**Database Management Systems of**

**Mindful Body Fitness Company**

Mindful Body Fitness Company is planning to design a data model to hold information relating to their programs, coaches, exercise logs, and their customers. As a database designer for Mindful Body Fitness, you need to ensure that each Exercise Log has a unique ID number, Date, Exercise category, Exercise subcategory, Time spent, and calories burnt.

Each customer has a unique Cust\_number and their first name, last name, address, program number, program tile, and Payment must be recorded. Every program has a program number (PNum) that uniquely identifies the program.

The program title, StartDate, Finish Date, and Price must also be recorded. Each coach has a unique ID number. Their first name, last name, date of birth, address, contact No, user name, and password must be recorded.

A coach can work on only one program. A program can have one or more coaches associated with it. One customer can enroll in only one program at one time.

I have developed a database management system to maintain the database of Mindful Body Fitness Company. I have used SQL and MySQL workbench and draw.io to create the database.

Below I have done all the steps one by one.

**Step-1)** Identifying and listing the entities described in all user requirements

given above.

The assumptions I have made- For payment record, payment ID, payment date, transaction mode, transaction ID needs to be recorded. Each customer takes a program, and there is an exercise log. One or multiple coaches can be in the same program so adding coach ID and exercise no to track which coach is attached to program.

Exercise Log (Ex\_ID\_number, Date, Exercise\_category, Exercise\_sub\_category, Time\_spent, calories\_burnt)

Customer (Cust\_number, first\_name, last\_name, address, program\_number, program\_tile, Payment\_id, payment\_date, transaction\_mode, transaction\_id, program\_number, Coach\_ID, Ex\_id\_number)

program (program\_number, program\_title, Start\_Date, Finish\_Date, Price, Coach\_ID, Ex\_id\_number)

coach (Coach\_ID, first\_name, last\_name, date\_of\_birth, address, contact\_No, user\_name, password)

**Step-2)** Now adding attributes to these entities and represent them as a collection of tables and attributes that is the database schema.

**Exercise Log program**

|  |  |
| --- | --- |
| program\_number  primary key | Int |
| program\_title | Varchar 50 |
| Start\_Date | date |
| Finish\_Date | Date |
| Price | Int |
| Coach\_ID | Int |
| Ex\_id\_number | Int |

**Coach**

|  |  |
| --- | --- |
| Coach\_ID  primary key | Int |
| first\_name | Varchar 50 |
| last\_name | Varchar 50 |
| date\_of\_birth | Date |
| address | Varchar 200 |
| contact\_No | Int |
| user\_name | Varchar 20 |
| password | Varchar 20 |

|  |  |
| --- | --- |
| Ex\_ID\_number  Primary key | Int |
| Date | Date |
| Exercise\_category | Varchar 10 |
| Exercise\_sub\_category | Varchar 10 |
| Time\_spent | Int |
| calories\_burnt | Int |

**Customer**

|  |  |
| --- | --- |
| Cust\_number  Primary key | Int |
| first\_name | Varchar 50 |
| last\_name | Varchar 50 |
| address | Varchar 200 |
| program\_number | Int |
| program\_tile | Varchar 50 |
| Payment\_id | Int 50 |
| payment\_date | Date |
| transaction\_mode | Varchar 10 |
| transaction\_id | Int |
| program\_number | Int |
| Coach\_ID | Int |
| Ex\_id\_number | Int |

**Step 3)** Identifying three business rules that describe the relationships between entities.

The business rules are

1)A Coach can work on only one program.

2) A program can have one or more coaches associated with it.

3)One customer can enroll in only one program at one time.

**Step-4)** Now I have determined all the functional dependencies of the above database schema.

The functional dependencies are-

Cust\_Num → Cust\_First\_Name

program\_number → Coach\_ID

program\_number → program\_title

Coach\_ID → first\_name

Coach\_ID → user\_name

Customer\_Num → program\_number

Coach\_ID → Customer\_Num

Cust\_number → Payment\_id

Step-5) Then normalisation of these tables. I have made the normalization to 3NF. And give an explanation for every step in the normalization, and which functional dependency causes it.

Payment record is partially dependent on customer number. So, a separate table is made for payment records.

**payment**

**Customer**

|  |  |
| --- | --- |
| Cust\_number  Primary key | Int |
| first\_name | Varchar 50 |
| last\_name | Varchar 50 |
| address | Varchar 200 |
| program\_number | Int |
| program\_tile | Varchar 50 |
| Coach\_ID | Int |
| Ex\_id\_number | Int |
| Payment\_id | Int |

|  |  |
| --- | --- |
| Payment\_id  Primary key | Int |
| Cust\_number | Int |
| payment\_date | Date |
| transaction\_mode | Varchar 10 |
| transaction\_id | Int |

User name is partially dependent on coach. So made a separate table for user name.

**Coach**

**User name**

|  |  |
| --- | --- |
| Coach\_ID  primary key | Int |
| user\_name | Varchar 20 |
| password | Varchar 20 |

|  |  |
| --- | --- |
| Coach\_ID  primary key | Int |
| first\_name | Varchar 50 |
| last\_name | Varchar 50 |
| date\_of\_birth | Date |
| address | Varchar 200 |
| contact\_No | Int |
| user\_name | Varchar 20 |

A program can have one or more coaches associated with it. So is transient dependency.

**program**

**Program-coach**

|  |  |
| --- | --- |
| program\_number  primary key | Int |
| Coach\_ID | Int |
| Ex\_id\_number | Int |

|  |  |
| --- | --- |
| program\_number  primary key | Int |
| program\_title | Varchar 50 |
| Start\_Date | date |
| Finish\_Date | Date |
| Price | Int |

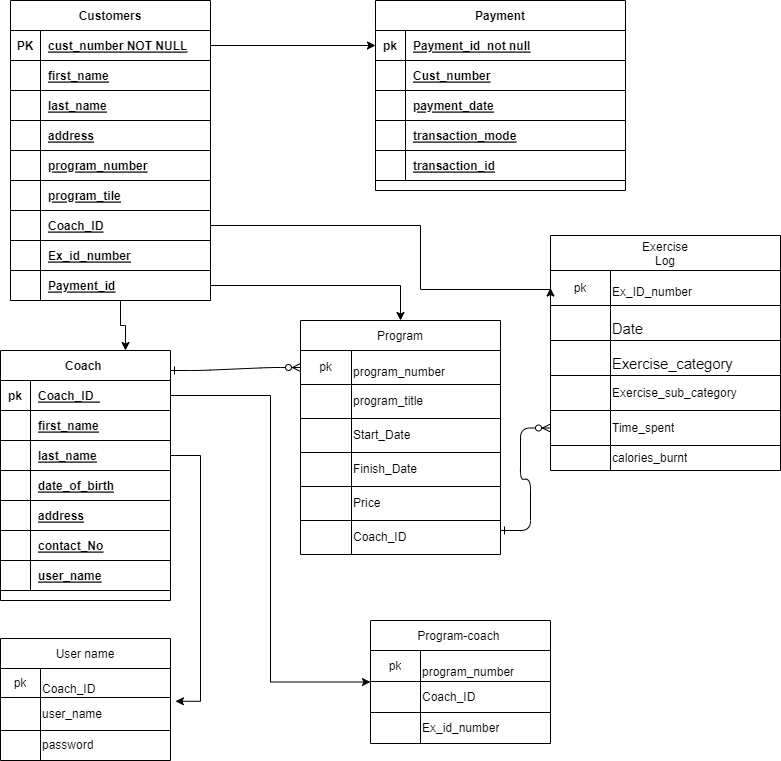
**Exercise Log**

|  |  |
| --- | --- |
| Ex\_ID\_number  Primary key | Int |
| Date | Date |
| Exercise\_category | Varchar 10 |
| Exercise\_sub\_category | Varchar 10 |
| Time\_spent | Int |
| calories\_burnt | Int |

Now the final normalised database schema is ready. Also, I have mentioned the primary key, and datatypes of every entity for all the tables.

**Step-6)** Representing the structure of this database visually by using an entity-relationship (E-R) diagram.

I have made is schema using Draw.io. the file link is given at the end of this report.



**Step-6)** Now creating this table in SQL using MySQL workbench. Also Populating this table using sample data.

The SQL is written below and the screenshot of the output panel is also attached here.

*--Creating a schema in SQL server--*

create schema Mindful\_Body\_Fitness;

use Mindful\_Body\_Fitness;

*--Creating customer table--*

CREATE TABLE Customer(

Cust\_number int,

first\_name Varchar(50),

last\_name Varchar(50),

address Varchar(200),

program\_number Int,

program\_tile Varchar(50),

Coach\_ID Int,

Ex\_id\_number Int,

Payment\_id Int

);

*--Inserting sample data--*

INSERT INTO Customer (Cust\_number, first\_name, last\_name, address, program\_number, program\_tile, Coach\_ID, Ex\_id\_number, Payment\_id)

VALUES

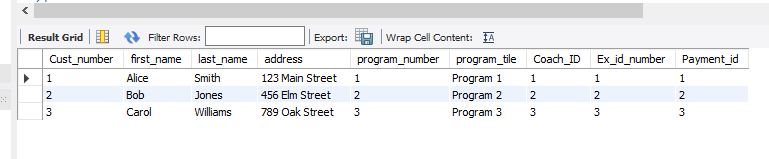
(1, 'Alice', 'Smith', '123 Main Street', 1, 'Program 1', 1, 1, 1),

(2, 'Bob', 'Jones', '456 Elm Street', 2, 'Program 2', 2, 2, 2),

(3, 'Carol', 'Williams', '789 Oak Street', 3, 'Program 3', 3, 3, 3);

select \* from customer;

**Output-**

****

*--Creating Payment table--*

CREATE TABLE payment(

Payment\_id int,

Cust\_number Int ,

payment\_date Date,

transaction\_mode Varchar (100),

transaction\_id Int

);

*--Inserting data into payment table--*

INSERT INTO payment (Payment\_id, Cust\_number, payment\_date, transaction\_mode, transaction\_id)

VALUES

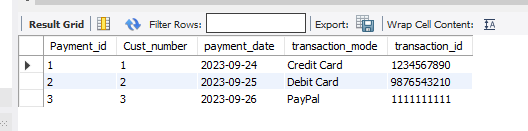
(1, 1, '2023-09-24', 'Credit Card', 1234567890),

(2, 2, '2023-09-25', 'Debit Card', 9876543210),

(3, 3, '2023-09-26', 'PayPal', 1111111111);

select \* from payment;

**Output-**



*--Creating coach table--*

CREATE TABLE Coach(

Coach\_ID int,

first\_name Varchar(50),

last\_name Varchar (50),

date\_of\_birth Date,

address Varchar(200),

contact\_No varchar(100),

user\_name Varchar(20)

);

*--inserting data into coach table--*

INSERT INTO Coach (Coach\_ID, first\_name, last\_name, date\_of\_birth, address, contact\_No, user\_name)

VALUES

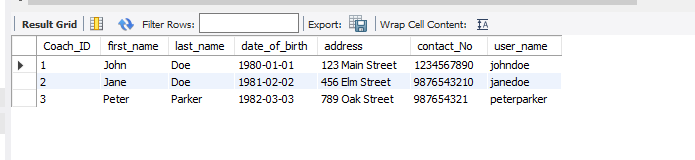
(1, 'John', 'Doe', '1980-01-01', '123 Main Street', 1234567890, 'johndoe'),

(2, 'Jane', 'Doe', '1981-02-02', '456 Elm Street', 9876543210, 'janedoe'),

(3, 'Peter', 'Parker', '1982-03-03', '789 Oak Street', 0987654321, 'peterparker');

select \* from coach;

**Output-**

****

*--creating table User\_name--*

CREATE TABLE User\_name(

Coach\_ID int,

user\_name Varchar(20),

password Varchar(20));

*--inserting data into user\_name table--*

INSERT INTO User\_name (Coach\_ID, user\_name, password)

VALUES

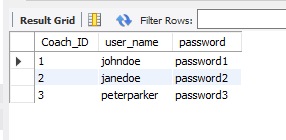
(1, 'johndoe', 'password1'),

(2, 'janedoe', 'password2'),

(3, 'peterparker', 'password3');

select \* from User\_name;

**Output-**



*--Creating table program--*

CREATE TABLE program(

program\_number int,

program\_title Varchar(50),

Start\_Date date,

Finish\_Date Date,

Price Int

);

*--inserting data into program table--*

INSERT INTO program (program\_number, program\_title, Start\_Date, Finish\_Date, Price)

VALUES

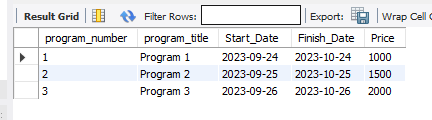
(1, 'Program 1', '2023-09-24', '2023-10-24', 1000),

(2, 'Program 2', '2023-09-25', '2023-10-25', 1500),

(3, 'Program 3', '2023-09-26', '2023-10-26', 2000);

select \* from program;

**Output-**



*--Creating table program\_coach--*

CREATE TABLE Program\_coach (

program\_number int,

Coach\_ID Int,

Ex\_id\_number Int

);

*--inserting data into program\_coach table--*

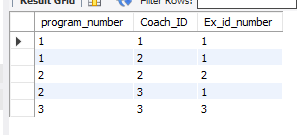
INSERT INTO Program\_coach (program\_number, Coach\_ID, Ex\_id\_number)

VALUES

(1, 1, 1), (1, 2, 1), (2, 2, 2), (2, 3, 1), (3, 3, 3);

select \* from Program\_coach;

**Output-**



*--Creating table exercise\_log--*

CREATE TABLE Exercise\_Log(

Ex\_ID\_number int,

Date date,

Exercise\_category varchar(10),

Exercise\_sub\_category Varchar(10),

Time\_spent Int(10),

calories\_burnt Int(10)

);

*--inserting data into Exercise\_Log table--*

INSERT INTO Exercise\_Log (Ex\_ID\_number, Date, Exercise\_category, Exercise\_sub\_category, Time\_spent, calories\_burnt)

VALUES

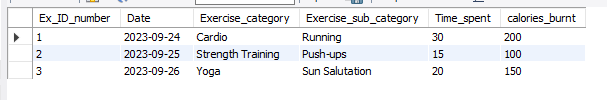
(1, '2023-09-24', 'Cardio', 'Running', 30, 200),

(2, '2023-09-25', 'Strength Training', 'Push-ups', 15, 100),

(3, '2023-09-26', 'Yoga', 'Sun Salutation', 20, 150);

select \* from Exercise\_Log;

**Output-**



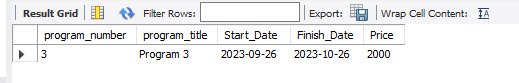
Now the database is ready to explore. We can now input more data in this database schema, and explore various queries in this. Some of the aggregate function and join function is shown below.

1) I have written a SQL query to show the program number, program title, and price of that program which price is greater than 1500. The output is program number 3.

select \* from program

where price>1500;

**Output**

****

2) I have executed to join function on the table from the database.

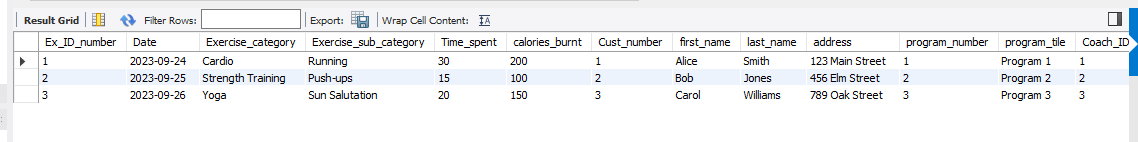
a) joining Exercise\_Log and Customer table based on the ex\_ID\_number of the exercise\_log table and Ex\_id\_number of the customer table. As output, there is the date, cust\_number. First\_name, program\_number in the joined table.

select \* from exercise\_log as e

join customer as c

on e.ex\_ID\_number=c.ex\_ID\_number;

**OUTPUT**



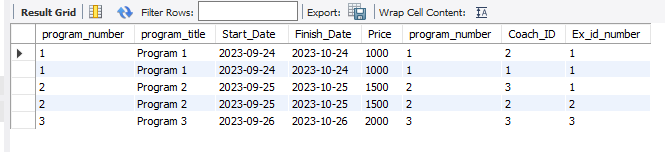
b) joining the program and program\_coach table based on the program number of the program table and the program number of the customer table. As output, the column in the newly joined table is- program\_number, program\_title, Start\_date, finish\_date, Price, program\_number, Coach\_ID, and Ex\_id\_number.

select \* from program as p

left join program\_coach as pc

on p.program\_number=pc.program\_number;

**OUTPUT**



We can also explore more queries like having, init, left join, and aggregate functions in this schema. That’s all for this Mind Full Body Fitness Database schema.