

Učebny

Dokumentácia pre predmet IDS

Matej Marušák a Ondřej Budai

Apríl 2016

1 Zadanie

Navrhňte modul fakultného informačného systému, ktorý bude umožňovať správu učeben a laboratórií na FITu. Systém musí uchovávať základní informácie o učebnách, jejich umiestnení, kapacitu a vybavení (projektor, klimatizace, kamera, počet a typ tabulí, ...). Dále musí systém umožňovať rezervaci učeben pro různé akce (zkoušky, semináře, cvičení, ...). Rezervace učeben a laboratórií mohou provádět jen akademičtí pracovníci, změny informací o učebnách a laboratóriích provádí pouze správce systému. Systém musí umožňovat vypisovat rozvrh učeben po oborech, ročnících, předmětech nebo po učebnách dle potřeby, musí umožnit zjistit volné učebny v daném termínu, případně s konkrétním vybavením. Systém také musí uchovávat informace o tom, kdo a kdy danou učebnu rezervoval.

2 ER diagram a vzťah špecializácie

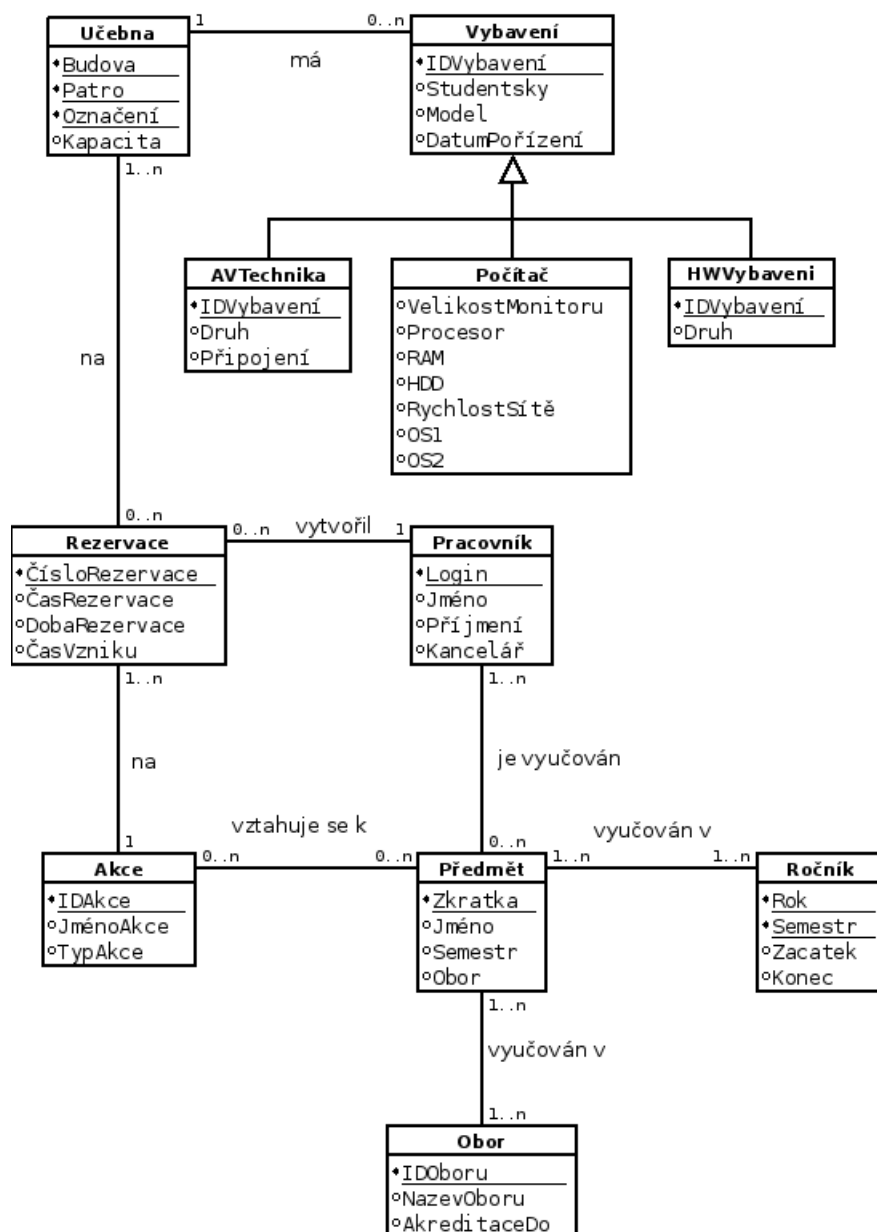


Figure 1: ER diagram

3 Prevod na relačnú databázu

Entity boli modelované 1:1 na tabuľky relačnej databázy. V Figure 2 sú primárne kľúče znázornené červenou farbou. Do tabuliek, ktoré sú viazané na iné, boli pridané primárne kľúče tabuliek bazových, takéto atribúty sú v Figure 2 znázornené modrou farbou. Ak je jeden atribút zároveň kľúčom primárnym aj cudzím je v Figure 2 znázornený farbou zelenou.

Pre výskyt vzťahom M:N bolo potrebné pridať 3 tabuľky a to konkrétne PREDMET_V_OBORU, PRACOVNIK_V_PREDMETU a REZERVACE_NA_UCEBNU.

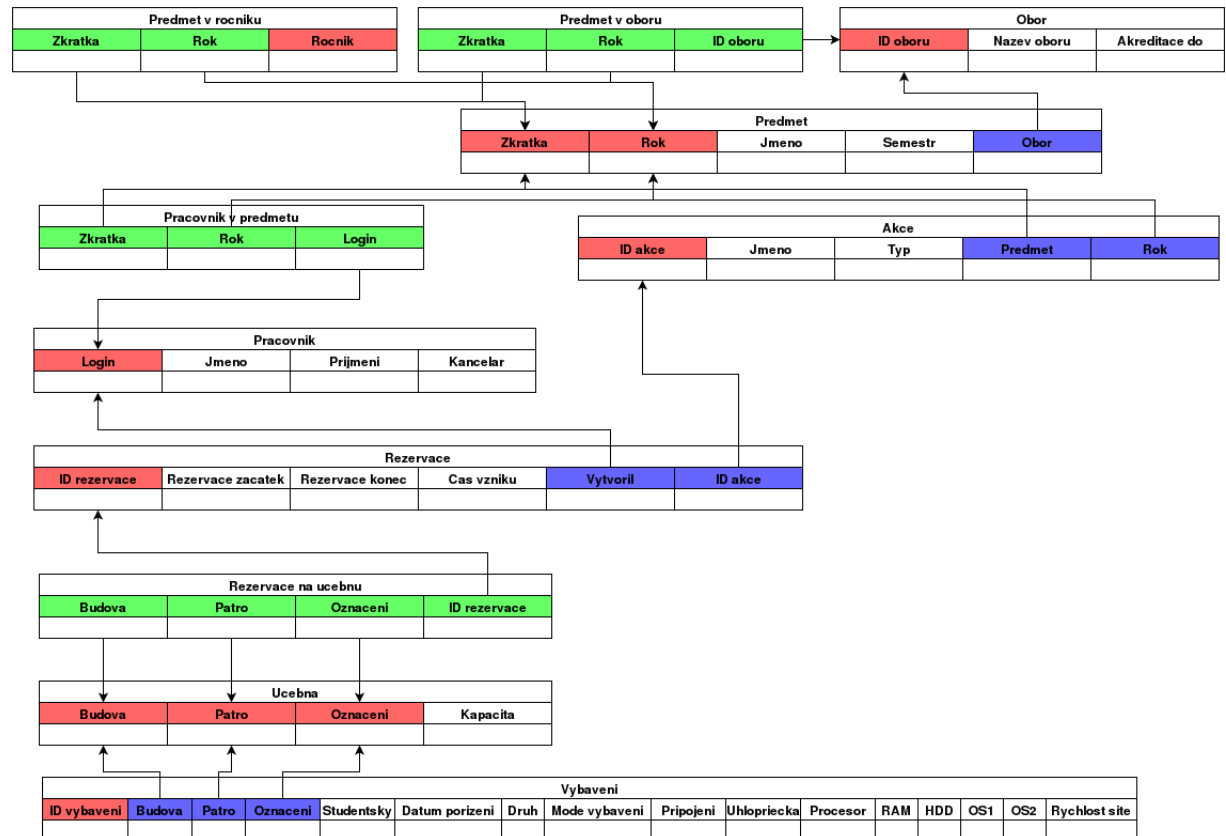


Figure 2: Schéma relační databáze

4 Trigger

4.1 OBOR_TRIG

Tento trigger slouží k automatickému generování primárního klíče do tabulky OBOR ze sekvence ID_OBORU_S. Trigger se spouští před operací insert nad touto tabulkou a do sloupce ID_OBORU vloží další hodnotu ze sekvence. Vložení se nicméně provede pouze

tehdy, když před spuštěním triggeru dotaz obsahuje hodnotu NULL v daném sloupci. Tato vlastnost umožňuje, aby si uživatel mohl explicitně zvolit hodnotu sloupce, aniž by mu tento trigger vnutil hodnotu ze sekvence. Nicméně je potřeba být na pozoru - pokud se trigger spustí a v tabulce již je obsažen záznam s vygenerovanou hodnotou ze sekvence (který tam uživatel pravděpodobně ručně vložil), dotaz selže.

4.2 REZERVACE_NA_UCEBNU_TRIG

Tento trigger kontroluje, zda aktuálně vkládaná rezervace na danou učebnu do tabulky REZERVACE_NA_UCEBNU, nekoliduje s jinou, již vloženou rezervací. Jádrem tohoto triggeru je dotaz, který kontroluje, zda v dané učebně neexistuje v časovém intervalu nově vkládané rezervace už nějaká jiná rezervace. Počet vzniklých kolizí se uloží pomocí operace SELECT INTO do proměnné POCET_K. Následně otestuje, jestli je v této proměnné kladná hodnota. Pokud ano, pomocí funkce RAISE_APPLICATION_ERROR dojde k přechodu do chybového stavu a dotaz se neprovede.

5 Procedúry

Pre danú databázu boli vytvorené dve uložené procedúry.

5.1 PERCENT_DNA

Táto procedúra vypočíta na koľko percent sú využité všetky učebny v deň, ktorý je zadaný argumentom procedúry. Procedúra uvažuje deň od 7:00 do 20:00. Výstupom je číslo v percentách.

Procedúra je užitočná na plánovanie väčších akcií, keď je vhodné si vybrať deň, keď sú čo najmenej vyťažené učebne.

Vrámcí tejto procedúry boli využité aj **výnimky**. Nakoľko sa delí počtom učebien, mohla by nastať situácia keď by v databáze neboli zadané žiadne vyhovujúce učebne a teda by mohlo nastať delenie nulou (ZERO_DIVIDE). Rovnako výnimka kontroluje aj stav, ak by neboli dodané žiadne rezervácie (NO_DATA_FOUND).

5.2 JMENA_UCEBEN

Druhá procedúra sa využíva na výpis skutočných názvov učebien. Táto procedúra je veľmi užitočná, nakoľko dáta v tabuľke UCEBNY sú rozdelené na 3 položky a to BUDOVA, PATRO a OZNACENI. Výpis takýchto troch atribútov môže byť pre užívateľa nepohodlný. Procedúra okrem spájania atribútov v správnom poradí, zároveň dopĺňa nuly (napr. E104 vs E112).

Vrámcí tejto procedúry bol využitý **kurzor**, do ktorého sa vyberú učebne o ktoré je záujem a následne v **for** smyčke sú tieto atribúty editované.

V procedúre sa vyskytuje aj **premenná s dátovým typom odkazujúcim na stĺpec tabuľky**, do ktorej sa ukladá medzivýsledok pri spájaní atribútov.

6 Index a explain plan

Funkce indexu byla demonstrována na dotazu, který filtroval data z tabulky OBOR podle sloupce NAZEV_OBORU. Ve výpisu je jasně vidět, že je nutné přistoupit ke všem datům sloupce v této tabulce (řádek 4) a jednotlivě je porovnat s názvem oboru specifikovaným v dotazu. Jak je možné ve sloupci “time” ve výpise vidět, časová náročnost této operace jsou 3 časové jednotky.

```
('Plan hash value: 1326656592',)
(' ',)
('-----',)
(' | Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time |
('-----',)
(' | 0 | SELECT STATEMENT | | 2 | 228 | 9 (12)| 00:00 |
(' | 1 | HASH GROUP BY | | 2 | 228 | 9 (12)| 00:00 |
(' | 2 | NESTED LOOPS | | 2 | 228 | 8 (0)| 00:00 |
(' |* 3 | HASH JOIN | | 2 | 194 | 6 (0)| 00:00 |
(' |* 4 | TABLE ACCESS FULL| OBOR | 1 | 40 | 3 (0)| 00:00 |
(' | 5 | TABLE ACCESS FULL| PREDMET | 6 | 342 | 3 (0)| 00:00 |
(' |* 6 | INDEX RANGE SCAN | PRAC_V_PREDMETU_PK | 1 | 17 | 1 (0)| 00:00 |
('-----',)
(' ',)
('Predicate Information (identified by operation id):',)
('-----',)
(' ',)
(' 3 - access("OBOR"."ID_OBORU"="PREDMET"."OBOR")',)
(' 4 - filter("OBOR"."NAZEV_OBORU"=\ 'Pocitacove systemy\')',)
(' 6 - access("PREDMET"."ZKRATKA"="PRACOVNIK_V_PREDMETU"."ZKRATKA" AND ',)
(' "PREDMET"."ROK"="PRACOVNIK_V_PREDMETU"."ROK")',)
(' ',)
('Note',)
('-----',)
(' - dynamic statistics used: dynamic sampling (level=2)',)
(' - this is an adaptive plan',)
```

Po přidání indexu na sloupec NAZEV_OBORU v tabulce OBOR je ve výpisu vidět, že místo porovnání názvu oboru v dotazu se všemi prvky v tabulce (což je operace s časovou složitostí $O(n)$, kde n je počet řádků v tabulce), se pro vyhledání řádku využije indexu (při použití například binárního stromu je časová složitost $O(\log n)$). Ve výpise se toto projevilo časovou náročností 2 jednotky. Můžeme proto říci, že zavedením tohoto indexu došlo k urychlení vyhledávání v tabulce OBOR, pokud filtrujeme dle sloupce NAZEV_OBORU.

```
('Plan hash value: 538994587',)
(' ',)
```

```

('-----
(' | Id | Operation | Name | Rows | Bytes |
('-----
(' | 0 | SELECT STATEMENT | | 2 | 228 |
(' | 1 | HASH GROUP BY | | 2 | 228 |
(' | 2 | NESTED LOOPS | | 2 | 228 |
(' |* 3 | HASH JOIN | | 2 | 194 |
(' | 4 | TABLE ACCESS BY INDEX ROWID BATCHED | OBOR | 1 | 40 |
(' |* 5 | INDEX RANGE SCAN | OBOR_INDEX | 1 | |
(' | 6 | TABLE ACCESS FULL | PREDMET | 6 | 342 |
(' |* 7 | INDEX RANGE SCAN | PRAC_V_PREDMETU_PK | 1 | 17 |
('-----
(' ',)
('Predicate Information (identified by operation id):',)
('-----',)
(' ',)
(' 3 - access("OBOR"."ID_OBORU"="PREDMET"."OBOR")',)
(' 5 - access("OBOR"."NAZEV_OBORU"=\ 'Pocitacove systemy\')',)
(' 7 - access("PREDMET"."ZKRATKA"="PRACOVNIK_V_PREDMETU"."ZKRATKA" AND ',)
(' "PREDMET"."ROK"="PRACOVNIK_V_PREDMETU"."ROK")',)
(' ',)
('Note',)
('-----',)
(' - dynamic statistics used: dynamic sampling (level=2)',)
(' - this is an adaptive plan',)

```

7 Prístupové práva pre druhého člena tímu

Väčšina tohto tímového projektu bola vytvorená pri spoločných stretnutiach s využitím priameho prístupu len z jedného účtu. Neskôr avšak pre vyladenie skriptu, sme potrebovali s dátami pracovať obaja. Preto boli nastavené práva príkazom GRANT.

8 Materializovaný pohľad

Materializované pohľady byly vytvořeny na účtu druhého uživatele s využitím práv k tabulkám prvního uživatele.

8.1 PRACOVNICI_PREDMETY

Tento pohled slouží k jednoduchému vyhledávání pracovníků učících jednotlivé předměty nebo naopak, kde nejsou pro vyhledávání použity zkratky předmětu či loginy, nýbrž název předmětu či jméno nebo příjmení pracovníka. Pro optimalizaci dotazů je při

tvorbě pohledu použít parametr **CACHE**, který často vyhledávaná data uloží do vyrovnávací paměti, ze které se v příštím dotazu dají data velmi rychle využít. Parametr **BUILD IMMEDIATE** způsobí, že pohled je vybudován hned. Konečně parametr **REFRESH ON COMMIT** způsobí, že pohled je aktualizován v průběhu jakékoliv změny v tabulkách, na které tento pohled odkazuje. Vzhledem k tomu, že v pohledu jsou spojena data z více tabulek, není ho možné využít na vkládání dat - proto je v SQL skriptu demonstrována nad pohledem pouze operace **SELECT**.

8.1.1 AKTUALNI.PREDMET

Pohled slouží k rychlému přístupu k předmětům, které jsou vyučovány v aktuálním roce. Její využití může mít velký efekt, neboť většina uživatelských akcí v informačním systému se vztahuje na předměty vyučované právě tento školní rok. Opět je použit parametr **CACHE** na vytvoření vyrovnávací paměti.

9 Python App?

10 Záver

Script bol vytvorený a testovaný za pomoci SQL Developer na Oracle servery Gort. Skript správne fungoval vo všetkých častiach a všetky selekty, procedúry, triggery a ostatné boli dôkladne otestované.