

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 групи КВ-23

Паламарчук Максим

Перевірив: Петрашенко А.В.

Київ – 2024

*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.

**Опис предметної області «Система управління та аналізу даних в галузі робототехніки»**

Ця предметна область передбачає управління виробництвом пристроїв і аналіз даних про їхні компоненти та закупівлі.

**Опис сутностей предметної області**

1. Фабрика (factory) має атрибути: код фабрики, назва фабрики, спеціалізацію, адресу. Ця сутність описує фабрику та її місцезнаходження.

2. Пристрій (device) має атрибути: код пристрою, назва пристрою, задача, операційна система, код фабрики, дата виготовлення. Ця сутність описує пристрої, які виготовляє фабрика.

3. Комплектуючі (components) має атрибути: код компонента, назва компонента, вага, категорія, код пристрою. Ця сутність описує комплектуючі, які замовляє фабрика для виготовлення пристроїв.

4. Купівля (buy) має атрибути: код компонента, код фабрики, дата купівлі, ціна. Ця сутність описує купівлю комплектуючих заводом.

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

Сутність factory має зв’язок 1:M по відношенню до сутності device, оскільки один завод може виготовити багато пристроїв, але один пристрій може бути виготовлений тільки однією фабрикою.

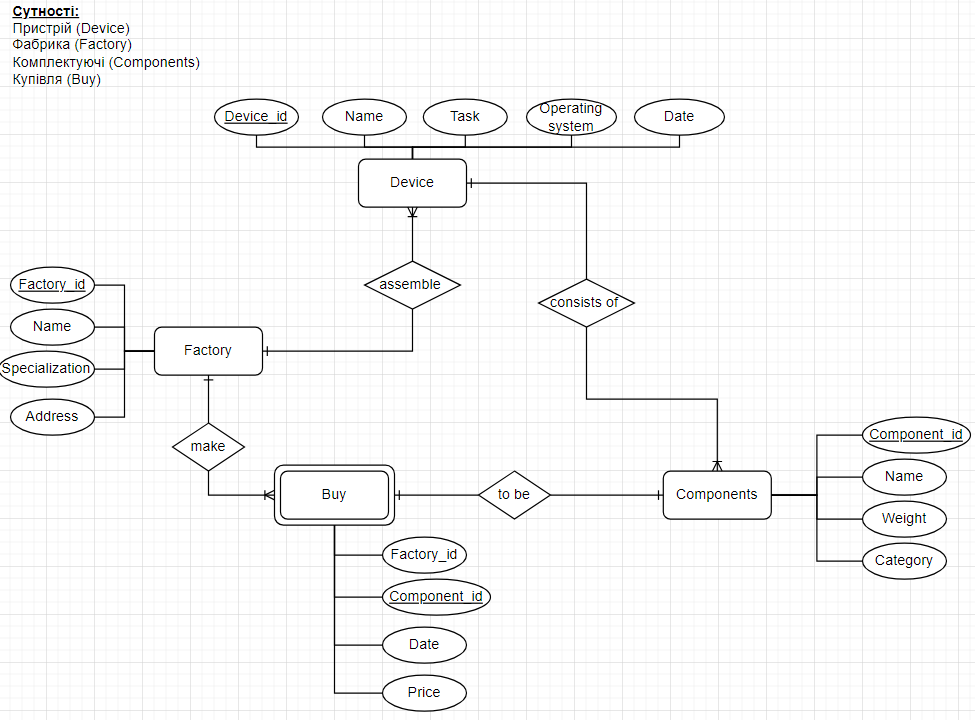
Сутність device має зв’язок 1:M по відношенню до сутності components, оскільки один пристрій може бут зібраний із багатьох комплектуючих, але один компонент не може бути в декількох пристроях.

Сутність components має зв’язок 1:1 по відношенню до сутності buy, оскільки один компонент може бути куплений лише один раз.

Сутність factory має зв’язок 1:M по відношенню до сутності buy, оскільки одна фабрика може зробити багато покупок, але одна купівля може відноситися лише до однієї фабрики.

**Концептуальна модель предметної області**

**(нотація “Пташиної лапки”)**



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

Сутність Factory перетворено у таблицю factory.

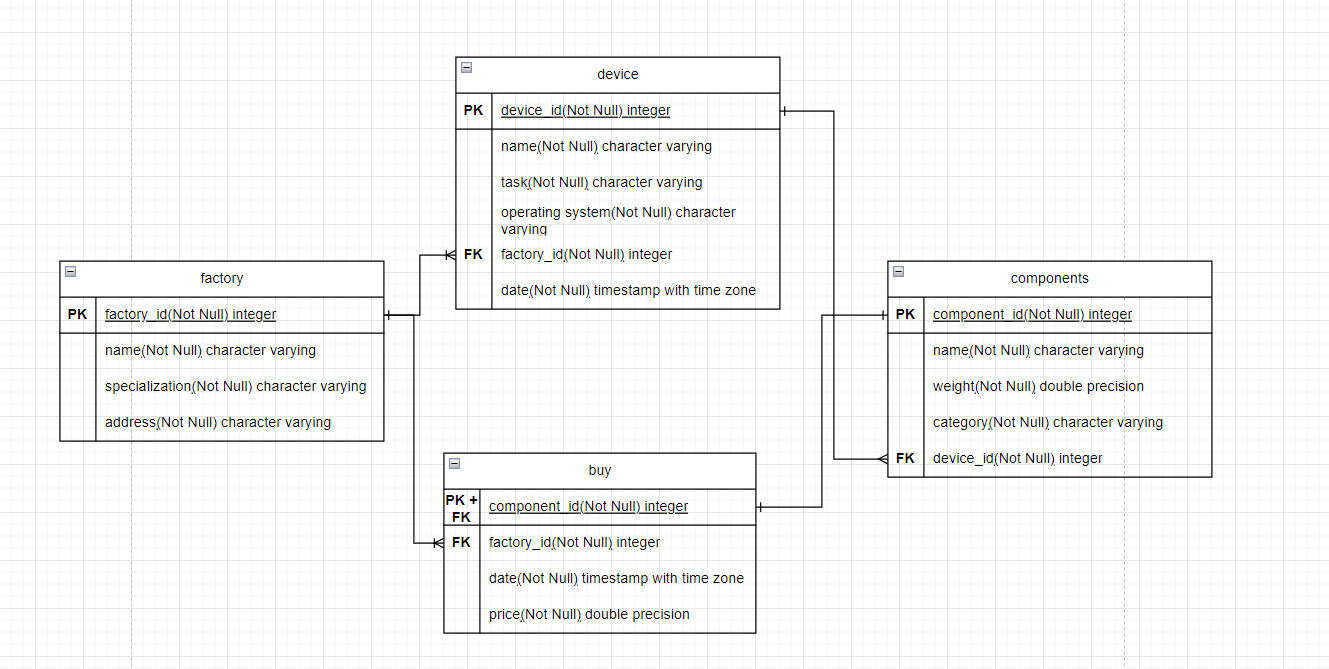
Сутність Device перетворено у таблицю device.

Сутність Components перетворено у таблицю components.

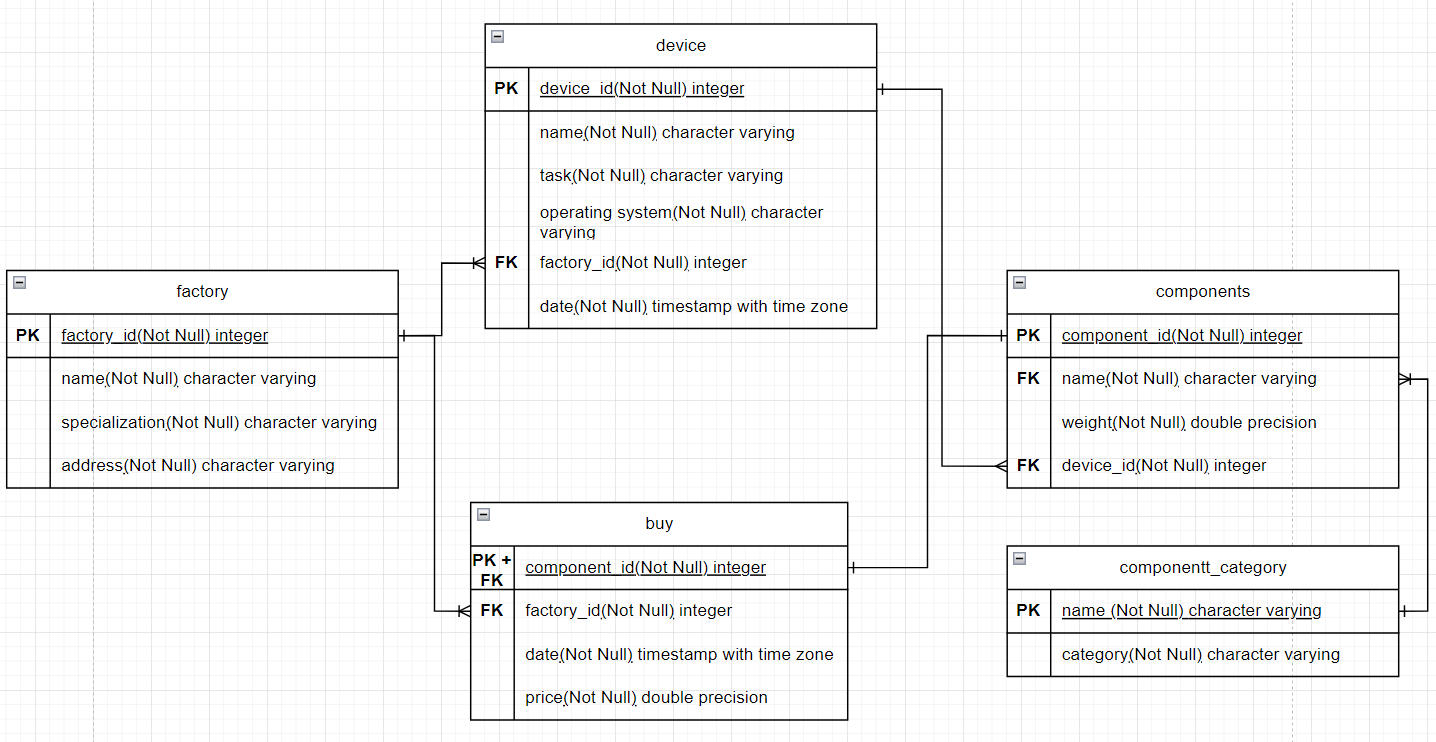
Сутність Buy перетворено у таблицю buy.

**Логічна модель (схема) “Система управління та аналізу даних в галузі робототехніки”**

До нормалізації



Після нормалізації



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

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип атрибуту |
| **factory** – має інформацію про фабрику | **factory\_id –** унікальний ідентифікатор фабрики. Не допускає NULL.  **name –** назва фабрики. Не допускає NULL.  **specialization –** спеціалізація фабрики. Не допускає NULL.  **address –** адреса фабрики. Не допускає NULL. | **integer** (числовий)  **character varying** (рядок)  **character varying** (рядок)  **character varying** (рядок) |
| **device** – має інформацію про пристрій | **device\_id –** унікальний ідентифікатор пристрою. Не допускає NULL.  **name –** назва пристрою. Не допускає NULL.  **task** – задача пристрою. Не допускає NULL.  **operating system** – операційна система пристрою. Не допускає NULL.  **factory\_id –** унікальний ідентифікатор фабрики. Не допускає NULL.  **date** – дата виготовлення пристрою. Не допускає NULL. | **integer** (числовий)  **character varying** (рядок)  **character varying** (рядок)  **character varying** (рядок)  **integer** (числовий)  **timestamp with time zone** (дата та час) |
| **components** – має інформацію про комплектуючі | **component\_id –** унікальний ідентифікатор компонента. Не допускає NULL.  **name –** назва компонента. Не допускає NULL.  **weight** – вага компонента. Не допускає NULL.  **category** – категорія компонента. Не допускає NULL. (при нормалізації цей атрибут зникає із цієї таблиці)  **device\_id –** унікальний ідентифікатор пристрою. | **integer** (числовий)  **character varying** (рядок)  **double precision**  (число з плаваючою комою)  **character varying** (рядок)  **integer** (числовий) |
| **components\_category –** має інформацію про категорію компонента | **name –** назва компонента (PK). Не допускає NULL.  **category** – категорія компонента. Не допускає NULL. | **character varying** (рядок)  **character varying** (рядок) |
| **buy** – має інформацію про здійснення купівлі комплектуючих | **component \_id –** унікальний ідентифікатор компонента. Не допускає NULL.  **factory \_id –** унікальний ідентифікатор фабрики. Не допускає NULL.  **date** – дата купівлі комплектуючих. Не допускає NULL.  **price –** ціна комплектуючих | **integer** (числовий)  **integer** (числовий)  **timestamp with time zone** (дата та час)  **double precision**  (число з плаваючою комою) |

**Функціональні залежності для кожної таблиці**

**factory:**

factory\_id → name, specialization, address

factory\_id → name

factory\_id → specialization

factory\_id → address

address → name

factory\_id → address → name

~~name → address~~

~~factory\_id → name → address~~

Ця таблиця задовольняє першу нормальну форму, оскільки рядки не повторюються, і кожне значення є атомарним.

Задовольняє другу нормальну форму автоматично, оскільки є лише один первинний ключ, і всі не ключові атрибути залежать від цього ключа.

І задовольняє третю нормальну форму, хоч є транзитивний зв’язок, оскільки address є кандидатом в ключі.

**device:**

device\_id → name, task, operating system, date, factory\_id

device\_id → name

device\_id → task

device\_id → operating system

device\_id → date

device\_id → factory\_id

Задовольняє всі 3 нормальні форми.

**components:**

component\_id → name, weight, category, device\_id

component\_id → weight,

component\_id → category

component\_id → device\_id

name → category

component\_id → name → category

~~category → name~~

~~component\_id → category → name~~

Задовольняє перші дві нормальні форми, проте є транзитивний зв’язок component\_id → name → category. name – не є кандидатом в ключі. Тому потрібна нормалізація.

Таблиця components розділяється на таблицю **components** (component\_id, name, weight, device\_id) і **component\_category** (name, category). Тепер атрибут name – це зовнішній ключ в таблиці components і первинний ключ в таблиці component\_category.

Сутність components має зв’язок M:1 по відношенню до сутності component\_category, оскільки є багато комплектуючих з одним іменем, які можуть бути купленими чи зібраними в пристрої може, але до одного імені прив’язана тільки одна категорія.

**buy:**

component\_id → factory\_id, date, price

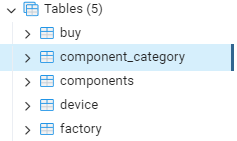
component\_id → factory\_id

component\_id → date

component\_id → price

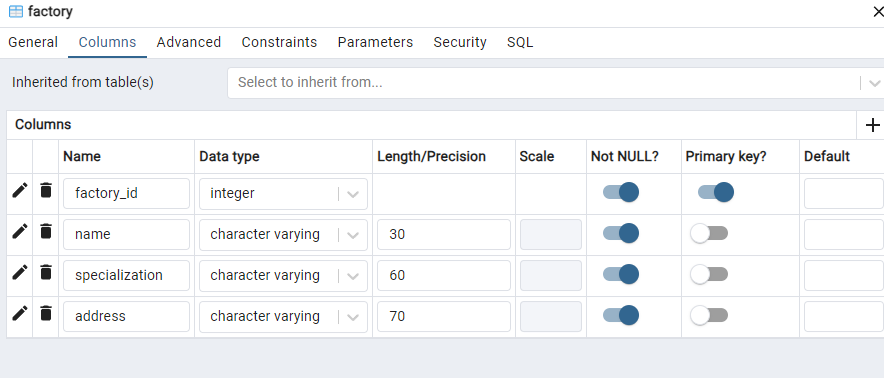
Задовольняє всі 3 нормальні форми.

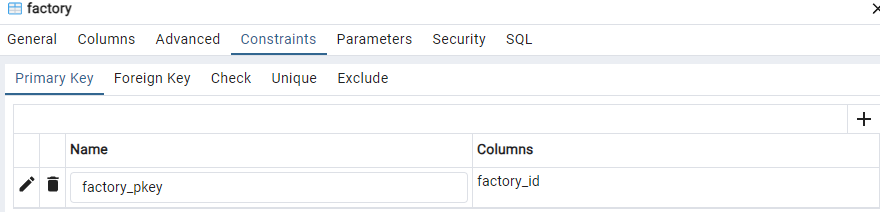
**Таблиці бази даних у pgAdmin4**

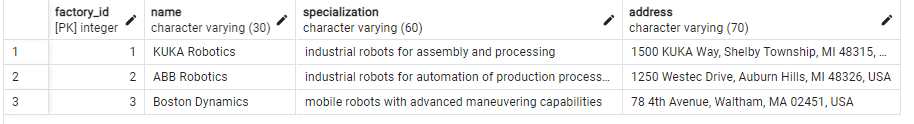


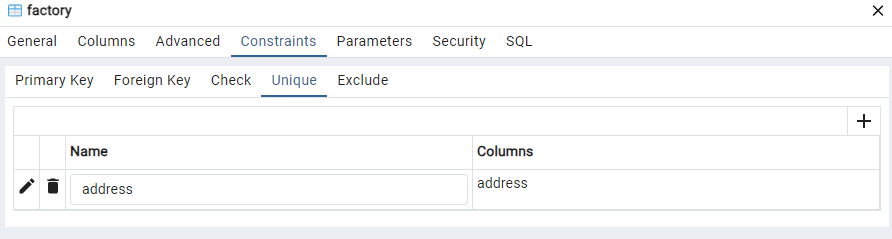
**Вміст таблиць**

**Factory**

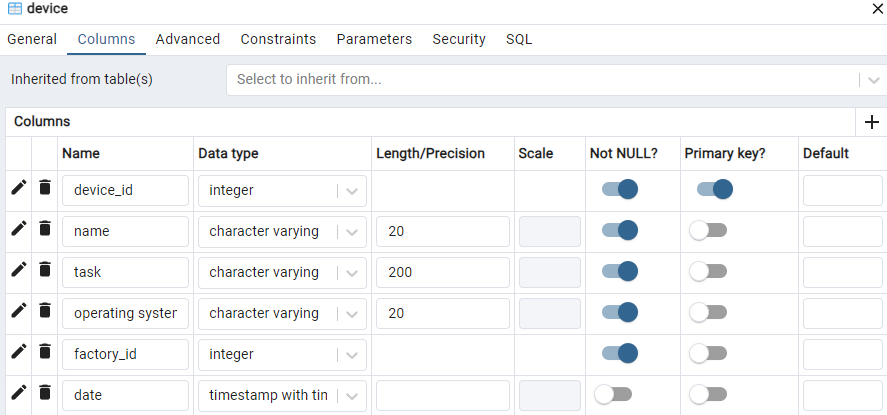
****

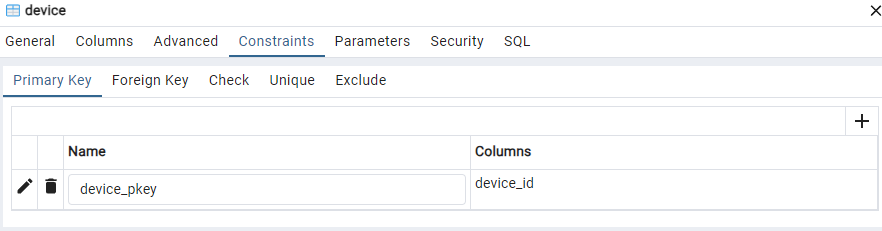


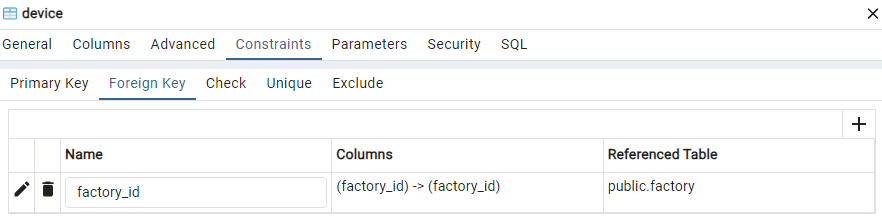


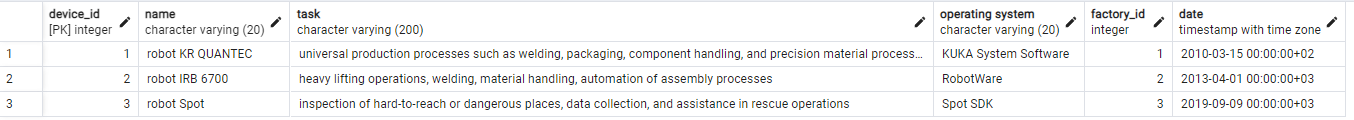


**device**

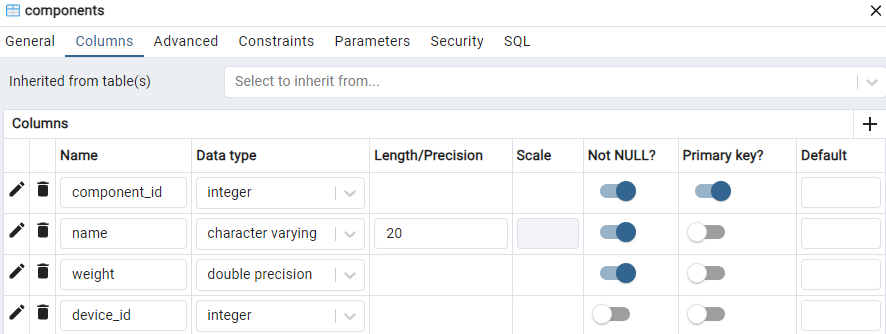
****

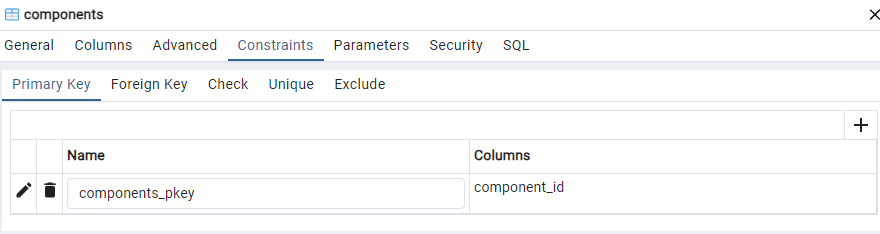
****

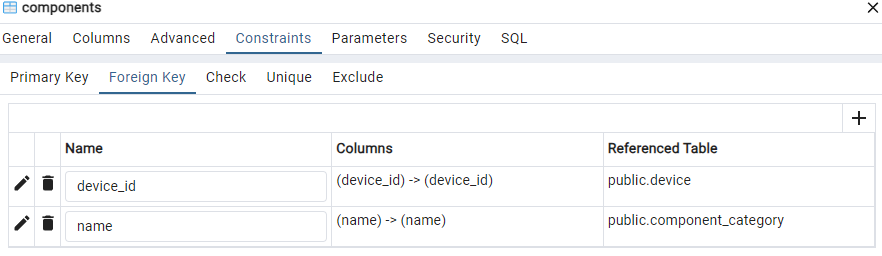
****

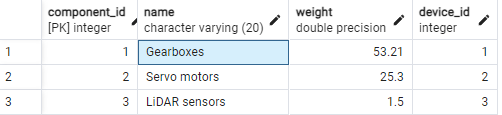
****

**components**

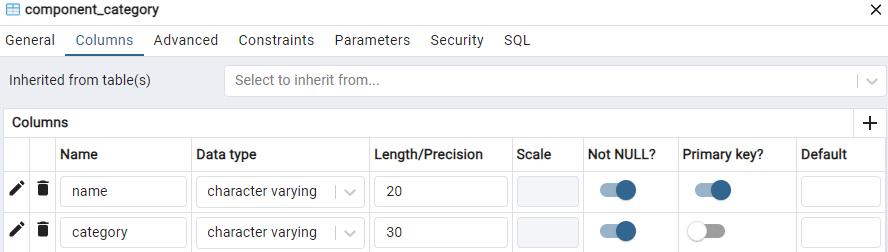
****

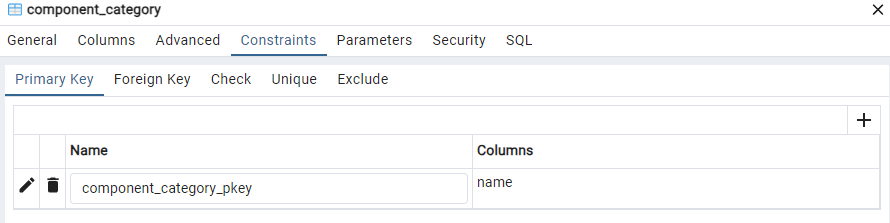
****

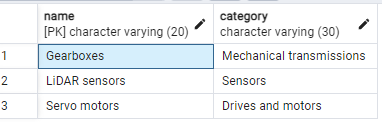
****

****

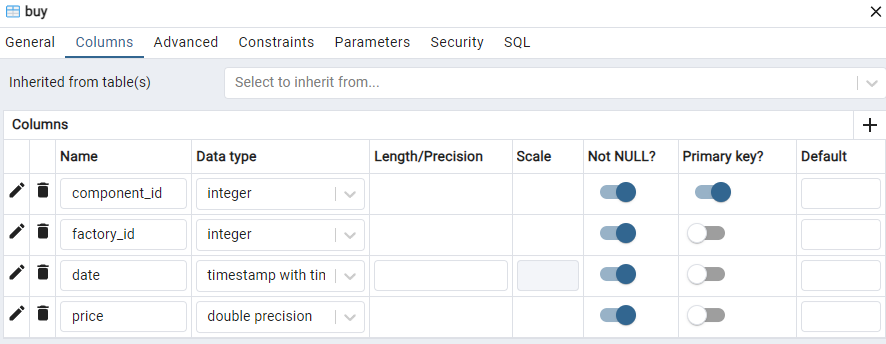
**component\_category**

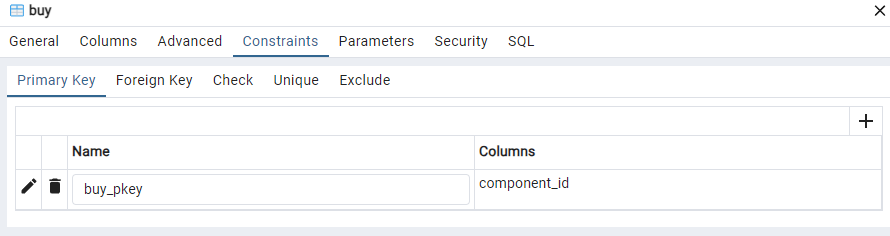
****

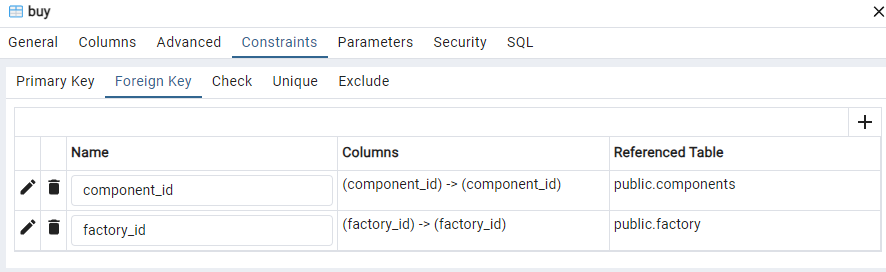
****

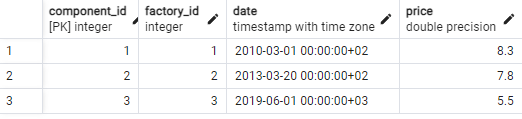
****

**buy**

****

****

****

****

**Посилання на репозиторій:** [**https://github.com/HomerLund/DB\_Lab1.git**](https://github.com/HomerLund/DB_Lab1.git)

**SQL текст опису БД**

--

-- PostgreSQL database dump

--

-- Dumped from database version 16.4

-- Dumped by pg\_dump version 16.4

-- Started on 2024-10-04 19:18:15

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;

--

-- TOC entry 218 (class 1259 OID 16424)

-- Name: buy; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.buy (

component\_id integer NOT NULL,

factory\_id integer NOT NULL,

date timestamp with time zone NOT NULL,

price double precision NOT NULL

);

ALTER TABLE public.buy OWNER TO postgres;

--

-- TOC entry 219 (class 1259 OID 16490)

-- Name: component\_category; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.component\_category (

name character varying(20) NOT NULL,

category character varying(30) NOT NULL

);

ALTER TABLE public.component\_category OWNER TO postgres;

--

-- TOC entry 217 (class 1259 OID 16414)

-- Name: components; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.components (

component\_id integer NOT NULL,

name character varying(20) NOT NULL,

weight double precision NOT NULL,

device\_id integer

);

ALTER TABLE public.components OWNER TO postgres;

--

-- TOC entry 216 (class 1259 OID 16404)

-- Name: device; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.device (

device\_id integer NOT NULL,

name character varying(20) NOT NULL,

task character varying(200) NOT NULL,

"operating system" character varying(20) NOT NULL,

factory\_id integer NOT NULL,

date timestamp with time zone

);

ALTER TABLE public.device OWNER TO postgres;

--

-- TOC entry 215 (class 1259 OID 16399)

-- Name: factory; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.factory (

factory\_id integer NOT NULL,

name character varying(30) NOT NULL,

specialization character varying(60) NOT NULL,

address character varying(70) NOT NULL

);

ALTER TABLE public.factory OWNER TO postgres;

--

-- TOC entry 4866 (class 0 OID 16424)

-- Dependencies: 218

-- Data for Name: buy; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.buy (component\_id, factory\_id, date, price) FROM stdin;

3 3 2019-06-01 00:00:00+03 5.5

1 1 2010-03-01 00:00:00+02 8.3

2 2 2013-03-20 00:00:00+02 7.8

\.

--

-- TOC entry 4867 (class 0 OID 16490)

-- Dependencies: 219

-- Data for Name: component\_category; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.component\_category (name, category) FROM stdin;

LiDAR sensors Sensors

Servo motors Drives and motors

Gearboxes Mechanical transmissions

\.

--

-- TOC entry 4865 (class 0 OID 16414)

-- Dependencies: 217

-- Data for Name: components; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.components (component\_id, name, weight, device\_id) FROM stdin;

2 Servo motors 25.3 2

3 LiDAR sensors 1.5 3

1 Gearboxes 53.21 1

\.

--

-- TOC entry 4864 (class 0 OID 16404)

-- Dependencies: 216

-- Data for Name: device; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.device (device\_id, name, task, "operating system", factory\_id, date) FROM stdin;

3 robot Spot inspection of hard-to-reach or dangerous places, data collection, and assistance in rescue operations Spot SDK 3 2019-09-09 00:00:00+03

2 robot IRB 6700 heavy lifting operations, welding, material handling, automation of assembly processes RobotWare 2 2013-04-01 00:00:00+03

1 robot KR QUANTEC universal production processes such as welding, packaging, component handling, and precision material processing. KUKA System Software 1 2010-03-15 00:00:00+02

\.

--

-- TOC entry 4863 (class 0 OID 16399)

-- Dependencies: 215

-- Data for Name: factory; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.factory (factory\_id, name, specialization, address) FROM stdin;

1 KUKA Robotics industrial robots for assembly and processing 1500 KUKA Way, Shelby Township, MI 48315, USA

3 Boston Dynamics mobile robots with advanced maneuvering capabilities 78 4th Avenue, Waltham, MA 02451, USA

2 ABB Robotics industrial robots for automation of production processes 1250 Westec Drive, Auburn Hills, MI 48326, USA

\.

--

-- TOC entry 4704 (class 2606 OID 16515)

-- Name: factory address; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.factory

ADD CONSTRAINT address UNIQUE (address);

--

-- TOC entry 4712 (class 2606 OID 16474)

-- Name: buy buy\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.buy

ADD CONSTRAINT buy\_pkey PRIMARY KEY (component\_id);

--

-- TOC entry 4714 (class 2606 OID 16498)

-- Name: component\_category component\_category\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.component\_category

ADD CONSTRAINT component\_category\_pkey PRIMARY KEY (name);

--

-- TOC entry 4710 (class 2606 OID 16418)

-- Name: components components\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.components

ADD CONSTRAINT components\_pkey PRIMARY KEY (component\_id);

--

-- TOC entry 4708 (class 2606 OID 16408)

-- Name: device device\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.device

ADD CONSTRAINT device\_pkey PRIMARY KEY (device\_id);

--

-- TOC entry 4706 (class 2606 OID 16403)

-- Name: factory factory\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.factory

ADD CONSTRAINT factory\_pkey PRIMARY KEY (factory\_id);

--

-- TOC entry 4718 (class 2606 OID 16477)

-- Name: buy component\_id; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.buy

ADD CONSTRAINT component\_id FOREIGN KEY (component\_id) REFERENCES public.components(component\_id) NOT VALID;

--

-- TOC entry 4716 (class 2606 OID 16419)

-- Name: components device\_id; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.components

ADD CONSTRAINT device\_id FOREIGN KEY (device\_id) REFERENCES public.device(device\_id);

--

-- TOC entry 4715 (class 2606 OID 16409)

-- Name: device factory\_id; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.device

ADD CONSTRAINT factory\_id FOREIGN KEY (factory\_id) REFERENCES public.factory(factory\_id);

--

-- TOC entry 4719 (class 2606 OID 16466)

-- Name: buy factory\_id; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.buy

ADD CONSTRAINT factory\_id FOREIGN KEY (factory\_id) REFERENCES public.factory(factory\_id) NOT VALID;

--

-- TOC entry 4717 (class 2606 OID 16509)

-- Name: components name; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.components

ADD CONSTRAINT name FOREIGN KEY (name) REFERENCES public.component\_category(name) NOT VALID;

-- Completed on 2024-10-04 19:18:15

--

-- PostgreSQL database dump complete

--