-------------Лабораторная 11

**Курсор** — объект БД, который позволяет приложениям работать с записями построчно

**Поддерживаются два типа курсоров:**

**явный** — объявляется разработчиком

**неявный** — не требует объявления

Курсор может возвращать одну строку, несколько строк или ни одной строки

**Операторы управления явным курсором**

**DECLARE** — выполняет объявление явного курсора.

**OPEN** — открывает курсор, создавая новый результирующий набор на базе указанного запроса.

**FETCH** — выполняет последовательное извлечение строк из результирующего набора от начала до конца.

**CLOSE** — закрывает курсор и освобождает занимаемые им ресурсы.

**Неявный курсор**

* Не требуют объявления
* Не требуют OPEN, FETCH, CLOSE

**%TYPE** для переменной предоставляет тип данных столбца базы данных. Это особенно полезно при объявлении переменных, которые будут содержать значения столбцов таблиц базы данных.

**LOOP** используется тогда, когда вы не уверены сколько раз вы хотите выполнить тело цикла или выполнить тело цикла, по крайней мере, один раз.

**%ROWTYPE** предоставляет тип записи, представляющий строку в таблице (или представлении) базы данных Oracle.

**WHILE LOOP**, будет выполняться до тех пор, пока условие истинно.

**FOR LOOP** позволяет выполнить код повторно в течение фиксированного количества раз

**Курсорная переменная** ссылается на курсор. С помощью же курсорной переменной можно выполнить любой запрос и даже несколько разных запросов в одной программе.

**Курсорные подзапросы или вложенные курсорные выражения** – механизм встраивания курсоров в операторы SQL.

Если вы планируете обновлять или удалять записи, на которые ссылается оператор **SELECT FOR UPDATE**, вы можете использовать оператор **WHERE CURRENT OF**

**ROWID** — идентификатор строки (ROW IDentifier), а точнее, двоичное значение, однозначно идентифицирующее строку данных в таблице Oracle, даже если таблица не имеет уникального ключа.

# --1. Что такое *точная* и *неточная* выборки?

# 2. Объясните действие конструкций *WHEN OTHERS, WHEN TO\_MANY\_ROWS, WHEN NO\_DATA\_FOUND* в секции исключения.

**WHEN OTHERS** перехват всех остальных исключительных ситуаций, которые не были обработаны встроенными исключительными ситуациями или исключительными ситуациями, определенные программистом

**WHEN TO\_MANY\_ROWS** запрос вернул более одной строки

**WHEN NO\_DATA\_FOUND** неявный курсор не возвращает строк вообще

# 3. Объясните назначение функций *SQLERRM* и *SQLCODE.*

**SQLERRM** возвращает сообщение об ошибке, связанной с исключительной ситуацией

**SQLCODE** возвращает номер ошибки, связанной с исключительной ситуацией

# 4. Что такое *атрибут курсора*? Перечислите все атрибуты курсора и объясните их назначение.

**Атрибут курсора** имеет форму %имя\_атрибута и добавляется к имени курсора или курсорной переменной. Это что-то вроде внутренней пере-менной Oracle, возвращающей информацию о состоянии курсора

**%ISOPEN** — возвращает значение TRUE, если курсор открыт

**%FOUND** — определяет, найдена ли строка, удовлетворяющая условию

**%NOTFOUND** — возвращает TRUE, если строка не найдена

**%ROWCOUNT** — возвращает номер текущей строки

# 5. Объясните назначение конструкции *RETURNING* в операторах *INSERT, DELETE, UPDATE.*

**RETURNING** в выражениях INSERT, UPDATE, DELETE используется для получения данных, измененных соответствующим выражением

Позволяет избежать дополнительного SELECT для уточнения результатов

# 6. В чем отличие *явного* и *неявного* курсоров.

**явный** — объявляется разработчиком

**неявный** — не требует объявления(не требуют OPEN, FETCH, CLOSE)

# 7. Объясните схему работы с явным курсором в PL/SQL-блоке.

* Открытие явного курсора:
* OPEN cursor\_name [(argument [,argument ...])];
* Выборка из явного курсора:
* FETCH cursor\_name INTO record or variable\_list;
* Закрытие явного курсора:
* CLOSE cursor\_name;

# 8. Что происходит по команде *OPEN* курсора?

**OPEN** — открывает курсор, создавая новый результирующий набор на базе указанного запроса.

# 9. Объясните особенность применения *FOR-цикла* при работе с явным курсором.

**FOR UPDATE** используется для блокировки строк в результирующем наборе

**OPEN FOR, FETCH и CLOSE** - динамические многострочные запросы

# 10. Для чего применяются параметры курсора? Где они указываются?

Параметризация курсоров помогает повысить степень их повторного использования.

# 11. Что такое *курсорная переменная*? Как ее можно объявить?

**Курсорные переменные** - это структуры данных, которые указывают на курсорный объект

Используются для:

* Передачи курсора в качестве параметра,
* Чтобы отложить связь курсора с SELECT-запросом до выполнения команды OPEN

# 12. Что значит *курсорный подзапрос*?

Курсорные подзапросы или вложенные курсорные выражения – механизм встраивания курсоров в операторы SQL.

# 13. Объясните назначение конструкции *CURRENT OF*.

Оператор **WHERE CURRENT OF** позволяет обновить или удалить запись, которая была в курсоре последней

# 14. Для чего применяется псевдостолбец *ROWID*? Поясните его структуру.

**ROWID** — это псевдостолбец, который является уникальным идентификатором строки в таблице и фактически описывает точное физическое расположение данной конкретной строки.

# 15. Для чего применяется псевдостолбец *ROWNUM*?

**ROWNUM** - логический номер записи в запросе

**Rownum** – это псевдостолбец, значения которого можно увидеть, включив его в любой результирующий набор, например в список столбцов оператора SELECT. Значениям столбца ROWNUM присваиваются номера 1, 2, 3, 4, ... N, где N – число строк результирующего набора запроса.

-------------Лабораторная 12

**INSTR возвращает n-е вхождение подстроки в строке.**

INSTR( string, substring [, start\_position [, nth\_appearance ] ] )

* **string** является строка для поиска.
* **substring** подстрока для поиска в строке.
* **start\_position** символ в строке, с которого начнется поиск. Если аргумент опущен, то по умолчанию 1.
* **nth\_appearance** является n-м вхождением подстроки. Если опущен, то по умолчанию 1.

**SUBSTR позволяет извлекать подстроку из строки**

SUBSTR( string, start\_position, [ length ] )

* **string** исходная строка.
* **start\_position** позиция для начала извлечения подстроки. Первая позиция в строке всегда равна 1.
* **length** количество символов для извлечения.

**TO\_CHAR преобразует число или дату в строку**

TO\_CHAR( value, [ format\_mask ], [ nls\_language ] )

* value может принимать числовое значение или дату, которые будут преобразованы в строку.
* format\_mask формат, который будет использоваться для преобразования значения в строку.

**EXTRACT извлекает значение из даты или значения интервала**

**TO\_NUMBER преобразует строку в число**

TO\_NUMBER( string1, [ format\_mask ], [ nls\_language ] )

* **string1** строка, которая будет преобразована в число.
* **format\_mask** формат, который будет использоваться для преобразования string1 в число.

**MOD возвращает остаток от деления m на n**

**FLOOR возвращает наибольшее целое значение, равное или меньшее, чем число.**

- Числовые функции

- Символьные функции

- Функции по работе с датами

- Конвертирование

- Функции обработки ошибок

**Представление (VIEW**) - объект данных который не содержит никаких данных его владельца. Это - тип таблицы, чье содержание выбирается из других таблиц с помощью выполнения запроса.

**Запись** – структура данных, составленная из нескольких частей информации, называемых полями

Для объявления записи вначале надо определить как тип, а потом объявить переменную типа «запись»

**Типы записей:**

* Табличные
* Курсорные
* Программно-определенные

Record представляет собой группу связанных элементов данных, хранящихся в полях, каждая со своим именем и типом данных. Вы можете использовать Record как переменную, которая может содержать строку таблицы или некоторые столбцы (поля) из строки таблицы.

Атрибут [%ROWTYPE](https://oracleplsql.ru/rowtype-oracle-plsql.html) позволяет объявить запись, представляющую строку в таблице базы данных, без перечисления всех столбцов. Ваш код продолжает работать даже после добавления столбцов в таблицу. Если вы хотите представить подмножество столбцов в таблице или столбцы из разных таблиц, вы можете определить [VIEW](https://oracleplsql.ru/view.html) или объявить [CURSOR](https://oracleplsql.ru/cursors.html) для выбора правильных столбцов и любых необходимых объединений, а затем применить атрибут %ROWTYPE к VIEW или CURSOR.

**Запись (record)** представляет собой составную структуру данных; другими словами, запись состоит из нескольких полей, каждое из которых обладает собственным значением. Записи в [программах PL/SQL](https://oracle-patches.com/db/sql) очень похожи на строки в таблицах баз данных.

# 1. Объясните разницу между функциями floor(), ceil(), round() и trunc().

**FLOOR ()** — выполняет округление числа в меньшую к нулю сторону, кратно заданному значению точности.

**CEILING ()** — выполняет округление числа в большую сторону до ближайшего целого, кратно значению из второго параметра.

**ROUND ()** — производит математическое округление числа до указанной точности округления.

**ROUND** очень похожа на функцию TRUNC, только **TRUNC** — просто обрезает дробную часть, а ROUND именно округляет математически.

**TRUNC ()** — округляет число до целого, обрезая его дробную часть

# 2. Перечислите встроенные функции для работы с датами, исследованные в лабораторной работе.

**TO\_CHAR** преобразует число или дату в строку

[**EXTRACT**](https://oracleplsql.ru/extract-function.html) извлекает значение из даты или значения интервала.

**SYSDATE** возвращает текущую системную дату и время на вашей локальной базе данных.

**TO\_DATE** преобразует строку в дату.

# 3. Перечислите встроенные функции конвертации, исследованные в лабораторной работе.

**TO\_CHAR** преобразует число или дату в строку

**TO\_NUMBER** преобразует строку в число

# 4. Объясните назначение встроенных функций sqlcode и sqlerrm.

**SQLERRM** возвращает сообщение об ошибке, связанной с исключительной ситуацией

**SQLCODE** возвращает номер ошибки, связанной с исключительной ситуацией

-------------Лабораторная 13

**Процедура** — это подпрограмма, которая выполняет специфическое действие (CREATE PROCEDURE).

**Функция** — это подпрограмма, которая вычисляет значение (CREATE FUNCTION).

1. Процедура называется блоком PL / SQL, который выполняет одну или несколько задач. где функция называется блоком PL / SQL, который выполняет определенное действие.
2. Процедура может возвращать или не возвращать значение, когда функция должна возвращать одно значение.

**Пакеты** - коллекция PL/SQL объектов, сгруппированных вместе.

***Преимущества:***

Скрытие информации

Объектно-ориентированный дизайн

Постоянство объектов в транзакциях

Улучшенная производительность

**Спецификация пакета (package)** – обязательна, содержит список объектов для общего доступа из других модулей или приложения

**Реализация пакета (package body)** – содержит весь программный код для реализации процедур и функций и спецификации, приватные объекты и секцию инициализации

# 1. Что такое PL/SQL-тип данных запись?

# 2. Перечислите способы создания и принцип применения.

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

# 4. Чем отличаются процедуры от функций?

# 5. Что такое локальные процедуры и функции?

# 6. Что такое параметрическая, позиционная и смешанная формы передачи параметров?

# 7. Что значит «пакет сохраняет состояние переменных, объявленных в спецификации, на время сессии»?