

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Інститут комп'ютерних наук та інформаційних технологій
Кафедра систем штучного інтелекту



РОЗРАХУНКОВА РОБОТА
з дисципліни «Організація баз даних та знань»
на тему:
«Електронна бібліотека»

Виконала студентка групи КН-211

Досяк Ірина

Бали	Дата
	22.05.2020

Викладач

Мельникова Н.І.

ЗМІСТ

ТЕМА ПРОЕКТУ	3
ВСТУП.....	4
ЛОГІЧНА СХЕМА БД ПРОЕКТУ	6
ОПИС СТРУКТУРИ БД	7
ФІЗИЧНА МОДЕЛЬ БД	10
ДІЛОВА МОДЕЛЬ	14
ЗАПИТИ БД	17
ВИСНОВКИ	26
СПИСОК ВИКОРИСТАНИХ ЛІБЕРАТУРНИХ ДЖЕРЕЛ	15

ТЕМА ПРОЕКТУ

Темою проекту є онлайн бібліотека BookSpace.

BookSpace – це місце, де кожен зможе знайти щось для себе, адже вона наповнена світовими бестселерами, які відомі у всьому світі; класичною літературою, про яку хоч раз в житті чула кожна людина

Крім того, використовуючи BookSpace можна створити власну бібліотеку, щоб завжди тримати усі свої улюблені чи ті, які хочеш прочитати книги в одному місці.

Також можна залишати відгуки про книги, щоб допомогти іншим користувачам швидше вибрати необхідну. Якщо користувач забув пароль від своєї особистої бібліотеки, він зможе його відновити. Для цього потрібно вказати електронну пошту. Якщо така існує в бд, то користувач з посилання одноразового використання на зміну пароллю.

ВСТУП

На мою думку, книги – актуальність, яка не підвладна часу. Їх використовують для навчання, розваг, саморозвитку, читають дітям. Крім того, у сучасний період комп'ютеризації зростає рівень психологічного напруження. Книга допомагає відволіктись від усіх проблем і відпочити.

Щодо статистики, щороку попит на книги зростає майже на 2%, і ця тенденція спостерігається вже кілька років підряд. Деякі країни мають високий рівень освіти, а також стійкі традиції читання, що формують попит на книги загалом.

Отже книги, без сумніву, є актуальними. Відкритим питанням залишається формат книги: паперовий чи електронний.

Багато людей вітають появу нових технологій і читають електронні книги. Основними перевагами використання такого типу книг є:

- можливість економити;
- легкість;
- компактність, що дозволяє взяти з собою в подорож декілька книг;
- можливість вбудувати медіа-матеріали, що служить гарним доповненням до книг;
- легко відновити втрачену книгу;
- можливість підібрати шрифт.

Отже, електронні книги є дійсно зручним способом насолодитись читанням. Крім того сьогодні дуже ціниться час і кожен намагається оптимізувати все, у тому числі й пошук бажаних книг. Звідси появляється актуальність електронних бібліотек.

На сьогоднішній день існує багато веб-сайтів, які пропонують завантаження електронних книг, але вони, як правило, значно нижчі за якість паперових книг, що робить їх складнішими та менш приємними для читання.

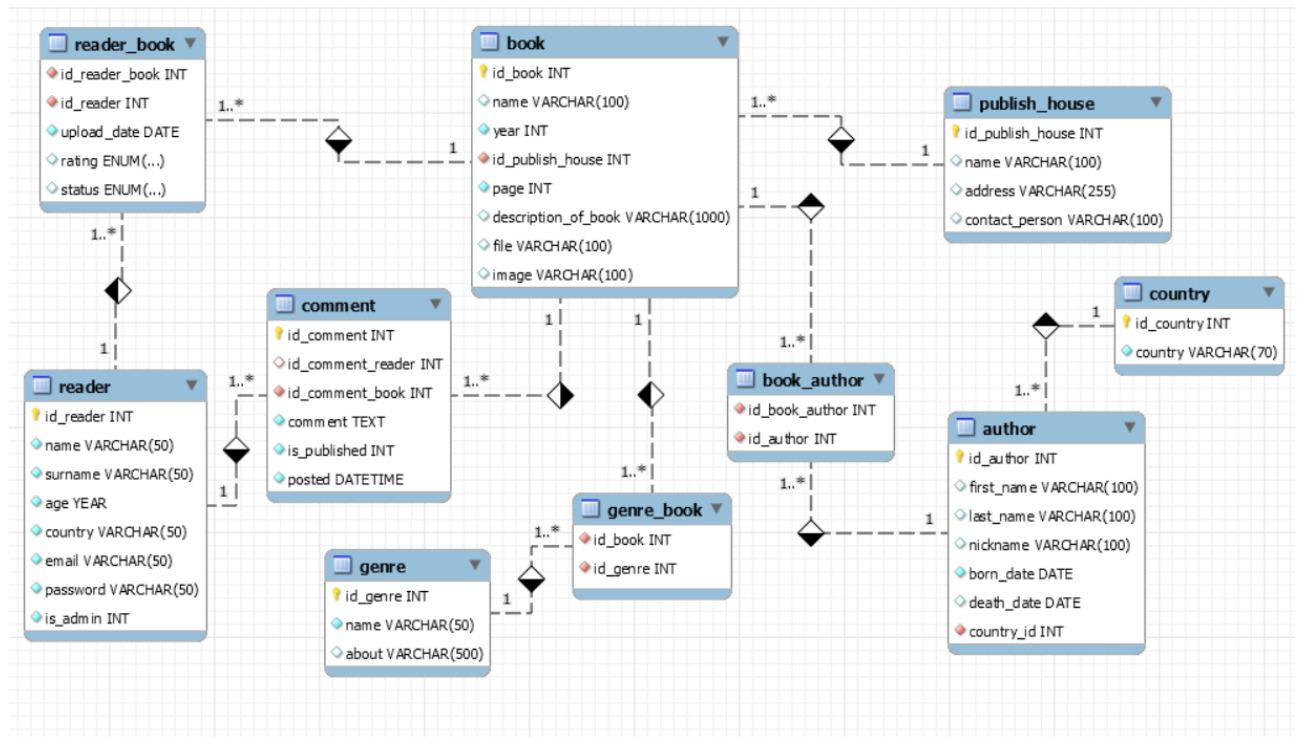
Проаналізувавши ситуацію, ми вирішити це змінити. Наша мета - опублікувати невеликий вибір, проте справді високоякісних електронних книг - справжню альтернативу для читачів, які хочуть насолоджуватися читанням книги.

Отже, BookSpace - це прекрасний спосіб відволіктися від усіх проблем та зануритися у світ цікавих персонажів, надзвичайних подій, загадкових розслідувань та безмежної любові.

Крім того, ми дозволяємо користувачеві створити власну бібліотеку, щоб він завжди мав усі свої улюблені чи ті, які хоче прочитати, книги в одному місці і міг прочитати їх у будь-який час.

Користувач також може залишати відгуки про книги, щоб допомогти іншим швидше вибрати необхідну.

ЛОГІЧНА СХЕМА БД ПРОЕКТУ



ОПИС СТРУКТУРИ БД

База даних складається з десяти таблиць: book, author, country, publish_house, genre, reader, reader_book, genre_book, book_author, comment.

Таблиця publish_house містить такі поля: назва видавництва типу varchar (може містити до 100 символів, є обов'язковим і унікальним), адреса типу varchar (може містити до 255 символів, є обов'язковим), контактна особа типу varchar (може містити до 100 символів, є обов'язковим) та первинний ключ - id видавництва типу int (є обов'язковим, заповнюється автоматично).

Таблиця genre містить такі поля: назва жанру типу varchar (може містити до 50 символів, є обов'язковим і унікальним), коротке пояснення типу varchar (може містити до 500 символів, є обов'язковим) та первинний ключ - id жанру типу int (є обов'язковим, заповнюється автоматично).

Таблиця country містить такі поля: назва країни типу varchar (може містити до 70 символів, є обов'язковим і унікальним) та первинний ключ - id країни типу int (є обов'язковим, заповнюється автоматично).

Таблиця book містить такі поля: ім'я типу varchar (може містити до 100 символів, є обов'язковим), рік видачі книги типу int (не є обов'язковим), кількість сторінок типу int (є обов'язковим), опис книги типу varchar (може містити до 1000 символів, є обов'язковим і унікальним), посилання на файл типу varchar (може містити до 100 символів, є обов'язковим), посилання на зображення типу varchar (може містити до 100 символів, є обов'язковим). А також первинний ключ - id книги типу int (є обов'язковим, заповнюється автоматично) та зовнішній ключ id видавництва типу int (є обов'язковим). Я використала каскадне оновлення та видалення без змін.

Таблиця author містить такі поля: ім'я та прізвище автора типу varchar (можуть містити до 100 символів, є обов'язковим), псевдонім типу varchar (може містити до 100 символів, є обов'язковим і унікальним), дата народження та смерті типу date (не є обов'язковим). А також первинний ключ - id книги типу int (є обов'язковим, заповнюється автоматично) та зовнішній ключ id країни типу int (є обов'язковим). Я використала оновлення та видалення без змін.

Таблиця book_author є проміжною, містить зовнішні ключі `id_book_author` та `id_author` типу `int` (є обов'язковими).

Таблиця genre_book є проміжною, містить зовнішні ключі `id_book` та `id_genre` типу `int` (є обов'язковими).

Таблиця reader містить такі поля: ім'я користувача типу `varchar` (може містити до 50 символів, є обов'язковим), прізвище користувача типу `varchar` (може містити до 50 символів, є обов'язковим), дату народження користувача типу `date` (є обов'язковим), пошта користувача типу `varchar` (може містити до 50 символів, є обов'язковим), емейл користувача типу `varchar` (може містити до 50 символів, є обов'язковим і унікальним), перевірка на доступ адміністратора типу `bool` (за замовчуванням отримує 0, є обов'язковим). А також первинний ключ - `id` користувача типу `int` (є обов'язковим, заповнюється автоматично).

Таблиця reader_book є проміжною, містить такі поля: дата додання книги в особисту бібліотеку (є обов'язковим, за замовчуванням отримує дату в даний момент), рейтинг типу `enum` (не є обов'язковим), статус типу `enum` (не є обов'язковим), дата створення коментаря (є обов'язковим, за замовчуванням отримує дату в даний момент), первинний ключ – `id_reader_book` типу `int` (є обов'язковим, заповнюється автоматично). А також зовнішні ключі `id_reader_book` та `id_reader` типу `int` (є обов'язковими). Я використала каскадне оновлення та видалення.

Таблиця comment є проміжною, містить такі поля: коментар типу `text` (є обов'язковим), дата створення коментаря (є обов'язковим, за замовчуванням отримує дату в даний момент), первинний ключ - `id` коментаря типу `int` (є обов'язковим, заповнюється автоматично). А також зовнішній ключ `id_comment_book` типу `int` (є обов'язковим) та `id_comment_reader` типу `int` (не є обов'язковим). Я використала каскадне оновлення та видалення для ключа `id_comment_book`. Для ключа `id_comment_reader` я використала каскадне оновлення та значення нуль при видаленні користувача.

Зв'язок один-до-багатьох використаний між таблицями book і publish_house, а також між author та country. Отже, одному запису з однієї таблиці відповідає кілька записів у іншій таблиці.

Зв'язок багато-до-багатьох використаний між таблицями book і author, book і genre а також між book та reader. Багато-до-багатьох — безлічі записів з однієї таблиці відповідає кілька записів в іншій таблиці. Відношення багато-до-багатьох замінюємо двома двонаправленими зв'язками один-до-багатьох, з'єднуючи їх через проміжну таблицю.

ФІЗИЧНА МОДЕЛЬ БД

```
CREATE SCHEMA IF NOT EXISTS library_online DEFAULT CHARACTER SET utf8
COLLATE utf8_general_ci;
USE library_online;
```

```
CREATE TABLE library_online.publish_house (
    id_publish_house int NOT NULL AUTO_INCREMENT,
    name VARCHAR(100) UNIQUE NOT NULL,
    address VARCHAR(255) NOT NULL,
    contact_person VARCHAR(100) NOT NULL,
    PRIMARY KEY (`id_publish_house`)
);
```

```
CREATE TABLE library_online.genre (
    id_genre int NOT NULL AUTO_INCREMENT,
    name VARCHAR(50) UNIQUE NOT NULL,
    about VARCHAR(500) NOT NULL,
    PRIMARY KEY (`id_genre`)
);
```

```
CREATE TABLE library_online.country (
    id_country int NOT NULL AUTO_INCREMENT,
    country VARCHAR(70) UNIQUE NOT NULL,
    PRIMARY KEY (`id_country`)
);
```

```
CREATE TABLE library_online.book (
    id_book int NOT NULL AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    year int(4) NULL,
    id_publish_house int NOT NULL,
    page int NOT NULL,
    description_of_book VARCHAR(1000) NOT NULL UNIQUE,
    file VARCHAR(100) NOT NULL,
    image VARCHAR(100) NOT NULL,
    PRIMARY KEY (`id_book`),
    INDEX id_publish_house_idx (`id_publish_house` ASC) VISIBLE,
    CONSTRAINT id_publish_house
```

```

        FOREIGN KEY (`id_publish_house`)
REFERENCES library_online.publish_house (`id_publish_house`)
        ON DELETE NO ACTION
        ON UPDATE CASCADE
);

CREATE TABLE library_online.author (
    id_author int NOT NULL AUTO_INCREMENT,
    first_name VARCHAR(100) NOT NULL,
    last_name VARCHAR(100) NOT NULL,
    nickname VARCHAR(100) NOT NULL UNIQUE,
    born_date DATE NULL,
    death_date DATE NULL,
    country_id int NOT NULL,
    PRIMARY KEY (`id_author`),
    INDEX id_country_idx (`country_id` ASC) VISIBLE,
    CONSTRAINT id_country
        FOREIGN KEY (`country_id`)
REFERENCES library_online.country (`id_country`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
);

CREATE TABLE library_online.book_author (
    id_book_author int NOT NULL,
    id_author int NOT NULL,
    INDEX id_book_author_idx (`id_book_author` ASC) VISIBLE,
    INDEX id_author_idx (`id_author` ASC) VISIBLE,
    CONSTRAINT id_book_author
        FOREIGN KEY (`id_book_author`)
REFERENCES library_online.book (`id_book`)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    CONSTRAINT id_author
        FOREIGN KEY (`id_author`)
REFERENCES library_online.author (`id_author`)
        ON DELETE CASCADE

```

```

        ON UPDATE CASCADE
    );

CREATE TABLE library_online.genre_book (
    id_book int NOT NULL,
    id_genre int NOT NULL,
    INDEX id_book_idx (`id_book` ASC) VISIBLE,
    INDEX id_genre_idx (`id_genre` ASC) VISIBLE,
    CONSTRAINT id_book
        FOREIGN KEY (`id_book`)
        REFERENCES library_online.book (`id_book`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT id_genre
        FOREIGN KEY (`id_genre`)
        REFERENCES library_online.genre (`id_genre`)
        ON DELETE NO ACTION
        ON UPDATE CASCADE
);

CREATE TABLE library_online.reader (
    id_reader int NOT NULL AUTO_INCREMENT,
    name VARCHAR(50) NOT NULL,
    surname VARCHAR(50) NOT NULL,
    date_of_birth DATE NOT NULL,
    email VARCHAR(50) NOT NULL UNIQUE,
    password VARCHAR(50) NOT NULL,
    is_admin bool NOT NULL DEFAULT 0,
    PRIMARY KEY (`id_reader`)
);

CREATE TABLE library_online.reader_book (
    id int NOT NULL AUTO_INCREMENT,
    id_reader_book int NOT NULL,
    id_reader int NOT NULL,
    upload_date DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    rating ENUM('1','2','3','4','5') NULL,
    status ENUM ('not_read', 'want_to_read', 'in_progress', 'read') NULL,

```

```

PRIMARY KEY (`id`),
INDEX id_reader_book_idx (`id_reader_book` ASC) VISIBLE,
INDEX id_reader_idx (`id_reader` ASC) VISIBLE,
CONSTRAINT id_reader_book
    FOREIGN KEY (`id_reader_book`)
    REFERENCES library_online.book (`id_book`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
CONSTRAINT id_reader
    FOREIGN KEY (`id_reader`)
    REFERENCES library_online.reader (`id_reader`)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

CREATE TABLE library_online.comment (
    id_comment int NOT NULL AUTO_INCREMENT,
    id_comment_reader int NULL,
    id_comment_book int NOT NULL,
    comment TEXT NOT NULL,
    posted DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (`id_comment`),
INDEX id_comment_reader_idx (`id_comment_reader` ASC) VISIBLE,
INDEX id_comment_book_idx (`id_comment_book` ASC) VISIBLE,
CONSTRAINT id_comment_reader
    FOREIGN KEY (`id_comment_reader`)
    REFERENCES library_online.reader (`id_reader`)
    ON DELETE SET NULL
    ON UPDATE CASCADE,
CONSTRAINT id_comment_book
    FOREIGN KEY (`id_comment_book`)
    REFERENCES library_online.book (`id_book`)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

```

ДІЛОВА МОДЕЛЬ

Таблиці	book	author	genre	country	comment	book_ author	book_ genre	reader_ book	publish_ house	reader
Функції										
Особиста бібліотека користувача	*							*		*
Список коментарів до книги	*				*					*
Список книг і їх зображень	*									
Реєстрація										*
Детальна інформація про автора книги	*	*		*		*				
Детальна інформація про жанр книги	*		*				*			
Детальна інформація про видавництво книги	*								*	
Детальна інформація про книгу	*									
Список авторів		*								
Список жанрів			*							
Пошук книг за автором	*	*				*				

Пошук книг за жанром	*		*				*			
Отримати файл книги	*									
Підрахунок віку користувача										*
Особисті дані користувача										*
Кодування паролю										*

- Функція «особиста бібліотека користувача» відсортовує книги , які додає читач до своєї бібліотеки. Для цього потрібно подати id користувача. Вона об'єднує таблиці book, reader_book та reader.
- Функція «список коментарів до книги» показує коментарі до книг, коли вони були написані, до якої книги та яким читачем. Сортують коментарі за датою за допомогою ORDER BY та DESC.
- Функція «список книг і їх зображень» відтворює список всіх книг, які є в бібліотеці, та їх зображень в алфавітному порядку. Сортують за полем name за допомогою ORDER BY.
- Функція «реєстрація» додає нового читача в базу даних для цього ми використовуємо INSERT INTO в таблицю reader.
- Функція «детальна інформація про автора книги» відтворює детальну інформацію про автора чи авторів конкретної книги. Для цього потрібно вказати id книги. Якщо авторів декілька, то виводимо їх в алфавітному порядку для цього сортуємо за полем first_name за допомогою ORDER BY.
- Функція «детальна інформація про жанр книги» відтворює детальну інформацію про жанр чи жанри конкретної книги. Для цього потрібно вказати id книги. Якщо жанрів декілька, то виводимо їх в алфавітному порядку для цього сортуємо за полем name за допомогою ORDER BY.

- Функція «детальна інформація про видавництво книги» відтворює детальну інформацію про видавництво конкретної книги. Для цього потрібно вказати id книги.
- Функція «список авторів» показує всіх авторів в бд.
- Функція «список жанрів» показує всі жанри в бд.
- Функція «пошук книги за автором» показує всі книги за вказаним автором. Для цього необхідно обрати автора.
- Функція «пошук книги за жанром» показує всі книги за вказаним жанром. Для цього необхідно обрати жанр.
- Функція «отримати файл книги» подає шлях до файла обраної книги. Для цього необхідно подати id книги.
- Функція «підрахунок віку користувача» рахує вік користувача. Для цього потрібно подати дату народження користувача.
- Функція «особисті дані користувача» показує ім'я, прізвище, вік та емейл певного користувача. Для цього потрібно подати id користувача.
- Функція «кодування паролів» кодує пароль функцією md5 для цього створюємо тригер, який перед додаванням даних до таблиці reader буде кодувати атрибут password та заносити вже закодований пароль в таблицю.

ЗАПИТИ БД

Відсортований список всіх наявних книг та їх зображень

```
SELECT book.name, book.image FROM book
```

```
ORDER BY name
```

Результат:

	name	image
►	A Tale of Two Cities	images/a-tale-of-two-cities.jpg
	Alice`s Adventures in Wonderland	images/alices-adventures-in-wonderland.jpg
	All the Light We Cannot See	images/all-the-light-we-cannot-see.jpg
	Anna Karenina	images/anna-karenina.jpg
	Atlas Shrugged	images/atlas-shrugged.jpg
	Fahrenheit 451	images/fahrenheit-451.jpg
	Hamlet	images/hamlet.jpg
	Harry Potter and the Deathly Hallows	images/harry-potter-and-the-deathly-hallows.jpg
	Harry Potter And The Philosopher`s Stone	images/harry-potter-and-the-philosophers-ston...
	Harry Potter And The Prisoner of Azkaban	images/harry-potter-and-the-prisoner-of-azkab...
	In Search of Lost Time	images/in-search-of-lost-time.jpg
	Jane Eyre	images/jane-eyre.jpg
	Murder on the Orient Express	images/murder-on-the-orient-express.jpeg
	My Brilliant Friend	images/my-brilliant-friend.jpeg
	Picture of Dorian Gray	images/the-picture-of-dorian-gray.jpeg
	Robinson Crusoe	images/robinson-crusoe.jpeg
	Sapiens: A Brief History Of Humankind	images/sapiens-a-brief-history-of-humankind.jpeg
	The Alchemist	images/the-alchemist.jpeg
	The Da Vinci Code	images/the-da-vinci-code.jpeg
	The Forest Song	images/forest-song.jpeg
	The Great Gatsby	images/the-great-gatsby.jpeg
	The Handmaid`s Tale	images/the-handmaids-tale.jpeg
	The Little Prince	images/the-little-prince.jpeg
	The Lord of the Rings	images/the-lord-of-the-rings.jpeg
	The Master and Margarita	images/the-master-and-margarita.jpeg

Список авторів

```
SELECT author.nickname as author
```

```
FROM author
```

```
ORDER BY author.nickname
```

Результат:

	author
►	Agatha Christie
	Anthony Doerr
	Antoine de Saint-Exup��ry
	Ayn Rand
	Charles Dickens
	Charlotte Bronte
	Dan Brown
	Daniel Defoe
	Elena Ferrante
	Ernest Hemingway
	F. Scott Fitzgerald
	Ivan Franko
	J. K. Rowling
	J. R. R. Tolkien

Список жанрів

```
SELECT genre.name as genre
FROM genre
ORDER BY genre.name
```

Результат:

genre
Adventure
Biographies and Autobiographies
Classics
Detective fiction
Drama
Fantasy
Historical
History
Horror
Literary Fiction
Mystery
Romance
Self-Help
Thriller

Пошук книг за жанром

```
DELIMITER //
CREATE PROCEDURE get_books_by_genre (IN genre_name char(30))
BEGIN
    SELECT book.name as book, book.image
    FROM (book INNER JOIN genre_book)
    INNER JOIN genre
    ON genre.id_genre=genre_book.id_genre
    AND genre_book.id_book=book.id_book
    WHERE genre.name = genre_name
    ORDER BY book.name;
END;
// DELIMITER ;
```

Результат:

call library_online.get_books_by_genre('drama')

book	image
Anna Karenina	images/anna-karenina.jpg
Atlas Shrugged	images/atlas-shrugged.jpg
Hamlet	images/hamlet.jpg
Murder on the Orient Express	images/murder-on-the-orient-express.jpeg
Picture of Dorian Gray	images/the-picture-of-dorian-gray.jpeg
The Alchemist	images/the-alchemist.jpeg
The Forest Song	images/forest-song.jpeg

Пошук книг за автором

DELIMITER //

CREATE PROCEDURE get_books_by_author (IN nickname char(30))

BEGIN

SELECT book.name as book, book.image

FROM (book INNER JOIN book_author)

INNER JOIN author

ON author.id_author=book_author.id_author

AND book_author.id_book_author=book.id_book

WHERE author.nickname = nickname

ORDER BY book.name;

END;

// DELIMITER ;

Результат:

call library_online.get_books_by_author('J. K. Rowling')

	book	image
▶	Harry Potter and the Deathly Hallows	images/harry-potter-and-the-deathly-hallows.jpg
	Harry Potter And The Philosopher`s Stone	images/harry-potter-and-the-philosophers-ston...
	Harry Potter And The Prisoner of Azkaban	images/harry-potter-and-the-prisoner-of-azkab...

Отримати детальну інформацію про жанр чи жанри певної книги

DELIMITER //

CREATE PROCEDURE get_genre (IN id int)

BEGIN

DECLARE error CHAR(30);

SET error = 'Invalid id entered';

IF (id >= 0) THEN

BEGIN

SELECT genre.name as genre, genre.about as about_genre

FROM (genre INNER JOIN genre_book)

INNER JOIN book

ON genre.id_genre=genre_book.id_genre

AND genre_book.id_book=book.id_book

WHERE book.id_book = id

ORDER BY genre.name;

END;

ELSE SELECT error;

END IF;

END // DELIMITER ;

Результат:

call library_online.get_genre(1)

	genre	about_genre
►	History	These books chronicle and layout a specific mo...
	Romance	Emotion-driven stories that are primarily focuse...

Отримати детальну інформацію про автора чи авторів певної книги

DELIMITER //

CREATE PROCEDURE get_author (IN id int)

BEGIN

DECLARE error CHAR(30);

SET error = 'Invalid id entered';

IF (id >= 0) THEN

BEGIN

SELECT author.first_name, author.last_name, author.born_date, author.death_date,
country.country

FROM (author INNER JOIN book_author)

INNER JOIN book

ON author.id_author=book_author.id_author

AND book_author.id_book_author=book.id_book

INNER JOIN country

ON author.country_id=country.id_country

WHERE book.id_book = id

ORDER BY author.first_name;

END;

ELSE SELECT error;

END IF;

END // DELIMITER ;

Результат:

call library_online.get_author(6)

	first_name	last_name	born_date	death_date	country
►	Ray Douglas	Bradbury	1920-08-22	2012-06-05	United States

Отримати детальну інформацію про видавництво певної книги

DELIMITER //

CREATE PROCEDURE get_publish_house (IN id int)

BEGIN

DECLARE error CHAR(30);

SET error = 'Invalid id entered';

IF (id >= 0) THEN

BEGIN

SELECT publish_house.name, publish_house.address, publish_house.contact_person

FROM publish_house

INNER JOIN book

ON book.id_publish_house=publish_house.id_publish_house

WHERE book.id_book = id;

END;

ELSE SELECT error;

END IF;

END // DELIMITER ;

Результат:

call library_online.get_publish_house(10)

	name	address	contact_person
►	Bloomsbury Publishing	31 Bedford Avenue, London, UK	contact@bloomsbury.com

Отримати детальну інформацію про книгу

DELIMITER //

CREATE PROCEDURE get_book_detail (IN id int)

BEGIN

DECLARE error CHAR(30);

SET error = 'Invalid id entered';

IF (id >= 0) THEN

BEGIN

SELECT book.name as book, book.page as number_of_pages, book.year,
book.description_of_book, book.image

FROM book

WHERE book.id_book = id;

END;

```
ELSE SELECT error;
```

```
END IF;
```

```
END // DELIMITER ;
```

Результат:

call library_online.get_book_detail(1)

	book	number_of_pages	year	description_of_book	image
▶	A Tale of Two Cities	537	1859	A Tale of Two Cities deals with the major theme...	images/a-tale-of-two-cities.jpg

Отримати файл книги

```
DELIMITER //
```

```
CREATE PROCEDURE get_file (IN id int)
```

```
BEGIN
```

```
DECLARE error CHAR(30);
```

```
SET error = 'Invalid id entered';
```

```
IF (id >= 0) THEN
```

```
BEGIN
```

```
SELECT book.file
```

```
FROM book
```

```
WHERE book.id_book = id;
```

```
END;
```

```
ELSE SELECT error;
```

```
END IF;
```

```
END // DELIMITER ;
```

Результат:

call library_online.get_file(4)

	file
▶	books/anna-karenina.pdf

Отримати впорядкований за датою створення список коментарів до певної КНИГИ

```
DELIMITER //
```

```
CREATE PROCEDURE get_comment (IN id int)
```

```
BEGIN
```

```
DECLARE error CHAR(30);
```

```

SET error = 'Invalid id entered';
IF (id >= 0) THEN
BEGIN
    SELECT reader.name, reader.surname, comment.comment, comment.posted
    FROM (comment INNER JOIN reader)
    INNER JOIN book
    ON comment.id_comment_book=book.id_book
    AND comment.id_comment_reader=reader.id_reader
    WHERE book.id_book = id
    ORDER BY comment.posted DESC;
END;
ELSE SELECT error;
END IF;
END // DELIMITER ;

```

Результат:

call library_online.get_comment(1)

	name	surname	comment	posted
►	Iryna	Dosiak	Very interesting book!	2020-05-23 13:02:01
	Christina	Dosiak	I like it	2020-04-23 11:02:01
	Oleksandra	Dypko	Amazing!!!	2020-02-02 15:02:01

Функція, яка рахує вік користувача

```

CREATE FUNCTION age(date_of_birth DATE)
RETURNS INT
DETERMINISTIC RETURN (YEAR(CURRENT_DATE) - YEAR(date_of_birth)) -
    (DATE_FORMAT(CURRENT_DATE, '%m%d') < DATE_FORMAT(date_of_birth, '%m%d'));

```

Результат:

	name	age
►	Iryna	19
	Christina	19
	Ivan	19
	Oleksandra	20

Отримати дані про певного користувача:

```

DELIMITER //
CREATE PROCEDURE get_reader_data (IN id int)
BEGIN

```

```

DECLARE error CHAR(30);
SET error = 'Invalid id entered';
IF (id >= 0) THEN
BEGIN
    SELECT reader.name, reader.surname, age(reader.date_of_birth) as age, reader.email
    FROM reader
    WHERE reader.id_reader = id;
END;
ELSE SELECT error;
END IF;
END // DELIMITER ;

```

Результат:

CALL library_online.get_reader_data(1)

	name	surname	age	email
►	Iryna	Dosiak	19	faaaagaaa@gmail.com

Отримати список книг, які додав певний користувач в свою бібліотеку

```

DELIMITER //
CREATE PROCEDURE get_reader_book(IN id int)
BEGIN
    DECLARE error CHAR(30);
    SET error = 'Invalid id entered';
    IF (id >= 0) THEN
    BEGIN
        SELECT book.name as book, book.image, reader_book.status, reader_book.rating
        FROM reader_book
        INNER JOIN book
        ON reader_book.id_reader_book=book.id_book
        WHERE reader_book.id_reader = id
        ORDER BY reader_book.upload_date;
    END;
    ELSE SELECT error;
    END IF;
END // DELIMITER ;

```

Результат:

CALL library_online.get_reader_book(1)

	book	image	status	rating
►	A Tale of Two Cities	images/a-tale-of-two-cities.jpg	read	5
	Alice`s Adventures in Wonderland	images/alices-adventures-in-wonderland.jpg	want_to_read	4

Тригер, який кодує пароль користувача:

*CREATE TRIGGER reader_password BEFORE INSERT ON reader
FOR EACH ROW SET NEW.password = md5(NEW.password);*

Результат:

	id_reader	name	surname	date_of_birth	email	password	is_admin
►	1	Iryna	Dosiak	2000-10-28	faaaagaaa@gmail.com	688aecc1a563ee29130fcff2145e918a	1
	2	Christina	Dosiak	2000-10-28	cristina@gmail.com	0310b9cc50347ee943a949dadb4eb042	0
	3	Ivan	Fedkiv	2001-04-06	ivan@gmail.com	688aecc1a563ee29130fcff2145e918a	0
	4	Oleksandra	Dypko	1999-08-02	oleksandra@gmail.com	b905aa02b2ae2b90ef0c5ce073422c64	0
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Реєстрація

*INSERT INTO reader(name, surname, date_of_birth, email, password)
VALUES ('Vitalik', 'Smith', '1996-10-03', 'vitalik@gmail.com', '123456789vitalik');*

Результат:

	id_reader	name	surname	date_of_birth	email	password	is_admin
►	1	Iryna	Dosiak	2000-10-28	faaaagaaa@gmail.com	688aecc1a563ee29130fcff2145e918a	1
	2	Christina	Dosiak	2000-10-28	cristina@gmail.com	0310b9cc50347ee943a949dadb4eb042	0
	3	Ivan	Fedkiv	2001-04-06	ivan@gmail.com	688aecc1a563ee29130fcff2145e918a	0
	4	Oleksandra	Dypko	1999-08-02	oleksandra@gmail.com	b905aa02b2ae2b90ef0c5ce073422c64	0
	5	Vitalik	Smith	1996-10-03	vitalik@gmail.com	f0f0ae5df5a6dfb5a6b34b793f30bf65	0

ВИСНОВКИ

Виконуючи розрахункову роботу я закріпила набуті навички роботи з MySQL.

Я обрала саме цю СУБД, оскільки MySQL — одна з найпоширеніших систем керування базами даних, вона проста у встановленні та використанні, гарантує високу швидкість виконання команд. Крім того, MySQL підтримується у Django, що було використано для виконання проекту.

У розрахунковій роботі я створила логічну та фізичну модель бази даних. Логічна схема містить 10 таблиць. Описала детально структурну модель бази даних, тобто кожну таблицю з вказанням призначення, описом атрибутів, ключів та індексів.

Навчилась створювати ділову модель, яка містить опис функціоналу проекту у вигляді таблиці. Подала ділову модель у вигляді запитів до БД мовою MySQL та показала результати виконання.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. <https://www.hostinger.ru/rukovodstva/shto-takoje-mysql/>
2. <https://metanit.com/sql/mysql/>
3. <https://www.mysql.com/products/workbench/>
4. <https://tproger.ru/translations/sql-vs-nosql/>
5. <https://docs.djangoproject.com/en/3.0/ref/databases/>
6. <https://www.8host.com/blog/zaprosy-v-mysql/>
7. <https://uk.wikipedia.org/wiki/MySQL>
8. https://www.w3schools.com/sql/sql_ref_mysql.asp
9. https://www.w3schools.com/python/python_mysql_create_table.asp
10. <https://www.webmasterwiki.ru/MySQL>
11. Методичні вказівки до лабораторних робіт