DBMS Fitness Club

CIS 9340 Section UTA [26769]

Group 15

Baruch Spring 2023

Professor Gao Qiang

Farjana Miah, farjana.miah@baruchmail.cuny.edu

Jieshan Liang, jieshan.liang@baruchmail.cuny.edu

Jefferson Veliz, Jefferson.veliz@baruchmail.cuny.edu

Devona Samwaru, devona.samwaru@baruchmail.cuny.edu

Cassie Wu, cassie.wu@baruchmail.cuny.edu

Palak Wadel, palak.wadel@baruchmail.cuny.edu

EXECUTIVE SUMMARY

Business Scenario

A newly independently-owned fitness club called DBMS has just opened in Manhattan. Since they have just opened up, they are currently using a manual system to keep track of all the services and activities offered at their club. The use of a manual system did not work efficiently for DBMS as its client base suddenly began to grow tremendously. There were instances where customers weren't able to participate in class sessions that they had originally signed up for because a staff member had accidentally overbooked a class session. In other instances, the facilities couldn't provide customers with the necessary gym equipment because the inventory list wasn't up to date.

The owners have recently reached out to our team because they want to utilize a database system that has the capability to store and manage all the data necessary to run their operations. The owners want to be able to manage their resources and staff all through one uniform system. Specifically, they are seeking our professional skills to create a system that will allow them to keep track of the members of the club, implement a reservation system for private rooms, allow members to schedule one-on-one sessions with personal trainers, and keep track of inventory so that they know when they need to replenish supplies. DBMS wants to encourage its customers to take part in the different types of services offered by the club. We plan to create a database system that makes scheduling activities easier and avoid problems such as overcrowding rooms, overlapping class sessions, and minimizing downtime. DBMS owners hope to efficiently manage fitness club operations through one central platform. The owners of the fitness club hope that by utilizing a single database system for all the facilities and operations, the staff can avoid mistakes such as overbooking, and the members will be more encouraged to take part in all the services and activities that the club has to offer.

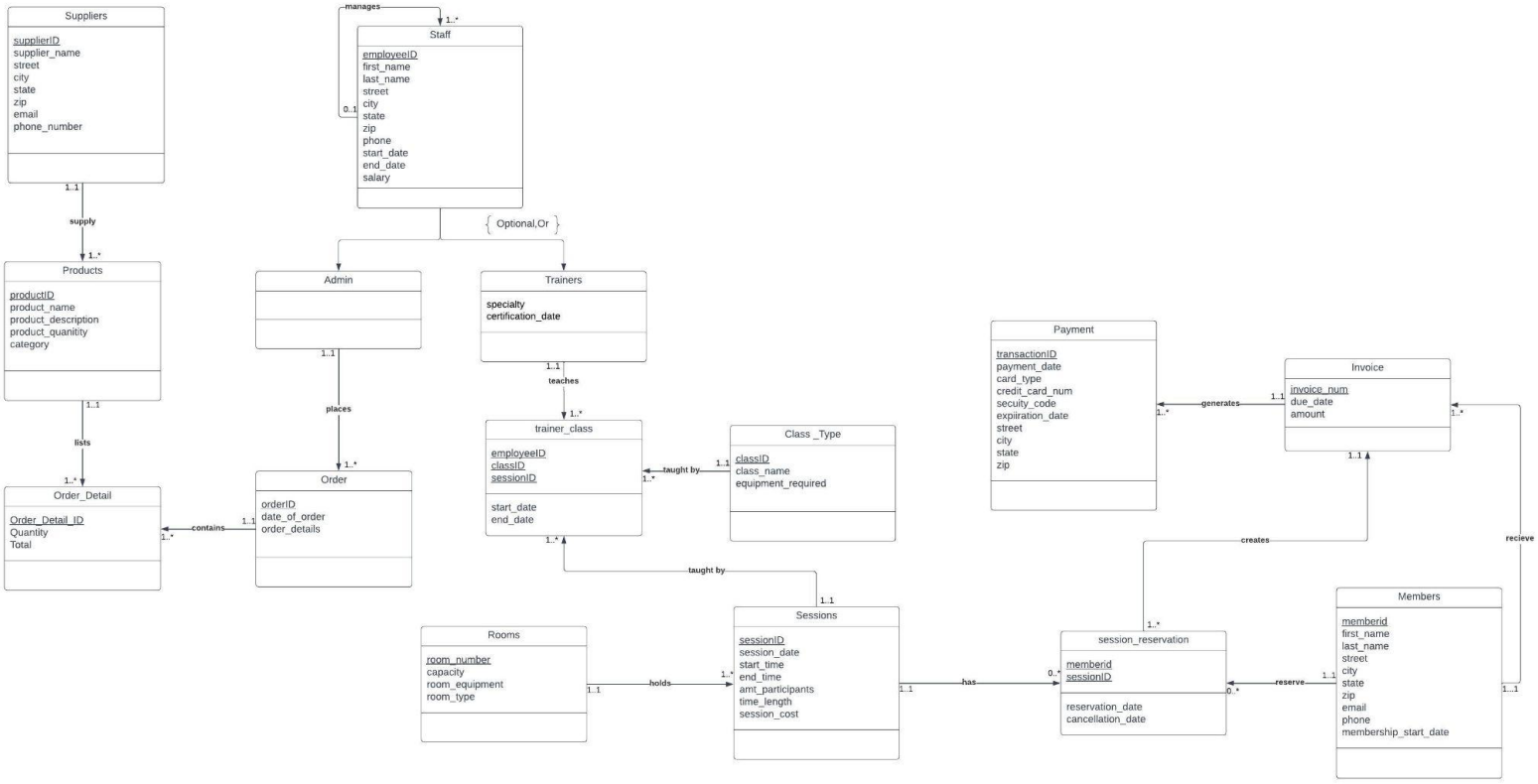
Our team has distributed roles to each member so that we can effectively work together to help DBMS create a successful database system. Farjana and Jieshan will be the system analysts. Jefferson and Palak will be the application developers. Devona and Cassie will be the documentation writers.

Roles	Services
Members	● Pay for monthly memberships and class sessions ● Reserves class sessions
Staff	● Teaches class sessions ● Places inventory orders
Suppliers	● Provides inventory
Sessions	● Provides classes for members

Information Needs

- Keep track of members' payment information to process charges for monthly membership and class sessions attended
- Keep track of members' session reservations to know which class each member signs up for
- Keep track of which staff member will teach each class offered
- Keep track of when each class session takes place, in which training room, and the duration of the class
- Keep track of inventory items to know when certain items need to be replenished

ENTITY RELATIONSHIP MODEL DIAGRAM



Relationship Sentences

- One **Staff** can manage many **Staff**
- One **Admin** can place many **Orders**
- One **Order** contains at least one **Order Details**
- One **Supplier** can supply many **Products**
- One **Product** can be listed in many **Order Details**
- One **Trainer** can teach many **Sessions**
- One **Room** can hold many **Sessions**
- One **Session** can have many **Session Reservations**
- One **Member** can reserve many **Session Reservations**
- One **Member** can receive many **Invoices**
- One **Invoice** is created by at least one **Session Reservation**
- One **Invoice** generates at least one **Payment**

RELATIONAL MODEL

- **Supplier** (supplierID,supplier_name,street,city,state,zip,email,phone_number)
- **Products** (productID, product_name, product_description, product_quanitity category, supplierID(fk))
- **Order_Detail** (Order_Detail_ID,Quantity,Total, productID(fk),orderID(fk))
- **Staff**(employeeID,first_name,last_name,street,city,state,zip,phone,start_date,end_date,salary)
- **Parent_Staff** (employeeID, parent_employeeID(fk))
- **Admin**(employeeID,first_name,last_name,street,city,state,zip,phone,start_date,end_date, salary)
- **Trainers**(employeeID,first_name,last_name,street,city,state,zip,phone,start_date,end_date,salary, speciality, ceritification_date)
- **Order** (orderID,date_of_order,order_details, employeeID(fk))
- **Class_Type** (classID, class_name, equipment_required)
- **Trainer_class** (employeeID(fk),classID(fk),sessionID(fk), start_date, end_date)
- **Sessions**(sessionID,session_date,start_time,end_time,amt_participants,time_length,session_cost,room_number(fk))
- **Room** (room_number, capacity , room_equipment , room_type)
- **Session_reservation**(memberid(fk),sessionID(fk),reservation_date,cancellation_date,invoice_num(fk))
- **Members**(memberid,first_name,last_name,street,city,state,zip,email,phone,membership_start_date)
- **Invoice** (invoice_num, due_date, amount,memberID(fk))
- **Payment**(transactionID,payment_date,card_type,credit_card_num,secuity_code,expiiration_date,street,city,state,zip, invoice_num(fk))

NORMALIZATION

Supplier (supplierID, supplier_name, street, city, state, zip, email, phone_number)
Step 1: Primary Key: supplierID Yes, 1NF
FD1: supplierID → supplier_name, street, city, state, zip, email, phone_number
FD2: zip → city, state
Step 2: Partial dependency? No, 2NF
Step 3: Transitive dependency? Yes
~~R1 (supplierID, supplier_name, street, city, state, zip, email, phone_number)~~
R2 (zip, city, state)
R3 (supplierID, supplier_name,street,zip,email,phone_number)
No more, 3NF
Step 4: Any determinants are not candidate keys? No, BCNF
Total 2 Relations: R2, R3

Products (productID, product_name, product_description, product_quanitity, category, supplierID(fk))
Step 1: Primary Key: productID Yes, 1NF
FD1: productID → product_name, product_description, product_quantity, category, supplierID
Step 2: Partial dependency? No, 2NF
Step 3: Transitive dependency? No, 3NF
Step 4: Any determinants are not candidate key? No, BCNF
R1 (productID, product_name, product_description, product_quanitity, category, supplierID)
Total 1 Relations: R1

Order_Detail (Order_Detail_ID, Quantity, Total, productID(fk), orderID(fk))
Step 1: Primary key: Order_Detail_ID. Yes, 1NF
FD1: Order_Detail_ID → Quantity, Total, productID, orderID
Step 2: Partial dependency? No, 2NF
Step 3: Transitive dependency? No, 3NF
Step 4: Any determinants are not candidate key? No, BCNF
R1 (Order_Detail_ID, Quantity, Total, productID, orderID)
Total 1 Relation: R1

Staff (employeeID, first_name, last_name, street, city, state, zip, phone, start_date, end_date, salary)
Step 1: Primary key: employeeID Yes, 1NF
FD1: employeeID →first_name, last_name, street, city, state, zip, phone, start_date, end_date, salary
FD2: zip →city,state

Step 2: Partial dependency? No, 2NF
Step 3: Transitive dependency? Yes
R1 (zip, city, state)
R2 (employeeID, first_name, last_name,street, zip, phone, start_date, end_date, salary)
Step 4: Any determinants are not candidate key? No, BCNF
Total 2 Relations: R1, R2

Parent_staff (employeeID, parent_employeeID(fk))
Step 1: Primary key: employeeID Yes, 1NF
FD1: employeeID → parent_employeeID
Step 2: Partial dependency? No, 2NF
Step 3: Transitive dependency? No, 3NF
Step 4: Any determinants are not candidate key? No, BCNF
R1 (employeeID, parent_employeeID)
Total 1 Relation: R1

Admin(employeeID,first_name,last_name,street,city,state,zip,phone,start_date,end_date,salary)
Step 1: Primary Key: employeeID Yes, 1NF
FD1: employeeID→first_name,last_name,street,city,state,zip,phone,start_date,
end_date, salary
FD2: zip → city, state
Step 2: Partial dependency? No, 2NF
Step 3: Transitive dependency? Yes
R1(zip,city,state)
R2(employeeID→ first_name, last_name, street,zip, email, phone, start_date, end_date, salary)
Step 4: Any determinants are not candidate keys? No, BCNF
Total 2 Relations: R1, R2

Trainers(employeeID,first_name,last_name,street,city,state,zip,phone,start_date,end_date,salary, speciality, ceritification_date)
Step 1: Primary Key: employeeID Yes, 1NF
FD1:employeeID→first_name,last_name,street,city,state,zip,phone,start_date,end_date,s
alary, speciality, ceritification_date
FD2: zip → city, state
Step 2: Partial dependency No, 2NF
Step 3: Transitive dependency? Yes
R1: (zip,city,state,street)
R2: (employeeID, first_name,last_name,zip, phone,start_date,end_date,salary, speciality, ceritification_date)
Step 4: Any determinants are not candidate keys? No, BCNF

Total 2 Relations: R1, R2

Order(orderID,date_of_order,order_details, employeeID(fk))
Step 1: Primary key: orderID Yes, 1NF
FD1: orderID →date_of_order,order_details, employeeID
Step 2: Partial dependency? No, 2NF
Step 3: Transitive dependency? No, 3NF
Step 4: Any determinants are not candidate keys? No, BCNF
R1(orderID,date_of_order,order_details, employeeID)
Total 1 Relation: R1

Class_Type (classID, class_name, equipment_required)
Step 1: Primary key: classID Yes, 1NF
FD1: classID → class_name, equipment_required
Step 2: Partial dependency? No, 2NF
Step 3: Transitive dependency? No, 3NF
Step 4: Any determinants are not candidate keys? No, BCNF
R1(classID,class_name, equipment_required)
Total 1 Relation: R1

Trainer_class (employeeID(fk), classID(fk), sessionID(fk), start_date, end_date)
Step 1: Primary Key: employeeID, classID, sessionID Yes, 1NF
FD1:employeeID, classID, sessionID → start_date, end_date
Step 2: Partial dependency? No 2NF
Step 3: Transitive dependency? No 3NF
Step 4: Any determinants are not candidate keys? No, BCNF
R1:(employeeID, classID, sessionID, start_date, end_date)
Total 1 Relation: R1

Sessions (sessionID, session_date, start_time, end_time, amt_participants, time_length, session_cost, room_number(fk))
Step 1: Primary Key: sessionID Yes, 1NF
FD1:sessionID →session_date, start_time, end_time, amt_participants, time_length, session_cost, room_number
Step 2: Partial dependency? No 2NF
Step 3: Transitive Dependency? No, 3NF
Step 4: Any determinants are not candidate keys? No, BCNF
R1: (sessionID, session_date, start_time, end_time, time_length,amt_participants, session_cost, room_number)
Total 1 Relation: R1

Room (room_number, capacity, room_equipment , room_type)
Step 1: Primary Key: room_number Yes, 1NF
FD1:room_number → capacity , room_equipment , room_type
Step 2: Partial dependency? No 2NF
Step 3: Transitive Dependency? No, 3NF
Step 4: Any determinants are not candidate keys? No, BCNF
R1: (room_number, capacity , room_equipment , room_type)
Total 1 Relation: R1

Session_reservation(memberid,sessionID,reservation_date,cancellation_date,invoice_num(fk))
Step 1: Primary Key: memberID,SessionID Yes, 1NF
FD1: memberID,sessionID→ reservation_date, cancellation_date, invoice_num
Step 2: Partial Key Dependency? No, 2NF
Step 3: Transitive Dependency? No, 3NF
Step 4: Any determinants that are not a candidate key? No, BCNF
R1(memberID,sessionID,reservation_date,cancellation_date,invoice_num)
Total 1 Relation: R1

Members(memberid,first_name,last_name,street,city,state,zip,email,phone,membership_start_date)
Step 1: Primary Key? memberID Yes,1NF
FD1:memberID→first_name,last_name,street,city,state,zip,email,phone,membership_start_date
FD2: Zip→ city,state
Step 2: Partial Dependency? No, 2NF
Step 3: Transitive Dependency? Yes
R1(Zip,City,State)
R2(memberID,first_name,last_name,street,zip,email,phone,membership_start_date)
Step 4: Any determinants that are not a candidate key? No, BCNF
Total 2 Relations: R1, R2

Invoice(invoice_num, due_date, amount, memberid(fk))
Step 1: Primary Key? invoice_num Yes,1NF
FD1: invoice_num→ due_date, amount, memberID
Step 2: Partial Key Dependency? No, 2NF
Step 3: Transitive Dependency? No, 3NF
Step 4: Any determinants that are not a candidate key? No, BCNF
R1(invoice_num, due_date, amount, memberid(fk))
Total 1 Relation: R1

Payment(transactionID,payment_date,card_type,credit_card_num,security_code,expiration_date,street,city,state,zip, invoice_num(fk))

Step 1: Primary Key? transactionID Yes,1NF

FD1: transactionID→ payment_date, card_type, credit_card_num, security_code, expiration_Date, street, city, state, zip, invoice_num

FD2: Zip→ city, state

FD3: Credit_card_num→card_type, security_code, expiration_date

Step 2: Partial Key Dependency? No,2NF

Step 3: Transitive Dependency? Yes

R1(Credit_card_num, card_type, security_code, expiration_date)

R2(Zip, City, State)

R3(transactionID, payment_date, credit_card_num, street, zip, invoice_num)

Step 4: Any determinants that are not a candidate key? No, BCNF

Total 3 Relations: R1,R2, R3

TABLE CREATION IN SQL

CREATE TABLE Zipcodes (
Zip INTEGER NOT NULL PRIMARY KEY,
City TEXT,
State_usa TEXT
);

INSERT INTO Zipcodes (Zip, City, state_usa)
VALUES (10001, ‘Central City’, 'IL');
INSERT INTO Zipcodes (Zip, City, state_usa)
VALUES (10002, 'South Park', 'CO');
INSERT INTO Zipcodes (Zip, City, state_usa)
VALUES (10003, 'North Beach', 'CA');

Zip	City	state_usa
10001	Central City	IL
10002	South Park	CO
10003	North Beach	CA
10004	New York City	NY
10005	Buffalo	NJ
10006	Syracuse	NY
10007	Albany	MA
10008	New Rochelle	CO
10009	yonkers	NY
10010	Rochester	NY
12345	Smallville	NY
12346	Springfield	IL
23456	Shelbyville	IL
23457	Capital City	IL
45678	Fresno	CA
54321	Metropolis	NY
56789	Queens	NY
67890	Gotham City	NY
78901	San Diego	CA
98376	Bronx	NY

CREATE TABLE Suppliers (
SupplierID TEXT NOT NULL PRIMARY KEY,
Supplier_name TEXT,
Street TEXT,
Zip INTEGER,

Email TEXT,
Phone_INTEGER INTEGER,
FOREIGN KEY (Zip) REFERENCES Zipcodes(Zip)
);

INSERT INTO Suppliers (SupplierID, Supplier_name, Street, Zip, Email, Phone_INTEGER)
VALUES (‘S758’, ‘Fitness Depot’, ‘123 Main St’, 12345, ‘fitnessdepot@example.com’, ‘'(800) 555-5656’);
INSERT INTO Suppliers (SupplierID, Supplier_name, Street, Zip, Email, Phone_INTEGER)
VALUES (‘S923’, ‘Gym Supply’, ‘456 Elm St’, 54321, ‘gymsupply@example.com’, ‘'(555) 132-1234’);
INSERT INTO Suppliers (SupplierID, Supplier_name, Street, Zip, Email, Phone_INTEGER)
VALUES (‘S129’, ‘Workout World’, ‘789 Oak Ave’, 67890, ‘workoutworld@example.com’, ‘'(800) 724-9876’);

SupplierID	Supplier_name	Street	Zip	Email	Phone_INTEGER
S758	Fitness Depot	123 Main St	12345	fitnessdepot@example.com	(800) 555-5656
S923	Gym Supply	456 Elm St	54321	gymsupply@example.com	(555) 132-1234
S129	Workout World	789 Oak Ave	67890	workoutworld@example.com	(800) 724-9876
S768	Fitness Equipment	321 Oak Lane	12346	fitnessequipment@example.com	(718) 611-9845
S784	Sports Gear	654 Pine St	23456	sportsgear@example.com	(929) 471-8204
S348	Fitness Plus	987 Maple Rd	23457	fitnessplus@example.com	(555) 897-0063

CREATE TABLE Products (
ProductID TEXT NOT NULL PRIMARY KEY,
Product_name TEXT,
Product_description TEXT,
Product_quantity INTEGER,
Category TEXT,
SupplierID TEXT,
FOREIGN KEY (SupplierID) REFERENCES Suppliers (SupplierID)
);

INSERT INTO Products (ProductID, Product_name, Product_description, Product_quantity, Category, SupplierID)
VALUES (‘P101’, ‘Dumbbells’, ‘Set of 3 dumbbells’, 20, ‘Weights’, ‘S758’);
INSERT INTO Products (ProductID, Product_name, Product_description, Product_quantity, Category, SupplierID)
VALUES (‘P654’, ‘Yoga Mat’, ‘Non-slip yoga mat’, 30, ‘Yoga’, ‘S923’);

INSERT INTO Products (ProductID, Product_name, Product_description, Product_quantity, Category, SupplierID)
VALUES (‘P902’, ‘Treadmill’, ‘High-performance treadmill’, 5, ‘Cardio’, ‘S129’);

ProductID	Product_name	Product_description	Product_quantity	Category	SupplierID
P101	Dumbbells	Set of 3 dumbbells	20	Weights	S758
P654	Yoga Mat	Non-slip yoga mat	30	Yoga	S923
P902	Treadmill	High-performance treadmill	5	Cardio	S129
P765	Kettlebell set	Set of 3 kettlebells with stand	10	Weights	S768
P753	Resistance band	Pack of 5 different resistance bands	25	Weights	S784
P954	Exercise bike	Recumbent exercise bike with LCD screen	7	Cardio	S348

CREATE TABLE Staff (
EmployeeID TEXT NOT NULL PRIMARY KEY,
First_Name TEXT,
Last_Name TEXT,
Street TEXT,
Zip INTEGER,
Phone_INTEGER INTEGER,
Start_date INTEGER,
End_date INTEGER,
Salary INTEGER,
FOREIGN KEY (Zip) REFERENCES Zipcodes (Zip)
);

INSERT INTO Staff (EmployeeID, First_Name, Last_Name, Street, Zip, Phone_INTEGER, Start_date, End_date, Salary)
VALUES (‘E967’, ‘John’, ‘Smith’,’123 Main St’, 10007, ‘(212) 644-1234’, 20180508, NULL, 50000);
INSERT INTO Staff (EmployeeID, First_Name, Last_Name, Street, Zip, Phone_INTEGER, Start_date, End_date, Salary)
VALUES (‘E462’, ‘Mary’, ‘Johnson’, ‘456 Park Ave’, 45678, ‘(917) 832-6192’, 20180502, NULL, 55000);
INSERT INTO Staff (EmployeeID, First_Name, Last_Name, Street, Zip, Phone_INTEGER, Start_date, End_date, Salary)
VALUES (‘E764’, ‘David’, ‘Brown’, ‘789 Broadway’, 98376, ‘(682) 881-3640’, 20180514, 20320117, 25000);

EmployeeID	First_Name	Last_Name	Street	Zip	Phone_INTEGER	Start_date	End_date	Salary
E967	John	Smith	123 Main St	10007	(212) 644-1234	20180508		50000
E462	Mary	Johnson	456 Park Ave	45678	(917) 832-6192	20180502		55000
E764	David	Brown	789 Broadway	98376	(682) 881-3640	20180514	20230117	35000
E325	Sarah	Wilson	101 Fifth Ave	78901	(718) 235-4567	20180518		25000
E784	James	Anderson	222 Second St	45678	(347) 171-6437	20180524		45000
E124	Emily	Martinez	333 Third Ave	10006	(212) 358-2473	20180526		38000
E735	Ryan	Clark	444 Fourth St	45678	(718) 731-5736	20180508		40000
E759	Amy	Lee	555 Fifth Ave	10012	(359) 692-8901	20180527		42000

CREATE TABLE Orders (
OrderID TEXT NOT NULL PRIMARY KEY,
Date_of_order INTEGER,
Order_details TEXT,
EmployeeID TEXT,
FOREIGN KEY (EmployeeID) REFERENCES Staff (EmployeeID)
);

INSERT INTO Orders (OrderID, Date_of_order, Order_details, EmployeeID)
VALUES (‘O1001’, 20220101, ‘Set of dumbbells’, ‘E764’);
INSERT INTO Orders (OrderID, Date_of_order, Order_details, EmployeeID)
VALUES (‘O2378’, 20220105, ‘Yoga mat’, ‘E124’);
INSERT INTO Orders (OrderID, Date_of_order, Order_details, EmployeeID)
VALUES (‘O4729’, 20220110, ‘Treadmill’, ‘E735’);

OrderID	Date_of_order	Order_details	EmployeeID
O1001	20220101	Set of dumbbells	E764
O2378	20220105	Yoga mat	E124
O4729	20220110	Treadmill	E735
O3712	20220115	Set of dumbbells	E784
O5827	20220120	Treadmill	E735
O7382	20220123	Yoga mat	E967

CREATE TABLE Order_Detail (
Order_Detail_ID TEXT NOT NULL PRIMARY KEY,
Quantity INTEGER,
Total INTEGER,
ProductID TEXT,

OrderID TEXT,
FOREIGN KEY (OrderID) REFERENCES Orders (OrderID),
FOREIGN KEY (ProductID) REFERENCES Products (ProductID)
);

INSERT INTO Order_Detail (Order_Detail_ID, Quantity, Total, ProductID, OrderID)
VALUES ('OD45', 2, 60, 'P101', 'O1001');
INSERT INTO Order_Detail (Order_Detail_ID, Quantity, Total, ProductID, OrderID)
VALUES ('OD76', 5, 50, 'P654', 'O2378');
INSERT INTO Order_Detail (Order_Detail_ID, Quantity, Total, ProductID, OrderID)
VALUES ('OD83', 3, 100, 'P902', 'O4729');

Order_Detail_ID	Quantity	Total	ProductID	OrderID
OD45	2	60	P101	O1001
OD76	5	50	P654	O2378
OD83	3	100	P902	O4729
OD78	4	90	P765	O3712
OD86	2	100	P753	O5827
OD36	7	60	P954	O7382

CREATE TABLE Class_type (
ClassID TEXT NOT NULL PRIMARY KEY,
Class_name TEXT,
Equipment_required TEXT
);

INSERT INTO Class_type (ClassID, Class_name, Equipment_required)
VALUES ('C732', 'Yoga', 'Yoga mats');
INSERT INTO Class_type (ClassID, Class_name, Equipment_required)
VALUES ('C742', 'Spinning', 'Spinning bikes');
INSERT INTO Class_type (ClassID, Class_name, Equipment_required)
VALUES ('C893', 'Body Pimp', 'Dumbbells');

ClassID	Class_name	Equipment_required
C732	Yoga	Yoga mats
C742	Spinning	Spinning bikes
C893	Body Pump	Dumbbells
C382	Pilates	Yoga mats
C193	Zumba	None
C745	Boxing	Gloves and Punching bags

CREATE TABLE Rooms (
Room_number TEXT NOT NULL PRIMARY KEY,
Capacity INTEGER,
Room_equipment VARCHAR (300),
Room_type TEXT
);

INSERT INTO Rooms (Room_number, Capacity, Room_equipment, Room_type)
VALUES (‘R1’, 20, ‘Treadmills’, ‘Cardio’);
INSERT INTO Rooms (Room_number, Capacity, Room_equipment, Room_type)
VALUES (‘R2’, 10, ‘Free weights’, ‘Strength’);
INSERT INTO Rooms (Room_number, Capacity, Room_equipment, Room_type)
VALUES (‘R3’, 15, Ellipticals’, ‘Cardio’);

Room_number	Capacity	Room_equipment	Room_type
R1	20	Treadmills	Cardio
R2	10	Free weights	Strength
R3	15	Ellipticals	Cardio
R4	25	Rowing machines	Cardio
R5	20	Weight machines	Strength

CREATE TABLE Sessions (
SessionID TEXT NOT NULL PRIMARY KEY,
Session_date DATE,
Start_time TIME,
End_time TIME,

Amt_participants INTEGER,
Time_length INTEGER,
Session_cost INTEGER,
Room_number TEXT,
FOREIGN KEY (Room_number) REFERENCES Rooms (Room_INTEGER)
);

INSERT INTO Sessions (SessionID, Session_date, Start_time, End_time, Amt_participants, Time_Length, Session_cost, Room_number)
VALUES (‘S0632’, 20210502, '10:00:00', '11:30:00', 16, ‘1.5 hours’, 20, ‘R1’);
INSERT INTO Sessions (SessionID, Session_date, Start_time, End_time, Amt_participants, Time_Length, Session_cost, Room_number)
VALUES (‘S0485’, 20230922, '18:00:00', '19:30:00', 10, ‘1.5 hours’, 25, ‘R2’);
INSERT INTO Sessions (SessionID, Session_date, Start_time, End_time, Amt_participants, Time_Length, Session_cost, Room_number)
VALUES (‘S0574, 20191016, '11:00:00', '12:30:00', 7, ‘1.5 hours’, 15, ‘R3’);

SessionID	Session_date	Start_time	End_time	Amt_participants	Time_length	Session_cost	Room_number
S0632	20210502	10:00:00	11:30:00	16	1.5 hours	20	R1
S0485	20230922	18:00:00	19:30:00	10	1.5 hours	25	R2
S0574	20191016	11:00:00	12:30:00	7	1.5 hours	15	R3
S0237	20220308	9:00:00	10:30:00	18	1.5 hours	30	R4
S0372	20190516	17:00:00	18:30:00	14	1.5 hours	25	R5
S0683	20220704	14:00:00	15:30:00	8	1.5 hours	20	R2
S5732	20230514	9:00:00	10:30:00	5	1.5 hours	15	R3
S0375	20231014	16:00:00	17:30:00	17	1.5 hours	25	R4
S9573	20191220	13:00:00	14:30:00	13	1.5 hours	30	R1
S0674	20220829	11:00:00	12:30:00	8	1.5 hours	25	R2

CREATE TABLE Trainer_Class (
EmployeeID TEXT NOT NULL,
ClassID TEXT NOT NULL,
SessionID TEXT NOT NULL,
Start_date INTEGER,
End_date INTEGER,
FOREIGN KEY (EmployeeID) REFERENCES Staff (EmployeeID),
FOREIGN KEY (ClassID) REFERENCES Class_type (ClassID),
FOREIGN KEY (SessionID) REFERENCES Sessions (SessionID),
PRIMARY KEY (EmployeeID,ClassID,SessionID)
);

INSERT INTO Trainer_Class (EmployeeID, ClassID, SessionID, Start_date, End_date)

VALUES (‘E967’, ‘C732’, ‘S0632’, 20210502, 20210802);
INSERT INTO Trainer_Class (EmployeeID, ClassID, SessionID, Start_date, End_date)
VALUES (‘E764’, ‘C893’, ‘S0485’, 20230922, 20231222);
INSERT INTO Trainer_Class (EmployeeID, ClassID, SessionID, Start_date, End_date)
VALUES (‘E124’, ‘C193’, ‘S0574’, 20191016, 20191116);

EmployeeID	ClassID	SessionID	Start_date	End_date
E967	C732	S0632	20210502	20210802
E764	C893	S0485	20230922	20231222
E124	C193	S0574	20191016	20191116
E967	C742	S0237	20220308	20221008
E967	C732	S0372	20190516	20190916
E759	C193	S0674	20191016	20191116

CREATE TABLE Members (
MemberID TEXT NOT NULL PRIMARY KEY,
First_name TEXT,
Last_name TEXT,
Street TEXT,
Zip INTEGER,
Email TEXT,
Phone_INTEGER INTEGER,
Membership_start_date DATE,
FOREIGN KEY (Zip) REFERENCES Zipcodes (Zip)
);

INSERT INTO Members (MemberID, First_name, Last_name, Street, Zip, Email,
Phone_INTEGER, Membership_start_date)
VALUES (‘M835’, ‘John’, ‘Doe’, ‘123 Main St’, 12345, ‘john.doe@example.com’,
‘917-951-8503’, 20180607);
INSERT INTO Members (MemberID, First_name, Last_name, Street, Zip, Email,
Phone_INTEGER, Membership_start_date)
VALUES (‘M924’, ‘Jane’, ‘Smith’, ‘456 Broadway’, 54321, ‘jane.smith@example.com’,
‘718-396-4710’, 20180716);
INSERT INTO Members (MemberID, First_name, Last_name, Street, Zip, Email,
Phone_INTEGER, Membership_start_date)

VALUES (‘M852’, ‘Tom’, ‘Brown’, ‘789 Oak Ave’, 67890, ‘tom.brown@example.com’, ‘718-841-8492’, 20180529);

MemberID	First_name	Last_name	Street	Zip	Email	Phone_INTEGER	Membership_start_date
M835	John	Doe	123 Main St	12345	john.doe@example.com	917-951-8503	20180607
M924	Jane	Smith	456 Broadway	54321	jane.smith@example.com	718-396-4710	20180716
M852	Tom	Brown	789 Oak Ave	67890	tom.brown@example.com	718-841-8492	20180529
M754	Amy	Garcia	456 Elm St	56789	amy.garcia@example.com	917-582-2958	20180914
M124	Chris	Lee	789 Pine St	98376	chris.lee@example.com	347-850-9357	20180810
M854	Katie	Wong	321 Oak St	23456	katie.wong@example.com	718-682-9811	20180705
M234	Michael	Brown	987 High St	34567	michael.brown@example.com	646-981-0491	20180817
M975	Sarah	Kim	345 Maple Ave	45678	sarah.kim@example.com	646-918-5214	20180815
M656	David	Lopez	567 Cherry St	78901	david.lopez@example.com	718-529-1246	20180701
M739	Emily	Jones	234 Park Ave	12345	emily.jones@example.com	555-124-6921	20180912
M023	Grace	Lin	543 Elm St	67890	grace.lin@example.com	555-634-4192	20180804
M865	Ryan	Gomez	678 Maple Ave	34567	ryan.gomez@example.com	718-638-1923	20181204
M235	Michelle	Lee	123 Oak Ave	45678	michelle.lee@example.com	580-682-6823	20180814
M546	Brandon	Park	456 Pine St	78901	brandon.park@example.com	808-124-8123	20180712

CREATE TABLE Invoice (
Invoice_num TEXT NOT NULL PRIMARY KEY,
Due_date INTEGER,
Amount INTEGER,
MemberID TEXT,
FOREIGN KEY (MemberID) REFERENCES Members (MemberID)
);

INSERT INTO Invoice (Invoice_num, Due_date, Amount, MemberID)
VALUES (‘I125’, 20210418, 150, ‘M924’);
INSERT INTO Invoice (Invoice_num, Due_date, Amount, MemberID)
VALUES (‘I823’, 20190403, 175, ‘M656’);
INSERT INTO Invoice (Invoice_num, Due_date, Amount, MemberID)
VALUES (‘I964’, 20210516, 200, ‘M754’);

Invoice_num	Due_date	Amount	MemberID
I125	20210418	150	M924
I823	20190403	175	M656
I964	20210516	200	M754
I238	20230412	275	M854
I974	20220703	300	M975
I365	20220722	325	M975
I865	20191205	350	M924
I832	20220607	375	M924
I567	20191130	200	M854
I024	20230830	400	M739
I902	20220304	200	M865
I784	20220828	145	M854

CREATE TABLE Session_reservation (
MemberID TEXT NOT NULL,
SessionID TEXT NOT NULL,
Reservation_date DATE,
Cancellation_date DATE,
Invoice_num TEXT,
FOREIGN KEY (Invoice_num) REFERENCES Invoice (Invoice_num),
FOREIGN KEY (SessionID) REFERENCES Sessions (Session_ID),
FOREIGN KEY (MemberID) REFERENCES Members (MemberID),
PRIMARY KEY (MemberID,SessionID)
);

INSERT INTO Session_reservation (MemberID, SessionID, Reservation_date, Cancellation_date, Invoice_num)
VALUES ('M924', 'S0632', 20210416, NULL, 'I125');
INSERT INTO Session_reservation (MemberID, SessionID, Reservation_date, Cancellation_date, Invoice_num)
VALUES ('M656', 'S0372', 20190401, NULL, 'I823');
INSERT INTO Session_reservation (MemberID, SessionID, Reservation_date, Cancellation_date, Invoice_num)
VALUES ('M754', 'S0632', 20210514, NULL, 'I964');

MemberID	SessionID	Reservation_date	Cancellation_date	Invoice_num
M924	S0632	20210416		I125
M656	S0372	20190401		I823
M754	S0632	20210514		I964
M234	S0485	20230920	20230930	
M546	S0372	20190420	20190501	
M854	S5732	20230410		I238
M975	S0683	20220701		I974
M975	S0375	20220720		I365
M924	S9573	20191203		I865
M924	S0683	20220605		I832
M854	S9573	20191128		I567
M739	S0485	20230828		I024
M865	S0237	20220302		I902
M854	S0674	20220828		I784

CREATE TABLE Credit_card (
Credit_card_num INTEGER PRIMARY KEY,
Card_type TEXT,
Security_code INTEGER,
Expiration_date INTEGER
);

INSERT INTO Credit_card (Credit_card_num, Card_type, Security_code, Expiration_date)
VALUES (333322000000000, 'Visa', 432, 20241021);
INSERT INTO Credit_card (Credit_card_num, Card_type, Security_code, Expiration_date)
VALUES (1111222233334444, 'Visa', 567, 20240713);
INSERT INTO Credit_card (Credit_card_num, Card_type, Security_code, Expiration_date)
VALUES (1234567890123456, 'Visa', 123, 20240525);

Credit_card_num	Card_type	Security_code	Expiration_date
333322000000000	Visa	432	20241021
1111222233334444	Visa	567	20240713
1234567890123456	Visa	123	20240525
2222333344445555	Mastercard	234	20240813
4444333322221111	Discover	321	20240704
4444333344445555	American express	974	20241215
5555444433332222	Mastercard	678	20240110
5555666677778888	American Express	789	20240730
6666777788889999	American Express	432	20240607
7777666655554444	Discover	420	20240530
7777888899991111	Discover	876	20240906
9876543210987654	Mastercard	456	20241016
9999888877776666	Visa	345	20240625

CREATE TABLE Payment (
TransactionID TEXT NOT NULL PRIMARY KEY,
Payment_date DATE,
Credit_card_num INTEGER,
Street TEXT,
Zip INTEGER,
Invoice_num TEXT,
FOREIGN KEY (Invoice_num) REFERENCES Invoice (Invoice_num),
FOREIGN KEY (Zip) REFERENCES Zipcodes (Zip),

FOREIGN KEY (Credit_card_num) REFERENCES Credit_card (Credit_card_num)
);

INSERT INTO Payment (TransactionID, Payment_date, Credit_card_num, Street, Zip, Invoice_num)
VALUES (‘T436’, 20210417, 1234567890123456, ‘123 Main St’, 12345, ‘I125’);
INSERT INTO Payment (TransactionID, Payment_date, Credit_card_num, Street, Zip, Invoice_num)
VALUES (‘T896’, 20190403, 9876543210987654, ‘456 Broadway’, 54321, ‘I823’);
INSERT INTO Payment (TransactionID, Payment_date, Credit_card_num, Street, Zip, Invoice_num)
VALUES (‘T867’, 20210515, 5555666677778888, ‘789 Oak Ave’, 67890, ‘I964’);

TransactionID	Payment_date	Credit_card_num	Street	Zip	Invoice_num
T436	20210417	1234567890123456	123 Main St	12345	I125
T896	20190403	9876543210987654	456 Broadway	54321	I823
T867	20210515	5555666677778888	789 Oak Ave	67890	I964
T823	2023041	4444333322221111	456 Elm St	56789	I238
T865	20220703	1111222233334444	789 Pine St	98376	I974
T129	20220721	2222333344445555	321 Oak St	23456	I365
T845	20191204	6666777788889999	987 High St	34567	I865
T970	20220606	7777888899991111	345 Maple Ave	45678	I832
T456	20191130	9999888877776666	567 Cherry St	78901	I567
T346	20220901	333322000000000	876 Main St	54321	I784

SCENARIOS

1. **A new Pilates Instructor is joining DBMS Fitness Club: Update the database to include the new Instructor and the training session that he will teach.**

INSERT INTO Staff (EmployeeID, First_Name, Last_Name, Street, Zip, Phone_INTEGER, Start_date, End_date, Salary)
VALUES ('E829', 'Jefferson', 'Veliz', '288 First Ave', '78901', '(932) 912-3948', '20230630', NULL, 40000);

EmployeeID	First_Name	Last_Name	Street	Zip	Phone_INTEGER	Start_date	End_date	Salary
E967	John	Smith	123 Main St	10007	(212) 644-1234	20180508		50000
E462	Mary	Johnson	456 Park Ave	45678	(917) 832-6192	20180502		55000
E764	David	Brown	789 Broadway	98376	(682) 881-3640	20180514	20230117	35000
E325	Sarah	Wilson	101 Fifth Ave	78901	(718) 235-4567	20180518		25000
E784	James	Anderson	222 Second St	45678	(347) 171-6437	20180524		45000
E124	Emily	Martinez	333 Third Ave	10006	(212) 358-2473	20180526		38000
E735	Ryan	Clark	444 Fourth St	45678	(718) 731-5736	20180508		40000
E759	Amy	Lee	555 Fifth Ave	10012	(359) 692-8901	20180527		42000
E829	Jefferson	Veliz	288 First Ave	78901	(932) 912-3948	20230630		40000

INSERT INTO Trainer_Class (EmployeeID, ClassID, SessionID, Start_date, End_date)
VALUES ('E829', 'C732', 'S0973', 20230705, 20231205);

EmployeeID	ClassID	SessionID	Start_date	End_date
E967	C732	S0632	20210502	20210802
E764	C893	S0485	20230922	20231222
E124	C193	S0574	20191016	20191116
E967	C742	S0237	20220308	20221008
E967	C732	S0372	20190516	20190916
E759	C193	S0674	20191016	20191116
E829	C732	S0973	20230705	20231205

INSERT INTO Sessions (SessionID, Session_date, Start_time, End_time, Amt_participants, Time_Length, Session_cost, Room_number)
VALUES ('S0973', 20230705, '11:00:00', '12:30:00', 25, '1.5 hours', 25, 'R2');

SessionID	Session_date	Start_time	End_time	Amt_participants	Time_length	Session_cost	Room_number
S0632	20210502	10:00:00	11:30:00	16	1.5 hours	20	R1
S0485	20230922	18:00:00	19:30:00	10	1.5 hours	25	R2
S0574	20191016	11:00:00	12:30:00	7	1.5 hours	15	R3
S0237	20220308	9:00:00	10:30:00	18	1.5 hours	30	R4
S0372	20190516	17:00:00	18:30:00	14	1.5 hours	25	R5
S0683	20220704	14:00:00	15:30:00	8	1.5 hours	20	R2
S5732	20230514	9:00:00	10:30:00	5	1.5 hours	15	R3
S0375	20231014	16:00:00	17:30:00	17	1.5 hours	25	R4
S9573	20191220	13:00:00	14:30:00	13	1.5 hours	30	R1
S0674	20220829	11:00:00	12:30:00	8	1.5 hours	25	R2
S0973	20230705	11:00:00	12:30:00	25	1.5 hours	25	R2

2. **DBMS wants to offer a special promotion to certain customers: List the name of the members who pay more than $250 and give the members a 10% discount for the month of January.**

```
SELECT t.First_name,t.Last_Name, t.BeforeDiscount,
(t.BeforeDiscount-(t.BeforeDiscount*.1)) AS AfterDiscount
FROM (SELECT m.First_name, m.Last_name, m.MemberID, i.Amount AS
BeforeDiscount
FROM Members m, Invoice i
WHERE m.MemberID=i.MemberID AND Amount> 300
GROUP BY m.Last_name) t;
```

First_name	Last_name	BeforeDiscount	AfterDiscount
Emily	Jones	400	360.0
Sarah	Kim	325	292.5
Jane	Smith	350	315.0

3. **DBMS wants to reduce shipping costs from suppliers outside of NY as well as support local businesses who have been struggling since the pandemic. What suppliers are located outside of NY and what kind of products do they provide? This will allow owners to know what supplier they need to find locally.**

```
SELECT t.SupplierID, t.Supplier_Name, t.Location,p.Category, p.Product_Name,
p.Product_Description
FROM Products p, (SELECT SupplierID, Supplier_name, z.state_usa AS Location
FROM Suppliers s, Zipcodes z
WHERE s.Zip=z.Zip AND state_usa!='NY') t
WHERE t.SupplierID=p.SupplierID;
```

SupplierID	Supplier_name	Location	Category	Product_name	Product_description
S768	Fitness Equipment	IL	Weights	Kettlebell set	Set of 3 kettlebells with stand
S784	Sports Gear	IL	Weights	Resistance band	Pack of 5 different resistance bands
S348	Fitness Plus	IL	Cardio	Exercise bike	Recumbent exercise bike with LCD screen

4. **DBMS is planning to do some construction in Room 3 and would like to know what classes are held in Room 2 since it is the largest room. They would like to know what classes are currently held there and which employees lead those classes?**

```
SELECT t.employeeID, s.first_name, s.last_name, ss.sessionID, c.class_name
FROM Trainer_Class t, Staff s, Sessions ss, Class_type c
WHERE  t.employeeID = s.EmployeeID AND ss.Sessionid = t.Sessionid AND t.classID
= c.classID
AND  ss.room_number = 'R2';
```

EmployeeID	First_Name	Last_Name	SessionID	Class_name
E759	Amy	Lee	S0674	Zumba
E764	David	Brown	S0485	Body Pump
E829	Jefferson	Veliz	S0973	Yoga

5. **Customers are complaining that the sessions for using ellipticals are always full and unavailable. DBMS would like to know which sessions utilize the room with ellipticals and how many sessions are currently offered?**

```
SELECT sessions.SessionID, sessions.room_number, rooms.room_equipment
FROM sessions, rooms
WHERE sessions.room_number = rooms.Room_number AND rooms.room_equipment =
'Ellipticals';
```

SessionID	Room_number	Room_equipment
S0574	R3	Ellipticals
S5732	R3	Ellipticals

6. **DBMS would like to add more sessions at their gym. Which class is the most popular class among members?**

```
SELECT c.Class_name, COUNT(DISTINCT sr.MemberID) AS MemberCount
FROM Class_type c
INNER JOIN Trainer_Class tc ON c.ClassID = tc.ClassID
INNER JOIN Session_reservation sr ON tc.SessionID = sr.SessionID
GROUP BY c.Class_name
ORDER BY MemberCount DESC
LIMIT 1;
```

Class_name	MemberCount
Yoga	4

CONCLUSION

Working on the DBMS project has been a challenging yet rewarding experience for our team. We encountered a few difficulties during the project, such as deciding on the appropriate tables and their attributes, as well as ensuring that each table had the necessary foreign keys. However, with team collaboration and thorough research, we were able to successfully create a database system that met the client's requirements.

One of the easiest steps was creating the relationship sentences since they provided a clear guideline for developing the tables. We also learned that effective communication and coordination between team members are essential for the success of a project.

We believe that the proposed benefits of the new system can be realized by DBMS. The system will allow the fitness club to efficiently manage its resources, staff, and inventory all through one uniform system. It will also enable DBMS to keep track of members' payment information and session reservations, which will allow them to process charges for monthly memberships and class sessions attended accurately. Additionally, the system will allow DBMS to keep track of which staff member will teach each class, the timing of each class session, in which training room, and the duration of the class. This will minimize overcrowding in rooms, avoid overlapping class sessions, and minimize downtime. For example, our queries can be used to extract important data from a database that can help a business owner make informed decisions. For example, adding a new Pilates instructor would require updating the staff and session tables. If DBMS wants to reduce shipping costs by sourcing only local suppliers, they can query the database to find which suppliers they use outside of NY and who they can use locally instead. Identifying the top 5 members with the most outstanding invoice amount due and their location can help a business owner prioritize collection efforts. Queries can also help identify which sessions utilize a specific room or equipment, determine the most popular class among members, and track employee schedules and class assignments. Additionally, a special promotion can be offered to members who pay more than $250 by listing their names and providing a 10% discount for the month of January.

Overall, we are pleased to have completed this project successfully, and we believe that the new system will help DBMS run its operations efficiently. Our team would have ensured that we had more frequent check-ins with the client to ensure we were meeting their expectations. Nonetheless, we are confident that the database system will help the fitness club to meet its goals of efficiently managing its operations, encouraging customers to take part in all their services and activities, and avoiding errors like overbooking.

Database Link: https://replit.com/@CISFITNESS23/FITNESS-DATABASE?v=1