

Лабораторная работа №9

Выборка данных из одной таблицы

1. Общая запись запроса на выборку

Целью любой системы управления базами данных является управление данными: ввод, изменение, удаление и выборка. Выборка данных – наиболее часто используемая команда управления данными. Выборка осуществляется с помощью команды **SELECT**.

Результатом выполнения данной команды является **виртуальная таблица** (итоговый набор данных), состоящая из **заголовков полей** и **записей данных**. Этот итоговый набор должен содержать не менее одного поля, а также ноль или более записей.

Общий синтаксис запроса **SELECT**:

```
SELECT [ ALL | DISTINCT ] СписокВыборки  
FROM ИмяТаблицы  
[WHERE УсловиеОтбора]  
[GROUP BY ИменаПолейГруппировки]  
[HAVING ОграничениеНаГруппы]  
[ORDER BY ИменаПолейСортировки {ASC | DESC}]  
[ LIMIT { n | ALL } ] [ OFFSET m ] ;
```

Если перевести команды на русский язык, то это будет примерно так:

SELECT [ALL DISTINCT] <i>СписокВыборки</i>	ВЫБРАТЬ [ВСЕ БЕЗ ПОВТОРЕНИЙ] <i>СписокВыборки</i>
FROM <i>ИмяТаблицы</i>	ИЗ <i>ИмяТаблицы</i>
[WHERE <i>УсловиеОтбора</i>]	[ГДЕ <i>УсловиеОтбора</i>]
[GROUP BY <i>ИменаПолейГруппировки</i>]	[ГРУППИРОВАТЬ ПО <i>ИменаПолейГруппировки</i>]
[HAVING <i>ОграничениеНаГруппы</i>]	[ФИЛЬТРОВАТЬ <i>УсловиеОтбораПоГруппе</i>]
[ORDER BY <i>ИменаПолейСортировки</i> { ASC DESC }]	[СОРТИРОВАТЬ ПО <i>ИменаПолейСортировки</i> { ПО ВОЗРАСТАНИЮ ПО УБЫВАНИЮ }]
[LIMIT { <i>n</i> ALL }] [OFFSET <i>m</i>]	[ПЕРВЫЕ <i>n</i> ВСЕ] [ПРОПУСТИТЬ <i>m</i> строк]

По умолчанию команда **SELECT** возвращает все записи, включая дубликаты, что определяется ключевым словом **ALL**, которое может быть опущено.

DISTINCT указывается для получения набора уникальных неповторяющихся записей.

FROM определяет имя таблицы, из которой отбираются записи.

После ключевого слова **WHERE** указывается условие, по которому будут отбираться записи. Этот раздел запроса необязателен, как и последующие.

После **GROUP BY** перечисляются поля, по которым производится группировка.

HAVING используется только при наличии группировки, после этого ключевого слова указывают ограничение на группы.

После **ORDER BY** перечисляются поля, по которым производится сортировка. По умолчанию сортировка осуществляется по возрастанию, что соответствует зарезервированному слову **ASC**, которое может опускаться, для сортировки в убывающем

порядке указывается – **DESC**. Вместо имен полей после **ORDER BY** можно указывать их порядковые номера в списке выборки или псевдонимы.

LIMIT предписывает выводить не все записи итогового набора, а не больше заданного числа строк. **OFFSET** указывает пропустить указанное число строк, прежде чем начать выдавать строки. Если указано и **OFFSET**, и **LIMIT**, сначала система пропускает **OFFSET** строк, а затем начинает подсчитывать строки для ограничения **LIMIT**.

Применяя **LIMIT**, важно использовать также предложение **ORDER BY**, чтобы строки результата выдавались в определённом порядке. Иначе будут возвращаться непредсказуемые подмножества строк.

Параметр **СписокВыборки** может содержать следующие элементы:

{* | ИмяПоля | Выражение} [[AS] Псевдоним]

Символ ***** указывается для выборки всех полей из таблицы.

ИмяПоля1, ИмяПоля2, ... – вывод указанных полей таблицы;

Выражение, по которому будут вычисляться значения **нового поля**, оно может содержать константы, имена полей, функции и их комбинации.

В одном списке выборки могут участвовать и имена полей и выражения.

[[AS] Псевдоним] – ключевое слово **AS** позволяет заменить в итоговом наборе данных обычные имена полей псевдонимами при необходимости. Псевдонимы особенно важно присваивать новым полям, вычисляемым с помощью выражений, в противном случае такие поля будут безымянными (вместо имени поля выводится надпись «?column?»). Имя псевдонима должно удовлетворять стандартным правилам именования объектов. При необходимости включить недопустимые символы, имя псевдонима обязательно заключается в кавычки.

Обратите внимание, что псевдонимы могут присваиваться как столбцам, так и таблицам.

Пример. Рассмотрим запрос для получения списка заказчиков с указанием адреса электронной почты из таблицы DimCustomer БД:

```
SELECT FirstName AS Имя,  
       LastName AS Фамилия,  
       EmailAddress AS "E-mail"  
FROM DimCustomer;
```

Поля данных будут представлены пользователю в порядке, определенном в списке выборки.

Пример. Запрос, выводящий список сотрудников с указанием идентификационного номера, а также фамилии и первого символа имени в одном поле:

```
SELECT EmployeeKey, LastName || ' ' || LEFT(FirstName,1) || '.' AS "Имя сотрудника"  
FROM DimEmployee;
```

Задание 1. Создайте запрос, выводящий фамилии, имена и отчества всех студентов с указанием их адреса, телефона и e-mail. Для полей задайте псевдонимы «Адрес проживания», «Телефон» и «Корпоративный e-mail» соответственно.

Задание 2. Создайте запрос, выводящий в алфавитном порядке общий список специальностей, которые имеются на факультете. Причем предусмотрите, чтобы одинаковые специальности выводились только по одному разу.

Задание 3. Создайте запрос, выводящий троих самых молодых преподавателей с указанием их даты рождения и возраста (количество лет). Предусмотрите, чтобы фамилия с инициалами имени и отчества выводились в одном поле.

2. Условия отбора

Условие отбора определяет критерий отбора записей, включаемых в итоговый набор. В результат будут включены только те строки, которые соответствуют наложенным условиям.

Условие может включать выражения, образованные с помощью операторов сравнения или логических операторов. Условия могут также объединяться и с помощью логических операндов AND, OR и NOT.

Операторы сравнения в условиях

Условие может включать выражения, образованные с помощью операторов сравнения:

=, >, <, <=, >=, !<, !>, <>, !=, BETWEEN.

Например, чтобы получить список сотрудников, дата рождения которых находится в диапазоне от 01.12.1989 г. до 31.12.1989 г., можно выполнить любой из следующих вариантов:

```
SELECT Фамилия, Имя, Отчество, ДатаРождения  
FROM Сотрудник  
WHERE (ДатаРождения >='01.12.1989') AND  
      (ДатаРождения <='15.12.1989');
```

```
SELECT Фамилия, Имя, Отчество, ДатаРождения  
FROM Сотрудник  
WHERE ДатаРождения BETWEEN '01.12.1989'  
      AND '31.12.1989';
```

Задание 4. Создайте запрос, отображающий информацию о преподавателях (ФИО, дату принятия на работу, должность, ученую степень), которые приняты на работу не ранее августа 2015 года.

Задание 5. Выведите список учебных групп, для которых не указан профиль, но указано направление подготовки.

Поиск по шаблону

Для поиска по шаблону символьных строк используется логический оператор **LIKE**, который обычно применяется в ситуациях, когда не известно точное совпадение.

В шаблоне могут использоваться следующие универсальные символы:

% – подразумевает любую строку, состоящую из 0 или более символов;

_ (или ?) – ровно один символ.

Пример:

Вывести товары, наименования которых содержат слово «бумага»:

```
SELECT *  
FROM DimProduct  
WHERE EnglishProductName LIKE '%бумага%';
```

Задание 6. Создайте запрос, выводящий список компьютерных кабинетов, расположенных в 14 и 16 корпусах.

Задание 7. Создайте запрос, выводящий в алфавитном порядке список факультативных дисциплин (в названии присутствуют сочетания «факультатив», «факультативная дисциплина» и т.п.), по которым предусмотрены лекции.

Определение соответствия значения одному из списка

Для определения соответствия выражения одному из перечисленных в заданном списке значений применяется логический оператор **IN**. Данный оператор всегда может быть записан и в виде группы условий, объединенных операндом **OR**.

Например, с помощью следующего запроса может быть получен список городов Германии и Франции из таблицы DimGeography:

```
SELECT City, EnglishCountryRegionName  
FROM DimGeography  
WHERE EnglishCountryRegionName IN ('Germany', 'France');
```

Задание 8. Создайте запрос, выводящий список дисциплин, ведущихся в нечетном семестре.

Задание 9.

А) Выведите список всех студентов из учебных групп определенного профиля, определяющих кодом учебной группы (например, МКб- или ФИб-).

Б) Определите три дисциплины с наибольшим общим объемом часов.

3. Аналитическая выборка данных (запросы с групповыми операциями)

Аналитическими запросами к БД принято называть запросы, сводные (агрегатные) результаты которых вычисляются над детальными данными, хранящимися в таблицах БД. В этом смысле любой запрос на языке SQL, результат которого основан на вычислении агрегатных функций, можно назвать аналитическим. Характерная особенность аналитических запросов состоит в том, что, как правило, они применяются к большим по объему базам данных, и выполнение таких запросов вызывает существенные накладные расходы СУБД.

Агрегатные функции

Запросы с групповыми операциями позволяют подсчитать общий результат по сгруппированному данным. Такие запросы используют агрегатные функции:

Avg ([all | distinct] *выражение*) – среднее арифметическое значений.

Count ([all | distinct] *выражение* | *) – количество значений в списке, отличных от NULL. При использовании символа * подсчитывается количество значений, включая значения NULL или повторяющиеся значения.

Sum ([all | distinct] *выражение*) – сумма всех значений списка.

Max ([all | distinct] *выражение*) – максимальное значение.

Min ([all | distinct] *выражение*) – минимальное значение.

Ключевое слово **all** предписывает выполнять агрегирование (группировку) всех записей в результирующем наборе данных, **distinct** – агрегирование только уникальных записей. По умолчанию используется **all**.

Пример. Вывод даты самой первой продажи:

SELECT min(Дата) FROM Продажа;

При выполнении агрегатной функции осуществляется объединение значений отдельного поля таблицы или части записей, после чего выполняется указанное агрегирование.

!!! Агрегатная функция (примененная без группировки записей) возвращает одно единственное значение, поэтому **использование других имен полей в списке выборки запрещено**.

Задание 10. Создайте запрос, выводящий дату принятия на работу преподавателя, работающего дольше всех на факультете.

Задание 11. Создайте запрос, выводящий общее количество студентов на факультете и их средний возраст (с точностью до одного знака после запятой). Назовите выводимые столбцы «Количество студентов» и «Средний возраст» соответственно.

Группировка записей

Чтобы получить агрегированные данные не по всем записям, а по группам, необходимо в запросе выполнить группировку. Для группировки записей по полям или выражениям применяется раздел **GROUP BY** оператора **SELECT**, что позволяет применять для каждой группы функции агрегирования.

В общем виде, при группировке по одному полю и вычислению только одного группового значения, запрос выглядит так:

```
SELECT ПолеГруппировки, агрегатная_функция(ПолеДляПодсчета)  
FROM Таблица  
GROUP BY ПолеГруппировки
```

Обратите внимание, что в списке выборки после **SELECT** указываются только **поля**, по которым группируются записи, и **поля**, по которым подсчитывается значение агрегатной функции. Других полей в этом списке быть не должно!

Например, чтобы определить количество сотрудников каждой должности, необходимо сгруппировать записи таблицы **Сотрудник** с одинаковыми значениями поля **Должность** и подсчитать количество записей в каждой группе:

```
SELECT Должность, count(*) AS "Количество сотрудников"  
FROM Сотрудник  
GROUP BY Должность;
```

При использовании GROUP BY для каждой определенной группы значений выводится только одна запись в итоговом наборе данных.

При группировке записей допускается также использование раздела WHERE, в этом случае группируются записи, удовлетворяющие этому условию.

Например, пусть в предыдущем примере надо подсчитать количество сотрудников мужского пола по должностям. Тогда в запрос добавится условие:

```
SELECT Должность, count(*) AS "Кол-во сотрудников мужского пола"  
FROM Сотрудник  
WHERE Пол='м'  
GROUP BY Должность;
```

Задание 12.

- А) Создайте запрос, подсчитывающий количество студентов в каждой учебной группе.
- Б) Создайте запрос, подсчитывающий количество «остепененных» преподавателей по видам ученых степеней. Данные по преподавателям «без степени» выводить не нужно.

Задание 13. Создайте запрос, выводящий средний балл по каждой дисциплине.

Задание 14. Создайте запрос, подсчитывающий в каждом корпусе количество кабинетов с проекторами вместимостью более 50 человек.

Задание 15.

- А) Создайте запрос, выводящий список студентов, у которых средний балл выше 4,4.
- Б) Создайте запрос, выводящий список дисциплин, по которым в прошлом учебном году средний балл оказался ниже 3,5.