

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

*Передерей Б. О.*

Керівник:

*ст. викладач Бай Ю. П.*

## ЗМІСТ

ЗАВДАННЯ	2
Завдання 1	3
Завдання 2	5
Завдання 3	9
Список літератури	10

## ЗАВДАННЯ

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

Власник гугл-акаунту завантажує файли на гугл-диск.

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

1б) 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>) (6 балів):

2а) Скільком покупцям продано найдешевший товар?

2б) Який PROD\_ID товару з найдовшою назвою?

2с) Вивести імена постачальників, які мають товар і його хтось купив. Імена вивести у верхньому регістрі, назвавши це поле *vendor\_name*.

**Завдання 3.** Виконати запити 2а), 2б) 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:

**Власник гугл-акаунту завантажує файли на гугл-диск.**

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

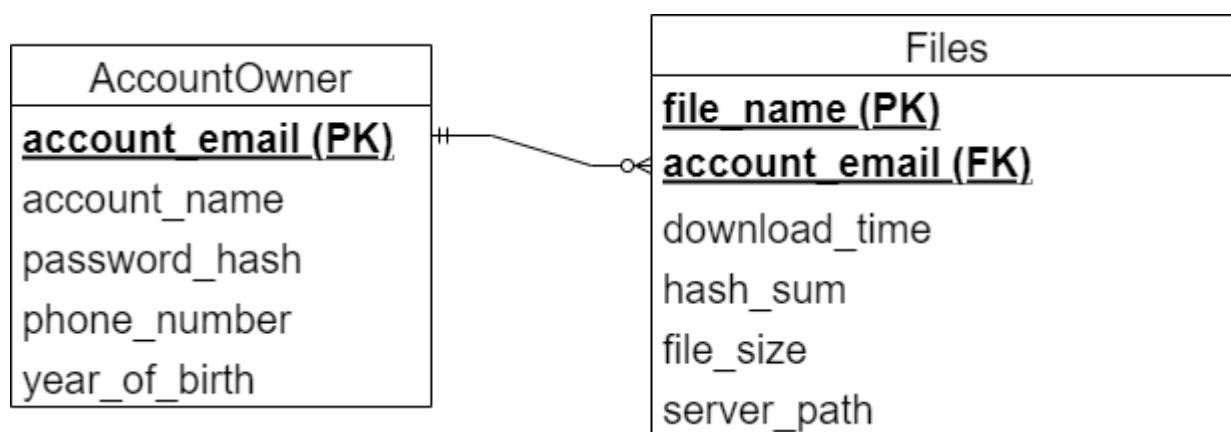
Основні дві сутності – власник гугл-акаунту та файли. Оскільки один власник може завантажити будь-яку кількість файлів, а окремий файл може бути завантаженим лише одним власником, то їхній зв'язок один до багатьох.

Основні атрибути виділені на ER-діаграмі:

Власник гугл-акаунту завантажує  
файл на гугл-диск



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:

```
CREATE TABLE AccountOwner (
  account_email varchar(80) NOT NULL,
  account_name varchar(60) NOT NULL,
  password_hash bytea NOT NULL,
  phone_number char(20) NULL,
  year_of_birth date NULL
);
```

```
CREATE TABLE Files (
  file_name varchar(255) NOT NULL,
  account_email varchar(80) NOT NULL,
  download_time TIMESTAMP NOT NULL,
  hash_sum bytea NOT NULL,
  file_size INTEGER NOT NULL,
  server_path varchar(512) NOT NULL
);
```

```
ALTER TABLE AccountOwner ADD CONSTRAINT PK_AccountOwner PRIMARY KEY (account_email);
```

```
ALTER TABLE Files ADD CONSTRAINT PK_Files PRIMARY KEY (file_name);
```

```
ALTER TABLE Files ADD CONSTRAINT FK_Files_AccountOwner FOREIGN KEY (account_email)
REFERENCES AccountOwner (account_email);
```

Результати виконання команд:

25

SELECT

\*

FROM

accountowner;

Data output

Messages

Notifications

account\_email

[PK] character varying (80)

account\_name

character varying (60)

password\_hash

bytea

phone\_number

character (20)

year\_of\_birth

date

26

SELECT

\*

FROM

files;

Data output

Messages

Notifications

file\_name

[PK] character varying (255)

account\_email

character varying (80)

download\_time

timestamp without time zone

hash\_sum

bytea

file\_size

integer

server\_path

character varying (512)

## Завдання 2

Згенерувати базу даних з книги Б. Форта та виконати запити:

### 2а) Скільком покупцям продано найдешевший товар?

Для початку треба отримати prod\_id товарів з мінімальною ціною:

**SELECT prod\_id FROM products**

**WHERE prod\_price = (SELECT MIN(prod\_price) FROM products);**

The screenshot shows a SQL query execution interface. The query is: `SELECT prod_id FROM products WHERE prod_price = (SELECT MIN(prod_price) FROM products);`. The results are displayed in a table with the following data:

	prod_id [PK] character (10)
1	BNBG01
2	BNBG02
3	BNBG03

Тепер потрібно вибрати усі замовлення (order\_num) з orderitems, які замовляли вище отримані товари, для видалення дублікатів використовується distinct():

**SELECT DISTINCT(order\_num) FROM orderitems**

**WHERE prod\_id in (SELECT prod\_id FROM products**

**WHERE prod\_price = (SELECT MIN(prod\_price) FROM products))**

The screenshot shows a SQL query execution interface. The query is: `SELECT DISTINCT(order_num) FROM orderitems WHERE prod_id in (SELECT prod_id FROM products WHERE prod_price = (SELECT MIN(prod_price) FROM products))`. The results are displayed in a table with the following data:

	order_num integer
1	20008
2	20009
3	20007

Після цього залишається порахувати кількість унікальних покупців, які замовляли найдешевший товар, рахуючи їх cust\_id, якщо по замовленням, які враховували у себе замовлення найдешевшого товару, використовуючи наступний вкладений запит:

**SELECT COUNT(distinct(cust\_id)) FROM orders**

**WHERE order\_num in ((SELECT distinct(order\_num) FROM orderitems**

```
WHERE prod_id in (SELECT prod_id FROM products
WHERE prod_price = (SELECT MIN(prod_price) FROM products))))
```

```

9 SELECT COUNT(distinct(cust_id)) FROM orders
10 WHERE order_num in ((SELECT distinct(order_num) FROM orderitems
11 WHERE prod_id in (SELECT prod_id FROM products
12 WHERE prod_price = (SELECT MIN(prod_price) FROM products))))
13

```

Data output		Messages	Notifications
	count bigint		
1	3		

## 2b) Який PROD\_ID товару з найдовшою назвою?

Спочатку знаходиться довжина найдовшого ім'я за допомогою наступного запиту і функції MAX():

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

```

15 SELECT MAX(LENGTH(prod_name)) FROM product

```

Data output		Messages	Notifications
	max integer		
1	19		

Наступним кроком знаходяться всі prod\_id, які мають довжину prod\_name, яка дорівнює довжині, визначеній запитом вище:

```
SELECT prod_id FROM products
```

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

```

17 SELECT prod_id FROM products
18 WHERE LENGTH(prod_name) = (SELECT MAX(LENGTH(prod_name)) FROM products)
19

```

Data output		Messages	Notifications
	prod_id [PK] character (10)		
1	BNBG03		

**2с) Вивести імена постачальників, які мають товар і його хтось купив. Імена вивести у верхньому регістрі, назвавши це поле vendor\_name.**

Першим кроком знаходяться всі prod\_id товарів, які замовлялися (таблиця orderitems):

**SELECT distinct(prod\_id) FROM orderitems**

26 **SELECT distinct**(prod\_id) **FROM** orderitems

Data output Messages Notifications

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

Використовуючи вкладені запити, потрібно знайти vend\_id постачальників, які мають товар та їх товар замовляли:

**SELECT distinct(vend\_id) FROM products**

**WHERE prod\_id IN (SELECT distinct(prod\_id) FROM orderitems)**

28 **SELECT distinct**(vend\_id) **FROM** products  
29 **WHERE** prod\_id **IN** (**SELECT distinct**(prod\_id) **FROM** orderitems)

Data output Messages Notifications

	vend_id character (10) 🔒
1	BRS01
2	DLL01

У кінці лишається знайти vend\_name постачальників з отриманими vend\_id та вивести їх у верхньому регістрі за допомогою UPPER() і позначити це поле як vendor\_name:

**SELECT UPPER(vend\_name) AS vendor\_name FROM vendors**

**WHERE vend\_id IN (SELECT distinct(vend\_id) FROM products**



**WHERE prod\_id IN (SELECT distinct(prod\_id) FROM orderitems))**

```

31 SELECT UPPER(vend_name) AS vendor_name FROM vendors
32 WHERE vend_id IN (SELECT distinct(vend_id) FROM products
33     WHERE prod_id IN (SELECT distinct(prod_id) FROM orderitems))

```

Data output		Messages	Notifications
<div> <div>≡</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div>			
	<div> <div>vendor_name</div> <div>text</div> <div>🔒</div> </div>		
1	BEARS R US		
2	DOLL HOUSE INC.		

Крім цього, такий самий результат можна було отримати, використовуючи декартовий добуток відповідних таблиць:

**SELECT DISTINCT(UPPER(vend\_name)) AS vendor\_name from vendors, products, orderitems**

**WHERE vendors.vend\_id = products.vend\_id and products.prod\_id = orderitems.prod\_id**

```

38 SELECT DISTINCT(UPPER(vend_name)) AS vendor_name from vendors, products, orderitems
39 WHERE vendors.vend_id = products.vend_id and products.prod_id = orderitems.prod_id

```

Data output		Messages	Notifications
<div> <div>≡</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div>			
	<div> <div>vendor_name</div> <div>text</div> <div>🔒</div> </div>		
1	DOLL HOUSE INC.		
2	BEARS R US		

### Завдання 3

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

#### 2a) Скільком покупцям продано найдешевший товар?

Першим кроком буде знаходження мінімальної ціни товару:

$$R1 \leftarrow MIN(\pi_{prod\_price}(products))$$

Знаходження prod\_id товарів з мінімальною ціною:

$$R2 \leftarrow \pi_{prod\_id}(\sigma_{prod\_price=R1}(products))$$

Знаходження order\_num з відповідними prod\_id:

$$R3 \leftarrow \pi_{DISTINCT(order\_num)}(\sigma_{prod\_id \text{ IN } R2}(orderitems))$$

Отримання значення покупців, яким було продано найдешевший товар:

$$R4 \leftarrow COUNT(\pi_{DISTINCT(cust\_id)}(\sigma_{order\_num \text{ IN } R3}(orders)))$$

#### 2b) Який PROD\_ID товару з найдовшою назвою?

Для початку треба визначити максимальну довжину ім'я:

$$R1 \leftarrow MAX(\pi_{LENGTH(prod\_name)}(products))$$

Потім вибрати всі рядки, які відповідають R1:

$$R2 \leftarrow \sigma_{LENGTH(prod\_name) = R1}(products)$$

І у кінці вибрати стовпець prod\_id з R2:

$$R3 \leftarrow \pi_{prod\_id}(R2)$$

### **Список літератури**

1. An Introduction to Database Systems, 8th edition / C.J.Date. – Pearson Education, 2003. – 1040 p.
2. Database systems. A practical approach to design, implementation and management / T.M. Connolly, C.E. Begg. – Addison-Wesley, 2005. – 1374 p.
3. Relational Database Design and Implementation, 4th Edition / J. Harrington.– Morgan Kaufmann, 2016. – 712 p