

# РАЗРАБОТКА БАЗ ДААННЫХ

ФИО преподавателя: Богомольная Г.В.

e-mail: bogomolnaya@mirea.ru

[Online-edu.mirea.ru](http://Online-edu.mirea.ru)

[online.mirea.ru](http://online.mirea.ru)

# ТЕМА СТРУКТУРА SQL

# План лекции

- Основные объекты структуры базы данных SQL-сервера.
- Синтаксис оператора создания таблиц.
- Синтаксис операторов обновления и удаления таблиц.
- Создание индекса.

# Структура SQL

## Основные объекты структуры базы данных SQL-сервера

Объекты	Смысл
Tables	Таблицы базы данных, в которых хранятся собственно данные
Views	Представления (виртуальные таблицы) для отображения данных из таблиц
Stored Procedures	Хранимые процедуры
Triggers	Триггеры – специальные хранимые процедуры, вызываемые при изменении данных в таблице
User Defined function	Создаваемые пользователем функции
Indexes	Индексы – дополнительные структуры, призванные повысить производительность работы с данными
User Defined Data Types	Определяемые пользователем типы данных
Keys	Ключи – один из видов ограничений целостности данных
Constraints	Ограничение целостности – объекты для обеспечения логической целостности данных
Users	Пользователи, обладающие доступом к базе данных
Roles	Роли, позволяющие объединять пользователей в группы
Rules	Правила базы данных, позволяющие контролировать логическую целостность данных
Defaults	Умолчания или стандартные установки базы данных

# Структура SQL

## *Операторы определения данных DDL (Data Definition Language)*

Оператор	Смысл	Действие
CREATE TABLE	Создать таблицу	Создает новую таблицу в БД
DROP TABLE	Удалить таблицу	Удаляет таблицу из БД
ALTER TABLE	Изменить таблицу	Изменяет структуру существующей таблицы или ограничения целостности, задаваемые для данной таблицы
CREATE VIEW	Создать представление	Создает виртуальную таблицу, соответствующую некоторому SQL-запросу
ALTER VIEW	Изменить представление	Изменяет ранее созданное представление
DROP VIEW	Удалить представление	Удаляет ранее созданное представление
CREATE INDEX	Создать индекс	Создает индекс для таблицы для обеспечения быстрого доступа по атрибутам, входящим в индекс
DROP INDEX	Удалить индекс	Удаляет ранее созданный индекс

# Операторы определения данных

## *Создание базы данных в среде MS SQL Server*

```
<определение_базы_данных> ::=  
CREATE DATABASE имя_базы_данных  
[ON [PRIMARY]  
[ <определение_файла> [...n] ]  
[,<определение_группы> [...n] ] ]  
[ LOG ON {<определение_файла>[,...n] } ]  
[ FOR LOAD | FOR ATTACH ]
```

# Операторы определения данных

## ***Создание таблиц***

*Базовый упрощенный синтаксис оператора создания  
таблицы **CREATE TABLE***

```
<определение_таблицы> ::=  
CREATE TABLE имя_таблицы  
(имя_столбца тип_данных  
[NULL | NOT NULL ] [,...n])
```

*Пример оператора создания таблицы:*

```
CREATE TABLE s1  
(ФИО VARCHAR (20) NOT NULL,  
Дисциплина VARCHAR (20) NOT NULL,  
Оценка SMALLINT NOT NULL);
```

# Операторы определения данных

## **Создание таблиц**

*Базовое полное определение оператора **CREATE***

### **TABLE**

```
CREATE TABLE имя_таблицы  
( { имя_столбца тип_данных [NOT NULL] [UNIQUE]  
[DEFAULT значение по умолчанию]  
[CHECK (условие проверки на допустимость) [,...]]  
[PRIMARY KEY (список столбцов),]  
{[UNIQUE (список столбцов),] [,...]}  
{[FOREIGN KEY {список столбцов внешних ключей)  
REFERENCES имя родительской таблицы [(список столбцов ключей-кандидатов)]}  
[ON UPDATE правило ссылочной целостности]  
[ON DELETE правило ссылочной целостности]] [,...]}  
{[CHECK (условие проверки на допустимость)] [,...]} )
```

*Пример оператора создания таблицы:*

```
CREATE TABLE s1 (ФИО VARCHAR (20) NOT NULL, Дисциплина VARCHAR (20) NOT NULL, Оценка  
SMALLINT NOT NULL);  
PRIMARY KEY (ФИО, Дисциплина),  
FOREIGN KEY ФИО REFERENCES S2  
ON UPDATE CASCADE  
ON DELETE CASCADE);
```



# Правила ссылочной целостности

*Правило целостности внешних ключей:*

- для каждого значения внешнего ключа должно существовать соответствующее значение первичного ключа в родительском отношении.

*Ссылочная целостность может быть нарушена при выполнении операций:*

- 1) обновление кортежа в родительском отношении;
- 2) удаление кортежа в родительском отношении;
- 3) вставка кортежа в дочернее отношение;
- 4) обновление кортежа в дочернем отношении.

# Стратегии поддержания ссылочной целостности

## Основные стратегии поддержания ссылочной целостности:

1. RESTRICT – не разрешать выполнение операции, приводящей к нарушению ссылочной целостности.
2. CASCADE – разрешить выполнение требуемой операции, но внести при этом необходимые поправки в других кортежах отношений так, чтобы не допустить нарушения ссылочной целостности и сохранить все имеющиеся связи.

## Дополнительные стратегии поддержания ссылочной целостности:

1. NONE – никаких операций по поддержке ссылочной целостности не выполняется.
2. SET NULL – разрешить выполнение требуемой операции, но все возникающие некорректные значения внешних ключей заменять на неопределенные значения (null-значения).
3. SET DEFAULT – разрешить выполнение требуемой операции, но все возникающие некорректные значения внешних ключей изменять на некоторое значение, принятое по умолчанию.

# Операторы определения данных

## **Обновление таблиц**

*Обобщенный формат оператора ALTER*

### **TABLE**

ALTER TABLE имя\_таблицы

[ADD [COLUMN] имя столбца тип данных [NOT NULL] [UNIQUE]

[DEFAULT значение по умолчанию] [CHECK (условие проверки на допустимость)]]

[DROP [COLUMN] ] имя\_столбца [RESTRICT | CASCADE]]

[ADD [CONSTRAINT [(имя ограничения)] ограничение]

[DROP CONSTRAINT имя ограничения [RESTRICT | CASCADE]]

[ALTER [COLUMN] SET DEFAULT значение по умолчанию]

[ALTER (COLUMN) DROP DEFAULT]

*Пример оператора обновления таблицы:*

ALTER TABLE s1

ADD Группа varchar (7) NOT NULL;

# Операторы определения данных

## Обновление таблиц

*Модификация структуры таблицы*

```
ALTER TABLE имя_таблицы  
  {[ADD [COLUMN] имя_столбца тип_данных [NULL | NOT NULL]]  
  | [DROP [COLUMN] имя_столбца]}
```

*Модификация таблицы*

```
ALTER TABLE имя_таблицы  
  {[ALTER COLUMN имя_столбца  
  {новый_тип_данных [(точность[,масштаб))] [ NULL | NOT NULL ]}  
  | ADD { [имя_столбца тип_данных]  
  | имя_столбца AS выражение } [...n]  
  | DROP {COLUMN имя_столбца} [...n]  
  }  
}
```

# Операторы определения данных

## Удаление

**таблицы** `DROP TABLE имя_таблицы [RESTRICT | CASCADE]`

*Пример оператора удаления таблицы:*

```
DROP TABLE s1;
```

## Операторы создания и удаления

**индексов** `CREATE INDEX имя_индекса`

```
CREATE [UNIQUE] INDEX имя_индекса  
ON имя_таблицы (столбец [ASC | DESC] [,_.])
```

*Удалить индекс:*

```
DROP INDEX имя_индекса
```

### *Способы определения индекса*

- автоматическое создание индекса при создании первичного ключа;
- автоматическое создание индекса при определении ограничения целостности UNIQUE ;
- создание индекса с помощью команды CREATE INDEX.

# Индексы и методы доступа

**Индексы** – это механизмы быстрого доступа к данным в таблицах БД.

## Физическая структура

**таблицы**

Порядковый № записи	Дата прихода товара	Наименование товара	Количество
1	10.01.2020	Сахар	10
2	12.01.2020	Картофель	50
3	12.01.2020	Свекла	20
4	14.01.2020	Сахар	50
5	14.01.2020	Свекла	10
6	16.01.2020	Сливы	4

## Логическая структура

**индексов**

По дате прихода товара		По наименованию товара		По количеству	
Дата прихода	№ записи	Товар	№ записи	Количество	№ записи
10.01.2020	1	Картофель	2	4	6
12.01.2020	2	Сахар	1	10	1
12.01.2020	4	Сахар	4	10	5
14.01.2020	3	Свекла	3	20	3
14.01.2020	5	Свекла	5	50	2
16.01.2020	6	Сливы	6	50	4

# Индексы и методы доступа

## ***Последовательный метод доступа к данным в таблицах БД:***

- просматриваются все записи таблицы, от первой к последней.

## ***Индексно-последовательный метод*** доступа к данным в таблицах БД:

- поиск ведется по индексу, а не по самой таблице;
- поиск в индексе начинается только с первой строки, удовлетворяющей, условию запроса или его части («прямой доступ»);
- строки в индексе, начиная с такой записи, просматриваются последовательно.

**Спасибо за  
внимание!**