

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

Варіант 6

Виконала:

студентка групи КМ-93

Германчук А. В.

Керівник:

ст. викладач Бай Ю. П.

Київ — 2021

ЗАВДАННЯ

Варіант №6

Завдання 1. Згенерувати базу даних з книги Б. Форта (*create.txt, populate.txt*), та виконати запити (6 балів):

1a) Як звуть постачальника, що продав найдешевший товар?

1b) Вивести імена покупців, що мають поштову адресу та живуть в USA, назвавши це поле `client_name`.

1c) Вивести імена постачальників у нижньому регістрі, назвавши це поле `vendor_name`, що мають товар, але його ніхто не купляв.

Завдання 2. Виконати запити 1a), 1b), використовуючи операції реляційної алгебри Кодда та агрегатні функції мови SQL (4 бали)

Завдання 3. За допомогою команд мови SQL створити таблиці, згідно з умовою:

Громадянин України має власне житло та автомобіль.

Визначити поля та типи. Головні та зовнішні ключі створювати окремо від таблиць, використовуючи команду ALTER TABLE. (5 балів)

Завдання 1

Згенерувати базу даних з книги Б. Форта (*create.txt*, *populate.txt*), та виконати запити (6 балів):

1а) Як звуть постачальника, що продав найдешевший товар?

В запиті використані агрегатна функція MIN(), аргументом якої є стовпчик, для підрахунку мінімуму.

Визначимо мінімальну вартість проданого товару:

```
SELECT MIN(item_price)
```

```
FROM orderitems
```

Результат виконання:

	min numeric
1	2.49

Використаємо отримане значення у вигляді вкладеного запиту, виведемо імена постачальників, які продавали товари з отриманою ціною:

```
SELECT vend_name
```

```
FROM (orderitems NATURAL JOIN products NATURAL JOIN vendors)
```

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

Результат виконання:

	vend_name character (50)
1	Doll House Inc.
2	Doll House Inc.
3	Doll House Inc.

Прибираємо повтори:

```
SELECT DISTINCT vend_name  
FROM (orderitems NATURAL JOIN products NATURAL JOIN vendors)  
WHERE item_price = (SELECT MIN(item_price) FROM orderitems)
```

Результат виконання:

	vend_name character (50)	
1	Doll House Inc.	...

1b) Вивести імена покупців, що мають поштову адресу та живуть в USA, назвавши це поле client_name.

Виводимо імена усіх покупців, які живуть в

```
USA SELECT cust_name FROM  
customers WHERE cust_country = 'USA'
```

Результат виконання:

Data Output		Explain
	cust_name character (50)	
1	Village Toys	...
2	Kids Place	...
3	Fun4All	...
4	Fun4All	...
5	The Toy Store	...

Серед отримани назв обираємо тих у кого є поштова адреса:

```
SELECT cust_name FROM  
customers WHERE cust_country  
= 'USA'  
AND cust_email IS NOT NULL
```


Результат виконання:

Data Output		Explain
	cust_name character (50) 	
1	Village Toys	...
2	Fun4All	...
3	Fun4All	...

Змінюємо назву поля на **client_name**:

```
SELECT  
cust_name AS  
client_name  
FROM customer  
WHERE cust_country =  
'USA' AND cust_email IS  
NOT NULL
```

Результат виконання:

	client_name character (50) 	
1	Village Toys	...
2	Fun4All	...
3	Fun4All	...

1c) Вивести імена постачальників у нижньому регістрі, назвавши це поле `vendor_name`, що мають товар, але його ніхто не купляв.

Виводимо id товарів, які ніхто не купляв:

```
SELECT prod_id
FROM products
WHERE prod_id NOT IN (SELECT prod_id FROM orderitems)
```

Результат виконання:

	Data Output	Explain
	prod_id [PK] character (10)	
1	RYL01	
2	RYL02	

Виводимо імена продавців цього товару:

```
SELECT vend_name
FROM vendors, products
WHERE prod_id NOT IN (SELECT prod_id
FROM orderitems)
AND vendors.vend_id = products.vend_id
```

Результат виконання:

	Data Output	Explain	Message
	vend_name character (50)		
1	Fun and Games		...
2	Fun and Games		...

Змінюємо назву поля на **vendor_name**, прибираємо повтори, виводимо імена у нижньому регістрі:

```
SELECT DISTINCT
```

```
  LOWER(vend_name) AS vendor_name
```

```
FROM vendors, products
```

```
WHERE prod_id NOT IN (SELECT prod_id FROM
```

```
orderitems) AND vendors.vend_id = products.vend_id
```

Результат виконання:

Data Output		Explain
	vendor_name text	
1	fun and games	

Завдання 2

Виконати запити 1a), 1b), використовуючи операції реляційної алгебри Кодда та агрегатні функції мови SQL (4 бали)

1a)

Найдешевший товар:

$$M_p = MIN(\pi_{item_price}(orderitems))$$

Продавці, що продали найдешевший товар:

$$\pi_{vend_name}(\sigma_{item_price = M_p}(orderitems \bowtie products \bowtie vendors))$$

1b)

Покупці, що живуть в USA:

$$R_1 = \sigma_{cust_country="USA"}(customers)$$

Покупці, що мають поштову адресу:

$$R_2 = \sigma_{cust_email \neq NULL}(customers)$$

Виводимо імена покупців з USA, які мають поштову адресу, назвавши поле client_name:

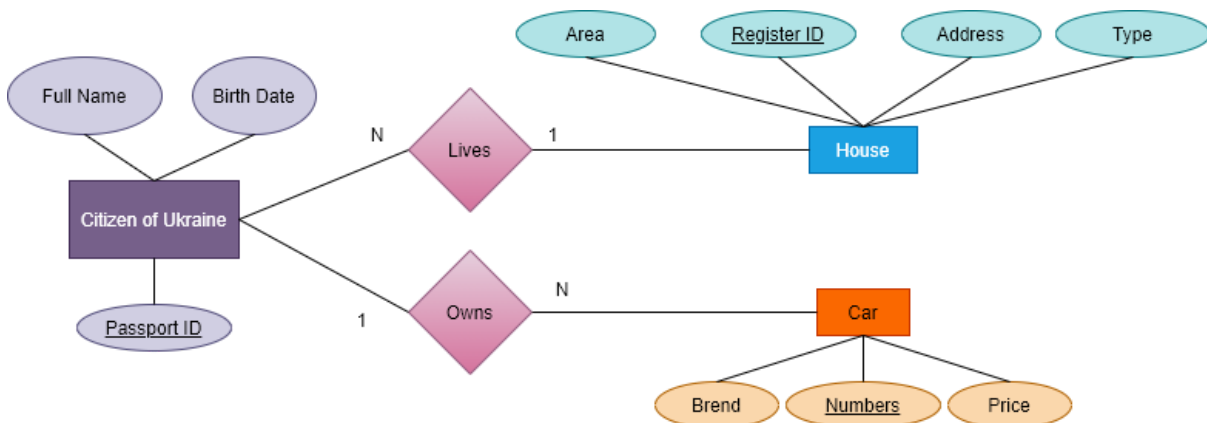
$$\rho_{client_name}(R_1 \cap R_2)$$

Завдання 3

За допомогою команд мови SQL створити таблиці, згідно з умовою:

Громадянин України має власне житло та автомобіль.

Визначити поля та типи. Головні та зовнішні ключі створювати окремо від таблиць, використовуючи команду ALTER TABLE. (5 балів) ER схема:



Команди створення таблиць:

```
CREATE TABLE
Houses (
  reg_id int UNIQUE NOT NULL ,
  house_area int NOT NULL ,
  house_address char(50) NOT
  NULL, house_type char(50) NOT
  NULL
);
```

```
CREATE TABLE
CitizensOfUkraines (
  passport_id char(10) UNIQUE
  NOT NULL , full_name char(50)
  NOT NULL , birth_date date
  NOT NULL , reg_id int NULL
);
```

```
CREATE TABLE
Cars (
car_numb char(10)
UNIQUE NOT NULL ,
car_brend char(50)
NOT NULL , car_price
char(50) NOT NULL ,
passport_id char(10) NOT
NULL
);
```

Команди налаштування первинних та зовнішніх ключів:

```
ALTERTABLE CitizensOfUkraines ADD CONSTRAINT
PK_Citizens PRIMARY KEY
(passport_id);
ALTER TABLE Cars ADD CONSTRAINT
PK_Cars PRIMARY KEY
(car_num);
ALTER TABLE Houses ADD CONSTRAINT PK_Houses PRIMARY KEY
(reg_id);

ALTER TABLE CitizensOfUkraines
ADD CONSTRAINT FK_Citizens_Houses FOREIGN KEY (reg_id)
REFERENCES
Houses (reg_id);
ALTER TABLE Cars
ADD CONSTRAINT FK_Cars_Citizens FOREIGN
KEY (passport_id) REFERENCES
CitizensOfUkraines
(passport_id);
```

Таблиця CitizensOfUkraines:

Data Output	Explain	Messages	Query History	Notifications
	passport_id [PK] character (10) 	full_name character (50) 	birth_date date 	reg_id character (10) 

Таблиця Houses:

Data Output	Explain	Messages	Query History	Notifications
	reg_id [PK] character (1) 	house_area integer 	house_address character (50) 	house_type character (50) 

Таблица Cars:

Data Output					Explain	Messages	Query History	Notifications
	car_num [PK] character (10) 	car_brend character (50) 	car_price character (50) 	passport_id character (10) 				

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.