



IDS - projekt

Relačná databáza

2. mája 2022

Diana Maxima Držíková (xdrzik01)
Mária Nováková (xnovak2w)

Obsah

1	Úvod	2
1.1	Zadanie	2
2	1. časť	3
2.1	Cieľ	3
2.2	Dátový model (ER diagram)	3
2.2.1	Vzťahy entít	3
2.3	Model prípadov použitia (Use-case diagram)	3
2.3.1	Vysvetlenie funkcionality IS	3
3	2. časť	5
3.1	Cieľ	5
3.2	Stručný popis skriptu	5
3.3	Dátový model (vygenerovaný zo skriptu)	6
4	3. časť	7
5	4. - 5. časť	8
5.1	Implementácia	8
5.1.1	TRIGGERS	8
5.1.2	PROCEDURES	8
5.1.3	EXPLAIN PLAN	8
5.1.4	Privilégia	9
5.1.5	MATERIALIZED VIEW	9

1 Úvod

Cieľom riešeného projektu je návrh a implementácia relačnej databázy na zvolenú tému. Projekt pozostáva z piatich častí.

1.1 Zadanie

Pivovarníci (IUS2020/21 - verzia zadania 57.)

Vytvorte informačný systém pre fanúšikov dobrého piva. Každé pivo má názov, farbu (tmavé, polotmavé, svetlé; môžete uvažovať stupnicu EBC), štýl kvasenia (vrchne, spodne,.), bližšiu špecifikáciu typu (ale, stout, ipa, apa, red ipa, .) a obsah alkoholu. Pivá sa primárne varia z rôznych sladov (rôzne farby, pôvodu, extraktu, apod.), chmeľov (o rôznych arómach, horkosti, podielu alfa kyselín, a rôzneho pôvodu a doby zberu) a kvasníc o rôznych skupenstvách (navyše rozlišujeme medzi vrchnými a spodnými), pričom všetky suroviny sú dodávané iba z certifikovaných kamenných predajní. Uvarené pivá, potom môžu byť v niekoľkých formách distribúcií, ako je napríklad pet fľaša, sklenená fľaša, súdok, a pod. a v týchto distribúciách môžu byť v rôznych množstvách dostupné, ako v krčmách, tak v kamenných predajniach. Každé pivo je varené sládkom, pričom rovnaké pivo môže byť varené v rôznych obdobiach rôznymi sládkami (často hovoríme o generáciách sládkov). Niektorí z týchto sládkov však môžu byť amatéri alebo sládci na voľnej nohe, iní sú zamestnaní v rámci pivovarov, ktoré tiež svojou značkou zaštitujú rad z uvarených pív. Niektoré krčmy navyše majú s niektorými pivovarmi rámcovú zmluvu v rôznych obdobiach, kedy sa zaväzujú dlhodobo odoberať pivá od pivovaru výmenou za možnú zľavu v rámci percent. S informačným systémom môžu manipulovať aj bežní užívatelia, hodnotiť svoje obľúbené krčmičky, pivá, vyhľadávať podobné pivá, prípadne zaznamenávať svoje úspechy v podobe množstva vypitého piva, sledovať svoje obľúbené pivovary a odoberať od nich novinky (napr. keď sládok uvarí nové pivo; pozn. novinky v databáze nie je potrebné uvažovať). Niektorí z týchto bežných užívateľov môžu byť povýšení na certifikované sládky. Stáva sa tak dodaním overeného sládkovského diplomu potvrdeného notárom a umožňuje tak užívateľovi dodávať do systému svoje vlastné recepty a pokusy pri varení piva či amatérskej distribúcie bez záštitvy väčšími pivovarmi. Cidery, nealkoholické pivá a paródie na pivá (spravidla uvádzané s predponou "staro") neuvažujte.

2 1. časť

2.1 Cieľ

Zostrojiť dátový model (ER diagram) zachytávajúci štruktúru dát, vyjadrený ako diagram tried v notácii UML alebo ako ER diagram a model prípadov použitia vyjadrený ako diagram prípadov použitia v notácii UML reprezentujúci požiadavky na poskytovanú funkcionálnu aplikáciu používajúcu databázu navrhnutého dátového modelu. Dátový model musí obsahovať aspoň jeden vzťah generalizácie/specializácie.

2.2 Dátový model (ER diagram)

2.2.1 Vzťahy entít

Ako prvé sme vytvárali vzťahy medzi entitou **PIVO** a **SUROVINA**. Pivo sa skladá z troch ingrediencií - **CHMEL'**, **SLAD**, **KVASNICE**. Každá z nich tvorí osobitnú entitu, ktorá má špecifické vlastnosti. Tieto tri entity sú generalizované entitou **SUROVINA** z ktorej vedie vzťah do entity **PIVO** a popisuje, že z viacerých surovín (vždy tri) sa skladá pivo. Rovnako platí, že nejaké suroviny s konkrétnymi atribútami môžu byť obsiahnuté aj vo viacerých pivách.

Entita **PIVO** je ďalej spojená s entitou **UŽÍVATEL'** pričom platí, že užívateľ (ak má sládkovský diplom) môže variť viacero druhov piva a pivo môže byť varené viacerými užívateľmi/sládkami.

UŽÍVATEL' má vzťah so samým sebou, kedy do systému môže aj nemusí nahráť sládkovský diplom a tým sa povýšiť na certifikovaného sládku.

Medzi entitami **UŽÍVATEL'** a **PIVOVAR** sú dva vzťahy. Prvý vzťah hovorí o tom, že užívateľ, ktorý bol povýšený na sládku, môže pracovať pod nejakým pivovarom a pivovar môže mať viacerých sládkov. Druhý vzťah predstavuje sledovanie noviniek obľúbených pivovarov. Užívateľ nemusí odoberať novinky od žiadneho alebo viacero pivovarov a pivovar nemusí mať žiadnych alebo viacerých sledovateľov.

Entita **PODNIK** zastrešuje hospody a predajne, do ktorých sa dováža pivo. **PODNIK** potrebuje disponovať aspoň jednou distribúciou piva v rôznej forme a množstve, lebo by inak nemalo zmysel ho viesť v databáze (ak by podnik neponúkal pivo).

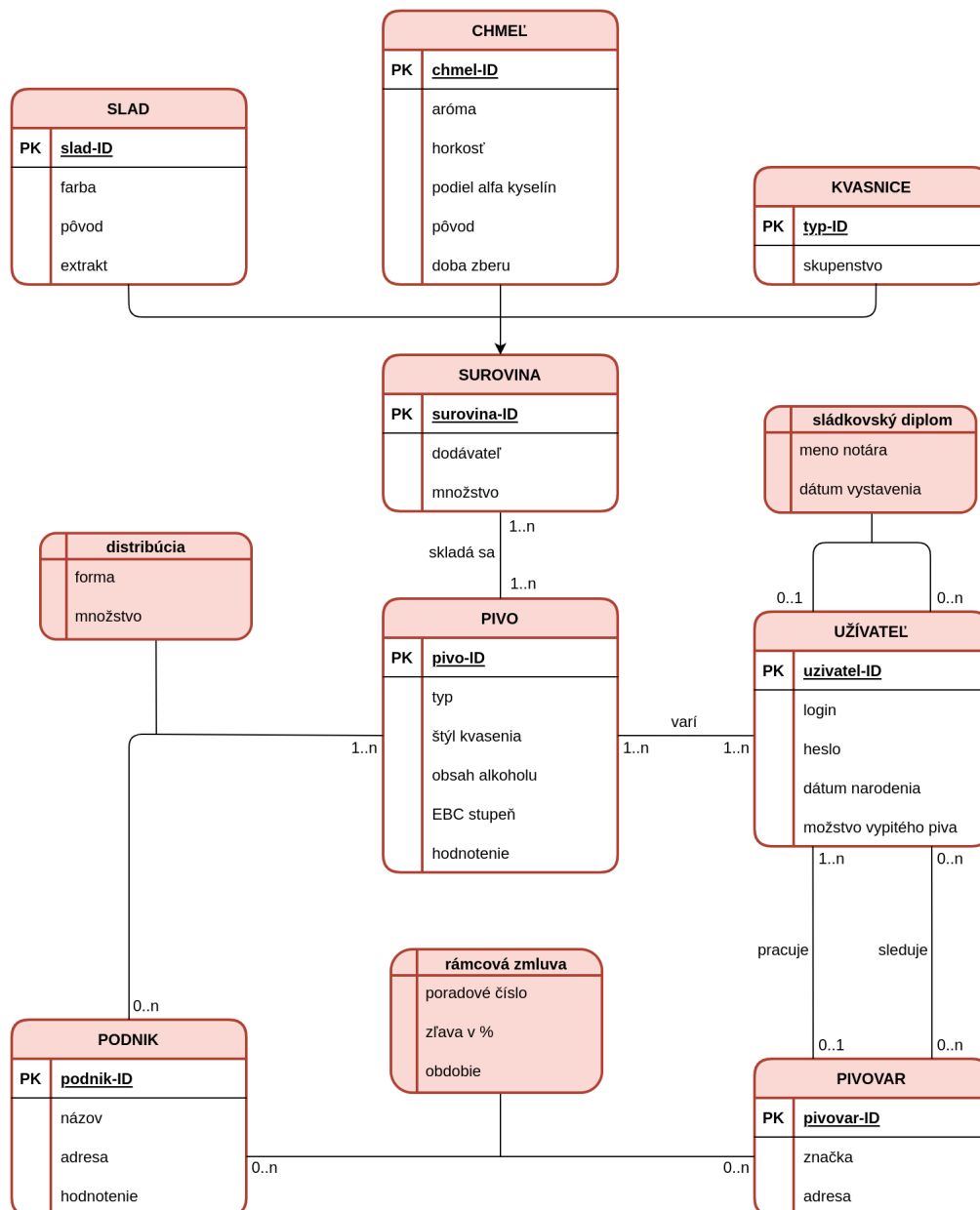
PODNIK môže, ale aj nemusí naviazať spoluprácu s pivovarom, lebo môže prijímať produkty iba od sládku na voľnej nohe. Pri entite **PIVOVAR** predpokladáme, že môžu existovať pivovary, ktoré svoje produkty nechcú distribuovať do rôznych podnikov a chcú ich predávať len u seba v pivovare.

2.3 Model prípadov použitia (Use-case diagram)

2.3.1 Vysvetlenie funkcionality IS

Informačný systém môže byť používaný aj **anonymom**, ale iba v obmedzenom režime. Anonym môže vyhľadávať značky pív a ich distribútorov. Ak si anonym zaregistruje účet, dostáva možnosť pracovať so systémom ako bežný užívateľ.

Bežný užívateľ okrem vyhľadávania značiek môže pridávať hodnotenia pivám alebo podnikom a rovnako môže odoberať novinky od distribútorov podľa jeho voľby. Okrem toho môže zaznamenať

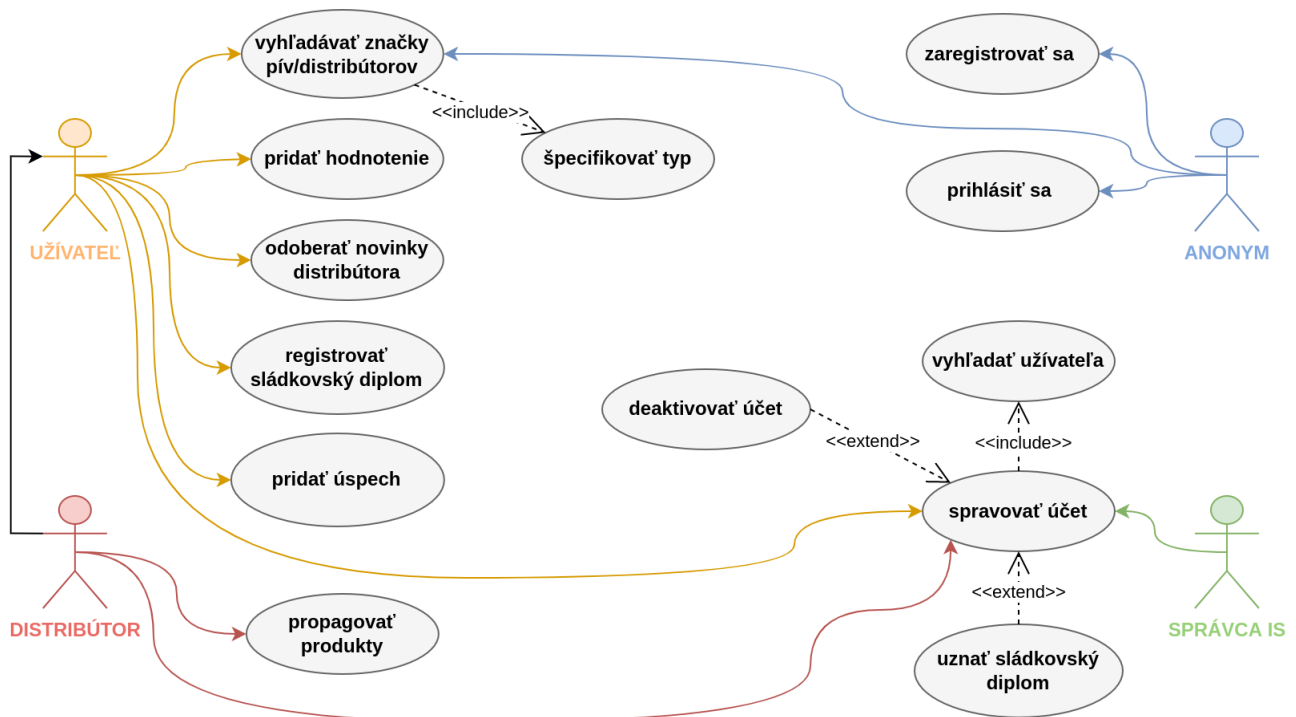


Obr. 1: ERD

svoje úspechy, v podobe množstva vypitého piva. Užívateľ môže získať povýšenie pridaním sládkovského diplomu a tým sa z neho môže stať distribútor.

Distribútor má možnosť vykonávať všetky veci rovnako, ako bežný užívateľ a navyše môže propagovať svoje produkty (napr. môže pridávať svoje recepty).

Správca IS spravuje účty. Môže užívateľom deaktivovať účet. Správca IS zároveň potvrdzuje sládkovské diplomy a tým udeľuje rolu distribútora.



Obr. 2: UCD

3 2. časť

3.1 Cieľ

Vytvorenie SQL skriptu, ktorý generuje základné objekty schémy databázy (tabuľky, definície integritných obmedzení) a následne naplnenie tabuliek ukázkovými dátami.

3.2 Stručný popis skriptu

Skript sme vytvárali v prostredí DataGrip. Vytvorili sme tabuľky pomocou príkazu `create table` pre jednotlivé entity z ER diagramu. Zaviedli sme generovanie primárneho kľúču pomocou príkazu `INT GENERATED BY DEFAULT AS IDENTITY NOT NULL PRIMARY KEY`, ktorý umožňuje ako automatické generovanie primárneho kľúča, tak aj manuálne nastavenie primárneho kľúča. Do tabuliek sme prepísali atribúty jednotlivých entít a určili ich dátové typy.

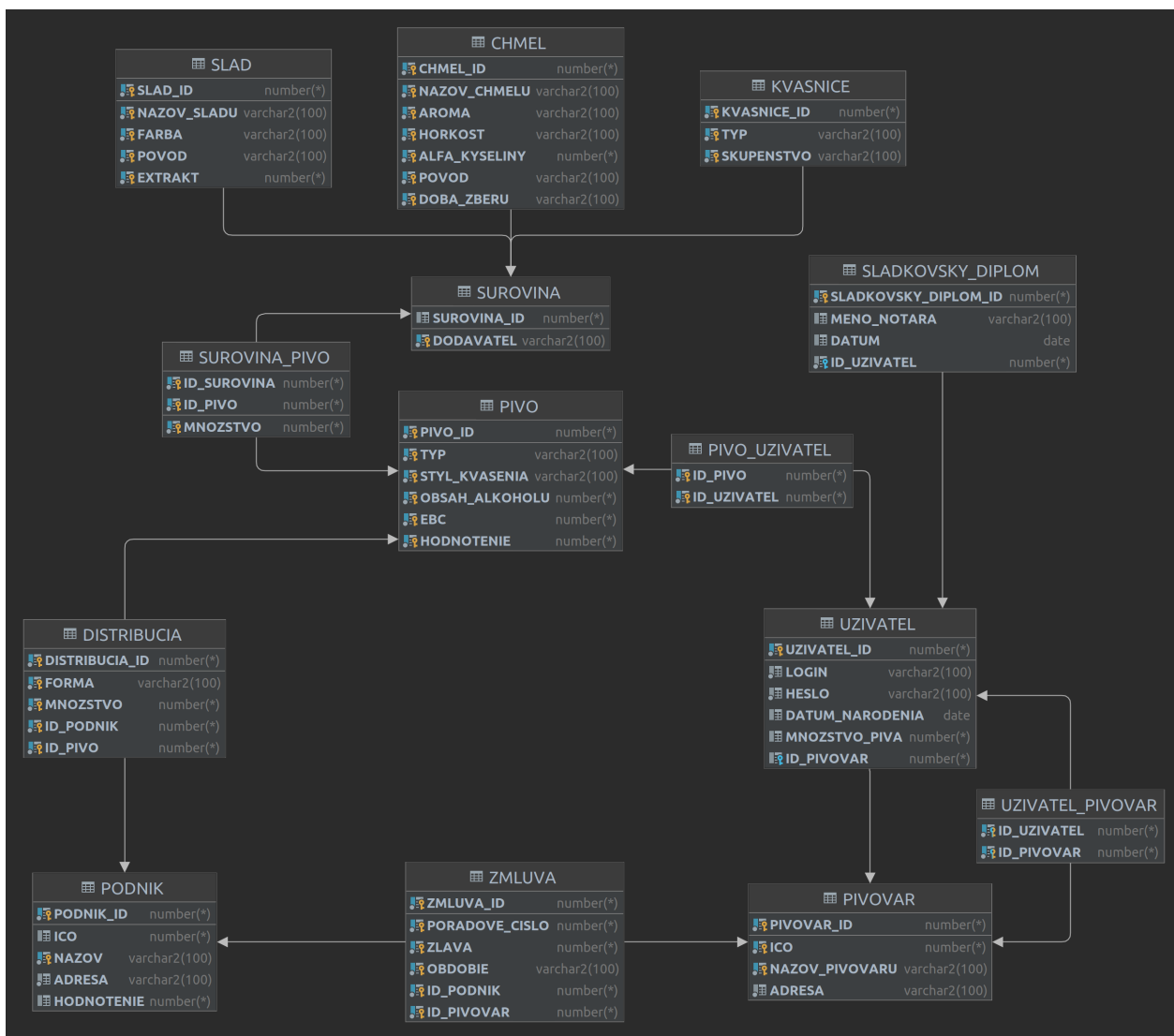
Následne sme začali pridávať vzťahy medzi jednotlivými entitami. V prípade vzťahu $1..n$ (alebo $0-1..n$) sme pridávali do tabuliek, ktoré referovali na práve jednu inštanciu druhej entity, cudzie kľúče prvkov. Pre vzťah $n..n$ sme vytvorili nové tabuľky. Celkovo nových tabuliek bolo päť z toho tri niesli dodatočné informácie ako atribúty vzťahu.

Generalizáciu sme spravili pomocou metódy, ktorá vyžaduje vytvorenie jednej tabuľky pre nadtyp a tabuliek pre podtypy s primárnym kľúčom ich nadtypu. To znamená, že tabuľky **SLAD**, **CHMEL** a **KVASNICE** nemali vlastný, automaticky generovaný primárny kľúč, ale ich primárny kľúč bol explicite definovaný z primárneho kľúča tabuľky **SUROVINA**. L'ubovoľne sme si určili, ktorý primárny kľúč prvku v tabuľke **SUROVINA** bude určovať konkrétnu surovinu.

Po overení správnosti asociácií medzi tabuľkami sme dané tabuľky naplnili pomocou príkazu `INSERT INTO` a `VALUES` ukázkovými hodnotami. Pre aktualizáciu tabuľiek pri každom novom spusení sme využili funkcie `drop table`.

Pre integritné obmedzenie sme zaviedli nový atribút `ICO` v tabuľkách **PODNIK** a **PIVOVAR**.

3.3 Dátový model (vygenerovaný zo skriptu)



Obr. 3: Dátový model

4 3. časť

V tretej časti sme pridávali do sql súboru selecty, ktoré špecifikovali aké údaje z daných tabuliek chceme vypísať.

Vytvorili sme dokopy osem selectov v ktorých používame zlučovanie tabuliek, agregácie funkcie s kazuľou GROUP BY, vylučovanie riadkov z tabuľky pomocou NOT EXISTS a prehľadávanie tabuliek pomocou IN a zanorenými selectami.

```
-- Ktoré zmluvy boli vystavené v roku 2020 na Slovensku? (zmluva_id, zlava, id_podnik, id_pivovar)
SELECT zmluva_id, zlava, id_podnik, id_pivovar
FROM zmluva
WHERE id_podnik IN
(SELECT podnik_id FROM podnik
WHERE id_pivovar IN
(SELECT pivovar_id FROM pivovar
WHERE pivovar.adresa LIKE '%SK%' AND zmluva.obdobie = '2020'));
```

Obr. 4: Dátový model

	ZMLUVA_ID	ZLAVA	ID_PODNIK	ID_PIVOVAR
1	4	20	4	3
2	5	15	4	4

Obr. 5: Dátový model

5 4. - 5. časť

V predposlednej časti sme vytvárali pokročilejšie objekty schémy typu: TRIGGER, PROCEDURE, EXPLAIN PLAN a MATERIALIZED VIEW.

5.1 Implementácia

Použili sme vytvorený skript z pôvodných zadání na aplikovanie rozšírení. Vytvorené tabuľky sme naplnili a pracovali ďalej s dátami.

5.1.1 TRIGGERS

Prvý z nich (kontrola_veku) vytvára vekové obmedzenie pre užívateľov pred registráciou do systému. Použili sme vstupnú premennú SYSDATE a vyrátali počet mesiacov medzi momentálnym dátumom a dátumom narodenia užívateľa, ktorého práve pridávame do systému. Tento mesačný rozdiel sme podelili dvanástimi aby sme dostali vek užívateľa. Ak je užívateľ mladší, tak systém vyhlási error.

Druhý (nazov_piva) sme vytvorili pre skúšku s aktualizáciou názvu piva. Ak máme v databáze pivo, ktorého názov chceme aktualizovať, tak ho aktualizujeme aj v tabuľke pivo_uzivatel.

Posledný (na_sklade) kontroluje či pri vytváraní distribúcie pív do podnikov neprekračujeme množstvo piva na sklade. Vždy po vytvorení distribúcie odpočíta zo skladu počet položiek, ktoré boli distribuované. Ak sa vytvára distribúcia, ktorá chce dodať viac piva, ako je na sklade, skript vyhlási chybu.

5.1.2 PROCEDURES

Procedúra sladkoviaVSnesladkovia vypočíta pomer sládkov a nesládkov v databáze. Zrátame počet užívateľov, ktorých id_pivovar je NULL (nesládkovia) a počet užívateľov, ktorých id_pivovar nie je NULL a dáme to do pomeru s počtom všetkých užívateľov. Desatinné čísla zaokrúhlujeme na dve miesta.

Druhá procedura pocet_predaja slúži na to, aby sme zistili koľkokrát boli zakúpené suroviny od zadaného dodávateľa pre varenie pív, ktoré vedíme v databáze. Používame premennú, ktorá je rovnakého dátového typu ako id_surovina z tabuľky surovina_pivo, vytvorenú pomocou surovina_pivo.id_surovina%TYPE. Používame kurzor, ktorý zastrešuje primárne kľúče suroviny (v tomto kontexte dodávateľa) z tabuľky surovina_pivo. Podľa názvu dodávateľa nájdeme v tabuľke surovina jeho primárny kľúč. V cykle LOOP prechádzame všetky primárne kľúče, ktoré sme získali pomocou kurzoru a porovnávame ich s chceným primárnym kľúčom. Počet vyskytov zaznamenávame v premennej pouzity_pocet. Na konci vypíšeme koľko surovín na pivo bolo zakúpených u konkrétného dodávateľa.

5.1.3 EXPLAIN PLAN

EXPLAIN PLAN zobrazuje, koľko odoberateľov majú dané pivovary. Zlúčili sme dve tabuľky uzivatel_pivovar a pivovar, porovnali primárne kľúče a pomocou agregáčnej funkcie COUNT a kauzality GROUP BY sme zráтали pomocou nazov_pivovaru.

Pridali sme index `pivovary_index` na `nazov_pivovaru` pre optimalizáciu. Vytvorenie tohto indexu urýchlí uje vyhľadávanie, lebo systém nemusí prechádzať všetky názvy pivovarov sekvencne.

5.1.4 Privilégia

Pridali sme privilégia pre prístup k tabuľkám pomocou `GRANT ALL` a k procedúram pomocou `GRANT EXECUTE`.

5.1.5 MATERIALIZED VIEW

Pohľad obsahuje užívateľov, ktorí vypili viac, ako dvadsať litrov do teraz. Vyskúšali sme aktualizáciu dát v tabuľke `uzivatel` a pridali sme jednému fanúšikovi piva o sto litrov piva viac. Tabuľka sa aktualizovala no pohľad nie, takže sme použili `COMMIT` a tým sme pridali zmeny aj do pohľadu.