



# Temporální databáze a TSQL

Jan Ulrych, Vít Urbánek, Petr Uher  
(v pořadí přednášení)

# Obsah

- Motivace
- Studie konkrétních případů
- Vztah událostí a času
- Ukázky vybraných datových modelů
- Shrnutí teorie
- TSQL



# Motivace

- Potřebujeme v databázích čas?
- Studijní informační systém
- Skladová evidence
- Účetní a bankovní systémy
- Docházkové systémy
- ... a mnoho dalších

# Klasické vs. temporální databáze

- Klasický databázový systém
  - Zachycen stav systému v aktuálním časovém okamžiku
  - Problém: co dělat se starými daty
- Temporální databázový systém
  - Databáze určitým způsobem podporující čas
  - Jednodušší dotazy
  - Jednodušší udržování aplikací

# Studie konkrétního případu

## ■ Příklad v SQL:

- Zaměstnanec(  
    Jméno, Plat, Funkce)

- Jaký plat má Pepa?

- **SELECT** Plat  
    **FROM** Zaměstnanec  
    **WHERE** Jméno = 'Pepa'

# Studie konkrétního případu

## ■ Příklad v SQL:

□ Zaměstnanec(Jméno, Plat, Funkce,  
Datum\_narození **DATE**)

□ Kdy se Pepa narodil?

□ **SELECT** Datum\_narození  
**FROM** Zaměstnanec  
**WHERE** Jméno = 'Pepa'

# Studie konkrétního případu

## ■ Příklad v SQL:

□ Zaměstnanec(Jméno, Plat, Funkce,  
Datum\_narození,  
Platí\_od **DATE**, Platí\_do **DATE**)

□ Jaký je Pepův aktuální plat?

□ **SELECT** Plat  
**FROM** Zaměstnanec  
**WHERE** Jméno = 'Pepa'  
**AND** Platí\_od <= **CURRENT\_DATE**  
**AND** **CURRENT\_DATE** <= Platí\_do

# Temporální projekce

- Jako projekce v klasických databázích
- Navíc bere v úvahu čas
  - Srůstání
- Nedostatečná podpora v klasických databázových systémech



# Temporální projekce – příklad

Jméno	Plat	Funkce	Datum narození	Platí_od	Platí_do
Pepa	60000	Vrátný	1945-04-09	1995-01-01	1995-06-01
Pepa	70000	Vrátný	1945-04-09	1995-06-01	1995-10-01
Pepa	70000	Vrchní vrátný	1945-04-09	1995-10-01	1995-02-01
Pepa	70000	Ředitel bezpečnosti	1945-04-09	1996-02-01	1997-01-01

## ■ Příklad:

- Jaká Pepova platová historie?

Jméno	Plat	Platí_od	Platí_do
Pepa	60000	1995-01-01	1995-06-01
Pepa	70000	1995-06-01	1997-01-01

# Temporální spojení

- Stejně jako spojení (join) v klasických databázích
- Navíc bere v úvahu čas události
- Příklad v SQL:
  - Zaměstnanec1(Jméno, Plat, Platí\_od, Platí\_do)  
Zaměstnanec2(Jméno, Funkce, Platí\_od, Platí\_do)
  - Pro každého zaměstnance najděte historii platů a funkcí
  - Myšlenka: Pro každou dvojici (Plat, Funkce) najít průnik intervalů (Platí\_od, Platí\_do) z obou relací.
  - Realizace: Rozbor případů

# Studie konkrétního případu

- Historie platů a funkcí v TSQL
  - **SELECT** Zaměstnanec1.Plat,  
Zaměstnanec2.Funkce  
**FROM** Zaměstnanec1, Zaměstnanec2  
**WHERE** Zaměstnanec1.Jméno =  
Zaměstnanec2.Jméno

# Modely času

- Temporální logika: čas je libovolná množina okamžiků s daným uspořádáním
- Modely času podle uspořádání
  - Lineární
  - Větvený (čas možných budoucností)
  - Cyklický
- Modely času podle hustoty
  - Diskrétní
  - Hustý
  - Spojitý
- Omezenost času
- Absolutní / relativní čas

# Datové typy pro čas

- Časový okamžik (instant) (SQL-92)
  - DATE
  - TIME
  - TIMESTAMP
- Časový úsek (time period)
  - Doba mezi dvěma časovými okamžiky
  - 15:30 – 15:50
- Časový interval (interval)
  - Doba o specifikované délce, ale bez konkrétních krajních bodů
  - 30 minut
- Množina časových okamžiků (instant set)
- Množina časových úseků (temporal elements)

# Vztah událostí a času

- Čas platnosti (valid time)
  - Čas, kdy byla událost pravdivá v reálném světě
  - Může být v minulosti, přítomnosti i budoucnosti
- Transakční čas (transaction time)
  - Čas, kdy byl fakt reprezentován v databázi
  - Nabývá **pouze** aktuální hodnoty
  - Monotónně roste
- Čas platnosti a transakční čas jsou ortogonální

# Vztah událostí a času

- snapshot
  - Datový model nepodporující čas platnosti ani transakční čas
- valid-time
  - Datový model podporující pouze čas platnosti
- transaction-time
  - Datový model podporující pouze transakční čas
- bitemporální
  - Datový model podporující čas platnosti i transakční čas
- temporální
  - Datový model podporující čas platnosti nebo transakční čas
- Obvykle založeno na relačním datovém modelu nebo objektově orientovaném datovém modelu.

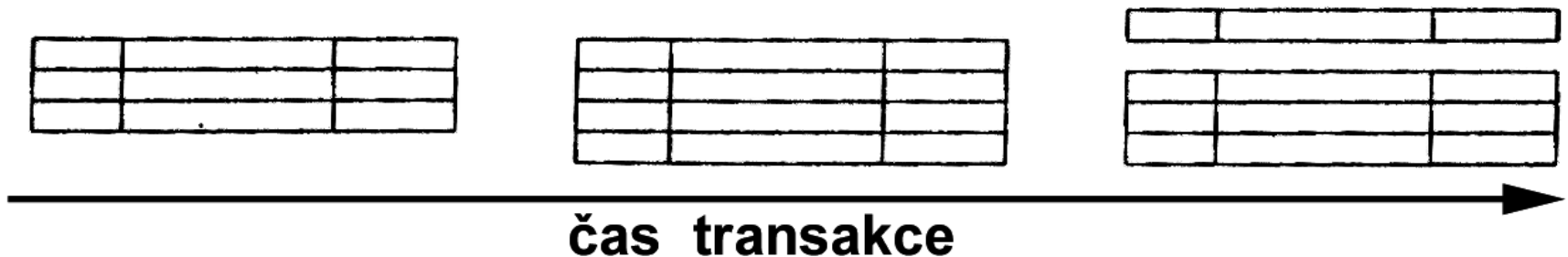
# snapshot relace

- Datový model nepodporující čas platnosti ani transakční čas
- Klasický relační model
- Každá n-tice je fakt platný v reálném světě
- Při změně reálného světa jsou do relace prvky přidávány nebo z ní odebírány



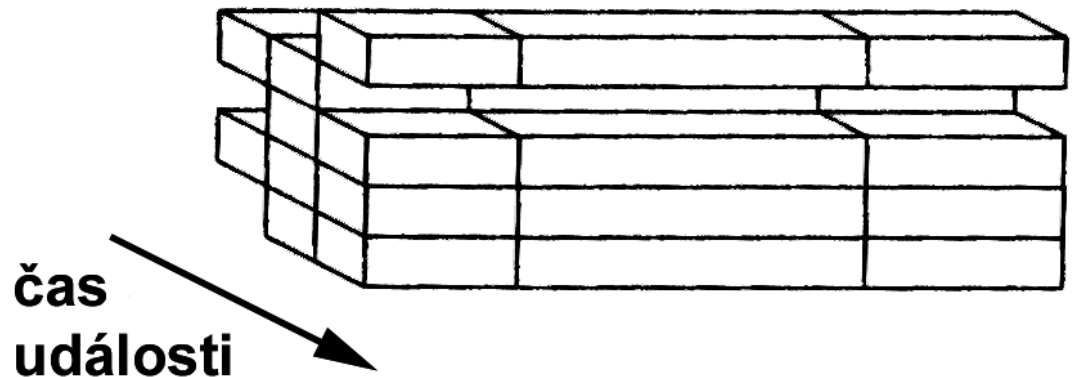

# transaction-time relace

- Datový model podporující pouze transakční čas
- Posloupnost snapshot-ů indexované transakčním časem
- Umožňuje získat informaci ze stavu databáze v nějakém okamžiku minulosti
- Je možné uvažovat i větvení



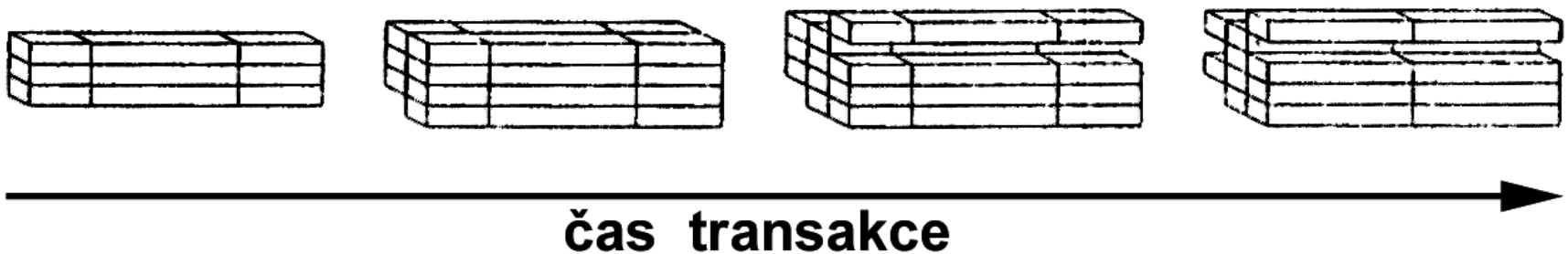
# valid-time relace

- Datový model podporující pouze čas platnosti
- Cokoliv v relaci může být upraveno
  - Hodnoty n-tic
  - Čas události (začátek i konec)
- Umožňuje klást dotazy o faktech platných v minulosti i budoucnosti



# bitemporální relace

- Datový model podporující čas platnosti a transakční čas
- append-only



# Reprezentace „času platnosti“

- Reprezentace času platnosti
  - časový okamžik
  - doba
  - časový úsek
  - množina okamžiků
- Čas platnosti může být přidružen k
  - atributu
  - množině atributů
  - celé n-tici nebo objektu
- Rozšíření operací relační algebry

# Reprezentace „transakčního času“

- Reprezentace transakčního času
  - časový okamžik
    - nová n-tice se stejným klíčem => logické odstranění původní
  - časový úsek
    - (teď, *dokud nezměněno*)
  - tři časové okamžiky
    - čas zaznamenání začátku události v reálném světě
    - čas zaznamenání konce události v reálném světě
    - čas logického odstranění události z databáze
  - Množina časových úseků
- Někdy podporován i versioning

# Příklady datových modelů (1)

- Segevův datový model

- Podporuje pouze čas platnosti
- Razítko označuje, kdy fakt začal platit

<i>Jméno</i>	<i>Oddělení</i>	<i>Čas</i>
Erik	Obuv	1
Erik	Knihy	6
Erik	Obuv	11
Erik	Null	13

# Příklady datových modelů (2)

- Sardův datový model
  - Podporuje jen čas platnosti
  - Razítko má podobu intervalu

<i>Jméno</i>	<i>Oddělení</i>	<i>Čas</i>
Erik	Obuv	[1-5]
Erik	Knihy	[6-10]
Erik	Obuv	[11-12]

# Příklady datových modelů (3)

- Datový model HRDM
  - Podporuje jen čas platnosti
  - Razítkují se hodnoty atributů

<i>Jméno</i>	<i>Oddělení</i>
1 → Erik	1 → Obuv
...	...
	5 → Obuv
12 → Erik	6 → Knihy
	...
	10 → Knihy
	11 → Obuv
	12 → Obuv



# Příklady datových modelů (4)

- Extensionální datový model

<i>Jméno</i>	<i>Odd</i>	<i>VT</i>	<i>TT</i>
Erik	Obuv	1	1
Erik	Obuv	2	1
...	...	...	...
Erik	Obuv	1	2
Erik	Obuv	2	2
...	...	...	...
Erik	Obuv	1	8
...	...	...	...
Erik	Obuv	5	8
Erik	Knihy	6	8
...	...	...	...
Erik	Obuv	1	11
...	...	...	...
Erik	Obuv	5	11

<i>Jméno</i>	<i>Odd</i>	<i>VT</i>	<i>TT</i>
Erik	Knihy	6	11
...	...	...	...
Erik	Knihy	10	11
Erik	Obuv	11	11
...	...	...	...
Erik	Obuv	1	13
...	...	...	...
Erik	Obuv	5	13
Erik	Knihy	6	13
...	...	...	...
Erik	Knihy	10	13
Erik	Obuv	11	13
Erik	Obuv	12	13

# Příklady datových modelů (5)

## ■ Bhargavův datový model

Jméno	Oddělení
[1, 12] x [1, ∞] Erik	[1, 7] x [1, ∞] Obuv
[13, →] x [1, 12] Erik	[8, 10] x [1, 5] Obuv
	[8, 10] x [6, ∞] Knihy
	[11, 12] x [1, 5] Obuv
	[11, 12] x [6, 10] Knihy
	[11, 12] x [11, ∞] Obuv
	[13, →] x [1, 5] Obuv
	[13, →] x [6, 10] Knihy
	[13, →] x [11, 12] Obuv

# Temporální dotazovací jazyky

- Temporální datový model
  - Objekty se přesně danou strukturou
  - Omezení pro dané objekty
  - Operace na daných objektech
    - Temporální dotazovací jazyky

# Temporální dotazovací jazyky

- Velké množství
- Nejčastěji založeny na SQL
- Typy
  - Relační
    - př.: HQL, HSQL, TDM, TQueI, TSQL, TSQL2
  - Objektově orientované
    - př.: MATISSE, OSQL, OQL, TMQL

# Shrnutí

- Ideální temporální datový model
  - Minimální rozšíření existujícího DM
  - Souvislá prezentace chování měnícího se v čase
  - Snadná implementace
  - Vysoký výkon
- Dosažení ideálu je prakticky nemožné

# Shrnutí

- Hlavní cíl temporálního DM by měl být zachytit sémantiku dat měnící se v čase
  - Ale často se dostává do pozadí
- Mnoho nekompatibilních datových modelů s mnoha dotazovacími jazyky



# TSQL2



# Obsah

- Čas v TSQL2
- Datový model
- Vytvoření relace
- Příkaz SELECT
- Modifikační příkazy
- Shrnutí



# TSQL2

- Temporal Structured Query Language 2
- Měl sjednotit přístupy k temporálním datovým modelům
- Nadmnožina SQL-92

# Pojetí času

- Časová osa TSQL2 je na obou koncích omezena, ale dostatečně daleko (18 miliard let)
- U časových údajů jsou možné různé granularity
- Časové typy
  - DATE, TIME, TIMESTAMP, INTERVAL, PERIOD

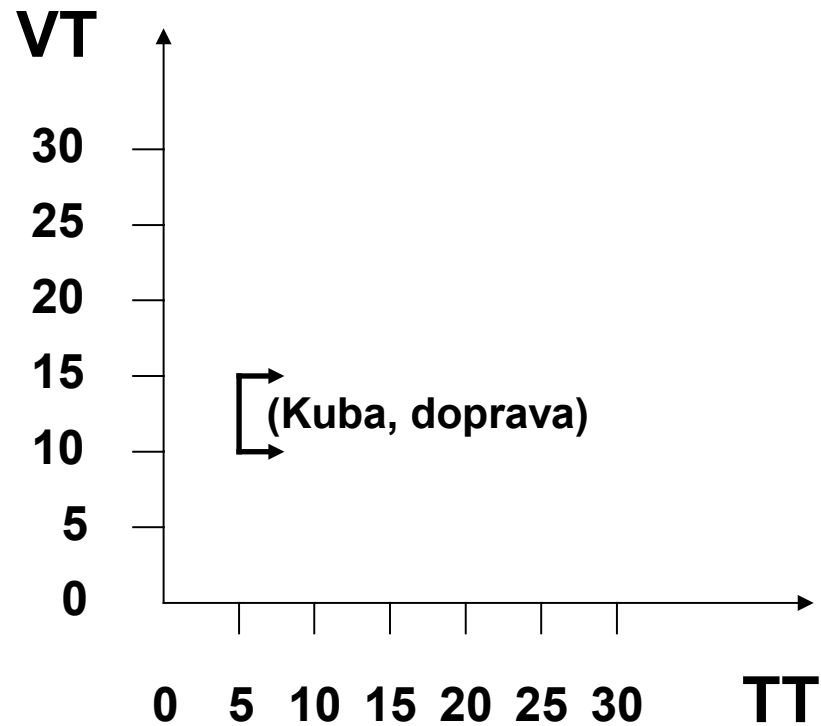
# Datový model

- Je použit BCDM (Bitemporal Conceptual Data Model)
- Řádek relace je orazítkován množinou bitemporálních chrononů
- Bitemporální chronon je dvojice (chronon transakčního času, chronon času platnosti)

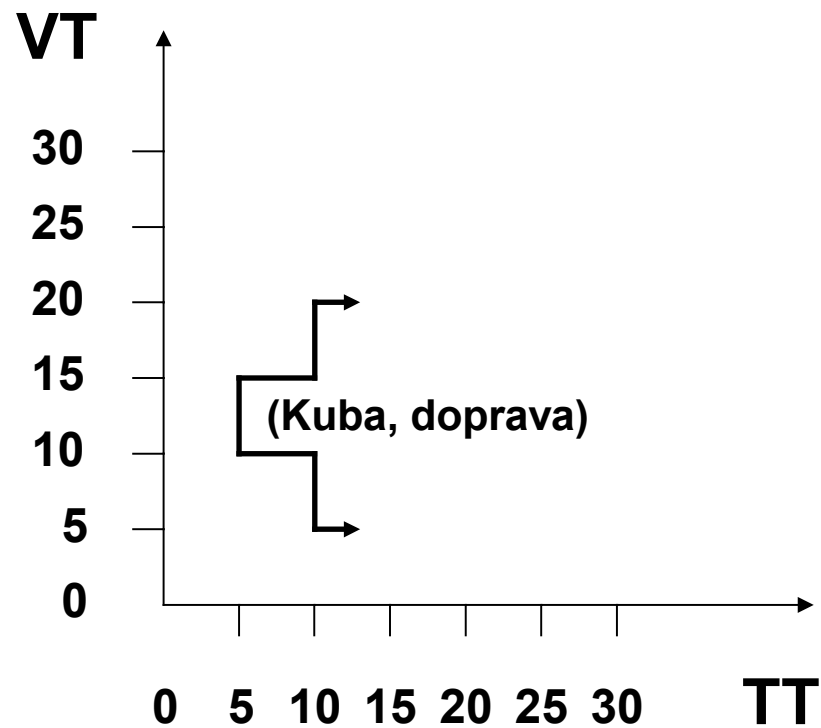
# Příklad bitemporální relace (1)

- Relace Zaměstnananec – umístění lidí v odděleních určitého podniku
- Schéma (Jméno, Oddělení) + časové razítko
- Předpokládejme granularitu 1 den u času platnosti i u transakčního času

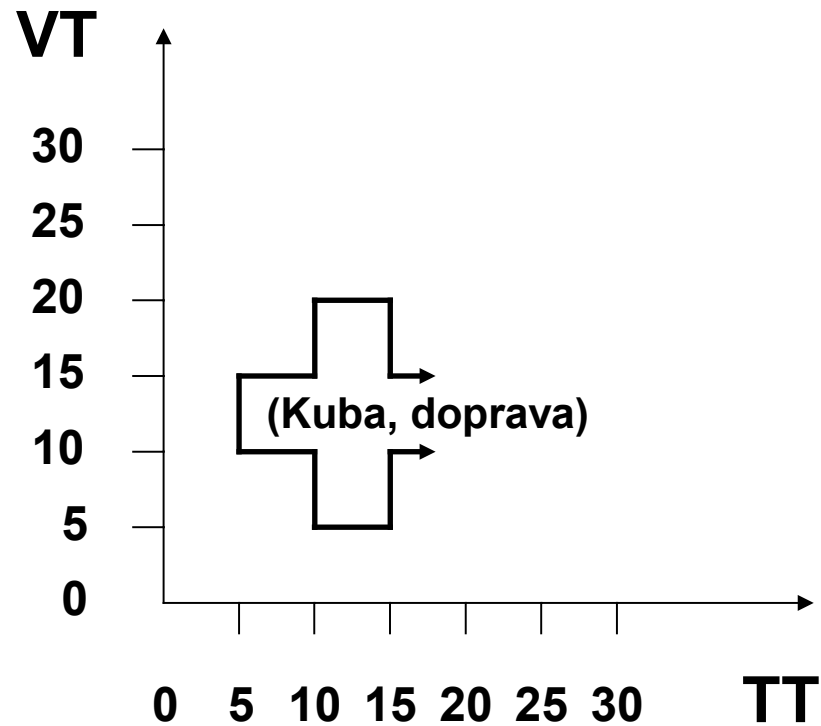
# Příklad bitemporální relace (2)



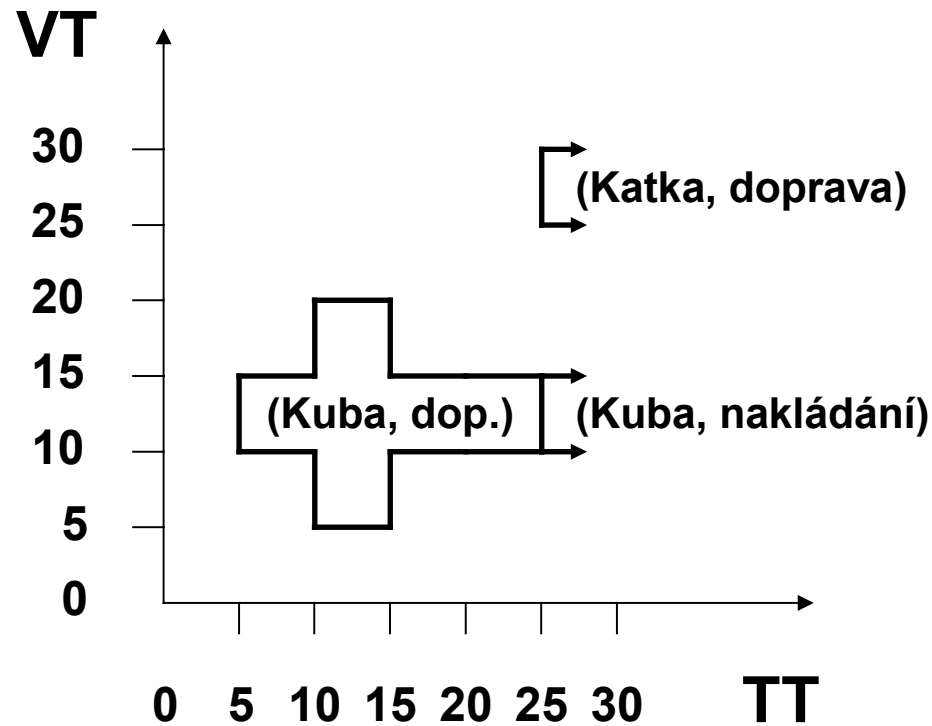
# Příklad bitemporální relace (3)



# Příklad bitemporální relace (4)



# Příklad bitemporální relace (5)





# Příklad bitemporální relace (6)

<i>Jméno</i>	<i>Oddělení</i>	<i>Časové razítko</i>
Kuba	Doprava	$\{(5, 10), \dots, (5, 15), \dots, (9, 10), \dots, (9, 15), (10, 5), \dots, (10, 20), \dots, (14, 5), \dots, (14, 20), (15, 10), \dots, (15, 15), \dots, (24, 10), \dots, (24, 15)\}$
Kuba	Nakládání	$\{(U.C., 10), \dots, (U.C., 15)\}$
Katka	Doprava	$\{(U.C., 25), \dots, (U.C., 30)\}$

# Definice schématu

## ■ Příklad

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

- Čas platnosti má granularitu 1 den
- Transakční čas má granularitu 1 ms nebo menší

# Příkaz SELECT

- Komu byl předepsán nějaký lék?

```
SELECT SNAPSHOT Jméno
```

```
FROM Předpis
```

- Kdo bral (bere) lék Proventil?

```
SELECT SNAPSHOT Jméno
```

```
FROM Předpis
```

```
WHERE Léč = 'Proventil'
```

# Příkaz SELECT

- Komu byl předepsán nějaký lék a kdy?

**SELECT** Jméno

**FROM** Předpis

# Příkaz SELECT

- Které léky byly užívány současně 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

- Na úrovni klauzule FROM se provede projekce a slévání
- Kdo měl (má) brát nějaký lék déle než 6 měsíců (v souhrnu)?

**SELECT** Jméno, Lék

**FROM** Předpis(Jméno, Lék) **AS** P

**WHERE CAST(VALID(P) AS INTERVAL  
MONTH) > INTERVAL '6' MONTH**

# Restrukturalizace

- Kdo po celou dobu léčby bere Proventil?  
**SELECT SNAPSHOT P1.Jméno**  
**FROM Předpis(Jméno) AS P1, P1(Lék) AS P2**  
**WHERE P2.Lék = 'Proventil' AND**  
**VALID(P2) = VALID(P1)**

# Partitioning

- Kdo měl předepsaný jeden lék déle než 6 měsíců nepřerušovaně?
  - **SELECT SNAPSHOT** Jméno, Lék, **VALID(P)**  
**FROM** Předpis(Jméno, Lék)(**PERIOD**) **AS P**  
**WHERE CAST(VALID(P) AS INTERVAL**  
**MONTH) > INTERVAL '6' MONTH;**



# Jde to i jinak

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

- **VALID(P)** lze užít v **SELECT** pouze při použití **PERIOD**

# Klauzule VALID

- Jaké léky měla Marie předepsané v roce 1996?

- **SELECT** Lék

- VALID INTERSECT (VALID(Předpis),  
PERIOD '[1996]' DAY)**

- FROM** Předpis

- WHERE** Jméno='Marie'



# Modifikace údajů v tabulce

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

# INSERT

- Vlož nový předpis.

- **INSERT INTO** Předpis

- VALUES** ( 'Marie', 'Dr. Beran', 'Proventil',  
'100mg', **INTERVAL** '8:00' **MINUTE**)

- Defaultní hodnota

- **VALID PERIOD(CURRENT\_TIMESTAMP,**  
**NOBIND(CURRENT\_TIMESTAMP))**

# INSERT

- Vlož nový předpis s omezenou dobou platnosti

- **INSERT INTO** Předpis

- VALUES** ( 'Marie', 'Dr. Beran', 'Proventil', '100mg', **INTERVAL** '8:00' **MINUTE**)

- VALID PERIOD** '[1996-01-01 – 1996-06-30]'

# DELETE

- Marie neměla v červnu 1996 předepsaný žádný lék.
  - **DELETE FROM** Předpis  
**WHERE** Jméno = 'Marie'  
**VALID PERIOD** '[1996-06-01 – 1996-06-30]'

# UPDATE

- Změň dávkování léku Proventil na 50 mg.

- **UPDATE** Předpis

- SET** Dávkování **TO** '50mg'

- WHERE** Jméno = 'Marie' **AND**

- Lék = 'Proventil'

# UPDATE

- Změň dávkování léku od března do května.

- **UPDATE** Předpis

- SET** Dávkování **TO** '50mg'

- VALID PERIOD** '[1996-03-01 – 1996-05-31]'

- WHERE** Jméno = 'Marie' **AND**

- Lék = 'Proventil'



# Zaznamenání událostí

- Definice tabulky laboratorních testů.
  - **CREATE TABLE** LabTest ( Jméno **CHAR(30)**, Lékař **CHAR(30)**, ČísloTestu **INTEGER**)  
**AS VALID EVENT HOUR AND TRANSACTION**  
( x **AS VALID STATE DAY AND TRANSACTION** )

# Restrukturalizace

- Který lékař objednával testy jedinému pacientovi a současně tento lékař byl jediný, kdo danému pacientovi testy objednával?

- **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)

# Transakční čas

- Jaké léky měla Marie předepsány?
  - **SELECT** Lék
  - FROM** Předpis
  - WHERE** Jméno = 'Marie'

# Transakční čas

- Jaké záznamy o lécích, které měla předepsány Marie, byly uvedeny v databázi 1. června 1996?

- **SELECT** Lék

- FROM** Předpis

- WHERE** Jméno = 'Marie' **AND**  
**TRANSACTION(P) OVERLAPS**

- DATE '1996-06-01'**

# Transakční čas

- Kdy byly naposledy změněny záznamy, které se vztahují ke dni 1. 6. 1996?
  - **SELECT SNAPSHOT BEGIN(TRANSACTION(P2))  
FROM Předpis AS P1, P2  
WHERE P1.Jméno = 'Marie' AND  
P2.Jméno = 'Marie' AND  
VALID(P1) OVERLAPS DATE '1996-06-01' AND  
VALID(P2) OVERLAPS DATE '1996-06-01' AND  
TRANSACTION(P1) MEETS TRANSACTION(P2)**

# Agregační funkce

- Kolik léků Marie brala?
  - **SELECT COUNT(\*)**  
**FROM Předpis**  
**WHERE Jméno = 'Marie'**
- Kolik lidí má předepsané jednotlivé léky?
  - **SELECT Lék, COUNT(\*)**  
**FROM Předpis**  
**GROUP BY Lék**

# RISING

- Jaké bylo nejdelší období, kdy dávka Proventilu pro Marie rostla?
  - **SELECT SNAPSHOT RISING(Dávka)**  
**FROM Předpis**  
**WHERE Jméno = 'Marie' AND**  
**Lék = 'Proventil'**

# Změna schématu tabulky

- 20. 8. 1996 přidáme k tabulce nový sloupec IdČíslo.

- **ALTER TABLE** Předpis

- ADD COLUMN** IdČíslo **INTEGER**

- Zobrazení dat ve starém schématu

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



# Převody času

- **CAST** vždy vrací přesný výsledek
- **SCALE** může vrátit „nepřesný“ výsledek

**CAST(TIMESTAMP '04-19-1996 15:24:00' AS DAY)**

**SCALE(TIMESTAMP '04-19-1996 15:24:00' AS DAY)**

U obou výsledek '04-19-1996'

**CAST(DAY '04-19-1996' AS SECOND)**

Výsledkem je 1. sekunda tohoto dne.

**SCALE(DAY '04-19-1996' AS SECOND)**

Výsledkem je nějaká sekunda tohoto dne.

# Shrnutí

- Výsledek bez podpory času **SELECT SNAPSHOT**
- **Restrukturalizace** – provede projekci na vybrané sloupce a provede slití času platnosti ekvivalentních řádků
- **Partitioning** – vybere nejdelší interval(y) času platnosti pro každou řádku