

Инструкция SELECT

Лекция 7

Доцент каф. ВТиКГ Данилова Е.В.

План

1. Описание инструкции
2. Предложение SELECT
3. Предложение WHERE
4. Предложение GROUP BY

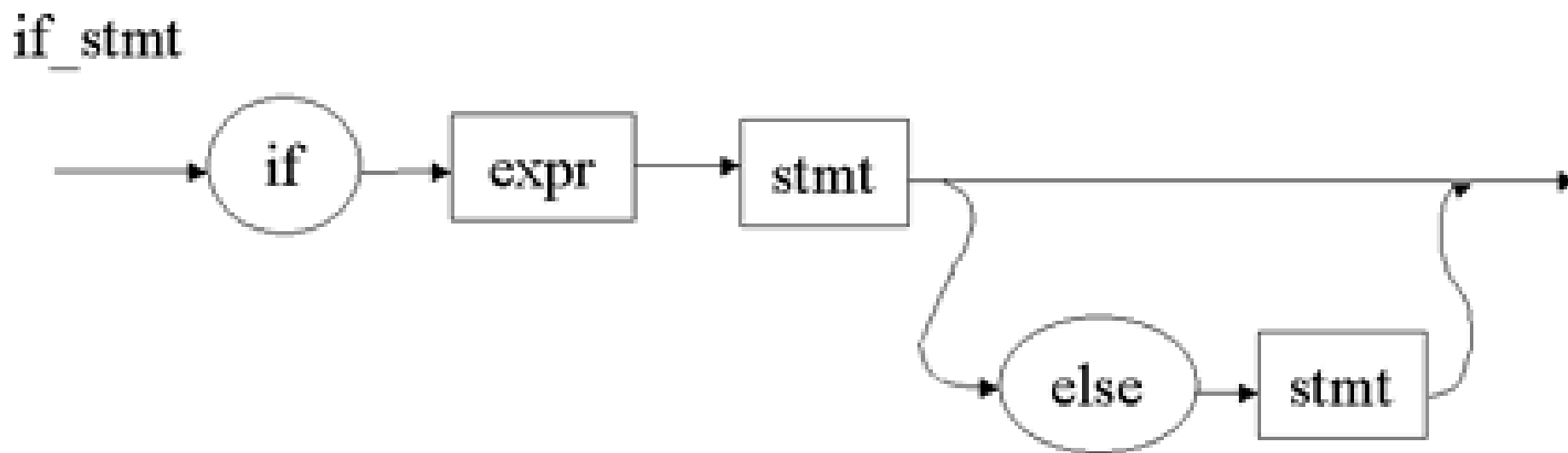
Описание инструкции

1

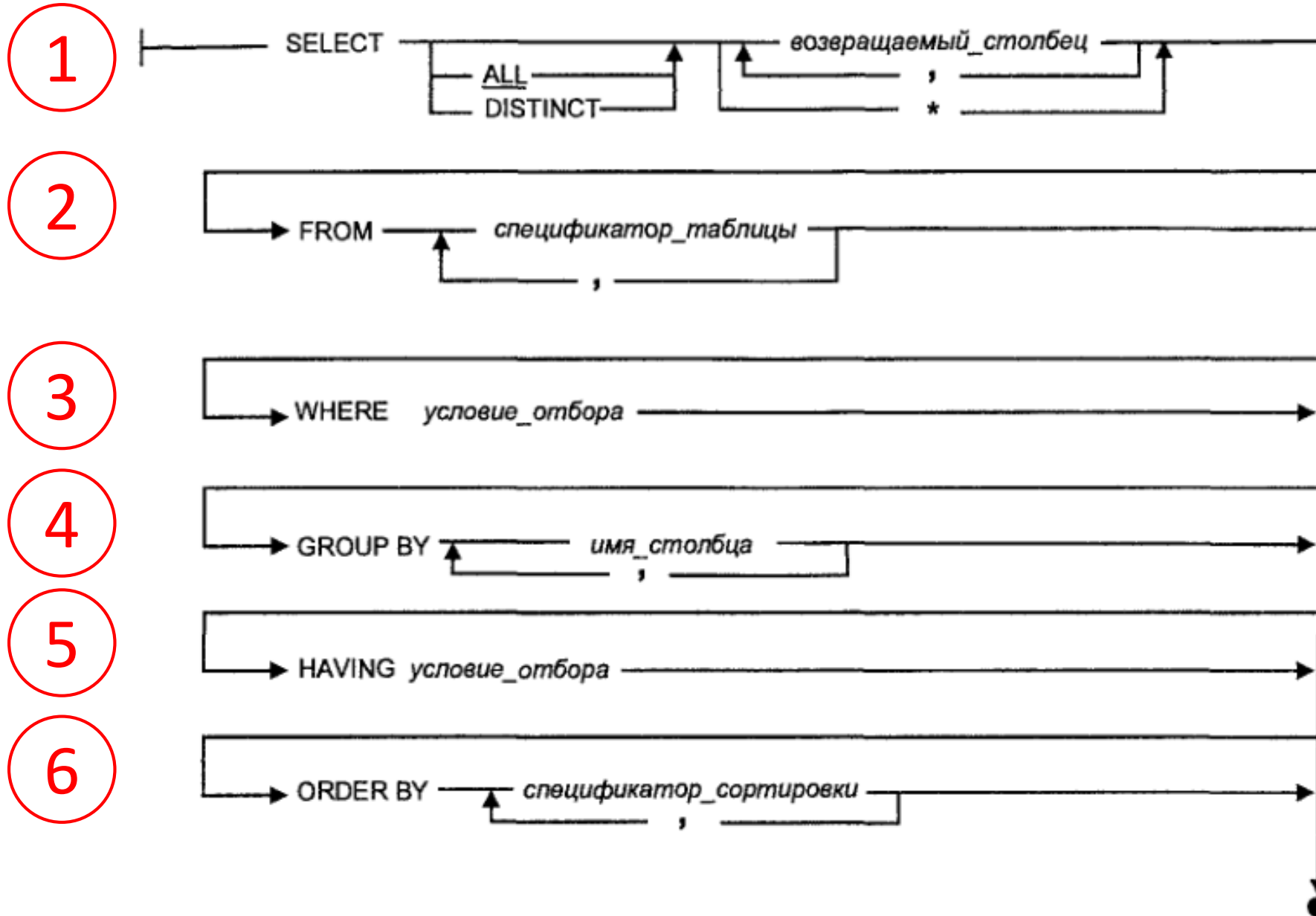
- Относится к компоненту DML (Data Manipulation Language, язык запросов к данным)
- Инструкция SELECT – наиболее мощная и сложная из всех инструкций языка SQL
- Извлекает информацию из базы данных и возвращает ее в виде результатов запроса
- Состоит из шести предложений (стандарт SQL:2003)

Синтаксическая диаграмма

- направленный граф с одним входным ребром и одним выходным ребром, имеет помеченные вершины
- Предназначена для графического изображения структуры синтаксической единицы языка.



Синтаксическая диаграмма



Синтаксис инструкции SELECT для СУБД MySQL 8.0

<https://dev.mysql.com/doc/refman/8.0/en/select.html>

```
1 SELECT
2     [ALL | DISTINCT | DISTINCTROW ]
3     [HIGH_PRIORITY]
4     [STRAIGHT_JOIN]
5     [SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
6     [SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
7     select_expr [, select_expr] ...
8     [into_option]
9     [FROM table_references
10      [PARTITION partition_list]]
11     [WHERE where_condition]
12     [GROUP BY {col_name | expr | position}, ... [WITH ROLLUP]]
13     [HAVING where_condition]
14     [WINDOW window_name AS (window_spec)
15      [, window_name AS (window_spec)] ...]
16     [ORDER BY {col_name | expr | position}
17      [ASC | DESC], ... [WITH ROLLUP]]
18     [LIMIT {[offset,] row_count | row_count OFFSET offset}]
19     [into_option]
20     [FOR {UPDATE | SHARE}
21      [OF tbl_name [, tbl_name] ...]
22      [NOWAIT | SKIP LOCKED]
23      | LOCK IN SHARE MODE]
24     [into_option]
25
26 into_option: {
27     INTO OUTFILE 'file_name'
28         [CHARACTER SET charset_name]
29         export_options
30 | INTO DUMPFILE 'file_name'
31 | INTO var_name [, var_name] ...
32 }
```

1. Предложение SELECT

- Перечисляются элементы данных, которые должны быть выбраны инструкцией SELECT (столбцы базы данных или столбцы, вычисляемые при выполнении запроса).
- Элементы данных задаются в виде списка выбора (списка возвращаемых столбцов), состоящего из выбираемых элементов, разделенных запятыми.
- Для каждого элемента из списка выбора (списка возвращаемых столбцов) в таблице результатов запроса будет создан один столбец; столбцы в таблице результатов будут расположены в том же порядке слева направо, что и элементы списка возвращаемых столбцов.

Возвращаемые элементы


- **Имя столбца** (идентифицирует один из столбцов, содержащихся в таблицах, которые перечислены в предложении FROM).
- **Константа** (показывает, что в каждой строке результатов запроса должно содержаться одно и то же значение).
- **Выражение** (указывает, что SQL должен вычислить значение, помещаемое в результаты запроса, по формуле, определенной в выражении).

2. Предложение FROM

- Указывается список таблиц или представлений, которые содержат элементы данных, извлекаемые запросом.
- Каждая спецификация таблицы идентифицирует таблицу или представление, содержащие данные, которые извлекает запрос.
- Такие таблицы называются исходными таблицами запроса (и инструкции SELECT), так как они являются источниками всех данных, содержащихся в таблице результатов запроса.


Виды запросов в зависимости от количества таблиц в предложении FROM

Однотабличные
(простые)
запросы



```
SELECT ORDER_NUM , NAME  
FROM ORDERS  
WHERE ORDER_NUM <> 123 ;
```

Многотабличные
запросы
(соединения)



```
SELECT ORDER_NUM , AMOUNT  
FROM ORDERS , CUSTOMERS  
WHERE CUST = CUST_NUM ;
```

3. Предложение WHERE

- Указывает, что в результаты запроса следует включать только некоторые строки. Для отбора строк используется условие отбора.
- Допустимо использовать в предложении WHERE вложенные подзапросы.

4. Предложение GROUP BY

- Позволяет создать итоговый запрос.
- Вместо генерации одной строки результата для каждой строки данных в базе данных, итоговый запрос вначале группирует строки базы данных по определенному признаку, а затем включает в результаты запроса одну итоговую строку для каждой группы.

5. Предложение HAVING

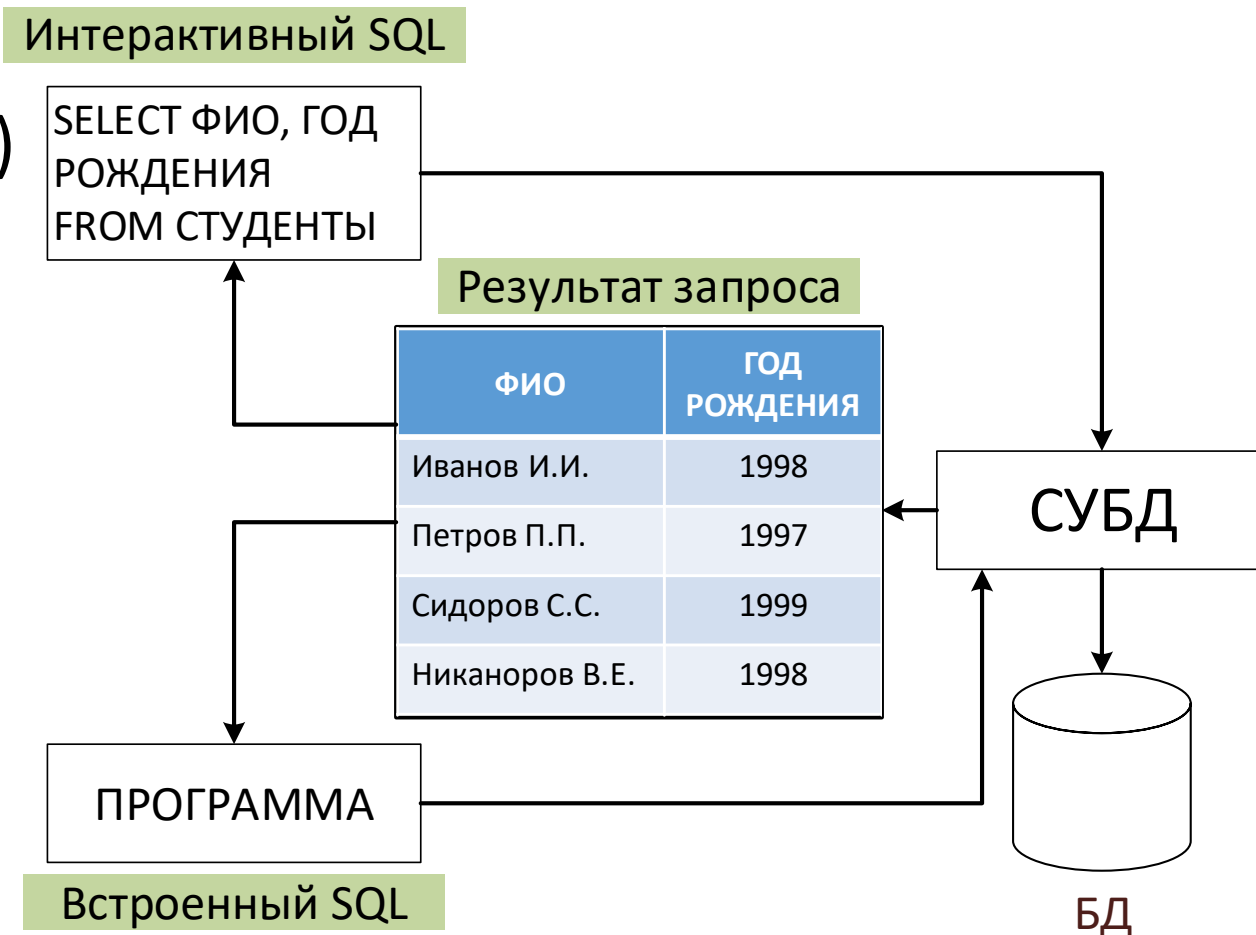
- Указывает, что в результаты запроса следует включать только некоторые из групп, созданных с помощью предложения GROUP BY.
- Для отбора включаемых групп используется условие отбора.

6. Предложение ORDER BY

- Сортирует результаты запроса по данным, содержащимся в одном или нескольких столбцах.
- Если это предложение отсутствует, результаты запроса не будут отсортированы.

Результаты запроса на выборку

- Результат SQL-запроса (*результатирующий набор, result set*) – таблица, не отличающаяся от других таблиц базы данных.
- Выводится на экран или возвращается приложению в табличной форме.
- Имеет такой же формат, как и обычные таблицы базы данных.
- Может содержать несколько строк, одну строку или ноль строк.



Результат запроса
на выборку -
таблица

```
graph TD; A[Результат запроса на выборку - таблица] --- B[Сохранение в виде таблицы БД]; A --- C[Объединение результатов подобных запросов в таблицу]; A --- D[Результаты – предмет дальнейших запросов];
```

Сохранение в виде
таблицы БД

Объединение
результатов
подобных
запросов в таблицу

Результаты –
предмет
дальнейших
запросов

Предложение SELECT

2

Вычисляемые столбцы

2.1

Вычисляемые столбцы

- Значения вычисляемых столбцов определяются на основании значений, хранящихся в базе данных.
- Для получения такого столбца в списке возвращаемых столбцов инструкции SELECT необходимо указать выражение (допустимы операции сложения, вычитания, умножения и деления, скобки).

КУРС	ФИО	ГОД_РОЖД	СРЕДНИЙ БАЛЛ	ГОД_ПОСТ
4	Федоров А.И.	1996	3	2014
2	Иванов И.И.	1998	4	2015
3	Петров П.В.	1997	3	2016
1	Сидоров С.А.	1999	5	2017

Таблица СТУДЕНТЫ

Вычисляемый
столбец

SQL-запрос на выборку

Выражение

```
SELECT ФИО, ГОД_ПОСТ – ГОД_РОЖД  
FROM СТУДЕНТЫ;
```

Результат запроса

ФИО	Expr1001
Федоров А.И.	18
Иванов И.И.	17
Петров П.В.	19
Сидоров С.А.	18

Примеры запросов, создающих вычисляемые столбцы

```
SELECT CITY, REGION, (SALES - TARGET)
       FROM OFFICES;
```

```
SELECT MFR_ID, PRODUCT_ID, DESCRIPTION, (QTY_ON_HAND * PRICE)
       FROM PRODUCTS;
```

```
SELECT NAME, QUOTA, (QUOTA + (.03 * SALES))
       FROM SALESREPS;
```

Другие элементы в выражении в списке возвращаемых столбцов

- арифметические операции

Обозначение	Описание
%, MOD	Деление по модулю
*	Умножение
+	Сложение
-	Вычитание
-	Унарный минус (меняет знак операнда)
/	Деление
DIV	Целочисленное деление

- операции над строками символов
- встроенные функции
- константы

Встроенные функции в списке возвращаемых столбцов

Исходная таблица ДОГОВОР

ФИО	ДАТА_ДОГОВОРА	СПЕЦИАЛЬНОСТЬ
Завгородний А.П.	12.11.2012	ИВТ
Светличная Е.И.	24.04.2011	БИ
Карандышев В.П.	01.02.2013	ФОИ
Иванов И.И.	17.09.1999	УК

SQL-запрос (MS Access)

```
SELECT ФИО, Year(ДАТА_ДОГОВОРА),  
           Month(ДАТА_ДОГОВОРА)  
FROM ДОГОВОР;
```

Результат запроса

ФИО	Expr1001	Expr1002
Завгородний А.П.	2012	11
Светличная Е.И.	2011	4
Карандышев В.П.	2013	2
Иванов И.И.	1999	9

Константы в списке возвращаемых столбцов

Таблица ДОГОВОР

ФИО	ДАТА_ДОГОВОРА	СПЕЦИАЛЬНОСТЬ
Завгородний А.П.	12.11.2012	ИБТ
Светличная Е.И.	24.04.2011	БИ
Карандышев В.П.	01.02.2013	ФОИ
Иванов И.И.	17.09.1999	УК

SQL-запрос (MS Access)

```
SELECT ФИО, 'Договор заключен',  
           Month(ДАТА_ДОГОВОРА)  
FROM ДОГОВОР;
```

Результат запроса

ФИО	Expr1001	Expr1002
Завгородний А.П.	Договор заключен	11
Светличная Е.И.	Договор заключен	4
Карандышев В.П.	Договор заключен	2
Иванов И.И.	Договор заключен	9

Выборка всех столбцов

2.2

- В списке возвращаемых столбцов используется символ звездочки *

```
SELECT * FROM STUDENTS;
```

- Столбцы в результате запроса располагаются в том же порядке, что и в исходной таблице
- Не использовать * в программном SQL, т.к. возможны ошибки выполнения приложения. Допустимо в программном SQL перечислять требуемые столбцы по именам.

Пример программной ошибки при использовании знака *

Предположим, что таблица **STUDENTS** была удалена из базы данных, а затем создана вновь, при этом был изменен порядок столбцов, тип данных для значений столбцов и добавлен новый, седьмой, столбец. Если программа ожидает, что запрос **SELECT * FROM STUDENTS** возвратит результат, содержащий шесть столбцов определенных типов, она почти наверняка перестанет работать после изменения порядка столбцов и добавления нового столбца.

Повторяющиеся строки

2.3

Исходная таблица

ДОГОВОР		
ФИО	ДАТА_ДОГОВОРА	СПЕЦИАЛЬНОСТЬ
Завгородний А.П.	12.11.2012	ИВТ
Светличная Е.И.	24.04.2011	БИ
Карандышев В.П.	01.02.2013	ФОИ
Иванов И.И.	17.09.1999	УК
Атанов Д.П.	23.02.2000	УК
Буряков С.М.	19.08.2012	ФОИ
Максименко А.Ю.	03.07.2015	БИ

SQL-запрос

```
SELECT СПЕЦИАЛЬНОСТЬ  
FROM ДОГОВОР;
```

Задача: узнать, по каким специальностям обучаются студенты

Результат запроса

СПЕЦИАЛЬНОСТЬ

ИВТ

БИ

ФОИ

УК

УК

ФОИ

БИ

В результате
запроса могут
присутствовать
повторяющиеся
записи



Исходная таблица

ДОГОВОР		
ФИО	ДАТА_ДОГОВОРА	СПЕЦИАЛЬНОСТЬ
Завгородний А.П.	12.11.2012	ИВТ
Светличная Е.И.	24.04.2011	БИ
Карандышев В.П.	01.02.2013	ФОИ
Иванов И.И.	17.09.1999	УК
Атанов Д.П.	23.02.2000	УК
Буряков С.М.	19.08.2012	ФОИ
Максименко А.Ю.	03.07.2015	БИ

- Ключевое слово DISTINCT перед списком возвращаемых столбцов удаляет повторяющиеся строки в результате запроса
- Ключевое слово ALL явно указывает, что повторяющиеся строки следует оставить (используется по умолчанию)

SQL-запрос

```
SELECT DISTINCT СПЕЦИАЛЬНОСТЬ  
FROM ДОГОВОР;
```

Результат запроса

СПЕЦИАЛЬНОСТЬ

БИ

ИВТ

УК

ФОИ

Предложение WHERE

Отбор строк

3.1

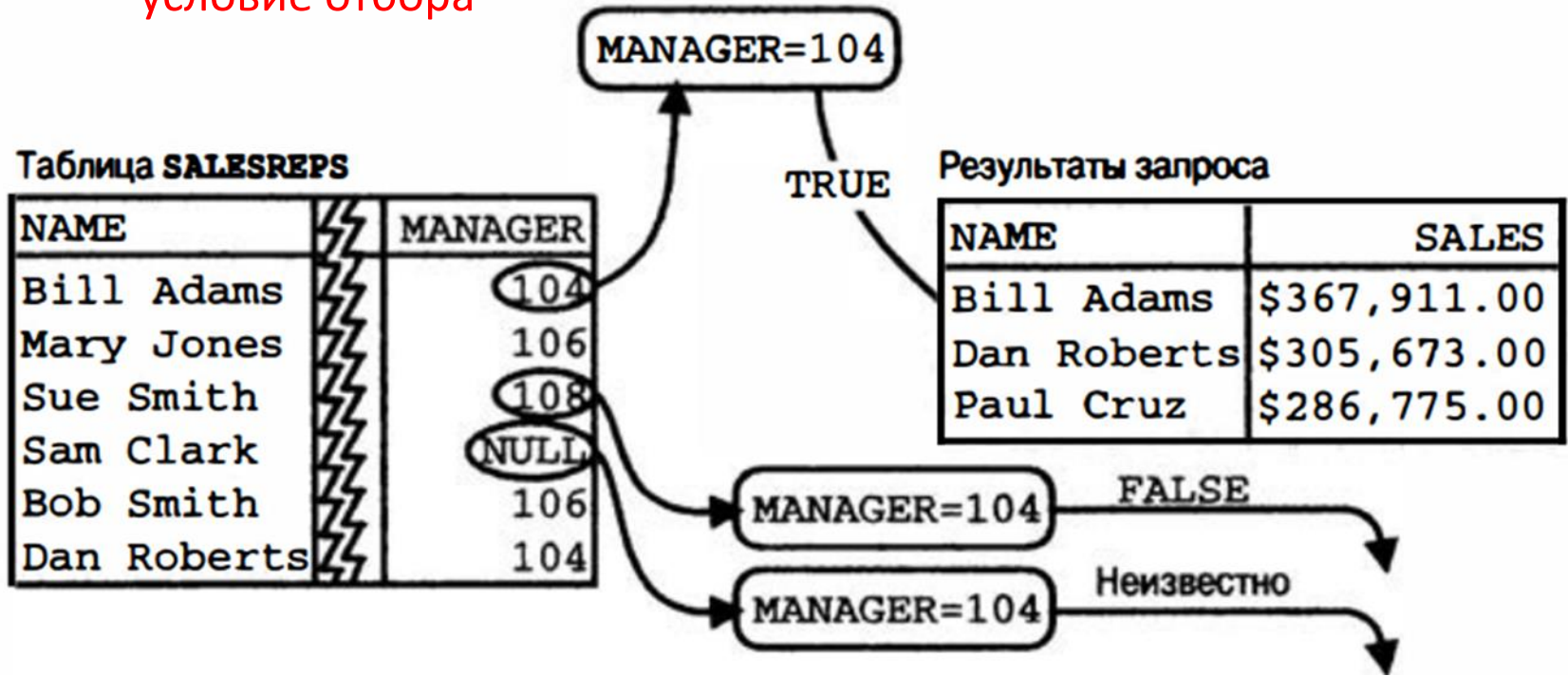
- Для отбора строк в SQL-запросе используется предложение WHERE
- Состоит из ключевого слова WHERE и выражения с условием отбора

```
SELECT NAME, SALES
FROM SALESREPS
WHERE MANAGER = 104;
```

Выражение –
условие отбора

Если условие отбора имеет значение:

- **TRUE**, строка из исходной таблицы будет включена в результаты запроса;
- **FALSE**, то строка исключается из результатов запроса;
- **NULL**, то строка исключается из результатов запроса.



Условия отбора

Пять основных условий отбора («**предикаты**» по стандарту ANSI/ISO).

1. Сравнение

2. Проверка на принадлежность диапазону

3. Проверка наличия во множестве

4. Проверка на соответствие шаблону

5. Проверка на равенство значению NULL

1. Сравнение

=, <>, <, <=, >, >=

```
SELECT CITY, SALES, TARGET
FROM OFFICES
WHERE SALES < (.8 * TARGET);
```

```
SELECT CITY, MGR
FROM OFFICES
WHERE MGR <> 108;
```

```
SELECT COMPANY, CREDIT_LIMIT
FROM CUSTOMERS
WHERE CUST_NUM = 2107;
```

При сравнении СУБД значений двух выражений могут быть получены три результата:

- сравнение истинно, результат проверки имеет значение **TRUE**;
- сравнение ложно, результат проверки имеет значение **FALSE**;
- если хотя бы одно из двух выражений имеет значение **NULL**, то результатом сравнения будет **NULL**

2. Проверка на принадлежность диапазону

- Оператор BETWEEN ... AND
- Верхняя и нижняя границы считаются частью диапазона
- Проверяемые выражения могут быть любыми допустимыми выражениями в SQL



```
SELECT ФИО  
FROM ДОГОВОР  
WHERE НОМЕР BETWEEN 100 AND 200;
```

```
SELECT NAME, SALES, QUOTA  
FROM SALESREPS  
WHERE SALES NOT BETWEEN (.8 * QUOTA) AND (1.2 * QUOTA);
```

Исходная таблица

ДОГОВОР

ФИО	ДАТА_ДОГОВОРА	СПЕЦИАЛЬНОСТЬ
Завгородний А.П.	12.11.2012	ИВТ
Светличная Е.И.	24.04.2011	БИ
Карандышев В.П.	01.02.2013	ФОИ
Иванов И.И.	17.09.1999	УК
Атанов Д.П.	23.02.2000	УК
Буряков С.М.	19.08.2012	ФОИ
Максименко А.Ю.	03.07.2015	БИ

SQL-запрос 1 (MS Access)

```
SELECT ФИО  
FROM ДОГОВОР  
WHERE ДАТА_ДОГОВОРА BETWEEN DateValue('01.01.2012')  
AND DateValue('31.12.2012');
```

Результат запроса 1

ФИО
Завгородний А.П.
Буряков С.М.

Задача: узнать, с какими студентами были заключены договоры в 2012 году

SQL-запрос 2 (MS Access)

```
SELECT ФИО  
FROM ДОГОВОР  
WHERE ДАТА_ДОГОВОРА NOT BETWEEN  
DateValue('01.01.2012') AND  
DateValue('31.12.2012');
```

Результат запроса 2

ФИО
Светличная Е.И.
Карандышев В.П.
Иванов И.И.
Атанов Д.П.
Максименко А.Ю.

Правила обработки значений NULL в проверке BETWEEN (стандарт ANSI/ISO)

- Если проверяемое выражение имеет значение NULL либо оба выражения, определяющие диапазон, равны NULL, то проверка BETWEEN возвращает NULL.
- Если выражение, определяющее нижнюю границу диапазона, имеет значение NULL, то проверка BETWEEN возвращает FALSE, когда проверяемое значение больше верхней границы диапазона, и NULL – в противном случае.
- Если выражение, определяющее верхнюю границу диапазона, имеет значение NULL, то проверка BETWEEN возвращает FALSE, когда проверяемое значение меньше нижней границы диапазона, и NULL – в противном случае.

3. Проверка наличия во множестве (IN)



- Выполняется проверка, соответствует ли значение какому-либо элементу заданного списка.
- С помощью NOT IN проверяется, что элемент данных не является членом заданного множества.
- Проверяемое выражение в операторе IN может быть любым допустимым SQL-выражением
- Если результатом проверяемого выражения является значение NULL, то проверка IN также возвращает NULL.
- Все элементы в списке заданных значений должны иметь один и тот же тип данных, который должен быть сравним с типом данных проверяемого выражения.
- В стандарте ANSI/ISO не определено максимальное количество элементов множества. Явный верхний предел не задан в большинстве СУБД.
- Следует избегать множеств, содержащих один элемент.

Исходная таблица

ДОГОВОР

ФИО	ДАТА_ДОГОВОРА	СПЕЦИАЛЬНОСТЬ
Завгородний А.П.	12.11.2012	ИВТ
Светличная Е.И.	24.04.2011	БИ
Карандышев В.П.	01.02.2013	ФОИ
Иванов И.И.	17.09.1999	УК
Атанов Д.П.	23.02.2000	УК
Буряков С.М.	19.08.2012	ФОИ
Максименко А.Ю.	03.07.2015	БИ

SQL-запрос 1 (MS Access)

```
SELECT ФИО  
FROM ДОГОВОР  
WHERE Year(ДАТА_ДОГОВОРА) IN (2000, 2013, 2015)
```

Результат запроса 1

ФИО
Карандышев В.П.
Атанов Д.П.
Максименко А.Ю.

Задача: узнать, с какими студентами были заключены договоры в определенных годах: 2000, 2013, 2015

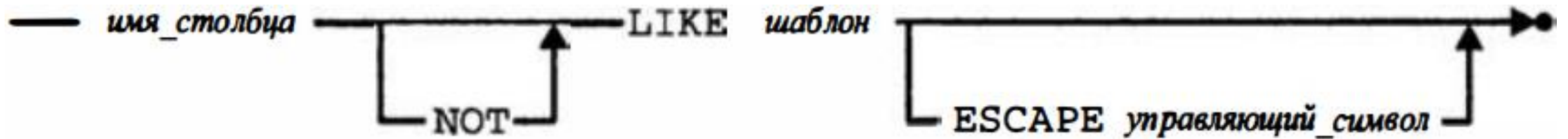
SQL-запрос 2 (MS Access)

```
SELECT ФИО  
FROM ДОГОВОР  
WHERE Year(ДАТА_ДОГОВОРА) NOT IN (2000,  
2013, 2015)
```

Результат запроса 2

ФИО
Завгородний А.П.
Светличная Е.И.
Иванов И.И.
Буряков С.М.

4. Проверка на соответствие шаблону



- Используется оператор **LIKE**
- Позволяет определить, соответствует ли значение данных в столбце некоторому шаблону.
- **Шаблон** – строка, в которую может входить один или несколько подстановочных символов/знаков.

- Виды подстановочных символов/знаков:
 - % заменяет любую последовательность символов (от нуля до ...);
 - _ (нижнее подчеркивание) заменяет любой один символ.
- Подстановочные символы/знаки можно помещать в любое место строки шаблона.
- В одной строке может содержаться несколько подстановочных символы/знаков.
- Проверку **LIKE** можно применять только к столбцам, имеющим строковый тип данных.
- Если в столбце содержится значение **NULL**, то результатом проверки будет **NULL**.

Исходная таблица

ДОГОВОР

ФАМИЛИЯ	ДАТА_ДОГОВОРА	СПЕЦИАЛЬНОСТЬ
Завгородний	12.11.2012	ИВТ
Светличная	24.04.2011	БИ
Карандышев	01.02.2013	ФОИ
Иванов	17.09.1999	УК
Атанов	23.02.2000	УК
Буряков	19.08.2012	ФОИ
Максименко	03.07.2015	БИ

SQL-запрос 2

```
SELECT ФАМИЛИЯ  
FROM ДОГОВОР  
WHERE ФАМИЛИЯ Like ' __ _анов ';
```

Результат запроса 2

ФАМИЛИЯ

Иванов

Атанов

SQL-запрос 1

```
SELECT ФАМИЛИЯ  
FROM ДОГОВОР  
WHERE ФАМИЛИЯ Like ' %в ';
```

Результат запроса 1

ФАМИЛИЯ

Карандышев

Иванов

Атанов

Буряков

SQL-запрос 3

```
SELECT ФАМИЛИЯ  
FROM ДОГОВОР  
WHERE ФАМИЛИЯ Like ' _а%о% ';
```

Результат запроса 3

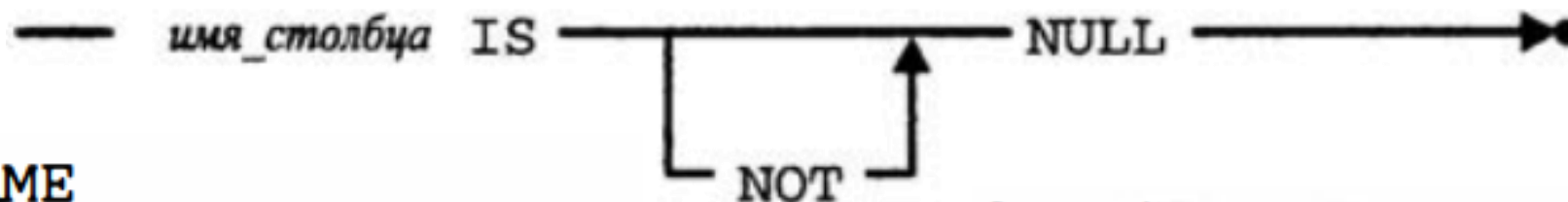
ФАМИЛИЯ

Завгородний

Максименко

5. Проверка на равенство NULL

- Используется оператор IS NULL. Обеспечивается возможность трехзначной логики в условиях отбора.
- Проверка на равенство NULL не может вернуть значение NULL в качестве результата. Она всегда возвращает TRUE ИЛИ FALSE.



```
SELECT NAME  
FROM SALESREPS  
WHERE REP_OFFICE IS NULL;
```

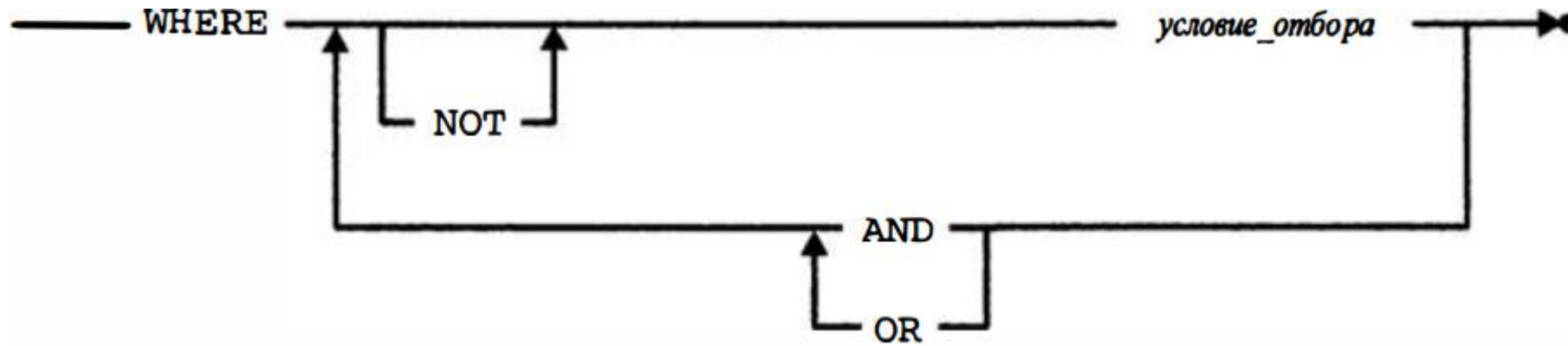
```
SELECT NAME  
FROM SALESREPS  
WHERE REP_OFFICE IS NOT NULL;
```

```
SELECT NAME  
FROM SALESREPS  
WHERE REP_OFFICE = NULL;
```

Составные условия отбора

3.2

- Применяются для создания сложных условий отбора



```
SELECT NAME, QUOTA, SALES
FROM SALESREPS
WHERE SALES < QUOTA
```

```
SELECT NAME, QUOTA, SALES
FROM SALESREPS
WHERE SALES < QUOTA
OR SALES < 300000.00;
```

```
SELECT NAME, QUOTA, SALES
FROM SALESREPS
WHERE SALES < QUOTA
AND SALES < 300000.00;
```

- Значения NULL влияют на интерпретацию составных условий отбора

Таблица истинности оператора AND

AND	TRUE	FALSE	NULL
TRUE	TRUE	FALSE	NULL
FALSE	FALSE	FALSE	FALSE
NULL	NULL	FALSE	NULL

Таблица истинности оператора OR

AND	TRUE	FALSE	NULL
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	NULL
NULL	TRUE	NULL	NULL

Таблица истинности оператора NOT

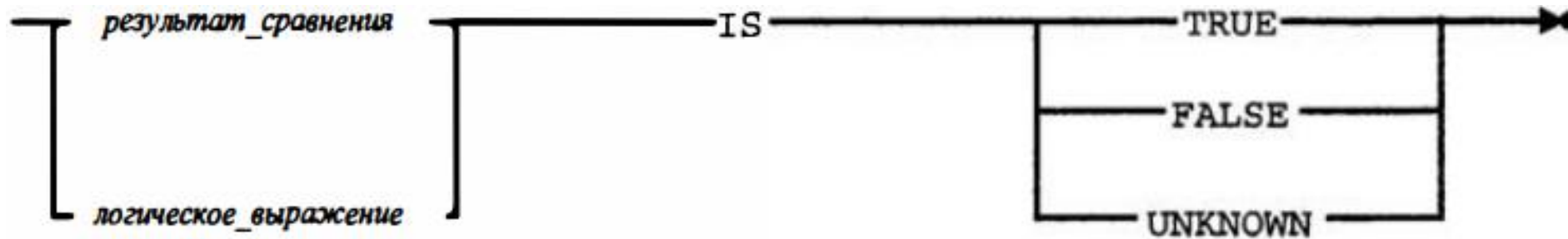
NOT	TRUE	FALSE	NULL
	FALSE	TRUE	NULL

- В соответствии со стандартом ANSI/ISO, если с помощью операторов AND, OR и NOT объединяется более двух условий отбора, то приоритеты выполнения операций следующие: NOT, AND, OR.
- Всегда следует использовать круглые скобки, что позволит устранить все возможные неоднозначности.

```
SELECT NAME
  FROM SALESREPS
 WHERE (REP_OFFICE IN (22, 11, 12))
        OR (MANAGER IS NULL AND HIRE_DATE >= '2006-06-01')
        OR (SALES > QUOTA AND NOT SALES > 600000.00);
```

Проверка IS

- Проверяет значение результата логического выражения



- Поиск строк, в которых нельзя выполнить сравнение из-за того, что либо столбец SALES, либо столбец QUOTA имеют значение NULL:

```
((SALES - QUOTA) > 10000.00) IS UNKNOWN
```


Предложение GROUP BY

4

Общие сведения

- Предложение инструкции SELECT, начинающееся с команды GROUP BY, предназначено для группирования результатов запроса
- Такой запрос формирует строки с промежуточными итогами
- Запрос, в котором присутствует конструкция GROUP BY, называется **группирующим запросом**
- Столбцы, перечисленные после GROUP BY, называются **группируемыми столбцами**

Исходная таблица

ДОГОВОР		
ФИО	ДАТА_ДОГОВОРА	СПЕЦИАЛЬНОСТЬ
Завгородний А.П.	12.11.2012	ИВТ
Светличная Е.И.	24.04.2011	БИ
Карандышев В.П.	01.02.2013	ФОИ
Иванов И.И.	17.09.1999	УК
Атанов Д.П.	23.02.2000	УК
Буряков С.М.	19.08.2012	ФОИ
Максименко А.Ю.	03.07.2015	БИ

SQL-запрос

```
SELECT СПЕЦИАЛЬНОСТЬ  
FROM ДОГОВОР;
```

Задача: узнать, по каким специальностям обучаются студенты

Результат запроса

СПЕЦИАЛЬНОСТЬ
ИВТ
БИ
ФОИ
УК
УК
ФОИ
БИ

В результате запроса могут присутствовать повторяющиеся записи



Список литературы

1. Современный SQL. URL: <https://modern-sql.com/>
2. Коннолли, Томас; Бегг, Каролин. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. 3-е издание. : Пер. с англ. — М. : Издательский дом "Вильямс", 2017. — 1440 с. : ил.
3. Дейт, К. Д. Введение в системы баз данных : Пер. с англ. / К. Д. Дейт. - 8-е изд. - Москва : Диалектика, 2019. - 1328 с. : ил.
4. Стандарты ISO. URL: <https://standards.iso.org/iso/>