
DATABASE DESIGN

TIPIČNE GREŠKE



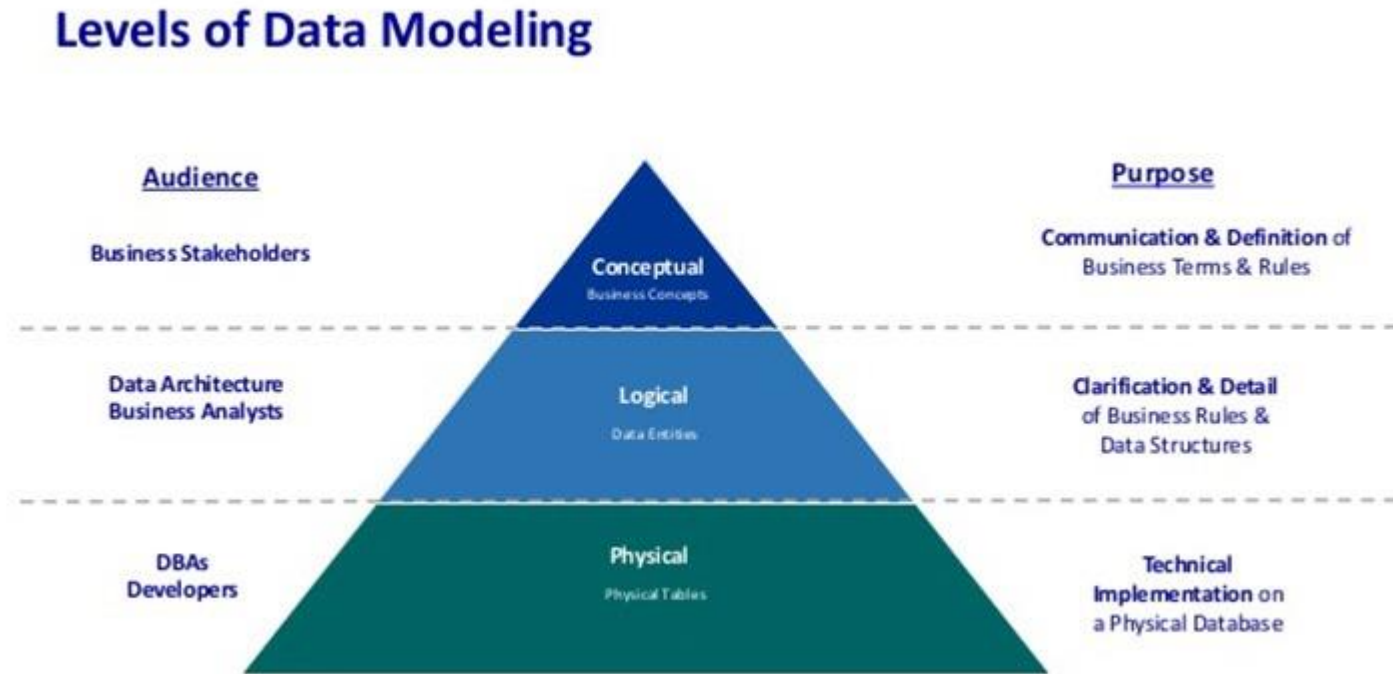
MILOŠ RADIVOJEVIĆ, MICROSOFT DATA PLATFORM MVP

Zašto je bitan dobar 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?

Problems with database design

- U svakom većem projektu trebalo bi da budu implementirana tri tipa db dizajna
 - konceptualni
 - logički
 - fizički



• ALI

Problemi sa dizajnom baza podataka

- Danas svi igraju fudbal <<< Danas svi rade u IT-u
- Gde se uči dejtabejz dizajn?

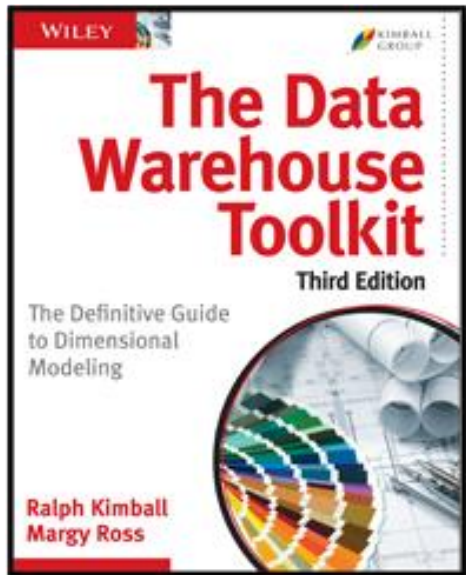


- Nedovoljno prisutna tema u moru dejtabejz tema
- SQLSaturday Conference:
 - 7390 naslova sesija
 - samo 16 naslova imaju u imenu sintagmu database design

0, 22%

Ko bi trebalo da dizajnira baze i tabele?

- Da nije možda...
... dejtabejz dizajner?
- Application developeri prave tabele!



Problemi sa dizajnom baza podataka

- RDBMS vendori gotovo da ne podržavaju alate za dizajn (SQL Server npr.)
- Agile
 - brt mora da završimo do srede, nemamo kad da razmatramo sva ta pitanja, posle ćemo
- Ljudi ne arhiviraju podatke, sve je u glavnoj bazi
 - baza raste
 - indeksiraš stare podatke
 - čitaš više nego što ti treba
 - u memoriji budu i podaci koji ti ne trebaju
- Koristan link: <http://www.databaseanswers.org/>

1

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
- Razmišljajte o podacima: procena količine podataka, koliko će se i kako menjati
- Uveriti biznis stranu da je to bitno, da treba vremena i da je to investicija u skalabilnost i stabilnost projekta
- Ozbiljne promene i redizajn su skupe, dugotrajne, neretko i mučne

2

NAMING CONVENTIONS

Imenovanje objekata

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

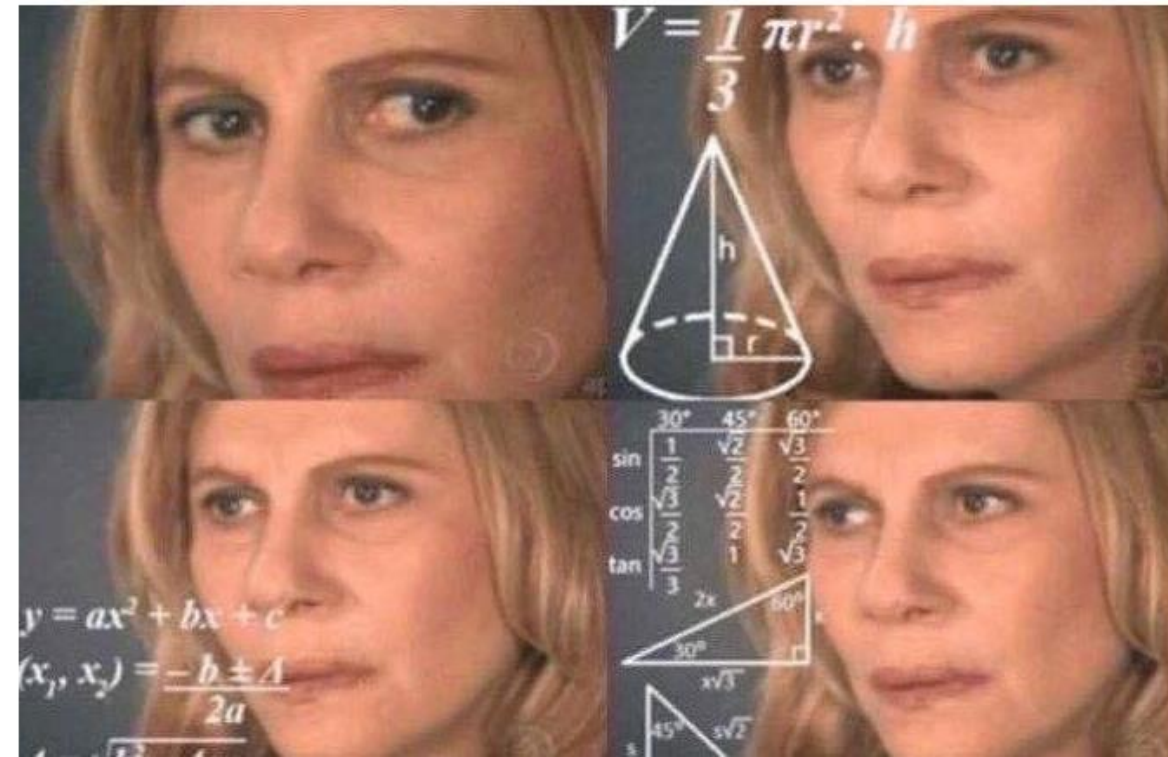
Imenovanje objekata

- Dijalog u vezi sa imenom kolone 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, znaš valjda da od 15. aprila u sistemu X podsistem Y koristimo samo UTC, a GMT samo u sistemu Z i to u verziji v3.x
- - A što onda kolonu nisi nazvao fStartDateUTC, pa bismo obojica bili pošteđeni ovog ispraznog dijaloga, ntk i tebe i konvenciju?!

Imenovanje objekata

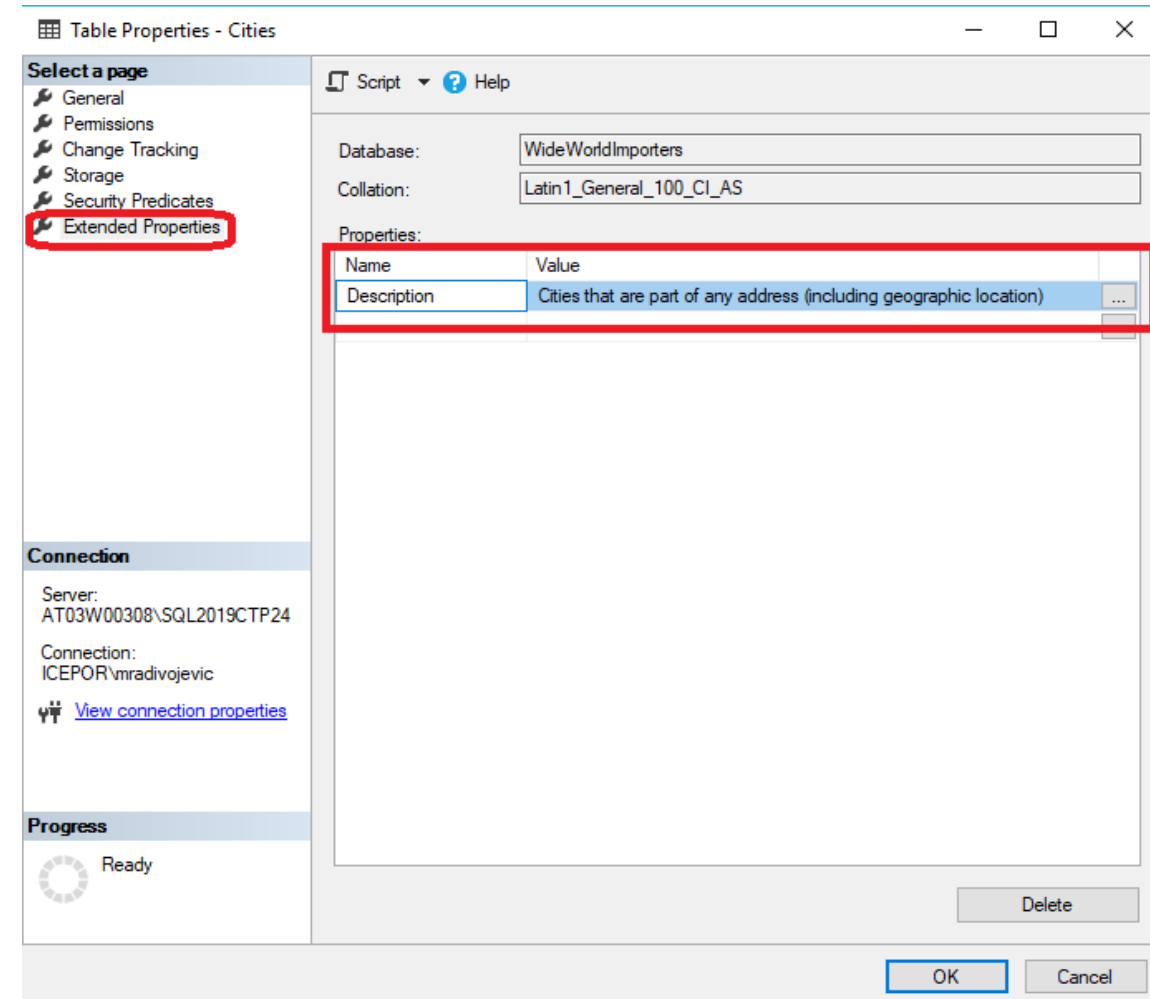
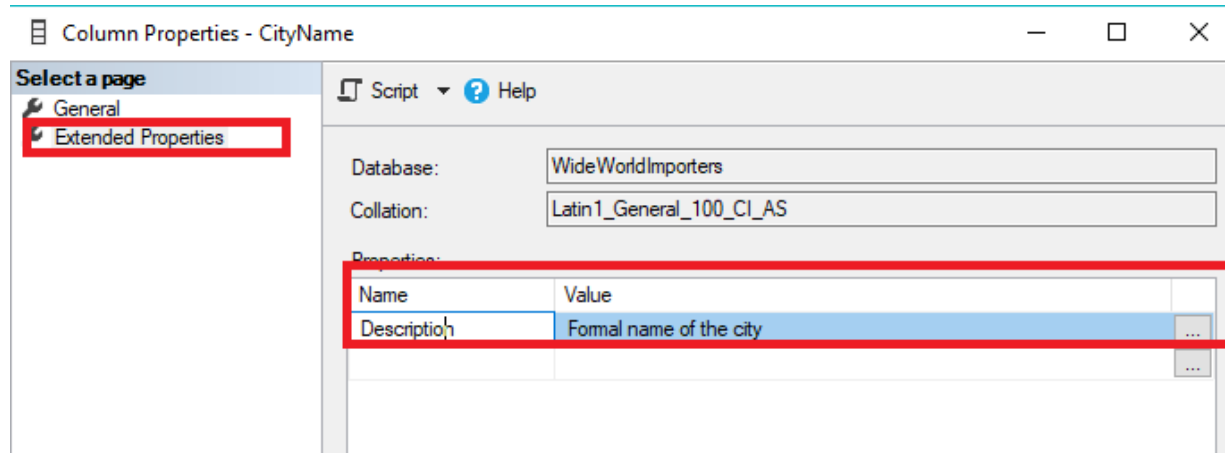
- Izbegavati:
 - Spejs karakter u imenu
 - Ključne reči
 - Skraćenice
 - Negacije (isValid umesto isValid)

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



Imenovanje objekata

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



3

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: 4294967295

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

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

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



Odabir odgovarajućeg tipa podataka

- Izaberite namanji tip podatka koji je dovoljan da pokrije potrebni domen
- Ako izabereš robusniji tip podataka
 - Potrebno je više prostora na disku na serveru i kopijama, bekap je sporiji...
 - SQL Server mora da učitati manje stranice za isti rezultat
- Izaberi 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
 - Ažuriranje parametara uskladištenih procedura koje koriste kolonu čiji se tip menja
 - Ažuriranje klase u aplikacionom kodu
 - Ažuriranje aplikacionog interfejsa prema podacima tzv. dejta akses lejera
- Efekti promene su teško merljivi i potpuno nevidljivi za krajnjeg korisnika kod koga se obično nalazi kesa sa novcima

Odabir odgovarajućeg tipa podataka

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

```
IF OBJECT_ID('dbo.insertStudentExams','U') IS NULL
    DROP PROCEDURE dbo.insertStudentExams
GO
CREATE PROCEDURE dbo.insertStudentExams
@student_id int, @exam_id int, @exam_note tinyint
AS
INSERT dbo.StudentExams VALUES (@student_id, @exam_id, @exam_note)
```

```
//se.ExamNote = dr.GetInt32(2);
se.ExamNote = (int) dr.GetByte(2);
```

4 UPOTREBA OGRANIČENJA (CONSTRAINTA)

Upotreba ograničenja (constraints)

- CHECK Constraints
 - Ograničite vrednosti koje su dozvoljene u okviru kolone
 - Npr. ako je kolona Ocena onda je smisleno postaviti ograničenje 1-5; tada niko ne može da unese ništa drugo
 - Svi korisnici će interpretirati vrednosti u koloni na isti način
 - Ako kolona ima tip podataka INT, bez ček konstrejnta, neko će kad-tad da upiše recimo 999 i to da interpretira po sopstvenom nađenju
 - Podiže kvalitet podataka za potonju analizu (DWH/BI)
- UNIQUE Constraints
 - Ako je nešto jedinstveno, upotrebite UNIQUE konstrejnt i podelite informaciju o jedinstvenosti sa RDBMS sistemom

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.

5

REDUKOVANJE NULL-ABILNIH 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: 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.

- JTŽMB
- 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.

- Programer jebe mater i gradu i datumu rođenja i bazi i sve upm

Primer fleksibilnosti

- Scena 2: programer omogućava INSERTU da radi
`ALTER TABLE Musterija ALTER COLUMN Grad NVARCHAR(50) NULL;`
Uvijek tako počinje...
- Programer pomisli ma komujebemater, skloniću i sa datuma rođenja
`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)`

- Programer pokido! Neka se i drugi raduju.
- Drugi:
`INSERT INTO Musterija(Ime, Prezime) VALUES('asdf', 'dfdfdfdggd');`

Epilog

- Sve radi, ALI

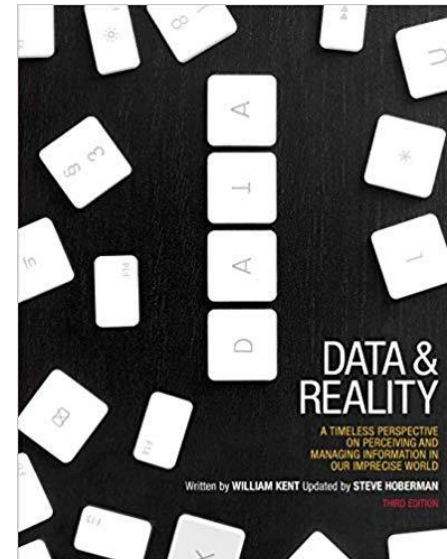
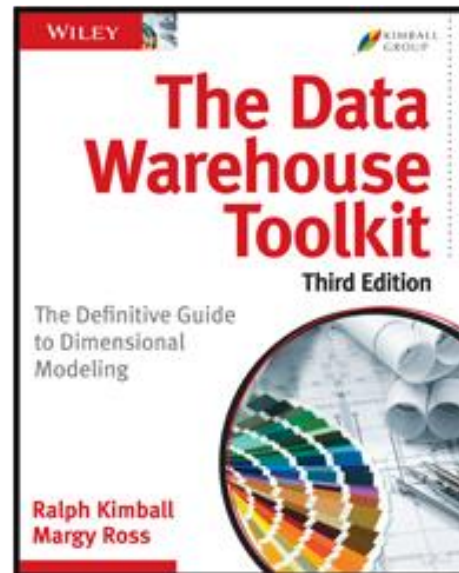
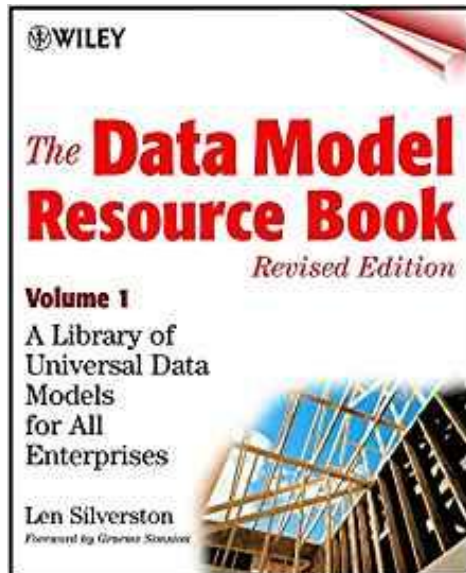
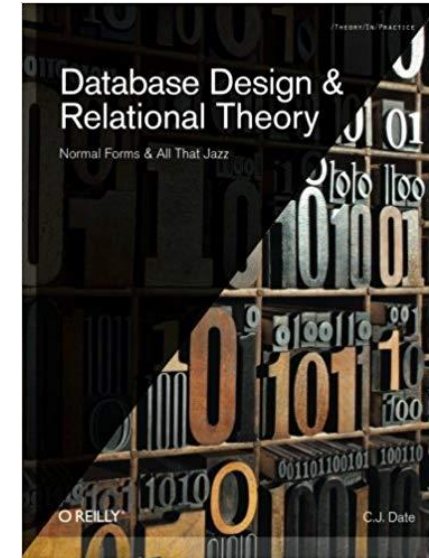
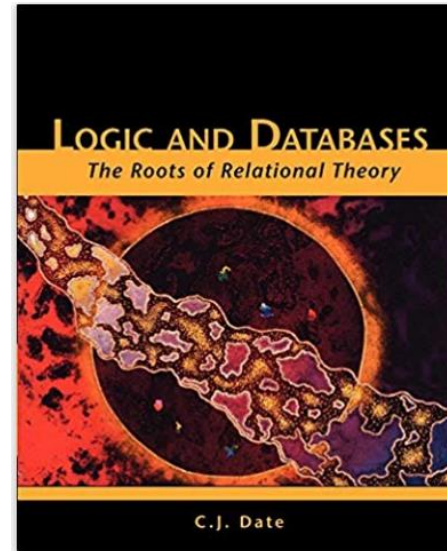
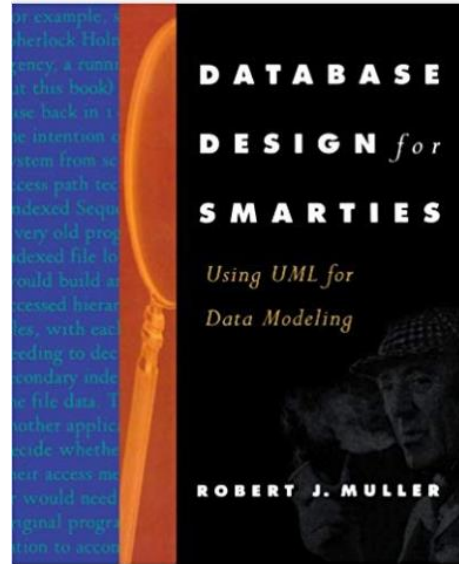
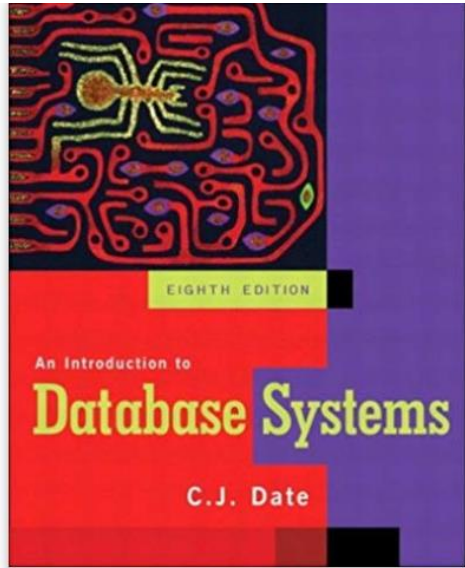
Id	Titula	Ime	Prezime	DatumRodjenja	Grad	Adresa	Telefon	Email
1	NULL	Mitar	Miric	1957-01-16 00:00:00.000	Beograd	Ulica u kojoj živi Mitar Miric	011123123	info@nemozenamnikonista.
4	NULL	Marinko	Rokvic	1954-01-27 00:00:00.000	NULL	NULL	NULL	NULL
5	NULL	asdf	dfdfdfdggd	NULL	NULL	NULL	NULL	NULL

- Koko imaš mušterija brt?
- Kako da ovi iz marketinga kontaktiraju ovog brta asdfa?

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 kolna 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

Preporuke - knjige




Preporuke - kursevi


<https://www.pluralsight.com>



Relational Database Design
by Hugo Kornelis Intermediate Jan 5 2014 7h 33m ★★★★★ (353)



SQL Server: Why Physical Database Design Matters
by Kimberly L. Tripp Intermediate Sep 26 2013 3h 56m ★★★★★ (506)



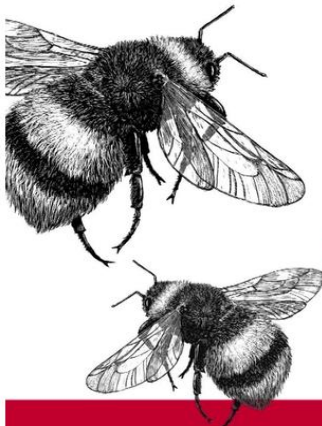
Course
Working With Temporal Data in SQL Server
Intermediate • 3h 6m • May 28, 2014

<https://www.oreilly.com/>

O'REILLY® Video Training Sign In Cart

An Introduction to Set Theory

An Introduction to Set Theory
A Chris Date Master Class



Set Theory Fundamentals
for Database Professionals
Part 1

O'REILLY

SUMMARY

- Implementacija integriteta podataka na nivou aplikacije
- Baza je trajnija od aplikacije
- „Poslovna logika pripada aplikaciji, baza je samo magacin sa podacima“
 - Jedan od opasnijih mitova u programiranju uopšte
 - Sorry brt, ali SELECT stejtment je implementacija brutalne biznis logike, čak i CREATE TABLE uvodi ozbiljne restrikcije u dozvoljeni tip i skup podataka
- Obezbedite sebi dovoljno vremena za implementaciju

SUMMARY – ZA SVAKU TABELU

- Pažljivo i konzistentno imenujte tabele i kolone poštujući konvenciju o imenovanju
- Za svaku kolonu u tabeli:
 - odaberite najmanji tip podatka koji pokriva domen atributa (ne uzimajte INT za kolone status, tip isl.)
 - Promene tipa podataka su skupe i niskog prioriteta u odnosu na druge zahteve, pa uglavnom ne dođe na red za implementaciju
- Ako kolona može da prihvata samo određene vrednosti, obavezno upotrebite CHECK constraint da to potpuno osigurate
 - nemojte se uzdati u validacije u aplikacijama koje koriste bazu!
- Redukujte broj kolona koje prihvataju NULL, to doprinosi kvalitetu i upotrebljivosti podataka