

Procedures and Views

Procedures:

We have included 1 procedure for every table present in the ERP database and 3 others for resolving the Addition, Substitution, and Withdrawal queries.

1. addStudent

This procedure first checks if the length of 'name' and 'username' are less than 20 and 40 characters respectively. Then it checks for any duplicates present in the 'Student' table. If all these requirements are fulfilled then the data is inserted in the 'Student' table and the changes are committed, else the transaction is rolled back.

2. addTeacher

This procedure first checks if the length of 'name' and 'username' are less than 20 and 40 characters respectively. Then it checks for any duplicates present in the 'Teacher' table. If all these requirements are fulfilled then the data is inserted in the 'Teacher' table and the changes are committed, else the transaction is rolled back.

3. addCourse

This procedure first checks if the length of 'name' is less than 20 characters and 'capacity' is less than 100. Then it checks whether there exist an Instructor-in-charge for any particular course. If it doesn't, it will create one with the same ID provided in the argument. If it does, then it will verify the ID of IC by the criteria of teachers. If all the requirements are fulfilled then the data is inserted into 'Course' table and the changes are committed, else the transaction is rolled back.

4. addTakes

This procedure first checks if the student id and the course id provided in the argument exist. Then it checks if there are any seats left in the course. Then it checks if there are any duplicates present in the 'Takes' table. If all the requirements are fulfilled then the data is inserted in the table, the number of seats in the course is decreased by one, and the changes are committed, else the transaction is rolled back.

5. addAssists

This procedure first checks if the course id and teacher id provided in the argument exist. Then it checks whether there is any duplicate entries in the 'Assists' table. If the requirements are fulfilled, the transaction is committed, else the transaction is rolled back

6. addAdditions

This procedure first checks if student id and course id exist. Then it verifies whether the student has not taken the particular course. Then it checks whether the student has already applied for the particular course addition request. If all the requirements are fulfilled, then the data is added to the 'Addition course' table and the transaction is committed, else the transaction is rolled back.

7. addSubstitutions

This procedure first checks if the student id, current course id and the substitution course id exist. Then it verifies whether the student has the current course and doesn't have the substitution course in his list of course. Then it checks whether the student has already applied for the Substitution request. If all the requirements are fulfilled, then the data is inserted in the 'Substitution course' table and the transaction is committed, else the transaction is rolled back.

8. addWithdraw

This procedure first checks if the student id, current course id and the substitution course id exist. Then it verifies whether the student has the course in his list of course. Then it checks whether the student has already applied for the Withdrawal request. If all the requirements are fulfilled, then the data is inserted in the 'Substitution course' table and the transaction is committed, else the transaction is rolled back.

9. resolveAddition

The procedure first verifies whether the status of query is 'Pending'. Then it verifies whether there are any seats left, and whether the student already has the course in his course list. If all the requirements are fulfilled, then the status of request is changed to 'Successful', the course is added to the course list of the student, and the transaction is committed, else the transaction is rolled back.

10. resolveSubstitution

The procedure first verifies whether the status of the query is 'Pending'. Then it verifies whether there are any seats left in the substitution course, whether the student has not already put a request for the same substitution request, and whether the current course is present in the course list of the student. If all the requirements are fulfilled then the status is changed to 'Successful', the Substitution course is added to course list, current is removed from the course list, and the changes are committed, else the transaction is rolled back.

11. resolveWithdrawal

The procedure only verifies whether the status of request is 'Pending'. If it is satisfied, then the status is changed to 'Successful', the course is deleted from the students list, the seats available is incremented by one, the course is added to the Withdraw table, and then all the changes are committed. However, if the condition is not satisfied, the transaction is rolled back.

Following ACID properties:

1. Atomicity

The procedures follow atomicity, since the procedures act on one request at a time.

2. Consistency

3. Integrity

4. Durability

Views:

We have provided 4 views for displaying the data on the front-end page:

1. myCourses:

This view displays all the courses present in the list of courses for the student

2. myAdditions:

This view displays all the Addition request which the student has applied for.

3. mySubstitutions:

This view displays all the Substitution request which the student has applied for

4. myWithdrawals:

This view displays all the Withdrawal request which the student has applied for.