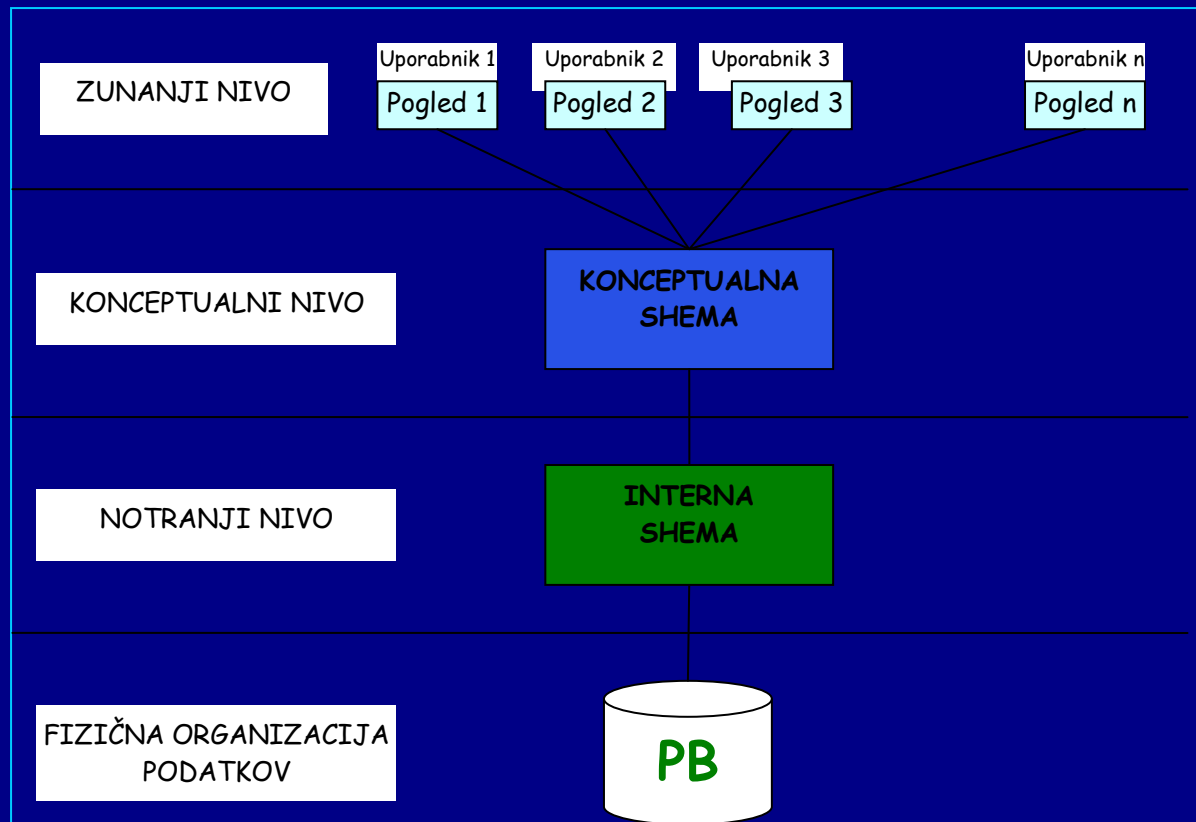


Arhitektura PB in podatkovna neodvisnost

Trinivojska arhitektura PB

- Drugo ime je ANSI/SPARC arhitektura
- 1975. jo je razvil Standards Planning and Requirements Committee (SPARC) American National Standards Institute (ANSI)
- Opredeljuje 3 nivoje opisov PB:
 - zunanji nivo (opredeljuje različne poglede uporabnikov na podatke)
 - konceptualni nivo (celovit pregled vseh vrst podatkov in povezav med podatki v PB)
 - fizični nivo (opredeljuje fizični način shranjevanja in dostope do podatkov)
- Zelo poenostavljeno: zunanji nivo opisuje uporabo podatkov, konceptualni pomen podatkov, notranji pa način shranjevanja podatkov.
- Opis posameznega nivoja se imenuje shema

Trinivojska ali ANSI/SPARC arhitektura PB (nad.)



Zunanji nivo

- različni pogledi uporabnikov na PB
- Vsak pogled = ena zunanja shema
- Ena shema opisuje podatke, zanimive za posameznega uporabnika
- Pogledi – prilagojeni potrebam uporabnika
- Za različne uporabnike obstajajo različne sheme za prikaz istih podatkov
- Zunanji pogledi (pogosto) prikazujejo izračunane podatke
- Pomagajo pri zagotavljanju varnosti // uporabnik vidi le omejeno množico podatkov
- Število zunanjih shem je (teoretično) neomejeno

Konceptualni nivo

- Vsebuje celovito informacijo o vsebini in strukturi podatkovne baze
- Opisani so podatki in povezave med podatki v PB
- Konceptualna shema vsebuje le logične podatke o PB
- Vsaka PB ima le 1 konceptualno shemo
- Konceptualna shema:
 - odraža potrebe uporabnikov
 - popolnoma je neodvisna od SUPB-ja in HW
- Vsaka izmed zunanjih shem sloni na enem delu konceptualne sheme
 - ➔ vsi podatki, ki jih prikažemo na zunanjem nivoju morajo biti zajeti ali izpeljani iz podatkov, opisanih na konceptualnem nivoju.

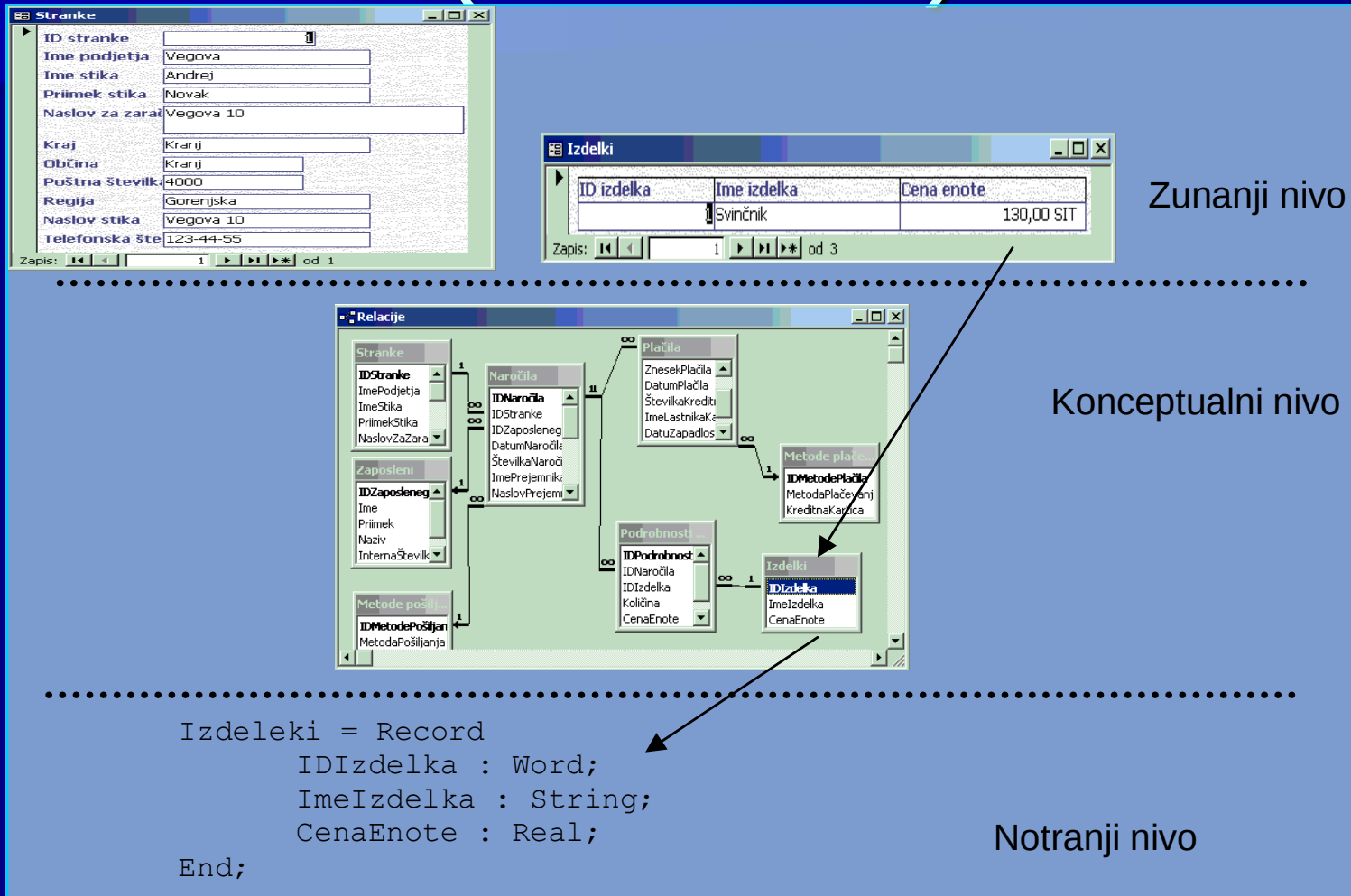
Notranji nivo

- fizična predstavitev PB v računalniku
 - opisuje KAKO so podatki shranjeni v PB: opisuje:
 - podatkovne strukture
 - datotečne organizacije,
 - Indekse, ...
 - Kakovost notranje sheme določa učinkovitost PB in optimalnost izrabe prostora na disku!
 - Notranja shema je odvisna od SUPB-ja!
-
- Pod notranjim nivojem obstaja še fizični nivo, s katerim upravlja OS na osnovi navodil, ki jih dobi od SUPB-ja. Ločitev funkcij med SUPB-jem in OS na fizičnem nivoju ni striktno določena in se med sistemi zelo razlikuje.

Preslikave med nivoji / shemami

- Izvaja SUPB
- SUPB preverja tudi skladnost med nivoji
- Samodejno izvajanje preslikav s strani SUPB-ja zagotavlja neodvisnost med posameznimi nivoji opisov PB!

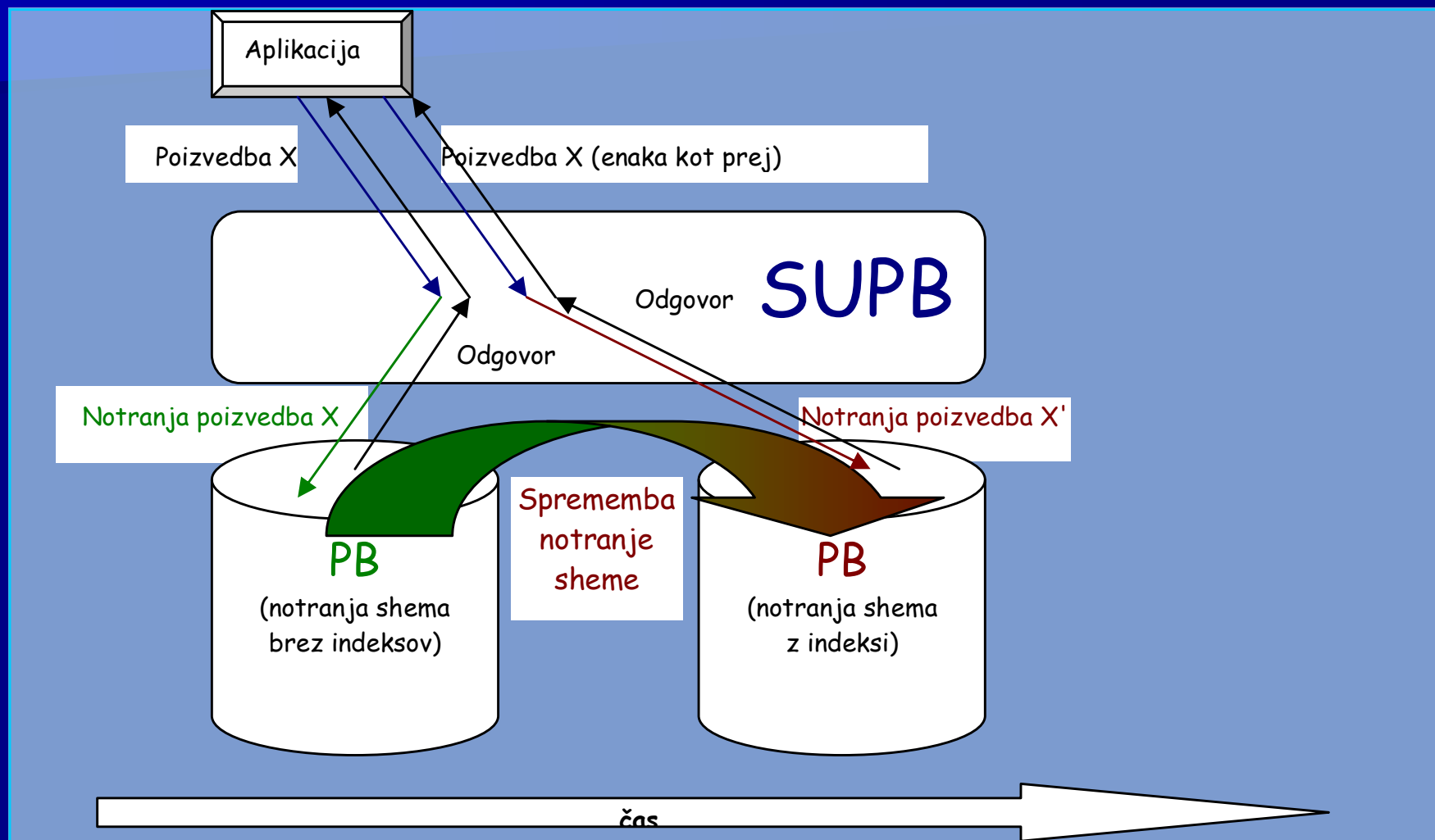
Trinivojska arhitektura PB v praksi (MS Access)



Podatkovna neodvisnost

- Podatek //sam po sebi// = neodvisen resurs (*načeloma neodvisen od programov / aplikacij, ki ga uporabljajo*)
- Ena od glavnih pridobitev uporabe podatkovnih baz in SUPB-jev = neodvisnost aplikacij od podatkov
- Podatkovna neodvisnost (na splošno) pomeni imunost višjega nivoja na spremembe, ki se zgodijo na nižjem nivoju opisov PB.
- 2 vrsti podatkovne neodvisnosti:
 - fizična in
 - logična podatkovna neodvisnost.

Podatkovna neodvisnost (slika)



Fizična podatkovna neodvisnost

- neodvisnost konceptualne sheme od sprememb v notranji shemi
- Programi ne smejo biti odvisni od načinov shranjevanja podatkov in obratno: strukturo podatkovne datoteke ne sme narekovati aplikacijski program.
- zmanjšuje potrebo po vzdrževanju:
 - programov in
 - konceptualnega modela PB

Logična podatkovna neodvisnost

- neodvisnost zunanjih shem od sprememb v konceptualni shemi
- omogoča spreminjanje konceptualne sheme, ne da bi pri tem morali nujno spremeniti tudi zunanje sheme
//predvsem možnost dodajanja novih podatkov v tabele
- Uporaba:
 - Logična podatkovna neodvisnost zagotavlja starim aplikacijskim programom neodvisnost od novo-dodanih podatkov, ki jih ti programi ne potrebujejo.
 - Logično podatkovno neodvisnost lahko uporabimo tudi pri integraciji prej razdeljenih / različnih podatkovnih baz.