

Додаток А  
(Рекомендований)

КОПІЇ ПРЕЗЕНТАЦІЇ

## КВАЛІФІКАЦІЙНА РОБОТА

Проектування та оптимізація бази даних для ефективного зберігання та управління інформацією про студентів з використанням PostgreSQL

Виконав: Перцевий М. А.

Керівник: ст.викл. Ганшин Д.Г.

## ВСТУП

Сучасний світ вимагає від інформаційних систем великої ефективності та швидкості обробки даних. Особливо важливою є ця проблематика у сферах, де великі обсяги даних обробляються одночасно та потрібно забезпечити їхню надійність та доступність. Одним з ключових інструментів для розв'язання цих завдань є системи управління базами даних (СУБД).

Метою даної дипломної роботи є дослідження та розробка бази даних для ведення обліку студентів, груп та дисциплін на кожному курсі університету. Враховуючи потреби сучасної освітньої системи, де необхідно забезпечувати ефективне управління та аналіз даних про навчальний процес, ця робота є актуальною та важливою.

## Види та сучасні системи керування базами даних

- файл-серверні СУБД;
- клієнт-серверні СУБД;
- вбудовувані СУБД.

У файл-серверних СУБД (Microsoft Access, Paradox, FoxPro, dBase тощо) для застосунків відкрито спільний доступ до всіх файлів бази даних (які зазвичай зберігаються в якомусь файловому сховищі, що розділяється), і вони можуть спільно обробляти ці дані. Кожен додаток обробляє дані самостійно. На сьогодні файлсерверна технологія вважається застарілою.

У клієнт-серверних СУБД (InterBase, Microsoft SQL Server, Firebird, Oracle, MySQL, PostgreSQL та ін.) уся обробка даних виконується в одному місці - на сервері, там же, де зазвичай зберігаються дані, при цьому доступ до файлів даних має тільки один сервер.

## Характеристика реляційних СУБД

У реляційному підході організації СУБД передбачається наявність набору відношень (двовимірних таблиць), які є пов'язаними між собою. Зв'язок у цьому випадку - це асоціювання двох або більше відносин (таблиць). База даних, що має обмежену структуру і не має зв'язків між таблицями, не може називатися реляційною.

Реляційний підхід у побудові СУБД має низку переваг:

- невеликий набір абстракцій, які дають змогу легко моделювати більшу частину поширених предметних областей і допускають точні формальні визначення, залишаючись зрозумілими;
- наявність простого і водночас потужного математичного апарату, що спирається головним чином на теорію множин і математичну логіку і забезпечує теоретичний базис реляційного підходу до організації баз даних реляційного підходу до організації баз даних;
- можливість ненавігаційного маніпулювання даними без необхідності знання конкретної фізичної організації баз даних у зовнішній пам'яті.

## Моделі даних

Моделлю даних називають формалізований опис структури одиниць інформації та операцій над ними в інформаційній системі.

Модель даних - це деяка абстракція, у якій відображаються найважливіші аспекти функціонування виділеної предметної області, а другорядні - ігноруються. Модель даних містить у собі набір понять для опису даних, зв'язків між ними та обмежень, що накладаються на дані.

Однією з основних переваг реляційної моделі є її однорідність. Усі дані розглядаються як такі, що зберігаються в таблицях і тільки в таблицях. Кожен рядок такої таблиці має один і той самий формат.

Реляційна база даних - це кінцевий (обмежений) набір відносин. Відносини використовуються для представлення сутностей, а також для представлення зв'язків між сутностями.

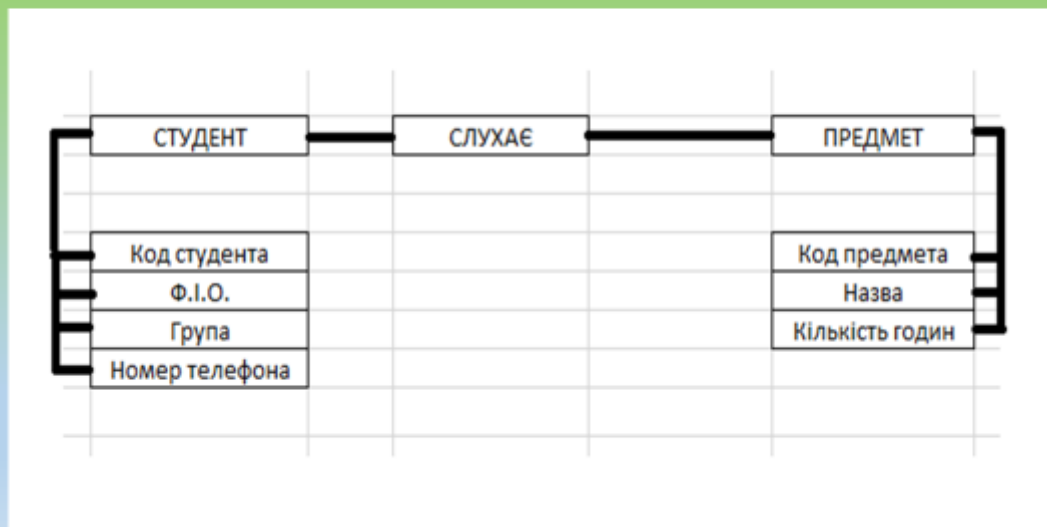
## Характеристики відносин

СТУДЕНТ			
Р#	Ф.І.О.	Група	Номер телефона
P1	Шевченко Т.Г.	ІТІР-20-1	95546525
P2	Сковорода Г.С.	ІТІР-20-1	66789635
P3	Косач Л.П.	ІТІР-20-1	93587452

Діаграма показує таблицю "СТУДЕНТ" з атрибутами: Р#, Ф.І.О., Група, Номер телефона. Таблиця має первинний ключ (Р#), відношення (Р1, Р2, Р3) та атрибути (Ф.І.О., Група, Номер телефона). Також вказано ім'я стосунків (СТУДЕНТ), домену (Номер телефона) та кортежі (Рядки).

- відношення має ім'я, яке відрізняється від імен усіх інших відносин;
- відношення подається у вигляді табличної структури;
- кожен атрибут має унікальне ім'я, його значення беруться з одного й того самого домену;
- кожен компонент кортежу є простим, атомарним значенням, що не складається з групи значень;
- упорядкування атрибутів теоретично несуттєве, однак воно може впливати на ефективність доступу до кортежів;
- усі рядки (кортежі) мають бути різними;
- теоретично порядок слідування кортежів не має значення.

## Концептуальна модель



## Реляційні відносини моделі

Код std.	Ф.І.О.	Група	Номер тел.			Код предмета	Код std.
1	Шевченко	ІТІР-20-1	56204786			1	2
2	Гусін	ІТІР-20-1	65489635			2	2
3	Ребров	ІТІР-20-1	49625873			3	4
4	Зінченко	ІТІР-20-1	45385453			4	1
						5	3
						6	1

Код предмета	Назва	Кількість годин
1	Інформатика	20
2	Бази даних	24
3	ООП	18
4	С++	40
5	ІоТ	55
6	Схемотехніка	25

## Створення БД в PostgreSQL

```

Query  Query History
1  CREATE TABLE перший_курс (
2      id SERIAL PRIMARY KEY,
3      прізвище_студента VARCHAR(255) NOT NULL,
4      номер_групи VARCHAR(255) NOT NULL,
5      номер_телефону VARCHAR(255) NOT NULL
6  );
7  INSERT INTO перший_курс (прізвище_студента, номер_групи, номер_телефону)
8  VALUES ('Гаврилов', 'ITIP-23-1', '062658971'),
9          ('Слюсаренко', 'ITIP-23-1', '098652876'),
10         ('Іванова', 'ITIP-23-1', '279563528');
11

```

## Таблиця для зберігання відношення багато до багатьох

```

-- Створення таблиці для зберігання відношення багато до багатьох між
CREATE TABLE студенти_дисципліни (
    id SERIAL PRIMARY KEY,
    id_студента INTEGER REFERENCES перший_курс(id),
    id_дисципліни INTEGER REFERENCES дисципліни(id)
);

-- Додавання зв'язків між студентами та дисциплінами
INSERT INTO студенти_дисципліни (id_студента, id_дисципліни)
VALUES (1, 1), (1, 4), -- Гаврилов вивчає Бази даних та Операційні системи
       (2, 2), (2, 1), -- Слюсаренко вивчає Алгоритми та структури даних та
       (3, 5), (3, 2); -- Іванова вивчає Web-програмування та Алгоритми та ст

```

## Управління БД Отримання списку студентів та їх дисциплін

Query Query History

```

1 SELECT перший_курс.прізвище_студента, дисципліни.назва_дисципліни
2 FROM студенти_дисципліни
3 INNER JOIN перший_курс ON студенти_дисципліни.id_студента = перший_курс.id
4 INNER JOIN дисципліни ON студенти_дисципліни.id_дисципліни = дисципліни.id;
5 |

```

Data Output Messages Notifications

	прізвище_студента character varying (255)	назва_дисципліни character varying (255)
1	Гаврилов	Бази даних
2	Гаврилов	Операційні системи
3	Слюсаренко	Алгоритми та структури даних
4	Слюсаренко	Бази даних
5	Іванова	Web-програмування
6	Іванова	Алгоритми та структури даних

✓ Successfully run. Total query r

## UPDATE

Query Query History

```

1 UPDATE перший_курс
2 SET номер_телефону = '052986257'
3 WHERE id = 1; -- Оновлюємо номер телефону для студента з id=1
4

```

Data Output Messages Notifications

UPDATE 1

Query returned successfully in 85 msec.

## Додавання нових записів



```
Query  Query History
```

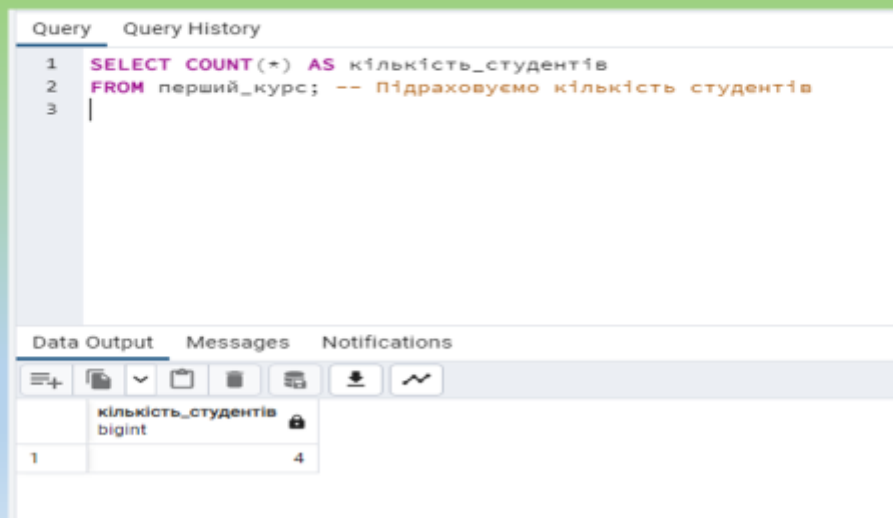
```
1 INSERT INTO перший_курс (прізвище_студента, номер_групи, номер_телефону)
2 VALUES ('Перцевий', 'ITIP-23-1', '035689541'); -- Додаємо нового студента
3
```

Data Output Messages Notifications

```
INSERT 0 1
```

Query returned successfully in 74 msec.

## Агрегація даних



```
Query  Query History
```

```
1 SELECT COUNT(*) AS кількість_студентів
2 FROM перший_курс; -- Підраховуємо кількість студентів
3
```

Data Output Messages Notifications

	кількість_студентів	
1	bigint	4

## ВИСНОВКИ

У ході роботи була розроблена база даних для ведення обліку студентів, груп та дисциплін на кожному курсі університету. Ця база даних є актуальною та важливою для сучасної освітньої системи, де необхідно забезпечувати ефективне управління та аналіз даних про навчальний процес.

Впровадження такої бази даних дозволить покращити організацію навчального процесу, забезпечити зручний доступ до інформації для викладачів та адміністрації навчального закладу, а також підвищить якість управління та контролю над навчальним процесом.

Додаток Б  
(Обов'язковий)

ВІДОМОСТІ АТЕСТАЦІЙНОГО ПРОЕКТУ

