



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»
ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування та спеціалізованих комп'ютерних
систем

Лабораторна робота №1

з дисципліни
«Бази даних і засоби управління»

Тема: **«Проектування бази даних та ознайомлення з базовими операціями
СУБД PostgreSQL»**

Виконав: студент 3 курсу
ФПМ групи КВ-94
Гераймович Д. Ю.
Перевірів:

Лабораторна робота №1.

Метою роботи є здобуття вмінь проектування бази даних та практичних навичок створення реляційних баз даних за допомогою PostgreSQL.

Завдання роботи полягає у наступному:

1. Розробити модель «сутність-зв'язок» предметної галузі, обраної студентом самостійно, відповідно до пункту «Вимоги до ER-моделі».
2. Перетворити розроблену модель у схему бази даних (таблиці) PostgreSQL.
3. Виконати нормалізацію схеми бази даних до третьої нормальної форми (3НФ).
4. Ознайомитись із інструментарієм PostgreSQL та pgAdmin 4 та внести декілька рядків даних у кожен з таблиць засобами pgAdmin 4.

Вимоги до ER-моделі

1. Сутності моделі предметної галузі мають містити зв'язки типу 1:N або N:M.
2. Кількість сутностей у моделі – 3-4. Кількість атрибутів у кожній сутності: від двох до п'яти.
3. Передбачити наявність зв'язку з атрибутом.
4. Для побудови ER-діаграм використовувати одну із нотацій: Чена, "Пташиної лапки (Crow's foot)", UML.

Додаток А. Концептуальна модель

В концептуальній моделі предметної області "Сервіс `питання-відповіді`" (Рисунок 1) виділяються наступні сутності та зв'язки між ними.

Сутність "Користувачі" з атрибутами: ім'я, дата реєстрації, рейтинг.

Сутність "Питання" з атрибутами: заголовок, текст, дата створення;

Сутність "Відповіді" з атрибутами заголовок, текст, дата створення;

Сутність "Теги" з атрибутами: назва, опис;

Між сутностями "Користувач" та "Питання" зв'язок R(1:N), тому що одне питання може бути створене лише одним користувачем, але один користувач може створити багато питань.

Між сутностями “Користувач” та "Відповіді" зв’язок R(1:N), тому що одну відповідь може створити лише один користувач, але один користувач може створити багато відповідей.

Між сутностями “Питання” та "Відповіді" зв’язок R(1:N), тому що відповідь може бути лише до одного питання, але питання може містити багато відповідей.

Між сутностями “Питання” та "Теги" зв’язок R(N:M), тому що питання може мати багато тегів, і один тег може бути присвоєний декільком питанням.

Між сутностями Users та Google Profile є зв’язок R(1:1), тому що один профіль Google може бути зв’язаний лише з одним локальним профілем і навпаки.

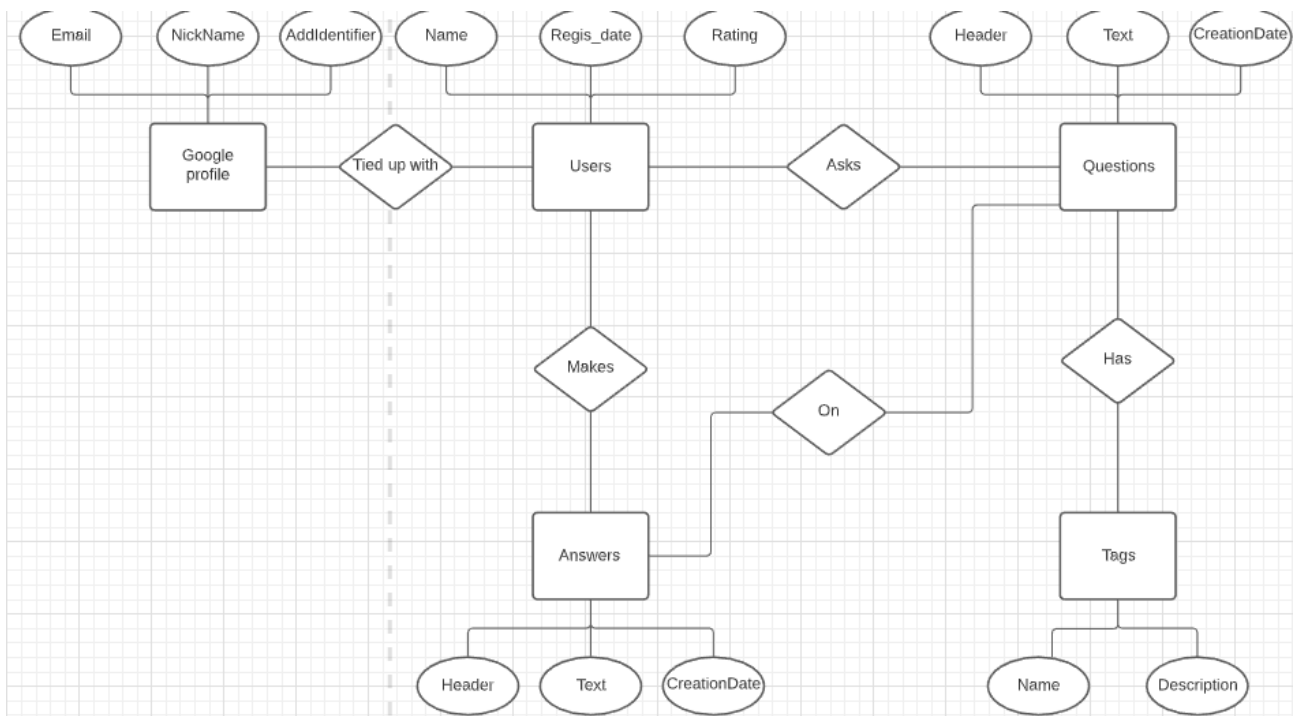


Рисунок 1 - Концептуальна модель

Додаток Б. Логічна модель (схема) БД

В логічній моделі (Рисунок 2):

Сутність "Користувачі" перетворена в таблицю "User".

Сутність "Питання" була перетворена в таблицю "Question".

Сутність "Відповіді" була перетворена в таблицю "Answer".

Сутність "Теги" була перетворена в таблицю "Tags".

Сутність "Google Profile" була перетворена в таблицю "Google Profile".

Зв'язок між тегами та питаннями привів до створення таблиці "QuestionTagsBindings".

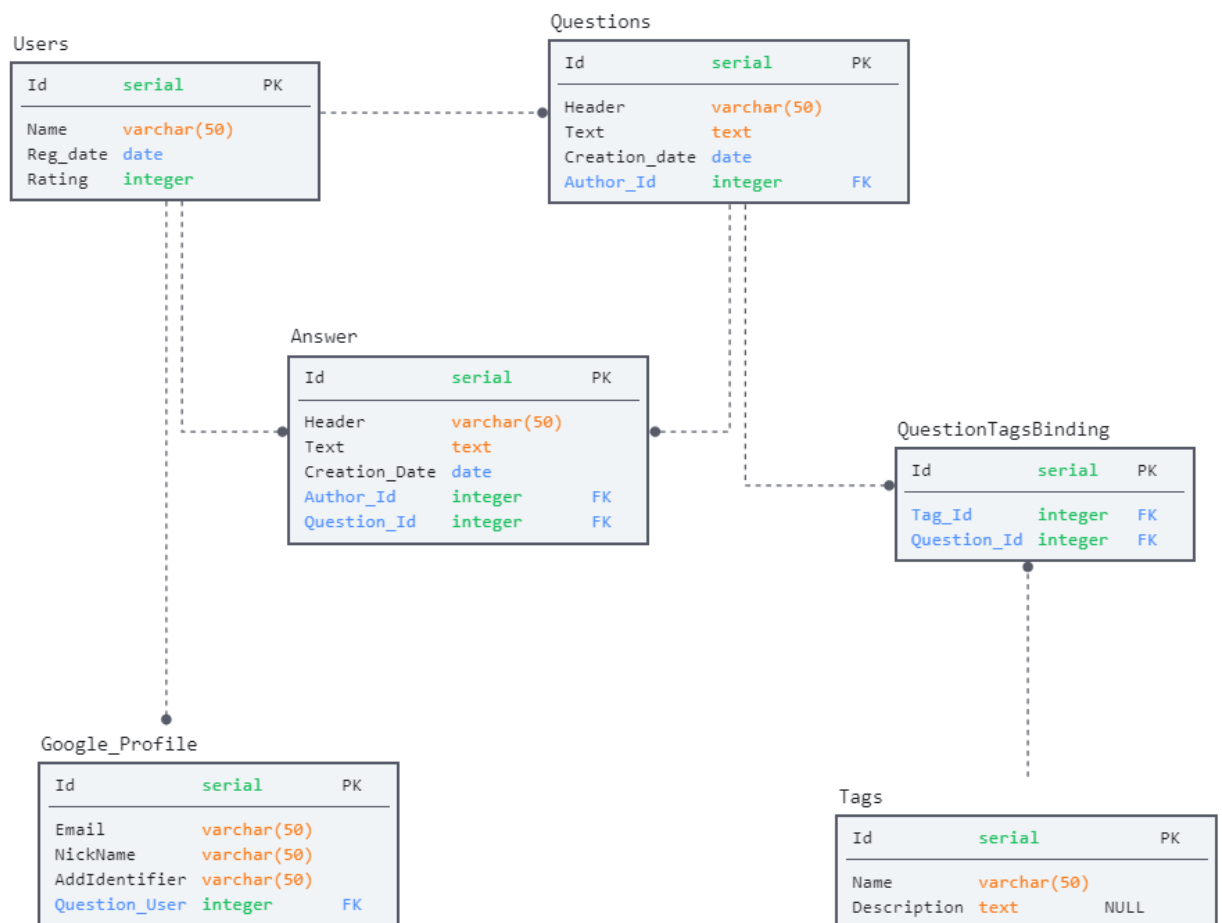


Рисунок 2 - Логічна модель

Схема бази даних відповідає 1НФ тому, що всі рядки унікальні, всі атрибути прості і не мають нереляційних структур (масивів, об'єктів і тд).

Схема бази даних відповідає 2НФ тому, що всі таблиці мають первинні ключі, всі атрибути яких залежать лише від цього ключа.

Схема бази даних відповідає 3НФ тому, що у базі даних нема неключових транзитивних залежностей.

Додаток В. Структура БД

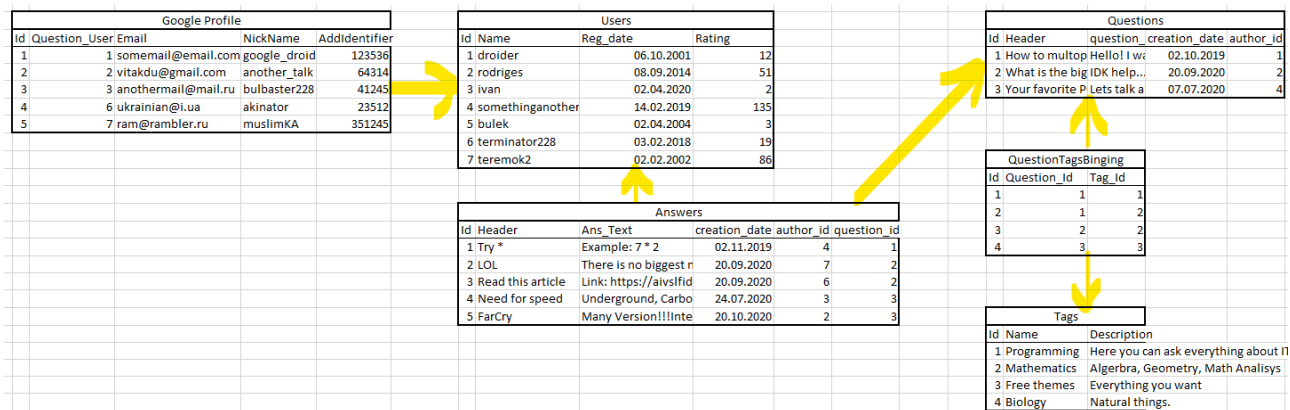
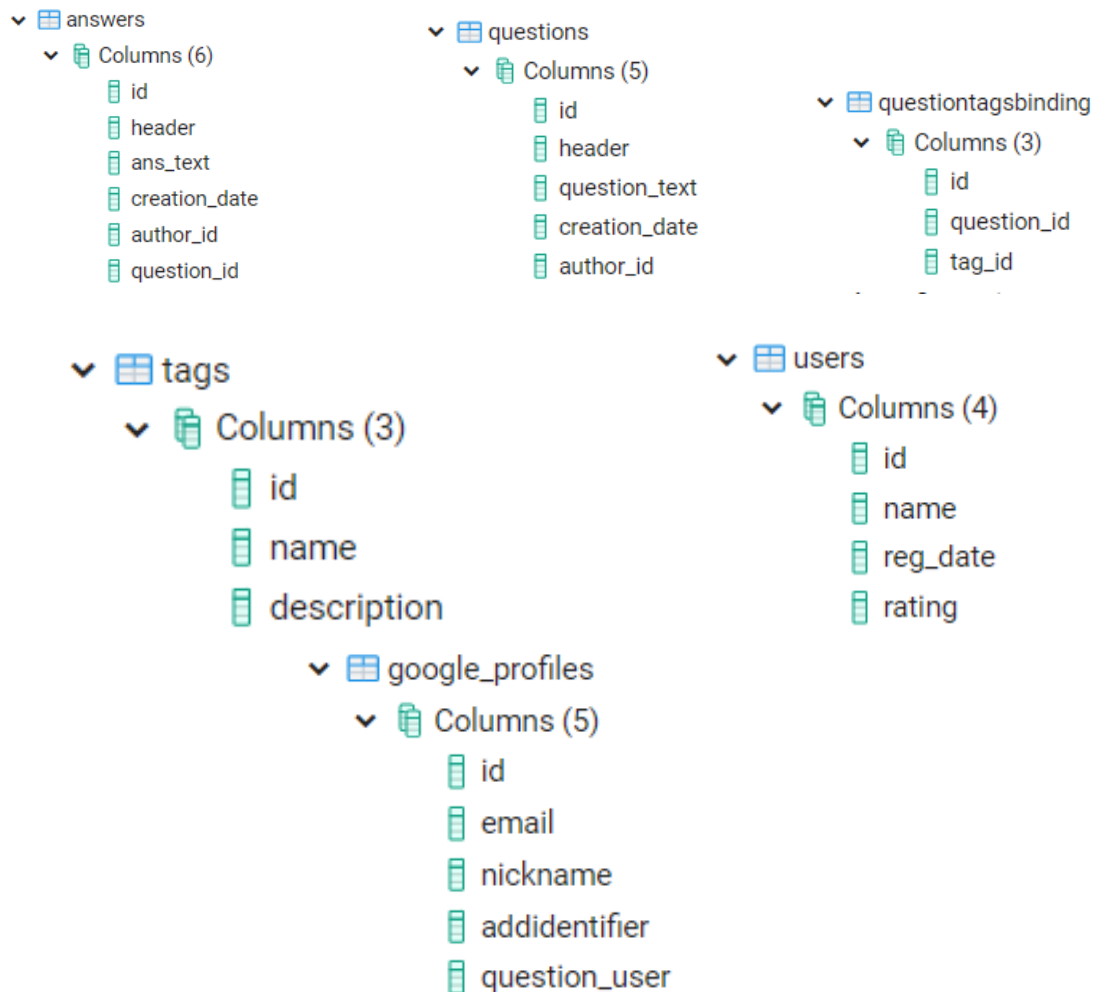


Рисунок 3 - Структурна модель

Додаток Г. Опис структури БД
Текстове представлення логічної моделі (схеми) БД

Відношення	Атрибут	Тип
Відношення "Users" містить інформацію про користувачів сервісу.	Id – унікальний номер користувача в БД Name – ім'я користувача. Не допускає NULL. Reg_date – дата реєстрації. Не допускає NULL. Rating – рейтинг користувача. Не допускає NULL.	Числовий, SERIAL PK Текстовий(50) Дата DATE Числовий
Відношення "Answers" містить інформацію про відповіді в сервісі.	Id – унікальний номер відповіді в БД. Header – заголовок відповіді. Не допускає NULL. Text – текст відповіді. Не допускає NULL. Creation_date – дата створення. Не допускає NULL. Author_Id – унікальний номер автора запитання. Не допускає NULL.	Числовий, SERIAL PK Текстовий(50) Текстовий Дата DATE Числовий FK
Відношення "Questions" містить інформацію про питання в сервісі.	Id – унікальний номер відповіді в БД. Header – заголовок питання. Не допускає NULL. Text – текст питання. Не допускає NULL. Creation_date – дата створення. Не допускає NULL. Author_Id – унікальний номер автора запитання. Не допускає NULL. Question_Id – унікальний номер запитання. Не допускає NULL.	Числовий, SERIAL PK Текстовий(50) Текстовий Дата DATE Числовий FK Числовий FK
Відношення "Tags" містить інформацію про теги, які використовують питання сервісу.	Id - унікальний номер тега в БД Name – Назва тегу. Не допускає NULL. Description – Опис тега. Не допускає NULL.	Числовий, SERIAL PK Текстовий(50) Текстовий
Відношення "QuestionTagsBinding" містить інформацію про зв'язок питань та тегів.	Id – унікальний номер зв'язка. Tag_id – унікальний номер тега. Не допускає NULL. Question_Id – унікальний номер питання. Не допускає NULL.	Числовий SERIAL PK Числовий FK Числовий FK
Відношення "Google_Profile" містить інформацію про профілі гугл користувачів сайту а також їх ідентифікатор для реклами.	Id – унікальний номер профілю. Email – електронна пошта користувача. Не допускає NULL і повторень. NickName – нікнейм користувача для сервісів Google. Не допускає NULL і повторень. AddIdentifier – унікальний ідентифікатор користувача для реклами. Не допускає NULL. Question_User – унікальний номер користувача в БД.	Числовий SERIAL PK Текстовий (50) Текстовий(50) Числовий Числовий FK

Додаток Г. Структура БД в pgAdmin 4



Опис таблиць БД в pgAdmin 4




Таблиця “Users”

```
CREATE TABLE Users(  
    id serial NOT NULL,  
    name varchar(50) NOT NULL,  
    Reg_date Date NOT NULL,  
    Rating int NOT NULL,  
    CONSTRAINT users_pkey PRIMARY KEY (id)  
)
```

	id [PK] integer	name character varying (50)	reg_date date	rating integer
1	1	droider	2001-10-06	12
2	2	rodriques	2014-09-08	51
3	3	ivan	2020-04-02	2
4	4	somethinganother	2019-02-14	135
5	5	bulek	2004-04-02	3
6	6	terminator228	2018-02-03	19
7	7	teremok2	2002-02-02	86






Таблиця “Tags”

```
CREATE TABLE Tags(  
    id serial NOT NULL,  
    name varchar(50) NOT NULL,  
    description text not null,  
    CONSTRAINT tags_pkey PRIMARY KEY (id)  
)
```




	 id [PK] integer	 name character varying (50)	 description text
1	1	Programming	Here you can as...
2	2	Mathematics	Algebra, Geom...
3	3	Free themes	Everything you ...
4	4	Biology	Natural things.

Таблиця “Questions”

```
CREATE TABLE Questions(  
    id serial not null,  
    header varchar(50) NOT NULL,  
    question_text text not null,  
    creation_date date not null,  
    author_id integer not null,  
    CONSTRAINT questions_pkey PRIMARY KEY (id),  
    CONSTRAINT ques_author_id_FK FOREIGN KEY(author_id)  
        REFERENCES Users(id)  
)
```

	 id [PK] integer	 header character varying (50)	 question_text text	 creation_date date	 author_id integer
1	1	How to multply in c#	Hello! I want to mu...	2019-10-02	1
2	2	What is the biggest number?	IDK help...	2020-09-20	2
3	3	Your favorite PC game?	Lets talk about so...	2020-07-07	4

Таблиця “QuestionTagsBinding”

	 id [PK] integer	 question_id integer	 tag_id integer
1	1	1	1
2	2	1	2
3	3	2	2
4	4	3	3

```
CREATE TABLE QuestionTagsBinding(
    id serial NOT NULL,
    question_id integer not null,
    tag_id integer not null,
    CONSTRAINT binding_pkey PRIMARY KEY (id),
    CONSTRAINT bind_tag_id_FK FOREIGN KEY(tag_id)
        REFERENCES Tags(id),
    CONSTRAINT bind_question_id_FK FOREIGN KEY(question_id)
        REFERENCES Questions(id)
)
```

Таблиця “Answers”

```
CREATE TABLE Answers(
    id serial NOT NULL,
    header varchar(50) NOT NULL,
    ans_text text not null,
    creation_date date not null,
    author_id integer not null,
    question_id integer not null,
    CONSTRAINT answers_pkey PRIMARY KEY (id),
    CONSTRAINT ans_author_id_FK FOREIGN KEY(author_id)
        REFERENCES Users(id),
    CONSTRAINT ans_question_id_FK FOREIGN KEY(question_id)
        REFERENCES Questions(id)
)
```

	id [PK] integer	header character varying (50)	ans_text text	creation_date date	author_id integer	question_id integer
1	1	Try *	Example: 7 * 2	2019-11-02	4	1
2	2	LOL	There is no b...	2020-09-20	7	2
3	3	Read this article	Link: https://...	2020-09-20	6	2
4	4	Need for speed	Undergroun...	2020-07-24	3	3
5	5	FarCry	Many Versio...	2020-10-20	2	3

Таблица “Google_Profiles”

```
create table Google_Profiles(
    Id serial not null,
    Email varchar(50) not null,
    NickName varchar(50) not null,
    AddIdentifier varchar(50) not null,
    Question_User integer not null,
    Unique(Question_User),
    Unique(NickName),
    Unique(AddIdentifier),
    Unique(Email),
    Primary Key(Id),
    Foreign Key(Question_User) references Users(Id)
)
```

	id [PK] integer	email character varying (50)	nickname character varying (50)	addidentifier character varying (50)	question_user integer
1	1	somemail@email.com	google_droid	123536	1
2	2	vitakdu@gmail.com	another_talk	64314	2
3	3	anothermail@mail.ru	bulbaster228	41245	3
4	4	ukrainian@i.ua	akinator	23512	6
5	5	ram@rambler.ru	muslimKA	351245	7