

# Базе података SQL сервер

Милорад Паскаш

`milorad.paskas@ict.edu.rs`

# Предавања

- *Предавања*: 3 часа седмично
- *Вежбе*: 3 часа седмично
- Материјали са предавања: *[kursevi.ict.edu.rs](http://kursevi.ict.edu.rs)*
- *Консултације*: мејл, платформа *Google Meet* (заказивање путем мејла) или уживо (уколико буду постојали услови за то)

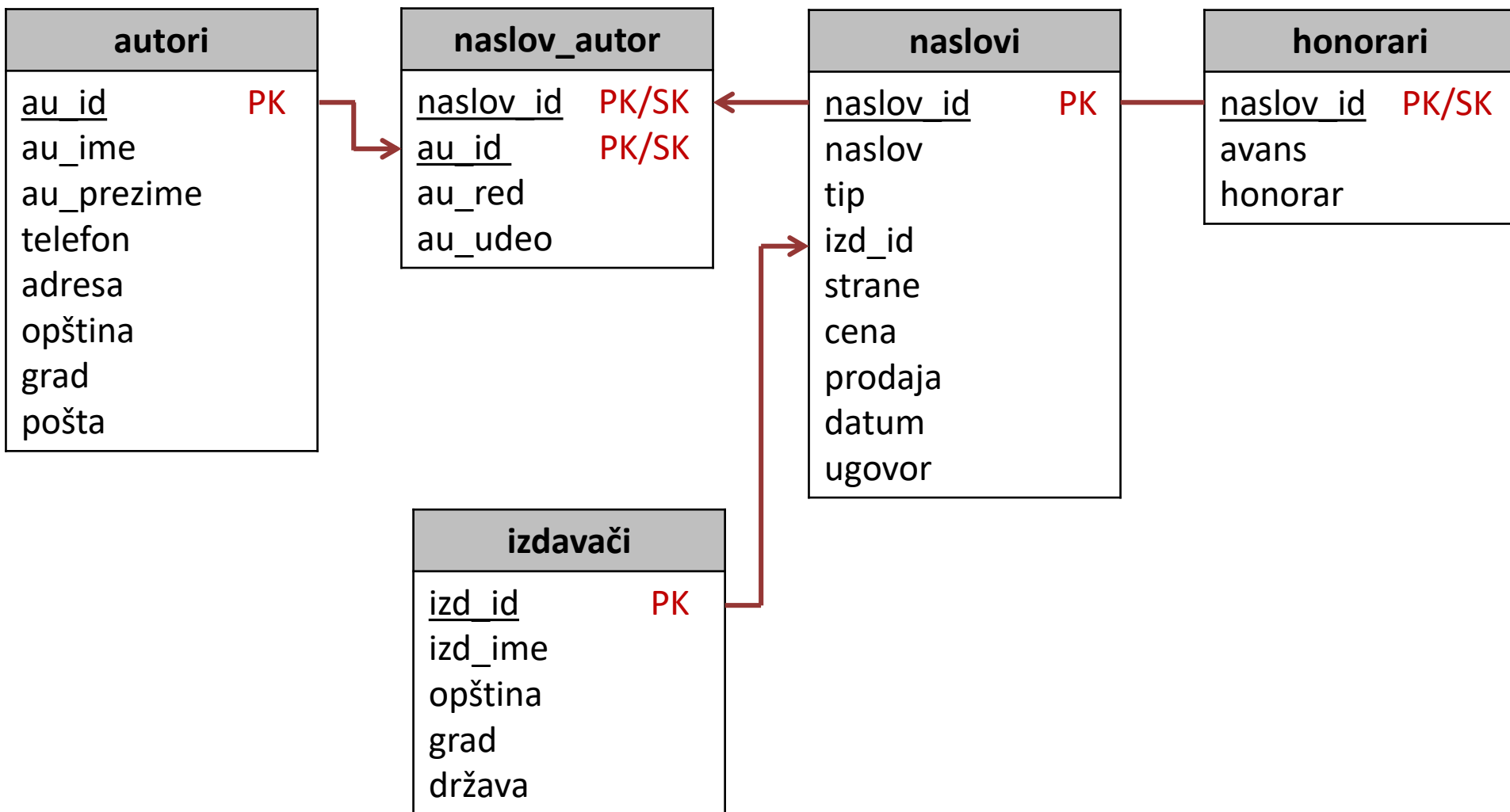
# Испитне обавезе

- Први колоквијум (**K1**):
  - на рачунару
  - 15 поена
- Други колоквијум (**K2**):
  - на рачунару
  - 35 поена
  - обавезан
- Испит:
  - услов:  $K1 + K2 > 25$  поена
  - на рачунару
  - 50 поена
  - освојених  $> 50\%$

# 1. Релациони модел

- Примери база података: резервације авионских карата, куповина преко интернета, база филмова, студентски сервиси на факултетима и др.
- База података представља скуп једне или више *табела*.
- Табела је структура која садржи податке. Подаци у свакој табели се односе на одређен *ентитет*. Нпр. табела која чува податке о студентима (ентитет) ће садржавати податке који описују тај ентитет: име, презиме, датум рођења студента и сл.
- Свака табела има јединствено име у оквиру једне базе података.

*Пример:*



- Табела је организована у виду дводимензионалне мреже коју чине *редови* и *колоне*. Табела садржи најмање једну колону и ниједан или више редова. Табела без редова је *празна табела*.
- Пример табеле **authori**:

au_id	au_ime	au_prezime
A01	Marija	Pavić
A02	Dragana	Kostić
...	...	...

- Свака колона представља једну особину ентитета табеле.
- Редослед колона је *небитан*, односно колоне не могу да се позивају по свом редоследу већ само по називу.
- Свака колона подразумева ограничења с обзиром на податке у тој колони с обзиром на њихов тип, дужину, формат, јединственост, опсег вредности и др.
- Име сваке колоне је јединствено у табели, док две различите табеле из исте базе могу да садрже једну или више колона са истим именом.

au_id	au_ime	au_prezime
A01	Marija	Pavić
A02	Dragana	Kostić
...	...	...



au_prezime	au_id	au_ime
Kostić	A02	Dragana
Pavić	A01	Marija
...	...	...

- Ред садржи конкретну вредност.
- Редослед редова је небитан.
- Не постоје два идентична реда у табели. То гарантује посебна колона (*примарни кључ*).

au_id	au_ime	au_prezime
A01	Marija	Pavić
A02	Dragana	Kostić
A03	Tanja	Marković
A04	Nikola	Marković



- У пракси се број редова често мења, односно табеле се тако похрањују подацима. У примеру табеле *autori*, сваког месеца се додају нови аутори (нова имена, презимена и др.).
- Колоне се релативно ретко мењају, тј. ретко се бришу неке од постојећих или додају нове колоне. Један од ризика је нарушавање интегритета/јединствености редова.
- Дефинисање колона зависи од потреба корисника.  
Нпр, у табели *autori* бројеви телефона могу да се уносе у једној колони, *telefon*, или у више колона (нпр. колоне *pozivni\_broj* и *tel\_broj*).

## Примарни кључ

- База података има смисла само уколико је сваки податак (вредност) у њој доступан.
- Конкретној вредности можемо приступити само уколико познајемо табелу, колону и ред у којој се та вредност налази. Колоне у табели носе јединствене називе, док редовима приступамо захваљујући *примарном кључу*, који је:

*неопходан*: свака табела садржи тачно један примарни кључ. У табелама не постоји редослед редова (у смислу претходни и следећи ред), па се оне не могу позивати по позицији у табели;

*јединствен*: не постоје два реда са истим примарним кључем;

*прост* или *композитан*: прост је уколико садржи само једну колону, а композиран уколико садржи две или више колона;

његова вредност не може бити празно поље. У случају композираног кључа, вредност ниједне колоне не сме бити празно;

*стабилан*: вредности примарног кључа се ретко мењају. Када се избрише неки ентитет, вредност његовог примарног кључа се не узима за неки други ентитет;

*минималан*: примарни кључ садржи само онолико колона колико је потребно да би био јединствен.

- Лоше изабран примарни кључ може онемогућити додавање редова у табелу. Избор примарног кључа зато је, најчешће, у рукама креатора базе.

- Пример:

Примарни кључ



au_id	au_ime	au_prezime
A01	Marija	Pavić
A02	Dragana	Kostić
A03	Tanja	Marković
A04	Nikola	Marković

Nikola	Marković
--------	----------

У оригиналној табели редови су различити. Зато примарни кључ може бити композитан и да га чине све колоне.

Уколико би корисник касније додао ред са вредностима које се понављају, та два реда не бисмо могли да разликујемо.

- Најчешћи називи примарних кључева укључују речи: `id`, `key`, `num` и сл.
- Креатори база, најчешће, избегавају уобичајене идентификаторе за примарне кључеве, као што је ЛМБГ (један од разлога је и приватност). Обично се користе интерни кључеви, који имају смисла само унутар базе.

*Пример:* примарни кључ у табели `authori` може бити тренутак (датум и време) додавања у табелу.

## Страни кључ

- Страни кључ служи како би се повезало више различитих табела у бази.
- То је, заправо, колона/колоне у табели чије вредности референцирају вредности у некој другој табели. На тај начин, редови једне табеле имају одговарајуће редове у другој табели.
- Табела која садржи страни кључ је *референцирајућа* или табела *дете*, док је она друга табела *референцирана* или табела *родитељ*.
- Вредност страног кључа **може** бити празно поље.
- Име колоне страног кључа може имати другачији назив од колоне примарног кључа у табели родитеља.
- Вредности страног кључа нису јединствене у табели.

- *Пример:*

### izdavači

izd_id	izd_ime
I01	Akademski pregled
I02	Novi izdavači
I03	Milan Press
I04	ABC Knjige

### naslovi

naslov_id	naslov	izd_id
N01	1918.	I01
N02	Veliki rat	I03
N03	Relacione baze	I04
N04	Jungove ličnosti	I04

Примарни кључ

Страни кључ

# Релације

- Како би се табеле, односно редови табела, повезале користе се релације. Оне се деле у три групе:
  - 1) релација један на један;
  - 2) релација један ка више;
  - 3) релација више ка више.



## Релација један на један

- Код релација један на један сваком реду једне табеле одговара највише један ред друге табеле и обрнуто.
- Ова релација се користи када примарни кључ посматране табеле референцира примарни кључ друге табеле, односно када је он истовремено и страни кључ посматране табеле.
- Једноставније решење би било да се сви подаци из две табеле са овом врстом релације обједине у једну табелу.  
**Мане:** успорава се извршење упита (обимна табела), угрожава се поверљивост података, постојање празних поља.

- *Пример:*

**naslovi**

naslov_id	naslov	izd_id
N01	1918.	I01
N02	Veliki rat	I03
N03	Relacione baze	I04
N04	Jungove ličnosti	I04

**zarade**

naslov_id	avans
N01	50000
N03	35000

naslovi
<u>naslov_id</u> naslov izd_id

zarade
<u>naslov_id</u> avans

## Релација један ка више

- Код овог типа релација сваком реду табеле 1 одговара више ( $\geq 0$ ) редова табеле 2, док сваком реду табеле 2 одговара само један ред табеле 1.
- Овај тип релације се јавља код табела када је примарни кључ табеле 1 страни кључ табеле 2.

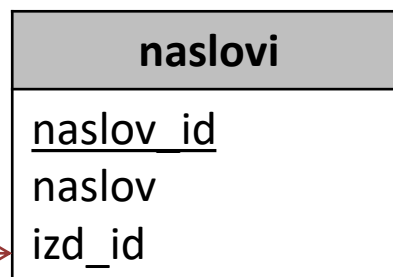
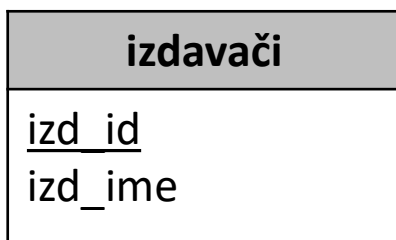
- *Пример:*

**izdavači**

izd_id	izd_ime
I01	Akademski pregled
I02	Novi izdavači
I03	Milan Press
I04	ABC Knjige

**naslovi**

naslov_id	naslov	izd_id
N01	1918.	I01
N02	Veliki rat	I03
N03	Relacione baze	I04
N04	Jungove ličnosti	I04

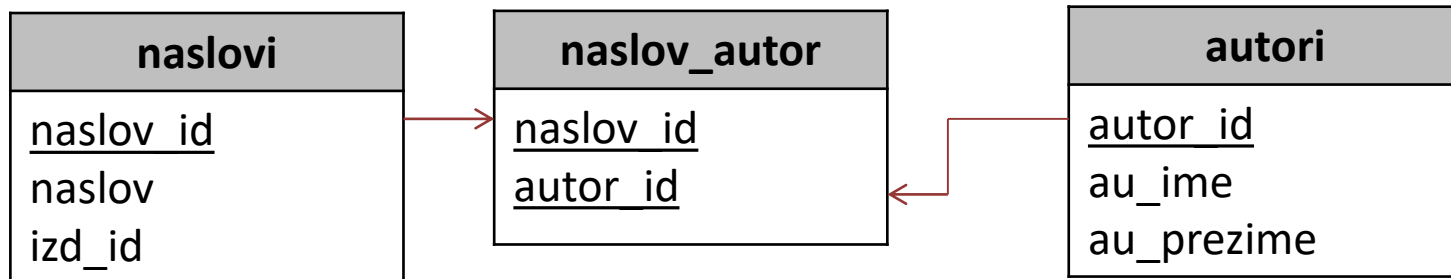


## Релација више ка више

- Подразумева се да сваком реду табеле 1 одговара више редова табеле 2 и да сваком реду табеле 2 одговара више редова табеле 1.
- Ова врста релације се може остварити само кроз трећу табелу (табела 3) која ће “спојити” табеле 1 и 2. Овим се релација типа “више ка више” разлаже на две релације типа “један ка више”.

- *Пример:*

naslovi		naslov_autor		autori		
naslov_id	naslov	naslov_id	autor_id	au_id	au_ime	au_prezime
N01	1918.	N01	A01	A01	Marija	Pavić
N02	Veliki rat	N02	A01	A02	Dragana	Kostić
N03	Relacione baze	N02	A02			
		N03	A02			



# Нормализација

- Уколико би сви подаци из базе података били смештени у једну једину табелу, тада би се многи подаци појављивали више пута у табели. Таква база би имала “лошу” структуру и садржала би сувишне (*редундантне*) податке. Другим речима, зависности између вредности колона нису добро дефинисане.
- Последице таквог приступа су: база (табела) постаје прегломазна (што успорава упите), отежано је ажурирање базе (исти податак треба изменити на више места).
- Решење: измештање неких колона у нову табелу, односно декомпозиција (подела) табеле на више табела.

# Нормализација

- Сличан проблем се јавља и у случају када се база састоји из више табела, али су неки подаци и даље редундантни. Да би се редови у таквим табелама избрисали или ажурирали, потребно је водити рачуна о местима на којима се појављују редундантни подаци.
- Да би се оптимизовао рад са табелама, врши се елиминација редундантности која се назива *нормализацијом*. Она се спроводи у корацима, и у сваком кораку табела је у одређеном степену нормализације, тј. у једној од *нормалних форми (НФ)*: 1НФ, 2НФ и 3НФ.



# Нормализација

- Свака наредна нормална форма је строжа од претходне. Зато више нормалне форме, у односу на ниже, имају већи број табела (декомпозиција).
- Процес нормализације се одвија итеративно (у фазама које се понављају) кроз поделу и спајање табела. Крај нормализације зависи од процене (и искуства) креатора базе.

## Прва нормална форма (1НФ)

- Табела има форму 1НФ ако:
  - 1) колоне садрже само скаларне вредности (не могу се даље делити/декомпоновати);
  - 2) не постоје “понављајуће групе”.
- Решење: декомпоновати (разбити) нескаларне вредности и понављајуће групе.

- *Пример: Колоне са нескаларним вредностима*

**naslovi**

naslov_id	naslov	autori
N01	1918.	A01,A02
N02	Veliki rat	A01,A02,A03
N03	Relacione baze	A04
N04	Jungove ličnosti	A05

- *Пример: Понављајуће групе*

**naslovi**

naslov_id	naslov	autor1	autor2	autor3
N01	1918.	A01	A02	
N02	Veliki rat	A01	A02	A03
N03	Relacione baze	A04		
N04	Jungove ličnosti	A05		

- *Пример: Решење по 1НФ*

naslovi		naslov_autor	
naslov_id	naslov	naslov_id	au_id
N01	1918.	N01	A01
N02	Veliki rat	N01	A02
N03	Relacione baze	N02	A01
N04	Jungove ličnosti	N02	A02
		N02	A03
		N03	A04
		N04	A05

- Пример: Решење по 1НФ

naslovi		naslov_autor	
naslov_id	naslov	naslov_id	au_id
N01	1918.	N01	A01
N02	Veliki rat	N01	A02
N03	Relacione baze	N02	A01
N04	Jungove ličnosti	N02	A02
		N02	A03
		N03	A04
		N04	A05



## Прва нормална форма (1НФ)

- Са 1НФ избегавају се следећи проблеми:
  - јединствена вредност у пресеку сваке колоне и редове
  - у случају вишеструких вредности у пресеку колоне и редова тешко је приступити једној од вредности, што доводи до лошијих перформанси базе (спорије извршавање упита);
  - ажурирање табела са вишеструким вредностима је отежано, јер се тада мора водити рачуна о редоследу тих вредности.

## Друга нормална форма (2НФ)

- Табела има форму 2НФ ако:

1) табела задовољава 1НФ услове;

2) једна од вредности композитног кључа не сме да одређује (условљава) вредности колоне које нису део тог кључа (*парциона функционална зависност*).

- **Решење:** колоне које зависе само од дела композитног кључа изместити из табеле (постиже се *потпуна функционална зависност* од композитног кључа).



## Друга нормална форма (2НФ)

- Уколико 1НФ табела нема композитни, већ прост примарни кључ тада она сигурно задовољава 2НФ.
- Уколико су све колоне 1НФ табеле део примарног кључа, тада она сигурно задовољава услове за 2НФ.

- Пример:

naslov_autor
<u>naslov_id</u>
<u>au_id</u>
au_tel
tip
au_redosled

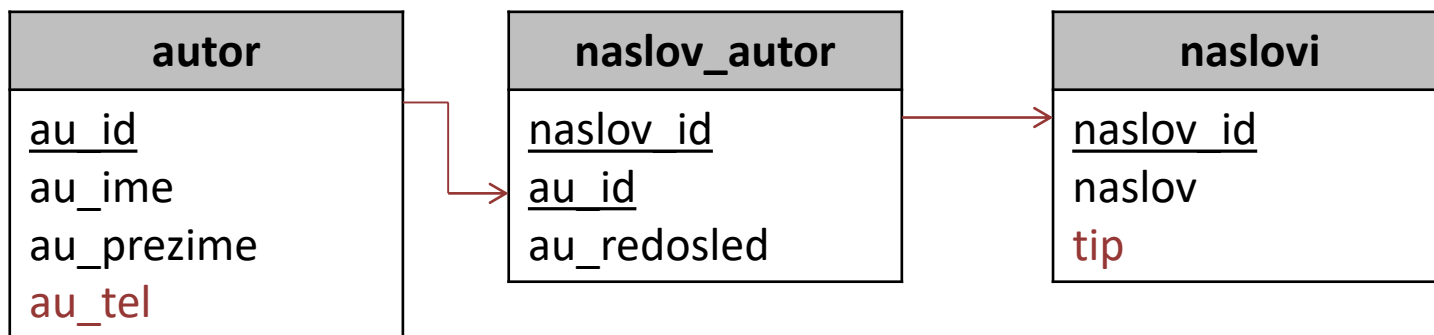
Телефон аутора (**au\_tel**)  
зависи само од аутора (**au\_id**),  
али не и од наслова (**naslov\_id**).  
И тип књиге (**tip**) је парционо зависан од  
наслова књиге (**naslov\_id**).

- Пример:



Телефон аутора (**au\_tel**)  
зависи само од аутора (**au\_id**),  
али не и од наслова (**naslov\_id**).  
И тип књиге (**tip**) је парционо зависан од  
наслова књиге (**naslov\_id**).

- Решење:



## Трећа нормална форма (3НФ)

- Табела има форму 3НФ ако:

1) табела задовољава 2НФ услове;

2) вредности колона које нису део кључа не смеју да одређују (условљавају) вредности других колона које, такође, нису део кључа (*транзитивна зависност*).

- **Решење:** колоне које зависе једна од друге (а нису део кључа) изместити из табеле.

- *Пример:*

naslovi
<u>naslov_id</u>
naslov
izd_id
izd_država

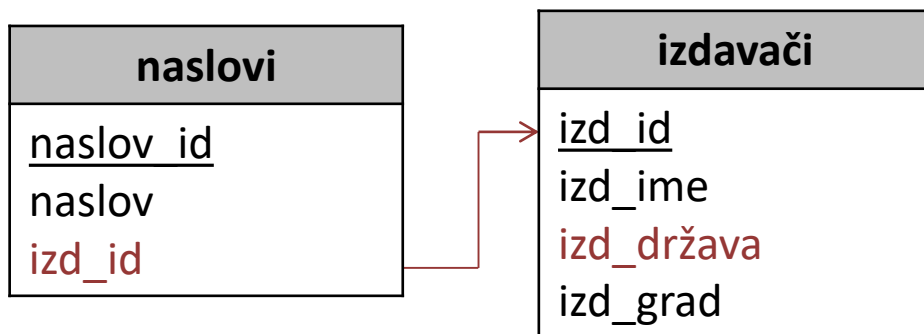
Држава издавача(**izd\_država**)  
зависи од издавача (**izd\_id**).

- Пример:

naslovi
<u>naslov_id</u>
naslov
izd_id
izd_država

Држава издавача(**izd\_država**)  
зависи од издавача (**izd\_id**).

- Решење:



## Остале нормалне форме

У релационом моделу се дефинишу и наредне нормализационе форме, како би се елиминисали други облици редундантности:

- Бојс-Кодова (*Boyce-Codd*) нормална форма
- 4НФ
- 5НФ

Релациони модел их не захтева.

## Нормалне форме: Напомене

У процесу нормализације број табела расте што захтева додатно повезивање колона табела. То додатно успорава упите.

Како би се постигао компромис између редундантности података и брзине извршавања упита, користи се *денормализација*.



## 2. SQL: Типови података

- SQL је стандардизован програмски језик за рад (креирање, ажурирање, претраживање података) са базама података.
- SQL се сматра скраћеницом за “*Structured Query Language*”.
- Постоји више софтверских система за управљање базама података који користе SQL језик. Сваки од система има специфичности које га разликују од осталих на тржишту: *MS Access*, *MS SQL Server*, *Oracle*, *MySQL*, *IBM DB2*, *SAS*, *Firebird*, *SQLite* и др.

- SQL је јавно доступан стандард (*ISO/IEC 9075*) који се појавио 1986. године. Потом су уследиле нове побољшане верзије овог стандарда. Последња верзија је из 2016. године.
- Постојећи системи за управљање базама података користе (*Core*) SQL стандард, али уносе и одређене измене и одступања од стандарда. То значи да упити креирани у једном систему, највероватније, неће моћи да се покрену у неком другом систему за управљање базама података, већ ће бити потребна корекција упита.

- За сваку колону се дефинише тип вредности који се може наћи у тој колони.
- Свака колона има само један тип података.
- Тип података у колони, истовремено, дефинише и операције које се могу применити на податке те колоне.
- Уколико се за колону при креирању табеле изабере неодговарајући тип податка, каснијом променом типа податка те колоне се губе подаци из те колоне.
- У зависности од типа података у колони зависиће и тип сортирања.

- Основне категорије типова података:
  1. стрингови
  2. прецизни нумерички
  3. нумерички са покретним зарезом
  4. бинарни нумерички
  5. бинарни подаци
  6. подаци о датуму и времену
  7. (интервални)

# Стрингови

Овај тип података се користи за унос текста. У оквиру *MS SQL Server*-а подразумевају се следећи стрингови:

- **CHAR(L)**: дефинише стринг са фиксним бројем карактера дужине  $L$ . Уколико се унесе стринг дужине мање од  $L$ , он ће бити допуњен размацама до дужине  $L$ .

*Примери:*

CHAR(4) : 'Test'

CHAR(8): 'Test '

CHAR(4): '2018'

CHAR(16): 'Znak navoda je "'. '

# Стрингови

- **VARCHAR[(L|max)]**: дефинише стринг променљиве дужине са максималном дужином  $L$ . Уколико се унесе стринг дужине мање од  $L$ , он **неће** бити допуњен размацима до дужине  $L$ . Ако се уместо дужине  $L$  унесе ознака **max** тада ће се омогућити унос већег текста (до 2GB).

*Примери:*

VARCHAR(4) : 'Test'

VARCHAR(8): 'Test'

VARCHAR(9): '2018'

VARCHAR(0): ''

# Стрингови

- **NCHAR(L)**: дефинише низ карактера са неенглеског говорног подручја (*Unicode*) фиксне дужине  $L$ . Уколико се унесе стринг дужине мање од  $L$ , он ће бити допуњен размацима до дужине  $L$ .

*Примери:*

NCHAR(5) :    N'Lišće'

NCHAR(9):    N'Lišće    '

NCHAR(8):    N'Шума    '

# Стрингови

- **NVARCHAR(L)**: дефинише стринг са неенглеског говорног подручја (*Unicode*) променљиве дужине са максималном дужином  $L$ . Уколико се унесе стринг дужине мање од  $L$ , он **неће** бити допуњен размацима до дужине  $L$ . Ако се уместо дужине  $L$  унесе ознака **max** тада ће се омогућити унос већег текста (до 2GB).

*Примери:*

NVARCHAR(5) :	N'Lišće'
NVARCHAR(9):	N'Lišće'
NVARCHAR(8):	N'Шума'



## Прецизни нумерички тип

Овај тип података се користи за унос (тачних) нумеричких вредности. У оквиру *MS SQL Server*-а подразумевају се следећи прецизни нумерички типови:

- **INT, BIGINT, SMALLINT, TINYINT**: дефинишу се цели бројеви и то у опсегу:  $-2^{31} \div 2^{31}-1$ ,  $-2^{63} \div 2^{63}-1$ ,  $-2^{15} \div 2^{15}-1$ ,  $0 \div 2^8-1$ , редом.

## Прецизни нумерички тип

- **NUMERIC[(p[,s])]** и **DECIMAL[(p[,s])]**: дефинишу се децимални бројеви у опсегу  $-10^{38}+1 \div 10^{38}-1$ . Број цифара са леве стране од зареза је **p-s**, а са десне **s**. Ако **s** није дефинисано подразумевана вредност му је 0.

*Пример: Интерпретација броја 55,67*

NUMERIC(4) :	56
NUMERIC(4,0) :	56
NUMERIC(3,2) :	грешка
NUMERIC(3,1) :	55,7
NUMERIC(1,0) :	грешка

## Нумерички тип са покретним зарезом

Овај тип података се користи за унос реалних бројева (са, теоријски, бесконачно много децимала). Они су, најчешће, резултат математичких израчунавања. У оквиру *MS SQL Server*-а подразумевају се следећи нумерички типови са покретним зарезом:

- **FLOAT[(n)]**: дефинишу се реални бројеви. **n** означава број бита потребних за меморисање **мантисе**:  $1 < n < 24$  за 7 цифара,  $25 < n < 53$  за 15 цифара прецизности.

$$7,35 * 10^3 = 7,35E3 = 7350$$

- **REAL**: исто што и **FLOAT(24)**.

## Бинарни нумерички тип

Овај тип података се користи за бинарне бројеве (0,1), а може да има и вредност *NULL* (неодређено):

- **BIT**: дефинишу се бројеви 0 и 1, а дозвољено је и *NULL*.

## Бинарни подаци

Овај тип података се користи за складиштење великих фајлова мултимедијалног типа (слика, звук, видео) или научних података (медицинска слика и сл.).

- **BINARY[(n)]:** дефинишу се подаци фиксне дужине **n** бајтова ( $1 < n < 8000$ ).
- **VARBINARY[(n|max)]:** дефинишу се подаци променљиве дужине са максималном дужином од **n** бајтова ( $1 < n < 8000$ ). Ако се уместо **n** стави **max** онда је максимална дужина податка  $2^{31}-1$  бајтова.

## Подаци о датуму и времену

Овај тип података се користи за приказивање датума и времена (0-24ч).

- **DATETIME**: приказују се датум и време, са децималном прецизношћу секунди, у формату:

ГГГГ-ММ-ДД чч:мм:сс[.mmm].

- **SMALLDATETIME[(n|max)]**: приказују се датум и време, без децималне прецизности секунди, у формату:

ГГГГ-ММ-ДД чч:мм:сс.

*Примери:*

2018-04-25 10:15:32.999

'1995-07-15'

Непознате (неодређене) вредности: *NULL*

За приказивање вредности које су недефинисане или које недостају, користи се реч **NULL**.

Када се вредности уносе у табелу, за вредности неке колоне је потребно предвидети да неће увек (у свим редовима) бити познате при уносу, али ни касније.

**NULL** није исто што и нумеричка вредност 0 (нула) или празан знаковни низ ''.

Примарни кључ не сме садржавати **NULL**. У било којој колони можемо забранити унос вредности **NULL**.

Непознате (неодређене) вредности: *NULL*

Више “различитих” **NULL** не може да се пореди (нису међусобно исти). **NULL** није вредност.

Аритметичке операције са **NULL** даће као резултат **NULL**.



Табела: **autori**

Јединствени идентификатор аутора. Тип: <b>CHAR(3)</b>	Име аутора. Тип: <b>NVARCHAR(15)</b>	Презиме аутора. Тип: <b>NVARCHAR(15)</b>	Телефон аутора. Тип: <b>VARCHAR(12)</b> <b>NULL</b>	Адреса аутора. Тип: <b>NVARCHAR(20)</b> <b>NULL</b>	Општина аутора. Тип: <b>NVARCHAR(15)</b> <b>NULL</b>	Град аутора. Тип: <b>NCHAR(2), NULL</b>	Поштански број. Тип: <b>CHAR(5)</b> <b>NULL</b>
<u>au_id</u>	au_ime	au_prezime	telefon	adresa	opština	grad	pošta
A01	Marija	Pavić	0117352881	27. Marta 73	Palilula	BG	11120
A02	Dragana	Kostić	0324471358	Karađorđeva 77	Čačak	ČA	32013
A03	Tanja	Marković	0212548140	Nikole Tesle 115	Sremska Kamenica	NS	21010
A04	Nikola	Marković	0219813522	Vuka Karadžića 57	Sremska Kamenica	NS	21102
A05	Marko	Petrov	0117512442	Svetogorska 18	Stari grad	BG	11115
A06		Petrović	0218223511	Dunavska 5	Novi Sad	NS	21131
A07	Petar	Milenković	0184428513	Glavna 171	Medijana	NI	18100

## Табела: **izdavači**

Јединствени  
идентификатор  
издавача. Тип:  
**CHAR(3)**

Име издавача. Тип:  
**NVARCHAR(20)**

Општина издавача. Тип:  
**NVARCHAR(15), NULL**

Град издавача.  
Тип: **NCHAR(2)**

Држава. Тип:  
**NCHAR(3)**

<u>izd_id</u>	izd_ime	opština	grad	država
I01	Akademski pregled	Stari grad	BG	SRB
I02	Novi izdavači	Sremska Kamenica	NS	SRB
I03	Milan Press	NULL	MI	ITA
I04	ABC Knjige	Sremska Kamenica	NS	SRB

# Табела: **naslovi**

Јединствени идентификатор наслова. Тип: <b>CHAR(3)</b>	Наслов. Тип: <b>NVARCHAR(40)</b>	Област књиге. Тип: <b>NVARCHAR(10), NULL</b>	Ј. идентификатор издавача. Тип: <b>CHAR(3)</b>	INT, NULL	Цена књиге. Тип: <b>DECIMAL(6,2)</b> NULL	Број продатих књига. Тип: <b>INT, NULL</b>	Датум објављивања. Тип: <b>DATETIME, NULL</b>	Податак да ли је аутор потписао уговор. Тип: <b>SMALLINT</b>
<u>naslo v_id</u>	naslov	tip	izd _id	strane	cena	prodaja	datum	ugovor
N01	1918.	istorija	I01	520	789.99	605	2016-08-01	1
N02	Veliki rat	istorija	I03	350	659.99	1300	2014-04-01	1
N03	Baze podataka	računarstvo	I02	337	500.00	515	2016-09-01	1
N04	Jungove ličnosti	psihologija	I04	225	329.99	1260	2015-05-31	1
N05	Snaga volje	psihologija	I04	120	379.99	1135	2017-01-01	1
N06	Dobro jutro	biografija	I01	230	449.99	1523	2016-07-31	1
N07	Sve o mojoj majci	biografija	I03	157	399.99	1057	2015-10-01	1
N08	Na farmi	deca	I04	25	299.99	875	2017-06-01	1

# Табела: **naslovi**- наставак

Јединствени идентификатор наслова. Тип: <b>CHAR(3)</b>	Наслов. Тип: <b>NVARCHAR(40)</b>	Област књиге. Тип: <b>NVARCHAR(10), NULL</b>	Ј. идентификатор издавача. Тип: <b>CHAR(3)</b>	INT, NULL	Цена књиге. Тип: <b>DECIMAL(6,2)</b> NULL	Број продатих књига. Тип: <b>INT, NULL</b>	Датум објављивања. Тип: <b>DATETIME, NULL</b>	Податак да ли је аутор потписао уговор. Тип: <b>SMALLINT</b>
<u>naslo v_id</u>	naslov	tip	izd _id	strane	cena	prodaja	datum	ugovor
N09	Reci: Zašto?	deca	I04	85	329.99	923	2018-05-31	1
N10	Srpski pisci	biografija	I01	NULL	NULL	NULL	NULL	0
N11	Moć podsvesti	psihologija	I04	230	459.99	883	2016-11-30	1
N12	Među maskama	biografija	I01	205	559.99	1038	2016-08-31	1
N13	Osvrt na 1980-e	istorija	I03	307	659.99	754	2015-05-31	1

## Табела: **naslov\_autor**

Јединствени  
идентификатор  
наслова. Тип:  
**CHAR(3)**

Редни број  
аутора на  
књизи. Тип:  
**SMALLINT**

Ј. идентификатор  
аутора. Тип:  
**CHAR(3)**

Расподела хонорара (у %)  
међу ауторима. Тип:  
**DECIMAL(5,2)**

naslov\_id

au\_id

au\_red

au\_udeo

N01

A01

1

1.00

N02

A01

1

1.00

N03

A05

1

1.00

N04

A03

1

0.60

N04

A04

2

0.40

N05

A04

1

1.00

N06

A02

1

1.00

N07

A02

1

0.50

## Табела: **naslov\_autor**- наставак

Јединствени  
идентификатор  
наслова. Тип:  
**CHAR(3)**

Редни број  
аутора на  
књизи. Тип:  
**SMALLINT**

Ј. идентификатор  
аутора. Тип:  
**CHAR(3)**

Расподела хонорара (у %)  
међу ауторима. Тип:  
**INT**

naslov\_id

au\_id

au\_red

au\_udeo

N07	A04	2	0.50
N08	A06	1	1.00
N09	A06	1	1.00
N10	A02	1	1.00
N11	A03	2	0.30
N11	A04	3	0.30
N11	A06	1	0.40
N12	A02	1	1.00
N13	A01	1	1.00

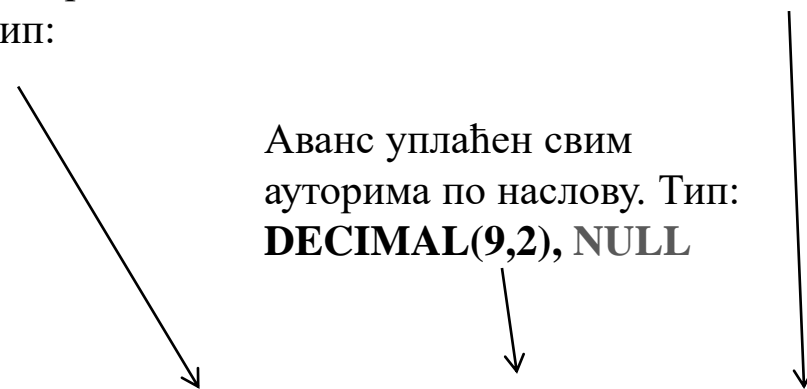
# Табела: **honorari**

Јединствени  
идентификатор  
наслова. Тип:  
**CHAR(3)**

Хонорар исплаћен свим  
ауторима по наслову. Тип:  
**DECIMAL(5,2), NULL**

Аванс уплаћен свим  
ауторима по наслову. Тип:  
**DECIMAL(9,2), NULL**

Табела садржи  
податке о хонорару  
исплаћеном свим  
ауторима по наслову,  
као и о укупном  
авансу плаћеном  
ауторима по наслову.



<u>naslov_id</u>	avans	honorar
N01	50000.00	0.05
N02	5000.00	0.06
N03	75000.00	0.07
N04	100000.00	0.08
N05	500000.00	0.09
N06	100000.00	0.08
N07	5000000.00	0.11
N08	0.00	0.04

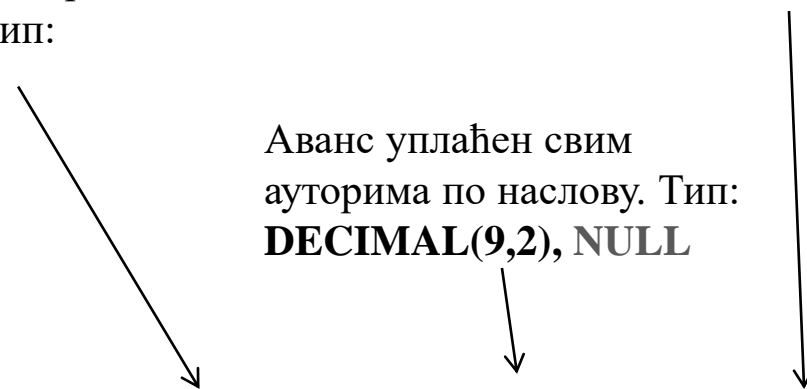
## Табела: **honorari**-наставак

Јединствени  
идентификатор  
наслова. Тип:  
**CHAR(3)**

Хонорар исплаћен свим  
ауторима по наслову. Тип:  
**DECIMAL(5,2), NULL**

Аванс уплаћен свим  
ауторима по наслову. Тип:  
**DECIMAL(9,2), NULL**

Табела садржи  
податке о хонорару  
исплаћеном свим  
ауторима по наслову,  
као и о укупном  
авансу плаћеном  
ауторима по наслову.



<u>naslov_id</u>	avans	honorar
N09	0.00	0.05
N10	NULL	NULL
N11	500000.00	0.07
N12	250000.00	0.09
N13	100000.00	0.06



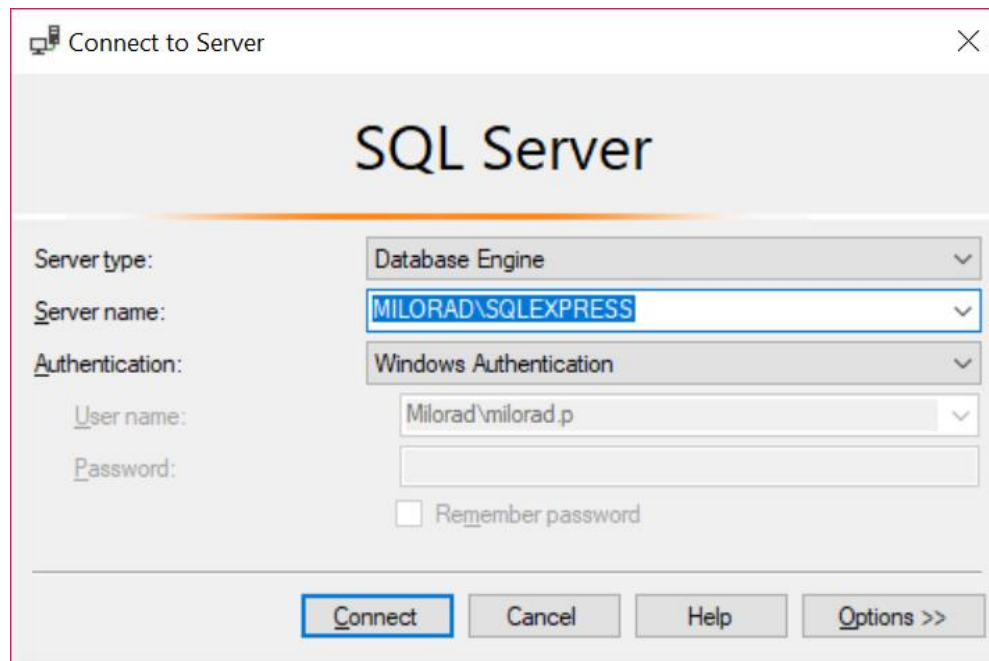
# 3. MS SQL Server

- Постоји више софтверских система за управљање базама података који користе SQL језик. Сваки од система има специфичности које га разликују од осталих на тржишту: *MS Access*, ***MS SQL Server***, *Oracle*, *MySQL*, *IBM DB2*, *SAS*, *Firebird*, *SQLite* и др.
- За рад са базама података надаље ћемо користити *MS SQL Server* (***Express edition***). Инсталација: <https://www.microsoft.com/en-us/sql-server/sql-server-editions-express>  
<https://www.microsoft.com/en-us/download/details.aspx?id=55994>  
(Изабрати ***Basic*** верзију, а касније изабрати ***Install SSMS*** и инсталирати)
- Dokumentacija: <https://docs.microsoft.com/en-us/sql/sql-server/sql-server-technical-documentation?view=sql-server-2017>

# Окружење у *MS SQL Server*-у

- Покретање инсталираног система за рад са базама:

Microsoft SQL Server Tools 17 → Microsoft SQL Server Management Studio 17



- Microsoft SQL Server Management Studio (SMSS)* је апликација унутар *MS SQL Server*-а за рад са базама у SQL језику.