

Zkoušková práce na Databázové systémy - Sebastian Uhlík (uhlikse)

Opravené relační schéma

Parrent(**ParrentID**, Name, Surname, Phone, Email, Address, Passwd)

Child(**ChildID**, **ParrentID**, Name, Surname, Phone, Email, Address, Birth, Allergies)

Child.ChildID \subset Parrent.ParrentID

Event(**EventID**, Type, Name, Begins, Ends, Capacity)

Payment(**PaymentID**, **ParrentID**, Due, Date, Type)

Payment.PaymentID \subset Parrent.ParrentID

User(**UserID**, Email, Passwd)

Admin(**AdminID**, Email, Passwd)

Accountant(**AccountantID**, Email, Passwd)

GoesTo(**ChildID**, **EventID**)

GoesTo.ChildID \subset Child.ChildID

GoesTo.EventID \subset Event.EventID

PaidForWhom(**PaymentID**, **ChildID**)

PaidForWhom.PaymentID \subset Payment.PaymentID

PaidForWhom.ChildID \subset Child.ChildID

PaidForWhat(**PaymentID**, **EventID**)

PaidForWhat.PaymentID \subset Payment.PaymentID

PaidForWhat.EventID \subset Event.EventID

1. NÁVRH RELAČNÍCH SCHÉMAT

Vezměte relace odpovídající třem třídám Vašeho logického relačního modelu (úloha 1.2 zápočtového testu), které jsou propojené M:N a 1:N vztahem.

V případě existence více tříd tři vhodné vyberte a ostatní ignorujte.

Parrent(PID,PN,PS,PD,PE,PA,Pass)

Child(CHID,PID,ChN,ChS,ChP,ChE,ChA,ChB,CHAI)

Event(EID,ET,EN,EB,EE,EC)

- Vložte atributy všech tří relací do jediné univerzální relace. Pokud je „obyčejných“, nic neurčujících atributů v relaci hodně, stačí dva nejdůležitější.

Uni(PID,PN,PS,PE,CHID,ChN,ChS,EID,ET,EN,EB,EE,EC)

- Doplněte univerzální relaci o všechny závislosti:

- klíče původních relací určují vše v dané relaci

PID->PN,PS,PE

CHID->PID,ChN,ChS

EID->ET,EN,EB,EE,EC

- unikátní sloupce původních relací určují klíč

PE->PID

EN->EID

EB->EID

- případné další závislosti mezi atributy.

EB,ET->EE

ET->EC

EC->ET

- Určete redundantní atributy a závislosti, minimální pokrytí a všechny klíče.

1) \forall na elementární

PID->PN

EID->EE

PID->PS

EID->EC

PID->PE

PE->PID

CHID->PID

EN->EID

CHID->ChN

EB->EID

CHID->ChS

EB,ET->EE

EID->ET

ET->EC

EID->EN

EC->ET

EID->EB

2) redukce levých stran

Předp. V EB,ET->EE je ET redundantní

$$EB^+ = \{EID, ET, EN, EB, EE, EC\}$$

EE leží v at. Uzávěru EB, tedy ET je redundantní.

3) odstranění redundantních pravidel

PID->PN – není redundantní

$$PID^+ = \{PID, PS, PE\}$$

PID->PS – není redundantní

$$PID^+ = \{PID, PN, PE\}$$

PID->PE – není redundantní

$$PID^+ = \{PID, PN, PS\}$$

CHID->PID – není redundantní

$$CHID^+ = \{CHID, ChN, ChS\}$$

CHID->ChN – není redundantní

$$CHID^+ = \{CHID, PID, ChS, PN, PS, PE\}$$

CHID->ChS – není redundantní

$$CHID^+ = \{CHID, PID, ChN, PN, PS, PE\}$$

EID->ET – je redundantní

$$EID^+ = \{EID, EN, EB, EE, EC, ET\}$$

EID->EN – není redundantní

$$EID^+ = \{EID, ET, EB, EE, EC\}$$

EID->EB – není redundantní

$$EID^+ = \{EID, ET, EN, EE, EC\}$$

EID->EE – je redundantní

$$EID^+ = \{EID, ET, EN, EB, EC, EE\}$$

EID->EC – není redundantní

$$EID^+ = \{EID, EN, EB, EE\}$$

PE->PID – není redundantní

$$PE^+ = \{PE\}$$

EN->EID – není redundantní

$$EN^+ = \{EN\}$$

EB->EID – není redundantní

$$EB^+ = \{EB, EE\}$$

EB->EE – není redundantní

$$EB^+ = \{EB, EN\}$$

ET->EC – není redundantní

$$ET^+ = \{ET\}$$

EC->ET – není redundantní

$$EC^+ = \{EC\}$$

Minimální pokrytí:

PID->PN,PS,PE

CHID->PID,ChN,ChS

EID-> EN,EB,EC

PE->PID

EN->EID

EB->EID

EB->EE

ET->EC

EC->ET

Hledání klíče:

$(PID, PN, PS, PE, CHID, ChN, ChS, EID, ET, EN, EB, EE, EC)^+$
= {PID, PN, PS, PE, CHID, ChN, ChS, EID, ET, EN, EB, EE, EC}

$(PN, PS, PE, CHID, ChN, ChS, EID, ET, EN, EB, EE, EC)^+$
= {PN, PS, PE, CHID, ChN, ChS, EID, ET, EN, EB, EE, EC}

$(PS, PE, CHID, ChN, ChS, EID, ET, EN, EB, EE, EC)^+$
= {PID, PN, PS, PE, CHID, ChN, ChS, EID, ET, EN, EB, EE, EC}

$(PE, CHID, ChN, ChS, EID, ET, EN, EB, EE, EC)^+$
= {PID, PN, PS, PE, CHID, ChN, ChS, EID, ET, EN, EB, EE, EC}

$(CHID, ChN, ChS, EID, ET, EN, EB, EE, EC)^+$
= {PID, PN, PS, PE, CHID, ChN, ChS, EID, ET, EN, EB, EE, EC}

$(CHID, ChS, EID, ET, EN, EB, EE, EC)^+$
= {PID, PN, PS, PE, CHID, ChN, ChS, EID, ET, EN, EB, EE, EC}

$(CHID, ChS, EID, ET, EN, EB, EE, EC)^+$
= {PID, PN, PS, PE, CHID, ChN, ChS, EID, ET, EN, EB, EE, EC}

$(CHID, EID, ET, EN, EB, EE, EC)^+$
= {PID, PN, PS, PE, CHID, ChN, ChS, EID, ET, EN, EB, EE, EC}

$(CHID, ET, EN, EB, EE, EC)^+$
= {PID, PN, PS, PE, CHID, ChN, ChS, EID, ET, EN, EB, EE, EC}

$(CHID, EN, EB, EE, EC)^+$
= {PID, PN, PS, PE, CHID, ChN, ChS, EID, ET, EN, EB, EE, EC}

$(CHID, EN, EE, EC)^+$
= {PID, PN, PS, PE, CHID, ChN, ChS, EID, ET, EN, EB, EE, EC}

$$(\text{CHID}, \text{EN}, \text{EC})^+ = \{\text{PID}, \text{PN}, \text{PS}, \text{PE}, \text{CHID}, \text{ChN}, \text{ChS}, \text{EID}, \text{ET}, \text{EN}, \text{EB}, \text{EE}, \text{EC}\}$$

$$(\text{CHID}, \text{EN})^+ = \{\text{PID}, \text{PN}, \text{PS}, \text{PE}, \text{CHID}, \text{ChN}, \text{ChS}, \text{EID}, \text{ET}, \text{EN}, \text{EB}, \text{EE}, \text{EC}\}$$

(CHID, EN) je klíč

$$K' = (K \setminus Y) \cup X$$

$$\text{EID} \rightarrow \text{EN} \Rightarrow (\text{CHID}, \text{EID})$$

$$\text{EN} \rightarrow \text{EID} \Rightarrow (\text{CHID}, \text{EN})$$

$$\text{EB} \rightarrow \text{EID} \Rightarrow (\text{CHID}, \text{EB})$$

$$\text{EID} \rightarrow \text{EB} \Rightarrow (\text{CHID}, \text{EID})$$

$\{(\text{CHID}, \text{EN}), (\text{CHID}, \text{EID}), (\text{CHID}, \text{EB}), (\text{CHID}, \text{EID})\}$ jsou všechny klíče

2. DOTAZOVACÍ FORMALISMY

Nad Vámi vytvořenými relacemi (úloha 1.2 zápočtového testu) zformulujte **Dotaz 1** (úloha 2 zápočtového testu) v doménovém relačním kalkulu.

„Vyber všechny děti Petra Horkého.“

```
SELECT Ch.Name, Ch.Surname, Ch.Birth FROM Child AS Ch, Parent AS P WHERE  
P.Name = 'Petr' and P.Surname = 'Horký' and P.ParentID = Ch.ParentID ORDER  
BY Ch.Birth, Ch.Name
```

Child budiž $Ch(ChID, PID, ChP, ChE, ChAd, ChAl)$

Parent budiž $P(PID, PN, PS, PP, PE, PAd, PPass)$

$$\begin{aligned} & \{(ChN, ChS, ChB) | \\ & \exists ChID, PID, ChP, ChE, ChAd, ChAl, PN, PS, PP, PE, PA, PPass: \\ & \quad (Ch(ChID, PID, ChP, ChE, ChAd, ChAl) \\ & \quad \& P(PID, PN, PS, PP, PE, PAd, PPass) \\ & \quad \& PN = \text{"Petr"} \& PS = \text{"Horký"})\} \end{aligned}$$

Nad Vámi vytvořenými tabulkami (úloha 1.2 zápočtového testu) zformulujte **Dotaz 2** (úloha 2 zápočtového testu) v doménové relační algebře.

(Nový dotaz, ten v zápočtu byl špatně...)

„Vyber všechny rodiče, jejichž děti mají nějakou alergii.“

```
SELECT P.Name, P.Surname FROM Parent as P WHERE NOT EXISTS ( SELECT * FROM  
Child as Ch WHERE Ch.ParentID = P.ParentID and Ch.Alergies IS NULL )
```

Child budiž $Ch(ChID, PID, ChP, ChE, ChAd, ChAl)$

Parent budiž $P(PID, PN, PS, PP, PE, PAd, PPass)$

$$\begin{aligned} & P[PS, PN, PE] \setminus (P * Ch)(Ch.PID = P.PID \& Ch.Al = NULL) \\ & \quad [P.PS, P.PN, P.PE] \end{aligned}$$

3. TRANSAKCE

Z následujících dvou tabulek vyberte dvě transakce na základě svého **unikátního čísla studenta**: Z první tabulky vyberte řádku transakce T_1 , která odpovídá **první cifře**. Ze druhé tabulky vyberte řádku transakce T_2 , které odpovídá **poslední cifře**.

Vytvořte rozvrh obou transakcí tak, že operace proložíte mezi sebe. Vždy operace z T_1 následovanou operací z T_2 atd.

K Vámi vytvořeném rozvrhu:

- Nakreslete precedenční graf.

Rozvrh k číslu 5XXXXXX4:

T_1	T_2
R(B)	
	R(B)
W(A)	
	R(B)
W(A)	
	R(A)
R(C)	
	R(C)
R(D)	
	W(E)
COMMIT	
	COMMIT

- Rozhodněte, zda je konfliktově uspořádatelný (conflict serializable)

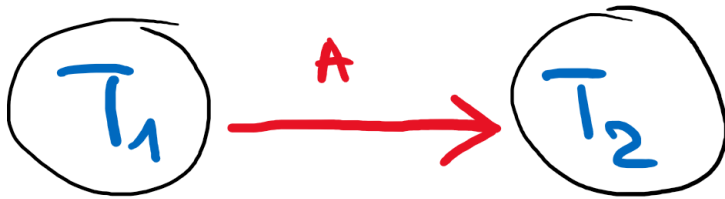
Když si zjednodušíme precedenční graf pouze na položky T_1 , T_2 dostaneme:



Což je orientovaný acyklický graf a proto víme, že toto naplánování transakcí je konfliktově uspořádatelné.

- Rozhodněte, zda je zotavitelný (recoverable) a proč

Víme, že existuje graf závislostí:



Transakce T2 je závislá na transakci T1, která je korektně ukončena před koncem transakce T2. Z toho vyplývá, že toto naplánování je zotavitelné.

- Pokud není zotavitelný, upravte jej, aby byl a přitom se nezměnilo pořadí čtení/zápisů

Je zotavitelný.

4. IMPLEMENTACE DB STRUKTUR

Rozhodněte, které (nejvýše tři) sloupce Vašeho relačního modelu (úloha 1.2 zápočtového testu) by bylo nejvhodnější opatřit indexem, založeným na B-Stromu a proč. Sloupce deklarované jako primární klíče a unikátní sloupce neuvažujte, ty se indexují automaticky.

- 1) Parrent.Email – tento údaj rodiče používají pro své přihlášení, bude tedy používán velice často
- 2) Child.ParrentID – tento údaj je velice užitečný pro zjištění všech dětí konkrétního rodiče (což je jedna z hlavních funkcí, která se volá po tom co se rodič přihlásí)