

Oracle Partitioning в Oracle Database 12c

Эффективное управление данными и повышение производительности для каждой системы

СТАТЬЯ ORACLE | СЕНТЯБРЬ 2014



Содержание

Общие сведения.....	3
Основы секционирования.....	4
Концепция секционирования	4
Секционирование для производительности.....	5
Секционирование для управляемости	7
Секционирование для доступности.....	7
Преимущества секционирования для управления жизненным циклом информации	8
Стратегии секционирования.....	8
Методы распределения данных для секционированных объектов.....	8
Простое (одноуровневое) секционирование	9
Композитное (двухуровневое) секционирование	9
Расширения секционирования	10
Интервальное секционирование	10
Ссылочное секционирование	10
Секционирование по виртуальному столбцу	11
Partition Advisor.....	11
Краткий обзор функциональных возможностей секционирования	12
Заключение	13

Общие сведения

Опция Oracle Partitioning была впервые представлена в Oracle 8.0 в 1997 г. Это одна из самых успешных и часто используемых опций СУБД Oracle Database. Oracle Partitioning позволяет разделять большие таблицы и индексы на меньшие части для повышения производительности, управляемости и доступности десятков тысяч приложений. Запросы и операции обслуживания ускоряются на порядок для критически важных систем любого типа (OLTP, хранилища данных или смешанные рабочие нагрузки) и любого размера (от сотен гигабайт до петабайтов). Если секционирование Oracle используется вместе с картами зон (zone maps) (новой возможностью в СУБД Oracle Database 12 Release 12.1.0.2 для Engineered Systems), таблицы и секции разбиваются на зоны меньшего размера, что обеспечивает практически неограниченное отсечение данных (data pruning).

С помощью секционирования архитекторы и администраторы баз данных могут решать ряд сложных проблем современных приложений. Секционирование — это основной инструмент для создания многотерабайтных систем или систем с чрезвычайно жесткими требованиями по доступности. Кроме того, секционирование может существенно снизить совокупную стоимость владения данными посредством подхода «многоуровневой архивации».

При этом более старая, но все еще актуальная информация по-прежнему доступна онлайн, но находится на недорогих устройствах хранения в оптимальном сжатом формате. Если секционирование используется вместе с Automatic Data Optimization и Heat Maps, новой функциональностью в Oracle Database 12c, оно обеспечивает простой и автоматический способ внедрения стратегии управления жизненным циклом информации (ILM).

Уже более десяти лет Oracle Partitioning повышает производительность, управляемость и доступность десятков тысяч систем. Oracle Database 12c представляет 10-е поколение секционирования, и клиенты могут использовать преимущества Oracle Partitioning для своих нужд, как никогда раньше.

Основы секционирования

Концепция секционирования

Секционирование дает возможность разделять таблицы и индексы на отдельные части меньшего размера. Каждая часть объекта базы данных называется секцией. У секции есть собственное имя и могут быть собственные характеристики хранения. С точки зрения администратора базы данных, секционированный объект состоит из нескольких частей, которыми можно управлять всеми вместе или по отдельности. Это обеспечивает администратору значительную гибкость в управлении секционированным объектом. Однако для приложения секционированная таблица выглядит так же, как несекционированная; для доступа

к секционированной таблице с помощью команд SQL DML никакие изменения не требуются. Логически это по-прежнему одна таблица.

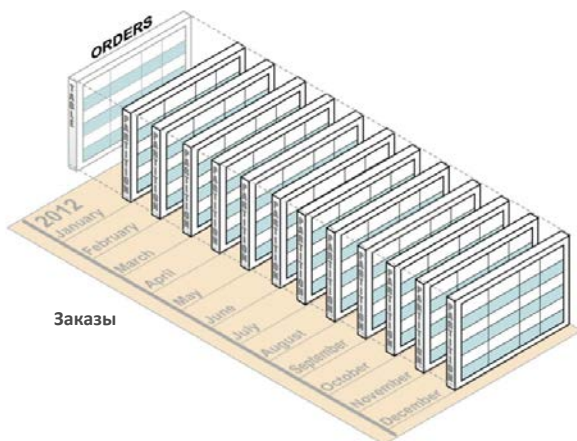


Рис. 1. Секционированная таблица с точки зрения АБД и приложения

Объекты базы данных (таблицы и индексы) секционируются с помощью **ключа секционирования** (partitioning key) — набора столбцов, которые определяют, в какую секцию будет попадать та или иная строка. В случае композитного секционирования таблицы секция делится дальше на подсекции, для чего используется еще один набор столбцов; размещение данных каждой строки определяется критериями обоих ключей секционирования, а строка помещается в соответствующую подсекцию¹. Например, таблица Orders (таблица заказов), показанная на рис. 1, секционирована

по диапазону значений столбца даты заказа и по месяцам, но для приложений это выглядит как единая «обычная» таблица.

Однако администратор базы данных может управлять секцией каждого месяца и хранить ее отдельно. Возможно, будут использоваться разные уровни хранения, к более старые данные будут сжиматься, а целые диапазоны старых данных будут храниться в табличных пространствах, доступных только для чтения. Разработчикам приложений обычно не нужно беспокоиться о том, секционирована таблица или нет, но они также могут выгодно использовать секционирование. Например, ресурсоемкую операцию DML для очистки данных можно реализовать с помощью операций управления секциями, что кардинально улучшит время выполнения и существенно снизит потребление ресурсов.

Независимо от выбранной стратегии секционирования, каждый индекс секционированной таблицы либо связан, либо не связан с базовой стратегией секционирования своей таблицы. В Oracle Database 12c различают три типа индексов.

Локальный индекс — это индекс секционированной таблицы, связанный со своей секционированной таблицей. Он «наследует» стратегию секционирования этой таблицы. Следовательно, каждая секция локального индекса соответствует одной-единственной секции базовой таблицы. Такая связь оптимизирует управление секциями. Например, когда секция таблицы удаляется, Oracle просто убирает соответствующую секцию индекса. Никакого дорогостоящего обслуживания индекса не требуется, так как по определению секция индекса связана только со своей секцией таблицы; локальный сегмент индекса никогда не будет содержать данные других секций. Локальные индексы наиболее распространены в средах хранилищ данных.

¹ Для облегчения понимания в остальной части этого документа мы будем говорить только о секциях.

Глобально секционированный индекс (global partitioned index) — это индекс секционированной или несекционированной таблицы, который секционирован по другому ключу секционирования или использует отличную от своей таблицы стратегию секционирования. Глобально секционированные индексы могут секционироваться по диапазону значений ключа или методом хеш-секционирования и их секционирование не связано с секционированием базовой таблицы. Например, таблица может секционироваться по диапазонам значений ключа по месяцам и иметь двенадцать секций, а ее индекс может быть секционирован методом хеш-секционирования с использованием другого ключа секционирования и иметь другое количество секций. Отделение индекса от его таблицы автоматически означает, что любая операция обслуживания секции таблицы может потребовать операций обслуживания индекса. Глобально секционированные индексы чаще используются в средах OLTP, чем в средах хранилищ данных.

Глобальный несекционированный индекс (global non-partitioned index) по существу подобен индексу несекционированной таблицы. Структура индекса не секционирована и отделена от секционирования базовой таблицы. В средах хранилищ данных глобальные несекционированные индексы чаще всего используются для обеспечения ограничений первичного ключа. OLTP окружения, с другой стороны, в основном полагаются на глобальные несекционированные индексы



Все вышеуказанные типы индексов создаются для всех секций секционированной таблицы (так называемая **«полная индексация»** (full indexing)) или, начиная с Oracle Database 12c, только для поднабора секций секционированной таблицы (так называемая **«частичная индексация»** (partial indexing)).

Частичными индексами могут быть только индексы секционированных таблиц. Будет ли индексироваться конкретная секция, определяется свойствами секции, и результат применяется ко всем частичным индексам. Например, если используется частичная индексация, вы можете решить не индексировать более старые секции, где доступ к данным всегда осуществляется без предикатов фильтра, которые могли бы выиграть от индексов. Часто не производится индексация вновь созданной секции, чтобы не выполнять работу по обслуживанию индекса во время вставки данных. Таким образом максимально ускоряется загрузка данных. Если использовать отсечение данных (data pruning) с помощью карты зон (zone map), то возможный ущерб от отсутствия индексов для выборочного доступа к данным самой новой секции сводится к минимуму.

Соответствующая стратегия индексации выбирается в зависимости от бизнес-требований и модели доступа к данным. Это дает возможность использовать секционирование для любых типов приложений.

Рис. 2. Индексация секционированных таблиц

Секционирование для производительности

Размещение определенной строки зависит от значения ключа секционирования в этой строке. Критерии того, как данные таблицы делятся по секциям, хранятся как метаданные секционирования таблицы или индекса. Для каждой операции SQL (запросы, DML и операции обслуживания секций) эти метаданные определяют секции таблицы, которых касается данная операция. База данных автоматически работает только с соответствующими секциями или, если используются карты зон (zone maps), только с частями секции или таблицы. Ограничивая объем данных, который нужен для просмотра или работы, секционирование обеспечивает повышение производительности.

Отсечение секции (partitioning pruning) (т. е. исключение секции) — это простейший и одновременно самый эффективный способ повышения производительности. Обработка запросов часто ускоряется на несколько порядков, так как метаданные секционирования ограничивают объем данных для операции SQL. Предположим, что приложение содержит таблицу Orders (таблицу заказов) с историей заказов и эта таблица секционирована по дням по дате заказа. Запросу данных одной недели потребовался бы доступ только к семи секциям таблицы заказов. Если таблица содержит исторические данные за 2 года, то запросу потребовался бы доступ только к семи секциям вместо 730. Благодаря отсечению секций этот запрос может выполняться в 100 раз быстрее. Отсечение секций работает со всеми другими возможностями Oracle для повышения производительности. Oracle может использовать отсечение секций в сочетании с любым другим методом индексирования, соединения или параллельного доступа.

Zone maps² (Карты зон) — это новое средство в Oracle Database 12c Release 12.1.0.2 для систем «под ключ» (Engineered Systems). Оно расширяет возможности отсечения данных в базе данных Oracle по сравнению с использованием метаданных секционирования таблицы. Отсечение данных возможно на уровне секций и даже на еще более мелком уровне так называемых «зон». Зона — это непрерывная последовательность блоков, для которой карта зон отслеживает минимальное и максимальное значения в заданных столбцах. Если таблица секционирована, карта зон также содержит агрегированные минимальные и максимальные значения столбцов для каждой секции. Каждый раз, когда операция SQL использует столбцы, указанные в карте зон, для ограничения (фильтрации) требуемых данных, Oracle сравнивает фильтр с информацией карты зон и не обращается к зонам и секциям, которые не содержат соответствующих данных. В этом смысле карты зон аналогичны индексам хранения Exadata (Exadata Storage Index), но предоставляют новые преимущества в дополнение к индексам хранения. Карты зон — это сохраняемые структуры данных, которые обрабатываются в базе данных. В них можно задавать локальные столбцы (столбцы таблицы с картой зон) и столбцы с соединением (joined columns).

Наличие карты зон в базе данных удобно для создания различных отчетов. Если использовать таблицу заказов из предыдущего примера, то любому запросу информации о заказах, отгруженных в определенный период времени, потребуются доступ ко всем секциям таблицы заказов (потому что ключом секционирования была дата заказа, а не дата отгрузки). Хотя между датой заказа и датой отгрузки есть корреляция, невозможно ограничить число секций для доступа только датой отгрузки. Но если используются карты зон, базе данных также известны минимальное и максимальное значения дат отгрузок; карта зон хранит эту информацию для каждой секции. Если дата заказа и дата отгрузки находятся в пределах одной рабочей недели, то запросам о продуктах, отгруженных за последние три недели, потребуются доступ только к секциям заказов последних четырех недель, а внутри этих секций — только к зонам отгрузок за этот период времени.

Секционирование может также ускорить многотабличные соединения с помощью метода **соединения по секциям** (partition-wise join). Соединения по секциям могут применяться, когда соединяются две таблицы и, по крайней мере, одна из них секционирована по ключу соединения. Соединения по секциям разбивают большое соединение на меньшие с «идентичными» наборами данных для соединяемых таблиц. «Идентичность» здесь определяется как совпадение наборов значений ключа секционирования на обоих концах соединения, поэтому только соединение этих «идентичных» наборов данных даст результат и рассматривать другие наборы данных не требуется. Oracle или использует факт уже эквисекционированных (физических) таблиц для соединения, или прозрачно перераспределяет («пересекционирует») одну таблицу, меньшую по размеру, во время выполнения запроса для создания эквисекционированных наборов данных, соответствующих секционированию другой таблицы. Таким образом, вся операция соединения выполняется быстрее и с меньшими потребностями в ресурсах. Это дает значительные преимущества производительности как для последовательного, так и для параллельного выполнения.

² Карты зон подробно обсуждаются в руководстве по хранилищам данных Oracle (Oracle Data Warehousing Guide).

Секционирование для управляемости

Поскольку секционирование разбивает таблицы и индексы на более мелкие и более управляемые части, администраторы баз данных могут использовать подход «разделяй и властвуй» для управления данными. Oracle предоставляет полный набор команд SQL для управления секционированием таблиц. Это команды для добавления новых секций, а также для удаления, разделения, переноса, слияния, быстрой очистки секций и обмена ими.

Если используется секционирование, операции обслуживания могут выполняться с определенными частями таблицы. Например, администратор базы данных может сжать не всю таблицу, а только одну секцию, содержащую данные за 2012 г. Как часть операции сжатия, эта секция может быть также перенесена на менее дорогой уровень хранения, что еще больше снизит совокупную стоимость владения хранимыми данными. В Oracle Database 12c эта операция обслуживания секции может быть полностью выполнена в режиме онлайн, благодаря чему во время выполнения операции возможны как запросы, так и операции DML.

Oracle Database 12c также обеспечивает возможность выполнять секционное обслуживание сразу для нескольких секций в виде единой атомарной операции. Например, можно слить три секции месяцев: January 2012, February 2012 и March 2012 — в единую секцию первого квартала Q1 2012, выполнив всего одну операцию слияния секций (merge partition).

Еще одно типичное использование секционирования для повышения управляемости — это поддержка процесса загрузки со «скользящим окном» (rolling window) в хранилище данных. Предположим, АБД ежедневно загружает новые данные в таблицу. Эта таблица секционирована по диапазонам значений ключа таким образом, что каждая секция содержит данные только за один день. Процесс загрузки — это просто добавление новой секции. Добавление одной секции гораздо более эффективно, чем изменение всей таблицы, так как администратору базы данных не требуется изменять другие секции.

Удаление данных очень экономичным и элегантным способом — это еще одно преимущество секционирования. Например, чтобы очистить данные секционированной таблицы, достаточно выполнить операцию drop или truncate для одной или нескольких секций. Это очень дешевая и быстрая операция словаря данных по сравнению с эквивалентной командой delete, использующей массу ресурсов и затрагивающей все строки. Распространенный способ удаления данных с помощью такой операции секционного обслуживания, как drop или truncate, оптимизирован в Oracle Database 12c: эти команды не требуют немедленных операций обслуживания индексов, чтобы они оставались действующими, т. е. это быстрые операции над метаданными.

Секционирование для доступности

Секционированные объекты базы данных обеспечивают независимость секций. Эта характеристика независимости секций может быть важной частью стратегии высокой доступности. Например, если одна секция секционированной таблицы недоступна, все остальные секции этой таблицы остаются доступными. Приложение может по-прежнему выполнять запросы и транзакции с этой секционированной таблицей, и эти операции базы данных будут выполняться успешно, если им не нужно обращаться к недоступной секции.

Администратор базы данных может задать хранение каждой секции в отдельном табличном пространстве. Это даст администратору возможность выполнять операции резервного копирования и восстановления на отдельной секции или наборе секций (благодаря привязке секций к табличным пространствам), независимо от других секций таблицы. Поэтому в случае аварии можно сначала восстановить базу данных только с секциями, содержащими активные данные, а позже в удобное время восстановить неактивные данные в других секциях. Это сокращает время простоя системы.

Наиболее важные данные становятся доступными снова в кратчайшие сроки, независимо от размера всей базы данных.

Секционирование может также уменьшить плановые простои. Преимуществом секционирования с точки зрения производительности является то, что администратор может выполнять операции обслуживания на больших объектах базы данных за относительно короткое время.

Преимущества секционирования для управления жизненным циклом информации

Задача сегодняшнего дня для обработки огромных объемов данных с наименьшими расходами может быть решена с помощью Oracle Partitioning, если использовать Automatic Data Optimization (автоматическая оптимизация данных) и Heat Map (температурная карта) — новые функции опции Advanced Compression для СУБД Oracle Database 12c. Независимость отдельных секций в сочетании с эффективными и прозрачными операциями обслуживания для секций — это основное условие для успешной стратегии «многоуровневой архивации» в части, касающейся онлайн-доступа. Если говорить конкретнее, то в таблицах, содержащих исторические данные, важность данных и схема доступа к ним сильно зависят от возраста этих данных.

Секционирование дает возможность хранить отдельные секции (или группы секций) на разных уровнях хранения с разными физическими атрибутами (такими как компрессия) и разной стоимостью. Например, в таблице заказов, содержащей сведения о заказах за 2 года, можно хранить данные последнего квартала на дорогостоящем уровне хранения высшего класса, а остальную часть таблицы (почти 90 % данных) — на недорогом уровне с низкими затратами.

Добавление средства автоматической оптимизации данных (Automatic Data Optimization, ADO) дает возможность задавать политики, указывающие, когда конкретную секцию нужно перемещать на другой уровень хранения или сжатия в зависимости от данных, автоматически собранных модулем Heat Map (температурная карта). Политики автоматической оптимизации данных (ADO) оцениваются и выполняются СУБД Oracle Database без участия человека. Таким образом, можно использовать распределение по уровням хранения и сжатия для экономии средств и повышения производительности, не создавая сложные скрипты и задания.

Стратегии секционирования

Oracle Database 12c предоставляет самый полный набор стратегий секционирования, и администратор может оптимально разделить данные с учетом реальных требований бизнеса. Все доступные стратегии секционирования опираются на **основные методы распределения данных**, которые могут использоваться для таблиц с простым (одноуровневым) или композитным (двухуровневым) секционированием. Oracle также предоставляет различные **расширения секционирования**, обеспечивающие больше гибкости для выбора ключа секционирования, автоматического создания секций по мере необходимости, использования одних и тех же стратегий секционирования для групп логически связанных таблиц (через отношения «родитель-потомок») и выработки рекомендаций по стратегиям секционирования для несекционированных объектов.

Методы распределения данных для секционированных объектов

Oracle Partitioning предлагает три основных метода распределения данных, которые определяют размещение данных в разных секциях. Это следующие методы.

- » **Range (Диапазон).** Данные распределяются на основе диапазона значений ключа секционирования (если в качестве ключа секционирования используется столбец даты, секция January-2013 содержит строки со значениями ключа секционирования в диапазоне от 01-JAN-2013 до 31-JAN-2013). Распределение по диапазону ключей непрерывно без каких-либо промежутков. Для определения диапазона всегда указывается верхняя граница секции, исключая само указанное значение, а нижняя граница определяется автоматически по верхней границе предыдущей секции. Значения границ секций всегда увеличиваются. Как следствие, первая секция таблицы (с самой низкой границей диапазона) всегда открыта в сторону меньших значений. Последнюю секцию (с самым высоким значением границы секции) при необходимости также можно сделать открытой в сторону больших значений.

- » **List (Список).** Распределение данных определяется списком отдельных значений ключа секционирования (например, если в качестве ключа секционирования используется столбец региона, то секция North America может включать значения Canada, USA и Mexico). Можно задать секцию DEFAULT (ПО УМОЛЧАНИЮ) для всех значений ключа секционирования, не вошедших явно ни в один список.
- » **Hash (Хеш).** Секция для данного ключа секционирования определяется путем применения внутреннего алгоритма хеширования к значению ключа секционирования. В отличие от двух других методов секционирования, хеш не обеспечивает никакой логической связи между данными и секцией, но поддерживает примерно одинаковые размеры секций. Вы получаете наиболее сбалансированные размеры секций, если ключ секционирования содержит достаточное количество различных значений, и вы задаёте число секций, равное степени числа 2 (например, 4, 16, 64).

Эти три основных метода распределения данных (диапазон, список и хеш) дают возможность задавать таблицу с простым или композитным секционированием.

Помимо основных методов распределения данных, Oracle предлагает **системное секционирование (System Partitioning)**: база данных только обеспечивает структуру для секционирования таблицы, но не хранит никаких метаданных, определяющих размещение данных. Размещением данных управляет уровень приложений как для их вставки, так и для доступа к ним (если приложению необходимо использовать отсечение секций). Системное секционирование предоставляет интерфейс для разработки приложений с особыми требованиями к размещению данных или доступа к ним (например, индексы доменов) и поддерживает только простое (одноуровневое) секционирование. Определение ключа секционирования и управление эквивалентом ключа секционирования полностью остаются за приложением.

Простое (одноуровневое) секционирование

Таблица определяется путем указания одной из вышеуказанных методологий распределения данных, а в качестве ключа секционирования используются один или больше столбцов. Например, рассмотрим таблицу со столбцом number в качестве ключа секционирования и двумя секциями less_than_five_hundred (менее 500) и less_than_thousand (менее 1000), то секция less_than_thousand содержит строки, где верно следующее условие. $500 \leq \text{ключ секционирования} < 1000$. Секции таблицы или индекса с простым секционированием представляют собой отдельные физические сегменты в базе данных, где хранятся фактические данные объекта.

Вы можете указать методы секционирования Range (диапазон), List (список), Hash (хеш) и System (системное секционирование) для обычных (heap) таблиц и индекс-таблиц (index-organized таблиц). Хеш-кластеры можно секционировать только по диапазонам ключей (Range)³.

Композитное (двухуровневое) секционирование

Для определения таблицы с композитным секционированием используется комбинация двух методов распределения данных. Сначала таблица секционируется одним методом распределения данных. Затем каждая секция подразделяется на подсекции с использованием другого метода распределения данных. Например, если для таблицы задано композитное секционирование range-hash (диапазон-хеш), то такая таблица сначала секционируется по диапазонам ключей, а затем каждая секция делится на подсекции методом хеш-секционирования. Секции таблицы с композитным секционированием существуют только как метаданные и не предусматривают фактического хранения данных: подсекции конкретной секции таблицы или индекса с композитным секционированием представляют собой физические сегменты базы данных, где хранятся данные секции.

Доступные методы композитного секционирования — это range-hash (диапазон-хеш), range-list (диапазон-список), range-range (диапазон-диапазон), list-range (список-диапазон), list-list (список-список) и list-hash (список-хеш), а также hash-hash (хеш-хеш), hash-range (хеш-диапазон) и hash-list (хеш-список). Композитное секционирование поддерживается только для традиционных (heap) таблиц и не поддерживается для индекс-таблиц и хеш-кластеров.

³ Кластер — это объект схемы, состоящий из нескольких таблиц, данные которых хранятся вместе в его блоках данных. Если используется хеш-кластер, база данных хранит вместе строки с одним и тем же хеш-значением. Кластеры используются преимущественно в средах OLTP для сведения к минимуму операций ввода-вывода.

Глобально секционированные индексы могут секционироваться по диапазонам ключей или хеш-значениям. Композитное секционирование не поддерживается для глобально секционированных индексов.

Расширения секционирования

Oracle предоставляет расширения секционирования, которые расширяют возможности использования основных стратегий секционирования. Расширения секционирования делают секционированные объекты более управляемыми и дают больше гибкости при определении ключа секционирования одной таблицы и даже групп таблиц, логически связанных отношениями «родитель-потомок».


Интервальное секционирование

Интервальное секционирование (Interval Partitioning) расширяет возможности метода секционирования по диапазонам значений. Оно определяет эквивалентно секционированные диапазоны для любых будущих секций и использует для этого определение интервала как части метаданных таблицы. Таблица с интервальным секционированием может автоматически расти до максимального количества секций (1048575) без какого-либо участия пользователя, даже если секционированная таблица была сначала создана с одной-единственной секцией. Вместо того чтобы явно создавать будущие секции по диапазонам, Oracle автоматически формирует новые секции по мере необходимости. Новая секция создается, как только поступают первые данные для нее. Интервальное секционирование значительно повышает возможности управления секционированной таблицей. Например, таблица с интервальным секционированием может быть определена таким образом, что Oracle будет создавать новую секцию для каждого дня календарного года; тогда секция September, 19th 2013 (19 сентября 2013) будет создана автоматически, как только первая запись для этой даты будет вставлена в базу данных.

Доступные методы интервально секционированной таблицы — это Interval (интервал), Interval-List (интервал-список), Interval-Hash (интервал-хеш) и Interval-Range (интервал-диапазон). Oracle Database 12c также поддерживает совместное использование интервального секционирования (Interval Partitioning) и ссылочного секционирования (Reference Partitioning). Интервальное секционирование поддерживается только для традиционных таблиц и не поддерживается для индекс-таблиц и хеш-кластеров. Интервальное секционирование как стратегия подсекционирования для любого метода секционирования верхнего уровня (*-Interval) сейчас не поддерживается.

Ссылочное секционирование

Ссылочное секционирование (Reference Partitioning) позволяет разделить таблицу, используя существующие отношения «родитель-потомок». Отношение «первичный ключ — внешний ключ» используется для того, чтобы дочерняя таблица могла унаследовать стратегию секционирования родительской таблицы без хранения столбцов родительской таблицы, составляющих ключ секционирования, в дочерней таблице. Стратегия секционирования родительской и дочерней таблиц становится идентичной. Для каждой секции в родительской таблице существует одна-единственная секция в дочерней таблице, и стратегия секционирования дочерней таблицы определяется исключительно через отношение «первичный ключ — внешний ключ». Все дочерние записи с определенным значением первичного ключа хранятся в той же секции дочерней таблицы, соответствующей секции с родительской записью. Без ссылочного секционирования приходится дублировать все столбцы ключа секционирования родительской таблицы в дочерней таблице, если вы хотите использовать одинаковую стратегию секционирования. Ссылочное секционирование дает возможность использовать отношение «родитель-потомок» в естественном формате логической модели данных без дублирования столбцов ключа секционирования, что снижает количество ручной работы по денормализации и экономит пространство. Кроме того, если используется ссылочное секционирование, то дочерняя таблица прозрачно наследует от родительской операции обслуживания секций, которые меняют логическую форму таблицы. Соединения по секциям (partition-wise joins) включаются автоматически, если соединяются эквивалентные секции родительской и дочерней таблиц. В итоге данная операция ускоряется. Например, родительская таблица Orders секционирована по диапазонам значений столбца даты заказа, ее дочерняя таблица Order Items не содержит столбец даты заказа, но может быть секционирована



путем ссылки на таблицу заказов. Если таблица Orders секционирована по месяцам, то все позиции для заказов в марте 2013 г. (March 2013) будут храниться в одной и той же секции таблицы Order Items, секционирование которой эквивалентно секционированию (equi-partitioned) родительской таблицы Orders. Если секция April 2013 добавляется в таблицу Orders (явно или через интервальное секционирование), Oracle прозрачно добавляет эквивалентную секцию в таблицу Order Items.

Ссылочное секционирование поддерживается только для традиционных таблиц и не поддерживается для индекс-таблиц и хеш-кластеров. Oracle Database 12c поддерживает ссылочное секционирование в комбинации с секционированием по виртуальному столбцу и интервальным секционированием.

Секционирование по виртуальному столбцу

Секционирование по виртуальному столбцу дает возможность определять ключ секционирования с помощью выражения с одним или несколькими столбцами таблицы и хранить это выражение только в виде метаданных. Секционирование по виртуальному столбцу дает возможность более точно соответствовать бизнес-требованиям. Бизнес-атрибуты, не определенные явным образом в виде столбцов таблицы, могут использоваться для определения стратегии секционирования объекта. Столбцы таблицы нередко перегружены информацией. Например, 10-значный идентификатор счета может включать отделение банка (первые три цифры идентификатора). Если используется расширение секционирования по виртуальному столбцу, то таблица Accounts (Счета), содержащая столбец с account id (идентификатором счета), может быть расширена виртуальным (производным) столбцом Account_Branch (Отделения банка), который создается из трех первых символов столбца account id и становится ключом секционирования для этой таблицы.

Секционирование по виртуальному столбцу не поддерживается для индекс-таблиц. Oracle Database 12c поддерживает сочетание секционирования по виртуальному столбцу со ссылочным секционированием и интервальным секционированием.

Partition Advisor

Модуль SQL Access Advisor создает рекомендации по секционированию, помимо рекомендаций по индексам, материализованным представлениям и журналам материализованных представлений. Рекомендации, создаваемые SQL Access Advisor, показывают возможные выгоды для производительности, если им последовать. Созданный скрипт с рекомендациями можно либо выполнить вручную целиком или только команды отдельных рекомендаций, либо отправить в очередь заданий Oracle Enterprise Manager.

Компонент Partition Advisor, интегрированный в SQL Access Advisor, является частью опции базы данных Oracle Tuning Pack, которая требует дополнительного лицензирования.

Краткий обзор функциональных возможностей секционирования

В следующей таблице показаны все базовые методы секционирования, доступные в Oracle Database 12c.


БАЗОВЫЕ МЕТОДЫ СЕКЦИОНИРОВАНИЯ В ORACLE DATABASE 12C

Стратегия секционирования	Распределение данных	Пример использования для бизнеса
Секционирование по диапазонам ключей	Последовательные диапазоны значений.	Таблица заказов, секционированная по диапазонам order_date
Секционирование по списку значений ключа	Неупорядоченные списки значений.	Таблица Orders, секционированная по списку значений по стране
Хеш-секционирование	Внутренний алгоритм хеширования	Таблица заказов, секционированная по хешу идентификатора заказчика (customer_id)
Композитное секционирование	Комбинация двух из вышеуказанных базовых методов	Таблица заказов Orders секционирована по диапазонам дат заказа (order_date) и подсекционирована по хешированным идентификаторам заказчиков (customer_id).
• Диапазон-диапазон	Range (диапазон), List (список)	Таблица заказов Orders секционирована по диапазонам дат заказа (order_date) и подсекционирована по диапазонам дат отгрузки (shipment_date).
• Диапазон-список	и Hash (хеш)	Таблица заказов Orders секционирована по списку значений по стране и подсекционирована по диапазонам дат заказа.
• Диапазон-диапазон Список-список		Таблица заказов Orders секционирована по списку значений по стране и подсекционирована по хешу идентификатора заказчика.
• Список-диапазон		
• Список-хеш		
• Хеш-хеш		
• Хеш-список		
• Хеш-диапазон		

Базовые методы секционирования могут использоваться в сочетании со следующими расширениями секционирования. Все комбинации расширений секционирования поддерживаются для традиционных таблиц.

РАСШИРЕНИЯ СЕКЦИОНИРОВАНИЯ В ORACLE DATABASE 12C

Расширение секционирования	Описание	Пример использования для бизнеса
Интервальное секционирование	Расширение к секционированию по диапазону значений. Определяется интервалом, обеспечивающим диапазоны эквивалентной ширины.	Таблица заказов секционирована по датам заказа с предопределенным дневным интервалом начиная с 1 января 2013 г. (01-Jan-2013)
• Интервал	Все секции, за исключением первой, автоматически создаются по требованию, когда поступают соответствующие данные.	
• Интервал-диапазон		
• Интервал-список		
• Интервал-хеш		
Ссылочное секционирование	Секционирование дочерней таблицы наследуется от родительской через отношение «первичный ключ — внешний ключ». Ключи секционирования не хранятся в реальных столбцах дочерней таблицы.	Таблица заказов Orders (родительская) секционирована по диапазонам дат заказов и передает метод секционирования таблице позиций заказов order lines (дочерней). Столбец order_date присутствует только в родительской таблице заказов
Секционирование по виртуальному столбцу	Любые методы секционирования, где ключ секционирования основан на виртуальном столбце. Виртуальные столбцы не хранятся на диске и существуют только как метаданные.	Таблица заказов Orders имеет виртуальный столбец с кодом региона, состоящим из первых трех цифр номера счета заказчика. Таблица заказов Orders секционируется по списку регионов продаж.



Заключение

С момента представления СУБД Oracle 8.0 в 1997 г. компания Oracle расширяла функциональные возможности Oracle Partitioning с каждым новым выпуском, добавляя новые методы секционирования, упрощая масштабирование и улучшая управляемость и возможности обслуживания. Oracle Database 12c, как и предыдущие выпуски, предлагает расширенные стратегии композитного секционирования и значительно расширенные операции обслуживания секций.

Опция Oracle Partitioning подходит для всех. Секционирование может значительно улучшить управляемость, производительность и доступность практически любого приложения базы данных. Поскольку секционирование прозрачно для приложения, оно может быть легко реализовано для приложений любых типов, так как никакие дорогостоящие и трудоемкие изменения в них не требуются.


ORACLE®

СВЯЖИТЕСЬ С НАМИ

blogs.oracle.com/russiafacebook.com/oracle.russiatwitter.com/oracleRUoracle.com/ru

Корпорация Oracle, головной офис

500 Oracle Parkway

Redwood Shores, CA 94065, USA

Для запросов со всего мира

Тел.: +1-650-506-7000

Факс: +1-650-506-7200

Hardware and Software, Engineered to Work Together

© Oracle и аффилированные компании, 2014 г. Все права защищены. Данный документ предоставляется исключительно в информационных целях, и его содержание может меняться без уведомления. Документ может содержать ошибки, и на него не распространяются никакие гарантии или условия, выраженные устно или предусмотренные законодательством, включая подразумеваемые гарантии, условия товарной пригодности и соответствия определенной цели. Oracle не несет никакой ответственности в связи с данным документом. Документ также не создает никаких договорных обязательств ни прямо, ни косвенно. Воспроизведение или передача этого документа в любой форме, любым способом (электронным или физическим) и для любой цели возможны только с предварительного письменного разрешения Oracle.

Oracle и Java являются зарегистрированными товарными знаками корпорации Oracle и/или ее аффилированных компаний. Другие названия могут быть товарными знаками соответствующих владельцев.

Intel и Intel Xeon являются товарными знаками или зарегистрированными товарными знаками корпорации Intel. Все товарные знаки SPARC используются по лицензии и являются товарными знаками или зарегистрированными товарными знаками компании SPARC International, Inc. AMD, Opteron, логотипы AMD и AMD Opteron являются товарными знаками или зарегистрированными товарными знаками компании Advanced Micro Devices. UNIX является зарегистрированным товарным знаком The Open Group. 0914



Разрабатывая свои программы и продукцию, корпорация Oracle заботится об окружающей среде.