

Vysoké učení technické v Brně

Fakulta informačních technologií

Databázové systémy 2020/2021

Projekt

Zadání č. 57 – **Pivovárníci**

Obsah

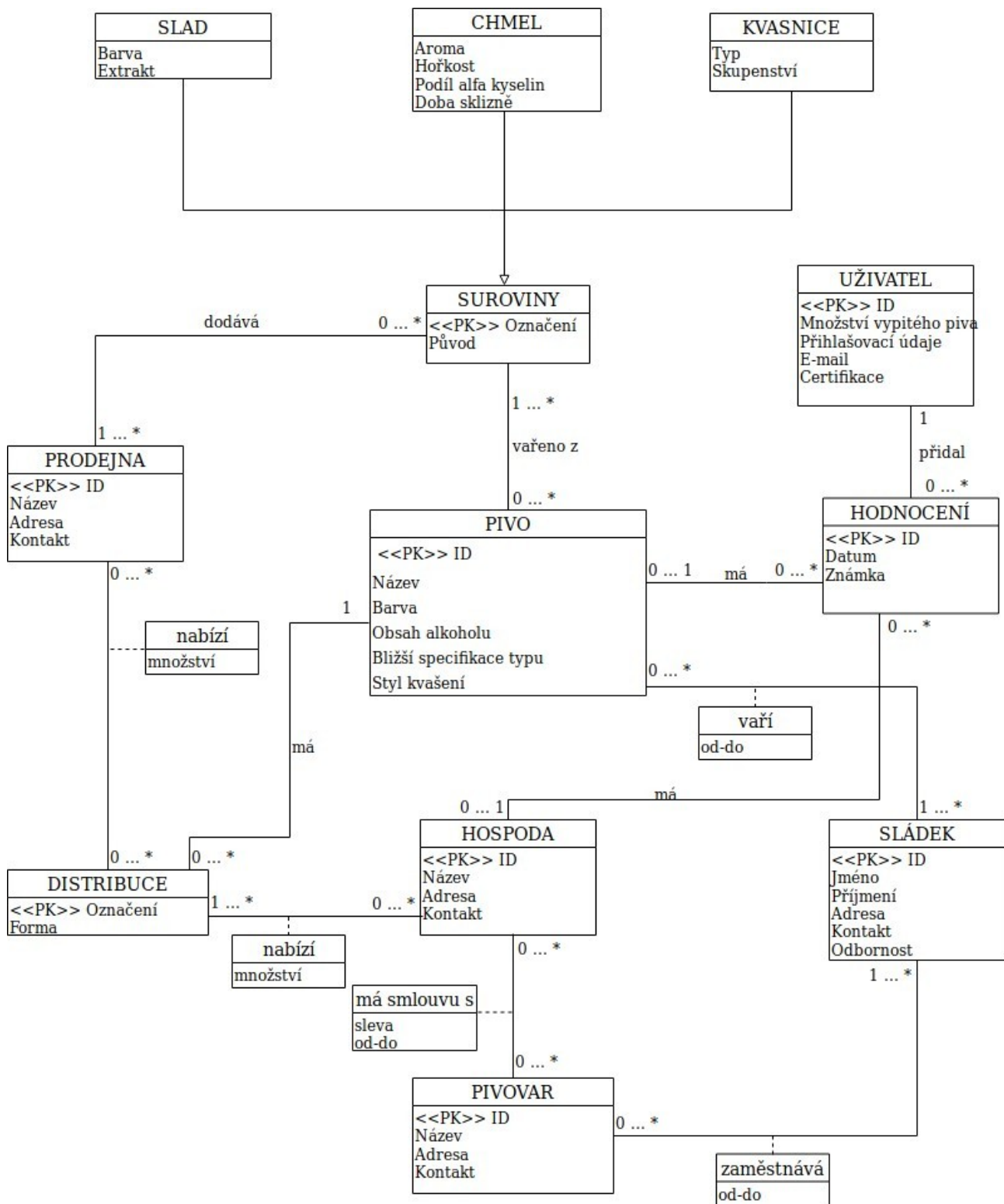
1. Zadání.....	3
2. Entity Relationship Diagram.....	4
3. Popis implementace.....	5
3. 1. Triggery.....	5
3. 2. Procedury.....	5
3. 3. EXPLAIN PLAN s využitím databázového indexu.....	6
3. 4. Přístupová práva.....	8
3. 5. Materializovaný pohled.....	8

1. Zadání

Pivovárníci

Vytvořte informační systém pro fanoušky dobrého piva. Každé pivo má název, barvu (tmavé, polotmavé, světlé; můžete uvažovat stupnici EBC), styl kvašení (svrchně, spodně,.), bližší specifikaci typu (ale, stout, ipa, apa, red ipa, .) a obsah alkoholu. Piva se primárně vaří z různých sladů (různé barvy, původu, extraktu, apod.), chmelů (o různých aromatech, hořkosti, podílu alfa kyselin, a různého původu a doby sklizně) a kvasnic o různých skupenstvích (navíc rozlišujeme mezi svrchními a spodními), přičemž všechny suroviny jsou dodávány pouze z certifikovaných kamenných prodejen. Uvařená piva, pak mohou být v několika formách distribucí, jako je například pet láhev, skleněná láhev, soudek, apod. a v těchto distribucích mohou být v různých množstvích dostupné jak v hospodách tak v kamenných prodejnách. Každé pivo je vařeno sládkem, přičemž stejné pivo může být vařeno v různých obdobích různými sládky (často hovoříme o generacích sládků). Někteří z těchto sládků však mohou být amatéři nebo sládci na volné noze, jiní jsou zaměstnání v rámci pivovarů, které rovněž svou značkou zaštiťují řadu z uvařených piv. Některé hospody navíc mají s některými pivovary rámcovou smlouvu v různých obdobích, kdy se zavazují dlouhodobě odebírat piva od pivovaru výměnou za možnou slevu v rámci procent. S informačním systémem můžou manipulovat i běžní uživatelé, hodnotit své oblíbené hospůdky, piva, vyhledávat podobné piva, případně zaznamenávat své úspěchy v podobě množství vypitého piva, sledovat své oblíbené pivovary a odebírat od nich novinky (např. když sládek uvaří nové pivo; pozn. novinky v databázi není třeba uvažovat). Někteří z těchto běžných uživatelů mohou být povýšení na certifikované sládky. Stává se tak dodáním ověřeného sládkovského diplomu potvrzeného notářem a umožňuje tak uživateli dodávat do systému své vlastní recepty a pokusy při vaření piva či amatérské distribuce bez záštity většími pivovary. Cidery, nealkoholické piva a parodie na piva (zpravidla uváděna s předponou "staro") neuvažujte.

2. Entity Relationship Diagram



3. Popis implementace

Tato dokumentace popisuje způsob implementace 4. části projektu do předmětu Databázové systémy - konkrétně vytvoření dvou databázových triggerů, dvou procedur, udělení přístupových práv uživateli, použití indexu, materializovaného pohledu a EXPLAIN PLAN.

3. 1. Triggery

Trigger s názvem `Kontrola_RC` slouží ke kontrole správnosti rodného čísla, postupně kontroluje, zda se nachází v rodném čísle pouze číslice, správnost délky rodného čísla – pokud je rodné číslo devítiznakové, zkontroluje se, zda nekončí třemi nulami. Pokud je desetiznakové, zkontroluje se, zda je dělitelné jedenácti. Dále je kontrolováno, zda je část rodného čísla, která obsahuje datum narození validní. Pokud se během kontroly zjistí, že rodné číslo není validní, tak se za pomoci funkce `Raise_Application_Error` vypíše chybové hlášení.

Trigger s názvem `autogener` automaticky generuje hodnoty primárního klíče tabulky `Pivo` ze sekvence `id_pivo_pocet`.

3. 2. Procedury

Procedura s názvem `Piva_v_hospode` zjistí názvy piv prodávaných v zadané hospodě. Procedura nejdříve zkontroluje, zda se zadaný název hospody vyskytuje v databázi. Poté prochází piva distribuovaná do hospod a pokud je aktuální pivo distribuováno do zadané hospody, je uloženo do proměnné `prodavana_piva`, která je nakonec vypsána uživateli. Pokud hospoda žádná piva neprodává, je to sděleno uživateli.

Procedura s názvem `Prumerne_hodnoceni_piva` zjistí průměrné hodnocení zadaného piva. Procedura nejdříve zkontroluje, zda se zadaný název piva vyskytuje v databázi. Poté zjišťuje, zda má zadané pivo alespoň jedno hodnocení – pokud ne, vypíše uživateli, že pivo nemá žádné hodnocení. Jinak vypočítá průměrné hodnocení piva pomocí funkce `AVG` a výsledek uloží do proměnné `prumer_hodnoceni` a tu poté vypíše uživateli.

3. 3. EXPLAIN PLAN s využitím databázového indexu

Příkaz `EXPLAIN PLAN` byl použit pro dotaz `SELECT`, který zjistí průměrné hodnocení pro pivo s názvem Pilsner Urquell.

Výsledek byl následující:

```
PLAN_TABLE_OUTPUT
```

```
Plan hash value: 2625215990
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		2	132	6 (0)	00:00:01
1	SORT GROUP BY NOSORT		2	132	6 (0)	00:00:01
* 2	HASH JOIN		2	132	6 (0)	00:00:01
* 3	TABLE ACCESS FULL	PIVO	1	40	3 (0)	00:00:01
4	TABLE ACCESS FULL	HODNOCENI	5	130	3 (0)	00:00:01

```
PLAN_TABLE_OUTPUT
```

```
Predicate Information (identified by operation id):
```

```
2 - access("PIVO"."id"="HODNOCENI"."id_pivo")
3 - filter("PIVO"."NAZEV"='Pilsner Urquell')
```

```
Note
```

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

Ve výpisu lze vidět, že je třeba přistoupit ke všem datům sloupce `Pivo.nazev`.

Pro optimalizaci dotazu byl vytvořen index s názvem `index_piva`, který indexuje sloupec `nazev` z tabulky `Pivo`. Příkaz `EXPLAIN PLAN` byl následně zavolán znovu s využitím tohoto indexu.

Výsledek byl následující:

PLAN_TABLE_OUTPUT

Plan hash value: 1481927610

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		2	132	5 (0)	00:00:01
1	SORT GROUP BY NOSORT		2	132	5 (0)	00:00:01
* 2	HASH JOIN		2	132	5 (0)	00:00:01
3	TABLE ACCESS BY INDEX ROWID	PIVO	1	40	2 (0)	00:00:01
* 4	INDEX RANGE SCAN	INDEX_PIVA	1		1 (0)	00:00:01
5	TABLE ACCESS FULL	HODNOCENI	5	130	3 (0)	00:00:01

PLAN_TABLE_OUTPUT

Predicate Information (identified by operation id):

- 2 - access("PIVO"."id"="HODNOCENI"."id_pivo")
- 4 - access("PIVO"."NAZEV"='Pilsner Urquell')

Note

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

Z uvedeného výsledku lze vidět, že místo přístupu ke všem datům sloupce `Pivo.nazev` a jejich následnému porovnávání s hledaným názvem `piva` bylo provedeno skenování indexovaných dat. Ze souhrnu je zřejmé, že došlo k mírnému snížení ceny jednotlivých operací, ačkoli k výraznější úspoře výkonu a času by došlo, kdyby databáze obsahovala daleko rozsáhlejší množství dat.

3. 4. Přístupová práva

Přístupová práva byla nastavena uživateli `xcibul12`. Byla udělena všechna privilegia k tabulkám použitím příkazu `GRANT ALL`. Práva k volání procedur byla udělena použitím příkazu `GRANT EXECUTE`.

3. 5. Materializovaný pohled

Materializovaný pohled pro člena týmu byl vytvořen pro uživatele `xcibul12` a je demonstrován na dotazu `SELECT`, který zjistí všechny názvy piv, jež mají distribuci. Daný člen týmu použije materializovaný pohled, poté vloží novou distribuci piva. Následně se znovu podívá, zda je distribuce zvolena a zjistí že není, protože před změnou je zapotřebí použít příkaz `COMMIT`. Po jeho použití je změna v tabulce vidět.