

LABORATORIUM
Z ROZPROSZONYCH I OBIEKTOWYCH
SYSTEMÓW BAZ DANYCH

Mechanizm replikacji
w środowisku Oracle -
wyszukiwarka obrazów

AUTORZY:

Michał Czapowski 181225
Grzegorz Grzegorzczak 181121

PROWADZĄCY ZAJĘCIA:

Dr inż. Robert Wójcik, W4/I-6

OCENA PRACY:

Wrocław 2013

Spis treści

1. Wstęp.....	5
1.1. Cel projektu.....	5
1.2. Zakres pracy.....	5
2. Analiza i specyfikacja wymagań.....	6
2.1. Model logiczny i koncepcja działania systemu.....	6
2.2. Wymagania funkcjonalne.....	6
2.2.1. Diagram przypadków użycia.....	6
2.2.2. Scenariusze użycia wybranych przypadków.....	6
2.3. Wymagania nie funkcyjne.....	6
2.3.1. Technologie i narzędzia.....	6
2.3.2. Inne wymagania.....	7
2.3.3. Harmonogram realizacji.....	7
3. Replikacja w systemie baz danych Oracle	8
3.1. Pojęcia i rodzaje replikacji.....	8
3.2. Replikacja migawkowa.....	8
3.3. Pojęcia i rodzaje fragmentacji.....	8
3.4. Fragmentacja pozioma.....	8
3.5. Model replikacji.....	9
4. Konfiguracja środowiska uruchomieniowego.....	10
4.1. Instalacja VirtualBox w systemie Windows.....	10
4.2. Instalacja Linux Oracle.....	12
4.3. Instalacja Oracle Database.....	16
4.4. Konfiguracja połączenia sieciowego.....	20
5. Projekt i implementacja bazy danych.....	22
5.1. Model bazy danych.....	22
5.2. Tworzenie bazy danych.....	22
5.3. Konfiguracja Local Net Service Name.....	23
5.4. Stworzenie dziennika zdarzeń.....	25
5.5. Tworzenie replikacji migawkowej	26
5.6. System zabezpieczeń.....	27

6. Projekt i implementacja aplikacji.....	28
6.1. Model aplikacji.....	28
6.2. Instalacja i przygotowanie środowiska programistycznego.....	28
6.3. Implantacja połączenia z rozproszoną bazą danych.....	29
7. Instalowanie i testowanie.....	31
7.1. Instrukcja obsługi aplikacji.....	31
7.2. Przygotowanie środowiska Windows.....	31
7.3. Przygotowanie środowiska Linux.....	31
7.4. Testowanie mechanizmu replikacji.....	31
7.5. Testowanie aplikacji.....	33
7.6. Wnioski z testów.....	33
8. Podsumowanie	34
Literatura.....	35

Spis rysunków

Ilustracja 1: Diagram przypadków użycia dla dwóch rodzajów użytkowników.....	7
Ilustracja 2: Kierunki replikacji pomiędzy bazami danych.....	10
Ilustracja 3: Instalacja wirtualnej maszyny - wybór systemu operacyjnego.....	11
Ilustracja 4: Instalacja wirtualnej maszyny - rozmiar pamięci RAM.....	12
Ilustracja 5: Instalacja wirtualnego dysku - tworzenie dysku startowego.....	12
Ilustracja 6: Umieszczenie obrazu płyty w wirtualnym napędzie.....	13
Ilustracja 7: Instalacja systemu – przeznaczenie systemu	13
Ilustracja 8: Instalacja systemu - nazwa hosta.....	14
Ilustracja 9: Instalacja systemu - konto administratora.....	14
Ilustracja 10: Instalacja systemu - przestrzeń dysku do instalacji.....	15
Ilustracja 11: Instalacja systemu - lokalizacja boot loadera.....	15
Ilustracja 12: Instalacja systemu - zastosowanie systemu.....	16
Ilustracja 13: Instalacja systemu - licencja Oracle'a.....	16
Ilustracja 14: Instalacja systemu - tworzenie konta użytkownika.....	17
Ilustracja 15: Instalacja systemu - Kdump.....	17
Ilustracja 16: Instalacja bazy danych - aktualizacje.....	23
Ilustracja 17: Instalacja bazy danych - cel instalacji.....	24
Ilustracja 18: Instalacja bazy danych - zastosowanie bazy danych.....	24
Ilustracja 19: Instalacja bazy danych - samodzielna instancja.....	25
Ilustracja 20: Instalacja bazy danych - typ instalacji.....	26
Ilustracja 21: Instalacja bazy danych - szczegóły konfiguracji.....	26
Ilustracja 22: Instalacja bazy danych - podsumowanie testowania systemu.....	27
Ilustracja 23: Konfiguracja portów sieciowych.....	27
Ilustracja 24: Schemat sieci.....	28
Ilustracja 25: Model struktury bazy danych.....	29
Ilustracja 26: Wybór konfiguracji Local Net Service Name.....	30
Ilustracja 27: Dodanie nowej nazwy.....	30
Ilustracja 28: Wprowadzenie nazwy sieciowej.....	31
Ilustracja 29: Wybór protokołu komunikacji z bazą danych.....	31
Ilustracja 30: Wprowadzenie adresu IP pod którym znajduje się serwer bazy danych.....	32
Ilustracja 31: Testowanie połączenia.....	32
Ilustracja 32: Nazwa usługi.....	32
Ilustracja 33: Model okna klienta A.....	35
Ilustracja 34: Model okna klienta B.....	35
Ilustracja 35: Uruchamianie usługi.....	38
Ilustracja 36: Przed replikacji.....	40
Ilustracja 37: Po replikacji.....	40
Ilustracja 38: Działanie aplikacji.....	41

1. Wstęp

1.1. Cel projektu

Celem tego projektu jest zapoznanie się ze sposobem tworzenia aplikacji opartych na rozproszonych bazach danych. W ramach projektu należy wykonać aplikację działającą w oparciu o bazę danych, dodatkowo baza ta musi znajdować się na różnych systemach. Należy poznać i wykorzystać mechanizm replikacji i scalania.

1.2. Zakres pracy

W ramach projektu została wykonana aplikacja pozwalająca na przechowywanie obrazów, oraz wyszukiwanie obrazów. Cały system będzie rozmieszczony na trzech maszynach. Dwie posłużą do wprowadzania danych. Ważne jest to że każda z tych maszyn posiada tylko część zbioru obrazów. Obrazy o nazwach od A do Z mogą być wprowadzane z jednej maszyny, a pozostałe (od P do Z) z drugiej. Trzecia maszyna będzie służyć wyłącznie do przeglądania całego zbioru obrazu. Sama aplikacja powinna umożliwić edycję zawartości bazy danych, oraz przeglądanie jej zawartości, czyli obrazów.

2. Analiza i specyfikacja wymagań

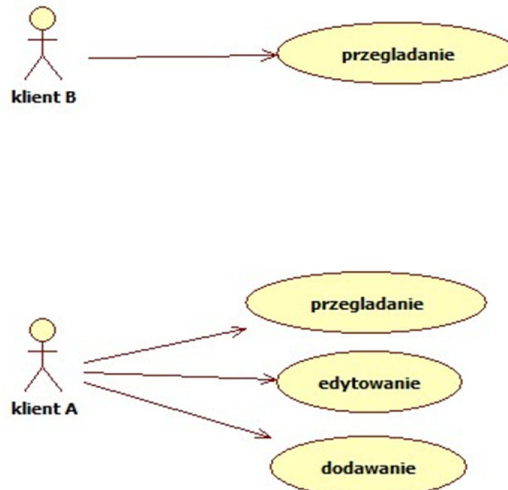
2.1. Model logiczny i koncepcja działania systemu

System ma umożliwiać wprowadzanie obrazów do bazy danych, oraz na przeszukiwanie obiektów które już w niej się znalazły. System może rozpoznawać dwa typy użytkowników. Jeden będzie odczytywał dane a drugi typ zezwoli na ich wprowadzanie.

Dane będą z jednego (logicznie odseparowanego) komputera wprowadzane do bazy, aby następnie zmiana ta została przeniesiona do bazy u klienta zbiorczego. Następnie klient typu B zobaczy zmianę i będzie miał możliwość zobaczenia wprowadzonego obrazu.

2.2. Wymagania funkcjonalne

2.2.1. Diagram przypadków użycia



Ilustracja 1: Diagram przypadków użycia dla dwóch rodzajów użytkowników

Aplikacja musi posiadać dwa typy użytkowników (klientów). Klient A ma możliwość edycji zawartości bazy danych. Może dodawać nowe obrazy, edytować dane już zapisane. Natomiast Klient B jest ograniczony do czytania informacji z bazy danych.

W systemie będzie dwóch klientów typu A i jeden B. Klient B będzie czytał informacje zebrane od pozostałych użytkowników.

2.2.2. Scenariusze użycia wybranych przypadków

Podstawowym przypadkiem jest wyświetlenie listy obrazków znajdujących się na serwerze. Użytkownik uruchamia aplikację, wyświetla się wtedy lokalna lista obrazów, po chwili, gdy baza się zaktualizuje zmiany zostaną wprowadzone. Użytkownik odświeża okno zdanymi i widzi już dane aktualne ze wszystkich serwerów.

Drugim kluczowym scenariuszem jest wprowadzenie nowego obrazu. Użytkownik typu A wybierze opcję „nowy rekord” i następnie wprowadzi nazwę obrazu, oraz jego lokalizację na dysku. Po zatwierdzeniu danych rekord pojawi się w jego lokalnej bazie danych, po chwili zmiana ta zostanie przeniesiona na serwer zbiorczy.

Mniej istotnym przypadkiem użycia jest edycja, aby zmodyfikować już wprowadzony rekord należy zaznaczyć ten wiersz i wybrać opcję edycji. Po zatwierdzeniu zmian nowy wiersz pojawi się w lokalnej bazie, a po chwili zostanie przeniesiony do bazy zbiorczej.

2.3. Wymagania nie funkcyjne

2.3.1. Technologie i narzędzia

Wykorzystaną bazą danych w projekcie będzie baza danych firmy *Oracle*, a jako serwer *Apache*. Interfejs użytkownika zrealizowany zostanie w języku *Java SE* przy użyciu wzorca projektowego *MVP*. Jako interfejs dostępu do bazy danych użyty zostanie framework *Hibernate*. Bazy zostały zainstalowane na dwóch różnych maszynach wirtualnych *Oracle*

VirtualBox 4.3.6. Dla ujednolicenia wybraliśmy jako system operacyjny *Linux* w dystrybucji *Oracle*.

2.3.2. Inne wymagania

Program do wyszukiwania obrazków nie posiada żadnych dodatkowych wymagań.

2.3.3. Harmonogram realizacji

Uwzględniając czas jaki został nam dany na realizację projektu, wyróżniliśmy w projekcie „kamienie milowe”.

Zdobywając informacje na temat bazy danych Oracle odkryliśmy, że nie jest to trywialne zadanie. Dlatego pierwszym krokiem była instalacja i przetestowanie działania samej bazy danych. Wiązało się to z konfiguracją systemu Linux i świadomą instalacją bazy danych Oracle.

Kolejnym ważnym punktem było skonfigurowanie replikacji między węzłami. Na realizację tego przedsięwzięcia zarezerwowaliśmy dużo czasu, ponieważ jest to główny cel tego projektu.

Następnie w wyniku problemów z komunikacją serwerów przez warstwę sieciową trzeba było poprawnie skonfigurować połączenie pomiędzy wirtualnymi maszynami.

Gdy wszystko było gotowe zostało tylko napisać aplikację.

Ostatnim krokiem jest testowanie oprogramowania i tworzenie dokumentacji.

3. Replikacja w systemie baz danych *Oracle*

3.1. Pojęcia i rodzaje replikacji

Replikacja danych w systemach bazodanowych polega na propagowaniu (powielaniu) zmian, lub danych pomiędzy poszczególnymi stacjami (węzłami). Zabieg ten można stosować w wielu różnych celach: zabezpieczanie danych, synchronizacja baz danych. Należy zwrócić uwagę, że Oracle umożliwił replikację zarówno danych (kopiowanie całych rekordów), oraz samych zapytań (to samo zapytanie wykona się na innych węzłach).

Baza danych firmy Oracle daje możliwość kilku różnych rodzajów replikacji. Odróżnia je sposób komunikacji między węzłami infrastruktury.

Podstawowym sposobem jest replikacja migawkowa. Charakteryzuje ją hierarchiczność węzłów. Konfigurując bazy danych należy wyróżnić węzeł nadrzędny (z niego zostaną zreplikowane dane) i węzły podrzędne. Oczywiście zabieg można powtarzać i węzeł podrzędny może być nadrzędnym dla innego węzła. Nadrzędne węzły nie „wiedzą” jakie bazy danych replikują z niej dane. Węzeł główny jedynie udostępnia dziennik zmian, każdej bazie która zechce zreplikować dane.

Zupełnie inaczej zachowuje się replikacja typu multimaster. Wszystkie węzły są równoważne, panuje „płaska” hierarchia - każdy węzeł jest węzłem nadrzędnym. Oznacza to, że zmiana wprowadzona w którymkolwiek węźle zostaje zaaplikowana we wszystkich pozostałych węzłach środowiska, a zreplikowane obiekty są identyczne we wszystkich węzłach - zarówno pod względem struktury, jak i zawartości.

Ostatnim możliwym rozwiązaniem jest replikacja hybrydowa. W takiej infrastrukturze występują zarówno hierarchiczne relacje między węzłami, jak i równoważne połączenia.

3.2. Replikacja migawkowa

Postanowiliśmy wykorzystać do realizacji aplikacji replikację migawkową. Środowisko replikacji migawkowej (ang. materialized view replication) najczęściej tworzy jeden węzeł nadrzędny oraz wiele węzłów migawkowych. Grupa replikacji w węźle nadrzędnym zawiera obiekty (tabele), będące źródłami danych dla migawek, należących do grup replikacji w węzłach migawkowych. Węzłem nadrzędnym może być również węzeł migawkowy, wówczas źródłami danych są zdefiniowane w nim migawki (jest to tzw. architektura wielowarstwowa).

Same migawki mogą być dwójakiego typu. ROWID – wykorzystuje do identyfikacji rekordów fizyczny adres pamięci na dysku. Rozwiązanie to może stać się problematyczne, jeśli zaczniemy przemieszczać tabele. Zostaną wtedy zmienione adresy komórek. Drugi typ to *primary key* – do identyfikacji wykorzystuje klucz główny tabeli. Jest to również domyślna konfiguracja replikacji migawkowej. Sam dystrybutor zaleca odchodzenie od pierwszego rozwiązania i zaleca stosowanie *primary key*.

Najbardziej charakterystycznym pojęciem dla tego rodzaju replikacji jest - dziennik migawki. Jest to plik w którym są zapisywane wszystkie zmiany dokonywane na replikowanej tabeli (tabela bazowa dziennika). Każda tabela posiada swój własny dziennik. W takim pliku znajdują się wszystkie wykonane zapytania modyfikujące zawartość tabeli. Dzięki temu mechanizmowi można replikować tylko zapytania, w ten sposób zmniejszamy obciążenie łącza pomiędzy węzłami. To rozwiązanie ma oczywistą wadę, jeśli zapytanie może dać inny wynik na innej bazie (np. funkcja losująca).

3.3. Pojęcia i rodzaje fragmentacji

Fragmentacja polega na podziale tabeli na obszary niezależnie replikowane. Baza danych Oracle umożliwiają kilka rodzajów (technik) fragmentacji.

Fragmentacja pozioma polega na dzieleniu zawartości (rekordów) tabeli względem klucza głównego. Powstałe podzbiory posiadają taką samą jak przed podziałem listę atrybutów. Różnią się liczbą posiadanych rekordów.

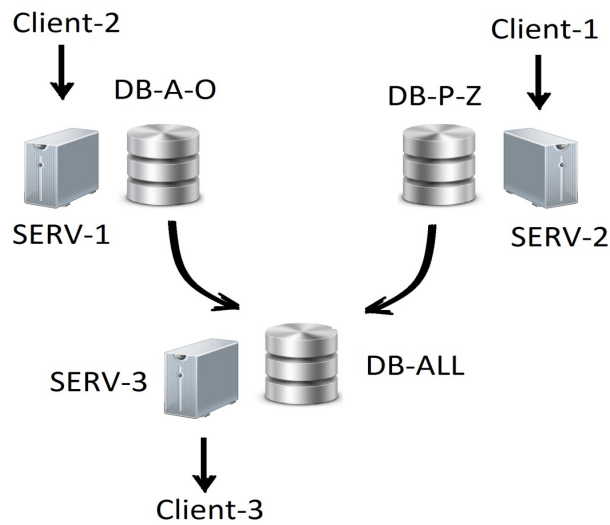
Pionowy podział dzieli tabele ze względu na atrybuty. Powstałe podzbiory posiadają różne zbiory atrybutów, ale po scaleniu odtworzymy wszystkie atrybuty tabeli bazowej.

Ostatnim udostępnionym przez firmę Oracle trybem jest replikacja mieszana. Podaje się w niej zarówno oś podziału pionowego, jak i sposób poziomego podziału.

3.4. Fragmentacja pozioma

W naszym systemie informacje będą podzielone ze względu na nazwę przechowywanego obrazu. Oznacza to fragmentację poziomą. Dwie bazy danych posiadają informację o obrazach o nazwach z określonego przedziału. Są to nazwy od A-O i od P-Z.

3.5. Model replikacji



Ilustracja 2: Kierunki replikacji pomiędzy bazami danych

Powyższy rysunek przedstawia przepływ danych. Dane (obrazy) mogą zostać dodane do bazy danych tylko przez określonych użytkowników z maszyn SERV-1 i 2, na których są zainstalowane bazy DB-A-O i DB-P-Z. Wszystkie dodane do nich obiekty są replikowane do bazy zbiorczej DB-ALL. Znajduje się ona na serwerze SERV-3 i tylko z niej mogą być odczytywane wszystkie dane .

4. Konfiguracja środowiska uruchomieniowego

4.1. Instalacja *VirtualBox* w systemie *Windows*

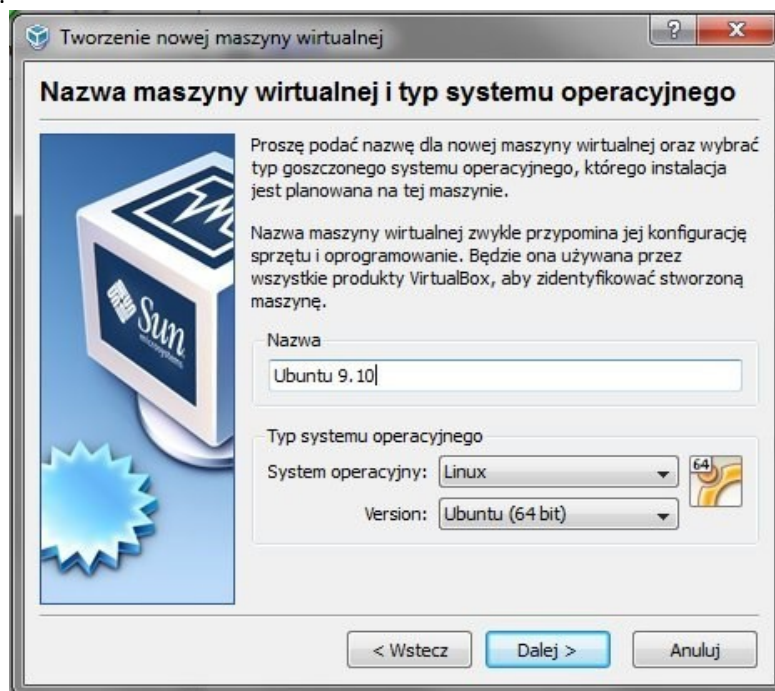
Zdecydowaliśmy się na instalację systemów na wirtualnych maszynach, ponieważ ułatwiło to proces instalacji. Ominęliśmy w ten sposób proces instalacji systemów na różnych fizycznych maszynach.

Aby zainstalować wirtualną maszynę należy najpierw pobrać plik instalacyjny *VirtualBox* ze strony: <http://www.virtualbox.org/wiki/Downloads>.

Instalowanie programu wymaga jedynie potwierdzenia licencji. Ważne jest aby instalując ten program przygotować bardzo dużo wolnego miejsca na dysku. Sam program po zainstalowaniu nie zajmuje dużo pamięci, ale pojedyncza wirtualna maszyna wymaga znacznej ilości miejsca.

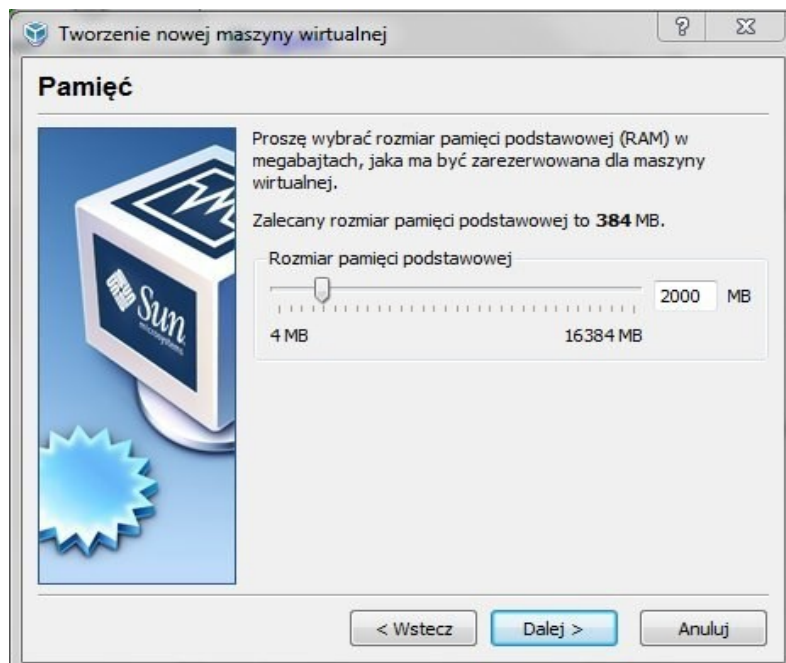
Nim stworzymy wirtualną maszynę warto pobrać obraz płyty instalacyjnej systemu który chcemy zainstalować. W naszym przypadku będzie to *Linux Oracle 6* ze strony producenta: <http://www.oracle.com/technetwork/server-storage/linux/downloads>.

Pop zainstalowaniu i uruchomieniu programu wybieramy niebieską ikonę z napisem „Nowy”. W ten sposób tworzymy nową wirtualną maszynę. Pierwsze pytanie dotyczy nazwy partycji jaką tworzymy.



Ilustracja 3: Instalacja wirtualnej maszyny - wybór systemu operacyjnego

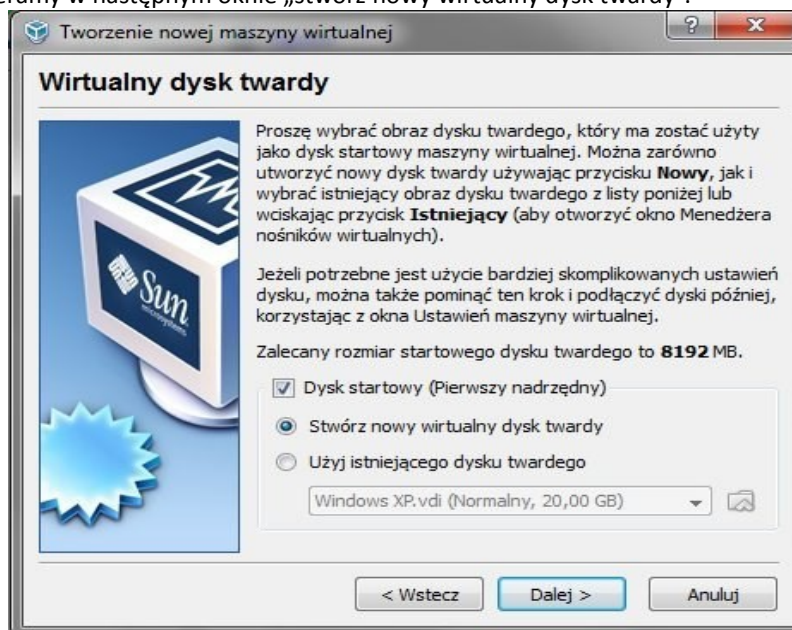
Program podpowie nazwę kiedy wybierzemy jaki system i jaką wersję chcemy zainstalować. W kolejnym kroku jesteśmy pytani o rozmiar pamięci RAM którą oddamy pod działanie tej maszyny wirtualnej.



Ilustracja 4: Instalacja wirtualnej maszyny - rozmiar pamięci RAM

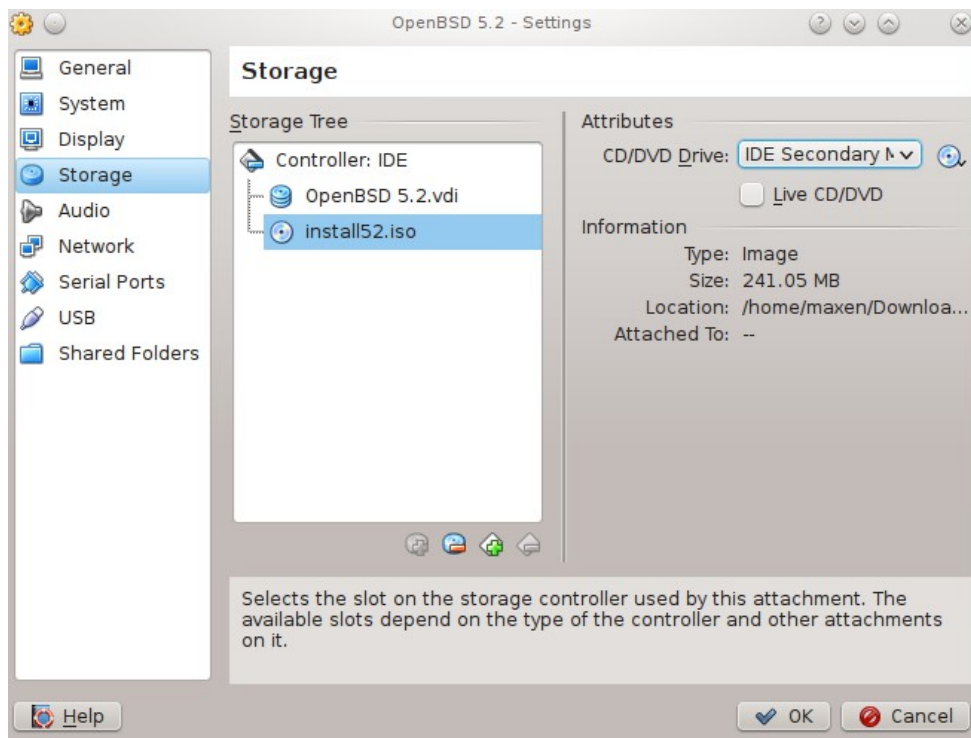
Warto zwrócić uwagę, że poza uruchomioną wirtualną maszyną będzie również działał system zainstalowany na fizycznej maszynie, mało tego pewnie będą działać się w tle jakieś dodatkowe operacje. Jeśli stosujemy *Windows 7* lub wyższą wersję, polecam wyłączyć na czas pracy z *VirtualBox'em* automatyczne pobieranie aktualizacji. Nie można też przydzielić zbyt mało pamięci, gdyż system nie będzie działał poprawnie. Zalecam (jeśli jest taka możliwość) oddanie wirtualnej maszynie 1024 MB RAMu.

Tworzymy pierwszy wirtualny dysk, i nie posiadamy jeszcze żadnego gotowego dysku, dlatego wybieramy w następnym oknie „stwórz nowy wirtualny dysk twardy”.



Ilustracja 5: Instalacja wirtualnego dysku - tworzenie dysku startowego

Po zatwierdzeniu tego wyboru stworzyliśmy wirtualną maszynę. Teraz należy zainstalować na niej system. Aby to zrobić trzeba umieścić pobrany obraz płyty *Linux Oracle* w wirtualnym napędzie. Klikamy naszą nową maszynę i wybieramy żółtą ikonę „ustawienia”.



Ilustracja 6: Umieszczenie obrazu płyty w wirtualnym napędzie

W zakładce „storage” tworzymy nowy wirtualny napęd. Podczas tworzenia podajemy lokalizację obrazu płyty instalacyjnej. Teraz gdy w oknie głównym *VirtualBox'a* wybierzemy zieloną strzałkę z napisem „uruchom” zainicjujemy rozruch maszyny wirtualnej z obrazu płyty.

4.2. Instalacja Linux Oracle

Instalacja bazy danych w środowisku Linux Ubuntu jest wyjątkowo skomplikowana. Pociąga za sobą konieczność zmiany wielu zmiennych środowiskowych, a nawet niektórych wartości rejestrów. Próby skonfigurowania systemu Linux w dystrybucji Ubuntu zakończyła się niepowodzeniem. Zdecydowaliśmy się na instalację dystrybucji *Linux'a* stworzonej przez firmę *Oracle*. Obraz płyty niezbędny do instalacji na wirtualnej maszynie jest dostępna ze strony dystrybutora.

Sama instalacja systemu wyglądem przypomina typową instalację systemu *Linux*.



Ilustracja 7: Instalacja systemu – przeznaczenie systemu

Wybieramy opcję pierwszą, ponieważ nasz system nie będzie miał przeznaczenia korporacyjnego.



Ilustracja 8: Instalacja systemu - nazwa hosta

Nazwa hosta służy do identyfikacji komputera w sieci. Sugeruję nazwę „oracle-1”. Albo inną odpowiadającą przeznaczeniu tego komputera.

Kolejnym krokiem jest wybór strefy czasowej, dla ułatwienia dodano mapę stref czasowych.

Następnie system chce utworzyć konto administratora.



Ilustracja 9: Instalacja systemu - konto administratora

Warto zapisać sobie hasło, ponieważ przed utworzeniem konta użytkownika to hasło będzie potrzebne do logowania.



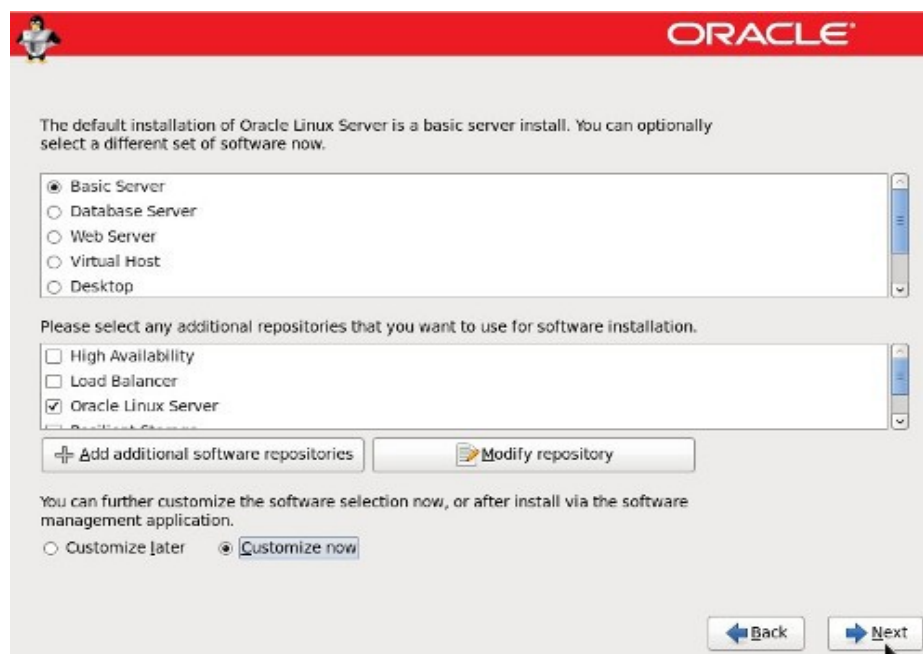
Ilustracja 10: Instalacja systemu - przestrzeń dysku do instalacji

Instalator pyta gdzie chcemy zainstalować system, na wirtualnej maszynie nie ma żadnego systemu więc bez wahania można wybrać pierwszą opcję. Następnie instalator pokaże jak wygląda podział dysku na partycje przed instalacją.



Ilustracja 11: Instalacja systemu - lokalizacja boot loadera

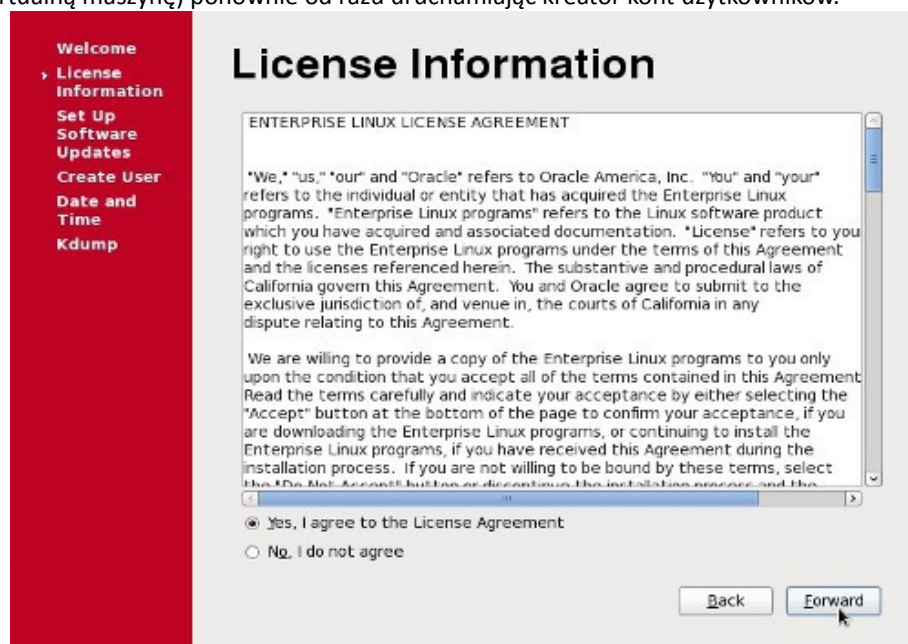
Domyślną lokalizacją jest katalog /dev, zalecam pozostawienie tego wyboru bez zmian.



Ilustracja 12: Instalacja systemu - zastosowanie systemu

Przedostatnie pytanie instalatora dotyczy przeznaczenia systemu. Zamierzamy korzystać z systemu jako zwykłego serwera. Ponieważ wtedy możemy samodzielnie zainstalować bazę danych. Po zatwierdzeniu instalator zapyta tylko z jakich narzędzi chcemy korzystać. Nie potrzebujemy żadnych specjalnych programów, dlatego warto sprawdzić, czy wśród zaznaczonych nie ma jakiś dużych programów, które mogłyby niepotrzebnie zająć miejsce na dysku.

System przeprowadzi instalację, potrwa to kilka minut. Gdy skończy uruchomi komputer (wirtualną maszynę) ponownie od razu uruchamiając kreator kont użytkowników.



Ilustracja 13: Instalacja systemu - licencja Oracle'a

Nim stworzymy jakiegokolwiek konto, każdy użytkownik musi zatwierdzić warunki licencji dystrybutora. Potem może przystąpić do podania loginu i hasła klienta.

Ilustracja 14: Instalacja systemu - tworzenie konta użytkownika

Ze stworzeniem konta łączy się podanie aktualnej godziny i dnia do kalendarza użytkownika.

Ostatnim rokiem jest ustawienie *Kdump*. Usługa pozwala na powstanie systemu na wypadek awarii. Wystarczy nam podstawowe ustawienia tej opcji.

Ilustracja 15: Instalacja systemu - Kdump

Zakończenie pracy kreatora skutkuje przeniesieniem do widoku logowania. Kiedy podamy login i hasło znajdziemy się na pulpicie w systemie *Linux Oracle*.

4.3. Instalacja Oracle Database

Instalację bazy danych na systemie należy rozpocząć od pobrania pliku instalacyjnego ze strony Oracle'a:

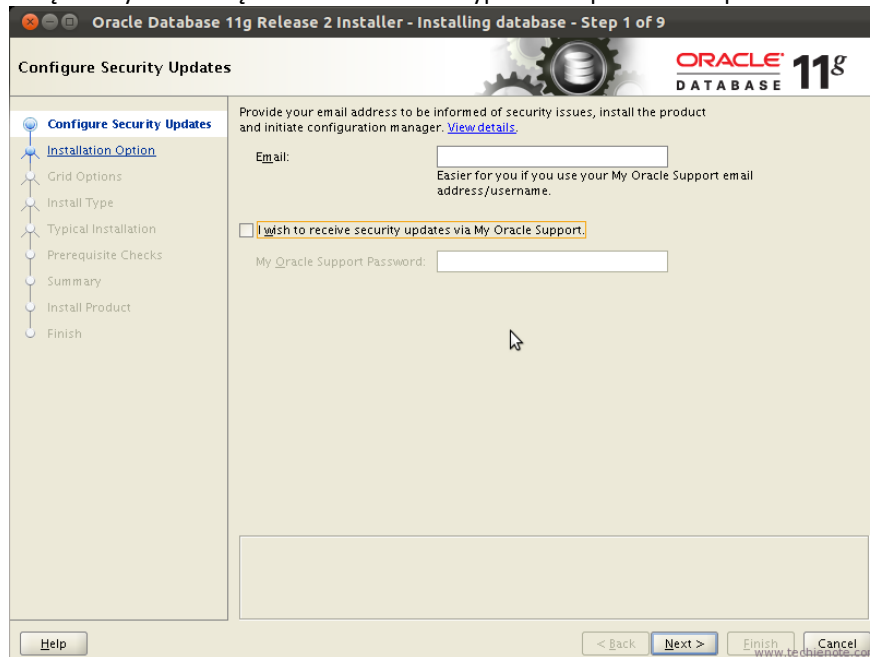
<http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html> .

Użyta do projektu została wersja *11g R2* (wersja *11.2.0.2.0*). Należy oczywiście zwrócić uwagę na architekturę procesora (x86, x64). Aby dostępne były zaawansowane prace z serwerem bazodanowym musimy wybrać dystrybucję *Enterprise*, nie *Express*.

Pobrany plik należy rozpakować, a następnie uruchomić plik instalacyjny:

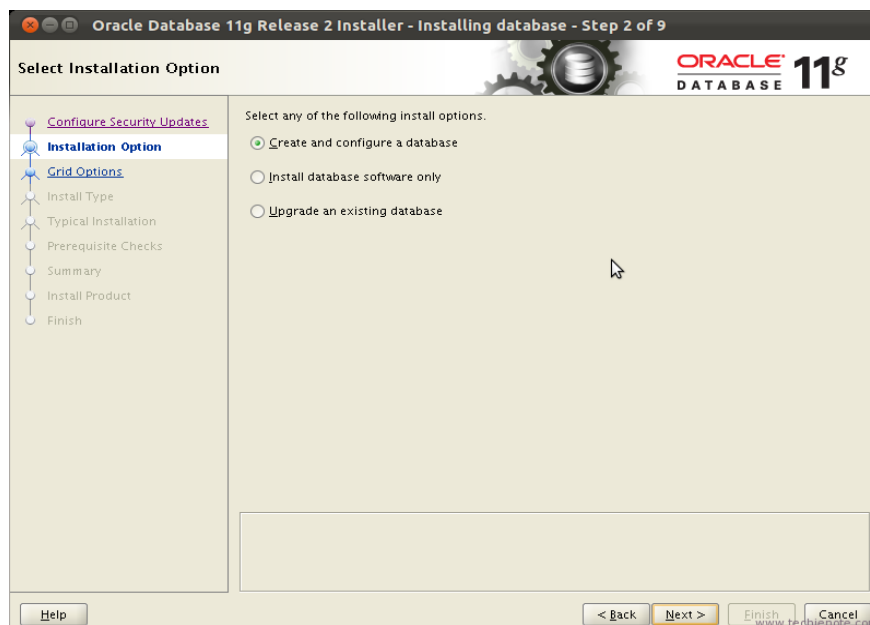
```
$ ./runInstaller -ignoreSysPrereqs
```


Uruchomiony zostanie asystent instalacji, który ułatwi konfigurację bazy danych, dzięki czemu nie będziemy musieli ręcznie uruchamiać skryptów z odpowiednimi parametrami.



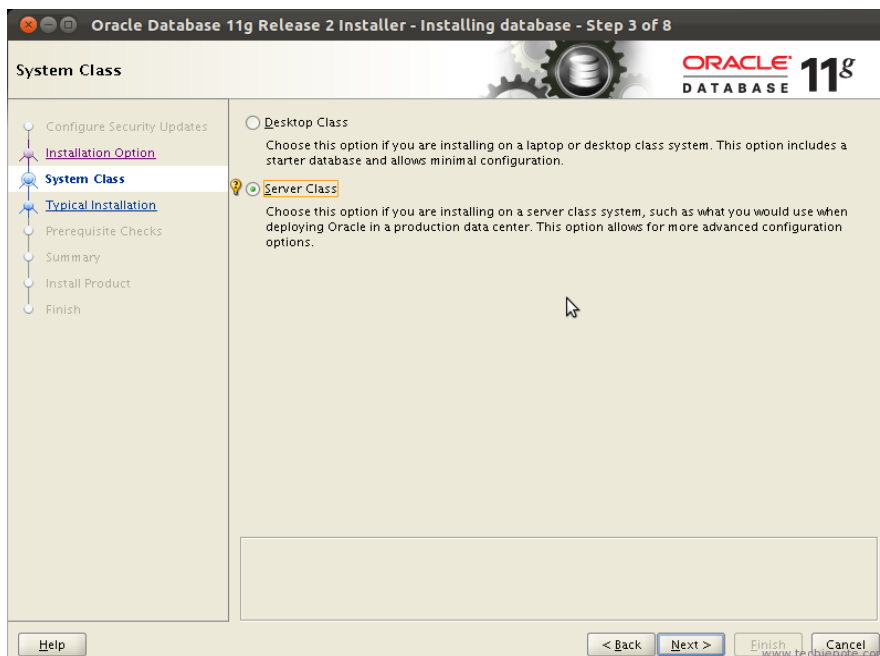
Ilustracja 16: Instalacja bazy danych - aktualizacje

Nie chcemy aby aktualizacje były dokonywane automatycznie. Gdy zaznaczymy tą opcję pojawi się error który wywołał brak podania adresu e-mail. Klikamy „Yes” i nie przejmujemy się tym.



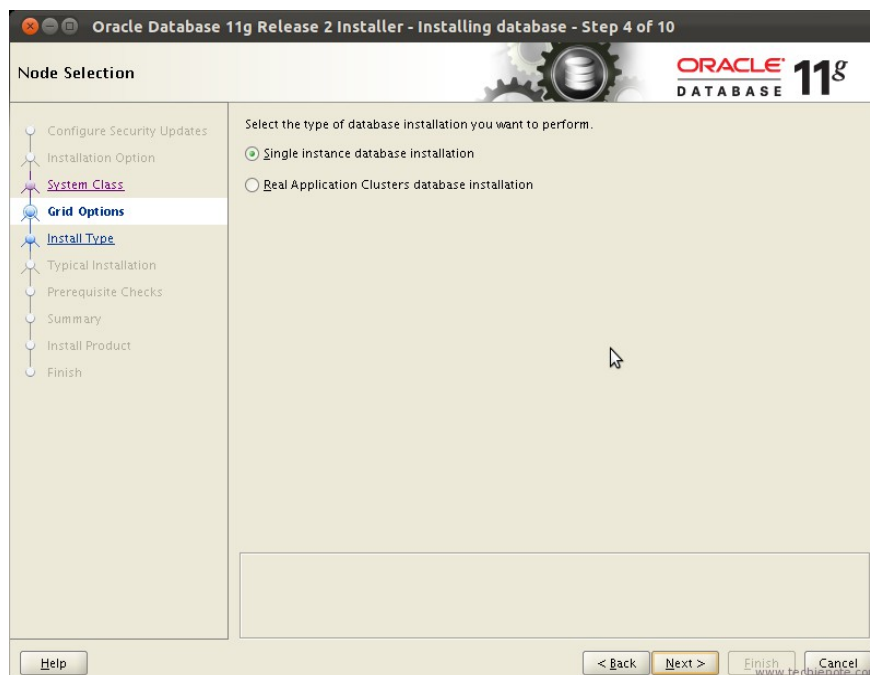
Ilustracja 17: Instalacja bazy danych - cel instalacji

Uruchomiliśmy instalator aby stworzyć i skonfigurować bazę danych i taką opcję zaznaczymy na formularzu instalacyjnym.



Ilustracja 18: Instalacja bazy danych - zastosowanie bazy danych

Instalujemy bazę danych, aby pełniła rolę serwera, nie zamierzamy wykorzystywać ją wyłącznie lokalnie. Dlatego będzie należała do klasy serwerów.

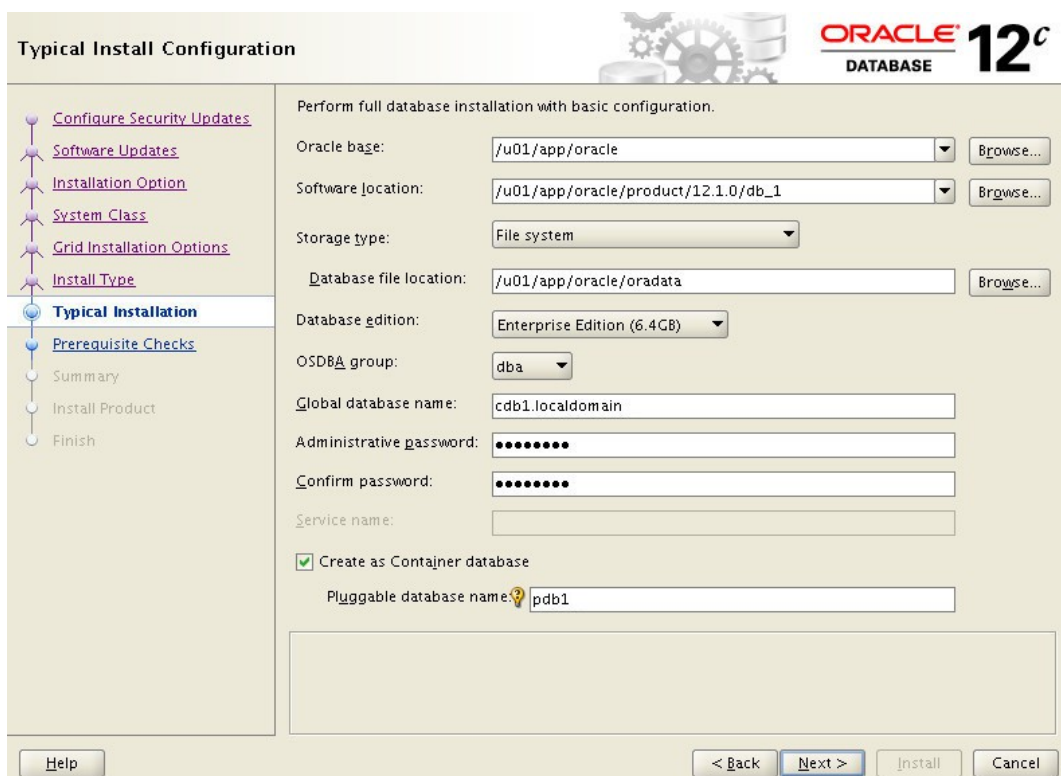


Ilustracja 19: Instalacja bazy danych - samodzielna instancja

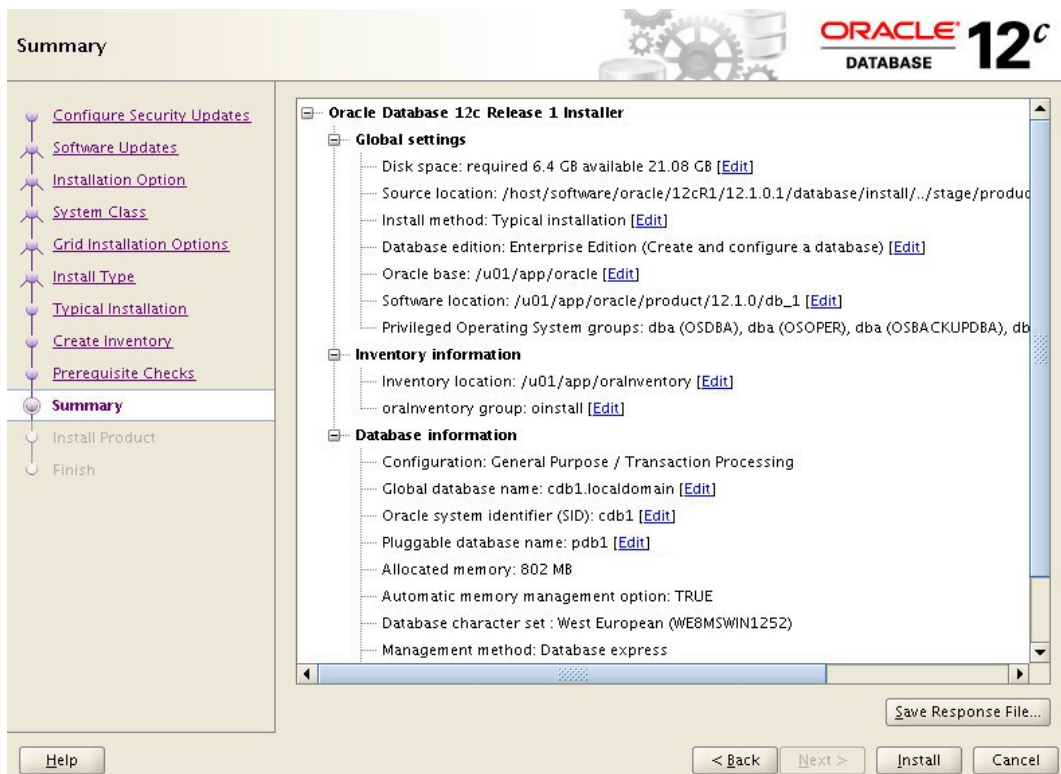
Instalowana baza ma w całości znajdować się na dysku, nie będzie częścią klastra.



Ilustracja 20: Instalacja bazy danych - typ instalacji
 Nasza baza danych bazuje na podstawowych (typowych) konfiguracjach.



Ilustracja 21: Instalacja bazy danych - szczegóły konfiguracji
 Wybieramy nazwę bazy danych i jej lokalizację. Również hasło administratora bazy.
 Następnie instalator przystąpi do weryfikacji ustawień systemu. Sprawdza, czy dysponujemy odpowiednio dużą wolną przestrzenią na dysku twardym, czy posiadamy dość pamięci RAM, oraz czy wszystkie aplikacje i zmienne systemowe są dostępne. Na końcu sprawdzania instalator zwróci raport z którego dowiemy się które elementy są w systemie, a których brak.



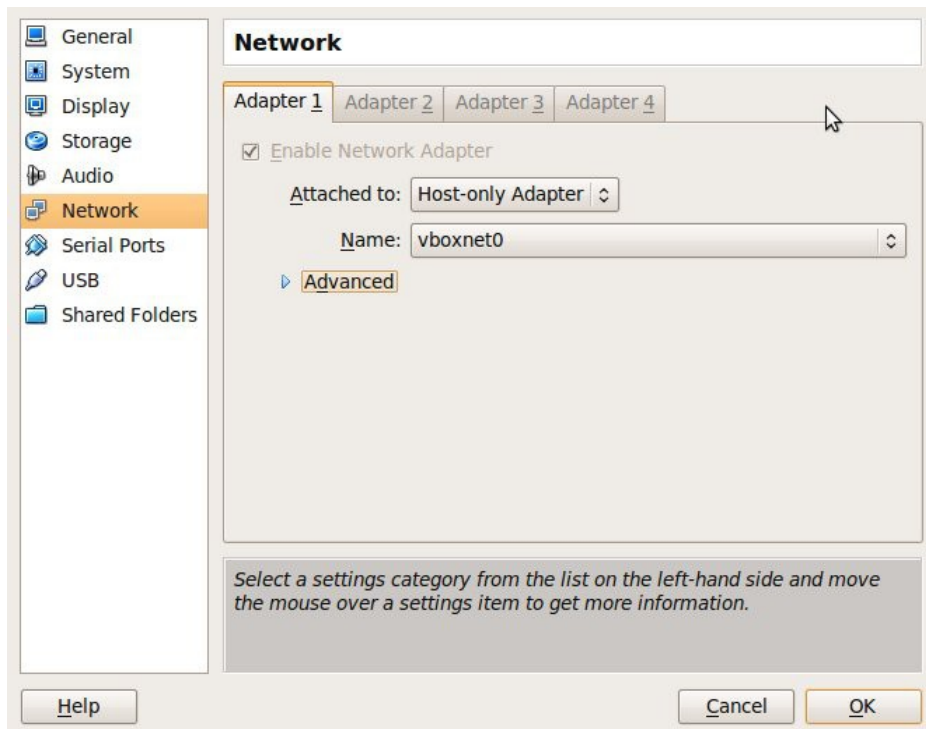
Ilustracja 22: Instalacja bazy danych - podsumowanie testowania systemu

Gdy wciśniemy „Install” rozpocznie się instalowanie bazy danych. Poprawnie zainstalowana baza danych można uruchomić poprzez Terminal dzięki komendzie:

```
$ SQLPLUS / AS SYSDBA
```

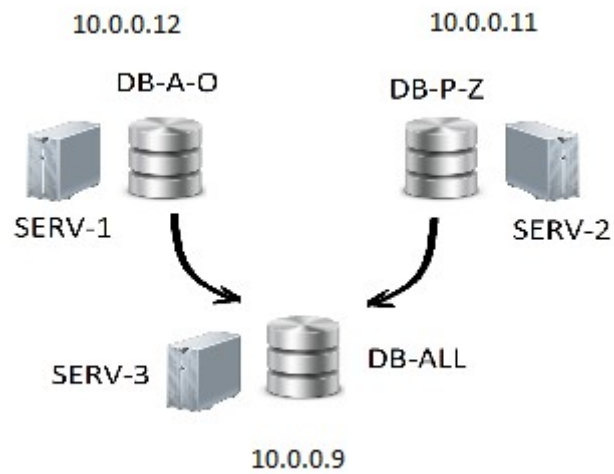
4.4. Konfiguracja połączenia sieciowego

Konfiguracja połączenia sieciowego sprawiła wiele problemów, wszystko spowodowane systemami zainstalowanymi na wirtualnych maszynach. Aby umożliwić łączenie się między wirtualnymi maszynami konieczne jest ustawienie adapterów w ustawieniach maszyn.



Ilustracja 23: Konfiguracja portów sieciowych

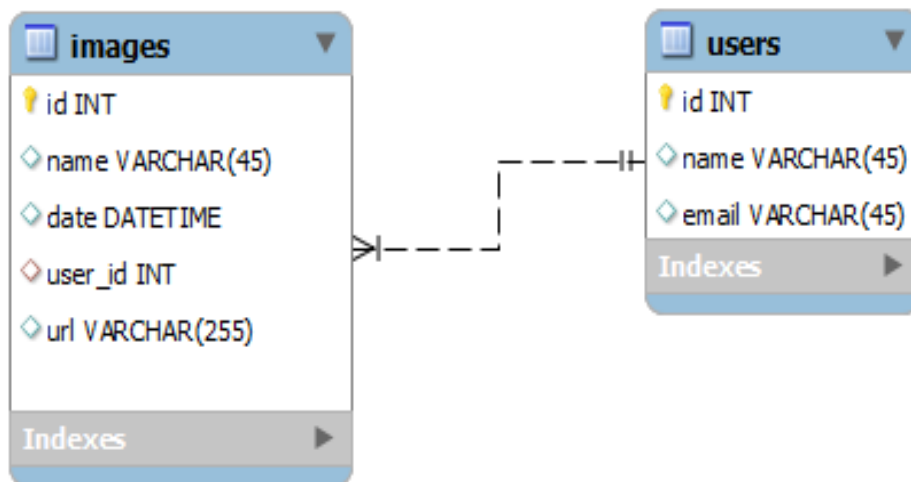
Na wszystkich maszynach ustawiliśmy połączenie tylko do hostów adaptera.
Teraz możemy ustawić adresy IP według poniższego schematu.



Ilustracja 24: Schemat sieci

5. Projekt i implementacja bazy danych

5.1. Model bazy danych



Ilustracja 25: Model struktury bazy danych

Struktura bazy danych na wszystkich serwerach będzie taka sama. Tabele będą dwie. Jedna do przechowywania obrazów wraz z informacjami o nich (nazwa, czas wprowadzenia, id użytkownika wprowadzającego). Druga tabela przechowuje informacje o użytkownikach (nazwę i e-mail).

5.2. Tworzenie bazy danych

Najpierw na trzech różnych maszynach tworzymy trzy bazy danych:

```
> CREATE DATABASE DB-A-O;
> CREATE DATABASE DB-P-Z;
> CREATE DATABASE DB-ALL;
```

Baza DB-ALL jest zbiorcza dlatego nie tworzymy na niej tabel, zostaną utworzone podczas pierwszej replikacji. Należy tylko utworzyć linki do tych tabel (5.5.). Natomiast na pozostałych bazach tworzymy tabele na obrazy.

```
> CREATE TABLE "SCOTT"."IMAGES"
(
    "ID" NUMBER(*,0),
    "NAME" VARCHAR2(255 BYTE),
    "DATE_CR" DATE,
    "USER_ID" NUMBER(*,0),
    "IMG_DATA" BLOB,
    PRIMARY KEY ("ID")
);
```

Oraz tabelę z użytkownikami:

```
> CREATE TABLE "SCOTT"."IMG_USERS"
(
    "ID" NUMBER NOT NULL ENABLE,
    "NAME" VARCHAR2(20 BYTE),
    "EMAIL" VARCHAR2(20 BYTE),
    CONSTRAINT "IMG_USER_PK" PRIMARY KEY ("ID")
);
```

Podczas testowania aplikacji pojawił się problem z replikacją obrazów. Są to obiekty o dużych rozmiarach (BLOB). Podczas próby przeniesienia takiego obiektu przez sieć od innej bazy pojawiała się komunikat o bł

```
ERROR at line 1:ORA-22992: cannot use LOB locators  
selected from remote tablesORA-06512: at line 4
```

Rozwiązaniem okazało się stworzenie perspektywy (widoku) na bazie którego będą replikowane dane.

```
> CREATE GLOBAL TEMPORARY TABLE LocalTempTableWithBlob  
> CREATE OR REPLACE FUNCTION GETBLOB  
> CREATE VIEW myRemoteData  
> SELECT id, name, date_cr, user_id, img_data, mytext  
FROM myRemoteData;
```

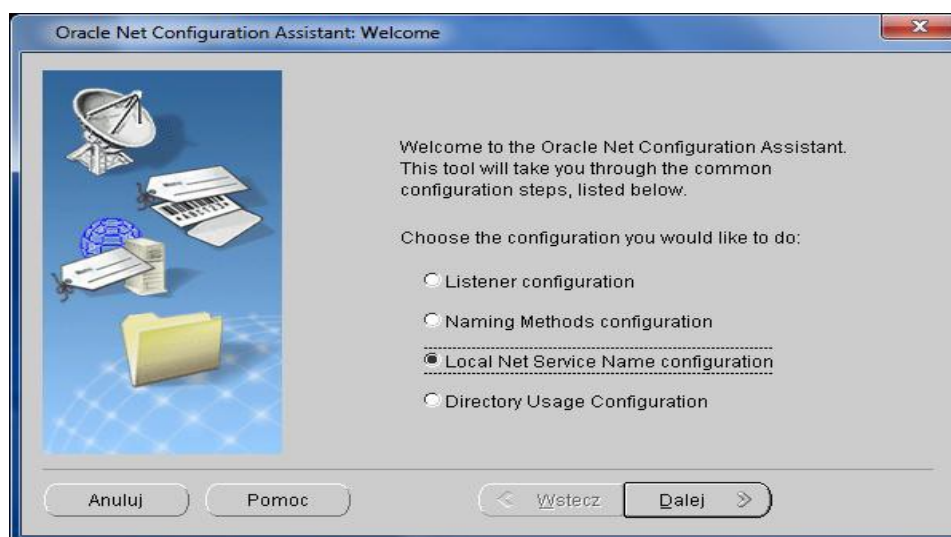
5.3. Konfiguracja *Local Net Service Name*

Usługa ta umożliwia stworzenie unikalnego identyfikatora do rozpoznawanie tego serwera bazodanowego w sieci lokalnej. Oracle umożliwia nam wykonanie tego przy pomocy intuicyjnego asystenta instalacji.

Asystenta uruchamiamy wywołując z terminala program:

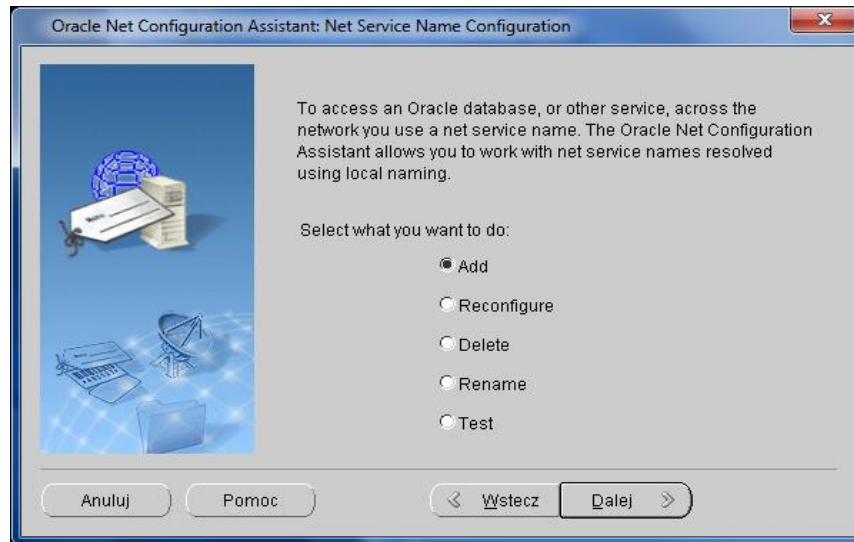
```
$ netca
```

Pierwszym krokiem jest wybór konfiguracji.



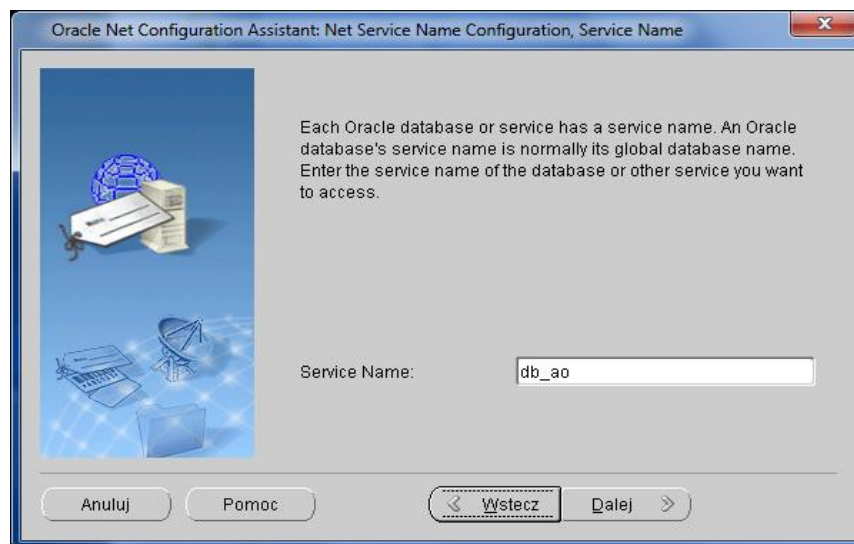
Ilustracja 26: Wybór konfiguracji Local Net Service Name

Następnie wybieramy opcję *Add* aby dodać nazwę.



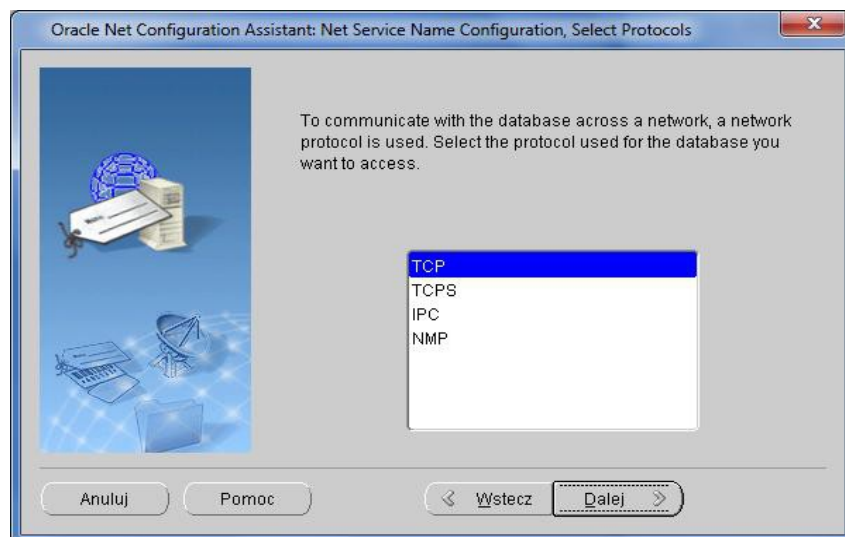
Ilustracja 27: Dodanie nowej nazwy

Wprowadzamy nazwę bazy danych którą chcemy wystawić do sieci. Nasza baza to *db_ao*, ponieważ przechowuje ona rekordy o nazwach z przedziału A-O.



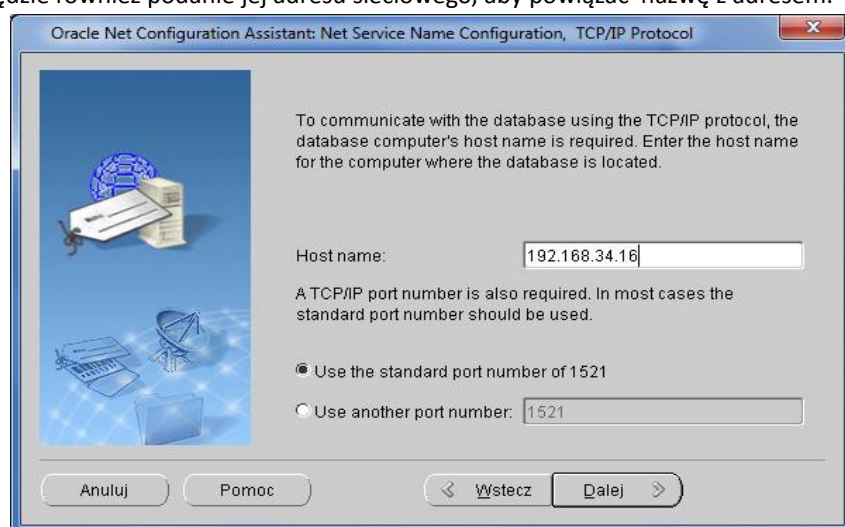
Ilustracja 28: Wprowadzenie nazwy sieciowej

Wybór protokołu komunikacji interfejsu sieciowego. Wybraliśmy TCP, ponieważ jest najprostszy.



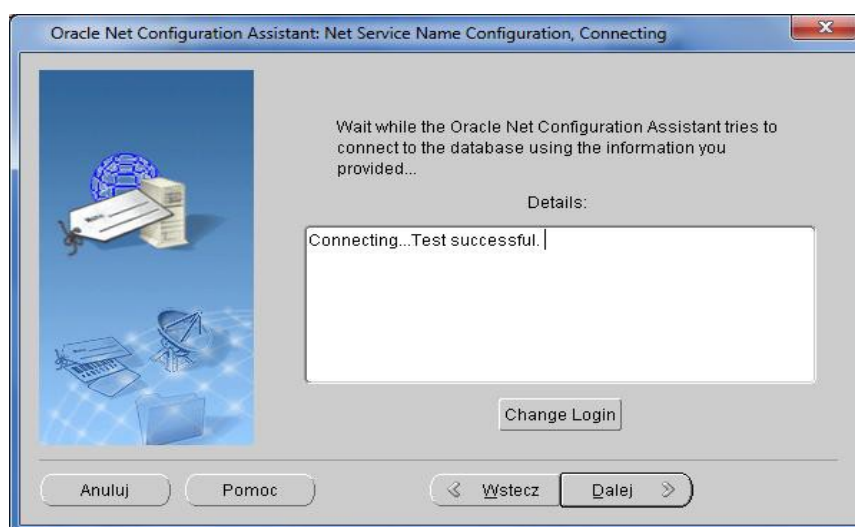
Ilustracja 29: Wybór protokołu komunikacji z bazą danych.

Nazwa którą podaliśmy wcześniej posłuży do rozpoznawania bazy danych w sieci, niezbędne będzie również podanie jej adresu sieciowego, aby powiązać nazwę z adresem.



Ilustracja 30: Wprowadzenie adresu IP pod którym znajduje się serwer bazy danych

Asystent instalacji sam przetestuje połączenie i zweryfikuje adres IP komputera.



Ilustracja 31: Testowanie połączenia

Wprowadzamy nazwę po której ma być rozpoznawana baza danych w sieci. Tą bazę nazwaliśmy *db_ao*.



Ilustracja 32: Nazwa usługi

Jeśli wszystko zostanie wykonane poprawnie pojawi się niniejszy wpis w pliku *tnsnames.ora*.

```
DB_AO =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)
        (HOST = 192.168.34.16) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = db_ao)
    )
  )
```

5.4. Stworzenie dziennika zdarzeń

Na serwerze, który ma replikować dane musimy stworzyć dziennik migawek (zdarzeń). Dokonuje się to wprowadzając do terminala bazy danych komendę:

```
> CREATE SNAPSHOT LOG ON images;
```

Gdzie *images* oznacza nazwę tabeli dla której tworzymy dziennik. Przykładowy wpis do dziennika wygląda jak poniżej.

ID	SNAPTIME	D	O	CHANGE_VECTOR\$\$	XID\$\$
1	00/01/01	U	U	04	1,1259E+15

Jeśli zadeklarowaliśmy że replikacja będzie wykonywana w trybie przyrostowym, to dziennik jest wymagany. W przeciwnym razie pojawi się taki błąd podczas próby nawiązania połączenia.

```
Błąd w linii 1:
ORA-23413: tabela "SCOTT"."IMAGES" nie ma dziennika
zmaterializowanej perspektywy
```

5.5. Tworzenie replikacji migawkowej

W standardowym przypadku definicja migawki składa się z następujących elementów:

- nazwy migawki,
- momentu wypełnienia migawki danymi,
- specyfikacji sposobu odświeżania,
- specyfikacji momentu rozpoczęcia automatycznego odświeżania,
- specyfikacji częstotliwości odświeżania,
- typu migawki i zapytania określającego zakres danych dostępnych w migawce.

Migawkę tworzy się poleceniem SQL *create snapshot* lub *create materialized view*, którego podstawową składnię przedstawiono poniżej:

```
> CREATE SNAPSHOT images
  REFRESH
  NEXT sysdate + (1/(24*60*2))
  AS
  SELECT id, name, date_cr, user_id, url
  FROM images@db_ao
  UNION
  SELECT id, name, date_cr, user_id, url
  FROM images@db_pz
```

Specyfikacja sposobu odświeżania migawki (klauzula *refresh* sposób odświeżania) umożliwia określenie, czy migawka ma być odświeżana w sposób pełny, czy przyrostowy. Odświeżanie przyrostowe jest możliwe jeśli na tabeli źródłowej migawki został utworzony dziennik migawki oraz migawka jest typu prostego. Tryb przyrostowy oznacza właśnie replikację zapytań, nie stricte danych.

Migawka może być odświeżana:

- manualnie, czyli na żądanie użytkownika,
- automatycznie po zatwierdzeniu transakcji modyfikującej dane źródłowe,
- automatycznie z zadaną częstotliwością,
- może nie być nigdy odświeżona.

W definicji migawki można określić moment i częstotliwość jej odświeżania. Służy do tego celu klauzula *refresh* o następującej składni:

```
refresh
{fast | complete | force} [{on demand | on commit}]
[start with data_pierwszego_odświeżenia]
[next częstotliwość_odświeżania]
```

W naszym przypadku replikacja jest w trybie fast i jest odświeżana co 5 min. Druga część polecenia określa atrybuty i tabelę której migawka ma powstać.

Na bazie DB-ALL tworzymy dowiązania do obu baz:

```
> CREATE DATABASE LINK db_ao CONNECT TO scott
  IDENTIFIED BY TRIGER USING 'db_ao';
> CREATE SNAPSHOT rep_dept
  REFRESH fast
  NEXT sysdate + 1/17280
  AS
  SELECT deptno, dname, loc
  FROM dept@db_ao;
> CREATE SNAPSHOT rep_images
  REFRESH fast
  NEXT sysdate + 1/17280
  AS
  SELECT id, name, date_cr, user_id, url
  FROM images@db_ao;
> CREATE SNAPSHOT images
  REFRESH
  NEXT sysdate + (1/(24*60*2))
  AS
  SELECT id, name, date_cr, user_id, img_data
  FROM images@db_ao
  UNION
  SELECT id, name, date_cr, user_id, img_data
  FROM images@db_pz;
```

5.6. System zabezpieczeń

Podczas projektowania aplikacji założyliśmy, że będzie ona zainstalowana na bezpiecznych komputerach, dlatego nie podjęliśmy żadnych dodatkowych zabezpieczeń. Do bazy danych mają dostęp tylko osoby posiadające konto i znają swoje hasło.

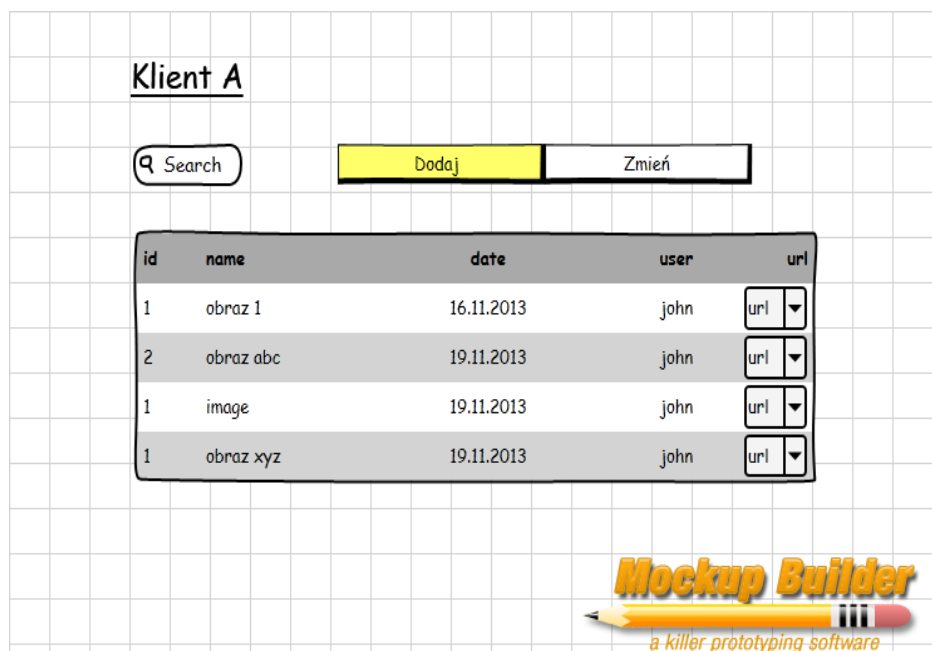
Do replikacji wykorzystujemy mechanizm perspektywy, co może pozwolić na utajnienie niektórych danych. Przed innymi użytkownikami.

6. Projekt i implementacja aplikacji

6.1. Model aplikacji

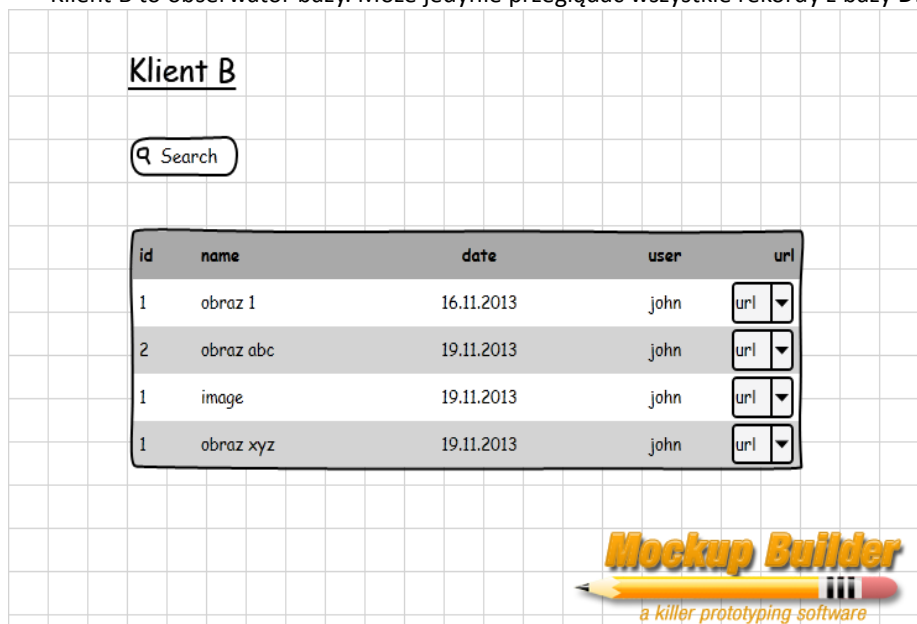
Przed implementacją przygotowaliśmy model okna aplikacji przy pomocy programu Mockup Builder. Model nie jest odwzorowaniem obiektów graficznych z żadnej biblioteki, dlatego model jest bardziej uniwersalny (abstrakcyjny).

Aplikacja składa się z dwóch rodzi interfejsów. Klient A to typ użytkownika który ma możliwość wprowadzania obrazów do bazy. Dlatego te interfejsy będą uruchomione na SERV-1 i 2.



Ilustracja 33: Model okna klienta A

Klient B to obserwator bazy. Może jedynie przeglądać wszystkie rekordy z bazy DB-ALL.



Ilustracja 34: Model okna klienta B

6.2. Instalacja i przygotowanie środowiska programistycznego

Przygotowanie środowiska programistycznego nie wymagało wielu dodatkowych zabiegów. Do stworzenia graficznych elementów wykorzystana została standardowa biblioteka Swing. Natomiast połączenie z bazą danych za pośrednictwem Hibernate'a nie wymagało dodatkowej biblioteki.

6.3. Implantacja połączenia z rozproszoną bazą danych

Aplikacja łączy się z bazą danych za pośrednictwem rozszerzenia *Hibernate 2.0*. Jest to narzędzie typu ORM służące do rzutowania relacyjnej bazy danych na obiektową. Jest to narzędzie większe i potężniejsze od *JDBC*. Pozwala na łatwiejsze tworzenie rozbudowanych zapytań.

Aby móc korzystać z łączenia z bazą danych należy stworzyć plik *XML* w którym podajemy lokalizację serwera bazy danych, oraz wszystkie niezbędne do tego parametry.

```
<hibernate-configuration>
  <session-factory>
    <property name="hibernate.connection.driver_class">
      oracle.jdbc.driver.OracleDriver
    </property>
    <property name="hibernate.connection.url">
      jdbc:oracle:thin:@127.0.0.1:1521:DBALL
    </property>
    <property name="hibernate.connection.username">
      scott
    </property>
    <property name="hibernate.connection.password">
      tiger
    </property>
    <mapping class = "data.Image" />
    <mapping class = "data.User" />
  </session-factory>
</hibernate-configuration>
```

Taki plik należy stworzyć dla każdego połączenia z bazą danych. Czyli dla każdej instancji programu i dla każdej bazy danych. *hibernate_DBALL*, *hibernate_DBAO*, *hibernate_DBPZ*. Różnica między plikami jest wyłącznie w adresie URL.

Po skonfigurowaniu połączenia trzeba stworzyć interfejsy do komunikacji między serwerem a warstwą kliencką. Interfejsy te zawierają definicję metody będącej dostępną na serwerze. Dzięki temu odseparowaniu zapytania i cała komunikacja z bazą danych odbywa się po stronie serwera. Natomiast klient może wywołać metodę z serwera i posługiwać się obiektem zwróconym przez serwer.

Połączenie z baz danych następuje przy pomocy obiektu klasy *SessionFactory* który na podstawie pliku *XML* tworzy połączenie. Użytkownik wykorzystując metody *save* i *commit* może przekazać obiekty do bazy danych. Jednak aby obiekt mógł być rzutowany na bazę danych wymagana jest definicja encji, w której powiążemy kolumny bazy danych z polami obiektów. Ważne aby nie pomylić się w nazwach atrybutów tabel, oraz w typach. Próba zapisania obiektu typu *String* do kolumny przechowującej typ *VARCHAR* zakończy się błędem rzutowania.

Obiekty do tabeli *images* są tłumaczone następująco:

```
@Entity
@Table(name = "IMAGES")
public class Image implements Serializable {
    @Id
    @GenericGenerator(name="imggen" , strategy="increment")
    @GeneratedValue(generator="imggen")
    @Column(name = "id")
    private Long id;

    @Column(name = "name", nullable=true)
    private String name;

    @Column(name = "date_cr", nullable=true)
    private Date date;

    @ManyToOne
    @JoinColumn(name = "user_id")
    private User user;

    @Column(name = "img_data", nullable=true)
    private byte[] imgData;
}
```

Przykładowa komunikacja z bazą danych odbywa się w klasie *GenericHibernateDAO.java* metoda *makePersistant()* zapisuje obiekt do bazy danych.

```
public T makePersistent(T entity) {
    Session s = getSession();
    Transaction t = s.beginTransaction();

    try {
        s.saveOrUpdate(entity);
        t.commit();
    } catch (Exception e) {
        t.rollback();
    }

    return entity;
}
```

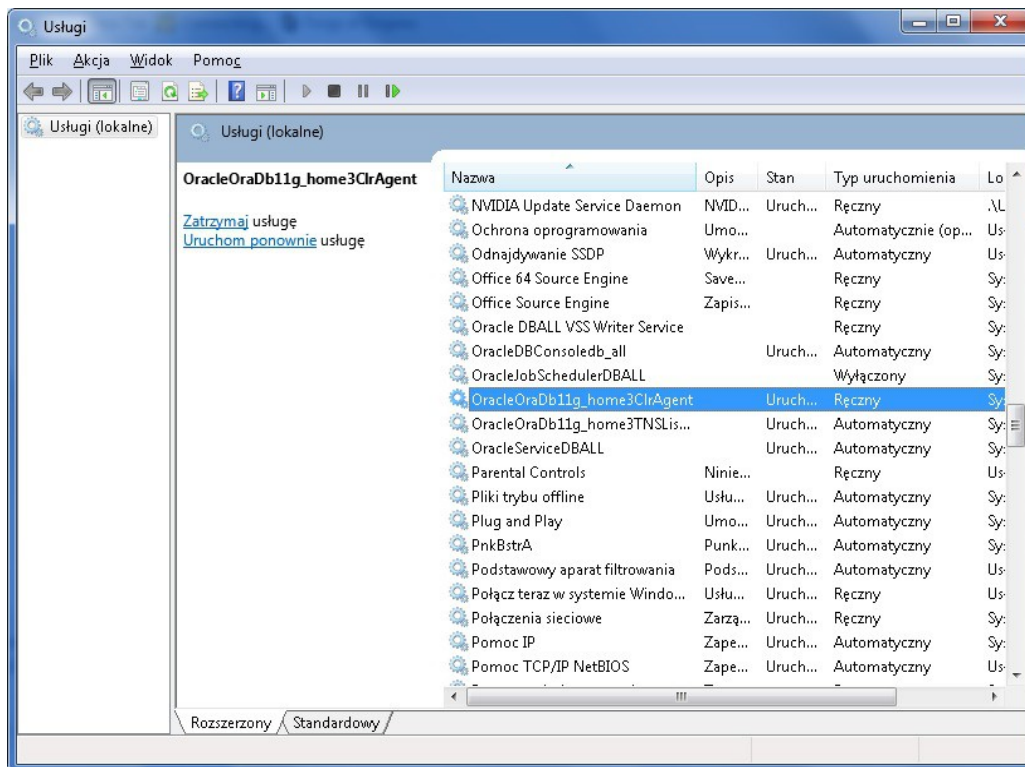
7. Instalowanie i testowanie

7.1. Instrukcja obsługi aplikacji

Program uruchamia się przy pomocy skryptów. Plik wykonywalny Javy wymaga podania jako parametru tylko nazwy bazy danych znajdującej się na maszynie

7.2. Przygotowanie środowiska Windows

Sekwencja uruchomienia bazy danych:



Ilustracja 35: Uruchamianie usługi

Uruchamiamy usługę *OracleServiceDBALL*. Następnie w terminalu wywołujemy programy:

```
$ sqlplus /nolog
$ sqlplus
$ connect as sysdba
```

Podaj login i hasło

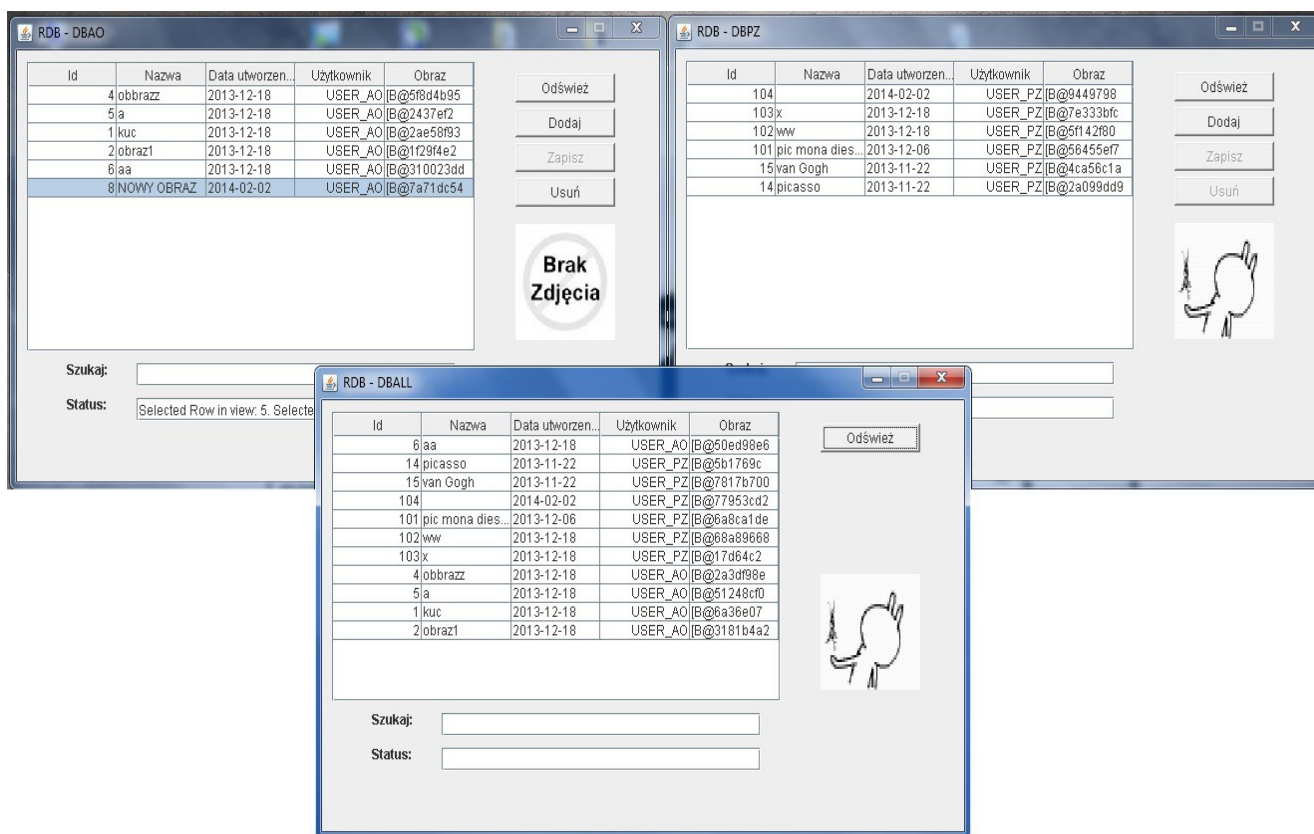
```
$ startup
$ lsnrctl start
$ emctl start dbconsole
```

7.3. Przygotowanie środowiska Linux

Aplikacja może działać pod systemem Linux, wymaga to takiej samej sekwencji uruchamiania programów.

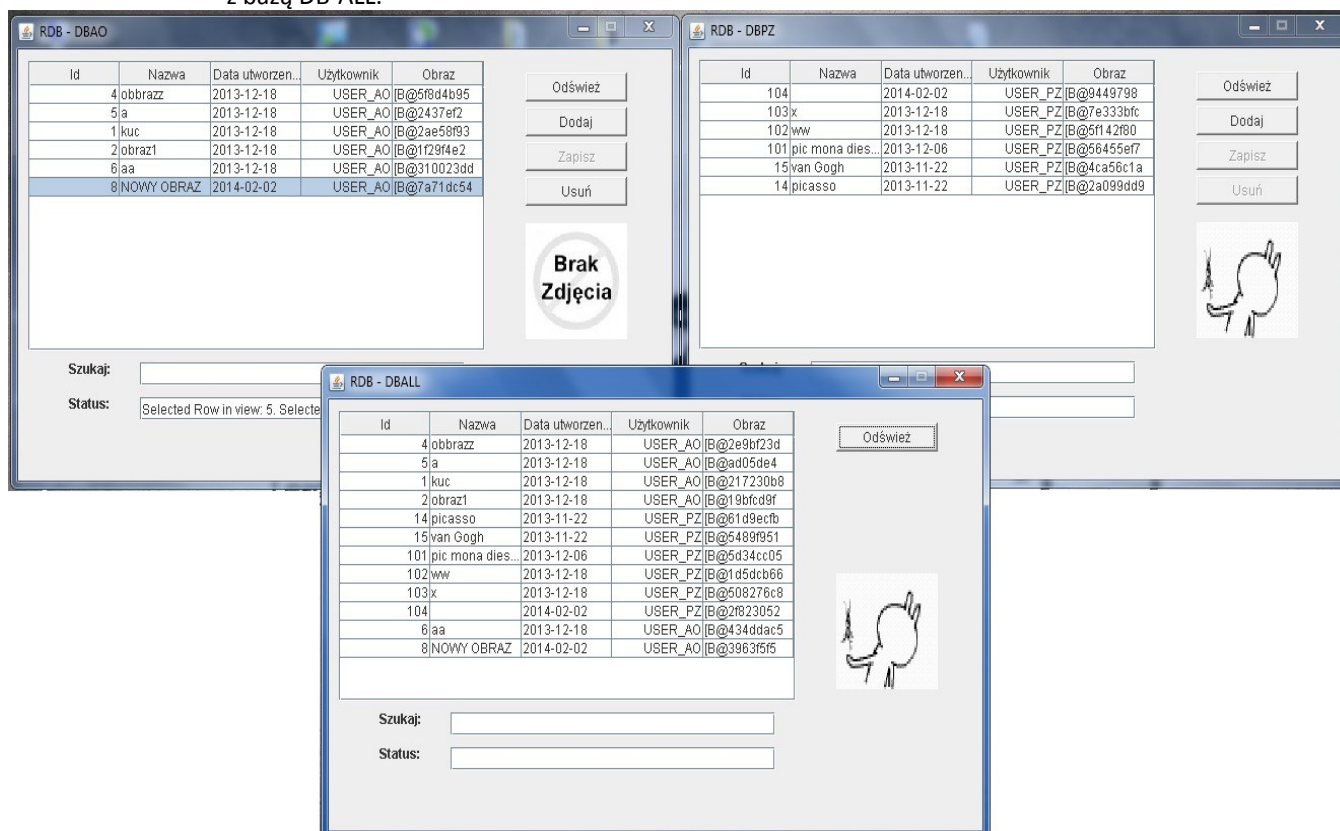
7.4. Testowanie mechanizmu replikacji

Efekt działania replikacji migawkowej są pojawiające się informacje o obrazkach w oknie drugiego użytkownika. Przed dodaniem nowego rekordu.



Ilustracja 36: Przed replikacji

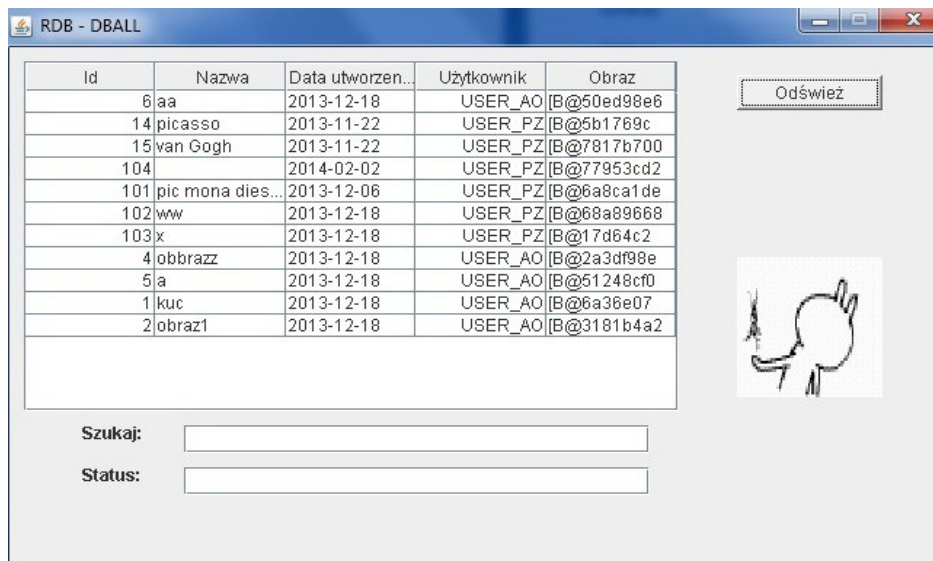
Po dodaniu obrazu „NOWY OBRAZ” do bazy u klienta z bazą DB-A-O rekord pojawił się u klienta z bazą DB-ALL.



Ilustracja 37: Po replikacji

7.5. Testowanie aplikacji

Podstawową funkcjonalnością aplikacji jest wyświetlanie obrazków i informacji na ich temat. Na obrazie widać tabelę z informacjami, natomiast po prawej widać miniaturę obrazu.



Ilustracja 38: Działanie aplikacji

7.6. Wnioski z testów

Po przeprowadzonych testach akceptacyjnych aplikacja działa bez zarzutów. Mechanizm replikacji kopiuje wszystkie dane. Nawet duże obiekty jak obrazy. Sama replikacja następuje błyskawicznie.

Wyszukiwanie również działa bez zarzutów. Podczas wpisywania w okno słowa od razu dane w tabeli są filtrowane. Zaznaczony rekord sprawia, że pojawia się po prawej stronie miniatura obrazu z bazy danych. Wymaga to wciśnięcia guzika „Odśwież”.

8. Podsumowanie

Projekt ukończyliśmy zgodnie z harmonogramem i projekt zakończył się realizacją pełnych wymagań. Duże problemy sprawiła nam instalacja bazy danych. Próba konfiguracji systemu Linux Ubuntu zakończyła się niepowodzeniem. Niestety była bardzo uciążliwa i czasochłonna. Instalacja systemu Linux Oracle okazała się dużo prostsza. Mimo to zainstalowania samej bazy danych sprawiła też mały problem.

Konfiguracja replikacji bazy danych nie był zadaniem trywialnym, głównie z powodu dużych obiektów. Rozwiązaniem było stworzenie widoku.

Napisanie aplikacji bazowało w dużej mierze na technologiach nam znanych. Dlatego było to nietrudne.

Program bazujący na rozproszonych bazach danych posiada dużo więcej możliwości. Umożliwia prostsze automatyzowanie niektórych działań. Aplikacje nie muszą się komunikować bezpośrednio, wystarczy aby bazy danych komunikowały się między sobą. Dodatkowo przenosimy logikę operacji na warstwę bazy danych.

Rozproszone bazy danych zwiększają bezpieczeństwo danych, dzięki automatycznemu replikowaniu ich na inne serwery. Program napisany z użyciem tego typu baz danych jest również bardziej elastyczny. Można łatwo zwiększać liczbę baz przechowujących informacje, oraz zmieniać fizyczne ich położenie. Zaletą jest również wzrost dostępności, dzięki zdecentralizowanie systemu. Wiele baz z tymi samymi danymi pozwala na działanie usługi pomimo uszkodzenia jednego z serwerów.

Rozproszone bazy danych wymagają zaawansowanej wiedzy z dziedziny baz danych, ale ich stosowanie daje olbrzymie możliwości. Co za tym idzie również wiele zalet.

Literatura

- [1] Górski J. *Inżynieria oprogramowania w projekcie informatycznym*, Mikom, Warszawa, 2000
- [2] Bębel B., Wrembel R. *Replikacja danych w bazach danych Oracle9i*, Politechnika Poznańska, Instytut Informatyki 2003
- [3] Jendrock E., Evans I., Gollapudi D. *Java EE 6. Przewodnik*, wyd. Helion, 2012
- [4] Beynon-Davies P. *Systemy baz danych*, WNT, Warszawa, 2000