

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií

DATABÁZOVÉ SYSTÉMY
2015/2016

Projekt č. 4 a 5
SQL skript

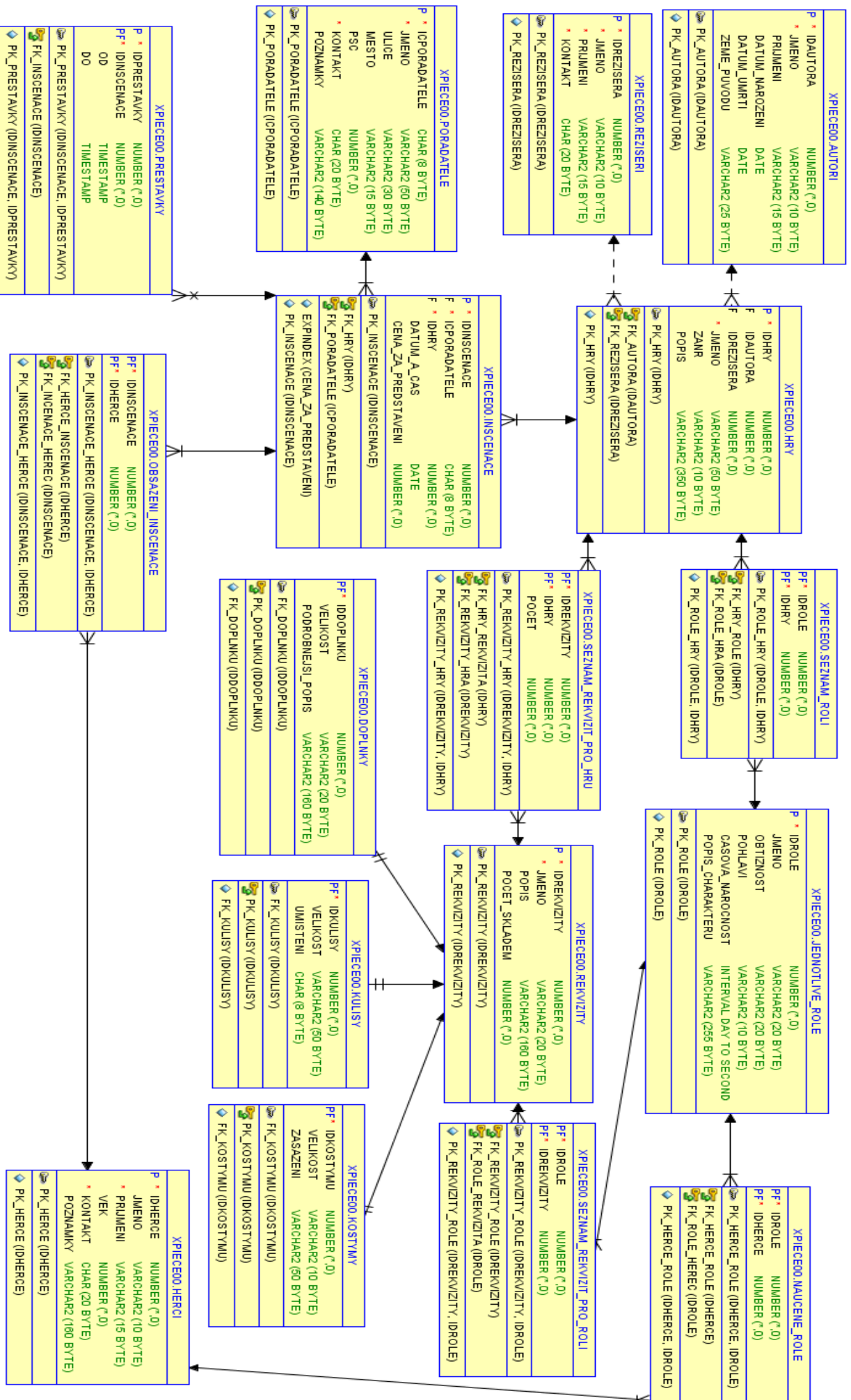
Zadání č. 42
Plánování divadelního představení

Jan Pavlica (xpavli78)

Adam Piecek (xpiece00)

Brno, 28. dubna 2016

Namodelujte informační systém pro divadelní představení. Divadelní scéna má ve svém repertoáru více her od různých autorů. Každá hra má seznam rolí, které ve hře vystupují, jednotlivé role u jedné hry může mít nastudováno více herců, ale každá hra má pouze jednoho režiséra. Je tedy možné mít různé inscenace pro jednu hru. Informační systém umožňuje pořadatelům plánovat jednotlivé inscenace na konkrétní datum a čas, zadat čas a počet přestávek, a naplánovat obsazení inscenace herci, kteří mají role nastudované. Každá role má určitou obtížnost, časovou náročnost na jevišti, popis charakteru postavy a potřebné rekvizity k postavě (klobouk, hůl, atd.). Informační systém tedy umožňuje spravovat také rekvizity, které jsou ke hře potřeba a v jakém počtu. Zároveň systém umožňuje kontrolu, zda jsou potřebné rekvizity a jejich počet skladem. Systém umožňuje pořadatelům vyhledat si požadovanou hru nejen podle herců či názvu, ale také podle informací o autorovi a režisérovi. V případě, že divadelní scéna do svého repertoáru zadává nastudování nové hry a počet potřebných rekvizit pro hru přesahuje počet rekvizit skladem, systém na tuto skutečnost pořadatele upozorní. Herci si mohou v systému vyhledávat pouze informace o hrách a rolích ve kterých hrají. Oproti tomu režiséři a pořadatelé mají přístup ke všem záznamům v systému a mohou je měnit.



Explain plan

Použití explain plan bylo aplikováno nad select dotazem, který má za úkol sečíst všechny inscenace, u kterých je cena za představení větší než 250 Kč. V tabulce Inscenace je 6 záznamů v tabulce Poradatele 3. Bez použití indexu je výpis následující:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT		4	264	7 (15)
1	HASH GROUP BY		4	264	7 (15)
*2	HASH JOIN		4	264	6 (0)
3	TABLE ACCESS FULL	PORADATELE	3	120	3 (0)
*4	TABLE ACCESS FULL	INSCENACE	4	104	3 (0)

Při indexaci atributu Cena_za_predstaveni indexem ExpIndex:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT		4	264	6 (17)
1	HASH GROUP BY		4	264	6 (17)
*2	HASH JOIN		4	264	5 (0)
3	TABLE ACCESS FULL	PORADATELE	3	120	3 (0)
4	*TABLE ACCESS BY INDEX	INSCENACE	4	104	2 (0)
*5	INDEX RANGE SCAN	EXPINDEX	4		1 (0)

*TABLE ACCESS BY INDEX ROWID BATCHED – nezkráceně

Jak je z tabulek patrné, zavedením indexu jsme celkově snížili hodnoty u přístupů na disk (26 vs. 23). Avšak hodnoty CPU se zvýšily (30 vs. 34).

Dále z tabulek můžeme vyčíst, že v situaci bez použití indexu, prochází celé tabulky (TABLE ACCESS FULL) Poradatele a Inscenace.

V druhém případě prochází celou tabulku Poradatele, avšak tabulku Inscenace za použití indexu (TABLE ACCESS BY INDEX ROWID BATCHED). Přístup k tabulkám potom optimalizátor použil test rozsahu či hodnot v sloupci (INDEX RANGE SCAN).

Materializovaný pohled

Při implementaci materializovaného pohledu patřící druhému členu týmu bylo nutné vytvořit materializovaný log. Aby měl člen data v pohledu po příkazu commit aktuální, bylo přidáno `REFRESH FAST ON COMMIT`. Nastavení `CACHE`, aby čtené bloky byly umísťovány na začátku rychlé vyrovnávací paměti. Pro naplnění pohledu bezprostředně pro jeho vytvoření bylo doplněno `BUILD IMMEDIATE`. Dále povolení optimalizátoru příkazem `ENABLE QUERY REWRITE` a zakázání historie operací nad pohledem příkazem `NOLOGGING`. Díky tomuto nastavení druhý člen viděl ihned změnu, jakmile byly do tabulek vloženy nové data.

Triggery

V rámci skriptu byly implementovány dva triggery. První slouží k automatickému doplnění primárního klíče v tabulce `Autori` v případě, že bude zadán klíč `NULL`. Vzhledem k tomu, že v rámci celé databáze jsou klíče generovány ze sekvence, nebyl problém samotný trigger implementovat.

Druhý trigger pak slouží k ověření identifikačního čísla osoby nebo podnikatelského subjektu. Tato informace slouží jako primární klíč v tabulce `Poradatele`. Trigger se spouští při pokusu o přidání nového primárního klíče do tabulky. Kontrola IČ je ověřována jednoduchým vzorcem (*Rovnice 1*). V případě, že poslední člen podmínku nesplňuje, je vypsána chyba.

$$n_0 = (11 - (n_1 * 2 + n_2 * 3 + n_3 * 4 + n_4 * 5 + n_5 * 6 + n_6 * 7 + n_7 * 8) \bmod 11) \bmod 10$$

Rovnice 1: Vzorec pro kontrolu IČ

Procedury

Skript také obsahuje dvě procedury. První procedura slouží k vypůjčení rekvizit ze skladu pro potřeby dané hry. Argumentem procedury je název hry, pro kterou chceme vypůjčit rekvizity. V případě, že se hra s daným názvem v databázi vyskytuje, je nejprve provedena kontrola dostupnosti potřebných rekvizit. V případě, že je zjištěna chybějící rekvizita, tak je tato rekvizita vypsána. V případě, že jsou všechny rekvizity skladem je jejich počet v skladu snížen o požadované množství pro hru.

Druhá procedura slouží k výpisu inscenací v zadaném období. Argumenty jsou data odkdy, a dokdy chceme vypsát inscenace. V případě, že jsou v daném období nalezeny inscenace je vypsán jejich pořadatel, název hry, datum, cena a také jednotlivé přestávky.