



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: *«Створити БД "сервіс `питання-відповіді` " в СУБД PostgreSQL  
з допомогою конструктора PgAdmin 4»*

Виконав: студент 3 курсу

ФПМ групи КВ-83

Бусленко В.

Перевірів: Павловський В.І.

Київ – 2020

Лабораторна робота №1.  
**Створити БД "Сервіс `питання-відповіді`" в СУБД PostgreSQL  
з допомогою конструктора PgAdmin 4**

**Мета роботи:** створити БД Сервіс `питання-відповіді` та сформулювати обмеження цілісності на значення даних.

**Порядок виконання роботи**

1. Розробити концептуальну модель вибраного предметного середовища. Концептуальну модель предметного середовища Сервіс `питання-відповіді` наводиться в Додатку А до лабораторної роботи;
2. Розробити логічну модель (схему) даних БД. Логічна модель (схема) даних БД Сервіс `питання-відповіді` наводиться в Додатку ? до лабораторної роботи;
3. Вивчити склад та правила роботи з СУБД PostgreSQL;
4. Створити в СУБД PostgreSQL БД Сервіс `питання-відповіді` , використовуючи конструктори таблиць та стовпчиків (краще колонок). Схема даних БД Сервіс `питання-відповіді` наводиться в Додатку Б до лабораторної роботи. Перелік атрибутів наводиться в Додатку В до лабораторної роботи;
5. Сформулювати обмеження цілісності, що забезпечують:
  - унікальність та обов'язковість вводу первинних ключів для всіх таблиць;
  - перевірка на відповідність зовнішніх ключів таблиць;
  - обмеження на значення даних для атрибутів "expiredln", "name", "sex", "age", "surname" і вивід відповідних повідомлень при їх порушенні (опис всіх атрибутів наводиться нижче в Додатку В до лабораторної роботи);
  - обов'язковість вводу атрибутів "expiredln", "name", "sex", "age", "surname" та інших.
6. Фізична модель (схема) даних БД Сервіс `питання-відповіді` наводиться в Додатку ? до лабораторної роботи;
7. Заповнити створену БД даними (порядку 5-10 записів в кожній таблиці).

**Зміст звіту**

1. Склад СУБД PostgreSQL.
2. Концептуальна модель предметної області.
3. Логічна модель (схема) БД.
4. Склад обмежень цілісності в термінах СУБД PostgreSQL.
5. Фізична модель БД в термінах СУБД PostgreSQL.
6. Представлення БД в pgAdmin 4

## Додаток А. Концептуальна модель предметної області "Сервіс `питання-відповіді` "

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.

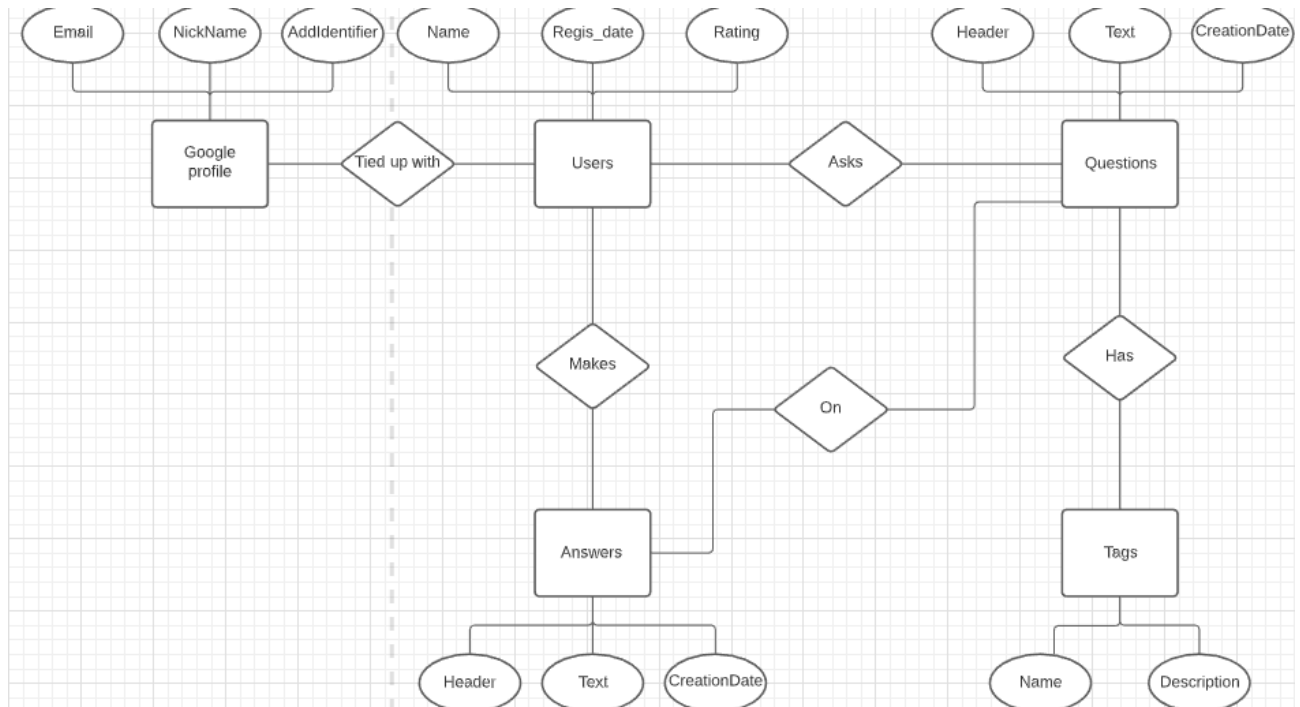


Рисунок 1 - Концептуальна модель предметної області "Сервіс `питання-відповіді` "

## Додаток Б. Логічна модель (схема) БД "Сервіс `питання-відповіді` "

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".

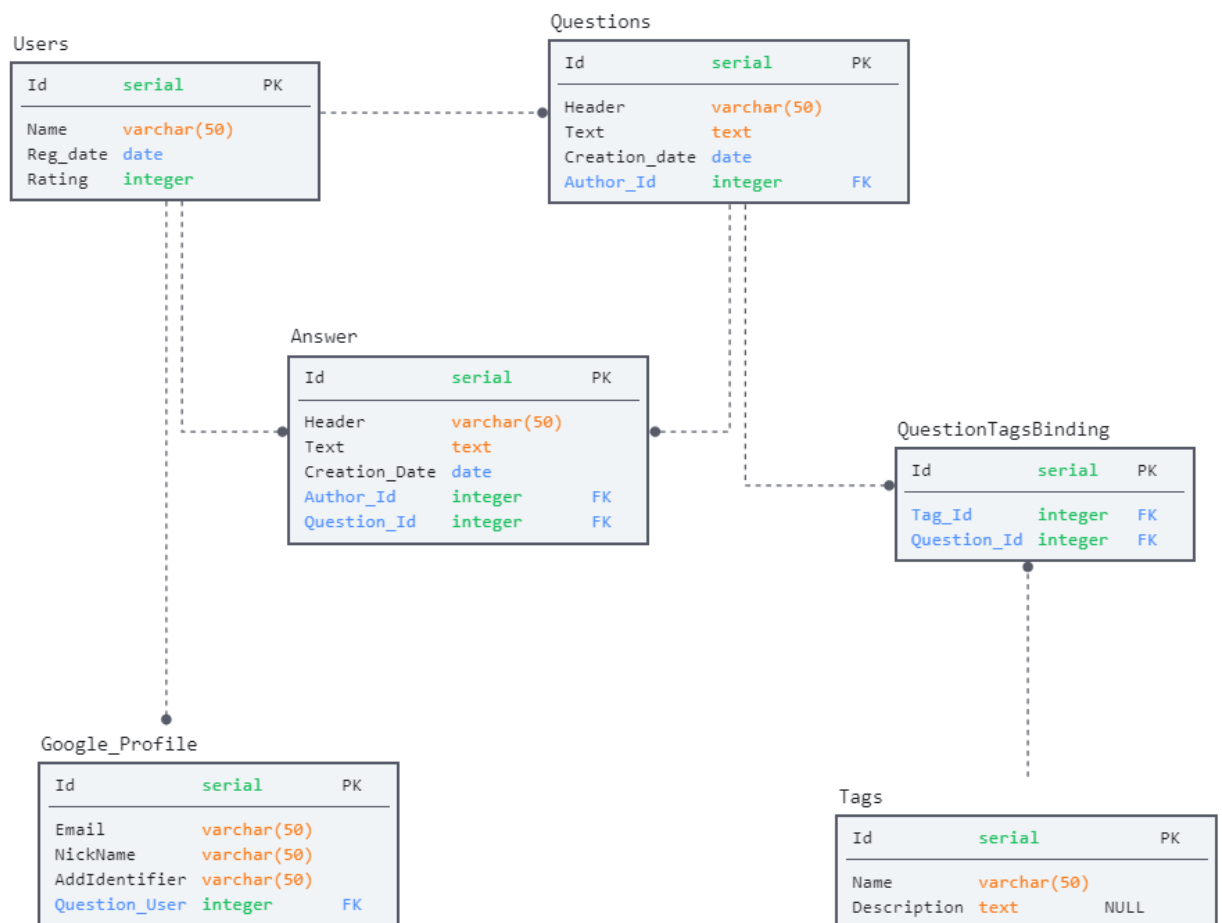


Рисунок 2 - Логічна модель предметної області "Сервіс `питання-відповіді` "

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 “Сервіс `питання-відповіді` ”

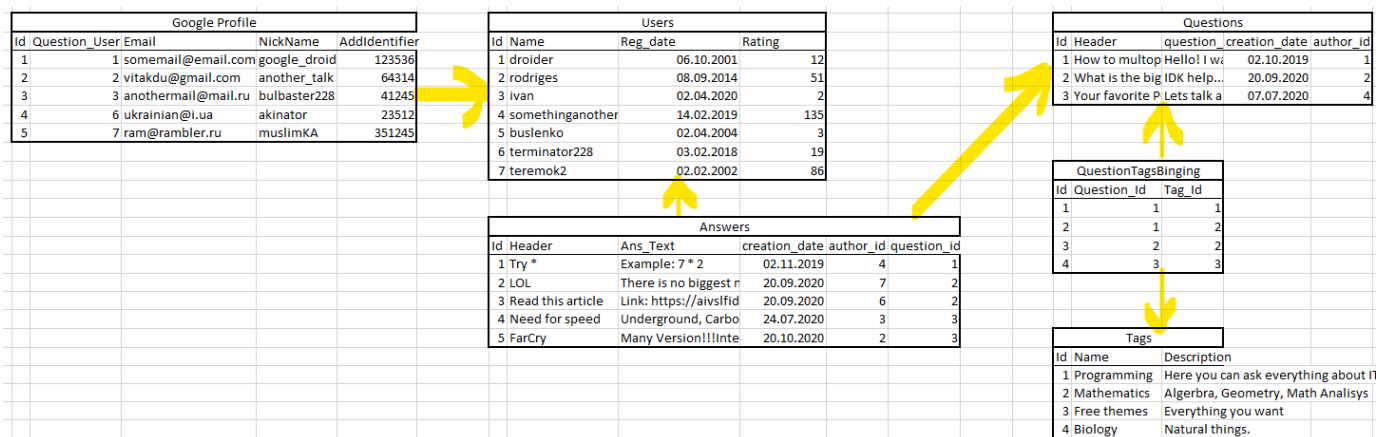
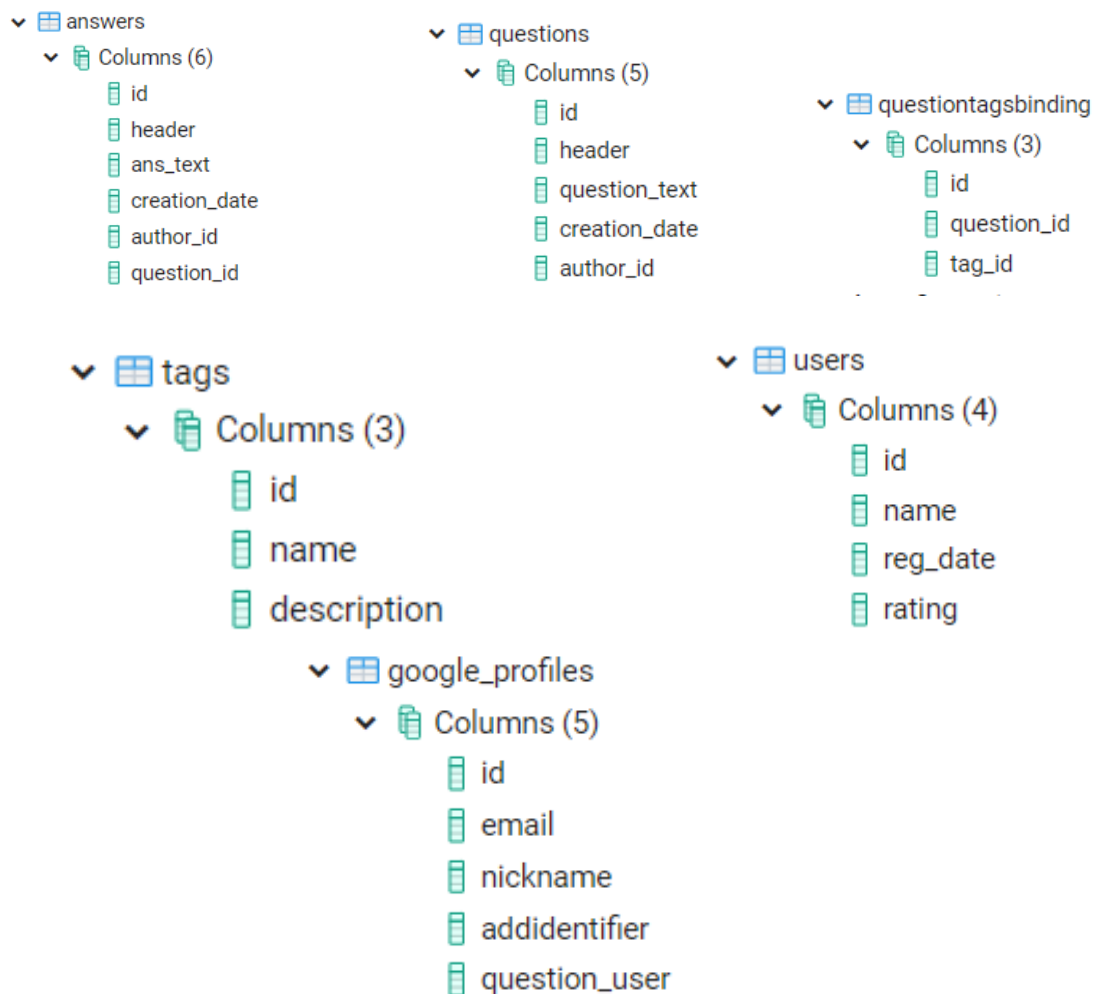


Рисунок 3 - Структурна модель предметної області "Сервіс `питання-відповіді`"

**Додаток Г. Опис структури БД "Сервіс `питання-відповіді` "**  
**Текстове представлення логічної моделі (схеми) БД**

<b>Відношення</b>	<b>Атрибут</b>	<b>Тип</b>
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 "Сервіс `питання-відповіді` " в pgAdmin 4



## Опис таблиць БД "Сервіс `питання-відповіді` " 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)  
)
```

		<b>id</b> [PK] integer 	<b>name</b> character varying (50) 	<b>reg_date</b> date 	<b>rating</b> integer 
1		1	droider	2001-10-06	12
2		2	rodrigues	2014-09-08	51
3		3	ivan	2020-04-02	2
4		4	somethinganother	2019-02-14	135
5		5	buslenko	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)
)
```

		<b>id</b> [PK] integer 	<b>name</b> character varying (50) 	<b>description</b> 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 multoply 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	

**Таблица “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)
)
```

**Таблица “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