НАЦIОНАЛЬНИЙ ТЕХНIЧНИЙ УНIВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛIТЕХНIЧНИЙ IНСТИТУТ IМЕНI IГОРЯ СIКОРСЬКОГО» ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

**Кафедра системного програмування і спеціалізованих комп'ютерних систем**

**Лабораторна робота №1**

з дисципліни «[Бази даних і засоби управління](http://wiki.kpi.ua/index.php/%D0%91%D0%B0%D0%B7%D0%B8%20%D0%B4%D0%B0%D0%BD%D0%B8%D1%85%20%D1%96%20%D0%B7%D0%B0%D1%81%D0%BE%D0%B1%D0%B8%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D1%96%D0%BD%D0%BD%D1%8F_(19301930))»

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

Виконав: студент ІІI курсу

ФПМ групи КВ-81

Поляков Є.А.

Викладач: [Петрашенко А. В.](http://rozklad.kpi.ua/Schedules/ViewSchedule.aspx?v=dc2ccd54-2021-4dbe-a8b3-d354f484e595)

Київ 2020

**Завдання:**

*У звіті щодо пункту №1 завдання має бути:*

* перелік сутностей з описом їх призначення;
* графічний файл розробленої моделі «сутність-зв’язок»;
* назва нотації.

*У звіті щодо пункту №2 завдання має бути:*

* опис процесу перетворення (наприклад, “сутність А було перетворено у таблицю А, а зв’язок R (M:N) зумовив появу додаткової таблиці R1 тощо);
* схему бази даних у графічному вигляді **з назвами таблиць (!) та зв’язками між ними.**

*У звіті щодо пункту №3 завдання має бути:*

* пояснення (**обґрунтування!**) щодо відповідності схеми бази даних нормальним формам НФ1, НФ2 та НФ3. Пояснення ***полягає у наведенні функціональних залежностей***, що демонструють висновки. У випадку невідповідності надати опис необхідних змін у схемі;
* У випадку проведення змін у схемі бази даних надати оновлену версію схеми, інакше - не наводити схему.

*У звіті щодо пункту №4 завдання має бути:*

* навести копії екрану з pgAdmin4, що відображають назви, типи та обмеження на стовпці (доступне у закладці “Columns” та “Constraints” властивостей “Properties” таблиць дерева об’єктів у pgAdmin4);
* навести копії екрану з pgAdmin4, що відображають вміст таблиць бази даних у PostgreSQL. Таблиці на зображенні обов'язково **повинні мати назву**!

**Пункт №1 завдання:**

Предметна галузь **блог (користувачі, дописи, коментарі, рейтинги)**

Опис обраної предметної галузі :

* Сутність користувачі (Users) містить у собі інформацію про користувачів.

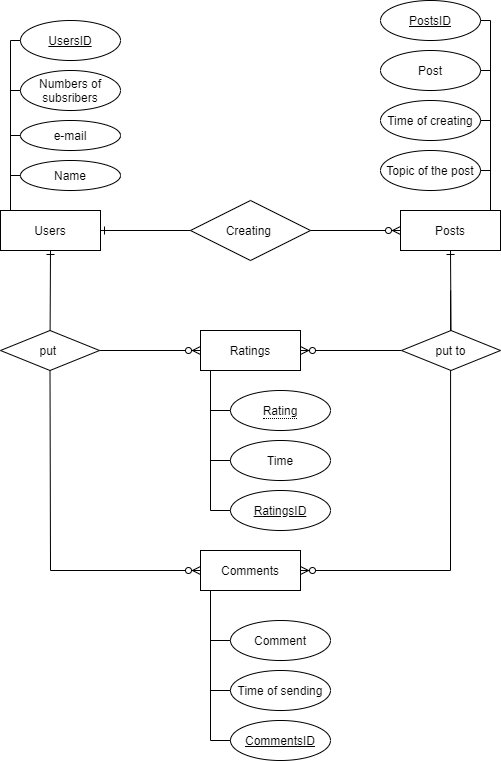
Містить поле Name (ім’я) , E-mail (електронна пошта), Numbers of subscribers (кількість підписників).

* Сутність дописи (Posts) містить у собі інформацію про допис, сам допис (текст допису) і посилання на користувача який створив пост.

Містить поле Topic of the post (Заголовок, тема допису), Post (містить у собі текст допису), Time of creating (час створення допису).

* Сутність коментарі (Comments) містить у собі інформацію про коментар, сам коментар, час створення, посилання на допис до якого написаний і користувача, що написав коментар. Містить поле Comment (текст коментаря), Time of sending (час надсилання).
* Сутність рейтинги (Ratings) містить у собі інформацію про рейтинг, користувача, що поставив рейтинг і посилання на допис до якого рейтинг поставлений.

Модель «сутність-зв’язок» (ER - модель):



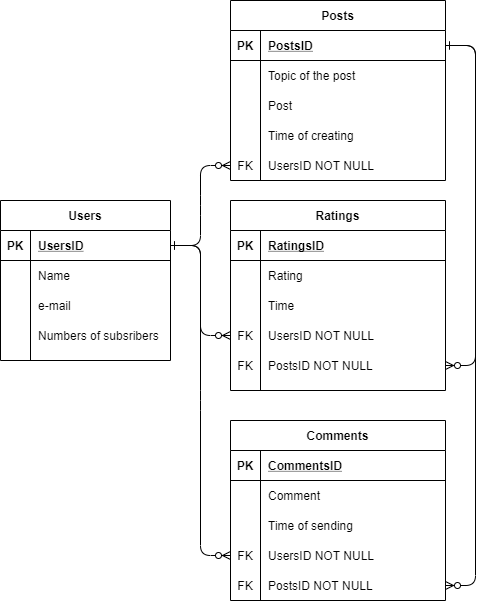
Сутність Users пов’язана з сутностями Posts, Comments і Ratings зв’язками 1:N, де N необов’язкові.

Сутність Posts пов’язана з Ratings і Comments зв’язками 1:N, де N необов’язкові.

При побудові моделі використовувались нотація “Пташиної лапки (Crow’s foot)”.

**Пункт №2 завдання:**

Даталогічна модель:



* сутність Users перетворена у таблицю Users;
* сутність Posts перетворена у таблицю Posts;
* сутність Ratings перетворена у таблицю Ratings;
* сутність Comments перетворена у таблицю Comments;
* зв’язки не призвели до створення додаткових таблиць.

Users <-> Posts (1:N), де N необов’язковий.

Users <-> Ratings (1:N), де N необов’язковий.

Users <-> Comments (1:N), де N необов’язковий.

Posts <-> Ratings (1:N), де N необов’язковий.

Posts <-> Comments (1:N), де N необов’язковий.

Для забезпечення зв’язку між таблицями використовуються зовнішні ключі:

* Posts і Users пов’язані за допомогою ключа FK UserID в Posts.
* Ratings пов'язаний з Users за допомогою ключа FK UserID, а з Posts ключем FK PostID.
* Comments пов'язаний з Users за допомогою ключа FK UserID, а з Posts ключем FK PostID.

**Пункт №3 завдання:**

Cхеми бази даних знаходяться у **1 НФ** (нормальній формі), так як кожна комірка в таблиці несе тільки одне (атомарне) значення.

В комірці стовпчика Name таблиці Users зберігається ім'я тільки одного користувача і т.д.

Cхеми бази даних знаходяться у **2 НФ** (нормальній формі), так як знаходяться у 1 НФ і у таблицях немає композитних первинних ключей, тому автоматично кожен атрибут в таблиці залежить тільки від одного ключа.

Таблиця **Users**:

UserID – Primary key (PK);

UserID -> Name

UserID ->E-mail

UserID -> Numbers of subscribers

Таблиця **Posts**:

PostID – Primary key (PK);

PostID -> Topic of the post

PostID ->Post

PostID -> Time of creating

Таблиця **Ratings**:

RatingID – Primary key (PK);

RatingID -> Rating

RatingID ->Time

Таблиця **Comments**:

CommentID – Primary key (PK);

CommentID -> Comment

CommentID ->Time of sending

Кожен не ключовий атрибут залежить тільки від одного первинного ключа.

Cхеми бази даних знаходяться у **3 НФ** (нормальній формі), так як знаходяться у 2 НФ і транзитивно не залежать один від одних.

У таблиці **Users** :

UserID -> Name (Unique)

UserID -> E-mail (Unique)

UserID ->Numbers of subscribers

Name ->E-mail

E -mail -> Name

UserID -> Name -> E-mail

UserID ->E-mail ->Name

Останні два зв’язки є транзитивними, але в них поля Name і E-mail унікальні, що запобігає повторенню інформації і всі поля напряму пов’язані з ключем UserID.

У таблиці **Post**:

PostID ->Topic of the post

PostID -> Post

PostID -> Time of creating

Атрибути не вказують на інші атрибути в таблиці. Знаючи назву допису (Topic if the post) не завжди можна дізнатися іншу інформацію, бо поле не є унікальним, але повторення цих полів не є дублювання інформації, бо повторювана інформація вважається закріпленою за різними користувачами. Звісно один і той самий користувач може називати дописи і писати в них одну і ту саму інформацію, але це не дублювання інформації базою даних, а дублювання інформації користувачем. Кожне поле має унікальне ID.

У таблиці **Ratings**:

RatingID ->Rating

RatingID -> Time of sending

Rating і Time of sending просто числові і часові значення які не надають ніякої конкретної інформації, тому пошук відбувається за допомогою RatingID, який є унікальним.

У таблиці **Comment**:

CommentID ->Comment

CommentID -> Time of sending

Так само як і рейтинг поля Comment і Time of sending не вказують ні на що конкретне. Пошук відбувається за допомогою унікального ключа CommentID.

Тому можна сказати, що умови для 3 НФ виконуються, так як транзитивність не виконується.

**Пункт №4 завдання:**

Таблиця Users:

Зображення, що містить знімок екрана

Автоматично згенерований опис

Зображення, що містить знімок екрана

Автоматично згенерований опис

Зображення, що містить знімок екрана, екран

Автоматично згенерований опис

Зображення, що містить знімок екрана, екран, дисплей

Автоматично згенерований опис

Таблиця Posts:

Зображення, що містить знімок екрана

Автоматично згенерований опис

Зображення, що містить знімок екрана

Автоматично згенерований опис

Зображення, що містить знімок екрана

Автоматично згенерований опис

Зображення, що містить знімок екрана, вулиця

Автоматично згенерований опис

Таблиця Ratings:

Зображення, що містить знімок екрана, ноутбук, комп’ютер

Автоматично згенерований опис

Зображення, що містить знімок екрана

Автоматично згенерований опис

Зображення, що містить знімок екрана

Автоматично згенерований опис

Зображення, що містить знімок екрана

Автоматично згенерований опис

Зображення, що містить знімок екрана, сидить, автобус, шафа

Автоматично згенерований опис

Таблиця Comments:

Зображення, що містить знімок екрана, комп’ютер, ноутбук

Автоматично згенерований опис

Зображення, що містить знімок екрана

Автоматично згенерований опис

Зображення, що містить знімок екрана

Автоматично згенерований опис

Зображення, що містить екран

Автоматично згенерований опис