New Information Management System for Rise-and-Grind Fitness Center

Team A+

December 6, 2022

Amanda Barnett, Julianna Budriss, Ryan Curling,

Nik Henegar, and Wesley Weeks

Table of Contents

Executive Summary

We wanted to create a system that would be able to take influxes of business data being collected and properly store/manage that data. This information management system would be able to correctly sort any transactional data from customers as well as data we collected on employees and other operational data. The goal of this system is to increase efficiency across the company and combine it in one place making information more accessible and easier to understand.

The new information system will have many capabilities to help keep track of key metrics of the business and its performance. This system will help you keep track of the revenue from sales, or memberships. You will be able to track enrollment of the courses, any course information and how much money is earned from the course. This system is also capable of keeping track of information about classes such as which classroom, equipment, and instructors are a part of the class. Information about employees, such as basic information about them as well as instructors' courses and certifications.

The store will be able to track and store all transaction data as well as inventory levels that alert you when inventory gets to the reorder point. The system also allows for maintenance information, and all equipment information to be stored, such as when equipment needs maintenance and the manufacturer information. It also allows you to see the details of the work order as well.

The system is also capable of storing a large amount of customer information. It will keep track of what customers buy, their enrollment in courses and instructors, as well as their level of membership.

We have 8 assumptions that affect the design of our system. Our first assumption states that 'Equipment Type' will only be recorded if that piece of equipment is in the facility. This helps eliminate clutter from our database, because old equipment or equipment that hasn't arrived yet will not be stored. Our second assumption allows us to create classes, before scheduling a class date. This assumption was necessary because a class date can not be set before the class is even created. Our third assumption is that a 'Clerk' may not have made a transaction. This could be because the employee is brand new, so to properly recognize the employees existence we needed to add this assumption. Our fourth assumption stated that only one 'Clerk' can make a transaction, since it is not logical for two 'Clerks' to make the same transaction. The fifth assumption of our database is that not all classes may need equipment so we need to make the assumption here in order to recognize the classes that do not use equipment so this field is not required. Our sixth assumption is to only recognize current

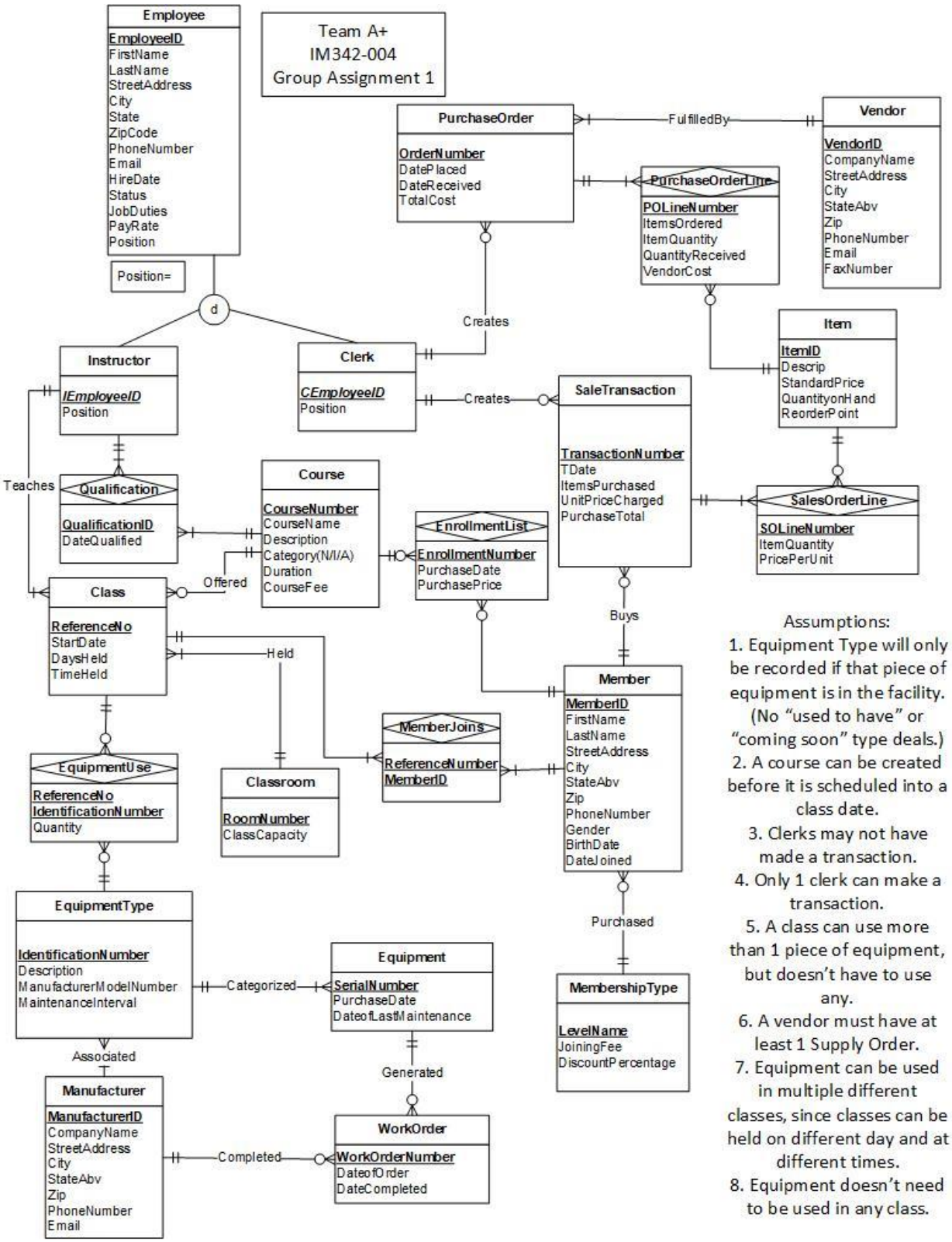
vendors. We would need to require a vendor that has fulfilled at least one supply order, otherwise we would have vendors stored that may have never completed an order for us which could lead to unnecessarily using storage for those vendors. For the seventh assumption, we stated that since classes are held at different times and days, the same piece of equipment can be used for multiple classes. If we did not make this assumption our database would assign one class to one piece of equipment, and would not be accurately displaying which classes use which piece of equipment. Finally, for our eighth assumption, we assumed that equipment doesn't need to be used in any class. Although this is unlikely, an assumption was needed in order to accurately record data if no classes used equipment.

With this overview you will also receive other important information including:

- **Conceptual Data Model**
    - The conceptual data model is a visual depiction of how information is related to each other.
- **Logical Data Model**
    - This will be a written version of tables as well as the fields that are related to them. Foreign Keys are also identified by their formatting.
- **Data Dictionary**
    - This is a description of each field within the system. This also includes information about the data type, data length and if it has any default values.
- **Sample Report Designs**
    - These are examples of some reports that you may be able to make through the system to use in operation.

Phase II of the project will include the implemented system and software application prototype that you will be able use in order to form reports, or view data. It will also include a user manual including examples of how to use it along with sample data to experiment with.

ERD


Employee
EmployeeID
FirstName
LastName
StreetAddress
City
State
ZipCode
PhoneNumber
Email
HireDate
Status
JobDuties
PayRate
Position
Position=
d
Team A+
IM342-004
Group Assignment 1
PurchaseOrder
OrderNumber
DatePlaced
DateReceived
TotalCost
FulfilledBy
Vendor
VendorID
CompanyName
StreetAddress
City
StateAbv
Zip
PhoneNumber
Email
FaxNumber
PurchaseOrderLine
POLineNumber
ItemsOrdered
ItemQuantity
QuantityReceived
VendorCost
Creates
Item
ItemID
Descrip
StandardPrice
QuantityonHand
ReorderPoint
Instructor
IEmployeeID
Position
Clerk
CEmployeeID
Position
Creates
SaleTransaction
TransactionNumber
TDate
ItemsPurchased
UnitPriceCharged
PurchaseTotal
Teaches
Qualification
QualificationID
DateQualified
Course
CourseNumber
CourseName
Description
Category(N/I/A)
Duration
CourseFee
EnrollmentList
EnrollmentNumber
PurchaseDate
PurchasePrice
SalesOrderLine
SOLineNumber
ItemQuantity
PricePerUnit
Offered
Class
ReferenceNo
StartDate
DaysHeld
TimeHeld
Held
Buys
Member
MemberID
FirstName
LastName
StreetAddress
City
StateAbv
Zip
PhoneNumber
Gender
BirthDate
DateJoined
MemberJoins
ReferenceNumber
MemberID
EquipmentUse
ReferenceNo
IdentificationNumber
Quantity
Classroom
RoomNumber
ClassCapacity
EquipmentType
IdentificationNumber
Description
ManufacturerModelNumber
MaintenanceInterval
Categorized
Equipment
SerialNumber
PurchaseDate
DateofLastMaintenance
Purchased
MembershipType
LevelName
JoiningFee
DiscountPercentage
Associated
Generated
Manufacturer
ManufacturerID
CompanyName
StreetAddress
City
StateAbv
Zip
PhoneNumber
Email
Completed
WorkOrder
WorkOrderNumber
DateofOrder
DateCompleted
Assumptions:
1. Equipment Type will only be recorded if that piece of equipment is in the facility. (No “used to have” or “coming soon” type deals.)
2. A course can be created before it is scheduled into a class date.
3. Clerks may not have made a transaction.
4. Only 1 clerk can make a transaction.
5. A class can use more than 1 piece of equipment, but doesn’t have to use any.
6. A vendor must have at least 1 Supply Order.
7. Equipment can be used in multiple different classes, since classes can be held on different day and at different times.
8. Equipment doesn’t need to be used in any class.

Logical Model

Employee (**EmployeeID**, FirstName, LastName, StreetAddress, City, State, ZipCode, PhoneNumber, Email, HireDate, Status, JobDuties, PayRate, **Position**)

Instructor (***IEmployeeID*, *Position***)

Clerk (***CEmployeeID*, *Position***)

Qualification (**QualificationID**, DateQualified, ***IEmployeeID, CourseNumber***)

Course (**CourseNumber**, CourseName, Description, Category, Duration, CourseFee)

EnrollmentList (**EnrollmentNumber**, PuchaseDate, PurchasePrice, ***CourseNumber, MemberID***)

Member (**MemberID**, FirstName, LastName, StreetAddress, City, StateAbv, Zip, PhoneNumber, Gender, BirthDate, DateJoined, ***LevelName***)

MembershipType (**LevelName**, JoiningFee, DiscountPercent)

MemberJoins (***MemberID, ReferenceNo.***)

Class (**ReferenceNo.**, StartDate, DaysHeld, TimeHeld, ***IEmployeeID***, ***RoomNumber, CourseNumber***)

EquipmentUse (***ReferenceNo***, ***IdentificationNumber***, Quantity)

EquipmentType (**IdentificationNumber**, Description, ManufacturerModelNumber, MaintenanceInterval, ***ManufacturerID***)

Classroom (**RoomNumber**, ClassCapacity)

WorkOrder (**WorkOrderNumber**, DateofOrder, DateCompleted, ***ManufacturerID***, ***SerialNumber***)

Manufacturer (**ManufacturerID**, CompanyName, StreetAddress, City, StateAbv, Zip, PhoneNumber, Email)

Equipment (**SerialNumber**, PurchaseDate, DateOfLastMaintance, ***IdentificationNumber***)

Vendor (**VendorID**,CompanyName, StreetAddress, City, StateAbv, Zip, PhoneNumber, Email, FaxNumber)

PurchaseOrder (**OrderNumber**, DatePlaced, DateReceived, TotalCost, ***VendorID, CEmployeeID***)

PurchaseOrderLine (**POLineNumber**, ItemsOrdered, ItemQuantity, QuantityReceived, VendorCost, ***CEmployeeID, ItemID***)

Item (**ItemID**, Descrip, StandardPrice, QuantityonHand, ReorderPoint)

SaleOrderLine (**SOLineNumber**, ItemQty, PricePerUnit, ***ItemID, TransactionNumber***)

SaleTransaction (**TransactionNumber**, TDate, ItemsPurchased, UnitPriceCharged, PurchaseTotal, ***CEmployeeID, MemberID***)

Create Statements

```
CREATE TABLE Employee (
EmployeeID  INT     NOT NULL    IDENTITY(1,1)       PRIMARY KEY,
FirstName     VARCHAR(25),
LastName     VARCHAR(25),
StreetAddress                 VARCHAR(50),
City     VARCHAR(30),
State   CHAR(2),
ZipCode       CHAR(5),
PhoneNumber                   CHAR(10),
Email  VARCHAR(50),
HireDate       DATE,
Status          VARCHAR(10)       CHECK(Status IN ('Active', 'Inactive', 'Leave',
'Terminated')),
JobDuties     VARCHAR(50),
PayRate       MONEY       CHECK(PayRate > 0),
Position       VARCHAR(15)       NOT NULL
);

CREATE TABLE Instructor (
IEmployeeID          INT              NOT NULL,
Position       VARCHAR(15)       NOT NULL    CHECK(Position = 'Instructor'),
        CONSTRAINT InstructorPK PRIMARY KEY(IEmployeeID),
        CONSTRAINT       IEmployeeFK          FOREIGN KEY (IEmployeeID)
        REFERENCES Employee(EmployeeID)
        ON UPDATE CASCADE ON DELETE NO ACTION
);

CREATE TABLE Clerk (
CEmployeeID                   INT              NOT NULL,
Position       VARCHAR(15)       NOT NULL    CHECK(Position = 'Clerk'),
        CONSTRAINT ClerkPK PRIMARY KEY(CEmployeeID),
        CONSTRAINT       CEmployeeFK        FOREIGN KEY (CEmployeeID)
        REFERENCES Employee(EmployeeID)
        ON UPDATE CASCADE ON DELETE NO ACTION
);

CREATE TABLE MembershipType (
LevelName   VARCHAR(8)        PRIMARY KEY       NOT NULL    CHECK(LevelName
IN ('Platinum', 'Gold', 'Silver', 'Bronze')),
```

```
JoiningFee    MONEY       CHECK(JoiningFee > 0),
DiscountPercent     DECIMAL(1,1)              CHECK(DiscountPercent >= 0.0 AND
DiscountPercent <= 1.0)
);

CREATE TABLE Course (
CourseNumber       CHAR(5)             NOT NULL    PRIMARY KEY,
CourseName         VARCHAR(15),
Description        VARCHAR(50),
Category           VARCHAR(15)              CHECK(Category IN ('Novice',
'Intermediate', 'Advanced')),
Duration       Decimal(6,2)         CHECK(Duration > 0),
CourseFee    MONEY
);

CREATE TABLE Qualification (
QualificationID              CHAR(5)      NOT NULL    PRIMARY KEY,
DateQualified DATE,
IEmployeeID          INT             NOT NULL,
        CONSTRAINT IEmployeeIDFK FOREIGN KEY(IEmployeeID)
        REFERENCES Instructor(IEmployeeID)
        ON UPDATE CASCADE ON DELETE NO ACTION,
CourseNumber       CHAR(5)              NOT NULL,
        CONSTRAINT CourseNumFK FOREIGN KEY(CourseNumber)
        REFERENCES Course(CourseNumber)
        ON UPDATE CASCADE ON DELETE NO ACTION
);

CREATE TABLE Member (
MemberID    INT     NOT  NULL   IDENTITY(1001,1)     PRIMARY KEY ,
FirstName       VARCHAR (25),
LastName        VARCHAR (25),
StreetAddress   VARCHAR(50),
City    VARCHAR (30),
StateAbv      CHAR (2),
Zip             CHAR(5),
PhoneNumber  CHAR(10),
Gender        VARCHAR(20)              CHECK(Gender IN ('Male', 'Female', 'Neither',
'Prefer Not To Say')),
BirthDate      DATE,
```

```
DateJoined    DATE,
LevelName    VARCHAR(8)        NOT NULL    CHECK(LevelName IN ('Platinum',
'Gold', 'Silver', 'Bronze')),
        CONSTRAINT LevelNameFK FOREIGN KEY(LevelName)
        REFERENCES MembershipType(LevelName)
        ON UPDATE CASCADE ON DELETE NO ACTION
);

CREATE TABLE EnrollmentList (
EnrollmentNumber          INT     NOT NULL    IDENTITY(10000001,1)     PRIMARY
KEY,
PurchaseDate              DATE,
PurchasePrice             MONEY,
CourseNumber      CHAR(5)             NOT NULL,
        CONSTRAINT      CourseNumberFK    FOREIGN KEY(CourseNumber)
        REFERENCES      Course(CourseNumber)
        ON UPDATE CASCADE ON DELETE NO ACTION,
MemberID    INT     NOT  NULL,
        CONSTRAINT      MemberIDFK        FOREIGN KEY(MemberID)
        REFERENCES      Member(MemberID)
        ON UPDATE CASCADE ON DELETE NO ACTION
);

CREATE TABLE Classroom (
RoomNumber        CHAR(2)             NOT NULL    PRIMARY KEY,
ClassCapacity     INT             CHECK(ClassCapacity > 0)
);

CREATE TABLE Class (
ReferenceNo CHAR(5)     NOT NULL    PRIMARY KEY,
StartDate     DATE,
DaysHeld      VARCHAR(7),
TimeHeld      TIME,
IEmployeeID           INT             NOT NULL,
        CONSTRAINT ClassInstructorFK FOREIGN KEY(IEmployeeID)
        REFERENCES Instructor(IEmployeeID)
        ON UPDATE CASCADE ON DELETE NO ACTION,
RoomNumber        CHAR(2)             NOT NULL,
        CONSTRAINT RoomNumberFK FOREIGN KEY(RoomNumber)
        REFERENCES Classroom(RoomNumber)
```

```
        ON UPDATE CASCADE ON DELETE NO ACTION,
CourseNumber        CHAR(5)        NOT NULL,
        CONSTRAINT CourseNumbFK    FOREIGN KEY(CourseNumber)
        REFERENCES Course(CourseNumber)
        ON UPDATE CASCADE ON DELETE NO ACTION
);

CREATE TABLE MemberJoins (
MemberID     INT     NOT NULL,
ReferenceNo CHAR(5)      NOT NULL,
        CONSTRAINT        MemberPK    PRIMARY KEY(MemberID, ReferenceNo),
        CONSTRAINT        ReferenceNoFK              FOREIGN KEY(ReferenceNo)
        REFERENCES        Class(ReferenceNo)
        ON UPDATE CASCADE ON DELETE NO ACTION,
        CONSTRAINT        MemberFK             FOREIGN KEY(MemberID)
        REFERENCES        Member(MemberID)
        ON UPDATE CASCADE ON DELETE NO ACTION
);

CREATE TABLE Manufacturer (
ManufacturerID     INT   NOT NULL    IDENTITY(201,1)    PRIMARY KEY,
CompanyName     VARCHAR(40),
StreetAddress   VARCHAR(50),
City     VARCHAR (30),
StateAbv       CHAR (2),
Zip              CHAR(5),
PhoneNumber  CHAR(10),
Email  VARCHAR(50)
);

CREATE TABLE EquipmentType (
IdentificationNumber          CHAR(3)       NOT NULL    PRIMARY KEY,
Description    VARCHAR(50),
ManufacturerModelNumber VARCHAR(15),
MaintenanceInterval          VARCHAR(10),
ManufacturerID     INT   NOT NULL,
        CONSTRAINT EquipTypeManufFK FOREIGN KEY(ManufacturerID)
        REFERENCES Manufacturer(ManufacturerID)
        ON UPDATE CASCADE ON DELETE NO ACTION,
);
```

```
CREATE TABLE EquipmentUse (
ReferenceNo CHAR(5)      NOT NULL,
IdentificationNumber        CHAR(3)      NOT NULL,
Quantity       INT,
        CONSTRAINT UsePK       PRIMARY KEY(ReferenceNo, IdentificationNumber),
        CONSTRAINT IdentificationNumberFK FOREIGN KEY(IdentificationNumber)
        REFERENCES EquipmentType(IdentificationNumber)
        ON UPDATE CASCADE ON DELETE NO ACTION,
        CONSTRAINT CReferenceNoFK FOREIGN KEY(ReferenceNo)
        REFERENCES Class(ReferenceNo)
        ON UPDATE CASCADE ON DELETE NO ACTION
 );

CREATE TABLE Equipment (
SerialNumber        VARCHAR(25)       NOT NULL            PRIMARY KEY,
PurchaseDate        DATE,
DateOfLastMaintance                DATE,
IdentificationNumber        CHAR(3)      NOT NULL,
        CONSTRAINT EquipmentIdentificationFK FOREIGN KEY(IdentificationNumber)
        REFERENCES EquipmentType(IdentificationNumber)
        ON UPDATE CASCADE ON DELETE NO ACTION
);

CREATE TABLE WorkOrder (
WorkOrderNumber          INT   NOT NULL   IDENTITY(0000001,1)   PRIMARY KEY ,
DateofOrder                DATE,
SerialNumber        VARCHAR(25)       NOT NULL,
ManufacturerID     INT   NOT NULL,
DateCompleted              DATE,
        CONSTRAINT ManufacturerIDFK FOREIGN KEY(ManufacturerID)
        REFERENCES Manufacturer(ManufacturerID)
        ON UPDATE CASCADE ON DELETE NO ACTION,
        CONSTRAINT SerialNumFK FOREIGN KEY(SerialNumber)
        REFERENCES Equipment(SerialNumber)
        ON UPDATE NO ACTION ON DELETE NO ACTION
);

CREATE TABLE Vendor (
VendorID      INT            NOT NULL   IDENTITY(1,1)       PRIMARY KEY,
```

```
CompanyName     VARCHAR(40),
StreetAddress   VARCHAR(50),
City    VARCHAR (30),
StateAbv        CHAR (2),
Zip             CHAR(5),
PhoneNumber  CHAR(10),
Email  VARCHAR(50),
FaxNumber   CHAR(10),
);

CREATE TABLE PurchaseOrder (
OrderNumber             INT             NOT NULL    IDENTITY(1,1)       PRIMARY
KEY,
DatePlaced          DATE,
DateReceived            DATE,
TotalCost           MONEY,
VendorID    INT             NOT NULL,
CEmployeeID             INT             NOT NULL,
        CONSTRAINT VendorIDFK FOREIGN KEY(VendorID)
        REFERENCES Vendor(VendorID)
        ON UPDATE CASCADE ON DELETE NO ACTION,
        CONSTRAINT CEmpIDFK FOREIGN KEY(CEmployeeID)
        REFERENCES Clerk(CEmployeeID)
        ON UPDATE CASCADE ON DELETE NO ACTION
);

CREATE TABLE Item (
ItemIDINT           NOT NULL    IDENTITY(1,1)       PRIMARY KEY,
Descrip         VARCHAR(50),
StandardPrice       MONEY,
QuantityonHand      INT,
ReorderPointINT
);

CREATE TABLE PurchaseOrderLine (
POLineNumber    INT     NOT NULL        IDENTITY(1,1)   PRIMARY KEY,
ItemsOrdered        VARCHAR(25),
ItemQuantity        INT,
QuantityReceived  INT,
VendorCost   MONEY,
```

```
CEmployeeID             INT          NOT NULL,
ItemIDINT          NOT NULL,
        CONSTRAINT CEmployeeIDFK FOREIGN KEY(CEmployeeID)
        REFERENCES Clerk(CEmployeeID)
        ON UPDATE CASCADE ON DELETE NO ACTION,
        CONSTRAINT ItemIDFK FOREIGN KEY(ItemID)
        REFERENCES Item(ItemID)
        ON UPDATE CASCADE ON DELETE NO ACTION
);

CREATE TABLE SaleTransaction (
TransactionNumber INT          NOT NULL   IDENTITY(1,1)      PRIMARY KEY,
TDate DATE,
ItemsPurchased     VARCHAR(10),
UnitPriceCharged   MONEY,
PurchaseTotal           MONEY,
CEmployeeID             INT          NOT NULL,
MemberID    INT          NOT NULL,
        CONSTRAINT CEmployIDFK FOREIGN KEY(CEmployeeID)
        REFERENCES Clerk(CEmployeeID)
        ON UPDATE CASCADE ON DELETE NO ACTION,
        CONSTRAINT      MembIDFK FOREIGN KEY(MemberID)
        REFERENCES Member(MemberID)
        ON UPDATE CASCADE ON DELETE NO ACTION
);

CREATE TABLE SaleOrderLine (
SOLineNumber      INT    NOT NULL   IDENTITY(1,1)      PRIMARY KEY,
ItemQty           INT,
PricePerUnit      MONEY,
ItemID             INT   NOT NULL,
TransactionNumber  INT   NOT NULL,
        CONSTRAINT ItemFK FOREIGN KEY(ItemID)
        REFERENCES Item(ItemID)
        ON UPDATE CASCADE ON DELETE NO ACTION,
        CONSTRAINT TransNumFK FOREIGN KEY(TransactionNumber)
        REFERENCES SaleTransaction(TransactionNumber)
        ON UPDATE CASCADE ON DELETE NO ACTION
);
```

Insert Statements

```
INSERT INTO Employee(FirstName, LastName, StreetAddress, City, State, ZipCode,
PhoneNumber, Email, HireDate, Status, JobDuties, PayRate, Position)
VALUES('Nyle', 'Waller', '271 County Ave.', 'Jamaica', 'NY', '11432', '2029182132',
'uriah28@yahoo.com',  '8/17/2020',  'Active', 'Instruct Courses', 13, 'Instuctor');
INSERT INTO Employee (FirstName, LastName, StreetAddress, City, State, ZipCode,
PhoneNumber, Email, HireDate, Status, JobDuties, PayRate, Position)
VALUES('Cheyanne', 'Patrick', '8007 Virginia St.', 'Bay Shore', 'NY', '11706',
'2025550170', 'enrique5@hotmail.com',  '10/16/2020', 'Inactive', 'Inventory, Cash
Register, Store, Sell Memberships', 14, 'Clerk');
INSERT INTO Employee(FirstName, LastName, StreetAddress, City, State, ZipCode,
PhoneNumber, Email, HireDate, Status, JobDuties, PayRate, Position)
VALUES('Pollyanna', 'Mcgregor', '9437 North Nicolls St.', 'Bronx', 'NY', '10466',
'2025550147', 'herminio_ohara@gmail.com',  '10/16/2020', 'Active', 'Instruct Courses
in healthy eating', 13, 'Instructor');
INSERT INTO Employee (FirstName, LastName, StreetAddress, City, State, ZipCode,
PhoneNumber, Email, HireDate, Status, JobDuties, PayRate, Position)
VALUES('Areeba', 'Bishop', '586 Blackburn Circle', 'Hempstead', 'NY',  '11550',
'2025550146', 'lacey.jones@hotmail.com',  '11/3/2020',  'Active', 'Inventory, Cash
Register, Store, Sell Memberships', 11, 'Clerk');
INSERT INTO Employee (FirstName, LastName, StreetAddress, City, State, ZipCode,
PhoneNumber, Email, HireDate, Status, JobDuties, PayRate, Position)
VALUES('Said', 'Valenzuela', '83 Wakehurst Street', 'Bronx', 'NY', '10456',
        '2025550169',  'florian_koss66@gmail.com',  '12/25/2020', 'Active',
'Inventory, Cash Register, Store, Sell Memberships', 12, 'Clerk');
INSERT INTO Employee (FirstName, LastName, StreetAddress, City, State, ZipCode,
PhoneNumber, Email, HireDate, Status, JobDuties, PayRate, Position)
VALUES('Walid', 'Pena', '8017 Elmwood Lane', 'Brooklyn', 'NY', '11234',
        '2025550171',  'stephania.heller89@hotmail.com',  '1/21/2021',
        'Active', 'Instruct Courses in fitness', 13, 'Instructor');
INSERT INTO Employee (FirstName, LastName, StreetAddress, City, State, ZipCode,
PhoneNumber, Email, HireDate, Status, JobDuties, PayRate, Position)
VALUES('Ariah', 'Lugo', '915 Front Lane',  'Brooklyn',  'NY', '11218',
'3032419016', 'florine_zboncak@gmail.com',  '4/22/2021',  'Inactive', 'Instruct
Courses', 12, 'Instructor');
INSERT INTO Employee (FirstName, LastName, StreetAddress, City, State, ZipCode,
PhoneNumber, Email, HireDate, Status, JobDuties, PayRate, Position)
VALUES('Amanda', 'Barnett', '555 B Street',  'LaVale',  'MD', '21502',
'4234941039', 'abc427@gmail.com',  '4/01/2021',  'Active', 'Instruct Courses', 20,
'Instructor');
```

```
INSERT INTO Employee (FirstName, LastName, StreetAddress, City, State, ZipCode,
PhoneNumber, Email, HireDate, Status, JobDuties, PayRate, Position)
VALUES('Marwan', 'Ramos', '82 SE. Vale Drive', 'Bronx', 'NY',  '10461',
'8172083834', 'jocelyn.bosco@yahoo.com',  '7/5/2021',  'Active', 'Instruct Courses',
12, 'Instructor');
INSERT INTO Employee (FirstName, LastName, StreetAddress, City, State, ZipCode,
PhoneNumber, Email, HireDate, Status, JobDuties, PayRate, Position)
VALUES('Ember', 'Fuller', '7438 Orange Rd.', 'New York', 'NY', '10016',
        '7252048292',  'iva88@yahoo.com','7/6/2021',  'Active', 'Inventory, Cash
Register, Store, Sell Memberships', 11, 'Clerk');
INSERT INTO Employee (FirstName, LastName, StreetAddress, City, State, ZipCode,
PhoneNumber, Email, HireDate, Status, JobDuties, PayRate, Position)
VALUES('Isabella', 'Ross', '8763 Dogwood Ave.', 'Endicott', 'NY',  '13760',
'5689969427',  'cassandra.koch85@gmail.com', '7/12/2021', 'Active', 'Inventory,
Cash Register, Store, Sell Memberships', 13, 'Clerk');
INSERT INTO Employee (FirstName, LastName, StreetAddress, City, State, ZipCode,
PhoneNumber, Email, HireDate, Status, JobDuties, PayRate, Position)
VALUES('Bruce', 'Sullivan', '36 North Talbot Lane', 'New York', 'NY', '10023',
        '8560780734', 'pamela47@hotmail.com','10/20/2021', 'Terminated', 'Inventory,
Cash Register, Store, Sell Memberships', 14, 'Clerk');
INSERT INTO Employee (FirstName, LastName, StreetAddress, City, State, ZipCode,
PhoneNumber, Email, HireDate, Status, JobDuties, PayRate, Position)
VALUES('Frederick', 'Foster',  '395 Harvey Dr.', 'Westbury', 'NY', '11590',
'9106263144',  'jairo.hills47@yahoo.com',  '10/28/2021', 'Inactive', 'Inventory, Cash
Register, Store, Sell Memberships', 13, 'Clerk');
INSERT INTO Employee(FirstName, LastName, StreetAddress, City, State, ZipCode,
PhoneNumber, Email, HireDate, Status, JobDuties, PayRate, Position)
VALUES('Honey', 'Gray', '7826 W. Manor Station Street', 'Lindenhurst', 'NY', '11757',
'3420427015',  'oliver.berge@yahoo.com', '12/7/2021',  'Active', '', 11, 'Clerk');
INSERT INTO Employee (FirstName, LastName, StreetAddress, City, State, ZipCode,
PhoneNumber, Email, HireDate, Status, JobDuties, PayRate, Position)
VALUES('Adrian', 'Taylor', '8583B Baker Drive', 'Brooklyn', 'NY',  '11221',
'2029218226',  'nathanial_klein71@yahoo.com',  '3/18/2022',  'Active', '', 12,
'Clerk');
INSERT INTO Employee (FirstName, LastName, StreetAddress, City, State, ZipCode,
PhoneNumber, Email, HireDate, Status, JobDuties, PayRate, Position)
VALUES('Adrian', 'Taylor', '8583B Baker Drive', 'Brooklyn', 'NY',  '11221',
'2029218226',  'nathanial_klein71@yahoo.com',  '3/18/2022',  'Active', '', 12,
'Clerk');
```

```
INSERT INTO Employee (FirstName, LastName, StreetAddress, City, State, ZipCode,
PhoneNumber, Email, HireDate, Status, JobDuties, PayRate, Position)
VALUES('Aston', 'Stewart', '563 Glen Eagles St.', 'Poughkeepsie',    'NY', '12603',
'8248060878', 'marta_spencer@hotmail.com',   '4/11/2022',   'Active', '', 13, 'Clerk');
INSERT INTO Employee (FirstName, LastName, StreetAddress, City, State, ZipCode,
PhoneNumber, Email, HireDate, Status, JobDuties, PayRate, Position)
VALUES('Lana', 'Lloyd', '418 Court Ave.', 'Bronx', 'NY', '10468', '3005738158',
        'enrique13@gmail.com',     '4/14/2022',   'Leave', 'Instruct Courses', 15,
'Instructor');
INSERT INTO Employee (FirstName, LastName, StreetAddress, City, State, ZipCode,
PhoneNumber, Email, HireDate, Status, JobDuties, PayRate, Position)
VALUES('Eleanor',  'Harper', '36 Theatre St.', 'Bronx', 'NY', '10458', '9612398259',
        'amiya.kessler@yahoo.com',        '7/25/2022',   'Active', 'Instruct Courses', 12,
'Instructor');
INSERT INTO Employee (FirstName, LastName, StreetAddress, City, State, ZipCode,
PhoneNumber, Email, HireDate, Status, JobDuties, PayRate, Position)
VALUES('James', 'Stewart', '148 Randall Mill Street',   'Brooklyn', 'NY', '11214',
'2201592133',         'delilah23@hotmail.com',   '8/22/2022',   'Active', 'Instruct Courses',
12, 'Instructor');
INSERT INTO Employee (FirstName, LastName, StreetAddress, City, State, ZipCode,
PhoneNumber, Email, HireDate, Status, JobDuties, PayRate, Position)
VALUES('Melanie',  'Clark', '898 Lawrence St.', 'Brooklyn', 'NY', '11228', '1189191237',
        'amber23@yahoo.com',     '9/2/2022',     'Active', 'Instruct Courses', 11,
'Instructor');


INSERT INTO Instructor (IEmployeeID, Position)
VALUES(1, 'Instructor');
INSERT INTO Instructor (IEmployeeID, Position)
VALUES(3, 'Instructor');
INSERT INTO Instructor (IEmployeeID, Position)
VALUES(6, 'Instructor');
INSERT INTO Instructor (IEmployeeID, Position)
VALUES(7, 'Instructor');
INSERT INTO Instructor (IEmployeeID, Position)
VALUES(8, 'Instructor');
INSERT INTO Instructor (IEmployeeID, Position)
VALUES(9, 'Instructor');
INSERT INTO Instructor (IEmployeeID, Position)
VALUES(17, 'Instructor');
```

```
INSERT INTO Instructor (IEmployeeID, Position)
VALUES(18, 'Instructor');
INSERT INTO Instructor (IEmployeeID, Position)
VALUES(19, 'Instructor');
INSERT INTO Instructor (IEmployeeID, Position)
VALUES(20, 'Instructor');


INSERT INTO Clerk (CEmployeeID, Position)
VALUES(002, 'Clerk');
INSERT INTO Clerk (CEmployeeID, Position)
VALUES(004, 'Clerk');
INSERT INTO Clerk (CEmployeeID, Position)
VALUES(005, 'Clerk');
INSERT INTO Clerk (CEmployeeID, Position)
VALUES(010, 'Clerk');
INSERT INTO Clerk (CEmployeeID, Position)
VALUES(012, 'Clerk');
INSERT INTO Clerk (CEmployeeID, Position)
VALUES(013, 'Clerk');
INSERT INTO Clerk (CEmployeeID, Position)
VALUES(014, 'Clerk');
INSERT INTO Clerk (CEmployeeID, Position)
VALUES(015, 'Clerk');
INSERT INTO Clerk (CEmployeeID, Position)
VALUES(016, 'Clerk');
INSERT INTO Clerk (CEmployeeID, Position)
VALUES(021, 'Clerk');


INSERT INTO COURSE VALUES('C0001',      'Power Hour',                'Hour of HIT
Workout', 'Novice',   '1.00', 10);
INSERT INTO COURSE VALUES('C0002',      'Curl&Crunch',       'Abs and Bicepts',
'Novice',       '1.00', 20);
INSERT INTO COURSE VALUES('C0003',      'FabFitFun',         'getting started in
the healthy lifestyle', 'Novice',       '0.30', 15);
INSERT INTO COURSE VALUES('C0004',      'Fab Abs',      'getting fabulous abs',
'Intermediate',         '1.50', 30);
INSERT INTO COURSE VALUES('C0005',      'Walk this Weigh',           'weigh ins
and accountability', 'Advanced',    '1.50', 30);
```

```
INSERT INTO COURSE VALUES('C0006', 'Shrink', 'tailored workout sessions',
'Advanced', '2.00', 25);
INSERT INTO COURSE VALUES('C0007', 'Sanity Session', 'healthy mindset
and reducing stress', 'Novice', '1.00', 20);
INSERT INTO COURSE VALUES('C0008', 'Werk It!', 'HIT full
starter', 'Novice', '2.00', 15);
INSERT INTO COURSE VALUES('C0009', 'Transform', 'strength
building', 'Intermediate', '1.50', 30);
INSERT INTO COURSE VALUES('C0010', 'Sweat Fest', 'cardio, cardio,
and cardio', 'Intermediate', '1.50', 30);


INSERT INTO QUALIFICATION VALUES('Q0001', '3/11/2020', 001, 'C0001');
INSERT INTO QUALIFICATION VALUES('Q0002', '3/31/2020', 001, 'C0002');
INSERT INTO QUALIFICATION VALUES('Q0008', '9/14/2020', 001, 'C0008');
INSERT INTO QUALIFICATION VALUES('Q0011', '11/4/2020', 020, 'C0001');
INSERT INTO QUALIFICATION VALUES('Q0005', '5/24/2021', 003, 'C0005');
INSERT INTO QUALIFICATION VALUES('Q0006', '10/29/2021', 003, 'C0006');
INSERT INTO QUALIFICATION VALUES('Q0010', '9/9/2021', 018,
    'C0010');
INSERT INTO QUALIFICATION VALUES('Q0009', '9/17/2021', 018, 'C0001');
INSERT INTO QUALIFICATION VALUES('Q0012', '12/8/2021', 006, 'C0007');
INSERT INTO QUALIFICATION VALUES('Q0013', '3/18/2022', 006, 'C0003');
INSERT INTO QUALIFICATION VALUES('Q0014', '4/7/2022', 006,
    'C0004');
INSERT INTO QUALIFICATION VALUES('Q0015', '4/15/2022', 006, 'C0001');
INSERT INTO QUALIFICATION VALUES('Q0016', '7/25/2022', 008, 'C0001');
INSERT INTO QUALIFICATION VALUES('Q0017', '4/14/2022', 007, 'C0010');
INSERT INTO QUALIFICATION VALUES('Q0018', '8/4/2022', 007,
    'C0003');
INSERT INTO QUALIFICATION VALUES('Q0019', '9/21/2021', 009, 'C0001');
INSERT INTO QUALIFICATION VALUES('Q0020', '8/22/2022', 009, 'C0007');
INSERT INTO QUALIFICATION VALUES('Q0021', '8/15/2022', 007, 'C0006');
INSERT INTO QUALIFICATION VALUES('Q0022', '9/17/2022', 019, 'C0009');
INSERT INTO QUALIFICATION VALUES('Q0003', '8/4/2022', 007,
    'C0003');


INSERT INTO MEMBERSHIPTYPE VALUES('Bronze', 50, 0.0);
INSERT INTO MEMBERSHIPTYPE VALUES('Silver', 100, 0.1);
```

```
INSERT INTO MEMBERSHIPTYPE VALUES('Gold',      200,        0.2);
INSERT INTO MEMBERSHIPTYPE VALUES('Platinum',        300,        0.3);


INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Davis',            'Mayer',                    '91 Cobblestone Street',
       'Brooklyn',                 'NY',   '11230',       '781091168', 'Male',
              '6/12/1942',   '8/25/2020',   'Platinum')
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Connie',           'Ellis',          '57 Clinton Lane',                'Rochester',
       'NY',   '14609',       '119351619', 'Female',                '10/7/1948',
       '10/6/2020',   'Gold')
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Harold',           'Reed',                     '5 High Ridge Dr.',
       'Buffalo',                  'NY',   '14221',       '666641859', 'Male',
              '2/2/1949',            '10/15/2020', 'Silver')
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Maya',             'Wright',               '64 Oak Valley Street',           'New
York',               'NY',   '10016',       '170271198', 'Female',
       '5/19/1950',   '10/30/2020', 'Silver')
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Lyndon',           'Brown',                '8253 Spring Lane',               'West
Babylon',            'NY',   '11704',       '093723729', 'Neither',       '9/13/1951',
       '11/3/2020',   'Silver')
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Jacob',            'Cunningham',       '7723 S. Corona Rd.',             'Far
Rockaway',           'NY',   '11691',       '036280664',  'Male',
       '2/25/1952',   '11/17/2020', 'Gold')
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Dexter',           'Spencer',              '143 Front Ave.',
       'StateAbvn Island',  'NY',   '10312',       '992157416', 'Male',
       '4/8/1952',            '11/20/2020', 'Bronze')
```

```
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Vincent', 'Cole', '19 Clark Court', 'New
York', 'NY', '10032', '242875932', 'Neither',
'10/2/1952', '12/8/2020', 'Silver')
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Alina', 'Mitchell', '2 NW. Third St.',
'Tonawanda', 'NY', '14150', '889249087', 'Female',
'3/14/1956', '1/12/2021', 'Silver')
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Hailey', 'Warren', '12 North King Road',
'Brooklyn', 'NY', '11220', '723153433', 'Female',
'10/31/1961', '1/14/2021', 'Bronze')
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Max', 'Tucker', '15 South Brickell Ave.',
'Bronx', 'NY', '10467', '282827651', 'Male',
'10/19/1966', '5/24/2021', 'Silver')
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Albert', 'Turner', '13 Carson St.',
'Bronx', 'NY', '10472', '440300169', 'Male',
'5/6/1968', '6/16/2021', 'Bronze')
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Clark', 'Parker', '631 Maple Lane',
'Spring Valley', 'NY', '10977', '969087772', 'Male',
'1/21/1971', '6/18/2021', 'Platinum')
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Preston', 'Myers', '17 Wild Rose Dr.',
'Brooklyn', 'NY', '11201', '944433564', 'Male',
'3/26/1971', '7/20/2021', 'Gold')
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Chloe', 'Cameron', '730 Manchester Street', 'Webster',
'NY', '14580', '201359623', 'Female',
'8/19/1971', '8/24/2021', 'Bronze')
```

```
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Naomi',          'Hill',              '280 Walnut Road',
       'Ridgewood',      'NY',   '11385',     '334047938', 'Neither',     '10/30/1973',
       '9/28/2021',   'Gold')
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Grace',          'Russell',           '8341 Brewery St.',
       'Westbury',             'NY',   '11590',     '722740914', 'Female',
            '3/16/1979',   '11/22/2021', 'Bronze')
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Maya',           'Payne',             '8263 SW. Hill Street',
       'Staten Island',  'NY',   '10306',     '196553594', 'Neither',
       '1/30/1985',   '1/19/2022',   'Platinum')
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Vincent',        'Brooks',            '736 Carson Ave.',
       'Bronx',                'NY',   '10465',     '632785243', 'Male',
            '4/18/1985',   '2/14/2022',   'Silver')
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Daryl',          'Perkins',           '9107 San Carlos Street',   'Bronx',
            'NY',   '10463',     '301859975', 'Male',
       '4/17/1986',   '6/3/2022',          'Silver')
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Elian',          'Montgomery',        '8612 North Gates St.',
       'Brooklyn',             'NY',   '11213',     '281360754', 'Male',
            '9/16/1992',   '7/22/2022',   'Platinum')
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Carlos',         'Nelson',            '7384 George Ave.',
       'Brooklyn',             'NY',   '11216',     '310026546', 'Male',
            '10/13/1993', '8/17/2022',   'Silver')
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Leonardo',       'Myers',             '7 Old Thompson Drive',
       'Elmont',               'NY',   '11003',     '703913979', 'Male',
            '6/2/1995',          '9/16/2022',   'Platinum')
```

```
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Emma',      'Richards',      '211 Wayne Street',      'Woodside',
            'NY',   '11377',      '498604767', 'Female',
      '1/26/1998',   '9/23/2022',   'Bronze')
INSERT INTO MEMBER (FirstName, LastName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Gender, BirthDate, DateJoined, LevelName)
VALUES('Oliver',      'Chapman',      '698 Leatherwood St.',      'New
York',      'NY',   '10128',      '753980520', 'Prefer Not To Say',
      '5/25/2004',   '11/2/2022',   'Bronze')


INSERT INTO ENROLLMENTLIST (PurchaseDate, PurchasePrice,CourseNumber,
MemberID)
VALUES('12-30-2020',      10,     'C0001',      1003);
INSERT INTO ENROLLMENTLIST (PurchaseDate, PurchasePrice,CourseNumber,
MemberID)
VALUES('8/26/2021',      20,     'C0002',      1004);
INSERT INTO ENROLLMENTLIST (PurchaseDate, PurchasePrice,CourseNumber,
MemberID)
VALUES('9/13/2021',      15,     'C0008',      1005);
INSERT INTO ENROLLMENTLIST (PurchaseDate, PurchasePrice,CourseNumber,
MemberID)
VALUES('11/24/2021',      30,     'C0009',      1006);
INSERT INTO ENROLLMENTLIST (PurchaseDate, PurchasePrice,CourseNumber,
MemberID)
VALUES('12/1/2021',      30,     'C0005',      1001);
INSERT INTO ENROLLMENTLIST (PurchaseDate, PurchasePrice,CourseNumber,
MemberID)
VALUES('12/23/2021',      25,     'C0006',      1010);
INSERT INTO ENROLLMENTLIST (PurchaseDate, PurchasePrice,CourseNumber,
MemberID)
VALUES('1/26/2022',      20,     'C0007',      1016);
INSERT INTO ENROLLMENTLIST (PurchaseDate, PurchasePrice,CourseNumber,
MemberID)
VALUES('2/7/2022',    30,  'C0010',      1005);
INSERT INTO ENROLLMENTLIST (PurchaseDate, PurchasePrice,CourseNumber,
MemberID)
VALUES('8/29/2022',      15,     'C0003',      1006);
```

```
INSERT INTO ENROLLMENTLIST (PurchaseDate, PurchasePrice,CourseNumber,
MemberID)
VALUES('10/6/2022',        30,     'C0004',        1018);


INSERT INTO Classroom VALUES('01' ,23);
INSERT INTO Classroom VALUES('02' ,9);
INSERT INTO Classroom VALUES('03' ,25);
INSERT INTO Classroom VALUES('04' ,24);
INSERT INTO Classroom VALUES('05' ,6);
INSERT INTO Classroom VALUES('06' ,17);
INSERT INTO Classroom VALUES('07' ,9);
INSERT INTO Classroom VALUES('08' ,27);
INSERT INTO Classroom VALUES('09' ,12);
INSERT INTO Classroom VALUES('10' ,7);


INSERT INTO Class VALUES('CL001' , '10/6/2022','M,W,F','8:00 AM',006, '01',
'C0001');
INSERT INTO Class VALUES('CL002' , '3/18/2022','T,R','8:00 AM',006, '02', 'C0002');
INSERT INTO Class VALUES('CL003' , '4/7/2022','T,R','10:00 AM',006, '01','C0003');
INSERT INTO Class VALUES('CL004' , '7/25/2022','M,W,F','10:00 AM',008, '06',
'C0004');
INSERT INTO Class VALUES('CL005' , '4/14/2022','M,W,F','12:00 PM',007, '07',
'C0005');
INSERT INTO Class VALUES('CL006' , '8/4/2022','M','9:00 AM',007, '08', 'C0006');
INSERT INTO Class VALUES('CL007' , '9/21/2021','T','12:00 PM',009, '01', 'C0007');
INSERT INTO Class VALUES('CL008' , '8/22/2022','F','4:00 PM',009, '05', 'C0008');
INSERT INTO Class VALUES('CL009' , '8/15/2022','T,R','2:00 PM',007, '10', 'C0009');
INSERT INTO Class VALUES('CL010' , '9/17/2022','M,W,F','2:00 PM',017, '07',
'C0010');


INSERT INTO MemberJoins VALUES(1003 , 'CL001');
INSERT INTO MemberJoins VALUES(1007 , 'CL006');
INSERT INTO MemberJoins VALUES(1008 , 'CL004');
INSERT INTO MemberJoins VALUES(1021 , 'CL010');
INSERT INTO MemberJoins VALUES(1022 , 'CL001');
INSERT INTO MemberJoins VALUES(1023 , 'CL001');
INSERT INTO MemberJoins VALUES(1002 , 'CL007');
```

```
INSERT INTO MemberJoins VALUES(1018 , 'CL007');
INSERT INTO MemberJoins VALUES(1001 , 'CL009');
INSERT INTO MemberJoins VALUES(1017 , 'CL002');


INSERT INTO Manufacturer (CompanyName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Email)
VALUES('ABC Co.', '123 A St.', 'Clevland', 'TN', '37900', '1234567890', 'info@abc.com');
INSERT INTO Manufacturer (CompanyName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Email)
VALUES('123 Co.', 'ABC A St.', 'Loudon', 'TN', '37904', '1452086248', 'info@123.com');
INSERT INTO Manufacturer (CompanyName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Email)
VALUES('WestRock', '24 N 1 St.', 'Knoxville', 'TN', '37900', '8886665555',
'info@westrock.com');
INSERT INTO Manufacturer (CompanyName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Email)
VALUES('HIT Is Us', '222 West St.', 'Knoxville', 'TN', '37919', '8658658650',
'info@hitisus.com');
INSERT INTO Manufacturer (CompanyName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Email)
VALUES('Solinity', '714 S Gay St.', 'Knoxville', 'TN', '37902', '4234941039',
'Amanda@solinity.com');
INSERT INTO Manufacturer (CompanyName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Email)
VALUES('Good 4 U', '424 Candy Cane Lane', 'Knoxville', 'TN', '37900', '2528685741',
'info@g4u.com');
INSERT INTO Manufacturer (CompanyName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Email)
VALUES('Sweatys', '8 That Drive', 'Knoxville', 'TN', '37900', '6666666666',
'info@sweatys.com');
INSERT INTO Manufacturer (CompanyName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Email)
VALUES('SweatRUs', '90210 Sunset Blvd.', 'Sweetwater', 'TN', '37900', '888888888',
'info@sweatrus.com');
INSERT INTO Manufacturer (CompanyName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Email)
VALUES('Everything Is Bigger', '66 A St.', 'Dallas', 'TX', '55555', '111111111',
'info@everythingisbiggerTX.com');
```

```
INSERT INTO Manufacturer (CompanyName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Email)
VALUES('Meat Heads', '26 G St.', 'SanDiego', 'CA', '11601', '9898989898',
'info@meatheads.com');


INSERT INTO EquipmentType VALUES('E01','bands','483','3 mths',201);
INSERT INTO EquipmentType VALUES('E02','treadmill','7013','1 yr',209);
INSERT INTO EquipmentType VALUES('E03','leg press','5939','1 mth',205);
INSERT INTO EquipmentType VALUES('E04','ball','2860','6mth',202);
INSERT INTO EquipmentType VALUES('E05','shoulder press','9453','3 mths',210);
INSERT INTO EquipmentType VALUES('E06','bar','6674','1 mth',207);
INSERT INTO EquipmentType VALUES('E07','stepper','8733','3 mths',206);
INSERT INTO EquipmentType VALUES('E08','elyptical','534','1 yr',204);
INSERT INTO EquipmentType VALUES('E09','pull-up machine','2693','1 mth',203);
INSERT INTO EquipmentType VALUES('E10','row machine','9850','6mth',208);


INSERT INTO EquipmentUse VALUES('CL001','E09',23);
INSERT INTO EquipmentUse VALUES('CL002','E09',9);
INSERT INTO EquipmentUse VALUES('CL003','E08',25);
INSERT INTO EquipmentUse VALUES('CL004','E02',24);
INSERT INTO EquipmentUse VALUES('CL005','E10',6);
INSERT INTO EquipmentUse VALUES('CL006','E05',17);
INSERT INTO EquipmentUse VALUES('CL007','E02',9);
INSERT INTO EquipmentUse VALUES('CL008','E08',27);
INSERT INTO EquipmentUse VALUES('CL009','E01',12);
INSERT INTO EquipmentUse VALUES('CL010','E04',7);


INSERT INTO Equipment VALUES('262115','11/11/2020','11/16/2020','E01');
INSERT INTO Equipment VALUES('1801862','12/17/2020','2/19/2021','E02');
INSERT INTO Equipment VALUES('3157605','2/24/2021','4/27/2021','E03');
INSERT INTO Equipment VALUES('9911213','3/2/2021','5/31/2021','E04');
INSERT INTO Equipment VALUES('703978','9/7/2021','11/15/2021','E05');
INSERT INTO Equipment VALUES('4618851','11/19/2021','12/3/2021','E06');
INSERT INTO Equipment VALUES('2503934','11/24/2021','12/30/2021','E07');
INSERT INTO Equipment VALUES('4901783','2/14/2022','2/23/2022','E08');
INSERT INTO Equipment VALUES('4303233','5/4/2022','5/19/2022','E09');
INSERT INTO Equipment VALUES('8481043','6/10/2022','10/11/2022','E10');
```

```
INSERT INTO WorkOrder (DateofOrder, SerialNumber, ManufacturerID,
DateCompleted)
VALUES('8/23/2021', '262115',210,'2/15/2021');
INSERT INTO WorkOrder (DateofOrder, SerialNumber, ManufacturerID,
DateCompleted)
VALUES('9/14/2021', '4618851',201,'2/22/2021');
INSERT INTO WorkOrder (DateofOrder, SerialNumber, ManufacturerID,
DateCompleted)
VALUES('11/11/2021', '8481043',209,'3/4/2021');
INSERT INTO WorkOrder (DateofOrder, SerialNumber, ManufacturerID,
DateCompleted)
VALUES('11/12/2021', '4901783',202,'4/14/2021');
INSERT INTO WorkOrder (DateofOrder, SerialNumber, ManufacturerID,
DateCompleted)
VALUES('12/27/2021', '4303233',208,'9/10/2021');
INSERT INTO WorkOrder (DateofOrder, SerialNumber, ManufacturerID,
DateCompleted)
VALUES('2/16/2022', '2503934',206,'12/3/2021');
INSERT INTO WorkOrder (DateofOrder, SerialNumber, ManufacturerID,
DateCompleted)
VALUES('3/23/2022', '9911213',207,'1/5/2022');
INSERT INTO WorkOrder (DateofOrder, SerialNumber, ManufacturerID,
DateCompleted)
VALUES('6/15/2022', '1801862',203,'3/16/2022');
INSERT INTO WorkOrder (DateofOrder, SerialNumber, ManufacturerID,
DateCompleted)
VALUES('8/1/2022', '3157605',205,'5/17/2022');
INSERT INTO WorkOrder (DateofOrder, SerialNumber, ManufacturerID,
DateCompleted)
VALUES('10/10/2022', '703978',204,'8/22/2022');


INSERT INTO VENDOR (CompanyName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Email, FaxNumber)
VALUES('Get Jacked',       '404 North Liberty Ave.', 'Lindenhurst', 'NY',
'11757','6322917218', 'k.sullivan@randatmail.com',     '1580565663');
```

```
INSERT INTO VENDOR (CompanyName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Email, FaxNumber)
VALUES('Ab City',   '398 Elizabeth Rd.', 'Brooklyn', 'NY',        '11235', '6241097849',
'n.edwards@randatmail.com', '9751001985');
INSERT INTO VENDOR (CompanyName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Email, FaxNumber)
VALUES('Bis and Tris', '4 Ocean Lane', 'Bronx', 'NY',   '10457', '4154312021',
'h.dixon@randatmail.com', '3447862673');
INSERT INTO VENDOR(CompanyName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Email, FaxNumber)
VALUES('Pump Iron', '7721 Ridgewood Drive', 'Poughkeepsie',       'NY', '12603',
        '5071041463',         'a.walker@randatmail.com',         '8001015019');
INSERT INTO VENDOR (CompanyName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Email, FaxNumber)
VALUES('Dumbbell', '804 West Hill St.',  'Buffalo',        'NY',   '14221',
        '5411717639',         'e.adams@randatmail.com',         '9998635734');
INSERT INTO VENDOR (CompanyName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Email, FaxNumber)
VALUES('Olympic Fit',       '9437 Durham Street',       'Rome',         'NY',   '13440',
        '5934212475',         'n.taylor@randatmail.com', '2032359626');
INSERT INTO VENDOR (CompanyName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Email, FaxNumber)
VALUES('Fitfinity',   '384 St Louis Street',          'Staten Island',        'NY',   '10312',
        '6201291901',         'v.chapman@randatmail.com',      '6334300288');
INSERT INTO VENDOR (CompanyName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Email, FaxNumber)
VALUES('Muscle Mass',    '51 Thatcher Drive', 'Poughkeepsie',       'NY',   '12601',
        '4349047475',         'a.cunningham@randatmail.com', '8114470115');
INSERT INTO VENDOR (CompanyName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Email, FaxNumber)
VALUES('Flexion',   '36 West Market Ave.',       'Rego Park',  'NY',   '11374',
        '3733736937',         'n.harper@randatmail.com',         '6612946177');
INSERT INTO VENDOR (CompanyName, StreetAddress, City, StateAbv, Zip,
PhoneNumber, Email, FaxNumber)
VALUES('Plyometry',         '7670 West Amerige Drive',          'Westbury',    'NY', '11590',
        '8388866563', 'e.farrell@randatmail.com',         '6061352407');


INSERT INTO PURCHASEORDER (DatePlaced, DateReceived, TotalCost, VendorID,
CEmployeeID)
```

```
VALUES('1/28/2021',  '3/24/2021',  7473, 4,  005);
INSERT INTO PURCHASEORDER (DatePlaced, DateReceived, TotalCost, VendorID,
CEmployeeID)
VALUES('4/6/2021', '4/19/2021',  1536, 4,  004);
INSERT INTO PURCHASEORDER (DatePlaced, DateReceived, TotalCost, VendorID,
CEmployeeID)
VALUES('4/23/2021',  '8/6/2021',  2385, 10,  016);
INSERT INTO PURCHASEORDER (DatePlaced, DateReceived, TotalCost, VendorID,
CEmployeeID)
VALUES('8/9/2021', '10/20/2021', 6844, 1,  010);
INSERT INTO PURCHASEORDER (DatePlaced, DateReceived, TotalCost, VendorID,
CEmployeeID)
VALUES('12/21/2021',  '12/24/2021', 570, 6,  021);
INSERT INTO PURCHASEORDER (DatePlaced, DateReceived, TotalCost, VendorID,
CEmployeeID)
VALUES('2/21/2022',  '4/11/2022',  2596, 6,  021);
INSERT INTO PURCHASEORDER (DatePlaced, DateReceived, TotalCost, VendorID,
CEmployeeID)
VALUES('4/27/2022',  '6/7/2022',  4248, 6,  002);
INSERT INTO PURCHASEORDER (DatePlaced, DateReceived, TotalCost, VendorID,
CEmployeeID)
VALUES('8/5/2022', '8/8/2022',  2432, 2,  013);
INSERT INTO PURCHASEORDER (DatePlaced, DateReceived, TotalCost, VendorID,
CEmployeeID)
VALUES('11/4/2022',  '11/11/2022', 7125, 8,  002);
INSERT INTO PURCHASEORDER (DatePlaced, DateReceived, TotalCost, VendorID,
CEmployeeID)
VALUES('12/1/2022',  '12/12/2022', 2048, 1,  004);


INSERT INTO ITEM (Descrip, StandardPrice, QuantityonHand, ReorderPoint)
VALUES('Fitness Equipment',  16,  35,  50);
INSERT INTO ITEM (Descrip, StandardPrice, QuantityonHand, ReorderPoint)
VALUES('Fitness Equipment',  53,  52,  50);
INSERT INTO ITEM (Descrip, StandardPrice, QuantityonHand, ReorderPoint)
VALUES('Fitness Equipment',  57,  94,  75);
INSERT INTO ITEM (Descrip, StandardPrice, QuantityonHand, ReorderPoint)
VALUES('Fitness Equipment',  48,  75,  100);
INSERT INTO ITEM (Descrip, StandardPrice, QuantityonHand, ReorderPoint)
VALUES('Fitness Equipment',  59,  84,  25);
```

```
INSERT INTO ITEM (Descrip, StandardPrice, QuantityonHand, ReorderPoint)
VALUES('Fitness Equipment',     80,     61,     25);
INSERT INTO ITEM (Descrip, StandardPrice, QuantityonHand, ReorderPoint)
VALUES('Fitness Equipment',     68,     65,     10);
INSERT INTO ITEM (Descrip, StandardPrice, QuantityonHand, ReorderPoint)
VALUES('Fitness Equipment',     90,     42,     50);
INSERT INTO ITEM (Descrip, StandardPrice, QuantityonHand, ReorderPoint)
VALUES('Fitness Equipment',     43,     70,     25);
INSERT INTO ITEM (Descrip, StandardPrice, QuantityonHand, ReorderPoint)
VALUES('Fitness Equipment',     15,     38,     25);


INSERT INTO PURCHASEORDERLINE (ItemsOrdered, ItemQuantity,
QuantityReceived, VendorCost, CEmployeeID, ItemID)
VALUES('socks',     141,    141,    53,     002,    2);
INSERT INTO PURCHASEORDERLINE (ItemsOrdered, ItemQuantity,
QuantityReceived, VendorCost, CEmployeeID, ItemID)
VALUES( 'shirts',     96,             96,             16, 002,        1);
INSERT INTO PURCHASEORDERLINE (ItemsOrdered, ItemQuantity,
QuantityReceived, VendorCost, CEmployeeID, ItemID)
VALUES('towel',     45,             45,             53,     004,    2);
INSERT INTO PURCHASEORDERLINE (ItemsOrdered, ItemQuantity,
QuantityReceived, VendorCost, CEmployeeID, ItemID)
VALUES('keychains',         116,    116,    59,     004,    5);
INSERT INTO PURCHASEORDERLINE (ItemsOrdered, ItemQuantity,
QuantityReceived, VendorCost, CEmployeeID, ItemID)
VALUES('shirts',     38,             38,             15,     005,    10);
INSERT INTO PURCHASEORDERLINE(ItemsOrdered, ItemQuantity,
QuantityReceived, VendorCost, CEmployeeID, ItemID)
VALUES('cleaner',   44,             44,             59,     005,    5);
INSERT INTO PURCHASEORDERLINE(ItemsOrdered, ItemQuantity,
QuantityReceived, VendorCost, CEmployeeID, ItemID)
VALUES('papertowle',        72,             72,             59,     010,    5);
INSERT INTO PURCHASEORDERLINE (ItemsOrdered, ItemQuantity,
QuantityReceived, VendorCost, CEmployeeID, ItemID)
VALUES('vitamins', 152,    152,    16,     010,    1);
INSERT INTO PURCHASEORDERLINE (ItemsOrdered, ItemQuantity,
QuantityReceived, VendorCost, CEmployeeID, ItemID)
VALUES('drinks',    125,    125,    57,     012,    3);
```

```
INSERT INTO PURCHASEORDERLINE (ItemsOrdered, ItemQuantity,
QuantityReceived, VendorCost, CEmployeeID, ItemID)
VALUES('pre-workout',      128,   128,   16,     013,   1);


INSERT INTO SaleTransaction (TDate, ItemsPurchased, UnitPriceCharged,
PurchaseTotal, CEmployeeID, MemberID)
VALUES('2/24/2021',        1,      17.6,  105.6, 010,   1019);
INSERT INTO SaleTransaction (TDate, ItemsPurchased, UnitPriceCharged,
PurchaseTotal, CEmployeeID, MemberID)
VALUES('3/24/2021',        7,      58.3,  233.2, 004,   1010);
INSERT INTO SaleTransaction (TDate, ItemsPurchased, UnitPriceCharged,
PurchaseTotal, CEmployeeID, MemberID)
VALUES('4/15/2021',        2,      62.7,  188.1, 002,   1003);
INSERT INTO SaleTransaction (TDate, ItemsPurchased, UnitPriceCharged,
PurchaseTotal, CEmployeeID, MemberID)
VALUES('4/26/2021',        4,      52.8,  528    ,       004,   1008);
INSERT INTO SaleTransaction (TDate, ItemsPurchased, UnitPriceCharged,
PurchaseTotal, CEmployeeID, MemberID)
VALUES('5/6/2021', 7,      64.9,  194.7, 002,   1007);
INSERT INTO SaleTransaction (TDate, ItemsPurchased, UnitPriceCharged,
PurchaseTotal, CEmployeeID, MemberID)
VALUES('9/22/2021',        9,      88,            792,   014,   1020);
INSERT INTO SaleTransaction (TDate, ItemsPurchased, UnitPriceCharged,
PurchaseTotal, CEmployeeID, MemberID)
VALUES('1/3/2022', 1,      74.8,  673.2, 005,   1016);
INSERT INTO SaleTransaction (TDate, ItemsPurchased, UnitPriceCharged,
PurchaseTotal, CEmployeeID, MemberID)
VALUES('7/7/2022', 7,      99,            396,   010,   1008);
INSERT INTO SaleTransaction (TDate, ItemsPurchased, UnitPriceCharged,
PurchaseTotal, CEmployeeID, MemberID)
VALUES('8/9/2022', 1,      47.3,  236.5, 013,   1008);
INSERT INTO SaleTransaction (TDate, ItemsPurchased, UnitPriceCharged,
PurchaseTotal, CEmployeeID, MemberID)
VALUES('8/23/2022',        3,      16.5,  16.5,  015,   1004);


INSERT INTO SaleOrderLine (ItemQty, PricePerUnit, ItemID, TransactionNumber)
VALUES(6,  17.6,  1,      1);
INSERT INTO SaleOrderLine (ItemQty, PricePerUnit, ItemID, TransactionNumber)
```

```
VALUES(4,   58.3,   7,      2);
INSERT INTO SaleOrderLine (ItemQty, PricePerUnit, ItemID, TransactionNumber)
VALUES(3,   62.7,   2,      3);
INSERT INTO SaleOrderLine (ItemQty, PricePerUnit, ItemID, TransactionNumber)
VALUES(10, 52.8,   4,      4);
INSERT INTO SaleOrderLine (ItemQty, PricePerUnit, ItemID, TransactionNumber)
VALUES(3,   64.9,   7,      5);
INSERT INTO SaleOrderLine (ItemQty, PricePerUnit, ItemID, TransactionNumber)
VALUES(9,   88,             9,      6);
INSERT INTO SaleOrderLine (ItemQty, PricePerUnit, ItemID, TransactionNumber)
VALUES(9,   74.8,   1,      7);
INSERT INTO SaleOrderLine (ItemQty, PricePerUnit, ItemID, TransactionNumber)
VALUES(4,   99,             7,      8);
INSERT INTO SaleOrderLine (ItemQty, PricePerUnit, ItemID, TransactionNumber)
VALUES(5,   47.3,   1,      9);
INSERT INTO SaleOrderLine(ItemQty, PricePerUnit, ItemID, TransactionNumber)
VALUES(1,   16.5,   3,      10);
```

Data Dictionary

Employee

Name	Description	Data Type	Data Length	Required	Default Value
EmployeeID	Unique ID given to each employee upon hire. No duplicates and it stays with the employee if they leave and come back to the company. This is the primary key.	Auto Number		Y	
FirstName	Full first name of the individual.	character	25 variable	N	
LastName	Full last name of the individual.	character	25 variable	N	
StreetAddress	Number and street name of the individual.	character	50 variable	N	
City	City name where the individual lives.	character	30 variable	N	
State	State abbreviation where the individual lives.	character	2 fixed	N	
ZipCode	Zip code where the individual lives.	character	5 fixed	N	
PhoneNumber	10 digit code of the individual's primary phone number.	character	10 fixed	N	
Email	The full primary email address of the individual.	character	50 variable	N	
HireDate	Date of hire (Month/Day/Year Format)	date	8 fixed	N	
Status	Status of employment. This is a state of employment that the	character	10 variable	N	
PayRate		Money		N	>0
Position	Name of the position the employee currently	character	15 variable	N	

	holds.				

Instructor

Name	Description	Data Type	Data Length	Required	Default Value
IEmployeeID	Unique ID given to each employee upon hire. No duplicates and it stays with the employee if they leave and come back to the company. This is a foreign primary key and is pulled from the EmployeeID in the Employee table. This is the primary key.	Auto Number		Y	
Position	This is a foreign key and is pulled from the Position in the Employee table. This must be Instructor for this table.	VARCHAR	15	Y	Instructor

Clerk

Name	Description	Data Type	Data Length	Required	Default Value
CEmployeeID	Unique ID given to each employee upon hire. No duplicates and it stays with the employee if they leave and come back to the company. This is a foreign primary key and is pulled from the EmployeeID in the Employee table. This is the primary key.	Auto Number		Y	
Position	This is a foreign key and is pulled from the Position in the Employee table. This must be Clerk for this table.	VARCHAR	15	Y	Clerk

MembershipType

Name	Description	Data Type	Data Length	Required	Default Value
LevelName	The name given to the membership level. This is the primary key.	character	8 variable	Y	Platinum', 'Gold', 'Silver', 'Bronze'
JoiningFee	Fee charged upon joining	Money		N	>0
DiscountPercentage	Decimal value of percent discount allotted by membership	integer		N	>=0 & <=1.0

Course

Name	Description	Data Type	Data Length	Required	Default Value
CourseNumber	Unique number assigned to course when it is first created. If a course is canceled, this are not deleted since it may return in the future. This is the primary key.	character	5 fixed	Y	
CourseName	Name of course	character	15 variable	N	
Description	Description of the course	character	50 variable	N	
Category(N/I/A)	Categorized as Novice, Intermediate, or Advanced based on skill level and movement required	character	15 variable	N	Novice', 'Intermediate', 'Advanced'
Duration	Time each class will take (including 5 minute set up and intro and 5 minute clean up)	Decimal	(6,2)	N	>0
CourseFee	Fee charged	Money		N	

Qualification

Name	Description	Data Type	Data Length	Required	Default Value
QualificationID	Unique number assigned to qualification. This is the primary key.	Character	5 fixed	Y	
DateQualified	Date of completion and passing of certification training	Date		N	
IEmployeeID	This is a foreign key called IEmployeeID from the Instructor table. This is the ID of the employee who has qualified.	Auto Number		Y	
CourseNumber	Unique number assigned to course the employee got qualified for. This is a foreign key called CourseNumber from the Course table.	Character	5 fixed	Y	

Member

Name	Description	Data Type	Data Length	Required	Default Value
MemberID	Unique ID given to each member upon opening of the membership. No duplicates. If a member changes levels or churns and returns, a new number is assigned. This is the primary key.	Auto Number		Y	
FirstName	Full first name of the individual.	character	25 variable	N	
LastName	Full last name of the individual.	character	25 variable	N	
StreetAddress	Number and street name of the individual.	character	50 variable	N	
City	City name where the individual lives.	character	30 variable	N	
State	State abriviation where the individual lives.	character	2 fixed	N	
Zip	Zip code where the individual lives.	character	5 fixed	N	
PhoneNumber	10 digit code of the individual's primary phone number.	character	10 fixed	N	
Gender	1 of 4 options provided	character	20 variable	N	Male', 'Female', 'Neither', or 'Prefer Not To Say'
BirthDate	Full birthday of the member (Month/Day/Year Format)	date		N	

DateJoined	Date of start of membership (Month/Day/Year Format)	date		N	
LevelName	This is a foreign key called LevelName from the MembershipType table. It can be 1 of 4 available membership levels.	character	8 variable	Y	Platinum', 'Gold', 'Silver', 'Bronze'

EnrollmentList

Name	Description	Data Type	Data Length	Required	Default Value
EnrollmentNumber	Unique ID given to each enrollment when a member enrolls into a course. This is the primary key.	Auto Number		Y	
PurchaseDate	Date the purchase occurs.	Date		N	
PurchasePrice	Price of purchased enrollment	Money		N	
CourseNumber	This is a foreign key called CourseNumber from the Course table.	character	5 fixed	Y	
MemberID	This is a foreign key called MemberID from the Member table.	Auto Number		Y	

Classroom

Name	Description	Data Type	Data Length	Required	Default Value
RoomNumber	Room number assigned to the room the class is held. This is the primary key.	Character	2 fixed	Y	
ClassCapacity	Maximum number of people allowed in the class	integer		N	>0

Class

Name	Description	Data Type	Data Length	Required	Default Value
ReferenceNo.	Number used to reference the class. This is the primary key.	Character	5 fixed	Y	
StartDate	Date the class starts	Date		N	
DaysHeld	Days of the week of which the class is held. Entered in 1 letter abriviations.	Character	7 variable	N	
TimeHeld	Start time of the class	Time		N	
IEmployeeID	This is an ID of the employee teaching the class. This is a foreign key called IEmployeeID from the Instructor table.	Auto Number		Y	
RoomNumber	Room number assigned to the room the class is held. This is a foreign key called RoomNumber from the Classroom table.	Character	2 fixed	Y	

CourseNumber	Identification number of the course that this class is a option of. This is a foreign key called CourseNumber from the Course table.	Character	5 fixed	Y	

MemberJoins

Name	Description	Data Type	Data Length	Required	Default Value
MemberID	This is an identification of the member. It is a foreign key called MemberID from the Member table and part of the primary key.	Auto Number		Y	
ReferenceNo	Number used to reference the class. This is a foreign key called ReferenceNo from the Class table and part of the primary key.	Character	5 fixed	Y	

Manufacturer

Name	Description	Data Type	Data Length	Required	Default Value
ManufacturerID	Unique ID given to each manufacturer upon initial purchase of equipment. No duplicates and it stays with the company if they change names. Convert all records to more frequent ID if a merger occurs. This is the primary key.	Auto Number		Y	
CompanyName	Full legal name of the company	Character	40 variable	N	
StreetAddress	Number and street name of the manufacturer.	Character	50 variable	N	
City	City name where the manufacturer is located.	Character	30 variable	N	
State	State abriviation where the manufacturer is located.	Character	2 fixed	N	
Zip	Zip code where the manufacturer is located.	Character	5 fixed	N	
PhoneNumber	10 digit code of the manufacturer's primary phone number.	Character	10 fixed	N	
Email	The full primary email address of the manufacturer contact.	Character	50 variable	N	

EquipmentType

Name	Description	Data Type	Data Length	Required	Default Value
IdentificationNumber	Unique number assigned to equipment type. This is the primary key.	character	3 fixed	Y	
Description	Description of the type of equipment	character	50 variable	N	
ManufacturerModelNumber	The model number printed on that equipment type	character	15 variable	N	
MaintenanceInterval	The interval at which maintenance is recommended for the type of equipment	character	10 variable	N	
ManufacturerID	This is a foreign key called ManufacturerID from the Manufacturer table that connects the manufacturer to the equipment type.	Auto Number		Y	

EquipmentUse

Name	Description	Data Type	Data Length	Required	Default Value
IdentificationNumber	Unique number assigned to equipment type. This is a foreign key called IdentificationNumber from the EquipmentType table. This is part of the primary key.	Character	3 fixed	Y	
ReferenceNo	Number used to reference the class. This is a foreign key called ReferenceNo from the Class table. This is part of the primary key.	Character	5 fixed	Y	
Quantity	number of pieces of equipment needed	Integer		N	

Equipment

Name	Description	Data Type	Data Length	Required	Default Value
SerialNumber	Serial Number printed on the equipment from production. This is the primary key.	character	25 variable	Y	
PurchaseDate	Date the piece of equipment is purchased	Date		N	
DateofLastMaintenance	Date of last maintenance	Date		N	
IdentificationNumber	This is a foreign key called IdentificationNumber from the EquipmentType table that lidentifies what type of equipment this unit is.	Character	3 fixed	Y	

WorkOrder

Name	Description	Data Type	Data Length	Required	Default Value
WorkOrderNumber	Work order number created automatically when a piece of equipment needs to be worked on. This is the primary key.	Auto Number		Y	
DateofOrder		Date		N	
DateCompleted	Date the work is completed and signed off after inspection.	Date		N	
ManufacturerID	This is a foreign key called ManufacturerID from the Manufacturer table that links the manufacturer of the equipment being worked on.	Auto Number		Y	
SerialNumber	This is a foreign key called SerialNumber from the Equipment table that links which piece of equipment is being worked on.	Character	25 variable	Y	

Vendor

Name	Description	Data Type	Data Length	Required	Default Value
VendorID	Unique ID given to each vendor upon initial order with vendor. No duplicates and it stays with the vendor. If they change names or merge, they get a new ID. This is the primary key.	Auto Number		Y	
CompanyName	Full legal name of the company	character	40 variable	N	
StreetAddress	Number and street name of the vender.	character	50 variable	N	
City	City name where the vendor is located.	character	30 variable	N	
State	State abbreviation where the vendor is located.	character	2 fixed	N	
Zip	Zip code where the vendor is located.	character	5 fixed	N	
PhoneNumber	10 digit code of the vendor's primary phone number.	character	10 fixed	N	
Email	The full primary email address of the vendor.	character	50 variable	N	
FaxNumber	10 digit code of the vendor's fax number.	character	10 fixed	N	

PurchaseOrder

Name	Description	Data Type	Data Length	Required	Default Value
OrderNumber	Unique ID given to each supply order upon creation. This is the primary key.	Auto Number		Y	
DatePlaced	Date the order is placed	Date	6 fixed	N	
DateReceived	Date the entire order is received. Does not exist if the order hasn't been received.	Date	6 fixed	N	
TotalCost	Total cost of order. This includes shipping and any fees.	Money		N	
CEmployeeID	This is a foreign key and is pulled from the CEmployeeID in the Clerk table.	Auto Number		Y	
VendorID	This is a foreign key and is pulled from the VendorID in the Vendor table.	Auto Number		Y	

Item

Name	Description	Data Type	Data Length	Required	Default Value
ItemID	Unique ID given to each item upon creation of an order for a new item. This is the primary key.	Auto Number		Y	
Descrip	Description of the item.	Character	50 variable	N	
StandardPrice	Price the item is marked in the store.	Money		N	
QuantityonHand	The count of the item that is in stock in the store.	integer		N	
ReorderPoint	The inventory count at which the item should be reordered.	integer		N	0

PurchaseOrderLine

Name	Description	Data Type	Data Length	Required	Default Value
POLineNumber	Unique ID given to each SOLine upon creation of a purchase order. No duplicates. This is the primary key.	Auto Number		Y	
ItemsOrdered	Name of item being ordered	character	10 variable	N	
ItemQuantity	quantity of item ordered	integer		N	
QuantityReceived	total number received	integer		N	
VendorCost	Cost charged by the vendor	Money		N	
ItemID	This is foreign key called ItemID from the Item table.	Auto Number		Y	
CEmployeeID	Unique ID that is a foreign key called CEmployeeID and taken from the Clerk table.	Auto Number		Y	

SaleTransaction

Name	Description	Data Type	Data Length	Required	Default Value
TransactionNumber	Unique ID given to each transaction upon creation. This is the primary key.	Auto Number		Y	
TDate	Date the transaction occurs.	Date	6 fixed	N	
ItemsPurchased	Name of item being ordered	character	10 variable	N	
UnitPriceCharged	Price charged per item	Money		N	
PurchaseTotal	Total sum of the cart, including taxes and fees if applied.	Money		N	
CEmployeeID	Unique ID of the clerk who is completing the transaction. It is a foreign key called CEmployeeID from the Clerk table.	Auto Number		Y	
MemberID	Unique ID of the member who is purchasing the items. It is a foreign key called MemberID from the Member table.	Auto Number		Y	

SaleOrderLine

Name	Description	Data Type	Data Length	Required	Default Value
SOLineNumber	Unique ID given to each SOLine upon creation. No duplicates. This is the primary key.	Auto Number		Y	
ItemQuantity	quantity of item purchased	integer		N	
PricePerUnit	price per unit of item	Money		N	
ItemID	This is the unique identification foreign key called ItemID and pulled from the Item Table.	Auto Number		Y	
TransactionNumber	This is the unique identification foreign key called TransactionNumber and pulled from the SaleTransaction Table.	Auto Number		Y	

Queries

```
--1.
CREATE VIEW MemberCourseInfo AS
SELECT Member.MemberID, Member.FirstName, Member.LastName,
MembershipType.LevelName, MembershipType.DiscountPercent,
Course.CourseName,
        Course.CourseFee AS StandardCourseFee, ((1-
MembershipType.DiscountPercent)*Course.CourseFee) AS DiscountedFeePaid
FROM Member INNER JOIN MembershipType
ON Member.LevelName = MembershipType.LevelName
INNER JOIN EnrollmentList
ON EnrollmentList.MemberID = Member.MemberID
INNER JOIN Course
ON EnrollmentList.CourseNumber = Course.CourseNumber;

--2.
CREATE VIEW InstructorClassSchedule AS
SELECT Employee.FirstName, Employee.LastName, Instructor.IEmployeeID,
Class.DaysHeld, Class.StartDate, Class.TimeHeld,
        COUNT(MemberJoins.MemberID) AS NumMemberEnrolled,
Classroom.ClassCapacity, Classroom.RoomNumber,
        (Classroom.ClassCapacity-COUNT(MemberJoins.MemberID)) AS SpotsLeft
FROM Employee INNER JOIN Instructor
ON Employee.EmployeeID = Instructor.IEmployeeID
INNER JOIN Class
ON Instructor.IEmployeeID = Class.IEmployeeID
INNER JOIN Classroom
ON Class.RoomNumber = Classroom.RoomNumber
LEFT OUTER JOIN MemberJoins
ON Class.ReferenceNo = MemberJoins.ReferenceNo
GROUP BY Class.ReferenceNo, Employee.FirstName, Employee.LastName,
Instructor.IEmployeeID, Class.DaysHeld,
        Class.StartDate, Class.TimeHeld, Classroom.ClassCapacity,
Classroom.RoomNumber;

--3.
CREATE VIEW SupplyOrderInfo AS
SELECT Clerk.CEmployeeID, Employee.FirstName, Employee.LastName,
PurchaseOrder.DatePlaced AS OrderDate,
```

```
        PurchaseOrder.DateReceived,
DATEDIFF(DAYOFYEAR,PurchaseOrder.DatePlaced, PurchaseOrder.DateReceived)
AS NumDaysTorReceiveOrder,
        PurchaseOrderLine.QuantityReceived, PurchaseOrderLine.ItemQuantity,
        (SUM(PurchaseOrderLine.QuantityReceived)-
SUM(PurchaseOrderLine.ItemQuantity)) AS QuantityNotReceived,
        Vendor.CompanyName, Vendor.PhoneNumber
FROM Employee INNER JOIN Clerk
ON Employee.EmployeeID = Clerk.CEmployeeID
INNER JOIN PurchaseOrder
ON Clerk.CEmployeeID= PurchaseOrder.CEmployeeID
INNER JOIN Vendor
ON PurchaseOrder.VendorID = Vendor.VendorID
LEFT OUTER JOIN PurchaseOrderLine
ON PurchaseOrder.OrderNumber = PurchaseOrderLine.POLineNumber
GROUP BY Clerk.CEmployeeID, Employee.FirstName, Employee.LastName,
PurchaseOrder.DatePlaced, PurchaseOrder.DateReceived,
        PurchaseOrderLine.QuantityReceived, PurchaseOrderLine.ItemQuantity,
Vendor.CompanyName, Vendor.PhoneNumber;

--4.
CREATE VIEW ClerkSalesAbvAvg AS
SELECT Clerk.CEmployeeID, Employee.FirstName, Employee.LastName,
SUM(SaleTransaction.PurchaseTotal) AS TotalSales,
        SaleTransaction.TDate,
        MAX(SaleTransaction.PurchaseTotal) AS TopSaleTotal,
COUNT(DISTINCT(TransactionNumber)) AS NumSalesTransactions,
        (COUNT(SaleTransaction.TransactionNumber) - (SELECT AVG(NumSales) AS
AvgNumOrders
                                                                FROM
(SELECT CEmployeeID, COUNT(TransactionNumber) AS NumSales

        FROM SaleTransaction

        GROUP BY CEmployeeID) AS OrderCountTbl)) AS NumSalesAbvAvg,
        (SUM(SaleTransaction.PurchaseTotal) - (SELECT AVG(Sales) AS
AvgNumOrders

        FROM (SELECT CEmployeeID, SUM(PurchaseTotal) AS Sales
```

```
                    FROM SaleTransaction

                    GROUP BY CEmployeeID) AS OrderCountTbl)) AS
TotalSalesAbvAvg
FROM Employee INNER JOIN Clerk
ON Employee.EmployeeID = Clerk.CEmployeeID
LEFT OUTER JOIN SaleTransaction
ON Clerk.CEmployeeID = SaleTransaction.CEmployeeID
GROUP BY Clerk.CEmployeeID, Employee.FirstName, Employee.LastName,
SaleTransaction.TDate
HAVING SUM(SaleTransaction.PurchaseTotal) > (SELECT AVG(Sales) AS
AvgNumOrders

        FROM (SELECT CEmployeeID, SUM(PurchaseTotal) AS Sales

                FROM SaleTransaction

                GROUP BY CEmployeeID) AS OrderCountTbl);

--5.
CREATE VIEW EquipmentTypesByCourse AS
SELECT EquipmentType.IdentificationNumber, EquipmentType.Description,
Manufacturer.ManufacturerID, Manufacturer.CompanyName,
        COUNT(Course.CourseNumber) AS NumCourses,
COUNT(DISTINCT(Equipment.SerialNumber)) AS NumUnitsEquipment,
        COUNT(DISTINCT(WorkOrder.WorkOrderNumber)) AS NumWorkOrdersByType
FROM Course INNER JOIN Class
ON Course.CourseNumber = Class.CourseNumber
INNER JOIN EquipmentUse
ON Class.ReferenceNo = EquipmentUse.ReferenceNo
INNER JOIN EquipmentType
ON EquipmentUse.IdentificationNumber = EquipmentType.IdentificationNumber
INNER JOIN Manufacturer
ON Manufacturer.ManufacturerID = EquipmentType.ManufacturerID
INNER JOIN Equipment
ON EquipmentType.IdentificationNumber = Equipment.IdentificationNumber
LEFT OUTER JOIN WorkOrder
ON Equipment.SerialNumber = WorkOrder.SerialNumber
```

```
GROUP BY EquipmentType.IdentificationNumber, EquipmentType.Description,
Manufacturer.ManufacturerID, Manufacturer.CompanyName;


--6.
CREATE VIEW MembersForEnrollmentSales AS
SELECT Course.Category, Course.CourseNumber, Course.CourseName,
Course.CourseFee, Member.MemberID,
        Member.FirstName, Member.LastName, Member. PhoneNumber
FROM Member LEFT OUTER JOIN EnrollmentList
ON Member.MemberID = EnrollmentList.MemberID
LEFT OUTER JOIN Course
ON EnrollmentList.CourseNumber = Course.CourseNumber
GROUP BY Course.Category, Course.CourseNumber, Course.CourseName,
Course.CourseFee, Member.FirstName,
        Member.MemberID, Member.LastName, Member.PhoneNumber
HAVING COUNT(Course.CourseNumber) >= 1;


--7.
CREATE VIEW CoursesByInstructor AS
SELECT Instructor.IEmployeeID, Employee.FirstName, Employee.LastName,
Qualification.CourseNumber, Course.CourseName,
        COUNT(Class.ReferenceNo) AS NumClasses, Qualification.DateQualified,
Class.StartDate
FROM Employee INNER JOIN Instructor
ON Employee.EmployeeID = Instructor.IEmployeeID
LEFT OUTER JOIN Qualification
ON Qualification.IEmployeeID = Instructor.IEmployeeID
LEFT OUTER JOIN Course
ON Qualification.CourseNumber = Course.CourseNumber
LEFT OUTER JOIN Class
ON Course.CourseNumber = Class.CourseNumber
GROUP BY Instructor.IEmployeeID, Employee.FirstName, Employee.LastName,
Qualification.CourseNumber,
        Course.CourseName, Qualification.DateQualified, Class.StartDate;
–8.
CREATE VIEW ItemInformation AS
SELECT SaleTransaction.TransactionNumber, Descrip , ItemsOrdered,  PricePerUnit,
PurchaseTotal AS TotalRevenue , TotalCost , PurchaseOrder.DatePlaced,
```

```
PurchaseOrder.DateReceived , ItemQuantity AS AmountOrdered , SUM(ItemQty) AS
TotalSold
FROM SaleOrderLine , SaleTransaction , Item , PurchaseOrderLine , PurchaseOrder
WHERE SaleTransaction.TransactionNumber = SaleOrderLine.TransactionNumber
AND SaleOrderLine.ItemID = Item.ItemID
AND PurchaseOrderLine.ItemID=Item.ItemID
AND PurchaseOrder.CEmployeeID = PurchaseOrderLine.CEmployeeID
GROUP BY Descrip , ItemsOrdered,  PricePerUnit,  PurchaseTotal , TotalCost ,
ItemQuantity , PurchaseOrder.DatePlaced,
PurchaseOrder.DateReceived ,SaleTransaction.TransactionNumber
```

–9.

```
CREATE VIEW MembershipSavings AS
SELECT Member.MemberID, FirstName, LastName, StreetAddress, City, StateAbv,
Zip, PhoneNumber, Member.LevelName, Sum(PurchasePrice) AS StandardTotal,
Sum(PurchasePrice)-Sum(PurchasePrice)*DiscountPercent AS DiscountTotal ,
DiscountPercent*Sum(PurchasePrice) AS Savings
FROM Member LEFT OUTER JOIN EnrollmentList
ON Member.MemberID = EnrollmentList.MemberID
LEFT OUTER JOIN MembershipType
ON Member.LevelName = MembershipType.LevelName
GROUP BY Member.MemberID, FirstName, LastName, StreetAddress, City, StateAbv,
Zip, PhoneNumber, Member.LevelName, DiscountPercent
```

–10.

```
CREATE VIEW TopClassPerCourse AS
SELECT CourseNumber, CourseName, Description, Category, Duration, CourseFee,
ReferenceNo, StartDate, DaysHeld, TimeHeld, IEmployeeID, RoomNumber,
SpacesLeft, MAX(NumOfEnrollments) AS NumOfEnrolls--, ClassCapacity-
MAX(NumOfEnrollments) As SpacesLeft
FROM(SELECT Course.CourseNumber, CourseName, Description, Category, Duration,
CourseFee, ReferenceNo, StartDate, DaysHeld, TimeHeld, IEmployeeID,
Classroom.RoomNumber, Count(MemberID) AS NumOfEnrollments, ClassCapacity-
Count(MemberID) As SpacesLeft
        FROM Class, Course, EnrollmentList, Classroom
        WHERE Class.CourseNumber = Course.CourseNumber
        AND EnrollmentList.CourseNumber = Course.CourseNumber
```

```
        AND Class.RoomNumber = Classroom.RoomNumber
        GROUP BY Course.CourseNumber, CourseName, Description, Category,
Duration, CourseFee, ReferenceNo, StartDate, DaysHeld, TimeHeld, IEmployeeID,
Classroom.RoomNumber, ClassCapacity) AS EnrollTbl
GROUP BY CourseNumber, CourseName, Description, Category, Duration,
CourseFee, ReferenceNo, StartDate, DaysHeld, TimeHeld, IEmployeeID,
RoomNumber, SpacesLeft
```

Example Reports

Courses By Member

LastName	FirstName	MemberID	LevelName	DiscountPercent	CourseName	StandardCourseFee	DiscountedFeePaid
Brown	Lyndon	1005	Silver	0.1	Werk It!	$15.00	13.5
Brown	Lyndon	1005	Silver	0.1	Sweat Fest	$30.00	27
Cunningham	Jacob	1006	Gold	0.2	FabFitFun	$15.00	12
Cunningham	Jacob	1006	Gold	0.2	Transform	$30.00	24
Hill	Naomi	1016	Gold	0.2	Sanity Session	$20.00	16
Mayer	Davis	1001	Platinum	0.3	Walk this Weigh	$30.00	21
Payne	Maya	1018	Platinum	0.3	Fab Abs	$30.00	21
Reed	Harold	1003	Silver	0.1	Power Hour	$10.00	9
Warren	Hailey	1010	Bronze	0	Shrink	$25.00	25
Wright	Maya	1004	Silver	0.1	Curl&Crunch	$20.00	18

Saturday, December 3, 2022 Page 1 of 1

This report shows the courses each member has enrolled in, the course price, the discount as a decimal, and the price the member paid for each course in which they have enrolled. The owner, managers, and finance/accounting/analytics (if these positions exist) would get this report.

InstructorClassSchedule

LastName	FirstName	IEmployeeID	DaysHeld	StartDate TimeHeld	Number of Member Enrolled	Class Capacity	Room Number	Spots Left
Barnett	Amanda	8	M,W,F	7/25/2022 10:00:00.00000	1	17	06	16
Lugo	Ariah	7	M,W,F	4/14/2022 12:00:00.00000	0	9	07	9
Lugo	Ariah	7	M	8/4/2022 09:00:00.00000	1	27	08	26
Lugo	Ariah	7	T,R	8/15/2022 14:00:00.00000	1	7	10	6
Pena	Walid	6	M,W,F	10/6/2022 08:00:00.00000	3	23	01	20
Pena	Walid	6	T,R	3/18/2022 08:00:00.00000	1	9	02	8
Pena	Walid	6	T,R	4/7/2022 10:00:00.00000	0	23	01	23
Ramos	Marwan	9	T	9/21/2021 12:00:00.00000	2	23	01	21
Ramos	Marwan	9	F	8/22/2022 16:00:00.00000	0	6	05	6
Stewart	Aston	17	M,W,F	9/17/2022 14:00:00.00000	1	9	07	8

Sunday, December 4, 2022 Page 1 of 1

The Instructor Class Schedule report shows a full list of classes each instructor is teaching and the number of participants enrolled as well as how many spots are left.

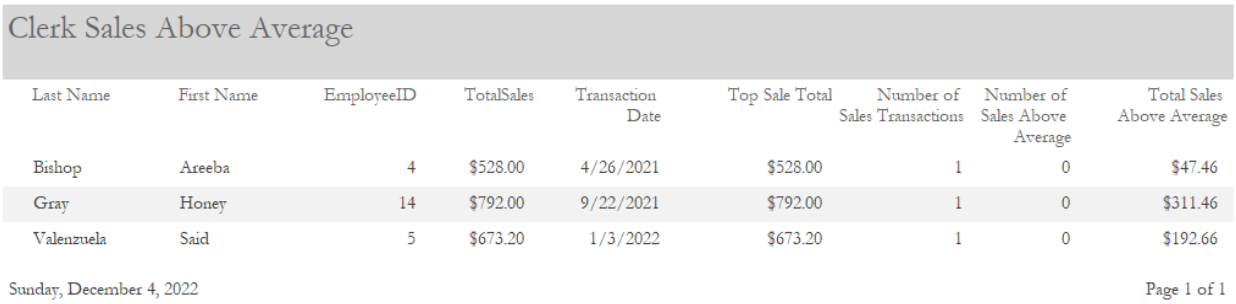
This would be given to the owners/managers to evaluate class schedules, instructor appeal, and class capacity/room allotment.

Supply Order Information

CompanyName	Clerk EmployeeID	Order Date	FirstName	LastName	Date Received	Number of Days To Receive Order	Quantity Received	Item Quantity	Quantity Not Received	Phone Number
Ab City	13	8/5/2022	Frederick	Foster	8/8/2022	3	152	152	0	6241097
Get Jacked	4	12/1/2022	Areeba	Bishop	12/12/2022	11	128	128	0	6322917
Get Jacked	10	8/9/2021	Ember	Fuller	10/20/2021	72	116	116	0	6322917
Muscle Mass	2	11/4/2022	Cheyanne	Patrick	11/11/2022	7	125	125	0	4349047
Olympic Fit	2	4/27/2022	Cheyanne	Patrick	6/7/2022	41	72	72	0	5934212
Olympic Fit	21	12/21/2021	Melanie	Clark	12/24/2021	3	38	38	0	5934212
Olympic Fit	21	2/21/2022	Melanie	Clark	4/11/2022	49	44	44	0	5934212
Plyometry	16	4/23/2021	Adrian	Taylor	8/6/2021	105	45	45	0	8388866
Pump Iron	4	4/6/2021	Areeba	Bishop	4/19/2021	13	96	96	0	5071041
Pump Iron	5	1/28/2021	Said	Valenzuela	3/24/2021	55	141	141	0	5071041

Sunday, December 4, 2022 Page 1 of 1

This report shows the item quantities, time each order takes to be received, and company information by order. This would be given to owners/managers as well as the clerks/people in charge of inventory. This would help evaluate vendors by their ability to complete orders completely and in a timely manner.



Clerk Sales Above Average

Last Name	First Name	EmployeeID	TotalSales	Transaction Date	Top Sale Total	Number of Sales Transactions	Number of Sales Above Average	Total Sales Above Average
Bishop	Areeba	4	$528.00	4/26/2021	$528.00	1	0	$47.46
Gray	Honey	14	$792.00	9/22/2021	$792.00	1	0	$311.46
Valenzuela	Said	5	$673.20	1/3/2022	$673.20	1	0	$192.66

Sunday, December 4, 2022 Page 1 of 1

The Clerk Sales Above Average report shows the number of sales, total sales by amount, amounts above average, and top sale total for the Clerks that have a total sales amount above the average for all of the clerks. This would be shared with the owners/managers to evaluate the clerks, which are similar to an in-house sales team. This could be used to given bonuses or recognition as well as evaluate if they are doing smaller sales transactions or upselling to help generate larger totals per transaction.

Equipment Types With Manufacturer

ManufacturerID	IdentificationNumber	Description	CompanyName	Number of Courses	Number of Units of Equipment	Number of Work Orders By Type
201	E01	bands	ABC Co.	1	1	1
202	E04	ball	123 Co.	1	1	1
203	E09	pull-up machine	WestRock	2	1	1
204	E08	elyptical	HIT Is Us	2	1	1
208	E10	row machine	SweatRUs	1	1	1
209	E02	treadmill	Everything Is Bigger	2	1	1
210	E05	shoulder press	Meat Heads	1	1	1

Sunday, December 4, 2022 Page 1 of 1

This report shows the number units of equipment, the number of courses each equipment type is used, and number of work orders created by equipment type and manufacturer. This would go to the owners/managers and anyone in charge of ordering equipment and maintenance. This would be able to help evaluate which manufacturers produce equipment with less down time from maintenance and repairs as well as which equipment may need more units based on these results. It can also help track maintenance at a glance. For example, if there are 10 pieces of equipment and they need to be maintained once a month, there should be at least 10 work orders for those units this month.

Enrollment of Members for Sales

CourseNumber	LastName	FirstName	Category	CourseName	CourseFee	MemberID	PhoneNumber
C0001	Reed	Harold	Novice	Power Hour	$10.00	1003	666641859
C0002	Wright	Maya	Novice	Curl&Crunch	$20.00	1004	170271198
C0003	Cunningham	Jacob	Novice	FabFitFun	$15.00	1006	036280664
C0004	Payne	Maya	Intermediate	Fab Abs	$30.00	1018	196553594
C0005	Mayer	Davis	Advanced	Walk this Weigh	$30.00	1001	781091168
C0006	Warren	Hailey	Advanced	Shrink	$25.00	1010	723153433
C0007	Hill	Naomi	Novice	Sanity Session	$20.00	1016	334047938
C0008	Brown	Lyndon	Novice	Werk It!	$15.00	1005	093723729
C0009	Cunningham	Jacob	Intermediate	Transform	$30.00	1006	036280664
C0010	Brown	Lyndon	Intermediate	Sweat Fest	$30.00	1005	093723729

Sunday, December 4, 2022 Page 1 of 1

This report is very helpful to know where to spend time on sales promotions since it shows the members and phone numbers by course number and category. This would be given to the owner/managers as well as the sales team to define and drive the focus of future sales strategies.

Courses By Instructor Qualification

LastName	FirstName	CourseNumber	IEmployeeID	CourseName	Number of Classes	DateQualified	StartDate
Barnett	Amanda	C0001	8	Power Hour	1	7/25/2022	10/6/2022
Harper	Eleanor	C0009	19	Transform	1	9/17/2022	8/15/2022
Lloyd	Lana	C0001	18	Power Hour	1	9/17/2021	10/6/2022
Lloyd	Lana	C0010	18	Sweat Fest	1	9/9/2021	9/17/2022
Lugo	Ariah	C0003	7	FabFitFun	2	8/4/2022	4/7/2022
Lugo	Ariah	C0006	7	Shrink	1	8/15/2022	8/4/2022
Lugo	Ariah	C0010	7	Sweat Fest	1	4/14/2022	9/17/2022
Mcgregor	Pollyanna	C0005	3	Walk this Weigh	1	5/24/2021	4/14/2022
Mcgregor	Pollyanna	C0006	3	Shrink	1	10/29/2021	8/4/2022
Pena	Walid	C0001	6	Power Hour	1	4/15/2022	10/6/2022
Pena	Walid	C0003	6	FabFitFun	1	3/18/2022	4/7/2022
Pena	Walid	C0004	6	Fab Abs	1	4/7/2022	7/25/2022
Pena	Walid	C0007	6	Sanity Session	1	12/8/2021	9/21/2021
Ramos	Marwan	C0001	9	Power Hour	1	9/21/2021	10/6/2022
Ramos	Marwan	C0007	9	Sanity Session	1	8/22/2022	9/21/2021
Stewart	Aston		17		0		
Stewart	James	C0001	20	Power Hour	1	11/4/2020	10/6/2022
Waller	Nyle	C0001	1	Power Hour	1	3/11/2020	10/6/2022
Waller	Nyle	C0002	1	Curl&Crunch	1	3/31/2020	3/18/2022
Waller	Nyle	C0008	1	Werk It!	1	9/14/2020	8/22/2022

Sunday, December 4, 2022 Page 1 of 1

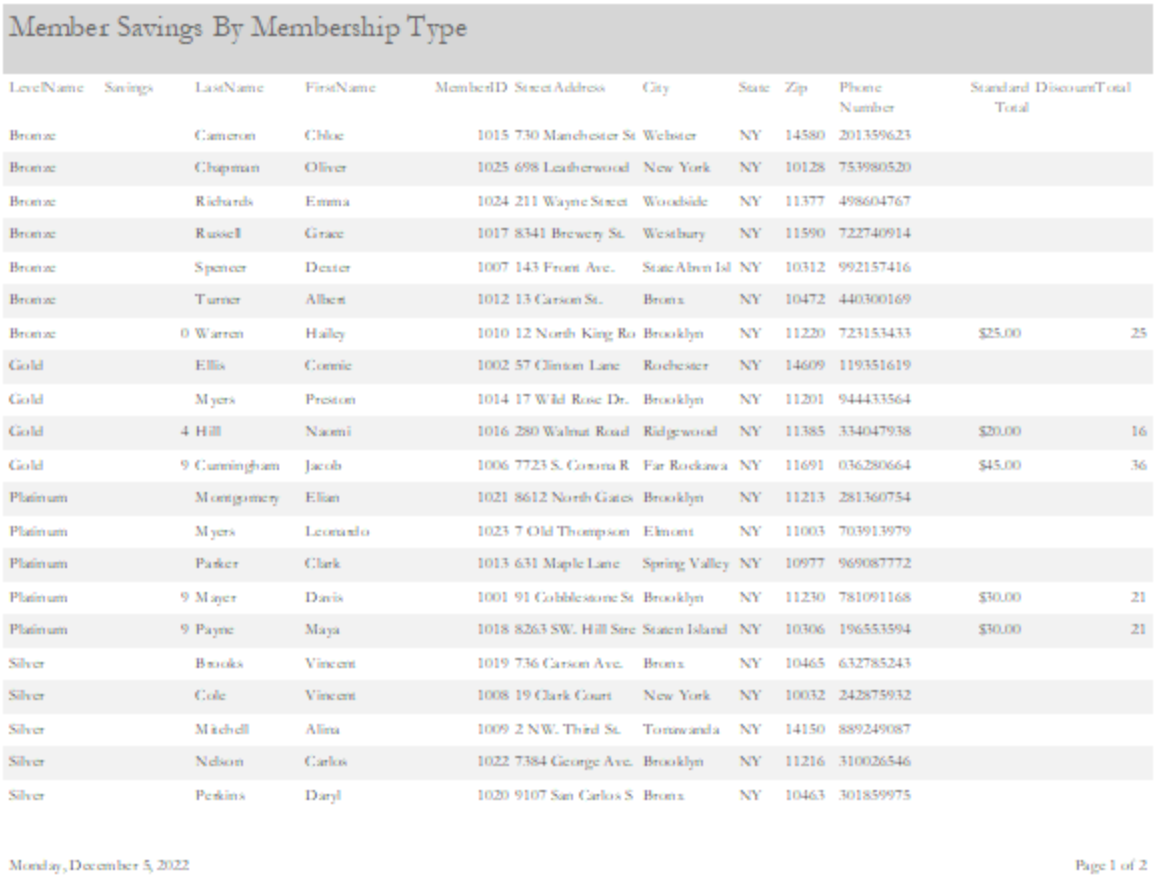
This report shows all the courses an instructor is qualified to teach, if any, and the date they qualified for that course. This would be given to the owners/managers as well as someone creating the class schedule to have a full list of which courses each instructor can teach.

dbo_Order Item Information By Date

DatePlaced	DateReceived	Transaction Number	Description	ItemsOrdered	PricePerUnit	TotalRevenue	TotalCost	Amount Ordered	Total Sold
4/6/2021	4/19/2021	3	Fitness Equipment	towel	$62.70	$188.10	$1,536.00	45	3
8/9/2021	10/20/2021	1	Fitness Equipment	vitamins	$17.60	$105.60	$6,844.00	152	6
8/9/2021	10/20/2021	9	Fitness Equipment	vitamins	$47.30	$236.50	$6,844.00	152	5
8/9/2021	10/20/2021	7	Fitness Equipment	vitamins	$74.80	$673.20	$6,844.00	152	9
4/27/2022	6/7/2022	1	Fitness Equipment	shirts	$17.60	$105.60	$4,248.00	96	6
4/27/2022	6/7/2022	9	Fitness Equipment	shirts	$47.30	$236.50	$4,248.00	96	5
4/27/2022	6/7/2022	7	Fitness Equipment	shirts	$74.80	$673.20	$4,248.00	96	9
4/27/2022	6/7/2022	3	Fitness Equipment	socks	$62.70	$188.10	$4,248.00	141	3
8/5/2022	8/8/2022	1	Fitness Equipment	pre-workout	$17.60	$105.60	$2,432.00	128	6
8/5/2022	8/8/2022	9	Fitness Equipment	pre-workout	$47.30	$236.50	$2,432.00	128	5
8/5/2022	8/8/2022	7	Fitness Equipment	pre-workout	$74.80	$673.20	$2,432.00	128	9
11/4/2022	11/11/2022	1	Fitness Equipment	shirts	$17.60	$105.60	$7,125.00	96	6
11/4/2022	11/11/2022	9	Fitness Equipment	shirts	$47.30	$236.50	$7,125.00	96	5
11/4/2022	11/11/2022	7	Fitness Equipment	shirts	$74.80	$673.20	$7,125.00	96	9
11/4/2022	11/11/2022	3	Fitness Equipment	socks	$62.70	$188.10	$7,125.00	141	3
12/1/2022	12/12/2022	3	Fitness Equipment	towel	$62.70	$188.10	$2,048.00	45	3

Monday, December 5, 2022 Page 1 of 1

This report shows a list of all of the transactions broken out by item. This shows revenue and cost, so it would benefit the financial department as well as the owners/managers.



Member Savings By Membership Type

LevelName	Savings	LastName	FirstName	MemberID	StreetAddress	City	State	Zip	Phone Number	Standard Total	DiscountTotal
Bronze		Cameron	Chloe	1015	730 Manchester St	Webster	NY	14580	2013596123		
Bronze		Chapman	Oliver	1025	698 Leatherwood	New York	NY	10128	7539805520		
Bronze		Richards	Emma	1024	211 Wayne Street	Woodside	NY	11377	498604767		
Bronze		Russell	Grace	1017	8341 Brewery St.	Westbury	NY	11590	722740914		
Bronze		Spencer	Dexter	1007	143 Front Ave.	State Alwn Isl	NY	10312	992157416		
Bronze		Turner	Albert	1012	13 Carson St.	Bronx	NY	10472	440300169		
Bronze	0	Warren	Hailey	1010	12 North King Ro	Brooklyn	NY	11220	723153433	$25.00	25
Gold		Ellis	Connie	1002	57 Clinton Lane	Rochester	NY	14609	119351619		
Gold		Myers	Preston	1014	17 Wild Rose Dr.	Brooklyn	NY	11201	944433564		
Gold	4	Hill	Naomi	1016	280 Walnut Road	Ridgewood	NY	11385	334047938	$20.00	16
Gold	9	Cunningham	Jacob	1006	7723 S. Corona R	Far Rockawa	NY	11691	0362806664	$45.00	36
Platinum		Montgomery	Elias	1021	8612 North Gates	Brooklyn	NY	11213	281360754		
Platinum		Myers	Leonardo	1023	7 Old Thompson	Elmont	NY	11003	703913979		
Platinum		Parker	Clark	1013	631 Maple Lane	Spring Valley	NY	10977	9690877772		
Platinum	9	Mayer	Davis	1001	91 Cobblestone St	Brooklyn	NY	11230	781091168	$30.00	21
Platinum	9	Payne	Maya	1018	8263 SW. Hill Stre	Staten Island	NY	10306	1965533594	$30.00	21
Silver		Brooks	Vincent	1019	736 Carson Ave.	Bronx	NY	10465	632785243		
Silver		Cole	Vincent	1008	19 Clark Court	New York	NY	10032	242875932		
Silver		Mitchell	Alina	1009	2 NW. Third St.	Tonawanda	NY	14150	889249087		
Silver		Nelson	Carlos	1022	7384 George Ave.	Brooklyn	NY	11216	310026546		
Silver		Perkins	Daryl	1020	9107 San Carlos S	Bronx	NY	10463	301859975		

Monday, December 5, 2022 Page 1 of 2

This report sorts members by type and details how much they pay based on their membership level compared to the standard price. This could be given to the owners/managers, financial, and sales department to analyze the levels and which are more likely to make purchases.

Top Class Per Course

Number Of Enrolls	Course Number	Reference Number	Course Name	Description	Category	Duration	CourseFee	StartDate	Days Held	TimeHeld	IEmployeeID	Room Number	SpacesLeft
1	C0001	CL001	Power H	Hour of HIT W	Novice	1	$10.00	10/6/2022	M,W,F	08:00:00.0000000	6	01	22
1	C0002	CL002	Curl&Cr	Abs and Bicepts	Novice	1	$20.00	3/18/2022	T,R	08:00:00.0000000	6	02	8
1	C0003	CL003	FabFitFu	getting started in	Novice	0.3	$15.00	4/7/2022	T,R	10:00:00.0000000	6	01	22
1	C0004	CL004	Fab Abs	getting fabulous	Intermed	1.5	$30.00	7/25/2022	M,W,F	10:00:00.0000000	8	06	16
1	C0005	CL005	Walk this	weigh ins and ac	Advance	1.5	$30.00	4/14/2022	M,W,F	12:00:00.0000000	7	07	8
1	C0006	CL006	Shrink	tailored workou	Advance	2	$25.00	8/4/2022	M	09:00:00.0000000	7	08	26
1	C0007	CL007	Sanity Se	healthy mindset	Novice	1	$20.00	9/21/2021	T	12:00:00.0000000	9	01	22
1	C0008	CL008	Werk It!	HIT full starter	Novice	2	$15.00	8/22/2022	F	16:00:00.0000000	9	05	5
1	C0009	CL009	Transfor	strength building	Intermed	1.5	$30.00	8/15/2022	T,R	14:00:00.0000000	7	10	6
1	C0010	CL010	Sweat Fe	cardio, cardio, a	Intermed	1.5	$30.00	9/17/2022	M,W,F	14:00:00.0000000	17	07	8

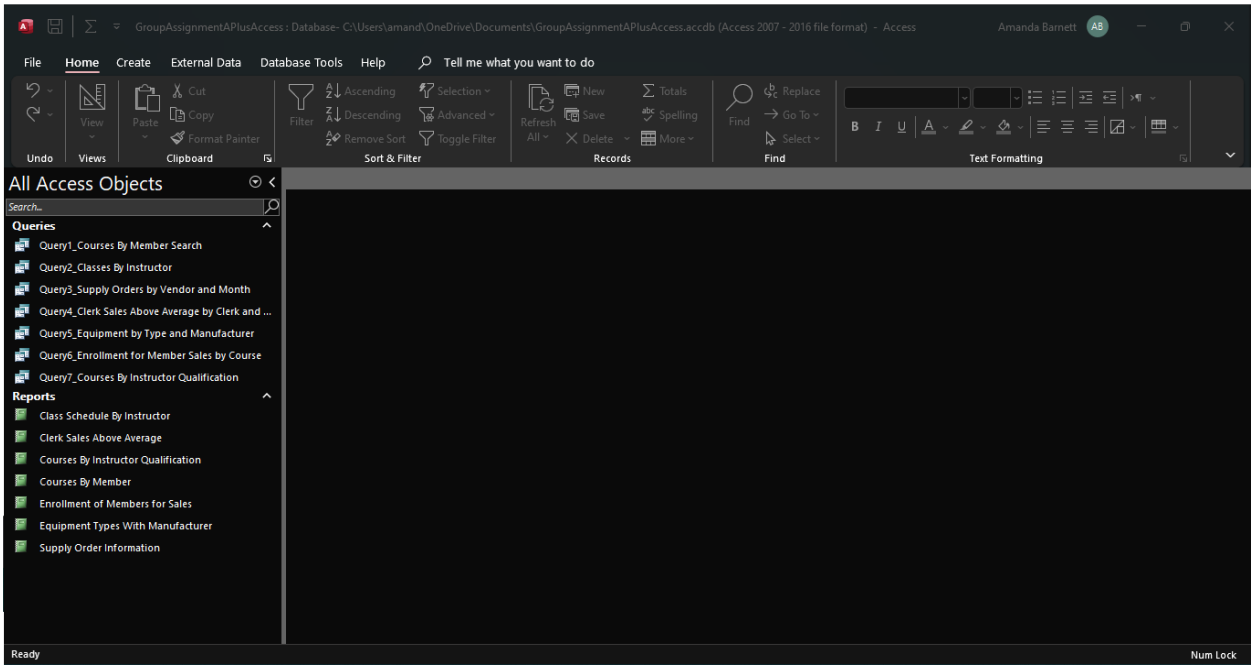
Monday, December 5, 2022 Page 1 of 1

This report shows the courses with the highest levels of enrollment. This should be given to the owners/managers and anyone involved in class scheduling. This report would be great in assessing if more classes should become available for a course and when the optimal times for classes are during the week.

Using this System

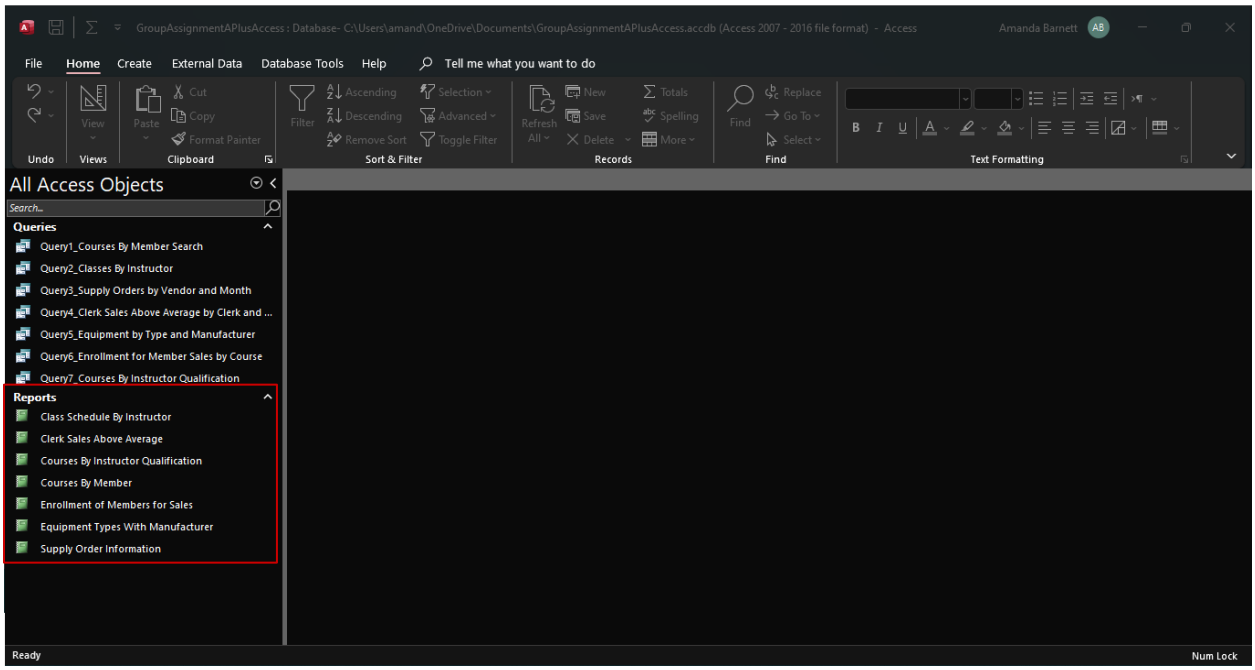
View When Opening the System

The below graphic is what you'll see when you open the database in Access.

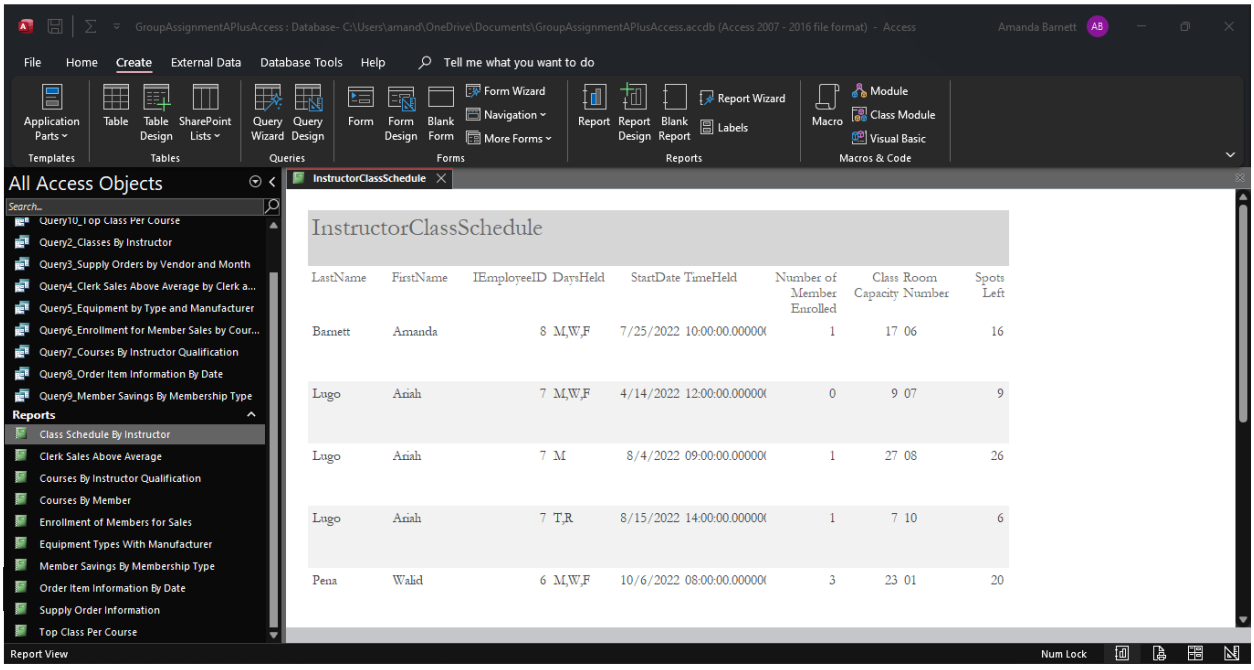


To View Reports

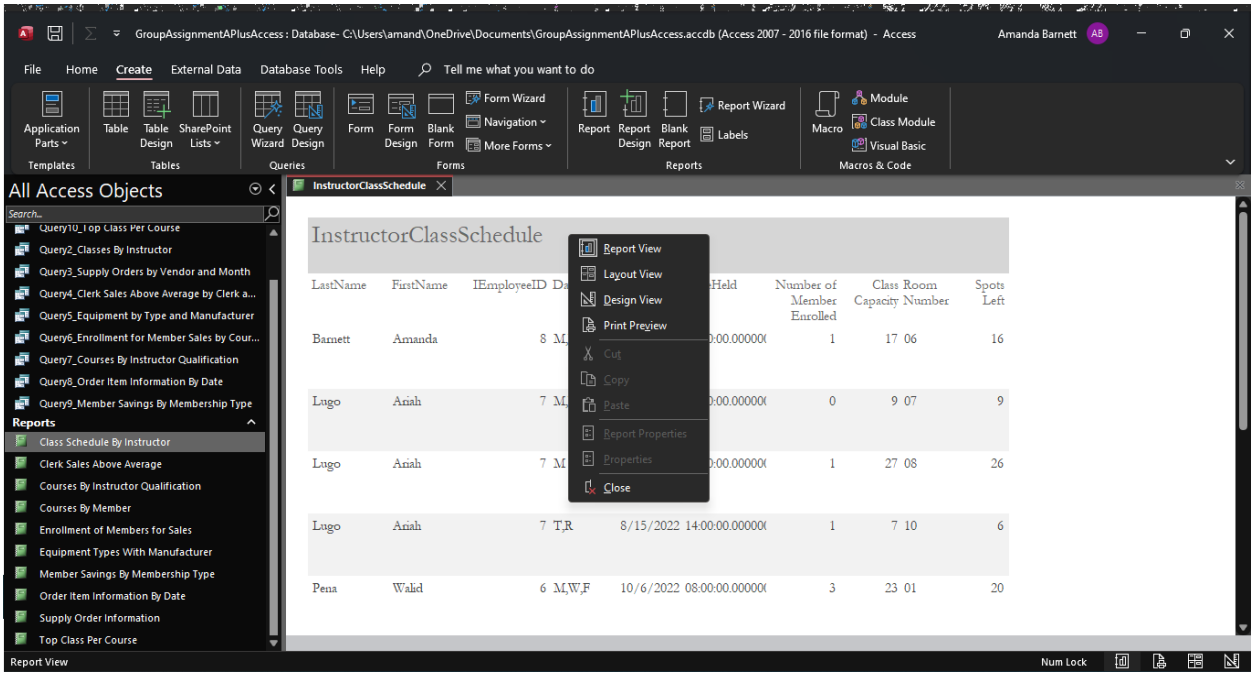
1. Select the report you would like to view.



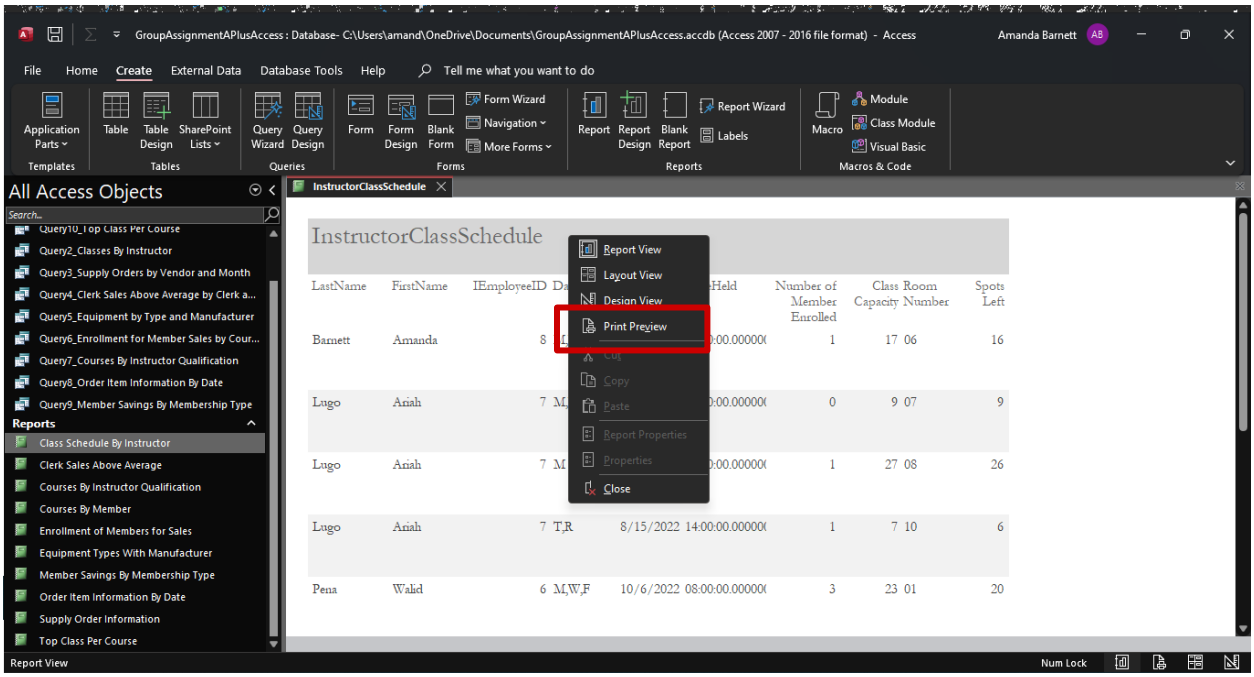
2. The report will populate. Review the needed information.



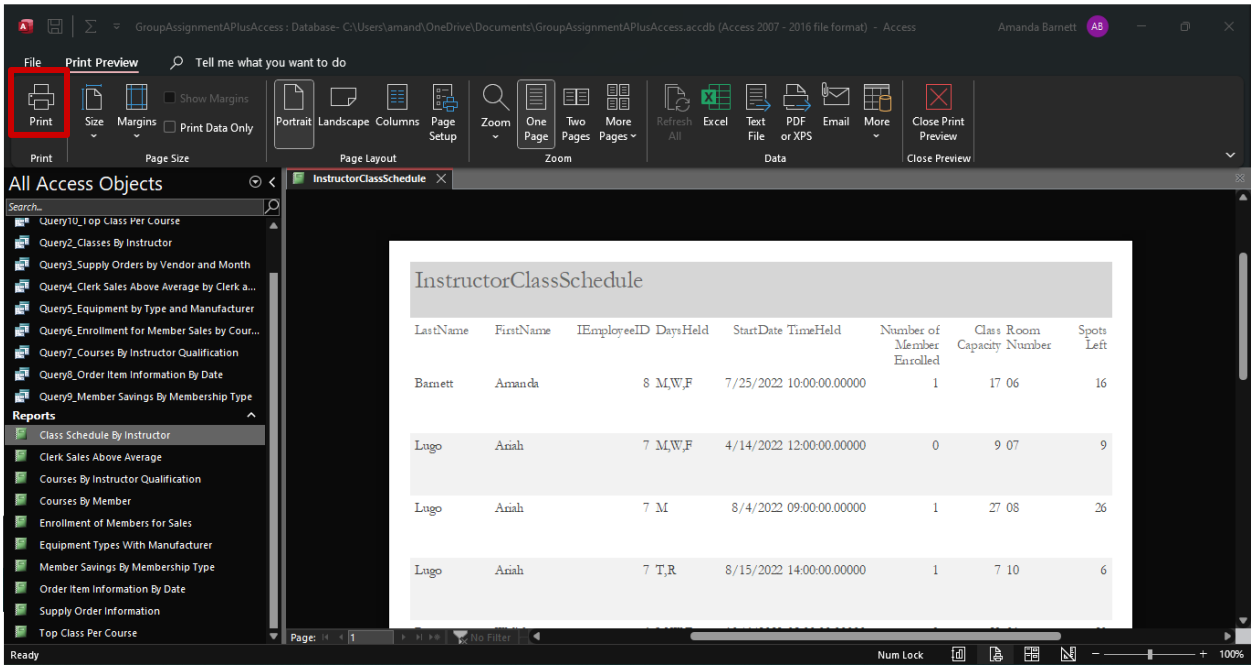
3. To Print this report, right click on the report.



4. Select Print Preview



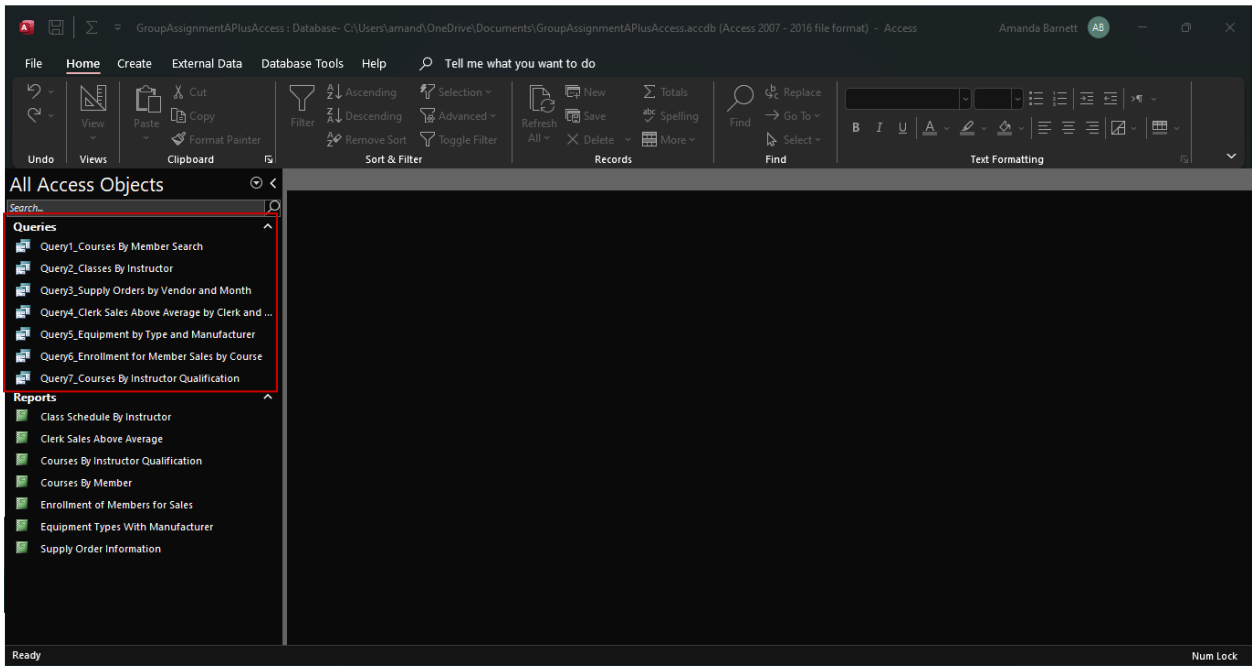
5. Select Print and follow normal computer prompts.



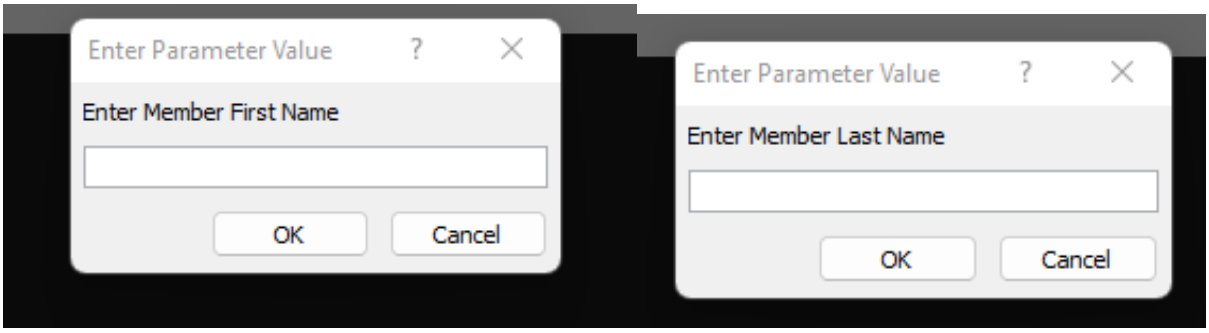
6. To compare two reports, To View Reports steps 1 and 2 and toggle back and forth.

To Use A Query

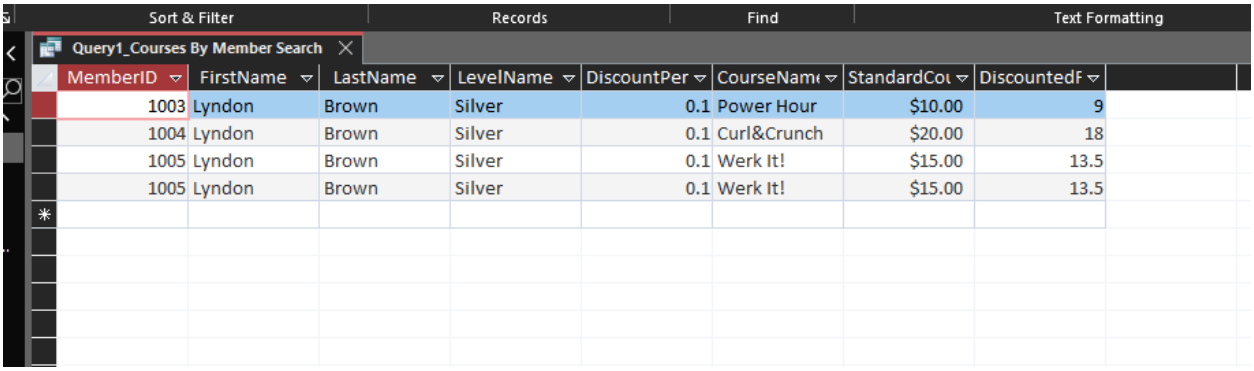
1. Select the query you want to view.



2. Add in the necessary information. You will be prompted, but below is a list for preparation.
    a. Query 1 needs the member's first and last name.
    b. Query 2 needs the instructor's first and last name as well as the numeric month the class starts.
    c. Query 3 needs the company's name and the numeric month of the order(s).
    d. Query 4 needs the clerk's first and last name and the date range. You'll enter the date at the start of the range first and the date of the end of the range after.
    e. Query 5 needs the equipment type and manufacturer number.
    f. Query 6 needs the course number and the category.
    g. Query 7 needs the instructor's first and last name as well as the course number.
    h. Query 8 needs
    i. Query 9 needs
    j. Query 10 needs



3. Hit enter or ok after each criteria.
4. Your results will populate.



MemberID	FirstName	LastName	LevelName	DiscountPer	CourseNam	StandardCo	DiscountedF
1003	Lyndon	Brown	Silver	0.1	Power Hour	$10.00	9
1004	Lyndon	Brown	Silver	0.1	Curl&Crunch	$20.00	18
1005	Lyndon	Brown	Silver	0.1	Werk It!	$15.00	13.5
1005	Lyndon	Brown	Silver	0.1	Werk It!	$15.00	13.5

5. You can exit from this view.
6. If you'd like to compare, repeat the To Use A Query instructions again without exiting and you can toggle back and forth between the two or more results.