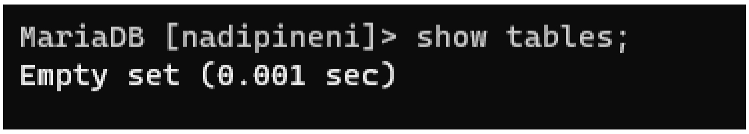
CSCI 3901 LAB ASSIGNMENT - 8

Instructor: Dr. Michael McAllister
Name: Nadipineni Hemanth Kumar
Banner ID: B00899473
Date: 2021-11-27

1) The outcomes of each test operation in the lab steps:

BEFORE CREATING TABLES:



```
MariaDB [nadipineni]> show tables;
Empty set (0.001 sec)
```

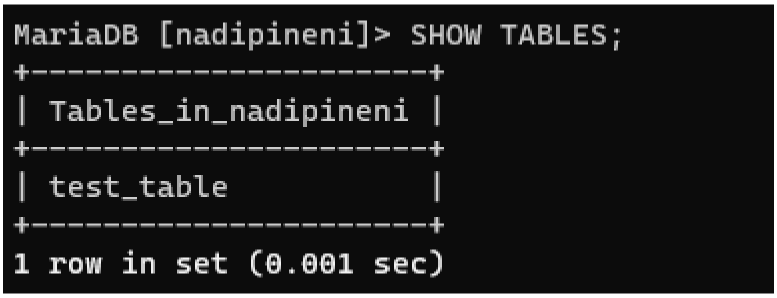
All the transactions below are captured in the order of 1) Before Transaction 2) After Transaction 2.1)Before Rollback 2.2)After Rollback.

I am not giving more details for each of the lab steps as it would be redundant and may confuse the above order.

A)CREATE A TABLE:

Before ROLLBACK:

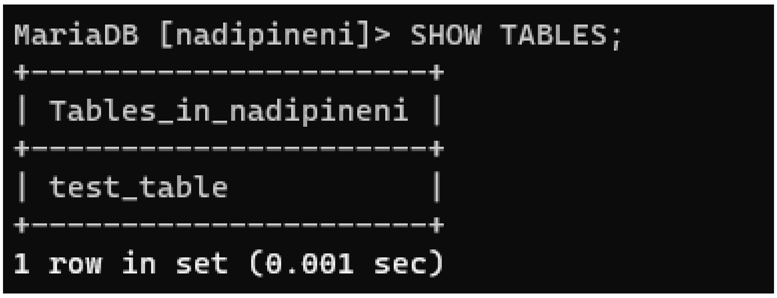
- `START TRANSACTION;`
- `CREATE TABLE test_table (id int not null auto_increment primary key, name varchar(50), age int);`



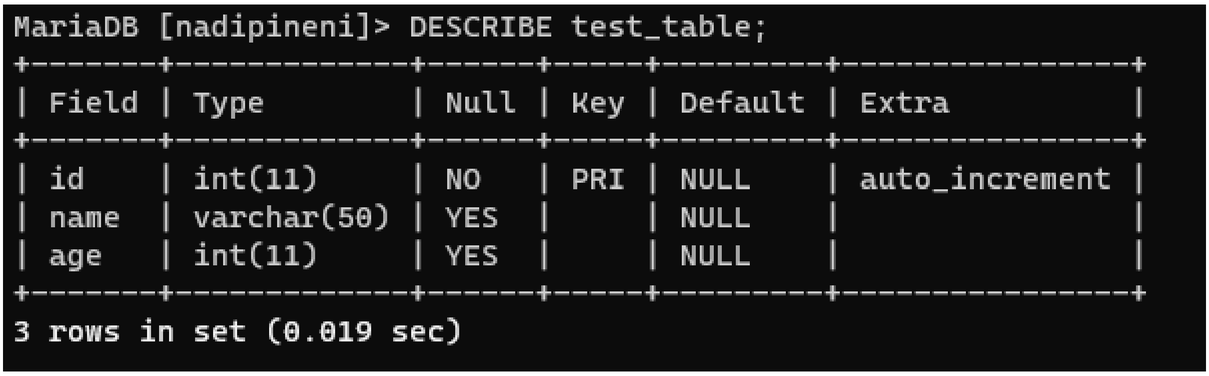
```
MariaDB [nadipineni]> SHOW TABLES;
+-----------------------+
| Tables_in_nadipineni |
+-----------------------+
| test_table            |
+-----------------------+
1 row in set (0.001 sec)
```

After ROLLBACK:

- `START TRANSACTION;`
- `CREATE TABLE test_table (id int not null auto_increment primary key, name varchar(50), age int);`
- `ROLLBACK;`



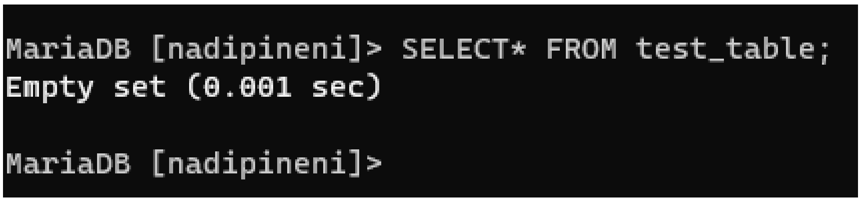
```
MariaDB [nadipineni]> SHOW TABLES;
+------------------------+
| Tables_in_nadipineni |
+------------------------+
| test_table             |
+------------------------+
1 row in set (0.001 sec)
```



```
MariaDB [nadipineni]> DESCRIBE test_table;
+-------+-------------+------+-----+---------+----------------+
| Field | Type        | Null | Key | Default | Extra          |
+-------+-------------+------+-----+---------+----------------+
| id    | int(11)     | NO   | PRI | NULL    | auto_increment |
| name  | varchar(50) | YES  |     | NULL    |                |
| age   | int(11)     | YES  |     | NULL    |                |
+-------+-------------+------+-----+---------+----------------+
3 rows in set (0.019 sec)
```

B)CREATE A ROW OF DATA IN A TABLE:

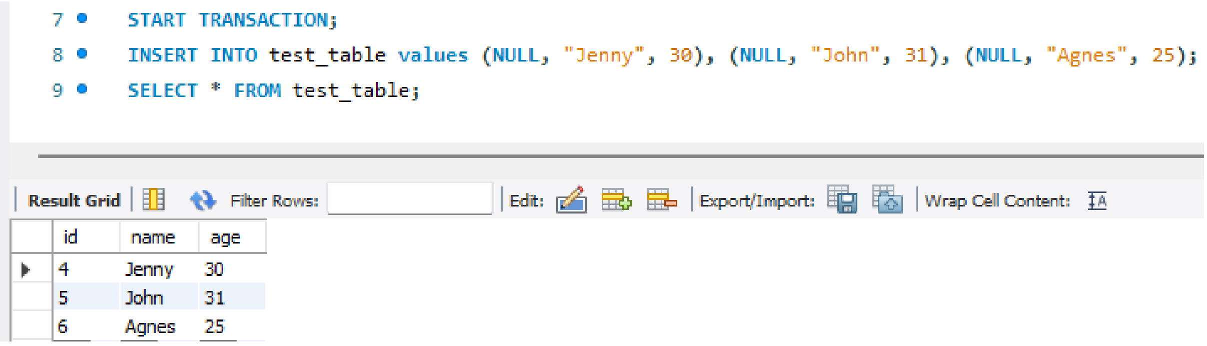
BEFORE TRANSACTION:



```
MariaDB [nadipineni]> SELECT* FROM test_table;
Empty set (0.001 sec)

MariaDB [nadipineni]>
```

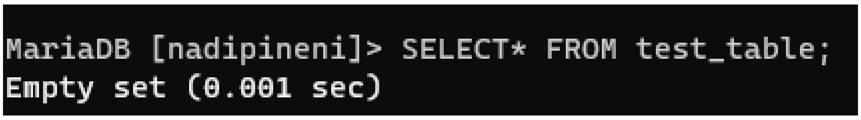
AFTER TRANSACTION:

- BEFORE ROLLBACK:



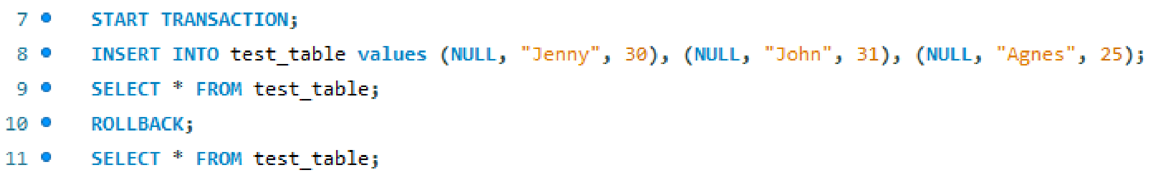
```
START TRANSACTION;
INSERT INTO test_table values (NULL, "Jenny", 30), (NULL, "John", 31), (NULL, "Agnes", 25);
SELECT * FROM test_table;
```

id	name	age
4	Jenny	30
5	John	31
6	Agnes	25

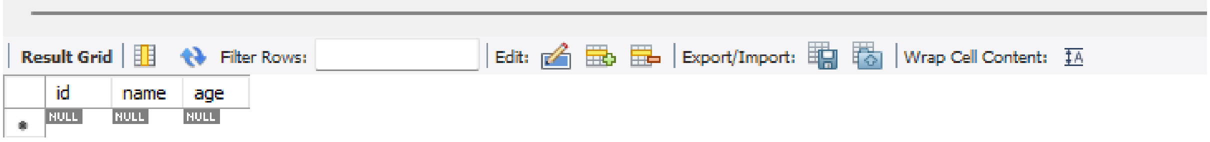


```
MariaDB [nadipineni]> SELECT* FROM test_table;
Empty set (0.001 sec)
```

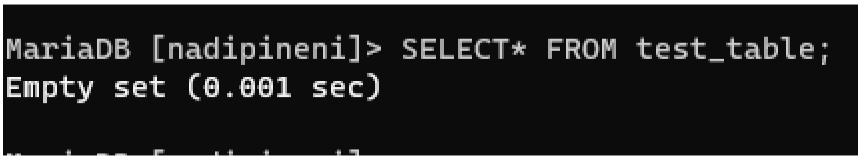
- AFTER ROLLBACK:



```sql
START TRANSACTION;
INSERT INTO test_table values (NULL, "Jenny", 30), (NULL, "John", 31), (NULL, "Agnes", 25);
SELECT * FROM test_table;
ROLLBACK;
SELECT * FROM test_table;
```



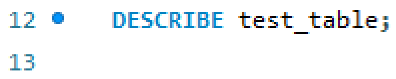
	id	name	age
*	NULL	NULL	NULL



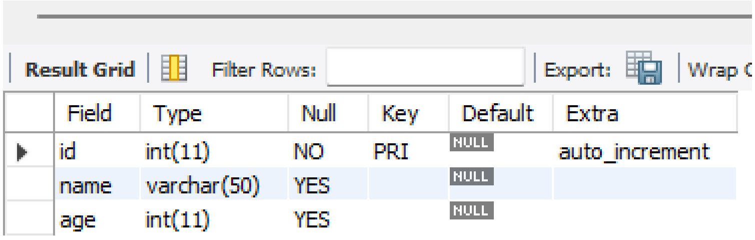
```
MariaDB [nadipineni]> SELECT* FROM test_table;
Empty set (0.001 sec)
```

C)ADD A COLUMN TO AN EXISTING TABLE:

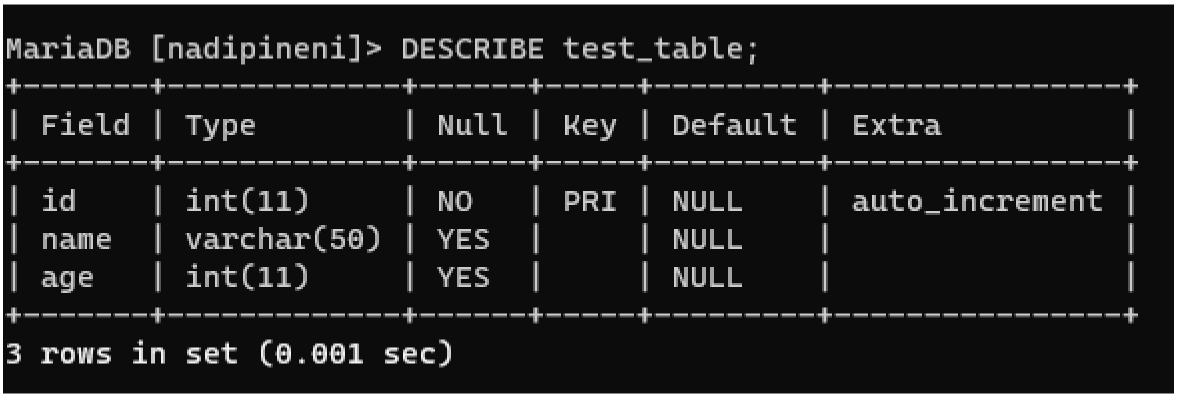
BEFORE TRANASCTION:



```sql
DESCRIBE test_table;
```



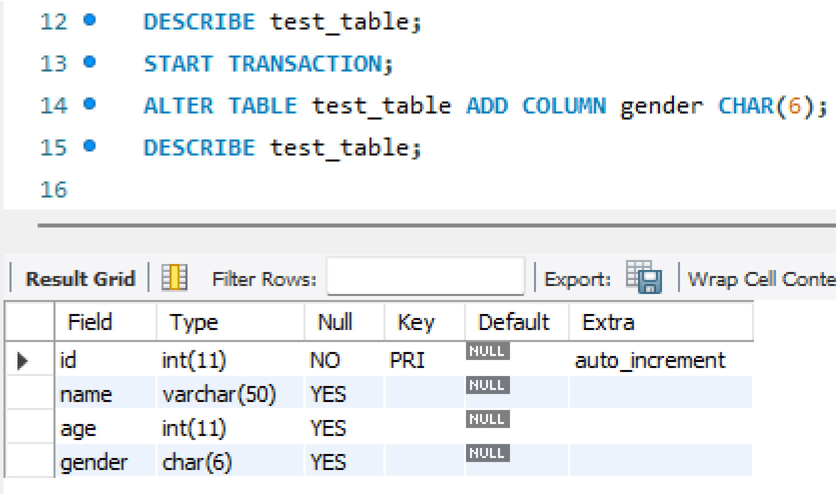
	Field	Type	Null	Key	Default	Extra
▶	id	int(11)	NO	PRI	NULL	auto_increment
	name	varchar(50)	YES		NULL	
	age	int(11)	YES		NULL	



```
MariaDB [nadipineni]> DESCRIBE test_table;
+-------+-------------+------+-----+---------+----------------+
| Field | Type        | Null | Key | Default | Extra          |
+-------+-------------+------+-----+---------+----------------+
| id    | int(11)     | NO   | PRI | NULL    | auto_increment |
| name  | varchar(50) | YES  |     | NULL    |                |
| age   | int(11)     | YES  |     | NULL    |                |
+-------+-------------+------+-----+---------+----------------+
3 rows in set (0.001 sec)
```

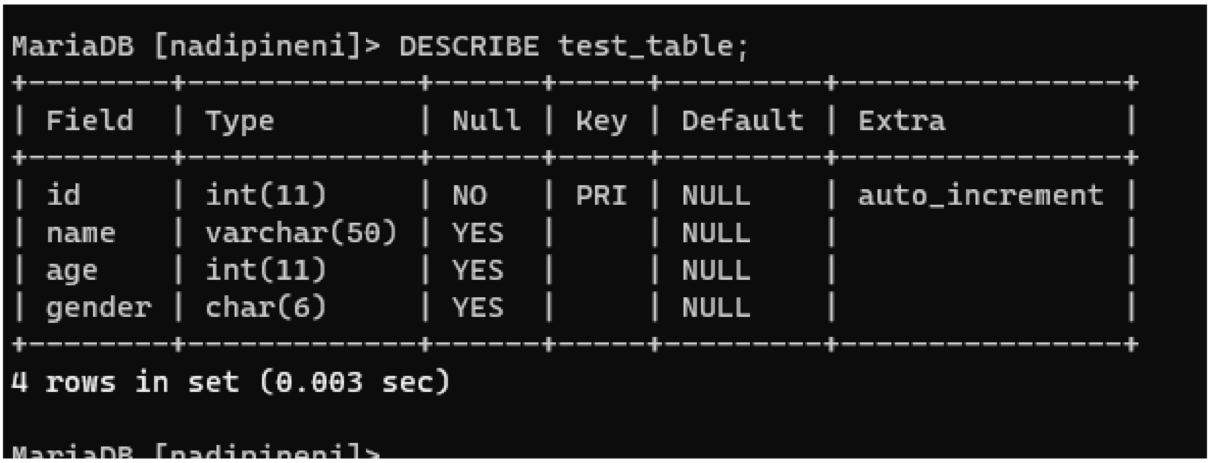
AFTER TRANSACTION:

- BEFORE ROLLBACK:



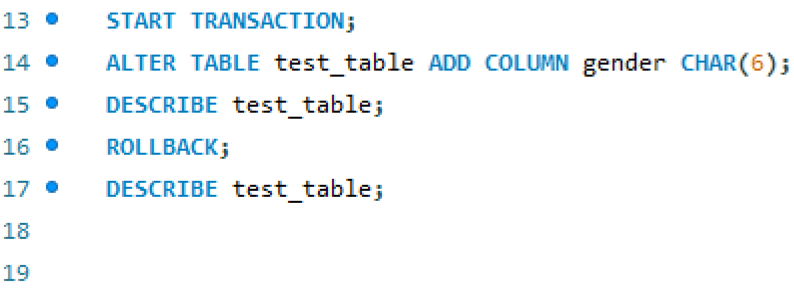
```sql
DESCRIBE test_table;
START TRANSACTION;
ALTER TABLE test_table ADD COLUMN gender CHAR(6);
DESCRIBE test_table;

```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(50)	YES		NULL	
age	int(11)	YES		NULL	
gender	char(6)	YES		NULL	

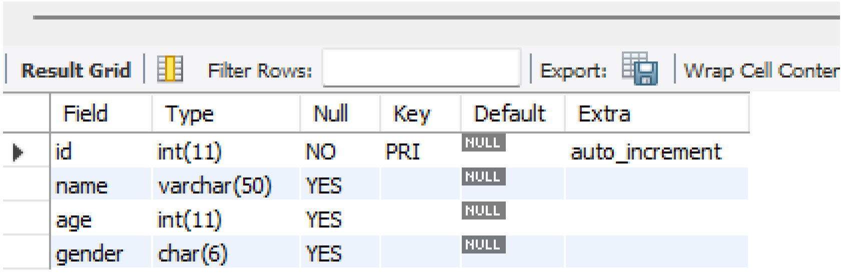


```
MariaDB [nadipineni]> DESCRIBE test_table;
+--------+-------------+------+-----+---------+----------------+
| Field  | Type        | Null | Key | Default | Extra          |
+--------+-------------+------+-----+---------+----------------+
| id     | int(11)     | NO   | PRI | NULL    | auto_increment |
| name   | varchar(50) | YES  |     | NULL    |                |
| age    | int(11)     | YES  |     | NULL    |                |
| gender | char(6)     | YES  |     | NULL    |                |
+--------+-------------+------+-----+---------+----------------+
4 rows in set (0.003 sec)

MariaDB [nadipineni]>
```

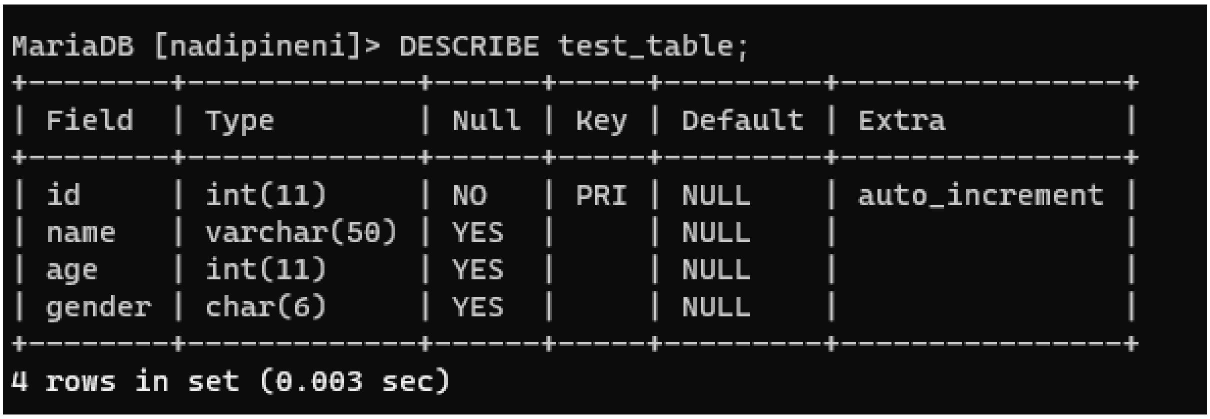
- AFTER ROLLBACK:



```sql
START TRANSACTION;
ALTER TABLE test_table ADD COLUMN gender CHAR(6);
DESCRIBE test_table;
ROLLBACK;
DESCRIBE test_table;


```



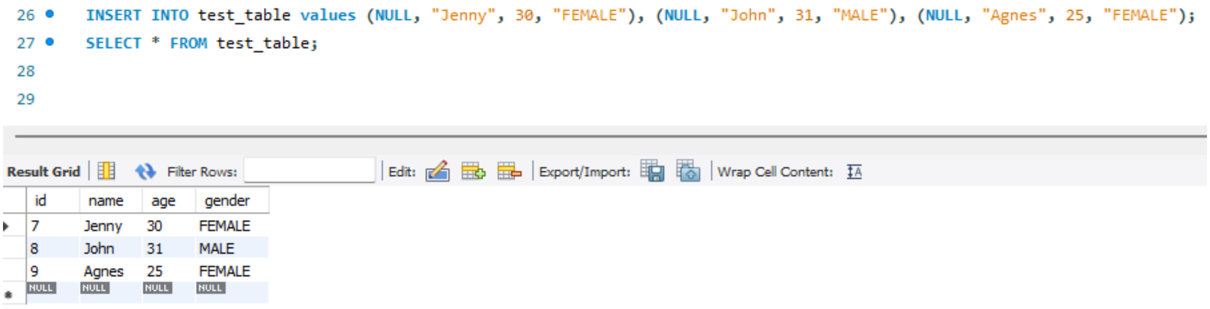
Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(50)	YES		NULL	
age	int(11)	YES		NULL	
gender	char(6)	YES		NULL	



```
MariaDB [nadipineni]> DESCRIBE test_table;
+--------+-------------+------+-----+---------+----------------+
| Field  | Type        | Null | Key | Default | Extra          |
+--------+-------------+------+-----+---------+----------------+
| id     | int(11)     | NO   | PRI | NULL    | auto_increment |
| name   | varchar(50) | YES  |     | NULL    |                |
| age    | int(11)     | YES  |     | NULL    |                |
| gender | char(6)     | YES  |     | NULL    |                |
+--------+-------------+------+-----+---------+----------------+
4 rows in set (0.003 sec)
```

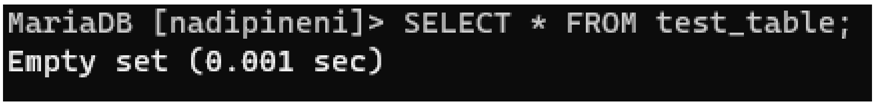
D)UPDATE A ROW OF DATA IN A TABLE:

BEFORE TRANSACTION:



```
INSERT INTO test_table values (NULL, "Jenny", 30, "FEMALE"), (NULL, "John", 31, "MALE"), (NULL, "Agnes", 25, "FEMALE");
SELECT * FROM test_table;
```

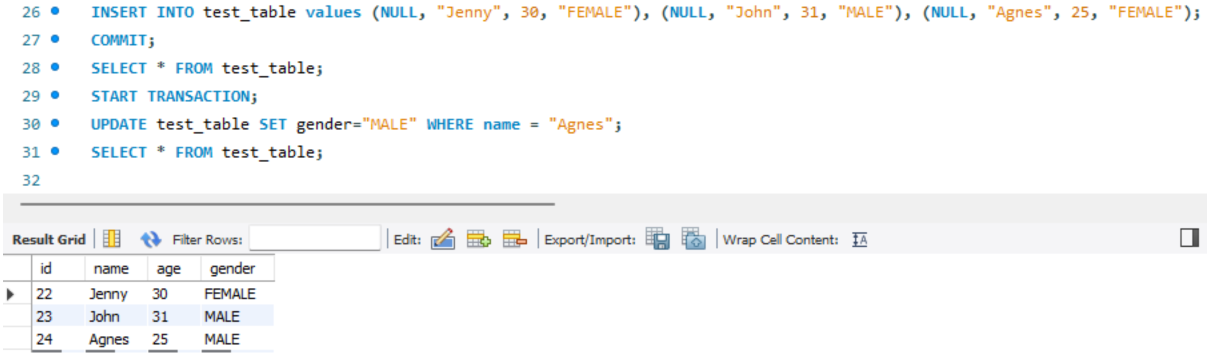
id	name	age	gender
7	Jenny	30	FEMALE
8	John	31	MALE
9	Agnes	25	FEMALE
NULL	NULL	NULL	NULL



```
MariaDB [nadipineni]> SELECT * FROM test_table;
Empty set (0.001 sec)
```

AFTER TRANSACTION:

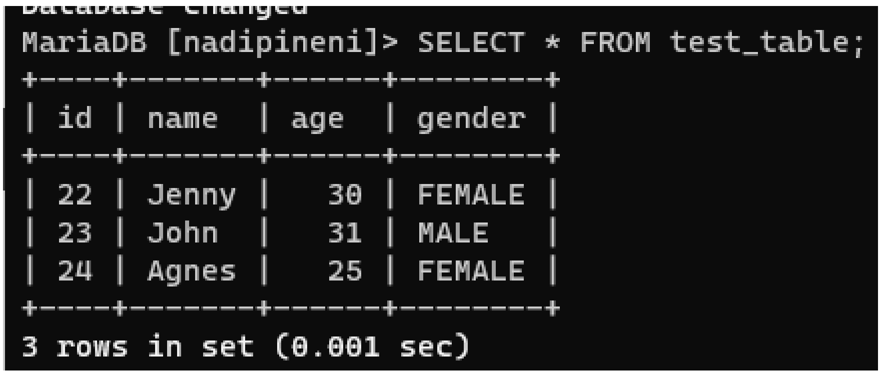
- BEFORE ROLLBACK:



```
INSERT INTO test_table values (NULL, "Jenny", 30, "FEMALE"), (NULL, "John", 31, "MALE"), (NULL, "Agnes", 25, "FEMALE");
COMMIT;
SELECT * FROM test_table;
START TRANSACTION;
UPDATE test_table SET gender="MALE" WHERE name = "Agnes";
SELECT * FROM test_table;
```

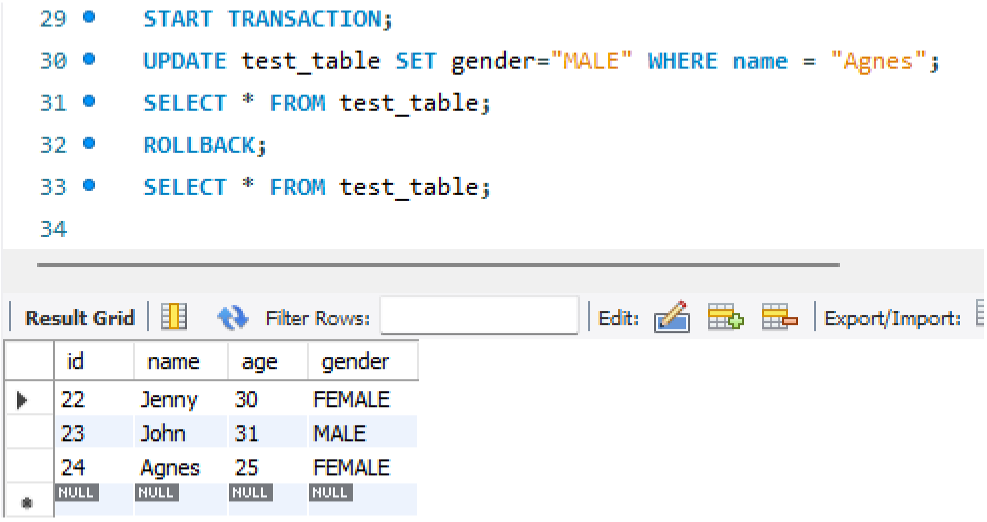
id	name	age	gender
22	Jenny	30	FEMALE
23	John	31	MALE
24	Agnes	25	MALE

(UPDATE doesn’t change putty version because I didn’t commit that)



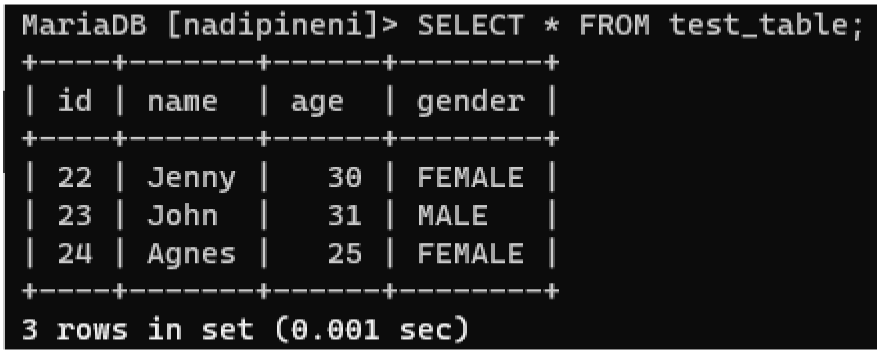
```
Database changed
MariaDB [nadipineni]> SELECT * FROM test_table;
+----+-------+------+--------+
| id | name  | age  | gender |
+----+-------+------+--------+
| 22 | Jenny |   30 | FEMALE |
| 23 | John  |   31 | MALE   |
| 24 | Agnes |   25 | FEMALE |
+----+-------+------+--------+
3 rows in set (0.001 sec)
```

- AFTER ROLLBACK:



```
START TRANSACTION;
UPDATE test_table SET gender="MALE" WHERE name = "Agnes";
SELECT * FROM test_table;
ROLLBACK;
SELECT * FROM test_table;
```

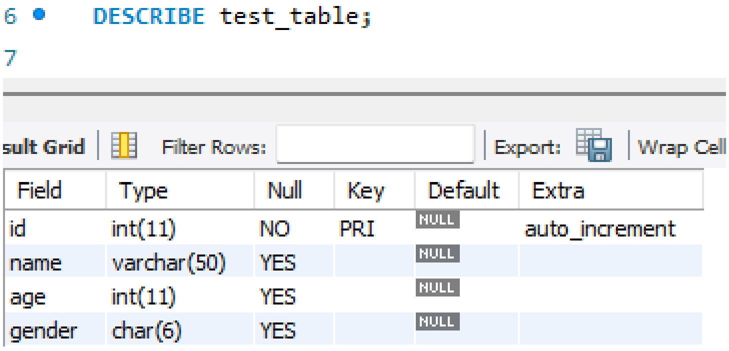
id	name	age	gender
22	Jenny	30	FEMALE
23	John	31	MALE
24	Agnes	25	FEMALE
NULL	NULL	NULL	NULL



```
MariaDB [nadipineni]> SELECT * FROM test_table;
+----+-------+------+--------+
| id | name  | age  | gender |
+----+-------+------+--------+
| 22 | Jenny |   30 | FEMALE |
| 23 | John  |   31 | MALE   |
| 24 | Agnes |   25 | FEMALE |
+----+-------+------+--------+
3 rows in set (0.001 sec)
```

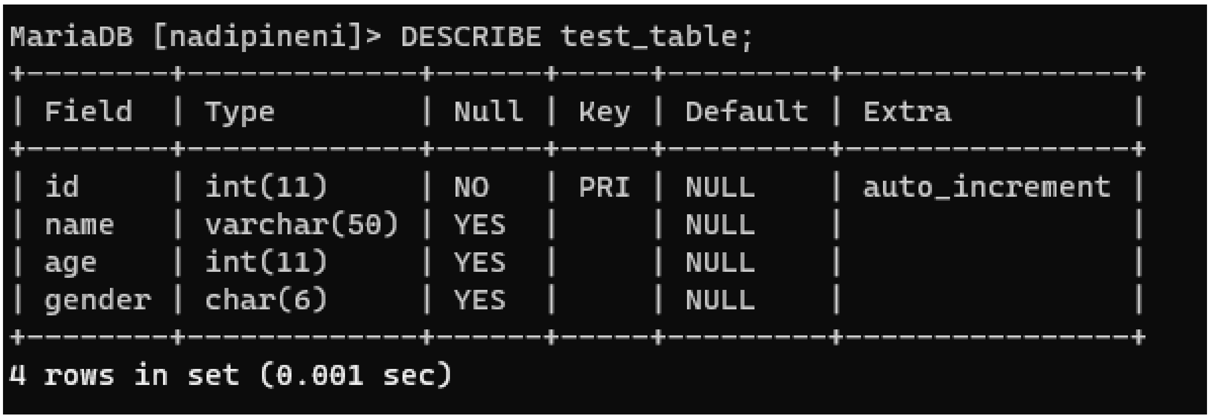
E)UPDATE THE DATA TYPE OF A COLUMN IN A TABLE:

BEFORE TRANSACTION:



```
DESCRIBE test_table;
```

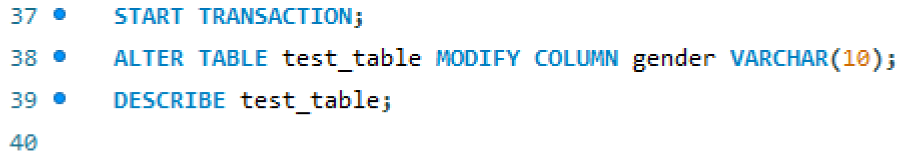
Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(50)	YES		NULL	
age	int(11)	YES		NULL	
gender	char(6)	YES		NULL	



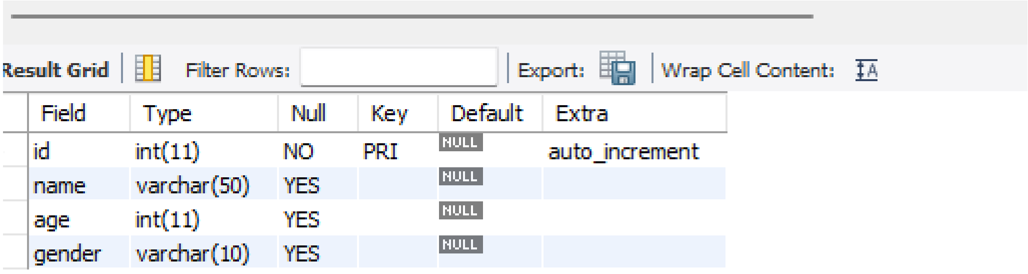
```
MariaDB [nadipineni]> DESCRIBE test_table;
+--------+-------------+------+-----+---------+----------------+
| Field  | Type        | Null | Key | Default | Extra          |
+--------+-------------+------+-----+---------+----------------+
| id     | int(11)     | NO   | PRI | NULL    | auto_increment |
| name   | varchar(50) | YES  |     | NULL    |                |
| age    | int(11)     | YES  |     | NULL    |                |
| gender | char(6)     | YES  |     | NULL    |                |
+--------+-------------+------+-----+---------+----------------+
4 rows in set (0.001 sec)
```

AFTER TRANSACTION:

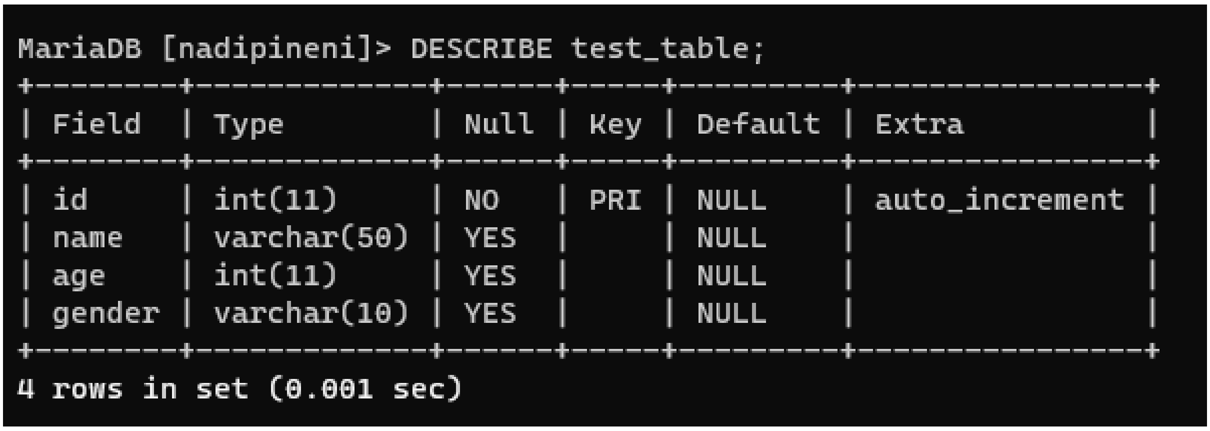
- BEFORE ROLLBACK:



```
START TRANSACTION;
ALTER TABLE test_table MODIFY COLUMN gender VARCHAR(10);
DESCRIBE test_table;
```

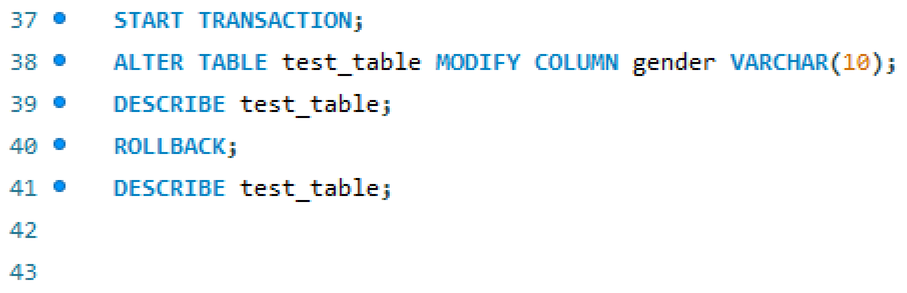


Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(50)	YES		NULL	
age	int(11)	YES		NULL	
gender	varchar(10)	YES		NULL	

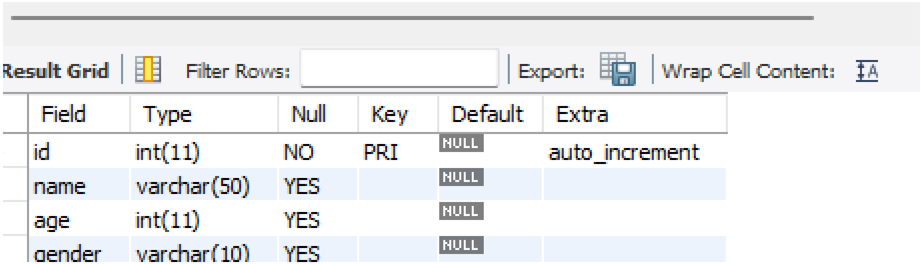


```
MariaDB [nadipineni]> DESCRIBE test_table;
+--------+-------------+------+-----+---------+----------------+
| Field  | Type        | Null | Key | Default | Extra          |
+--------+-------------+------+-----+---------+----------------+
| id     | int(11)     | NO   | PRI | NULL    | auto_increment |
| name   | varchar(50) | YES  |     | NULL    |                |
| age    | int(11)     | YES  |     | NULL    |                |
| gender | varchar(10) | YES  |     | NULL    |                |
+--------+-------------+------+-----+---------+----------------+
4 rows in set (0.001 sec)
```

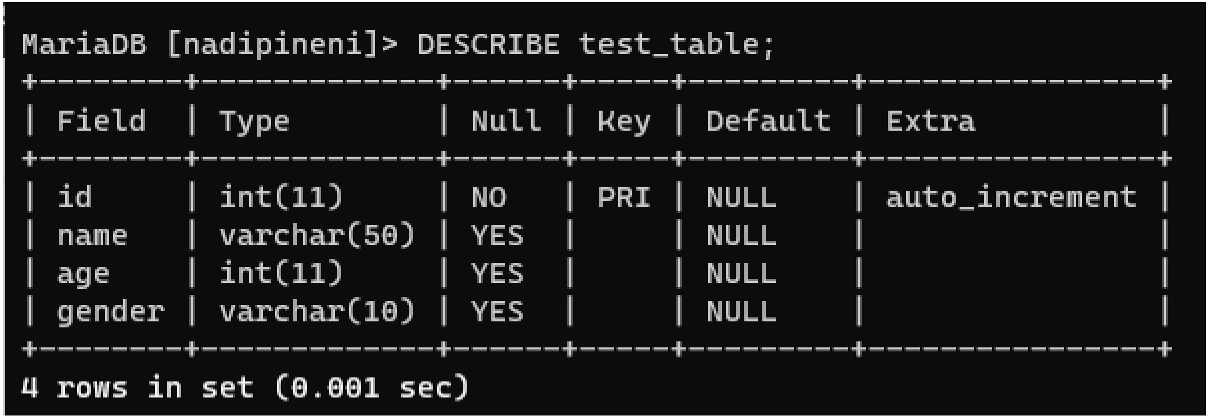
- AFTER ROLLBACK:



```
START TRANSACTION;
ALTER TABLE test_table MODIFY COLUMN gender VARCHAR(10);
DESCRIBE test_table;
ROLLBACK;
DESCRIBE test_table;
```



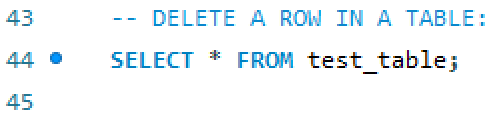
Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(50)	YES		NULL	
age	int(11)	YES		NULL	
gender	varchar(10)	YES		NULL	



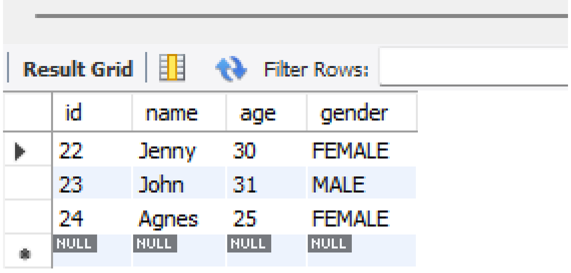
```
MariaDB [nadipineni]> DESCRIBE test_table;
+--------+-------------+------+-----+---------+----------------+
| Field  | Type        | Null | Key | Default | Extra          |
+--------+-------------+------+-----+---------+----------------+
| id     | int(11)     | NO   | PRI | NULL    | auto_increment |
| name   | varchar(50) | YES  |     | NULL    |                |
| age    | int(11)     | YES  |     | NULL    |                |
| gender | varchar(10) | YES  |     | NULL    |                |
+--------+-------------+------+-----+---------+----------------+
4 rows in set (0.001 sec)
```

F)DELETE A ROW IN A TABLE:

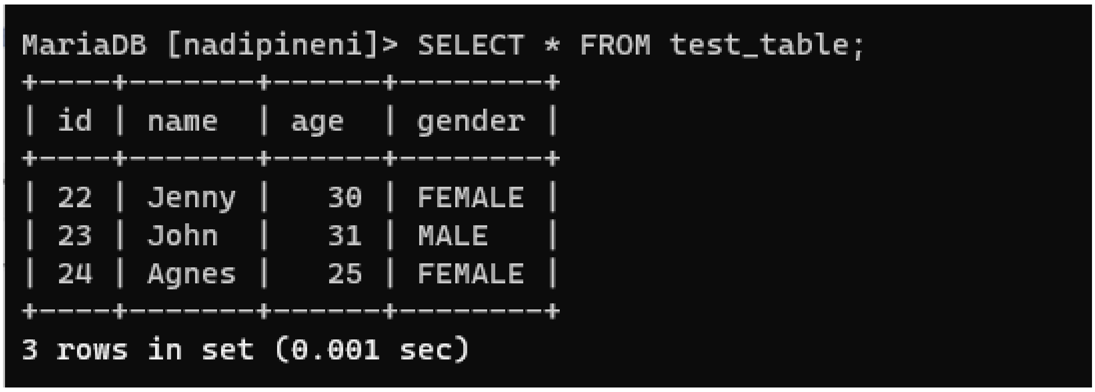
BEFORE TRANSACTION:



```
-- DELETE A ROW IN A TABLE:
SELECT * FROM test_table;
```



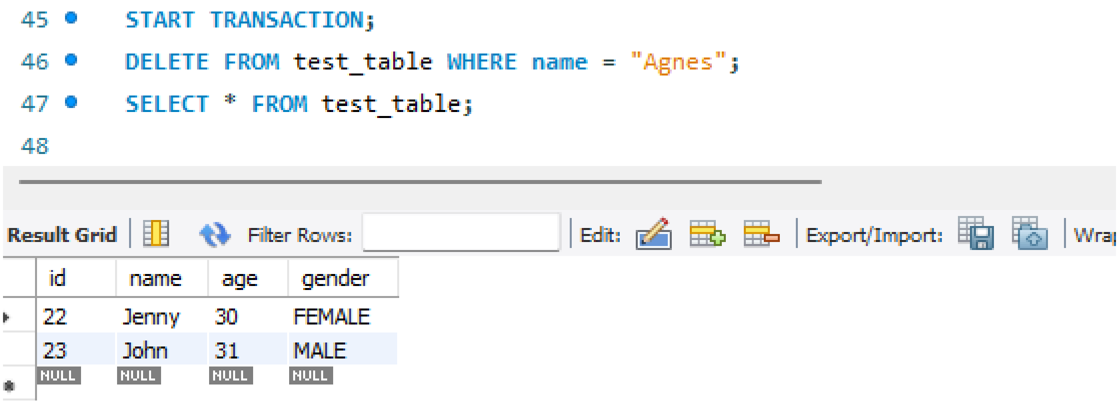
id	name	age	gender
22	Jenny	30	FEMALE
23	John	31	MALE
24	Agnes	25	FEMALE
NULL	NULL	NULL	NULL



```
MariaDB [nadipineni]> SELECT * FROM test_table;
+----+-------+------+--------+
| id | name  | age  | gender |
+----+-------+------+--------+
| 22 | Jenny |   30 | FEMALE |
| 23 | John  |   31 | MALE   |
| 24 | Agnes |   25 | FEMALE |
+----+-------+------+--------+
3 rows in set (0.001 sec)
```

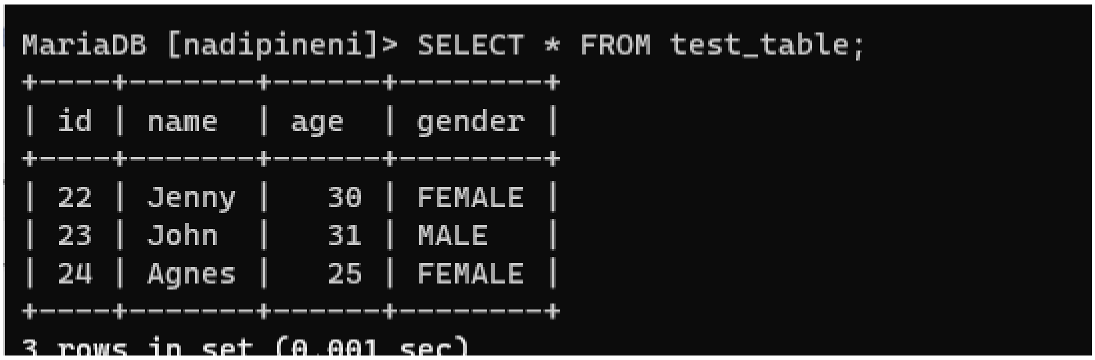
AFTER TRANSACTION:

- BEFORE ROLLBACK



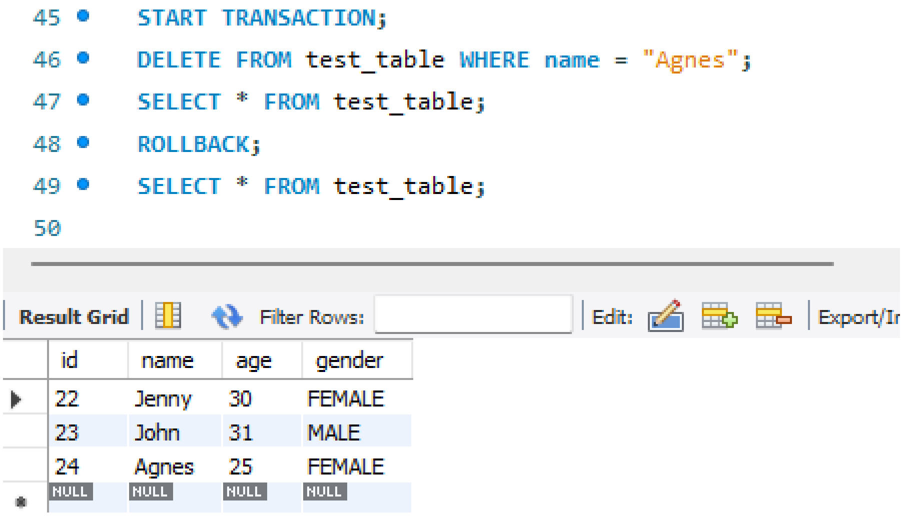
```
START TRANSACTION;
DELETE FROM test_table WHERE name = "Agnes";
SELECT * FROM test_table;
```

id	name	age	gender
22	Jenny	30	FEMALE
23	John	31	MALE
NULL	NULL	NULL	NULL



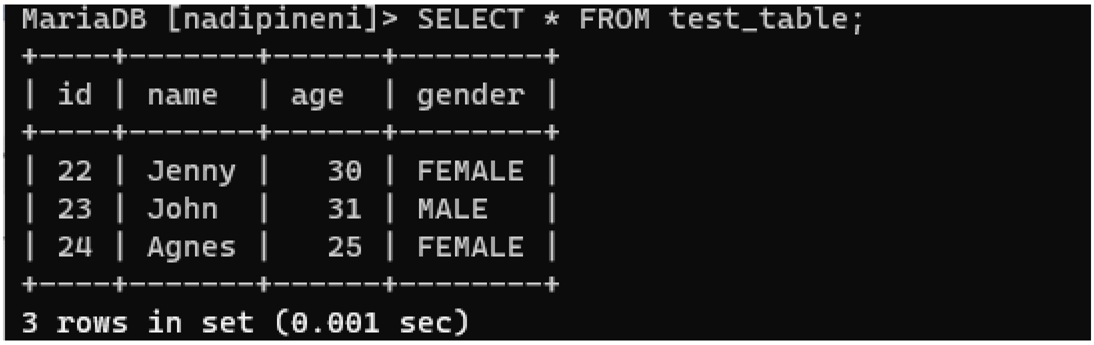
```
MariaDB [nadipineni]> SELECT * FROM test_table;
+----+-------+------+--------+
| id | name  | age  | gender |
+----+-------+------+--------+
| 22 | Jenny |   30 | FEMALE |
| 23 | John  |   31 | MALE   |
| 24 | Agnes |   25 | FEMALE |
+----+-------+------+--------+
3 rows in set (0.001 sec)
```

- AFTER ROLLBACK:



```
START TRANSACTION;
DELETE FROM test_table WHERE name = "Agnes";
SELECT * FROM test_table;
ROLLBACK;
SELECT * FROM test_table;
```

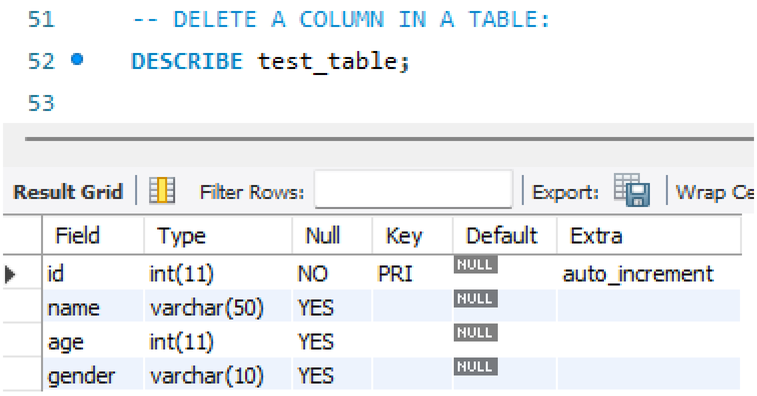
id	name	age	gender
22	Jenny	30	FEMALE
23	John	31	MALE
24	Agnes	25	FEMALE
NULL	NULL	NULL	NULL



```
MariaDB [nadipineni]> SELECT * FROM test_table;
+----+-------+------+--------+
| id | name  | age  | gender |
+----+-------+------+--------+
| 22 | Jenny |   30 | FEMALE |
| 23 | John  |   31 | MALE   |
| 24 | Agnes |   25 | FEMALE |
+----+-------+------+--------+
3 rows in set (0.001 sec)
```

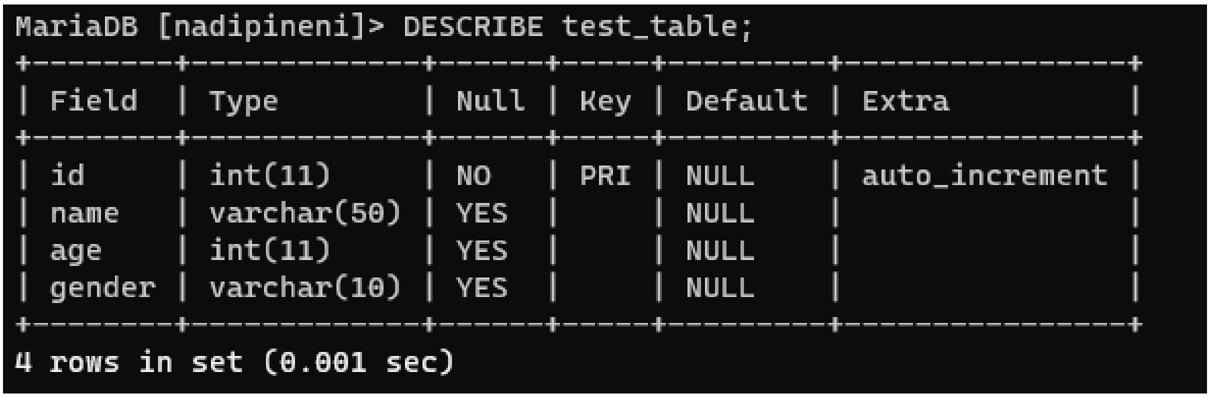
G)DELETE A COLUMN IN A TABLE:

BEFORE TRANSACTION:



```
-- DELETE A COLUMN IN A TABLE:
DESCRIBE test_table;
```

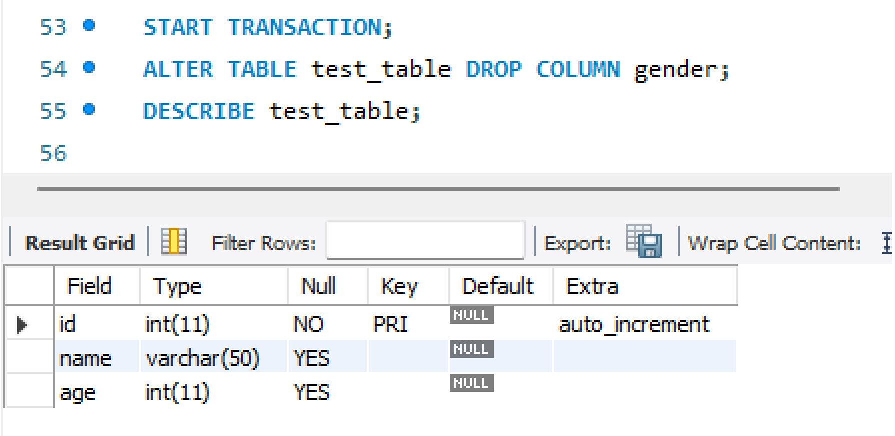
Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(50)	YES		NULL	
age	int(11)	YES		NULL	
gender	varchar(10)	YES		NULL	



```
MariaDB [nadipineni]> DESCRIBE test_table;
+--------+-------------+------+-----+---------+----------------+
| Field  | Type        | Null | Key | Default | Extra          |
+--------+-------------+------+-----+---------+----------------+
| id     | int(11)     | NO   | PRI | NULL    | auto_increment |
| name   | varchar(50) | YES  |     | NULL    |                |
| age    | int(11)     | YES  |     | NULL    |                |
| gender | varchar(10) | YES  |     | NULL    |                |
+--------+-------------+------+-----+---------+----------------+
4 rows in set (0.001 sec)
```

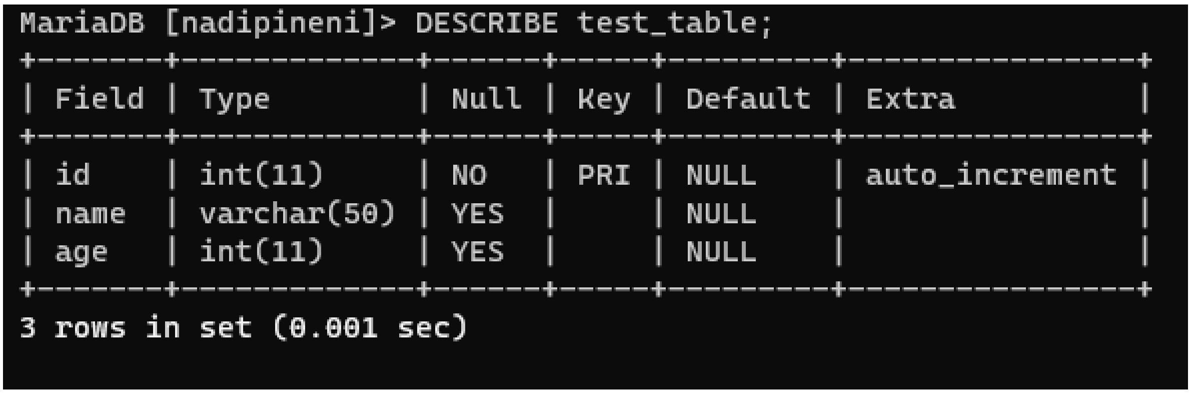
AFTER TRANSACTION:

- BEFORE ROLLBACK:



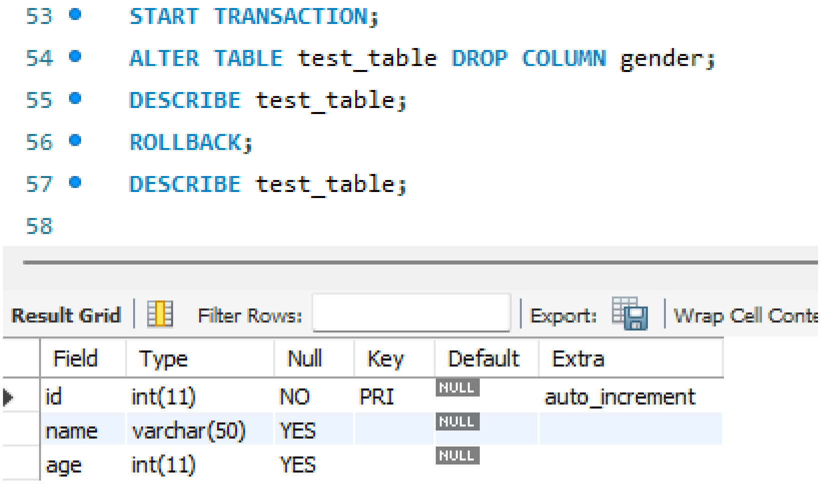
```
START TRANSACTION;
ALTER TABLE test_table DROP COLUMN gender;
DESCRIBE test_table;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(50)	YES		NULL	
age	int(11)	YES		NULL	



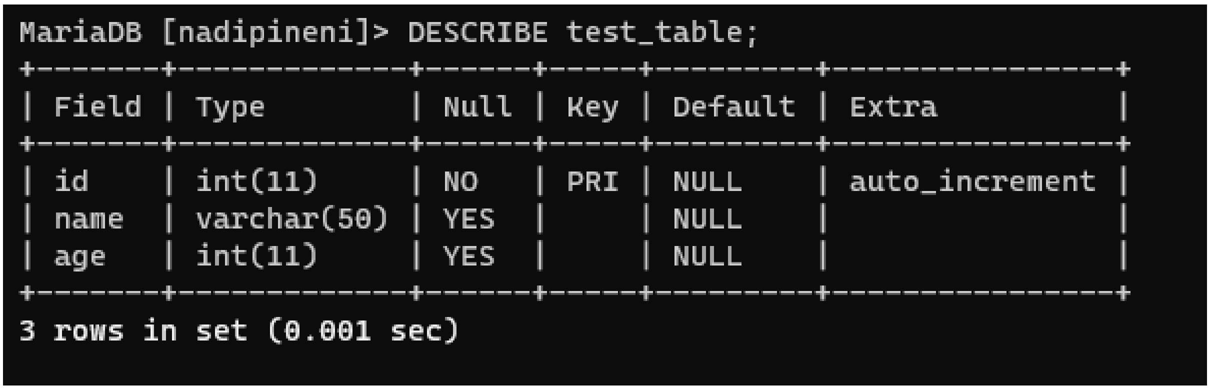
```
MariaDB [nadipineni]> DESCRIBE test_table;
+-------+-------------+------+-----+---------+----------------+
| Field | Type        | Null | Key | Default | Extra          |
+-------+-------------+------+-----+---------+----------------+
| id    | int(11)     | NO   | PRI | NULL    | auto_increment |
| name  | varchar(50) | YES  |     | NULL    |                |
| age   | int(11)     | YES  |     | NULL    |                |
+-------+-------------+------+-----+---------+----------------+
3 rows in set (0.001 sec)
```

- AFTER ROLLBACK:



```
START TRANSACTION;
ALTER TABLE test_table DROP COLUMN gender;
DESCRIBE test_table;
ROLLBACK;
DESCRIBE test_table;
```

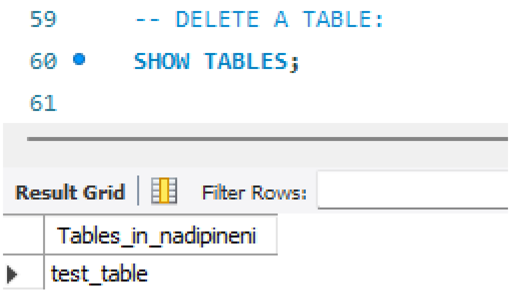
Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(50)	YES		NULL	
age	int(11)	YES		NULL	



```
MariaDB [nadipineni]> DESCRIBE test_table;
+-------+-------------+------+-----+---------+----------------+
| Field | Type        | Null | Key | Default | Extra          |
+-------+-------------+------+-----+---------+----------------+
| id    | int(11)     | NO   | PRI | NULL    | auto_increment |
| name  | varchar(50) | YES  |     | NULL    |                |
| age   | int(11)     | YES  |     | NULL    |                |
+-------+-------------+------+-----+---------+----------------+
3 rows in set (0.001 sec)
```

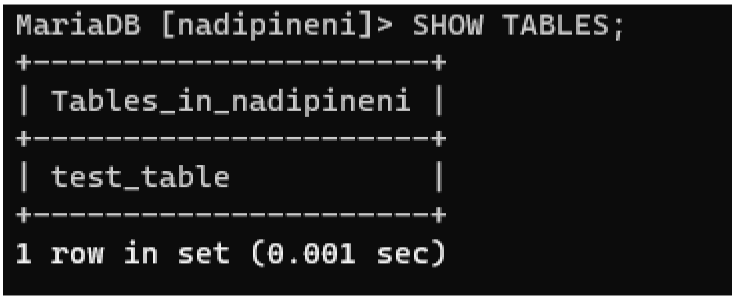
H)DELETE A TABLE:

BEFORE TRANSACTION:



```
-- DELETE A TABLE:
SHOW TABLES;
```

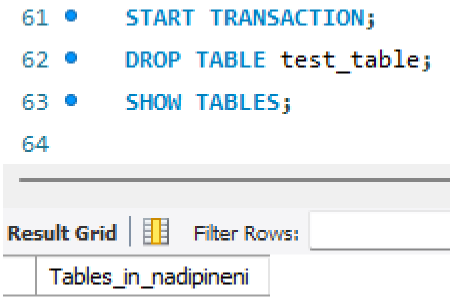
Tables_in_nadipineni
test_table



```
MariaDB [nadipineni]> SHOW TABLES;
+----------------------+
| Tables_in_nadipineni |
+----------------------+
| test_table           |
+----------------------+
1 row in set (0.001 sec)
```

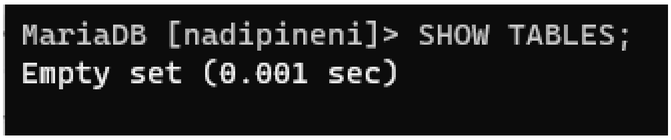
AFTER TRANSACTION:

- BEFORE ROLLBACK:



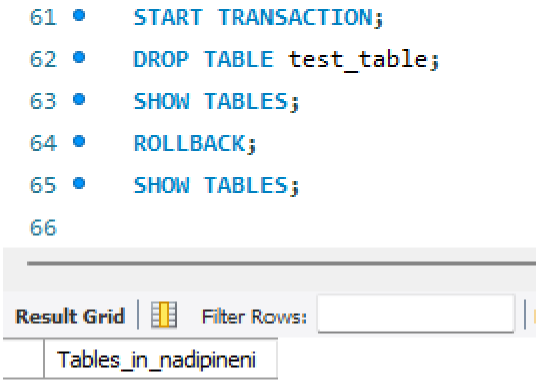
```
START TRANSACTION;
DROP TABLE test_table;
SHOW TABLES;
```

```
Tables_in_nadipineni
```



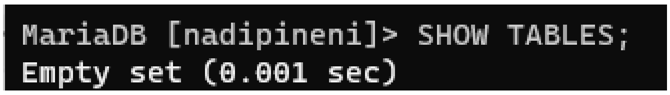
```
MariaDB [nadipineni]> SHOW TABLES;
Empty set (0.001 sec)
```

- AFTER ROLLBACK:



```
START TRANSACTION;
DROP TABLE test_table;
SHOW TABLES;
ROLLBACK;
SHOW TABLES;
```

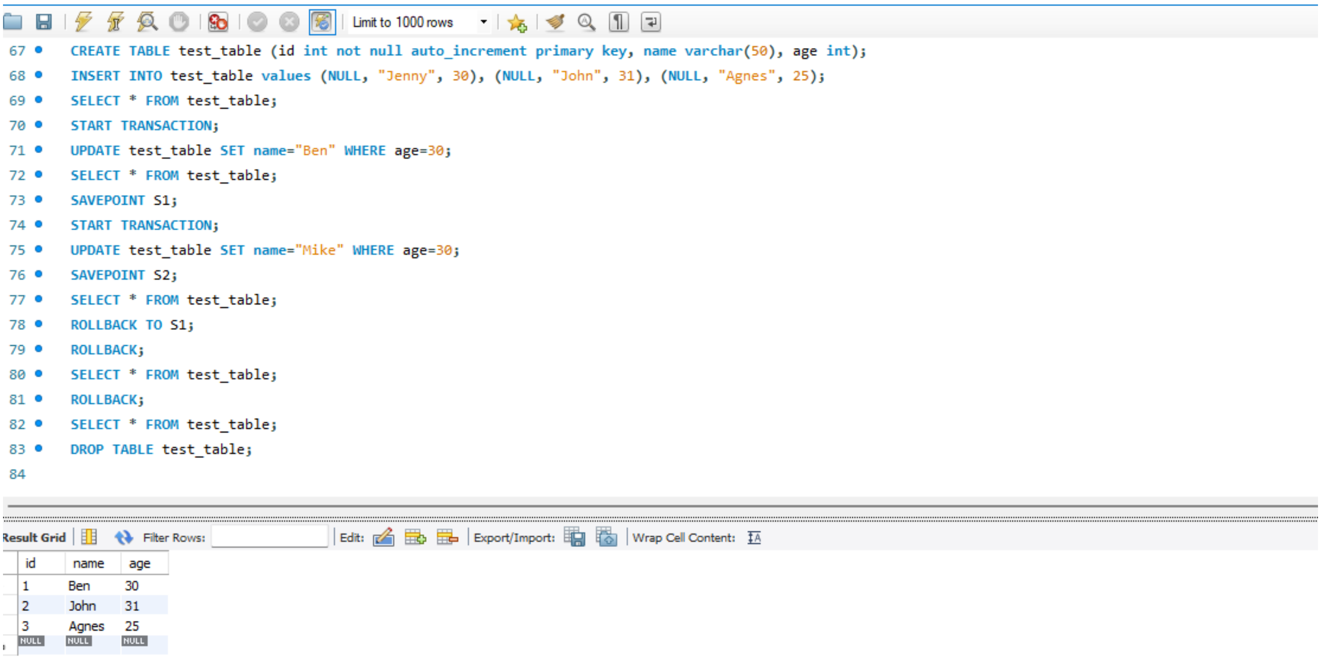
```
Tables_in_nadipineni
```



```
MariaDB [nadipineni]> SHOW TABLES;
Empty set (0.001 sec)
```

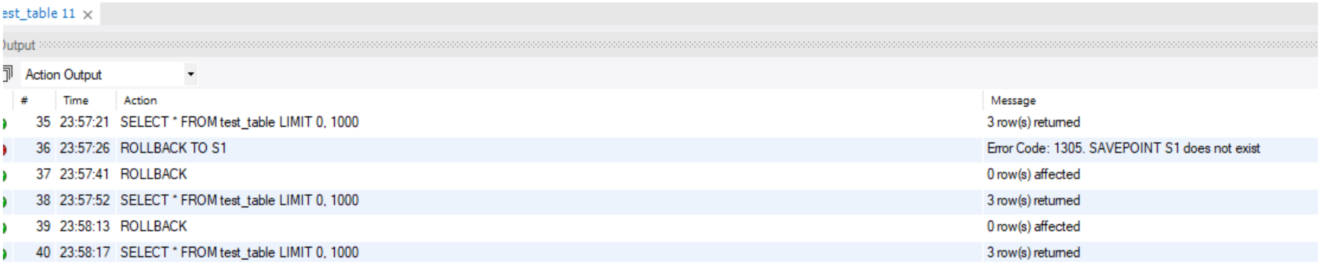
2) The test for nesting of transactions and the outcome of the test:

- TEST ONE: Test on DML commands



```sql
CREATE TABLE test_table (id int not null auto_increment primary key, name varchar(50), age int);
INSERT INTO test_table values (NULL, "Jenny", 30), (NULL, "John", 31), (NULL, "Agnes", 25);
SELECT * FROM test_table;
START TRANSACTION;
UPDATE test_table SET name="Ben" WHERE age=30;
SELECT * FROM test_table;
SAVEPOINT S1;
START TRANSACTION;
UPDATE test_table SET name="Mike" WHERE age=30;
SAVEPOINT S2;
SELECT * FROM test_table;
ROLLBACK TO S1;
ROLLBACK;
SELECT * FROM test_table;
ROLLBACK;
SELECT * FROM test_table;
DROP TABLE test_table;
```

id	name	age
1	Ben	30
2	John	31
3	Agnes	25
NULL	NULL	NULL



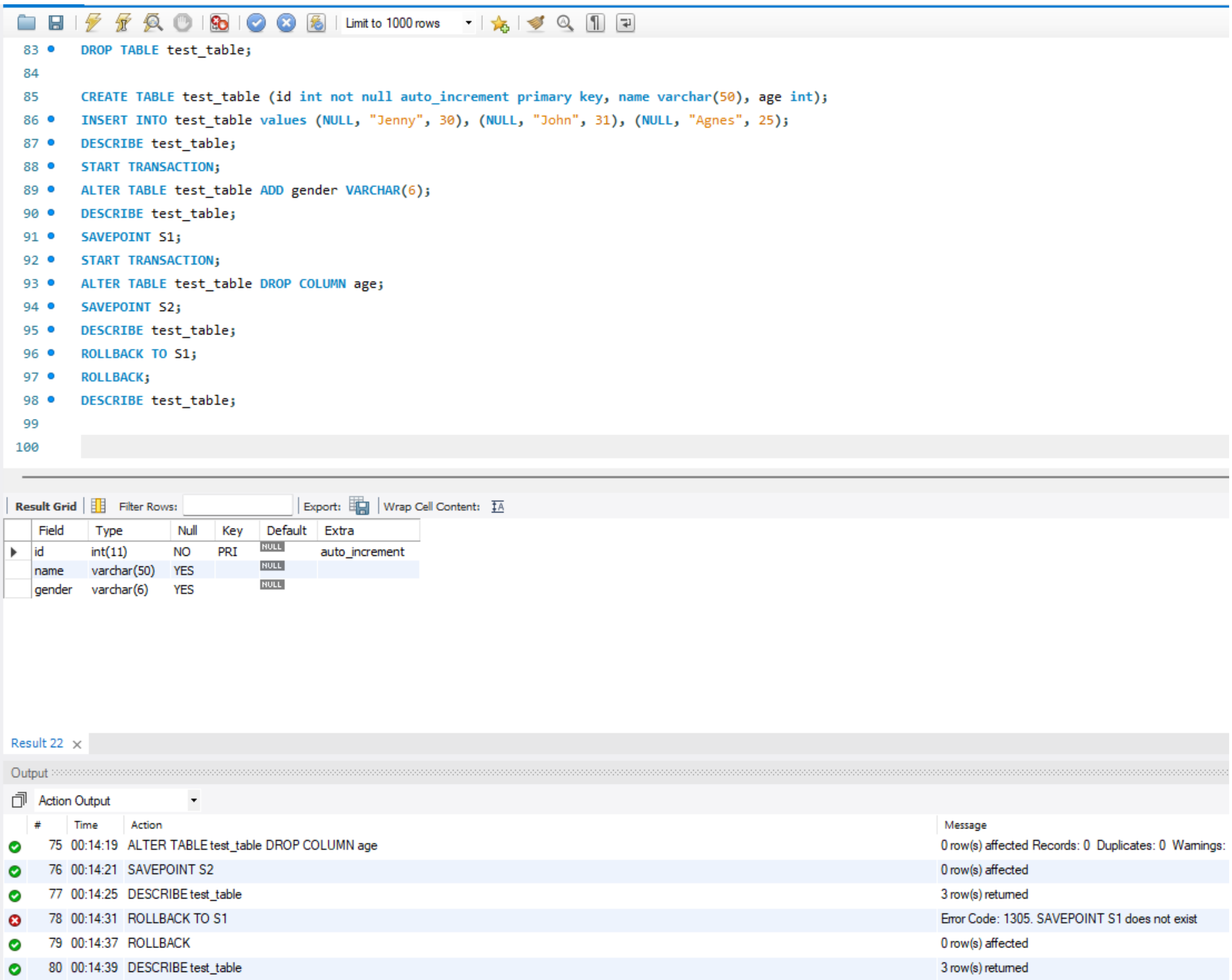
#	Time	Action	Message
35	23:57:21	SELECT * FROM test_table LIMIT 0, 1000	3 row(s) returned
36	23:57:26	ROLLBACK TO S1	Error Code: 1305. SAVEPOINT S1 does not exist
37	23:57:41	ROLLBACK	0 row(s) affected
38	23:57:52	SELECT * FROM test_table LIMIT 0, 1000	3 row(s) returned
39	23:58:13	ROLLBACK	0 row(s) affected
40	23:58:17	SELECT * FROM test_table LIMIT 0, 1000	3 row(s) returned

The above figure shows the test for nesting of transactions, and it clarifies that once we enter a second transaction we can’t roll back to the previous transaction. The order of test is as below:

1) START A TRANSACTION
2) DO SOME UPDATE OR CHANGE
3) SAVE A CHECKPOINT, LET IT BE S1
4) START ANOTHER TRANSACTION
5) DO SOME OTHER UPDATE OR CHANGE
6) SAVE ANOTHER CHECKPOINT, LET IT BE S2
7) TRIED ROLLING BACK TO S1 (DIDN’T WORK)
8) TRIED ROLLING BACK TWICE (DIDN’T WORK)

So, this test deduces that MySQL doesn’t support nested transactions

- TEST ONE: Test on DDL commands



```sql
DROP TABLE test_table;

CREATE TABLE test_table (id int not null auto_increment primary key, name varchar(50), age int);
INSERT INTO test_table values (NULL, "Jenny", 30), (NULL, "John", 31), (NULL, "Agnes", 25);
DESCRIBE test_table;
START TRANSACTION;
ALTER TABLE test_table ADD gender VARCHAR(6);
DESCRIBE test_table;
SAVEPOINT S1;
START TRANSACTION;
ALTER TABLE test_table DROP COLUMN age;
SAVEPOINT S2;
DESCRIBE test_table;
ROLLBACK TO S1;
ROLLBACK;
DESCRIBE test_table;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(50)	YES		NULL	
gender	varchar(6)	YES		NULL	

```
#   Time      Action                                    Message
75  00:14:19  ALTER TABLE test_table DROP COLUMN age    0 row(s) affected Records: 0  Duplicates: 0  Warnings:
76  00:14:21  SAVEPOINT S2                              0 row(s) affected
77  00:14:25  DESCRIBE test_table                       3 row(s) returned
78  00:14:31  ROLLBACK TO S1                            Error Code: 1305. SAVEPOINT S1 does not exist
79  00:14:37  ROLLBACK                                  0 row(s) affected
80  00:14:39  DESCRIBE test_table                       3 row(s) returned
```

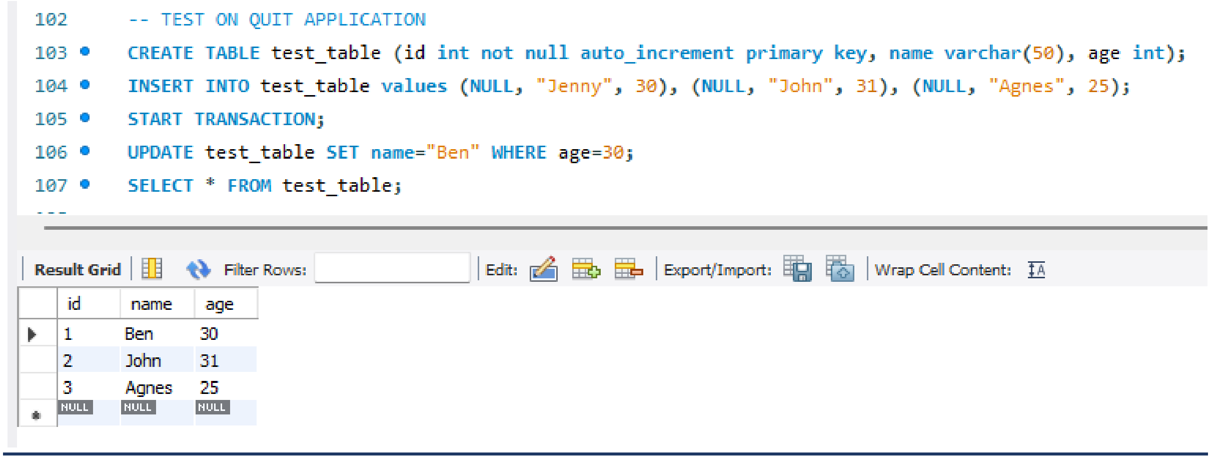
The above figure shows a test for the nesting of transactions, and it shows that once we enter a second transaction, we can't rollback. Also, as it is a DDL command, even rollback is not permitted. You can see that the drop column age is not rolled back. The order of test is as below:

1) START A TRANSACTION
2) DO SOME UPDATE OR CHANGE
3) SAVE A CHECKPOINT, LET IT BE S1
4) START ANOTHER TRANSACTION
5) DO SOME OTHER UPDATE OR CHANGE
6) SAVE ANOTHER CHECKPOINT, LET IT BE S2
7) TRIED ROLLING BACK TO S1 (DIDN'T WORK)
8) TRIED ROLLING BACK ONCE (DIDN'T WORK)

So, this test deduces that MySQL doesn't support nested transactions in DDL queries as well.

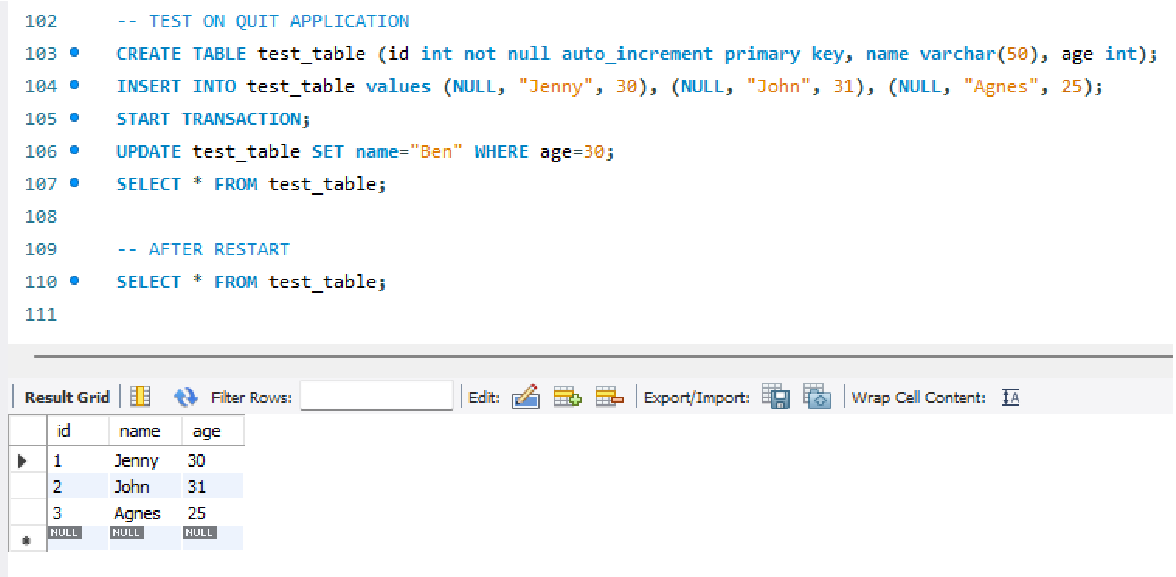
3) The test of the transactions extending to a restart of MySQLWorkbench:

These transactions test whether there will be a change in data after quitting the mysql application in between a transaction. And it is found that the update is not reflected after reopening the application. That means **the change is not visible in the database past the restart of MySQLWorkbench.**



```
-- TEST ON QUIT APPLICATION
CREATE TABLE test_table (id int not null auto_increment primary key, name varchar(50), age int);
INSERT INTO test_table values (NULL, "Jenny", 30), (NULL, "John", 31), (NULL, "Agnes", 25);
START TRANSACTION;
UPDATE test_table SET name="Ben" WHERE age=30;
SELECT * FROM test_table;
```

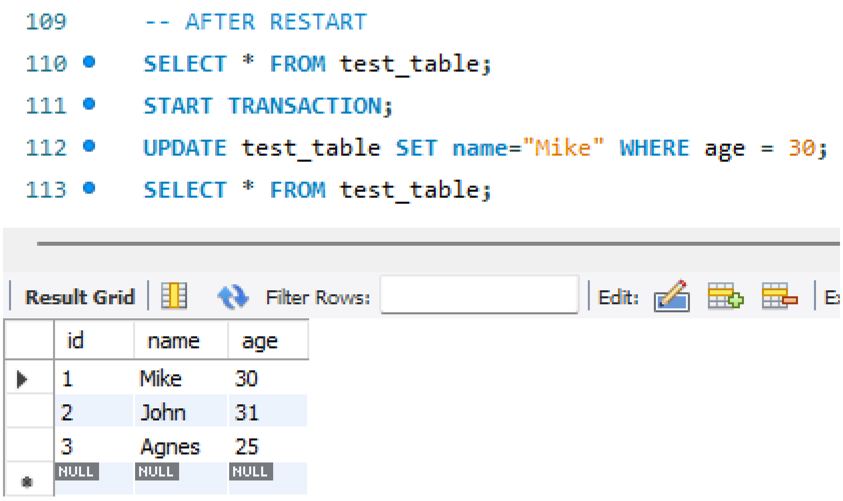
id	name	age
1	Ben	30
2	John	31
3	Agnes	25
NULL	NULL	NULL



```
-- TEST ON QUIT APPLICATION
CREATE TABLE test_table (id int not null auto_increment primary key, name varchar(50), age int);
INSERT INTO test_table values (NULL, "Jenny", 30), (NULL, "John", 31), (NULL, "Agnes", 25);
START TRANSACTION;
UPDATE test_table SET name="Ben" WHERE age=30;
SELECT * FROM test_table;

-- AFTER RESTART
SELECT * FROM test_table;

```

id	name	age
1	Jenny	30
2	John	31
3	Agnes	25
NULL	NULL	NULL

And **it allows me to make changes further on the data** as it is shown in the figure below:



```
-- AFTER RESTART
SELECT * FROM test_table;
START TRANSACTION;
UPDATE test_table SET name="Mike" WHERE age = 30;
SELECT * FROM test_table;
```

id	name	age
1	Mike	30
2	John	31
3	Agnes	25
NULL	NULL	NULL

4) Answer to the broadening question:

From the observation of the various type of queries from all the sets trying to roll back, I can say that we can roll back DML (Data Manipulation Language) queries such as the queries that manipulate data. For example, we can roll back the changes that are done on data inside the table where we UPDATE COLUMN VALUE, DELETE ROW …. etc. But we cannot roll back any changes that are done on the table itself (DDL-Data Driven Lang Queries) such as DROP TABLE, ALTER TABLE.

In summary, rollback can be done on DML Queries but not on DDL Queries.

References:

https://joshuaotwell.com/15501-2/

https://www.w3schools.com/sql/

https://www.youtube.com/watch?v=7S_tz1z_5bA