



## 1. Úvod

Pre testovanie dátovej vrstvy som použila aplikáciu zo stránky dbedu, ku ktorej som pridala vlastné ORM. Aplikácia obsahuje základné CRUD operácie + vybrané operácie z funkčnej analýzy zvolené pre testovanie. Pre testovanie som vybrala 8 funkcií, pričom 2 z nich po zmene využívajú stránkovanie (tm4, ti3), 5 z nich využíva novo-vytvorené indexy (s4, s2, pth4, p9, ti4) a 1 z nich vkladá dáta do tabuľky (pi1). Pomenovanie operácií v tomto texte je zhodné s pomenovaním z predchádzajúcej úlohy projektu.

## 2. Spôsob generovania parametrov

Pri operáciách, ktoré využívali parametre, ktoré boli celé čísla som toto číslo vygenerovala v rozmedzí +- 1 až počet záznamov v tabuľke, ku ktorej sa parameter vzťahuje. Napr. ak operácia prijímala parameter *playerID* číslo bolo zvolené od 1 do maximálneho ID v tabuľke *Player*.

Pri operácii vkladania do tabuľky (pi1) je ako parameter poslaný objekt triedy *Pitch*. V tomto prípade som vygenerovala objekt s náhodnými dátami za použitia zabudovanej

```
public static Pitch FakeData()
{
    var faker = new Faker("cz");
    var size = 100;

    return new Pitch()
    {
        capacity = faker.Random.Int(50, 2000),
        name = faker.Random.Chars(count: size).ToString()
    };
}
```

knižnice *Faker*.

## 3. Vykonávanie operácií

Počet vykonávania operácií bol odvodený od odhadovaného reálneho počtu použitia v IS, kde početnosť 1 odpovedá 100 opakovaniam, početnosť 2 odpovedá 200 opakovania, atď.

Operácia	Tabuľka	Veľkosť výsledku	Počet opakovaní
tm4	TeamMatch	20 (po stránkovaní) - +- 1 200 000 (pred)	100
ti3	Ticket	20 (po stránkovaní) - +- 3 500 000 (pred)	100
s4	Statistic	-	400
s2	Statistic	1	1000
pth4	PlayerTransferHistory	-	300
p9	Player	1	500
ti4	Ticket	-	300
pi1	Pitch	1	100
Operácií celkom			2800

#### 4. Výsledky

Testované v škole, pripojením na školskú databázu. Testovanie som spúšťala s 8 vláknami a výsledne časy sú priemery viacerých testov.

	Počet operácií	Čas vykonania [s]	Počet operácií/s
Pred zmenou	2800	1564s	1,79
Po zmene	2800	11s	254,54

Z tabuľky je jasné, že došlo k výraznému zlepšeniu dátovej vrstvy. Najviac sa o to zaslúžilo stránkovanie, ktoré **pred zmenou** zaberalo jednoznačne najviac času, aj keď počet opakovaní bol najmenší. Treba poznamenať, že v prípade stránkovania hralo rolu to, že sa vyberalo z najväčších tabuliek v databáze. Pri výbere z menších tabuliek bol čas vykonania pred zmenou menší (zhruba o polovicu), avšak stále znateľne väčší ako **po zmene**.

Na výsledkoch vidíme zlepšenie zhruba **143x** (v prípade stránkovania v menších tabuľkách **71x**) – môžeme teda povedať, že zlepšenie fyzického návrhu bolo úspešné.