

Врсте база података

База података је скуп података о неком систему, организован на основу захтева и потреба корисника.

Базе података су основни објекти о којима се брине релациони систем за управљање базама података.

У оквиру базе података, кориснички и системски подаци се чувају у табелама.

У SQL серверу постоје две основне врсте база података: системске и корисничке базе.

Системске базе се креирају током инсталације SQL сервера и могу бити:

- Мастер – главна база података са табелама преко којих SQL сервер прати рад целог система; у њој су подаци о свим базама у систему, процедурама, корисницима
- MSDB – је модел на којем се заснивају све нове базе података на SQL серверу, које креирају корисници
- Tempdb – база која чува привремене податке
- Resource – није видљива и може само да се из ње читају подаци; чува дефиниције свих системских објеката

Табеле

Табеле се састоје од колекције редова које имају једну или више колона.

Ред је најмања јединица података која се може унети или обрисати из табеле.

Свака табела може да садржи нула, један или више редова.

Табеле чувају податке о објектима (студенти, запослени, књиге...).

Сваки ред табеле су детаљи једног објекта, тачно специфицираног студента, запосленог...

Колоне одређују које се информације чувају о сваком објекту и дефинишу се типом података (нпр за студента колоне број индекса, име, презиме, датум рођења...).

Постоје две врсте табела: системске (креира SQL сервер и не могу се мењати) и корисничке.

Креирање табела

Наредба за креирање табела је CREATE TABLE а затим следи назив табеле са заградама.

Назив табеле мора бити јединствен и у складу са именовањем идентификатора.

У загради се наводи листа колона које су дефинисане именом и типом података.

Опционо, колоне могу бити ограничене провером вредности, јединственим кључевима...

Најбитнији део дефиниције колона јесте да ли се дозвољава NULL вредност.

Дефиниција табеле:

```
CREATE TABLE naziv_tabele
{
    naziv_kolone_1 tip_kolone [duzina] NULL|NOT NULL,
    ...
    naziv_kolone_n tip_kolone [duzina] NULL|NOT NULL
};
```

У имену табеле треба навести и назив шеме, којој ће табела припадати.

Шеме се користе као контејнери за објекте и осим табела могу да имају функције, процедуре, погледе...

Табела се модификује преко наредбе ALTER TABLE, за додавање или уклањање колона из табеле.

Табеле се бришу преко DROP TABLE (DROP TABLE IF EXISTS sema.tabela); ако су табеле повезане преко кључева, ниједна од њих се не може обрисати.

Пример: креирати табелу Ученик у шеми dbo па проверити да ли је табела креирана

```
CREATE TABLE [dbo].[Ucenik]
(
    [BrojUcenika] NVARCHAR(10) NOT NULL,
    [Ime] NCHAR(20) NOT NULL,
    [Prezime] NCHAR(20) NOT NULL,
    [ImeOcaIliMajke] NCHAR(20) NOT NULL,
    [JMBG] CHAR(13) NOT NULL,
    [Telefon] VARCHAR(15) NULL,
    [Adresa] NVARCHAR(40) NULL,
    [Email] NVARCHAR(256) NULL,
    [Lozinka] NVARCHAR(15) NULL
);
```

Направити нови упит над табелом Ученик: SELECT * FROM sys.tables;

Овиме се добија садржај свих системских табела у којима су сви подаци и о управо креираној корисничкој табели.

Други начин за проверу да ли је корисничка табела креирана је преглед хијерархијске структуре у SQL Sqever Object Explorer панелу.

Пример: Изменити садржај табеле Ученик додавањем нове колоне чији се садржај добија израчунавањем вредности из других колона у табели.

Десни клик на табелу Ученик унутар панела Server Explorer и изабрати опцију Show Table Data – отвара се нова картица са приказом табеле, па попунити један ред са подацима.

Унутар T-SQL карице додати нов ред и унети нову колону:

```
[GodinaRodjenja] AS SUBSTRING([JMBG], 5, 3, NULL)
```

После освежавања садржаја табеле, конекција са сервером и садржаја табеле на гриду, појављује се нова колона:

dbo.Ucenik [Data] dbo.Ucenik [Design]* Form1.cs [Design]										
	BrojUcenika	Ime	Prezime	ImeOcaIliMajke	JMBG	Telefon	Adresa	Email	Lozinka	GodinaRodjenja
	10	Miki	Mikic	Ana	1210007111111	NULL	NULL	NULL	NULL	007
**	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Колона ГодинаРођења је израчуната колона пошто њене вредности се добијају командама из других колона.

У овом случају то је команда SUBSTRING(kolona_sa_stringom, od_pozicije, broj_znakova) помоћу које се добија string са три знака који представљају три последње цифре године рођења из ЈМБГ.

Идентитет и секвенце

Често је потребно да се једна колона попуни низом целих бројева.

IDENTITY својство у колони табеле указује да вредност за ту колону аутоматски обезбеђује SQL сервер.

Приликом спецификације IDENTITY својства, потребно је да се одреди почетна вредност и вредност за повећање.

Seed представља почетну вредност а Increment је вредност која указује за колико ће се тренутна вредност за IDENTITY својство увећавати.

Подразумевана почетна вредност и вредност за повећање су 1, уколико није другачије специфицирано.

Помоћу секвенци се обезбеђују кључне вредности за више табела у оквиру исте шеме.

Секвенце су кориснички дефинисани објекти у бази и приликом њиховог креирања, потребно је да се наведе почетна вредност и вредност за увећање.

Секвенце су цикличне, тј када достигне своју максималну вредност, почиње поново од почетне вредности.

Пример: Дефинисати за колону PredmetID идентитет својство

```
CREATE TABLE [dbo].[Predmeti]
(
    [PredmetID] INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
    [NazivPredmeta] NVARCHAR(20) NOT NULL
);
```

Сада ће се, приликом додавања сваког новог реда у табелу Предмети, генерисати вредност за колону PredmetID.

Пошто је почетна вредност за идентитет својство 1 а увећање 1, колона у првом реду ће имати вредност 1, у другом 2...

Обично се у исту колону поставља јединствено идентификовање једног реда у односу на друге редове, па се додаје PRIMARY KEY.

Пример: унети секвенцу која се назива ID у шеми dbo.; секвенца је целобројног типа и дефинисана је за генерисање целобројних вредности; вредност секвенце почиње са 1 и повећавање се за 1.

```
CREATE SEQUENCE [dbo].[ID] AS int
START WITH 10
INCREMENT BY 1;
GO --razdvaja dve razlicite naredbe u jednom fajlu
CREATE TABLE [dbo].[Zaposleni]
(
    [ZaposleniId] INT NOT NULL DEFAULT (NEXT VALUE FOR dbo.ID) PRIMARY
    KEY,
    [Ime] NVARCHAR(20) NULL,
    [Prezime] NVARCHAR(20) NULL,
    [DatumRodjenja] DATE NULL
);
```

Задаци за самосталан рад

1. У оквиру базе података о ученицима, креирати табелу Одељење у оквиру шеме dbo. Табела треба да садржи следеће колоне: OdeljenjeID (int тип, вредност аутоматски обезбеђује SQL сервер преко идентитета својства, примарни кључ), Naziv (мора да садржи 20 знакова), Lokacija (може да садржи 30 знакова)
2. У бази података, у оквиру шеме dbo, креирати табелу PlatniRazredi: PlatniRazredID (int тип, обавезна колона), Min_zarada (int тип, обавезна колона), Max_zarada (int тип, обавезна колона)
3. У оквиру шеме dbo, креирати табелу Zaposleni: ZaposleniID (int тип, обавезна колона), Ime (мора да садржи до 20 знакова), Prezime (мора да садржи до 20 знакова), PunIme (израчуната колона, састоји се од имена и презимена са једним размаком између), DatumRodjenja (може да садржи једино исправне датуме), DatumZaposlenja (мора да садржи исправан датум), Plata (мора да садржи износ зараде у децималном формату), NadredjeniID (може да садржи шифру надређеног шефа који је уједно и сам запослен и чији се подаци налазе у истој табели), Telefon (може да садржи до 15 карактера), Adresa (може да садржи до 40 знакова), Email (може да садржи до 256 знакова)