НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра автоматизованих систем обробки інформації і управління

**КУРСОВА РОБОТА**

з дисципліни: «Бази даних»

на тему: «ІС Магазин комп’ютерних комплектуючих»

Студента 2 курсу, групи ІП-92

напряму підготовки «ІПЗ»

спеціальності 121

Глечковського Богдана Валентиновича

Керівник ст. вик. Недашківський Є. А.

Київ - 2020 рік

Національний технічний університет України “КПІ”

Кафедра автоматизованих систем обробки інформації і управління

Дисципліна Бази даних

Курс 2 Група ІП-92 Семестр 1

**ЗАВДАННЯ**

**на курсову роботу студентів**

|  |
| --- |
| **Глечковського Богдана Валентиновича** |
|  |

(прізвище, ім’я, по батькові)

|  |  |
| --- | --- |
| 1. Тема роботи | ІС «Магазин комп’ютерних комплектуючих» |
|  | |
|  | |

|  |  |
| --- | --- |
| 2. Строк здачі студентами закінченої роботи | «30» грудня 2020 р. |

|  |
| --- |
| 3. Технічне завдання:  Створено реляційну базу даних у третій нормальній формі. Структура узгоджена з керівником. База включає щонайменше шість таблиць з не менш ніж десятьма рядками у кожній для репрезентації даних. Базу даних доповнює застосунок, що дозволяє вивести на екран вміст однієї з таблиць.  Застосунок дозволяє проводити фільтрацію виведення. Додана можливість посторінкового виведення даних задавши кількість рядків на сторінку та можливість обрати номер сторінки. Передбачено інтерфейс користувача.  Реалізована модифікація даних (додавання, редагування, видалення) через інтерфейс користувача.  Структура проекту на рівні бази даних містить наступні об’єкти: представлення (view), табличну та скалярну функції, збережену процедуру, тригер, принаймні одна з таблиць передбачає значення «за замовчанням», принаймні в одному з вище описаних об’єктів присутні оператори JOIN, CASE, UNION.  Передбачено окремого користувача для застосунку, що не матиме прав на редагування структури бази даних, проте матиме досить прав для роботи. |

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)

|  |
| --- |
| Вступ, постановка задач, теоретичні відомості, опис програмнго забезпечення, |
| інструкція користувача, висновок, перелік посилань |
|  |
|  |

**Зміст**

[Вступ 4](#_Toc59964204)

[Постановка задачі 6](#_Toc59964205)

[Проектування бази даних 7](#_Toc59964206)

[Опис програмного забезпечення 11](#_Toc59964207)

[Висновок 20](#_Toc59964208)

[Література 21](#_Toc59964209)

[Додатки 22](#_Toc59964210)

# **Вступ**

Темою курсової роботи є проектування бази даних для магазину комп’ютерних комплектуючих та створення додатку для виводу змісту бази даних на екран користувача. Для зручності управління магазином повинна існувати інформаційна система в якій зберігаються усі необхідні дані для автоматизації процесів отримання та обробки замовлень, інформація про клієнтів та працівників цього магазину, дані про наявність певних товарів на кожному з складів магазину, інформація про товари, їх виробників та дані про категорії цих товарів.

Дана інформаційна система є однією з найпопулярніших причин необхідності бази даних. На прикладі даної курсової роботи буде розглянуто проектування бази даних для зберігання й обробки інформації, використання сучасних інформаційних технологій для організації збору, обробки й зберігання даних, створення додатку для управління базою даних. Мета курсової роботи - створення бази даних для магазину комп’ютерних комплектуючих, розробка додатку для управління створеною базою даних.

Причина вибору саме цієї теми, як зазначено вище, популярність питання розробки інформаційної системи для зберігання необхідної інформації звичайних або онлайн магазинів.

Для виконання даної курсової роботи була розроблена реляційна база даних, структура якої узгоджена із викладачем. Для бази даних використана мова PostgreSQL, а також застосунок із інтерфейсом користувача, мова написання якого JavaScript.

Створений застосунок виконує наступні задачі: відображення змісту кожної з таблиць, фільтрація відображених даних, редагування, додавання та видалення записів з відображених таблиць.

**Анотація**

Курсова робота на тему «Магазин комп’ютерних комплектуючих» виконана з метою закріплення, поглиблення та узагальнення знань, які отримані під час вивчення дисципліни «Організація баз даних», втілення практичних навичок в області створення сучасних систем управління базами даних з оформленням відповідного текстового, програмного та ілюстративного матеріалу.

В курсовій роботі розроблена ER-діаграма предметної області «Hardware Shop» обрана СУБД Postgres і засоби розробки «VS Code» для реалізації заданих функцій. Створена база даних «Hardware Shop», що складається з 8 таблиць і заповнена даними, близькими до реальних умов використання інформаційної системи.

# **Постановка задачі**

Поставлена задача була визначена наступним чином: створити реляційну базу даних у 3 нормальній формі. Створити 8 таблиць, дані яких будуть відповідати обраній темі. Заповнити створені таблиці даними, близькими до реальних даних за умов використання інформаційної системи.

Створити застосунок за допомогою якого можна виводити дані бази даних у вигляді таблиць. Передбачити можливість вибору окремих таблиць для відображення даних які вони зберігають.

Створений застосунок повинен мати можливість фільтрації даних за обраним стовпцем, можливість додавання нових записів до обраної таблиці, редагування та видалення існуючих записів обраної таблиці.

Структура проекту на рівні бази даних має містити наступні об’єкти: представлення, табличну та скалярну функції, збережену процедуру, тригер, значення «за замовчанням» хоча б у одній таблиці, принаймні в одному з вище описаних об’єктів присутні оператори JOIN, CASE, UNION.

# **Проектування бази даних**

Опис сутностей:

|  |  |
| --- | --- |
| **Сутність** | **Опис сутності** |
| Order | Містить інформацію про замовлення. |
| Warehouse | Містить інформацію про кількість товарів на складі та їх ціну. |
| Hardware | Містить основну інформацію про товар. |
| Categories | Назва та опис категорій товарів. |
| Manufacturer | Інформація про виробника товару. |
| Client | Клієнти магазину, їх контакти. |
| Employee | Працівники магазину, їх контакти, місце проживання та дата народження. |

Зв’язки між сутностями:

|  |  |
| --- | --- |
| **Сутність** | **Зв’язок** |
| Order | Багато до одного із Warehouse  Один до багатьох із Client  Один до багатьох із Employee |
| Warehouse | Один до багатьох із Order  Один до одного із Hardware |
| Hardware | Один до одного із Warehouse |
| Categories | Багато до одного із Hardware |
| Manufacturer | Багато до одного із Hardware |
| Client | Багато до одного із Order |
| Employee | Багато до одного із Order |

Діаграма сутностей та бази даних

На основі опису сутностей та зв’язків між ними було побудовано дві діаграми: ER – діаграма сутностей, та DB – діаграма бази даних.

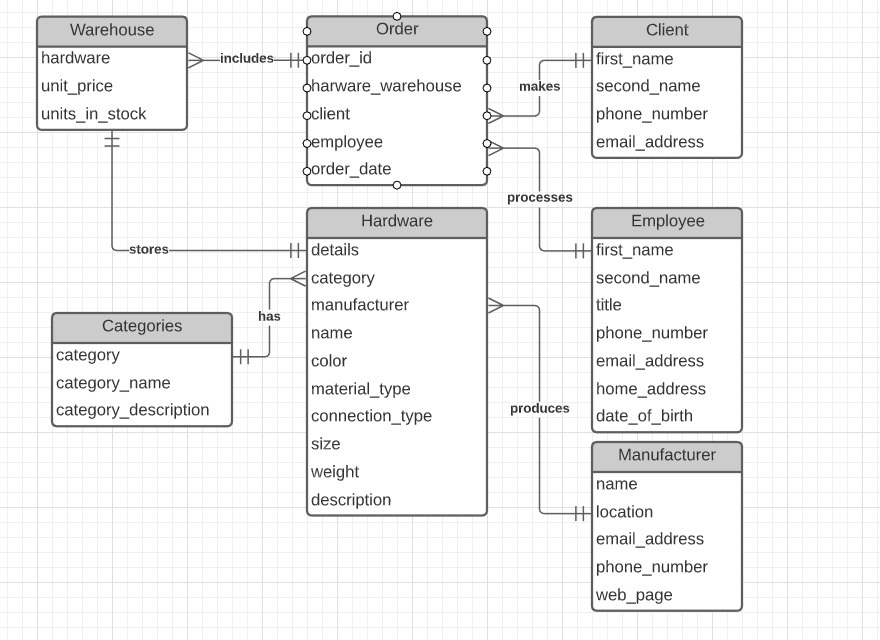


Рисунок 1. ER діаграма

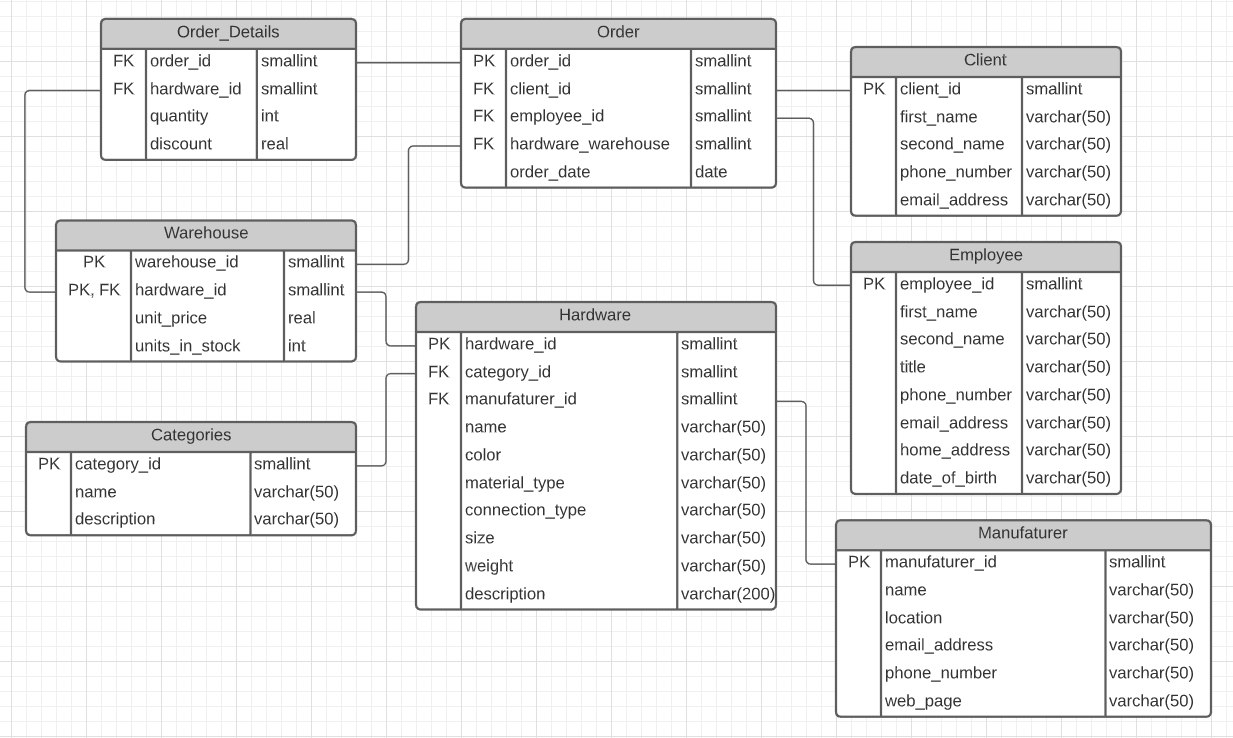


Рисунок 2. DB діаграма

# **Опис програмного забезпечення**

За умовами поставленими в розділі Постановка Задачі був реалізований наступний функціонал: перегляд всіх наявних у базі даних таблиць, фільтрація даних обраної таблиці за обраним стовпцем. Реалізовані можливості додавання запису до таблиці, редагування вже існуючих записів та видалення вже існуючих записів з бази даних.

Для роботи з базою даних були створені наступні запити:

--=========================================================================

-- Display all items that need restock (CASE + JOIN)

create or replace view items\_restock as

select name, description, units\_in\_stock,

case

when units\_in\_stock >= 5 then 'Do not need restock'

when units\_in\_stock < 5 then 'Needs restock'

end needsRestock

from hardware inner join warehouse on hardware.hardware\_id = warehouse.hardware\_id;

-- Display all clients that have email address or phone number

create or replace view clients\_to\_contact as

select \* from client where email\_address is not null or phone\_number is not null;

-- Display all people in db (UNION)

create or replace view display\_all\_people as

select first\_name, second\_name from client

union

select first\_name, second\_name from employee;

--===========================================================================

Представлення (views), де item\_restock виводить на екран інформацію про теперішній стан товарів на складі використовуючи CASE + JOIN; clients\_to\_contact виводить на екран інформацію про клієнтів які залишили дані для зв’язку, email або номер телефону; display\_all\_people виводить на екран інформацію про всіх людей записаних у базі даних використовуючи UNION.

--===========================================================================

-- Display full price of order

create or replace function order\_price(quantity int, price real, discount real)

returns real

language 'plpgsql'

as $$

begin

return @quantity \* @price \* @discount;

end;

$$;

-- Display all orders between two dates

create or replace function orders\_between(date1 date, date2 date)

returns setof "order"

language 'plpgsql'

as $$

begin

if date1 = null or date2 = null then return query select \* from "order" where order\_date = current\_date;

else return query select \* from "order" where order\_date between date1 and date2;

end if;

end;

$$;

--===========================================================================

Скалярна та таблична функції, де order\_price (скалярна) виводить на екран повну ціну замовлення враїовуючи ціну одиниці товару, загальну замовлену кількість товару та наявність знижки; orders\_between (таблична) виводить на екран інформацію про замовлення які були створені між двома заданими датами.

--===========================================================================

-- Insert new client

create or replace function addNewClient(\_first\_name varchar, \_second\_name varchar, \_phone\_number varchar, \_email\_address varchar)

returns void

language 'plpgsql'

as $$

begin

insert into client (first\_name, second\_name, phone\_number, email\_address) values (\_first\_name, \_second\_name, \_phone\_number, \_email\_address);

end;

$$;

-- Insert new employee

create or replace function addNewEmployee(\_first\_name varchar, \_second\_name varchar, \_title varchar, \_phone\_number varchar, \_email\_address varchar, \_home\_address varchar, \_date\_of\_birth date)

returns void

language 'plpgsql'

as $$

begin

insert into employee (first\_name, second\_name, title, phone\_number, email\_address, home\_address, date\_of\_birth) values (\_first\_name, \_second\_name, \_title, \_phone\_number, \_email\_address, \_home\_address, \_date\_of\_birth);

end;

$$;

--===========================================================================

Процедури, де addNewClient додає до таблиці client нову запис в яку включає задані параметри; addNewEmployee додає до таблиці employee нову запис в яку включає задані параметри.

--===========================================================================

-- Moving deleted orders to orders\_archive table

create table if not exists orders\_archive (

date\_of\_delete timestamp,

delete\_by text,

like "order"

);

create or replace function move\_to\_archive()

returns trigger

language 'plpgsql'

as $$

begin

insert into orders\_archive (order\_id, client\_id, employee\_id, hardware\_warehouse, order\_date, date\_of\_delete, delete\_by)

select old.\*, statement\_timestamp(), session\_user;

return new;

end;

$$

create trigger move\_to\_archive

before delete on "order"

for each row execute function move\_to\_archive();

--===========================================================================

Триггер move\_to\_archive який створює новий запис у таблиці orders\_archive з даними запису із таблиці order, кожен раз, коли видаляється запис з таблиці order.

--===========================================================================

-- Default value

alter table hardware

alter description set default 'No Description';

--===========================================================================

Було передбачено використання значення за замовчуванням в таблиці hardware.

Приклади роботи

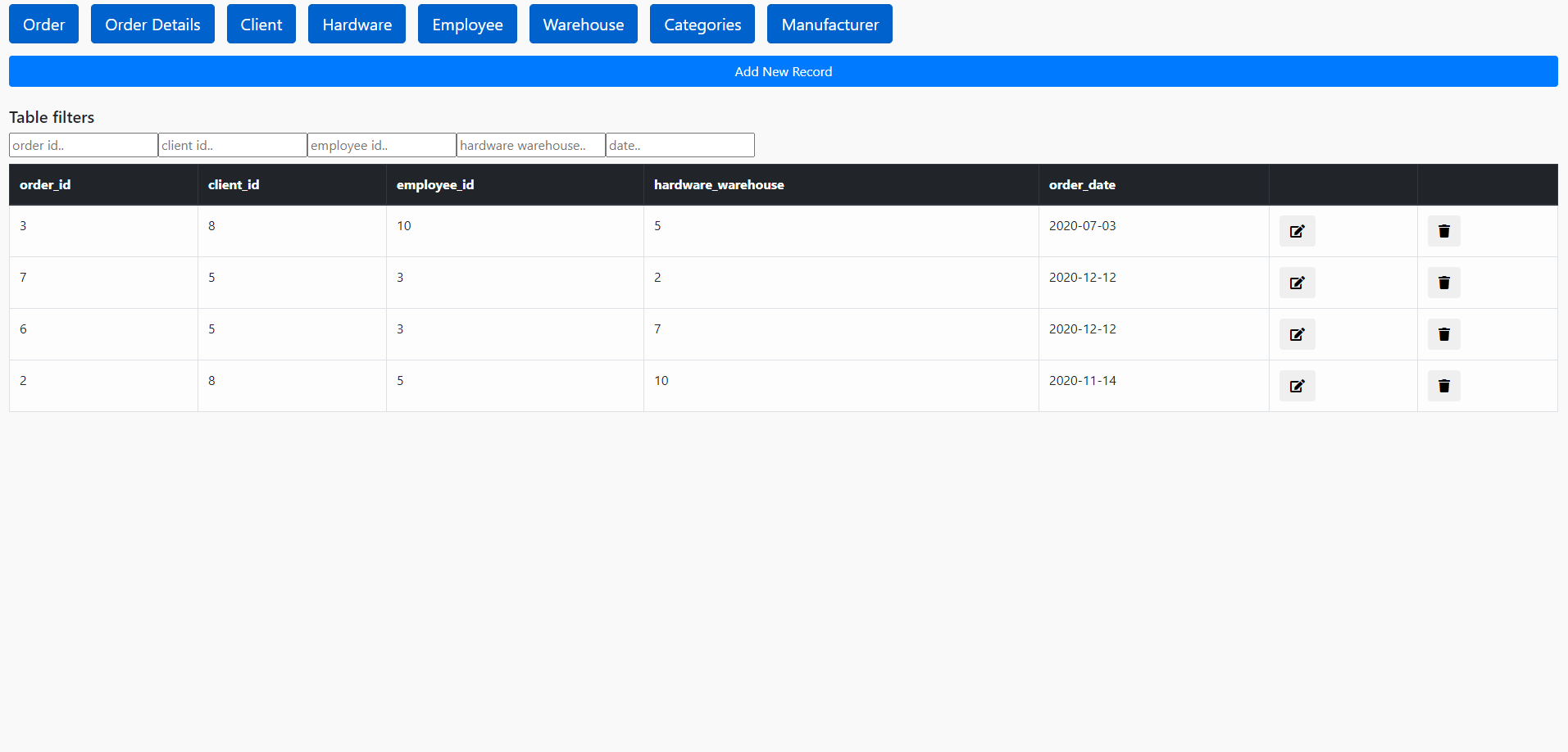


Рисунок 3. Перегляд таблиці “Order”

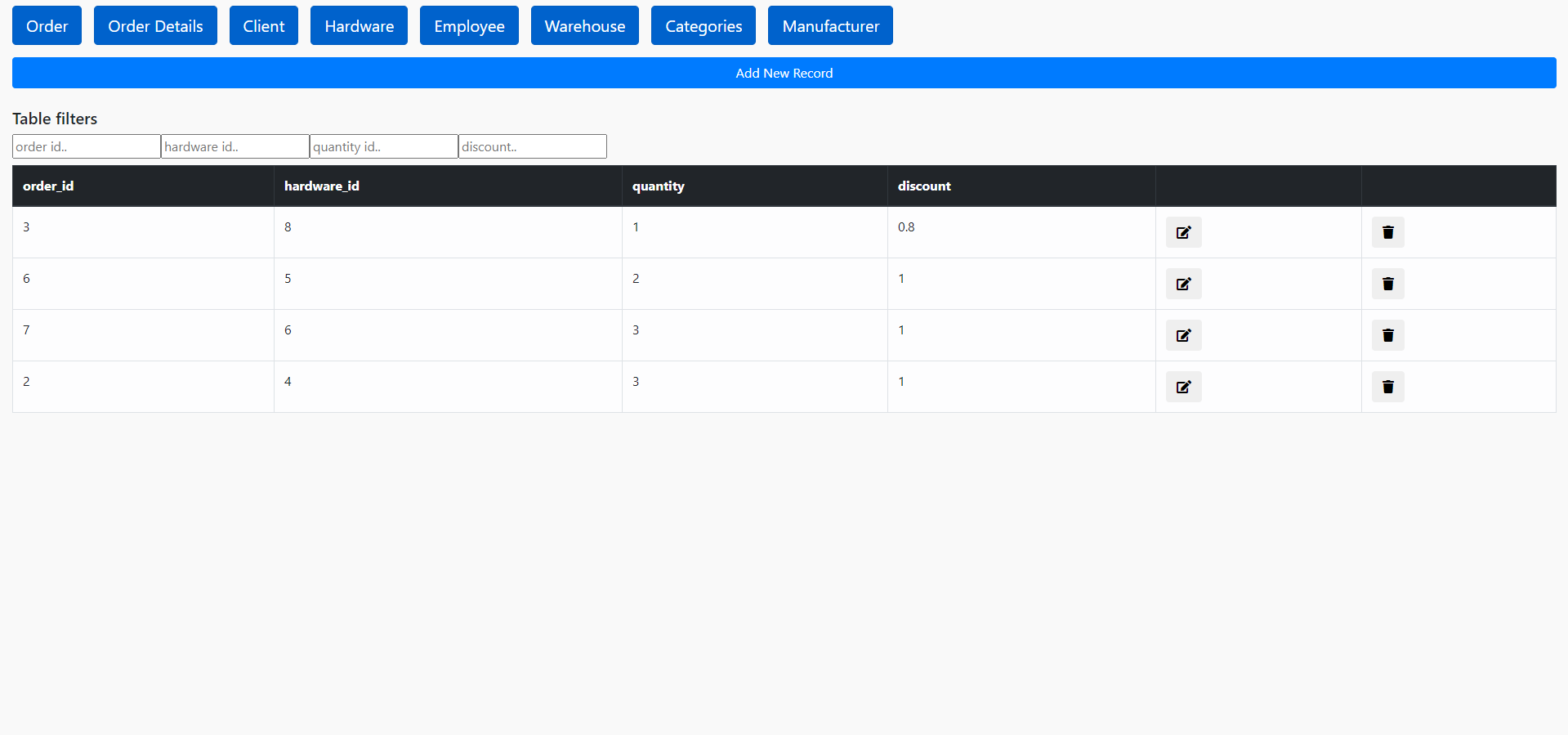


Рисунок 4. Перегляд таблиці “Order\_Details”

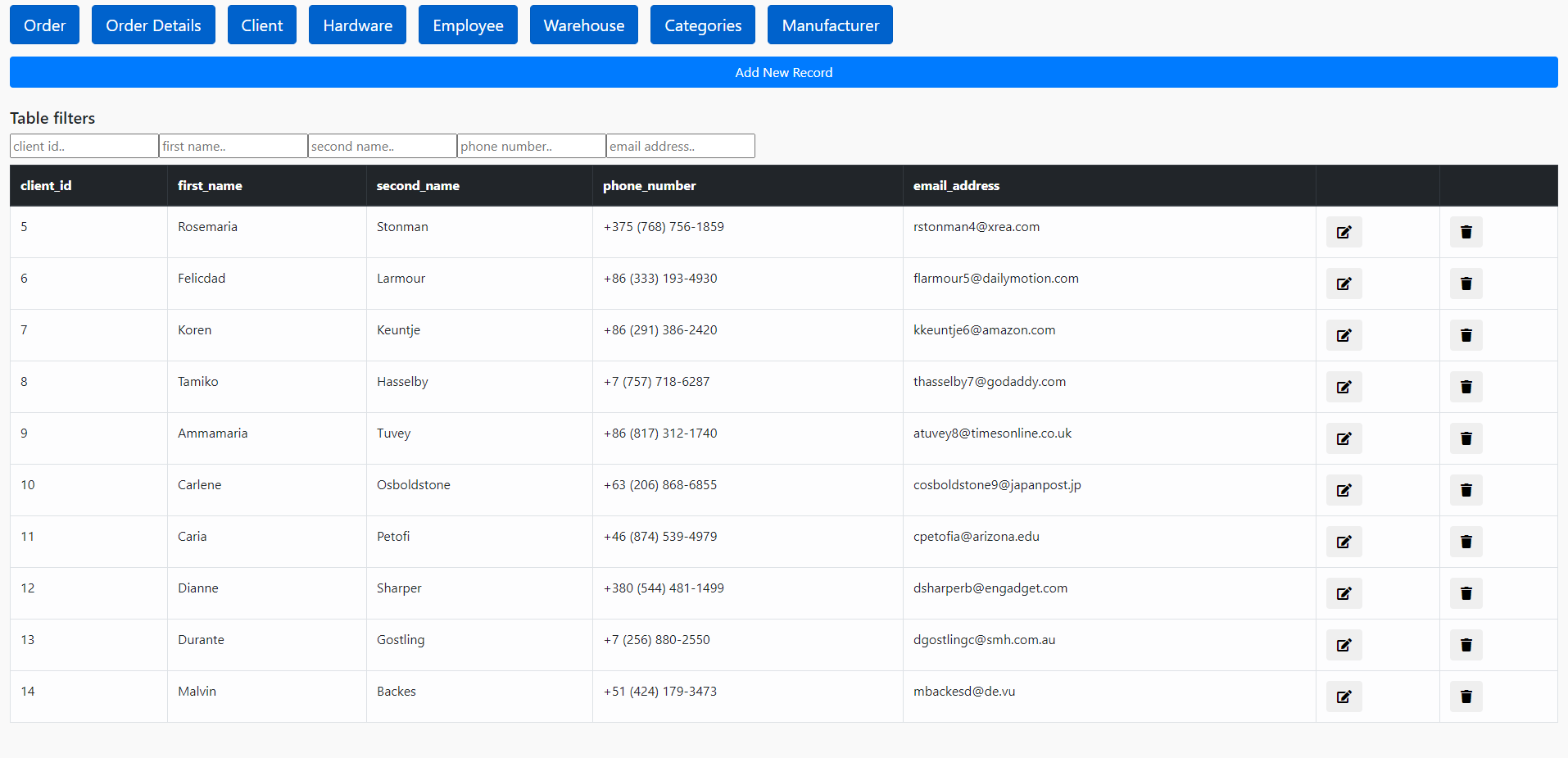


Рисунок 5. Перегляд таблиці “Client”

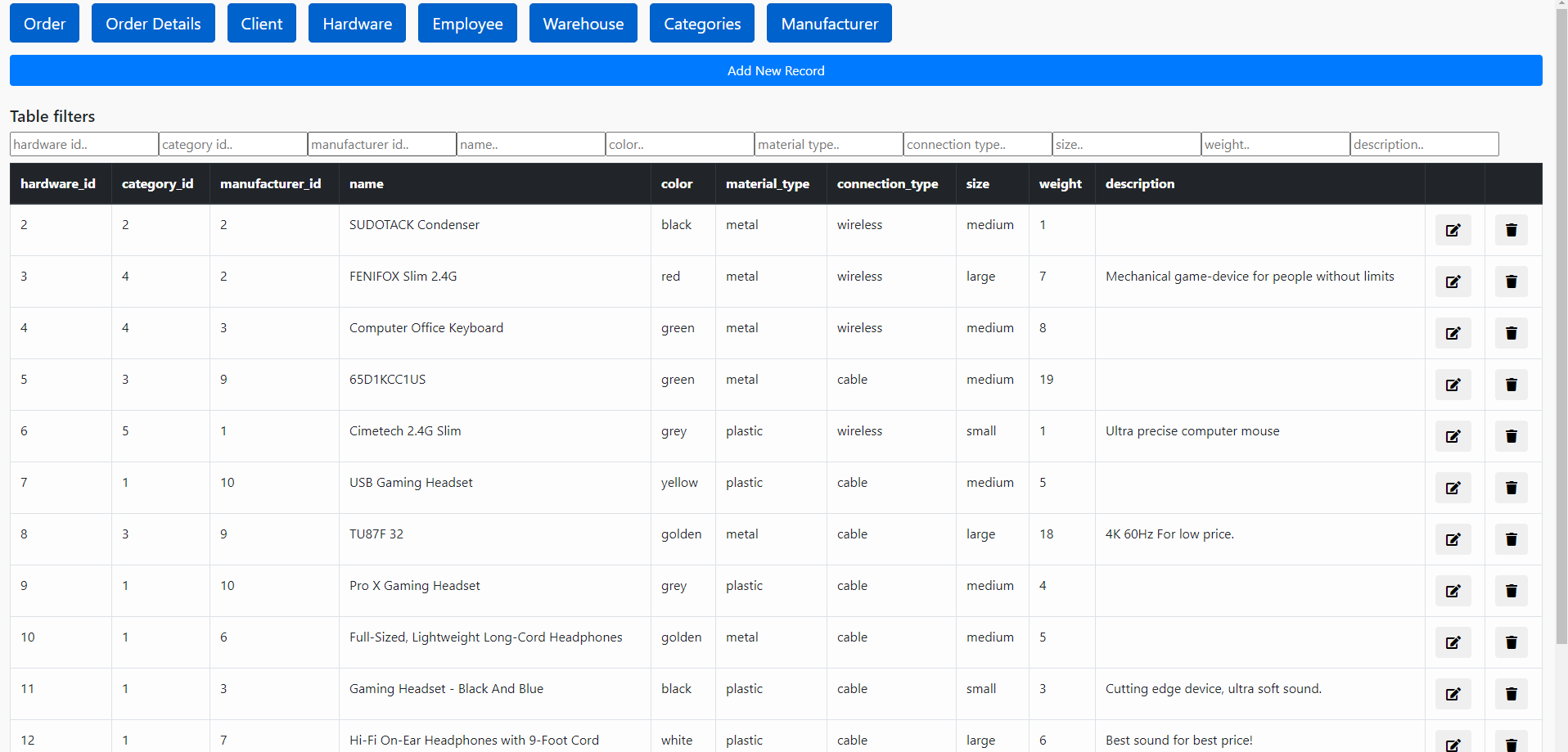


Рисунок 6. Перегляд таблиці “Hardware”

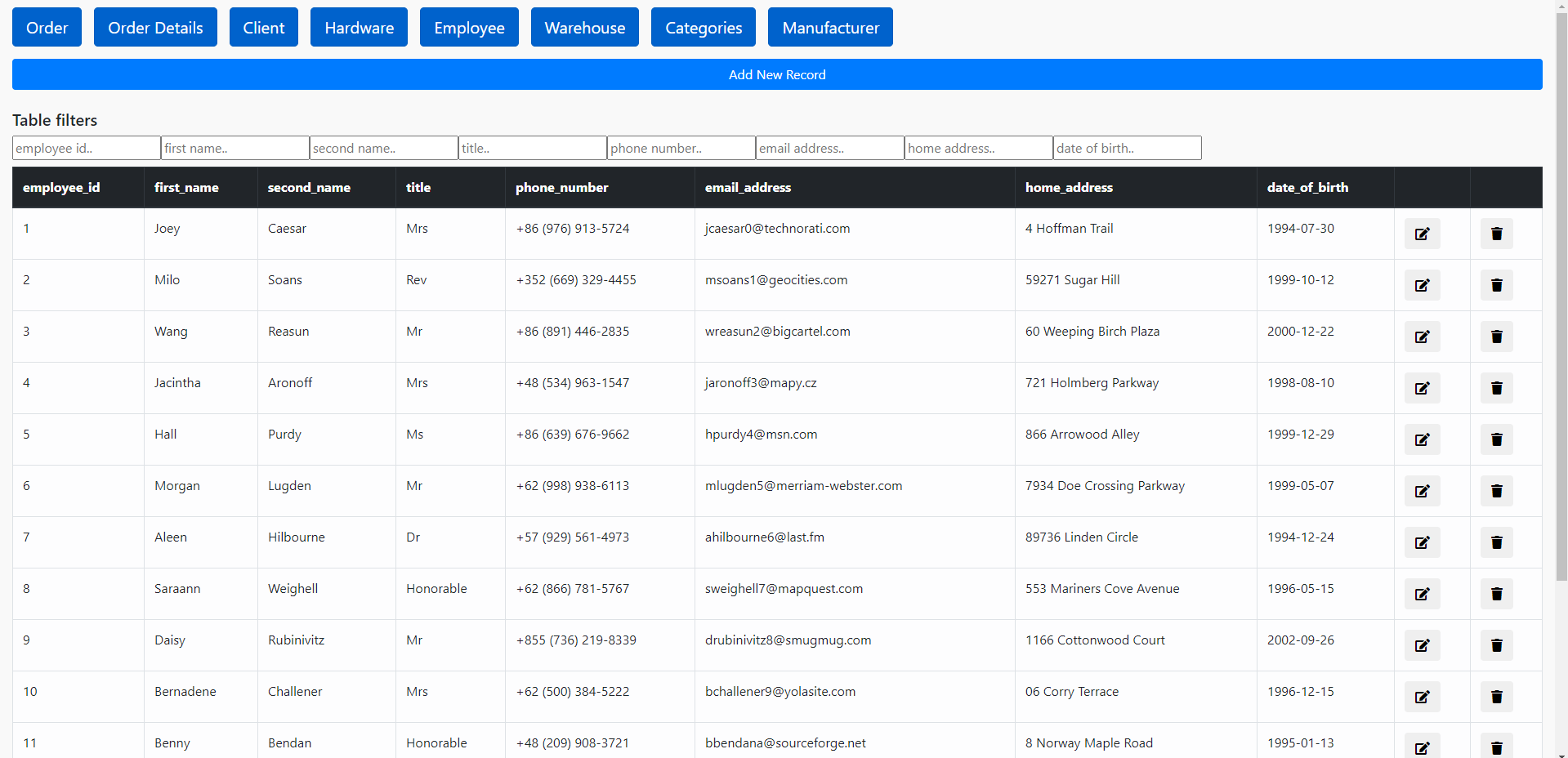


Рисунок 7. Перегляд таблиці “Employee”

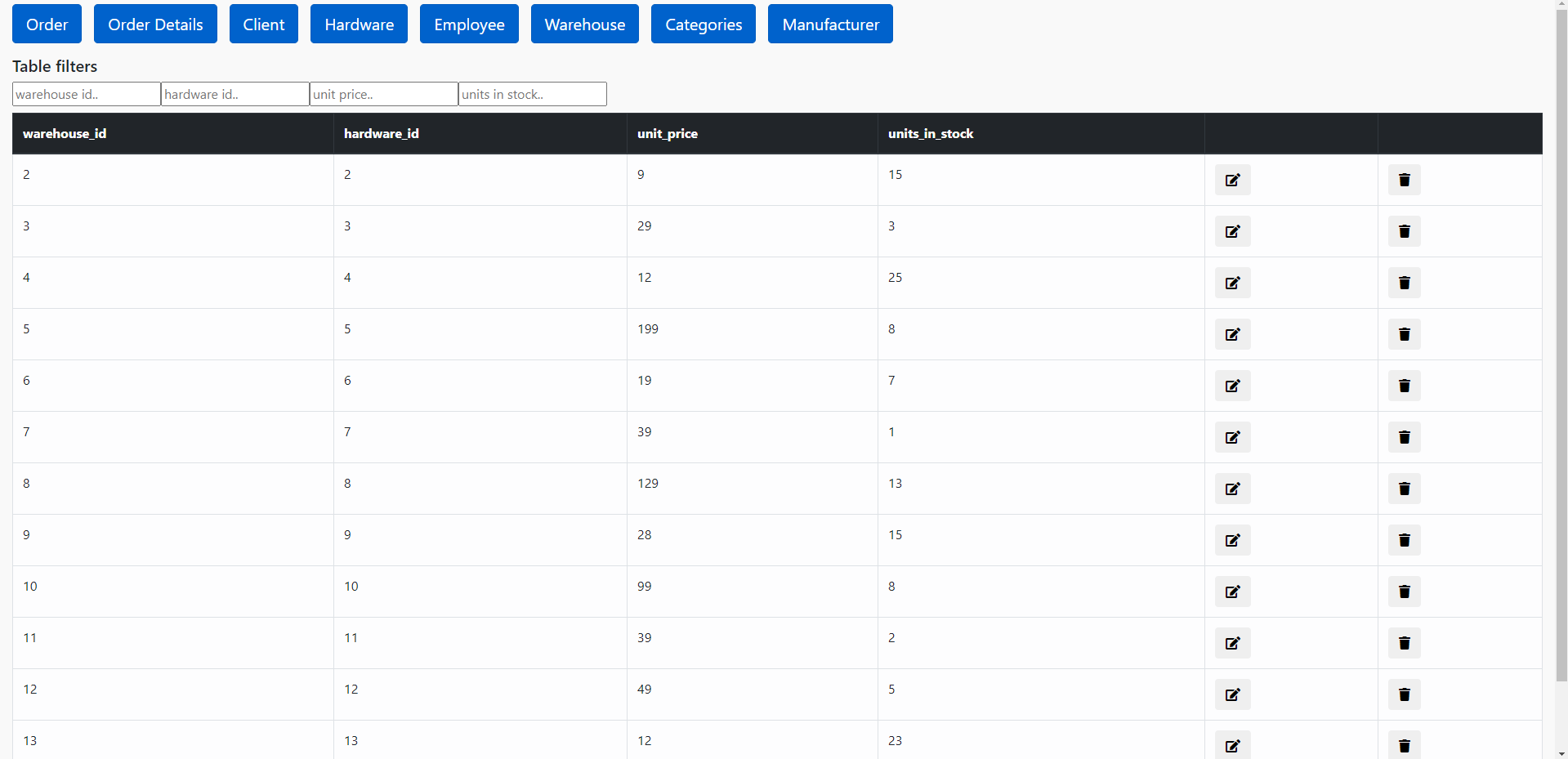


Рисунок 8. Перегляд таблиці “Warehouse”

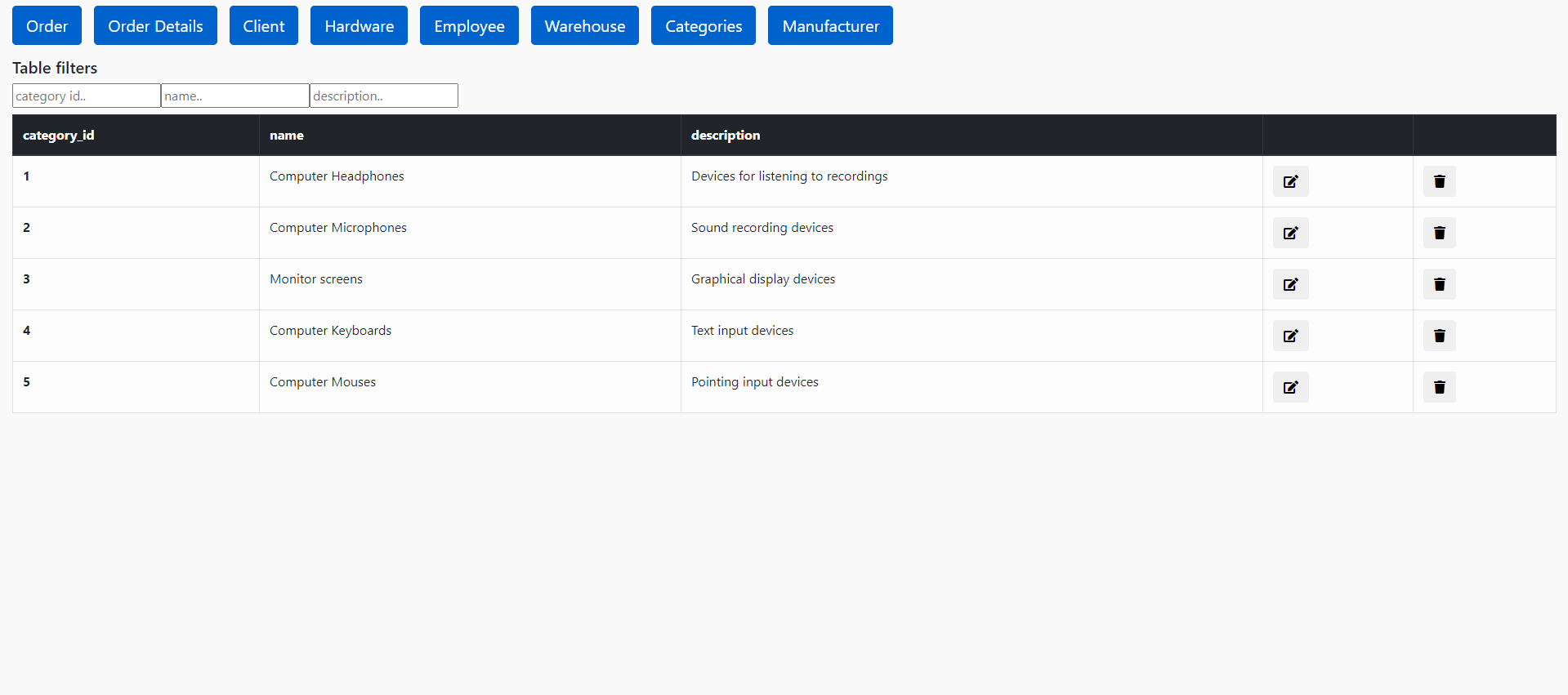


Рисунок 9. Перегляд таблиці “Categories”

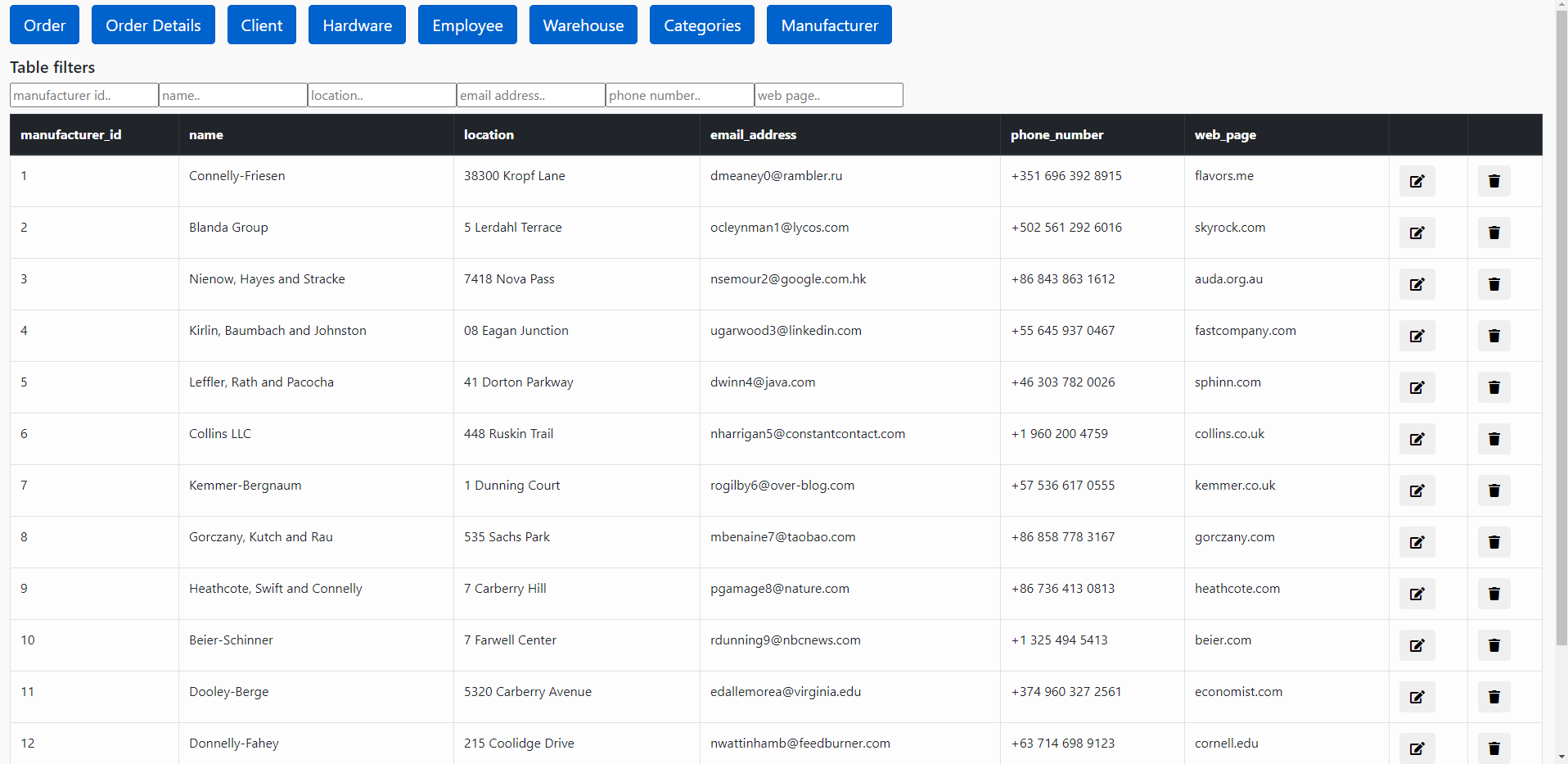


Рисунок 10. Перегляд таблиці “Manufacturer”

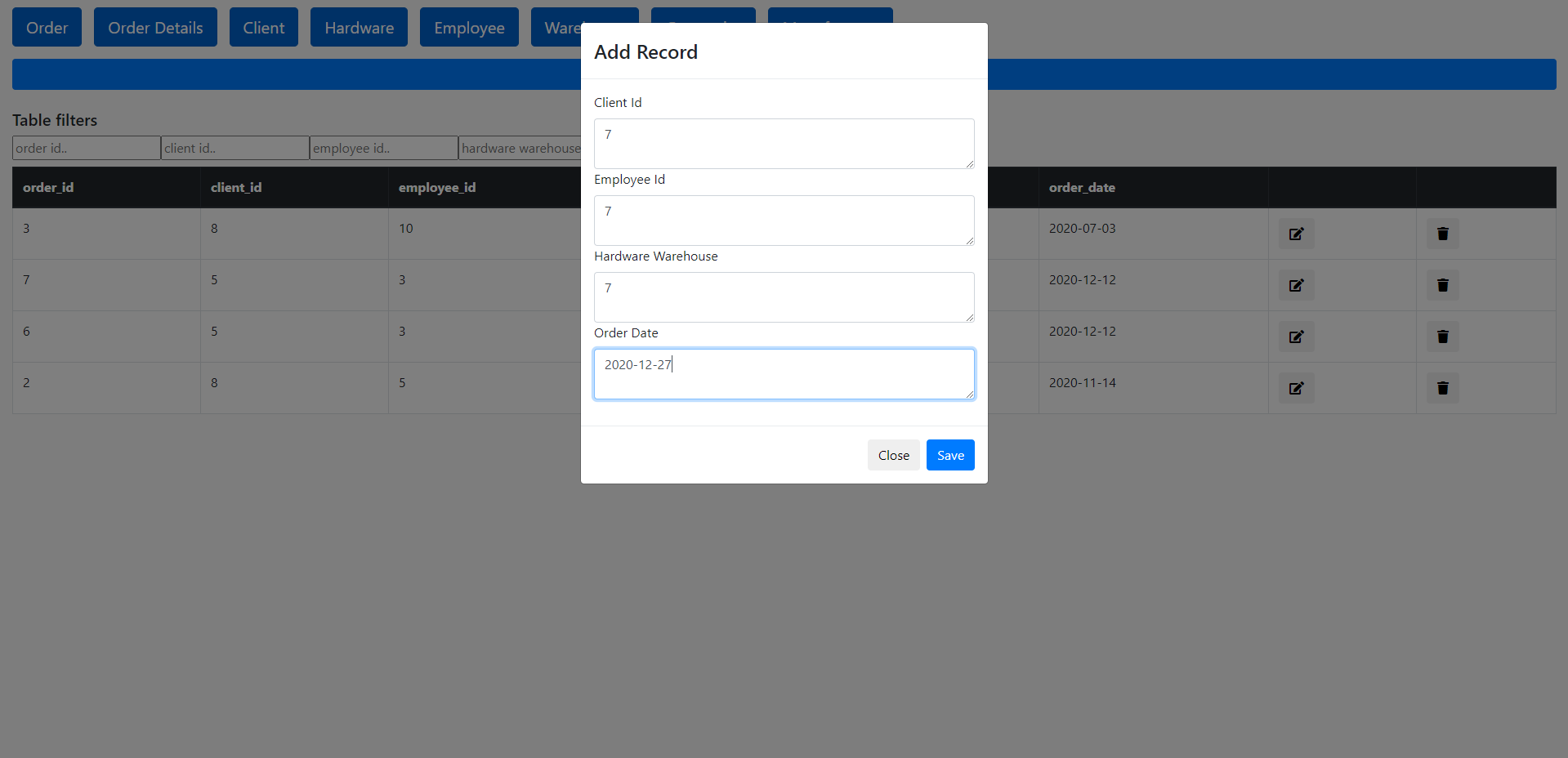


Рисунок 11. Додання нового запису до таблиці

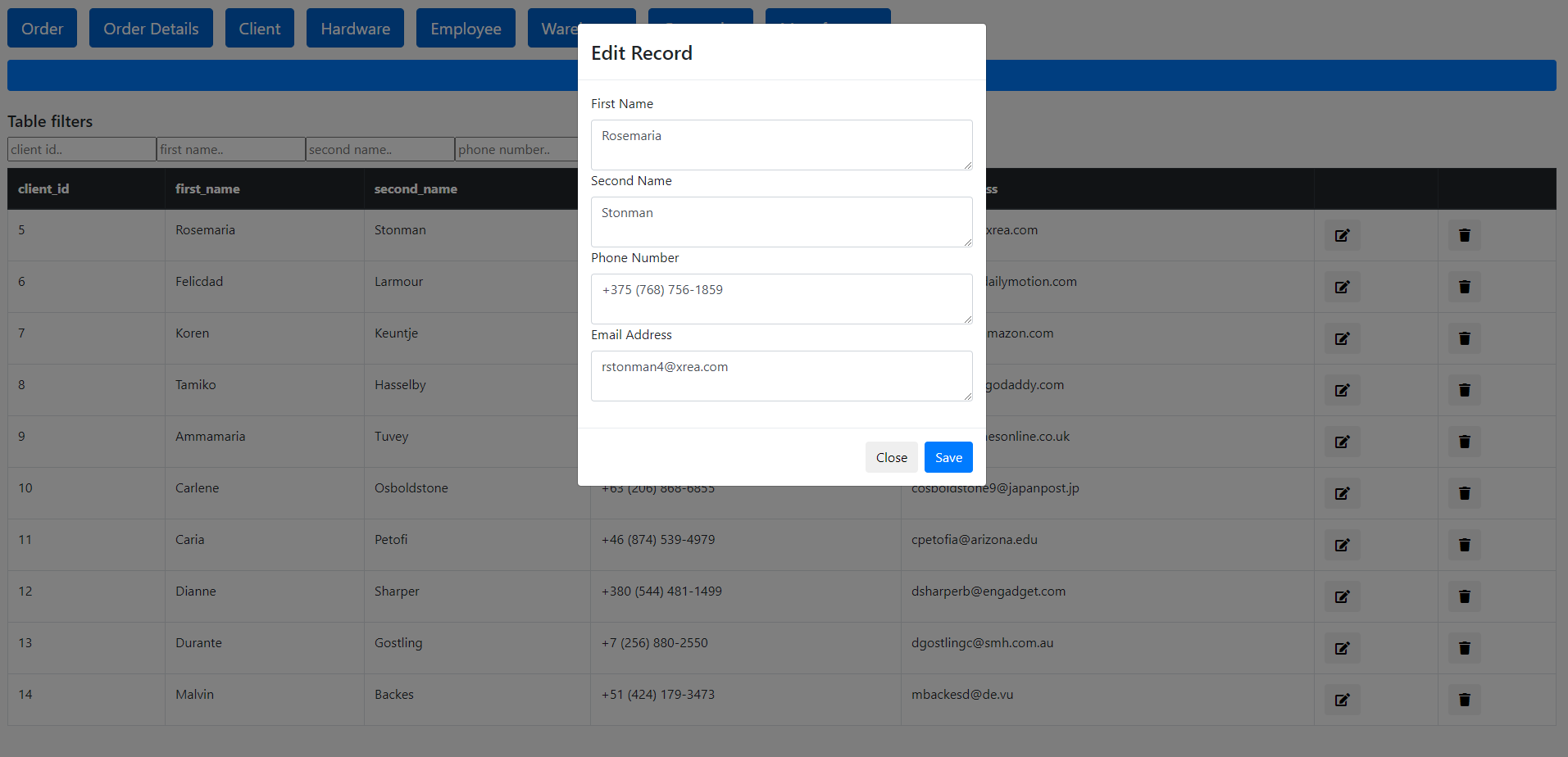


Рисунок 12. Редагування існуючого запису в таблиці

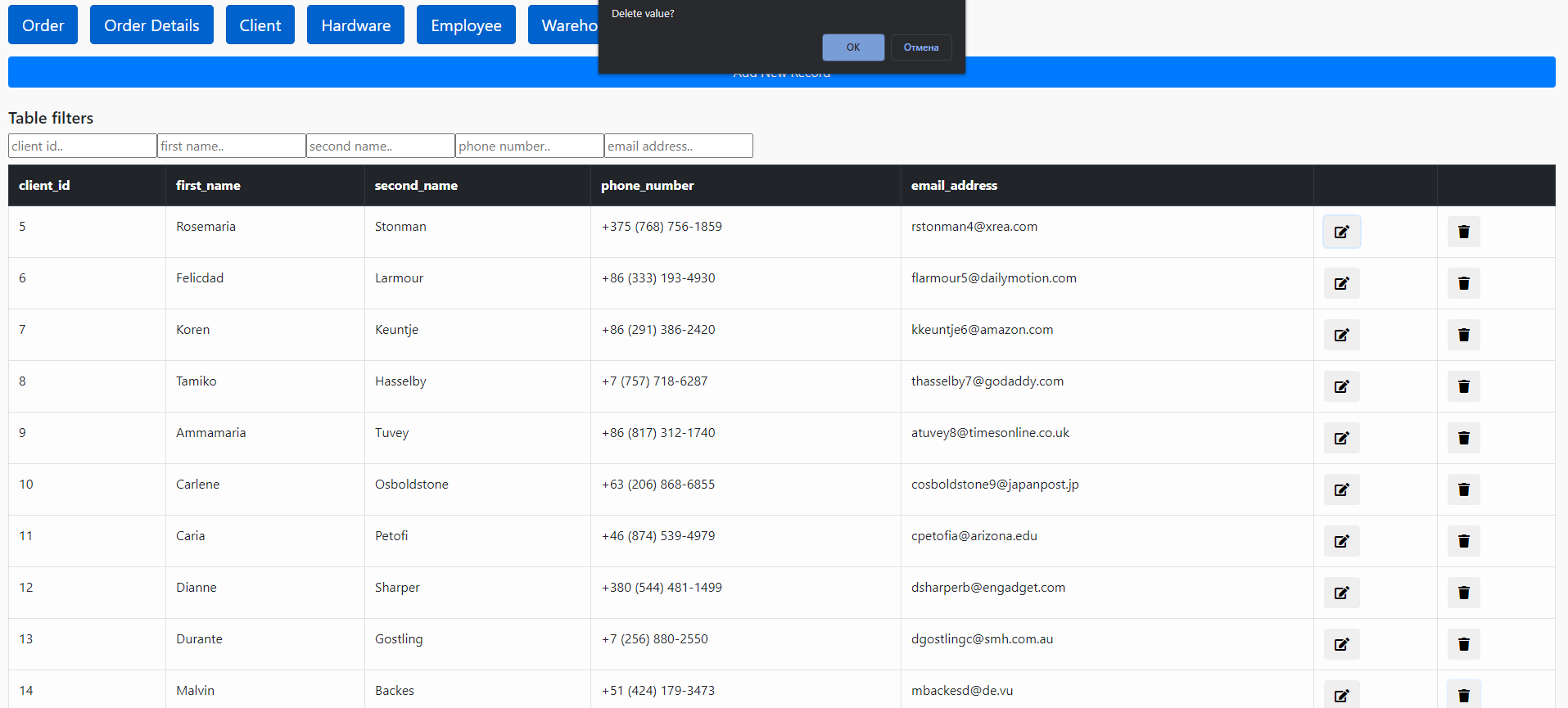


Рисунок 13. Видалення запису з таблиці

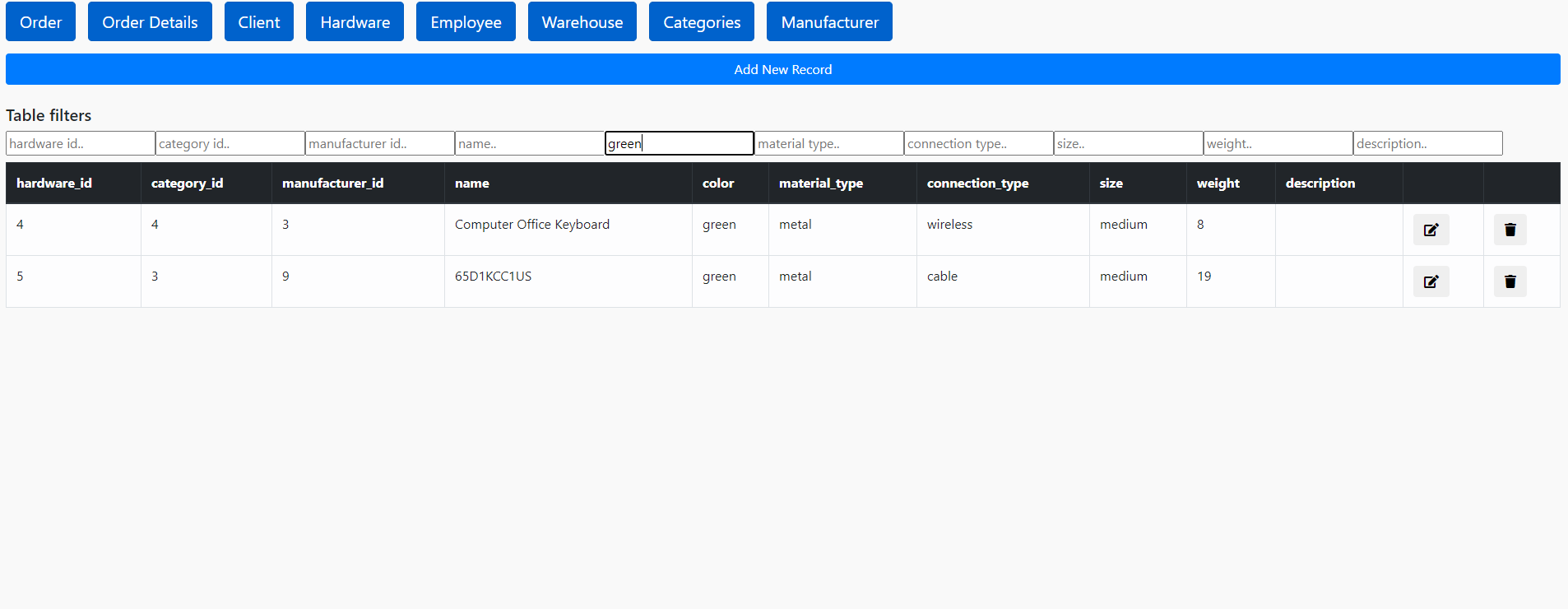


Рисунок 14. Фільтрація даних в таблиці

# **Висновок**

В результаті виконання даної курсової роботи були створені сутності Order, Warehouse, Hardware, Client, Employee, Categories та Manufacturer які покривають функціонал необхідний для роботи магазину комп’ютерних комплектуючих. Створені діаграма сутностей та діаграма фізичної бази даних, які були затверджені з викладачем. За діаграмами була створена база даних hardware\_shop та застосунок, в якому був реалізований функціонал, що передбачував перегляд даних в кожній з наявних в базі даних таблиць, додавання, редагування та видалення записів обраної таблиці, фільтрація виведених даних таблиці за обраним стовпцем.

У майбутньому планується реалізація можливості додавання, редагування та видалення записів для всіх наявних таблиць. Також планується реалізація розбиття таблиць на сторінки й можливість обирати кількість записів на одній сторінці та сторінку перегляду записів. Планується додання різних користувачів для застосунку та загальне поліпшення користувацького інтерфейсу.

Під час написання бази даних та допоміжного застосунку були використані PostgreSQL версії 12.2, VS Code версії 1.52, мова програмування JavaScript та бібліотека node-postgres версії 8.5.0 для користувацького застосунку.

Були покращені знання PostgreSQL та освоєна робота з бібліотекою node-postgres яка дозволяє з’єднати базу даних з користувацьким інтерфейсом користуючись мовою програмування JavaScript.

# **Література**

1. Документація мови PostgreSQL. [Електронний ресурс] - https://www.postgresql.org/docs/
2. Документація мови програмування JavaScript. [Електронний ресурс] - https://developer.mozilla.org/uk/docs/Web/JavaScript
3. Документація бібліотека node-postgres. [Електронний ресурс] - https://node-postgres.com/
4. Github репозиторій бібліотеки node-postgres. [Електронний ресурс] - https://github.com/brianc/node-postgres
5. Онлайн генератор записів до таблиць. [Онлайн генератор] - https://mockaroo.com/
6. Приклади товарів для запису до таблиць. [Онлайн магазин] - https://hard.rozetka.com.ua/

# **Додатки**

Додаток А: код програмного застосунку

-- Creating database

create database hardware\_shop;

-- Creating tables

create table client (

client\_id serial primary key,

first\_name varchar(50) not null,

second\_name varchar(50) not null,

phone\_number varchar(50),

email\_address varchar(50)

);

create table employee (

employee\_id serial primary key,

first\_name varchar(50) not null,

second\_name varchar(50) not null,

title varchar(50) not null,

phone\_number varchar(50) not null,

email\_address varchar(50) not null,

home\_address varchar(50) not null,

date\_of\_birth varchar(50) not null

);

create table categories (

category\_id serial primary key,

name varchar(50) not null,

description varchar(50) not null

);

create table manufacturer (

manufacturer\_id serial primary key,

name varchar(50) not null,

location varchar(50) not null,

email\_address varchar(50) not null,

phone\_number varchar(50) not null,

web\_page varchar(50) not null

);

create table hardware (

hardware\_id serial primary key,

category\_id smallint not null references categories(category\_id) on delete cascade,

manufacturer\_id smallint not null references manufacturer(manufacturer\_id) on delete cascade,

name varchar(50) not null,

color varchar(50) not null,

material\_type varchar(50) not null,

connection\_type varchar(50) not null,

size varchar(50) not null,

weight varchar(50) not null,

description varchar(200)

);

create table warehouse (

warehouse\_id smallint primary key,

hardware\_id smallint unique references hardware(hardware\_id) on delete cascade,

unit\_price real not null,

units\_in\_stock int not null

);

create table "order" (

order\_id serial primary key,

client\_id smallint not null references client(client\_id) on delete cascade,

employee\_id smallint not null references employee(employee\_id) on delete cascade,

hardware\_warehouse smallint not null references warehouse(warehouse\_id) on delete cascade,

order\_date varchar(50) not null

);

create table order\_details (

order\_id smallint not null references "order"(order\_id) on delete cascade,

hardware\_id smallint not null references warehouse(warehouse\_id) on delete cascade,

quantity int not null,

discount real not null

);

-- Inserting values

-- client

insert into client (first\_name, second\_name, phone\_number, email\_address) values ('Caye', 'Tootal', '+420 (964) 177-1667', '[ctootal0@shop-pro.jp](mailto:ctootal0@shop-pro.jp)');

insert into client (first\_name, second\_name, phone\_number, email\_address) values ('Biron', 'Penticoot', '+86 (828) 583-5643', '[bpenticoot1@fda.gov](mailto:bpenticoot1@fda.gov)');

insert into client (first\_name, second\_name, phone\_number, email\_address) values ('Amerigo', 'McKitterick', '+98 (820) 568-4051', '[amckitterick2@shutterfly.com](mailto:amckitterick2@shutterfly.com)');

insert into client (first\_name, second\_name, phone\_number, email\_address) values ('Anna-maria', 'Lemm', '+62 (743) 872-9887', '[alemm3@icio.us](mailto:alemm3@icio.us)');

insert into client (first\_name, second\_name, phone\_number, email\_address) values ('Rosemaria', 'Stonman', '+375 (768) 756-1859', '[rstonman4@xrea.com](mailto:rstonman4@xrea.com)');

insert into client (first\_name, second\_name, phone\_number, email\_address) values ('Felicdad', 'Larmour', '+86 (333) 193-4930', '[flarmour5@dailymotion.com](mailto:flarmour5@dailymotion.com)');

insert into client (first\_name, second\_name, phone\_number, email\_address) values ('Koren', 'Keuntje', '+86 (291) 386-2420', '[kkeuntje6@amazon.com](mailto:kkeuntje6@amazon.com)');

insert into client (first\_name, second\_name, phone\_number, email\_address) values ('Tamiko', 'Hasselby', '+7 (757) 718-6287', '[thasselby7@godaddy.com](mailto:thasselby7@godaddy.com)');

insert into client (first\_name, second\_name, phone\_number, email\_address) values ('Ammamaria', 'Tuvey', '+86 (817) 312-1740', '[atuvey8@timesonline.co.uk](mailto:atuvey8@timesonline.co.uk)');

insert into client (first\_name, second\_name, phone\_number, email\_address) values ('Carlene', 'Osboldstone', '+63 (206) 868-6855', '[cosboldstone9@japanpost.jp](mailto:cosboldstone9@japanpost.jp)');

insert into client (first\_name, second\_name, phone\_number, email\_address) values ('Caria', 'Petofi', '+46 (874) 539-4979', '[cpetofia@arizona.edu](mailto:cpetofia@arizona.edu)');

insert into client (first\_name, second\_name, phone\_number, email\_address) values ('Dianne', 'Sharper', '+380 (544) 481-1499', '[dsharperb@engadget.com](mailto:dsharperb@engadget.com)');

insert into client (first\_name, second\_name, phone\_number, email\_address) values ('Durante', 'Gostling', '+7 (256) 880-2550', '[dgostlingc@smh.com.au](mailto:dgostlingc@smh.com.au)');

insert into client (first\_name, second\_name, phone\_number, email\_address) values ('Malvin', 'Backes', '+51 (424) 179-3473', '[mbackesd@de.vu](mailto:mbackesd@de.vu)');

insert into client (first\_name, second\_name, phone\_number, email\_address) values ('Cherie', 'Sidnell', '+86 (292) 273-4307', '[csidnelle@shinystat.com](mailto:csidnelle@shinystat.com)');

-- employee

insert into employee (first\_name, second\_name, title, phone\_number, email\_address, home\_address, date\_of\_birth) values ('Joey', 'Caesar', 'Mrs', '+86 (976) 913-5724', '[jcaesar0@technorati.com](mailto:jcaesar0@technorati.com)', '4 Hoffman Trail', '1994-07-30');

insert into employee (first\_name, second\_name, title, phone\_number, email\_address, home\_address, date\_of\_birth) values ('Milo', 'Soans', 'Rev', '+352 (669) 329-4455', '[msoans1@geocities.com](mailto:msoans1@geocities.com)', '59271 Sugar Hill', '1999-10-12');

insert into employee (first\_name, second\_name, title, phone\_number, email\_address, home\_address, date\_of\_birth) values ('Wang', 'Reasun', 'Mr', '+86 (891) 446-2835', '[wreasun2@bigcartel.com](mailto:wreasun2@bigcartel.com)', '60 Weeping Birch Plaza', '2000-12-22');

insert into employee (first\_name, second\_name, title, phone\_number, email\_address, home\_address, date\_of\_birth) values ('Jacintha', 'Aronoff', 'Mrs', '+48 (534) 963-1547', '[jaronoff3@mapy.cz](mailto:jaronoff3@mapy.cz)', '721 Holmberg Parkway', '1998-08-10');

insert into employee (first\_name, second\_name, title, phone\_number, email\_address, home\_address, date\_of\_birth) values ('Hall', 'Purdy', 'Ms', '+86 (639) 676-9662', '[hpurdy4@msn.com](mailto:hpurdy4@msn.com)', '866 Arrowood Alley', '1999-12-29');

insert into employee (first\_name, second\_name, title, phone\_number, email\_address, home\_address, date\_of\_birth) values ('Morgan', 'Lugden', 'Mr', '+62 (998) 938-6113', '[mlugden5@merriam-webster.com](mailto:mlugden5@merriam-webster.com)', '7934 Doe Crossing Parkway', '1999-05-07');

insert into employee (first\_name, second\_name, title, phone\_number, email\_address, home\_address, date\_of\_birth) values ('Aleen', 'Hilbourne', 'Dr', '+57 (929) 561-4973', '[ahilbourne6@last.fm](mailto:ahilbourne6@last.fm)', '89736 Linden Circle', '1994-12-24');

insert into employee (first\_name, second\_name, title, phone\_number, email\_address, home\_address, date\_of\_birth) values ('Saraann', 'Weighell', 'Honorable', '+62 (866) 781-5767', '[sweighell7@mapquest.com](mailto:sweighell7@mapquest.com)', '553 Mariners Cove Avenue', '1996-05-15');

insert into employee (first\_name, second\_name, title, phone\_number, email\_address, home\_address, date\_of\_birth) values ('Daisy', 'Rubinivitz', 'Mr', '+855 (736) 219-8339', '[drubinivitz8@smugmug.com](mailto:drubinivitz8@smugmug.com)', '1166 Cottonwood Court', '2002-09-26');

insert into employee (first\_name, second\_name, title, phone\_number, email\_address, home\_address, date\_of\_birth) values ('Bernadene', 'Challener', 'Mrs', '+62 (500) 384-5222', '[bchallener9@yolasite.com](mailto:bchallener9@yolasite.com)', '06 Corry Terrace', '1996-12-15');

insert into employee (first\_name, second\_name, title, phone\_number, email\_address, home\_address, date\_of\_birth) values ('Benny', 'Bendan', 'Honorable', '+48 (209) 908-3721', '[bbendana@sourceforge.net](mailto:bbendana@sourceforge.net)', '8 Norway Maple Road', '1995-01-13');

insert into employee (first\_name, second\_name, title, phone\_number, email\_address, home\_address, date\_of\_birth) values ('Margi', 'Lumpkin', 'Honorable', '+63 (713) 712-2053', '[mlumpkinb@nationalgeographic.com](mailto:mlumpkinb@nationalgeographic.com)', '7999 Walton Center', '2001-08-04');

insert into employee (first\_name, second\_name, title, phone\_number, email\_address, home\_address, date\_of\_birth) values ('Hamish', 'Verrico', 'Rev', '+86 (272) 400-9652', '[hverricoc@desdev.cn](mailto:hverricoc@desdev.cn)', '21039 Sachs Crossing', '1996-07-23');

insert into employee (first\_name, second\_name, title, phone\_number, email\_address, home\_address, date\_of\_birth) values ('Corene', 'Hitschke', 'Ms', '+381 (473) 682-5475', '[chitschked@buzzfeed.com](mailto:chitschked@buzzfeed.com)', '116 Lake View Alley', '1995-03-13');

insert into employee (first\_name, second\_name, title, phone\_number, email\_address, home\_address, date\_of\_birth) values ('Gabrila', 'McKnish', 'Rev', '+51 (747) 661-7187', '[gmcknishe@diigo.com](mailto:gmcknishe@diigo.com)', '0 Talmadge Center', '1996-08-05');

-- categories

insert into categories (name, description) values ('Computer Headphones','Devices for listening to recordings');

insert into categories (name, description) values ('Computer Microphones','Sound recording devices');

insert into categories (name, description) values ('Monitor screens','Graphical display devices');

insert into categories (name, description) values ('Computer Keyboards','Text input devices');

insert into categories (name, description) values ('Computer Mouses','Pointing input devices');

-- manufacturer

insert into manufacturer (name, location, email\_address, phone\_number, web\_page) values ('Connelly-Friesen', '38300 Kropf Lane', '[dmeaney0@rambler.ru](mailto:dmeaney0@rambler.ru)', '+351 696 392 8915', 'flavors.me');

insert into manufacturer (name, location, email\_address, phone\_number, web\_page) values ('Blanda Group', '5 Lerdahl Terrace', '[ocleynman1@lycos.com](mailto:ocleynman1@lycos.com)', '+502 561 292 6016', 'skyrock.com');

insert into manufacturer (name, location, email\_address, phone\_number, web\_page) values ('Nienow, Hayes and Stracke', '7418 Nova Pass', '[nsemour2@google.com.hk](mailto:nsemour2@google.com.hk)', '+86 843 863 1612', 'auda.org.au');

insert into manufacturer (name, location, email\_address, phone\_number, web\_page) values ('Kirlin, Baumbach and Johnston', '08 Eagan Junction', '[ugarwood3@linkedin.com](mailto:ugarwood3@linkedin.com)', '+55 645 937 0467', 'fastcompany.com');

insert into manufacturer (name, location, email\_address, phone\_number, web\_page) values ('Leffler, Rath and Pacocha', '41 Dorton Parkway', '[dwinn4@java.com](mailto:dwinn4@java.com)', '+46 303 782 0026', 'sphinn.com');

insert into manufacturer (name, location, email\_address, phone\_number, web\_page) values ('Collins LLC', '448 Ruskin Trail', '[nharrigan5@constantcontact.com](mailto:nharrigan5@constantcontact.com)', '+1 960 200 4759', 'collins.co.uk');

insert into manufacturer (name, location, email\_address, phone\_number, web\_page) values ('Kemmer-Bergnaum', '1 Dunning Court', '[rogilby6@over-blog.com](mailto:rogilby6@over-blog.com)', '+57 536 617 0555', 'kemmer.co.uk');

insert into manufacturer (name, location, email\_address, phone\_number, web\_page) values ('Gorczany, Kutch and Rau', '535 Sachs Park', '[mbenaine7@taobao.com](mailto:mbenaine7@taobao.com)', '+86 858 778 3167', 'gorczany.com');

insert into manufacturer (name, location, email\_address, phone\_number, web\_page) values ('Heathcote, Swift and Connelly', '7 Carberry Hill', '[pgamage8@nature.com](mailto:pgamage8@nature.com)', '+86 736 413 0813', 'heathcote.com');

insert into manufacturer (name, location, email\_address, phone\_number, web\_page) values ('Beier-Schinner', '7 Farwell Center', '[rdunning9@nbcnews.com](mailto:rdunning9@nbcnews.com)', '+1 325 494 5413', 'beier.com');

insert into manufacturer (name, location, email\_address, phone\_number, web\_page) values ('Dooley-Berge', '5320 Carberry Avenue', '[edallemorea@virginia.edu](mailto:edallemorea@virginia.edu)', '+374 960 327 2561', 'economist.com');

insert into manufacturer (name, location, email\_address, phone\_number, web\_page) values ('Donnelly-Fahey', '215 Coolidge Drive', '[nwattinhamb@feedburner.com](mailto:nwattinhamb@feedburner.com)', '+63 714 698 9123', 'cornell.edu');

insert into manufacturer (name, location, email\_address, phone\_number, web\_page) values ('Homenick-Stiedemann', '3 Manufacturers Crossing', '[roveyc@amazon.co.jp](mailto:roveyc@amazon.co.jp)', '+63 428 507 6347', 'homenick.es');

insert into manufacturer (name, location, email\_address, phone\_number, web\_page) values ('Swift LLC', '289 High Crossing Parkway', '[mmcphated@home.pl](mailto:mmcphated@home.pl)', '+231 919 389 1087', 'networksolutions.com');

insert into manufacturer (name, location, email\_address, phone\_number, web\_page) values ('Dare LLC', '39 Erie Court', '[gvowlese@g.co](mailto:gvowlese@g.co)', '+7 269 686 5371', 'fema.gov');

-- hardware

insert into hardware (category\_id, manufacturer\_id, name, color, material\_type, connection\_type, size, weight, description) values (5, 2, 'DeathAdder V2 Gaming Mouse', 'golden', 'metal', 'cable', 'large', 2, 'Large mouse for true gamers.');

insert into hardware (category\_id, manufacturer\_id, name, color, material\_type, connection\_type, size, weight, description) values (2, 2, 'SUDOTACK Condenser ', 'black', 'metal', 'wireless', 'medium', 1, '');

insert into hardware (category\_id, manufacturer\_id, name, color, material\_type, connection\_type, size, weight, description) values (4, 2, 'FENIFOX Slim 2.4G', 'red', 'metal', 'wireless', 'large', 7, 'Mechanical game-device for people without limits');

insert into hardware (category\_id, manufacturer\_id, name, color, material\_type, connection\_type, size, weight, description) values (4, 3, 'Computer Office Keyboard', 'green', 'metal', 'wireless', 'medium', 8, '');

insert into hardware (category\_id, manufacturer\_id, name, color, material\_type, connection\_type, size, weight, description) values (3, 9, '65D1KCC1US', 'green', 'metal', 'cable', 'medium', 19, '');

insert into hardware (category\_id, manufacturer\_id, name, color, material\_type, connection\_type, size, weight, description) values (5, 1, 'Cimetech 2.4G Slim', 'grey', 'plastic', 'wireless', 'small', 1, 'Ultra precise computer mouse');

insert into hardware (category\_id, manufacturer\_id, name, color, material\_type, connection\_type, size, weight, description) values (1, 10, 'USB Gaming Headset', 'yellow', 'plastic', 'cable', 'medium', 5, '');

insert into hardware (category\_id, manufacturer\_id, name, color, material\_type, connection\_type, size, weight, description) values (3, 9, 'TU87F 32', 'golden', 'metal', 'cable', 'large', 18, '4K 60Hz For low price.');

insert into hardware (category\_id, manufacturer\_id, name, color, material\_type, connection\_type, size, weight, description) values (1, 10, 'Pro X Gaming Headset ', 'grey', 'plastic', 'cable', 'medium', 4, '');

insert into hardware (category\_id, manufacturer\_id, name, color, material\_type, connection\_type, size, weight, description) values (1, 6, 'Full-Sized, Lightweight Long-Cord Headphones', 'golden', 'metal', 'cable', 'medium', 5, '');

insert into hardware (category\_id, manufacturer\_id, name, color, material\_type, connection\_type, size, weight, description) values (1, 3, 'Gaming Headset - Black And Blue', 'black', 'plastic', 'cable', 'small', 3, 'Cutting edge device, ultra soft sound.');

insert into hardware (category\_id, manufacturer\_id, name, color, material\_type, connection\_type, size, weight, description) values (1, 7, 'Hi-Fi On-Ear Headphones with 9-Foot Cord', 'white', 'plastic', 'cable', 'large', 6, 'Best sound for best price!');

insert into hardware (category\_id, manufacturer\_id, name, color, material\_type, connection\_type, size, weight, description) values (5, 4, 'MM057 2.4G', 'black', 'metal', 'wireless', 'small', 2, '');

insert into hardware (category\_id, manufacturer\_id, name, color, material\_type, connection\_type, size, weight, description) values (1, 9, 'bopmen T3', 'white', 'metal', 'wireless', 'medium', 7, '');

insert into hardware (category\_id, manufacturer\_id, name, color, material\_type, connection\_type, size, weight, description) values (1, 4, 'ZX Series On-Ear Headphones', 'green', 'metal', 'wireless', 'large', 5, 'Next gen expirience for our customers!');

-- warehouse

insert into warehouse (warehouse\_id, hardware\_id, unit\_price, units\_in\_stock) values (1, 1, 49, 5);

insert into warehouse (warehouse\_id, hardware\_id, unit\_price, units\_in\_stock) values (2, 2, 9, 15);

insert into warehouse (warehouse\_id, hardware\_id, unit\_price, units\_in\_stock) values (3, 3, 29, 3);

insert into warehouse (warehouse\_id, hardware\_id, unit\_price, units\_in\_stock) values (4, 4, 12, 25);

insert into warehouse (warehouse\_id, hardware\_id, unit\_price, units\_in\_stock) values (5, 5, 199, 8);

insert into warehouse (warehouse\_id, hardware\_id, unit\_price, units\_in\_stock) values (6, 6, 19, 7);

insert into warehouse (warehouse\_id, hardware\_id, unit\_price, units\_in\_stock) values (7, 7, 39, 1);

insert into warehouse (warehouse\_id, hardware\_id, unit\_price, units\_in\_stock) values (8, 8, 129, 13);

insert into warehouse (warehouse\_id, hardware\_id, unit\_price, units\_in\_stock) values (9, 9, 28, 15);

insert into warehouse (warehouse\_id, hardware\_id, unit\_price, units\_in\_stock) values (10, 10, 99, 8);

insert into warehouse (warehouse\_id, hardware\_id, unit\_price, units\_in\_stock) values (11, 11, 39, 2);

insert into warehouse (warehouse\_id, hardware\_id, unit\_price, units\_in\_stock) values (12, 12, 49, 5);

insert into warehouse (warehouse\_id, hardware\_id, unit\_price, units\_in\_stock) values (13, 13, 12, 23);

insert into warehouse (warehouse\_id, hardware\_id, unit\_price, units\_in\_stock) values (14, 14, 22, 5);

insert into warehouse (warehouse\_id, hardware\_id, unit\_price, units\_in\_stock) values (15, 15, 29, 8);

-- order

insert into "order" (client\_id, employee\_id, hardware\_warehouse, order\_date) values (6, 9, 15, '2020-04-10');

insert into "order" (client\_id, employee\_id, hardware\_warehouse, order\_date) values (8, 5, 10, '2020-11-11');

insert into "order" (client\_id, employee\_id, hardware\_warehouse, order\_date) values (8, 10, 5, '2020-07-03');

insert into "order" (client\_id, employee\_id, hardware\_warehouse, order\_date) values (4, 10, 12, '2020-07-23');

insert into "order" (client\_id, employee\_id, hardware\_warehouse, order\_date) values (4, 4, 15, '2020-04-12');

insert into "order" (client\_id, employee\_id, hardware\_warehouse, order\_date) values (5, 3, 7, '2020-03-20');

insert into "order" (client\_id, employee\_id, hardware\_warehouse, order\_date) values (6, 4, 2, '2020-01-11');

insert into "order" (client\_id, employee\_id, hardware\_warehouse, order\_date) values (6, 9, 12, '2020-11-14');

insert into "order" (client\_id, employee\_id, hardware\_warehouse, order\_date) values (5, 4, 12, '2020-05-18');

insert into "order" (client\_id, employee\_id, hardware\_warehouse, order\_date) values (9, 1, 9, '2020-10-31');

-- order\_details

insert into order\_details (order\_id, hardware\_id, quantity, discount) values (1, 6, 1, 0.7);

insert into order\_details (order\_id, hardware\_id, quantity, discount) values (2, 8, 1, 1);

insert into order\_details (order\_id, hardware\_id, quantity, discount) values (3, 8, 1, 0.8);

insert into order\_details (order\_id, hardware\_id, quantity, discount) values (4, 4, 3, 0.6);

insert into order\_details (order\_id, hardware\_id, quantity, discount) values (5, 4, 1, 1);

insert into order\_details (order\_id, hardware\_id, quantity, discount) values (6, 5, 2, 1);

insert into order\_details (order\_id, hardware\_id, quantity, discount) values (7, 6, 3, 1);

insert into order\_details (order\_id, hardware\_id, quantity, discount) values (8, 6, 3, 1);

insert into order\_details (order\_id, hardware\_id, quantity, discount) values (9, 5, 1, 0.5);

insert into order\_details (order\_id, hardware\_id, quantity, discount) values (10, 9, 3, 0.5);

-- Creating VIEWS

--===============================================================================================

-- Display all items that need restock (CASE + JOIN)

create or replace view items\_restock as

select name, description, units\_in\_stock,

case

when units\_in\_stock >= 5 then 'Do not need restock'

when units\_in\_stock < 5 then 'Needs restock'

end needsRestock

from hardware inner join warehouse on hardware.hardware\_id = warehouse.hardware\_id;

-- Display all clients that have email address or phone number

create or replace view clients\_to\_contact as

select \* from client where email\_address is not null or phone\_number is not null;

-- Display all people in db (UNION)

create or replace view display\_all\_people as

select first\_name, second\_name from client

union

select first\_name, second\_name from employee;

--===============================================================================================

-- Creating SCALAR and TABLE functions

--===============================================================================================

-- Display full price of order

create or replace function order\_price(quantity int, price real, discount real)

returns real

language 'plpgsql'

as $$

begin

return @quantity \* @price \* @discount;

end;

$$;

-- Display all orders between two dates

create or replace function orders\_between(date1 date, date2 date)

returns setof "order"

language 'plpgsql'

as $$

begin

if date1 = null or date2 = null then return query select \* from "order" where order\_date = current\_date;

else return query select \* from "order" where order\_date between date1 and date2;

end if;

end;

$$;

--===============================================================================================

-- Creating PROCEDURES

--===============================================================================================

-- Insert new client

create or replace function addNewClient(\_first\_name varchar, \_second\_name varchar, \_phone\_number varchar, \_email\_address varchar)

returns void

language 'plpgsql'

as $$

begin

insert into client (first\_name, second\_name, phone\_number, email\_address) values (\_first\_name, \_second\_name, \_phone\_number, \_email\_address);

end;

$$;

-- Insert new employee

create or replace function addNewEmployee(\_first\_name varchar, \_second\_name varchar, \_title varchar, \_phone\_number varchar, \_email\_address varchar, \_home\_address varchar, \_date\_of\_birth date)

returns void

language 'plpgsql'

as $$

begin

insert into employee (first\_name, second\_name, title, phone\_number, email\_address, home\_address, date\_of\_birth) values (\_first\_name, \_second\_name, \_title, \_phone\_number, \_email\_address, \_home\_address, \_date\_of\_birth);

end;

$$;

--===============================================================================================

-- Creating TRIGGERS

--===============================================================================================

-- Moving deleted orders to orders\_archive table

create table if not exists orders\_archive (

date\_of\_delete timestamp,

delete\_by text,

like "order"

);

create or replace function move\_to\_archive()

returns trigger

language 'plpgsql'

as $$

begin

insert into orders\_archive (order\_id, client\_id, employee\_id, hardware\_warehouse, order\_date, date\_of\_delete, delete\_by)

select old.\*, statement\_timestamp(), session\_user;

return new;

end;

$$

create trigger move\_to\_archive

before delete on "order"

for each row execute function move\_to\_archive();

--===============================================================================================

-- Default value

alter table hardware

alter description set default 'No Description';