

13. TEMPORÁLNÍ DATABÁZE

Temporální databáze - vychází z relačních databází, které jsou rozšířeny o časovou dimenzi T
 - jsou velmi praktické, používají se v bankovních, lékařských, statistických IS, ...

Relační databáze - založena na relačním modelu

- $DTB = (R, R^*)$
 - relace se skládá ze schéma relace ($R = \{A_1:D_1, A_2:D_2, \dots, A_n:D_n\}$),
 kde jsou pojmenované atributy nad určitými doménami a těla relace
 $R^* \subseteq D_1 \times D_2 \times \dots \times D_n$, které je podmnožina kartézského součinu
 domén atributů - jsou to n-tice, které obsahují hodnoty pro
 všechny atributy kde tyto hodnoty jsou spolu v relaci
 - operace: přímě, sjednocení, relace, projekce, spojení (JOIN), ...

Čas

čas platnosti 1823 - 1902 x čas transakce - do DB vložené 2019 a
 čas platnosti - kdy něco platilo v daném čase v reálném světě, platnost daného
 čas transakce - fakt / stav reálného světa
 - časový okamžik - konkrétní časový údaj - bod na časové ose T_1 odebrání
 - časový interval - rozsah mezi dvěma časovými okamžiky T_1 T_2
 - doba trvání - více jak dlouho něco trvalo ale musíme vědět to začalo ani skončilo
 - časová doména - lineární uspořádání na množině časových okamžiků
 (lineární uspořádaná množina)
 - N, Z - diskrétní čas, Q - čas s hustotou, R - spojité čas
 - N, Z - diskrétní čas, Q - čas s hustotou, R - spojité čas
 - když se něco děje a chceme to uložit do DB
 - relace časového uspořádání

Modely času - dva jsou hlavně

• Snímkový model času (snapshot)

- databáze obsahuje snímky, kde snímek pro daný časový okamžik popisuje stav světa
- uspořádané snímky můžeme dát do času a vytvořit nějakou řadu / stav v čase
- je to potom soběstačné: $T \rightarrow DB(D, P)$ nebo také $T \rightarrow (R, R^*)$

Kde je daný časový okamžik soběstačný na daný stav DB v tom okamžiku

- není vhodný na dotazy typu - "dej mi všechny časy kdy platil nějaký stav"
- \Rightarrow musel by procházet všechny snímky a zjistit jestli je podmínka splněna

• Model času platnosti (časová ražítka)

- základním jednotkou je Chronon m -tice + čas $(t_0, t_1, \dots, t_n, t)$ (neefektivní)
- rozdělí m -tici relace rozšíření o časové ražítka (timestamp)
- čas je jen jedním z datových typů
- časové ražítka modelují stav DB po určitém časovém okamžiku, kde se po
 určité době vytvoří intervaly od kdy do kdy platil \Rightarrow intervaly jsou ale
- je to tedy realizovatelný dotaz typu všechny časy kdy | jen některé podmínky

- datové typy:

Smíšený model
- v daném okamžiku
nejedná o datovou platň
nebo ne

$$T \rightarrow (D^n \rightarrow \text{bool})$$

Model času platnosti

- pro každou stav a každý čas je
dané jestli to platí nebo ne

$$(T \times D^n) \rightarrow \text{bool}$$

- rozdíl !: ve smíšeném modelu máme vždy jen čas změny, tedy od kdy
platí nebo platil ten stav. X v modelu času platnosti máme
u každého stavu uvedeno pomocí časových směrů od kdy do kdy
platil a kdy kdy má práva od kdy do kdy platil novější stav, takže
se podíváme na jeho počátek a vidíme to. Jedná se o smíšený model
použití všech směrů a počítat se jestli ten stav platil nebo ne

Spojení času (modelu) s databází:

1) Smíšená tabulka

- smíšený model času
- reprezentován stavem dat v určitých časech - čarou se vyhraněn například pro
směru dat
- více jich je problém
- 1900 Aristotel 1995 Odborný Aristotel 2000 dozent → různé modely danou
upřesněním, ale čas
- historii

2) Tabulka času platnosti

- model času platnosti
- pro každý stav reprezentován jím danou platí - od kdy do kdy
- každé má hodnotu intervalu
- obnovení rozdílu
- rozšíření datových sloupců o sloupce času platnosti → $(x_0, x_1, \dots, x_n, t)$ m-li rozšíření
o časovou
dimenzi
- když se stane nějaká změna, tak se ten interval skončí na teď a začne se
nový počíná s novým stavem začínající od teď

3) Transakční tabulka

- obsahuje pouze časy transakcí, kdy byla transakce učiněna ve DB, kdy
byl ten stavem upraven a kdy odebran (deleted - at)
- ke sloupcům o daty navíc připojíte sloupce s časem ale ještě čas transakce

4) Tabulka obvyklá čas

- jít čas transakce (created-at, updated-at, deleted-at) tak čas platnosti

Techniky temporálnych databází

• Kódova'ní intervalu

- máme trieba časovú rovnicu a chceme mať plnú množinu faktov či istinu dat dánu jako interval a tak tento interval je nutné rozšírovať
- máme časovú doménu T a intervalovú časovú doménu nad touto doménou je dána jako $T_I = \{T(I)_1, \dots, T(I)_n\}$ kde $T(I)$ je množina intervalu
$$T(I) = \{[a, b) \mid a \in T \cup \{-\infty\} \wedge b \in T \cup \{+\infty\} \wedge a \leq b\}$$

 $a < -\infty$ atď. pre operácie dotazu interval sa číta interval - je to prebiehajúce atď.

• Shluková'ní

- temporálnu DB dosahuje jasnou štruktúrou pre každý fakt platí že je platný v intervaloch ktoré sa neprekrývajú a navyše je to spoľahlivé
- $(a_1, [0, 2])$ a $(a_1, [1, 4]) \Rightarrow (a_1, [0, 4])$ - musí to byť pre ten fakt a_1 - nemáme šlukovať dva rôzne faktory
- každý je potrebný spojiť fakt pre danú operáciu
- jednoduchšie je to relatívne operácie jako selekcia (máme min. množinu toho čo šlukujeme tak je dôležité jednoduchšie), projekcia a spojenie (JOIN - case min. množina toho čo šlukujeme je join)
- temporálnu databázu to jednoduchšie
 \Rightarrow shluková'ní je ale nevyhnutné pre viac rozmanitosti

↳ Generičnosť dotazu

- dotaz je generický jestliže dáva rovnaké výsledky nezávisle na tom, jakým časom intervalom v databázi keď napíšeme jestliže je fakt šlukovaný pre časovú doménu
- pre $(a_1, [0, 2]), (a_1, [1, 4])$ musí byť generický dotaz rovnaké výsledky jako pre $(a_1, [0, 4])$ - at' to je čas intervalu tak či onak tak rovnaké

- tento musí byť generický $\exists i, j: \exists x (R(i, x) \wedge R(j, x) \wedge i \neq j)$ - je to fakt x je v relácii i a j a x je v relácii j a i (výsledky sú generické)
- shluková'ní musí byť generický dotaz - fakt se dotazuje vždy (výsledky sú generické)

• Rozdeľova'ní

- celá shlukovaná databáza je pre všetky dotazy generická (všetky podmienky musia byť generické)
- opäť shluková'ní - intervaly sa rozdeľujú - jak majú byť rozložené a rozdeľujú? - problém
- $(a_1, [0, 4]) \rightarrow (a_1, [0, 2])$ a $(a_1, [1, 4])$ alebo takto ma
 $(a_1, [0, 2])$ a $(a_1, [2, 4])$ atď..

• Odsa'va'ní dat

- čím je väčší počet dat, tým je ťažšie a potrebné dôležité postupy
- v relatívnych se datu upraviť, aby je rovná a tým vznikne 2 rovnice rovné 1 rovnici. ale tak je ťažšie, má a takto to pôsobí komplexne
- musí sa byť rovná, rovná sa data

• Indexová'ní

- pomocou R množiny a AP množiny
- musíme indexovať aby operácie ktoré majú prebiehať se intervaly mohli nájsť (3)

Integritni' omezení

- jsou to určité formule 1. stupeň temporálního datového jazyka!
- chceme třeba zhlédnout jen nejnovější data
- každý seřazený DB aplikace - třeba DB aplikace je HR systém firmy a temporální 'omezení' bude že nemáme být přijat nový zaměstnanec dřív než než je firma schopná ať už dříve než za určitou dobu (nemáme peníze)
- historie splnění 10 jistě je to 10 pravidel (splnění) v každém stavu historie
- ~~historie~~ konečná historie (odněkdy do někdy) potenciálně splnění 10 jistě může být rozšířena na nekonečno a bude pak splňovat to IO

↳ tedy nekonečná historie splnění jistě její stav splnění

↳ konečná historie jen potenciálně splnění jistě může být rozšířena na nekonečno a pokud bude splňovat

Granularita - jak má být rozděl jen na nejmenší prvky, tedy obdržíme nějaké data události a prvky na roky, měsíce, dny, hodiny, ...

Dotazovací jazyky

- nejsou standardizované a tak se místo nich používá třeba TSQL2 což je rozšíření SQL o časovou dimenzi - množina $\{x_0, x_1, \dots, x_n, t\}$ čas a jsou to tedy relace obdrží se rozšířením o podmínku na čas - vyjádření temporálního relace bude obsahovat nějaké minuty, s časem platnosti, s časem události, s časem koncese a další...

↳ spíše k těm standardizovaným:

Temporální výroková logika

- klasická výroková logika rozšířená o časové spj. proměnné a kvantifikátory

$\square x$ - někdy v budoucnu x platí

$\lozenge x$ - někdy v minulosti

$\Box x$ - vždy v budoucnu

$\blacksquare x$ - vždy v minulosti

$\circ x$ - v kointerpretačním bodu

$\bullet x$ - v předchozím bodu > diskretní čas

UNTIL - $X1$ until $X2$

- platí $X1$ dokud někdy v budoucnu nebude platit $X2$

- $\overbrace{X1}^{X2}$ $X2$ když někdy bude jistě platit $X1$
↑ když jsem teď

SINCE - $X1$ since $X2$

- to $X1$ platí až od doby co bylo $X2$

$\overbrace{X1}^{X2}$ $X2$ až od doby kdy bylo $X2$
↑ od této doby platí

$X1$ a $X2$ jsou formule

Př.: určitelné lide' x kteří byli přijati znovu do stejné společnosti

$\exists y: ZAH(x, y) \wedge \lozenge (ZAH(x, y) \wedge \blacksquare ZAH(x, y))$ > tedy je ale někdy v minulosti platilo že se znovu našel ať už kdy platilo že někdy bylo