

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Databázové systémy (IDS)

Dokumentace k finálnímu řešení databáze

Téma – Klub anonymních alkoholiků

2018/2019

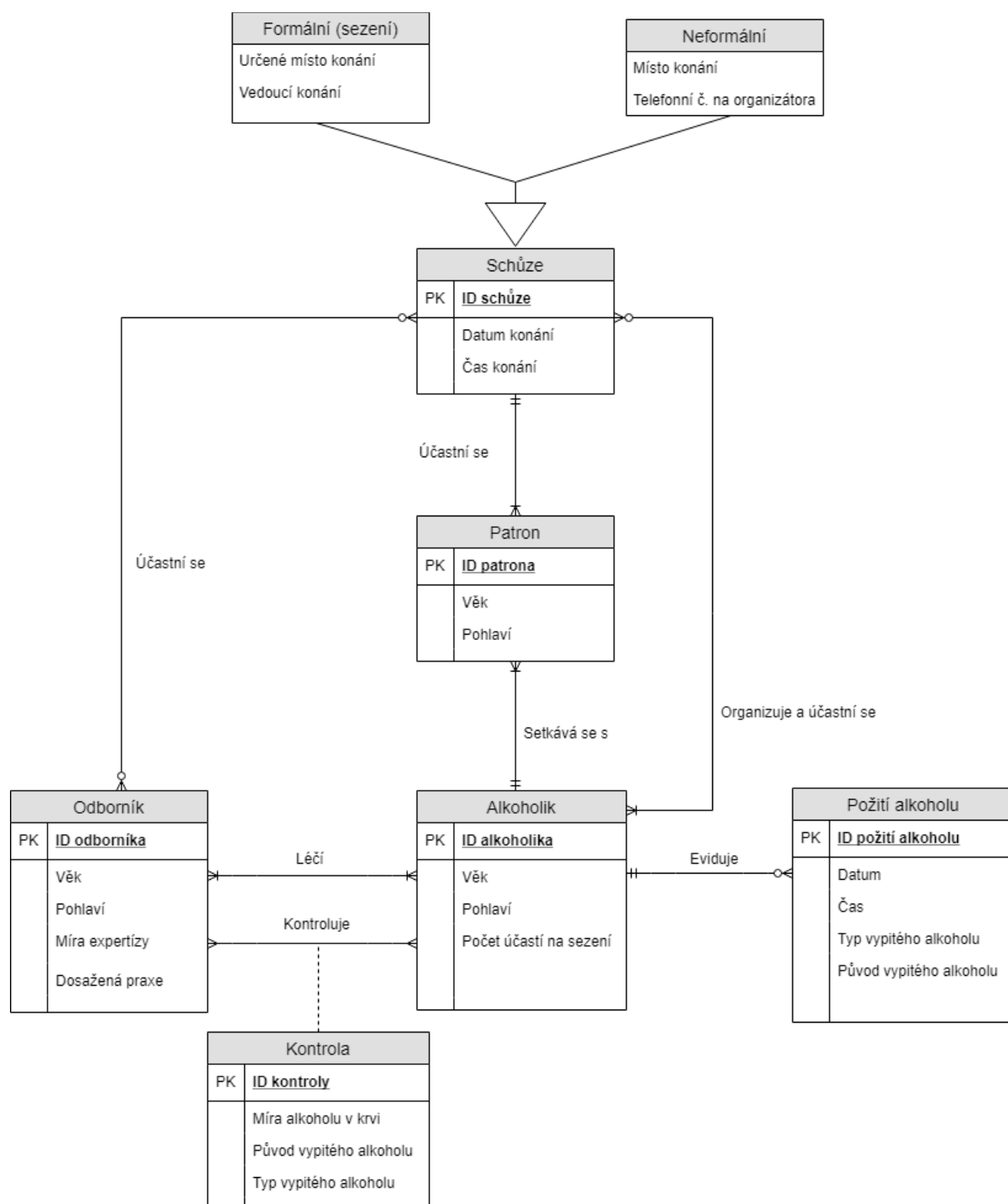
Kateřina Fořtová (xforto00)

Jaromír Homolka (xhomol21)

1 Zadání tématu

Navrhněte informační systém, který bude podporovat anonymní alkoholiky k organizaci sezení a evidenci vypitého alkoholu. Systém uchovává základní informace o alkoholících, jako je jejich věk, pohlaví, patrony, kteří je podporují a se kterými se nepravidelně scházejí na různých místech a v různých datech a rovněž i informace o odbornících, kteří na ně lékařsky dohlíží. Odborníci musí mít patřičnou expertízu pro pečování o alkoholiky, a mít minimální lékařskou praxi, která je v systému evidována. Patronem však může být kdokoliv. Pravidelně se konají sezení, kterých se účastní až dvanáct alkoholiků a navíc můžou být přítomni jak patroni tak i odborníci a dohlížet nad diskuzí. U každého sezení nás zajímá datum, čas, a místo konání. Těchto míst je pouze několik oficiálních a dedikovaných. Každé sezení je vedeno jednou osobou. Neformální schůzky s patrony však mohou být organizovány v libovolné lokalitě. Alkoholici se musí alespoň třikrát ročně účastnit nějakého sezení, a v případě, že se více jak tři měsíce nedostaví na žádné sezení je jim systémem zaslána upomínka. U alkoholiků jsou pravidelně (i nepravidelně a nečekaně) prováděny kontroly odborníky, na kterých se měří míra alkoholu v jejich krvi. Tato míra vypitého alkoholu je pak evidována do systému, rovněž s původem a typem vypitého alkoholu. Alkoholici však mohou sami zaevidovat (ze špatného svědomí), že alkohol požili (tedy mimo prováděné kontroly) a tuto informaci rovněž přidat do systému.

2 Schéma databáze



3 Popis implementace databáze

3.1 Vytvoření základních objektů databáze

Na samotném začátku tvorby skriptu v jazyce SQL bylo potřeba vytvořit tabulky v databázi – vytvořili jsme tedy tabulky `Alkoholik`, `Patron`, `Odbornik`, `Poziti_alkoholu` a tabulku `Schuze`, ke které jsme následně přidali dvě tabulky `Formalni` a `Neformalni`, které jsou vztahem generalizace/specializace a vztahují se k tabulce `Schuze`, obsahují stejný primární klíč.

3.2 Tvorba triggerů a práce s CHECK

Klauzule `CHECK` byla použita při ověření formátu správného telefonního čísla u organizátora neformální schůze. Kontrolujeme ho jednoduchým regexem ve formátu `'%[^0-9]%'`.

Triggerů bylo vytvořeno několik. Jelikož ze zadání vyplývá, že formální schůze se koná pouze na nějakých předem určených místech, testujeme, zda zadané místo u formální schůze odpovídá některému z pole, přitom kontrolujeme bez ohledu na velikost znaků. Další triggerů kontrolují platné pohlaví u alkoholika, patrona a odborníka.

U odborníka také testujeme míru expertízy spadající do daných kategorií – základní, střední, vysoká a excelentní.

Jsou obsaženy i povinné triggerů pro generování indexů primárních klíčů, pokud je nezadán určený primární klíč. Dále kontrolujeme správné zadání jména a příjmení u vedoucího formální schůzky regexem ve tvaru `'^[a-zA-Z]*$'`.

Dále jsme definovali pole všech určených typů alkoholu, které akceptujeme při zadání kontroly a požití alkoholu. Jsou to např. bílé víno, červené víno, medovina atd.

3.3 Tvorba procedur

První procedura využívá kurzor, proměnnou odkazující se na řádek tabulky `table_name%ROWTYPE` (implementováno jako `a Alkoholik%ROWTYPE`) a ošetřující výjimky u nezadaných dat, neočekávané chyby a u vybraných alkoholiků bez zadaného věku a pohlaví. Jejím úkolem je v cyklu zjistit všechny věky a pohlaví u alkoholiků, rozdělit na muže a ženy a roztrždit je do zadaných věkových kategorií (do 18 let, 18 až 29 let, 30 až 39 let, 40 až 49 let, 50 až 59 let, 60 až 69 let, 70 až 79 let, 80 až 89 let a 90 let a více). Na konci běhu procedury se vypíše procentuální zastoupení každé věkové skupiny a procentuální zastoupení mužů a žen a procentuální zastoupení každé věkové skupiny. Můžeme tedy lehce zjistit zajímavosti o tom např. kolik žen je mezi alkoholiky nebo zda jsou mezi alkoholiky i ti, kteří ještě nedosáhli plnoletosti.

Druhá procedura využívá atributu `%TYPE` a na základě `SELECT` dotazu vybere informace o příslušné kontrole, na základě identifikátoru kontroly, která je v argumentu procedury předána. Atributy z příslušných tabulek jsou vloženy do proměnných v proceduře a následně vypsány. Vypíše se tak identifikátor kontroly, jaký alkoholik byl vyšetřen, jaký odborník kontrolu provedl, jaká byla míra alkoholu, původ a typ.

Poslední procedura funguje jako nápověda. Po zavolání vypíše přivítání, a hlavně pak výčet určených míst pro formální schůze, typy míry expertízy u odborníka a výčet typů alkoholu. Tato procedura byla vložena jako nápověda hlavně kvůli tomu, že uživatel databáze může být někdy, co se týče omezujících triggerů ztracen a může tápat, v jakém formátu má data zadat.

3.4 Naplnění tabulky modelovými údaji

Jsou vytvořeny modelové položky, některé bez definovaného primárního klíče, aby se demonstrovalo použití generátoru primárních klíčů ze sekvence. Alkoholiků je vytvořeno více, zvláště pro demonstraci procedury o procentuálním zastoupení.

3.5 SELECT dotazy

Databáze obsahuje dva dotazy využívající spojení dvou tabulek. Jeden z nich vybírá, kteří odborníci a alkoholici mají věk mezi 40 a 50 lety. Druhý dotaz vybírá ženy mezi patrony a odborníky.

Databáze obsahuje jeden dotaz využívající spojení tří tabulek. Vybírá, kdo všechno se účastní schůze s id 401.

Databáze obsahuje dva dotazy s klauzulí GROUP BY a agregační funkcí. První vybírá počet odborníků se střední mírou expertízy. Druhý se ptá, kolik požití má zapsán alkoholik s id 002, zobrazí i informace o něm – id a původ typ alkoholu.

Databáze obsahuje jeden dotaz obsahující predikát EXISTS. Vybere alkoholiky, co si zaevidovali pouze požití některého míchaného nápoje.

Databáze obsahuje jeden dotaz s predikátem IN s vnořeným selectem. Vybírá, kteří alkoholici požili alkohol v březnu 2019.

Dále jsou uvedeny dva triviálnější dotazy navíc.

3.6 INDEX a EXPLAIN PLAN

Pro vyzkoušení INDEX a EXPLAIN PLAN jsme vytvořili jednoduchý dotaz SELECT:

```
SELECT A.alkoholik_id, A.vek, COUNT(*) AS pocet_poziti
FROM Alkoholik A, Poziti_alkoholu P
WHERE A.alkoholik_id = P.alkoholik_id_poziti
GROUP BY A.alkoholik_id, A.vek;
```

Po prvním běhu byl vytvořen index a ten byl použit při druhém běhu SELECT dotazu. Pak jsme oba běhy porovnali:

První běh bez použití indexu:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		14	546	7 (15)	00:00:01
1	HASH GROUP BY		14	546	7 (15)	00:00:01
* 2	HASH JOIN		14	546	6 (0)	00:00:01
3	TABLE ACCESS FULL	POZITI_ALKOHOLU	14	182	3 (0)	00:00:01
4	TABLE ACCESS FULL	ALKOHOLIK	20	520	3 (0)	00:00:01

Druhý běh po použití indexu:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		14	546	4 (25)	00:00:01
1	HASH GROUP BY		14	546	4 (25)	00:00:01
2	NESTED LOOPS		14	546	3 (0)	00:00:01
3	TABLE ACCESS FULL	ALKOHOLIK	20	520	3 (0)	00:00:01
* 4	INDEX RANGE SCAN	INDEX_POZITI_ALKOHOLIK	1	13	0 (0)	00:00:01

Při druhém běhu se parametr `Cost` (počet přístupů na disk) snížil, avšak procentuálně se zátěž u CPU zvýšila. Při provedení se využívá hashovací klíč. Druhý parametr `Table Access Full` (procházení tabulky od začátku do konce) byla v druhém běhu nahrazena použitím indexu, hledá se platné ROWID z použitého indexu. Další zrychlení by mohlo nastat při spojení dvou tabulek například ve formě použití klauzule `JOIN` a s různými jejími variantami.

3.7 Přístupová práva pro druhého člena týmu

Přístupová práva pro druhého člena týmu byla definována následovně – druhý člen představuje asistenta v Klubu anonymních alkoholiků, který má na starost spravovat veškeré schůze v klubu. Dostane tedy povolení pro veškerou práci s tabulkami `Schuze`, `Formalni` a `Neformalni` a další omezenější v práci s ostatními – tam získává pouze právo provádět `SELECT` dotazy u těchto tabulek. Získá také právo vykonávat procedury, které byly vytvořeny.

3.8 Materializovaný pohled

Materializovaný pohled patří uživateli `xhomo121`. V něm se provede `SELECT` dotaz, který spočítá kolikrát se dané stejné datum a čas u schůzky v tabulce vyskytlo. Pohled se naplní ihned po vytvoření, bude využit optimalizátor. `SELECT` dotaz je poté proveden poprvé a poté podruhé, s vložením nového údaje, tedy se zvýší počet určitého vloženého umístění. Nutno však podotknout, že se materializovaný pohled aktualizuje až po příkazu `COMMIT`.