

**SQL оператори за
създаване на бази от
данни и дефиниране на
схеми и таблици**

Основни теми

- Създаване на базата от данни
- Дефиниране на таблици
- Дефиниране интегритета на данните
- Модифициране структурата и свойствата на таблици
- Дефиниране на схеми

Създаване на база от данни

- `CREATE DATABASE db_name [.....]`
- При създаването на БД се дефинира:
 - Името на базата от данни.
 - Файловете, в които ще се съхранява базата от данни – логическо име; физическо име; разположение, размери и други характеристики.
 - Файловете, които ще се използват за транзакционен лог - логическо име; физическо име; разположение, размери и други характеристики.

Частичен синтаксис на SQL оператора за създаване на база от данни

- CREATE DATABASE *database_name*
[ON [PRIMARY] [<filespec>
[, [LOG ON <filespec>
- <filespec> ::= (NAME = *logical_file_name*
, FILENAME = '*os_file_name*'
[, SIZE = *size* [KB | MB | GB | TB]]
[, MAXSIZE = { *max_size* [KB | MB | GB |
TB] | UNLIMITED }]
[, FILEGROWTH = *growth_increment* [KB
| MB | GB | TB | %]])

Пример:

USE Master

GO

CREATE DATABASE Personnel

ON PRIMARY

(NAME = N'PersData'

, FILENAME = N'C:\Demoproject\Pers_Data.mdf'

, SIZE = 5MB, MAXSIZE = 10 MB, FILEGROWTH=1)

LOG ON

(NAME = N'PersLog'

, FILENAME = 'C:\Demoproject\Pers_Log.ldf'

, SIZE = 25MB, MAXSIZE = 50 MB, FILEGROWTH=10%)

Създаване на база от данни с характеристики по подразбиране

- SQL:
 - CREATE DATABASE <db_name>
- В Object Explorer на SQL Server Management Studio
 - От контекстното меню на възела **Databases** опция **New Database**
- Създаваната нова база от данни по подразбиране приема характеристиките на базата от данни model, ако не се зададени други.

Create database template

```
USE master
```

```
GO
```

```
-- Drop the database if it already exists
```

```
IF EXISTS ( SELECT name
```

```
FROM sys.databases
```

```
WHERE name = N'<Database_Name>'
```

```
)
```

```
DROP DATABASE <Database_Name>
```

```
GO
```

```
CREATE DATABASE <Database_Name>
```

```
GO
```

Дефиниране на таблици в базата от данни

- CREATE TABLE
[[*име на БД.*][*схема*].] *име на таблица*
.....
- Създава се описание на таблицата в специфицираната база от данни и в специфицираната схема
- Ако таблица със същото име в същата схема на БД съществува, то операторът не се изпълнява и се извежда съобщение

Дефиниране на таблици в базата от данни

- Частичен синтаксис на оператора:

CREATE TABLE

[[*име на БД.*][*схема*].] *име на таблица*

({<*дефиниция на колона*>

| *име на колона AS израз* [, *n*] })

Дефиниране на таблици в базата от данни

< дефиниция на колона > ::=
{име_на_колона тип_на_данните

[PRIMARY KEY | UNIQUE]
[DEFAULT израз_константа]
[IDENTITY [(n,k)] | [ROWGUIDCOL]]
[NOT NULL] ... }

Някои често използвани типове данни

- char(n)
- varchar(n)
- nchar(n)
- nvarchar(n)
- smallmoney
- money
- Decimal
[(p[,s])] – $p \leq 18$
- numeric[(p[,s])]
- float
- real
- int ($-2^{31} - 2^{31}-1$)
- smallint ($-2^{15} - 2^{15}-1$)
- tinyint (0 – 255)
- bigint ($-2^{63} - 2^{63}-1$)
- date (yyyy-mm-dd)
- datetime
- smalldatetime
- time
- uniqueidentifier
- image
- text

**Max
4000**

Свойства на колоните в таблицата

- PRIMARY KEY – първичен ключ
- NOT NULL – не се допуска неопределена стойност,
т.е. изисква задължително задаване на стойност
- UNIQUE – колоната може да приема само уникални стойности
- DEFAULT *израз_константа* – задава стойност по подразбиране
- IDENTITY (*seed, increment*) | ROWGUIDCOL – задава уникален идентификатор

Пример за създаване на таблица в T-SQL

```
USE Northwind
```

```
GO
```

```
CREATE TABLE My_Contacts.Employees
```

```
(
```

```
Emp_ID char(4) PRIMARY KEY
```

```
,FirstName varchar(30)
```

```
,LastName varchar(30) NOT NULL
```

```
)
```

Дефиниране на стойности по подразбиране

```
USE Northwind
```

```
GO
```

```
CREATE TABLE My_Contacts.Jobs
```

```
(JobsId int PRIMARY KEY
```

```
, Jobs_Name varchar(30) NOT NULL
```

```
, Comment varchar(30)
```

```
DEFAULT 'missing ')
```

Създаване на идентификаторни колони

- `IDENTITY (seed ,increment)` - уникалност в рамките на таблицата
 - *seed* – начална стойност, която се присвоява на колоната при въвеждане на първия ред в таблицата;
 - *increment* – стъпка на нарастване.

!!! Само една колона може да има свойство `IDENTITY`.

Пример:

```
USE Northwind
```

```
CREATE TABLE Employees0
```

```
(
```

```
Emp_ID SMALLINT IDENTITY(101,1)
```

```
,EmpName NVARCHAR(50) NOT NULL
```

```
)
```

Глобален уникален идентификатор

- **ROWGUIDCOL** - уникалност в рамките на всеки компютър; генерират се случайни 16-ични числа.
- Задава се само за колони **uniqueidentifier**
- **Свойството ROWGUIDCOL** не води до генериране на стойности за колоната.
- За да се генерират стойности за колоната се използва функцията NEWID или NEWSEQUENTIALID като стойност по подразбиране.

Генериране стойности на глобалния уникален идентификатор при дефиниране на колоната

- За колоната се задава:
 - Тип на данните UNIQUEIDENTIFIER
 - Клауза DEFAULT с функция NEWID() или NEWSEQUENTIALID())
 - При добавяне на редове в таблицата за колоната автоматично се генерират 16-ични стойности, които са уникални за всеки компютър в мрежата.

Пример на глобален уникален идентификатор

```
USE Northwind
```

```
GO
```

```
CREATE TABLE Employees1
```

```
(
```

```
Emp_ID UNIQUEIDENTIFIER DEFAULT NEWID()
```

```
NOT NULL
```

```
,EmpName NVARCHAR(60) NOT NULL
```

```
)
```

```
/* ROWGUIDCOL може да се изпусне*/
```

Модифициране структурата и свойствата на таблица

Частичен синтаксис:

```
ALTER TABLE table_name
{ ALTER COLUMN column_name
  { new_data_type [ ( precision [ , scale ] ) ]
    [ NULL | NOT NULL ] }
  | ADD
    { [ < column_definition > ]
      | column_name AS
computed_column_expression } [ ,...n ]
  | DROP COLUMN column_name }
```

- **ALTER COLUMN** <име на колона>.... – задава промяна типа на данните и дали колоната може да приема неопределени стойности
- **ADD** <дефиниция на колона> – добавяне на нова колона
- **DROP COLUMN** <име на колона> – премахване на колона

Пример:

```
USE Northwind
```

```
GO
```

```
/* добавяне на нова колона */
```

```
ALTER TABLE Orders
```

```
    ADD Comments nvarchar(50)
```

```
/* промяна размера на колона */
```

```
ALTER TABLE Employees
```

```
ALTER COLUMN LastName nvarchar(25)
```

Дефиниране на изчисляеми колони

<computed_column_definition> ::=

column_name

AS computed_column_expression

[PERSISTED [NOT NULL]]

- Колоната се изчислява въз основа на израз, използващ други колони от същата таблица.

Дефиниране на виртуална изчисляема колона

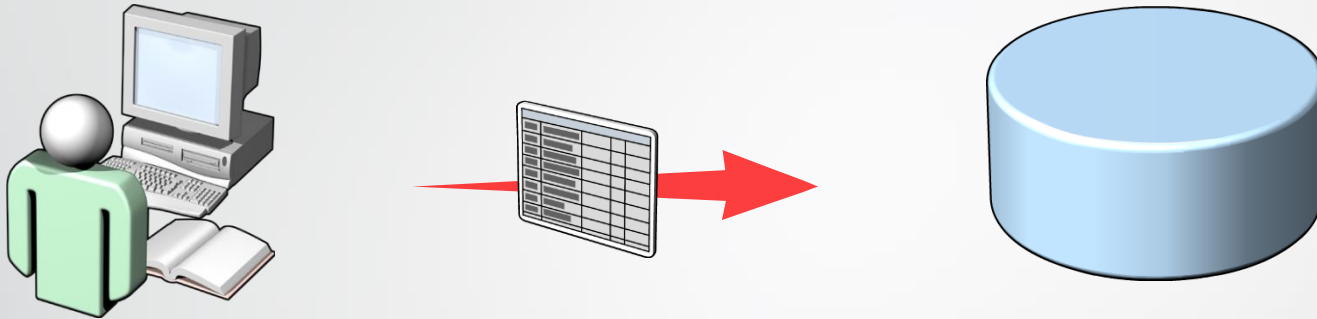
- `<computed_column_definition> ::= column_name AS computed_column_expression`
- Пример:
`ALTER TABLE [Order Details]`
`ADD Total AS Quantity*UnitPrice`
- ! Изчисляемата колона е виртуална (физически не се съхранява в таблицата). Генерира се при извличане на данни от таблицата.

Дефиниране на PERSISTED изчисляема колона

- `<computed_column_definition> ::= column_name
AS computed_column_expression PERSISTED
[NOT NULL]`
- Колоната физически се съхранява в таблицата и се обновява при промяна в някоя от колоните, участващи в израза, по който се изчислява.
- Пример:
`ALTER TABLE [Order Details]

ADD Total_euro AS Quantity*UnitPrice/1,96
PERSISTED`

Създаване на схеми



```
CREATE SCHEMA  
schema_name | AUTHORIZATION  
owner_name  
[ table_definition  
| view_definition ]
```

Създаване на схеми

- *schema_name* – име на схемата в базата от данни.
- `AUTHORIZATION owner_name` - задава се името на собственика на схемата; по подразбиране собственик е потребителят, който създава схемата.
- *table_definition* – дефиниция на таблица в създаваната схема.
- *view_definition* – дефиниция на изглед в създаваната схема.

Пример за създаване на схема в базата от данни Northwind и дефиниране на таблица в нея

```
USE Northwind
```

```
GO
```

```
CREATE SCHEMA My_Contacts
```

```
CREATE TABLE Contacts
```

```
/* създава се в схемата My_Contacts */
```

```
(FirstName Nchar(15), LastName Nchar(30)
```

```
, Home_number Nchar(20)
```

```
, Business_number Nchar(20)
```

```
, Mobile Nchar(10), Birthday datetime)
```

Преместване на обект от една схема в друга

- `ALTER SCHEMA schema_name TRANSFER object_name`
- *schema_name* – име на схема от текущата база от данни, в която ще се премести обекта.
- *object_name* – име на обекта, който ще се премести (**име_на схема.име_на_обект**)
- Пример:

USE Northwind

GO

```
ALTER SCHEMA My_Contacts TRANSFER  
dbo.new_table
```