Elektronické stávkovanie

Správa(1)

Projekt databázy (2)

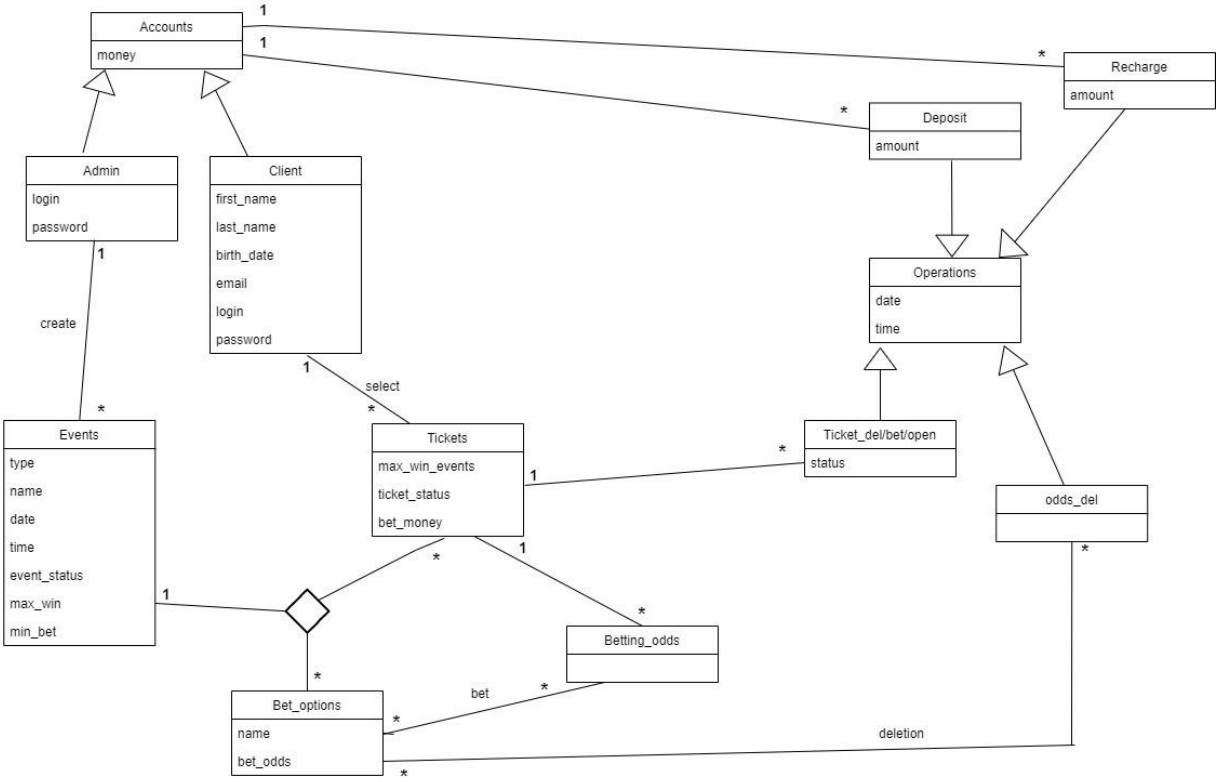
Ján Zdarilek

10.05.2020

1.O dokumente

Tento dokument predstavuje záverečnú správu k odovzdanej práci z predmetu Databázy 2. Názov projektu, ktorému som sa venoval je Elektronické stávkovanie. V tomto dokumente ukážem dátový model, relačnú databázu. Budem sa venovať aj organizácii kódu v programe a na konci aj samotnej optimalizácií niektorej časti programu.
Táto aplikácia je konzolová. Pomocou príkazov napísaných v konzole je možné zakladať klientov stávkovej kancelárie, vytvárať tikety a rôzne ďalšie.

2. Dátový model



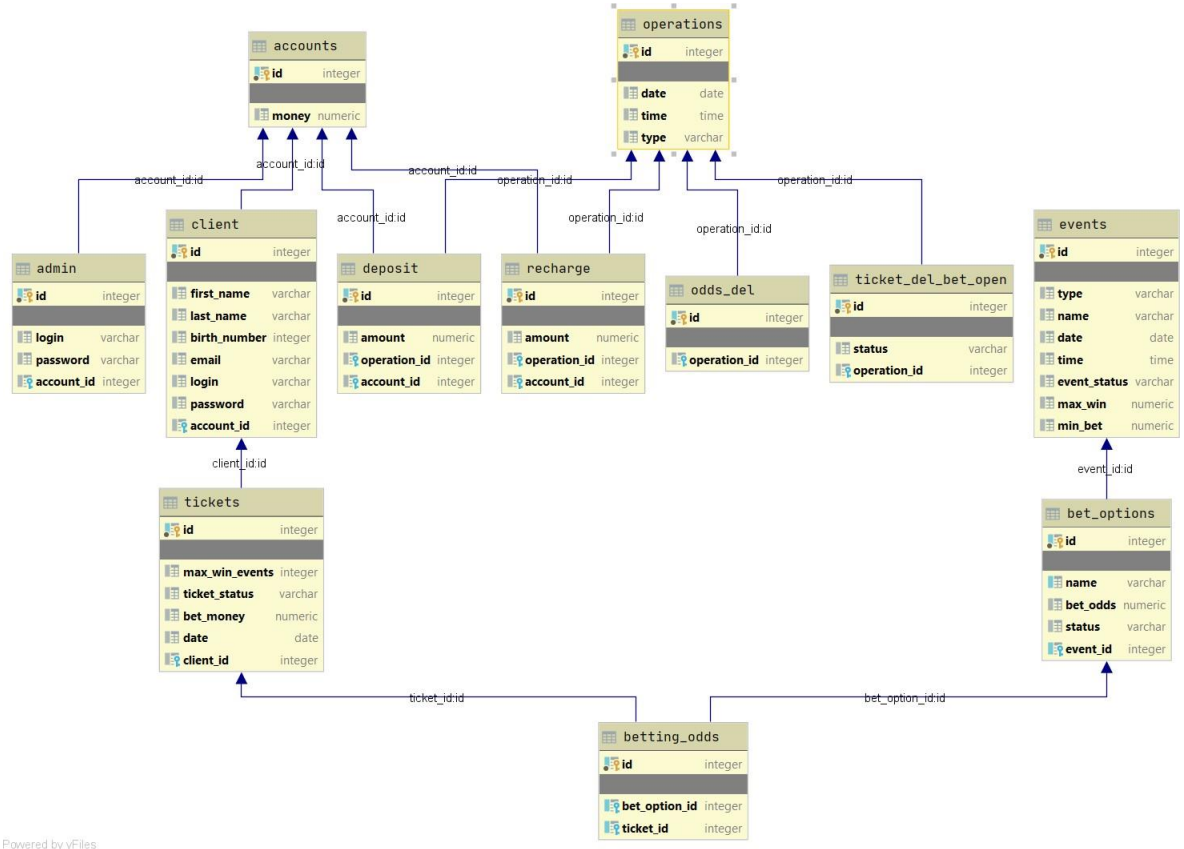
Popis dátového modelu:

- V tabuľke Accounts sú uložené údaje o financiách. Má dve podmnožiny Admin a Client (údaje k obom typom prístupov)
- V tabuľke Events sú údaje k podujatiu, ktoré vytvára admin. Položka max_win je koľko je možné maximálne vyhrať na danej udalosti. A min_bet je minimálna stávka. Event_status je údaj, či je0 daná akcia skončená alebo ešte nie.
- V Bet_options sa nachádzajú údaje k možnostiam stávky. K udalosti sú vytvorené možné typy stávok a kurzov k nim.
- V Tickets sa nachádzajú údaje o tom v akom stave je tiket(ticket_status môže byť open/lost/bet/win). Potom max_win_events je nastavený na inú hodnotu pri akumulovanej stávke. A bet_money je, že koľko daný klient vsadil na tiket.
- V tabuľke Operations je uložený dátum a čas vykonanej operácie. Má podmnožiny možných operácií. Deposit ukladá údaj o vložení peňazí na účet klienta. Recharge

ukladá údaj o výbere peňazí. Odds_del má údaj o vymazaní kurzu z otvoreného tiketu. Ticket_del/bet/open má údaje v akom je stave tiket.

3. Relačná databáza

Ukazuje výslednú relačnú databázu, ktorá vznikla transformovaním entitno relačného modelu. Prepojenie klúčov sa nachádza na šípkach.



4. Organizácia kódu

Aplikácia je naprogramovaná v jazyku Java a využíva vzory Row data Gateway a Transaction Script. Prístup do databázy je riešený cez JDBC.

V zdrojovo súbore sa nachádzajú triedy predstavujúce tabuľky. Každá z nich má funkcie insert, delete, update... podľa vzoru RDG. Cez triedu Main spúšťam celú aplikáciu. Spojenie sa udržiava v triede DBContext.

Zložitejšie doménové operácie sú realizované pomocou vzoru Transaction Script. Súvisiace doménové funkcie sú zoskupené do jednej triedy. Sú to:

Auto_evaluate_tickets – automatické vyhodnocovanie tiketov

Delete_bet_from_ticket – vymazanie stávky z tiketu

Delete_notbet_ticket – vymazanie nepodaného tiketu

Deposit_player – vklad peňazí na konto

Withdraw – výber peňazí z účtu

Win_statistic_player - štatistika hráča / percento úspešnosti

Revenue_from_event – Zárobok podľa udalosti

Funkcie find a tomu podobné boli realizované ako statické metódy v triedach tabuliek.

Aplikácia obsahuje sql súbory. Jeden na vytváranie tabuliek a druhá na generovanie dát.

5. Optimalizácia SQL

Optimalizoval som časť v triede Revenue_from_event. Ide o časť kde vybrám všetky tikety ktoré boli v stave lost.

Pôvodne som mal kód rozdelený na dva selecty. A cez cyklus a podmienky som zisťoval, ktoré údaje sa mi hodia. A výslednú sumu som spočítaval v premenných v programe.

Ako optimalizáciu som dal, že som ich spojil do jedného a ako výsledok dostávam priamo údaje ktoré potrebujem. Sumu podaných peňazí počítam tiež priamo v sql.

Pôvodný kód: (Kod nie je uplne presny, len priblizny)

```
Select ticket_id FROM
(Select ticket_id from bet_options JOIN betting_odds ON
bet_option.id=betting_odds.bet_option_id
WHERE bet_option.event_id=?)
ResultSet r = s.executeQuery()
While(true):
        Pole.add(ticket_id)
```

```
Select bet_money, ticket_id from
Tickets WHERE tickets.ticket_status='lost';.
ResultSet r = s.executeQuery()
While(true):
        <Tickets>Pole2.add(bet_money, ticket_id)
```

```
For (i : pole){
        For (j:pole2){
                If(i==pole2[1]){
                        bet_money = bet_money+pole[0]
                }
        }
}
```

Upravený kód:

```
Select sum(bet_money) From
(Select ticket_id from bet_options JOIN betting_odds ON
bet_option.id=betting_odds.bet_option_id
WHERE bet_option.event_id=?)t1 JOIN
Tickets ON t1.ticket_id=tickets.id  WHERE tickets.ticket_status='lost';
```

Veľkosť tabuľky	Rýchlosť pred optimalizáciou	Rýchlosť po optimalizáciou
1000	34ms	15ms
10000	124ms	63ms
100000	721ms	363ms
1000000	-	3914ms

*Pri údajoch z merania času pred optimalizáciou mi chýba údaj pri 1milióne. Avšak je vidno, že optimalizácia pomohla pri zrýchlení programu.

6. Vybraný riešený problém

Pôvodne som mal v pláne riešiť Administrátorov, ktorí by jediný mohli mazať/upravovať... skoro ľubovolné údaje. Avšak po konzultácii sa dohodlo, že to nie je potrebné a do tabuľky Admin som nakoniec nevkladal nič. Táto tabuľka tam však ostáva pre možný ďalší vývoj kedže neskôr to môže byť veľmi užitočné. Admin je teda len taký pseudo. Avšak účet s peniazmi má vytvorený kvôli tomu aby pri výhre nejakého klienta bolo možné odčítať peniaze z Admin účtu a pripočítať ich klientovi.