

ОСНОВЫ ЯЗЫКА SQL

1. Общая характеристика языка
2. Основные операторы языка SQL MS ACCESS

1. Общая характеристика языка

Типы команд стандарта ANSI SQL:

- команды языка определения данных (Data Definition Language – DDL)
- команды языка обработки данных (Data Manipulation Language – DML)
- команда языка запросов данных (Data Query Language – DQL)
- команды управления данными (Data Control Language – DCL)
- команды языка обработки транзакций (Transaction processing Language – TPL)
- команды языка управления курсором (Cursor Control Language – CCL)

2. Основные операторы языка SQL MS ACCESS

<i>Подмножество языка</i>	<i>Команда</i>	<i>Назначение</i>
DDL	CREATE TABLE	Создать таблицу
	DROP TABLE	Удалить таблицу
	ALTER TABLE	Изменение структуры таблицы
	CREATE INDEX	Создать индекс
	ALTER INDEX	Модифицировать индекс
	DROP INDEX	Удалить индекс
	CREATE VIEW	Создать представление
	DROP VIEW	Удалить представление
DML	SELECT	Выборка записей
	UPDATE	Изменение записей
	INSERT	Вставка новых записей
	DELETE	Удаление записей

Запрос MS ACCESS →

В режиме **Конструктора**
SQL

В режиме **Таблицы**

Запрос_2_3 : запрос на выборку

Студенты

- * КодСтудента
- Фамилия
- Имя
- Отчество
- КодГорода

Платежи

- * КодСтудента
- ДатаОплаты
- СуммаОплаты
- ЦельОплаты

Поле: КодСтудента

Поле:	КодСтудента	Фамилия	Имя	Отчество	ДатаОплаты	СуммаОплаты	ЦельОплаты
Имя таблицы:	Студенты	Студенты	Студенты	Студенты	Платежи	Платежи	Платежи
Сортировка:							
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:							
или:							

Ввод платежей : запрос на выборку

```
SELECT Студенты.КодСтудента, [Фамилия] & " " & [Имя] & " " & [Отчество] AS Выражение1,  
Студенты.Факультет, Студенты.Группа, Платежи.ДатаОплаты, Платежи.СуммаОплаты,  
Платежи.ЦельОплаты  
FROM Студенты INNER JOIN Платежи ON Студенты.КодСтудента=Платежи.КодСтудента;
```

Ввод платежей : запрос на выборку

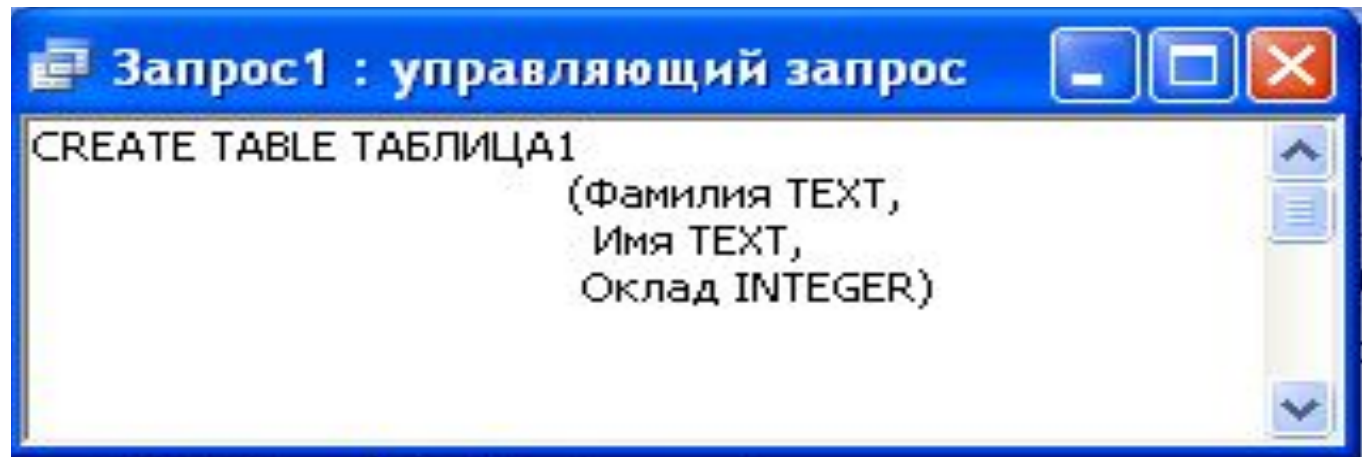
	КодСтудента	Выражение1	Факультет	Группа	ДатаОплаты	СуммаОплаты	ЦельОплаты
▶		1 Иванов Иван И	ГЭФ	МТ-11	05.01.2008	1 200 000р.	Обучение
	1	Иванов Иван И	ГЭФ	МТ-11	20.12.2007	10 000р.	Отработка
	1	Иванов Иван И	ГЭФ	МТ-11	12.04.2008	18 000р.	Пересдача
	2	Петров Петр П	ЭФ	Э-22	10.01.2008	1 200 000р.	Обучение
	2	Петров Петр П	ЭФ	Э-22	28.03.2008	35 000р.	Досдача
	2	Петров Петр П	ЭФ	Э-22	18.11.2028	10 000р.	Отработка

Создать таблицу. Инструкция **CREATE TABLE**

```
CREATE TABLE <таблица>  
    (<поле1><тип>[(<размер>)][<ограничения1>][<связь1>]  
    [, <поле2><тип>[(<размер>)][<ограничения2>][<связь2>]  
    [, ...]]  
    )
```

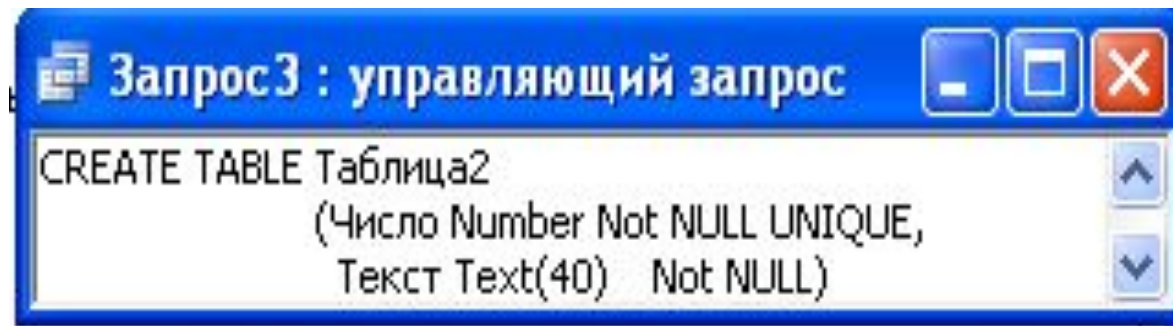
<i>Формат</i>	<i>Тип данных</i>	<i>Размер</i>
<i>Text</i>	Текстовый. Текст или числа, не требующие расчетов	Не превышает 255
<i>Memo</i>	Длинный текст	До 65535 символов
<i>Number</i>	Числовые данные, используемые для проведения расчетов	1, 2, 4 или 8 байт
<i>Date/Time</i>	Даты и время, относящиеся к годам с 100-го по 9999-й	8 байт
<i>Currency</i>	Денежные значения, используемые в расчетах	8 байт
<i>Counter</i>	Счетчик	4 байта
<i>Boolean</i>	Логический	1 бит

Создание новой таблицы с заданным именем и набором столбцов



Задание ограничений на данные создаваемой таблицы

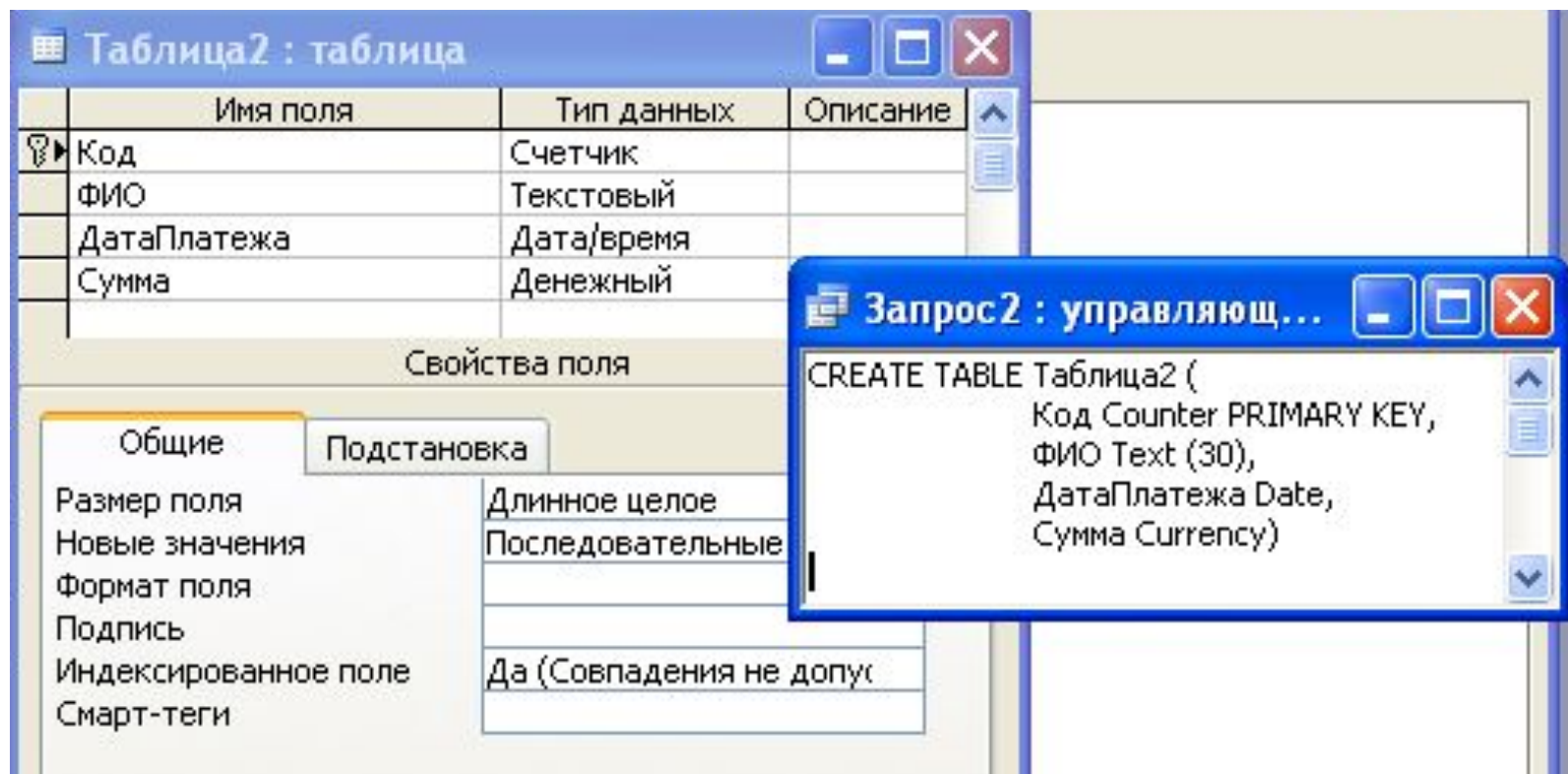
UNIQUE *NULL* или *NOT NULL*

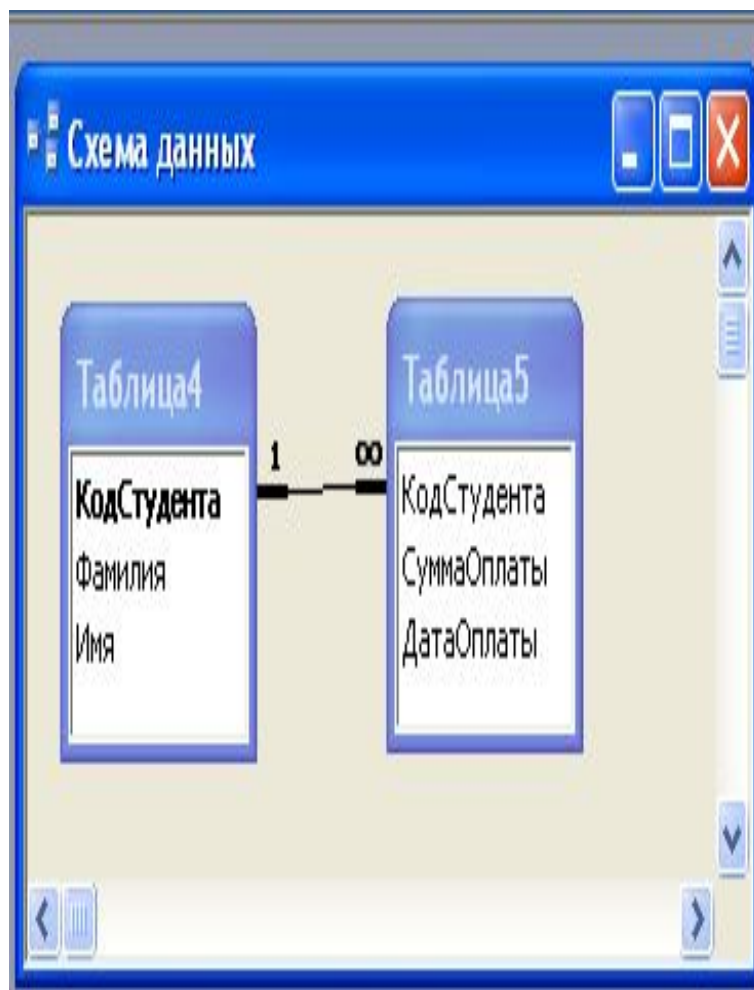


Создание связей между таблицами

PRIMARY KEY

REFERENCES <внеш_таблица> [(<внеш_поле1>[,<внеш_поле2>[, ...]])]
[ON UPDATE CASCADE | SET NULL]
[ON DELETE CASCADE | SET NULL] -





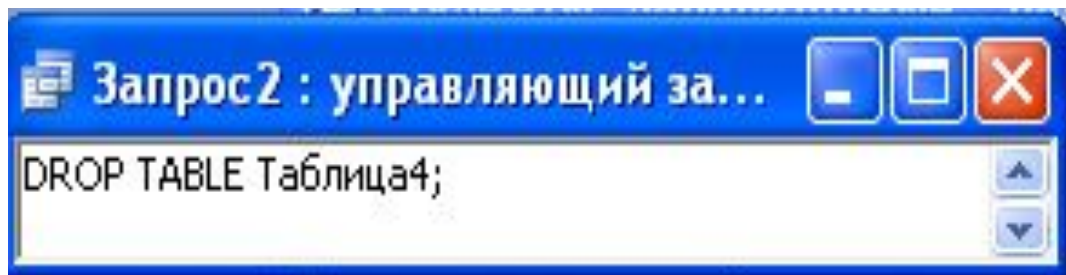
```
CREATE TABLE Таблица4  
(КодСтудента Counter PRIMARY KEY,  
Фамилия Text (30),  
Имя Text (15) );
```

Запрос_4_создание таблицы и установление связи по...

```
CREATE TABLE Таблица5  
(КодСтудента Long Not NULL REFERENCES Таблица4(КодСтудента),  
СуммаОплаты Currency,  
ДатаОплаты Date )
```


Удалить таблицу. Инструкция **DROP TABLE**

DROP TABLE <таблица>



Изменение структуры таблицы.

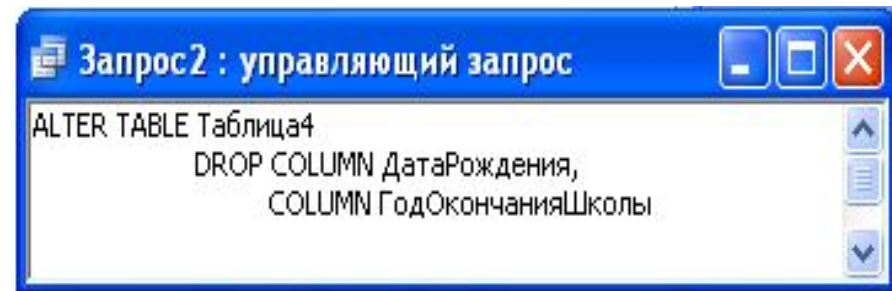
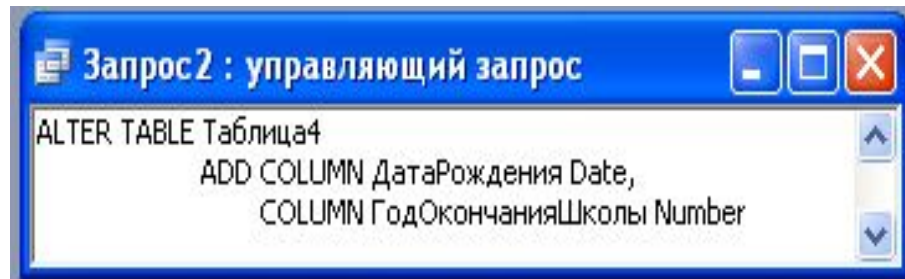
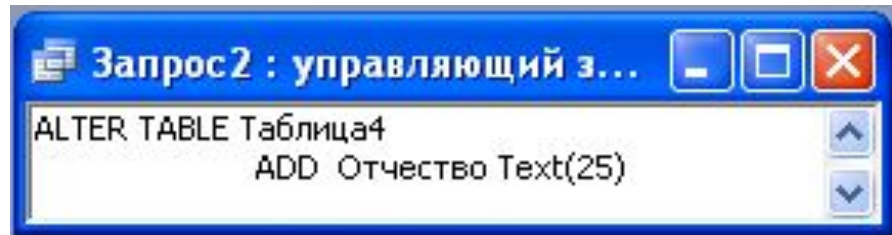
Инструкция **ALTER TABLE**

ALTER TABLE <таблица>
{ [COLUMN]<поле> <тип> [(<размер>)]
[, COLUMN ...]
DROP [COLUMN] <поле>[,COLUMN ...]}

ADD

DROP

COLUMN



Выборка записей. Инструкция *SELECT*

SELECT [<предикат>] <список
данных>

FROM <список
таблиц>

[WHERE <условие
выборки>]

[GROUP BY
<столбцы>]

[HAVING <условие
поиска>]

[ORDER BY
<спецификация>]
DISTINCT

ALL

TOP

Выбор источников данных. Предложение **FROM**

FROM <таблица1>[, <таблица2>[, ...]][**IN** <внешняя база>]

SELECT *
FROM Студенты

SELECT *
FROM Студенты IN Оплата

SELECT TOP 25 *
FROM Платежи IN "Y:\СУБД\Студенты.mdb"

```
SELECT *  
      FROM [Order]
```

```
SELECT d.Фамилия, d.Имя  
      FROM Студенты  
AS d;
```

Фильтрация данных. Предложение **WHERE**

WHERE <условие выборки>

<i>Операция</i>	<i>Оператор</i>
<i>Меньше</i>	<
<i>Больше</i>	>
<i>Равно</i>	=
<i>Меньше или равно</i>	<=
<i>Больше или равно</i>	>=
<i>Не равно</i>	<>

SELECT *

FROM Студенты

WHERE ГодПоступления

>2006;

Специальные функции *BETWEEN, LIKE, IN, NULL*

```
SELECT *  
    FROM Студенты  
    WHERE ГодПоступления BETWEEN 2000 and  
2008;
```

```
SELECT *  
    FROM Студенты  
    WHERE  
Фамилия Like "*а*";
```

```
SELECT *  
    FROM Студенты  
    WHERE ГодПоступления IN  
(2007, 2008);
```

```
SELECT *  
    FROM Студенты  
    WHERE Факультет IN  
("ГЭФ", "ЭФ");
```



```
SELECT *  
      FROM Студенты  
      WHERE Факультет NOT IN  
("ГЭФ", "ЭФ");
```

```
SELECT Фамилия  
      FROM  
Студенты  
      WHERE  
Телефон Is NULL;
```

```
SELECT Фамилия  
      FROM Студенты  
      WHERE Телефон Is  
NOT NULL,
```

Логические функции :

```
SELECT *  
    FROM Студенты  
    WHERE (Телефон Is Not NULL  
           AND  
           ГодПоступления > 2005 Or ГодПоступления  
< 2000)  
           OR  
           Факультет IN ("ГЭФ","ЭФ")
```

```
SELECT *  
    FROM Студенты  
    WHERE Телефон Is Not NULL  
           AND  
           ГодПоступления Not BETWEEN 2005 And 2008  
Or  
           ГодПоступления < 2000  
           OR  
           Факультет Not IN ("ГЭФ","ЭФ")
```

Выделение групп. Предложение **GROUP BY**

Агрегирующие функции:

AVG (среднее значение в группе);

MAX и **MIN** (соответственно максим. и минимальное значение в группе);

SUM (сумма значений в группе);

COUNT (число значений в группе).

GROUP BY <имя столбца>[, <имя столбца>[, ...]]

```
SELECT Count(Фамилия)
FROM Студенты
GROUP BY
```

ГодПоступления

Условие поиска для групп. Предложение **HAVING**

HAVING <условие поиска>

SELECT Count(Фамилия)
FROM Студенты
GROUP BY

ГодПоступления

HAVING

Count(Фамилия)>30

Упорядочивание данных. Предложение **ORDER BY**

ORDER BY <спецификация>[, <спецификация>[, ...]]

ASC

DESC

SELECT *

FROM Студенты

ORDER BY Факультет,

Группа, Фамилия

SELECT *

FROM

Студенты

ГодПоступления DESC