

64. Mafie

Projektová dokumentácia

Natália Marková (xmarko20)

Tereza Burianová (xburia28)

Obsah

1	Úvod	2
2	Zadanie	2
3	2. časť - Vytvorenie základných objektov schémy databázy	3
3.1	Vytvorenie tabuliek	3
3.2	Naplnenie tabuliek dátami	3
4	3. časť - Výber dát pomocou SELECT	3
4.1	Spojenie dvoch tabuliek	3
4.2	Spojenie troch tabuliek	4
4.3	GROUP BY + agregáčna funkcia	4
4.3.1	agregáčna funkcia COUNT	4
4.3.2	agregáčna funkcia SUM	4
4.4	EXISTS	4
4.5	Predikát IN s vnoreným SELECT	5
5	4. časť - Vytvorenie pokročilých objektov schémy databázy	5
5.1	Triggery	5
5.1.1	uzemi_seq()	5
5.1.2	familie_check()	5
5.2	Procedúry	6
5.2.1	vek_pri_prijeti	6
5.2.2	statistika_ucasti_donu_na_setkani	6
5.3	EXPLAIN PLAN	6
5.3.1	Bez indexu	7
5.3.2	S indexom	7
5.4	Prístupové práva	8
5.5	Materializovaný pohľad	8
6	Spúšťanie skriptu	9

1 Úvod

Cieľom riešeného projektu je návrh a implementácia relačnej databázy na tému **Mafie**, ktorá je špecifikovaná v nasledovej sekcii.

2 Zadanie

Padrino Guiseppe, priateľ všetkých brnenských mafiánov, Vás požiadal o vytvorenie informačného systému pre organizáciu kriminálnej činnosti a stretnutia popredných brnenských rodín. V čele každej rodiny je Don (napr. Don Calzone, Don Quatro Formagi, Don Funghi, apod.), Rešpektovaná osoba, ktorá organizuje dianie rodiny, určená bežnými atribútmi (meno, vek, veľkosť topánok, ???). Samotná família potom pozostáva z radových členov, ktorí nemusia byť pokrvne spriaznení a ktorí sprostredkovávajú všetku kriminálnu činnosť, ako je odstraňovanie nevhodných osôb, dealovanie drog, tlačenie falošných peňazí, atď., Don si však nikdy nešpiní ruky. Kriminálna činnosť prebieha v rámci územia, určených gps súradnicami, adresou, rozlohou, a môže byť buď riadnym rajónom jednej z familií, alebo neutrálnym územím, kde môže operovať väčší počet familii. Do samotnej kriminálnej činnosti sú potom zapojení konkrétne radoví členovia v rôznych obdobiach (tzn. Radový člen môže najprv dealovať drogy a potom prejsť na odstraňovanie osôb), pričom každá činnosť je priamo vedená jednou z familií a má rôznu dobu trvania a táto operácia má vždy unikátne meno. Do týchto činností môže byť zapojených viac rodín, to v prípade, že dve rodiny uzavrú alianciu, ktoré však nemusia mať dlhé trvanie. Jednou z hlavných činností sú potom vraždy, pri ktorých je vedené nielen, kde boli vykonané, ale aj kto ich vykonal a na kom ich vykonal. Niektoré z týchto vražd sú potom objednané samotnými Donmi, iné sú spontánne, bez bližšej objednávky. Cieľom vraždy však nikdy nie je Don. Systém pravidelne organizuje stretnutia Donov, kde sa preberajú výsledky jednotlivých rodín a prebiehajú buď v rámci neutrálneho územia, alebo sú hostené niektorou z popredných rodín. Systém potom zasiela Donom pozvánky a očakáva od nich potvrdenie účasti.

3 2. časť - Vytvorenie základných objektov schémy databázy

3.1 Vytvorenie tabuliek

Don
Familie
Uzemi
R_Spada_Pod
Setkani_Donu
R_Ucast_Na_Setkani
Radovy_clen
R_Clen_Role
Aliance
Kriminalni_cinnost
R_Clen_Cinnost
Objednavka

Primárne kľúče tabuliek sú vo väčšine prípadov celé čísla (int) alebo stringy (varchar (255)). Pri vzťahoch n..n sme využili zložené primárne kľúče (napr. tabuľka R_Clen_Cinnost). Pri špecifickom definovaní atribútov, sme využívali aj regulárne výrazy príkazom REGEXP_LIKE.

3.2 Naplnenie tabuliek dátami

Dáta sme do tabuliek plnili pomocou príkazu INSERT INTO a následne konkrétne definovali do VALUES.

```
INSERT INTO Objednavka(ID_objednavky, Datum, Stav, Alias, Jmeno_operace)
VALUES( ID_objednavky: 1, TO_DATE( '2020-03-01 15:15', 'YYYY-MM-DD HH24:MI' ), Stav: 'prijata', Alias: 'Veduci', Jmeno_operace: 'Zabitie Fera');
INSERT INTO Objednavka(ID_objednavky, Datum, Stav, Alias, Jmeno_operace)
VALUES( ID_objednavky: 2, TO_DATE( '2020-06-14 23:00', 'YYYY-MM-DD HH24:MI' ), Stav: 'prijata', Alias: 'Zabijak', Jmeno_operace: 'Operace Jarmila');
```

4 3. časť - Výber dát pomocou SELECT

4.1 Spojenie dvoch tabuliek

Funkcia: Výber všetkých rolí Joža Poplety

```
SELECT
    R.Role,
    R.Role_od,
    R.Role_do
FROM Radovy_clen C, R_Clen_Role R
WHERE C.Jmeno='Jozo' and C.Prijmeni='Popleta' and R.ID_clena=C.ID_clena;
```

Funkcia: Výber všetkých členov famílie Kolárovyh

```
SELECT
    R.Jmeno,
    R.Prijmeni
FROM Familie F, Radovy_clen R
WHERE R.Nazev_familie='Kolarovi' and R.Nazev_familie=F.Nazev_familie;
```

4.2 Spojenie troch tabuliek

Funkcia: Výber všetkých vražd objednaných Vedúcim

```
SELECT
    K.Jmeno_operace,
    D.Alias,
    O.ID_objednavky
FROM Kriminalni_cinnost K, Objednavka O, Don D
WHERE K.type='Vrazda' and O.Alias=D.Alias and D.Alias='Veduci' and O.Jmeno_operace=K.Jmeno_operace;
```

4.3 GROUP BY + agregáčná funkcia

4.3.1 agregáčná funkcia COUNT

Funkcia: Koľko stretnutí sa jednotliví Donovia zúčastnili

```
SELECT
    D.Alias,
    COUNT(U.ID_setkani) AS Pocet_setkani
FROM R_Ucast_Na_Setkani U, Don D
WHERE U.Alias=D.Alias
GROUP BY D.Alias;
```

4.3.2 agregáčná funkcia SUM

Funkcia: Akú celkovú rozlohu majú územia, ktoré v histórii patrili jednotlivým famíliám

```
SELECT
    S.Nazev_familie,
    SUM(U.Rozloha) AS Celkova_rozloha
FROM Uzemi U, R_Spada_Pod S
WHERE S.ID_uzemi=U.ID_uzemi and S.Nazev_familie IS NOT NULL
GROUP BY S.Nazev_familie;
```

4.4 EXISTS

Funkcia: Výber členov, ktorí sa zúčastnili len operácie 'Zabitie Fera'

```
SELECT C.*
FROM Radvy_clen C, R_Clen_Cinnost R
WHERE R.Jmeno_operace='Zabitie Fera' and C.ID_clena=R.ID_clena and
NOT EXISTS (
    SELECT *
    FROM R_Clen_Cinnost R
    WHERE R.Jmeno_operace<>'Zabitie Fera' and C.ID_clena=R.ID_clena
);
```

4.5 Predikát IN s vnoreným SELECT

Funkcia: Výber stretnutí donov, ktoré sa konali na územiach s rozlohou menšou než 100

```
SELECT *
FROM Setkani_Donu
WHERE ID_uzemi IN (
    SELECT ID_uzemi
    FROM Uzemi
    WHERE Rozloha < 100
);
```

5 4. časť - Vytvorenie pokročilých objektov schémy databázy

5.1 Triggery

Implementovali sme 2 typy databázových triggerov.

5.1.1 uzemi_seq()

Úlohou tohto triggeru je generovať ID tabuľke Území. Generuje ich od čísla 1 a postupne pridáva +1 každému nasledujúcemu riadku v tabuľke. Trigger je spustený vždy pri pridaní novej položky a pomocou NEXTVAL ukladá nové ID danej položke.

V tabuľke môžeme vidieť, že každé ID_UZEMI v každej položke je inkrementované podľa aritmetickej postupnosti: $a_{n+1} = a_n + d$; kde $d = 1$

	ID_UZEMI	ULICE	MESTO	PSC	ROZLOHA	GPS
1	1	Sedmikraskova	Brno	95501	35	34 N, 180 W
2	2	Pampeliskova	Cesky Krumlov	45477	64	39 N, 120 W
3	3	Tulipanova	Sered	25874	112	69 N, 69 E

5.1.2 familie_check()

Nakoľko sa vytvorenej kriminálnej činnosti môžu zúčastniť iba členovia famílie, ktorá činnosť zastrešuje, vytvorili sme tento databázový trigger. Je spúšťaný pri pridávaní novej položky a jeho úlohou je skontrolovať či pridávaný člen patrí do famílie, ktorá činnosť vedie. V priaznivom prípade danú položku pridá, inak spustí príkaz RAISE_APPLICATION_ERROR s hodnotou –20000, ktorý užívateľovi vypíše aj nasledovnú chybovú hlášku.

```
[2021-04-28 16:36:43] [72000][20000]
[2021-04-28 16:36:43] ORA-20000: Tento radovy clen nepatri do Familie vedouci tuto cinnost.
[2021-04-28 16:36:43] ORA-06512: na "XBURIA28.FAMILIE_CHECK", line 10
[2021-04-28 16:36:43] ORA-04088: chyba během provádění triggeru 'XBURIA28.FAMILIE_CHECK'
[2021-04-28 16:36:43] Position: 12
```

5.2 Procedúry

Vytvorili sme 2 procedúry, ktoré pre demonštráciu v kóde spúšťame nasledovným spôsobom:

```
BEGIN
vek_pri_prijeti();
statistika_ucasti_donu_na_setkani();
END;
```

5.2.1 vek_pri_prijeti

Procedúra spočíta vek členov, ktorý mali pri prijatí do famílie. Využívame údaje z dátumu narodenia a dátumu prijatia, kde pomocou príkazu EXTRACT vždy vyberieme parameter (rok, mesiac alebo deň) s ktorým operujeme v podmienkach, ktoré daný vek počítajú.

Po výpočte sa do konzoly vypíše každý radový člen s príslušným vekom nasledovne:

```
[2021-04-28 16:40:12] Vek clenu pri jejich prijeti do Familie:
[2021-04-28 16:40:12] ID 1, Jozo Popleta - 28 let
[2021-04-28 16:40:12] ID 2, Jan Spageta - 19 let
[2021-04-28 16:40:12] ID 3, Hildegarda Nugeta - 34 let
[2021-04-28 16:40:12] ID 4, Berta Koketa - 20 let
```

5.2.2 statistika_ucasti_donu_na_setkani

Úlohou tejto procedúry je štatistika. V prvom rade procedúra pomocou agregáčnej funkcie COUNT spočíta všetkých súčasných Donov, ktorí sa stretnutí môžu zúčastniť. Potom kontroluje každé stretnutie osobitne. Účasť počítame nasledovne so zaokrúhlením na dve desatinné miesta:

$$ucast := ROUND(pocet_ucastniku / celkem_donu * 100, 2);$$

Po výpočte sa do konzoly vypíše účasť na stretnutí v percentách nasledovne:

```
[2021-04-28 16:40:12] Percentualni ucast na setkanih Donu:
[2021-04-28 16:40:12] Setkani 1: 100 %
[2021-04-28 16:40:12] Setkani 2: 66,67 %
[2021-04-28 16:40:12] Setkani 3: 33,33 %
```

5.3 EXPLAIN PLAN

Pomocou EXPLAIN PLAN sme zobrazili nasledovné informácie pre vybraný príkaz.

- postupnosť realizácie operácií,
- informácie o výkonnostnej cene pre každú operáciu
- čas vypracovania

Pre demonštráciu sme zvolili nasledovný príkaz:

```
EXPLAIN PLAN FOR
SELECT
  S.Nazev_familie,
  SUM(U.Rozloha) AS Celkova_rozloha
FROM Uzemi U, R_Spada_Pod S
WHERE S.ID_uzemi=U.ID_uzemi and S.Nazev_familie IS NOT NULL
GROUP BY S.Nazev_familie;
SELECT * FROM table (DBMS_XPLAN.DISPLAY);
```

Jeho úlohou je zobraziť názov famílie a jej vlastnené územie v km^2 počas pôsobenia v krajine.

5.3.1 Bez indexu

Zavolaním EXPLAIN PLAN vidíme, že boli postupne vykonané operácie SELECT, GROUP BY, JOIN tabuliek a následné prístupenie k ich dátam potrebným na vykonanie príkazu. Taktiež je zobrazená cena a čas trvania príkazu.

	PLAN_TABLE_OUTPUT							
1	Plan hash value: 4054663961							
2								
3	-----							
4	Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	
5	-----							
6	0	SELECT STATEMENT		4	672	7 (15)	00:00:01	
7	1	HASH GROUP BY		4	672	7 (15)	00:00:01	
8	* 2	HASH JOIN		4	672	6 (0)	00:00:01	
9	3	TABLE ACCESS FULL	UZEMI	3	78	3 (0)	00:00:01	
10	* 4	TABLE ACCESS FULL	R_SPADA_POD	4	568	3 (0)	00:00:01	
11	-----							
12								
13	Predicate Information (identified by operation id):							
14	-----							
15								
16	2	access("S"."ID_UZEMI"="U"."ID_UZEMI")						
17	4	filter("S"."NAZEV_FAMILIE" IS NOT NULL)						
18								
19	Note							
20	-----							
21	- dynamic statistics used: dynamic sampling (level=2)							

5.3.2 S indexom

Pre zníženie možnej ceny, sme vytvorili nasledovné indexy:

```
CREATE INDEX Uzemi_index ON Uzemi(ID_uzemi, Rozloha);
CREATE INDEX Spada_pod_index ON R_Spada_Pod(ID_uzemi, Nazev_familie);
```

Sú vytvorené pre stĺpce, ku ktorým v tabuľke pristupujeme a sú priamo využité vo výberovej podmienke WHERE a následne použité v príkaze SELECT.

Po druhom spustení EXPLAIN PLAN už s aplikovaným indexom vidíme, že na prehľadávanie už nebola zvolená celá tabuľka, ale len stĺpce definované indexom. Táto zmena priaznivo pôsobila na cenu, nakoľko môžeme vidieť, že sa podstatne znížila.

	PLAN_TABLE_OUTPUT							
1	Plan hash value: 2938523198							
2								
3	-----							
4	Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	
5	-----							
6	0	SELECT STATEMENT		4	672	3 (34)	00:00:01	
7	1	HASH GROUP BY		4	672	3 (34)	00:00:01	
8	* 2	HASH JOIN		4	672	2 (0)	00:00:01	
9	3	INDEX FULL SCAN	UZEMI_INDEX	3	78	1 (0)	00:00:01	
10	* 4	INDEX FULL SCAN	SPADA_POD_INDEX	4	568	1 (0)	00:00:01	
11	-----							
12								
13	Predicate Information (identified by operation id):							
14	-----							
15								
16	2 - access("S"."ID_UZEMI"="U"."ID_UZEMI")							
17	4 - filter("S"."NAZEV_FAMILIE" IS NOT NULL)							
18								
19	Note							
20	-----							
21	- dynamic statistics used: dynamic sampling (level=2)							
22	- this is an adaptive plan							

5.4 Prístupové práva

Prístupové práva boli pridané vedúcim tímu (xmarmo20) pre druhého člena (xburia28) na každú tabuľku v databáze a taktiež k oboj procedúram pomocou príkazu GRANT ALL ON na tabuľky a GRANT EXECUTE ON na procedúry.

5.5 Materializovaný pohľad

Implementovaný materializovaný pohľad sme aplikovali na zobrazenie radových členov, nakoľko sa k tejto tabuľke často prístupuje.

	ID_CLENA	JMENO	PRIJMENI
1	1	Jozo	Popleta
2	2	Jan	Spageta

Sledujú sa pridané zmeny v tejto tabuľke, ktoré sa aktualizujú po vykonaní potvrdzujúceho príkazu COMMIT.

Demonštrácia:

Po vložení novej položky do tabuľky bol pridaný príkaz COMMIT, a tak sa zobrazí už aktualizovaná tabuľka:

	ID_CLENA	JMENO	PRIJMENI
1	1	Jozo	Popleta
2	2	Jan	Spageta
3	5	Klementina	Spagetova

V prípade, že po vložení novej položky do tabuľky nebol pridaný príkaz COMMIT, sa táto zmena v pohľade neprejaví, nakoľko nie je aktualizovaná. Preto sa zobrazí pôvodná.

	ID_CLENA	JMENO	PRIJMENI
1	1	Jozo	Popleta
2	2	Jan	Spageta

6 Spúšťanie skriptu

Pri spúšťaní sa spustia všetky časti projektu zároveň aj s demonštráciou. Táto demonštrácia skončí vždy s chybovou hláškou, nakoľko vď aka nej prezentujeme funkčnosť triggeru.

V prípade kontroly funkčnosti celého skriptu je možnosť zakomentovať/odstrániť riadky 526-528, ktoré túto výnimku vyvolávajú.