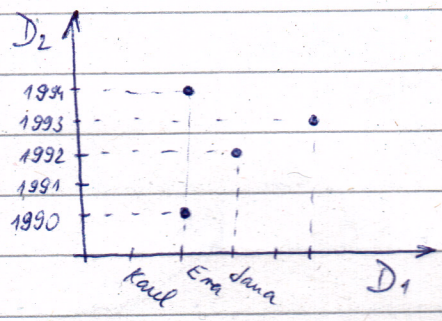


RELACNI MODEL

- model na rozdíl od ERA m. modelujeme uložení dat + PC
- relační - model dat založen na matematic. pojmu relace
- tab. součin, zkratkou do, na, ...
- CODD
- TODD
- relační MD synonyma s 'řový' MD
- dnes tedy v SRBD najdeme jen relační

Definice řeči je daný systém $\{D_i, i \in [1, n]\}$ množin (domén). Potom podmnožinou kartézského součinu $R \subset D_1 \times D_2 \times \dots \times D_n$ nazýváme relaci skupin n nad D_1, D_2, \dots, D_n . Prvky relace R jsou uspořádané n -tice d_1, d_2, \dots, d_n takové, že d_i je prvkem $D_i, i \in [1, n]$.



$D_1 \dots$ jména
 $D_2 \dots$ rok narození

- nejvh. datovou strukturou, jak zobrazit relaci, je tabulka
- tabulka má n ^{pojmenovaných} sloupců (odpovídá atributů)
- relace je repr. tabulkou
- sloupce jsou homogenní
- v tabulce se nemusí opakovat řádky, protože je popisujeme jen jednou

- pořadí řádků je nevýznamné
- pořadí sloupců také nevýznamné, ale musíme pokračovat i s názvem
- možná přípustných hodnot daného atributu
- atribut je název sloupce

Pohled na data

souborové orientovaný	E-R-A model	Datová struktura	Relační schéma
soubor	entita možná	tabulka	relace
názvem	entita	řádek	uspořádaná n-tice
položka	atribut	sloupec (název)	název

Relační schéma

- název relace s uspořádanými atributy

$R(A_1, A_2, \dots, A_n)$

Pr:

Student (číslo-studanta, jméno, příjmení, adresa)

- obvykle atribut nebo množ. atributů, kl. sloví
klíč, podklíčové

Pr:

Značka 1 (značka, ID-student, ekrasna předměta,
pokus, datum)

Značka 2 (značka, ID-student, zk. předměta,
↓
pokus, datum)

možné např. i 4ky s datem, kdy
je dostal

Relační schéma databáze

- dvojice (R, I)

R ... množina všech relačních schémat

I ... množina všech integritních omezení

- problém je v efektivním popisu I

Relační algebra

- poskytuje základ, jak pracovat s relacemi

- operace RA :

- o přímik, sjednocení

- o množinový rozdíl

- o kartézský součin

- myšleli bychom stejné typy atributů
(tj. stejné atributy)

$R(A, B)$

R :	a	1
	b	2

$S(C, D)$

S :	A	α
	B	β

$R \times S$

A B C D

a 1 A α

a 1 B β

b 2 A α

b 2 B β

- byly definovány další 3 operace:

- o projekce

- o selekce

- o spojení (3 typy spojení)

Projekce relace R s atributy A na množinu B , kde $B \subseteq A$.

Operace vytvoří relaci se schématem B a prvky, kterým vsmíknou s přírodní relace odstraněním hodnot atributu $A - B$.

Odstranění jsou případy duplicity prvků (řádků).

Selektce relace R s atributy A podle logické podmínky Φ vytvoří relaci s týmž schématem (stejně atributy (doména) jako R) a ponechá ty prvky přírodní relace, kterým splňují podmínku Φ . Formule Φ je Boolský výraz, jehož atomické formule mají tvar

$$t_1 \theta t_2,$$

kde $\theta \in (<, \leq, =, \geq, >)$, t_1 je buď konstanta nebo jméno atributu.

Značení: $R(\theta)$.

Spojení relací R a S se schématy (atributy) A , resp. B .

Operace vytvoří relaci se schématem $A \cup B$, jejíž projekce na A je relace R a projekce na B je relace S .

Značení: $R * S$

Spojení $R [A \oplus B] S$

$$\{ (r \parallel s); r \in R, s \in S \wedge (r(A) \oplus s(B)) \}$$

a) Ⓜ - spojění (Abi'la spojění)

$R(M, P, Q, T)$ $S(A, B)$ $R[Q > A]S$

Prvky R :

x	101	5	a
y	103	3	a
z	500	9	a
w	30	1	b
v	10	2	b
m	300	4	b

S :

5	a
10	b
15	c
2	d
6	a
1	b
5	b

$R * S$

x	101	5	a	2	d
x	100	5	a	1	b
y	103	3	a	2	d
y	103	3	a	1	b
z	500	9	a	5	a

b) přirození spojění

- nalze se maximální množství shodujících abikudů $\approx R$ a S

Pr: Výrobek (č. nje, cena)

Zahoby (č. skladu, č. nje, množství)

Výrobek * Zahoby (přes č. nje, jiný abikud není shodný)

Výrobek	TX10	25	Zahoby	003	TX10	5000
	LE38	50		003	LE38	150
	DP35			003	PS64	4
	DP35	5000		004	DP35	29
	YL101	10000		003	DP85	8

- množství shodujících abikudů se uvede jen jednou

TK10	25	003	5000
LE38	50	003	150

DP35	5000	003	8
DP35	5000	004	19

a) kompozice

- jako pětice spojená, ale jako množina shodných a nevede vidět
- pokud nebude uvedeno jinak, budeme považovat ad b) pětice spojené
- musí být shodná
- shoda spojená musí ~~rozložit~~ musí 2 atributy a stejnou dominantou (oproti jin.)
- některé operace lze vyjádřit pomocí jiných operací
(přínik - pom. množ. rozdílů spojená - kartézský součin)
- lze definovat minimální množ. operací
(ji. sjednocení, kart. součin, množinový rozdíl, relace, projekce)

EXERCISE (1980, a. nakop, č. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20)

Výsledkem operací nad relacemi je opět relace

musí být tabulka s 1 sloupcem a 1 řádkem

Příklad: Mějme relaci **EXEMPLAR** (ISBN, inv.č.,
d-nákupe, cena, země-vyd.), **VYDÁNÍ** (inv.č., ...)

Dobaz: zapíše příkaz pomocí relací algebry,
klíč má být ISBN všech nypřijemých anglických
knih

$(EXEMPLAR \text{ (Země-vyd = GB) (ISBN, inv.č.)} * VYDÁNÍ) (ISBN)$

Příklad:

STUDENT

 $\xrightarrow{1}$ $\xrightarrow{2}$

ZNAŤKA

Student (ID student, jméno, příj., adresa, rok narození)

Značka (zkratka-předmětu, datum, známka, počet, ID student)

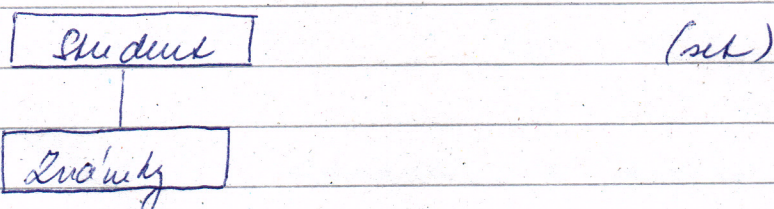
Úkol: Vypis jména studentů, kteří mají > DB1 za 1.

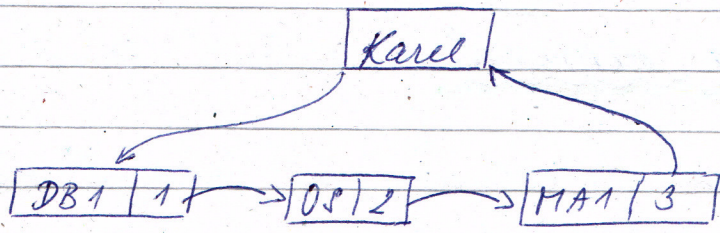
$(ZNAŤKA \text{ (zkr-předmětu = 'DB1' a známka = 1)} * \text{STUDENT}) [\text{jméno}]$

$(ZNAŤKA * \text{STUDENT}) \text{ (zkr-předm. = DB1 a známka = 1)} [\text{jméno}]$

ŘÍTOVÝ MODEL DAT

- předcházel relačním MD
- vycházel ze podobnosti - orient. pohledu na MD
- 1941 - norma DBTG CODASYL
- princip: práce se se sázováním a
spojkov (ukazatel na sázovani)
- pojmy SET (C-mnozina) - obecný
VYSKYT SETU - konkrétní





nýšky k selu

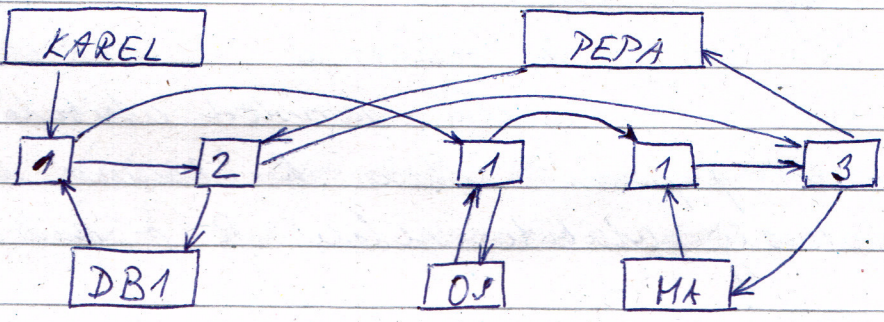
Terminologie:

členky ... ČLEN ... MEMBER
 student ... VLASTNÍK ... OWNER

automatická
 manuální (založíme
 to my nebo on sám)

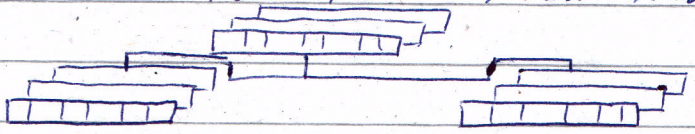
potrava (může být
 nepotraviná list. bez sladk.)

- musí existovat vazba M:N



Jeden člen může být na dvou různých úrovních.

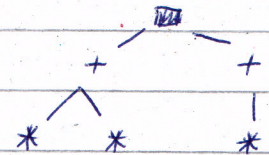
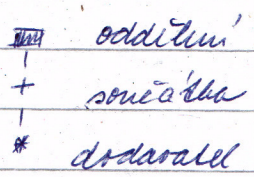
HIERARCHICKÉ SYSTÉMY - nosí stranu



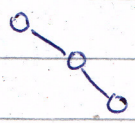
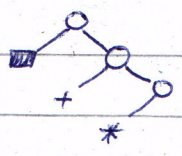
- dva přístupy k implementaci stromu
 - informace byla jen v listech
 - " - všech úrovních stromu

- způsob implementace stromu

a) v listech



b) v listech



5

FUNKČNÍ ZÁVISLOST ATRIBUTŮ

UNIVERZÁLNÍ SCHEMÁ = relace, kt. obsahuje všechny atributy, kt. se vyskytují

Kyška (předseda, učitel, mistrnost, medicína, student, známka)

R' {PU, HMP, PSZ, HPM}

R'' {PU, HSP, PSZ, HPM}

R {PSZ, PSUMM}

- rekordy lepší - neobsahuje duplicitu, a dává to co očekáváme

→ funkční závislost atributů

POPULACE = populaci relace R s určitými rozumnými hodnoty atributů relace R v daném čase

A, B ... atributy relace R

B ... atribut jeví závisí na atributu A, jestliže

✓ populace relace R platí pro libovolná

n - lici $w, r \in R$

$$w.A = r.A \Rightarrow w.B = r.B$$

a značíme $A \rightarrow B$

	A	B
$w \rightarrow$	•	•
$r \rightarrow$	•	•

↑

kyž byle hodnoty jsou stejné na straně A, pak budou i r B

$R \cong \mathbb{Z}[x]$ (přednáška, učitel, mírnost, hod, známka)

$P \rightarrow U$

$HM \rightarrow P$

$HU \rightarrow M$

(pouze pokud bychom přednášeli učitel)

učitel závislý na přednášce

EP-130, úk. 10.15 \rightarrow DB1

Pozn. : V každé hodnotě atributu A existuje
nejvýše 1 hodnota atributu B

Pozn. : Ve funkcií závislosti $A \rightarrow B$ je A
nezbytně determinant.

Ž jedinečné dané populace minimálně učitel není
závislost, ale můžeme učitel, je není závislost
nplati.

Pomocí funkcií závislosti lze definovat klíč:

Př. $R(\Omega)$ Ω ... množina atributů

a K ; $K \subseteq \Omega$... klíč

potom K je klíčová schéma R, jestliže

a) všechny atributy funkce závisí na K

b) existuje $\exists K'$ takové, že $K' \subset K$, že

by platilo $K' \rightarrow \Omega$

Pozn. : Všechny atributy A závisí na klíči.

Existuje právě jedna podm. K, kt. má tu vlastnost,

že na ni závisí všechny atributy.

ARMSTRONGOVA PRAVIDLA

1) TRIVIALNÍ FUNKČNÍ ZÁVISLOST

jestliže je dána R ^{relace} ~~na~~ Ω , X, Y atd.

$$R(\Omega), X, Y \subset \Omega$$

$$Y \subseteq X \Rightarrow X \rightarrow Y$$

Podm. atributů je závislá na té přírodní msn.

2) TRANZITIVITA

$$R(\Omega), X, Y, Z \subset \Omega$$

$$X \rightarrow Y \wedge Y \rightarrow Z \Rightarrow X \rightarrow Z$$

3) KOMPOZICE

$$R(\Omega), X, Y, Z \subset \Omega$$

$$X \rightarrow Y \wedge X \rightarrow Z \Rightarrow X \rightarrow YZ$$

Form. AP nemusí být v tomto směru, ale musí splňovat tyto vlastnosti:

- **konkurence** - co pomocí nich odvodíme, to platí
- **vplnost** - pomocí nich lze odvodit všechny tři závislosti, které samy platí
- **nezávislost** - ani jedno z těch pravidel nepustit (pominutí by bylo vplnost)

DALŠÍ ODVODITELNÁ PRAVIDLA

- **doplňitelnost**

$$X \rightarrow Y \wedge X \subseteq W \Rightarrow W \rightarrow Y$$

Důkaz: ad 1) $X \subseteq W \Rightarrow W \rightarrow X$

ad 2) $W \rightarrow X \wedge X \rightarrow Y \Rightarrow W \rightarrow X$
přidání

• aditivnost kvadransu (spojitáii)

$$X \rightarrow Y \wedge W \rightarrow Z \Rightarrow XW \rightarrow YZ$$

ad ① $XW \rightarrow X$ předpoklad $X \rightarrow Y \Rightarrow XW \rightarrow Y$
 $XW \rightarrow W$ $W \rightarrow Z$ ② $\Rightarrow XW \rightarrow Z$

\approx ③ $XW \rightarrow Y \wedge XW \rightarrow Z \Rightarrow XW \rightarrow YZ$

• projekce (dekompozice)

$$X \rightarrow YZ \Rightarrow X \rightarrow Y \text{ a } X \rightarrow Z$$

• součet

$$X \rightarrow Y \wedge YW \rightarrow Z \Rightarrow XW \rightarrow Z$$

Použili A.P.

- úkol: hledání rozkladu unit. schématu

1) určit množinu $F+$

množina všech fúních závislostí odvozených

$\approx F$ (NP problem)

2) Elementární fúních závislost

- na pravé straně je pouze jeden atribut

3) pokrytí

- množina G (nm. jinych fúních závislostí)

$\approx G+$ je stejný jako $F+$

4) kanonické pokrytí

$$F'+ = F+$$

F' ... množina element. fúních závislostí

5) $f \in F$ je redukovaný v F

$$(F - \{f\})+ = F+$$

6) neredukovaný pokrytí

- určuje rozklad unit. schématu

Pr. $HM \rightarrow U$, pak relace $R_1 (H, M, U)$

to, co je determinované, je klíčem

Rozklad není jednoznačný, záleží na tom, jak odobráme f .

Pr. nalezneme klíč

Některé možnosti atributů X + rozkladu $\in F$ (min. závis.) je možná všech atributů není závislost na X .

$R(W, X, Y, Z)$

$F: W \rightarrow Z$

$YZ \rightarrow X$

$WZ \rightarrow Y$

$(WZ)^+ = \{X, Y, Z, W\}$

je WZ klíčem?

$WZ \rightarrow Y$ ~~ad 1~~ ... dáno

$WZ \rightarrow Z$ ad 1,

$WZ \rightarrow W$ ad 1,

$WZ \rightarrow Y$ } ad 3,

$WZ \rightarrow Z$ } $\Rightarrow WZ \rightarrow YZ \wedge YZ \rightarrow X \Rightarrow \underline{\underline{WZ \rightarrow X}}$ dáno ad 2,

je W klíčem?

$W \rightarrow W$ ad 1)

$W \rightarrow Z$ dáno

$W \rightarrow W \wedge W \rightarrow Z \xrightarrow{\text{ad 3}} W \rightarrow WZ \wedge WZ \rightarrow Y \Rightarrow$ dáno ad 2,

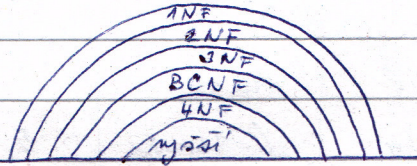
$\Rightarrow W \rightarrow Y$

$W \rightarrow Y \wedge W \rightarrow Z \xrightarrow{\text{ad 2}} W \rightarrow YZ \wedge YZ \rightarrow X \Rightarrow$ dáno

$\Rightarrow W \rightarrow X$

NORMÁLNÍ FORMY

- normální formy se zabývají návrhem relací, tak, aby byly odstraněny neoprávněné duplikáty
- využívají pojem tzv. závislost atributů



- významní NF využívá usměrni se směrem NF
- anomálií je BCNF
- v praxi končíme u 3.NF

1NF

- všechny hodnoty atributů jsou atomické (tj. ^{neoddělitelné} min. opět tabulka)

OK	101	Praha	Bratislava	Po	10.00
+	+	+	+	Út	18.30
OK	102	Bra	Praha	St	13.00
				Čt	8.00

OK	101	Pra	Bra	Po	10.00
OK	101	Pra	Bra	Út	18.30

- využíváme jinou relaci
- pokud se odíváme do tabulky, pak je to v 1.NF

2NF

- relace R je v 2.NF, jestliže je v 1NF a jestliže každý atribut, kt. nepatří k kódu klíč relace R závisí na klíči R

Nechť v relaci R platí $A \rightarrow B$. Řekneme, že B ^(závisí) závisí na atrib. A, jestliže neexistuje žádná vlastní podm. A' stejného atributu A taková, že platí $A' \rightarrow B$.

2NF znamená tedy, když klíč bude mít 2 a více atributů (jinak $A \rightarrow A'$)

Každý sekundární atribut musí záviset na celé kombinaci klíčů.

- zkoumá klíčové atrib.!

Př. Dodávka (č. dod., č. suvčárky, adresa - č. dod., množ.)

84	158	Praha	3
88	194	Ostava	5
21	004	Pleš	26
⋮	⋮	⋮	⋮

Dodávka (č. dod., č. suvč., množství)

Dodavatel (č. dod., adresa - č. dodavatel)

84	158	3	84	Praha
88	194	5	88	Ostava
21	004	26	21	Pleš
⋮	⋮	⋮	⋮	⋮

- z logiky plyne fč. závis., že z č. suvč. č. dod. plyne adresa č. dod.

- závislost tedy na celém klíči, pouze na jeho části

- při změně adresy problémy (opráva v cel. tab. dodávka)

- 1 tab. v náhlém světě by se měla projevit 1 změnou v DB

- např. první relace rozloženy na 2

- fč. závis., která kaže ^(č. dod. → adresa) 2NF a podle ní provedu rozklad

3NF

- stejný problém

Relace R je v 3NF, pokud je v 2NF a

žádný atribut, který není složkou klíče máce R,

není tranzitivně závislý na ~~žádném~~ klíči

relace R.

- mami' uistoma káristost mami' melli'čoxim' alibuty

Pr ZAMEŠTNANCI (č. zam, oddělení, budova)

- předpokládáme, že 1 odd. je v 1 budově

- je v ZNF, kč je pouze 1 alibut

- mami' v ZNF, oddělení → budova

- při stěhování oddělení jech musela mami' s
v každém samostatně

001 KIV PC

002 KKY MN

003 KMA UV

004 KIV PC

- mapování reklamních podle form. sch.

001 KIV KIV PC

002 KKY KKY MN

003 KMA KMA UV

004 KIV

- zkouška melli'čoxi' alibuty !

8. - SQL prezentace

DISTINCT ... odstraní duplicitní řádky
AS ... změni název atributu
'%A' ... podmínka, že poslední písmeno je A

9.

SPOT - slovník odborné terminologie

WEB: AIESEC.PLZEN.CZ

FB: AIESEC.PLZEN

22.11. 16:00 INFO SETUŽKA ZASEDÁČKA CIV (U1125)

ALTER TABLE - změna definice tabulky

Pohledy - chová se jako tabulka, ale je to
dynamická struktura (neexistuje)

- CREATE VIEW jm-pohl [

- definuje se v již existujících struktur

- data se ukládají až v okamžiku
pohledy

CREATE VIEW známky_2 AS SELECT * FROM známky
WHERE id-student = 'A0B120';

Elegantní způsob pohledu, že různé přístupy k různým
tabulkám podle oprávnění.

CREATE VIEW známky_1 AS SELECT id-student, předmět,
známka FROM známky WHERE id-student = 'A0B12P'

CREATE VIEW známky_1 AS SELECT id-student, předmět,
známka FROM známky_2

CREATE VIEW počty-student AS SELECT a.id-student,
jméno, COUNT(*), FROM známky a, student b,

WHERE a.id-student = b.id-student

GROUP BY a.id-student;

Pokud se s pohled. může prac. jako s tabulkou, může nastat situace, že chceme změnit hodnotu polozky, která je pro více podstatná adresa LIKE % PLZEN % (chceme změnit na neplzeňskou adresu)

Přístupová práva v SQL (základní slovy v přehledu)

- V novém SQL je, že tabulka patří tomu, kdo tabulku vytvořil (buď CREATE TABLE nebo spustí script CREATE TABLE)
- vlastník musí uživatelům předat práva
 - příkaz GRANT (předává práva)
 - příkaz REVOKE (odčerpá příslušný)

$G_1, \dots, G_{i-1}, G_i, G_{i+1}, \dots, G_m$
 \uparrow
 P_i

- problem:

Pi A ... zaměstnanci (tabulka)

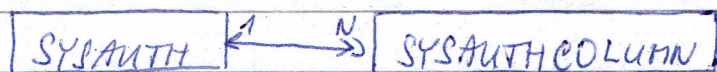
A: GRANT read, insert ON zaměstnanci TO B WITH GRANT OPTION

A: GRANT read ON zaměstnanci TO X WITH GRANT OPTION

A: GRANT ~~read~~ read ON zaměstnanci TO Y

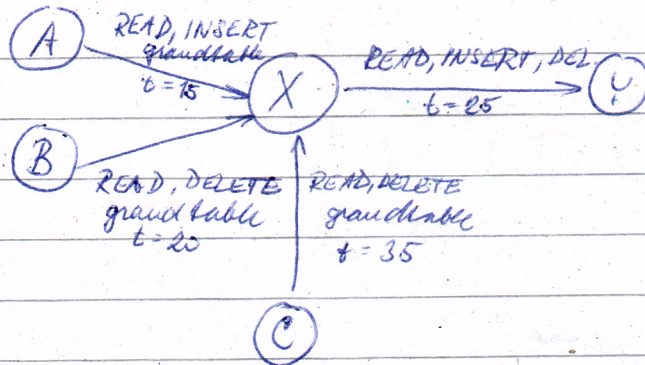
B: GRANT read, insert ~~ON~~ ON zaměstnanci TO X

komu	tabulka	role	read	insert		
B	zam	A	Y	Y	N	N
X	zam	A	Y	N	↑	police



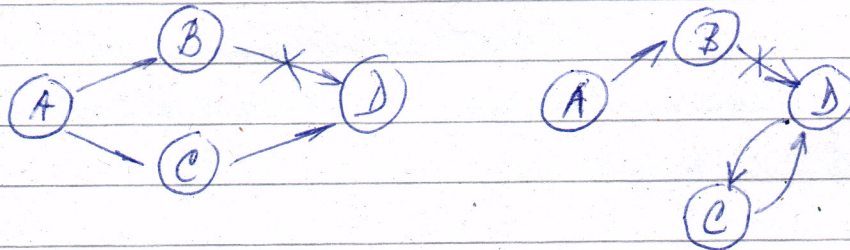
monotonně a' absol. čas, ale je to
pod. úst, se kčec' usmýjeme, ~ jako'm
pořadí nastaly jery.

Pr:



=> konzistentní

t = 35 : B ≠ REVOKE ...



- ke každému'mu' robu. REVOKE je nutno sledovat
zlo cykly

TRANSAKCE A TRANSAKČNÍ ZPRACOVÁNÍ

Ochrana integrity dat
- integrity

- škála dat : odejde disk, nebo počítač (D)
- STAV DATABÁZE - dan' rozhodnutí objekty
(objekt - muba, sa'nam, spojenu, ...)
- něco jako populace
- KONZISTENTNÍ STAV - odpovídá nějakému
možnému reálnému svetu
- např. ke všem' dostupno 1, ale rychle

do STAGU napiše 2

- určit, zda je stav konzistentní, pomáhejí určit integritní omezení

(= určit referenční, doménové, ...)

- co považuje integritní omezení:

- HW chyba (odloup disk)
- selhání OS
- selhání obsluhy
- chyba v aplikačním programu
- chyba dat

- příčin je mnoho, samostatně se jim těžko vyhneme

- ochrany:

- OS
- SRBD - vytvořit si kopii, nejlépe na jiné médium (nejednodušší)

- typ: KONTROLNÍ BOD - časný okamžik, kdy se vytvoří plná kópia DB systému
- v kontrolním bodě je DB konzistentní

TRANSAKČNÍ ZPRACOVÁNÍ - metoda

TRANSAKCE - pojem převzat z účetnictví

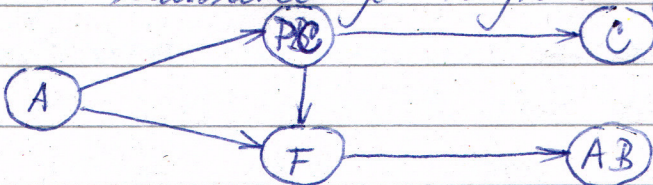
- minimálně 2 kroky (např. přičten se peníze, odečten se jinde)

- v DB je transakce i změna čísla účtů u všech výpisů

- po dobu transakce není DB v konzise.

stav, ale přechází z jednoho KS do druhého

- transakce je logickou jednotkou práce



- A = aktivní stav - skupině se transakce
 PC = částečně potvrzený
 = stav, kdy proběhly všechny klm. operace, ale transakce byla ještě "odsouhlasena"
 F = fail - něco se pokazilo
 - sem přijde, když byla potvrzená nebo po spuštění
 C = komitovaný stav - transakce proběhla v průběhu
 AB = aboard - hodnoty objektů transakce se přenesou do př. stavu

ŽURNAL - spec. soubor, do kterého se ukládají
 všechny změny od koncov. bodu
 - obsahuje

- ID transakce
- identifikátor objektu
- uživatel (kdo psal nam)
- čas spuštění
- nová hodnota objektu
- měly / stará hodn. objektu

- používá žurnál:

REDO • Dopřední použití ž. - zchátrá se z koncov. bodu a nahrazení se až do okamžiku chyby

UNDO • zpětné použití ž. - zchátrá se z chybového

ROLLBACK (nekončí ž.) stav a rolluje se tak dlouho dokud nedostaneme konz. stav

- nepracuje se s daty, ale s chyb. stavem
- systém si pamatuje pouze rozpracované transakce a zhasne ~~se~~ ukončení trans.
- rozprac. trans. má v na začátku (připíše jí na začátek)
- nepřesná pořadí rozle velikost

- 2 základní principy, jak se používají trans., ke zadržení chyby

• **Drůufá'xoni' p'ovzro'adni'**

Tvorba žurnálu s odloženou realizací změn

= transakce musí mít kodový obj. číslo, má je část p'ovzro'adni' a trans. musí být část p'ovzro'adni', pokud nejsou vyložený příslušní záznamy

- nejprve se do žurnálu a pak se do DB

• **Přímy' zápis do báze dat**

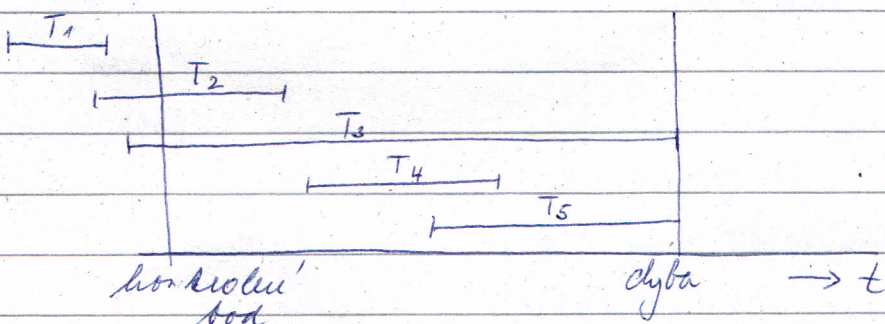
- změna se zapisuje okamžitě, jak je vyvolána

- chyba - nutné provést ROLLBACK

(v tom případě musím mít v žurnálu své kodový)

- může dojít k dominanci některých dat a k původní neúspěšné transakci

- přesto je tento způsob rychlejší, urychluje paralel. běh transakcí



(kompletně paralelní DB)

← UNDO záznamy T_2, T_3, T_5, T_4
REDO T_1, T_4

T_1 je uložena v záloze (vyřeseno)

1) do UNDO uložíme všechny trans. v kr. bodě (T_2 a T_3)

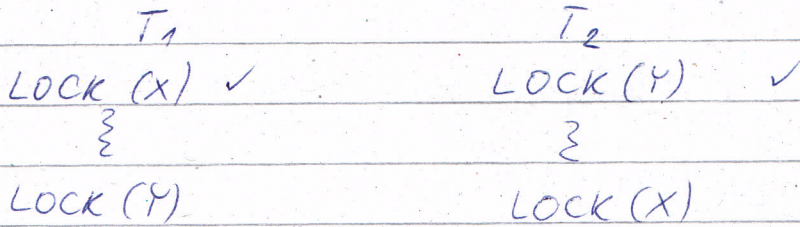
2) přechod žurnálu od KB do chyby

- pokud narazím na začátek transakce do UNDO

T_1	Plan 4	T_2
LOCK (X)		LOCK (Y)
		$B \leftarrow \text{read}(Y)$
		$B \leftarrow 2 * B$
		write (Y, B)
		UNLOCK (Y)
LOCK (Y)		
$A \leftarrow \text{read}(X)$		
$B \leftarrow \text{read}(Y)$		
$B \leftarrow B + A$		
$A \leftarrow A - B$		
write (Y, B)		
UNLOCK (Y)		
write (X, A)		
UNLOCK (X)		
		LOCK (X)
		$A \leftarrow \text{read}(X)$
		$A \leftarrow A + 1$
		write (X, A)
		UNLOCK (X)

$$X = 1, Y = 3$$

- Objekt může být v daném okamžiku uzamčen pouze pro 1 transakci.
- Jakmile transakce nějaký objekt odemkne, už nemá žádný objekt zamknut
 \Rightarrow důsledek: chyba napravena T_2



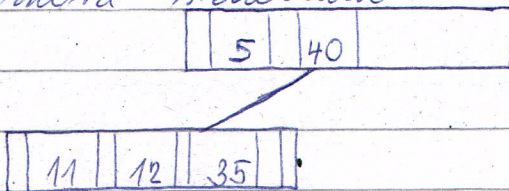
UVÁZNUTÍ = jeo známý \Leftarrow OS

- řešení snadné pomocí ŽURNÁLU
- kramake se stopou a spusti znovu

Jiné než dvoufázové protokoly

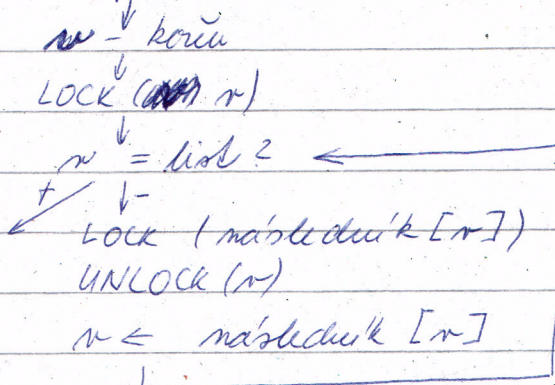
- vyžadují znalost \forall dat v DB
- nejlepší, když je to silněji STRONG
- Protokoly Δ může samkovat litost. projekt
- Δ další objekty mohou samkovat jen, když je samkovat jeho předchůdce
- Δ odumknout může když chci
- Δ objekty kl. musí samkva a odumkva, musí znova samkovat (ten samý - rozdíl spusti parat. titu)

skomora' struktura:

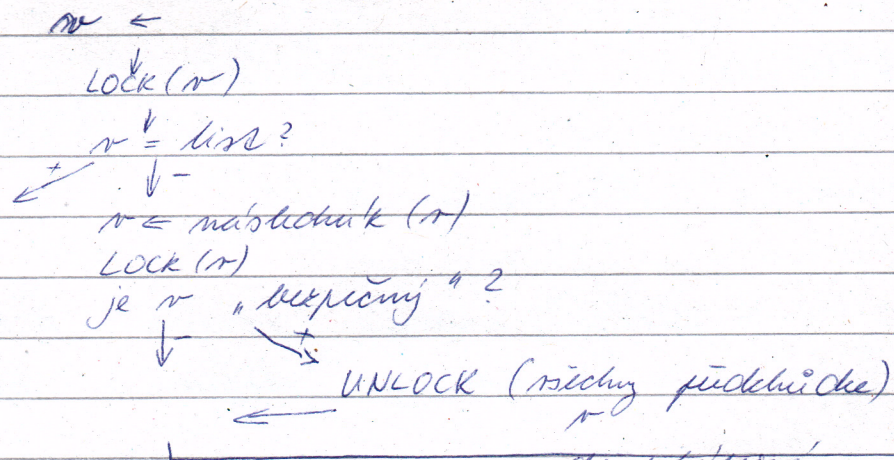


B- skomv, všechny zadržiny

protokol po READ



protokol WRITE (UPDATE)



Které protokoly jsou lepší? \leftarrow dostupnost? Neověřeno.
hodnotěme zamýkací protokoly

- dobu uzamčení
- počet zámků a další okružku
- skomozí protokol zamýkací
- dvoufázový - " na delší dobu

Jiné než uzamkací protokoly

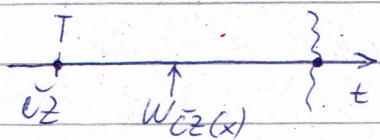
- doazít: upřádanost, ale nepoužije zámků
- intertax; fix bind
- princip: používat časový značek
 - nemáme abstr. čas
 - umíme říct, co bylo dřív a co později (posloupnost jím)
- musíme mít časovou značku začátku transakce

Transakce (pořádek)

ČZ

- u každého obj musím vidovat ČZ
 - posledního čtení $R_{cz}(x)$
 - posledního zápisu $W_{cz}(x)$
- protokol pro čtení objektu X v čase ČZ
 - $ČZ < W_{cz}(x)$
 - odmítnuti pořádku
- $R_{cz}(x) \leftarrow \max(ČZ, R_{cz}(x))$

- čas. znaky samičij konzistencei



prohled pořadank na řápis X r čas čz

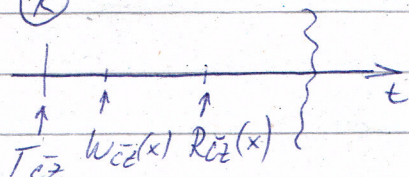
$$\checkmark \text{čz} < W_{\check{c}z}(x) \text{ nebo } R_{\check{c}z}(x)$$

rychlost

odmknout

$$W_{\check{c}z}(x) \leftarrow \check{c}z$$

(K)



- SRBD roz. na čas. značkách je rychlejší než se samky

Thomasův plánovač

pořadank na řápis X r čas čz

$$\checkmark \text{čz} < R_{\check{c}z}(x)$$

-

odmknout

$$\checkmark \text{čz} < W_{\check{c}z}(x)$$

-

+

$$(K) \rightarrow \text{splnit } W_{\check{c}z}(x) \leftarrow \check{c}z$$

vzhledem k pořadanku řápisu je sděleno, že řápis prokázal, ale do DB a nic měříte