

# Napredni modeli i baze podataka

Predavanja  
Studen 2015.

---

## 4. Vremenske baze podataka

# Sadržaj

- Upravljanje vremenom u standardnim SUBP - motivacijski primjeri
- Modeliranje vremena
- SQL standard i temporalno proširenje
  - Vremenski tipovi podataka i operacije
  - Vremenske dimenzije (vrijeme valjanosti i transakcijsko vrijeme)
  - Temporalne relacije
  - Temporalna ograničenja integriteta
  - Temporalni upiti
- PostgreSQL

# Motivacijski primjer 1

## Dužnosnici na FER-u

REPUBLICA HRVATSKA  
SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

**DOPUNSKA ISPRAVA  
O STUDIJU**

Dopunska isprava o studiju nadležna je prema predlošku koji su razvili Europska komisija, Vijeće Europe i UNESCO/CERES. Svrha dopunske isprave jest pružanje dodatnih informacija o studiju i o studiju u skladu s međunarodnom transparentnošću i pravilima akademskog i stručnog priznavanja obrazovnih kvalifikacija (diploma, uspjeha, certifikata i dr.). Dopunska isprava sadrži opće podatke, naziv, katedru, sadržaj i status studija koji je prethodila i svojstvo studija u skladu s izvornom ispravom o stručnoj kvalifikaciji. Dopunska isprava ne sadrži vrijednosne sudove, izjave o prihvatljivosti ili prijedloge za priznavanje. Svaki osam potpisuje mora biti potpisan. Ako neki podatak nedostaje, mora biti potpisan odgovarajuće.

Ukupno ECTS bodova: 121      Ukupna ocjena studija: 5.000  
(težinski prosjek ocjena)

U Zagrebu, 13. srpnja 2015.

M.P.      DEKAN

Prof. dr. sc. Mislav Grgić

Model omogućuje pohranu aktualnih dužnosnika.

Što je s prethodnicima?

- Trenutni dekan: 1.10.2014 – 30.09.2016
- Prethodnik: 1.10.2010 – 30.09.2014
- Prethodnikov prethodnik: 1.10.2006 – 30.09.2010
- Itd.

**Kako popraviti model podataka?**

orgJed

sifOrgJed	nazivOrgJed	sifNadOrgJed
100	Fakultet elektrotehnike i računarstva	
103	Zavod za primijenjenu matematiku	1
...	...	...

funkcija

sifFun	nazivFun
10	Dekan
11	Prodekan za nastavu
12	Predstojnik zavoda
...	...

osoba

sifOsoba	ime	prezime
100	Mislav	Grgić
101	Marko	Delimar
102	Ilko	Brnetić
...	...	...

obavljaFunkciju

sifFun	sifOrgJed	sifOsoba
10	100	100
11	100	101
12	103	102
...	...	...

U Zagrebu, 25. veljače 2013.

M.P.

DEKAN

Prof. dr. sc. Nedjeljko Perić

U Zagrebu, 29. lipnja 2010.

M.P.

DEKAN

Prof. dr. sc. Vedran Morar

# Motivacijski primjer 1

## Dužnosnici na FER-u

Prvi pokušaj popravljivanja modela:

osoba			obavljaFunkciju			
<u>sifOsoba</u>	ime	prezime	<u>sifFun</u>	<u>sifOrgJed</u>	sifOsoba	<u>datumOd</u>
50	Nedjeljko	Perić	10	100	51	1.10.2006
51	Vedran	Mornar	10	100	50	1.10.2010
...	...	...	10	100	100	1.10.2014
100	Mislav	Grgić	...	...	...	...

Dekan na dan '1.11.2008'?

```
SELECT osoba.ime, osoba.prezime
FROM osoba, obavljaFunkciju
WHERE osoba.sifOsoba = obavljaFunkciju.sifOsoba
AND sifFun = 10
AND sifOrgJed = 100
AND '1.11.2008' >= obavljaFunkciju.datumOd
AND '1.11.2008' <
  (SELECT MIN(datumOd)
   FROM obavljaFunkciju of
   WHERE obavljaFunkciju.sifFun = of.sifFun
        AND obavljaFunkciju.sifOrgJed = of.sifOrgJed
        AND obavljaFunkciju.datumOd < of.datumOd);
```

ime	prezime
Vedran	Mornar

Dekan na dan '1.11.2014'?

ime	prezime

Zašto?

Trovalentna logika.  
NULL vrijednosti.  
Upiti postaju kompliciraniji.  
Kako izgleda upit koji uvijek vraća ispravnu vrijednost?

# Motivacijski primjer 1

## Dužnosnici na FER-u

```
SELECT osoba.ime, osoba.prezime
FROM osoba, obavljaFunkciju
WHERE osoba.sifOsoba = obavljaFunkciju.sifOsoba
AND sifFun = 10
AND sifOrgJed = 100
AND datumOd <= '15.11.2014'
AND ('15.11.2014' <
    (SELECT MIN(datumOd)
     FROM obavljaFunkciju of
     WHERE obavljaFunkciju.sifFun = of.sifFun
           AND obavljaFunkciju.sifOrgJed = of.sifOrgJed
           AND obavljaFunkciju.datumOd < of.datumOd)
OR NOT EXISTS
 (SELECT *
  FROM obavljaFunkciju of
  WHERE obavljaFunkciju.sifFun = of.sifFun
        AND obavljaFunkciju.sifOrgJed = of.sifOrgJed
        AND obavljaFunkciju.datumOd < of.datumOd) )
```

### obavljaFunkciju

<u>sifFun</u>	<u>sifOrgJed</u>	sifOsoba	<u>datumOd</u>
10	100	51	1.10.2006
10	100	50	1.10.2010
10	100	100	1.10.2014
...	...	...	...

Dekan na dan '1.11.2014'?

ime	prezime
Mislav	Grgić

Ne želim pisati  
komplicirane upite kao  
odgovore na jednostavna  
pitanja!

Ima li model još mana?

1. Model dozvoljava unos dekana za *datumOd*  
2.10.2006 ili 3.10.2006

Mandat dužnosnika traje određeni vremenski period:  
*datumOd-datumDo*.

Kako to osigurati modelom?

# Motivacijski primjer 1

## Dužnosnici na FER-u

Drugi pokušaj popravljivanja modela:

### obavljaFunkciju

<u>sifFun</u>	<u>sifOrgJed</u>	<u>sifOsoba</u>	<u>datumOd</u>	<u>datumDo</u>
10	100	51	1.10.2006	30.09.2010
10	100	50	1.10.2010	30.09.2014
10	100	100	1.10.2014	30.09.2016
...	...	...	...	...

Upit za dohvat dekana na određeni datum postaje jednostavniji.

Međutim, u općem slučaju *datumDo* ne mora biti poznat => ni upit ne mora biti jednostavniji.

Ima li model još mana?

1. Je li {*sifFun*, *sifOrgJed*, *datumOd*} dobar ključ?
2. Je li {*sifFun*, *sifOrgJed*, *datumDo*} dobar ključ?
3. Je li {*sifFun*, *sifOrgJed*, *datumOd*, *datumDo*} dobar ključ?

Nijedan od gornjih kandidata nije ispravan ključ! Jedinstvenost se provjerava operatorom =, a to nije dovoljno.

Treba mi novi tip podatka koji će obuhvaćati sve datume u periodu *datumOd* - *datumDo*.

<u>sifFun</u>	<u>sifOrgJed</u>	<u>sifOsoba</u>	<u>periodOdDo</u>
10	100	51	[1.10.2006, 30.09.2010]
10	100	50	[1.10.2010, 30.09.2014]
10	100	100	[1.10.2014, 30.09.2016]
...	...	...	...

Dodatno:

- kraj perioda ne smije biti < početak perioda
- ne smije postojati dekan za period koji je sadržan u nekom od postojećih perioda
- ...

# Motivacijski primjer 1

## Dužnosnici na FER-u

Treba mi:

- novi tip podatka – „period” (DATE period, TIMESTAMP period,...)
- funkcije za obavljanje operacija s novim tipovima podataka npr.
  - preklapaju li se period1 i period2
  - sadržava li period1 period2
  - period1 + period2, period1 – period2
  - ...
- podrška za novu paradigmu pri osiguravanju
  - PRIMARY KEY ograničenja
  - FOREIGN KEY ograničenja
  - ...

kako ih definirati s logičkog stanovišta?
- podrška za nove metode spajanja (JOIN) koje uključuju operatore za rad s novim tipovima podataka (temporalni upiti)
- ...

## Motivacijski primjer 2

### Evidencija održavanja i prisustvovanja nastavi

predmet		dvorana		osoba				
sifPredmet	nazivPredmet	oznDvorana	kapacPred	sifOsoba	ime	prezime		
rezDvorana				prisustvo				
oznDvorana	sifPredmet	sifNast	terminOd	terminDo	oznDvorana	sifStudent	terminOd	terminDo
A101	1	100	12.11.2015 08:00	12.11.2015 11:00	A101	23459789	12.11.2015 08:17	12.11.2015 10:55
A101	2	101	12.11.2015 11:00	12.11.2015 13:00	A101	23459789	12.11.2015 08:21	12.11.2015 10:59
...	...				...			

Što je ključ *rezDvorana* i *prisustvo*?

Kada bismo imali *PERIOD* tip podatka na raspolaganju:

rezDvorana				prisustvo		
oznDvorana	sifPredmet	sifNast	terminOdDo	oznDvorana	sifStudent	terminOdDo
A101	1	100	[12.11.2015 08:00, 2015-11-12 11:00]	A101	23459789	12.11.2015 08:17, 12.11.2015 10:55
A101	2	101	[12.11.2015 11:00, 2015-11-12 13:00]	A101	20006789	12.11.2015 08:21, 12.11.2015 10:59
...	...			...		

$K1_{\text{REZDORANA}} = \{\text{oznDvorana}, \text{terminOdDo}\}$   
 $K2_{\text{REZDORANA}} = \{\text{sifNastavnik}, \text{terminOdDo}\}$

$K_{\text{PRISUSTVO}} = \{\text{sifStudent}, \text{terminOdDo}\}$

Smije li u *rezDvorana* postojati n-torka: 

A101	3	101	[12.11.2015 09:15, 2015-11-12 10:00]
------	---	-----	--------------------------------------

n-torka: 

A101	3	101	[12.11.2015 07:30, 2015-10-05 09:30]
------	---	-----	--------------------------------------



## Motivacijski primjer 2

### Evidencija održavanja i prisustvovanja nastavi

rezDvorana				prisustvo		
<u>oznDvorana</u>	<u>sifPredmet</u>	<u>sifNastavnik</u>	<u>terminOdDo</u>	<u>oznDvorana</u>	<u>sifStudent</u>	<u>terminOdDo</u>
A101	1	100	[2015-11-12 08:00, 2015-10-05 11:00)	A101	23459789	2015-11-12 08:17, 2015-10-05 10:55
A101	2	101	[2015-11-12 11:00, 2015-11-12 13:00)	A101	20006789	2015-11-12 08:21, 2015-10-05 10:59
...	...			...		

Ako je  $K_{\text{REZDVORANA}} = \{\text{oznDvorana}, \text{terminOdDo}\}$  što znači sljedeći strani ključ :

```
ALTER table prisustvo
```

```
... FOREIGN KEY (oznDvorana, terminOdDo) REFERENCES rezDvorana (oznDvorana, terminOdDo)
```

- Da bi se n-torka iz *prisustvo* referencirala na n-torku iz *rezDvorana* moraju li *rezDvorana.terminOdDo* i *prisustvo.terminOdDo* biti identični?
- Je li dovoljno da *prisustvo.terminOdDo* bude
  - sadržan u *rezDvorana.terminOdDo*?
  - obuhvaća *rezDvorana.terminOdDo*?
  - Djelomično se preklapa s *rezDvorana.terminOdDo*?
  - ...
- U tradicionalnom relacijskom modelu se integriteta ključa (bilo primarnog bilo stranog) uvijek čuvao pomoću operatora jednakosti (=).
- To nam više nije dovoljno dobro.

# Motivacijski primjer 3

## FER-ova organizacijska shema kroz povijest

Od 2005

...

1926 -1956

...

1919-1926

### Fakultet elektrotehnike i računarstva

- Zavod za primijenjenu fiziku
- Zavod za primijenjenu matematiku
- Zavod za primijenjeno računarstvo
- Zavod za osnove elektrotehnike i električka mjerenja
- Zavod za elektrostrojstvo i automatizaciju
- Zavod za visoki napon i energetiku
- Zavod za telekomunikacije
- Zavod za elektroničke sustave i obradbu informacija
- Zavod za automatiku i računalno inženjerstvo
- Zavod za elektroakustiku
- Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave
- Zavod za radiokomunikacije

### Tehnički fakultet

- Zavod za primijenjenu fiziku
- Zavod za primijenjenu matematiku
- Zavod za osnove elektrotehnike i električka mjerenja
- Zavod za elektrostrojstvo i automatizaciju
- Zavod za visoki napon i energetiku
- Zavod za telekomunikacije
- Zavod za elektroničke sustave i obradbu informacija
- Zavod za automatiku i računalno inženjerstvo
- Zavod za elektroakustiku
- Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave
- Zavod za radiokomunikacije

### Visoka tehnička škola

- Zavod za primijenjenu matematiku
- Zavod za osnove elektrotehnike i električka mjerenja
- Zavod za elektrostrojstvo i automatizaciju

Kako odgovoriti na pitanja:

- Promjene u broju zaposlenih po zavodima
- Promjene u predmetima po zavodima
- Promjene u broju znanstvenih publikacija po zavodima
- ...

Što je s FER-ovom organizacijskom shemom između 1956 i 2005?

## Motivacijski primjer 3

### FER-ova organizacijska shema kroz povijest

orgJed	<u>sifOrgJed</u>	nazivOrgJed	sifNadOrgJed
	100	Fakultet elektrotehnike i računarstva	
	...	...	...

Kako pratiti povijesne promjene u organizacijskoj strukturi?

Prvi pokušaj popravljjanja modela:

<u>sifOrgJed</u>	nazivOrgJed	...	vrijediOd	vrijediDo
1	Visoka tehnička škola		dd.mm.1919	dd.mm.1926
2	Zavod za primijenjenu matematiku		dd.mm.1919	∞
5	Tehnički fakultet	...	dd.mm.1926	30. 06.1956
78	Elektrotehnički fakultet	...	01.07.1956	06. 02.1995
100	Fakultet elektrotehnike i računarstva	...	07.02.1995	∞
135	Zavod za primijenjeno računarstvo		19.10.2005	∞

Ima li mana?

- Kada je 7.2.1995 promijenjen naziv institucije što se dogodilo s n-torkama koje su se referencirale na *orgjed.sifOrgJed* = 78 (npr. *orgJed.sifNadOrgJed*, obnašatelji funkcija,...)?
- Kada je 2005 osnovan Zavod za primijenjeno računarstvo što se dogodilo sa *zaposlenik.sifOrgJed* u n-torkama ZPR-ovih zaposlenika koji su do tada bili zaposlenici Zavoda za primijenjenu matematiku?

Što sa sifOrgJed za npr. prof. Kalpića?

zaposlenik	<u>sifOsoba</u>	...	sifOrgJed	datZaposl
	598		12	01.05.1994
	798		12	01.03.1978
	...			

## Motivacijski primjer 3

### FER-ova organizacijska shema kroz povijest

<u>sifOrgJed</u>	<u>nazivOrgJed</u>	...	<u>vrijediOd</u>	<u>vrijediDo</u>
1	Visoka tehnička škola		dd.mm.1919	dd.mm.1926
2	Zavod za primijenjenu matematiku		dd.mm.1919	∞
5	Tehnički fakultet	...	dd.mm.1926	30. 06.1956
78	Elektrotehnički fakultet	...	01.07.1956	06. 02.1995
100	Fakultet elektrotehnike i računarstva	...	07.02.1995	∞
135	Zavod za primijenjeno računarstvo		19.10.2005	∞

- Zadržavanje stare vrijednosti referencirajućeg atributa ne prati promjene.
- Izmjena referencirajućeg atributa na novu vrijednost zaboravlja povijest.

Drugi pokušaj popravljavanja modela:

#### orgJed

<u>sifOrgJed</u>	<u>nazivOrgJed</u>	...
12	Zavod za primijenjenu matematiku	
1	Fakultet elektrotehnike i računarstva	...
135	Zavod za primijenjeno računarstvo	

#### orgJedPovijest

<u>sifOrgJed</u>	<u>nazivOrgJed</u>	...	<u>vrijediOd</u>	<u>vrijediDo</u>
1	Visoka tehnička škola	...	dd.mm.1919	dd.mm.1926
1	Tehnički fakultet		dd.mm.1926	30. 06.1956
1	Elektrotehnički fakultet	...	01.07.1956	06. 02.1995

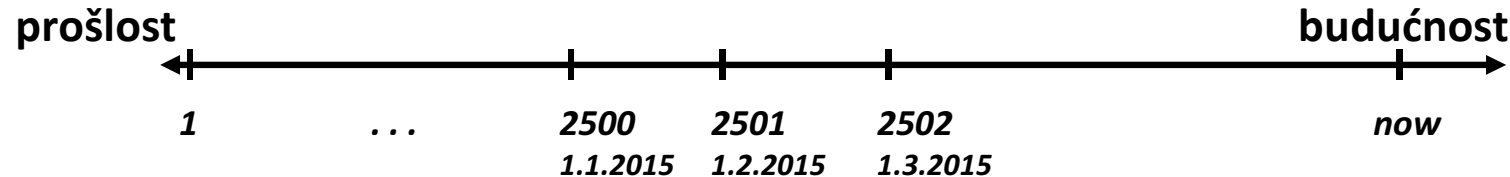
- Ključ u orgJed se ne mijenja
  - Nestaju problemi sa stranim ključem
- orgJed odražava trenutno stanje
- Povijesne promjene moraju se moći rekonstruirati pomoću *orgJed* i *orgJedPovijest*

# Modeliranje vremena

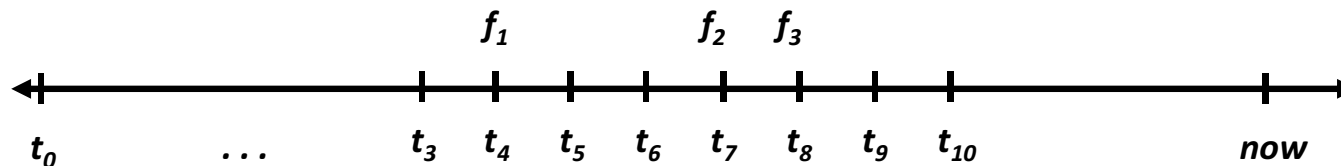
- Vrijeme se uglavnom shvaća kao jednodimenzionalni kontinuum koji se prostire od prošlosti prema budućnosti
- Može se modelirati kao
  - **konačno/beskonačno**
    - Početak – početak vremenske linije (PostgreSQL – 4713 p.n.e.)
    - Zauvijek – kraj vremenske linije (PostgreSQL – 294276 n.e.)
  - **diskretno/kontinuirano**
    - Diskretni modeli izomorfni su prirodnim ili cijelim brojevima
      - Svaki prirodan ili cijeli broj odgovara osnovnoj jedinici vremena – ***chrononu*** (kvantum vremena, diskretna i nedjeljiva jedinica vremena kao dio hipoteze da vrijeme nije kontinuirano)
      - Chronone možemo grupirati u veće jedinice vremena (sati, dani ...)
    - Kontinuirani modeli izomorfni su realnim brojevima
      - Svaki realan broj odgovara jednom vremenskom trenutku
  - **apsolutno/relativno**
    - 31. listopada 2015. 9:15

# Vrijeme

- U diskretnom modelu kalendarske vrijednosti (godine, datume, ...) možemo preslikati u cjelobrojne konstante:



- Fenomeni iz stvarnog svijeta se
  - dogode u određenom trenutku (*chrononu*) i nemaju trajanje ili
  - traju/istinite su u nekom vremenskom intervalu ili periodu



- $\langle t_4, f_1 \rangle, \langle t_7, f_2 \rangle, \langle t_8, f_3 \rangle$  npr.  $f_1$ : dana  $t_4$  je osobi na tekući račun uplaćena plaća
- $\langle [t_4, t_7), f_1 \rangle, \langle [t_7, t_8), f_2 \rangle, \langle [t_8, now), f_3 \rangle$  npr.  $f_1$ : osoba je u periodu  $[t_4, t_7)$  mogla biti u minusu na tekućem računu u iznosu od 5000.00 kn

Sadašnji trenutak **NOW**

- Teško ga je formalizirati
- Granica između prošlosti i budućnosti
- Stalno se povećava

## Stanja i događaji

- Fenomeni iz stvarnog svijeta čija su nam vremenska svojstva zanimljiva mogu se opisati kao **stanja** ili kao **događaji**
  - **Stanja** opisuju činjenice vezane uz neki objekt u bazi podataka koje su istinite u nekom vremenskom intervalu ili periodu. Te se činjenice ne smatraju točnima izvan pridruženog perioda.
  - **Događaji** opisuju činjenice vezane uz neki objekt u bazi podataka koje su se dogodile u određenom trenutku (*chrononu*) i nemaju trajanje.

# Stanja i događaji

- Primjer relacija *tekRacunLimit* i *uplataSplata* kao relacije stanja i relacije događaja.

tekRacunLimit

brTekRacun	iznosLimit	vrijediOd	vrijediDo
2341906787	5000	1.1.2015.	1.5.2015.
2341906787	8000	2.6.2015.	NULL
2211906783	10000	1.1.2005.	NULL
2192345667	9000	1.1.2015.	1.7.2015.
2192345667	5000	2.7.2015.	NULL
5361906785	7000	20.5.2015.	NULL

stanje

uplataSplata

brTekRacun	iznos	datum
2341906787	5000.00	10.06.2015.
2341906787	-3500.00	12.06.2015.
2192345667	-1000.00	01.07.2015.
2192345667	-5000	20.05.2015.

događaj

vlasnik

id	ime	prezime
100	Ivan	Ivkošić
101	Petar	Petrović
102	Marko	Marković
105	Ante	Antić

tekRacun

brTekRacun	idVlasnik
2341906787	100
2211906783	101
2192345667	102
5361906785	105



## Upravljanje vremenom u standardnim relacijskim SUBP

- Baze podataka kojima upravljaju standardni relacijski SUBP najčešće odražavaju trenutno stanje svijeta (u mjeri u kojoj je ono poznato bazi podataka)
- UPDATE: izmjenom se gubi informacija o prethodnom stanju.
- Čuvanje povijesnih informacija u standardnim SUBP je moguće jedino ako **aplikacija** sama upravlja vremenom.
- Standardni SUBP/SQL (uključujući SQL Server, IBM Informix, Microsoft Access, MySQL,...) pružaju ograničenu potporu upravljanju vremenom u obliku nekoliko tipova podataka
  - **date** (definira godinu, mjesec i dan)
  - **time** (definira sat, minutu i sekundu)
  - **timestamp** (definira godinu, mjesec, dan, sat, minutu i sekundu)
- Mane upravljanja vremenom kroz **aplikaciju**:
  - Semantika vremena, operacije te ograničenja integriteta moraju biti ugrađeni izravno u aplikaciju
  - Složeni upiti podložni greškama
  - Nedjelotvorno izvođenje upita

# Implementacija vremenskih koncepata

Dva su moguća pristupa. Podršku vremenskim konceptima ugraditi u:

## 1. korisničku aplikaciju

- Koristiti postojeće tipova podataka i postojeći upitni jezik
- Sve vremenske koncepti ugraditi izravno u aplikaciju
- Nedjelotvorno i podložno greškama

## 2. SUBP

- Proširiti nevremensku shemu vremenskim tipovima podataka
- Proširiti algebru i upitni jezik dodatnim operatorima kao što je temporalno spajanje, temporalna selekcija, temporalna projekcija i sl.
- Implementirati temporalna integritetska ograničenja
- Omogućiti praćene povijesnih podataka (sa stanovišta vremena valjanosti podataka u stvarnom svijetu i sa stanovišta vremena evidentiranja u bazu podataka)

# Vrijeme i SQL standard

- Tipovi podataka
  - SQL-86 i SQL-89: ne predviđaju vremenske tipove podataka
  - SQL-92: DATE, TIME, TIMESTAMP i INTERVAL
  - SQL-99: PERIOD ali ne kao novi tip nego objekt čije su granice neki od osnovnih vremenskih tipova
- Operacije
  - SQL-99: za PERIOD (osim =, <, IS NULL) predviđa samo OVERLAPS
- Upitni jezik
  - Richard Snodgrass 1992 predlaže proširivanje SQL-a vremenskim konceptima
  - 1993 pojavio se TSQL2 (razvila ga „temporal database community”)
  - SQL-99: uključeni dijelovi TSQL2 (ali je zbog brojnih kritika doživio izmjene)
  - SQL-2011: izmjene TSQL2

## SQL:2011 i temporalna podrška

- Definicija *closed-open* semantike
- Definicija relacija vremena valjanosti
- Definicija relacija transakcijskog vremena. Periodi transakcijskog vremena se održavaju automatski.
- Definicija bitemporalnih relacija
- Sintaksa za izmjenu i brisanje n-torki kojima se prate vrijeme valjanosti s automatskim dijeljenjem (*split*) vremenskog perioda
- Temporalni primarni ključ (uzima u obzir period vremena valjanosti)
- Temporalni strani ključ
- Proširenje upitnog jezika temporalnim predikatima za vremenske periode: CONTAINS, OVERLAPS, EQUALS, PRECEDES, SUCCEEDS, IMMEDIATELY PRECEDES, i IMMEDIATELY SUCCEEDS (izmijenjeni Allenovi operatori)
- Sintaksa za trenutne i sekvencirane upite

# DATE, DATETIME, INTERVAL, PERIOD

- Osnovni vremenski tipovi podataka:
  - **instant**  
Određeni chronon na vremenskoj liniji diskretnog modela ili točka na vremenskoj liniji kontinuiranog modela (npr. 12. studeni 2015, 8:50:59)
  - **interval**  
Neusidreni interval na vremenskoj liniji, ima samo trajanje (npr. 2 sata, 3 dana,...)
  - **period**  
Usidreni (apsolutno definirani) interval na vremenskoj liniji (npr. zimski semestar 2015/16: 05.10.2015 – 29.1.2016)
    - Za razliku od standardnih tipova podataka tip period nije uređen, nad njim je definirano samo parcijalno uređenje
  - **periods**  
Skup disjunktnih usidrenih intervala, naziva se još i vremenskim elementom (engl. *temporal element*)

# PostgreSQL: Vremenski tipovi podataka

Tip	Opis	donja granica	gornja granica	rezolucija
date	datum	4713 BC	5874897 AD	dan
timestamp[ (p) ] [without time zone]	datum i vrijeme, ne uključuje vremensku zonu	4713 BC	294276 AD	mikrosekunda
timestamp[ (p) ] with time zone	datum i vrijeme, uključuje vremensku zonu	4713 BC	294276 AD	mikrosekunda
time [ (p) ][without time zone ]	vrijeme u danu (bez datuma)	00:00:00	24:00:00	mikrosekunda
time [ (p) ] with time zone	vrijeme u danu (s vremenskom zonom)	0:00:00+1459	24:00:00-1459	mikrosekunda
<b>interval i period</b>				
interval [ fields ] [ (p) ]	Vremenski interval (neusidreni)	-178000000 godina	+178000000 godina	mikrosekunda
daterange	Usidreni vremenski interval (period) s granicama <i>date</i> tipa	-178000000 godina	+178000000 godina	mikrosekunda
tsrange	Usidreni vremenski interval (period) s granicama <i>timestamp</i> tipa, bez vremenske zone	4713 BC	294276 AD	mikrosekunda
tstzrange	Usidreni vremenski interval (period) s granicama <i>timestamp</i> tipa, s vremenskom zonom	4713 BC	294276 AD	mikrosekunda
<b>periods – nije implementirano</b>				

timestamp, timestampz, time, interval

P – broj decimalnih mjesta u vrijednosti za sekunde (0-6).

# PostgreSQL: timestamp, timestampz, vremenske zone

- Timestamp pohranjuje informaciju o vremenu od godine do dijelova sekunde

```
SET DateStyle = 'German, DMY';  
SELECT '2015-11-12 08:15:00.123456'::timestamp(3)
```

**timestamp(3) without time zone**

12.11.2015 08:15:00.123

```
SELECT '12.11.2015 08:15:00.123456'::timestamp(6)
```

**timestamp(3) without time zone**

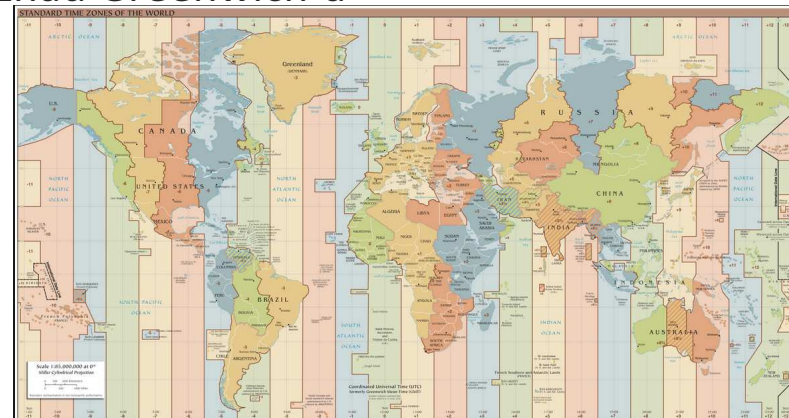
12.11.2015 08:15:00.123456

```
SELECT '12.11.2015 08:15:00.123456'::timestamp(6)  
WITH TIME ZONE
```

**timestamp(3) with time zone**

12.11.2015 08:15:00.123456 CET

- SQL koristi Coordinated Universal Time (UTC)
- Greenwich Mean Time (GMT) – sunce je u podne iznad Greenwich-a
- Pomak istočno od 0-tog meridijana +, a zapadno – (Zagreb: UTC+1; New York: UTC-5)
- Podaci s opcijom WITH TIME ZONE uključuju, zajedno s pohranjenom vrijednošću, eksplicitan pomak (offset) u odnosu na UTC



# PostgreSQL: vremenske zone

- Vremenska zona se u PostgreSQL-u može specificirati na različite načine, pomoću:
  - imena (npr. "Europe/Paris", "Australia/Tasmania",...) ili
  - kratice (npr. CEST, EST)

Vremenska zona koju koristi PostgreSQL na „našem” virtualnom računalu:

```
SELECT current_setting('TIMEZONE')
```

text

Europe/Rome

To je i vremenska zona Zagreba:

Pomoću donjeg upita možemo doznati pomak vremenske zone u odnosu na UTC.

```
SELECT now()::DATE, name, abbrev, utc_offset
FROM pg_timezone_names
WHERE name IN ('Australia/Tasmania',
               'Europe/Rome',
               'Europe/Zagreb')
```

Koliki pomak očekujemo za Zagreb?

Upit izveden dana

**01.10.2015**

now date	name text	abbrev text	utc_offset interval
01.10.2015	Australia/Tasmania	EST	11:00:00
01.10.2015	Europe/Rome	CEST	02:00:00
01.10.2015	Europe/Zagreb	CEST	02:00:00

**10.11.2015**

now date	name text	abbrev text	utc_offset interval
10.11.2015	Australia/Tasmania	EST	11:00:00
10.11.2015	Europe/Rome	CEST	01:00:00
10.11.2015	Europe/Zagreb	CEST	01:00:00

Zbog čega gornji upit vraća različite vrijednosti utc\_offset ako se izvede na različite datume?



# PostgreSQL: vremenske zone

- Interno se podaci WITH TIME ZONE spremaju u UTC ili GMT vremenskoj zoni
- Ako zona nije eksplicitno navedena podrazumijeva se da je podatak u vremenskoj zoni definiranoj parametrom TIMEZONE `SELECT current_setting('TIMEZONE')`

- Provodi se konverzija iz podrazumijevane vremenske zone u UTC vrijeme

text

Europe/Rome

Rezultati upita na dan:

01.10.2015

10.11.2015

1. `SELECT now()`

now  
timestamp(3) with time zone

01.10.2015 15:17:19.592437 CEST

now  
timestamp(3) with time zone

10.11.2015 15:17:19.592437 CEST

2. `SELECT now() AT TIME ZONE 'UTC';`

now  
timestamp(3) without time zone

01.10.2015 13:17:19.592437 CEST

now  
timestamp(3) without time zone

10.11.2015 14:17:19.592437 CEST

3. `SELECT now() AT TIME ZONE 'UTC-5';`

now  
timestamp(3) without time zone

01.10.2015 18:17:19.592437 CEST

now  
timestamp(3) without time zone

10.11.2015 19:17:19.592437 CEST

Zbog čega upiti 2. i 3. vraćaju različite vrijednosti ako se pokrenu na različite datume?

# PostgreSQL: interval

interval [ fields ] [ (p) ]	Vremenski interval	-178000000 godina	+178000000 godina	mikrosekunda
-----------------------------	--------------------	-------------------	-------------------	--------------

Granule od *fields* u *interval* tipu:

	kratica	
• YEAR	Y	• YEAR TO MONTH
• MONTH	M	• DAY TO HOUR
• DAY	D	• DAY TO MINUTE
• HOUR	H	• DAY TO SECOND
• MINUTE	M	• HOUR TO MINUTE
• SECOND	S	• HOUR TO SECOND
		• MINUTE TO SECOND

```
SELECT '1Y'::interval
```

interval

1 year

```
SELECT '1D'::interval
```

interval

1 day

```
SELECT '1M'::interval
```

interval

00:01:00

```
SELECT 'P1-2'::interval
```

interval

1 year 2 mons

Interval tip podatka se može specificirati na više načina:

```
P quantity unit [ quantity unit ... ] [ T [ quantity unit ... ] ]
```

```
P [ years-months-days ] [ T hours:minutes:seconds ]
```

```
SELECT 'P1-2-3T4:55:59'::interval
```

interval

1 year 2 mons 3 days 04:55:59

```
SELECT 'P1Y2M3DT4H55M59S'::interval
```

interval

1 year 2 mons 3 days 04:55:59

```
SELECT 'P1-2-3T4:55:59'::interval +  
      'P1Y2M3DT4H55M59S'::interval
```

interval

2 year 4 mons 6 days 09:51:58

# PostgreSQL: daterange, tsrange, tstzrange

- Range tip odgovara PERIOD-u iz SQL standarda. Koristi se za prezentaciju raspona vrijednosti nekog elementarnog tipa (DATE, TIMESTAMP,... - podtip range tipa).
- Zanimljivi su nam rasponi za tipove *date* i *timestamp* (za *time* range ne postoji).
- Definira se na jedan od sljedećih načina:

```
( lower-bound, upper-bound )  
( lower-bound, upper-bound ]  
[ lower-bound, upper-bound )  
[ lower-bound, upper-bound ]  
empty
```

"[" odnosno "]" - granica uključena  
"(" odnosno ")" - granica isključena

- Granica može biti:
  - Niz znakova – koji predstavlja vrijednost valjanu za podtip
  - Izostavljena – signalizira da nema granice
- Ako za raspon nije navedena
  - donja granica tada ona iznosi "*minus beskonačno*",  
a raspon uključuje sve vrijednosti manje (ili manje ili jednake) od gornje granice
  - Gornja granica tada ona iznosi "*plus beskonačno*",  
a raspon uključuje sve vrijednosti veće od (ili veće ili jednake) donje granice
- `dateRange` tip (uvijek) koristi oblik koji uključuje donju i isključuje gornju granicu: `[]` (tzv. *closed-open* notaciju za `dateRange` ćemo koristiti u predavanjima)

# PostgreSQL: range - primjeri

obavljaFunkciju

```
CREATE TABLE obavljaFunkciju(  
  sifFun      smallint REFERENCES ...,  
  sifOrgJed   smallint REFERENCES ...,  
  sifOsoba    smallint REFERENCES ...,  
  periodOdDo dateRange);
```

sifFun	sifOrgJed	sifOsoba	periodOdDo
10	100	51	[, 1.10.2010)
10	100	50	[1.10.2010, 1.10.2014)
10	100	100	[01.10.2014,)
...	...	...	...

```
INSERT INTO obavljaFunkciju VALUES (10, 100, 51, '[, 1.10.2010)');  
INSERT INTO obavljaFunkciju VALUES (10, 100, 50, '[1.10.2010, 1.10.2014)');  
INSERT INTO obavljaFunkciju VALUES (10, 100, 100, '[01.10.2014,)');
```

```
SELECT periodOdDo FROM obavljaFunkciju
```

periodOdDo daterange
(, 1.10.2010)
[01.10.2010, 1.10.2014)
[01.10.2014,)

Odstupanje od *closed-open* notacije kada je donja granica "minus beskonačno"

```
SELECT '(09.11.2015, 15.11.2015]'::daterange
```

daterange
[10.11.2015,16.11.2015)

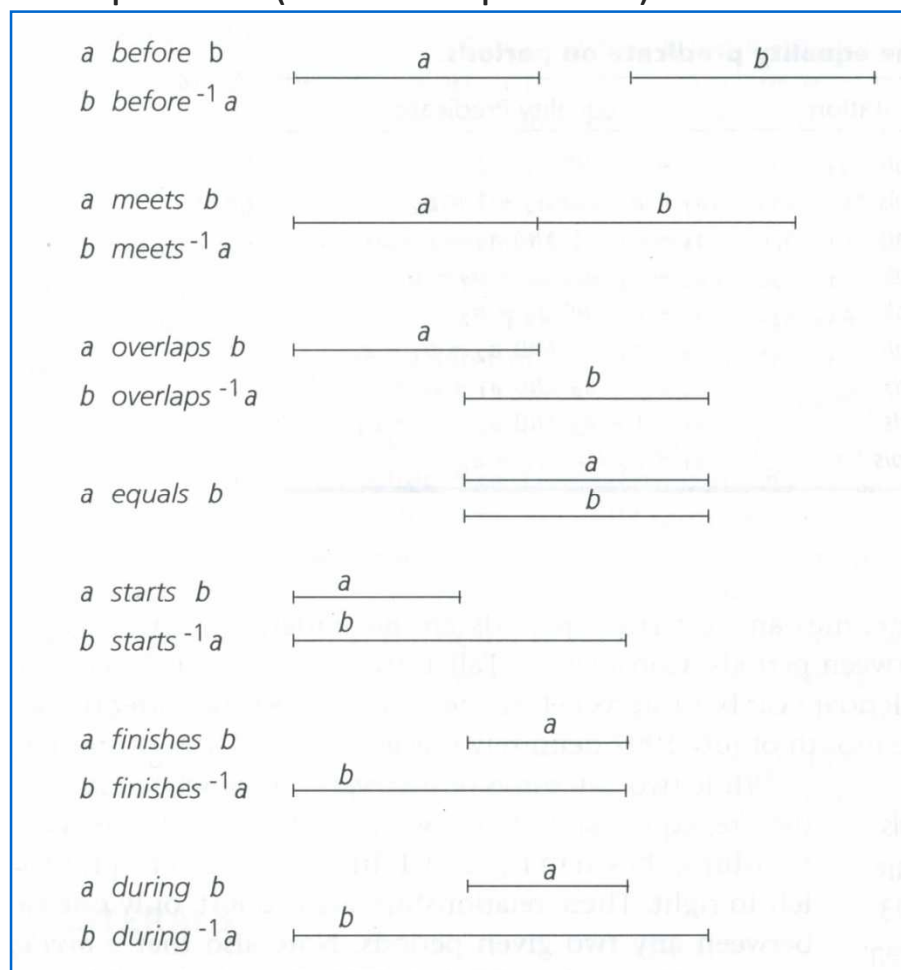
```
SELECT '(2015-11-12 08:15, 2015-11-12 11:00]'::tsrange
```

Za tsRange *closed-open* notacija se ne primjenjuje.

tsrange
("12.11.2015 08:15:00","12.11.2015 11:00:00"]

# Odnosi između perioda

- Operacije definirane SQL standardom za PERIOD podudaraju se u velikoj mjeri s Allenovom klasifikacijom
- Postoji 13 mogućih relacija/predikata između dva perioda (Allenovi operatori)
- Taj skup operacija je:
  - Jasan - za dva perioda samo jedan operator je istinit
  - Iscrpan - za dva perioda uvijek je jedan operator istinit



# Odnosi između perioda

Operator	Opis pomoću kranjih točaka
$I_1$ before $I_2$	$\text{end}(I_1) < \text{begin}(I_2)$
$I_1$ after $I_2$	$\text{End}(I_2) < \text{negin}(I_1)$
$I_1$ during $I_2$	$\text{begin}(I_1) > \text{begin}(I_2) \wedge \text{end}(I_1) \leq \text{end}(I_2) \vee$ $\text{begin}(I_1) \geq \text{begin}(I_2) \wedge \text{end}(I_1) < \text{end}(I_2)$
$I_1$ contains $I_2$	$\text{begin}(I_2) > \text{begin}(I_1) \wedge \text{end}(I_2) \leq \text{end}(I_1) \vee$ $\text{begin}(I_2) \geq \text{begin}(I_1) \wedge \text{end}(I_2) < \text{end}(I_1)$
$I_1$ overlaps $I_2$	$\text{begin}(I_1) < \text{begin}(I_2) \wedge \text{end}(I_1) > \text{begin}(I_2) \wedge$ $\text{end}(I_1) < \text{end}(I_2)$
$I_1$ overlapped_by $I_2$	$\text{begin}(I_2) < \text{begin}(I_1) \wedge \text{end}(I_2) > \text{begin}(I_1) \wedge$ $\text{end}(I_2) < \text{end}(I_1)$
$I_1$ meets $I_2$	$\text{end}(I_1) = \text{begin}(I_2)$
$I_1$ met_by $I_2$	$\text{end}(I_2) = \text{begin}(I_1)$
$I_1$ starts $I_2$	$\text{begin}(I_1) = \text{begin}(I_2) \wedge \text{end}(I_1) < \text{end}(I_2)$
$I_1$ started by $I_2$	$\text{begin}(I_1) = \text{begin}(I_2) \wedge \text{end}(I_2) < \text{end}(I_1)$
$I_1$ finishes $I_2$	$\text{begin}(I_1) > \text{begin}(I_2) \wedge \text{end}(I_1) = \text{end}(I_2)$
$I_1$ finished_by $I_2$	$\text{begin}(I_2) > \text{begin}(I_1) \wedge \text{end}(I_1) = \text{end}(I_2)$
$I_1$ equals $I_2$	$\text{begin}(I_1) = \text{begin}(I_2) \wedge \text{end}(I_1) = \text{end}(I_2)$

## ■ Drugi operatori nad vremenskim podacima

- Zbrajanje perioda
  - Razlika perioda
  - Izdvajanje donje ili gornje granice perioda
  - Pripadnost vremenskog trenutka periodu
  - Odnosi između vremenskih trenutaka (jednako, različito)
- 
- Svi su matematički prilično jednostavni (pogotovo u usporedbi s operacijama nad prostornim podacima)

## PostgreSQL: Operatori nad rasponima (daterange, tsrange, tstzrange)

=	jednakost ( <i>equals</i> )	Jesu li rasponi jednaki, obje granice i svojstva zatvorenosti
<>	različitost ( <i>not equal</i> )	Jesu li rasponi različiti
<, <=	manji od ( <i>less then</i> )	Prvo uspoređuju lijevu granicu i tek ako je ona jednaka uspoređuju desnu, služe za implementaciju B stabala nad rasponima
>, >=	veći od ( <i>greater then</i> )	
@>	sadrži raspon ili element ( <i>contains</i> )	Može se primijeniti na dva raspona ili na raspon i element
<@	sadržan je u	
&&	preklapanje ( <i>overlap</i> )	Imaju li dva raspona zajedničkih točaka
<<, >>	strogo lijevo/desno ( <i>strictly left/right</i> )	
&<, &>	ne prostire se desno/lijevo od ( <i>does not extend to the right/left of</i> )	
- -	susjedan ( <i>is adjacent to</i> )	Jesu li rasponi susjedni, zanemaruje zatvorenost

SQL standard: CONTAINS @>, <@), OVERLAPS (&&), EQUALS (=), PRECEDES (<<), SUCCEEDS (>>), IMMEDIATELY PRECEDES, i IMMEDIATELY SUCCEEDS



## PostgreSQL: Operatori nad rasponima (daterange, tsrange, tstzrange)

+	unija	Operacije unije i razlike će prijaviti pogrešku ako bi rezultat tih operacija trebao sadržavati dva disjunktna raspona.
*	presjek	
-	razlika	

Operatori <<, >>, &<, &>, -|- uvijek vraćaju vrijednost false ako je jedan od operanada prazni raspon

- Funkcije nad rasponima :

Lower	Donja granica raspona
upper	Gornja granica raspona
isempty	Je li raspon prazan
lower_inc	Je li donja granica zatvorena ( <i>inclusive</i> )
upper_inc	Je li gornja granica zatvorena ( <i>inclusive</i> )
lower_inf	Je li donja granica beskonačna ( <i>infinite</i> )
upper_inf	Je li gornja granica beskonačna ( <i>infinite</i> )

# PostgreSQL: Operatori nad rasponima - primjer

rezDvorana			prisustvo		
oznDvorana	...	terminOdDo	oznDvorana	sifStudent	terminOdDo
B5	...	[2015-11-12 08:00, 2015-11-12 11:00)	B5	23459789	[2015-11-12 08:30, 2015-11-12 11:00)
B5	...	[2015-11-12 11:00, 2015-11-12 13:00)	B5	20006789	[2015-11-12 08:31, 2015-11-12 10:59)
...	...		...		

Ispiši prezimena i imena studenata koji su boravili u dvorani B5 12.11.2015 između 8:30 i 11:00.  
Smatrati da je student boravio u dvorani i ako nije došao na početku i ako je otišao prije kraja predavanja.

student		
sifStudent	ime	prezime

U upitu treba dovesti u vezu `prisustvo.terminOdDo` i `rezDvorana.terminOdDo`.

Operatori koji možda dolaze u obzir:

1. `@>` sadrži raspon ili element

```
prisustvo.terminOdDo @> rezDvorana.terminOdDo
```

2. `<@` sadržan je u

```
prisustvo.terminOdDo <@ rezDvorana.terminOdDo
```

3. `&&` Imaju li dva raspona zajedničkih točaka

```
prisustvo.terminOdDo && rezDvorana.terminOdDo
```

# PostgreSQL: Operatori nad rasponima - primjer

1. @> sadrži raspon ili element

```
prisustvo.terminOdDo @> rezDvorana.terminOdDo
```

2. <@ sadržan je u

```
prisustvo.terminOdDo <@ rezDvorana.terminOdDo
```

3. && Imaju li dva raspona zajedničkih točaka

```
prisustvo.terminOdDo && rezDvorana.terminOdDo
```

1. Uvjet u rezultatu neće imati studente koji nisu boravili u dvorani čitav period *rezDvorana.terminOdDo* - npr. ušli su u dvoranu X minuta nakon početka ili su izašli iz dvorane X minuta prije kraja.
2. Uvjet u rezultatu neće imati studente koji su u dvorani boravili dulje od perioda *rezDvorana.terminOdDo* - npr. ušli su u dvoranu X minuta prije početka ili su izašli iz dvorane X minuta nakon kraja
3. Uvjet će uzeti u obzir sve studente koji su boravili u dvorani u periodu koji ima zajedničkih "točaka" s periodom *rezDvorana.terminOdDo*.

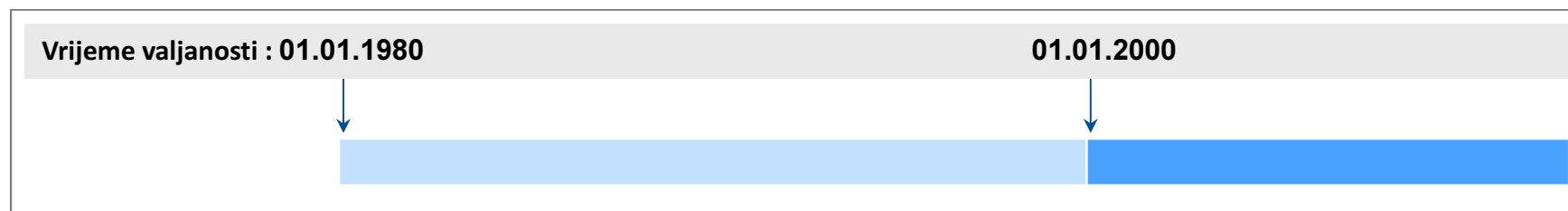
```
SELECT student.JMBAG, ime, prezime
FROM prisustvo, rezDvorana, student
WHERE prisustvo.oznDvorana = rezDvorana.oznDvorana
AND prisustvo.JMBAG = student.JMBAG
AND rezDvorana.terminOdDo = '[2015-11-12 08:15.000, 2015-11-12 11:00.000]'
AND prisustvo.terminOdDo && rezDvorana.terminOdDo
```

## Vremenske dimenzije

- U kontekstu temporalnih baza podataka značajne su dvije ortogonalne (međusobno nezavisne) dimenzije vremena
  - Vrijeme valjanosti (***valid time***)  
Vrijeme u stvarnom svijetu kada se neki događaj dogodio ili period u kojem neka činjenica važeća, nezavisno od trenutka kada je informacija o tom događaju/činjenici zapisana u bazu podataka
  - Transakcijsko vrijeme (***transaction time ili system time***)  
Vrijeme kada je određena promjena zabilježena u bazi podataka ili vremenski interval tijekom kojeg se baza podataka nalazi u određenom stanju
- SQL:2011 standard:
  - vrijeme valjanosti je podržano relacijama koje sadrže *application-time* period
  - Transakcijsko vrijeme je podržano relacijama koje sadrže *system-time* period, a nazivaju se *system-versioned* relacijama
  - predviđa samo jedan *application-time* i samo jedan *system-versioned* period po relaciji

## Vrijeme valjanosti - primjer

	Datum	Što se dogodilo u stvarnom svijetu	Akcija u bazi podataka	Stanje u bazi podataka
1	01.01.1980	U Supetru je rođena Ana Par	nema	nema osobe Ana Par
	01.02.1980	Roditelji su prijavili rođenje Ane Par	INSERT	Ana Par prebiva u Supetru
2	01.01.2000	Ana Par se preselila iz Supetra u Zagreb, ali nije prijavila MUP-u	nema	Ana Par prebiva u Supetru
	02.01.2000	Ana Par je prijavila novo prebivalište MUP-u	INSERT	Ana Par prebiva u Zagrebu



- Vrijeme valjanosti možemo izraziti pomoću perioda pocetakVV-krajVV

1	...	ime	prezime	postOznPreb	pocetakVV	krajVV
		Ana	Par	21400	01.01.1980	$\infty$

2	...	ime	prezime	postOznPreb	pocetakVV	krajVV
		Ana	Par	21400	01.01.1980	01.01.2000
		Ana	Par	10000	01.01.2000	$\infty$

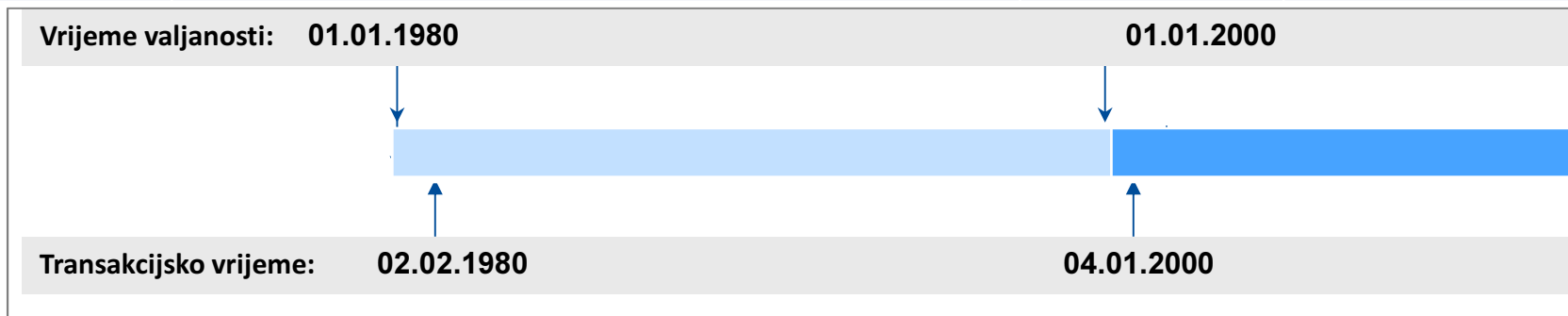
Što je ključ n-torke?  
Može li biti npr. OIB?

Closed-open notacija

Razmislite: Bi li bilo prikladnije n-torke koje nisu valjane u trenutku "sada" izdvojiti u zasebnu relaciju?

## Transakcijsko vrijeme - primjer

	Datum	Što se dogodilo u stvarnom svijetu	Akcija u bazi podataka	Stanje u bazi podataka
	01.01.1980	U Supetru je rođena Ana Par	nema	nema osobe Ana Par
	01.02.1980	Roditelji su prijavili rođenje Ane Par	nema	nema osobe Ana Par
1	02.02.1980	Administrator je unio podatke o Ani Par u bazu podataka	INSERT	Ana Par prebiva u Supetru
	01.01.2000	Ana Par se preselila iz Supetra u Zagreba, ali nije prijavila MUP-u	nema	Ana Par prebiva u Supetru
	02.01.2000	Ana Par je prijavila novo prebivalište MUP-u	nema	Ana Par prebiva u Supetru
2	04.01.2000	Administrator je unio podatke o Ani Par u bazu podataka	INSERT	Ana Par prebiva u Zagrebu



- Transakcijsko vrijeme možemo izraziti pomoću perioda pocetakVT - krajVT

1	...	ime	prezime	postOznPreb	pocetakVT	krajVT
		Ana	Par	21400	02.02.1980	∞

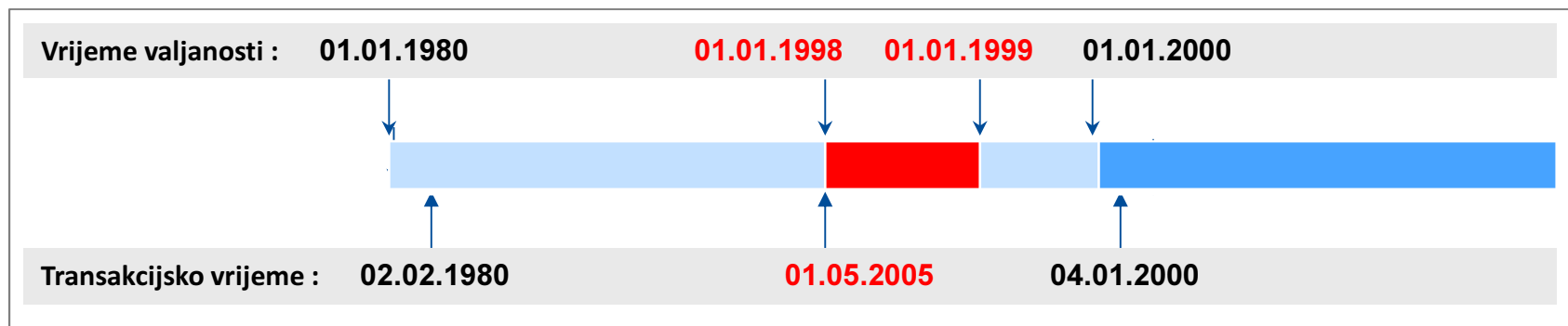
Što je ključ n-torke?

2	...	ime	prezime	postOznPreb	pocetakVT	krajVT
		Ana	Par	21400	02.02.1980	04.01.2000
		Ana	Par	10000	04.01.2000	∞

Što ako želimo pratiti i vrijeme valjanosti i transakcijsko vrijeme?

## Vrijeme valjanosti - primjer

- Porezna uprava je 01.01.2005 otkrila da je Ana Par u periodu 01.01.1998 - 01.01.1999 prebivala u Splitu, ali to nije prijavila (da bi izbjegla plaćanje prireza - Supetar 0%, Split 10%)



...	ime	prezime	postOznPreb	pocetakVV	krajVV
	Ana	Par	21400	01.01.1980	01.01.1998
	Ana	Par	21000	01.01.1998	01.01.1999
	Ana	Par	21400	01.01.1999	01.01.2000
	Ana	Par	10000	01.01.2000	∞

Boravak u Splitu 01.01.1998-01.01.1999

Boravak u Supetru 01.01.1999-01.01.2000

- Nemamo informaciju da je sve do 01.01.2005 "baza podataka tvrdila" da Ana Par u periodu 01.01.1998 - 01.01.1998 prebiva u Supetru
- Ta je informacija važna kao dokaz porezne prevare i kao opravdanje izračuna poreza za Anu Par prema podacima koji nisu odgovarali istini

# Vrijeme valjanosti i transakcijsko vrijeme - primjer

	Datum	Što se dogodilo u stvarnom svijetu	Akcija u BP	Stanje u bazi podataka
1	01.01.1980	U Supetru je rođena Ana Par	nema	nema osobe Ana Par
	01.02.1980	Roditelji su prijavili rođenje Ane Par	nema	nema osobe Ana Par
2	02.02.1980	Administrator je unio podatke o Ani Par u bazu podataka	INSERT	Ana Par prebiva u Supetru
3	01.01.2000	Ana Par se preselila iz Supetra u Zagreba, ali nije prijavila MUP-u	nema	Ana Par prebiva u Supetru
	02.01.2000	Ana Par je prijavila novo prebivalište MUP-u	nema	Ana Par prebiva u Supetru
4	04.01.2000	Administrator je unio podatke o Ani Par u bazu podataka	INSERT	Ana Par prebiva u Zagrebu
5	01.01.2005	Porezna uprava je otkrila da je Ana Par u periodu 01.01.1998 - 31.12.1998 prebivala u Splitu		

1+2	...	ime	prezime	postOznPreb	pocetakVV	krajVV	pocetakVT	krajVT
		Ana	Par	21400	01.01.1980	∞	02.02.1980	∞

3+4	...	ime	prezime	postOznPreb	pocetakVV	krajVV	pocetakVT	krajVT
		Ana	Par	21400	01.01.1980	∞	02.02.1980	04.01.2000
		Ana	Par	21400	01.01.1980	01.01.2000	04.01.2000	∞
		Ana	Par	10000	01.01.2000	∞	04.01.2000	∞



# Vrijeme valjanosti i transakcijsko vrijeme - primjer

	Datum	Što se dogodilo u stvarnom svijetu	Akcija u BP	Stanje u bazi podataka
1	01.01.1980	U Supetru je rođena Ana Par	nema	nema osobe Ana Par
	01.02.1980	Roditelji su prijavili rođenje Ane Par	nema	nema osobe Ana Par
2	02.02.1980	Administrator je unio podatke o Ani Par u bazu podataka	INSERT	Ana Par prebiva u Supetru
3	01.01.2000	Ana Par se preselila iz Supetra u Zagreba, ali nije prijavila MUP-u	nema	Ana Par prebiva u Supetru
	02.01.2000	Ana Par je prijavila novo prebivalište MUP-u	nema	Ana Par prebiva u Supetru
4	04.01.2000	Administrator je unio podatke o Ani Par u bazu podataka	INSERT	Ana Par prebiva u Zagrebu
5	01.01.2005	Porezna uprava je otkrila da je Ana Par u periodu 01.01.1998 - 31.12.1998 prebivala u Splitu		

3+4	...	ime	prezime	postOznPreb	pocetakVV	krajVV	pocetakVT	krajVT
		Ana	Par	21400	01.01.1980	∞	02.02.1980	04.01.2000
		Ana	Par	21400	01.01.1980	01.01.2000	04.01.2000	∞
		Ana	Par	10000	01.01.2000	∞	04.01.2000	∞



5	...	ime	prezime	postOznPreb	pocetakVV	krajVV	pocetakVT	krajVT
		Ana	Par	21400	01.01.1980	∞	02.02.1980	04.01.2000
		Ana	Par	21400	01.01.1980	01.01.2000	04.01.2000	01.01.2005
		Ana	Par	10000	01.01.2000	∞	04.01.2000	∞
		Ana	Par	21400	01.01.1980	01.01.1998	01.01.2005	∞
		Ana	Par	21000	01.01.1998	01.01.1999	01.01.2005	∞
		Ana	Par	21400	01.01.1999	01.01.2000	01.01.2005	∞

## Temporalne relacije

- S obzirom na sposobnost upravljanja vremenom valjanosti i transakcijskim vremenom, razlikujemo četiri vrste relacija
  - Trenutačne relacije (*snapshot tables*)
  - Relacije vremena valjanosti (*aplication-time period tables*)
  - Relacije transakcijskog vremena (*system-versioned tables*)
  - Bitemporalne relacije (*system-versioned aplication-time period tables*)
- Semantika vremena valjanosti i transakcijskog vremena je različita - ne *ponašaju* se jednako
- PostgreSQL ne nudi izravnu podršku za upravljanje vremenom valjanosti niti transakcijskim vremenom

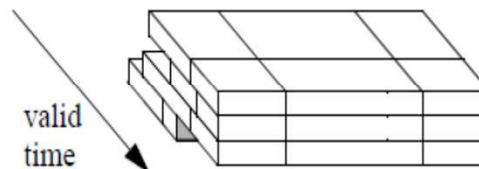
Link za dodatna pojašnjenja: [Temporal Features in SQL standard](#)

# Temporalne relacije

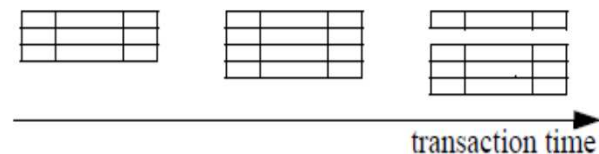
## Trenutačne relacije (snapshot)



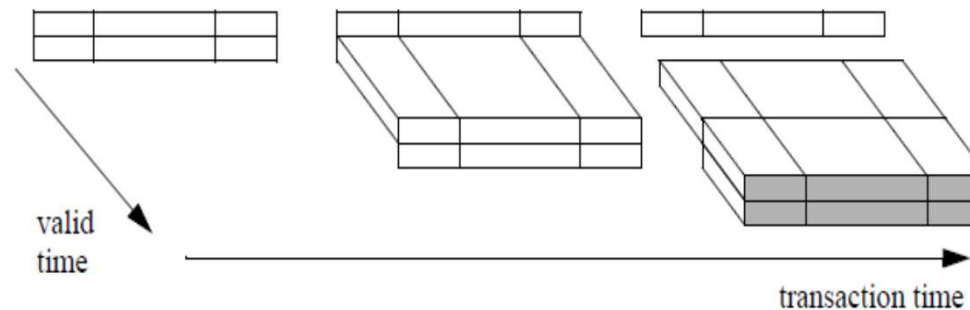
## Relacije vremena valjanosti



## Relacije transakcijskog vremena

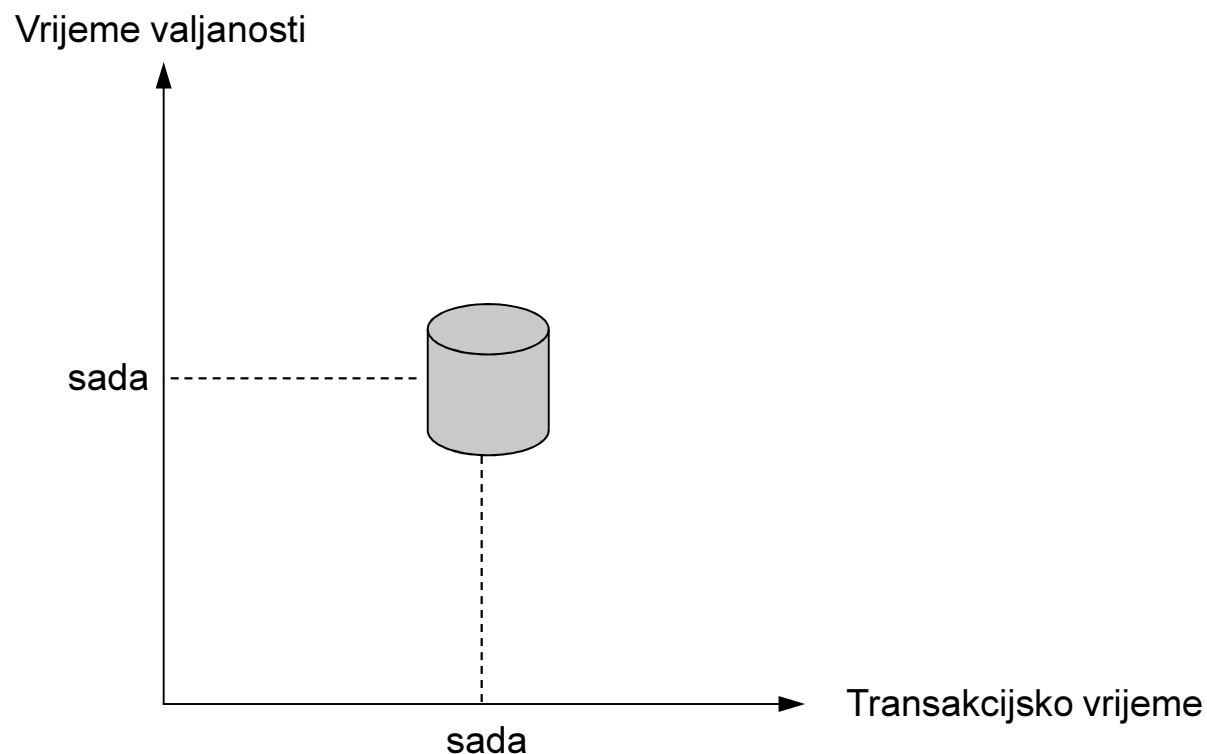


## Bitemporalne relacije



# Trenutačne relacije

- Trenutačne baze podataka (engl. *snapshot database*)
  - Opisuju jedan trenutak u stvarnom svijetu, najčešće sadašnjost
  - Stanje baze mijenja se operacijama INSERT, UPDATE i DELETE
  - Staro stanje baze se zaboravlja

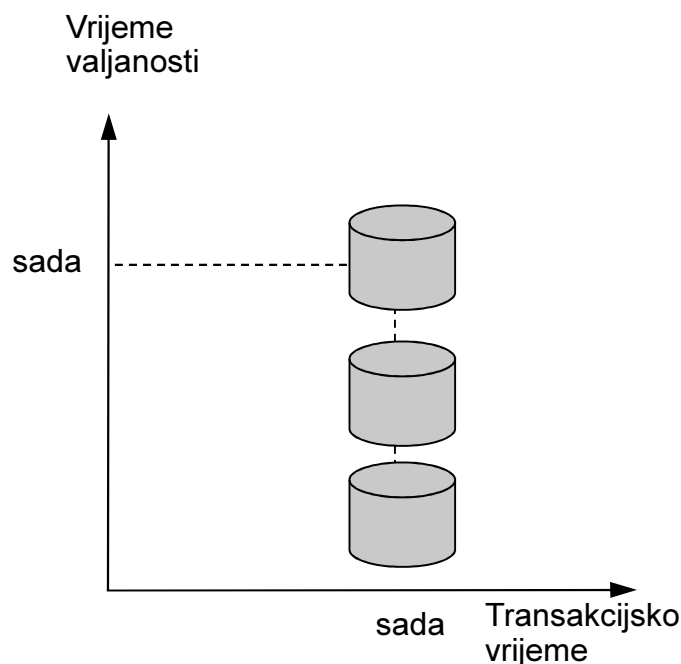


## Relacije vremena valjanosti (*application-time period tables*)

- Pohranjuju povijest podataka u stvarnom svijetu, tj. pohranjuju stanje baze podataka duž osi vremena valjanosti

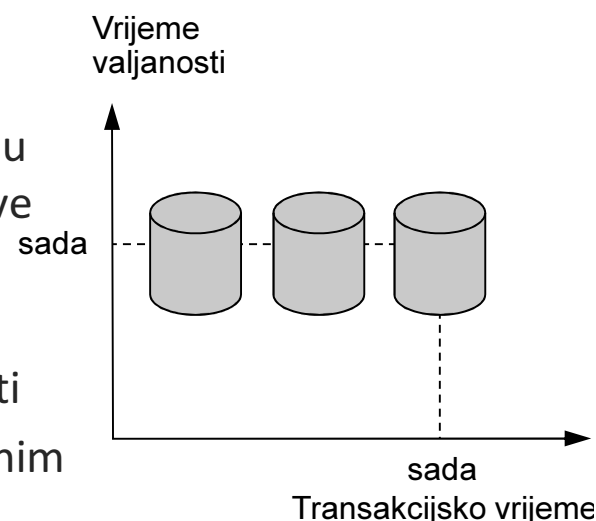
SQL:2011 za ove relacije predviđa:

- Sadrže dva dodatna atributa – početak i kraj perioda valjanosti
- Sadrže period valjanosti kao metapodatak definiran početkom i krajem
- Korisnik definira početak i kraj vremena valjanosti
- sintaksu za
  - definiranje **primarnog** ključa koja osigurava da se periodi dvije n-torke s različitim ključevima ne preklapaju
  - definiranje **referencijskog** integriteta koji osigurava da je period referencirajuće n-torke u potpunosti sadržan u periodu jedne referencirane n-torke ili u kombiniranom periodu dvije ili više uzastopnih referenciranih n-torki
  - UPDATE i DELETE naredbe za izmjenu ili brisanje n-torke u parcijalnom periodu



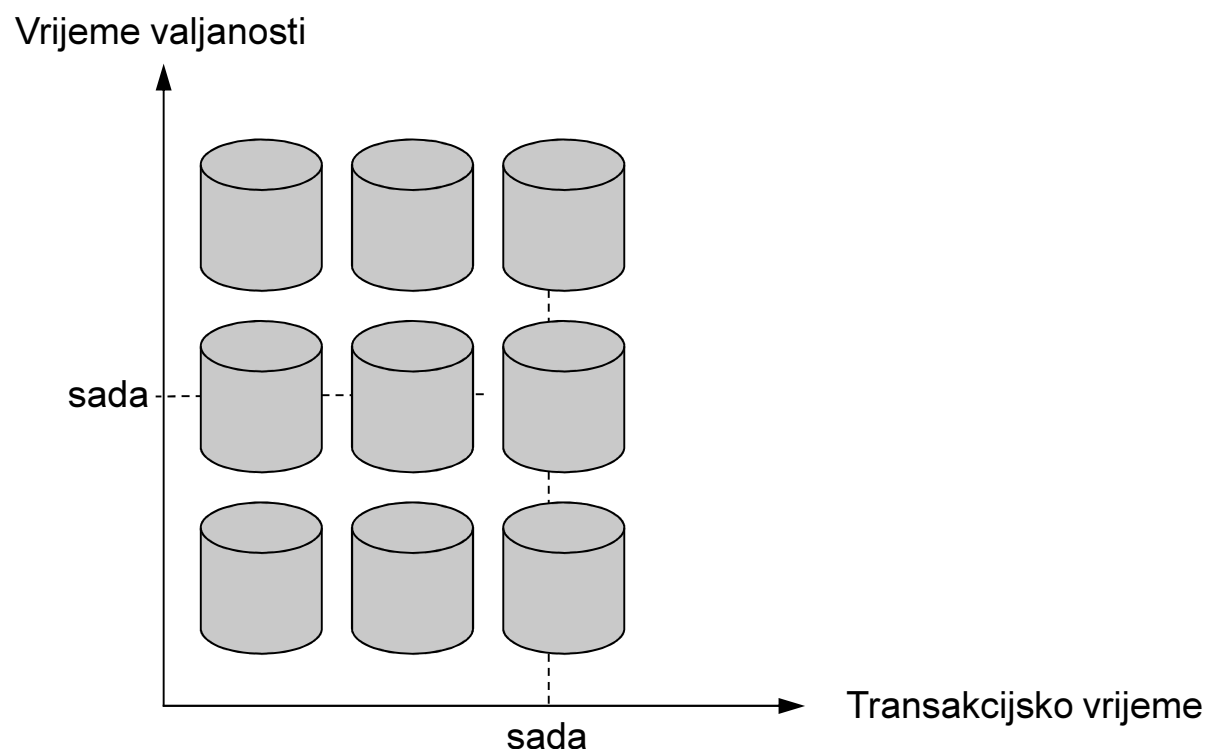
## Relacije transakcijskog vremena (*system-versioned tables*)

- Pohranjuju izmjene u bazi podataka tj. stanje baze podataka duž osi transakcijskog vremena
- Moguće je reproducirati stanje baze (ne nužno stanje stvarnog svijeta) iz bilo kojeg trenutka u prošlosti
- SQL:2011 predviđa:
  - relacija sadrži period, standardom definiranog imena (SYSTEM\_TIME) čija vrijednost se održava navođenjem sintakse WITH SYSTEM VERSIONING
  - relacija sadrži dva dodatna atributa – početak i kraj SYSTEM\_TIME perioda
  - vrijednosti oba atributa održava SUBP, korisnicima nije dozvoljeno ažuriranje tih vrijednosti
- stare vrijednosti izmijenjenih n-torki se čuvaju
- n-torke čiji period ima presjek s trenutkom sada se nazivaju trenutnim sistemskim n-torkama (current system rows), sve ostale se nazivaju povijesnim sistemskim n-torkama (historical system rows)
- samo trenutne sistemske n-torke se smiju mijenjati i brisati
- sva integritetska ograničenja se provode samo nad trenutnim sistemskim n-torkama



# Bitemporalne relacije (system-versioned application-time period tables)

- Poshranjuju stanje baze podataka duž obje vremenske osi
- Podržavaju svojstva i relacija vremena valjanosti i relacija transakcijskog vremena



Linkovi za dodatna pojašnjenja: [The Case for Bitemporal Data 1/7](#) , [2/7](#) , [3/7](#) ,...

## Ograničenja integriteta: Temporalni primarni ključ

obavljaFunkciju				
sifFun	sifOrgJed	sifOsoba	datumOd	datumDo
10	100	51	1.10.2006	01.10.2010
10	100	50	1.10.2010	01.10.2014
10	100	100	1.10.2014	01.10.2016
10	100	80	1.10.2012	01.10.2013
...	...	...	...	...

Niti jedna od donjih definicija ključa ne sprječava ovo:  
Dekan u intervalu od 1.10.2012 do 30.09.2013.  
nije jedinstveno definiran.

Što bi bio dobar ključ?

1. {sifFun, sifOrgJed, datumOd}
2. {sifFun, sifOrgJed, datumDo}
3. {sifFun, sifOrgJed, datumOd, datumDo}

Dodatno, datumDo ne mora za svaku funkciju biti poznat (npr. Studentski savjetnik), a entitetski integritet ne dozvoljava NULL vrijednost niti u jednom atributu ključa

Ispravan ključ uključuje attribute

- sifFun i sifOrgjed za koje će se jedinstvenost ispitivati pomoću operatora =  
i
- vremenski atribut tipa vremenskiPeriod za koji će se jedinstvenost ispitivati pomoću operatora za rad s period tipom (spriječiti preklapanje, sadržavanje, postojanje presjeka,...)



# Ograničenja integriteta: Temporalni strani ključ

## OrgJed

			orgJedPeriod	
sifOrgJed	nazOrgJed	...	datumOd	datumDo
100	ETF	...	1.10.1926	29.06.1995
100	FER	...	29.06.1995	31.12.2999
...	...		...	

$K_{ORGJED} = \{sifOrgJed, orgJedPeriod\}$

## obavljaFunkciju

			obFunPeriod	
sifFun	sifOrgJed	sifOsoba	datumOd	datumDo
10	100	12	1.10.1990	01.10.1992
10	100	50	1.10.2010	01.10.2014
...	...	...	...	

$K_{OBAVLJAFUNKCIJU} = \{sifFun, sifOrgJed, obFunPeriod\}$

$FK_{OBFUN\_ORGJED} = \{sifOrgJed, obFunPeriod\}$

- Strani ključ koji uključuje atribut nekog od *period* tipova podataka podrazumijeva:
  - Referencirana relacija mora imati ključ s atributom tipa period
  - Period referencirajuće n-torke mora biti u potpunosti sadržan u periodu jedne referencirane n-torke ili u kombiniranom periodu dvije ili više uzastopnih referenciranih n-torki
- Prva n-torka relacije *obavljaFunkciju* referencira se na prvu n-torku relacije *orgJed*, a druga n-torka relacije *obavljaFunkciju* na drugu n-torku relacije *orgJed*
- Ograničenje stranog ključa  $FK_{OBFUN\_ORGJED}$  trebalo bi provjeravati sljedećim uvjetima:

`obavljaFunkciju.sifOrgJed = orgJed.sifOrgJed AND  
obFunPeriod jeSadržanU jednom ili više uzastopnih orgJedPeriod`

# SQL standard: Relacije vremena valjanosti

OrgJed

sifOrgJed	nazOrgJed	,,,	datumOd	datumDo
100	ETF	...	1.10.1926	29.06.1995
100	FER	...	29.06.1995	31.12.2999
...	...	...	...	...

obavljaFunkciju

sifFun	sifOrgJed	sifOsoba	datumOd	datumDo
10	100	12	1.10.1990	01.10.1992
10	100	50	1.10.2010	01.10.2014
...	...	...	...	...

```
CREATE TABLE orgJed
(sifOrgJed    INTEGER,
 nazOrgJed    VARCHAR(50) NOT NULL,
 . . .
 datumOd      DATE NOT NULL,
 datumDo      DATE NOT NULL,
 PERIOD FOR orgJedPeriod (datumOd, datumDo),
 PRIMARY KEY (sifOrgJed, orgJedPeriod WITHOUT OVERLAPS));
```

Početak i kraj perioda

- Određuje korisnik
- može biti u prošlosti, sadašnjosti ili budućnosti

```
CREATE TABLE obavljaFunkciju
(sifFun        INTEGER,
 sifOrgJed     INTEGER,
 . . .
 datumOd       DATE NOT NULL,
 datumDo       DATE NOT NULL,
 PERIOD FOR obFunPeriod (startDate, endDate),
 PRIMARY KEY (sifFun, sifOrgJed, obFunPeriod WITHOUT OVERLAPS),
 FOREIGN KEY (sifOrgJed, PERIOD obFunPeriod) REFERENCES
 orgJed (sifOrgJed, PERIOD orgJedPeriod));
```

# PostgreSQL: Relacije vremena valjanosti

OrgJed

sifOrgJed	nazOrgJed	...	orgJedPeriod
100	ETF	...	[1.10.1926, 29.06.1995)
100	FER	...	[29.06.1995, 31.12.2999)
...	...	...	...

obavljaFunkciju

sifFun	sifOrgJed	sifOsoba	obFunPeriod
10	100	12	[1.10.1990, 01.10.1992)
10	100	50	[1.10.2010, 01.10.2014)
...	...	...	...

```
CREATE EXTENSION bTree_gist;
CREATE TABLE orgJed
(sifOrgJed      INTEGER,
 nazOrgJed      VARCHAR(50) NOT NULL,
 . . .
 orgJedPeriod DATERANGE,
 PRIMARY KEY (sifOrgJed, orgJedPeriod),
 CONSTRAINT pkOrgJed EXCLUDE USING gist
    (sifOrgJed WITH =, orgJedPeriod WITH &&)
)
```

```
CREATE table obavljaFunkciju (
  sifFun          SMALLINT CONSTRAINT fkOFFun      REFERENCES fun(sifFun),
  sifOrgJed       SMALLINT CONSTRAINT fkOFOrgJed REFERENCES orgJed(sifOrgJed),
  sifOsoba        SMALLINT CONSTRAINT fkOFOsoba  REFERENCES osoba(sifOsoba),
  obFunPeriod     DATERANGE,
 PRIMARY KEY (sifFun, sifOrgJed, obFunPeriod),
 CONSTRAINT pkObavljaFunkciju EXCLUDE USING gist
    (sifFun WITH =, sifOrgJed WITH =, obFunperiod WITH &&)
);
```

# PostgreSQL: Temporalni primarni ključ

obavljaFunkciju	sifFun	sifOrgJed	sifOsoba	obFunPeriod
	10	100	51	[01.10.2006, 01.10.2010)
	10	100	50	[01.10.2010, 01.10.2014)
	10	100	100	[01.10.2014, 01.10.2016)
	...	...	...	...

Implementacija perioda u PostgreSQL-u:

- dateRange
- tsRange
- tstzRange

PK = {*sifFun*, *sifOrgJed*, *obFunPeriod*}

- Može se implementirati pomoću tzv. ograničenja isključivanja (EXCLUDE CONSTRAINT)
- Iskazuje se deklarativno pri definiciji ograničenja

```
CONSTRAINT constraint_name
EXCLUDE [ USING index_method ] ( exclude_element WITH operator [, ... ] )
index_parameters [ WHERE ( predicate ) ] |
```

- Za period tip podatka EXCLUDE prihvaća

komutativne operatore: && i -|- (preklapaju li se, jesu li susjedni)

ne prihvaća nekomutativne : @>, <@, <<, >>, &<, &>

- Prednosti deklarativnog načina očuvanja integriteta u odnosu na implementaciju pomoću okidača:
  - jednostavnost,
  - manja podložnost greškama,
  - bolje performance

Link za dodatna pojašnjenja: <http://www.postgresql.org/docs/9.4/static/sql-createtable.html#SQL-CREATETABLE-EXCLUDE>

# PostgreSQL: Temporalni primarni ključ

obavljaFunkciju	sifFun	sifOrgJed	sifOsoba	obFunPeriod
	10	100	51	[01.10.2006, 01.10.2010)
	10	100	50	[01.10.2010, 01.10.2014)
	10	100	100	[01.10.2012, 01.10.2014)
...	...	...	...	...

```
INSERT INTO obavljaFunkciju VALUES (10, 100, 51, '[01.10.2006, 01.10.2010)');
```



```
INSERT INTO obavljaFunkciju VALUES (10, 100, 50, '[01.10.2010, 01.10.2014)');
```



```
INSERT INTO obavljaFunkciju VALUES (10, 100, 100, '[01.10.2012, 01.10.20114)');
```



ERROR: conflicting key value violates exclusion constraint "pkobavljafunkciju"

DETAIL: Key (siffun, siforgjed, obfunperiod)=(10, 100, [2012-10-01,2014-10-01)) conflicts with existing key (siffun, siforgjed, obfunperiod)=(10, 100, [2010-10-01,2014-10-01)).

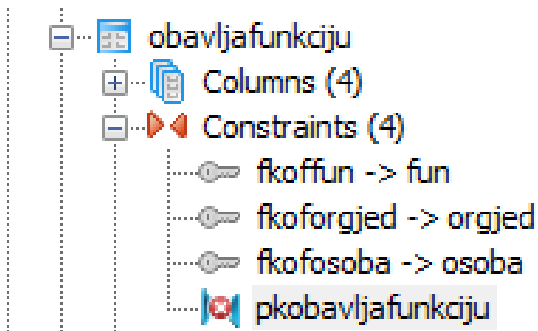
# PostgreSQL: Temporalni primarni ključ

```
...  
CONSTRAINT pkObavljaFunkciju EXCLUDE USING gist  
    (sifFun WITH =, sifOrgJed WITH =, obFunPeriod WITH &&)  
...
```

```
INSERT INTO obavljaFunkciju VALUES(10, 100, 100, '[1.10.2012, 01.10.2013)');
```

Pri provjeri integriteta ključa provjerava se postoji li n-torka

- koja ima jednaku vrijednost atributa sifFun i sifOrgjed (operator = ) i
- s kojom se preklapa vrijednost atributa obFunPeriod (operator &&)
- Za operator && se koristi gist indeks



Property	Value
Name	pkobavljafunkciju
OID	17158
Index OID	17157
Tablespace	pg_default
Columns	siffun WITH =, siforgjed WITH =, periododdo WITH &&
Unique?	No
Primary?	No
Clustered?	No
Valid?	Yes
Access method	gist

# PostgreSQL: Temporalni strani ključ

```
CREATE EXTENSION bTree_gist;
CREATE TABLE orgJed
(sifOrgJed    INTEGER,
 nazOrgJed    VARCHAR(50) NOT NULL,
 . . .
 orgJedPeriod DATERANGE,
 PRIMARY KEY (sifOrgJed, orgJedPeriod),
 CONSTRAINT pkOrgJed EXCLUDE USING gist
    (sifOrgJed WITH =, orgJedPeriod WITH &&)
)
```

```
CREATE table obavljaFunkciju (
  sifFun        SMALLINT CONSTRAINT fkOFFun    REFERENCES fun(sifFun) ,
  sifOrgJed      SMALLINT CONSTRAINT fkOFOrgJed REFERENCES orgJed(sifOrgJed) ,
  sifOsoba       SMALLINT CONSTRAINT fkOFOsoba  REFERENCES osoba(sifOsoba) ,
  obFunPeriod    DATERANGE,
  PRIMARY KEY (sifFun, sifOrgJed, obFunPeriod) ,
  CONSTRAINT pkObavljaFunkciju EXCLUDE USING gist
    (sifFun WITH =, sifOrgJed WITH =, obFunperiod WITH &&) ,
  FOREIGN KEY (sifOrgJed, obFunPeriod) REFERENCES
  orgJed (sifOrgJed, orgJedPeriod)
  USING gist (oznDvorana WITH =, terminOdDo WITH &&)
);
```

Nije implementirano

Moglo bi se implementirati pomoću okidača i procedura.

# SQL standard: Relacije vremena valjanosti - UPDATE

- Predviđena (specijalna) sintaksa koja omogućava specificiranje perioda na kojeg se UPDATE odnosi.
- UPDATE će zahvatiti samo n-torke koje se nalaze unutar specificiranog perioda.
- UPDATE može uzrokovati razdvajanje perioda – INSERT do dvije n-torke koje se odnose na periode izvan specificiranog perioda.

				<u>osobaPeriod</u>		osoba
<u>sifOsoba</u>	ime	prezime	pbrPreb	datumOd	datumDo	
100	Ana	Par	21400	01.01.1980	01.01.2000	←
100	Ana	Par	10000	01.01.2000	31.12.9999	

```
UPDATE osoba FOR PORTION OF osobaPeriod FROM
DATE '01.01.1998' TO DATE '01.01.1999'
SET pbrPreb = '21000'
WHERE sifOsoba = 100
```

				<u>osobaPeriod</u>	
<u>sifOsoba</u>	ime	prezime	pbrPreb	datumOd	datumDo
100	Ana	Par	21400	01.01.1980	01.01.1998
100	Ana	Par	21000	01.01.1998	01.01.1999
100	Ana	Par	21400	01.01.1999	01.01.2000
100	Ana	Par	10000	01.01.2000	31.12.9999

Razdvajanje perioda –  
1 UPDATE+ 2 INSERT-a



# SQL standard: Relacije vremena valjanosti - DELETE

- Predviđena (specijalna) sintaksa koja omogućava specificiranje perioda na kojeg se DELETE odnosi.
- DELETE će zahvatiti samo n-torke koje se nalaze unutar specificiranog perioda.
- DELETE može uzrokovati razdvajanje perioda – INSERT do dvije n-torke koje se odnose na periode izvan specificiranog perioda.

				<u>osobaPeriod</u>		osoba
<u>sifOsoba</u>	ime	prezime	pbrPreb	datumOd	datumDo	
100	Ana	Par	21400	01.01.1980	01.01.2000	
100	Ana	Par	10000	01.01.2000	31.12.9999	

```
DELETE FROM osoba FOR PORTION OF osobaPeriod FROM
DATE '01.01.1998' TO DATE '01.01.1999'
WHERE sifOsoba = 100
```

				<u>osobaPeriod</u>	
<u>sifOsoba</u>	ime	prezime	pbrPreb	datumOd	datumDo
100	Ana	Par	21400	01.01.1980	01.01.1998
100	Ana	Par	21400	01.01.1999	01.01.2000
100	Ana	Par	10000	01.01.2000	31.12.9999

Razdvajanje perioda –  
1 DELETE + 2 INSERT-a

- PostgreSQL nema podršku za automatsko razdvajanje perioda kod UPDATE i DELETE naredbe za dio perioda.

# Prednosti relacija vremena valjanosti

- Većina poslovnih podataka je podložna promjenama tj. zahtijeva praćenje perioda u kojem je određena vrijednost valjana
- Današnji SUBP uglavnom ne pružaju potporu za:
  - Definiranje perioda valjanosti za n-torku
  - Definiranje ograničenja tipa „osoba može prebivati samo u jednom mjestu u određenom vremenskom periodu”
  - UPDATE/DELETE n-torke za dio perioda valjanosti
- Trenutno, najčešće aplikacije pružaju podršku za gornje zahtjeve
- Glavni problemi:
  - Kompleksnost programskog koda
  - Loše performance
- Relacije vremena valjanosti omogućavaju:
  - Pojednostavljenje programskog koda
  - Poboljšane performance
  - Transparentne su u odnosu na naslijeđene aplikacije

# SQL standard: Relacije transakcijskog vremena

```
CREATE TABLE osoba
(sifOsoba      INTEGER PRIMARY KEY,
. . . .
postOznPreb   VARCHAR(15) REFERENCES mjesto (postOzn),
system_start  TIMESTAMP(6) GENERATED ALWAYS AS ROW START,
system_end    TIMESTAMP(6) GENERATED ALWAYS AS ROW END,
PERIOD FOR    SYSTEM_TIME (system_start, system_end),
) WITH SYSTEM VERSIONING;
```

## SYSTEM\_TIME

- Standardom rezervirano ime
- počinje u trenutku *system\_start*, i završava u trenutku *system\_end*
- Koristi *closed-open* notaciju
- Pri INSERT-u n-torke, *system\_start* se automatski postavlja na trenutak obavljanja transakcije, a *system\_end* na najveću *timestamp* vrijednost.

Sljedeća INSERT naredba je obavljena 02.02.1980.

```
INSERT INTO osoba VALUES (100, 'Ana', 'Par', '21400');
```

				SYSTEM_PERIOD	
sifOsoba	ime	prezime	pbrPreb	system_start	system_end
100	Ana	Par	21400	02.02.1980	31.12.9999

Početak i kraj perioda **SYSTEM\_TIME** su uvijek **timestamp** tipa. Jednostavnosti radi uzimamo samo datum.

## SQL standard: Relacije transakcijskog vremena - UPDATE

- Vrijednost n-torke prije UPDATE naredbe („old value”) se automatski INSERT-ira u relaciju
- Kraj perioda stare n-torke i početak perioda nove n-torke postavljaju se na vrijeme transakcije
- Korisnici ne smiju mijenjati početak i kraj perioda

SYSTEM_PERIOD					
sifOsoba	ime	prezime	pbrPreb	system_start	system_end
100	Ana	Par	21400	02.02.1980	31.12.9999

Sljedeća UPDATE naredba je obavljena 04.01.2000.

```
UPDATE osoba SET postOznPreb = 10000 where sifOsoba = 100;
```

SYSTEM_PERIOD					
sifOsoba	ime	prezime	pbrPreb	system_start	system_end
100	Ana	Par	21400	02.02.1980	04.01.2000
100	Ana	Par	10000	04.01.2000	31.12.9999

INSERT stare n-torke  
+ UPDATE system\_end

Nova n-torke  
UPDATE system\_start

## SQL standard: Relacije transakcijskog vremena - DELETE

- DELETE naredbom se n-torka ne briše, kraj perioda n-torke se postavlja na vrijeme transakcije

				SYSTEM_PERIOD	
sifOsoba	ime	prezime	pbrPreb	system_start	system_end
100	Ana	Par	21400	02.02.1980	04.01.2000
100	Ana	Par	10000	04.01.2000	31.12.9999

Sljedeća DELETE naredba je obavljena 10.11.2015.

```
DELETE FROM osoba where sifOsoba = 100;
```

				SYSTEM_PERIOD	
sifOsoba	ime	prezime	pbrPreb	system_start	system_end
100	Ana	Par	21400	02.02.1980	04.01.2000
100	Ana	Par	10000	04.01.2000	10.11.2015

# PostgreSQL: Relacije transakcijskog vremena

Podržane kroz proširenje:

```
CREATE EXTENSION temporal_tables;
```

Relacija osoba sadrži timestamp range atribut system\_period za transakcijski period. U njoj se čuvaju aktualne n-torke

```
CREATE TABLE osoba
( sifosoba      SERIAL PRIMARY KEY,
  ime           VARCHAR(50) NOT NULL,
  prezime      VARCHAR(50) NOT NULL,
  postOznPreb  INTEGER,
  system_period tstzrange  NOT NULL DEFAULT tstzrange(current_timestamp, null));
```

U relaciji *osobaHistory* se čuvaju n-torke koje nisu aktualne u trenutku **sada**

```
CREATE TABLE osobaHistory (LIKE osoba);

CREATE TRIGGER versioning_trigger
BEFORE INSERT OR UPDATE OR DELETE ON osoba
FOR EACH ROW EXECUTE PROCEDURE versioning(
  'system_period', 'osobaHistory', true
);
```

# PostgreSQL: Relacije transakcijskog vremena

	Datum	Što se dogodilo u stvarnom svijetu	Akcija u bazi podataka	Stanje u bazi podataka
1	01.01.1980	U Supetru (21400) je rođena Ana Par	nema	nema osobe Ana Par
	01.02.1980	Roditelji su prijavili rođenje Ane Par	nema	nema osobe Ana Par
2	02.02.1980	Administrator je unio podatke o Ani Par u bazu podataka	INSERT	Ana Par prebiva u Supetru
3	01.01.2000	Ana Par se preselila iz Supetra u Zagreb, ali nije prijavila MUP-u	nema	Ana Par prebiva u Supetru
	02.01.2000	Ana Par je prijavila novo prebivalište MUP-u	nema	Ana Par prebiva u Supetru
4	04.01.2000	Administrator je unio podatke o Ani Par u bazu podataka	INSERT	Ana Par prebiva u Zagrebu
5	01.01.2005	Porezna uprava je otkrila da je Ana Par u periodu 01.01.1998 - 31.12.1998 prebivala u Splitu (21000)		

osoba	sifOsoba	ime	prezime	pbrPreb	SYSTEM_PERIOD
	100	Ana	Par	21000	[01.05.2005,)

osobaHistory	sifOsoba	ime	prezime	pbrPreb	SYSTEM_PERIOD
	100	Ana	Par	21400	[02.02.1980, 04.01.2000
	100	Ana	Par	10000	[04.01.2000, 01.05.2005)

Što baza kaže, gdje Ana prebiva sada? Je li to doista mjesto u kojem Ana prebiva sada?  
 Transakcijski period ne može dati točan odgovor na gornje pitanje.  
 Treba pamtit i period valjanosti n-torke.

# SQL standard: Bitemporalne relacije

- Relacija koja je istovremeno transakcijska i relacija vremena valjanosti.
- Podržava period vremena valjanosti i period transakcijskog vremena

```
CREATE TABLE osoba
(sifOsoba      INTEGER PRIMARY KEY,
 . . .
 postOznPreb  VARCHAR(15) REFERENCES mjesto (postOzn) ,
 datumOd     DATE NOT NULL,
 datumDo     DATE NOT NULL,
 system_start TIMESTAMP(6) GENERATED ALWAYS AS ROW START,
 system_end   TIMESTAMP(6) GENERATED ALWAYS AS ROW END,
 PERIOD FOR   osobaPeriod (datumOd, datumDo) ,
 PERIOD FOR   SYSTEM_TIME (system_start, system_end) ,
 PRIMARY KEY (sifOsoba, osobaPeriod WITHOUT OVERLAPS) ,
) WITH SYSTEM VERSIONING;
```



# SQL standard: Bitemporalne relacije

02.02.1980 je administrator evidentirao podatke o Ani Par koja prebiva u Supetru od 01.01.1980

```
INSERT INTO osoba VALUES ('Ana', 'Par', '21400', '01.01.1980', '31.12.9999');
```

				<u>osobaPeriod</u>		SYSTEM_TIME	
<u>sifOsoba</u>	ime	prezime	postOznPreb	datumOd	datumDo	system_start	system_end
100	Ana	Par	21400	01.01.1980	31.12.9999	02.02.1980	31.12.9999

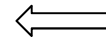
04.01.2000 je administrator evidentirao da Ana Par prebiva u Zagrebu od 01.01.2000

```
UPDATE osoba FOR PORTION OF osobaPeriod FROM DATE '01.01.2000'
                                     TO DATE '31.12.9999'
SET pbrPreb = '10000'
WHERE sifOsoba = 100
```

				<u>osobaPeriod</u>		SYSTEM_TIME	
<u>sifOsoba</u>	ime	prezime	postOznPreb	datumOd	datumDo	system_start	system_end
100	Ana	Par	21400	01.01.1980	31.12.9999	02.02.1980	04.01.2000
100	Ana	Par	21400	01.01.1980	01.01.2000	04.01.2000	31.12.9999
100	Ana	Par	10000	01.01.2000	31.12.9999	04.01.2000	31.12.9999

# SQL standard: Bitemporalne relacije

				<u>osobaPeriod</u>		SYSTEM_TIME	
<u>sifOsoba</u>	ime	prezime	postOznPreb	datumOd	datumDo	system_start	system_end
100	Ana	Par	21400	01.01.1980	31.12.9999	02.02.1980	04.01.2000
100	Ana	Par	21400	01.01.1980	01.01.2000	04.01.2000	31.12.9999
100	Ana	Par	10000	01.01.2000	31.12.9999	04.01.2000	31.12.9999



01.01.2005 je administrator evidentirao da je Ana Par boravila u Splitu u periodu 01.01.1998-01.01.1999

```
UPDATE osoba FOR PORTION OF osobaPeriod FROM DATE '01.01.1998'
                                TO DATE '01.01.1999'
SET pbrPreb = '21000'
WHERE sifOsoba = 100
```

				<u>osobaPeriod</u>		SYSTEM_TIME	
<u>sifOsoba</u>	ime	prezime	postOznPreb	datumOd	datumDo	system_start	system_end
100	Ana	Par	21400	01.01.1980	31.12.9999	02.02.1980	04.01.2000
100	Ana	Par	21400	01.01.1980	01.01.2000	04.01.2000	01.01.2005
100	Ana	Par	21400	01.01.1980	01.01.1998	01.01.2005	31.12.9999
100	Ana	Par	21000	01.01.1998	01.01.1999	01.01.2005	31.12.9999
100	Ana	Par	21400	01.01.1999	01.01.2000	01.05.2005	31.12.9999
100	Ana	Par	10000	01.01.2000	31.12.9999	04.01.2000	31.12.9999

## SQL standard: upiti nad transakcijskim relacijama

Postojeća sintaksa proširuje se opcijama za transakcijski period  
SYSTEM\_TIME:

- FOR SYSTEM\_TIME AS OF
- FOR SYSTEM\_TIME BETWEEN < datetime value expression 1>  
AND < datetime value expression 2>
- FOR SYSTEM\_TIME FROM < datetime value expression 1>  
TO < datetime value expression 2>

PostgreSQL ne podržava nijednu od gornjih opcija.

# PostgreSQL: Što sve mogu doznati upitima nad bitemporalnom relacijom?

1. Gdje prebiva Ana Par trenutno (ili u periodu [datumOd, datumDo))?
2. Što je baza "znala" o prebivalištima Ane Par u periodu [timeStampOd, timeStampDo) ?
3. Što baza u periodu [timeStampOd, timeStampDo) "zna" gdje Ana Par prebiva u periodu [datumOd, datumDo)

1. Upit nad vremenom valjanosti
2. Upit nad transakcijskim vremenom
3. Upit nad vremenom valjanosti i transakcijskim vremenom

## osoba

<u>sifOsoba</u>	ime	prezime	postOznPreb	<u>osobaPeriod</u>	SYSTEM_TIME
100	Ana	Par	21400	[01.01.1980, 01.01.1998)	[01.01.2005 00:00:00.000000 CET,)
100	Ana	Par	21000	[01.01.1998, 01.01.1999)	[01.01.2005 00:00:00.000000 CET,)
100	Ana	Par	21400	[01.01.1999, 01.01.2000)	[01.05.2005 00:00:00.000000 CET,)
100	Ana	Par	10000	[01.01.2000, 31.12.9999)	[04.01.2000 00:00:00.000000 CET,)

## osobaHistory

sifOsoba	ime	prezime	postOznPreb	osobaPeriod	SYSTEM_TIME
100	Ana	Par	21400	[01.01.1980, 31.12.9999)	[02.02.1980 00:00:00.000000 CET, 04.01.2000 00:00:00.000000 CET)
100	Ana	Par	21400	[01.01.1980, 01.01.2000)	[04.01.2000 00:00:00.000000 CET, 1.01.2005 00:00:00.000000 CET)

# PostgreSQL: Što sve mogu doznati upitima nad bitemporalnom relacijom?

1. Gdje prebiva Ana Par na današnji dan (12.11.2015)?

## osoba

<u>sifOsoba</u>	ime	prezime	postOznPreb	<u>osobaPeriod</u>	SYSTEM_TIME
100	Ana	Par	21400	[01.01.1980, 01.01.1998)	[01.01.2005 00:00:00.000000 CET,)
100	Ana	Par	21000	[01.01.1998, 01.01.1999)	[01.01.2005 00:00:00.000000 CET,)
100	Ana	Par	21400	[01.01.1999, 01.01.2000)	[01.05.2005 00:00:00.000000 CET,)
100	Ana	Par	10000	[01.01.2000, 31.12.9999)	[04.01.2000 00:00:00.000000 CET,)

## osobaHistory

sifOsoba	ime	prezime	postOznPreb	osobaPeriod	SYSTEM_TIME
100	Ana	Par	21400	[01.01.1980, 31.12.9999)	[02.02.1980 00:00:00.000000 CET, 04.01.2000 00:00:00.000000 CET)
100	Ana	Par	21400	[01.01.1980, 01.01.2000)	[04.01.2000 00:00:00.000000 CET, 1.01.2005 00:00:00.000000 CET)

```
SELECT postOznPreb FROM osoba
WHERE sifOsoba = 100
      AND osobaPeriod && '[12.11.2015, 13.11.2015)'
```

## postOznPreb

10000

# PostgreSQL: Što sve mogu doznati upitima nad bitemporalnom relacijom?

1. Što je baza "znala" o prebivalištima Ane Par u periodu [1.1.2002 00:00:00.000000 CET, 1.2.2002 00:00:00.000000 CET) ?

<u>sifOsoba</u>	ime	prezime	postOznPreb	<u>osobaPeriod</u>	SYSTEM_TIME
100	Ana	Par	21400	[01.01.1980, 01.01.1998)	[01.01.2005 00:00:00.000000 CET,)
100	Ana	Par	21000	[01.01.1998, 01.01.1999)	[01.01.2005 00:00:00.000000 CET,)
100	Ana	Par	21400	[01.01.1999, 01.01.2000)	[01.05.2005 00:00:00.000000 CET,)
100	Ana	Par	10000	[01.01.2000, 31.12.9999)	[04.01.2000 00:00:00.000000 CET,)

sifOsoba	ime	prezime	postOznPreb	osobaPeriod	SYSTEM_TIME
100	Ana	Par	21400	[01.01.1980, 31.12.9999)	[02.02.1980 00:00:00.000000 CET, 04.01.2000 00:00:00.000000 CET)
100	Ana	Par	21400	[01.01.1980, 01.01.2000)	[04.01.2000 00:00:00.000000 CET, 1.01.2005 00:00:00.000000 CET)

```
SELECT postOznPreb, osobaPeriod FROM osoba
WHERE sifOsoba = 100
      AND system_period && ' [1.1.2002 00:00:00.000000 CET,
                          1.2.2002 00:00:00.000000 CET) '
UNION
SELECT postOznPreb, osobaPeriod FROM osobaHistory
WHERE sifOsoba = 100
      AND system_period && ' [1.1.2002 00:00:00.000000 CET,
                          1.2.2002 00:00:00.000000 CET) '
```

postOznPreb	osobaPeriod
21400	[01.01.1980, 01.01.2000)
10000	[01.01.2000, 31.12.9999)

# PostgreSQL: Što sve mogu doznati upitima nad bitemporalnom relacijom?

- Što je baza "znala" u periodu [1.1.2002 00:00:00.000000 CET, 1.2.2002 00:00:00.000000 CET) gdje je prebivala Ana Par u periodu [01.05.1998, 01.06.1998)

sifOsoba	ime	prezime	postOznPreb	osobaPeriod	SYSTEM_TIME
100	Ana	Par	21400	[01.01.1980, 01.01.1998)	[01.01.2005 00:00:00.000000 CET,)
100	Ana	Par	21000	[01.01.1998, 01.01.1999)	[01.01.2005 00:00:00.000000 CET,)
100	Ana	Par	21400	[01.01.1999, 01.01.2000)	[01.05.2005 00:00:00.000000 CET,)
100	Ana	Par	10000	[01.01.2000, 31.12.9999)	[04.01.2000 00:00:00.000000 CET,)

sifOsoba	ime	prezime	postOznPreb	osobaPeriod	SYSTEM_TIME
100	Ana	Par	21400	[01.01.1980, 31.12.9999)	[02.02.1980 00:00:00.000000 CET, 04.01.2000 00:00:00.000000 CET)
100	Ana	Par	21400	[01.01.1980, 01.01.2000)	[04.01.2000 00:00:00.000000 CET, 1.01.2005 00:00:00.000000 CET)

```

SELECT postOznPreb, osobaPeriod FROM osoba
WHERE sifOsoba = 100
AND osobaPeriod && '[01.05.1998, 1.06.1998)'
AND system_period && '[1.1.2002 00:00:00.000000 CET,
1.2.2002 00:00:00.000000 CET)'

UNION

SELECT postOznPreb, osobaPeriod FROM osobaHistory
WHERE sifOsoba = 100
AND osobaPeriod && '[01.05.1998, 1.06.1998)'
AND system_period && '[1.1.2002 00:00:00.000000 CET,
1.2.2002 00:00:00.000000 CET)'

```

postOznPreb	osobaPeriod
21400	[01.01.1980, 01.01.2000)

- A gdje je ona zapravo prebivala u tom periodu?
- Kada je baza toga "postala svjesna"?

## Literatura

- An Introduction to Temporal Databases; PgConf US 2015  
<https://www.youtube.com/watch?v=1VYhTOHhzbc>
- Range Types and Temporal: Past, Present, and Future (2012)  
[https://www.youtube.com/watch?v=n20hc7\\_y9bE](https://www.youtube.com/watch?v=n20hc7_y9bE)
- Temporal in SQL Server 2016  
[https://www.youtube.com/watch?v=OTi8yn\\_501Q](https://www.youtube.com/watch?v=OTi8yn_501Q)
- [Developing Time-Oriented Database Applications in SQL, Richard T. Snodgrass, Morgan Kaufmann Publishers, Inc., San Francisco, July, 1999, 504+xxiii pages, ISBN 1-55860-436-7.](#)