

Projekt bazy danych dla Restauracji

Szymon Gołębiowski

Michał Gniadek

Dominika Bocheńczyk

Funkcje

Obsługa

- Podgląd aktualnych zamówień
- Akceptacja oczekujących zamówień
- W przypadku kiedy klient zamawia na miejscu, możliwość wprowadzenia do systemu zamówienia (również na wynos) i zajęcia stolika
- Zmiana informacji dotyczących zamówienia i rezerwacji (np. jeśli klient wyjdzie wcześniej)
- Wygenerowanie faktury za zamówienie

Klienci

- Złożenie zamówienia na wynos (przez internet; z limitem czasu do kiedy ustalone jest menu)
- Rezerwacja stolika (przez internet) + ew. złożenie zamówienia (z limitem czasu do kiedy ustalone jest menu) - system weryfikuje czy w podanym czasie i dla podanej liczby osób jest miejsce, a w przypadku zamówienia czy dane danie jest dostępne (a dla owoców morza czy zamówienie jest składane z odpowiednim wyprzedzeniem)
- Możliwość anulowania rezerwacji i zamówienia
- Sprawdzenie statusu rezerwacji
- Wygenerowanie faktury za pojedyncze zamówienie i zbiorczej za cały miesiąc
- Przeglądanie historii zamówień i dostępnych rabatów i wygenerowanie raportów z historią

Kierownictwo

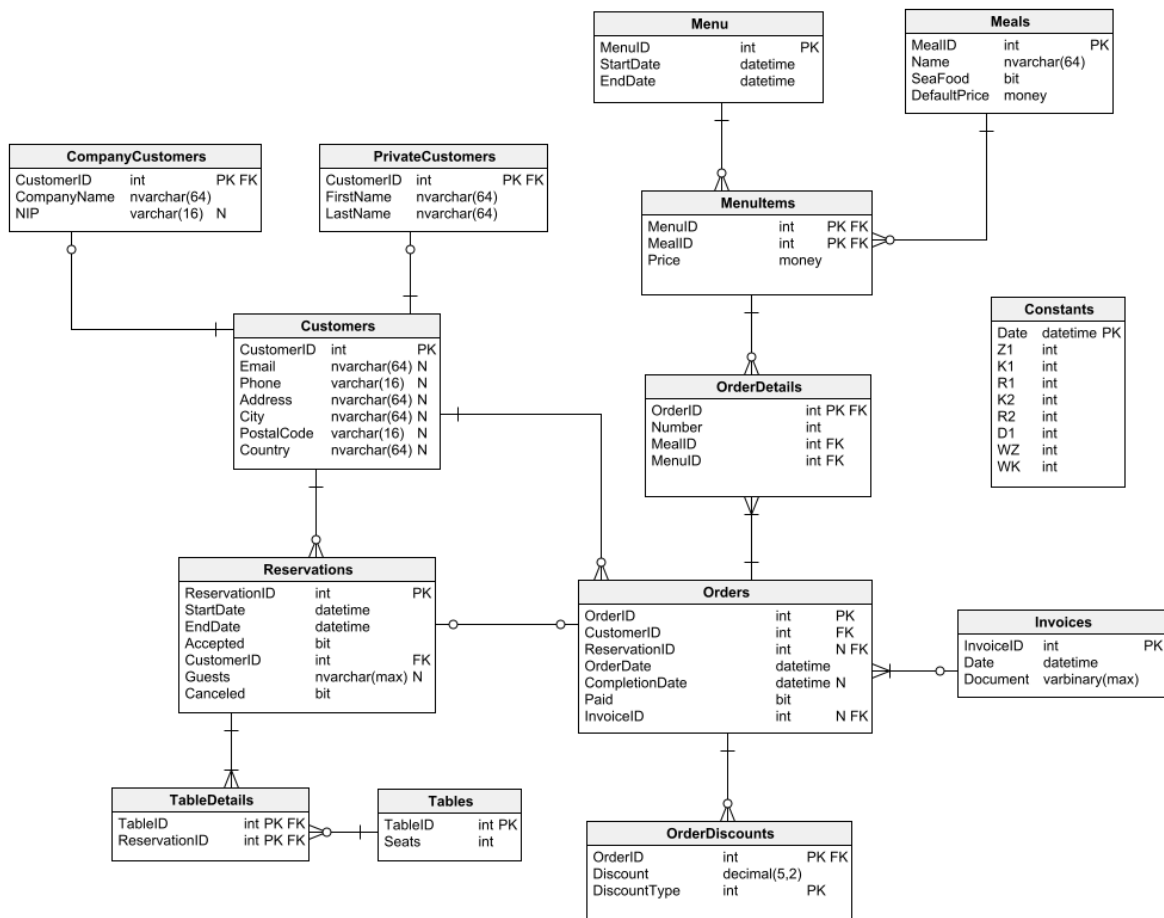
- Generowanie raportów (miesięcznych i tygodniowych) dotyczących rezerwacji, rabatów, menu, statystyk zamówień (kwoty, terminy czy zamówienie zostało złożone przez klienta indywidualnego, czy przez firmę)
- Modyfikacje menu — system sprawdza, czy menu jest zgodne z zasadami
- Możliwość zmiany parametrów rabatów
- Podgląd zamówień z owocami morza

Interpretacja rabatów

Zniżki dwóch typów można łączyć.

- Zniżka typu pierwszego — po realizacji Z1 zamówień, każde za co najmniej kwotę K1 klient dostaje stałą zniżkę R1% na wszystkie zamówienia
- Zniżka typu drugiego — po realizacji zamówień za łączną kwotę K2, przez następne D1 dni każde zamówienie otrzymuje zniżkę R2%. Jeśli ktoś spełni ponownie warunek, to zniżki nie łączą się, tylko okres zostanie przedłużony.

Schemat



Klienci

Przechowuje informacje wspólne o klientach indywidualnych i firmach. Informacje adresowe są opcjonalne (w przypadku kiedy są potrzebne, można je uzupełnić później).

```
CREATE TABLE Customers (  
    CustomerID int NOT NULL,  
    Email nvarchar(64) NULL,  
    Phone varchar(16) NULL,  
    Address nvarchar(64) NULL,  
    City nvarchar(64) NULL,  
    PostalCode varchar(16) NULL,  
    Country nvarchar(64) NULL,  
    CONSTRAINT Customers_pk PRIMARY KEY (CustomerID)  
);
```

Klienci indywidualni

Przechowuje informacje o klientach indywidualnych: numer klienta, imię, nazwisko.

```
CREATE TABLE PrivateCustomers (  
    CustomerID int NOT NULL,  
    FirstName nvarchar(64) NOT NULL,  
    LastName nvarchar(64) NOT NULL,  
    CONSTRAINT PrivateCustomers_pk PRIMARY KEY (CustomerID)  
);  
  
ALTER TABLE PrivateCustomers ADD CONSTRAINT Customers_PrivateCustomers  
    FOREIGN KEY (CustomerID)  
    REFERENCES Customers (CustomerID);
```

Firmy

Przechowuje informacje o firmach: numer firmy, nazwa firmy, (opcjonalny) NIP.

```
CREATE TABLE CompanyCustomers (  
    CustomerID int NOT NULL,  
    CompanyName nvarchar(64) NOT NULL,  
    NIP varchar(16) NULL,  
    CONSTRAINT CompanyCustomers_pk PRIMARY KEY (CustomerID)  
);  
  
ALTER TABLE CompanyCustomers ADD CONSTRAINT Customers_CompanyCustomers  
    FOREIGN KEY (CustomerID)  
    REFERENCES Customers (CustomerID);
```

Rezerwacje

Przechowuje informacje o rezerwacjach: numer rezerwacji, czas rozpoczęcia, czas zakończenia, czy rezerwacja została zaakceptowana przez obsługę, numer klienta, lista gości (opcjonalna, w przypadku firmy jako klienta) oraz czy rezerwacja została anulowana.

```
CREATE TABLE Reservations (  
    ReservationID int NOT NULL,  
    StartDate datetime NOT NULL,  
    EndDate datetime NOT NULL,  
    Accepted bit NOT NULL,  
    CustomerID int NOT NULL,  
    Guests nvarchar(max) NULL,  
    Canceled bit NOT NULL,  
    CONSTRAINT ReservationStartBeforeEnd CHECK (StartDate < EndDate),  
    CONSTRAINT Reservations_pk PRIMARY KEY (ReservationID)  
);  
  
ALTER TABLE Reservations ADD CONSTRAINT Reservations_Customers  
    FOREIGN KEY (CustomerID)  
    REFERENCES Customers (CustomerID);
```

Stoliki

Zawiera informacje o stolikach: numer stolika, liczba dostępnych miejsc. Liczba dostępnych miejsc musi być większa od zera.

```
CREATE TABLE Tables (  
    TableID int NOT NULL,  
    Seats int NOT NULL,  
    CONSTRAINT Tables_pk PRIMARY KEY (TableID),  
    CONSTRAINT PositiveSeatsNumber CHECK (Seats > 0)  
);
```

Rezerwacje stolików

Zawiera informacje o rezerwacjach stolików: numer stolika, numer rezerwacji.

```
CREATE TABLE TableDetails (  
    TableID int NOT NULL,  
    ReservationID int NOT NULL,  
    CONSTRAINT TableDetails_pk PRIMARY KEY (TableID, ReservationID)  
);  
  
ALTER TABLE TableDetails ADD CONSTRAINT Reservations_TableDetails  
    FOREIGN KEY (ReservationID)  
    REFERENCES Reservations (ReservationID);  
  
ALTER TABLE TableDetails ADD CONSTRAINT TableDetails_Tables  
    FOREIGN KEY (TableID)  
    REFERENCES Tables (TableID);
```

Zamówienia

Przechowuje informacje o zamówieniach: numer zamówienia, numer klienta, numer rezerwacji (lub null jeśli jej nie ma), czas złożenia zamówienia, czas wykonania zamówienia (jeśli zostało ukończone), czy zostało opłacone, numer faktury (lub null jeśli jej nie ma).

```
CREATE TABLE Orders (  
    OrderID int NOT NULL,  
    CustomerID int NOT NULL,  
    ReservationID int NULL,  
    OrderDate datetime NOT NULL,  
    CompletionDate datetime NULL,  
    Paid bit NOT NULL,  
    InvoiceID int NULL,  
    CONSTRAINT OrderedBeforeCompleted CHECK (CompletionDate >=  
OrderDate),  
    CONSTRAINT Orders_pk PRIMARY KEY (OrderID)  
);  
  
ALTER TABLE Orders ADD CONSTRAINT Orders_Customers  
    FOREIGN KEY (CustomerID)  
    REFERENCES Customers (CustomerID);  
  
ALTER TABLE Orders ADD CONSTRAINT Order_Reservations  
    FOREIGN KEY (ReservationID)  
    REFERENCES Reservations (ReservationID);  
  
ALTER TABLE Orders ADD CONSTRAINT Invoices_Orders  
    FOREIGN KEY (InvoiceID)  
    REFERENCES Invoices (InvoiceID);
```

Szczegóły zamówień

Zawiera informacje szczegółowe dla zamówień: numer zamówienia, ilość zamówionych dań danego typu, numer dania, numer menu.

```
CREATE TABLE OrderDetails (  
    OrderID int NOT NULL,  
    Number int NOT NULL,  
    MealID int NOT NULL,  
    MenuID int NOT NULL,  
    CONSTRAINT PositiveNumber CHECK (Number > 0),  
    CONSTRAINT OrderDetails_pk PRIMARY KEY (OrderID)  
);  
  
ALTER TABLE OrderDetails ADD CONSTRAINT Orders_OrderDetails  
    FOREIGN KEY (OrderID)  
    REFERENCES Orders (OrderID);  
  
ALTER TABLE OrderDetails ADD CONSTRAINT MenuItems_OrderDetails  
    FOREIGN KEY (MenuID,MealID)  
    REFERENCES MenuItems (MenuID,MealID);
```

Rabaty

Zawiera informacje o udzielonych rabatach: numer zamówienia, wartość rabatu (%), typ rabatu.

```
CREATE TABLE OrderDiscounts (  
    OrderID int NOT NULL,  
    Discount decimal(5,2) NOT NULL,  
    DiscountType int NOT NULL,  
    CONSTRAINT OrderDiscounts_pk PRIMARY KEY (OrderID,DiscountType)  
);  
  
ALTER TABLE OrderDiscounts ADD CONSTRAINT OrdersDiscounts_Orders  
    FOREIGN KEY (OrderID)  
    REFERENCES Orders (OrderID);
```

Menu

Przechowuje informacje o menu: numer menu, daty kiedy menu zaczyna i przestaje być dostępne, a także cena w danym okresie. W tabeli Meals są dania razem z ich podstawową ceną i informacją czy są to owoce morza.

```
CREATE TABLE Menu (  
    MenuID int NOT NULL,  
    StartDate datetime NOT NULL,  
    EndDate datetime NOT NULL,  
    CONSTRAINT MenuStartBeforeEnd CHECK (StartDate < EndDate),  
    CONSTRAINT Menu_pk PRIMARY KEY (MenuID)  
);  
  
CREATE TABLE Meals (  
    MealID int NOT NULL,  
    Name nvarchar(64) NOT NULL,  
    SeaFood bit NOT NULL,  
    DefaultPrice money NOT NULL,  
    CONSTRAINT Meals_pk PRIMARY KEY (MealID)  
);  
  
CREATE TABLE MenuItems (  
    MenuID int NOT NULL,  
    MealID int NOT NULL,  
    Price money NOT NULL,  
    CONSTRAINT MenuItems_pk PRIMARY KEY (MenuID,MealID)  
);  
  
ALTER TABLE MenuItems ADD CONSTRAINT MenuItems_Meals  
    FOREIGN KEY (MealID)  
    REFERENCES Meals (MealID);  
  
ALTER TABLE MenuItems ADD CONSTRAINT Menu_MenuItems  
    FOREIGN KEY (MenuID)  
    REFERENCES Menu (MenuID);
```


Faktury

Zawiera informacje o fakturach: numer faktury, data wystawienia faktury, faktura w formacie PDF.

```
CREATE TABLE Invoices (  
    InvoiceID int NOT NULL,  
    Date datetime NOT NULL,  
    Document varbinary(max) NOT NULL,  
    CONSTRAINT Invoices_pk PRIMARY KEY (InvoiceID)  
);
```

Stałe rabatowe

Zawiera informacje o wartościach stałych potrzebnych do wyznaczenia rabatów w danym okresie:

Z1 - minimalna liczba zamówień dla rabatu 1,

K1 - minimalna wydana kwota dla rabatu 1,

R1 - procent zniżki na wszystkie zamówienia po udzieleniu rabatu 1,

K2 - minimalna wydana kwota dla rabatu 2,

R2 - procent zniżki na zamówienie po udzieleniu rabatu 2,

D1 - maksymalna liczba dni na wykorzystanie rabatu 2 począwszy od dnia przyznania zniżki,

WZ - minimalna wartość zamówienia w przypadku wypełniania formularza do rezerwacji,

WK - minimalna ilość wykonanych zamówień w przypadku wypełniania formularza do rezerwacji.

```
CREATE TABLE Constants (  
    Date datetime NOT NULL,  
    Z1 int NOT NULL,  
    K1 int NOT NULL,  
    R1 int NOT NULL,  
    K2 int NOT NULL,  
    R2 int NOT NULL,  
    D1 int NOT NULL,  
    WZ int NOT NULL,  
    WK int NOT NULL,  
    CONSTRAINT ConstantChecks CHECK (Z1 >= 0 AND K1 >= 0 AND R1 >= 0  
AND R1 <= 100 AND K2 >= 0 AND R2 >= 0 AND R2 <= 100 AND D1 >= 0 AND  
WZ >= 0 AND WK >= 0 ),  
    CONSTRAINT Constants_pk PRIMARY KEY (Date)  
);
```