

## 2. Язык SQL

### §2.1 Основные понятия

SQL - это язык, ориентированный специально на реляционные базы данных. Он устраняет много работы которую вы должны были бы сделать если бы вы использовали универсальный язык программирования, например С. Чтобы сформировать реляционную базу данных на С, вам необходимо было бы начать с самого начала. Вы должны были бы определить объект, называемый таблицей которая могла бы расти чтобы иметь любое число строк, а затем создавать постепенно процедуры для помещения значений в нее и извлечения из них. Если бы вы захотели найти некоторые определенные строки, вам необходимо было бы выполнить по шагам процедуру, подобную следующей :

1. Рассмотрите строку таблицы.
2. Выполните проверку - является ли эта строка одной из строк, которая вам нужна.
3. Если это так, сохраните ее где-нибудь, пока вся таблица не будет проверена.
4. Проверьте, имеются ли другие строки в таблице.
5. Если имеются, возвратитесь на шаг 1.
6. Если строк больше нет, вывести все значения, сохраненные в шаге 3.

(Конечно, это не фактический набор С команд, а только логика шагов которые должны были бы быть включены в реальную программу). SQL экономит вам все это. Команды в SQL могут работать со всеми группами таблиц как с единым объектом и могут обрабатывать любое количество информации извлеченной или полученной из их, в виде единого модуля.

### ТИПЫ ДАННЫХ В SQL

Типы данных SQL формата ANSI состоят из символов и различных типов чисел, которые могут классифицироваться как точные числа и приблизительные числа. Точные числовые типы - это номера, с десятичной точкой или без десятичной точки. Приблизительные числовые типы - это номера в показательной (экспоненциальной по основанию - 10) записи.

Для все прочих типов, отличия слишком малы чтобы их как-то классифицировать.

Иногда типы данных используют аргумент, который я называю размером аргумента, чей точный формат и значение меняется в зависимости от конкретного типа. Значения по умолчанию обеспечены для всех типов, если размер аргумента отсутствует.

### ===== ТИПЫ ANSI =====

Ниже представлены типы данных ANSI (имена в круглых скобках - это синонимы):

ТЕХТ ТЕКСТ

-----  
CHAR 1                      0Строка текста в реализационно-определенном формате. Размер аргумента здесь это единственное неотрицательное целое число, которое ссылается к максимальной длине строки. Значения этого типа,

должны быть заключены в одиночные кавычки, например 'text'. Две рядом стоящие одиночные кавычки (') внутри строки будет пониматься как одна одиночная кавычка (').

#### ПРИМЕЧАНИЕ:

Здесь и далее, фраза Реализационно-Определенный или Реализационно-Зависимый, указывает, что этот аргумент или формат зависит от конкретной программы в которой реализуются данные.

#### EXACT NUMERIC 1 ТОЧНОЕ ЧИСЛО

---

DEC  
(или DECIMAL)

Десятичное число; то есть, число которое может иметь десятичную точку. Здесь аргумент размера имеет две части: точность и масштаб. Масштаб не может превышать точность. Сначала указывается точность, разделительная запятая и далее аргумент масштаба. Точность указывает, сколько значащих цифр имеет число. Максимальное десятичное число составляющее номер - реализационно-определенное значение, равное или большее чем этот номер. Масштаб указывает максимальное число цифр справа от десятичной точки. Масштаб = нулю делает поле эквивалентом целого числа.

NUMERIC

Такое же как DECIMAL за исключением того, что максимальное десятичное не может превышать аргумента точности.

INT  
(или INTEGER)

Число без десятичной точки. Эквивалентно DECIMAL, но без цифр справа от десятичной точки, то-есть с масштабом равным 0. Аргумент размера не используется (он автоматически устанавливается в реализационно-зависимое значение).

SMALLINT

Такое же, как INTEGER, за исключением того, что, в зависимости от реализации, размер по умолчанию может (или не может) быть меньше, чем INTEGER.

#### APPROXIMATE NUMERIC ПРИБЛИЗИТЕЛЬНОЕ ЧИСЛО

---

FLOAT

Число с плавающей запятой на основе 10 показательной функции. Аргумент размера состоит из одного числа определяющего минимальную точность.

REAL

Такое же, как FLOAT, за исключением того, что никакого аргумента размера не используется. Точность установлена реализационно-зависимая по умолчанию.

DOUBLE  
PRECISION  
(или DOUBLE)

Такое же, как REAL, за исключением того, что реализационно-определяемая точность для DOUBLE PRECISION должна превышать реализационно-определяемую точность REAL.