

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií

IDS – DATABÁZOVÉ SYSTÉMY
2019/2020



Zadanie č. 53 - Fitness Centrum

Mário Gažo (xgazom00)

Matej Otčenáš (xotcen01)

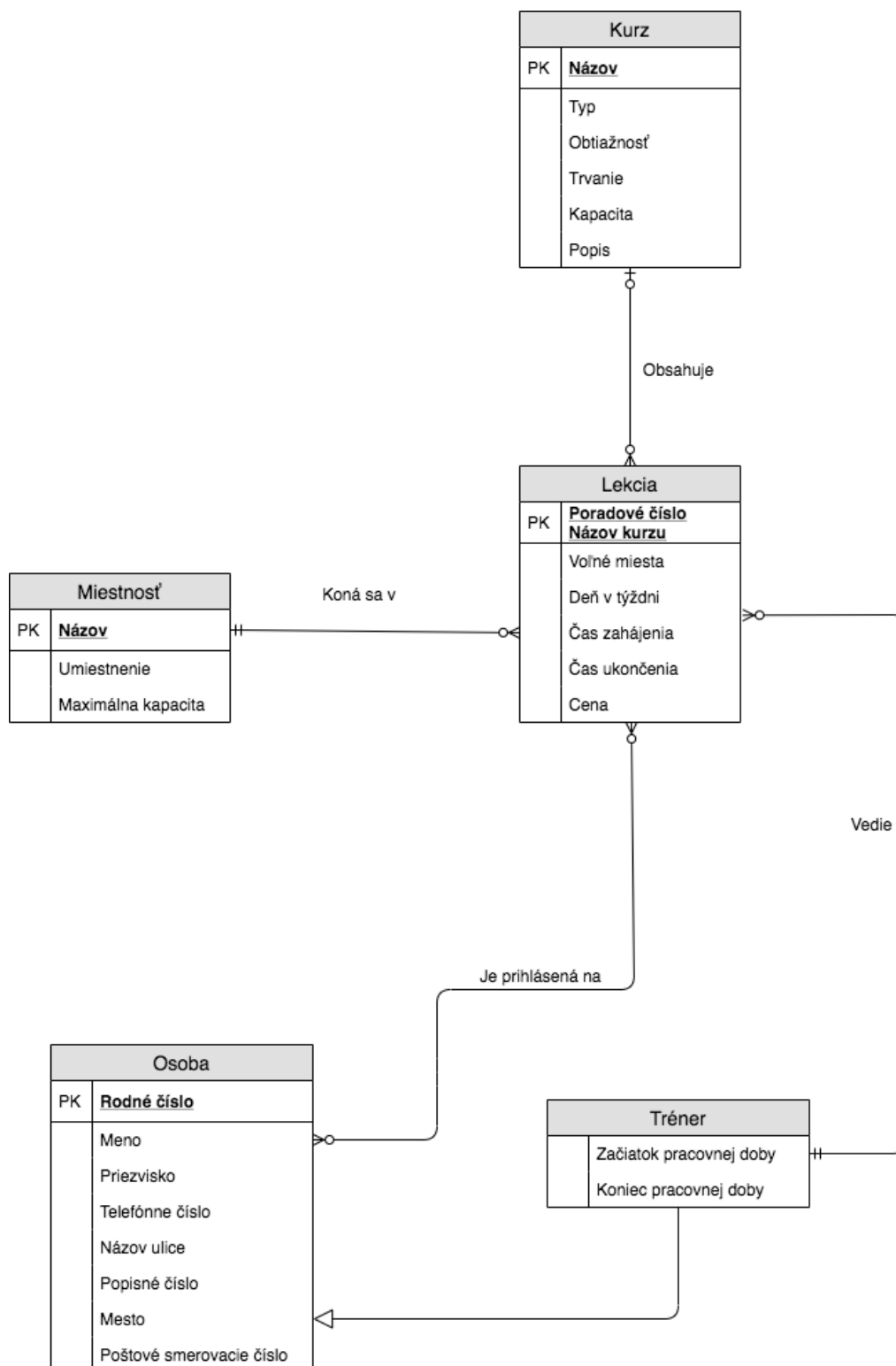
Piešťany, 4. 20.2020

1. Text zadania

Navrhněte jednoduchý IS fitness centra, které organizuje různé kurzy skupinových lekcí (zumba, TRX, kruhový trénink, atd.). Ve fitness centru pracují instruktoři, kteří vedou jednotlivé skupinové lekce, a lidé na recepci, kteří se musí kromě vítání příchozích klientů a mixování proteinových koktejlů zapojit do práce s IS fitness centra prostřednictvím vytváření členských karet pro jednotlivé klienty, kteří se rozhodli pravidelně trápit svá těla ve fitness centru a chtějí využít členské výhody. Aby karta nebyla využívána jinými klienty než jejím vlastníkem, musí být v IS uloženy základní informace o klientech, jejich rodná čísla a adresy. Zákazník si může vypsát kurzy, které navštěvuje a informace o jednotlivých lekcích. Navíc si může zobrazit rozvrh vypisovaných kurzů a zjistit počet volných míst na jednotlivých lekcích a jejich cenu. Zákazník se může registrovat buď na jednu lekci nebo na celý kurz. Kurzy mají svou délku trvání, obtížnost a popis. Skupinové lekce probíhají v různých sálech fitness centra, které mají konkrétní název, umístění a maximální kapacitu. Lekce jsou vedené jedním instruktorem, mají maximální kapacitu účastníků a odehrávají se v daném sále v určitý čas a den v týdnu. Předpokládejte, že jeden instruktor může být vyškolen pro vedení různých kurzů, toto modelujte. Kromě pravidelných skupinových lekcí nabízí fitness centrum i individuální lekce, na kterých se instruktor věnuje pouze jednomu klientovi. Tyto lekce jsou podobného charakteru jako ty skupinové, jen je konkrétnímu klientovi věnováno více pozornosti. Instruktor má možnost vložit do systému nové typy kurzů a konkrétní lekce (a to jak skupinové, tak i individuální) a měnit čas a sál, ve kterém se lekce konají. Systém musí být na požádání schopen vypsát rozvrh pro jednotlivé místnosti.

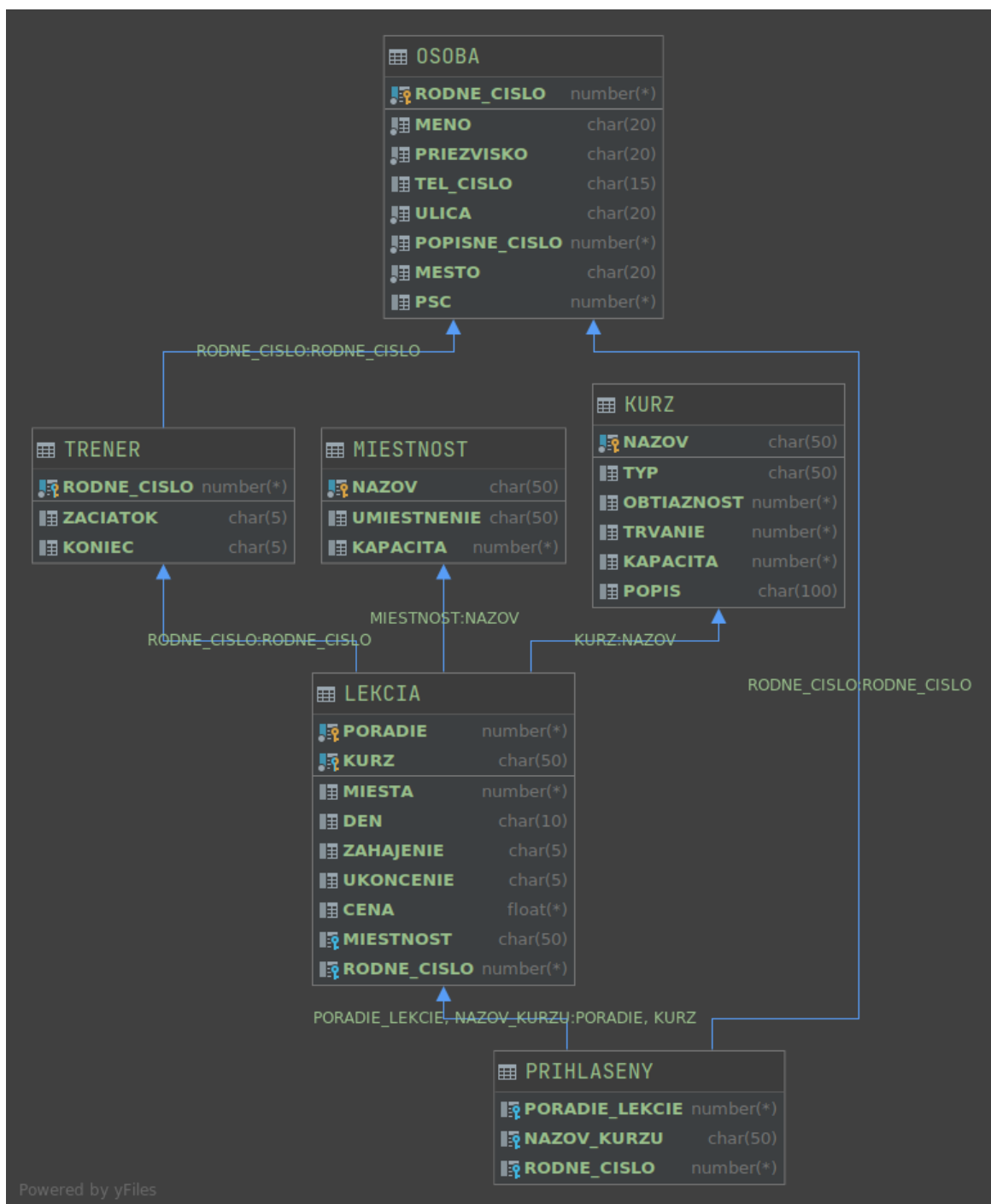
2. ER diagram

Oproti pôvodnému vypracovaniu sme sa rozhodli zrušiť špecializáciu lekcie na individuálne a skupinové, nakoľko tabuľky ktoré ich predstavovali boli zhodné. Nahradili sme ju m:m vzťahom medzi osobou a lekciami. Taktiež sme pridali ďalší primárny kľúč pre identifikáciu lekcí, teda názov kurzu do ktorého patrí.



3. Schéma databáze

Diagram vygenerovaný vývojovým prostredím PHPStrom. Zobrazuje názvy tabuliek, atribúty a ich typy, kľúče a vzťahy medzi jednotlivými tabuľkami.



3.1.Generalizácia

V našej databáze sme generalizáciu využili v tabuľke osoba, kde osobou môžeme rozumieť aj trénera ktorý vedie jednotlivé lekcie, no zároveň môže byť na tieto lekcie prihlásený.

4. Implementácia

4.1. DROP TABLE

Hneď pri spustení skriptu sa zmažú všetky tabuľky, ktoré mohli byť vytvorené pri predchádzajúcich spúšťaniach.

4.2. CREATE TABLE

Nasleduje tvorba databáze. Jednotlivé tabuľky sú pomenované a sú im pridelené atribúty na základe schémy. Taktiež sme definovali kľúče tabuliek teda:

- primárne (Kurz, Miestnosť, Osoba),
- zložený (Lekcia),
- a cudzie (Tréner, Prihlásený).

4.3. TRIGGER

Náš skript obsahuje celkom dva jednoduché trigger-y, oba sú aktivované po príslušnej operácii (AFTER).

- Po vložení (INSERT) nového kurzu do databáze je automaticky to lekcii pridaná úvodná lekcia, ktorá je zadarmo a je označená ako 0.
- Po výmaze (DELETE) kurzu sú zmazané všetky (DELETE) lekcie ktoré patria do tohto kurzu, nakoľko už nemá zmysel aby boli naďalej vedené v databáze.

4.4. INSERT INTO / VALUES

Do tabuliek sú pridané vzorové hodnoty.

4.5. SELECT

Ďalej sú z tabuliek vyberané hodnoty na základe určitých kritérií ktoré špecifikuje query:

- Informácie o lekcií ktorú vedie určitý tréner.
- Informácie o kurze, ktorého piata lekcie sa odohráva v pondelok a stojí viac ako 10.
- Informácie o trénerovi, ktorý vedie lekcie crossfitu s kapacitou miestnosti maximálne 60.
- Vyhľadá názvy kurzov, ktorých trvanie je dlhšie ako 45 minút.
- Vyhľadá názov a typ kurzu, ktorého priemerná obtiažnosť je menšia ako 3.

- Informácie o osobách ktoré sú prihlásené na lekcie.
- Informácie o miestnostiach v ktorých sa odohrávajú lekcie.
- Popis kurzu, ktorého lekcia začína 12:45 a jej tréner začína 17:25.

4.6. FUNCTION

Skript obsahuje dvojicu funkcií ktoré neskôr využívajú procedúry.

- Počítanie celkovej kapacity fitness centra, teda sčítanie kapacity všetkých miestností.
- Výpočet percentuálneho podielu časti z celku.

4.7. PROCEDURE

Procedúry zbierajú informácie a vypisujú ich ako reťazec:

- Informácie o bydlisku osoby s určitým priezviskom.
- Percentuálny podiel jednotlivých miestností na celkovej kapacite fitness centra.

4.8. EXPLAIN PLAN VS EXPLAIN PLAN (s použitím indexu)

Dotaz EXPLAIN PLAN zobrazí plán vykonania dotazov SELECT, UPDATE, INSERT, DELETE. V našom prípade bol použitý dotaz SELECT, kde sme zistili, že celková cena operácií je v súčte 26.

```

1 Plan hash value: 910825460
2
3 -----
4 | Id | Operation          | Name          | Rows | Bytes | Cost (%CPU)| Time     |
5 -----
6 |  0 | SELECT STATEMENT   |               |      5 | 580 | 7 (15)| 00:00:01 |
7 |*  1 | FILTER             |               |      |      |      |          |
8 |  2 | HASH GROUP BY      |               |      5 | 580 | 7 (15)| 00:00:01 |
9 |*  3 | HASH JOIN          |               |      5 | 580 | 6 (0)| 00:00:01 |
10 |  4 | TABLE ACCESS FULL| PRIHLASENY    |      3 | 39 | 3 (0)| 00:00:01 |
11 |  5 | TABLE ACCESS FULL| LEKCIA        |      6 | 618 | 3 (0)| 00:00:01 |
12 -----
13
14 Predicate Information (identified by operation id):
15 -----
16
17 1 - filter(SUM("LEKCIA"."CENA")/COUNT("LEKCIA"."CENA")<10)
18 3 - access("LEKCIA"."RODNE_CISLO"="PRIHLASENY"."RODNE_CISLO")
19
20 Note
21 ----
22 - dynamic statistics used: dynamic sampling (level=2)

```

(explain plan)

Pri identickom dotaze EXPLAIN PLAN avšak so zavedením indexácie CREATE INDEX (indexovanie bolo zvolené pre tabuľku *Prihlaseny* s najčastejšie používanou položkou *Rodne_cislo*) môžeme vidieť, že celková cena operácií klesne až na hodnotu 14. Indexovanie nám dokázalo značne databázu optimalizovať.

```

1 Plan hash value: 662839363
2
3 -----
4 | Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |
5 -----
6 | 0 | SELECT STATEMENT | | 5 | 580 | 4 (25) | 00:00:01 |
7 |* 1 | FILTER | | | | | |
8 | 2 | HASH GROUP BY | | 5 | 580 | 4 (25) | 00:00:01 |
9 | 3 | NESTED LOOPS | | 5 | 580 | 3 (0) | 00:00:01 |
10 | 4 | TABLE ACCESS FULL | LEKCIA | 6 | 618 | 3 (0) | 00:00:01 |
11 |* 5 | INDEX RANGE SCAN | IDX | 1 | 13 | 0 (0) | 00:00:01 |
12 -----
13
14 Predicate Information (identified by operation id):
15 -----
16
17 1 - filter(SUM("LEKCIA"."CENA")/COUNT("LEKCIA"."CENA")<10)
18 5 - access("LEKCIA"."RODNE_CISLO"="PRIHLASENY"."RODNE_CISLO")
19
20 Note
21 ----
22 - dynamic statistics used: dynamic sampling (level=2)

```

(explain plan indexed)

4.9. GRANT

Pridelenie práv na zobrazenie tabuliek a na používanie funkcií a procedúr, pridáva vedúci tímu členovi tímu.

4.10. MATERIALIZED VIEW

Materializovaný pohľad *mat_pohľad* je vytvorený s použitím jednoduchého dotazu SELECT, pre určenie ktoré z osôb sú trénermi.

Parameter CACHE zabezpečí ukladanie často vyhľadávaných dát do pamäti cache (vyrovnávacia pamäť), vďaka čomu sa zvýši rýchlosť a efektivita.

Parameter BUILD IMMEDIATE okamžite vytvorí pohľad.

Parameter REFRESH ON COMMIT zabezpečí aktualizáciu pohľadu pri zmene v tabuľke *Osoba*.

5. Záver

Skript bol písaný v SQL dialekte Oracle a testovaný na školskom servery ktorý nám bol poskytnutý. Pre ladenie a generovanie schémy bolo použité vývojové prostredie PHPStorm. Informácie k vypracovaniu sme čerpali z opory predmetu IDS, oficiálnej dokumentácie na stránke Oracle a zo StackOverflow.