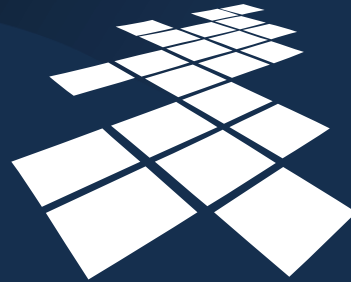


Rozproszone bazy danych – 1

Replikacja danych

Laboratorium przygotował:
Robert Wrembel



UCZELNIA
ONLINE



Plan laboratorium

- Dostęp do zdalnej bazy danych - łącznik bazy danych
- Replikowanie danych - migawka
- Parametry migawki
- Rodzaje migawek
- Przyrostowe odświeżanie migawki
- Grupa odświeżania

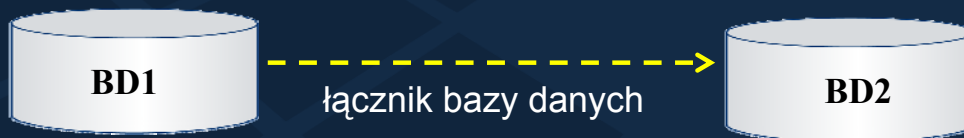
W ramach pierwszego laboratorium zostaną omówione następujące zagadnienia:

- sposób dostępu do zdalnej bazy danych za pomocą tzw. łącznika bazy danych,
- replikowanie danych w systemie rozproszonej bazy danych za pomocą tzw. migawki,
- parametry migawki definiowane przez użytkownika, mające wpływ na jej własności,
- problematyka przyrostowego odświeżania migawki,
- grupa odświeżania jako mechanizm jednoczesnego odświeżania wielu migawek.



Dostęp do zdalnej bazy danych

- Realizowany za pomocą tzw. łącznika bazy danych (ang. database link)
 - łącznik prywatny
 - łącznik publiczny



Łącznik bazy danych, zwany również łączem bazy danych (ang. database link) jest obiektem bazy danych, który umożliwia dostęp z bazy lokalnej do zdalnej. Za jego pomocą można więc wykonywać polecenia *select*, *insert*, *update*, *delete* i *lock table* na tabelach lub perspektywach znajdujących się w zdalnej bazie danych. Można również wywoływać znajdujące się tam procedury i funkcje. Wyróżnia się dwa rodzaje łączników, tj. prywatne i publiczne.

Łącznik prywatny (ang. private database link) jest własnością użytkownika, który go utworzył. Inni użytkownicy bazy danych nie mogą korzystać z łączników prywatnych innych użytkowników. Natomiast **łącznik publiczny** (ang. public database link) jest dostępny dla wszystkich użytkowników bazy danych.



Definiowanie łącznika

- Ogólna składnia polecenia

```
create database link nazwa
connect to użytkownik_zdalny identified by hasło
using 'nazwa_usługi';
```

- Przykład

```
create database link lab92
connect to scott identified by tiger
using 'LAB92.II.PP';
```

nazwa: jest nazwą łącznika;

użytkownik_zdalny: jest nazwą użytkownika istniejącego w zdalnej bazie danych;

hasło: jest hasłem użytkownika zdalnego;

nazwa_usługi: jest nazwą usługi zdefiniowaną w pliku *tnsnames.ora*

Przykładowe polecenie ze slajdu tworzy łącznik o nazwie *lab92* wskazujący na schemat użytkownika *scott* z hasłem *tiger*, znajdujący się w bazie danych określonej usługą nazwie *LAB92.II.PP*.

W momencie tworzenia łącznika system nie sprawdza czy istnieje usługa o podanej nazwie, ani czy wyspecyfikowana nazwa i hasło użytkownika są poprawne. Weryfikacja odbywa się dopiero w momencie odwołania się do zdalnej bazy za pomocą łącznika.



Przykłady wykorzystania łącznika

```
select * from rachunki@lab92;
```

```
delete from rachunki@lab92;
```

```
create table rachunki_kopia  
as  
select * from rachunki@lab92;
```

Za pomocą łącznika można następnie sięgnąć do wskazywanej przez niego bazy danych w poleceniach INSERT, UPDATE, DELETE, SELECT, podając jego nazwę po nazwie zdalnej tabeli i znaku @. Przykładowo, pierwsze polecenie ze slajdu wybiera wszystkie rekordy ze zdalnej tabeli *rachunki*, drugie – usuwa zawartość zdalnej tabeli *rachunki*, a trzecie – tworzy tabelę *rachunki_kopia* jako kopię zdalnej tabeli *rachunki*.



- Perspektywa USER_DB_LINKS

```
SQL> select * from user_db_links;
```

DB_LINK	USERNAME	PASSWORD	HOST	CREATED
DBLINK_LAB92	DEMO	DEMO	LAB92.II.PP	03/02/21

Informacje o prywatnych łącznikach użytkownika dostępne są za pomocą tzw. perspektywy słownikowej *USER_DB_LINKS*. Przykładowe zapytanie do tej perspektywy i jego wynik przedstawiono na slajdzie.

Znaczenie atrybutów perspektywy *USER_DB_LINKS* jest następujące.

DB_LINK jest nazwą łącznika;

USERNAME jest nazwą użytkownika wykorzystywanego w definicji łącznika, czyli użytkownika zdalnej bazy danych

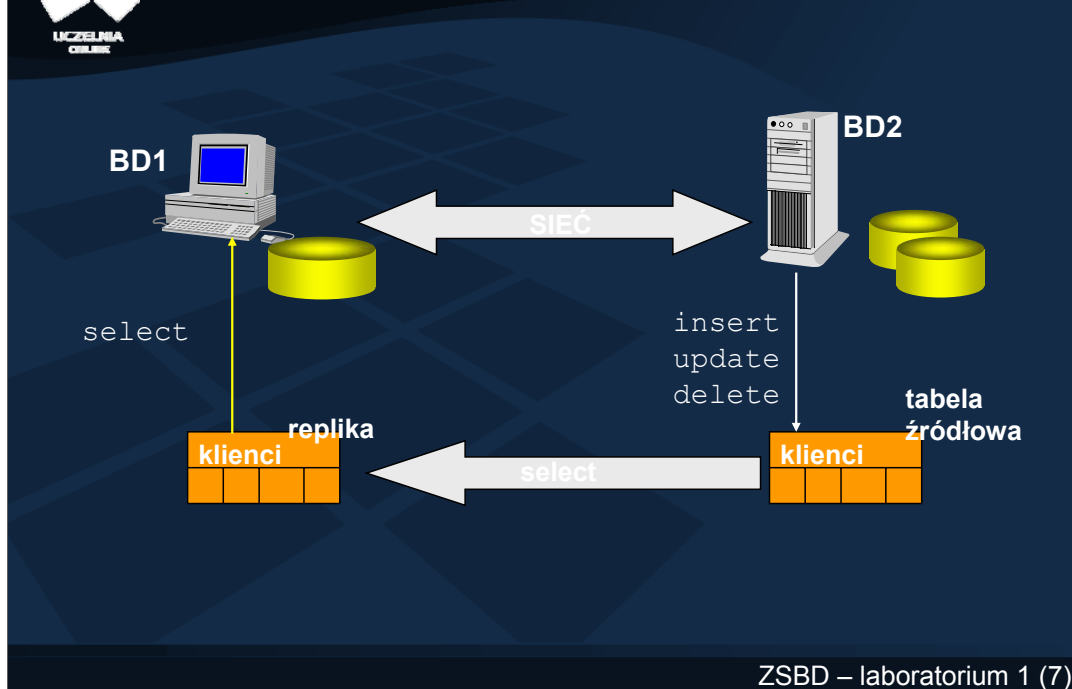
PASSWORD jest hasłem użytkownika zdalnej bazy danych

HOST jest nazwą sieciowej usługi bazy danych wskazanej w klauzuli USING polecenia tworzącego łącznik

CREATED zawiera date utworzenia łącznika



Standardowa replikacja danych



ZSBD – laboratorium 1 (7)

Standardowa replikacja polega na utworzeniu kopii zdalnej tabeli z bazy lokalnej. Tabela będąca kopią nazywa się **repliką**, a tabela na podstawie, której utworzono replikę nazywa się **tabelą źródłową** (nazywaną również tabelą master lub tabelą bazową).

Replika może zawierać wszystkie atrybuty i rekordy tabeli źródłowej lub ich podzbiór.

W architekturze standardowej replikacji replika jest tylko do odczytu. Replika posiada cechę automatycznego odświeżania, tzn. zmiany zawartości tabeli źródłowej propagują się do repliki automatycznie.

Podstawowa architektura standardowej replikacji danych została przedstawiona na slajdzie.



Migawka

- Replika implementowana jako tzw. migawka (ang. snapshot), zwana również perspektywą zmaterializowaną (ang. materialized view)
- W swojej definicji zawiera zapytanie udostępniające rekordy tabeli źródłowej
- Uprawnienia systemowe użytkownika umożliwiające tworzenie migawek (min. CREATE SNAPSHOT, CREATE MATERIALIZED VIEW)
- Fizycznie przechowywana jako tabela z indeksem
- Stowarzyszony z migawką proces odświeżania

W systemie Oracle replika jest implementowana jako tzw. migawka (ang. snapshot), zwana również perspektywą zmaterializowaną (ang. materialized view).

Migawka w swojej definicji zawiera zapytanie do tabeli źródłowej. Wynik tego zapytania jest dostępny w migawce.

Użytkownik tworzący migawkę musi posiadać odpowiednie uprawnienia systemowe (m.in. CREATE SNAPSHOT lub CREATE MATERIALIZED VIEW).

Migawka jest fizycznie przechowywana jako tabela z indeksem. Z migawką jest zwykle stowarzyszony proces systemowy odpowiedzialny za jej odświeżanie.



Definicja migawki

- 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
 - zapytania określającego zakres danych dostępnych w migawce

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
- zapytania określającego zakres danych dostępnych w migawce



Tworzenie migawki

```
create snapshot nazwa_migawki
build moment_wypełnienia_danymi
refresh sposób_odświeżania
start with data_rozpoczęcia_odświeżania
next okres_odświeżania
with identyfikacja_rekordów
as zapytanie;
```

Podstawowa składnia polecenia tworzącego migawkę została przedstawiona na slajdzie.

Migawkę tworzy się poleceniem SQL *create snapshot* lub *create materialized view*, oba o identycznej składni.

Klauzula *build* określa moment pierwszego wypełnienia migawki danymi.

Klauzula *refresh* określa sposób odświeżania.

Klauzula *start with* określa datę i czas pierwszego odświeżenia po utworzeniu migawki.

Klauzula *next* określa okres odświeżania mierzony np. w dniach, godzinach, lub minutach.

Klauzula *as* zawiera zapytanie do tabel źródłowych.



Klauzula BUILD

- Wskazuje moment wypełnienia migawki danymi, po jej utworzeniu
- IMMEDIATE – wypełnienie natychmiastowe
- DEFERRED – wypełnienie z opóźnieniem

Klauzula BUILD wskazuje moment wypełnienia migawki danymi, po jej utworzeniu poleceniem *create snapshot (create materialized view)*. Można w niej wyspecyfikować albo IMMEDIATE albo DEFERRED. W pierwszym przypadku migawka jest wypełniana danymi zaraz po jej utworzeniu. W drugim przypadku, migawka jest wypełniana danymi po upływie czasu (od jej utworzenia) określonego wyrażeniem w klauzuli START WITH.



Klauzula REFRESH

- Określa sposób odświeżania
- FAST – odświeżanie przyrostowe
- COMPLETE – odświeżanie pełna
- FORCE – odświeżanie wybierane automatycznie, takie jakie jest możliwe dla migawki, ale z preferencją odświeżania przyrostowego

Klauzula REFRESH określa sposób odświeżania migawki. Można w niej wyspecyfikować jedno z trzech słów kluczowych, tj. FAST, COMPLETE, lub FORCE.

FAST oznacza odświeżanie przyrostowe. W tym przypadku z tabeli źródłowej do repliki są przesyłane tylko zmiany dokonane od czasu ostatniego odświeżania. COMPLETE oznacza odświeżanie pełne. W tym przypadku z tabeli źródłowej są przesyłane wszystkie dane spełniające warunki zapytania definiującego replikę. FORCE umożliwia automatyczne wybranie sposobu odświeżania, z preferencją odświeżania przyrostowego (jeśli takie jest możliwe). Decyduje o tym system.



Klauzula START WITH

- Określa czas po jakim migawka zostanie wypełniona danymi, po jej utworzeniu
- Brana pod uwagę tylko jeśli wyspecyfikowano BUILD DEFERRED

Klauzula START WITH określa czas po jakim migawka zostanie wypełniona danymi, po jej utworzeniu. Klauzula ta obowiązuje (jest uwzględniana) tylko jeśli wyspecyfikowano klauzulę BUILD DEFERRED.



Klauzula NEXT

- Określa czas po jakim replika jest ponownie odświeżana
- Czas ten jest mierzony od momentu zakończenia poprzedniego odświeżenia
- Jeśli nie wyspecyfikowano tej klauzuli, wówczas migawka nie jest odświeżana automatycznie
 - można ją odświeżyć "ręcznie" za pomocą odpowiedniej procedury systemowej

Klauzula NEXT określa czas po jakim replika jest ponownie odświeżana. Czas ten jest mierzony od momentu zakończenia poprzedniego odświeżenia.

Jeśli nie wyspecyfikowano tej klauzuli, wówczas migawka nie jest odświeżana automatycznie.

Migawkę można zawsze odświeżyć "ręcznie" za pomocą odpowiedniej procedury systemowej omówionej na następnym slajdzie.



Odświeżanie "ręczne" (1)

Procedury systemowe (można stosować zamiennie)
 DBMS_SNAPSHOT.REFRESH lub
 DBMS_MVIEW.REFRESH
 Ogólna składnia wywołania

```
DBMS_SNAPSHOT.REFRESH
('sn1, sn2, ..., snn',
'metoda')
```

- sn_1, sn_2, \dots, sn_n : migawki
- *metoda*: metoda odświeżania
 - *f* lub *F*: FAST
 - *c* lub *C*: COMPLETE
 - *?*: domyślny

Migawkę można odświeżyć "ręcznie" zawsze, nawet gdy jest odświeżana automatycznie. Jeśli jednak nie wyspecyfikowano klauzuli NEXT, wówczas jedynym sposobem odświeżenia jest odświeżenie "ręczne". Służy do tego celu procedura REFRESH znajdująca się w pakiecie systemowym DBMS_SNAPSHOT lub DBMS_MVIEW. W praktyce pakiet drugi jest synonimem pierwszego i można je stosować zamiennie.

Ogólną składnię wywołania procedury REFRESH przedstawiono na slajdzie. Przyjmuje ona dwa argumenty wywołania. Pierwszy jest zbiorem nazw migawek do odświeżenia. Poszczególne nazwy są oddzielone przecinkami. Zbiór ten jest przekazywany do procedury jako ciąg znaków. Drugim argumentem jest metoda odświeżenia. Drugi argument może przyjąć wartość:

- *f* lub *F* oznaczającą odświeżanie przyrostowe,
- *c* lub *C* oznaczającą odświeżanie pełne,
- *?* oznaczającą odświeżanie takie jak zdefiniowano tworząc migawkę.



Odświeżanie "ręczne" (2)

• Przykład 1

DBMS_SNAPSHOT.REFRESH ('s_dept, s_emp, s_emp1', 'C')

• Przykład 2

DBMS_SNAPSHOT.REFRESH ('s_dept, s_emp, s_emp1', 'CF')

W przykładzie 1 są odświeżane trzy migawki o nazwach *s_dept*, *s_emp*, *s_emp1*, każda z nich jest odświeżana w trybie *C*, czyli pełnym.

W przykładzie 2 są odświeżane te same trzy migawki. Pierwsza z nich w trybie *C* (pełnym), druga w trybie *F* (przyrostowym), a trzecia w trybie domyślnym (określonym w momencie tworzenia migawki).



Klauzula WITH (1)

- Określa sposób identyfikowania rekordów w tabeli źródłowej i migawce
- Wykorzystywana tylko w przypadku odświeżania przyrostowego
- ROWID – identyfikowanie rekordów za pomocą fizycznych adresów rekordów

Klauzula ta określa sposób identyfikowania rekordów w tabeli źródłowej i w migawce. Możliwe są tu dwa sposoby identyfikacji. Pierwszy z nich wykorzystuje fizyczne adresy rekordów na dysku — ROWID, a migawkę wykorzystującą te adresy będziemy dalej nazywać migawką typu ROWID.

ROWID zawiera w sobie adres pliku, numer bloku w ramach tego pliku i adres rekordu w bloku



Klauzula WITH (2)

- PRIMARY KEY – identyfikowanie za pomocą klucza podstawowego tabeli
 - klauzula SELECT musi zawierać wszystkie atrybuty wchodzące w skład klucza podstawowego tabeli master
 - tabela master musi posiadać włączone ograniczenie PRIMARY KEY

W przypadku klauzuli WITH PRIMARY KEY do identyfikacji rekordów źródłowych i migawki są wykorzystywane wartości klucza podstawowego tabeli źródłowej. Taką migawkę będziemy dalej nazywać typu PRIMARY KEY. Począwszy od wersji *Oracle* (rel. 8.0) udostępnione zostały migawki typu *primary key*, jako domyślny typ. Stosowanie tego typu migawek jest zalecane przez producenta oprogramowania.

Należy zwrócić uwagę, że migawka typu PRIMARY KEY musi udostępniać wszystkie atrybuty wchodzące w skład klucza tabeli źródłowej, a tabela źródłowa musi posiadać aktywne ograniczenie PRIMARY KEY. W przeciwnym przypadku tworzenie migawki zakończy się błędem.



Klauzula AS

- Zawiera zapytanie udostępniające dane z tabel źródłowych
- Z punktu widzenia złożoności tego zapytania migawki dzieli się na:
 - proste
 - złożone

Klauzula AS umożliwia zdefiniowanie zapytania sięgającego do tabel źródłowych. Wynik tego zapytania jest materializowany w migawce.

Z punktu widzenia złożoności tego zapytania, migawki dzieli się na proste i złożone.



Migawka prosta

- Bazująca na jednej tabeli źródłowej
- Brak klauzul: GROUP BY, CONNECT BY, DISTINCT
- Brak wyrażeń, funkcji, połączeń, operatorów zbiorowych
- Daje się odświeżać przyrostowo

Migawka prosta posiada następujące cechy:

- jej zapytanie odwołuje się do jednej tabeli źródłowej,
- jej zapytanie nie wykorzystuje klauzul GROUP BY, CONNECT BY, DISTINCT,
- jej zapytanie nie wykorzystuje wyrażeń, funkcji SQL, połączeń, operatorów zbiorowych.

Migawkę prostą daje się zawsze odświeżać przyrostowo.



Migawka złożona

- Nie spełnia warunków migawki prostej
- Przy bardzo złożonych zapytaniach nie daje się odświeżać przyrostowo

Migawka złożona to taka, która nie spełnia przynajmniej jednego warunku dot. złożoności zapytania migawki prostej. Migawki złożone nie zawsze dają się odświeżać przyrostowo. Zależy to od stopnia złożoności zapytania. W wielu przypadkach jedynym sposobem odświeżenia migawki złożonej jest odświeżanie pełne.



Tworzenie migawki - przykład (1)

```
create materialized view mv_sprzedaz1
build immediate
refresh complete
next sysdate+(1/(24*60*6))
as
select * from sprzedaz@lab92
where data like '%2003';
```

Przykładowe polecenie ze slajdu definiuje migawkę o nazwie *mv_sprzedaz1* o następującej charakterystyce:

- wypełnienie danymi w momencie tworzenia (klauzula *build immediate*);
- odświeżanie pełne (klauzula *refresh complete*);
- okres odświeżania - 10 sekund (*next sysdate+(1/(24*60*6))*);
- identyfikacja rekordów za pomocą klucza podstawowego (domyślna, niewyspecyfikowana klauzula *with primary key*);
- udostępnia informacje o sprzedaży produktów w roku 2003 z tabeli wskazywanej łącznikiem o nazwie *lab92*.



Tworzenie migawki - przykład (2)

```
create materialized view mv_sprzedaz2
build deferred
refresh force
start with sysdate+(1/(24*60))
next sysdate+(1/(24*30))
with rowid
as
select * from sprzedaz@lab92
where data like '%2004';
```

Przykładowe polecenie ze slajdu definiuje migawkę o nazwie *mv_sprzedaz2* o następującej charakterystyce:

- wypełnienie danymi z opóźnieniem (klauzula *build deferred*);
- odświeżanie wybierane automatycznie przez system (klauzula *refresh force*);
- moment rozpoczęcia odświeżania automatycznego - 1 minuta po utworzeniu migawki (klauzula *start with sysdate+(1/(24*60))*);
- okres odświeżania - 2 minuty (*next sysdate+(1/(24*30))*);
- identyfikacja rekordów za pomocą fizycznych adresów (klauzula *with rowid*)
- udostępnia informacje o sprzedaży produktów w roku 2004 z tabeli wskazywanej łącznikiem o nazwie *lab92*.



Odświeżanie przyrostowe (1)

- Warunki
 - migawka jest typu prostego i
 - istnieje dziennik migawki (por. slajdy kolejne) na wszystkich jej tabelach źródłowych

Przyrostowe odświeżanie migawki jest możliwe jeśli:

- migawka jest typu prostego i
- istnieje dziennik migawki na wszystkich jej tabelach źródłowych



Odświeżanie przyrostowe (2)

- Warunki
 - migawka jest typu złożonego umożliwiającego odświeżanie przyrostowe
 - istnieje dziennik migawki na wszystkich jej tabelach źródłowych

Przyrostowe odświeżanie migawki jest możliwe jeśli:

- migawka jest typu złożonego ale umożliwiającego odświeżanie przyrostowe i
- istnieje dziennik migawki na wszystkich jej tabelach źródłowych



Odświeżanie przyrostowe migawki złożonej (1)

- Jeśli migawka wylicza agregaty, m.in. *count*, *sum*, *avg*, *variance*, *stdev*, wówczas:
 - dziennik tworzony z klauzulą *including new values*
 - dziennik zawiera wszystkie atrybuty wymienione po *select*, również będące argumentami wywołania funkcji grupowych
 - funkcja *count* zawsze umieszczana w zapytaniu, gdy są wyliczane *sum*, *avg*, *variance*, *stdev*

Jeżeli migawka wylicza agregaty, m.in. *count*, *sum*, *avg*, *variance*, *stdev*, wówczas:

1. dziennik migawki musi zostać utworzony z klauzulą *including new values*,
2. dziennik migawki musi przechowywać wszystkie atrybuty wymienione w klauzuli *select*, również te, które są argumentami wywołania funkcji grupowych,
3. zapytanie musi zawierać dodatkowo funkcję *count* jeśli są wyliczane *sum*, *avg*, *variance*, *stdev*.



Odświeżanie przyrostowe migawki złożonej - przykład

```
create snapshot mv_suma_sprzedazy
build immediate
refresh fast
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;
```

ZSBD – laboratorium 1 (27)

W przykładzie ze slajdu zapytanie definiujące migawkę zostało wzbogacone o trzy funkcje *count*, zapewniające możliwość przyrostowego odświeżania tej migawki.



Dziennik migawki

- Obiekt bazy danych rejestrujący operacje modyfikowania zawartości (wstawiania, modyfikowania, usuwania rekordów) tabeli źródłowej
- Tworzony dla pojedynczej tabeli
- Wykorzystywany do odświeżania przyrostowego
- Implementacyjnie dziennik to tabela z mechanizmami zarządzania jej zawartością
 - tabela o nazwie `MLOG$_nazwa_tabeli_bazowej`

Dziennik migawki (ang. snapshot log), zwany również dziennikiem perspektywy zmaterializowanej (ang. materialized view log) jest obiektem bazy danych, którego zadaniem jest rejestrowanie operacji modyfikowania zawartości tabeli, dla której go utworzono. Dziennik jest tworzony dla konkretnej (jednej) tabeli. Jest on wykorzystywany do przyrostowego odświeżania migawki korzystającej z tabeli z dziennikiem.

Implementacyjnie dziennik jest tabelą o nazwie `MLOG$_nazwa_tabeli_bazowej`. Z dziennikiem są związane systemowe mechanizmy zarządzania jego zawartością, m.in. wstawianie rekordów, usuwanie rekordów wykorzystanych przez migawki.



Tworzenie dziennika

```
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 tworzy się poleceniem CREATE SNAPSHOT LOG lub CREATE MATERIALIZED VIEW LOG, którego ogólną składnię przedstawiono na slajdzie.

tabela_bazowa oznacza tabelę, dla której dziennik jest tworzony. Pozostałe klauzule i parametry tego polecenia są opcjonalne.

WITH PRIMARY KEY oznacza dziennik dla migawki typu PRIMARY KEY; jest to klauzula domyślna

WITH ROWID oznacza dziennik dla migawki typu ROWID

WITH PRIMARY KEY, ROWID oznacza dziennik dla migawek zarówno typu PRIMARY KEY jak i ROWID

lista_kolumn_filtrujacych jest to lista zawierająca atrybuty wykorzystywane w klauzuli WHERE zapytania definiującego migawkę i atrybuty występujące jako argumenty wywołania funkcji grupowych w zapytaniu

INCLUDING NEW VALUES zapewnia przyrostowe odświeżanie migawek wliczających agregaty



Tworzenie dziennika - przykład

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

Przykład ze slajdu zawiera polecenie tworzące dziennik migawki dla tabeli *sprzedaz*. Dziennik ten jest tworzony zarówno dla migawki typu PRIMARY KEY, jak i ROWID. Dodatkowo, w dzienniku będą rejestrowane zmiany wartości atrybutów *l_sztuk* i *cena_jedn*. Zawartość tak utworzonego dziennika będzie wykorzystywana do przyrostowego odświeżania migawki ze slajdu 25.



Modyfikowanie migawki

```
alter snapshot nazwa_migawki
refresh sposób_odświeżania
start with data_rozpoczęcia_odświeżania
next okres_odświeżania
with PRIMARY KEY;
```

```
alter snapshot sn_emp1
refresh complete
start with sysdate
next
sysdate+1/(24*60*10)
with primary key;
```

Istniejącą migawkę można zmodyfikować za pomocą polecenia ALTER SNAPSHOT lub ALTER MATERIALIZED VIEW, którego podstawową składnię przedstawiono na slajdzie. Za pomocą tego polecenia można zmienić sposób odświeżania, okres odświeżania, zamienić migawkę typu ROWID na PRIMARY KEY (ale nie odwrotnie). Klauzula *start with* umożliwia wskazanie czasu po którym migawka zostanie odświeżona po zmianie jej definicji.

W przykładowym poleceniu na slajdzie modyfikowana jest migawka o nazwie *sn_emp1*.



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} ];
```

Istniejący dziennik migawki można zmodyfikować za pomocą polecenia ALTER SNAPSHOT LOG lub ALTER MATERIALIZED VIEW LOG, którego podstawową składnię przedstawiono na slajdzie.



Usuwanie migawki i jej dziennika

- Usuwanie migawki

```
drop snapshot nazwa_migawki;
```

- Usuwanie dziennika migawki

```
drop snapshot log on  
nazwa_tabeli_bazowej;
```

Migawkę usuwa się poleceniem DROP SNAPSHOT lub DROP MATERIALIZED VIEW LOG wskazując migawkę za pomocą jej nazwy. Dziennik migawki usuwa się poleceniem DROP SNAPSHOT LOG ON lub DROP MATERIALIZED VIEW LOG ON wskazując tabelę bazową.



Informacje o migawkach

- USER_SNAPSHOTS
- USER_MVIEWS

```
select name, table_name, master,
       master_link, refresh_method, type
from user_snapshots;
```

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

Informacje o utworzonych przez użytkownika migawkach są dostępne za pomocą perspektywy słownikowej o nazwie USER_SNAPSHOTS lub USER_MVIEWS. Przykładowe zapytanie do perspektywy USER_SNAPSHOTS i jego wynik przedstawiono na slajdzie.

Najczęściej wykorzystywane atrybuty perspektywy USER_SNAPSHOTS omówiono na następnym slajdzie.



Informacje o migawkach cd.

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

NAME - nazwa migawki;

TABLE_NAME - nazwa tabeli implementującej migawkę;

MASTER - nazwa tabeli źródłowej migawki; w przypadku gdy migawka posiada kilka tabel źródłowych, jako wartość *MASTER* pojawia się nazwa tej tabeli, którą wymieniono jako ostatnią w klauzuli *from* zapytania definiującego migawkę;

MASTER_LINK - nazwa łącznika bazy danych, wykorzystywanego do zrealizowania dostępu do ostatniej tabeli wymienionej w klauzuli *from* zapytania definiującego migawkę;

UPDATABLE - przyjmuje wartość *YES* dla migawki modyfikowalnej, w przeciwnym przypadku przyjmuje *NO*; migawki modyfikowalne są wykorzystywane w tzw. zaawansowanej opcji replikacji;

REFRESH_METHOD - określa sposób identyfikacji rekordów migawki; przyjmuje wartość *PRIMARY KEY* dla migawki odświeżanej przyrostowo, zdefiniowanej z klauzulą *with primary key*; przyjmuje wartość *ROWID* dla migawki odświeżanej przyrostowo, zdefiniowanej z klauzulą *with rowid*;



Informacje o migawkach cd.

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

TYPE - określa sposób odświeżania i może przyjąć jedną z czterech wartości: *FAST*, *FORCE*, *COMPLETE*, *NEVER*;

NEXT - wartość wyspecyfikowana w klauzuli next;

START_WITH - wartość wyspecyfikowana w klauzuli start with;

REFRESH_GROUP - numer grupy odświeżania, do której należy migawka;

QUERY - tekst zapytania (klauzula *select*) definiującego migawkę;

REFRESH_MODE - przyjmuje jedną z trzech wartości: *PERIODIC* — jeśli wyspecyfikowano zarówno klauzulę start with, jak i next; *COMMIT* — jeśli wyspecyfikowano klauzulę on commit; *NEVER* — jeśli wyspecyfikowano klauzulę never refresh, *DEMAND* — w pozostałych przypadkach.



Informacje o dziennikach migawek

USER_SNAPSHOT_LOGS
USER_MVIEW_LOGS

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

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

ZSBD – laboratorium 1 (37)

Informacje o utworzonych przez użytkownika dziennikach migawek są dostępne za pomocą perspektywy słownikowej o nazwie USER_SNAPSHOT_LOGS lub USER_MVIEW_LOGS. Przykładowe zapytanie do perspektywy USER_SNAPSHOT_LOGS i jego wynik przedstawiono na slajdzie.

Najczęściej wykorzystywane atrybuty perspektywy USER_SNAPSHOT_LOGS omówiono na następnym slajdzie.



Informacje o dziennikach migawek (cd)

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

LOG_OWNER - nazwa użytkownika będącego właścicielem dziennika migawki;

MASTER - nazwa tabeli, dla której utworzono dziennik;

LOG_TABLE - nazwa tabeli implementującej dziennik;

ROWIDS - przyjmuje wartość *YES* dla dziennika utworzonego z klauzulą *with ROWID*, w przeciwnym przypadku przyjmuje wartość *NO*;

PRIMARY_KEY - przyjmuje wartość *YES* dla dziennika utworzonego z klauzulą *with primary key*, w przeciwnym przypadku przyjmuje wartość *NO*;

FILTER_COLUMNS - przyjmuje wartość *YES* dla dziennika zawierającego kolumny filtrujące;

INCLUDE_NEW_VALUES - przyjmuje wartość *YES* dla dziennika utworzonego z klauzulą *include new values*, w przeciwnym przypadku przyjmuje wartość *NO*;

CURRENT_SNAPSHOTS - data ostatniego odświeżenia migawki z wykorzystaniem zawartości dziennika;

SNAPSHOT_ID - identyfikator migawki, która korzysta z dziennika.



Informacje o odświeżaniu migawek

```
USER_SNAPSHOT_REFRESH_TIMES  
USER_MVIEW_REFRESH_TIMES
```

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

NAME	LAST_REFRESH
MV_SKLEPY	12.02.2002:18:05:00

Informacje o odświeżaniu migawek są dostępne za pomocą perspektywy słownikowej o nazwie `USER_SNAPSHOT_REFRESH_TIMES` lub `USER_MVIEW_REFRESH_TIMES`. Przykładowe zapytanie do perspektywy `USER_SNAPSHOT_REFRESH_TIMES` i jego wynik przedstawiono na slajdzie. Atrybut *last_refresh* udostępnia datę i czas ostatniego odświeżenia migawki, której nazwa jest udostępniana atrybutem *name*.



Grupa odświeżania

- Obiekt zawierający jedną lub wiele migawek
- Wszystkie migawki w grupie są odświeżane w tym samym momencie
- Niejawnie każda migawka jest umieszczana w swojej grupie, tworzonej automatycznie przez system
- Dana migawka może należeć wyłącznie do jednej grupy
- Migawka, która nie należy do grupy odświeżania nie jest odświeżana automatycznie

Grupa odświeżania (ang. refresh group) jest obiektem grupującym wiele migawek, z których każda jest odświeżana w tym samym momencie. Grupa odświeżania zapewnia, że wszystkie umieszczone w niej migawki są odświeżane w tym samym momencie.

Niejawnie grupa odświeżania jest tworzona przez system dla każdej migawki. Tak więc każda migawka jest umieszczona domyślnie w swojej grupie. Grupa odświeżania może się składać z wielu migawek, ale migawka może być umieszczona tylko w jednej grupie.

Automatycznie są odświeżane tylko migawki należące do grupy odświeżania



Tworzenie grupy odświeżania

DBMS_REFRESH.MAKE

<code>(name,</code>	←	nazwa grupy
<code>list,</code>	←	lista migawek przypisywanych do grupy;
<code>next_date,</code>	←	data odświeżenia po utworzeniu grupy
<code>interval,</code>	←	okres odświeżania
<code>implicit_destroy)</code>	←	TRUE: usunięcie grupy jeżeli nie zawiera migawek (zob. SUBTRACT) domyślnie FALSE

lista migawek

migawki muszą być w tej samej bazie danych
w jednej grupie może się znaleźć co najwyżej 100
migawek

Grupy odświeżania mogą być również tworzone jawnie. Służy do tego celu procedura MAKE pakietu systemowego o nazwie DBMS_REFRESH. Podstawową składnię wywołania procedury MAKE przedstawiono na slajdzie. Procedura MAKE posiada 13 argumentów wywołania, z których cztery pierwsze muszą posiadać jawnie wyspecyfikowane wartości. Pozostałe 9 argumentów może przyjąć wartości domyślne. Pierwszym argumentem wywołania (name typu *varchar2*) jest nazwa tworzonej grupy odświeżania. Drugim argumentem jest nazwa migawki, lub lista migawek (list typu *varchar2*), bądź też zmienna typu tablica PL/SQL zawierająca nazwy migawek (tab typu *table of varchar2*). Argument trzeci (next_date typu *date*) określa datę i czas pierwszego odświeżenia migawek w grupie. Natomiast argument czwarty (interval typu *varchar2*) określa wspólną dla wszystkich migawek częstotliwość odświeżania. Dodatkowo, wartością piątego argumentu (implicit_destroy typu *boolean*) można określić zachowanie się grupy w przypadku usunięcia z niej ostatniej migawki. Domyślnie argument ten przyjmuje wartość *TRUE*, co oznacza automatyczne usunięcie grupy w przypadku, gdy usunięto z niej ostatnią migawkę. Wartość *FALSE* tego argumentu spowoduje zachowanie grupy w systemie.



Tworzenie grupy odświeżania - przykład

```
exec dbms_refresh.make -  
(name=>'rg_firma', -  
 list=>'mv_sprzedaz', -  
 next_date=>sysdate + (1/(24*60*60)), -  
 interval=>'sysdate + (1/(24*60*10))', -  
 implicit_destroy=>FALSE)
```

Jako przykład rozważmy polecenie ze slajdu tworzące grupę odświeżania o nazwie *rg_firma* zawierającą jedną migawkę — *mv_sprzedaz*. Grupa zostanie odświeżona sekundę po jej utworzeniu i będzie automatycznie odświeżana co 6 sekund. Po usunięciu ostatniej migawki z grupy nie zostanie ona automatycznie usunięta z systemu.

Znaki "-" na końcu poszczególnych linii polecenia oznaczają podział linii.



Dodanie migawki do grupy

- Procedura DBMS_REFRESH.ADD

```
dbms_refresh.add ('nazwa_grupy', -  
                  'lista migawek')
```

- Przykład

```
exec dbms_refresh.add ('rg_firma', -  
                       'mv_sklepy, mv_produkty')
```

Dodanie nowej migawki do istniejącej grupy realizuje się za pomocą procedury DBMS_REFRESH.ADD. Jej wywołanie wymaga podania wartości przynajmniej dwóch argumentów — nazwy grupy odświeżania (name typu *varchar2*) i listy (list typu *varchar2*) bądź tablicy (tab typu *table of varchar2*) migawek. Przykładowe wywołanie tej procedury przedstawiono na slajdzie. Polecenie to dodaje do grupy *rg_firma* dwie migawki *mv_sklepy* i *mv_produkty*.



Usunięcie migawki z grupy

- Procedura DBMS_REFRESH.SUBTRACT

```
dbms_refresh.subtract ('nazwa_grupy', -  
                        'lista migawek')
```

- Przykład

```
exec dbms_refresh.subtract ('rg_firma', -  
                            'mv_sklepy, mv_produkty')
```

Migawkę usuwa się z grupy za pomocą procedury DBMS_REFRESH.SUBTRACT. Jej wywołanie wymaga podania wartości przynajmniej dwóch argumentów — nazwy grupy odświeżania i listy bądź tablicy migawek. Przykładowe wywołanie tej procedury przedstawiono na slajdzie. Polecenie to usuwa z grupy *rg_firma* dwie migawki *mv_sklepy* i *mv_produkty*.



"Ręczne" odświeżanie grupy

- Procedura DBMS_REFRESH.REFRESH
- Przykład

```
exec dbms_refresh.refresh ('rg_firma')
```

Grupę można odświeżać manualnie za pomocą procedury DBMS_REFRESH.REFRESH, wymagającej podania nazwy grupy, jako argumentu wywołania.



Usunięcie grupy

- Procedura DBMS_REFRESH.DESTROY
- Przykład

```
exec dbms_refresh.destroy ('rg_firma')
```

Usunięcie grupy odświeżania, bez fizycznego usuwania migawek, realizuje się za pomocą procedury DBMS_REFRESH.DESTROY. Argumentem jej wywołania jest nazwa grupy.

Możliwa jest zmiana parametrów grupy. Służy do tego celu procedura DBMS_REFRESH.CHANGE.



Informacje na temat utworzonych grup odświeżania (1)

Perspektywa słownikowa USER_REFRESH

```
select rname, refgroup, implicit_destroy,
       to_char(next_date, 'dd.mm.yyyy:hh24:mi:ss')
next_date,
       interval, broken
from user_refresh;
```

RNAME	REFGROUP	IMPLICIT DESTROY	NEXT_DATE	INTERVAL	BROKEN
MV_SKLEP	94	Y	12.02.2002:18:36:28	sysdate+(1/(24*60*30))	N

Informacje o istniejących w systemie grupach odświeżania są udostępniane za pomocą perspektywy słownikowej o nazwie USER_REFRESH. Przykładowe zapytanie do tej perspektywy i jego wynik przedstawiono na slajdzie.

RNAME - oznacza nazwę grupy odświeżania;

REFGROUP - jest identyfikatorem grupy;

IMPLICIT_DESTROY - przyjmuje wartość *Y* lub *N*; w pierwszym przypadku grupa zostanie automatycznie usunięta po usunięciu z niej ostatniej migawki;

NEXT_DATE - przechowuje datę i czas następnego odświeżenia grupy; wartość ta zmienia się po każdym kolejnym odświeżeniu grupy;



Informacje na temat utworzonych grup odświeżania (2)

Perspektywa słownikowa USER_REFRESH

RNAME	REFGROUP	IMPLICIT		NEXT_DATE	INTERVAL	BROKEN
		DESTROY				
MV_SKLEP	94	Y		12.02.2002:18:36:28	sysdate+(1/(24*60*30))	N

INTERVAL - jest wyrażeniem określającym częstotliwość odświeżania grupy, przekazaną argumentem wywołania *interval*;

BROKEN - określa, czy grupa jest odświeżana automatycznie jeśli wyspecyfikowano wartość *interval*; przyjmuje wartość *N* lub *Y*; w pierwszym przypadku grupa jest odświeżana automatycznie; w drugim przypadku grupa nie jest odświeżana. Dla danej grupy zdefiniowanej jako odświeżana automatycznie, system nada atrybutowi *BROKEN* wartość *Y* jeżeli nie uda się odświeżyć grupy predefiniowaną liczbę razy, np. z powodu czasowej niedostępności źródła danych. W takim przypadku grupę należy odświeżyć manualnie i jeśli taka próba się powiedzie, system zmieni wartość *BROKEN* na *N*, co przywróci odświeżanie automatyczne.



Informacje na temat migawek w grupie odświeżania

- Perspektywa słownikowa
USER_REFRESH_CHILDREN

```
select name, rname, refgroup  
from user_refresh_children;
```

NAME	RNAME	REFGROUP
MV_SPRZEDAZ	RG_FIRMA	100

Informacje na temat migawek umieszczonych w danej grupie odświeżania można uzyskać za pomocą perspektyw słownikowej o nazwie USER_REFRESH_CHILDREN.

Przykładowe zapytanie do tej perspektywy i jego wynik przedstawiono na slajdzie.

NAME - oznacza nazwę migawki znajdującej się w grupie;

RNAME - oznacza nazwę grupy odświeżania;

REFGROUP - jest numerem grupy odświeżania. Jeden rekord tej perspektywy opisuje pojedynczą migawkę. Dla migawek znajdujących się w tej samej grupie wartość *RNAME* i *REFGROUP* jest identyczna.



Środowisko do ćwiczeń

- Korzystamy z dwóch baz danych oznaczonych w ćwiczeniach jako BD1 i BD2
- W bazie DB1 korzystamy z użytkownika *scott* z hasłem *tiger*
- W bazie DB2 korzystamy z własnego użytkownika

W ćwiczeniu korzystamy z dwóch baz danych oznaczonych jako BD1 i BD2. W bazie DB1 korzystamy z użytkownika *scott* z hasłem *tiger*. W bazie DB2 korzystamy z własnego użytkownika.



Ćwiczenie 1 - łączniki bazy danych (1)

1. Dołączyć się do bazy danych BD2
2. Utworzyć łącznik bazy danych wskazujące na schemat użytkownika *scott* z hasłem *tiger* w bazie danych BD1
3. Odczytać zbiór tabel użytkownika *scott*
4. Utworzyć synonimy do tabel: *KLIENCI*, *R_ROR* i *R_TERMINOWE* w schemacie użytkownika *scott*
5. Korzystając z utworzonych synonimów odczytać zawartość tabel: *KLIENCI*, *R_ROR* i *R_TERMINOWE*

Celem ćwiczenia pierwszego jest zapoznanie się z łącznikami bazy danych jako mechanizmami dostępu do zdalnych baz danych.



Ćwiczenie 1 - łączniki bazy danych (2)

6. Korzystając z odpowiednich perspektyw słownikowych odczytać informacje o utworzonym łączniku i synonimach
7. Utworzyć kopię tabeli R_ROR korzystając z polecenia: `CREATE TABLE ... AS SELECT`
8. Zmodyfikować zawartość skopiowanej tabeli
9. Utworzyć perspektywę udostępniającą zawartość tabel R_ROR w obu bazach danych, tj. BD1, BD2
10. Odczytać dane za pomocą powyższej perspektywy



Ćwiczenie 2 - migawki (1)

1. Utworzyć migawkę SN_ROR dla tabeli scott.r_ror@BD1, o następujących parametrach
 - zapytanie wyznacza wszystkie rekordy tabeli
 - migawka typu PRIMARY KEY
 - pierwsze odświeżenie: 1 minuta po utworzeniu
 - odświeżanie manualne
 - odświeżanie pełne
2. Odczytać dane z migawki
3. Odświeżyć ręcznie migawkę

Celem ćwiczenia 2 jest zapoznanie się z migawkami jako mechanizmami replikowania i odświeżania danych pochodzących ze zdalnych baz danych.



Ćwiczenie 2 - migawki (2)

4. Utworzyć migawkę SN_TERM dla tabeli scott.r_terminowe@BD1, o następujących parametrach
 - zapytanie wyznacza wszystkie rekordy tabeli
 - migawka typu PRIMARY KEY
 - pierwsze odświeżenie: natychmiast po utworzeniu
 - okres odświeżania: 20 sekund
 - odświeżanie przyrostowe



Ćwiczenie 3 - grupy odświeżania (1)

1. Utworzyć pustą grupę odświeżania o następujących parametrach
 - pierwsze odświeżenie: natychmiast po utworzeniu
 - okres odświeżania: 10 sekund
2. Dodać do grupy migawki *sn_ror* i *sn_term*
3. Zmodyfikować zawartość tabel master obu migawek
4. Odświeżyć ręcznie grupę
5. Sprawdzić zawartość migawek
6. Usunąć grupę

Celem ćwiczenia 3 jest zapoznanie się z grupami odświeżania jako mechanizmami jednoczesnego odświeżania wielu migawek.



Ćwiczenie 4 - perspektywy słownikowe

1. Korzystając perspektyw słownikowych wyświetlić informacje na temat:
 - utworzonych przez siebie łączników bazy danych
 - utworzonych przez siebie migawek i aktualnych czasów ich odświeżania
 - utworzonych przez siebie dzienników migawek
 - utworzonych przez siebie grup odświeżania i migawek w grupach

Celem ćwiczenia 4 jest zapoznanie się z perspektywami słownikowymi udostępniającymi różne informacje nt. łączników i migawek.