

Partycjonowanie

1. Utwórz tabelę PURCHASES partycjonowaną zakresowo, z czterema partycjami. Schemat PURCHASES przedstawiono poniżej. Pierwsze trzy partycje tabeli PURCHASES mają przechowywać dane o sprzedaży odpowiednio: w styczniu 2017, lutym 2017 i marcu 2017. Czwarta partycja ma przechowywać pozostałe dane o sprzedaży. Atrybutem partycjonującym jest purchase_date.

Uwaga: zakładamy, że w hurtowni danych nie będą przechowywane dane wcześniejsze niż ze stycznia 2017.

PURCHASES		
Name	Null?	Type
-----	-----	-----
PRODUCT_ID	NOT NULL	VARCHAR2 (8)
TIME_KEY	NOT NULL	DATE
CUSTOMER_ID	NOT NULL	VARCHAR2 (10)
PURCHASE_DATE		DATE
PURCHASE_TIME		NUMBER (4)
PURCHASE_PRICE		NUMBER (6, 2)
SHIPPING_CHARGE		NUMBER (5, 2)
STATE_ID		VARCHAR2 (2)
TODAY_SPECIAL_OFFER		VARCHAR2 (1)

2. Dodaj do tabeli PURCHASES nową partycję przechowującą rekordy sprzedaży z kwietnia 2017.

3. Utwórz tabelę PURCHASES1 partycjonowaną haszowo, z trzema partycjami. PURCHASES1 ma mieć taki sam schemat jak tabela PURCHASES. Kluczem partycjonującym ma być atrybut PRODUCT_ID.

4. Utwórz tabelę REGIONAL_SALES partycjonowaną listowo. Schemat REGIONAL_SALES przedstawiono poniżej. Tabela ma posiadać 6 partycji, których atrybutem partycjonującym jest STATE. Nazwy poszczególnych partycji i zbiory wartości atrybutu STATE przedstawiono poniżej.

NORTHEAST -> 'NH', 'VT', 'MA', 'RI', 'CT'
SOUTHEAST -> 'NC', 'GA', 'FL'
NORTHWEST -> 'WA', 'OR'
MIDWEST -> 'IL', 'WI', 'OH'
WEST -> 'CA', 'NV', 'AZ'
OTHERSTATES -> pozostałe

REGIONAL_SALES		
Name	Null?	Type
-----	-----	-----
STATE	NOT NULL	VARCHAR2 (2)
STATE_NAME	NOT NULL	VARCHAR2 (20)
PRODUCT_ID	NOT NULL	VARCHAR2 (8)
PRODUCT_SALE	NOT NULL	NUMBER (8)

5. Utwórz tabelę PURCHASES_JAN2017 jako kopię tabeli ZSBD_TOOLS.PURCHASES (wykorzystaj polecenie create table as select), zawierającą wyłącznie dane o sprzedaży ze stycznia 2017 (atrybut purchase_date).

6. Wymień dane między tabelą PURCHASES_JAN2017, a partycją tabeli PURCHASES przechowującą dane ze stycznia 2017 (utworzoną w ćwiczeniu 1).

7. Wczytaj do tabeli PURCHASES dane o sprzedaży w lutym 2017. Źródłem danych jest tabela ZSBD_TOOLS.PURCHASES. Odczytaj zawartość partycji przechowującej dane z lutego 2017 (z własnej tabeli PURCHASES).

8. Porównaj plany wykonania poniższych zapytań (ikona Explain Plan... w SQL Developer).

```
select count(*) from purchases
where TO_CHAR(TIME_KEY, 'YYYY-MM') = '2017-01';
```

```
select count(*) from purchases
where TIME_KEY between DATE '2017-01-01' and DATE '2017-01-31';
```

9. Utwórz tabelę PRODUCTS_PART, o schemacie przedstawionym poniżej, partycjonowaną listowo z trzema partycjami: pierwsza dla produktów z kategorii 'HDRW', druga - 'ELEC', a trzecia - 'MUSC'.

product_id	varchar2(8) PRIMARY KEY
product_name	varchar2(30)
category	varchar2(4)
cost_price	number(6,2)
sell_price	number(6,2)
weight	number(4,2)
shipping_charge	number(5,2)
manufacturer	varchar2(20)
supplier	varchar2(10)

10. Wczytaj dane do tabeli PRODUCTS_PART z tabeli ZSBD_TOOLS.PRODUCT.

11. Utwórz tabelę SALES_PART_PROD, o schemacie przedstawionym poniżej, partycjonowaną referencyjnie, tak, aby podział na partycje odpowiadał podziałowi na partycje tabeli PRODUCTS_PART.

product_id	varchar2(8) FK -> Products_Part(product_id)
time_key	date
customer_id	varchar2(10)
purchase_date	date
purchase_time	number(4,0)
purchase_price	number(6,2)
shipping_charge	number(5,2)
state_id	varchar2(2)
today_special_offer	varchar2(1)

Uwaga: Poprawnie zdefiniuj ograniczenie klucza obcego.

12. Sprawdź jakie partycje posiada tabela SALES_PART_PROD.

13. Wczytaj z tabeli ZSBD_TOOLS.PURCHASES dane do tabeli SALES_PART_PROD i sprawdź ile rekordów znajduje się w każdej partycji.

14. Zbadaj plany wykonania poniższych zapytań (ikona Explain Plan... w SQL Developer).

```
select product_name, count(1)
from Products_Part p, Sales_Part_Prod s
where p.product_id=s.product_id
and p.category='ELEC'
group by product_name;
```

```
select product_name, count(1)
from Products_Part p, Sales_Part_Prod s
where p.product_id=s.product_id
group by product_name;
```

15. Zbierz statystyki dla obu tabel wykorzystywanych w powyższych zapytaniach, a następnie ponownie zbadaj plany wykonania tych zapytań. Wykorzystaj poniższy szablon polecenia zbierającego statystyki:

```
EXEC DBMS_STATS.gather_table_stats('nazwa-uzyt', 'nazwa-tabeli',
estimate_percent => DBMS_STATS.auto_sample_size);
```

16. Utwórz tabelę SALES_DYN_PART o poniższym schemacie, z dynamicznymi partycjami, z których każda przechowuje dane z jednego miesiąca. Warunek „less than” niech przyjmuje wartość pierwszego dnia bieżącego miesiąca.

Name	Null?	Type
PRODUCT_ID	NOT NULL	VARCHAR2(8)
PURCHASE_DATE		DATE
QUANTITY		NUMBER(3)

17. Wstaw do tabeli SALES_DYN_PART kilka rekordów testowych z datami z różnych miesięcy. Sprawdź w słowniku bazy danych czy i jakie partycje zostały utworzone dla tej tabeli.

18. Utwórz indeks lokalny poniższym poleceniem. Wyświetl informacje o partycjach utworzonego indeksu. Następnie usuń ten indeks.

```
create index purchases_local_idx
on purchases(purchase_date)
local;
```

19. Utwórz indeks lokalny poniższym poleceniem. Wyświetl informacje o partycjach utworzonego indeksu. Następnie usuń ten indeks.

```
create index purchases_local_idx
on purchases(purchase_date)
local
(partition indx_purch_jan2017,
partition indx_purch_feb2017,
partition indx_purch_mar2017,
partition indx_purch_recent);
```

20. Utwórz indeks globalny bez prefiksu na atrybucie PURCHASES.PRODUCT_ID. Indeks ma być podzielony na trzy partycje: pierwsza o wartościach PRODUCT_ID < 'SP1000', druga o wartościach PRODUCT_ID < 'SP2000', a trzecia o wartościach PRODUCT_ID >='SP2000'. Wyświetl informacje o partycjach utworzonego indeksu. Następnie usuń ten indeks.

21. Usuń wszystkie utworzone w czasie ćwiczeń tabele.