

SKLEP WĘDKARSKI

BAZA DANYCH DOKUMENTACJA

Aleksander Staszków, Kacper Wołczyński,
Igor Tujek, Borys Łukojć

Wstęp

Ta dokumentacja opisuje strukturę i zawartość bazy danych "entervive_sklepwendkarski", która została zaprojektowana w celu zarządzania danymi związanymi z sklepami wędkarskimi, klientami, zamówieniami, produktami, producentami, kategoriami towarów, kurierami oraz regionami.

Struktura Bazy Danych

Baza danych składa się z następujących tabel:

1. kategoria_towaru: Tabela przechowująca informacje o kategoriach towarów oferowanych przez sklepy wędkarskie.
2. klient: Tabela zawierająca dane klientów, takie jak imię, nazwisko, adres, kontakt itp.
3. kurier: Tabela zawierająca informacje o kurierach obsługujących dostawy zamówień.
4. producent: Tabela przechowująca dane producentów oferujących produkty wędkarskie.
5. ranga_klienta: Tabela zawierająca informacje o rangach klientów w sklepach wędkarskich.
6. rejon: Tabela przechowująca dane o regionach, w których znajdują się sklepy wędkarskie.
7. sklep: Tabela zawierająca informacje o sklepach wędkarskich, takie jak adres, kontakt, lokalizacja itp.
8. towar: Tabela przechowująca dane o produktach oferowanych w sklepach wędkarskich, takie jak nazwa, cena, producent itp.
9. zamowienia: Tabela zawierająca informacje o zamówieniach klientów, takie jak data zamówienia, data wysyłki, sklep, kurier, towar itp.

Cel Bazy Danych

Baza danych "entervive_sklepwendkarski" została zaprojektowana w celu efektywnego zarządzania danymi związanymi z działalnością sklepów wędkarskich. Umożliwia ona śledzenie zamówień klientów, zarządzanie produktami, producentami, kurierami oraz informacjami o klientach i regionach, co pozwala sklepom na lepszą organizację działalności handlowej oraz usprawnienie obsługi klienta.

Schemat Tabel w Bazie

Tabele (Przedstawiono Tekstowo)

- kategoria_towaru
 1. id_kategori, int(52), primary key
 2. kategoria, varchar(255)
- towar
 1. id_towaru, int(11), primary key
 2. id_kategori, int(11), foreign key
 3. id_producenta, id(11), foreign key
 4. nazwa, varchar(255)
 5. cena, int(11)
- producent
 1. id_producenta, int(52), primary key
 2. nazwa, varchar(255)
 3. nip, int(52)
 4. adres, varchar(255)
 5. telefon, int(52)
 6. email, varchar(255)
- zamowienia
 1. id_zamowienia, int(52), primary key
 2. data_zamowienia, date
 3. data_wysylki, date

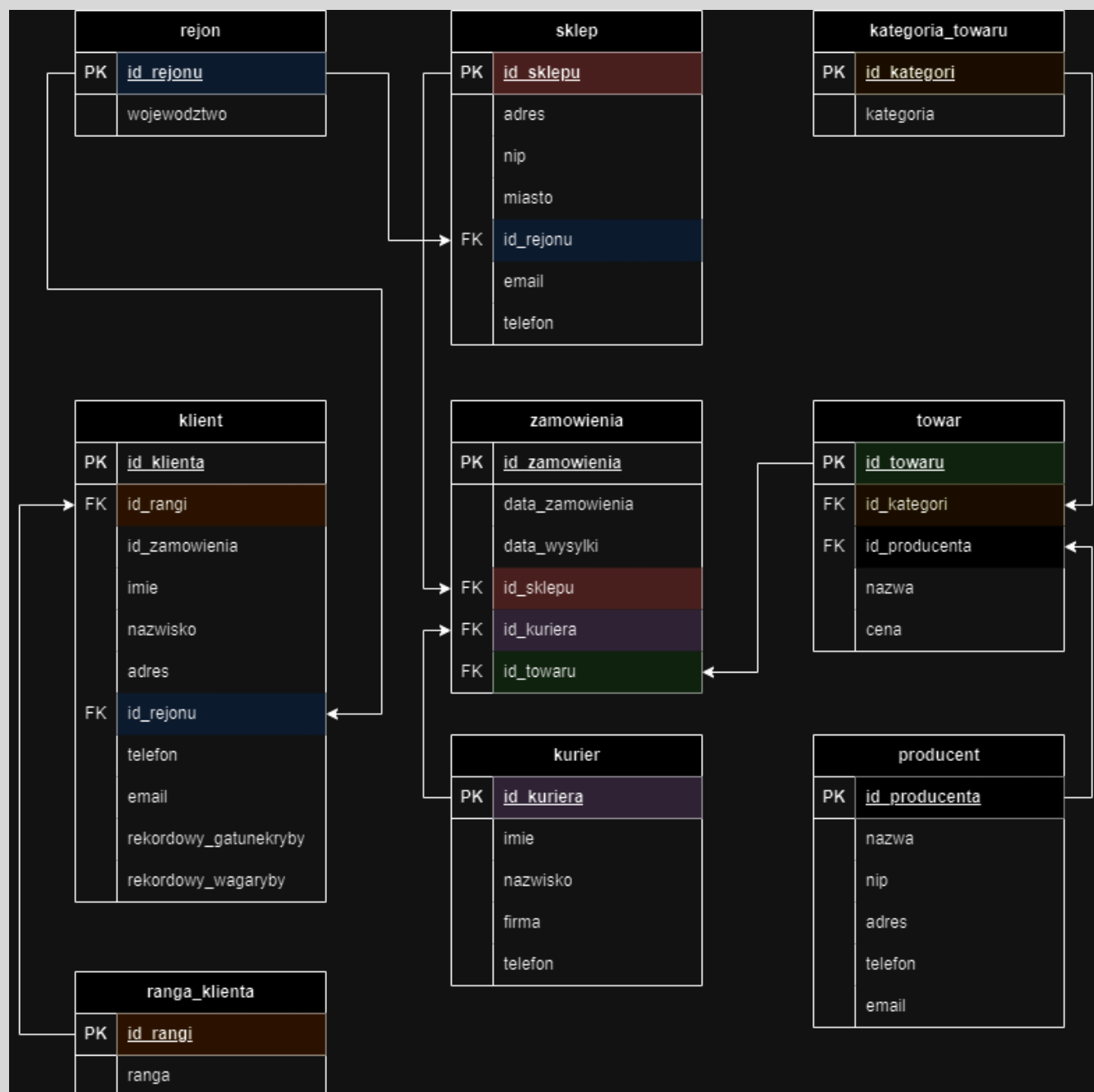
- 4. id_sklepu, int(11), foreign key
- 5. id_kuriera, int(11), foreign key
- 6. id_towaru, int(11), foreign key
- kurier
 - 1. id_kuriera, int(52), primary key
 - 2. imie, varchar(255)
 - 3. nazwisko, varchar(255)
 - 4. firma, varchar(255)
 - 5. telefon, int(52)
- sklep
 - 1. id_sklepu, int(52), primary key
 - 2. adres, varchar(255)
 - 3. nip, int(52)
 - 4. miasto, varchar(52)
 - 5. id_rejonu, int(52), foreign key
 - 6. email, varchar(52)
 - 7. telefon, int(52)
- rejon
 - 1. id_rejonu, int(52), primary key
 - 2. wojewodztwo, varchar(255)
- klient
 - 1. id_klienta, int(52), primary key
 - 2. id_rangi, int(52), foreign key
 - 3. id_zamowienia, int(11), foreign key
 - 4. imie, varchar(255)
 - 5. nazwisko, varchar(255)
 - 6. adres, varchar(255)
 - 7. id_rejonu, int(52), foreign key
 - 8. telefon, int(52)
 - 9. email, varchar(255)
 - 10. rekordowy_gatunekryby, varchar(255)
 - 11. rekordowy_wagaryby, varchar(255)
- ranga_klienta
 - 1. id_rangi, int(52), primary key

2. ranga, varchar(255)

Łączenia Tabel (Przedstawiono tekstowo)

(primary) ranga_klienta.id_rangi → klient.id_rangi (foreign)
(foreign) sklep.id_rangi → ranga_klienta.id_rangi (primary)
(foreign) sklep.id_zamowienia → zamowienia.id_zamowienia (primary)
(foreign) sklep.id_rejonu → rejon.id_rejonu (primary)
(primary) rejon.id_rejonu → sklep.id_rejonu (foreign)
(primary) sklep.id_sklepu → zamowienia.id_sklepu (foreign)
(foreign) sklep.id_rejonu → rejon.id_rejonu (primary)
(primary) zamowienia.id_zamowienia → klient.id_zamowienia (foreign)
(foreign) zamowienia.id_sklepu → sklep.id_sklepu (primary)
(foreign) zamowienia.id_kuriera → kurier.id_kuriera(primary)
(foreign) zamowienia.id_towaru → towar.id_towaru (primary)
(primary) kurier.id_kuriera → zamowienia.id_kuriera (foreign)
(primary) kategoria_towaru.id_kategori → towar.id_kategori (foreign)
(primary) towar.id_towaru → zamowienia.id_towaru (foreign)
(foreign) towar.id_kategori → kategoria_towaru.id_kategori (primary)
(foreign) towar.id_producenta → producent.id_producenta (primary)
(primary) producent.id_producenta → towar.id_producenta (foreign)

Wizualizacja tabeli



Przykładowa procedura dodawanie danych do sklepu

Aby dodać zamówienia do tabeli danych, musimy najpierw uzupełnić inne tabele, których indeksy będą wykorzystywane przez te zamówienia. Zalecam rozpoczęcie od uzupełnienia tabeli `ranga_klienta`. Poniżej znajduje się przykładowy zestaw danych:

```
INSERT INTO `ranga_klienta` (`id_rangi`, `ranga`) VALUES
(1, 'Borys'),
(2, 'Giga Karp'),
(3, 'Szczupak'),
(4, 'Okoń'),
(5, 'Karaś Złocisty'),
(6, 'Krap'),
(7, 'Płotka');
```

Po uzupełnieniu rang klientów, możemy przejść do tabeli producenta. Oto przykładowe dane:

```
INSERT INTO `producent` (`id_producenta`, `nazwa`, `nip`, `adres`, `telefon`, `email`) VALUES
(1, 'Mikado', 11111111, 'Fabryczna 14', 694202137, 'mikado@gmail.com'),
(2, 'Delphin', 22222222, 'Karpiowa 44', 723743123, 'delphin@gmail.com'),
(3, 'Osmo', 33333333, 'Zanętowa 69', 568976534, 'osmo@gmail.com'),
(4, 'Abu Garcia', 44444444, 'Spiningisty 4/5', 432543654, 'abugarcia@gmail.com'),
(5, 'Flagman', 55555555, 'Gruntowa 21', 756438478, 'flagman@gmail.com'),
(6, 'Daiwa', 66666666, 'Karpiarza 11', 789235684, 'daiwa@gmail.com');
```

Następnie wstawiamy kategorie towaru:

```
INSERT INTO `kategoria_towaru` (`id_kategori`, `kategoria`) VALUES
(3, 'Zanęta'),
(4, 'Liquid/Atraktory'),
(5, 'Kulki'),
(6, 'Pellet'),
(7, 'Haczyk'),
(8, 'Przypona'),
(9, 'Linka'),
(10, 'Kołowrotek'),
(11, 'Wędka');
```

Po uzupełnieniu kategori towarów i producentów możemy wstawić towar:

```
INSERT INTO `towar` (`id_towaru`, `id_kategori`, `id_producenta`, `nazwa`, `cena`)
VALUES
(1, 11, 1, 'Noctis Pro', 799),
(2, 10, 5, 'Sherman Pro', 599),
(3, 3, 3, 'Zanęta mega karp', 39),
(4, 9, 5, 'Żyłka Sherman Pro', 28),
(5, 5, 3, 'Kulki Larwa', 19),
(6, 7, 1, 'Haczyk Pro 10', 18),
(7, 4, 3, 'Liquid Truskawka', 23);
```

Teraz dodajemy informacje o kurierach:

```
INSERT INTO `kurier` (`id_kuriera`, `imie`, `nazwisko`, `firma`, `telefon`) VALUES
(1, 'Adolf', 'Eichmann', 'DPD', 452369875),
(2, 'Włodzimierz', 'Biały', 'Pocztex', 462626526),
(3, 'Jeremiasz', 'Różowyczłowiek', 'Inpost', 682168566),
(4, 'Admin', 'Femboy', 'UPS', 639466926),
(5, 'Jeffrey', 'Epstein', 'FedEx', 93593292),
(6, 'Hermann', 'Goring', 'DPD', 526522152),
(7, 'Herr', 'Igor', 'Pocztex', 326269259),
(8, 'Warol', 'Kojtyła', 'Inpost', 816921828),
(9, 'Ewa', 'Braun', 'FedEx', 962919829),
(10, 'Boorys', 'Megakarp', 'UPS', 526592629),
(11, 'Ryszard', 'Nikson', 'Inpost', 908299289),
(12, 'Adam', 'Hilt', 'DPD', 19591981),
(13, 'Charlie', 'Morningstar', 'FedEx', 151986165),
(14, 'Partyk', 'Nietoprzerzłowiek', 'Pocztex', 189962896),
(15, 'Jonasz', 'Koran-Mekka', 'DPD', 18996546);
```

Następnie informacje o sklepach:

```
INSERT INTO `sklep` (`id_sklepu`, `adres`, `nip`, `miasto`, `id_rejonu`, `email`,
`telefon`) VALUES
(1, 'ul. Mickiewicza 1', 2147483647, 'Wrocław', 10, 'Sklepwroclaw@ryby.pl',
889889889),
(2, 'ul. Długa 5', 2147483647, 'Gdańsk', 2, 'SklepGdansk@ryby.pl', 123456789),
(3, 'ul. Jagiellońska 10', 2147483647, 'Olsztyn', 3, 'SklepOlsztyn@ryby.pl',
234567890),
(4, 'ul. Warszawska 20', 2147483647, 'Białystok', 4, 'SklepBialystok@ryby.pl',
345678901),
(5, 'ul. Paderewskiego 15', 2147483647, 'Zielona Góra', 5, 'SklepZielonaGora@ryby.pl',
456789012),
(6, 'ul. Piastowska 30', 2147483647, 'Poznań', 6, 'SklepPoznan@ryby.pl', 567890123),
(7, 'ul. Leśna 25', 2147483647, 'Bydgoszcz', 7, 'SklepBydgoszcz@ryby.pl', 678901234),
```



```
(8, 'ul. Nowy Świat 40', 2147483647, 'Warszawa', 8, 'SklepWarszawa@ryby.pl',
789012345),
(9, 'ul. Słowackiego 35', 123456789, 'Lublin', 9, 'SklepLublin@ryby.pl', 890123456),
(10, 'ul. Piotrkowska 50', 2147483647, 'Łódź', 11, 'SklepLodz@ryby.pl', 901234567),
(11, 'ul. Krakowska 55', 2147483647, 'Kielce', 12, 'SklepKielce@ryby.pl', 12345678),
(12, 'ul. Legionów 60', 2147483647, 'Rzeszów', 13, 'SklepRzeszow@ryby.pl', 123456789),
(13, 'ul. Opolska 65', 2147483647, 'Opole', 14, 'SklepOpole@ryby.pl', 234567890),
(14, 'ul. Katowicka 70', 2147483647, 'Katowice', 15, 'SklepKatowice@ryby.pl',
345678901),
(15, 'ul. Krakowska 75', 1234567890, 'Kraków', 16, 'SklepKrakow@ryby.pl', 456789012),
(16, 'ul. Portowa 80', 2147483647, 'Szczecin', 1, 'SklepSzczecin@ryby.pl', 567890123);
```

Przed ostatnio uzupełniami zamówienia:

```
INSERT INTO `zamowienia` (`id_zamowienia`, `data_zamowienia`, `data_wyslki`,
`id_sklepu`, `id_kuriera`, `id_towaru`) VALUES
(1, '2024-02-01', '2024-02-02', 13, 4, 5),
(2, '2024-02-02', '2024-02-03', 1, 2, 4),
(3, '2024-02-04', '2024-02-05', 4, 15, 4),
(4, '2024-02-05', '2024-02-06', 3, 7, 3),
(5, '2024-01-01', '2024-01-02', 3, 3, 3),
(6, '2024-01-03', '2024-01-06', 7, 9, 5);
```

Na ostatnio dodajemy Klientów:

```
INSERT INTO `klient` (`id_klienta`, `id_rangi`, `id_zamowienia`, `imie`, `nazwisko`,
`adres`, `id_rejonu`, `telefon`, `email`, `rekordowy_gatunekryby`,
`rekordowy_wagaryby`) VALUES
(2, 4, 3, 'Jan', 'Kowalski', '1 Maja', 10, 208210207, 'jestemjan@gmail.com', 'Karp',
'7 Kg'),
(3, 1, 2, 'Oliwia', 'Medyńska', 'Powiat', 12, 555467890,
'oliwia.medynska@powiatwroclawski.pl', 'Płotka', '4 Kg'),
(4, 4, 1, 'Jan', 'Kowalski', '1 Maja', 10, 208210207, 'jestemjan@gmail.com', 'Karp',
'7 Kg'),
(5, 7, 5, 'Borys', 'Łukojć', 'Rybna 2/10', 10, 666999111, 'borys.lukojc@teb.edu.pl',
'Igor', '78 Kg'),
(6, 5, 4, 'Wuzeta', 'Wołczyński', 'Fabryczna', 10, 532879090, 'strzelnica@wroclaw.pl',
'', ''),
(7, 3, 6, 'Mara', 'Ks', 'Capitol', 10, 789123567, 'mara.ks@teb.edu.pl', 'Admin', '70
Kg');
```

W taki sposób baza będzie gotowa do użycia

Przykładowe zapytania

Najlepszym sposobem uczenia się pracy to według nas praktyka, aby było łatwiej zacząć z naszą bazą przygotowaliśmy przykładowe zapytania.

Poniżej są zapisane pytania, oraz odpowiedzi do najpopularniejszych pytań:

1. Kto jest największym klientem pod względem zamówień?

```
SELECT k.imie, k.nazwisko, COUNT(z.id_zamowienia) AS  
liczba_zamowien  
FROM klient k  
JOIN zamowienia z ON k.id_klienta = z.id_klienta  
GROUP BY k.id_klienta  
ORDER BY liczba_zamowien DESC  
LIMIT 1;
```

2. Jakie towary zostały zamówione przez klienta o nazwisku „Kowalski”

```
SELECT t.nazwa  
FROM towar t  
JOIN zamowienia z ON t.id_towaru = z.id_towaru  
JOIN klient k ON z.id_klienta = k.id_klienta  
WHERE k.nazwisko = 'Kowalski';
```

3. Jakie są najczęściej zamawiane produkty?

```
SELECT t.nazwa, COUNT(z.id_zamowienia) AS liczba_zamowien  
FROM towar t  
JOIN zamowienia z ON t.id_towaru = z.id_towaru  
GROUP BY t.id_towaru  
ORDER BY liczba_zamowien DESC;
```

4. Który kurier obsłużył najwięcej zamówień?

```
SELECT k.imie, k.nazwisko, COUNT(z.id_zamowienia) AS  
liczba_zamowien  
FROM kurier k  
JOIN zamowienia z ON k.id_kuriera = z.id_kuriera  
GROUP BY k.id_kuriera  
ORDER BY liczba_zamowien DESC  
LIMIT 1;
```

5. Gdzie znajduje się sklep o adresie „ul. Długa 5”?
SELECT miasto
FROM sklep
WHERE adres = 'ul. Długa 5';
6. Jakie są wszystkie dostępne kategorie towaru?
SELECT DISTINCT kategoria
FROM kategoria_towaru;
7. Kto jest producentem z największą liczbą produktu w ofercie?
SELECT p.nazwa, COUNT(t.id_towaru) AS liczba_produktow
FROM producent p
JOIN towar t ON p.id_producenta = t.id_producenta
GROUP BY p.id_producenta
ORDER BY liczba_produktow DESC
LIMIT 1;
8. Jakie są rekordowe gatunki ryb klientów
SELECT rekordowy_gatunekryby, COUNT(id_klienta) AS liczba_klientow
FROM klient
GROUP BY rekordowy_gatunekryby
ORDER BY liczba_klientow DESC;
9. Który sklep wysłał najwięcej zamówień?
SELECT s.miasto, COUNT(z.id_zamowienia) AS liczba_zamowien
FROM sklep s
JOIN zamowienia z ON s.id_sklepu = z.id_sklepu
GROUP BY s.id_sklepu
ORDER BY liczba_zamowien DESC
LIMIT 1;
10. Który klient mieszka w rejonie „Podlaskie”
SELECT k.imie, k.nazwisko
FROM klient k
JOIN rejon r ON k.id_rejonu = r.id_rejonu
WHERE r.wojewodztwo = 'Podlaskie';