

МОДЕЛОВАЊЕ

Најчешће је случај да се подаци који се у животу виде заједно на једном месту, у **релационој бази података** чувају у различитим **табелама**. Процес којим се од описа реалног пословања и података који су потребни за то пословање долази до распоређивања података у различите табеле, назива се **моделовање**. Моделовање је јако важно и подразумева детаљну анализу пословања, као и креирање **модела** на основу којег ће се креирати база података употребом конкретне система за управљање базама података.

Основе моделовања ћемо објаснити кроз конкретан пример једног пословања - **библиотеке**. У библиотеци се налази велики број књига и за сваку књигу имамо више података који је описују: назив, аутор, издавачка кућа, година издања... Од сваке књиге можемо да имамо један, али можемо да имамо и више примерака, као што може да се види на слици која следи. Сваки примерак у библиотеци је обележен инвентарским бројем. Члановима библиотеке издајемо примерке књига које траже и водимо рачуна о томе када су узели књигу, и да ли су је вратили у року. Члан библиотеке се постаје након уклањања, процеса који подразумева да се оставе лични подаци као што су: име, презиме, адреса, телефон, број личне карте... Као што може да се види, рад библиотеке подразумева рад са подацима, и то са подацима о књигама, члановима, издавањем књига. Сви ови подаци су се обрађивали и пре постанка рачунара, ручним путем и на папиру. У данашње време свака библиотека је дигитализовала своје пословање, има рачунаре и има базу података и апликацију помоћу које користи ту базу.



Први корак приликом моделовања је уочавање **ентитета**. Ентитет је нешто што је важно за пословање и о чему чувамо податке. У случају библиотеке, ентитети би били: књига, примерак, издавач, аутор, члан, позајмица...

KNJIGA

PRIMERAK

IZDAVAČ

AUTOR

POZAJMICA

ČLAN

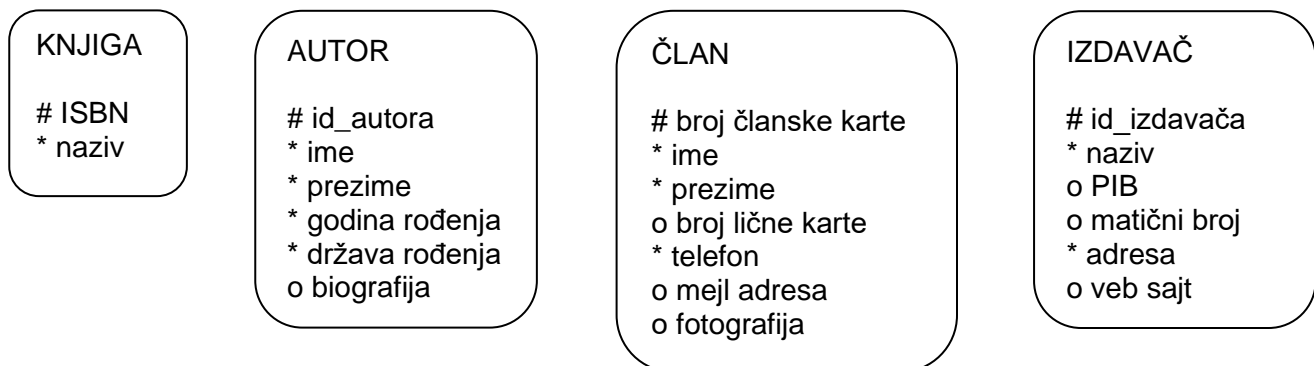
Сваки ентитет има своје **инстанце** са којима се сусрећемо у пословању. Инстанца је један конкретан примерак.

ЕНТИТЕТ	ИНСТАНЦЕ
КЊИГА	<ul style="list-style-type: none"> • Рачунарство и информатика за 3. разред гимназије • Рачунарство и информатика за 4. разред гимназије • Веб програмирање • Програмирање - класе и објекти • На западу ништа ново • Тријумфална капија
АУТОР	<ul style="list-style-type: none"> • Станка Матковић • Мијодраг Ђуришић • Марко Видојковић • Ерих Марија Ремарк
ИЗДАВАЧ	<ul style="list-style-type: none"> • СЕТ • Завод за уџбенике
ЧЛАН	<ul style="list-style-type: none"> • Милица Зорановић • Миљана Вуковић • Петар Петровић

Ентитети могу да буду јаки и слаби. Јаки ентитети су независни ентитети као што су, у примеру библиотеке: књига, члан, издавач. Инстанце слабих ентитета не постоје без присуства инстанци јаких ентитета. На пример, један примерак постоји само уколико постоји књига чији је то примерак. Други пример, позајимца је могућа само уколико постоји одређени члан који узима одређени примерак књиге.

Сваки ентитет је описан **атрибутима**. Вредности атрибута за конкретну инстанцу ентитета су подаци који се чувају у бази података.

На моделу се ентитет представља правоугаоником са заобљеним ћошковима. Назив ентитета је увек у једнини и наводи се унутар правоуганика. Атрибути се такође набрајају унутар ентитета (правоугаоника) и означавају се једним од следећих симбола: # (примарни јединствени идентификатор), * (обавезан атрибут), или o (опциони атрибут).



Вредност сваког атрибута је најчешће један краћи податак, и то у великом броју случајева текстуални или бројевни. Вредности атрибута *ime* су краћи текстуални подаци (највише до неколико десетина знакова), на пример: Ана, Milica, Petar... Вредности атрибута *godina rođenja* су бројеви, на пример: 1998, 2001, 1979... Међутим, вредност атрибута може да буде и већи податак. На пример, сваки аутор има једну биографију, а то је најчешће текстуални документ који има неколико стотина речи (више хиљада знакова). Члан библиотеке, на пример, може да остави своју фотографију. У свим примерима који следе ћемо се држати краћих и једноставнијих података (најчешће краћи текстуални податак, или

бројевни податак), али је потребно запамтити да се у базама података као подаци чувају и велики подаци као што су: текст од неколико стотина речи, фотографија, па чак и звучни запис, видео, и сл.

Сваком ентитету у релационој бази података одговара једна табела. Називи табела су увек у множини. Следи пример како би изгледала табела *knjige* са унетим подацима, која би одговарала ентитету *KNJIGA*. Једној инстанци ентитета одговара један ред у табели.

ISBN	naziv
978-86-7991-383-8	Veb programiranje
978-86-17-18654-6	Računarstvo i informatika za 3. razred gimnazije
978-86-7991-411-8	Programiranje - klase i objekti

Један од атрибута обележимо тарабицом (#). Тај атрибут је **примарни јединствени идентификатор**. Вредност овог атрибута мора да постоји за сваку инстанцу и мора да је јединствена. Дакле, свака књига мора да има свој ISBN број и две књиге не могу да имају исти ISBN број. Неколико атрибута у истом ентитету могу да имају особину да за сваку инстанцу имају јединствену вредност. Један обележавамо тарабицом, а за остале морамо у додатној документацији да наведемо да имају ову особину да би се то касније реализовало додатним правилима у бази података.

За ентитет *ČLAN* имамо следеће атрибује који испуњавају услов да су њихове вредности јединствене за сваку инстанцу:

- број чланске карте - обележен тарабицом, примарни јединствени идентификатор;
- број личне карте - два члана не могу да имају исти број личне карте.

За ентитет *IZDAVAČ* имамо следеће атрибује који испуњавају услов да су њихове вредности јединствене за сваку инстанцу:

- *id_izdavača* - обележен тарабицом, примарни јединствени идентификатор;
- *naziv* - два правна лица не могу да имају исти назив;
- *PIB* - два издавача не могу да имају исти ПИБ - порески идентификациони број, јединствени број који се додељује сваком правном лицу (фирми, компанији, предузећу);
- *matični broj* - два издавача не могу да имају исти матични број, јединствени број који се додељује сваком правном лицу.

Два члана могу да имају исто име, па чак и исто име и презиме, исту адрсу становања и тако даље. Али мора да постоји атрибут по чијој вредности ћемо их разликовати. То је управо примарни јединствени идентификатор.

Иако, на пример, два издавача не могу да имају исти назив, тај атрибут никад нећемо изабрати да буде примарни јединствени идентификатор зато што је текстуални податак. Примарни јединствени идентификатор је најћешће број зато што је бројеве у рачунарству лако упоређивати и сортирати.

Примарни јединствени идентификатор може бити **природан и вештачки**.

Природни су они који постоје и ван базе података. На пример, ISBN број се додељивао књигама и пре постанка рачунара. Такође, чланови библиотеке добијају чланске карте и бројеве чланских карата формиране на одређене начине невезано од рачунара. Ове податке можемо да употребимо у рачунарству и да их у базама података искористимо као јединствене идентификаторе. За разлику од ових атрибута, *id_autora* и *id_izdavača* су бројеви који ће постојати само у бази података.

И ситуацијама када постоје природни примарни јединствени идентификатори, можемо увек да уведемо вештачке. Тако смо могли за књиге и чланове да уведемо вештачке примарне јединствене идентификаторе: *id_knjige* и *id_člana*. У том случају би ентитети изгледали другачије, што је приказано на следећој илустрацији.

KNJIGA

id_knjige
* ISBN
* naziv

ČLAN

id_člana
* broj članske karte
* ime
* prezime
o broj lične karte
* telefon
o mejl adresa
o fotografija

Табела *knjige* са унетим подацима, која би одговарала измењеном ентитету *KNJIGA* би такође изгледала другачије. Са стране рачунарског система, рад са вештачким примарним јединственим идентификаторима је једноставнији.

id_knjige	ISBN	naziv
1	978-86-7991-383-8	Veb programiranje
2	978-86-17-18654-6	Računarstvo i informatika za 3. razred gimnazije
3	978-86-7991-411-8	Programiranje - klase i objekti

Друга подела примарних јединствених идентификатора је на **просте** и **сложене**. Сваки ентитет има увек тачно један примарни јединствени идентификатор, али он може да се састоји од једног атрибута (прост) или од комбинације неколико атрибута (сложен). Сви до сада наведени ентитети су имали просте примарне јединствене идентификаторе. Сложен можемо да видимо на примеру ентитета *POZAJMICA*. Јединствена је комбинација атрибута који означавају који члан је коју књигу позајмио ког дана. Ова ситуација се другачије црта на моделу, али ће о томе бити више речи касније.

POZAJMICA

broj članske karte
inventarski broj primerka knjige
datum pozajmice
o datum vraćanja

Осим примарног јединственог идентификатора, сви остали атрибути се обележавају са:

- звездицом (*) - обавезни;
- кружићем (o) - опциони.

За сваки атрибут морамо да одлучимо, на основу описа пословања, да ли је обавезан и одговарајући податак мора да се унесе у базу података, или није.

Неки подаци морају да се унесу. На пример, нема смисла да упишемо члана у библиотеку ако немамо његово име и презиме. Књига мора да има назив.

Опциони атрибути су или они за које може да се догоди да за неку инстанцу не постоји одговарајући податак или, податак увек постоји али за пословање није важан и може да се не унесе у базу, или може накнадно да се унесе. Следи неколико примера опционих атрибута.

Опциони атрибути и разлог зашто су опциони:

- број личне карте члана - млађи од 16 година немају личну карту;

- датум враћања за позајмицу - за примерке који се тренутно налазе изнајмљени код чланова немамо овај податак;
- мејл адреса - свако је има, али библиотеци овај податак није важан, потребан им је број телефона, а мејл адресу чланови могу да оставе само уколико желе да добијају мејлове са промоцијама;
- веб сајт издавача - можда неки издавач нема веб сајт;
- ПИБ издавача - сваки издавач има ПИБ, али то није неопходан податак за библиотеку, може и накнадно да се унесе.

Након што се уоче ентитети и наброје њихови атрибути, потребно је да се ентитети повежу. Свака **веза** спаја два разлитита ентитета или, искључиво у случају рекурзивне везе, ентитет сам са собом. Везе препознајемо у опису посвовања као глаголе: аутор је *написао* књигу, члан је *позајмио* примерак, издавачка кућа је *издала* књигу, и тако даље.

Свака веза има: назив, опционаност и кардиналност. Назив је глагол који препознајемо у опису пословања.

Опционалност нам говори да ли нека инстанца мора, или може, да буде у вези са инстанцом другог ентитета. Примери:

- књигу *мора* да напише неки аутор (обавезна веза);
- члан *може, али не мора*, да позајми неку књигу (опциона веза);
- књигу *мора* да објави неки издавач (обавезна веза).

Некад нам опционална веза служи и за то да можемо у одређеном редоследу да уносимо податке у базу. На пример, наравно да је аутор написао бар једну књигу, али ћемо ставити да аутор може, а не мора, да напише књигу (опциона веза), само зато да бисмо у базу одвојено прво унели податке о аутору, а затим податке о његовим књигама. Обавезна веза би нас условила да уз податке о аутору морамо да унесемо и податке о једној његовој књизи, а онда тај процес уноса не би био баш јасан и чист. Из овог разлога су опционе везе чешће на моделу.

Кардиналност нам говори да ли је једна инстанца једног ентитета и вези са једном или са више инстанци другог ентитета. Примери:

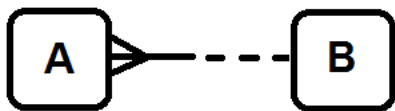
- аутор је написао више књига (више)
- одређени примерак књиге је издао тачно један издавач (један);
- један издавач је издао више књига (више).

Према кардиналности везе се деле на:

- један према један, 1:1
- један према више, 1:M
- више према више, M:M

Као ознака за више може да се користи слово М (енг. Many), или слово N (слово које користимо за обележавање скупа природних бројева) или ознака за бесконачно.

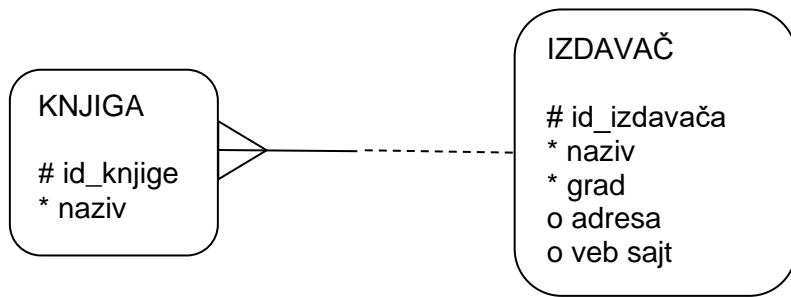
Најчешћа веза у релационим базама података је веза **један према више, 1:M**.



Ову везу могу да опишу две реченице:

1. Сваки А мора да буде у вези да тачно једним В.
2. Сваки В може, а не мора, да буде у вези са једним или више А.

Погледајмо пример издавача и књига.



Ову везу могу да опишу следеће две реченице:

1. Сваку књигу мора да објави тачно један издавач.
2. Сваки издавач може, а не мора, да објави једну или више књига.

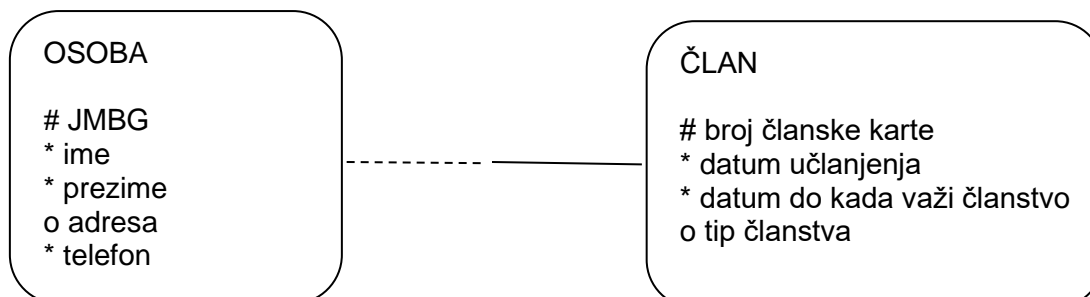
Напомена: Овај модел је исправан уколико се узме у обзир претпоставка да сви примерци исте књиге у библиотеци су од истог издавача. Уколико се у библиотеци налазе примерци исте књиге различитих издавача, ови ентитети би морали да се повежу везом М:М. **Запамтити да је јако важно анализирати конкретан пословни захтев и креирати модел који одговара управо њему!** Две библиотеке могу да имају различите облике пословања. Једна може да инсистира да се узимају мејл адресе од свих чланова, а другој то може да буде небитно, и тако даље. Према детаљном опису пословања се креира модел.

На следећој слици је приказано како би ове табеле, повезане кључевима, изгледале у бази података. Јасно се види да један издавач може да објави више књига.

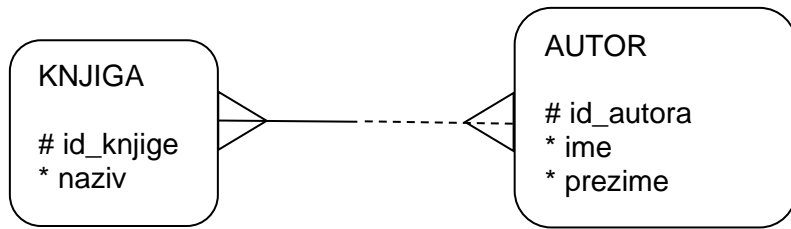
Naziv table: izdavaci				
id_izdavaca	naziv	grad	adresa	sajt
1	Zavod za udžbenike	Beograd	Obilićev venac 5	http://www.knjizara.zavod.co.rs/
2	CET	Beograd	Knez Mihailova 6	http://cet.rs/

Naziv table: knjige		
id_knjige	Naziv	id_izdavaca
111	Računarstvo i informatika za 4. razred gimnazije	1
222	Računarstvo i informatika za 3. razred gimnazije	1
321	Osnovi programiranja, programski jezik c#	1
555	PROGRAMIRANJE - klase i objekti	2
323	Veb programiranje	2

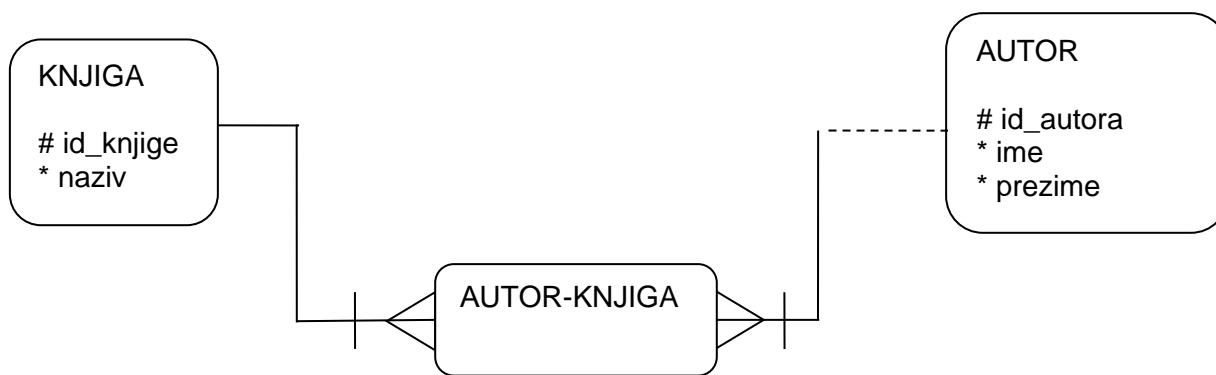
Веза 1:1 је јако ретка. Следи један пример у којем издвајамо податке о личности члана библиотеке у један ентитет, а податке о његовом чланству у други (особа и улога).



Веза више према више се јако често среће у животу и одговара многим ситуацијама у оквиру разних пословања. Ако као књиге посматрамо школске уџбенике, приметимо да свака књига има више аутора, а аутори најчешће током свог рада напишу више књига.



Овакву везу је немогуће реализовати у релационој бази података, па се на моделу уклања и уместо ње се поставља нови ентитет. Уколико нема бољег решења, назив новог ентитета може да буде комбинација назива почетна два ентитета повезана везом М:М.



Кратке цртице којима су везе прецртане имају значење да везе учествују у примарном јединственом идентификатору новог ентитета. Тако је примарни јединствени идентификатор новог ентитета који је дошао на место везе типа М:М, ентитета *AUTOR-KNJIGA*, заправо сложен и представља комбинацију примарних јединствених идентификатора ентитета *AUTOR* и *KNJIGA*: (*id_knjige*, *id_atora*).

На следећој слици је приказано како би ове табеле, повезане кључевима, изгледале у бази података.

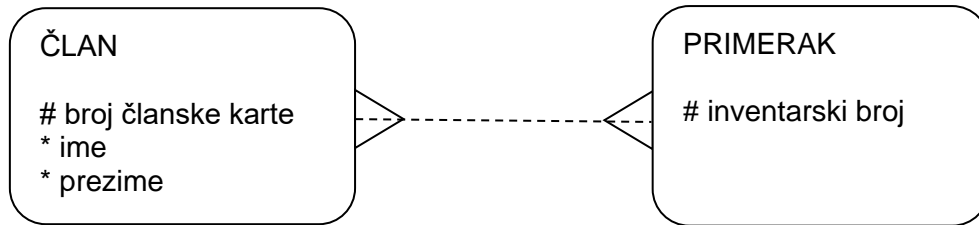
Naziv tabele: autori_knjige	
id_atora	id_knjige
101	111
101	222
101	321
102	321
103	321
102	111
102	222
103	111
103	555
103	323
101	323
102	323

Naziv tabele: knjige		
id_knjige	Naziv	id_izdavaca
111	Računarstvo i informatika za 4. razred gimnazije	1
222	Računarstvo i informatika za 3. razred gimnazije	1
321	Osnovi programiranja, programski jezik c#	1
555	PROGRAMIRANJE - klase i objekti	2
323	Veb programiranje	2

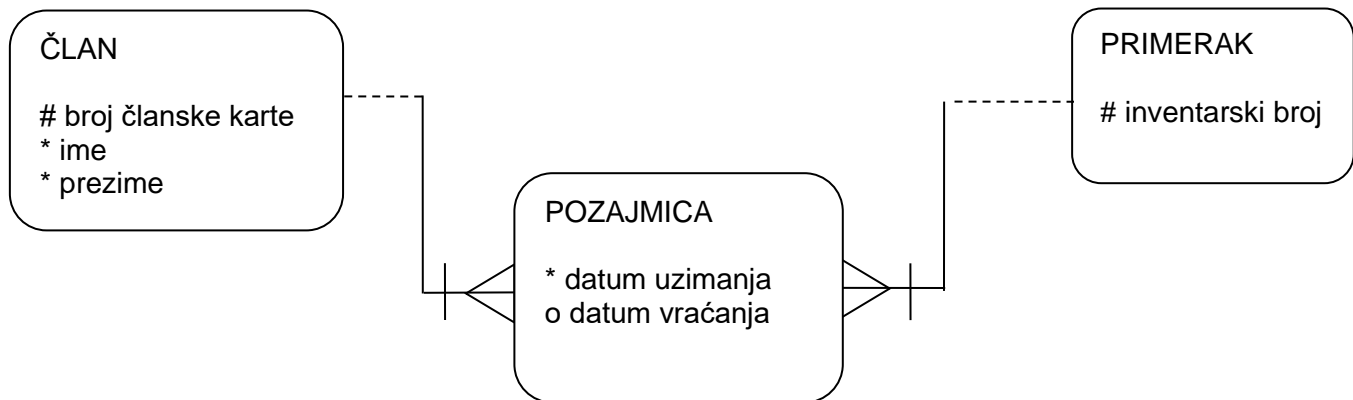
Naziv tabele: autori		
id_atora	ime	prezime
101	Stanka	Matković
102	Mijodrag	Đurišić
103	Duša	Vuković

Напомена: Овај модел је исправан уколико се узме у обзир претпоставка да библиотека чува уџбенике. Уколико се у библиотеци налазе само романи, онда је веза ентитета *AUTOR* и *KNJIGA* могла да буде 1:M (роман има једног аутора).

Веза више према више често има своје атрибуте који је описују. Погледајмо пример са позајмљивањем књига.

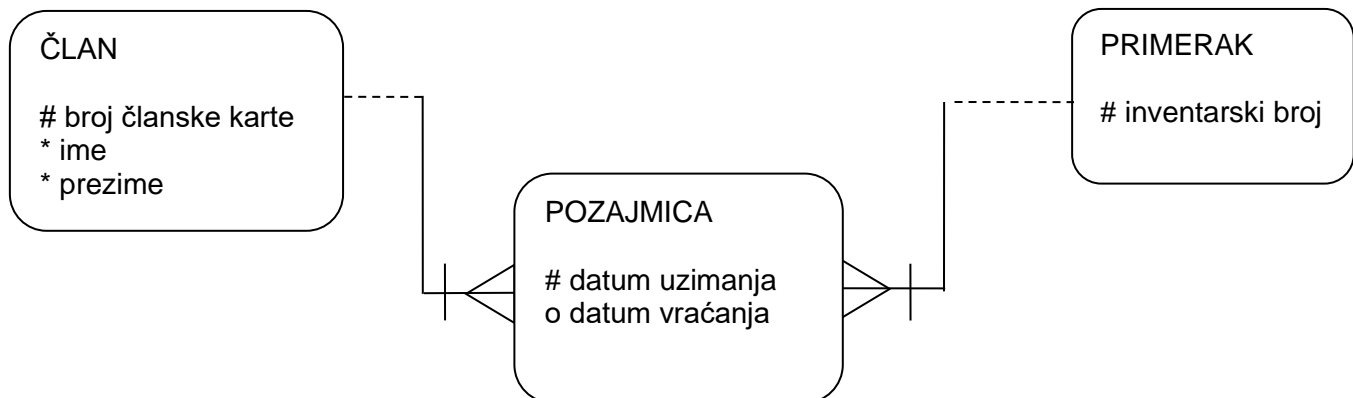


Овакву везу је немогуће реализовати у релационој бази података, па се на моделу уклања и уместо ње се поставља нови ентитет, али у овом случају тај ентитет има своје атрибуте који описују позајмицу.



Кратке цртице којима су везе прецртане имају значење да везе учествују у примарном јединственом идентификатору новог ентитета. Тако је примарни јединствени идентификатор новог ентитета који је дошао на место везе типа М:М, ентитета *POZAJMICA*, заправо сложен и представља комбинацију примарних јединствених идентификатора ентитета *ČLAN* и *PRIMERAK*: (*broj članske karte, inventarski broj*).

Комбинација ове две вредности мора да буде јединствена, тако да није могуће да се понови да исти члан позајми исти примерак више пута. Уколико то желимо да дозволимо, онда можемо датум узимања књиге да додамо да буде треће поље сложеног примарног идентификатора. У следећем примеру је примарни јединствени идентификатор ентитета *POZAJMICA* комбинација: (*broj članske karte, inventarski broj, datum uzimanja*).



Сва правила пословања која смо до сада видели могу да се нацртају на моделу. Таква правила се зову **структурна правила**. Нека правила није могуће нацтати. Њих документујемо да би касније била обрађена програмерски. Правила која не могу да се нацртају на моделу већ морају да се документују се називају **процедурална правила**.

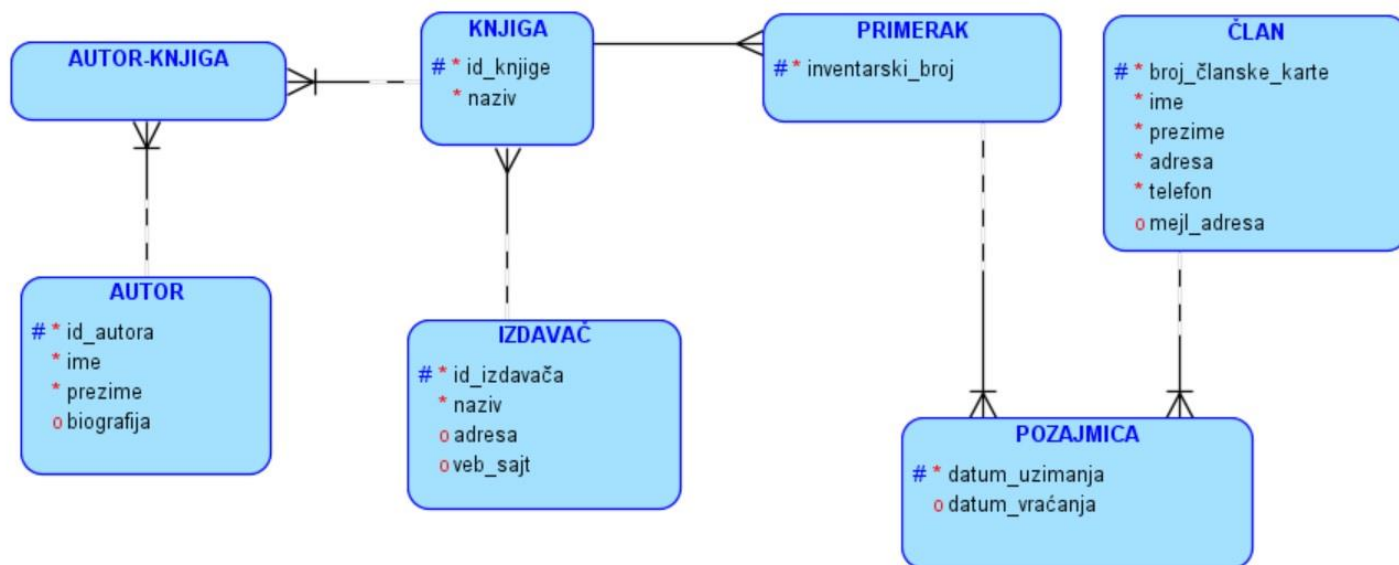
На пример, можемо да нацртамо да један члан библиотеке може да позајми више књига и то је структурно правило. Оно што не можемо да нацртамо већ морамо да документујемо да би касније било обрађено програмерски, је правило да у једном тренутку члан библиотеке не може да позајми више од 3 примерка разних књига, правило које свака библиотека има.

Модел се назива још и **дијаграм ентитета и веза**, ЕР дијаграм, или ЕРД (енг. Entity Relationship Diagram). У литератури се користи још и назив: **модел објекти везе**.

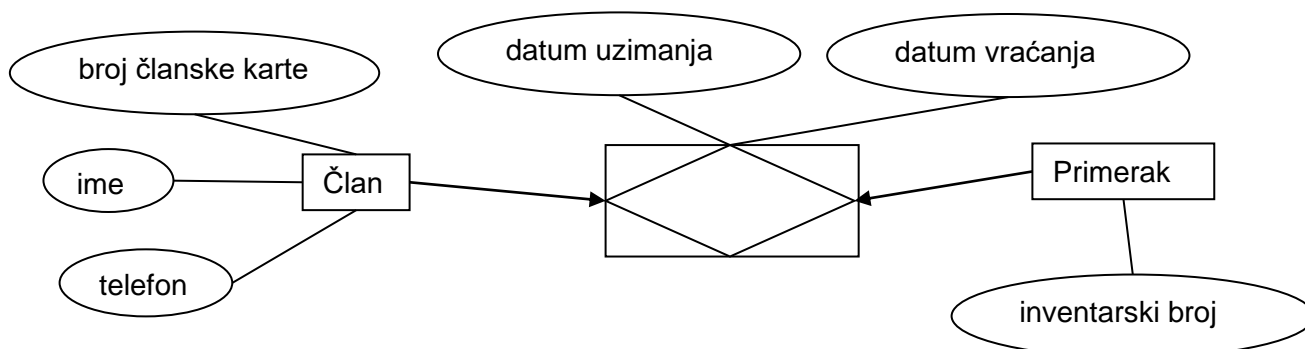
За цртање дијаграма може да се користи алат **Oracle SQL Developer Data Modeler**, који за потребе учења, а након прихватања услова употребе, може бесплатно да се преузме са следеће адресе:

<https://www.oracle.com/fr/database/technologies/appdev/datamodeler.html>

Модел библиотеке је приказан на следећој слици.



Постоје и други начини цртања модела. На следећој слици се виде исте ситуације које смо већ размотрили, али нацртане на други начин.



За који год да се начин цртања модела одлучимо, важно је да се придржавамо доследно изабраних правила цртања, а кораци приликом моделовања су увек исти:

- 1. уочавање ентитета,**
- 2. набрајање атрибута за сваки ентитет,**
- 3. повезивање ентитета везама.**

Правила која су важна приликом моделовања:

- на моделу морају да се налазе побројани сви подаци потребни за посао,
- сваки податак мора да се појави тачно једном (нема понављања истог податка на више места),
- подаци су распоређени по целинама (ентитима) и сваки се налази тачно тамо где га очекујемо.