

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**

*на тему: “Проектування бази даних та ознайомлення з базовими операціями СУБД PostgreSQL”*

Виконав:

студент ІII курсу

групи КВ-02

Войтович Гліб Олександрович

Перевірив: Павловський Володимир Ілліч

Київ – 2022

*Метою роботи є* здобуття 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 кожну з таблиць засобами pgAdmin 4.

*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 області «Комп’ютерний клуб»**

Ця предметна область передбачає збереження інформації про будівлю комп'ютерного клубу, його клієнтів, кімнати, столи та обладнання (комп'ютери). Комп'ютерний клуб має кілька кімнат, у кімнатах розташовані столи, на яких є комп'ютери. За столами працюють клієнти.

**Опис сутностей предметної області**

* 1. Кімната (Room), з атрибутами: код кімнати (RoomID) та її номер (Number). Призначена для зберігання інформації про кімнату;
  2. Стіл (Table), з атрибутами: код столу (TableID), код кімнати (RoomID) та його номер (Number). Призначена для зберігання інформації про стіл;
  3. Комп'ютер (Computer) з атрибутами: код комп'ютера (ComputerID), код стола (Table ID), серійний номер комп'ютера (Serial), номер комп’ютера (Number) та його виробник (Brand). Призначена для зберігання інформації про комп’ютер;
  4. Клієнт (Client), з атрибутами: код клієнта (ClientID), повне ім’я (FullName), дата реєстрації у клубі (RegDate) та дата народження клієнта (BirthDate). Призначена для зберігання інформації про клієнтів комп'ютерного клубу;
  5. Асоціативна сутність Сесія (Session), з атрибутами: код сесії (SessionID), код клієнта (ClientID), код стола (TableID), дату початку сеансу (StartDate), дату закінчення сеансу (EndDate) і статус сеансу (Status). Призначена для створення зв’язку N:M між сутностями «Клієнт» та «Стіл».

**Опис зав’язків між сутностями предметної області**

Сутність «Кімната» має 1:N зв'язок по відношенню до сутності «Стіл», тому що в кімнаті може розташовуватися багато столів, але столи закріплені за конкретною кімнатою.

Сутність «Стіл» має 1:N зв'язок по відношенню до асоціативної сутності «Сесія» тому що сесій може бути багато, але одна сесія може мати лише один стіл.

Сутність «Клієнт» має зв'язок 1:N по відношенню до сутності «Сесія», оскільки один клієнт може мати безліч сесій (як активних, і неактивних), але конкретна сесія може бути прив'язана лише до одного клієнта.

Сутність «Стіл» має зв'язок 1:1 по відношенню до сутності «Комп’ютер», оскільки кожний комп’ютер є унікальним (зі своїм серійним номером) і може стояти тільки на одному столі.

Асоціативна сутність «Сесія» – додаткова сутність, що допомагає реалізувати зв’язки N:M між сутностями «Стіл» та «Клієнт».

**Концептуальна модель предметної області «Комп'ютерний клуб»**

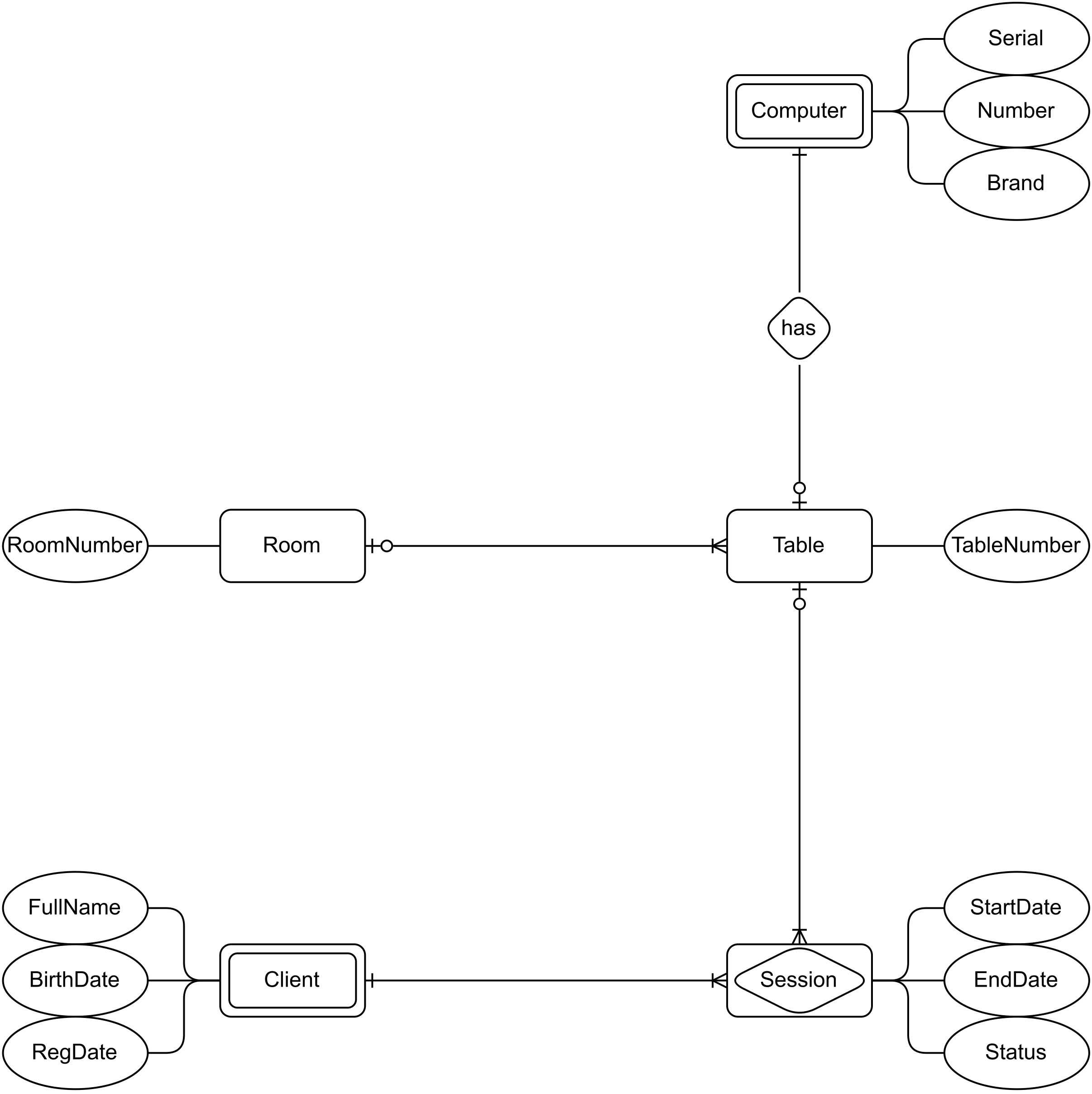


Рисунок 1. ER-діаграма (інструмент: draw.io)

**Логічна модель (схема) БД «Комп'ютерний клуб»**

Сутність «Кімната» перетворена на таблицю «Room».

Сутність «Стіл» перетворено на таблицю «Table».

Сутність «Комп'ютер» перетворено на таблицю «Computer».

Сутність «Клієнт» перетворена на таблицю «Client».

Сутність «Сесія» перетворена на таблицю «Session».

**Логічна модель (схема) БД «Комп'ютерний клуб»**

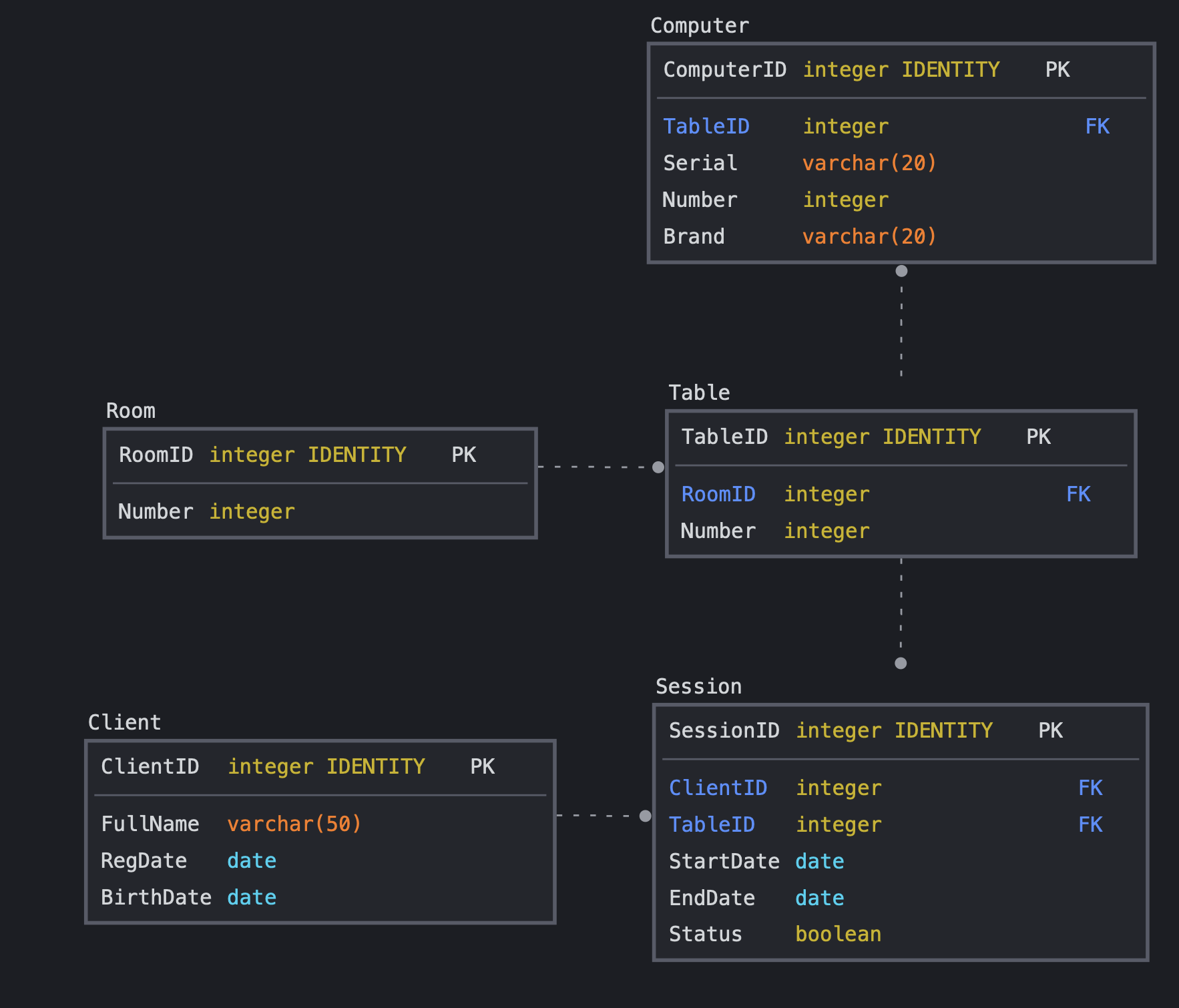


Рисунок 2. Схема бази даних (інструмент: sqldbm.com)

**Опис об’єктів бази даних**

Таблиця 1. Опис структури БД

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип атрибуту |
| **Room** – *містить інформацію про те, що знаходиться в кімнаті* | **RoomID** *- унікальний ідентифікатор кімнати. Не допускає NULL.*  **RoomNumber***– номер кімнати. Не допускає NULL.* | **integer** *(числовий)*  **integer** *(числовий)* |
| **Table** – *містить інформацію про стіл* | **TableID** *- унікальний ідентифікатор стола. Не допускає NULL.*  **RoomID** *- унікальний ідентифікатор кімнати. Не допускає NULL.*  **TableNumber** *– номер стола. Не допускає NULL.* | **integer** *(числовий)*  **integer** *(числовий)*  **integer** *(числовий)* |
| **Computer** – *містить інформацію про компʼютер* | **ComputerID** *- унікальний ідентифікатор комп’ютера. Не допускає NULL.*  **TableID** *- унікальний ідентифікатор стола. Не допускає NULL.*  **Serial –** *серійний номер. Не допускає NULL.*  **Number –** *номер комп’ютера. Не допускає NULL.*  **Brand –** *назва виробника комп’ютера. Не допускає NULL.* | **integer** *(числовий)*  **integer** *(числовий)*  **varchar(20)** *(строковий)*  **integer** *(числовий)*  **varchar(20)** *(строковий)* |
| **Client** – *містить інформацію про клієнтів комп'ютерного клубу* | **ClientID** *- унікальний ідентифікатор клієнта. Не допускає NULL.*  **FullName** *– імʼя кліента. Не допускає NULL.*  **RegDate** - *дата реєстрації клієнта. Не допускає NULL.*  **BirthDate** *– дата народження клієнта. Не допускає NULL.* | **integer** *(числовий)*  **varchar(50)** *(символьний)*  **date** *(дата й час)*  **date** *(дата й час)* |
| **Session** – *містить інформацію про сесії* | **SessionID** *- унікальний ідентифікатор сесії. Не допускає NULL.*  **TableID** *- унікальний ідентифікатор стола. Не допускає NULL.*  **StartDate** *– час початку сесії. Не допускає NULL.*  **EndDate** *– час кінця сесії. Не допускає NULL.*  **Status** *– стан сесії, Не допускає NULL.* | **integer** *(числовий)*  **integer** *(числовий)*  **date** *(дата й час)*  **date** *(дата й час)*  **boolean** *(логiчний)* |

**Функціональні залежності для кожної таблиці**

CLIENT:

ClientID - FullName, RegDate, BirthDate

ClientID – FullName

ClientID – RegDate

ClientID – BirthDate

ROOM:

RoomID – Number

RoomID – Number

TABLE:

TableID – RoomID, Number

TableID - RoomID

TableID – Number

COMPUTER:

ComputerID – TableID, Serial, Number, Brand

ComputerID – TableID

ComputerID – Serial

ComputerID – Number

ComputerID – Brand

SESSION:

SessionID – ClientID, TableID, StartDate, EndDate, Status

SessionID – ClientID

SessionID – TableID

SessionID – StartDate

SessionID – EndDate

SessionID – Status

**Відповідність схеми бази даних до третьої нормальної форми**

В таблиці немає дубльованих рядків, у кожній комірці зберігається атомарне значення, а також у кожному стовпчику зберігаються дані одного типу. Це означає, що схема відповідає 1НФ.

Всі атрибути залежать від усього первинного ключа цілком, а не від якоїсь його частини. Також схема відповідає 1НФ. Звідси слідує, що схема бази даних відповідає 2НФ.

Кожний неключовий атрибут таблиці залежить тільки від первинного ключа, а не має деяку транзитивну залежність, не залежить від інших неключових атрибутів. Також схема відповідає 2НФ. Звідси слідує, що схема бази даних відповідає 3НФ.

**Таблиці бази даних у pgAdmin4**

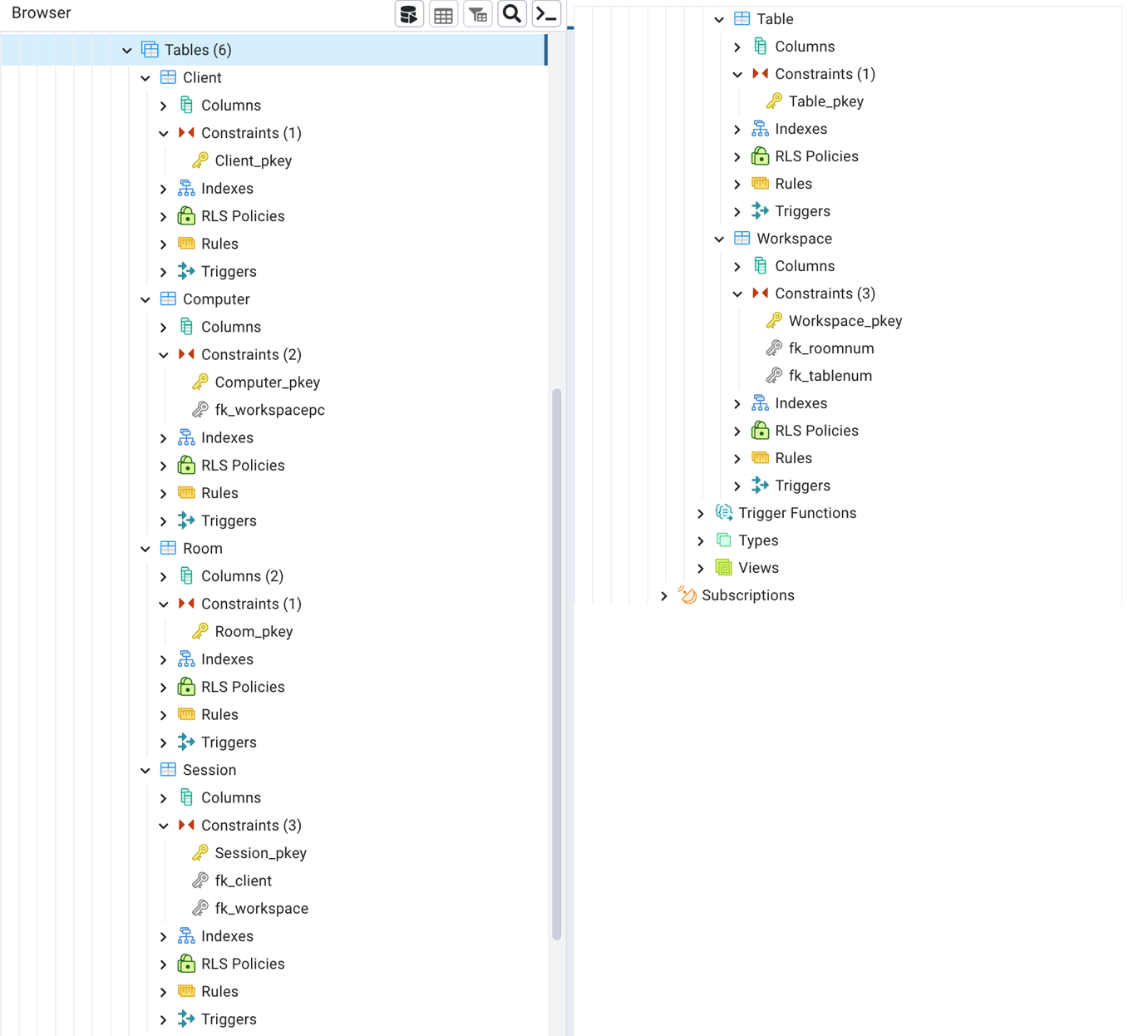


Рисунок 3 -Схема бази даних у pgAdmin 4

**Зображення вмісту таблиць**

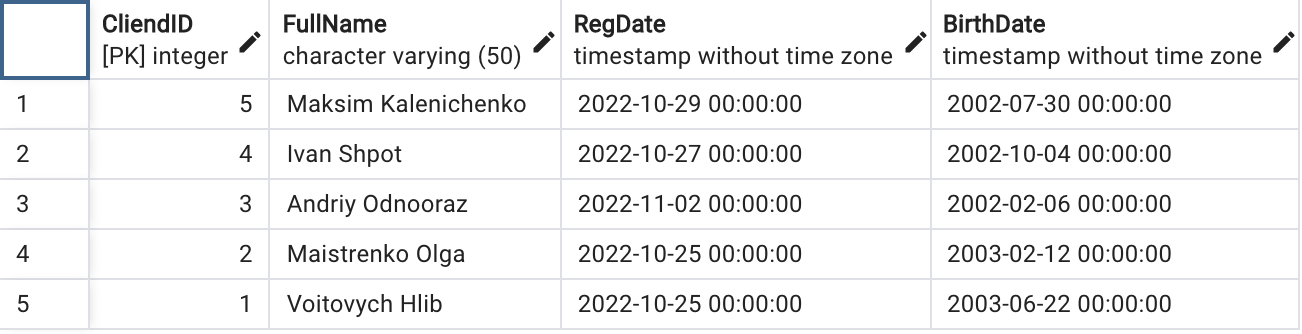


Рисунок 4. Дані таблиці Client



Рисунок 5. Дані таблиці Computer

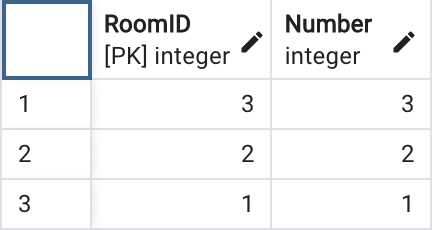


Рисунок 6. Дані таблиці Room

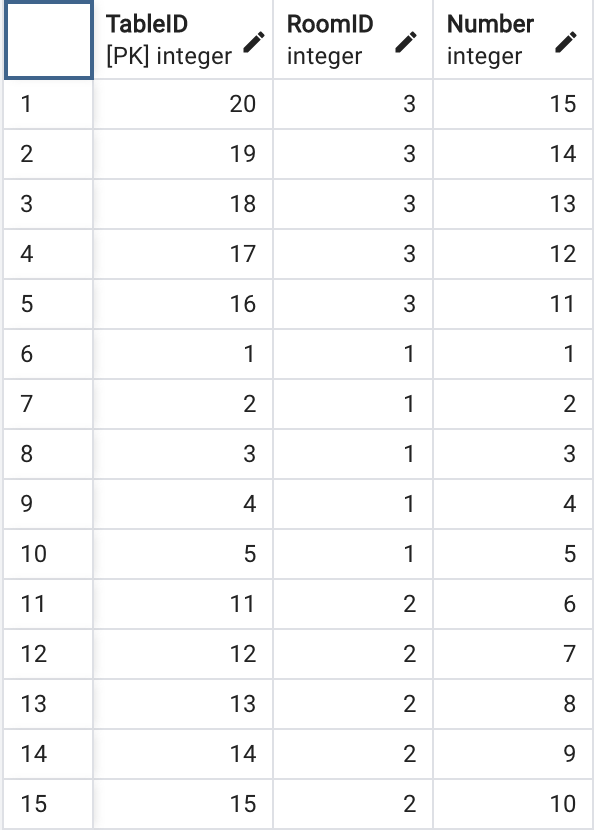


Рисунок 7. Дані таблиці Table

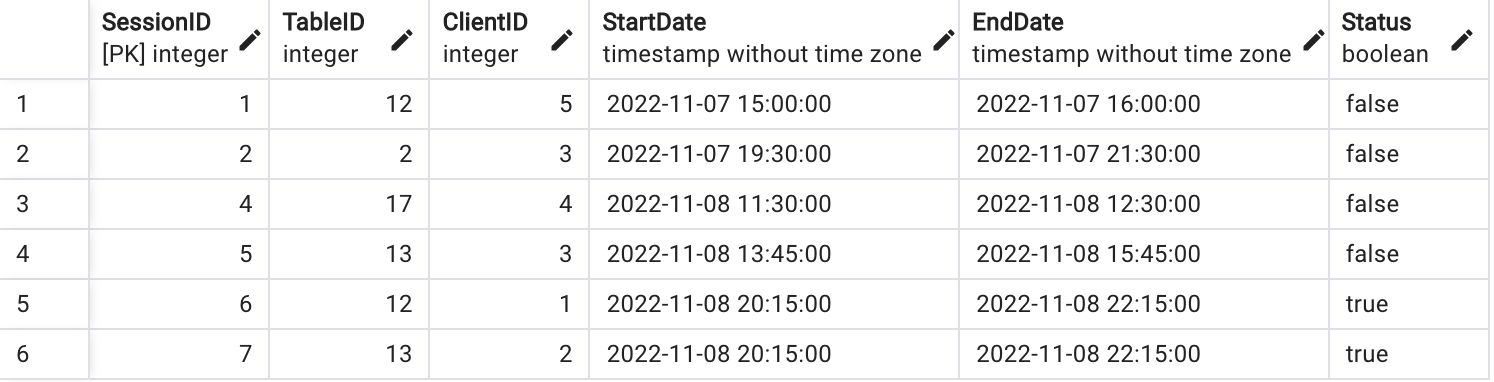


Рисунок 8. Дані таблиці Session

**SQL-текст опису БД**

*-- Database: ComputerClub*  
*-- DROP DATABASE IF EXISTS "ComputerClub";*  
**CREATE** **DATABASE** **"ComputerClub"**  
    **WITH**  
    OWNER = postgres  
    **ENCODING** = **'UTF8'**  
    LC\_COLLATE = **'C'**  
    LC\_CTYPE = **'C'**  
    **TABLESPACE** = pg\_default  
    **CONNECTION** **LIMIT** = -1  
    IS\_TEMPLATE = **False**;

*-- Table: public.Client*  
*-- DROP TABLE IF EXISTS public."Client";*  
**CREATE** **TABLE** **IF** **NOT** **EXISTS** public.**"Client"**  
(  
    **"CliendID"** integer **NOT** **NULL** **GENERATED** **ALWAYS** **AS** **IDENTITY** ( **INCREMENT** 1 **START** 1 **MINVALUE** 1 MAXVALUE 2147483647 **CACHE** 1 ),  
    **"FullName"** character varying(50) **COLLATE** pg\_catalog.**"default"** **NOT** **NULL**,  
    **"RegDate"** timestamp **without** time zone **NOT** **NULL**,  
    **"BirthDate"** timestamp **without** time zone **NOT** **NULL**,  
    **CONSTRAINT** **"Client\_pkey"** PRIMARY **KEY** (**"CliendID"**)  
)  
**TABLESPACE** pg\_default;  
**ALTER** **TABLE** **IF** **EXISTS** public.**"Client"**  
    OWNER **to** postgres;

*-- Table: public.Computer*  
*-- DROP TABLE IF EXISTS public."Computer";*  
**CREATE** **TABLE** **IF** **NOT** **EXISTS** public.**"Computer"**  
(  
    **"ComputerID"** integer **NOT** **NULL** **GENERATED** **ALWAYS** **AS** **IDENTITY** ( **INCREMENT** 1 **START** 1 **MINVALUE** 1 MAXVALUE 2147483647 **CACHE** 1 ),  
    **"TableID"** integer **NOT** **NULL**,  
    **"Brand"** character varying(20) **COLLATE** pg\_catalog.**"default"** **NOT** **NULL**,  
    **"Serial"** character varying(20) **COLLATE** pg\_catalog.**"default"** **NOT** **NULL**,  
    **"Number"** integer **NOT** **NULL**,  
    **CONSTRAINT** **"Computer\_pkey"** PRIMARY **KEY** (**"ComputerID"**),  
    **CONSTRAINT** fk\_maintable **FOREIGN** **KEY** (**"TableID"**)  
        **REFERENCES** public.**"Table"** (**"TableID"**) **MATCH** SIMPLE  
        **ON** **UPDATE** **NO** **ACTION**  
        **ON** **DELETE** **NO** **ACTION**  
)  
**TABLESPACE** pg\_default;  
**ALTER** **TABLE** **IF** **EXISTS** public.**"Computer"**  
    OWNER **to** postgres;

*-- Table: public.Room*  
*-- DROP TABLE IF EXISTS public."Room";*  
**CREATE** **TABLE** **IF** **NOT** **EXISTS** public.**"Room"**  
(  
    **"RoomID"** integer **NOT** **NULL** **GENERATED** **ALWAYS** **AS** **IDENTITY** ( **INCREMENT** 1 **START** 1 **MINVALUE** 1 MAXVALUE 2147483647 **CACHE** 1 ),  
    **"Number"** integer **NOT** **NULL**,  
    **CONSTRAINT** **"Room\_pkey"** PRIMARY **KEY** (**"RoomID"**)  
)  
**TABLESPACE** pg\_default;  
**ALTER** **TABLE** **IF** **EXISTS** public.**"Room"**  
    OWNER **to** postgres;

*-- Table: public.Session*  
*-- DROP TABLE IF EXISTS public."Session";*  
**CREATE** **TABLE** **IF** **NOT** **EXISTS** public.**"Session"**  
(  
    **"SessionID"** integer **NOT** **NULL** **GENERATED** **ALWAYS** **AS** **IDENTITY** ( **INCREMENT** 1 **START** 1 **MINVALUE** 1 MAXVALUE 2147483647 **CACHE** 1 ),  
    **"TableID"** integer **NOT** **NULL**,  
    **"ClientID"** integer **NOT** **NULL**,  
    **"StartDate"** timestamp **without** time zone **NOT** **NULL**,  
    **"EndDate"** timestamp **without** time zone **NOT** **NULL**,  
    **"Status"** boolean **NOT** **NULL**,  
    **CONSTRAINT** **"Session\_pkey"** PRIMARY **KEY** (**"SessionID"**),  
    **CONSTRAINT** fk\_client **FOREIGN** **KEY** (**"ClientID"**)  
        **REFERENCES** public.**"Client"** (**"CliendID"**) **MATCH** SIMPLE  
        **ON** **UPDATE** **NO** **ACTION**  
        **ON** **DELETE** **NO** **ACTION**,  
    **CONSTRAINT** fk\_table **FOREIGN** **KEY** (**"TableID"**)  
        **REFERENCES** public.**"Table"** (**"TableID"**) **MATCH** SIMPLE  
        **ON** **UPDATE** **NO** **ACTION**  
        **ON** **DELETE** **NO** **ACTION**  
)  
**TABLESPACE** pg\_default;  
**ALTER** **TABLE** **IF** **EXISTS** public.**"Session"**  
    OWNER **to** postgres;

*-- Table: public.Table*  
*-- DROP TABLE IF EXISTS public."Table";*  
**CREATE** **TABLE** **IF** **NOT** **EXISTS** public.**"Table"**  
(  
    **"TableID"** integer **NOT** **NULL** **GENERATED** **ALWAYS** **AS** **IDENTITY** ( **INCREMENT** 1 **START** 1 **MINVALUE** 1 MAXVALUE 2147483647 **CACHE** 1 ),  
    **"RoomID"** integer **NOT** **NULL**,  
    **"Number"** integer **NOT** **NULL**,  
    **CONSTRAINT** **"Table\_pkey"** PRIMARY **KEY** (**"TableID"**),  
    **CONSTRAINT** fk\_room **FOREIGN** **KEY** (**"RoomID"**)  
        **REFERENCES** public.**"Room"** (**"RoomID"**) **MATCH** SIMPLE  
        **ON** **UPDATE** **NO** **ACTION**  
        **ON** **DELETE** **NO** **ACTION**  
)  
**TABLESPACE** pg\_default;  
**ALTER** **TABLE** **IF** **EXISTS** public.**"Table"**  
    OWNER **to** postgres;

GitHub Repository: https://github.com/prodamgarajj/DatabaseKPI