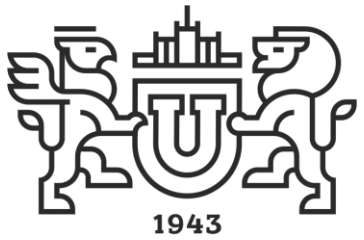


Южно-Уральский государственный университет

Национальный исследовательский университет

КРОК

SQL
PostgreSQL

КРОК
Челябинск, ул. Карла Маркса, д. 38

Смирнов Анатолий
Технический менеджер

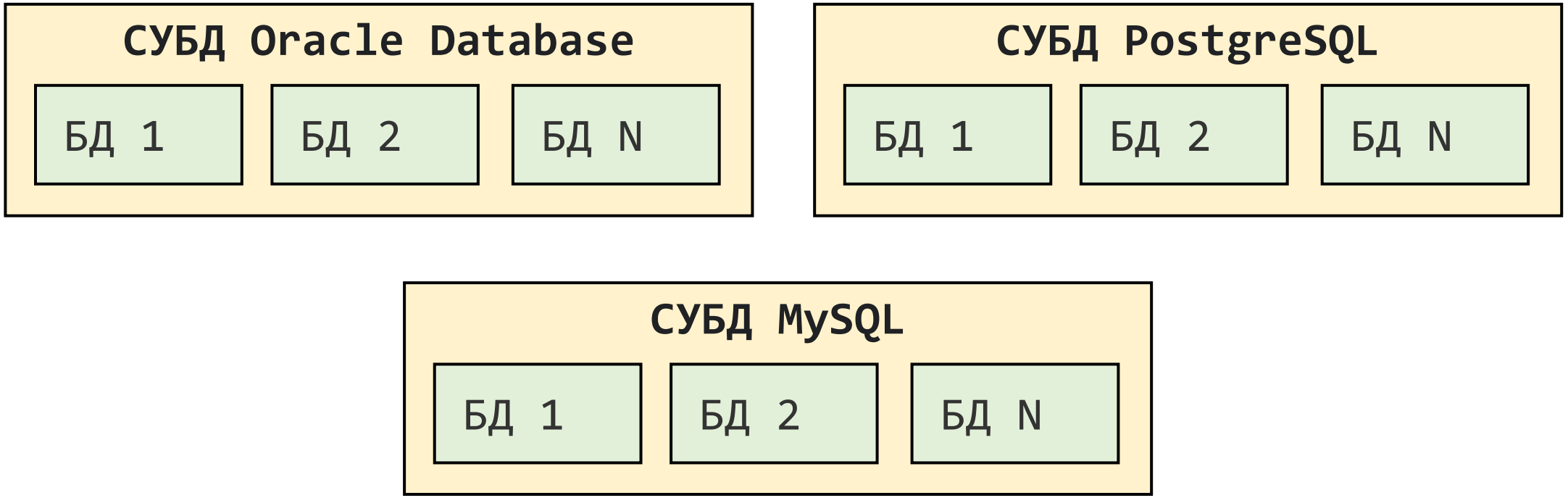
Кузнецов Сергей
Старший инженер-разработчик

Фоменко Алексей
Младший инженер-разработчик

СУБД и БД

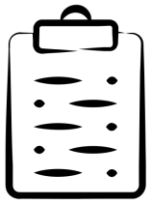
КРОК

СУБД – система управления базами данных



Типы данных

КРОК

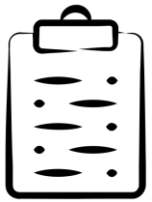


СУБД PostgreSQL поддерживает достаточно большое количество типов данных. Самые основные, которые чаще всего используются на практике:

- числовые типы
  - целочисленные типы
  - числа с произвольной точностью
  - типы с плавающей точкой
  - последовательные типы
- символьные типы
- типы даты/времени.

Целочисленные типы

КРОК

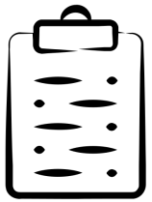


Наиболее используемым типом данных среди целочисленных является тип данных integer. Данный тип имеет достаточно широкий диапазон значений, которого хватает для большинства задач.

Имя	Размер	Описание	Диапазон
smallint	2 байта	целое в небольшом диапазоне	-32768 ... +32767
integer	4 байта	типичный выбор для целых чисел	-2147483648 ... +2147483647
bigint	8 байт	целое в большом диапазоне	-9223372036854775808 ... 9223372036854775807

Числа с произвольной точностью

КРОК

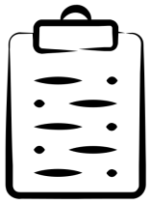


Числа с произвольной точностью используются, когда необходимо обеспечить точность вычислений, например, при операциях с денежными суммами. Однако, вычисления числами произвольной точности производятся медленнее, чем целыми числами или числами с плавающей точкой.

Имя	Размер	Описание	Диапазон
numeric	переменный	вещественное число с указанной точностью	до 131072 цифр до десятичной точки и до 16383 - после запятой

Числа с произвольной точностью

КРОК



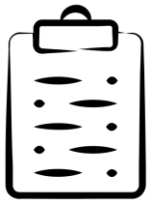
Синтаксис типа данных числа с произвольной точностью имеет вид:

```
NUMERIC(точность, масштаб)
```

- Точность - это общее количество разрядов в числе.
- Масштаб - количество разрядов в числе после запятой.
- Допускается не задавать масштаб числа с произвольной точностью. В этом случае масштаб по умолчанию будет равен нулю.
- Если не задать точность и масштаб, то они будут равны пределам, установленным в postgres.
- Пример. Допустим, мы хотим сохранить в базе данных число 1234.56, тогда мы должны использовать тип NUMERIC(6, 2).

Типы с плавающей точкой

КРОК

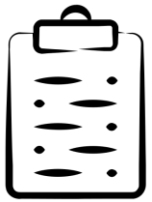


В PostgreSQL имеет два типа с плавающей точкой: real и double precision. Эти типы данных реализованы в соответствии со стандартом IEEE 754. Вычисления с использованием этих типов данных не совсем точные. Поэтому, если важна точность, то лучше использовать тип numeric

Имя	Размер	Описание	Диапазон
real	4 байта	вещественное число с переменной точностью	точность в пределах 6 десятичных цифр
double precision	8 байт	вещественное число с переменной точностью	точность в пределах 15 десятичных цифр

Последовательные типы

КРОК

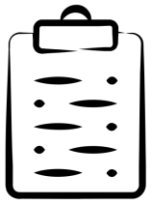


Последовательные типы данных представляют собой удобные средства для создания столбцов с уникальными значениями.

Имя	Размер	Описание	Диапазон
smallserial	2 байта	небольшое целое с автоувеличением	1 ... 32767
serial	4 байта	целое с автоувеличением	1 ... 2147483647
bigserial	8 байт	большое целое с автоувеличением	1 ... 9223372036854775807

Символьные типы данных

КРОК

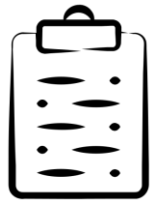


Существует три основных символьных типа данных в PostgreSQL. Это varchar(n), char(n) и text. n - это максимальное количество символов в строке переменной длины.

Имя	Описание
varchar(n), character varying(n)	строка ограниченной переменной длины
char(n)	строка фиксированной длины, дополненная пробелами
text	строка неограниченной переменной длины

Типы даты/времени

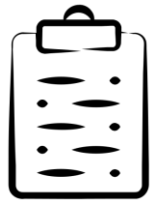
КРОК



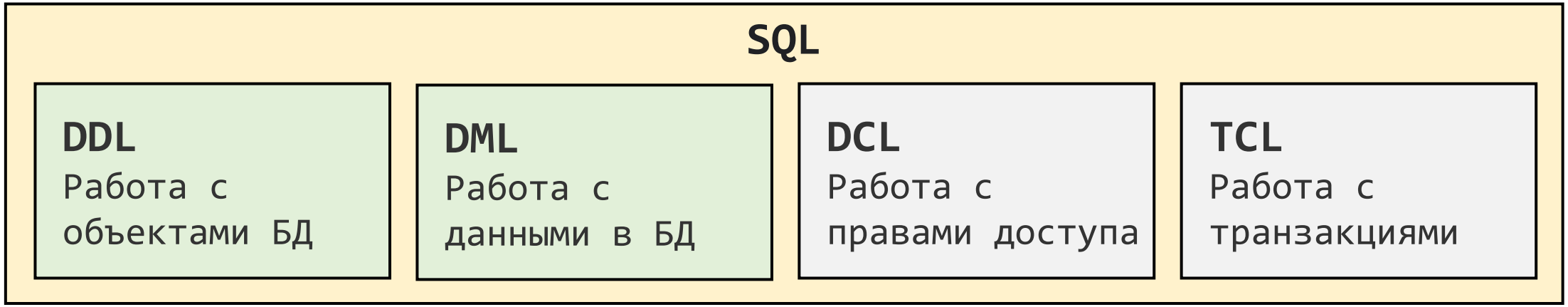
Типы time, timestamp принимают необязательное значение точности p, определяющее, сколько знаков после запятой должно сохраняться в секундах. По умолчанию точность не ограничивается. Допустимые значения p лежат в интервале от 0 до 6

Имя	Размер	Описание	Наименьшее значение	Наибольшее значение	Точность
`timestamp [ (p) ] [ without time zone ]`	8 байт	дата и время (без часового пояса)	4713 до н. э.	294276 н. э.	1 микросекунда
`timestamp [ (p) ] with time zone`	8 байт	дата и время (с часовым поясом)	4713 до н. э.	294276 н. э.	1 микросекунда
`date`	4 байта	дата (без времени суток)	4713 до н. э.	5874897 н. э.	1 день
`time [ (p) ] [ without time zone ]`	8 байт	время суток (без даты)	00:00:00	24:00:00	1 микросекунда
`time [ (p) ] with time zone`	12 байт	время дня (без даты), с часовым поясом	00:00:00+1559	24:00:00-1559	1 микросекунда

DDL, DML, DCL и TCL

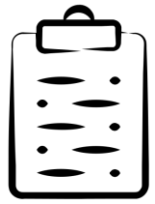


SQL представляет собой набор операторов, которые делятся на определенные группы и у каждой группы есть свое назначение. В сокращенном виде эти группы называются DDL, DML, DCL и TCL.



DCL - Data Control Language

КРОК



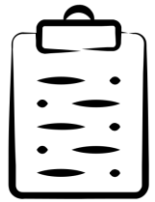
Data Control Language (DCL) – группа операторов определения доступа к данным. Иными словами, это операторы для управления разрешениями, с помощью них мы можем разрешать или запрещать выполнение определенных операций над объектами базы данных.

Сюда входят:

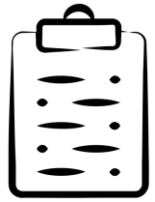
- GRANT – предоставляет пользователю или группе разрешения на определённые операции с объектом;
- REVOKE – отзывает выданные разрешения;
- DENY – задаёт запрет, имеющий приоритет над разрешением.

TCL - Transaction Control Language

КРОК



- Transaction Control Language (TCL) – группа операторов для управления транзакциями.
- Транзакция – это команда или блок команд (инструкций), которые успешно завершаются как единое целое, при этом в базе данных все внесенные изменения фиксируются на постоянной основе или отменяются, т.е. все изменения, внесенные любой командой, входящей в транзакцию, будут отменены.

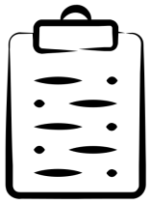


Группа операторов TCL предназначена для реализации и управления транзакциями. Сюда можно отнести:

- BEGIN TRANSACTION – служит для определения начала транзакции;
- COMMIT TRANSACTION – применяет транзакцию;
- ROLLBACK TRANSACTION – откатывает все изменения, сделанные в контексте текущей транзакции;
- SAVE TRANSACTION – устанавливает промежуточную точку сохранения внутри транзакции.

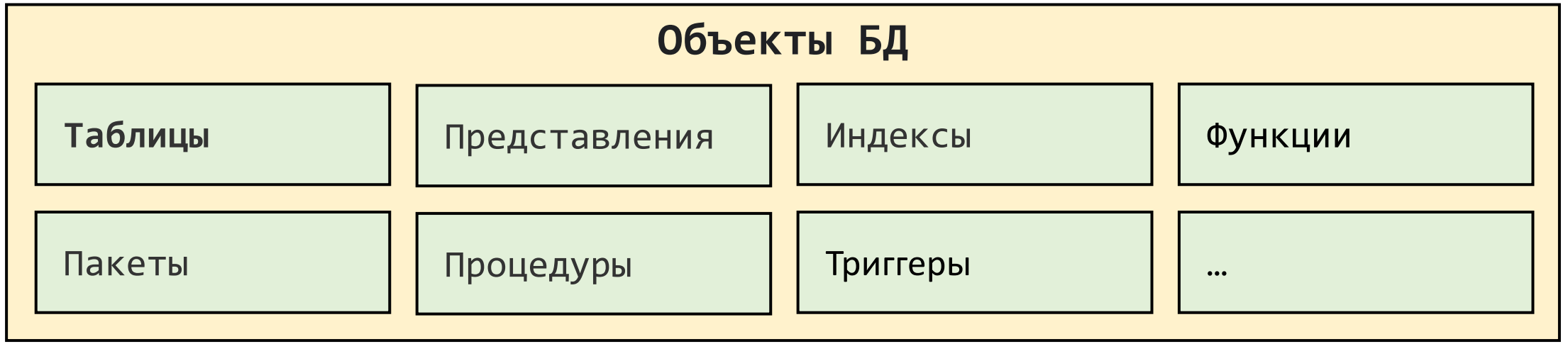
DDL - Data Definition Language

КРОК



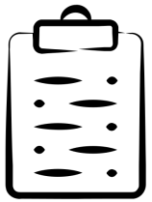
Data Definition Language (DDL) – это группа операторов определения данных. Создание, удаление, изменение объектов структуры БД.

- CREATE – используется для создания объектов базы данных;
- ALTER – используется для изменения объектов базы данных;
- DROP – используется для удаления объектов базы данных.



DDL. Создание таблицы

КРОК

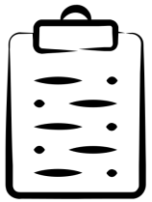


- Основными объектами в реляционной базе данных являются таблицы. Именно в них происходит хранение данных. В данном разделе мы рассмотрим вопросы, связанные с созданием и удалением таблиц в базе данных.
- Базовый синтаксис создания таблицы имеет следующий вид:

```
create table имя_таблицы (
    имя_столбца_1 тип_данных_столбца_1,
    имя_столбца_2 тип_данных_столбца_2,
    ...
    имя_столбца_N тип_данных_столбца_N
);
```

DDL. Пример создания таблицы

КРОК



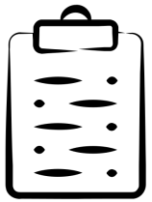
Создадим таблицу users

Название	Тип данных	Длина	Описание
surname	character varying	40	Фамилия
firstname	character varying	40	Имя
lastname	character varying	40	Отчество
flights	integer		Количество полетов

```
create table users
(
    "surname"   character varying(40),
    "firstname" character varying(40),
    "lastname"  character varying(40),
    "flights"   integer
);
```

DDL. Удаление таблицы

КРОК



Для того, чтобы удалить таблицу необходимо воспользоваться командой:

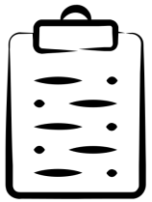
```
drop table имя_таблицы;
```

Чтобы удалить таблицу users надо указать:

```
drop table users;
```

DML

КРОК



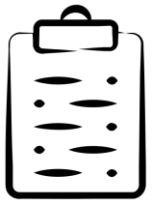
Data Manipulation Language (DML) – это группа операторов для манипуляции данными. С помощью этих операторов мы можем добавлять, изменять, удалять и выгружать данные из базы, т.е. манипулировать ими.

В эту группу входят самые распространённые операторы языка SQL:

- SELECT – осуществляет выборку данных;
- INSERT – добавляет новые данные;
- UPDATE – изменяет существующие данные;
- DELETE – удаляет данные.

DML. Вставка данных

КРОК



Синтаксис вставки данных в таблицу имеет следующий вид:

```
insert into ИМЯ_ТАБЛИЦЫ (НАЗВАНИЕ_СТОЛБЦА_1, НАЗВАНИЕ_СТОЛБЦА_2...)
values (ЗНАЧЕНИЕ_СТОЛБЦА_1, ЗНАЧЕНИЕ_СТОЛБЦА_2)
```

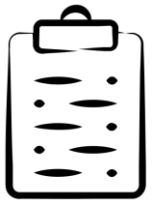
Названия столбцов таблицы могут быть перечислены через запятую в любом порядке.

```
insert into users (surname, firstname, lastname, flights) VALUES
  ('Степанов',    'Иван',      'Дмитриевич',    1),
  ('Носова',      'Анастасия', 'Демьяновна',    2),
  ('Сафонов',     'Данил',     'Владимирович',  3),
  ('Романова',    'Ева',       'Степановна',    4),
  ('Виноградова', 'Марина',    'Александровна', 4),
  ('Сергеева',    'Анна',      'Серафимовна',   4),
  ('Головин',     'Иван',      'Андреевич',     3),
  ('Васильев',    'Василий',   'Тимофеевич',    null),
  ('Ермакова',    'Анастасия', 'Алексеевна',    null),
  ('Орлов',       'Лев',       'Артёмович',     1);
```

Необязательно заполнять некоторые столбцы в таблице. Значения в данных столбцах могут остаться пустыми

DML. Получение данных

КРОК



Для чтения данных из таблицы используется SQL-команда **select**. Синтаксис запроса для чтения всех данных из таблицы имеет вид:

```
select СТОЛБЕЦ_ТАБЛИЦЫ_1,
       СТОЛБЕЦ_ТАБЛИЦЫ_2,
       ...
       СТОЛБЕЦ_ТАБЛИЦЫ_N
  from НАЗВАНИЕ_ТАБЛИЦЫ
```

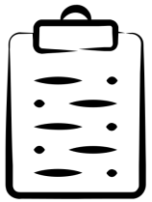
```
select *
  from НАЗВАНИЕ_ТАБЛИЦЫ
```

```
select surname,
       firstname,
       lastname,
       flights
  from users
```

surname character varying (40)	firstname character varying (40)	lastname character varying (40)	flights integer
Степанов	Иван	Дмитриевич	1
Носова	Анастасия	Демьяновна	2
Сафонов	Данил	Владимирович	3
Романова	Ева	Степановна	4
Виноградова	Марина	Александровна	4
Сергеева	Анна	Серафимовна	4
Головин	Иван	Андреевич	3
Васильев	Василий	Тимофеевич	[null]
Ермакова	Анастасия	Алексеевна	[null]
Орлов	Лев	Артёмович	1

DML. Фильтрация

КРОК



Для этого в запрос необходимо добавить еще одну секцию **where**. Данная секция предназначена для фильтрации строк в исходной таблице по указанным условиям.

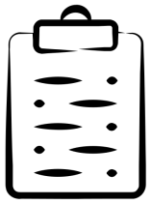
Секция **where** в отличии от секций **select** и **from** необязательна.

```
select СТОЛБЕЦ_ТАБЛИЦЫ_1
  from НАЗВАНИЕ_ТАБЛИЦЫ
 where УСЛОВИЕ

select СТОЛБЕЦ_ТАБЛИЦЫ_1
  from НАЗВАНИЕ_ТАБЛИЦЫ
 where УСЛОВИЕ1
   and УСЛОВИЕ2
    or (УСЛОВИЕ3 and УСЛОВИЕ4)
   and not УСЛОВИЕ5
```

DML. Фильтрация

КРОК



В секции where может находится как простое, так и составное логическое выражение. В качестве логических операторов используются следующие:

- логическое И - **and**
- логическое ИЛИ - **or**
- логическое отрицание - **not** или **!**

Кроме логических операторов в where используются операторы сравнения:

- больше **>**
- меньше **<**
- больше или равно **>=**
- меньше или равно **<=**
- равно **=**
- не равно **!=** или **<>**

Специальные операторы:

- like
- between
- in
- exists
- is null / is not null

DML. Примеры фильтрации

КРОК

```
select surname,
       firstname
  from users
 where firstname = 'Иван'
```

surname character varyin	firstname character var
Степанов	Иван
Головин	Иван

```
select surname,
       flights
  from users
 where flights > 3
```

surname character varying	flights integer
Романова	4
Виноградова	4
Сергеева	4

```
select surname,
       flights
  from users
 where flights is null
```

surname character varying	flights integer
Васильев	[null]
Ермакова	[null]

```
select surname,
       flights
  from users
 where flights in (1, 3)
```

surname character varying	flights integer
Степанов	1
Сафонов	3
Головин	3
Орлов	1

```
select firstname,
       flights
  from users
 where firstname <> 'Иван'
   and flights = 1
```

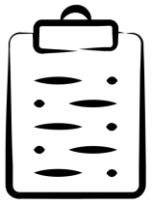
firstname character vary	flights integer
Лев	1

```
select firstname,
       flights
  from users
 where flights * 2 < 5
   and firstname not in
       ('Лев', 'Алексей')
```

firstname character varyi	flights integer
Иван	1
Анастасия	2

DML. Модификация данных

КРОК



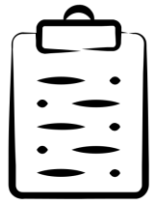
При работе информационных систем кроме добавления данных в таблицу, чтения данных из таблицы возникает потребность обновлять данные в некоторых строках таблицы. Для обновления данных в таблице в языке SQL используется команда UPDATE.

```
update НАЗВАНИЕ_ТАБЛИЦЫ set
    СТОЛБЕЦ_1_ТАБЛИЦЫ = ЗНАЧЕНИЕ_1
    СТОЛБЕЦ_2_ТАБЛИЦЫ = ЗНАЧЕНИЕ_2
    ...
 where УСЛОВИЕ_ОТБОРА_СТРОК

update users
   set flights = 6
 where firstname = 'Иван'
   and surname = 'Головин'
```

DML. Удаление данных

КРОК



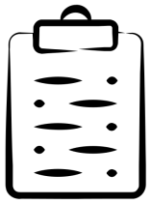
Чтобы удалить данные из таблицы, в SQL используется команда delete. Синтаксис команды имеет следующий вид:

```
delete
  from НАЗВАНИЕ_ТАБЛИЦЫ
 where УСЛОВИЕ_ОТБОРА_СТРОК
```

```
delete
  from users
 where firstname = 'Иван'
   and surname = 'Головин'
```

Псевдонимы (ALIASES) таблиц и столбцов

КРОК



ALIASES можно использовать для создания временного имени для таблиц или столбцов.

- Псевдонимы столбцов используются для облегчения чтения заголовков столбцов в наборе результатов
- Псевдонимы таблиц используются для сокращения SQL-кода

```
column_name [ AS ] alias_name
table_name [ AS ] alias_name
```

```
select u.surname col1
  from users u
 where u.flights > 3
```

col1 character varying (40)
Романова
Виноградова
Сергеева

```
select u.surname "Столбик"
  from users u
  where u.flights > 3
```

Столбик character varying (40)
Романова
Виноградова
Сергеева

Конкатенация данных

КРОК

Для объединения (сложения) строк используется оператор ||

```
select 'Привет, ' || surname hey_str,
       firstname || ' ' || lastname name1
  from users
```

hey_str text	name1 text
Привет, Степанов	Иван Дмитриевич
Привет, Носова	Анастасия Демьяновна
Привет, Сафонов	Данил Владимирович
Привет, Романова	Ева Степановна
Привет, Виноградова	Марина Александровна
Привет, Сергеева	Анна Серафимовна
Привет, Васильев	Василий Тимофеевич
Привет, Ермакова	Анастасия Алексеевна
Привет, Орлов	Лев Артёмович
Привет, Головин	Иван Андреевич

```
select *
  from users
 where surname || firstname = 'СтепановИван'
```

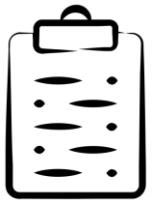
surname character varying	firstname character var	lastname character varying	flights integer
Степанов	Иван	Дмитриевич	1

```
select surname,
       flights || ' шт.' flights
  from users
 where flights = 3
```

surname character varying (	flights text
Сафонов	3 шт.
Головин	3 шт.

Сортировка

КРОК



- При извлечении данных из таблицы, можно указать способ сортировки данных при помощи оператора order by

```
select u.surname,
       u.flights
  from users u
 order by u.surname
```

surname character varying (40)	integer
Васильев	[null]
Виноградова	4
Головин	3
Ермакова	[null]
Носова	2
Орлов	1
Романова	4
Сафонов	3
Сергеева	4
Степанов	1

```
select u.surname,
       u.flights
  from users u
 order by u.surname desc
```

surname character varying (40)	integer
Степанов	1
Сергеева	4
Сафонов	3
Романова	4
Орлов	1
Носова	2
Ермакова	[null]
Головин	3
Виноградова	4
Васильев	[null]

```
select u.surname,
       u.flights
  from users u
 order by u.flights, u.surname
```

surname character varying (40)	integer
Орлов	1
Степанов	1
Носова	2
Головин	3
Сафонов	3
Виноградова	4
Романова	4
Сергеева	4
Васильев	[null]
Ермакова	[null]

Сортировка

КРОК

```
select u.surname,
       u.flights
  from users u
 order by u.flights desc,
          u.surname
```

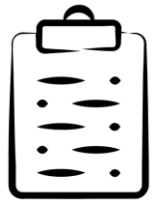
surname character varying (40)	integer
Васильев	[null]
Ермакова	[null]
Виноградова	4
Романова	4
Сергеева	4
Головин	3
Сафонов	3
Носова	2
Орлов	1
Степанов	1

```
select u.surname,
       u.flights
  from users u
 order by u.flights desc nulls last,
          u.surname
```

surname character varying (40)	integer
Виноградова	4
Романова	4
Сергеева	4
Головин	3
Сафонов	3
Носова	2
Орлов	1
Степанов	1
Васильев	[null]
Ермакова	[null]

Оператор case

КРОК



- Выражение CASE в SQL представляет собой общее условное выражение, напоминающее операторы if/else в других языках программирования

```
case when условие then результат
     [WHEN ...]
     [ELSE результат]
end
```

```
select firstname,
       case firstname
         when 'Ева' then 'SOME 1'
         when 'Данил' then 'SOME 2'
         else firstname
       end test
  from users
```

firstname character varying (40)	test character varying
Иван	Иван
Анастасия	Анастасия
Данил	SOME 2
Ева	SOME 1
Марина	Марина
Анна	Анна
Василий	Василий
Анастасия	Анастасия
Лев	Лев
Иван	Иван

Оператор case

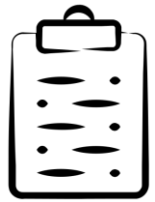
КРОК

```
select firstname,
       case when flights >= 3
              then 'Много!'
            else 'Мало'
       end flights_txt,
       flights
  from users
```

firstname character varying (40)	flights_txt text	flights integer
Иван	Мало	1
Анастасия	Мало	2
Данил	Много!	3
Ева	Много!	4
Марина	Много!	4
Анна	Много!	4
Василий	Мало	[null]
Анастасия	Мало	[null]
Лев	Мало	1
Иван	Много!	3

Оператор like

КРОК



Выражение LIKE возвращает true, если строка соответствует заданному шаблону.

```
строка LIKE шаблон [ESCAPE спецсимвол]
```

Первый символ равен A

```
select firstname
  from users
 where firstname like 'A%'
```

firstname character varying (40)
Анастасия
Анна
Анастасия

Последний символ равен a

```
select firstname
  from users
 where firstname like '%a'
```

firstname character varying (40)
Ева
Марина
Анна

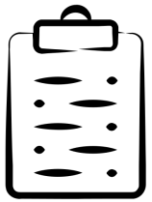
Есть вхождение подстроки «на» в любом месте

```
select firstname
  from users
 where firstname like '%на%'
```

firstname character varying (40)
Анастасия
Марина
Анна
Анастасия

Примеры встроенных функций

КРОК



- replace(<строка>, <искомая подстрока>, <строка замены>) – заменяет искомую подстроку на нужную
- substr(<строка>, <позиция>, <длина>) – выделяет подстроку указанной длины из строки начиная с указанной позиции
- length(<строка>) – длина строки
- reverse(<строка>) – перевернуть строку
- lower(<строка>) – привести строку к нижнему регистру
- upper(<строка>) – привести строку к верхнему регистру

Примеры встроенных функций

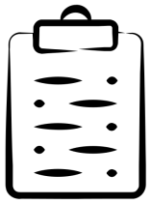
КРОК

```
select u.surname,
       u.firstname,
       replace(u.firstname, 'a', '###') replace_test,
       substr(u.surname, 2, 2) substr_test,
       length(u.firstname) * 2 length_test,
       upper(reverse(u.surname)) reverse_up_test
  from users u
```

surname character varying (	firstname character varying	replace_test text	substr_test text	length_test integer	reverse_up_test text
Степанов	Иван	Ив###н	те	8	ВОНАПЕТС
Носова	Анастасия	Ан###ст###сия	ос	18	АВОСОН
Сафонов	Данил	Д###нил	аф	10	ВОНОФАС
Романова	Ева	Ев###	ом	6	АВОНАМОР
Виноградова	Марина	М###рин###	ин	12	АВОДАРГОНИВ
Сергеева	Анна	Анн###	ер	8	АВЕЕГРЕС
Васильев	Василий	В###силий	ас	14	ВЕЬЛИСАВ
Ермакова	Анастасия	Ан###ст###сия	рм	18	АВОКАМРЕ
Орлов	Лев	Лев	рл	6	ВОЛРО
Головин	Иван	Ив###н	ол	8	НИВОЛОГ

Примеры агрегатных функций

КРОК



- count() – кол-во записей
- sum() – сумма по указанному полю
- max() – максимальное значение
- min() – минимальное значение
- avg() – среднее значение

```
select avg(flights),
       sum(flights),
       count(*),
       sum(case when firstname = 'Иван'
                  then 1
                else 0
            end),
       max(flights)
  from users
```

avg numeric	sum bigint	count bigint	sum bigint	max integer
2.7500000000000000	22	10	2	4

Пример 1

КРОК

Вывести ФИО в формате Фамилия И.О. одной строкой. Отсортировать результаты по фамилии в порядке убывания по алфавиту.

```
select u.surname || ' ' || substr(u.firstname, 1, 1) || '.' ||
       substr(u.lastname, 1, 1) || '.' FIO
  from users u
 order by u.surname desc
```

surname character varying	firstname character varyi	lastname character varying (	flights integer
Степанов	Иван	Дмитриевич	1
Носова	Анастасия	Демьяновна	2
Сафонов	Данил	Владимирович	3
Романова	Ева	Степановна	4
Виноградова	Марина	Александровна	4
Сергеева	Анна	Серафимовна	4
Головин	Иван	Андреевич	3
Васильев	Василий	Тимофеевич	[null]
Ермакова	Анастасия	Алексеевна	[null]
Орлов	Лев	Артёмович	1

fio text
Степанов И.Д.
Сергеева А.С.
Сафонов Д.В.
Романова Е.С.
Орлов Л.А.
Носова А.Д.
Ермакова А.А.
Головин И.А.
Виноградова М.А.
Васильев В.Т.

Пример 2

КРОК

Вывести Фамилию и Имя тех, у кого в поле «Имя» есть буква «е». Регистр не учитывать. Отсортировать результаты по фамилии в порядке возрастания по алфавиту.

```
select u.surname,
       u.firstname
  from users u
 where lower(u.firstname) like '%e%'
 order by u.surname
```

Аналогичный результат

```
upper(u.firstname) like '%E%'
```

surname character varying	firstname character varyi	lastname character varying (	flights integer
Степанов	Иван	Дмитриевич	1
Носова	Анастасия	Демьяновна	2
Сафонов	Данил	Владимирович	3
Романова	Ева	Степановна	4
Виноградова	Марина	Александровна	4
Сергеева	Анна	Серафимовна	4
Головин	Иван	Андреевич	3
Васильев	Василий	Тимофеевич	[null]
Ермакова	Анастасия	Алексеевна	[null]
Орлов	Лев	Артёмович	1

surname character varying (40)	firstname character varying (40)
Орлов	Лев
Романова	Ева

Пример 3

КРОК

Вывести Фамилию и Кол-во полетов тех, у кого в поле «Имя» строго больше 3 букв. Если полеты равны NULL, то выводить слово «ПУСТО!» Отсортировать результаты по фамилии в порядке возрастания по алфавиту.

```
select u.surname,
       case when u.flights is null then 'ПУСТО!'
            else u.flights || ''
       end flights2,
       u.flights
  from users u
 where length(u.firstname) > 3
 order by u.surname
```

surname character varying	firstname character varyi	lastname character varying (	flights integer
Степанов	Иван	Дмитриевич	1
Носова	Анастасия	Демьяновна	2
Сафонов	Данил	Владимирович	3
Романова	Ева	Степановна	4
Виноградова	Марина	Александровна	4
Сергеева	Анна	Серафимовна	4
Головин	Иван	Андреевич	3
Васильев	Василий	Тимофеевич	[null]
Ермакова	Анастасия	Алексеевна	[null]
Орлов	Лев	Артёмович	1

→

surname character varying (40)	flights2 text	flights integer
Васильев	ПУСТО!	[null]
Виноградова	4	4
Головин	3	3
Ермакова	ПУСТО!	[null]
Носова	2	2
Сафонов	3	3
Сергеева	4	4
Степанов	1	1

Пример 4

КРОК

Вывести весь список со столбцами Фамилия, Имя, Отчество. Если **фамилия** начинается на букву «В», то выводить вместо **отчества** слово «FOUND».

```
select u.surname,
       u.firstname,
       case when u.surname like 'B%'
              then 'FOUND'
            else u.lastname
       end lastname
  from users u
```

Аналогичный результат

```
when substr(u.surname, 1, 1) = 'B'
```

surname character varying	firstname character varyi	lastname character varying (	flights integer
Степанов	Иван	Дмитриевич	1
Носова	Анастасия	Демьяновна	2
Сафонов	Данил	Владимирович	3
Романова	Ева	Степановна	4
Виноградова	Марина	Александровна	4
Сергеева	Анна	Серафимовна	4
Головин	Иван	Андреевич	3
Васильев	Василий	Тимофеевич	[null]
Ермакова	Анастасия	Алексеевна	[null]
Орлов	Лев	Артёмович	1

surname character varying (40)	firstname character varying (40)	lastname character varying
Степанов	Иван	Дмитриевич
Носова	Анастасия	Демьяновна
Сафонов	Данил	Владимирович
Романова	Ева	Степановна
Виноградова	Марина	FOUND
Сергеева	Анна	Серафимовна
Головин	Иван	Андреевич
Васильев	Василий	FOUND
Ермакова	Анастасия	Алексеевна
Орлов	Лев	Артёмович

Южно-Уральский государственный университет
Национальный исследовательский университет

1943

КРОК

Спасибо за внимание!

КРОК
Челябинск, ул. Карла Маркса, д. 38