Assignment 5

Jiaxi Zhang

April 20, 2025

Author: [Jiaxi Zhang, jz5751]

Assignment: HW05

Problem 2.3

The output in the blue terminal and in the red terminal are different.

The output for the blue terminal is:

```
+------------+--------------------+---------+
| username   | fullname           | balance |
+------------+--------------------+---------+
| alyssa     | Alyssa P. Hacker   |      79 |
| bitdiddle  | Ben Bitdiddle      |      65 |
| jones      | Alice Jones        |      82 |
| mike       | Michael Dole       |      73 |
+------------+--------------------+---------+
4 rows in set (0.0039 sec)
```

The output for the red terminal is:

```
+------------+--------------------+---------+
| username   | fullname           | balance |
+------------+--------------------+---------+
| alyssa     | Alyssa P. Hacker   |      79 |
| bitdiddle  | Ben Bitdiddle      |      65 |
| chuck      | Charles Robinson   |      55 |
| jones      | Alice Jones        |      82 |
| mike       | Michael Dole       |      73 |
+------------+--------------------+---------+
5 rows in set (0.0019 sec)
```

The reason for the difference is that the red terminal has not committed the transaction yet, so the changes made in this transaction are not visible to blue terminal, especially when it is also in a transaction. Both of them are currently locked.

Problem 2.5

No. The output of current blue terminal is:

```
+-----------+--------------------+---------+
| username  | fullname           | balance |
+-----------+--------------------+---------+
| alyssa    | Alyssa P. Hacker   |      79 |
| bitdiddle | Ben Bitdiddle      |      65 |
| jones     | Alice Jones        |      82 |
| mike      | Michael Dole       |      73 |
+-----------+--------------------+---------+
4 rows in set (0.0072 sec)
```

It does not include the changes made in the red terminal even after commit, because the blue terminal is in a transaction and the changes made in the red terminal are not visible to the locked blue terminal until the blue-terminal transaction is committed.

Problem 2.6

The output of the blue terminal is:

```
+-----------+--------------------+---------+
| username  | fullname           | balance |
+-----------+--------------------+---------+
| alyssa    | Alyssa P. Hacker   |      79 |
| bitdiddle | Ben Bitdiddle      |      65 |
| chuck     | Charles Robinson   |      55 |
| jones     | Alice Jones        |      82 |
| mike      | Michael Dole       |      73 |
+-----------+--------------------+---------+
5 rows in set (0.0057 sec)
```

It is different from the output of Problem 2.5 because the blue terminal has committed the transaction, so the changes made in the red terminal are visible to the blue terminal since the lock is released.

Problem 2.9

The second update in the red terminal is stuck because the blue terminal is in a transaction, holding a lock on the row of "mike", which is not released until the blue terminal commits or rolls back the transaction. Therefore, the second update in the red terminal, which is also trying to write to the same row, is blocked and cannot proceed until the blue terminal releases the lock.

Problem 2.11

The UPDATE in the red terminal is successfully executed.

```
> UPDATE accounts SET balance = balance - 10 WHERE username = 'mike';
Query OK, 1 row affected (5.9192 sec)

Rows matched: 1  Changed: 1  Warnings: 0
```

Problem 2.12

```
The output is:
> SELECT * FROM accounts WHERE username = 'mike';
+----------+---------------+---------+
| username | fullname      | balance |
+----------+---------------+---------+
| mike     | Michael Dole  |      63 |
+----------+---------------+---------+
1 row in set (0.0019 sec)
```

The output is the same for both terminals since the red terminal has committed the transaction. It is -10 from the original balance of 73, which means the update in the red terminal is effective while the blue terminal rolled back successfully.

Problem 2.14

The result has not changed in the blue terminal. The output of the blue terminal before the transaction started is:

```
> SELECT username, balance FROM accounts;
+----------+---------+
| username | balance |
+----------+---------+
```

```
| alyssa     |       79 |
| bitdiddle  |       65 |
| chuck      |       55 |
| jones      |       82 |
| mike       |       63 |
+------------+----------+
5 rows in set (0.0011 sec)
```

The output of the blue terminal after the transaction started is:

```
> SELECT username, balance FROM accounts;
+------------+----------+
| username   | balance  |
+------------+----------+
| alyssa     |       79 |
| bitdiddle  |       65 |
| chuck      |       55 |
| jones      |       82 |
| mike       |       63 |
+------------+----------+
5 rows in set (0.0008 sec)
```

Since the red terminal has not committed the transaction yet, the changes made are not visible to the blue terminal due to atomicity.

Problem 2.15

The changes made in the red terminal are visible to the blue terminal when the red terminal commits the transaction. Blue Output after

```
> UPDATE accounts SET balance = balance + 15 WHERE username = 'alyssa';
```

from the red terminal.

```
> SELECT username, balance FROM accounts;
+------------+----------+
| username   | balance  |
+------------+----------+
| alyssa     |       79 |
| bitdiddle  |       65 |
| chuck      |       55 |
| jones      |       82 |
```

```
| mike        |       63 |
+-------------+----------+
5 rows in set (0.0048 sec)
```

Blue Output after

```
> COMMIT;
```

from the red terminal:

```
> SELECT username, balance FROM accounts;
+-------------+----------+
| username    | balance  |
+-------------+----------+
| alyssa      |       94 |
| bitdiddle   |       50 |
| chuck       |       55 |
| jones       |       82 |
| mike        |       63 |
+-------------+----------+
5 rows in set (0.0036 sec)
```

So the changes made in the red terminal are visible to the blue terminal after the red terminal commits the transaction. It is because the red terminal has committed the transaction and released the lock on the rows it was holding (in this case, the row of "alyssa" and "bitdiddle"). The DBMS follows the ACID properties of transactions, which ensure that only when a transaction is committed, its changes are made visible to other transactions.