

SQL query that will return "all the StudyGroups which have at least a user with 'name' starting on 'M' sorted by 'creation date'" like "Miguel" or "Manuel".

```
SELECT sg.*
FROM StudyGroups sg
INNER JOIN Users u ON u.StudyGroupId = sg.StudyGroupId
WHERE u.Name LIKE 'M%'
GROUP BY sg.StudyGroupId
HAVING COUNT(*) > 0
ORDER BY sg.CreateDate;
```

### High Level Steps and Expectations:

1. User creates a new StudyGroup with valid inputs (name, subject)
2. System validates the inputs and assigns a unique ID.
3. System creates a new StudyGroup object with the provided information.
4. User can add and remove users from the group.
5. System updates the user list and maintains the integrity of the group.

#### Assumptions:

- The system provides a mechanism for users to create new StudyGroups.
- Users can be identified and associated with StudyGroups.
- Valid subjects are defined by the Subject enum.

Test Case	Description	Inputs	Expected Outcome	Testing Level	Regression
1,1	Create a new StudyGroup with valid name and subject	Name: "Calculus Study Group", Subject: Math	StudyGroup object is created with assigned ID, "Calculus Study Group", Math, and current date	Unit	Yes
1,2	Create a new StudyGroup with empty name	Name: "", Subject: Math	Validation error is thrown indicating name cannot be empty	Unit	Yes
1,3	Create a new StudyGroup with invalid subject	Name: "Physics Study Group", Subject: "Biology"	Validation error is thrown indicating subject is invalid	Unit	Yes
2,1	Add a user to an existing group	User: user1, Group: existing group	User1 is added to the group's user list	Unit	Yes
2,2	Add a user already present in the group	User: user1, Group: existing group containing user1	Validation error is thrown indicating user already exists in the group	Unit	Yes

2,3	Remove a user from the group	User: user1, Group: existing group containing user1	User1 is removed from the group's user list	Unit	Yes
2,4	Remove a user not present in the group	User: user2, Group: existing group	Validation error is thrown indicating user is not present in the group	Unit	Yes
3,1	Verify user list size after adding and removing users	Group: existing group	User list size reflects the expected number of users after adding and removing	Component	Yes
4,1	Manually create a new study group with UI	Name: "Chemistry Study Group", Subject: Chemistry	StudyGroup is created and displayed in the UI	E2E (manual)	Yes
4,2	Manually add and remove users from a study group	User1, User2, Group: existing group	Users are added and removed successfully, and the UI reflects the changes	E2E (manual)	Yes

### Test cases for the StudyGroupController:

#### CreateStudyGroup:

- Input: Valid StudyGroup object with a unique subject, a name with 10 characters, and a valid create date.
- Expected Output: HTTP 200 (OK)

#### CreateStudyGroup - Duplicate Subject:

- Input: Valid StudyGroup object with a subject that already exists.
- Expected Output: HTTP 4xx (Bad Request) - Duplicate subject error

#### CreateStudyGroup - Invalid Name:

- Input: Valid StudyGroup object with a name less than 5 characters.
- Expected Output: HTTP 4xx (Bad Request) - Invalid name error

#### CreateStudyGroup - Invalid Subject:

- Input: Valid StudyGroup object with an invalid subject.
- Expected Output: HTTP 4xx (Bad Request) - Invalid subject error

#### GetStudyGroups:

- Input: None
- Expected Output: HTTP 200 (OK) with a list of existing Study Groups

#### SearchStudyGroups:

- Input: Valid subject for filtering
- Expected Output: HTTP 200 (OK) with a list of Study Groups for the given subject

#### JoinStudyGroup:

- Input: Valid studyGroupId and userId
- Expected Output: HTTP 200 (OK)

#### LeaveStudyGroup:

- Input: Valid studyGroupId and userId for a user who has already joined the group
- Expected Output: HTTP 200 (OK)

LeaveStudyGroup - User Not Joined:

- Input: Valid studyGroupId and userId for a user who hasn't joined the group
- Expected Output: HTTP 4xx (Bad Request) - User not joined error