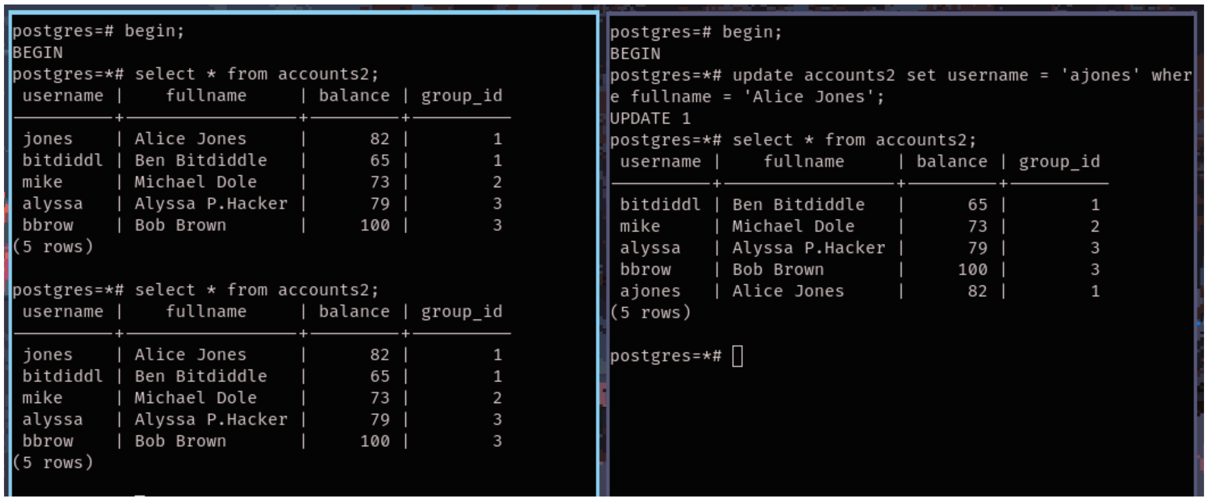
Task2 explanations

Read committed:

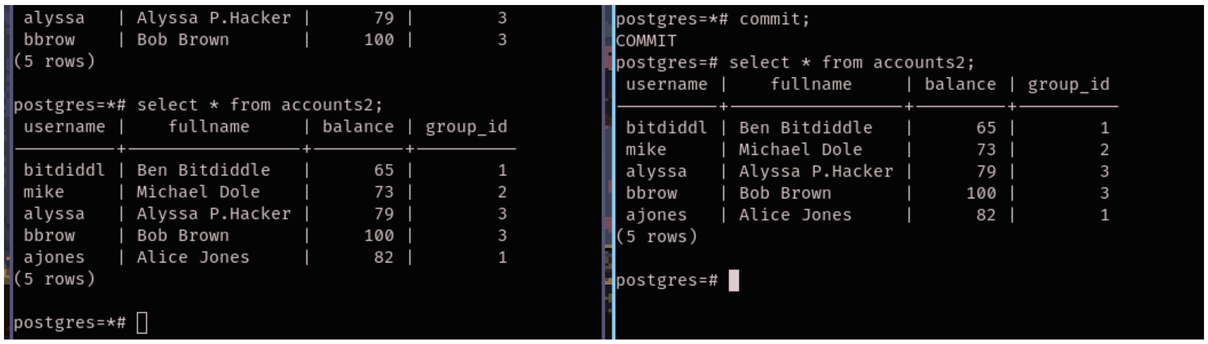
2.1.



```
postgres=# begin;
BEGIN
postgres=*# select * from accounts2;
 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
 bbrow    | Bob Brown       |     100 |        3
(5 rows)

postgres=*# select * from accounts2;
 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
 bbrow    | Bob Brown       |     100 |        3
(5 rows)
```

```
postgres=# begin;
BEGIN
postgres=*# update accounts2 set username = 'ajones' wher
e fullname = 'Alice Jones';
UPDATE 1
postgres=*# select * from accounts2;
 username |    fullname     | balance | group_id
----------+-----------------+---------+----------
 bitdiddl | Ben Bitdiddle   |      65 |        1
 mike     | Michael Dole    |      73 |        2
 alyssa   | Alyssa P.Hacker |      79 |        3
 bbrow    | Bob Brown       |     100 |        3
 ajones   | Alice Jones     |      82 |        1
(5 rows)

postgres=*#
```

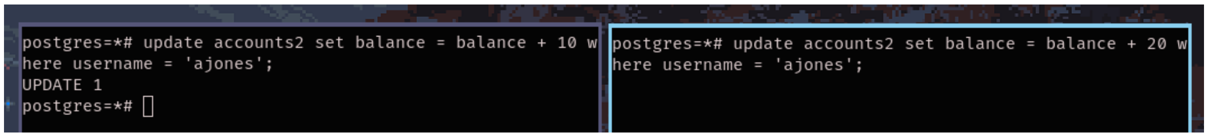
The output of the terminals is different because the transaction in the second terminal is not committed yet.



```
 alyssa   | Alyssa P.Hacker |      79 |        3
 bbrow    | Bob Brown       |     100 |        3
(5 rows)

postgres=*# select * from accounts2;
 username |    fullname     | balance | group_id
----------+-----------------+---------+----------
 bitdiddl | Ben Bitdiddle   |      65 |        1
 mike     | Michael Dole    |      73 |        2
 alyssa   | Alyssa P.Hacker |      79 |        3
 bbrow    | Bob Brown       |     100 |        3
 ajones   | Alice Jones     |      82 |        1
(5 rows)

postgres=*#
```

```
postgres=*# commit;
COMMIT
postgres=# select * from accounts2;
 username |    fullname     | balance | group_id
----------+-----------------+---------+----------
 bitdiddl | Ben Bitdiddle   |      65 |        1
 mike     | Michael Dole    |      73 |        2
 alyssa   | Alyssa P.Hacker |      79 |        3
 bbrow    | Bob Brown       |     100 |        3
 ajones   | Alice Jones     |      82 |        1
(5 rows)

postgres=#
```

After the second transaction has been committed, both terminals show the same output.

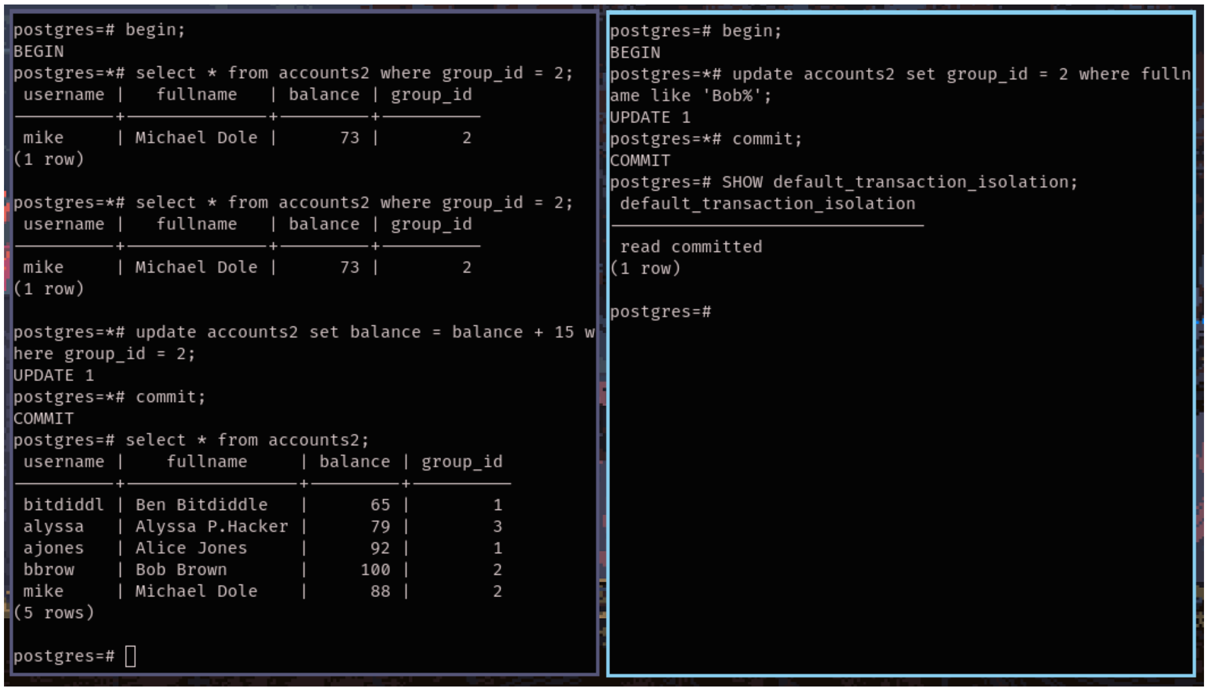


```
postgres=*# update accounts2 set balance = balance + 10 w
here username = 'ajones';
UPDATE 1
postgres=*# 
```

```
postgres=*# update accounts2 set balance = balance + 20 w
here username = 'ajones';
```

The second terminal waits for the first transaction to commit the changes, because I may update an unwanted value.

2.2

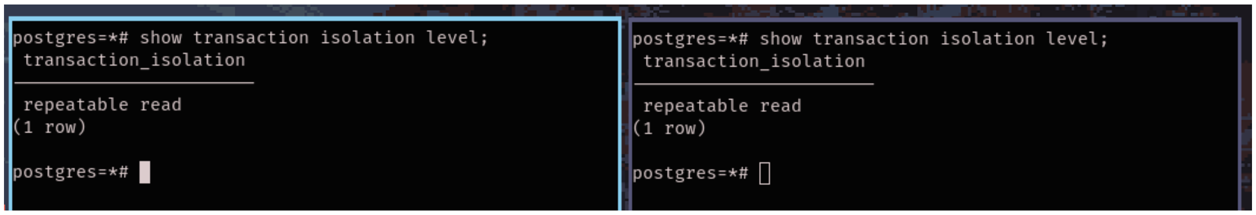


```
postgres=# begin;
BEGIN
postgres=*# select * from accounts2 where group_id = 2;
 username |   fullname   | balance | group_id
----------+--------------+---------+----------
 mike     | Michael Dole |      73 |        2
(1 row)

postgres=*# select * from accounts2 where group_id = 2;
 username |   fullname   | balance | group_id
----------+--------------+---------+----------
 mike     | Michael Dole |      73 |        2
(1 row)

postgres=*# update accounts2 set balance = balance + 15 w
here group_id = 2;
UPDATE 1
postgres=*# commit;
COMMIT
postgres=# select * from accounts2;
 username |    fullname     | balance | group_id
----------+-----------------+---------+----------
 bitdiddl | Ben Bitdiddle   |      65 |        1
 alyssa   | Alyssa P.Hacker |      79 |        3
 ajones   | Alice Jones     |      92 |        1
 bbrow    | Bob Brown       |     100 |        2
 mike     | Michael Dole    |      88 |        2
(5 rows)

postgres=# 
```

```
postgres=# begin;
BEGIN
postgres=*# update accounts2 set group_id = 2 where fulln
ame like 'Bob%';
UPDATE 1
postgres=*# commit;
COMMIT
postgres=# SHOW default_transaction_isolation;
 default_transaction_isolation
-------------------------------
 read committed
(1 row)

postgres=#
```

The first transaction does not see uncommitted update statement because of the isolation level, that is why it only updated the balance of Michael Dole.

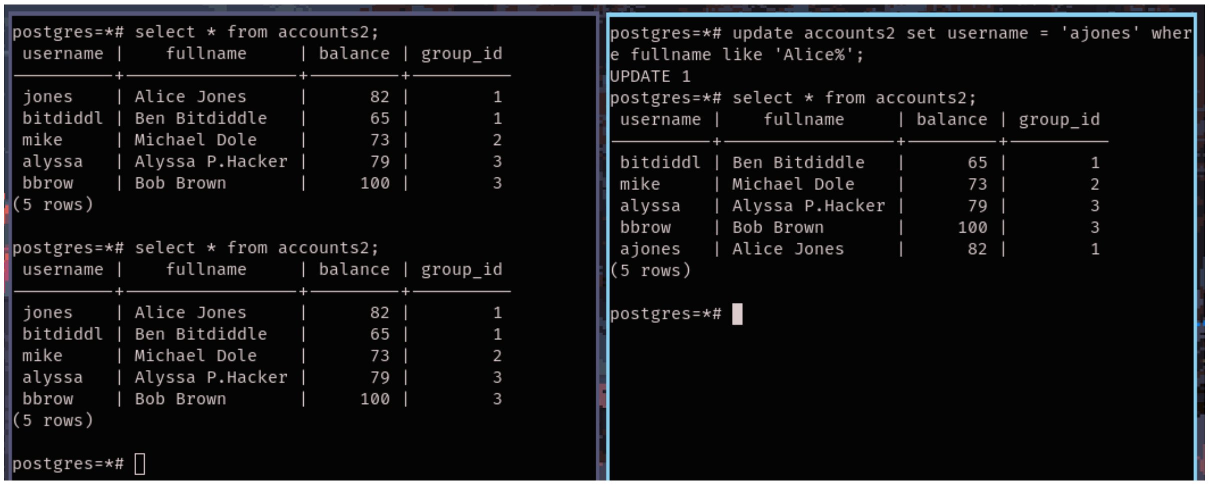
Repeatable read:



```
postgres=*# show transaction isolation level;
 transaction_isolation
-----------------------
 repeatable read
(1 row)

postgres=*# 
```

```
postgres=*# show transaction isolation level;
 transaction_isolation
-----------------------
 repeatable read
(1 row)

postgres=*# 
```

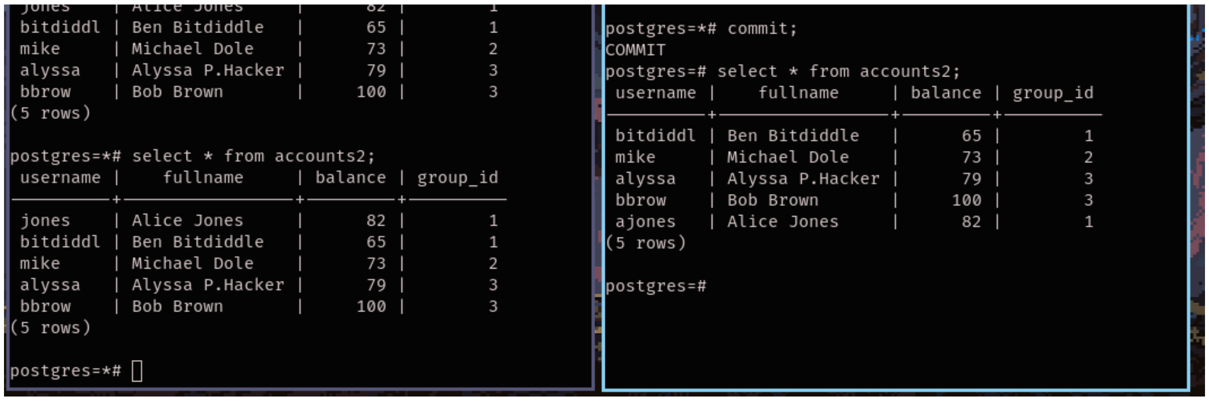
2.1



```
postgres=*# select * from accounts2;
 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
 bbrow    | Bob Brown       |     100 |        3
(5 rows)

postgres=*# select * from accounts2;
 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
 bbrow    | Bob Brown       |     100 |        3
(5 rows)

postgres=*#
```

```
postgres=*# update accounts2 set username = 'ajones' wher
e fullname like 'Alice%';
UPDATE 1
postgres=*# select * from accounts2;
 username |    fullname     | balance | group_id
----------+-----------------+---------+----------
 bitdiddl | Ben Bitdiddle   |      65 |        1
 mike     | Michael Dole    |      73 |        2
 alyssa   | Alyssa P.Hacker |      79 |        3
 bbrow    | Bob Brown       |     100 |        3
 ajones   | Alice Jones     |      82 |        1
(5 rows)

postgres=*#
```

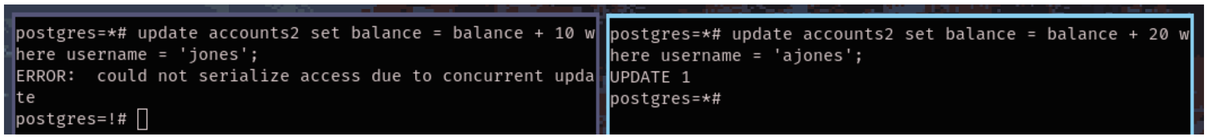
They still show different results because repeatable read cannot read uncommitted operations.



```
 jones    | Alice Jones     |      82 |        1
 bitdiddl | Ben Bitdiddle   |      65 |        1
 mike     | Michael Dole    |      73 |        2
 alyssa   | Alyssa P.Hacker |      79 |        3
 bbrow    | Bob Brown       |     100 |        3
(5 rows)

postgres=*# select * from accounts2;
 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
 bbrow    | Bob Brown       |     100 |        3
(5 rows)

postgres=*#
```

```
postgres=*# commit;
COMMIT
postgres=# select * from accounts2;
 username |    fullname     | balance | group_id
----------+-----------------+---------+----------
 bitdiddl | Ben Bitdiddle   |      65 |        1
 mike     | Michael Dole    |      73 |        2
 alyssa   | Alyssa P.Hacker |      79 |        3
 bbrow    | Bob Brown       |     100 |        3
 ajones   | Alice Jones     |      82 |        1
(5 rows)

postgres=#
```

After commit, first terminal still shows old data because current transaction had not change any values, second terminal shows updated values.

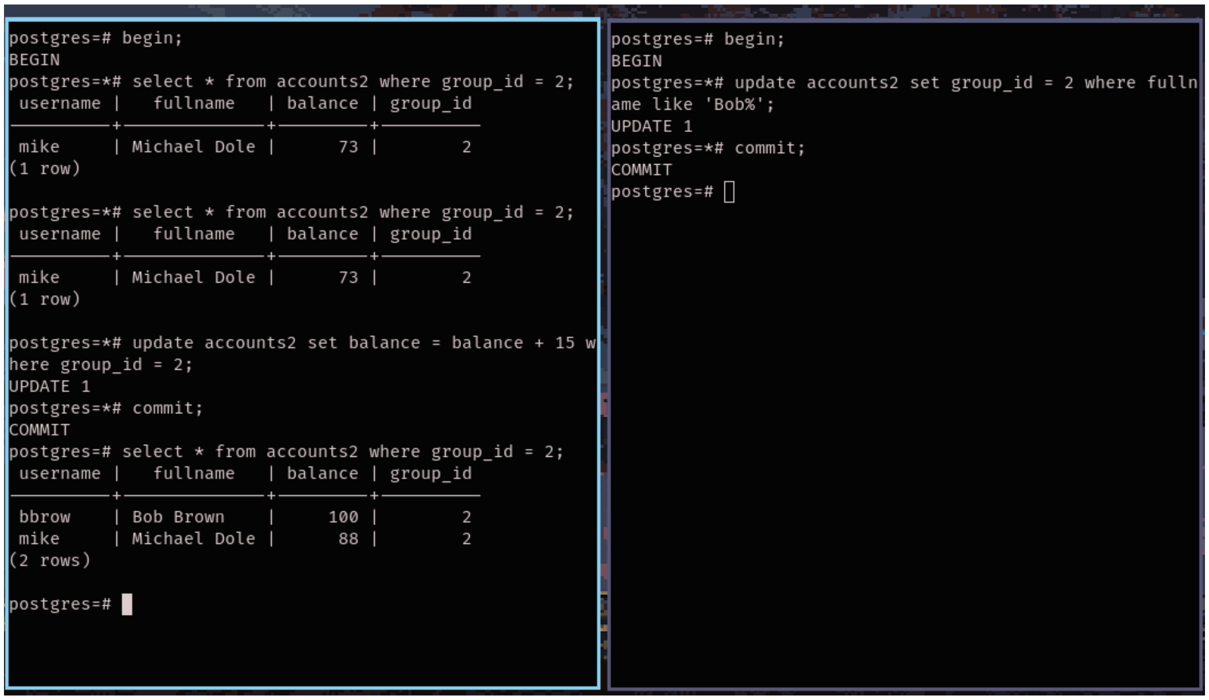


```
postgres=*# update accounts2 set balance = balance + 10 w
here username = 'jones';
ERROR:  could not serialize access due to concurrent upda
te
postgres=!#
```

```
postgres=*# update accounts2 set balance = balance + 20 w
here username = 'ajones';
UPDATE 1
postgres=*#
```

There is an error in the first terminal because both transaction are trying to update the same cell.

2.2



```
postgres=# begin;
BEGIN
postgres=*# select * from accounts2 where group_id = 2;
 username |   fullname   | balance | group_id
----------+--------------+---------+----------
 mike     | Michael Dole |      73 |        2
(1 row)

postgres=*# select * from accounts2 where group_id = 2;
 username |   fullname   | balance | group_id
----------+--------------+---------+----------
 mike     | Michael Dole |      73 |        2
(1 row)

postgres=*# update accounts2 set balance = balance + 15 w
here group_id = 2;
UPDATE 1
postgres=*# commit;
COMMIT
postgres=# select * from accounts2 where group_id = 2;
 username |   fullname   | balance | group_id
----------+--------------+---------+----------
 bbrow    | Bob Brown    |     100 |        2
 mike     | Michael Dole |      88 |        2
(2 rows)

postgres=# 
```

```
postgres=# begin;
BEGIN
postgres=*# update accounts2 set group_id = 2 where fulln
ame like 'Bob%';
UPDATE 1
postgres=*# commit;
COMMIT
postgres=# 
```

The first transaction does not see uncommitted update statement because of the isolation level, that is why it only updated the balance of Michael Dole.