

CPSC 304 Project Cover Page

Milestone #: 4

Date: Apr 5, 2023

Group Number: 1 (project mentor: Suki)

Name	Student Number	CS Alias (userid)	Preferred Email Address
Noreen Chan	89278618	s4v2b	chan.noreen609@gmail.com
Andy Liang	47847876	b4e3b	andyliang1000@gmail.com
Jerry Shao	84982321	c1i3b	shaojerry@icloud.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student ids are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

1b. Project Summary

This application is a management tool for gym managers to keep track of gym logistics and users. Gym managers can query information like a list of users, equipment information such as exercises that use it, and the body parts that those exercises target.

1c. Repository Link

https://github.students.cs.ubc.ca/CPSC304-2022W-T2/project_b4e3b_c1i3b_s4v2b

2. SQL Script

Note: This can also be found in *milestone_4_assets/Milestone4InitScript.sql*

```
CREATE TABLE Users (
    Name varchar(127),
    Email varchar(127),
    MembershipType varchar(31) NOT NULL,
    PRIMARY KEY (Email)
);

CREATE TABLE Does_Exercise (
    Number_of_sets int,
    Exercise_Name varchar(127),
    PRIMARY KEY (Exercise_Name),
);
;

CREATE TABLE BodyPart (
    bp_Name varchar(127),
    PRIMARY KEY (bp_Name)
);
;

CREATE TABLE Targets(
    Exercise_Name varchar(127),
    BodyPart_name varchar(127),
    Intensity_rating int,
    PRIMARY KEY (Exercise_Name, BodyPart_name),
    FOREIGN KEY (Exercise_Name) references
Tutorial2.dbo.Does_Exercise(Exercise_Name) ON DELETE CASCADE,
    FOREIGN KEY (BodyPart_name) references Tutorial2.dbo.BodyPart(bp_Name)
);
```

```

CREATE TABLE Class (
    Class_ID varchar(127),
    Price int,
    Name varchar(127),
    Start_time TIME(0) NOT NULL,
    End_time TIME(0) NOT NULL,
    Instructor_name varchar(127) NOT NULL,
    Exercise_Name varchar(127) NOT NULL,
    PRIMARY KEY (Class_ID),
    FOREIGN KEY (Exercise_Name) REFERENCES Does_Exercise(Exercise_Name) ON
DELETE CASCADE
);

INSERT INTO Does_Exercise VALUES (5, 'Bench Press');
INSERT INTO Does_Exercise VALUES (5, 'Squat');
INSERT INTO Does_Exercise VALUES (3, 'Deadlift');
INSERT INTO Does_Exercise VALUES (5, 'Curls');
INSERT INTO Does_Exercise VALUES (4, 'Shoulder Press');

INSERT INTO Class VALUES ('Class1', 40, 'C1', '10:00:00', '11:00:00',
'Anon Atom', 'Bench Press');
INSERT INTO Class VALUES ('Class2', 15, 'C2', '11:00:00', '12:00:00',
'Anon Beaker', 'Squat');
INSERT INTO Class VALUES ('Class3', 10, 'C3', '12:00:00', '13:00:00',
'Anon Beaker', 'Deadlift');
INSERT INTO Class VALUES ('Class4', 30, 'C4', '13:00:00', '14:00:00',
'Anon Atom', 'Curls');
INSERT INTO Class VALUES ('Class5', 1, 'C5', '14:00:00', '15:00:00', 'Anon
Comp', 'Shoulder Press');
INSERT INTO Class VALUES ('Class6', 15, 'C6', '12:00:00', '13:00:00',
'Anon Book', 'Deadlift');
INSERT INTO Class VALUES ('Class7', 50, 'C7', '13:00:00', '14:00:00',
'Anon Book', 'Curls');
INSERT INTO Class VALUES ('Class8', 20, 'C8', '14:00:00', '15:00:00',
'Anon Atom', 'Shoulder Press');

```

```
INSERT INTO BodyPart VALUES
    ('Chest'),
    ('Arms'),
    ('Legs'),
    ('Back'),
    ('Shoulders');

INSERT INTO Targets VALUES
    ('Curls', 'Arms', 2),
    ('Shoulder Press', 'Shoulders', 2),
    ('Bench Press', 'Chest', 9),
    ('Bench Press', 'Arms', 7),
    ('Bench Press', 'Legs', 2),
    ('Bench Press', 'Back', 2),
    ('Bench Press', 'Shoulders', 7),
    ('Squat', 'Arms', 2),
    ('Squat', 'Legs', 10),
    ('Squat', 'Back', 5),
    ('Squat', 'Shoulders', 2);

INSERT INTO Users (Name, Email, MembershipType) VALUES
    ('Raymond Ng', 'rng@gmail.com', 'BASIC'),
    ('Jessica Wong', 'jwong@ubc.ca', 'BASIC'),
    ('Jeff Clune', 'jeffclune@gmail.com', 'PRO'),
    ('JJ Jim', 'jjim@ubc.ca', 'BASIC'),
    ('Norm Hutchinson', 'norm@ubc.ca', 'PRO');
```

3a. Final Project Description

The application is a management tool for gym managers to keep track of gym logistics and users, organized into 4 tabs.

Details on the tabs are as follows:

- The first tab, Users, allows you to:
 - Add new users
 - Delete users
 - Update 1+ users (*Query c, update*)
 - Filter for only PRO and/or BASIC user membership types (*Query d, selection*)
 - View total amount of users with a certain membership type (*Query g, group by*)
 - Filter for classes with certain exercises (*Query f, join*)
- The second tab, Exercises, allows you to:
 - Show exercises that target all body parts (*Query j, division*)
 - Delete exercises, which also deletes any classes that used that exercise (*Query b, delete with FK*)
- The third tab, Class, allows you to:
 - Insert a new class, only if the class' exercise exists (*Query a, insert with FK*)
 - Find the cheapest class given the instructor's popularity rating (*Query i, nested aggregation*)
 - Select which columns of the class to view (*Query e, projection*)
- The fourth tab, Targets, allows you to:
 - Select exercises that have a given average intensity (*Query h, agg w/ having*)

3b. Schema Differences & Screenshots

Differences

- User's **MembershipType** was made *NOT NULL* to facilitate with implementing selection
- **Does_Exercise**'s foreign key to **Class** was reversed (i.e. now **Class** has the foreign key referencing **Does_Exercise**) to support *on delete cascade* due to an original design mistake from Milestone 2.
- **BodyPart**'s **Name** was renamed to **bp_Name** so that it's easier to read
- **Offers_Class_1** was renamed to **Class** because we no longer have 2 tables named Class. Additionally, we are not implementing Gym, so we don't need the foreign key **Gym_Address**.
- Any instances of type **Date** have been changed to type **Time** because we don't use
- The remainder of the tables are out of scope for this project.

Screenshots

The screenshot shows three windows from SQL Server Management Studio (SSMS) illustrating the implementation of three queries related to a Gym Management system.

Object Explorer: Shows the database structure for DESKTOP-0GSIURS (SQL Server 16.0.1000.6). A green box highlights the tables under the 'Tutorial2' database node: `dbo.BodyPart`, `dbo.Class`, `dbo.Does_Exercise`, `dbo.Targets`, and `dbo.Users`.

Query Results (Top Window): Displays the results of three queries:

- BodyPart:** A table showing body parts with their names: Arms, Back, Chest, Legs, and Shoulders.
- Class:** A table showing exercise classes with columns: Class_ID, Price, Name, Start_time, End_time, Instructor_name, and Exercise_Name. Data includes entries like Class1 (\$40, C1, 10:00-11:00, Anon Atom, Bench Press), Class2 (\$15, C2, 11:00-12:00, Anon Beaker, Squat), etc.
- Does_Exercise:** A table showing exercises assigned to body parts with columns: Number_of_sets, Exercise_Name, and BodyPart_name. Data includes entries like 5 sets of Bench Press for Arms, 3 sets of Deadlift for Back, etc.

Query Results (Bottom Window): Displays the results of three queries:

- Targets:** A table showing target exercises with columns: Exercise_Name, BodyPart_name, and Intensity_rating. Data includes entries like Bench Press for Arms (rating 7), Bench Press for Back (rating 2), etc.
- Users:** A table showing user information with columns: Name, Email, and MembershipType. Data includes entries like Jeff Clune (jeffclune@gmail.com, PRO), Jim (jim@ubc.ca, BASIC), etc.

Status Bar: Shows the message "Query executed successfully." and the details "DESKTOP-0GSIURS (16.0 RTM) DESKTOP-0GSIURS\andy1 ... Tutorial2 | 00:00:00 | 11 rows".

3d & 3e. Queries

The query, the in-implementation reference of it, and screenshots of query results are included in each subsection.

For each query, the data tables screenshot follows the frontend screenshot.

Query a. INSERT

File:

project_b4e3b_c13b_s4v2b\Server\GymManagement.Infrastructure\Persistence\ClassRepository.cs

Line: 18 - 40

Before:

The screenshot shows the 'Gym Manager' application interface. On the left, there's a navigation menu with tabs: 'USERS', 'EXERCISES', 'CLASS' (which is highlighted), and 'TARGETS'. The main area has a title 'Add New Entry / View Insights' and contains a table with 8 rows of data. The columns are labeled: name, price(\$), start_time, end_time, instructor_name, class_ID, and exercise_name. The data rows are: C1 (40, 10:00:00, 11:00:00, Anon Atom, Class1, Bench Press); C2 (15, 11:00:00, 12:00:00, Anon Beaker, Class2, Squat); C3 (10, 12:00:00, 13:00:00, Anon Beaker, Class3, Deadlift); C4 (30, 13:00:00, 14:00:00, Anon Atom, Class4, Curls); C5 (1, 14:00:00, 15:00:00, Anon Comp, Class5, Shoulder Press); C6 (15, 12:00:00, 13:00:00, Anon Book, Class6, Deadlift); C7 (50, 13:00:00, 14:00:00, Anon Book, Class7, Curls); C8 (20, 14:00:00, 15:00:00, Anon Atom, Class8, Shoulder Press). To the right of the table is a sidebar with a heading 'Select classes that teach:' followed by a list of checkboxes: Name, Price, Start time, End time, Instructor name, Class ID, and Exercise Name. All checkboxes are checked.

The screenshot shows the SSMS interface with two tabs: 'SQLQuery2.sql' and 'SQLQuery1.sql'. The 'SQLQuery2.sql' tab contains the query 'Select * from Class'. The results pane shows a table with 8 rows of data, matching the data in the Gym Manager application. The columns are: Class_ID, Price, Name, Start_time, End_time, Instructor_name, and Exercise_Name. The data is identical to the table in the Gym Manager screenshot.

	Class_ID	Price	Name	Start_time	End_time	Instructor_name	Exercise_Name
1	Class1	40	C1	10:00:00	11:00:00	Anon Atom	Bench Press
2	Class2	15	C2	11:00:00	12:00:00	Anon Beaker	Squat
3	Class3	10	C3	12:00:00	13:00:00	Anon Beaker	Deadlift
4	Class4	30	C4	13:00:00	14:00:00	Anon Atom	Curls
5	Class5	1	C5	14:00:00	15:00:00	Anon Comp	Shoulder Press
6	Class6	15	C6	12:00:00	13:00:00	Anon Book	Deadlift
7	Class7	50	C7	13:00:00	14:00:00	Anon Book	Curls
8	Class8	20	C8	14:00:00	15:00:00	Anon Atom	Shoulder Press

During:

Gym Manager

USERS
EXERCISES
CLASS
TARGETS

Start_time(*) : 11:11 AM
End_time(*) : 11:11 AM
Class_ID(*) : Class 9
Instructor_name(*) : Stormi
Exercise_name(*) : Bench Press

name	price(\$)	start_time	end_time	instructor_name	class ID
C5	1	14:00:00	15:00:00	Anon Comp	Class5
C6	15	12:00:00	13:00:00	Anon Book	Class6
C7	50	13:00:00	14:00:00	Anon Book	Class7
C8	20	14:00:00	15:00:00	Anon Atom	Class8

Select classes that teach:

- Name
- Price
- Start time
- End time
- Instructor name
- Class ID
- Exercise Name

After:

SQLQuery2.sql - (l...B3CRBI2\chann (63)) * X SQLQuery1.sql - (l...B3CRBI2\chann (73)) *

Select * from Class

100 %

Results Messages

	Class_ID	Price	Name	Start_time	End_time	Instructor_name	Exercise_Name
1	Class 9	40	C9	11:11:00	11:11:00	Stormi	Bench Press
2	Class1	40	C1	10:00:00	11:00:00	Anon Atom	Bench Press
3	Class2	15	C2	11:00:00	12:00:00	Anon Beaker	Squat
4	Class3	10	C3	12:00:00	13:00:00	Anon Beaker	Deadlift
5	Class4	30	C4	13:00:00	14:00:00	Anon Atom	Curls
6	Class5	1	C5	14:00:00	15:00:00	Anon Comp	Shoulder Press
7	Class6	15	C6	12:00:00	13:00:00	Anon Book	Deadlift
8	Class7	50	C7	13:00:00	14:00:00	Anon Book	Curls
9	Class8	20	C8	14:00:00	15:00:00	Anon Atom	Shoulder Press

Query b. DELETE

File:

project_b4e3b_c1i3b_s4v2b\Server\GymManagement.Infrastructure\Persistence\ExerciseRepository.cs

Line: 36 - 50

Before:

NOTE: Deleting an exercise affects a class with that exercise too. Hence, screenshots will include both tables.

Gym Manager

USERS
EXERCISES
CLASS
TARGETS

exerciseName	numberOfSets
Bench Press	5
Curls	5
Deadlift	3
Shoulder Press	4
Squat	5

Show Exercises That Target Everything

Gym Manager

USERS
EXERCISES
CLASS
TARGETS

Add New Entry / View Insights

name	start_time	end_time	instructor_name	exercise_Name
C1	10:00:00	11:00:00	Anon Atom	Bench Press
C2	11:00:00	12:00:00	Anon Beaker	Squat
C3	12:00:00	13:00:00	Anon Beaker	Deadlift
C4	13:00:00	14:00:00	Anon Atom	Curls
C5	14:00:00	15:00:00	Anon Comp	Shoulder Press
C6	12:00:00	13:00:00	Anon Book	Deadlift
C7	13:00:00	14:00:00	Anon Book	Curls
C8	14:00:00	15:00:00	Anon Atom	Shoulder Press

Select classes that teach:

- Name
- Price
- Start time
- End time
- Instructor name
- Class ID
- Exercise Name

SQLQuery2.sql - (l...B3CRBI2\chann (63))*

SQLQuery1.sql - (l...B3CRBI2\chann (73))*

```
Select * from Does_Exercise;
Select * from Class;
```

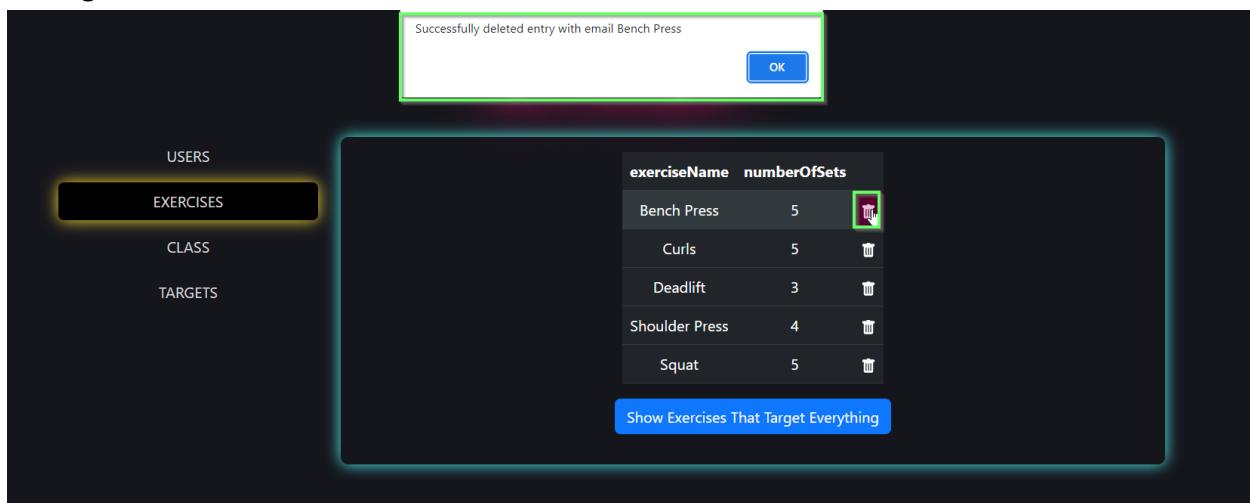
100 %

Results Messages

	Number_of_sets	Exercise_Name
1	5	Bench Press
2	5	Curls
3	3	Deadlift
4	4	Shoulder Press
5	5	Squat

	Class_ID	Price	Name	Start_time	End_time	Instructor_name	Exercise_Name
1	Class1	40	C1	10:00:00	11:00:00	Anon Atom	Bench Press
2	Class2	15	C2	11:00:00	12:00:00	Anon Beaker	Squat
3	Class3	10	C3	12:00:00	13:00:00	Anon Beaker	Deadlift
4	Class4	30	C4	13:00:00	14:00:00	Anon Atom	Curls
5	Class5	1	C5	14:00:00	15:00:00	Anon Comp	Shoulder Press
6	Class6	15	C6	12:00:00	13:00:00	Anon Book	Deadlift
7	Class7	50	C7	13:00:00	14:00:00	Anon Book	Curls
8	Class8	20	C8	14:00:00	15:00:00	Anon Atom	Shoulder Press

During:



After:

Gym Manager

- USERS
- EXERCISES**
- CLASS
- TARGETS

exerciseName	numberOfSets
Curls	5
Deadlift	3
Shoulder Press	4
Squat	5

Show Exercises That Target Everything

Gym Manager

- USERS
- EXERCISES
- CLASS**
- TARGETS

Add New Entry / View Insights				
name	start_time	end_time	instructor_name	exercise_Name
C2	11:00:00	12:00:00	Anon Beaker	Squat
C3	12:00:00	13:00:00	Anon Beaker	Deadlift
C4	13:00:00	14:00:00	Anon Atom	Curls
C5	14:00:00	15:00:00	Anon Comp	Shoulder Press
C6	12:00:00	13:00:00	Anon Book	Deadlift
C7	13:00:00	14:00:00	Anon Book	Curls
C8	14:00:00	15:00:00	Anon Atom	Shoulder Press

Select classes that teach:

- Name
- Price
- Start time
- End time
- Instructor name
- Class ID
- Exercise Name

SQLQuery2.sql - (...B3CRBI2\chann (63))*

SQLQuery1.sql - (...B3CRBI2\chann (73))*

```

Select * from Does_Exercise;
Select * from Class;

```

100 %

Results Messages

	Number_of_sets	Exercise_Name
1	5	Curls
2	3	Deadlift
3	4	Shoulder Press
4	5	Squat

	Class_ID	Price	Name	Start_time	End_time	Instructor_name	Exercise_Name
1	Class2	15	C2	11:00:00	12:00:00	Anon Beaker	Squat
2	Class3	10	C3	12:00:00	13:00:00	Anon Beaker	Deadlift
3	Class4	30	C4	13:00:00	14:00:00	Anon Atom	Curls
4	Class5	1	C5	14:00:00	15:00:00	Anon Comp	Shoulder Press
5	Class6	15	C6	12:00:00	13:00:00	Anon Book	Deadlift
6	Class7	50	C7	13:00:00	14:00:00	Anon Book	Curls
7	Class8	20	C8	14:00:00	15:00:00	Anon Atom	Shoulder Press

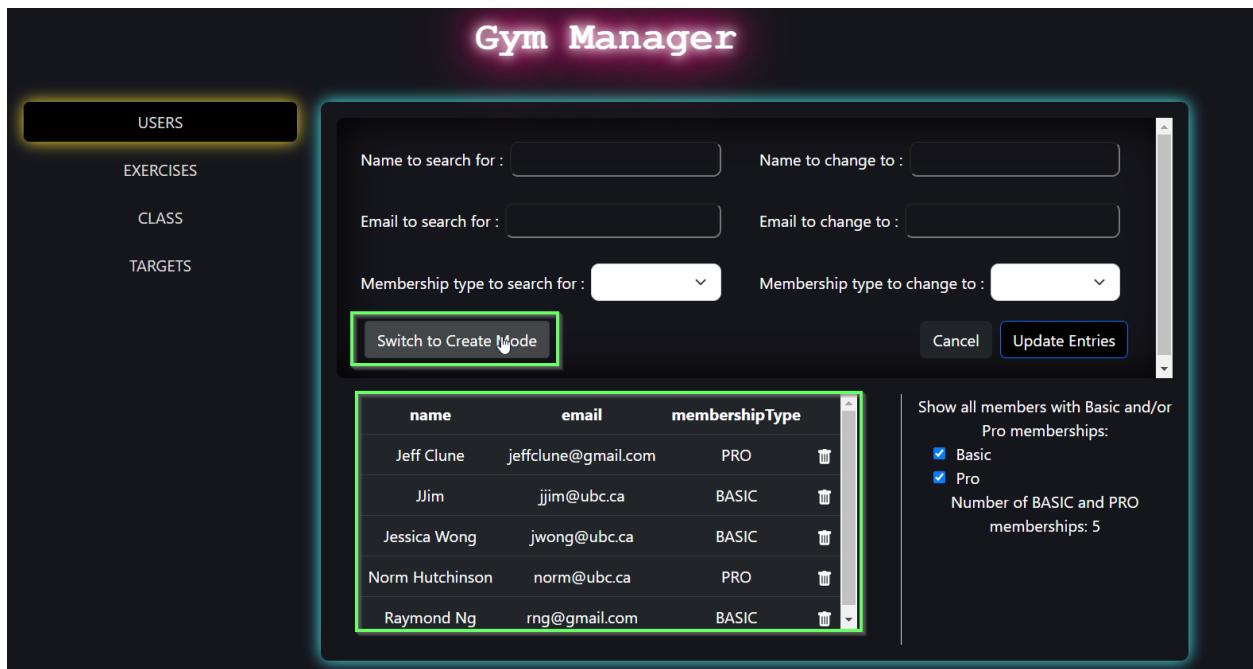
Query c. UPDATE

File:

project_b4e3b_c13b_s4v2b\Server\GymManagement.Infrastructure\Persistence\UserRepository.cs

Line: 48-89

Before:



The screenshot shows the SSMS interface with two tabs: 'SQLQuery2.sql' and 'SQLQuery1.sql'. The 'Results' tab displays the output of the query 'Select * from Users'. The data is:

	Name	Email	MembershipType
1	Jeff Clune	jeffclune@gmail.com	PRO
2	JJim	jjim@ubc.ca	BASIC
3	Jessica Wong	jwong@ubc.ca	BASIC
4	Norm Hutchinson	norm@ubc.ca	PRO
5	Raymond Ng	rng@gmail.com	BASIC

During:

The screenshot shows the 'Gym Manager' application interface. On the left, a sidebar menu includes 'USERS' (highlighted in yellow), 'EXERCISES', 'CLASS', and 'TARGETS'. The main area is titled 'Gym Manager' and contains search and filter fields: 'Name to search for : Jeff Clune', 'Name to change to : Stormi', 'Email to search for : jeffclune@gmail.com', 'Email to change to : Stormi@gmail.com', 'Membership type to search for : PRO', and 'Membership type to change to : BASIC'. Below these are buttons for 'Switch to Create Mode', 'Cancel', and 'Update Entries' (highlighted in blue). A table lists member data with columns 'name', 'email', and 'membershipType'. The table shows five rows: Jeff Clune (PRO), JJim (BASIC), Jessica Wong (BASIC), Norm Hutchinson (PRO), and Raymond Ng (BASIC). To the right, a sidebar displays membership statistics: 'Show all members with Basic and/or Pro memberships:', checked boxes for 'Basic' and 'Pro', and 'Number of BASIC and PRO memberships: 5'.

name	email	membershipType
Jeff Clune	jeffclune@gmail.com	PRO
JJim	jjim@ubc.ca	BASIC
Jessica Wong	jwong@ubc.ca	BASIC
Norm Hutchinson	norm@ubc.ca	PRO
Raymond Ng	rng@gmail.com	BASIC

After:

The screenshot shows the 'Gym Manager' application interface after the update. The sidebar menu remains the same. The main area is titled 'Gym Manager' and shows an 'Add or Update Entry' form. The table below it lists member data with columns 'name', 'email', and 'membershipType'. The table now shows six rows: JJim (BASIC), Jessica Wong (BASIC), Norm Hutchinson (PRO), Raymond Ng (BASIC), Stormi (BASIC), and a new entry 'Stormi' (BASIC) added in the last row. To the right, the sidebar displays the same membership statistics as before.

name	email	membershipType
JJim	jjim@ubc.ca	BASIC
Jessica Wong	jwong@ubc.ca	BASIC
Norm Hutchinson	norm@ubc.ca	PRO
Raymond Ng	rng@gmail.com	BASIC
Stormi	Stormi@gmail.com	BASIC

The screenshot shows the 'SQL Server Management Studio' interface with a query window titled 'SQLQuery2.sql'. The query 'Select * from Users' is executed, and the results are displayed in a table. The table has columns 'Name', 'Email', and 'MembershipType'. The data matches the updated list in the Gym Manager application, showing six entries: JJim (BASIC), Jessica Wong (BASIC), Norm Hutchinson (PRO), Raymond Ng (BASIC), Stormi (BASIC), and a new entry 'Stormi' (BASIC).

	Name	Email	MembershipType
1	JJim	jjim@ubc.ca	BASIC
2	Jessica Wong	jwong@ubc.ca	BASIC
3	Norm Hutchinson	norm@ubc.ca	PRO
4	Raymond Ng	rng@gmail.com	BASIC
5	Stormi	Stormi@gmail.com	BASIC

Query d. Selection

File:

project_b4e3b_c1i3b_s4v2b\Server\GymManagement.Infrastructure\Persistence\UserRepository.cs

Line: 129 - 189

Before:

Gym Manager

USERS

EXERCISES

CLASS

TARGETS

Add or Update Entry

name	email	membershipType	
Jeff Clune	jeffclune@gmail.com	PRO	trash
JJim	jim@ubc.ca	BASIC	trash
Jessica Wong	jwong@ubc.ca	BASIC	trash
Norm Hutchinson	norm@ubc.ca	PRO	trash
Raymond Ng	rng@gmail.com	BASIC	trash

Show all members with Basic and/or Pro memberships:
 Basic
 Pro
Number of BASIC and PRO memberships: 5

SQLQuery2.sql - (l...B3CRBI2\chann (63)) X SQLQuery1.s

Select * from Users

	Name	Email	MembershipType
1	Jeff Clune	jeffclune@gmail.com	PRO
2	JJim	jim@ubc.ca	BASIC
3	Jessica Wong	jwong@ubc.ca	BASIC
4	Norm Hutchinson	norm@ubc.ca	PRO
5	Raymond Ng	rng@gmail.com	BASIC

During:

Gym Manager

USERS

EXERCISES

CLASS

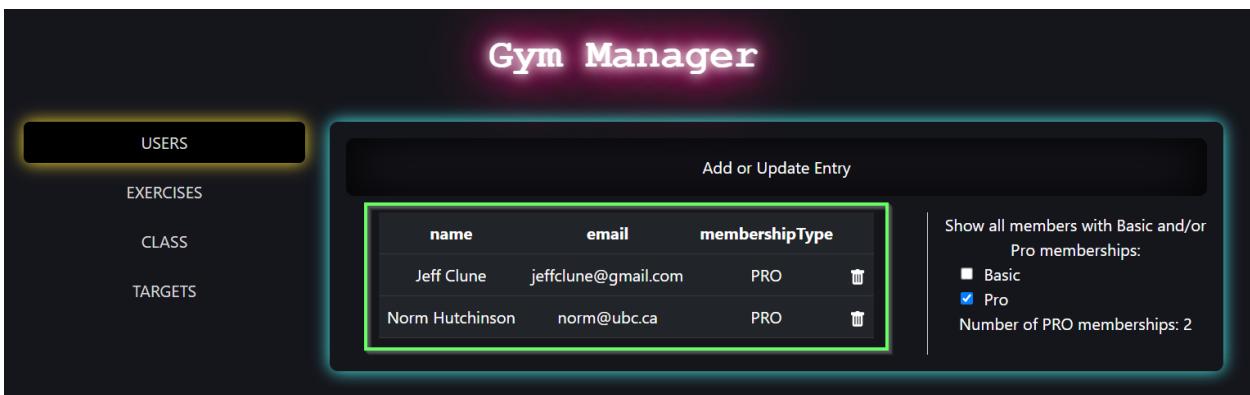
TARGETS

Add or Update Entry

name	email	membershipType	
Jeff Clune	jeffclune@gmail.com	PRO	trash
JJim	jim@ubc.ca	BASIC	trash
Jessica Wong	jwong@ubc.ca	BASIC	trash
Norm Hutchinson	norm@ubc.ca	PRO	trash
Raymond Ng	rng@gmail.com	BASIC	trash

Show all members with Basic and/or Pro memberships:
 Basic
 Pro
Number of BASIC and PRO memberships: 5

After:



The screenshot shows the 'SQLQuery2.sql' window in SQL Server Management Studio. The query 'Select * from Users' has been run, and the results are displayed in a table:

	Name	Email	MembershipType
1	Jeff Clune	jeffclune@gmail.com	PRO
2	JJim	jim@ubc.ca	BASIC
3	Jessica Wong	jwong@ubc.ca	BASIC
4	Norm Hutchinson	norm@ubc.ca	PRO
5	Raymond Ng	rng@gmail.com	BASIC

Query e. Projection

File:

project_b4e3b_c13b_s4v2b\Server\GymManagement.Infrastructure\Persistence\ClassRepository.cs

Line: 42-111

Before:

The screenshot shows a dark-themed application window titled "Gym Manager". On the left, there's a sidebar with tabs: "USERS", "EXERCISES", "CLASS" (which is highlighted), and "TARGETS". The main area has a title "Add New Entry / View Insights" and a table containing 8 rows of class data. The columns are: name, price(\$), start_time, end_time, instructor_name, class_ID, and exercise_name. The data looks like this:

	name	price(\$)	start_time	end_time	instructor_name	class_ID	exercise_name
1	C1	40	10:00:00	11:00:00	Anon Atom	Class1	Bench Press
2	C2	15	11:00:00	12:00:00	Anon Beaker	Class2	Squat
3	C3	10	12:00:00	13:00:00	Anon Beaker	Class3	Deadlift
4	C4	30	13:00:00	14:00:00	Anon Atom	Class4	Curls
5	C5	1	14:00:00	15:00:00	Anon Comp	Class5	Shoulder Press
6	C6	15	12:00:00	13:00:00	Anon Book	Class6	Deadlift
7	C7	50	13:00:00	14:00:00	Anon Book	Class7	Curls
8	C8	20	14:00:00	15:00:00	Anon Atom	Class8	Shoulder Press

To the right of the table is a sidebar with the heading "Select classes that teach:" followed by a list of checkboxes:

- Name
- Price
- Start time
- End time
- Instructor name
- Class ID
- Exercise Name

The screenshot shows the SQL Server Management Studio interface with two panes: "SQLQuery2.sql" and "SQLQuery1.sql". The "SQLQuery2.sql" pane contains the query "Select * from Class". The results pane shows the same data as the application screenshot, with 8 rows of class information. The columns are: Class_ID, Price, Name, Start_time, End_time, Instructor_name, and Exercise_Name.

	Class_ID	Price	Name	Start_time	End_time	Instructor_name	Exercise_Name
1	Class1	40	C1	10:00:00	11:00:00	Anon Atom	Bench Press
2	Class2	15	C2	11:00:00	12:00:00	Anon Beaker	Squat
3	Class3	10	C3	12:00:00	13:00:00	Anon Beaker	Deadlift
4	Class4	30	C4	13:00:00	14:00:00	Anon Atom	Curls
5	Class5	1	C5	14:00:00	15:00:00	Anon Comp	Shoulder Press
6	Class6	15	C6	12:00:00	13:00:00	Anon Book	Deadlift
7	Class7	50	C7	13:00:00	14:00:00	Anon Book	Curls
8	Class8	20	C8	14:00:00	15:00:00	Anon Atom	Shoulder Press

During:

Gym Manager

Add New Entry / View Insights

name	price(\$)	start_time	end_time	instructor_name	class_ID	exercise_name
C1	40	10:00:00	11:00:00	Anon Atom	Class1	Bench Press
C2	15	11:00:00	12:00:00	Anon Beaker	Class2	Squat
C3	10	12:00:00	13:00:00	Anon Beaker	Class3	Deadlift
C4	30	13:00:00	14:00:00	Anon Atom	Class4	Curls
C5	1	14:00:00	15:00:00	Anon Comp	Class5	Shoulder Press
C6	15	12:00:00	13:00:00	Anon Book	Class6	Deadlift
C7	50	13:00:00	14:00:00	Anon Book	Class7	Curls
C8	20	14:00:00	15:00:00	Anon Atom	Class8	Shoulder Press

Select classes that teach:

- Name
- Price
- Start time
- End time
- Instructor name
- Class ID
- Exercise Name

After:

Gym Manager

Add New Entry / View Insights

name	class_ID	exercise_name
C1	Class1	Bench Press
C2	Class2	Squat
C3	Class3	Deadlift
C4	Class4	Curls
C5	Class5	Shoulder Press
C6	Class6	Deadlift
C7	Class7	Curls
C8	Class8	Shoulder Press

Select classes that teach:

- Name
- Price
- Start time
- End time
- Instructor name
- Class ID
- Exercise Name

SQLQuery2.sql - (l...B3CRBI2\chann (63))*

SQLQuery1.sql - (l...B3CRBI2\chann (7)

```
Select * from Class
```

100 %

	Class_ID	Price	Name	Start_time	End_time	Instructor_name	Exercise_Name
1	Class1	40	C1	10:00:00	11:00:00	Anon Atom	Bench Press
2	Class2	15	C2	11:00:00	12:00:00	Anon Beaker	Squat
3	Class3	10	C3	12:00:00	13:00:00	Anon Beaker	Deadlift
4	Class4	30	C4	13:00:00	14:00:00	Anon Atom	Curls
5	Class5	1	C5	14:00:00	15:00:00	Anon Comp	Shoulder Press
6	Class6	15	C6	12:00:00	13:00:00	Anon Book	Deadlift
7	Class7	50	C7	13:00:00	14:00:00	Anon Book	Curls
8	Class8	20	C8	14:00:00	15:00:00	Anon Atom	Shoulder Press

Query f. Join

File:

project_b4e3b_c13b_s4v2b\Server\GymManagement.Infrastructure\Persistence\ClassRepository.cs

Line: 42-111

Before:



The screenshot shows the SQL Server Management Studio (SSMS) interface with two tabs: 'SQLQuery2.sql' and 'SQLQuery1.sql'. The 'SQLQuery2.sql' tab contains the query:

```
Select * from Class
```

The results pane shows the data from the 'Class' table:

	Class_ID	Price	Name	Start_time	End_time	Instructor_name	Exercise_Name
1	Class1	40	C1	10:00:00	11:00:00	Anon Atom	Bench Press
2	Class2	15	C2	11:00:00	12:00:00	Anon Beaker	Squat
3	Class3	10	C3	12:00:00	13:00:00	Anon Beaker	Deadlift
4	Class4	30	C4	13:00:00	14:00:00	Anon Atom	Curls
5	Class5	1	C5	14:00:00	15:00:00	Anon Comp	Shoulder Press
6	Class6	15	C6	12:00:00	13:00:00	Anon Book	Deadlift
7	Class7	50	C7	13:00:00	14:00:00	Anon Book	Curls
8	Class8	20	C8	14:00:00	15:00:00	Anon Atom	Shoulder Press

During:

The screenshot shows a dark-themed application window titled "Gym Manager". On the left, a sidebar lists "USERS", "EXERCISES", "CLASS" (which is highlighted with a yellow glow), and "TARGETS". The main area has a title bar "Add New Entry / View Insights" and contains a table with columns "name", "class_ID", and "exercise_Name". The table data is as follows:

	name	class_ID	exercise_Name
1	C1	Class1	Bench Press
2	C2	Class2	Squat
3	C3	Class3	Deadlift
4	C4	Class4	Curls
5	C5	Class5	Shoulder Press
6	C6	Class6	Deadlift
7	C7	Class7	Curls
8	C8	Class8	Shoulder Press

To the right of the table is a sidebar titled "Select classes that teach:" containing checkboxes for various fields. The "Exercise Name" checkbox is checked.

After:

The screenshot shows the same application window after modification. The sidebar and table structure remain the same, but the "num_of_reps" column has been added to the table. The table data is now:

	name	class_ID	num_of_reps	exercise_Name
1	C2	Class2	5	Squat

The "Select classes that teach:" sidebar remains the same, with the "Exercise Name" checkbox checked.

A screenshot of SQL Server Management Studio (SSMS) showing the results of a query. The query is:

```
Select * from Class
```

The results grid displays the following data:

	Class_ID	Price	Name	Start_time	End_time	Instructor_name	Exercise_Name
1	Class1	40	C1	10:00:00	11:00:00	Anon Atom	Bench Press
2	Class2	15	C2	11:00:00	12:00:00	Anon Beaker	Squat
3	Class3	10	C3	12:00:00	13:00:00	Anon Beaker	Deadlift
4	Class4	30	C4	13:00:00	14:00:00	Anon Atom	Curls
5	Class5	1	C5	14:00:00	15:00:00	Anon Comp	Shoulder Press
6	Class6	15	C6	12:00:00	13:00:00	Anon Book	Deadlift
7	Class7	50	C7	13:00:00	14:00:00	Anon Book	Curls
8	Class8	20	C8	14:00:00	15:00:00	Anon Atom	Shoulder Press

Query g. Aggregation w/ Group By

File:

project_b4e3b_c13b_s4v2b\Server\GymManagement.Infrastructure\Persistence\UserRepository.cs

Line: 129-189

Before:

The screenshot shows the 'Gym Manager' application interface. On the left is a sidebar with 'USERS' selected, and other options like 'EXERCISES', 'CLASS', and 'TARGETS'. The main area is titled 'Add or Update Entry' and displays a table of user data:

	name	email	membershipType
Jeff Clune	jeffclune@gmail.com	PRO	
JJim	jjim@ubc.ca	BASIC	
Jessica Wong	jwong@ubc.ca	BASIC	
Norm Hutchinson	norm@ubc.ca	PRO	
Raymond Ng	rng@gmail.com	BASIC	

To the right of the table is a sidebar with the following text:

Show all members with Basic and/or Pro memberships:
 Basic
 Pro
Number of BASIC and PRO memberships: 5

The screenshot shows the SSMS interface with a query window open. The query is:

```
Select * from Users
```

The results pane shows the following data:

	Name	Email	MembershipType
1	Jeff Clune	jeffclune@gmail.com	PRO
2	JJim	jjim@ubc.ca	BASIC
3	Jessica Wong	jwong@ubc.ca	BASIC
4	Norm Hutchinson	norm@ubc.ca	PRO
5	Raymond Ng	rng@gmail.com	BASIC

During:

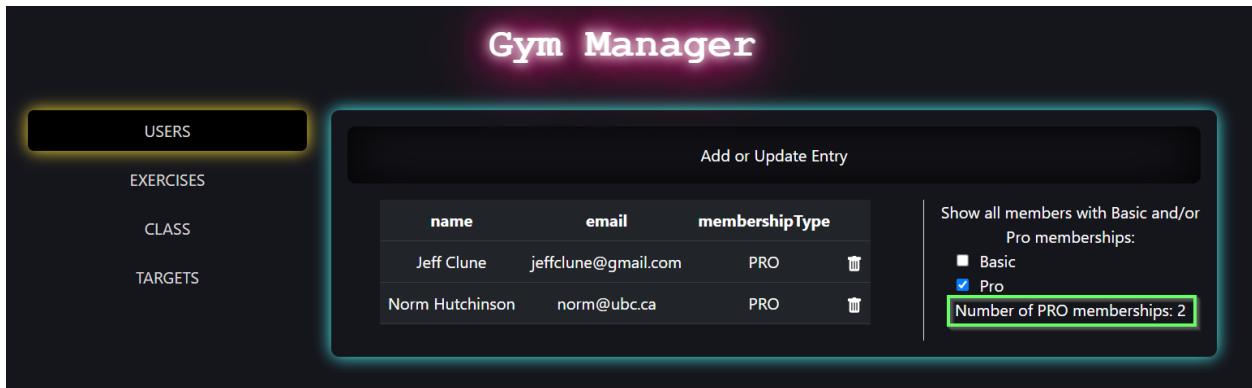
The screenshot shows the 'Gym Manager' application interface during development. The sidebar on the left has 'USERS' selected. The main area is titled 'Add or Update Entry' and displays a table of user data, identical to the 'Before' screenshot.

To the right of the table is a sidebar with the following text:

Show all members with Basic and/or Pro memberships:
 Basic
 Pro
Number of BASIC and PRO memberships: 5

The 'Basic' checkbox is highlighted with a green box.

After:



The screenshot shows the SSMS interface with a query window open. The query is:

```
Select * from Users
```

The results pane shows the following data:

	Name	Email	MembershipType
1	Jeff Clune	jeffclune@gmail.com	PRO
2	JJim	jim@ubc.ca	BASIC
3	Jessica Wong	jwong@ubc.ca	BASIC
4	Norm Hutchinson	norm@ubc.ca	PRO
5	Raymond Ng	rng@gmail.com	BASIC

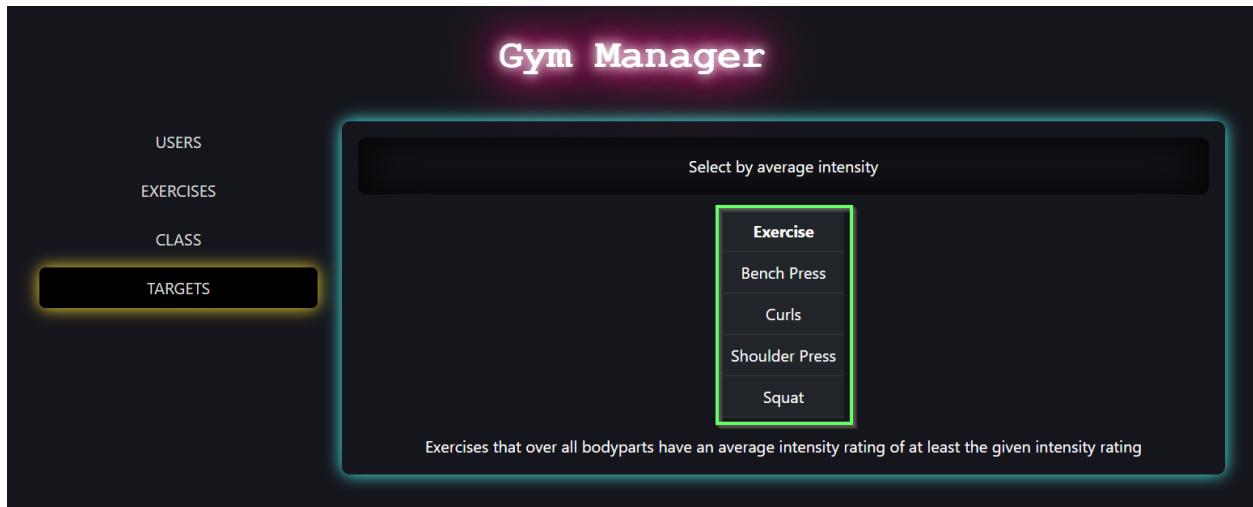
Query h. Aggregation w/ Having

File:

project_b4e3b_c13b_s4v2b\Server\GymManagement.Infrastructure\Persistence\TargetsRepository.cs

Line: 15-34

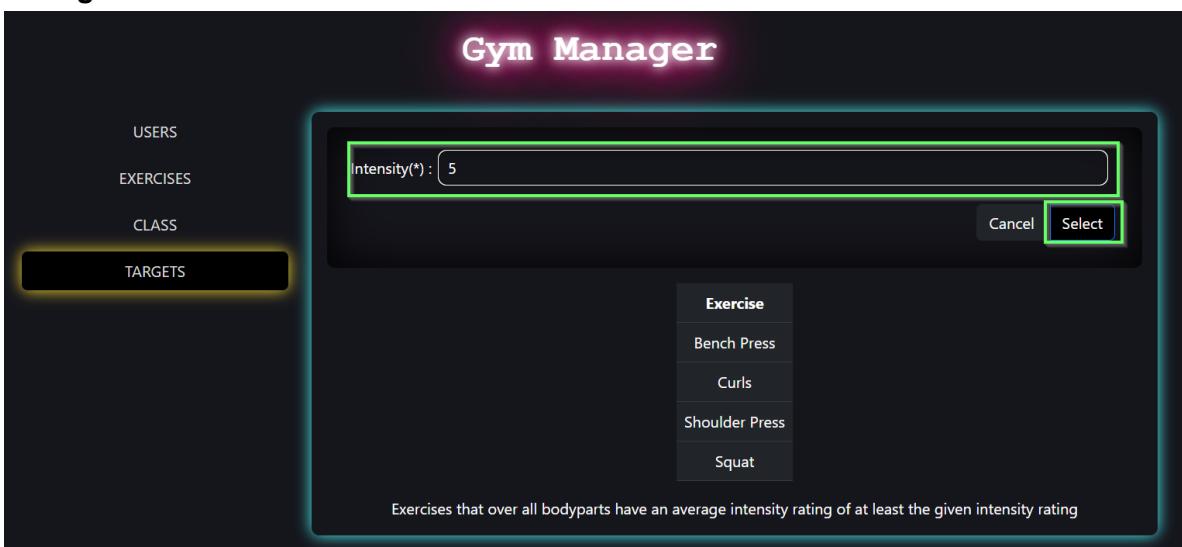
Before:



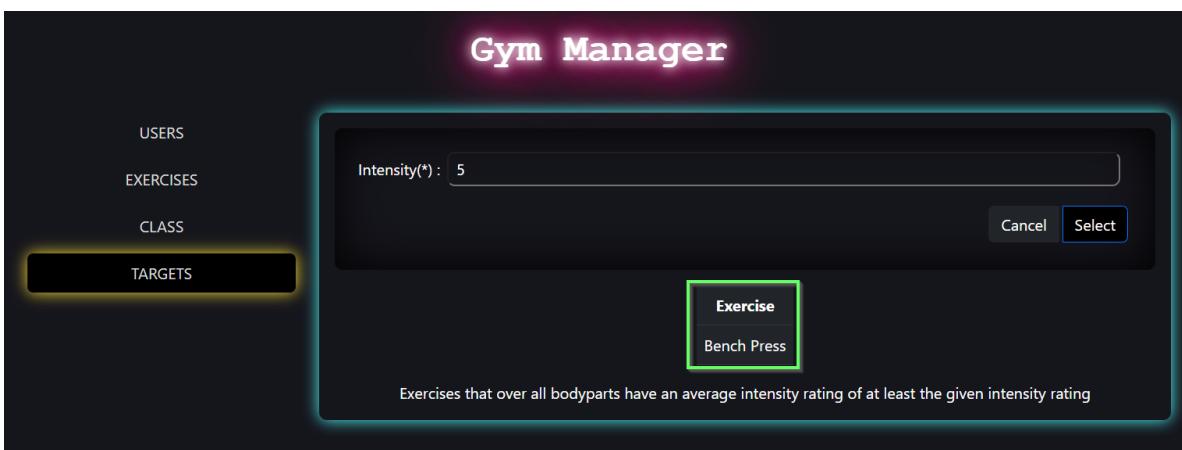
The screenshot shows the 'SQLQuery2.sql' query results in SQL Server Management Studio. The query is 'Select * from Targets'. The results grid displays the following data:

	Exercise_Name	BodyPart_name	Intensity_rating
1	Bench Press	Arms	7
2	Bench Press	Back	2
3	Bench Press	Chest	9
4	Bench Press	Legs	2
5	Bench Press	Shoulders	7
6	Curls	Arms	2
7	Shoulder Press	Shoulders	2
8	Squat	Arms	2
9	Squat	Back	5
10	Squat	Legs	10
11	Squat	Shoulders	2

During:



After:



A screenshot of SQL Server Management Studio (SQLOutput) showing the results of a query. The query is: `Select * from Targets`. The results grid displays the following data:

	Exercise_Name	BodyPart_name	Intensity_rating
1	Bench Press	Arms	7
2	Bench Press	Back	2
3	Bench Press	Chest	9
4	Bench Press	Legs	2
5	Bench Press	Shoulders	7
6	Curls	Arms	2
7	Shoulder Press	Shoulders	2
8	Squat	Arms	2
9	Squat	Back	5
10	Squat	Legs	10
11	Squat	Shoulders	2

Query i. Nested Aggregation w/ Group By

File:

project_b4e3b_c1i3b_s4v2b\Server\GymManagement.Infrastructure\Persistence\ClassRepository.cs

Line: 113-154

Before:

The screenshot shows a dark-themed application window titled "Gym Manager". On the left, there is a navigation menu with tabs: "USERS", "EXERCISES", "CLASS" (which is highlighted in yellow), and "TARGETS". The main area contains a table with the following data:

name	price(\$)	start_time	end_time	instructor_name	class_ID	ex
C1	40	10:00:00	11:00:00	Anon Atom	Class1	B
C2	15	11:00:00	12:00:00	Anon Beaker	Class2	
C3	10	12:00:00	13:00:00	Anon Beaker	Class3	
C4	30	13:00:00	14:00:00	Anon Atom	Class4	
C5	1	14:00:00	15:00:00	Anon Comp	Class5	Sh
C6	15	12:00:00	13:00:00	Anon Book	Class6	
C7	50	13:00:00	14:00:00	Anon Book	Class7	
C8	20	14:00:00	15:00:00	Anon Atom	Class8	Sh

To the right of the table is a sidebar with the heading "Select classes that teach:" followed by a list of checkboxes:

- Name
- Price
- Start time
- End time
- Instructor name
- Class ID
- Exercise Name

The screenshot shows the SQL Server Management Studio interface with two panes: "SQLQuery2.sql" and "SQLQuery1.sql". The "SQLQuery2.sql" pane contains the query:

```
Select * from Class
```

The results pane shows the following data:

	Class_ID	Price	Name	Start_time	End_time	Instructor_name	Exercise_Name
1	Class1	40	C1	10:00:00	11:00:00	Anon Atom	Bench Press
2	Class2	15	C2	11:00:00	12:00:00	Anon Beaker	Squat
3	Class3	10	C3	12:00:00	13:00:00	Anon Beaker	Deadlift
4	Class4	30	C4	13:00:00	14:00:00	Anon Atom	Curls
5	Class5	1	C5	14:00:00	15:00:00	Anon Comp	Shoulder Press
6	Class6	15	C6	12:00:00	13:00:00	Anon Book	Deadlift
7	Class7	50	C7	13:00:00	14:00:00	Anon Book	Curls
8	Class8	20	C8	14:00:00	15:00:00	Anon Atom	Shoulder Press

During:

Gym Manager

Class Cost Estimator

Instructor popularity rating : 2

Note: popularity is counted by the amount of classes that an instructor teaches.

Switch to Create Mode Refresh Cancel Submit

name	price(\$)	start_time	end_time	instructor_name	class_ID
C1	40	10:00:00	11:00:00	Anon Atom	Class1
C2	15	11:00:00	12:00:00	Anon Beaker	Class2
C3	10	12:00:00	13:00:00	Anon Beaker	Class3
C4	30	13:00:00	14:00:00	Anon Atom	Class4
C5	1	14:00:00	15:00:00	Anon Comp	Class5
C6	15	12:00:00	13:00:00	Anon Book	Class6

Select classes that teach:

- Name
- Price
- Start time
- End time
- Instructor name
- Class ID
- Exercise Name

After:

Gym Manager

Class Cost Estimator

Instructor popularity rating :

The cheapest class offered by instructors with popularity 2 is \$10

Note: popularity is counted by the amount of classes that an instructor teaches.

Switch to Create Mode Refresh Cancel Submit

name	price(\$)	start_time	end_time	instructor_name	class_ID
C1	40	10:00:00	11:00:00	Anon Atom	Class1
C2	15	11:00:00	12:00:00	Anon Beaker	Class2
C3	10	12:00:00	13:00:00	Anon Beaker	Class3
C4	30	13:00:00	14:00:00	Anon Atom	Class4
C5	1	14:00:00	15:00:00	Anon Comp	Class5
C6	15	12:00:00	13:00:00	Anon Book	Class6

Select classes that teach:

- Name
- Price
- Start time
- End time
- Instructor name
- Class ID
- Exercise Name

SQLQuery2.sql - (L...B3CRBI2\chann (63)) * SQLQuery1.sql - (L...B3CRBI2\chann (7)

```
Select * from Class
```

100 %

Results Messages

Class_ID	Price	Name	Start_time	End_time	Instructor_name	Exercise_Name	
1	Class1	40	C1	10:00:00	11:00:00	Anon Atom	Bench Press
2	Class2	15	C2	11:00:00	12:00:00	Anon Beaker	Squat
3	Class3	10	C3	12:00:00	13:00:00	Anon Beaker	Deadlift
4	Class4	30	C4	13:00:00	14:00:00	Anon Atom	Curls
5	Class5	1	C5	14:00:00	15:00:00	Anon Comp	Shoulder Press
6	Class6	15	C6	12:00:00	13:00:00	Anon Book	Deadlift
7	Class7	50	C7	13:00:00	14:00:00	Anon Book	Curls
8	Class8	20	C8	14:00:00	15:00:00	Anon Atom	Shoulder Press

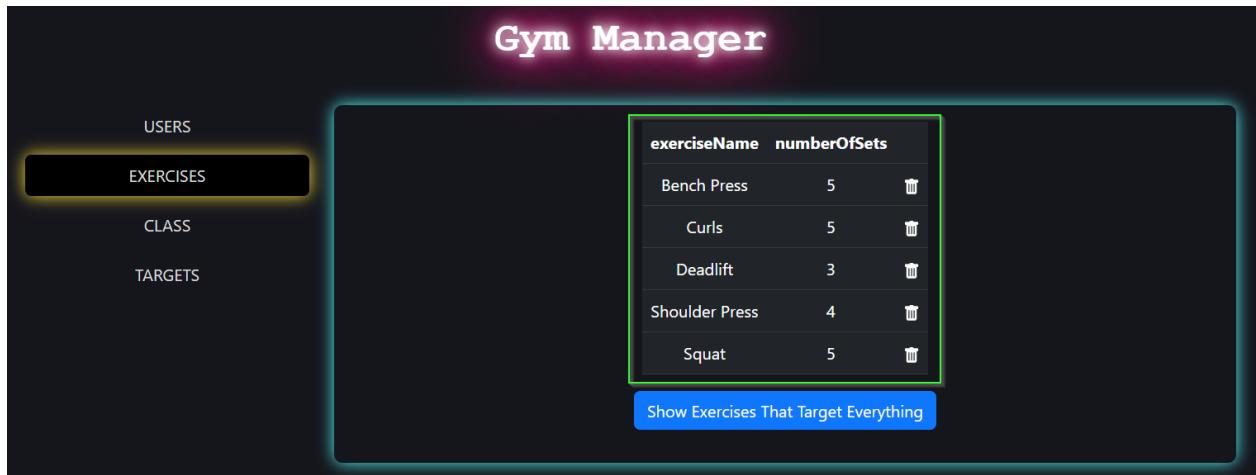
Query j. Division

File:

project_b4e3b_c13b_s4v2b\Server\GymManagement.Infrastructure\Persistence\ExerciseRepository.cs

Line: 52-78

Before:



The screenshot shows a SQL query window titled "SQLQuery2.sql - (l...B3CRBI2\chann (63))". The query is:

```
Select * from Does_Exercise
```

The results pane shows the following data:

	Number_of_sets	Exercise_Name
1	5	Bench Press
2	5	Curls
3	3	Deadlift
4	4	Shoulder Press
5	5	Squat

During:

Gym Manager

USERS

EXERCISES

CLASS

TARGETS

exerciseName	numberOfSets	trash
Bench Press	5	trash
Curls	5	trash
Deadlift	3	trash
Shoulder Press	4	trash
Squat	5	trash

Show Exercises That Target Everything

After:

Gym Manager

USERS

EXERCISES

CLASS

TARGETS

exerciseName	numberOfSets	trash
Bench Press	5	trash
Curls	5	trash
Deadlift	3	trash
Shoulder Press	4	trash
Squat	5	trash

Hide Exercises That Target Everything

exerciseName	numberOfSets	trash
Bench Press	5	trash

SQLQuery2.sql - (l...B3CRBI2\chann (63))*

```
Select * from Does_Exercise|
```

100 %

Results Messages

	Number_of_sets	Exercise_Name
1	5	Bench Press
2	5	Curls
3	3	Deadlift
4	4	Shoulder Press
5	5	Squat

9. README.txt

Please refer to the ./README.md in the root of the repository

Specifically, the changes section is under the `## Changes since Milestone 2` header.