



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»

Виконав: студент III курсу

ФПМ групи КВ-82

Чернухін Владислав

Перевірів: Павловський В.І.

Київ – 2020

Ознайомлення з базовими операціями СУБД PostgreSQL

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

Завдання роботи полягає у наступному:

1. Провести аналіз та опис предметного середовища;
2. Розробити концептуальну модель - модель «сутність-зв'язок» предметної галузі, обраної студентом самостійно, відповідно до пункту «Вимоги до ER-моделі»;
3. Розробити логічну модель (схему даних) БД;
4. Ознайомитись із інструментарієм PostgreSQL та pgAdmin 4;
5. Створити в СУБД PostgreSQL фізичну модель БД, використовуючи конструктори таблиць та стовпчиків;
6. Сформувати обмеження цілісності, що забезпечують:
 - а. унікальність та обов'язковість вводу первинних ключів для всіх таблиць;
 - б. перевірка на відповідність зовнішніх ключів таблиць;
 - с. обмеження на значення даних відповідних атрибутів і вивід відповідних повідомлень при їх порушенні;
 - д. обов'язковість вводу даних відповідних атрибутів;
 - е. сформувати маску вводу для відповідних атрибутів;
7. Проаналізувати фізичну модель створеної БД;
8. Заповнити створену БД даними (порядку 5-10 записів в кожній таблиці).
9. Вивести вміст таблиць створеної БД.

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. Сутності мають включати атрибути для коректної реалізації особливостей пошуку, наведених у варіанті;
4. Для побудови ER-діаграм використовувати одну із нотацій: Чена, “Пташиної лапки (Crow's foot)”, UML.

Зміст звіту

1. Опис проблемного середовища;
2. Концептуальна модель предметної області;
3. Логічна модель (схема) даних БД;
4. Склад СУБД PostgreSQL;
5. Список обмежень цілісності в термінах СУБД PostgreSQL;
6. Фізична модель (схема) даних БД в pgAdmin III;
7. Приклад вмісту БД.

Опис предметної галузі

Проектуючи базу даних Ігрова платформа, я виділив такі сутності:

Користувач (user) - створена для ініціалізації користувача.

Гра (game)-створена для ініціалізації гри.

Гравець (player) - створена для ініціалізації користувача як гравця певної гри та зберігання деякої інформації про цього гравця.

Рейтинг гри (game_rate) - створена для зберігання оцінки певної гри.

Список бажаного (wish_list) - створена для зберігання інформації про ігри, які певний користувач хотів би придбати або отримати у подарунок.

У одного користувача може бути декілька ігор у списку бажаного(1:N)

У однієї гри є єдина оцінка(1:1)

Один користувач може грати в декілька ігор, а також в одну гру грає багато користувачів(N:M)

Одна гра може бути в списку бажаного у багатьох користувачів (1:N)

Концептуальна модель предметної області “Ігрова платформа”

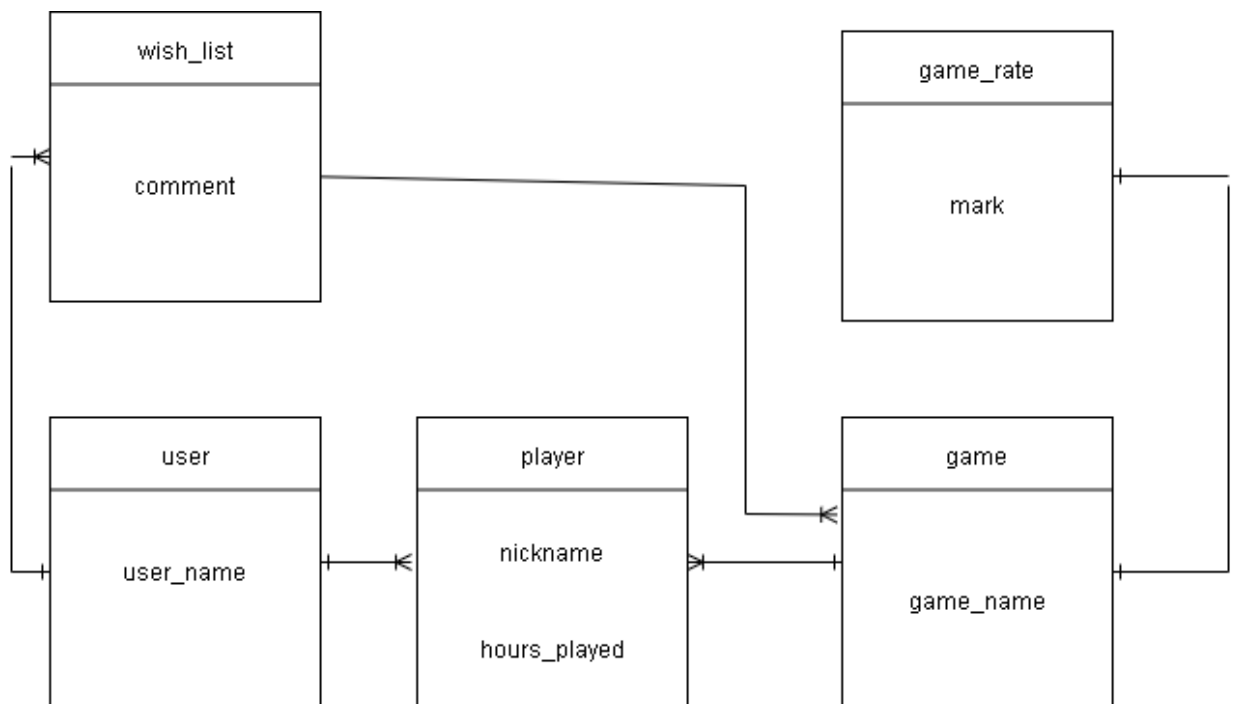


Рисунок 1 - Концептуальна модель предметної області "Ігрова платформа"

Нотація: Пташина лапка + засоби програми drow.io

Пояснення щодо відповідності схеми бази даних нормальним формам:

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

Схема бази даних відповідає 2НФ тому, що по-перше відповідає 1НФ, а також будь-який неключовий атрибут залежить лише від ключа і не залежить від підмножини його атрибутів.

Схема бази даних відповідає 3НФ тому, що по-перше вона відповідає 2НФ, а також відсутні транзитивні функціональні залежності неключових атрибутів від ключових.

Опис структури БД “ Ігрова платформа ”

Таблиця 1 – Опис структури БД “ Ігрова платформа ”

ВІДНОШЕННЯ	АТРИБУТ	ТИП (Розмір)
Відношення “ <i>user</i> ” Вміщує інформацію про користувача	user_id -унікальний id користувача user_name-ім`я користувача	Числовий Текстовий(30)
Відношення “ <i>game</i> ” Вміщує інформацію про гру	game_id - унікальний id гри game_name-назва гри	Числовий Текстовий(30)
Відношення “ <i>player</i> ” Вміщує інформацію про гравця певної гри	user_id - унікальний id користувача game_id - унікальний id гри nickname- нік гравця у певній грі hours_played-кількість годин зіграних у відповідній грі	Числовий Числовий Текстовий(30) Числовий
Відношення “ <i>wish_list</i> ” Вміщує інформацію про список бажаних ігор користувача	user_id - унікальний id користувача comment-коментар користувача game_id - унікальний id гри	Числовий Текстовий(100) Числовий
Відношення “ <i>game_rate</i> ” Вміщує в собі рейтинг певної гри	mark-оцінка game_id - унікальний id публікації	Числовий Числовий

Нотація: Червоний колір – первинний ключ. Синій колір – зовнішній ключ.

Логічна модель (Структура) БД “ Ігрова платформа ” (засобами SqlDMB)

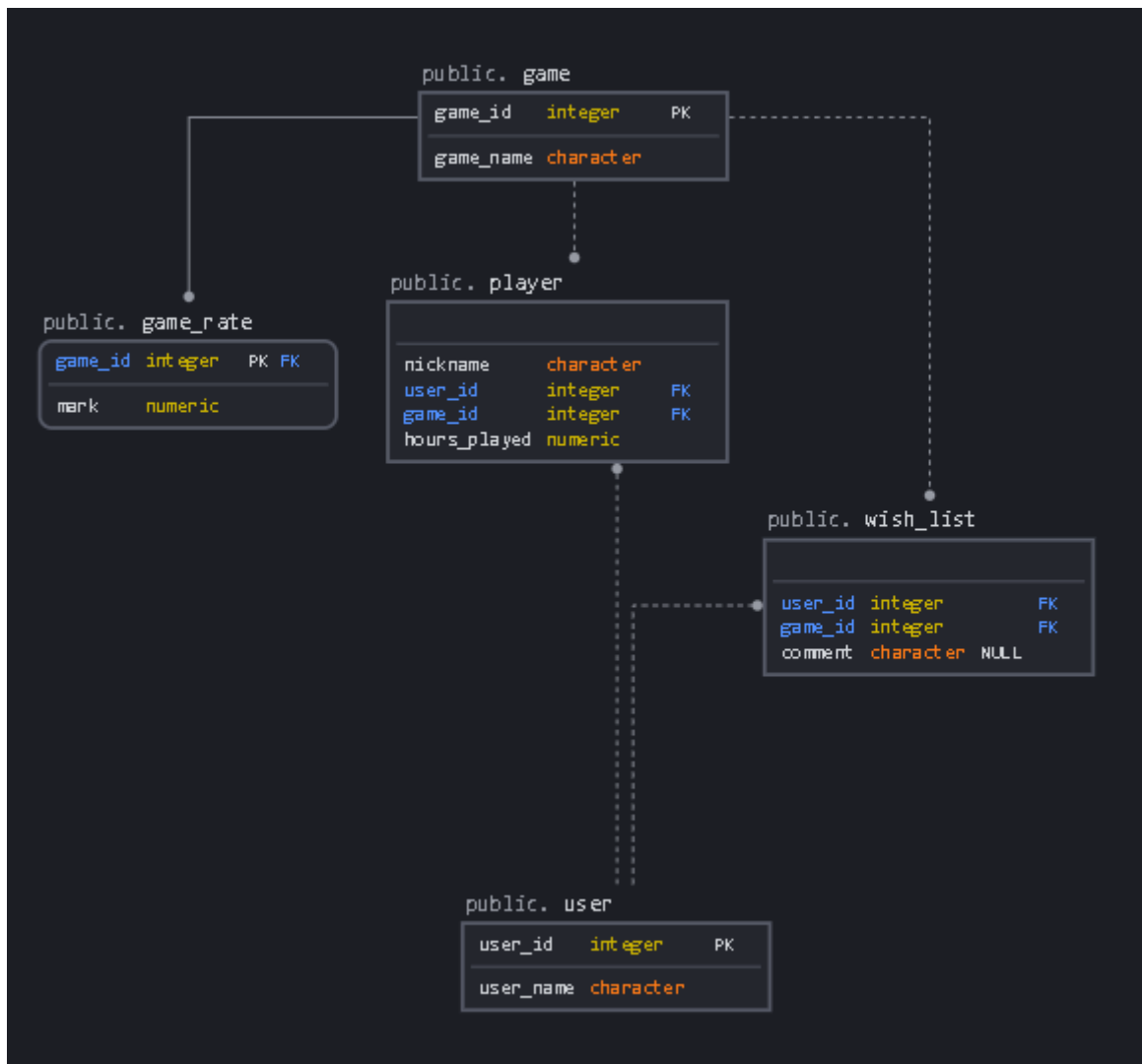
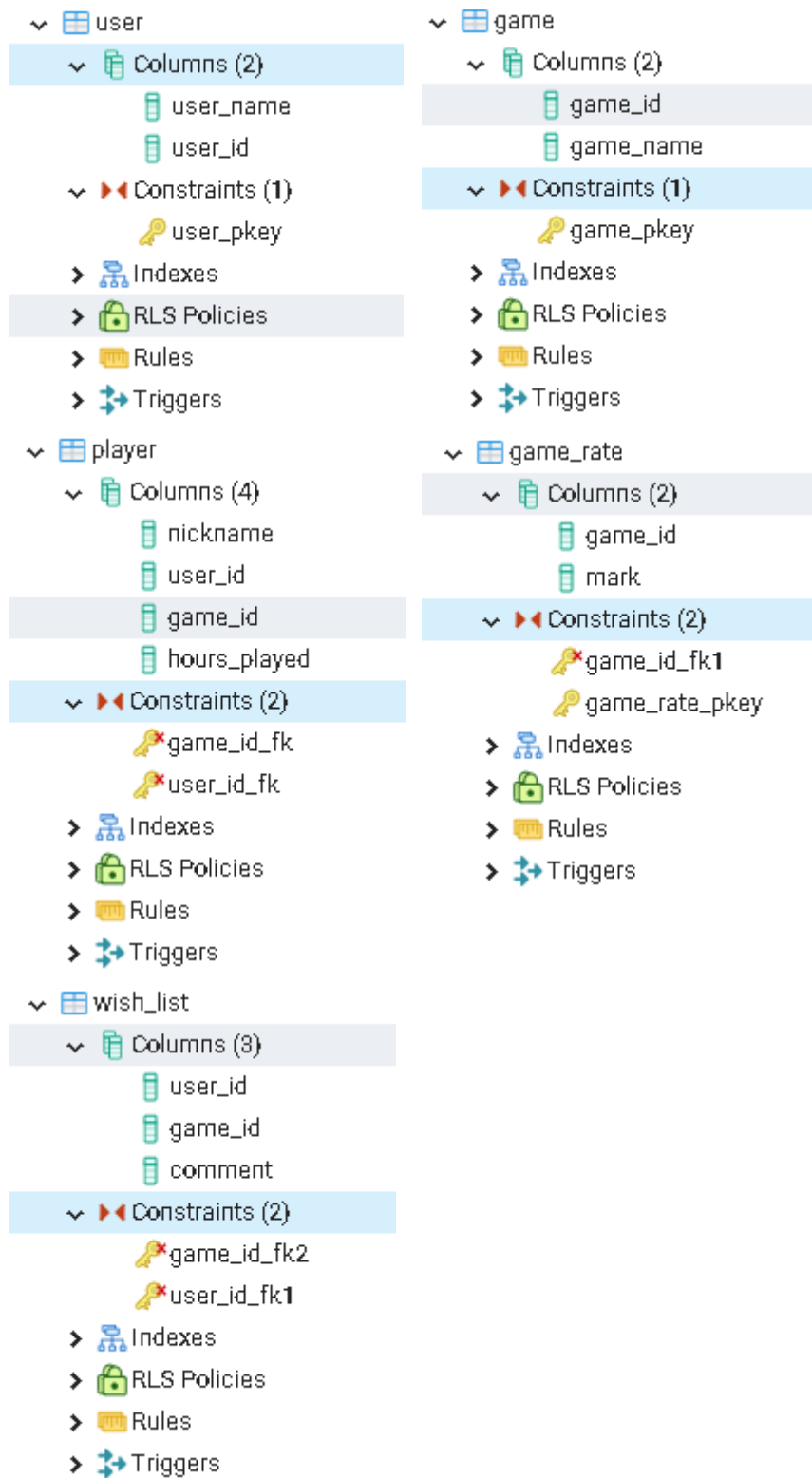


Рисунок 2 - Логічна модель (Структура) БД “ Ігрова платформа ” (засобами SqlDMB)

Фотографії таблиць мовою SQL засобами СУБД PostgreSQL



```
1 -- Table: public.user
2
3 -- DROP TABLE public."user";
4
5 CREATE TABLE public."user"
6 (
7     user_name character varying(30) COLLATE pg_catalog."default" NOT NULL,
8     user_id integer NOT NULL,
9     CONSTRAINT user_pkey PRIMARY KEY (user_id)
10 )
11
12 TABLESPACE pg_default;
13
14 ALTER TABLE public."user"
15     OWNER to postgres;
```

```
1 -- Table: public.game
2
3 -- DROP TABLE public.game;
4
5 CREATE TABLE public.game
6 (
7     game_id integer NOT NULL,
8     game_name character varying(30) COLLATE pg_catalog."default" NOT NULL,
9     CONSTRAINT game_pkey PRIMARY KEY (game_id)
10 )
11
12 TABLESPACE pg_default;
13
14 ALTER TABLE public.game
15     OWNER to postgres;
```



```

1 -- Table: public.player
2
3 -- DROP TABLE public.player;
4
5 CREATE TABLE public.player
6 (
7     nickname character varying(30) COLLATE pg_catalog."default" NOT NULL,
8     user_id integer NOT NULL,
9     game_id integer NOT NULL,
10    hours_played numeric NOT NULL,
11    CONSTRAINT game_id_fk FOREIGN KEY (game_id)
12        REFERENCES public.game (game_id) MATCH SIMPLE
13        ON UPDATE NO ACTION
14        ON DELETE NO ACTION
15        NOT VALID,
16    CONSTRAINT user_id_fk FOREIGN KEY (user_id)
17        REFERENCES public."user" (user_id) MATCH SIMPLE
18        ON UPDATE NO ACTION
19        ON DELETE NO ACTION
20        NOT VALID
21 )
22
23 TABLESPACE pg_default;
24
25 ALTER TABLE public.player
26     OWNER to postgres;
27 -- Index: fki_game_id_fk
28
29 -- DROP INDEX public.fki_game_id_fk;
30
31 CREATE INDEX fki_game_id_fk
32     ON public.player USING btree
33     (game_id ASC NULLS LAST)
34     TABLESPACE pg_default;
35 -- Index: fki_user_id_fk
36
37 -- DROP INDEX public.fki_user_id_fk;
38
39 CREATE INDEX fki_user_id_fk
40     ON public.player USING btree
41     (user_id ASC NULLS LAST)
42     TABLESPACE pg_default;

```

Активаци
Чтобы актив
раздел "Па

```
1 -- Table: public.game_rate
2
3 -- DROP TABLE public.game_rate;
4
5 CREATE TABLE public.game_rate
6 (
7     game_id integer NOT NULL,
8     mark numeric NOT NULL,
9     CONSTRAINT game_rate_pkey PRIMARY KEY (game_id),
10    CONSTRAINT game_id_fk1 FOREIGN KEY (game_id)
11        REFERENCES public.game (game_id) MATCH SIMPLE
12        ON UPDATE NO ACTION
13        ON DELETE NO ACTION
14        NOT VALID
15 )
16
17 TABLESPACE pg_default;
18
19 ALTER TABLE public.game_rate
20     OWNER to postgres;
21 -- Index: fki_game_id_fk1
22
23 -- DROP INDEX public.fki_game_id_fk1;
24
25 CREATE INDEX fki_game_id_fk1
26     ON public.game_rate USING btree
27     (game_id ASC NULLS LAST)
28     TABLESPACE pg_default;
```

```
1 -- Table: public.wish_list
2
3 -- DROP TABLE public.wish_list;
4
5 CREATE TABLE public.wish_list
6 (
7     user_id integer NOT NULL,
8     game_id integer NOT NULL,
9     comment character varying COLLATE pg_catalog."default",
10    CONSTRAINT game_id_fk2 FOREIGN KEY (game_id)
11        REFERENCES public.game (game_id) MATCH SIMPLE
12        ON UPDATE NO ACTION
13        ON DELETE NO ACTION
14        NOT VALID,
15    CONSTRAINT user_id_fk1 FOREIGN KEY (user_id)
16        REFERENCES public."user" (user_id) MATCH SIMPLE
17        ON UPDATE NO ACTION
18        ON DELETE NO ACTION
19        NOT VALID
20 )
21
22 TABLESPACE pg_default;
23
24 ALTER TABLE public.wish_list
25     OWNER to postgres;
26 -- Index: fki_game_id_fk2
27
28 -- DROP INDEX public.fki_game_id_fk2;
29
30 CREATE INDEX fki_game_id_fk2
31     ON public.wish_list USING btree
32     (game_id ASC NULLS LAST)
33     TABLESPACE pg_default;
34 -- Index: fki_user_id_fk1
35
36 -- DROP INDEX public.fki_user_id_fk1;
37
38 CREATE INDEX fki_user_id_fk1
39     ON public.wish_list USING btree
40     (user_id ASC NULLS LAST)
41     TABLESPACE pg_default;
```

Вміст таблиць

	user_name character varying (30)	user_id [PK] integer
1	alex	1
2	vlad	2

	game_id [PK] integer	game_name character varying (30)
1	1	csgo
2	2	dota
3	3	outlast

	nickname character varying (30)	user_id integer	game_id integer	hours_played numeric
1	CHIPER	1	1	1000
2	CHIPERO20	1	2	3000

	game_id [PK] integer	mark numeric		user_id integer	game_id integer	comment character varying
1	1	4.2	1	1	1	mom,plz
2	2	3				
3	3	5				