

Лабораторная работа № 8 Создание SQL-запросов.

Что делать, если нам нужны студенты, которые учатся в 10 классе? Да, это просто, если в нашей БД только от 6 до 10 человек, но что, если их будет 6000? Без дополнительных инструментов узнать это будет сложно. Именно в этой ситуации к нам на помощь приходят SQL запросы, которые помогают изъять лишь необходимую информацию.

Виды запросов

SQL синтаксис реализует принцип **CRUD** (сокр. от англ. **create, read, update, delete** — «создать, прочесть, обновить, удалить»). Т.е. с помощью запросов вы сможете реализовать все эти функции.

Запросы на выборку данных (оператор SELECT).

Оператор **SELECT** (выбрать) является одним из основных операторов, применяемых в SQL. С помощью него осуществляется выборка данных из таблиц базы данных. Формат SQL-запроса с оператором **SELECT** выглядит следующим образом:

SELECT [**DISTINCT**] <список атрибутов>

FROM <список таблиц>

[**WHERE** <условие выборки>]

[**ORDER BY** <список атрибутов>]

[**GROUP BY** <список атрибутов>]

[**HAVING** <условие>]

[**UNION** <выражение с оператором **SELECT**>]

В квадратных скобках [] указаны элементы, которые могут отсутствовать в запросе. После слова **SELECT** через запятую перечисляются наименования полей (список атрибутов), содержимое которых запрашивается.

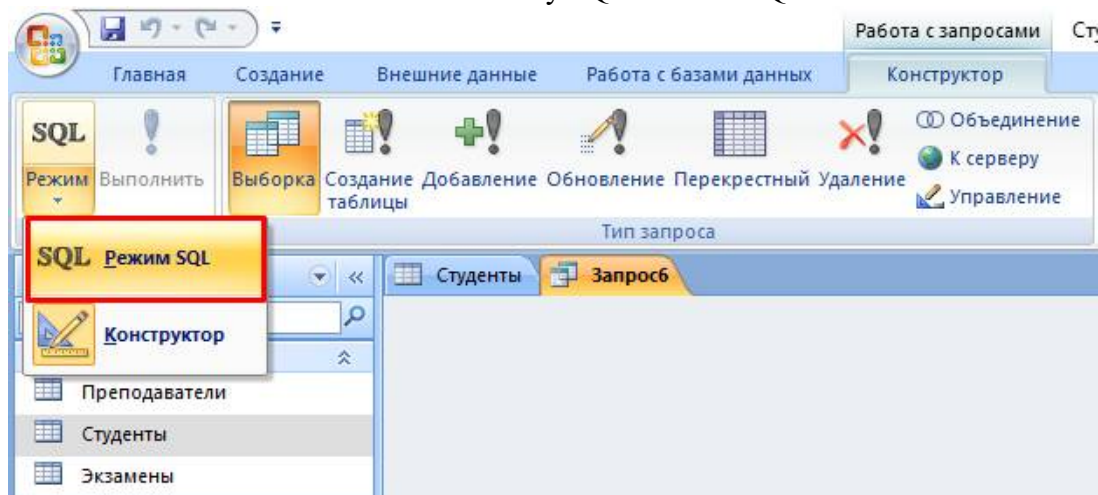
Обязательным ключевым словом в предложении-запросе **SELECT** является слово **FROM** (из). За ключевым словом **FROM** указывается список разделенных запятыми имен таблиц, из которых извлекается информация.

Ход работы:

Задание 1. Создайте запроса на выборку.

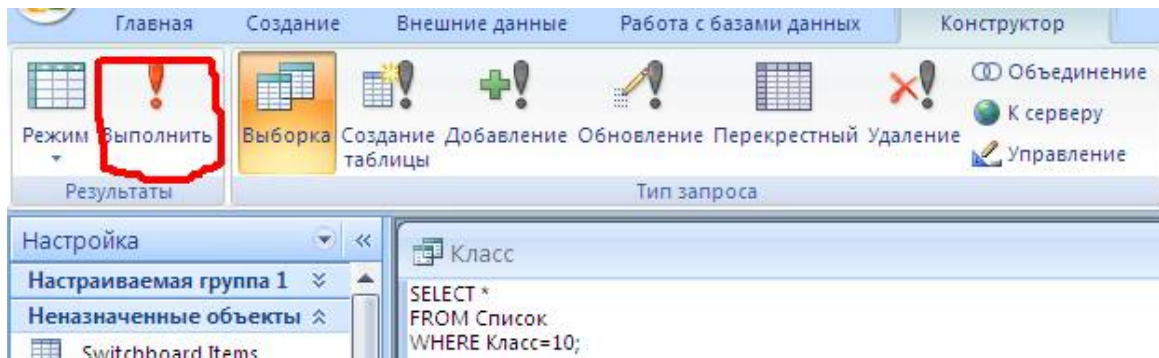
В этом случае в ход вступает принцип «прочесть». Например, нам нужно найти всех студентов, которые учатся в 10 классе. Для этого нужно:

- Перейдите во вкладку **Создание**.
- Нажмите кнопку **Конструктор запросов** в разделе **Другие**.
- В новом окне нажмите на кнопку **SQL/ Режим SQL**.

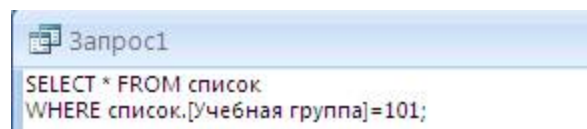


- В текстовое поле введите команду: **SELECT * FROM Список WHERE Класс=10;** где «**SELECT ***» означает, что выбираются все студенты, «**FROM Список**» — из какой таблицы, «**Класс = 10**» — условие, которое обязательно должно выполняться. Обязательным ключевым словом в предложении-запросе **SELECT** является слово **FROM** (из). За ключевым словом **FROM** указывается список разделенных запятыми имен таблиц, из которых извлекается информация.

SQL-Запрос ВСЕГДА должен оканчиваться точкой с запятой ; .



- Нажмите кнопку «Выполнить».
- Сохраните запрос под именем **Класс**.
- Создайте запрос на выборку студентов, которые обучаются в 101 группе.
- В текстовое поле введите команду: **SELECT * FROM Список WHERE Список.[Учебная группа]= 101;** (Поля содержащие недопустимые символы – пробел – заключаются в квадратные скобки).
- Нажмите кнопку «Выполнить». В таблице останутся только учащиеся 101 группы.
- Сохраните запрос под именем **Группа_101**.



- А что делать, если нас интересуют студенты из группы 102, которые младше 1983 года рождения? Тогда наш запрос будет выглядеть следующим образом:
- SELECT * FROM Список WHERE (((Список.[Учебная группа]) = 102) AND ((Список.[Год рождения]) > 1983));**
- Тогда результирующая таблица Список выдаст только учащихся 102 группы младше 1983 года рождения.
- Сохраните запрос под именем **Группа_102**.

Задание 2. Запрос на добавление таблицы.

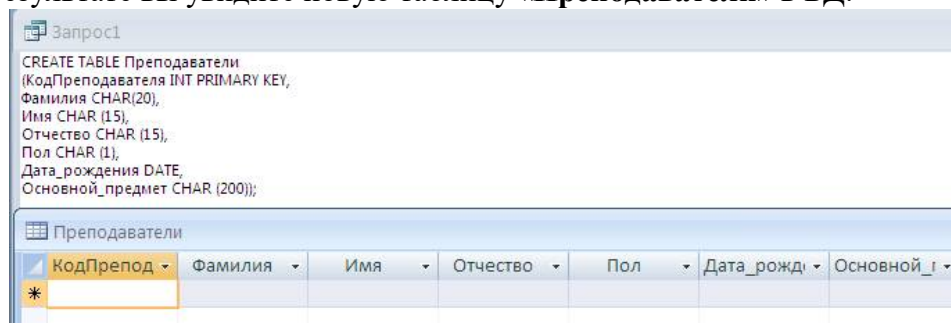
Кроме добавления таблицы с помощью встроенного конструктора, иногда может потребоваться выполнение этой операции с помощью SQL запроса. В большинстве случаев это нужно во время выполнения лабораторных или курсовых работ в рамках университетского курса, ведь в реальной жизни необходимости в этом нет. Если вы, конечно, не занимаетесь профессиональной разработкой приложений. Итак, для создания запроса нужно:

- Перейти во вкладку **Создание**.
- Нажать кнопку **Конструктор запросов** в разделе **Другие**.
- В новом окне нажмите на кнопку **SQL**, после чего в текстовое поле введите команду:

- **CREATE TABLE Преподаватели**
(КодПреподавателя INT PRIMARY KEY,
Фамилия CHAR(20),
Имя CHAR (15),
Отчество CHAR (15),
Пол CHAR (1),
Дата_рождения DATE,
Основной_предмет CHAR (200));

Команда «**PRIMARY KEY**» создает Ключевое поле в таблице «**Преподаватели**»; «**CREATE TABLE**» означает создание таблицы «**Преподаватели**», а «**CHAR**», «**DATE**» и «**INT**» - типы данных для соответствующих значений: «**INT**» - означает тип данных поля - целое число, «**CHAR**» - символьный тип данных с количеством выделенных символов, т. е. **CHAR(15)** выделяет поле, которое должно содержать слово не более 15 букв; «**DATE**»- тип данных дата.

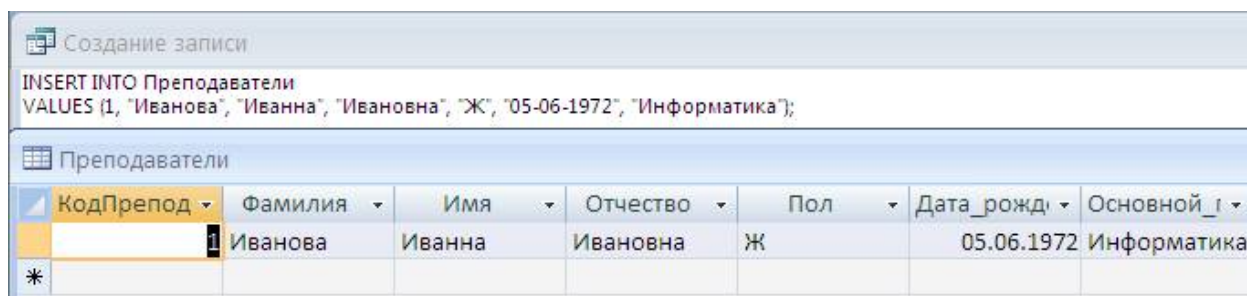
- Кликните по кнопке «Выполнить».
- Откройте созданную таблицу. Сохраните запрос под именем **Создание_таблицы**.
- В результате вы увидите новую таблицу «**Преподаватели**» в БД:



Задание 3. Запрос на добавление, удаление и редактирование записей в таблице.

Снова перейдите в поле для создания запроса и введите следующие команды:

- для заполнения кортежа (кавычки должны быть на английской раскладке):
INSERT INTO Преподаватели
VALUES (1, "Иванова", "Иванна", "Ивановна", "Ж", "05-06-1972",
"Информатика");



- Сохраните запрос под именем **Запись1**.
- для редактирования записи создайте запрос:
UPDATE Преподаватели SET Имя = "Анна";
- Самостоятельно добавьте в таблицу **Преподаватели** 3 новых записи для преподавателей. Сохраните запросы **Запись2, Запись3, Запись4**.

- **Добавьте новое поле в таблицу Преподаватели:** Создайте запрос на добавление поля с помощью команды **ALTER TABLE**. Она может добавлять столбцы к таблице, удалять столбцы, изменять их размеры, а также добавлять или удалять ограничения. Эта команда не часть стандарта ANSI, поэтому в разных системах она имеет разные возможности.

- Типичный синтаксис чтобы добавить столбец к таблице:

**ALTER TABLE <table name> ADD <column name>
<data type> <size>;**

Например:

ALTER TABLE Преподаватели ADD Часы INT – добавляет в таблицу **Преподаватели** новый столбец **Часы** (Количество отведенных преподавателем академических часов по дисциплине) целочисленного типа.

- Посмотрите изменения в таблице Преподаватели. Сохраните запрос под именем **ПолеЧасы**.
- Самостоятельно задайте запрос на добавление столбца **Оплата** (Оплата 1 академ. часа преподавателя в рублях) целочисленного типа. Сохраните запрос под именем **ПолеОплата**.
- Убедитесь, что произошло добавление столбцов в таблицу **Преподаватели** и заполните их необходимыми данными.
- Для удаления записей из таблицы создайте запрос:
DELETE * FROM Преподаватели WHERE Имя="Анна";
- Убедитесь, что запись о преподавателе Ивановой Анне Ивановне удалена.

Задание 4. Создание запросов на объединение записей.

Этот тип запроса позволяет объединить в одном результирующем наборе результаты нескольких запросов, таблиц и инструкций SELECT.

- Создадим запрос на объединение данных, представленных в двух таблицах.

Например, нам необходимо посмотреть оценки учащихся всех групп по текстовому редактору Writer.

- Тогда нам нужны две таблицы: **Список** (поля **Фамилия, Имя, Отчество**) и таблица **Личные данные** (поле **Writer**).
- Создадим новый SQL-запрос:

**SELECT список.Фамилия, список.Имя, список.Отчество, [Личные данные].Writer
FROM список
INNER JOIN [Личные данные]
ON список.код = [Личные данные].кодСтудента;**

Как можно видеть, поля таблицы представлены названием таблицы и названием поля, разделенные точкой. Если таблица или поле содержит запрещенные символы (пробел), то поле или таблица представляются в квадратных скобках.

Команда **INNER JOIN** означает объединение данных из двух таблиц. Команда **ON** устанавливает связи между таблицами.

- Нажмите кнопку **Выполнить**
- Сохраните запрос с именем **ОценкиWriter**.
- Добавим сортировку по полю **Фамилия (по возрастанию)**. Команда **ORDER BY**.
- Выберем тех студентов, оценки которых только **4** или **5**. Команда **WHERE**.

- Нажмите кнопку **Выполнить**.

```
Оценки Writer
SELECT список.Фамилия, список.Имя, список.Отчество, [Личные данные].Writer
FROM список INNER JOIN [Личные данные] ON список.код=[Личные данные].кодСтудента
WHERE ((([Личные данные].Writer)=4 Or ([Личные данные].Writer)=5))
ORDER BY список.Фамилия;
```

Результат выполнения запроса:

Фамилия	Имя	Отчество	Writer
Терещенко	Инна	Станиславовн	5
Фролова	Марина	Викторовна	5
Чернова	Кристина	Ивановна	4

- Самостоятельно создайте запросы по учащимся, (по возрастанию поля Отчество) чьи оценки по **Calc** не более 3.
- Самостоятельно создайте запрос по учащимся, установите сортировку по **Именам** (по убыванию) и которые получили оценки по **Access** только 5.
- Сохраните запросы под именами **ОценкиCalc** и **ОценкиAccess**.

Инструкция **ASC** используется как дополнение к инструкции **ORDER BY** и служит для определения сортировки по возрастанию. Инструкция **DESC** используется как дополнение к инструкции **ORDER BY** и служит для определения сортировки по убыванию. В случае, когда ни **ASC**, ни **DESC** не указаны, подразумевается наличие **ASC**. Общий вид команды: **ORDER BY ИмяПоля DESC**;

Задание 5. Создайте подчиненный запрос.

Некоторые подчиненные запросы можно использовать в перекрестных запросах как предикаты в предложении **WHERE**. Подчиненные запросы, применяемые для вывода результатов (в списке **SELECT**), нельзя применять в перекрестных запросах. В отличие от запросов на объединение, подчиненный запрос можно создать в режиме **Конструктора**. В строке **Условия отбора** в качестве условия отбора следует указать, например, зарезервированное слово **IN**, а затем ввести инструкцию **SELECT**. А можно сначала создать в режиме **Конструктора** подчиненный запрос, переключиться в режим **SQL**, скопировать получившуюся инструкцию **SQL** в буфер обмена, а затем создать главный запрос и скопировать в строку **Условия отбора** подчиненный запрос из буфера.

Создадим вложенный запрос, отображающий информацию об учащихся, которые учатся у определенного преподавателя (определим это по фамилии преподавателя).

- Попробуем реализовать это через вложенный запрос. Например, мы знаем Фамилию преподавателя, но не знаем, в какой группе он ведет занятия. Для начала определим номер группы, если нам известна фамилия преподавателя, например, Белоусов А.И.. Создадим соответствующий SQL-запрос:

SELECT [Учебная группа] FROM Группы
WHERE Преподаватель = "Белоусов А.И.";

- Убедитесь, что запрос работает, и вы видите номер группы - **102**.
- Вернитесь в SQL-запрос. Теперь добавим список учеников выбранной группы из таблицы **Список**. Полностью запрос будет иметь вид:

```
Номер группы
SELECT *
FROM Список
WHERE [Учебная группа] =
(SELECT [Учебная группа] FROM Группы
WHERE Преподаватель = "Белоусов А.И.");
```

- Сохраните запрос под именем **Белоусов**.
- Самостоятельно создайте запрос для преподавателя Новиковой Е.В. Сохраните запрос под именем **Новикова**. Список какой группы вы увидите при выполнении этого запроса?

Задание 6. Создайте вычисляемый запрос.

Такие функции как **SUM** - сумма, **COUNT** - количество, **AVG** – среднее арифметическое значение, **MAX** – максимальное значение, **MIN** – минимальное значение называются **агрегатными функциями**. Они принимают множество значений, и после их обработки возвращают единственное значение.

- Например, для подсчета суммы произведения полей **Часы** и **Оплата** таблицы **Преподаватели** при помощи агрегатной функции **SUM** создайте запрос вида:

```
SELECT SUM(Часы*Оплата) AS Общая_Сумма  
FROM Преподаватели;
```

- В результате вы получите только одно значение – Общую сумму.
- Нажмите кнопку **Выполнить**. Сохраните запрос под именем **Общая сумма**.
- Или можно подсчитать Сумму к выплате для какого-либо преподавателя, например, по его коду в таблице, если указать условие:

```
SELECT SUM(Часы*Оплата) AS Выплата  
FROM Преподаватели  
WHERE КодПреподавателя = 3;
```

- Выполните запрос и сохраните под именем **Выплата1**.
- Самостоятельно создайте запрос по Выплате одного из преподавателей, указав его **Фамилию** в условии. Выполните запрос и сохраните как **Выплата2**.
- Возможно использование нескольких агрегатных функций в запросе. Например, подсчитаем среднюю оценку учащихся по каждому из предметов и определим количество учащихся, сдавших экзамены:

```
SELECT Avg(Writer) AS AvgWriter, Avg(Calc) AS AvgCalc,  
Avg(Access) AS AvgAccess, Count(*) AS ЧислоСтудентов  
FROM [Личные данные];
```

- Выполните и сохраните запрос под именем **СредняяОценка**.

Самостоятельно.

1. Создайте запрос для подсчета количества пятерок по дисциплине **Calc**. Выполните и сохраните запрос под именем **Calc_5**.
2. Попробуйте изменить запрос, чтобы средняя оценка рассчитывалась для конкретного учащегося по его коду. Выполните и сохраните запрос под именем **ОценкаУчащегося**.