

Baza danych

Dokumentacja projektu

Temat: "Baza danych do obsługi
wypożyczalni samochodów"

Kamil Madej
161876
3EF-ZI

Data wykonania: 12.14.2020

Grupa: L02
Kamil Madej

Spis treści

Spis treści	2
1. Określenie projektu	3
1.1 Określenie tematyki i zakresu projektu,	3
1.2 Określenie funkcji bazy danych i ich priorytetu.....	3
1.3 Wybór technologii oraz typu bazy danych oraz narzędzi do zrealizowania projektu ...	3
1.4 Repozytorium	3
2. Prezentacja Diagramu ERD	4
2.1 Diagram ERD	4
2.2 Opis tabel i ich funkcji	5
3. Tworzenie i wypełnianie tabel	6
3.1 Tworzenie tabeli samochod.	6
3.2 Tworzenie tabeli wypożyczenia.	6
3.3. Dodawanie przykładowych danych do tabeli samochodu.....	6
3.4 Dodawanie danych do tabeli klient.	7
4. Funkcje i procedury	7
4.1 Funkcja „zwróć_adresID”	7
4.2 Procedura „dodaj_klienta”	8
4.3 Procedura „usuń_klienta”	9
4.4 Procedura „wyświetl_wszystkie_samochody”	9

1.1 Określenie tematyki i zakresu projektu, przedstawienie zagadnień związanych z tematem.

Tematem projektu będzie baza danych do obsługi wypożyczalni samochodów. Baza ma za zadanie usprawnić działanie firmy poprzez ułatwieni dostępu do danych związanych z funkcjonowaniem firmy.

1.2 Określenie funkcji bazy danych i ich priorytetu.

- przechowywanie danych: klientów, pracowników, pojazdów, informacje o wypożyczeniach,
- dodawanie, modyfikowanie, usuwanie danych klientów
- dodawanie, modyfikowanie, usuwanie danych o pojazdach
- dodawanie, modyfikowanie, usuwanie danych o pracownikach
- wypożyczanie samochodów,

1.3 Wybór technologii, typu bazy danych oraz narzędzi do zrealizowania projektu.

Projekt będzie realizowany za pomocą relacyjnej bazy danych, przy pomocy SQL Workbench.

1.4 Repozytorium.

<https://github.com/KamilMadej/ProjektBazyDanych>

2.2. Opis tabel.

- **wypożyczenia** – tabela zawierająca podstawowe informacje na temat wypożyczenia.
- **samochód** – tabela zawierająca podstawowe informacje na temat samochodów.
- **osoba, adres** – tabele zawierające podstawowe dane o klientach i pracownikach.
- **pracownik** – zawiera dane o pracownikach rozszerzone o dane z tabel osoba oraz adres.
- **klient** – zawiera dane o klientach rozszerzone o dane z tabel osoba oraz adres.
- **marka** – zawiera informacje o nazwie marki samochodowej.
- **model** – zawiera informacje o modelu samochodu.
- **dostępność** – zawiera informacje o dostępności samochodu.
- **segmentauta** - zawiera informacje o segmencie auta, na podstawie tej tabeli ustalany jest cennik.
- **cennik** – zawiera informacje o cenie wypożyczenia, ustalonej na podstawie segmentu auta.
- **status_wypożyczenia** – zawiera informacje o statusie wypożyczenia.

3. Tworzenie i wypełnianie tabel.

3.1 Tworzenie tabeli samochod.

```
CREATE TABLE `samochód` (
  `nr_rejestracyjny` varchar(10) NOT NULL,
  `przebieg` varchar(15) NOT NULL,
  `pojemność_silnika` varchar(6) NOT NULL,
  `moc_silnika` varchar(4) NOT NULL,
  `rok_produkcji` varchar(4) NOT NULL,
  `typ_paliwa` varchar(10) NOT NULL,
  `kolor` varchar(20) NOT NULL,
  `model_id` int NOT NULL,
  `segmentauta_id` int NOT NULL,
  `dostępność` int NOT NULL,
  PRIMARY KEY (`nr_rejestracyjny`),
  UNIQUE KEY `id_UNIQUE` (`nr_rejestracyjny`),
  KEY `fk_dostępność_idx` (`dostępność`),
  KEY `fk_Samochód_Model1_idx` (`model_id`),
  KEY `fk_samochód_segmentauta_id` (`segmentauta_id`),
  CONSTRAINT `fk_dostępność` FOREIGN KEY (`dostępność`) REFERENCES `dostępność` (`id`),
  CONSTRAINT `fk_samochód_model` FOREIGN KEY (`model_id`) REFERENCES `model` (`id`),
  CONSTRAINT `fk_samochód_segmentauta` FOREIGN KEY (`segmentauta_id`) REFERENCES `segmentauta` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

3.2 Tworzenie tabeli wypożyczenia.

```
CREATE TABLE `wypożyczenia` (
  `id` int NOT NULL AUTO_INCREMENT,
  `klient_id` int NOT NULL,
  `samochód_nr_rejestracyjny` varchar(10) NOT NULL,
  `pracownik_id` int NOT NULL,
  `data_wypożyczenia` date NOT NULL,
  `data_zwrotu` date DEFAULT NULL,
  `status_wypożyczenia_id` int NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `id_UNIQUE` (`id`),
  KEY `fk_wypożyczenie_samochód1_idx` (`samochód_nr_rejestracyjny`),
  KEY `fk_wypożyczenie_status_wypożyczenia1_idx` (`status_wypożyczenia_id`),
  KEY `fk_wypożyczenia_klient_idx` (`klient_id`),
  KEY `fk_wypożyczenia_pracownik_idx` (`pracownik_id`),
  CONSTRAINT `fk_wypożyczenia_klient` FOREIGN KEY (`klient_id`) REFERENCES `klient` (`id`),
  CONSTRAINT `fk_wypożyczenia_pracownik` FOREIGN KEY (`pracownik_id`) REFERENCES `pracownik` (`id`),
  CONSTRAINT `fk_wypożyczenie_samochód1` FOREIGN KEY (`samochód_nr_rejestracyjny`) REFERENCES `samochód` (`nr_rejestracyjny`),
  CONSTRAINT `fk_wypożyczenie_status_wypożyczenia1` FOREIGN KEY (`status_wypożyczenia_id`) REFERENCES `status_wypożyczenia` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

3.3. Dodawanie przykładowych danych do tabeli samochodu.

```
call projekt_wypożyczalnia_samochodów.dodaj_samochód('RNI KK55', '50000', '3000', '220', '2009', 'Benzyna', 'Fioletowy', 'BMW', 'e60', 'I');
call projekt_wypożyczalnia_samochodów.dodaj_samochód('RIO SD55', '54660', '1550', '120', '2029', 'Diesel', 'Srebrny', 'Audi', 'A4B8', 'G');
call projekt_wypożyczalnia_samochodów.dodaj_samochód('RLE LCD4', '100000', '1900', '110', '2012', 'Benzyna', 'Czarny', 'Subaru', 'Impreza', 'M');
call projekt_wypożyczalnia_samochodów.dodaj_samochód('ASDA RTG4', '200000', '40000', '420', '2017', 'Benzyna', 'Czarny', 'BMW', 'F20', 'C');
call projekt_wypożyczalnia_samochodów.dodaj_samochód('UFO TR86', '80000', '1400', '86', '2017', 'Diesel', 'Srebrny', 'Seat', 'Cordoba', 'F');
call projekt_wypożyczalnia_samochodów.dodaj_samochód('RKL 9KGF', '147200', '2100', '143', '2006', 'Benzyna', 'Zielony', 'Dodge', 'Charger', 'E');
call projekt_wypożyczalnia_samochodów.dodaj_samochód('RLE 10j7', '125000', '5000', '4000', '2012', 'Benzyna', 'Biały', 'Dodge', 'Viper', 'E');
call projekt_wypożyczalnia_samochodów.dodaj_samochód('RLU 4ULO', '20000', '1300', '260', '2017', 'Benzyna', 'Czarny', 'Mazda', 'RX8', 'K');
call projekt_wypożyczalnia_samochodów.dodaj_samochód('KGU 0896', '543535', '3200', '310', '2010', 'Diesel', 'Czerwony', 'Opel', 'Zafira', 'A');
call projekt_wypożyczalnia_samochodów.dodaj_samochód('MFG DVD1', '250000', '1800', '133', '2000', 'Benzyna', 'Granatowy', 'Ford', 'Mondeo', 'M');
```

3.4 Dodawanie danych do tabeli klient.

```
call projekt_wypożyczalnia_samochodów.add_client('Adam', 'Zarzycki', '1995-07-29', '95072998675', '876940276', 'AdamZarzycki@gmail.com', 'Jeżowe', '37-430', 'Lipowa', 'Posiada');
call projekt_wypożyczalnia_samochodów.add_client('Konrad', 'Jankowski', '1967-09-13', '67091368436', '345671235', 'KonradJankowski@gmail.com', 'Abisynia', '34-546', 'Mazowiecka', 'Posiada');
call projekt_wypożyczalnia_samochodów.add_client('Krystian', 'Chmielewski', '1975-07-06', '75070609275', '866343414', 'KrystianChmielewski@gmail.com', 'Adamczowice', '37-432', 'Tłomackie', 'Posiada');
call projekt_wypożyczalnia_samochodów.add_client('Hubert', 'Makowski', '1987-12-29', '871229682365', '987875214', 'HubertMakowski@gmail.com', 'Adamowo', '55-532', 'Teatralny', 'Posiada');
call projekt_wypożyczalnia_samochodów.add_client('Konstanty', 'Tomaszewski', '1989-01-06', '890106234611', '123567478', 'KonstanTomaszewski@gmail.com', 'Kraśnik', '43-123', 'Defilad', 'Posiada');
call projekt_wypożyczalnia_samochodów.add_client('Radosław', 'Krupa', '1967-06-29', '67062967361', '864577890', 'RadosławKrupa@gmail.com', 'Warszawa', '34-942', 'Próżna', 'Posiada');
call projekt_wypożyczalnia_samochodów.add_client('Kewin', 'Lis', '1971-08-14', '71081498410', '875020586', 'KewinLis@gmail.com', 'Kraków', '22-205', 'Zielona', 'Posiada');
call projekt_wypożyczalnia_samochodów.add_client('Artur', 'Kaczmarczyk', '1955-08-29', '55082972345', '555666777', 'ArturKaczmarczyk@gmail.com', 'Szczecin', '00-032', 'Marianka', 'Posiada');
call projekt_wypożyczalnia_samochodów.add_client('Dobromiła', 'Sadowska', '1958-11-17', '58111765738', '988355067', 'DobromiłaSadowska@gmail.com', 'Wrocław', '12-234', 'Dzika', 'Posiada');
call projekt_wypożyczalnia_samochodów.add_client('Amadeusz', 'Pawlak', '1997-07-02', '97070223856', '609300165', 'AmadeuszPawlak@gmail.com', 'Stalowa Wola', '11-111', 'Błońska', 'Posiada');
call projekt_wypożyczalnia_samochodów.add_client('Denis', 'Gajewski', '2000-01-27', '00012775923', '568505436', 'DenisGajewski@gmail.com', 'Nisko', '19-197', 'Rybaki', 'Posiada');
```

4. Funkcje i procedury

4.1 Funkcja „zwróć_adresID”

Nazwa: zwróć_adresID

Parametry: Wszystkie atrybuty z tabeli adres

Wynik: Funkcja zwraca id na podstawie przekazanych parametrów.

Zasada działania:

1. Deklaracja zmiennej idAdres.
2. Inicjalizacja zmiennej idAdres przy pomocy zapytania select, które zwraca konkretne id z tabeli adres porównując przekazane do parametry z atrybutami tabeli.

Ciało funkcji:

```
CREATE DEFINER='root'@'localhost' FUNCTION `zwróć_adresID`(  
  miejscowość varchar(25), kod_pocztowy varchar(10), ulica varchar(25)  
) RETURNS int  
BEGIN  
  DECLARE idAdres INT DEFAULT 0;  
  
  set idAdres = (select id from adres where adres.miejscowość = miejscowość AND adres.kod_pocztowy = kod_pocztowy AND adres.ulica = ulica LIMIT 1);  
  
  RETURN idAdres;  
END
```

4.2 Procedura „dodaj_klienta”

Nazwa: dodaj_klienta

Parametry: Wszystkie atrybuty z tabel: osoba, adres, klient

Wynik: Procedura przy pomocy przekazanych parametrów inicjalizuje tabele klient, bądź zwraca komunikat jeżeli dane istnieją już w bazie.

Zasada działania:

1. Deklaracja zmiennych idAdres, idOsoba.
2. Sprawdzenie, czy parametr unikalny dla każdego klienta tzn. pesel, nr_telefonu, email widnieje już w bazie. Jeżeli tak to procedura wyświetla komunikat „Osoba o podanych danych widnieje w bazie”. Jeżeli nie :
3. do tabel osoba i adres zostają wpisane odpowiednie dane.
4. Inicjalizacja zmiennych idAdres i idOsoba za pomocą funkcji zwróć_adresID, zwróć_id_osoba.
5. Wstawienie do tabeli klient odpowiednich danych.

Ciało procedury:

```
CREATE DEFINER='root'@'localhost' PROCEDURE `dodaj_klienta`(  
  imie varchar(20), nazwisko varchar(20), data_urodzenia DATE, pesel varchar(11), nr_telefonu varchar(10), email varchar(45),  
  miejscowość varchar(25), kod_pocztowy varchar(10), ulica varchar(25),  
  prawo_jazdy varchar(15)  
)  
BEGIN  
  DECLARE idAdres INT DEFAULT 0;  
  DECLARE idOsoba INT DEFAULT 0;  
  
  IF NOT EXISTS (SELECT pesel,nr_telefonu,email from osoba WHERE osoba.pesel = pesel OR osoba.nr_telefonu = nr_telefonu OR osoba.email = email) THEN  
  
    INSERT INTO osoba(imie, nazwisko, data_urodzenia, pesel, nr_telefonu, email) VALUES(imie, nazwisko, data_urodzenia, pesel, nr_telefonu, email);  
    INSERT INTO adres(miejscowość, kod_pocztowy, ulica) VALUES(miejscowość, kod_pocztowy, ulica);  
  
    set idAdres = (select projekt_wypożyczalnia_samochodów.zwróć_adresID(miejscowość, kod_pocztowy, ulica) LIMIT 1);  
    set idOsoba = (select projekt_wypożyczalnia_samochodów.zwróć_id_osoba(imie, nazwisko, data_urodzenia, pesel, nr_telefonu, email) LIMIT 1);  
  
    INSERT INTO klient(prawo_jazdy, adres_id, osoba_id) VALUES(prawo_jazdy,idAdres, idOsoba);  
  
  ELSE  
    SELECT 'Osoba o podanych danych widnieje w bazie';  
  END IF;  
END
```


4.3 Procedura „usuń_klienta”

Nazwa: usuń_klienta

Parametry: Id klienta

Wynik: Usuwa klienta i powiązane z nim tabele z bazy.

Zasada działania:

1. Usunięcie tabeli klient i powiązanych tabel osoba oraz adres, jeżeli podany parametr jest taki sam jak wartość w tabeli klient.

Ciało procedury:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `usuń_klienta`(id INT)
BEGIN
    DELETE adres, osoba, klient
    FROM klient
    INNER JOIN adres on klient.adres_id = adres.id
    INNER JOIN osoba on klient.osoba_id = osoba.id
    WHERE klient.id = id;
END
```

4.4 Procedura „wyświetl_wszystkie_samochody”

Nazwa: wyświetl_wszystkie_samochody

Parametry:

Wynik: Wyświetla wszystkie samochody znajdujące się w bazie. Od najnowszych do najstarszych.

Zasada działania:

Procedura wyświetla zawartość tabeli samochod i powiązane z nią tabele segment_auta, model, marka, dostępność i grupuje je po atrybucie rok_produkcji w porządku malejącym.

Ciało procedury:

```
CREATE DEFINER='root'@'localhost' PROCEDURE `wyświetl_wszystkie_samochody`()
BEGIN

    SELECT marka.nazwa as "Marka", model.nazwa as 'Model', rok_produkcji as 'Rok produkcji',
    przebieg as 'Przebieg', moc_silnika as 'Moc', pojemność_silnika as 'Pojemność',
    nr_rejestracyjny as 'Nr rejestracyjny', typ_paliwa as 'Typ paliwa', kolor as 'Kolor',
    segment_auta.nazwa as 'Segment', dostępność.Opis
    FROM samochód
    INNER JOIN model
    ON samochód.model_id = model.id
    INNER JOIN marka
    on model.marka_id = marka.id
    INNER JOIN dostępność
    ON samochód.dostępność = dostępność.id
    INNER JOIN segment_auta
    ON samochód.segment_auta_id = segment_auta.id

    ORDER BY rok_produkcji DESC;
END
```

5. Wnioski

Celem projektu było zrealizowanie bazy danych do obsługi wypożyczalni samochodów.

Początkowo projekt zakładał, że baza będzie miała następujące funkcjonalności: dodawanie pracowników, klientów, samochodów, realizacje wypożyczeń, płatności, pobieranie kaucji oraz nie zwracanie jej w przypadku uszkodzenia pojazdu oraz bazowanie ceny wypożyczenia na tym jak długo klient posiada prawo jazdy oraz w jakim jest wieku. Nie udało się jednak zrealizować całej funkcjonalności wokół kaucji i ceny wypożyczenia.