

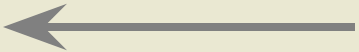
	поле 1	поле 2	поле 3
запись 1			
запись 2			
запись 3			
. . .			
. . .			
. . .			
. . .			
. . .			
запись N			



	Ф	И	О



	тел	адрес	рожд



**БАЗЫ  
ДАННЫХ - 5**

1. Введение в SQL
2. Основные предложения SQL и примеры их использования

Вопрос 1

**Введение в SQL**

**Язык SQL (Structured Query Language)** – структурированный язык запросов, ориентированный на операции с данными, представленными в виде логически взаимосвязанных таблиц.

---

**Особенности SQL** заключаются в том, что этот язык ориентирован на **конечный результат**, т.е. нам не нужно знать как происходит обработка нашего запроса в БД, достаточно **правильной формулировки запроса**, а как его выполнять, какие действия и в каком порядке производить – **«укажет» SQL**

---

**Здесь можно привести такое сравнение:**

Мы пришли в магазин за покупками, какими именно – объясняем продавцу, или менеджеру торгового зала – т.е. **делаем запрос**. Остальное нас не касается: специалисты магазина знают, где размещены необходимые товары, какие выполнить расчеты, во что завернуть ... и мы получаем **конечный результат**.

Примерно по такому же принципу работает SQL. Наша задача - корректно сформулировать запрос

## Немного о развитии языка:

---

Рождение языков запросов связано с созданием Коддом теории **реляционных баз данных** ( языки ISBL, SQL, QBE, PIQUE, QUEL)

В начале 80-х годов **SQL** "победил" другие языки запросов и стал фактическим стандартом таких языков для профессиональных реляционных СУБД.

В 1987 году **SQL** был признан **международным стандартом языка баз данных** и начал внедряться во все распространенные СУБД персональных компьютеров.

---

Наиболее широко применяется **SQL** в **клиент – серверных приложениях**, где клиент посылает SQL – запрос к серверу, который его обрабатывает и возвращает клиенту результат

Поэтому все современные версии профессиональных реляционных СУБД (DB2, Oracle, Ingres, Informix, Sybase, Progress, Rdb) используют технологию "Клиент-Сервер" и язык **SQL**.

Кроме того, многие локальные СУБД, в том числе **MS ACCESS**, имеют возможности работы с SQL

Итак, **SQL** – простой и компактный язык указаний по работе с базой данных, называемых **предложениями SQL** (которых около 30)

---

**Основными предложениями SQL являются:**

- ☐ создание и уничтожение таблиц
  - ☐ запрос на выборку данных
  - ☐ модификация данных (добавление, удаление, изменение)
  - ☐ управление данными (привилегии, доступ)
  - ☐ вычисления по условиям
  - ☐ упорядочение, группирование и т.д.
- 

Рассмотрим на примерах создание предложений SQL (естественно, учитывая время на изучения SQL в школе, ограничимся знакомством с самыми простыми примерами)

Вопрос 2

# Основные предложения SQL и примеры их использования

Для создания запросов SQL нужна таблица, поэтому начнем с ее создания

---

Для создания таблицы служит предложение :  
**CREATE TABLE** имя\_таблицы, после которого в скобках указываются названия полей и типы данных через запятую:

---

Например:

```
CREATE TABLE ПРОДУКТЫ  
( ПРОДУКТ CHAR(20),  
БЕЛКИ INT,  
ЖИРЫ INT,  
УГЛЕВОДЫ INT,  
КАЛОРИЙНОСТЬ INT )
```

При выполнении такого запроса будет создана таблица ПРОДУКТЫ с полями:

**ПРОДУКТ CHAR(20)** - поле ПРОДУКТ, тип текстовый, длиной до 20 символов

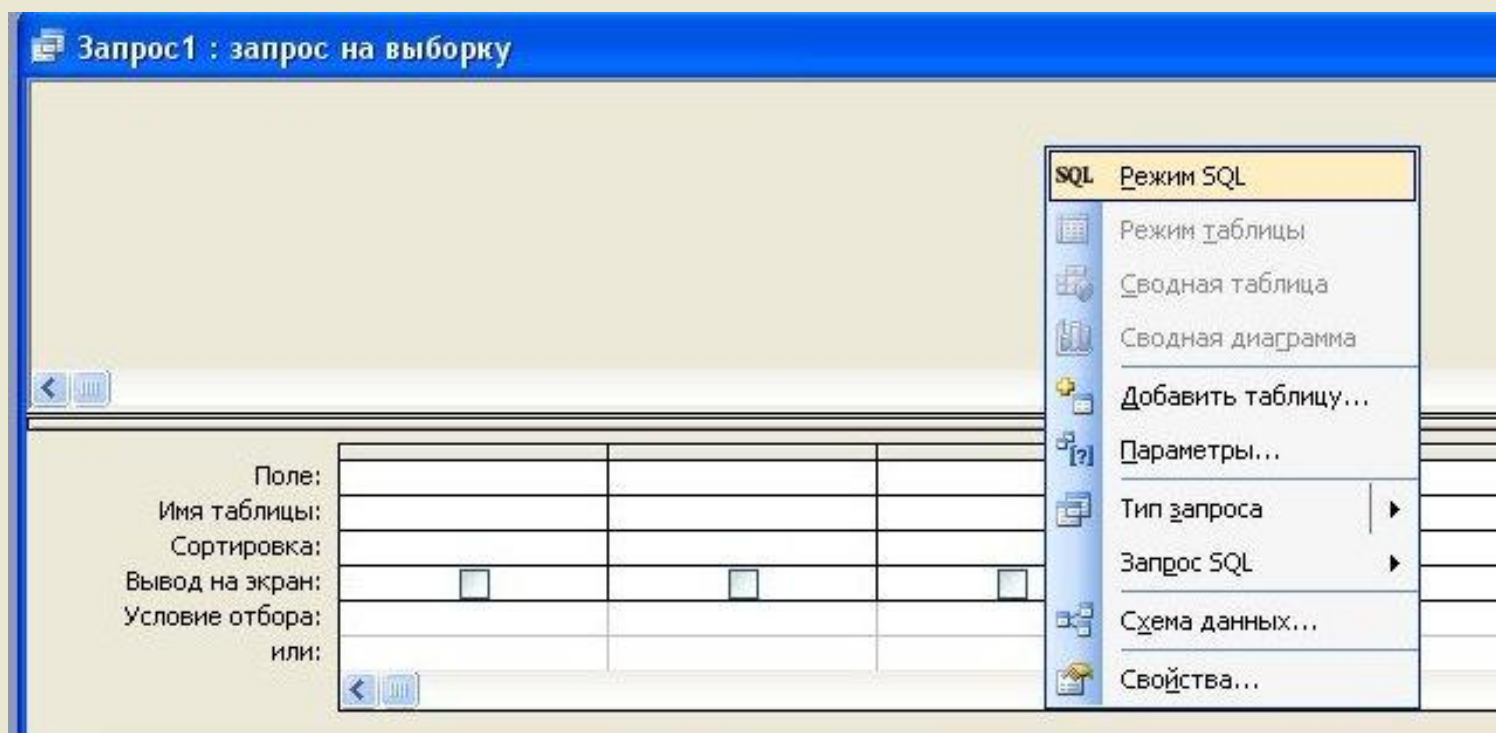
**БЕЛКИ INT** поле БЕЛКИ с целым типом данных (Integer)

**ЖИРЫ INT, УГЛЕВОДЫ INT, КАЛОРИЙНОСТЬ INT** – это тоже понятно

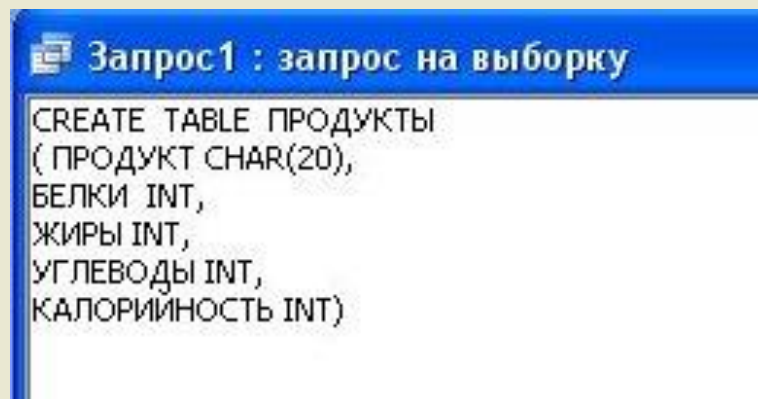


## Сделаем это в Access:

1. Откроем Access: Файл – создать – Новая база данных – назовем ее «SQL»
2. На вкладке ЗАПРОСЫ выберем СОЗДАТЬ В РЕЖИМЕ КОНСТРУКТОРА, окно добавления таблицы закроем, щелкнем правой по полю запроса и выберем РЕЖИМ SQL



### 3. Напечатаем наш запрос



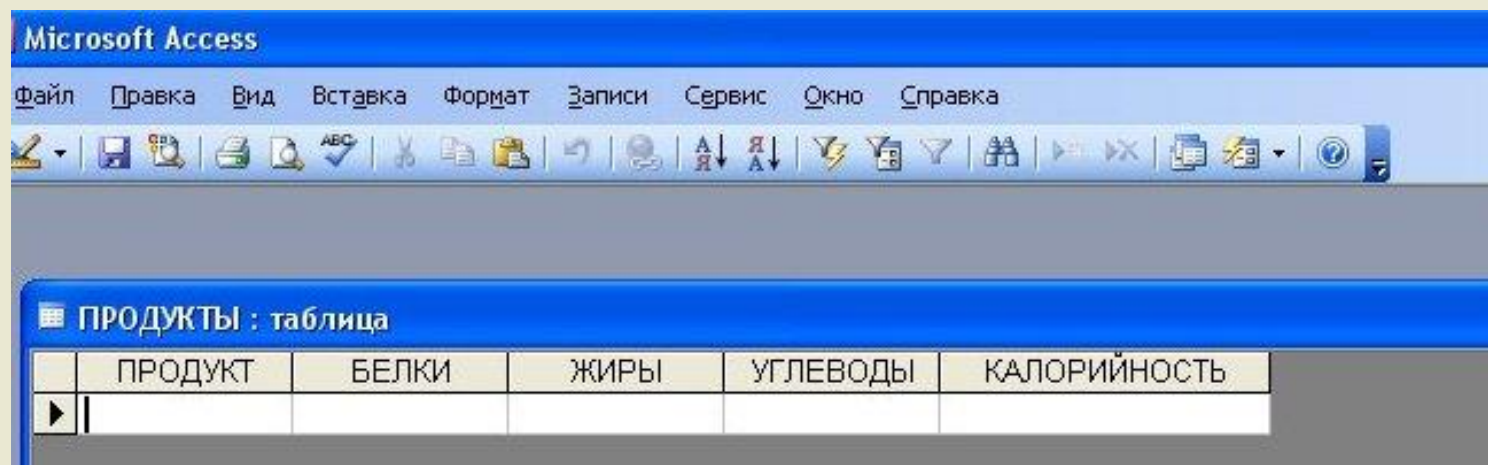
```
CREATE TABLE ПРОДУКТЫ  
( ПРОДУКТ CHAR(20),  
БЕЛКИ INT,  
ЖИРЫ INT,  
УГЛЕВОДЫ INT,  
КАЛОРИЙНОСТЬ INT)
```

### 4. Закроем окно запроса и сохраним его

### 5. Наш запрос появился на вкладке базы ЗАПРОСЫ. Запустим его.

### 6. Перейдем на вкладку ТАБЛИЦЫ и мы видим, что там появилась таблица ПРОДУКТЫ

### 7. Откроем таблицу и перед нами таблица с заданными полями – постарался SQL:



Microsoft Access

Файл Правка Вид Вставка Формат Записи Сервис Окно Справка

ПРОДУКТЫ : таблица

	ПРОДУКТ	БЕЛКИ	ЖИРЫ	УГЛЕВОДЫ	КАЛОРИЙНОСТЬ
▶					

Сейчас надо внести данные в таблицу

---

Для вставки данных в таблицу служит предложение **INSERT**

---

Напишем SQL – запрос на добавление записи:

```
INSERT  
INTO ПРОДУКТЫ (ПРОДУКТ,БЕЛКИ, ЖИРЫ, УГЛЕВОДЫ, КАЛОРИЙНОСТЬ)  
VALUES (ПРОДУКТ,БЕЛКИ, ЖИРЫ, УГЛЕВОДЫ, КАЛОРИЙНОСТЬ)
```

---

Здесь:

**INSERT** – оператор добавления записи в таблицу

**INTO ПРОДУКТЫ (ПРОДУКТ,БЕЛКИ, ЖИРЫ, УГЛЕВОДЫ, КАЛОРИЙНОСТЬ)** – показываем в какую таблицу и в какие поля добавляются данные

**VALUES (ПРОДУКТ,БЕЛКИ, ЖИРЫ, УГЛЕВОДЫ, КАЛОРИЙНОСТЬ)** – здесь будут запрашиваться значения полей при вводе данных

Сделаем этот запрос в Access:

---

Для этого создаем запрос и переходим в режим SQL (как в предыдущем примере) и печатаем текст запроса, сохраняем.

А сейчас запустим запрос и на предложение ввести данные введем следующее:



**ПРОДУКТ – мясо говядина**  
**БЕЛКИ - 16**  
**ЖИРЫ – 4**  
**УГЛЕВОДЫ - 0**  
**КАЛОРИЙНОСТЬ - 108**

---

Откроем таблицу ПРОДУКТЫ и видим, что появилась запись с нашими данными

Microsoft Access

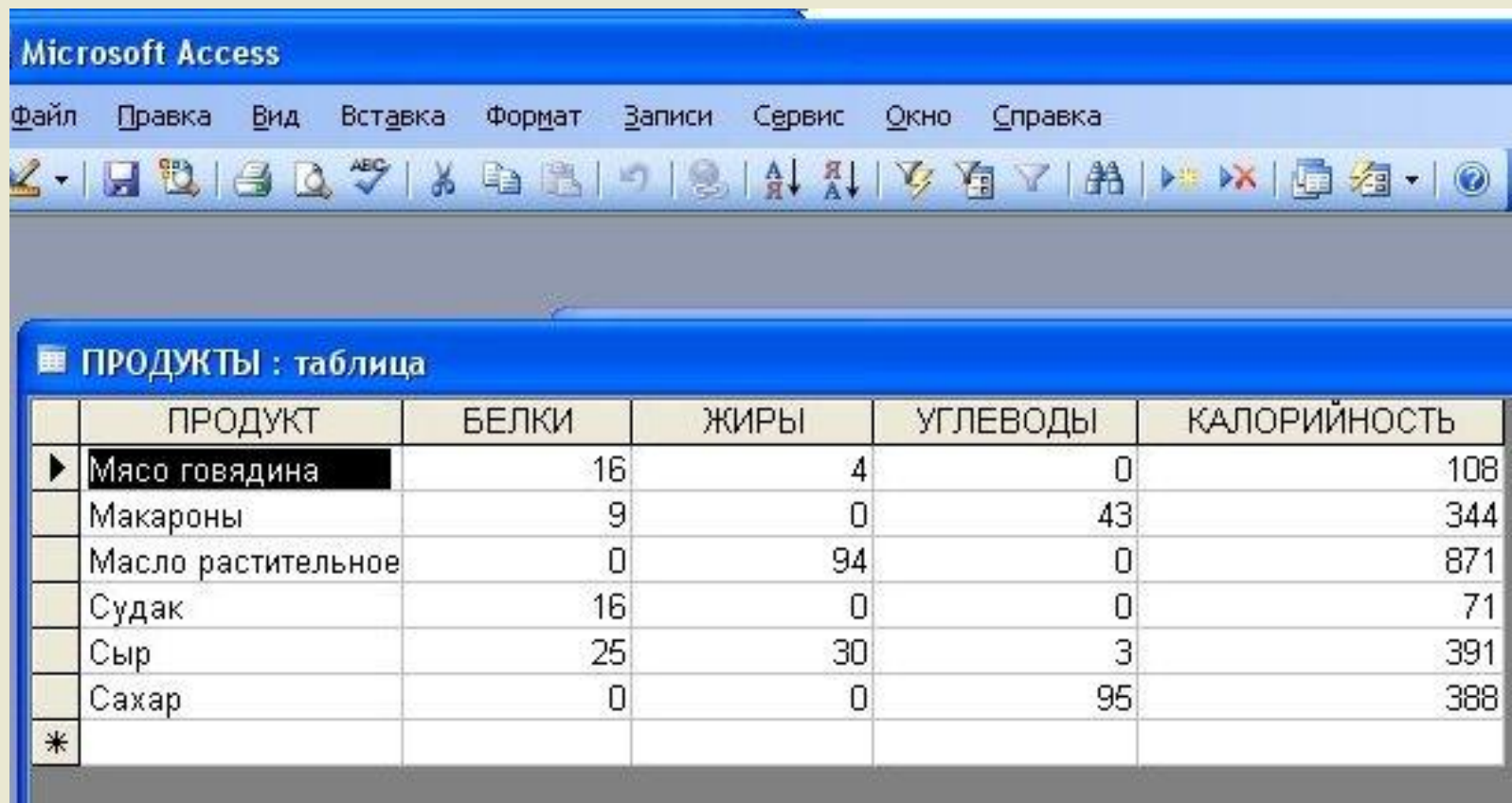
Файл Правка Вид Вставка Формат Записи Сервис Окно Справка

ПРОДУКТЫ : таблица

	ПРОДУКТ	БЕЛКИ	ЖИРЫ	УГЛЕВОДЫ	КАЛОРИЙНОСТЬ
▶	Мясо говядина	16	4	0	108
*					

А сейчас самостоятельно, используя этот запрос на добавление данных, внесите в таблицу следующие данные:

( Они нужны нам для рассмотрения следующего **предложения SELECT** )



Microsoft Access

Файл Правка Вид Вставка Формат Записи Сервис Окно Справка

ПРОДУКТЫ : таблица

	ПРОДУКТ	БЕЛКИ	ЖИРЫ	УГЛЕВОДЫ	КАЛОРИЙНОСТЬ
►	Мясо говядина	16	4	0	108
	Макароны	9	0	43	344
	Масло растительное	0	94	0	871
	Судак	16	0	0	71
	Сыр	25	30	3	391
	Сахар	0	0	95	388
*					

### Следующее предложение SQL - **SELECT**

**SELECT** – это запрос на выборку данных, обычно по условию или с вычислениями

**SELECT** – самая распространенная инструкция SQL

---

### Структура запроса **SELECT**:

**SELECT** – здесь указываются имена полей, из которых производится выборка

**FROM** – здесь указывается таблица (таблицы), из которых производится выборка

**WHERE** – здесь указывается условие (условия) выборки

При формировании запроса мы можем объединять условия с помощью AND (и), OR (или), NOT (не)

Организуем следующий запрос к нашей базе:

**Найти продукты, не содержащие жиров и углеводов**

---

Запрос выглядит так:

```
SELECT ПРОДУКТ, ЖИРЫ, УГЛЕВОДЫ  
FROM ПРОДУКТЫ  
WHERE ЖИРЫ = 0 AND УГЛЕВОДЫ = 0
```

Сформируйте запрос, запустите его и Вы должны получить следующее:



	ПРОДУКТ	ЖИРЫ	УГЛЕВОДЫ
▶	Судак	0	0
*			

**В нашей базе только судак не содержит жиров и углеводов**

**А сейчас попробуйте организовать самостоятельно запросы**

---

- 1. Вывести перечень продуктов с указанием содержания в них белков**
  - 2. В каких продуктах содержание белка больше 16, а углеводов меньше 10**
  - 3. Есть ли продукты, не содержащие белков и жиров**
  - 4. В каком из продуктов содержание жиров максимально**
  - 5. Какой продукт самый калорийный**
-



Итак, мы создали таблицу, поработали с ней, пришла пора удалить ее

---

Для уничтожения таблицы служит оператор **DROP TABLE**

Создадим запрос:

**DROP TABLE ПРОДУКТЫ**

Запустите запрос, затем посмотрите на вкладке ТАБЛИЦЫ – ее там уже нет

---

### **ЗАКЛЮЧЕНИЕ:**

Мы познакомились с элементарными конструкциями SQL, не рассмотрев еще очень много возможностей языка

Для продолжения изучения языка к презентации приложен учебник по SQL, также можно посетить сайт [www.sql.ru](http://www.sql.ru), где можно найти много полезного и интересного