

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

# Кафедра системного програмування і спеціалізованих комп’ютерних систем

**Лабораторна робота №** 1

з дисципліни «Бази даних і засоби управління»

Тема: **«Проектування бази даних та ознайомлення з базовими операціями СУБД PostgreSQL»**

Виконав: студент ІІІ курсу

ФПМ групи КВ-91

Дріф Руслан Кхеліфович

Перевірив: Павловський В.І.

Київ 2021

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.

* **Модель учбової предметної області «IT-компанія» (diagrams.net)**

Обрана предметна галузь являє собою структуру продуктової IT-компанії.

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

1. Project (проєкт) — проєкт компанії, має такі атрибути: кількість користувачів, рік відкриття, та назву;
2. Position (позиція в компанії) — позиція, яку займає робітник. Має атрибути — заробітна плата та робочі години;
3. Team (команда розробників) — команда що розробляє якийсь проєкт;
4. Buyer (покупець) — покупець, що купує проєкт;
5. Project order (замовлення проекту) — замовлення проєкт;
6. Employee (робітник) з вкладеним атрибутом ПІБ та номером паспорту.

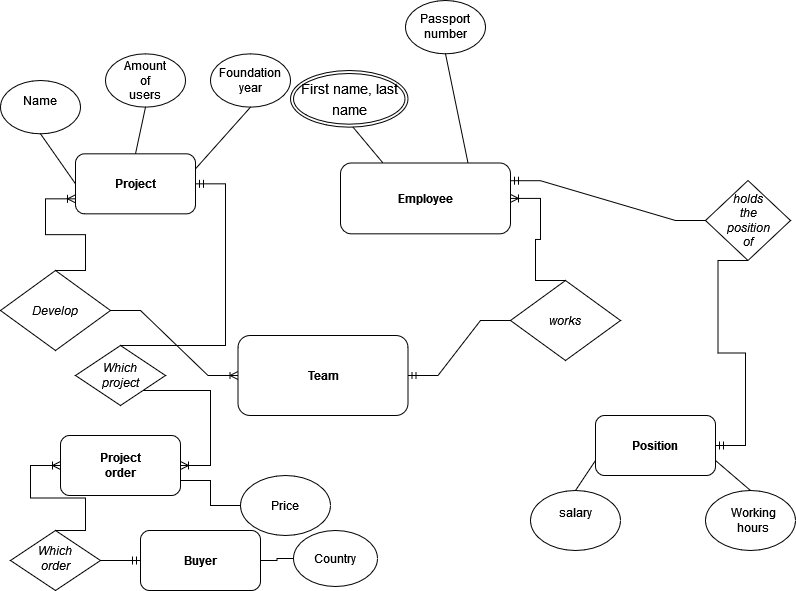


Рисунок 1 - ER-діаграма, побудована за нотацією Чена (інструмент: draw.io)

**Опис зв'язків**

1. Сутність “Команда” має зв'язок 1:N з сутністю “Працівник”, оскільки в команді є багато працівників;
2. Сутність “проєкт” має зв'язок M:N з сутністю “команда”, оскільки один проєкт може підтримувати декілька команд, і одна команда може розробляти декілька проєктів;
3. Сутність “проєкт” має зв'язок N:M з сутністю “Покупець”, оскільки один покупець може придбати декілька проєктів, а у одного проєкта може бути декілька покупців;
4. Сутність «Робітник» має зв’язок 1:1 з сутністю «позиція», оскільки один розробник може обіймати лише одну позицію в компанії.
5. Сутність-зв’язок "Замовлення проекту" (Project order) пов’язує сутності “Buyer” та “Project”, так як вони мають зв’язок M:N.

**Перетворення моделі у схему бази даних**

На рисунку 2: сутність “проєкт” було перетворено у таблицю “Project”, сутність “Покупець” було перетворено у таблицю “Buyer”.

Було створено таблицю “Development”,яка реалізовує зв’язок M:N між сутностями “проєкт” та “команда”.

Сутність “замовлення проекту” було перетворено у таблицю “project order”.

Сутність “робітник” було перетворено у таблицю “Employee”.

Сутність “команда” було перетворено у таблицю “Team”.

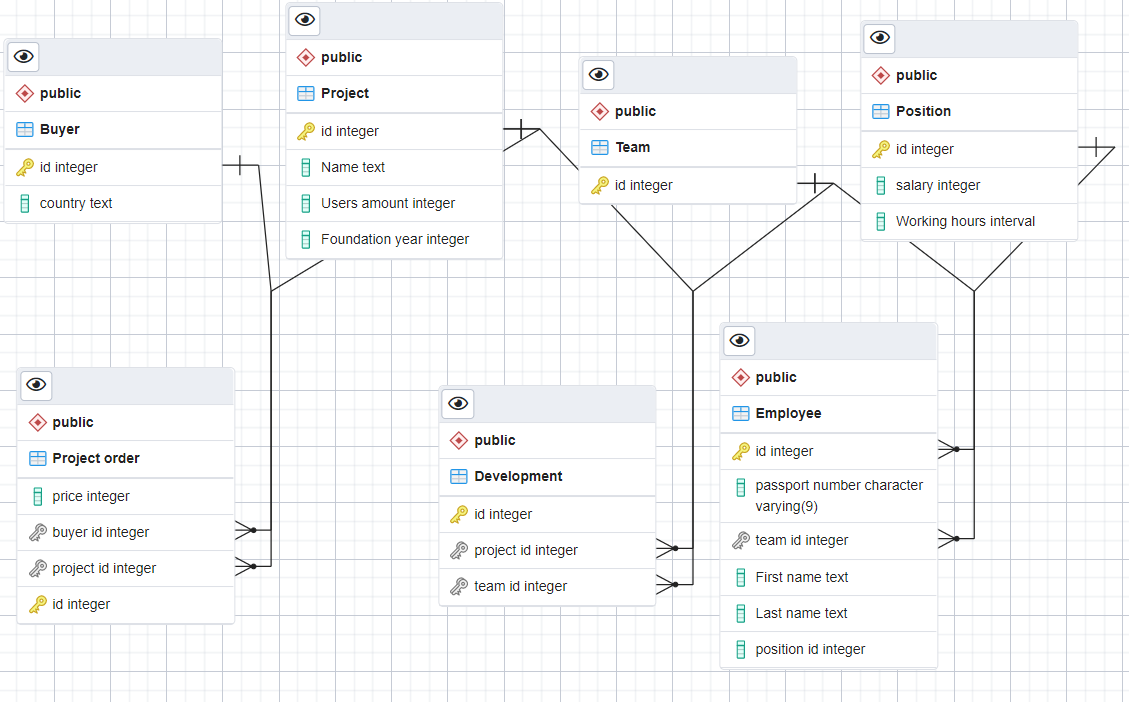


Рисунок 2 - Схема бази даних

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

В таблиці 1 наведено опис таблиць бази даних та їх атрибутів.

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибути | Типи атрибутів |
| **Buyer** | **id** – Primary key покупця, не NULL  **country** — країна, не NULL | **integer**  **text** |
| **Project** | **id** – Primary key проєкту, не NULL  **name** — назва проекту, не NULL  **users amount** – кількість користувачів, не NULL  **foundation year** — рік заснування, не NULL | **integer**  **text**  **integer**  **integer** |
| **Project order** | **price** – ціна замовлення, не NULL  **buyer id** — ідентифікатор замовника, Foreign key, не NULL  **project id –** ідентифікатор проєкту, не NULL  **id —** Primary key замовлення, не NULL | **integer**  **integer**  **integer**  **integer** |
| **Team** | **Id — і**дентифікатор команди, Primary key, не NULL | **integer** |
| **Employee** | **Id —** ідентифікатор розробника, Primary key, не NULL  **passport number –** номер паспорту, не NULL  **team id —** ідентифікатор команди в якій працює розробник, може бути NULL  **first name -** ім'я, не NULL  **last name -** прізвище, не NULL  **position id —** ідентифікатор позиції в компанії, Foreign key не NULL | **integer**  **character varying**  **integer**  **text**  **text**  **integer** |
| **Position** | **Id —** ідентифікатор позиції, Primary key, не NULL  **salary –** ЗП на позиції, не NULL  **working hours —** робочі години, не NULL | **integer**  **integer**  **interval** |
|  |  |  |
| **Development** | **project identifier –** ідентифікатор проєкту  **team identifier –** ідентифікатор команди | **integer**  **integer** |

Таблиця 1 – Опис об’єктів бази даних

**Функціональні залежності для кожної таблиці**

**Buyer**

*Primary key*: id

country — країна, з якої родом покупець, (або де заснована компанія-покупець), залежить від ідентифікатору покупця.

**Project**

*Primary key*: id

name – ім'я проєкту, залежить номеру проекту

amount of users — кількість користувачів проєкту, залежить від ідентифікатору проєкту

foundation year — рік заснування, у всіх проєктів різний, залежить від ідентифікатору проєкту

**Project** **order**

*Primary key*: id

price – ціна, залежить від номеру замовлення

**Team**

*Primary key*: id

**Employee**

*Primary key*: id

passport number – номер паспорту, у кожного робітника різний, а тому залежить від уникального ідентифікатора

team id — ідентифікатор команди, в якій працює розробник, залежить від ідентифікатору розробника

first name — ім'я розробника, залежить від його ідентифікатору

last name — прізвище розробника, залежить від його ідентифікатору

position id — ідентифікатор прозиції в компанії, на якій знаходиться розробник, залежить від його ідентифікатору

**Position**

*Primary key*: id

salary — заробітна плата, залежить він унікального ідентифікатору позиції

working hours — робочі години, залежить він унікального ідентифікатору позиції

**Development**

*Primary key:* id

Project identifier – ідентифікатор проекту, залежить від унікального ідентифікатору

Team identifier – ідентифікатор команди, залежить від унікального ідентифікатору

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

Схема відповідає 1НФ, тому що:

1. В таблиці немає дубльованих рядків.
2. В кожній комірці зберігається атомарне (скалярне) значення.
3. В кожному стовпці зберігаються дані одного типу.

Схема відповідає 2НФ, тому що:

1. Вона відповідає 1НФ.
2. Має первинний ключ, а всі не ключові стовпці таблиці залежать від первинного ключа.

Схема відповідає 3НФ, тому що:

1. Вона відповідає 2НФ.
2. Всі не ключові атрибути таблиці залежать винятково від усього первинного ключа, а не його частини. Тобто кожен неключовий атрибут не транзитивно (без посередника) залежить від первинного ключа.

**Таблиці бази даних у pgAdmin4**

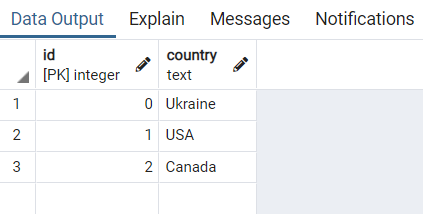
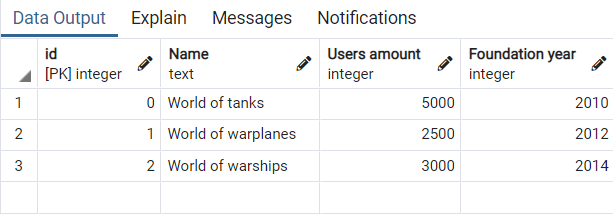
Рисунок 3 - таблиця “Buyer” ****

Рисунок 4 - таблиця “Project” ****

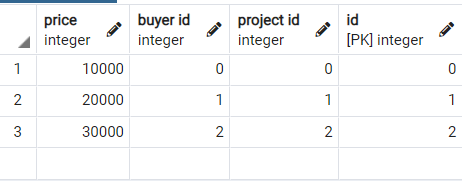


Рисунок 5 - таблиця “Project order”

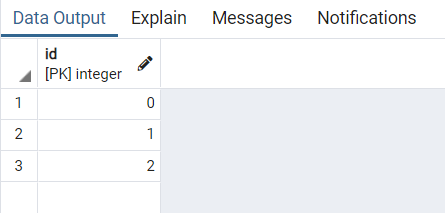


Рисунок 6 – таблиця “Team”

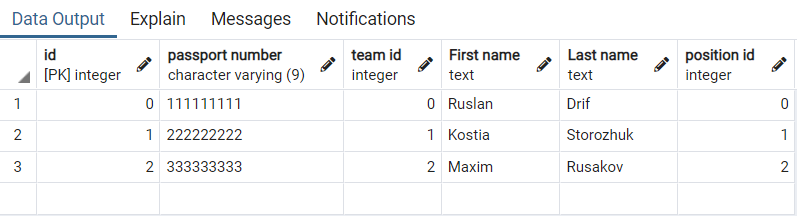
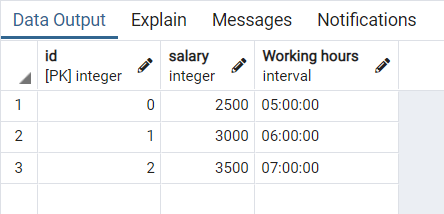
Рисунок 7 - таблиця “Employee” ****

Рисунок 8 - таблиця “Position”****

**SQL-текст опису БД**

SET statement\_timeout = 0;

SET lock\_timeout = 0;

SET idle\_in\_transaction\_session\_timeout = 0;

SET client\_encoding = 'UTF8';

SET standard\_conforming\_strings = on;

SELECT pg\_catalog.set\_config('search\_path', '', false);

SET check\_function\_bodies = false;

SET xmloption = content;

SET client\_min\_messages = warning;

SET row\_security = off;

SET default\_tablespace = '';

SET default\_table\_access\_method = heap;

CREATE TABLE public."Buyer" (

id integer NOT NULL,

country text NOT NULL

);

ALTER TABLE public."Buyer" OWNER TO postgres;

CREATE TABLE public."Development" (

id integer NOT NULL,

"project id" integer NOT NULL,

"team id" integer NOT NULL

);

ALTER TABLE public."Development" OWNER TO postgres;

CREATE TABLE public."Employee" (

id integer NOT NULL,

"passport number" character varying(9) NOT NULL,

"team id" integer,

"First name" text NOT NULL,

"Last name" text NOT NULL,

"position id" integer

);

ALTER TABLE public."Employee" OWNER TO postgres;

CREATE TABLE public."Position" (

id integer NOT NULL,

salary integer NOT NULL,

"Working hours" interval NOT NULL

);

ALTER TABLE public."Position" OWNER TO postgres;

CREATE TABLE public."Project" (

id integer NOT NULL,

"Name" text NOT NULL,

"Users amount" integer NOT NULL,

"Foundation year" integer

);

ALTER TABLE public."Project" OWNER TO postgres;

CREATE TABLE public."Project order" (

price integer NOT NULL,

"buyer id" integer NOT NULL,

"project id" integer NOT NULL,

id integer NOT NULL

);

ALTER TABLE public."Project order" OWNER TO postgres;

CREATE TABLE public."Team" (

id integer NOT NULL

);

ALTER TABLE public."Team" OWNER TO postgres;

ALTER TABLE ONLY public."Buyer"

ADD CONSTRAINT "Customer\_pkey" PRIMARY KEY (id);

ALTER TABLE ONLY public."Development"

ADD CONSTRAINT "Development\_pkey" PRIMARY KEY (id);

ALTER TABLE ONLY public."Employee"

ADD CONSTRAINT "Employee\_pkey" PRIMARY KEY (id);

ALTER TABLE ONLY public."Position"

ADD CONSTRAINT "Position\_pkey" PRIMARY KEY (id);

ALTER TABLE ONLY public."Project order"

ADD CONSTRAINT "Project order\_pkey" PRIMARY KEY (id);

ALTER TABLE ONLY public."Project"

ADD CONSTRAINT "Project\_pkey" PRIMARY KEY (id);

ALTER TABLE ONLY public."Team"

ADD CONSTRAINT "Team\_pkey" PRIMARY KEY (id);

CREATE INDEX "fki\_customer id" ON public."Project order" USING btree ("buyer id");

CREATE INDEX "fki\_ordering project id" ON public."Project order" USING btree ("project id");

CREATE INDEX "fki\_project identifier" ON public."Development" USING btree ("project id");

CREATE INDEX "fki\_team id" ON public."Employee" USING btree ("team id");

CREATE INDEX "fki\_team identifier" ON public."Development" USING btree ("team id");

ALTER TABLE ONLY public."Project order"

ADD CONSTRAINT "buyer id" FOREIGN KEY ("buyer id") REFERENCES public."Buyer"(id) NOT VALID;

ALTER TABLE ONLY public."Project order"

ADD CONSTRAINT "ordering project id" FOREIGN KEY ("project id") REFERENCES public."Project"(id) NOT VALID;

ALTER TABLE ONLY public."Employee"

ADD CONSTRAINT "position id" FOREIGN KEY (id) REFERENCES public."Position"(id) NOT VALID;

ALTER TABLE ONLY public."Development"

ADD CONSTRAINT "project identifier" FOREIGN KEY ("project id") REFERENCES public."Project"(id) NOT VALID;

ALTER TABLE ONLY public."Employee"

ADD CONSTRAINT "team id" FOREIGN KEY ("team id") REFERENCES public."Team"(id) NOT VALID;

ALTER TABLE ONLY public."Development"

ADD CONSTRAINT "team identifier" FOREIGN KEY ("team id") REFERENCES public."Team"(id) NOT VALID;