

BD2 - Katering

Jakub Mazurkiewicz (300226), Damian Piotrowski,
Konrad Wojewódzki(300286), Przemysław Wieczorkowski

Semestr 21L

Spis treści

1	Wstęp	3
1.1	Historia przedsiębiorstwa	3
1.2	Zasoby firmy	3
1.3	Cel projektu	3
2	Etap pierwszy	4
2.1	Model behawioralny	4
2.1.1	Aktorzy i ich przypadki użycia	4
2.1.2	Rozszerzone przypadki użycia	6
2.2	Model strukturalny	7
2.2.1	Słownik pojęć	7
2.2.2	Model ER	8
2.2.3	Macierz CRUD	8
2.3	Wybór narzędzi	8
3	Etap drugi	9
3.1	Model logiczny	9
3.2	Trzecia postać normalna	9
3.3	Opis więzów integralności	9
3.4	Projekt aplikacji	9
3.5	Opis wymagań funkcjonalnych	9
3.5.1	Bezpieczeństwo	9
3.5.2	Szybkość	9

3.6	Projekt testów	10
3.7	Projekt raportów analitycznych	10
3.7.1	Analiza ilości zamawianych produktów	10
3.7.2	Analiza opłacalności świadczonych usług	12
4	Etap trzeci	13
4.1	Wolumetria utworzonych tabel	13
4.2	Denormalizacja	18
4.3	Najczęstsze zapytania	18
4.4	Przykładowe dane	19
4.5	Aplikacja użytkowa	19
4.5.1	Zrealizowane przypadki użycia	19
4.5.2	Korzystanie z aplikacji	20
5	Podsumowanie	22

1 Wstęp

1.1 Historia przedsiębiorstwa

Przedsiębiorstwo “Cabbage Catering” zajmuje się dostarczaniem smacznych i pożywnych posiłków na imprezy okolicznościowe. Przedsiębiorstwo zostało założone w 2015 roku przez Przemysława Kapustkę, którego celem było stworzenie dobrze prosperującej firmy cateringowej. Nie spodziewał się on jednak, że firma urośnie do rozmiarów cateringowego królestwa i będzie realizowała bardzo duże ilości zamówień dla zróżnicowanych grup klientów. Z tego też powodu pojawiła się potrzeba stworzenia systemu komputerowego, który będzie wspomagał przedsiębiorstwo w sprawnym realizowaniu zamówień.

1.2 Zasoby firmy

Siedziba firmy umiejscowiona jest przy ulicy Urzędniczej 2 w Toruniu. W budynku znajduje się biuro, kuchnia, chłodnia oraz magazyn. W firmie są zatrudnieni: 8 kucharzy w tym 2 szefów kuchni, 3 cukierników, 2 dostawców oraz 15 kelnerów obsługujących gości na wydarzeniach. Przedsiębiorstwo dysponuje dwoma pojazdami transportowymi typu Mercedes Sprinter oraz dwoma pojazdami typu Mercedes AMG G63 dla przedstawicieli. Firma nabywa produkty spożywcze w sieci hurtowni Makro. Dzięki naszej infrastrukturze jesteśmy w stanie obsłużyć wydarzenia nawet do 400 osób. W ofercie znajduje się szeroki wybór dań, w tym dania wegańskie, wegetariańskie, bezglutenowe i tym podobne.

1.3 Cel projektu

Celem projektu jest stworzenie relacyjnej bazy danych do wspomagania obsługi klientów oraz logistyki przedsiębiorstwa. Powstanie także aplikacja, która wspomogę harmonogramowanie zamówień oraz monitorowanie stanu magazynu.

2 Etap pierwszy

2.1 Model behawioralny

2.1.1 Aktorzy i ich przypadki użycia

Przypadki użycia implementowane w aplikacji opisane są w pliku `app.pdf`.

1. **Pracownik recepcjonista** - przyjmuje od klientów zamówienia na dostarczanie usług cateringowych (za pośrednictwem telefonu) i wprowadza za pośrednictwem SZBD zlecenie do systemu. W razie wypadku informuje klienta o braku możliwości realizacji zamówienia. Przypadki użycia:

- Sprawdzenie dostępności terminu - sprawdzane przez aplikację,
- Dodanie zamówienia do terminarza,
- Usunięcie zamówienia z systemu,
- Zmiana szczegółów zamówienia,
- Przypisywanie kelnera do wydarzenia,
- Przypisywanie dostawcy do wydarzenia,
- Zatrudnianie nowych pracowników,
- Aktualizacja informacji o pracowniku.

2. **Szef kuchni** - pobiera z bazy danych informacje o zbliżających się wydarzeniach, sprawdza dostępność produktów na stanie (w magazynie/chłodni) i, w razie potrzeby, zamawia produkty niezbędne do przygotowania potraw. Sprawdza przepisy na zamówione potrawy. Może dodać własne przepisy i modyfikować menu. Przypadki użycia:

- Pobieranie listy dań do zamówienia,
- Edycja menu,
- Sprawdzanie dostępności produktów w magazynie,
- Pobranie listy produktów potrzebnych do wykonania dania,
- Edycja listy produktów potrzebnych do wykonania dania,
- Zamawianie potrzebnych produktów.

3. **Kucharz** - sprawdza dostępność produktów i przepisy. Przypadki użycia:
 - Sprawdzanie dostępności produktów w magazynie,
 - Pobranie listy produktów potrzebnych do wykonania dania.
4. **Pracownik dostawca** - odczytuje z systemu harmonogram wydarzeń i ustala trasę przejazdu, odbiera produkty z hurtowni i weryfikuje zgodność zamówień ze stanem faktycznym. Przypadki użycia:
 - Sprawdzanie grafiku,
 - Sprawdzanie listy dań do załadowania do pojazdu,
 - Sprawdzanie listy dostępnych pojazdów,
 - Sprawdzanie miejsce wydarzenia, do którego należy dostarczyć jedzenie.
5. **Pracownik kelner** - sprawdza harmonogram wydarzeń i obsługuje wydarzenie. Przypadki użycia:
 - Akceptuje zaproponowany mu w systemie termin (w przypadku pracownika okresowego),
 - Uwzględnianie dodatkowych kosztów (np. zniszczenia asortymentu) w trakcie wydarzenia
6. **Klient** - sprawdza dostępność wolnych terminów oraz koszt świadczonych usług i ew. składa zamówienie na wybrane menu (albo samodzielnie ustala listę potraw), określa liczbę gości, datę wydarzenia i lokalizację (ew. precyzuje rodzaj wydarzenia). Podaje podstawowe dane kontaktowe (imię, nazwisko, telefon, ew. e-mail). Dodatkowo może zrezygnować z korzystania z usług firmy/odwołać zaplanowane wydarzenie (najpóźniej na 2 tygodnie przed). Przypadki użycia:
 - Przeglądanie dostępnych dań.
 - Przeglądanie informacji o alergenach.

2.1.2 Rozszerzone przypadki użycia

Złożenie zamówienia

1. Klient składa zamówienie telefonicznie lub osobiście na recepcji co najmniej tydzień przed wydarzeniem
2. Pracownik wprowadza zlecenie do SZBD
3. System weryfikuje dostępność terminu
4. Jeśli termin jest wolny wydarzenie zostaje zapisane w systemie
5. Dane klienta, adres dostawy i lista dań są zapisywane w systemie

Sprawdzenie dostępności produktów

1. Szef kuchni sprawdza w systemie czy w magazynie znajdują się produkty potrzebne do wykonania zlecenia
2. Jeśli wszystkie produkty znajdują się na stanie zlecenie jest przekazywane do kuchni

Alternatywa:

1. Punkt pierwszy z przypadku podstawowego
2. Punkt drugi z przypadku podstawowego
3. Jeśli brakuje produktów zostaje złożone zamówienie w hurtowni

Przeprowadzenie dostawy:

1. Kurier pobiera z systemu adres i datę dostawy
2. System oblicza ile samochodów potrzeba do realizacji zamówienia
3. Pracownik sprawdza kompletność zamówienia
4. Jeśli zamówienie jest kompletne pracownik dostarcza posiłki

Alternatywa:

1. Punkt pierwszy z przypadku podstawowego

2. Punkt drugi z przypadku podstawowego
3. Punkt trzeci z przypadku podstawowego
4. Jeśli zamówienie nie jest kompletne pracownik informuje kuchnię o brakach w zamówieniu

2.2 Model strukturalny

2.2.1 Słownik pojęć

- **Produkt** - pojedynczy składnik przechowywany w magazynie lub w chłodni.
- **Pozycja na karcie** - posiłek lub napój do wyboru z naszej karty menu. Może składać się z wielu produktów oraz być różnej wielkości zgodnie z wolą klienta. Jest też udostępniona informacja o alergenach.
- **Informacje o daniu** - wszelkie przydatne dla klienta informacje o konkretnym daniu (np. czy danie jest wegańskie).
- **Przechowalnia** - miejsce, w którym przechowywane są nasze produkty spożywcze. Jest to magazyn lub chłodnia zależnie od rodzaju artykułu.
- **Zamówienie** - proces wyboru konkretnych dań z naszego menu przez klienta (możliwe są modyfikacje dania). Zamówienie musi zostać złożone co najmniej tydzień przed datą wydarzenia.
- **Klient** - podmiot składający zamówienie w naszej firmie. Może to być osoba fizyczna lub zarejestrowana firma.
- **Miejsce zamówienia** - lokalizacja, którą klient wybrał do dostarczenia zamówienia.
- **Pracownik** - osoba, do której należy obsługa wydarzenia.
- **Grafik pracownika** - grafik zawierający wydarzenia z określonymi ramami czasowymi. Wydarzenia przypisane są do konkretnego pracownika.
- **Dostawa** - proces dostarczenia zamówienia na miejsce.
- **Samochód** - pojazd używany do realizacji dostawy.

2.2.2 Model ER

Link do modelu ER w Miro: [LINK](#)

2.2.3 Macierz CRUD

Link do macierzy CRUD w arkuszu: [LINK](#)

2.3 Wybór narzędzi

Element	Narzędzie
Ekrany/UML	Miro
Dokumentacja	L ^A T _E X
System zarządzania bazą danych	MS SQL
Język programowania aplikacji	Python 3.9.5
Chmura	Microsoft Azure
Repozytorium kodu	Github

3 Etap drugi

3.1 Model logiczny

Model logiczny dostępny jest w pliku pdf w repozytorium Github: [LINK](#)

3.2 Trzecia postać normalna

Stworzony przez nas model logiczny został sprowadzony do trzeciej postaci normalnej. Świadczy o tym fakt, że w encjach żaden atrybut niekluczowy nie zależy od innych atrybutów niekluczowych.

3.3 Opis więzów integralności

Opis więzów integralności znajduje się pod adresem: [LINK](#)

3.4 Projekt aplikacji

Projekt aplikacji (realizowane przypadki użycia oraz dodatkowe diagramy) znajduje się w pliku `app.pdf`.

3.5 Opis wymagań funkcjonalnych

3.5.1 Bezpieczeństwo

Każdy użytkownik bazy danych ma generowane swój własny login i hasło do logowania do aplikacji oraz do bazy danych. Każdemu użytkownikowi nadawana jest rola i idą za nią uprawnienia. Oprócz tego konta administratorów są chronione “firewallem” i niezbędne jest podanie swojego adresu IP i wprowadzenie go w panelu administratora na stronie Microsoft Azure.

3.5.2 Szybkość

W aplikacji obsługującej harmonogram dostaw i układanie jadłospisów prędkość nie jest kluczowa. Pewne opóźnienia w działaniu zarówno aplikacji jak i bazy danych nie są krytyczne i w prawie żaden sposób nie wpływają na jakość usługi. Drobne opóźnienia wystąpią ze względu na to, że baza danych znajduje się w chmurze i synchronizacja zachodzących zmian nie jest natychmiastowa, w niektórych przypadkach może zająć to nawet kilka minut.

3.6 Projekt testów

- Ładujemy poprawnie wygenerowane dane w ilości zgodnej z wymaganiami
- Przeprowadzenie testów jednostkowych sprawdzających poprawność działania wyzwalaczy i weryfikacji:
 - Dodanie wydarzenia na konkretny dzień i godzinę;
 - Dodanie wydarzenia na zarezerwowany wcześniej termin;
 - Usunięcie nieistniejącego wydarzenia (o zadanej porze);
 - Zmiana daty wydarzenia na inny, wolny termin;
 - Zmiana daty wydarzenia na zajęty termin;
 - Próba wprowadzenia błędnej daty;
 - Próba ponownego zatrudnienia zatrudnionego pracownika o identycznych danych osobowych;
 - Próba ponownego dodania produktu o tej samej nazwie;
 - Sprawdzanie dostępności nieistniejących w magazynie produktów;
 - Próba ponownego dodania potrawy o tej samej nazwie i z tą samą listą produktów;
 - Wstawienie danych w złym formacie – daty (wydarzenia), kodu pocztowego, numeru telefonu, adresu e-mail, numeru NIP, numeru PESEL, numer konta
 - bankowego, wstawienie złego typu danych do określonego pola;
 - Próba dodania danych o nieprawidłowych kluczach obcych dla każdej z tabel

Testy przeprowadzane są z pomocą skryptu w języku Python.

3.7 Projekt raportów analitycznych

3.7.1 Analiza ilości zamawianych produktów

Chcemy przeanalizować ilość zamawianych z hurtowni produktów pod kątem realnego zapotrzebowania na nie. Pozwoli to lepiej oszacować zapotrzebowanie na produkty, zaplanować dostawy i ograniczyć straty związane z

upłynięciem terminu przydatności. W tym celu odpytujemy naszą bazę danych o wszystkie posiłki wykonane w zadanym przez użytkownika przedziale czasowym (miesiąc, tydzień etc.) i na podstawie tego szacujemy odsetek wykorzystanych produktów.

Nr	Produkt	Zakupiono	Wykorzystano	Jednostka	Procent
1	Ser_żółty	102	57	kg	55.9%
2	Twaróg	50	42	kg	84.0%
3	Jajka	200	170	szt	85.0%
4	Mleko	205	121	L	59.0%
5	Kalafior	10	7	kg	70.0%
6	Por	20	5	kg	25.0%
7	Marchew	20	10	kg	50.0%
8	Ziemniaki	222	217	kg	97.7%
9	Koperek	3000	2731	g	91.0%

3.7.2 Analiza opłacalności świadczonych usług

Chcemy przeanalizować świadczone przez firmę usługi pod kątem opłacalności - które wydarzenia przynoszą największe zyski przy jak najmniejszym nakładzie finansowym. W ten sposób możemy traktować priorytetowo niektóre formy działalności. Dzielimy zatem świadczone przez nas usługi na kategorie (urodziny, imieniny, chrzciny etc.) i analizujemy koszty związane z organizacją posiłków (koszty produktów, liczba kelnerów i ich wynagrodzenie, liczba potrzebnych samochodów dostawczych etc.) i porównujemy ze stawką którą zgodził się zapłacić klient.

Nr	Rodzaj	Przychody	Koszty	Zysk proc.
1	Cat.Rodzinny	31232	21212	32.1%
2	Cat.Dieta	30123	22543	25.2%
3	Chrzcziny	11222	10579	5.73%
4	Komunie	45631	31672	30.6%
5	Urodziny	23001	21521	6.43%
6	Firmowe	67123	59999	10.6%

4 Etap trzeci

Etap trzeci polegał na utworzeniu modelu fizycznego, implementacji bazy danych i aplikacji, oraz na utworzeniu przykładowych danych.

4.1 Wolumetria utworzonych tabel

Nazwa tabeli	Zawartość	Wolumetria
order	Zawiera wszystkie złożone zamówienia.	Jest to jedna z większych tabel w bazie danych. Wykonuje się na niej zapytania SELECT i INSERT .
additional_costs	Zawiera dodatkowe koszty dotyczące konkretnego zamówienia.	Może być to dosyć duża tabela. Wykonuje się na niej zapytania INSERT (wstawianie dodatkowych kosztów zamówień) oraz SELECT .
additional_information	Zawiera informacje o rodzaju jedzenia (np. czy jest wegańskie).	Jest to stosunkowo mała tabela, na której wykonuje się przeważnie operację SELECT .
address	Zawiera wszystkie używane w bazie adresy budynków i mieszkań.	Jest to duża tabela przechowująca informacje o adresie dla każdego pracownika, klienta oraz dostawy. Bardzo często wykonywane złączenie z tabelą City . Oprócz tego przy każdym nowym kliencie, zamówieniu oraz pracowniku wykonywane jest zapytanie INSERT oraz SELECT w celu pobrania id adresu i wstawienia go do odpowiedniej tabeli.
allergen	Zawiera alergenów występujących w produktach.	Jest to mała tabela na której wykonujemy tylko zapytanie SELECT (lista alergenów nie zmienia się).

business	Zawiera firmy będące klientami naszego przedsiębiorstwa.	Jest to dosyć duża tabela. Wykonujemy na niej zarówno operacje INSERT jak i SELECT .
city	Zawiera wszystkie miasta używane w bazie danych.	Jest to nieduża tabela niezbędna dla tabeli address . Najczęściej wykonywanym zapytaniem jest SELECT , ponieważ miasta i dzielnice do których dostarczamy pożywnie nie zmieniają się. Zapytanie INSERT wykonywane jest tylko w przypadku otwarcia nowej filii.
client	Zawiera wszystkich klientów, niezależnie od tego czy są firmą czy osobą fizyczną.	Jest to duża tabela łącząca tabele person oraz business . Przechowuje informacje o id, rodzaju klienta oraz jego adresie. Główne zapytania wykonywane na tej tabeli to INSERT oraz SELECT . SELECT jest wykonywane za każdym razem, gdy dodajemy do którejś z tabel dziedziczącej (business lub person) nową wartość. Jest to skutek działania wyzwalacza sprawdzającego czy przypisujemy klienta do odpowiedniego id.
delivery	Zawiera wszystkie zapisane w bazie dostawy.	Jest to duża tabela, niewiele mniejsza od tabeli Order . Wykonujemy na niej zapytania SELECT (sprawdzanie dostaw) oraz INSERT (dodawanie dostaw).
drink_sizes	Zawiera dostępne rozmiary napojów.	Jest to mała tabela (mniej niż 10 rzędów). Wykonujemy na niej zapytania SELECT w po

<code>drinks</code>	Zawiera wszystkie dostępne w menu napoje.	Jest to tabela średniej wielkości. Ilość rzędów zależy od ilości oferowanych napojów. Najczęściej wykonywanym zapytaniem jest <code>SELECT</code> . Rzadziej wykonywane jest zapytanie <code>INSERT</code> (tylko w przypadku zmiany menu).
<code>employee</code>	Zawiera wszystkich pracowników naszego przedsiębiorstwa.	Jest to tabela średniego rozmiaru. Głównie wykonujemy na niej zapytania <code>SELECT</code> w celu przypisania pracownika do konkretnego zamówienia lub dostawy. Zapytanie <code>INSERT</code> wykonywane jest tylko gdy zatrudniamy nowego pracownika.
<code>employee_schedule</code>	Zawiera zadania pracowników wraz z godzinami pracy.	Jest to duża tabela rosnąca w czasie. Częstym zapytaniem jest <code>INSERT</code> , ponieważ przypisywanie pracownika do zamówienia wiąże się z dodaniem nowego wydarzenia w jego grafiku. Innym częstym zapytaniem <code>SELECT</code> wykonywane w celu naniesienia zadań pracownika na jego kalendarz.
<code>employee_type</code>	Zawiera rodzaje pracowników (np. recepcjonista, dostawca).	Jest to tabela o stałym rozmiarze - w zasadzie nigdy nie powinno dojść do sytuacji, w której pojawia się nowy rodzaj pracownika. Najczęściej wykonywanym zapytaniem jest <code>SELECT</code> w celu znalezienia pracownika odpowiedniego rodzaju.

event_type	Zawiera wszystkie rodzaje wydarzeń.	Jest to tabela o niedużym rozmiarze. Najczęściej wykonywanym zapytaniem jest SELECT w celu pobrania znalezienia wydarzeń o konkretnym typie.
item_on_the_menu	Zawiera wszystkie pozycje dostępne w menu.	Rozmiar tej tabeli zależy od ilości oferowanych dań i napoi. Najczęściej wykonywanym zapytaniem jest SELECT (sprawdzanie dostępnych dań). Rzadziej wykonywane są INSERT i DELETE (edycja menu).
meals	Zawiera wszystkie dania dostępne w ofercie.	Rozmiar tej tabeli zależy od ilości dostępnych pozycji w menu. Najczęściej wykonywanym zapytaniem jest SELECT .
ordered_meals	Zawiera listę posiłków wybranych w konkretnym zamówieniu.	Jest to duża tabela - jej rozmiar rośnie wraz ze wzrostem rozmiaru tabeli order . Najczęściej wykonywanym zapytaniem na tej tabeli jest SELECT .
person	Zawiera wszystkich klientów będących osobami fizycznymi.	Rozmiar tej tabeli będzie rósł z czasem (im więcej zamówień tym więcej klientów). Najczęściej wykonywane zapytania to SELECT i INSERT .
product	Zawiera wszystkie produkty spożywcze, które mogą pojawić się w magazynie.	Jest to tabela średniej wielkości, najczęściej wykonywanym zapytaniem jest SELECT . Zapytanie INSERT (uwzględnianie nowych produktów) jest wykonywane dużo rzadziej.
storage	Zawiera możliwe przechowania.	Jest to mała tabela (2 rzędy), gdyż przedsiębiorstwo przechowuje produkty w chłodni lub w magazynie.

<code>stored_products</code>	Zawiera listę produktów przechowywanych obecnie w dostępnych przechowalniach.	Jest to tabela średniego rozmiaru, gdyż odwzorowuje ona stan magazynu. Często wykonuje się na niej operacje SELECT (sprawdzanie dostępności produktu) oraz INSERT i DELETE (aktualizacja zawartości magazynu).
<code>vehicles</code>	Zawiera wszystkie dostępne do w firmie pojazdy dostawcze.	Jest to mała tabela, operacje SELECT i DELETE wykonuje się na niej bardzo rzadko.
<code>ingredients</code>	Zawiera wszystkie produkty potrzebne do wytworzenia konkretnego dania.	Rozmiar tej tabeli zależy od rozmiaru tabeli <code>item_on_the_menu</code> i tego z ilu produktów składa się dane danie.
<code>employees_for_order</code>	Zawiera wszystkich pracowników (kelnerów) przypisanych do obsługi konkretnego zamówienia.	Rozmiar tej tabeli rośnie w czasie, wraz z rozmiarem tabeli <code>order</code> (im więcej zamówień tym więcej kelnerów z czasem).
<code>employees_for_delivery</code>	Jest to tabela zawierająca pracowników (dostawców) przypisanych do konkretnych dostaw.	Rozmiar tej tabeli rośnie w czasie, wraz z rozmiarem tabeli <code>delivery</code> .
<code>allergens_in_product</code>	Jest to tabela łącząca tabele <code>allergen</code> i <code>product</code> .	Rozmiar tej tabeli zależy głównie od rozmiaru tabeli <code>product</code> (jeden produkt może mieć wiele alergenów).
<code>size_of_drink</code>	Zawiera możliwe rozmiary dostępnych napoi.	Rozmiar tej tabeli zależy od rozmiarów tabel <code>drinks</code> i <code>drink_sizes</code>

4.2 Denormalizacja

1. Tabelę `drink_sizes` należy usunąć i w tabeli `drinks` dodać kolumnę `drink_size`. Przyspieszy to zapytania pobierające informacje o aktualnym menu. Z perspektywy aplikacji ułatwi to też edycję menu - operacja `INSERT` będzie w zasadzie wymagała podania rozmiaru napoju bez konieczności sprawdzania czy taki rozmiar istnieje w tabeli `drink_sizes`.
2. Tabelę `employee_type` należy usunąć i w tabeli `employee` dodać kolumnę `type`. Usunie to konieczność wykonywania zapytania `SELECT` gdy chcemy dodać pracownika odpowiedniego typu do obsługi zamówienia lub dostawy.
3. Tabelę `storage` należy usunąć i w tabeli `stored_products` dodać kolumnę `storage_type`. Ułatwi to wyszukiwanie produktów, ponieważ nie będzie konieczności wykonania operacji `JOIN` podczas wykonywania zapytania `SELECT`.
4. Warto rozważyć usunięcie tabeli `event_type` i dodanie kolumny `event_type` do tabeli `order`. Ułatwi to tworzenie zapytań wyszukujących wydarzenia danego typu. Pozytywnym skutkiem takiej denormalizacji może być przyspieszenie analizy przychodów z konkretnych rodzajów wydarzeń.

4.3 Najczęstsze zapytania

Zapytania najczęściej wykonywane przez aplikację:

1. **Pobieranie listy wydarzeń.** Zapytanie to wybiera wszystkie wydarzenia z danego miesiąca konkretnego roku z wykorzystaniem operacji `SELECT`. Ilość zwróconych rekordów może wahać się od 0 (brak zamówień w danym miesiącu) do około 31 (nasze małe przedsiębiorstwo realizuje zazwyczaj jedno większe zamówienie na dzień). W tym wypadku kolumnę `start_date` z tabeli `order` należy zaindeksować dla przyspieszenia tego zapytania (może być ono wykonywane wyjątkowo często przez recepcjonistę).
2. **Pobieranie informacji o adresie z konkretnego zamówienia.** Zapytanie to wykorzystuje operację `SELECT` w celu pobrania informacji o

adresie (uzyskiwane kolumny to m.in ulica, kod pocztowy, number budynku i mieszkania). Zapytanie to powinno zwrócić jeden rekord - dane jednego, konkretnego zamówienia.

3. **Dodawanie pracownika oraz jego adresu do bazy danych.** Zapytanie to wykorzystuje operacje INSERT w celu dodania nowego pracownika oraz jego adresu do bazy danych. Operacja dodania wykonuje się więc na trzech tabelach: `employee`, `address` oraz `city`. Operacja ta wykorzystuje również zapytania SELECT, aby znaleźć numery id nowo dodanych miast oraz adresów.
4. **Dodawanie nowego zamówienia.** Operacja ta polega na wykonaniu instrukcji INSERT. Dane o zamówieniu przenoszone są z formularza do zapytania po czym po DBMS sprawdza prawidłowość wprowadzonych danych.
5. **Pobieranie listy dań dostępnych w bazie.** Zapytanie to pobiera wszystkie dostępne pozycje w menu, w celu wyświetlenia ich nazw oraz cen. W tym wypadku nie warto rozważać indeksów pomocniczych - pobierane są wszystkie dane z tabeli `item_on_the_menu` (tabela ta nie jest też specjalnie duża).
6. **Przypisywanie pracowników do konkretnego zamówienia.** Operacja ta polega na pobraniu id zamówienia o wybranej dacie startowej, pobraniu id pracownika o danych podanych przez użytkownika, a następnie wstawieniu obu wartości do tabeli łączącej obie tabele. Operacja wykonuje się zatem na 3 tabelach.

4.4 Przykładowe dane

Skrypty DML generujące przykładowe dane zostały umieszczone w katalogu `/database/dml` w repozytorium [Github](#).

4.5 Aplikacja użytkowa

4.5.1 Zrealizowane przypadki użycia

Przypadki użycia, które zostały zrealizowane aplikacji użytkowej (nazwy takie same jak w pliku `app.pdf`):

1. Dodawanie zamówienia,
2. Usuwanie zamówienia,
3. Sprawdzanie dostępności terminu,
4. Przypisywanie pracowników do wydarzenia,
5. Dodawanie pracowników do bazy danych,
6. Uzyskiwanie informacji o zamówieniu (częściowo),
7. Sprawdzanie aktualnego menu,
8. Sprawdzanie stanu magazynu,
9. Edycja listy produktów (częściowo),
10. Sprawdzanie lokalizacji wydarzenia, do którego należy dostarczyć jedzenie,
11. Uwzględnianie dodatkowych kosztów zamówienia (np. z powodu uszkodzenia mienia).

4.5.2 Korzystanie z aplikacji

Pierwszym widokiem który widzi użytkownik jest ekran autoryzacji. Po podaniu nieprawidłowych danych użytkownikowi pokazuje się błąd. Po poprawnym zalogowaniu aplikacja przechodzi do panelu głównego. Zaimplementowany został ekran który widzi recepcjonista. Na ekranie widnieje kilka przycisków, między innymi pobranie listy dań, dodawanie nowego produktu, klienta i pracownika. Oprócz tego użytkownik ma możliwość przejścia do widoku kalendarza zamówień.

W widoku kalendarza użytkownikowi ukazuje się terminarz. Po kliknięciu przycisku **Show Events** na kalendarz nanoszone są wydarzenia i zostają zaznaczone kolorem żółtym. Po wybraniu daty zamówienia na kalendarzu użytkownik może kliknąć przycisk **Add Employee**. Po jego naciśnięciu na ekranie pojawiają się dwa formularze na wprowadzenie imienia i nazwiska oraz przycisk **Save**. Po wprowadzeniu danych i naciśnięciu przycisku **Save** pracownik zostaje przypisany do wybranego zamówienia. Z tego widoku

można również dodać zamówienie do bazy danych. W formularzu wprowadzamy adres zamówienia oraz dane klienta, który może być firmą lub osobą fizyczną (rodzaj klienta wybieramy odpowiednim przyciskiem).

Z panelu głównego możemy także zobaczyć aktualne menu. Służy do tego przycisk **Menu**, po którego kliknięciu ukaże się ekran z listą aktualnie dostępnych dań i napoi. Inną możliwą opcją jest dodanie nowego produktu, który **może** pojawić się w magazynie. Służy do tego przycisk **Products** w panelu głównym.

W panelu głównym możemy także znaleźć przycisk **Add Employee** (nie jest to ten sam przycisk co w kalendarzu). Jego naciśnięcie powoduje otwarcie formularza dodawania nowego pracownika do bazy danych. Podajemy w nim podstawowe dane, czyli imię i nazwisko, adres e-mail, numer telefonu i konta oraz adres. Formularz dodawania nowego klienta (przycisk **Add Client** w panelu głównym) jest bardzo podobny. Różni się w zasadzie podaniem rodzaju klienta i ewentualnym numerem NIP w przypadku firmy.

5 Podsumowanie

Zrealizowane zadania:

1. Opracowanie modelu pojęciowego (E-R) [Etap 1],
2. Zaprojektowanie funkcjonalności aplikacji: części operacyjnej oraz części raportowej [Etap 2],
3. Na podstawie modelu pojęciowego opracowanie relacyjnego logicznego modelu danych [Etap 2],
4. Optymalizacja modelu logicznego (w szczególności denormalizacja) w celu maksymalizacji wydajności systemu [Etap 2],
5. Dobór technologii bazodanowej [Etap 1], instalacja i konfiguracja środowiska [Etap 3],
6. Dobór technologii realizacji aplikacji [Etap 2], instalacja i konfiguracja środowiska rozwojowego [Etap 3],
7. Opracowanie, wdrożenie i optymalizacja modelu fizycznego [Etap 3],
8. Opracowanie dokumentacji analityczno-projektowej (w szczególności diagramów modeli danych z opisami) [Etap 3],
9. Opracowanie dokumentacji użytkowej aplikacji [Etap 3].