

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

Виконав:

студент групи КМ-03

Куцалаба Н. В.

Керівник:

ст. викладач Бай Ю. П.

Київ 2022

ЗАВДАННЯ

ВАРІАНТ №8

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/alinxSorcere/SQL-in-10-minutes-with-notes.git>) (6 балів):

2a) Скільки продано найдешевшого товару?

2b) Як звуть покупця з найдовшим іменем – поле назвати `long_name`?

2c) Вивести ID та імена постачальників, що не мають жодного товару. Результат вивести у нижньому регістрі, як єдине поле `vendor_name`.

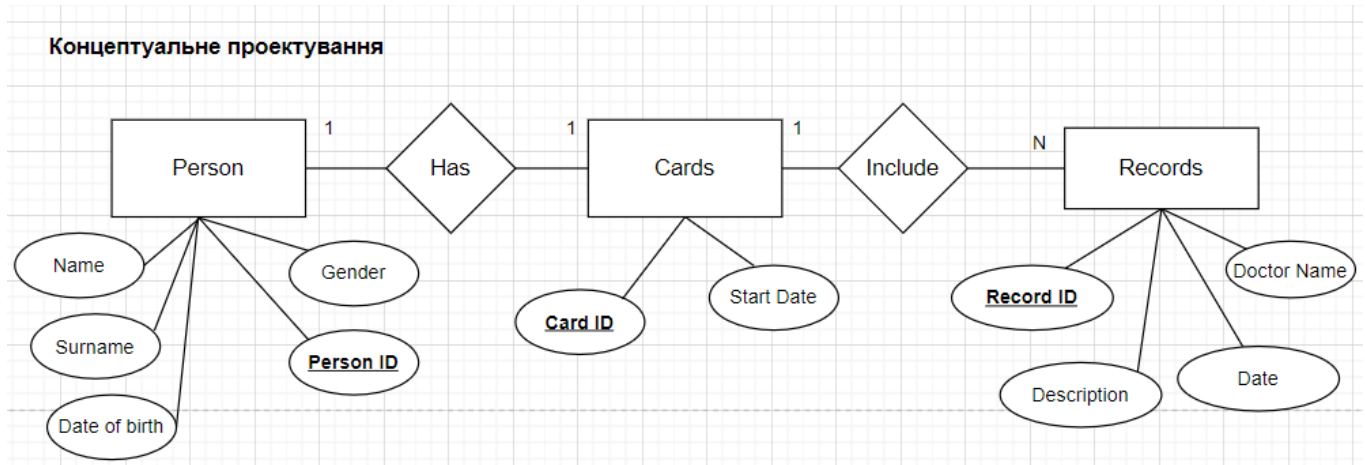
Завдання 3. Виконати запити 2a), 2b) з попереднього завдання, використовуючи операції реляційної алгебри Кодда та агрегатні функції мови SQL (4 бали).

Завдання 1

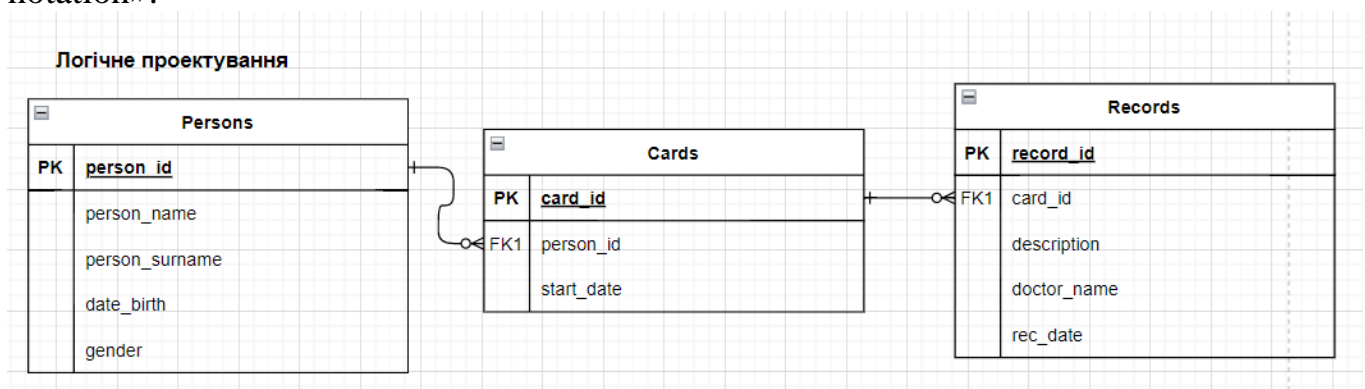
Спроекувати базу даних, що дозволить відобразити наступні події:

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.

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 “1 : N” будемо використовувати так звану з'єднуючу таблицю, яка містить значення первинних ключів.

Команди створення таблиць:

```
CREATE TABLE persons (  
    person_id int UNIQUE NOT NULL,  
    person_name char(50) NOT NULL,  
    person_surname char(50) NOT NULL,  
    date_birth date NOT NULL,  
    gender char(50) NOT NULL  
);
```

```
CREATE TABLE cards (  
    card_id int UNIQUE NOT NULL,  
    person_id int NOT NULL,  
    start_date date NOT NULL  
);
```

```
CREATE TABLE records (  
    record_id int UNIQUE NOT NULL,  
    card_id int UNIQUE NOT NULL,  
    description char(200) NULL,  
    doctor_name char(50) NOT NULL,  
    rec_date date NOT NULL  
);
```









Команди налаштування первинних та зовнішніх ключів:

```
ALTER TABLE persons ADD PRIMARY KEY (person_id);  
ALTER TABLE cards ADD PRIMARY KEY (card_id);  
ALTER TABLE records ADD PRIMARY KEY (record_id);
```

```
ALTER TABLE cards ADD CONSTRAINT FK_cards_persons FOREIGN KEY  
(person_id) REFERENCES persons (person_id);  
ALTER TABLE records ADD CONSTRAINT FK_records_cards FOREIGN KEY  
(card_id) REFERENCES cards (card_id);
```









Таблиця persons:

```
30 SELECT * FROM persons
```

Data output Messages Notifications					
       					
	person_id [PK] integer	person_name character (50)	person_surname character (50)	date_birth date	gender character (50)









Таблиця cards:

```
30 SELECT * FROM cards
```

Data output Messages Notifications			
       			
	card_id [PK] integer	person_id integer	start_date date

Таблиця records:

```
30 SELECT * FROM records
```

Data output Messages Notifications					
       					
	record_id [PK] integer	card_id integer	description character (200)	doctor_name character (50)	rec_date date

Завдання 2

Згенерувати базу даних з книги Б. Форта (*create.txt*, *populate.txt*), та виконати запити:

2а) Скільки продано найдешевшого товару?

Визначимо мінімальну вартість товарів:

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

Query

Query History

1

select min(item_price) from orderitems

2

Data output

Messages

Notifications

min

numeric

1

2.49

Використаємо це значення у вигляді вкладеного запиту, аби отримати quantity всіх проданих товарів із найменшою ціною:

```
SELECT quantity FROM orderitems  
WHERE item_price = (SELECT MIN(item_price) FROM orderitems)
```

Query

Query History

1

SELECT quantity FROM orderitems

2

WHERE item_price = (SELECT MIN(item_price) FROM orderitems)

3

Data output

Messages

Notifications

+










▼

↓

	quantity integer	
1	250	
2	250	
3	250	

Нарешті визначимо кількість продано найдешевшого товару:

```
SELECT SUM(quantity) FROM orderitems  
WHERE item_price = (SELECT MIN(item_price) from orderitems)
```

Query		Query History
1	SELECT SUM(quantity) FROM orderitems	
2	WHERE item_price = (SELECT MIN(item_price) from orderitems)	
3		
Data output		Messages Notifications
<div></div>		
	sum bigint	
1	750	

2b) Як звуть покупця з найдовшим іменем – поле назвати long_name?

Визначимо довжину найдовшого імені покупця:

```
SELECT MAX(LENGTH(cust_name)) FROM customers
```

The screenshot shows a database query tool interface. The 'Query' tab is active, displaying the SQL query: `SELECT MAX(LENGTH(cust_name)) FROM customers`. Below the query editor, the 'Data output' tab is selected, showing a single row of results. The first column is labeled 'max integer' and the value is '13'.

	max integer
1	13

Отримане значення використовуємо для отримання cust_name покупця, що має найдовше ім'я:

```
SELECT cust_name AS long_name FROM customers  
WHERE LENGTH(cust_name) = (SELECT MAX(LENGTH(cust_name)) FROM  
customers)
```

The screenshot shows a database query tool interface. The 'Query' tab is active, displaying the SQL query: `SELECT cust_name AS long_name FROM customers WHERE LENGTH(cust_name) = (SELECT MAX(LENGTH(cust_name)) FROM customers)`. Below the query editor, the 'Data output' tab is selected, showing a single row of results. The first column is labeled 'long_name character (50)' and the value is 'The Toy Store'.

	long_name character (50)
1	The Toy Store

2с) Вивести ID та імена постачальників, що не мають жодного товару. Результат вивести у нижньому регістрі, як єдине поле vendor_name.

Виберемо vend_id всіх товарів(без повторів):

```
SELECT distinct vend_id FROM products
```

Query		Query History
1	SELECT distinct vend_id FROM products	
2		
3		
Data output		Messages Notifications
	vend_id character (10) 🔒	
1	FNG01	
2	BRS01	
3	DLL01	









Наступним кроком виберемо vend_id та vend_name тих постачальників, які не мають товарів:

```
SELECT vend_id, vend_name FROM vendors  
WHERE vend_id NOT IN (SELECT distinct vend_id FROM products)
```

Query		Query History
1	SELECT vend_id, vend_name FROM vendors	
2	WHERE vend_id NOT IN (SELECT distinct vend_id FROM products)	
3		
Data output		Messages Notifications
	vend_id [PK] character (10) ✎	vend_name character (50) ✎
1	BRE02	Bear Emporium
2	FRB01	Furball Inc.
3	JTS01	Jouets et ours

Знайдемо та виведемо у нижньому реєстрі id та імена постачальників які не мають товарів, як єдине поле vendor_name:

```
SELECT CONCAT(LOWER(vend_id), ' ', LOWER(vend_name)) AS vendor_name
FROM vendors
WHERE vend_id NOT IN (SELECT distinct vend_id FROM products)
```

Query		Query History
1	SELECT CONCAT(LOWER(vend_id), ' ', LOWER(vend_name)) AS vendor_name FROM vendors	
2	WHERE vend_id NOT IN (SELECT distinct vend_id FROM products)	
3		
4		
Data output		Messages Notifications
<div></div>		
	vendor_name	
	text	
1	bre02 bear emporium	
2	frb01 furball inc.	
3	jts01 jouets et ours	

Завдання 3

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

2a)

Введемо допоміжне позначення мінімальної ціни товару:

$$Mp = MIN(\pi_{item_price}(orderitems));$$

Далі визначимо quantity проданих товарів з найменшою ціною:

$$Q = \pi_{quantity}(\sigma_{item_price = Mp}(orderitems))$$

Нарешті визначимо кількість продано найдешевшого товару:

$$SUM(Q)$$

2b)

Позначимо кількість літер у найдовшій назві товару наступним чином:

$$Ml = MAX(\pi_{LENGTH(cust_name)}(customers))$$

Визначимо cust_name покупця, що має найдовше ім'я:

$$Ln = \pi_{cust_name}(\sigma_{LENGTH(cust_name) = Ml}(customers))$$

Замінімо назву поля на long_name:

$$P_{long_name}(Ln)$$

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. Системы 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. Форта Б. Освой самостоятельно SQL. 3-е изд.: Пер. с англ. – М.: Изд. дом “Вильямс”, 2006. – 288 с.