Lab - 10: Database transactions

Team Members:

Jay Sanjaybhai Patel (jy451478@dal.ca)

Date:

March 28, 2024

Subject:

Software Development Concepts

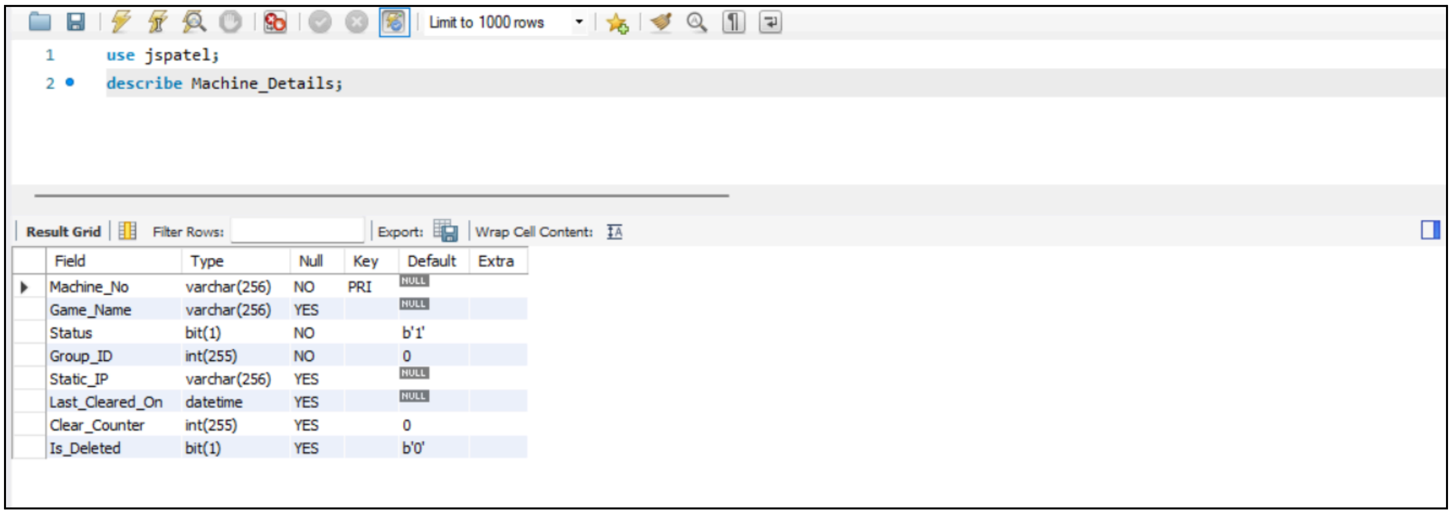
Professor:

Michael McAllister

Set-up

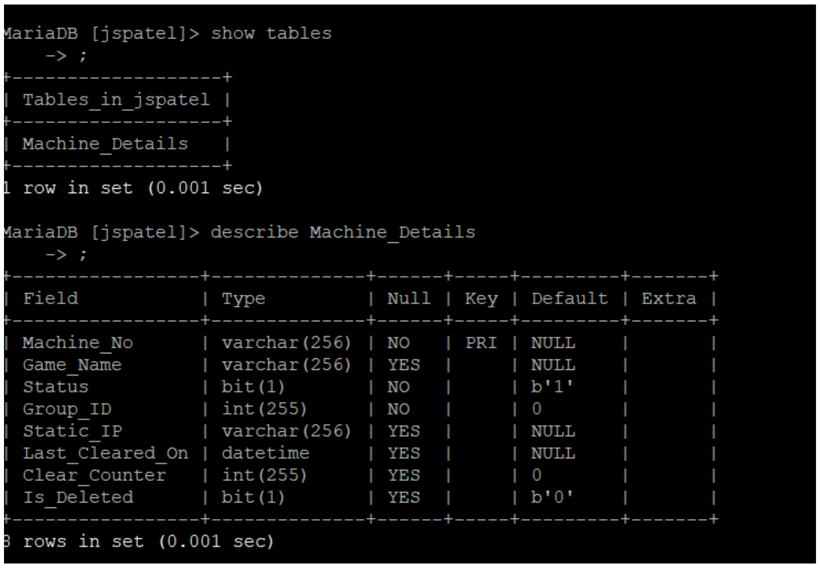
Create Table

use jspatel;
CREATE TABLE Machine_Details (
Machine_No VARCHAR(256) PRIMARY KEY,
Game_Name VARCHAR(256),
Status BIT(1) NOT NULL DEFAULT 1,
Group_ID INT(255) NOT NULL DEFAULT 0,
Static_IP VARCHAR(256),
Last_Cleared_On DATETIME,
Clear_Counter INT(255) DEFAULT 0,
Is_Deleted BIT(1) DEFAULT 0
);



```
use jspatel;
describe Machine_Details;
```

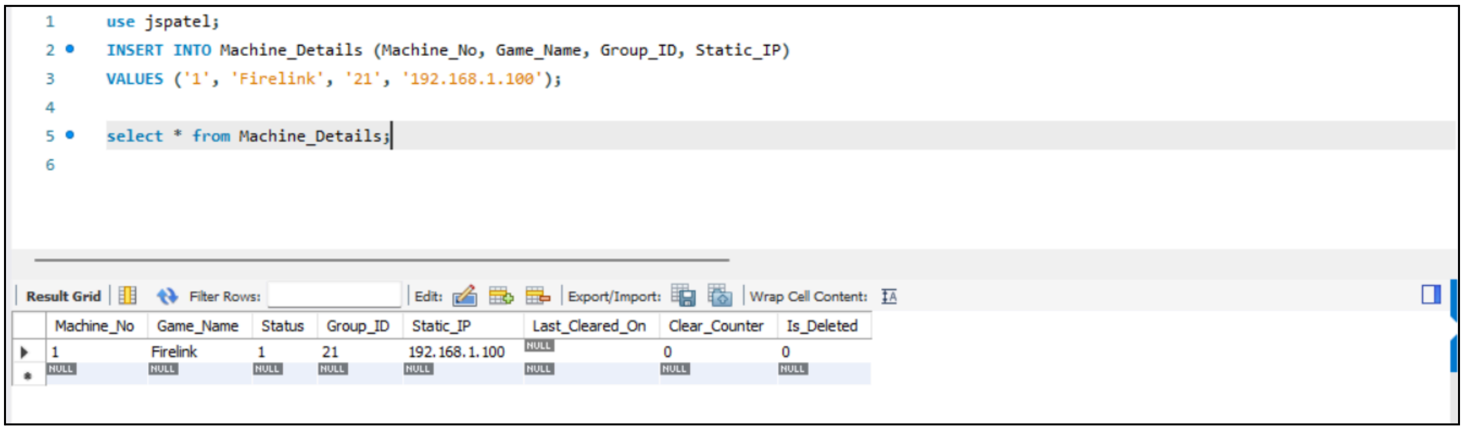
Field	Type	Null	Key	Default	Extra
Machine_No	varchar(256)	NO	PRI	NULL	
Game_Name	varchar(256)	YES		NULL	
Status	bit(1)	NO		b'1'	
Group_ID	int(255)	NO		0	
Static_IP	varchar(256)	YES		NULL	
Last_Cleared_On	datetime	YES		NULL	
Clear_Counter	int(255)	YES		0	
Is_Deleted	bit(1)	YES		b'0'	



```
MariaDB [jspatel]> show tables
    -> ;
+-------------------+
| Tables_in_jspatel |
+-------------------+
| Machine_Details   |
+-------------------+
1 row in set (0.001 sec)

MariaDB [jspatel]> describe Machine_Details
    -> ;
+-----------------+--------------+------+-----+---------+-------+
| Field           | Type         | Null | Key | Default | Extra |
+-----------------+--------------+------+-----+---------+-------+
| Machine_No      | varchar(256) | NO   | PRI | NULL    |       |
| Game_Name       | varchar(256) | YES  |     | NULL    |       |
| Status          | bit(1)       | NO   |     | b'1'    |       |
| Group_ID        | int(255)     | NO   |     | 0       |       |
| Static_IP       | varchar(256) | YES  |     | NULL    |       |
| Last_Cleared_On | datetime     | YES  |     | NULL    |       |
| Clear_Counter   | int(255)     | YES  |     | 0       |       |
| Is_Deleted      | bit(1)       | YES  |     | b'0'    |       |
+-----------------+--------------+------+-----+---------+-------+
8 rows in set (0.001 sec)
```

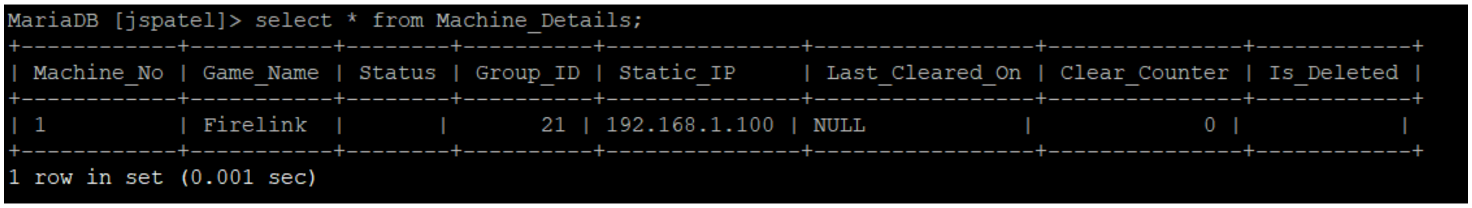
Insert

use jspatel;
INSERT INTO Machine_Details (Machine_No, Game_Name, Group_ID, Static_IP)
VALUES ('1', 'Firelink', '21', '192.168.1.100');



```
use jspatel;
INSERT INTO Machine_Details (Machine_No, Game_Name, Group_ID, Static_IP)
VALUES ('1', 'Firelink', '21', '192.168.1.100');

select * from Machine_Details;
```

Machine_No	Game_Name	Status	Group_ID	Static_IP	Last_Cleared_On	Clear_Counter	Is_Deleted
1	Firelink	1	21	192.168.1.100	NULL	0	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

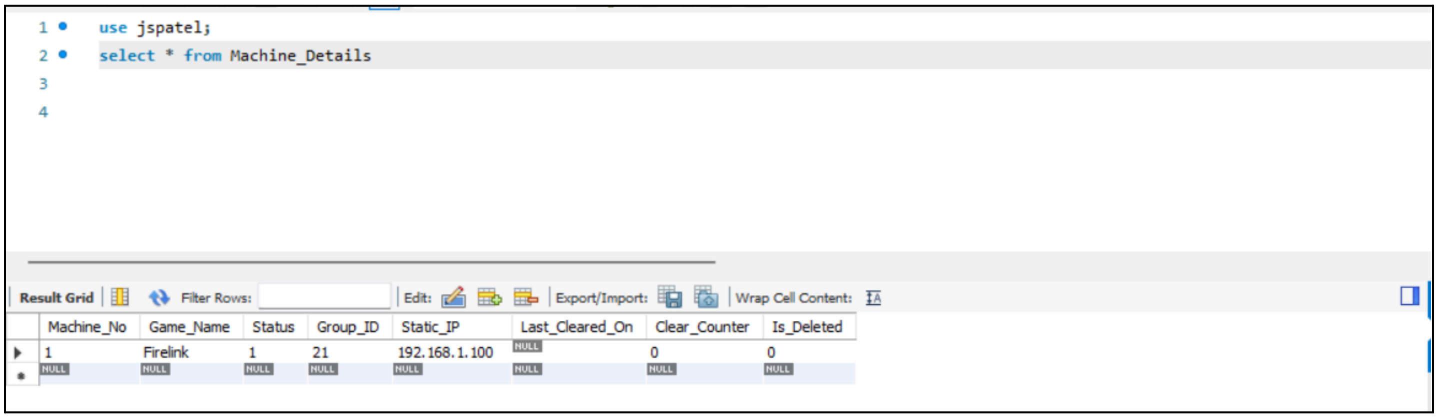


```
MariaDB [jspatel]> select * from Machine_Details;
+------------+-----------+--------+----------+---------------+-----------------+---------------+------------+
| Machine_No | Game_Name | Status | Group_ID | Static_IP     | Last_Cleared_On | Clear_Counter | Is_Deleted |
+------------+-----------+--------+----------+---------------+-----------------+---------------+------------+
| 1          | Firelink  |        |       21 | 192.168.1.100 | NULL            |             0 |            |
+------------+-----------+--------+----------+---------------+-----------------+---------------+------------+
1 row in set (0.001 sec)
```

Part - 1

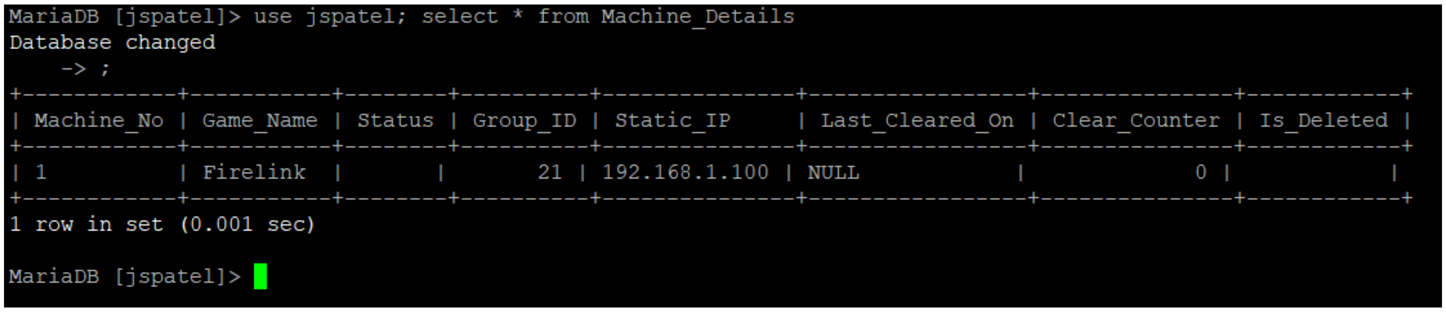
1) Execute SQL commands inside a transaction in the MySQLWorkbench window and verify the results in the putty window. Verify the results both before you do the command, after you do the command within the transaction, and then after you rollback the transaction.

Insert Data



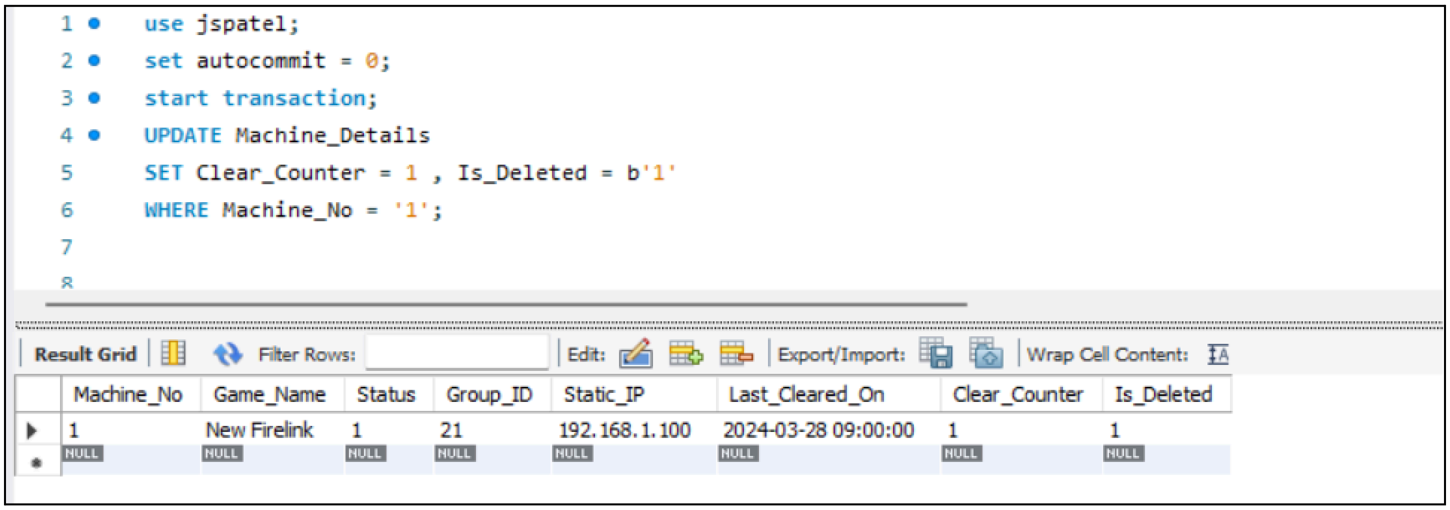
```
use jspatel;
select * from Machine_Details
```

Machine_No	Game_Name	Status	Group_ID	Static_IP	Last_Cleared_On	Clear_Counter	Is_Deleted
1	Firelink	1	21	192.168.1.100	NULL	0	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



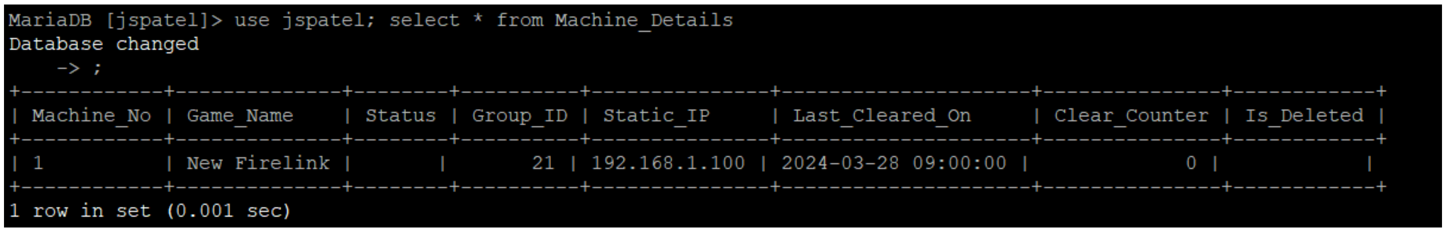
```
MariaDB [jspatel]> use jspatel; select * from Machine_Details
Database changed
    -> ;
+------------+-----------+--------+----------+---------------+-----------------+---------------+------------+
| Machine_No | Game_Name | Status | Group_ID | Static_IP     | Last_Cleared_On | Clear_Counter | Is_Deleted |
+------------+-----------+--------+----------+---------------+-----------------+---------------+------------+
| 1          | Firelink  |        |       21 | 192.168.1.100 | NULL            |             0 |            |
+------------+-----------+--------+----------+---------------+-----------------+---------------+------------+
1 row in set (0.001 sec)

MariaDB [jspatel]>
```

Start Transaction (Not Commit - Update the data which was recently added)



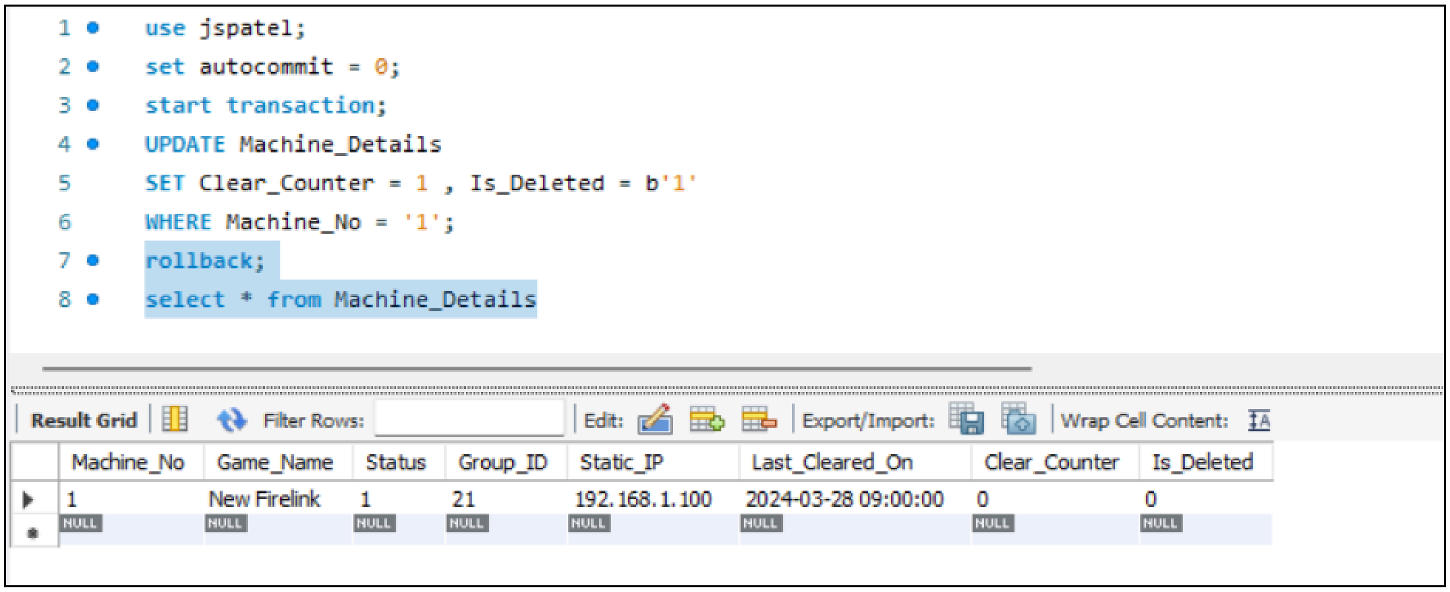
```
use jspatel;
set autocommit = 0;
start transaction;
UPDATE Machine_Details
SET Clear_Counter = 1 , Is_Deleted = b'1'
WHERE Machine_No = '1';
```

Machine_No	Game_Name	Status	Group_ID	Static_IP	Last_Cleared_On	Clear_Counter	Is_Deleted
1	New Firelink	1	21	192.168.1.100	2024-03-28 09:00:00	1	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



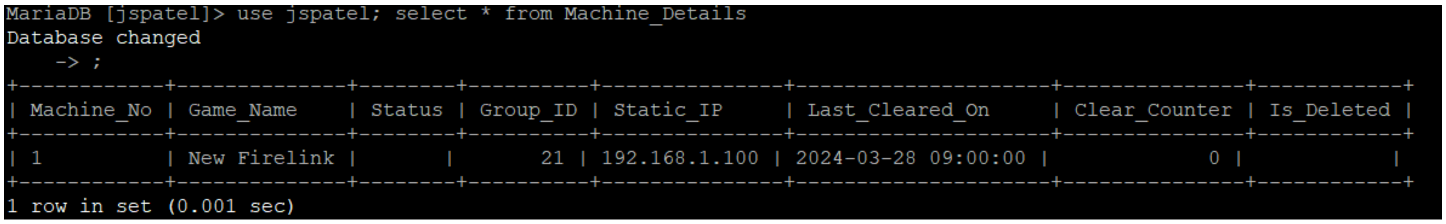
```
MariaDB [jspatel]> use jspatel; select * from Machine_Details
Database changed
    -> ;
+------------+--------------+--------+----------+---------------+---------------------+---------------+------------+
| Machine_No | Game_Name    | Status | Group_ID | Static_IP     | Last_Cleared_On     | Clear_Counter | Is_Deleted |
+------------+--------------+--------+----------+---------------+---------------------+---------------+------------+
| 1          | New Firelink |        |       21 | 192.168.1.100 | 2024-03-28 09:00:00 |             0 |            |
+------------+--------------+--------+----------+---------------+---------------------+---------------+------------+
1 row in set (0.001 sec)
```

Rollback Transaction (Before and after this transaction there was no change in putty window)



```
use jspatel;
set autocommit = 0;
start transaction;
UPDATE Machine_Details
SET Clear_Counter = 1 , Is_Deleted = b'1'
WHERE Machine_No = '1';
rollback;
select * from Machine_Details
```

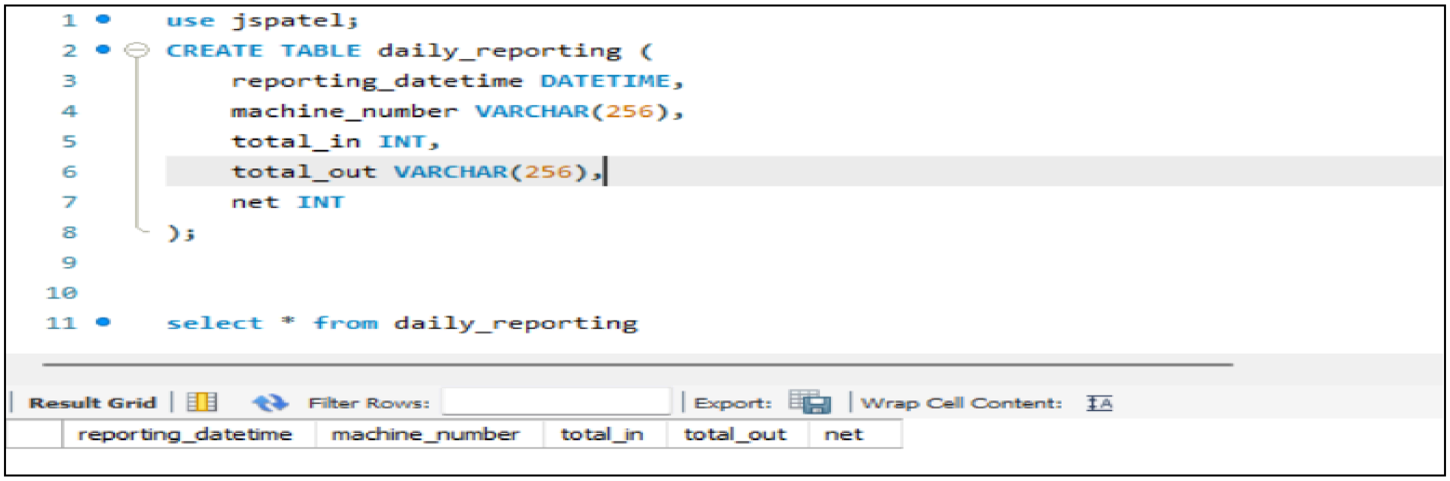
Machine_No	Game_Name	Status	Group_ID	Static_IP	Last_Cleared_On	Clear_Counter	Is_Deleted
1	New Firelink	1	21	192.168.1.100	2024-03-28 09:00:00	0	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



```
MariaDB [jspatel]> use jspatel; select * from Machine_Details
Database changed
    -> ;
+------------+--------------+--------+----------+---------------+---------------------+---------------+------------+
| Machine_No | Game_Name    | Status | Group_ID | Static_IP     | Last_Cleared_On     | Clear_Counter | Is_Deleted |
+------------+--------------+--------+----------+---------------+---------------------+---------------+------------+
| 1          | New Firelink |        |       21 | 192.168.1.100 | 2024-03-28 09:00:00 |             0 |            |
+------------+--------------+--------+----------+---------------+---------------------+---------------+------------+
1 row in set (0.001 sec)
```

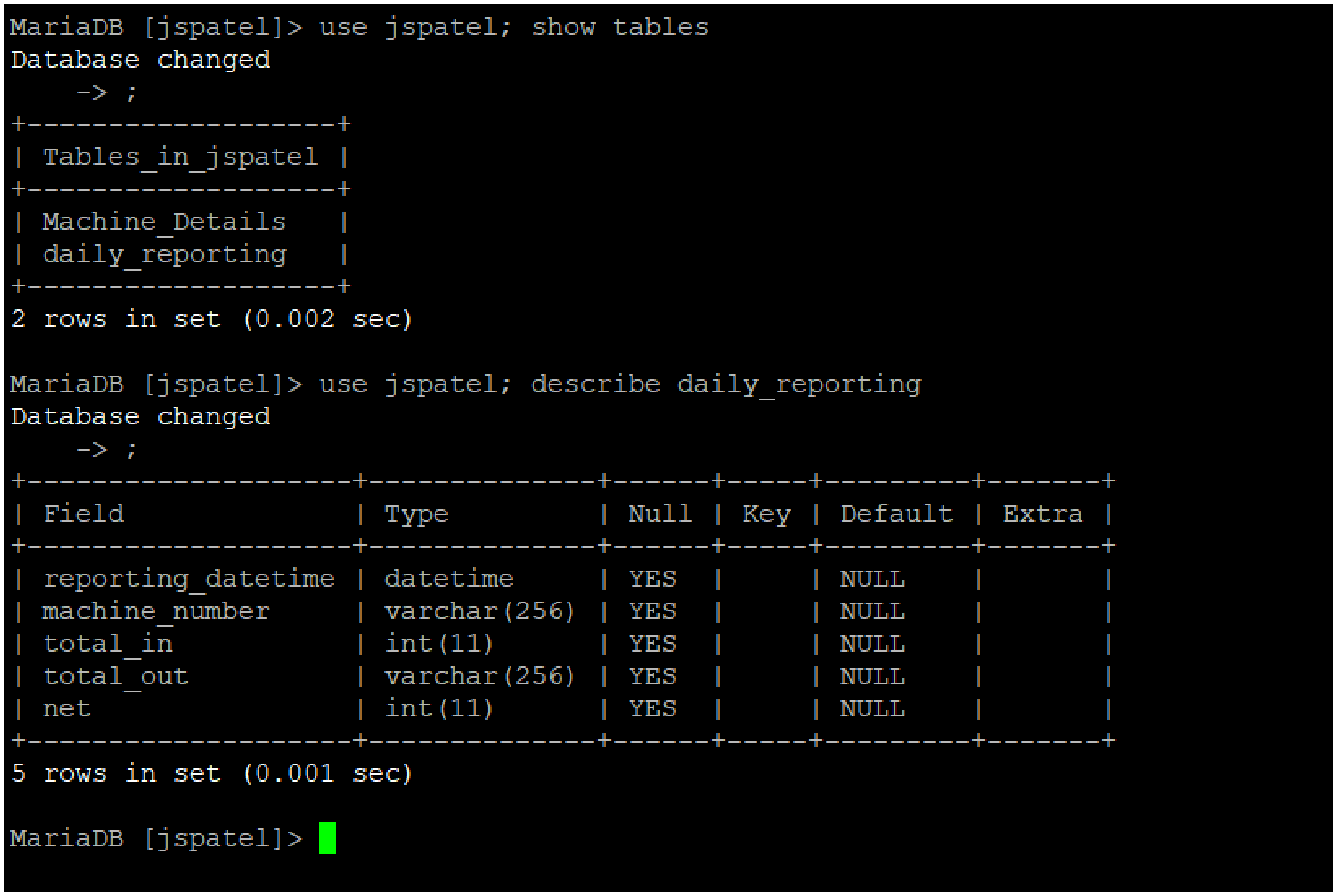
2) The SQL commands to try will be a set of CRUD (create, read, update, delete) commands on the tables…well, without the read one since that doesn't cause race conditions:

a. Create a table



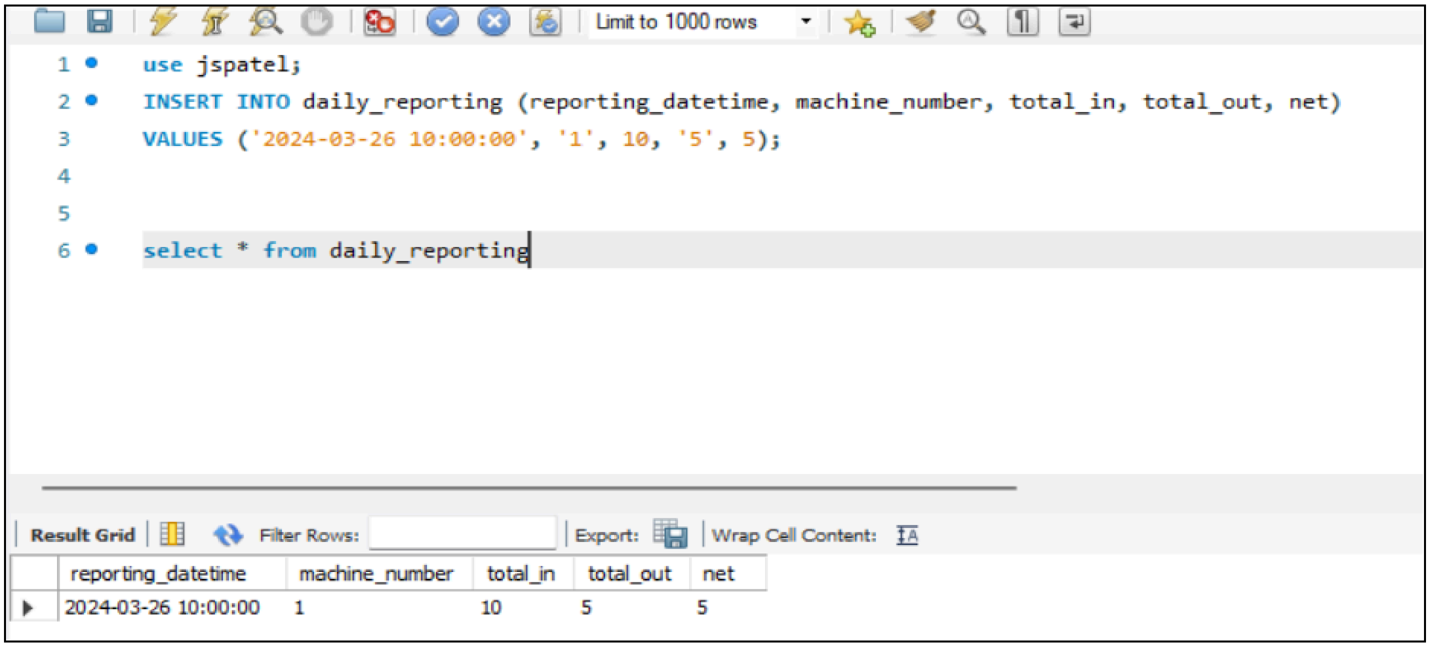
```
use jspatel;
CREATE TABLE daily_reporting (
    reporting_datetime DATETIME,
    machine_number VARCHAR(256),
    total_in INT,
    total_out VARCHAR(256),
    net INT
);


select * from daily_reporting
```

reporting_datetime	machine_number	total_in	total_out	net



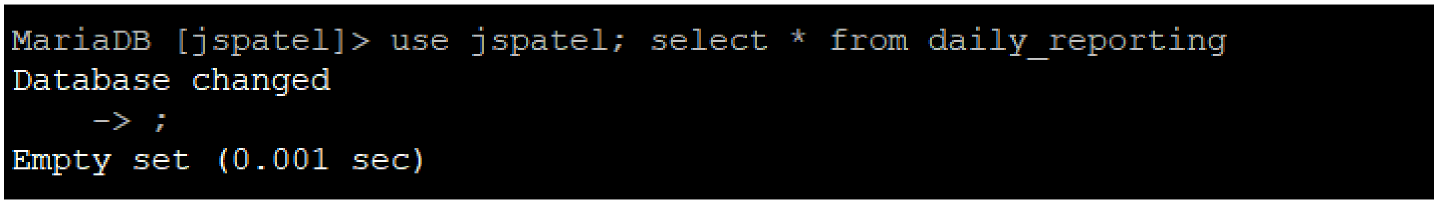
```
MariaDB [jspatel]> use jspatel; show tables
Database changed
    -> ;
+-------------------+
| Tables_in_jspatel |
+-------------------+
| Machine_Details   |
| daily_reporting   |
+-------------------+
2 rows in set (0.002 sec)

MariaDB [jspatel]> use jspatel; describe daily_reporting
Database changed
    -> ;
+--------------------+--------------+------+-----+---------+-------+
| Field              | Type         | Null | Key | Default | Extra |
+--------------------+--------------+------+-----+---------+-------+
| reporting_datetime | datetime     | YES  |     | NULL    |       |
| machine_number     | varchar(256) | YES  |     | NULL    |       |
| total_in           | int(11)      | YES  |     | NULL    |       |
| total_out          | varchar(256) | YES  |     | NULL    |       |
| net                | int(11)      | YES  |     | NULL    |       |
+--------------------+--------------+------+-----+---------+-------+
5 rows in set (0.001 sec)

MariaDB [jspatel]>
```

b. Create a row of data in a table



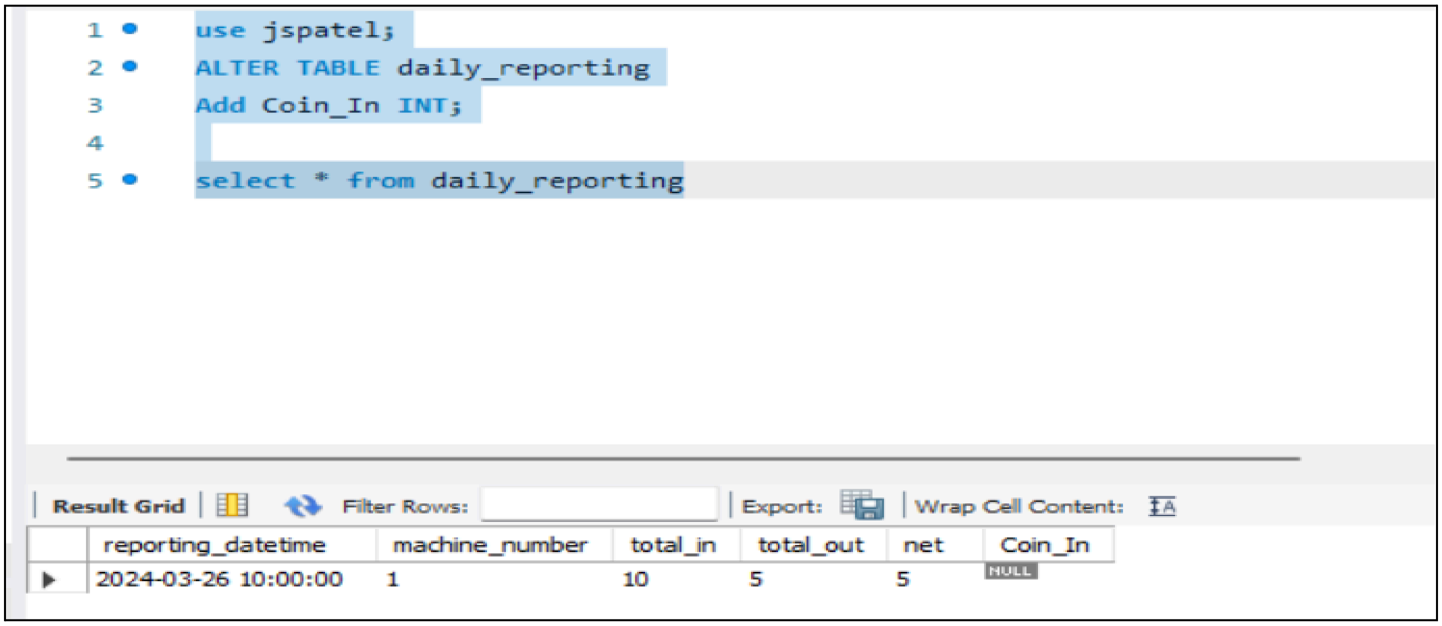
```
use jspatel;
INSERT INTO daily_reporting (reporting_datetime, machine_number, total_in, total_out, net)
VALUES ('2024-03-26 10:00:00', '1', 10, '5', 5);


select * from daily_reporting
```

reporting_datetime	machine_number	total_in	total_out	net
2024-03-26 10:00:00	1	10	5	5



```
MariaDB [jspatel]> use jspatel; select * from daily_reporting
Database changed
    -> ;
Empty set (0.001 sec)
```

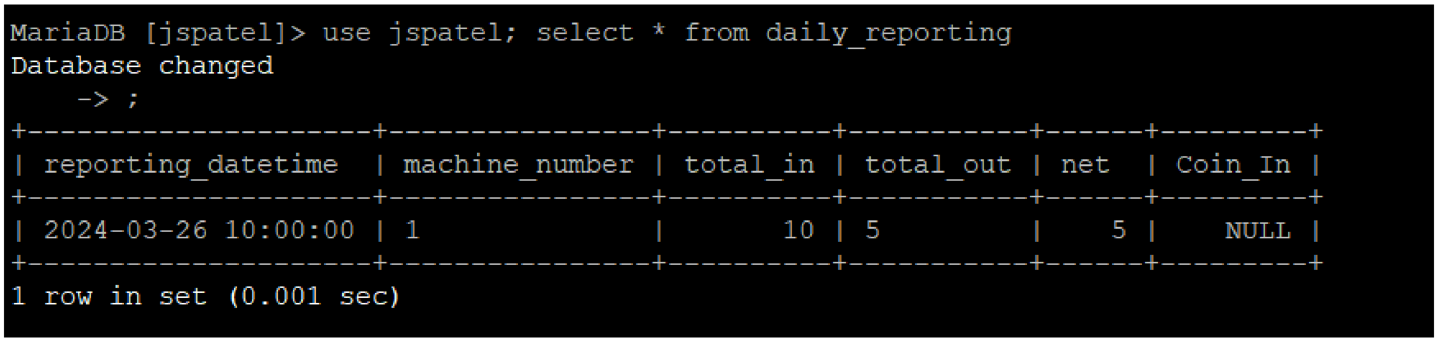
c. Add a column to an existing table



```
use jspatel;
ALTER TABLE daily_reporting
Add Coin_In INT;

select * from daily_reporting
```

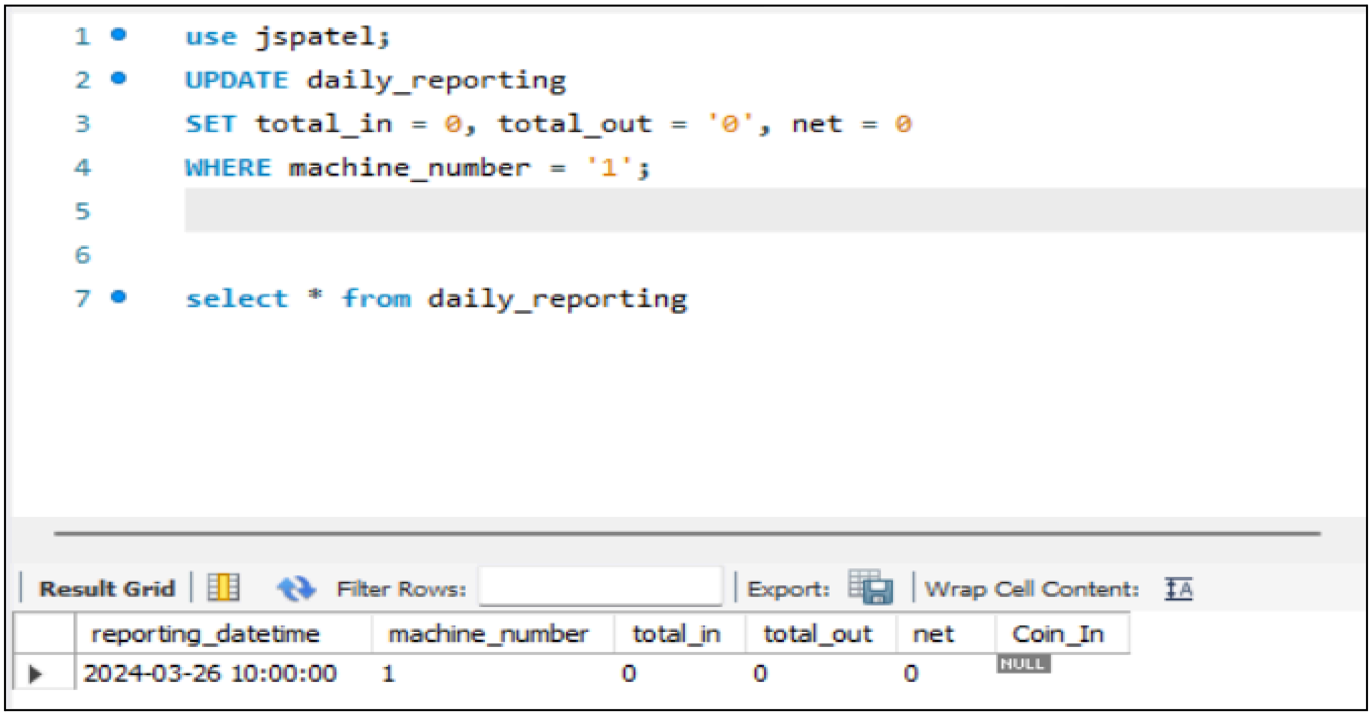
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

reporting_datetime	machine_number	total_in	total_out	net	Coin_In
2024-03-26 10:00:00	1	10	5	5	NULL



```
MariaDB [jspatel]> use jspatel; select * from daily_reporting
Database changed
    -> ;
+---------------------+----------------+----------+-----------+------+---------+
| reporting_datetime  | machine_number | total_in | total_out | net  | Coin_In |
+---------------------+----------------+----------+-----------+------+---------+
| 2024-03-26 10:00:00 | 1              |       10 | 5         |    5 |    NULL |
+---------------------+----------------+----------+-----------+------+---------+
1 row in set (0.001 sec)
```

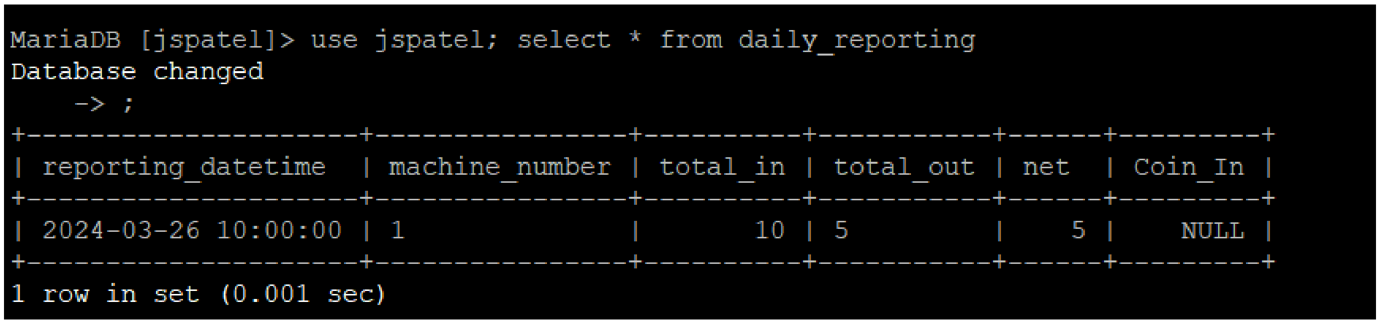
d. Update a row of data in a table



```
use jspatel;
UPDATE daily_reporting
SET total_in = 0, total_out = '0', net = 0
WHERE machine_number = '1';


select * from daily_reporting
```

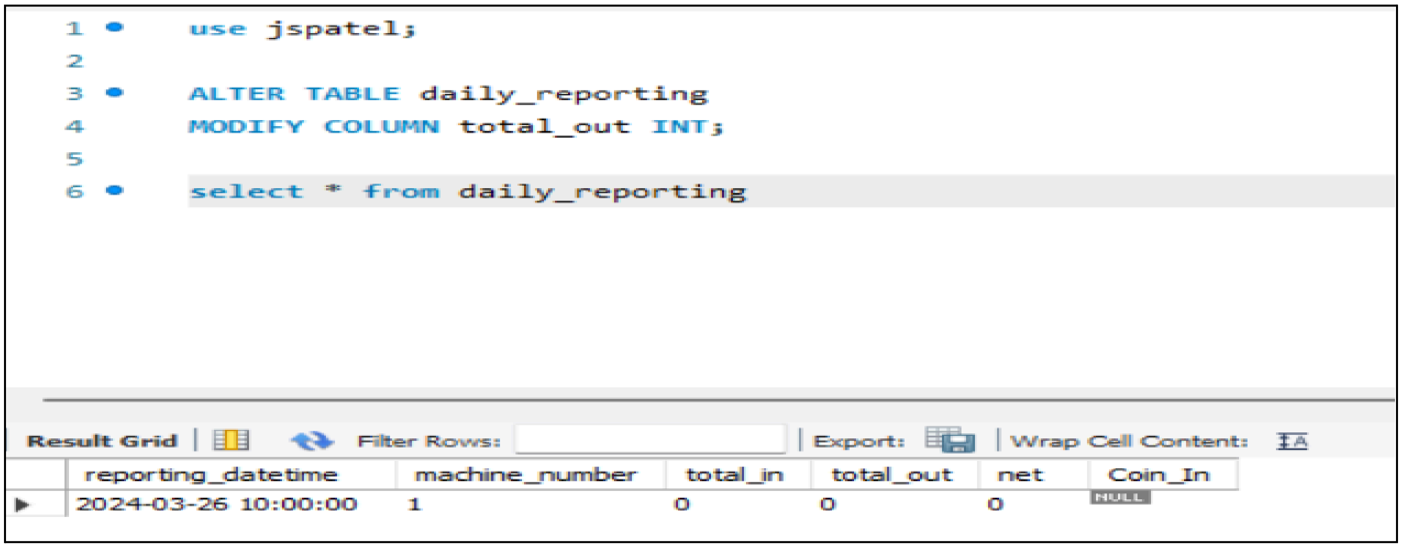
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

reporting_datetime	machine_number	total_in	total_out	net	Coin_In
2024-03-26 10:00:00	1	0	0	0	NULL



```
MariaDB [jspatel]> use jspatel; select * from daily_reporting
Database changed
    -> ;
+---------------------+----------------+----------+-----------+------+---------+
| reporting_datetime  | machine_number | total_in | total_out | net  | Coin_In |
+---------------------+----------------+----------+-----------+------+---------+
| 2024-03-26 10:00:00 | 1              |       10 | 5         |    5 |    NULL |
+---------------------+----------------+----------+-----------+------+---------+
1 row in set (0.001 sec)
```

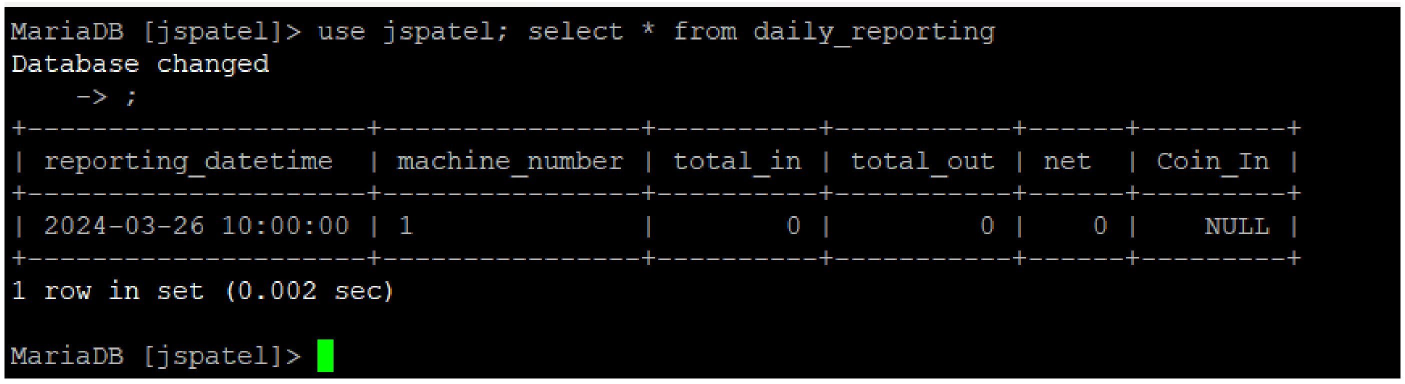
e. Update the data type of a column in a table



```
use jspatel;

ALTER TABLE daily_reporting
MODIFY COLUMN total_out INT;

select * from daily_reporting
```

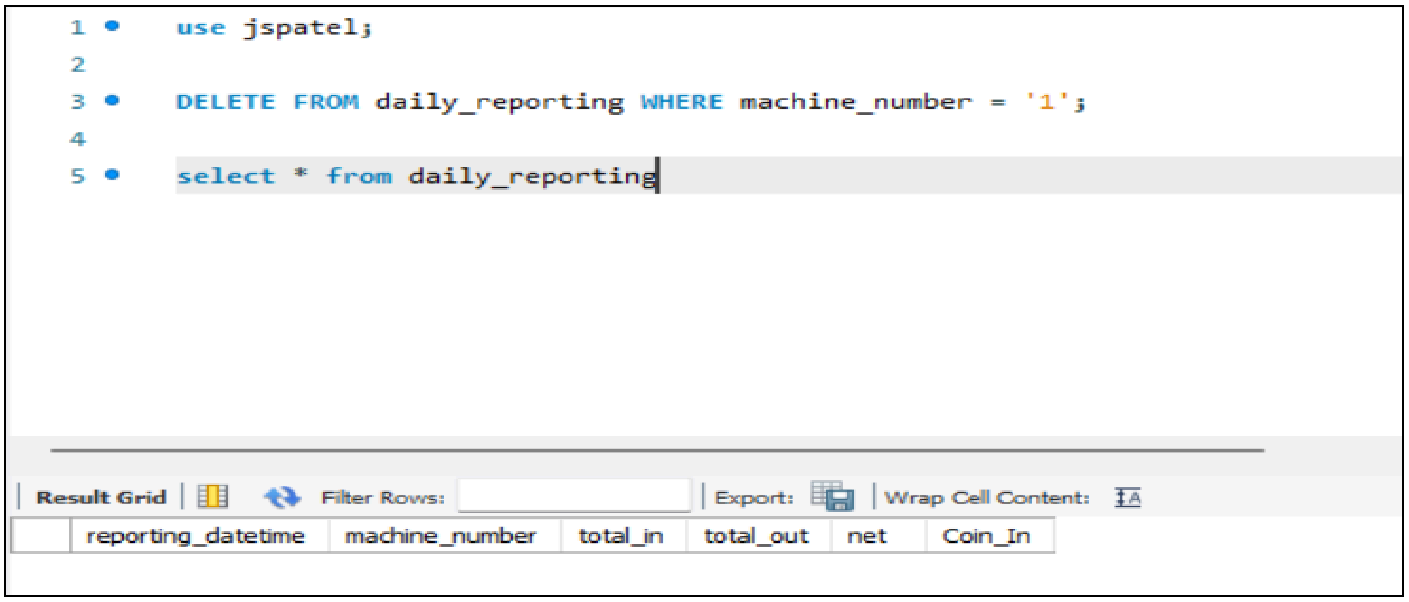
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

reporting_datetime	machine_number	total_in	total_out	net	Coin_In
2024-03-26 10:00:00	1	0	0	0	NULL



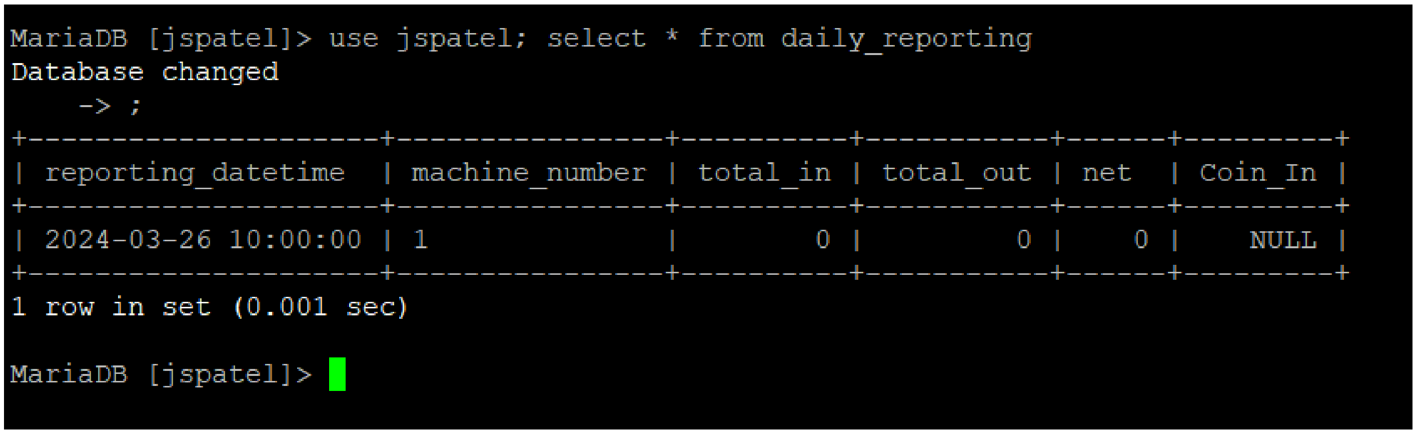
```
MariaDB [jspatel]> use jspatel; select * from daily_reporting
Database changed
    -> ;
+---------------------+----------------+----------+-----------+------+---------+
| reporting_datetime  | machine_number | total_in | total_out | net  | Coin_In |
+---------------------+----------------+----------+-----------+------+---------+
| 2024-03-26 10:00:00 | 1              |        0 |         0 |    0 |    NULL |
+---------------------+----------------+----------+-----------+------+---------+
1 row in set (0.002 sec)

MariaDB [jspatel]>
```

f. Delete a row in a table



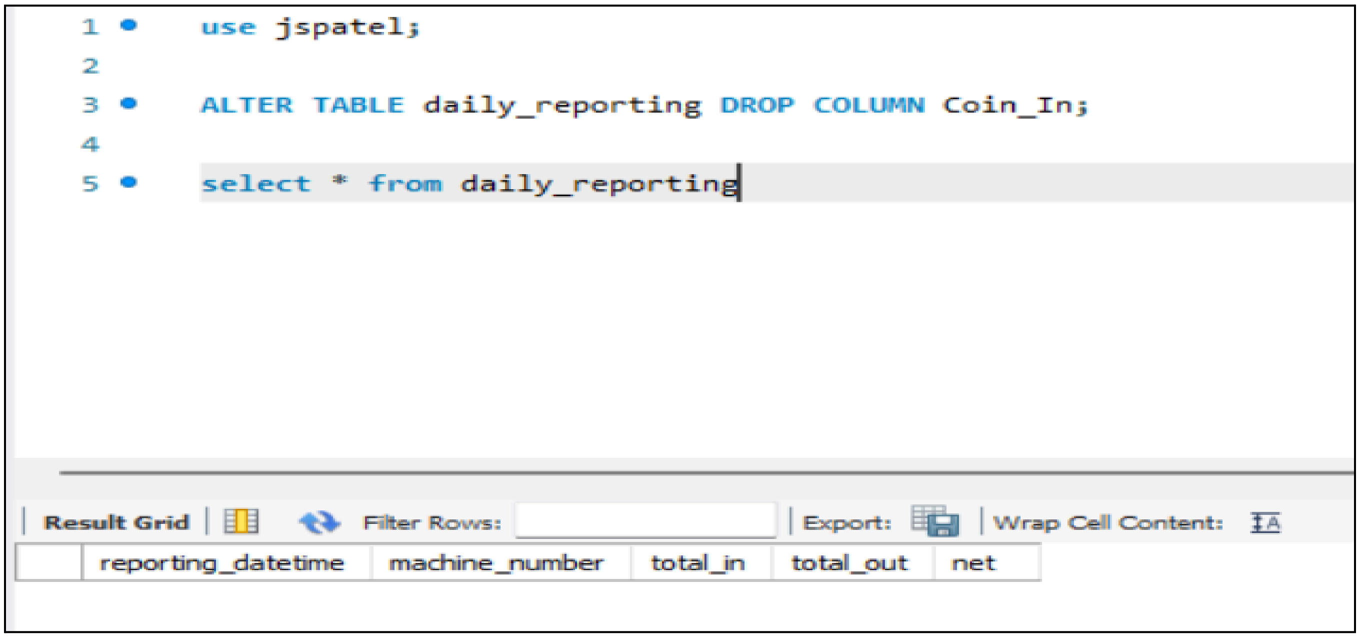
```
use jspatel;

DELETE FROM daily_reporting WHERE machine_number = '1';

select * from daily_reporting
```

	reporting_datetime	machine_number	total_in	total_out	net	Coin_In



```
MariaDB [jspatel]> use jspatel; select * from daily_reporting
Database changed
    -> ;
+---------------------+----------------+----------+-----------+------+---------+
| reporting_datetime  | machine_number | total_in | total_out | net  | Coin_In |
+---------------------+----------------+----------+-----------+------+---------+
| 2024-03-26 10:00:00 | 1              |        0 |         0 |    0 |    NULL |
+---------------------+----------------+----------+-----------+------+---------+
1 row in set (0.001 sec)

MariaDB [jspatel]>
```

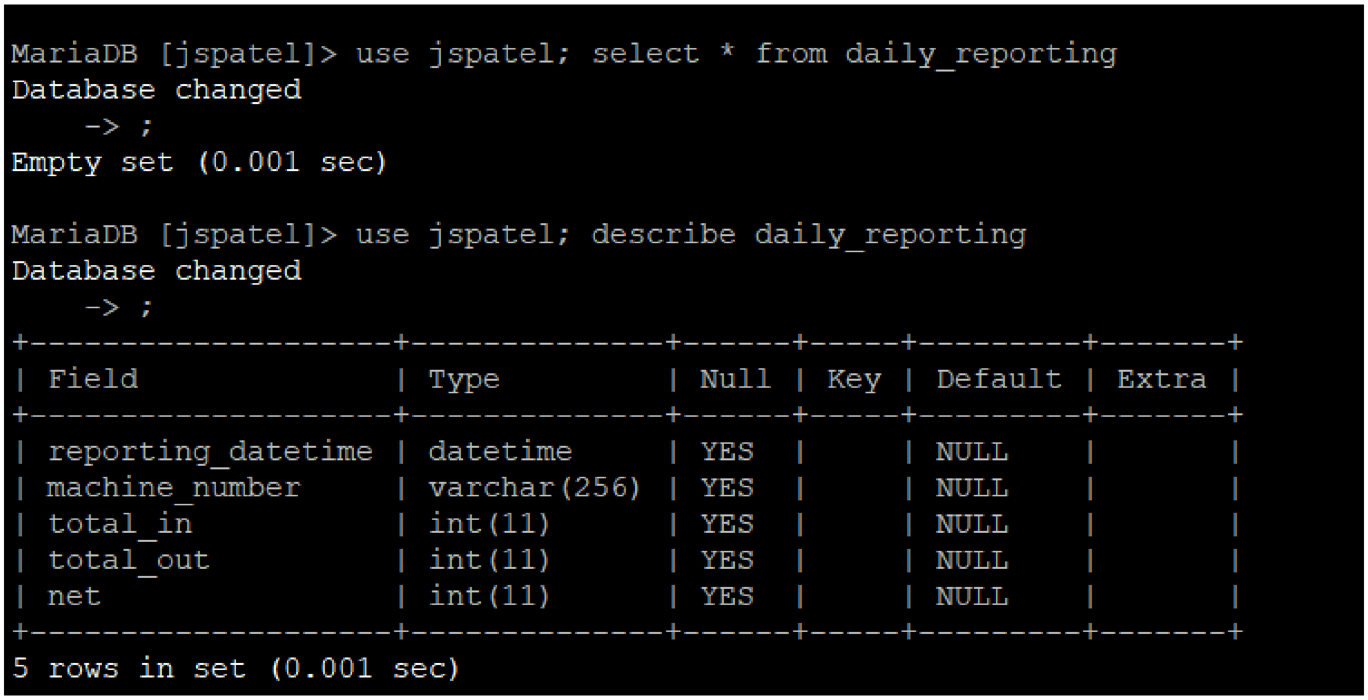
g. Delete a column in a table



```
use jspatel;

ALTER TABLE daily_reporting DROP COLUMN Coin_In;

select * from daily_reporting
```

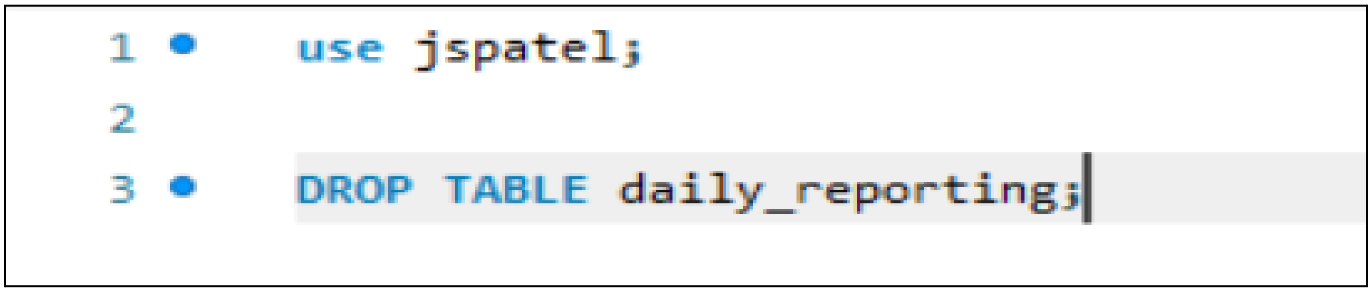
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	reporting_datetime	machine_number	total_in	total_out	net

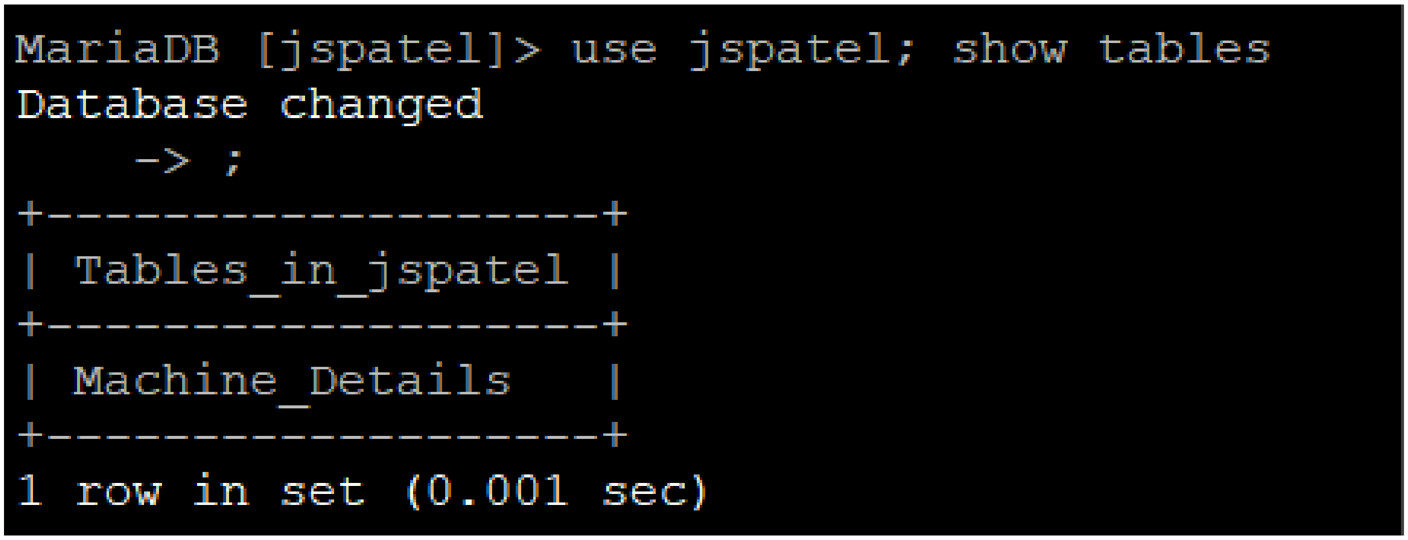


```
MariaDB [jspatel]> use jspatel; select * from daily_reporting
Database changed
    -> ;
Empty set (0.001 sec)

MariaDB [jspatel]> use jspatel; describe daily_reporting
Database changed
    -> ;
+--------------------+--------------+------+-----+---------+-------+
| Field              | Type         | Null | Key | Default | Extra |
+--------------------+--------------+------+-----+---------+-------+
| reporting_datetime | datetime     | YES  |     | NULL    |       |
| machine_number     | varchar(256) | YES  |     | NULL    |       |
| total_in           | int(11)      | YES  |     | NULL    |       |
| total_out          | int(11)      | YES  |     | NULL    |       |
| net                | int(11)      | YES  |     | NULL    |       |
+--------------------+--------------+------+-----+---------+-------+
5 rows in set (0.001 sec)
```

h. Delete a table

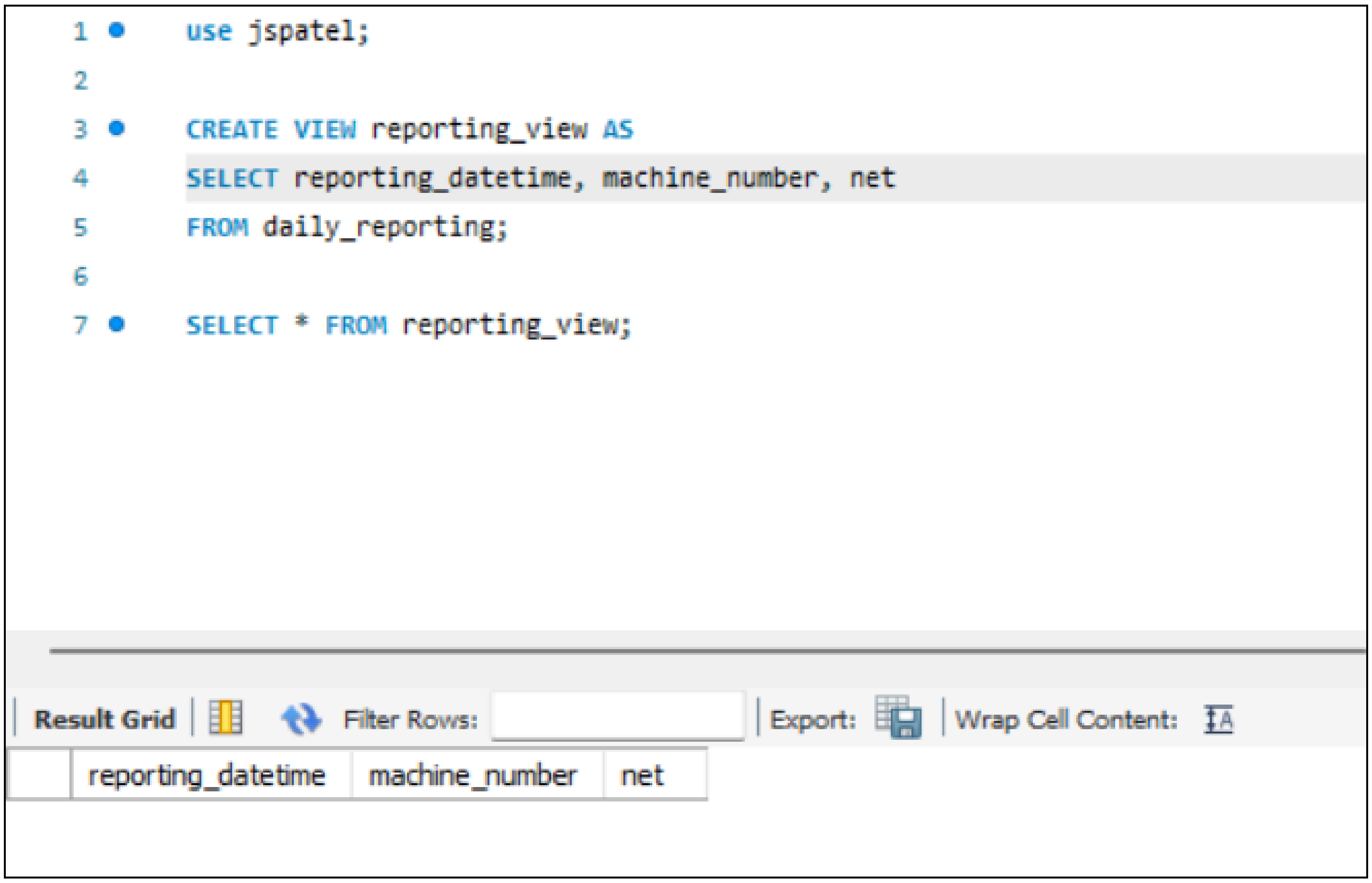


```
use jspatel;

DROP TABLE daily_reporting;
```



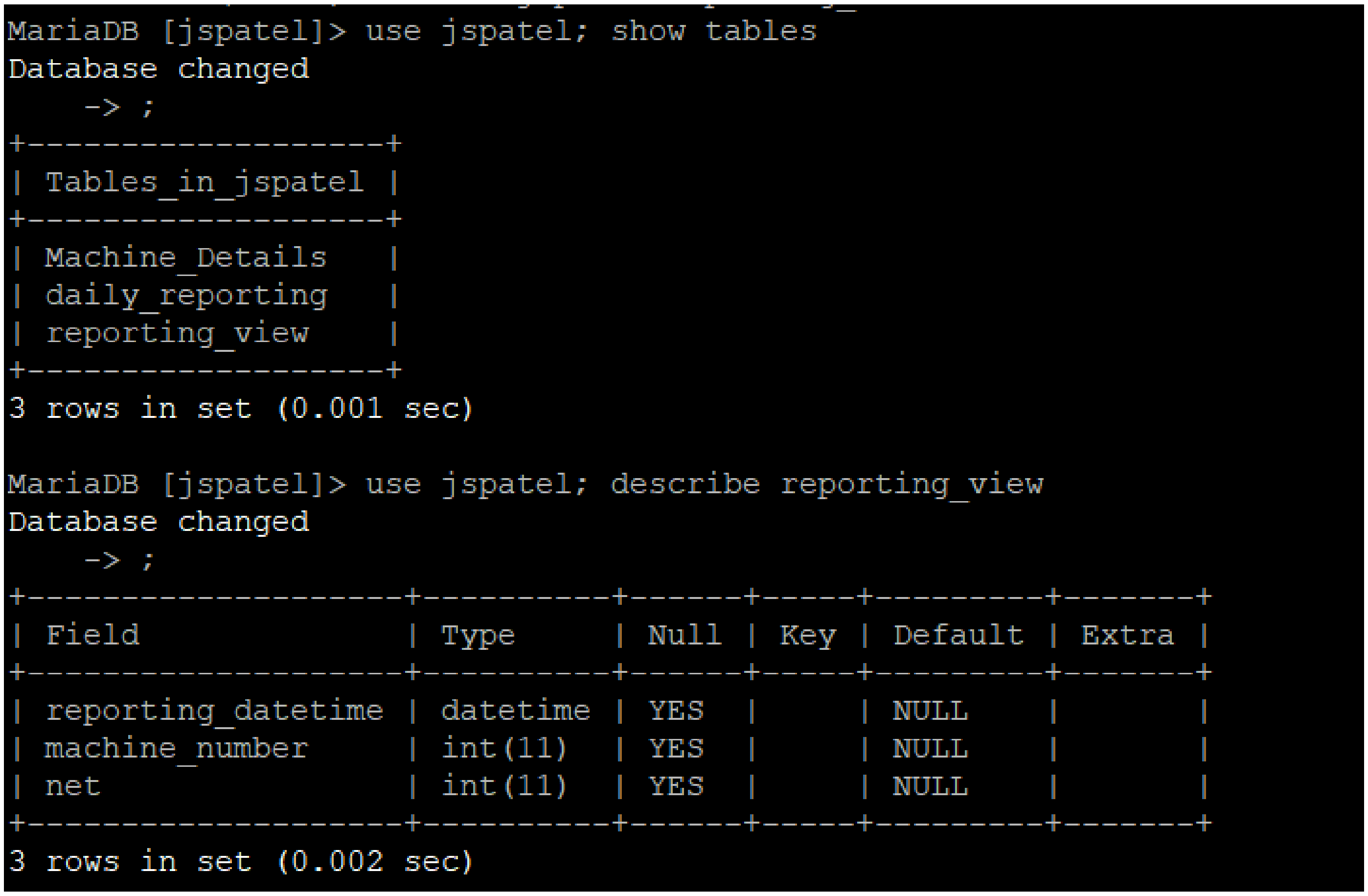
```
MariaDB [jspatel]> use jspatel; show tables
Database changed
    -> ;
+-------------------+
| Tables_in_jspatel |
+-------------------+
| Machine_Details   |
+-------------------+
1 row in set (0.001 sec)
```

i. Create a view



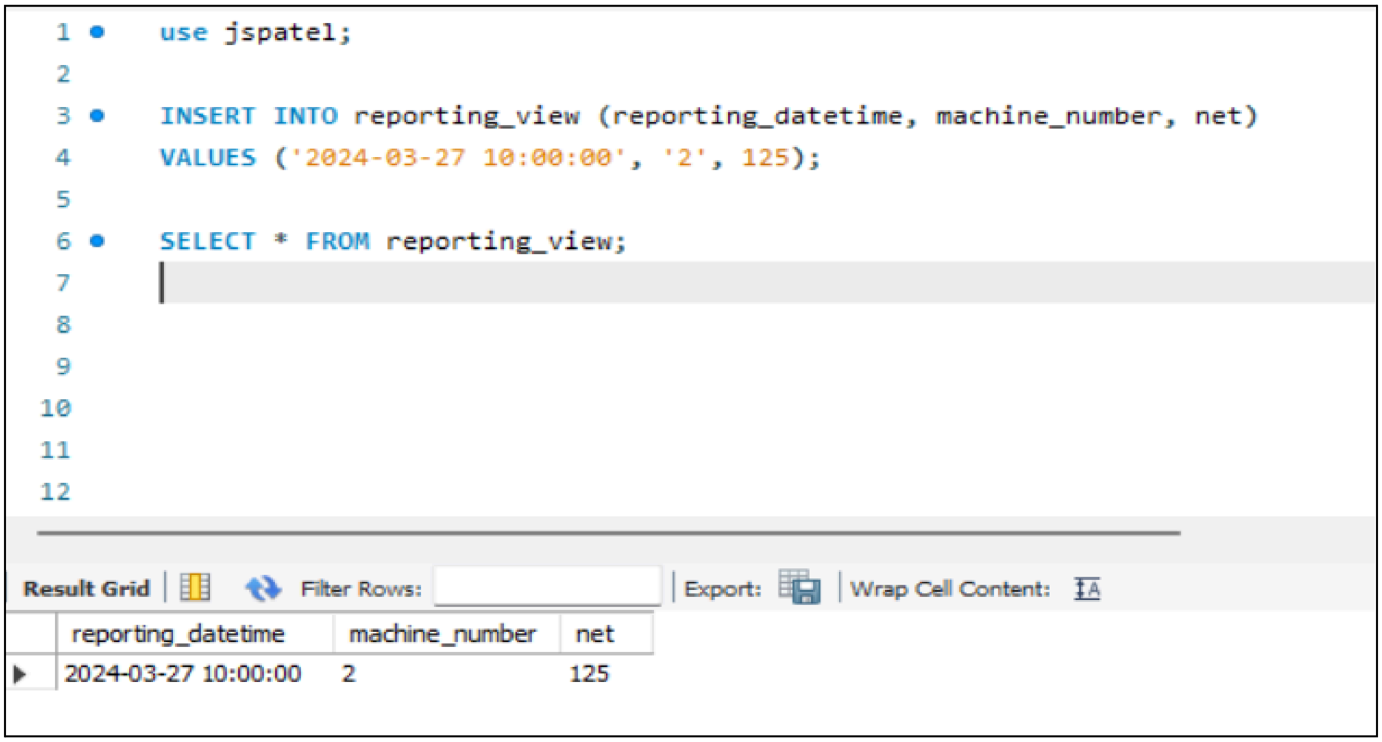
```
use jspatel;

CREATE VIEW reporting_view AS
SELECT reporting_datetime, machine_number, net
FROM daily_reporting;

SELECT * FROM reporting_view;
```

	reporting_datetime	machine_number	net



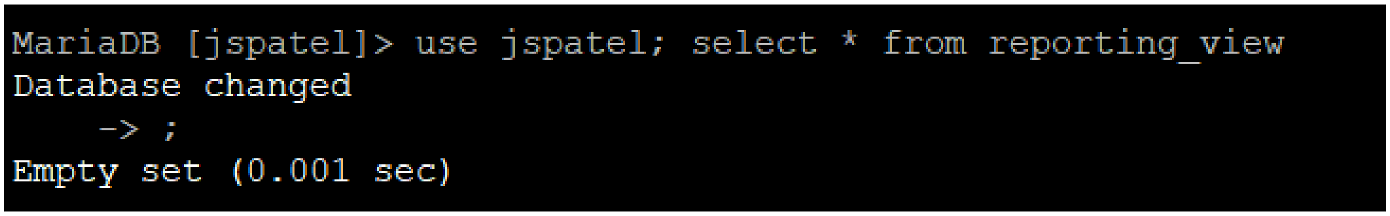
```
MariaDB [jspatel]> use jspatel; show tables
Database changed
    -> ;
+-------------------+
| Tables_in_jspatel |
+-------------------+
| Machine_Details   |
| daily_reporting   |
| reporting_view    |
+-------------------+
3 rows in set (0.001 sec)

MariaDB [jspatel]> use jspatel; describe reporting_view
Database changed
    -> ;
+--------------------+----------+------+-----+---------+-------+
| Field              | Type     | Null | Key | Default | Extra |
+--------------------+----------+------+-----+---------+-------+
| reporting_datetime | datetime | YES  |     | NULL    |       |
| machine_number     | int(11)  | YES  |     | NULL    |       |
| net                | int(11)  | YES  |     | NULL    |       |
+--------------------+----------+------+-----+---------+-------+
3 rows in set (0.002 sec)
```

<u>j. Add a row to a view</u>

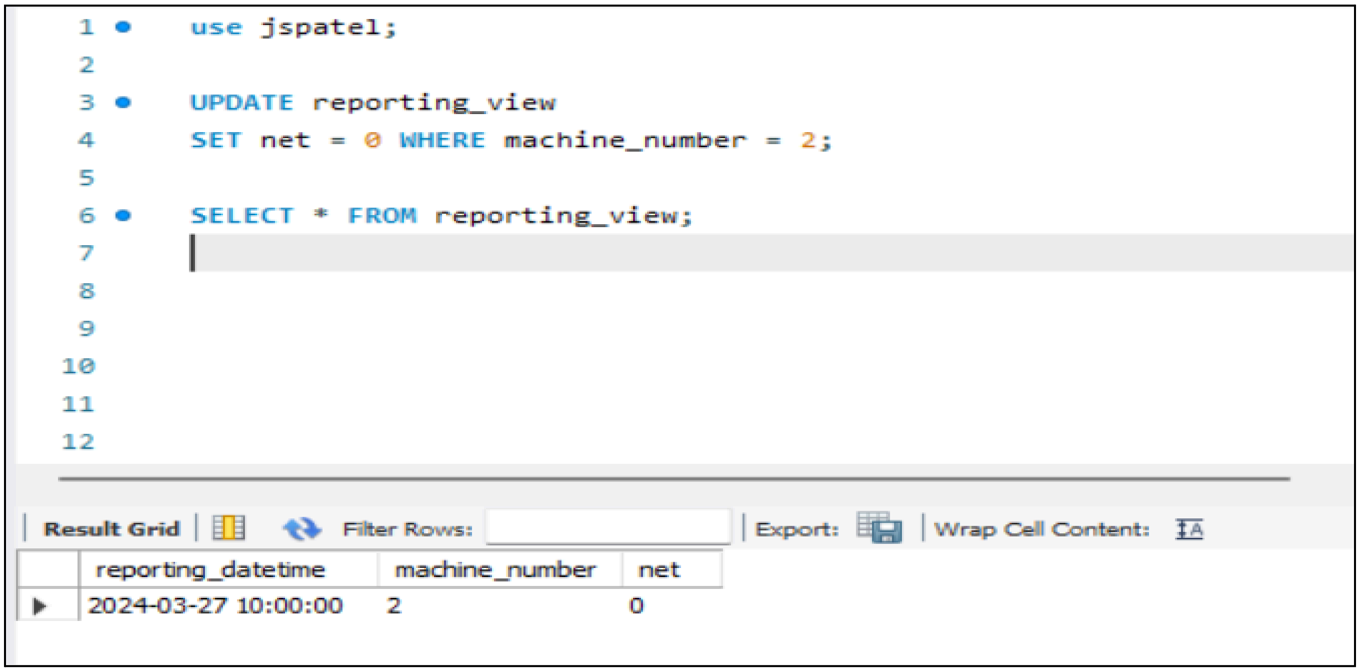


```sql
use jspatel;

INSERT INTO reporting_view (reporting_datetime, machine_number, net)
VALUES ('2024-03-27 10:00:00', '2', 125);

SELECT * FROM reporting_view;
```

reporting_datetime	machine_number	net
2024-03-27 10:00:00	2	125

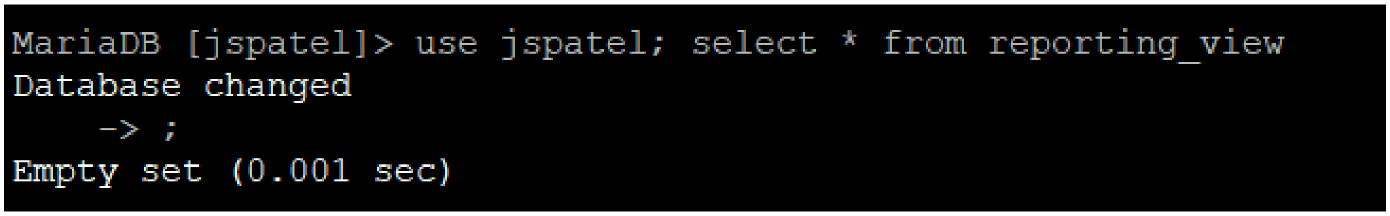

```
MariaDB [jspatel]> use jspatel; select * from reporting_view
Database changed
    -> ;
Empty set (0.001 sec)
```

k. Update a row in a view

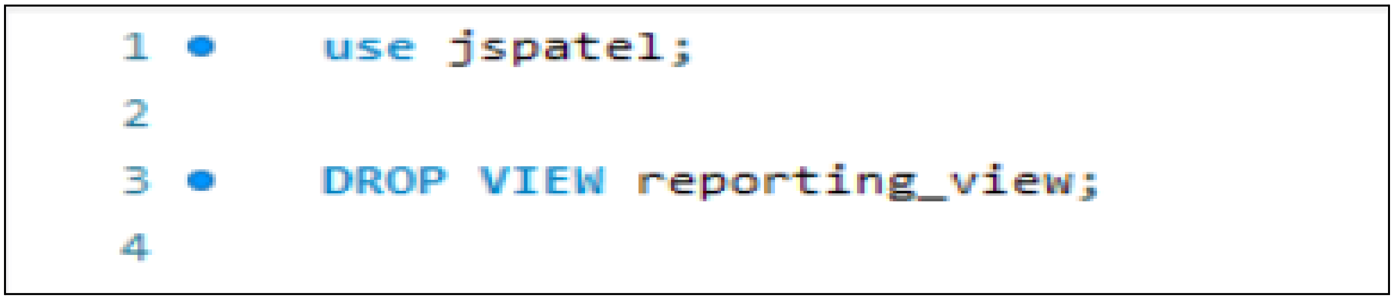

```
use jspatel;

UPDATE reporting_view
SET net = 0 WHERE machine_number = 2;

SELECT * FROM reporting_view;
```

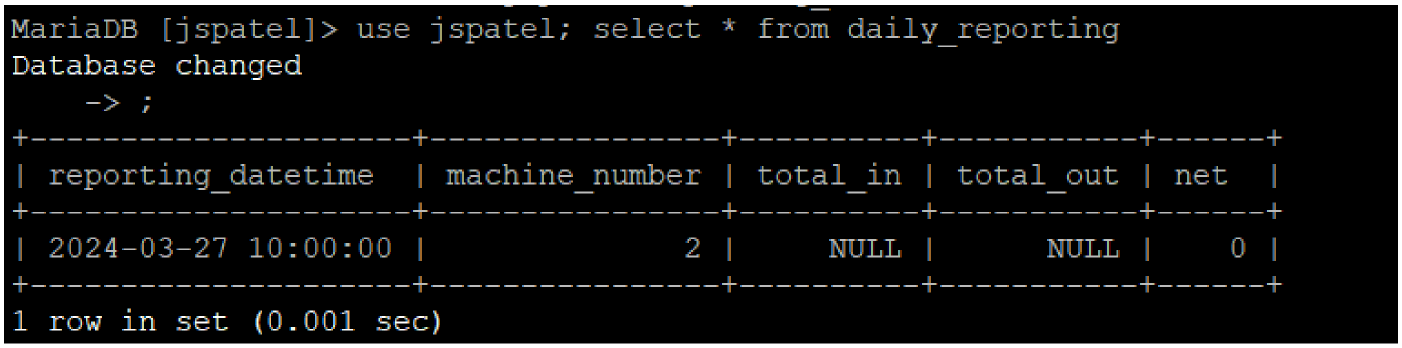
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

reporting_datetime	machine_number	net
2024-03-27 10:00:00	2	0


```
MariaDB [jspatel]> use jspatel; select * from reporting_view
Database changed
    -> ;
Empty set (0.001 sec)
```

l. Delete a view


```
use jspatel;

DROP VIEW reporting_view;

```



```
MariaDB [jspatel]> use jspatel; select * from daily_reporting
Database changed
    -> ;
+---------------------+----------------+----------+-----------+------+
| reporting_datetime  | machine_number | total_in | total_out | net  |
+---------------------+----------------+----------+-----------+------+
| 2024-03-27 10:00:00 |              2 |     NULL |      NULL |    0 |
+---------------------+----------------+----------+-----------+------+
1 row in set (0.001 sec)
```

3) Track and report the outcome from each of the SQL commands. The outcome to observe and report is if or when the changes from the MySQLWorkbench window become visible in the putty window, or if the changes never appear in the putty window.

   a. Create a table - After creating table in MySQLWorkbench, it will directly reflect in putty window.

   b. Create a row of data in a table - It will not reflect in putty window.

   c. Add a column to an existing table - After this query the putty window shows the data added in (b) and also the added column in (c).

   d. Update a row of data in a table - It will not reflect in putty window.

   e. Update the data type of a column in a table - After this query the putty window shows the data updated in (d) and also the updated data type of that particular column happened in (e).

   f. Delete a row in a table - It will not reflect in putty window.

   g. Delete a column in a table - After this query the putty window does not have the data deleted in (f) and the column deleted in (g).

h. Delete a table - After deleting table in MySQLWorkbench, it will directly reflect in putty window.

i. Create a view - After creating view in MySQLWorkbench, it will directly reflect in putty window.

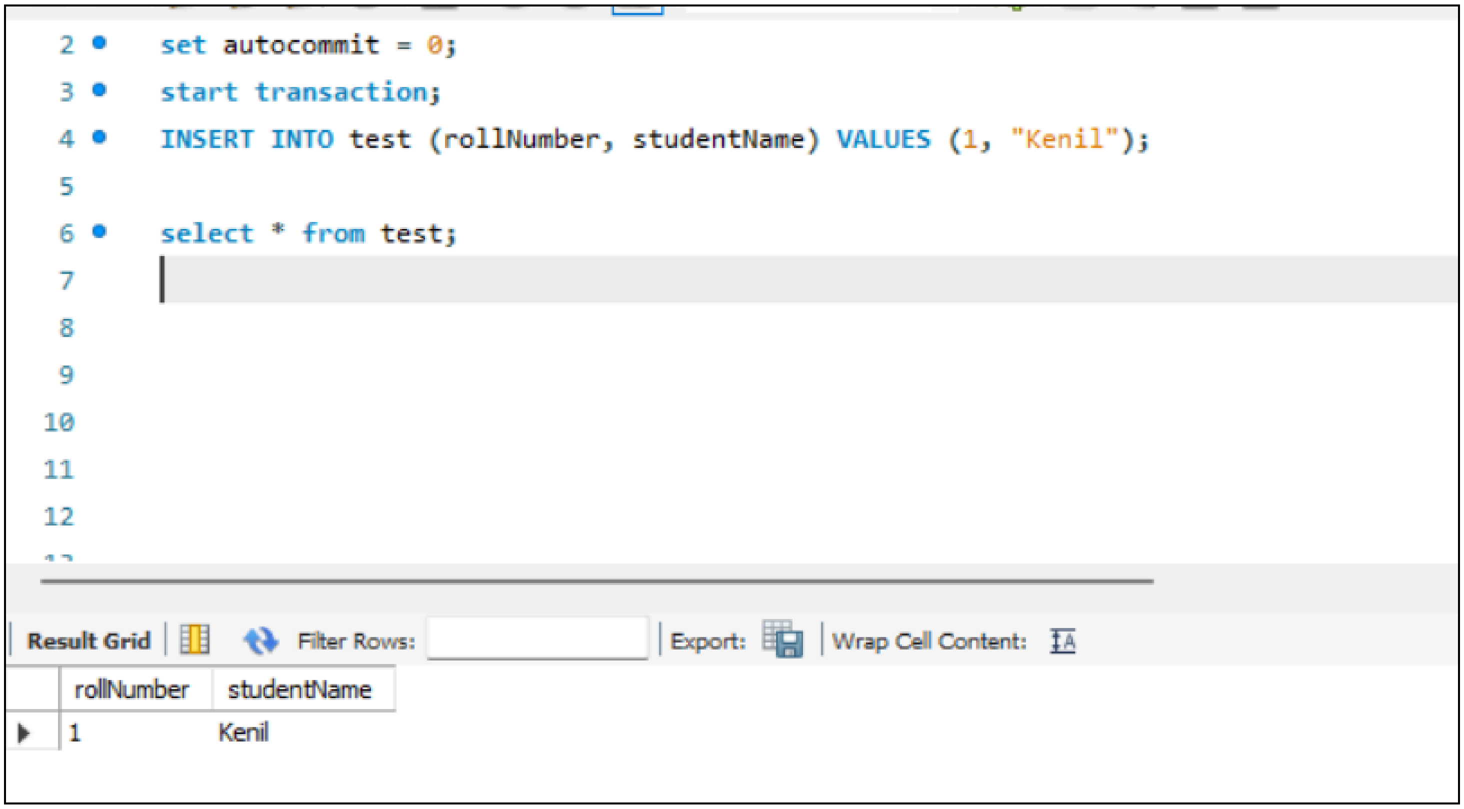
j. Add a row to a view - It will not reflect in putty window.

k. Update a row in a view - It will not reflect in putty window.

l. Delete a view - After deleting view in MySQLWorkbench, the update perform in view are now reflect in original table in putty window.

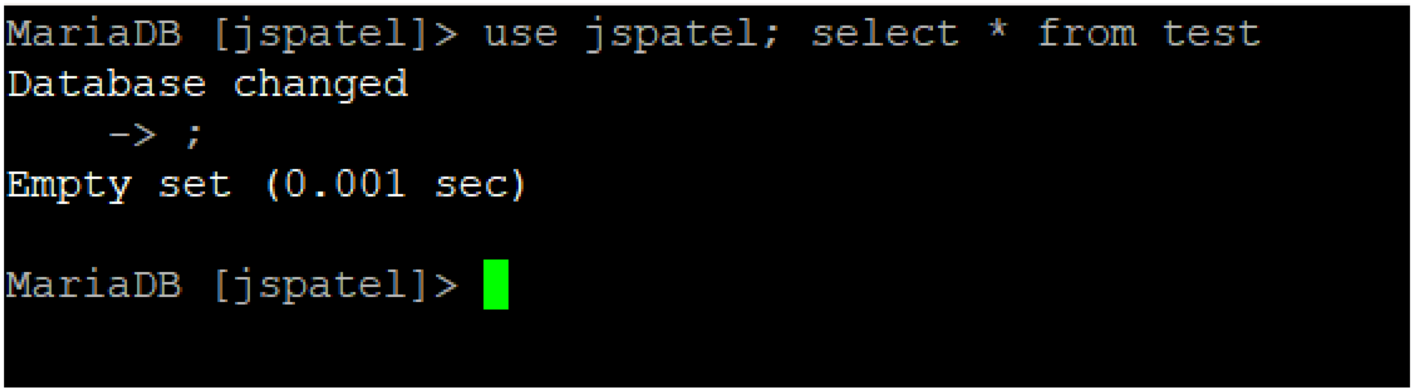
4) Describe a set of actions you can do to verify whether or not you can have a transaction inside another transaction. Try your set of actions and report on whether or not mysql will allow nested transactions.

Start First Transaction (Not Commit)



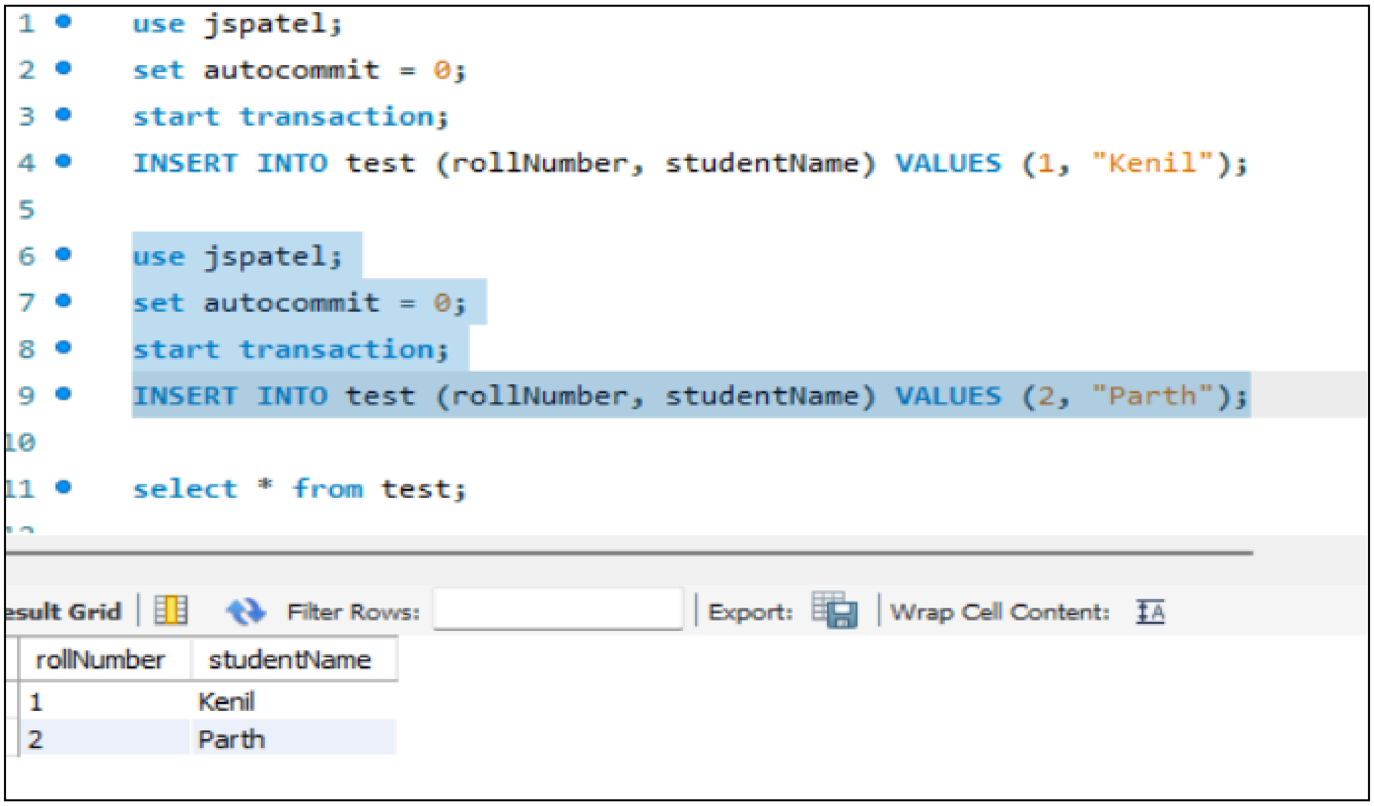
```
set autocommit = 0;
start transaction;
INSERT INTO test (rollNumber, studentName) VALUES (1, "Kenil");

select * from test;
```

```
rollNumber  studentName
1           Kenil
```



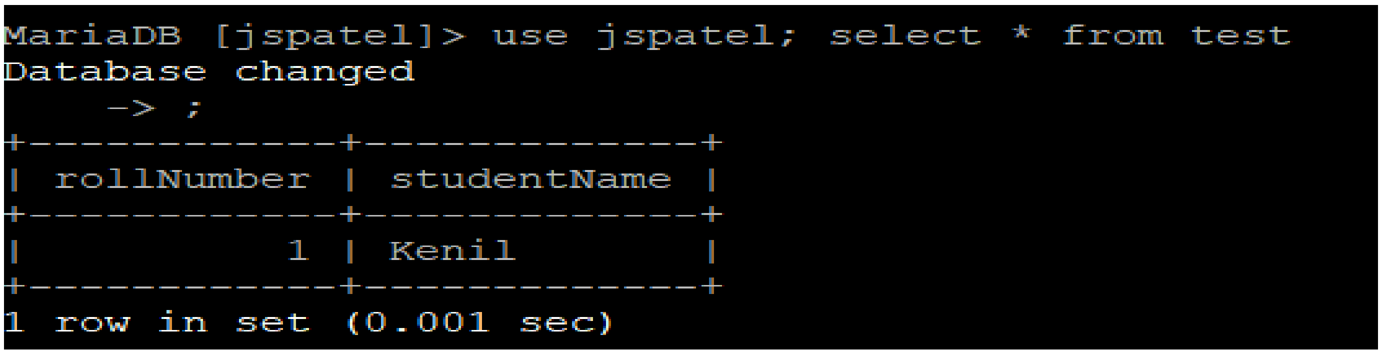
```
MariaDB [jspatel]> use jspatel; select * from test
Database changed
    -> ;
Empty set (0.001 sec)

MariaDB [jspatel]>
```

Start Second Transaction (After this second transaction, first transaction was automatically commited and we can see that first transaction data into putty window)



```
use jspatel;
set autocommit = 0;
start transaction;
INSERT INTO test (rollNumber, studentName) VALUES (1, "Kenil");

use jspatel;
set autocommit = 0;
start transaction;
INSERT INTO test (rollNumber, studentName) VALUES (2, "Parth");

select * from test;
```

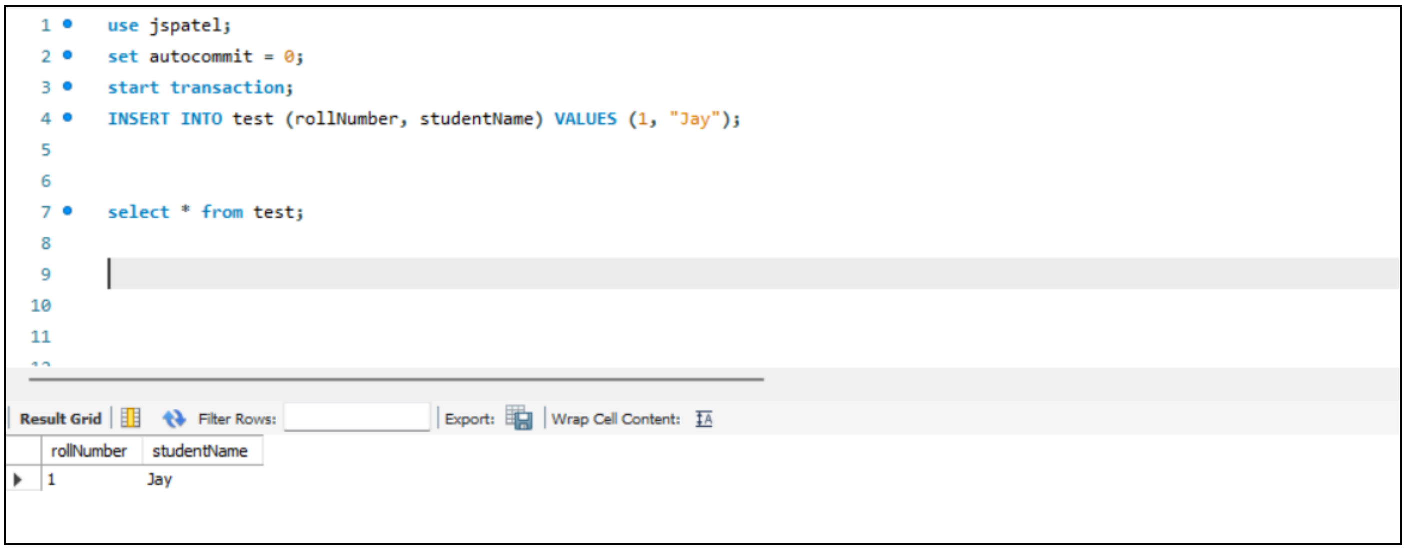
rollNumber	studentName
1	Kenil
2	Parth



```
MariaDB [jspatel]> use jspatel; select * from test
Database changed
    -> ;
+------------+-------------+
| rollNumber | studentName |
+------------+-------------+
|          1 | Kenil       |
+------------+-------------+
1 row in set (0.001 sec)
```

5) Determine whether a transaction will survive beyond your session:

a. Start a transaction in MySQLWorkbench and make a change to data in the table that you know the transaction will protect from other database users. Do not commit or roll-back the change.



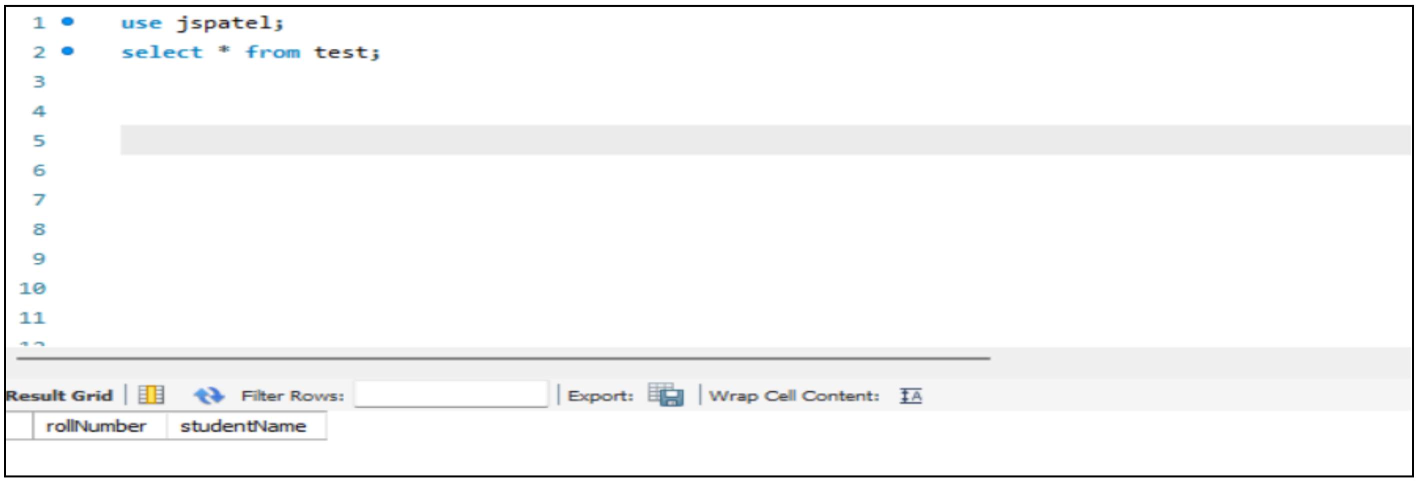
```
use jspatel;
set autocommit = 0;
start transaction;
INSERT INTO test (rollNumber, studentName) VALUES (1, "Jay");


select * from test;
```

rollNumber	studentName
1	Jay

b. Exit MySQLWorkbench

c. Restart MySQLWorkbench and return to your same test database. Remember to actually quit the application; don't just close the current query tab or window.

d. Report if your change remains visible in the database past the restart of MySQLWorkbench.



```
use jspatel;
select * from test;
```

rollNumber	studentName

e. Report on whether you can make further changes to the same data as in step 5a after the restart.

- After restarting MysqlWorkbench, we can’t have the data that we just added before closing it, so we can’t perform any operations on that data.

Part - 2

1) Characterize the kinds of changes that a transaction can let you recover from versus the changes that a transaction doesn’t let you recover from? Look for a characterization beyond itemizing the commands that do or don’t allow recovery.

    - Transactions in database management systems provide a means to ensure data integrity and consistency. It allow a sequence of database operations to be treated as a single logical unit, ensuring that either all operations within the transaction are completed successfully or none of them are.

    - Recoverable
        - ❖ Transactions can typically recover from partial failures, where only some operations within the transaction fail while others succeed.
        - ❖ Sequence of command within same transaction can be rollback if transaction is not committed.
        - ❖ Modern databases maintain logs of operations that occur within a particular transaction.In the event of a power failure or any other system failure, the system can recover that transaction from the log.
        - ❖ Only DML(Data Manipulation Language) commands are typically recoverable within a transaction in MySQL. This means that changes made by INSERT, UPDATE, and DELETE statements can be rolled back using the ROLLBACK command if the transaction is not committed.

- Not Recoverable
    - ❖ Transactions cannot recover changes that have been committed by other transactions.
    - ❖ Transactions do not protect against lost updates, where one transaction overwrites changes made by another transaction without being aware of those changes.
    - ❖ DDL (Data Definition Language) and DCL(Data Control Language) commands are not recoverable within a transaction and are automatically committed, meaning their effects cannot be undone by a rollback operation.

2) If nested transactions are supported in a DBMS, describe any deadlock conditions that must be managed.

- Circular Dependency within Nested Transactions.
  For example, Transaction A at the outer level might hold a lock on Resource X, while Transaction B within Transaction A holds a lock on Resource Y that Transaction C at the outer level needs.
- Child Transactions Holding Locks Needed by Parent Transactions.
- Inconsistent Locking and Unlocking Order: For instance, if a child transaction releases a lock before its parent transaction, and the parent transaction is waiting for that lock, it can result in deadlock.
- Inappropriate Locking and Unlocking during Rollback Operations.