Slezská univerzita v Opavě, FPF *Podklady k přednáškám*
Studijní obor: IVT *Ročník: IV.*

Předmět:	**Projektování IS I**
Téma:	**Denormalizace databáze**

Vyučující:	dr. Dušan Kajzar	Školní rok:	nezařazeno

Obsah:

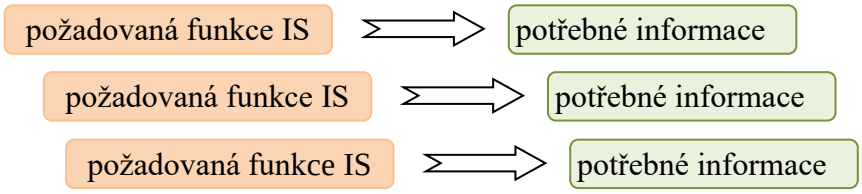
1. Normalizovaná databáze a normální formy

Návrh databáze probíhá „shora dolů“:

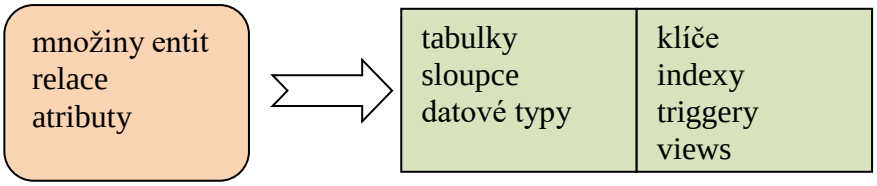
- návrh konceptu (pojetí, představy) řešení,
  - definice účelu databáze, tj. k čemu bude sloužit,
  - jaké druhy informací musí uchovávat, k jakému účelu (vzhledem k funkcím IS), ...
  - identifikace základních množin entit a rozdělení informací do jednotlivých množin entit,
  - zachycení vazeb mezi množinami entit a požadovanou funkcionalitou IS,
  - kontrola pokrytí požadavků vytvořeným konceptuálním návrhem,
- návrh logické struktury dat
  - návrh db množin entit (tabulek) a jejich atributů,
  - hledání souvislostí a návrh vazeb mezi množinami entit,
  - dekompozice návrhu, normalizace,
  - datové typy a omezení pro ukládané hodnoty,
  - zpracování vazeb mezi množinami entit tabulkami a funkcemi IS,
  - kontrola úplnosti – zda model pokrývá požadavky řešení,
  - tj. úplnost a správnost množin entit, jejich atributů, klíčů, vazeb, ...
- návrh fyzického modelu
  - implementace návrhu do konkrétního databázového systému,
  - např. Oracle, MS SQL, MySQL, Sybase, Postgre, ...
  - využití vlastností konkrétního db systému.

Úrovně DB návrhu:

- konceptuální model



- logický a fyzický (relační) db design.



Normalizace databáze:

- jistá pravidla procesu převodu logického návrhu db do návrhu relačního,
- neklíčové položky normalizované tabulky závisí na celém klíči a na ničem jiném.

Normální formy:

- 1. NF
  - každý sloupec obsahuje atomické hodnoty, nedá se rozdělit na více sloupců,

cis_zam	prijme	mistnost	adresa
20156	Horák	A258, C556	Vysoká 556, Lipno

- 2. NF
  - tabulky jsou v 1. NF,
  - každá neklíčová položka závisí na celém klíči, nikoliv jen na části klíče,

zavod	stredisko	nazev_zavodu	nazev_strediska
04	20156	DrevoPlech	Lisovna

- 3. NF
  - tabulky jsou ve 2. NF,
  - neklíčová položka nezávisí na jiné neklíčové položce.

cis_zam	prijme	ulice	cislo_pop	psc	mesto
20156	Horák	Vysoká	556	382 78	Lipno

Výhody normalizované databáze:

- minimalizace redundance dat,
- eliminace množství insertů, update, delete,
- efektivnější I/O operace,
- zvýšení výkonnosti db serveru.

Možná úzká místa normalizované databáze:

- v dotazech jsou spojení (joiny) mnoha tabulek,
- v dotazech časté opakované sumace číselných údajů – např. částek za období, zboží na skladě apod.,
- kolize přístupu uživatelů k tabulkám – zamykání, čekání dotazů na uvolnění zámků,
- kolize operací typů „modifikace“ a typů „read“,
- časem velké množství řádků v tabulkách (např. 10 – 100 mil.),
- atd.

2. Denormalizace databáze

Denormalizace DB:

- záměrný ústup od normalizovaného schématu
  - tj. DB záměrně nesplňuje požadavky NF (např. 3.NF),
- denormalizace musí být vždy účelově zaměřená a odůvodněná
  - sledujeme určitý záměr, účel, dobře jej posuzujeme,
  - odůvodněna speciálními požadavky na zvýšení výkonnosti určité funkcionality systému,
- denormalizace vyžaduje
  - předchozí normalizaci,
  - znalost procesů, kterými jsou data zpracovávána, znalost dané aplikace,
  - důslednou evidenci denormalizačních úprav v dokumentaci k IS.

Rizika denormalizace:

- může způsobit problém zachování integrity dat,
- může vést ke zhoršení výkonnosti v jiné části aplikace.

Denormalizační „Trade- Off“:

- něco za něco,
- např. (insert, update, delete) vs. (select).

Příklady možných přínosů denormalizace:

- minimalizace joins,
- redukce počtu cizích klíčů,
- redukce počtu indexů,
- redukce počtu tabulek,
- rychlejší odezva systému – předem spočtené agregace,
- apod.

Příklady negativních dopadů denormalizace:

- citelné zpomalení operací modifikujících data (insert, update, delete),
- aplikační závislost
  - po úpravě v aplikaci (nová verze) nemusí již být denormalizace pro danou aplikaci vhodná,
  - aplikační řešení je nutno „hlídat“ a revidovat,
- časová závislost
  - co se jevilo vhodné před 2 měsíci, nyní již způsobuje problémy (jiné objemy dat v tabulkách apod.),
  - nutno hlídat a revidovat,
- někdy zjednodušuje kód programu, jindy naopak bude kód složitější.

3. Techniky denormalizace databáze

Denormalizační techniky:

- přidání redundandních sloupců,
- přidání odvozených (derived) sloupců,
- shlukování (collapsing) tabulek,
- duplikace tabulek,
- rozdělení (splitting) tabulek.

Přidání redundantního sloupce:

- např. řešení problému intenzivního přístupu ke sloupci „název podavatele“,
- eliminace joinů mnoha tabulek,

podaci_cislo	id_pod
856442	88123

id_pod	nazev_pod
88123	Dřevoplech a.s.

- SELECT p1.podaci_cislo, p1.id_pod, p2.nazev_pod
  FROM tab_podani p1, tab_podavatele p2 WHERE p1.id_pod = p2.id_pod;

podaci_cislo	id_pod	nazev_pod
856442	88123	Dřevoplech a.s.

id_pod	nazev_pod
88123	Dřevoplech a.s.

- SELECT podaci_cislo, id_pod, nazev_pod FROM tab_podani;
- úprava vyžaduje:
  - údržbu nového sloupce (aktualizace obou tabulek),
  - více prostoru na disku.

Přidání odvozeného sloupce (derived column):

- např. řešení problému intenzivního joinu pro výpočet sumací,

podaci_cislo	id_pod	castka
856442	88123	21 852
54833	88123	11 553
		...

id_pod	nazev_pod
88123	Dřevoplech a.s.

- např. častý select pro zjišťování sumací za jednotlivé podavatele,

podaci_cislo	id_pod	castka
85644	88123	21 852
54833	88123	11 553
		...

id_pod	nazev_pod	sum_kc
88123	Dřevoplech a.s.	43 505

- úprava vyžaduje:
  - údržbu nového sloupce (aktualizace hodnot odvozeného sloupce),
  - více prostoru na disku.

Shlukování (collapsing) tabulek:

zakaznik_id	zakaznik_name	....
789	Horáček	

zakaznik_id	objednavka
789	AB753-14

zakaznik_id	zakaznik_name	objednavka	....
789	Horáček	AB753-14	
789	Horáček	CD888-66	

- nevýhody:
  - údaje o zákazníkovi se zbytečně duplikují s každou objednávkou,
  - klíčem se stává zakaznik_id + objednavka.

Duplikace tabulek:

- např. řešení problému intenzivního přístupu k tabulce
  - ze strany účetních - nové záznamy, modifikace stavu objednávek apod.,
  - i ze strany pracoviště analytiků – tisk průběžných reportů.

cis_uctu	objednavka	castka	...
745 449	AXC255	55 000	

originální tabulka pro operativní přístup k datům (Read/Write)

cis_uctu	objednavka	castka	...
745 449	AXC255	55 000	

duplikovaná tabulka s výběrem sloupců, které zajímají analytiky (Read)

Vertikální rozdělení (splitting) tabulky:

- rozdělení tabulek s velkým množstvím záznamů na části:
  - aktivní tabulka – probíhá operativní zpracování,
  - archívní tabulka – operativní zpracování je ukončeno, výpisy reportů, analýzy dat apod.

cis_uctu	objednavka	castka	status	...
745 449	AXC255	55 000	active	
555 123	DC548	12 100	active	
035 226	GH444	5 000	inactive	

cis_uctu	objednavka	castka	status	...
745 449	AXC255	55 000	active	
555 123	DC548	12 100	active	

aktivní RW

cis_uctu	objednavka	castka	status	...
035 226	GH444	5 000	inactive	
025 449	XR5589	20 000	inactive	

archivní R

- poznámky použití v praxi:
  - v praxi dochází k tvorbě archivních tabulek např. po měsících, vznikají tak měsíční archivní tabulky,
  - v praxi se rovněž dělí i celé databáze na db operativní a archivní.

Horizontální rozdělení (splitting) tabulky:

- např. rozdělení tabulky na část s adresními (evidenčními) údaji
- a část s údaji finančními - pro zpracování částek Kč,

cis_pk	podaci_posta	jméno	adresa	...	castka	...
2589	749 01				5 000	
8569	753 20				800	

cis_pk	podaci_posta	jméno	adresa	...
2589	749 01			
8569	753 20			

cis_pk	podaci_posta	castka	...
2589	749 01	5 000	
8569	753 20	800	

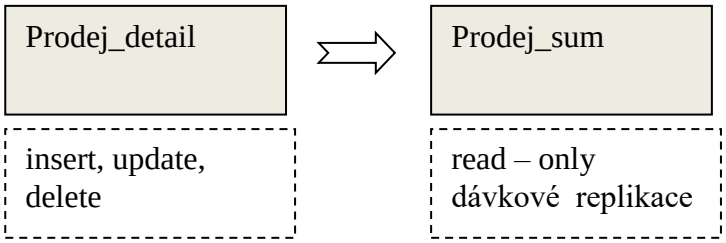
- zde nemusí jít o případ denormalizace, nýbrž o běžné úvahy nad efektivností návrhu databáze.

4. Údržba denormalizované databáze

Údržba denormalizované databáze:

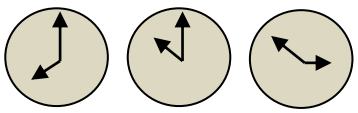
- tj. zajištění konzistence denormalizovaných tabulek,
- metody
  - tvorba Read Only tabulek,
  - dávková údržba,
  - použití triggerů,
  - údržba v rámci aplikační logiky.

Použití Read Only tabulek:



Dávková údržba:

- odvozených (derived) sloupců, duplicitních sloupců, pravidelný měsíční splitting tabulek apod.,
- použití naplánovaných „maintenance jobů“.



Použití triggerů:

- jde o on-line údržbu (!),
- definujeme trigger nad primární tabulkou,
- trigger reaguje na modifikaci a provede on-line úpravu v sekundární tabulce,
- např. sumace hodnot, celkový počet kusů na skladě apod.

podaci_cislo	id_pod	castka
85644	88123	21 852
54833	88123	11 553

id_pod	nazev_pod	sum_kc
88123	Dřevoplech a.s.	43 505

Údržba v rámci aplikační logiky:

- změny na originální tabulce jsou prováděné i na odvozených datech programovým kódem aplikace,
- změna v rámci transakce – begin tran, ..., commit.

Periodické prověřování přínosů denormalizace:

- při úvahách o provedení denormalizace,
- při pravidelném prověřování provozu, zda je denormalizace ještě přínosem.

Doporučený postup:

- vyberte skupinu tabulek – kandidáti k denormalizaci,
- zaznamenejte si v tabulce měření tzv. baseline
  - výkonnostní profile stávajícího stavu práce s tabulkami,
  - sledování dotazů - total I/O, Tran/Sec, Response time, Lock Waiting, ...
- denormalizujte danou skupinu tabulek
  - zaznamenejte provedené úpravy v programátorské dokumentaci (!),
- zpracujte měření po denormalizaci
  - tzv. additional profile nového stavu,
  - sledování dotazů - total I/O, Tran/Sec, Response time, Lock Waiting, ...
- zvažte přínosy / nevýhody (trade-off) daného stavu,
- pokračujte v úvahách pro další skupinu tabulek.

5. Závěrem k tématu

Denormalizaci chápejme:

- jakožto účelově zaměřenou techniku,
- tj. ústup od zásad normalizace z určitého důvodu,
- proto je nutné ji ze strany vývojáře v projektové dokumentaci zaevidovat.

Zaznamenat v projektové dokumentaci:

- proč jsem se k denormalizaci rozhodl, na základě jakých měření výkonnosti či úvah,
- čeho se denormalizace týká, o jaký typ denormalizace se jedná.

Pravidelné ověřování důvodů a přínosů:

- zda je denormalizace ještě přínosem
  - po úpravě v aplikaci (nová verze aplikace),
  - po významné změně v objemu dat tabulkách,
- co mohlo být přínosem pro daný programový kód či dané objemy dat v tabulkách, může být zdrojem problémů výkonnosti databáze po určitém čase.

Častým zdrojem problémů výkonnosti db bývá:

- „zapomenutá“ denormalizace,
- například tzv. hinty (pokyny) pro optimalizátor SQL dotazů
  - „použij toto pořadí tabulek při joinu“,
  - „použij tento index“, apod.