

IDS Projekt

Dokumentácia

Zadanie č.4 – Továreň na syry

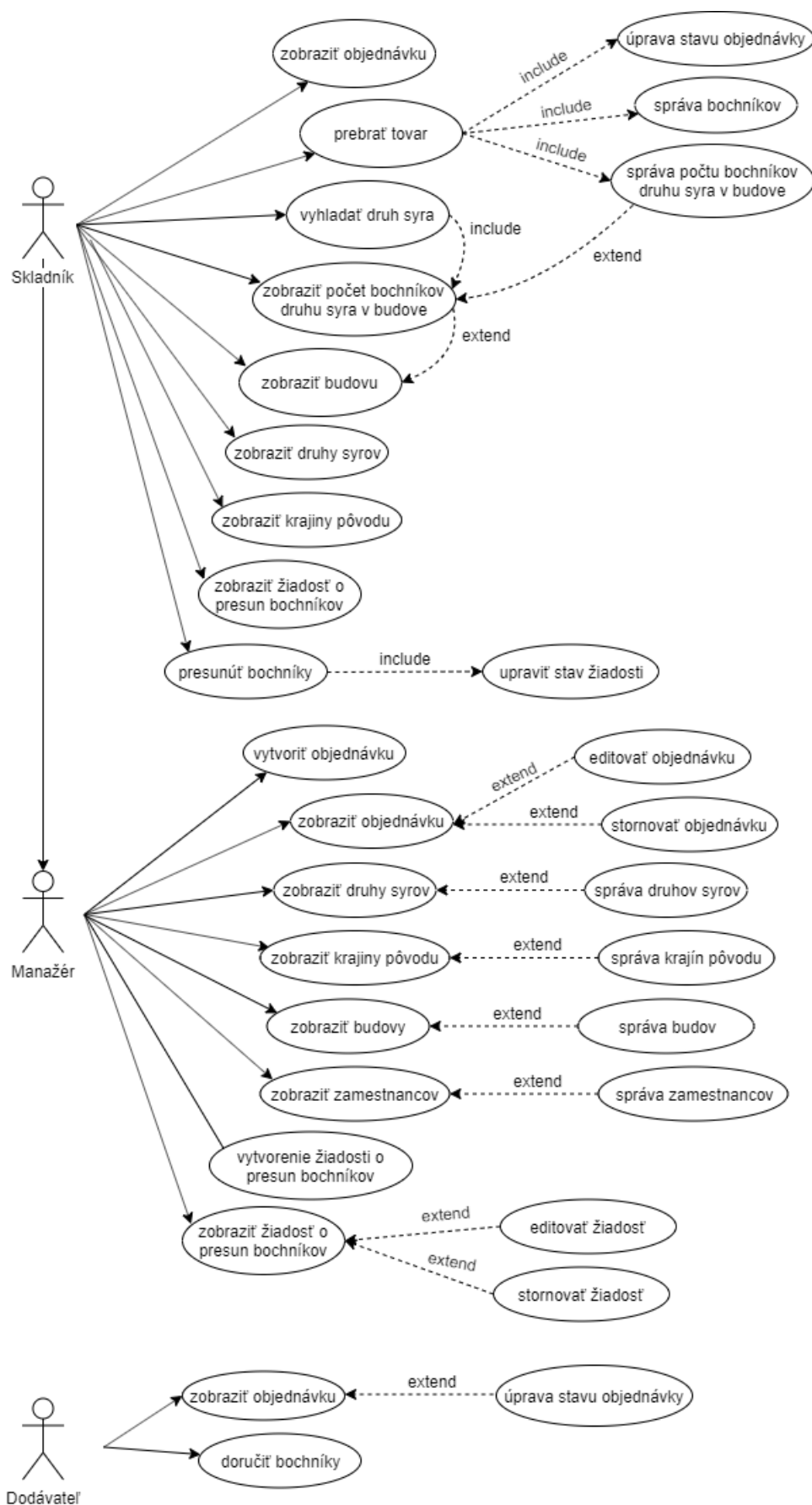
Jakub Duda, xdudaj02

Adam Kľučiar, xkluci01

Obsah

1.	Use case diagram.....	3
2.	ER diagram.....	4
3.	SQL skript.....	5
3.1.	Funkcia DROP_IF_EXISTS.....	5
3.2.	Vytvorenie tabuliek	5
3.3.	Triggery.....	5
3.4.	Procedúry	5
3.5.	Vloženie príkladných dát	5
3.6.	Demonštrácia funkcionality procedúr	5
3.7.	Výpis dát	6
3.8.	EXPLAIN PLAN bez/s využitím INDEXu.....	6
3.9.	Udelenie privilégií.....	6
3.10.	Materializovaný pohľad.....	6
3.11.	Demonštrácia funkcionality materializovaného pohľadu	6

1. Use case diagram



Obrázok č. 1

3. SQL skript

3.1. Funkcia DROP_IF_EXISTS

Funkcia `DROP_IF_EXISTS` je jednoduchá funkcia, ktorá prijíma jeden argument – meno tabuľky. Následne funkcia vyhľadá túto tabuľku a v prípade, že ju nájde, spustí príkaz `DROP` meno tabuľky. Táto funkcia je volaná na každú tabuľku pred jej vytvorením.

3.2. Vytvorenie tabuliek

Ďalej nasleduje vytvorenie príslušných tabuliek pre jednotlivé typy entít. Využitím príkazu `CONSTRAINT` sú definované primárne aj cudzie kľúče. Taktiež sú vytvorené tabuľky pomocou, ktorých sú reprezentované vzťahy medzi typmi entít typu nekonečno – nekonečno. Vzťah typu generalizácia je realizovaný pomocou atribútu `dobu_udenia`, ktorý je v prípade neúdeného typu syra defaultne 0.

3.3. Triggery

Trigger `Check_cislo_uctu` slúži na kontrolu správnosti atribútu `cislo_uctu` entity `zamestnanec`. Správnosť je kontrolovaná na základe výpočtu uvedeného na webovej stránke <https://www.kutac.cz/pocitace-a-internety/generovani-a-kontrola-ibanu-pro-cz-i-sk-ucty>.¹

Trigger `Generate_zamestnanec_id` slúži na generovanie hodnôt primárneho kľúča tabuľky `zamestnanec`. V prípade, že pri vkladaní do tejto tabuľky nie je primárny kľúč špecifikovaný, je tento nastavený na ďalšiu hodnotu z pomocnej sekvencie `id_sekvencia`.

3.4. Procedúry

Procedúra `CHEESE_TYPE_IN_ORDERS` vypíše počet objednávok, v ktorých sa vyskytuje vybraný druh syra, ktorého názov je predávaný ako parameter. Procedúra demonštruje využitie príkazu `CURSOR`, ktorý je konkrétne využitý na prechádzanie cez jednotlivé druhy syrov v objednávkach. V tejto funkcii je ošetrená výnimka `NO_DATA_FOUND`, ktorá môže byť aktivovaná v prípade zvolenia typu syru, ktorý sa nenachádza v databáze.

Procedúra `PERCENTAGE_OF_CHEESE_TYPE` je volaná s rovnakým parametrom ako predchádzajúca funkcia. Táto funkcia vypíše percentuálne zastúpenie bochníkov zvoleného druhu syra na sklade. Obsahuje taktiež ošetrenie výnimky `NO_DATA_FOUND` a výnimky `ZERO_DIVIDE`, ktorá môže byť aktivovaná v prípade, že sa na sklade nenachádzajú žiadne bochníky.

3.5. Vloženie príkladných dát

Následne je do databázy vložených niekoľko jednoduchých položiek.

3.6. Demonštrácia funkcionality procedúr

Na vstupných dátach je ukázaná funkcionality vyššie definovaných procedúr.

¹ Generování a kontrola IBANu pro CZ i SK účty, Kutáč.cz, 3.4.2021, <https://www.kutac.cz/pocitace-a-internety/generovani-a-kontrola-ibanu-pro-cz-i-sk-ucty>

3.7. Výpis dát

Následne je pomocou príkazu `SELECT` demonštrovaná extrakcia konkrétnych dát z databázy.

3.8. EXPLAIN PLAN bez/s využitím INDEXu

Pomocou príkazu `EXPLAIN PLAN` je demonštrovaná funkcionálnosť príkazu `INDEX`. Najprv je vytvorený `EXPLAIN PLAN` bez vytvorenia `INDEXu` a následne je vytvorený `INDEX` a spustený rovnaký `EXPLAIN PLAN`. `EXPLAIN PLAN` je využitý na analýzu príkazu zobrazenia počtu bochníkov jednotlivých druhov syra na sklade. `INDEX` je vytvorený na type entity `bochnik` a táto je zoradená podľa druhu syra, čím zjednoduší dané vyhľadávanie, pretože nebude potrebné prechádzať všetky bochníky (viď Obrázok č.3 a Obrázok č.4).

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT		49	2744	7 (15)
1	HASH GROUP BY		49	2744	7 (15)
* 2	HASH JOIN		49	2744	6 (0)
* 3	TABLE ACCESS FULL	DRUH_SYRA	2	86	3 (0)
4	TABLE ACCESS FULL	BOCHNIK	97	1261	3 (0)

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT		49	2744	4 (25)
1	HASH GROUP BY		49	2744	4 (25)
2	NESTED LOOPS		49	2744	3 (0)
* 3	TABLE ACCESS FULL	DRUH_SYRA	2	86	3 (0)
* 4	INDEX RANGE SCAN	BOCHNIK_PODLA_DRUHU	24	312	0 (0)

Obrázky č. 3 a č.4: Demonštrácia efektu využitia `INDEXu`

3.9. Udelenie privilégíí

Nasleduje udelenie privilégíí na manipuláciu s tabuľkami a procedúrami pre tímového kolegu.

3.10. Materializovaný pohľad

Je vytvorený materializovaný pohľad, ktorý obsahuje jednotlivé druhy syra a ich počty bochníkov na sklade. Z dôvodu neschopnosti udeliť práva na vytváranie materializovaného pohľadu druhému kolegovi kvôli nedostatočným právam prvého kolegu, vytvára tento pohľad prvý kolega.

3.11. Demonštrácia funkcionality materializovaného pohľadu

Pre demonštráciu funkcionality materializovaného pohľadu je vypísaný jeho obsah pred a po vložení nového bochníku do databázy. Materializovaný pohľad je potrebné v opakovaných intervaloch aktualizovať, pretože pri zmene v databázovej tabuľke, z ktorej bol vytvorený sa tento nemení. Z toho dôvodu sú obidva spomínané výpisy rovnaké.