Project Deliverable 4

Writing your Application Program

Presented to

Bettina Kemme COMP-421 – Database Systems

By Patrick Desmarais (260 329 253) Simon Hlywa (260 231 002) Guillaume Viger (260 309 396)

> McGill University March 28th, 2011

1. Stored Procedure

The stored procedure is called refresh_privileges. It looks up transactions to find users with 25 transactions or more, and updates their privilege_status to the appropriate code for the month. Gold members have access to special offered_entities depending on their privilege code. They can buy one special offered_entity per month. This privilege is renewed the first day of each month.

This stored procedure will be run over all users; hence, in the worst case, all N users will have their Users tuple updated. All Transactions are analyzed to account if they are non-zero transactions and then, counted to verify if the users will have its privilege_status updated to the special monthly code.

1.1 Procedure

Data:

Data have been generated using the Java program called DataGenerator in the dataGen directory. The generated data have been inserted into DB2 commands such as db2 -tf users.del. It is assumed that tables were created and that no tuples were inserted to avoid any collisions.

Script:

The script file can be found in stored_procedure/store_procedure.clp. The README_StoredProcedure file explains how to test the stored procedure.

New Users table:

db2 => describe table users					
Column name	Type schema	Type name	Length	Scale	Nulls
EMAIL PSWD CREDIT_CARD PRIVILEGE_STATUS	SYSIBM SYSIBM SYSIBM SYSIBM	VARCHAR VARCHAR CHARACTER SMALLINT	100 50 16 2	0 0 0 0	No Yes Yes Yes

4 record(s) selected.

Note that the privilege/monthly code must be different from 0 and 1, since these offered_entities status are already taken. The range is now between 2 and 14.

1.2 Procedure in Action

Before calling procedure:

db2 => select email, privilege_status from users

EMAIL	PRIVILEGE_STATUS
specialOffers@website.com	0
ldln@86n5pnwy.vgo	0
2ustranspuec9pntlhmgjpgs@epmm.hb	0
75jm2tjpojkuq1qbjobz@e.ag	0
76wruj18sz89@rnzouwq9.wl	0
77pldnqff@mcd.jvm	0
78t5ibpkvgxwq5j42isy4zm9eip@w9gyrb.xs	0
79gsbtbuyo1fyajic5lyqigfto@rkdlvzjy.skh	0
80pcqlof@iklvkr411.fso	0

Calling procedure:

db2 => call refresh_privileges(4)

Return Status = 0

[Privilege code 4 for March special offers.]

After calling procedure:

db2 => select email, privilege_status from users

EMAIL	PRIVILEGE_STATUS
specialOffers@website.com	4
1dln@86n5pnwy.vgo	4
2ustranspuec9pntlhmgjpgs@epmm.hb	4
75jm2tjpojkuq1qbjobz@e.ag	4
76wruj18sz89@rnzouwq9.wl	0
77pldnqff@mcd.jvm	4
78t5ibpkvgxwq5j42isy4zm9eip@w9gyrb.xs	0
79gsbtbuyolfyajic5lyqigfto@rkdlvzjy.skh	0
80pcqlof@iklvkr411.fso	4

Verify validity of procedure:

One example of with a user that has more than or equal to 25 transactions, which means that his privilege is updated.

```
db2 => select count(*) from Users U, Transactions T where U.email = '1dln@86n5pnwy.vgo' and
(U.email = T.provider or U.email = T.receiver)
```

1 -----111

1 record(s) selected.

One example of with a user that has less than 25 transactions, which means that his privilege is not updated.

```
db2 => select count(*) from Users U, Transactions T where U.email = '76wruj18sz89@rnzouwq9.wl'
and (U.email = T.provider or U.email = T.receiver)
1
```

± _____

```
23
```

1 record(s) selected.

Operations are valid!

test procedure.clp demonstrates a similar and simpler behaviour with two users.

2. User-Friendly Application

Example of menu and run of applications:

```
*****
*Welcome to WoW Store!*
******
-WECLOME-
      1. Login
      2. Register a new user
      3. Quit
Your selection: 2
--Register New User--
Username: patrick@wow.com
Password: p
Credit card (16 numbers no spaces): 000000000000000
Registration succesful!
-MAIN MENU-
Please select and option:
      1. Browse all offers
      2. Search offers
      3. Buy an offer
      4. Post an offer
      5. View transaction history
       6. Logout
      7. Quit
Your selection: 7
*****
*Welcome to WoW Store!*
-WECLOME-
      1. Login
      2. Register a new user
      3. Quit
Your selection: 2
--Login--
Username: patrick@wow.com
Password: p
Login succesful!
-MAIN MENU-
Please select and option:
      1. Browse all offers
       2. Search offers
      3. Buy an offer
      4. Post an offer
      5. View transaction history
       6. Logout
      7. Quit
Your selection: ...
```

See WowStore.java, WowDisplay.java, RunStore.java (main class) and ResultSetWrapper.java for the code files of the program inside the application directory.

3. Indexes

Indexes will be created on Character races and levels.

See scriptIndex.clp for the Indexes creation.

See timeDB.java for the java program timing the performance with and without the indexes.

See scriptTime.sh for the script executing scriptIndex.clp and timeDB.java.

The average results are the following after 10 runs:

Before the index creation:	Query 1: 471 ms Query 2: 224 ms
After the index creation:	Query 1: 291 ms Query 2: 225 ms