

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:

студентка групи КМ-91

Павловська К.І.

Керівник:

ст. викладач Бай Ю.П.

ЗМІСТ

ЗАВДАННЯ	3
Завдання 1	4
Завдання 2	7
Завдання 3	8
СПИСОК ЛІТЕРАТУРИ	12

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:

	min_item_price	
	numeric	
1	2.49	

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:

	vend_name	
	character (50)	
1	Doll House Inc.	

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:

	prod_id [PK] character (10)	prod_name character (255)	length_prod_name integer
1	BR01	8 inch teddy bear ...	17
2	BR02	12 inch teddy bear ...	18
3	BR03	18 inch teddy bear ...	18
4	BNBG01	Fish bean bag toy ...	17
5	BNBG02	Bird bean bag toy ...	17
6	BNBG03	Rabbit bean bag toy ...	19
7	RGAN01	Raggedy Ann ...	11
8	RYL01	King doll ...	9
9	RYL02	Queen doll ...	10

Визначимо максимальну довжину назви товару:

```
select max(length(prod_name)) as length_prod_name_max
from products
```

Результат виконання:

	length_prod_name_max integer
1	19

Використаємо одержану інформацію як вкладений запит:

```
select prod_id
from products
where length(prod_name) = (select max(length(prod_name)) from products)
```

Результат виконання:

	prod_id [PK] character (10)
1	BNBG03

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:

	prod_id character (10)
1	RYL01
2	RYL02

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:

	vendor_name text
1	fun and games

Завдання 2

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

1a)

Введемо допоміжне позначення максимальної ціни проданого товару:

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

Визначимо, як звуть постачальника, що продав найдорожчий товар:

$$\pi_{vend_name}(\sigma_{item_price=Mp}(orderitems)).$$

1b)

Позначимо кількість літер у найдовшій назві товару як:

$$ML = MAX\left(\pi_{LENGTH(prod_name)}(products)\right);$$

Визначимо prod_id товару з найдовшою назвою:

$$\pi_{prod_id}\left(\sigma_{LENGTH(prod_name)=ML}(products)\right).$$

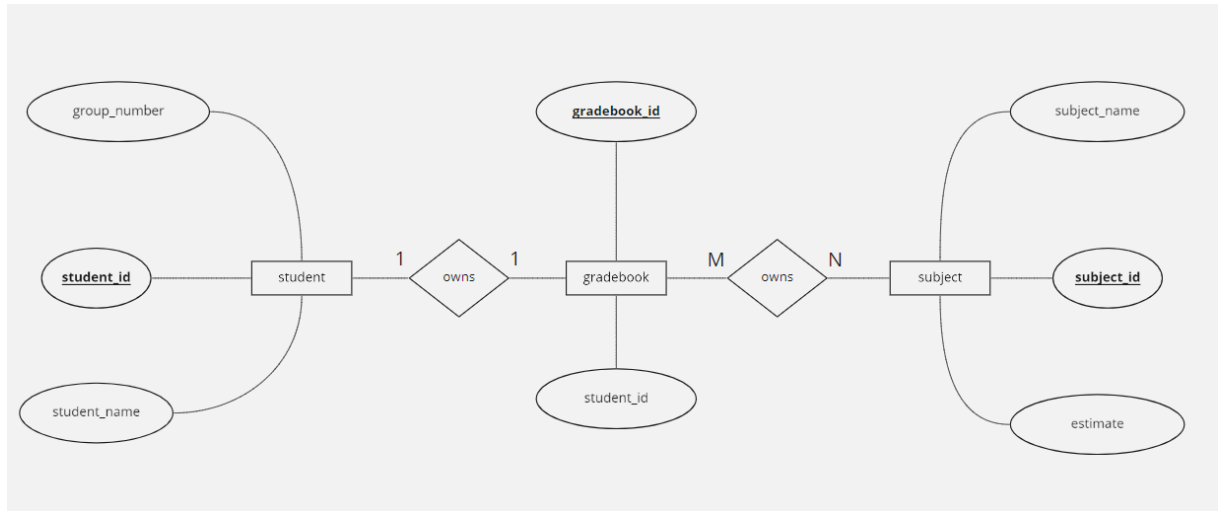
Завдання 3

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

Студент має заліковку що містить записи про дисципліни.

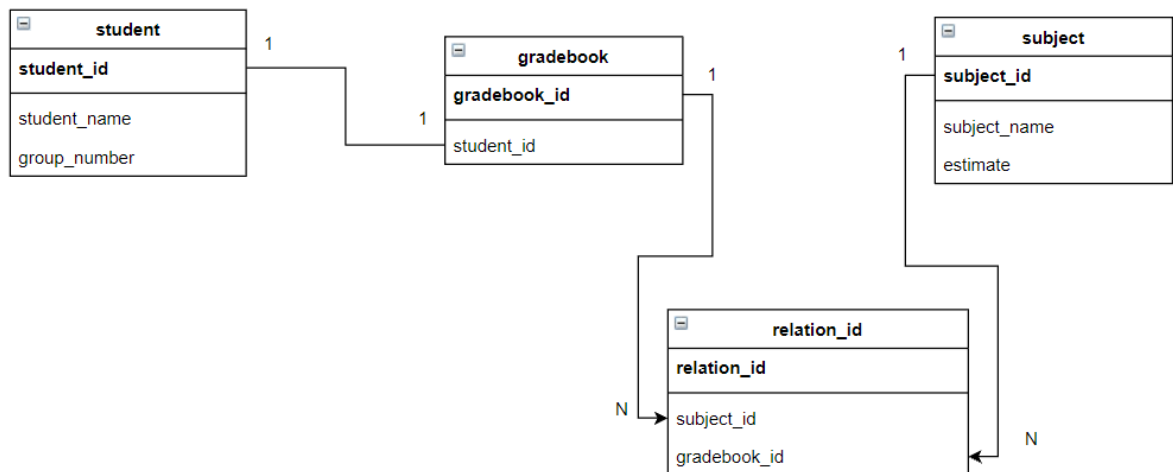
Визначити поля та типи. Головні та зовнішні ключі створювати окремо від таблиць, використовуючи команду ALTER TABLE.

ER-схема:



Реляційна модель:

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;
```

	student_id [PK] character (50)	student_name character (60)	group_number character (20)

```
select *
from gradebook;
```

	gradebook_id [PK] character (50)	student_id character (50)

```
select *
from subject;
```

	subject_id [PK] character (50)	student_name character (100)	estimate integer

```
select *
from relation_gradebook subject;
```

[Data Output](#) [Explain](#) [Messages](#) [Notifications](#)

	relation_id [PK] character (50)	subject_id character (50)	gradebook_id character (50)

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