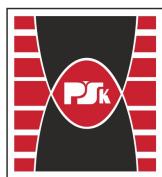


# Projekt aplikacji klienckiej do obsługi bazy danych

mgr inż. Bartłomiej Bugara



Politechnika Świętokrzyska  
Kielce University of Technology

## Spis treści

<b>1 Wstęp</b>	<b>2</b>
<b>2 Wybór technologii</b>	<b>2</b>
<b>3 Funkcjonalności aplikacji</b>	<b>2</b>
3.1 Obsługa pracowników . . . . .	2
3.2 Obsługa towarów . . . . .	2
3.3 Obsługa sprzedaży . . . . .	3
<b>4 Struktura aplikacji</b>	<b>3</b>
4.1 Warstwa prezentacji . . . . .	3
4.2 Warstwa logiki biznesowej . . . . .	3
4.3 Warstwa dostępu do danych . . . . .	3
<b>5 Przykładowy kod</b>	<b>4</b>
<b>6 Uwagi końcowe</b>	<b>4</b>
<b>7 Klient bez aplikacji</b>	<b>4</b>
<b>8 Punktacja projektu</b>	<b>5</b>

# 1 Wstęp

Celem mini-projektu jest stworzenie klienta w postaci aplikacji umożliwiającej pełną obsługę bazy danych na wybrany przez zespół temat oraz zaprezentowanie wszystkich instrukcji SQL, w tym:

- instrukcji SELECT,
- funkcji i procedur składowanych,
- instrukcji DML (INSERT, UPDATE, DELETE),
- instrukcji DDL (CREATE, ALTER, DROP).

Aplikacja powinna umożliwiać wykonywanie wszystkich typowych czynności charakterystycznych dla działalności firmy lub organizacji, której dotyczy baza danych.

# 2 Wybór technologii

Nie ma obowiązku używania bazy danych Oracle. Możliwe użyć przykładowo:

- **PHP + MySQL** (np. aplikacja webowa),
- **C# + SQL Server** (np. aplikacja desktopowa),
- **Java + PostgreSQL** (np. aplikacja desktopowa lub webowa),
- inne połączenia technologiczne umożliwiające komunikację z bazą danych.

# 3 Funkcjonalności aplikacji

Przykładowe funkcjonalności dla sklepu:

## 3.1 Obsługa pracowników

- Dodawanie nowego pracownika,
- Usuwanie pracownika,
- Modyfikacja danych pracownika,
- Wyświetlanie listy pracowników.

## 3.2 Obsługa towarów

- Dodawanie nowego towaru,
- Usuwanie towaru,
- Modyfikacja danych towaru,
- Wyświetlanie listy towarów.

### **3.3 Obsługa sprzedaży**

- Wystawianie faktury z dowolną liczbą pozycji,
- Wystawianie paragonu,
- Generowanie raportów sprzedaży.

## **4 Struktura aplikacji**

### **4.1 Warstwa prezentacji**

Aplikacja powinna posiadać interfejs użytkownika umożliwiający:

- wprowadzanie danych (formularze),
- wyświetlanie wyników zapytań,
- wybór operacji (menu, przyciski).

### **4.2 Warstwa logiki biznesowej**

Logika aplikacji powinna realizować:

- sprawdzanie poprawności danych,
- wywoływanie odpowiednich procedur SQL,
- obsługę transakcji.

### **4.3 Warstwa dostępu do danych**

Warstwa dostępu do bazy danych realizuje:

- połączenie z bazą danych,
- wykonywanie zapytań SQL,
- obsługę błędów.

## 5 Przykładowy kod

Poniżej przykład fragmentu kodu PHP do dodania nowego pracownika w bazie MySQL:

```
1 <?php
2 $servername = "localhost";
3 $username = "root";
4 $password = "";
5 $dbname = "sklep";
6
7 $conn = new mysqli($servername, $username, $password, $dbname);
8
9 if ($conn->connect_error) {
10     die("Connection failed: " . $conn->connect_error);
11 }
12
13 $imie = "Jan";
14 $nazwisko = "Kowalski";
15 $stanowisko = "Sprzedawca";
16
17 $sql = "INSERT INTO pracownicy (imie, nazwisko, stanowisko)
18             VALUES ('$imie', '$nazwisko', '$stanowisko')";
19
20 if ($conn->query($sql) === TRUE) {
21     echo "Nowy pracownik został dodany";
22 } else {
23     echo "Błąd: " . $sql . "<br>" . $conn->error;
24 }
25
26 $conn->close();
27 ?>
```

Listing 1: Dodawanie pracownika w PHP + MySQL

Analogicznie można przygotować instrukcje do:

- modyfikacji danych pracownika (UPDATE),
- usunięcia pracownika (DELETE),
- wyświetlania listy pracowników (SELECT).

## 6 Uwagi końcowe

Wszystkie skrypty powinny zapewniać pełną funkcjonalność typową dla działalności firmy lub organizacji obsługiwanej przez bazę danych. Konieczne jest także wykorzystanie instrukcji SQL stworzonych w poprzednich pracach domowych, w tym funkcji i procedur składowanych.

## 7 Klient bez aplikacji

Alternatywnie, na niższą ocenę można wykonać podstawowe skrypty konsolowe (.bat), odpowiednio powinny one łączyć się z bazą danych, tworzyć tabele oraz mieć możliwość wpisywania danych i wyboru opcji, po której wybraniu na bazie wykona się odpowiednie polecenie SQL.

## 8 Punktacja projektu

Projekt jest oceniany w skali 30 punktów według następujących kryteriów:

- **Poprawność działania aplikacji** – 10 pkt

Aplikacja działa poprawnie, umożliwia dodawanie, modyfikowanie, usuwanie i wyświetlanie danych.

- **Poprawność instrukcji SQL** – 8 pkt

Wykorzystanie wszystkich rodzajów instrukcji SQL stworzonych w poprzednich pracach domowych: SELECT, INSERT, UPDATE, DELETE, funkcje i procedury.

- **Interfejs użytkownika i ergonomia** – 4 pkt

Aplikacja posiada czytelny i intuicyjny interfejs, umożliwia wygodną obsługę wszystkich funkcjonalności.

- **Struktura aplikacji i modularność** – 4 pkt

Poprawne rozdzielenie warstwy prezentacji, logiki biznesowej i dostępu do danych.

- **Dokumentacja i prezentacja projektu** – 4 pkt

Kompletny opis projektu, zawierający wszystkie wymagane sekcje, zrzuty ekranu, przykłady kodu i instrukcje SQL.

**Łącznie: 30 pkt**