|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** ***ИУК «Информатика и управление»***

**КАФЕДРА** \_\_***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

**ЛАБОРАТОРНАЯ РАБОТА №4**

**«Использование хранимых процедур»**

**ДИСЦИПЛИНА: «Базы данных»**

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК4-52Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Карельский М.К. )  (Подпись) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Глебов С.А. )  (Подпись) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |

Калуга, 2022

**Цель:** сформировать практические навыки разработки хранимых процедур.

**Задачи:**

* Научиться создавать хранимые процедуры выборки
* Изучить оператор suspend
* Научиться создавать хранимые процедуры действия

**Предметная область** – социальная сеть.



Рис. 1. Связи таблиц

В базе данных, созданной в предыдущей лабораторной работе, разработать хранимые процедуры выборки и действия.

**Результат:**

Посчитать количество групп и чатов, в которых участвует пользователь:

create procedure COUNT\_GROUPS\_AND\_CHATS (

IN\_USER\_ID bigint)

returns (

OUT\_GROUPS\_NUM integer,

OUT\_CHATS\_NUM integer)

as

BEGIN

SELECT COUNT(\*)

FROM groupmembers

WHERE MEMBERID = :in\_user\_id

INTO :out\_groups\_num;

SELECT COUNT(\*)

FROM groupchatmembers

WHERE MEMBERID = :in\_user\_id

INTO :out\_chats\_num;

END

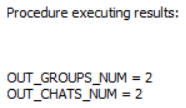


Рис. 2. Результат выполнения при *IN\_USER\_ID* = 1

Найти группы, на которые подписаны друзья пользователя, но не он сам:

create FIND\_INTERESTING\_GROUPS (

IN\_USER\_ID bigint)

returns (

OUT\_GROUP\_ID bigint,

OUT\_GROUP\_NAME varchar(20))

as

BEGIN

for select id, name

from

(select \*

from groups

where id in

(select groupid

from groupmembers

where memberid in

(select secondfriendid as friend

from friends

where firstfriendid = :in\_user\_id

union

select firstfriendid as friend

from friends

where secondfriendid = :in\_user\_id)))

where id not in

(select groupid

from groupmembers

where memberid = :in\_user\_id)

into :out\_group\_id, :out\_group\_name

do suspend;

END

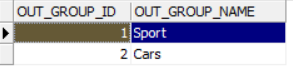


Рис. 3. Результат выполнения при *IN\_USER\_ID* = 3

Найти людей, имеющих общих друзей с пользователем, но не являющихся его друзьями:

create procedure FIND\_POTENTIAL\_FRIENDS (

IN\_USER\_ID bigint)

returns (

OUT\_FRIEND\_ID bigint)

as

BEGIN

for select pf

from

(select secondfriendid as pf

from friends

where firstfriendid in

(select secondfriendid as friend

from friends

where firstfriendid = :in\_user\_id

union

select firstfriendid as friend

from friends

where secondfriendid = :in\_user\_id)

and secondfriendid not in

(select secondfriendid as friend

from friends

where firstfriendid = :in\_user\_id

union

select firstfriendid as friend

from friends

where secondfriendid = :in\_user\_id)

union

select firstfriendid as pf

from friends

where secondfriendid in

(select secondfriendid as friend

from friends

where firstfriendid = :in\_user\_id

union

select firstfriendid as friend

from friends

where secondfriendid = :in\_user\_id)

and firstfriendid not in

(select secondfriendid as friend

from friends

where firstfriendid = :in\_user\_id

union

select firstfriendid as friend

from friends

where secondfriendid = :in\_user\_id))

where pf <> :in\_user\_id

into :out\_friend\_id

do suspend;

END

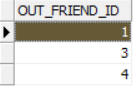


Рис. 4. Результат выполнения при *IN\_USER\_ID* = 6

Подписаться на описанные выше «интересные» группы:

create procedure ADD\_INTERESTING\_GROUPS (

IN\_USER\_ID bigint)

as

declare variable INTERESTING\_GROUP bigint;

declare variable NEXT\_ID bigint;

BEGIN

for select out\_group\_id from find\_interesting\_groups(:in\_user\_id)

into :interesting\_group

do

BEGIN

select max(id) from groupmembers into :next\_id;

:next\_id = :next\_id + 1;

INSERT INTO GroupMembers(Id, GroupId, MemberId)

VALUES (:next\_id, :interesting\_group, :in\_user\_id);

END

END

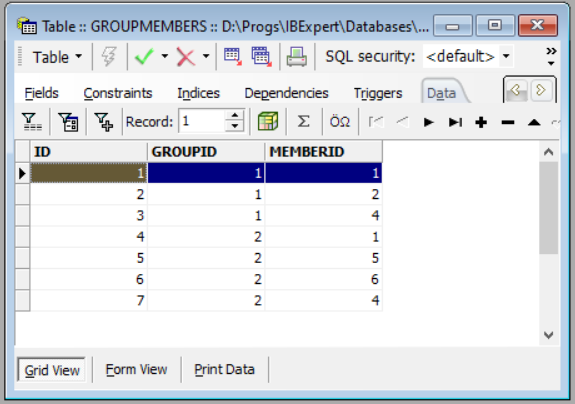
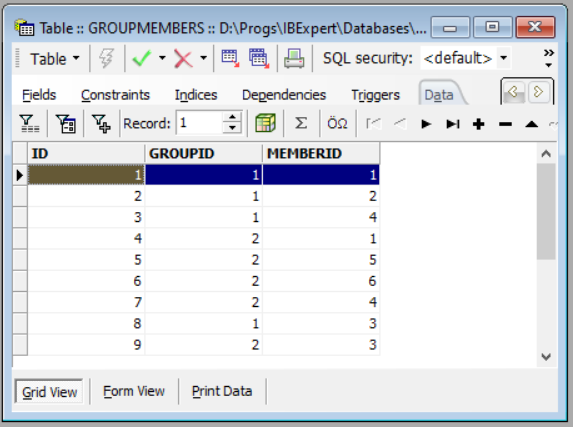
 

Рис. 5. Результат выполнения при *IN\_USER\_ID* = 3

Добавить описанных выше возможных друзей:

create procedure ADD\_POTENTIAL\_FRIENDS (

IN\_USER\_ID bigint)

as

declare variable POTENTIAL\_FRIEND bigint;

declare variable NEXT\_ID bigint;

BEGIN

for select \* from find\_potential\_friends(:in\_user\_id)

into :potential\_friend

do

BEGIN

select max(id) from friends into :next\_id;

:next\_id = :next\_id + 1;

INSERT INTO Friends(Id, FirstFriendId, SecondFriendId)

VALUES (:next\_id, :in\_user\_id, :potential\_friend);

END

END

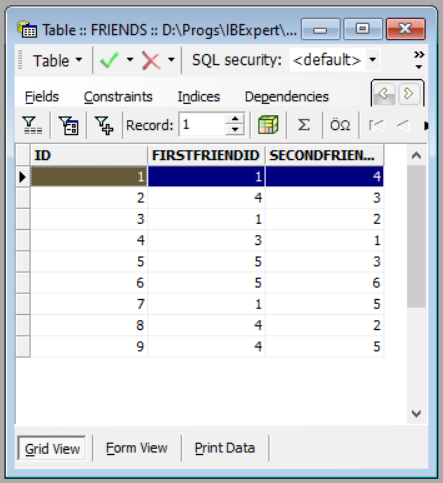
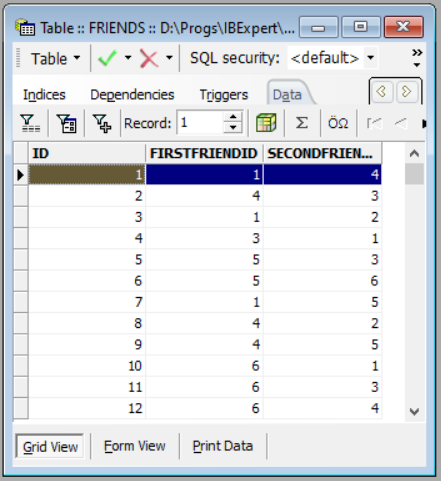
 

Рис. 6. Результат выполнения при *IN\_USER\_ID* = 6

Удалить переписку:

create procedure DELETE\_DIALOG (

IN\_USER\_ID bigint,

IN\_INTERLOCUTOR\_ID bigint)

as

declare variable MESSAGE\_ID bigint;

BEGIN

for select id

from messages

where (senderid = :in\_user\_id and recipientid = :in\_interlocutor\_id) or (senderid = :in\_interlocutor\_id and recipientid = :in\_user\_id)

into :message\_id

do delete from messages

where id = :message\_id;

END

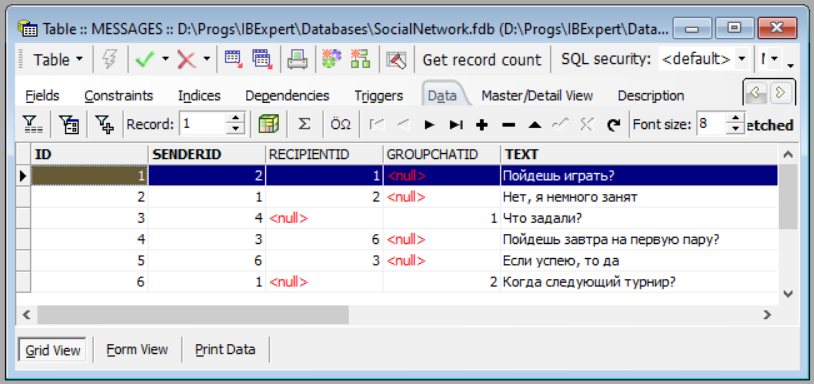
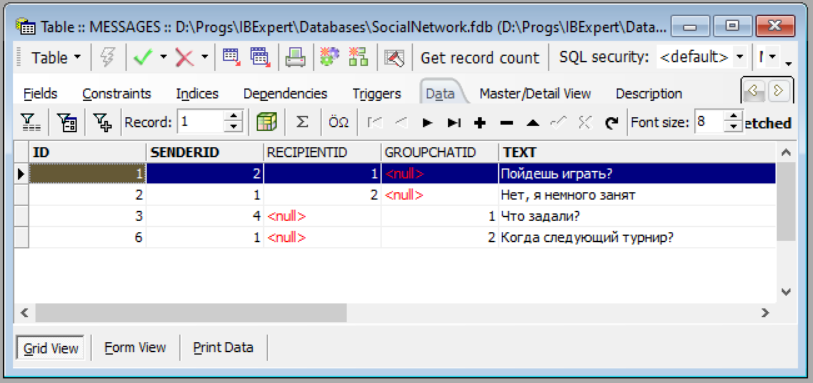
 

Рис. 7. Результат выполнения при *IN\_USER\_ID* = 3 и *IN\_INTERLOCUTOR\_ID* = 6

**Вывод:** в ходе выполнения лабораторной работы были получены практические навыки разработки хранимых процедур.