**Задача за изпълнение**

Да се създаде информационна система подпомагаща дейността на библиотека. Базата данни да съхранява информация за книги, читатели, издателства, автори и жанрове. В таблиците да се включат атрибути като: сигнатура на книга, заглавие, автор, издателство, жанр, номер на читателска карта, ЕГН, име, презиме, фамилия, адрес и телефон на читател, дата на издаване на читателска карта, име на издателство, адрес на издателство, име, фамилия и народност на автор, вид жанр и др. При създаване на таблиците да се избере подходящ тип данни и други свойствата на полетата. Да се валидират данните. Да се свържат таблиците с подходящи релации. В основните таблици да има въведени минимум 10 записа коректни данни. Системата да позволява генериране на справки и отчети.

1. **Анализ на проблема**

В една библиотека се съхраняват много книги. За всяка книга съществува информация като заглавие, автор и идентификационен номер (сигнатура), под която тя е записана. Всички книги трябва да бъдат водени на отчет и във всеки един момент служителите в библиотеката трябва да знаят, кои книги са в наличност. Всяко напускане на дадена книга извън пределите на библиотеката трябва да бъде отбелязано, както и датата, на която тя трябва да бъде върната. Този срок трябва да бъде следен и когато той изтече, и книгата не е върната, трябва да се потърси лицето, на което тя е била предоставена. Това налага съхраняване на данни за читател - име, адрес и телефонен номер. Когато една книга бъде върната в установения срок от читателя, трябва да бъде отбелязано, че той няма задължения към библиотеката. Необходимо е съхраняване и обработка на големи масиви от данни за читателите, книгите (наличните и предоставените за ползване от читателите), жанровете, авторите и издателствата. Трябва да се създаде единна база данни, в която да се съхранява големият обем информация.

Предимствата от създаване и използване на информационна система са:

* улесняване на дейността на служителите;
* минимизира се процеса по допускане на грешки;
* по-бързо обслужване на читателите и ускоряване на процеса по завеждане на нова книга;
* по-добро управление на библиотеката.

1. **Създаване на концептуален модел** – представяне на всички обекти и взаимовръзките между тях.

Концептуална схема наричаме универсално представяне на структурата на данните в рамките на определена предметна област, независимо от крайната реализация на базата данни и апаратната платформа.

Един от най-популярните концептуални модели на данни е моделът „Същност-Връзка“, накратко наричан ER-модел. В диаграма ER (същности – връзки) обектите се представят като именувани правоъгълници, връзките (отношенията) – чрез именуван ромб, който е свързан с класовете обекти чрез линии, а атрибутите (свойствата) – чрез именувана елипса .

**Концептуална схема**

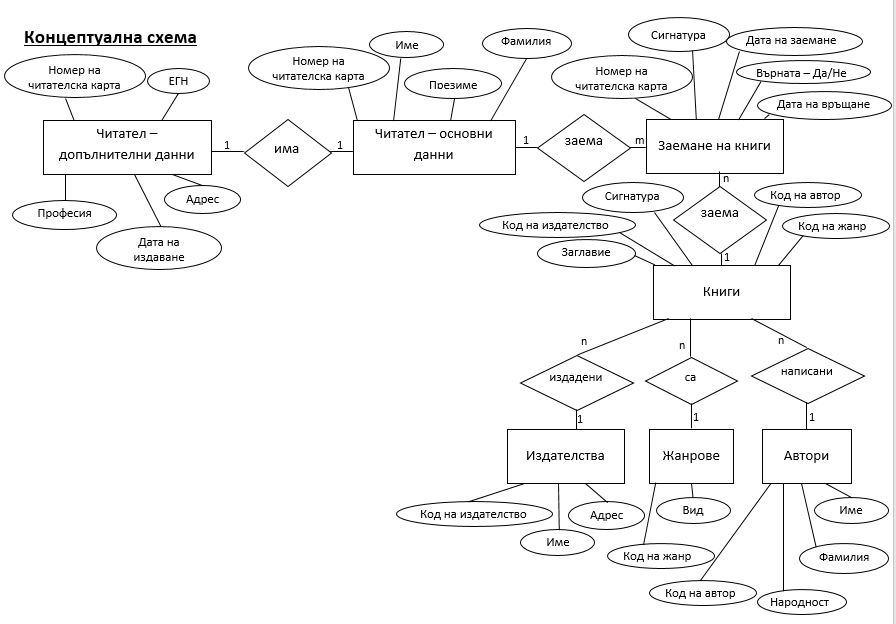
m

n

заема

Книги

Читател

****

1. **Създаване на логически модел**
   1. **Определяне на таблиците, полетата във всяка таблица и осигуряване на коректност на данните**

При определяне на полетата във всяка таблица се задава:

* Име на поле;
* Тип на данни;
* Размер на поле;
* Формат на данни;
* Стойност по подразбиране;
* Условие за допустима стойност;
* Определяне на първичен ключ (ключове);

Коректността на данните се осигурява чрез:

* Тип на данни;
* Входни маски;
* Валидиращи правила;
* Допустими стойности;
* Справочни списъци;
* Задължително въвеждане на данни;
* Индексиране;

|  |
| --- |
| **Читател – допълнителни данни** |
| **Номер на читателска** **карта**  Първичен ключ, Целочислен тип данни, Задължително поле, Индексирано |
| **ЕГН**  Текстово поле с размер 10 символа, Входна маска |
| **Адрес**  Текстово поле с размер до 100 символа |
| **Телефон**  Текстово поле с размер до 10 символа |
| **Професия**  Текстово поле с размер до 30 символа, |

|  |
| --- |
| **Книги** |
| **Сигнатура**  Първичен ключ, Целочислен тип данни, Задължително поле, Индексирано |
| **Заглавие**  Текстово поле с размер до 255 символа |
| **Код на автор**  Външен ключ, Целочислен тип данни, Задължително поле, Индексирано |
| **Код на издателство**  Външен ключ, Целочислен тип данни, Задължително поле, Индексирано |
| **Код на жанр**  Външен ключ, Целочислен тип данни, Задължително поле, Индексирано |

|  |
| --- |
| **Читател – основни данни** |
| **Номер на читателска** **карта**  Първичен ключ, Автоматично номериране, Задължително поле, Индексирано |
| **Име**  Текстово поле с размер до 15 символа, Входна маска |
| **Презиме**  Текстово поле с размер до 15 символа, Входна маска |
| **Фамилия**  Текстово поле с размер до 15 символа, Входна маска |
| **Дата на издаване**  Кратък формат за дата, Стойността по подразбиране е текущата дата |

|  |
| --- |
| **Заемане на книги** |
| **Номер на читателска карта**  Първичен ключ, Целочислен тип данни, Задължително поле, Индексирано |
| **Сигнатура**  Първичен ключ, Целочислен тип данни, Задължително поле, Индексирано |
| **Дата на заемане**  Първичен ключ, Кратък формат за дата, Задължително поле, Индексирано, Стойността по подразбиране е текущата дата, Условие за допустима стойност и извеждане на съобщение при не спазено правило за валидност |
| **Дата на връщане**  Кратък формат за дата |
| **Върната Да/Не**  Поле съдържащо само една от две стойности - „Yes”, “No”. Стойността по подразбиране е “No”. |

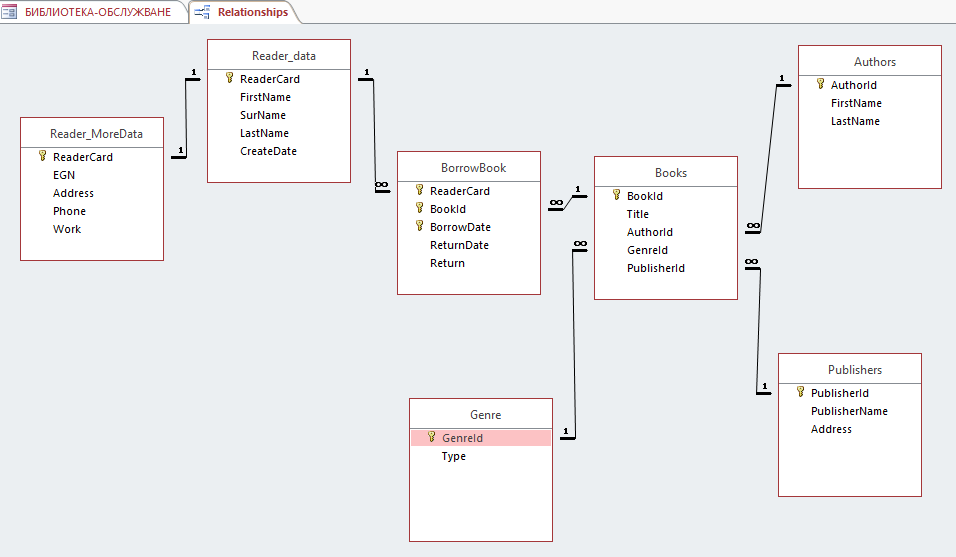
|  |
| --- |
| **Издателства** |
| **Код на издателство**  Първичен ключ, Автоматично номериране, Задължително поле, Индексирано |
| **Име**  Текстово поле с размер до 50 символа |
| **Адрес**  Текстово поле с размер до 100 символа |

**Съставен ключ**

|  |
| --- |
| **Автори** |
| **Код на автор**  Първичен ключ, Автоматично номериране, Задължително поле, Индексирано |
| **Име**  Текстово поле с размер до 15 символа, Входна маска |
| **Фамилия**  Текстово поле с размер до 15 символа, Входна маска |

|  |
| --- |
| **Жанрове** |
| **Код на жанр**  Първичен ключ, Автоматично номериране, Задължително поле, Индексирано |
| **Вид**  Текстово поле с размер до 50 символа |

**Релационна схема**

****

**Релации**

1. Предназначение на релациите

Релациите осъществяват връзката между таблиците и помагат за избягване на изолирани записи в таблиците. Заявки, форми и отчети могат да включват повече от една таблица.

1. Създаване на релации
   * Всички таблици трябва да са затворени.
   * От Database Tools се избира Relationships.
   * Добавят се таблиците.
   * Свързваме полетата (полето, което е първичен ключ на първата таблица се свързва с полето, което е външен ключ за втората таблица (1:N) или полето, което е първичен ключ на първата таблица се свързва с полето, което е първичен ключ за втората таблица (1:1).
   * Цялост на връзките – система от правила, които гарантират, че отношенията между записите са изпълнени и не позволяват случайно да се изтриват или променят данни от свързани в релация таблици. Задават се в прозореца Edit Relationships, чрез активиране на **опцията Enforce Referential Integrity**. Чрез тази опция се визуализира и вида на връзката (1:1 или 1: ∞). Включване на опциите за каскадно обновяване (**Cascade Update**) и изтриване (**Delete Related Records**) гарантира, че промените които се правят в основната таблица ще се разпростират и над съответните записи от свързаната таблица, участваща в релацията.