Первое занятие SQL

Сортировка значений ORDER BY

Сотировка по возрастанию ASC

Сортировка по убыванию DESC

Ограничение числа записей LIMIT

Длинна столбца LENGTH – пишется после селекта

UPPER – приведение к верхнему регистру upper(строка) – пишется после селекта

LEFT – возвращение первых символов с строке left (строка, кол-во символов) – пишется после селекта

SPLIT\_PART – разбивает строку на несколько частей – SPLIT\_PART(строка, разделитель, номер части) – пишется после селекта

CAST - Изменение типа данных , запись (CAST( название колонки AS тип данных))

Второй вариант изменения это( название колонки::тип данных) – пишется после селекта

CONCAT – соединение в одну строку значения из нескольких столбцов (CONCAT(‘ ’, ‘ ’, ‘ ’))

DATE\_PART – извлечение части даты, времени, или др. DATE\_PART(‘year’,название столбца)

COALESCE (‘название колонки’, ‘навзывагние колонки)’

В этом нам поможет функция COALESCE, которая возвращает первое не NULL значение из списка поданных ей на вход аргументов.

Работу COALESCE можно описать следующим образом: она буквально читает список значений слева направо и, как только видит значение, которое не является NULL, сразу же возвращает его и прекращает чтение списка. Посмотрите внимательно на следующие примеры:

SELECT COALESCE(NULL, 25, 100, 150)

Результат:25

ROUND – округление вещественных чисел до знака, ROUND (число, сколько знаков)

CASE

WHEN <имя столбца> = <значение или текст> THEN 'значение или текст'

WHEN <имя столбца> = <значение или текст> THEN 'значение или текст'

ELSE 'значение или текст'

END

Когда в столбце значение правильно возвращается после then, если нет идет дальше до ELSE

Известные данные по первому занятию

SELECT -- перечисление полей результирующей таблицы

FROM -- указание источника данных

ORDER BY -- сортировка результирующей таблицы

LIMIT -- ограничение количества выводимых записей

Второе занятие SQL

WHERE - фильтр данных, пишется после select (where название столбца = значение )

Оператор LIKE не просто сравнивает строки на полное совпадение (или несовпадение), а проверяет их на соответствие заданному шаблону: если строка ему соответствует, то возвращается TRUE, в противном случае — FALSE. Как можно догадаться, конструкция NOT LIKE работает с точностью до наоборот.

Шаблоны могут содержать как обычные символы, так и символы-шаблоны: знак процента ( % ) и подчёркивание ( \_ ). Подчёркивание подменяет любой одиночный символ, а знак процента — любую (в том числе и пустую) последовательность символов: