Temporální databáze

Jan Kolárik
Miroslav Macík

2012

Úvod

- jak zachytit časově proměnnou povahu jevů
- konvenční databáze
  - stav pouze v jednom bodě časové linie
  - aktuální obsah ~ statický snímek (snapshot)
- temporální databáze
  - zohlednění časových vlastností vkládaných dat
  - propracovanější dotazy přes časová období

Studie případů

Zamestnanec(Jmeno,Plat,Titul)

Jaký je Karlův plat?

```
SELECT Plat
FROM Zamestnanec
WHERE Jmeno = 'Karel'
```

Studie případů

Zamestnanec(Jmeno,Plat,Titul,Narozen **DATE**)

Kdy se narodil Karel?

```
SELECT Narozen
FROM Zamestnanec
WHERE Jmeno = 'Karel'
```

- omezená podpora temporálních dat v SQL

Studie případů - temporální projekce

Zamestnanec(Jmeno,Plat,Titul,Narozen,Start **DATE**, Stop **DATE**)

Jaký je aktuální Karlův plat?

```
SELECT Plat
FROM Zamestnanec
WHERE Jmeno = 'Karel' AND Start <=
CURRENT_DATE AND CURRENT_DATE <= Stop
```

Studie případů - temporální projekce

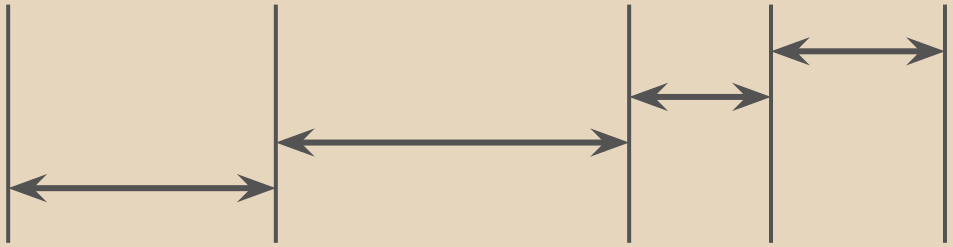
Jmeno	Plat	Titul	Narozen	Start	Stop
Karel	60 000	Asistent	09.04.1945	01.01.1995	01.06.1995
Karel	70 000	Asistent	09.04.1945	01.06.1995	01.10.1995
Karel	70 000	Docent	09.04.1945	01.10.1995	01.02.1996
Karel	70 000	Profesor	09.04.1945	01.02.1996	01.01.1997

Jmeno	Plat	Start	Stop
Karel	60 000	01.01.1995	01.06.1995
Karel	70 000	01.06.1995	01.01.1997

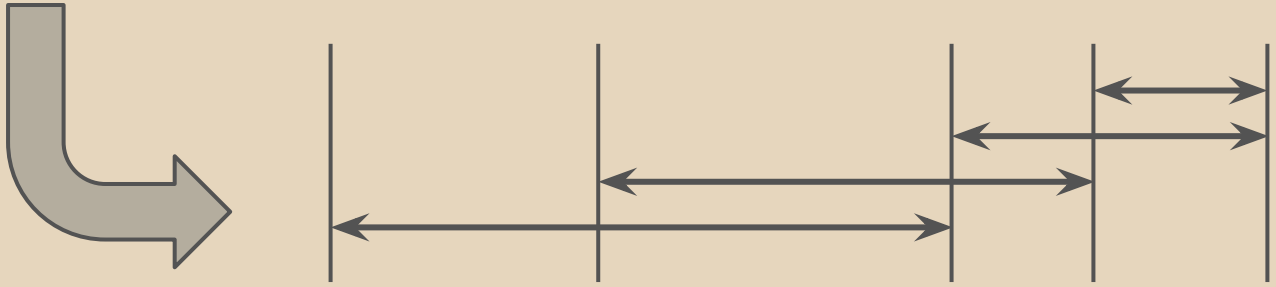
Jaká je historie platů Karla ? (1.)

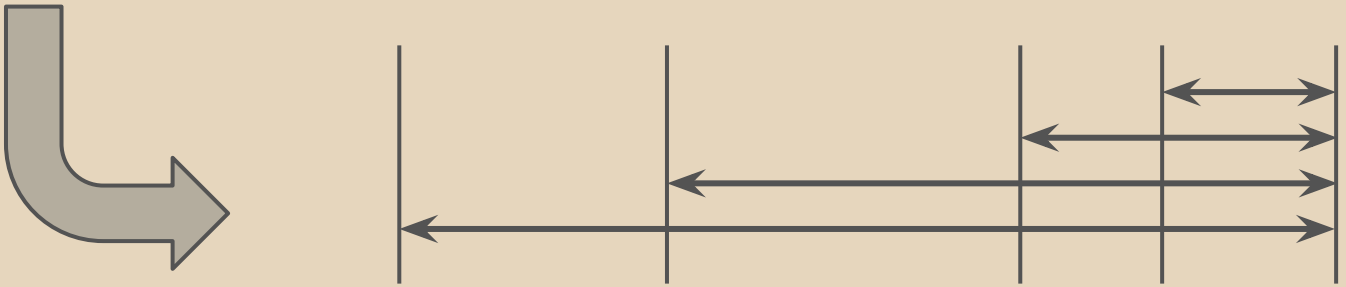
```
CREATE TABLE Temp(Plat,Start,Stop)
AS SELECT Plat,Start,Stop FROM Zamestnanec WHERE Jmeno = 'Karel';

repeat
    UPDATE Temp T1
    SET (T1.Stop) = ( SELECT MAX(T2.Stop) FROM Temp AS T2
        WHERE T1.Plat = T2.Plat AND T1.Start < T2.Start
            AND T1.Stop >= T2.Start AND T1.Stop < T2.Stop )
        WHERE EXISTS ( SELECT * FROM Temp AS T2
            WHERE T1.Plat = T2.Plat AND T1.Start < T2.Start
                AND T1.Stop >= T2.Start AND T1.Stop < T2.Stop )
until no updates;

DELETE FROM Temp T1 WHERE EXISTS ( SELECT * FROM Temp AS T2
    WHERE T1.Plat = T2.Plat AND ( (T1.Start > T2.Start AND T1.Stop <= T2.Stop)
        OR (T1.Start >= T2.Start AND T1.Stop < T2.Stop) )
```

Jaká je historie platů Karla ? (1.)



průběh iterací





Jaká je historie platů Karla ? (2.)

```
CREATE TABLE Temp(Plat,Start,Stop)
AS SELECT Plat,Start,Stop FROM Zamestnanec WHERE Jmeno = 'Karel';

SELECT DISTINCT F.Plat,F.Start,L.Stop FROM Temp AS F, Temp AS L
WHERE F.Start < L.Stop AND F.Plat = L.Plat
      AND NOT EXISTS ( SELECT *
            FROM Temp AS M WHERE M.Plat = F.Plat AND F.Start < M.Start
            AND M.Start < L.Stop
            AND NOT EXISTS ( SELECT *
                  FROM Temp AS T1 WHERE T1.Plat = F.Plat
                  AND T1.Start < M.Start AND M.Start <= T1.Stop ) )
      AND NOT EXISTS ( SELECT *
            FROM Temp AS T2 WHERE T2.Plat = F.Plat
                  AND ((T2.Start < F.Start AND F.Start <= T2.Stop) OR
                        (T2.Start < L.Stop AND L.Stop < T2.Stop)) )
```

Jaká je historie platů Karla ? (3.)

SELECT Plat
FROM Zamestnanec
WHERE Jmeno = 'Karel'

- použití temporálního jazyka TSQL2

Studie případů - temporální spojení

Zamestnanec  Zamestnanec1(Jmeno,Plat,Start,Stop)
Zamestnanec2(Jmeno,Titul,Start,Stop)

Jaká je historie platů Karla?

```
SELECT Plat,Start,Stop
FROM Zamestnanec1
WHERE Jmeno = 'Karel'
```

Jaká je historie platů a titulů zaměstnanců ?

- rozborem případů

```
SELECT Zamestnanec1.Jmeno,Plat,Titul,Zamestnanec1.Start,Zamestnanec1.Stop
FROM Zamestnanec1, Zamestnanec2
WHERE Zamestnanec1.Jmeno = Zamestnanec2.Jmeno
    AND Zamestnanec2.Start <= Zamestnanec1.Start
    AND Zamestnanec1.Stop <= Zamestnanec2.Stop
UNION
SELECT Zamestnanec1.Jmeno,Plat,Titul,Zamestnanec1.Start,Zamestnanec1.Stop
FROM Zamestnanec1, Zamestnanec2
...
```

Jaká je historie platů a titulů zaměstnanců ?

```
SELECT Zamestnanec1.Jmeno,Plat,Titul
FROM Zamestnanec1,Zamestnanec2
WHERE Zamestnanec1.Jmeno =
          Zamestnanec2.Jmeno
```

- použití temporálního jazyka TSQL2

Shrnutí studie

- s časově závislými daty se běžně pracuje
- klasické SŘBD neposkytují dostatečnou podporu
- pro tyto případy jsou vhodné temporální systémy

Časová doména

- čas je libovolná množina okamžiků s částečným uspořádáním

- další axiomy popisují detailnější modely
  - lineární model - definované úplné uspořádání
  - rozvětvený model - větvení do několika časových linií
  - cyklický model - rekurentní události

Časová doména

- dělení podle hustoty
  - diskrétní model
    - každé přirozené číslo odpovídá jednotce času, která je dále nedělitelná ~ chronon
  - hustý model
    - může obsahovat "časové mezery"
  - spojitý model
    - každé reálné číslo odpovídá časovému bodu
- v praxi je nejpoužívanější diskrétní model

Časová doména

- omezení časové domény
- koncept vzdálenosti
  - zavedení metriky
- absolutní vs. relativní čas

Časová doména - datové typy

- okamžik
  - specifický chronon na časové ose
  - v SQL: DATE, TIME, TIMESTAMP
- časová perioda
- časový interval
  - předem známý časový úsek
  - není zasazen do časové linie

Časová doména - asociace faktů s časem

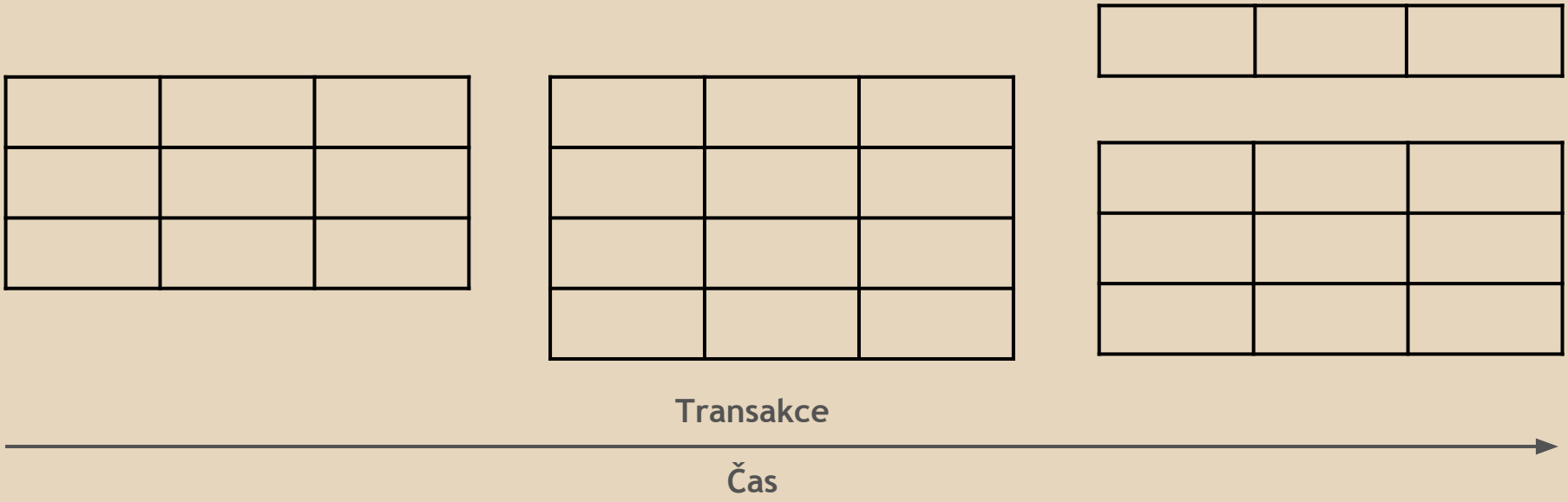
- dvě ortogonální časové dimenze
- čas platnosti (valid time)
  - časové období, kdy je daný fakt pravdivý
- transakční čas (transaction time)
  - období, kdy byl fakt reprezentován v databázi

Časová doména - asociace faktů s časem

- Snímek (snapshot)
  - nepodporuje ani jednu dimenzi
  - při změně reality dochází ke změně relace

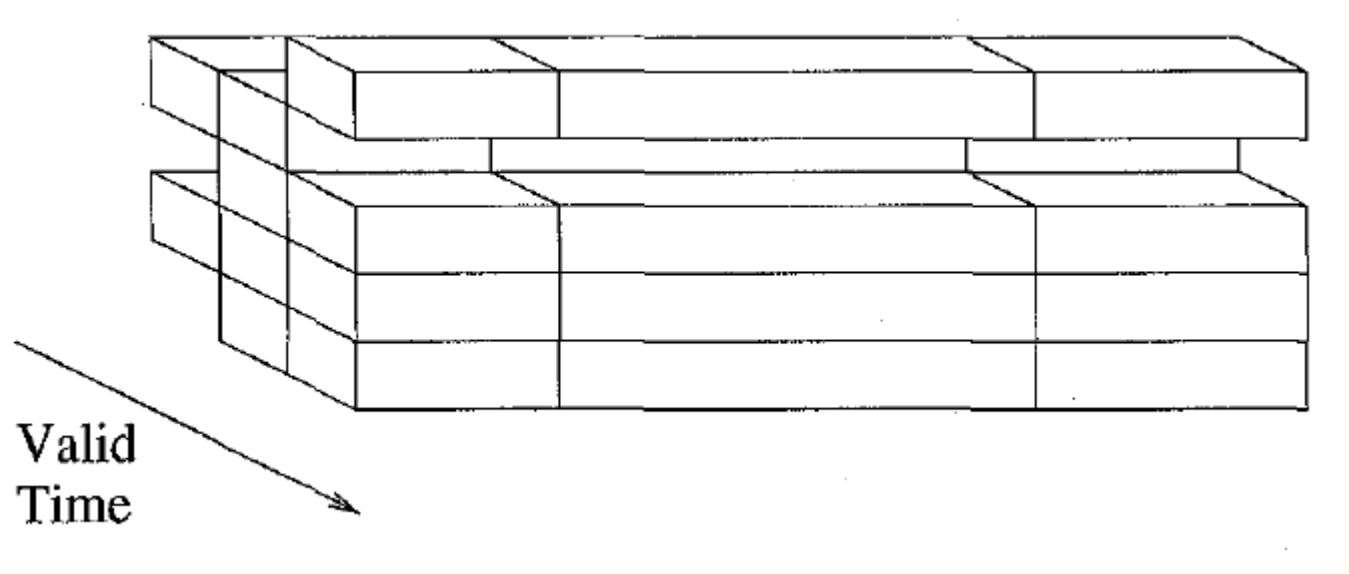
Časová doména - asociace faktů s časem

- Transaction-time model
  - podporuje pouze transakční čas
  - nemění existující data
  - vhodné pro dotazy do minulosti
  - povoluje pouze přidávání - je append-only



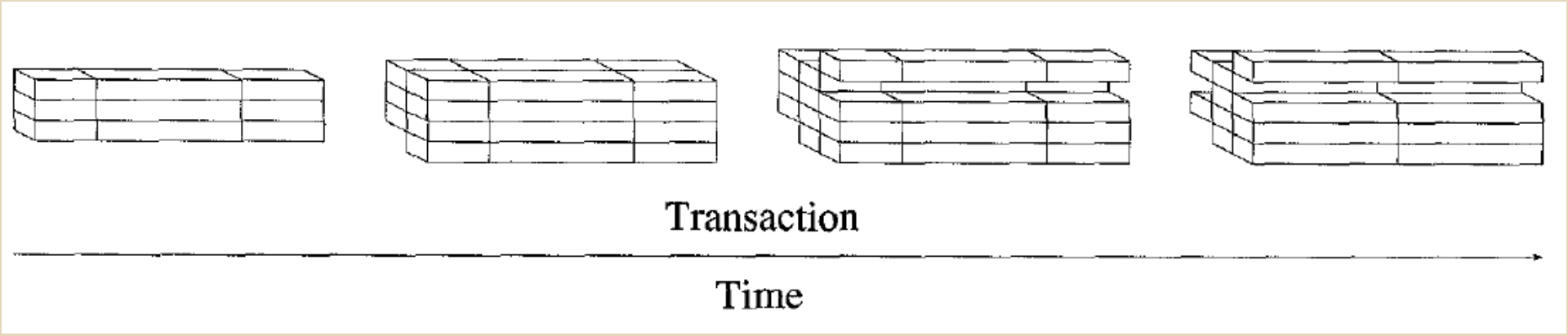
Časová doména - asociace faktů s časem

- Valid-time model
  - podporuje pouze čas platnosti



Časová doména - asociace faktů s časem

- Bitemporální model
  - 4dimenzionální struktura
  - valid-time i transaction-time dotazy
  - append-only



Shrnutí

- temporální datové modely se snaží plnit spousta požadavků
  - jasná a výstižná sémantika aplikace
  - konzistence
  - minimální rozšíření dosavadního datového modelu
  - prezentace faktů souvisle v čase
  - jednoduchá implementace
  - vysoký výkon
- dosažení ideálního modelu je téměř nemožné
  - existence mnoha nekompatibilních datových modelů
  - existence různých nekompatibilních dotaz. jazyků

TSQL2

TSQL2

- Temporal Structured Query Language

- sjednocení přístupu k temporálním datovým modelům a dotazovacím jazykům
- základem je SQL-92
- částečně zakomponováno ve standardu SQL3

Reprezentace času

- reprezentace pomocí časové osy
  - diskrétní
  - omezena z obou stran
    - dostatečně velká
- chronony
  - atomické části
  - tvoří časovou osu
- odpovědi na dotazy jsou závislé na zvolené granularitě

Datový model

- Bitemporal Conceptual Data Model
- v relaci je každému řádku přiřazena množina bitemporálních chrononů
- bitemporální chronon je dvojice chrononu transakčního času a chrononu času platnosti

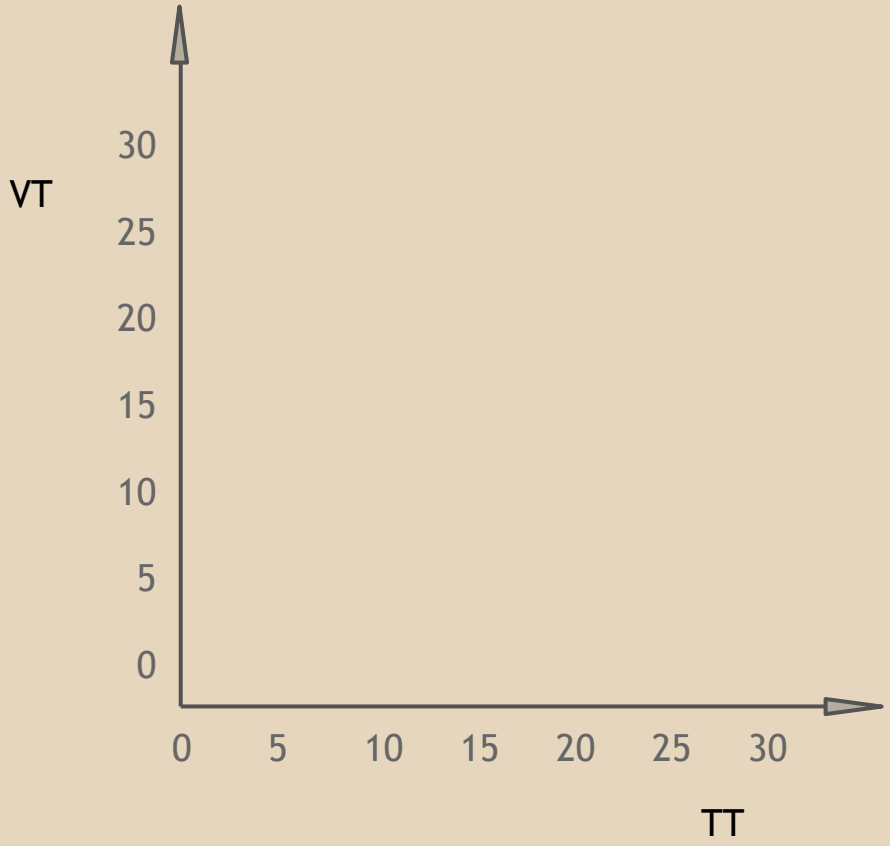
Příklad bitemporální relace

- relace zaměstnanec - oddělení
  - *Jake pracuje pro spediční oddělní*

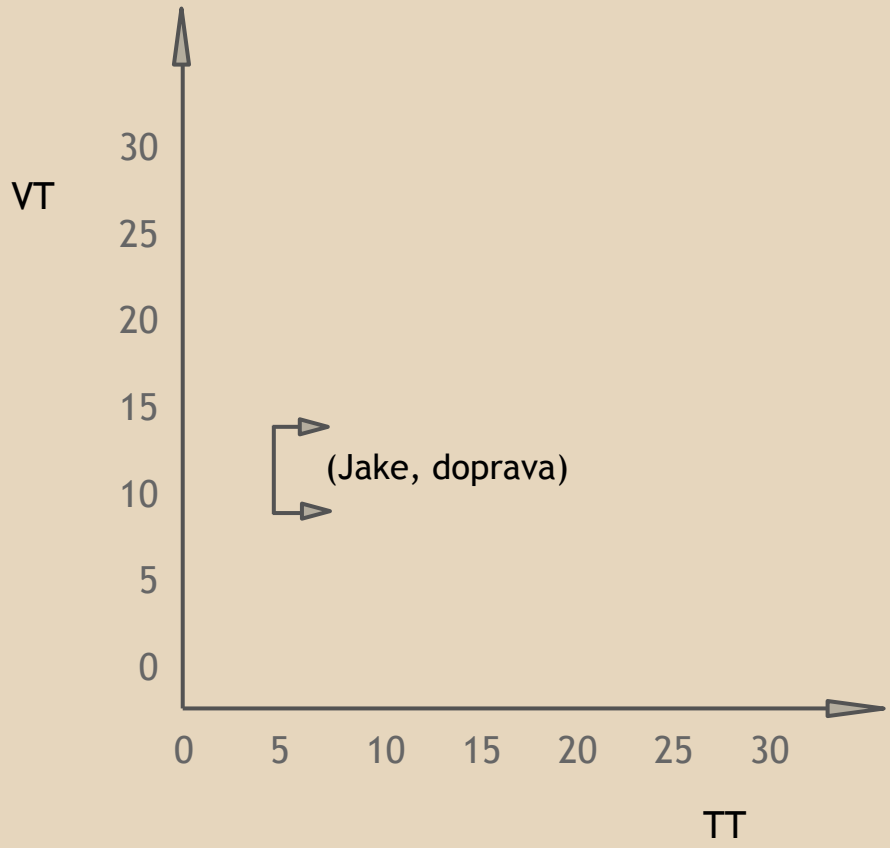
- granularita 1 den pro transakční čas i čas platnosti
- schéma (jméno, oddělení) + časové razítko

- možnost grafické reprezentace

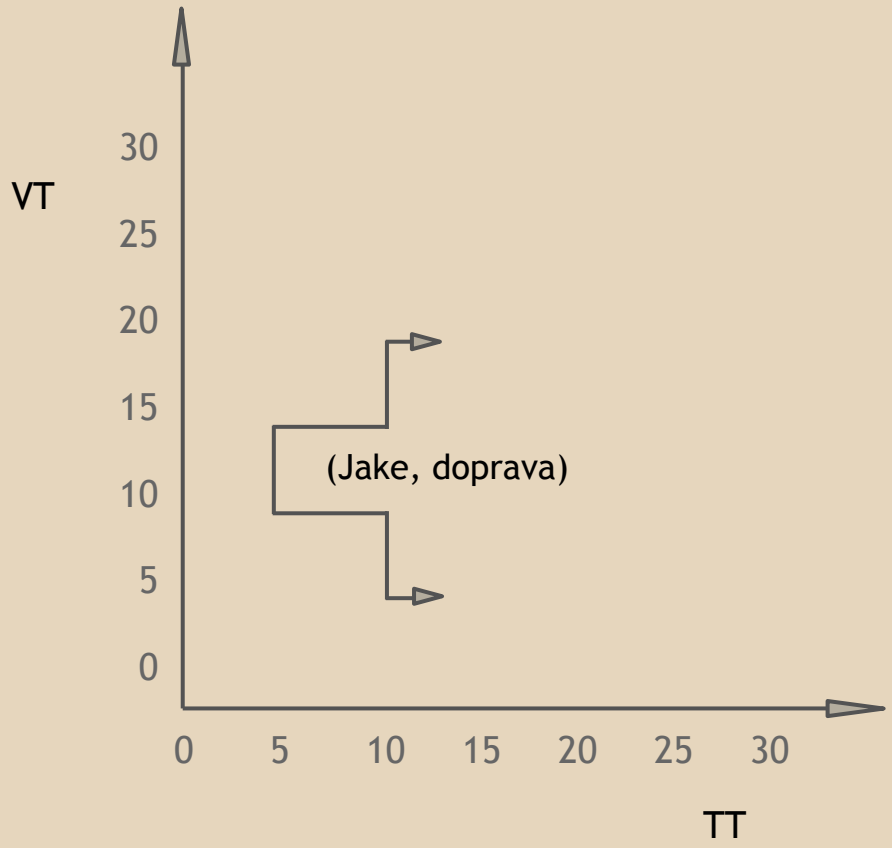
Příklad bitemporální relace



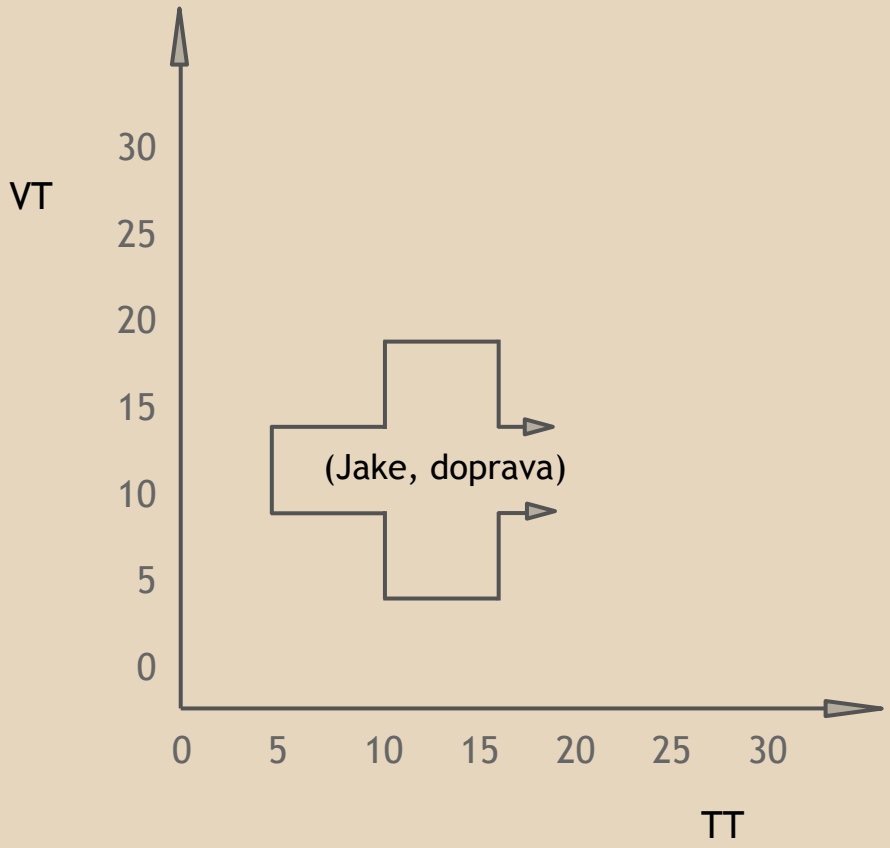
Příklad bitemporální relace



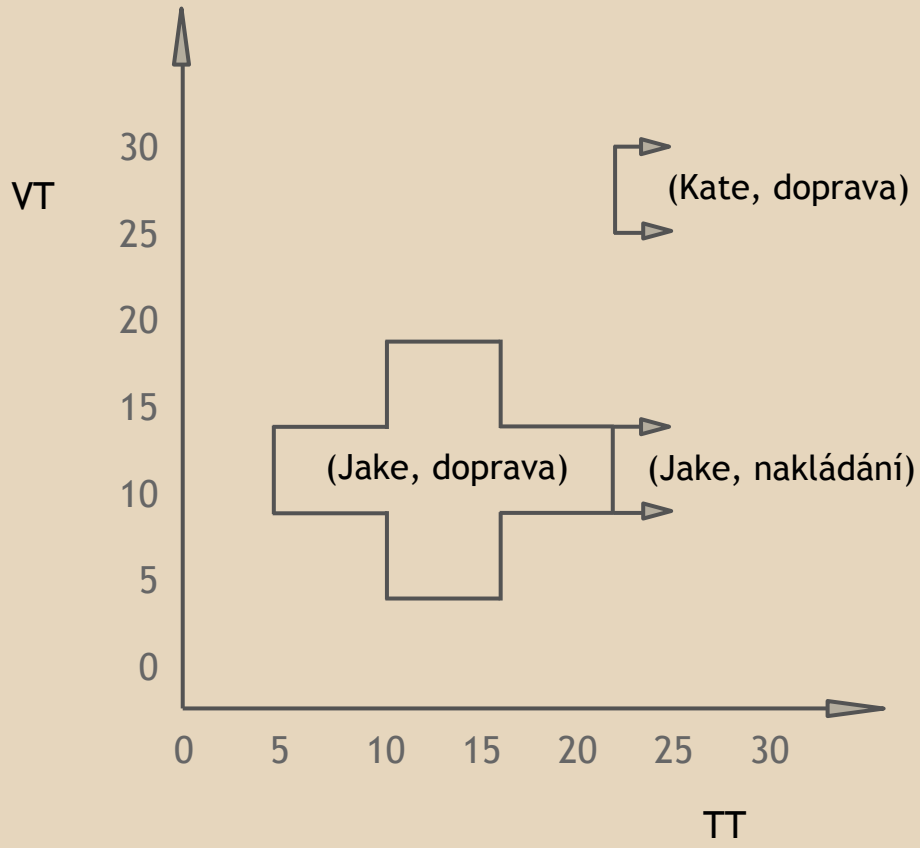
Příklad bitemporální relace



Příklad bitemporální relace



Příklad bitemporální relace



Příklad bitemporální relace

Jméno	Oddělení	Bitemporální chronony
Jake	Doprava	{(5, 10), .., (5, 15), .., (9, 10), .., (9, 15), (10, 5), .., (10, 20), .., (14, 5), .., (14, 20), (15, 10), ..,(15, 15), .., (19, 10), .., (19, 15)}
Jake	Nakládání	{(U. C., 10), .., (U. C., 15)}
Kate	Doprava	{(U. C., 25), .., (U. C., 30)}

Definice schématu

Vytvoření tabulky pro předepisování léků.

CREATE TABLE Předpis (Jméno **CHAR**(30),
Lékař **CHAR**(30), Lék **CHAR**(30), Dávkování
CHAR(30), Frekvence **INTERVAL MINUTE**)
AS VALID STATE DAY AND TRANSACTION

frekvence - počet minut mezi dávkami
valid time - kdy je lék předepsán
transaction time - přidání záznamu do databáze

Příkaz SELECT

- nové klíčové slovo **SNAPSHOT** pro snímek databáze

Kdo měl nebo má předepsané léky?
SELECT SNAPSHOT Jméno
FROM Předpis

Příkaz SELECT

Kdo měl nebo má předepsaný lék Proventil?

SELECT SNAPSHOT Jméno
FROM Předpis
WHERE Lék = 'Proventil'

Příkaz SELECT

Kdo měl předepsané léky a kdy?

SELECT Jméno
FROM Předpis

- defaultně vrací historii
- automaticky je provedena koalescence
- výsledkem je množina řádků, každý s maximální periodou, kdy pacient užíval jeden nebo více léků

Příkaz SELECT

Jaké léky byly užívány společně s lékem Proventil?

```
SELECT P1.Jméno, P2.Lék
FROM Předpis AS P1, Předpis AS P2
WHERE P1.Lék = 'Proventil'
    AND P2.Lék <> 'Proventil'
    AND P1.Jméno = P2.Jméno
```

Restrukturalizace

- koalescence na úrovni klauzule **FROM**

Kdo užíval lék déle než celkově šest měsíců?

```
SELECT Jméno, Lék
FROM Předpis(Jméno, Lék) AS P
WHERE CAST(VALID(P) AS INTERVAL MONTH)
    > INTERVAL '6' MONTH
```

Restrukturalizace

SELECT Jméno, Lék
FROM Předpis(Jméno, Lék) **AS** P
WHERE CAST(**VALID**(P) **AS INTERVAL MONTH**)
> **INTERVAL** '6' **MONTH**

- **VALID**(P) vrátí pro každý řádek P dobu platnosti
- **CAST** provede konverzi na typ **INTERVAL MONTH** součtem časových bloků

Restrukturalizace

Kdo užíval Proventil po celou dobu své léčby?

```
SELECT SNAPSHOT P1.Jméno
FROM Předpis(Jméno) AS P1, P1(Lék) AS P2
WHERE P2.Lék = 'Proventil'
    AND VALID(P2) = VALID(P1)
```

Restrukturalizace

- dotaz lze zapsat i bez restrukturalizace

Kdo užíval Proventil po celou dobu své léčby?

```
SELECT SNAPSHOT P1.Jméno
FROM (SELECT Jméno FROM Předpis) AS P1,
    (SELECT Jméno, Lék FROM Předpis) AS P2
WHERE P1.Jméno = P2.Jméno AND P2.Lék =
    'Proventil' AND VALID(P2) = VALID(P1)
```

Rozkládání

- chceme zkoumat maximální periody

Kdo užíval stejný lék nepřerušovaně po dobu alespoň 6 měsíců?

```
SELECT Jméno, Lék
FROM Předpis(Jméno, Lék)(PERIOD) AS P
WHERE CAST(VALID(P) AS INTERVAL MONTH)
    > INTERVAL '6' MONTH
```

Klauzule VALID

- pro každý řádek je čas platnosti vypočten jako průnik časů platnosti relací uvedených v klauzuli **FROM**
- klauzule **VALID** upravuje toto standardní chování

Klauzule VALID

Jaké léky užívala Melanie během roku 1996?

```
SELECT Drug
VALID INTERSECT(VALID(Předpis),
    PERIOD '[1996]' DAY)
FROM Předpis
WHERE Jméno = 'Melanie'
```

Práce s daty

- příkazy pro vkládní, mazání a úpravu dat v databázi

- **INSERT**
- **DELETE**
- **UPDATE**

INSERT

- vkládání dat do databáze

Vložení nového předpisu.

```
INSERT INTO Předpis
VALUES ('Melanie', 'Dr. Beren', 'Proventil',
    '100mg', INTERVAL '8:00' MINUTE)
```

INSERT

- automatická doba platnosti

```
INSERT INTO Předpis
VALUES ('Melanie', 'Dr. Beren', 'Proventil',
    '100mg', INTERVAL '8:00' MINUTE)

VALID PERIOD(CURRENT_TIMESTAMP,
    NOBIND(CURRENT_TIMESTAMP))
```

INSERT

Vložení nového předpisu se známou dobou platnosti.

```
INSERT INTO Předpis
VALUES ('Melanie', 'Dr. Beren', 'Proventil',
    '100mg', INTERVAL '8:00' MINUTE)
VALID PERIOD '[1996-01-01 - 1996-06-03]'
```

DELETE

- odstranění dat z databáze

Melanie neměla v červnu roku 1996 předepsaný žádný lék.

DELETE FROM Předpis
WHERE Jméno = 'Melanie'
VALID PERIOD '[1996-06-01 - 1996-06-30]'

UPDATE

- úprava dat v databázi

Upravení dávkování léku Proventil na 50 mg.

UPDATE Předpis
SET Dávkování **TO** '50 mg'
WHERE Jméno = 'Melanie' **AND** Lék = 'Proventil'

UPDATE

Upravení dávkování léku Proventil na 50 mg ***od března do května.***

UPDATE Předpis
SET Dávkování **TO** '50 mg'
VALID PERIOD '[1996-03-01 - 1996-05-30]'
WHERE Jméno = 'Melanie' **AND** Lék = 'Proventil'

Události

- zaznamenání okamžité události

Vytvoření tabulky pro evidenci laboratorních vyšetření pacientů.

CREATE TABLE LabTest (Jméno **CHAR**(30),
    Lékař **CHAR**(30), TestID **INTEGER**)
AS VALID EVENT HOUR AND TRANSACTION

Události

Pro kterého lékaře platí, že objednal testy pouze jednomu pacientovi a zároveň byl tento pacient na testech objednaných pouze tímto lékařem?

```
SELECT L1.Jméno, L2.Lékař
FROM LabTest(Jméno) AS L1, L1(Lékař) AS L2,
    LabTest(Lékař) AS L3
WHERE VALID(L1) = VALID(L2)
    AND L2.Lékař = L3.Lékař
    AND VALID(L1) = VALID(L3)
```

Podpora transakčního času

- umožňuje nám zjistit jaké byly hodnoty v databázi v námi určené době

Podpora transakčního času

Jaké záznamy o předepsaných lécích měla Melanie v databázi 1. června 1996?

```
SELECT Lék
FROM Předpisy AS P
WHERE Jméno = 'Melanie'
    AND TRANSACTION(P) OVERLAPS DATE
     '1996-06-01'
```

Agregační funkce

- podpora známých funkcí z SQL-92
- **MIN**, **MAX**, **COUNT**, **SUM** a **AVG**

Kolik léků Melanie brala?

```
SELECT COUNT(*)
FROM Předpis
WHERE Jméno = 'Melanie'
```

Agregační funkce

Kolik pacientů má předepsané jednotlivé léky?

```
SELECT Lék, COUNT(*)
FROM Předpis
GROUP BY Lék
```

Agregační funkce

- TSQL přidává novou funkci **RISING**
  - nejdelší období ve kterém daný atribut monotónně rostl

Jak dlouhé bylo nejdelší období, ve kterém dávka léku Proventil rostla?

SELECT SNAPSHOT RISING(Dávkování)
FROM Předpis
WHERE Jméno = 'Melanie' **AND** Lék = 'Proventil'

Úprava a verzování schématu

- úprava schématu příkazem **ALTER**
- schéma je automaticky verzováno
  - pokud má relace podporu transakčního času

Úprava a verzování schématu

Rozšíření tabulky Předpis o sloupec ID.

```
ALTER TABLE Předpis
    ADD COLUMN ID INTEGER
```

Přechod na verzi schématu ze dne 19. srpna 1996.

```
SET SCHEMA DATE '1996-08-19'
```

Shrnutí

- práce s prvky, které se mění v čase

- dotazy bez podpory času pomocí **SNAPSHOT**
- slévání časů platnosti v klauzuli **FROM** pomocí restrukturalizace
- výběr nejdelších časů platnosti pomocí rozkládání klauzulí **PERIOD**
- ověření validity dat pomocí klauzule **VALID**

Zdroje

ZANIOLO Carlo. Advanced Database Systems.
ISBN 155860443X