



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 KB-82

Бодашевський Даніла Романович

Перевірив: Павловський В.І.

Київ – 2020

Ознайомлення з базовими операціями СУБД PostgreSQL

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.

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. Структура БД;
4. Опис структури БД;
5. Текст програми БД;
6. Нормалізація БД

Опис предметної галузі

При проектуванні бази даних “Інтернет-аукціон картин” можна виділити такі сутності: Клієнт (Client), Картина (Painting), Автор Картини (Author), Профіль клієнта (Profile), Номер мобільного (Phone number).

У клієнта обов’язково має бути власний профіль (один до одного).

У клієнта є один або більше номерів мобільного (один до багатьох).

У автора може бути 1 або більше картин (один до багатьох).

У свою чергу у своєму профілі клієнт може мати 0 або більше картин (один до багатьох).

Картини одного автора можуть бути купленими багатьма клієнтами і навпаки клієнт може придбати картини багатьох авторів (багато до багатьох).

Концептуальна модель предметної області “Інтернет-аукціон картин”(Рис.1)

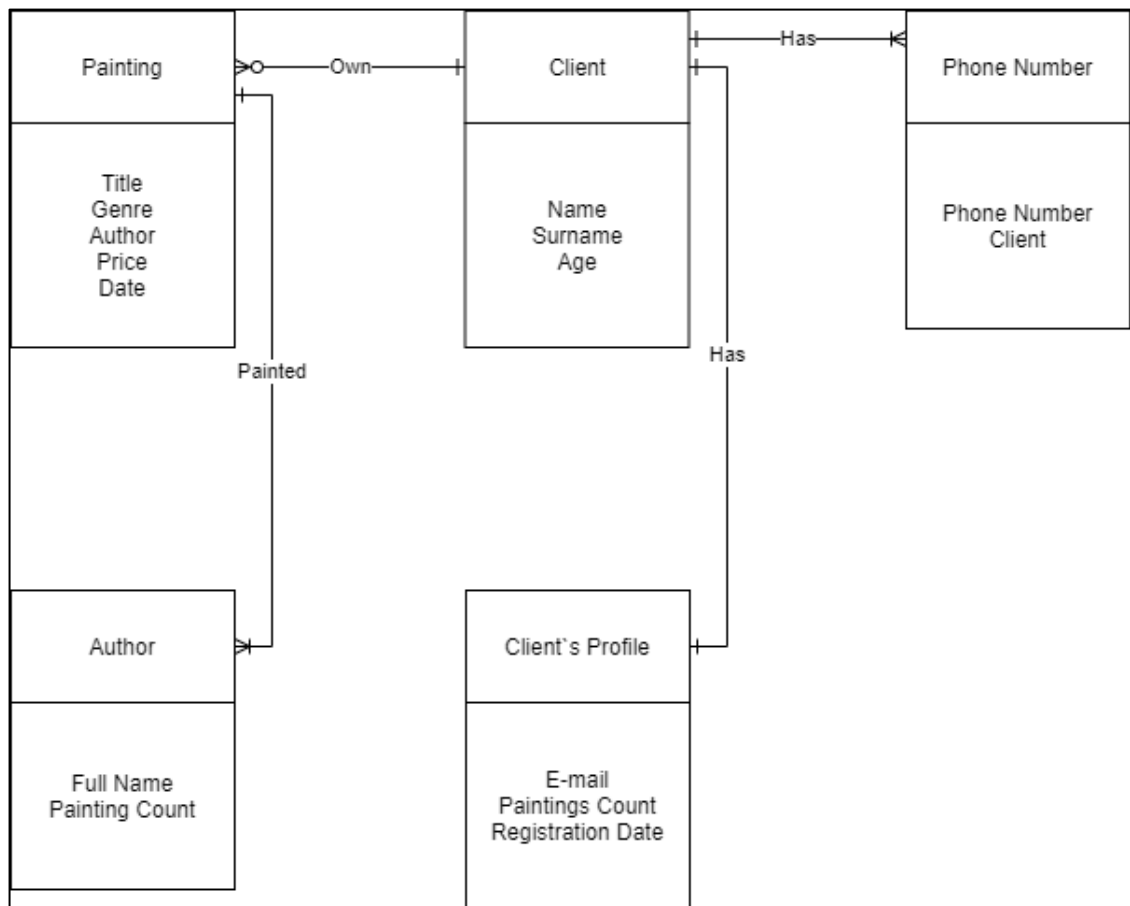


Рис.1 - Концептуальна модель «Інтернет-аукціон картин»

Структура БД «Интернет-аукціон картин»(Рис.2)

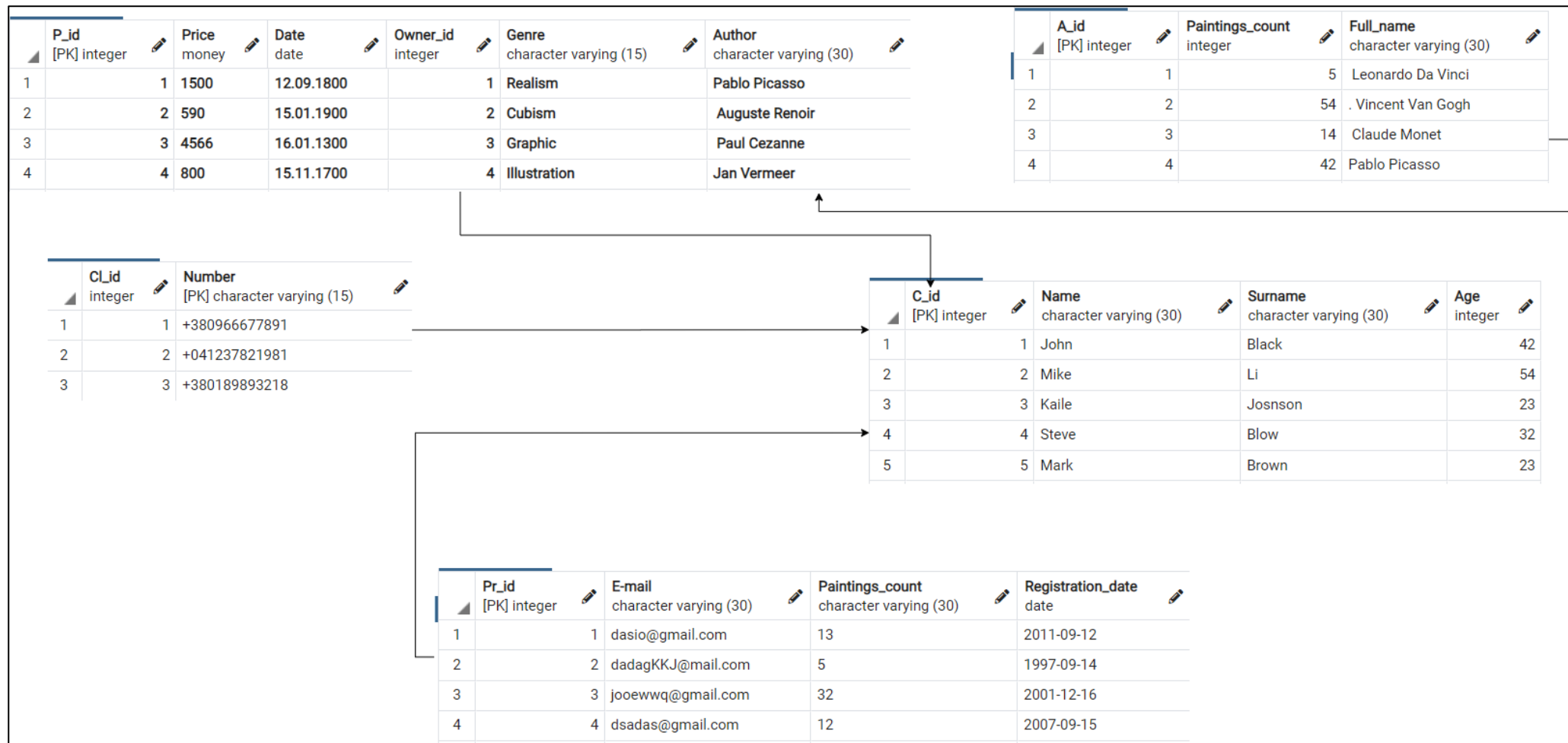


Рис.2 - Структура БД «Интернет-аукціон картин»

Опис структури БД “Інтернет-аукціон картин”

ВІДНОШЕННЯ	АТРИБУТ	ТИП(Розмір)
Відношення “ <i>Author</i> ” Вміщує інформацію про автора	A_Id – унікальний ID автора в БД Full_name – Ім’я та прізвище автора Paintings_count – Кількість картин автора	Числовий Текстовий Числовий
Відношення “ <i>Client</i> ” Вміщує інформацію про клієнта	C_Id - унікальний ID клієнта в БД Name – ім’я клієнта Surname – прізвище клієнта Age – вік клієнта	Числовий Текстовий Текстовий Числовий
Відношення “ <i>Profile</i> ” Вміщує інформацію про профіль клієнта інтернет-аукціону	Pr_Id – унікальний ID профіля в БД e-mail – e-mail клієнта Paintings_count – кількість куплених картин Registration_date – дата реєстрації профілю	Числовий Текстовий Числовий Дата
Відношення “ <i>Painting</i> ” Вміщує в собі інформацію про картину	P_Id – унікальний ID картини в БД Author – автор картини Genre – жанр картини Price – ціна картини Date – дата створення картини Owner_id – ID власника картини	Числовий Текстовий Текстовий Грошовий Дата Числовий
Відношення “ <i>Phone number</i> ” Вміщує в собі інформацію про мобільний номер клієнта	Number – мобільний номер Cl_Id – ID клієнта, якому належить мобільний номер	Текст Числовий

Логічна модель (Структура) БД “Інтернет-аукціон картин” (засобами SqlDMB) (Рис.3)

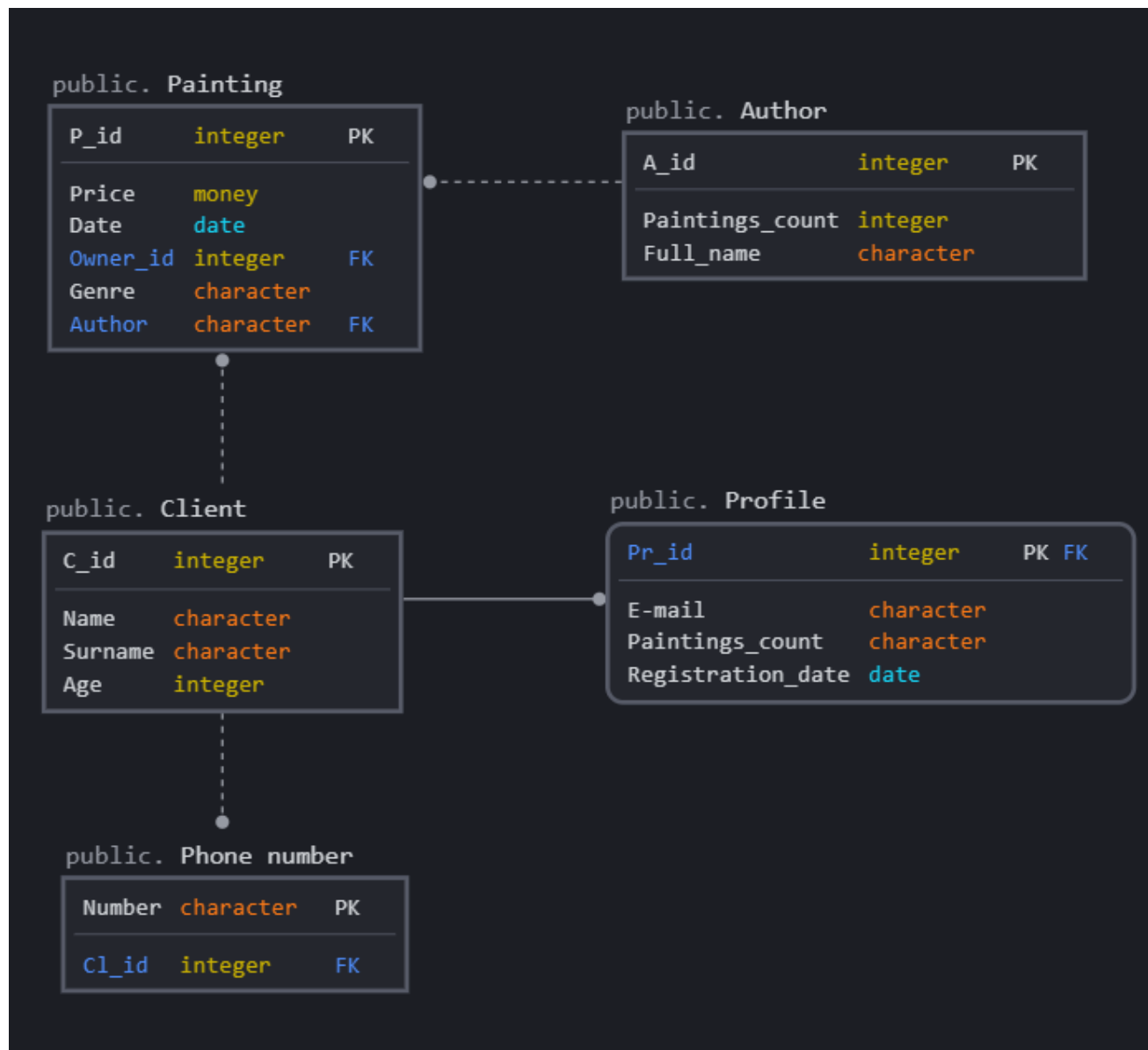
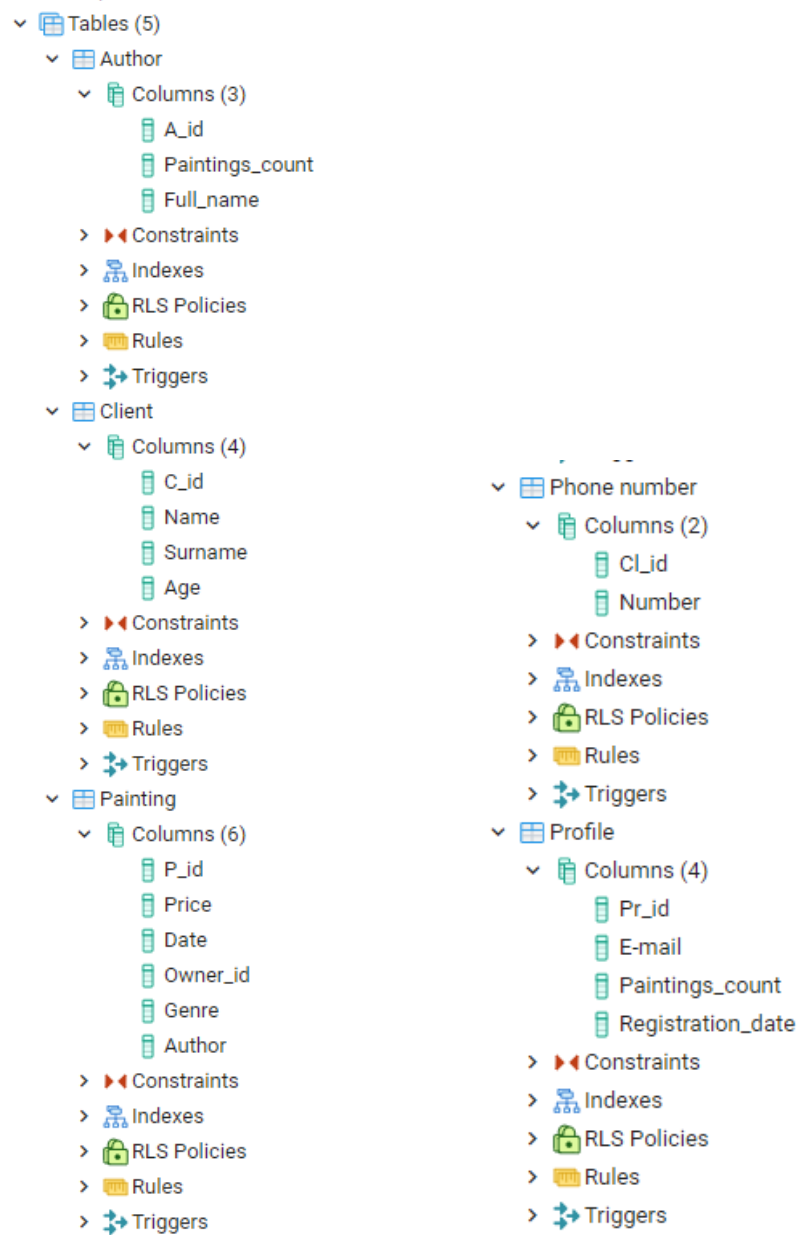


Рис.3 - Логічна модель БД «Інтернет-аукціон картин»

Структура БД “Інтернет-аукціон картин” в pgAdmin 4



Текст БД мовою SQL

```
1 -- Table: public.Profile
2
3 -- DROP TABLE public."Profile";
4
5 CREATE TABLE public."Profile"
6 (
7     "Pr_id" integer NOT NULL,
8     "E-mail" character varying(30) COLLATE pg_catalog."default" NOT NULL,
9     "Paintings_count" character varying(30) COLLATE pg_catalog."default" NOT NULL,
10    "Registration_date" date NOT NULL,
11    CONSTRAINT "Profile_pkey" PRIMARY KEY ("Pr_id"),
12    CONSTRAINT "Pr_id" UNIQUE ("Pr_id")
13        INCLUDE("Pr_id"),
14    CONSTRAINT "Clinet_id" FOREIGN KEY ("Pr_id")
15        REFERENCES public."Client" ("C_id") MATCH SIMPLE
16        ON UPDATE NO ACTION
17        ON DELETE NO ACTION
18        NOT VALID
19 )
20
21 TABLESPACE pg_default;
22
23 ALTER TABLE public."Profile"
24     OWNER to postgres;
```

```
1 -- Table: public.Author
2
3 -- DROP TABLE public."Author";
4
5 CREATE TABLE public."Author"
6 (
7     "A_id" integer NOT NULL,
8     "Paintings_count" integer NOT NULL,
9     "Full_name" character varying(30) COLLATE pg_catalog."default" NOT NULL,
10    CONSTRAINT "Author_pkey" PRIMARY KEY ("A_id"),
11    CONSTRAINT "Name" UNIQUE ("Full_name")
12        INCLUDE("Full_name")
13 )
14
15 TABLESPACE pg_default;
16
17 ALTER TABLE public."Author"
18     OWNER to postgres;
```



```
1 -- Table: public.Phone number
2
3 -- DROP TABLE public."Phone number";
4
5 CREATE TABLE public."Phone number"
6 (
7     "Cl_id" integer NOT NULL,
8     "Number" character varying(15) COLLATE pg_catalog."default" NOT NULL,
9     CONSTRAINT "Number" PRIMARY KEY ("Number")
10        INCLUDE("Number"),
11     CONSTRAINT "ClientFK" FOREIGN KEY ("Cl_id")
12        REFERENCES public."Client" ("C_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."Phone number"
21     OWNER to postgres;
```

```
-- Table: public.Client
```

```
-- DROP TABLE public."Client";
```

```
CREATE TABLE public."Client"
(
    "C_id" integer NOT NULL,
    "Name" character varying(30) COLLATE pg_catalog."default" NOT NULL,
    "Surname" character varying(30) COLLATE pg_catalog."default" NOT NULL,
    "Age" integer NOT NULL,
    CONSTRAINT "Client_pkey" PRIMARY KEY ("C_id"),
    CONSTRAINT "ClientID" UNIQUE ("C_id")
        INCLUDE("C_id")
)
```

```
TABLESPACE pg_default;
```

```
ALTER TABLE public."Client"
    OWNER to postgres;
```

```

1 -- Table: public.Painting
2
3 -- DROP TABLE public."Painting";
4
5 CREATE TABLE public."Painting"
6 (
7     "P_id" integer NOT NULL,
8     "Price" money NOT NULL,
9     "Date" date NOT NULL,
10    "Owner_id" integer NOT NULL,
11    "Genre" character varying(15) COLLATE pg_catalog."default" NOT NULL,
12    "Author" character varying(30) COLLATE pg_catalog."default" NOT NULL,
13    CONSTRAINT "Painting_pkey" PRIMARY KEY ("P_id"),
14    CONSTRAINT "PaintingID" UNIQUE ("P_id")
15        INCLUDE("P_id"),
16    CONSTRAINT "AuthorFK" FOREIGN KEY ("Author")
17        REFERENCES public."Author" ("Full_name") MATCH FULL
18        ON UPDATE NO ACTION
19        ON DELETE NO ACTION
20        NOT VALID,
21    CONSTRAINT "OwnerFK" FOREIGN KEY ("Owner_id")
22        REFERENCES public."Profile" ("Pr_id") MATCH SIMPLE
23        ON UPDATE NO ACTION
24        ON DELETE NO ACTION
25        NOT VALID
26 )
27
28 TABLESPACE pg_default;
29
30 ALTER TABLE public."Painting"
31     OWNER to postgres;

```

Вміст таблиць

Profile

	Pr_id [PK] integer	E-mail character varying (30)	Paintings_count character varying (30)	Registration_date date
1	1	dasio@gmail.com	13	2011-09-12
2	2	dadagKKJ@mail.com	5	1997-09-14
3	3	jooewwq@gmail.com	32	2001-12-16
4	4	dsadas@gmail.com	12	2007-09-15

Author

	A_id [PK] integer	Paintings_count integer	Full_name character varying (30)
1	1	5	Leonardo Da Vinci
2	2	54	. Vincent Van Gogh
3	3	14	Claude Monet
4	4	42	Pablo Picasso

Client

	C_id [PK] integer	Name character varying (30)	Surname character varying (30)	Age integer
1	1	John	Black	42
2	2	Mike	Li	54
3	3	Kaile	Josnson	23
4	4	Steve	Blow	32
5	5	Mark	Brown	23

Painting

	P_id [PK] integer	Price money	Date date	Owner_id integer	Genre character varying (15)	Author character varying (30)
1	1	1500	12.09.1800	1	Realism	Pablo Picasso
2	2	590	15.01.1900	2	Cubism	Auguste Renoir
3	3	4566	16.01.1300	3	Graphic	Paul Cezanne
4	4	800	15.11.1700	4	Illustration	Jan Vermeer

Phone Number

	CL_id integer	Number [PK] character varying (15)
1	1	+380966677891
2	2	+041237821981
3	3	+380189893218

Нормалізація БД

Перша нормальна форма

Перша нормальна форма (1НФ, 1NF) утворює ґрунт для структурованої схеми бази даних:

Кожна таблиця повинна мати основний ключ: мінімальний набір колонок, які ідентифікують запис.

Уникнення повторень груп (категорії даних, що можуть зустрічатись різну кількість разів в різних записах) правильно визначаючи неключові атрибути.

Атомарність: кожен атрибут повинен мати лише одне значення, а не множину значень.

Друга нормальна форма

Друга нормальна форма (2НФ, 2NF) вимагає, аби дані, що зберігаються в таблицях із композитним ключем, не залежали лише від частини ключа:

Схема бази даних повинна відповідати вимогам першої нормальної форми.

Дані, що повторно з'являються в декількох рядках, виносяться в окремі таблиці.

Третя нормальна форма

Третя нормальна форма (3НФ, 3NF) вимагає, аби дані в таблиці залежали винятково від основного ключа:

Схема бази даних повинна відповідати всім вимогам другої нормальної форми.

Будь-яке поле, що залежить від основного ключа та від будь-якого іншого поля, має виноситись в окрему таблицю.