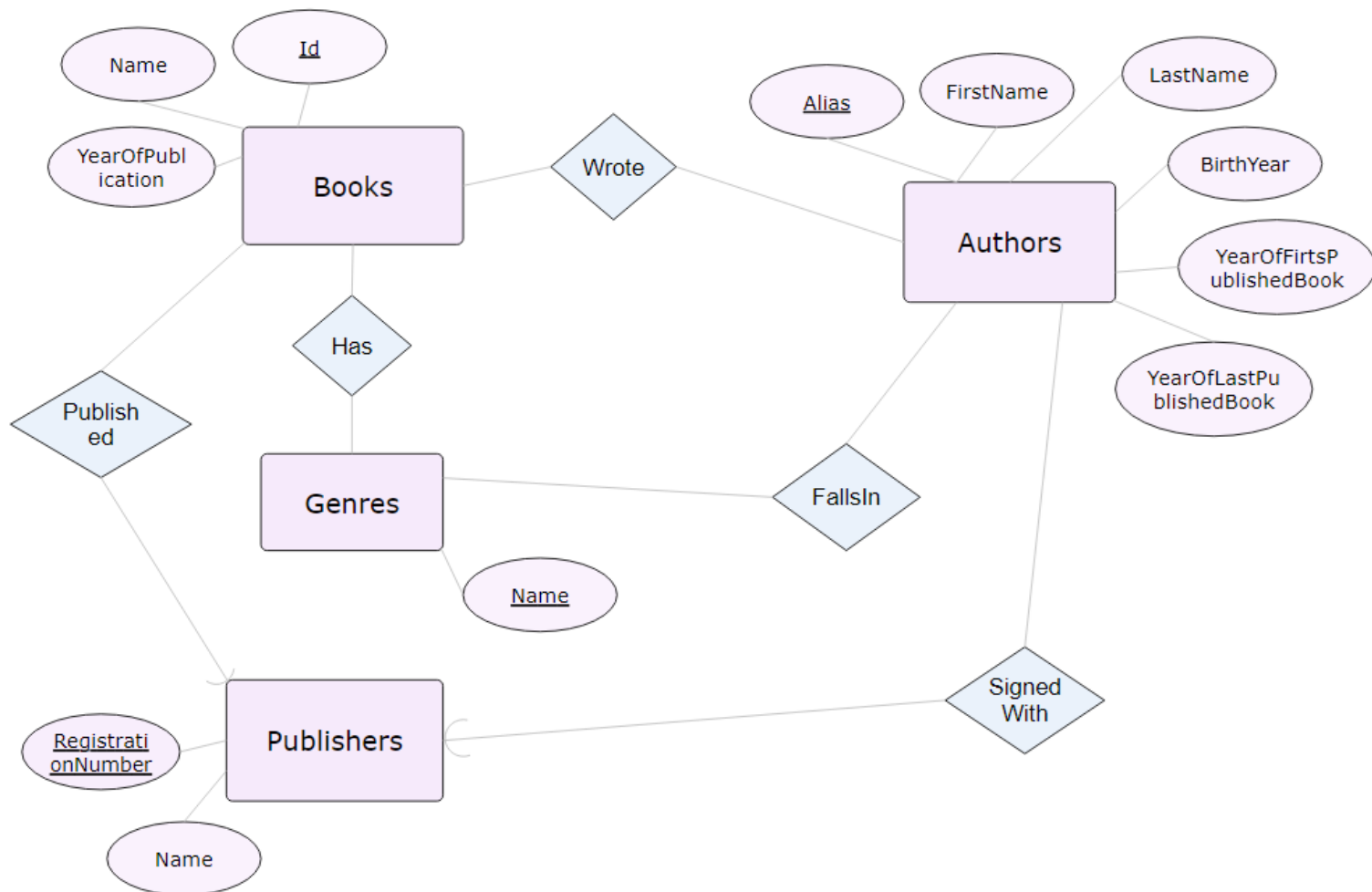
Аналогично, всяка книга има един или повече автори и всеки автор може да е написал 0, 1 или повече от наличните в книжарницата книги. Пазят се данните за всеки автор – псевдоним(уникален), две имена, рождена дата, година на първата му публикувана книга, година на последната му публикувана книга(може да има null стойност). Всяки автор попада в поне една жанрова категория и всяка жанрова категория включва поне един автор. Всяка книга има точно една издателска къща, но всяка издателска къща може да включва 0 или повече книги както 0 или повече автори. Считаме, че всеки автор издава книгите си в точно една издателска къща.

2. Правила и проверки

За базата от данни са в сила следните правила и проверки:

- Годиная на първата издадена книга на всеки автор е не по-голяма от годината на последната му издадена книга и двете години са не по-големи от текущата

3. E/R модел на данни



4. Релационен модел на данни

След преобразуване на модела „Същност-връзки“ към релационен модел и оптимизиране на връзките много-един се получава следния релационен модел на базата от данни:

⑩ Books(Id, Name, YearOfPublication, PublisherRegistrationNumber)

- ⑩ Authors(Alias, FirstName, LastName, BirthYear, YearOfFirstPublishedBook, YearOfLastPublishedBook, PublisherRegistrationNumber)
- ⑩ Wrote(BookId, AuthorAlias)
- ⑩ FallsIn(AuthorAlias, GenreName)
- ⑩ Genre(Name)
- ⑩ HasGenre(BookId, GenreName)
- ⑩ Publishers(RegistrationNumber, Name)

Ограничения:

- **Първични ключове (Primary key)**

Books: PK (Id);
 Authors: PK (Alias);
 Wrote: PK(BookId, AuthorAlias);
 FallsIn: PK(AuthorAlias, GenreName);
 Genre: PK(Name);
 HasGenre: PK(BookId, GenreName);
 Publisher: PK(RegistrationNumber);

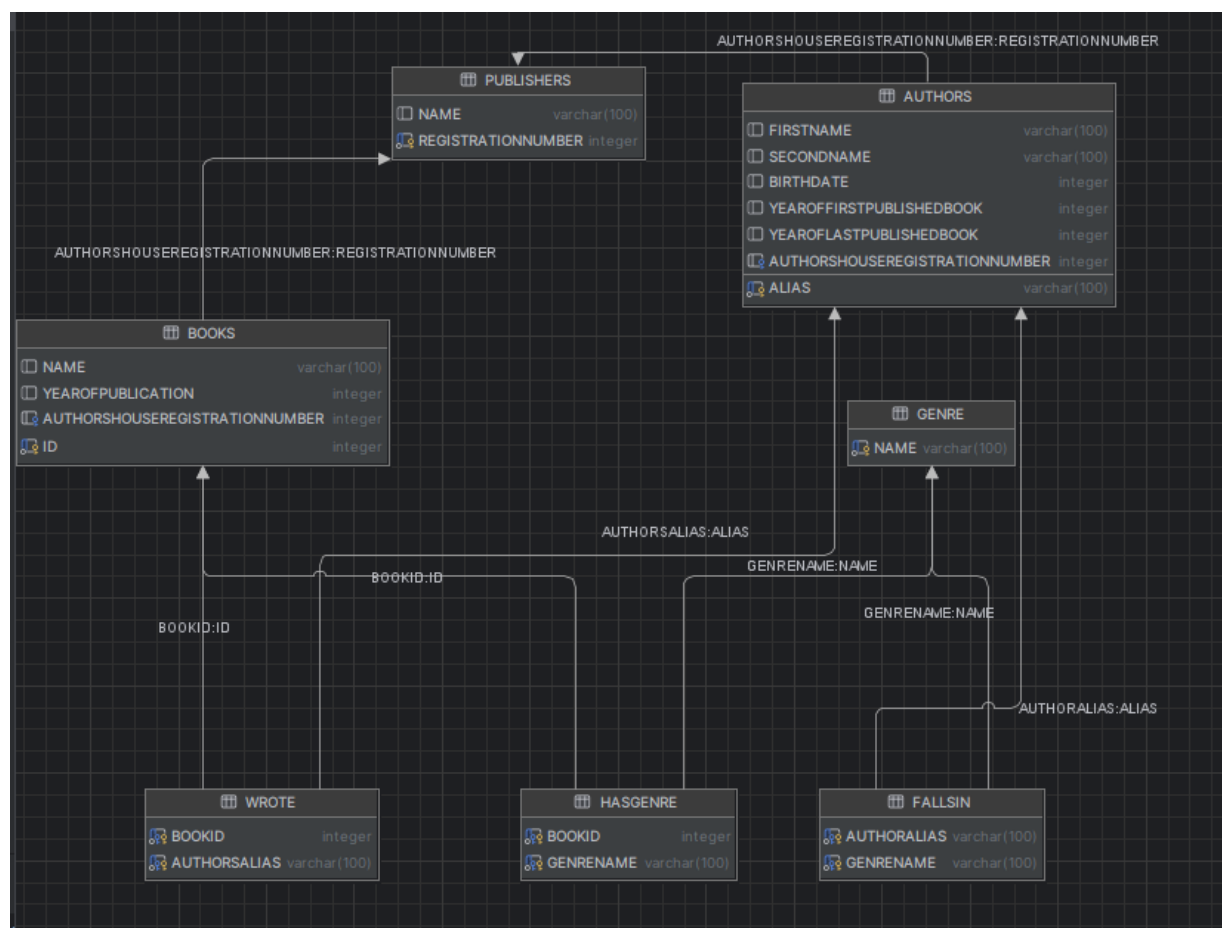
- **Външни ключове (Foreign keys)**

Books: FK(PublisherRegistrationNumber) → Publisher(RegistrationNumber);
 Authors: FK(PublisherRegistrationNumber) → Publisher(RegistrationNumber);

- **Други ограничения (Check constraint)**

Authors: CK (YearOfFirstPublishedBook <= YearOfLastPublishedBook
 and YearOfFirstPublishedBook < CURRENT_YEAR)

5. Схема на базата от данни



6. Функции, тригери и изгледи

- Функции

Описание на функция 1:

По дадена година връща, имената на всички книги, издадени в същата година.

SQL:

```
CREATE FUNCTION GetBooksByYear(Year INT)
RETURNS TABLE (BOOKNAME VARCHAR(100))
RETURN
SELECT Name FROM Books
WHERE Year = YEAROF PUBLICATION;
```

Извикване на функцията:

```
SELECT * FROM TABLE(FN24_45750.GetBooksByYear(1980)) T;
```

Резултат:

	BOOKNAME
1	The Restaurant at the End of the Universe
2	The Number of the Beast
3	The Snow Queen

Описание на функция 2:

По подаден псевдоним на автор връща имената и годините на издаване на всички книги, издадени от автора със същия псевдоним.

SQL:

```
CREATE FUNCTION GetBooksByAuthorAlias(AliasInput VARCHAR(100))
RETURNS TABLE (NAME VARCHAR(100), YEAROFPUBLICATION INT)
RETURN
SELECT Name, YEAROFPUBLICATION FROM Books
WHERE AUTHORSHOUSEREGISTRATIONNUMBER IN
(SELECT A.AUTHORSHOUSEREGISTRATIONNUMBER from AUTHORS A where
AliasInput = A.ALIAS);
```

Извикване на функцията:

```
SELECT * FROM TABLE(FN24_45750.GetBooksByAuthorAlias('Isaac Asimov')) T;
```

Резултат:

	NAME	YEAROFPUBLICATION
1	The Gods Themselves	1972
2	The Naked Sun	1957
3	I, Robot	1950

- Тригери

Описание на тригер 1:

При въвеждане на нов автор проверява годината на последно издадената от него книга и, ако тя е по-голяма от текущата, извежда съобщение за грешка.

SQL:

```
CREATE OR REPLACE TRIGGER TRIG_BEFORE_INSERT_AUTHOR
BEFORE INSERT ON Authors
REFERENCING NEW AS N
FOR EACH ROW
WHEN (N.YEAROFLASTPUBLISHEDBOOK > YEAR(CURRENT_DATE))
  SIGNAL SQLSTATE '75999' ('YEAROFLASTPUBLISHEDBOOK is INVALID!!!!');
```

Тестване на тригер 1:

```
INSERT INTO Authors(ALIAS, FIRSTNAME, SECONDNAME, BIRTHDATE, YEAROFFIRSTPUBLISHEDBOOK,
YEAROFLASTPUBLISHEDBOOK)
VALUES ('Unknown', 'Unknown', 'Unknown', 2000, 2000, 2050);
```

Резултат:

```
FN24_45750> INSERT INTO Authors(ALIAS, FIRSTNAME, SECONDNAME, BIRTHDATE, YEAROFFIRSTPUBLISHEDBOOK, YEAROFLASTPUBLISHEDBOOK)
VALUES ('Unknown', 'Unknown', 'Unknown', 2000, 2000, 2050)
[2024-06-04 15:10:59] [75999] [-438] Application raised error or warning with diagnostic text: "YEAROFLASTPUBLISHEDBOOK is INVALID!!!!".
```

Описание на тригер 2:

При въвеждане на нова книга проверява дали тя вече е добавена, ако да, извежда съобщение за грешка.

SQL:

```
CREATE OR REPLACE TRIGGER TRIG_BEFORE_INSERT_BOOKS
BEFORE INSERT ON Books
REFERENCING NEW AS B
FOR EACH ROW
WHEN (B.ID IN (SELECT Id FROM BOOKS))
  SIGNAL SQLSTATE '75999' ('Book already added!!!!');
```

Тестване на тригер 1:

```
INSERT INTO Books(Id, Name, YEAROFFPUBLICATION) VALUES (default, 'Harry Potter', 1997);
```

- Изгледи

Описание на изглед 1:

Изглед, който връща резултатно множество с информация за име на книга и съответните й автори.

SQL:

```
CREATE VIEW BookAuthor
AS
SELECT B.Name AS BOOKNAME, A.Alias AS AUTHORALIAS
FROM BOOKS B, AUTHORS A
WHERE B.AUTHORSHOUSEREGISTRATIONNUMBER = A.AUTHORSHOUSEREGISTRATIONNUMBER;
```

Извикване на изглед 1:

```
SELECT * FROM BookAuthor;
```

Резултат:

	BOOKNAME	AUTHORALIAS
1	The Gods Themselves	Isaac Asimov
2	The Naked Sun	Isaac Asimov
3	I, Robot	Isaac Asimov
4	Eden	Stanisław Lem
5	The Hammer of God	Arthur Clarke
6	The Restaurant at the End of the Universe	Douglas Adams
7	Unicorn Variations	Roger Zelazny
8	Lord of Light	Roger Zelazny
9	The Number of the Beast	Robert Heinlein
10	The Snow Queen	Joan Vinge

Описание на изглед 2:

Изглед, който връща резултатно множество с информация за име на книга и съответните й жанрове.

SQL:

```
CREATE VIEW BOOKSGENRES
AS
SELECT B.Name AS BOOKNAME, G.GENRENAME AS GENRE
FROM BOOKS B JOIN HASGENRE G ON B.ID = G.BOOKID;
```

Извикване на изглед 2:

```
SELECT * FROM BOOKSGENRES;
```

Резултат:

	BOOKNAME	GENRE
1	The Gods Themselves	Science Fiction
2	The Naked Sun	Mystery
3	The Naked Sun	Science Fiction
4	I, Robot	Science Fiction
5	Eden	Science Fiction
6	The Hammer of God	Science Fiction
7	The Restaurant at the End of the Universe	Comedy
8	The Restaurant at the End of the Universe	Science Fiction
9	Unicorn Variations	Fantasy
10	Unicorn Variations	Science Fiction
11	Lord of Light	Fantasy
12	Lord of Light	Science Fiction

7. Приложение за достъп до базата

Java код:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class DB2Test {

    private Connection connection;
    private Statement statement;
    private ResultSet resultSet;

    public void openConnection(){
        try { DriverManager.registerDriver(new com.ibm.db2.jcc.DB2Driver());}
        catch(Exception cnfex) {
            System.out.println("Problem in loading or registering IBM DB2 JDBC driver");
            cnfex.printStackTrace();
        }
        try {
            connection = DriverManager.getConnection("jdbc:db2://62.44.108.24:50000/SAMPLE", "db2admin", "db2admin");
            statement = connection.createStatement();
        } catch(SQLException s){
            s.printStackTrace();
        }
    }
}
```



```

}

public void closeConnection(){
    try {
        if(null != connection) {
            resultSet.close();
            statement.close();
            connection.close();
        }
    }
    catch (SQLException s) {
        s.printStackTrace();
    }
}

public void select(String stmtnt, int column) {
    try{
        resultSet = statement.executeQuery(stmtnt);
        String result = "";
        while(resultSet.next()) {
            for (int i = 1; i <= column; i++) {
                result += resultSet.getString(i);
                if (i == column) result += "\n";
                else result += ", ";
            }
        }
        System.out.println("Executing query: " + stmtnt + "\n");
        System.out.println("Result output \n");
        System.out.println("----- \n");
        System.out.println(result);
    }
    catch (SQLException s)
    {
        s.printStackTrace();
    }
}

public void insert(String stmtnt) {
    try{
        statement.executeUpdate(stmtnt);
        System.out.println("Successfully inserted!");
    }
    catch (SQLException s){
        System.out.println("NOT inserted!");
        s.printStackTrace();
    }
}

public void delete(String stmtnt) {
    try{ statement.executeUpdate(stmtnt);}
    catch (SQLException s){
        s.printStackTrace();
    }
    System.out.println("Successfully deleted!");
}

public static void main(String[] args) {
    DB2Test db2Obj = new DB2Test();
    String stmtnt = "";
    db2Obj.openConnection();
    stmtnt = ""SELECT B.NAME, A.ALIAS, B.YearOfPublication FROM FN24_45750.BOOKS B JOIN
FN24_45750.AUTHORS A ON B.AuthorsHouseRegistrationNumber = A.AuthorsHouseRegistrationNumber";

```

```
        db2Obj.select(stmtnt, 3);  
        db2Obj.closeConnection();  
    }  
}
```

Резултат:

Result output

The Gods Themselves, Isaac Asimov, 1972

The Naked Sun, Isaac Asimov, 1957

I, Robot, Isaac Asimov, 1950

Eden, Stanisław Lem, 1958

The Hammer of God, Arthur Clarke, 1993

The Restaurant at the End of the Universe, Douglas Adams, 1980

Unicorn Variations, Roger Zelazny, 1983

Lord of Light, Roger Zelazny, 1967

The Number of the Beast, Robert Heinlein, 1980

The Snow Queen, Joan Vinge, 1980 |

Process finished with exit code 0