

ACCESS PRIMER  
**BIBLIOTEKA 2008**

**Zadata su sledeća polja u bazi podataka:**

NASLOV (TXT) (REQUIRED) (naslov knjige iz naše biblioteke)

IME (TXT) VELIČINA = 3, (ime pisca)

NOBEL (Y/N) (da li je pisac dobio Nobelovu nagradu)

BR\_STRANA (NUM) (broj strana knjige)

ZANR (TXT) VELIČINA = 3 (naziv žanra kome pripada knjiga)

RODZEN (NUM) DEFAULT = 0 (godina rođenja pisca)

UMRO (NUM) CAPTION = "GODINA SMRTI" (godina smrti pisca)

**Podaci su:**

NASLOV KNJIGE	IME PISCA	NOBEL	BROJ STRANA	NAZIV ŽANRA	RODZEN	UMRO
Prokleta avlija	Ivo Andrić	YES	150	IST	1892	1975
Isijavanje	Stiven King	NO	250	HOR	1947	1947
Mračna kula	Stiven King	NO	450	FAN	1947	1947
Knjige krvi	Klajv Barker	NO	600	HOR	1960	1960
Utkani svet	Klajv Barker	NO	500	FAN	1960	1960
Čekajući Godoa	Semjuel Beket	YES	90	DRA	1906	1989

1. Izvršiti normalizaciju (dve tabele).
2. Kreirati i povezati tabele.
3. Uspostaviti pravilo da broj strana knjige mora biti veći od nule. Ukoliko se pravilo prekrši, ispisati poruku "Unesite ispravan broj strana".
4. Uspostaviti pravilo da godina rođenja mora biti manja ili jednaka godini smrti pisca (**ukoliko su godine rođenja i smrti jednake, znamo da je pisac još živ**). Ukoliko se prekrši ovo pravilo, ispisuje se poruka: "Neispravna godina smrti".
5. Popuniti tabele podacima.
6. Kreirati jednostavan upit u kome će biti prikazani naslovi knjiga po abecednom redu, imena pisaca i da li su dobili Nobelovu nagradu, ali samo za one pisce koji jesu.
7. Kreirati jednostavan upit u kome će biti prikazana imena pisaca koji su još uvek živi.
8. Kreirati upit u kome se vide imena pisaca, godine rođenja i smrti i novo polje ZIVOT koje se dobija kao razlika godina smrti i rođenja, s tim što se prikazuju samo pisci koji nisu živi (polje ZIVOT veće od nule).
9. Kreirati upit za tabelu sa podacima o piscima i knjigama kojim se izdvajaju knjige horor žanra koje su pisali pisci rođeni posle 1940 i čije ime počinje na slovo "S".

ACCESS PRIMER  
**BIBLIOTEKA 2008**  
**- UPUTSTVO -**

## 1. NORMALIZACIJA

Prvi zadatak je razdvojiti podatke na tabele. Jedan od načina na koji posmatramo podatke je da uočimo one koji se ponavljaju:

NASLOV KNJIGE	IME PISCA	NOBEL	BROJ STRANA	NAZIV ŽANRA	ROĐEN	UMRO
Prokleta avlija	Ivo Andrić	YES	150	IST	1892	1975
Isijavanje	Stiven King	NO	250	HOR	1947	1947
Mračna kula	Stiven King	NO	450	FAN	1947	1947
Knjige krvi	Klajv Barker	NO	600	HOR	1960	1960
Utkani svet	Klajv Barker	NO	500	FAN	1960	1960
Čekajući Godoa	Semjuel Beket	YES	90	DRA	1906	1989

Za svaku knjigu koju je napisao Stiven King moramo da ponovimo sve podatke koji su vezani za njega. Isto to važi i za Klajva Barkera. Očigledno je da ćemo izdvojiti podatke o piscima u zasebnu tabelu i kasnije ćemo pozivati te podatke preko njihovog ključa.

Drugim rečima, pokušaćemo da bazu dovedemo u formu sličnu ovoj:

NASLOV KNJIGE	STRANI KLJUČ	BROJ STRANA	NAZIV ŽANRA
Prokleta avlija	<b>a</b>	150	IST
Isijavanje	<b>b</b>	250	HOR
Mračna kula	<b>b</b>	450	FAN
Knjige krvi	<b>c</b>	600	HOR
Utkani svet	<b>c</b>	500	FAN
Čekajući Godoa	<b>d</b>	90	DRA

PRIMARNI KLJUČ	IME PISCA	NOBEL	ROĐEN	UMRO
<b>a</b>	Ivo Andrić	YES	1892	1975
<b>b</b>	Stiven King	NO	1947	1947
<b>c</b>	Klajv Barker	NO	1960	1960
<b>d</b>	Semjuel Beket	YES	1906	1989

Što se tiče samih polja koja su nam data, odmah možemo da ih rasporedimo u ove dve tabele, na sledeći način:

### KNJIGE

NASLOV (txt) required = yes
BR_STRANA (num)
ZANR (txt) size = 3

### PISCI

IME (txt) size = 50
NOBEL (y / n)
RODJEN (num) default = 0
UMRO (num) caption = "Godina smrti"

Sledeći korak je određivanje primarnog ključa. Pošto ne možemo sa sigurnošću da tvrdimo da se nikada neće pojaviti dve knjige istog naslova ili dva pisca istog imena, moraćemo sami da odredimo dva nova polja koja će biti primarni ključevi za tabele.

KNJIGE	PISCI
<b>SIFRA (autonumber)</b>	<b>ID (txt) size = 1</b>
NASLOV (txt) required = yes	IME (txt) size = 50
BR_STRANA (num)	NOBEL (y / n)
ZANR (txt) size = 3	RODJEN (num) default = 0
	UMRO (num) caption = "Godina smrti"

U ovom slučaju smo se odlučili da primarni ključ za tabelu KNJIGE bude autonumber (Access će sam unositi vrednosti u tabeli), dok će primarni ključ u tabeli PISCI biti tekstualnog tipa, veličine 1 (*znači samo jedan znak, odnosno slovo - u praksi bi ovako nešto bilo teško izvodljivo zbog mogućeg velikog broja pisaca, ali za ovaj naš primer je prihvatljivo*).

Konačno, moramo nekako da povežemo ove dve tabele. U najvećem broju slučajeva koristimo veze tipa **jedan-prema-više**. Ova veza podrazumeva dodavanje novog polja (stranog ključa) u jednoj od tabela koje će se povezivati sa primarnim ključem druge tabele.

Veza **jedan-prema-jedan** bi podrazumevala da jednom redu prve tabele odgovara tačno jedan red druge tabele, pri čemu bi se povezivali primarni ključevi. Pošto očigledno nije slučaj da imamo tačno jednu knjigu od jednog pisca, ovde će se koristiti veza jedan-prema-više.

Naša situacija je da imamo VIŠE knjiga koje je pisao JEDAN pisac. Po pravilu, tabela koja sadrži **strani ključ** može imati VIŠE podataka koji se odnose na JEDAN podatak iz druge tabele. To znači da će tabela KNJIGE imati strani ključ koji se povezuje sa primarnim ključem iz tabele PISCI.

Strani ključ mora biti istog tipa kao primarni ključ sa kojim se povezuje. Tako dodajemo polje PISAC u tabelu KNJIGE, koje je istog tipa kao polje ID iz tabele PISCI.

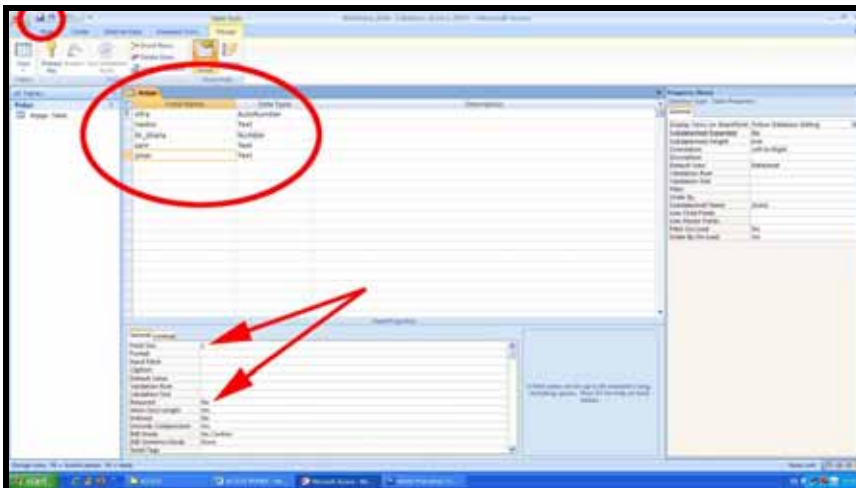
KNJIGE	PISCI
<b>SIFRA (autonumber)</b>	<b>ID (txt) size = 1</b>
NASLOV (txt) required = yes	IME (txt) size = 50
BR_STRANA (num)	NOBEL (y / n)
ZANR (txt) size = 3	RODJEN (num) default = 0
<b>PISAC (txt) size = 1</b>	UMRO (num) caption = "Godina smrti"

## 2. KREIRANJE I POVEZIVANJE TABELA

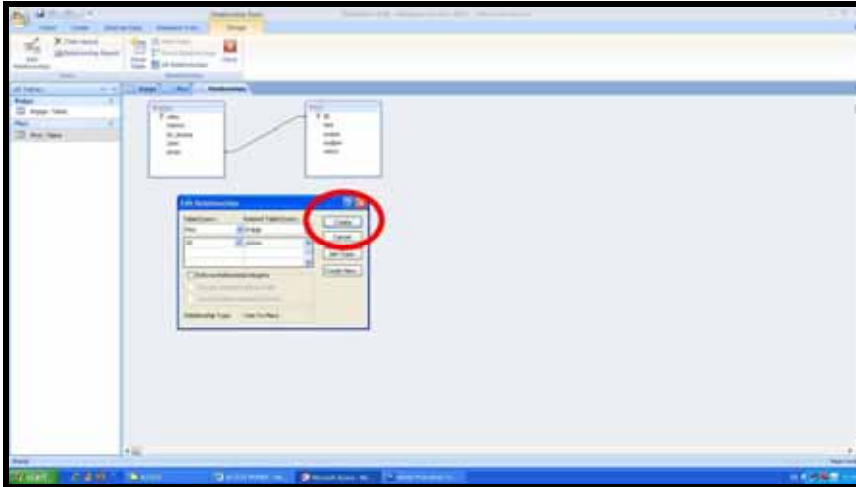
Ulazimo u program ACCESS. U njemu kreiramo novu bazu podataka (*Blank Database*) i unesemo ime za bazu (npr. *Biblioteka 2008*). Da bi kreirali bazu, kliknemo na dugmence *Create*.



Access nam odmah otvara novu praznu tabelu. Da bi kreirali njenu strukturu (uneli polja i tipove), ulazimo u *Design View*. Access će tražiti da odmah unesemo ime tabele (*Knjige*). Tablicu kreiramo prema normalizaciji koju smo izvršili. Ne treba da zaboravimo ni posebna svojstva u tabeli koja se određuju za svako polje. Kada završimo sa jednom tabelom, snimamo je i prelazimo na drugu (*Create – Table*).



Povezivanje tabela se vrši preko komande *Relationships* koja se nalazi u *Database Tools*. Da bi povezali ove dve tabele, potrebno je da ih obe „dodamo“ u prostor za povezivanje, a onda mišem prevučemo polje PISAC na polje ID (ili obrnuto). Prikazanu relaciju u prozoru, potvrdimo sa *Create*. Kada kreiramo vezu, potrebno je ponovo snimiti sa *Save*.

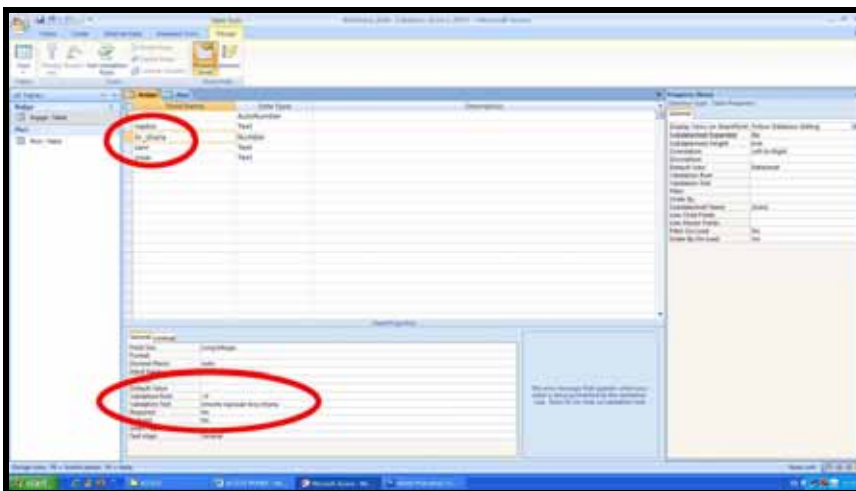


### 3. PRAVILO PROVERE ZA POLJE (VALIDATION RULE – FIELD)

Za polje BR\_STRANA iz tabele KNJIGE postavljamo VALIDATION RULE, koji se nalazi u tablici FIELD PROPERTIES. Na ovom mestu se unose kriterijumi za proveru ispravnosti podataka koji se odnose samo na pojedinačna polja.

U ćeliji VALIDATION RULE se unosi kriterijum  $>0$ , a u ćeliji VALIDATION TEXT, se unosi tekst koji se ispisuje ako se ovo pravilo prekrši.

Svaku izmenu u strukturi baze moramo da snimimo sa *Save*.



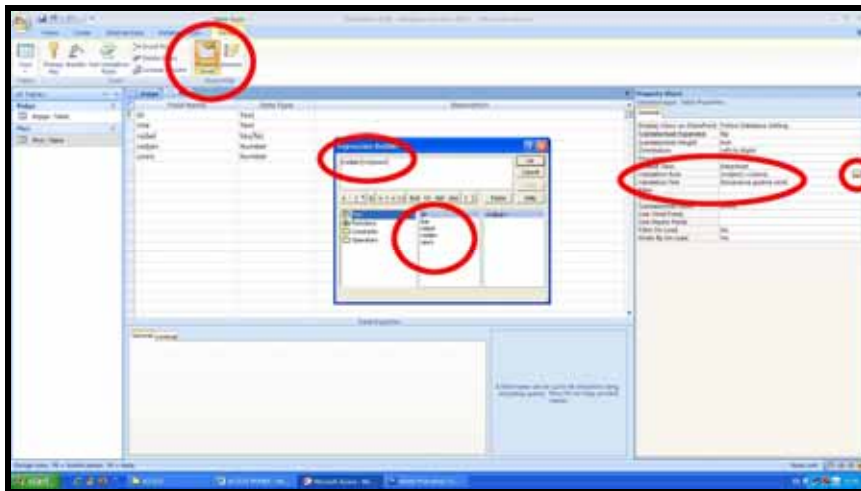
### 4. PRAVILO PROVERE ZA CEO RED (RECORD VALIDATION)

U tabeli PISCI pozivamo komandu *Property Sheet*. Prikazuje se prozor sa svojstvima za celu tabelu. U ovoj tablici se takođe nalaze VALIDATION RULE i VALIDATION TEXT.

Pravila koja ovde postavljamo se proveravaju kada se unese ceo red tabele. Jedan od primera kada nam je ovo pravilo neophodno je kada treba uporediti vrednost jednog polja sa drugim (godina rođenja i godina smrti).

U samoj ćeliji za VALIDATION RULE se nalazi i dugmence označeno sa tri tačkice, kojim pozivamo *Expression Builder* u kome definišemo izraz **[rodjen]<=[umro]**.

Svaku izmenu u strukturi baze moramo da snimimo sa *Save*.



## 5. UNOS PODATAKA

Unos podataka se obavlja kada za odgovarajuću tabelu pozovemo *Datasheet View*. Podaci koji se unose su definisani u delu koji se bavio normalizacijom.



## 6. - 9. UPITI

Da bi kreirali upit, pozivamo komandu *Create – Query Design*. Prvo što se traži od nas je da odaberemo tabele (i eventualno upite) na osnovu kojih kreiramo novi upit.

Prvi upit se zasniva na poljima iz tabela KNJIGE i PISCI, što znači da obe treba da dodamo u prostor za tabele.



Slede rešenja za upite.

## 6. KNJIGE NOBELOVACA

Field:	naslov	ime	nobel	
Table:	Knjige	Pisci	Pisci	
Sort:	Ascending			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Criteria:			Yes	
or:				

## 7. ŽIVI PISCI

(za kriterijum polja RODJEN se poziva *Expression Builder* da bi se uporedilo sa poljem UMRO)

Field:	ime	rodjen		
Table:	Pisci	Pisci		
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteria:		[Pisci]![umro]		
or:				

## 8. ŽIVOT PISACA

Field:	ime	rodjen	umro	Zivot: [umro]-[rodjen]	
Table:	Pisci	Pisci	Pisci		
Sort:					
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Criteria:				>0	
or:					

## 9. KNJIGE PISACA

Field:	naslov	ime	zanr	rodjen	
Table:	Knjige	Pisci	Knjige	Pisci	
Sort:					
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Criteria:		Like "S*"	"HOR"	>1940	
or:					