

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

із лабораторної роботи №1

із 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-93

Торба С.О.

Керівник:

ст. викладач Бай Ю. П.

Київ — 2021

ЗМІСТ

Завдання **Ошибка! Закладка не определена.**

Завдання 1 **Ошибка! Закладка не определена.**

Завдання 2 **Ошибка! Закладка не определена.**

Завдання 3 **Ошибка! Закладка не определена.**

Список літератури10

ЗАВДАННЯ

Варіант №18

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

1a) Яка назва проданого найдорожчого товару?

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

1c) Вивести ім'я та пошту покупця, як єдине поле `client_name`, для тих покупців, що не мають замовлення.

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:

Людина танцює під музику.

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) Яка назва проданого найдорожчого товару?

В запиті використані агрегатні функції MAX() та SUM() для підрахунку максимуму та суми відповідно. Аргументами даних функцій є окремі стовпчики

Визначимо максимальну вартість проданого товару

SELECT MAX(item_price) FROM orderitems

Результат виконання:

| Data Output | Explain | Messages | Notifications |
|---|---------|----------|---------------|
| <div>max numeric</div> <div>11.99</div> | | | |

Використаємо отримане значення у вигляді вкладеного запиту:

SELECT (prod_name) FROM products WHERE prod_price = (SELECT MAX(item_price) FROM orderitems)

Результат виконання:

| Data Output | Explain | Messages | Notifications |
|--|---------|----------|---------------|
| <div>prod_name character (255)</div> <div>18 inch teddy bear ...</div> | | | |

Додаток 1

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

Отримаємо імена усіх покупців за допомогою запиту

SELECT (cust_name) AS client_name FROM customers WHERE cust_country='USA' and cust_email IS NOT NULL

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

1c) Вивести ім'я та пошту покупця, як єдине поле client_name, для тих покупців, що не мають замовлення.

В запиті використана операція поєднання таблиць NATURAL JOIN та конкатенація рядків (операція «||»). Для обчислення різниці між таблицями застосовується операція EXCEPT.

Отримаємо імена та пошти усіх покупців:

SELECT (cust_name || ' ' || COALESCE(cust_email, '')) AS client_name FROM customers;

В даному запиті використана функція COALESCE(cust_email, ''), яка повертає

cust_email, якщо він не NULL, в іншому випадку повертається пустий рядок.

Результат виконання:

| | client_name text |
|---|------------------------------------|
| 1 | Village Toys sales@villagetoys.com |
| 2 | Kids Place |
| 3 | Fun4All jjones@fun4all.com |
| 4 | Fun4All dstephens@fun4all.com |
| 5 | The Toy Store |

Додаток 1

Визначимо імена та пошти покупців, що зробили замовлення:

```
SELECT (cust_name || ' ' || COALESCE(cust_email, '')) AS client_name FROM
(customers NATURAL JOIN orders)
```

Результат виконання

| | client_name text |
|---|------------------------------------|
| 1 | Village Toys sales@villagetoys.com |
| 2 | Fun4All jjones@fun4all.com |
| 3 | Fun4All dstephens@fun4all.com |
| 4 | The Toy Store |
| 5 | Village Toys sales@villagetoys.com |

За допомогою віднімання таблиць, визначимо покупців, що не мають замовлень:

```
SELECT(cust_name || ' ' || COALESCE(cust_email, '')) AS client_name FROM
customers
```

```
EXCEPT
```

```
SELECT (cust_name || ' ' || COALESCE(cust_email, '')) AS client_name FROM
(customers NATURAL JOIN orders)
```

Результат виконання:

| | client_name text |
|---|---------------------|
| 1 | Kids Place |

Додаток 1

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

1a) Введемо допоміжне значення максимальної ціни товару:

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

Визначимо назву найдорожчого проданого товару

$$(\pi_{prod_name} (\sigma_{prod_price = Mp} (products)));$$

$$1b) \rho_{S(client_name)}(\sigma_{cust_country = USA \Delta cust_email \neq NULL}(customers))$$

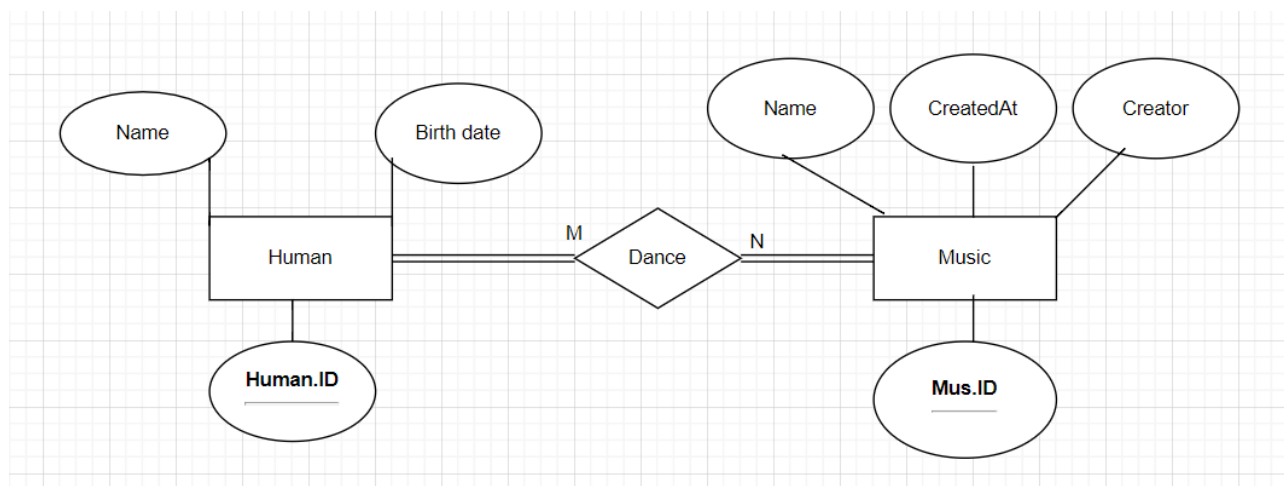
Додаток 1

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

Людина танцює під музику.

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

ER схема:



Для реалізації зв'язку “M:N” створимо додаткову сутність Human_Music , де додамо атрибути – “зовнішні ключі”.

Команди створення таблиць :

```
CREATE TABLE Humans
```

```
(
    hum_id    int    UNIQUE NOT NULL ,
    hum_name  char(50) NOT NULL ,
    hum_bdate date    NOT NULL
);
```

```
CREATE TABLE Musics
```

```
(
    mus_id    int    UNIQUE NOT NULL ,
    mus_name  char(50) NOT NULL ,
    mus_creator char(50) NOT NULL ,
```


Додаток 1

```
mus_crdate date NOT NULL

);
```

```
CREATE TABLE Humans_Musics
```

```
(
    hum_id int NOT NULL ,
    mus_id int NOT NULL
);
```

Команди налаштування первинних та зовнішніх ключів

```
ALTER TABLE Humans ADD CONSTRAINT PK_Humans PRIMARY KEY
(hum_id);
```

```
ALTER TABLE Musics ADD CONSTRAINT PK_Musics PRIMARY KEY
(mus_id);
```

```
ALTER TABLE Humans_Musics
```

```
ADD CONSTRAINT FK_Human_ID FOREIGN KEY (hum_id) REFERENCES
Humans (hum_id);
```

```
ALTER TABLE Humans_Musics
```






```
ADD CONSTRAINT FK_Music_ID FOREIGN KEY (mus_id) REFERENCES
Musics (mus_id);
```

Таблиця Humans:




| hum_id | hum_name | hum_bdate |
|------------|----------------|-----------|
| [PK] integ | character (50) | date |
| | | |

Таблиця Musics:

Додаток 1

| Data Output | Explain | Messages | Notifications |
|---|---|--|---|
|  mus_id [PK] integ  | mus_name character (50)  | mus_creator character (50)  | mus_crdate date  |
| | | | |

Таблиця Humans_Musics :

| Data Output | Explain | Messages | Notificat |
|--|--|----------|-----------|
|  hum_id integer  | mus_id integer  | | |
| | | | |

Додаток 1

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. 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.