

Review 5

Implementation of concurrency control and recovery mechanisms.

Transaction:

A transaction is a group of one or more SQL operations (like INSERT, UPDATE, DELETE, etc.) that are executed as a single unit of work. A transaction ensures data consistency and integrity.

Rollback:

A rollback undoes all operations in the current transaction, reverting the database to its previous consistent state.

Commit:

A commit permanently saves all changes made during the transaction to the database. After a commit, changes become visible to other users and survive system crashes.

Transaction:

1. Add a new Bonus for Employee 105:

START TRANSACTION;

```
INSERT INTO Bonus (Bonus_ID, Employee_ID, Amount)
VALUES (404, 105, 4500);
```

COMMIT;

2. Add a new Deduction for Employee 101:

START TRANSACTION;

```
INSERT INTO Deductions (Deduction_ID, Amount, Reason)
VALUES (4, 300.00, 'Misuse of office internet');
```

COMMIT;

3. Add a new Salary entry for Employee 105:

START TRANSACTION;

```
INSERT INTO Salary (Salary_employeeID, Salary_Base_Amount, Salary_Desc)
VALUES (105, 55000.00, 'Mid-level Developer');
```

```
COMMIT;
```

4. Add a Salary_Report (for tax calculation record):

```
START TRANSACTION;
```

```
INSERT INTO Salary_Report (Salary_Report_ID, Salary_employeeID, Tax_ID, Final_Salary)
VALUES (301, 105, 2, 49500.00); -- Assuming Tax_ID = 2 exists
```

```
COMMIT;
```

5. Insert a Payroll_Report entry (complete payroll calculation):

```
START TRANSACTION;
```

```
INSERT INTO Payroll_Report (Payroll_Report_ID, Employee_ID, Salary_ID, Bonus_ID,
Deduction_ID, Final_Amount, Generated_On)
VALUES (3, 105, 105, 404, 3, 53000.00, NOW());
```

```
COMMIT;
```

6. Update Salary of Employee 101:

```
START TRANSACTION;
```

```
UPDATE Salary
SET Salary_Base_Amount = 72000.00
WHERE Salary_employeeID = 101;
```

```
COMMIT;
```

7. Simulate an error & ROLLBACK if bonus doesn't exist:

```
START TRANSACTION;
```

```
INSERT INTO Payroll_Report (Payroll_Report_ID, Employee_ID, Salary_ID, Bonus_ID,
Deduction_ID, Final_Amount, Generated_On)
VALUES (4, 106, 106, 999, 1, 58000.00, NOW()); -- Bonus_ID 999 doesn't exist
```

```
ROLLBACK; -- Will be triggered due to FK error
```

8. Delete and revert a Deduction entry:

```
START TRANSACTION;
```

```
DELETE FROM Deductions  
WHERE Deduction_ID = 4;
```

```
ROLLBACK; -- Cancel the deletion to restore it
```