

Профессионалу-разработчику.

Секционирование и сжатие в хранилищах данных Oracle.

Источник: сайт Realcoding.NET “Программирование для всех”,
статья <http://www.realcoding.net/article/view/4495>



Екатерина Кондратьева
ekondratieva@bk.ru

СОДЕРЖАНИЕ

- Введение
- Описание проблемы
- Предлагаемое решение
- Заключение

Введение

Сегодня на многих предприятиях идет активное внедрение информационно-аналитических хранилищ данных. В результате обозначались и сформировались подходы и принципы, позволяющие оптимизировать построение систем подобного рода на платформе Oracle Database. Один из таких подходов и будет рассмотрен в статье.

Важной характеристикой хранилища данных является использование такого специализированного измерения как “Время”. В абсолютном большинстве случаев значения показателей равномерно распределяются по этому измерению.

Учитывая вышеописанное свойство хранилища данных, в качестве метода для поддержки больших объемов данных информации, чаще всего применяют опцию Partitioning СУБД Oracle. При этом происходит разбиение (секционирование) больших таблиц на относительно небольшие разделы (секции). Каждая секция может храниться в отдельном табличном пространстве, что позволяет повысить управляемость, доступность и производительность системы.

Описание проблемы

К основным проблемным местам хранилища относят следующие:

- Постоянно растущий объем данных по мере эксплуатации системы, и как следствие, высокие

требования к объему дискового пространства;

- Трудно прогнозируемое время выполнения запросов, и длительное время их выполнения, как следствие, большая нагрузка на дисковую подсистему.

Предлагаемое решение

Для “борьбы” со вышеперечисленными недостатками хранилищ данных предлагается разбивать факторные таблицы на секции и периодически сжимать архивные секции. Необходимо отметить, что в хранилищах данных основной объем информации приходится именно на факторные таблицы.

Далее будет рассмотрен законченный пример с помощью которого можно будет реально оценить эффективность предлагаемого подхода.

На схеме приведена структура факторной таблицы sample, которая содержит данные по продажам.

Sample
time_id
product_id
customer_id
department_id
sale

Рис.1 Структура факторной таблицы “Sample”

Данные по продажам формируются по четырем аналитикам: – “Время”, “Продукт”, “Клиент”, “Отдел продаж”. Минимальный уровень по измерению “Время” – год. Данные факторной таблицы можно разбить на логические диапазоны по годам. Т.е. в каждой секции

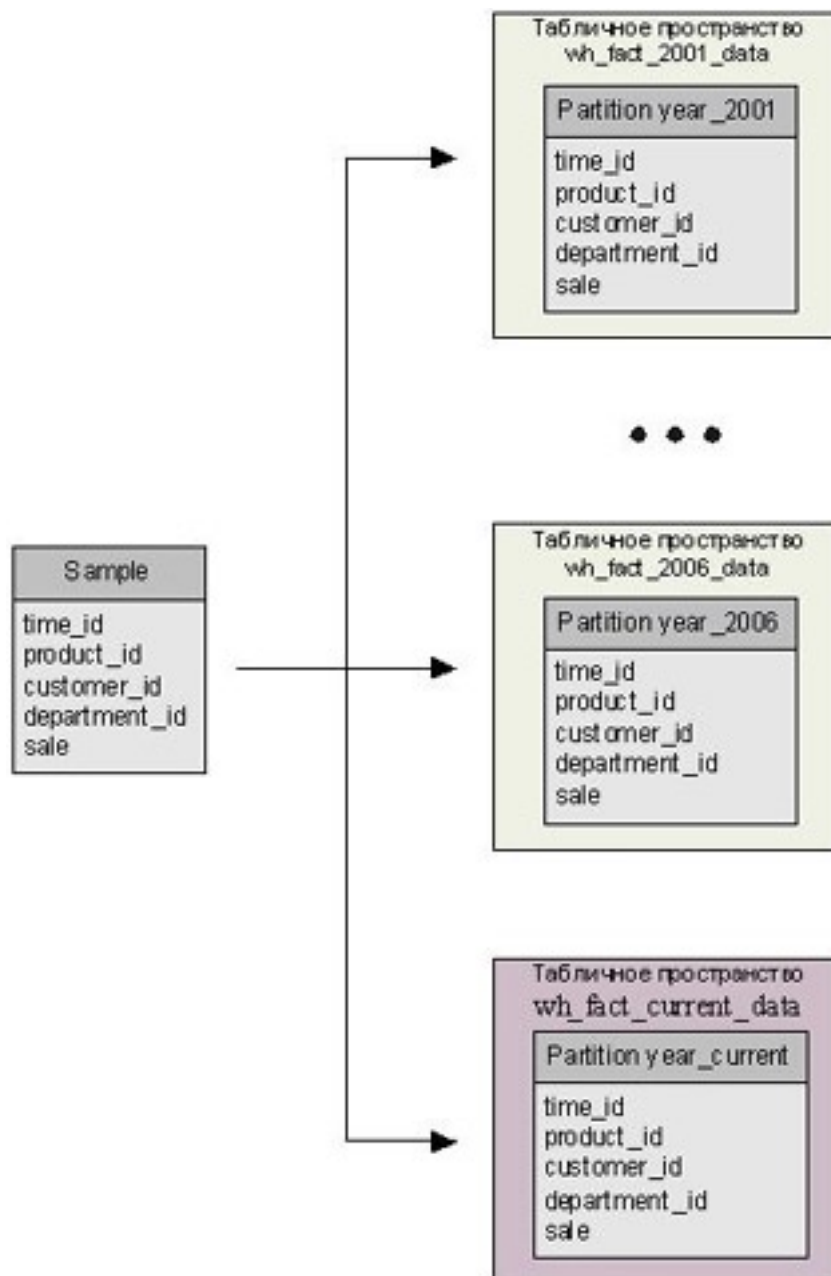


Рис.2 Секционирование факторной таблицы "Sample"

будут содержаться данные за определенный год. Наибольшая производительность при работе с секционированными таблицами достигается при равномерном распределении данных по секциям. На схеме представлена таблица sample, секционированная по годам. Каждый раздел хранится в отдельном табличном пространстве. Секция year_current является загрузочной, то есть в текущий период времени (на рисунке - за 2007 год) данные из информационных источников поступают только в эту секцию; в остальных секциях хранятся архивные данные.

Перед созданием таблицы необходимо сформировать табличные пространства для ее секций. Пример:

```
create tablespace wh_fact_2001_
data datafile 'c:\oracle\oradata\
wawh_fact_2001_data01.dbf'
size 50M autoextend on next 10M
maxsize 2048M
extent management local
segment space management auto;
... ..
```

В прилагаемом файле (http://www.realcoding.net/dn/oracle_scripts/scripts/create_tablespace.sql) приведен полный скрипт по созданию табличных пространств. Далее создаем факторную таблицу sample. Пример:

```

create table sample
(
  time_id      number(9),
  product_id   number(9),
  customer_id  number(9),
  department_id number(9),
  sale         number
)
nologging
partition by range(time_id)
(
  partition year_2001 values less than
(-3995) nocompress tablespace wh_fact_
2001_data,
  ... ..

```

В целях эксперимента, индексы на столбцы таблицы отсутствуют.

Законченный скрипт создания факторной таблицы приведен в файле (http://www.realcoding.net/dn/oracle_scripts/scripts/create_table.sql).

Следующим шагом заполняем факторную таблицу sample случайными значениями.

Приведенный ниже код служит лишь для заполнения данных в таблицу Sales; в реальной производственной системе загрузка данных будет происходить с помощью специализированных ETL-процедур. Код для загрузки данных в факторную таблицу:

```

... ..
  for v_xTimeId in 1..1005
  loop
    for v_xVersionId in 1..10
    loop
      for v_xProductId in 1..10
      loop
        for v_xCustomerId in 1..10
        loop
          for v_xDepartmentId in
1..20
          loop
            v_xCount := v_xCount + 1;
            v_xFactValue := to_number(to_
char(sysdate,'ss'));

            v_xTimeIds(v_xCount)      :=
v_xTimeId;
            v_xProductIds(v_xCount)
:= v_xProductId;
            v_xCustomerIds(v_xCount)
:= v_xCustomerId;
            v_xDepartmentIds(v_xCount)

```

```

:= v_xDepartmentId;
            v_xFactValues(v_xCount)
:= v_xFactValue;

            if v_xCount = v_xCount-
Limit then
              v_xCount := 0;

              forall v_xIndex in 1..v_
xCountLimit
                insert into sample
values(v_xTimeIds(v_xIndex),
       v_xProductIds(v_xIndex),
       v_xCustomerIds(v_xIndex),
       v_xDepartmentIds(v_xIndex),
       v_xFactValues(v_xIndex));
            end if;
          end loop;
        end loop;
      end loop;
    end loop;
  end loop;

```

Законченный скрипт приведен в файле по следующей ссылке (http://www.realcoding.net/dn/oracle_scripts/scripts/fill_table.sql).

Далее проверяем: какое количество записей содержится в секции *year_current*.

```

select count(*) from sample partition
(year_current);

```

Итак, сейчас в таблице samples содержится 20 миллионов записей.

После вставки записей собираем статистику для таблицы sample. Для этого необходимо вызвать процедуру GATHER_TABLE_STATS из пакета DBMS_STATS. Код для сбора статистики:

```

begin
  DBMS_STATS.GATHER_TABLE_STATS (own-
name => 'DEMO', tabname => 'sample');
end;

```

Скрипт сбора статистики приведен в файле по следующей ссылке (http://www.realcoding.net/dn/oracle_scripts/scripts/stats_sample.sql).

Повторно запускаем sql-запрос на подсчет количества записей в секции *year_current* и sql-запрос на поиск записи с кодом клиента для полного сканирования всей таблицы и анализируем время выполнения этих запросов.

```
select count(*) from sample partition
(year_current);

select * from sample partition (year_
2007) where customer_id = 999;
```

Скрипты представлены в файле (http://www.realcoding.net/dn/oracle_scripts/scripts/sql_analyze.sql).

Время выполнения запроса А – 7,77 секунд (время сканирования секции *year_current*), время выполнения запроса В – 21,20 секунд (время полного сканирования таблицы).

Следующим шагом, с помощью оператора *split partition* секцию *year_current* разбиваем на две – на секцию *year_2007*, куда относятся значения показателя для 2007 года, и на секцию *year_current*, готовую для следующей порции загружаемых данных. Предварительно создаем табличное пространство для данных 2007 года.

```
create tablespace wh_fact_2007_data
datafile 'c:oracleoradatadwawwh_fact_
2007_data01.dbf'
size 50M autoextend on next 10M
maxsize 2048M
extent management local
segment space management auto;
```

Скрипт по созданию табличного пространства приведен в файле (http://www.realcoding.net/dn/oracle_scripts/scripts/create_tablespace_2007.sql).

Расщепляем партицию *year_current* на две, используя диапазон значений поля *time_id*, принадлежащих 2007 году.

```
alter table sample split partition
year_current at (1005)
into
(
partition year_2007 tablespace
wh_fact_2007_data,
partition year_current
);
```

Скрипт приведен в файле (http://www.realcoding.net/dn/oracle_scripts/scripts/alter_table.sql).

Следующим шагом исследования будет сжатие (*compress*) вновь полученной архивной секции (в нашем случае - *year_2007*) и выяснение: насколько это повысит быстродействие системы. Выясняем: сколько места на диске занимает секция *year_2007*, - для этого используем представление *dba_segments*.

```
select * from dba_segments where owner
= 'DEMO';
```

Размер секции *year_2007* составляет 464 МБ.

Далее, с помощью оператора, *move partition* производим сжатие секции *year_2007*.

```
alter table sample move partition
year_2007 tablespace wh_fact_2007_data
compress;
```

Повторно выясняем размер секции *year_2007*. Теперь, после сжатия, размер секции *year_2007* составляет 240 МБ. Таким образом, коэффициент сжатия равен 1,93.

После того, как поступит уведомление о том, что данные 2007 года больше модифицироваться не будут (т.е. этот период в учетных системах закрыт), для ускорения времени выполнения запросов на чтение, и контроля на изменение данных, переведем табличное пространство *wh_fact_2007_data* в режим “только для чтения” (*read only*).

```
alter tablespace wh_fact_2007_data
read only;
```

Проверяем как изменилось время выполнения тестовых запросов (файл - http://www.realcoding.net/dn/oracle_scripts/scripts/sql_analyze.sql).

Время выполнения запроса А – 3,11 секунд, время выполнения запроса В – 9,06 секунд.

Время выполнения запросов существенно сократилось - примерно в 2,5 раза !

Далее выполняем команду на уменьшение файла данных вновь полученного табличного пространства с архивными данными.

```
alter database datafile 'C:oracleora-
datadwawwh_fact_current_data01.dbf'
resize 245M;
```

Заключение

Были получены следующие положительные результаты:

- Достаточно большой коэффициент сжатия секции, что позволяет уменьшить размер табличного пространства;
- Увеличение скорости выполнения запросов;
- Контроль на DML-операции после завершения формирования загрузки.

Законченные скрипты, использованные в статье, можно загрузить по следующей ссылке (http://www.realcoding.net/dn/oracle_scripts/scripts/oracle_scripts.rar).