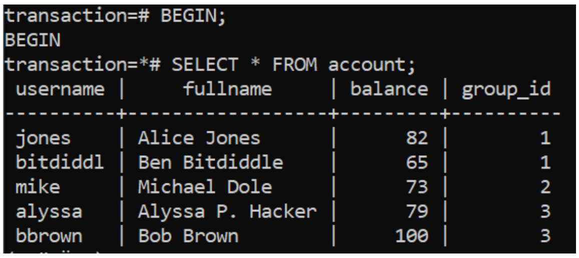
EXERCISE 1

```
CREATE TABLE account(
        username varchar(50) NOT NULL,
        fullname varchar(50) NOT NULL,
        balance integer,
        group_id integer NOT NULL);
INSERT INTO account VALUES
        ('jones','Alice Jones',82,1),
        ('bitdiddl','Ben Bitdiddle',65,1),
        ('mike','Michael Dole',73,2),
        ('alyssa','Alyssa P. Hacker',79,3),
        ('bbrown','Bob Brown',100,3);
```

<table>
<tr><th colspan="2">Read committed isolation level</th></tr>
<tr><th>Terminal 1</th><th>Terminal 2</th></tr>
<tr><td>

1. BEGIN TRANSACTION;
SELECT * FROM account;



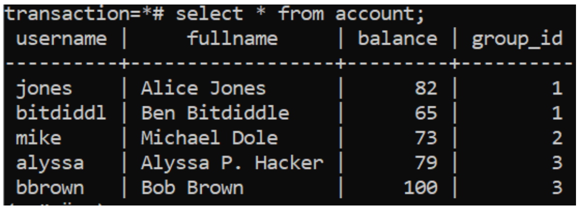
```
transaction=# BEGIN;
BEGIN
transaction=*# SELECT * FROM account;
 username |     fullname     | balance | group_id
----------+------------------+---------+----------
 jones    | Alice Jones      |      82 |        1
 bitdiddl | Ben Bitdiddle    |      65 |        1
 mike     | Michael Dole     |      73 |        2
 alyssa   | Alyssa P. Hacker |      79 |        3
 bbrown   | Bob Brown        |     100 |        3
```

</td><td>

2. BEGIN;
UPDATE account SET username='ajones'
WHERE fullname='Alice Jones';

</td></tr>
<tr><td>

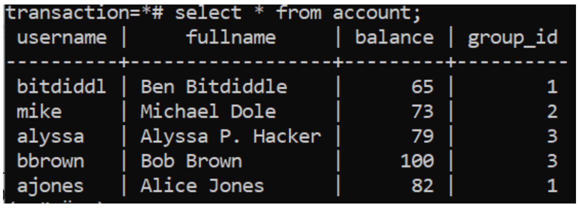
3. SELECT * FROM account;



```
transaction=*# select * from account;
 username |     fullname     | balance | group_id
----------+------------------+---------+----------
 jones    | Alice Jones      |      82 |        1
 bitdiddl | Ben Bitdiddle    |      65 |        1
 mike     | Michael Dole     |      73 |        2
 alyssa   | Alyssa P. Hacker |      79 |        3
 bbrown   | Bob Brown        |     100 |        3
```

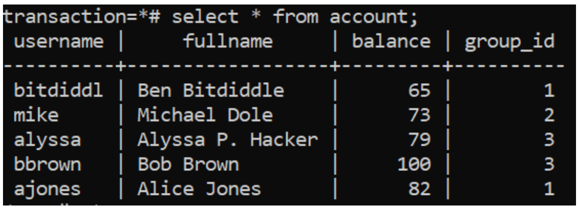
</td><td>

4. SELECT * FROM account;



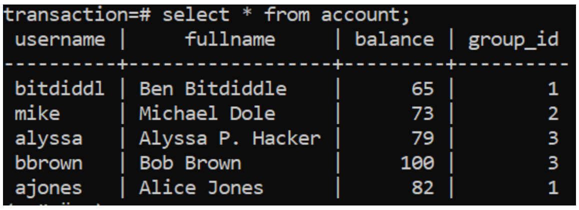
```
transaction=*# select * from account;
 username |     fullname     | balance | group_id
----------+------------------+---------+----------
 bitdiddl | Ben Bitdiddle    |      65 |        1
 mike     | Michael Dole     |      73 |        2
 alyssa   | Alyssa P. Hacker |      79 |        3
 bbrown   | Bob Brown        |     100 |        3
 ajones   | Alice Jones      |      82 |        1
```

</td></tr>
<tr><td colspan="2">Outputs are different, because in the second transaction we didn’t commit changes and the first user gets old data (dirty read).</td></tr>
<tr><td></td><td>5. COMMIT;</td></tr>
<tr><td>



```
transaction=*# select * from account;
 username |     fullname     | balance | group_id
----------+------------------+---------+----------
 bitdiddl | Ben Bitdiddle    |      65 |        1
 mike     | Michael Dole     |      73 |        2
 alyssa   | Alyssa P. Hacker |      79 |        3
 bbrown   | Bob Brown        |     100 |        3
 ajones   | Alice Jones      |      82 |        1
```

</td><td>



```
transaction=# select * from account;
 username |     fullname     | balance | group_id
----------+------------------+---------+----------
 bitdiddl | Ben Bitdiddle    |      65 |        1
 mike     | Michael Dole     |      73 |        2
 alyssa   | Alyssa P. Hacker |      79 |        3
 bbrown   | Bob Brown        |     100 |        3
 ajones   | Alice Jones      |      82 |        1
```

</td></tr>
<tr><td></td><td>6. BEGIN;</td></tr>
</table>

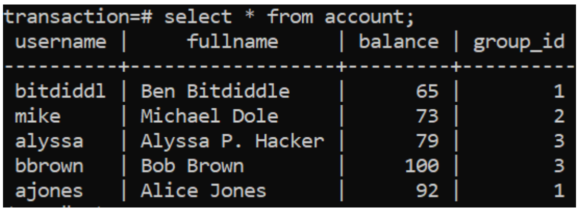
7. UPDATE account SET balance=balance+10 WHERE fullname='Alice Jones';

8. UPDATE account SET balance=balance+20 WHERE fullname='Alice Jones';

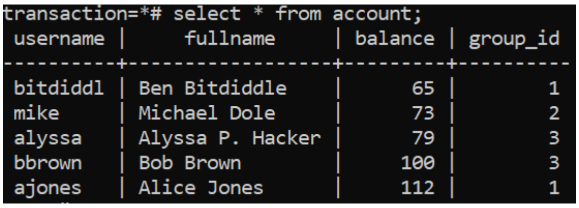
Waiting

In the second exercise, the first transaction puts a “lock” on modifying data inside the table and puts other users who want to commit in the queue (like Mutex). Once we commit changes in the first terminal, we become able to modify data in the second terminal. This works because Read committed isolation level.

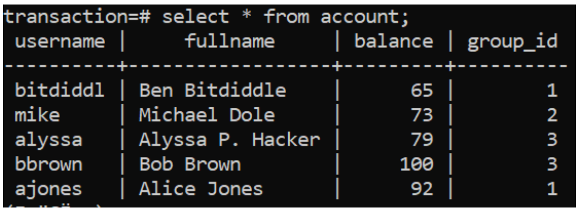
9. COMMIT;



```
transaction=# select * from account;
 username |      fullname      | balance | group_id
----------+-------------------+---------+----------
 bitdiddl | Ben Bitdiddle     |      65 |        1
 mike     | Michael Dole      |      73 |        2
 alyssa   | Alyssa P. Hacker  |      79 |        3
 bbrown   | Bob Brown         |     100 |        3
 ajones   | Alice Jones       |      92 |        1
```

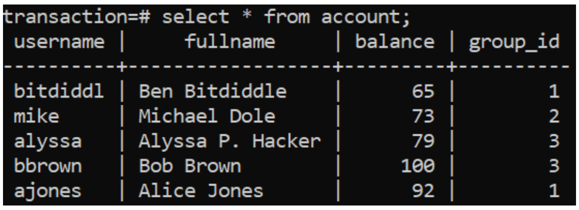


```
transaction=*# select * from account;
 username |      fullname      | balance | group_id
----------+-------------------+---------+----------
 bitdiddl | Ben Bitdiddle     |      65 |        1
 mike     | Michael Dole      |      73 |        2
 alyssa   | Alyssa P. Hacker  |      79 |        3
 bbrown   | Bob Brown         |     100 |        3
 ajones   | Alice Jones       |     112 |        1
```



```
transaction=# select * from account;
 username |      fullname      | balance | group_id
----------+-------------------+---------+----------
 bitdiddl | Ben Bitdiddle     |      65 |        1
 mike     | Michael Dole      |      73 |        2
 alyssa   | Alyssa P. Hacker  |      79 |        3
 bbrown   | Bob Brown         |     100 |        3
 ajones   | Alice Jones       |      92 |        1
```

10. ROLLBACK;

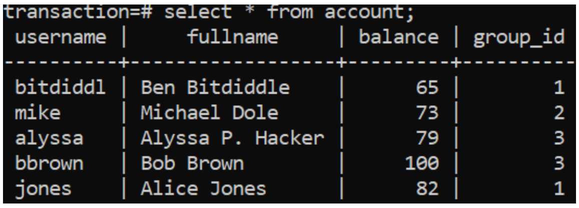


```
transaction=# select * from account;
 username |      fullname      | balance | group_id
----------+-------------------+---------+----------
 bitdiddl | Ben Bitdiddle     |      65 |        1
 mike     | Michael Dole      |      73 |        2
 alyssa   | Alyssa P. Hacker  |      79 |        3
 bbrown   | Bob Brown         |     100 |        3
 ajones   | Alice Jones       |      92 |        1
```

BEGIN;
SET transaction isolation level repeatable read;

<table>
<tr><th colspan="2">Read commited isolation level</th></tr>
<tr><th>Terminal 1</th><th>Terminal 2</th></tr>
<tr><td>

1. BEGIN TRANSACTION;
SELECT * FROM account;



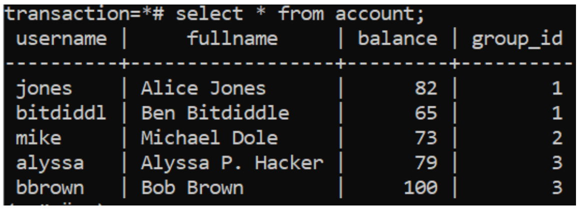
```
transaction=# select * from account;
 username |      fullname      | balance | group_id
----------+-------------------+---------+----------
 bitdiddl | Ben Bitdiddle     |      65 |        1
 mike     | Michael Dole      |      73 |        2
 alyssa   | Alyssa P. Hacker  |      79 |        3
 bbrown   | Bob Brown         |     100 |        3
 jones    | Alice Jones       |      82 |        1
```

</td><td>

2. BEGIN;
UPDATE account SET username='ajones' WHERE fullname='Alice Jones';

</td></tr>
<tr><td>

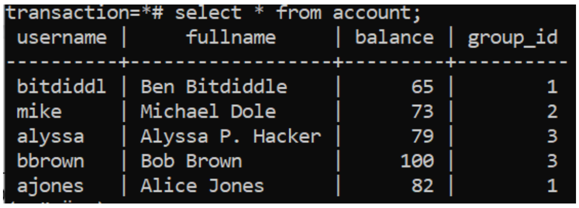
3. SELECT * FROM account;



```
transaction=*# select * from account;
 username |      fullname      | balance | group_id
----------+-------------------+---------+----------
 jones    | Alice Jones       |      82 |        1
 bitdiddl | Ben Bitdiddle     |      65 |        1
 mike     | Michael Dole      |      73 |        2
 alyssa   | Alyssa P. Hacker  |      79 |        3
 bbrown   | Bob Brown         |     100 |        3
```

</td><td>

4. SELECT * FROM account;



```
transaction=*# select * from account;
 username |      fullname      | balance | group_id
----------+-------------------+---------+----------
 bitdiddl | Ben Bitdiddle     |      65 |        1
 mike     | Michael Dole      |      73 |        2
 alyssa   | Alyssa P. Hacker  |      79 |        3
 bbrown   | Bob Brown         |     100 |        3
 ajones   | Alice Jones       |      82 |        1
```

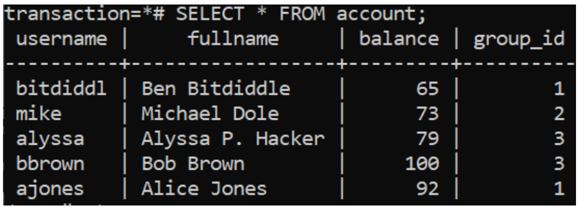
</td></tr>
<tr><td colspan="2">Outputs are different, because in the second transaction we didn’t commit changes and the first user gets old data (dirty read).</td></tr>
<tr><td></td><td>5. COMMIT;</td></tr>
<tr><td></td><td>6. BEGIN;</td></tr>
<tr><td>7. UPDATE account SET balance=balance+10 WHERE fullname='Alice Jones';</td><td>

8. UPDATE account SET balance=balance+20 WHERE fullname='Alice Jones';

Waiting

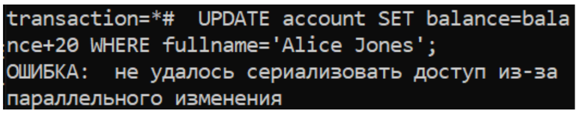
</td></tr>
<tr><td colspan="2">In the second exercise, the first transaction puts a “lock” on modifying data inside the table and puts other users who want to commit in the queue (like Mutex). Once we commit changes in the first terminal, we become able to modify data in the second terminal. This works because Read committed isolation level.</td></tr>
<tr><td>

9. COMMIT;



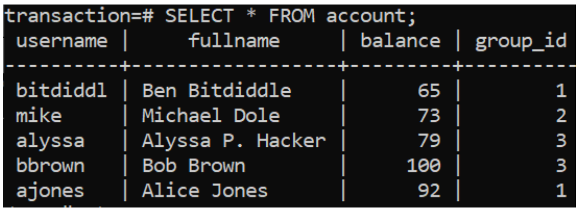
```
transaction=*# SELECT * FROM account;
 username |      fullname      | balance | group_id
----------+-------------------+---------+----------
 bitdiddl | Ben Bitdiddle     |      65 |        1
 mike     | Michael Dole      |      73 |        2
 alyssa   | Alyssa P. Hacker  |      79 |        3
 bbrown   | Bob Brown         |     100 |        3
 ajones   | Alice Jones       |      92 |        1
```

</td><td>



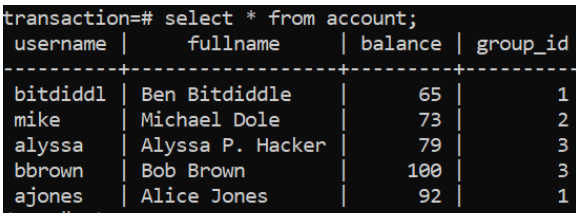
```
transaction=*#  UPDATE account SET balance=bala
nce+20 WHERE fullname='Alice Jones';
ОШИБКА:  не удалось сериализовать доступ из-за
параллельного изменения
```

</td></tr>
<tr><td colspan="2">After COMMIT in the first terminal we have ERROR in the second terminal, because we work in Read commited isolation level.</td></tr>
</table>



```
transaction=# SELECT * FROM account;
 username |      fullname      | balance | group_id
----------+-------------------+---------+----------
 bitdiddl | Ben Bitdiddle     |      65 |        1
 mike     | Michael Dole      |      73 |        2
 alyssa   | Alyssa P. Hacker  |      79 |        3
 bbrown   | Bob Brown         |     100 |        3
 ajones   | Alice Jones       |      92 |        1
```

10. ROLLBACK;



```
transaction=# select * from account;
 username |      fullname      | balance | group_id
----------+-------------------+---------+----------
 bitdiddl | Ben Bitdiddle     |      65 |        1
 mike     | Michael Dole      |      73 |        2
 alyssa   | Alyssa P. Hacker  |      79 |        3
 bbrown   | Bob Brown         |     100 |        3
 ajones   | Alice Jones       |      92 |        1
```