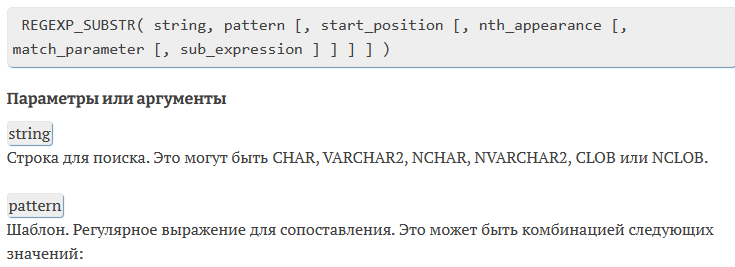
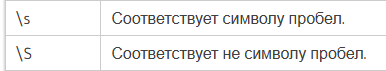
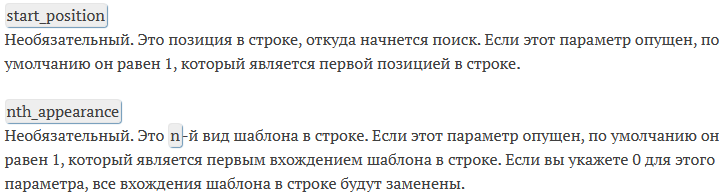
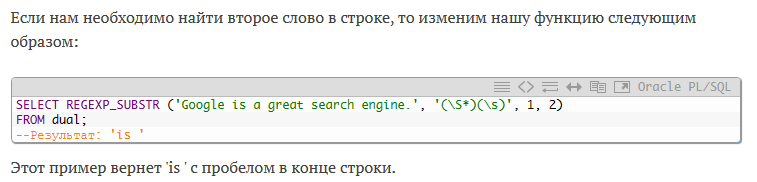
**12 лаба**









**13 лаба**

**3. Что такое локальные процедуры и функции PL/SQL?**

Локальным такой модуль называется из-за того, что он определяется только внутри родительского блока PL/SQL и не может быть вызван из другого блока, определенного вне родительского.

**4. Чем отличаются процедуры от функций?**

Функция – именованный модуль, который выполняет ноль или более выражений через фразу Return

Может быть вызвана следующим образом:

- В присвоении начального значения переменной

- В выражении присвоения

- В булевом выражении

- В SQL запросе

- Как аргумент в списке параметров другой функции или процедуры

Процедура – именованный модуль, который выполняет одно или несколько выражений и может принимать или возвращать значения через список параметров

**5. Что такое локальные процедуры и функции?**

Локальной функцией или процедурой называется подпрограмма, которая описана внутри другой подпрограммы.

Локальный программный модуль – это процедура или функция, определенная в секции декларации PL/SQL блока

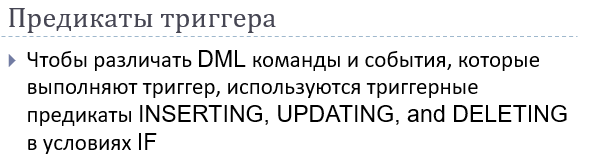
Объявление локальных процедур и функций должно размещаться в конце секции декларации после всех типов, записей, курсоров, переменных и исключений

Локальные процедуры и функции могут быть использованы только в рамках блока, в котором они объявлены

Локальные процедуры и функции могут быть перегружены

**15 лаба**

**Триггер –** особый вид процедур, которые срабатывают по запускающему их событию



**Перечислите типы триггеров, поддерживаемых Oracle.**

Классификация триггеров:

*По привязанному объекту:*

На таблице

На представлении - instead of trigger

*По событиям запуска:*

Вставка записей - insert

Обновление записей - update

Удаление записей - delete

*По области действия:*

Уровень оператора - statement level triggers

Уровень записи - row level triggers

Составные триггеры - compound triggers

*По времени срабатывания:*

Перед выполнением операции – before

После выполнения операции – after

**Можно ли выполнять TCL-операторы в триггерах Oracle. Если да, то при каких условиях?**

TCL–операторы нельзя выполнять в триггерах Oracle (исключение составляют триггеры, содержащие в себе автономные триггеры).

**Поясните правило: триггер является частью транзакции.**

Триггер – часть транзакции, ошибка в триггере откатывает операцию, изменения таблиц в триггере становятся частью транзакции.

Если откатывается транзакция, изменения триггера тоже откатываются

**Перечислите привилегии необходимые для работы с триггерами.**

*CREATE TRIGGER* - создавать, удалять, изменять в своей подсхеме

*CREATE ANY TRIGGER* - создать любой триггер в любой схеме, кроме SYS, не рекомендуется для словаря, не разрешает менять текст триггера

*ALTER ANY TRIGGER* - разрешать, запрещать, изменять, компилировать, любые, кроме SYS-триггеров, триггеры

*DROP ANY TRIGGER* - удалять любой триггер, кроме SYS-триггеров

*ADMINISTER DATABASE**TRIGGER* - создавать, изменять, удалять системные триггеры, должен иметь привилегию CREATE TRIGGER или CREATE ANY TRIGGER

**Перечислите события, на которые могут срабатывать DML-триггеры.**

|  |  |
| --- | --- |
| INSERT | Событие возникает, когда добавляется строка (строки) в таблицу или представление. |
| UPDATE | Событие возникает, когда выполняется операция UPDATE над данными в таблице или представлении.  Можно дополнительно задавать выражение OF для указания полей, при изменении которых срабатывает триггер. |
| DELETE | Событие возникает, когда удаляется строка (строки) из таблицы или представления.  Триггер не срабатывает при выполнении команды TRUNCATE table. |

**Объясните, что такое BEFORE-триггеры и AFTER-триггеры.**

AFTER (после события) – после записи в журнал,

BEFORE (до события) – до записи в журнал;

**В каких случаях нельзя обойтись только BEFORE-триггерами или только AFTER-триггерами?**

Если вы меняете запись, на которую действует триггер, используйте триггер BEFORE. Если вы выполняете какую-то сложную логику, которая может предотвратить изменение записи, используйте триггер BEFORE.

Почти во всем остальном используйте триггер AFTER. Примером может быть вставка дочерних записей, которые зависят от первичного ключа вставляемой записи. Например, если вы добавляете запись в таблицу истории для новой вставленной строки. Родительская строка не будет существовать в триггере BEFORE, поэтому проверка внешнего ключа не будет выполнена

**Перечислите уровни срабатывания триггеров.**

1) операторные BEFORE;

2) для каждой строки BEFORE;

3) выполняется оператор;

4) для каждой строки AFTER;

5) операторные AFTER.

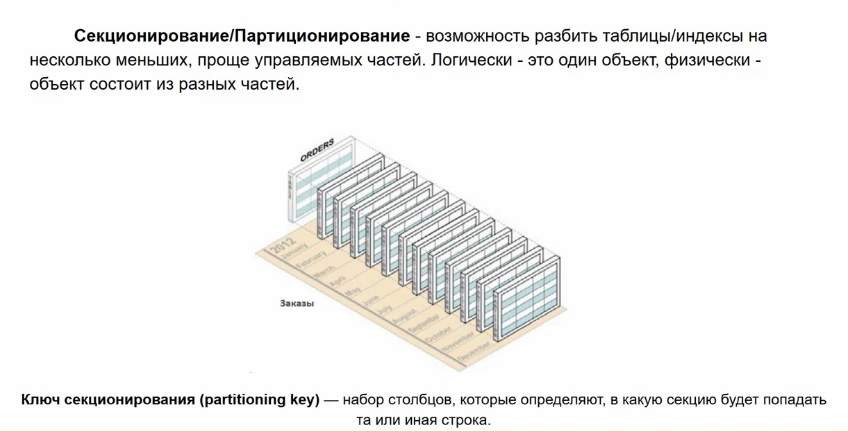
**Поясните принцип применения INSTEADOF-триггеров в Oracle.**

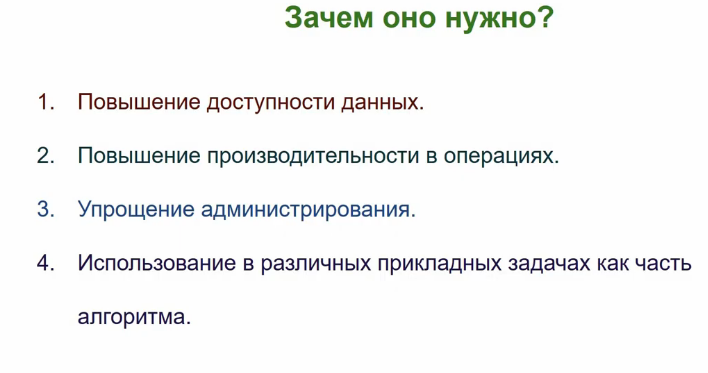
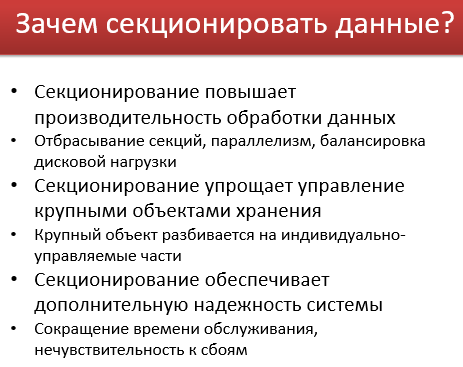
Триггеры INSTEAD OF предназначены в PL/SQL для выполнения операций вставки, обновления и удаления элементов представлений базы данных Oracle, но не таблиц. С их помощью можно сделать необновляемое представление обновляемым и изменить поведение обновляемого пред­ставления по умолчанию.

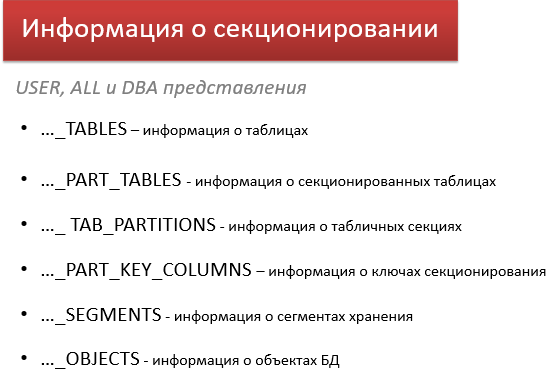
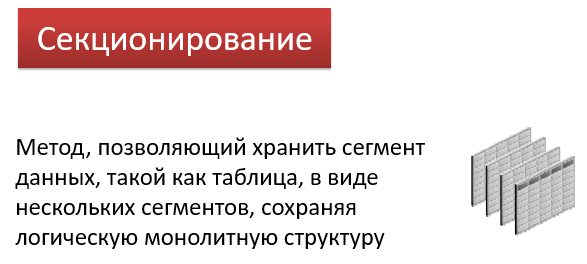
**Что такое мутирование таблиц?**

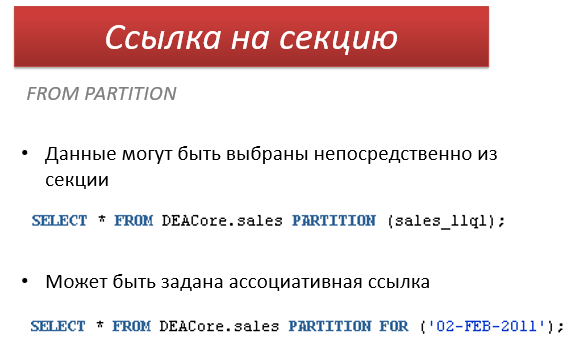
Мутирование таблиц (ошибка ORA-04091) возникает, если в триггере уровня строки выполняется изменение или чтение данных из той же самой таблицы, для которой данный триггер должен был сработать.

**16 лаба**

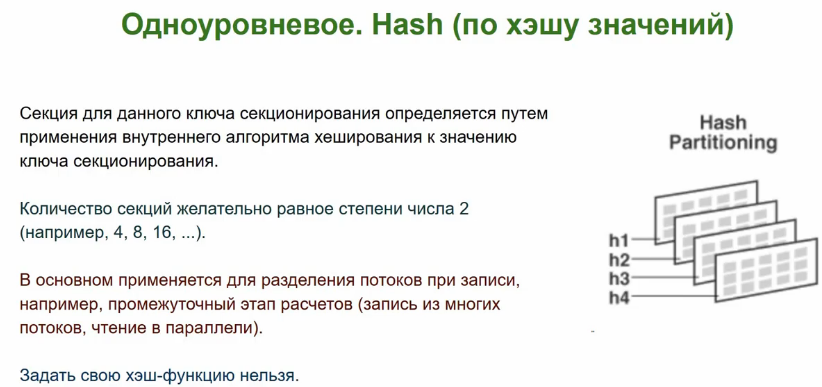
****

****









**1. Что такое секционирование таблиц?**

Метод, позволяющий хранить такой сегмент данных, как таблица, в виде нескольких сегментов, сохраняя логическую монолитную структуру

**2. В каких случаях целесообразно применять секционирование?**

1) Для повышения производительности работы SQL-запросов и DML-операций по модификации строк таблицы. Это достигается за счет того, что поиск и модификация строк в таблице идут не по всей таблице, а только в ее части (в одной или нескольких секциях).

2) Быстрое удаление значительного числа строк в больших таблицах за счет выполнения операции truncate секций.

**3. Объясните принцип секционирования для всех типов секционирования, которые использовались в заданиях лабораторной работы.**

1. Диапазонное секционирование используется для данных, которые разделяются на диапазоны на основе некоторого критерия. Наилучший результат от диапазонного секционирования можно получить, если данные распределены равномерно по создаваемым диапазонам. Диапазоны могут быть установлены на основе номера последовательности или номера раздела, но техника диапазонного секционирования обычно основана на времени (например, на ежеквартальных или ежемесячных данных).

2. Интервальное секционирование — это расширение традиционного метода секционирования по диапазону ключей. Чтобы реализовать интервальное секционирование, сначала потребуется специфицировать минимум один диапазонный раздел таблицы. Используете вы минимальный однодиапазонный раздел или многодиапазонные разделы, максимальное значение ключа диапазонного секционирования называется точкой перехода (transition point). После того, как данные пересекают точку перехода, база данных автоматически создает интервальные разделы.

**4. Перечислите названия типов секционирования, которые не использовались в заданиях лабораторной работы.**

Композитное секционирование

Секционирование по ссылке

**5. Объясните действие оператора ALTER TABLE MERGE.**

**Соединяет секции**

**6. Объясните действие оператора ALTER TABLE SPLIT.**

**Рфзделяет на две секции**

**7. Объясните действие оператора ALTER TABLE EXCHANGE**

Обмен данных секций с таблицей