

# KOME TREBA DEJTABEJZ DIZAJN, BRT?

*MILOŠ RADIVOJEVIĆ, MICROSOFT DATA PLATFORM MVP*





# Digital. Since Always.

November 25th - 26th 2020 Belgrade

Powered by

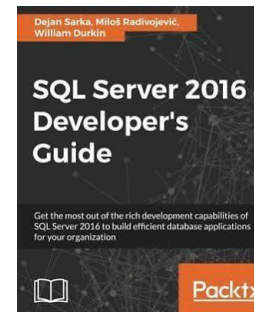
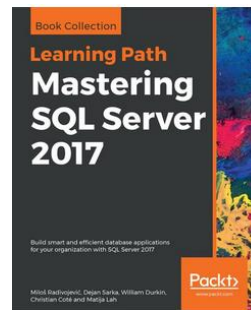
**Microsoft**

# MILOŠ RADIVOJEVIĆ

- Data Platform MVP
- Principal Database Consultant at bwin, Vienna, Austria
- Co-Founder: SQL Pass Austria
- Conference Speaker, Book Author
- Contact:
  - E: [milos.radivojevic@chello.at](mailto:milos.radivojevic@chello.at)
  - LinkedIn: [milos-radivojevic-2907bb1b](https://www.linkedin.com/in/milos-radivojevic-2907bb1b)



Microsoft®  
Most Valuable  
Professional



# OSVRT NA LEARNING TENDENCIJE



Nije bitno znanje, bitno je da si spreman na promene!

Ja nisam znao ništa o tome, ali sam se ipak upustio!

Taj video je duži od 10 minuta, ne mogu tako da učim, brt!

- Šta je to treća normalna forma (3NF)?
- Treća šta?

# ZAŠTO JE BITAN DOBAR DB DIZAJN?

- Koko god to izlizano zvučalo, db design je temelj projekta
- Baza podataka obično nadživi aplikacije i servise
- Promene su veoma skupe i u kasnim fazama projekta praktično nemoguće
- Dobar dizajn čuva pare, vreme i reputaciju
  - Ko ti je ovo dizajniro, brt?





# PROBLEMI SA DIZAJNOM BAZA PODATAKA

- Danas svi igraju fudbal <<<<  
danas svi rade u IT-u
- Nedovoljno prisutna tema
- Alat
- **Agile**





**DO NOT CREATE A TABLE IN TWO MINUTES!**

# DO NOT CREATE A TABLE IN TWO MINUTES!

- Kako programer obično pravi tabelu?
  - nađe CREATE TABLE neke tabele,
  - udari Copy
  - udari Paste
  - malo promeni imena kolona
  - izabere primarni ključ i toeto
- Do not create table with copy/paste!
- Nemojte ljudi!





# DO NOT CREATE A TABLE IN TWO MINUTES!



- Implementacija poslovne logike počinje sa CREATE TABLE
- Odvojite dovoljno vremena prilikom analize zahteva!
- Ozbiljne promene i redizajn su skupe, dugotrajne, neretko i mučne



# KONVENCIJA O IMENOVANJU OBJEKATA

# IMENOVANJE OBJEKATA

- Poštujte konvencije o imenovanju
- Kolege će lakše čitati vaš kod i smanjíte mogućnost različitih interpretacija
- Preimenovanje postojećih objekata veoma je skupo
- Ne pravite izuzetke – izuzeci ubrzo postaju pravila

# IMENOVANJE OBJEKATA — PRIMER DIJALOGA

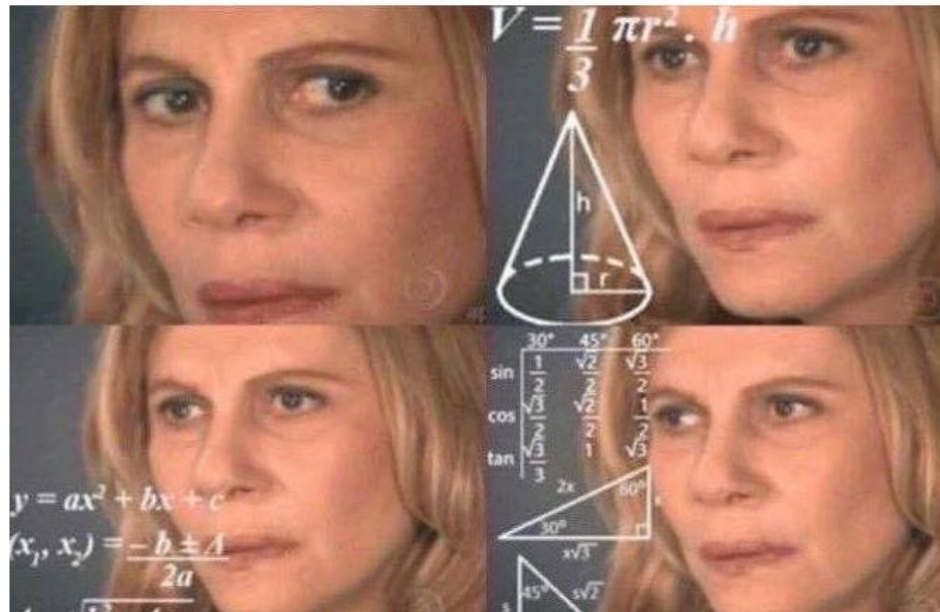
- Neko je imenovao kolonu **fDate**
  - - Koji je to datum: start, kat of, end ili transakcion tajm stemp?
  - - Svašta pitaš, brt, pa start dejt!
  - - A je l' UTC, GMT ili koji već format?
  - - Pa UTC, brt, u ovoj bazi ovde koristimo UTC, a u onoj tamo GMT
- - JTŽMB! A što onda kolonu nisi nazvao fStartDateUTC?



# IMENOVANJE OBJEKATA - IZBEGAVAJTE

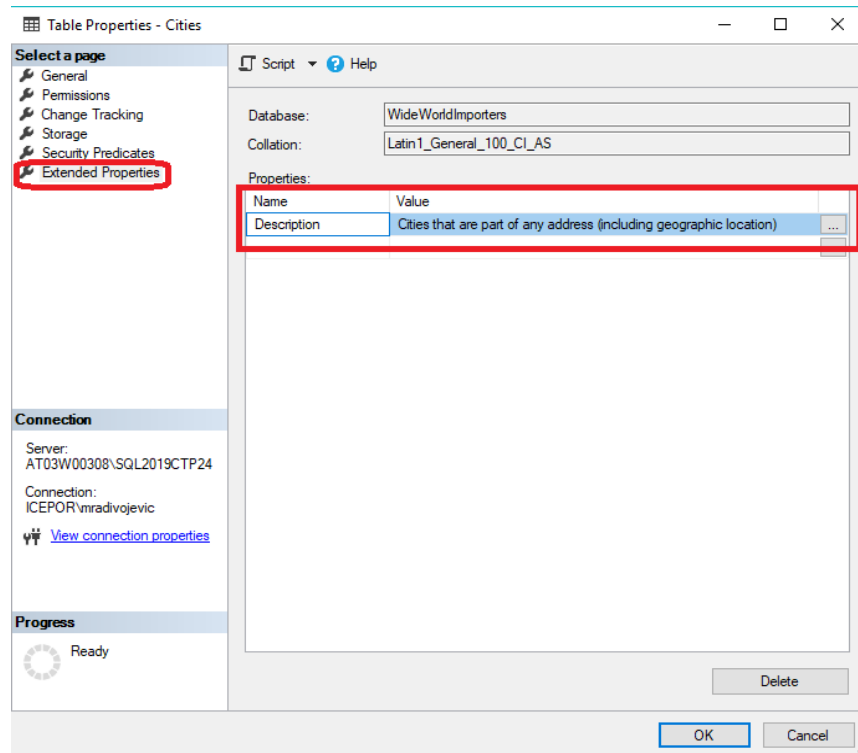
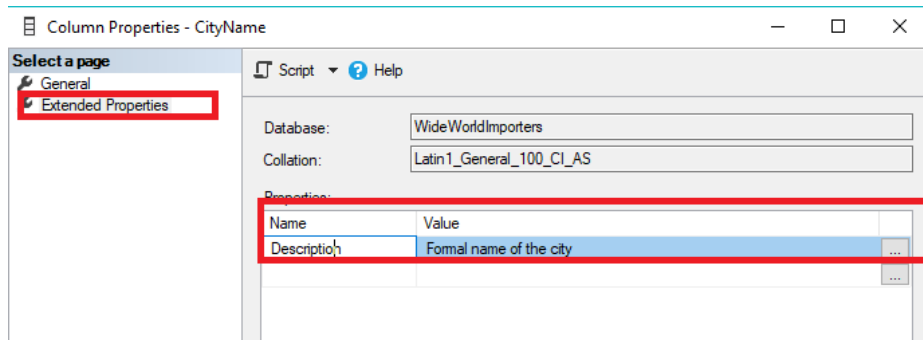
- Spejs karakter u imenu
- Ključne reči
- Skraćenice
- Negacije
  - isValid >>> isValid

```
WHERE isValid NOT IN ('isValid', 'expired')
```



# IMENOVANJE OBJEKATA

- Dokumentovati tabele i kolone:
  - ExtendedProperties
- Bolje ikakva konvencija nego nikakva







# ODABIR ODGOVARAJUĆEG TIPA PODATAKA

# ODABIR ODGOVARAJUĆEG TIPA PODATAKA



Koji tip podataka programeri najčešće koriste za polje Status?

INT

INT je veoma fleksibilan tip podatka, dozvoljava vam da imate mnogo različitih statusa: 4 294 967 295

$$-2^{31} \leq INT \leq 2^{31} - 1$$

$$-2,147,483,648 \leq INT \leq 2,147,483,647$$

- Za storage mu treba 4 bajta
- Nemojte INT za status, uzmite TINYINT



**Sinergija20**  
DIGITAL. SINCE ALWAYS.

# NIJE SVEJEDNO

```
CREATE TABLE dbo.StudentExams1(  
    exam_number int NOT NULL,  
    student_id int NOT NULL,  
    exam_id int NOT NULL,  
    exam_note int NULL,  
    exam_date datetime NULL,  
CONSTRAINT PK_StudentExams1 PRIMARY KEY CLUSTERED (exam_number ASC)  
)  
GO
```

```
CREATE TABLE dbo.StudentExams2(  
    exam_number int NOT NULL,  
    student_id int NOT NULL,  
    exam_id int NOT NULL,  
    exam_note tinyint NULL,  
    exam date date NULL,  
CONSTRAINT PK_StudentExams2 PRIMARY KEY CLUSTERED (exam_number ASC)  
)  
GO
```

# NIJE SVEJEDNO

```
DECLARE @date_from DATETIME = '20000101';
DECLARE @date_to DATETIME = '20171231';
DECLARE @number_of_rows INT = 10000000;
INSERT INTO dbo.StudentExams1
SELECT n AS exam_number,
       1 + ABS(CHECKSUM(NEWID())) % 50000 AS student_id,
       1 + ABS(CHECKSUM(NEWID())) % 40 AS exam_id,
       5 + ABS(CHECKSUM(NEWID())) % 6 AS exam_note,
       (SELECT(@date_from +(ABS(CAST(CAST( NewID() AS BINARY(8) )AS INT)))
% CAST((@date_to - @date_from)AS INT)))) exam_date
FROM dbo.GetNums(@number_of_rows)
GO
```

# NIJE SVEJEDNO

--Check table size

```
SELECT OBJECT_NAME(s.object_id) AS table_name, CAST((s.used_page_count/128.0) AS int) ta
FROM sys.dm_db_partition_stats AS s
INNER JOIN sys.indexes AS i
    ON s.[object_id] = i.[object_id] AND s.index_id = i.index_id
INNER JOIN sys.tables AS t
    ON s.[object_id] = t.[object_id]
WHERE s.object_id IN ( OBJECT_ID('dbo.StudentExams1'), OBJECT_ID('dbo.StudentExams2')) AN
```

/\*

table_name	table_size_MB
------------	---------------

StudentExams1	320
---------------	-----

StudentExams2	240
---------------	-----

\*/



**Sinergija**20

DIGITAL. SINCE ALWAYS.

# NIJE SVEJEDNO

```
--Check the data page number
SELECT OBJECT_NAME(p.object_id) AS table_name, data_pages
FROM sys.allocation_units AS a
INNER JOIN sys.partitions AS p
    ON a.container_id = p.partition_id
WHERE p.object_id IN ( OBJECT_ID('dbo.StudentExams1'), OBJECT_ID('dbo.StudentExams2'))
GO

/*
table_name                data_pages
-----
StudentExams1            40842
StudentExams2            30960
*/
```



# ODABIR ODGOVARAJUĆEG TIPA PODATAKA

- Izaberite najmanji tip podataka koji je dovoljan da pokrije potrebni domen
- Ako izaberete robusniji tip podataka
  - Potrebno je više prostora na disku na serveru i kopijama, bekap je sporiji...
  - SQL Server mora da učitati manje stranica za isti rezultat
- Izaberite DATE tip podatka, ako nije potrebna preciznost u satima ili minutima
- Promene tipa podataka su veoma skupe, teško se prodaju, a uglavnom niko nije zainteresovan da za njih plati

# ODABIR ODGOVARAJUĆEG TIPA PODATAKA

- Promene koje su neophodne u sistemu kada menjate tip podataka:

- Ažuriranje tipa podataka na nivou tabele

- ```
ALTER TABLE dbo.StudentExams ALTER COLUMN exam_note TINYINT;
```

lonu čiji se tip menja

- ```
IF OBJECT_ID('dbo.insertStudentExams','U') IS NULL  
    DROP PROCEDURE dbo.insertStudentExams
```

- ```
GO  
CREATE PROCEDURE dbo.insertStudentExams
```

- Efe 

```
@student_id int, @exam_id int, @exam_note tinyint  
AS //se.ExamNote = dr.GetInt32(2);  
kor INSE se.ExamNote = (int) dr.GetByte(2); id,@exam_note)
```

kses lejera

rajnjeg





# UPOTREBA OGRANIČENJA (CONSTRAINTS)

# UPOTREBA OGRANIČENJA (CONSTRAINTA)

- CHECK Constraints
  - Ograničite vrednosti koje su dozvoljene u okviru kolone
  - Svi korisnici će interpretirati vrednosti u koloni na isti način
- UNIQUE constraints
  - Ako je nešto jedinstveno, upotrebite UNIQUE konstrejnt i podelite informaciju o jedinstvenosti sa RDBMS sistemom
- To je dobro i za integritet i za performanse!

# UPOTREBA OGRANIČENJA (CONSTRAINTA)

- Imenovanje konstrejnta
- Uvek ih imenujte, u suprotnom nema promena na koloni dok se ne ukloni constraint, a imena mu ne znate

```
create table dbo.k (id int not null,  
  c1 date default getdate()  
)  
go  
alter table k alter column c1 datetime
```

100 %

Messages

Msg 5074, Level 16, State 1, Line 5  
The object 'DF\_k\_c1\_276EDEB3' is dependent on column 'c1'.  
Msg 4922, Level 16, State 9, Line 5  
ALTER TABLE ALTER COLUMN c1 failed because one or more objects access this column.



# REDUKOVANJE BROJA NULLABILNIH KOLONA



# PRIMER FLEKSIBILNOSTI

- Tabela Musterija

| \SQL201...9 - dbo.Musterija |               |              |                                     |
|-----------------------------|---------------|--------------|-------------------------------------|
|                             | Column Name   | Data Type    | Allow Nulls                         |
| ▶                           | Id            | int          | <input type="checkbox"/>            |
|                             | Titula        | nvarchar(8)  | <input checked="" type="checkbox"/> |
|                             | Ime           | nvarchar(50) | <input checked="" type="checkbox"/> |
|                             | Prezime       | nvarchar(50) | <input checked="" type="checkbox"/> |
|                             | DatumRodjenja | datetime     | <input type="checkbox"/>            |
|                             | Grad          | nvarchar(50) | <input type="checkbox"/>            |
|                             | Adresa        | nvarchar(50) | <input checked="" type="checkbox"/> |
|                             | Telefon       | nvarchar(50) | <input checked="" type="checkbox"/> |
|                             | Email         | nvarchar(50) | <input checked="" type="checkbox"/> |
|                             |               |              | <input type="checkbox"/>            |

# PRIMER FLEKSIBILNOSTI

- Scena 1: programer testira INSERT stejtment

```
INSERT INTO Musterija(Ime, Prezime) VALUES('Marinko', 'Rokvić');
```

Msg 515, Level 16, State 2, Line 20

Cannot insert the value NULL into column 'DatumRodjenja', table 'AdventureWorks2019.dbo.Musterija'; column does not allow nulls. INSERT fails.  
The statement has been terminated.



# PRIMER FLEKSIBILNOSTI

- Programer koriguje stejtment i daje dodatne info

```
INSERT INTO Musterija(Ime, Prezime, DatumRodjenja) VALUES  
( 'Marinko', 'Rokvić', '19540127' );
```

Msg 515, Level 16, State 2, Line 22

Cannot insert the value NULL into column 'Grad', table 'AdventureWorks2019.dbo.Musterija'; column does not allow nulls. INSERT fails.  
The statement has been terminated.



# PRIMER FLEKSIBILNOSTI

- Scena 2: programer omogućava INSERTU da radi
- `ALTER TABLE Musterija ALTER COLUMN Grad NVARCHAR(50) NULL;`
- `ALTER TABLE Musterija ALTER COLUMN DatumRodjenja DATETIME NULL;`
- Programer ponovo proba INSERT stejtment:
- `INSERT INTO Musterija(Ime, Prezime, DatumRodjenja)  
VALUES('Marinko', 'Rokvić', '19540127');`

`(1 row affected)`

# PRIMER FLEKSIBILNOSTI

- Ali:
- `INSERT INTO Musterija(Ime, Prezime) VALUES('asdf', 'dfdfdfdggd');`

`(1 row affected)`

# EPILOG

| Id | Titula | Ime     | Prezime    | DatumRodjenja           | Grad    | Adresa                         | Telefon   | Email                       |
|----|--------|---------|------------|-------------------------|---------|--------------------------------|-----------|-----------------------------|
| 1  | NULL   | Mitar   | Miric      | 1957-01-16 00:00:00.000 | Beograd | Ulica u kojoj živi Mitar Miric | 011123123 | info@nemozenamnikonista.com |
| 4  | NULL   | Marinko | Rokvic     | 1954-01-27 00:00:00.000 | NULL    | NULL                           | NULL      | NULL                        |
| 5  | NULL   | asdf    | dfdfdfdggd | NULL                    | NULL    | NULL                           | NULL      | NULL                        |

Sve radi, ALI

- Koko imaš mušterija, brt?
- Oćemo da brišemo ovog brta asdfa?
- Stani, nemoj da brišeš, aj stavi neki fleg

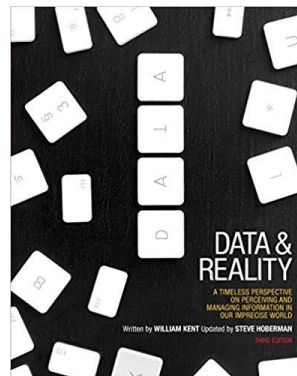
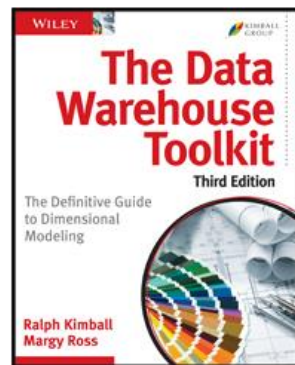
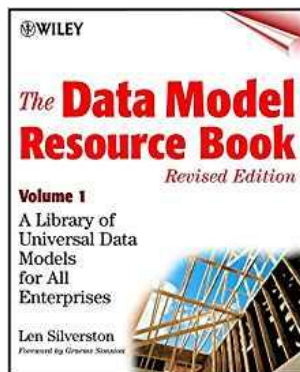
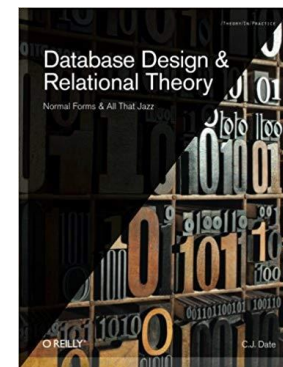
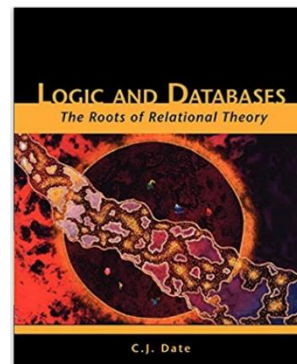
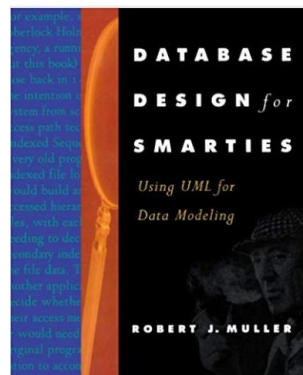
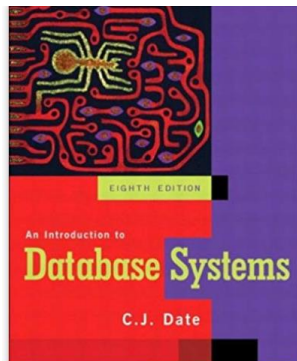




# REDUKOVANJE NULL-ABILNIH KOLONA

- Smanjiti broj kolona koje prihvataju NULL
  - Unosi overhead na nivou svakog reda
  - Aplikacioni kod mora da hendluje setove podataka koji sadrže NULL
- Različiti aktori obično imaju različite interpretacije redova u kojima kolona ima NULL
- Pokažite da vam je stalo do kvaliteta i konzistentnosti podataka
- Nekada je nemoguće izbeći NULL, ali budite sigurni da ste bar pokušali
- Ako ne umete da objasnite zašto kolona podržava NULL onda verovatno i ne mora

# PREPORUČENA LITERATURA



# ZAKLJUČAK

- Implementacija poslovne logike počinje od baze podataka
- „Poslovna logika pripada aplikaciji, baza je samo magacin sa podacima“ - jedan od opasnijih mitova u programiranju
- Baza je trajnija od aplikacije
- Obezbedite sebi dovoljno vremena za implementaciju

# ZAKLJUČAK

- Pažljivo i konzistentno imenujte tabele i druge objekte poštujući konvenciju o imenovanju
- Za svaku kolonu u tabeli:
  - odaberite najmanji tip podatka koji pokriva domen atributa
  - promene tipa podataka su skupe
  - ako kolona može da prihvata samo određene vrednosti, obavezno upotrebite CHECK constraint
  - redukujte broj kolona koje prihvataju NULL

# Thank you

Powered by

**Microsoft**