

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Fitness Centrum - Zadanie č. 58

DATABÁZOVÉ SYSTÉMY

2020/2021

ZOZNAM AUTOROV:

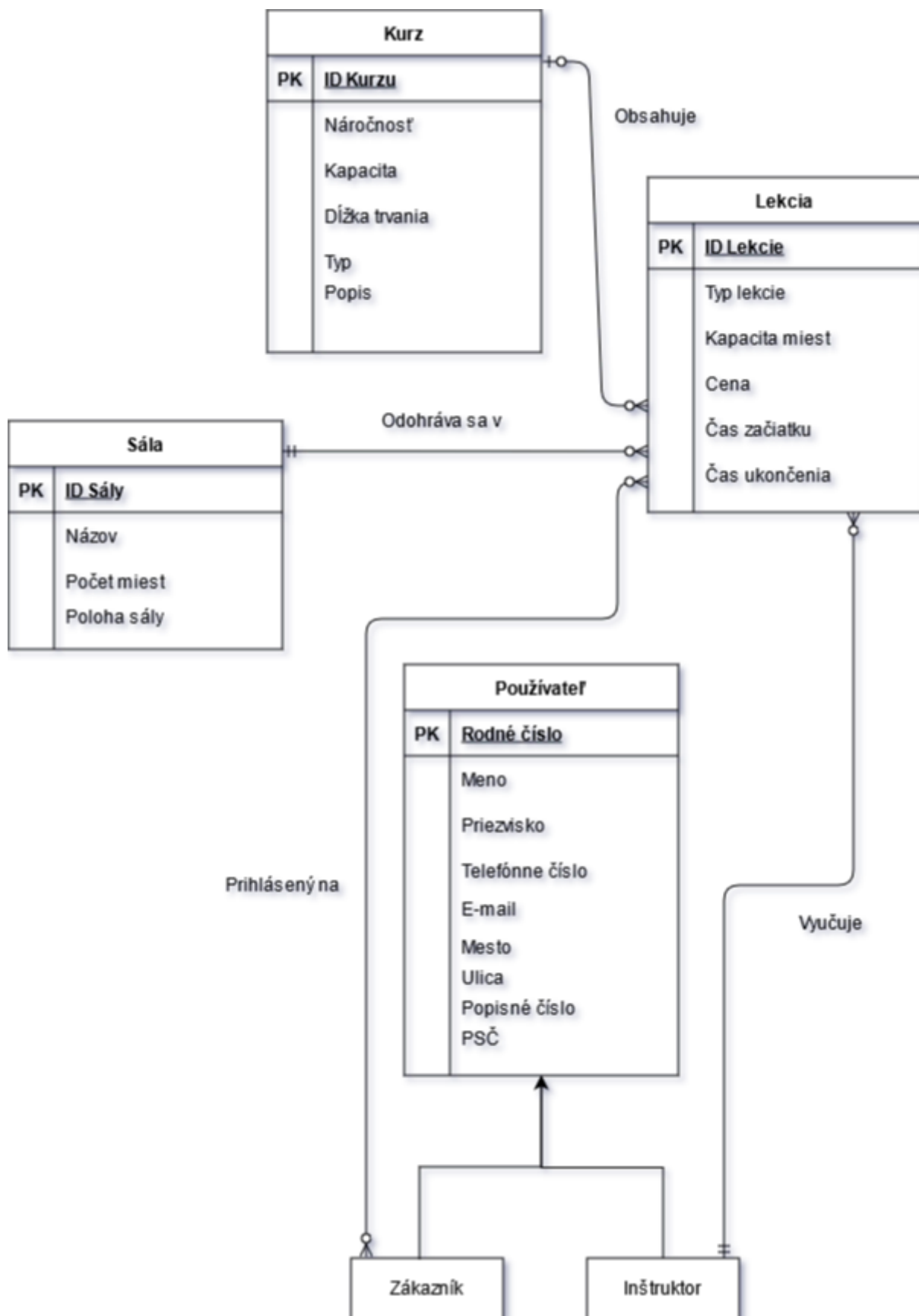
DUŠAN ČIČMIŠ (XCICMI00)

TOMÁŠ ČECHVALA (XCECHV03)

1 Zadanie projektu

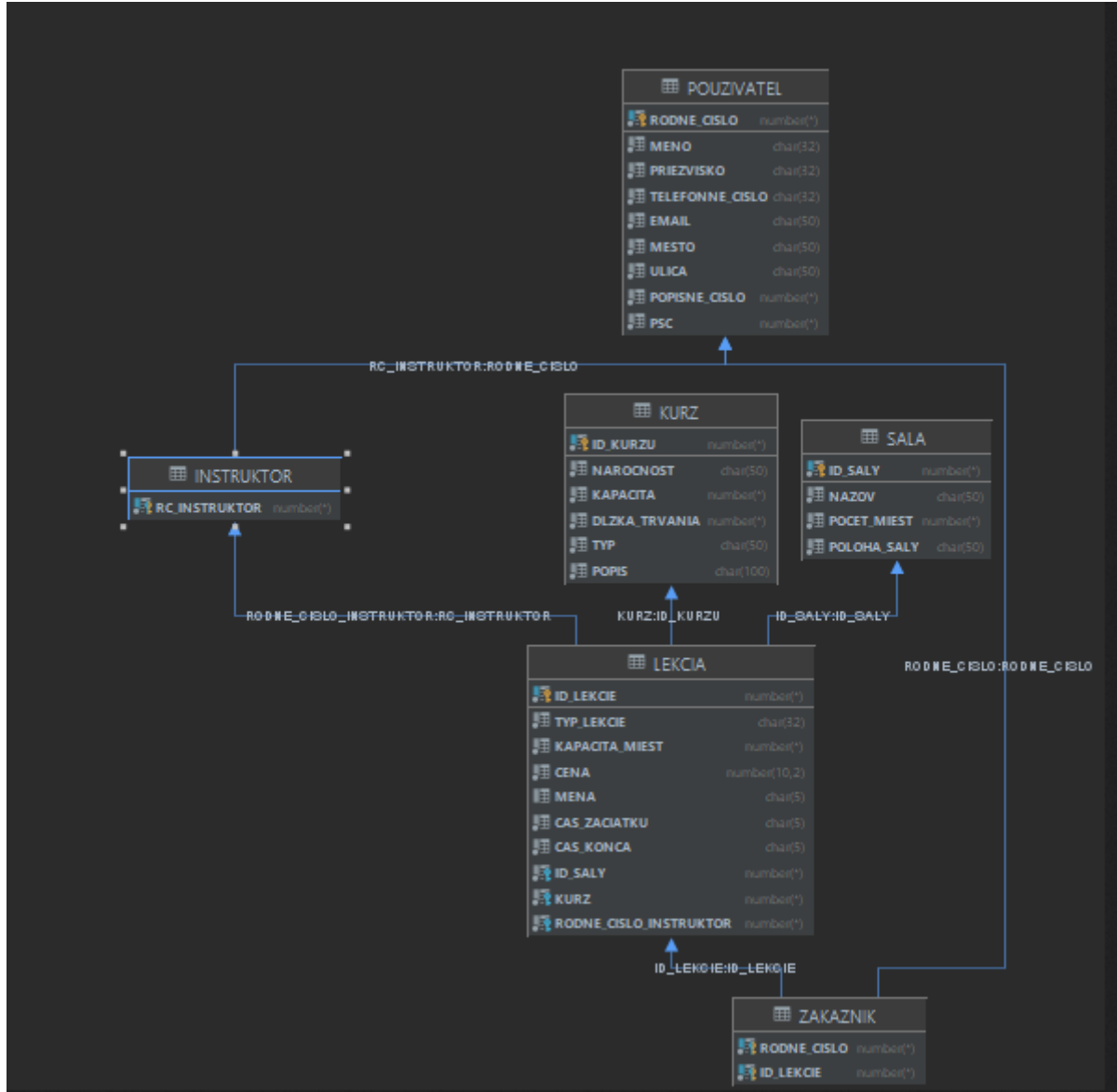
Navrhněte jednoduchý IS fitness centra, které organizuje různé kurzy skupinových lekcí (zumba, TRX, kruhový trénink, atd.). Ve fitness centru pracují instruktoři, kteří vedou jednotlivé skupinové lekce, a lidé na recepci, kteří se musí kromě vítání příchozích klientů a mixování proteinových koktejlů 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



3 Implementácia databázy

Nasledujúci diagram databázy zobrazuje názvy tabuliek, atribúty ich typy, kľúče a vzťahy medzi jednotlivými tabuľkami.



3.1 Generalizácia

Generalizáciu sme použili v tabuľke Používateľ, kde pod používateľom rozumieme zákazníka, ktorý môže byť prihlásený na lekcie a inštruktora, ktorý vyučuje lekcie.

3.2 DROP TABLE

Po spustení skriptu sa zmažú všetky tabuľky, ktoré sme vytvorili pri predchádzajúcich spusteniach.

3.3 CREATE TABLE

Vytvoria sa tabuľky, ktoré sú pomenované a majú pridelené atribúty a väzby na základe schémy databázy.

Kľúče daných tabuliek:

- primárne(Kurz, Pouzivatel, Sala, Lekcia)
- zložené(Zakaznik)
- cudzie(Instruktor)

3.4 CREATE TRIGGER

Triggery sa aktivujú po operácii (AFTER).

- po zmazení kurzu sa zmaže aj jeho lekcia, keď že už nie je potrebné uchovávať tieto informácie
- po vyhodení inštruktora je inštruktor presunutý do kancelárie raditeľa, kde dostane výpoveď

3.5 INSERT INTO

Vložíme dáta do jednotlivých tabuliek.

3.6 SELECT

Z tabuliek sú podľa daných podmienok vybrané hodnoty, ktoré sú bližšie špecifikované v SELECTE.

- Vypíše klientov a ich zapísané lekcie (Spájanie 3 tabuliek).
- Vypíše lekcie, ktoré sú lacnejšie ako 50€.
- Vypíše lekcie, do ktorých kurzov patria + ich cenu (Spájanie 2 tabuliek).
- Vypíše počet zákazníkov z daného mesta a zoradí ich zostupne (GROUP BY).
- Vypíše počet ťažkých, stredne ťažkých a ľahkých kurzov.
- Vypíše zákazníkov, ktorí majú zapísanú aspoň jednu lekciiu (Spájanie 2 tabuliek + EXISTS).
- Vypíše kurzy, ktoré začínajú o 8:00 alebo 12:00.

3.7 FUNCTION

3.8 PROCEDURE

3.9 EXPLAIN PLAN BEZ INDEXU

Pre dotaz EXPLAIN PLAN sme použili dotaz SELECT a zistili, že celkový počet operácii je 26, obrázok nižšie.

```
Plan hash value: 3297457190

-----
| Id | Operation          | Name                | Rows | Bytes | Cost (%CPU)| Time     |
-----+-----+-----+-----+-----+-----+-----+
| 0  | SELECT STATEMENT   |                     |      |       |             |          |
| 1  |   SORT ORDER BY    |                     |      |       |             |          |
|* 2  |    HASH JOIN SEMI  |                     |      |       |             |          |
| 3  |     TABLE ACCESS FULL | POUZIVATEL          |    11 |    891 |  3 (0) | 00:00:01 |
| 4  |      INDEX FAST FULL SCAN| SYS_C002480001      |    17 |    221 |  2 (0) | 00:00:01 |
-----

PLAN_TABLE_OUTPUT

Predicate Information (identified by operation id):
-----

   2 - access("POUZIVATEL"."RODNE_CISLO"="ZAKAZNIK"."RODNE_CISLO")

Note
-----
   - dynamic statistics used: dynamic sampling (level=2)
   - this is an adaptive plan

21 rows selected.
```

3.10 EXPLAIN PLAN S INDEXOM

EXPLAIN PLAN s indexom sme použili na ten istý dotaz SELECT a výsledný počet operácii je 14, tým pádom sme dosiahli optimalizáciu tabuľky vďaka indexu.

```
Plan hash value: 2038743016

-----
| Id | Operation          | Name                | Rows | Bytes | Cost (%CPU)| Time     |
-----+-----+-----+-----+-----+-----+
| 0  | SELECT STATEMENT   |                     |      |       |             |          |
| 1  |   SORT ORDER BY    |                     |      |       |             |          |
| 2  |    NESTED LOOPS SEMI |                     |      |       |             |          |
| 3  |     TABLE ACCESS FULL| POUZIVATEL          |    11 |    891 |  3 (0) | 00:00:01 |
|* 4  |      INDEX RANGE SCAN| IDX                  |    11 |    143 |  0 (0) | 00:00:01 |
-----

PLAN_TABLE_OUTPUT

Predicate Information (identified by operation id):
-----

   4 - access("POUZIVATEL"."RODNE_CISLO"="ZAKAZNIK"."RODNE_CISLO")

Note
-----
   - dynamic statistics used: dynamic sampling (level=2)

20 rows selected.
```

3.11 GRANT

Vedúci tímu prideluje práva ďalšiemu členovi na zobrazenie tabuliek a manipuláciu s funkciami a procedúrami.

3.12 MATERIALIZED VIEW

Materializovaný pohľad vygenerovaný pomocou dotazu SELECT na určenie používateľov, ktorí sú inštruktormi. CACHE - slúži na ukladanie často hľadaných dát do vyrovnávacej pamäti, kvôli tomu

sa zvýši rýchlosť. BUILD IMMEDIATE - okamžite vytvorí pohľad. REFRESH ON COMMIT - pri zmene dát v tabuľke sa materializovaný pohľad aktualizuje.

4 Záver

Skript je napísaný v SQL Oracle, testovanie prebiehalo na pridelenom školskom serveri. Informácie sme čerpali z opory predmetu, prednášok a z internetu. Pre vytvorenie databázy sme používali program SQL developer, ktorý bol ukázaný na demonštračnom cvičení.