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-91

Щикалюк Г. І.

Перевірив:

Київ – 2021

*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 кожну з таблиць засобами pgAdmin 4.

*Зміст звіту*

1. Опис предметної області;
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 Ⅳ;
7. Приклад вмісту БД;

**Опис предметної області “Телефонний вузол”**

Обрана предметна область передбачає облік абонентів та дзвінків, їх тарифікацію в реальному часі. Телефонна компанія надає свої послуги абонентам за певними тарифами. Абоненти роблять виклики. Кожен абонент може бути як робити виклики так і приймати їх

**Опис сутностей предметної області**

Для побудови бази даних для обраної області було виділено сутності, зображені на рисунку 1:

1. Тарифікація (Tarrification), з атрибутами: тариф, плата, ідентифікаційний номер абонента. Призначена для збереження інформації про тарифікацію абонентів
2. Абонент (Subscriber) з атрибутами: ім’я абонента, ідентифікаційний номер абонента, номер телефону на який абонент робить виклик. Призначена для збереження інформації про абонентів
3. Дзвінок (Call), з атрибутами: ідентифікаційний номер дзвінка, перший абонент, другий абонент, час початку дзвінка, час кінця дзвінка. Призначена для збереження інформації про дзвінок.

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

Сутність “Тарифікація” має зв’язок 1:N по відношенню до абонента, тому що багато абонентів можуть обслуговуватися за одним й тим самим тарифним планом

Сутність “Абонент” має зв’язок 1:N по відношенню до дзвінка, тому що один абонент може як приймати так і здійснювати багато викликів

**Концептуальна модель предметної області “Телефонний вузол**

Концептуальна модель наведена на рисунку 1.

У діаграмі присутній лише зв’язок 1:N. Між абонентом та дзвінком присутньо два відношення – абонент може як робити дзвінки так і приймати їх.

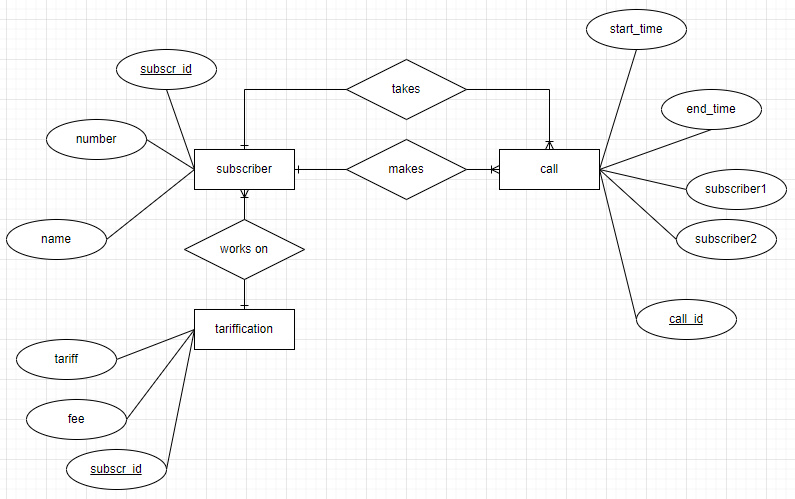


Рисунок 1 – ER-діаграма, побудована за нотацією Чена (інструмент: draw.io)

**Перетворення концептуальної моделі у логічну схему бази даних**

Сутність “Тарифікація” перетворено у таблицю “tariffication”.

Сутність “Абонент” перетворено у таблицю “subscriber”

Сутність “Дзвінок” перетворено у таблицю “call”

**Логічна модель (схема) БД «Телефонний вузол»**

Логічну модель (схему бази даних наведено на рисунку 2).

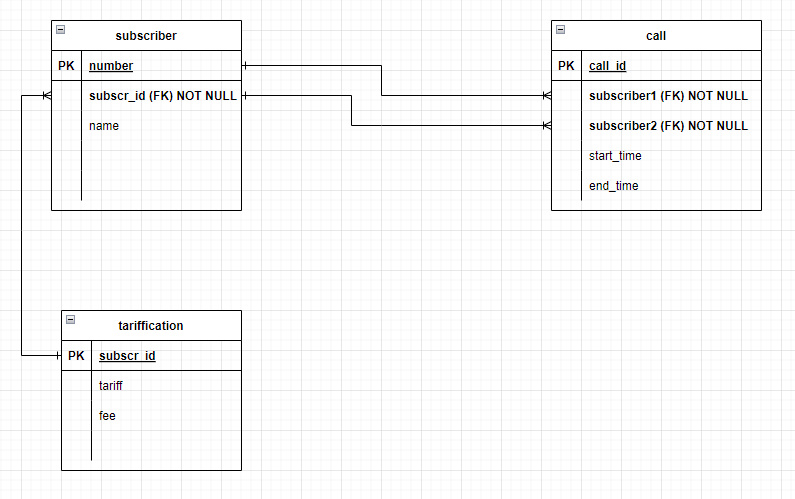


Рисунок 2 – Схема бази даних (інструмент: draw.io)

**Опис об’єктів бази даних у вигляді таблиці**

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип атрибуту |
| **tarrification** – *містить данні про тарифи закріплені за абонентами* | **subscr\_id** – *унікальний ідентифікатор абонента в базі*  **tariff** – *тариф, за яким абоненту надаються послуги*  **fee** – *ціна, яку абонент сплачує за користування тарифом* | **integer** *(числовий)*  **text** *(текстовий)*  **integer** *(числовий)* |
| **subscriber** – *містить інформацію про абонента* | **number** – *номер, на який абонент робить виклик*  **subscr\_id** – *унікальний ідентифікатор абонента в базі*  **name** – *ім’я абонента* | **text** *(текстовий)*  **integer** *(числовий)*  **text** *(текстовий)* |
| **call** – *містить дані про дзвінок* | **call\_id** – *ідентифікатор дзвінка*  **subscriber1** – *абонент, який приймає виклик*  **subscriber2** – *абонент, який робить виклик*  **start\_time** – *час початку виклику*  **end\_time** – *час кінця виклику* | **integer** *(числовий)*  **text** *(текстовий)*  **integer** *(числовий)*  **time without timezone** *(час без часового поясу)*  **time without timezone** *(час без часового поясу)* |

**Функціональні залежності для кожної таблиці**

TARIFFICATION:

subscriber\_id → tariff (абонент обслуговуються за певним тарифом)

tariff → fee (тариф визначає плату за обслуговування)

SUBSCRIBER:

number → name (в одному дзвінку є лише один абонент що здійснює виклик, отже його ім’я закріплене саме за ним)

CALL:

call\_id → start\_time; end\_time

call\_id → start\_time (дзвінок має лише один час початку)

call\_id → end\_time (дзвінок має лише один час кінця)

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

Схема відповідає 1НФ, тому що для всіх таблиць виконуються наступні властивості:

* Кожна клітинка містить єдине значення
* Кожен запис є унікальним

Схема відповідає 2НФ, тому що для всіх таблиць виконуються наступні властивості:

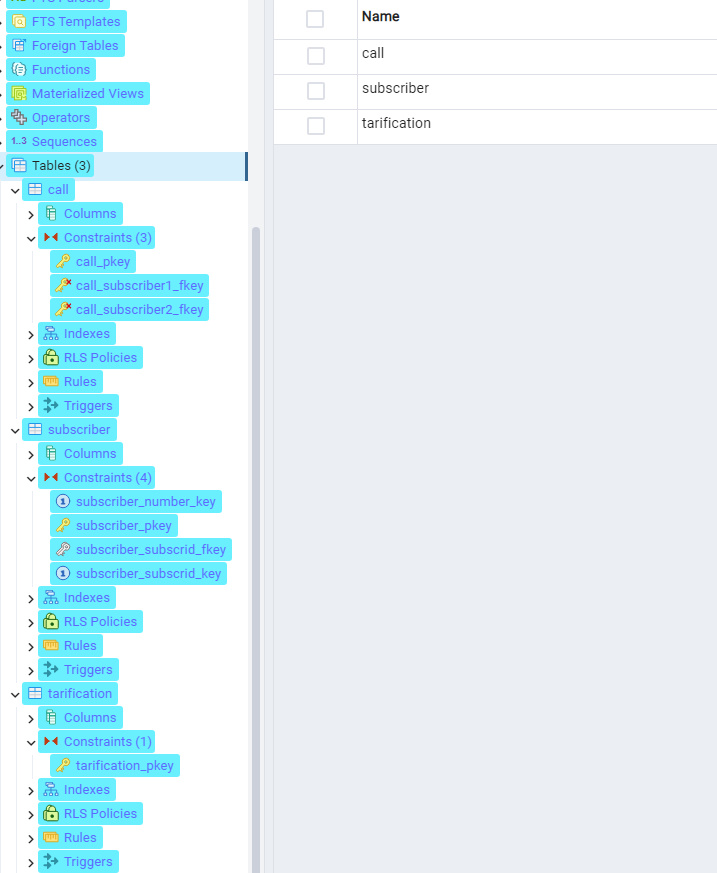
* Таблиці знаходяться в першій нормальній формі
* Кожен не первинний атрибут залежить від первинного ключа цілком та повністю (в таблиці немає складених первинних ключів)

Схема не відповідає 3НФ, тому що для таблиці tariffication не виконується наступна умова:

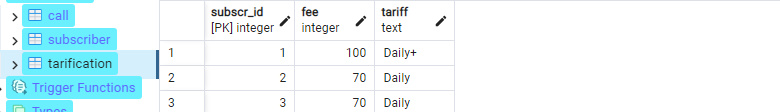
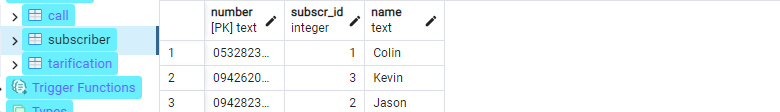
* Кожен не первинний атрибут має бути нетранзитивно залежним від кожного ключа

Можливим рішенням цієї проблеми для приведення цієї таблиці до 3НФ є створення додаткової таблиці fee, в яку буде переміщено атрибут fee, а атрибут subscriber буде зовнішнім ключем

**Фізична модель БД «Кінотеатр» у pgAdmin4**



**Фотографії вмісту таблиць**



**SQL-текст опису БД «Телефонний вузол»**

-- Database: postgres

-- DROP DATABASE IF EXISTS postgres;

**CREATE** **DATABASE** postgres

**WITH**

**OWNER** = postgres

**ENCODING** = 'UTF8'

LC\_COLLATE = 'English\_United States.1252'

LC\_CTYPE = 'English\_United States.1252'

TABLESPACE = pg\_default

**CONNECTION** **LIMIT** = -**1**

IS\_TEMPLATE = **False**;

**COMMENT** **ON** **DATABASE** postgres

**IS** 'default administrative connection database';

-- SCHEMA: public

-- DROP SCHEMA IF EXISTS public ;

**CREATE** **SCHEMA** IF **NOT** **EXISTS** **public**

**AUTHORIZATION** postgres;

**COMMENT** **ON** **SCHEMA** **public**

**IS** 'standard public schema';

**GRANT** **ALL** **ON** **SCHEMA** **public** **TO** **PUBLIC**;

**GRANT** **ALL** **ON** **SCHEMA** **public** **TO** postgres;

-- Table: public.tarification

-- DROP TABLE IF EXISTS public.tarification;

**CREATE** **TABLE** IF **NOT** **EXISTS** **public**.tarification

(

subscr\_id integer **NOT** **NULL**,

fee integer,

tariff text **COLLATE** pg\_catalog."default",

**CONSTRAINT** tarification\_pkey **PRIMARY** **KEY** (subscr\_id)

)

**WITH** (

**OIDS** = **FALSE**

)

TABLESPACE pg\_default;

**ALTER** **TABLE** IF **EXISTS** **public**.tarification

**OWNER** **to** postgres;

-- Table: public.subscriber

-- DROP TABLE IF EXISTS public.subscriber;

**CREATE** **TABLE** IF **NOT** **EXISTS** **public**.subscriber

(

"number" text **COLLATE** pg\_catalog."default" **NOT** **NULL**,

subscr\_id integer,

name text **COLLATE** pg\_catalog."default",

**CONSTRAINT** subscriber\_pkey **PRIMARY** **KEY** ("number"),

**CONSTRAINT** subscriber\_number\_key **UNIQUE** ("number"),

**CONSTRAINT** subscriber\_subscrid\_key **UNIQUE** (subscr\_id),

**CONSTRAINT** subscriber\_subscrid\_fkey **FOREIGN** **KEY** (subscr\_id)

**REFERENCES** **public**.tarification (subscr\_id) **MATCH** **SIMPLE**

**ON** **UPDATE** **NO** ACTION

**ON** **DELETE** **NO** ACTION

)

**WITH** (

**OIDS** = **FALSE**

)

TABLESPACE pg\_default;

**ALTER** **TABLE** IF **EXISTS** **public**.subscriber

**OWNER** **to** postgres;

-- Table: public.call

-- DROP TABLE IF EXISTS public.call;

**CREATE** **TABLE** IF **NOT** **EXISTS** **public**.**call**

(

call\_id integer **NOT** **NULL**,

subscriber1 text **COLLATE** pg\_catalog."default",

subscriber2 integer,

start\_time time **without** time **zone**,

end\_time time **without** time **zone**,

**CONSTRAINT** call\_pkey **PRIMARY** **KEY** (call\_id),

**CONSTRAINT** call\_subscriber1\_fkey **FOREIGN** **KEY** (subscriber1)

**REFERENCES** **public**.subscriber ("number") **MATCH** **SIMPLE**

**ON** **UPDATE** **NO** ACTION

**ON** **DELETE** **NO** ACTION

**NOT** **VALID**,

**CONSTRAINT** call\_subscriber2\_fkey **FOREIGN** **KEY** (subscriber2)

**REFERENCES** **public**.subscriber (subscr\_id) **MATCH** **SIMPLE**

**ON** **UPDATE** **NO** ACTION

**ON** **DELETE** **NO** ACTION

**NOT** **VALID**

)

**WITH** (

**OIDS** = **FALSE**

)

TABLESPACE pg\_default;

**ALTER** **TABLE** IF **EXISTS** **public**.**call**

**OWNER** **to** postgres;