Gym Management System

DATABASE IMPLEMENTATION REPORT

CONTENTS

1. PROJECT SUMMARY ................................................................................................................ 3
2. ENTITY AND ATTRIBUTE TABLE........................................................................................... 4
3. ENTITY-RELATIONSHIP DIAGRAM........................................................................................ 9
4. CREATION OF TABLES ............................................................................................................17
5. POPULATING THE TABLES…………………………………………………………………………18
6. MAJOR DATA QUESTIONS...................................................................................................... 27
7. INTERFACES ............................................................................................................................. 38
8. REPORT ...................................................................................................................................... 46

PROJECT SUMMARY

This project focuses on designing a database for Gym Management System. There are a number of gyms all over the world and this system could be implemented by a majority of the gyms. There is so much data involved in this such as: Which Trainer is training which Member in the gym? What supplements are available in the gym? What plan has a particular member opted for? Is this Trainer available for personal training? How many customer’s membership is going to expire? (Particular Month). How many customers joined the Gym in a particular month? How many customers live in certain areas? (Marketing Purpose). Which customers do we have to inform that their membership expire next month?

Suppose, a member wants to opt for a personal trainer. So, the receptionist calls the trainer, asks whether he/she is available at member’s preferred time and then co-ordinates accordingly. This is not a professional practice. There is a notice board in the gym which is used for delivering any important news to the members (plan/offer) and other updates. 1 out of the 10 members looks at this board which does not drive business.

The current member enrollment process is paper-based i.e. all of the member’s records are maintained in a paper file, which are used for later reference. This occupies a large amount of space for information in paper files and hence it is quite tedious, complicated and time-consuming in order to fetch details of a particular member. Also, physical data can be misplaced or damaged easily. Sometimes, manual calculations can be incorrect thus, leading to inaccurate information in payment of a member’s plan or product.

During enrollment, a paper-based form will be filled by member which will be then handed over to the receptionist. The receptionist will then feed the system with the member details. Member details include First Name, Last Name, Age (to make sure he is 18 or above), Sex, PhNo (So that important reminders can be sent via PhNo.), Street No, Street Name (For Marketing Purpose). Along with member details, the system will also record the employee details. Also, what plan has a particular member enrolled for, what offer did he redeem while enrolling for membership. There are different types of products and supplements available in the gym. Members can order these products. Member’s payment information is also stored in this system. Hence, there is a vast amount of data that is being collected.

The purpose of Gym Management System is to increase automation and reduce manual effort. This report comprises of the system design for the proposed system and stating their functions, dependencies and relationships between the various entities and attributes. The analysis of the project will be shown in this report. The business rules of this database management system are also stated. The report will show a fully attributed entity relationship diagram (ERD). The report also addresses the major queries and information that may be required by various users of this system.

ENTITY AND ATTRIBUTE TABLE

**Member** – This entity captures information about the member

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name – Member** | **ATTRIBUTE NAME** | **FIELD TYPE** | **NULL/NOT NULL** | **EXPLANATION** |
| Primary Key | mID | INT | NOT NULL | Primary Key or Unique Identifier of the Member |
|  | mFName | VARCHAR(20) | NOT NULL | First Name of the Member |
|  | mLName | VARCHAR(20) | NOT NULL | Last Name of the Member |
|  | mSex | CHAR(1) | NOT NULL | Sex of the Member |
|  | mAge | VARCHAR(3) | NOT NULL | Age of the Member |
|  | mPhNo | NUM | NOT NULL | Phone No. of the Member |
|  | mStreetNo | VARCHAR(20) | NOT NULL | Street No. of where the member lives |
|  | mStreetName | VARCHAR(250) | NOT NULL | Street Name of where the member lives |
| Foreign Key | trainerID | INT | NULL | Foreign key identifying the trainer associated with the Member |

**Offer** – This entity captures various enrollment / membership offers available during different time of the year.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name – Offer** | **ATTRIBUTE NAME** | **FIELD TYPE** | **NULL/NOT NULL** | **EXPLANATION** |
| Primary Key | oID | INT | NOT NULL | Primary Key or Unique Identifier of the Offer |
|  | oName | VARCHAR(20) | NOT NULL | Description of the offer (For Example, Spring Offer, Fall Offer, Other Offers ) |
|  | oStartDate | DATE | NOT NULL | Start Date of Offer |
|  | oEndDate | DATE | NOT NULL | End Date of Offer |
|  | oIntake | VARCHAR(20) | NOT NULL | Number of people who can apply for this offer |

**Plan**- This entity stores different plans members can enroll for.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name – Plan** | **ATTRIBUTE NAME** | **FIELD TYPE** | **NULL/NOT NULL** | **EXPLANATION** |
| Primary Key | pID | INT | NOT NULL | Primary Key/Unique identifier of the plan |
|  | pName | VARCHAR(20) | NOT NULL | Description of the plan. (For Example, Zumba,Crossfit, Aerobics etc.) |
|  | PIntake | VARCHAR(3) | NOT NULL | Number of people who can apply for this plan. |
|  | PAmount | VARCHAR(10) | NOT NULL | Price for the plan |
|  | PDuration | VARCHAR(10) | NOT NULL | Duration in months |

**Employee** - This entity stores details of different Employees at the gym.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name – Employee** | **ATTRIBUTE NAME** | **FIELD TYPE** | **NULL/NOT NULL** | **EXPLANATION** |
| Primary Key | eID | INT | NOT NULL | Primary Key or Unique Identifier of the Employee |
|  | eFName | VARCHAR(20) | NOT NULL | First name of the Employee |
|  | eLName | VARCHAR(20) | NOT NULL | Last name of the Employee |
|  | eAge | VARCHAR(3) | NOT NULL | Age of the Employee |
|  | eSex | VARCHAR(1) | NOT NULL | Sex of the Employee |
|  | eType | VARCHAR(20) | NOT NULL | Type of Employee (For example receptionists, trainer, cleaner ) |

**Trainer -** This entity stores details of different trainers at the gym.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name – Trainer** | **ATTRIBUTE NAME** | **FIELD TYPE** | **NULL/NOT NULL** | **EXPLANATION** |
| Primary Key | trainerID | INT | NOT NULL | Primary Key or Unique Identifier of the trainer |
| Foreign Key | eID | INT | NOT NULL | Foreign Key referring to employee details |
|  | tType | VARCHAR(20) | NOT NULL | Either Personal Trainer or Floor Trainer |
|  | TExperience | VARCHAR(2) | NOT NULL | Experience in years |

**Receptionist** - This entity stores details of different receptionists at the gym

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name – Receptionist** | **ATTRIBUTE NAME** | **FIELD TYPE** | **NULL/NOT NULL** | **EXPLANATION** |
| Primary Key | recpID | INT | NOT NULL | Primary Key or Unique Identifier of the receptionist |
| Foreign Key | eID | INT | NOT NULL | Foreign Key referring to employee details |

**Product** - This entity stores different products a member can buy from the gym.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name – Product** | **ATTRIBUTE NAME** | **FIELD TYPE** | **NULL/NOT NULL** | **EXPLANATION** |
| Primary Key | UPC | INT | NOT NULL | Unique identifier/Primary Key of the Product |
|  | productName | VARCHAR(20) | NOT NULL | Name of the Product (For example, OptimumNutrition, UltimateNut ) |
|  | productType | VARCHAR(20) | NOT NULL | Example, Fat Burner, Protein, BCAA etc. |
|  | productPrice | VARCHAR(20) | NOT NULL | Price of the product |

**Memberships** – This entity acts as an Associative entity between Member and Plan. It stores different plans a particular member is opting for, what offer has been redeemed and which receptionist is enrolling the member.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name – Memberships** | **ATTRIBUTE NAME** | **FIELD TYPE** | **NULL/NOT NULL** | **EXPLANATION** |
| Primary Key & Foreign Key | mID | INT | NOT NULL | Primary Key & Foreign Key referring to Member details |
| Primary Key & Foreign Key | pID | INT | NOT NULL | Primary Key & Foreign Key referring to Plan details |
| Primary Key | enrollDate | DATE | NOT NULL | Date of enrollment |
| Foreign Key | oID | INT | NOT NULL | Foreign key referring to the offer redeemed |
| Foreign Key | recpID | INT | NOT NULL | Foreign key referring to the Receptionist |
|  | ExpDate | DATE | NOT NULL | End Date of the plan |

**Order** – This entity stores the details of the order placed by the member

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name – Order** | **ATTRIBUTE NAME** | **FIELD TYPE** | **NULL/NOT NULL** | **EXPLANATION** |
| Primary Key | orderID | INT | NOT NULL | Primary Key or Unique Identifier of the order |
| Primary Key & Foreign Key | mID | INT | NOT NULL | Primary Key & Foreign Key referring to the member details. |
|  | orderDate | DATE | NOT NULL | Date the order was placed |

**Orderline** – This entity acts as an Associative entity between order and product. It keeps track of products a particular member has ordered and the quantity of the order.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name – Orderline** | **ATTRIBUTE NAME** | **FIELD TYPE** | **NULL/NOT NULL** | **EXPLANATION** |
| Primary Key & Foreign Key | orderID | INT | NOT NULL | Primary Key & Foreign key referring to the order details. |
| Primary Key & Foreign Key | mID | INT | NOT NULL | Primary Key & Foreign Key referring to the Member details |
| Primary Key & Foreign Key | UPC | INT | NOT NULL | Primary Key & Foreign Key referring to the Product Details |
|  | quantity | VARCHAR(10) | NOT NULL | Quantity of the product ordered |

**Payment** – This entity stores the Payment details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name – Payment** | **ATTRIBUTE NAME** | **FIELD TYPE** | **NULL/NOT NULL** | **EXPLANATION** |
| Primary Key | payID | INT | NOT NULL | Primary Key/Unique identifier of the payment |
|  | cardNo | VARCHAR(20) | NOT NULL | Card number |
|  | payAmount | VARCHAR(20) | NOT NULL | Payment Amount |
|  | payDate | DATE | NOT NULL | Payment Date |

**OrderPayment** – This entity stores the member’s order payment details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name – Payment** | **ATTRIBUTE NAME** | **FIELD TYPE** | **NULL/NOT NULL** | **EXPLANATION** |
| Primary Key & Foreign Key | payID | INT | NOT NULL | Primary Key of the payment & Foreign key referring to the payment details |
| Foreign Key | orderID | INT | NOT NULL | Foreign key referring to the order details. |
| Foreign Key | mID | INT | NOT NULL | Foreign key referring to the member details. |
| Foreign Key | UPC | INT | NOT NULL | Foreign key referring to the product details. |
|  | oAmount | VARCHAR(20) | NOT NULL | Order Amount |

**PlanPayment** – This entity stores the member’s plans payment details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name – Payment** | **ATTRIBUTE NAME** | **FIELD TYPE** | **NULL/NOT NULL** | **EXPLANATION** |
| Primary Key & Foreign Key | payID | INT | NOT NULL | Primary Key of the payment & Foreign key referring to the payment details |
| Foreign Key | pID | INT | NOT NULL | Foreign key referring to the order details. |
| Foreign Key | mID | INT | NOT NULL | Foreign key referring to the member details. |
| Foreign Key | enrollDate | DATE | NOT NULL | Foreign key |

ENTITY RELATIONSHIP DIAGRAM



CREATION OF TABLES

/\*creating table\*/

/\*creating OFFER table\*/

CREATE TABLE Offer (

oID INT NOT NULL,

oName VARCHAR(20) NOT NULL,

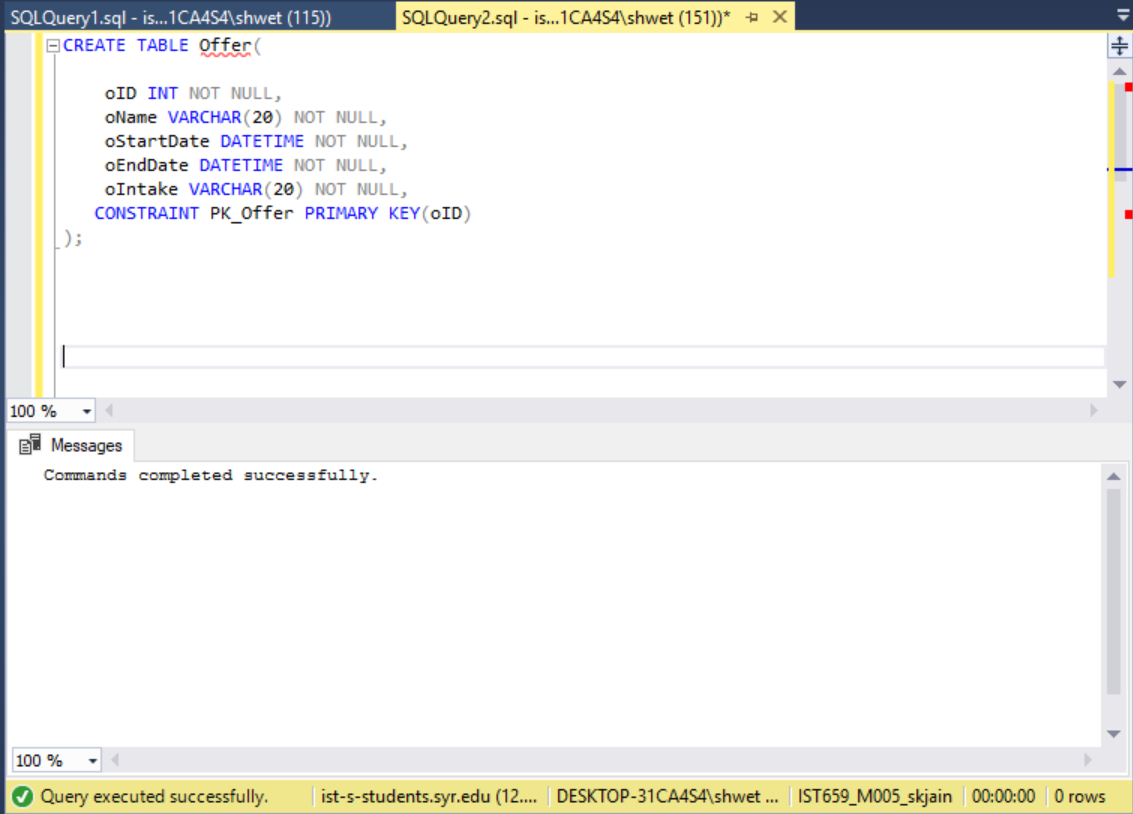
oStartDate DATETIME NOT NULL,

oEndDate DATETIME NOT NULL,

oIntake VARCHAR(20) NOT NULL,

CONSTRAINT PK\_Offer PRIMARY KEY(oID)

);



/\*creating Plan table\*/

CREATE TABLE Plan1 (

pID INT NOT NULL,

pName VARCHAR(20) NOT NULL,

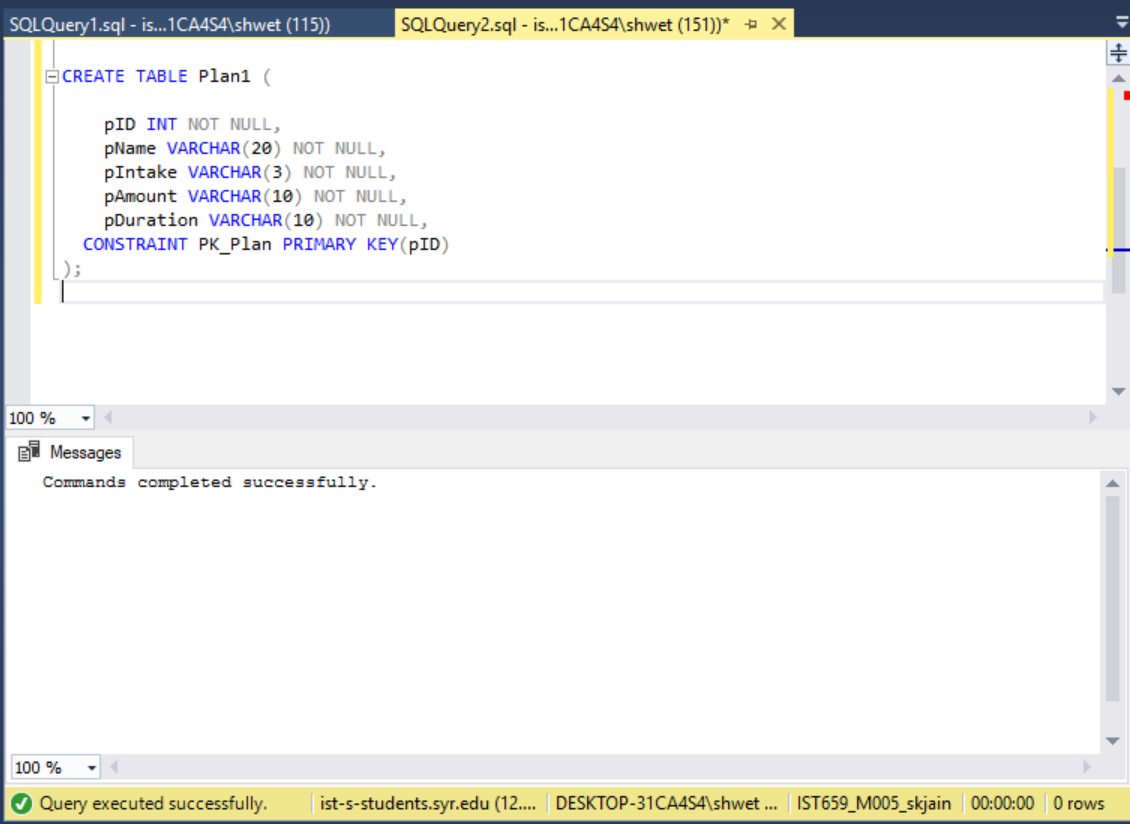
pIntake VARCHAR(3) NOT NULL,

pAmount VARCHAR(10) NOT NULL,

pDuration VARCHAR(10) NOT NULL,

CONSTRAINT PK\_Plan PRIMARY KEY(pID)

);



/\*creating Product table\*/

CREATE TABLE Product1 (

UPC INT NOT NULL,

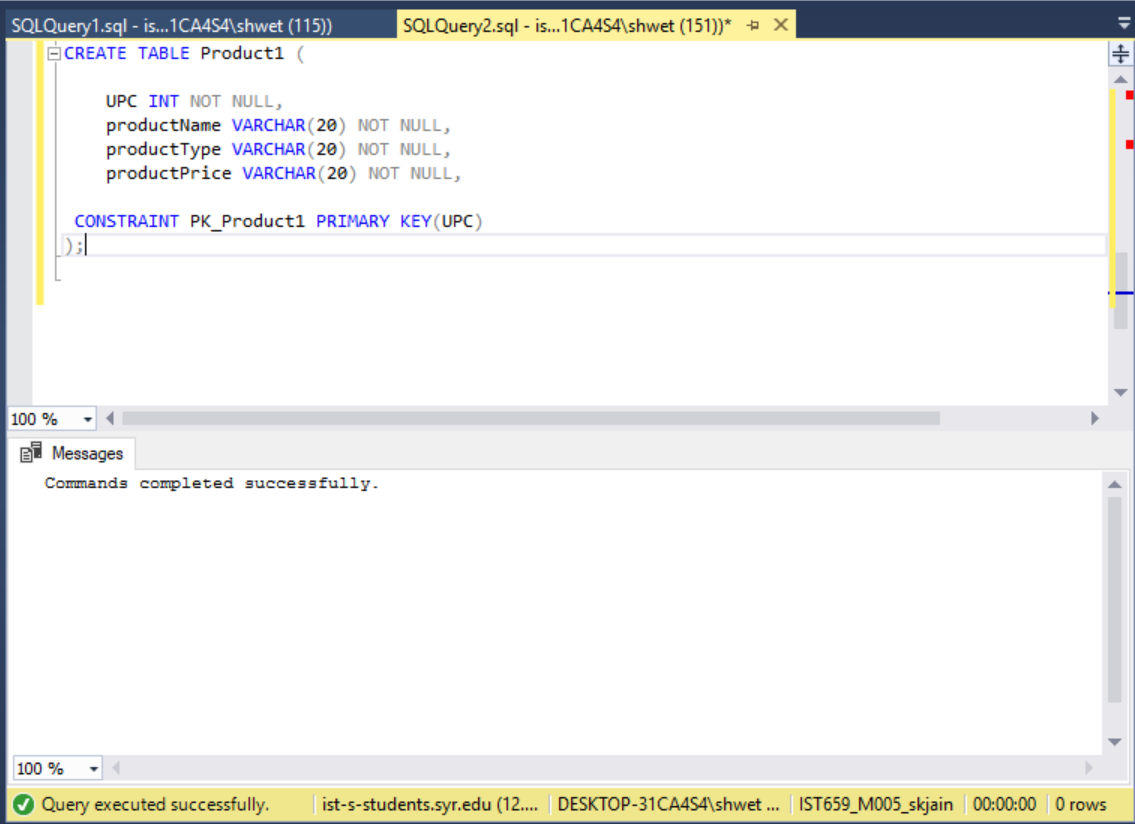
productName VARCHAR(20) NOT NULL,

productType VARCHAR(20) NOT NULL,

productPrice VARCHAR(20) NOT NULL,

CONSTRAINT PK\_Product1 PRIMARY KEY(UPC)

);



/\*creating Employee table\*/

CREATE TABLE Employee (

eID INT NOT NULL,

eFName VARCHAR(20) NOT NULL,

eLName VARCHAR(3) NOT NULL,

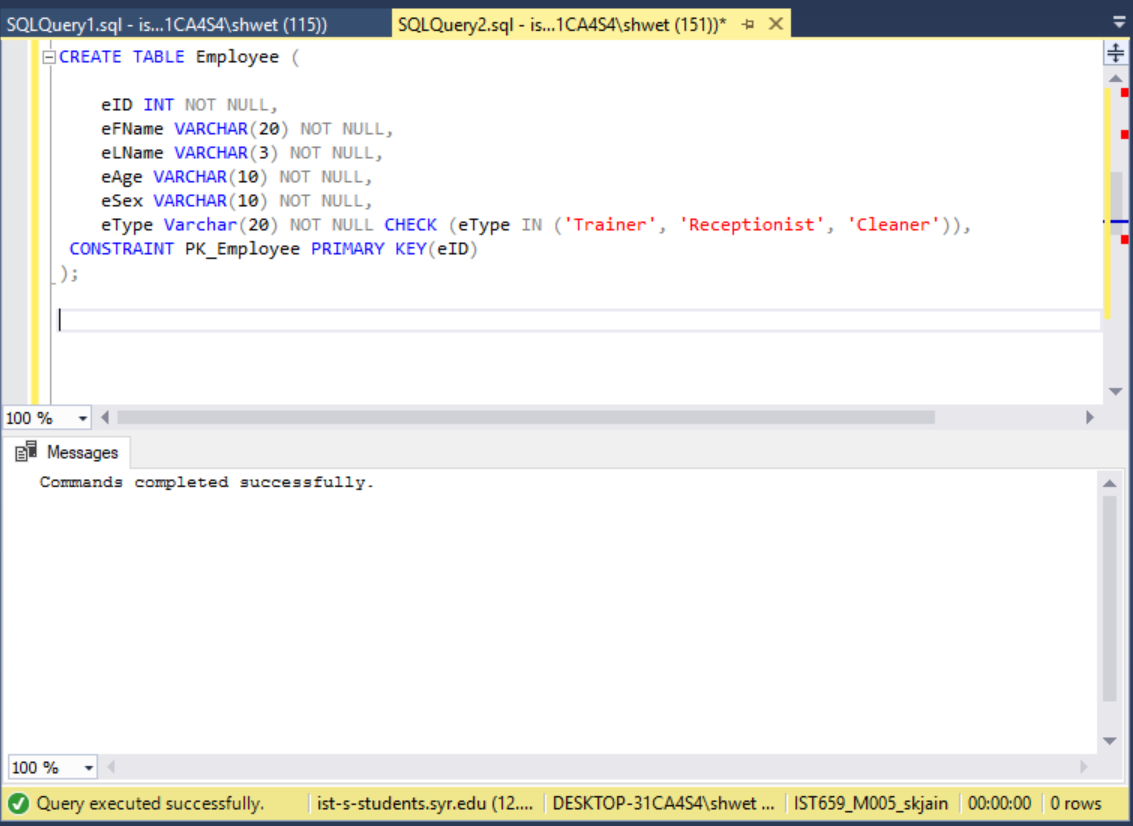
eAge VARCHAR(10) NOT NULL,

eSex VARCHAR(10) NOT NULL,

eType Varchar(20) NOT NULL CHECK (eType IN ('Trainer', 'Receptionist', 'Cleaner')),

CONSTRAINT PK\_Employee PRIMARY KEY(eID)

);



/\*creating Payment table\*/

CREATE TABLE Payment (

payID INT NOT NULL,

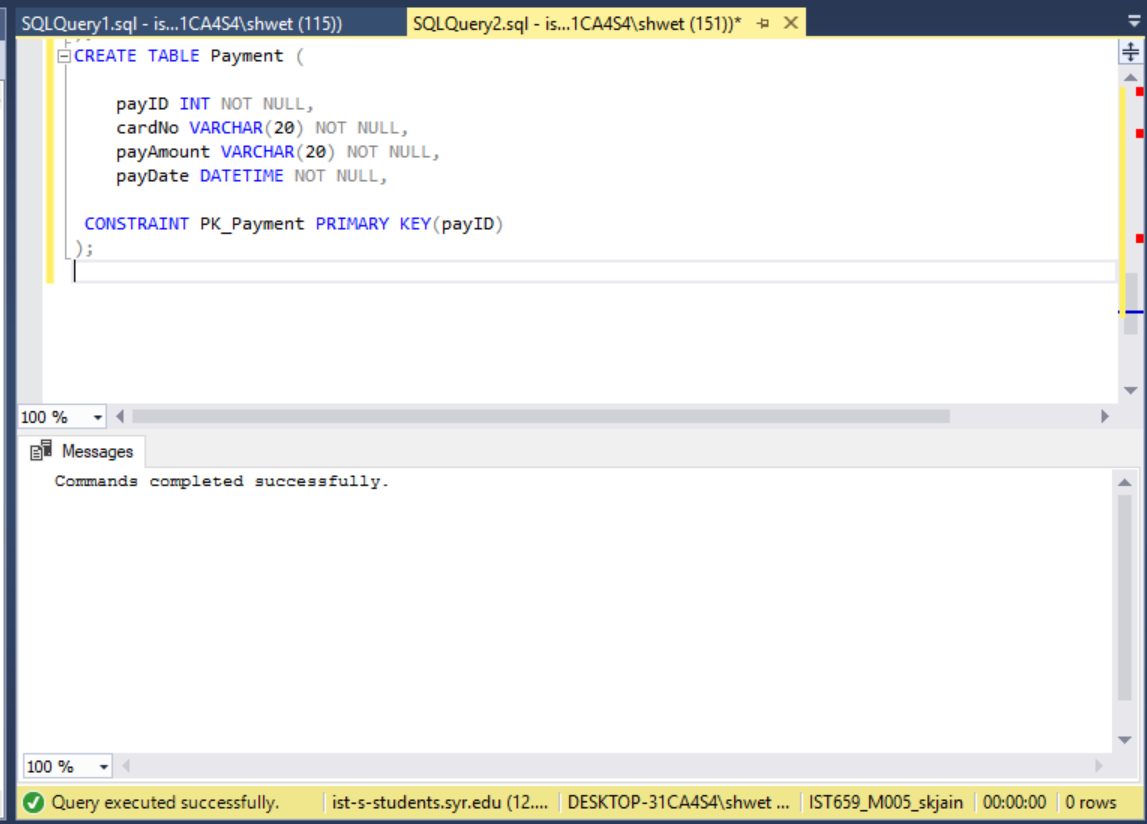
cardNo VARCHAR(20) NOT NULL,

payAmount VARCHAR(20) NOT NULL,

payDate DATETIME NOT NULL,

CONSTRAINT PK\_Payment PRIMARY KEY(payID)

);



/\*creating Trainer table\*/

CREATE TABLE Trainer (

trainerID INT NOT NULL,

eID INT NOT NULL,

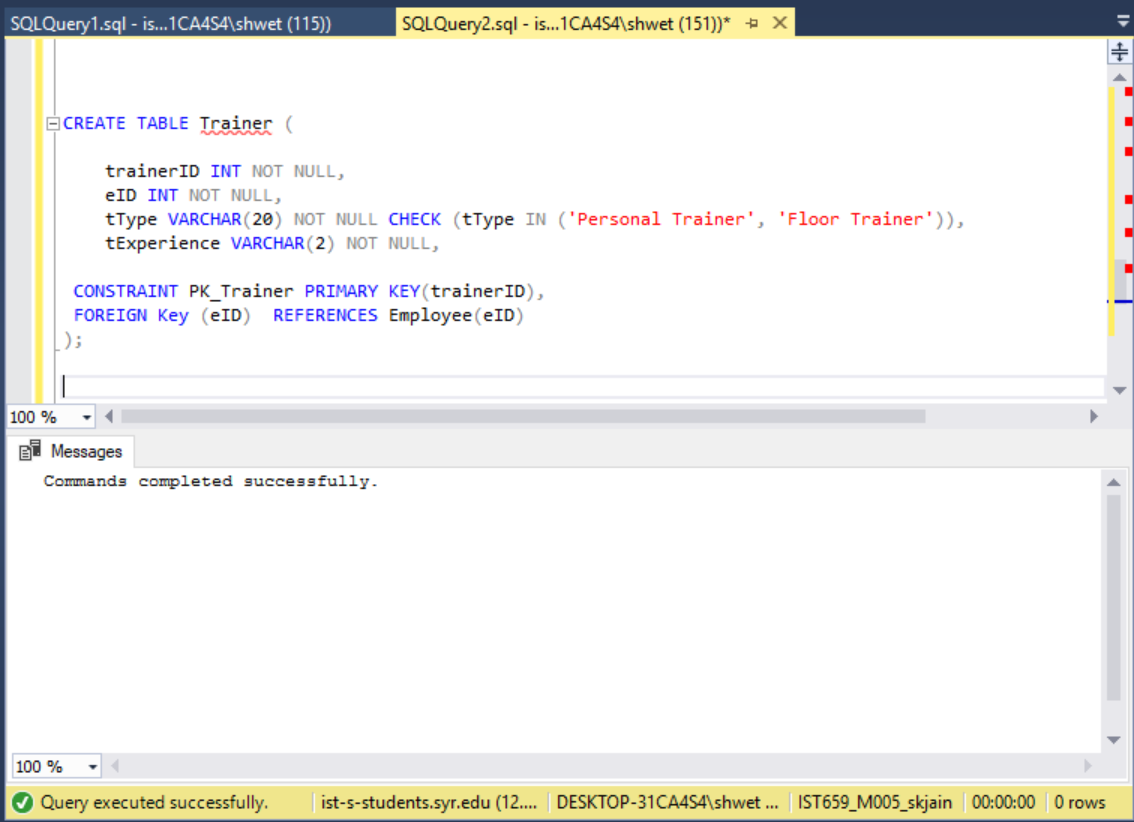
tType VARCHAR(20) NOT NULL CHECK (tType IN ('Personal Trainer', 'Floor Trainer')),

tExperience VARCHAR(2) NOT NULL,

CONSTRAINT PK\_Trainer PRIMARY KEY(trainerID),

FOREIGN Key (eID) REFERENCES Employee(eID)

);



/\*creating Receptionist table\*/

CREATE TABLE Receptionist (

recepID INT NOT NULL,

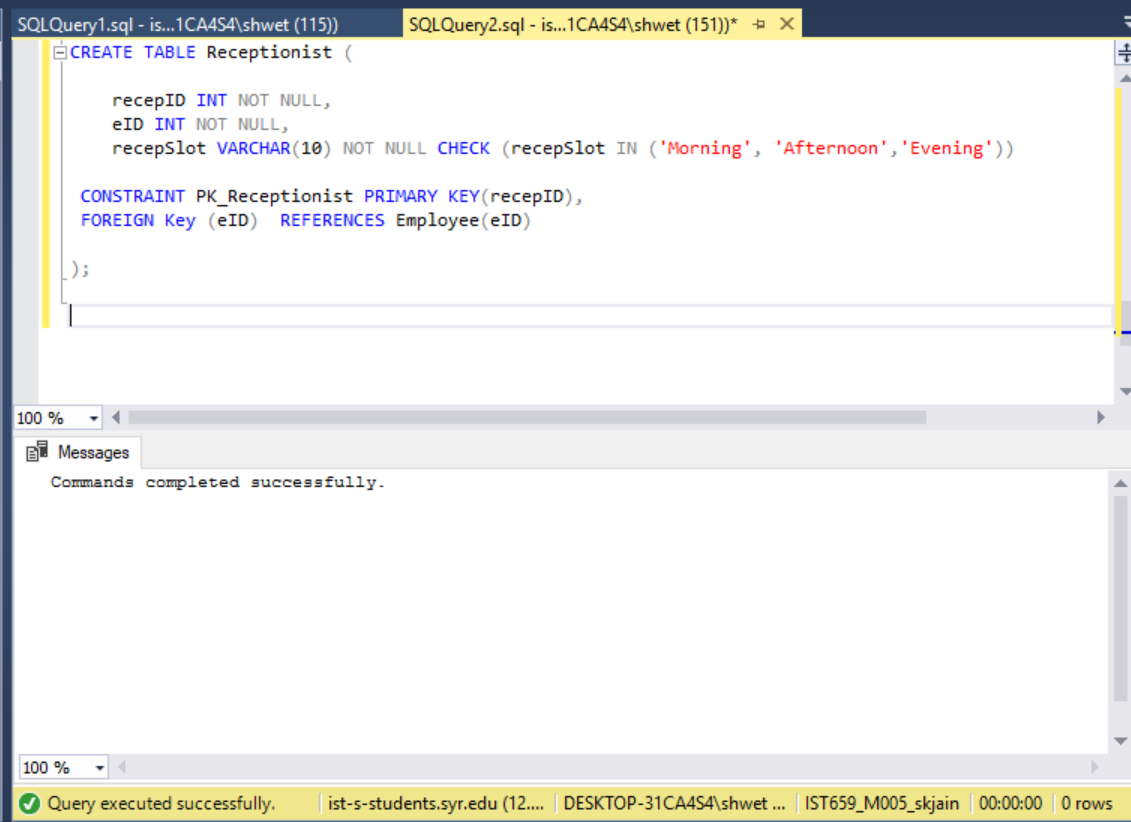
eID INT NOT NULL,

recepSlot VARCHAR(10) NOT NULL CHECK (recepSlot IN ('Morning', 'Afternoon','Evening'))

CONSTRAINT PK\_Receptionist PRIMARY KEY(recepID),

FOREIGN Key (eID) REFERENCES Employee(eID)

);



/\*creating Member table\*/

CREATE TABLE Member (

mID INT NOT NULL,

mFName VARCHAR(20) NOT NULL,

mLName VARCHAR(20) NOT NULL,

mSex VARCHAR(1) NOT NULL,

mAge VARCHAR(3) NOT NULL,

mPhNo VARCHAR(20) NOT NULL,

mSreetNo varchar(20) NOT NULL,

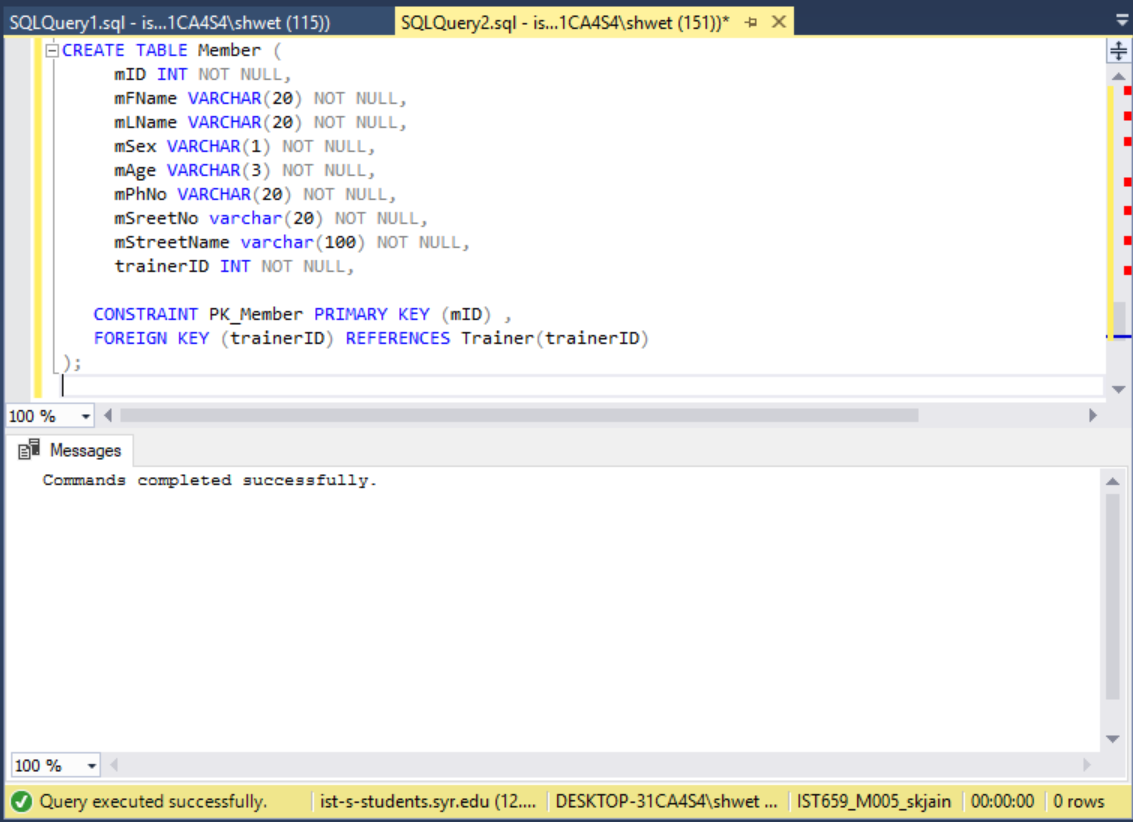
mStreetName varchar(100) NOT NULL,

trainerID INT NOT NULL,

CONSTRAINT PK\_Member PRIMARY KEY (mID) ,

FOREIGN KEY (trainerID) REFERENCES Trainer(trainerID)

);



/\*creating Membership table\*/

CREATE TABLE Membership (

mID INT NOT NULL,

pID INT NOT NULL,

enrollDate DATETIME NOT NULL,

oID INT NOT NULL,

recepID INT NOT NULL,

expDate DATETIME NOT NULL,

FOREIGN KEY (mID) REFERENCES Member(mID) ,

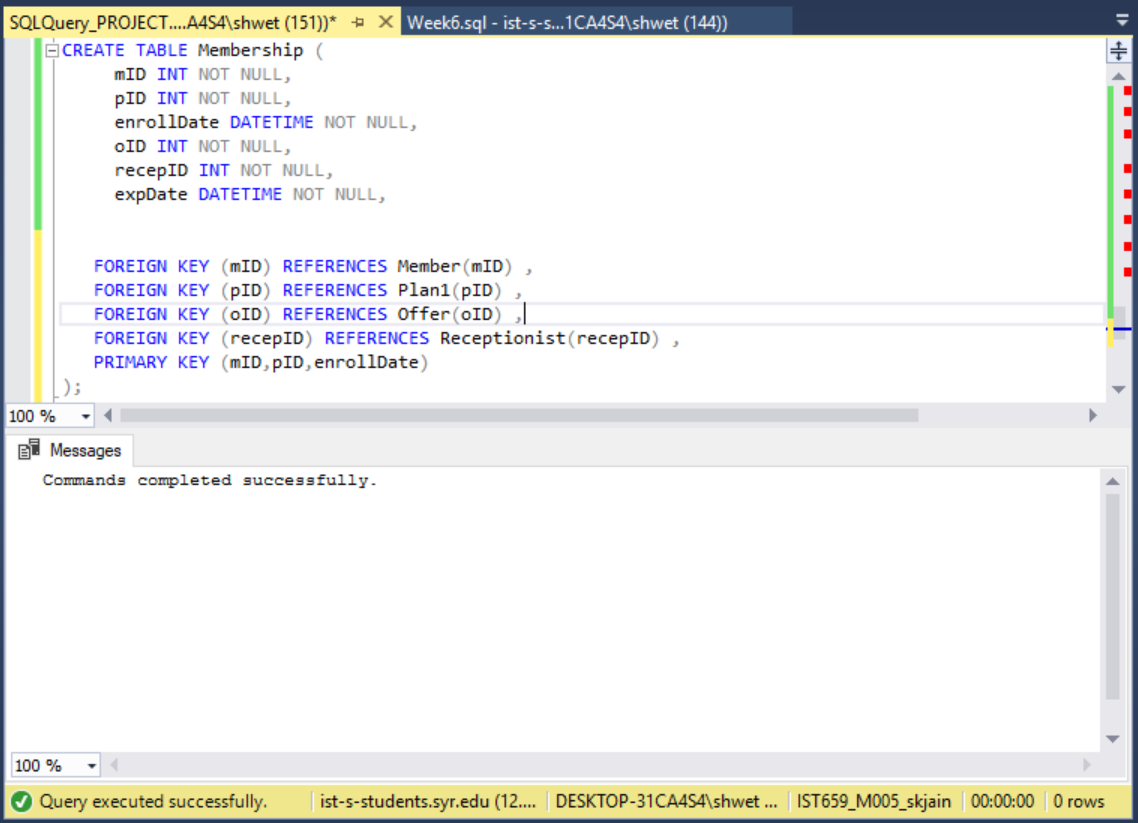
FOREIGN KEY (pID) REFERENCES Plan1(pID) ,

FOREIGN KEY (oID) REFERENCES Offer(oID) ,

FOREIGN KEY (recepID) REFERENCES Receptionist(recepID) ,

PRIMARY KEY (mID,pID,enrollDate)

);



/\*creating Order table\*/

CREATE TABLE order1 (

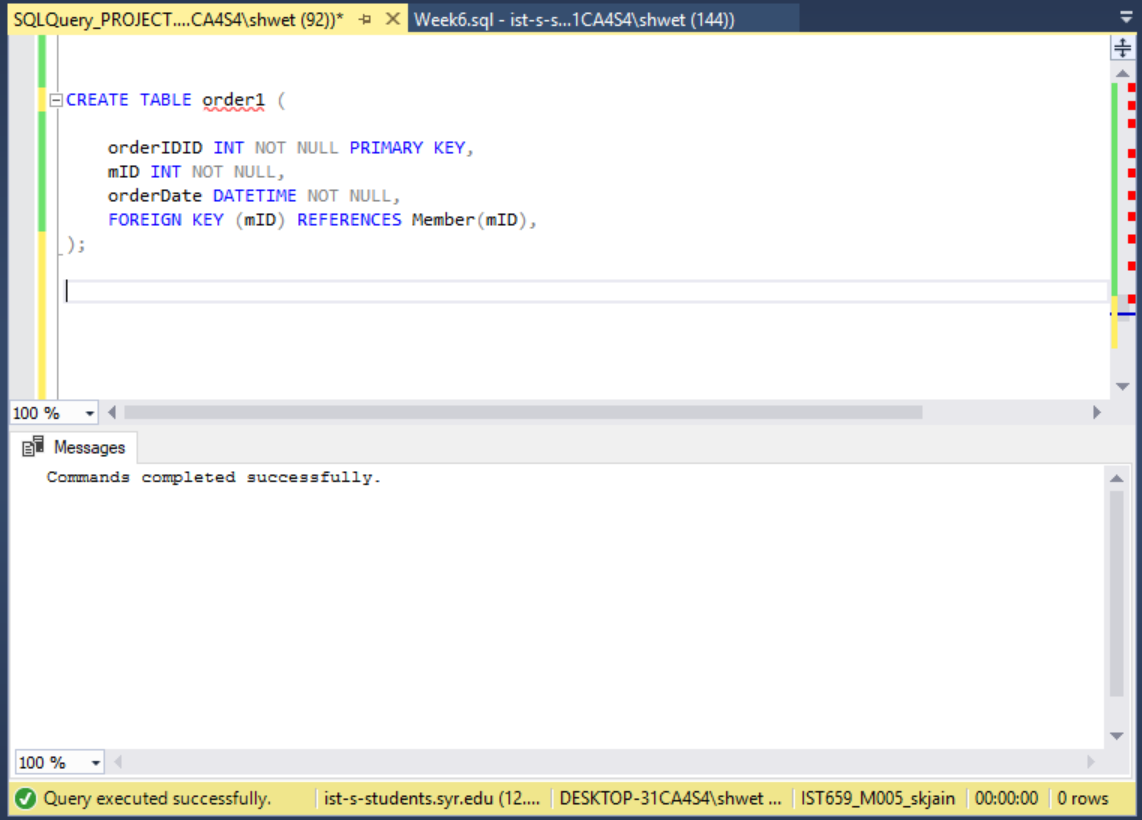
orderIDID INT NOT NULL PRIMARY KEY,

mID INT NOT NULL,

orderDate DATETIME NOT NULL,

FOREIGN KEY (mID) REFERENCES Member(mID),

);



/\*creating orderLine table\*/

CREATE TABLE orderLine1 (

orderIDID INT NOT NULL,

UPC INT NOT NULL,

mID INT Not NULL,

quantity VARCHAR(10) NOT NULL,

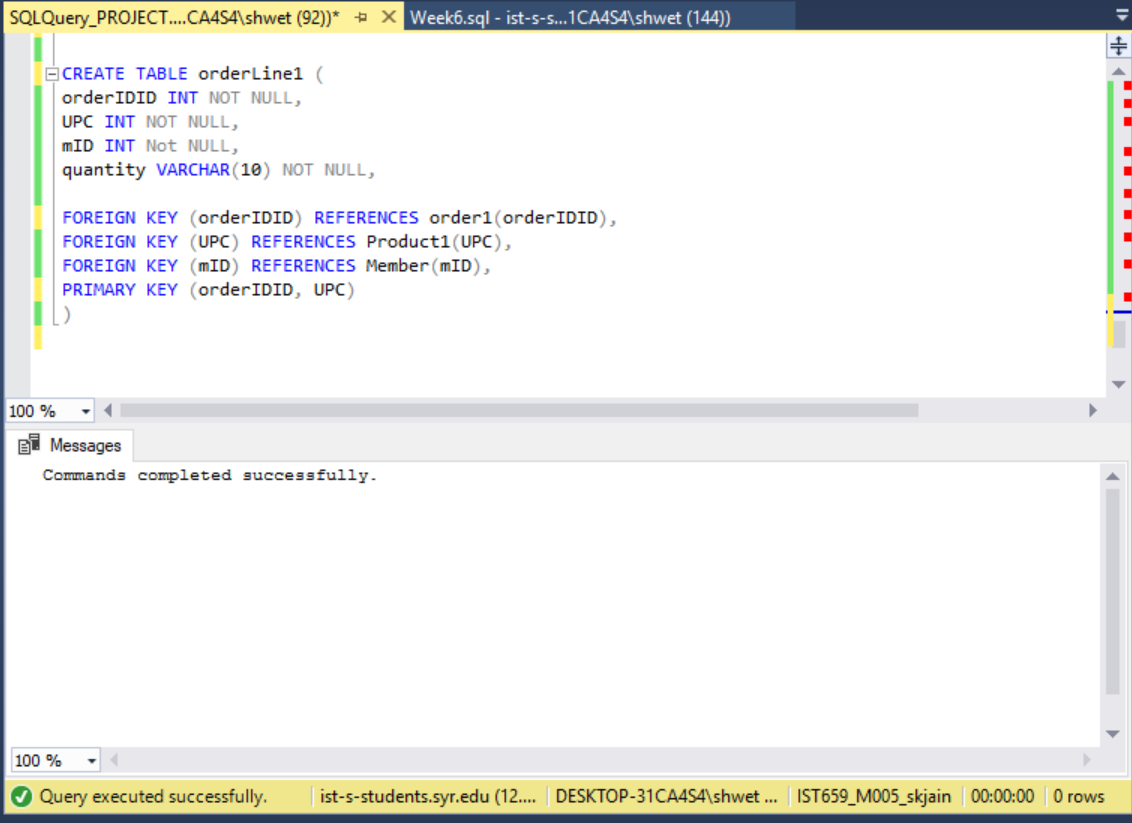
FOREIGN KEY (orderIDID) REFERENCES order1(orderIDID),

FOREIGN KEY (UPC) REFERENCES Product1(UPC),

FOREIGN KEY (mID) REFERENCES Member(mID),

PRIMARY KEY (orderIDID, UPC)

)



/\*creating orderPayment table\*/

CREATE TABLE orderPayment (

payID INT NOT NULL FOREIGN KEY (payID) REFERENCES Payment(payID),

orderIDID INT NOT NULL,

UPC INT NOT NULL,

mID INT Not NULL,

oAmount VARCHAR(20) NOT NULL,

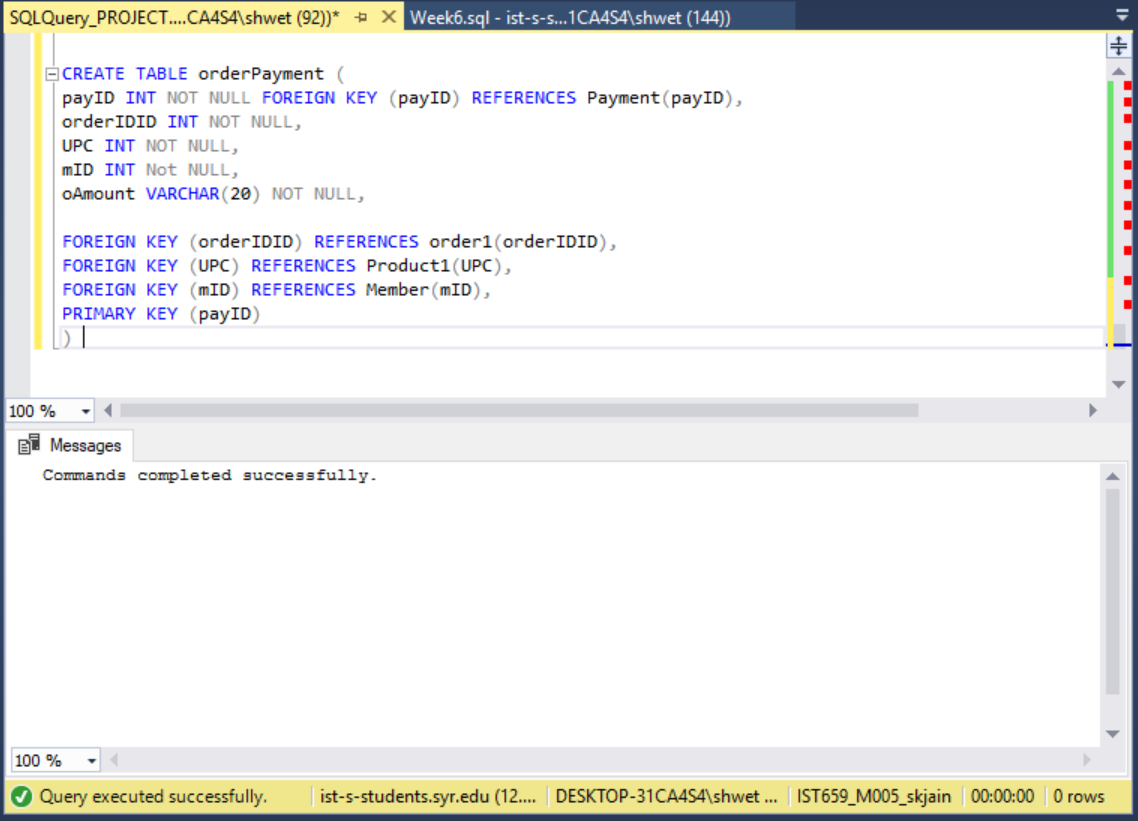
FOREIGN KEY (orderIDID) REFERENCES order1(orderIDID),

FOREIGN KEY (UPC) REFERENCES Product1(UPC),

FOREIGN KEY (mID) REFERENCES Member(mID),

PRIMARY KEY (payID)

)



/\*creating planPayment table\*/

CREATE TABLE planPayment (

payID INT NOT NULL FOREIGN KEY (payID) REFERENCES Payment(payID),

orderIDID INT NOT NULL,

pID INT NOT NULL,

mID INT Not NULL,

enrollDate DATETIME NOT NULL,

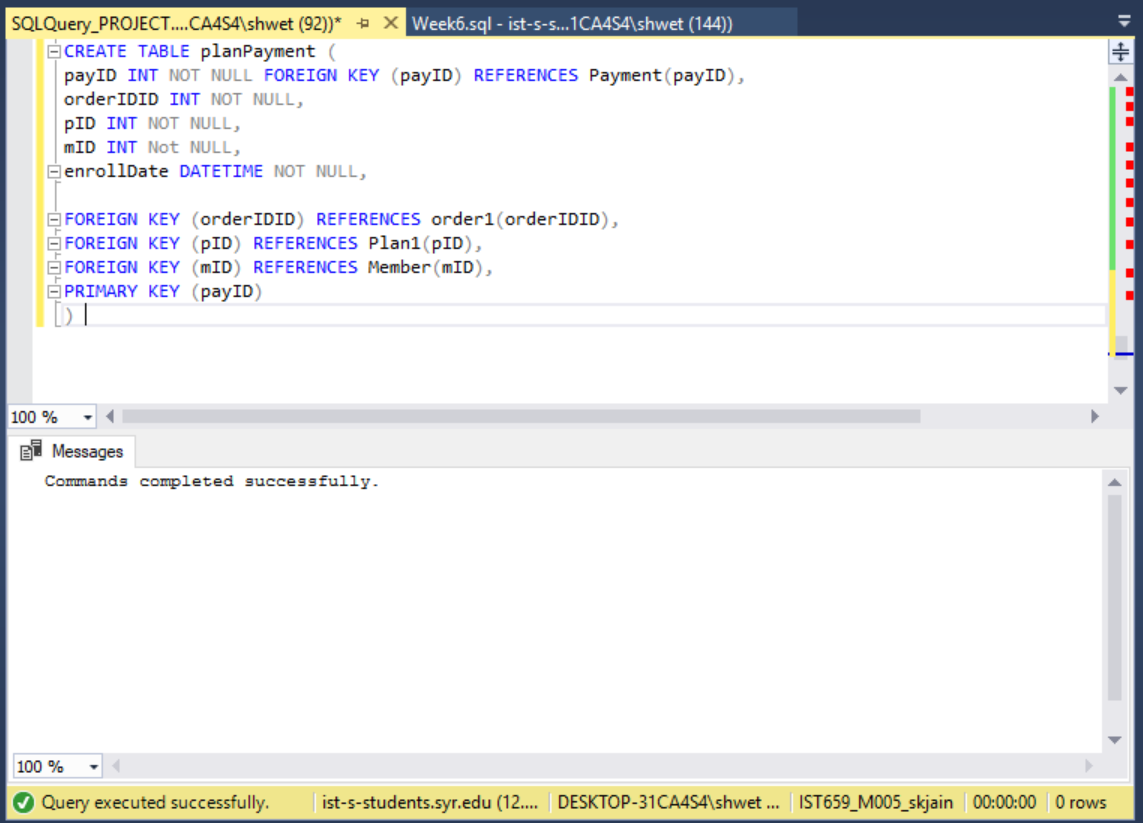
FOREIGN KEY (orderIDID) REFERENCES order1(orderIDID),

FOREIGN KEY (pID) REFERENCES Plan1(pID),

FOREIGN KEY (mID) REFERENCES Member(mID),

PRIMARY KEY (payID)

)



POPULATING THE TABLES

--Populating the Offer table

INSERT INTO Offer

VALUES

(1,'Spring','2018-03-05','2018-05-05',20),

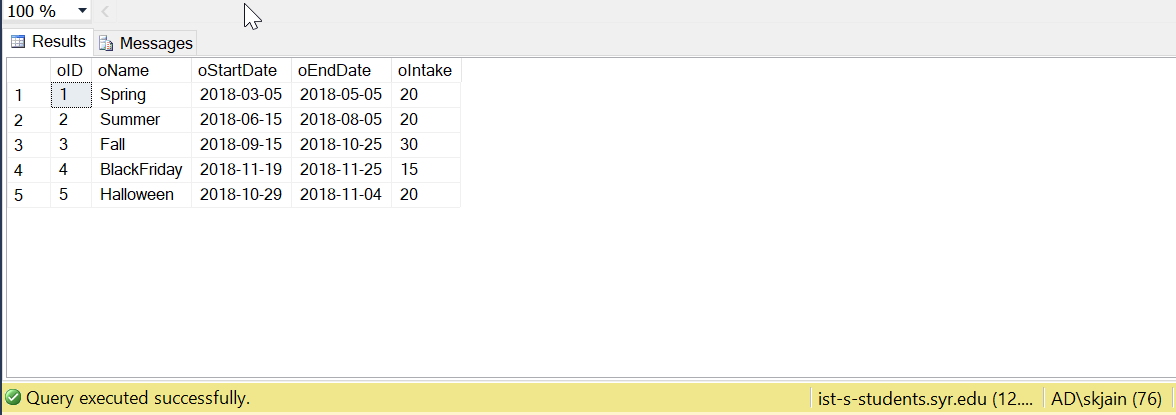
(2,'Summer','2018-06-15','2018-08-05',20),

(3,'Fall','2018-09-15','2018-10-25',30),

(4,'BlackFriday','2018-11-19','2018-11-25',15),

(5,'Halloween','2018-10-29','2018-11-04',20)

select\* from Offer



--Populating the Plan Table

INSERT INTO Plan1

VALUES

(1, 'Regular',200,350,12),

(2, 'Zumba',60,100,3),

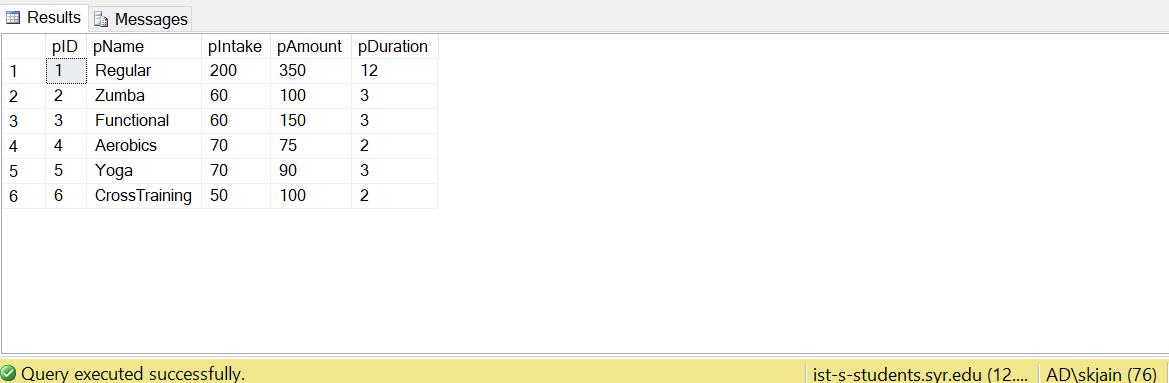
(3, 'Functional',60,150,3),

(4, 'Aerobics',70,75,2),

(5,'Yoga',70,90,3),

(6,'CrossTraining',50,100,2)

select \* from plan1



--Populating the Employee Table

INSERT INTO Employee

VALUES

(1, 'John','F',25,'M','Trainer'),

(2, 'Kim','D',24,'M','Trainer'),

(3, 'Sam','J',27,'M','Receptionist'),

(4, 'Emily','S',28,'F','Receptionist'),

(5,'Hannah','B',24,'F','Trainer'),

(6,'Tim','J',28,'M','Trainer'),

(7, 'Rahul','R',22,'M','Trainer'),

(8, 'Ritika','S',23,'F','Trainer'),

(9, 'Collins','D',35,'M','Receptionist'),

(10, 'Kanisha','P',22,'F','Cleaner'),

(11,'Kirk','D',27,'M','Cleaner'),

(12,'Jolton','A',27,'M','Cleaner'),

(13,'Merick','G',34,'M','Trainer'),

(14,'Austin','Q',34,'M','Trainer'),

(15,'Anushka','P',29,'F','Trainer'),

(16,'Malika','J',27,'F','Trainer'),

(17,'Jina','N',25,'F','Trainer'),

(18,'Tina','Q',34,'F','Cleaner'),

(19,'Juli','M',36,'F','Cleaner'),

(20,'Kristen','S',45,'F','Cleaner'),

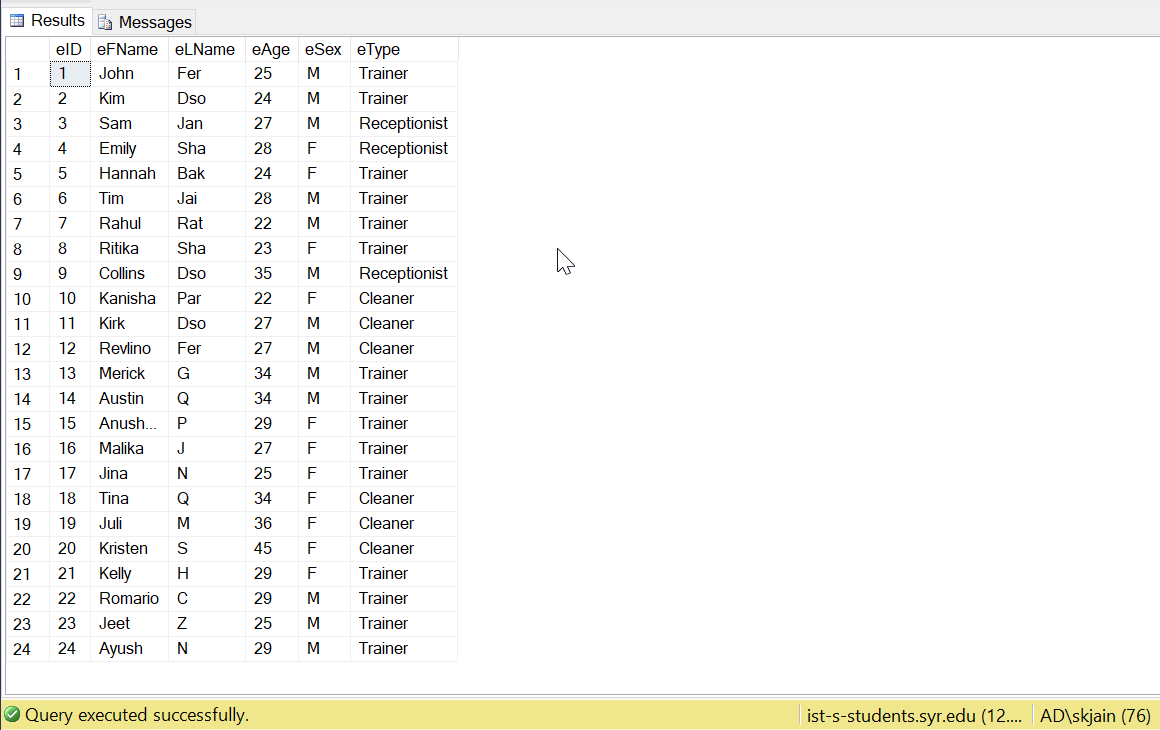
(21,'Kelly','H',29,'F','Trainer'),

(22,'Romario','C',29,'M','Trainer'),

(23,'Jeet','Z',25,'M','Trainer'),

(24,'Ayush','N',29,'M','Trainer')

select \* from Employee



--Populating the Product Table

INSERT INTO Product1

VALUES

(14534, 'Ultimate Nutrition','WheyProtein',50),

(23434, 'Optimum Nutrtion','MassGainer',45),

(32432, 'NitroTech','FatBurner',28),

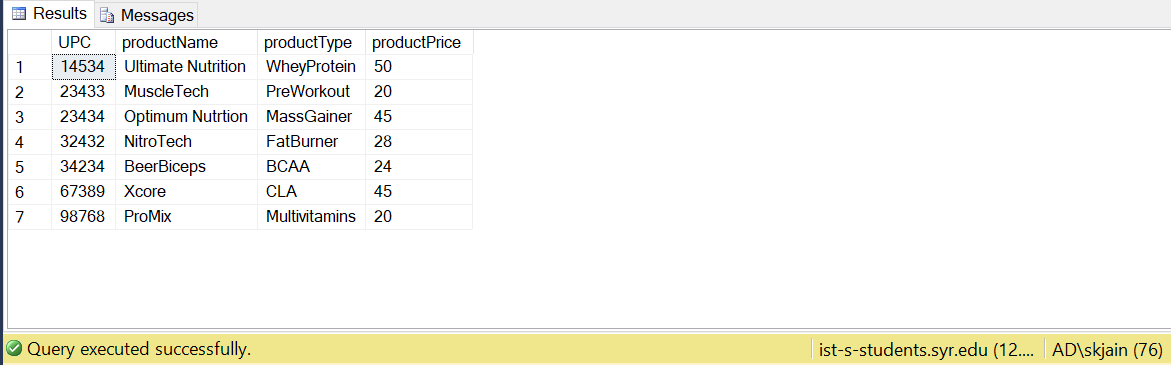
(23433, 'MuscleTech','PreWorkout',20),

(34234,'BeerBiceps','BCAA',24),

(98768,'ProMix','Multivitamins',20),

(67389,'Xcore','CLA',45)

Select \* from Product1



--Populating the Payment Table

INSERT INTO Payment

VALUES

(1,3785698768762098,50,'2018-08-16'),

(2,9867098739750932,73,'2018-06-15'),

(3,7467594774577830,50,'2018-05-15'),

(4,4797487974398709,45,'2018-04-17'),

(5,6903087588874798,50,'2018-05-05'),

(6,8098477809239487,65,'2018-09-13'),

(7,7736878768768767,50,'2018-10-03'),

(8,7347688746287672,45,'2018-10-19'),

(9,1939737987987987,24,'2018-10-17'),

(10,8737873987948798,69,'2018-11-24'),

(11,9373798798739878,28,'2018-11-27'),

(12,7738767787877678,150,'2018-07-17'),

(13,7987878787987898,350,'2018-11-23'),

(14,7798978987767769,100,'2018-10-25'),

(15,2546577987358778,350,'2018-04-05'),

(16,9887773798667366,350,'2018-04-05'),

(17,9887773798667366,350,'2018-04-05'),

(18,6487378787677736,350,'2018-07-17'),

(19,1787898273767489,350,'2018-07-17'),

(20,3697980307687367,350,'2018-07-17'),

(21,0937937794739787,100,'2018-10-25'),

(22,9372789498837984,100,'2018-10-25'),

(23,1943989898498709,150,'2018-10-25'),

(24,3893843479387898,150,'2018-10-25'),

(25,3439837987298738,75,'2018-10-25'),

(26,9033663897436788,75,'2018-11-23'),

(27,3980997487748978,90,'2018-11-23'),

(28,6479833988778878,100,'2018-11-23'),

(29,1292983939438283,350,'2018-11-23'),

(30,9374387389739398,100,'2018-11-23'),

(31,8737387789798788,150,'2018-11-23'),

(32,3928979879837878,75,'2018-11-23'),

(33,9887487874987898,90,'2018-10-30'),

(34,8747879879878897,100,'2018-10-30'),

(35,8998779889898798,90,'2018-10-30'),

(36,4597899879889399,100,'2018-10-30'),

(37,2838948778787849,100,'2018-10-30'),

(38,1298978979898988,100,'2018-10-30'),

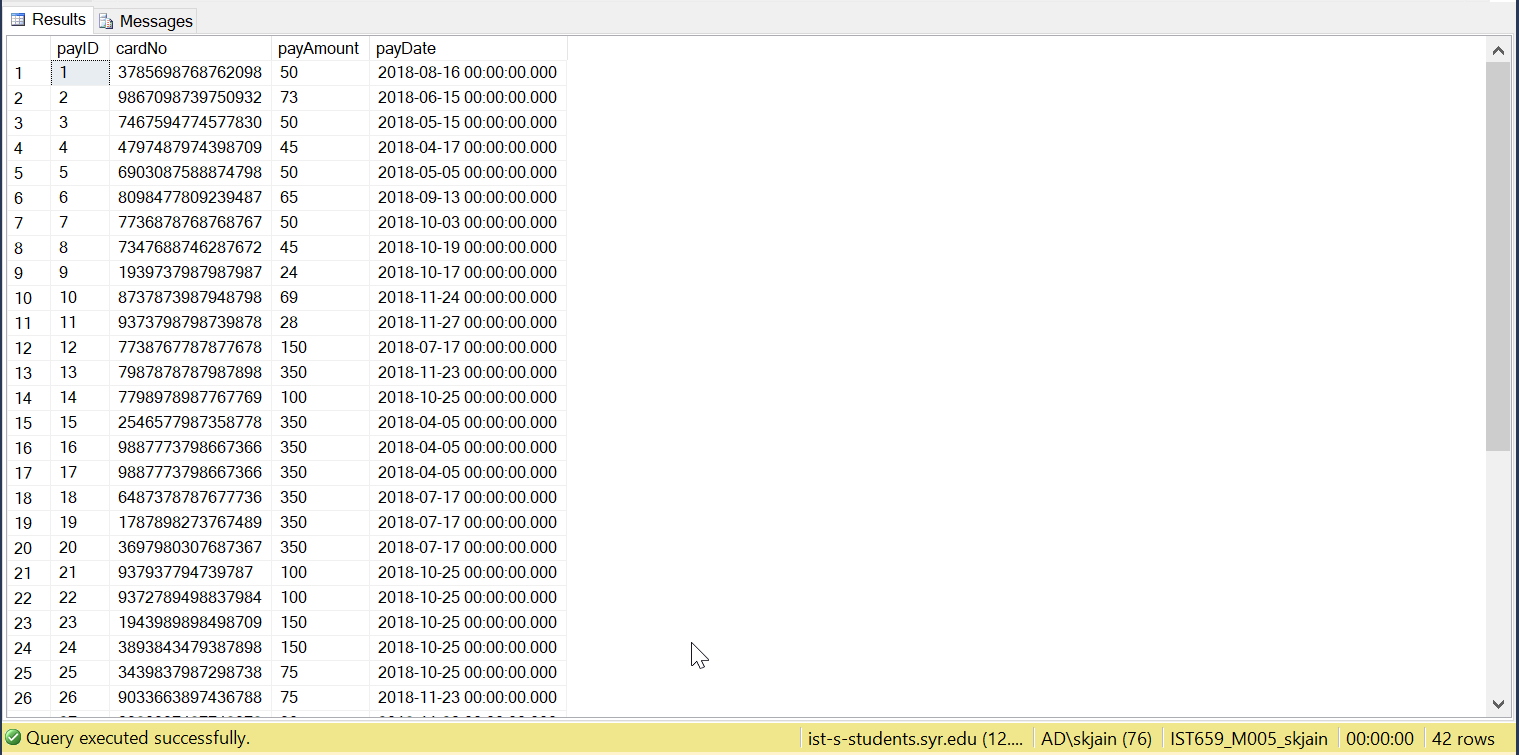
(39,7977377898387879,150,'2018-11-20'),

(40,4877987779798798,100,'2018-11-20'),

(41,9877868746767674,350,'2018-11-20'),

(42,1209098873779987,150,'2018-11-20')

select \* from Payment



--Populating the Trainer Table

INSERT INTO Trainer

VALUES

(1,1,'Personal Trainer',2),

(2,2,'Personal Trainer',3),

(3,5,'Floor Trainer',1),

(4,6,'Floor Trainer',1),

(5,7,'Personal Trainer',3),

(6,8,'Floor Trainer',1)

(7,13,'Floor Trainer',2),

(8,14,'Floor Trainer',3),

(9,15,'Floor Trainer',3),

(10,16,'Personal Trainer',3),

(11,17,'Personal Trainer',4),

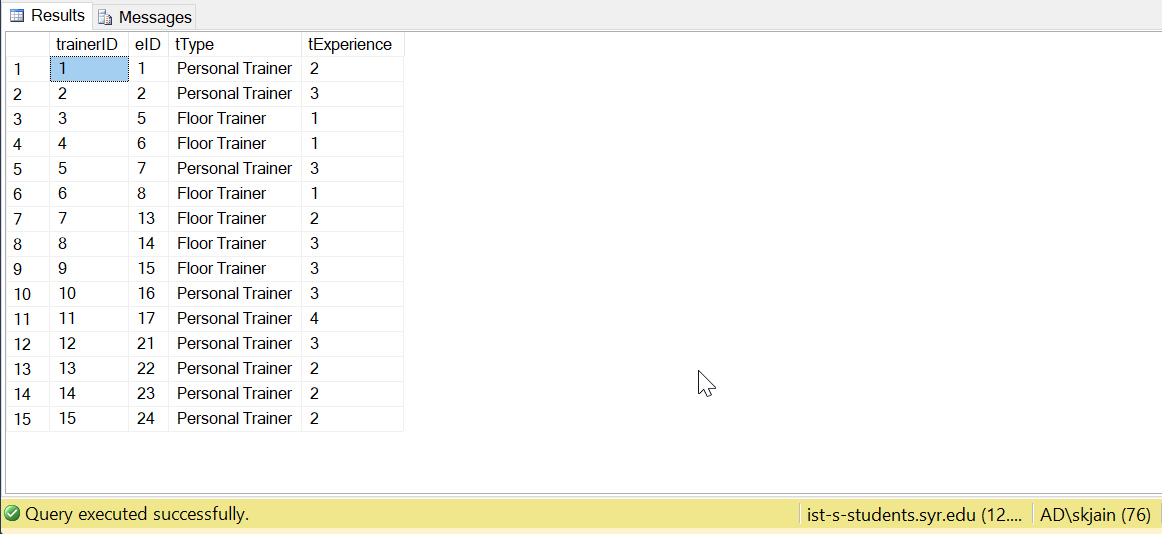
(12,21,'Personal Trainer',3),

(13,22,'Personal Trainer',2),

(14,23,'Personal Trainer',2),

(15,24,'Personal Trainer',2)

select \* from Trainer



--Populating the Receptionist Table

Insert into Receptionist

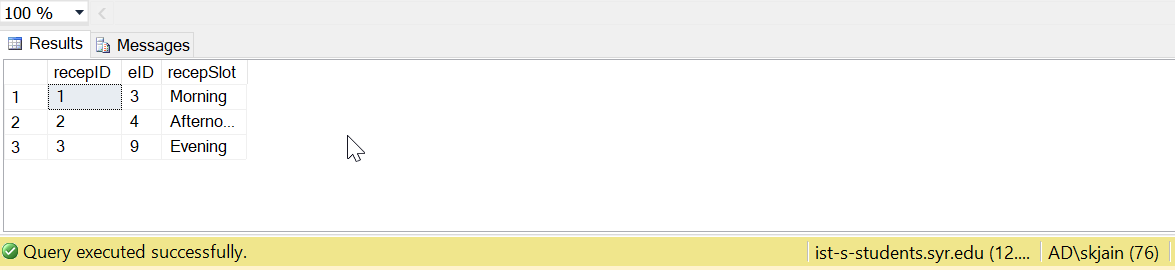
Values

(1,3,'Morning'),

(2,4,'Afternoon'),

(3,9,'Evening')

select\* from Receptionist



--Populating the Member Table

Insert into Member

VALUES

(1,'Shwet','Jain','M',21,3157068148,716,'SouthBeech',1),

(2,'Yash','Bhansali','M',23,3158075163,712,'SouthBeech',2),

(3,'Devansh','Khakhar','M',22,3159806783,710,'Westcott',5),

(4,'Harsh','Darji','M',21,3158469372,654,'RedField',3),

(5,'Emma','Shah','F',24,3159084356,456,'Waverly',6),

(7,'Ritika','Watson','F',29,3158569871,1020,'Westcott',10),

(8,'Zaheer','Khan','M',29,3158754354,210,'Westcott',11),

(9,'Chris','Moris','M',29,3158790567,987,'SouthBeech',10),

(10,'Steve','Smith','M',29,3155765788,1020,'Maryland',10),

(11,'Pravesh','Jain','M',29,3156576576,980,'Maryland',1),

(12,'Dishan','Bokadia','M',29,3155896876,546,'Maryland',12),

(13,'Shlok','Chauhan','M',29,3159875647,987,'Colombus',12),

(14,'Arpan ','Rathod','M',29,3152434333,321,'RedField',13),

(15,'Rishi','Smith','M',29,3159876477,789,'Waverly',14),

(16,'Akshat','Fagania','M',29,3159896764,798,'Westcott',13),

(17,'Anisha','Desai','F',29,3152435687,987,'RedField',2),

(18,'Trishna','Soor','F',29,3159776587,987,'Waverly',1),

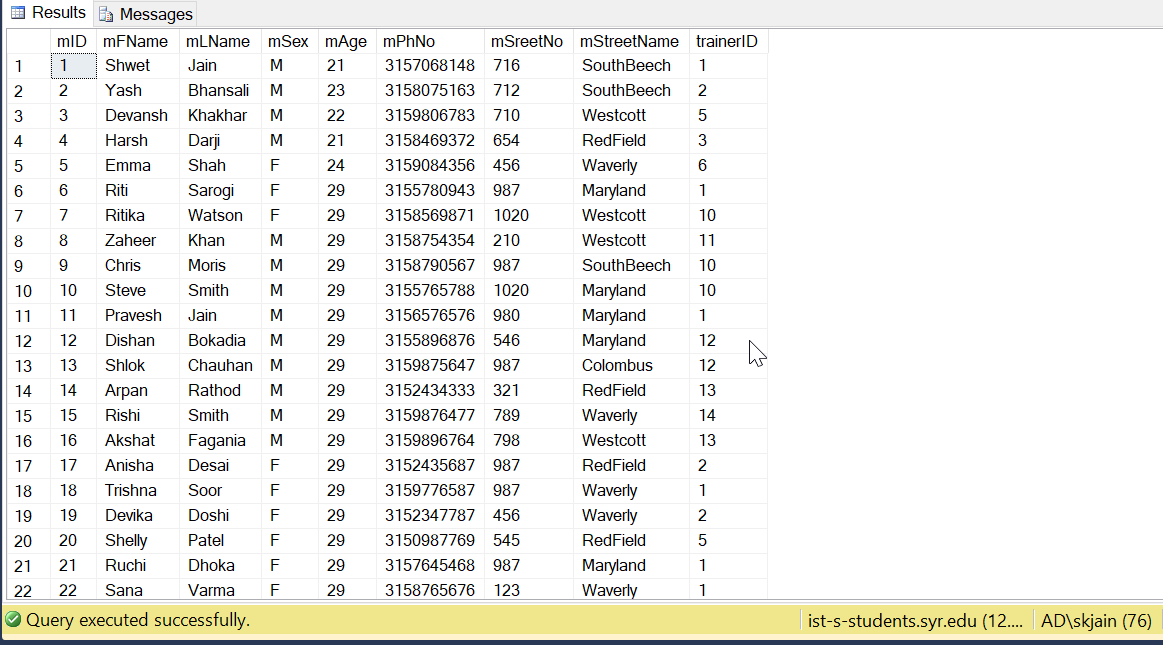
(19,'Devika','Doshi','F',29,3152347787,456,'Waverly',2),

(20,'Shelly','Patel','F',29,3150987769,545,'RedField',5),

(21,'Ruchi','Dhoka','F',29,3157645468,987,'Maryland',1),

(22,'Sana','Varma','F',29,3158765676,123,'Waverly',1)

select \* from Member



--Populating the Membership Table

Insert into Membership

VALUES

(1,3,'2018-07-17',2,'2018-10-17',1),

(2,1,'2018-11-23',4,'2019-11-23',2),

(2,2,'2018-10-25',3,'2019-1-25',3)

(3,1,'2018-04-05',1,'2019-04-05',1),

(4,1,'2018-04-05',1,'2019-04-05',2),

(5,1,'2018-04-05',1,'2019-04-05',3),

(6,1,'2018-07-17',2,'2019-07-17',1),

(7,1,'2018-07-17',2,'2019-07-17',2),

(8,1,'2018-07-17',2,'2019-07-17',3),

(9,2,'2018-10-25',3,'2019-1-25',1),

(10,2,'2018-10-25',3,'2019-11-23',2),

(11,3,'2018-10-25',3,'2019-11-23',3),

(12,3,'2018-10-25',3,'2019-11-23',1),

(13,4,'2018-10-25',3,'2019-11-23',2),

(14,4,'2018-11-23',4,'2019-02-23',3),

(15,5,'2018-11-23',4,'2019-02-23',1),

(16,6,'2018-11-23',4,'2019-01-23',2),

(17,1,'2018-11-23',4,'2019-11-23',3),

(18,2,'2018-11-23',4,'2019-02-23',1),

(19,3,'2018-11-23',4,'2019-02-23',2),

(20,4,'2018-11-23',4,'2019-02-23',3),

(21,5,'2018-10-30',5,'2019-01-30',1),

(22,6,'2018-10-30',5,'2018-12-30',2),

(10,5,'2018-10-30',5,'2019-01-30',3),

(11,6,'2018-10-30',5,'2018-12-30',1),

(12,6,'2018-10-30',5,'2018-12-30',2),

(13,6,'2018-10-30',5,'2018-12-30',3),

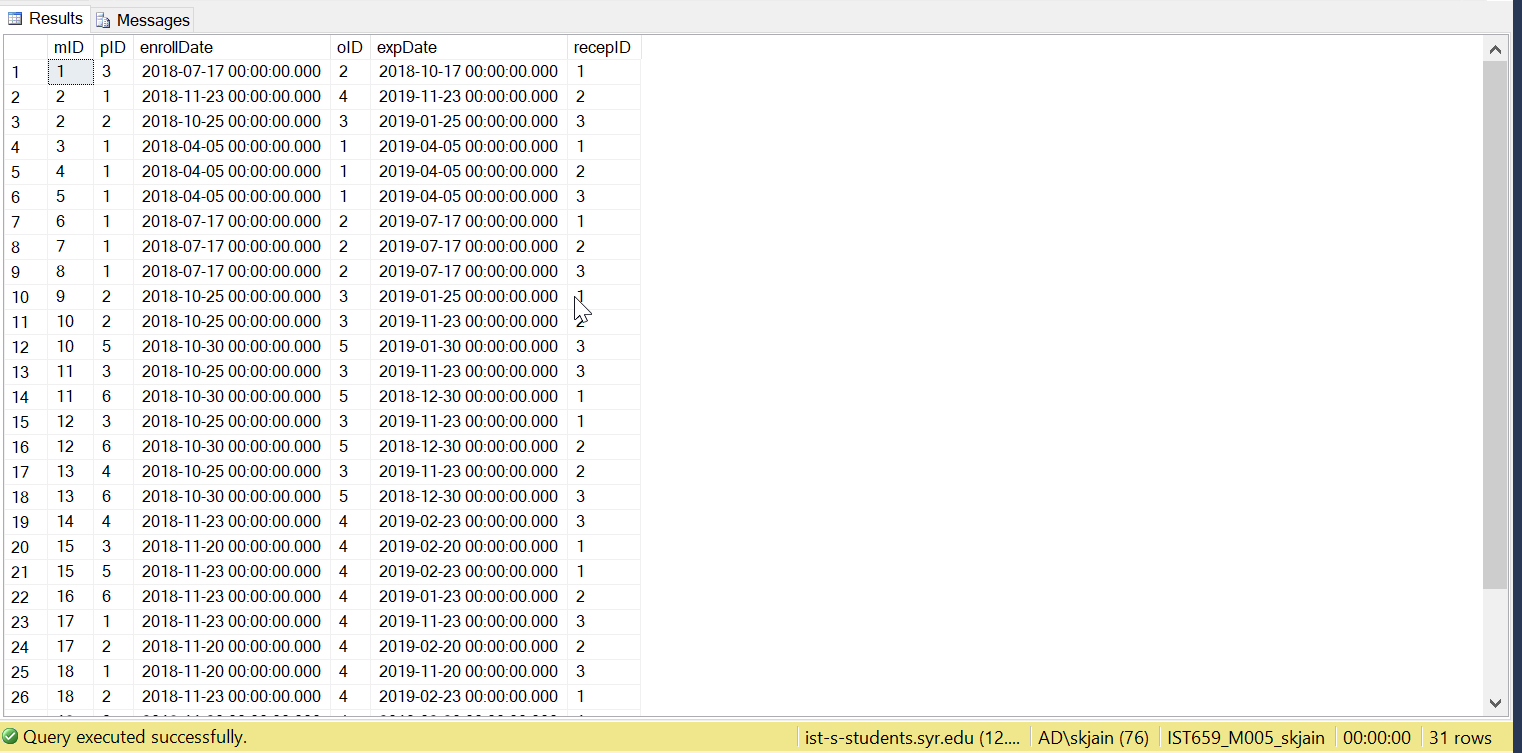
(15,3,'2018-11-20',4,'2019-02-20',1),

(17,2,'2018-11-20',4,'2019-02-20',2),

(18,1,'2018-11-20',4,'2019-11-20',3),

(18,3,'2018-11-20',4,'2019-02-20',1)

Select\* from Membership



--Populating the Order Table

Insert into order1

VALUES

(1,1,'2018-08-16'),

(2,2,'2018-06-15'),

(3,12,'2018-05-15'),

(4,17,'2018-04-17'),

(5,7,'2018-05-05'),

(6,9,'2018-09-13'),

(7,9,'2018-10-03'),

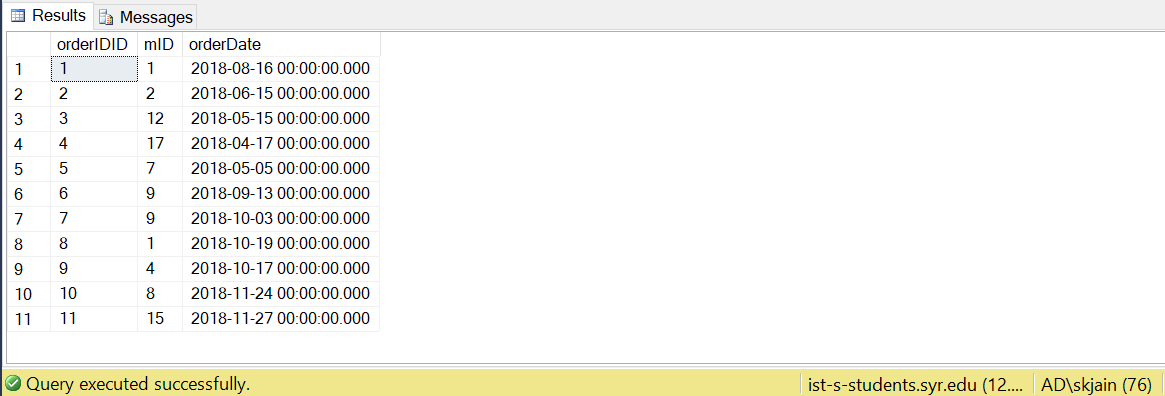
(8,1,'2018-10-19'),

(9,4,'2018-10-17'),

(10,8,'2018-11-24'),

(11,15,'2018-11-27')

select \* from order1



--Populating the OrderLine Table

Insert Into orderline1

Values

(1,14534,1,1),

(2,23434,2,1),

(2,32432,2,1),

(3,14534,12,1),

(4,23434,17,1),

(5,14534,7,1),

(6,23434,9,1),

(6,23433,9,1),

(7,14534,9,1),

(8,67389,1,1),

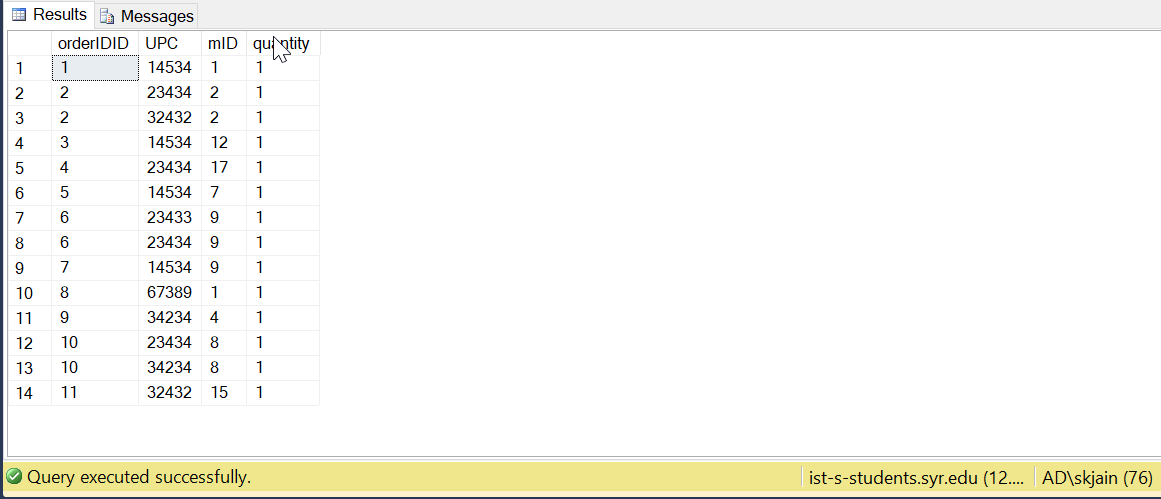
(9,34234,4,1),

(10,23434,8,1),

(10,34234,8,1),

(11,32432,15,1)

select \* from orderLine1



--Populating the OrderPayment Table

Insert Into orderPayment

Values

(1,1,1,50),

(2,2,2,73),

(3,3,12,50),

(4,4,17,45),

(5,5,7,50),

(6,6,9,65),

(7,7,9,50),

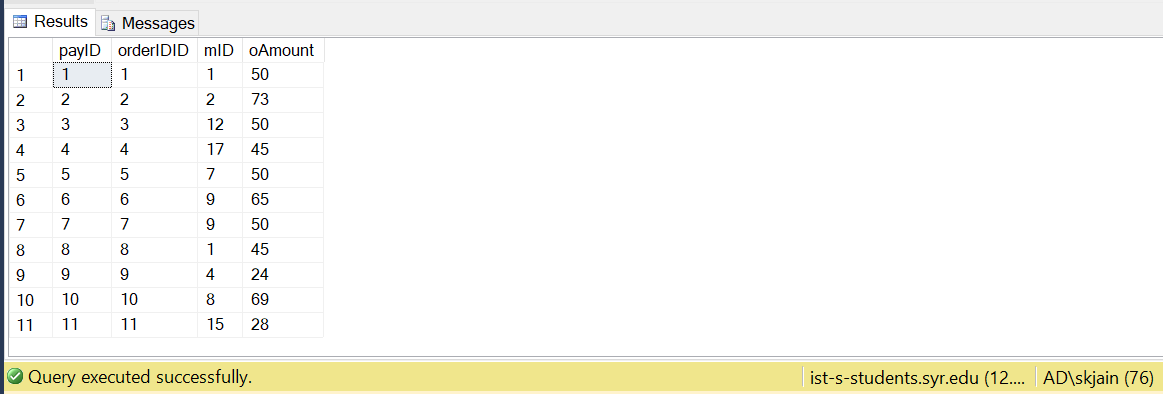
(8,8,1,45),

(9,9,4,24),

(10,10,8,69),

(11,11,15,28)

select \* from orderPayment



--Populating the PlanPayment Table

Insert Into planPayment

Values

(12,3,1,'2018-07-17'),

(13,1,2,'2018-11-23'),

(14,2,2,'2018-10-25'),

(15,1,3,'2018-04-05'),

(16,1,4,'2018-04-05'),

(17,1,5,'2018-04-05'),

(18,1,6,'2018-07-17'),

(19,1,7,'2018-07-17'),

(20,1,8,'2018-07-17'),

(21,2,9,'2018-10-25'),

(22,2,10,'2018-10-25'),

(23,3,11,'2018-10-25'),

(24,3,12,'2018-10-25'),

(25,4,13,'2018-10-25'),

(26,4,14,'2018-11-23'),

(27,5,15,'2018-11-23'),

(28,6,16,'2018-11-23'),

(29,1,17,'2018-11-23'),

(30,2,18,'2018-11-23'),

(31,3,19,'2018-11-23'),

(32,4,20,'2018-11-23'),

(33,5,21,'2018-10-30'),

(34,6,22,'2018-10-30'),

(35,5,10,'2018-10-30'),

(36,6,11,'2018-10-30'),

(37,6,12,'2018-10-30'),

(38,6,13,'2018-10-30'),

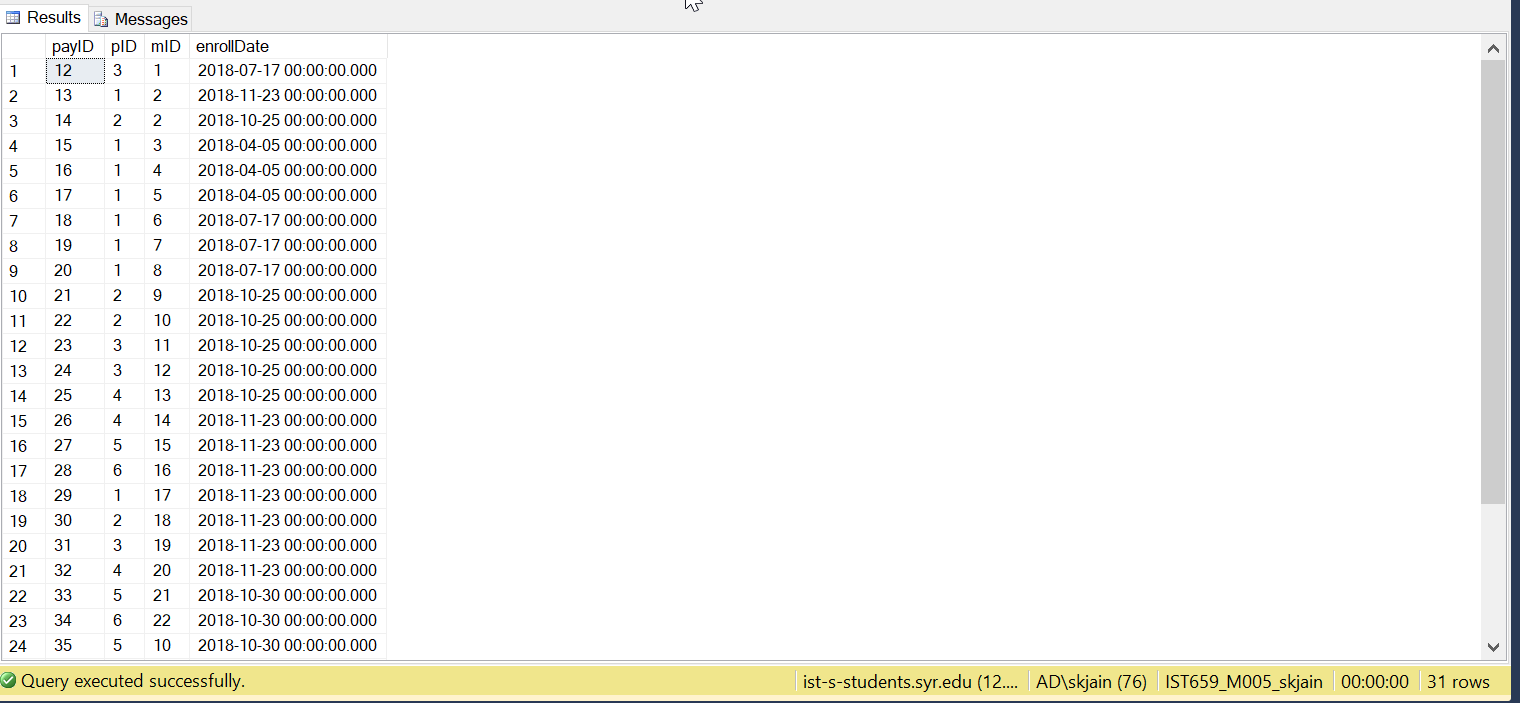
(39,3,15,'2018-11-20'),

(40,2,17,'2018-11-20'),

(41,1,18,'2018-11-20'),

(42,3,18,'2018-11-20')

select \* from planPayment



MAJOR DATA QUESTIONS:

--Major Data Questions

1. Which member's membership is getting expired between a particular date and for which plan along with the date of expiry? Show Member ID,Member’s First Name, Member’s Last Name, Plan ID, Plan Name and Expiry Date.

* Why is this Query important and how will it impact the business?

**It is important to know when a particular member’s plan is getting expired so that we can send them a reminder for renewal of the Plan (via Member’s Ph.no)**

[Customer Satisfaction and Drive Business]

SELECT mem.mID,mFName,mLName,p.pID,pName,expDate

FROM member mem

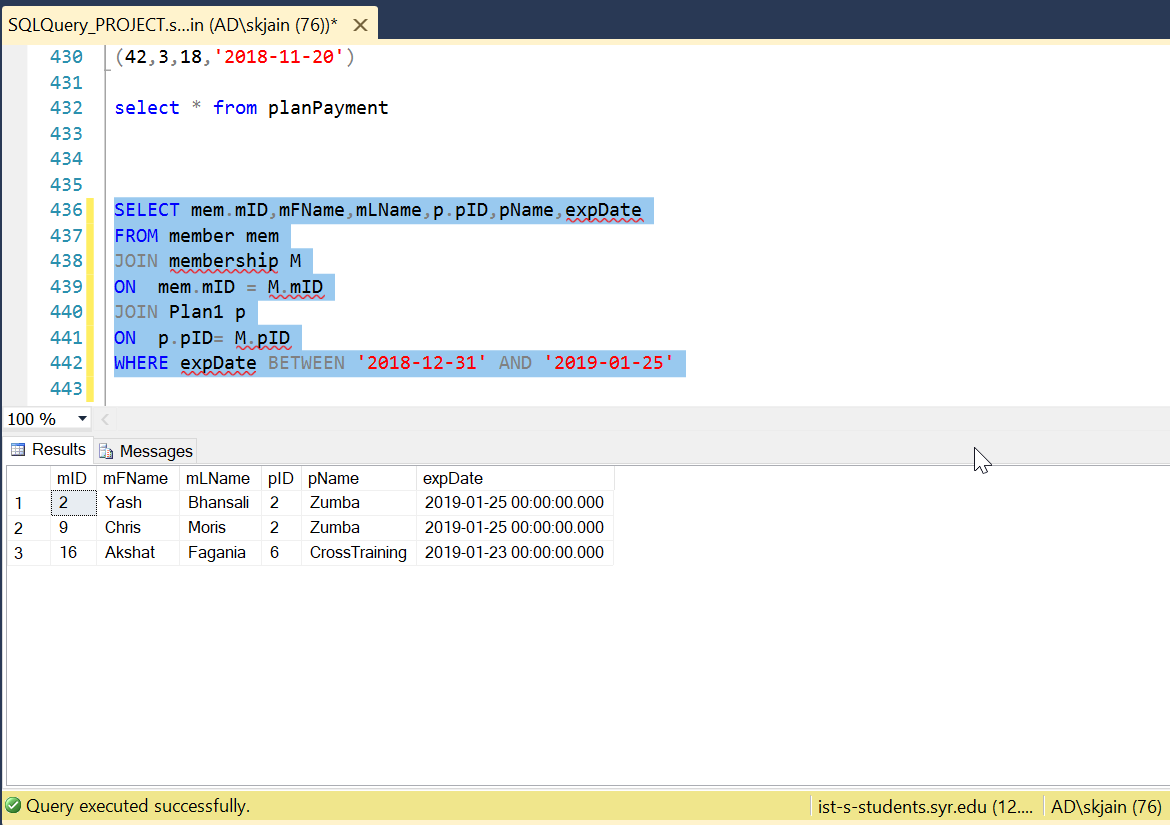
JOIN membership M

ON mem.mID = M.mID

JOIN Plan1 p

ON p.pID= M.pID

WHERE expDate BETWEEN '2018-12-31' AND '2019-01-25'



1. Display the ‘Personal Trainers’ and the no. of Members each of them is training. Show TrainerID, Trainer’s First Name

* Why is this Query important and how will it impact the business?

**There is a limit to no. of member’s a trainer can train.(Max=7)**

**Hence, when a member is associated with a trainer this query is useful.**

[Efficiency and to ensure no violation of business rules]

Select T.trainerID,eFName,COUNT(E.eID) 'No.of Members Trained'

from Trainer T

Join Member mem

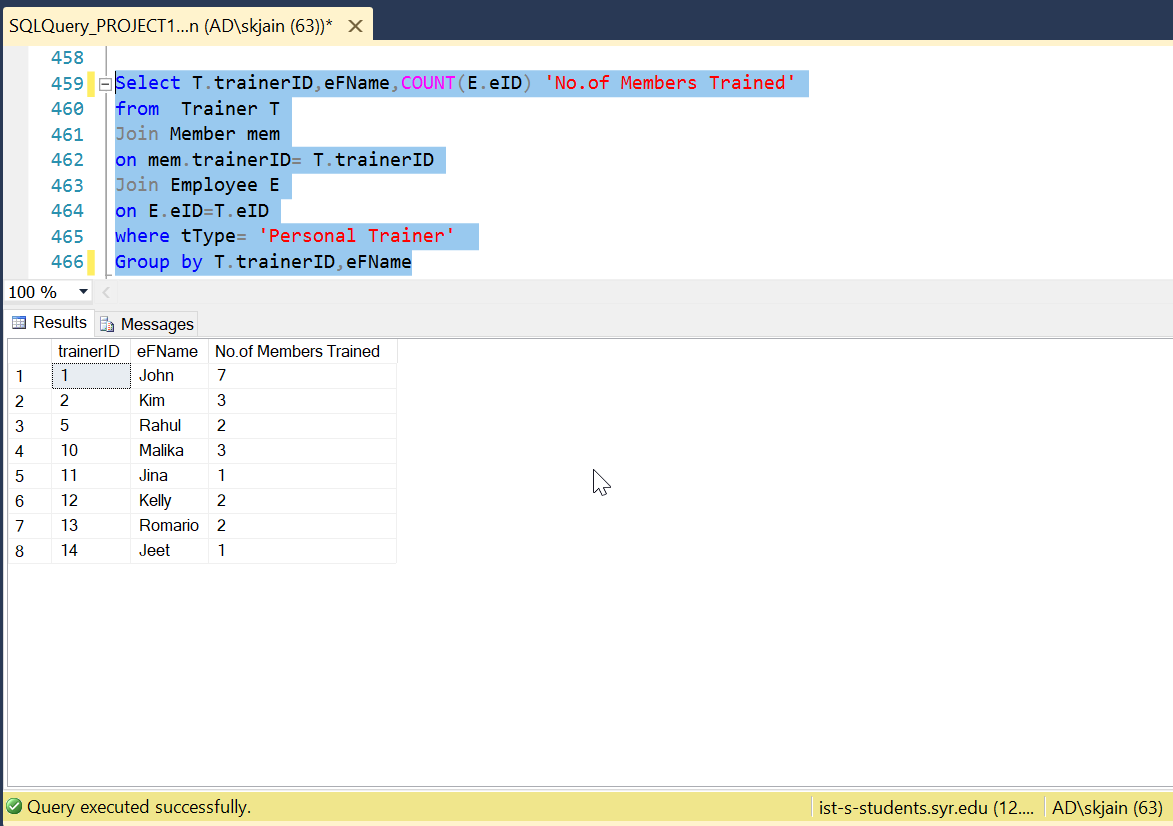
on mem.trainerID= T.trainerID

Join Employee E

on E.eID=T.eID

where tType= 'Personal Trainer'

Group by T.trainerID,eFName



1. Number of Members enrolled for a particular plan ‘Regular’ with a particular offer ‘BlackFriday’ ? Show Plan ID, Plan Name

* Why is this Query important and how will it impact the business?

**This is a query which is useful in extracting information about a particular plan and offer.(Helpful for Data Analyst and the Gym owner to carry out analysis)**

[Insights about different plans and offer which can be useful in future]

Select P.pID,pName,oName,count(P.pID) 'No. Of Members enrolled'

from Member mem

JOIN Membership M

ON M.mID = mem.mID

Join Plan1 P

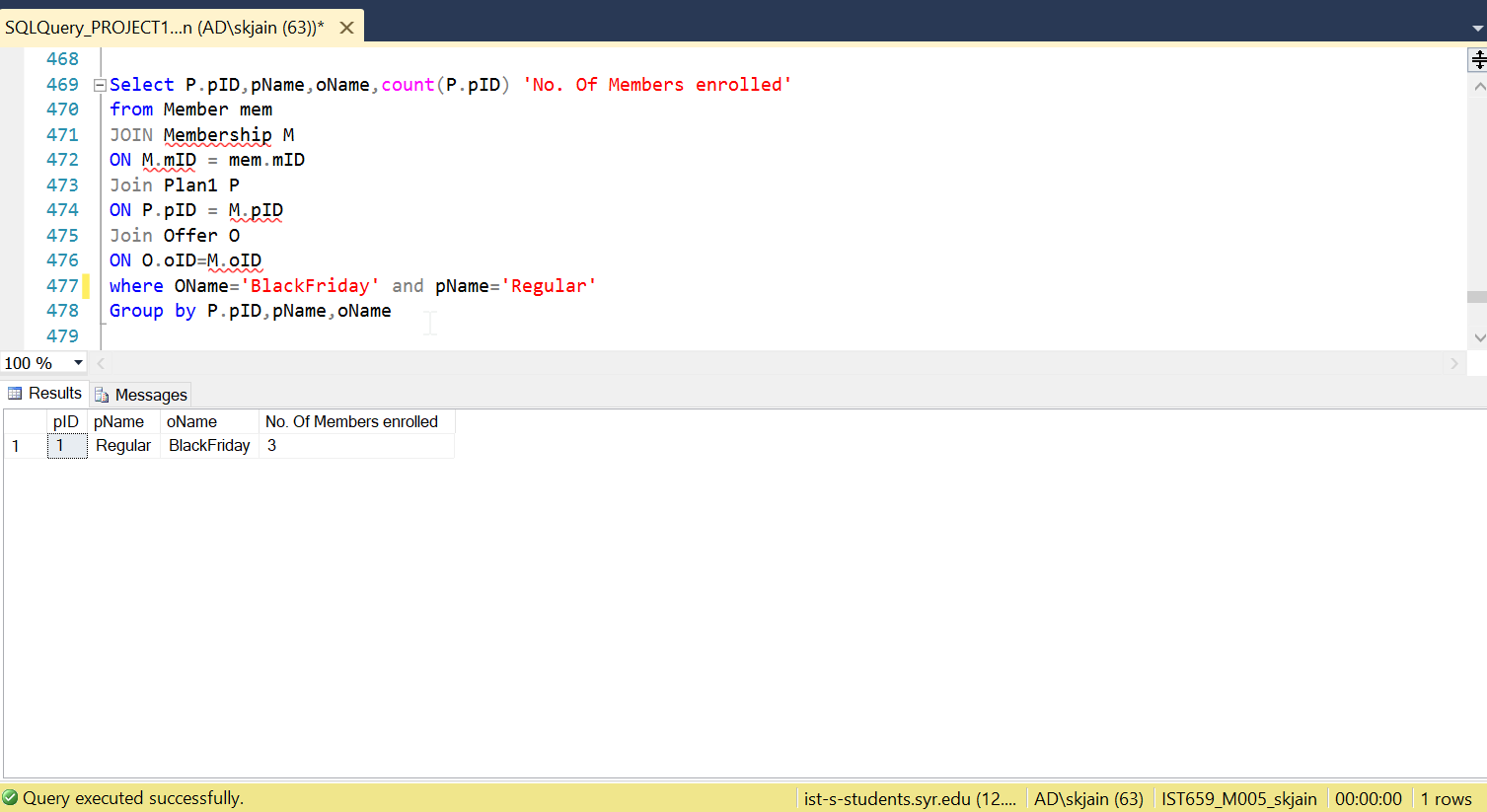
ON P.pID = M.pID

Join Offer O

ON O.oID=M.oID

where OName='BlackFriday' and pName='Regular'

Group by P.pID,pName,oName



1. Create a View to find out plans a particular member has enrolled for.

Find the plans for which Member ‘Yash’ has enrolled for. Show Member ID, First Name, Last Name, Plan ID, Plan Name

* Why is this Query important and how will it impact the business?

**It is possible to view different plans each member in the gym has enrolled for.**

**In the previous system, it was difficult and time-consuming as the system was paper-based.**

[Effective, efficient system functioning]

Create View View\_MP as

Select mem.mID,mFName,mLName,P.pID,pName

from Member mem

JOIN Membership M

ON M.mID = mem.mID

Join Plan1 P

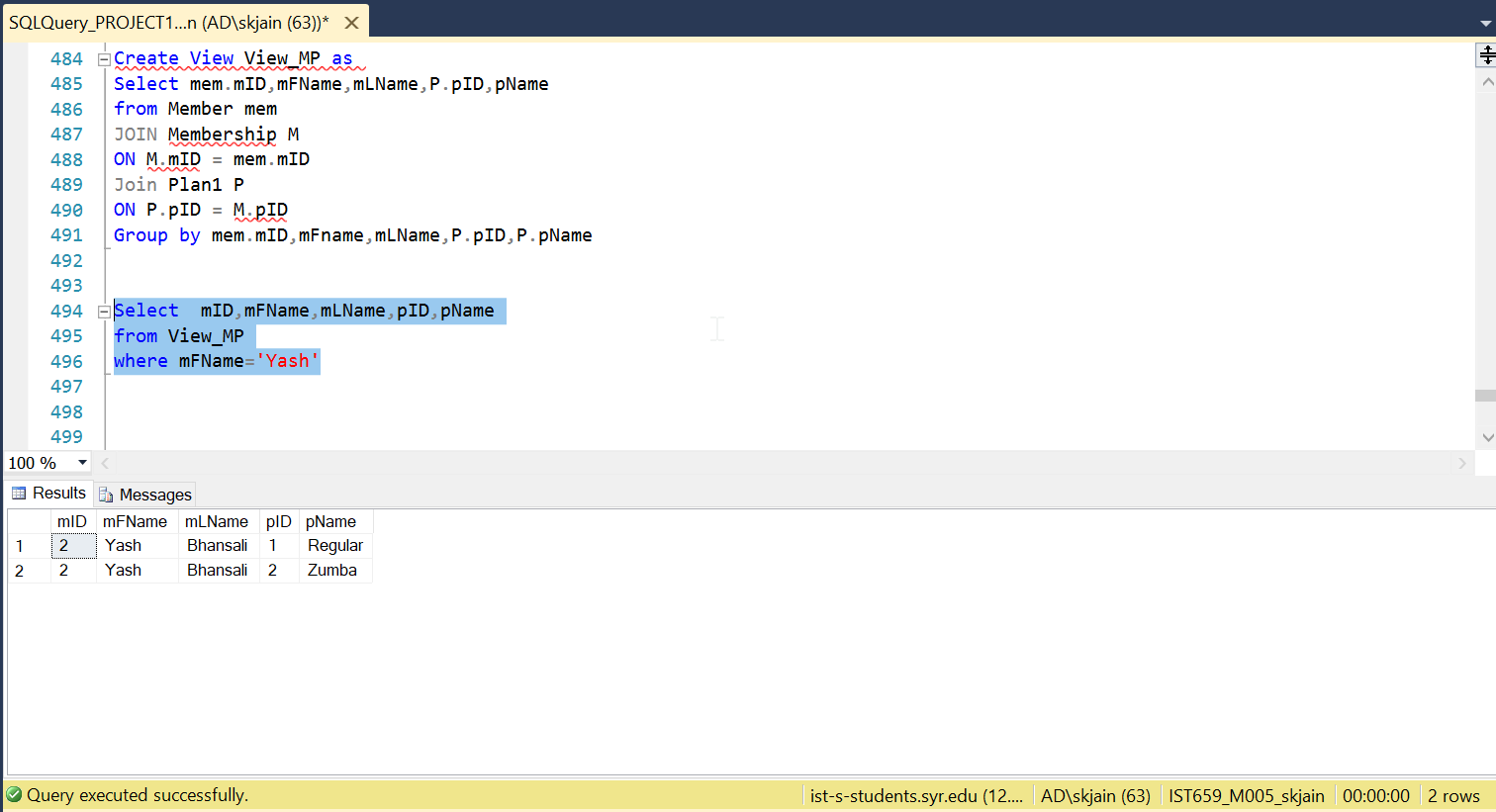
ON P.pID = M.pID

Group by mem.mID,mFname,mLName,P.pID,P.pName

Select \*

from View\_MP

where mFName='Yash'



1. Number of members enrolled in the gym from each Street.

* Why is this Query important and how will it impact the business?

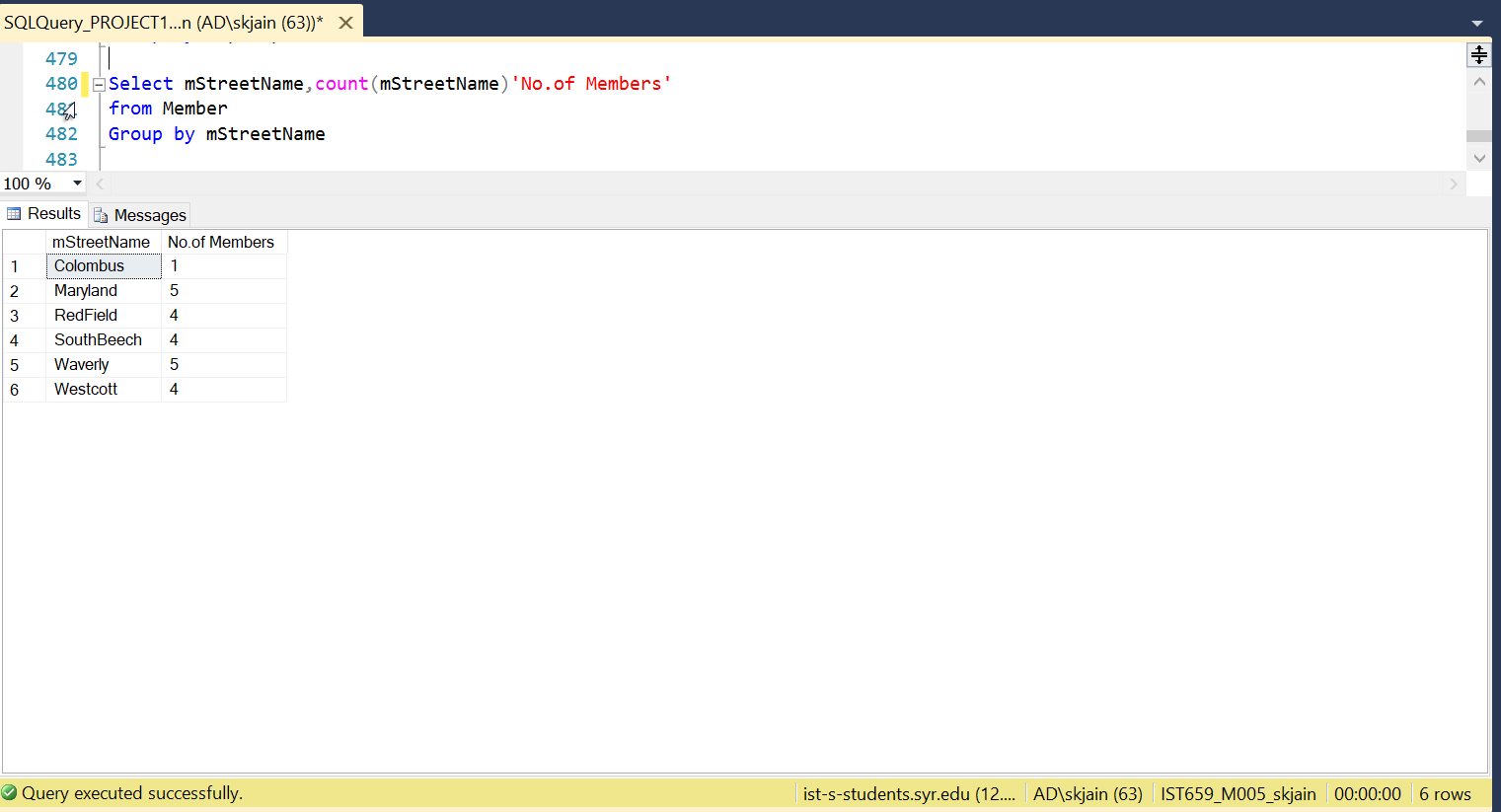
**It is important to know how many members are coming from a certain Street or area. This will help us in Marketing and targeting the audience in those Streets.**

[Marketing and thus improving business]

Select mStreetName,count(mStreetName)

from Member

Group by mStreetName



1. No. of products ordered by enrolled members in the gym.

Select mem.mID,mFName,mLName,count(mem.mID) 'No.of Products Ordered'

from Member mem

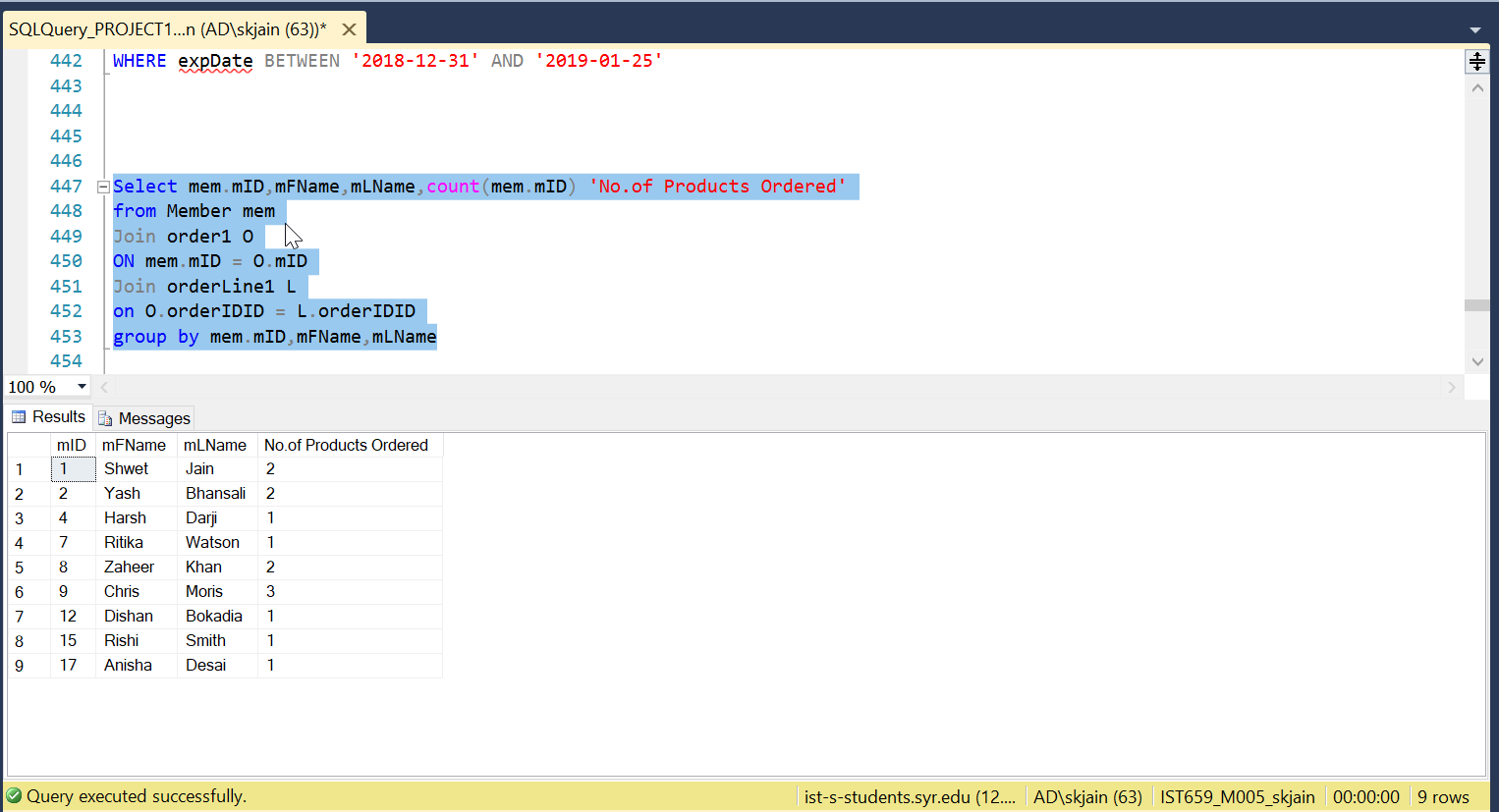
Join order1 O

ON mem.mID = O.mID

Join orderLine1 L

on O.orderIDID = L.orderIDID

group by mem.mID,mFName,mLName



1. Created a view to find the product which was ordered the most by the members. Show Product’s UPC, Product Name, Product Type.

* Why is this Query important and how will it impact the business?

**This query helps us to know which is the most popular product in the gym. This is helpful for inventory purposes to make sure such products are not out of stock**

**[Inventory, Customer Satisfaction and Analysis]**

Create View View\_ProdMax1 as

select OL.UPC,productName,productType,count(OL.UPC) 'No\_of\_Orders'

from order1 O

Join orderline1 OL

ON O.orderIDID = OL.orderIDID

Join Product1 P

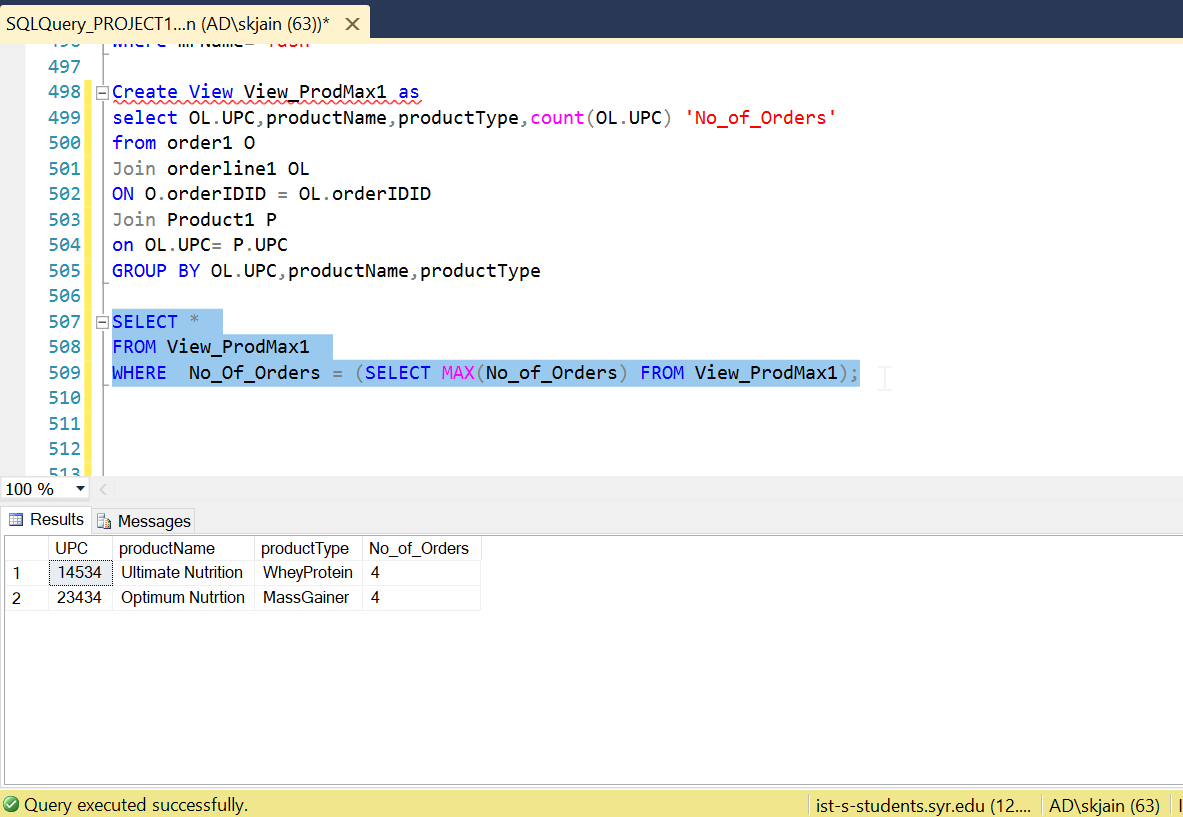
on OL.UPC= P.UPC

GROUP BY OL.UPC,productName,productType

SELECT \*

FROM View\_ProdMax1

WHERE No\_Of\_Orders = (SELECT MAX(No\_of\_Orders) FROM View\_ProdMax1);



1. Transaction (Member is enrolled for a particular plan):

* Why is this Query important and how will it impact the business?

**Suppose, I enrolled member X for a particular plan Y and say suppose that was a wrong entry or the entire transaction/Process of enrolling the member to a particular plan failed. It is important that we can Rollback and achieve the old/desired state.**

SET IMPLICIT\_TRANSACTIONS ON;

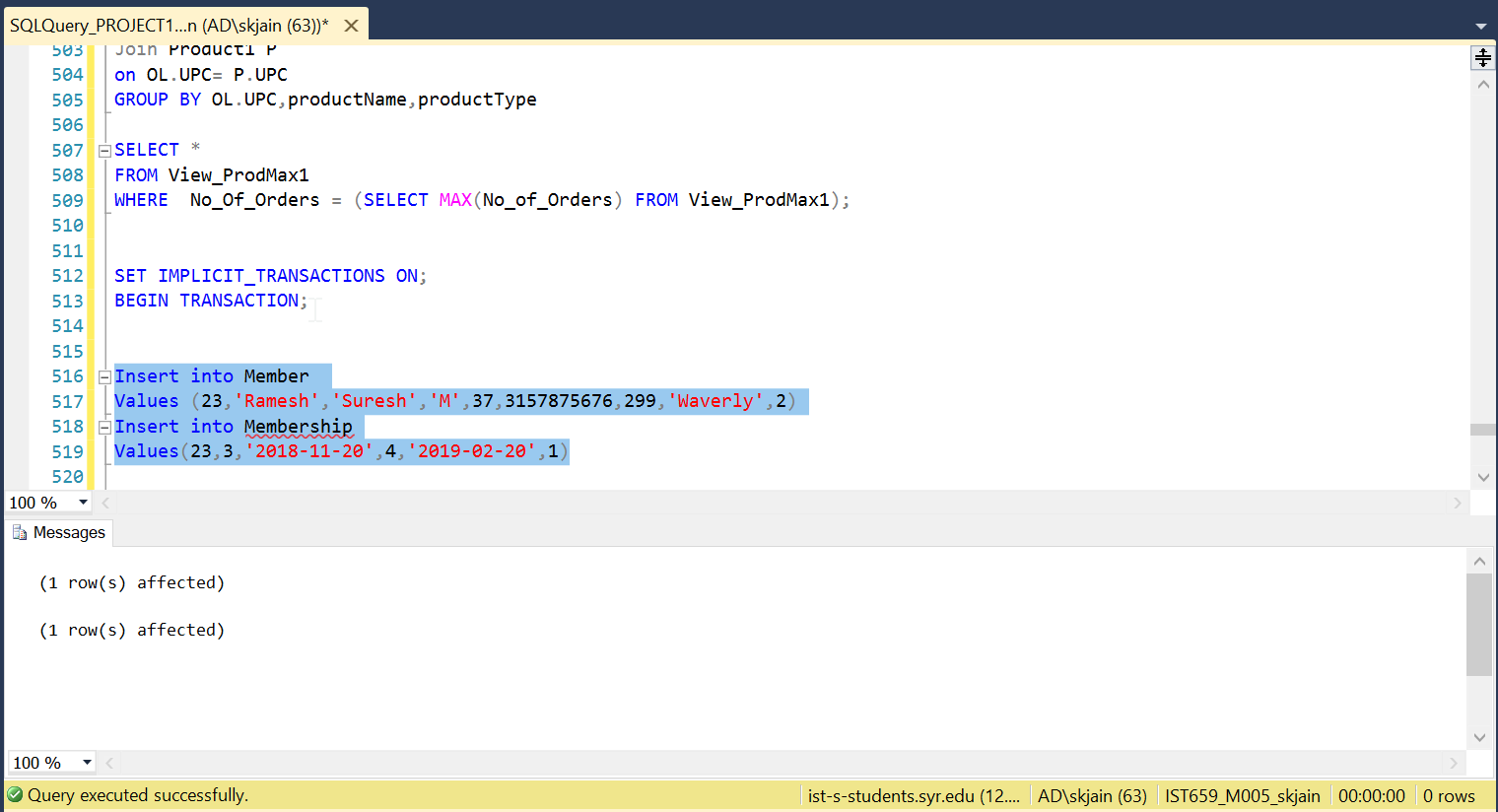
BEGIN TRANSACTION;

Insert into Member

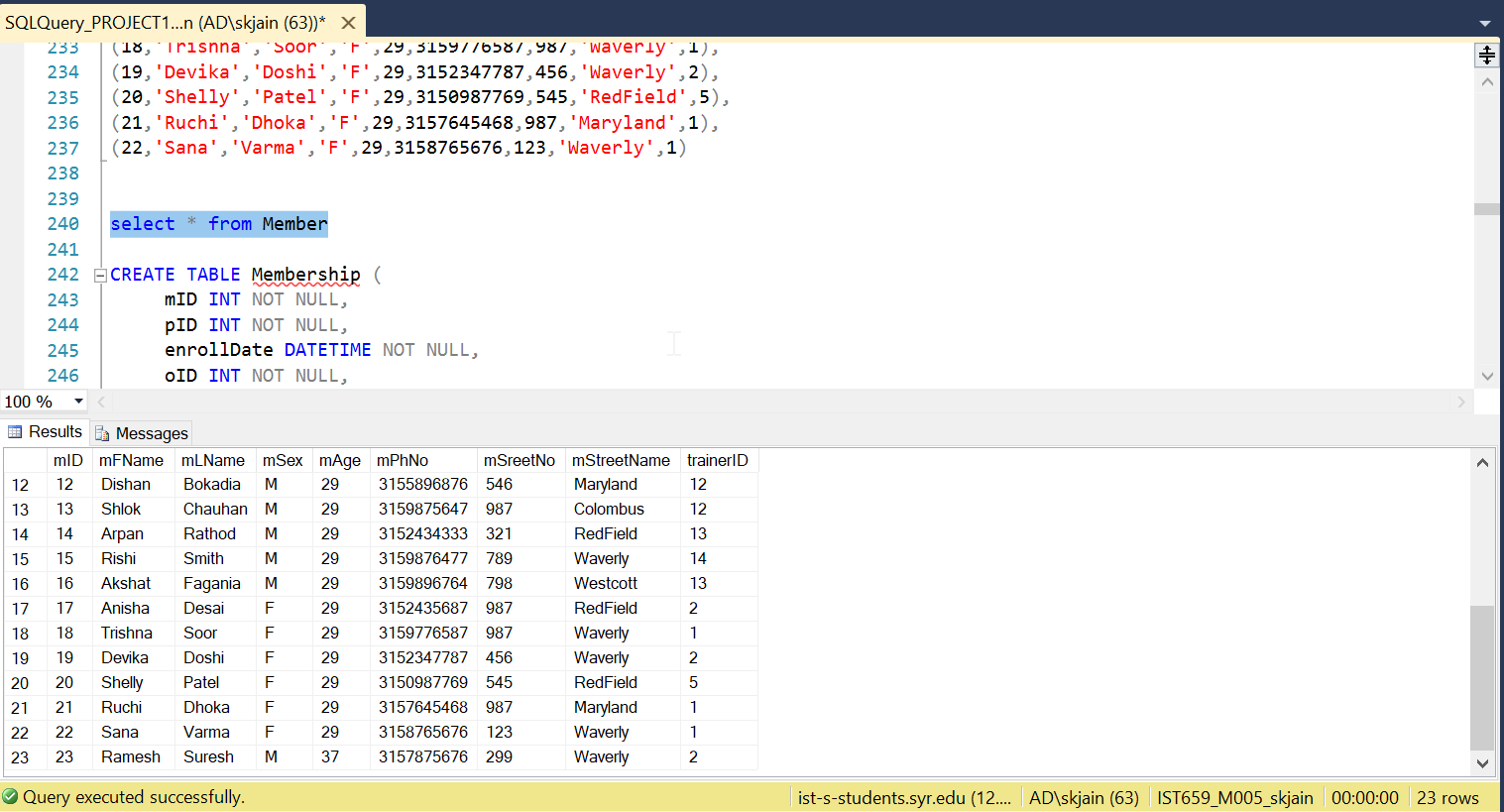
Values (23,'Ramesh','Suresh','M',37,3157875676,299,'Waverly',2)

Insert into Membership

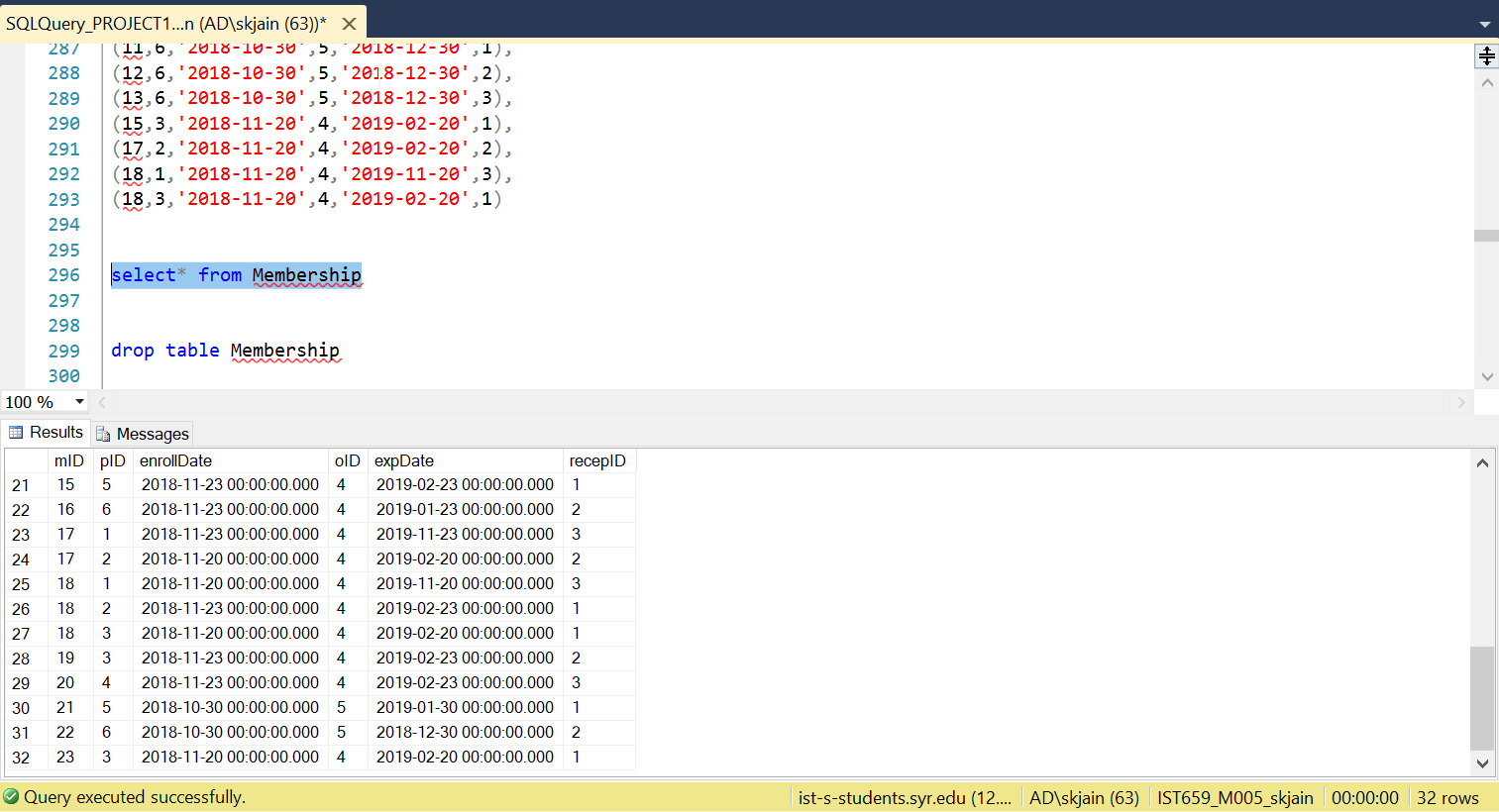
Values (23,3,'2018-11-20',4,'2019-02-20',1)



select \* from Member



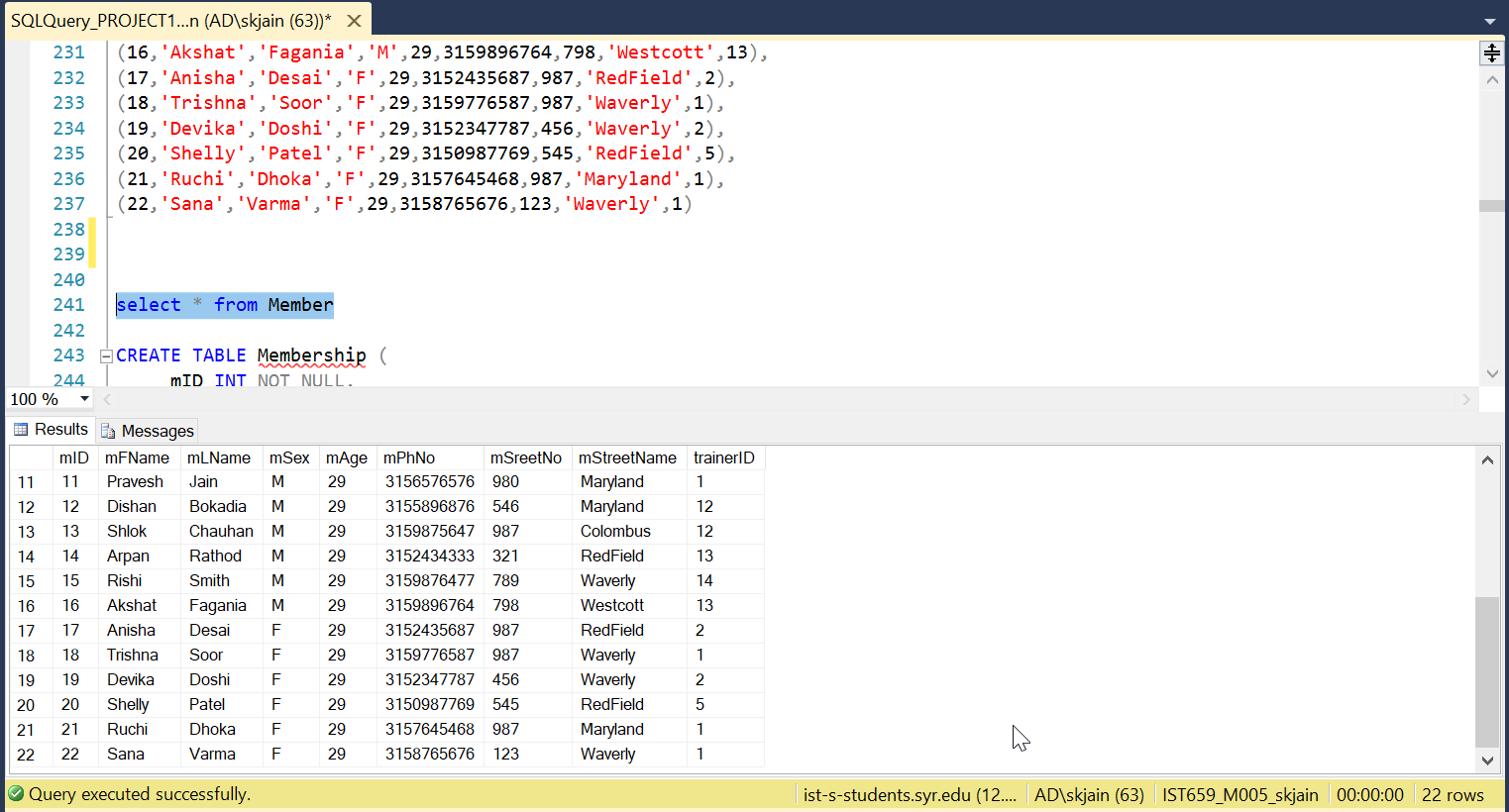
select\* from Membership



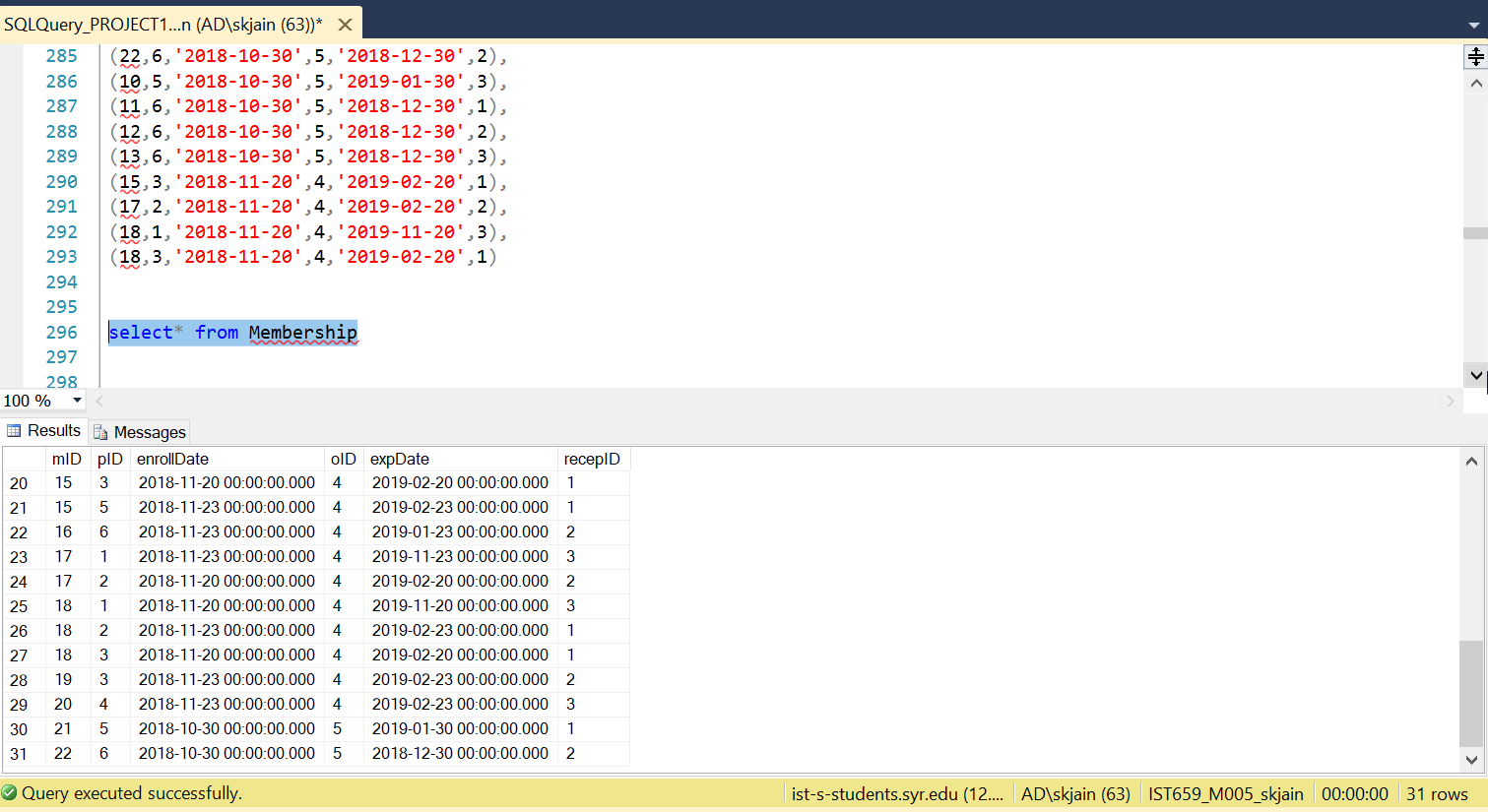
ROLLBACK --successfully rollback to the beginning of the transaction



select \* from Member

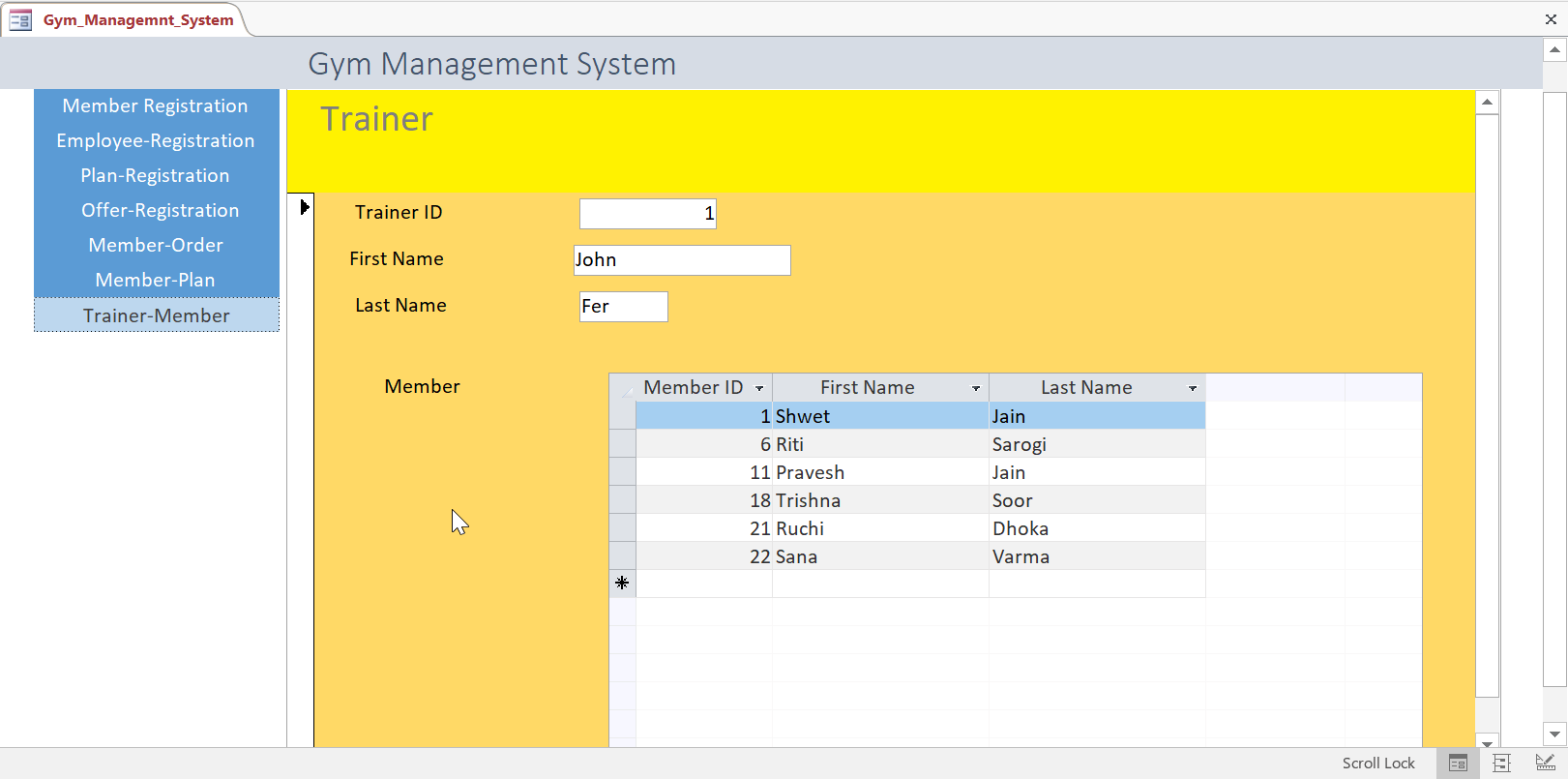


select\* from Membership

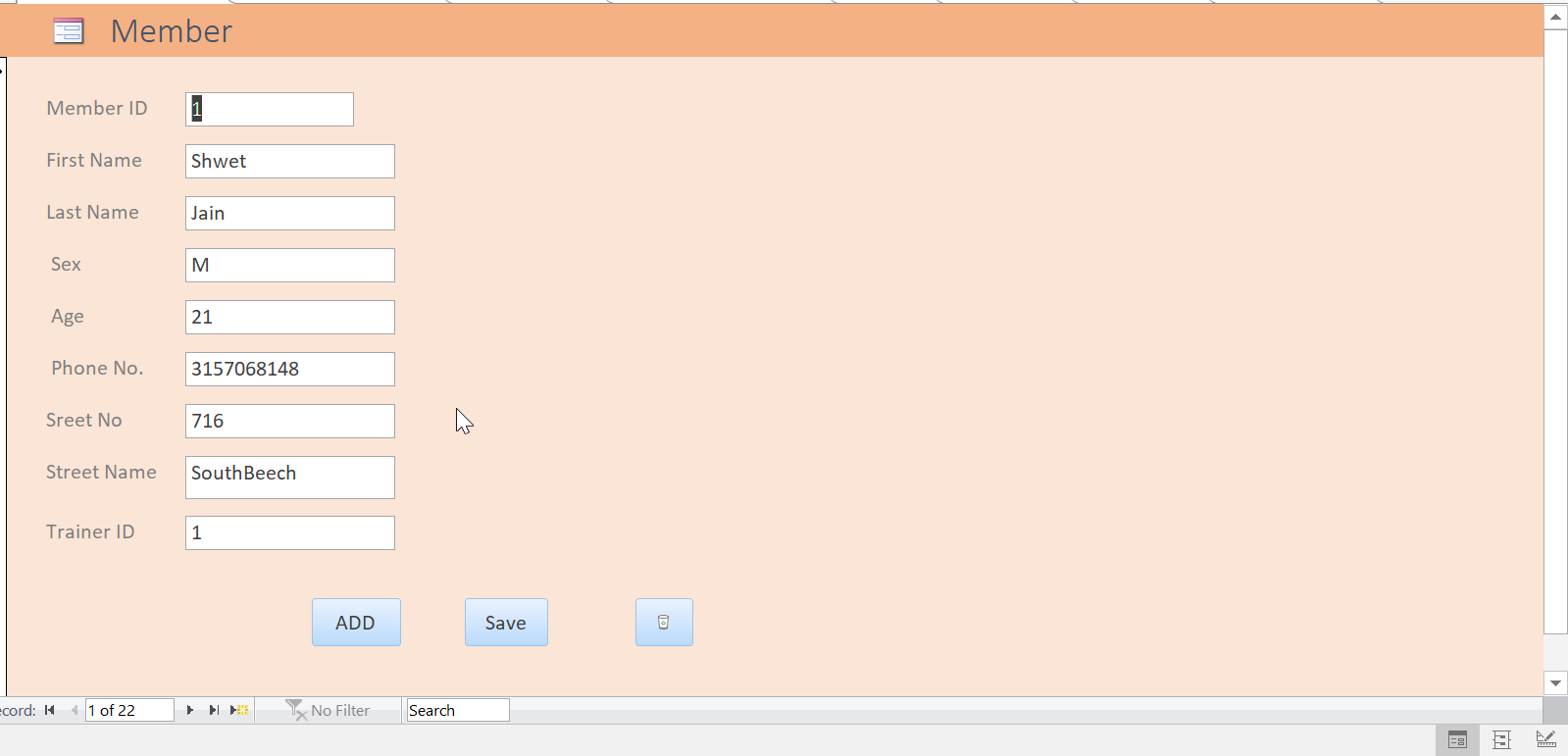


INTERFACES

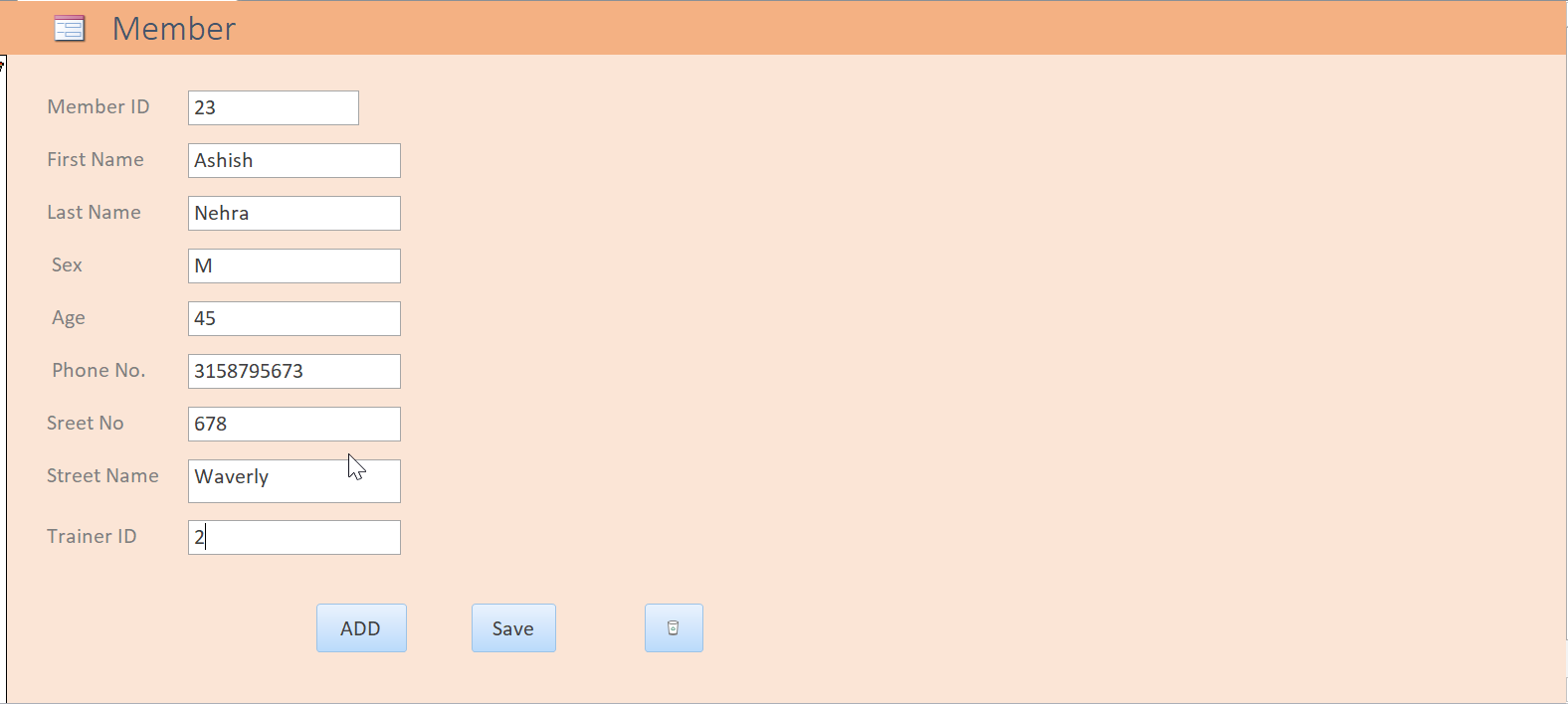
* Initial Home Page for Gym Management System



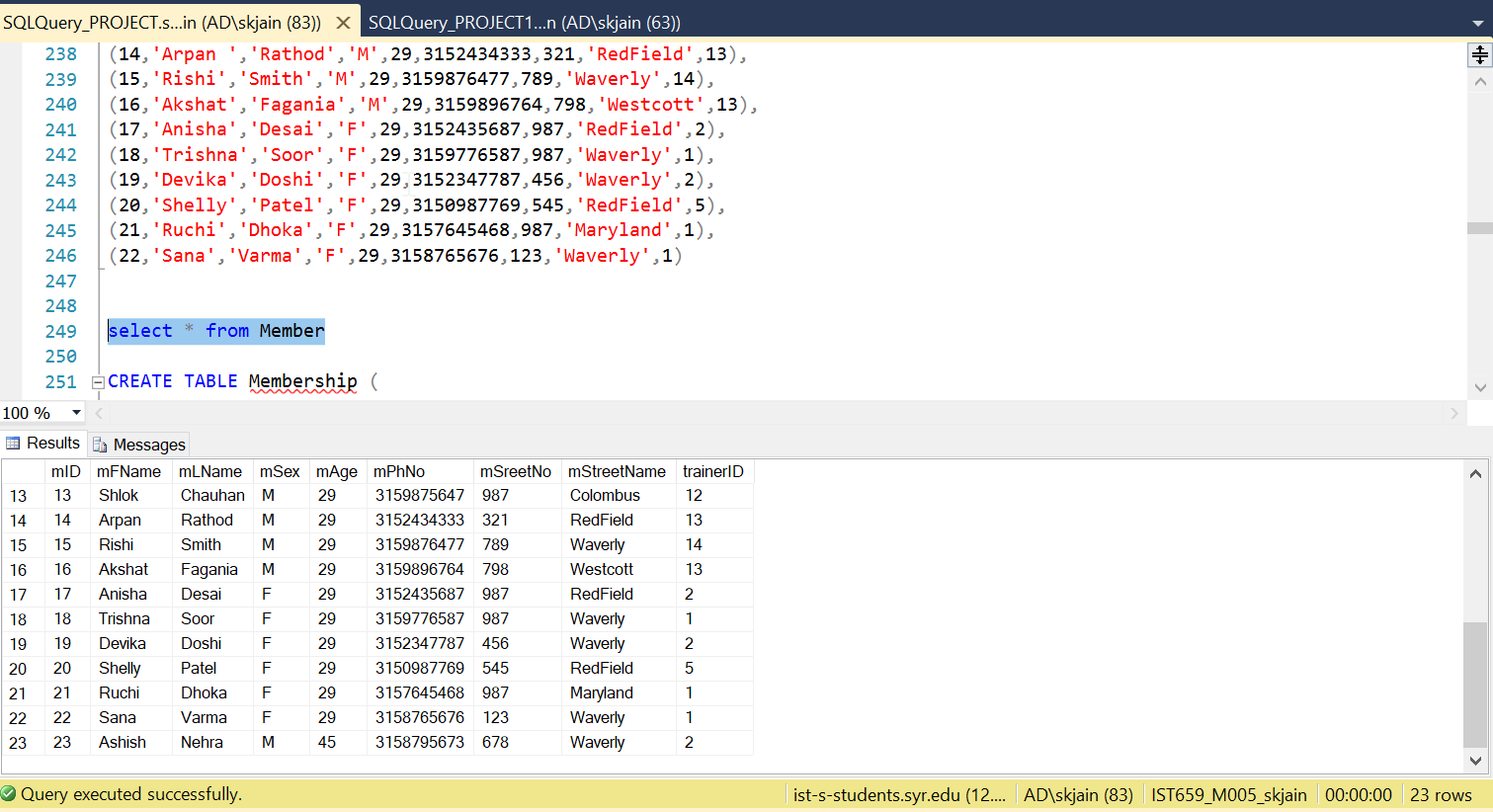
* As You Navigate through , first we come across Member Registration form



* You can simply add a new Member by clicking the ‘ADD’ button and then the ‘ Save ’ button

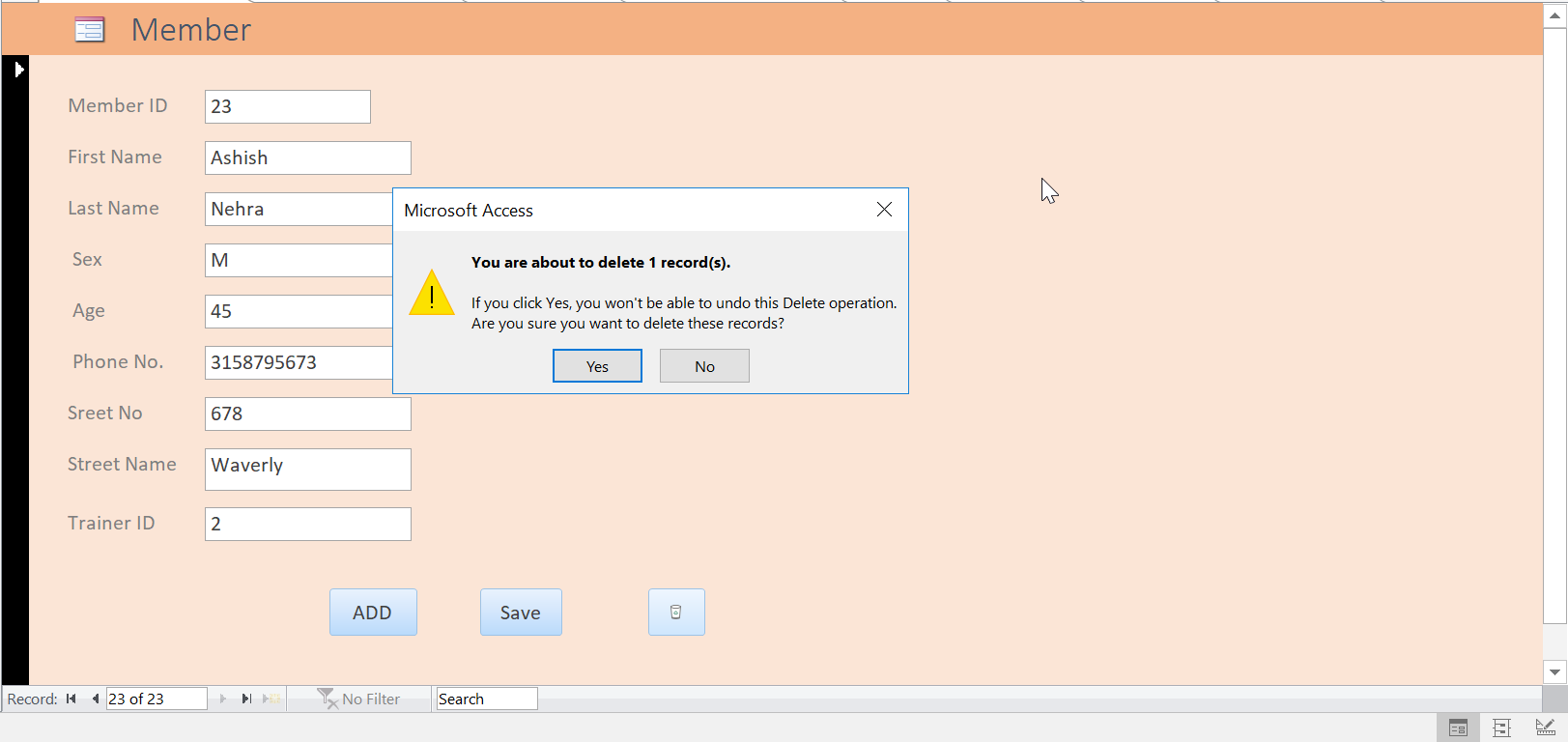


* Check if the added Member has reflected in the SQL database

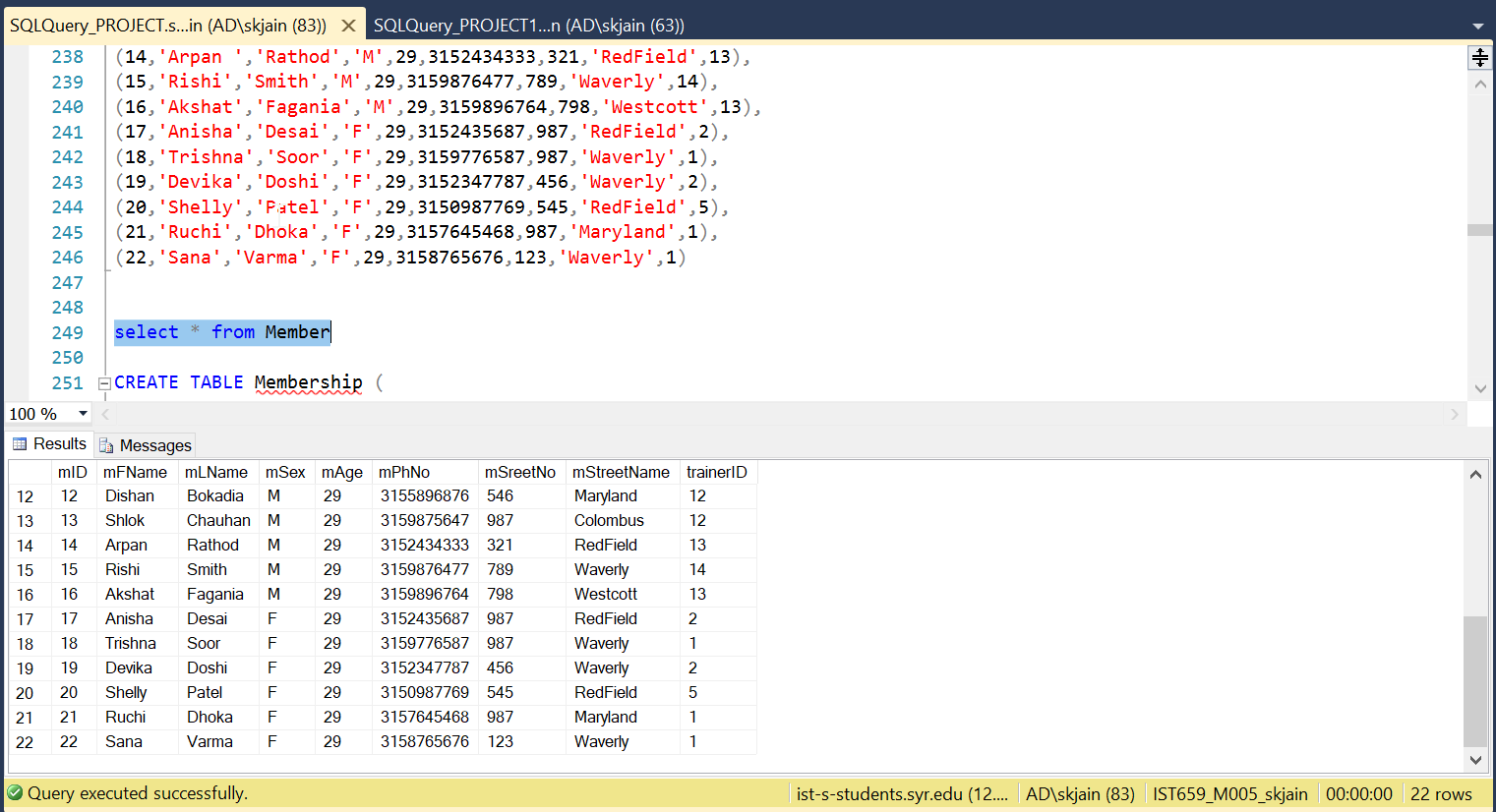


**Yes** , it has been reflected in the database.

* You can also **delete** the Member from the database



* Then check if the Member has been deleted successfully from the SQL database.



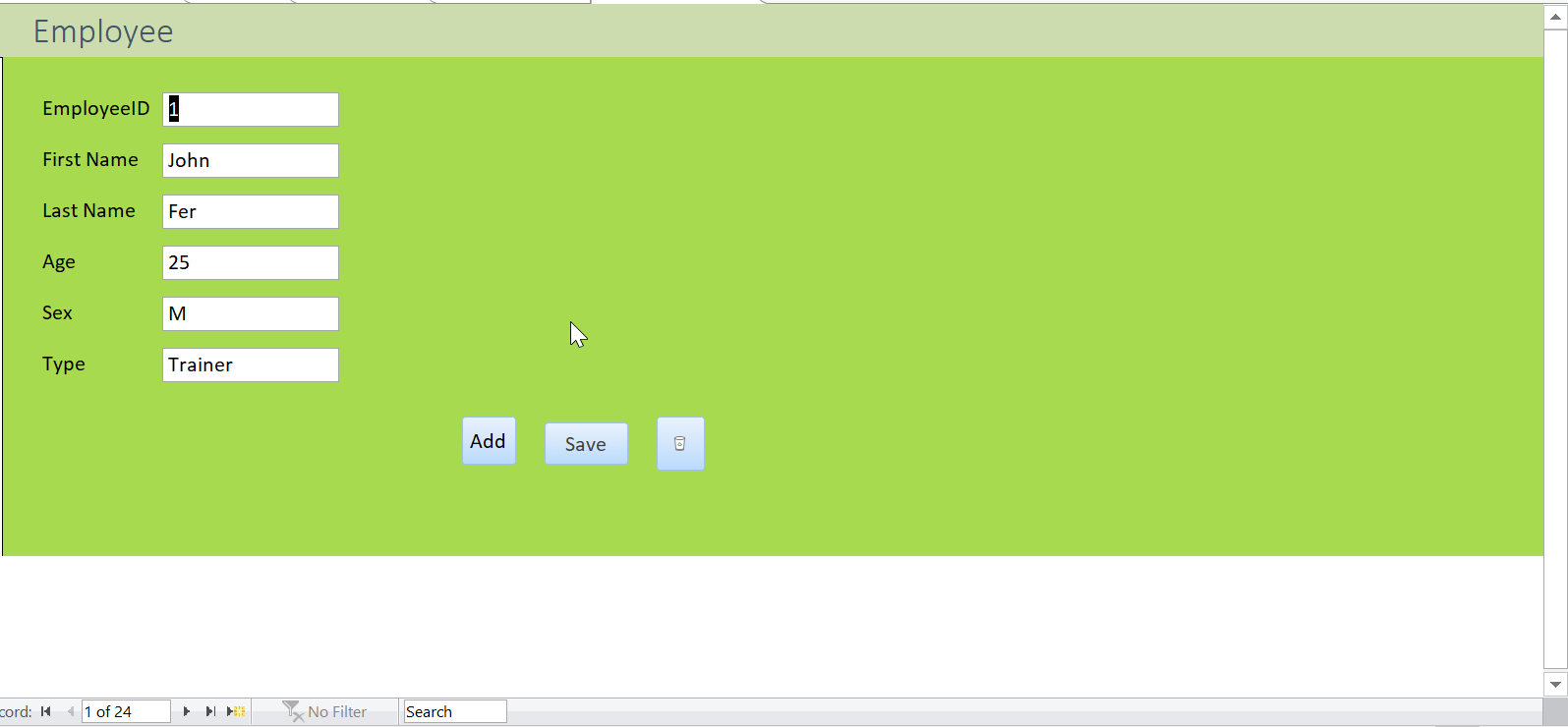
Yes , it has been deleted from the database.

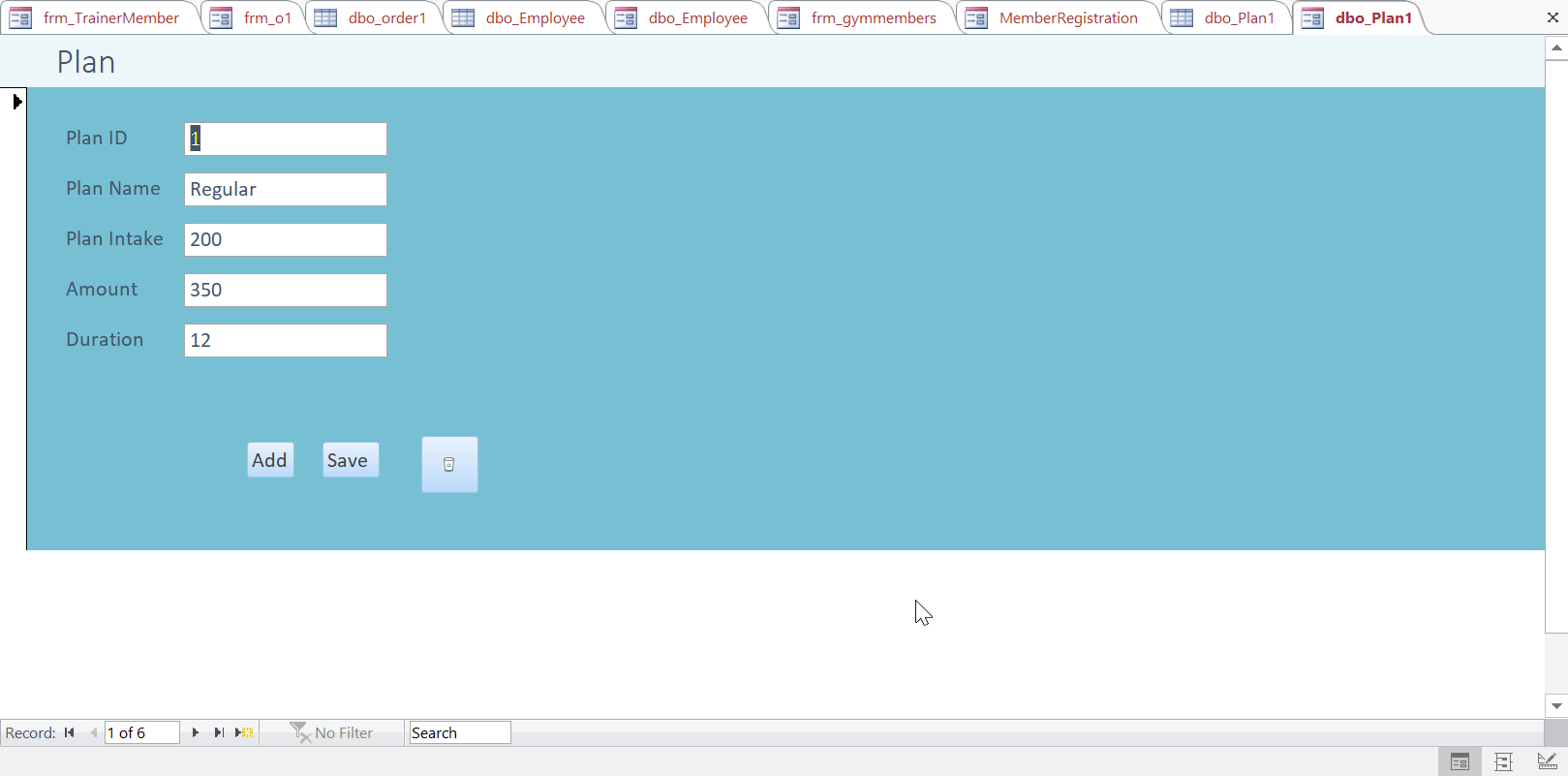
* On navigating further, you can explore i.e. View, Add, Delete any of the three tabs:

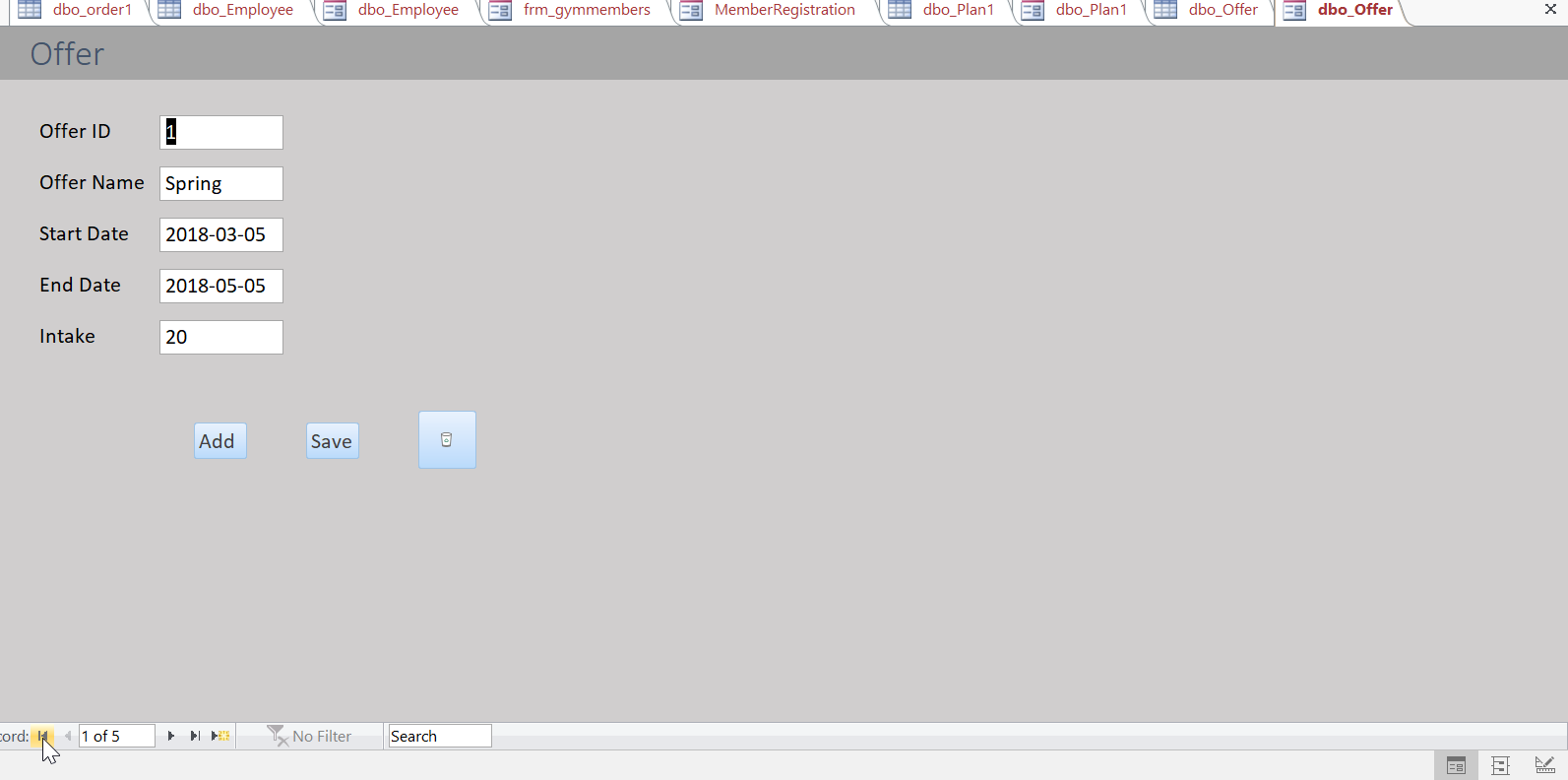
1)Emloyee

2)Plan

3)Offers





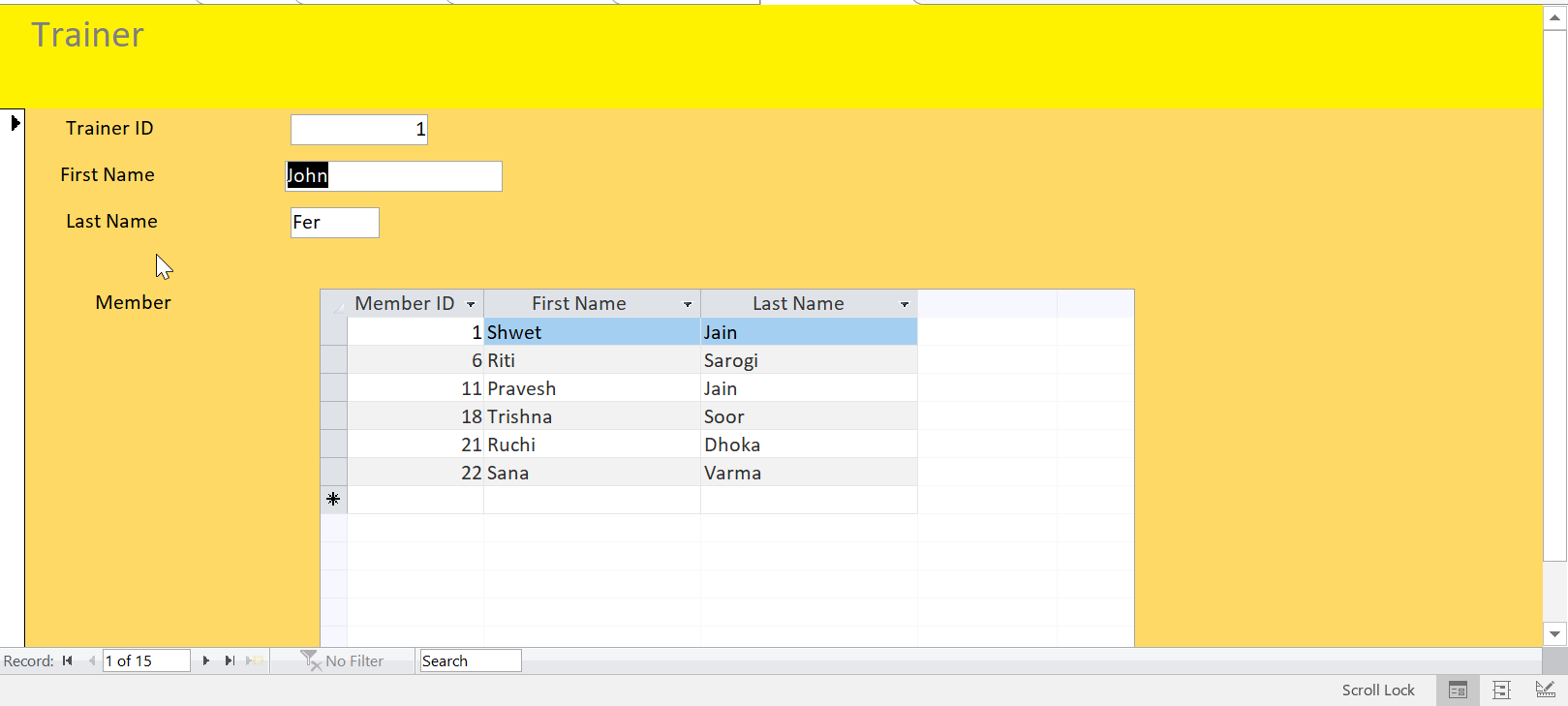


* On navigating further, there is a trainer tab which gives information about which trainer is training which member in the gym

As you can see, Trainer with trainerID=1 is training 6 members i.e.

(Attached are the screenshots of the 6 members trained by TrainerID=1)

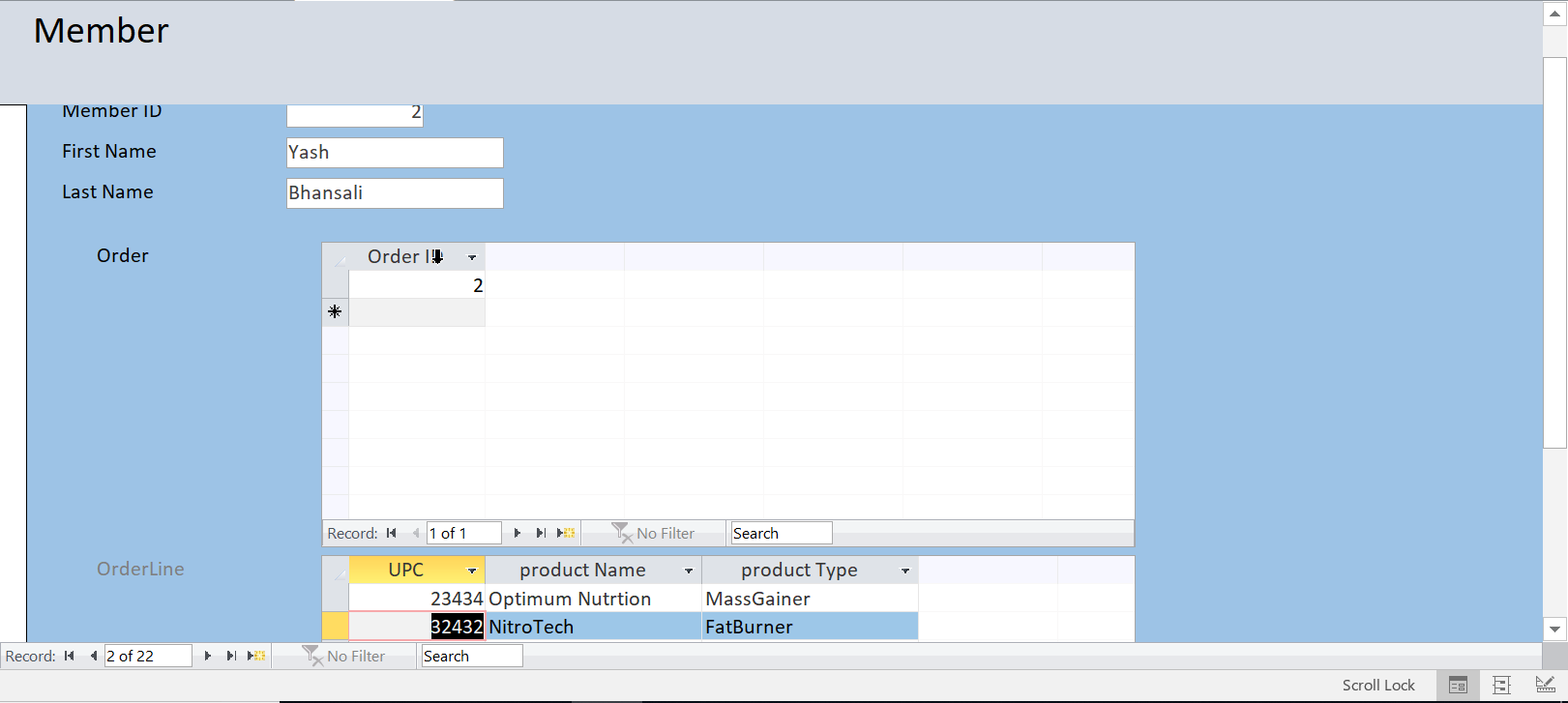
1. **Shwet Jain**
2. **Riti Sarogi**
3. **Pravesh Jain**
4. **Trishna Soor**
5. **Ruchi Dhoka**
6. **Sana Varma**



* On navigating further, there is an OrderTab tab which gives information about how many orders has been placed by each member along with the product.

As we can see, member ID= 2 is associated OrderID=2 which has 2 products in it :

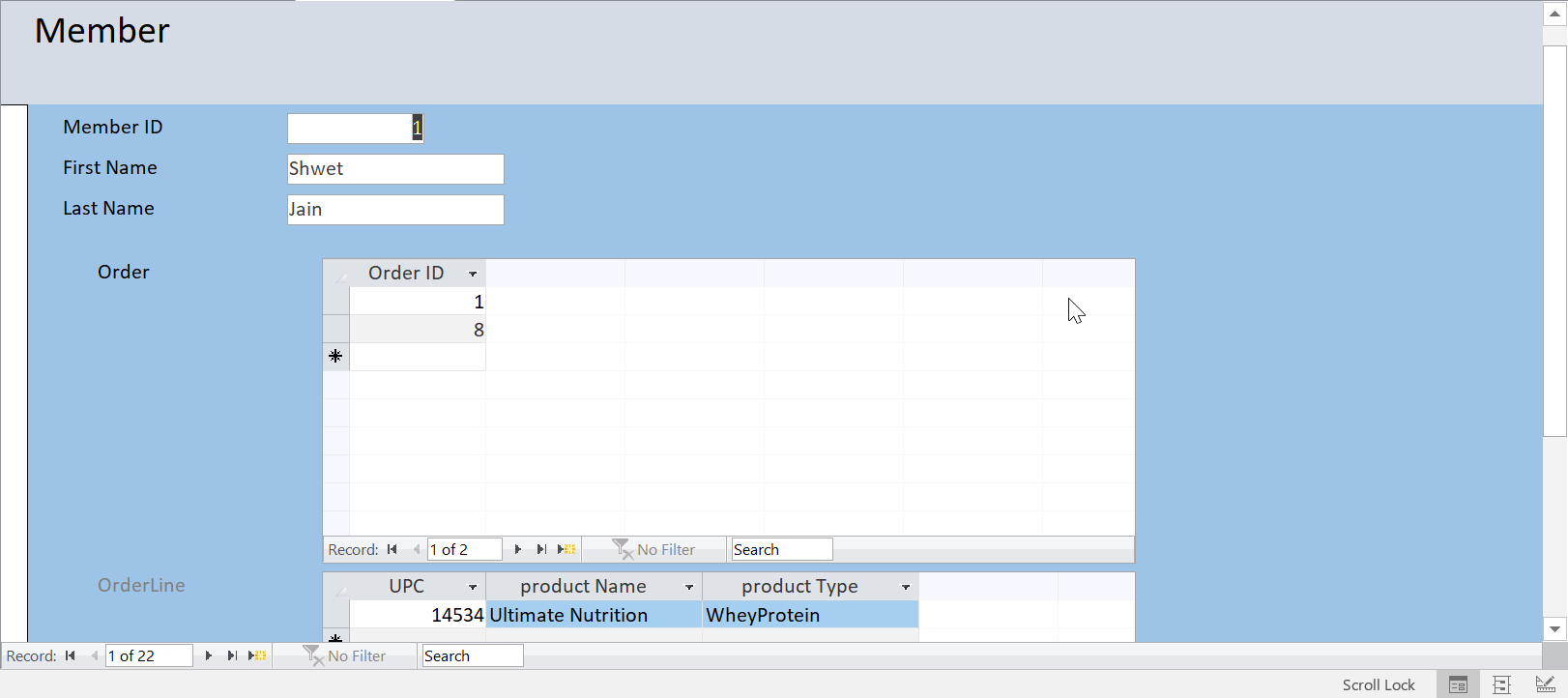
1. **Optimum Nutrition**
2. **NitroTech**

****

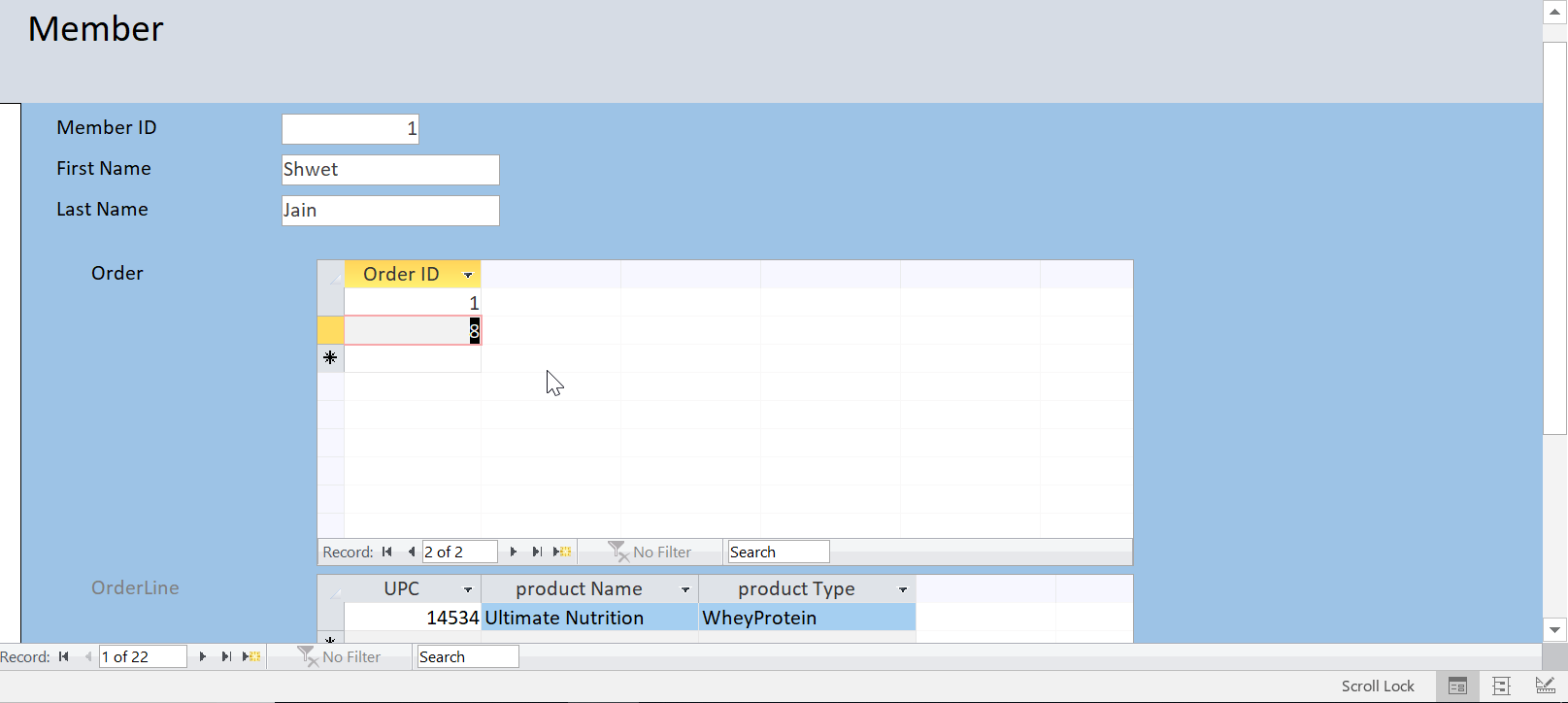
Similarly,

As you can see, Member with member ID=1 has placed 2 orders with OrderID= 1 AND OrderID=8

OrderID=1 has 1 product i.e. Whey Protein (14534)



OrderID=8 also has one product i.e. Ultimate Nutrition Whey Protein



* On navigating further, there is a MemberPlan tab which gives information about Plans each member has enrolled for.

As we can see,

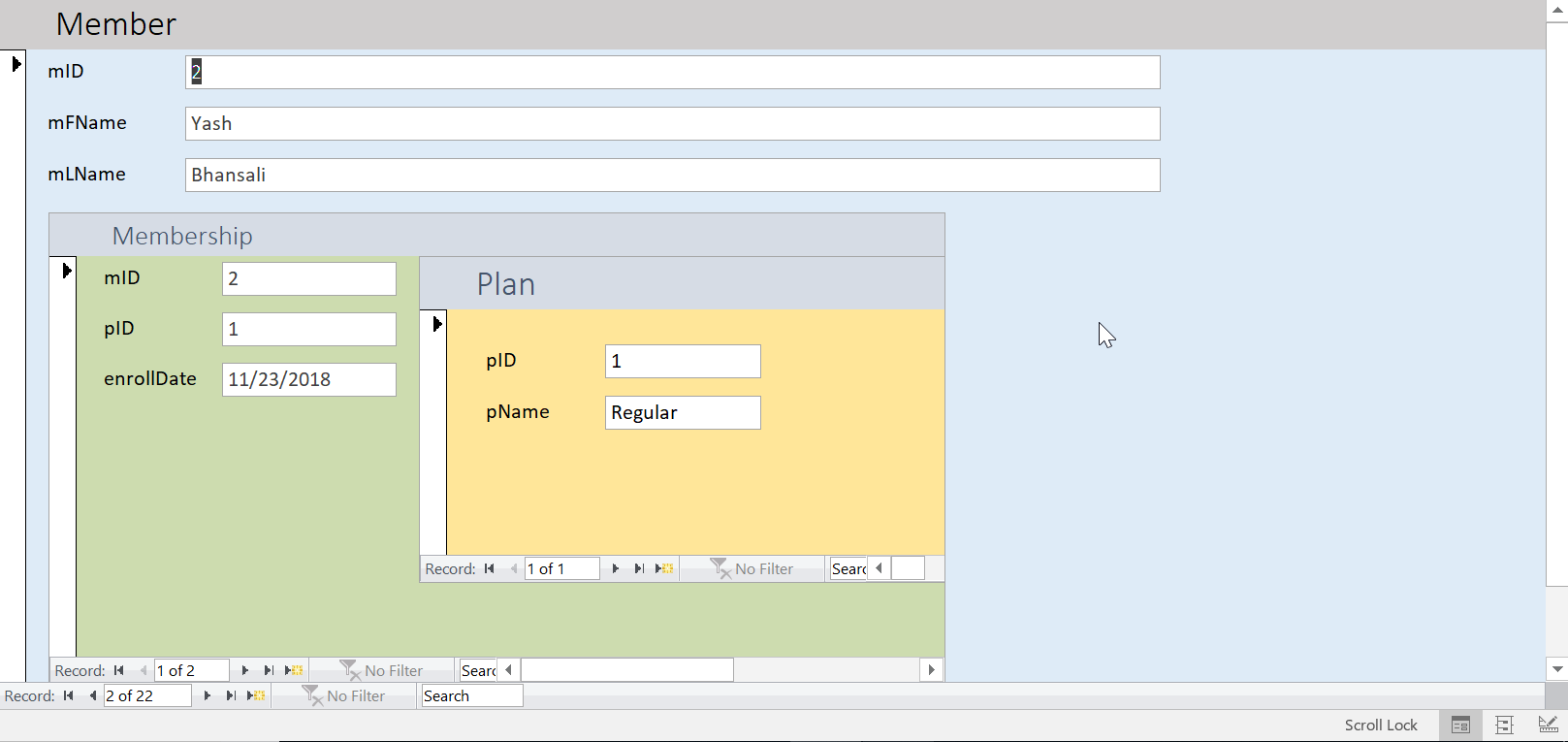
Member ID=2

First Name: Yash

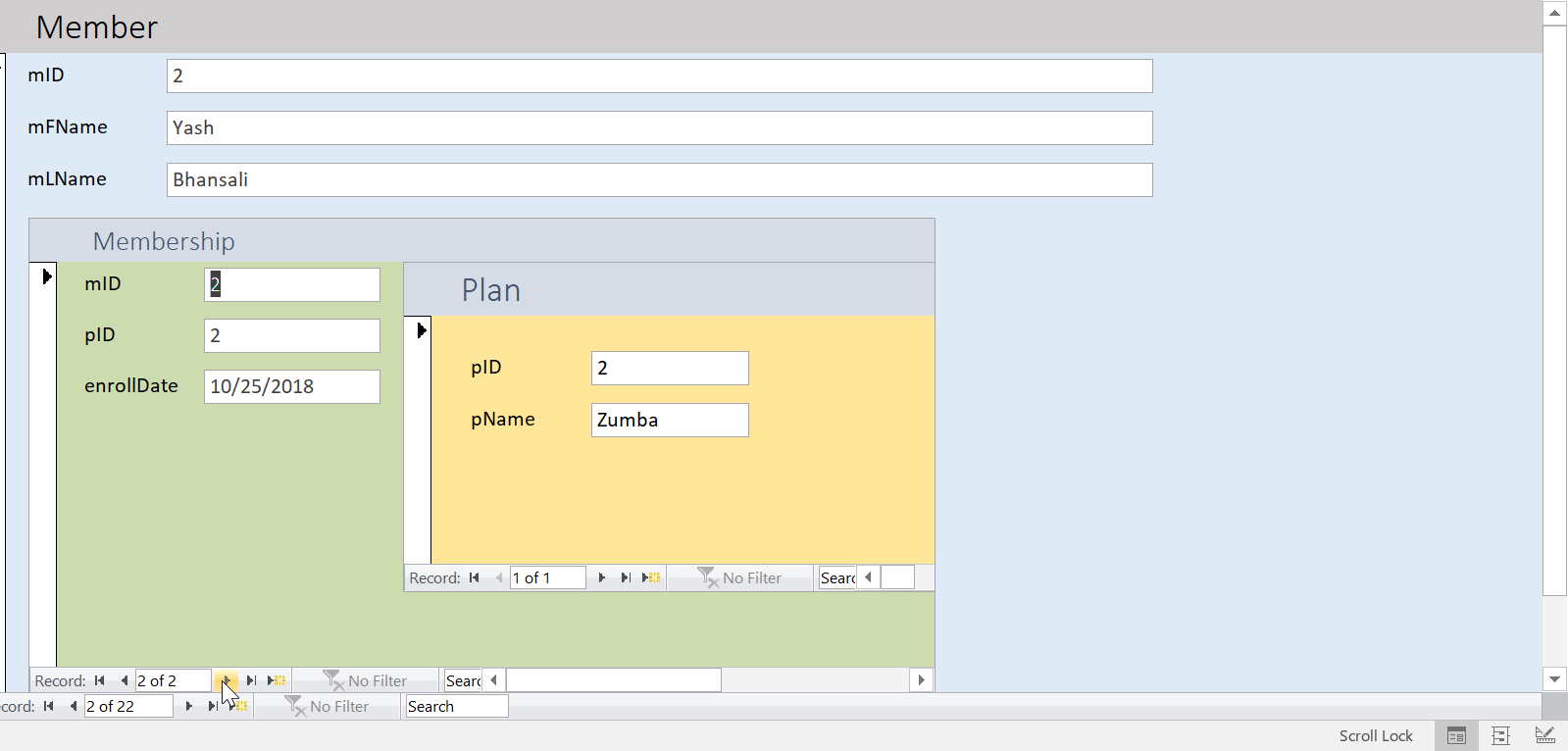
Last Name: Bhansali has enrolled for 2 plans

1. **Regular**

**(Screenshot below)**



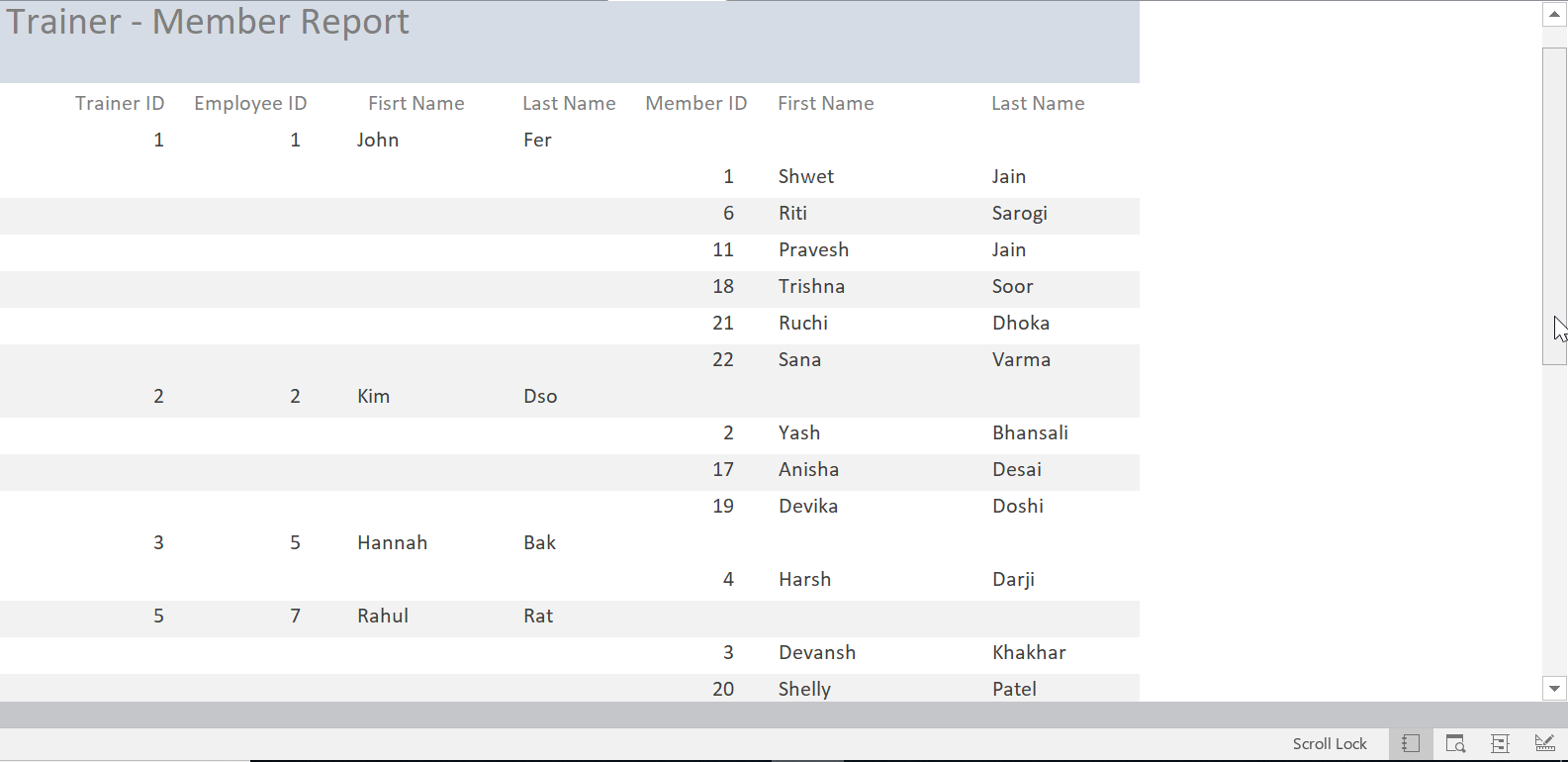
1. **Zumba**



REPORT

1. PersonalTrainer-Member Report

[Members trained by each Personal Trainer]





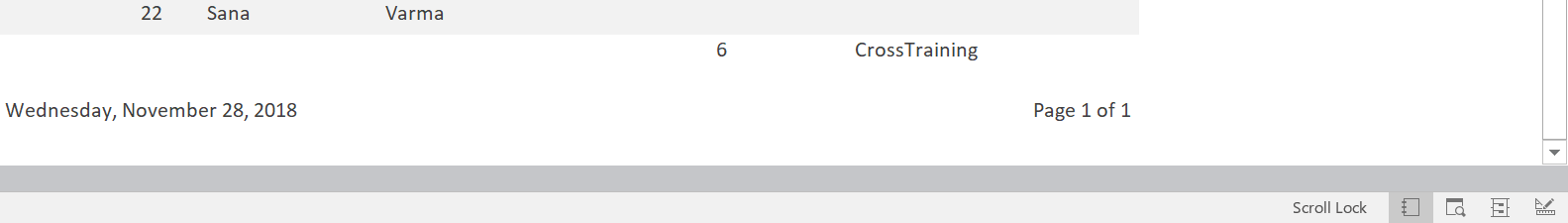
1. Member-Plan Report

[Plans enrolled by each Member]









1. Member-Order-Product Report

[Products in each and every Order/Orders placed by Members]

