Uniwersytet Rzeszowski
Kolegium Nauk Przyrodniczych
Instytut Informatyki

Bazy danych

Baza danych portalu z ogłoszeniami kupna i sprzedaży

Prowadzący:
dr hab. Barbara Pękala, prof. UR

Autor:
Michał Pasiecznik
nr albumu: 123004
Kierunek: IiE/Ist/2021/22

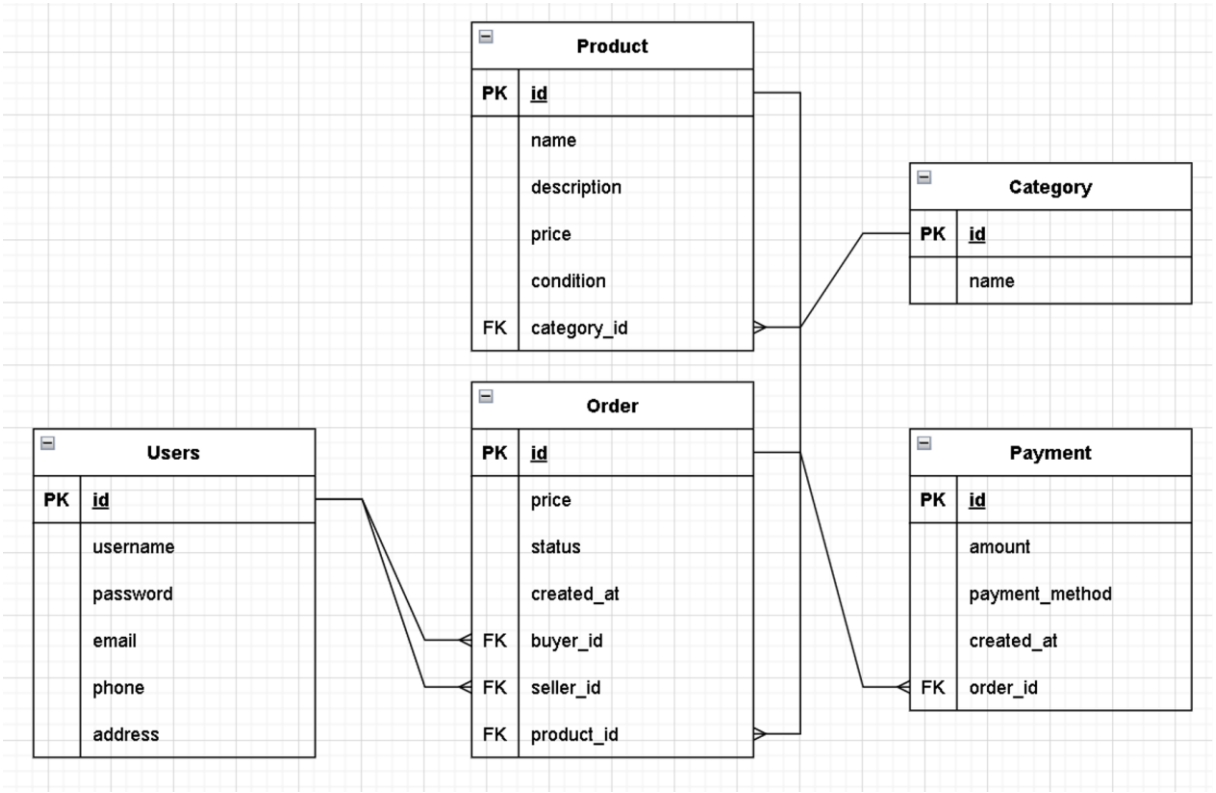
Rzeszów 2023

Spis treści

1. Opis założeń projektu

Baza danych dla portalu z ogłoszeniami kupna i sprzedaży powinna zawierać informacje o produktach oraz o użytkownikach tego portalu. Dodatkowo produkty są posortowane według kategorii. Najważniejsza jest tabela zawierająca informację o ofertach, połączona jest z użytkownikami, płatnościami oraz produktami.

2. Schemat ERD i jego wyjaśnienie



Rysunek 1. Schemat ERD

Wyjaśnienie schematu ERD:

Każdy użytkownik ma swój unikalny identyfikator (id), nazwę użytkownika, hasło, adres e-mail, numer telefonu i adres dostawy.

Każdy produkt ma swój unikalny identyfikator (id), nazwę, opis, cenę, stan (nowy/ używany), kategorię i identyfikator sprzedającego.

Każda kategoria ma swój unikalny identyfikator (id) i nazwę.

Każde zamówienie ma swój unikalny identyfikator (id), identyfikatory kupującego, sprzedającego i produktu, cenę, status (np. złożone, opłacone, wysłane, otrzymane, zwrot) i datę utworzenia.

Każda płatność ma swój unikalny identyfikator (id), identyfikator zamówienia, metodę płatności, kwotę i datę utworzenia.

W tym schemacie ERD istnieją związki między encjami:

Produkty są przypisane do kategorii.

Zamówienia są powiązane z użytkownikami jako kupujący i sprzedający, a także z produktami.

Płatności są powiązane z zamówieniami.

3. Tabele bazy danych

	Tabela	Działanie	Rekordy	Typ	Metoda porównywania napisów	Rozmiar
☐	category	Przeglądaj Struktura Szukaj Wstaw Opróżnij Usuń	13	InnoDB	utf8mb4_general_ci	16.0 KB
☐	orders	Przeglądaj Struktura Szukaj Wstaw Opróżnij Usuń	5	InnoDB	utf8mb4_general_ci	64.0 KB
☐	payment	Przeglądaj Struktura Szukaj Wstaw Opróżnij Usuń	5	InnoDB	utf8mb4_general_ci	32.0 KB
☐	product	Przeglądaj Struktura Szukaj Wstaw Opróżnij Usuń	13	InnoDB	utf8mb4_general_ci	32.0 KB
☐	users	Przeglądaj Struktura Szukaj Wstaw Opróżnij Usuń	13	InnoDB	utf8mb4_general_ci	16.0 KB
	5 tabele	Suma	49	InnoDB	utf8mb4_general_ci	160.0 KB

Rysunek 2. Wszystkie tabele

id	name
1	Elektronika
2	Moda
3	Dom i ogród
4	Supermarket
5	Dziecko
6	Uroda
7	Zdrowie
8	Kultura i rozrywka
9	Sport i turystyka
10	Motoryzacja
11	Nieruchomości
12	Kolekcje i sztuka
13	Firma i usługi

Rysunek 3. Dane tabeli „category”

id	buyer_id	seller_id	product_id	price	status	creation_date
1	4	1	4	1499.99	paid	2023-05-23 14:37:29
2	5	3	5	199.00	shipped	2023-05-23 14:37:29
3	2	4	6	99.99	shipped	2023-05-23 14:37:29
4	1	5	7	2.99	received	2023-05-23 14:37:29
5	3	2	8	49.99	return	2023-05-23 14:37:29

Rysunek 4. Dane tabeli „orders”

id	order_id	payment_method	is_successful	creation_date
1	4	Karta debetowa	1	2023-05-23 14:37:29
2	5	Przelew bankowy	1	2023-05-23 14:37:29
3	2	Płatność online	1	2023-05-23 14:37:29
4	1	Gotówka	1	2023-05-23 14:37:29
5	3	Karta kredytowa	1	2023-05-23 14:37:29

Rysunek 5. Dane tabeli „payment”

id	name	description	product_condition	category_id
1	Telewizor Samsung	Nowy telewizor 4K o przekątnej 55 cali	new	1
2	Kurtka zimowa	Ciepła kurtka z futerkiem w kolorze czarnym	new	2
3	Odkurzacz bezworkowy	Mocny odkurzacz do sprzątania	used	3
4	Smartfon iPhone 12	Nowy smartfon z ekranem OLED i potrójnym aparatem	new	1
5	Sukienka wieczorowa	Elegancka sukienka z koronkowym wzorem	new	2
6	Kosiarka elektryczna	Efektywna kosiarka zasilana prądem	used	3
7	Mleko UHT	Półtłuste mleko w kartonie	new	4
8	Zabawka interaktywna	Zestaw klocków do budowania z dźwiękami i światłam...	new	5
9	Krem nawilżający	Nawilżający krem do twarzy o działaniu antyoksydac...	new	6
10	Termometr elektroniczny	Precyzyjny termometr cyfrowy z wyświetlaczem LCD	new	7
11	Bilety na koncert	Bilety na koncert zespołu rockowego	new	8
12	Namiot turystyczny	Wytrzymały namiot dla dwóch osób	used	9
13	Opona letnia	Opona samochodowa letnia o rozmiarze 195/65 R15	new	10

Rysunek 6. Dane tabeli „product”

id	username	password	email	phone_number	address
1	JanKowalski	haslo123	jan.kowalski@example.com	123456789	ul. Przykładowa 1
2	AnnaNowak	test456	anna.nowak@example.com	987654321	ul. Inna 2
3	PiotrZielinski	passwd789	piotr.zielinski@example.com	111222333	ul. Różana 3
4	AdamKowalski	haslo123	adam.kowalski@example.com	111111111	ul. Kwiatowa 1
5	EwaNowak	test456	ewa.nowak@example.com	222222222	ul. Słoneczna 2
6	MartaWójcik	passwd789	marta.wojcik@example.com	333333333	ul. Parkowa 3
7	PiotrKwiatkowski	test123	piotr.kwiatkowski@example.com	444444444	ul. Polna 4
8	KarolinaKaczmarek	pass456	karolina.kaczmarek@example.com	555555555	ul. Leśna 5
9	MarcinNowicki	test789	marcin.nowicki@example.com	666666666	ul. Ogrodowa 6
10	KatarzynaJankowska	haslo123	katarzyna.jankowska@example.com	777777777	ul. Kwiatowa 7
11	MichałWojciechowski	test456	michal.wojciechowski@example.com	888888888	ul. Słoneczna 8
12	AlicjaKamińska	passwd789	alicja.kaminska@example.com	999999999	ul. Parkowa 9
13	TomaszKowalczyk	test123	tomasz.kowalczyk@example.com	000000000	ul. Polna 10

Rysunek 7. Dane tabeli „users”

4. Przedstawienie od strony kodu bazy danych

```sql
CREATE DATABASE IF NOT EXISTS projektbd;

USE projektbd;

CREATE TABLE Users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(255) NOT NULL,
    password VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL,
    phone_number VARCHAR(20) NOT NULL,
    address VARCHAR(255) NOT NULL
);

CREATE TABLE Category (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL
);

CREATE TABLE Product (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    description TEXT,
    product_condition ENUM('new', 'used') NOT NULL,
    category_id INT,
    price DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (category_id) REFERENCES Category(id)
);

CREATE TABLE Orders (
    id INT AUTO_INCREMENT PRIMARY KEY,
    buyer_id INT,
    seller_id INT,
    product_id INT,
    price DECIMAL(10, 2) NOT NULL,
    status ENUM('placed', 'paid', 'shipped', 'received', 'return'),
    creation_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (buyer_id) REFERENCES Users(id),
    FOREIGN KEY (seller_id) REFERENCES Users(id),
    FOREIGN KEY (product_id) REFERENCES Product(id)
);

CREATE TABLE Payment (
    id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT,
    payment_method VARCHAR(255) NOT NULL,
    is_successful BOOLEAN NOT NULL,
    creation_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (order_id) REFERENCES Orders(id)
);
```

Rysunek 8. Kod pozwalający utworzyć tabele bazy danych

```sql
USE projektbd;

-- Dodawanie przykładowych danych do tabeli Users
INSERT INTO Users (username, password, email, phone_number, address)
VALUES ('JanKowalski', 'haslo123', 'jan.kowalski@example.com', '123456789', 'ul. Przykładowa 1'),
       ('AnnaNowak', 'test456', 'anna.nowak@example.com', '987654321', 'ul. Inna 2'),
       ('PiotrZielinski', 'passwd789', 'piotr.zielinski@example.com', '111222333', 'ul. Różana 3'),
       ('AdamKowalski', 'haslo123', 'adam.kowalski@example.com', '111111111', 'ul. Kwiatowa 1'),
       ('EwaNowak', 'test456', 'ewa.nowak@example.com', '222222222', 'ul. Słoneczna 2'),
       ('MartaWójcik', 'passwd789', 'marta.wojcik@example.com', '333333333', 'ul. Parkowa 3'),
       ('PiotrKwiatkowski', 'test123', 'piotr.kwiatkowski@example.com', '444444444', 'ul. Polna 4'),
       ('KarolinaKaczmarek', 'pass456', 'karolina.kaczmarek@example.com', '555555555', 'ul. Leśna 5'),
       ('MarcinNowicki', 'test789', 'marcin.nowicki@example.com', '666666666', 'ul. Ogrodowa 6'),
       ('KatarzynaJankowska', 'haslo123', 'katarzyna.jankowska@example.com', '777777777', 'ul. Kwiatowa 7'),
       ('MichałWojciechowski', 'test456', 'michal.wojciechowski@example.com', '888888888', 'ul. Słoneczna 8'),
       ('AlicjaKamińska', 'passwd789', 'alicja.kaminska@example.com', '999999999', 'ul. Parkowa 9'),
       ('TomaszKowalczyk', 'test123', 'tomasz.kowalczyk@example.com', '000000000', 'ul. Polna 10');

-- Dodawanie przykładowych danych do tabeli Category
INSERT INTO Category (name)
VALUES ('Elektronika'),
       ('Moda'),
       ('Dom i ogród'),
       ('Supermarket'),
       ('Dziecko'),
       ('Uroda'),
       ('Zdrowie'),
       ('Kultura i rozrywka'),
       ('Sport i turystyka'),
       ('Motoryzacja'),
       ('Nieruchomości'),
       ('Kolekcje i sztuka'),
       ('Firma i usługi');

-- Dodawanie przykładowych danych do tabeli Product
INSERT INTO Product (name, description, product_condition, price, category_id)
VALUES ('Telewizor Samsung', 'Nowy telewizor 4K o przekątnej 55 cali', 'new', 1999.99, 1),
       ('Kurtka zimowa', 'Ciepła kurtka z futerkiem w kolorze czarnym', 'new', 209.49, 2),
       ('Odkurzacz bezworkowy', 'Mocny odkurzacz do sprzątania', 'used', 299.99, 3),
       ('Smartfon iPhone 12', 'Nowy smartfon z ekranem OLED i potrójnym aparatem', 'new', 1499.99, 1),
       ('Sukienka wieczorowa', 'Elegancka sukienka z koronkowym wzorem', 'new', 199.00, 2),
       ('Kosiarka elektryczna', 'Efektywna kosiarka zasilana prądem', 'used', 99.99, 3),
       ('Mleko UHT', 'Półtłuste mleko w kartonie', 'new', 2.99, 4),
       ('Zabawka interaktywna', 'Zestaw klocków do budowania z dźwiękami i światłami', 'new', 49.99, 5),
       ('Krem nawilżający', 'Nawilżający krem do twarzy o działaniu antyoksydacyjnym', 'new', 5.98, 6),
       ('Termometr elektroniczny', 'Precyzyjny termometr cyfrowy z wyświetlaczem LCD', 'new', 129.99, 7),
       ('Bilety na koncert', 'Bilety na koncert zespołu rockowego', 'new', 30.00, 8),
       ('Namiot turystyczny', 'Wytrzymały namiot dla dwóch osób', 'used', 429.39, 9),
       ('Opona letnia', 'Opona samochodowa letnia o rozmiarze 195/65 R15', 'new', 80.00, 10);


-- Dodawanie przykładowych danych do tabeli Orders
INSERT INTO Orders (buyer_id, seller_id, product_id, price, status)
VALUES (4, 1, 4, 1499.99, 'placed'),
       (5, 3, 5, 199.00, 'paid'),
       (2, 4, 6, 99.99, 'shipped'),
       (1, 5, 7, 2.99, 'received'),
       (3, 2, 8, 49.99, 'return');

-- Dodawanie przykładowych danych do tabeli Payment
INSERT INTO Payment (order_id, payment_method, is_successful)
VALUES (4, 'Karta debetowa', true),
       (5, 'Przelew bankowy', true),
       (2, 'Płatność online', true),
       (1, 'Gotówka', true),
       (3, 'Karta kredytowa', true);
```

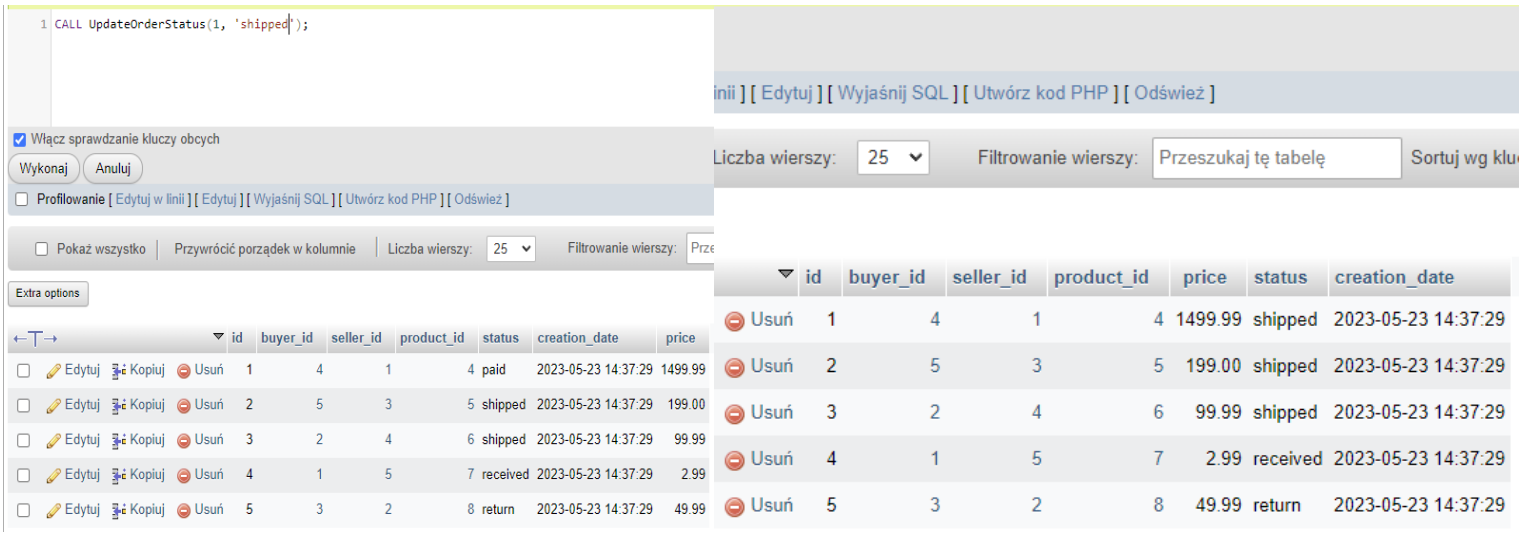
Rysunek 9. Kod pozwalający dodać dane do tabel bazy danych

5. Procedury i funkcje

Procedura UpdateOrderStatus przyjmuje dwa parametry: orderId - identyfikator zamówienia oraz newStatus - nowy status, który chcesz ustawić.
Procedura sprawdza aktualny status zamówienia i wykonuje odpowiednie aktualizacje tylko wtedy, gdy przejście między statusami jest poprawne.
W przeciwnym razie zwracane jest odpowiednie komunikaty o błędzie.

```
DELIMITER //
CREATE PROCEDURE UpdateOrderStatus(IN orderId INT, IN newStatus ENUM('placed', 'paid', 'shipped', 'received', 'return'))
BEGIN
    DECLARE currentStatus ENUM('placed', 'paid', 'shipped', 'received', 'return');
    SELECT status INTO currentStatus FROM Orders WHERE id = orderId;

    IF currentStatus = 'placed' THEN
        IF newStatus = 'paid' OR newStatus = 'shipped' THEN
            UPDATE Orders
            SET status = newStatus
            WHERE id = orderId;
        ELSE
            SELECT 'Invalid status transition.';
        END IF;

    ELSEIF currentStatus = 'paid' THEN
        IF newStatus = 'shipped' OR newStatus = 'received' OR newStatus = 'return' THEN
            UPDATE Orders
            SET status = newStatus
            WHERE id = orderId;
        ELSE
            SELECT 'Invalid status transition.';
        END IF;

    ELSEIF currentStatus = 'shipped' THEN
        IF newStatus = 'received' OR newStatus = 'return' THEN
            UPDATE Orders
            SET status = newStatus
            WHERE id = orderId;
        ELSE
            SELECT 'Invalid status transition.';
        END IF;

    ELSEIF currentStatus = 'received' THEN
        IF newStatus = 'return' THEN
            UPDATE Orders
            SET status = newStatus
            WHERE id = orderId;
        ELSE
            SELECT 'Invalid status transition.';
        END IF;

    ELSE
        SELECT 'Invalid current status.';
    END IF;

END //
DELIMITER ;
```

Rysunek 10. Kod pozwalający utworzyć procedurę aktualizującą status zamówienia

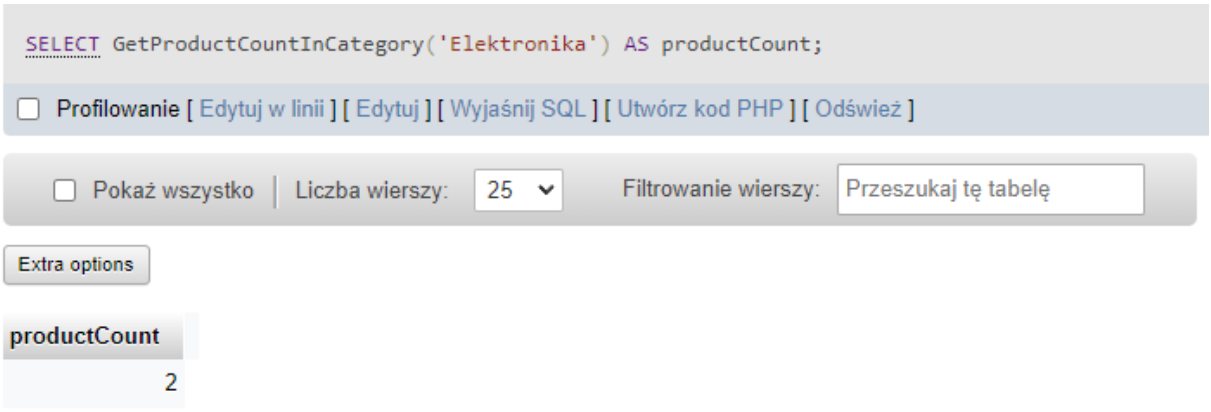


Rysunek 11. Widok tabeli „orders” przed i po wykonaniu procedury UpdateOrderStatus

Funkcja GetProductCountInCategory ma za zadanie zwrócić liczbę produktów w danej kategorii.

```
DELIMITER //
CREATE FUNCTION GetProductCountInCategory(categoryName VARCHAR(255)) RETURNS INT
BEGIN
    DECLARE categoryId INT;
    DECLARE productCount INT;

    SELECT id INTO categoryId FROM Category WHERE name = categoryName;

    SELECT COUNT(*) INTO productCount FROM Product WHERE category_id = categoryId;

    RETURN productCount;
END //
DELIMITER ;
```

Rysunek 12. Kod pozwalający utworzyć funkcję liczącą produkty w kategorii

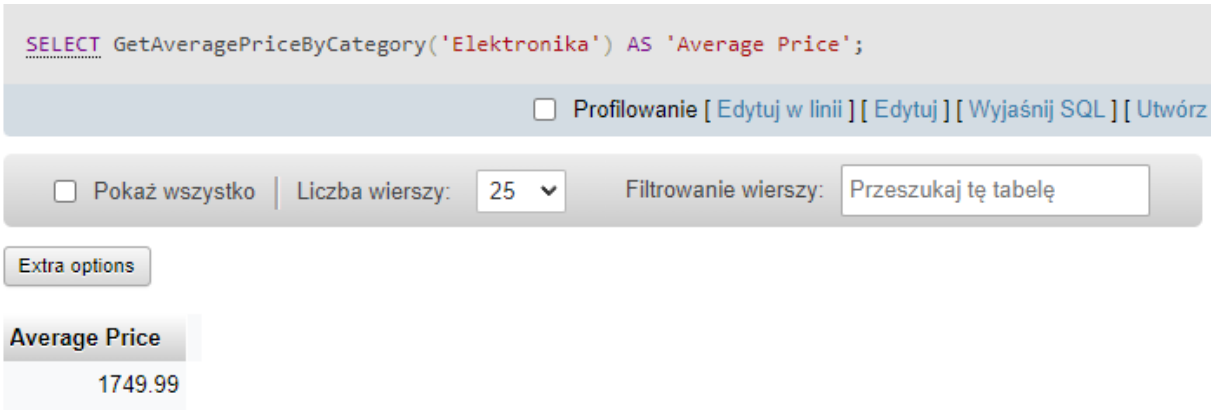


Rysunek 13. Widok ukazujący wynik wykonania funkcji GetProductCountInCategory

Funkcja GetAveragePriceByCategory ma za zadanie zwrócić średnią cenę produktów w danej kategorii.

```
DELIMITER //

CREATE FUNCTION GetAveragePriceByCategory(categoryName VARCHAR(255)) RETURNS DECIMAL(10, 2)
BEGIN
    DECLARE avg_price DECIMAL(10, 2);
    DECLARE categoryId INT;

    SELECT id INTO categoryId FROM Category WHERE name = categoryName;

    SELECT AVG(price) INTO avg_price
    FROM Product
    WHERE category_id = categoryId;

    RETURN avg_price;
END //

DELIMITER ;
```

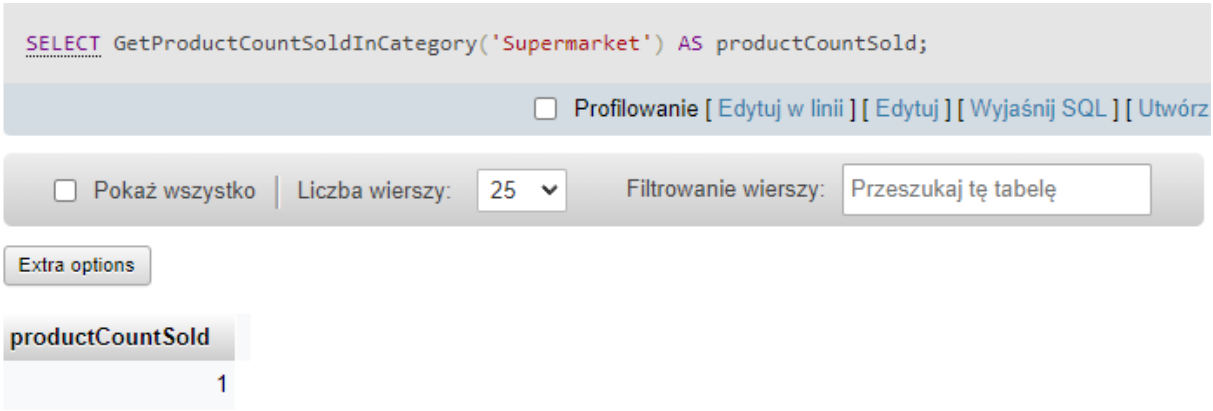
Rysunek 14. Kod pozwalający utworzyć funkcję liczącą średnią cenę produktów w kategorii



Rysunek 15. Widok ukazujący wynik wykonania funkcji GetAveragePriceByCategory

```sql
DELIMITER //

CREATE FUNCTION GetProductCountSoldInCategory(categoryName VARCHAR(255)) RETURNS INT
BEGIN
    DECLARE categoryId INT;
    DECLARE productCountSold INT;

    -- Pobranie identyfikatora kategorii na podstawie nazwy
    SELECT id INTO categoryId FROM Category WHERE name = categoryName;

    -- Zliczanie ilości sprzedanych produktów w danej kategorii
    SELECT COUNT(*) INTO productCountSold
    FROM Orders
    INNER JOIN Product ON Orders.product_id = Product.id
    WHERE Product.category_id = categoryId
    AND Orders.status = 'received';

    RETURN productCountSold;
END //

DELIMITER ;




SELECT GetProductCountSoldInCategory('Supermarket') AS productCountSold;
```

Rysunek 16. Kod pozwalający utworzyć funkcję liczącą sprzedane produkty w kategorii



Rysunek 17. Widok ukazujący wynik wykonania funkcji GetProductCountSoldInCategory

6. Specyfikacja techniczna

Do wykonania projektu użyto:

XAMPP Control Panel v3.3.0. – środowisko programistyczne

MySQL – silnik bazy danych

Apache – serwer http

Notepad++ - edytor tekstu