

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií

DATABÁZOVÉ SYSTÉMY
2015/2016

Projekt č. 4,5 – Dokumentácia popisujúca finálny návrh

Zadanie – Hotel

Maroš Cocul'a (xcocul00)

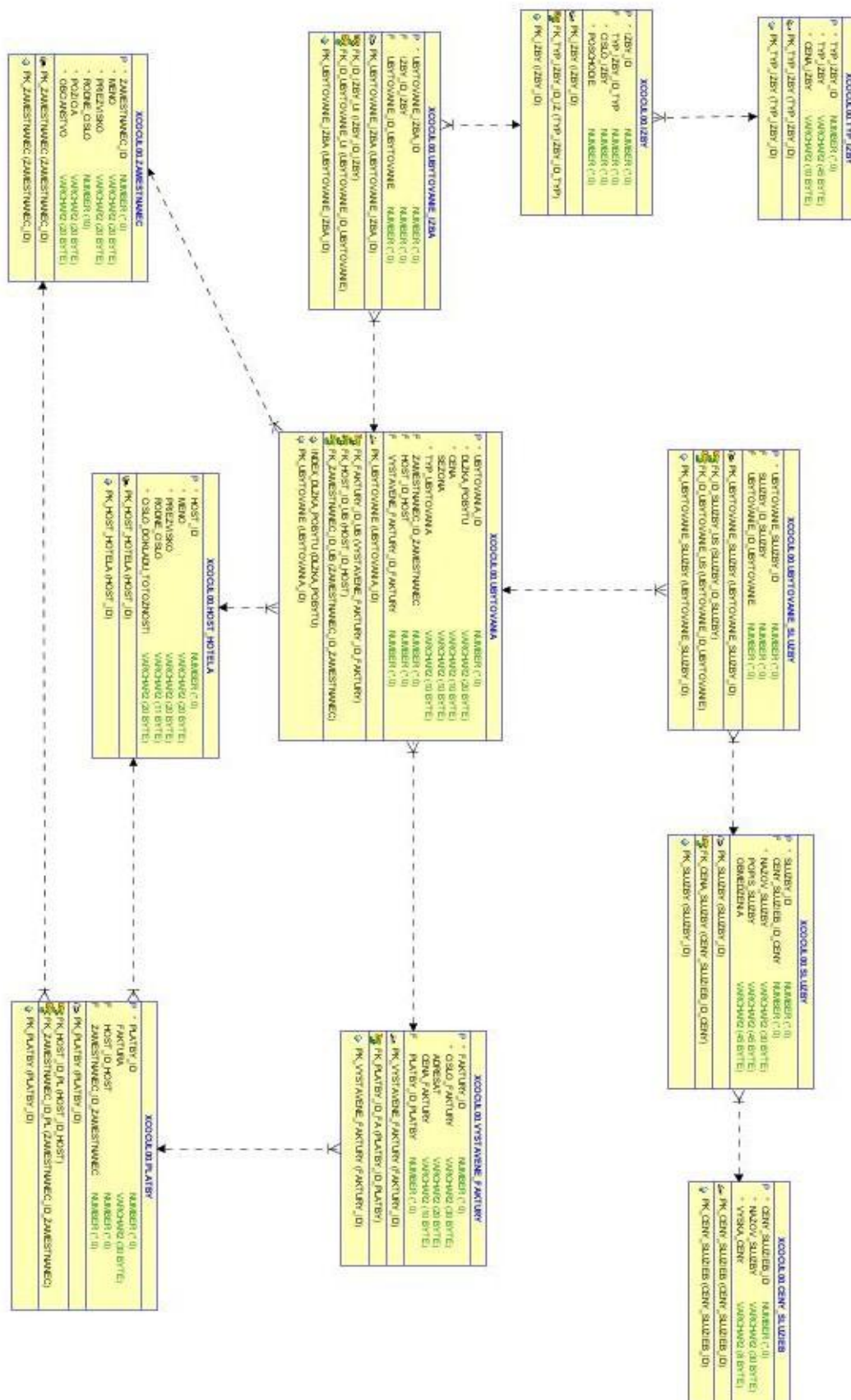
Dávid Prexta (xprext00)

Brno, 30. apríla 2016

Model prípadov užitia



ER diagram



1.0 Generalizácia/specializácia :

Pre vzťah generalizácia/specializácie sme zvolili tabuľku Ubytovania. Ubytovanie môže byť buď typu „rezervácia“, alebo „ubytovanie“. Rezervácia sa od ubytovania líši, tým že hosť nenastúpi na pobyt do hotela. Izba sa tak uvoľní pre ďalšieho hosťa.

2.0 Implementácia databázy

Skript implementuje databázu následovne. Vytvorí tabuľky aj s primárnymi a cudzími kľúčmi. Pri vytváraní tabuliek sa každému riadku pridajú potrebné vlastnosti ako dátový typ, obmedzenia či prípadne nutnosť údajov pri vkladaní dát (NOT NULL). Následne je potreba tieto tabuľky naplniť údajmi o čo sa taktiež stará skript. Skript ďalej obsahuje objekty ako trigre, materializovaný pohľad, procedúry, selekty či index. V skripte sa nachádzajú aj ukážky funkčnosti a použitia jednotlivých objektov a priradenie práv užívateľovi, v našom prípade išlo o zamestnanca.

2.1 Trigre

Skript obsahuje dva databázové trigre. Prvý slúži na automatické pridanie identifikačného čísla pri vkladaní do tabuľky HOST_HOTELA. Je implementovaný pomocou sekvencie ktorá sa pri každom spustení trigra inkrementuje o 1 a toto číslo sa vloží do tabuľky na pozíciu HOST_ID. Podstatou tohto trigra je, že užívateľ ktorý chce vložiť údaje do tabuľky sa nemusí zaoberať identifikačným číslom hosťa.

Druhý trigre sa zaoberá správnosťou zadaného rodného čísla v tabuľke ZAMESTNANEC. Rodné číslo sa skladá z desiatich znakov. Prvých 6 číslic je nasledovaných znakom „/“, potom nasledujú ďalšie štyri čísla. Avšak znak „/“ je nepovinný, tým pádom sme ho vynechali a mohli sme použiť dátový typ NUMBER. To nám veľmi uľahčilo situáciu. Prvých 6 číslic vyjadrujú dátum narodenia v tvare RR.MM.DD. Posledné štyri číslice sú kontrolné. Trigre sa spustí automaticky pri vložení nových dát, alebo aktualizácií už vložených dát. Pri nesprávnom formáte nastane výnimka a dáta sa nepodarí vložiť/aktualizovať.

Pravidla pre rodné číslo :

- Rodné číslo musí byť deliteľné bezo zvyšku číslom 11
- Dátum narodenia musí existovať. (Jednoduché ošetrenie vložení do dátového typu DATE. Pri zlome dátumu nastane výnimka)
- Ak je mesiac väčší ako 50 jedná sa o ženu. (Odčítame 50 od mesiaca)

2.2 Procedúry

Pri vytváraní procedúr sme využili kurzor, ktorý slúži na prácu s viac ako jedným riadkom v prípade že nám selekt vráti viac ako jeden riadok. Ošetrenia výnimiek, ktoré môžu nastať a premennú s dátovým typom, ktorý sa odkazuje na daný riadok.

Prvá procedúra (percento_povolania()) vypíše percentuálne rozloženie zamestnancov daného povolania. Procedúra má jeden parameter a to názov povolania, ktoré chceme hľadať. Výstupom procedúry je výpis percentuálneho rozloženia hľadaného povolania medzi zamestnancami. Pri nulovom výskyte zamestnania v tabuľke sa nevypíše percento, ale správa o nenájdení povolania. Ďalej môže nastať situácia, že tabuľka je prázdna. Tým pádom by sme delili nulou a nastane výnimka delenie nulou. V našom prípade to neberieme ako chybu, ale vypíšeme že tabuľka v ktorej chceme hľadať je prázdna.

Druhá procedúra (zaznam_ubytovani()) vypíše všetky ubytovania zadaného hosťa podľa identifikačného čísla. Procedúra má jeden parameter a to identifikačné číslo hľadaného hosťa. Výstupom je výpis všetkých ubytovaní a rezervácií daného hosťa. Ak hosť nemal žiadny záznam vypíše sa správa, že sa záznam nenašiel. Pri situácii že daný selekt nevráti žiaden záznam, nastane výnimka ktorú vypíšeme.

2.3 Explain plan a index

Pri vytváraní indexu sme využili zároveň aj explain plan. Ten funguje tak že vypíše plán ako spracuje daný selekt. Pri vytváraní selektu sme použili spojenie tabuliek HOST_HOTELA a UBYTOVANIA pomocou natural join. Agregáčnú funkciu maximum a klauzulu group by. Pre demonštráciu sme najprv použili explain plan bez indexu a potom sme vytvorili index ktorý sme použili pri druhom explain plan.

EXPLAIN PLAN bez použitia indexu :

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		5	250	4 (25)	00:00:01
1	HASH GROUP BY		5	250	4 (25)	00:00:01
2	NESTED LOOPS		5	250	3 (0)	00:00:01
3	NESTED LOOPS		5	250	3 (0)	00:00:01
4	TABLE ACCESS FULL	UBYTOVANIA	5	125	3 (0)	00:00:01
5	INDEX UNIQUE SCAN	PK_HOST_HOTELA	1		0 (0)	00:00:01
6	TABLE ACCESS BY INDEX ROWID	HOST_HOTELA	1	25	0 (0)	00:00:01

EXPLAIN PLAN s použitím indexu :

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		5	250	3 (34)	00:00:01
1	HASH GROUP BY		5	250	3 (34)	00:00:01
2	NESTED LOOPS		5	250	2 (0)	00:00:01
3	NESTED LOOPS		5	250	2 (0)	00:00:01
4	TABLE ACCESS BY INDEX ROWID BATCHED	UBYTOVANIA	5	125	2 (0)	00:00:01
5	INDEX FULL SCAN	INDEX_DLZKA_POBYTU	5		1 (0)	00:00:01
6	INDEX UNIQUE SCAN	PK_HOST_HOTELA	1		0 (0)	00:00:01
7	TABLE ACCESS BY INDEX ROWID	HOST_HOTELA	1	25	0 (0)	00:00:01

V prvom prípade databáza nepoužila index pretože pri malom počte riadkov to nie je najefektívnejšie riešenie. V druhom prípade sme však index vytvorili ručne a použili ho. V druhom prípade sa znížil počet prístupov na disk, ale zvýšilo sa percento vyťaženia procesoru.

Prvé štyri kroky sú rovnaké. Nultý krok značí selekt ktorý bol vykonaný. Prvý krok znamená, že sa používa hashovací kľúč. Druhý a tretí krok znamená že spojenie tabuliek je také, že pre každú jednu položku sa prejdú všetky riadky z druhej tabuľky.

V prvom prípade štvrtý krok znamenal že sa tabuľka prechádzala úplne celá bez akéhokoľvek indexu. Krok päť znamená že sa k tabuľkám pristupuje cez B-strom, kde vypadne riadok PK_HOST_HOTELA. Následne sa vypíše výsledok podľa indexu.

V druhom prípade sa kroky líšia. Štvrtý krok znamená, že s k tabuľke pristupuje cez konkrétny riadok s použitím indexu. Ďalej nasledoval výpis s použitím indexu.

2.4 Materializovaný pohľad

Materializovaný pohľad sme implementovali pre druhého člena ktorý používa tabuľky vytvorené prvým členom. Obsahuje informácie o počte jednotlivých zamestnancov podľa ich povolání.

Využili sme materializované logy, s ktorými pri zmene tabuľky stačí použiť rýchlu obnovu namiesto kompletnej obnovy, kde by bola potreba spúšťať celý pohľad. Použili sme vlastnosti ako cache (optimalizuje dáta), build immediate (naplnenie pohľadu hneď pri vytvorení) a enable query rewrite (optimalizátor).

V ukážke funkčnosti sme poukázali nato, že sa pohľad zmení až po príkaze commit. Najprv vytvoríme logy a samotný pohľad. Pohľad sa naplní aktuálnymi dátami. Potom do tabuľky ZAMESTNANEC vložíme nového záznam. Materializovaný pohľad sa nezmení. Zadáme príkaz commit, po ktorom sa zmení aj náš pohľad.

2.5 Prístupové práva

Prístupové práva sme simulovali pre zamestnanca ktorý je recepčný. Nemá práva k všetkým tabuľkám. Môže len prezerat' a vytvárať nových host'ov a vytvárať/upravovať ubytovania. Ďalej môže prezerat' izby a služby, či zapisovať faktúry a platby. Zamestnanec ma login xprext00.

Nemôže zapisovať nové služby, ani pridávať nové izby. Má celkovo odoprený prístup k tabuľke ZAMESTNANCI. Pretože nemôže meniť žiadne údaje o zamestnancov hotelov.

Práva pre zamestnanca recepčný :

- ALL pre tabuľky UBYTOVANIA, VYSTAVENE_FAKTURY, HOST_HOTELA, PLATBY a ich sekvencie
- READ pre ostatné tabuľky, aby mohol vytvoriť nové ubytovanie