

**Projekt z rozproszonych i obiektowych  
systemów baz danych**

**Projekt aplikacji wykorzystującej  
mechanizmy replikacji Oracle**

**AUTOR:**

Adam Kowalski

**Indeks: 112345**

E-mail:

**PROWADZĄCY ZAJĘCIA:**

Dr inż. Robert Wójcik, W4/I-6

**OCENA PRACY:**

## Spis treści

Spis rysunków .....	3
Spis tabel.....	4
1. Wstęp.....	5
1.1. Cele projektu .....	5
1.2. Założenia projektowe .....	5
1.3. Zakres projektu .....	5
2. Replikacja w systemie baz danych Oracle .....	6
2.1. Pojęcie replikacji i podstawowe informacje.....	6
2.2. Replikacja migawkowa .....	6
2.3. Replikacja multimaster .....	8
2.4. Narzędzia graficznego zarządzania replikacją w środowisku Oracle 10g.....	9
3. Model konceptualny i fizyczny baz danych .....	10
3.1. Model konceptualny.....	10
3.2. Model fizyczny .....	11
4. Implementacja baz danych w środowisku Oracle .....	12
4.1. Realizacja bazy danych .....	12
4.2. Definiowanie łączników.....	16
4.3. Wykorzystanie mechanizmów replikacji migawkowej.....	17
4.4. Wykorzystanie mechanizmów replikacji multimaster .....	21
5. Projekt i implementacja aplikacji klienckiej.....	23
5.1. Funkcje aplikacji - diagram przypadków użycia.....	23
5.2. Realizacja wybranych funkcjonalności .....	24
6. Wdrożenie i testowanie aplikacji.....	25
7. Podsumowanie .....	33
Literatura .....	34

# Spis rysunków

RYSUNEK 1. NARZĘDZIE ORACLE ENTERPRISE MANAGER CONSOLE.....	9
RYSUNEK 2. MODEL KONCEPTUALNY BAZY DANYCH .....	10
RYSUNEK 3. MODEL FIZYCZNY BAZY DANYCH .....	11
RYSUNEK 4. NARZĘDZIE DATABASE CONFIGURATION ASSISTANT .....	12
RYSUNEK 5. TWORZENIE DOMYŚLNEJ BAZY DANYCH.....	13
RYSUNEK 6. NARZĘDZIE ORACLE NET MANAGER .....	13
RYSUNEK 11. TWORZENIE WĘZŁA NADRZĘDNEGO W REPLIKACJI MIGAWKOWEJ .....	17
RYSUNEK 12. TWORZENIE UŻYTKOWNIKA ODPOWIEDZIALNEGO ZA REPLIKACJĘ.....	17
RYSUNEK 13. TWORZENIE NADRZĘDNEJ GRUPY REPLIKACJI .....	18
RYSUNEK 14. TWORZENIE DZIENNIKÓW MIGAWEK .....	18
RYSUNEK 15. TWORZENIE WĘZŁA MIGAWKOWEGO .....	19
RYSUNEK 16. TWORZENIE GRUPY MIGAWEK - WYBIERANIE OBIEKTÓW DO REPLIKOWANIA.....	19
RYSUNEK 17. USTAWIANIE PARAMETRÓW MIGAWEK.....	20
RYSUNEK 18. ZMIANA SPOSOBU ODŚWIEŻANIA MIGAWEK .....	20
RYSUNEK 19. TWORZENIE WĘZŁÓW NADRZĘDNYCH DLA REPLIKACJI MULTIMASTER .....	21
RYSUNEK 20. TWORZENIE UŻYTKOWNIKÓW ODPOWIEDZIALNYCH ZA REPLIKACJĘ MIGAWKOWĄ.....	21
RYSUNEK 21. TWORZENIE GRUPY REPLIKACJI MULTIMASTER.....	22
RYSUNEK 22. DODAWANIE OBIEKTÓW DO GRUPY REPLIKACJI MULTIMASTER.....	22
RYSUNEK 23. DIAGRAM PRZYPADKÓW UŻYCIA.....	23
RYSUNEK 24. ZMIANA DANYCH PO STRONIE KRAKOWA (1) .....	25
RYSUNEK 25. ZMIANA DANYCH PO STRONIE KRAKOWA (2) – ZATWIERDZENIE NOWYCH DANYCH.....	26
RYSUNEK 26. ZMIANA DANYCH PO STRONIE KRAKOWA (3) – POBRANIE DANYCH Z WĘZŁA NADRZĘDNEGO .....	26
RYSUNEK 27. ZMIANA DANYCH PO STRONIE KRAKOWA (4) - ZMIANY W WĘZLE NADRZĘDNYM .....	27
RYSUNEK 28. ZMIANA DANYCH PO STRONIE WROCŁAWIA (1) .....	27
RYSUNEK 29. ZMIANA DANYCH PO STRONIE WROCŁAWIA (2) – ZATWIERDZENIE NOWYCH DANYCH .....	28
RYSUNEK 30. ZMIANA DANYCH PO STRONIE WROCŁAWIA (3) - POBRANIE DANYCH Z WĘZŁA PODRZĘDNEGO.....	28
RYSUNEK 31. ZMIANA DANYCH PO STRONIE WROCŁAWIA (4) – OCZEKIWANIE NA ZMIANY W WĘZLE PODRZĘDNYM .....	29
RYSUNEK 32. ZMIANA DANYCH PO STRONIE WROCŁAWIA (5) – PONOWNE POBRANIE DANYCH Z WĘZŁA PODRZĘDNEGO .....	29
RYSUNEK 33. ZMIANA DANYCH PO STRONIE WROCŁAWIA (6) – ZMIANA W WĘZLE PODRZĘDNYM.....	30
RYSUNEK 34. USUNIĘCIE REKORDU Z TABELI UZYTKOWNICY (1).....	30
RYSUNEK 35. USUNIĘCIE REKORDU Z TABELI UZYTKOWNICY (2) – ZAPISANIE ZMIAN .....	31
RYSUNEK 36. USUNIĘCIE REKORDU Z TABELI UZYTKOWNICY (3) – PONOWNE POBRANIE DANYCH Z WĘZŁA W KRAKOWIE ..	31
RYSUNEK 37. USUNIĘCIE REKORDU Z TABELI UZYTKOWNICY (3) – ZMIANY W WĘZLE W KRAKOWIE .....	32

# Spis tabel

TABELA 1. NARZĘDZIE ORACLE ENTERPRISE MANAGER CONSOLE .....	9
TABELA 2. MODEL KONCEPTUALNY BAZY DANYCH .....	10
TABELA 3. MODEL FIZYCZNY BAZY DANYCH .....	11

# 1. Wstęp

Prezentowany projekt został opracowany w trakcie realizowania laboratorium z przedmiotu *Rozproszone systemy baz danych*. W trakcie prac projektowych stworzona została rozproszona baza danych samochodów. W bazie tej przechowywane są informacje na temat samochodów, serwisów samochodowych, pracowników i klientów firmy zajmującej się wypożyczaniem samochodów na terenie Europy.

## 1.1. Cele projektu

Celem projektu jest stworzenie bazy danych na potrzeby firmy zajmującej się wypożyczaniem samochodów. Firma ta posiada swe oddziały w kilku europejskich miastach. We Wrocławiu znajduje się główny serwer należący do systemu informatycznego firmy. W innych miastach znajdują się węzły związane z działalnością poszczególnych serwisów. Najważniejsze z punktu widzenia działalności firmy jest utrzymanie aktualnych i spójnych danych na serwerze wrocławskim. Pozostałe serwery służą do modyfikacji danych na serwerze głównym oraz do udostępniania w serwisach informacji zawartych na serwerze wrocławskim.

## 1.2. Założenia projektowe

Projekt został wykonany przy użyciu środowiska *Oracle 10g*. Z powodu ograniczeń sprzętowych rozproszona baza danych składa się tylko z dwóch węzłów. Jednym z nich będzie wspomniany węzeł wrocławski a drugim węzeł krakowski. W trakcie realizacji projektu zostały wykorzystane mechanizmy replikacji migawkowej i multimaster dostępne w środowisku *Oracle 10g*. Do wykonania projektu bazy danych wykorzystane zostało narzędzie *PowerDesigner 12*. Aplikacja kliencka została wykonana w technologii *Java* przy użyciu środowiska *NetBeans 6.1*.

## 1.3. Zakres projektu

W rozdziale drugim przedstawiono ... , .....

## 2. Replikacja w systemie baz danych Oracle

### 2.1. Pojęcie replikacji i podstawowe informacje

Replikacja danych to proces powielania informacji pomiędzy różnymi serwerami baz danych. W środowisku Oracle dostępne są następujące typy replikacji [1]:

- **replikacja migawkowa** (ang. materialized view replication)

W skład środowiska replikacji migawkowej wchodzi węzły nadrzędne (ang. master sites) i węzły migawkowe (ang. materialized view sites). Węzeł migawkowy to taka instancja bazy danych, która pobiera dane do zawartych w niej migawek z tabel węzła nadrzędnego. Należy zaznaczyć, że węzeł migawkowy może być równocześnie węzłem nadrzędnym dla innych węzłów.

- **replikacja multimaster**

W replikacji multimaster inaczej niż w replikacji migawkowej wszystkie węzły są równorzędne

(każdy z węzłów jest węzłem nadrzędnym dla pozostałych). W takim przypadku zmiana w którymkolwiek węzle zostaje zaaplikowana we wszystkich węzłach.

- **replikacja hybrydowa** (mieszana)

Replikacja ta jest kombinacją dwóch powyższych replikacji.

### 2.2. Replikacja migawkowa

Jak już wspomniano środowisko replikacji migawkowej tworzą węzły nadrzędne i węzły migawkowe. Każdy węzeł migawkowy jest połączony dokładnie z jednym węzłem nadrzędnym, a węzły migawkowe mogą być węzłami nadrzędnymi dla pozostałych węzłów.

*Replikacja migawkowa* jest replikacją na poziomie rekordu (każda operacja na rekordzie musi zostać przesłana i wykonana we wszystkich węzłach).

Replikacja migawkowa jest używana najczęściej w sytuacjach, gdy wymagana jest replikacja jedynie jakiegoś podzbioru danych.

Aby skorzystać z mechanizmu replikacji migawkowej należy stworzyć odpowiednie **łącznie** (ang. *links*).

**Migawka** (ang. *snapshot*) to inaczej perspektywa zmaterializowana. Jest to obiekt bazy stanowiący w najprostszym przypadku kopię danych z tabeli bazy zdalnej. Moment oraz częstotliwość odświeżania migawki są określone w definicji.

Migawki możemy ze względu na różne kryteria. Najważniejszym z podziałów jest podział na **migawki modyfikowalne** (ang. *updatable snapshots*), **migawki zapisywalne** (ang. *writable snapshots*) i **migawki tylko do odczytu** (ang. *read-only snapshots*).

Do migawki modyfikowalnej można wstawić rekord, zmienić rekord lub go usunąć a mechanizmy replikacji zapewnią, że zmiany wprowadzone do migawki zostaną przesłane i wprowadzone w tabeli źródłowej. Migawka modyfikowalna musi należeć do grupy migawek o takiej samej nazwie jak nadrzędna grupa replikacji do której należy tabela źródłowa.

Do migawki zapisywalnej można również dodać, usunąć lub zmienić rekord. Zmiany te jednak nie są przesyłane do tabeli źródłowej i zostają utracone po pierwszej operacji odświeżenia migawki od wprowadzenia zmian.

Na migawkach tylko do odczytu nie można dokonywać zmian na rekordach.

Aby stworzyć migawkę modyfikowalną lub zapisywalną należy w trakcie jej tworzenia użyć klauzuli *for update* [3][4].

Migawki możemy również podzielić ze względu na sposób odświeżania:

- **odświeżanie pełne** – polega na skopiowaniu całej zawartości tabeli źródłowej do migawki
- **odświeżanie przyrostowe** – polega na skopiowaniu do migawki wyłącznie zmian jakie nastąpiły w tabeli źródłowej. Jeśli zaś migawka ma być odświeżana przyrostowo musi zostać dla niej stworzony dziennik migawki (ang. *materialized view log*).

Migawki mogą być również odświeżane synchronicznie lub asynchronicznie.

Możliwe jest również stworzenie grupy odświeżania (ang. *refresh group*) dzięki której możliwe jest jednoczesne odświeżanie wielu migawek znajdujących się w tej samej grupie odświeżania. Migawki z jednej grupy migawek mogą należeć do różnych grup odświeżania, a grupa odświeżania może zawierać migawki z różnych grup migawek.

## 2.3. Replikacja multimaster

Jak już wspomniano wcześniej cechami charakterystycznymi replikacji multimaster są równorzędność węzłów oraz identyczność replikowanych obiektów (najczęściej tabel). Należy również nadmienić, że podstawowym celem stosowania replikacji multimaster jest zwiększenie dostępności danych. Replikacja multimaster daje też możliwość rozłożenia obciążenia pomiędzy poszczególne węzły.

Domyślnym trybem działania replikacji multimaster w środowisku Oracle jest **replikacja na poziomie rekordu** [1]. Oracle udostępnia również **replikację proceduralną** (pomiędzy węzłami przesyłane są wywołania procedur).

Aby skorzystać z replikacji multimaster należy zdefiniować odpowiednie łączniki (jeżeli środowisko składa się z M węzłów to w każdym z nich musi zostać zdefiniowanych M-1 łączników do węzłów pozostałych).

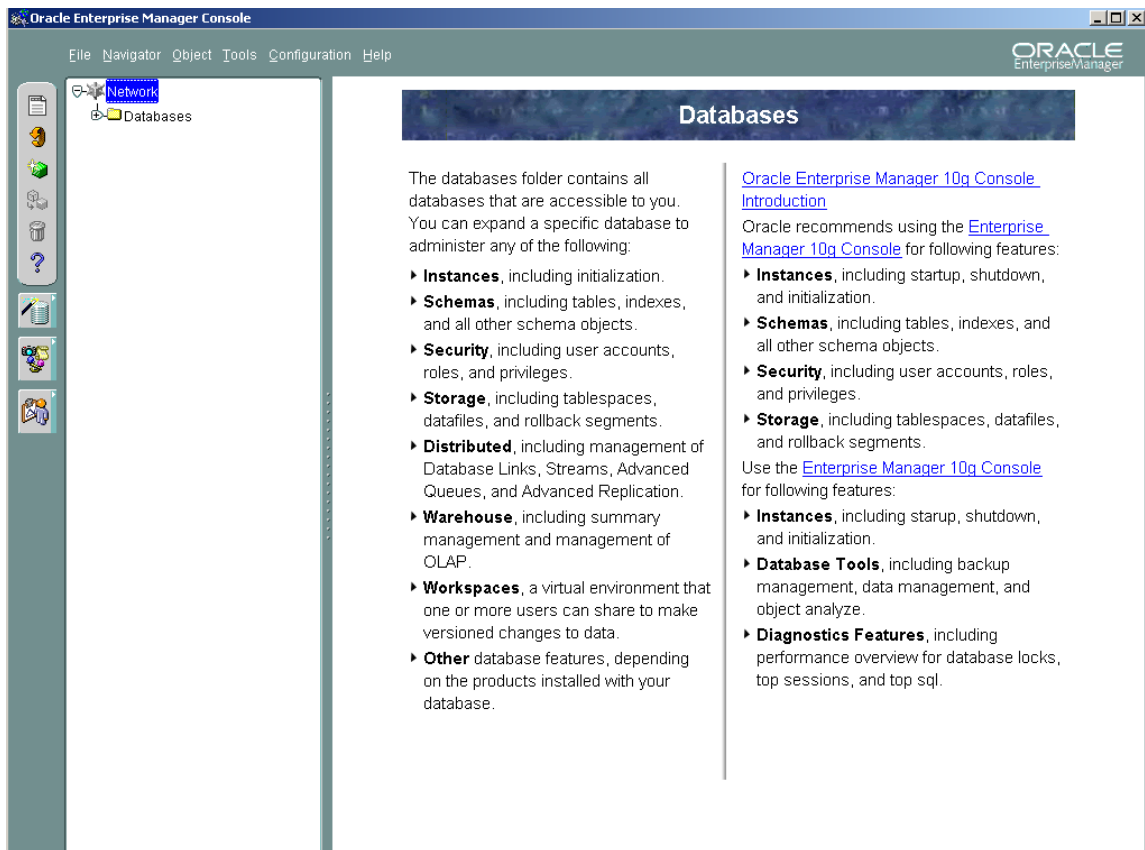
We wszystkich węzłach znajduje **nadrzędna grupa replikacji** (ang. *master group*) o tej samej nazwie i zawartości [3][4]. W każdym węźle grupa ta jest związana z listą węzłów do których mają być replikowane wszystkie operacje.

Domyślnym sposobem działania replikacji multimaster jest **replikacja asynchroniczna**. W replikacji tej transakcje na lokalnej replice danych zostają oddzielone od transakcji przesyłającej modyfikacje do zdalnych węzłów. Natomiast w **replikacji synchronicznej** dane są modyfikowane lokalnie i przesyłane do zdalnych węzłów w ramach jednej transakcji (wykorzystuje się tu protokół zatwierdzania dwufazowego).



## 2.4. Narzędzia graficznego zarządzania replikacją w środowisku Oracle 10g

Środowisko Oracle 10g dostarcza wielu narzędzi do graficznej konfiguracji bazy danych. Jednym z nich jest *Oracle Enterprise Manager Console* (Rysunek 1) w skład którego wchodzi *Replication Management Tool*.

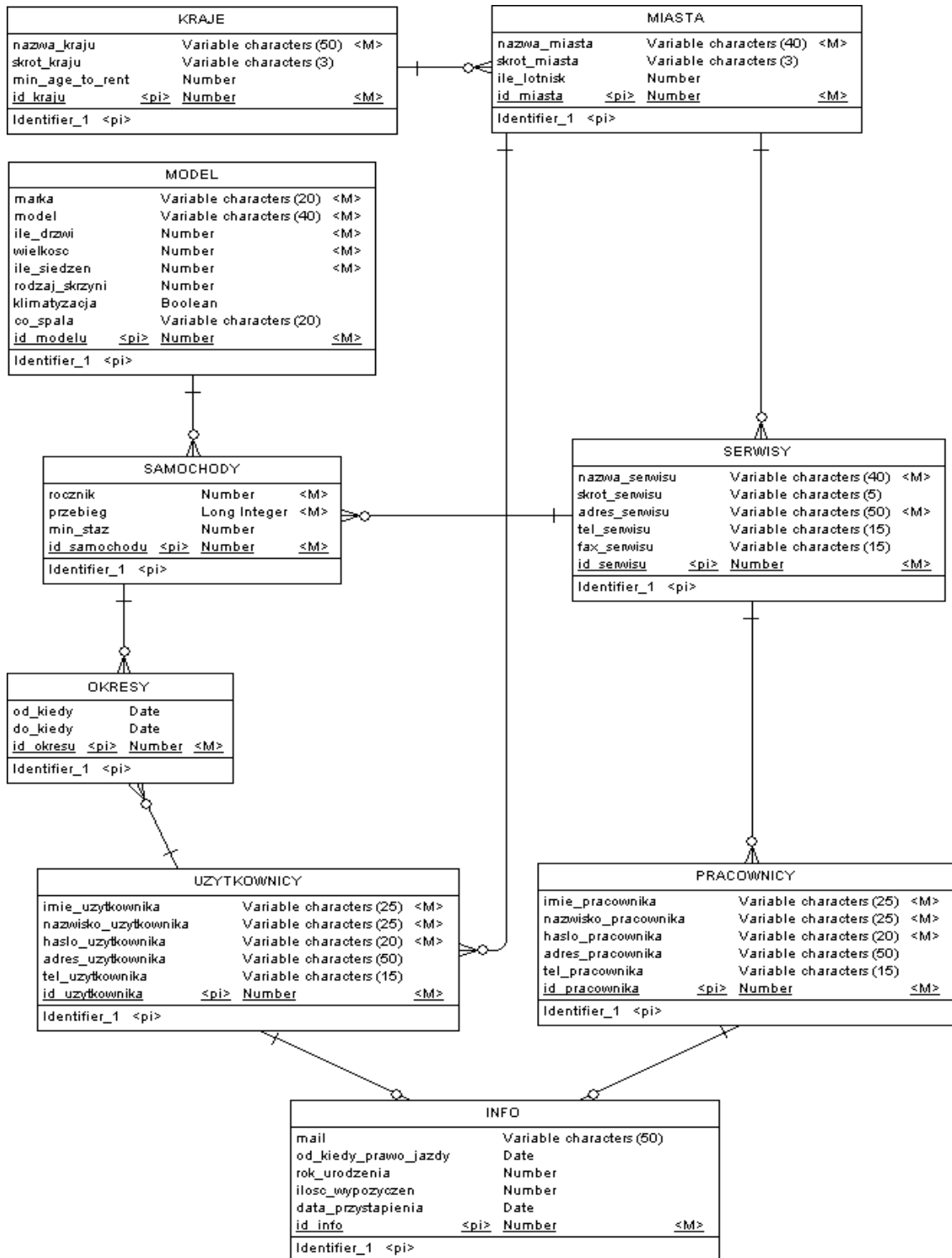


RYSUNEK 1 NARZĘDZIE ORACLE ENTERPRISE MANAGER CONSOLE

Z narzędzia tego mogą korzystać wszyscy użytkownicy z przywilejem *SELECT ANY DICTIONARY*. W trakcie realizacji projektu właśnie to narzędzie zostało wykorzystane do zrealizowania zarówno replikacji migawkowej jak i replikacji multimaster.

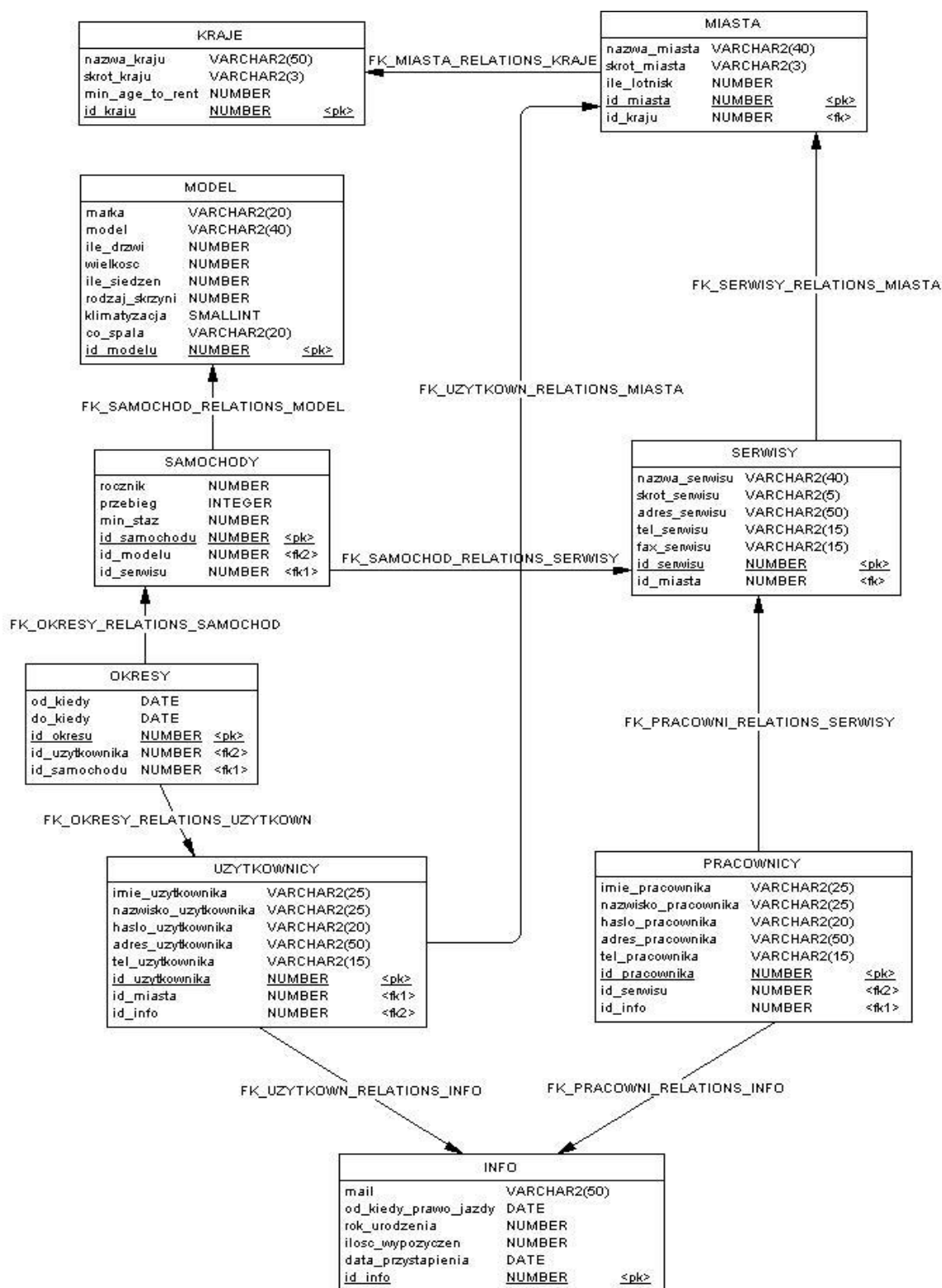
### 3. Model konceptualny i fizyczny baz danych

#### 3.1. Model konceptualny



RYSUNEK 2 MODEL KONCEPTUALNY BAZY DANYCH

### 3.2. Model fizyczny



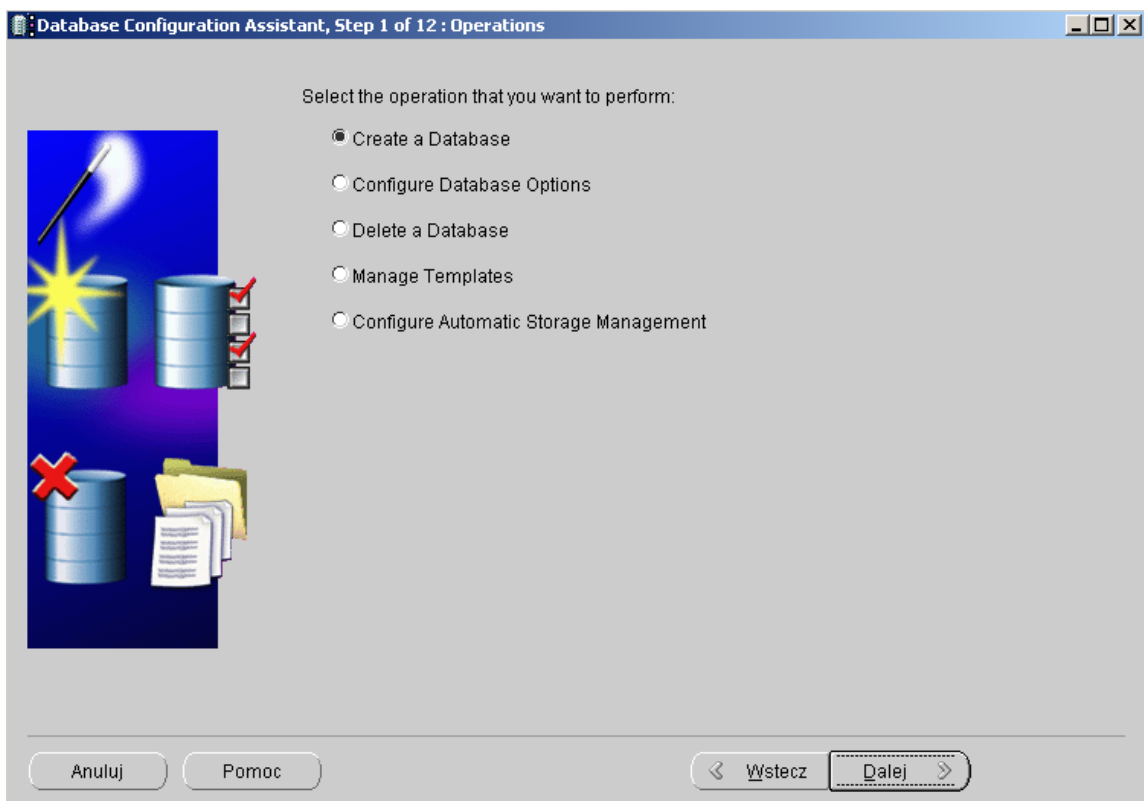
RYSUNEK 3 MODEL FIZYCZNY BAZY DANYCH

## 4. Implementacja baz danych w środowisku Oracle

### 4.1. Realizacja bazy danych

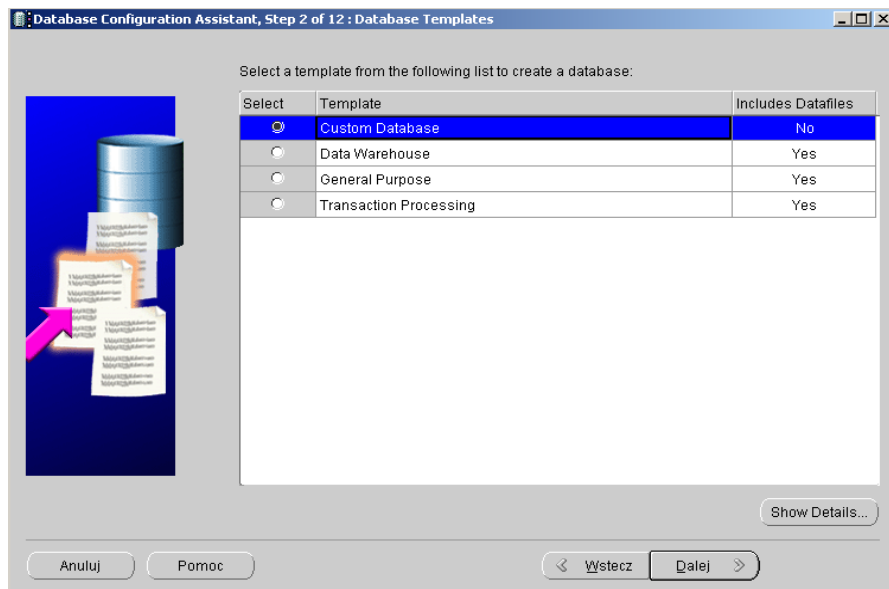
Aby stworzyć instancję bazy Oracle 10g należało zainstalować odpowiednie oprogramowanie na wszystkich używanych węzłach. Problemem przy realizacji tego kroku okazały się duże wymagania sprzętowe. W dokumentacji napisano, że minimalna ilość pamięci RAM aby można było korzystać z oprogramowania Oracle 10g to 256 MB [5]. Doświadczenie jednak uczy, że aby swobodnie z niego korzystać potrzeba co najmniej 512 MB. Z powodu powyższych wymagań w skład realizowanej rozproszonej bazy weszły tylko dwa węzły.

Po zainstalowaniu oprogramowania (w tym również pakietu klienckiego), w każdym z węzłów została stworzona nowa instancja bazy danych przy pomocy narzędzia *Database Configuration Assistant* (Rysunek 4).



RYSUNEK 4 NARZĘDZIE DATABASE CONFIGURATION ASSISTANT

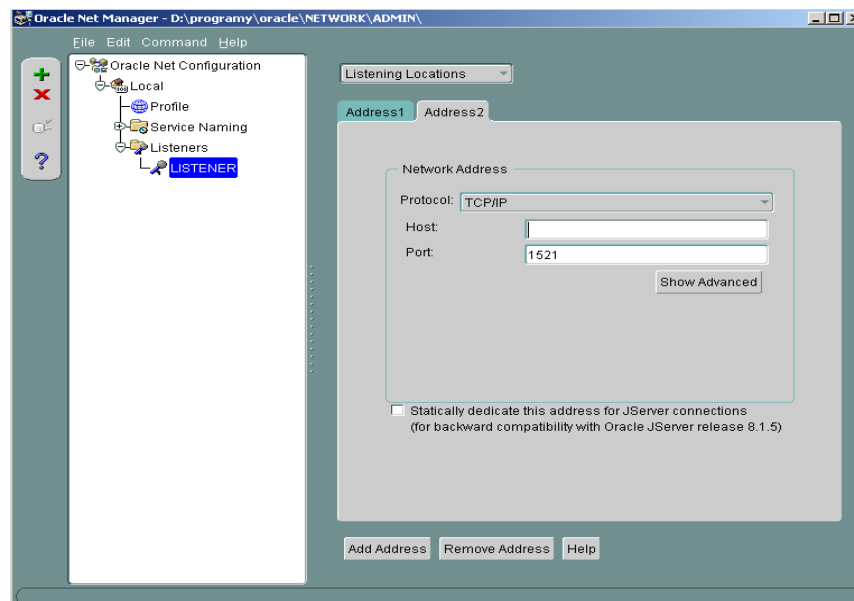
W trakcie tworzenia instancji bazy danych wybrano opcję Custom Database.



RYSUNEK 5 TWORZENIE DOMYŚLNEJ BAZY DANYCH

Następnie wykonano pozostałe kroki konfiguracji bazy danych (m.in. nadano bazie globalną nazwę oraz SID).

Na koniec sprawdzono czy istnieje odpowiedni proces nasłuchu (ang. *listener*). Konfiguracji procesu nasłuchu można dokonać za pomocą programu *Oracle Net Manager* (Rysunek 6).



RYSUNEK 6 NARZĘDZIE ORACLE NET MANAGER

Skrypt do realizacji bazy danych został wygenerowany na podstawie modelu fizycznego bazy przy użyciu programu *PowerDesigner 12*. Następnie został napisany skrypt tworzący sekwencje i triggery w celu automatycznego dodawania do rekordów identyfikatorów będących kluczami głównymi.

Poniżej znajduje się część skryptu odpowiedzialna za tworzenie tabel **KRAJE** i **MIASTA** oraz relacji między nimi:

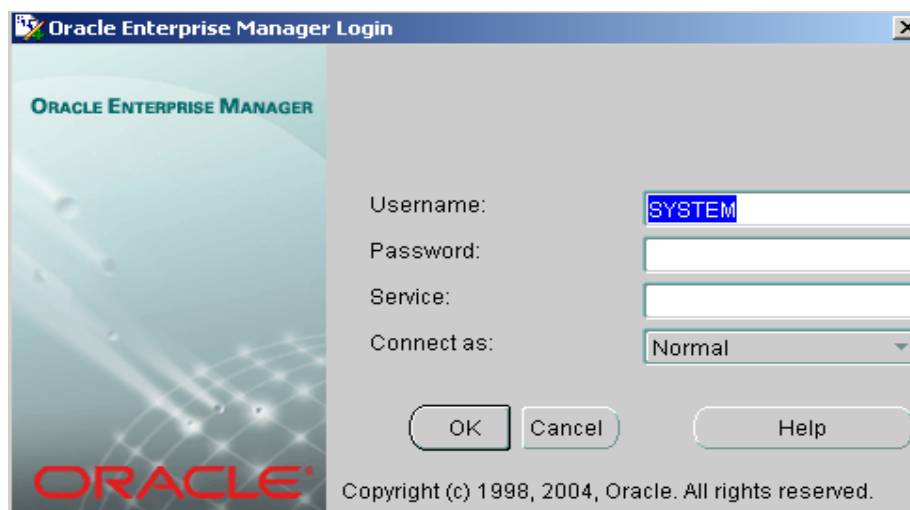
```
/*=====
*/
/* Table: KRAJE */
/*=====
*/
create table MVREP.KRAJE (
  NAZWAKRAJU    VARCHAR2(50)          not null,
  SKROTKRAJU    VARCHAR2(3),
  MINAGETORENT NUMBER                default 21,
  IDKRAJU       NUMBER                not null,
  constraint PK_KRAJE primary key (IDKRAJU)
);

/*=====
*/
/* Table: MIASTA */
/*=====
*/
create table MVREP.MIASTA (
  NAZWAMIASTA   VARCHAR2(40)          not null,
  SKROTMIASTA   VARCHAR2(3),
  ILELOTNISK    NUMBER,
  IDMIASTA      NUMBER                not null,
  IDKRAJU       NUMBER                not null,
  constraint PK_MIASTA primary key (IDMIASTA)
);

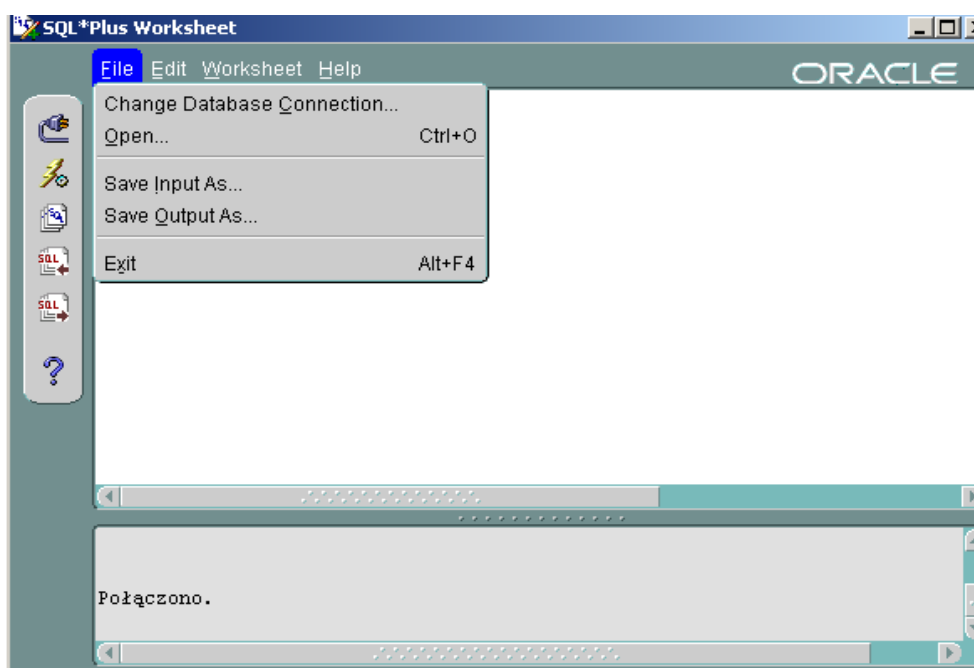
/*=====
*/
/* Index: "Relationship_1_FK" */
/*=====
*/
create index "Relationship_1_FK" on MVREP.MIASTA (
  IDKRAJU ASC
);

alter table MVREP.MIASTA
  add constraint FK_MIASTA_RELATIONS_KRAJE foreign key (IDKRAJU)
  references MVREP.KRAJE (IDKRAJU);
```

Skrypt uruchomiono przy użyciu programu *SQL\*Plus Worksheet* (Rysunek 7) będącego częścią pakietu Oracle Client.



**RYСУNEK 7 LOGOWANIE DO BAZY PRZY UŻYCIU NARZĘDZIA SQL\*PLUS WORKSHEET**

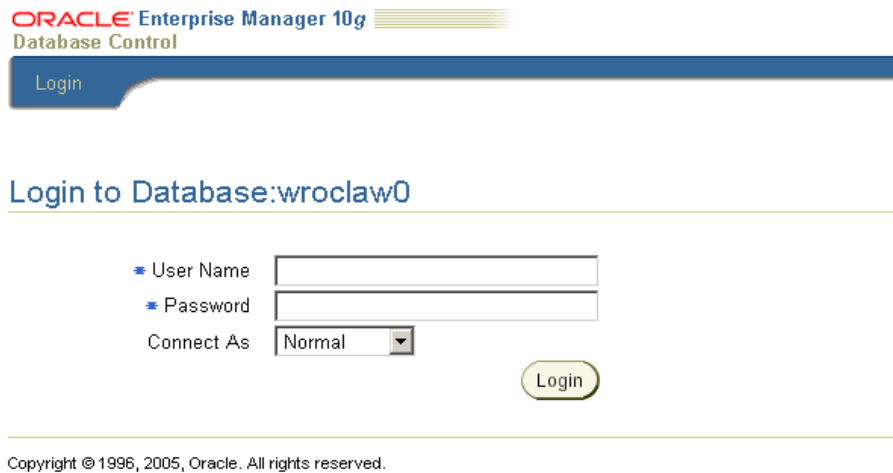


**RYСУNEK 8 NARZĘDZIE SQL\*PLUS WORKSHEET**

## 4.2. Definiowanie łączników

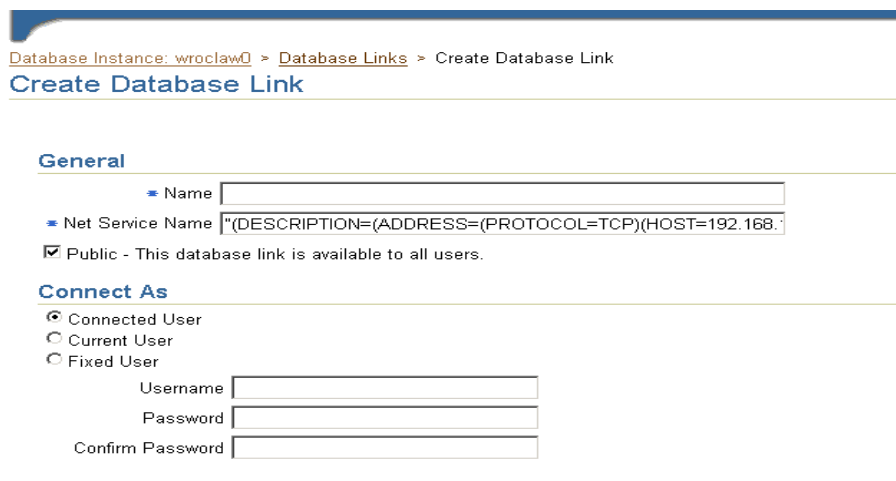
Łączniki potrzebne do wykorzystania mechanizmów replikacji zostały zdefiniowane przy użyciu konsoli dostępnej przez przeglądarkę internetową (Rysunek 9). Narzędzie to pozwala w łatwy sposób zdefiniować linki (Rysunek 10).

Ponieważ stworzono dwie instancje bazy danych (wroclaw0 i krakow1) w każdej z nich zdefiniowano linki do drugiej instancji.



The screenshot shows the Oracle Enterprise Manager 10g Database Control interface. At the top, it says "ORACLE Enterprise Manager 10g Database Control". Below that is a blue bar with the word "Login". The main heading is "Login to Database:wroclaw0". There are three input fields: "User Name", "Password", and "Connect As" (set to "Normal"). A "Login" button is to the right. At the bottom, it says "Copyright © 1996, 2005, Oracle. All rights reserved."

RYSUNEK 9 ORACLE ENTERPRISE MANAGER Z POZIOMU PRZEGLĄDARKI



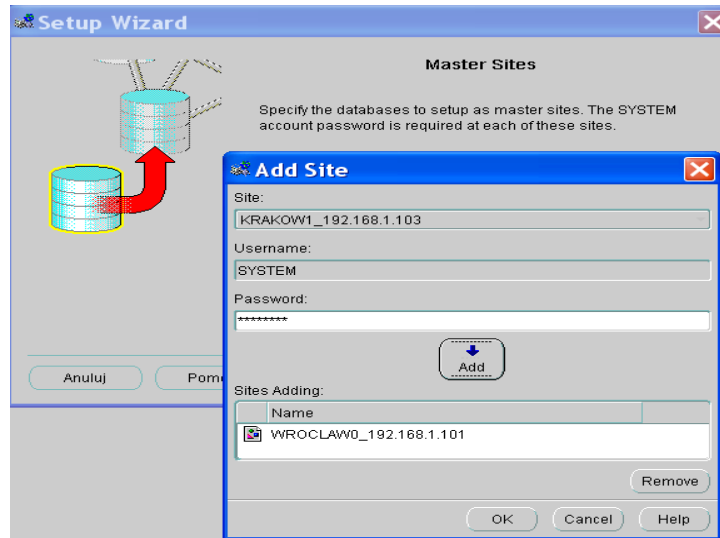
The screenshot shows the "Create Database Link" page in Oracle Enterprise Manager. The breadcrumb is "Database Instance: wroclaw0 > Database Links > Create Database Link". The page title is "Create Database Link". Under the "General" section, there is a "Name" field, a "Net Service Name" field with a value starting with "(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=192.168...)", and a checked "Public" checkbox. Under the "Connect As" section, there are radio buttons for "Connected User", "Current User", and "Fixed User". Below these are fields for "Username", "Password", and "Confirm Password".

RYSUNEK 10 TWORZENIE ŁĄCZNIKÓW



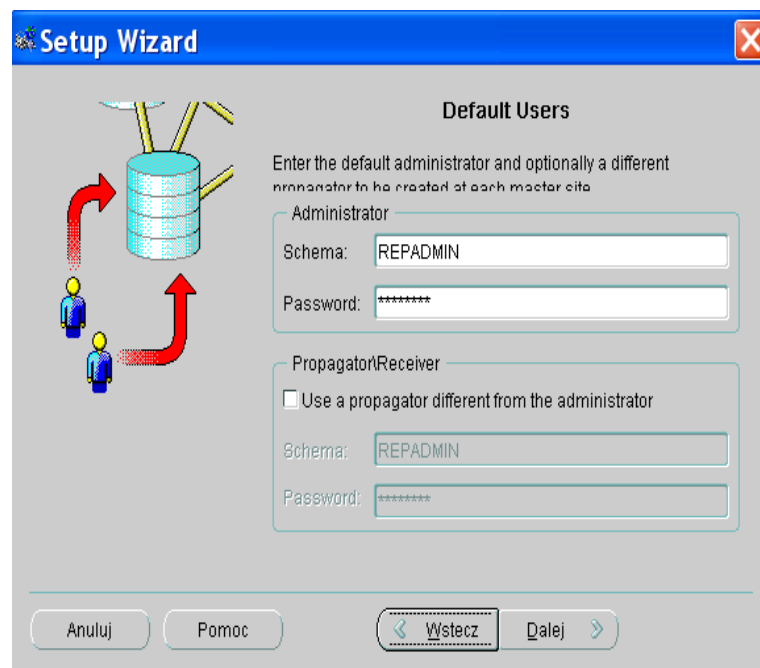
### 4.3. Wykorzystanie mechanizmów replikacji migawkowej

Aby skorzystać z mechanizmu replikacji migawkowej, jak już wcześniej wspomniano, należy utworzyć jeden węzeł nadrzędny (Rysunek 11). Węzłem nadrzędnym został węzeł **wroclaw0**.



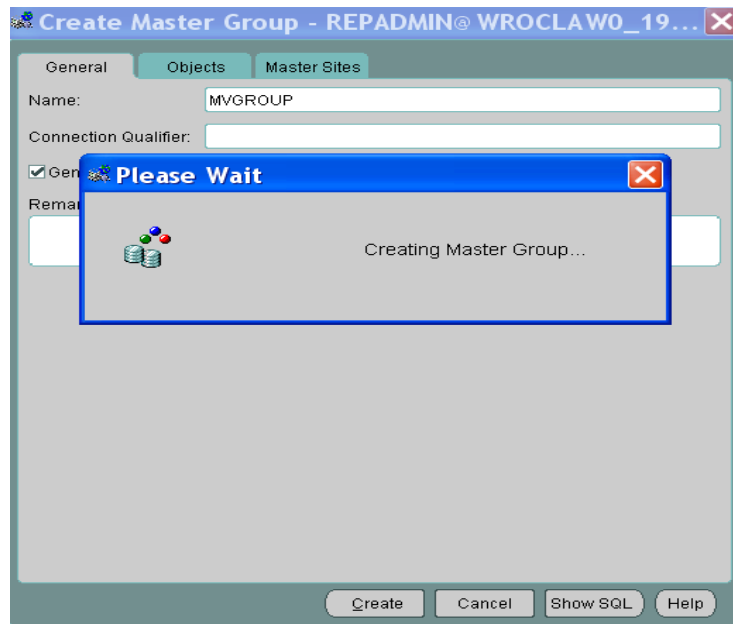
RYSUNEK 11 TWORZENIE WĘZŁA NADRZĘDNEGO W REPLIKACJI MIGAWKOWEJ

Narzędzie *Replication Management Tool* daje możliwość stworzenia wraz z węzłem nadrzędnym użytkownika o domyślnej nazwie **REPADMIN** odpowiedzialnego za replikację.



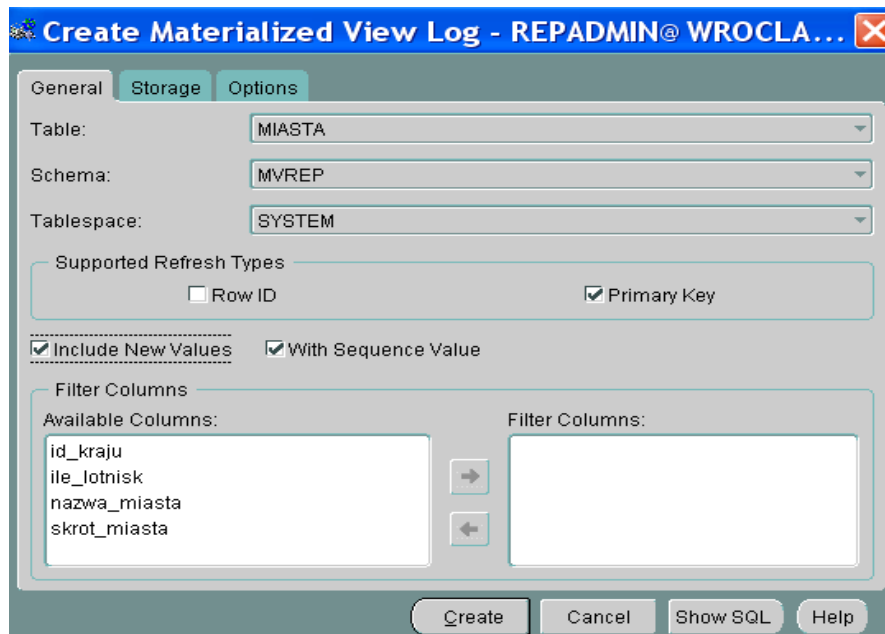
RYSUNEK 12 TWORZENIE UŻYTKOWNIKA ODPOWIEDZIALNEGO ZA REPLIKACJĘ

Kolejnym krokiem było stworzenie również po stronie instancji **wroclaw0** nadrzędnej grupy replikacji (ang. *master group*).



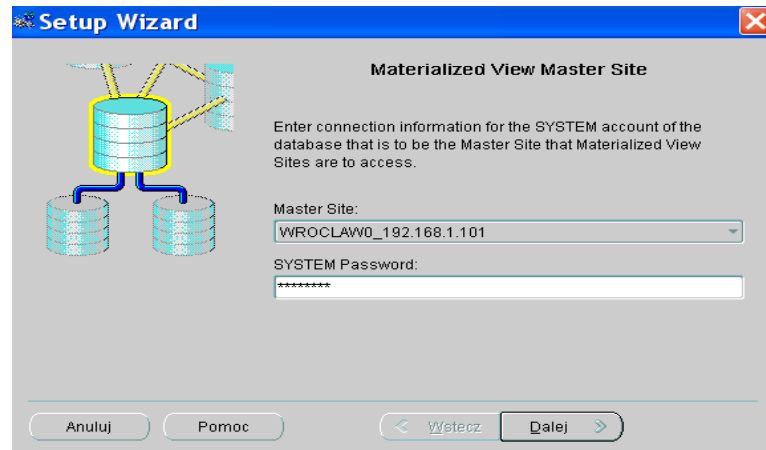
RYSUNEK 13 TWORZENIE NADRZĘDNEJ GRUPY REPLIKACJI

Aby móc korzystać z migawek modyfikowalnych stworzono następnie po stronie bazy **wroclaw0** dzienniki migawek.



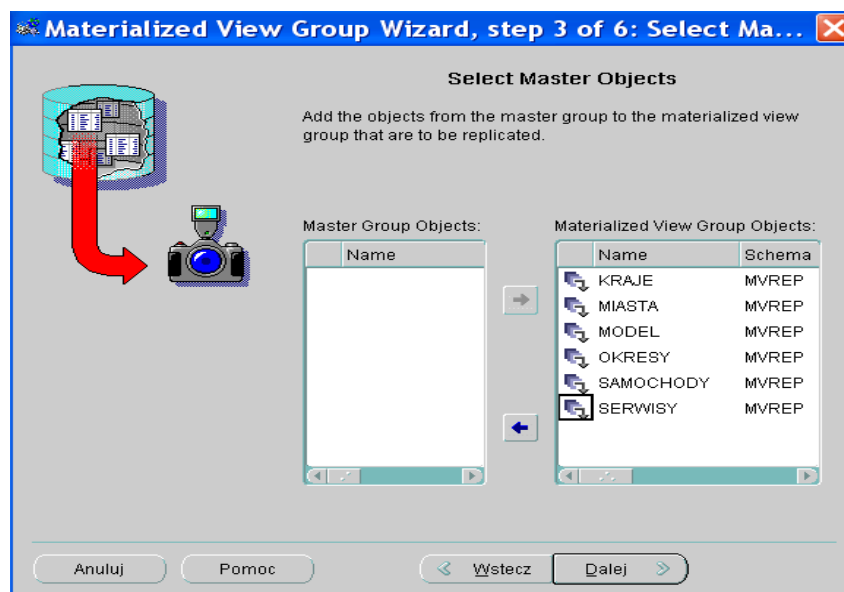
RYSUNEK 14 TWORZENIE DZIENNIKÓW MIGAWEK

Kolejnym krokiem było stworzenie węzła migawkowego (**krakow1**). Należało w nim wybrać jeden z istniejących węzłów nadrzędnych. Następnie tak jak w przypadku węzła nadrzędnego zdefiniować nowego użytkownika (o nazwie MVADMIN) odpowiedzialnego za mechanizmy replikacji.



**RYСУNEK 15 TWORZENIE WĘZŁA MIGAWKOWEGO**

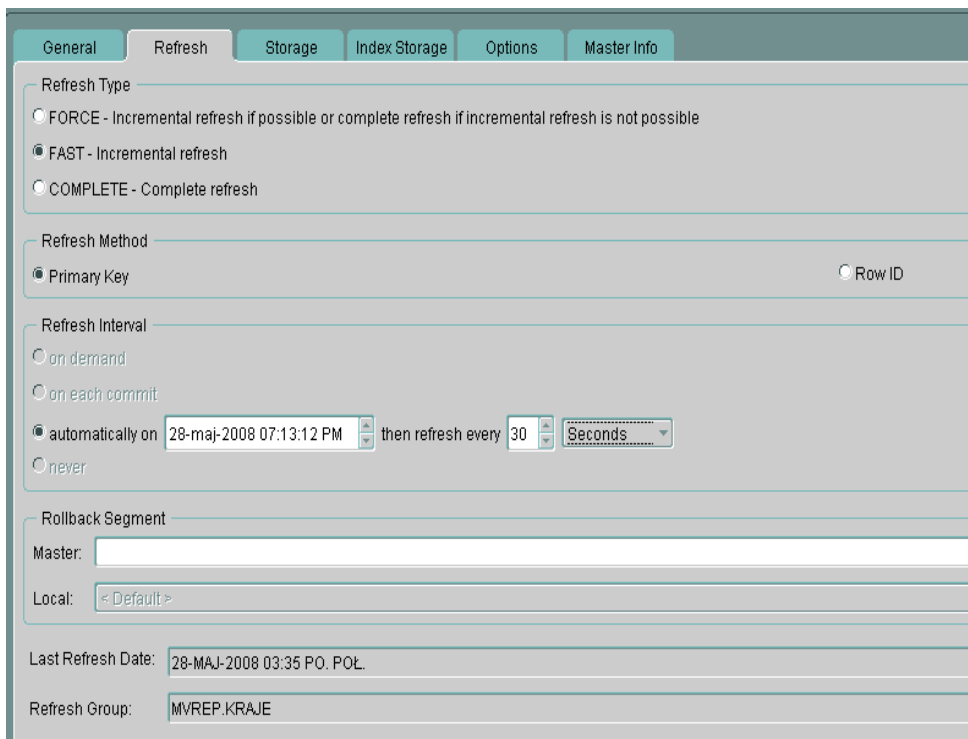
Następnie po stronie bazy krakow1 zdefiniowano grupę migawek (ang. *materialized view site*). W trakcie jej tworzenia określono obiekty z bazy zdalnej, które weszły w skład grupy migawek (Rysunek 16). Następnie wybrano ustawienia replikacji (Rysunek 17). Na sam koniec zostały zmienione ustawienia odświeżania migawek (Rysunek 18).



**RYСУNEK 16 TWORZENIE GRUPY MIGAWEK - WYBIERANIE OBIEKTÓW DO REPLIKOWANIA**



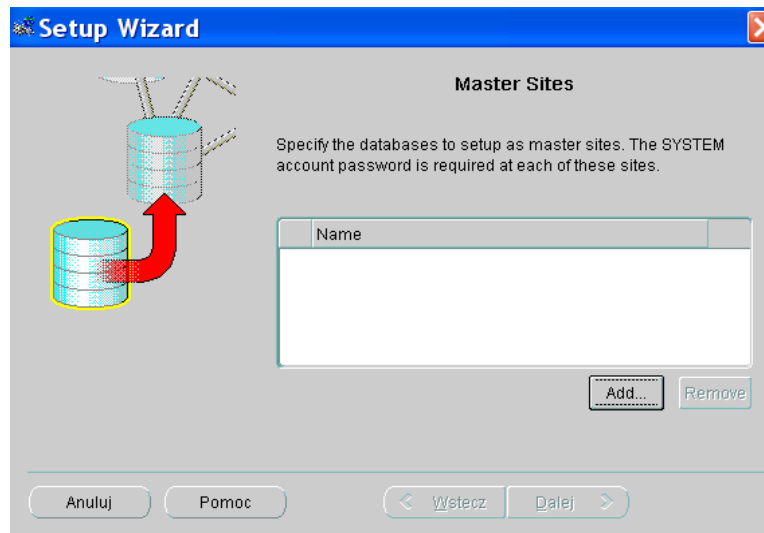
**RYSUNEK 17 USTAWIANIE PARAMETRÓW MIGAWEK**



**RYSUNEK 18 ZMIANA SPOSOBU ODŚWIEŻANIA MIGAWEK**

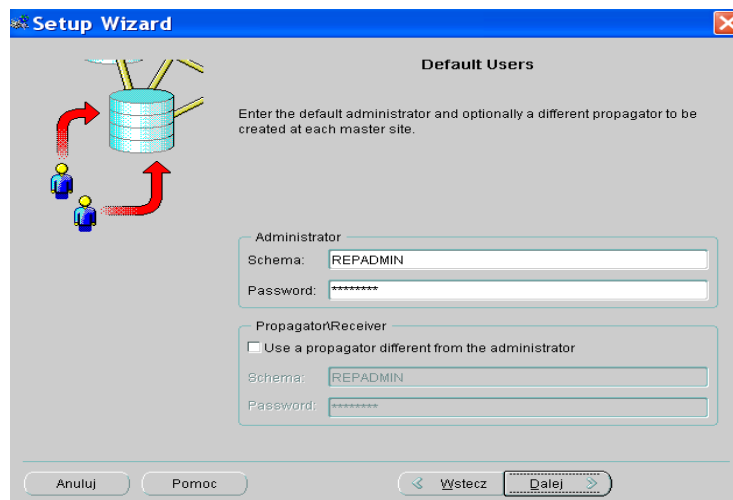
## 4.4. Wykorzystanie mechanizmów replikacji multimaster

W przypadku konfigurowania replikacji multimaster na początku należy stworzyć węzły nadrzędne (Rysunek 19) – są to wszystkie węzły biorące udział w tej replikacji.



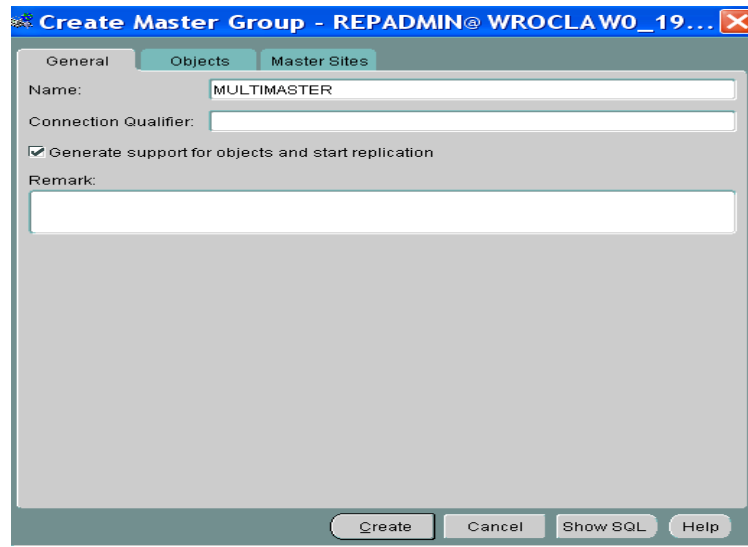
RYSUNEK 19 TWORZENIE WĘZŁÓW NADRZĘDNYCH DLA REPLIKACJI MULTIMASTER

Następnie stworzono użytkowników odpowiedzialnych za replikację (w naszym przypadku zostali wykorzystani ci sami użytkownicy co w przypadku replikacji migawkowej).

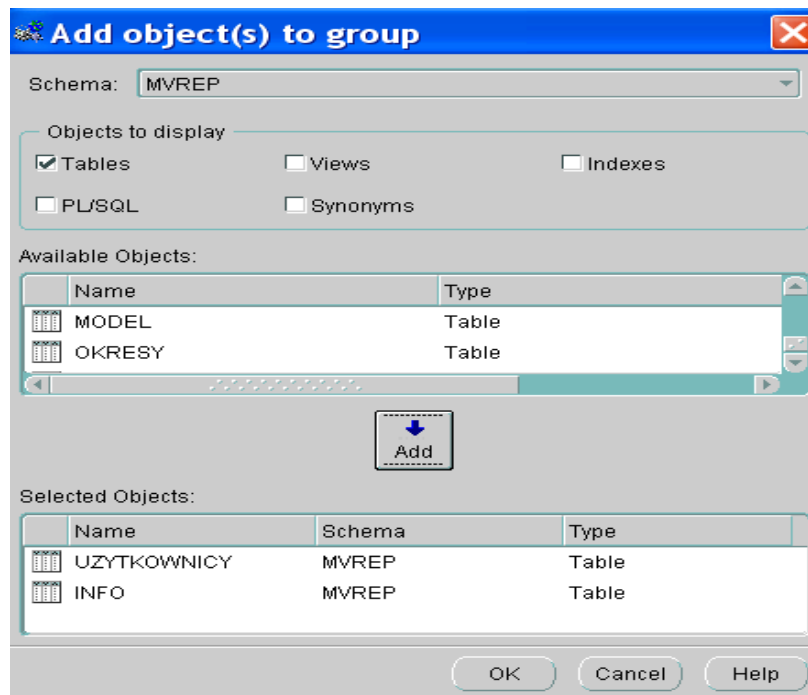


RYSUNEK 20 TWORZENIE UŻYTKOWNIKÓW ODPOWIEDZIALNYCH ZA REPLIKACJĘ MIGAWKOWĄ

Kolejno utworzono we wszystkich węzłach grupy replikacji (Rysunek 21) do których dodano tabele **PRACOWNICY**, **UZYTKOWNICY** i **INFO** (Rysunek 22).



RYSUNEK 21 TWORZENIE GRUPY REPLIKACJI MULTIMASTER

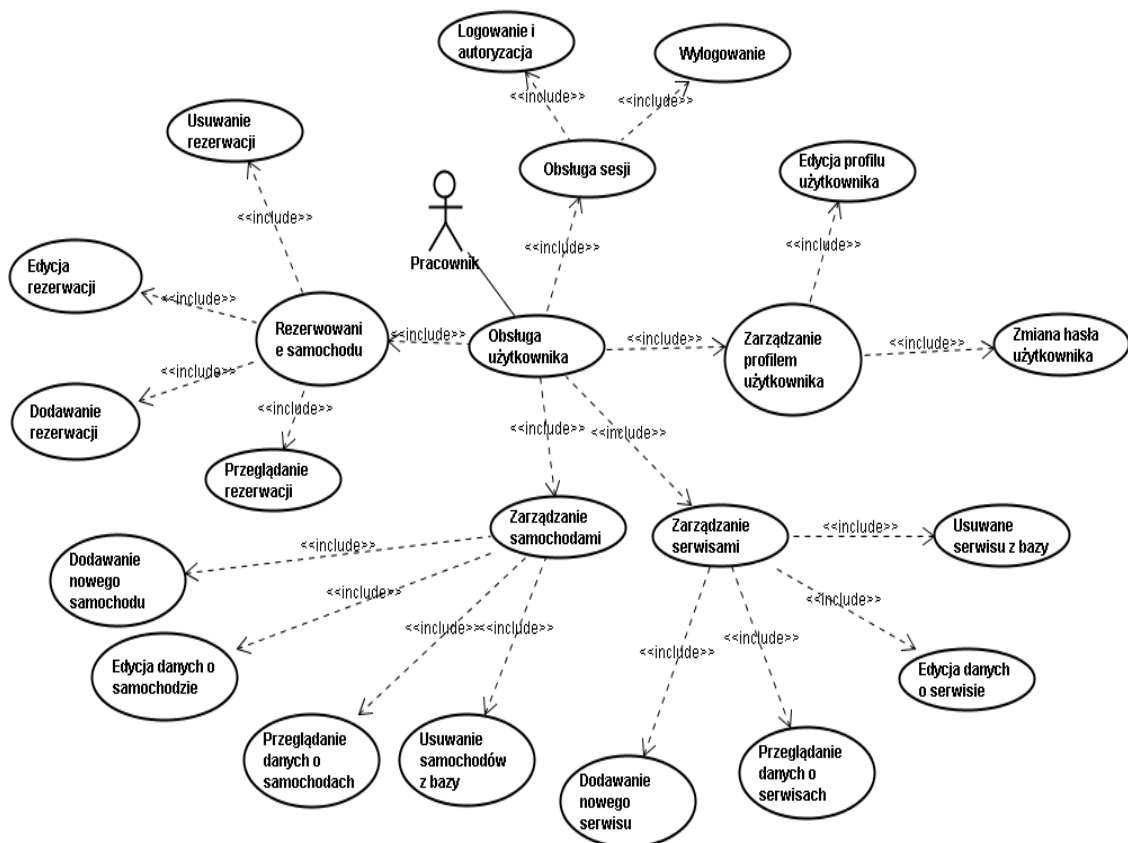


RYSUNEK 22 DODAWANIE OBIEKTÓW DO GRUPY REPLIKACJI MULTIMASTER

## 5. Projekt i implementacja aplikacji klienckiej

### 5.1. Funkcje aplikacji - diagram przypadków użycia

Na poniższym rysunku przedstawiono diagram przypadków użycia stworzonej aplikacji klienckiej. Nie wszystkie przedstawione na diagramie funkcjonalności zostały zaimplementowane, gdyż nie to było celem projektu. Spośród przedstawionych przypadków użycia wybrane zostały tylko wybrane, które zademonstrują działanie mechanizmów replikacji.



RYSUNEK 23 DIAGRAM PRZYPADKÓW UŻYCIA

## 5.2. Realizacja wybranych funkcjonalności

Aplikacja kliencka została zrealizowana w języku Java przy wykorzystaniu możliwości narzędzia NetBeans 6.1.

Poniżej przedstawiony został kod dotyczący zapisywania wprowadzonych zmian do bazy danych rezerwacji samochodów.

```
@Action(enabledProperty = "saveNeededWO", selectedProperty = "recordSelectedWO")
/aby dało się nacisnąć przycisk Save zmienna saveNeededWO musi być ustawiona na true
public Task saveWO() {
    return new SaveTaskWO(getApplication()); /funkcja tworzy nowy obiekt klasy SaveTaskWO czyli
wywołuje zadanie zapisania danych do bazy
}

private class SaveTaskWO extends Task { /klasa dziedziczy po Task, czyli jest zadaniem

    SaveTaskWO(org.jdesktop.application.Application app) {
        super(app);
    }

    @Override
    protected Void doInBackground() {
        try {
            entityManagerWro.getTransaction().commit(); /potwierdzenie zmian w bazie
            entityManagerWro.getTransaction().begin(); /rozpoczęcie nowej transakcji
        } catch (RollbackException rex) { /w przypadku niepowodzenia
            rex.printStackTrace(); /wypisanie błędu
            entityManagerWro.getTransaction().begin(); /rozpoczęcie transakcji
            List<Okresy> merged = new ArrayList<Okresy>(okresyListWO.size());
            for (Okresy u : okresyListWO) {
                merged.add(entityManagerWro.merge(u)); /próba zapisania zmian wszystkich
                                                         elementów z listy okresów i dodanie ich
                                                         do tymczasowej listy
            }
            okresyListWO.clear(); /wyczyszczenie oryginalnej listy
            okresyListWO.addAll(merged); /przepisanie listy tymczasowej do oryginalnej listy
        }
        return null;
    }

    @Override
    protected void finished() {
        setSaveNeededWO(false); /ustawienie zmiennej saveNeededWO na false
    }
}
```

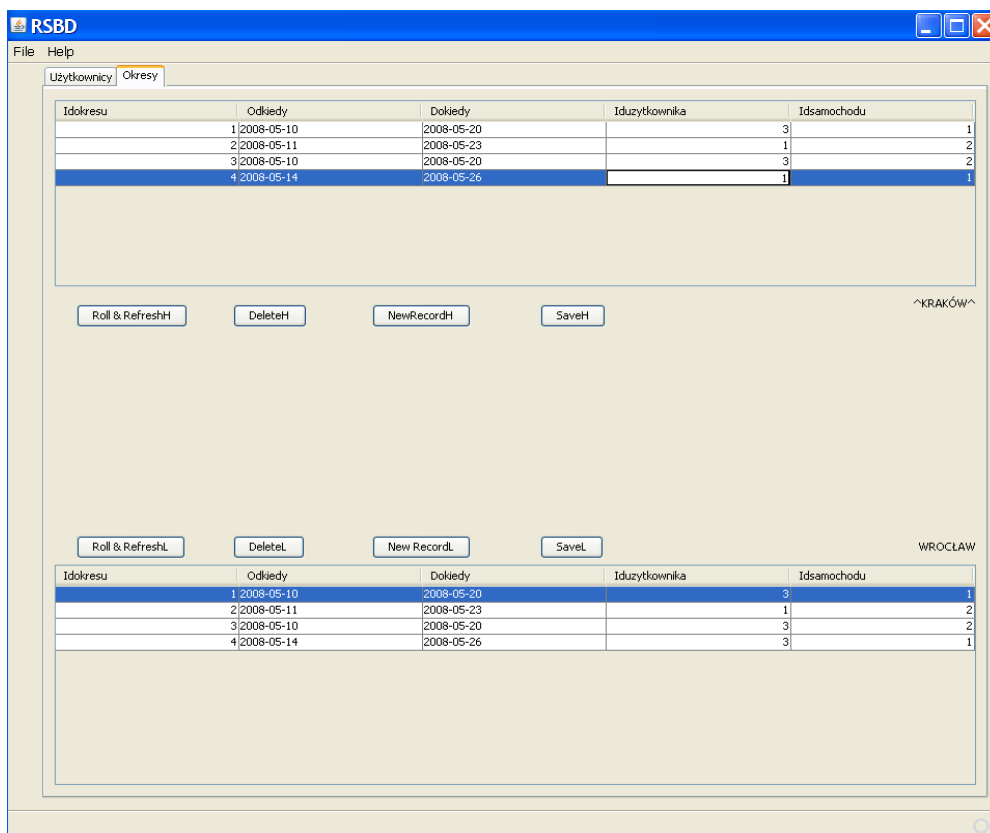


## 6. Wdrożenie i testowanie aplikacji

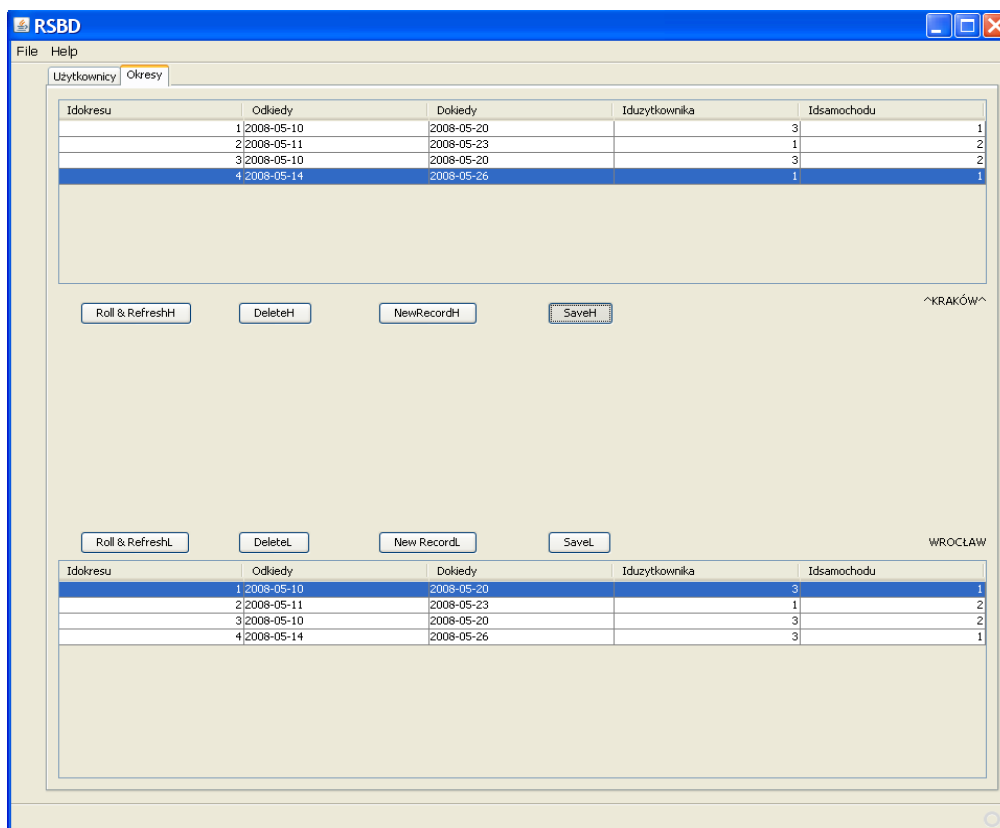
W tej części zostały zamieszczone obrazy działania wprowadzonej aplikacji. Aplikacja działa poprawnie i zgodnie z przewidywaniami.

Najpierw zostało przedstawione działanie mechanizmu replikacji migawkowej na przykładzie tabeli OKRESY zawierającej dane o rezerwacjach. W pierwszej kolejności pokazany został mechanizm propagowania zmian z węzła podrzędnego do nadrzędnego.

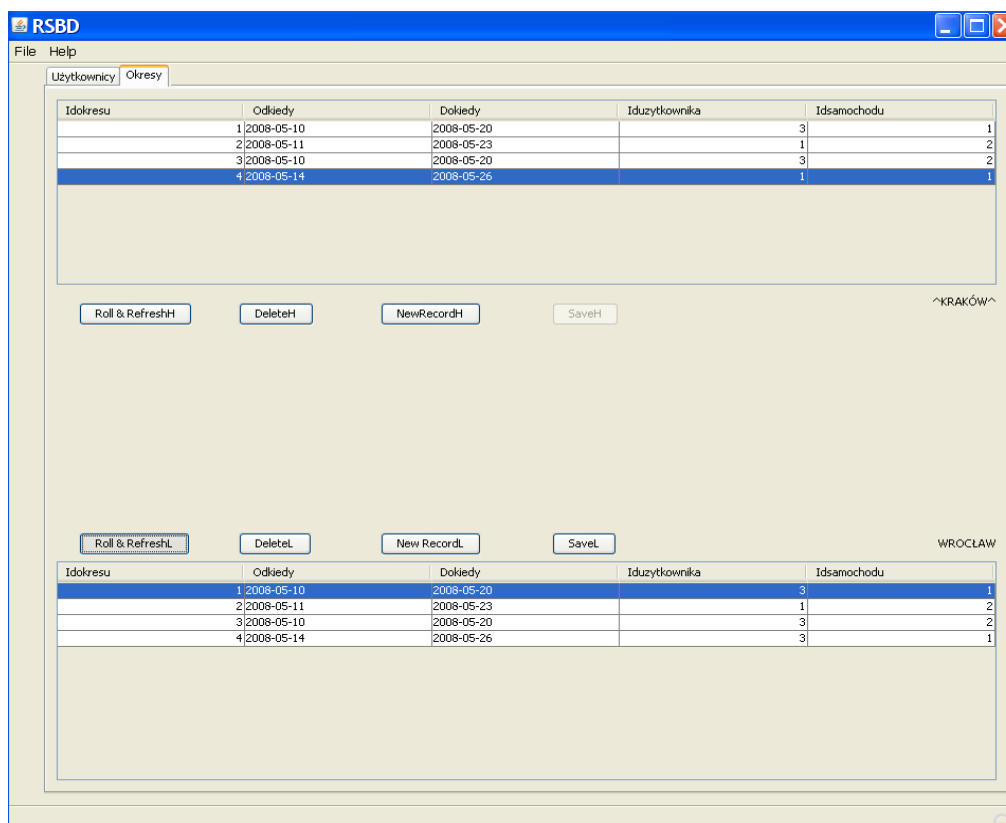
W górnej części ekranu znajdują się dane z bazy w Krakowie a w dolnej części ekranu z bazy we Wrocławiu. Przyciski *Save* służą do zapisywania zmian w bazie, *NewRecord* do dodawania nowych wierszy, *Delete* do usuwania wierszy a *Roll & Refresh* do odświeżania widoku tabeli w aplikacji.



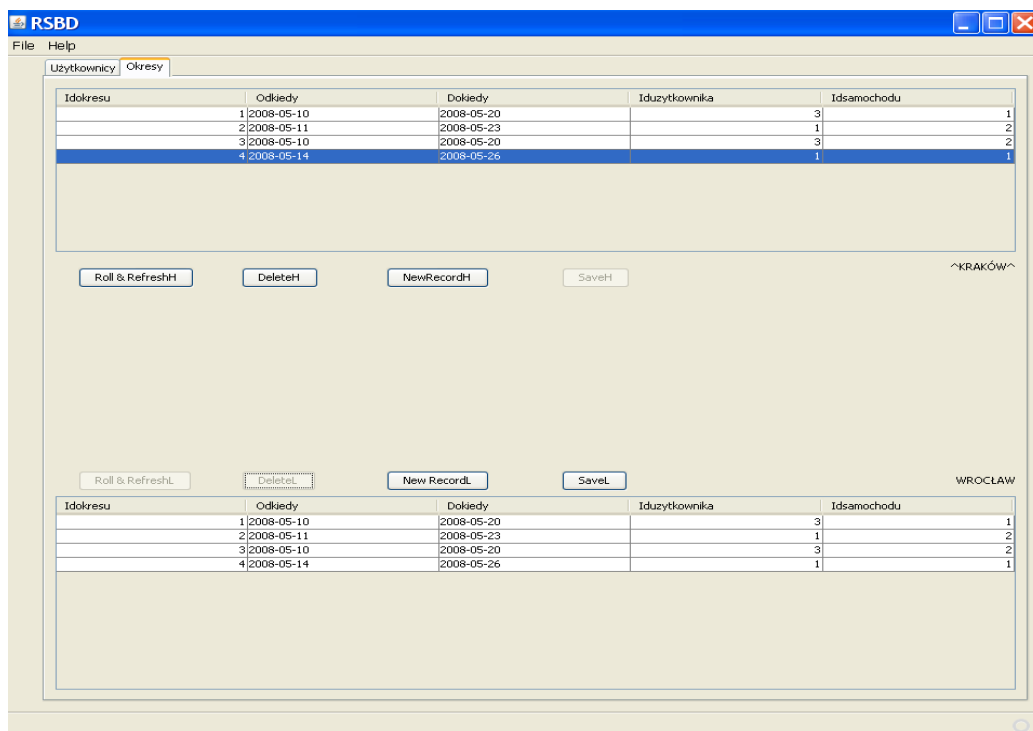
RYSUNEK 24 ZMIANA DANYCH PO STRONIE KRAKOWA (1)



RYSUNEK 25 ZMIANA DANYCH PO STRONIE KRAKOWA (2) – ZATWIERDZENIE NOWYCH DANYCH

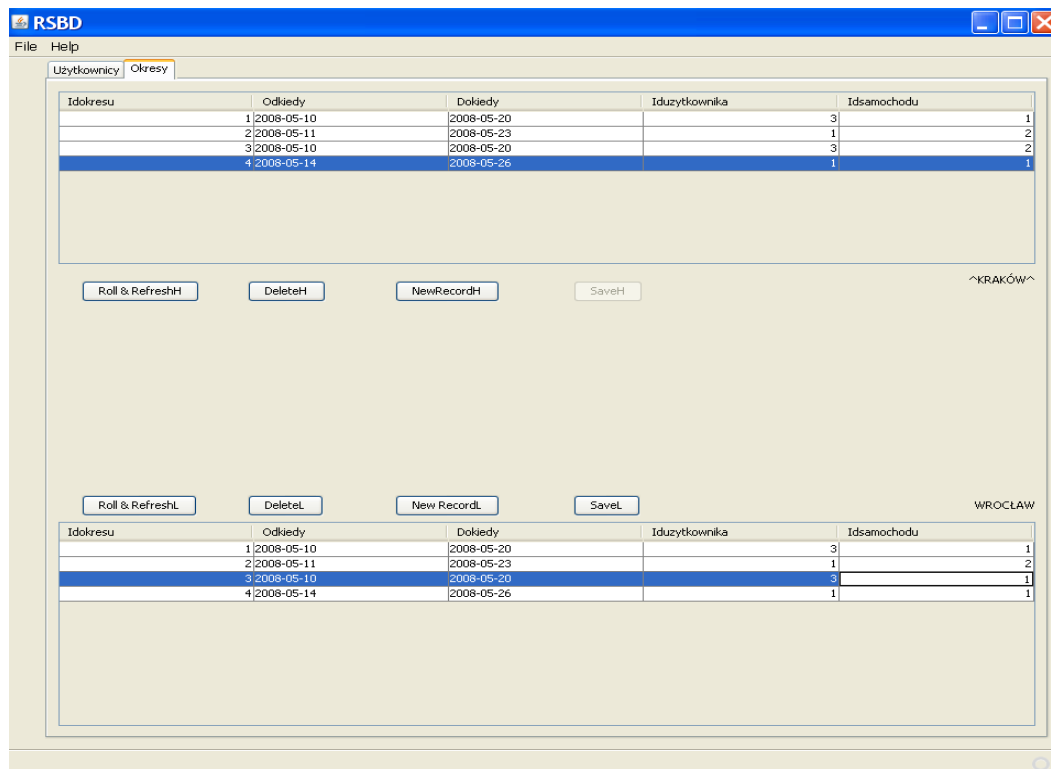


RYSUNEK 26 ZMIANA DANYCH PO STRONIE KRAKOWA (3) – POBRANIE DANYCH Z WĘZŁA NADRZĘDNEGO

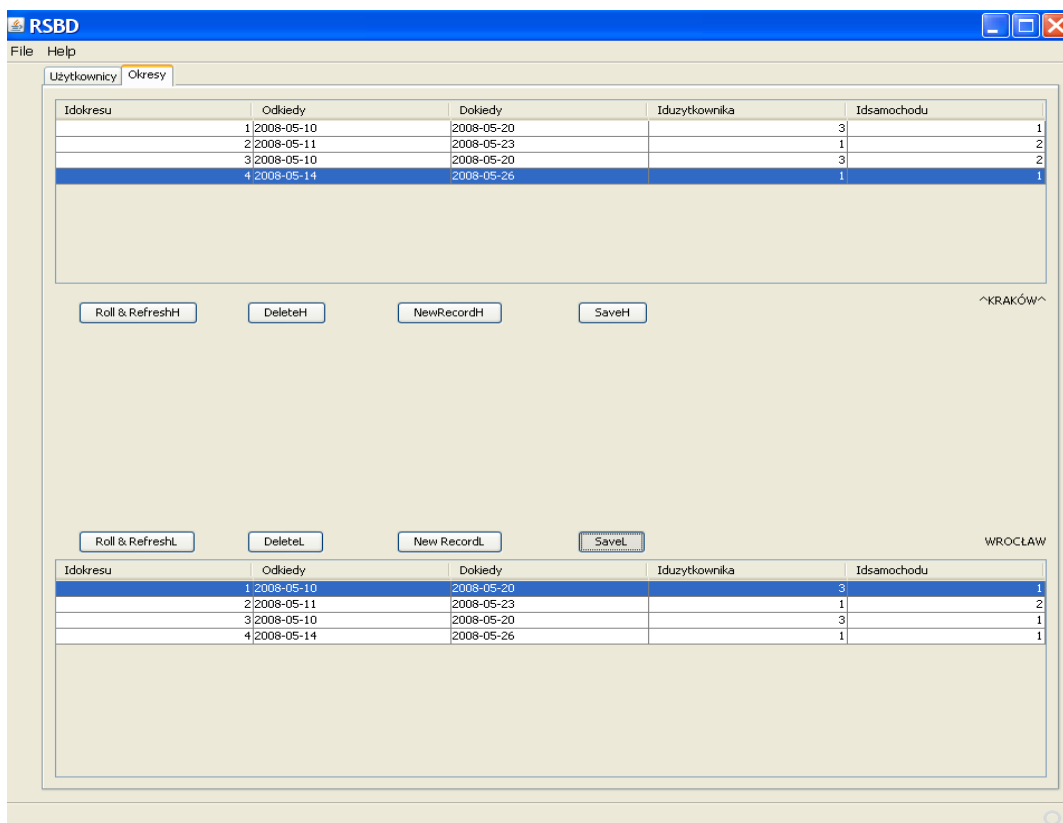


**RYСУNEK 27 ZMIANA DANYCH PO STRONIE KRAKOWA (4) - ZMIANY W WĘZŁE NADRZĘDNYM**

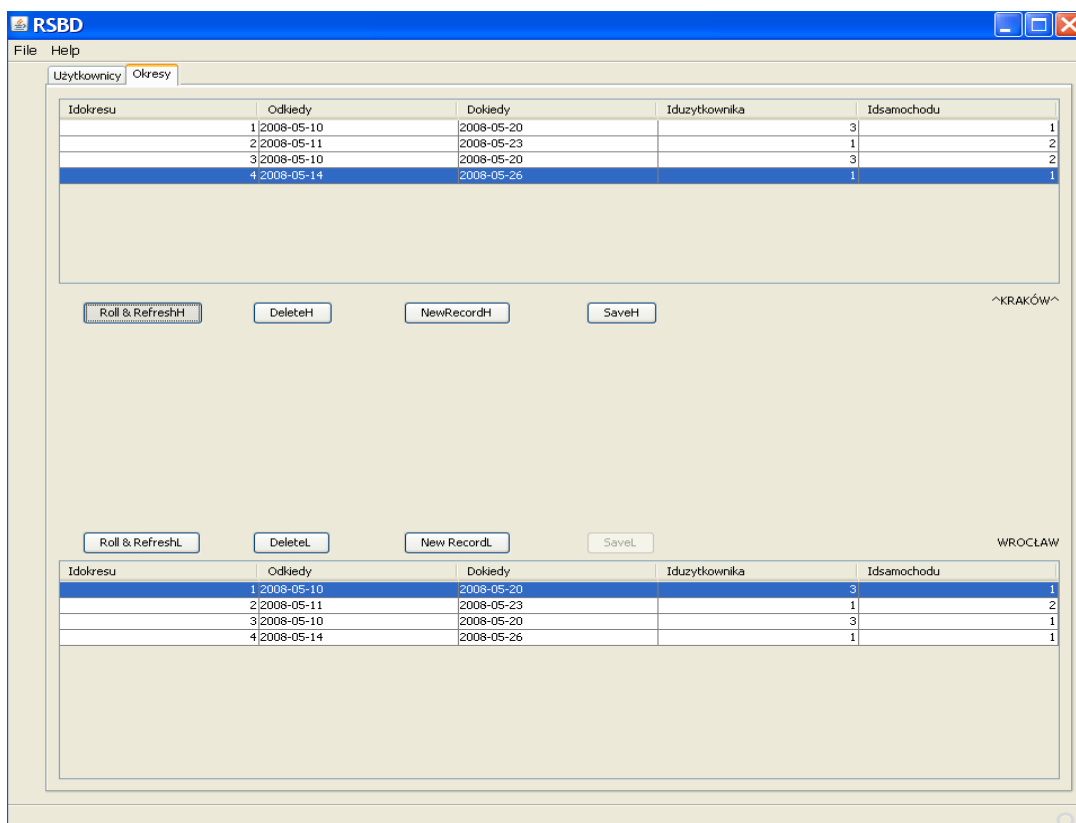
Na kolejnych rysunkach zostało natomiast przedstawione propagowanie zmian z węzła nadrzędnego do węzła podrzędnego.



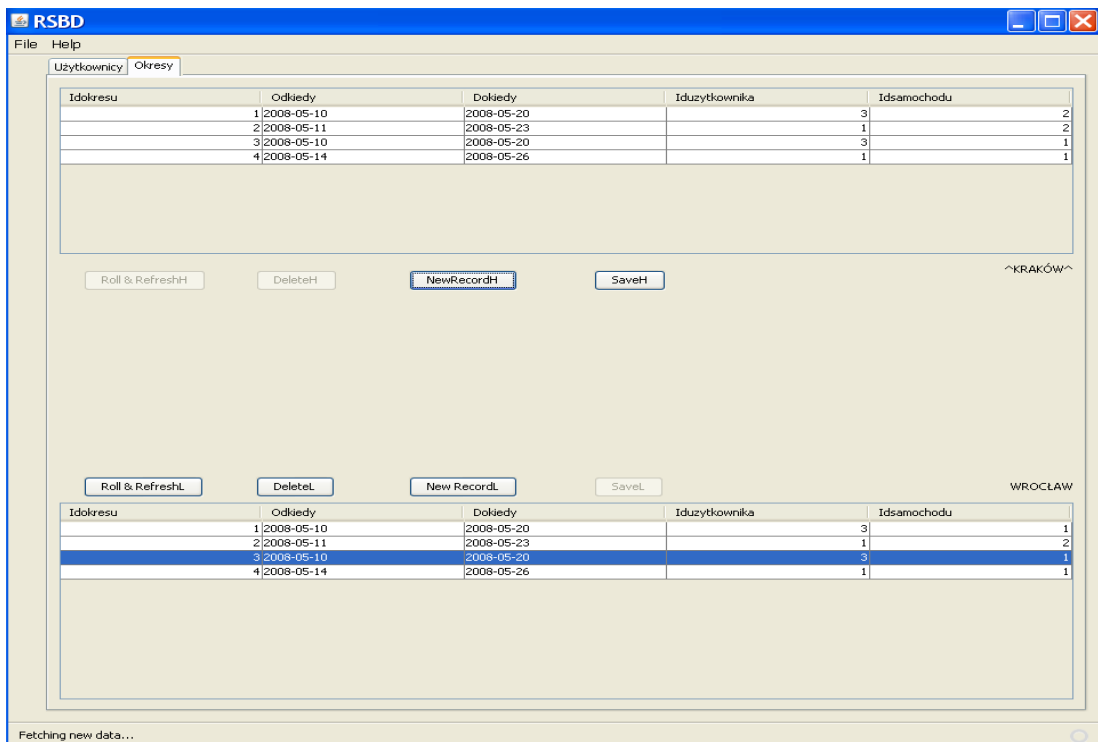
**RYСУNEK 28 ZMIANA DANYCH PO STRONIE WROCLAWIA (1)**



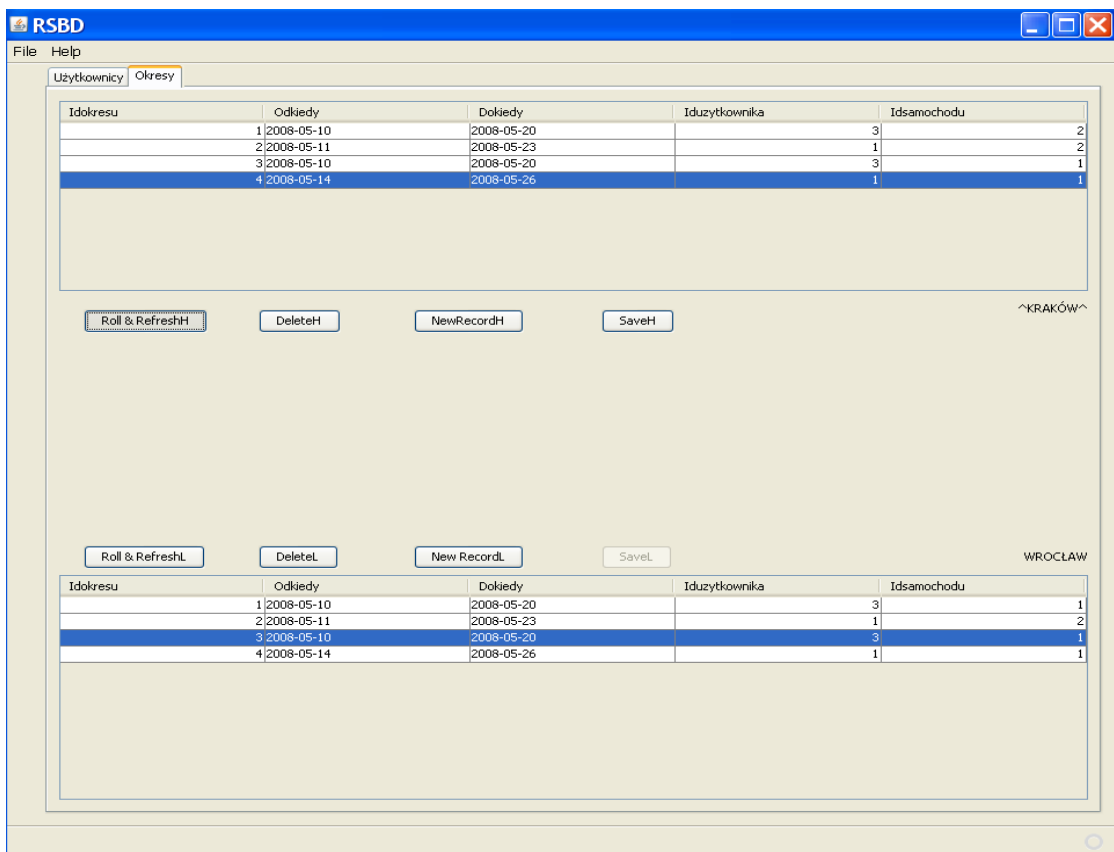
RYSUNEK 29 ZMIANA DANYCH PO STRONIE WROCLAWIA (2) – ZATWIERDZENIE NOWYCH DANYCH



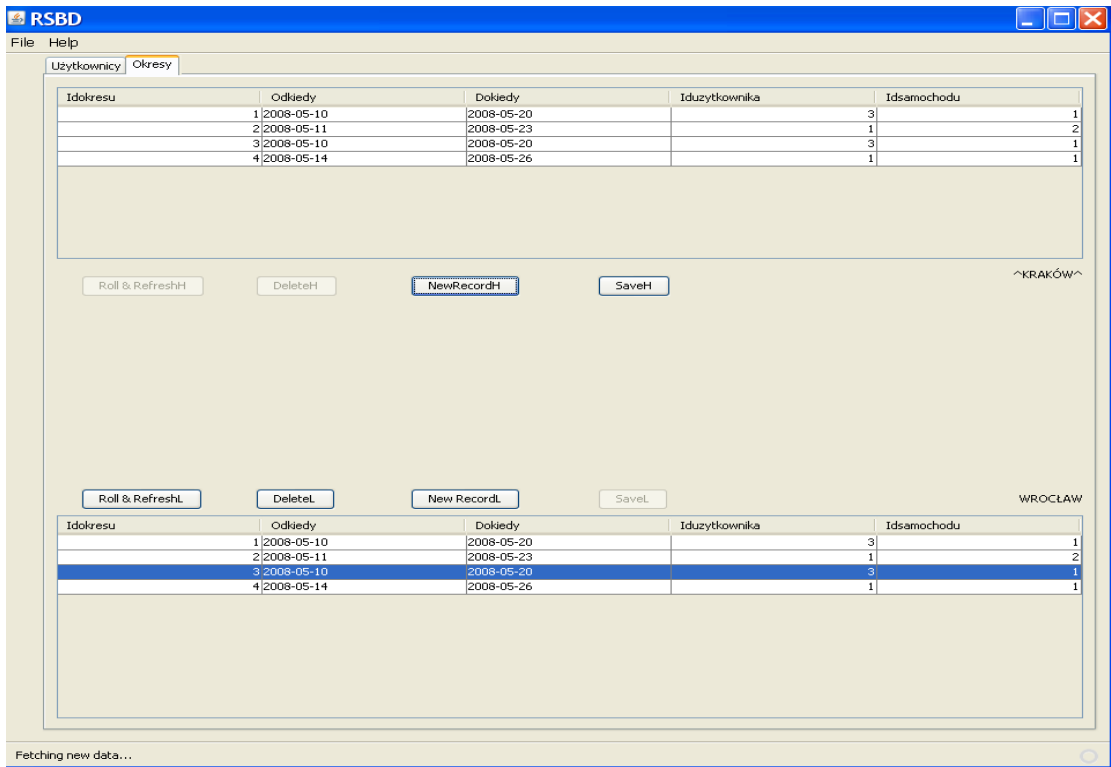
RYSUNEK 30 ZMIANA DANYCH PO STRONIE WROCLAWIA (3) - POBRANIE DANYCH Z WĘZŁA PODRZĘDNEGO



**RYСУNEK 31 ZMIANA DANYCH PO STRONIE WROCLAWIA (4) – OCZEKIWANIE NA ZMIANY W WĘZLE PODRZĘDNYM (ponieważ odświeżanie następuje co 15 sekund)**

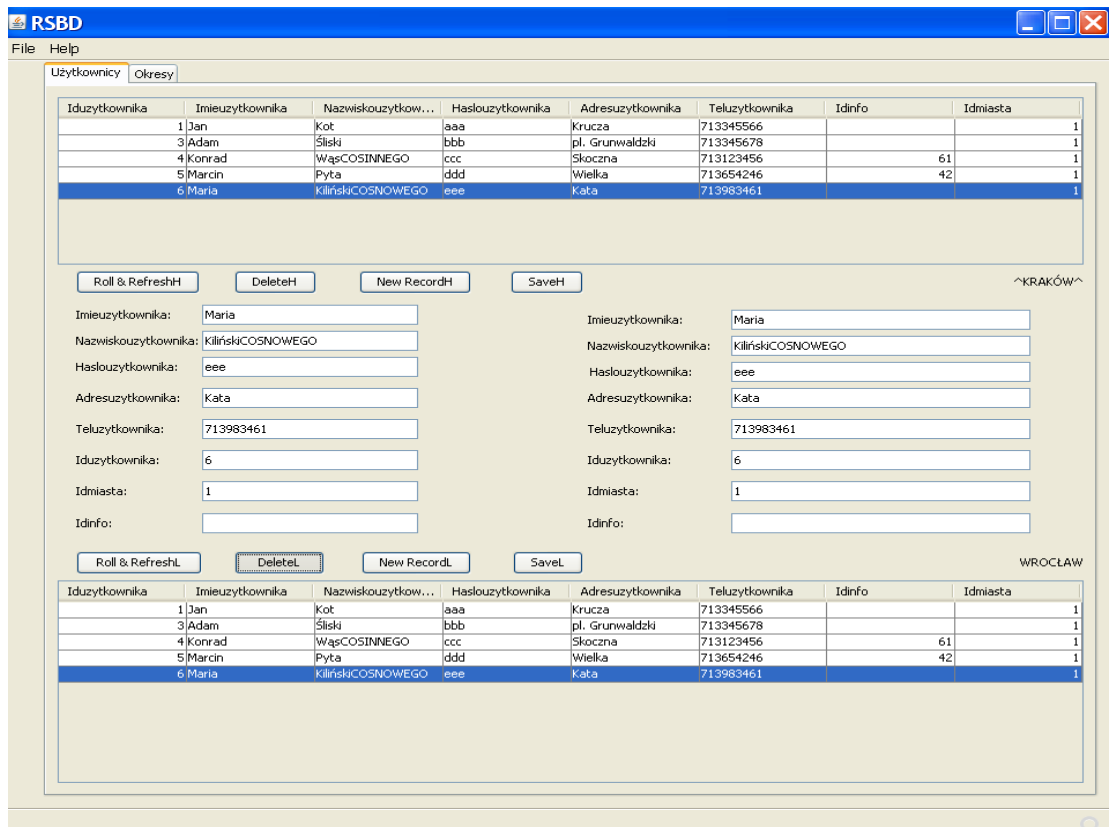


**RYСУNEK 32 ZMIANA DANYCH PO STRONIE WROCLAWIA (5) – PONOWNE POBRANIE DANYCH Z WĘZŁA PODRZĘDNEGO**

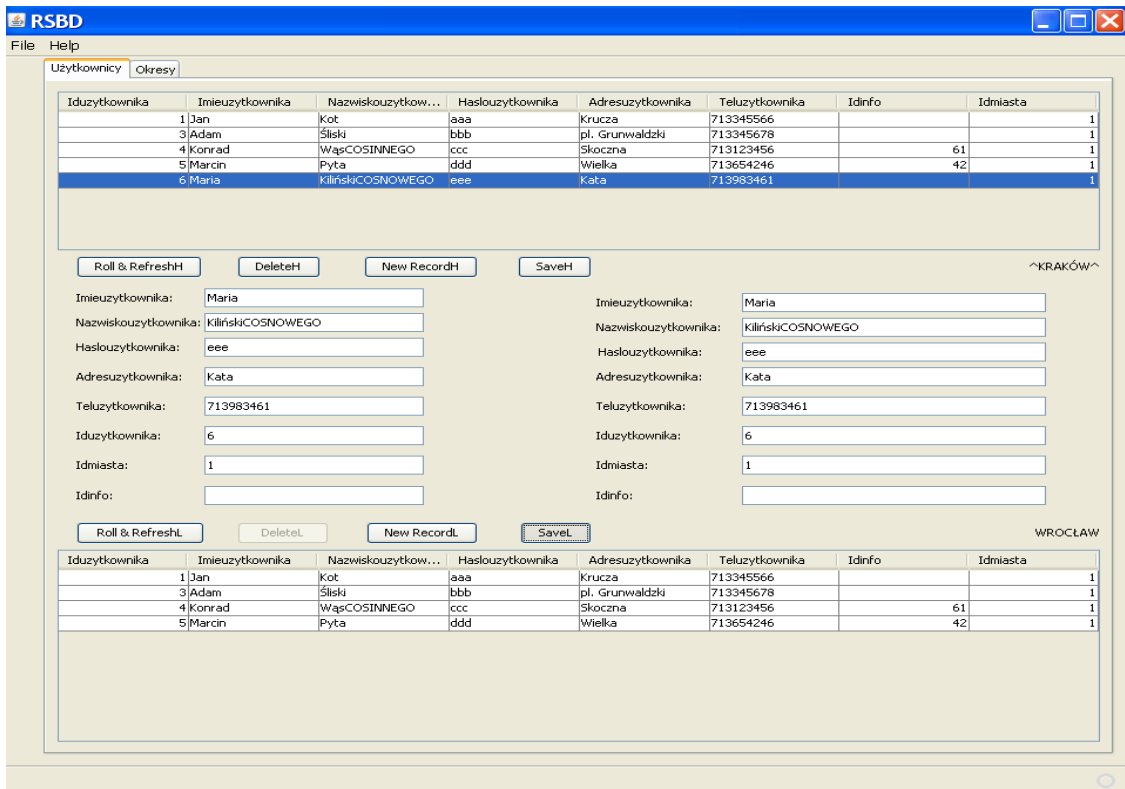


RYSUNEK 33 ZMIANA DANYCH PO STRONIE WROCLAWIA (6) – ZMIANA W WĘZLE PODRZĘDNYM

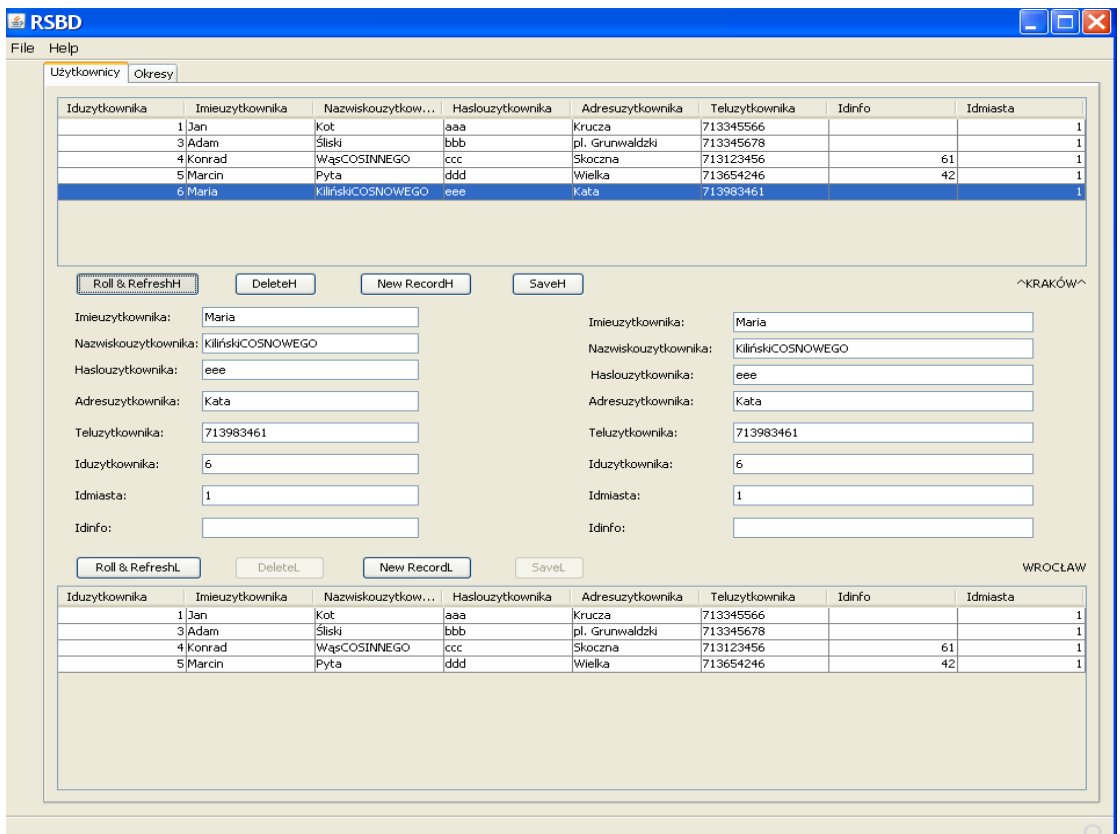
W dalszej części pokazane zostało działanie mechanizmu replikacji multimaster na przykładzie wprowadzania zmian w tabeli UZYTEKOWNICY.



RYSUNEK 34 USUNIĘCIE REKORDU Z TABELI UZYTEKOWNICY (1)



RYSUNEK 35 USUNIĘCIE REKORDU Z TABELI UZYTKOWNICY (2) – ZAPISANIE ZMIAN



RYSUNEK 36 USUNIĘCIE REKORDU Z TABELI UZYTKOWNICY (3) – PONOWNE POBRANIE DANYCH Z WĘZŁA W KRAKOWIE

RSBD

File Help

Uzytkownicy Okresy

Iduzytkownika	Imieuzytownika	Nazwiskouzytkow...	Haslouzytkownika	Adresuzytkownika	Teluzytownika	Idinfo	Idmiasta
1	Jan	Kot	aaa	Krucza	713345566		1
3	Adam	Śliski	bbb	pl. Grunwaldzki	713345678		1
4	Konrad	WąsCOSINNEGO	ccc	Skoczna	713123456	61	1
5	Marcin	Pyta	ddd	Wielka	713654246	42	1

Roll & RefreshH DeleteH New RecordH SaveH ^KRAKÓW^

Imieuzytownika:  Imieuzytownika:

Nazwiskouzytkownika:  Nazwiskouzytkownika:

Haslouzytkownika:  Haslouzytkownika:

Adresuzytkownika:  Adresuzytkownika:

Teluzytownika:  Teluzytownika:

Iduzytkownika:  Iduzytkownika:

Idmiasta:  Idmiasta:

Idinfo:  Idinfo:

Roll & RefreshL DeleteL New RecordL SaveL WROCLAW

Iduzytkownika	Imieuzytownika	Nazwiskouzytkow...	Haslouzytkownika	Adresuzytkownika	Teluzytownika	Idinfo	Idmiasta
1	Jan	Kot	aaa	Krucza	713345566		1
3	Adam	Śliski	bbb	pl. Grunwaldzki	713345678		1
4	Konrad	WąsCOSINNEGO	ccc	Skoczna	713123456	61	1
5	Marcin	Pyta	ddd	Wielka	713654246	42	1

RYSUNEK 37 USUNIĘCIE REKORDU Z TABELI UZYTKOWNICY (3) – ZMIANY W WĘZLE W KRAKOWIE



## 7. Podsumowanie

W trakcie prac nad realizacją projektu udało się wykorzystać mechanizm replikacji migawkowej i replikacji multimaster. Do ich konfiguracji wykorzystane zostały narzędzia graficzne dostarczane wraz z oprogramowaniem Oracle 10g. Obsługa tych narzędzi okazała się nie zbyt skomplikowana. Najwięcej problemów napotkano przy konfiguracji samego środowiska, które jest dość wymagające i w razie problemów generuje komunikaty, które niekoniecznie związane są z problemem, który wystąpił lub po prostu nie wiele mówią użytkownikowi.

Zrealizowana została również aplikacja kliencka w języku Java łącząca się z bazą Oracle przy pomocy Toplinka. Narzędzie NetBeans 6.0.1 dostarcza mechanizmów i wzorców, które w dużym stopniu ułatwiły realizację projektu.

Cały projekt zakończył się sukcesem a działanie programu klienckiego spełnia wszystkie wcześniejsze założenia.

## Literatura

- [1] Bębel B., Wrembel R., *Oracle. Projektowanie rozproszonych baz danych*, Helion, Gliwice 2003
- [2] Pribyl B., Feuerstein S. *Oracle PL/SQL. Wprowadzenie* Helion, Gliwice , 2002
- [3] Bębel B., Wykłady , 2007
- [4] Wrembel R., Wykłady, 2007
- [5] Dokumentacja *Oracle Database 10g Release 2 (10.2.0.1.0)*