Міністерство освіти і науки України

Одеський національний політехнічний університет

Iнститут компьютерних систем

Кафедра компьютерних інтеллектуальних систем і мереж

Розрахунково-графічна робота

З дисципліни: “Організація баз данних”

Предметна область «Бiблiотека»

Зробив

Ст. Веденьєв А.О.

Групи АМ-181

Перевiрили:

Головачова О.В.

Одеса 2020

**Вступ**

Головні ідеї сучасних інформаційних технологій грунтуються на концепції, відповідно до якої будь-яка інформація повинна бути організована в базу даних (далі БД) для адекватного відтворення змінюється реальної обстановки в світі та задоволенні інформаційної потреби різних користувачів. Дані БД формуються і діють під управлінням спеціальних програмних комплексів, званих системами управління БД (СУБД).

Одним з основних напрямків у сфері автоматизації бізнес-процесів з вживанням інформаційних технологій є розробка БД, що дозволяє вирішити проблему зберігання і систематизації даних відповідно до індивідуальних вимог компанії.

Головною метою даної роботи є створення і розробка подібної бази даних з нуля. Дана робота містить 5 розділів, у яких описані усі кроки створеннябази даних.

Мета роботи: Формування структури БД бібліотеки ті її реалізація за допомогою СУБД MySql.  
**1. Опис предметної області**

Предметна область - ***бібліотека***.

***Бібліотека*** - це сховище книг.

***Завдання бібліотеки*** - видавати читачам книги і приймати їх.

У бібліотеці повинна зберігатись інформація про книги:

Кожна книга має свій ID, назву, рік видання і кількість сторінок. Також в бібліотеці повинна зберігатись інформація про виданні книги - це автор, назва видавництва, кількість примірників і жанр.

Оскільки книгу міг писати не 1 автор, то повинні бути вказані співавтори. Примірники бувають різні, отже вони мають свій номер і якість примірника.

У бібліотеку приходять читачі. Розрізняти ми їх будемо за номером читача в базі, номером телефону та ПІБ.

У кожного читача є доступ до свого облікового запису. У ньому вказується інформація про читача, а саме дата видачі книги, дата повернення книги, можливе продовження терміну на який видається книга, дата реєстрації аккаунта і ID аккаунта.

Книги можна забрати лише на певний час (з можливістю продовження цього часу).

Коли ***Читач*** приходить до ***Бібліотеки***, спочатку він повинен зареєструвати електроний ***Акаунт***, який складається з дати реєстрації, дати повернення книги і поля "Час продовження терміну", якщо комусь знадобиться взяти книжку на довше.

Від самого ***Читача*** ми будемо вимагати всього лише номер телефону і його ПІБ, більш ***Бібліотеці*** - не потрібно (в моєму розумінні структури Бібліотека).

Тобто тепер у нас існує ***Читач*** і його ***Аккаунт***, треба це якось пов'язати.

На допомогу приходить зв'язок ***Доступ***, яка складається з id ***Читача*** та id ***Акаунта*** (для зв'язку цих двох таблиць).

Все, з ***Читачем*** розібралися. Навіщо нам потрібен ***Аккаунт***?

На ***Аккаунт*** будуть видаватися ***Книги*** (живемо в 21 столітті, все пов'язане з електронікою). Тобто нам потрібна зв'язок для таблиць ***Книга*** і ***Аккаунт***.

Назвемо її ***Видача***.

У ***Видачі*** буде збережуться: id ***Видачі*** (тобто. Код самої операції, грубо кажучи її номер), id ***Читача***, id ***Акаунта***, а також id ***Примірника***.

Що таке ***Примірник***?

У кожної ***Книги*** може бути 1000 ***Примірників***. Тому прив'язувати зв'язок ***Видача*** будемо з ***Примірником***.

У таблиці ***Примірник*** будуть зберігатися: **Дата надходження, Стан Примірника**, а також його **Ціна**.

Як же ми отримуємо цей ***Примірник***?

***Примірник*** - це копія ***Книги***, кожна ***Книга*** має свого ***Автора***, ***Видавництво*** та ***Жанр***.

Оскільки, ***Авторів***, ***Видавництв***, ***Книг*** і ***Жанрів*** багато, ми створимо зв'язок між ними і назвемо її ***Видання***.

У ***Виданні*** будуть зберігається наступні дані - це id ***Книги***, id ***Жанру***, id ***Видавництва***, id ***Автора*** і id ***Примірника***.

Тепер пройдемося по кожній сутності, які об'єднує цей зв'язок:

Про ***Примірник*** ми сказали вище, його пропустимо.

Почнемо з ***Книги***.

Кожна ***Книга*** в нашій базі даних буде мати **Назва**, **Рік видання** і **К-ть** сторінок. Це основні атрибути сутності ***Книга***, тому що вони єдині для кожної ***Книги***.

Далі йде ***Автор***.

Як виявилося у ***Книги*** може бути **Автор** і **Співавтор**. Тому сутність ***Автор*** матиме: **ПІБ Автора і ПІБ Співавтора** (якщо такий існує).

Далі у нас йде ***Жанр***, тут все досить просто, у нас буде всього 1 текстове поле з назвою ***Жанру***. Аналогічна ситуація і у ***Видавництва***, в цій суті буде зберігається тільки назва (ну і, природно її id).

**2. Реляційна модель + словесний опис логічного рівня**

У базі даних можна виділити наступні основні сутності: ***Книга, Примірник, Читач, Автор, Жанр, Видавництво, Аккаунт.***

Ось їх атрибути:

Книга:

⦁ Код книги (Primary key)

⦁ Найменування

⦁ Кількість стор.

⦁ Рік видання

⦁ Код автора (Foreign key)

⦁ Код жанру (Foreign key)

⦁ Код видавництва (Foreign key)

Примірник:

⦁ Код екземпляра (Primary key)

⦁ Ціна

⦁ Код книги (Foreign key)

⦁ Стан книги

⦁ Дата надходження

Читач:

⦁ Код читача (Primary key)

⦁ ПІБ

⦁ Телефон

Автор:

⦁ Код автора (Primary key)

⦁ Автор

⦁ Співавтори

Жанр:

⦁ Код жанру (Primary key)

⦁ Жанр

Видавництво:

⦁ Код видавництва (Primary key)

⦁ Назва

Аккаунт:

⦁ Код аккаунта (Primary key)

⦁ Код читача (Foreign key)

⦁ Код екземпляра (Foreign key)

⦁ Дата реєстрації

⦁ Дата видачі

⦁ Дата повернення

⦁ Час продовження терміну

Primary key і Foreign key були додані після створення логічного рівня БД для зручності відмінності і більш точної інформації.

**2.1 Концептуальний рівень**

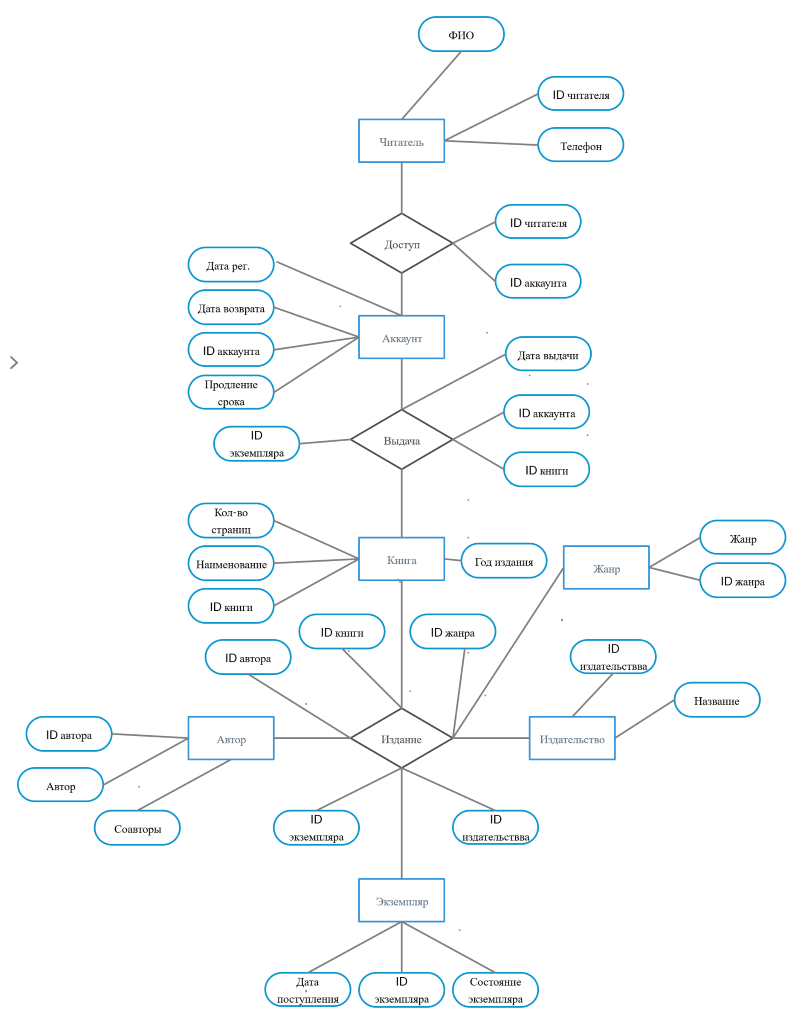


Схема концептуально рівня

**2.2 Логічний рівень**

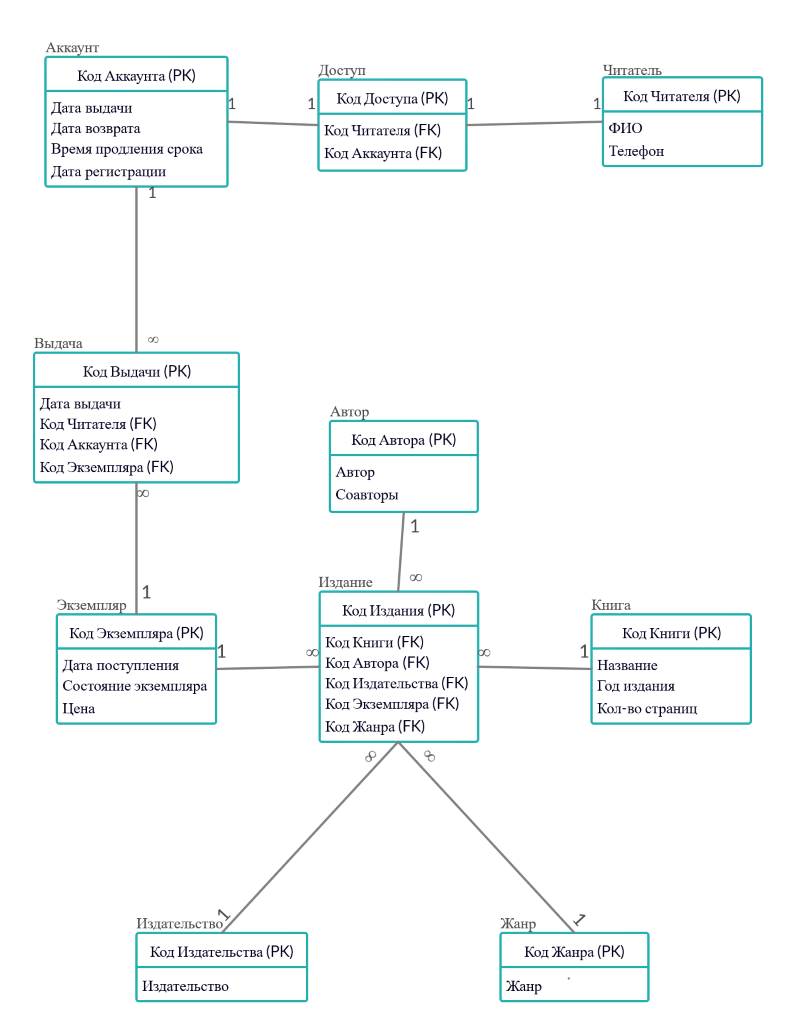


Схема логічного рівня

Атрибути сутності ***Акаунт***: його ID,дата видачі, дата возврату, час подовження сроку,дата реєстрації.

Атрибути зв'язку ***Доступ***: йогоID, ID Акаунту, код читача.

Атрибути сутності ***Читач***: його ID, ПІБ, телефон.

Атрибути зв'язку ***Видача***: його ID,дата видачі, код читача, код акакунта, код примірника.

Атрибути сутності ***Автор***: його ID, ім’я, соавтори.

Атрибути сутності ***Примiрник***: його ID,дата надходження, стан, вартість.

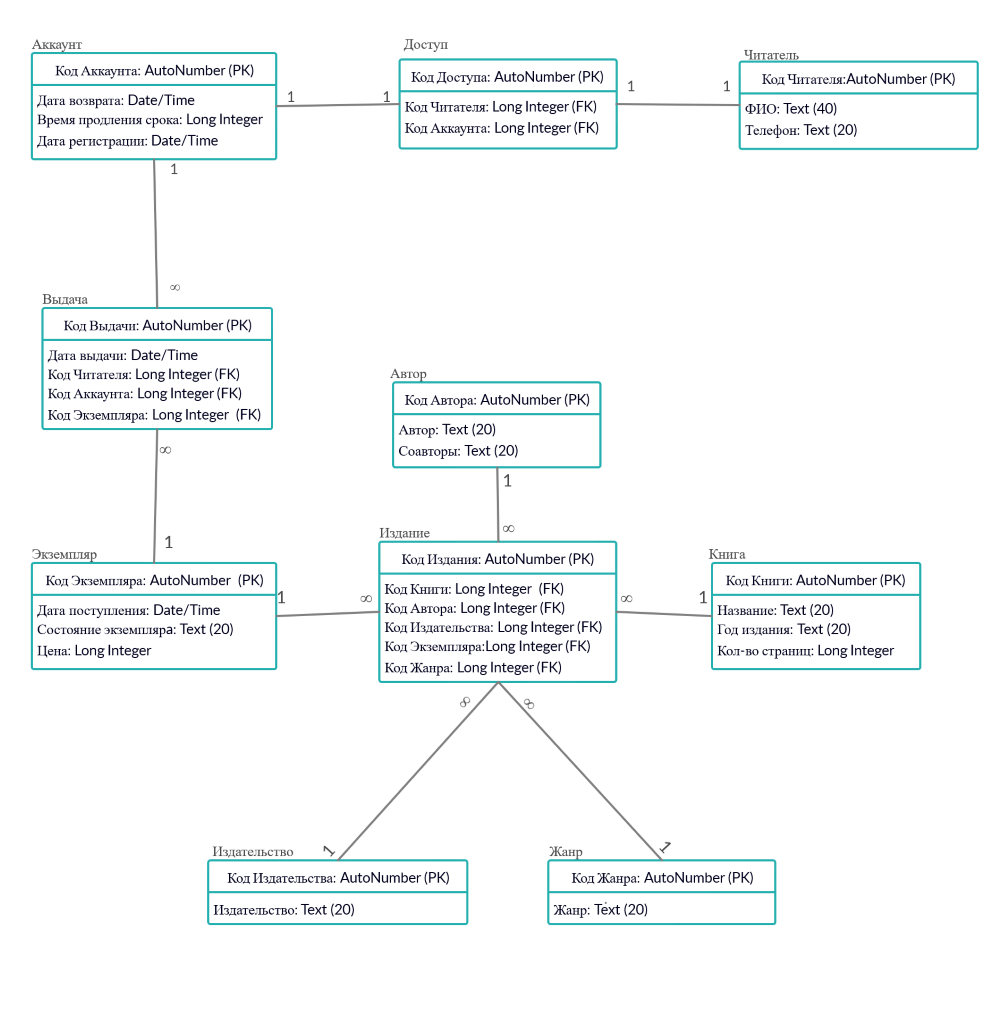
Атрибути сутності ***Видавництво***: її ID, назва.

Атрибути зв'язку ***Видання***: його ID, код книги, код автору, код видавництва, примiрнику та жанру.

Атрибути сутності ***Книга***: його ID, назва, год видання, к-ть сторінок.

Атрибути сутності ***Жанр***: його ID, назва.

**2.3. Фізичний рівень**



Фізичний рівень

Для усіх текстових полів у таблицях використовується тип даних TEXT з довжиною 20 символів, окрім поля ПІБ, де зберігається повне ім’я читача. Воно має розмірність 40. Тобто:

Таблиця ***ЧИТАЧ****:*

ПІБ – Text(40);

Телефон – Text(20);

Таблиця ***АВТОР***:

Автор – Text(20);

Співавтор – Text(20);

Таблиця ***ЖАНР:***

Жанр – Text(20);

Таблиця ***ВИДАВНИЦТВО:***

Видання – Text(20);

Для дат використовується DATE/TIME.

Усi числові значення, первинні і зовнішні ключі зберігаються у LONG INT.

Таблиця ***КНИГА:***

Назва – Text(20);

Рік видання – Text(20);

К-ть сторінок – Long Integer;

Таблиця ***АККАУНТ:***

Дата повернення – Date/Time;

Час продовження терміну – Long Integer;

Дата реєстрації – Date/Time;

Таблиця ***ВИДАННЯ:***

Id Книги – Long Integer;

Id Автора – Long Integer;

Id Видавництва – Long Integer;

Id Примірника – Long Integer;

Id Жанра – Long Integer;

Таблиця ***ПРИМІРНИК:***

Дата надходження - Date/Time;

Стан – Text(20);

Ціна – Long Integer;

Таблиця ***ДОСТУП:***

Id Читача – Long Integer;

Id Аккаунта – Long Integer;

Таблиця ***ВИДАЧА:***

Дата видачі – Date/Time;

Id Читача – Long Integer;

Id Аккаунта – Long Integer;

Id Примірника – Long Integer;

**Приведення до 3 НФ**

**Автор Книга**

Братья Стругацкие «Трудно быть Богом», «Улитка на склоне»

Сергей Лукьяненко «Ночной дозор»

Спочатку приведемо до 1 НФ:

**Автор Книга**

Братья Стругацкие «Трудно быть Богом»

Братья Стругацкие «Улитка на склоне»

Сергей Лукьяненко «Ночной дозор»

**2 НФ**:

**Автор Книга Жанр Ціна Кол-во стр.**

Братья Стругацкие «Трудно быть Богом» Фантастика 200 350

Братья Стругацкие «Улитка на склоне» Фил. Роман 200 390

Сергей Лукьяненко «Ночной дозор» Роман 175 200

**Автор Книга Жанр Кол-во стр.**

Братья Стругацкие «Трудно быть Богом» Фантастика 350

Братья Стругацкие «Улитка на склоне» Фил. Роман 390

Сергей Лукьяненко «Ночной дозор» Роман 200

**Автор Ціна Издательство**

Братья Стругацкие 200 «Альфа - Книга»

Сергей Лукьяненко 175 «РОСМЭН»

**3 НФ:**

**Автор Цена Стан екз.**

Братья Стругацкие 200 чудове

Братья Стругацкие 150 добре

Сергей Лукьяненко 175 чудове

Сергей Лукьяненко 100 погане

**Автор Стан екз.**

Братья Стругацкие чудове

Братья Стругацкие добре

Братья Стругацкие чудове

Братья Стругацкие погане

**Стан екз. Ціна**

Чудове -0

Добре -50

Погане -75

**Створення таблиць для бази даних та їх заповнення**

Створені таблиці та занесенні у них данні прикріплені у Додатку А.

Для початку потрібно створити базу данних та обрати її для роботи:

create database library;

use library;

Далі у обраній бд потрібно створити усі визначені таблиці.

CREATE TABLE accesses (

id int(11) NOT NULL,

reader\_id int(11) NOT NULL,

account\_id int(11) NOT NULL

);

CREATE TABLE accounts (

id int(11) NOT NULL,

account\_id int(11) NOT NULL,

return\_date datetime NOT NULL,

registration\_date datetime NOT NULL,

delivery\_period smallint(6) NOT NULL

);

CREATE TABLE authors (

id int(11) NOT NULL,

author varchar(120) NOT NULL,

collaborators varchar(500) DEFAULT NULL

) ;

CREATE TABLE books (

id int(11) NOT NULL,

name varchar(120) NOT NULL,

publish\_date date NOT NULL,

num\_page mediumint(9) NOT NULL

);

CREATE TABLE deliveries (

id int(11) NOT NULL,

account\_id int(11) NOT NULL,

instance\_id int(11) NOT NULL,

book\_id int(11) NOT NULL,

delivery\_date datetime NOT NULL

);

CREATE TABLE editions (

id int(11) NOT NULL,

genre\_id int(11) NOT NULL,

book\_id int(11) NOT NULL,

author\_id int(11) NOT NULL,

instance\_id int(11) NOT NULL,

publisher\_id int(11) NOT NULL,

isbn bigint(20) DEFAULT NULL

);

CREATE TABLE genres (

id int(11) NOT NULL,

genre varchar(120) NOT NULL

) ;

CREATE TABLE instances (

id int(11) NOT NULL,

receipt\_date datetime NOT NULL,

instance\_state enum('DAMAGED','VERY\_OLD','OLD','NORMAL','NEW') NOT NULL

);

CREATE TABLE publishers (

id int(11) NOT NULL,

name varchar(120) NOT NULL

);

CREATE TABLE readers (

id int(11) NOT NULL,

full\_name varchar(120) NOT NULL,

phone varchar(20) NOT NULL

);

**Введення даних**

INSERT INTO accesses ( id , reader\_id , account\_id ) VALUES

(1, 1, 1),

(2, 2, 2),

(3, 3, 3),

(4, 4, 4),

(5, 5, 5),

(6, 6, 6),

(7, 7, 7),

(8, 8, 8),

(9, 9, 9),

(10, 10, 10);

INSERT INTO accounts ( id , return\_date , registration\_date , delivery\_period ) VALUES

(1, '2020-05-30 14:29:38', '2020-05-24 14:29:38', 7),

(2, '2020-05-30 14:29:38', '2020-05-12 14:29:38', 6),

(3, '2020-05-30 14:29:38', '2020-05-12 14:29:38', 8),

(4, '2020-05-30 14:29:38', '2020-05-12 14:29:38', 5),

(5, '2020-05-30 14:29:38', '2020-05-12 14:29:38', 7),

(6, '2020-05-30 14:29:38', '2020-05-12 14:29:38', 14),

(7, '2020-05-30 14:29:38', '2020-05-12 14:29:38', 7),

(8, '2020-05-30 14:29:38', '2020-05-12 14:29:38', 5),

(9, '2020-05-30 14:29:38', '2020-05-12 14:29:38', 4),

(10, '2020-05-30 14:29:38', '2020-05-12 14:29:38', 2);

INSERT INTO authors ( id , author , collaborators ) VALUES

(1, 'William Shakespeare', NULL),

(2, 'Agatha Christie', NULL),

(3, 'Barbara Cartland', NULL),

(4, 'Danielle Steel', NULL),

(5, 'Harold Robbins', NULL),

(6, 'Georges Simenon', NULL),

(7, 'Enid Blyton', NULL),

(8, 'Sidney Sheldon', NULL),

(9, 'J. K. Rowling', NULL),

(10, 'Gilbert Patten', NULL),

(11, 'George R.R. Martin', NULL);

INSERT INTO books ( id , name , publish\_date , num\_page ) VALUES

(1, 'Harry Potter and the Philosopher\'s Stone', '1997-06-26', 223),

(2, 'Harry Potter and the Chamber of Secrets', '1998-07-02', 251),

(3, 'Harry Potter and the Prisoner of Azkaban', '1999-07-08', 317),

(4, 'Harry Potter and the Goblet of Fire', '2000-07-08', 636),

(5, 'Harry Potter and the Order of the Phoenix', '2003-06-21', 766),

(6, 'Harry Potter and the Half-Blood Prince', '2005-07-16', 607),

(7, 'Harry Potter and the Deathly Hallows', '2007-07-21', 607),

(8, 'Fantastic Beasts and Where to Find Them', '2001-12-01', 128),

(9, 'A Game of Thrones ', '1996-08-06', 835),

(10, 'A Clash of Kings', '1998-11-16', 969);

INSERT INTO deliveries ( id , account\_id , instance\_id , book\_id , delivery\_date ) VALUES

(1, 1, 1, 1, '2020-06-24 14:32:09'),

(2, 2, 2, 2, '2020-06-11 14:32:26'),

(3, 3, 3, 3, '2020-06-11 14:32:26'),

(4, 4, 4, 4, '2020-06-11 14:32:26'),

(5, 5, 5, 5, '2020-06-11 14:32:26'),

(6, 6, 6, 6, '2020-06-11 14:32:26'),

(7, 7, 7, 7, '2020-06-11 14:32:26'),

(8, 8, 8, 8, '2020-06-11 14:32:26'),

(9, 9, 9, 9, '2020-06-11 14:32:26'),

(10, 10, 10, 10, '2020-06-11 14:32:26');

INSERT INTO editions ( id , genre\_id , book\_id , author\_id , instance\_id , publisher\_id , isbn ) VALUES

(1, 1, 1, 9, 1, 1, 7800941520756),

(2, 1, 2, 9, 2, 1, 8390932357369),

(3, 1, 3, 9, 3, 1, 7667563561254),

(4, 1, 4, 9, 4, 1, 3931308812010),

(5, 1, 5, 9, 5, 1, 4842982085734),

(6, 1, 6, 9, 6, 1, 8389141085589),

(7, 1, 7, 7, 7, 1, 7676617556804),

(8, 1, 8, 9, 8, 1, 6586547030720),

(9, 1, 9, 11, 9, 3, 9780553588484),

(10, 1, 10, 11, 10, 3, 9780553381696);

INSERT INTO genres ( id , genre ) VALUES

(1, 'Fantasy Fiction'),

(2, 'Action and adventure'),

(3, 'Crime'),

(4, 'Drama'),

(5, 'Fantasy'),

(6, 'History'),

(7, 'Horror'),

(8, 'Mystery'),

(9, 'Science fiction'),

(10, 'Thriller');

INSERT INTO instances ( id , receipt\_date , instance\_state ) VALUES

(1, '2020-05-30 11:52:09', 'NEW'),

(2, '2020-05-30 11:52:09', 'NEW'),

(3, '2020-05-30 11:52:09', 'NEW'),

(4, '2020-05-30 11:52:09', 'NEW'),

(5, '2020-05-30 11:52:09', 'NEW'),

(6, '2020-05-30 11:52:09', 'NEW'),

(7, '2020-05-30 11:52:09', 'NEW'),

(8, '2020-05-30 11:52:09', 'NEW'),

(9, '2020-05-30 11:52:09', 'NEW'),

(10, '2020-05-30 11:52:09', 'NEW');

INSERT INTO publishers ( id , name ) VALUES

(1, 'Bloomsbury Publishing'),

(2, 'Scholastic Press'),

(3, 'Bantam'),

(4, 'Hachette Livre'),

(5, 'HarperCollins'),

(6, 'Macmillan Publishers'),

(7, 'Simon & Schuster'),

(8, 'McGraw-Hill Education'),

(9, 'Houghton Mifflin Harcourt'),

(10, 'Pearson Education');

INSERT INTO readers ( id , full\_name , phone ) VALUES

(1, 'Mr. Machine', '5554443322'),

(2, 'Vladimir Zelenskiy', '5554443322'),

(3, 'Mr. Shobik', '5554443322'),

(4, 'Steve From Minecraft', '5554443322'),

(5, 'Hermione Granger', '5554443322'),

(6, 'Ginevra Weasley', '5554443322'),

(7, 'User\_1', '5554443322'),

(8, 'User\_2', '5554443322'),

(9, 'User\_3', '5554443322'),

(10, 'User\_4', '5554443322');

**Отримання інформації**

Виконання запитів до бази даних наведені у Додатку Б.

Код:

1. SELECT B.name, MAX(E.isbn) FROM editions E INNER JOIN books B ON B.id = E.book\_id;
2. SELECT B.name, MIN(E.isbn) FROM editions E INNER JOIN books B ON B.id = E.book\_id;
3. SELECT B.name AS bookname, A.author AS author, E.isbn AS isbn FROM editions E, authors A, books B WHERE A.id = E.author\_id AND B.id = E.book\_id ;
4. SELECT B.name AS bookname, A.author AS author, E.isbn AS isbn FROM editions E LEFT JOIN authors A ON A.id = E.author\_id RIGHT JOIN books B ON B.id = E.book\_id;
5. SELECT SUM(delivery\_period) FROM accounts WHERE 1;
6. SELECT AVG(delivery\_period) FROM accounts WHERE 1;
7. SELECT author FROM authors WHERE author LIKE '%Martin';
8. SELECT author FROM authors WHERE author LIKE '%Martin';
9. SELECT \* FROM accounts WHERE delivery\_period BETWEEN 3 AND 5;
10. SELECT \* FROM authors WHERE author REGEXP("[GJ]");
11. SELECT \* FROM books ORDER BY publish\_date DESC;
12. SELECT COUNT(E.id), A.author FROM editions E LEFT JOIN authors A ON A.id = E.author\_id GROUP BY E.author\_id;
13. SELECT \* FROM books WHERE name LIKE '%Harry%' ;
14. SELECT MIN(delivery\_period) FROM accounts ;
15. SELECT MAX(delivery\_period) FROM accounts;
16. SELECT \* FROM authors where id = (SELECT COUNT(id) FROM books);
17. SELECT \* FROM deliveries GROUP BY book\_id HAVING delivery\_date > 2020 AND instance\_id > 8;
18. SELECT B.name, D.deliveries FROM ( SELECT book\_id, COUNT(id) AS deliveries FROM deliveries GROUP BY book\_id ) AS D LEFT JOIN books B ON B.id = D.book\_id;
19. SELECT B.name, (SELECT COUNT(D.id) FROM deliveries D WHERE D.book\_id = B.id) as delivery\_count FROM books B;
20. UPDATE deliveries SET delivery\_date = '1998-12-11 10:00:00' WHERE id = ROUND(RAND() \* ( SELECT MAX(id) FROM books));
21. SELECT name FROM books WHERE id = ANY (SELECT book\_id FROM deliveries WHERE delivery\_date < '2000-01-01 00:00:00');
22. SELECT name FROM books WHERE id = SOME (SELECT book\_id FROM deliveries WHERE delivery\_date < '2000-01-01 00:00:00");
23. SELECT name FROM books WHERE id != ALL (SELECT DISTINCT book\_id FROM deliveries);
24. DELETE FROM books WHERE id != ALL (SELECT DISTINCT book\_id FROM deliveries);
25. INSERT INTO deliveries(account\_id,instance\_id,book\_id,delivery\_date) VALUES( ROUND(RAND() \* (SELECT MAX(id) FROM accounts)), 10, 10, '2020-01-01 10:00:00' );

**Висновки**

Організація даних є ключовим моментом при роботі з великими обсягами інформації. Надзвичайно важливо впорядкувати інформацію таким чином, щоб легко і швидко знаходити потрібні відомості.

Перед створенням бази даних необхідно відповісти на наступні питання:

Яке призначення бази даних і хто буде нею користуватися?

Які таблиці (дані) буде містити база даних?

Які запити і звіти можуть знадобитися користувачам цієї бази даних?

Які форми може знадобитися створити?

Відповідаючи на ці питання, можна розробити проект бази даних і створити корисну і зручну у використанні базу даних.

**Список використаних джерел**

6. 1) <https://el.opu.ua/course/view.php?id=611>

(Усi лекції)

2) <https://dev.mysql.com/doc/>

(Документацiя MySQL, в основному для розгляду прикладiв рiзних запитiв)

3) <https://www.w3schools.com/sql/sql_union.asp>

4) <https://www.sqlshack.com/sql-overview-usage-and-examples/>

5) <https://www.guru99.com/limit.html>

6) <https://study.com/academy/lesson/regular-expressions-in-sql-server-databases-implementation-use.html>

7) <https://www.geeksforgeeks.org/mysql-regular-expressions-regexp/>

8) <https://dev.mysql.com/doc/refman/8.0/en/sql-data-definition-statements.html>

9) <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

10) <https://dev.mysql.com/doc/refman/8.0/en/non-typed-operators.html>

11) https://dev.mysql.com/doc/refman/8.0/en/group-by-functions-and-modifiers.html