

Нормализација базе података – Прва нормална форма

Да би се подаци правилно и ефикасно сместили у табелу db користи се процес који се назива нормализација. База података се сматра нормализованом када испуњава сет правила познатих као Три правила нормализације. Ово значи да је у db не постоји дупликација података, db је ефикасна и да је db квалитетно структурирана. Нормализација се примењује на сваку табелу понаособ. За табелу се каже да је испунила прву нормалну форму (1НФ) када не садржи ниједну колону која се може користити за смештање поновљених група података. То значи да не сме постојати ниједна колона која чува више од једне вредности или вишеструке колоне које чувају сличне вредности.

Ако је табела Контакти попуњена као на слици:

	Kontakt_ID	Ime	Prezime	Datum_Rodjenja	Brojevi_Telefona	Dozvoljen_Kontakt_Telefonom	Adresa_Prva	Adresa_Druga	Uloga_ID	Uloga_Titula	Beleska_1	Beleska_2
1	Mica	Micko	1974-10-03	011456234, 064345...	True	Bulevar 24 Beog...	Drugi Sor bb Vinca 1	1	Korisnik db	Volj teske zadatke	Volj SQLS	
3	Pera	Petko	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

Види се да су у колони Brojevi_Telefona убачена два телефонска броја одвојена зарезом. Лоше стране ове табеле су баш ова колона у коју се смештају вишеструке вредности. Такође лоше страна су и колоне Beleska_1 и Beleska_2 које нас лимитирају на две белешке по контакту (можда желимо више информација о контактима од две белешке). Ова db је лоше дизајнирана и структурирана. Све што се користи за смештање више од једног податка је понављајућа група па је зато и колона Brojevi_Telefona понављајућа група. Зато прво треба променити име колоне у Broj_Telefona а затим и поделити колону у две колоне. Ова модификација се може извести коришћењем SSMS или system stored процедуре.

Пример14: Модификација постојеће табеле коришћењем SSMS.

Column Name	Data Type	Allow Nulls
Kontakt_ID	int	<input type="checkbox"/>
Ime	varchar(40)	<input checked="" type="checkbox"/>
Prezime	varchar(40)	<input checked="" type="checkbox"/>
Datum_Rodjenja	date	<input checked="" type="checkbox"/>
Brojevi_Telefona	varchar(200)	<input checked="" type="checkbox"/>
Dozvoljen_Kontakt_Telefonom	bit	<input checked="" type="checkbox"/>
Adresa_Prva	varchar(200)	<input checked="" type="checkbox"/>
Adresa_Druga	varchar(200)	<input checked="" type="checkbox"/>
Uloga_ID	int	<input checked="" type="checkbox"/>
Uloga_Titula	varchar(200)	<input checked="" type="checkbox"/>
Beleska_1	varchar(200)	<input checked="" type="checkbox"/>
Beleska_2	varchar(200)	<input checked="" type="checkbox"/>
Broj_Vozacke_Doizvole	varchar(40)	<input checked="" type="checkbox"/>
Broj_Pasosa	varchar(40)	<input checked="" type="checkbox"/>
Potvrđen_Kontakt	bit	<input checked="" type="checkbox"/>
Datum_Kreiranja	datetime	<input checked="" type="checkbox"/>

Да би се табела модификовала мора се прво затворити. Десни клик на назив табеле у ОЕ и избор опције Design омогућава едитовање било која колоне у табели. Клик на име колоне омогућава преименовање колоне а клик на DataType ћелију омогућава измену типа података у колони. Колона Allow Nulls одређује да ли се null вредности могу сместити у колону. Null вредност значи да никаква вредност није обавезна за ову колону и то је опционо. Детаљније измене су могуће десним кликом на име колоне, чиме се добија падајући мени са опцијама. Променити Brojevi_Telefona у Broj_Telefona. Затим сачувати модификовану табелу као File/Save Kontakti.

Пример15: Модификација постојеће табеле коришћењем system stored процедуре.

За потребе вежбе вратити име колоне у претходно. Администратори db не могу користити SSMS већ морају радити са скриптама. Пошто не постоји SQL наредба за измену имена колоне, користи се опција преименовања замотана као system stored процедура звана sp_rename. Ове процедуре су уграђене SQLS процедуре и обично почињу са sp_. Процедура sp_rename се користи за преименовање објеката, попут табела, других процедура и колоне. Скрипт за промену имена колоне:

```
USE Adresar;
EXEC sp_rename
@objname = 'dbo.Kontakti.Brojevi_Telefona',
@newname = 'Broj_Telefona',
@objtype = 'COLUMN';
GO
```

Стартовати скрипт.

После тога треба отворити 02 – Kreiraj Tabelu Kontakata.sql и извршити преименовање колоне Brojevi_Telefona у Broj_Telefona.

Овај корак је неопходан да би сви објекти базе података били update са променама у бази.

Промена имена колоне у бази података неће аутоматски променити садржај скрипте.

	Kontakt ID	Ime	Prezime	Datum Ro...	Broj Telefona	Dozvoljen ...	Adresa Prva	Adresa Dru...
▶	1	Mica	Micko	1974-10-03	011456234,...	True	Bulevar 24 B...	Drugi Sor bb...
	3	Pera	Petko	NULL	NULL	NULL	NULL	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Сада је поправљено име колоне али не и подаци у колони Broj_Telefona пошто су у њој и даље два телефонска броја. Проблем са колонама Beleska_1 и Beleska_2 су што ако је потребно више колона да презентује исти тип података, тртебало би да се ставе у сопствену табелу.

Тако ће бити могуће да сваки контакт има већи број белешки.

Ова табела ће бити дете табела табеле Контакти, и зато ће бити потребна Контакти_ID колона.

Она неће бити IDENTITY колона.

Такође ће бити потребна накаква Beleska_ID колона како би се јединствено дефинисале свака од бележака.

Пример16: Отворити нови упит и направити нову скрипту:

```
USE Adresar;
```

```
IF EXISTS
(SELECT 1 FROM sys.table_types WHERE [Name] = 'Kontakti_Beleske')
BEGIN
    DROP TABLE dbo.Kontakti_Beleske;
END;

CREATE TABLE dbo.Kontakti_Beleske
(
    Beleska_ID INT IDENTITY(1,1),
    Kontakt_ID INT,
    Beleske VARCHAR(200),
    CONSTRAINT PK_Kontakti_Beleske PRIMARY KEY CLUSTERED (Beleska_ID)
);
```

GO

Сачувати скрипту табеле: C:\...\Ime_Prezime\primeni\03 – Kreiraj Tabelu Kontakti_Beleske.sql.