

Projekt **Bazy Danych 2**

Wydział Elektrotechniki Automatyki i Informatyki

Politechnika Świętokrzyska

Studia: **Stacjonarne I stopnia**

Kierunek: **Informatyka**

Data oddania: **12.06.2017**

Grupa: **2ID15**

Ocena:

1. Bartłomiej Osak
2. Tomasz Pasternak
3. Monika Molenda

Temat projektu:

Hurtownia danych do analiz – Warsztat szewsko - kaletniczy

1. Opis projektu.

Projekt przedstawia hurtownię danych do analiz. Tematem przewodnim jest warsztat szewsko-kaletniczy. Hurtownia została zbudowana na podstawie schematu płatka śniegu, co wiąże się z normalizacją wymiarów do trzeciej postaci normalnej.

W projekcie zdefiniowaliśmy dwanaście tabel.

- **TABELA FAKTÓW: ZLECENIE** - główna tabela w naszym schemacie. Przechowujemy w niej pojedyncze fakty. Zdefiniowaliśmy również cztery miary w celu przeprowadzania poprawnych analiz.
- **TABELE WYMIARÓW: TYP ZLECENIA, KLIENT, DATA** oraz **PRACOWNIK**. Wymiary opisują dany fakt poprzez atrybuty wymiaru. Każdy z wymiarów jest znormalizowany do trzeciej postaci normalnej.
- **TABELE PODRZĘDNE WYMIARÓW: TYP PRODUKTU, GRUPA KIENTELI, STANOWISKO, DZIEŃ TYGODNIA, MIESIĄC, KWARTAŁ** oraz **ROK**. Tabele podrzędne wymiarów powstały na wskutek normalizacji wymiarów naszej hurtowni danych do trzeciej postaci normalnej. Posiadają one atrybuty opisujące poszczególne wymiary.

Celem analizy hurtowni danych napisaliśmy zapytania analityczne. Zastosowaliśmy operatory grupujące takie jak:

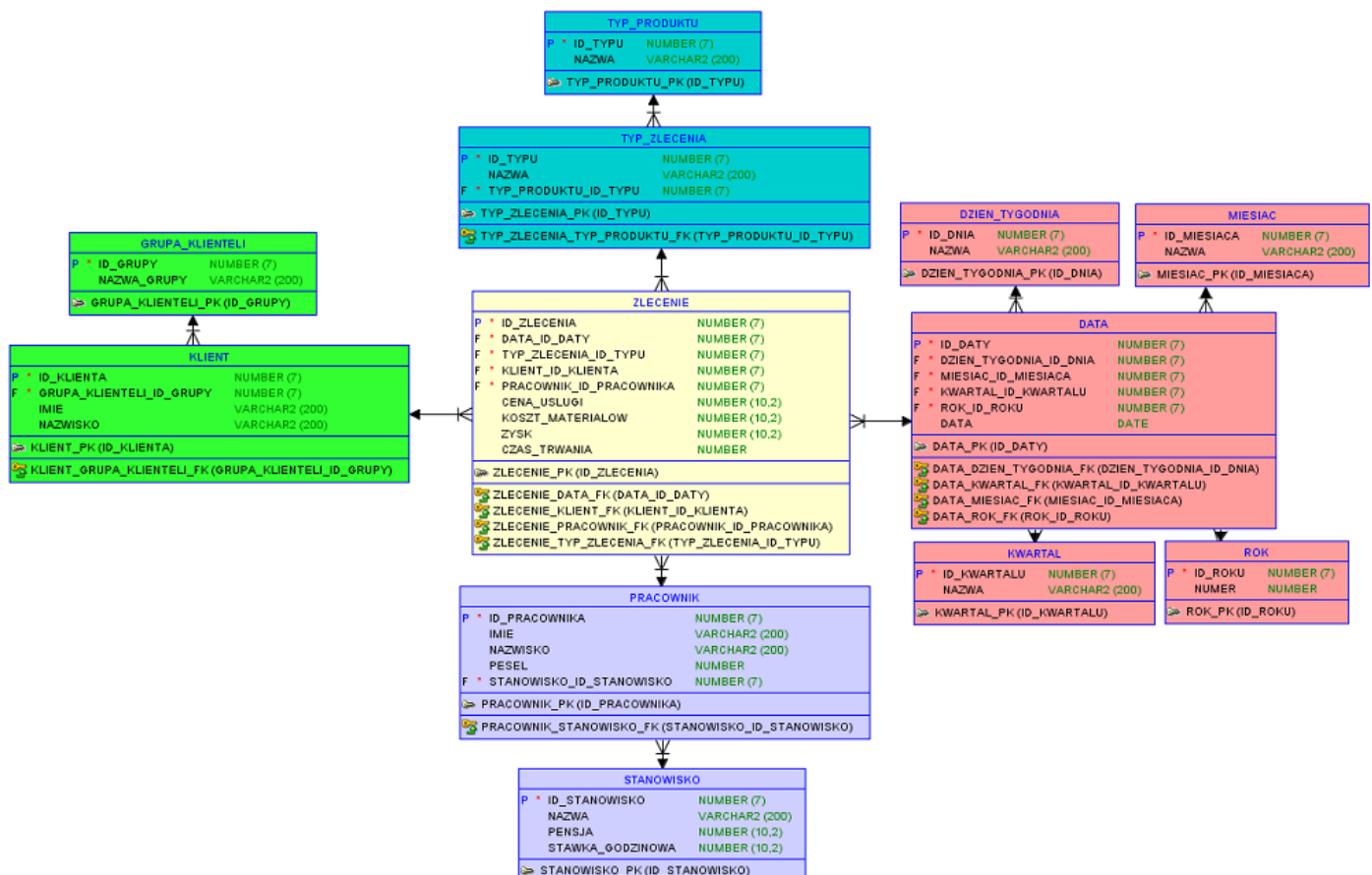
- **ROLLUP**
- **CUBE**
- **GROUPING SETS**

Ponadto napisaliśmy zapytania tworzące:

- **PARTYCJE OBLICZENIOWE**
- **OKNA RUCHOME**
- **RANKINGI**

Hurtownia danych posiadająca opisane powyżej komponenty może usprawnić proces przetwarzania analitycznego, czyli w naszym przypadku może pomóc kierownictwu warsztatu szewsko-kaletniczego w podejmowaniu poprawnych i przyszłościowych decyzji biznesowych w celu rozwoju swojego przedsiębiorstwa.

2. Schemat ERD.



3. Przygotowanie hurtowni do pracy, sposób ładowania danych, dostęp do interfejsu.

Wszystkie pliki tj. pliki ładowania danych lub pliki tworzące hurtownię korzystają z konta hr dostępnego domyślnie w bazie Oracle.

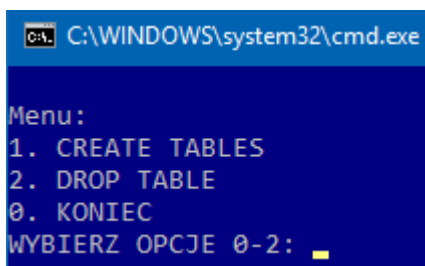
- **LOGIN:** hr
- **HASŁO:** hr

W przypadku posiadania nieodblokowanego konta hr należy uruchomić aplikację „Run SQL Command Line” i wykonać poniższe polecenia:

- połączenie z kontem administratora – system:
`CONNECT system/hasło_do_konta;`
lub
`CONNECT sys AS SYSDBA;`
- nadanie hasła dla konta hr i jego odblokowanie:
`ALTER USER hr IDENTIFIED BY hr ACCOUNT UNLOCK;`
- połączenie się z kontem hr:
`CONNECT hr/hr;`

W przypadku niepowodzenia procedurę należy powtórzyć.

Przed rozpoczęciem procesu ładowania danych do hurtowni należy stworzyć tabele. Aby to uczynić należy posłużyć się klientem napisanym w SQL * Plus. Znajduje się on w folderze „Tworzenie hurtowni”. Należy uruchomić plik „create&drop.bat”. Po uruchomieniu pojawi się okno:



Należy wybrać opcję numer 1, aby stworzyć wymagane tabele. Po zakończeniu pracy z naszą hurtownią danych można usunąć wszystkie tabele korzystając z opcji numer 2. Opcja numer 0 odpowiada za wyjście z klienta.

Ładowanie danych do naszej hurtowni realizujemy korzystając z programu SQL Loader. W celu wygodnego ładowania danych stworzyliśmy plik „load_data.bat” zawierający wszystkie komendy umożliwiające załadowanie danych. Znajduje się on w folderze „Ładowanie danych”. Koniecznym jest, aby znajdował się on bezpośrednio w folderze z plikami .csv przechowującymi dane do ładowania. Komendy znajdujące się w pliku load_data.bat:

```
sqlldr hr/hr control=stanowiskoctl  
sqlldr hr/hr control=pracownikctl  
sqlldr hr/hr control=grupa_klientelictl  
sqlldr hr/hr control=klientctl  
sqlldr hr/hr control=typ_produktyctl  
sqlldr hr/hr control=typ_zleczeniactl  
sqlldr hr/hr control=dzienctl  
sqlldr hr/hr control=miesiacctl  
sqlldr hr/hr control=kwartalctl  
sqlldr hr/hr control=rokctl  
sqlldr hr/hr control=datactl  
sqlldr hr/hr control=zleczeniectl
```

Komendy przytoczone powyżej zawierają login oraz hasło do konta hr oraz ścieżkę domyślną dla plików kontrolnych .ctl. Po uruchomieniu programu pojawi się okno i proces ładowania danych rozpocznie się automatycznie.

```
WybierzC:\WINDOWS\system32\cmd.exe

Commit point reached - logical record count 3

C:\Users\Bartłomiej Osak\Disk Google\Studia\INFORMATYKA SEMESTR IV\INFORMATYKA SEMESTR IV\Bazy danych 2\Projekt\ładowanie danych>sqlldr hr/hr control=klent.ctl

SQL*Loader: Release 11.2.0.2.0 - Production on So Cze 10 18:44:57 2017

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.

Commit point reached - logical record count 64
Commit point reached - logical record count 128
Commit point reached - logical record count 192
Commit point reached - logical record count 256
Commit point reached - logical record count 320
Commit point reached - logical record count 384
Commit point reached - logical record count 448
Commit point reached - logical record count 512
Commit point reached - logical record count 576
Commit point reached - logical record count 640
Commit point reached - logical record count 704
Commit point reached - logical record count 768
Commit point reached - logical record count 832
Commit point reached - logical record count 896
Commit point reached - logical record count 960
Commit point reached - logical record count 1024
Commit point reached - logical record count 1088
Commit point reached - logical record count 1152
Commit point reached - logical record count 1216
Commit point reached - logical record count 1280
Commit point reached - logical record count 1344
Commit point reached - logical record count 1408
Commit point reached - logical record count 1472
Commit point reached - logical record count 1536
Commit point reached - logical record count 1600
Commit point reached - logical record count 1664
Commit point reached - logical record count 1728
Commit point reached - logical record count 1792
Commit point reached - logical record count 1856
Commit point reached - logical record count 1920
Commit point reached - logical record count 1984
Commit point reached - logical record count 2048
Commit point reached - logical record count 2112
Commit point reached - logical record count 2176
Commit point reached - logical record count 2240
Commit point reached - logical record count 2304
Commit point reached - logical record count 2368
Commit point reached - logical record count 2432
Commit point reached - logical record count 2496
```

Po zakończeniu ładowania danych należy potwierdzić operację klawiszem ENTER. Po każdym uruchomieniu programu automatycznie tworzą się pliki .log zawierające dane o procesie ładowania. W przypadku niepowodzenia załadowania danych tworzą się również pliki o rozszerzeniu .bad. Oba wyżej wymienione pliki mają taką samą nazwę jak plik sterujący.

Przykładowy plik .log:

```
SQL*Loader: Release 11.2.0.2.0 - Production on So Cze 10 18:44:57 2017
Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.
Control File:   zlecenie.ctl
Data File:     zlecenie.csv
  Bad File:     zlecenie.bad
  Discard File: none specified
(Allow all discards)
Number to load: ALL
Number to skip: 0
Errors allowed: 50
Bind array:    64 rows, maximum of 256000 bytes
Continuation:  none specified
Path used:     Conventional
```

Table ZLECENIE, loaded from every logical record.
Insert option in effect for this table: APPEND

Column Name	Position	Len	Term	Encl	Datatype
-----	-----	-----	-----	-----	-----
ID_ZLECENIA	FIRST	*	,		CHARACTER
DATA_ID_DATY	NEXT	*	,		CHARACTER
TYP_ZLECENIA_ID_TYPU	NEXT	*	,		CHARACTER
KLIENT_ID_KLIENTA	NEXT	*	,		CHARACTER
PRACOWNIK_ID_PRACOWNIKA	NEXT	*	,		CHARACTER
CENA_USLUGI	NEXT	*	,		CHARACTER
SQL string for column : "TO_NUMBER(:CENA_USLUGI,'999999.99')"					
KOSZT_MATERIALOW	NEXT	*	,		CHARACTER
SQL string for column : "TO_NUMBER(:KOSZT_MATERIALOW,'999999.99')"					
ZYSK	NEXT	*	,		CHARACTER
SQL string for column : "TO_NUMBER(:ZYSK,'999999.99')"					
CZAS_TRWANIA	NEXT	*	,		CHARACTER

Table ZLECENIE:
30000 Rows successfully loaded.
0 Rows not loaded due to data errors.
0 Rows not loaded because all WHEN clauses were failed.
0 Rows not loaded because all fields were null.

```

Space allocated for bind array:          148608 bytes(64 rows)
Read   buffer bytes: 1048576

Total logical records skipped:          0
Total logical records read:             30000
Total logical records rejected:         0
Total logical records discarded:        0

Run began on So Cze 10 18:44:57 2017
Run ended on So Cze 10 18:44:58 2017

Elapsed time was:      00:00:00.55
CPU time was:         00:00:00.13

```

Plik sterujący to plik o rozszerzeniu .ctl. **Przykład:**

```

load data
infile 'zlecenie.csv'
append into table ZLECENIE
fields terminated by ','
(ID_ZLECENIA,DATA_ID_DATY,TYP_ZLECENIA,ID_TYPU,KLIENT_ID_KLIENTA,
PRACOWNIK_ID_PRACOWNIKA,CENA_USLUGI,"TO_NUMBER(:CENA_USLUGI,'999999.99')",
KOSZT_MATERIALOW "TO_NUMBER(:KOSZT_MATERIALOW,'999999.99')",
ZYSK "TO_NUMBER(:ZYSK,'999999.99')",CZAS_TRWANIA)

```

Plik ten zawiera wskazanie na plik csv zawierający dane do wczytania. Ponadto w polu **append into table** wskazujemy do jakiej tabeli chcemy załadować dane. Pole **fields terminated by** określa jakim znakiem są rozdzielane dane wewnątrz pliku z pola **infile**. Na końcu podawane są nazwy kolumn, które będą odpowiednio wypełniane danymi pochodzącymi z pliku .csv. W niektórych przypadkach należy zastosować odpowiednie rzutowanie celem poprawnego załadowania danych.

Przed rozpoczęciem korzystania z interfejsu należy przygotować deskryptor dostępu do bazy. Należy uruchomić program „Run SQL Command Line” i wykonać poniższe polecenia:

- połączenie z kontem system:

```

CONNECT system/hasło_do_konta;
lub
CONNECT sys AS sysdba;

```

- przypisanie użytkownikowi hr prawa XDBADMIN:

```
GRANT XDBADMIN TO hr;
```

- napisanie procedury tworzącej deskryptor:

```

BEGIN
DBMS_EPG.CREATE_DAD(dad_name=>'dad',path=>'/plsql/*');
END;
/

```

Następnie należy uruchomić plik „load.bat”, który znajduje się w folderze „Interfejs”. Uruchomienie tego programu spowoduje automatyczne załadowanie procedur. Ładowanie odbywa się poprzez polecenie loadpsp.

Zawartość programu „load.bat”:

```

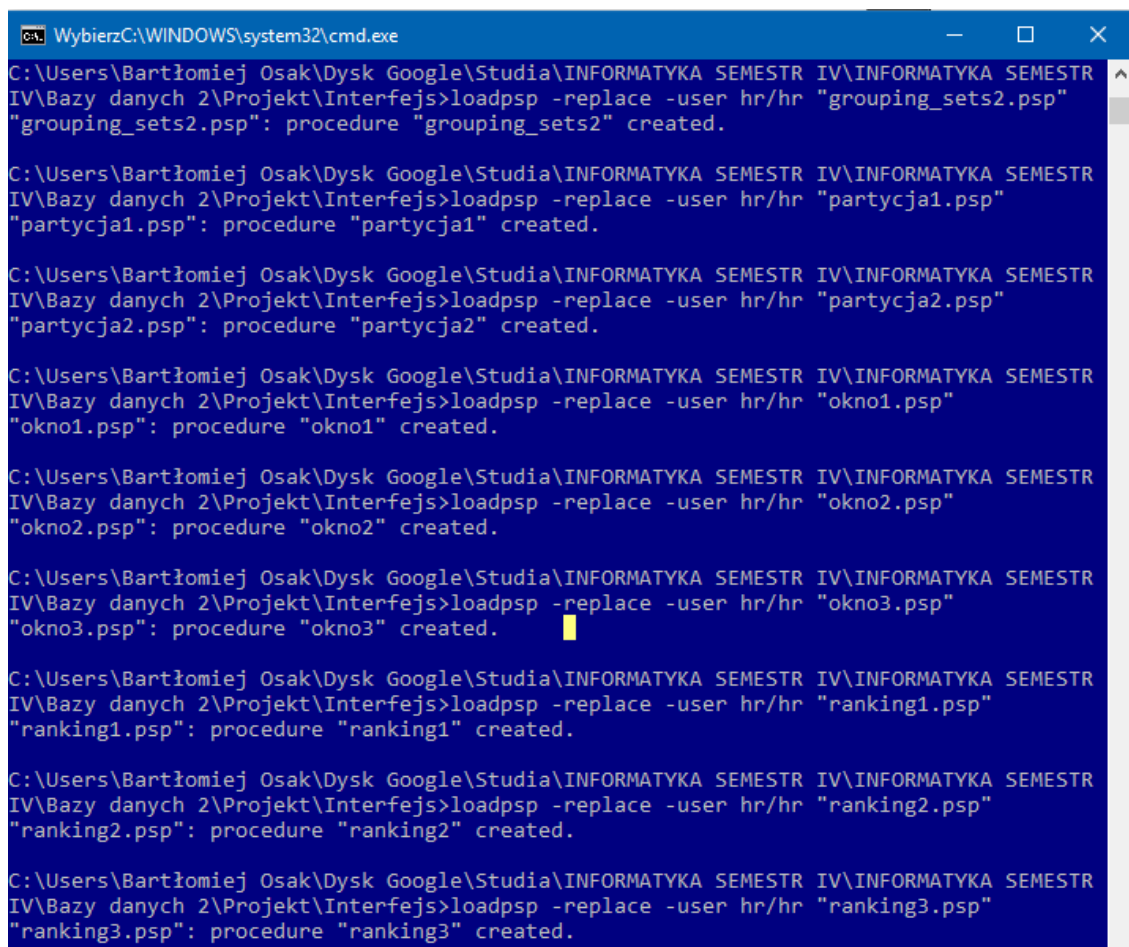
loadpsp -replace -user hr/hr "strona_glowna.psp"
loadpsp -replace -user hr/hr "tabela_zlecenie.psp"
loadpsp -replace -user hr/hr "tabela_typ_produkta.psp"
loadpsp -replace -user hr/hr "tabela_typ_zlecenia.psp"
loadpsp -replace -user hr/hr "tabela_grupa_klienteli.psp"
loadpsp -replace -user hr/hr "tabela_klient.psp"
loadpsp -replace -user hr/hr "tabela_pracownik.psp"
loadpsp -replace -user hr/hr "tabela_stanowisko.psp"
loadpsp -replace -user hr/hr "tabela_data.psp"
loadpsp -replace -user hr/hr "tabela_dzien_tygodnia.psp"
loadpsp -replace -user hr/hr "tabela_miesiac.psp"

```

```
loadpsp -replace -user hr/hr "tabela_kwartal.psp"
loadpsp -replace -user hr/hr "tabela_rok.psp"
loadpsp -replace -user hr/hr "zapytania_rollup_all.psp"
loadpsp -replace -user hr/hr "zapytania_cube_all.psp"
loadpsp -replace -user hr/hr "zapytania_grouping_sets_all.psp"
loadpsp -replace -user hr/hr "zapytania_partycje_all.psp"
loadpsp -replace -user hr/hr "zapytania_okna_all.psp"
loadpsp -replace -user hr/hr "zapytania_rankingi_all.psp"
loadpsp -replace -user hr/hr "rollup1.psp"
loadpsp -replace -user hr/hr "rollup2.psp"
loadpsp -replace -user hr/hr "cube1.psp"
loadpsp -replace -user hr/hr "cube2.psp"
loadpsp -replace -user hr/hr "grouping_sets1.psp"
loadpsp -replace -user hr/hr "grouping_sets2.psp"
loadpsp -replace -user hr/hr "partycja1.psp"
loadpsp -replace -user hr/hr "partycja2.psp"
loadpsp -replace -user hr/hr "okno1.psp"
loadpsp -replace -user hr/hr "okno2.psp"
loadpsp -replace -user hr/hr "okno3.psp"
loadpsp -replace -user hr/hr "ranking1.psp"
loadpsp -replace -user hr/hr "ranking2.psp"
loadpsp -replace -user hr/hr "ranking3.psp"
```

Komenda **loadpsp** umożliwia załadowanie procedur zapisanych w plikach .psp na lokalny serwer. Ponadto komenda **-replace** spowoduje zastąpienie procedury na serwerze. Ponadto wszystkie powyższe komendy zostały napisane dla użytkownika **hr**.

Przykładowy wynik uruchomienia programu:



```
WybierzC:\WINDOWS\system32\cmd.exe
C:\Users\Bartłomiej Osak\Dysk Google\Studia\INFORMATYKA SEMESTR IV\INFORMATYKA SEMESTR IV\Bazy danych 2\Projekt\Interfejs>loadpsp -replace -user hr/hr "grouping_sets2.psp"
"grouping_sets2.psp": procedure "grouping_sets2" created.

C:\Users\Bartłomiej Osak\Dysk Google\Studia\INFORMATYKA SEMESTR IV\INFORMATYKA SEMESTR IV\Bazy danych 2\Projekt\Interfejs>loadpsp -replace -user hr/hr "partycja1.psp"
"partycja1.psp": procedure "partycja1" created.

C:\Users\Bartłomiej Osak\Dysk Google\Studia\INFORMATYKA SEMESTR IV\INFORMATYKA SEMESTR IV\Bazy danych 2\Projekt\Interfejs>loadpsp -replace -user hr/hr "partycja2.psp"
"partycja2.psp": procedure "partycja2" created.

C:\Users\Bartłomiej Osak\Dysk Google\Studia\INFORMATYKA SEMESTR IV\INFORMATYKA SEMESTR IV\Bazy danych 2\Projekt\Interfejs>loadpsp -replace -user hr/hr "okno1.psp"
"okno1.psp": procedure "okno1" created.

C:\Users\Bartłomiej Osak\Dysk Google\Studia\INFORMATYKA SEMESTR IV\INFORMATYKA SEMESTR IV\Bazy danych 2\Projekt\Interfejs>loadpsp -replace -user hr/hr "okno2.psp"
"okno2.psp": procedure "okno2" created.

C:\Users\Bartłomiej Osak\Dysk Google\Studia\INFORMATYKA SEMESTR IV\INFORMATYKA SEMESTR IV\Bazy danych 2\Projekt\Interfejs>loadpsp -replace -user hr/hr "okno3.psp"
"okno3.psp": procedure "okno3" created.

C:\Users\Bartłomiej Osak\Dysk Google\Studia\INFORMATYKA SEMESTR IV\INFORMATYKA SEMESTR IV\Bazy danych 2\Projekt\Interfejs>loadpsp -replace -user hr/hr "ranking1.psp"
"ranking1.psp": procedure "ranking1" created.

C:\Users\Bartłomiej Osak\Dysk Google\Studia\INFORMATYKA SEMESTR IV\INFORMATYKA SEMESTR IV\Bazy danych 2\Projekt\Interfejs>loadpsp -replace -user hr/hr "ranking2.psp"
"ranking2.psp": procedure "ranking2" created.

C:\Users\Bartłomiej Osak\Dysk Google\Studia\INFORMATYKA SEMESTR IV\INFORMATYKA SEMESTR IV\Bazy danych 2\Projekt\Interfejs>loadpsp -replace -user hr/hr "ranking3.psp"
"ranking3.psp": procedure "ranking3" created.
```

Następnie dostęp do głównego interfejsu uruchamiamy w przeglądarce internetowej wpisując następujący adres:

http://127.0.0.1:8080//dad/plsql/strona_glowna

Przy pierwszym logowaniu użytkownik zostanie poproszony o zalogowanie się. Należy wpisać dane użytkownika mającego uprawnienia XDBADMIN, czyli w naszym przypadku: hr/hr.

4. Zapytania analityczne.

Celem analizy hurtowni danych napisaliśmy zapytania analityczne. Zastosowaliśmy operatory grupujące takie jak:

- **ROLLUP**
- **CUBE**
- **GROUPING SETS**

Ponadto napisaliśmy zapytania tworzące:

- **PARTYCJE OBLICZENIOWE**
- **OKNA RUCHOME**
- **RANKINGI**

a) Zapytanie ROLLUP I:

Kod zapytania:

```
SELECT NVL(TO_CHAR(ZLECENIE.TYP_ZLECENIA_ID_TYPU), ' ') AS "TYP ZLECENIA",
NVL(TO_CHAR(ZLECENIE.CZAS_TRWANIA), ' ') AS "CZAS TRWANIA",
TO_CHAR(ROUND(AVG(ZLECENIE.CENA_USLUGI),2)) || 'Zł' AS "CENA USLUGI",
TO_CHAR(ROUND(AVG(ZLECENIE.KOSZT_MATERIALOW),2)) || 'Zł' AS "KOSZT MATERIAŁU",
TO_CHAR(ROUND(AVG(ZLECENIE.ZYSK),2)) || 'Zł' AS "ZYSK"
FROM ZLECENIE
GROUP BY ROLLUP(ZLECENIE.TYP_ZLECENIA_ID_TYPU, ZLECENIE.CZAS_TRWANIA);
```

Opis:

Pierwsze zapytanie ROLLUP zwraca podsumowanie średnich cen za usługę, kosztów materiałów oraz zysków za poszczególne typy zlecenia względem czasu wykonania. Dla każdego typu zlecenia wypisywanie jest podsumowanie częściowe. Na końcu wypisywanie jest podsumowanie całkowite.

Wynik zadziałania – fragment tabeli wynikowej:

WYNIK: ZAPYTANIE ROLLUP				
ID TYPU ZLECENIA	CZAS TRWANIA USŁUGI	CENA ZA USŁUGĘ	KOSZT MATERIAŁU	ZYSK
1	1	89.12Zł	22.77Zł	66.36Zł
1	2	91.82Zł	25.17Zł	66.66Zł
1	3	49.1Zł	17.93Zł	31.17Zł
1	4	59.47Zł	17.31Zł	42.16Zł
1	5	62.27Zł	26.79Zł	35.48Zł
1	6	85.65Zł	22.83Zł	62.81Zł
1	7	74.09Zł	21.77Zł	52.32Zł
1	8	59.69Zł	22.36Zł	37.33Zł
1	9	68.23Zł	24.5Zł	43.73Zł
1	10	80.78Zł	18.89Zł	61.89Zł
1	11	74.64Zł	19.75Zł	54.89Zł
1	12	86.5Zł	24.95Zł	61.54Zł
1	13	82.07Zł	27.93Zł	54.13Zł
1	14	89.57Zł	21.47Zł	68.09Zł
1		75.82Zł	22.53Zł	53.29Zł

b) Zapytanie ROLLUP II:

Kod zapytania:

```
SELECT NVL(TO_CHAR(ZLECENIE.PRACOWNIK_ID_PRACOWNIKA), ' ') AS "ID PRACOWNIKA",
NVL(TO_CHAR(ZLECENIE.TYP_ZLECENIA_ID_TYPU), ' ') AS "TYP ZLECENIA",
NVL(TO_CHAR(ZLECENIE.CZAS_TRWANIA), ' ') AS "CZAS TRWANIA",
ROUND(COUNT(ZLECENIE.ID_ZLECENIA),0) AS "ILOSC ZLECEN",
TO_CHAR(ROUND(SUM(ZLECENIE.KOSZT_MATERIALOW),2)) || 'Zł' AS "KOSZT MATERIAŁOW",
TO_CHAR(ROUND(SUM(ZLECENIE.ZYSK),2)) || 'Zł' AS "ZYSK"
FROM ZLECENIE
GROUP BY
ROLLUP(ZLECENIE.PRACOWNIK_ID_PRACOWNIKA, ZLECENIE.TYP_ZLECENIA_ID_TYPU, ZLECENIE.CZAS_TRWANIA);
```

Opis:

Drugie zapytanie ROLLUP zwraca podsumowanie, ile dany pracownik dla danego typu zlecenia oraz dla danego czasu trwania przyjął zleceń oraz ile wyniósł koszt materiałów oraz zysk od liczby przyjętych zleceń. Ponadto wypisywanie jest podsumowanie częściowe dla każdego pracownika oraz podsumowanie całkowite.

Wynik zadziałania – fragment tabeli wynikowej:

WYNIK: ZAPYTANIE ROLLUP					
ID PRACOWNIKA	ID TYPU ZLECENIA	CZAS TRWANIA USŁUGI	IŁOŚĆ ZLECEŃ	KOSZT MATERIAŁÓW	ZYSK
1	1	1	1	39.19ZŁ	19.54ZŁ
1	1	2	1	25.29ZŁ	38.5ZŁ
1	1	3	1	14.98ZŁ	7.56ZŁ
1	1	4	1	16.11ZŁ	16.65ZŁ
1	1	6	2	70.57ZŁ	259.68ZŁ
1	1	7	2	32.3ZŁ	98.8ZŁ
1	1	8	1	19.64ZŁ	27.64ZŁ
1	1	9	1	23.77ZŁ	8.65ZŁ
1	1	10	1	2.03ZŁ	367.97ZŁ
1	1	11	2	32.55ZŁ	90.15ZŁ
1	1	12	1	16.32ZŁ	22.72ZŁ
1	1		14	292.75ZŁ	957.86ZŁ

c) Zapytanie CUBE I:**Kod zapytania:**

```
SELECT NVL(TO_CHAR(ZLECENIE.PRACOWNIK_ID_PRACOWNIKA), ' ') AS "ID PRACOWNIKA",
NVL(TO_CHAR(ZLECENIE.TYP_ZLECENIA_ID_TYPU), ' ') AS "TYP ZLECENIA",
NVL(TO_CHAR(ZLECENIE.CZAS_TRWANIA), ' ') AS "CZAS TRWANIA",
ROUND(COUNT(ZLECENIE.KLIENT_ID_KLIENTA),0) AS "ILOSC KLIENTOW",
TO_CHAR(ROUND(AVG(ZLECENIE.ZYSK),2))|| ' ZŁ' AS "SREDNI ZYSK",
TO_CHAR(ROUND(SUM(ZLECENIE.ZYSK),2))|| ' ZŁ' AS "SUMA ZYSKU"
FROM ZLECENIE
GROUP BY CUBE(ZLECENIE.PRACOWNIK_ID_PRACOWNIKA, ZLECENIE.TYP_ZLECENIA_ID_TYPU,
ZLECENIE.CZAS_TRWANIA);
```

Opis:

Pierwsze zapytanie CUBE zwraca, ile dany pracownik dla danego typu zlecenia oraz dla danego czasu trwania zlecenia obsłużył klientów oraz ile wyniósł średni zysk z danego typu zlecenia o danym czasie trwania oraz sumę zysków dla danego typu zlecenia o danym czasie trwania dla danej liczby klientów. Ponadto wypisywane są podsumowania częściowe dla pracowników, typów zlecenia i czasów trwania wykonywania zleceń. Wypisywanie jest również całkowite.

Wynik zadziałania – fragment tabeli wynikowej:

WYNIK: ZAPYTANIE CUBE					
ID PRACOWNIKA	ID TYPU ZLECENIA	CZAS TRWANIA USŁUGI	IŁOŚĆ KLIENTÓW	ŚREDNI ZYSK	SUMA ZYSKÓW
			30000	54.49 ZŁ	1634729.65 ZŁ
		1	2183	52.39 ZŁ	114373.62 ZŁ
		2	2160	54.1 ZŁ	116863.94 ZŁ
		3	2157	56.23 ZŁ	121292.89 ZŁ
		4	2146	54.64 ZŁ	117258.13 ZŁ
		5	2122	53.81 ZŁ	114179.25 ZŁ
		6	2127	54.54 ZŁ	115999.01 ZŁ
		7	2154	55.87 ZŁ	120348.21 ZŁ
		8	2129	54.61 ZŁ	116259.89 ZŁ
		9	2070	54.03 ZŁ	111851.76 ZŁ
		10	2145	56.47 ZŁ	121138.48 ZŁ
		11	2110	53.45 ZŁ	112776.52 ZŁ
		12	2148	53.56 ZŁ	115057.38 ZŁ
		13	2183	53.8 ZŁ	117445.27 ZŁ
		14	2166	55.35 ZŁ	119885.3 ZŁ
	1		282	53.29 ZŁ	15027.29 ZŁ
	1	1	22	66.36 ZŁ	1459.81 ZŁ
	1	2	11	66.66 ZŁ	733.22 ZŁ
	1	3	10	31.17 ZŁ	311.74 ZŁ
	1	4	19	42.16 ZŁ	801.01 ZŁ

d) Zapytanie CUBE II:

Kod zapytania:

```
SELECT NVL(TO_CHAR(ZLECENIE.KLIENT_ID_KLIENTA), ' ') AS "ID KLIENTA",
NVL(TO_CHAR(ZLECENIE.TYP_ZLECENIA_ID_TYPU), ' ') AS "TYP ZLECENIA",
ROUND(COUNT(ZLECENIE.ID_ZLECENIA),0) AS "ILOSC ZLECEN",
TO_CHAR(ROUND(SUM(ZLECENIE.ZYSK),0)) || ' Zł' AS "ZYSK"
FROM ZLECENIE
GROUP BY CUBE(ZLECENIE.KLIENT_ID_KLIENTA, ZLECENIE.TYP_ZLECENIA_ID_TYPU);
```

Opis:

Drugie zapytanie CUBE zwraca ilość zleceń zleconych przez danego klienta z podziałem na typ zlecenia oraz zysk z tych zleceń. Ponadto wypisywane są podsumowania częściowe dla typów zleceń oraz podsumowanie całkowite.

Wynik zadziałania – fragment tabeli wynikowej:

WYNIK: ZAPYTANIE CUBE			
ID KLIENTA	ID TYPU ZLECENIA	ILOŚĆ ZLECEŃ	ZYSK
		30000	1634730 Zł
	1	282	15027 Zł
	2	305	18134 Zł
	3	305	15874 Zł
	4	311	18434 Zł
	5	270	15589 Zł
	6	258	13400 Zł
	7	313	16758 Zł
	8	321	14767 Zł
	9	337	18080 Zł
	10	291	17002 Zł
	11	306	17923 Zł
	12	299	15062 Zł

e) Zapytanie GROUPING SETS I:

Kod zapytania:

```
SELECT NVL(TO_CHAR(ZLECENIE.KLIENT_ID_KLIENTA), ' ') AS "ID KLIENTA"
NVL(TO_CHAR(ZLECENIE.CZAS_TRWANIA), ' ') AS "CZAS TRWANIA",
NVL(TO_CHAR(ZLECENIE.TYP_ZLECENIA_ID_TYPU), ' ') AS "TYP USLUGI",
NVL(TO_CHAR(ZLECENIE.PRACOWNIK_ID_PRACOWNIKA), ' ') AS "ID PRACOWNIKA",
TO_CHAR(SUM(ZLECENIE.CENA_USLUGI)) || ' Zł' AS "CENA USLUGI", TO_CHAR(SUM(ZLECENIE.ZYSK)) || ' Zł'
AS "ZYSK"
FROM ZLECENIE
GROUP BY GROUPING SETS(ZLECENIE.CZAS_TRWANIA, ZLECENIE.TYP_ZLECENIA_ID_TYPU,
ZLECENIE.PRACOWNIK_ID_PRACOWNIKA, (ZLECENIE.KLIENT_ID_KLIENTA, ZLECENIE.TYP_ZLECENIA_ID_TYPU))
ORDER BY ZLECENIE.KLIENT_ID_KLIENTA, ZLECENIE.CZAS_TRWANIA, ZLECENIE.TYP_ZLECENIA_ID_TYPU,
ZLECENIE.PRACOWNIK_ID_PRACOWNIKA;
```

Opis:

Pierwsze zapytanie GROUPING SETS zwraca jaki typ usługi zlecił dany klient wraz z ceną za tą usługę. Ponadto zapytanie zwraca sumę cen usług oraz zysk dla danego czasu trwania usługi. Zwraca również sumę cen usług oraz zysk dla danego typu zlecenia. Na końcu zapytanie zwróciło sumę cen usług oraz zysk dla danego pracownika z przyjętych przez niego zleceń.

Wynik zadziałania – fragment tabeli wynikowej:

WYNIK: ZAPYTANIE GROUPING SETS					
ID KLIENTA	CZAS TRWANIA USŁUGI	ID TYPU USŁUGI	ID PRACOWNIKA	CENA USŁUGI	ZYSK
1		12		49.15ZŁ	5.47ZŁ
1		38		56.46ZŁ	31.54ZŁ
1		39		45.99ZŁ	25.01ZŁ
1		45		55.31ZŁ	45.81ZŁ
1		67		82.28ZŁ	80.62ZŁ
2		14		79.51ZŁ	76.38ZŁ
2		32		42.32ZŁ	9.28ZŁ
2		50		85.12ZŁ	82.38ZŁ
2		52		79.36ZŁ	52.15ZŁ
2		68		63.83ZŁ	15.44ZŁ
2		90		85.2ZŁ	64.54ZŁ
3		11		67.91ZŁ	48.83ZŁ
3		15		286ZŁ	269.44ZŁ

f) Zapytanie GROUPING SETS II:

Kod zapytania:

```
SELECT NVL(TO_CHAR(ZLECENIE.CZAS_TRWANIA), ' ') AS "CZAS TRWANIA",
NVL(TO_CHAR(ZLECENIE.TYP_ZLECENIA_ID_TYPU), ' ') AS "TYP USŁUGI",
NVL(TO_CHAR(ZLECENIE.PRACOWNIK_ID_PRACOWNIKA), ' ') AS "ID PRACOWNIKA",
TO_CHAR(SUM(ZLECENIE.ZYSK)) || 'ZŁ' AS "ZYSK"
FROM ZLECENIE
GROUP BY GROUPING SETS (ZLECENIE.CZAS_TRWANIA, (ZLECENIE.CZAS_TRWANIA,
ZLECENIE.TYP_ZLECENIA_ID_TYPU), ZLECENIE.PRACOWNIK_ID_PRACOWNIKA)
ORDER BY ZLECENIE.CZAS_TRWANIA, ZLECENIE.TYP_ZLECENIA_ID_TYPU, ZLECENIE.PRACOWNIK_ID_PRACOWNIKA;
```

Opis:

Drugie zapytanie GROUPING SETS zwraca sumę zysków dla danego czasu trwania usługi względem danego typu usługi. Na końcu zapytanie zwróciło sumę cen usług oraz zysk dla danego pracownika z przyjętych przez niego zleceń.

Wynik zadziałania – fragment tabeli wynikowej:

WYNIK: ZAPYTANIE GROUPING SETS			
CZAS TRWANIA USŁUGI	ID TYPU USŁUGI	ID PRACOWNIKA	ZYSK
1	1		1459.81ZŁ
1	2		984.76ZŁ
1	3		1558.99ZŁ
1	4		723.92ZŁ
1	5		612.1ZŁ
1	6		580.79ZŁ
1	7		935.43ZŁ
1	8		1179.52ZŁ
1	9		709.21ZŁ
1	10		1007.45ZŁ
1	11		768.91ZŁ
1	12		509.17ZŁ
1	13		788.9ZŁ
1	14		1828.36ZŁ

g) Zapytanie OKNO RUCHOME I:

Kod zapytania:

```
SELECT ROK.NUMER AS "ROK", MIESIAC.ID_MIESIACA AS "MIESIAC", TO_CHAR(SUM(ZLECENIE.ZYSK))||'ZŁ'
AS "ZYSK",
TO_CHAR(NVL(TO_CHAR(LAG(SUM(ZLECENIE.ZYSK),1) OVER (ORDER BY ROK.NUMER,
MIESIAC.ID_MIESIACA)), '---'))||'ZŁ' AS "LAG",
TO_CHAR(NVL(TO_CHAR(SUM(ZLECENIE.ZYSK) - (LAG(SUM(ZLECENIE.ZYSK),1) OVER (ORDER BY ROK.NUMER,
MIESIAC.ID_MIESIACA))), '---'))||'ZŁ' AS "WZROST/SPADEK"
FROM ROK, MIESIAC, ZLECENIE, DATA
WHERE ROK.ID_ROKU = DATA.ROK_ID_ROKU
AND MIESIAC.ID_MIESIACA = DATA.MIESIAC_ID_MIESIACA
AND DATA.ID_DATY = ZLECENIE.DATA_ID_DATY
AND ROK.NUMER = 2015
GROUP BY ROK.NUMER, MIESIAC.ID_MIESIACA;
```

Opis:

Pierwsze okno ruchome prezentuje zysk dla danego miesiąca w roku wybranym przez użytkownika w formularzu. Zapytanie pobiera również wartość zysku z poprzedniego miesiąca za pomocą funkcji LAG() oraz w następnej kolumnie obliczany jest wzrost lub spadek względem poprzedniego miesiąca.

Wynik zadziałania – fragment tabeli wynikowej:

WYNIK: OKNO RUCHOME				
OKNO RUCHOME I				
Zapytanie wywołano z parametrem:				
Rok: 2015				
ROK	MIESIĄC	ZYSK	ZYSK Z POPRZEDNIEGO MIESIĄCA	WZROST/SPADEK WZGLĘDEM POPRZEDNIEGO MIESIĄCA
2015	1	28166.23ZŁ	--ZŁ	--ZŁ
2015	2	25030.76ZŁ	28166.23ZŁ	-3135.47ZŁ
2015	3	23809.43ZŁ	25030.76ZŁ	-1221.33ZŁ
2015	4	20657.96ZŁ	23809.43ZŁ	-3151.47ZŁ
2015	5	27155.91ZŁ	20657.96ZŁ	6497.95ZŁ
2015	6	19528.85ZŁ	27155.91ZŁ	-7627.06ZŁ
2015	7	19294.55ZŁ	19528.85ZŁ	-234.3ZŁ
2015	8	23023.38ZŁ	19294.55ZŁ	3728.83ZŁ
2015	9	28561.38ZŁ	23023.38ZŁ	5538ZŁ
2015	10	21438.07ZŁ	28561.38ZŁ	-7123.31ZŁ
2015	11	22155.83ZŁ	21438.07ZŁ	717.76ZŁ
2015	12	24544.86ZŁ	22155.83ZŁ	2389.03ZŁ

h) Zapytanie OKNO RUCHOME II:

Kod zapytania:

```
SELECT ROK.NUMER AS "ROK", MIESIAC.NAZWA AS "MIESIAC",
TO_CHAR(SUM(ZLECENIE.ZYSK))||'ZŁ' AS "ZYSK",
TO_CHAR(SUM(SUM(ZLECENIE.ZYSK)) OVER (ORDER BY MIESIAC.ID_MIESIACA ROWS BETWEEN UNBOUNDED
PRECEDING AND CURRENT ROW))||'ZŁ' AS "SUMA KUMULACYJNA",
TO_CHAR(FIRST_VALUE(SUM(ZLECENIE.ZYSK)) OVER (ORDER BY MIESIAC.ID_MIESIACA ROWS BETWEEN 1
PRECEDING AND 1 FOLLOWING))||'ZŁ' AS "POPRZEDNI MIESIAC",
TO_CHAR(LAST_VALUE(SUM(ZLECENIE.ZYSK)) OVER (ORDER BY MIESIAC.ID_MIESIACA ROWS BETWEEN 1
PRECEDING AND 1 FOLLOWING))||'ZŁ' AS "KOLEJNY MIESIAC"
FROM ZLECENIE, MIESIAC, DATA, ROK
WHERE MIESIAC.ID_MIESIACA = DATA.MIESIAC_ID_MIESIACA
AND ROK.ID_ROKU = DATA.ROK_ID_ROKU
AND DATA.ID_DATY = ZLECENIE.DATA_ID_DATY
AND ROK.NUMER = 2015
GROUP BY ROK.ID_ROKU, MIESIAC.ID_MIESIACA, ROK.NUMER, MIESIAC.NAZWA
ORDER BY ROK.NUMER, MIESIAC.ID_MIESIACA;
```

Opis:

Drugie okno ruchome prezentuje zysk dla danego miesiąca w roku wybranym przez użytkownika w formularzu. Zapytanie oblicza sumę kumulacyjną sumując zyski z poszczególnych miesięcy umieszczając ją w kolejnych wierszach tabeli. Ponadto dla każdego z miesięcy, tam gdzie jest to możliwe pobierany jest zysk z poprzedniego oraz z kolejnego miesiąca za pomocą funkcji FIRST_VALUE() oraz LAST_VALUE().

Wynik zadziałania – fragment tabeli wynikowej:

WYNIK: OKNO RUCHOME					
OKNO RUCHOME II					
Zapytanie wywołano z parametrem:					
Rok: 2015					
ROK	MIESIĄC	ZYSK	SUMA KUMULACYJNA	ZYSK Z POPRZEDNIEGO MIESIĄCA	ZYSK ZA KOLEJNY MIESIĄC
2015	styczen	28166.23ZŁ	28166.23ZŁ	28166.23ZŁ	25030.76ZŁ
2015	luty	25030.76ZŁ	53196.99ZŁ	28166.23ZŁ	23809.43ZŁ
2015	marzec	23809.43ZŁ	77006.42ZŁ	25030.76ZŁ	20657.96ZŁ
2015	kwiecień	20657.96ZŁ	97664.38ZŁ	23809.43ZŁ	27155.91ZŁ
2015	maj	27155.91ZŁ	124820.29ZŁ	20657.96ZŁ	19528.85ZŁ
2015	czerwiec	19528.85ZŁ	144349.14ZŁ	27155.91ZŁ	19294.55ZŁ
2015	lipiec	19294.55ZŁ	163643.69ZŁ	19528.85ZŁ	23023.38ZŁ
2015	sierpień	23023.38ZŁ	186667.07ZŁ	19294.55ZŁ	28561.38ZŁ
2015	wrzesień	28561.38ZŁ	215228.45ZŁ	23023.38ZŁ	21438.07ZŁ
2015	październik	21438.07ZŁ	236666.52ZŁ	28561.38ZŁ	22155.83ZŁ
2015	listopad	22155.83ZŁ	258822.35ZŁ	21438.07ZŁ	24544.86ZŁ
2015	grudzien	24544.86ZŁ	283367.21ZŁ	22155.83ZŁ	24544.86ZŁ

i) Zapytanie OKNO RUCHOME III:

Kod zapytania:

```
SELECT ROK.NUMER AS "ROK", KWARTAL.ID_KWARTALU AS "KWARTAL", PRACOWNIK.ID_PRACOWNIKA AS "ID",  
COUNT(ZLECENIE.ID_ZLECENIA) AS "ILOSC OBSLUZONYCH KLIENTOW",  
SUM(COUNT(ZLECENIE.ID_ZLECENIA)) OVER (PARTITION BY KWARTAL.ID_KWARTALU, ROK.NUMER ORDER BY  
KWARTAL.ID_KWARTALU ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS "SUMA ZLECEN",  
SUM(COUNT(ZLECENIE.ID_ZLECENIA)) OVER (PARTITION BY ROK.NUMER ORDER BY ROK.NUMER ROWS BETWEEN  
UNBOUNDED PRECEDING AND CURRENT ROW) AS "SUMA ZLECEN1",  
TO_CHAR(SUM(ZLECENIE.ZYSK))||'ZŁ' AS "ZYSK",  
TO_CHAR(SUM(SUM(ZLECENIE.ZYSK)) OVER (PARTITION BY KWARTAL.ID_KWARTALU, ROK.NUMER ORDER BY  
KWARTAL.ID_KWARTALU ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW))||'ZŁ' AS "SUMA  
KUMULACYJNA",  
TO_CHAR(SUM(SUM(ZLECENIE.ZYSK)) OVER (PARTITION BY ROK.NUMER ORDER BY ROK.NUMER ROWS BETWEEN  
UNBOUNDED PRECEDING AND CURRENT ROW))||'ZŁ' AS "SUMA KUMULACYJNA1"  
FROM ZLECENIE, ROK, DATA, KWARTAL, PRACOWNIK  
WHERE ROK.ID_ROKU = DATA.ROK_ID_ROKU  
AND KWARTAL.ID_KWARTALU = DATA.KWARTAL_ID_KWARTALU  
AND DATA.ID_DATY = ZLECENIE.DATA_ID_DATY  
AND PRACOWNIK.ID_PRACOWNIKA = ZLECENIE.PRACOWNIK_ID_PRACOWNIKA  
AND ROK.NUMER BETWEEN 2012 AND 2016  
AND KWARTAL.ID_KWARTALU BETWEEN 1 AND 3  
GROUP BY ROK.NUMER, KWARTAL.ID_KWARTALU, PRACOWNIK.ID_PRACOWNIKA  
ORDER BY ROK.NUMER, KWARTAL.ID_KWARTALU;
```

Opis:

Trzecie okno ruchome prezentuje ilość obsłużonych klientów wraz z zyskiem związanym z ich obsłużeniem przez danego pracownika w kwartale danego roku - przedział kwartalny oraz roczny jest wybierany przez użytkownika w formularzu. Ponadto obliczana jest suma kumulacyjna ilości zleceń za dany kwartał, suma kumulacyjna ilości zleceń za dany rok, suma kumulacyjna zysku za dany kwartał oraz suma kumulacyjna zysku za dany rok.

Wynik zadziałania – fragment tabeli wynikowej:

WYNIK: OKNO RUCHOME								
OKNO RUCHOME III Zapytanie wywołano z parametrem: Przedział lat: 2012-2016 Przedział kwartalny: 1-3								
ROK	KWARTAL	ID PRACOWNIKA	IŁOŚĆ OBSŁUŻONYCH KLIENTÓW	SUMA KUMULACYJNA IŁOŚCI ZLECEŃ ZA KWARTAŁ	SUMA KUMULACYJNA IŁOŚCI ZLECEŃ ZA ROK	ZYSK	SUMA KUMULACYJNA ZYSKU ZA KWARTAŁ	SUMA KUMULACYJNA ZYSKU ZA ROK
2012	1	1	74	74	74	4216.97ZŁ	4216.97ZŁ	4216.97ZŁ
2012	1	2	84	158	158	5370.29ZŁ	9587.26ZŁ	9587.26ZŁ
2012	1	3	81	239	239	3424.7ZŁ	13011.96ZŁ	13011.96ZŁ
2012	1	4	81	320	320	5229.42ZŁ	18241.38ZŁ	18241.38ZŁ
2012	1	5	78	398	398	3635.91ZŁ	21877.29ZŁ	21877.29ZŁ
2012	1	6	91	489	489	5446.92ZŁ	27324.21ZŁ	27324.21ZŁ
2012	1	7	86	575	575	4923.6ZŁ	32247.81ZŁ	32247.81ZŁ
2012	1	8	82	657	657	4925.13ZŁ	37172.94ZŁ	37172.94ZŁ
2012	1	9	73	730	730	3603.87ZŁ	40776.81ZŁ	40776.81ZŁ
2012	1	10	76	806	806	3888.95ZŁ	44665.76ZŁ	44665.76ZŁ
2012	1	11	74	880	880	3952.39ZŁ	48618.15ZŁ	48618.15ZŁ
2012	1	12	96	976	976	5444.12ZŁ	54062.27ZŁ	54062.27ZŁ
2012	1	13	86	1062	1062	4326.64ZŁ	58388.91ZŁ	58388.91ZŁ
2012	1	14	90	1152	1152	4460.07ZŁ	62848.98ZŁ	62848.98ZŁ
2012	1	15	87	1239	1239	3843.2ZŁ	66692.18ZŁ	66692.18ZŁ
2012	2	1	83	83	1322	4507.05ZŁ	4507.05ZŁ	71199.23ZŁ

j) Zapytanie PARTYCJA OBLICZENIOWA I:

Kod zapytania:

```

SELECT DISTINCT ZLECENIE.TYP_ZLECENIA_ID_TYPU AS "TYP ZLECENIA", PRACOWNIK.ID_PRACOWNIKA AS
"ID PRACOWNIKA",
TO_CHAR(ROUND(SUM(ZLECENIE.ZYSK) OVER (PARTITION BY ZLECENIE.TYP_ZLECENIA_ID_TYPU,
PRACOWNIK.ID_PRACOWNIKA),2))||'ZŁ' AS "ZYSK",
TO_CHAR(ROUND((100*SUM(ZLECENIE.ZYSK) OVER (PARTITION BY ZLECENIE.TYP_ZLECENIA_ID_TYPU,
PRACOWNIK.ID_PRACOWNIKA)/SUM(ZLECENIE.ZYSK) OVER (PARTITION BY
ZLECENIE.TYP_ZLECENIA_ID_TYPU)),2)||'%' AS "UDZIAŁ % W ZYSKU DLA TYPU",
TO_CHAR(ROUND(SUM(ZLECENIE.CENA_USLUGI) OVER (PARTITION BY
ZLECENIE.TYP_ZLECENIA_ID_TYPU,PRACOWNIK.ID_PRACOWNIKA),2))||'ZŁ' AS "CENA USLUGI",
TO_CHAR(ROUND((100*SUM(ZLECENIE.CENA_USLUGI) OVER (PARTITION BY ZLECENIE.TYP_ZLECENIA_ID_TYPU,
PRACOWNIK.ID_PRACOWNIKA)/SUM(ZLECENIE.CENA_USLUGI) OVER (PARTITION BY
ZLECENIE.TYP_ZLECENIA_ID_TYPU)),2)||'%' AS "UDZIAŁ % W CENIE DLA TYPU",
TO_CHAR(ROUND(SUM(ZLECENIE.KOSZT_MATERIALOW) OVER (PARTITION BY
ZLECENIE.TYP_ZLECENIA_ID_TYPU,PRACOWNIK.ID_PRACOWNIKA),2))||'ZŁ' AS "KOSZT MATERIALU",
TO_CHAR(ROUND((100*SUM(ZLECENIE.KOSZT_MATERIALOW) OVER (PARTITION BY
ZLECENIE.TYP_ZLECENIA_ID_TYPU, PRACOWNIK.ID_PRACOWNIKA)/SUM(ZLECENIE.KOSZT_MATERIALOW) OVER
(PARTITION BY ZLECENIE.TYP_ZLECENIA_ID_TYPU)),2)||'%' AS "UDZIAŁ % W MATERIALE DLA TYPU"
FROM ZLECENIE,PRACOWNIK, DATA, MIESIAC, ROK
WHERE PRACOWNIK.ID_PRACOWNIKA = ZLECENIE.PRACOWNIK_ID_PRACOWNIKA
AND MIESIAC.ID_MIESIACA = DATA.MIESIAC_ID_MIESIACA
AND ROK.ID_ROKU = DATA.ROK_ID_ROKU
AND DATA.ID_DATY = ZLECENIE.DATA_ID_DATY
AND MIESIAC.ID_MIESIACA BETWEEN 1 AND 9
AND ROK.NUMER BETWEEN 2011 AND 2015
ORDER BY ZLECENIE.TYP_ZLECENIA_ID_TYPU, PRACOWNIK.ID_PRACOWNIKA;

```

Opis:

Pierwsza partycja oblicza udział procentowy zysku, ceny usługi oraz kosztów materiału dla danego typu zlecenia, które zostało wykonane przez danego pracownika. Partycja obliczeniowa ograniczona jest czasowo - przedział miesięczny oraz roczny użytkownik wybiera poprzez formularz.

Wynik zadziałania – fragment tabeli wynikowej:

WYNIK: PARTYCJA OBLICZENIOWA							
PARTYCJA OBLICZENIOWA I Zapytanie wywołano z parametrami: Przedział lat: 2011 - 2015 Przedział miesięczny: 1 - 9							
ID TYPU ZLECENIA	ID PRACOWNIKA	ZYSK	UDZIAŁ % W ZYSKU DLA TYPU ZLECENIA	CENA USŁUGI	UDZIAŁ % W CENIE USŁUGI DLA TYPU ZLECENIA	KOSZT	UDZIAŁ % W KOSZTACH MATERIAŁU DLA TYPU ZLECENIA
1	1	189.82ZŁ	2.15%	325.41ZŁ	2.61%	189.82ZŁ	3.73%
1	2	864.06ZŁ	9.77%	1413.15ZŁ	11.32%	864.06ZŁ	15.09%
1	3	349.89ZŁ	3.96%	584.27ZŁ	4.68%	349.89ZŁ	6.44%
1	4	368.09ZŁ	4.16%	499.43ZŁ	4%	368.09ZŁ	3.61%
1	5	314.98ZŁ	3.56%	569.72ZŁ	4.56%	314.98ZŁ	7%
1	6	865.65ZŁ	9.79%	1155.66ZŁ	9.26%	865.65ZŁ	7.97%
1	7	433.68ZŁ	4.9%	594.43ZŁ	4.76%	433.68ZŁ	4.42%
1	8	795.95ZŁ	9%	1067.64ZŁ	8.55%	795.95ZŁ	7.47%
1	9	709.09ZŁ	8.02%	1062.86ZŁ	8.51%	709.09ZŁ	9.72%
1	10	444.51ZŁ	5.03%	607.2ZŁ	4.86%	444.51ZŁ	4.47%
1	11	373.07ZŁ	4.22%	585.17ZŁ	4.69%	373.07ZŁ	5.83%
1	12	504.57ZŁ	5.7%	758.1ZŁ	6.07%	504.57ZŁ	6.97%
1	13	967.09ZŁ	10.93%	1247.73ZŁ	10%	967.09ZŁ	7.71%
1	14	1038.22ZŁ	11.74%	1254.55ZŁ	10.05%	1038.22ZŁ	5.95%
1	15	626.27ZŁ	7.08%	758.21ZŁ	6.07%	626.27ZŁ	3.63%
2	1	351.51ZŁ	3.16%	514.59ZŁ	3.35%	351.51ZŁ	3.86%

k) Zapytanie PARTYCJA OBLICZENIOWA II:

Kod zapytania:

```
SELECT DISTINCT ROK.NUMER AS "ROK", KWARTAL.ID_KWARTALU AS "KWARTAL" ,
GRUPA_KLIENNELI.NAZWA_GRUPY AS "GRUPA KLIENNELI",
COUNT(ZLECENIE.ID_ZLECENIA) OVER (PARTITION BY ROK.NUMER, KWARTAL.ID_KWARTALU ,
GRUPA_KLIENNELI.NAZWA_GRUPY) AS "ILOSC ZLECEN",
TO_CHAR(ROUND((100*COUNT(ZLECENIE.ID_ZLECENIA) OVER (PARTITION BY ROK.NUMER,
KWARTAL.ID_KWARTALU, GRUPA_KLIENNELI.NAZWA_GRUPY)/COUNT(ZLECENIE.ID_ZLECENIA) OVER (PARTITION
BY ROK.NUMER, KWARTAL.ID_KWARTALU)),2)||'%') AS "UDZIAŁ % DLA KWARTALU",
TO_CHAR(ROUND((100*COUNT(ZLECENIE.ID_ZLECENIA) OVER (PARTITION BY ROK.NUMER,
KWARTAL.ID_KWARTALU, GRUPA_KLIENNELI.NAZWA_GRUPY)/COUNT(ZLECENIE.ID_ZLECENIA) OVER (PARTITION
BY ROK.NUMER)),2)||'%') AS "UDZIAŁ % DLA ROKU"
FROM ROK, KWARTAL, GRUPA_KLIENNELI, DATA, KLIENT, ZLECENIE
WHERE ROK.ID_ROKU = DATA.ROK_ID_ROKU
AND KWARTAL.ID_KWARTALU = DATA.KWARTAL_ID_KWARTALU
AND GRUPA_KLIENNELI.ID_GRUPY = KLIENT.GRUPA_KLIENNELI_ID_GRUPY
AND DATA.ID_DATY = ZLECENIE.DATA_ID_DATY
AND KLIENT.ID_KLIENTA = ZLECENIE.KLIENT_ID_KLIENTA
AND ROK.NUMER BETWEEN 2012 AND 2016
ORDER BY ROK.NUMER, KWARTAL.ID_KWARTALU , GRUPA_KLIENNELI.NAZWA_GRUPY;
```

Opis:

Druga partycja oblicza udział procentowy ilości zleceń zleconych przez daną grupę klienteli w danym kwartale danego roku względem liczby zleceń danego kwartalu oraz udział procentowy ilości zleceń zleconych przez daną grupę klienteli w danym kwartale danego roku względem liczby zleceń danego roku. Przedział lat oraz miesięczny jest wybierany przez użytkownika poprzez dostępny formularz.

Wynik zadziałania – fragment tabeli wynikowej:

WYNIK: PARTYCJA OBLICZENIOWA					
PARTYCJA OBLICZENIOWA II Zapytanie wywołano z parametrami: Przedział lat: 2012 - 2016					
ROK	KWARTAŁ	NAZWA GRUPY KLIENTELI	IŁOŚĆ ZLECENI	UDZIAŁ % W IŁOŚCI ZLECENI DLA KWARTAŁU	UDZIAŁ % W IŁOŚCI ZLECENI DLA ROKU
2012	1	Konkurencja	428	34.54%	8.87%
2012	1	Nowy	385	31.07%	7.98%
2012	1	Staly	426	34.38%	8.83%
2012	2	Konkurencja	377	32.78%	7.81%
2012	2	Nowy	392	34.09%	8.12%
2012	2	Staly	381	33.13%	7.89%
2012	3	Konkurencja	360	30.72%	7.46%
2012	3	Nowy	399	34.04%	8.27%
2012	3	Staly	413	35.24%	8.56%
2012	4	Konkurencja	421	33.25%	8.72%
2012	4	Nowy	425	33.57%	8.8%
2012	4	Staly	420	33.18%	8.7%
2013	1	Konkurencja	447	34.89%	8.82%
2013	1	Nowy	425	33.18%	8.39%

l) Zapytanie RANKING I:

Kod zapytania:

```
SELECT DZIE_N_TYGODNIA.NAZWA AS "DZIE_N", COUNT(ZLECENIE.ID_ZLECENIA) AS "ILOSC_ZLECEN",  
DENSE_RANK() OVER (ORDER BY COUNT(ZLECENIE.ID_ZLECENIA) DESC) AS "RANKING"  
FROM DZIE_N_TYGODNIA, DATA, ZLECENIE, MIESIAC, ROK  
WHERE DZIE_N_TYGODNIA.ID_DNIA = DATA.DZIE_N_TYGODNIA_ID_DNIA  
AND DATA.ID_DATY = ZLECENIE.DATA_ID_DATY  
AND ROK.ID_ROKU = DATA.ROK_ID_ROKU  
AND MIESIAC.ID_MIESIACA = DATA.MIESIAC_ID_MIESIACA  
AND ROK.NUMER BETWEEN 2011 AND 2014  
AND MIESIAC.ID_MIESIACA BETWEEN 1 AND 9  
GROUP BY DZIE_N_TYGODNIA.NAZWA;
```

Opis:

Pierwszy ranking wyświetla klasyfikację dni tygodnia ze względu na ilość zleceń przyjętych w tym dniu za pomocą funkcji DENSE_RANK(). Zapytanie ograniczone jest ze względu na rok oraz miesiąc. Przedział lat oraz miesięczny jest wybierany za pomocą dostępnego formularza.

Wynik zadziałania – fragment tabeli wynikowej:

WYNIK: RANKING		
RANKING I Zapytanie wywołano z parametrem: Przedział lat: 2011-2014 Przedział miesięczny: 1-9		
DZIE_N	IŁOŚĆ ZLECENI	MIJSCJE W RANKINGU
niedziela	2308	1
sobota	2143	2
piatek	2127	3
sroda	2074	4
wtorek	2063	5
poniedzialek	1995	6
czwartek	1976	7

m) Zapytanie RANKING II:

Kod zapytania:

```
SELECT ZLECENIE.TYP_ZLECENIA_ID_TYPU AS "ID TYPU" ,TYP_ZLECENIA.NAZWA AS "NAZWA ZLECENIA",  
COUNT(ZLECENIE.KLIENT_ID_KLIENTA) AS "ILOSC KLIENTOW",  
RANK() OVER (ORDER BY COUNT(ZLECENIE.KLIENT_ID_KLIENTA) DESC) AS "RANKING" FROM  
ZLECENIE,TYP_ZLECENIA,ROK,DATA, MIESIAC  
WHERE TYP_ZLECENIA.ID_TYPU = ZLECENIE.TYP_ZLECENIA_ID_TYPU  
AND ROK.ID_ROKU = DATA.ROK_ID_ROKU  
AND MIESIAC.ID_MIESIACA = DATA.MIESIAC_ID_MIESIACA  
AND DATA.ID_DATY = ZLECENIE.DATA_ID_DATY  
AND ROK.NUMER BETWEEN 2012 AND 2013  
AND MIESIAC.ID_MIESIACA BETWEEN 1 AND 12  
GROUP BY ZLECENIE.TYP_ZLECENIA_ID_TYPU,TYP_ZLECENIA.NAZWA;
```

Opis:

Drugi ranking wyświetla klasyfikację typów zleceń ze względu na ilość klientów, którzy je złożyli za pomocą funkcji RANK(). Zapytanie ograniczone jest ze względu na rok oraz miesiąc. Przedział lat oraz miesięczny jest wybierany za pomocą dostępnego formularza.

Wynik zadziałania – fragment tabeli wynikowej:

WYNIK: RANKING			
RANKING II			
Zapytanie wywołano z parametrem:			
Przedział lat: 2012-2013			
Przedział miesięczny: 1-12			
ID TYPU ZLECENIA	NAZWA ZLECENIA	ILOŚĆ KLIENTÓW	MIEJSCE W RANKINGU
26	Szycie - zamsz	121	1
8	Wymiana flekow	118	2
100	Wymiana podeszwy	117	3
88	Wymiana wkładki	116	4
28	Szycie - zamsz	116	4
64	Regeneracja	114	6
40	Szycie - syntetyk	114	6
81	Wymiana flekow	114	6
54	Przerobki drobne - skora	113	9
9	Naprawa suwaka	113	9
55	Przerobki drobne - skora	112	11
98	Wymiana podeszwy	112	11

n) Zapytanie RANKING III:

Kod zapytania:

```
SELECT PRACOWNIK.ID_PRACOWNIKA, PRACOWNIK.IMIE, PRACOWNIK.NAZWISKO, STANOWISKO.NAZWA,  
COUNT( ZLECENIE.ID_ZLECENIA) AS "ILOSC ZLECEN",  
DENSE_RANK() OVER (ORDER BY COUNT(ZLECENIE.ID_ZLECENIA) DESC) AS "RANKING"  
FROM PRACOWNIK, STANOWISKO, ZLECENIE, DATA, MIESIAC, ROK  
WHERE STANOWISKO.ID_STANOWISKO = PRACOWNIK.STANOWISKO_ID_STANOWISKO  
AND PRACOWNIK.ID_PRACOWNIKA = ZLECENIE.PRACOWNIK_ID_PRACOWNIKA  
AND MIESIAC.ID_MIESIACA = DATA.MIESIAC_ID_MIESIACA  
AND ROK.ID_ROKU = DATA.ROK_ID_ROKU  
AND DATA.ID_DATY = ZLECENIE.DATA_ID_DATY  
AND MIESIAC.ID_MIESIACA BETWEEN 1 AND 12  
AND ROK.NUMER BETWEEN 2011 AND 2016  
GROUP BY PRACOWNIK.ID_PRACOWNIKA, PRACOWNIK.IMIE, PRACOWNIK.NAZWISKO, STANOWISKO.NAZWA;
```


Opis:

Trzeci ranking wyświetla klasyfikację pracowników ze względu na ilość przyjętych zleceń za pomocą funkcji DENSE_RANK(). Zapytanie ograniczone jest ze względu na rok oraz miesiąc. Przedział lat oraz miesięczny jest wybierany za pomocą dostępnego formularza.

Wynik zadziałania – fragment tabeli wynikowej:

WYNIK: RANKING					
RANKING III Zapytanie wywołano z parametrem: Przedział lat: 2011-2016 Przedział miesięczny: 1-12					
ID PRACOWNIKA	IMIĘ	NAZWISKO	STANOWISKO	ILOŚĆ ZLECEŃ	MIEJSCE W RANKINGU
15	Karol	Dudek	Magazynier	2087	1
2	Kamil	Wojcik	Szwacz	2059	2
11	Rafał	Krawczyk	Kaletnik	2040	3
1	Anna	Dudek	Kierownik	2038	4
13	Piotr	Kaczmarek	Kierowca	2026	5
3	Kamil	Adamczyk	Szwacz	2016	6
12	Michał	Mazur	Kaletnik	2008	7
14	Kamil	Nowak	Magazynier	2006	8
6	Piotr	Krawczyk	Szewc	2005	9
5	Paweł	Mazur	Szwacz	1976	10
8	Jan	Adamczyk	Szewc	1969	11
7	Sebastian	Wozniak	Szewc	1965	12
9	Jan	Wozniak	Szewc	1953	13
4	Jan	Nowak	Szwacz	1930	14
10	Sebastian	Dudek	Kaletnik	1922	15

5. Interfejs.

Interfejs użytkownika został napisany za pomocą hipertekstowego języka znaczników HTTP wraz z użyciem kaskadowych arkuszy stylów CSS. Ponadto wykorzystaliśmy język JQuery w celu wsparcia mechanizmów strony – interfejsu użytkownika. Mają one na celu wspomaganie obsługi interfejsu z poziomu użytkownika tj. przycisk szybkiego powrotu do góry strony.

Interfejs użytkownika łączy się z bazą Oracle za pomocą PSP, czyli dodatku do języka PL/SQL. Cały interfejs jest zachowany w plikach o rozszerzeniu .psp. Przechowywany jest tam zarówno kod HTML/CSS jak i mechanika zapytań analitycznych napisanych za pomocą PSP.

Strona została skonstruowana w jednolitym stylu. Po lewo użytkownik ma zawsze dostępne menu wertykalne, z którego może otrzymać bezpośredni dostęp do tabel lub zapytań analitycznych. Użyty font o nazwie ROBOTO. **Warto nadmienić, iż aby interfejs działał prawidłowo należy mieć stałe połączenie z siecią internetową.**

Wygląd strony głównej:

Na stronie głównej znajduje się tytuł projektu, opis jego możliwości, schemat ERD stworzonej hurtowni danych oraz podstawowe dane autorów projektu. Po lewej stronie znajduje się wertykalne menu obsługi interfejsu.

STRONA GŁÓWNA

TABELE

ZAPYTANIA ROLLUP

ZAPYTANIA CUBE

ZAPYTANIA GROUPING SETS

OKNA RUCHOME

PARTYCJE OBLICZENIOWE

RANKINGI

HURTOWNIA DANYCH DO ANALIZ

WARSZTAT SZEWSKO-KALETNICZY

OPIS PROJEKTU

Projekt przedstawia hurtownię danych do analiz. Tematem przewodnim jest warsztat szewsko-kaletniczy. Hurtownia została zbudowana na podstawie schematu płata śniegu, co wiąże się z normalizacją wymiarów do trzeciej postaci normalnej.

W projekcie zdefiniowaliśmy dwanaście tabel.

- TABELA FAKTÓW: ZLECENIE - główna tabela w naszym schemacie. Przechowujemy w niej pojedyncze fakty. Zdefiniowaliśmy również cztery miary w celu przeprowadzania poprawnych analiz.
- TABELE WYMIARÓW: TYP ZLECENIA, KLIENT, DATA oraz PRACOWNIK. Wymiary opisują dany fakt poprzez atrybuty wymiaru. Każdy z wymiarów jest znormalizowany do trzeciej postaci normalnej.
- TABELE PODRZĘDNE WYMIARÓW: TYP PRODUKTU, GRUPA KLIENTEL, STANOWISKO, DZIEŃ TYGODNIA, MIESIĄC, KWARTAŁ oraz ROK. Tabele podrzędne wymiarów powstały na wskutek normalizacji wymiarów naszej hurtowni danych do trzeciej postaci normalnej. Posiadają one atrybuty opisujące poszczególne wymiary.

Celem analizy hurtowni danych napisaliśmy zapytania analityczne. Zastosowaliśmy operatory grupujące takie jak:

- ROLLUP
- CUBE
- GROUPING SETS

Ponadto napisaliśmy zapytania tworzące:

- PARTYCJE OBLICZENIOWE
- OKNA RUCHOME
- RANKINGI

Hurtownia danych posiadająca opisane powyżej komponenty może usprawnić proces przetwarzania analitycznego, czyli w naszym przypadku może pomóc kierownictwu warsztatu szewsko-kaletniczego w podejmowaniu poprawnych i przyszłościowych decyzji biznesowych w celu rozwoju swojego przedsiębiorstwa.

TABELE

ZAPYTANIA ROLLUP

ZAPYTANIA CUBE

ZAPYTANIA GROUPING SETS

OKNA RUCHOME

PARTYCJE OBLICZENIOWE

RANKINGI

SCHEMAT ERD

```
graph TD
    TYP_PRODUKTU --> TYP_ZLECENIA
    TYP_ZLECENIA --> ZLECENIE
    KLIENT --> ZLECENIE
    PRACOWNIK --> ZLECENIE
    STANOWISKO --> ZLECENIE
    DATA --> ZLECENIE
    DZIEŃ_TYGODNIA --> ZLECENIE
    MIESIĄC --> ZLECENIE
    KWARTAŁ --> ZLECENIE
    ROK --> ZLECENIE
```

AUTORZY:

BARTŁOMIEJ OSAK, TOMASZ PASTERNAK, MONIKA MOLEND

Wygląd podstrony z daną tabelą:

Użytkownik klikając na przycisk TABELE otrzymuje menu z wyborem dostępnych tabel. Po kliknięciu w jedną z nich zostajemy przekierowani do strony takiej jak przytoczona na zdjęciu poniżej. Klikając na przycisk POKAŻ OPIS wysunie się opis tabeli, w którym zawarte są podstawowe informacje na temat kolumn.

STRONA GŁÓWNA

TABELE

ZLECENIE

TYP ZLECENIA

TYP PRODUKTU

KLIENT

GRUPA KIENTELI

PRACOWNIK

STANOWISKO

DATA

DZIEŃ TYGODNIA

MIESIĄC

KWARTAŁ

ROK

TABELA WYMIAR: TYP ZLECENIA

POKAŻ OPIS

Tabela TYP ZLECENIA jest jednym z czterech wymiarów w naszej hurtowni danych. Posiada ona kolumnę będącą jej kluczem głównym, kolumnę z nazwą typu zlecenia. Ostatnia kolumna przyjmuje klucz obcy pochodzący z tabeli podrzędnej TYP PRODUKTU.

ID TYPU ZLECENIA	NAZWA	ID TYPU PRODUKTU
1	Wymiana flekow	1
2	Wymiana flekow	2
3	Wymiana flekow	3
4	Wymiana flekow	4
5	Wymiana flekow	5
6	Wymiana flekow	6
7	Wymiana flekow	7
8	Wymiana flekow	8
9	Wymiana flekow	9

Wygląd podstrony z zapytaniem bezparametrowym:

Użytkownik klikając na przycisk ZAPYTANIA ROLLUP, ZAPYTANIA CUBE lub ZAPYTANIA GROUPING SETS zostaje przekierowany na podstronę wyglądającą podobnie do tej przytoczonej poniżej. Użytkownik ma możliwość przeczytania podstawowego opisu zapytania oraz możliwość poznania wyniku jego działania klikając na przycisk WYNIK ZAPYTANIA.

STRONA GŁÓWNA

TABELE

ZAPYTANIA ROLLUP

ZAPYTANIA CUBE

ZAPYTANIA GROUPING SETS

OKNA RUCHOME

PARTYCJE OBLICZENIOWE

RANKINGI

ZAPYTANIA ROLLUP

Zapytanie ROLLUP I:

Pierwsze zapytanie ROLLUP zwraca podsumowanie średnich cen za usługę, kosztów materiałów oraz zysków za poszczególny typ zlecenia względem czasu wykonania. Dla każdego typu zlecenia wypisywanie jest podsumowanie częściowe. Na końcu wypisywanie jest podsumowanie całkowite.

WYNIK ZAPYTANIA

Zapytanie ROLLUP II:

Drugie zapytanie ROLLUP zwraca podsumowanie ile dany pracownik dla danego typu zlecenia oraz dla danego czasu trwania przyjął zleceń oraz ile wyniósł koszt materiałów oraz zysk od liczby przyjętych zleceń. Ponadto wypisywanie jest podsumowanie częściowe dla każdego pracownika oraz podsumowanie całkowite.

WYNIK ZAPYTANIA

Użytkownik klikając na przycisk OKNA RUCHOME, PARTYCJE OBLICZENIOWE lub RANKINGI zostaje przekierowany na podstronę wyglądającą podobnie do tej przytoczonej poniżej. Użytkownik ma możliwość przeczytania podstawowego opisu danego zapytania oraz wyboru parametrów z jakimi chce ujrzeć wynik zapytania. Ma do wyboru przedział lat, przedział miesięczny, przedział kwartalny lub wybór samego roku – wszystko zgodnie z parametrami przyjmowanymi przez dane zapytanie. Kliknięcie przycisku pokaż wynik spowoduje przekierowanie na podstronę z wynikiem danego zapytania.

Wygląd podstrony z wynikiem zapytania bezparametrowego:

STRONA GŁÓWNA	WYNIK: ZAPYTANIE ROLLUP				
TABELE					
ZAPYTANIA ROLLUP					
ZAPYTANIA CUBE					
ZAPYTANIA GROUPING SETS					
OKNA RUCHOME					
PARTYCJE OBLICZENIOWE					
RANKINGI					

ID TYPU ZLECENIA	CZAS TRWANIA USŁUGI	CENA ZA USŁUGĘ	KOSZT MATERIAŁU	ZYSK
1	1	89.12ZŁ	22.77ZŁ	66.36ZŁ
1	2	91.82ZŁ	25.17ZŁ	66.66ZŁ
1	3	49.1ZŁ	17.93ZŁ	31.17ZŁ
1	4	59.47ZŁ	17.31ZŁ	42.16ZŁ
1	5	62.27ZŁ	26.79ZŁ	35.48ZŁ
1	6	85.65ZŁ	22.83ZŁ	62.81ZŁ
1	7	74.09ZŁ	21.77ZŁ	52.32ZŁ
1	8	59.69ZŁ	22.36ZŁ	37.33ZŁ
1	9	68.23ZŁ	24.5ZŁ	43.73ZŁ
1	10	80.78ZŁ	18.89ZŁ	61.89ZŁ
1	11	74.64ZŁ	19.75ZŁ	54.89ZŁ
1	12	86.5ZŁ	24.95ZŁ	61.54ZŁ
1	13	82.07ZŁ	27.93ZŁ	54.13ZŁ
1	14	89.57ZŁ	21.47ZŁ	68.09ZŁ

Wygląd podstrony z wynikiem zapytania parametrowego:

STRONA GŁÓWNA

TABELE

ZAPYTANIA ROLLUP

ZAPYTANIA CUBE

ZAPYTANIA GROUPING SETS

OKNA RUCHOME

PARTYJCJE OBLICZENIOWE

RANKINGI

WYNIK: OKNO RUCHOME

OKNO RUCHOME III

Zapytanie wywołano z parametrem:
Przedział lat: 2011-2013
Przedział kwartalny: 1-4

ROK	KWARTAŁ	ID PRACOWNIKA	ILOŚĆ OBSŁUŻONYCH KLIENTÓW	SUMA KUMULACYJNA ILOŚCI ZLECEN ZA KWARTAŁ	SUMA KUMULACYJNA ILOŚCI ZLECEN ZA ROK	ZYSK	SUMA KUMULACYJNA ZYSKU ZA KWARTAŁ	SUMA KUMULACYJNA ZYSKU ZA ROK
2011	1	1	86	86	86	4560.92ZŁ	4560.92ZŁ	4560.92ZŁ
2011	1	2	85	171	171	5655.34ZŁ	10216.26ZŁ	10216.26ZŁ
2011	1	3	93	264	264	5330.72ZŁ	15546.98ZŁ	15546.98ZŁ
2011	1	4	80	344	344	3902.47ZŁ	19449.45ZŁ	19449.45ZŁ
2011	1	5	70	414	414	3939.32ZŁ	23388.77ZŁ	23388.77ZŁ
2011	1	6	81	495	495	4787.53ZŁ	28176.3ZŁ	28176.3ZŁ
2011	1	7	76	571	571	4632.33ZŁ	32808.63ZŁ	32808.63ZŁ

6. Wnioski.

Tworzenie projektu hurtowni danych umożliwiło nam poznanie pojęć oraz zagadnień z tworzeniem hurtowni danych do analiz oraz budowania interfejsu użytkownika. Poznaliśmy budowę hurtowni danych na podstawie schematu płatka śniegu – podziały na tabelę faktów oraz jej wymiary znormalizowane do trzeciej postaci normalnej. Ponadto nabyliśmy wiedzę na temat praktycznego zastosowania nowych komponentów do zapytań takich jak klauzule: ROLLUP, CUBE czy GROUPING SETS. Ponadto poznaliśmy zasadę działania partycjonowania w celu stworzenia partycji obliczeniowych. W szczególności zainteresowało nas pojęcie okien ruchomych i możliwości, które one posiadają. Ciekawym typem zapytania okazały się być rankingi, które mogą się doskonale przydać do analizy biznesowej naszego przedsiębiorstwa.

Dzięki projektowi poznaliśmy rozszerzenie języka PL/SQL o pakiet PSP umożliwiające szybkie budowanie interfejsów bezpośrednio połączonych z bazą danych Oracle. Jest ono bardzo wygodnym narzędziem do budowania stron dla użytkowników, ponieważ projektant nie musi znać np. języka PHP celem połączenia się z bazą danych. Sam proces wstawiania zapytań do stron internetowych okazał się być bardzo przyjazny.

Projekt pozwolił na znaczne poznanie tajników budowania stron www od podstaw. Dzięki temu poznaliśmy składnię języka znaczników HTML oraz część ogromnych możliwości kaskadowych arkuszy stylu CSS. Pomimo, że nie są to języki programowania ich poznanie należało do bardzo przyjemnych. Ponadto chcąc projektować udogodnienia dla użytkownika nabyliśmy wiedzę na temat podstaw programowania w JQuery.

Oczywiście nasz projekt można rozwinąć. Głównym celem rozwoju byłoby napisanie jeszcze większej ilości zapytań analitycznych dość mocno zoptymalizowanych. Ponadto cały projekt można by umieścić na serwerze, aby dostęp do niego był możliwy gdziekolwiek jesteśmy. Mimo, iż jesteśmy zadowoleni z interfejsu oczywiście można go jeszcze ulepszyć dodając nowe klasy kaskadowych arkuszy stylu czy pisząc nowe skrypty w języku JavaScript.

Ciekawostką może być generator danych, które zasilają naszą hurtownię. Został on napisany w języku Java zgodnie z paradygmatem obiektowym. Zastosowaliśmy w nim dość ciekawe algorytmy – najciekawszym może być ustalanie realnej daty zgodnej z kalendarzem wpisywanej do pliku CSV. Tak wygenerowane dane zapisywane są zapisywanie do plików CSV a następnie ładowanie narzędziem SQLLoader, które umożliwia natychmiastowe ładowanie danych do hurtowni lub zwykłych tabel baz danych.

Naszym zdaniem hurtownie danych to olbrzymi potencjał dla większości przedsiębiorstw, w szczególności dla dużych firm. Odpowiednie zapytania analityczne pozwolą analitykom rynku lub ekonomistom nabycie odpowiedniej wiedzy na temat rozwoju przedsiębiorstwa i wdrożenie planów na dalszy jego rozwój.