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:

Створення таблиць. Виконання простих запитів на мові SQL та за допомогою алгебри Кодда. Використання вбудованих функцій мови SQL

Виконала:

студентка групи КМ-03
Пюстонен С.Р.

Керівник:

ст. викладач Бай Ю. П.

Київ-2022

ЗМІСТ

ПЕРЕЛІК ЗАВДАНЬ

ВАРІАНТ №23

Завдання 1. Спроектувати базу даних, що дозволить відобразити наступні події (*5 балів*):

Людина має лікарняну картку, що містить записи про історію хвороби.

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

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

1c) 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) (*6 балів*):

2a) Яка назва проданого найдешевшого товару?

2b) Який PROD_ID товару з найдовшою назвою?

2c) Вивести PROD_ID товарів та імена постачальників для тих товарів, що були продані. Результат вивести у верхньому регістрі, як єдине поле products_sold.

Завдання 3. Виконати запити 2a), 2b) з попереднього завдання, використовуючи операції реляційної алгебри Кодда та агрегатні функції мови SQL (*4 бали*).

ЗАВДАННЯ 1

1a) Визначаємо сутності та їх атрибути, встановити зв'язки між сутностями. Будуємо діаграму.

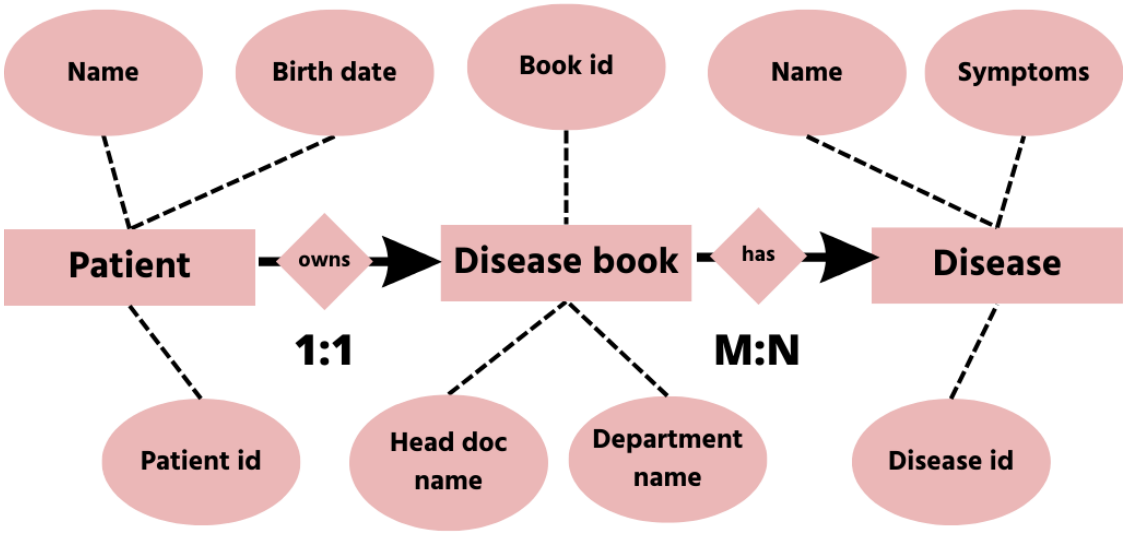


Рис.1

Для зв'язку **1:1** у відношенні створюється зовнішній ключ. Зовнішній ключ приймає значення тільки з множини значень первинного ключа відношення, що може існувати самостійно.

Для зв'язку **M:N** відношення розбивається на 2 окремі відношення зі зв'язками **1:M** та **1:N** за допомогою так званої з'єднуючої таблиці, яка містить значення первинних ключів.

1b) Будуємо логічну схему таблиць.

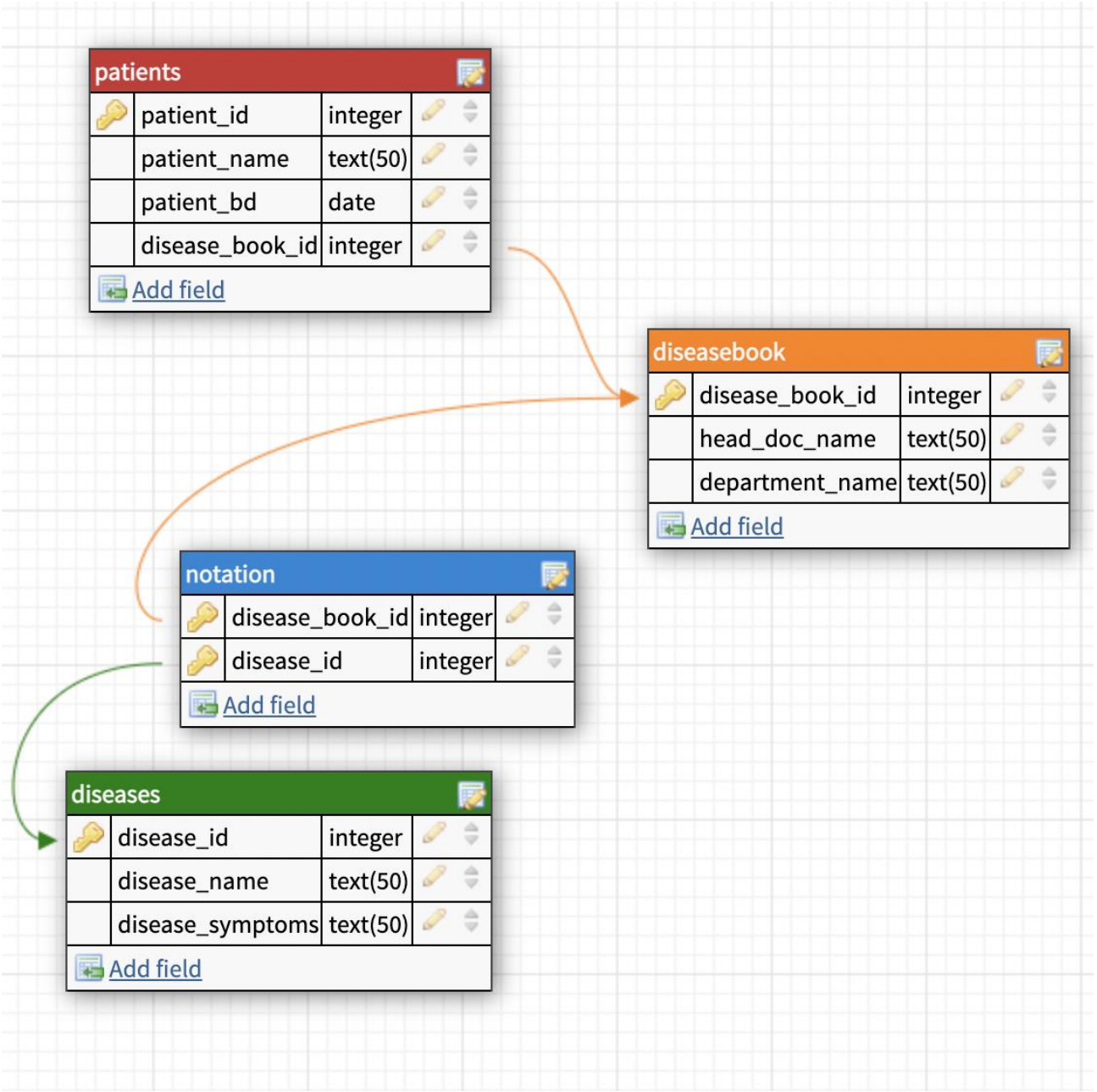


Рис.2

1c) Створюємо таблиці в СУБД PostgreSQL.

```
--Creating tables
CREATE TABLE patients(
     patient_id int UNIQUE NOT NULL,
     patient_name char(50) NOT NULL,
     patient_bd date NOT NULL,
     disease_book_id int NOT NULL);
```

```
CREATE TABLE diseasebook (
     disease_book_id int UNIQUE NOT NULL,
     head_doc_name char(50) NOT NULL,
     department_name char(50) NOT NULL);

CREATE TABLE diseases(
     disease_id int UNIQUE NOT NULL,
     disease_name char(50) NOT NULL,
     disease_symptoms char(50) NOT NULL);

CREATE TABLE notation(
     disease_book_id int UNIQUE NOT NULL,
     disease_id int UNIQUE NOT NULL);

--Setting key values
ALTER TABLE patients
ADD CONSTRAINT PK_patients
PRIMARY KEY (patient_id);

ALTER TABLE diseasebook
ADD CONSTRAINT PK_diseasebook
PRIMARY KEY (disease_book_id);

ALTER TABLE diseases
ADD CONSTRAINT PK_diseases
PRIMARY KEY (disease_id);

ALTER TABLE notation
ADD CONSTRAINT PK_notation
PRIMARY KEY (disease_book_id, disease_id);

--Relations
ALTER TABLE patients
ADD CONSTRAINT FK_patients_diseasebook
FOREIGN KEY (disease_book_id)
REFERENCES diseasebook(disease_book_id);

ALTER TABLE notation
```

```
ADD CONSTRAINT FK_notation_diseasebook
FOREIGN KEY (disease_book_id)
REFERENCES diseasebook(disease_book_id);

ALTER TABLE notation
ADD CONSTRAINT FK_notation_diseases
FOREIGN KEY (disease_id)
REFERENCES diseases(disease_id);
```

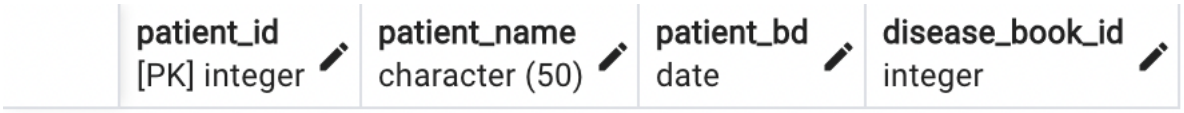


Рис. 3 *Таблиця patient*

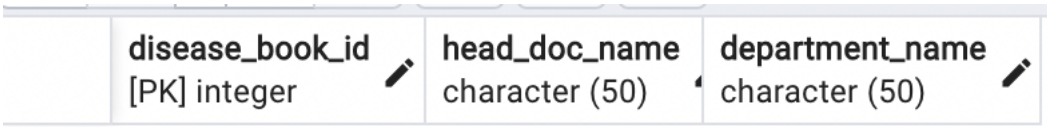


Рис.4 *Таблиця diseasebook*

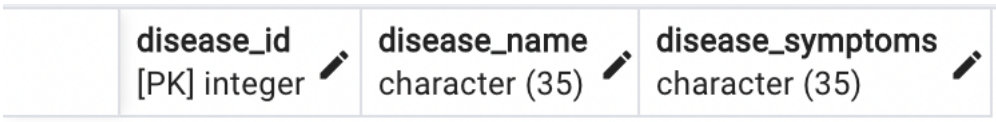


Рис.5 *Таблиця diseases*

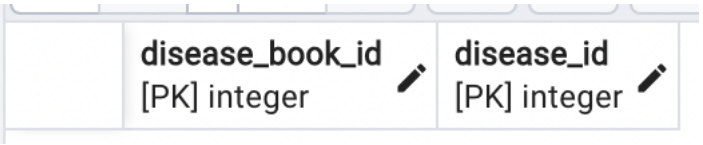


Рис. 6 *Таблиця notation (зв'язна таблиця)*

ЗАВДАННЯ 2

2a) Яка назва проданого найдешевшого товару?

- Знайдемо найдешевшу вартість:

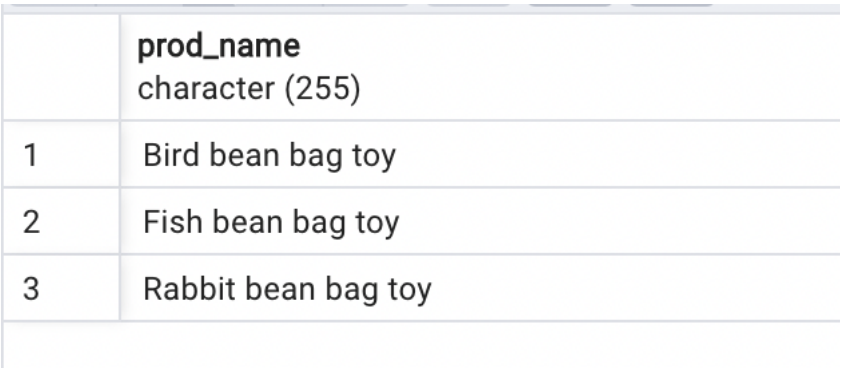
```
SELECT MIN(item_price) FROM orderitems
```

	min numeric
1	2.49

Рис.7

- Знайдемо назви найдешевших товарів (їх 3 у бд):

```
SELECT DISTINCT prod_name FROM orderitems, products
WHERE orderitems.prod_id = products.prod_id AND item_price
= (SELECT MIN(item_price) FROM  orderitems)
```



	prod_name character (255)
1	Bird bean bag toy
2	Fish bean bag toy
3	Rabbit bean bag toy

Рис.8

2b) Який PROD_ID товару з найдовшою назвою?

- Визначимо найдовшу назву

```
SELECT DISTINCT MAX(LENGTH(prod_name)) FROM products
```

	max integer
1	19

Рис.9

- Знайдемо які товари мають цю найдовшу назву

```
SELECT prod_id FROM products
WHERE LENGTH(prod_name) = (SELECT DISTINCT
MAX(LENGTH(prod_name))
FROM products)
```

	prod_id [PK] character (10)
1	BNBG03

Рис.10

2C)

- Виберемо prod_id всіх товарів, які купили:

```
SELECT DISTINCT prod_id FROM orderitems
```

	prod_id character (10)
1	BNBG01
2	BNBG03
3	BNBG02
4	BR03
5	BR01
6	RGAN01
7	BR02

Рис. 11

Виведемо PROD_ID товарів та імена постачальників для тих товарів, що були продані:

```
SELECT UPPER(prod_id)|| ''||UPPER(vend_name) AS products_sold
FROM vendors, products
WHERE vendors.vend_id = products.vend_id AND prod_id IN
(SELECT DISTINCT prod_id FROM orderitems)
```

	products_sold text
1	BR02BEARS R US
2	BR01BEARS R US
3	BR03BEARS R US
4	RGAN01DOLL HOUSE INC.
5	BNBG02DOLL HOUSE INC.
6	BNBG03DOLL HOUSE INC.
7	BNBG01DOLL HOUSE INC.

Рис.12

ЗАВДАННЯ 3

3a)

- Знайдемо найдешевшу вартість:

```
SELECT DISTINCT prod_name FROM orderitems, products WHERE
orderitems.prod_id = products.prod_id AND item_price =
(SELECT MIN(item_price) FROM  orderitems)
```

$$\boldsymbol{R1} \leftarrow \boldsymbol{MIN}(\boldsymbol{\pi_{item\_price}}(\boldsymbol{orderitems}))$$
$$\boldsymbol{R2} \leftarrow Vendors \times Products$$
$$\boldsymbol{R3} \leftarrow \sigma_{\text{orderitems.prod\_id= products.prod\_id} \wedge \text{item\_price =R1}}(R2)$$
$$\boldsymbol{R4} \leftarrow \pi_{\text{prod\_name}}(R3)$$

3b)

```
SELECT prod_id FROM products
WHERE LENGTH(prod_name) = (SELECT DISTINCT
MAX(LENGTH(prod_name))
FROM products)
```

$$\boldsymbol{R1} \leftarrow \boldsymbol{MAX}(\boldsymbol{\pi_{LENGTH(prod\_name)}}(\boldsymbol{products}))$$
$$\boldsymbol{R2} \leftarrow \sigma_{\text{item\_price =R1}}(products)$$
$$\boldsymbol{R3} \leftarrow \pi_{prod\_id}(R2)$$

3c)

```
SELECT UPPER(prod_id)|| ''||UPPER(vend_name) AS products_sold
FROM vendors, products
WHERE vendors.vend_id = products.vend_id AND prod_id IN
(SELECT DISTINCT prod_id FROM orderitems)
```

$$\boldsymbol{R1} \leftarrow \boldsymbol{\pi_{prod\_id}}(\boldsymbol{orderitems})$$
$$\boldsymbol{R2} \leftarrow Vendors \times Products$$
$$\boldsymbol{R3} \leftarrow \sigma_{\text{vendors.vend\_id= products.vend\_id} \wedge \text{prod\_id =R1}}(R2)$$
$$\boldsymbol{R4} \leftarrow \pi_{\text{UPPER(prod\_id)|| '' || UPPER(vend\_name)}}(R3)$$

СПИСОК ЛІТЕРАТУРИ

1. Дейт К. Введение в системы баз данных. − Пер. с англ. − 8-е изд. − К.: Изд. дом “Вильямс”, 2006. − 1326 с.
2. Берко А. Ю., Верес О. М., Пасічник В. В. Системи баз даних та знань. Книга 1. Організація баз даних та знань: Навчальний посібник. − Львів: “Магнолія 2006”, 2008. − 456 с.
3. Конноли Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Конноли, К. Бегг. − 3-е изд. − М.: Изд. дом “Вильямс”, 2003. − 1440 с.
4. Теория и практика построения баз данных / Д. Крёнке. − 8-е изд. − СПб: Питер, 2003. − 800 с.
5. Форта Б. Освой самостоятельно SQL. 3-е изд.: Пер. с англ. − М.: Изд. дом “Вильямс”, 2006. − 288 с.