

Лабораторная работа № 10

Тема: Реализация запросов в Access

Подтема: Создание и использование запросов SQL в Access

Цель: Ознакомление с объектом базы данных **Запрос**.

Теория

Запрос SQL – это запрос Access, который нельзя создать с помощью конструктора или мастера. Эти запросы могут быть созданы только с помощью языка SQL. К таким запросам относятся подчиненные запросы, запросы к серверу, запросы – объединения и управляющие запросы.

SQL состоит из инструкций, которые передаются программе, управляющей работой реляционной базы данных, предлагая ей выполнить определенные действия. Эти инструкции иногда называют предложениями.

Для создания запроса SQL надо выбрать в окне **База данных** вкладку **Запросы**, а затем выбрать **«Создание запроса в режиме конструктора»**.

Окно **Добавление таблицы** закрыть. На панели инструментов конструктора запросов щелкнуть по кнопке **SQL** или выбрать команду **Запрос – Запрос SQL**. Далее выбирается вид запроса: объединение, к серверу, управление. На экран будет выведено окно для ввода инструкций языка SQL. После написания инструкций запрос выполняется (кнопка **«Запуск»**) и сохраняется.

Примеры использования запросов SQL:

1. Выборка данных из таблиц.

В запросе SQL можно осуществлять выборку данных из таблиц по различным критериям (например, по классу). Но запрос – выборку нетрудно создать и в режиме **Конструктора запросов**, которым можно воспользоваться без знания инструкций SQL.

Простейшая инструкция поиска информации имеет вид:

```
SELECT <Список полей>  
FROM <Имена таблиц>;
```

Select – ключевое слово, оно сообщает базе данных, что инструкция SQL является запросом – выборкой.

<Список полей> – перечисление полей через запятую, которые надо вывести в новую таблицу. Символ * – на этом месте обозначает, что будут выбраны все поля указанных в предложении FROM таблиц.

From – всегда присутствует после слова Select и определяет, какие таблицы или запросы содержат поля, приведенные в инструкции Select.

Любая инструкция языка SQL оканчивается «;», что означает завершение запроса SQL.

2. Агрегирование данных.

Любую группу записей можно обобщить по какому-либо полю. Существуют специальные функции агрегирования, например:

Avg	–	среднее значение поля
Count	–	количество записей в таблице, удовлетворяющих условиям
Max	–	максимальное значение поля
Min	–	минимальное значение поля
Sum	–	суммирование численных значений поля

Пример 1. Из таблицы **Ведомость оценок** найти средний балл по предмету **«Алгебра»** и вывести его в графу **«Средний балл»**.

```
SELECT Avg([Оценка за год]) AS [Средний балл]
```

```
FROM [Ведомость оценок]
WHERE [Название предмета]="Алгебра";
```

Инструкция WHERE задает условие для отбора записей в таблице. В предложении WHERE можно использовать операции отношения (=, >, <, >=, <=, <>) и логические операторы (and, or, not).

Результат выполнения запроса приведен на рис. 1.

Средний балл
4

Рис.1. Результат выполнения запроса Средний балл по алгебре

3. Соединение таблиц.

Одной из важнейших черт запросов SQL является возможность определить связь между таблицами и работать с этими таблицами, как с единым целым. Операция соединения таблиц – самая мощная операция реляционных баз данных.

Пример 2. Даны две таблицы Ученик и Учитель. Найти классного руководителя каждого ученика.

Связь между таблицами устанавливается с помощью запятой при их перечислении после параметра FROM.

```
SELECT Фамилия, Имя, Ученик.Класс, Ученик.Буква, Учитель.[Фамилия, имя, отчество]
```

```
FROM Ученик, Учитель
```

```
WHERE Ученик.Класс=Учитель.Класс AND Ученик.Буква=Учитель.Буква;
```

Результат выполнения запроса приведен на рис. 2.

Фамилия	Имя	Класс	Буква	Фамилия, имя, отчество
Андреева	Наталья	10 а		Яковлева Нина Михайловна
Филиппов	Николай	10 а		Яковлева Нина Михайловна
Борисов	Михаил	10 а		Яковлева Нина Михайловна
Крылова	Анастасия	10 б		Смирнова Ирина Павловна

Рис.2. Результат выполнения запроса Соединение таблиц Ученик и учитель

4. Ввод, удаление, изменение значений полей.

Рассмотрим три команды обновления:

INSERT – вставить;

UPDATE – обновить;

DELETE – удалить.

4.1. Ввод значений в таблицу. Команда INSERT INTO

Пример 3. В таблицу **Список предметов** нужно внести сведения о новом предмете (одну новую запись).

```
INSERT INTO [Список предметов]
```

```
SELECT «Обществознание»;
```

В результате выполнения запроса в таблицу **Список предметов** будет добавлена новая запись.

Команду INSERT обычно используют для того, чтобы из одной таблицы взять данные и перенести их в другую таблицу.

Пример 4. Пусть существует таблица **Ученики 10а класса**, структура которой тождественна структуре таблицы **Ученик**. Из таблицы **Ученик** надо выбрать всех учеников 10а класса и перенести эти записи в пустую таблицу **Ученики 10а класса**.

```
INSERT INTO [Ученики 10а класс]
SELECT *
FROM Ученик
WHERE Класс=10 and Буква="а";
```

4.2. Исключение строк из таблицы. Команда DELETE

Команда DELETE не может удалять отдельные поля. Она удаляет только выбранные записи целиком.

Пример 5. Удалить из таблицы **Ученик** всех учеников 10б класса.

```
DELETE *
FROM Ученик
WHERE Класс=10 and Буква="б";
```

4.3. Обновление значений полей. Команда UPDATE.

Пример 6. Переименовать название кружка *Компьютерная графика*, в котором занимаются ученики, на название *Изучение PhotoShop*.

```
UPDATE Кружки
SET [Название кружка]="Компьютерная графика"
WHERE [Название кружка]="Изучение PhotoShop";
```

Задание

На основании приведенных выше примеров создать следующие запросы SQL:

1. Запрос на удаление из таблицы **Ученик** всех учащихся 11 класса.
2. Запрос на изменение номера класса всех учащихся (перевод всех учащихся в следующий класс) в таблице **Ученик**.
3. Запрос на удаление всех записей в таблице **Кружки**.
4. Запрос на создание резервной копии таблицы **Ученик** (для выполнения этого запроса создается новая таблица **Резервная копия** со структурой, аналогичной структуре таблицы **Ученик**).