

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ  
СІКОРСЬКОГО»

Факультет прикладної математики  
Кафедра програмного забезпечення комп'ютерних систем

**ЗВІТ**  
з лабораторної роботи № 2  
«Створення додатку бази даних, орієнтованого на взаємодію з СУБД  
PostgreSQL»

**Виконав:**

Варіант: 21

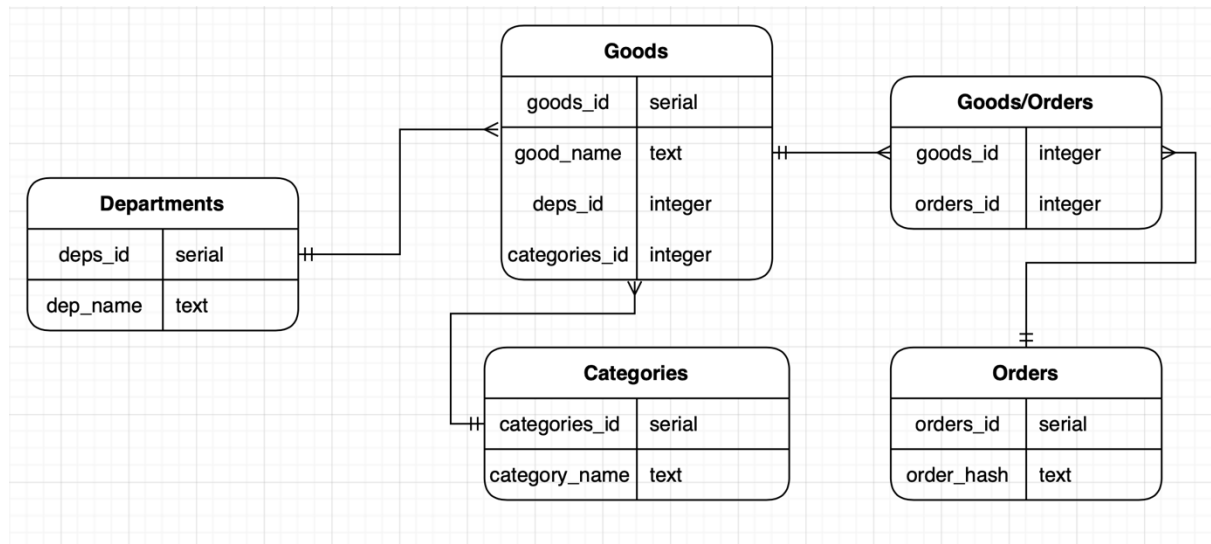
студент 3-го курсу, групи КП-83,  
спеціальності 121 – Інженерія  
програмного забезпечення

*Шаповалов Данило Данило*

Київ – 2020

## ТЕМА: «Магазин»

### Розроблена модель «сутність-зв'язок»:



### Загальне завдання роботи полягає у наступному:

1. Реалізувати функції внесення, редагування та видалення даних у таблицях бази даних, створених у лабораторній роботі No1, засобами консольного інтерфейсу.
2. Передбачити автоматичне пакетне генерування «рандомізованих» даних у базі.
3. Забезпечити реалізацію пошуку за декількома атрибутами з двох та більше сутностей одночасно: для числових атрибутів – у рамках діапазону, для рядкових – як шаблон функції LIKE оператора SELECT SQL, для логічного типу – значення True/False, для дат – у рамках діапазону дат.
4. Програмний код виконати згідно шаблону MVC (модель-подання-контролер).

## TASK 1

### 1. Додавання

```
[INSERT MODE]
Name:
Molokia Kephir
Department id:
1
Category id:
2
New good: Good {
  goods_id: 0,
  good_name: "Molokia Kephir\n",
  departments_id: 2,
  categories_id: 1,
  table_name: "Goods",
}
```

Бачимо доданий у назву пункт:

|    |                |   |   |    |
|----|----------------|---|---|----|
| 22 | Molokia Kephir | 2 | 1 | 24 |
|----|----------------|---|---|----|

### 2. Вилучення

```
[DELETE MODE]
Delete by id:
22
Good 22 was deleted
```

Бачимо, що видаляємий об'єкт у базі відсутній:

|    |                |   |   |    |
|----|----------------|---|---|----|
| 18 | Organic Milk   | 4 | 4 | 21 |
| 19 | Lego Robot X21 | 4 | 2 | 23 |

Нижче спробуємо видалити неіснуючий елемент:

```
[DELETING MODE]
Id:
1
Something wrong!
```

### 3. Редагування

Початковий об'єкт:

|   |      |   |   |   |
|---|------|---|---|---|
| 4 | Milk | 3 | 2 | 4 |
|---|------|---|---|---|

Редагування:

```
[EDITING MODE]
Edit by id:
4
Name:
Milk Organic
Department id:
1
Category id:
1
UPDATE public."Goods" SET good_name='Milk Organic', departments_id=1, categories_id=1 WHERE goods_id=4
Good was updated successfully: Good {
  goods_id: 4,
  good_name: "Milk Organic",
  departments_id: 1,
  categories_id: 1,
  table_name: "Goods",
}
```

База даних:

|    |              |   |   |   |
|----|--------------|---|---|---|
| 20 | Milk Organic | 1 | 1 | 4 |
|----|--------------|---|---|---|

## TASK 2

Ми можемо бачити, що останній елемент – 20:

|    |                |   |   |    |
|----|----------------|---|---|----|
| 19 | Molokia Kephir | 2 | 1 | 24 |
| 20 | Milk Organic   | 1 | 1 | 4  |
|    |                |   |   |    |

Генеруємо:

```
[RANDOM GENERATING]
Generated!
```

Отримали:

|    |                |   |   |    |
|----|----------------|---|---|----|
| 21 | Oil Oleyna     | 4 | 2 | 25 |
| 22 | Oil Oleyna     | 4 | 2 | 26 |
| 23 | Charger 50W    | 4 | 4 | 27 |
| 24 | Pencil         | 1 | 2 | 28 |
| 25 | Lego Robot X21 | 2 | 4 | 29 |
| 26 | Lego Robot X21 | 6 | 1 | 30 |
| 27 | Oil Oleyna     | 7 | 1 | 31 |
| 28 | Lamp 3W        | 3 | 1 | 32 |
| 29 | Organic Milk   | 4 | 1 | 33 |

## TASK 3:

Фільтрація по іменах хешів замовлень та відповідних товарів:

```
[FILTER MODE]
Input orders name filter:
r
Input goods name filter:
e
1f332srwq Rybay Steak
1f332srwq iPhone 11 Pro Max
0f978wrk Copybook Cells 24
0f978wrk Crocodile
Time: 1 ms
```

Фільтрація по іменах товарів та відповідних категорій:

```
[FILTER MODE]
Input goods name filter:
i
Input categories name filter:
e
MacBook Pro 16 inch 2019 Technics
MacBook Pro 13 inch 2020 Technics
iPhone XR Technics
iPhone 11 Pro Max Technics
Frosh Washing Home Chemistry
Palmolive Shampoo Home Chemistry
Crocodile Technics
Pencil Home Chemistry
Plastic Bottle Home Chemistry
Pencil Home Chemistry
Oil Oleyna Technics
Molokia Kephir
Technics
Milk Organic Technics
Oil Oleyna Technics
Organic Milk Technics
Time: 1 ms
```

Фільтрація по іменах товарів та відповідних відділів:

```
[FILTER MODE]
Input goods name filter:
i
Input departments name filter:
e
iPhone XR Mobile Phones
iPhone 11 Pro Max Mobile Phones
Frosh Washing Home Chemistry
Palmolive Shampoo Home Chemistry
Crocodile Meat
Pencil Vegetables
Oil Oleyna Home Chemistry
Organic Milk Mobile Phones
Molokia Kephir
Vegetables
Milk Organic Meat
Oil Oleyna Mobile Phones
Oil Oleyna Mobile Phones
Pencil Meat
Organic Milk Mobile Phones
Time: 1 ms
```

## TASK 4

Для демонстрації візьмемо модель товару:

```
use postgres::Row;

#[derive(Debug)]
pub struct Good {
    pub goods_id      : i32,
    pub good_name      : String,
    pub departments_id : i32,
    pub categories_id  : i32,
    pub table_name     : String
}

impl From<Row> for Good {
    fn from(row: Row) -> Self {
        Self {
            goods_id      : row.get("goods_id"),
            good_name      : row.get("good_name"),
            departments_id : row.get("departments_id"),
            categories_id  : row.get("categories_id"),
            table_name     : String::from("Goods")
        }
    }
}

impl<'a> Good {
    pub fn get_columns() -> Vec<&'a str> {
        let mut v = Vec::new();
        v.push("good_name");
        v.push("departments_id");
        v.push("categories_id");
        v.push("goods_id");
        v
    }
}
```

Picture 1. good.rs

```

use crate::entities::*;
use postgres::Client;

pub trait Create<T> {
    fn create(new : &T, connection : &mut Client) -> bool;
}

impl Create<Good> for Good {
    fn create(new : &Good, connection : &mut Client) -> bool {
        let query = format!(
            "INSERT INTO public.\"{table}\" VALUES ('{values}') returning *;",
            table = new.table_name,
            values = String::from(&new.good_name) + "\", "
                + &new.departments_id.to_string() + ", "
                + &new.categories_id.to_string()
        );

        connection.query(&query[..], &[]).is_ok()
    }
}

```

Picture 2. create.rs

```

use crate::entities::Good;

// Viewing of changes in database with entity Good

pub fn display_deleted_good(id : i32, is_deleted : bool) {
    println!(
        "Good {:#?} was{is_deleted} deleted",
        id = id,
        is_deleted = if !is_deleted { "n't" } else { "" }
    );
}

pub fn display_new_good(good : &Good) {
    println!("New good: {:#?}", good);
}

pub fn display_good_by_id(good : &Good) {
    println!("Good by id = {}: {:#?}", good.goods_id, good);
}

pub fn display_updated_good(good : &Good) {
    println!("Good was updated successfully: {:#?}", good);
}

```

Picture 3. goods\_view.rs



```

#[allow(unused)]
pub fn create(self : &mut Self) {
    let mut new_name = String::new();
    let mut dep = String::new();
    let mut cat = String::new();

    println!("[INSERT MODE]\n Name:");
    stdin().read_line(&mut new_name);
    println!(" Department id:");
    stdin().read_line(&mut dep);
    println!(" Category id:");
    stdin().read_line(&mut cat);

    let good = Good {
        good_name : String::from(new_name.trim()),
        categories_id : cat.trim().parse::<i32>().unwrap_or(1),
        departments_id : dep.trim().parse::<i32>().unwrap_or(1),
        goods_id : 0,
        table_name : String::from("Goods")
    };

    let is_ok = Good::create(&good, &mut self.connection);

    if is_ok {
        crate::views::display_new_good(&good);
    } else {
        crate::views::display_err();
    }

    stop();
}

```

Picture 4. goods\_controller.rs

## **ВИСНОВОК**

Здобули вміння щодо програмування прикладних додатків баз даних PostgreSQL.