Advanced Databases

INZ000109P

Project

Assignment 8 - Partition (spe.)

Group: A2

Furkan ÖCALAN 257638

Mustafa Tayyip BAYRAM 257639

1. Hash Partitions

Hash Partition useful for situations where the ranges are not applicable such as product ID, employee number and the like. For this spreading out, hash keys are used effectively and efficiently.

For optimal data distribution, the following requirements should be satisfied:

1. Choose a column or combination of columns that is unique or almost unique.
2. Create multiple partitions and subpartitions for each partition that is a power of two. For example, 2, 4, 8, 16, 32, 64, 128, and so on...

1st Partition: CREATE TABLE

```
 orders_hash (order_id  NUMBER,
  order_date DATE,
  store_id
                NUMBE
  R, staff_id
  NUMBER)
PARTITION BY HASH(order_id) PARTITIONS 4;
```

2nd Partition: CREATE TABLE

```
 brands_hash (brand_id  NUMBER,
  brand_name VARCHAR(100) NOT NULL,
)
PARTITION BY
HASH(brand_id) PARTITIONS
4
STORE IN (brand1, brand2, brand3, brand4);
```

2. Range Partitions

Range partitioning maps data to partitions based on ranges of values of the partitioning key that you establish for each partition. It is the most common type of partitioning and is often used with dates

Range partitioning is useful when you have distinct ranges of data you want to store together.

3rd partition: CREATE TABLE

```
orders_range ( order_id     NUMBER,
customer_id
                NUMBER,
order_date      DATE,
store_id     NUMBER
)
PARTITION BY RANGE (order_date)
( PARTITION sales_q1_2006 VALUES LESS THAN (TO_DATE('01-APR-2006','dd-MON-yyyy')
  TABLESPACE tsa
, PARTITION sales_q2_2006 VALUES LESS THAN (TO_DATE('01-JUL-2006','dd-MON-
  yyyy')) TABLESPACE tsb
, PARTITION sales_q3_2006 VALUES LESS THAN (TO_DATE('01-OCT-2006','dd-MON-
  yyyy')) TABLESPACE tsc
, PARTITION sales_q4_2006 VALUES LESS THAN (TO_DATE('01-JAN-2007','dd-MON-
  yyyy')) TABLESPACE tsd
);
```

3. Value List Partitions

Value List partitioning useful when you want to specifically map rows to partitions based on discrete values

Unlike range and hash partitioning, **multi-column partition keys are not supported** for list partitioning. If a table is partitioned by list, the partitioning key can only consist of a single column of the table.

4th Partition: CREATE TABLE

```
customers ( customer_id     NUMBER
, last_name VARCHAR (100) NOT NULL
, city       VARCHAR(255)
, state        VARCHAR(255)
)
PARTITION BY LIST (city)
( PARTITION p_northwest VALUES ('OR', 'WA')
, PARTITION p_southwest VALUES ('AZ', 'UT', 'NM')
, PARTITION p_northeast VALUES ('NY', 'VM', 'NJ')
, PARTITION p_southeast VALUES ('FL', 'GA')
, PARTITION p_northcentral VALUES ('SD', 'WI')
, PARTITION p_southcentral VALUES ('OK', 'TX')
);
```

5th Partition: CREATE TABLE
stores (store_ id NUMBER
, store_name NUMBER
, customer_id NUMBER
, city VARCHAR
, state VARCHAR(2)
, zip_code VARCHAR2(1)
)
PARTITION BY LIST (state)
(PARTITION p_northwest VALUES ('OR', 'WA')
, PARTITION p_southwest VALUES ('AZ', 'UT', 'NM')
, PARTITION p_northeast VALUES ('NY', 'VM', 'NJ')
, PARTITION p_southeast VALUES ('FL', 'GA')
, PARTITION p_northcentral VALUES ('SD', 'WI')
, PARTITION p_southcentral VALUES ('OK', 'TX')
);