ADBMS PRACTICAL - EXPERIMENT-10

Experiment: FeePayments Table with Transactions and Rollbacks

AIM

To demonstrate the use of SQL transactions (BEGIN, COMMIT, ROLLBACK) with constraints such as PRIMARY KEY, NOT NULL, and CHECK in a FeePayments table.

CODE

```
CREATE TABLE FeePayments (
    payment_id INT PRIMARY KEY,
    student_name VARCHAR(100) NOT NULL,
    amount DECIMAL(10,2) CHECK (amount > 0),
    payment_date DATE NOT NULL
);

BEGIN;
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;

BEGIN;
INSERT INTO FeePayments (payment_id, student_name, amount, payment_date)
VALUES (4, 'Kiran', 5200.00, '2024-06-04');
INSERT INTO FeePayments (payment_id, student_name, amount, payment_date)
VALUES (2, 'Sneha', -4700.00, '2024-06-08');
-- duplicate payment_id and negative amount (to force failure)
ROLLBACK;

BEGIN;
INSERT INTO FeePayments (payment_id, student_name, amount, payment_date)
VALUES (5, 'Aditya', 4600.00, '2024-06-05');
INSERT INTO FeePayments (payment_id, student_name, amount, payment_date)
VALUES (6, NULL, 4700.00, '2024-06-06');
-- NULL student_name (violates NOT NULL constraint)
ROLLBACK;

-- After above transactions, only valid committed data should exist:
SELECT * FROM FeePayments;
```

EXPECTED OUTPUT

```
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

(Only the first committed transaction is preserved, later ones were rolled back due to errors.)