

Основные концепции

Оглавление

Типы объектов

Детальное описание типов

Схема

Таблица

Индекс

Констрейнт

Представление

Последовательность

Триггер

Процедура/функция

Пакет

DBLink

Типы объектов

Объект	Описание
Схема	Коллекция объектов логической структуры базы данных (таблицы, последовательности, представления, снимки, индексы, процедуры и функции, пакеты, синонимы, связи базы данных, триггеры, кластеры и др.).
Таблица	Структура, состоящая из множества неупорядоченных (как правило) горизонтальных строк (rows), каждая из которых содержит одинаковое количество вертикальных столбцов (columns). Пересечение отдельной строки и столбца называется полем (field), которое содержит специфическую информацию
Индекс	Механизм для улучшения быстродействия поиска данных. Индекс определяет столбцы которые могут быть использованы для

	эффективного поиска и сортировки в таблице
Констрейнт	Правило, которое ограничивает значения в БД. Oracle позволяет вам создать 6 типов констрейнтов и позволяет вам описать их двумя способами.
Представление	Виртуальная таблица, представляющая собой поименованный запрос (синоним к запросу), который будет представлен как подзапрос при использовании представления
Последовательность	Объект БД, используя который пользователи могут генерировать уникальные числа. Как правило используется для генерации PK. Когда сиквенс сгенерировал число, значение увеличивается в не зависимости от того был commit или rollback.
Триггер	Фрагмент PL/SQL, который будет выполнен БД Oracle когда указанное условие будет исполнено.
Процедура/функция	Набор PL/SQL операторов, которые вы можете вызывать по имени.
Пакет	Объект схемы, который группирует логически связанные PL/SQL типы, константы, подпрограммы (функции/процедуры). Обычно, пакет имеет две части, спецификацию и тело, хотя тело иногда не обязательно
DB Link	Объект схемы в одной БД, который позволяет получить доступ к объектам в другой БД. Другая БД не обязательно должна быть БД Oracle

Детальное описание типов

Схема

это набор объектов логической структуры базы данных (таблицы, последовательности, представления, снимки, индексы, процедуры и функции, пакеты, синонимы, связи базы данных, триггеры, кластеры и др.).

Схема ассоциируется с именем пользователя-владельца ее объектов и имеет такое же имя.

Пользователь имеет доступ ко всем объектам в своей схеме.

Доступ пользователя к объектам "чужой" схемы возможен при наличии соответствующих привилегий

[Ссылка на сайт Oracle](#)

Таблица

Реляционные базы данных хранят все данные в таблицах. Таблица это структура, состоящая из множества неупорядоченных (как правило) горизонтальных строк (rows), каждая из которых содержит одинаковое количество вертикальных столбцов (columns). Пересечение отдельной строки и столбца называется полем (field), которое содержит специфическую информацию. Многие принципы работы реляционной базы данных взяты из определений отношений (relations) между таблицами.

[Ссылка на сайт Oracle](#)

В Oracle имеется 9 типов основных таблиц:

- Традиционные (heap organized table)
- Индекс-таблицы (index organized table - IOT)
- Кластеризованные индекс-таблицы (index clustered table)
- Кластеризованные хеш-таблицы (hash clustered table)
- Отсортированные кластеризованные хеш-таблицы (sorted hash clustered table)
- Вложенные таблицы (nested table)
- Временные таблицы (temporary table)
- Объектные таблицы (object table)
- Внешние таблицы (external table)

Традиционные таблицы

Стандартные таблицы базы данных. Данные распределены подобно куче. При добавлении данных для них используется первое обнаруженное в сегменте и подходящее по размеру пространство. При удалении данных из таблицы пространство, которое они занимали

становится доступным для повторного использования последующими операторами INSERT и UPDATE. Куча – область пространства, которая используется до определенной степени произвольно

Индекс-таблицы

Эти таблицы хранятся в индексной структуре, что накладывает определенный физический порядок на сами строки. Если в традиционных таблицах данные размещаются там, где они могут поместиться (в первом попавшемся месте), то в индекс-таблицах данные сохраняются в сортированном порядке, в соответствии с первичным ключом.

Кластеризованные индекс таблицы

Кластеры – это группы, состоящие из одной или более таблиц, которые физически хранятся в одинаковых блоках базы данных, и все строки в которых разделяют общее значение ключа кластера и физически находятся близко друг к другу. Во-первых, множество таблиц могут храниться вместе с физической точки зрения, что удешевляет стоимость запросов, которые получают, в подавляющем большинстве случаев, данные из объединения этих таблиц. Во-вторых, она позволяет хранить рядом все данные, содержащие одинаковое значение ключа кластера. Данные кластеризуются вокруг этого значения ключа кластера, ключ создается с помощью индекса B-tree

Кластеризованные хеш-таблицы

Эти таблицы похожи на кластеризованные индекс-таблицы, но вместо того, чтобы использовать индекс B-Tree для определения местонахождения данных по ключу кластера, хеш-кластер хеширует ключ кластера, чтобы найти блок базы данных, в котором должны располагаться данные. В хеш-кластере данные – это (образно говоря) и есть индекс. Такие таблицы подходят для хранения данных, которые читаются часто с помощью применяемой к ключу операции сравнения типа «равно».

Отсортированные кластеризованные хеш-таблицы

Эти таблицы появились с 10-й версии и сочетают качества кластеризованных хеш таблиц и индекс-таблиц. Концепция: имеется некоторое ключевое значение, по которому должны хешироваться строки, и ряд связанных с этим ключом записей, которые должны поступать в отсортированном порядке и обрабатываться в таком же отсортированном порядке. В такой системе отсортированный хеш-кластер может оказаться наиболее подходящей структурой.

Вложенные таблицы

Эти таблицы – часть объектно-реляционных расширений Oracle. Они представляют собой просто генерируемые и обслуживаемые системой дочерние таблицы, состоящие в отношениях «родитель-дочерний». Вложенная таблицы в этом случае не автономная таблица. В строке таблицы появляется столбец, значение которого – таблица.

Временные таблицы

Эти таблицы хранят временные данные на протяжении транзакции или сессии (on commit delete rows/on commit preserve rows). При необходимости они выделяют временные сегменты из временного табличного пространства текущего пользователя. Каждый сеанс видит только свои экстенды, он никогда не будет видеть данные созданные любым другим сеансом.

Внешние таблицы

Данные этих таблиц не хранятся в самой базе данных, они располагаются за пределами БД в обычных файлах операционной системы. Внешние таблицы позволяют запрашивать файл, хранящийся вне БД так, будто бы он является обычной таблицей внутри БД. Они наиболее полезны в качестве средства помещения информации в БД (являются очень мощным инструментом загрузки данных). Позволяют загружать данные в БД (а не выгружать!).

Индекс

это механизм для улучшения быстродействия поиска данных. Индекс определяет столбцы которые могут быть использованы для эффективного поиска и сортировки в таблице.

Однако, стоит понимать, что если индексов слишком много, то пострадает производительность модификации данных (insert/update/delete/merge).

Когда индексов слишком мало – страдает производительность операций DML(select/insert/update/delete).

Задача – найти оптимальную пропорцию для производительности

[Ссылка](#)

Типы индексов:

- Индексы со структурой В-дерева
 - о Индекс-таблицы
 - о Кластерные индексы со структурой В-дерева
 - о Индексы упорядоченные по убыванию

о Индексы по реверсивным ключам

- Битовый индексы
- Битовые индексы соединений
- Функциональные индексы
- Индексы предметной области

B-tree индексы

В простонародии – обычные индексы. Это индексы со структурой в-дерева. Самые распространенные в OLTP-системах. Подобен двоичному дереву, B-Tree индекс предоставлять быстрый доступ по ключу к индивидуальной строке или диапазону строк, обычно требуя нескольких операций чтения для нахождения нужной строки.

“B” - это НЕ binary (двоичный), а balanced (сбалансированный)

Индекс-таблицы (IOT)

Это таблицы, хранящиеся в структуре B-Tree. В то время как строки данных в традиционной таблице сохраняются как в куче, данные в индекс-таблице сохраняются и сортируются по первичному ключу. С точки зрения приложения индекс-таблицы ведут себя как «обычные» таблицы, для доступа используется обычный SQL.

Кластерные индексы со структурой B-дерева (B-Tree cluster index)

Это небольшая вариация обычных B-Tree индексов. Используются для индексации кластерных ключей. Вместо ключа, указывающего на строку, эти индексы имеют кластерные ключи, указывающие на блок, который содержит строки, объединенные по кластерному ключу.

Индексы, упорядоченные по убыванию (desc index)

Это индексы, которые позволяют данным быть отсортированным по убыванию в отличие от порядка по возрастанию, принятого в индексной структуре.

Индексы по реверсивным ключам

Это B-tree индексы, в которых байты ключа обращены. Применяются для достижения более равномерного распределения элементов в индексах, которые наполняются в порядке возрастания.

Битовые индексы (bitmap index).

B-tree – это отношение «один к одному» между элементом индекса и строкой.

Тут как раз наоборот, используется битовая карта для указания на множество строк одновременно. Идеально для

данных с высокой повторяемостью (DWH), которые в основном доступны только для чтения. **Не используются в OLTP.**

Битовые индексы соединений (bitmap join index).

Средство денормализации данных в индексной структуре вместо таблицы. Это битмап индекс для соединения двух или более таблиц. Хранит результат джойна, т.о. сам джойн БД уже делать не надо. **Так же не используется в OLTP**

Функциональные индексы (function-based index).

Это индексы B-tree (или битовые) которые хранят вычисленный результат функции по значению столбца(ов). Это как индексу по виртуальному столбцу, который не хранится в БД. Используем в select-ах так как написано в индексе upper(name)=:p_name.

Индексы предметной области (application domain index).

Эти индексы вы строите и сохраняете самостоятельно, либо в СУБД, либо за ее пределами. Вы сообщаете оптимизатору, насколько индекс селективен, насколько дорого его использование, а оптимизатор на основе этой информации решает использовать его или нет

Констрейнт

Правило которое ограничивает значения в БД.

Oracle позволяет вам создать 5 типов констрейнтов и позволяет вам описать их двумя способами.

Великолепная шестерка:

- **NOT NULL constraint**
- **Unique constraint**
- **Primary key constraint** = NOT NULL + unique
- **Foreign key constraint** – требует, чтобы значения в одной таблице соответствовали значениям в другой таблице
- **Check constraint** – требует, чтобы значение в БД соответствовало указанному условию
- **REF constraint** позволяет вам в будущем описать связь между REF column и объектом на который он ссылается

Синтаксически два пути:

- Как часть описания колонки или атрибута, это называется **inline specification**

- Как часть описания таблицы – **out-of-line** specification

Ссылка

Представление

Виртуальная таблица, представляющая собой поименованный запрос (синоним к запросу), который будет представлен как подзапрос при использовании представления.

Таблицы на которых основана VIEW называются базовыми таблицами.

Допускают DML, но имеется масса ограничений.

Последовательность

Объект БД, используя который пользователи могут генерировать уникальные числа. Как правило, используется для генерации Primary Key.

Когда сиквенс сгенерировал число, значение увеличивается вне зависимости от того был commit или rollback.

Если два пользователя конкурентно увеличивают один и тот же сиквенс, то значения сиквенса, которые получают два пользователя могут иметь дыры.

Один пользователь никогда не получит число сгенерированное для другого пользователя.

Сиквенсы не привязаны к таблице, вы можете использовать один сиквенс для нескольких таблиц.

После создания вы получаете значение используя CURRVAL, NEXTVAL

Триггер

Фрагмент PL/SQL, который будет выполнен БД Oracle когда указанное условие будет исполнено.

Как только вы создали триггер он включается автоматически, при этом вы можете его перевести в DISABLE или ENABLE.

Вы можете использовать следующие моменты (что-то не придумал лучшего слова ☺) выполнения:

- Before – перед событием
- After – после события

- Instead of (только для VIEW) – вместо события. Да да, ВМЕСТО выполнения вашего update-а (к примеру) будет выполнен код триггера.

И следующие события:

- DML
- Ddl_event
- Database_event

[Ссылка](#)

Процедура/функция

Набор PL/SQL операторов, которые вы можете вызывать по имени.

Хранимые процедуры и функции имеют преимущества в разработке, целостности, безопасности, производительности и распределения памяти.

Отличие функции от процедуры только в наличии возвращаемого значения (аналоги int/void JAVA в описании метода).

Функции могут использоваться как часть SQL-выражения (здесь подразумевается, что вы не используете часть параметров, как выходные).

Deterministic – указывается в сигнатуре, чтобы показать, что функция возвращает одно и то же значение для одних и тех же значений входных аргументов.

Для использования в функциональных индексах указание этого слова обязательно.

[Ссылка](#) и [еще](#)

Пакет

Объект схемы, который группирует логически связанные PL/SQL типы, константы, подпрограммы (функции/процедуры). Обычно, пакет имеет две части, спецификацию и тело, хотя тельце иногда не обязательно.

Спецификация (spec) – это интерфейс для вашего приложения. Описывает типы, переменные, константы, исключения, курсоры, и подпрограммы доступные для использования вовне.

Тело (body) – реализация спецификации.

DBLink

Объект схемы в одной БД который позволяет получить доступ к объектам в другой БД. Другая БД не обязательно должна быть БД Oracle.

Правда для не Oracle БД нам (им?) нужен Oracle Гетерогенный сервис.

После создания линка мы можем использовать ссылки на таблицы, вьюхи, PL/SQL объекта в другой БД добавляя @dblink_name к таблицам, вьюхам, pl/sql объектам.

Select/insert/update/delete – все это доступно с удаленными объектами.

Линки можно использовать в select, комбинируя таблицы из одной БД и из