НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО

КАФЕДРА ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

**до курсової роботи**

з дисципліни

«БАЗИ ДАНИХ ТА ІНФОРМАЦІЙНІ СИСТЕМИ»

на тему

«Інформаційна система бібліотеки»

Викладач М. В. Коломицев

Студент В. В. Бичок

Група ФБ-34

Варіант 1

Київ 2024

Інформаційна система бібліотеки.

БД повинна забезпечувати зберігання відомостей про наявні в бібліотеці книги, про читачів бібліотеки і читальних залах.

Для кожної книги в БД повинні зберігатися такі відомості:

* Назва книги
* Автор(и)
* Видавництво
* Рік видання
* Число екземплярів цієї книги в кожному залі бібліотеки
* Шифр книги
* Дата закріплення книги за читачем.

Відомості про читачів бібліотеки повинні включати в собі:

* Номер читацького квитка
* Прізвище читача
* Номер паспорта
* Дату народження
* Адресу
* Номер телефону
* Освіту
* Наявність наукового ступеня.

Читачі закріплюються за певним залом і можуть записуватися і виписуватися з бібліотеки.

Бібліотека має кілька читальних залів, які характеризуються:

* Номером
* Назвою
* Місткістю, тобто кількістю людей, які можуть одночасно працювати в залі.

Бібліотека може отримувати нові книги і списувати старі.

Шифр книги може змінитися в результаті перекласифікації, а номер читацького квитка в результаті перереєстрації.

Бібліотекарю можуть знадобитися такі відомості про поточний стан бібліотеки:

* Які книги закріплені за певним читачем?
* Як називається книга з заданим шифром?
* Який шифр у книги з заданим назвою?
* Коли книга була закріплена за читачем?
* Хто з читачів взяв книгу більше місяця тому?
* За ким із читачів закріплені книги, кількість примірників яких в бібліотеці не перевищує 2?
* Яке число читачів користується бібліотекою?
* Скільки в бібліотеці читачів молодших 20 років?
* Скільки читачів у відсотковому відношенні мають початкову освіту, середню, вищу, вчений ступінь?

Бібліотекар може виконувати наступні операції:

* Записати в бібліотеку нового читача.
* Виключити зі списку читачів людей, які записалися в бібліотеку більше року тому і які не пройшли перереєстрацію.
* Списати стару або втрачену книгу.
* Прийняти книгу в фонд бібліотеки.

Необхідно передбачити можливість видачі довідки про кількість книг певного автора в читальному залі і звіту про роботу бібліотеки протягом місяця.

Звіт повинен включати в себе наступну інформацію:

* Кількість книг і читачів на поточний день в кожному із залів і в бібліотеці в цілому.
* Кількість читачів, які записалися в бібліотеку за звітний місяць.
* Які книги і скільки разів були взяті цього місяця.
* Хто з читачів не брав книг.

Передбачити розробку тригерів, які забезпечують каскадні зміни в зв'язаних таблицях.

Зміст

[Вступ 6](#_Toc185810364)

[Аналіз предметної області 9](#_Toc185810365)

[Функціональне моделювання 13](#_Toc185810366)

[Концептуальна модель бази даних 14](#_Toc185810367)

[Реляційна модель бази даних 19](#_Toc185810368)

[Фізична модель бази даних 20](#_Toc185810369)

[Опис реалізації завдань відповідно обраного варіанту 35](#_Toc185810370)

[Висновок 46](#_Toc185810371)

# Вступ

У сучасному світі бібліотеки продовжують відігравати важливу роль у нашому житті. Саме тут зберігаються величезні обсяги знань, зібраних людством протягом століть. Але сьогодні бібліотеки стикаються з серйозною проблемою - як ефективно управляти книжковим фондом та обслуговувати велику кількість читачів? Старі методи ведення паперової документації вже не справляються з цим завданням.

Я вважаю розробку спеціальної бази даних для бібліотеки надзвичайно важливою. Ведення обліку в паперових журналах створює багато проблем - це забирає багато часу, часто виникають помилки, а пошук потрібної інформації перетворюється на справжнє випробування. З власного досвіду можу сказати, що коли бібліотечний фонд постійно поповнюється, а кількість читачів зростає, стає практично неможливо ефективно контролювати все в звичайних журналах.

Саме тому я поставив собі за мету створити зручну комп'ютерну систему для бібліотеки. Вона має автоматизувати всю щоденну роботу - від обліку книг до формування звітів. Це значно полегшить роботу бібліотекарів та покращить обслуговування читачів.

Для досягнення цієї мети мені потрібно:

1. Розібратися в тому, як працює бібліотека і що потрібно для її ефективної роботи

2. Створити загальний план бази даних

3. Розробити детальну модель з усіма необхідними таблицями

4. Реалізувати все це в MSSQL

5. Додати захист від помилок при введенні даних

6. Створити необхідні запити для отримання інформації

7. Зробити зручний інтерфейс

Я планую розробляти базу даних в три етапи:

- Спочатку намалюю загальну схему, щоб зрозуміти, як все пов'язано між собою

- Потім створю детальний план з усіма полями та їх властивостями

- І нарешті, реалізую все це в MSSQL

Для своєї роботи я обрав MSSQL, тому що ця система має все необхідне - можливість створювати тригери та процедури, які захистять дані від помилок. База даних буде зберігати всю важливу інформацію про книги, читачів та видачу літератури.

Особливу увагу я планую приділити захисту від помилок при введенні даних. Наприклад, система не дозволить видати книгу, якої немає в наявності, або зареєструвати читача з неправильними даними. Це допоможе уникнути плутанини та зробить роботу більш надійною.

В результаті бібліотека отримає сучасну комп'ютерну систему, яка зробить роботу простішою та ефективнішою. Бібліотекарі зможуть швидко знаходити потрібну інформацію, а читачі - отримувати якісне обслуговування без довгого очікування. Крім того, керівництво зможе легко отримувати різні звіти про роботу бібліотеки.

# Аналіз предметної області

Аналіз предметної області

Структура предметної області

Бібліотека як предметна область являє собою складну систему, що складається з наступних основних компонентів:

1. Книжковий фонд:
   * Включає всі наявні в бібліотеці книги
   * Кожна книга характеризується унікальним шифром, назвою, автором(ами), видавництвом, роком видання
   * Ведеться облік кількості примірників кожної книги в різних читальних залах
   * Здійснюється моніторинг видачі книг читачам
2. Читачі:
   * Реєструються в системі з унікальним номером читацького квитка
   * Надають особисті дані (ПІБ, паспортні дані, контактна інформація)
   * Характеризуються рівнем освіти та наявністю наукового ступеня
   * Закріплюються за певним читальним залом
3. Читальні зали:
   * Мають унікальний номер та назву
   * Характеризуються місткістю (максимальною кількістю читачів)
   * Містять певну кількість примірників різних книг

Основні завдання предметної області

1. Управління книжковим фондом:
   * Облік наявних книг
   * Прийом нових книг
   * Списання старих або втрачених книг
   * Перекласифікація книг (зміна шифру)
   * Розподіл книг по читальних залах
2. Робота з читачами:
   * Реєстрація нових читачів
   * Перереєстрація існуючих читачів
   * Видалення неактивних читачів
   * Закріплення читачів за читальними залами
3. Облік видачі книг:
   * Реєстрація видачі книг читачам
   * Контроль термінів повернення
   * Моніторинг наявності книг
4. Формування звітності:
   * Щомісячні звіти про роботу бібліотеки
   * Статистичні довідки
   * Аналіз використання книжкового фонду

Вхідні дані

1. Інформація про нові книги:
   * Назва книги
   * Автор(и)
   * Видавництво
   * Рік видання
   * Кількість примірників
   * Шифр книги
2. Дані про читачів:
   * Номер читацького квитка
   * Прізвище
   * Номер паспорта
   * Дата народження
   * Адреса
   * Телефон
   * Освіта
   * Інформація про науковий ступінь
3. Дані про читальні зали:
   * Номер залу
   * Назва залу
   * Місткість

Вихідні дані

1. Інформаційні довідки:
   * Список книг, закріплених за читачем
   * Інформація про книгу за шифром
   * Шифр книги за назвою
   * Дата видачі книги
   * Список боржників (книги на руках більше місяця)
   * Статистика по читачах з різним рівнем освіти
2. Щомісячні звіти:
   * Кількість книг і читачів по залах
   * Кількість нових читачів
   * Статистика використання книг
   * Список неактивних читачів

# Функціональне моделювання

Для того, щоб зручно зобразити процеси в інформаційній системі, використовується DFD-діаграма, яка показує потоки даних і процеси в системі. Діаграма показує взаємозв'язки між основними об'єктами та запитами в системі, що робить доступ до інформації зручним.

Наша DFD-діаграма відображає те що Бібліотекар може додавати книги і списувати за потреби і можливості, записувати читачів і виписувати читачів, у яких немає наразі книжок і які не захотіли перереєструватися, читач може отримувати книги за допомогою Бібліотекаря, який має певні права у базі даних.

Зображення, що містить текст, схема, малюнок, ескіз

Автоматично згенерований опис

# Концептуальна модель бази даних

За результатами аналізу ПО опишемо сутності та зв’язки між ними. Представимо наш опис у вигляді таблиць, так як сутності і їх атрибути дуже добре так представляються. Представимо опис у таких таблицях: таблиця сутностей, таблиця атрибутів для сутності та зв’язки.

Таблиця 4.1 | Таблиця сутностей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Назва сутності | Умовне позначення | Первинний ключ | Частота корекції | Завдання та запити | Обмеження на доступність |
| Книги | Books | Pressmark | Низька | Додавання, оновлення, видалення книг | Адміністратор,  Бібліотекар |
| Читачі | Readers | Ticket | Середня | Додавання, оновлення, видалення  читачів | Адміністратор, Бібліотекар |
| Зали | Halls | HallID | Дуже низька | Оновлення кількості місць в залі | Адміністратор,  Бібліотекар |
| Зв’язок через який реалізовано закріплення книг за конкретним читачем | Readers  TackingBook | IDBook  Reader | Висока | Додавання нових записів про взяття книги | Адміністратор,  Бібліотекар |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Назва  атри  бута | Умовне  позначення | Тип  значення | Формат | Діапазон  значень | Невиз  наче  ність | Уні  каль  ність | Обмеження  на доступність |
| Шифр книги | Pressmark  **PK** | Атома  рний | INT | Натураль  ні числа | Ні | Так | Адміністратор,  Бібліотекар |
| Назва книги | Named | Атома  рний | VARCHAR (50) | Текстовий  рядок до  50  символів | Ні | Так | Адміністратор,  Бібліотекар |
| Автор | Author | Атома  рний | VARCHAR (35) | Текстовий  рядок до  35  символів | Ні | Ні | Адміністратор,  Бібліотекар |
| Видавець | Publisher | Атома  рний | VARCHAR (50) | Текстовий  рядок до  50  символів | Ні | Ні | Адміністратор,  Бібліотекар |
| Дата видання | Publication  Date | Атома  рний | SMALL  DATE  TIME | Дата та час з  точністю до  хвилин | Ні | Ні | Адміністратор,  Бібліотекар |
| Кількість  книг у  першому  холі | Count1 | Атома  рний | INT | Натураль  ні числа | Ні | Ні | Адміністратор,  Бібліотекар |
| Кількість  книг у  другому  холі | Count2 | Атома  рний | INT | Натураль  ні числа | Ні | Ні | Адміністратор,  Бібліотекар |
| Кількість  книг у  третьому  холі | Count3 | Атома  рний | INT | Натураль  ні числа | Ні | Ні | Адміністратор,  Бібліотекар |

Таблиця 4.2 | Таблиця атрибутів для сутності “Книги”

Таблиця 4.3 | Таблиця атрибутів для сутності “Читачі”

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Назва  атри  бута | Умовне  позначення | Тип  значення | Формат | Діапазон  значень | Невиз  наче  ність | Уні  каль  ність | Обмеження  на доступність |
| Читацький квиток | Ticket  **PK** | Атома  рний | INT | Натураль  ні числа | Ні | Так | Адміністратор,  Бібліотекар |
| Призвище  Ім’я  По  батькові | PIB | Атома  рний | VARCHAR  (50) | Текстовий  рядок до  50  символів | Ні | Так | Адміністратор,  Бібліотекар |
| Номер  паспорту | PasportID | Атома  рний | CHAR  (9) | Текстовий  рядок до  9  символів | Ні | Так | Адміністратор,  Бібліотекар |
| Дата  наро  дження | DateOf  Birst | Атома  рний | SMALL  DATE  TIME | Дата та час  з точністю  до хвилин | Ні | Ні | Адміністратор,  Бібліотекар |
| Адреса | Addressa | Атома  рний | VARCHAR  (35) | Текстовий  рядок до  35  символів | Ні | Ні | Адміністратор,  Бібліотекар |
| Номер телефону | Phone  Number | Атома  рний | CHAR  (9) | Текстовий  рядок до  9  символів | Ні | Так | Адміністратор,  Бібліотекар |
| Освіта | Science | Атома  рний | VARCHAR  (50) | Текстовий  рядок до  50  символів | Ні | Ні | Адміністратор,  Бібліотекар |
| Наявність  наукової  ступені | Academic  Degree | Атома  рний | VARCHAR  (3) | Текстовий  рядок до  3  символів | Ні | Ні | Адміністратор,  Бібліотекар |
| Номер  залу | HallID | Атома  рний | INT | Натураль  ні числа | Ні | Ні | Адміністратор,  Бібліотекар |
| Дата  прив’язки  до залу | Date  Include | Атома  рний | SMALL  DATE  TIME | Дата та час  з точністю  до хвилин | Ні | Ні | Адміністратор,  Бібліотекар |

Таблиця 4.4 | Таблиця атрибутів для сутності “Зали”

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Назва  атри  бута | Умовне  позначення | Тип  значення | Формат | Діапазон  значень | Невиз  наче  ність | Уні  каль  ність | Обмеження  на  доступність |
| Номер  холу | HallID  **PK** | Атома  рний | INT | Натураль  ні числа | Ні | Так | Адміністратор,  Бібліотекар |
| Назва  холу | Named | Атома  рний | VARCHAR  (35) | Текстовий  рядок до  35  символів | Ні | Так | Адміністратор,  Бібліотекар |
| Місткість холу | Size | Атома  рний | INT | Натураль  ні числа | Ні | Ні | Адміністратор,  Бібліотекар |

Таблиця 4.5 | Таблиця атрибутів для сутності зв’язку через який реалізовано закріплення книг за конкретним читачем

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Назва  атри  бута | Умовне  позначення | Тип  значення | Формат | Діапазон  значень | Невиз наче ність | Уні каль ність | Обмеження на доступність |
| Іденти фікатор взяття  книги | IDBook  Reader  **PK** | Атома  рний | INT | Натураль  ні числа | Ні | Так | Адміністратор,  Бібліотекар |
| Шифр книги | Pressmark **FK** | Атома  рний | INT | Натураль  ні числа | Ні | Ні | Адміністратор,  Бібліотекар |
| Квиток читача | Ticket **FK** | Атома  рний | INT | Натураль  ні числа | Ні | Ні | Адміністратор,  Бібліотекар |
| Дата взяття  книги | IssueDate | Атома  рний | SMALL  DATE  TIME | Дата та час  з точністю  до хвилин | Ні | Ні | Адміністратор,  Бібліотекар |
| Дата повер нення книги | DateOf  Receipt | Атома  рний | SMALL  DATE  TIME | Дата та час  з точністю  до хвилин | Так | Ні | Адміністратор,  Бібліотекар |

**Адміністратор системи:**

* Має доступ до всіх функцій системи.
* Може додавати, видаляти або редагувати будь-які записи (аукціони, предмети, транзакції, особи тощо).
* Керує ролями користувачів.

**Бібліотекар:**

* Має доступ до виконання процедур «Які книги закріплені за певним читачем», «Як називається книга з заданим шифром», «Який шифр у книги з заданим назвою», «Коли книга була закріплена за читачем», «Хто з читачів взяв книгу більше місяця тому», «За ким із читачів закріплені книги, кількість примірників яких в бібліотеці не перевищує 2», «Яке число читачів користується бібліотекою», «Скільки в бібліотеці читачів молодших 20 років», «Скільки читачів у відсотковому відношенні мають початкову освіту, середню, вищу, вчений ступінь»,
* Та може виконувати операції, такі як «Записати в бібліотеку нового читача», «Виключити зі списку читачів людей, які записалися в бібліотеку більше року тому і які не пройшли перереєстрацію», «Списати стару або втрачену книгу», «Прийняти книгу в фонд бібліотеки»

# Реляційна модель бази даних

Далі для виконання курсової роботи потрібно створити ER-діаграму. Це модель даних, яка потрібна для того щоб переглядати зв’язки між сутностями, та об’єктами, також щоб визначити які саме зв’язки використовуються. Існують різні типи таких зв’язків, так як один-до-багатьох, багато-до-багатьох, один-до-одного і тд. Ми будемо використовувати ER-діаграму для того щоб продемонструвати ці зв’язки.

**Зображення, що містить текст, знімок екрана, схема, Паралель

Автоматично згенерований опис**Малюнок **5.1** | **ER** діаграма

# Фізична модель бази даних

При створенні таблиць одразу задаю ключі.

USE kursova;

DROP TABLE IF EXISTS ReadersTakingBook;

DROP TABLE IF EXISTS Books;

DROP TABLE IF EXISTS Readers;

DROP TABLE IF EXISTS Halls;

GO

CREATE TABLE Books(

Pressmark INT,

Named VARCHAR(50) NOT NULL UNIQUE,

Author VARCHAR(35) NOT NULL,

Publisher VARCHAR(50) NOT NULL,

PublicationDate SMALLDATETIME NOT NULL,

Count1 INT NOT NULL,

Count2 INT NOT NULL,

Count3 INT NOT NULL,

CONSTRAINT BooksPrimary PRIMARY KEY (Pressmark),

);

CREATE TABLE Halls(

HallID INT IDENTITY,

Named VARCHAR(35) NOT NULL UNIQUE,

Size INT NOT NULL,

CONSTRAINT HallsPrimary PRIMARY KEY (HallID)

);

CREATE TABLE Readers(

Ticket INT,

PIB VARCHAR(50) NOT NULL UNIQUE,

PasportID CHAR(9) NOT NULL UNIQUE,

DateOfBirst SMALLDATETIME NOT NULL,

Addressa VARCHAR(35) NOT NULL,

PhoneNumber CHAR(9) NOT NULL UNIQUE,

Science VARCHAR(50) NOT NULL,

AcademicDegree VARCHAR(3) NOT NULL,

HallID INT NOT NULL,

DateInclude SMALLDATETIME NOT NULL,

CONSTRAINT ReadersPrimary PRIMARY KEY (Ticket),

CONSTRAINT AcademicDegreeCheck CHECK (AcademicDegree IN ('Yes', 'No')),

CONSTRAINT HallsForeign FOREIGN KEY (HallID) REFERENCES Halls ON UPDATE CASCADE

);

CREATE TABLE ReadersTakingBook(

IDBookReader INT IDENTITY,

Pressmark INT NOT NULL,

Ticket INT NOT NULL,

IssueDate SMALLDATETIME NOT NULL,

DateOfReceipt SMALLDATETIME,

CONSTRAINT ReadersTakingBookPrimary PRIMARY KEY (IDBookReader),

CONSTRAINT BooksForeign FOREIGN KEY (Pressmark) REFERENCES Books ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT ReadersForeign FOREIGN KEY (Ticket) REFERENCES Readers ON DELETE CASCADE ON UPDATE CASCADE

);

GO

DROP TRIGGER IF EXISTS InsertedReadersTakingBooks

GO

CREATE TRIGGER InsertedReadersTakingBook

ON ReadersTakingBook

INSTEAD OF INSERT

AS

DECLARE @Pressmark INT, @HallID INT;

SELECT @Pressmark = Pressmark, @HallID = HallID FROM inserted I INNER JOIN Readers R ON I.Ticket = R.Ticket;

IF (@HallID = 1 AND (SELECT COUNT1 FROM Books WHERE Pressmark = @Pressmark) > 0)

BEGIN

INSERT INTO ReadersTakingBook (Pressmark, Ticket, IssueDate, DateOfReceipt)

SELECT Pressmark, Ticket, IssueDate, DateOfReceipt

FROM inserted;

END

ELSE IF (@HallID = 2 AND (SELECT COUNT2 FROM Books WHERE Pressmark = @Pressmark) > 0)

BEGIN

INSERT INTO ReadersTakingBook (Pressmark, Ticket, IssueDate, DateOfReceipt)

SELECT Pressmark, Ticket, IssueDate, DateOfReceipt

FROM inserted;

END

ELSE IF (@HallID = 3 AND (SELECT COUNT3 FROM Books WHERE Pressmark = @Pressmark) > 0)

BEGIN

INSERT INTO ReadersTakingBook (Pressmark, Ticket, IssueDate, DateOfReceipt)

SELECT Pressmark, Ticket, IssueDate, DateOfReceipt

FROM inserted;

END

ELSE

BEGIN

RAISERROR('There is no book in this room.', 16, 1)

ROLLBACK TRAN;

END

GO

DROP TRIGGER IF EXISTS CheckIssueDateBeforeDateInclude;

GO

CREATE TRIGGER CheckIssueDateBeforeDateInclude

ON ReadersTakingBook

AFTER INSERT, UPDATE

AS

BEGIN

IF EXISTS (SELECT 'TRUE'

FROM inserted I

INNER JOIN Readers R ON I.Ticket = R.Ticket

WHERE I.IssueDate <= R.DateInclude

)

BEGIN

RAISERROR('IssueDate must be later than DateInclude for the reader.', 16, 1);

ROLLBACK TRANSACTION;

END

END

GO

DROP TRIGGER IF EXISTS CheckDelete;

GO

CREATE TRIGGER CheckDelete

ON Readers

INSTEAD OF DELETE

AS

BEGIN

IF EXISTS (

SELECT 'TRUE'

FROM deleted D

INNER JOIN ReadersTakingBook RTB ON D.Ticket = RTB.Ticket

WHERE RTB.DateOfReceipt IS NULL

)

BEGIN

RAISERROR('Cannot delete reader because they have borrowed books.', 16, 1);

ROLLBACK TRANSACTION;

END

ELSE

BEGIN

DELETE FROM Readers

WHERE Ticket IN (SELECT Ticket FROM deleted);

END

END

GO

INSERT INTO Halls (Named, Size)

VALUES

('Study Room', 28),

('Innovation Hub', 15),

('Quiet Room', 35)

INSERT INTO Readers (Ticket, PIB, PasportID, DateOfBirst, Addressa, PhoneNumber, Science, AcademicDegree, HallID, DateInclude)

VALUES

(1, 'Bychok Vadym Viacheslavovych', '977577586', '2005-08-14', 'Kyiv, Ukraine', '050123456', 'Secondary', 'No', 1, '2022-08-14'),

(2, 'Pylypchuk Nazar Sergiyovich', '681860980', '1977-01-28', 'Lviv, Ukraine', '093123456', 'Higher', 'Yes', 2, '2021-11-17'),

(3, 'Lushuck Denis Yuriyovich', '555404640', '2003-03-18', 'Odessa, Ukraine', '067123456', 'Secondary', 'No', 3, '2023-01-10'),

(4, 'Kravchenko Oleksandr Volodymyrovych', '987654321', '1990-05-12', 'Kharkiv, Ukraine', '099123456', 'Higher', 'Yes', 3, '2022-03-24'),

(5, 'Shevchenko Kateryna Ivanivna', '501234567', '2011-08-16', 'Dnipro, Ukraine', '044123456', 'Primary' , 'No', 1, '2022-05-22'),

(6, 'Hryshchenko Yuriy Petrovych', '931112233', '2001-06-19', 'Vinnytsia, Ukraine', '095123456', 'Higher', 'Yes', 2, '2022-09-21'),

(7, 'Ivanenko Svitlana Olegivna', '677556677', '1970-03-20', 'Zhytomyr, Ukraine', '066123456', 'Higher', 'Yes', 2, '2022-12-07'),

(8, 'Bondarenko Oksana Mykolaivna', '503456789', '1998-04-01', 'Rivne, Ukraine', '095123457', 'Higher', 'No', 3, '2023-02-11'),

(9, 'Koval Andriy Serhiyovych', '632223344', '1984-12-10', 'Cherkasy, Ukraine', '067123457', 'Higher', 'Yes', 2, '2023-01-16'),

(10, 'Lytvynenko Halyna Volodymyrivna', '931009900', '2014-04-30', 'Khmelnytskyi, Ukraine', '063123456', 'Primary', 'No', 1, '2021-10-03'),

(11, 'Mykhailenko Artem Ihorovych', '931212121', '2006-10-17', 'Ivano-Frankivsk, Ukraine', '099123457', 'Secondary', 'No', 1, '2022-01-01'),

(12, 'Stasiuk Petro Volodymyrovych', '977878787', '2008-01-22', 'Zaporizhzhia, Ukraine' , '050123457', 'Secondary', 'No', 1, '2020-12-10'),

(13, 'Vasylenko Anastasiya Viktorivna', '999998877', '2000-11-03', 'Sumy, Ukraine', '073123456', 'Higher', 'Yes', 2, '2021-06-29'),

(14, 'Horshenin Volodymyr Viktorovych', '993398127', '2003-03-11', 'Kuiv, Ukraine', '073489764', 'Higher', 'No', 2, '2019-12-21'),

(15, 'Ivanenko Ivan Ivanovivh', '674566887', '1988-02-22', 'Zhytomyr, Ukraine', '000125606', 'Higher', 'Yes', 2, '2018-12-07'),

(16, 'Petrenko Petro Petrovych', '468213894', '1997-07-07', 'Lytsk, Ukraine', '843761259', 'Higher', 'No', 2, '2024-11-25');

INSERT INTO Books (Pressmark, Named, Author, Publisher, PublicationDate, Count1, Count2, Count3)

VALUES

(1, 'METRO 2033', 'Dmytro Oleksiiovych Hlukhovskyi', 'Eksmo', '2005-08-14', 12, 9, 8),

(2, 'The Witcher', 'Andrzej Sapkowski', 'SuperNowa', '1990-07-21', 10, 8, 6),

(3, 'God of War', 'J. M. Barlog', 'Sony Press', '2018-06-05', 14, 13, 10),

(4, '1984', 'George Orwell', 'Secker & Warburg', '1949-06-08', 13, 12, 7),

(5,'The Hobbit', 'J.R.R. Tolkien', 'Allen & Unwin', '1937-09-21', 8, 7, 6),

(6, 'The Shining', 'Stephen King', 'Doubleday', '1977-01-28', 12, 10, 7),

(7, 'The Handmaid''s Tale', 'Margaret Atwood', 'McClelland & Stewart', '1985-04-01', 11, 9, 8),

(8, 'To Kill a Mockingbird', 'Harper Lee', 'J.B. Lippincott & Co.', '1960-07-11', 9, 8, 7),

(9, 'The Da Vinci Code', 'Dan Brown', 'Doubleday', '2003-03-18', 14, 12, 10),

(10,'American Gods', 'Neil Gaiman', 'HarperCollins', '2001-06-19', 13, 12, 8),

(11, 'A Brief History of Humankind', 'Yuval Noah Harari', 'Harvill Secker', '2011-09-04', 12, 10, 9),

(12, 'The Great Gatsby', 'F. Scott Fitzgerald', 'Charles Scribner''s Sons', '1925-04-10', 10, 9, 8),

(13, 'Ready Player One', 'Ernest Cline', 'Crown Publishing', '2011-08-16', 9, 7, 6),

(14, 'War and Peace', 'Leo Tolstoy', 'The Russian Messenger', '1969-03-04', 12, 11, 9),

(15, 'Crime and Punishment', 'Fyodor Dostoevsky', 'The Russian Messenger', '1966-01-01', 11, 10, 8),

(16, 'Twenty Thousand Leagues Under Sea', 'Jules Verne', 'Pierre-Jules Hetzel', '1970-03-20', 12, 10, 8),

(17, 'Foundation', 'Isaac Asimov', 'Gnome Press', '1951-08-21', 14, 13, 9),

(18, 'Fahrenheit 451', 'Ray Bradbury', 'Ballantine Books', '1953-10-19', 11, 9, 7),

(19, 'Dune', 'Frank Herbert', 'Chilton Books', '1965-08-01', 13, 11, 9),

(20, 'The War of the Worlds', 'H.G. Wells', 'Heinemann', '1998-04-01', 10, 9, 7),

(21, 'Brave New World', 'Aldous Huxley', 'Chatto & Windus', '1932-12-31', 11, 10, 8),

(22, 'Frankenstein', 'Mary Shelley', 'Lackington, Hughes, Harding, Mavor & Jones', '1918-01-01', 12, 10, 9),

(23, 'The Iliad', 'Homer', 'Ancient Greece', '1910-01-06', 13, 12, 11),

(24, 'The Odyssey', 'Homer', 'Ancient Greece', '1920-07-11', 13, 12, 11),

(25, 'Pride and Prejudice', 'Jane Austen', 'T. Egerton', '1913-01-28', 14, 13, 12),

(26,'Jane Eyre', 'Charlotte Bronte', 'Smith, Elder & Co.', '1947-10-16', 11, 10, 8),

(27, 'Wuthering Heights', 'Emily Bronte', 'Thomas Cautley Newby', '1947-12-01', 12, 11, 10),

(28, 'A Tale of Two Cities', 'Charles Dickens', 'Chapman & Hall', '1959-04-30', 13, 11, 9),

(29, 'The Adventures of Huckleberry Finn', 'Mark Twain', 'Chatto & Windus', '1984-12-10', 11, 10, 8),

(30,'Les Misérables', 'Victor Hugo', 'A. Lacroix, Verboeckhoven & Cie.', '1962-06-01', 9, 8, 7),

(31, 'One Hundred Years of Solitude', 'Gabriel Garcia Marquez', 'Editorial Sudamericana', '1967-05-30', 10, 9, 8),

(32, 'The Catcher in the Rye', 'J.D. Salinger', 'Little, Brown and Company', '1951-07-16', 0, 1, 1),

(33, 'The', 'J', 'Little', '1914-03-26', 0, 0, 0),

(34, 'TheN', 'JOHN', 'Lile', '1999-12-11', 1, 4, 3),

(35, 'The2', 'J', 'Little', '1917-06-27', 3, 1, 4),

(36, 'TheN2', 'JOHN', 'Lile', '1999-12-11', 1, 4, 3),

(37, 'Zen4', 'AMD', 'AMD64', '1999-12-11', 2, 2, 2),

(39, 'METRO EXODUS', 'Dmytro Oleksiiovych Hlukhovskyi', 'Eksmo', '2019-11-24', 10, 5, 18);

--Triger1 - InsertedReadersTakingBook

------------INSERT ReadersTakingBook (Pressmark, Ticket, IssueDate)

------------VALUES (33, 15, '2024-12-07 12:00:00');

--Triger2 - CheckIssueDateBeforeDateInclude

------------INSERT ReadersTakingBook (Pressmark, Ticket, IssueDate)

------------VALUES (1, 16, '2024-10-01 12:00:00');

INSERT ReadersTakingBook (Pressmark, Ticket, IssueDate)

VALUES (1, 1, '2024-10-01 12:00:00');

INSERT ReadersTakingBook (Pressmark, Ticket, IssueDate)

VALUES (36, 1, '2024-10-01 12:00:00');

INSERT ReadersTakingBook (Pressmark, Ticket, IssueDate, DateOfReceipt)

VALUES (2, 2, '2024-10-02 13:00:00', '2024-10-06 16:00:00');

INSERT ReadersTakingBook (Pressmark, Ticket, IssueDate, DateOfReceipt)

VALUES (3, 3, '2024-10-03 15:00:00', '2024-10-07 10:00:00');

INSERT ReadersTakingBook (Pressmark, Ticket, IssueDate, DateOfReceipt)

VALUES (5, 6, '2024-10-04 09:30:00', '2024-10-08 12:00:00');

INSERT ReadersTakingBook (Pressmark, Ticket, IssueDate)

VALUES (6, 7, '2024-10-05 16:00:00');

INSERT ReadersTakingBook (Pressmark, Ticket, IssueDate, DateOfReceipt)

VALUES (7, 10, '2024-10-07 12:00:00', '2024-10-11 15:00:00');

INSERT ReadersTakingBook (Pressmark, Ticket, IssueDate)

VALUES (8, 11, '2024-10-08 14:30:00');

INSERT ReadersTakingBook (Pressmark, Ticket, IssueDate, DateOfReceipt)

VALUES (9, 12, '2024-10-09 10:00:00', '2024-10-13 11:00:00');

…

GO

DROP PROC IF EXISTS BooksInReaders;

GO

CREATE PROC BooksInReaders @RDR VARCHAR(35)

AS

SELECT Named, PIB

FROM Books B INNER JOIN ReadersTakingBook RTB

ON B.Pressmark = RTB.Pressmark

INNER JOIN Readers R

ON RTB.Ticket= R.Ticket

WHERE PIB = @RDR

AND DateOfReceipt IS NOT NULL;

GO

EXEC BooksInReaders 'Bychok Vadym Viacheslavovych';

GO

DROP PROC IF EXISTS PressmarkNamed;

GO

CREATE PROC PressmarkNamed @PM VARCHAR(35)

AS

SELECT Pressmark, Named

FROM Books b

WHERE Pressmark = @PM;

GO

EXEC PressmarkNamed 2

GO

DROP PROC IF EXISTS NamedPressmark

GO

CREATE PROC NamedPressmark @NM VARCHAR(35)

AS

SELECT Pressmark, Named

FROM Books b

WHERE Named = @NM;

GO

EXEC NamedPressmark 'METRO 2033'

GO

DROP PROC IF EXISTS DateBookReader;

GO

CREATE PROC DateBookReader @BOOK VARCHAR(35), @READER VARCHAR(35)

AS

SELECT Named, PIB, IssueDate

FROM Books B INNER JOIN ReadersTakingBook RTB

ON B.Pressmark = RTB.Pressmark

INNER JOIN Readers R

ON RTB.Ticket= R.Ticket

WHERE B.Named = @BOOK

AND R.PIB = @READER;

GO

EXEC DateBookReader 'METRO 2033', 'Bychok Vadym Viacheslavovych';

GO

DROP VIEW IF EXISTS MonthsReader;

GO

CREATE VIEW MonthsReader

AS

SELECT Named, PIB, IssueDate

FROM Books B INNER JOIN ReadersTakingBook RTB

ON B.Pressmark = RTB.Pressmark

INNER JOIN Readers R

ON RTB.Ticket = R.Ticket

WHERE RTB.IssueDate < DATEADD(MONTH, -1, GETDATE())

AND RTB.DateOfReceipt IS NOT NULL;

GO

SELECT \* FROM MonthsReader

GO

DROP VIEW IF EXISTS CountBooks2;

GO

CREATE VIEW CountBooks2

AS

SELECT Named, PIB, IssueDate

FROM Books B INNER JOIN ReadersTakingBook RTB

ON B.Pressmark = RTB.Pressmark

INNER JOIN Readers R

ON RTB.Ticket = R.Ticket

WHERE B.Pressmark IN (SELECT Pressmark

FROM Books

WHERE (Count1 + Count2 + Count3) <= 2)

AND RTB.DateOfReceipt IS NOT NULL;

GO

SELECT \* FROM CountBooks2;

GO

DROP VIEW IF EXISTS CountAllReaders;

GO

CREATE VIEW CountAllReaders

AS

SELECT COUNT(\*) AS All\_Readers

FROM Readers;

GO

SELECT \* FROM CountAllReaders;

GO

DROP VIEW IF EXISTS ReadersYounger20;

GO

CREATE VIEW ReadersYounger20

AS

SELECT PIB, DateOfBirst

FROM Readers

WHERE DateOfBirst > DATEADD(YEAR, -20, GETDATE());

GO

SELECT \* FROM ReadersYounger20;

GO

DROP VIEW IF EXISTS ProcentScience;

GO

CREATE VIEW ProcentScience

AS

SELECT Science,

COUNT(\*) AS CountOfReaders,

(COUNT(\*) \* 100.0 / (SELECT COUNT(\*) FROM Readers)) AS Percentage

FROM Readers

GROUP BY Science;

GO

SELECT \* FROM ProcentScience;

GO

DROP VIEW IF EXISTS ReadersWhoDont;

GO

CREATE VIEW ReadersWhoDont

AS

SELECT R.\*

FROM Readers R

WHERE NOT EXISTS (

SELECT 1

FROM ReadersTakingBook RTB

WHERE RTB.Ticket = R.Ticket AND RTB.DateOfReceipt IS NULL

);

GO

DROP PROC IF EXISTS DeleteMoreYear;

GO

CREATE PROC DeleteMoreYear @Ticket VARCHAR(50)

AS

DELETE FROM Readers

WHERE DateInclude < DATEADD(YEAR, -3, GETDATE())

AND Ticket = @Ticket;

GO

SELECT \* FROM ReadersWhoDont;

EXEC DeleteMoreYear 10

GO

DROP TABLE IF EXISTS RERegistrationBuffer;

DROP PROC IF EXISTS RERegistration;

GO

CREATE PROC RERegistration @R INT

AS

UPDATE Readers SET Ticket = Ticket + 100 WHERE Ticket = @R;

GO

EXEC RERegistration 8;

SELECT \* FROM Readers;

--GO

DROP PROC IF EXISTS REClasification;

GO

CREATE PROC REClasification @B INT

AS

UPDATE Books SET Pressmark = Pressmark + 1000 WHERE Pressmark = @B

GO

SELECT \* FROM Books

WHERE Pressmark = 1;

EXEC REClasification 1

SELECT \* FROM Books;

GO

DROP PROC IF EXISTS DeleteOldBooks;

GO

CREATE PROC DeleteOldBooks @OB INT

AS

IF @OB IN (

SELECT B.Pressmark

FROM Books B

INNER JOIN ReadersTakingBook RTB ON B.Pressmark = RTB.Pressmark

INNER JOIN Readers R ON RTB.Ticket = R.Ticket

WHERE RTB.DateOfReceipt IS NULL

AND PIB IS NOT NULL)

BEGIN

--PRINT('Can`t delete book if she isset in readers!!!')

RAISERROR('Can`t delete book if she isset in readers!!!', 16, 1);

END;

ELSE

BEGIN

IF @OB IN (

SELECT Pressmark

FROM Books

WHERE Count1 + Count2 + Count3 = 0)

OR @OB IN (

SELECT Pressmark

FROM Books

WHERE PublicationDate < DATEADD(YEAR, -75, GETDATE()))

BEGIN

DELETE FROM Books

WHERE Pressmark = @OB;

END;

ELSE

BEGIN

--PRINT('The book is not lost or old')

RAISERROR('The book is not lost or old', 16, 1);

END;

END;

GO

--SELECT \* FROM Books;

--EXEC DeleteOldBooks 33;

--GO

--EXEC DeleteOldBooks 1;

--GO

--EXEC DeleteOldBooks 34;

--GO

--EXEC DeleteOldBooks 35;

--GO

--EXEC DeleteOldBooks 36;

--GO

--EXEC DeleteOldBooks 37;

--GO

--SELECT \* FROM Books;

--GO

DROP PROC IF EXISTS CountBooksAuthHall;

GO

GO

CREATE PROC CountBooksAuthHall @A VARCHAR(50), @Hall INT

AS

SELECT Author,

CASE

WHEN @Hall = 1 THEN SUM(Count1)

WHEN @Hall = 2 THEN SUM(Count2)

WHEN @Hall = 3 THEN SUM(Count3)

ELSE 0

END AS TotalBooks

FROM Books

WHERE Author = @A

GROUP BY Author;

GO

--EXEC CountBooksAuthHall 'Dmytro Oleksiiovych Hlukhovskyi', 2;

--SELECT \* FROM Books;

GO

DROP VIEW IF EXISTS ReportM;

GO

CREATE VIEW ReportM

AS

SELECT 'BooksHalls1' Metric, SUM(Count1) AS Value FROM Books

UNION ALL SELECT 'BooksHalls2', SUM(Count2) FROM Books

UNION ALL SELECT 'BooksHalls3', SUM(Count3) FROM Books

UNION ALL SELECT 'TotalBooks', CAST(SUM(Count1 + Count2 + Count3) AS INT) FROM Books

UNION ALL SELECT 'ReadersHall1', COUNT(\*) FROM Readers WHERE HallID = 1

UNION ALL SELECT 'ReadersHall2', COUNT(\*) FROM Readers WHERE HallID = 2

UNION ALL SELECT 'ReadersHall3', COUNT(\*) FROM Readers WHERE HallID = 3

UNION ALL SELECT 'TotalReaders', COUNT(\*) FROM Readers

UNION ALL SELECT 'NewReaders', COUNT(\*) FROM Readers WHERE DateInclude > DATEADD(MONTH, -1, GETDATE())

UNION ALL SELECT 'BooksIssuedLastMonth', COUNT(\*) AS Value FROM ReadersTakingBook RTB INNER JOIN Books B ON B.Pressmark = RTB.Pressmark

WHERE RTB.IssueDate > DATEADD(MONTH, -1, GETDATE())

UNION ALL SELECT 'COUNT BOOKS ISSUED THIS MONTH:', (SELECT COUNT(\*) AS Value FROM ReadersTakingBook RTB INNER JOIN Books B ON B.Pressmark = RTB.Pressmark

WHERE RTB.IssueDate > DATEADD(MONTH, -1, GETDATE()))

UNION ALL SELECT B.Named, COUNT(\*) AS Value FROM ReadersTakingBook RTB INNER JOIN Books B ON B.Pressmark = RTB.Pressmark

WHERE RTB.IssueDate > DATEADD(MONTH, -1, GETDATE()) GROUP BY B.Named

UNION ALL SELECT 'READERS WHO DIDNT TAKE BOOKS LAST MONTH AND THEIR HallID', (SELECT COUNT(\*) AS Value FROM Readers

WHERE PIB NOT IN (SELECT PIB FROM Readers R INNER JOIN ReadersTakingBook RTB ON R.Ticket = RTB.Ticket WHERE RTB.IssueDate > DATEADD(MONTH, -1, GETDATE())))

UNION ALL SELECT PIB, HallID FROM Readers WHERE PIB NOT IN (SELECT PIB FROM Readers R INNER JOIN ReadersTakingBook RTB ON R.Ticket = RTB.Ticket

WHERE RTB.IssueDate > DATEADD(MONTH, -1, GETDATE()));

GO

SELECT \* FROM ReportM;

IF NOT EXISTS (SELECT 1 FROM sys.server\_principals WHERE name = 'Librarian')

BEGIN

CREATE LOGIN Librarian

WITH PASSWORD = 'kursova',

DEFAULT\_DATABASE = kursova;

CREATE USER Librarian FOR LOGIN Librarian;

END;

GRANT SELECT ON SCHEMA::dbo TO Librarian;

GRANT INSERT ON dbo.Readers TO Librarian;

GRANT INSERT, DELETE ON dbo.Books TO Librarian;

GRANT INSERT ON dbo.ReadersTakingBook TO Librarian;

GRANT EXEC ON BooksInReaders TO Librarian;

GRANT EXEC ON DateBookReader TO Librarian;

GRANT EXEC ON NamedPressmark TO Librarian;

GRANT EXEC ON PressmarkNamed TO Librarian;

GRANT EXEC ON DeleteMoreYear TO Librarian;

GRANT EXEC ON RERegistration TO Librarian;

GRANT EXEC ON REClasification TO Librarian;

GRANT EXEC ON DeleteOldBooks TO Librarian;

GRANT EXEC ON CountBooksAuthHall TO Librarian;

GO

DELETE FROM Readers WHERE Ticket = 1;

GO

--DELETE FROM Readers WHERE Ticket = 10;

SELECT \* FROM ReadersWhoDont;

SELECT \* FROM Readers;

SELECT \* FROM Books;

SELECT \* FROM Halls;

SELECT \* FROM ReadersTakingBook;

# Опис реалізації завдань відповідно обраного варіанту

У цьому розділі продемонструємо результати реалізації.

Таблиця читачів:  
Зображення, що містить текст, знімок екрана, число, Шрифт

Автоматично згенерований опис

Таблиця книг:  
Зображення, що містить текст, знімок екрана, число, меню

Автоматично згенерований опис

Таблиця зали:  
Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

Таблиця зв’язку через який реалізовано взяття книг:

Зображення, що містить текст, знімок екрана, меню, число

Автоматично згенерований опис…

Спрацювання тригерів:

Зображення, що містить текст, знімок екрана, Шрифт, ряд

Автоматично згенерований опис

Зображення, що містить текст, Шрифт, ряд, число

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт, ряд

Автоматично згенерований опис

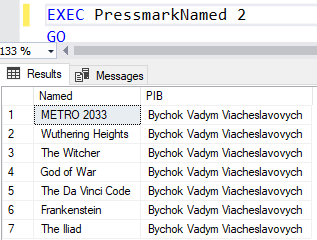
Зображення, що містить текст, знімок екрана, Шрифт, число

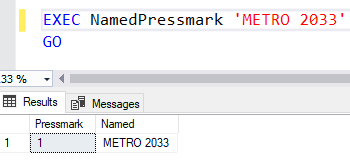
Автоматично згенерований опис

Відомості про поточний стан бібліотеки що можуть знадобитися Бібліотекарю (Представлення, там де немає варіативного пошуку, і процедури, для конкретного пошуку за аргументом):

Які книги закріплені за певним читачем?  
Зображення, що містить текст, електроніка, знімок екрана, програмне забезпечення

Автоматично згенерований опис

Як називається книга з заданим шифром?  


Який шифр у книги з заданим назвою?  


Коли книга була закріплена за читачем?  
Зображення, що містить текст, Шрифт, ряд, число

Автоматично згенерований опис

Хто з читачів взяв книгу більше місяця тому?  
Зображення, що містить текст, знімок екрана, число, Шрифт

Автоматично згенерований опис

За ким із читачів закріплені книги, кількість примірників яких в бібліотеці не перевищує 2?  
Зображення, що містить текст, знімок екрана, Шрифт, програмне забезпечення

Автоматично згенерований опис

Яке число читачів користується бібліотекою?  
Зображення, що містить текст, знімок екрана, Шрифт, ряд

Автоматично згенерований опис

Скільки в бібліотеці читачів молодших 20 років?  
Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

Скільки читачів у відсотковому відношенні мають початкову освіту, середню, вищу, вчений ступінь?  
Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

Виключити зі списку читачів людей, які записалися в бібліотеку більше року тому і які не пройшли перереєстрацію.  
Зображення, що містить текст, число, Шрифт, програмне забезпечення

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, число, Шрифт

Автоматично згенерований опис

Запис в бібліотеку нового читача виконується через звичайний INSERT.

Списати стару або втрачену книгу.  
Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, число, монітор

Автоматично згенерований опис

Видача довідки про кількість книг певного автора в читальному залі:  
Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

Звіт про роботу бібліотеки протягом місяця. Звіт включає в себе наступну інформацію: кількість книг і читачів на поточний день в кожному із залів і в бібліотеці в цілому, кількість читачів, які записалися в бібліотеку за звітний місяць, які книги і скільки разів були взяті за цього місяця, та хто з читачів не брав книг:  
Зображення, що містить текст, знімок екрана, програмне забезпечення, число

Автоматично згенерований опис

# Висновок

У ході виконнання курсової роботи ми розробили інформаційну систему бібліотеки на основі реляційної бази даних, яка повністю відповідає поставленим вимогам та забезпечує автоматизацію всіх ключових бібліотечних процесів.

Наші досягнені результати:

Ми створили ефективну структуру бази даних, відповідну до варіанту 1 курсової роботи, забезпечуючи надійне зберігання інформації про книжковий фонд, читачів та читальні, оптимальну організацію даних для швидкого пошуку та обробки інформації, підтримку цілісності даних через систему обмежень та тригерів.

Ми реалізували всі необхідні функціїї та задачі. Проводимо повний облік книжкового фонду, керуємо читацькими записами, автоматизували процеси видачі та повернення книг, сформували можливість видачі довідки і повноцінного звіту за останній місяць.

Також підвищили ефективність роботи бібліотеки тим що скоротили витрату часу на пошук необхідної інформації, автоматизували рутинні операції та зменшили ймовірність помилок при обліку.