

INMT 342
FINAL PROJECT
Section 3 Group 6



Kate Hudson, Ella Scholz, Kendal Ketron, and Audrey Rawlins

Table Of Contents

Table Of Contents	2
Executive Overview	3
Logical Data Model:	6
Conceptual Data Model	8
Data Dictionary	8
Create Statements	12
Insert Statements	18
Update Statements:	26
SQL Commands To Create Views	30
SQL Commands to Create Queries in Access	32
Descriptions of Prototype Elements	34
Instructions On How To Use Front-End Software	36
Sample Outputs	43

Executive Overview

Rise-and-Grind Fitness recently consulted with our group to design, assist and implement a database prototype to help with their business. The system we have designed will be able to capture information and data about members, employees, sales, orders, vendors, equipment, fitness courses, etc. As the business continues to grow, our vision for the database system is to help make the business run much more efficiently and effectively. Currently, our group is still in the early stages of designing a database. We have created a conceptual and logical diagram that will later be utilized to help build the physical database.

The relational database will capture and support many major business activities. For example, membership is one major business activity that the system will be capable of supporting in order to help management keep track of personal information and membership type of the member. Since Rise-and-Grind offers three different types of membership, it is important for this database system to track membership to easily identify the appropriate fees for each member. Another major business activity the database will be able to support is data about employees. By including employee information in the database system, top management of Rise-and-Grind Fitness will be able to clearly identify information about each employee, their position and their qualifications.

Fitness courses will also be tracked and supported through the database system. Since there are over thirty courses that Rise-and-Grind offers, the database will be able to clearly track data about the course and member involvement. This information will be helpful for management to make future decisions about which courses to offer and which ones may need to be retired. Course description, equipment, capacity, classroom size and time will all be captured. In addition, the fitness courses will need to be supported by the database system because they are a complex part of the business. For instance, each course can have multiple classes and are taught in a specific classroom, and fitness instructors teach the different courses and need certain qualifications in order to teach a class. As a result, Rise-and-Grind courses need to be modeled and implemented in the database since multiple parts of the business intertwine. In addition, the retail store will be supported in the database system. The retail store provides merchandise only

for its members. The database system needs to monitor the retail store because management will need to know when to restock the inventory and to keep track of sales and vendors.

In order to create and develop this system, our team made some assumptions regarding Rise-and-Grind's business model. For instance, we assumed that fitness clothing & nutritional supplements fall into the description of inventory. We made this assumption to clearly note that clothing and supplements are a part of the retail store. Another assumption we made is that every employee has their own Employee ID and respective ID number. There are no duplicates for Employee ID. We also assumed that every person taking a course is a member. For example, a person cannot come to Rise-and-Grind to just workout. He or she must have a membership to take courses, shop at the retail store and utilize the workout equipment.

Some assumptions we made about the classes is that a new class that has just become offered can have zero people enrolled. A class can exist without having any members; however, a class is dependent on a course and cannot exist without a course. We also assumed that a class is assigned to only one classroom and cannot be offered in any other classroom. Even though a class cannot exist without a course, a course can exist without a class being on the schedule. For example, if there are no current instructors available to teach a class one week, the class will not be on the schedule. However, the course can still be offered when instructors do become available. We also made some assumptions about Rise-and-Grind's equipment. We assume that every equipment type has at least one piece of equipment, and each equipment type is assigned to one course. If a piece of equipment needs to be worked on, then every work order is associated with one manufacturer; however, some manufacturers do not have work orders. We also assumed that price per unit is a derived value from membership discounts and standard price. and will be stored in the implementable model. Each member can also place multiple purchases. The last assumption we made is that only one vendor is associated with a supply order.

For this phase in developing the database, we have modeled the client's future system by creating a conceptual model diagram. The conceptual model shows all the business activities that the system will support. Within each business activity, we have listed the attributes that describe and help identify these sections of the business and have shown the relationships between the

business activities. In this phase, we also developed a logical model that will show the tables that will be needed to create the physical model. The logical model will identify the primary keys (or composite primary keys), the foreign keys and all the attributes. We also have created a data dictionary that defines all the entities in the conceptual model. The data dictionary shows the field name, data type, field size, description, primary key and/or foreign key designation, input mask, any constraints on values, foreign key referential integrity behavior and any other field properties. The data dictionary will help show how the database will be structured and the formatting of the information. We will also create mockup reports that the system will be able to create. For example, we will generate sale orders, vendor orders, membership contracts, etc. to show how the database will design these reports.

For the next phase of the project, we will develop an application prototype. This prototype will be using SQL server and front-end software. We will demonstrate the command of tables and the data by writing queries and reports. We will follow the wishes of the client and make this database design to appease their style. We will also incorporate documentation into our final deliverable that will include a user's manual and updated documents from our first deliverable we have presented. The manual will include a table of contents, SQL commands, descriptions of prototype elements, a "how to" document and an appendix containing sample outputs. Finally, we will include a prototype demonstration. All of Rise-and-Grind's top management will be able to see our group present our prototype. We will explain our database, the way we approached and thought about designing this database and give a detailed explanation on how to operate and implement this system smoothly. We are extremely grateful for this opportunity to work on this project and look forward to creating this database.

Logical Data Model:

EquipmentType(**EquipmentID**, Description, ManufactureModelNo, ReccomendedMaintenanceInterval, ***ManufacturerID***)

Equipment(**EquipmentSerialNumber**, ***EquipmentID***, PurchaseDate, DateofLastMaintenance)

WorkOrder(**WorkOrder#**, ***EquipmentSerialNumber***, ***ManufacturerID***, DateofOrder, ***SerialNumber***, ***Manufacturer***, DateOfCompletion)

Manufacturer(**ManufacturerID**, ***Course#***, CompanyName, Street, City, State, Zip, Phone, Email)

Courses(**Course#**, Subject, Course_name, Description, Category, Duration, Course_Fee, ***InstructorID***)

Class(**ReferenceNo**, ***Classroom#***, Start_date, Class_Held, Time_Period, ***MembersEnrolled***, ***Employee_ID***, ***InstructorID***)

Classroom(**Classroom#**, class_capacity)

PayFee(**ReferenceNo**, **Member_ID**, Paid)

Member(**Member_ID**, FirstName, LastName, Street, City, State, Zip, Phone_number, Gender, Birth_date, Date_joined, ***Membership_type***)

MembershipType(**MemberTypeID**, ***MemberID***, MembershipType, Fee, Discount)

Qualification(**Date_qualified**, ***InstructorID***, ***Course#***)

Employee(**Employee_ID**, FirstName, LastName, Street, City, State, Zip, Phone_Number, Email, Hire_Date, Status, EmployeePosition)

AdministrativePersonnel(**AdminID**, ***Employee_ID***, Salary)

SalesClerk(**SalesClerkID**, ***Employee_ID***, Salary)

Instructor(**InstructorID**, ***Employee_ID***, Salary)

SupplyOrder(**SupplyOrderID**, **DatePlaced**, DateRecieved, ***VendorID***, ***SalesClerkID***)

Vendor(**VendorID**, CompanyName, Street, City, State, Zip, Phone, Email, FaxNumber)

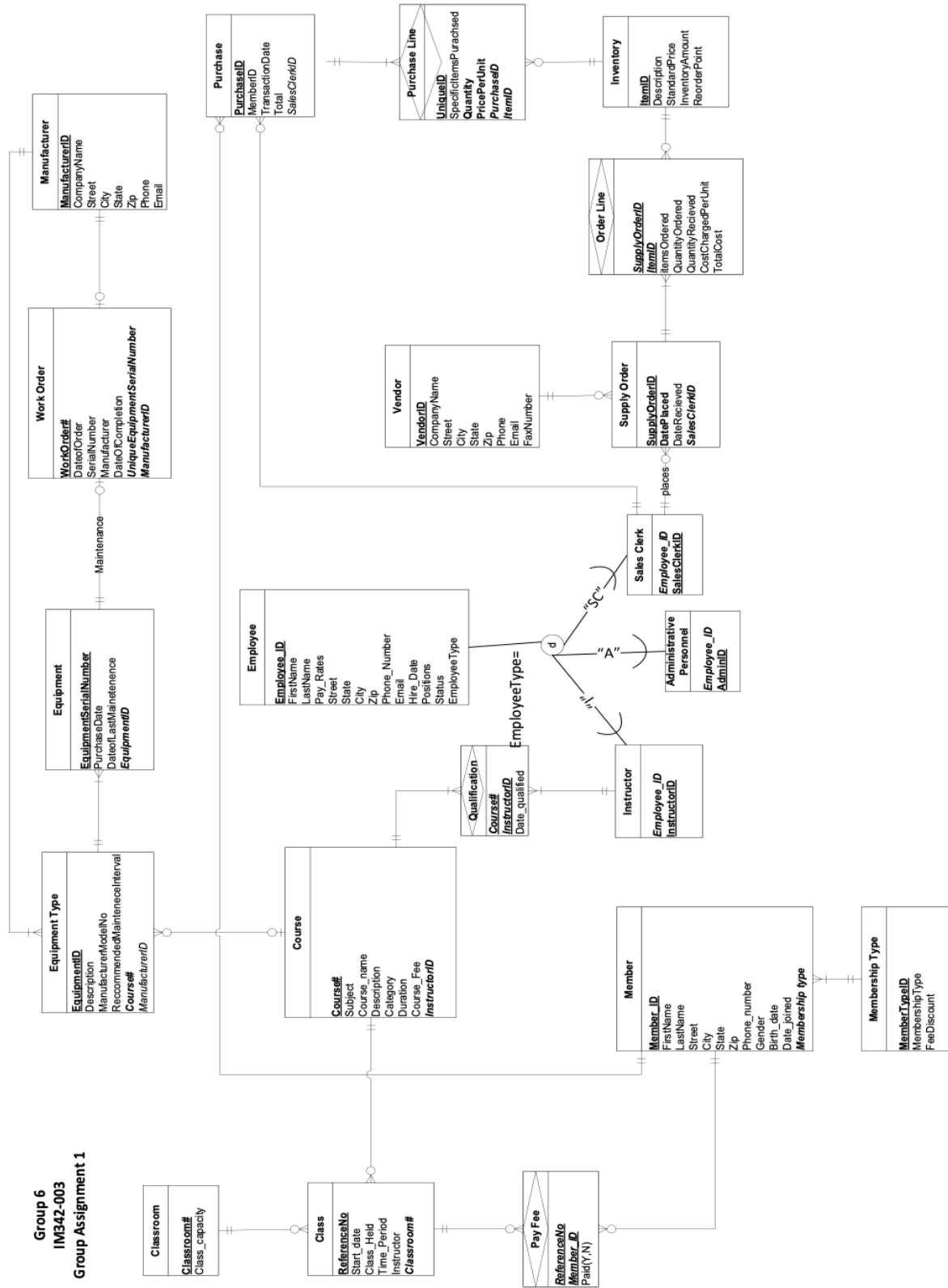
OrderLine(**SupplyOrderID**, ***ItemID***, itemsOrdered, QuantityOrdered, QuantityRecieved, CostChargedPerUnit, TotalCost)

Inventory(**ItemID**, ItemDescription, StandardPrice, InventoryAmount, ReorderPoint)

PurchaseLine(UniqueID, SpecificItemsPurchased, Quantity, PricePerUnit, *PurchaseID*, *ItemID*)

Purchase(PurchaseID, *MemberID*, *SalesClerkID*, TransactionDate, Total)

Conceptual Data Model



Data Dictionary

Table: Administrative Personnel

Field Name	Data Type	Description(Optional)	Field Size	Format (if applicable)	Input Mask	Default Value	Validation	Required
AdminID	Auto Number	PK. Unique sequential number of admin.	9					Y
Employee_ID	Auto Number	FK, refers to Employee_ID in Employee table. Unique sequential number of employee who is administrative personnel.	9					Y

Table: Class

Field Name	Data Type	Description(Optional)	Field Size	Format (if applicable)	Input Mask	Default Value	Validation	Required
ReferenceNo	Auto Number	PK. Unique sequential number.	9					Y
Classroom#	Auto Number	FK, references Classroom# in the Classroom table. Indicates classroom number where class is held.	1					Y
Start_date	Date	Class start date.	10	99/99/9999				N
Class_Held	Short Text	Day(s) of week class is held.	2				Must be M, T W TH, or F	N
Time_Period	Time	Time period of class.	11	(00:00-00:00)				N

Table: Classroom

Field Name	Data Type	Description(Optional)	Field Size	Format (if applicable)	Input Mask	Default Value	Validation	Required
ClassroomNumber	Auto Number	PK. Unique number assigned to each classroom.	1				Must be a number 1-7.	Y
Class_Capacity	Short Text	Maximum number of people assigned to classroom.	3					N

Table: Course

Field Name	Data Type	Description(Optional)	Field Size	Format (if applicable)	Input Mask	Default Value	Validation	Required
Course#	Auto Number	PK. Unique course number.	3					Y
Subject	Short Text	Subject of course.	20				Must be Yoga, Tai Chi, Weight Training, Aerobics, Nutrition, or Natural Medicine	N
Course_name	Short Text	Name of course.	50					N
Description	Short Text	Description of course.	250					N
Category	Short Text	Category of course.	20				Must be Novice, Intermediate, Advanced	N
Duration	Short Text	Duration of course.	2					N
Course_Fee	Short Text	Fee of course.	4					N
InstructorID	Auto Number	FK, references InstructorID in the Instructor table. Indicates unique sequential number of instructor that teaches course.	9					Y

Table: Employee

Field Name	Data Type	Description(Optional)	Field Size	Format (if applicable)	Input Mask	Default Value	Validation	Required
Employee_ID	Auto Number	PK. Employee's unique sequential number.	9					Y
First Name	Short Text	Employee's first name	20					N
Last Name	Short Text	Employee's last name	20					N
Qualifications	Short Text	Employee's specified qualifications	250					N
Pay_Rates	Currency	Employee's pay rates	7					N
Street	Short Text	Employee street name	30					N
City	Short Text	Employee city name	30					N
State	Short Text	Employee state name	30					N
Zip	Short Text	Employee zip code	6		999999			N
Phone_Number	Short Text	Employee phone number	10		(999)-999-9999			N
Email	Short Text	Employee email	30					N
Hire_Date	Date	Employee's date of hire	10	99/99/9999	Default: Today's date		Date_created <= today's date	N
Positions	Short Text	Employee's positions	30					N
Status	Short Text	Employee status	7					N
EmployeeType	Short Text	Type of employee	24					N

Table: Equipment

Field Name	Data Type	Description(Optional)	Field Size	Format (if applicable)	Input Mask	Default Value	Validation	Required
EquipmentSerialNumber	Auto Number	PK. Unique sequential number of equipment.	9					Y
EquipmentID	Auto Number	Unique equipment type ID number.	9					Y
PurchaseDate	Date	Date equipment was purchased.	10	99/99/9999	Default: Today's date		Date_created <= today's date	N
DateofLastMaintenance	Date	Date of last equipment maintenance.	10	99/99/9999			Date_created <= today's date	N

Table: Equipment Type

Field Name	Data Type	Description(Optional)	Field Size	Format (if applicable)	Input Mask	Default Value	Validation	Required
EquipmentID	Auto Number	PK. Unique equipment type ID number.	9					Y
Description	Short Text	Description of equipment type.	250					N
ManufacturerModelNumber	Short Text	Manufacturer model number.	9		CCC999			N
Interval	Short Text	Recommended maintenance time interval.	2					N
Course#	Auto Number	FK. Unique course number.	3					Y
ManufacturerID	Auto Number	FK. Unique sequential number of manufacturer.	9					Y

Table: Instructor								
Field Name	Data Type	Description(Optional)	Field Size	Format (if applicable)	Input Mask	Default Value	Validation	Required
InstructorID	Auto Number	PK. Unique sequential number of instructor.	9					Y
		FK, refers to EmployeeID in Employee table.						
Employee_ID	Auto Number	Indicates unique sequential employee number who is also an instructor.	9					Y

Table: Inventory								
Field Name	Data Type	Description(Optional)	Field Size	Format (if applicable)	Input Mask	Default Value	Validation	Required
ItemID	Auto Number	PK. Unique sequential number of item.	9					Y
Description	Short Text	Description of item.	250					N
StandardPrice	Currency	Standard price of item.	9					N
InventoryAmount	Int	Quantity of inventory on hand.	10					N
ReorderPoint	Int	Minimum quantity of inventory that should be kept on hand.	10					N

Table: Manufacturer								
Field Name	Data Type	Description(Optional)	Field Size	Format (if applicable)	Input Mask	Default Value	Validation	Required
ManufacturerID	Auto Number	PK. Unique sequential number of manufacturer.	9					Y
		FK, refers to Course# in Course table. Indicates unique course number.						
Course#	Auto Number		9					N
CompanyName	Short Text	Name of company.	30					N
Address	Short Text	Manufacturer address.	30					N
Phone	Short Text	Manufacturer phone.	10		(999)-999-9999			N
Email	Short Text	Manufacturer email.	30					N

Table: Member								
Field Name	Data Type	Description(Optional)	Field Size	Format (if applicable)	Input Mask	Default Value	Validation	Required
Member_ID	Auto Number	PK. Unique sequential number.	9					Y
First Name	Short Text	Member first name.	20					N
Last Name	Short Text	Member last name.	20					N
Street	Short Text	Member street address.	30					N
City	Short Text	Member city.	30					N
State	Short Text	Member state.	30					N
Zip	Short Text	Member zip.	6		999999			N
Phone_number	Short Text	Member phone number.	10		(999)-999-9999			N
Gender	Short Text	Member gender.	10					N
Birth_date	Date	Member birth date.	10	99/99/9999			Date_created <= today's date	N
Date_joined	Date	Member join date.	10	99/99/9999	Default: Today's date			N
		FK, refers to Membership Type in the Membership Type table. Indicates type of membership.						
Membership type	Short Text		8					Y

Table: Membership Type								
Field Name	Data Type	Description(Optional)	Field Size	Format (if applicable)	Input Mask	Default Value	Validation	Required
MemberTypeID	Auto Number	PK.	3					Y
		FK, refers to MemberID in the Member table. Indicates unique sequential number of member.						
MemberID	Auto Number		9					Y
MembershipType	Short Text	Type of membership.	8				Must be Bronze, Silver, Gold, or Platinum	N
Fee	Currency	Fee of membership type.	7				Must be 50, 100, 200, or 300.	N
Discount	Int	Discount of membership type.	3				Must be 0%,10%, 20%, or 30%	N

Table: Order Line								
Field Name	Data Type	Description(Optional)	Field Size	Format (if applicable)	Input Mask	Default Value	Validation	Required
SupplyOrderID	Auto Number	FK, refers to SupplyOrderID in supply order table. Unique sequential number of supply order.	9					Y
		FK, refers to ItemID in Item table. Indicates unique sequential number of item in order.						
ItemID	Auto Number		9					Y
ItemsOrdered	Short Text	Description of items ordered.	3					N
QuantityOrdered	Int	Quantity of items ordered.	9			Default: Quantity of 1		N
QuantityReceived	Int	Quantity of items received.	9					N
CostChargedPerUnit	Currency	Cost charged per unit in USD.	9					N
TotalCost	Currency	Total cost of items ordered in USD.	9					N

Table: Pay Fee								
Field Name	Data Type	Description(Optional)	Field Size	Format (if applicable)	Input Mask	Default Value	Validation	Required
ReferenceNo	Auto Number	FK, references ReferenceNo in the Class table. References unique sequential number of class.	9					Y
		FK, references Member_ID in the Member table. Indicates unique sequential number of member paying.						
Member_ID	Auto Number		7					Y
Paid	Yes/No	Whether fee is paid.	3				Must be Y or N	N

Table: Purchase								
Field Name	Data Type	Description(Optional)	Field Size	Format (if applicable)	Input Mask	Default Value	Validation	Required
PurchaseID	Auto Number	PK. Unique sequential number of purchase.	7	000####				Y
MemberID	Auto Number	FK, refers to MemberID in Member table. Indicates unique sequential number of member making purchase.	9					Y
SalesClerkID	Auto Number	FK, refers to SalesClerkID in Sales Clerk table. Indicates unique sequential number of employee associated with purchase.	9					Y
TransactionDate	Date	Date of purchase transaction.	10	99/99/9999		Default: Today's date	Date_created <= today's date	N
Total	Currency	Purchase total in USD. Calculated field from Purchase Line table.	9					N

Table: Purchase Line								
Field Name	Data Type	Description(Optional)	Field Size	Format (if applicable)	Input Mask	Default Value	Validation	Required
PurchaseLineID	Auto Number	PK. Unique sequential number of purchase line.	9					Y
SpecificItemsPurchased	Short Text	Description of specific item(s) purchased.	50					N
Quantity	Int	Quantity of items purchased.	9			Default: Quantity of 1		N
PricePerUnit	Currency	Price per unit of item in USD. Stored value.	9					N
PurchaseID	Auto Number	FK, refers to PurchaseID in Purchase table. Indicates unique sequential number of purchase.	9					Y
ItemID	Auto Number	FK, refers to ItemID in the Item table. Indicates Unique sequential number of item in Purchase line.	9					Y

Table: Qualification								
Field Name	Data Type	Description(Optional)	Field Size	Format (if applicable)	Input Mask	Default Value	Validation	Required
Date_Qualified	Auto Number	Date instructor is qualified to teach course.	10	99/99/9999	Default: Today's date		Date_created <= today's date	Y
InstructorID	Auto Number	FK, refers to InstructorID in the Instructor table. Indicates unique sequential number of instructor qualified.	9					Y
Course#	Auto Number	FK, refers to Course# in Course table. Indicates unique course number associated with qualification.	3					Y

Table: Sales Clerk								
Field Name	Data Type	Description(Optional)	Field Size	Format (if applicable)	Input Mask	Default Value	Validation	Required
SalesClerkID	Auto Number	PK. Unique sequential number of sales clerk.	9					Y
Employee ID	Auto Number	FK, refers to Employee_ID in Employee table. Unique sequential number of employee who is a sales clerk.	9					Y

Table: Supply Order								
Field Name	Data Type	Description(Optional)	Field Size	Format (if applicable)	Input Mask	Default Value	Validation	Required
SupplyOrderID	Auto Number	PK. Unique sequential number of supply order.	9					Y
DatePlaced	Date	Date order was placed.	10	99/99/9999	Default: Today's date		Date_created <= today's date	Y
DateReceived	Date	Date order was received.	10	99/99/9999				N
VendorID	Auto Number	FK, refers to VendorID in Vendor table. Indicates unique sequential number of vendor associated with supply order.	9					Y
SalesClerkID	Auto Number	FK, refers to SalesClerkID in SalesClerk table. Indicates unique sequential number of sales clerk associated with supply order.	9					Y

Table: Vendor								
Field Name	Data Type	Description(Optional)	Field Size	Format (if applicable)	Input Mask	Default Value	Validation	Required
VendorID	Auto Number	PK. Unique sequential number of vendor.	9					Y
CompanyName	Short Text	Vendor company name.	30					N
Street	Short Text	Vendor street.	30					N
City	Short Text	Vendor city.	30					N
State	Short Text	Vendor state.	30					N
Zip	Short Text	Vendor zip.	6	999999				N
Phone	Short Text	Vendor phone.	10		(999)-999-9999			N
Email	Short Text	Vendor email.	30					N
FaxNumber	Short Text	Vendor fax number.	8					N

Table: Work Order								
Field Name	Data Type	Description(Optional)	Field Size	Format (if applicable)	Input Mask	Default Value	Validation	Required
SequentialNumber	Auto Number	PK. Unique sequential number of work order.	9					Y
EquipmentSerialNumber	Auto Number	FK, refers to UniqueEquipmentSerialNumber in Equipment table. Unique sequential number of equipment in the work order.	9					Y
ManufacturerID	Auto Number	FK, refers to ManufacturerID in Manufacturer table. Unique sequential number of manufacturer of work order.	9					Y
DateOfOrder	Date	Date work order is placed.	10	99/99/9999	Default: Today's date		Date_created <= today's date	N
DateOfCompletion	Date	Date work order is completed.	10	99/99/9999				N

Create Statements

Administrative Personnel	Create Table AdministrativePersonnel (AdminID Varchar(3) Primary Key, Employee_ID Int, Salary Decimal (6,0), Constraint AdminFK Foreign Key (Employee_ID) References Employee (Employee_ID) On Update Cascade On Delete No action);
Class	CREATE TABLE Class(ReferenceNO INT PRIMARY KEY, StartDate DATE, Class_Held VARCHAR(10) CHECK (Class_Held IN ('Monday', 'Tuesday','Wednesday','Thursday','Friday', 'Saturday', 'Sunday')), Time_Period Time(5), Course# INT NOT NULL, ClassroomNumber INT NOT NULL, CONSTRAINT Course#fk FOREIGN KEY (Course#) REFERENCES Course(Course#) ON UPDATE CASCADE, CONSTRAINT ClassroomNumberfk FOREIGN KEY (ClassroomNumber) REFERENCES Classroom(ClassroomNumber) ON UPDATE CASCADE) Alter Table Class Add Employee_ID INT Constraint EmployeeFK FOREIGN KEY (Employee_ID) REFERENCES Employee (Employee_ID) On Update No action Alter Table Class Add MembersEnrolled Int Alter Table Class Add InstructorID Varchar(3) Constraint Instructor_FK FOREIGN KEY (InstructorID) REFERENCES Instructor (InstructorID) On Update No action
Classroom	CREATE TABLE Classroom(ClassroomNumber INT NOT NULL CHECK (ClassroomNumber IN (1,2,3,4,5,6,7,8,9,10)), Class_Capacity INT NOT NULL, CONSTRAINT ClassroomNumber PRIMARY KEY(ClassroomNumber));
Course	CREATE TABLE Course(Course# INT PRIMARY KEY, Subject VARCHAR(30), Course_Name VARCHAR(30), Description VARCHAR(300), Category VARCHAR(20) Check (Category IN ('Novice', 'Intermediate',

	'Advanced')), Duration Decimal (3,0) Check (Duration > 0), Course_Fee MONEY, InstructorID Varchar(3) not null, CONSTRAINT InstructorIDFK FOREIGN KEY (InstructorID) REFERENCES Instructor(InstructorID) ON UPDATE CASCADE ON DELETE NO ACTION);
Employee	Create Table Employee (Employee_ID Int Not null, FirstName Varchar(20), LastName Varchar(30), Street Varchar(100), City Varchar(50), State Char(2), Zip Char(5), Phone_Number Char(12), Email Varchar(50), Hire_Date Date, Status Varchar(7) Check (Status In ('Current', 'Former')), EmployeePosition Varchar(24) Check (EmployeePosition In ('Instructor', 'Sales Clerk', 'Administrative Personnel')) Constraint EmployeePK Primary Key (Employee_ID))
Equipment	Create Table Equipment (EquipmentSerialNumber VARCHAR(4) Primary Key NOT NULL, EquipmentID VARCHAR(3) NOT NULL, PurchaseDate DATE, DateofLastMaintenance DATE, CONSTRAINT EquipmentIDfk FOREIGN KEY (EquipmentID) REFERENCES EquipmentType (EquipmentID) ON UPDATE CASCADE On Delete No Action);
Equipment Type	Create Table EquipmentType (EquipmentID VARCHAR(3) Primary Key NOT NULL, Description VARCHAR(50), ManufacturerModelNo VARCHAR(6), RecommendedMaintenanceInterval VARCHAR(10), Course# INT NOT NULL, CONSTRAINT Course#fk_equiptype FOREIGN KEY (Course#) REFERENCES Course(Course#) ON UPDATE CASCADE On Delete No Action);
Instructor	Create Table Instructor (InstructorID Varchar(3) Primary Key,

	Employee_ID Int, Salary Decimal (5,0) Constraint InstructorFK Foreign Key (Employee_ID) References Employee (Employee_ID) On Update Cascade On Delete No action);
Inventory	Create Table Inventory (ItemID INT Primary Key, ItemDescription Varchar(50), StandardPrice Decimal (3,0) Check (StandardPrice > 0), Amount Int check (Amount > 0), ReorderPoint Int Check (ReorderPoint >= 20));
Manufacturer	Create Table Manufacturer (ManufacturerID VARCHAR(3) Primary Key, Course# INT, CompanyName VARCHAR(30), Street VARCHAR (30), City VARCHAR (30), State VARCHAR (20), Zip CHAR (5), Phone CHAR (12), Email VARCHAR (30), Constraint Course#_FK Foreign Key (Course#) References Course (Course#) On update Cascade On Delete No action);
Member	CREATE TABLE Member(Member_ID INT PRIMARY KEY, MemberFirstName VARCHAR(15), MemberLastName VARCHAR(15), Street VARCHAR(20), City VARCHAR(15), State CHAR(2), Zip CHAR(5), Phone_Number CHAR(10), Gender VARCHAR(6) CHECK (Gender IN ('Female', 'Male')), Birth_Date DATE CHECK (Birth_Date >= GETDATE()), Date_Joined DATE Check (Date_Joined >= GETDATE()) NOT NULL, MembershipTypeID INT FOREIGN KEY REFERENCES MembershipType(MembershipTypeID) ON UPDATE CASCADE NOT NULL);
MembershipType	CREATE TABLE MembershipType(MembershipTypeID INT PRIMARY KEY, MembershipType VARCHAR(10) CHECK (MembershipType IN

	('Bronze','Silver','Gold','Platinum')) NOT NULL, Fee MONEY CHECK (Fee IN (50, 100, 200, 300)), Discount INT CHECK (Discount IN (0, 10, 20, 30)));
Order Line	Create Table OrderLine(SupplyOrderID VARCHAR(4) NOT NULL, ItemID INT NOT NULL, ItemsOrdered VARCHAR(30), QuantityOrdered INT, QuantityReceived INT, CostChargedPerUnit MONEY, TotalCost MONEY, Constraint OrderLineID PRIMARY KEY (SupplyOrderID, ItemID), Constraint ItemID FOREIGN KEY(ItemID) REFERENCES Inventory(ItemID) ON UPDATE CASCADE, Constraint SupplyOrderID FOREIGN KEY(SupplyOrderID) REFERENCES SupplyOrder(SupplyOrderID) ON UPDATE CASCADE);
Pay Fee	CREATE TABLE PayFee(ReferenceNO INT NOT NULL, Member_ID INT NOT NULL, Paid CHAR(1) CHECK (Paid IN ('Y','N')) DEFAULT ('N'), Constraint PayFeeID PRIMARY KEY (ReferenceNO, Member_ID), Constraint ReferenceNOFK FOREIGN KEY (ReferenceNO) REFERENCES Class(ReferenceNO) ON UPDATE CASCADE, Constraint Member_IDFK FOREIGN KEY (Member_ID) REFERENCES Member(Member_ID) ON UPDATE CASCADE);
Purchase	Create Table Purchase (PurchaseID VARCHAR(3) NOT NULL Primary Key, Member_ID INT NOT NULL, Employee_ID INT NOT NULL, TransactionDate DATE, Total MONEY, Constraint PurchaseMemberFK Foreign Key (Member_ID) References Member (Member_ID) On Update Cascade On Delete No action, Constraint PurchaseEmpFK Foreign Key (Employee_ID) References Employee (Employee_ID) On Update Cascade On Delete No action);
Purchase Line	Create Table PurchaseLine (PurchaseLineID VARCHAR(3) NOT NULL,

	SpecificItemsPurchased VARCHAR(30), Quantity INT, PricePerUnit MONEY, Employee_ID INT NOT NULL, PurchaseID VARCHAR(3) NOT NULL, ItemID INT NOT NULL, Constraint PurchaseLinePK Primary Key (PurchaseLineID, PurchaseID, ItemID), Constraint PurchaseLineEmpFK Foreign Key (Employee_ID) REFERENCES Employee(Employee_ID) ON UPDATE Cascade ON DELETE NO action, Constraint PurchaseLinePurchFK Foreign Key (PurchaseID) References Purchase (PurchaseID) ON UPDATE NO ACTION, Constraint PurchaseLineItemFK Foreign Key (ItemID) References Inventory (ItemID) ON UPDATE NO ACTION);
Qualification	Create Table Qualification (InstructorID Varchar(3), Course# Int NOT NULL, Date_Qualified Date NOT NULL, Constraint QualificationPK Primary Key (InstructorID, Course#), Constraint QualificationInstructorFK Foreign Key (InstructorID) REFERENCES Instructor(InstructorID) ON UPDATE Cascade ON DELETE NO action, Constraint QualificationCourseFK Foreign Key (Course#) References Course (Course#) ON UPDATE NO ACTION);
Sales Clerk	Create Table SalesClerk (SalesClerkID Varchar(4) Primary Key, Employee_ID Int, Salary Decimal (6,0), Constraint SalesClerkFK Foreign Key (Employee_ID) References Employee (Employee_ID) On Update Cascade On Delete No action);
Supply Order	Create table SupplyOrder(SupplyOrderID VARCHAR(4) NOT NULL Primary Key, DatePlaced DATE NOT NULL, DateReceived DATE, VendorID INT NOT NULL, SalesClerkID VARCHAR(4) NOT NULL, Constraint SupplyOrderVendorFK Foreign Key (VendorID) References Vendor(VendorID)

	On Update Cascade On Delete No action, Constraint SupplyOrderSalesFK Foreign Key (SalesClerkID) References SalesClerk(SalesClerkID) On Update Cascade On Delete No action);
Vendor	Create table Vendor (VendorID Int NOT NULL Primary Key, CompanyName VARCHAR(30), Street VARCHAR(25), City VARCHAR(20), State VARCHAR(2), Zip VARCHAR(5), Phone VARCHAR(11), Email VARCHAR(30), FaxNumber VARCHAR(11));
Work Order	Create table WorkOrder (WorkOrder# VARCHAR(4) Primary key NOT NULL, EquipmentSerialNumber VARCHAR(4) NOT NULL, ManufacturerID VARCHAR (3) NOT NULL, DateofOrder DATE CHECK (DateofOrder<= getdate()) DEFAULT getdate(), DateofCompletion DATE Constraint EquipmentFK Foreign Key (EquipmentSerialNumber) REFERENCES Equipment (EquipmentSerialNumber) ON UPDATE CASCADE On Delete No Action, Constraint ManufacturerFK Foreign Key (ManufacturerID) REFERENCES Manufacturer(ManufacturerID) ON UPDATE No action);

Insert Statements

Administrative Personnel	Insert Into AdministrativePersonnel Values ('A01',4,105000) Insert Into AdministrativePersonnel Values ('A02',5,90000) Insert Into AdministrativePersonnel Values ('A03',24,70000) Insert Into AdministrativePersonnel Values ('A04',25,85000) Insert Into AdministrativePersonnel Values ('A05',12,100000) Insert Into AdministrativePersonnel Values ('A06',16,105000) Insert Into AdministrativePersonnel Values ('A07',26,75000) Insert Into AdministrativePersonnel Values ('A08',17,95000) Insert Into AdministrativePersonnel Values ('A09',18,80000) Insert Into AdministrativePersonnel Values ('A10',19,100000)
Class	INSERT INTO Class VALUES (1, '2022-09-15', 'Friday', '08:30', 2, 7); INSERT INTO Class VALUES (2, '2022-08-30', 'Wednesday', '06:30', 3, 1); INSERT INTO Class VALUES (3, '2022-06-25', 'Tuesday', '09:00', 4, 5); INSERT INTO Class VALUES (4, '2022-07-14', 'Monday', '10:30', 4, 4); INSERT INTO Class VALUES (5, '2022-08-12', 'Wednesday', '11:00', 6, 8); INSERT INTO Class VALUES (6, '2022-09-17', 'Tuesday', '11:00', 5, 6); INSERT INTO Class VALUES (7, '2022-06-15', 'Thursday', '12:30', 1, 2); INSERT INTO Class VALUES (8, '2022-07-23', 'Thursday', '10:00', 7, 7); INSERT INTO Class VALUES (9, '2022-06-30', 'Monday', '09:00', 9, 7); INSERT INTO Class VALUES (10, '2022-08-27', 'Saturday', '11:00', 2, 9);
Classroom	INSERT INTO Classroom VALUES (1, 20); INSERT INTO Classroom VALUES (2, 15); INSERT INTO Classroom VALUES (3, 30); INSERT INTO Classroom VALUES (4, 30); INSERT INTO Classroom VALUES (5, 30); INSERT INTO Classroom VALUES (6, 60); INSERT INTO Classroom VALUES (7, 10); INSERT INTO Classroom VALUES (8, 30); INSERT INTO Classroom VALUES (9, 30); INSERT INTO Classroom VALUES (10, 30);
Course	INSERT INTO Course VALUES (1, 'Yoga', 'Flow Yoga', 'A steady easy way to get your day started.', 'Intermediate', 60, 70, 'I01'); INSERT INTO Course VALUES (2, 'Yoga', 'Full-Body Yoga', 'An empowering way to strengthen and tone your body through fluid movement.', 'Advanced', 60, 80, 'I01'); INSERT INTO Course VALUES (3, 'Tai-Chi', 'ShadowBoxing', 'A cardio driven lesson proven to burn calories and increase endurance.', 'Advanced', 90, 90, 'I03'); INSERT INTO Course VALUES (4, 'Tai-Chi', 'ShadowBoxing', 'An introductory cardio driven lesson proven to burn calories and increase endurance', 'Novice', 90, 90, 'I03');

	INSERT INTO Course VALUES (5, 'Weight Training', 'Introductory Weight Training', 'An intro class to the fundamentals of proper weight training.', 'Novice', 120, 65, 'I04'); INSERT INTO Course VALUES (6, 'Weight Training', 'Advanced Weight Training', 'An emphasis on exponentially gaining strength and fine-tuning muscles.', 'Advanced', 120, 100, 'I04'); INSERT INTO Course VALUES (7, 'Aerobics', 'Aerobics', 'A vigorous course designed to improve endurance and mobility.', 'Advanced', 90, 75, 'I05'); INSERT INTO Course VALUES (8, 'Nutrition', 'Nutrition and Health', 'An in depth course of the benefits of proper nutrition and how to implement it into your daily life.', 'Novice', 60, 70, 'I06'); INSERT INTO Course VALUES (9, 'Natural Medicine', 'Spiritual Health', 'A look into the benefits of natural healing and techniques to cure.', 'Novice', 70, 95, 'I07'); INSERT INTO Course VALUES (10, 'Yoga', 'Strength Yoga', 'A course meant to empower and bring strength to mind and body.', 'Advanced', 90, 85, 'I01');
Employee	Insert Into Employee Values (01, 'Kate', 'Hudson', '1016 Hall Linden Dr.', 'Alcoa', 'TN', '37701', '865-111-1234', 'kh@gmail.com', '1/1/2022', 'Current', 'Instructor') Insert Into Employee Values (02, 'Ella', 'Scholz', '1234 Hill Rocky Rd.', 'Knoxville', 'TN', '37916', '865-551-2224', 'e.s11@gmail.com', '2/1/2022', 'Current', 'Sales Clerk') Insert Into Employee Values (03, 'Kendal', 'Ketron', '567 Merry Lane', 'Knoxville', 'TN', '37920', '865-121-1234', 'kk23@gmail.com', '1/15/2020', 'Former', 'Instructor') Insert Into Employee Values (04, 'Audrey', 'Rawlins', '6789 Fox Trot Way', 'Atlanta', 'GA', '30301', '404-221-1234', 'Araw@yahoo.com', '6/2/2022', 'Current', 'Administrative Personnel') Insert Into Employee Values (05, 'John', 'Smith', '1010 Sunset Blvd', 'Los Angeles', 'CA', '90201', '310-421-1234', 'JohnSmith23@gmail.com', '3/1/2019', 'Former', 'Administrative Personnel') Insert Into Employee Values (06, 'Jacob', 'Long', '1234 Park View Lane', 'Louisville', 'TN', '37777', '865-765-5668', 'Jacob.Long@utk.edu', '5/4/2020', 'Current', 'Instructor') Insert Into Employee Values (07, 'Sydney', 'Maloney', '453 Rockett Drive', 'Manhattan', 'NY', '10001', '646-664-3734', 'Syd.Mal43@icloud.com', '2/3/2015', 'Former', 'Sales Clerk') Insert Into Employee Values (08, 'William', 'Hound', '1320 Brooks View Avenue', 'Louisville', 'TN', '37701', '865-672-1234', 'WH90@gmail.com', '8/17/2021', 'Current', 'Sales Clerk') Insert Into Employee Values (09, 'Jackson', 'Bolt', '1234 Sweet Pea Road', 'Knoxville', 'TN', '37013', '615-451-1478', 'Jackson.Bolt@yahoo.com', '9/5/2020', 'Current', 'Instructor')

Insert Into Employee Values (10, 'Tiffany', 'Bing', '4221 Volunteer Drive ', 'Knoxville', 'TN', '37919', '865-231-8904', 'Tiif.Bing55@gmail.com', '7/22/2022', 'Current', 'Instructor') Insert Into Employee Values (11, 'Veronica', 'Sale', '5542 Grass Patch Rd', 'Austin', 'TX', '79835', '915-887-6634', 'Veronica.Sale22@ut.edu', '3/4/2018', 'Former', 'Instructor') Insert Into Employee Values (12, 'Reid', 'Haslam', '7754 Green Meadow Lane', 'Alcoa', 'TN', '37701', '865-333-1234', 'Haslam.Reid45@gmail.com', '11/8/2021', 'Current', 'Administrative Personnel') Insert Into Employee Values (13, 'Winston', 'Smart', '6674 Harvest Lane', 'Alcoa', 'TN', '37701', '865-333-3214', 'Winston.Smith53@gmail.com', '10/7/2021', 'Current', 'Sales Clerk') Insert Into Employee Values (14, 'Lily', 'White', '9065 White Valley Rd', 'Knoxville', 'TN', '37916', '865-777-1234', 'LilyWhite90@gmail.com', '1/9/2021', 'Current', 'Sales Clerk') Insert Into Employee Values (15, 'Marley', 'Mauve', '4567 hunter Lakes Way', 'Knoxville', 'TN', '37920', '865-666-5454', 'MarleyJane15@gmail.com', '8/18/2021', 'Former', 'Sales Clerk') Insert Into Employee Values (16, 'Chevy', 'Chase', '6674 Piper Trout Lane', 'Alcoa', 'TN', '37701', '865-303-1489', 'ChevyChase@gmail.com', '9/7/2022', 'Current', 'Administrative Personnel') Insert Into Employee Values (17, 'Blake', 'Lively', '5567 Scooter Rd', 'Knoxville', 'TN', '37916', '865-543-2224', 'LilyWhite90@gmail.com', '4/3/2021', 'Current', 'Administrative Personnel') Insert Into Employee Values (18, 'Taylor', 'Swift', '4567 Scary Hollow Avenue', 'Knoxville', 'TN', '37920', '865-660-5454', 'MarleyJane15@gmail.com', '8/18/2021', 'Former', 'Administrative Personnel') Insert Into Employee Values (19, 'Oprah', 'Winfrey', '1111 Rocket Lane', 'Knoxville', 'TN', '37919', '865-456-7894', 'OprahWinfey@icloud.com', '3/17/2020', 'Former', 'Administrative Personnel') Insert Into Employee Values (20, 'Nick', 'Cannon', '222 Forest Lane', 'Knoxville', 'TN', '37919', '865-456-8794', 'Cannon.Nick@icloud.com', '3/17/2021', 'Former', 'Instructor') Insert Into Employee Values (21, 'Robby', 'Williams', '333 Module Lane', 'Knoxville', 'TN', '37920', '865-654-8794', 'RWilliams@gmail.com', '8/16/2021', 'Current', 'Instructor') Insert Into Employee Values (22, 'Rebecca', 'Snu', '452 Candy Lane', 'Knoxville', 'TN', '37919', '865-546-8904', 'Rebecca.Snu11@icloud.com', '7/9/2021', 'Former', 'Instructor') Insert Into Employee Values (23, 'Carly', 'Corn', '777 Phoneix Ave',	
---	--

	'Knoxville','TN', '37929', '865-506-7864',' CarlyCorn@icloud.com ','3/7/2022','Current','Instructor') Insert Into Employee Values (24, 'Taylor', 'Swift','999 Lover Lane', 'Knoxville','TN', '37920', '865-963-5555','TSwizzle@gmail.com','2/17/2018','Former','Administrative Personnel') Insert Into Employee Values (25, 'Justin', 'Bieber','7234 Ghost Ave', 'Knoxville','TN', '37920', '865-860-5810','JBieb1@icloud.com','5/11/2022','Current','Administrative Personnel') Insert Into Employee Values (26, 'Jennifer', 'Lopez','786 Block Ave', 'Knoxville','TN', '37920', '865-705-8500','Jalo12@gmail.com','8/15/2022','Current','Administrative Personnel') Insert Into Employee Values (27, 'Harry', 'Styles','1234 Cinema Dr', 'Knoxville','TN', '37916', '865-998-1234','Harry.Styles@icloud.com','3/4/2022','Current','Sales Clerk') Insert Into Employee Values (28, 'John', 'Doe','8760 Cades Cove Lane', 'Knoxville','TN', '37916', '865-123-4456','JDoe334@icloud.com','2/8/2020','Former','Sales Clerk') Insert Into Employee Values (29, 'Nick', 'Jonas','1657 Lonas Ave', 'Knoxville','TN', '37919', '865-765-4450','NJonas@gmail.com','3/8/2019','Former','Sales Clerk') Insert Into Employee Values (30, 'Peyton', 'Manning','1234 Manning Ave', 'Knoxville','TN', '37916', '865-506-8900','Manning.Peyton@icloud.com','6/3/2022','Current','Sales Clerk')
Equipment	Insert Into Equipment Values ('ES1', 'E4', '09/12/2006', '07/05/2022'); Insert Into Equipment Values ('ES2', 'E1', '08/06/2004', '06/06/2022'); Insert Into Equipment Values ('ES3', 'E6', '04/14/2012', '02/04/2022'); Insert Into Equipment Values ('ES4', 'E8', '12/16/2007', '04/03/2022'); Insert Into Equipment Values ('ES5', 'E1', '11/18/2013', '07/02/2022'); Insert Into Equipment Values ('ES6', 'E4', '10/19/2020', '09/05/2022'); Insert Into Equipment Values ('ES7', 'E5', '04/15/2015', '08/16/2022'); Insert Into Equipment Values ('ES8', 'E3', '06/14/2006', '05/13/2022'); Insert Into Equipment Values ('ES9', 'E8', '07/23/2009', '07/28/2022'); Insert Into Equipment Values ('ES10', 'E9', '08/25/2010', '03/23/2022');
Equipment Type	Insert Into EquipmentType Values ('E1', 'Treadmill', 'CCC888', '2 months', 2); Insert Into EquipmentType Values ('E2', 'Elliptical', 'DDD888', '3 months', 4); Insert Into EquipmentType Values ('E3', 'Bike', 'CCV999', '4 months', 3); Insert Into EquipmentType Values ('E4', 'Rowing Machine', 'BBB444', '2 months', 7); Insert Into EquipmentType Values ('E5', 'Leg Press', 'LLL333', '1 month', 8); Insert Into EquipmentType Values ('E6', 'Stair Climber', 'UUU777', '2 months', 4);

	Insert Into EquipmentType Values ('E7', 'Balance Trainer', 'OOO000', '4 months', 6); Insert Into EquipmentType Values ('E8', 'Excercise ball', 'YYY888', '6 months', 8); Insert Into EquipmentType Values ('E9', 'Bicep Curler', 'RRR666', '3 months', 9); Insert Into EquipmentType Values ('E10', 'Compact treadmill', 'PPP555', '2 months', 2);
Instructor	Insert Into Instructor Values ('I01', 01, 50000) Insert Into Instructor Values ('I02', 03, 50000) Insert Into Instructor Values ('I03', 06, 60000) Insert Into Instructor Values ('I04', 09, 70000) Insert Into Instructor Values ('I05', 10, 45000) Insert Into Instructor Values ('I06', 11, 50000) Insert Into Instructor Values ('I07', 20,60000) Insert Into Instructor Values ('I08', 21,40000) Insert Into Instructor Values ('I09', 22, 65000) Insert Into Instructor Values ('I10', 23, 45000)
Inventory	Insert Into Inventory Values (1,'Woman Tank', 30.00, 200, 50) Insert Into Inventory Values (2,'Woman Shorts', 35.00, 200, 50) Insert Into Inventory Values (3,'Woman Leggings', 40.00, 250, 60) Insert Into Inventory Values (4,'Athletic Hoodie', 50.00, 150, 40) Insert Into Inventory Values (5,'Water Bottle', 30.00, 100, 20) Insert Into Inventory Values (6,'Yoga Mat', 25.00, 150, 20) Insert Into Inventory Values (7,'T shirt', 25.00, 300, 50) Insert Into Inventory Values (8,'Men Shorts', 40.00, 150, 40) Insert Into Inventory Values (9,'Muscle Tank', 30.00, 200, 30) Insert Into Inventory Values (10,'Dumbbells', 60.00, 100, 20)
Manufacturer	Insert into Manufacturer values ('M1', 2, 'Pilot Company', '5508 Lonas Drive', 'Knoxville', 'Tennessee', 37909, '865-999-9999', 'pilot@pilottravelcenters.com') Insert into Manufacturer values ('M2', 5, 'Regal Cinemas', '7132 Regal Lane', 'Knoxville', 'Tennessee', 37918, '865-999-9998', 'regal@regal cinemas.com') Insert into Manufacturer values ('M3', 9, 'Bush Beans', '1016 E Weisgarber Rd', 'Knoxville', 'Tennessee', 37909, '865-999-9996', 'bushbeans@beansco.com') Insert into Manufacturer values ('M4', 2, 'Tombras', '620 S Gay St', 'Knoxville', 'Tennessee', 37902, '865-999-9997', 'tombras@tomrbrasagency.com') Insert into Manufacturer values ('M5', 4, 'Regal Cinemas', '7132 Regal Lane', 'Knoxville', 'Tennessee', 37918, '865-999-9998', 'regal@regal cinemas.com') Insert into Manufacturer values ('M6', 6, 'Pilot Company', '5508 Lonas Drive', 'Knoxville', 'Tennessee', 37909, '865-999-9999', 'pilot@pilottravelcenters.com') Insert into Manufacturer values ('M7', 7, 'Tombras', '620 S Gay St', 'Knoxville', 'Tennessee', 37902, '865-999-9997', 'tombras@tomrbrasagency.com') Insert into Manufacturer values ('M8', 3, 'Pilot Company', '5508 Lonas Drive',

	'Knoxville', 'Tennessee', 37909, '865-999-9999', 'pilot@pilottravelcenters.com') Insert into Manufacturer values ('M9', 4, 'Regal Cinemas', '7132 Regal Lane', 'Knoxville', 'Tennessee', 37918, '865-999-9998', 'regal@regal cinemas.com') Insert into Manufacturer values ('M10', 7, 'Pilot Company', '5508 Lonas Drive', 'Knoxville', 'Tennessee', 37909, '865-999-9999', 'pilot@pilottravelcenters.com')
Member	INSERT INTO Member VALUES (1, 'Kelly', 'McDonald', '106 Market Square', 'Knoxville', 'TN', '37916', '8883334444', 'Female', '2002-09-04', '2021-02-04', 3); INSERT INTO Member VALUES (2, 'Sam', 'Johnson', '111 Ball Road', 'Farragut', 'TN', '37111', '3339998888', 'Male', '2001-01-02', '2020-01-15', 4); INSERT INTO Member VALUES (3, 'Carson', 'Anderson', '303 Curve Road', 'Loudon', 'MD', '21704', '4446661111', 'Male', '1994-07-14', '2022-06-14', 1); INSERT INTO Member VALUES (4, 'Laurie', 'Fox', '254 Harvest Lane', 'Brentwood', 'TN', '33527', '9996668888', 'Female', '1975-12-13', '2020-02-17', 4); INSERT INTO Member VALUES (5, 'Andy', 'Galloway', '554 Lixon Drive', 'Brunswick', 'MD', '87425', '9993332222', 'Male', '1999-04-12', '2022-04-15', 2); INSERT INTO Member VALUES (6, 'Grant', 'Leaver', '785 Router Way', 'Mascot', 'TN', '76783', '7772220000', 'Male', '1989-06-05', '2022-04-16', 3); INSERT INTO Member VALUES (7, 'Katie', 'Kirkland', '134 Flounder Way', 'Blaine', 'AZ', '77642', '9997774444', 'Female', '2001-03-19', '2021-09-17', 3); INSERT INTO Member VALUES (8, 'Mark', 'Fitz', '871 Victory Lane', 'Knoxville', 'TN', '37916', '7774445555', 'Male', '1997-11-14', '2022-05-25', 4); INSERT INTO Member VALUES (9, 'Riya', 'Anne', '555 Triton Lane', 'Frederick', 'MD', '21704', '3339996666', 'Female', '2002-05-17', '2022-08-29', 2); INSERT INTO Member VALUES (10, 'Lily', 'Henderson', '772 Randol Square', 'Wicker', 'AL', '65271', '6669992222', 'Female', '2002-09-04', '2022-10-22', 1);
Membership Type	INSERT INTO MembershipType VALUES (1, 'Bronze', 50, 0); INSERT INTO MembershipType VALUES (2, 'Silver', 100, 10); INSERT INTO MembershipType VALUES (3, 'Gold', 200, 20); INSERT INTO MembershipType VALUES (4, 'Platinum', 300, 30);
Order Line	INSERT INTO OrderLine VALUES (1, 2, 'Woman Shorts', 20, 20, 30.00, 600.00) INSERT INTO OrderLine VALUES (2, 1, 'Woman Tank', 20, 20, 30.00, 600.00) INSERT INTO OrderLine VALUES (3, 5, 'Water Bottle', 20, 20, 20.00, 400.00) INSERT INTO OrderLine VALUES (4, 10, 'Dumbbells', 10, 10, 40.00, 400.00) INSERT INTO OrderLine VALUES (5, 9, 'Muscle Tank', 20, 20, 25.00, 500.00) INSERT INTO OrderLine VALUES (6, 7, 'T shirt', 20, 20, 20.00, 400.00) INSERT INTO OrderLine VALUES (7, 6, 'Yoga Mat', 10, 10, 20.00, 200.00) INSERT INTO OrderLine VALUES (8, 8, 'Men Shorts', 20, 20, 30.00, 600.00) INSERT INTO OrderLine VALUES (9, 4, 'Athletic Hoodie', 20, 20, 30.00,

	600.00) INSERT INTO OrderLine VALUES (10, 3, 'Woman Leggings', 20, 20, 40.00, 800.00)
Pay Fee	INSERT INTO PayFee VALUES (1, 2, 'Y'); INSERT INTO PayFee VALUES (4, 6, 'Y'); INSERT INTO PayFee VALUES (2, 8, 'Y'); INSERT INTO PayFee VALUES (1, 7, 'Y'); INSERT INTO PayFee VALUES (3, 6, 'Y'); INSERT INTO PayFee VALUES (5, 9, 'Y'); INSERT INTO PayFee VALUES (6, 2, 'Y'); INSERT INTO PayFee VALUES (7, 1, 'Y'); INSERT INTO PayFee VALUES (10, 2, 'Y'); INSERT INTO PayFee VALUES (4, 1, 'Y');
Purchase	INSERT INTO Purchase VALUES (1, 1, 23, '12/30/2021', 30.00) INSERT INTO Purchase VALUES (2, 2, 16, '12/31/2021', 100.00) INSERT INTO Purchase VALUES (3, 2, 14, '01/21/2022', 120.00) INSERT INTO Purchase VALUES (4, 3, 14, '03/04/2022', 60.00) INSERT INTO Purchase VALUES (5, 6, 2, '03/06/2022', 160.00) INSERT INTO Purchase VALUES (6, 7, 5, '03/28/2022', 35.00) INSERT INTO Purchase VALUES (7, 7, 12, '04/01/2022', 250.00) INSERT INTO Purchase VALUES (8, 7, 25, '04/13/2022', 40.00) INSERT INTO Purchase VALUES (9, 8, 28, '05/12/2022', 60.00) INSERT INTO Purchase VALUES (10, 8, 18, '05/25/2022', 250.00)
Purchase Line	INSERT INTO PurchaseLine VALUES (1, 'Water Bottle', 1, 30.00, 23, 1, 5) INSERT INTO PurchaseLine VALUES (2, 'Yoga Mat', 10, 45.00, 21, 7, 6) INSERT INTO PurchaseLine VALUES (3, 'Woman Leggings', 1, 55.00, 16, 8, 3) INSERT INTO PurchaseLine VALUES (4, 'T shirt', 10, 30.00, 25, 10, 7) INSERT INTO PurchaseLine VALUES (5, 'Men Shorts', 4, 50.00, 12, 5, 8) INSERT INTO PurchaseLine VALUES (6, 'Muscle Tank', 2, 35.00, 12, 4, 9) INSERT INTO PurchaseLine VALUES (7, 'Athletic Hoodie', 2, 50.00, 17, 2, 4) INSERT INTO PurchaseLine VALUES (8, 'Dumbbells', 2, 60.00, 24, 3, 10) INSERT INTO PurchaseLine VALUES (9, 'Woman Shorts', 1, 50.00, 28, 6, 2) INSERT INTO PurchaseLine VALUES (10, 'Woman Tank', 2, 50.00, 19, 9, 1)
Qualification	Insert Into Qualification Values ('I01', 1, '12/30/2021') Insert Into Qualification Values ('I01', 2, '12/30/2021') Insert Into Qualification Values ('I01', 10, '11/30/2021') Insert Into Qualification Values ('I03', 3, '6/30/2021') Insert Into Qualification Values ('I03', 4, '5/20/2022') Insert Into Qualification Values ('I03', 5, '10/30/2021') Insert Into Qualification Values ('I04', 6, '9/24/2022') Insert Into Qualification Values ('I05', 7, '3/20/2022')

	Insert Into Qualification Values ('I06', 8, '2/10/2021') Insert Into Qualification Values ('I07', 9, '4/30/2022')
Sales Clerk	Insert Into SalesClerk Values ('SC01', 2, 100000) Insert Into SalesClerk Values ('SC02', 7, 85000) Insert Into SalesClerk Values ('SC03', 27, 60000) Insert Into SalesClerk Values ('SC04', 28, 75000) Insert Into SalesClerk Values ('SC05', 8, 90000) Insert Into SalesClerk Values ('SC06', 29, 60000) Insert Into SalesClerk Values ('SC07', 13, 100000) Insert Into SalesClerk Values ('SC08', 30, 75000) Insert Into SalesClerk Values ('SC09', 14, 80000) Insert Into SalesClerk Values ('SC10', 15, 65000)
Supply Order	INSERT INTO SupplyOrder VALUES(1, '12/21/2021', '12/22/2021', 3, 'SC09') INSERT INTO SupplyOrder VALUES(2, '12/25/2021', '12/27/2021', 1, 'SC05') INSERT INTO SupplyOrder VALUES(3, '01/04/2022', '01/05/2022', 1, 'SC06') INSERT INTO SupplyOrder VALUES(4, '01/13/2022', '01/14/2022', 2, 'SC08') INSERT INTO SupplyOrder VALUES(5, '02/09/2022', '02/09/2022', 4, 'SC07') INSERT INTO SupplyOrder VALUES(6, '02/17/2021', '02/18/2022', 5, 'SC01') INSERT INTO SupplyOrder VALUES(7, '02/22/2022', '02/23/2022', 6, 'SC04') INSERT INTO SupplyOrder VALUES(8, '03/02/2022', '03/03/2022', 7, 'SC02') INSERT INTO SupplyOrder VALUES(9, '03/12/2022', '03/13/2022', 8, 'SC03') INSERT INTO SupplyOrder VALUES(10, '03/25/2022', '03/25/2022', 10, 'SC08')
Vendor	INSERT INTO Vendor VALUES (1, 'Leo's Gymwear', '21 Lee St', 'Knoxville', 'TN', '37916', '8654329851', 'leogymwear@gmail.com', '8654329823') INSERT INTO Vendor VALUES (2, 'Sally's Swag', '345 Deary Lane', 'Birmingham', 'AL', '35005', '2056749851', 'salswag@gmail.com', '2054339432') INSERT INTO Vendor VALUES (3, 'Miguel Merch', '76 Juniper Ave', 'Atlanta', 'GA', '30338', '4043322351', 'miguelmerch@gmail.com', '4044323216') INSERT INTO Vendor VALUES (4, 'Nike', '54 Nike St', 'Beaverton', 'OR', '97005', '259342976', 'nike@gmail.com', '2594213421') INSERT INTO Vendor VALUES (5, 'Zoe Zigs', '221 Lewis St', 'Knoxville', 'TN', '37916', '8654218231', 'zoezigs@gmail.com', '8657865123') INSERT INTO Vendor VALUES (6, 'Marshall's Mad Swag', '5 Magnolia Ave', 'Atlanta', 'GA', '30338', '4043617342', 'marshswag@gmail.com', '4044322123') INSERT INTO Vendor VALUES (7, 'Marathon Mindset', '321 Highland Ave', 'Knoxville', 'TN', '37916', '8652184631', 'marathonmindset@gmail.com', '8651746193') INSERT INTO Vendor VALUES (8, 'Yolanda's Yoga Supplies', '64 Sir Keegan Way', 'Knoxville', 'TN', '37916', '8651984461', 'yolandayogi@gmail.com', '8651836584') INSERT INTO Vendor VALUES (9, 'Nick's Kicks', '786 Brighton Way',

	'Knoxville', 'TN', '37916', '8654329851', 'nick.kicks@gmail.com', '8657952470') INSERT INTO Vendor VALUES (10, 'Macie's Makes', '287 Lake Ave', 'Birmingham', 'AL', '35005', '2053817541', 'maciemakes@gmail.com', '2057323713')
Work Order	Insert Into WorkOrder values ('SN1', 'ES3', 'M1', '04/22/2022', '03/30/2022'); Insert Into WorkOrder values ('SN2', 'ES6', 'M5', '09/22/2022', '11/30/2022'); Insert Into WorkOrder values ('SN3', 'ES8', 'M6', '07/22/2022', '09/30/2022'); Insert Into WorkOrder values ('SN4', 'ES4', 'M7', '09/22/2022', '10/30/2022'); Insert Into WorkOrder values ('SN5', 'ES3', 'M5', '06/22/2022', '09/30/2022'); Insert Into WorkOrder values ('SN6', 'ES9', 'M4', '07/22/2022', '09/30/2022'); Insert Into WorkOrder values ('SN7', 'ES8', 'M7', '09/22/2022', '11/30/2022'); Insert Into WorkOrder values ('SN8', 'ES2', 'M8', '10/22/2022', '11/30/2022'); Insert Into WorkOrder values ('SN9', 'ES4', 'M9', '03/22/2022', '05/30/2022'); Insert Into WorkOrder values ('SN10', 'ES5', 'M3', '10/22/2022', '10/30/2022');

Update Statements:

Class	Update Class Set MembersEnrolled = 15 Where ClassroomNumber = 1 Update Class Set MembersEnrolled = 13 Where ClassroomNumber = 2 Update Class Set MembersEnrolled = 20 Where ClassroomNumber = 3 Update Class Set MembersEnrolled = 25 Where ClassroomNumber = 4 Update Class Set MembersEnrolled = 29 Where ClassroomNumber = 5 Update Class Set MembersEnrolled = 55 Where ClassroomNumber = 6
-------	--

	Update Class Set MembersEnrolled = 8 Where ReferenceNo = 1 Update Class Set MembersEnrolled = 10 Where ReferenceNo = 9 Update Class Set MembersEnrolled = 19 Where ClassroomNumber = 8 Update Class Set MembersEnrolled = 0 Where ClassroomNumber = 9 Update Class Set MembersEnrolled = 30 Where ClassroomNumber = 10
Class	Update Class Set InstructorID = 'I01' Where Course# = 2 Update Class Set InstructorID = 'I01' Where Course# = 1 Update Class Set InstructorID = 'I03' Where Course# = 3 Update Class Set InstructorID = 'I03' Where Course# = 4 Update Class Set InstructorID = 'I04' Where Course# = 6 Update Class Set InstructorID = 'I03' Where Course# = 5 Update Class Set InstructorID = 'I05'

	Where Course# = 7 Update Class Set InstructorID = 'I10' Where Course# = 9
Class	Update Class Set Employee_ID = 6 Where InstructorID = 'I03' Update Class Set Employee_ID = 9 Where InstructorID = 'I04' Update Class Set Employee_ID = 1 Where InstructorID = 'I01' Update Class Set Employee_ID = 10 Where InstructorID = 'I05' Update Class Set Employee_ID = 23 Where InstructorID = 'I10'
Qualification	Update Qualification Set InstructorID = 'I08' Where Course# = 8 Update Qualification Set InstructorID = 'I10' Where Course# = 9 Update Qualification Set Date_Qualified = '04/3/2021' Where Course# = 2 Update Qualification Set Date_Qualified = '2021-09-24' Where Course# = 6
Course	Update Qualification Set InstructorID = 'I08' Where Course# = 8

	Update Qualification Set InstructorID = 'I10' Where Course# = 9
--	---

SQL Commands To Create Views

Query	SQL Commands for Views
1	<pre>Create View Query1 as select Member.Member_ID, member.MemberFirstName, Member.MemberLastName, Member.Phone_Number, MembershipType.MembershipType, MembershipType.Discount, PayFee.Paid, PayFee.ReferenceNO, Course.Course_Fee, Course.Course_Name from Member inner join MembershipType on Member.MembershipTypeID=MembershipType.MembershipTypeID full outer join PayFee on Member.Member_ID=PayFee.Member_ID full outer join Class on Class.ReferenceNO=PayFee.ReferenceNO left join Course on Course.Course#=Class.Course#</pre>
2	<pre>Create view Query2 as Select Class.InstructorID, Employee.FirstName, Employee.LastName,Class.Course#,Class.ClassroomNumber,Class.Class_Held,Class. MembersEnrolled,Classroom.Class_Capacity,StartDate,Class.Time_Period, (Classroom.Class_Capacity - Class.MembersEnrolled) as OpenSpots From Class Inner Join Employee on Class.Employee_ID = Employee.Employee_ID Inner Join Classroom on Class.ClassroomNumber = Classroom.ClassroomNumber</pre>
3	<pre>Create view Query3 as select Employee.FirstName as SalesClerkFirstName, Employee.LastName as SalesClerkLastName, Vendor.CompanyName, Vendor.Email, Vendor.FaxNumber, Vendor.Phone,SupplyOrder.DatePlaced, SupplyOrder.DateReceived, OrderLine.QuantityOrdered, OrderLine.QuantityReceived from SupplyOrder inner join vendor on SupplyOrder.VendorID=vendor.VendorID inner join OrderLine on OrderLine.SupplyOrderID=SupplyOrder.SupplyOrderID inner join SalesClerk on SalesClerk.SalesClerkID=SupplyOrder.SalesClerkID inner join Employee on Employee.Employee_ID=SalesClerk.Employee_ID group by Employee.FirstName, Employee.LastName, Vendor.CompanyName, Vendor.Email, Vendor.FaxNumber, Vendor.Phone, SupplyOrder.DateReceived, SupplyOrder.DatePlaced, OrderLine.QuantityOrdered, OrderLine.QuantityReceived</pre>
5	<pre>Create View Query5 as SELECT Distinct EquipmentType.EquipmentID,EquipmentType.Description, WorkOrder.ManufacturerID, count(EquipmentType.Course#) as [Number of Courses], count(EquipmentType.EquipmentID) as [Pieces of Equipment],</pre>

	count(WorkOrder.WorkOrder#) as [Number of Work Orders] From Equipment inner join Equipmenttype On Equipment.EquipmentID= EquipmentType.EquipmentID Left join WorkOrder On Equipment.EquipmentSerialNumber= WorkOrder.EquipmentSerialNumber inner join Manufacturer on Manufacturer.ManufacturerID=WorkOrder.ManufacturerID Group By EquipmentType.Description,EquipmentType.EquipmentID, WorkOrder.ManufacturerID
7	Create view Query7 as Select Qualification.InstructorID, Employee.FirstName, Employee.LastName, Qualification.Course#, Qualification.Date_Qualified, Class.StartDate, Count(Class.Course#) as NumofClasses From Class Right Join Qualification on class.Course# = Qualification.Course# Left Join Employee on Class.Employee_ID =Employee.Employee_ID Group By Qualification.InstructorID,Qualification.Course#, Qualification.Date_Qualified, Class.StartDate,Employee.FirstName, Employee.LastName
8	Create View Query8 as Select Purchase.TransactionDate, OrderLine.ItemID as 'Item ID', OrderLine.ItemsOrdered as 'Items Ordered', PurchaseLine.Quantity as 'Units Sold', PurchaseLine.PricePerUnit as 'Price Per Unit', (PurchaseLine.PricePerUnit-OrderLine.CostChargedPerUnit) as 'Total Revenue Per Item', OrderLine.QuantityOrdered as 'Units Ordered for Own Inventory', OrderLine.CostChargedPerUnit as 'Total Cost Per Item Ordered' From OrderLine, PurchaseLine, Purchase WHERE OrderLine.ItemID=PurchaseLine.ItemID AND Purchase.PurchaseID=PurchaseLine.PurchaseID;
9	Create view Query9 as Select Member.Member_ID, Member.Phone_number, Member.MembershipTypeID, sum(Course.Course_Fee) as [Standard Total Fee], sum(Course.Course_Fee-(MembershipType.Discount* Course.Course_Fee)) as [Discounted Total Fee], sum(MembershipType.Discount*Course.Course_Fee) as [Total Savings] from Member inner join MembershipType on Member.MembershipTypeID=MembershipType.MembershipTypeID full outer join PayFee on Member.Member_ID=PayFee.Member_ID full outer join Class on Class.ReferenceNO=PayFee.ReferenceNO left join Course on Course.Course#=Class.Course#

	GROUP BY Member.Member_ID, Member.Phone_number, Member.MembershipTypeID
--	--

SQL Commands to Create Queries in Access

Query	SQL Command for Queries in Access
1	SELECT dbo_Query3.SalesClerkFirstName, dbo_Query3.SalesClerkLastName, dbo_Query3.CompanyName, dbo_Query3.VendorID, dbo_Query3.Email, dbo_Query3.FaxNumber, dbo_Query3.Phone, dbo_Query3.DatePlaced, dbo_Query3.DateReceived, dbo_Query3.QuantityOrdered, dbo_Query3.QuantityReceived, Day([DateReceived])-Day([DatePlaced]) AS [Days to Recieve] FROM dbo_Query3 WHERE (((dbo_Query3.CompanyName)=[Forms]![Query3Form]![Combo1]) AND ((dbo_Query3.DatePlaced)=[Forms]![Query3Form]![Combo2]));
2	SELECT dbo_Query2.InstructorID, dbo_Query2.FirstName, dbo_Query2.LastName, dbo_Query2.[Course#], dbo_Query2.ClassroomNumber, dbo_Query2.Class_Held, dbo_Query2.MembersEnrolled, dbo_Query2.Class_Capacity, dbo_Query2.StartDate, dbo_Query2.Time_Period, dbo_Query2.OpenSpots FROM dbo_Query2 WHERE (((dbo_Query2.LastName)=[Forms]![Query2Form]![Combo2]));
3	SELECT dbo_Query1.Member_ID, dbo_Query1.MemberFirstName, dbo_Query1.MemberLastName, dbo_Query1.Phone_Number, dbo_Query1.MembershipType, dbo_Query1.Discount, dbo_Query1.Paid, dbo_Query1.ReferenceNO, dbo_Query1.Course_Fee, dbo_Query1.Course_Name, ((100-[Discount])/100)*[Course_Fee] AS [Amount Paid] FROM dbo_Query1 WHERE (((dbo_Query1.Member_ID)=[Forms]![Query1Form]![cbomemberID]));
5	SELECT dbo_Query5.EquipmentID, dbo_Query5.Description, dbo_Query5.[Number of Courses], dbo_Query5.[Pieces of Equipment], dbo_Query5.[Number of Work Orders] FROM dbo_Query5 WHERE (((dbo_Query5.EquipmentID)=[Forms]![Query5Form]![Combo2]));
7	SELECT dbo_Query7.InstructorID, dbo_Query7.FirstName, dbo_Query7.LastName, dbo_Query7.[Course#], dbo_Query7.Date_Qualified, dbo_Query7.StartDate, dbo_Query7.NumofClasses FROM dbo_Query7 WHERE (((dbo_Query7.InstructorID)=[Forms]![Query7Form]![Combo2]) AND ((dbo_Query7.[Course#])=[Forms]![Query7Form]![Combo21]));
8	SELECT dbo_Query8.TransactionDate, dbo_Query8.[Item ID], dbo_Query8.[Items Ordered], dbo_Query8.[Units Sold], dbo_Query8.[Price Per Unit], dbo_Query8.[Total Revenue Per Item], dbo_Query8.[Units Ordered for Own Inventory],

	dbo_Query8.[Total Cost Per Item Ordered], [dbo_Query8]![Units Sold]*[dbo_Query8]![Total Revenue Per Item] AS [Total Revenue Per Order] FROM dbo_Query8 WHERE (((dbo_Query8.TransactionDate)=[Forms]![Query8Form]![Combo2]) AND ((dbo_Query8.[Item ID])=[Forms]![Query8Form]![Combo11]));
9	SELECT dbo_Query9.Member_ID, dbo_Query9.Phone_number, dbo_Query9.MembershipTypeID, dbo_Query9.[Standard Total Fee], dbo_Query9.[Discounted Total Fee], dbo_Query9.[Total Savings] FROM dbo_Query9;

Descriptions of Prototype Elements

Query	Description of Prototype Element
1	The first query is utilized to pull membership information and Query 1 Report displays this information to the member. The member enrollment table displays member information like identification number, first name, last name, phone number, and membership type. As well as course enrollment information including the course name, course number, the original price, the discount applied from their membership type and the price that they paid. The Query1 Form enables the user to search for their entire enrollment information where they can look at all the courses they have signed up for. The Query 1 report then allows the user to access this information to view.
2	The second query requested information about the instructor's class schedule, and the Query 2 Report is a visual representation of each Instructor's Schedule. The Instructor's Schedule shows the InstructorID, Name, and Course & Class Information. Though the database records all instructors (both former and current) the schedule only includes current Instructors that are currently teaching a course. One can see from the report that some instructors are qualified to teach many courses and currently are teaching a variety of different workout classes. The Query2 Form enables the search parameters, so the user can find information regarding certain instructors during certain time periods. The user can select the Last Name of the instructor and the date of the course for more information about the schedule. This feature will be helpful to sort through data when more instructors get added as Rise-and-Grind's business continues to grow.
3	The third query is utilized to look at Rise and Grinds supplier information and their respective orders. The Query 3 Form is used so that management can input the company's Identification number and the order date of a specific order in order to access the information relative to that order. The Query 3 Report is used for management to be able to actually look at the status of the orders and all of the relative information.
5	The fifth query displays information on each equipment type. The Query5 report shows each equipment type, the number of courses in which they are used, how many pieces of equipment Rise and Grind for each type, and the number of work orders associated with each type. The report also displays equipment types that are not associated with any work orders. The Query5 Form allows the user to select an EquipmentID and ManufacturerID to see more information about a specific equipment type and manufacturer. This feature allows users to narrow their searches more quickly and will be beneficial for Rise-and-Grind employees seeking data on equipment types.
7	The seventh query shows how many courses and classes the instructor is qualified to teach, and the Query 7 Report is a visual representation of Qualified Instructors. Currently, some instructors teach multiple classes of the same course. Though the database records all instructors (both former and current) the report and query only includes current Instructors. InstructorID 'I08' is qualified to teach a course, but they are not current classes of that course. On the Query 7 Form, the user can interact with the combo boxes and buttons to select the instructor and course they want to look up. This feature is helpful in order to narrow down the search and increase efficiency for

	Rise-and-Grind's management.
8	The eighth query shows information about items ordered by our customers. The Query8 report shows the transaction date, item ID, items ordered, units sold, price charged per unit, total revenue per item, units ordered for Rise&Grind's own inventory, total cost per item ordered, and total revenue per order. Using the Query8 Form, the user can select a transaction date and item ID to see specific customer order information. This feature will allow Rise-and-Grind employees to find item data more efficiently and allow management to see how much revenue is acquired through each customer order. For example, setting the parameters in the Query8 Form to TransactionDate= 2021-12-30 and Item ID= 5 would provide information about 1 water bottle being purchased for \$30.00, which produced \$10.00 in revenue.
9	The ninth query shows members and their total savings on course enrollment fees based on their membership type discount. The Query9 Report includes MembershipID, PhoneNumber, MembershipTypeID, Standard Total Fee, Discounted Total Fee, and Total Savings. Using the Query9 Form, the user can select MembershipID to see a specific member's contact information and total savings. This feature will allow Rise-and-Grind employees to find membership data more efficiently.

Instructions On How To Use Front-End Software

Query	How To Use:																														
1	 Select a Box to Below to look at that particular information Query 1 Query 7 Query 2 Query 8 Query 3 Query 9 Query 5 Select button Query 1 Select Your Member ID 1 Display Details Enter your member ID  Course Enrollment Details <table><thead><tr><th>ID Number</th><th>First Name</th><th>Last Name</th><th>Phone</th><th>Membership Type</th><th>Course</th><th>Course Number</th><th>Discount</th><th>Original Fee</th><th>Amount Paid</th></tr></thead><tbody><tr><td>1</td><td>Kelly</td><td>McDonald</td><td>8883334444</td><td>Gold</td><td>ShadowBoxing</td><td>4</td><td>20</td><td>\$90.00</td><td>72</td></tr><tr><td>1</td><td>Kelly</td><td>McDonald</td><td>8883334444</td><td>Gold</td><td>Flow Yoga</td><td>7</td><td>20</td><td>\$70.00</td><td>56</td></tr></tbody></table> Monday, December 5, 2022 Page 1 of 1 Course enrollment information about that member ID will be displayed.	ID Number	First Name	Last Name	Phone	Membership Type	Course	Course Number	Discount	Original Fee	Amount Paid	1	Kelly	McDonald	8883334444	Gold	ShadowBoxing	4	20	\$90.00	72	1	Kelly	McDonald	8883334444	Gold	Flow Yoga	7	20	\$70.00	56
ID Number	First Name	Last Name	Phone	Membership Type	Course	Course Number	Discount	Original Fee	Amount Paid																						
1	Kelly	McDonald	8883334444	Gold	ShadowBoxing	4	20	\$90.00	72																						
1	Kelly	McDonald	8883334444	Gold	Flow Yoga	7	20	\$70.00	56																						



Select a Box to Below to look at that particular information

Query 1	Query 7
Query 2	Query 8
Query 3	Query 9
Query 5	

Select the Button Query 2.

	Open Report
LastName	Hudson
StartDate	2022-09-15

Select the Last Name and Start Date of the Instructor's Schedule from the drop down box. Then, select the button Open Report.

Instructor's Schedule

InstructorID	First Name	Last Name	Course#	Classroom Number	Day	Members Enrolled	Class Capacity	Start Date	Time Period	Open Spots
101	Kate	Hudson	2	7	Friday	8	10	2022-09-15	08:30	2
101	Kate	Hudson	2	9	Saturday	0	30	2022-09-15	11:00	30

Sunday, December 4, 2022

Page 1 of 1

A report of the information will be given.



Select a Box to Below to look at that particular information

Query 1	Query 7
Query 2	Query 8
Query 3	Query 9
Query 5	

Select the button Query 3 to take you to the Query3 form.

Open Report	
CompanyName	Zoe Zigs
DatePlaced	2021-02-17

Select the company name and date placed and hit the button ‘Open Report’

Supply Orders Information									
First	Last	CompanyName	VendorID	Email	FaxNumber	Phone	DatePlaced	DateReceived	QuantityOrdered
Ella	Scholz	Zoe Zigs	5	zoezigs@gmail.com	8657865123	865421823	2021-02-17	2022-02-18	20

Then, the Supply Orders Report will appear.



Select a Box to Below to look at that particular informatic

Query 1	Query 7
Query 2	Query 8
Query 3	Query 9
Query 5	

Select the button Query 5 and it will take you to Query5 Form.

Open Report	EquipmentID		▼
	ManufacturerID		▼

Select EquipmentID and ManufacturerID in the dropdown box and select Open Report. This will take you to a report on the information.

Equipment Types

EquipmentID	Description	Number of Courses	Pieces of Equipment	Number of Work Orders	Manufacturer
E4	Rowing Machine	1	1	1	M5

Monday, December 5, 2022

Page 1 of 1

7



Select a Box to Below to look at that particular information

Query 1	Query 7
Query 2	Query 8
Query 3	Query 9
Query 5	

Select the button Query 7 and it will take you to Query7Form.

	Open Report
InstructorID	I01
Course#	2

Input the InstructorID and Course# and select 'Open Report' button

Qualified Instructors						
InstructorID	FirstName	LastName	Course#	Date_Qualified	StartDate	NumofClasses
I01	Kate	Huds	2	2021-04-03	2022-09-15	2

Sunday, December 4, 2022 Page 1 of 1

Then, the report will pop up with the Qualified Instructor.

8

Select the button Query 8 and it will take you to Query8Form.



Select a Box to Below to look at that particular information

Query 1	Query 7
Query 2	Query 8
Query 3	Query 9
Query 5	

Input the TransactionDate and Item ID and select 'Open Report' button.

	Open Report
TransactionDate	2021-12-30 ▾
Item ID	5 ▾

Then, an item order report will pop up with the parameters specified.

Rise-and-Grind Item Report						
TransactionDate	Item ID	Items Ordered	Units Sold	Price Per Unit	Total Revenue Per Item	Units Ordered for Own Inventory
2021-12-30	5	Water Bottle	1	\$30.00	\$10.00	20
						Total Cost Per Item Ordered
						\$20.00
						Total Revenue Per Order
						\$10.00
Monday, December 5, 2022						
Page 1 of 1						

9

Select the button Query 9 and it will take you to Query9 Form.



Select a Box to Below to look at that particular informatic

Query 1	Query 7
Query 2	Query 8
Query 3	Query 9
Query 5	

Select the button Query9 to open the Query 9 Form.

Open

There are no parameters to be selected. Select the Open button to see information.

Total Savings					
Wember_ID	Phone_number	MembershipTypeID	Standard Total Fee	Discounted Total Fee	Total Savings
1	8883334444	3	\$170.00		
2	3339998888	4	\$160.00	(\$3,040.00)	\$3,200.00
3	4446661111	1	\$225.00	(\$6,525.00)	\$6,750.00
4	9996668888	4			
5	9993332222	2			
6	7772220000	3	\$180.00	(\$3,420.00)	\$3,600.00
7	9997774444	3	\$80.00	(\$1,520.00)	\$1,600.00
8	7774445555	4	\$90.00	(\$2,610.00)	\$2,700.00
9	3339996666	2	\$100.00	(\$900.00)	\$1,000.00
10	6669992222	1			

Sample Outputs

Query	Sample Outputs:																																																		
1	An example of a report where a member is enrolled:  Course Enrollment Details <table><tr><th>ID Number</th><th>First Name</th><th>Last Name</th><th>Phone</th><th>Membership Type</th><th>Course</th><th>Course Number</th><th>Discount</th><th>Original Fee</th><th>Amount Paid</th></tr><tr><td>1</td><td>Kelly</td><td>McDonald</td><td>8883334444</td><td>Gold</td><td>ShadowBoxing</td><td>4</td><td>20</td><td>\$90.00</td><td> 72 </td></tr></table> <table><tr><td>1</td><td>Kelly</td><td>McDonald</td><td>8883334444</td><td>Gold</td><td>Flow Yoga</td><td>7</td><td>20</td><td>\$70.00</td><td> 56 </td></tr></table> Sunday, December 4, 2022 Page 1 of 1 An example of a report where a member is not enrolled:  Course Enrollment Details <table><tr><th>ID Number</th><th>First Name</th><th>Last Name</th><th>Phone</th><th>Membership Type</th><th>Course</th><th>Course Number</th><th>Discount</th><th>Original Fee</th><th>Amount Paid</th></tr><tr><td>5</td><td>Andy</td><td>Galloway</td><td>9993332222</td><td>Silver</td><td></td><td></td><td>10</td><td></td><td> </td></tr></table> Sunday, December 4, 2022 Page 1 of 1	ID Number	First Name	Last Name	Phone	Membership Type	Course	Course Number	Discount	Original Fee	Amount Paid	1	Kelly	McDonald	8883334444	Gold	ShadowBoxing	4	20	\$90.00	72	1	Kelly	McDonald	8883334444	Gold	Flow Yoga	7	20	\$70.00	56	ID Number	First Name	Last Name	Phone	Membership Type	Course	Course Number	Discount	Original Fee	Amount Paid	5	Andy	Galloway	9993332222	Silver			10		
ID Number	First Name	Last Name	Phone	Membership Type	Course	Course Number	Discount	Original Fee	Amount Paid																																										
1	Kelly	McDonald	8883334444	Gold	ShadowBoxing	4	20	\$90.00	72																																										
1	Kelly	McDonald	8883334444	Gold	Flow Yoga	7	20	\$70.00	56																																										
ID Number	First Name	Last Name	Phone	Membership Type	Course	Course Number	Discount	Original Fee	Amount Paid																																										
5	Andy	Galloway	9993332222	Silver			10																																												
2	An example of an instructor teaching multiple classes of the same course: Instructor's Schedule <table><tr><th>InstructorID</th><th>First Name</th><th>Last Name</th><th>Course#</th><th>Classroom Number</th><th>Day</th><th>Members Enrolled</th><th>Class Capacity</th><th>Start Date</th><th>Time Period</th><th>Open Spot</th></tr><tr><td>I03</td><td>Jacob</td><td>Long</td><td>4</td><td>5</td><td>Tuesday</td><td>29</td><td>30</td><td>2022-07-14</td><td>09:00</td><td>1</td></tr><tr><td>I03</td><td>Jacob</td><td>Long</td><td>4</td><td>4</td><td>Monday</td><td>25</td><td>30</td><td>2022-07-14</td><td>10:30</td><td>5</td></tr></table> Sunday, December 4, 2022 Page 1 of 1	InstructorID	First Name	Last Name	Course#	Classroom Number	Day	Members Enrolled	Class Capacity	Start Date	Time Period	Open Spot	I03	Jacob	Long	4	5	Tuesday	29	30	2022-07-14	09:00	1	I03	Jacob	Long	4	4	Monday	25	30	2022-07-14	10:30	5																	
InstructorID	First Name	Last Name	Course#	Classroom Number	Day	Members Enrolled	Class Capacity	Start Date	Time Period	Open Spot																																									
I03	Jacob	Long	4	5	Tuesday	29	30	2022-07-14	09:00	1																																									
I03	Jacob	Long	4	4	Monday	25	30	2022-07-14	10:30	5																																									
3	An example of a supply order: Supply Orders Information <table><tr><th>First</th><th>Last</th><th>CompanyName</th><th>VendorID</th><th>Email</th><th>FaxNumber</th><th>Phone</th><th>DatePlaced</th><th>DateReceived</th><th>QuantityOrdered</th><th>QuantityRe</th></tr><tr><td>Ella</td><td>Scholz</td><td>Zoe Zigs</td><td>5</td><td>zoezigs@gmail.com</td><td>8657865123</td><td>865421823</td><td>2021-02-17</td><td>2022-02-18</td><td>20</td><td>20</td></tr></table>	First	Last	CompanyName	VendorID	Email	FaxNumber	Phone	DatePlaced	DateReceived	QuantityOrdered	QuantityRe	Ella	Scholz	Zoe Zigs	5	zoezigs@gmail.com	8657865123	865421823	2021-02-17	2022-02-18	20	20																												
First	Last	CompanyName	VendorID	Email	FaxNumber	Phone	DatePlaced	DateReceived	QuantityOrdered	QuantityRe																																									
Ella	Scholz	Zoe Zigs	5	zoezigs@gmail.com	8657865123	865421823	2021-02-17	2022-02-18	20	20																																									

5

An example of Equipment Type 1 work order information:

Equipment Types

EquipmentID	Description	Number of Courses	Pieces of Equipment	Number of Work Orders	Manufacturer
E4	Rowing Machine	1	1	1	M5

Monday, December 5, 2022

Page 1 of 1

7

An example of an instructor teaching multiple classes:

Qualified Instructors

InstructorID	FirstName	LastName	Course#	Date_Qualified	StartDate	NumofClasses
I03	Jacob	Long	4	2022-05-20	2022-07-14	2

Sunday, December 4, 2022

Page 1 of 1

An example of an instructor qualified to teach a course that currently doesn't have a class:

Qualified Instructors

InstructorID	FirstName	LastName	Course#	Date_Qualified	StartDate	NumofClasses
I08			8	2021-02-10		0

Sunday, December 4, 2022

Page 1

8

An example of an item report:

 Rise-and-Grind Item Report

TransactionDate	Item ID	Items Ordered	Units Sold	Price Per Unit	Total Revenue Per Item	Units Ordered for Own Inventory	Total Cost Per Item Ordered
2022-01-21	10 Dumbbells		2	\$60.00	\$20.00	10	\$40.00

Total Revenue Per Order

\$40.00

Monday, December 5, 2022

Page 1 of 1

9

Total Savings

Member_ID	Phone_number	MembershipTypeID	Standard Total Fee	Discounted Total Fee	Total Savings
			\$170.00		
1	8883334444	3	\$160.00	(\$3,040.00)	\$3,200.00
2	3339998888	4	\$225.00	(\$6,525.00)	\$6,750.00
3	4446661111	1			
4	9996668888	4			
5	9993332222	2			
6	7772220000	3	\$180.00	(\$3,420.00)	\$3,600.00
7	9997774444	3	\$80.00	(\$1,520.00)	\$1,600.00
8	7774445555	4	\$90.00	(\$2,610.00)	\$2,700.00
9	3339996666	2	\$100.00	(\$900.00)	\$1,000.00
10	6669992222	1			