

# COIS 3400 Group Project

---

## Athletics Centre Database

---

Prepared by Nana Aba Ackah(0629227) & Batsirai Chayire (0683438)

---

### COIS 3400 Group Project

#### Athletics Centre Database

Prepared by Nana Aba Ackah(0629227) & Batsirai Chayire (0683438)

Introduction

Description of Athletics center

- I. Memberships
- II. Sports Team
- III. Employee
- IV. Room
- V. Programs
- VI. Equipment

Solution

1. Find and list all entities.
  - a. Membership;
  - b. Sports Team
  - c. Employee
  - d. Room
  - e. Member
  - f. Program
  - g. Equipment
2. Find the attributes of entities.
  - a. Member; memberid, fname, lname, email(FK), membership\_id(FK), join\_date, email, pnumber, address, emergency\_contact
  - b. Sports Team; teamid, name, player\_name, coach, graduation\_year
  - c. Employee; emp\_SSN, name, pnumber, email, role, salary
  - d. Room; room\_id, name, floor, type, capacity
  - e. Membership; membership\_id, memberid(FK), price, duration
  - f. Program; program\_id, type, trainer, room, day, time
  - g. Equipment; equipment\_id, type, quantity
3. Define the data type for each attribute.
4. Define relationships between entities.
  - a. Member -> Membership 1 to 1
  - b. Member -> program Many to Many
  - c. Program-> Room Many to Many
  - d. Employee -> program 1 to Many
  - e. Room -> equipment 1 to Many
  - f. Employee -> Team 1 to Many
  - g. Sports Team -> Room 1 to Many
5. Create UML or ER model of your database.
6. Convert the UML or ER model to relational data model (database schemas). Show Primary Keys and Foreign Keys for each table (total 5-7 tables).
  - a. Member(memberid, membership\_id(FK), email, fname, lname, join\_date, number, address, emergency\_contact)
  - c. Membership(membership\_id, memberid(FK), price, duration)
  - d. Program(program\_id, room(FK), trainer, day, time)
  - e. Room(room\_id, name, floor, type, capacity)

- f. Employee(emp\_SSN, name, role, salary, pnumber, email)
  - g. Sports Team(team\_id, name, player\_name, coach, graduation\_year)
  - h. Equipment(equipment\_id, type, quantity)
7. Design necessary queries by the following requirements:
- I. Five queries for one table;
    - a. Individuals that play for a team
    - b. Employees that are trainers
    - c. Members that joined after a specific date
    - d. Capacity for room
    - e. Number of student memberships
  - II. Queries for two tables
    - a. Employees that lead a specific program
    - b. Equipment in a specific room
    - c. Employees that coach a sports team
    - d. Members paying a certain amount
    - e. Room a program would be in
  - III. Queries for three tables
    - a. Members that are students that are registered for a program
    - b. Teams trained by employee in specific room
    - c. Programs in specific room with equipment
    - d. Members in a specific program that joined after ... date that their membership duration is
    - e. Member that uses specific program where they pay a specific price.
8. SQL commands to create tables and input data
9. Write the queries using SQL statements. Show the output of your SQL queries using screen shot.
- I. Five queries for one table;
    - a. Number of Teams
    - b. Employees that are trainers
    - c. Members that joined on a specific date
    - d. Capacity for room
    - e. Number of student memberships
  - II. Queries for two tables
    - a. Employees that lead a specific program
    - b. Equipment in a specific room
    - c. Employees that coach a sports team
    - d. Members paying a certain amount
    - e. Room a program would be in
  - III. Queries for three tables
    - a. Members that are students that are registered for a program
    - b. Teams trained by employee in specific room
    - c. Programs in specific room with equipment
    - d. Members in a specific program that joined after ... date that their membership duration is
    - e. Member that uses specific program where they pay a specific price.

## Introduction

Our project is based on the database of a university's athletics facility. We used different concepts taught in the course to create this database, including designing and creating a relational database, developing database schemas, composing SQL queries, and manipulating tables using SQL. The athletics center has different departments and offers a range of sports and other activities.

# Description of Athletics center

## I. Memberships

Current university students and people with memberships are welcome to use the athletic center. Individuals who pay a membership fee are given access to the athletics center. Students do not need to pay for membership because it is already covered by their tuition. The cost of membership is determined by the length of time it will be used. The costs are \$25 for one month, \$75 for three months, \$150 for six months, and \$300 for one year.

## II. Sports Team

The database keeps track of students registered for the different sport teams. The sports teams include basketball, volleyball, lacrosse, soccer, rowing and swimming. It will keep track of the names of players in each sport, the year the student will graduate, the coach of the team and which room they use for training.

## III. Employee

The database will store information about the athletics center's employees. The information includes the employee's social security number, name, email, phone number, role at the athletics center, and salary.

## IV. Room

The database will maintain track of the details of the rooms that will be used by members and for training purposes. Each room has a different name and can be identified using its room ID. Other information would be which floor the room is located, capacity for each room, equipment in the room and the program ID for a program that is done in that specific room.

## V. Programs

There are other programs offered at the athletics center besides sports and personal workout. The database will store information about the different programs offered. These programs include Yoga, Pilates, Personal training and swimming. The information stored will include a program ID to identify each program, the type of program, the name of the person who trains the program, which room they train in, which day of the week they train and what time they train.

## VI. Equipment

The database will keep track of all the equipment in the athletics center. Each type of equipment will have an identification number, the quantity of the equipment type and which room the equipment is stored in,

## Solution

We designed and defined the database using the steps outlined below. We identified the entities required to collect the data for the athletics center. Then, for each entity, we determined the required entities and data types. The next step was to determine the relationships between the entities in order to create links in the database. Based on the defined entities and their attributes, we created a UML diagram. Converting the UML diagram into a database schema guided us in determining which table we needed to create, as well as identifying the primary keys and foreign keys.

## 1. Find and list all entities.

- a. Membership;
- b. Sports Team
- c. Employee
- d. Room
- e. Member
- f. Program
- g. Equipment

## 2. Find the attributes of entities.

- a. Member; memberid, fname, lname, email(FK), membership\_id(FK), join\_date, email, pnumber, address, emergency\_contact
- b. Sports Team; teamid, name, player\_name, coach, graduation\_year
- c. Employee; emp\_SSN, name, pnumber, email, role, salary
- d. Room; room\_id, name, floor, type, capacity
- e. Membership; membership\_id, memberid(FK), price, duration
- f. Program; program\_id, type, trainer, room, day, time
- g. Equipment; equipment\_id, type, quantity

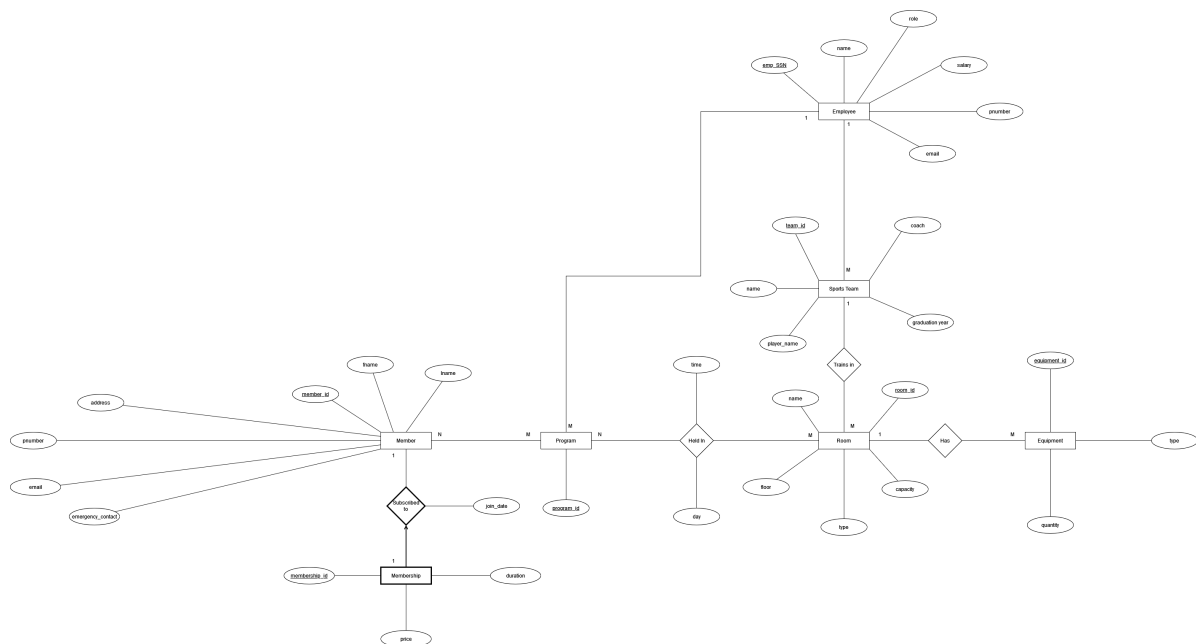
## 3. Define the data type for each attribute.

```
Member: memberid VARCHAR(10) PRIMARY KEY, membershipid VARCHAR(24),
        fname VARCHAR(15), lname VARCHAR(15),
        pnumber VARCHAR(15), address VARCHAR(255),
        email VARCHAR(255), emergency_contact VARCHAR(25),
        program VARCHAR(15), join_date VARCHAR(25));
Sports Team: teamid VARCHAR(15) PRIMARY KEY, name VARCHAR(15), coach VARCHAR(25),
Employee: emp_SSN VARCHAR(25) PRIMARY KEY, name VARCHAR(25),
        number VARCHAR(15), email VARCHAR(255), role VARCHAR(255),
        salary INTEGER(50));
Room: room_id VARCHAR(15) PRIMARY KEY, name VARCHAR(55), floor INTEGER(2),
      type VARCHAR (50), capacity INTEGER (3))
Membership: membership_id VARCHAR(10) PRIMARY KEY, member_id VARCHAR(24),
           price INTEGER(5), duration INTEGER(3))
Program: program_id VARCHAR(10) PRIMARY KEY, trainer VARCHAR(55),
        room VARCHAR(10), day VARCHAR(10), time time(6));
Equipment: equipment_id(equipment_id VARCHAR(25) PRIMARY KEY, type VARCHAR(20),
           quantity INTEGER(10));
```

#### 4. Define relationships between entities.

- a. Member -> Membership 1 to 1
- b. Member -> program Many to Many
- c. Program-> Room Many to Many
- d. Employee -> program 1 to Many
- e. Room -> equipment 1 to Many
- f. Employee -> Team 1 to Many
- g. Sports Team -> Room 1 to Many

#### 5. Create UML or ER model of your database.



#### 6. Convert the UML or ER model to relational data model (database schemas). Show Primary Keys and Foreign Keys for each table (total 5-7 tables).

- a. Member(memberid, membership\_id(FK), email, fname, lname, join\_date, number, address, emergency\_contact)
- c. Membership(membership\_id, memberid(FK), price, duration)
- d. Program(program\_id, room(FK), trainer, day, time)
- e. Room(room\_id, name, floor, type, capacity)
- f. Employee(emp\_SSN, name, role, salary, pnumber, email)

g. Sports Team(team\_id, name, player\_name, coach, graduation\_year)

h. Equipment(equipment\_id, type, quantity)

## 7. Design necessary queries by the following requirements:

### I. Five queries for one table;

- a. Individuals that play for a team
- b. Employees that are trainers
- c. Members that joined after a specific date
- d. Capacity for room
- e. Number of student memberships

### II. Queries for two tables

- a. Employees that lead a specific program
- b. Equipment in a specific room
- c. Employees that coach a sports team
- d. Members paying a certain amount
- e. Room a program would be in

### III. Queries for three tables

- a. Members that are students that are registered for a program
- b. Teams trained by employee in specific room
- c. Programs in specific room with equipment
- d. Members in a specific program that joined after ... date that their membership duration is
- e. Member that uses specific program where they pay a specific price.

## 8. SQL commands to create tables and input data

```
DROP TABLE IF EXISTS `Membership`;
CREATE TABLE Membership(membership_id VARCHAR(10) PRIMARY KEY, member_id
VARCHAR(24),
                        price INTEGER(5), duration INTEGER(3));
INSERT INTO Membership VALUES ('40-8419311', '94-7451558', '150', '128'),
                                ('35-7451558', '07-7012540', '75', '90'),
                                ('00-3037292', '74-5359443', '25', '30'),
                                ('35-7012540', '90-7996680', '75', '90'),
                                ('40-1296385', '83-1994311', '150', '128'),
                                ('40-5359443', '35-9698245', '150', '128'),
                                ('67-1248465', '17-8184794', '300', '365'),
                                ('67-9786394', '20-2416515', '300', '365'),
                                ('67-3951913', '64-9786394', '300', '365'),
                                ('00-8184794', '67-1248465', '25', '30'),
                                ('40-2416515', '09-3951913', '150', '128'),
                                ('40-7072073', '58-8419311', '150', '128'),
```

```

('35-1994311', '29-3037292', '75', '90'),
('35-9698245', '00-1296385', '75', '90');

DROP TABLE IF EXISTS `Member`;
CREATE TABLE Member (memberid VARCHAR(10) PRIMARY KEY, membershipid VARCHAR(24),
    fname VARCHAR(15), lname VARCHAR(15),
    pnumber VARCHAR(15), address VARCHAR(255),
    email VARCHAR(255), emergency_contact VARCHAR(25),
    program VARCHAR(15), join_date VARCHAR(25));

DROP TABLE IF EXISTS `Sports Team`;
CREATE TABLE `Sports Team` (teamid VARCHAR(15) PRIMARY KEY, name
    VARCHAR(15), coach VARCHAR(25));

DROP TABLE IF EXISTS `Employee`;
CREATE TABLE Employee (emp_SSN VARCHAR(25) PRIMARY KEY, name VARCHAR(25),
    number VARCHAR(15), email VARCHAR(255), role VARCHAR(255),
    salary INTEGER(50));

DROP TABLE IF EXISTS `Room`;
CREATE TABLE Room (room_id VARCHAR(15) PRIMARY KEY, name VARCHAR(55), floor
    INTEGER(2),
    type VARCHAR (50), capacity INTEGER (3));

DROP TABLE IF EXISTS `Program`;
CREATE TABLE Program (program_id VARCHAR(10) PRIMARY KEY, trainer VARCHAR(55),
    room VARCHAR(10), day VARCHAR(10), time time(6));

DROP TABLE IF EXISTS `Equipment`;
CREATE TABLE Equipment (equipment_id VARCHAR(25) PRIMARY KEY, type VARCHAR(20),
    quantity INTEGER(10));

```

## 9. Write the queries using SQL statements. Show the output of your SQL queries using screen shot.

### I. Five queries for one table;

#### a. Number of Teams

```
SELECT COUNT(name) FROM `Sports Team`;
```

Your SQL query has been executed successfully.

```
SELECT COUNT(name) FROM `Sports Team`
```

+ Options

COUNT(name)

## b. Employees that are trainers

```
SELECT name FROM Employee WHERE role LIKE '%trainer%';
```

✓ Showing rows 0 - 5 (6 total, Query took 0.0006 seconds.)

```
SELECT name FROM Employee WHERE role LIKE '%trainer%'
```

☐ Show all

Number of rows: 25 ▼

Filter rows:

Sort by key:  ▼

+ Options

name

Selestina Guynemer

Wylie Lumsden

Anastasie Fernihough

Swen Burkart

Henrieta Karim

Dahlia Vany

## c. Members that joined on a specific date

```
SELECT memberid, join_date FROM Member WHERE join_date = "05/03/2021"
```

✓ Showing rows 0 - 0 (1 total, Query took 0.0007 seconds.)

```
SELECT memberid, join_date FROM Member WHERE join_date = "05/03/2021"
```

☐ Profile

☐ Show all

Number of rows: 25 ▼

Filter rows:

+ Options

← T →

▼

memberid

join\_date

☐

Edit

Copy

Delete

00-1296385

05/03/2021





































## d. Capacity for room

```
SELECT room_id, capacity FROM Room WHERE capacity >= 25  
ORDER BY capacity DESC
```



Sort by key: None ▼

+ Options

						room_id	capacity	
<input type="checkbox"/>		Edit		Copy		Delete	AT27	94
<input type="checkbox"/>		Edit		Copy		Delete	AT05	93
<input type="checkbox"/>		Edit		Copy		Delete	AT09	87
<input type="checkbox"/>		Edit		Copy		Delete	AT79	81
<input type="checkbox"/>		Edit		Copy		Delete	AT57	80
<input type="checkbox"/>		Edit		Copy		Delete	AT12	72
<input type="checkbox"/>		Edit		Copy		Delete	AT34	66
<input type="checkbox"/>		Edit		Copy		Delete	AT82	61
<input type="checkbox"/>		Edit		Copy		Delete	AT74	44
<input type="checkbox"/>		Edit		Copy		Delete	AT28	35
<input type="checkbox"/>		Edit		Copy		Delete	AT16	33
<input type="checkbox"/>		Edit		Copy		Delete	AT93	28

### e. Number of student memberships

```
SELECT COUNT(membership_id) FROM Membership WHERE membership_id LIKE '67%';
```

Your SQL query has been executed successfully.

```
SELECT COUNT(membership_id) FROM Membership WHERE membership_id LIKE '67%';
```

☐ Profiling [ [Edit in](#)

+ Options

COUNT(membership\_id)

3

## II. Queries for two tables

### a. Employees that lead a specific program

```
SELECT E.name, P.program_id FROM Employee E, Program P WHERE E.name=P.trainer;
```

```
SELECT E.name, P.program_id FROM Employee E, Program P WHERE E.name=P.trainer
```

☐ Show all

Number of rows:

25

Filter rows:

+ Options

name	program_id
Selestina Guynemer	06-0092753
Selestina Guynemer	07-4236778
Dahlia Vany	19-6141286
Ad McHaffy	24-2386212
Wylie Lumsden	25-9447278
Ad McHaffy	26-7374937
Gretna Chapier	34-6858331
Dahlia Vany	52-1700420
Wylie Lumsden	54-9607283
Anastasie Fernihough	62-8220268
Dahlia Vany	64-8768200
Anastasie Fernihough	64-9069029
Gretna Chapier	67-7767546
Wylie Lumsden	81-1314617
Ad McHaffy	90-6950968

## b. Equipment in a specific room

```
SELECT R.name, Eq.type FROM Room R, Equipment Eq WHERE Eq.type = 'weights' AND R.room_id = 'AT34';
```

✓ Showing rows 0 - 0 (1 total, Query took 0.0008 seconds.)

```
SELECT R.name, Eq.type FROM Room R, Equipment Eq WHERE Eq.type = 'weights' AND R.room_id = 'AT34'
```

☐ Profiling [ [Edit inline](#) ] [

☐ Show all

Number of rows:

25

Filter rows:

+ Options

name	type
Kino	weights

## c. Employees that coach a sports team

```
SELECT E.name, S.name FROM Employee E, `Sports Team` S WHERE E.role LIKE '%coach%' AND S.name = 'Basketball';
```

✓ Showing rows 0 - 5 (6 total, Query took 0.0013 seconds.)

```
SELECT E.name, S.name FROM Employee E, `Sports Team` S WHERE E.role LIKE '%coach%' AND S.name = 'Basketball'
```

☐ Profiling [ Edit inline ] [ Edit ]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

name	name
Aridatha Minor	Basketball
Ernesto Kime	Basketball
Ad McHaffy	Basketball
Emmye Phizaclea	Basketball
Ricoriki Mungham	Basketball
Reinhold Collingdon	Basketball

#### d. Members paying a certain amount

```
SELECT M.memberid, MS.price FROM Member M, Membership MS WHERE  
MS.membership_id=M.membershipid AND MS.price > 50
```

✓ Showing rows 0 - 1 (2 total, Query took 0.0010 seconds.)

```
SELECT M.memberid, MS.price FROM Member M, Membership MS WHERE MS.membership_id=M.membershipid AND MS.price > 50
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

memberid	price
17-8184794	300
74-5359443	75

#### e. Room a program would be in

```
SELECT P.program_id, R.name FROM Program P, Room R WHERE P.room=R.room_id AND  
R.capacity >= 20;
```

✓ Showing rows 0 - 8 (9 total, Query took 0.0010 seconds.)

```
SELECT P.program_id, R.name FROM Program P, Room R WHERE P.room=R.room_id AND R.capacity >= 20
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain ]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

program_id	name
06-0092753	Livetube
07-4236778	Livetube
25-9447278	Kino
34-6858331	Realbuzz
54-9607283	Kino
62-8220268	Cogibox
64-9069029	Cogibox
67-7767546	Realbuzz
81-1314617	Kino

### III. Queries for three tables

#### a. Members that are students that are registered for a program

```
SELECT MS.membership_id FROM Member M, Program P, Room R, Membership MS WHERE P.room=R.room_id AND MS.membership_id LIKE '67%' GROUP BY MS.membership_id
```

✓ Showing rows 0 - 2 (3 total, Query took 0.0022 seconds.)

```
SELECT MS.membership_id FROM Member M, Program P, Room R, Membership MS WHERE P.room=R.room_id AND MS.membership_id LIKE '67%' GROUP BY MS.membership_id
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP Code ] [ Refresh ]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

membership_id
67-1248465
67-3951913
67-9786394

#### b. Teams trained by employee in specific room

```
SELECT E.name FROM `Sports Team` S, Employee E, Room R WHERE S.name = "volleyball" AND E.role LIKE '%trainer%' AND R.room_id = 'AT93'
```

✓ Showing rows 0 - 5 (6 total, Query took 0.0011 seconds.)

```
SELECT E.name FROM `Sports Team` S, Employee E, Room R WHERE S.name = "Volleyball" AND E.role LIKE '%trainer%' AND R.room_id = 'AT93'
```

☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP Co](#)

☐ Show all | Number of rows: 25 | Filter rows:

+ Options

name
Selestina Guynemer
Wylie Lumsden
Anastasie Fernihough
Swen Burkart
Henrieta Karim
Dahlia Vany

### c. Programs in specific room with equipment

```
SELECT P.program_id FROM Program P, Room R, Equipment Eq WHERE P.room=R.room_id AND Eq.type = 'treadmill' OR Eq.type='weights' GROUP BY P.program_id
```

+ Options

	program_id
<input type="checkbox"/> Edit Copy Delete	06-0092753
<input type="checkbox"/> Edit Copy Delete	07-4236778
<input type="checkbox"/> Edit Copy Delete	19-6141286
<input type="checkbox"/> Edit Copy Delete	24-2386212
<input type="checkbox"/> Edit Copy Delete	25-9447278
<input type="checkbox"/> Edit Copy Delete	26-7374937
<input type="checkbox"/> Edit Copy Delete	34-6858331
<input type="checkbox"/> Edit Copy Delete	52-1700420
<input type="checkbox"/> Edit Copy Delete	54-9607283
<input type="checkbox"/> Edit Copy Delete	62-8220268
<input type="checkbox"/> Edit Copy Delete	64-8768200
<input type="checkbox"/> Edit Copy Delete	64-9069029
<input type="checkbox"/> Edit Copy Delete	67-7767546
<input type="checkbox"/> Edit Copy Delete	81-1314617
<input type="checkbox"/> Edit Copy Delete	90-6950968

### d. Members in a specific program that joined after ... date that their membership duration is

```
SELECT M.memberid, M.join_date, P.program_id, MS.duration FROM Member M, Membership MS, Program P WHERE M.join_date = "10/26/2021" AND P.program_id LIKE '0%' AND MS.duration >= 50 GROUP BY M.memberid;
```

✓ Showing rows 0 - 0 (1 total, Query took 0.0012 seconds.)

```
SELECT M.memberid, M.join_date, P.program_id, MS.duration FROM Member M, Membership MS, Program P WHERE M.join_date = "10/26/2021" AND P.program_id LIKE '0%' AND MS.duration >= 50 GROUP BY M.memberid
```

☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP Code \]](#) [\[ Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows:

+ Options

memberid	join_date	program_id	duration
58-8419311	10/26/2021	06-0092753	90

### e. Member that uses specific program where they pay a specific price.

```
SELECT M.memberid FROM Member M, Membership MS, Program P WHERE M.memberid=MS.membership_id AND MS.price >= 100 AND P.day = 'Thursday' GROUP BY M.memberid
```

✓ Showing rows 0 - 0 (1 total, Query took 0.0012 seconds.)

```
SELECT M.memberid FROM Member M, Membership MS, Program P WHERE M.memberid=MS.membership_id AND MS.price >= 100 AND P.day = 'Thursday' GROUP BY M.memberid
```

☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP Code \]](#) [\[ Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows:

+ Options

← T →	memberid
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	67-1248465