



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»**

«Ознайомлення з базовими операціями СУБД PostgreSQL»

Виконав:
студент групи КВ-82
Кисельов Ростислав Юрійович

Перевірив:
Павловський Володимир Ілліч

Метою роботи є здобуття практичних навичок створення реляційних баз даних за допомогою PostgreSQL.

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 щодо відповідності схеми бази даних нормальним формам НФ1, НФ2 та НФ3. 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 та типи стовпців (доступне у закладці “Columns” 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!**

Опис предметної галузі

(Блог). При проектуванні даної галузі можна виділити такі сутності.

Блог (Blog) – призначений для ідентифікації блогу де будуть розмішуватись пости, назва блогу та його тема.

пост (post) – створений для ідентифікації поста, блогу, в якому він знаходиться та його наповнення.

рейтинг (rating) – призначений для ідентифікації рейтингу поста, його балів та оцінки цікавості для користувачів.

коментар (comment) – створений для ідентифікації коментаря, поста, до якого він прив'язаний, користувача, що написав його та вмісту.

користувач (user) – створений для ідентифікації користувача, його імені, дати реєстрації та електронної пошти.

В одному блозі може бути багато постів (1:N);

У кожного поста лише один рейтинг (1:1);

У кожного поста може бути багато коментарів або зовсім не бути їх (1:N);

Кожен користувач може додати декілька коментарів або не додавати їх

зовсім (1:N);

Багато користувачів можуть коментувати багато постів (M:N);

Концептуальна модель учбової предметної області " Блог "

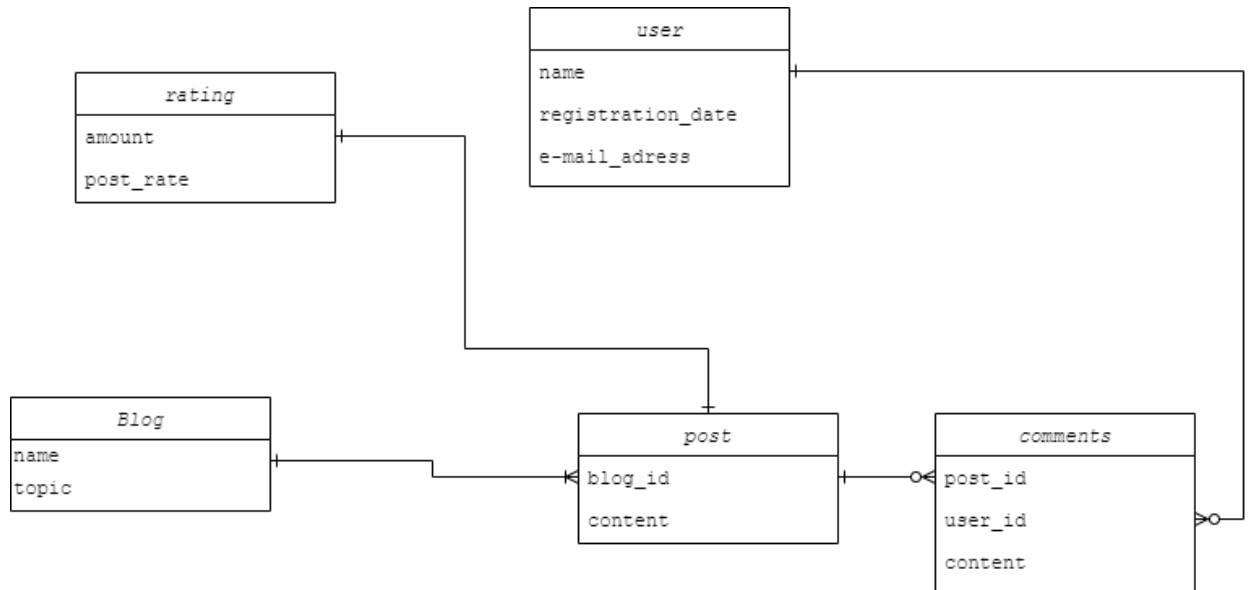


Рисунок 1 - Концептуальна модель предметної області " Блог "

Нотація: Пташина лапка + засоби програми draw.io

Пояснення щодо відповідності схеми бази даних нормальним формам:

Схема бази даних відповідає 1НФ тому, що схема передбачає 1 атомарне значення в кожній комірці.

Схема бази даних відповідає 2НФ тому, що, по-перше, відповідає 1НФ. По-друге, кожен атрибут в таблицях залежить від цілого ключа, а не від його частини.

Схема бази даних відповідає 3НФ тому, що, по-перше, вона відповідає 2НФ. По-друге, у ній відсутні транзитивні функціональні залежності неключових атрибутів від ключових.

Логічна модель (Структура) БД “ Блог ” (засобами SqlDMB)

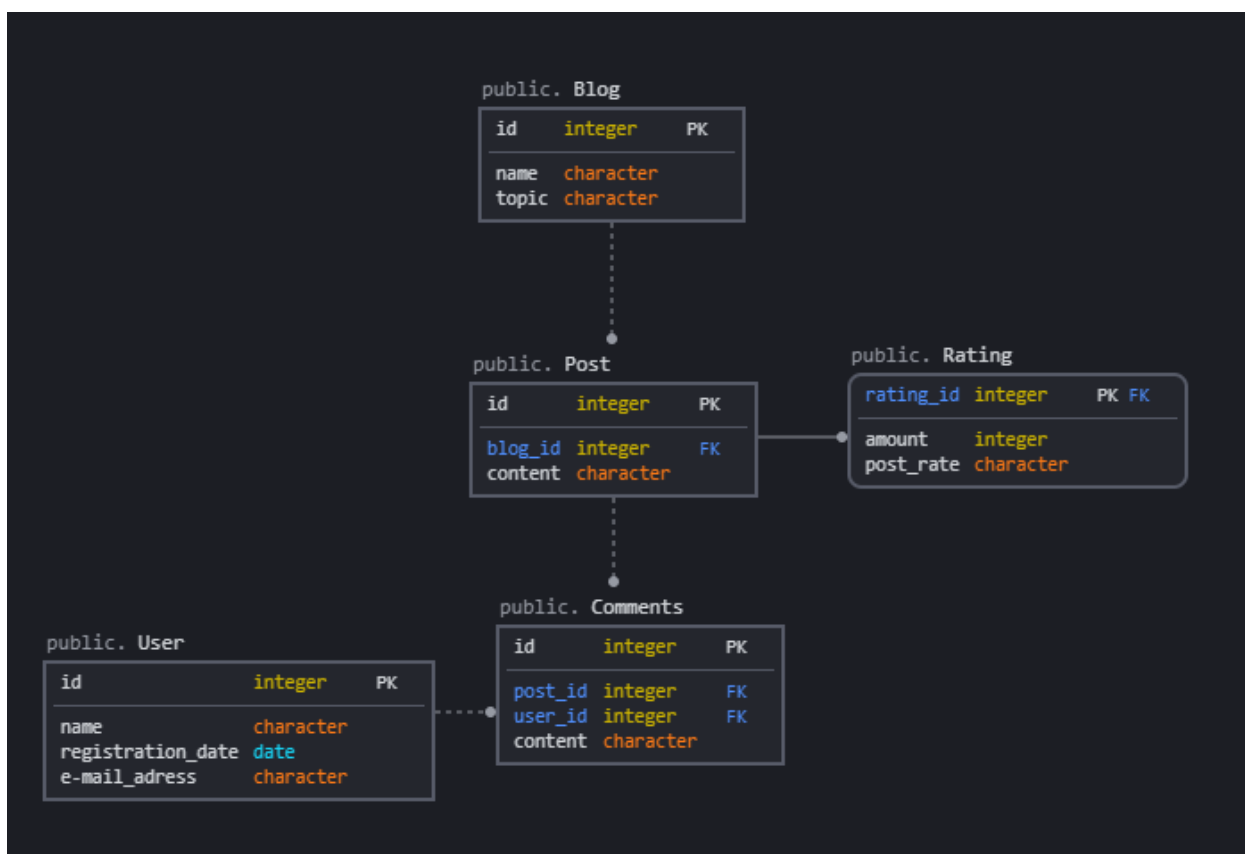


Рисунок 2 - Логічна модель (Структура) БД “ Блог ” (засобами SqlDMB)

Опис структури БД “ Блог ”

Таблиця 1 - Опис структури БД “ Блог ”

СУТНІСТЬ	АТРИБУТ	ТИП(Розмір)
Сутність “ <i>Blog</i> ” Вміщує інформацію про блог	id – унікальний ID блогу name – назва блогу topic – тема блогу	Числовий Текстовий(20) Текстовий(20)
Сутність “ <i>Post</i> ” Вміщує інформацію про пост	id - унікальний ID поста blog_id – ID блогу, в якому розміщено пост content – зміст поста	Числовий Числовий Текстовий(500)
Сутність “ <i>Rating</i> ” Вміщує інформацію про рейтинг поста	rating_id – унікальний ID, який вказує на пост, рейтинг якого розглядається amount – числовий рейтинг посту post_rate – рейтинг рівня викликаного інтересу у користувачів, що оцінили даний пост	Числовий Числовий Текстовий(20)
Сутність “ <i>Comments</i> ” Вміщує інформацію про коментарі до посту	id – унікальний ID коментаря post_id – ID посту, в якому розміщено коментар user_id – ID користувача, який розмістив коментар content – зміст коментарю	Числовий Числовий Числовий Текстовий(120)
Сутність “ <i>User</i> ” Вміщує інформацію про користувача	id – унікальний ID користувача name – ім’я користувача registration_date – дата реєстрації користувача e-mail_address – електронна адреса користувача	Числовий Текстовий(20) Дата Текстовий(40)

Додаток Б1. Структура БД " Блог "

User

id	name	registration_date	e-mail_adress
0	Victor	2007-10-13	victor@gmail.com
1	Varda	2017-12-23	varda2000@gmail.com
2	Oleksandr	2010-01-27	oleksandrKiselyov@gmail.com

Comments

id	post_id	user_id	content
0	2	1	Very interesting!!! Post more!
1	2	0	Very important topic! I want to know much more about it.
2	1	1	Healthy lifestyle is very important thing, I think..
3	1	2	I want to become stronger and faster, what can y recommend to me?

Post

id	blog_id	content
0	2	I like to watch Japanese cartoons, they are called anime.
1	2	Naruto is very cool manga-book
2	1	I do sports every morning, it helps me to keep myself in fit
3	1	Today i am going to play basketball with my friend till the night.

Rating

rating_id	amount	post_rate
0	10	very interesting
1	10	very interesting
2	9	interesting
3	8	popular

Blog

id	name	topic
1	Sport & Fitness	Healthy Lifestyle
2	Anime Amino	Anime

Рисунок 3 - Структура БД " Блог "

Структура БД “ Блог ” в pgAdmin IV

User					
Columns (4)					
id					
name					
registration_date					
e-mail_address					

	id	name	registration_date	e-mail_address
	[PK] integer	character varying (20)	date	character varying (40)
1		Victor	2007-10-13	victor@gmail.com
2		Varda	2017-12-23	varda2000@gmail.com
3		Oleksandr	2010-01-27	oleksandrKiselyov@gmail.com

Tables (5)					
Blog					
Comments					
Columns (4)					
id					
post_id					
user_id					
content					
Constraints					

	id	post_id	user_id	content
	[PK] integer	integer	integer	character varying (120)
1		0	2	Very interesting!!! Rost more!
2		1	2	Very important topic! I want to know much more about it.
3		2	1	Healthy lifestyle is very important thing, I think..
4		3	1	I want to become stronger and faster, what can y recommend to me?

Post					
Columns (3)					
id					
blog_id					
content					
Constraints					
Indexes					
RLS Policies					

	id	blog_id	content
	[PK] integer	integer	character varying (500)
1		0	I like to watch Japanese cartoons, they are called anime.
2		1	Naruto is very cool manga-book
3		2	I do sports every morning, it helps me to keep myself in fit
4		3	Today i am going to play basketball with my friend till the night.

Rating					
Columns (3)					
rating_id					
amount					
post_rate					
Constraints					
Indexes					
RLS Policies					

	rating_id	amount	post_rate
	[PK] integer	integer	character varying (20)
1		0	very interesting
2		1	very interesting
3		2	interesting
4		3	popular

Blog					
Columns (3)					
id					
name					
topic					
Constraints					

	id	name	topic
	[PK] integer	character varying (20)	character varying (20)
1		Sport & Fitness	Healthy Lifestyle
2		Anime Amino	Anime

```
1 -- Table: public.User
2
3 -- DROP TABLE public."User";
4
5 CREATE TABLE public."User"
6 (
7     id integer NOT NULL,
8     name character varying(20) COLLATE pg_catalog."default" NOT NULL,
9     registration_date date NOT NULL,
10    "e-mail_adress" character varying(40) COLLATE pg_catalog."default" NOT NULL,
11    CONSTRAINT "User_pkey" PRIMARY KEY (id)
12 )
13
14 TABLESPACE pg_default;
15
16 ALTER TABLE public."User"
17     OWNER to postgres;
```

```

1 -- Table: public.Comments
2
3 -- DROP TABLE public."Comments";
4
5 CREATE TABLE public."Comments"
6 (
7     id integer NOT NULL,
8     post_id integer NOT NULL,
9     user_id integer NOT NULL,
10    content character varying(120) COLLATE pg_catalog."default" NOT NULL,
11    CONSTRAINT "Comments_pkey" PRIMARY KEY (id),
12    CONSTRAINT "Post_id_fk" FOREIGN KEY (post_id)
13        REFERENCES public."Post" (id) MATCH SIMPLE
14        ON UPDATE NO ACTION
15        ON DELETE NO ACTION
16        NOT VALID,
17    CONSTRAINT "User_id_fk" FOREIGN KEY (user_id)
18        REFERENCES public."User" (id) MATCH SIMPLE
19        ON UPDATE NO ACTION
20        ON DELETE NO ACTION
21        NOT VALID
22 )
23
24 TABLESPACE pg_default;
25
26 ALTER TABLE public."Comments"
27     OWNER to postgres;
28 -- Index: fki_Post_id_fk
29
30 -- DROP INDEX public."fki_Post_id_fk";
31
32 CREATE INDEX "fki_Post_id_fk"
33     ON public."Comments" USING btree
34     (post_id ASC NULLS LAST)
35     TABLESPACE pg_default;
36 -- Index: fki_User_id_fk
37
38 -- DROP INDEX public."fki_User_id_fk";
39
40 CREATE INDEX "fki_User_id_fk"
41     ON public."Comments" USING btree
42     (user_id ASC NULLS LAST)
43     TABLESPACE pg_default;

```

```
1 -- Table: public.Post
2
3 -- DROP TABLE public."Post";
4
5 CREATE TABLE public."Post"
6 (
7     id integer NOT NULL,
8     blog_id integer NOT NULL,
9     content character varying(500) COLLATE pg_catalog."default" NOT NULL,
10    CONSTRAINT "Post_pkey" PRIMARY KEY (id),
11    CONSTRAINT blog_id_fk FOREIGN KEY (blog_id)
12        REFERENCES public."Blog" (id) MATCH SIMPLE
13        ON UPDATE NO ACTION
14        ON DELETE NO ACTION
15        NOT VALID
16 )
17
18 TABLESPACE pg_default;
19
20 ALTER TABLE public."Post"
21     OWNER to postgres;
22 -- Index: fki_blog_id_fk
23
24 -- DROP INDEX public.fki_blog_id_fk;
25
26 CREATE INDEX fki_blog_id_fk
27     ON public."Post" USING btree
28     (blog_id ASC NULLS LAST)
29     TABLESPACE pg_default;
```

```
1 -- Table: public.Rating
2
3 -- DROP TABLE public."Rating";
4
5 CREATE TABLE public."Rating"
6 (
7     rating_id integer NOT NULL,
8     "amount " integer NOT NULL,
9     post_rate character varying(20) COLLATE pg_catalog."default" NOT NULL,
10    CONSTRAINT "Rating_pkey" PRIMARY KEY (rating_id),
11    CONSTRAINT "Rating_id_fkey" FOREIGN KEY (rating_id)
12        REFERENCES public."Post" (id) MATCH SIMPLE
13        ON UPDATE NO ACTION
14        ON DELETE NO ACTION
15        NOT VALID
16 )
17
18 TABLESPACE pg_default;
19
20 ALTER TABLE public."Rating"
21     OWNER to postgres;
```

```
1 -- Table: public.Blog
2
3 -- DROP TABLE public."Blog";
4
5 CREATE TABLE public."Blog"
6 (
7     id integer NOT NULL,
8     name character varying(20) COLLATE pg_catalog."default" NOT NULL,
9     topic character varying(20) COLLATE pg_catalog."default" NOT NULL,
10    CONSTRAINT "Blog_pkey" PRIMARY KEY (id)
11 )
12
13 TABLESPACE pg_default;
14
15 ALTER TABLE public."Blog"
16     OWNER to postgres;
```