

Dokumentace

Návrh use case diagramu a ER diagramu

IDS - Databázové systémy

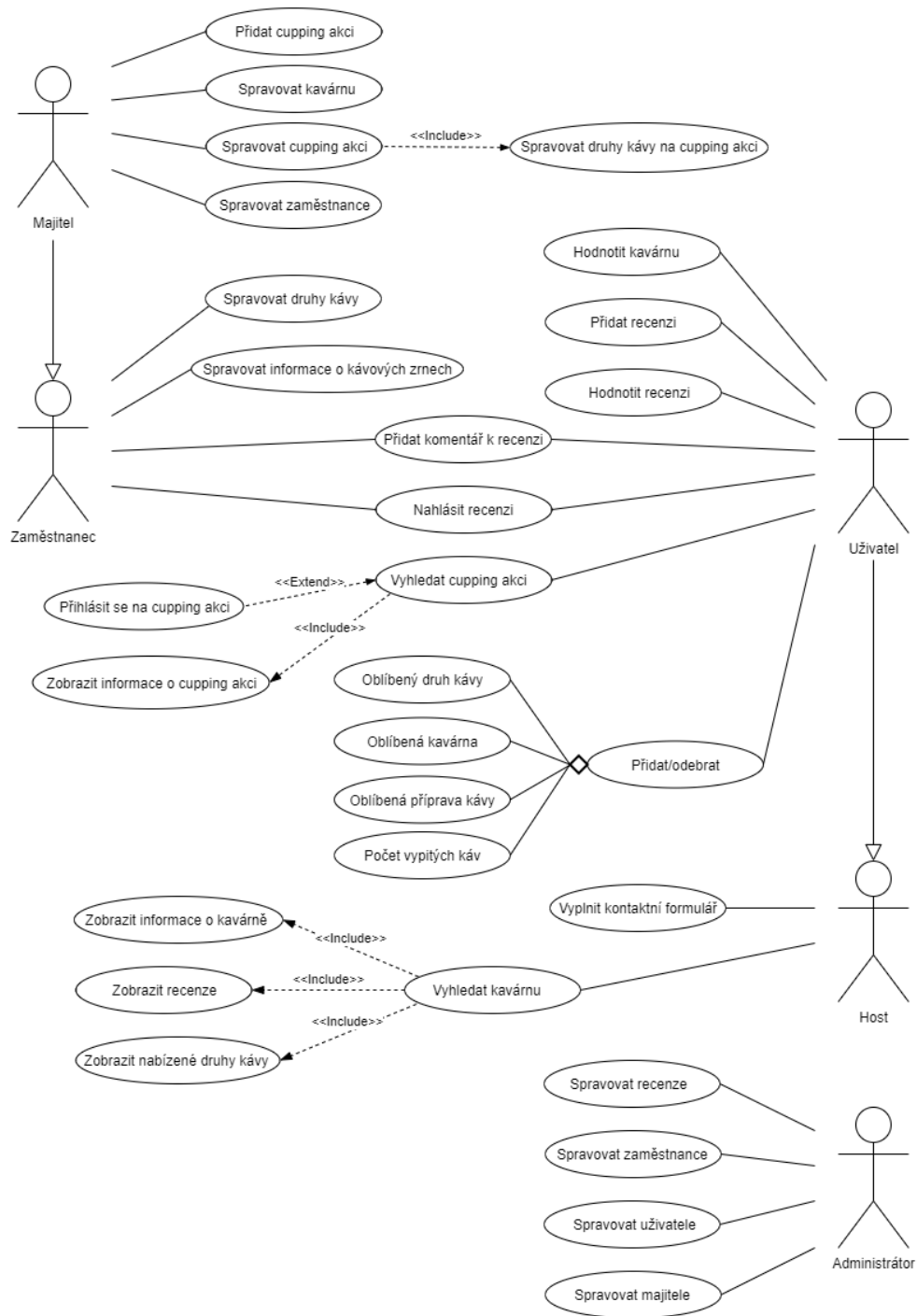
Obsah

1	Zadání - Kavárenský povaleč	2
2	Use case diagram	3
3	ER diagram	4
4	1. část - Diagramy	5
4.1	Use case diagram	5
4.2	ER diagram	5
5	2. část - Tvorba databáze	5
5.1	DROP TABLE	5
5.2	CREATE TABLE	5
5.3	INSERT INTO	5
5.4	UPDATE	5
6	3. část - Výtah dat z databáze	6
6.1	SELECT	6
7	4. část - Pokročilé objekty databáze	6
7.1	TRIGGER	6
7.2	EXPLAIN PLAN	6
7.3	PROCEDURE	6
7.4	GRANT	6
7.5	MATERIALIZED VIEW	6

1 Zadání - Kavárenský povaleč

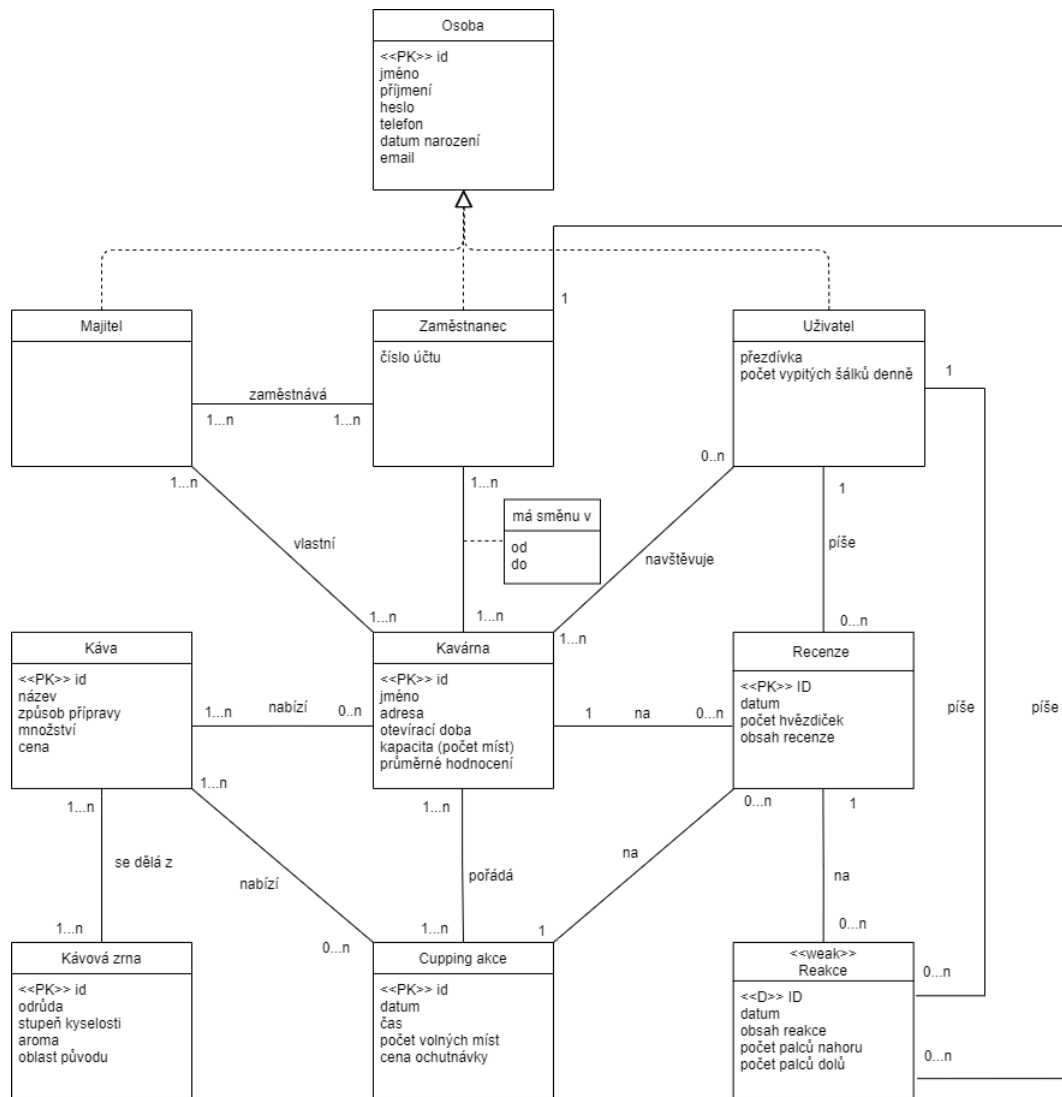
Představte si, že jste kavárenský povaleč, který tráví celé dny v kavárnách po celém Brně, a protože jste už vyzkoušeli spoustu kaváren a ne vždy vám v každé kavárně káva chutnala, rozhodli jste se vytvořit kavárenskou komunitu, ve které by si mohli lidé sdílet informace o brněnských kavárnách. Kavárny nabízejí kávy připravované z různých směsí kávových zrn, přičemž kávová zrna se liší odrudou, 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 otvírací 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 popis 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 Use case diagram



Obrázek 1: Use case diagram

3 ER diagram



Obrázek 2: ER diagram

4 1. část - Diagramy

4.1 Use case diagram

V use case diagramu jsme použili hned dvakrát zobecnění účastníka. Majitel podědí od zaměstnance všechny jeho relace k případům užití, to samé i u uživatele - ten podědí relace k případům užití od hosta. Hostovi či zaměstnanci nepatří ty relace, které náleží jen uživateli, respektive majiteli. K relaci zobrazit kavárnu náleží několik vazeb «include». Ty se spouští vždy, když je spuštěn případ, na které jsou napojeny. Mezi relacemi vyhledat cupping akci a přihlásit se na cupping akci náleží vazba «extend», jelikož přihlášení na cupping akci se nemusí provést vždy (může být až dalším krokem uživatele).

4.2 ER diagram

Entita **Recenze** je *weak* entitou, jelikož je přímo závislá na entitách **Uživatel**, **Kavárna** a **Cupping akce** (pokud by např. uživatel zanikl v systému, nemá smysl uchovávat recenzi daného uživatele). Entita **Reakce** je závislá na entitě **Recenze**, takže je také *weak* entitou. Dílčími klíči těchto entit (**Recenze**, **Reakce**) jsou jejich pořadí (od nejstaršího po nejnovější). Generalizaci jsme použili u entit **Majitel**, **Zaměstnanec** a **Návštěvník**. Tyto entity dědí společné vlastnosti entity **Osoba**. Vztah mezi **Zaměstnancem** a **Kavárnou** disponuje atributem "má směnu v", která obsahuje informace, od kdy do kdy směna probíhala.

Káva může být připravována z více druhů kávových zrn, proto je tam kardinalita $1...n$, ne 1 . Cupping akce nabízí alespoň jeden druh kávy (kardinalita $1...n$), ale jeden druh kávy pro ochutnání nemusí být zatím na nějaké cupping akci, ale pouze uchován v systému (kardinalita $0...n$). Cupping akce je pořádána kavárnou (nemusí tam být tedy vztah mezi entitami **Majitel** a **Cupping akce**).

5 2. část - Tvorba databáze

5.1 DROP TABLE

Při tvorbě databáze jsme jako první zrušili již existující tabulky, aby případně nedocházelo ke kolizím.

5.2 CREATE TABLE

Poté jsme vytvářeli jednotlivé tabulky podle ER diagramu, přidělovali jim primární (popřípadě cizí) klíče, dále jsme doplnili pomocné tabulky (tady jsme vycházeli z kardinalit), nadefinovali vlastnosti jednotlivých prvků databáze. Museli jsme si hlídat slabé entity, pomocné tabulky a generalizaci, kde se používají cizí klíče. Ošetřujeme také vlastnosti, aby např. nebyl zadán špatný formát bankovního účtu či rodného čísla.

5.3 INSERT INTO

Vytvořené tabulky jsme naplnili ukázkovými daty, s kterými pracujeme v dalších příkazech. Jsou zde již ošetřeny vstupy pomocí regexů v CREATE TABLE.

5.4 UPDATE

Pomocí příkazu UPDATE bereme z jedné (vlastnosti) tabulky konkrétní data, která jsou pak jedním souhrnným datem tabulky druhé. V našem případě jsme to použili pro průměrné hodnocení kavárny z recenzí.

6 3. část - Výtah dat z databáze

6.1 SELECT

Příkazem SELECT vybíráme data z databáze. Použili jsme hned několik variant příkazu:

- spojení dvou tabulek
- spojení tří tabulek
- klauzule GROUP BY
- EXISTS
- IN

7 4. část - Pokročilé objekty databáze

7.1 TRIGGER

Trigger definuje události nad tabulkou (hlídá je). Můžeme pomocí něj vkládat či mazat data. Jeden trigger jsme použili pro automatické generování hodnot primárního klíče tabulky Recenze. Druhý trigger kontroluje správnost otevírací doby kavárny.

7.2 EXPLAIN PLAN

Tento příkaz ukazuje detail daného příkazu. Náš EXPLAIN PLAN ukazuje, kolik kávových zrn z Etiopie je obsažen ve více jak v jedné kávě a kolika kávách se vyskytuje. Index nám pomáhá optimalizovat zpracování dotazů. Pomáhá nám v tom, že systém nemusí procházet všechna kávová zrna sekvenčně.

7.3 PROCEDURE

První procedura spočítá průměrnou cenu ochutnávek v rámci cupping akcí. Druhá procedura spočítá, kolik návštěvníků chodí do určité kavárny.

7.4 GRANT

Pomocí různých variant příkazu GRANT byla přidělena práva druhému členu týmu. Umožňuje mu pohled na databázi a práci s ní.

7.5 MATERIALIZED VIEW

Materializovaný pohled obsahuje souhrn káv a počet výskytů na cupping akcích. Můžeme vidět, která káva se vyskytuje na mnoha akcích, případně která se nevyskytuje (skoro) vůbec.