

## Poglavje 6

# Porazdeljene podatkovne baze

- Razlogi za pojav porazdeljenih PB
- Osnovne definicije
- Razlika med paralelnimi in porazdeljenimi PB
- Arhitekture porazdeljenih PB
- Prednosti porazdeljenih PB
- Dodatne funkcije porazdeljenih PB

## Porazdeljene podatkovne baze



### Kaj si bomo pogledali?

- Razlogi za pojav porazdeljenih PB
- Osnovne definicije
- Razlika med paralelnimi in porazdeljenimi PB
- Arhitekture porazdeljenih PB
- Prednosti porazdeljenih PB
- Dodatne funkcije porazdeljenih PB

## Pojav porazdeljenih podatkovnih baz

- Porazdeljene podatkovne baze se pojavijo kot odziv na porazdeljeno procesiranje in računanje, ki je zaznamovalo področje operacijskih sistemov.
- V raziskovalni sferi veliko truda vložijo v porazdeljenost podatkov. Nekatera področja so zelo težavna:
  - Porazdeljeno poizvedovanje
  - Obvladovanje transakcij
  - Varnost podatkov
  - Upravljanje z meta-podatki

## Definicije

- Porazdeljena podatkovna baza (PPB) predstavlja nabor več logično povezanih podatkovnih baz, fizično razpršenih (porazdeljenih) po računalnikih, povezanih z računalniškim omrežjem.
- Sistem za upravljanje s PPB (SUPPB) predstavlja programsko opremo, ki omogoča upravljanje s PPB na način, ki je uporabniku transparenten.

## Lastnosti porazdeljenih sistemov

### ▪ Lastnosti SUPPB:

- zajema eno samo logično podatkovno bazo, ki je razdeljena na fragmente.
- Fragmenti so shranjeni na enem ali več računalnikih pod nadzorom ločenih SUPB-jev.
- Fragmenti so lahko replicirani.
- Računalniki so povezani s komunikacijskim omrežjem.
- SUPB na povezanih računalnikih samostojno obdelujejo lokalne podatke.
- Vsak SUPB na povezanih računalnikih sodeluje v vsaj eni globalni aplikaciji.
- Porazdeljenost PB je za uporabnike transparentna.

## PPB in porazdeljeno procesiranje

- Porazdeljene podatkovne baze so fizično porazdeljene po več računalnikih.
- Če so podatki centralizirani, gre zgolj za porazdeljeno procesiranje, četudi do podatkov dostopa lahko več uporabnikov.

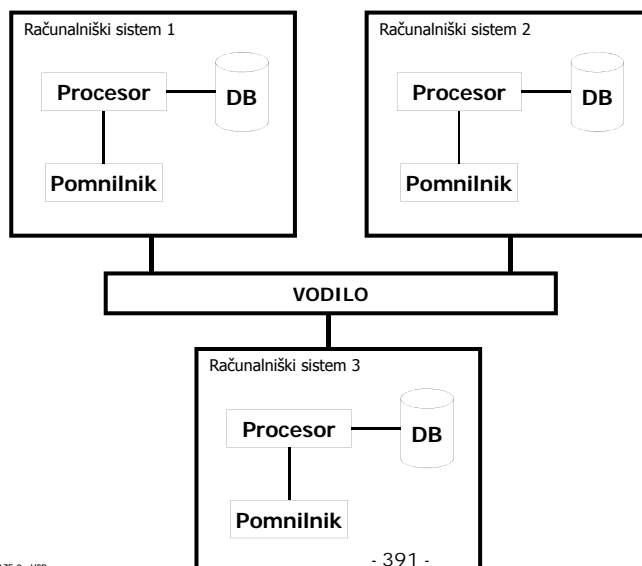
## Paralelne in porazdeljene PB...

- Paralelne in porazdeljene PB se razlikujejo.
- Paralelne PB označujejo PB, ki tečejo na več enakih sistemih (so narejene tako, da tečejo na več procesorjih in diskih) z namenom, da lahko izvajajo več operacij vzporedno. Cilj je povečati učinkovitost.
- Porazdeljene PB so navadno heterogene - porazdeljene po sistemih z različno strojno in programsko opremo.

## Paralelne in porazdeljene PB

- Tipične arhitekture za paralelne PB:
  - Deljen pomnilnik (tesno povezana arhitektura): več procesorjev si deli skupen sekundarni in primarni pomnilnik.
  - Deljeno diskovje (rahlo povezana arhitektura): več procesorjev si deli skupen sekundarni pomnilnik; vsak ima svoj primarni pomnilnik.
  - Brez deljenih sredstev: vsi procesorji imajo svoj primarni in sekundarni pomnilnik.
- Katera arhitektura je najbližja porazdeljenim PB?

## Paralelna PB Shema arhitekture brez deljenih sredstev

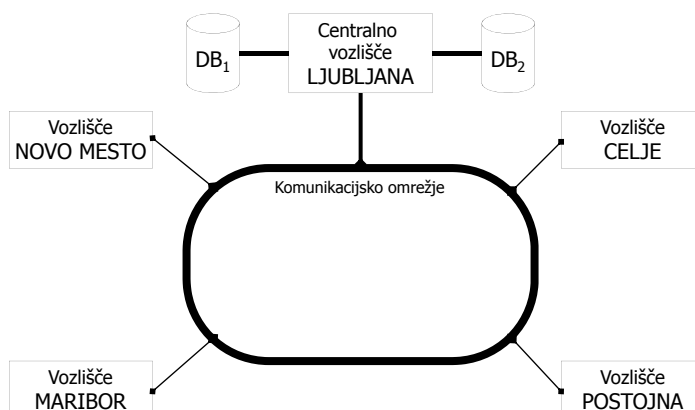


PODATKOVNE BAZE 2 - VSP

- 391 -

Fakulteta za računalništvo in informatiko  
Univerza v Ljubljani

## Centralizirana arhitektura Porazdeljeno procesiranje



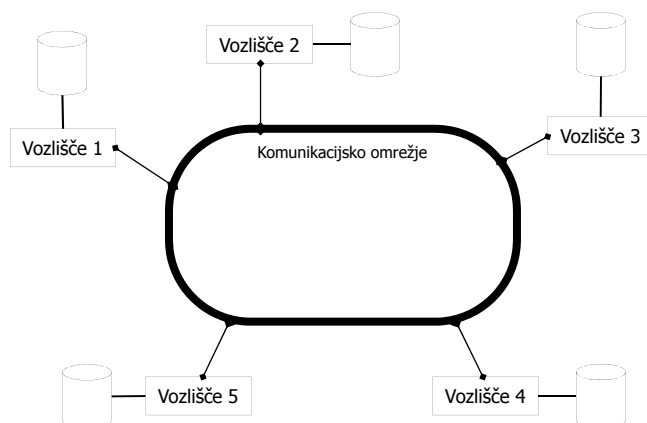
PODATKOVNE BAZE 2 - VSP

- 392 -

Fakulteta za računalništvo in informatiko  
Univerza v Ljubljani

## Porazdeljena arhitektura

Porazdeljena PB

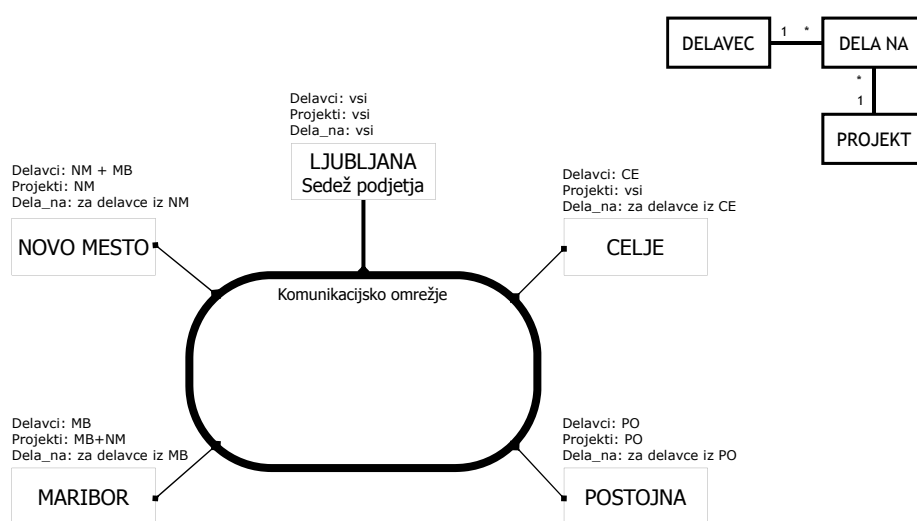


PODATKOVNE BAZE 2 - VSP

- 393 -

Fakulteta za računalništvo in informatiko  
Univerza v Ljubljani

## Podatkovna distribucija in replikacija



PODATKOVNE BAZE 2 - VSP

- 394 -

Fakulteta za računalništvo in informatiko  
Univerza v Ljubljani

## Prednosti in slabosti PPB...

### ▪ Prednosti PPB:

- Lahko izražajo porazdeljeno organizacijsko strukturo
- Izboljšajo porazdeljenost (dostop do vseh podatkov) in avtonomnost (podatki tam, kjer se uporabljajo)
- Izboljšajo razpoložljivost (sistem je porazdeljen. Če odpove eno vozlišče, so druga še vedno dostopna)
- Izboljšajo zanesljivost (replikacija)
- Izboljšajo učinkovitost (podatki porazdeljeni glede na poizvedovanje, paralelnost,...)

## Prednosti in slabosti PPB...

### ▪ Prednosti PPB (nadaljevanje):

- Ekonomičnost
  - 1965 - Grosch's Law (računska moč = cena<sup>2</sup>) npr. za trikrat večjo ceno dobimo devet krat večjo računsko moč
  - Danes: ceneje sestaviti sistem iz manjših računalnikov kot kupiti enega večjega! Prenos podatkov po omrežju (od centralnega računalnika) tudi stane!
- Modularna gradnja (enostavna razširljivost)
- Integracija (osnovna motivacija razvoja SUPB je integracija, ne centralizacija)
- Konkurenčnost (številne nove IT zahtevajo porazdeljenost, npr. e-business, skupinsko delo, obvladovanje procesov).

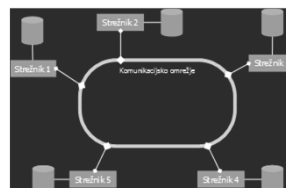
## Prednosti in slabosti PPB

### ▪ Slabosti PPB:

- Kompleksnost (transparentnost porazdelitve, replikacija,...)
- Cena (strošek nabave in vzdrževanja, strošek komunikacije)
- Varnost (podatki na več mestih; varnost komunikacijskega omrežja)
- Težje zagotavljati podatkovno skladnost
- Pomanjkanje standardov
- Pomanjkanje izkušenj
- Kompleksno načrtovanje

## SUPPB iz vidika strojne opreme

- Z vidika strojne opreme se porazdeljen SUPB od centraliziranega loči predvsem v naslednjih elementih:
  - Obstaja več računalnikov, imenovanih mesta ali vozlišča
  - Mesta oziroma vozlišča so povezana z računalniškim omrežjem, ki omogoča prenos podatkov in ukazov.
- Mesta ali vozlišča so lahko v neposredni bližini, lahko pa jih ločijo zelo velike razdalje.



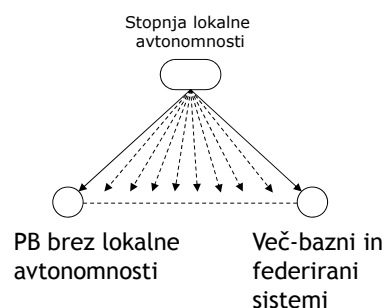
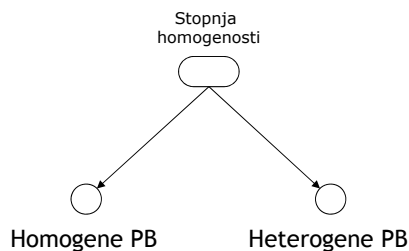


## Vpliv topologije omrežja

- Topologija omrežja ima lahko velik vpliv na učinkovitost in tudi realizacijo porazdeljenih sistemov.
- Posebnosti računalniških omrežij, ki vplivajo na porazdeljene sisteme, ne bomo obravnavali.
- V nadaljevanju predpostavljamo, da neka omrežna povezava med vozlišči obstaja (ne glede na njeno topologijo).

## Razlike med različnimi SUPPB

- Obstajajo številne vrste SUPPB
- Dva pomembna kriterija, po katerih se SUPPB razlikujejo, sta:
  - Stopnja homogenosti in
  - Stopnja lokalne avtonomnosti



## Homogeni in heterogeni SUPPB..

### ▪ Homogeni SUPPB:

- Vsa mesta uporabljajo enak SUPB;
- Enostavni za načrtovanje in obvladovanje;
- So razširljivi (enostavno dodajanje novih mest);
- Omogočajo paralelnost izvajanja.

## Homogeni in heterogeni SUPPB

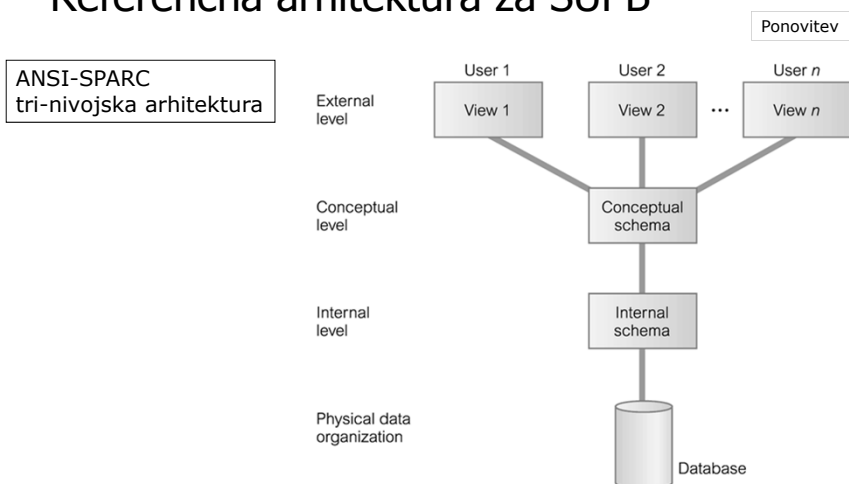
### ▪ Heterogeni SUPPB:

- Mesta uporabljajo različne SUPB. Lahko temeljijo na različnih podatkovnih modelih (relacijskih, objektnih,...)
- Navadno so posledica integracije sistemov, ki so samostojno nastali v različnih časovnih obdobjih;
- Za razumevanje med različnimi SUPB so potrebne prevedbe;
- Prevedbe se obnesejo za poizvedovanja, problematično obvladovanje transakcij, zagotavljanje sočasnosti dostopa, obnovitev podatkov ipd.;
- Potrebno zagotoviti globalno konceptualno shemo;
- Težko zagotoviti transparentnost heterogenosti za uporabnika;

## Funkcionalnosti SUPPB

- SUPPB mora zagotavljati vsaj naslednje funkcije:
  - Vodenje evidence o lokaciji podatkov (porazdelitev, fragmentacija, replikacija) – razširitev systemskega kataloga;
  - Porazdeljeno izvajanje poizvedb
  - Obvladovanje porazdeljenih transakcij
  - Obvladovanje replikacije
  - Obnavljanje PPB po nesrečah
  - Zagotavljanje varnosti
  - Vodenje razširjenega kataloga

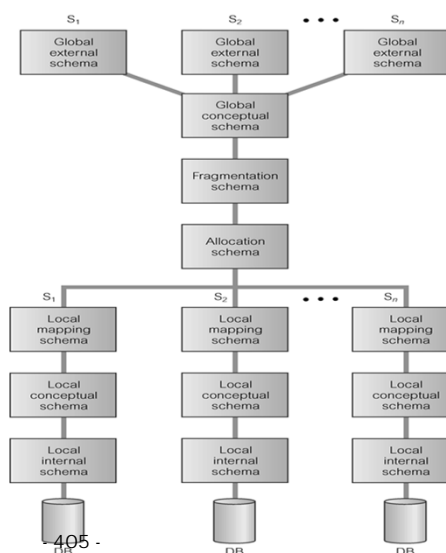
## Referenčna arhitektura za SUPB



## Referenčna arhitektura za SUPPB

Zaradi različnih vrst SUPPB težko določiti referenčno.

Shema predstavlja zgolj primer.



PODATKOVNE BAZE 2 - VSP

- 405 -

informatika  
v Ljubljani

## Fragmentacija podatkov v PPB..

- V PPB moramo določiti, katero vozlišče naj ima določen segment podatkov.
- Vzemimo primer, kjer velja:
  - Ni replikacije (vsak segment na samo enem vozlišču)
  - Gre za relacijsko PB (enako velja za ostale vrste PB)
  - Imamo relacijsko shemo, ki bi jo radi porazdelili po vozliščih

PODATKOVNE BAZE 2 - VSP

- 406 -

Fakulteta za računalništvo in informatiko  
Univerza v Ljubljani

## Fragmentacija podatkov v PPB

- Preden izvedemo fragmentacijo, določimo logično enoto za fragmentiranje.
- Najenostavnejše je za logično enoto vzeti relacijo.
- Kadar želimo manjše enote, se navadno odločimo za horizontalne ali vertikalne fragmente posameznih relacij.

### DELAVEC

Ime	Priimek	EMSO	Drojstva	Spol	Dohodek	Enota
Jože	Mali	1201975500100	12.01.1975	M	6.439.000	E1
Marko	Jug	2202979500222	22.02.1979	M	3.567.000	E2
Miha	Mitrov	1404970500801	14.04.1970	M	4.876.000	E2
Tadej	Plevel	1303969500103	13.03.1969	M	6.439.000	E1
Francka	Ris	1201975505999	12.01.1975	Ž	6.109.000	E3
Joža	Kruh	0612970505133	06.12.1970	Ž	2.189.000	E3
Karmen	Kruh	1808968505100	18.08.1968	Ž	8.978.000	E1
Klemen	Mihelj	1306972502122	13.06.1972	M	5.234.000	E4
Tilen	Fras	1207971500345	12.07.1971	M	6.439.000	E5
Gregor	Coli	0707969500566	07.07.1969	M	6.109.000	E3
Marija	Jeklič	1402965505456	14.02.1965	Ž	2.189.000	E3

Horizontalni fragment

Vertikalni fragment

## Horizontalna fragmentacija..

- Horizontalni fragment relacije je podmnožica n-teric relacije.
- N-terice, ki spadajo v fragment, so določene s pogojem nad enim ali več atributov relacije.
- Pri horizontalni fragmentaciji dobimo podmnožice n-teric, ki imajo nek logični pomen.
- Izpeljana horizontalna fragmentacija – fragmentiranje relacije na osnovi povezanih relacij...

## Horizontalna fragmentacija..

- Horizontalni fragment nad relacijo R lahko zapišemo kot  $\sigma_{C_i}(R)$ , kjer je:
  - $\sigma$  selekcija (operacija relacijske algebre)
  - $C_i$ ;  $i = 1..n$  nabor pogojev
  - R relacija
- Kadar vsaka n-terica relacije R zadošča pogoju  $(C_1 \text{ OR } C_2 \text{ OR } \dots \text{ OR } C_n)$  pravimo, da gre za popolno horizontalno fragmentacijo.

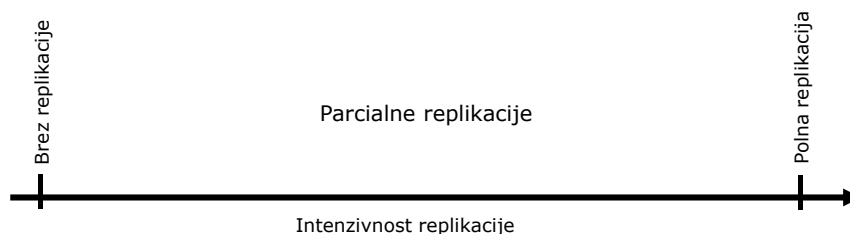
## Horizontalna fragmentacija

- Kadar je horizontalna fragmentacija disjunktna, velja, da nobena n-terica ni v več kot enem fragmentu.
- Tedaj velja, da nobena n-terica ne zadošča pogoju ( $C_i \text{ AND } C_j$ ), kjer  $i \neq j$
- Kadar je v uporabi horizontalna fragmentacija, lahko osnovno relacijo rekonstruiramo s pomočjo operacije UNION.

## Podatkovna replikacija..

- Z replikacijo večamo razpoložljivost podatkov.
- Polno replicirana PPB ima na vseh vozliščih vse podatke. Podatki so na voljo, če vsaj eno vozlišče deluje.
  - Večja učinkovitost globalnih poizvedb (podatki so velikokrat najdeni lokalno).
  - Močno obremeni sistem ob spremembah (dodajanje, spreminjanje)
- Drug ekstrem so PPB brez replikacije. Vsak fragment se nahaja na natanko enem vozlišču.

## Podatkovna replikacija



- Parcialne replikacije so značilne za okolja, kjer zaposleni delajo na terenu
  - PB nosijo s seboj na teren
  - Kasneje replicirajo s centralnim strežnikom.

## Alokacija podatkov

- Vsak fragment (in vsaka njegova kopija) mora biti dodeljen nekemu vozlišču v porazdeljenem sistemu.
- Proces dodeljevanja imenujemo podatkovna alokacija ali podatkovna distribucija.
- Izbira vozlišč in stopnja replikacije je odvisna od želene učinkovitosti sistema in razpoložljivosti podatkov ter pogostosti in tipa transakcij, ki naj bi na posameznem vozlišču tekle.



## Strategije alokacije podatkov

- Ločimo štiri strategije alokacije fragmentov:
  - Centralizirana: podatki na enem mestu
  - Porazdeljena: podatki porazdeljeni po mestih; ni replikacije
  - Polno replicirana: vsako mesto vsebuje popolno kopijo podatkovne baze
  - Delno replicirana: nekateri podatki so replicirani na več mestih

## Ravni transparentnosti v SUPPB

- V SUPPB porazdeljenost transparentna za uporabnike.
  - Podatki o implementaciji porazdeljenosti so uporabniku skriti.
- V SUPPB obstaja več ravni transparentnosti:
  - Transparentnost porazdeljenosti
  - Transparentnost transakcij
  - Transparentnost učinkovitosti
  - Transparentnost SUPB
- Popolna transparentnosti je predmet diskusij...

## Transparentnost porazdelitve

- Uporabnik PB dojema kot logično celoto.
- Obstaja več vrst transparentnosti porazdelitve:
  - Transparentnost fragmentacije: ni potrebno vedeti, ali so podatki fragmentirani;
  - Transparentnost lokacije podatkov: Ni potrebno vedeti, kje so podatki shranjeni;
  - Transparentnost lokalnega mapiranja: uporabnik mora vedeti, kakšna je fragmentacija in kje so fragmenti shranjeni.

## Transparentnost transakcij

- Zagotavlja, da porazdeljene transakcije ohranjajo celovitost in skladnost podatkov v PB.
- Porazdeljena transakcija
  - dostopa do podatkov, ki so shranjeni na različnih lokacijah.
  - sestavljena iz več podtransakcij - za vsako mesto dostopa svoja transakcija.
  - Vsako podtransakcijo predstavlja svoj agent.

## Transparentnost učinkovitosti

- Transparentnost učinkovitosti zahteva, da SUPPB deluje tako, kot bi bil centraliziran SUPB.
  - Porazdeljenost arhitekture ne sme vplivati na učinkovitost sistema...
- Transparentnost učinkovitosti zahteva tudi, da SUPPB identificira časovno najugodnejšo strategijo za izvedbo zahteve.

## Procesor poizvedb..

- SUPB
  - Procesor poizvedb izračuna vsako zahtevo po podatkih
  - Poišče najugodnejšo strategijo izvedbe, ki jo sestavlja zaporedje operacij nad podatkovno bazo
- SUPPB
  - V porazdeljenem okolju procesor poizvedb podatkovne zahteve pretvori v zaporedja operacij nad lokalnimi podatkovnimi bazami.
  - Večja kompleksnost zaradi fragmentacije, replikacije in alokacije podatkov.

## Procesor poizvedb

- Procesor poizvedb v SUPPB se odloča:
  - Kateri fragment prebrati?
  - Katero kopijo fragmenta uporabiti, če je repliciran?
  - Katero lokacijo uporabiti?
- Pri izbiri strategije procesor izračuna ceno vsake strategije:
  - Čas dostopa (I/O) do fizičnih podatkov na disku
  - Čas procesorja za obdelavo podatkov v glavnem pomnilniku
  - Čas prenosa podatkov po omrežju

## Transparentnost SUPB

- Transparentnost SUPB omogoča, da se uporabniku ni potrebno zavedati različnosti med lokalnimi SUPB.
- Velja za heterogene sisteme.
- Ena najtežjih oblik zagotavljanja transparentnosti.

## Dvanajst pravil za SUPPB

- Pravila napisal C. J. Date, 1987\*
- Osnovna ideja:
  - sistem za upravljanje s porazdeljenimi podatki – SUPPB bi moral za uporabnika delovati kot navaden sistem za upravljanje s podatki.

P<sub>1</sub>: Lokalna avtonomnost

P<sub>7</sub>: Porazdeljeno izvajanje poizvedb

P<sub>2</sub>: Neodvisnost od centralnega mesta

P<sub>8</sub>: Porazdeljeno procesiranje transakcij

P<sub>3</sub>: Neprekinjenost delovanja

P<sub>9</sub>: Neodvisnost od strojne opreme

P<sub>4</sub>: Lokalna neodvisnost

P<sub>10</sub>: Neodvisnost od operacijskega sistema

P<sub>5</sub>: Neodvisnost od fragmentacije

P<sub>11</sub>: Neodvisnost od omrežja

P<sub>6</sub>: Neodvisnost od replikacije

P<sub>12</sub>: Neodvisnost od SUPB

\* C. J. Date, 1987, Twelve Rules for a Distributed Database, *Computer World*, 21(23), 75-81.