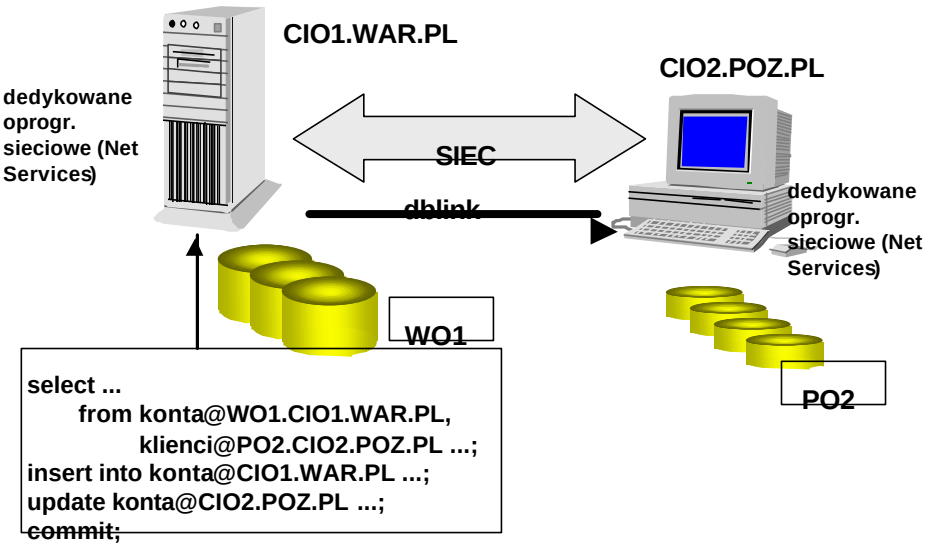
Rozproszone bazy danych Oracle

Ogólna architektura systemu



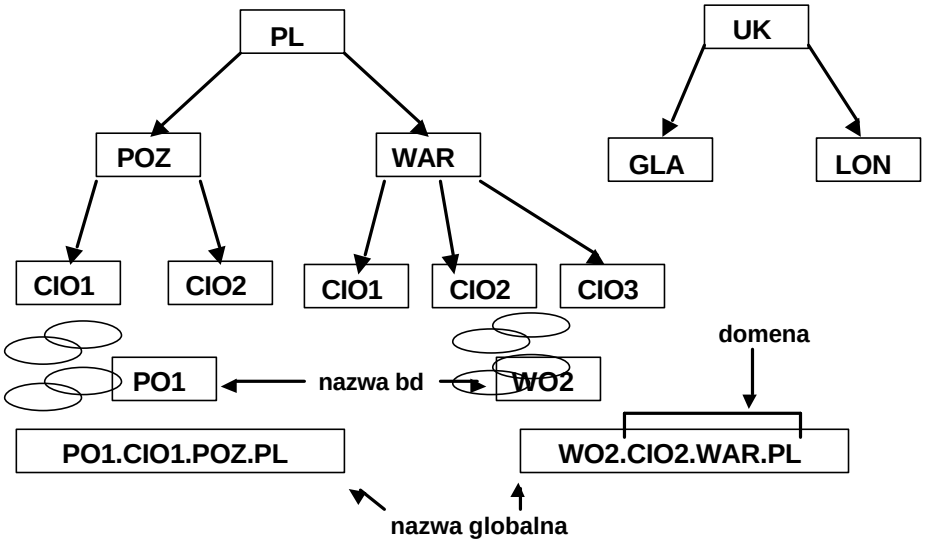
Komponenty architektury

- rozproszona bazy danych -> zbiór lokalnych baz danych w róznych wezlach sieci
  - z p. widzenia aplikacji stanowia jedna bd
  - autonomicznosc wezlów
- dedykowane oprogramowanie sieciowe (Oracle Net Services)
  - rozproszone transakcje
    - protokól zatwierdzania 2-fazowego
  - przezroczystosc lokalizacji bd (ang. location transparency)
- zbiór dostepnych baz danych w sieci
- autentykacja uzytkowników
  - lokalna autentykacja uzytkowników
  - globalna autentykacja uzytkowników
  - Oracle Security Server
- szyfrowanie danych
  - Advanced Networking Services
- zarzadzanie systemem rozproszonym
  - Oracle Enterprise Manager

Komponenty architektury (2)

- nazwa globalna bazy danych
  - kazda bd wchodzaca w sklad systemu rozproszonego jest identyfikowana unikalna nazwa globalna (ang. global database name)
- obiekty bazy danych
  - laczniki bazy danych
  - perspektywy
  - synonimy
  - migawki

Nazwy globalne

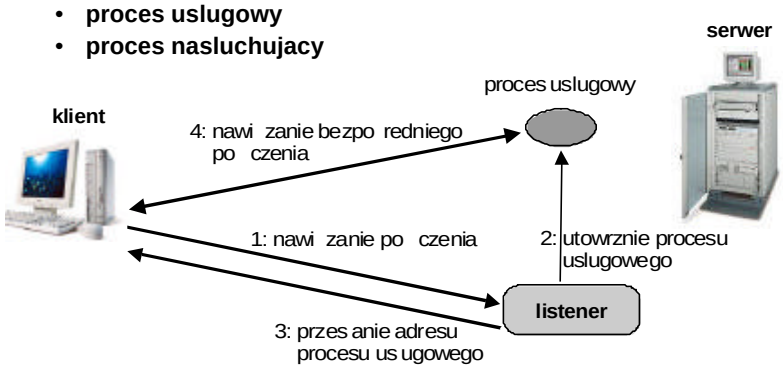


Nazwy globalne (2)

- nazwa bazy danych -> parametr kofiguracyjny DB_NAME
  - max. 8 znaków
- nazwa domeny -> parametr konfiguracyjny DB_DOMAIN
- parametr konfiguracyjny GLOBAL_NAMES
  - TRUE: dolaczenie bazodanowe musi miec nazwe identyczna z nazwa globalna bazy danych, na która wskazuje
    - zalecane
    - wymagane przy Advanced Replication Option
  - FALSE: nazwa dolaczenia i nazwa globalna bd moga byc rózne

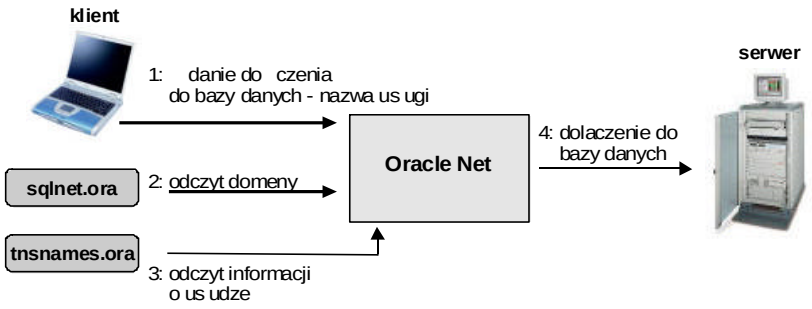
Net Services (1)

- **przezroczystosc lokalizacji**
- **niezaleznosc od systemu operacyjnego**
- **niezaleznosc od protokolów komunikacyjnych**
- **komunikacja aplikacja – baza danych**
  - **proces uslugowy**
  - **proces nasluchujacy**



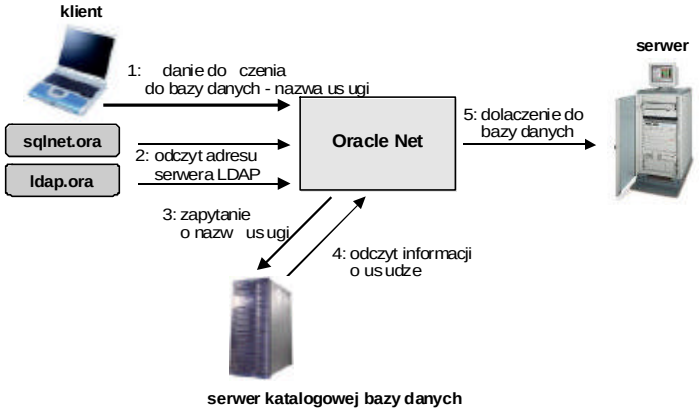
Net Services (1)

- **zbiór dostepnych baz danych w sieci**
  - **zapisany w pliku**



Net Services (2)

- zbiór dostepnych baz danych w sieci
  - zapisany w katalogu uslug (Oracle Names)
  - zapisany w katalogowej bazie danych (OID)



Net Services (3)

- zbiór dostepnych baz danych w sieci
  - adresowanie serwera
    - protokól TCP/IP
    - identyfikacja komputera: DNS, plik .hosts
    - pelna nazwa bazy danych musi byc identyczna z nazwa komputera, na którym zostala zainstalowana

Lacznik bazy danych

```
CREATE [PUBLIC] DATABASE LINK nazwa
CONNECT TO uzytkownik IDENTIFIED BY haslo
USING ‘nazwa.bazy.danych’;
```

```
create  database link LAB.WORLD
connect to scott identified by tiger
using 'LAB.WORLD';
```

- domyslnie tworzony lacznik prywatny
- tworzenie lacznika publicznego wymaga uprawnienia CREATE PUBLIC DATABASE LINK

```
select * from emp@LAB.WORLD;
```

```
select * from scott.emp@LAB.WORLD;
```

```
select * from usr1.accounts@LAB.WORLD;
```


Lacznik bazy danych (2)

```
update emp@LAB.WORLD set sal=sal*1.1 where deptno=10;
```

```
create table employees as select * from emp@LAB.WORLD;
```

- lacznik wskazujacy na biezacego uzytkownika
  - w bazie zdalnej musi istniec identyczny uzytkownik z identycznym haslem
- lacznik publiczny

```
create public database link lab92
connect to scott identified by tiger
using 'lab92.ii.pp';
```

```
scott> create public
database link lab92 using
'lab92.ii.pp';
```

```
bart> create database link
lab92 connect to demo
identified by demo;
```

Lacznik bazy danych (3)

- pierwsze odwolanie do lacznika otwiera je
- lacznik pozostaje otwarty do konca sesji lub jawnego jego zamkniecia poleceniem:

```
ALTER SESSION CLOSE DATABASE LINK nazwa;
```

  - przed zamknieciem lacznika nalezy zakonczyc transakcje korzystajaca z niego

- usuniecie lacznika

```
DROP [PUBLIC] DATABASE LINK nazwa;
```


Infromacje o utworzonych lacznikach

- ALL_DB_LINKS, DBA_DB_LINKS
  - OWNER
  - DB_LINK (nazwa)
  - USERNAME (klauzula CONNECT TO)
  - HOST (klauzula USING)
  - CREATED
- USER_DB_LINKS
  - DB_LINK
  - USERNAME
  - PASSWORD
  - HOST
  - CREATED

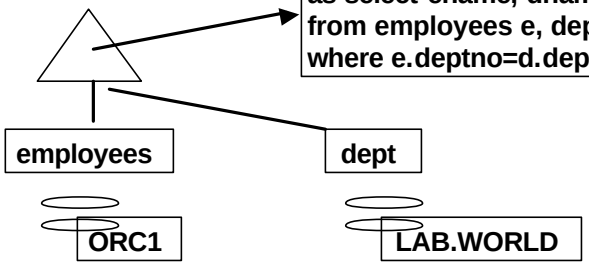
Ograniczenie liczby dolaczen

- ograniczenie liczby jednoczesnie aktywnych dolaczen w jednej sesji
  - parametr konfiguracyjny OPEN_LINKS
  - wartosc: 0-255
  - domyslnie: 4

Przezroczystosc lokalizacji

- perspektywa

```
create view emp_dept
as select ename, dname
from employees e, dept@LAB.WORLD d
where e.deptno=d.deptno;
```



- przeniesienie tabeli dept do innej bazy danych:
  - utworzenie nowego lacznika wskazujacego na dept
  - zmiana definicji perspektywy z uwzgl. nowego lacznika
  - aplikacje odwolujace sie do perspektywy pozostaja niezmienione

Przezroczystosc lokalizacji (2)

- program skladowany
  - lokalny program odwolujacy sie do danych w zdalnej bazie
- synonim

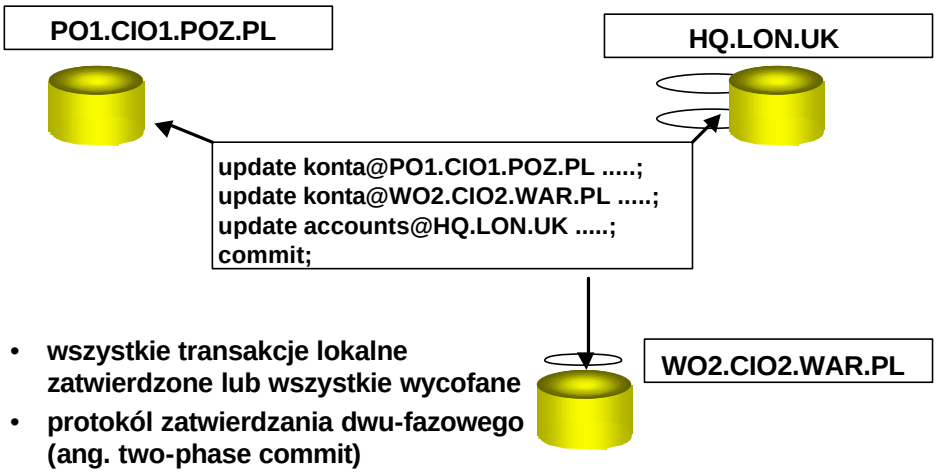
```
create synonym s_emp for emp@LAB.WORLD;

select * from s_emp;
```

Informacje o utworzonych synonimach

- DBA_SYNONYMS, ALL_SYNONYMS
  - OWNER, SYNONYM_NAME, TABLE_OWNER, TABLE_NAME, DB_LINK
- USER_SYNONYMS

Rozproszona transakcja



- wszystkie transakcje lokalne zatwierdzone lub wszystkie wycofane
- protokół zatwierdzania dwu-fazowego (ang. two-phase commit)
  - faza przygotowania
  - faza zatwierdzania

"Aktorzy"

- koordynator globalny: wezel sieci, w którym zainicjowano tranaskcje rozproszona
- uczestnik: wezel sieci z transakcja lokalna
- wezel zatwierdzania (ang. commit point site)
  - inicjowanie zatwierdzania lub wycofywania transakcji zgodnie z komunikatem od koordynatora globalnego
  - wybierany przez administratora systemu
    - parametr konfiguracyjny COMMIT_POINT_STRENGTH
    - wartosc 0-255; wartosc domyslna zalezna od systemu operacyjnego
    - odzwierciedla ilosc danych krytycznych w wezle
    - odzwierciedla niezawodnosc wezla
  - wezel o najwyzszej wartosci COMMIT_POINT_STRENGTH jest wezlem zatwierdzania
  - zawiera status zatwierdzania transakcji rozproszonej
    - odczytywany przez transakcje lokalne

“Aktorzy” (2)

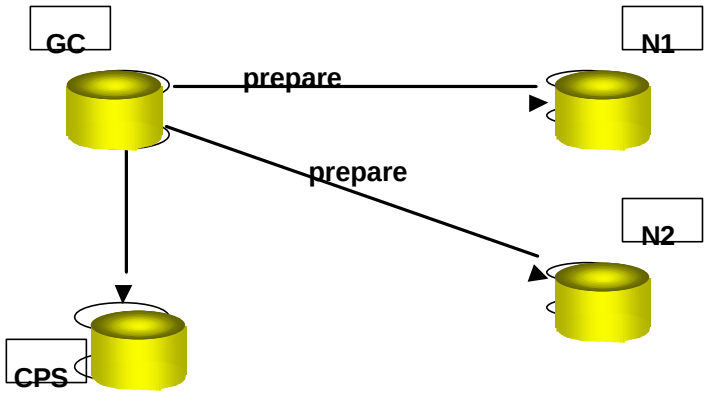
- wezel zatwierdzania
  - transakcja rozproszona jest uznawana za zatwierdzona jezeli zostanie zatwierdzona w wezle zatwierdzania, nawet jesli pozostale wezly jeszcze nie zatwierdzily swoich transakcji lokalnych

Protokól zatwierdzania 2-fazowego

- two-phase commit protocol (2PC)
- faza przygotowania (prepare)
- faza zatwierdzania (commit)
- faza zakonczenia (forget)

2PC - faza przygotowania (koordynator)

- koordynator globalny okresla wezel zatwierdzania
- koordynator globalny wysyla do uczestników zadanie przygotowania do zatwierdzania



Faza przygotowania - uczestnik

- odbiór komunikatu od koordynatora globalnego zadajacego przygotowania do zatwierdzania
- wyslanie zadania przygotowania do zdalnych wezlów, do których odwoluje sie uczestnik
- w przypadku braku modyfikacji danych -> wyslanie do koordynatora globalnego komunikatu READ-ONLY
- zapisanie zawartosci bufora dziennika powtórzen do pliku dziennika (on line redo log)
- inne zdalne wezly dolaczone do uczestnika zglosily gotowosc i sam uczestnik jest gotów -> wyslanie komunikatu PREPARED do koordynatora globalnego
  - w przeciwnym przypadku
    - wycofanie lokalnej transakcji
    - wyslanie ABORT

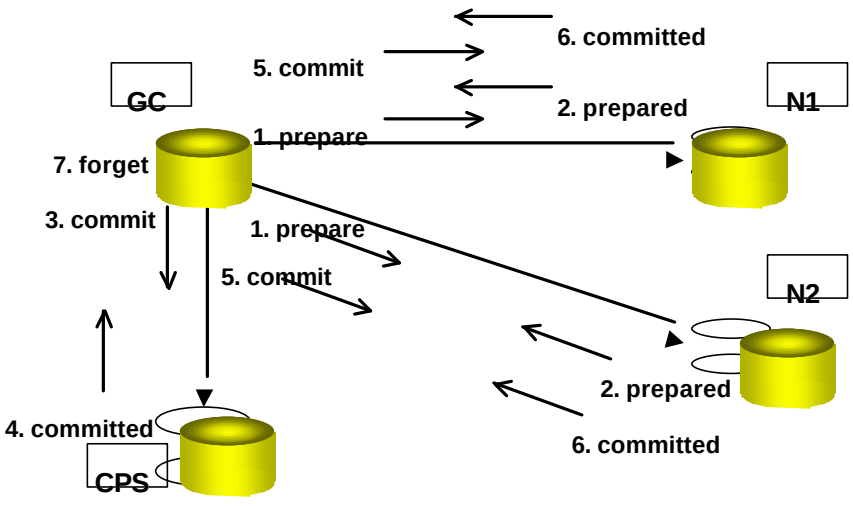
2PC - faza zatwierdzania

- koordynator globalny odbiera potwierdzenia od uczestników
  - PREPARED
  - READ-ONLY (brak modyfikacji)
  - ABORT (niemozliwosc przygotowania do zatwierdzania)
- jesli wszyscy odpowiedzieli PREPARED -> koordynator globalny wysyla zadanie zatwierdzenia transakcji do wezla zatwierdzania
  - wezel zatwierdzania zatwierdza tranaskcje i wysyla komunikat do koordynatora globalnego
  - koordynator globalny wysyla zadanie zatwierdzenia do pozostalych wezlów
- jesli choc jeden uczestnik odpowiedzial ABORT -> koordynator wysyla zadanie wycofania transakcji do wezla zatwierdzania
  - wezel zatwierdzania wycofuje transakcje i wysyla komunikat do koordynatora globalnego
  - koordynator wysyla zadanie wycofania do uczestników

Faza zatwierdzania - uczestnik

- odbiór od koordynatora globalnego komunikatu zadajacego zatwierdzenia transakcji
- zatwierdzenie lokalnej transakcji
- zwolnienie blokad
- zapis informacji o zatwierdzeniu w pliku dziennika powtórzen

2PC - podsumowanie



Ograniczenie liczby transakcji rozproszonych

- parametr konfiguracyjny DISTRIBUTED_TRANSACTIONS
- DISTRIBUTED_TRANSACTIONS=0
  - brak transakcji rozproszonych
  - nie startuje proces drugoplanowy RECO

Problemy sprzetowo-programowe

- w czasie fazy COMMIT (ROLLBACK) nastepuje awaria sieci, wezla lub zdalnej bazy danych
  - nie wszystkie wezly zatwierdzily (wycofaly)
  - nie wszystkie wezly potwierdzily zakonczenie operacji
  - transakcja rozproszona w stanie “in-doubt”
- automatyczne odtwarzanie transakcji rozproszonej (proces RECO) w stanie “in-doubt” po usunieciu awarii
  - wynik: wszystkie wezly zatwierdza lub wszystkie wycofaja

Blokowanie przez transakcje rozproszona

- transakcja rozproszona w stanie “in-doubt” blokuje dane
- inna transakcja zada blokdy na tych danych
  - ORA-01591: lock held by in-doubt distributed transaction <id>
  - polecenie zadajace blokady jest wycofywane i moze byc powtórzone

Blokowanie transakcji rozproszonej

- tranasakcja rozproszona zada w zdalnym wezle blokady danych zablokowanych wczesniej przez inna transakcje
  - czas oczekiwania na zwolnienie blokad -> parametr konfiguracyjny DISTRIBUTED_LOCK_TIMEOUT, wyrazony w sekundach
    - wartosc: 1- nieograniczona; domyslnie 60
  - po przekroczeniu czasu oczekiwania polecenie jest wycofywane i moze byc powtórzone
    - ORA-02049: time-out distributed transaction waiting for lock

Manualne zatwierdzanie lub wycofywanie transakcji w stanie “in-doubt”

- blokowane dane musza byc natychmiast zwolnione
- blokowanie segmentu wycofania
- czas usuniecia awarii sprzetowej bardzo dlugi

Manualne zatwierdzanie lub wycofywanie transakcji w stanie “in-doubt” (2)

- uzyskanie informacji, czy transakcja lokalna wchodzaca w sklad transakcji rozproszonej powinna byc zatwierdzona, czy wycofana -> perspektywa SYS.DBA_2PC_PENDING

```
ALTER SESSION ADVISE COMMIT;
   INSERT INTO emp@LAB.WORLD ... ;
        /* zalecane zatwierdzenie transakcji lokalnej w wezle LAB.WORLD */

ALTER SESSION ADVISE ROLLBACK;
   DELETE FROM emp@ORC1.WORLD ... ;
        /* zalecane wycofanie transakcji lokalnej w wezle ORC1.WORLD */

ALTER SESSION ADVISE NOTHING;

DBA_2PC_PENDING.ADVICE ──►  [ ]   [R]   [C]
```

Manualne zatwierdzanie lub wycofywanie transakcji w stanie “in-doubt” (3)

- uzyskanie informacji na temat tranaskcji rozproszonej inicjujacej COMMIT -> perspektywa SYS.DBA_2PC_PENDING

COMMIT COMMENT ‘komentarz;

↓

DBA_2PC_PENDING.TRAN_COMMENT

- komentarz
  - moze zawierac np. rodzaj aplikacji inicjujacej COMMIT, rodzaj operacji
  - max. 50 znaków

```
set transaction name 'modyfikacja salda';
```

```
SQL> select name, status from v$transaction;
NAME                 STATUS
-------------------- ----------------
modyfikacja salda    ACTIVE
```


Manualne zatwierdzanie lub wycofywanie transakcji w stanie “in-doubt” (4)

- uzytkownik realizujacy transakcje lokalna otrzymuje komunkat:

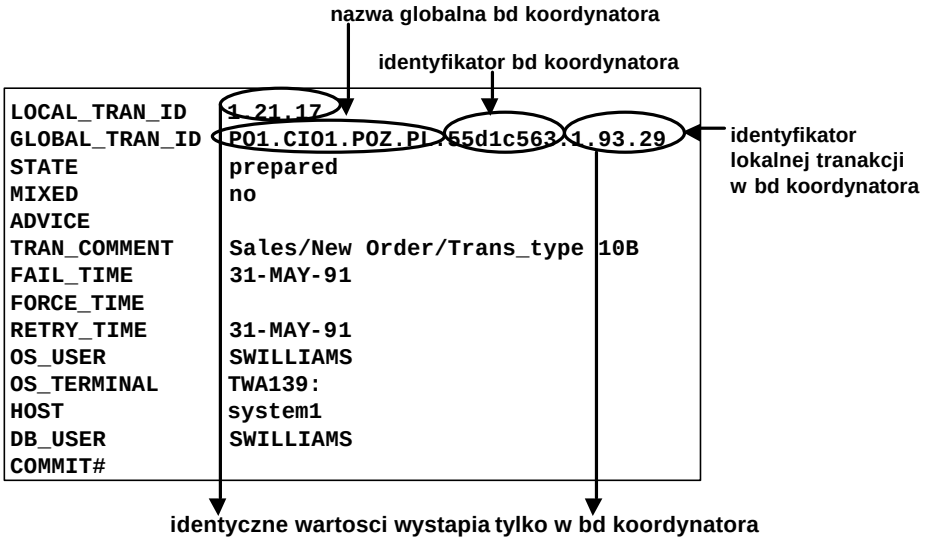
```
ORA-01591: lock held by in-doubt distributed transaction 1.21.17
```

identyfikator lokalnej transakcji bedacej czescia trans. rozproszonej

- analiza zawartosci DBA_2PC_PENDING w bazie lokalnej

```
SELECT * FROM sys.dba_2pc_pending
WHERE local_tran_id = '1.21.17';
```

Manualne zatwierdzanie lub wycofywanie transakcji w stanie “in-doubt” (5)

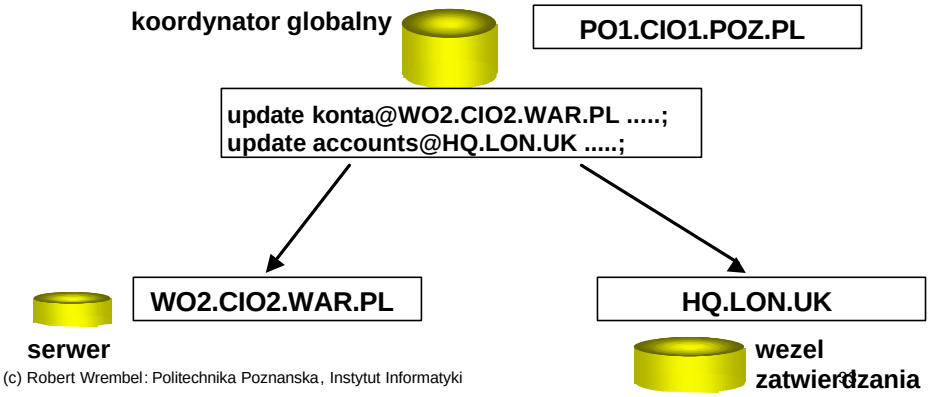


Manualne zatwierdzanie lub wycofywanie transakcji w stanie “in-doubt” (6)

- **atrybut STATE:**
  - **collecting**
    - **wystapi tylko dla koordynatora**
    - **gromadzenie potwierdzen z wezlów**
  - **prepared**
    - **transakcja przygotowana do zatwierdzenia**
    - **potwierdzenie przygotowania wyslane do koordynatora lub nie**
  - **committed**
    - **transakcja w wezle zatwierdzona**
  - **forced commit**
    - **transakcja w wezle zatwierdzona manualnie**
  - **forced abort**
    - **transakcja w wezle wycofana manualnie**

Manualne zatwierdzanie lub wycofywanie transakcji w stanie “in-doubt” (7)

- znalezienie wezla zatwierdzania -> zawiera informacje czy transakcja rozproszona powinna zostac zatwierdzona (wycofana)
- perspektywa SYS.DBA_2PC_NEIGHBORS



Manualne zatwierdzanie lub wycofywanie transakcji w stanie “in-doubt” (8)

WO2.CIO2.WAR.PL

ORA-01591: lock held by in-doubt distributed transaction 1.21.17

```
SELECT * FROM sys.dba_2pc_neighbors
WHERE local_tran_id = '1.21.17';
```

LOCAL_TRAN_ID	1.21.17
IN_OUT	in
DATABASE	PO1.CIO1.POZ.PL
DBUSER_OWNER	SCOTT
INTERFACE	N
DBID	000003F4
SESS#	1
BRANCH	0100

wezel jest serwerem zadania bazy danych

dolaczenie zrealizowane z konta

WO2.CIO2.WAR.PL nie jest wezlem zatwierdzania; zaden z wezlów podleglych nie jest wezlem zatwierdzania

Manualne zatwierdzanie lub wycofywanie transakcji w stanie “in-doubt” (9)

WO2.CIO2.WAR.PL

- **znalezienie identyfikatora transakcji globalnej w wezle WO2.CIO2.WAR.PL na podstawie indntyfikatora transakcji lokalnej**

```
SELECT local_tran_id, global_tran_id
FROM sys.dba_2pc_pending
WHERE local_tran_id = '1.21.17';
```

```
LOCAL_TRAN_ID   GLOBAL_TRAN_ID
-------------   --------------------------------
1.21.17         PO1.CIO1.POZ.PL.55d1c563.1.93.29
```


Manualne zatwierdzanie lub wycofywanie transakcji w stanie “in-doubt” (10)

PO1.CIO1.POZ.PL

- **znalezienie identyfikatora transakcji lokalnej w wezle PO1.CIO1.POZ.PL na podstawie id transakcji globalnej (zob. slajd “in-doubt” 4 i 5)**

```
SELECT local_tran_id FROM sys.dba_2pc_pending
WHERE global_tran_id='PO1.CIO1.POZ.PL.55d1c563.1.93.29';
```

```
LOCAL_TRAN_ID   GLOBAL_TRAN_ID
-------------   --------------------------------
1.93.29         PO1.CIO1.POZ.PL.55d1c563.1.93.29
```

- **wyswietlenie zawartosci SYS.DBA_2PC_NEIGHBORS**

```
SELECT * FROM dba_2pc_neighbors
WHERE local_tran_id = '1.93.29';
```

Manualne zatwierdzanie lub wycofywanie transakcji w stanie “in-doubt” (11)

PO1.CIO1.POZ.PL

```
LOCAL_TRAN_ID          1.93.29
IN_OUT                 OUT
DATABASE               WO2.CIO2.WAR.PL
DBUSER_OWNER           SWILLIAMS
INTERFACE              N
DBID                   55d1c563
SESS#                  1
BRANCH                 1

LOCAL_TRAN_ID          1.93.29
IN_OUT                 OUT
DATABASE               HQ.LON.UK
DBUSER_OWNER           ALLEN
INTERFACE              C
DBID                   00000390
SESS#                  1
BRANCH                 1
```

wezel zglasza zadanie do serwera

nie jest wezlem zatwierdzania

wezel zglasza zadanie do serwera

HQ.LON.UK jest wezlem zatwierdzania

Manualne zatwierdzanie lub wycofywanie transakcji w stanie “in-doubt” (12)

HQ.LON.UK

- odczytanie statusu transakcji w wezle zatwierdzania

```
SELECT local_tran_id, global_tran_id, state, commit#
FROM dba_2pc_pending
WHERE global_tran_id = 'PO1.CIO1.POZ.PL.55d1c563.1.93.29';
```

```
LOCAL_TRAN_ID          1.45.13
GLOBAL_TRAN_ID         SALES.ACME.COM.55d1c563.1.93.29
STATE                  COMMIT
COMMIT#                129314
```

- nalezy zatwierdzic transakcje lokalne we wszystkich wezlach

Manualne zatwierdzanie lub wycofywanie transakcji w stanie “in-doubt” (13)

- uprawnienia systemowe
  - FORCE TRANSACTION, FORCE ANY TRANSACTION

Czas aktywnosci dolaczenia bazodanowego w przypadku awarii transakcji rozproszonej

- DISTRIBUTED_RECOVERY_CONNECTION_HOLD_TIME
  - czas (w sekundach) przez który dolaczenie bazodanowe pozostaje aktywne w przypadku niemozliwosci zakonczenia transakcji rozproszonej
  - domyslnie 200 sekund

Crash testy

- symulacja 10 typów awarii transakcji rozproszonej

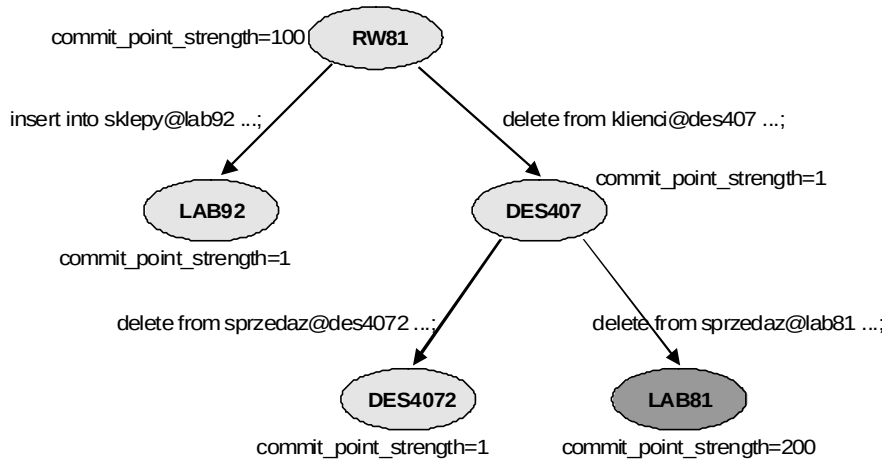
COMMIT COMMENT ‘ORA-2PC-CRASH-TEST-n’;

1 : Crash commit point site after collect
2 : Crash non-commit point site after collect
3 : Crash before prepare (non-commit point site)
4 : Crash after prepare (non-commit point site)
5 : Crash commit point site before commit
6 : Crash commit point site after commit
7 : Crash non-commit point site before commit
8 : Crash non-commit point site after commit
9 : Crash commit point site before forget
10: Crash non-commit point site before forget

Uwaga: wylaczyc automatyczne odtwarzanie transakcji rozproszonej we wszystkich wezlach uczestniczacych w tej transakcji

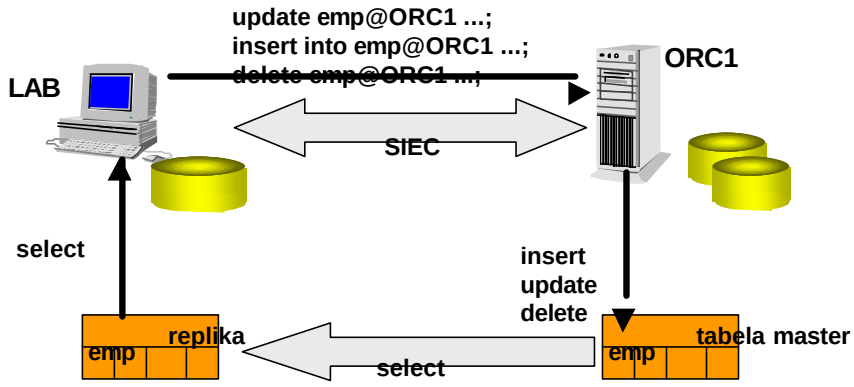
ALTER SYSTEM DISABLE DISTRIBUTED RECOVERY;

Cwiczenie



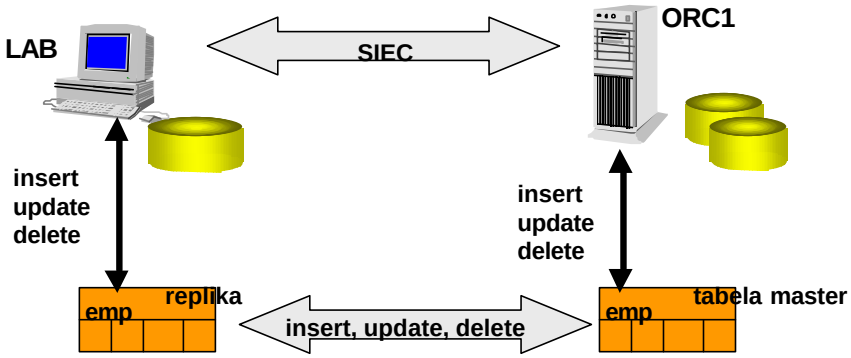
Replikacja danych

- **standardowa**



Replikacja danych (2)

- zaawansowana (Oracle Advanced Replication Option)



Migawka (ang. snapshot)

- kopia tabel znajdujacych sie w odleglych bazach danych
- standardowo tylko do odczytu
- przywileje:
  - CREATE SNAPSHOT, CREATE TABLE, CREATE VIEW
  - CREATE ANY SNAPSHOT
- rodzaje migawek
  - PRIMARY KEY
    - tabela master musi posiadac wlaczone ograniczenie PRIMARY KEY
    - klauzula SELECT musi zawierac wszystkie atrybuty wchodzace w sklad klucza podstawowego tabeli master
  - ROWID
- migawka -> tabela (+ perspektywa) + indeksy

Migawka (2)

```
create snapshot nazwa_migawki
refresh sposób_odswiezania
        start with data_pierwszego_odswiezenia
        next czestotliwosc_odswiezania
        with typ_migawki
as zapytanie;
```

- **migawka typu prostego**
  - **bazujaca na jednej tabeli master**
  - **brak klauzul: GROUP BY, CONNECT BY, DISTINCT**
  - **brak funkcji, polaczen, operatorów zbiorowych**
- **migawka typu zlozonego**
- **odswiezanie przyrostowe**
  - **migawka typu prostego**
  - **zapytanie z polaczeniem zastapione podzapytaniem skorelowanym**

Odswiezanie migawki

- **odswiezanie przyrostowe**
  - **migawka typu prostego**
  - **zapytanie z polaczeniem zastapione podzapytaniem skorelowanym**

```
select sk.nazwa, sk.sklep_id
from scott.sklepy@lab81.ii.pp sk,
scott.sprzedaz@lab81.ii.pp sp
where sp.sklep_id=sk.sklep_id
and sp.produkt_id=100
and sp.data='23.01.2002'
and sp.l_sztuk=2;
```

```
select sk.nazwa, sk.sklep_id
from scott.sklepy@lab81.ii.pp sk
where exists
        (select sp.sklep_id
         from scott.sprzedaz@lab81.ii.pp sp
         where sp.sklep_id=sk.sklep_id
         and sp.produkt_id=100
         and sp.data='23.01.2002'
         and sp.l_sztuk=2);
```

Odswiezanie migawki (2)

- **odswiezanie przyrostowe**
  - **migawka wyliczajaca agregaty: count, sum, avg, variance, stdev**
    - **dziennik utworzony z klauzula *including new values***
    - **dziennik zawiera wszystkie atrybuty wymienione po *select*, równiez bedace argumentami wywolania f. grupowych**
    - **count zawsze wyliczany w zapytaniu, gdy wyliczne *sum*, *avg*, *variance*, *stdev***

```
create materialized view mv_suma_sprzedazy
build immediate
refresh fast
start with sysdate
next sysdate+(1/(24*60*30))
as
select sklep_id, produkt_id, sum(l_sztuk), sum(l_sztuk*cena_jedn),
      count(l_sztuk), count(l_sztuk*cena_jedn), count(*)
from sprzedaz@lab92
group by sklep_id, produkt_id;
```


Tworzenie migawki

```
CREATE SNAPSHOT [schemat.]migawka
[ parametry_fizyczne ]
[ TABLESPACE nazwa_przestrzeni ]
[ USING INDEX
            [ parametry_fizyczne ]
            [ TABLESPACE nazwa_przestrzeni ] ]
[ REFRESH { FAST | COMPLETE | FORCE } ]
[ WITH { PRIMARY KEY | ROWID } ]
[ START WITH 'data' ]
[ NEXT 'data' ]
[ USING [ LOCAL ROLLBACK SEGMENT rbs ]
        [ MASTER ROLLBACK SEGMENT rbs ] ]
AS SELECT ...;
```

```
CREATE MATERIALIZED VIEW [schemat.]nazwa ...
```

Odswiezanie migawki

- sposób odswiezania
  - REFRESH FAST -> odswiezanie przyrostowe
    - dla migawek prostych
    - musi istniec SNAPSHOT LOG dla tabeli master
  - REFRESH COMPLETE -> odswiezanie pelne
  - REFRESH FORCE -> automatyczny wybór metody odswiezania; jezeli mozliwe to Oracle wybiera FAST
- okres odswiezania
  - START WITH -> data pierwszego odswiezenia
  - NEXT -> wyrazenie okreslajace czestotliwosc odswiezania

Odswiezanie automatyczne

- musi byc wyspecyfikowany parametr NEXT
- okreslenie czestoliwosci odswiezania
  - REFRESH FAST START WITH sysdate NEXT sysdate+1
  - REFRESH FAST NEXT sysdate+1

Odswiezanie automatyczne (2)

- wlaczenie procesu odpowiedzialnego za odswiezanie
  - parametr konfiguracyjny *JOB_QUEUE_PROCESSES* -> wartosc {1, ..., 36}, domyslnie 0
  - procesy drugoplanowe SNP_0 - SNP_9

Odswiezanie manualne

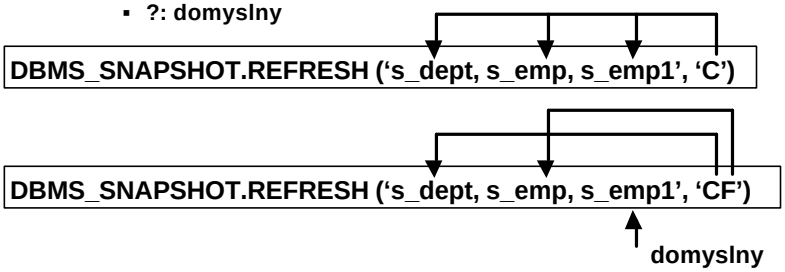
- wylaczenie procesu odpowiedzialnego za odswiezanie
  - *JOB_QUEUE_PROCESSES* =0
- brak parametru NEXT
  - REFRESH FAST START WITH sysdate
  - migawka odswiezona raz, w momencie jej tworzenia
- pakiet DBMS_SNAPSHOT
- pakiet DMBS_MVIEW (synonim do DBMS_SNAPSHOT)

Odswiezanie manualne (2)

- **procedura DBMS_SNAPSHOT.REFRESH**

DBMS_SNAPSHOT.REFRESH (‘sn_1, sn_2, ..., sn_n’, ‘metoda’)

  - **sn_1, sn_2, ..., sn_n: migawki**
  - **metoda: metoda odswiezania**
    - **f lub F: FAST**
    - **c lub C: COMPLETE**
    - **?: domyslny**



Moment odswiezania

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

- ***on commit* mozna stosowac jedynie, gdy:**
  - **zapytanie korzysta z tabel lokalnych**
  - **migawek opartych o jedna tabele, bez wyliczania agregatów**
  - **migawek, których zapytanie wyznacza agregaty w oparciu o pojedyncza tabele**
  - **migawek których zapytanie wykorzystuje laczenie tabel, ale bez wyliczania agregatów**
- **brak odswiezania**

```
create materialized view mv_test
never refresh
as select * from user1.sklepy@dbl1;
```

Przyklad

```
create snapshot sn_emp
pctfree 30 pctused 30
storage
    (initial 10K next 10K pctincrease 0 minextents 1 maxextents 10)
tablespace usr
refresh fast
start with sysdate+(1/(24*60))
next sysdate+(1/(24*60*6))
with rowid
using local rollback segment rb1 master rollback segment rb04
as select * from emp@lab.world;
```

odswiezanie co 10 sek.

```
SNAP$_SN_EMP    -> TABLE
I_SNAP$_SN_EMP  -> INDEX
SN_EMP          -> VIEW
```

zawiera kolumne M_ROW$$ przechowujaca ROWID rekordów tabeli emp@lab.world

na kolumnie SNAP$_SN_EMP.M_ROW$$

Moment wypelnienia danymi

- ***build immediate***
- ***build deferred***

```
create snapshot mv_sprzedaz_1
build deferred
refresh force
start with sysdate + (1/(24*6))
next sysdate+(1/(24*60))
with primary key
as
    select produkt_id, l_sztuk, cena_jedn, data, sklep_id
    from user1.sprzedaz@dbl1
    where sklep_id=1;
```

Modyfikowanie migawki

```
ALTER SNAPSHOT [schemat.]migawka
[ parametry_fizyczne ]
[ USING INDEX [ parametry_fizyczne ]
[ REFRESH { FAST | COMPLETE | FORCE } ]
                                        [{on demand | on commit]}
[ WITH PRIMARY KEY ]
[ START WITH 'data' ]
[ NEXT 'data' ]
[ USING  MASTER ROLLBACK SEGMENT rbs ];
```

- **parametry_fizyczne**
  - **bloku: PCTFREE, PCTUSED (nie dla indeksu), INITRANS, MAXTRANS**
  - **rozszerzen: STORAGE**
    - **NEXT, MINEXTENTS, MAXEXTENTS, PCTINCREASE**

```
alter snapshot sn_emp1
pctfree 20 pctused 40 initrans 4
storage (next 20K pctincrease 0
         minextents 1
         maxextents 20)
refresh complete
start with sysdate
next sysdate+1/(24*60*10)
with primary key
using master rollback segment rb03;
```


Dziennik migawki (ang. snapshot log)

- **tabela zwiazana z tabela master migawki**
- **przechowuje zmiany dokonane na danych tabeli master**
- **wykorzystywany do odswiezania przyrostowego**
- **tworzenie:**

```
create snapshot log
on tabela_bazowa
[with {primary key |
       ROWID |
       primary key, ROWID |
       ROWID (lista_kolumn_filtrujacych) |
       primary_key (lista_kolumn_filtrujacych)}]
[{ including new values | excludign new values }];
```

Dziennik migawki (2)

- **WITH PRIMARY KEY: dla rekordów uaktualninych wartosci atrybutów wchodzacych w sklad klucza sa rejestrowane w dzienniku**
- **WITH ROWID: ROWID rekordów uaktualnionych rejestrowane w dzienniku**
- **WITH PRIMARY KEY, ROWID: w dzienniku rejestrowane zarówno wartosci atr. kluczowych, jak i ROWID**
- **kolumna filtrujaca: atrybut wystepujacy w klauzuli *where* zapytania definiujacego migake**

- **including new values**
  - **konieczne dla migawek odswiezanych przyrostowo zawierajacych agregaty**

```
select sk.nazwa, sk.sklep_id
from scott.sklepy@lab81.ii.pp sk
where exists
        (select sp.sklep_id
         from scott.sprzedaz@lab81.ii.pp sp
         where sp.sklep_id=sk.sklep_id
         and sp.produkt_id=100
         and sp.data='23.01.2002'
         and sp.l_sztuk=2)
```


Dziennik migawki (3)

```
create materialized view mv_suma_sprzedazy
build immediate
refresh fast
start with sysdate
next sysdate+(1/(24*60*30))
as
select sklep_id, produkt_id, sum(l_sztuk), sum(l_sztuk*cena_jedn),
      count(l_sztuk), count(l_sztuk*cena_jedn), count(*)
from sprzedaz@lab92
group by sklep_id, produkt_id;
```

```
create materialized view log on sprzedaz
with primary key, rowid (l_sztuk, cena_jedn)
including new values;
```

Przyklad

```
create snapshot log on scott.emp
pctfree 30 pctused 30 initrans 2 maxtrans 10
storage
    (initial 10K next 10K pctincrease 0 minextents 1 maxextents 10)
tablespace lab_dane
with primary key, rowid;
```

```
MLOG$_EMP
Name                 Type
-------------------- ------------
EMPNO                NUMBER(4)
M_ROW$$              VARCHAR2(255)
SNAPTIME$$           DATE
DMLTYPE$$            VARCHAR2(1)
OLD_NEW$$            VARCHAR2(1)
CHANGE_VECTOR$$      RAW(255)
```


Przyklad (2)

```
insert into emp values (1000, 'BOND', 'MANAGER', NULL, '01-JAN-99',
                        6000, 500, 30);

update emp set comm=comm+300 where empno=7839;

delete from emp where empno=7698;
```

```
select * from mlog$_emp;
EMPNO M_ROW$$            SNAPTIME$ DML    OLD_  CHANGE_
                                   TYPE$$ NEW$$ VECTOR$$
----- ------------------ --------- ------ ----- --------
 1000 AAAApTAACAAAAn5AAD 01-JAN-00 I      N     FEFF
 7839 AAAApTAACAAAAn5AAA 01-JAN-00 U      U     8000
 7698 AAAApTAACAAAAn5AAB 01-JAN-00 D      O     0000
```

Modyfikowanie dziennika migawki

```
alter snapshot log
on tabela_bazowa
add {  primary key |
       ROWID |
       ROWID (lista_kolumn_filtrujacych) |
       primary_key (lista_kolumn_filtrujacych)}
[{including new values | excludign new values}];
```

Usuwanie migawki

```
DROP SNAPSHOT [schemat.]migawka;
```

Usuwanie dziennika migawki

```
DROP SNAPSHOT LOG ON [schemat.]tabela;
```


Informacje o migawkach

- **USER_SNAPSHOTS, ALL_SNAPSHOTS, DBA_SNAPSHOTS**

```
select name, table_name, master_owner,
       master, master_link, refresh_method,
       type, master_rollback_seg
from dba_snapshots;
```

NAME	TABLE_NAME	MASTER OWNER	MASTER	MASTER LINK	REFRESH METHOD	TYPE	MASTER RBS
SN_EMP	SNAP$_SN_EMP	SCOTT	EMP	@LAB.WORLD	ROWID	FAST	
SN_EMP1	SNAP$_SN_EMP1	SCOTT	EMP	@LAB.WORLD	PRIMARY KEY	COMPLETE	RB04

Informacje o dziennikach migawek

- **USER_SNAPSHOT_LOGS, ALL_SNAPSHOT_LOGS, DBA_SNAPSHOT_LOGS**

```
select log_owner, master, log_table, rowids, primary_key,
       filter_columns, current_snapshots, snapshot_id
from user_snapshot_logs;
```

```
LOG    MASTER     LOG_TABLE    ROWIDS PRIMARY FILTER CURRENT   SNAPS.
OWNER                                 KEYS    COLS.  SNAPS.    ID
------ ---------- ------------ ------ ------- ------ --------- --------
SCOTT  EMP        MLOG$_EMP    YES    YES     NO     25-JAN-00       57
SCOTT  EMP        MLOG$_EMP    YES    YES     NO     25-JAN-00       58
```


Informacje o odswiezaniu migawek

- **USER_SNAPSHOT_REFRESH_TMIES, ALL_SNAPSHOT_REFRESH_TIMES, DBA_SNAPSHOT_REFRESH_TIMES**

```
select owner, name, master_owner, master,
     to_char(last_refresh, 'dd.mm.yyyy:hh24:mi:ss') last_refresh
from user_snapshot_refresh_times;
```

```
OWNER NAME      MASTER_OWNER MASTER LAST_REFRESH
----- --------- ------------ ------ -------------------
DEMO  MV_SKLEPY USER1        SKLEPY 12.02.2002:18:05:00
```

Informacje o zarejestrowanych migawkach w bazie master

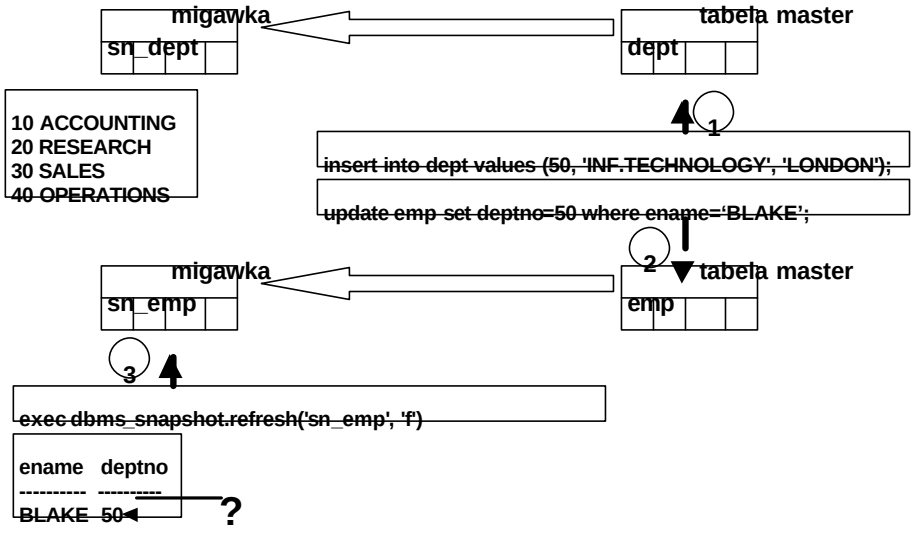
- **DBA_REGISTERED_SNAPSHOTS**

```
select owner, name, snapshot_site, can_use_log, updatable,
       refresh_method,snapshot_id
from user_registered_snapshots
where name='MV_SPRZEDAZ';
```

```
OWNER NAME        SNAPSHOT_SITE CAN UPD REFRESH_MET SNAPSHOT_ID
----- ----------- ------------- --- --- ----------- -----------
DEMO  MV_SPRZEDAZ DMINE.II.PP   YES NO  PRIMARY KEY          45
```

```
select sl.master "Master table", sl.log_table, rs.name as "Snp.name"
from dba_snapshot_logs sl, dba_registered_snapshots rs
where sl.snapshot_id=rs.snapshot_id;
```

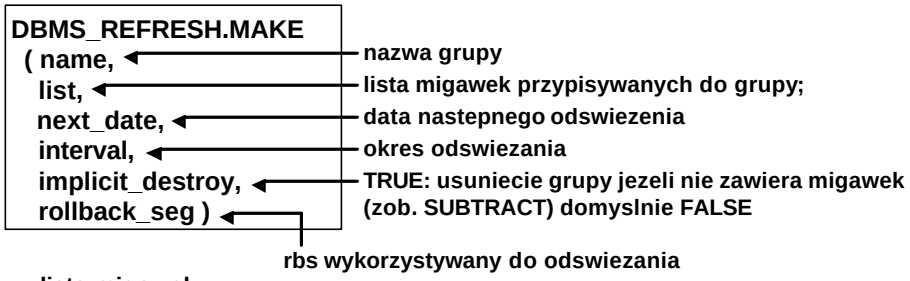

Wiele niezaleznych migawek - problem



Grupy odswiezania (ang. refresh groups)

- odswiezane jednoczesnie
- spójnosc danych migawek

Tworzenie grupy odswiezania



- lista migawek
  - migawki musza byc w tej samej bd
  - moga byc w róznych schematach
  - max. 100 migawek w grupie

Tworzenie grupy odswiezania(2)

```
exec DBMS_REFRESH.MAKE
  (name => 'orc1.rg_dept_emp', -
   list => 'orc1.sn_dept, orc1.sn_emp', -
   next_date => sysdate+(1/48), -
   interval => 'next_day(trunc(sysdate), ''FRIDAY'')+10/24', -
   implicit_destroy => TRUE, -
   rollback_seg => 'rb1')
```

Dodanie migawki do grupy

```
exec DBMS_REFRESH.ADD ('orc1.rg_dept_emp', 'orc1.sn_emp1')
```

Usuniecie migawki z grupy

```
exec DBMS_REFRESH.SUBTRACT('orc1.rg_dept_emp', 'orc1.sn_emp1')
```

Zmiana parametrów grupy

```
DBMS_REFRESH.CHANGE
  ( name,
    next_date,
    interval,
    implicit_destroy,
    rollback_seg )
```

```
exec DBMS_REFRESH.CHANGE
( name => 'orc1.rg_dept_emp', -
  next_date => sysdate+(1/(48*60)), -
  interval => 'next_day(trunc(sysdate), ''SATURDAY'')+8/24', -
  implicit_destroy => FALSE, -
  rollback_seg => 'rb0')
```

Manualne odswiezanie grupy

```
exec DBMS_REFRESH.REFRESH('orc1.rg_dept_emp')
```

Usuniecie grupy odswiezania

- usuwa grupe z migawkami lub pusta

```
exec DBMS_REFRESH.DESTROY('orc1.rg_dept_emp')
```


Informacje na temat utworzonych grup

- USER_REFRESH, ALL_REFRESH, DBA_REFRESH

```
select rowner, rname, refgroup, implicit_destroy, rollback_seg,
       next_date, interval, broken
from user_refresh;
```

```
                          Impl.  Rollb.
ROWNER RNAME       REFGROUP  destr. segm.  NEXT_DATE INTERVAL                Broken
------ ----------- --------- ------ ------ --------- ----------------------- ------
ORC1   RG_DEPT_EMP       96  N      RB0    26-JAN-00 next_day(trunc(sysdate),  N
                                                     'SATURDAY')+8/24
```

- jezeli automatyczne odswiezanie stalo sie niemozliwe:
  - proces odswiezajacy wykonuje 16 prób odswiezenia w pewnych odstepach czasu
    - jezeli 16-ta próba niepomyslna ustawiana wartosc BROKEN=Y
    - po usunieciu problemu odswiezenie manualne (BROKEN=N) -> przywrócenie odswiezania automatycznego

Informacje na temat migawek w grupie

- **USER_REFRESH_CHILDREN, ALL_REFRESH_CHILDREN, DBA_REFRESH_CHILDREN**

```
select owner, name, type, rowner, rname, refgroup
from user_refresh_children;
```

OWNER	NAME	TYPE	ROWNER	RNAME	REFGROUP
ORC1	SN_DEPT	SNAPSHOT	ORC1	RG_DEPT_EMP	96
ORC1	SN_EMP	SNAPSHOT	ORC1	RG_DEPT_EMP	96

Perspektywa zmaterializowana

- **wykorzystanie**
  - **optymalizacja zapytan**
  - **przetwarzanie OLAP**
- **przepisywanie zapytan (ang. query rewriting)**
- **przepisywanie zapytan bedzie mozliwe jesli**
  - **instancja Oracle zostanie wlaczona w trybie przepisywania zapytan**
  - **zostanie wykorzystany optymalizator kosztowy**
  - **perspektywa zmaterializowana zostanie utworzona z opcja przepisywania zapytan**
  - **zostana zebrane statystyki dla perspektywy**
  - **zapytanie definiujace perspektywe nie bedzie zawieralo pewnych konstrukcji**
  - **uzytkownik bedzie posiadal odpowiednie uprawnienie systemowe**

Przepisywanie zapytan

- **Optymalizator kosztowy**

```
alter session set optimizer_mode=CHOOSE;
```

- **Klauzula query rewrite**

```
create materialized view nazwa_perspektywy
[refresh sposób_odswiezania]
        [start with data_pierwszego_odswiezenia]
        [next czestotliwosc_odswiezania]
        [with typ_migawki]
[{enable | disable} query rewrite]
as zapytanie;
```

```
alter materialized view nazwa_perspektywy
                        {enable | disable} query rewrite;
```


Statystyki dla perspektywy zmaterializowanej

```
analyze table nazwa_perspektywy_zmaterializowanej
                        compute statistics;
```

Zapytanie definiujace perspektywe zmaterializowana

- **nie moze zawierac okreslonych konstrukcji, m.in.:**
  - **odwolan do zmiennych systemowych, np. *sysdate*, *user*, pseudokolumny *rownum*, sekwencerów, funkcji, atrybutów typu *raw*, *long raw*, *ref* (referencja do obiektu)**
  - **operatorów zbiorowych (*union*, *union all*, *intersect*, *minus*)**

Wykorzystanie perspektywy zmaterializowanej

```
create materialized view suma_sprzedazy
build immediate
refresh complete
enable query rewrite
as
select nazwa, sum(l_sztuk*cena_jedn) suma
    from sprzedaz sp, sklepy sk
    where sp.sklep_id=sk.sklep_id
    group by nazwa;
```

```
select nazwa, sum(l_sztuk*cena_jedn) suma
from sprzedaz sp, sklepy sk
where sp.sklep_id=sk.sklep_id
having sum(l_sztuk*cena_jedn) > 30000
group by nazwa;
```

```
Execution Plan
----------------------------------------------
   0      SELECT STATEMENT Optimizer=CHOOSE
   1    0   TABLE ACCESS (FULL) OF 'SUMA_SPRZEDAZY'
```


Wykorzystanie perspektywy zmaterializowanej (2)

```
select nazwa, sum(l_sztuk*cena_jedn) suma
from sprzedaz sp, sklepy sk
where sp.sklep_id=sk.sklep_id
and sk.miasto='Poznan'
group by nazwa;
```

```
Execution Plan
----------------------------------------------------------
   0      SELECT STATEMENT Optimizer=CHOOSE
   1    0   SORT (GROUP BY)
   2    1     NESTED LOOPS
   3    2       TABLE ACCESS (FULL) OF 'SKLEPY'
   4    2       TABLE ACCESS (FULL) OF 'SPRZEDAZ'
```