



POLITECHNIKA  
LUBELSKA  
WYDZIAŁ ELEKTROTECHNIKI  
I INFORMATYKI

# Zarządzanie bazami SQL i NoSQL

Sprawozdanie z projektu

Kacper Sokół

nr albumu: XXXXX

Michał Sołtys

nr albumu: XXXXX

studia stacjonarne, 3 rok

gr. 6.12, piątek godz. 14:15

Prowadzący laboratorium: dr inż. Sylwester Korga

Lublin rok 2025

## **Część 1 Zarządzanie bazami SQL**

### **1. Problem i konieczność zbudowania bazy danych**

#### **1.1. Opis firmy e\_kromka**

Firma e\_kromka to nowoczesna, internetowa piekarnia, która łączy tradycyjne receptury z innowacyjnymi rozwiązaniami technologicznymi, aby sprostać rosnącym oczekiwaniom klientów. Naszą misją jest dostarczanie najwyższej jakości wypieków, które łączą w sobie smak tradycji oraz komfort zamawiania online. Firma oferuje szeroką gamę wypiekanych chlebków co pozwala na indywidualne dopasowanie oferty do preferencji każdego konsumenta.

#### **1.2. Charakterystyka działalności i potrzeba przetwarzania danych**

W dobie dynamicznego rozwoju technologii, przedsiębiorstwa coraz częściej decydują się na integrację nowoczesnych narzędzi informatycznych, które usprawniają procesy zarządzania oraz obsługi klienta. e\_kromka nie będzie wyjątkiem. W ramach działalności firmy, klienci mają możliwość zakładania personalizowanych kont, co umożliwia im:

Dokonywanie zamówień online z pełnym dostępem do historii zakupów.

Śledzenie statusu realizacji zamówień w czasie rzeczywistym.

Korzystanie z indywidualnych promocji oraz rekomendacji, dopasowanych do wcześniejszych preferencji.

Dodatkowo, system informatyczny e\_kromka pełni kluczową rolę w zarządzaniu zasobami magazynowymi oraz kontrolą składników wykorzystywanych do wypieku produktów. Precyzyjna kontrola stanów magazynowych, harmonogramowanie dostaw surowców oraz monitorowanie terminów przydatności produktów są niezbędne do zachowania wysokiej jakości oferowanych wyrobów. W ramach wspierania systemu magazynowego baza posiadać będzie również kontrolę ze strony przepisów naszych wypieków aby zawsze zapewniać ciągłość produkcji.

Wprowadzenie kompleksowej bazy danych pozwala na:

Automatyzację procesów związanych z zamawianiem i rozliczaniem dostaw.

Usprawnienie komunikacji między działem produkcji a zespołem magazynowym.

Minimalizację ryzyka przestojów produkcyjnych oraz strat wynikających z nieprawidłowego zarządzania zasobami.

#### **1.3. Uzasadnienie budowy bazy danych**

Dynamiczny rozwój sprzedaży online, rosnąca konkurencja na rynku oraz zmieniające się oczekiwania konsumentów wymuszają na firmie e\_kromka konieczność inwestycji w nowoczesne rozwiązania informatyczne. Budowa bazy danych stanowi fundament:

Efektywnej obsługi klientów: Centralizacja informacji umożliwia szybkie i precyzyjne zarządzanie danymi klientów, co przekłada się na lepsze doświadczenie użytkownika.

Optymalizacji procesów produkcyjnych: Monitorowanie stanów magazynowych i kontrola składników pozwala na bieżące reagowanie na zmiany w zapotrzebowaniu oraz planowanie produkcji.

Wspierania rozwoju firmy: Zintegrowany system umożliwia analizę danych, co jest kluczowe przy podejmowaniu strategicznych decyzji oraz planowaniu przyszłych inwestycji.

Podsumowując, wdrożenie kompleksowej bazy danych w firmie e\_kromka nie tylko usprawni codzienne operacje, ale również umożliwi dynamiczny rozwój i elastyczne dostosowywanie się do potrzeb rynku, co jest niezbędne w kontekście dzisiejszej.

## **2. Opis struktury bazy danych**

W niniejszym rozdziale przedstawiono szczegółowy opis struktury bazy danych dla firmy internetowej piekarni e\_kromka. Baza została zaprojektowana z myślą o kompleksowym zarządzaniu procesami związanymi z obsługą klientów, realizacją zamówień, kontrolą stanów magazynowych oraz wsparciem działań operacyjnych firmy. Poniżej znajduje się opis poszczególnych tabel wraz z wyjaśnieniem ich kolumn.

### **2.1. Tabela Konta**

Tabela Konta stanowi fundament systemu zarządzania dostępem. Przechowuje ona dane niezbędne do identyfikacji i autoryzacji użytkowników oraz pracowników, zapewniając unikalność loginów.

id (INT): Klucz główny, automatycznie inkrementowany, identyfikuje unikalnie każde konto.

login (VARCHAR(45)): Unikalny identyfikator użytkownika (login), który musi być unikalny w całej bazie.

haslo (VARCHAR(45)): Hasło przypisane do konta, umożliwiające autoryzację. Pole to może przyjmować wartość NULL.

### **2.2. Tabela Uzytkownicy**

Tabela Uzytkownicy przechowuje dane personalne klientów, umożliwiając indywidualne zarządzanie informacjami oraz personalizację doświadczenia zakupowego.

id (INT): Klucz główny, automatycznie inkrementowany.

imie (VARCHAR(40)): Imię użytkownika.

nazwisko (VARCHAR(40)): Nazwisko użytkownika.

email (VARCHAR(40)): Adres e-mail, który musi być unikalny.

telefon (VARCHAR(40)): Numer telefonu kontaktowego.

Konta\_id (INT): Klucz obcy odnoszący się do tabeli Konta, wskazujący, do którego konta przypisane są dane użytkownika.

### **2.3. Tabela Adresy**

Tabela Adresy przechowuje informacje o adresach dostaw przypisanych do poszczególnych użytkowników, co umożliwia łatwe zarządzanie lokalizacjami wysyłki.

id (INT): Klucz główny, automatycznie inkrementowany.

uzytkownik\_id (INT): Klucz obcy odnoszący się do tabeli Uzytkownicy, wskazujący, któremu użytkownikowi przypisany jest dany adres.

adres (VARCHAR(255)): Szczegółowy opis adresu dostawy.

#### 2.4. Tabela Kategorie

Tabela Kategorie służy do grupowania produktów według określonych cech i umożliwia prezentację oferty w sposób uporządkowany.

id (INT): Klucz główny, automatycznie inkrementowany.

nazwa (VARCHAR(40)): Nazwa kategorii, np. „Chleby”, „Ciasta” lub „Desery”.

data\_poczatku\_wyswietlania (DATE): Data rozpoczęcia wyświetlania danej kategorii, co umożliwia sezonową promocję.

data\_konca\_wyswietlania (DATE): Data zakończenia wyświetlania kategorii, pozwalająca na dynamiczne zarządzanie ofertą.

#### 2.5. Tabela Przepisy

Tabela Przepisy przechowuje informacje o recepturach, które są wykorzystywane do tworzenia produktów piekarniczych. Dzięki tej tabeli możliwe jest zarządzanie procesami produkcyjnymi i standaryzacją jakości wyrobów.

id (INT): Klucz główny, automatycznie inkrementowany.

nazwa (VARCHAR(40)): Nazwa receptury lub przepisu.

opis (TEXT): Szczegółowy opis receptury.

instrukcje (TEXT): Instrukcje przygotowania, krok po kroku, niezbędne dla produkcji.

#### 2.6. Tabela Produkty

Tabela Produkty zawiera pełną ofertę wyrobów piekarniczych dostępnych w sklepie internetowym. Opisuje produkty pod kątem cech takich jak cena, ilość w magazynie czy powiązanie z przepisami i kategoriami.

id (INT): Klucz główny, automatycznie inkrementowany.

nazwa (VARCHAR(40)): Nazwa produktu, np. „Chleb razowy” lub „Ciasto czekoladowe”.

opis (TEXT): Szczegółowy opis produktu.

cena (DECIMAL(10,2)): Cena produktu.

ilosc\_na\_stanie (INT): Ilość dostępna w magazynie.

kategoria\_id (INT): Klucz obcy, który łączy produkt z odpowiednią kategorią w tabeli Kategorie.

Przepisy\_id (INT): Klucz obcy wskazujący na recepturę z tabeli Przepisy, która została wykorzystana przy produkcji.

#### 2.7. Tabela Koszyki

Tabela Koszyki reprezentuje tymczasowe zbiory produktów wybranych przez użytkowników do zakupu. Umożliwia przechowywanie i modyfikację listy produktów przed finalizacją zamówienia.

id (INT): Klucz główny, automatycznie inkrementowany.

uzytkownik\_id (INT): Klucz obcy odnoszący się do tabeli Uzytkownicy, dzięki któremu koszyk jest powiązany z konkretnym klientem.

## 2.8. Tabela PozycjeKoszyka

Tabela PozycjeKoszyka przechowuje poszczególne pozycje znajdujące się w koszyku, czyli produkty wybrane przez klienta wraz z określoną ilością.

id (INT): Klucz główny, automatycznie inkrementowany.

koszyk\_id (INT): Klucz obcy odnoszący się do tabeli Koszyki.

produkt\_id (INT): Klucz obcy odnoszący się do tabeli Produkty.

ilosc (INT): Ilość wybranego produktu.

## 2.9. Tabela Zamowienia

Tabela Zamowienia dokumentuje wszystkie realizowane zamówienia. Zawiera kluczowe informacje dotyczące procesu zakupu, takie jak data zamówienia, suma do zapłaty, czy status realizacji.

id (INT): Klucz główny, automatycznie inkrementowany.

uzytkownik\_id (INT): Klucz obcy odnoszący się do tabeli Uzytkownicy.

data\_zamowienia (DATETIME): Data i godzina złożenia zamówienia.

suma (DECIMAL(10,2)): Łączna wartość zamówienia.

kod\_rabatowy (VARCHAR(40)): Kod rabatowy wykorzystany przy zamówieniu (opcjonalnie).

adres\_dostawy (VARCHAR(255)): Adres, na który zamówienie ma być dostarczone.

metoda\_dostawy (VARCHAR(40)): Wybrana metoda dostawy (np. kurier, odbiór osobisty).

metoda\_platnosci (VARCHAR(40)): Sposób płatności (np. karta, przelew).

status (VARCHAR(40)): Status zamówienia (np. „w trakcie realizacji”, „zrealizowane”).

## 2.10. Tabela PozycjeZamowienia

Tabela PozycjeZamowienia zawiera szczegółowe informacje o produktach zawartych w poszczególnych zamówieniach. Umożliwia analizę zawartości zamówień i rozliczenia finansowe.

id (INT): Klucz główny, automatycznie inkrementowany.

zamowienie\_id (INT): Klucz obcy odnoszący się do tabeli Zamowienia.

produkt\_id (INT): Klucz obcy odnoszący się do tabeli Produkty.

ilosc (INT): Ilość danego produktu w zamówieniu.

cena (DECIMAL(10,2)): Cena produktu w momencie zamówienia.

## 2.11. Tabela Faktury

Tabela Faktury rejestruje dokumenty księgowe powiązane z zamówieniami. Dzięki niej możliwe jest monitorowanie i archiwizacja rozliczeń finansowych.

id (INT): Klucz główny, automatycznie inkrementowany.

zamowienie\_id (INT): Klucz obcy odnoszący się do tabeli Zamowienia.

data\_faktury (DATETIME): Data wystawienia faktury.  
suma (DECIMAL(10,2)): Łączna kwota faktury.

#### 2.12. Tabela Pracownicy

Tabela Pracownicy przechowuje informacje o zatrudnionych w firmie, którzy obsługują zamówienia, zarządzają produkcją oraz kontrolują procesy magazynowe.

id (INT): Klucz główny, automatycznie inkrementowany.  
imie (VARCHAR(40)): Imię pracownika.  
nazwisko (VARCHAR(40)): Nazwisko pracownika.  
email (VARCHAR(40)): Unikalny adres e-mail pracownika.  
telefon (VARCHAR(40)): Numer telefonu kontaktowego.  
stanowisko (VARCHAR(40)): Określenie stanowiska lub roli w organizacji.  
Konta\_id (INT): Klucz obcy odnoszący się do tabeli Konta, dzięki któremu pracownik ma przypisane dane logowania.

#### 2.13. Tabela StanyMagazynowe

Tabela StanyMagazynowe umożliwia bieżące monitorowanie ilości produktów dostępnych w magazynie, co jest kluczowe dla efektywnego zarządzania zasobami.

id (INT): Klucz główny, automatycznie inkrementowany.  
produkt\_id (INT): Klucz obcy odnoszący się do tabeli Produkty.  
ilosc (INT): Aktualna ilość produktu dostępna w magazynie.

#### 2.14. Tabela ZamowieniaDoRealizacji

Tabela ZamowieniaDoRealizacji wspiera proces realizacji zamówień poprzez przypisywanie zamówień do pracowników odpowiedzialnych za ich obsługę oraz monitorowanie statusu realizacji.

id (INT): Klucz główny, automatycznie inkrementowany.  
zamowienie\_id (INT): Klucz obcy odnoszący się do tabeli Zamowienia.  
pracownik\_id (INT): Klucz obcy odnoszący się do tabeli Pracownicy; wskazuje pracownika przypisanego do realizacji zamówienia (pole może przyjmować wartość NULL, jeśli nie został jeszcze przypisany).  
status (VARCHAR(40)): Aktualny status realizacji zamówienia (np. „oczekujące”, „w realizacji”, „zakończone”).

#### 2.15. Podsumowanie struktury

Przedstawiony schemat bazy danych dla firmy e\_kromka odzwierciedla kompleksowe podejście do zarządzania wszystkimi aspektami działalności internetowej piekarni. Podział na wyspecjalizowane tabele umożliwia:

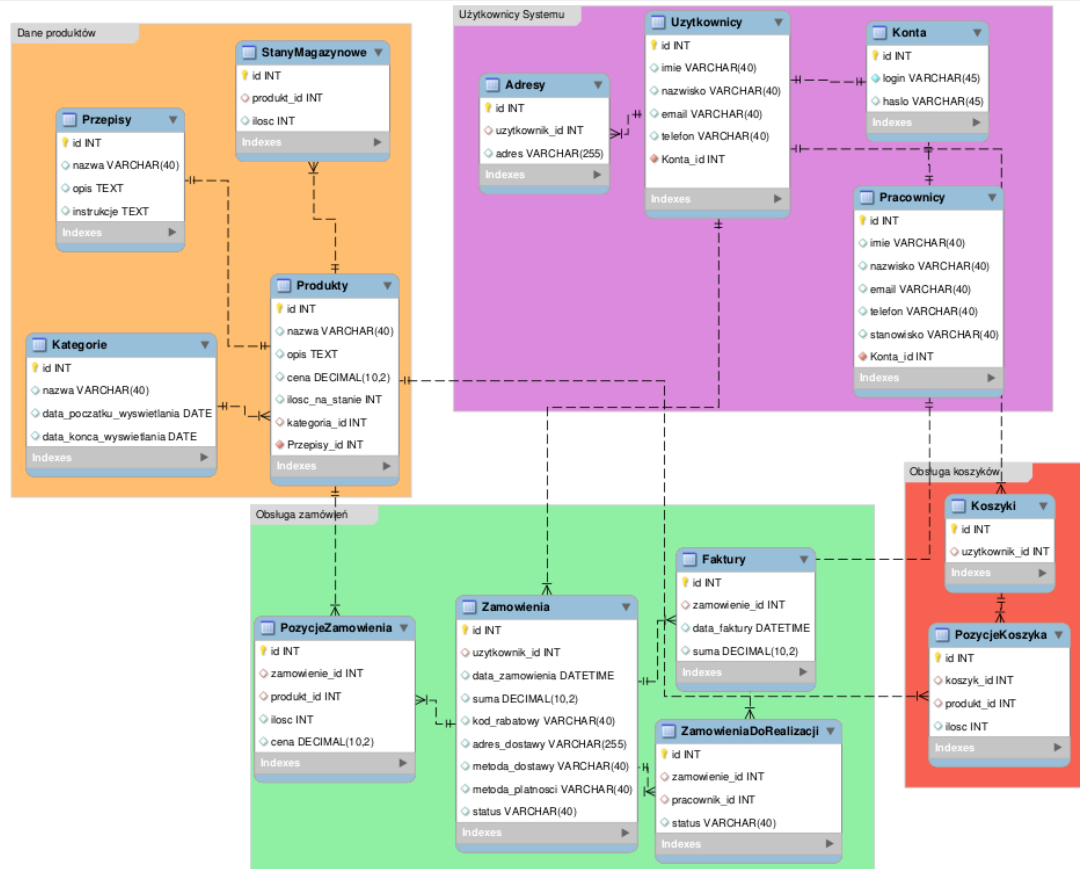
Centralizację informacji dotyczących użytkowników, produktów, zamówień oraz stanów magazynowych.

Elastyczne zarządzanie danymi, co przyczynia się do sprawniejszej obsługi klienta i optymalizacji procesów produkcyjnych.

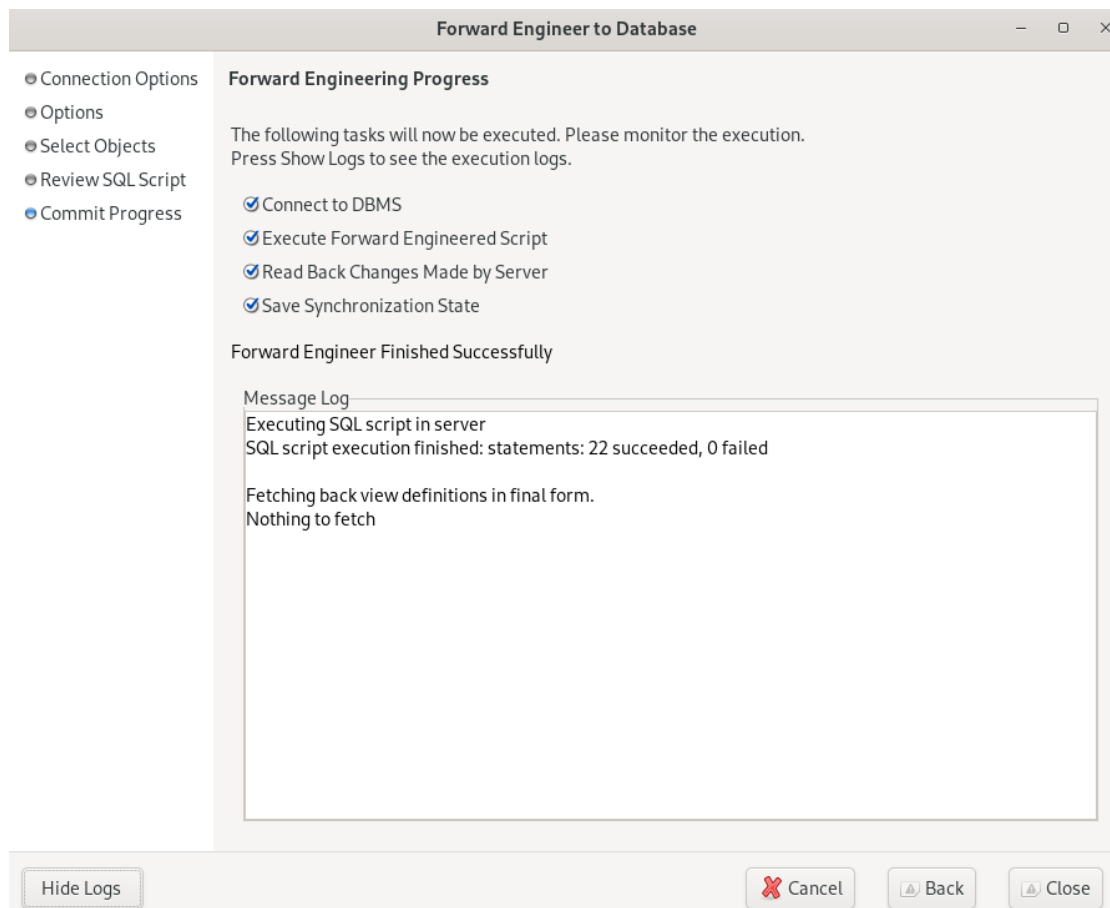
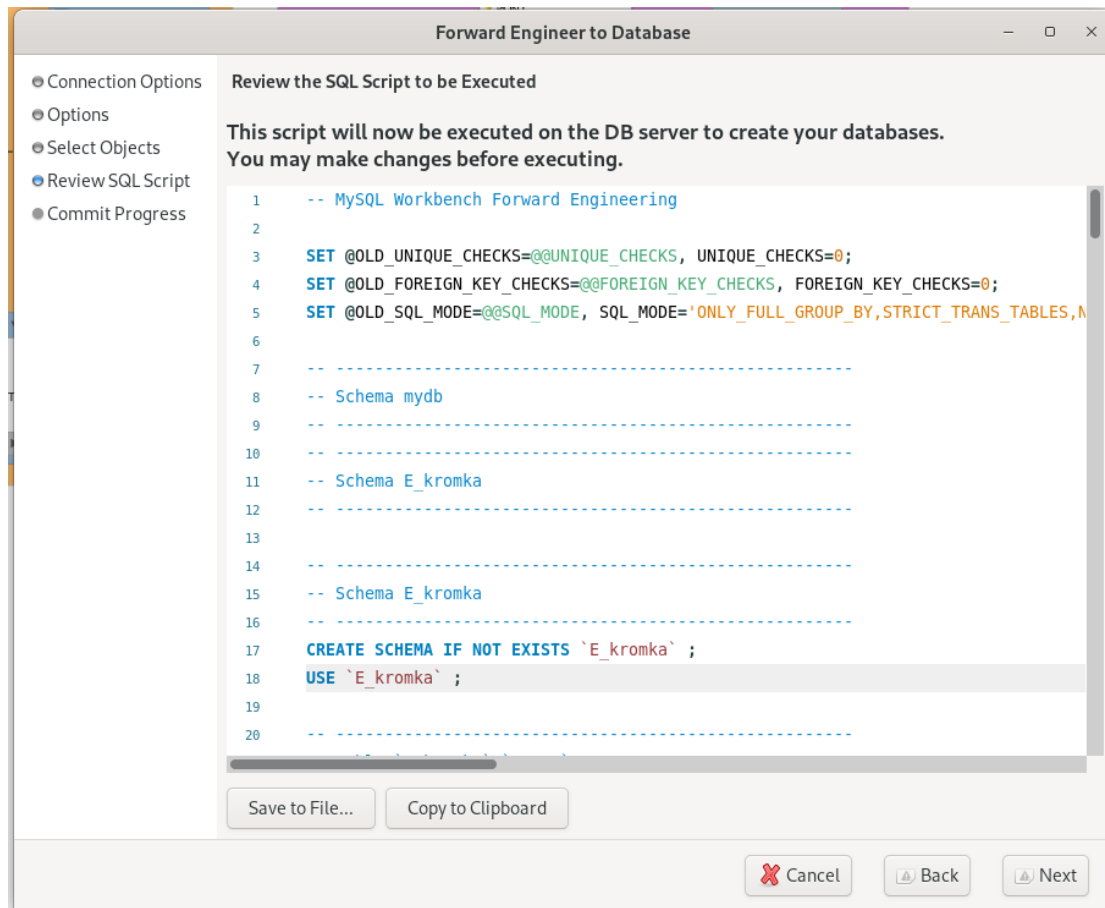
Łatwą integrację między systemami logowania, składania zamówień, realizacji zamówień oraz zarządzania magazynem, co w efekcie wspiera dynamiczny rozwój firmy na rynku usług online.

Tak skonstruowana baza danych stanowi solidny fundament dla wdrożenia systemu informatycznego, który sprosta wymaganiom nowoczesnej piekarni internetowej, umożliwiając zarówno bieżącą operacyjną obsługę, jak i wsparcie strategicznych decyzji biznesowych.

### 3. Tworzenie modelu bazy oraz graficzna interpretacja BD w formie diagramu EER



### 4. Opis procesu generowania skryptu BD





## 5. Opis wypełnienia bazy danych danymi

### -- Wypełnianie tabeli Konta

```
INSERT INTO `E_kromka`.`Konta` (`id`, `login`, `haslo`) VALUES  
(1, 'user1', 'haslo1'),  
(2, 'user2', 'haslo2'),  
(3, 'admin1', 'admin1'),  
(4, 'pracownik1', 'haslopracownika1');
```

### -- Wypełnianie tabeli Uzytkownicy

```
INSERT INTO `E_kromka`.`Uzytkownicy` (`id`, `imie`, `nazwisko`, `email`, `telefon`,  
`Konta_id`) VALUES  
(1, 'Jan', 'Kowalski', 'jan.kowalski@example.com', '123456789', 1),  
(2, 'Anna', 'Nowak', 'anna.nowak@example.com', '987654321', 2);
```

### -- Wypełnianie tabeli Adresy

```
INSERT INTO `E_kromka`.`Adresy` (`id`, `uzytkownik_id`, `adres`) VALUES  
(1, 1, 'ul. Kwiatowa 1, 00-001 Warszawa'),  
(2, 2, 'ul. Lipowa 2, 00-002 Kraków');
```

### -- Wypełnianie tabeli Kategorie

```
INSERT INTO `E_kromka`.`Kategorie` (`id`, `nazwa`, `data_poczatku_wyswietlania`,  
`data_konca_wyswietlania`) VALUES  
(1, 'Chleby regionalne', '2023-01-01', '2023-12-31'),  
(2, 'Bułki', '2023-01-01', '2023-12-31');
```

### -- Wypełnianie tabeli Przepisy

```
INSERT INTO `E_kromka`.`Przepisy` (`id`, `nazwa`, `opis`, `instrukcje`) VALUES  
(1, 'Chleb pszenny', 'Tradycyjny chleb pszenny', 'Wymieszaj składniki, wyrabiaj  
ciasto, piecz przez 30 minut.'),  
(2, 'Bułka kajzerka', 'Basic bułka', 'Wyrabiaj ciasto, formuj bułki, piecz przez 15  
minut.');
```

### -- Wypełnianie tabeli Produkty

```
INSERT INTO `E_kromka`.`Produkty` (`id`, `nazwa`, `opis`, `cena`,  
`ilosc_na_stanie`, `kategoria_id`, `Przepisy_id`) VALUES  
(1, 'Chleb pszenny', 'Świeży chleb pszenny', 5.99, 50, 1, 1),  
(2, 'Bułka kajzerka', 'Basic bułka', 2.50, 300, 2, 2);
```

### -- Wypełnianie tabeli Koszyki

```
INSERT INTO `E_kromka`.`Koszyki` (`id`, `uzytkownik_id`) VALUES  
(1, 1),  
(2, 2);
```

### -- Wypełnianie tabeli PozycjeKoszyka

```
INSERT INTO `E_kromka`.`PozycjeKoszyka` (`id`, `koszyk_id`, `produkt_id`, `ilosc`) VALUES  
(1, 1, 1, 2), -- Koszyk 1, Chleb pszenny, ilość 2  
(2, 2, 2, 5); -- Koszyk 2, Bułka kajzerka, ilość 5
```

-- Wypełnianie tabeli Zamowienia

```
INSERT INTO `E_kromka`.`Zamowienia` (`id`, `uzytkownik_id`, `data_zamowienia`,  
`suma`, `kod_rabatowy`, `adres_dostawy`, `metoda_dostawy`, `metoda_platnosci`,  
`status`) VALUES
```

```
(1, 1, '2023-10-01 12:00:00', 11.98, NULL, 'ul. Kwiatowa 1, 00-001 Warszawa',  
'Kurier', 'Karta', 'Nowe'), -- Zamówienie 1, Jan Kowalski
```

```
(2, 2, '2023-10-02 14:00:00', 12.50, 'RABAT10', 'ul. Lipowa 2, 00-002 Kraków',  
'Poczta', 'Przelew', 'W trakcie'); -- Zamówienie 2, Anna Nowak
```

-- Wypełnianie tabeli PozycjeZamowienia

```
INSERT INTO `E_kromka`.`PozycjeZamowienia` (`id`, `zamowienie_id`, `produkt_id`,  
`ilosc`, `cena`) VALUES
```

```
(1, 1, 1, 2, 5.99), -- Zamówienie 1, Chleb pszenny, ilość 2, cena 5.99
```

```
(2, 2, 2, 5, 2.50); -- Zamówienie 2, Bułka kajzerka, ilość 5, cena 2.50
```

-- Wypełnianie tabeli Faktury

```
INSERT INTO `E_kromka`.`Faktury` (`id`, `zamowienie_id`, `data_faktury`, `suma`) VALUES
```

```
(1, 1, '2023-10-01 12:30:00', 11.98), -- Faktura dla zamówienia 1
```

```
(2, 2, '2023-10-02 14:30:00', 12.50); -- Faktura dla zamówienia 2
```

-- Wypełnianie tabeli Pracownicy

```
INSERT INTO `E_kromka`.`Pracownicy` (`id`, `imie`, `nazwisko`, `email`, `telefon`,  
`stanowisko`, `Konta_id`) VALUES
```

```
(1, 'Piotr', 'Wiśniewski', 'piotr.wisniewski@example.com', '111222333', 'Kierownik',  
3), -- Pracownik 1, konto admin1
```

```
(2, 'Maria', 'Wójcik', 'maria.wojcik@example.com', '444555666', 'Magazynier', 4); --  
Pracownik 2, konto pracownik1
```

-- Wypełnianie tabeli StanyMagazynowe

```
INSERT INTO `E_kromka`.`StanyMagazynowe` (`id`, `produkt_id`, `ilosc`) VALUES
```

```
(1, 1, 50), -- Stan magazynowy dla Chleba pszennego
```

```
(2, 2, 300); -- Stan magazynowy dla Bułki kajzerki
```

-- Wypełnianie tabeli ZamowieniaDoRealizacji

```
INSERT INTO `E_kromka`.`ZamowieniaDoRealizacji` (`id`, `zamowienie_id`,  
`pracownik_id`, `status`) VALUES
```

```
(1, 1, 1, 'W trakcie'), -- Zamówienie 1 przypisane do Piotra Wiśniewskiego
```

```
(2, 2, 2, 'Nowe'); -- Zamówienie 2 przypisane do Marii Wójcik
```

## 6. Proces walidacyjny BD

wypisać co najmniej 15 przykładów kwerend napisanych w języku SQL.

Co najmniej 5 kwerend dotyczy języka DDL

Co najmniej 5 kwerend dotyczy języka DML

Co najmniej 5 kwerend dotyczy języka DCL Kwerendy z języka TCL realizowane są tylko przez osoby chętne.

## **Część 2 Zarządzanie bazami NoSQL**

### **1. Problem i konieczność zbudowania bazy danych**

Może być to opis firmy oraz jej działalności z uwzględnieniem potrzeby przetwarzania danych. Student sam wybiera firmę oraz rodzaj jej działalności (usługowa lub handlowa). Firma oraz jej sposób działalności może być wymyślona przez studenta na potrzeby projektu. W rozdziale drugim może być to ta sama firma co w rozdziale pierwszym lub nowo wybrana firma z nowym problemem.

### **2. Opis struktury bazy danych**

Opis struktury bazy danych zawiera opis jakie kolekcje są potrzebne w bazie oraz opis ich dokumentów

### **3. Graficzna interpretacja BD**

Rozdział nr 3 zawiera graficzną interpretację bazy danych w formie zrzutu ekranu gdzie widoczne są kolekcje.

### **4. Opis projektowania BD**

Treść rozdziału zawiera opis graficznego projektowania bazy oraz wykorzystania konsoli (powłoki MongoSH).

### **5. Opis wypełnienia bazy danych danymi**

Rozdział nr 5 zawiera opis wypełnienie bazy danych danymi. Zawarte są tutaj przykłady uzupełniania bazy „z palca” lub za pomocą „datasetów”

### **6. Kwerendy w języku MongoDB napisane za pomocą powłoki MONGOSH**

Rozdział 6 dotyczy napisania 15 kwerend w języku MongoDB za pomocą powłoki MONGOSH. Zapytania powinny zawierać takie elementy jak filtrowanie danych, projekcja danych, agregacje, łączenie kolekcji itd.

### **Podsumowanie:**

W tym rozdziale znajduje się opis porównawczy dwóch systemów bazodanowych. Opisane są tutaj pozytywne i negatywne aspekty pracy z systemami. Porównanie powinno zawierać opisy możliwości zastosowania systemów dla różnorodnych procesów bazodanowych.