

Module 16: MySQL Transactions

Transaction:

- ☐ Transaction is a group of SQL statements that
 - ☐ When to use transactions:
 - ☐ When we code two or more INSERT, UPDATE or DELETE statements that affect related data.
 - ☐ When we move rows from one table to another table by using INSERT and DELETE statements.
 - ☐ Whenever the failure of an INSERT, UPDATE or DELETE statement would violate data integrity.
-

InnoDB:

- ☐ The InnoDB storage engine provides transactional capabilities
 - ☐ InnoDB satisfies the conditions for ACID compliance.
 - ☐ **Atomic:** All the statements execute successfully or are canceled as a unit.
 - ☐ **Consistent:** A database that is in a consistent state when a transaction begins is left in a consistent state by the transaction.
 - ☐ **Isolated:** One transaction does not affect another.
 - ☐ **Durable:** All the changes made by a transaction that completes successfully are recorded properly in the database. Changes are not lost.
-

Transaction Model:

- ☐ **Model :**
 - ☐ We suspend the current autocommit mode by beginning a transaction explicitly by writing:
 - ☐ **START TRANSACTION; / BEGIN; /BEGIN WORK;**
 - ☐ After beginning a transaction autocommit remains disabled until we end the transaction by either commit or rollback.

Program with Transaction:

```
create procedure tr2()
begin
declare issue boolean default false;
declare continue handler for SQLEXCEPTION
    set issue = true;
```

```
mysql> select * from stu30;
+----+-----+
| rno | name |
+----+-----+
| 10 | amit |
| 20 | rahul |
| 30 | modi |
+----+-----+
3 rows in set (0.07 sec)
```

start transaction;

```
insert into stu30 values(40, 'maya');
insert into stu30 values (30, 'soniya');
insert into stu30 values(50, 'jaya');
```

```
+-----+
| transaction rollback |
+-----+
| transaction rollback |
+-----+
1 row in set (0.09 sec)
```

```
if ! issue then
```

commit;

```
select 'transaction committed';
```

```
else
```

rollback;

```
select 'transaction rollback ';
```

```
end if ;
```

```
end $$
```

```
mysql> select * from stu30;
+----+-----+
| rno | name |
+----+-----+
| 10 | amit |
| 20 | rahul |
| 30 | modi |
+----+-----+
3 rows in set (0.00 sec)
```

```
call tr2() $$
```

Savepoints:

- ❑ Save points are useful when a single transaction contains so many SQL statements that rolling back the entire transaction would be inefficient.
- ❑ We can use the SAVEPOINT statement to create a save point with the specified name.
- ❑ Syntax is:
 - ❑ **SAVEPOINT savepoint_name;**

Rollback:

- ❑ We can use the ROLLBACK TO SAVEPOINT statement to roll back a transaction to the specified save point.
- ❑ Syntax is:
 - ❑ **ROLLBACK TO SAVEPOINT savepoint_name**
- ❑ When we use save points we can roll back a transaction to the beginning of the transaction or to a particular save point.
- ❑ The transaction rolls back to the named savepoint and we can continue from there. Any savepoints that were set after the savepoint are deleted.

Program:

```
create procedure tr3()
```

```
begin
```

```
start transaction;
```

```
savepoint sp1;
```

```
insert into stu30 values(101, 'seema');
```

```
savepoint sp2;
```

```
insert into stu30 values(102, 'reema');
```

```
insert into stu30 values(103, 'reema');
```

```
rollback to sp2;
```

```
commit;
```

```
end $$
```

```
call tr3() $$
```

```
mysql> select * from stu30;
```

rno	name
10	amit
20	rahul
30	modi
40	maya
50	jaya

```
5 rows in set (0.00 sec)
```

```
mysql> select * from stu30;
```

rno	name
10	amit
20	rahul
30	modi
40	maya
50	jaya
101	seema

```
6 rows in set (0.00 sec)
```

MCQS:

Q1) By default MySQL runs in which mode?

Options:

- | | |
|----------------|----------------|
| A. auto save | B. select |
| C. auto ignore | D. auto commit |

Solution:

Q2) What is the way of starting a transaction?

Options:

- | | |
|-------------------|----------------------|
| A. start | B. start transaction |
| C. autocommit = 0 | D. set transaction |

Solution:

Q3) What are the ways of completing a transaction. Choose all that apply.

Options:

- | | |
|-----------|--------------|
| A. COMMIT | B. SAVEPOINT |
| C. SAVE | D. ROLLBACK |

Solution:

Q4) What does 'A' stand for in the ACID property of transactions?

Options:

- | | |
|------------------|--------------|
| a) Availability | b) Accuracy |
| c) Adjustability | d) Atomicity |

Solution:

Q5) What does 'C' stand for in the ACID property of transactions?

Options:

- | | |
|--------------|----------------|
| a) Compound | b) Concrete |
| c) Collision | d) Consistency |

Solution: