abs(X)

Функция abs(X) возвращает абсолютное значение (модуль) числового аргумента X. Abs(X) возвращает NULL если X является значением NULL. Abs(X) возвращает 0.0 если X - это строка или blob которые не могут быть преобразованы в число. Если X равен числу -9223372036854775808 тогда abs(X) приводит к ошибке переполнения, так как нет эквивалентного положительного 64-битного двухкомпонентного значения.

cast(X as Y)

Функция позволяет преобразовать (конвертировать) данные X в данные другого типа Y. Более подробную информацию можно посмотреть в предыдущей статье.

changes()

Функция changes() function возвращает количество строк базы данных, которые были изменены, вставлены или удалены последним завершенным INSERT, DELETE, или UPDATE операторами, исключая операторы в низкоуровневых триггерах. Функция changes() это функция-обёртка sqlite3_changes() C/C++ функции и следовательно, следует тем же правилам подсчета изменений.

char(X1,X2,...,XN)

Функция char(X1,X2,...,XN) function возвращает строку, состоящую из символов, имеющих unicode значения с X1 до XN.

coalesce(X,Y,...)

Функция coalesce() возвращает копию первого не NULL аргумента или NULL если все аргументы являются NULL. Coalesce() должна использовать как минимум 2 аргумента.

glob(X,Y)

Функция glob(X,Y) эквивалентна выражению "Y GLOB X". Note that the X and Y arguments are reversed in the glob() function relative to the infix GLOB operator. If the sqlite3_create_function() interface is used to override the glob(X,Y) function with an alternative implementation then the GLOB operator will invoke the alternative implementation.

hex(X)

Функция hex() интерпретирует аргумент как BLOB и возвращает строку, которая содержит шестнадцатеричное представление содержимого аргумента в верхнем регистре.

ifnull(X,Y)

Функция ifnull() возвращает копию первого не NULL аргумента, или NULL если оба аргумента являются NULL. В Ifnull() должно передаваться именно 2 аргумента. Функция эквивалентна функции coalesce() с двумя аргументами.

instr(X,Y)

Функция instr(X,Y) находит первое вхождение строки Y в строке X и возвращает количество символов начиная с начала до найденной строки плюс 1, или 0 если Y не был найден в X. Иначе говоря функция **instr(X,Y) возвращает позицию подстроки в строке** начиная с 1. Если оба аргумента X и Y являются BLOB, тогда instr(X,Y) возвращает количество байтов с начала до найденного Y или 0 если Y не был найден в X. Если оба аргумента X и Y не NULL и не BLOB, тогда оба аргумента интерпретируются как строки. Если любой из аргументов NULL, тогда функция вернет NULL.

last_insert_rowid()

Функция last_insert_rowid() возвращает ROWID последней вставленной строки оператором INSERT в контексте соединения с базой данных для которого вызвана функция. Если в таблице используется первичный ключ INTEGER PRIMARY KEY AUTOINCREMENT то будет возвращено его значение, т.е. id последней добавленной записи. last_insert_rowid() это функция-обёртка для sqlite3_last_insert_rowid() C/C++ функции.

length(X)

Для строкового аргумента X, функция length(X) возвращает количество символов в аргументе X до первого NUL символа. Поскольку строки SQLite обычно не содержат NUL символы, функция length(X) обычно возвращает длину строки X. Для blob аргумента X, функция length(X) возвращает количество байт. Если X является NULL тогда функция возвращает NULL. Если аргумент X - число, тогда функция length(X) возвращает длину строкового представления X.

like(X,Y)
like(X,Y,Z)

Функция like() используется для реализации выражения" Y LIKE X [ESCAPE Z]". Если присутствует необязательное ESCAPE-предложение, то функция like() вызывается с тремя аргументами. В противном случае он вызывается только с двумя аргументами. Обратите внимание, что параметры X и Y реверсируются в функции like() относительно оператора infix LIKE. Интерфейс sqlite3_create_function() может использоваться для переопределения функции like () и, таким образом, изменения операции оператора LIKE. При переопределении функции like() может быть важно переопределить обе версии функции Like () с двумя и тремя аргументами. В противном случае может быть вызван другой код для реализации оператора LIKE в зависимости от того, было ли указано ESCAPE-предложение.

likelihood(X,Y)

Функция likelihood(X,Y) возвращает аргумент X без изменений. Значение Y в likelihood(X, Y) должно быть константой с плавающей точкой между 0.0 и 1.0 включительно. Функция likelihood(X) - это функция no-op, которую генератор кода оптимизирует, чтобы она не потребляла циклов ЦП во время выполнения (то есть во время вызовов sqlite3_step()). likelihood(x,Y) функция должна давать подсказку планировщику, что аргумент x является

логическим значением True с вероятностью приблизительно Y. unlikely(x) функция - краткая запись likelihood(x,0.0625). Функция likely(X) является краткой записью likelihood(X,0.9375).

likely(X)

Функция likely (X) возвращает аргумент X без изменений. Функция likely(X) - это функция no-op, которую генератор кода оптимизирует, чтобы она не потребляла циклов ЦП во время выполнения (то есть во время вызовов sqlite3_step()). Функция likely(X) предназначена для указания планировщику запросов, что аргумент X является логическим значением, которое обычно является true. Функция likely(x) эквивалентна likelihood(x,0.9375). См. также: unlikely(X).

load_extension(X)
load_extension(X,Y)

Функция load_extension(X,Y) загружает расширения SQLite из файла общей библиотеки с именем X, используя точку входа Y. Результат load_extension () всегда равен NULL. Если Y опущен, то используется имя точки входа по умолчанию. Функция load_extension () вызывает исключение, если расширение не удается правильно загрузить или инициализировать.

Функция load_extension() завершится ошибкой, если расширение попытается изменить или удалить функцию SQL или последовательность сортировки. Расширение может добавлять новые функции или последовательности сортировки, но не может изменять или удалять существующие функции или последовательности сортировки, поскольку эти функции и / или последовательности сортировки могут использоваться в другом месте в текущей выполняемой инструкции SQL. Чтобы загрузить расширение, которое изменяет или удаляет функции или последовательности сортировки, используйте API языка C sqlite3_load_extension ().

По соображениям безопасности, загрузка расширения по умолчанию отключена и должна быть включена в sqlite3_enable_load_extension().

lower(X)

Функция lower (X) возвращает копию строки X со всеми символами ASCII, преобразованными в Нижний регистр. Встроенная функция lower() по умолчанию работает только для символов ASCII. Чтобы выполнить преобразования регистра символов, не являющихся ASCII, загрузите расширение ICU.

ltrim(X)
ltrim(X,Y)

Функция ltrim(X,Y) возвращает строку, сформированную путем удаления всех символов Y, которые есть слева в X. Если аргумент Y опущен, ltrim(X) удаляет пробелы слева в X. Например, ltrim("asd","a") вернет строку "sd".

max(X,Y,...)

Функция max() с несколькими аргументами возвращает аргумент с максимальным значением или NULL если все аргументы равны NULL. Функция max() с несколькими аргументами ищет в своих аргументах слева направо в соответствии с аргументом, определяющим функцию сортировки и использует эту функцию сортировки для всех сравнений строк. Если ни один из аргументов max() не определяет функцию сортировки, то используется двоичная функция сортировки. Обратите внимание, что max() - это простая функция, когда она имеет 2 или более аргументов, но работает как статистическая функция, если задан только один аргумент.

min(X,Y,...)

Функция min() с несколькими аргументами возвращает аргумент с минимальным значением. Функция min() с несколькими аргументами ищет в своих аргументах слева направо в соответствии с аргументом, определяющим функцию сортировки и использует эту функцию сортировки для всех сравнений строк. Если ни один из аргументов min() не определяет функцию сортировки, то используется двоичная функция сортировки. Обратите внимание, что min() - это простая функция, когда она имеет 2 или более аргументов, но работает как статистическая функция, если задан только один аргумент.

nullif(X,Y)

Функция nullif(X,Y) возвращает свой первый аргумент, если аргументы отличаются, и NULL, если аргументы совпадают. Функция nullif(X,Y) выполняет поиск аргументов слева направо для аргумента, который определяет функцию сортировки и использует эту функцию сортировки для всех сравнений строк. Если ни один аргумент nullif() не определяет функции сортировки, тогда используется двоичная функция сортировки.

printf(FORMAT,...)

Функция printf(FORMAT,...) - sql-функция форматируемого вывода, которая работает так же как sqlite3_mprintf() и так же как printf() из стандартной C библиотеки. Первый аргумент - это строка формата, указывающая, как построить выходную строку, используя значения, взятые из последующих аргументов. Если аргумент FORMAT отсутствует или равен NULL, то результатом будет NULL. Строка формата %n игнорируется и не выводит аргумент. Строка формата %p является псевдонимом %X. Строка формата %z взаимозаменяема с %s. Если в списке аргументов слишком мало аргументов, предполагается, что отсутствующие аргументы имеют значение NULL, которые преобразуются в 0 или 0.0 для числовых форматов или пустую строку для %s.

Примеры использования функции:
select printf("%5.2f", 42.424242) - возвращает 42.42
select printf("%X", 255) - возвращает "FF"
select printf("Строка %s и символ %c, целое десятичное число %d","ASD","QWE",10.55) - возвращает "Строка ASD и символ Q, целое десятичное число 10"
select printf("%07d",5) - возвращает "0000005"

quote(X)

Функция quote(X) возвращает текст литерала SQL, который является значением его аргумента, подходящего для включения в инструкцию SQL. Строки окружены

одинарными кавычками с экранированием внутренних кавычек по мере необходимости. Большие двоичные объекты кодируются как шестнадцатеричные литералы. Строки со встроенными символами NUL не могут быть представлены в SQL как строковые литералы, поэтому возвращаемый строковый литерал усекается до первого NUL.

random()

Функция random() возвращает псевдо-случайное целое число между -9223372036854775808 и +9223372036854775807.

randomblob(N)

Функция randomblob(N) возвращает N псевдо-случайных байт. Если N меньше 1, тогда будет возвращен 1 псевдо-случайный байт.

Подсказка: программы могут генерировать уникальные идентификаторы используя функцию вместе с hex() и/или lower() например, так:
hex(randomblob(16))
lower(hex(randomblob(16)))

replace(X,Y,Z)

Функция replace (X,Y,Z) возвращает строку, образованную заменой в строке X всех вхождений Y на X. Если Y-пустая строка, то возвращается X без изменений. Если Z изначально не является строкой, тогда перед обработкой она приводится к строке UTF-8.

Пример:
select replace('asad','a','b') - возвращает "bsbd"

round(X)
round(X,Y)

Функция round (X,Y) возвращает значение с плавающей запятой X, округленное до Y цифр справа от десятичной запятой. Если аргумент Y опущен, предполагается, что он равен 0.

rtrim(X)
rtrim(X,Y)

Функция rtrim(X,Y) возвращает строку, сформированную путем удаления всех символов Y, которые есть справа в X. Если аргумент Y опущен, rtrim(X) удаляет пробелы справа в X.

soundex(X)

Функция soundex (X) возвращает строку, которая является кодировкой soundex строки X. строка "?000" возвращается, если аргумент имеет значение NULL или не содержит код

ASCII буквы. По умолчанию эта функция опущена из SQLite. Это функция доступна только если была использована опция SQLITE_SOUNDEX во время компиляции SQLite.

sqlite_compileoption_get(N)

Функция sqlite_compileoption_get() является обёрткой C/C++ функции sqlite3_compileoption_get(). Эта процедура возвращает N-й параметр опции, используемый для построения SQLite или NULL, если N находится вне диапазона. См. также compile_options pragma.

sqlite_compileoption_used(X)

Функция sqlite_compileoption_used() является обёрткой C/C++ функции sqlite3_compileoption_used(). Когда аргумент X функции sqlite_compileoption_used(X) - строка, являющаяся названием опции компиляции, возвращается значение true (1) или false (0) в зависимости от того была ли использована опция при компиляции SQLite.

sqlite_offset(X)

Функция sqlite_offset(X) возвращает байтовое смещение в файле базы данных для начала записи, из которой будет считано значение. Если X не является столбцом в обычной таблице, то sqlite_offset (X) возвращает NULL. Значение, возвращаемое параметром sqlite_offset (X), может ссылаться на исходную таблицу или индекс в зависимости от запроса. Если значение X обычно извлекается из индекса, то функция sqlite_offset(X) возвращает смещение соответствующей записи индекса. Если значение X извлекается из исходной таблицы, то функция sqlite_offset(X) возвращает смещение записи таблицы.

Функция sqlite_offset(X) доступна если SQLite был откомпилирован с опцией -DSQLITE_ENABLE_OFFSET_SQL_FUNC.

sqlite_source_id()

Функция sqlite_source_id() возвращает строку, определяющую конкретную версию исходного кода, которая использовалась для построения библиотеки SQLite. Строка, возвращаемая функцией sqlite_source_id(), является датой и временем возврата исходного кода, а затем хэшем SHA1 для этого возврата. Эта функция является обёрткой Sqlite3_sourceid().

sqlite_version()

Функция sqlite_version() возвращает строку версии для запущенной библиотеки SQLite. Эта функция является обёрткой функции sqlite3_libversion ().

substr(X,Y,Z)
substr(X,Y)

Функция substr (X,Y,Z) возвращает подстроку входной строки X, которая начинается с Y-го символа и имеет длину Z символов. Если Z опущен, то substr (X,Y) возвращает все символы до конца строки X, начиная с Y-го символа. Самый левый символ X имеет

позицию 1. Если Y отрицательно, то первый символ подстроки находится путем подсчета справа, а не слева. Если Z отрицательный, то возвращаются символы abs(Z), предшествующие Y-му символу. Если X является строкой, то индексы символов ссылаются на фактические символы UTF-8. Если x - это blob, тогда индексы относятся к байтам.

Примеры:
select substr("asdfgh",2,3) - возвращает "sdf"
select substr("asdfgh",-2) - возвращает "gh"
select substr("asdfgh",-2,-2) - возвращает "df"

total_changes()

Функция total_changes () возвращает количество изменений строк, вызванных инструкциями INSERT, UPDATE или DELETE с момента открытия текущего соединения с базой данных. Эта функция является обёрткой C/C++ функции sqlite3_total_changes().

trim(X)
trim(X,Y)

Функция trim(X,Y) возвращает строку, сформированную путем удаления всех символов Y, которые есть слева и справа в X. Если аргумент Y опущен, trim(X) удаляет пробелы слева и справа в X.

typeof(X)

Функция typeof(X) возвращает строку, указывающую тип данных выражения X: "null", "integer", "real", "text", или "blob".

unicode(X)

Функция unicode(X) возвращает числовую позицию unicode, соответствующую первому символу строки X. Если аргумент unicode (X) не является строкой, то результат не определен.

unlikely(X)

Функция unlikely(X) возвращает аргумент X без изменений. Функция unlikely(X) - это функция no-op, которую генератор кода оптимизирует, чтобы она не потребляла циклов ЦП во время выполнения (то есть во время вызовов sqlite3_step()). Функция unlikely(X) предназначена для предоставления планировщику запросов подсказки о том, что аргумент X является логическим значением, которое обычно не соответствует действительности. Функция unlikely(x) эквивалентна likelihood(X,0.0625).

upper(X)

Функция upper(X) возвращает копию входной строки X, в которой все символы ASCII нижнего регистра преобразуются в их эквивалент верхнего регистра.

zeroblob(N)

Функция zeroblob(N) возвращает объект, состоящий из N байт 0x00. SQLite использует zeroblobs очень эффективно. Zeroblobs можно использовать для резервирования пространства для большого двоичного объекта, который позже записывается с помощью инкрементного ввода-вывода большого двоичного объекта. Эта функция SQL реализована с помощью подпрограммы sqlite3_result_zeroblob () из интерфейса C/C++