

## Experiment– 10

---

**Name :** Meenakshi

**UID :** 23BDO10011

**Subject :** ADBMS

**Section & Group :** 23BCC-BDO-2

---

### QUESTION:

Demonstrate the ACID properties of database transactions (focusing on Atomicity and Consistency) by performing multiple inserts into the FeePayments table, handling failures with ROLLBACK, and ensuring the database remains consistent.

Solution:

**Aim:** To showcase the working of ACID principles through practical SQL examples involving transaction control, rollback, and commit operations.

### Theory:

A transaction is a logical unit of work in a database, consisting of a sequence of SQL commands that execute together. Either all commands succeed, or none do. The ACID properties that ensure reliable transactions are:

- **Atomicity:** All operations succeed as one unit, or none are applied.
- **Consistency:** Guarantees the database moves from one valid state to another.
- **Isolation:** Multiple transactions run independently without interference.
- **Durability:** Once committed, changes are permanently stored.

### Implementation (SQL Commands):

```
-- Part A: Successful Transaction
START TRANSACTION;

INSERT INTO FeePayments (payment_id, student_name, amount, payment_date)
VALUES (1, 'Ashish', 5000.00, '2024-06-01');

INSERT INTO FeePayments (payment_id, student_name, amount, payment_date)
VALUES (2, 'Smaran', 4500.00, '2024-06-02');

INSERT INTO FeePayments (payment_id, student_name, amount, payment_date)
VALUES (3, 'Vaibhav', 5500.00, '2024-06-03');

COMMIT;

SELECT * FROM FeePayments;

-- Part B: Failed Transaction with Rollback
START TRANSACTION;

INSERT INTO FeePayments (payment_id, student_name, amount, payment_date)
VALUES (4, 'Kiran', 4800.00, '2024-06-04');

-- Attempting duplicate ID (will fail)
INSERT INTO FeePayments (payment_id, student_name, amount, payment_date)
VALUES (1, 'Ashish', 5000.00, '2024-06-01');

ROLLBACK;

SELECT * FROM FeePayments;
```

```
-- Part C: Partial Failure Example
START TRANSACTION;

INSERT INTO FeePayments (payment_id, student_name, amount, payment_date)
VALUES (5, 'Rohit', 5000.00, '2024-06-05');

-- Inserting NULL into student_name (invalid)
INSERT INTO FeePayments (payment_id, student_name, amount, payment_date)
VALUES (6, NULL, 4700.00, '2024-06-06');

ROLLBACK;

SELECT * FROM FeePayments;
```

Outputs:

payment_id	student_name	amount	payment_date
1	Ashish	5000.00	2024-06-01
2	Smaran	4500.00	2024-06-02
3	Vaibhav	5500.00	2024-06-03

Learning Outcomes:

- Understood the use of START TRANSACTION, COMMIT, and ROLLBACK.
- Observed how atomicity ensures all-or-none execution.
- Verified consistency is preserved across valid states.
- Learned to handle constraint violations using rollback.
- Strengthened understanding of ACID concepts for reliable database design.