СМОТРИ В БУДУЩЕЕ. ИНВЕСТИРУЙ В ЗНАНИЯ.

Работа с СУБД SQL

Типы данных.

Проверить, идет ли запись

Меня хорошо видно && слышно?

Знакомство

- настройка микрофона и аудио
- проверка работы чата

СМОТРИ В БУДУЩЕЕ. ИНВЕСТИРУЙ В ЗНАНИЯ.

Типы данных

Числовые типы данных

- **integer**: целые числа от -2147483648 до +2147483647 (4 байта)
- **smallint**: целые числа от -32768 до +32767 (2 байта)
- **bigint**: целые числа от -9223372036854775808 до +9223372036854775807 (8 байт)

- **serial**: целочисленный счётчик (автоинкремент) от 1 до 2147483647 (4 байта). Отсутствует в стандарте SQL
- **smallserial**: целочисленный счётчик (автоинкремент) от 1 до 32767 (2 байта)
- **bigserial**: целочисленный счётчик (автоинкремент) от 1 до 9223372036854775807 (8 байт)

- **numeric**: числа с фиксированной точностью - до 131072 знаков в целой части и до 16383 знаков после запятой.
  Описание: numeric(precision, scale). Параметр precision - максимальное количество цифр в числе. Параметр scale - максимальное количество цифр после запятой
- **decimal**: синоним numeric

- **real**: числа с плавающей точкой от 1E-37 до 1E+37 (4 байта)
- **double precision**: числа с плавающей точкой от 1E-307 до 1E+308 (8 байт)

- **money**: специальный тип для работы с денежными единицами, от -92233720368547758.08 до +92233720368547758.07, производный от numeric, с указанием денежной единицы (8 байт)

Типы даты/время

Наиболее удобный формат задания

- даты: yyyy mm dd 2020 01 08
- времени: hh:mi:ss 1:21:34

Типы:

- **timestamp** : дата и время без часового пояса от 4713 г. до н. э. до 294276 г. н. э., точность 1 мкс, 8 байт
- **timestamp** with time zone дата и время с часовым поясом, всё остальное как для предыдущего
- **date** дата (без времени суток), от 4713 г. до н. э. до 5874897 г. н. э., точность 1 день, 4 байта
- **time** : время суток (без даты), точность 1 мкс, 8 байт
- **time with time zone** : время суток (без даты) с часовым поясом (+ 1459), точность 1 мкс, 12 байт
- **interval** временной интервал от 178000000 лет до 178000000 лет, точность 1 мкс, 16 байт

Строковые типы данных

- **char (n)**: строка из фиксированного количества символов, n количество символов в строке
- **varchar(n)**: строка из произвольного количества символов, n максимальное количество символов в строке. Максимально возможный размер строки составляет около 1 ГБ
- **text** : текст произвольной длины. Не входит в стандарт SQL

Прочие типы данных

Логический тип:

- **boolean** : значения TRUE, FALSE, NULL:: boolean , размер 1 байт.

Геометрические типы:

- point, line, lseg , box, path, polygon, circle

Прочие типы:

- **json** : хранит данные json в текстовом виде
- **jsonb** : хранит данные json в бинарном формате (ускоряет обработку, поддерживает индексацию)
- **uuid** хранит (но не генерирует) универсальный уникальный идентификатор (UUID), например, a0eebc99 9c0b 4ef8 bb6d 6bb9bd380a11. 32 байта
- **xml** : хранит даные в формате XML

Двоичные типы:

- **bytea** : двоичная строка переменной длины . Входные данные по умолчанию принимаются в шестнадцатеричном формате: SELECT ' xDEADBE 01 отличается от стандарта SQL

Операции над строками

Для строковых типов определена операция – конкатенации, обозначаемая символом '+'. Например, 'abc'+'def' будет иметь результатом 'abcdef'.

Строки могут сравниваться. К операциям сравнения (применяемым не только к строкам) относятся:

- >
- <
- >=
- <=
- =
- <> или !=
- !<
- !>

Строковые функции

ASCII(s) - возвращает целый числовой код самого левого символа аргумента типа char или varchar

CHAR(n) - возвращает символ, имеющий код, равный аргументу

CHARINDEX(s1,s2[,n])

s1,s2 – строковые выражения. *n* – целое выражение. Функция возвращает номер символа, начиная с которого *s1* входит в *s2* (нумерация символов от 1). Поиск ведется в *s2*, начиная с символа *n*. Например, *charindex('код', 'крокодил')* возвратит значение 4. Если *s1* не входит в *s2*, то функция возвращает 0.

LEN(s) – возвращает длину строки *s*

Строковые функции

LOWER(s) – преобразует строку к нижнему регистру

UPPER(s) – преобразует строку к верхнему регистру

LTRIM(s) – возвращает строку, равную аргументу из которого удалены ведущие пробелы

RTRIM(s) – возвращает строку, равную аргументу из которого удалены хвостовые пробелы.

REPLACE(s1,s2, s3) – в *s1* отыскиваются все вхождения *s2* и заменяются на *s3*.

Аргументы могут иметь строковые или двоичные (binary) типы.

REPLICATE(s, n) – возвращает строку, равное первому аргументу, повторенную указанное вторым аргументом число раз.

Числовые типы данных

- *ABS(x)* – абсолютная величина числа
- *FLOOR(x)* – округление до ближайшего целого с недостатком
- *CEILING(x)* – округление до ближайшего целого с избытком
- *POWER(x,y)* – *x* в степени *y*
- *ROUND(x,n)* – округление *x* до *n* знаков после десятичной точки.
- *SIGN(x)* – возвращает –1 при $x<0$; 0 при $x=0$ и 1 при $x>0$
- *SQUARE(x)* – возвращает квадрат числа
- *SQRT(x)* – квадратный корень из *x*
- *Многие прочие не перечисляю*

Преобразования типов

Неявное преобразование

Когда два выражения разных типов участвуют как операнды в некоторой операции, то выражение младшего типа преобразуется к типу операнда старшего типа (с бóльшим номером), если это возможно.

Явное преобразование

Явное преобразование выполняется функциями *CAST* и *CONVERT*. Здесь мы рассмотрим только функцию *CAST*.

Синтаксис: *CAST(выражение as <имя типа>)*. Значение выражения преобразуется к указанному типу.
Пример:

```
CAST(23.765 as varchar(8))
```

СПАСИБО ЗА ВНИМАНИЕ!

#аис
#учисьваис