

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií

IDS – DATABÁZOVÉ SYSTÉMY

2020/2021



### **Zadani č. 53 - Fitness Centrum**

Danylo Pimenov (xpimen00)

Prokofiev Oleksandr (xproko40)

29 dubna 2021

## Uvod

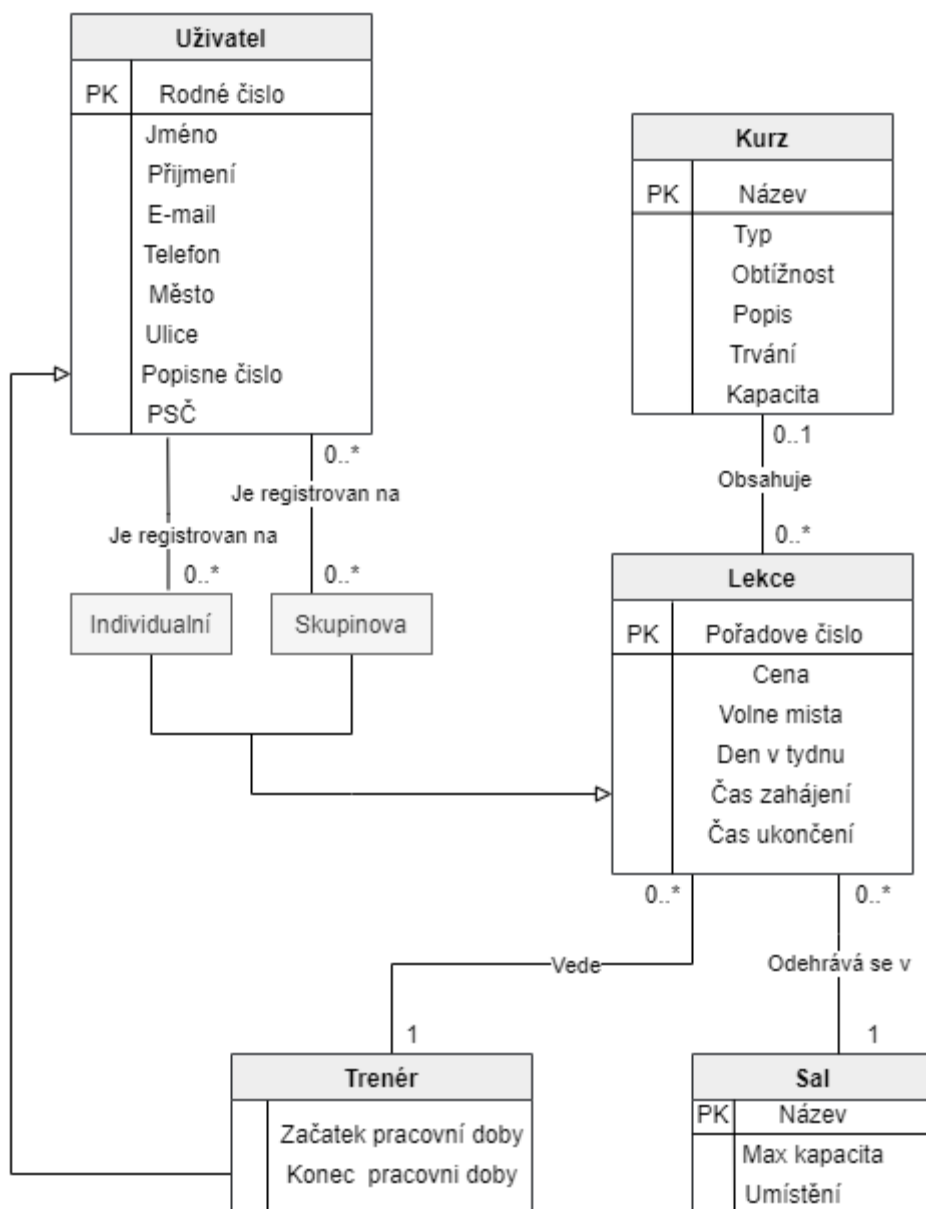
Cílem projektu bylo vytvořit skript v jazyce SQL, který bude obsahovat základní a pokročilý objekty schématu databáze. Kromě toho projekt byl zaměřen na seznámení s prostředím, které jsme využívaly k řešení projektů (vývojové prostředí Oracle SQL Developer a jazyk PL/SQL).

## Text zadání

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.

## Návrhový diagram

Podle zadání jsme vytvořili ER diagram, které současně byl naším návrhovým diagramem. V naší databázi jsme generalizací využili na entytách Uživatel (znamena že trenér je uživatel který vede jednotlivé lekce a zároveň může být na ně přihlášen) a Lekce (rozdělí Lekce do dvou typů Individualní a Skupinové).



## Implementace

Na začátku skript prochází blokem DROP, který smaže všechny tabulky, které mohly být vytvořeny při předchozích spouštění, tento blok pomáhá vyhnout se neočekávaná data v databáze.

Po smazání začíná se blok CREATE něm vytváříme tabulky předáváme jména a přiřazujeme atributy na základě návrhového diagramu.

Po bloku CREATE jde blok TRIGGER. Náš skript obsahuje celkem dva jednoduché trigger-y, oba jsou aktivovány po příslušné operaci(AFTER).

- Po vložení (INSERT) do databáze nového kurzu automaticky přidá se bezplatní úvodní lekci a bude označena jako 0 nulová .
- Po odstranění (DELETE) kurzu budou odstraněny všechny lekce (DELETE), které do tohoto kurzu patří.

Po bloku TRIGGER blok INSERT do tabulek přidá vzorové hodnoty. Dale jde blok SELECT v selekt-u nacházíme hodnoty, které odpovídají konkrétnímu dotazu.

- Informace o lekci kterou vede určitý trenér.
- Informace o kurzu, která třetí lekce se odehrává v pondělí a stojí víc než 15 eur.
- Informace o trenérovi, který vede lekce Body\_form s kapacitou místnosti maximálně 50.
- Vyhleda názvy kurzu, které trvají dele 45 minut.
- Vyhleda názvy a typ kurzu, který má průměrnou obtížnost menší než 3.
- Informace o uživatelích která jsou přihlášené na lekce.
- Informace o místnosti ve které se odehrává lekce.
- Popis kurzu, která lekce začíná 12:45 a ní trenér začíná 17:25.

Po bloku SELECT jdou 2 funkce a 2 procedury:

### FUNKCE

- Počíta celkovou kapacitu fitness centra(součet kapacity všech místností).
- Výpočet procentního podílu části z celku.

### PROCEDURY

- Vypočítava procentuální podíl jednotlivých místností na celkové kapacitě fitness centra.
- Nachází informace o bydlišti osoby s určitým příjmením.

## EXPLAIN PLAN VS EXPLAIN PLAN (s použitím indexu)

Dotaz EXPLAIN PLAN zobrazí plán provedení dotazů SELECT, UPDATE, INSERT, DELETE. V našem případě byl použit dotaz SELECT, kde jsme zjistili, že celková cena operací je v součtu 26.

```
Plan hash value: 2715296001
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		3	348	7 (15)	00:00:01
* 1	FILTER					
2	HASH GROUP BY		3	348	7 (15)	00:00:01
* 3	HASH JOIN		3	348	6 (0)	00:00:01
4	TABLE ACCESS FULL	PRIHLASENY	3	39	3 (0)	00:00:01
5	TABLE ACCESS FULL	LEKCE	6	618	3 (0)	00:00:01

PLAN\_TABLE\_OUTPUT

Predicate Information (identified by operation id):

```
1 - filter(SUM("LEKCE"."CENA")/COUNT("LEKCE"."CENA")<10)
3 - access("LEKCE"."RODNE_CISLO"="PRIHLASENY"."RODNE_CISLO")
```

Note

-----

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

Při identickém dotazu EXPLAIN PLAN se zavedením indexace CREATE INDEX (indexování bylo zvoleno pro tabulku *Prihlaseny* s nejčastěji používanou položkou *Rodne\_cislo*) můžeme vidět, že celková cena operací klesne až na hodnotu 14.

```
Plan hash value: 239537418
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		3	348	4 (25)	00:00:01
* 1	FILTER					
2	HASH GROUP BY		3	348	4 (25)	00:00:01
3	NESTED LOOPS		3	348	3 (0)	00:00:01
4	TABLE ACCESS FULL	LEKCE	6	618	3 (0)	00:00:01
* 5	INDEX RANGE SCAN	IDX	1	13	0 (0)	00:00:01

PLAN\_TABLE\_OUTPUT

Predicate Information (identified by operation id):

```
1 - filter(SUM("LEKCE"."CENA")/COUNT("LEKCE"."CENA")<10)
5 - access("LEKCE"."RODNE_CISLO"="PRIHLASENY"."RODNE_CISLO")
```

Note

-----

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

## GRANT

Přidělení práv k zobrazení tabulek a na používání funkcí a procedur, přiděluje vedoucí týmu členu týmu.

## MATERIALIZED VIEW

Materializovaný pohled *mat\_pohled* je vytvořen s použitím jednoduchého dotazu SELECT, pro určení kdo z uživatelů jsou trenéry.

Parameter CACHE zajistí ukládání často vyhledávaných dat do paměti cache (vyrovnávací paměť), díky čemu se zvýší rychlost a efektivita.

Parameter BUILD IMMEDIATE okamžitě vytvoří pohled.

Parameter REFRESH ON COMMIT zabezpečí aktualizaci pohledu při změně v tabulce *Uživatel*.

## **Záver**

Skript byl napsán v SQL dialekte Oracle a testován na školském serveru. Informácie k vypracování jsme čerpali z opory předmětu IDS, oficiální dokumentace na stránce Oracle a ze StackOverflow.