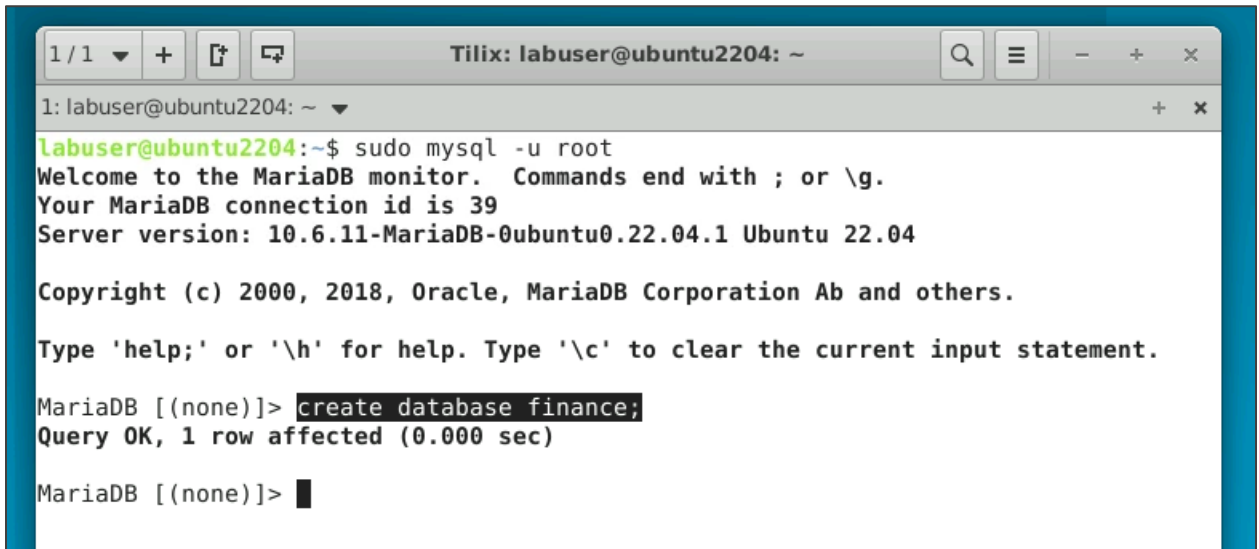


- 1.2 Create a new database named **finance**:
create database finance;

A terminal window titled 'Tilix: labuser@ubuntu2204: ~' showing a user running 'sudo mysql -u root'. The terminal displays the MariaDB monitor welcome message, including the connection ID (39) and server version (10.6.11-MariaDB-0ubuntu0.22.04.1). The user enters 'create database finance;' and receives the response 'Query OK, 1 row affected (0.000 sec)'.

```
1/1 ▼ + [ ] [ ] Tilix: labuser@ubuntu2204: ~
1: labuser@ubuntu2204: ~ ▼
labuser@ubuntu2204:~$ sudo mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 39
Server version: 10.6.11-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

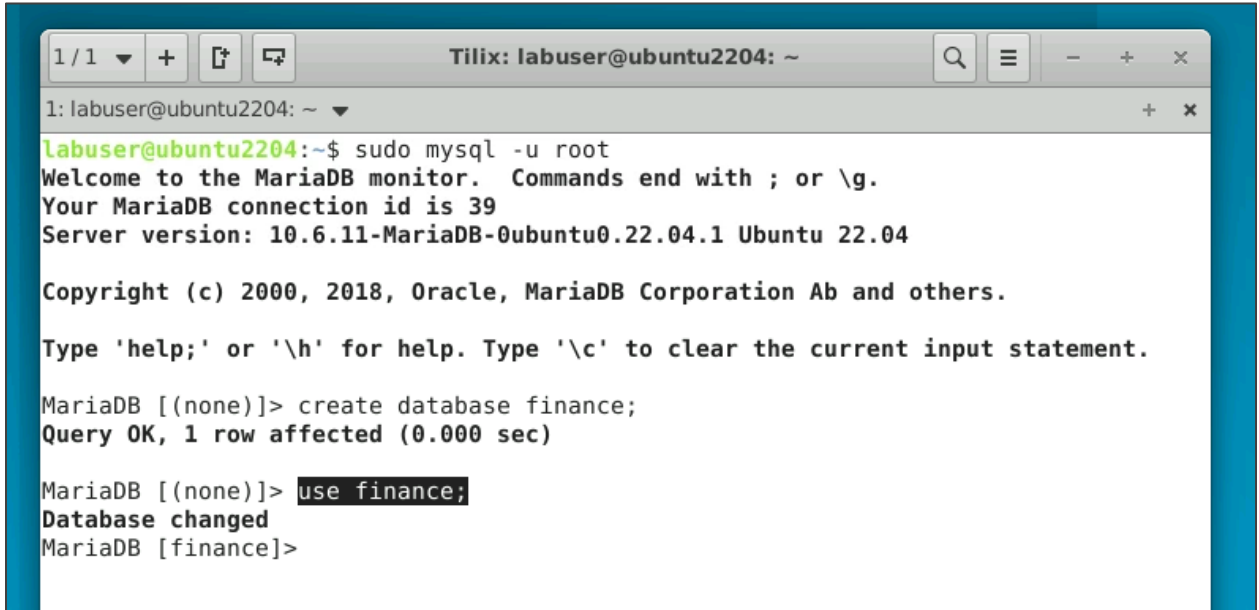
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database finance;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> █
```

- 1.3 Select the **finance** database:
use finance;

A terminal window titled 'Tilix: labuser@ubuntu2204: ~' showing the same initial steps as the previous screenshot. After creating the 'finance' database, the user enters 'use finance;' and receives the response 'Database changed'.

```
1/1 ▼ + [ ] [ ] Tilix: labuser@ubuntu2204: ~
1: labuser@ubuntu2204: ~ ▼
labuser@ubuntu2204:~$ sudo mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 39
Server version: 10.6.11-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

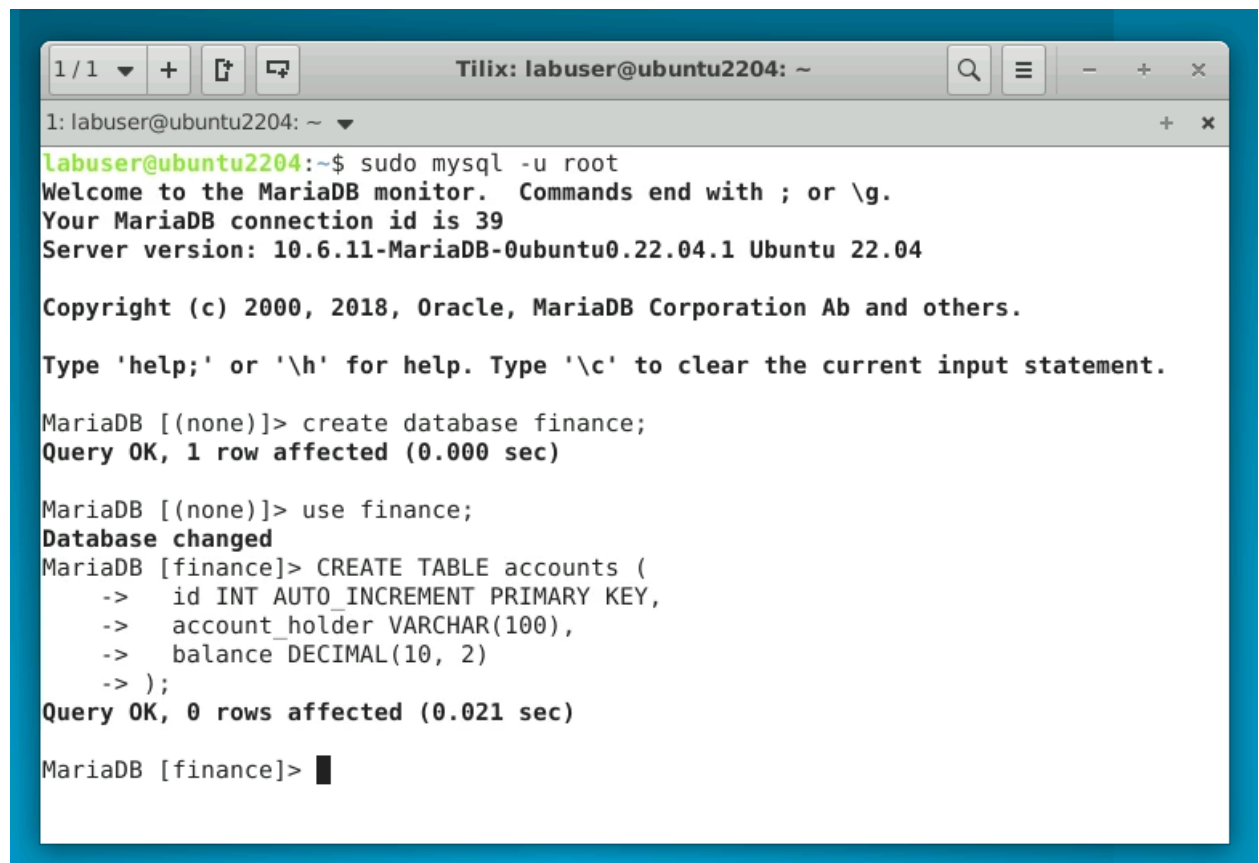
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database finance;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> use finance;
Database changed
MariaDB [finance]>
```

1.4 Create an **accounts** table with relevant fields:

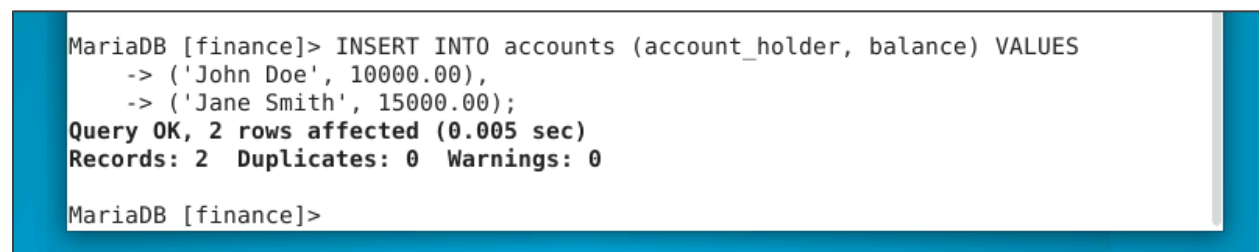
```
CREATE TABLE accounts (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  account_holder VARCHAR(100),  
  balance DECIMAL(10, 2)  
);
```

A screenshot of a terminal window titled 'Tilix: labuser@ubuntu2204: ~'. The terminal shows the user running 'sudo mysql -u root' to access the MariaDB monitor. The monitor displays a welcome message and the server version '10.6.11-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04'. The user then enters 'create database finance;', which is successful. Next, they enter 'use finance;', and the database changes to 'finance'. Finally, they enter the 'CREATE TABLE accounts' command with the specified fields: 'id INT AUTO_INCREMENT PRIMARY KEY', 'account_holder VARCHAR(100)', and 'balance DECIMAL(10, 2)'. This command is also successful. The terminal ends with the prompt 'MariaDB [finance]>' and a cursor.

```
1/1 ▼ + [T] [C] Tilix: labuser@ubuntu2204: ~  
1: labuser@ubuntu2204: ~ ▼  
labuser@ubuntu2204:~$ sudo mysql -u root  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 39  
Server version: 10.6.11-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]> create database finance;  
Query OK, 1 row affected (0.000 sec)  
  
MariaDB [(none)]> use finance;  
Database changed  
MariaDB [finance]> CREATE TABLE accounts (  
  -> id INT AUTO_INCREMENT PRIMARY KEY,  
  -> account_holder VARCHAR(100),  
  -> balance DECIMAL(10, 2)  
  -> );  
Query OK, 0 rows affected (0.021 sec)  
  
MariaDB [finance]> █
```

1.5 Insert initial data into the **accounts** table:

```
INSERT INTO accounts (account_holder, balance) VALUES  
('John Doe', 10000.00),  
('Jane Smith', 15000.00);
```

A screenshot of a terminal window showing the execution of an 'INSERT INTO' statement. The user enters the command to insert two rows into the 'accounts' table: one for 'John Doe' with a balance of 10000.00, and another for 'Jane Smith' with a balance of 15000.00. The terminal shows that the query was successful, affecting 2 rows in 0.005 seconds, with no duplicates or warnings. The prompt 'MariaDB [finance]>' is visible at the bottom.

```
MariaDB [finance]> INSERT INTO accounts (account_holder, balance) VALUES  
  -> ('John Doe', 10000.00),  
  -> ('Jane Smith', 15000.00);  
Query OK, 2 rows affected (0.005 sec)  
Records: 2  Duplicates: 0  Warnings: 0  
  
MariaDB [finance]>
```

Step 2: Demonstrate the TCL commands

2.1 Begin a new transaction using the following command:

START TRANSACTION;

```
MariaDB [finance]> INSERT INTO accounts (account_holder, balance) VALUES
-> ('John Doe', 10000.00),
-> ('Jane Smith', 15000.00);
Query OK, 2 rows affected (0.005 sec)
Records: 2 Duplicates: 0 Warnings: 0

MariaDB [finance]> START TRANSACTION;
Query OK, 0 rows affected (0.000 sec)

MariaDB [finance]>
```

2.2 Simulate transfer of funds between two accounts:

UPDATE accounts SET balance = balance - 1000 WHERE account_holder = 'John Doe';

UPDATE accounts SET balance = balance + 1000 WHERE account_holder = 'Jane Smith';

```
1/1 + [ ] [ ] Tilix: labuser@ubuntu2204: ~
1: labuser@ubuntu2204: ~

MariaDB [finance]> START TRANSACTION;
Query OK, 0 rows affected (0.000 sec)

MariaDB [finance]> UPDATE accounts SET balance = balance - 1000 WHERE account_holder = 'John Doe';
Query OK, 1 row affected (0.004 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [finance]> UPDATE accounts SET balance = balance + 1000 WHERE account_holder = 'Jane Smith';
Query OK, 1 row affected (0.000 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [finance]> SELECT * from accounts;
+----+-----+-----+
| id | account_holder | balance |
+----+-----+-----+
| 1 | John Doe      | 9000.00 |
| 2 | Jane Smith    | 16000.00 |
+----+-----+-----+
2 rows in set (0.000 sec)

MariaDB [finance]> █
```

2.3 Commit the transaction to make the changes permanent:

COMMIT;

```

1/1  +  [?] [?]  Tilix: labuser@ubuntu2204: ~  [Q] [≡] - + ×
1: labuser@ubuntu2204: ~  + ×

MariaDB [finance]> UPDATE accounts SET balance = balance - 1000 WHERE account_holder = 'John Doe';
Query OK, 1 row affected (0.004 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [finance]> UPDATE accounts SET balance = balance + 1000 WHERE account_holder = 'Jane Smith';
Query OK, 1 row affected (0.000 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [finance]> SELECT * from accounts;
+-----+-----+-----+
| id | account_holder | balance |
+-----+-----+-----+
| 1 | John Doe      | 9000.00 |
| 2 | Jane Smith    | 16000.00 |
+-----+-----+-----+
2 rows in set (0.000 sec)

MariaDB [finance]> COMMIT;
Query OK, 0 rows affected (0.003 sec)

MariaDB [finance]> 

```

2.4 Begin another transaction using the following command:

START TRANSACTION;

```

1/1  +  [?] [?]  Tilix: labuser@ubuntu2204: ~  [Q] [≡] - + ×
1: labuser@ubuntu2204: ~  + ×

Query OK, 1 row affected (0.004 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [finance]> UPDATE accounts SET balance = balance + 1000 WHERE account_holder = 'Jane Smith';
Query OK, 1 row affected (0.000 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [finance]> SELECT * from accounts;
+-----+-----+-----+
| id | account_holder | balance |
+-----+-----+-----+
| 1 | John Doe      | 9000.00 |
| 2 | Jane Smith    | 16000.00 |
+-----+-----+-----+
2 rows in set (0.000 sec)

MariaDB [finance]> COMMIT;
Query OK, 0 rows affected (0.003 sec)

MariaDB [finance]> START TRANSACTION;
Query OK, 0 rows affected (0.000 sec)

MariaDB [finance]> 

```

2.5 Attempt another transfer, but with an error (e.g., incorrect account holder name):

UPDATE accounts SET balance = balance - 500 WHERE account_holder = 'John Doe';

UPDATE accounts SET balance = balance + 500 WHERE account_holder = 'Jane Doe'; --

Incorrect name

```

1/1 + [ ] [ ] Tilix: labuser@ubuntu2204: ~
1: labuser@ubuntu2204: ~
| id | account_holder | balance |
+----+-----+-----+
| 1 | John Doe      | 9000.00 |
| 2 | Jane Smith    | 16000.00 |
+----+-----+-----+
2 rows in set (0.000 sec)

MariaDB [finance]> COMMIT;
Query OK, 0 rows affected (0.003 sec)

MariaDB [finance]> START TRANSACTION;
Query OK, 0 rows affected (0.000 sec)

MariaDB [finance]> UPDATE accounts SET balance = balance - 500 WHERE account_holder = 'John Doe';
Query OK, 1 row affected (0.000 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [finance]> UPDATE accounts SET balance = balance + 500 WHERE account_holder = 'Jane Doe'; -- Incorrect name
Query OK, 0 rows affected (0.000 sec)
Rows matched: 0  Changed: 0  Warnings: 0

MariaDB [finance]>

```

2.6 Rollback the transaction due to the error:

ROLLBACK;

```

MariaDB [finance]> UPDATE accounts SET balance = balance - 500 WHERE account_holder = 'John Doe';
Query OK, 1 row affected (0.000 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [finance]> UPDATE accounts SET balance = balance + 500 WHERE account_holder = 'Jane Doe'; -- Incorrect name
Query OK, 0 rows affected (0.000 sec)
Rows matched: 0  Changed: 0  Warnings: 0

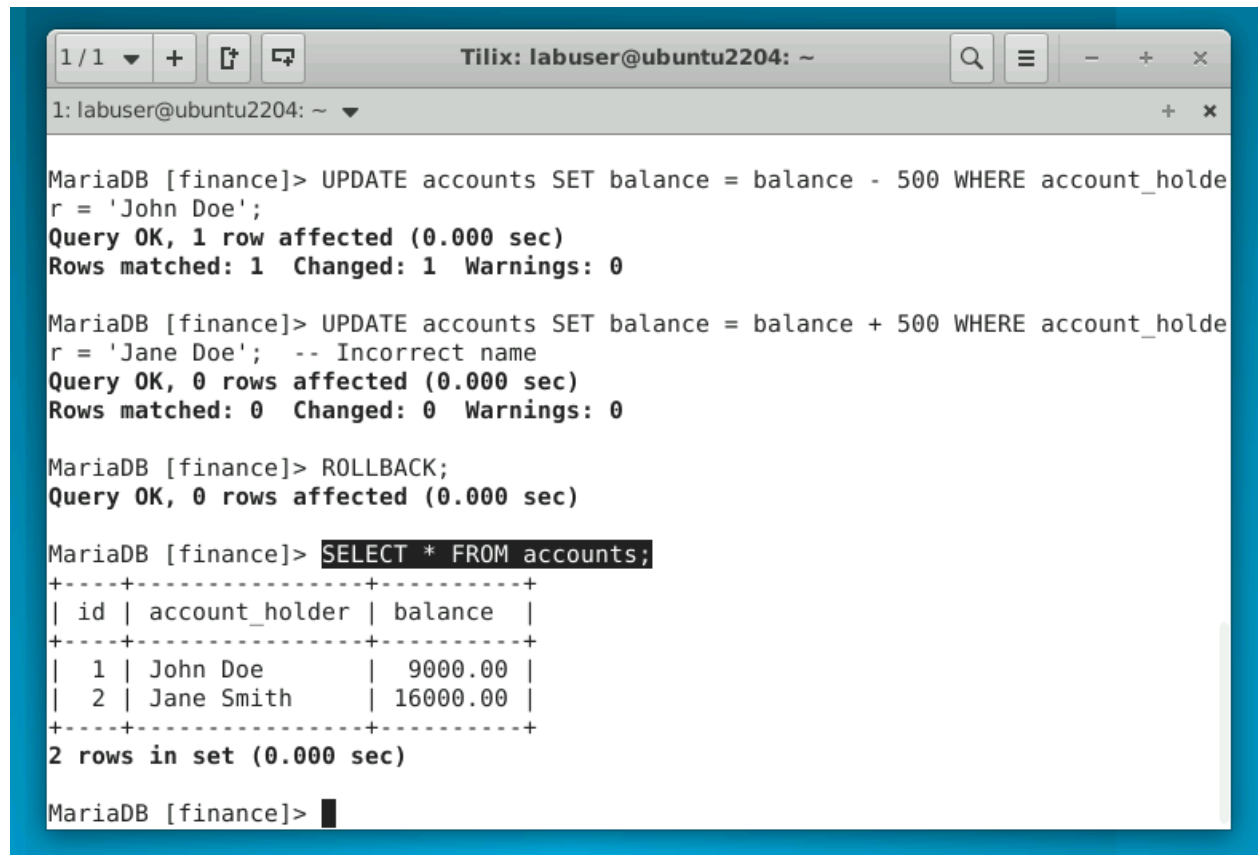
MariaDB [finance]> ROLLBACK;
Query OK, 0 rows affected (0.000 sec)

MariaDB [finance]>

```

2.7 Verify that the balances remain unchanged after the rollback:

SELECT * FROM accounts;



The screenshot shows a terminal window titled 'Tilix: labuser@ubuntu2204: ~'. The prompt is '1: labuser@ubuntu2204: ~'. The user enters several MySQL commands in the 'MariaDB [finance]' shell:

```
MariaDB [finance]> UPDATE accounts SET balance = balance - 500 WHERE account_holder = 'John Doe';
Query OK, 1 row affected (0.000 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [finance]> UPDATE accounts SET balance = balance + 500 WHERE account_holder = 'Jane Doe'; -- Incorrect name
Query OK, 0 rows affected (0.000 sec)
Rows matched: 0  Changed: 0  Warnings: 0

MariaDB [finance]> ROLLBACK;
Query OK, 0 rows affected (0.000 sec)

MariaDB [finance]> SELECT * FROM accounts;
```

The output of the SELECT statement is a table with 3 columns: id, account_holder, and balance. It shows 2 rows of data.

id	account_holder	balance
1	John Doe	9000.00
2	Jane Smith	16000.00

The terminal output concludes with '2 rows in set (0.000 sec)' and the prompt 'MariaDB [finance]>'.

By following these steps, you have successfully demonstrated the use of TCL commands in MySQL, including starting a transaction, committing the changes, and rolling back a transaction in case of errors.