

Vysoké Učení Technické v Brne  
Fakulta Informačných technológií

Projekt, Databázové systémy 2021  
Dokumentácia

Veronika Vengerová – xvenge01

Matej Viskupič – xvisku01

25.4.2021

# Obsah

1. Zadanie.....	3
2. Diagramy .....	4
2.1 ER diagram.....	4
2.2 Use Case diagram.....	5
3. Riešenie.....	6
3.1 Zahodenie existujúcich dát .....	6
3.2 Vytvorenie tabuliek.....	6
3.3 Naplnenie dátami .....	6
3.4 Dotazy nad databázou .....	7
3.5 Nastavenie privilégii .....	7
3.6 Pokročilé objekty schémy databázy .....	7
3.7 Transakcie.....	8

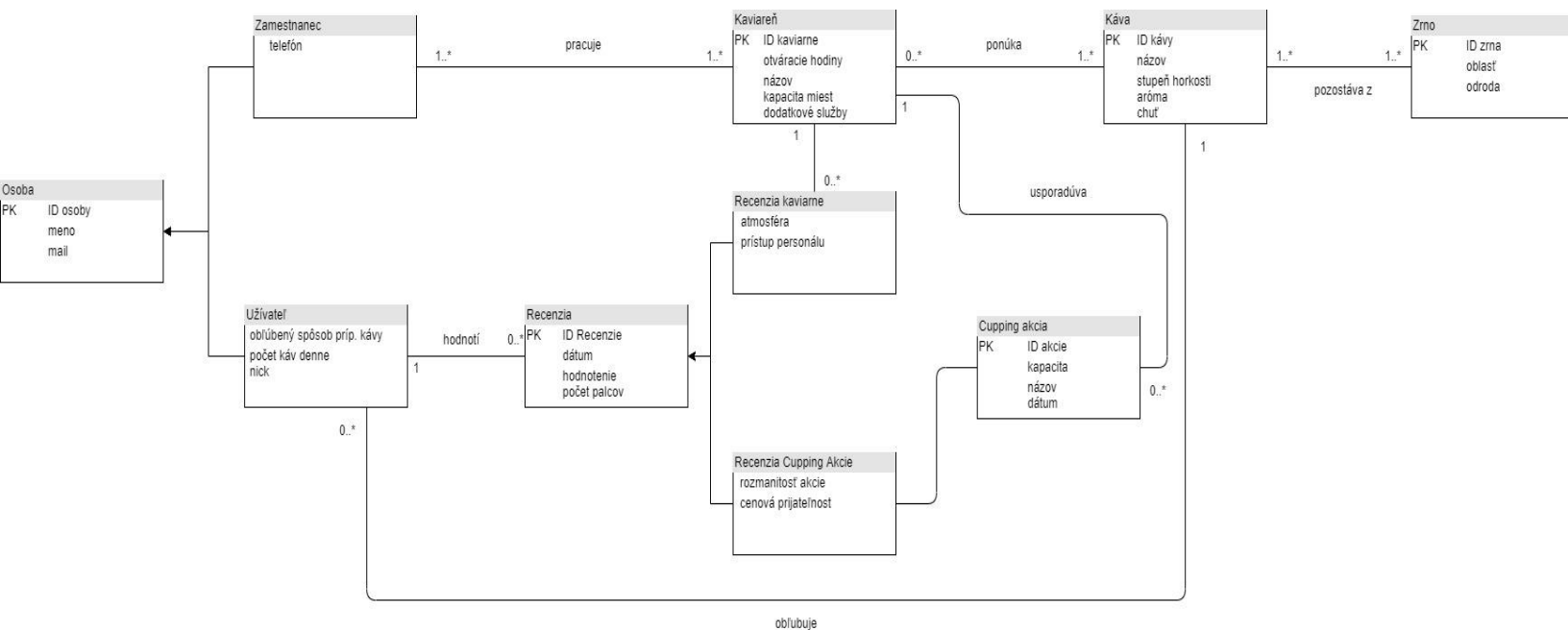
## 1. Zadanie

Zvolili sme si pokračovať v projekte z predmetu IUS, konkrétne v zadaní 61. Kavárenský Povaleč. Jeho zadanie je nasledovné:

*Představte si, že jste kavárenský povaleč, který tráví celé dny v kavárnách po celém Brně, a protože jste už vyzkoušeli spoustu kaváren a ne vždy vám v každé kavárně káva chutnala, rozhodli jste se vytvořit kavárenskou komunitu, ve které by si mohli lidé sdílet informace o brněnských kavárnách. Kavárny nabízejí kávy připravované z různých směsí kávových zrn, přičemž kávová zrna se liší odrůdou, stupněm kyselosti, aromatem, ale i chutí. Uživatel, který musí uvést kromě svých základních informací i údaje o oblíbeném druhu přípravy kávy (espresso, cappuccino, flat white, atd.), oblíbené kavárně, oblíbeném druhu kávy a počtu vypitých káv denně, může psát recenze k jednotlivým kavárnám, které navštívil. Pokud se tak uživatel kávové komunity rozhodne navštívit nějakou kavárnu, může si díky kavárenské komunitě zjistit informace o tom, kde se kavárna nachází, jaké má otevírací hodiny, kapacitu míst a její popis a informace o jejích zaměstnancích. Dále si může uživatel vyhledat jednotlivé recenze a podle toho se rozhodnout, zda kavárnu navštíví nebo nikoliv. V případě návštěvy kavárny, pak sám může buď sepsat recenzi, ve které uvede, jak se mu kavárna líbila, přidělí jí určitý počet hvězdiček a uvede, kdy kavárnu navštívil, nebo může reagovat na recenze ostatních uživatelů a uvést v reakci svůj vlastní názor. V systému se uchová nejen datum, kdy byla reakce napsána, ale i počet palců nahoru nebo dolů, které byly reakci přiděleny. Na recenze mohou kromě uživatelů reagovat i samotní zaměstnanci kaváren, aby se mohli ohradit vůči pomluvám. Každá kavárna má kromě zaměstnanců také svého majitele, a protože brněnské kavárny jsou většinou malé, předpokládejte, že tento majitel zároveň i vypomáhá jako řadový zaměstnanec v kavárně. Aby majitel přilákal více návštěvníků, pořádá často ve svých kavárnách tzv. cupping akce, na kterých můžou návštěvníci ochutnat různé druhy kávy. O jednotlivých druzích kávy, které se na cupping akcích obvykle ochutnávají, si může uživatel přečíst informace, a to konkrétně o oblasti původu, kvalitě a popisu chuti. Uživatelé si mohou u jednotlivých kaváren zjistit, kdy se nějaká cupping akce koná, jaká je cena ochutnávky a zda jsou ještě nějaká volná místa. Pokud ano, mohou se akce zúčastnit a napsat na ni recenzi.*

## 2. Diagramy

### 2.1 ER diagram



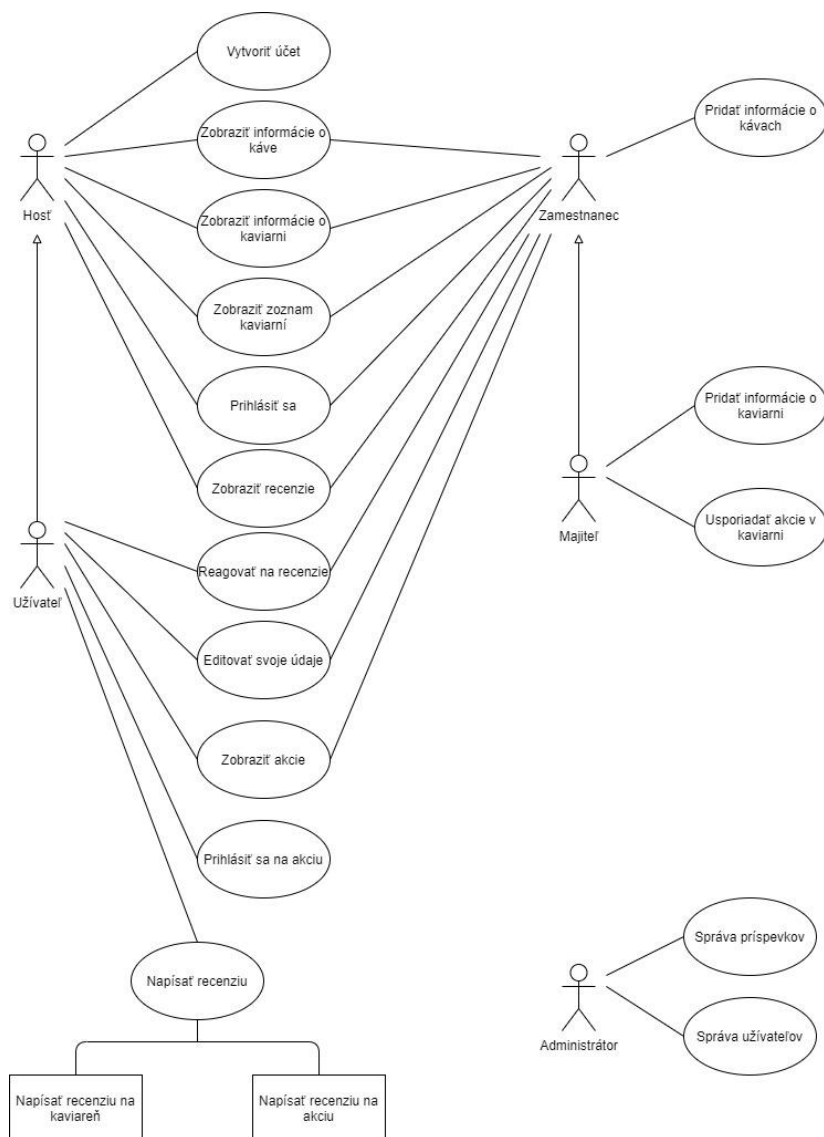
Na priloženom ER diagrame (v prílohe priložený v lepšom rozlíšení) zobrazujeme návrh pre aplikáciu „Kaviarenský povalač“.

Aplikácia si v prvom rade bude udržiavať prehľad o osobách, ktoré s ňou interagujú. Na rozlíšenie užívateľov sme zvolili generalizáciu entity „Osoba“, keďže chceme jasne odlíšiť majiteľov/zamestnancov kaviarne od užívateľov a ich vzťahy s kaviarňou.

Rovnako sme sa rozhodli pre generalizáciu entity „Recenzia“. Považujeme za dostatočne rozdielnu recenziu kaviarne od recenzie Cupping Akcie (akcie na zvýšenie povedomia o kaviarni) a chceme tento rozdiel premietnuť aj do samotného systému. Spomínaná Cupping Akcia je akcia na ktorej sa koná ochutnávka rôznych druhov káv nielen ponúkaných kaviarňami. Z dôvodu, že je podstatné odkomunikovať užívateľom konanie podobnej akcie a uchovávať o nej informácie, zahrnuli sme ju do modelu ako samostatnú entitu.

Systém by si samozrejme mal udržiavať informácie o jednotlivých kaviarňach, nielen nimi ponúkaných kávach a zrne, z ktorého sa dané kávy pripravujú. Preto sú všetky tieto objekty reálneho sveta zachytené v našom návrhu ako samostatné entity s príslušnými vzťahmi.

## 2.2 Use Case diagram



Na priloženom Usecase diagrame (v prílohe priložený v lepšom rozlíšení) reprezentujeme interakciu účastníkov so systémom pre aplikáciu „Kaviarenský povaľáč“.

Najskôr sme si rozlíšili 5 účastníkov pracujúcich so systémom. A postupne im pridelovali prípady využitia a interakcie.

Host' (do systému neprihlásený užívateľ zobrazujúci si aplikáciu) si môže v aplikácii vytvoriť účet alebo sa prihlásiť do už vytvoreného účtu (host' sa môže prihlásiť ako užívateľ alebo ako zamestnanec). Bez toho aby sa prihlásil si host' môže iba zobrazovať informácie o kaviarňach, káve a zoznam kaviarní v systéme, no nemá prístup k zobrazeniu špeciálnych cupping akcií a nemôže vytvoriť ani reagovať na recenzie.

Užívateľ (do systému prihlásený užívateľ) môže vykonávať všetky prípady využitia host'a ale navyše môže reagovať na už existujúce recenzie, editovať svoje údaje, zobrazovať si a prihlásiť sa na špeciálne cupping akcie kaviarní a napísať recenzie na kaviarne a cupping akcie.

Zamestnanec si vie zobrazit' informácie o kaviarňach, káve editovať svoje údaje a zobrazit' akcie. Taktiež umožňujeme zamestnancovi reagovať na recenziu či už aby sa mohli zamestnanci brániť pri nepriaznivých recenziách alebo poďakovať za pozitívnu recenziu. Zamestnanec môže taktiež pridávať informácie o kávach.

Majiteľ môže vykonávať všetky prípady využitia zamestnanca, ale vie navyše pridávať informácie o svojej kaviarni a usporadovať v nej cupping akcie.

Posledným účastníkom je administrátor, ktorý sa stará o správu užívateľov a správu príspevkov (odstraňovanie nemorálnych recenzií atď. ).

### 3. Riešenie

Projekt sme riešili v 4 fázach - vytvorenie ER a usecase diagramov, implementácia tabuliek, implementácia dotazov nad databázou, implementácia triggerov, procesov a pokročilých objektov databázy. Tieto fázy zodpovedajú požiadavkám jednotlivých termínov odovzdania. Pre prehľadnejšie priblíženie fungovania skriptu sme ho rozdelili na 7 častí, ktoré bližšie popisujeme v nasledujúcich podkapitolách.

#### 3.1 Zahodenie existujúcich dát

Ako prvý krok pri volaní skriptu sa zahodia všetky existujúce údaje o tabuľkách a štruktúrach. Tento krok je vykonaný pre umožnenie opakovaného volania.

#### 3.2 Vytvorenie tabuliek

V druhom kroku sa vytvoria tabuľky reprezentujúce jednotlivé entity v ER diagrame. Ak to návrh požaduje, sú tiež vytvorené tabuľky vzťahov jednotlivých entít a potrebných triggerov.

#### 3.3 Naplnenie dátami

Novo vytvorené tabuľky sú následne naplnené ukážkovými dátami. Každá tabuľka je naplnená zhruba štyrmi položkami. Jednotlivé dáta sú zvolené viac menej náhodne a ich vzťahy vybrané tak, aby čo najlepšie demonštrovali komplexnosť diagramu.

### 3.4 Dotazy nad databázou

Po naplnení tabuliek dátami môžeme demonštrovať niekoľko dotazov a vytvoriť procesy pracujúce s nimi. Vytvorili sme dva procesy – *kaviaren\_workers* a *ODSTRAN\_VSETKY\_KAVY\_OBASHUJUCE\_ZRNO\_AJ\_SO\_ZRNOM*. Proces *kaviaren\_workers* potrebuje 1 argument (meno kaviarne) a vracia počet zamestnancov pracujúcich v kaviarni (vrátane majiteľa a bez majiteľa). Proces *ODSTRAN\_VSETKY\_KAVY\_OBASHUJUCE\_ZRNO\_AJ\_SO\_ZRNOM* potrebuje 1 argument a to ID zrna kávy. Po jeho zavolaní sa vymažú z databázy všetky kávy obsahujúce dané zrno a aj samotné zrno. Taktiež sme vytvorili materializovaný pohľad *kava\_popularity*, ktorý obsahuje súhrn káv, a počet užívateľov ktorý danú kávu obľubujú. Ten sa dá využiť napríklad na určenie najobľúbenejších káv.

### 3.5 Nastavenie privilégií

V ďalšom kroku projektu sme prideliťovali práva. Tie boli dané druhému členovi tímu.

### 3.6 Pokročilé objekty schémy databázy

Nasledujúca tabuľka zobrazuje Explain plan vysvetľujúci postup databázy pri dotaze select, ktorý vyberá kávy s jemnou arómou a počíta priemer vypitých káv denne u užívateľov, ktorí ich obľubujú. Konkrétne sa jedná o nasledujúci príkaz:

```
EXPLAIN PLAN FOR SELECT u.ID_KAVA,  
    k.nazov,  
    AVG(u.pocet_kav_denne)  
FROM uzivatel u JOIN kava k ON u.ID_KAVA = k.ID_kavy WHERE aroma = 'Jemná'  
GROUP BY u.ID_KAVA, k.nazov;
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	23	7 (29)	00:00:01
1	HASH GROUP BY		1	23	7 (29)	00:00:01
2	MERGE JOIN		1	23	6 (17)	00:00:01
* 3	TABLE ACCESS BY INDEX ROWID	KAVA	1	17	2 (0)	00:00:01
4	INDEX FULL SCAN	SYS_C001908273	4		1 (0)	00:00:01
* 5	SORT JOIN		4	24	4 (25)	00:00:01
6	TABLE ACCESS FULL	UZIVATEL	4	24	3 (0)	00:00:01

Môžeme si všimnúť že databáza vykonáva až dve spojenia. Prvé na základe požiadavky aroma = 'Jemná' a druhé pri hľadaní správnych užívateľov. Tento postup sa dá urýchliť použitím indexu na tabuľke káva a jej stĺpci aroma. Zlepšenie môžeme vidieť na následnom Explain Plan:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	23	6 (17)	00:00:01
1	HASH GROUP BY		1	23	6 (17)	00:00:01
* 2	HASH JOIN		1	23	5 (0)	00:00:01
3	TABLE ACCESS BY INDEX ROWID BATCHED	KAVA	1	17	2 (0)	00:00:01
* 4	INDEX RANGE SCAN	AROMAINDEX	1		1 (0)	00:00:01
5	TABLE ACCESS FULL	UZIVATEL	4	24	3 (0)	00:00:01

Databáza využila novo vytvorený index. Vďaka nemu potrebovala už len jediný join na vykonanie dotazu. Podľa stĺpcu cena (Cost) si môžeme všimnúť, že dotaz s využitím indexu bol omnoho jednoduchší na vykonanie.

### 3.7 Transakcie

V poslednej časti sme sa rozhodli demonštrovať funkcionality transakcií. Transakcie splňujú všetky nasledujúce vlastnosti: atomicita, konzistentnosť, izolovanosť a trvalosť (ACID). Izolovanosť vieme pozorovať napríklad pri vracaní transakcie späť. Pri tomto úkone nie sú zasiahnuté iné transakcie a ak sú, tak sa všetky zasiahnuté transakcie vrátia späť. Trvalosť vieme vidieť na dátach, ktoré po úspešnom ukončení transakcie zostávajú zmenené. Konkrétne v našom skripte po úspešnej transakcii ukončenej COMMIT-om. Transakcie prechádzajú z jedného konzistentného stavu do druhého, čím sa demonštruje konzistentnosť. Atomicitu môžeme v transakcii sledovať pri vykonávaní jednotlivých príkazov. Všetky zmeny vykonané v rámci transakcie sú zapísané až po úspešnom realizovaní všetkých príkazov v danej transakcii. Ak ktorýkoľvek z príkazov zlyhá, zlyháva celá transakcia. Táto vlastnosť sa nám nepodarila demonštrovať v skripte, ktorý sme odovzdali. S úmyselnou chybou v transakcii sa skript odmietne vykonať.