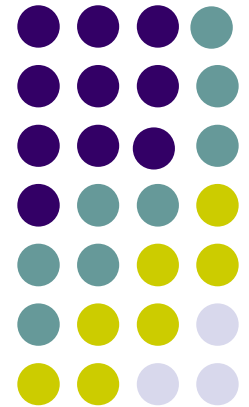


Aplicação Prática - Oracle

Pablo Vieira Florentino



Aplicação e suporte à fragmentação



- Existem hoje diversos SGBDs com suporte a distribuição de bases de dados, seja para alocação, seja para fragmentação de tabelas ou componentes de uma determinada base de dados. Entre estes SGBDs, podemos citar: Oracle, Non Stop SQL/Compaq Informix, SQL Server, entre outros.
- Serão apresentadas fragmentações horizontais de diversas tabelas utilizando o Oracle, tendo definido previamente os seus projetos de fragmentação.

Fragmentação horizontal no Oracle



- Suporte à 3 formas de fragmentação:
 - **List:**
 - controla explicitamente como será a fragmentação, determinando como cada registro é alocado a uma partição.
 - Especifica uma **lista** prévia de valores que servirão como chave de fragmentação na descrição de cada partição.
 - Exemplo: comando PARTITION BY LIST
 - recebe como parâmetro a coluna a ser utilizada na fragmentação (*state*), seguida da definição de cada partição contendo o nome da partição (*northwest, southwest, ...*), e os valores da coluna utilizada (*state*) que definem o conteúdo da partição.

Exemplo 1



```
CREATE TABLE sales_by_region (  
deptno          NUMBER(10),  
deptname       VARCHAR2(20),  
quarterly_sales NUMBER(10,2),  
state          VARCHAR2(2))  
PARTITION BY LIST (state) (  
  PARTITION westcent VALUES ('MT', 'GO'),  
  PARTITION northeast VALUES ('PI', 'PB', 'RN', 'PE'),  
  PARTITION southeast VALUES ('ES', 'MG'),  
  PARTITION north VALUES ('AM', 'PA'),  
  PARTITION south VALUES ('RS', 'PR'));
```

Fragmentação horizontal no Oracle



- **Range:** registros são mapeados para as partições baseando-se nos intervalos de valores de cada partição da(s) chave(s) definida(s) para fragmentação.
 - Exemplo: uso desta abordagem com chave de fragmentação simples, indicando em que *tablespace* a partição será alocada.
 - O comando PARTITION BY RANGE recebe como parâmetro a coluna a ser utilizada na fragmentação (*record_date*), seguida da definição de cada *tablespace* contendo o nome da partição (*yr0, yr1, ...*), e os intervalos de valores do campo usado como chave de fragmentação (*record_date*) que definem o conteúdo da partição (*até 01/01/2000, de 01/01/2001 a 01/01/2002, ...*).

Exemplo 2



```
CREATE TABLE professional_history (  
  prof_history_id      NUMBER(10),  
  person_id           NUMBER(10) NOT NULL,  
  organization_id     NUMBER(10) NOT NULL,  
  record_date         DATE NOT NULL,  
  ph_comments         VARCHAR2(2000))  
PARTITION BY RANGE (record_date) (  
  PARTITION yr0  
  VALUES LESS THAN (TO_DATE('01-JAN-2000','DD-MON-YYYY'))  
  TABLESPACE <tablespace_name>,  
  PARTITION yr1  
  VALUES LESS THAN (TO_DATE('01-JAN-2001','DD-MON-YYYY'))  
  TABLESPACE <tablespace_name>,  
  PARTITION yr2  
  VALUES LESS THAN (TO_DATE('01-JAN-2002','DD-MON-YYYY'))  
  TABLESPACE <tablespace_name>,  
  PARTITION yr9  
  VALUES LESS THAN (MAXVALUE)  
  TABLESPACE <tablespace_name>);
```

Exemplo 3



- A chave de fragmentação pode também envolver mais de um campo da tabela:
 - PARTITION BY RANGE recebe mais de um campo como parâmetro
 - Da mesma forma, a definição das faixas de valores recebe mais de um valor, na mesma ordem em que os campos usados como chave de fragmentação foram listados.

```
CREATE TABLE sales (  
  invoice_no NUMBER,  
  sale_year INT NOT NULL,  
  sale_month INT NOT NULL,  
  sale_day INT NOT NULL )  
PARTITION BY RANGE (sale_year, sale_month, sale_day)  
  ( PARTITION sales_q1 VALUES LESS THAN (1997, 04, 01)  
    TABLESPACE <tablespace_name>,  
    PARTITION sales_q2 VALUES LESS THAN (1997, 07, 01)  
    TABLESPACE <tablespace_name>,  
    PARTITION sales_q3 VALUES LESS THAN (1997, 10, 01)  
    TABLESPACE <tablespace_name>,  
    PARTITION sales_q4 VALUES LESS THAN (1998, 01, 01)  
    TABLESPACE <tablespace_name> );
```

Fragmentação horizontal no Oracle



- **Hash:** em geral, é utilizada quando não se encontram características de acesso que indiquem fragmentação por Range (faixa de valores) ou por List (lista de valores).
 - Utiliza uma função hash – interna ao Oracle - que faz o *split* dos registros entre as partições definidas no momento da fragmentação.
 - Exemplo: O comando PARTITION BY HASH recebe como parâmetro a coluna que possui os valores a serem utilizados pela função hash (neste caso, o campo *prof_history_id*) seguida do número de partições a serem criadas (neste caso, 4) e dos tablespaces onde serão alocadas as partições

Exemplo

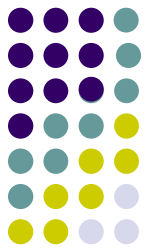


```
CREATE TABLE professional_history (  
  prof_history_id      NUMBER(10),  
  person_id           NUMBER(10) NOT NULL,  
  organization_id     NUMBER(10) NOT NULL,  
  record_date         DATE NOT NULL,  
  prof_hist_comments  VARCHAR2(2000))  
PARTITION BY HASH (prof_history_id)  
PARTITIONS 4  
STORE IN (  
  tablespace_n1,  
  tablespace_n2,  
  tablespace_n3,  
  tablespace_n4);
```

Fragmentação horizontal no Oracle



- **Composite Partitioning:** combinação de duas técnicas aplicadas conjuntamente sobre uma tabela. *Suporta* dois tipos de combinações para fragmentação: Range-List e Range-Hash.
 - Primeiramente a fragmentação por Range (faixa de valores) é aplicada e, posteriormente, são criadas subpartições, ou por List, ou por Hash.
 - Exemplo com Range-Hash:Primeiramente é informada a fragmentação principal - `PARTITION BY RANGE(sales_date)`, seguida imediatamente da definição das quatro subpartições por hash (`SUBPARTITION BY HASH(salesman_id)`) com seus respectivos nomes (*sp1, sp2, ...*). Somente após a definição das subpartições, são definidas as partições para a fragmentação principal por faixas de valores para o campo *sales_date*. Assim, cada fragmento principal possuirá quatro subpartições.



```
CREATE TABLE sales_composite (  
  salesman_id          NUMBER(5),  
  salesman_name       VARCHAR2(30),  
  sales_amount        NUMBER(10),  
  sales_date          DATE)  
PARTITION BY RANGE(sales_date)  
SUBPARTITION BY HASH(salesman_id)  
SUBPARTITION TEMPLATE(  
  SUBPARTITION sp1 TABLESPACE data1,  
  SUBPARTITION sp2 TABLESPACE data2,  
  SUBPARTITION sp3 TABLESPACE data3,  
  SUBPARTITION sp4 TABLESPACE data4)  
(PARTITION sales_jan2000 VALUES LESS THAN(TO_DATE  
'02/01/2000','DD/MM/YYYY'))  
PARTITION sales_feb2000 VALUES LESS THAN(TO_DATE  
'03/01/2000','DD/MM/YYYY'))  
PARTITION sales_mar2000 VALUES LESS THAN(TO_DATE  
'04/01/2000','DD/MM/YYYY'))  
PARTITION sales_apr2000 VALUES LESS THAN(TO_DATE  
'05/01/2000','DD/MM/YYYY'))  
PARTITION sales_may2000 VALUES LESS THAN(TO_DATE  
'06/01/2000','DD/MM/YYYY'))  
PARTITION sales_future VALUES LESS THAN(MAXVALUE))
```

Exemplo