

# FAKULTA INFORMAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Databázové systémy - Finální dokumentace  
Informační systém recepce hotelu

# 1 Převod generalizace/specializace

Vzhledem k tomu, že specializace je zároveň disjunktní (nelze aby pokoj byl typu VIP a standard najednou) a totální (musí vždy být alespoň jednoho z těchto typů), jediným řešením bylo zavedení typu jako atributu entity Pokoje.

# 2 Procedury

Procedura `pocet_rezervaci_v_obdobi` bere jako parametry 2 datумы, které vymezují časový interval, ve kterém se bude vyhledávat. Oba parametry mají datový typ odkazující na sloupec tabulky `Rezervace_pokoj`. Je očekáváno větší množství záznamů se kterými se bude pracovat, proto je využito kurzoru. Procedura obsahuje ošetření vyjímek. Výsledkem procedury je výpis na výstup s počtem rezervací v zadaném intervalu.

Druhá procedura s názvem `nejoblíbenější_sluzba` projde celou tabulkou `Objednani_sluzby` a vypočítá, kterého typu služby je v tabulce nejvíce. Tedy která služba je nejoblíbenější.

# 3 Triggery

První trigger je na ověření správnosti rodného čísla zákazníka. Nejdříve se ověří, že se v něm vyskytnou pouze čísla. Pokud je délka rodného čísla 10 znaků musí operace modulo 11 vyjít 0. Uvažujeme roky od 1930 do 2019. Správné měsíce jsou 0-12 pro muže, 50-62 pro ženy a v případě vyčerpání unikátního čtyřčíslí se k měsíci u obou pohlaví přičítá 20. Ověření dne jsme ponechali v rozmezí 1-31. Pokud rodné číslo splňuje všechny tyto podmínky, může se záznam zapsat do tabulky, v opačném případě se objeví patřičná chybová hláška.

Druhý trigger slouží k doplnění primárního klíče - `id_platby` v případě, že nebyl zadán při vkládání hodnot do tabulky. Zde se hodnota doplňuje automaticky ze sekvence `platba_gen`.

# 4 Materializovaný pohled

Námi vytvořený pohled se aktualizuje při commitu. Do jeho provedení, se žádné změny provedené v báze tabulce neprojeví v pohledu. Pohled vytvářel jeden člen týmu nad daty z tabulky druhého člena týmu. Nejprve se použije `BUILD IMMEDIATE`, díky kterému se daný dotaz ihned provede. Následně přidáme `REFRESH ON COMMIT`, který nám zajistí, že při zadání příkazu `COMMIT` se nám potvrdí veškeré změny, které jsme provedli.

## 5 Explain plan

Cílem Explain planu bylo zobrazení posloupnosti provedení jednotlivých operací a jejich složitosti. S tím se váže užití indexu a následná optimalizace. Bez využití indexu je nutné přistoupit ke všem záznamům, vyhledává probíhá tudíž sekvenčně. Tomu můžeme zamezit napsáním indexu, který složitost může snížit na logaritmickou. Z obrázků můžeme vidět, že čas se v obou případech nezměnil, tj. zůstává na 1 vteřině, ovšem Cost se snížil ze čtyřech jednotek na tři. Jako příklad jsme vybrali dotaz zobrazující počet lidí, co byli ubytováni v roce 2018. Index jsme aplikovali na sloupec `od_kdy` v tabulce `Rezervace_pokoje`.

Obrázek 1: Bez využití indexu

PLAN_TABLE_OUTPUT							
Plan hash value: 3672392879							
-----							
Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	
-----							
0	SELECT STATEMENT		1	25	4 (25)	00:00:01	
1	SORT AGGREGATE		1	25	4 (25)	00:00:01	
2	HASH GROUP BY		1	25	4 (25)	00:00:01	
* 3	TABLE ACCESS STORAGE FULL	REZERVACE_POKOJE	13	325	3 (0)	00:00:01	
-----							
Predicate Information (identified by operation id):							
-----							
3 - storage("REZERVACE_POKOJE"."OD_KDY">=TO_DATE(' 2018-01-01 00:00:00',							
'syyyy-mm-dd hh24:mi:ss') AND "REZERVACE_POKOJE"."OD_KDY"<=TO_DATE(' 2018-12-31							
00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND "REZERVACE_POKOJE"."DO_KDY">=TO_DATE('							
2018-01-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND							
"REZERVACE_POKOJE"."DO_KDY"<=TO_DATE(' 2018-12-31 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))							

Obrázek 2: S využití indexu

PLAN_TABLE_OUTPUT							
Plan hash value: 233569630							
-----							
Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	
-----							
0	SELECT STATEMENT		1	25	3 (34)	00:00:01	
1	SORT AGGREGATE		1	25	3 (34)	00:00:01	
2	HASH GROUP BY		1	25	3 (34)	00:00:01	
3	TABLE ACCESS BY INDEX ROWID BATCHED	REZERVACE_POKOJE	13	325	2 (0)	00:00:01	
* 4	INDEX RANGE SCAN	DATE_INDEX	1		1 (0)	00:00:01	
-----							
Predicate Information (identified by operation id):							
-----							
4 - access("REZERVACE_POKOJE"."OD_KDY">=TO_DATE(' 2018-01-01 00:00:00', 'syyyy-mm-dd							
hh24:mi:ss') AND "REZERVACE_POKOJE"."DO_KDY">=TO_DATE(' 2018-01-01 00:00:00', 'syyyy-mm-dd							
hh24:mi:ss') AND "REZERVACE_POKOJE"."OD_KDY"<=TO_DATE(' 2018-12-31 00:00:00', 'syyyy-mm-dd							
hh24:mi:ss') AND "REZERVACE_POKOJE"."DO_KDY"<=TO_DATE(' 2018-12-31 00:00:00', 'syyyy-mm-dd							
hh24:mi:ss'))							
filter("REZERVACE_POKOJE"."DO_KDY">=TO_DATE(' 2018-01-01 00:00:00', 'syyyy-mm-dd							
hh24:mi:ss') AND "REZERVACE_POKOJE"."DO_KDY"<=TO_DATE(' 2018-12-31 00:00:00', 'syyyy-mm-dd							
hh24:mi:ss'))							

