НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

з лабораторної роботи №*1*

з дисципліни

«Вступ до баз даних та інформаційних систем»

на тему: «Проектування реляційної бази даних

та формування запитів на основі операцій

реляційної алгебри Кодда»

|  |  |
| --- | --- |
| Виконала: | Керівник: |
| студентка групи КМ-01 | *ст. викладач Бай Ю. П.* |
| *Боженко А. О.* |  |

Київ — 2022

Зміст

[Мета 3](#_Toc115901797)

[Постановка завдань 4](#_Toc115901798)

[Основна частина 5](#_Toc115901799)

[1.1 Концептуальне проектування схеми 5](#_Toc115901800)

[1.2 Логічне проектування схеми 6](#_Toc115901801)

[1.3 Фізичне проектування схеми 7](#_Toc115901802)

[2.1 Генерація бази даних 8](#_Toc115901803)

[2.2 Виконання запитів 8](#_Toc115901804)

[Висновки 10](#_Toc115901805)

[Скрипти 11](#_Toc115901806)

[Скрипт 1.1 – створення схеми «Людина є автор підручника» 11](#_Toc115901807)

[Скрипт 1.2 – запити до бази даних «Магазин іграшок» 12](#_Toc115901808)

[Використані джерела 13](#_Toc115901809)

# Мета

Створити базу даних, спроектувати дану схему на концептуальному, логічному та фізичному (використовуючи діалект PostgreSQL) рівнях. Використати операції реляційної алгебри Кодда та агрегатні функції мови.

# Постановка завдань

**ВАРІАНТ №4**

Завдання 1. Спроектувати базу даних, що дозволить відобразити наступні події:

*Людина є автором підручника.*

1a) Визначити сутності та їх атрибути, встановити зв’язки між сутностями. Побудувати ER-діаграму.

1b) Побудувати логічну схему таблиць, використовуючи «crow's foot notation».

1с) За допомогою команд мови SQL створити таблиці в СУБД PostgreSQL. Визначити поля та типи. Первинні та зовнішні ключі створювати окремо від таблиць, використовуючи команду ALTER TABLE.

Завдання 2. Згенерувати базу даних з книги Б. Форта та виконати запити (потрібні для виконання завдань файли *create.txt, populate.txt* можна завантажити, наприклад, з <https://github.com/alinbxSorcerer/SQL-in-10-minutes-with-notes.git>).

2a) Яка назва проданого найдорожчого товару?

2b) Як звуть покупця з найдовшим іменем – поле назвати long\_name?

2c) Вивести PROD\_ID товарів та імена постачальників для тих товарів, які були продані. Результат вивести у верхньому регістрі, як єдине поле products\_sold.

Завдання 3. Виконати запити 2a), 2b), 2с) з попереднього завдання, використовуючи операції реляційної алгебри Кодда та агрегатні функції мови SQL.

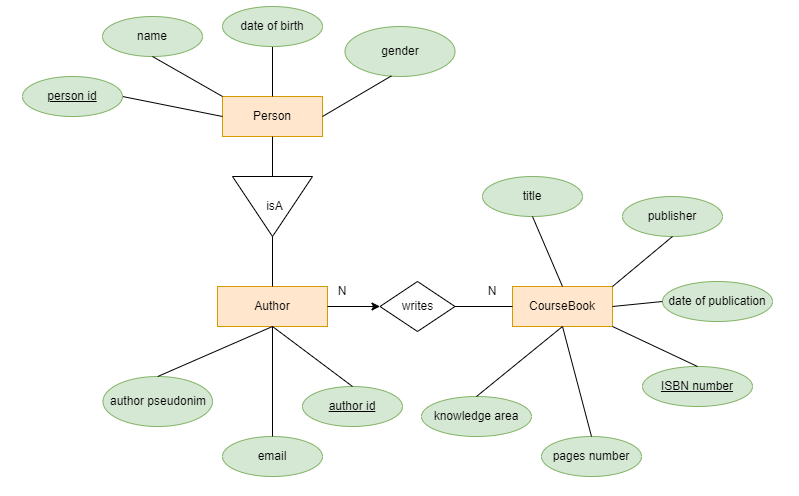
# Основна частина

* 1. **Концептуальне проектування схеми**

Подія: Людина є автором підручника.

Виділимо 3 сутності: людина, автор та підручник. Людина(Person) має атрибути: ім’я(name), ідентифікаційний номер документу (person\_id), дату народження(date of birth) та стать(gender). Автор(Author) має такі ж властивості, що й людина, тобто Автор є людина, окрім цього Автор має ідентифікаційний номер(author id), можливо псевдонім(author name) та адресу пошти(email). Підручник(Coursebook) має свій ISBN код(ISBN number), назву(title), видавця(publisher), дату видання(date of publication), кількість сторінок(pages number), область знань(knowledge area). Можна встановити зв’язок типу «Є»(isA) між людиною та автором, тобто автор є людиною але не кожна людина є автором(ієрахія). Зв’зок між автором та підручником –можна визначити як багато-до-багатьох, оскільки автор може написати багато підручників і один підручник може бути написаний багатьма авторами.

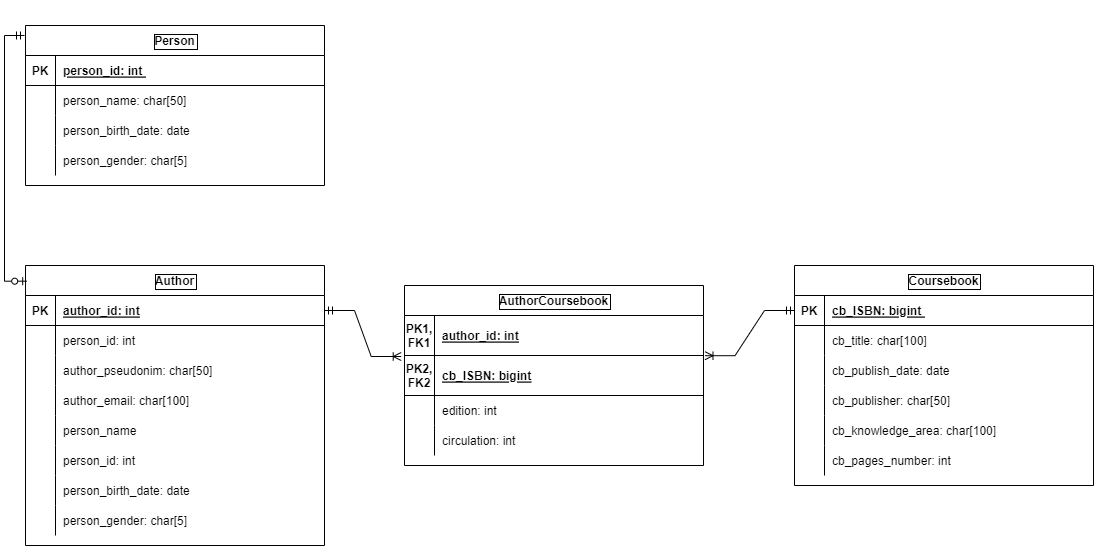
На малюнку 1.1 зображено концептуальне проектування схеми «Людина є автором підручника».

Мал 1.1 – концептуальна схема «Людина є автором підручника»

* 1. **Логічне проектування схеми**

Розглянемо сутності Person та Author. Нехай головним ключем сутності Person – це passport id, а головним ключем сутності Author – author id. Сутність Person є батьківською по відношенню до сутності Author, тому вважатимемо всі атребути Person успадкованими Author. Сутність Coursebook має головний ключ - cb\_ISBN. Розглянемо зв’язок між сутностями Author та Coursebook. Оскільки тип зв’язку мж ними багато-до-багатьох, то необхідно додатково створити сутність-зв’язку Author\_Coursebook. Сутність Author\_Coursebook має складений головний ключ - cb\_ISBN, author\_id. Атрибути Author\_Coursebook: cb\_ISBN (ключ-зв’язка із сутністю Coursebook) , author\_id (ключ-зв’язка із сутністю Author) та особисті атрибути: edition(видання) та circulation(наклад).

На малюнку 1.2 зображено логічне проектування схеми «Людина є автором підручника».



Мал 1.2 – логічна схема «Людина є автором підручника»

* 1. **Фізичне проектування схеми**

Визначимо типи атрибутів для таблиці(сутності) Person. Char(n): person\_name, person\_gender. Int: person\_id. Date: person\_birth\_date. Всі атрибути мають бути заповнені (NOT NULL). Область можливих значень атрибуту person\_gender: ‘woman’, ‘man’.

Всі атребути Person успадковуються таблицею Author. Для цього застосуємо ключове слово inherits, що дозволяє Postresql діалекту створювати ієрархічні зв’язки. Визначимо додаткові атребути. Int: author\_id. Char(n): author\_pseudonim, author\_email. Атрибути NOT NULL: author\_id.

Визначимо типи атрибутів для таблиці Coursebook. Bigint: cb\_ISBN. Int: cb\_pages\_number. Char(n): cb\_title, cb\_publisher, cb\_knowledge\_area. Date: cb\_publish\_date. Всі атрибути мають бути NOT NULL.

Визначимо типи атрибутів для таблиці Author\_coursebook. Bigint: cb\_ISBN, Int: author\_id, edition, circulation. Атрибути author\_id та cb\_ISBN мають бути NOT NULL.

Визначаючи ключі-зв’язки(FOREIGN KEY), визначимо дії на оновлення та видалення батьківських таблиць (ON UPDATE CASCADE, ON DELETE CASCADE).

Postrgesql-скрипт фізичної схеми знаходится в розділі Скрипти, Скрипт 1.1.

**2.1 Генерація бази даних**

Створено базу даних з книги Б. Форт та додано дані.

**2.2 Виконання запитів**

2a) Яка назва проданого найдорожчого товару?

select

distinct prod\_name

from products

inner join orderitems using(prod\_id)

where item\_price = (select

max(item\_price)

from orderitems)

2b) Як звуть покупця з найдовшим іменем – поле назвати long\_name?

select

cust\_name as long\_name

from customers

where length(cust\_name) = (select

max(length(cust\_name))

from customers)

2c) Вивести PROD\_ID товарів та імена постачальників для тих товарів, які були продані. Результат вивести у верхньому регістрі, як єдине поле products\_sold

select distinct

upper(products.prod\_id || ' - ' || vend\_name) as products\_sold

from orderitems inner join products using(prod\_id)

inner join vendors using(vend\_id)

# Висновки

Було створено базу даних зі схемою «Людина є автором підручника», визначено сутності: Людина, Автор, Підручник. Визначено атребути сутностей. Визначено зв’язки та їх типи між сутностями: Автор «Є» Людина (ієрархія), Людина «Пише» Книгу (багато-до-багатьох). Реалізовано схему на фізичному рівні за допомогою діалекту PostgreSQL.

Згенеровано іншу базу даних «Магазин іграшкових товарів» з даних скриптів, заповнено даними та виконано запити за допомогою базових операцій реляційної алгебри Кодда, в основному використовувалися такі операції як: проекція, предикат, перейменування та перетин.

# Скрипти

**Скрипт 1.1 – створення схеми «Людина є автор підручника»**

create table person (

person\_id int not null,

person\_name char(50) not null,

person\_gender char(5) check (person\_gender = 'woman' or person\_gender = 'man') not null,

person\_birth\_date date not null

);

create table author (

author\_id int not null,

author\_pseudonim char(50),

author\_email char(100)

) inherits (person);

create table coursebook (

cb\_ISBN bigint not null,

cb\_title char(100) not null,

cb\_publish\_date date not null,

cb\_publisher char(50) not null,

cb\_knowledge\_area char(100),

cb\_pages\_number int not null

);

create table author\_coursebook (

cb\_ISBN bigint unique not null,

author\_id int not null,

edition int,

circulation int

);

-- defining primary keys

alter table person

add constraint pk\_person primary key (person\_id);

alter table author

add constraint pk\_author primary key (author\_id);

alter table coursebook

add constraint pk\_coursebook primary key (cb\_ISBN);

-- defining foreign keys

alter table author\_coursebook

add constraint fk\_author

foreign key (author\_id)

references author (author\_id) on update cascade on delete cascade;

alter table author\_coursebook

add constraint fk\_coursebook

foreign key (cb\_ISBN)

references coursebook (cb\_ISBN) on update cascade on delete cascade;

**Скрипт 1.2 – запити до бази даних «Магазин іграшок»**

-- Яка назва проданого найдорожчого товару?

-- select

-- distinct prod\_name

-- from products

-- inner join orderitems using(prod\_id)

-- where item\_price = (select

-- max(item\_price)

-- from orderitems)

-- Як звуть покупця з найдовшим іменем – поле назвати long\_name

-- select

-- cust\_name as long\_name

-- from customers

-- where length(cust\_name) = (select

-- max(length(cust\_name))

-- from customers)

-- Вивести PROD\_ID товарів та імена постачальників для тих товарів,

-- які були продані. Результат вивести у верхньому регістрі, як єдине

-- поле products\_sold

-- select distinct

-- upper(products.prod\_id || ' - ' || vend\_name) as products\_sold

-- from orderitems inner join products using(prod\_id)

-- inner join vendors using(vend\_id)

# Використані джерела

1. «Database Systems. A Practical Approach to Design, Implementation, and Management», 6th Edition Global Edition, Thomas M. Connolly, Carolyn E. Begg, University of the west of Scotland, p. 167-185, p. 405-440.
2. <https://www.sqltutorial.org>
3. [https://www.postgresqltutorial.com/postgresql-tutorial](https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-create-table/)
4. <https://arctype.com/blog/inheritance-in-postgres/>