Maciej Domagalski

Bazy Danych 1 – projekt
Dokumentacja

link do projektu:
pascal.fis.agh.edu.pl/~6domagalski/BD/index.php

I. Projekt koncepcji, założenia

Projekt obsługuje „restaurację na dowóz” - jej pracowników oraz klientów.
Klient restauracji może składać zamówienia na wybrane produkty (jedno lub więcej z każdej kategorii: zupy, dania główne, 1 lub 2 dodatki oraz napój). Po złożeniu zamówienia może kontrolować jego przebieg. Jeśli nic obecnie nie zamawia, może przejrzeć swoją historię zamówień. Może również zmienić swoje dane lub usunąć konto.

Kucharz może wyświetlać zamówienia, które wymagają przygotowania oraz ich elementy. Może każde z nich przygotować.

Kurier rozpoczyna dzień pracy od wyboru pojazdu. Kurier odpowiada za odbieranie z restauracji gotowych zamówień (które widzi na swoim panelu), oraz dostarczanie ich klientom.

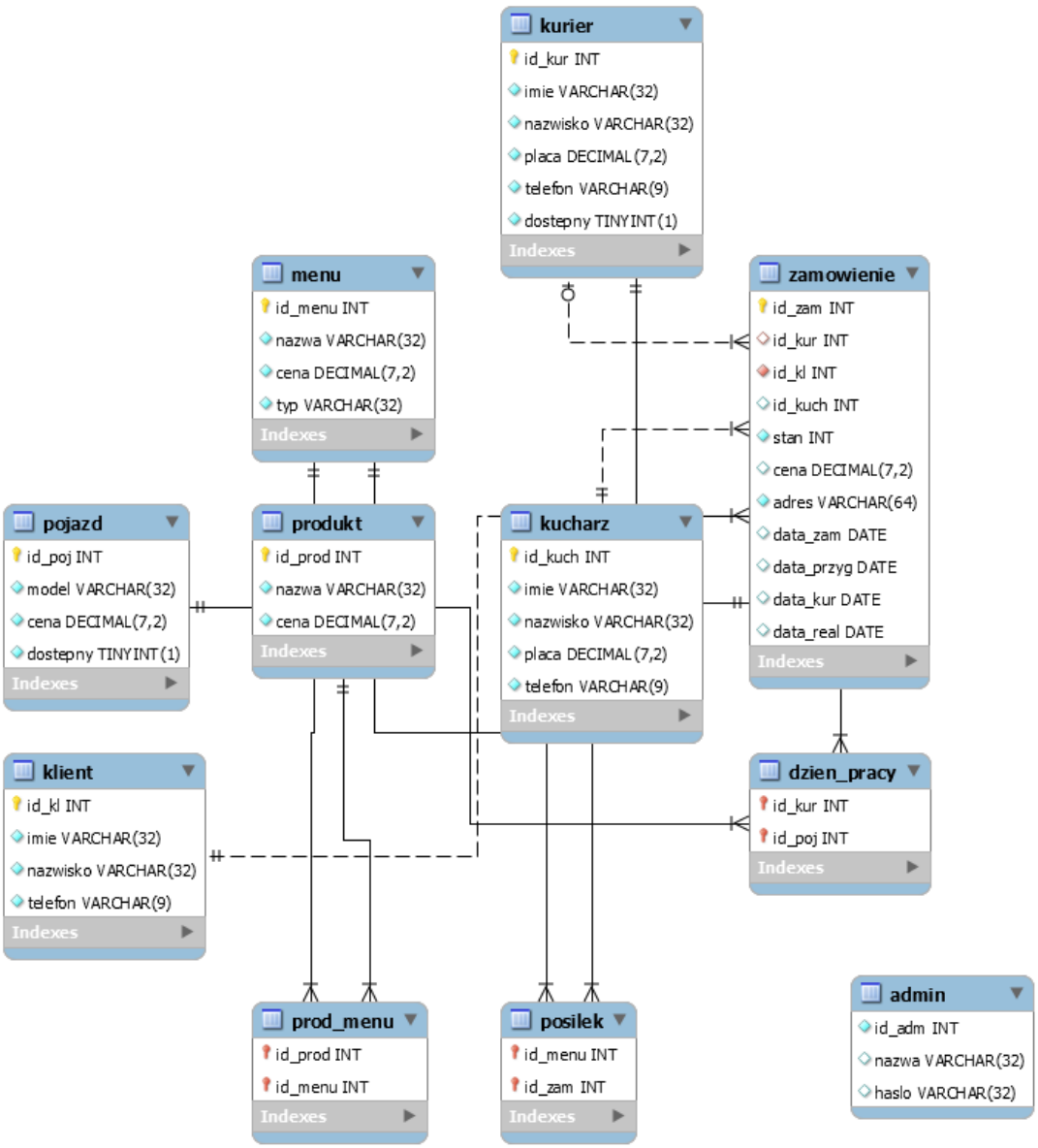
Administrator może wprowadzać nowe rekordy, wprowadzać zmiany w istniejących i wyświetlać pełny raport z historii zamówień.

II. Projekt diagramów (konceptualny)

Przedstawienie encji:

Administrator – przechowuje dane administratora.
Klient – przechowuje dane klienta.
Kucharz – przechowuje dane kucharza.
Kurier – przechowuje dane kuriera. Pole **dostepny** określa, czy w danej chwili kurier może odebrać zamówienie.
Pojazd – reprezentuje pojazd. Pole **dostepny** określa, czy w danej chwili pojazd jest w użyciu.
Dzień pracy – encja asocjacyjna łącząca kuriera z pojazdem.
Produkt – reprezentuje półprodukt, z którego zostanie przygotowane danie.
Menu – reprezentuje element menu. Pole **typ** określa, do której kategorii menu będzie należał.
Prod_menu – encja asocjacyjna, służąca do łączenia produktów z elementem menu.
Zamówienie – reprezentuje zamówienie. Poza polami oczywistymi przechowuje stan zamówienia (0 – przyjęte, 1 – przygotowane, 2 – wysłane, 3 – dostarczone). Przechowuje również wszystkie daty, gdy stan został zmieniony.
Posilek – encja asocjacyjna, która pozwala na przyporządkowanie elementów menu do zamówienia.

Diagram ERD (wykonany za pomocą programu MySQL Workbench):



III. Projekt logiczny.

Wszystkie tabele i widoki znajdują się w schemacie **projekt.**

Klucze tabel:

Admin – brak.
Klient – główny: **id_kl.**
Kucharz – główny: **id_kuch.**
Kurier – główny: **id_kur.**
Pojazd – główny: **id_poj.**
Dzień pracy - obce: **id_kur, id_poj.** Razem stanowią klucz główny.
Produkt – główny: **id_prod.**
Menu – główny: **id_menu.**
Prod_menu – obce: **id_prod, id_menu.** Razem stanowią klucz główny.
Zamówienie – główny: **id_zam**. Obce: **id_kl, id_kuch, id_kur**.
Posiłek – obce: **id_zam, id_menu.** Razem stanowią klucz główny.

Poza tabelami w bazie danych występują dwa tymczasowe widoki:
lista_kl – raport z historią zamówień klienta,
lista_ad – raport z historią wszystkich zamówień.

Spis wykorzystanych kwerend:
SET search_path TO projekt; – stosowana w każdym pliku, aby ustawić schemat.
Logowanie się:
Na konto admina:
SELECT COUNT(*) FROM admin WHERE nazwa = '$login' AND haslo = '$pass'; – jeśli wynikiem jest 1, to logowanie na konto administratora zakończyło się pomyślnie. Następnie program wybiera jego ID, aby przejść do ekranu głównego: *SELECT id_adm FROM admin WHERE nazwa = '$login' AND haslo = '$pass';*.
Na konto klienta (analogicznie):
SELECT COUNT(*) FROM klient WHERE telefon = '$telefon';
SELECT id_kl FROM klient WHERE telefon = '$telefon';
Na konto kuriera (analogicznie):
SELECT COUNT(*) FROM kurier WHERE telefon = '$telefon';
SELECT id_kur FROM kurier WHERE telefon = '$telefon';
Na konto kucharza (analogicznie):
SELECT COUNT(*) FROM kucharz WHERE telefon = '$telefon';
SELECT id_kuch FROM kucharz WHERE telefon = '$telefon';

Dodawanie danych:
INSERT INTO klient(imie, nazwisko, telefon) VALUES('$imie', '$nazwisko', '$telefon'); – dodaje nowego klienta.
INSERT INTO kucharz(imie, nazwisko, placa, telefon) VALUES('$imie', '$nazwisko', '$placa', '$telefon'); – dodaje nowego kucharza.
INSERT INTO kurier(placa, dostepny, nazwisko, imie, telefon) VALUES('$placa', 'false', '$nazwisko', '$imie', '$telefon'); – dodaje nowego kuriera (domyślnie nie może odbierać zamówień, bo nie zaczął dnia pracy).
INSERT INTO pojazd(model, cena, dostepny) VALUES('$model', '$cena', 'true'); – dodaje nowy pojazd. Domyślnie jest dostępny, bo żaden kurier go nie wybrał.
INSERT INTO produkt(cena, nazwa) VALUES('$cena', '$nazwa'); – dodaje nowy produkt.
Aby dodać nowy element menu, wykonywane są następujące kwerendy:
INSERT INTO menu(nazwa, cena, typ) VALUES('$nazwa', '$cena', '$typ'); – dodaje nowy element menu, na razie nie składający się z niczego.
INSERT INTO prod_menu (id_prod, id_menu) VALUES ($obecny_skladnik, $id_menu); – wykonywana dla każdego produktu, aby przyporządkować go do nowego elementu menu.

Ekran administratora:
SELECT nazwa FROM admin WHERE id_adm = '$id' – wyświetla jego dane.
W bazie występuje więcej SELECTów do wyświetlania list poszczególnych encji, ale w większości nie potrzebują omawiania. Wyjątki przedstawiam niżej:
CREATE VIEW lista_ad AS SELECT z.id_zam, z.adres, z.cena, z.stan, z.data_real, kl.imie ||' '|| kl.nazwisko AS klient, kur.imie ||' '|| kur.nazwisko AS kurier, kuch.imie ||' '|| kuch.nazwisko AS kucharz FROM zamowienie z LEFT JOIN klient kl ON kl.id_kl = z.id_kl LEFT JOIN kurier kur ON z.id_kur = kur.id_kur LEFT JOIN kucharz kuch ON z.id_kuch = kuch.id_kuch GROUP BY z.stan, z.id_zam, kur.imie, kur.nazwisko, kuch.imie, kuch.nazwisko, kl.imie, kl.nazwisko ORDER BY stan – sporządza raport z historii zamówień. Wybiera jego id, adres, cenę, datę zrealizowania oraz imiona i nazwiska klienta, kucharza i kuriera którzy są związani z zamówieniem.
SELECT z.id_kl, z.adres, z.cena, z.data_zam, z.data_przyg, z.data_kur, z.data_real, kl.imie ||' '|| kl.nazwisko AS klient, kur.imie ||' '|| kur.nazwisko AS kurier, kuch.imie ||' '|| kuch.nazwisko AS kucharz FROM zamowienie z LEFT JOIN klient kl ON kl.id_kl = z.id_kl LEFT JOIN kurier kur ON z.id_kur = kur.id_kur LEFT JOIN kucharz kuch ON z.id_kuch = kuch.id_kuch WHERE z.id_zam = $idZ – wyświetla WSZYSTKIE szczegóły danego zamówienia poza elementami menu (za to odpowiada kolejna kwerenda):

SELECT nazwa FROM menu INNER JOIN posilek ON menu.id_menu = posilek.id_menu WHERE posilek.id_zam = $idZ.
Ekran klienta:
SELECT imie FROM klient WHERE id_kl = '$id';, SELECT nazwisko FROM klient WHERE id_kl = '$id'; – wyświetla jego dane.
SELECT COUNT(*) FROM zamowienie WHERE id_kl='$id' AND stan <> 3 – sprawdza, czy klient coś właśnie zamawia. Jeśli zwróci 0, to klient obecnie nie czeka na zamówienie.
SELECT stan FROM zamowienie WHERE id_kl='$id' ORDER BY stan LIMIT 1 – jeśli klient coś właśnie zamawia, to ta kwerenda wybiera to zamówienie.
SELECT imie, nazwisko, telefon FROM kucharz INNER JOIN zamowienie on zamowienie.id_kuch = kucharz.id_kuch WHERE id_kl='$id' AND zamowienie.stan = 1 – jeśli klient czeka na zamówienie i zostało ono przygotowane, to ta kwerenda wyświetla dane kucharza, które za nie odpowiada.
SELECT imie, nazwisko, telefon FROM kucharz INNER JOIN zamowienie on zamowienie.id_kuch = kucharz.id_kuch WHERE id_kl='$id' AND zamowienie.stan = 2;
SELECT imie, nazwisko, telefon FROM kurier INNER JOIN zamowienie on zamowienie.id_kur = kurier.id_kur WHERE id_kl='$id' AND zamowienie.stan = 2; - jeśli klient czeka na zamówienie i zostało ono do niego wysłane, to ta kwerenda wyświetla dane kucharza i kuriera, którzy za nie odpowiadają.
CREATE VIEW lista_kl AS SELECT z.id_zam, z.adres, z.cena, z.data_real, kur.imie ||' '|| kur.nazwisko AS kurier, kuch.imie ||' '|| kuch.nazwisko AS kucharz FROM zamowienie z INNER JOIN kurier kur ON z.id_kur = kur.id_kur INNER JOIN kucharz kuch ON z.id_kuch = kuch.id_kuch WHERE z.id_kl = $id - sporządza raport z historii zamówień danego klienta. Wybiera jego id, adres, cenę, datę zrealizowania oraz imiona i nazwiska kucharza i kuriera którzy są związani z zamówieniem.
SELECT z.id_kl, z.adres, z.cena, z.data_zam, z.data_przyg, z.data_kur, z.data_real, kur.imie ||' '|| kur.nazwisko AS kurier, kuch.imie ||' '|| kuch.nazwisko AS kucharz FROM zamowienie z INNER JOIN kurier kur ON z.id_kur = kur.id_kur INNER JOIN kucharz kuch ON z.id_kuch = kuch.id_kuch WHERE z.id_zam = $idZ - wyświetla WSZYSTKIE szczegóły danego zamówienia poza elementami menu (za to odpowiada kolejna kwerenda):
SELECT nazwa FROM menu INNER JOIN posilek ON menu.id_menu = posilek.id_menu WHERE posilek.id_zam = $idZ.

Ekran kucharza:
SELECT imie FROM kucharz WHERE id_kuch = '$id';, SELECT nazwisko FROM kucharz WHERE id_kuch = '$id'; – wyświetla jego dane.
SELECT z.id_zam, k.imie, k.nazwisko, k.telefon FROM zamowienie z INNER JOIN klient k ON k.id_kl = z.id_kl WHERE stan = 0 – wybiera zamówienia, które czekają na przygotowanie bez elementów menu, za to odpowiada kolejna kwerenda:
SELECT nazwa FROM menu INNER JOIN posilek ON menu.id_menu = posilek.id_menu WHERE posilek.id_zam = $zam.
UPDATE zamowienie SET id_kuch = $id, stan = 1, data_przyg = now() WHERE id_zam = $zam – zmienia stan zamówienia, oznaczając, że jest gotowe do wysyłki.

Ekran kuriera:
SELECT imie FROM kurier WHERE id_kur = '$id';, SELECT nazwisko FROM kurier WHERE id_kur = '$id'; - wyświetla jego dane.
SELECT id_poj, model FROM pojazd WHERE dostepny='true' ORDER BY id_poj; - wyświetla dostępne w tej chwili pojazdy.
Aby umożliwić kurierowi wybranie pojazdu na początku dnia pracy, wykonywane są następujące kwerendy:
INSERT INTO dzien_pracy (id_kur, id_poj) VALUES($id, $wybor); - dobiera pojazd do kuriera
UPDATE kurier SET dostepny='true' WHERE id_kur = $id; - zmienia stan dostępności kuriera.
UPDATE pojazd SET dostepny='false' WHERE id_poj = $wybor; - zmienia stan dostępności pojazdu.
SELECT model FROM pojazd WHERE id_poj = '$pojazdID' – wyświetla pojazd, w którym znajduje się kurier.
SELECT z.id_zam, k.imie, k.nazwisko, k.telefon FROM zamowienie z INNER JOIN klient k ON k.id_kl = z.id_kl WHERE stan = 1; - wybiera zamówienia, które czekają na odbiór z restauracji.
SELECT k.imie, k.nazwisko, k.telefon, k.adres FROM klient k INNER JOIN zamowienie z ON z.id_kl = k.id_kl WHERE id_zam = $zam – wyświetla dane klienta, do którego należy dostarczyć zamówienie.
UPDATE zamowienie SET id_kur = $id, stan = 2, data_kur = now() WHERE id_zam = $zam - zmienia stan zamówienia, oznaczając, że zostało ono wysłane.
UPDATE zamowienie SET id_kur = $id, stan = 3, data_real = now() WHERE id_zam = $zam - zmienia stan zamówienia, oznaczając, że zostało ono dostarczone.
Gdy kurier kończy pracę, wykonywane są następujące kwerendy:
UPDATE kurier SET dostepny='false' WHERE id_kur = $id;
UPDATE pojazd SET dostepny='true' WHERE id_poj = $pojazdID;.

Aktualizowanie danych:
UPDATE klient SET imie = '$imie', nazwisko = '$nazwisko', telefon = '$telefon' WHERE id_kl = $id – pozwala klientowi zmienić jego dane.
DELETE FROM klient WHERE id_kl = $id; - pozwala klientowi usunąć konto.
UPDATE kurier SET imie = '$imie', nazwisko = '$nazwisko', placa = '$placa', telefon = '$telefon' WHERE id_kur = $id – pozwala administratorowi zmienić dane kuriera.
DELETE FROM kurier WHERE id_kur = $id; - pozwala administratorowi usunąć kuriera.
UPDATE kucharz SET imie = '$imie', nazwisko = '$nazwisko', placa = '$placa', telefon = '$telefon' WHERE id_kuch = $id; - pozwala administratorowi zmienić dane kucharza.
DELETE FROM kucharz WHERE id_kuch = $id – pozwala administratorowi usunąć kucharza.
UPDATE menu SET nazwa = '$nazwa', typ = '$typ' WHERE id_menu = $id; – pozwala administratorowi zmienić nazwę i typ elementu menu.
W celu zmiany produktów, jakie są potrzebne do elementu menu, realizowane są poniższe kwerendy:
UPDATE menu SET cena = $cena WHERE id_menu = $id; - ustalenie nowej ceny.
DELETE FROM prod_menu WHERE id_menu = $id; - resetuje listę produktów.
INSERT INTO prod_menu (id_prod, id_menu) VALUES ($obecny_skladnik, $id); - na nowo ustala listę składników.
Aby skasować element menu, wykonywane są poniższe kwerendy:
DELETE FROM posilek WHERE id_menu = $id;
DELETE FROM prod_menu WHERE id_menu = $id;
DELETE FROM menu WHERE id_menu = $id;.

Składanie zamówienia:
SELECT id_menu, nazwa,cena FROM menu WHERE typ = 'glowne';
SELECT id_menu, nazwa,cena FROM menu WHERE typ = 'zupa';
SELECT id_menu, nazwa,cena FROM menu WHERE typ = 'dodatek';
SELECT id_menu, nazwa,cena FROM menu WHERE typ = 'napoj';
SELECT id_menu, nazwa,cena FROM menu WHERE typ = 'glowne'; - wybierają wszystkie elementy z danej kategorii, aby klient mógł któreś wybrać.
SELECT COALESCE(MAX(id_zam),0) FROM zamowienie – ustala id zamówienia, które zostanie złożone.
Kiedy klient wybierze elementy: *INSERT INTO zamowienie(id_kl, adres, stan) VALUES('$idK', '$adres', 0);* tworzy nowe zamówienie. Następnie (w razie potrzeby) zostanie ono uzupełniane elementami menu:
INSERT INTO posilek(id_zam, id_menu) VALUES ($zam, $glowne);
INSERT INTO posilek(id_zam, id_menu) VALUES ($zam, $zupa);
INSERT INTO posilek(id_zam, id_menu) VALUES ($zam, $dodatek1);
INSERT INTO posilek(id_zam, id_menu) VALUES ($zam, $dodatek2);
INSERT INTO posilek(id_zam, id_menu) VALUES ($zam, $napoj);.
Następnie klientowi zostaje zaprezentowana cena, którą musi zapłacić za zamówienie. Jeśli zdecyduje się je potwierdzić, to wykonywana jest kwerenda *UPDATE zamowienie SET cena = $cena, data_zam = now() WHERE id_zam = $zam;*.
Jeśli klient się rozmyśli, to przywracane zostaje stan poprzedni:
DELETE FROM posilek WHERE id_zam = $zam;
DELETE FROM zamowienie WHERE id_zam = $zam;
ALTER SEQUENCE zamowienie_id_zam_seq RESTART WITH $zam;.

IV. Projekt funkcjonalny.

Wszystkie formularze są intuicyjne i nie wymagają objaśnień, z wyjątkiem dodawania elementu menu: wymaga on zdefiniowania liczby składników, zanim administrator je będzie mógł wybierać. Wszystkie raporty mają formę tabel o jasnej strukturze, gdzie w roli objaśnień występują nagłówki. Sterowanie aplikacją jest objaśniane na przyciskach. Projekt nie zawiera makropoleceń.

V. Dokumentacja.

Wszystkie dane wprowadzane są ręcznie. Aplikacja jest na tyle intuicyjna, że nie potrzebuje instrukcji. Wykorzystane technologie: baza danych PostgreSQL połączona poprzez PHP do aplikacji webowej, wykorzystującej HTML5, CSS (z frameworkiem Bootstrap) oraz JavaScript (miejscami z jQuery).

Spis literatury:
- wykłady i ćwiczenia z przedmiotu Bazy Danych
- https://www.php.net/docs.php
- How to Generate Database EER Diagrams from SQL Scripts using MySQL Workbench · Güngör Budak (tutorial z którego korzystałem aby wygenerować ERD)