НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

з 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»

|  |  |
| --- | --- |
| Виконала: | Керівник: |
| студентка групи КМ-91 | ст. викладач Бай Ю.П. |
| Павловська К.І. |  |

Київ – 2021

ЗМІСТ

[ЗАВДАННЯ 3](#_Toc84011023)

[Завдання 1 4](#_Toc84011024)

[Завдання 2 7](#_Toc84011025)

[Завдання 3 8](#_Toc84011026)

[СПИСОК ЛІТЕРАТУРИ 11](#_Toc84011027)

ЗАВДАННЯ

Варіант №8

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

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

1b) Який PROD\_ID товару з найдовшою назвою?

1c) Вивести імена постачальників у нижньому регістрі, назвавши це поле vendor\_name, які мають товар, але його ніхто не купляв.

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

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

Студент має заліковку що містить записи про дисципліни.

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

Завдання 1

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

**1a)** Як звуть постачальника, який продав найдорожчий товар?

У запиті використана агрегатна функція MAX() для підрахунку максимуму (визначення найдорожчого товару) і оператор DISTINCT для роботи з унікальними значеннями атрибуту.

Визначимо ціну найдорожчого проданого товару:

select min(item\_price) as min\_item\_price  
from orderitems;

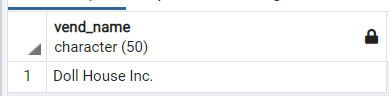
Результат виконання:



Використаємо отримане значення у вкладеному запиті:

select distinct vendors.vend\_name  
from orderitems,  
 products,  
 vendors  
where item\_price = (select min(item\_price) from orderitems)  
 and products.prod\_id = orderitems.prod\_id  
 and vendors.vend\_id = products.vend\_id

Результат виконання:



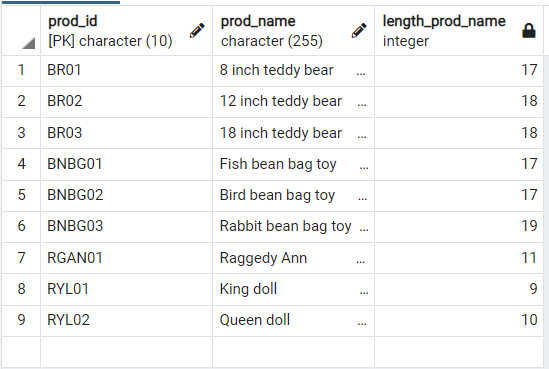
**1b)** Який PROD\_ID товару з найдовшою назвою?

Окрім агрегатної функції MAX(), у цьому завданні використовується функція LENGTH(), яка повертає довжину рядка.

Виведемо prod\_id кожного товару, його назву та довжину назви:

select prod\_id,  
 prod\_name,  
 length(prod\_name) as length\_prod\_name  
from products

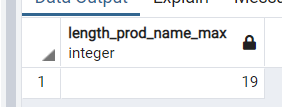
Результат виконання:



Визначимо максимальну довжину назви товару:

select max(length(prod\_name)) as length\_prod\_name\_max  
from products

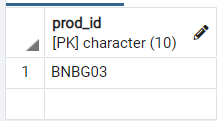
Результат виконання:



Використаємо одержану інформацію як вкладений запит:

select prod\_id  
from products  
where length(prod\_name) = (select max(length(prod\_name)) from products)

Результат виконання:



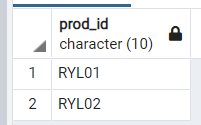
**1c)** Вивести імена постачальників у нижньому регістрі, назвавши це поле vendor\_name, які мають товар, але його ніхто не купляв.

У запиті використаний оператор DISTINCT для роботи з унікальними значеннями атрибуту, функція LOWER() для перетворення записів у нижній регістр та операція EXCEPT для обчислення різниці між таблицями.

Визначимо prod\_id товару, який ніхто не купляв, обчисливши різницю між таблицями products і orderitems:

select products.prod\_id  
from products except  
select orderitems.prod\_id  
from orderitems

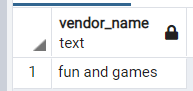
Результат виконання:



Використовуючи отриману інформацію, виведемо імена постачальників у нижньому регістрі, назвавши це поле vendor\_name, які мають товар, але його ніхто не купляв:

select distinct lower(vendors.vend\_name) as vendor\_name  
from vendors,  
 products,  
 orderitems  
where products.prod\_id in (select products.prod\_id from products except select orderitems.prod\_id from orderitems)  
 and vendors.vend\_id = products.vend\_id

Результат виконання:



Завдання 2

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

**1a)**

Введемо допоміжне позначення максимальної ціни проданого товару:

;

Визначимо, як звуть постачальника, що продав найдорожчий товар:

.

**1b)**

Позначимо кількість літер у найдовшій назві товару як:

Визначимо prod\_id товару з найдовшою назвою:

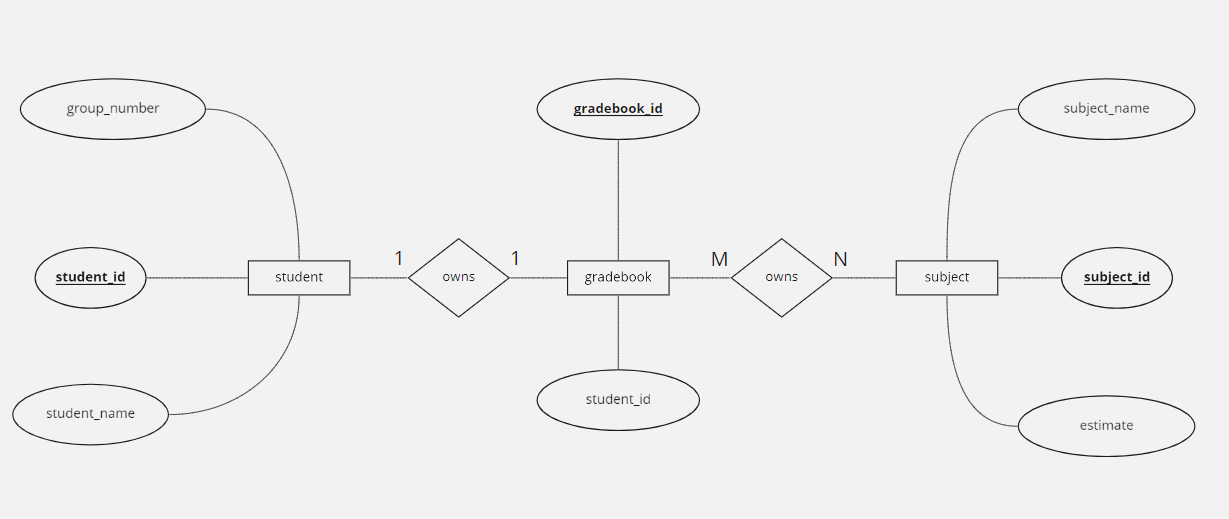
Завдання 3

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

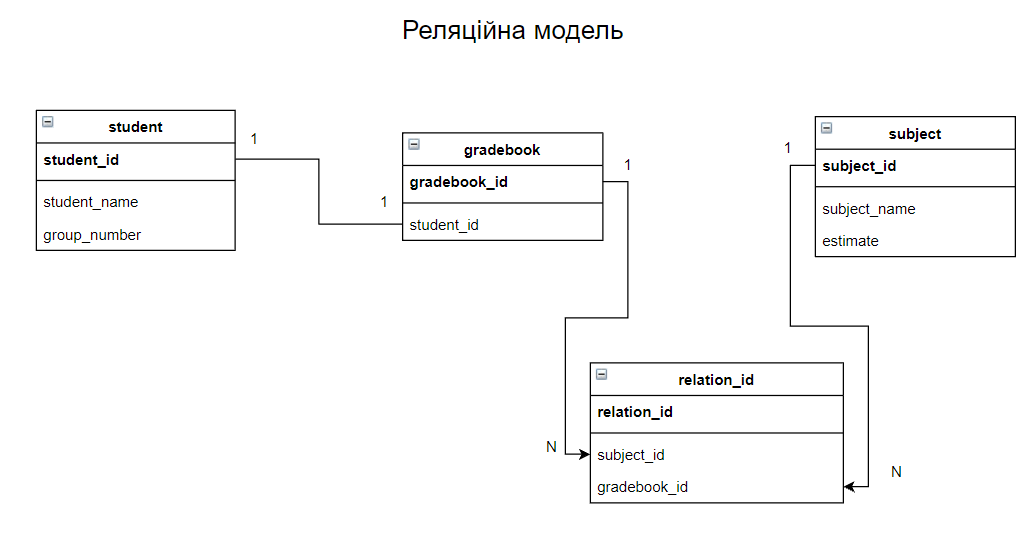
**Студент має заліковку що містить записи про дисципліни.**

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

ER-схема:



Реляційна модель:



**Концепція :** «Студент має залікову книжку, що містить записи про дисципліни»

**Сутності:**1. Студент  
2. Залікова книжка  
3. Дисципліна **Таблиці**  
1. student – таблиця з інформацією про студента

2. gradebook – таблиця з інформацією про залікову книжку

3. subject – таблиця з інформацією про дисципліни

4. relation\_gradebook\_subject – таблиця зв’язків для gradebook та subject

**Зв’язки**

1. Між сутностями Студент та Залікова книжка встановлено зв’язок типу 1:1.  
2. Між сутностями Залікова книжка та Дисципліна встановлено зв’язок M:N.

Для зв'язку «many to many» (пункт 2 «зв’язки») необхідно створити додаткову таблицю зв'язків, атрибутами якої є зовнішні ключі, відповідні первинним ключам таблиць, що пов’язуються.

Назвемо таблицю зв’язків «relation\_gradebook\_subject», та створимо необхідні ключі.

**Команди створення таблиць:**

create table student  
(  
 student\_id char(50) unique not null,  
 student\_name char(60) not null,  
 group\_number char(20) not null  
);

create table gradebook  
(  
 gradebook\_id char(50) unique not null,  
 student\_id char(50) not null  
);

create table subject  
(  
 subject\_id char(50) unique not null,  
 student\_name char(100) not null,  
 estimate int  
);

create table relation\_gradebook\_subject  
(  
 relation\_id char(50) unique not null,  
 subject\_id char(50) unique not null,  
 gradebook\_id char(50) unique not null  
  
);

**Команди налаштування первинних та зовнішніх ключів:**

alter table student  
 add constraint pk\_student primary key (student\_id);

alter table gradebook  
 add constraint pk\_gradebook primary key (gradebook\_id);

alter table subject  
 add constraint pk\_subject primary key (subject\_id);

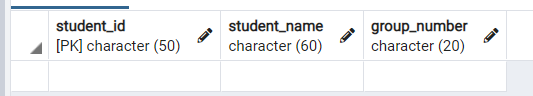
alter table relation\_gradebook\_subject  
 add constraint pk\_relation\_gradebook\_subject primary key (relation\_id);

alter table relation\_gradebook\_subject  
 add constraint fk\_relation\_gradebook foreign key (subject\_id) references subject (subject\_id);  
  
alter table relation\_gradebook\_subject  
 add constraint fk\_relation\_subject foreign key (gradebook\_id) references gradebook (gradebook\_id);

alter table gradebook  
 add constraint fk\_gradebook\_student foreign key (student\_id) references student (student\_id);

Перевірка:

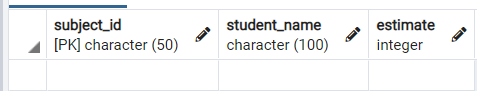
select \*  
from student;



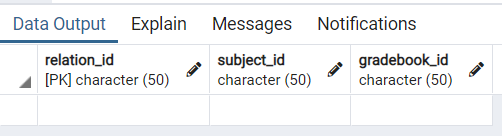
select \*  
from gradebook;



select \*  
from subject;



select \*  
from relation\_gradebook\_subject;



СПИСОК ЛІТЕРАТУРИ

1. Дейт К. Введение в системы баз данных. – Пер. с англ. – 8-е изд. – К.: Изд. дом «Вильямс», 2006. – 1326 с.

5. Форта Б. Освой самостоятельно SQL. 3-е изд.: Пер. с.англ. –М.: Изд. дом «Вильямс», 2006. – 288 с