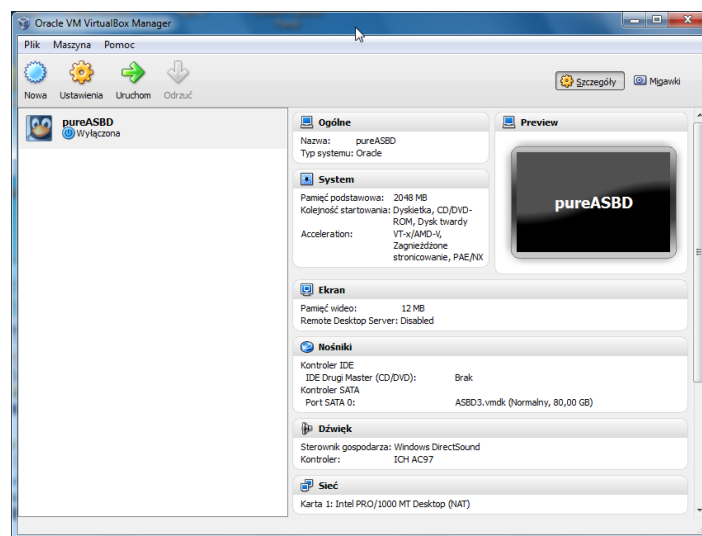


# Rozproszone bazy danych

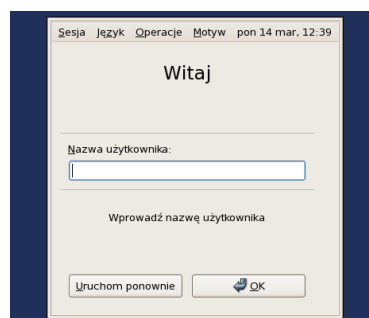
## 1. Uruchomienie i skonfigurowanie środowiska do ćwiczeń

Zadaniem niniejszych ćwiczeń jest przedstawienie podstawowych zagadnień dotyczących rozproszonych baz danych. Pierwsze ćwiczenia przygotowują środowisko, a następnie koncentrują się łącznikach bazy danych i replikacji asynchronicznej.

1. Uruchom środowisko wirtualizacji – kliknij na umieszczoną na pulpicie komputera-gospodarza ikonę *Oracle VM VirtualBox*.

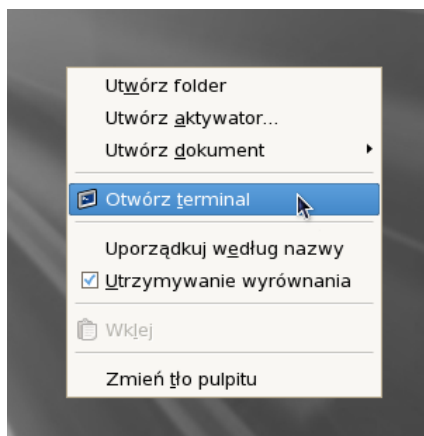


2. Spróbuj uruchomić maszynę wirtualną. W tym celu zaznacz w lewym panelu środowiska pozycję **pureASBD** i naciśnij umieszczony na pasku narzędzi przycisk **Uruchom** (możesz również wybrać pozycję **Uruchom** z menu kontekstowego, dostępnego po kliknięciu prawym klawiszem myszy na pozycji **pureASBD**).
3. Po pomyślnym uruchomieniu maszyny wirtualnej powinien zostać wyświetlony ekran logowania (jeśli ekran maszyny wirtualnej nie zajmuje całego ekranu komputera-gospodarza, użyj odpowiedniego skrótu klawiszowego aby to zmienić).



4. Zaloguj się do systemu operacyjnego maszyny wirtualnej jako użytkownik *oracle* z hasłem *oracle*. Podaj powyższe informacje i naciśnij przycisk **OK**.

5. Uruchom terminal. Można to wykonać przez kliknięcie prawym klawiszem myszki na pulpit i wybranie z menu kontekstowego pozycji **Otwórz terminal**.



6. Sprawdź wartości zmiennych środowiskowych dotyczących instancji *Oracle*. Wykorzystaj w tym celu polecenie: `set | grep ORACLE`

```
oracle@localhost:~  
Plik Edycja Widok Terminal Zakładki Pomoc  
[oracle@localhost ~]$ set | grep ORACLE  
ORACLE_BASE=/u01/app/oracle  
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1  
ORACLE_SID=baza01  
[oracle@localhost ~]$
```

7. Jeśli zmienne środowiskowe są ustawione poprawnie, uruchom program *sqlplus*. Wykorzystaj opcję *nolog*: `sqlplus /nolog`

```
oracle@localhost:~  
Plik Edycja Widok Terminal Zakładki Pomoc  
[oracle@localhost ~]$ set | grep ORACLE  
ORACLE_BASE=/u01/app/oracle  
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1  
ORACLE_SID=baza01  
[oracle@localhost ~]$ sqlplus /nolog  
  
SQL*Plus: Release 11.2.0.1.0 Production on Pn Mar 14 12:51:39 2011  
  
Copyright (c) 1982, 2009, Oracle. All rights reserved.  
  
SQL>
```

8. Następnie zaloguj się korzystając z autoryzacji użytkownika administracyjnego przez system operacyjny. Wykonaj w tym celu polecenie: `connect / as sysdba`  
Następnie uruchom bazę danych poleceniem `startup`

```
oracle@localhost:~  
Plik Edycja Widok Terminal Zakładki Pomoc  
[oracle@localhost ~]$ set | grep ORACLE  
ORACLE_BASE=/u01/app/oracle  
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1  
ORACLE_SID=baza01  
[oracle@localhost ~]$ sqlplus /nolog  
  
SQL*Plus: Release 11.2.0.1.0 Production on Pn Mar 14 12:51:39 2011  
  
Copyright (c) 1982, 2009, Oracle. All rights reserved.  
  
SQL> connect / as sysdba  
Połączono z nieaktywną instancją.  
SQL> startup  
Instancja ORACLE została uruchomiona.  
  
Total System Global Area 849530880 bytes  
Fixed Size 1339824 bytes  
Variable Size 503320144 bytes  
Database Buffers 339738624 bytes  
Redo Buffers 5132288 bytes  
Baza danych została zamontowana.  
Baza danych została otwarta.  
SQL>
```

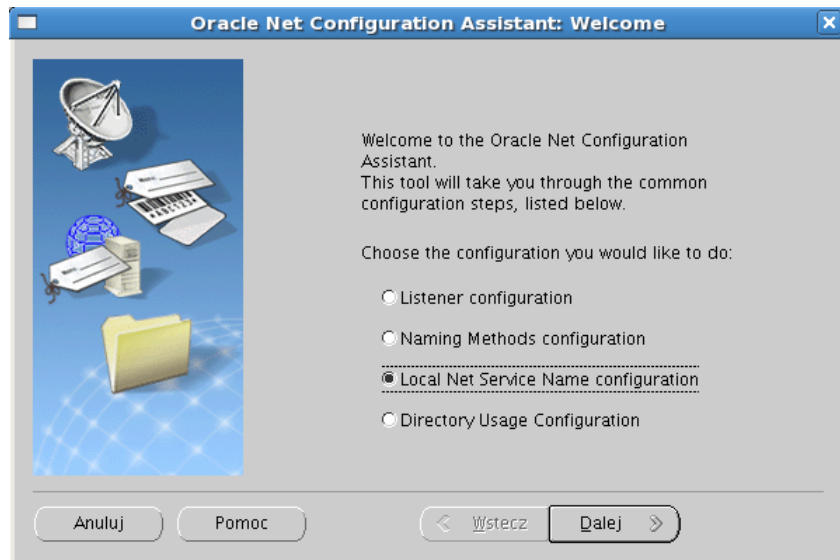
9. Aby uruchomić proces nasłuchowy uruchom terminal tekstowy a następnie wystartuj proces nasłuchowy poleceniem: **lsnrctl start**.

```
oracle@localhost:~  
Plik Edycja Widok Terminal Zakładki Pomoc  
[oracle@localhost ~]$ lsnrctl start  
  
LSNRCTL for Linux: Version 11.2.0.1.0 - Production on 16-MAR-2011 11:08:48  
  
Copyright (c) 1991, 2009, Oracle. All rights reserved.  
  
Uruchamianie /u01/app/oracle/product/11.2.0/dbhome_1/bin/tnslsnr: proszę czekać.  
..  
Nasłuch punktów końcowych - podsumowanie...  
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=localhost.localdomain)(PORT=1521)))  
Nasłuch nie zapewnia usług  
Polecenie zakończone powodzeniem  
[oracle@localhost ~]$
```

## 2. Konfiguracja parametrów połączenia do odległej bazy danych

Celem ćwiczenia jest skonfigurowanie środowiska komunikacyjnego systemu Oracle w celu możliwości nawiązania połączenia do odległej bazy danych o nazwie dblab11g.

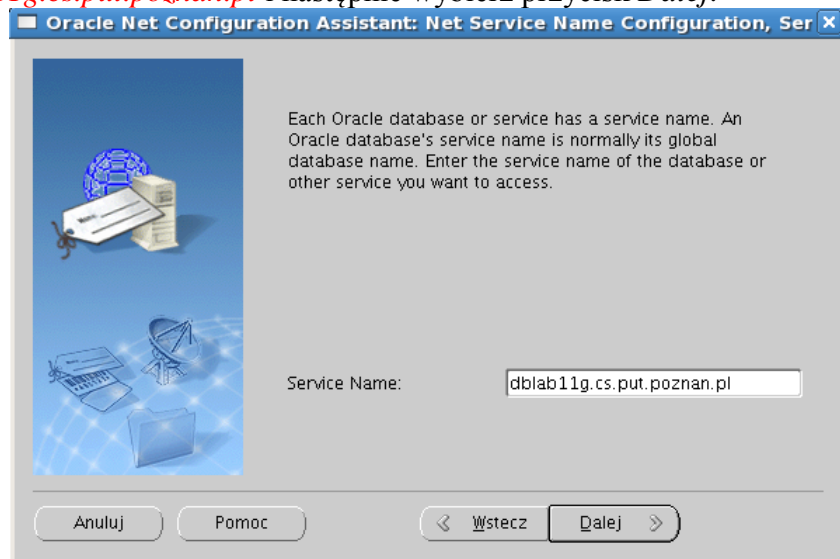
1. Konfigurację wykonamy za pomocą narzędzia *Oracle Network Configuration Assistant*. W tym celu w terminalu wprowadź polecenie **netca**.
2. Po chwili pojawi okno przywitane narzędzia, wybierz pozycję *Local Net Service Name configuration* i następnie wybierz przycisk *Dalej*.



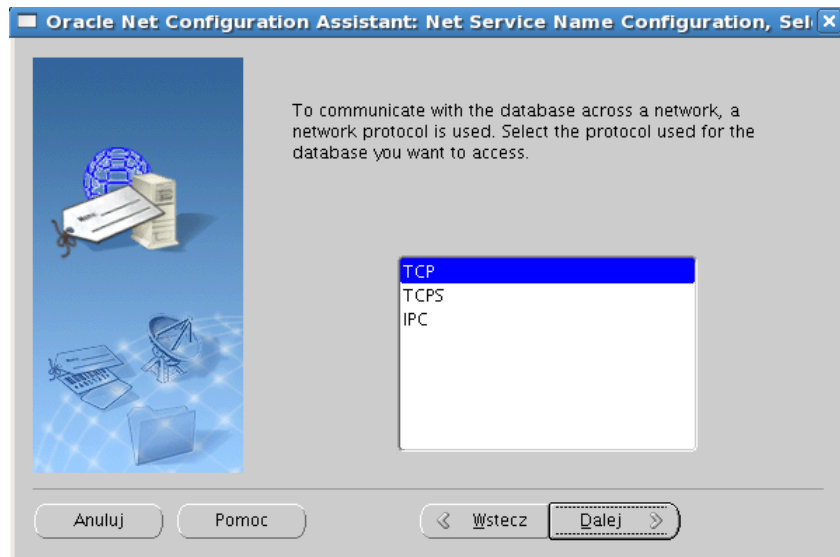
3. Wybierz przycisk radiowy *Add* i następnie wybierz przycisk *Dalej*.



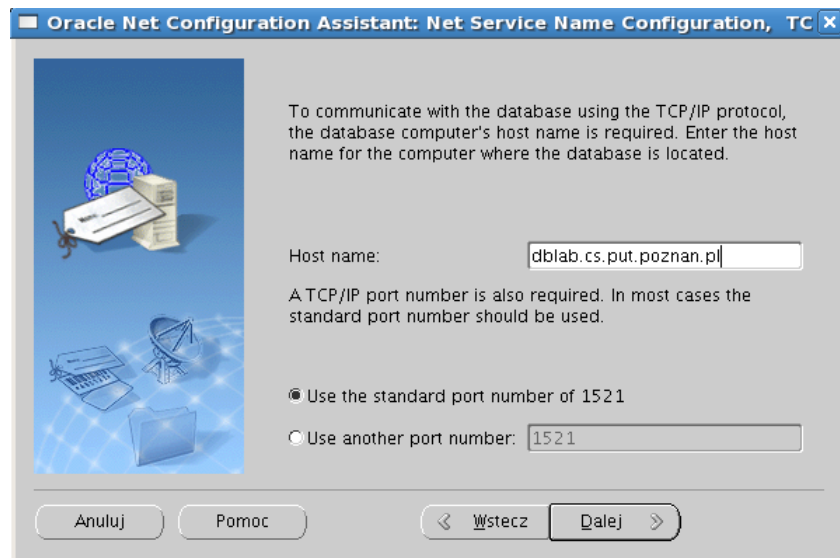
4. Wprowadź nazwę usługi dostarczanej przez odległą bazę danych *dblab11g.cs.put.poznan.pl* i następnie wybierz przycisk *Dalej*.



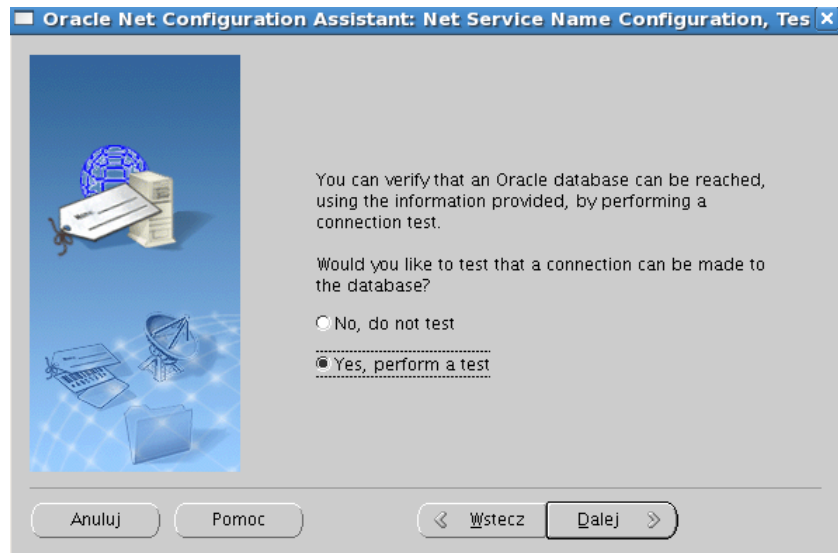
5. Wybierz protokół TCP, za pomocą którego będzie się odbywać komunikacja z odległą bazą danych i następnie wybierz przycisk *Dalej*.



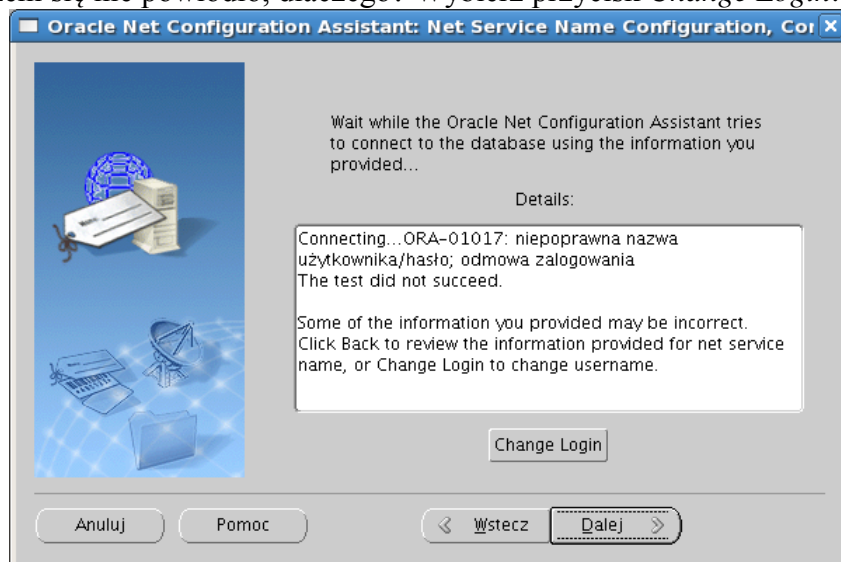
6. Wprowadź nazwę maszyny, na której jest zainstalowana baza danych [dblab.cs.put.poznan.pl](http://dblab.cs.put.poznan.pl) oraz wybierz standardowy port 1521, na którym nasłuchuje proces nasłuchu komunikujący nas z odległą bazą danych, następnie wybierz przycisk *Dalej*.



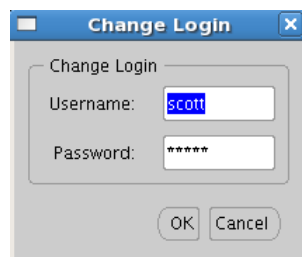
7. Wybierz przycisk radiowy w celu wykonania testu połączenia i następnie wybierz przycisk *Dalej*.



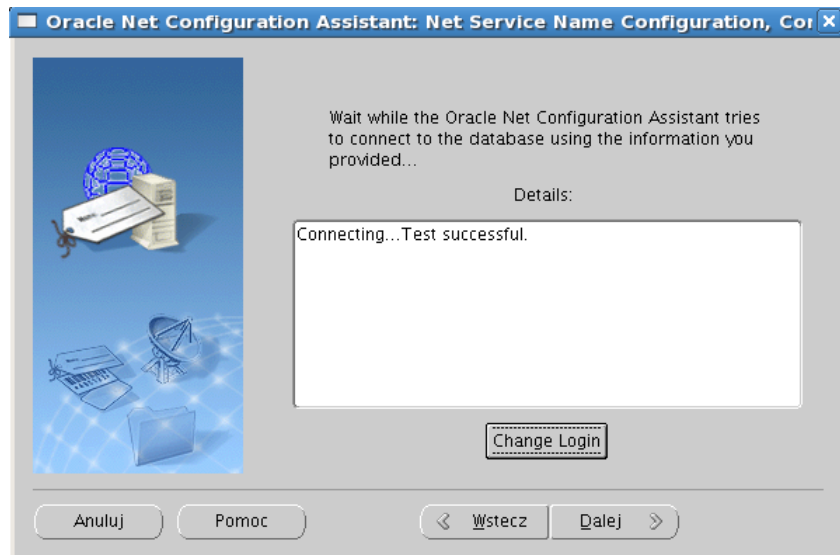
8. Połączeni się nie powiodło, dlaczego? Wybierz przycisk *Change Login*.



9. Wprowadź nazwę użytkownika **scott** i jego hasło **tiger**, a następnie wybierz przycisk *OK*.



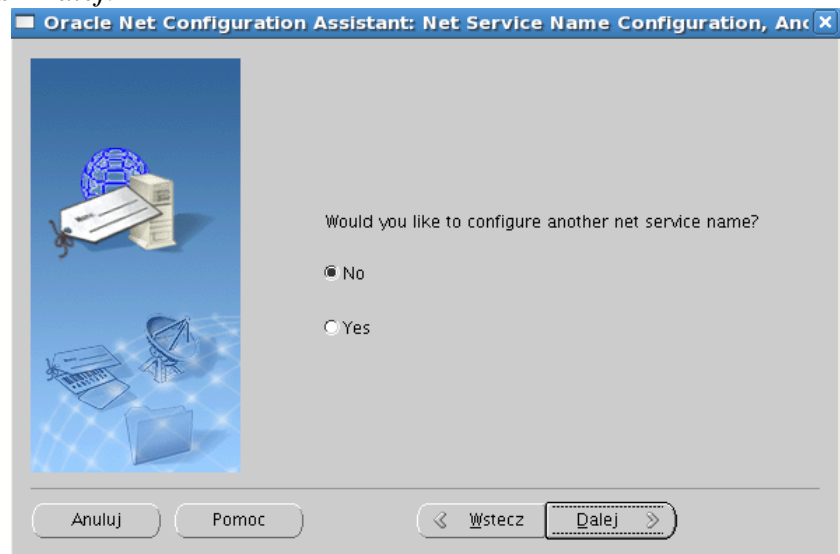
10. Czy tym razem test powiódł się? Wybierz przycisk *Dalej*.



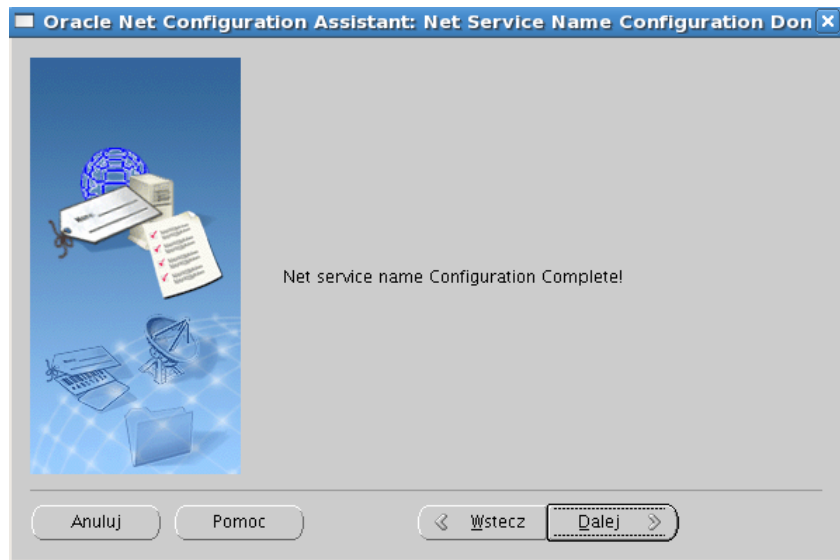
11. Następnie wprowadź nazwę usługi sieciowej, którą będziesz się posługiwał w odwołaniach do zdalnej bazy danych: *dblab11g*.



12. Skończyliśmy konfigurację, więc wybierz przycisk radiowy *No* i następnie wybierz przycisk *Dalej*.



13. Wybierz przycisk *Dalej*.



14. Opuść narzędzie, wybierając przycisk *Zakończ*.



15. Podejrzyj zawartość pliku `tnsnames.ora`, zawierającego specyfikację zdalnych baz danych (nazwy „usług” i odpowiadające im parametry połączenia). Jest tam również zapisana konfiguracja, która została przez Ciebie przed chwilą wprowadzona. Plik ten znajduje się w domyślnej lokalizacji `$ORACLE_HOME/network/admin`. Zwróć uwagę na wpis dotyczący bazy danych `dblab11g`. W terminalu wprowadź następujące połączenia.

```
cd $ORACLE_HOME/network/admin
cat tnsnames.ora
```

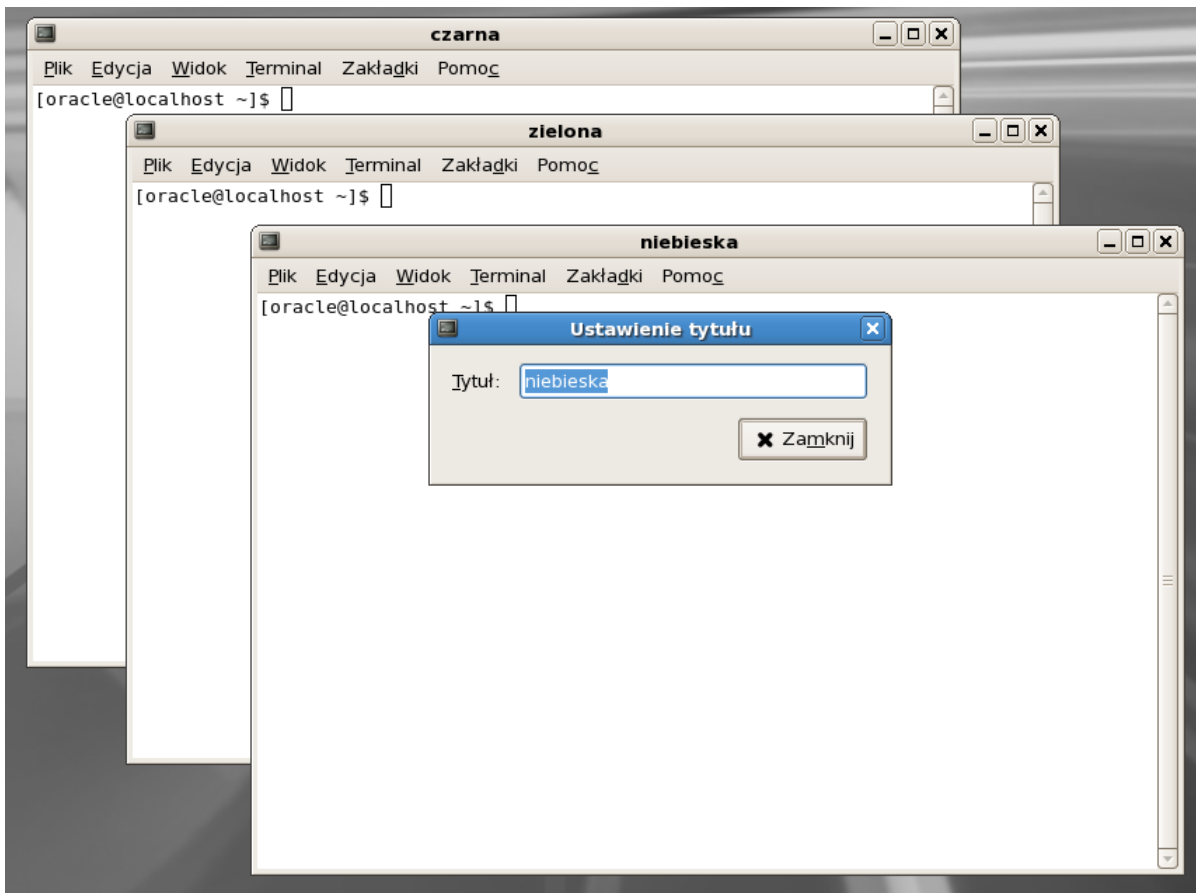
### 3. Łączy bazodanowe

Celem ćwiczenie jest przygotowanie łącza bazodanowego do odległej bazy danych.

1. Do wykonania dalszych ćwiczeń potrzebujemy trzy terminale, polecenia wydawane w pierwszy terminalu nazwiemy sesją czarną, polecenia wydawane w drugim terminalu nazwiemy sesją zieloną, a polecenia wydawane w trzecim terminalu nazwiemy sesją niebieską. Otwórz dwa dodatkowe terminale. W celu łatwiejszego rozróżnienia sesji



prorowadzonych w tych terminalach nazwiemy je, odpowiednio: *czarna*, *zielona* i *niebieska*. Aby zmienić nazwę terminala wybierz jego menu głównego pozycję Terminal, a następnie *Ustaw tytuł*.



2. W jednym oknie (sesja czarna) połącz się z lokalną bazą danych jako `scott`

```
sqlplus scott
```

3. W drugim oknie (sesja zielona) połącz się z bazą danych `dblab11g` korzystając z własnego konta (NAZWISKO\_I to Twoje nazwisko, znak podkreślenia, i pierwsza litera Twojego imienia)

```
sqlplus NAZWISKO_I@dblab11g
```

4. W sesji czarnej spróbuj utworzyć prywatne łącze bazodanowe o nazwie `dblab11g` do własnego konta w bazie danych `dblab11g`

```
create database link dblab11g  
connect to NAZWISKO_I identified by IMIE  
using 'dblab11g';
```

5. Otwórz nową sesję (niebieską) łącząc się z lokalną bazą danych jako administrator

```
sqlplus / as sysdba
```

6. W sesji niebieskiej nadaj użytkownikowi `scott` przywilej tworzenia łączy bazodanowych oraz przywilej tworzenia synonimów

```
grant create database link to scott;  
grant create synonym to scott;
```

7. W sesji czarnej spróbuj ponownie utworzyć łączy bazodanowe, tak jak w punkcie 4.
8. W sesji czarnej odczytaj zawartość tabeli `PRACOWNICY` w zdalnej bazie danych `dblab11g` poprzez utworzone łączy

```
select * from pracownicy@dblab11g;
```

9. Utwórz synonim `MPRACOWNICY` dla zdalnej tabeli `PRACOWNICY`. Odczytaj dane poprzez utworzony synonim

```
create synonym mpracownicy for pracownicy@dblab11g;  
select * from mpracownicy;
```

#### ***4. Transakcje rozproszone***

Celem ćwiczenia jest obserwacja wykonania rozproszonej transakcji, która modyfikuje dane w lokalnej i zdalnej bazie danych.

1. W sesji niebieskiej nadaj użytkownikowi `scott` przywilej testowania awarii i ręcznego naprawiania transakcji rozproszonych (użytkownicy w bazie `dblab11g` już posiadają ten przywilej)

```
grant force any transaction to scott;
```

2. W sesji niebieskiej wyłącz automatyczne naprawianie problemów z transakcjami rozproszonymi

```
alter system disable distributed recovery;
```

3. Wykonaj transakcję rozproszoną modyfikującą dane w bazie zdalnej i lokalnej (bez awarii)

```
update emp set sal = sal+10  
where empno = 7369;  
  
update pracownicy@dblab11g set placa_pod = placa_pod+10  
where id_prac = 100;  
  
commit;
```

4. Wykonaj ponownie tę samą rozproszoną transakcję symulując awarię po fazie `prepare` procesu 2-Phase Commit

```
update emp set sal = sal+10
```

```
where empno = 7369;
```

```
update pracownicy@dblab11g set placa_pod = placa_pod+10  
where id_prac = 100;
```

```
commit comment 'ORA-2PC-CRASH-TEST-4';
```

5. Spróbuj ponownie w sesji czarnej zmodyfikować dane najpierw w lokalnej. Zaobserwuj informacje o zablokowaniu danych przez transakcję rozproszoną w stanie wątpliwym.

```
update emp set sal = sal+10  
where empno = 7369;
```

6. W sesji niebieskiej podejrzuj informacje o transakcjach rozproszonych, które nie mogą zostać zakończone

```
select GLOBAL_TRAN_ID, STATE, DB_USER, FAIL_TIME  
from dba_2pc_pending;
```

7. W sesji niebieskiej włącz automatyczne naprawianie problemów z transakcjami rozproszonymi

```
alter system enable distributed recovery;
```

8. W sesji niebieskiej ponownie podejrzuj informacje o transakcjach rozproszonych, które nie mogą zostać zakończone

```
select * from dba_2pc_pending;
```

9. Jeszcze raz spróbuj ponownie w sesji czarnej zmodyfikować dane najpierw w lokalnej a później w zdalnej bazie danych. Zauważ, że tym razem operacje zakończyły się powodzeniem

```
update emp set sal = sal+10  
where empno = 7369;
```

```
update pracownicy@dblab11g set placa_pod = placa_pod+10  
where id_prac = 100;
```

```
commit;
```

## ***5. Migawki (perspektywy materializowane)***

Celem ćwiczenia jest zapoznanie się z mechanizmem asynchronicznej replikacji danych ze zdalnej do lokalnej bazy danych. Replikacja zostanie zaimplementowana za pomocą mechanizmu migawek.

1. W sesji niebieskiej nadaj użytkownikowi scott przywilej tworzenia migawek

```
grant create snapshot to scott;
```

2. W sesji niebieskiej sprawdź liczbę procesów umożliwiających automatyczne odświeżanie migawek

```
show parameter job_queue_processes;
```

3. Jeśli wartość parametru JOB\_QUEUE\_PROCESSES jest równa 0, to (w sesji niebieskiej) zmień jego wartość np. na 2

```
alter system set job_queue_processes=2;
```

4. W sesji czarnej utwórz migawkę REP\_ETATY będącą repliką zdalnej tabeli ETATY, odświeżaną w trybie COMPLETE bez automatycznego odświeżania.

```
create snapshot rep_etaty  
refresh complete  
as  
select * from etaty@dblab11g;
```

5. Odczytaj informacje o utworzonych przez siebie migawkach

```
select owner,name, refresh_mode, query from user_snapshots;
```

6. W sesji zielonej zmodyfikuj zawartość tabeli ETATY, dodając nowy etat „PORTIER”. Zatwierdź transakcję. W sesji czarnej wyświetl dane z migawki REP\_ETATY.

7. W sesji czarnej odśwież „ręcznie” migawkę REP\_ETATY w trybie COMPLETE. Wyświetl dane z migawki.

```
exec dbms_snapshot.refresh('SCOTT.REP_ETATY', 'C')  
select * from rep_etaty;
```

8. W sesji zielonej utwórz dziennik migawki na tabeli PRACOWNICY. Dziennik ten pozwoli na przyrostowe (FAST) odświeżanie migawek replikujących dane z tej tabeli.

```
create snapshot log on pracownicy;
```

9. W sesji czarnej utwórz migawkę REP\_PLACE replikującą identyfikatory, nazwiska, płace podstawowe i dodatkowe ze zdalnej tabeli PRACOWNICY. Migawka powinna być odświeżana w trybie FAST automatycznie co 6 minut.

```
create snapshot rep_place  
refresh fast  
next sysdate + 1/240  
as  
select id_prac, nazwisko, placa_pod, placa_dod  
from pracownicy@dblab11g;
```

10. W sesji czarnej wyświetl dane z migawki REP\_PLACE

```
select * from rep_place;
```

11. W sesji zielonej zmodyfikuj płacę któremuś z pracowników. Zatwierdź transakcję!

12. W sesji czarnej wyświetl dane z migawki REP\_PLACE. Następnie odśwież "ręcznie" migawkę REP\_PLACE w trybie FAST. Ponownie wyświetl dane z migawki REP\_PLACE

```
select * from rep_place;
exec dbms_snapshot.refresh('SCOTT.REP_PLACE', 'F')
select * from rep_place;
```

13. W sesji zielonej dodaj nowego pracownika. Zatwierdź transakcję!

14. W sesji czarnej wyświetlaj co jakiś czas zawartość migawki REP\_PLACE, czekając na automatyczne odświeżenie migawki.

15. W sesji czarnej utwórz grupę odświeżania zawierającą migawki REP\_ETATY i REP\_PLACE (pierwsze odświeżenie: natychmiast po utworzeniu, okres odświeżania: co 6 minut, zmiana grupy odświeżania dla migawek już automatycznie odświeżanych). Zatwierdź transakcję.

```
exec DBMS_REFRESH.MAKE (name => 'scott.rg_kadry', -
    list => 'scott.rep_etaty, scott.rep_place', -
    next_date => sysdate, -
    interval => 'sysdate + 1/240', -
    lax => true)
commit;
```

16. W sesji zielonej usuń nowego pracownika dodanego w punkcie 32. Zatwierdź transakcję!

17. W sesji czarnej odczytaj dane z migawki REP\_PLACE. Odśwież „ręcznie” grupę odświeżania. Ponownie odczytaj dane z migawki REP\_PLACE

```
select * from rep_place;
exec DBMS_REFRESH.REFRESH ('scott.rg_kadry')
select * from rep_place;
```

18. W sesji zielonej usuń z tabeli ETATY etat „PORTIER”. Zatwierdź transakcję.

19. W sesji czarnej wyświetlaj co jakiś czas zawartość migawki REP\_ETATY, czekając na automatyczne odświeżenie migawek z grupy odświeżania.

20. W sesji czarnej usuń grupę odświeżania, a następnie obie utworzone wcześniej migawki

```
exec DBMS_REFRESH.DESTROY ('scott.rg_kadry')  
drop snapshot rep_place;  
drop snapshot rep_etaty;
```

21. W sesji zielonej usuń dziennik migawki na tabeli PRACOWNICY

```
drop snapshot log on pracownicy;
```