

Dokumentacja projektu bazy danych

Serwis z jedzeniem na wynos

Jakub Popowski

Spis treści

1. Analiza biznesowa projektowanej rzeczywistości	3
2. Model logiczny i relacyjny bazy danych	4
3. Oprogramowanie tworzące bazę danych	7
3.1. Sekwencje	7
3.2. Wyzwalacz	7
3.3. Perspektywy	8
3.3.A. <i>VIEW_ZAMOWIENIA_KTO_ILE</i>	8
3.3.B. <i>VIEW_ZAMOWIENIA_CO_ILE</i>	9
3.3.C. <i>VIEW_RANKING_OCEN</i>	10
3.3.D. <i>VIEW_MENU</i>	10
3.3.E. <i>VIEW_OCENY_RESTAURACJI</i>	11
3.3.F. <i>VIEW_SKLAD</i>	12
4. Funkcje	12
3.4.A. <i>FN_ZYSK_OKRES</i>	12
3.4.B. <i>FN_NAJTANSZA_RESTAURACJA</i>	13
3.4.C. <i>FN_CZY_KLIENT_ZAMOWIL_Z</i>	14
5. Procedury	14
3.5.A. <i>PR_CZY_KLIENT_ZAMOWIL_Z</i>	14
3.5.B. <i>PR_INSERT_OCENA</i>	15
4. Struktura katalogu projektu i elementy w nim zawarte. Instrukcja instalacji	17
5. Wyniki działania perspektyw analitycznych w postaci raportów pdf	18
5.1. Menu	18
5.2. Ranking ocen	21
5.3. Podsumowanie zamówienia	22
5.4. Podsumowanie zamówień	23
6. Przykłady	24

1. Analiza biznesowa projektowanej rzeczywistości

Serwis został zaprojektowany z myślą o obsłudze lokalnego rynku gastronomicznego tj. na terenie jednego miasta (aglomeracji). Projekt zawiera wiele funkcjonalności, które można podzielić na usługi dla restauracji i usługi dla klienta.

Podstawowe możliwości restauratora to:

- przyjmowanie zamówień od klientów
- ustalanie wielu kategorii dla swojej restauracji
- ustalenie własnego menu wraz z kategoriami poszczególnych produktów
- ustalenie dzielnic w których oferuje dowóz
- definiowanie składu swoich produktów.

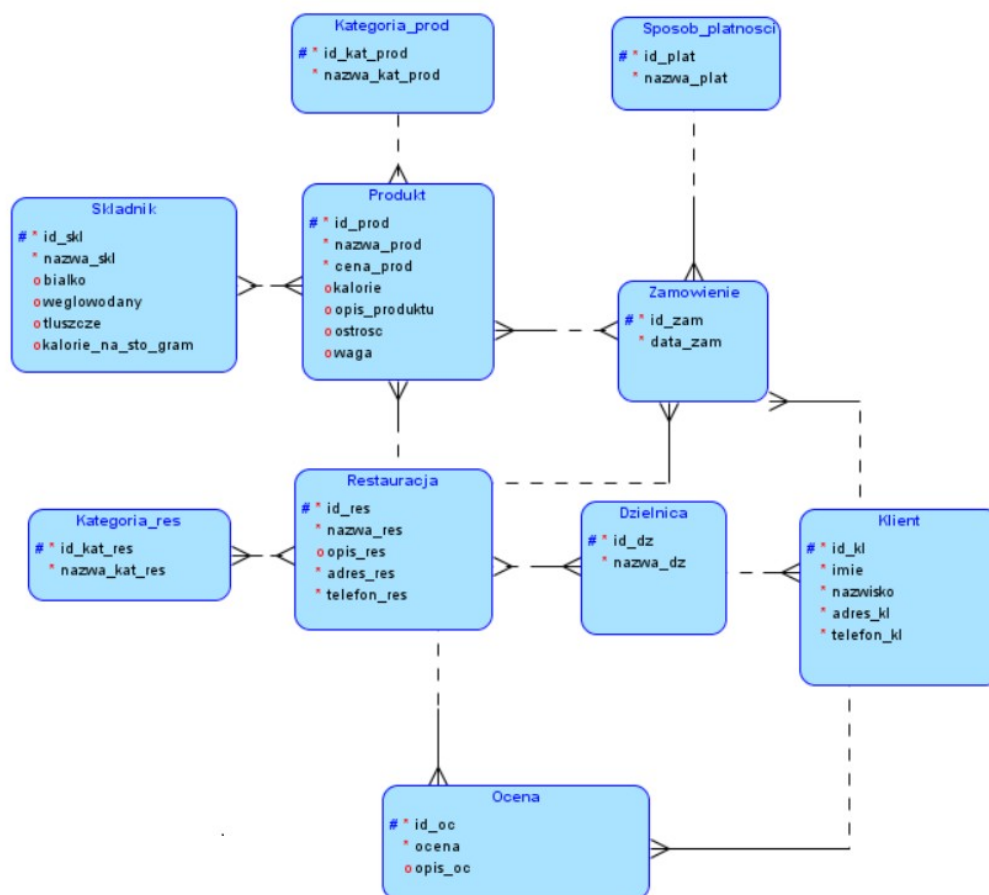
Podstawowe możliwości klienta to:

- złożenie zamówienia
- wyszukanie restauracji oferującej dowóz w wybranej lokalizacji (dzielnicy) i/lub filtrowanie po kategoriach, ocenach, średnich cenach itp.
- zarejestrowany klient nie musi każdorazowo wybierać lokalizacji jeżeli chce zamówić pod adres podany przy rejestracji
- ocenianie restauracji
- przeglądanie kalorii w produktach oraz bardziej szczegółowe przeglądanie składu produktów wraz z makroskładnikami i kaloriami.

Dodatkowo oprogramowanie tworzące bazę danych, opisane w punkcie 3, daje wiele innych funkcjonalności. Serwis automatyzuje proces tworzenia podstawowych dokumentów w formacie pdf takich jak podsumowanie zamówienia czy menu restauracji – szczegóły w punkcie 5.

2. Model logiczny i relacyjny bazy danych

Model logiczny został utworzony w programie Oracle Data Modeler:



Następnie w tym samym programie został przekształcony w model relacyjny:

Tabela produkt zawiera podstawowe i rozszerzone informacje o produkcie. Restaurator decyduje czy chce dodawać kalorie, ostrość czy listę składników do sprzedawanych przez siebie produktów.

Produkt może posiadać kategorię z tabeli kategoria_prod, która pozwoli na odpowiednie wylistowanie go w menu restauracji. Restaurator może skorzystać z gotowych nazw kategorii lub zdefiniować swoje.

Tabela restauracja zawiera podstawowe informację o restauracji.

Tabela kategoria_res definiuje kategorię, które restaurator może podpiąć pod swoją restaurację. Ułatwia to klientowi wyszukanie restauracji w zależności od tego na jakie jedzenie ma ochotę.

W przeciwieństwie do produktu, restauracja może posiadać wiele kategorii. Ich lista zawarta jest w tabeli kat_res – intersekcji między dwoma powyższymi tabelami.

Tabela klient zawiera podstawowe informację o kliencie.

W serwisie zaimplementowano system ocen restauracji – tabela ocena. Klient może wystawić opinię restauracji (założenie: nie musi wcześniej z niej zamówić, w punkcie 3 przedstawione jest rozwiązanie alternatywne). Opinia zawiera obowiązkowo „ilość gwiazdek” i opcjonalnie opis tekstowy.

Tabela dzielnica to tabela słownikowa zawierający listę dzielnic (obszarów) w danym mieście (aglomeracji). Klient przy rejestracji podaje dzielnicę. Restaurator natomiast definiuje dzielnicę pod które jego restauracja oferuje dowóz. Na ogół dowozy restauracji nie ograniczają się do jednej dzielnicy, ich lista znajduje się w tabeli rest_dziel. Atrybuty tabeli zamówienie to tak naprawdę tylko data zamówienia. Reszta to klucze obce z tabeli restauracja, klient i sposób_płatności.

Ta ostatnia to tabela słownikowa definiująca sposoby płatności dostępne w serwisie.

Ostatnią tabelą jest tabela zam_prod – intersekcja między tabelą zamówienie i tabelą produkt. Zawiera ona listę produktów zamówionych w jednym zamówieniu. Posiada 1 atrybut – ilość danego produktu.

3. Oprogramowanie tworzące bazę danych

3.1 Sekwencje

Dla tabel *składnik*, *restauracja*, *ocena*, *klient*, *produkt*, *zamówienie* zostały utworzone sekwencje odpowiednio *seq_skladnik*, *seq_restaracja*, *seq_ocena*, *seq_klient*, *seq_produkt*, *seq_zamówienie*. Automatyzują one tworzenie kluczy głównych w tych tabelach wykorzystując wartość domyślną odpowiedniego atrybutu. Wszystkie zostały utworzone w ten sam sposób, więc zamieszczony zostanie kod tylko jednej z nich:

```
create sequence seq_skladnik
increment by 1
start with 1
maxvalue 999999999999
minvalue 1
cache 20;
```

3.2 Wyzwalacz

Dla tabeli *skl_prod* został stworzony wyzwalacz *tr_skl_prod* dbający o walidację danych w tabeli *produkt*. Po dodaniu, usunięciu bądź zmodyfikowaniu elementu listy składników danego produktu wyzwalacz oblicza prawidłową wagę i ilość kalorii produktu. Kod wyzwalacza:

```
create or replace trigger tr_skl_prod
before insert or delete or update on skl_prod
for each row
begin
    if INSERTING then
        update produkt
        set waga=waga+:new.waga_skl
        where id_prod = :new.id_prod;
    update produkt
```

```

        set kalorie=kalorie+:new.waga_skl/100*(select
kalorie_na_sto_gram
        from skladnik where id_skl=:new.id_skl)
        where id_prod = :new.id_prod;
    end if;
    if UPDATING then
        update produkt
        set waga=waga+:new.waga_skl-:old.waga_skl
        where id_prod = :new.id_prod;

        update produkt
        set kalorie=kalorie+(:new.waga_skl/100 -
:old.waga_skl/100)*(select kalorie_na_sto_gram
        from skladnik where id_skl=:new.id_skl)
        where id_prod = :new.id_prod;
    end if;
    if DELETING then
        update produkt
        set waga=waga-:old.waga_skl
        where id_prod = :old.id_prod;
        update produkt
        set kalorie=kalorie-:old.waga_skl/100*(select
kalorie_na_sto_gram
        from skladnik where id_skl=:old.id_skl)
        where id_prod = :old.id_prod;
    end if;
end;

```

3.3 Perspektywy

3.3.A Perspektywa **VIEW_ZAMOWIENIA_KTO_ILE**

Perspektywa ta jest usługą dla restauracji. Wyświetla ona listę zamówień wraz z informacjami o kliencie, dacie, wartości zamówienia i sposobie płatności. Kod perspektywy:

```

CREATE OR REPLACE VIEW VIEW_ZAMOWIENIA_KTO_ILE AS
select zm.id_res as ID, nazwa_res as Restauracja, data_zam
as Data, imie || ' ' || nazwisko as Klient, telefon_kl as
Telefon, adres_kl as Adres, sum(cena_prod*ilosc) as Kwota,
nazwa_plat as Platnosc
from zamowienie zm
join klient k on zm.id_kl=k.id_kl
join restauracja r on zm.id_res = r.id_res
join sposob_platnosci s on zm.id_plat = s.id_plat
join zam_prod zp on zp.id_zam=zm.id_zam
join produkt p on p.id_prod=zm.id_prod
group by
data_zam, imie, nazwisko, telefon_kl, nazwa_plat, adres_kl, zm.id
_res, nazwa_res
order by data_zam;

```

Perspektywę należy wykonywać w zdaniu (1) aby pobrać całą historię zamówień, lub w zdaniu (2) jeżeli

interesuję nas jakiś okres czasu. Parametr *id* definiuję id restauracji, parametry *data_begin* i *data_end* datę od do w formacie *YYMMDD*.

```
select * from view_zamowienia_kto_ile where id=$P{id};--(1)

select * from view_zamowienia_kto_ile where id=$P{id}
and data between $P{data_begin} and $P{data_end}; --(2)
```

Wynik przykładowego wywołania perspektywy:

ID	RESTAURACJA	DATA	KLIENT	TELEFON	ADRES	KWOTA	PLATNOSC
3	WegeMANIA	21/01/01	Dian Piscopiello	9168086964	46 Arapahoe Road	94	Przy odbiorze gotówką
3	WegeMANIA	21/01/02	Laney Rainville	6202491101	1 Rowland Center	32	Przy odbiorze kartą
3	WegeMANIA	21/01/03	Flinn Calyton	1449339586	5110 Superior Alley	39	Przy odbiorze kartą
3	WegeMANIA	21/01/04	Jyoti Blaby	9597088012	23 Hanover Drive	24	Przez internet przelewy24
3	WegeMANIA	21/01/05	Maybelle Frentz	6737283137	453 Hintze Trail	287	Przez internet przelewy24
3	WegeMANIA	21/01/06	Celisse Parnby	5745183751	59156 Dawn Way	36	Przez internet przelewy24

3.3.B Perspektywa ***VIEW_ZAMOWIENIA_CO_ILE***

Perspektywa ta jest usługą dla klienta. Wyświetla ona podsumowanie zamówienia z informacjami takimi jak data, nazwa restauracji, adres i telefon klienta i restauracji, imię i nazwisko klienta, listę poszczególnych zamówionych produktów wraz z ceną i ilością oraz koszt całkowitym za daną pozycję. Kod perspektywy:

```
CREATE OR REPLACE VIEW VIEW_ZAMOWIENIA_CO_ILE AS
select zp.id_zam "ID",data_zam Data, nazwa_res Restauracja,
nazwa_prod Pozycja, adres_res, telefon_res,
imie || ' ' || nazwisko Klient,adres_kl, telefon_kl,
ilosc Ilosc, cena_prod Cena, ilosc*cena_prod Koszt,
nazwa_plat Platnosc
from zam_prod zp
join produkt p on zp.id_prod = p.id_prod
join restauracja r on p.id_res=r.id_res
join zamowienie zam on zp.id_zam = zam.id_zam
join klient k on zam.id_kl=k.id_kl
join sposob_platnosci sp on zam.id_plat = sp.id_plat;
```

Perspektywę należy wykonywać w prostym zdaniu określając jedynie id zamówienia – reprezentowane poniżej przez parametr *id*:

```
select * from view_zamowienia_co_ile where id=$P{id};
```

Wynik przykładowego wywołania perspektywy:

ID	DATA	RESTAURACJA	POZYCJA	ADRES_RES	TELEFON_RES	KLIENT	ADRES_KL	TELEFON_KL	ILOSC	CENA	KOSZT	PLATNOSC
17	21/01/07	Obiady domowe u Zosi Schabowy		Międzyborska 62/64	221290327	Molli Rosedale 7	Mendota Alley 8767896827		2	8	16	Przy odbiorze kartą
17	21/01/07	Obiady domowe u Zosi Mielony		Międzyborska 62/64	221290327	Molli Rosedale 7	Mendota Alley 8767896827		3	8	24	Przy odbiorze kartą
17	21/01/07	Obiady domowe u Zosi Kotlet z piersi kurczaka		Międzyborska 62/64	221290327	Molli Rosedale 7	Mendota Alley 8767896827		2	8	16	Przy odbiorze kartą
17	21/01/07	Obiady domowe u Zosi Ziemniaki		Międzyborska 62/64	221290327	Molli Rosedale 7	Mendota Alley 8767896827		2	2	4	Przy odbiorze kartą
17	21/01/07	Obiady domowe u Zosi Frytki		Międzyborska 62/64	221290327	Molli Rosedale 7	Mendota Alley 8767896827		4	5	20	Przy odbiorze kartą
17	21/01/07	Obiady domowe u Zosi Herbata		Międzyborska 62/64	221290327	Molli Rosedale 7	Mendota Alley 8767896827		3	2	6	Przy odbiorze kartą
17	21/01/07	Obiady domowe u Zosi Kompot		Międzyborska 62/64	221290327	Molli Rosedale 7	Mendota Alley 8767896827		3	1	3	Przy odbiorze kartą

3.3.C Perspektywa *VIEW_RANKING_OCEN*

Perspektywa ta wyświetla globalny ranking restauracji dostępnych w serwisie w oparciu o średnią z ocen klientów. Kod perspektywy:

```
create or replace view view_ranking_ocen as
select nazwa_res Restauracja, round(avg(ocena),2) "Średnia ocen"
from restauracja r
join ocena o on r.id_res=o.id_res
group by nazwa_res
order by "Średnia ocen" desc;
```

Aby wywołać perspektywę należy użyć zdania:

```
select * from view_ranking_ocen;
```

Wynik wywołania perspektywy:

Restauracja	Średnia ocen
1 Obiady domowe u Zosi	5
2 Big Johny PizzasPasta	4,8
3 Kuruahary Oriental	4,67
4 WegeMANIA	4,67
5 Sushi Cul	4,25
6 Burgerownia Bamboleo	4
7 Fast food Quickie	3,75
8 Pizzeria Hummus	3,5
9 Kuchnie świata	2,8
10 Kebab ALIF	1,8

3.3.D Perspektywa *VIEW_MENU*

Perspektywa ta wyświetla menu danej restauracji. Na menu standardowo składa się kategoria produktu, nazwa i cena. Kod perspektywy:

```
CREATE OR REPLACE VIEW VIEW_MENU AS
select produkt.id_res as Id, nazwa_res as Restauracja,
nazwa_kat_prod as Kategoria, nazwa_prod as Pozycja,
cena_prod as Cena
from produkt
join kategoria_prod on
produkt.id_kat_prod=kategoria_prod.id_kat_prod
join restauracja on produkt.id_res=restauracja.id_res;
```

Perspektywę należy wykonać w zdaniu określając id restauracji – reprezentowane przez parametr *id*:

```
select * from VIEW_MENU where id=$P{id }
```

Wynik przykładowego wywołania perspektywy:

	ID	RESTAURACJA	KATEGORIA	POZYCJA	CENA
1	6	Fast food Quickie Burgery	Burger Quick		14
2	6	Fast food Quickie Burgery	Burger Mate		13
3	6	Fast food Quickie Burgery	Burger Kurkum		19
4	6	Fast food Quickie Kurczak	Strips		12
5	6	Fast food Quickie Kurczak	BBQ wings		14
6	6	Fast food Quickie Przystawki	Frytki		7
7	6	Fast food Quickie Przystawki	Onion chips		10
8	6	Fast food Quickie Zimne napoje	Coca cola		6
9	6	Fast food Quickie Zimne napoje	Fanta		6
10	6	Fast food Quickie Zimne napoje	Sprite		6

3.3.E Perspektywa *VIEW_OCENY_RESTAURACJI*

Perspektywa ta pozwala na przeglądanie restauratorowi ocen wystawionych jego restauracji. Zawiera pola, które mogą być obiektem zainteresowania np. w przypadku wyjaśnienia negatywnej oceny. Z myślą o wcześniej wspomnianym wyjaśnianiu dane są domyślnie sortowane względem oceny rosnąco. Kod perspektywy:

```
create or replace view view_oceny_restauracji as
select id_res id, ocena Ocena, opis_oc Opis, imie Imie,
nazwisko Nazwisko, telefon_kl Telefon
from ocena oc
```

```
join klient kl on oc.id_kl=kl.id_kl
order by ocena;
```

Perspektywę należy wykonać w zdaniu określając id restauracji – reprezentowane przez parametr *id*:

```
select * from view_oceny_restauracji where id=$P{id };
```

Wynik przykładowego wywołania perspektywy:

ID	OCENA	OPIS	IMIE	NAZWISKO	TELEFON
1	1	2 Średnio. Jedzenie nie świeże.	Regan	Caldes	7354905735
2	1	3 (null)	Abigael	McJury	7048563988
3	1	4 (null)	Nisse	Pargetter	9186948835
4	1	5 Polecam!	Timotheus	Finders	4885788412

3.3.F Perspektywa *VIEW_SKLAD*

Perspektywa ta pozwala przeglądać skład produktu wraz z informacjami o kaloriach i ilości makroskładników w każdym z składników produktu. Kod perspektywy:

```
CREATE OR REPLACE VIEW VIEW_SKLAD AS
select sp.id_prod id, nazwa_skl Skladnik,
round(tluszcze*waga_skl/100,2) Tluszcze,
round(weglowodany*waga_skl/100,2) Weglowodany,
round(bialko*waga_skl/100,2) Bialko, waga_skl Waga,
round(kalorie_na_sto_gram*waga_skl/100,2) Kalorie
from skl_prod sp
join skladnik s on sp.id_skl=s.id_skl
join produkt p on sp.id_prod = p.id_prod
order by kalorie desc;
```

Perspektywę należy wykonać w zdaniu określając id produktu – reprezentowane przez parametr *id*:

```
select * from view_sklad where id=$P{id };
```

Wynik przykładowego wywołania perspektywy:

ID	SKŁADNIK	TLUSZCZE	WEGLOWODANY	BIALKO	WAGA	KALORIE
1	26ser aldamer	35,1	1,95	33,8	130	468
2	26jarmuż	0,6	5,4	2,85	150	45
3	26papryka	0,3	3,08	1,28	75	24
4	26pomidor	0,25	4,5	1,13	125	18,75
5	26ogórek	0,1	3,1	0,7	100	16

3.4 Funkcje

3.4.A Funkcja FN_ZYSK_OKRES

Funkcja oblicza zysk restauracji w danym okresie. Jako parametr przyjmuje id restauracji, datę początkową oraz datę końcową o typach odpowiednio *int*, *date*, *date*. Wartość zwracana to obliczony zysk o typie *int*. Kod funkcji:

```
create or replace function fn_zysk_okres
(v_id_res number, v_pocz date, v_koniec date)
return number
as
    v_zysk number(10);
begin
    select sum(ilosc*cena_prod) into v_zysk
    from zam_prod zp
    join produkt p on zp.id_prod = p.id_prod
    join zamowienie z on zp.id_zam = z.id_zam
    where z.id_res=v_id_res and data_zam between v_pocz and
v_koniec;
return v_zysk;
end;
```

Przykładowe wywołanie funkcji:

```
select fn_zysk_okres(3, '2021/01/01', '2021/01/10') Zysk
from dual;
```

3.4.B Funkcja FN_NAJTANSZA_RESTAURACJA

Funkcja znajduje najtańszą restaurację tj. restauracją z najniższą średnią ceną za produkt w dzielnicy określonej przez klienta. Przyjmuje ona jeden parametr – id dzielnicy na którą realizowane ma być zamówienie – o typie *int*. Wartość zwracana to wyliczone id restauracji o typie *int*. Kod funkcji:

```

create or replace function fn_najtansza_restauracja
    (v_id_dz number)
return number
as
    v_id_res number(4);
begin
    select id_res into v_id_res from restauracja
    where id_res =
        (select id_res from
            (select rownum lp, id_res from
                (select sl.id_res, avg(cena_prod) cena from
                    (select id_res from rest_dziel where
id_dz=v_id_dz) sl
                join produkt p on sl.id_res=p.id_res
                group by sl.id_res
                order by cena)
            where rownum=1)
        where lp=1);
return v_id_res;
end;

```

Przykładowe wywołanie funkcji:

```

select fn_najtansza_restauracja(1) from dual;

```

3.4.C Funkcja FN_CZY_KLIENT_ZAMOWIL_Z

Funkcja sprawdza czy klient kiedykolwiek zamawiał z restauracji. Robi to poprzez przejrzanie tabeli *zam_prod* i zliczenie pasujących rekordów. Przyjmuje dwa argumenty – id restauracji i id klienta – oba o typie *int*. Zwracana wartość to wartość logiczna typu *bool*. Kod funkcji:

```

create or replace function fn_czy_klient_zamowil_z
    (v_id_res number, v_id_kl number)
return boolean
as
    v_licznik number(3);
begin
    select count(zp.id_zam) into v_licznik from zam_prod zp
    join zamowienie z on zp.id_zam=z.id_zam
    join produkt p on zp.id_prod = p.id_prod
    join restauracja r on r.id_res = p.id_res
    where r.id_res=v_id_res and id_kl=v_id_kl;

```

```

    if v_licznik=0 then return false;
    else return true;
  end if;
end;
```

Funcja logiczna nie może być wywoływana samodzielnie. Procedura ją obsługująca opisana jest w punkcie 3.5.A.

3.5 Procedury

3.5.A Procedura PR_CZY_KLIENT_ZAMOWIL_Z

Procedura obsługującą funkcję logiczną o bliźniaczej nazwie. Przyjmuje takie same argumenty - id restauracji i id klienta - oba o typie *int*. Jej działanie ogranicza się do wywołania funkcji dla danych argumentów i wydrukowanie odpowiedniego komunikatu. Zastosowaniem tej pary procedura-funkcja może być przyznawanie nowym klientom zniżek. Kod procedury:

```

create or replace procedure pr_czy_klient_zamowil_z
  (v_id_res number, v_id_kl number)
as
  v_imie varchar2(12);
  v_nazwisko varchar2(20);
  v_nazwa_res varchar2(30);
begin
  select nazwa_res into v_nazwa_res from restauracja
where id_res=v_id_res;
  select imie, nazwisko into v_imie, v_nazwisko from
klient where id_kl = v_id_kl;
  if fn_czy_klient_zamowil_z(v_id_res, v_id_kl)
    then dbms_output.put_line(v_imie||' '||
v_nazwisko||' witamy ponownie w '||v_nazwa_res);
    elsif not fn_czy_klient_zamowil_z(v_id_res, v_id_kl)
then
      dbms_output.put_line(v_imie||' '||v_nazwisko||'
witamy w '||v_nazwa_res||'. Przygotowalismy dla Ciebie
znizki na pierwsze zamowienie');
    end if;
end;
```

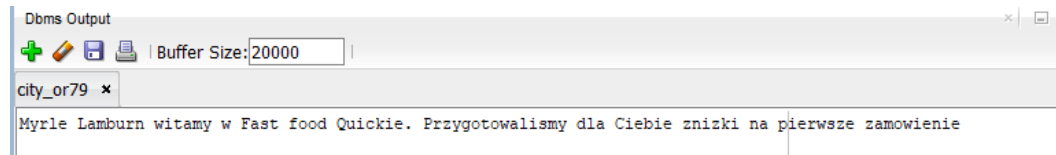
Wywołanie procedury z przykładowymi argumentami:

```

declare
begin
pr_czy_klient_zamowil_z(6, 98);
end;

```

Daje wynik:



W innym przypadku możemy otrzymać:



3.5.B Procedura PR_INSERT_OCENA

Procedura może mieć zastosowanie, gdy restaurator będzie chciał otrzymywać oceny tylko od osób, które już zamawiały w jego restauracji. Zastępuje ona standardowe zdanie insert do tabeli ocena. Argumenty procedury to wartość liczbowa oceny, id restauracji, id klienta, opis odpowiednio o *typach* *int*, *int*, *int*, *varchar2*. Procedura wykorzystuje funkcję *FN_CZY_KLIENT_ZAMÓWIŁ_Z* i w zależności od jej wyniku wykonuje odpowiednią akcję. W obu przypadkach drukowany jest odpowiedni komunikat.

Kod procedury:

```

create or replace procedure pr_insert_ocena
(
    v_ocena number,
    v_id_res number,
    v_id_kl number,
    v_opis varchar2 default null)
as
v_nazwa_res varchar2(30);
begin
    select nazwa_res into v_nazwa_res from restauracja
    where id_res=v_id_res;
    if fn_czy_klient_zamowil_z(v_id_res, v_id_kl) then
        insert into ocena(ocena, opis_oc, id_res, id_kl)
values(v_ocena, v_opis, v_id_res, v_id_kl);
        dbms_output.put_line('Wystawiono pomyslnie ocene

```



```

'||v_ocena||' restauracji '||v_nazwa_res);
    elsif not fn_czy_klient_zamowil_z(v_id_res, v_id_kl)
then
    dbms_output.put_line('Aby wystawic ocene restauracji
musisz z niej cos najpierw zamowic!');
    end if;
end;

```

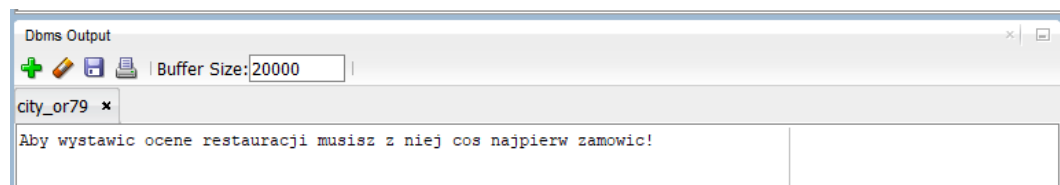
Wywołanie procedury z przykładowymi argumentami:

```

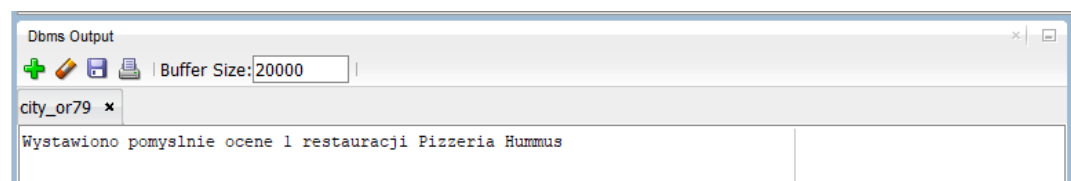
declare
begin
pr_insert_ocena(1, 8, 60);
end;

```

Daje wynik:



W innym przypadku możemy otrzymać:



4. Struktura katalogu projektu i elementy w nim zawarte. Instrukcja instalacji.

Katalog główny projektu zawiera następujące katalogi:

- data – przykładowe pliki csv do wypełnienia tabel
- datamodeler – folder ze schematem utworzonym w Oracle Data Modeler
- doc – dokumentacja projektu
- report – pliki aplikacji Jasper Reports, opisane w następnym punkcie
- script – skrypty wykorzystywane do instalacji projektu na serwerze

Ponadto w głównym katalogu projektu znajdują się pliki:

- install.sql – skrypt do instalacji projektu

- `install_example.sql` – skrypt do instalacji projektu z przykładowymi danymi
- `delete.ddl` – skrypt do deinstalacji projektu
- `example.sql` – przykładowe zdania *select*

Aby zainstalować projekt na komputerze z systemem operacyjnym MS Windows należy wykonać skrypt *install.sql* na serwerze bazodanowym. Można to zrobić instrukcją:

```
@<ścieżka do folderu z projektem>\gutfud\install.sql;
```

Deinstalacja bądź instalacja z przykładowymi danymi odbywa się w ten sam sposób – należy podać odpowiedni skrypt z głównego katalogu projektu.

Dla komputera z systemem operacyjnym z rodziny UNIX:

```
--path - ścieżka do folderu z projektem
--instalacja
@<path>/gutfud/script/create_structure.ddl
@<path>/gutfud/script/create_objects.sql

--instalacja z przykładowymi danymi
@<path>/gutfud/script/create_structure.ddl
@<path>/gutfud/script/push_data.sql
@<path>/gutfud/script/create_example_objects.sql

--deinstalacja
@<path>/gutfud/delete.ddl
```

5. Wyniki działania perspektyw analitycznych w postaci raportów pdf

Perspektywy zostały w większości wykorzystane do stworzenia systemu dynamicznych raportów z wykorzystaniem oprogramowania Jasper Reports. Niektóre,

na pierwszy rzut oka, niepotrzebne atrybuty perspektyw znalazły tu swoje zastosowanie. Folder report zawiera kod raportu oraz przykład .pdf dla każdego z utworzonych raportów.

5.1 Raport menu

Usługa dla restauratora pozwalająca na wygenerowanie menu restauracji w prostym szablonie. Na górze strony znajdują się nagłówek z nazwą restauracji i logiem serwisu. Tak wygenerowane menu wyszczególnia kategorię, na które pogrupowane są produkty. Przy każdym produkcie znajdują się jego cena. Do wygenerowania raportu wymagany jest jeden parametr – id restauracji. Przykład raportu:

Menu

Obiady domowe u Zosi



Dania obiadowe

Schabowy	8
Mielony	8
Kotlet z piersi kurczaka	8

Zupy

Pomidorówka	5
Rosół	6
Grochówka	8

Przystawki

Mizeria	4
Marchewka	4
Kapusta kiszona	4
Ziemniaki	2
Frytki	5

Ciepłe napoje

Herbata	2
---------	---

Zimne napoje

Kompot	1
--------	---

Pierogi

Pierogi z mięsem	12
Pierogi z serem	10
Pierogi ruskie	11
Pierogi z kapustą i grzybami	13

5.2 Raport ranking_ocen

Raport generuję ranking restauracji w oparciu o średnią z ocen. Na górze strony znajdują się nagłówek z datą wygenerowania raportu oraz logiem serwisu. Pod nagłówkiem znajdują się tabela z polami liczba porządkowa, nazwa restauracji, średnia ocen. Rekordy posegregowane są malejąco. Raport nie przyjmuje żadnego parametru. Przykład raportu:

Ranking restauracji



Stan na dzień: sobota 30 styczeń 2021

Lp	Restauracja	Średnia ocen
1	Obiady domowe u Zosi	5
2	Big Johny Pizza&Pasta	4.8
3	Kuruhary Oriental	4.67
4	WegeMANIA	4.67
5	Sushi Cul	4.25
6	Burgerownia Bamboleo	4
7	Fast food Quickie	3.75
8	Pizzeria Hummus	3.4
9	Kuchnie świata	2.8
10	Kebab ALIF	1.8

5.3 Raport zamowienia_co

Raport generuje rachunek / fakturę / podsumowanie zamówienia dla klienta. Na górze strony znajdują się nagłówek z datą złożenia zamówienia i logiem serwisu. Tuż pod nim drukowane są dane klienta (kupującego) i restauracji (sprzedającego). Poniżej znajduje się tabela zawierająca zamówione produkty – ich nazwy, cenę jednostkową, ilość i koszt. Pod tabelą drukowana jest informacja o całkowitym koszcie i sposobie płatności wybranym przez klienta. Raport przyjmuje jeden parametr – id zamówienia. Przykład raportu:

Podsumowanie zamówienia



Data złożenia zamówienia: 2021-01-08 00:00:00.0

Restauracja sprzedająca:

Burgerownia Bamboleo
Grochowska 72
612956213

Kupujący:

Roman Kinze
94239 Mallard Terrace
5228549786

Lp	Pozycja	Cena	Ilość	Koszt
1	Burger 1	25	2	50
2	Burger 4	21	2	42
3	Frytki	12	1	12
4	Frytki zygzaki	13	1	13
5	Frytki belgijskie	18	1	18
6	Salatka Coleslaw	9	1	9

Suma 144 PLN

Sposób płatności: Przez internet Dotpay

gutfud.pl

5.4 Raport zamowienia_kto

Raport generuję podsumowanie zamówień restauracji w danym okresie. Na górze strony znajdują się nagłówki zawierający nazwę restauracji, datę wygenerowania raportu oraz logo serwisu. Pod spodem drukowana jest tabela z informacją o dacie zamówienia, danymi klienta, kwotą zamówienia oraz sposobem płatności. Pod tabelą znajdują się informacje o okresie z którego generowany jest raport i sumie przychodów z zamówień w danym okresie. Raport przyjmuje 3 argumenty – id restauracji,

datę początkową okresu raportowania oraz datę końcową okresu raportowania. Przykład raportu:

Podsumowanie zamówień



WegeMANIA

Stan na dzień 29/01/2021

Data	Klient	Adres	Telefon	Kwota	Sposób płatności
01/01/21	Dian Piscopiello	46 Arapahoe Road	9168086964	94	Przy odbiorze gotówką
02/01/21	Laney Rainville	1 Rowland Center	6202491101	32	Przy odbiorze kartą
03/01/21	Flinn Calyton	5110 Superior Alley	1449339586	39	Przy odbiorze kartą
04/01/21	Jyoti Blaby	23 Hanover Drive	9597088012	24	Przez internet przelewy24
05/01/21	Maybelle Frentz	453 Hintze Trail	6737283137	287	Przez internet przelewy24
06/01/21	Celisse Parnby	59156 Dawn Way	5745183751	36	Przez internet przelewy24
07/01/21	Siward Batchelour	8736 Sunnyside Circle	4367701593	48	Przy odbiorze kartą
08/01/21	Debor Mansel	44 Mcguire Park	4518597072	28	Przez internet Dotpay
09/01/21	Roma Nacci	6 Lien Drive	6434914452	64	Przez internet PayPal
10/01/21	Diahann Aleksashin	8 Randy Trail	5748825470	190	Przy odbiorze gotówką

Okres od 01/01/2021 do 10/01/2021

Suma 842 PLN

gutfud.pl

6. Przykłady

Kod wszystkich poniższych przykładów znajduje się w pliku *examples.sql* w głównym katalogu projektu.

Pobranie listy restauracji oferujących dowóz do danej dzielnicy:

```
select nazwa_res from rest_dziel rd
join restauracja r on rd.id_res = r.id_res
join dzielnica d on rd.id_dz=d.id_dz
where nazwa_dz='Ursynów';
```

NAZWA_RES
1 WegeMANIA
2 Kebab ALIF
3 Fast food Quickie
4 Kuruahary Oriental

Pobranie listy restauracji oferujących dowóz do danej dzielnicy z uwzględnieniem ich miejsca w rankingu z perspektywy *view_ranking_ocen* oraz ograniczenie wyników do danej liczby pozycji:

```
select rownum Pozycja, "Restauracja", "Średnia ocen" from
(select "Restauracja", "Średnia ocen" from
view_ranking_ocen
where "Restauracja" in (select nazwa_res from rest_dziel rd
join restauracja r on rd.id_res = r.id_res
join dzielnica d on rd.id_dz=d.id_dz
where nazwa_dz='Ursynów')
order by "Średnia ocen" desc)
where rownum <=3;
```

POZYCJA	Restauracja	Średnia ocen
1	1 Kuruahary Oriental	4,67
2	2 WegeMANIA	4,67
3	3 Fast food Quickie	3,75

Pobranie listy restauracji o danej kategorii oferujących dowóz do danej dzielnicy:

```
select nazwa_res "Restauracja" from rest_dziel rd
join restauracja r on rd.id_res = r.id_res
join dzielnica d on rd.id_dz=d.id_dz
join kat_res kr on r.id_res = kr.id_res
join kategoria_res kres on kr.id_kat_res=kres.id_kat_res
where nazwa_dz='Ursynów' and nazwa_kat_res='Wegetariańskie';
```

Restauracja
1 WegeMANIA

Lista kategorii restauracji z ilością restauracji z nimi związaną:

```
select nazwa_kat_res Restauracja, count(kat_res.id_kat_res)
Ilosc from kat_res
join kategoria_res on
kat_res.id_kat_res=kategoria_res.id_kat_res
group by nazwa_kat_res
order by count(kat_res.id_kat_res) desc;
```

RESTAURACJA	ILOSC
1 Wegetariańskie	3
2 Amerykańskie	3
3 Ukraińskie	3
4 Włoskie	3
5 Orientalne	3
6 Niemieckie	2
7 Tureckie	1
8 Japońskie	1
9 Polskie	1
10 Indyjskie	1

Historia zamówień klienta posortowana względem daty malejąco:

```
select distinct data_zam Data, nazwa_res Restauracja,
(select sum(ilosc) from zam_prod join
zamowienie on zam_prod.id_zam=zamowienie.id_zam
where id_kl=9) "Ilosc produktow",
(select sum(ilosc*cena_prod) from zam_prod join
zamowienie on zam_prod.id_zam=zamowienie.id_zam
join produkt on zam_prod.id_prod=produkt.id_prod
where id_kl=9) "Koszt", nazwa_plat "Sposób platnosci"
from zam_prod zp join zamowienie z on zp.id_zam=z.id_zam
join restauracja r on z.id_res = r.id_res
join sposob_platnosci sp on z.id_plat = sp.id_plat
where id_kl=9
order by Data;
```

DATA	RESTAURACJA	Ilosc produktow	Koszt	Sposób platnosci
1 21/01/09	Pizzeria Hummus	3	42	Przez internet PayPal