Типы данных

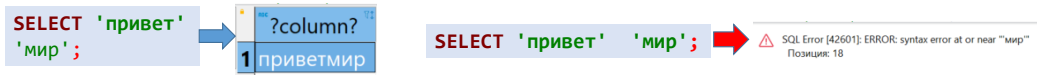
Типы данных

- Символьные
- Числовые
- Дата и время
- Логические
- Двоичные

Символьные данные

Символьные данные

- **varchar(n) , char(n) , text**
- Константные значения
  - Последовательность символов, заключённая в апострофы (') - 'PostgreSQL'
    - Две строковые константы, разделённые **пробельными символами** и **минимум одним переводом строки**, объединяются в одну

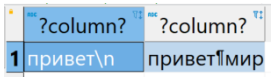


  - Строковая константы со спецпоследовательностями в стиле **C**
  - Строковые константы со спецпоследовательностями **Unicode**
  - Строковые константы, заключённые в доллары

Константы со спецпоследовательностями в стиле C

- Начинаются с буквы **E** (заглавной или строчной)

```
SELECT 'привет\n', E'привет\nмир';
```



Спецпоследовательность	Интерпретация
\b	Символ «забой»
\f	Разрыв страницы
\n	новая строка
\r	возврат каретки
\t	табуляция
\'	апостроф

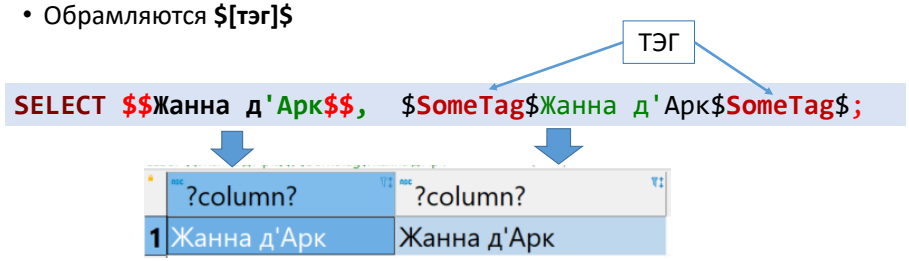
Строковые константы со спецпоследовательностями Unicode

- Позволяют включать в строки символы Unicode по их кодам
- Начинается с **U&** (строчная или заглавная **U** и амперсанд)
- Символы Unicode можно записывать двумя способами:
  - **** и код символа из **четырёх** шестнадцатеричных цифр (**\043B**)
  - **\+** и код символа **из шести** шестнадцатеричных цифр (**\+00043B**)

```
SELECT U&'\0441\043B\043E\043D';
SELECT U&'\+000441\+00043B\+00043E\+00043D';
```

Строковые константы, заключённые в доллары

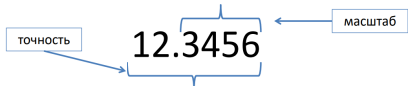
- Используются для работы со строками, содержащими много апострофов или обратных косых черт (\)
  - Позволяют избежать необходимости «зеркалирования»
  - Делают строки более читабельными
  - Обрамляются **$[тэг]$**



Числовые данные

Точные числовые данные

- Целочисленные типы – **smallint** (int2), **integer** (int4), **bigint** (int8)
  - **integer** обычно оптимален с точки зрения компромисса между диапазоном допустимых значений и затратами памяти
- Числа фиксированной точности – **numeric**(precision, scale) и **decimal** (precision, scale)
  - **Scale** (масштаб) – число значащих цифр, в дробной части числа
  - **Precision** (точность) – общее число цифр в числе
  - Могут хранить очень большое количество цифр: **131072** цифры — до десятичной точки, **16383** — после точки



Числовые данные с плавающей точкой

- **real**, **double precision** и **float**(p)
- Поддерживают специальные значения `'Infinity'`(бесконечность), `'-Infinity'`(отрицательная бесконечность) и `'NaN'` (не число)

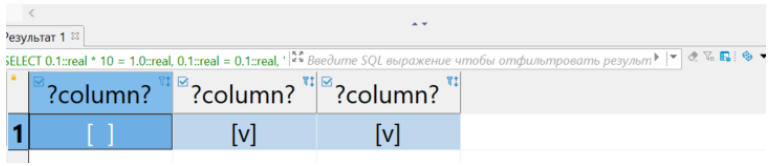
Тип данных	Диапазон значений	Точность
real	от $1E^{-37}$ до $1E^{+37}$	не меньше 6 десятичных цифр
double precision	от $1E^{-307}$ до 1^{E+308}	не меньше 15 десятичных цифр
float(p)	p = 1 до 24 →real; p = 25 до 53 →double precision	

  - Если точность вводимого числа выше допустимой - будет выполняться округление значения
  - При вводе слишком большого или очень маленького значения будет генерироваться ошибка

Числовые данные с плавающей точкой

- **ВНИМАНИЕ**!
  Сравнение двух чисел с плавающей точкой на предмет равенства их значений может привести к неожиданным результатам

```
SELECT  0.1::real * 10 = 1.0::real,
        0.1::real = 0.1::real,
        'Infinity'::real  > '-Infinity'::real;
```



Последовательные типы

- **serial** (int4)**, bigserial**(int8) **и smallserial**(int2)
- Реализованы как удобная замена целой группы SQL-команд:
  - Создание объекта **SEQUENCE** – генератор уникальных целых чисел
  - Получение чисел из последовательности с помощью функции **nextval**('имя_последовательности')
- Часто используются в качестве значений суррогатного первичного ключа (Primary Key)
  - Нет необходимости указывать явное значение для вставки в поле PK

Дата и время

Дата и время

- **date**, **time** и **time with time zone**
  - Даты обрабатываются в соответствии с григорианским календарем
  - **time** хранит время внутри суток
  - **time with time zone** хранит время с учетом смещения, соответствующего часовому поясу
  - При вводе значений их нужно заключать в одинарные кавычки, как и текстовые строки

```
SELECT  '22:25:35+01'::time with time zone,
        '22:25:35+02'::time with time zone,
        '22:25:35+03'::time with time zone;
```

Результат 1

	timetz	timetz	timetz
1	00:25:35	23:25:35	22:25:35

Форматы для ввода значений

Тип данных	Формат ввода	Пример
date	'yyyy-mm-dd' 'dd mmm, yyyy' 'mmm dd, yyyy'	'2022-06-15'::**date** '15 Jun, 2022'::**date** 'Jun 15, 2022'::**date** 'Jun 35, 2022'::**date - ERROR**
time	'hh:mm:[ss]' 'hh:mm:[ss] am' 'hh:mm:[ss] pm'	'22:15:16'::**time** '10:15:16 am'::**time** '10:15:16 pm'::**time** '25:15:68'::**time - ERROR**
time with time zone	'hh:mm:[ss]+tz' 'hh:mm:[ss] am +tz' 'hh:mm:[ss] pm +tz'	'10:25:35+01'::time with time zone '10:25:35 am +02'::time with time zone '10:25:35 pm +03'::time with time zone

Временная отметка (интегральный тип)

- **timestamp, timestamp with time zone** (timestamptz)
  - Получается в результате объединения типов даты и времени
  - Оба типа занимают 8 байтов
  - Значения типа **timestamptz** хранятся приведенными к нулевому часовому поясу (UTC), а перед выводом приводятся к часовому поясу пользователя

```
SELECT  '2022-09-21 22:25:35'::timestamptz,
        '2022-09-21 22:25:35'::timestamp;
```

Тип interval

- Представляет продолжительность отрезка времени между двумя моментами времени
- Формат: **quantity unit [quantity unit ...] direction**
  - **unit** – единица измерения (microsecond, millisecond, second, minute, hour, day, week, month, year, decade, century, millennium)
  - **quantity** – количество единиц измерения
  - **direction** – может принимать значение **ago** («тому назад») либо быть пустым

```
SELECT  '1 year 2 months ago'::interval,
        current_date,
        current_date - '1 year 2 months ago'::interval;
```

	interval	current_date	?column?
1	-1 years -2 mons	2022-06-30	2023-08-30 00:00:00

Тип interval – альтернативный формат

- Стандарт ISO 8601: **P[yyyy-mm-dd][Thh:mm:ss]**
  - **P** – обязательный символ в начале строки
  - **T** – разделяет дату и время

```
SELECT 'P0002-01-10T04:05:06'::interval,
       'P-00020110T04:05:06'::interval,
       'P-0002-01-03T04:05:06'::interval;
```

	interval	interval	interval
1	2 years 1 mon 10 days 04:05:06	-2 years -1 mons -10 days +04:05:06	-1 years -11 mons +3 days 04:05:06

Вычитание временных отметок

- Значения типа **interval** можно получить при вычитании одной временной отметки из другой:

```
SELECT  ('2022-09-16'::timestamp -
         '2022-05-01'::timestamp)::interval;
```

	interval
1	138 days

Логические и двоичные данные

Логический тип

- **boolean**
- Может иметь три состояния:
  - «**true**» - TRUE, 't', 'true', 'y', 'yes', 'on', '1'
  - «**false**» - FALSE, 'f', 'false', 'n', 'no', 'off', '0'
  - **NULL**
- Реализует трехзначную логику

```
SELECT  (5 = 5) = 'yes',
        (5 = 5) = 'no',
        null = 'yes',
        null = null;
```

Результат 1

SELECT (5 = 5) = 'yes', (5 = 5) = 'no', null = 'yes', null = null Введите SQL выражение чтобы отфильтро

	?column?	?column?	?column?	?column?
1	[v]	[ ]	[NULL]	[NULL]

Двоичные типы данных

- **bytea**
  - Позволяют хранить байты с кодом 0 и другими «непечатаемыми» значениями (значения вне десятичного диапазона 32..126)
  - В операциях с двоичными строками обрабатываются байты в чистом виде
- Поддерживает два формата ввода и вывода (параметр ***bytea_output***):
  - **hex** (шестнадцатеричный) - '**\x** коды символов в 16-ой системе'
  - **escape** (спецпоследовательностей) – '****коды символов в 8-ой системе'

```
SELECT  '\x48 45  4C  4C  4F  21  A9'::bytea as "_hex",
        '\110\105\114\114\117\041\251'::bytea as "_escape";
```

Результат 1

SELECT '\x48 45 4C 4C 4F 21 A9'::bytea as "_hex", '\110\105\1 Введите SQL выражение чтобы отфильтровать результаты

	_hex	_escape
1	HELLO!©	HELLO!©

http://webmasterschool.ru/supply/code.php