**Съдържание на текстовия документ с описание на проект- създаване на база данни**

1. ТЕМА: Организация на библиотека

2. АВТОРИ: Александър Николов и Диан Грозев

3. РЕЗЮМЕ: описание на заданието

3.1. Цели (предназначение): целта е да се създаде база данни за организиране на библиотека

3.2. Основни етапи в реализирането на проекта (основни дейности, роли на авторите)Реализиране на база данни с **MySQL** преминава през няколко основни етапа:

**1. Анализ и планиране**

* Определяне на изискванията към данните.
* Проектиране на релационния модел (ER диаграма).
* Идентифициране на **таблиците**, **атрибутите**, **първичните** и **външните ключове**.

**2. Създаване на базата данни**

 Код: CREATE DATABASE Library;

**3. Дефиниране на таблиците**

CREATE TABLE Authors(

author\_id int auto\_increment Primary key,

first\_name varchar(100) not null,

last\_name varchar(100) not null,

birth\_date DATE

);

CREATE TABLE Genres(

genre\_id INT AUTO\_INCREMENT PRIMARY KEY,

genre\_name VARCHAR(100) NOT NULL

);

**4. Определяне на релации**

* Външни ключове (FOREIGN KEY) се използват за свързване на таблиците:

В таблица Books: author\_id;

В таблица Loans: book\_id, member\_id;

В таблица Book\_genres: book\_id, genre\_id;

* Кода на свързващата таблица :

create table Book\_genres(

book\_id int,

genre\_id int,

Primary key(book\_id,genre\_id),

constraint foreign key(book\_id) references Books(book\_id),

constraint foreign key(genre\_id) references Genres(genre\_id)

);

**6. Попълване с тестови данни**

INSERT INTO Authors (first\_name, last\_name, birth\_date)

VALUES

('J.K.', 'Rowling', '1965-07-31'),

('George', 'Orwell', '1903-06-25'),

('J.R.R.', 'Tolkien', '1892-01-03'),

('Agatha', 'Christie', '1890-09-15'),

('Isaac', 'Asimov', '1920-01-02'),

('Stephen', 'King', '1947-09-21'),

('Margaret', 'Atwood', '1939-11-18'),

('Ray', 'Bradbury', '1920-08-22'),

('Haruki', 'Murakami', '1949-01-12'),

('Kurt', 'Vonnegut', '1922-11-11');

**7. Извличане на данни и тестване**

1) SELECT member\_id, COUNT(\*) AS total\_loans

FROM Loans GROUP BY member\_id;

2)SELECT m.first\_name, m.last\_name, b.title

FROM Members m

JOIN Loans l ON m.member\_id = l.member\_id

JOIN Books b ON l.book\_id = b.book\_id;

8.Роли на авторите:

Александър-Проектиране на структурата на базата данни, описание на документацията и 5 заявки

Диан-Описание на 5 заявки и заданието

3.3. Ниво на сложност на проекта − основни проблеми при реализация

**1) Дизайн и нормализация на базата**

**Проблеми:**

* **Лошо структурирани таблици** → Води до дублиране на данни и усложнено управление.
* **Прекомерна нормализация** → Усложнява заявките и може да забави работата.
* **Недостатъчна нормализация** → Води до излишно съхраняване на данни и проблеми при актуализация.

**Решение:**

* Спазване на **нормализационните форми (1NF, 2NF, 3NF)**, но с баланс между производителност и удобство.

**2) Определяне на правилните типове данни**

**Проблеми:**

* Използване на **неподходящи типове данни** → Например TEXT вместо VARCHAR за кратки текстове.
* Грешен избор на **ключове** → Например INT вместо BIGINT при очаквано голямо количество записи.

**Решение:**

* Правилен анализ на структурата и очаквания обем на данните.
* Използване на ENUM за ограничени списъци от стойности.
* И други проблеми , които сте имали

3.4. Логическо и функционално описание на решението – Проектът представлява **база данни за управление на библиотека**, създадена с **MySQL**. Целта  е да съхранява информация за книги, автори, жанрове, заеми и потребители, като поддържа ефективно извличане и обработка на данни.

**1. Логическо описание на решението**

Логическият модел на базата данни следва **релационния модел**, където всяка таблица представя конкретен обект в системата. Основните релации са **едно към много (1:M)** и **много към много (M:N)**.

**Основни обекти в системата:**

* **Книги (Books)** – съдържа информация за id,заглавието, автора, дата на издаване и наличността.
* **Автори (Authors)** – съхранява id, имената на авторите и дата на раждане.
* **Жанрове (Genres)** – определя жанра на книгите (напр. Фантастика, Научна литература) чрез id, наименование.
* **Потребители (Members)** – информация за регистрираните потребители на библиотеката чрез id, имената на потребителя, имейл и дата на започнато членство.
* **Заеми (Loans)** – записва id, книгата, потребителя, дата на заемане, дата на връщане.
* **Книги-Жанрове**  **(Book\_genres)** – свързва книгите с жанровете.

**Взаимовръзки между таблиците:**

1. **Една книга може да има само един автор (1:M)** → Books.Author\_ID реферира Authors.Author\_ID.
2. **Една книга може да принадлежи към няколко жанра, а един жанр може да има много книги (M:N)** → реализация със свързваща таблица BookGenres.

3.5. Реализация − обосновка за използвани технологични средства(MySQL)

По задание базата данни задължително трябва да е на MySQL.

MySQL е една от най-разпространените СУБД в света с дългогодишен опит в разработката и поддръжката си.

Системата е доказала своята стабилност и надеждност в различни бизнес среди – от малки до корпоративни решения.

MySQL поддържа транзакционен режим (InnoDB), който гарантира целостта на данните чрез ACID принципи (Atomicity, Consistency, Isolation, Durability).

Това е от ключово значение за приложения, където данните трябва да бъдат надеждно записвани и манипулирани.

Вградената поддръжка на външни ключове и каскадни действия улеснява управлението на взаимосвързаните данни и предотвратява неконсистентностите.

MySQL може да обслужва както малки, така и големи проекти, като се адаптира към нарастващия обем на данните и потребителския трафик.

Използването на **MySQL** като технологично средство за реализиране на базата данни в проекта се обосновава с нейната доказана надеждност, висока производителност, гъвкавост и силна поддръжка от страна на общността. Тези характеристики я правят подходящ избор за разработване на решения, които изискват надеждно съхранение, обработка и защита на данни.

3.6. Описание на базата данни - как се използва, как ще се поддържа.

Базата данни е проектирана за **библиотека**, която управлява информация за книги, автори, жанрове, потребители и заеми.

1)SELECT g.genre\_name

FROM Genres g

INNER JOIN Book\_genres bg ON g.genre\_id = bg.genre\_id

INNER JOIN Books b ON bg.book\_id = b.book\_id

WHERE b.title = 'The Hobbit';

2) SELECT first\_name, last\_name

FROM Authors

WHERE birth\_date > '1940-01-01';

3.7. Заключение –

Решението предоставя добре структурирана и оптимизирана релационна база данни, която позволява:

* Ефективно управление на книги и автори.   
  Лесно търсене и категоризиране на книги.    
  Използването на първични ключове и външни ключове гарантира, че връзките между автори, книги, членове, заеми и жанрове се поддържат последователно, улеснявайки ефикасното търсене.