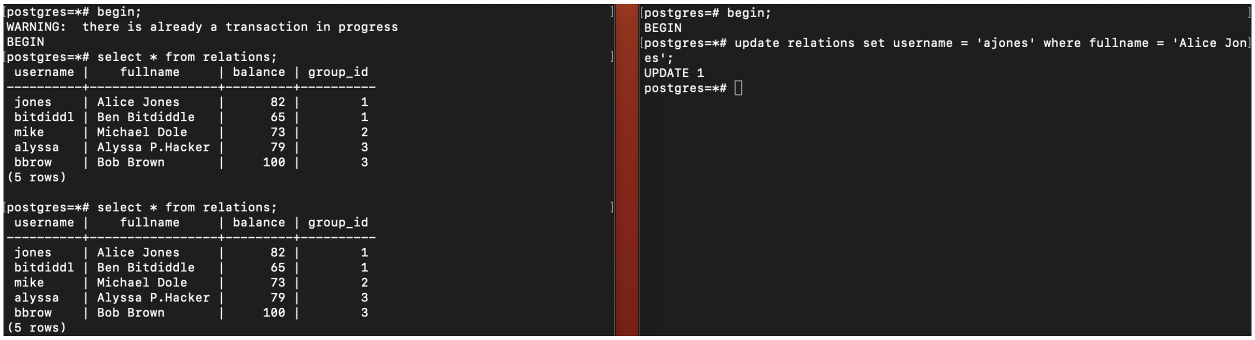
Task 2

Read committed

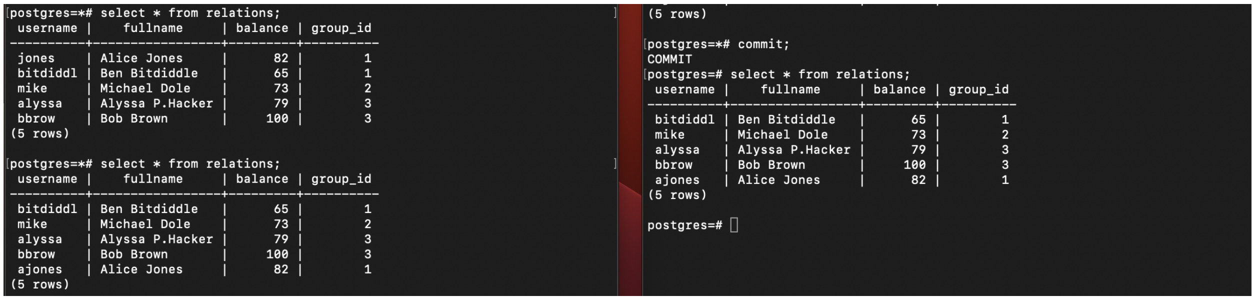
2.1



```
postgres=*# begin;
WARNING:  there is already a transaction in progress
BEGIN
postgres=*# select * from relations;
 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 relations;
 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 relations set username = 'ajones' where fullname = 'Alice Jon
es';
UPDATE 1
postgres=*# 
```

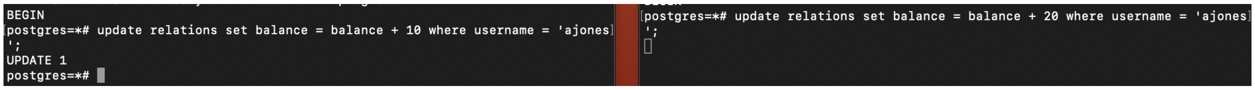
Because the transaction in the second terminal has not yet been committed, the output from the terminals is different.



```
postgres=*# select * from relations;
 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 relations;
 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)
```

```
(5 rows)

postgres=*# commit;
COMMIT
postgres=# select * from relations;
 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=# 
```

Once the second transaction has been committed, the same output is displayed on both terminals.

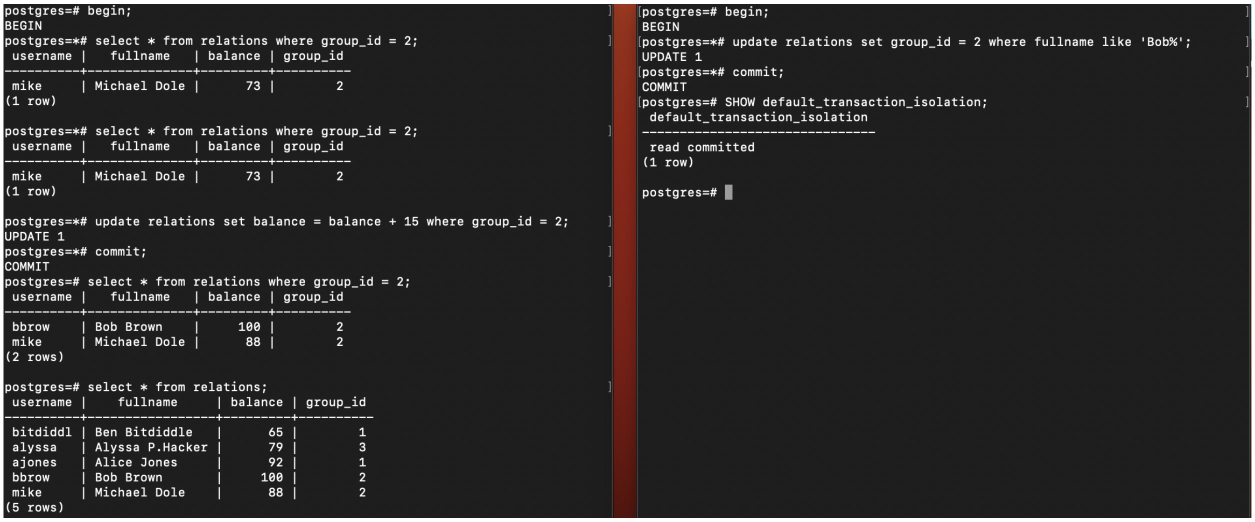


```
BEGIN
postgres=*# update relations set balance = balance + 10 where username = 'ajones
';
UPDATE 1
postgres=*# 
```

```
postgres=*# update relations set balance = balance + 20 where username = 'ajones
';
```

Because I could alter an undesirable value, the second terminal waits for the first transaction to commit the modifications.

2.2

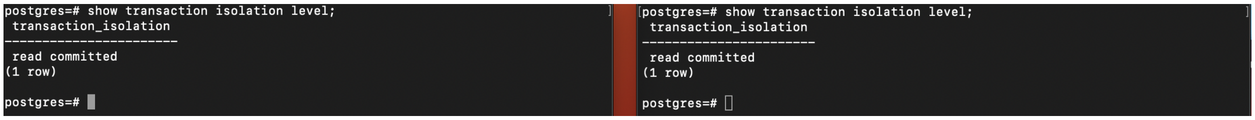


```
postgres=# begin;
BEGIN
postgres=*# select * from relations where group_id = 2;
 username |   fullname   | balance | group_id
----------+--------------+---------+----------
 mike     | Michael Dole |      73 |        2
(1 row)

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

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

postgres=# select * from relations;
 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=# begin;
BEGIN
postgres=*# update relations set group_id = 2 where fullname like 'Bob%';
UPDATE 1
postgres=*# commit;
COMMIT
postgres=# SHOW default_transaction_isolation;
 default_transaction_isolation
-------------------------------
 read committed
(1 row)

postgres=#
```

Because of the isolation level, the first transaction does not notice uncommitted update statements, and as a result, it solely modified Michael Dole's balance.

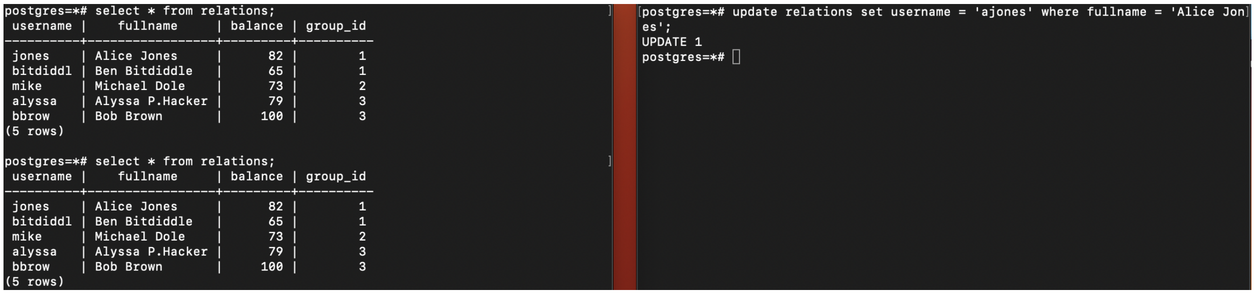
Repeatable read



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

postgres=#
```

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

postgres=#
```

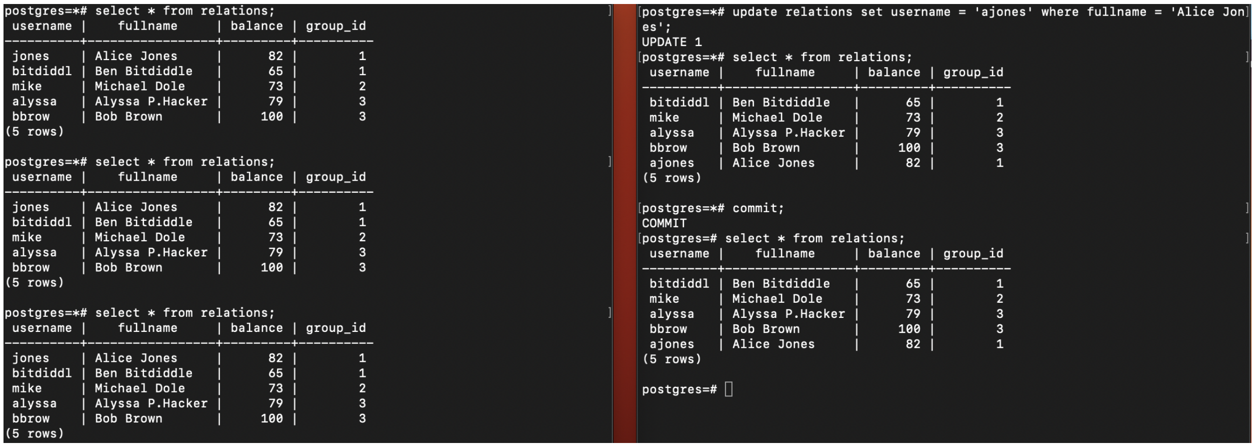
2.1



```
postgres=*# select * from relations;
 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 relations;
 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=*# update relations set username = 'ajones' where fullname = 'Alice Jon
es';
UPDATE 1
postgres=*# 
```

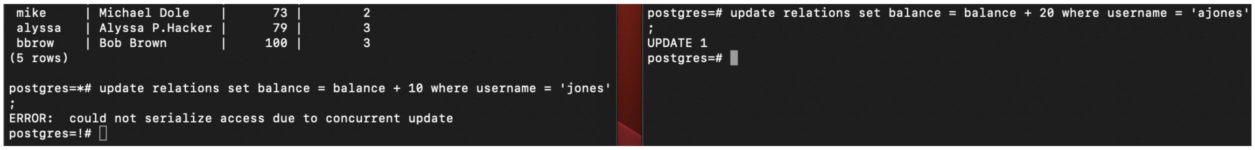
As repeated read is unable to read uncommitted actions, they continue to produce disparate outputs.



```
postgres=*# select * from relations;
 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 relations;
 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 relations;
 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=*# update relations set username = 'ajones' where fullname = 'Alice Jon
es';
UPDATE 1
postgres=*# select * from relations;
 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=*# commit;
COMMIT
postgres=# select * from relations;
 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 first terminal continues to display outdated data after commit because the current transaction did not alter any values, whereas the second terminal displays updated values.

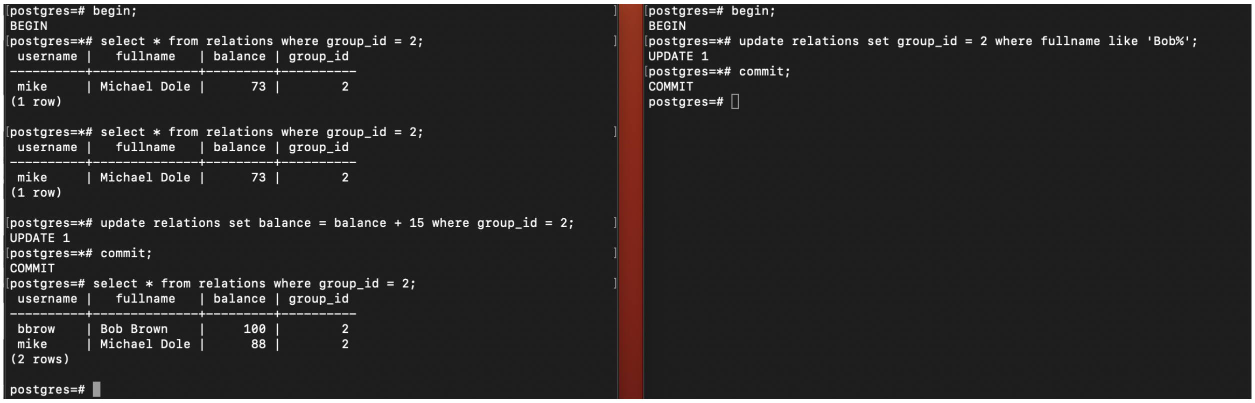


```
 mike     | Michael Dole    |      73 |        2
 alyssa   | Alyssa P.Hacker |      79 |        3
 bbrow    | Bob Brown       |     100 |        3
(5 rows)

postgres=*# update relations set balance = balance + 10 where username = 'jones'
;
ERROR:  could not serialize access due to concurrent update
postgres=!#
```

```
postgres=# update relations set balance = balance + 20 where username = 'ajones'
;
UPDATE 1
postgres=#
```

Because both transactions are attempting to update the same cell, there is a problem in the first terminal.

2.2



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

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

postgres=*# update relations set balance = balance + 15 where group_id = 2;
UPDATE 1
postgres=*# commit;
COMMIT
postgres=# select * from relations 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 relations set group_id = 2 where fullname like 'Bob%';
UPDATE 1
postgres=*# commit;
COMMIT
postgres=# 
```