

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií

DATABÁZOVÉ SYSTÉMY

2016/2017

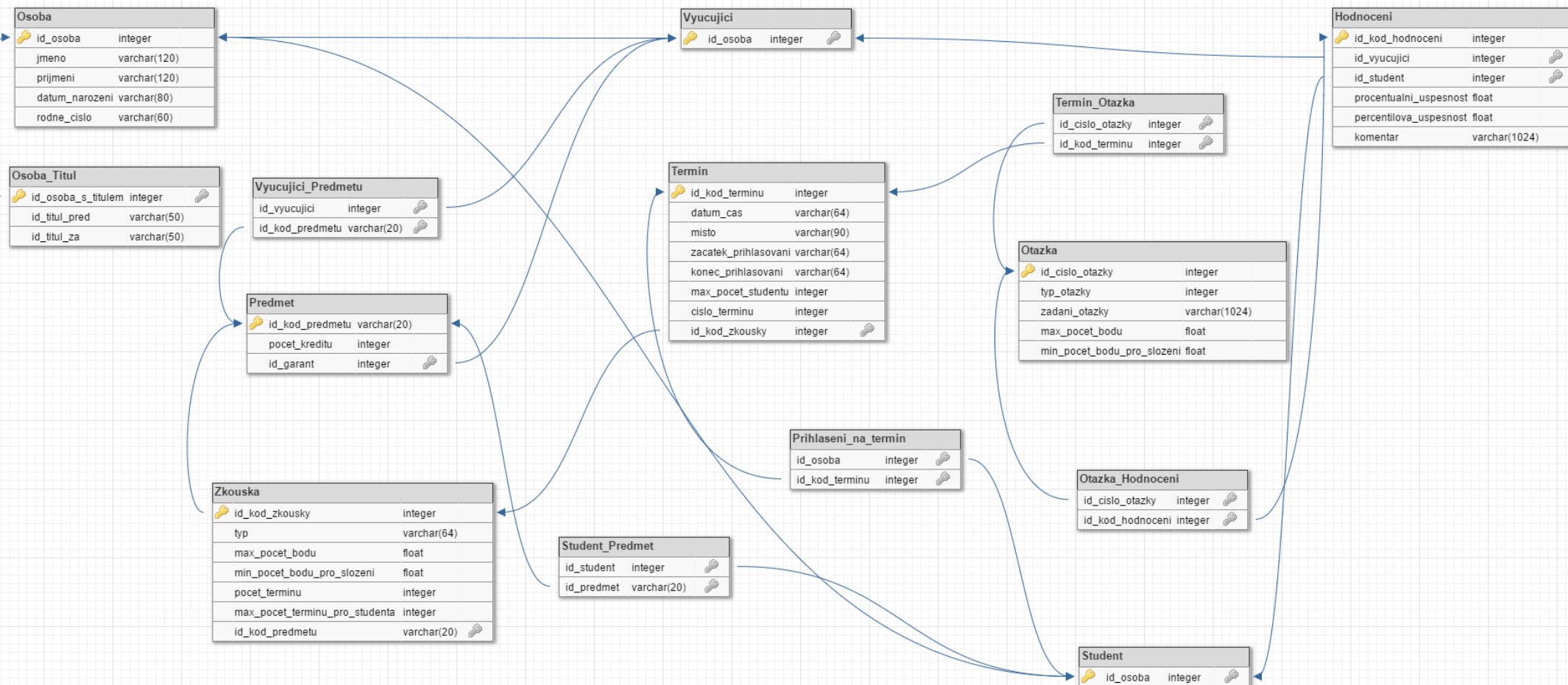
Projekt IDS

Zadání č. 20 – Zkoušky (IUS)

Zadání

Navrhněte modul fakultního informačního systému, který bude umožňovat studentům se přihlašovat na zkoušky a vyučujícím tyto zkoušky hodnotit. Zkouška může být buď semestrální nebo půl semestrální, přičemž semestrální zkouška má vždy alespoň 3 termíny. Studenti se mohou na zkoušky přihlašovat pouze pokud je přihlašování otevřeno, u semestrální zkoušky musí navíc systém zajistit, že se student nesmí přihlásit na více než 3 termíny a také, že na další termín se může přihlásit až po proběhnutí termínu, na který je aktuálně přihlášen. Pro každou zkoušku je potřeba uchovávat základní informace o ní (datum a čas, kdy se koná, maximální počet studentů, jenž se mohou přihlásit, maximální počet bodů ze zkoušky a minimální počet bodů potřebný pro úspěšné složení zkoušky, počet otázek, ...). Body ze zkoušky mohou vkládat pouze učitelé, jenž jsou k předmětu, pod který zkouška spadá, přiřazeni. Pokud má zkouška nastaveno více otázek, musí hodnocení obsahovat body pro každou z těchto otázek. Hodnocení také může obsahovat komentář. Kromě přihlašování musí systém poskytnout studentům i možnost se odhlásit (pokud je přihlašování stále otevřeno). Studenti také mohou zjišťovat body získané ze zkoušky spolu s informacemi kdo a kdy tuto zkoušku hodnotil.

Schéma databáze



žlutý klíč (primární klíč), šedý klíč (cizí klíč)

Generalizace/specializace

Generalizaci/specializaci jsme použili pro entitu Osoba, ta je specializována na Studenta a Vyučujícího. Tabulky věnující se této problematice mají stejné názvy jako entity.

Implementace

Navrhli jsme jednoduchý informační systém pro studenty a vyučující, který umožňuje základní správu zkoušek. Studentům systém umožňuje se přihlašovat a odhlašovat ze zkoušek (jak půl semestrálních, tak semestrálních) a zjišťovat své hodnocení. Vyučující pak mohou přidávat hodnocení jednotlivých otázek z jednotlivých termínů, jednotlivých zkoušek pro jednotlivé studenty.

Triggery

Pro naši databázi jsme implementovali dva triggery. První trigger (`auto_increment_osoba`) slouží pro automatickou inkrementaci primárního klíče u nově vkládaných položek tabulek Osoba. Druhý trigger (`rodne_cislo_kontrola`) slouží pro ověřování správnosti rodného čísla. To se skládá z devíti nebo desíti číslic. Prvních šest číslic vyjadřuje datum narození (formát `rrmmdd`), to může ale v některých případech vypadat odlišně od skutečného data narození (například u žen se k měsíci připočítává 50). Zbytek rodného čísla odděluje jednotlivé lidi stejného pohlaví narozené v jeden den. Desetimístná rodná čísla jsou beze zbytku dělitelná jedenácti, tuto vlastnost ověřujeme rozdílem lichých a sudých čísel (ten se rovná jedenácti).

Procedury

Pro naši databázi jsme implementovali dvě procedury. První procedura (`update_otazka`) má za úkol aktualizaci otázky identifikované pomocí sloupce `id_cislo_otazky`. Pokud zadané `id_cislo_otazky` neexistuje, vypíše se na výstup hlášení, že nebylo nic přidáno. Pokud je hodnota NULL, vypíše se výjimka. V případě, že byla nalezena jiná chyba (například ve formátu) vypíše se obecná výjimka. Pokud se procedura vykoná úspěšně, vrátí na výstup hlášení o úspěšném vykonání. Funkčnost je testována mimo jiné příkladem, který má na místě `id_cislo_otazky` NULL a pro který vypisuje procedura správně error. Druhá procedura (`insert_or_actualize_person`) vkládá osobu podle rodného čísla (využívá kurzoru). Pokud se osoba již v databázi vyskytuje její informace se pouze aktualizují (nedochází k duplicitě). Na výstup se vypíše, zdali došlo k aktualizaci nebo ke vložení nové osoby. Tato procedura podle argumentu zároveň vloží nebo aktualizuje data vkládané osoby v tabulkách Student/Vyucujici a Osoba_titul.

Přístupové práva

Přístupová práva jsou navržena pro konkrétního vyučujícího (XHOLIK13). Ten má garantovaný přístup k tabulkám Zkouska, Termin, Otazka, Hodnoceni... zároveň má nad dalšími tabulkami

garantovaná práva na provedení jednotlivých operací (select, insert, delete). A v poslední řadě má přístupové právo nad materializovaným pohledem Count_Student_IDS.

Materializovaný pohled

Vytvořili jsme materializovaný pohled (Count_Student_IDS) patřící stejnému vyučujícímu. Tento pohled umožňuje sledovat počet studentů v jednotlivých předmětech. Pro optimalizaci byl nejdříve vytvořen log.

Vytvoření indexu a explain planu

Následující dvě tabulky slouží pro porovnání náročnosti vykonávání operace bez indexu (první tabulka) a s indexem (druhá tabulka). Můžeme si všimnout snížené náročnosti na CPU ze 4 % na 2 % při maximálním zatížení.

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		9	792	4 (25)	00:00:01
1	HASH GROUP BY		9	792	4 (25)	00:00:01
2	NESTED LOOPS		9	792	3 (0)	00:00:01
3	TABLE ACCESS FULL	OSOBA	14	1050	3 (0)	00:00:01
* 4	INDEX UNIQUE SCAN	SYS_C007050	1	13	0 (0)	00:00:01

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		9	792	2 (0)	00:00:01
1	SORT GROUP BY NOSORT		9	792	2 (0)	00:00:01
2	NESTED LOOPS		9	792	2 (0)	00:00:01
3	TABLE ACCESS BY INDEX ROWID	OSOBA	14	1050	2 (0)	00:00:01
4	INDEX FULL SCAN	PERSONINDEX	14		1 (0)	00:00:01
* 5	INDEX UNIQUE SCAN	SYS_C007050	1	13	0 (0)	00:00:01

Ukázkové výběry z databáze

Pro demonstraci funkčnosti provádí skript několik výběrů z databáze. Výběry zahrnují:

- výpis všech osob s titulem
- výpis termínů a ke kterým předmětům náleží
- výpis všech Petřů, kteří studují IDS
- výpis všech studentů, kteří studují jen IZP
- výpis počtu vyučujících v jednotlivých předmětech, které mají více jak jednoho vyučujícího
- výpis počtu studentů přihlášených na jednotlivé termíny zkoušek
- výpis všech vyučujících s titulem Ph.D a bez titulu CSc.

- výpis všech studentů, kteří studují IZP nebo IDS
- výpis všech studentů, přihlášených na termín půl semestrální zkoušky z IDS, narozených v roce 1996