VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Databázové systémy (IDS)

2015/2016

Konceptuálny model

Veterinárna klinika

Michal Ormoš (xormos00)
Ľubomír Gallovič (xgallo03)

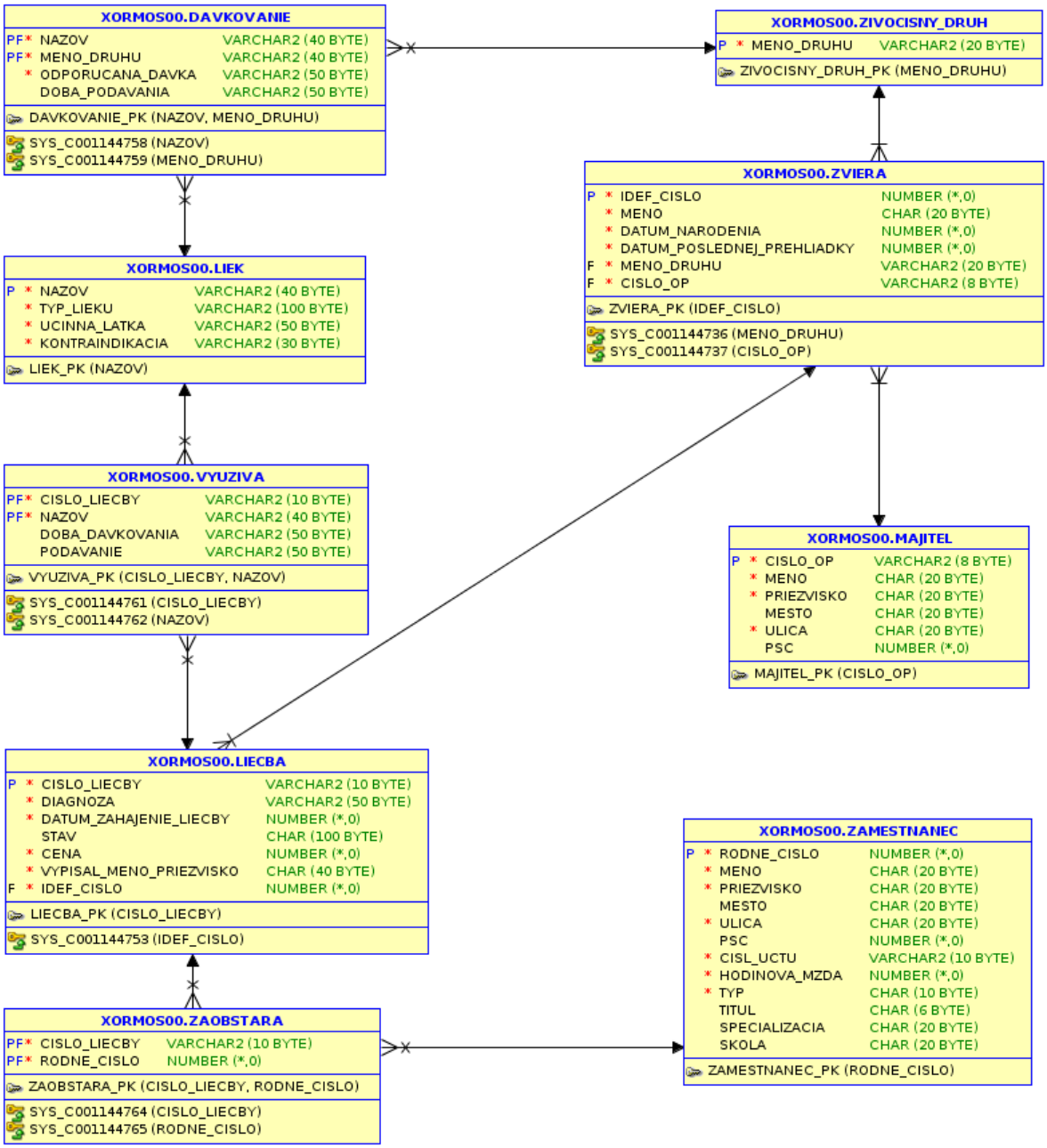
28.4.2015

Obsah

1 Zadanie

Namodelujte jednoduchý informační systém veterinární kliniky, kde se střídá více lékařů veterinářů a sester. O sestrách a veterinářích informační systém uchovává osobní informace jako jméno, titul, adresa, číslo účtu, hodinová mzda, přičemž lékaři mají přístup k údajům sester, ale sestry k údajům lékařů nikoliv. Hlavním účelem informačního systému je správa informací o léčených zvířatech, jednotlivých léčbách a jejich majitelích. U každého zvířete, které podstoupilo alespoň jednu léčbu, je v systému vytvořen záznam (o léčbě i o majiteli zvířete). Majitele zvířete lze v systému vyhledat podle jména, příjmení či adresy. Jeden majitel může samozřejmě vlastnit více domácích mazlíčků, o kterých je vytvořen záznam s jejich jménem, o jaký druh zvířete se jedná (kočka, pes, křeček, atd.), datum narození, datum poslední prohlídky a informace o jednotlivých léčbách, které dané zvíře podstoupilo (diagnóza, datum zahájení léčby, stav, cena). Při léčbě mohou být naordinovány léky s určitým dávkováním a pro specifikovanou dobu podávání. U léků předpokládejte možnost dohledat typ léku, jeho účinnou látku, kontraindikace, pro jaký druh zvířete se hodí (kočka, pes, ...) a doporučené dávkování pro daný druh zvířete. Jeden typ léku se může hodit pro více druhů zvířat s různým dávkováním a pro léčbu různých nemocí (nemoc modelovat nemusíte, postačí pouze název). V případě podání léku přímo v ordinaci (např. injekce) je k dispozici informace, kdo daný lék podal, stejně tak u léčby pro konkrétní zvíře lze dohledat, který veterinář ji vypsal.

2 Finálne schéma databáze



2.1 Generalizácia/Špecializácia

Vzťah genralizácia/špecializácia sme realizovali pri tabuľke `Zamestnanec` tak, že tabuľka `Zamestnanec` obsahuje položku `typ`, ktorý môže byť buď `Veterinár` alebo `Sestra`, obe obsahujú špeciálne položky a to `titul` a `špecializácia` pre `Veterinár`. `Škola` pre typ `Sestra`

3 Implementácia

Skript si najprv vytvorí základné objekty a naplní tabuľky ukážkovými hodnotami. Následne musí zadefinovať pokročile obmedzenia a objekty. Ďalej skript obsahuje povinné ukážkové príklady manipulácie s dátami a požiadavkami demonštrujúce obmedzenia tohoto skriptu.

3.1 Triggery

Implementácia triggerov pozostávala v našom prípade z overovania správnosti čísla účtu (čísla účtov sa udávájú v SK formáte v základnom tvare) v tabuľke `Zamestnanec`. Presnejšie tento trigger kontroluje či je dĺžka čísla účtu práve 10. Taktiež či každá číslica účtu vynásobené správnou váhou (podľa normy ISO 7046 MOD 11-2) spočítaná so všetkými ostatnými číslicami je delitelná číslom 11 bezo zvyšku.

Druhý triger realizuje automatickú inkrementáciu primárneho kľuča v tabuľke `Zviera`. Trigger je realizovaný sekvenciou, pre pamätanie si čísla posledného pridaného primárneho kľúča a teda postupného zvyšovania identifikačného čisla novo pridaných zvierat.

Oba triggery sa spúšťajú vždy pred vkladaním dát do danej tabuľky.

3.2 Procedúry

Našou úlohou bolo aj vytvorenie dvoch netriviálnych procedúr s využitím kurzorov a premenných s dátovým typom odkazujúcim sa na riadok alebo stĺpec tabuľky.

Prvou procedúrou je `priemerna_cena`, ktorá prijíma argumenty diagnóza a živočíšny druh a určí priemernú cenu zadanej diagnózy pre zadaný druh. Procedúra postupne prechádza záznamami o liečbach druhu, a ak ide o liečbu zadanej diagnózy pripočíta jej cenu k celkovej cene a inkrementuje počet výskytov. Výsledok získa na konci vydelením celkovej ceny a počtu výskytov. V prípade delenia nulou nastala výnimka a znamená to, že nebol nájdený žiadny záznam danej diagnózy pre daný živočíšny druh.

Druhá procedúra má názov `vydaje_majitela` a jej jediným argumentom je číslo majiteľa v systéme. Účelom tejto procedúry je zistenie koľko peňazí minul majiteľ na liečby svojich zvierat, a ktoré zviera ho stálo najviac. Procedúra najprv vytvorí tabuľku liečob všetkých zvierat a zoradí ju podľa ich identifikačného čísla. Túto tabuľku postupne prechádza a spočítava ceny jednotlivých liečob. Zároveň spočítava ceny liečob súčasného zvieraťa, porovnáva ich s dosiaľ najvyššou dosiahnutou cenou zvieraťa a prípadne nastaví novú hodnotu. V prípade zmeny identifikačného čísla sa počíta cena ďalšieho zvieraťa. Procedúra nakoniec vypíše celkovú cenu a meno zvieraťa, ktoré stálo majiteľa najviac. Ak sa nenašli žiadne informácie o liečbach, systém daného majiteľa nenašiel.

3.3 Explain plan a vytvorenie indexu

Explain plan sme demonštrovali na jednoduchom `SELECT` dotaze. Demonštrovali sme použitie bez indexu a následne sme zadefinovali index a spustili explain plan znovu. Index sme vytvorili na tabuľke s viacerými dátami, vystupom bolo:

Bez použitia indexu

Id	Operácia	Meno	Riadky	Bajty	Cena(%CPU)	Čas
0	SELECT STATEMENT		11	671	7 (15)	00:00:01
1	HASH GROUP BY		11	671	7 (15)	00:00:01
2	HASH JOIN		11	671	6 (0)	00:00:01
3	TABLE ACCESS FULL	ZVIERA	11	285	3 (0)	00:00:01
4	TABLE ACCESS FULL	LIECBA	11	286	3 (0)	00:00:01

S použitím indexu

Id	Operácia	Meno	Riadky	Bajty	Cena(%CPU)	Čas
0	SELECT STATEMENT		11	671	6 (17)	00:00:01
1	HASH GROUP BY		11	671	6 (17)	00:00:01
2	HASH JOIN		11	671	5 (0)	00:00:01
3	TABLE ACCESS BY INDEX ROWID BATCHEL	ZVIERA	11	385	2 (0)	00:00:01
4	INDEX FULL SCAN	INDEXEXPLAIN	11		1 (0)	00:00:01
5	TABLE ACCESS FULL	LIECBA	11	286	3 (0)	00:00:01

Ako môžeme vidieť databáza nami definovaný index nepoužila, kedže sama usúdila, že pri takom malom počte riadkov, aký naša databáza obsahuje, by to bola zbytočná operácia (príliš drahá pre cenu procesora). A však v druhej tabuľke sme databáze povedali, aby aj napriek tomu index použila. Ako môžeme vidieť, znížila sa počet prístupov na disk (cena), ale na druhej strane percento využitia procesora sa zvýšil (%CPU).

`SELECT STATEMENT` označuje uskutočnený `SELECT` dotaz, `HASH GROUP BY` znamená, že sa zoraďuje podľa hashovacieho kľúča. Následujúca operácia `HASH JOIN` znamená, že sa tabuľky spájajú. Ďalej sa nám *explain plan-y* líšia, v prípade nepoužitia indexu sa dotaz ďalej vykonával tak, že v tabuľke `Zviera` prechádzalo pomocou `TABLE ACCESS FULL`, kedy sa prechádza tabuľka celá od začiatku až po koniec bez použitia indexov. Na druhej strane, s použitím indexu sa vykonával `TABLE ACCESS BY INDEX ROWID` v tabuľke `Zviera`, kedy sa pristupuje do tabuľky cez konrétny riadok (vidíme ako sa použil náš index). Následne sa vykoval `INDEX FULL SCAN` s našim indexom `INDEXEXPLAIN` čo znamená, že sa použil index na výpis, bez toho aby sme museli nazrieť do tabuľky. Posledný krok je totožný s výpisom *explain plan* bez indexu.

3.4 Prístupové práva

Úloha simuluje udelenie prístupových práv zdravotnej sestre. Sestra má povolené všetky práva v tabuľkách zviera a majiteľ, teda ich môže editovať, zmazať alebo pridať nové záznamy. Čiastočné práva má v tabuľke liečba, ktorej záznamy môže zobraziť a editovať, ale nemôže ich mazať ani vytvárať nové. Úplne odoprený prístup má k tabuľke zamestnancov, pretože ide o dôverné informácie

s ktorými sestra nemá kompetenciu pracovať. Sestra má povolené používanie procedúr `priemerna_cena` a `vydaje_majitela`, pretože pracujú s informáciami, ktoré sú sestrám prístupné.

3.5 Meterializovaný pohľad

Pre materializovaný pohľad sme využili select, ktorý zistí počet príslušníkov každého živočíšneho druhu. Materializovaný pohľad sme demonštrovali výpisom daného selectu, pridaním novej hodnoty do tabuľky, potvrdením zmien pomocou `COMMIT` a ďalším selectom, v ktorom sa už vypísali aktualizované hodnoty. Pohľad využíva `REFRESH FAST ON COMMIT`, teda hodnoty sa okamžite aktualizujú.

4 Záver

Projekt nás naučil základnej tvorbe databázových systémov na praktickom príklade z reálneho sveta. Skript sme riadne testovali v prostredí Oracle a na školskom servery Oracle12c. Skript sme tvorili, vyvíjali a spravovali v prostredí Oracle SQL Developer odporučeného na prednáškach. K úspešnému vypracovaniu tejto úlohy sme si naštudovali učebnú látku a používali aj ostatné informácie získavané z rôznych neoficiálnych internetových zdrojov. Ako značne nápomocné nam prišli krátke projekty vypracované na prednáškach našimi kolegami našeho z ročníka.