FitLife Inc. Database

Richard Anderson

GLOSSARY

FitLife Inc - *Executive Summary*

FitLife Inc., one of its many gym-based business entities located in Des Moines, IA, wants to build a database system which would track the information of all its day-to-day operations. The business provides the users with services such as general fitness, specific fitness, trainer program, fitness consultation, body care products, specific lifestyle retail, yoga and swimming. The entire menu of service fees inclusive of monthly and annual memberships will be available in Appendix A.

As of today, FitLife consists of a workforce of twelve, which includes one head manager, one assistant manager, two receptionists/lifestyle specialists, five certified trainers, two yoga instructors, and two swimming instructors. Each member of the staff has personal details, ID numbers, contact information, and role specifications.

Users must get an appointment with the lifestyle specialist before enrolling into a fitness program through the receptionist. After the appointment, the user must meet with any trainer to discuss the specifics of the program they would like to enroll into taking into consideration the advice they received from their appointment. The appointment must include username, D.O.B., contact information, date and time of appointment, specific staff member requested (if any), and possible special requests. For example, Henry Peter makes an appointment for fitness consultation with Matt at 10:00 am on August 12th, additional requests being a visit to the lifestyle retail store to buy muscle protein powder.

The products offered through the retail store would include gym equipment, membership programs, and dietary supplements. The complete list of the specifics will be available in Appendix B.

When a user arrives at the appointment with the lifestyle specialist their information will be logged into the database. This database will be able to tell the lifestyle specialist, what classes the user has taken, what products this person has bought from the store, how long they have been a member, and the last time the person attended the gym. This will allow the lifestyle specialist to easily recommend classes within the user's capability which will enhance the users experience and help ensure they stay a member and work towards their fitness goals. Finally, all costs are compounded or added towards the end, and members are billed their respective amount.

For example, Henry decides to make an appointment with a lifestyle specialist with Alice at 2:00 pm. Once he has arrived his name will be logged into the system, so the lifestyle specialist knows Henry's information. With this Alice will be able to determine how committed Henry is to his overall health and can advise him on what classes he might like as well as the correct difficulty of the class he will be taking. This will ensure that Henry is not overwhelmed in his new class and continues to take it.

Objective and Scope

Objective:-

- The primary purpose of this database is to ensure that data is stored correctly and maintained properly.
- It can be altered, updated, deleted, and new information can be added accordingly following the initial design rules of the database, that can be found in the ERD and data dependency diagram further down the document.

Scope:-

- At the moment, this database is exclusive for branches in the state of Iowa. As soon as FitLife Inc expands, new branches will be set up and the database will be extended accordingly.
- The database can be used to manage knowledge and information resources. For FitLife gym corporation, it can be used to study the types of customers that use their resources, possible retail uses, pharmaceutical uses(only after expansion), and possible fitness certification agency.

Business Rules

1.**Branch** is identified by Branch_Id and contains the attributes Branch_Name, Branch_Contact, and Branch_Address(Street, City, Zip Code). The Branch is only one institution in the area and will have multiple customers.

2.**New User** is identified by SSN and contains the attributes User_Name, User_Age, Contact, Email and User_Address(Street, City, Zip Code). Each user can contact only one branch that is within their vicinity.

3.**Reception** is identified by Appointment_Number and contains the attributes Customer_DOB, Customer_Name, Customer_Appointment_date, and Customer_Appointment_Time. Each reception clerk or a receptionist will take details of only one customer at a time.

4.**Employee** is identified by Employee_Id and contains the attributes Employee_Name and Employee_description. Each consultant can take up at least one new member but each member can have only one consultant.

5.**Membership** is identified by Member_Id and contains the attributes Membership_Type and Membership_Cost. Many new members must enrol into any of the programs offered by the gym and can also enroll in more than one program that can be shown through overlapping constraints.

6.**General Fitness** is identified by Fitness_Id and contains the attributes Fitness_Name and Fitness_Cost. At Least one member can enroll into the program and the program itself can have any number of members.

7.**Specific Fitness** is identified by Specific_Id and contains the attributes Specific_Name and Specific_Cost. At Least one member can enroll into the program and the program itself can have any number of members.

8.**Other_Program** is identified by Other_Id and contains the attributes Other_Name and Other_Cost. At Least one member can enroll into the program and the program itself can have any number of members.

9.**Store** is identified by Product_Id and contains Product_Category, Product_Description, and Price. Each order can have multiple products and each member can use the store multiple times.

10.**Store Orders** is identified by Order_Id and contains the attribute Order_Cost. Each member can place multiple store orders and each order will be billed separately.

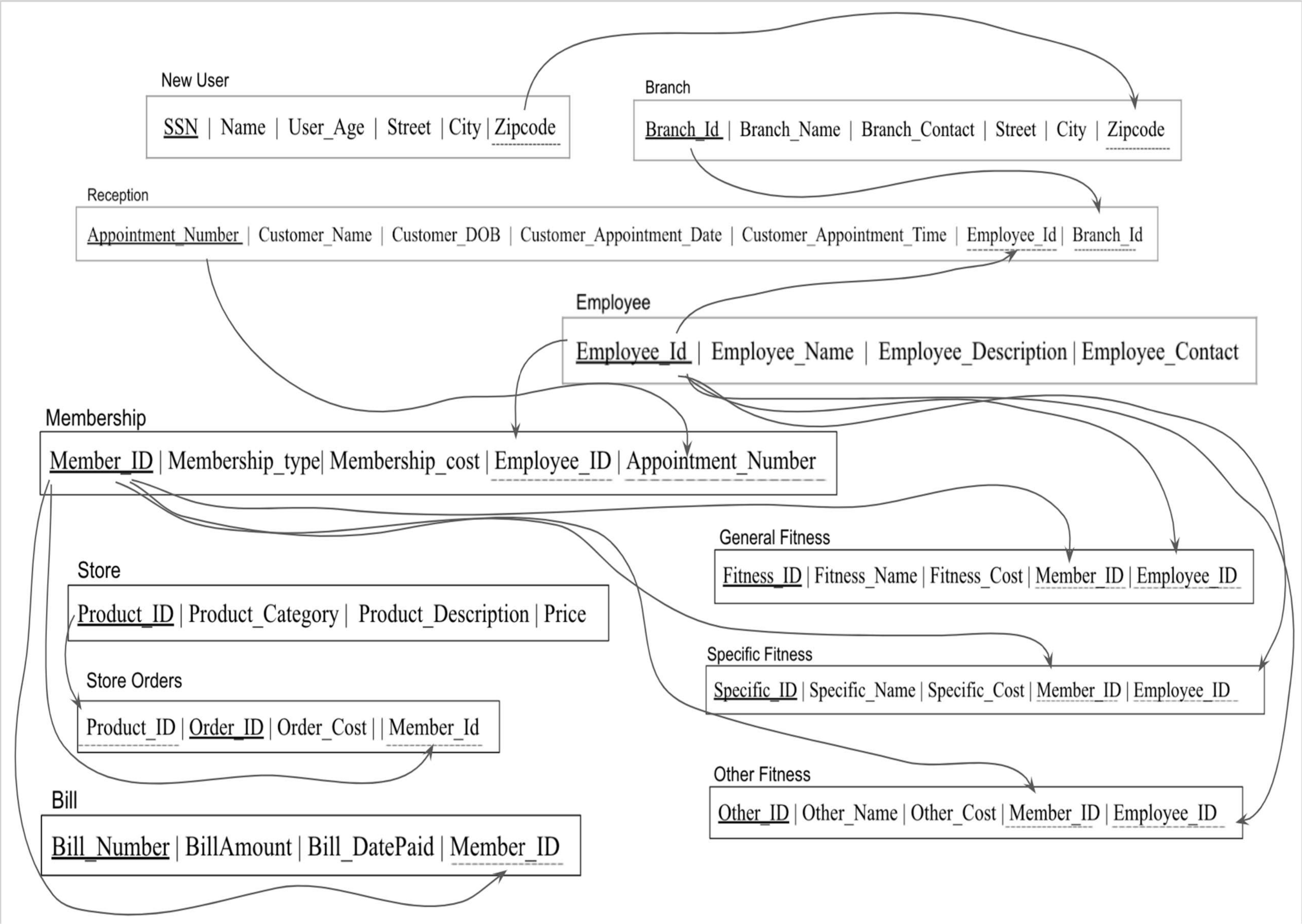
11.**Bill** is identified by Bill_Number and contains the attributes Bill_DatePaid, and BillAmount. Each bill has multiple can have at least one order and also have the complete cost of programs and fees of services for at least one member who has been offered.

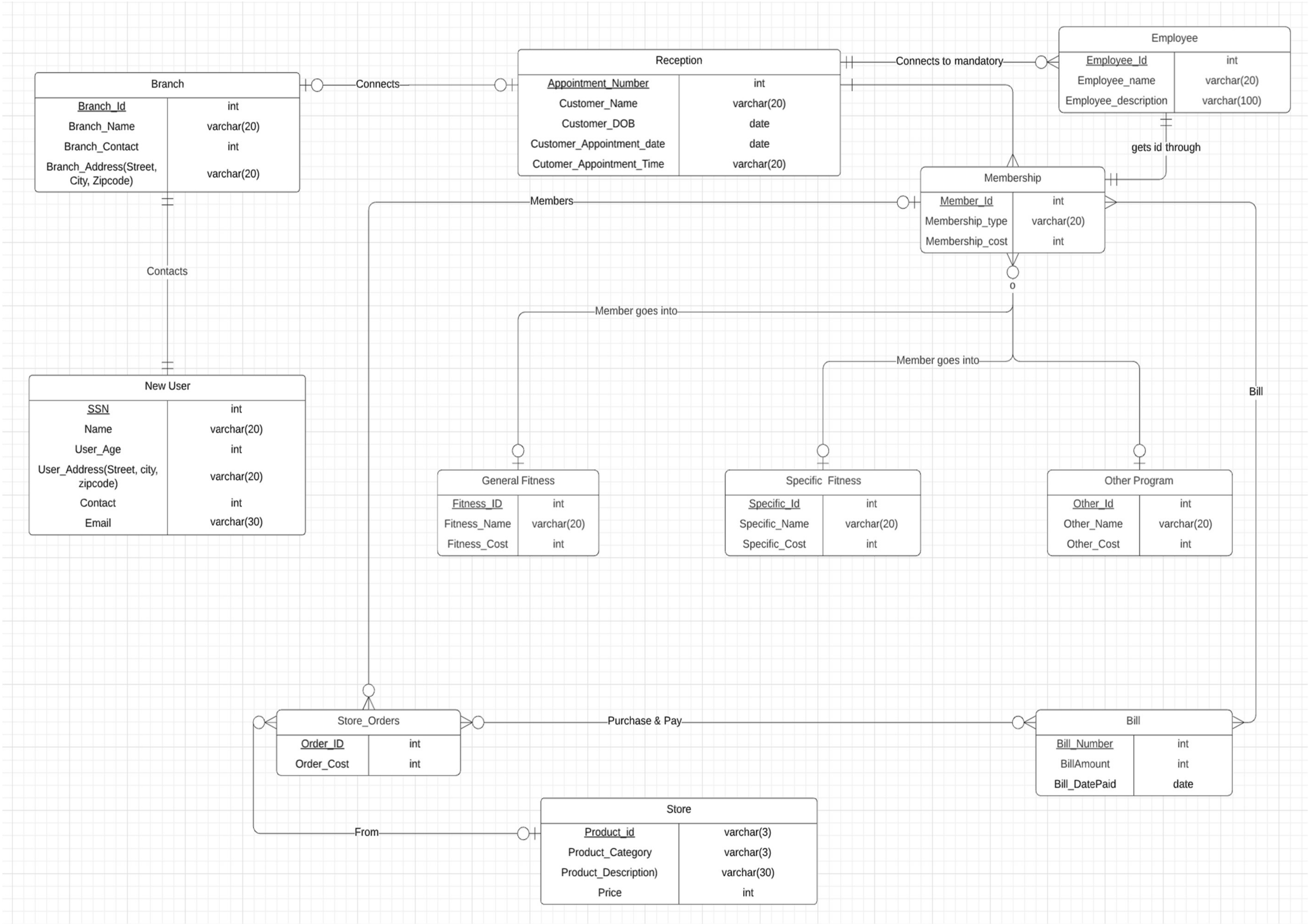
Appendix A: Gym Services Available at FitLife Inc.

Service Category	Service	Description	Price
General Fitness	Monthly Membership Fee	Gym and equipment use	$50.00
	Annual Membership Fee	Gym and equipment use	$500.00
Specific Fitness	Endurance	Shaping physic and improving stamina	$15.00
	Strength	Resistance training of muscles	$15.00
Fitness Consultation	Nutritionist	Will come up with a diet to help User gain or lose weight	$0.00
	General trainer	Will help come up with an ideal workout for user to do weekly on your own	$50.00
Others	Yoga Monthly	-	$25.00
	Swimming Monthly	-	$100.00

Appendix B: Products Available for Purchase at FitLife Store

Product Category	Description	Size	Price
Beverages	Bang Energy Drink	16.9 fl.oz	$5.00
	Red Bull Energy Drink	12.9 fl. oz	$6.00
	Reign Energy Drink	16.9 fl.oz	$5.00
	Muscle Milk	12.9 fl. oz	$7.00
	Moo Juice	8.9 fl. oz	$7.00
Merchandise	T-Shirt	Xs, S, M, L, XL, XXL	$25.00
	Sweatshirt	Xs, S, M, L, XL, XXL	$50.00
	Sweat Pants	Xs, S, M, L, XL, XXL	$50.00
	Yoga Pants	Xs, S, M, L, XL, XXL	$80.00
	Yoga Mat	-	$30.00
	Water bottle	24.00 fl. oz	$20.00
	Blender Bottle	24.00 fl. oz	$20.00


New User
SSN | Name | User_Age | Street | City | Zipcode
Branch
Branch_Id | Branch_Name | Branch_Contact | Street | City | Zipcode
Reception
Appointment_Number | Customer_Name | Customer_DOB | Customer_Appointment_Date | Customer_Appointment_Time | Employee_Id | Branch_Id
Employee
Employee_Id | Employee_Name | Employee_Description | Employee_Contact
Membership
Member_ID | Membership_type| Membership_cost | Employee_ID | Appointment_Number
General Fitness
Fitness_ID | Fitness_Name | Fitness_Cost | Member_ID | Employee_ID
Store
Product_ID | Product_Category | Product_Description | Price
Specific Fitness
Specific_ID | Specific_Name | Specific_Cost | Member_ID | Employee_ID
Store Orders
Product_ID | Order_ID | Order_Cost | | Member_Id
Other Fitness
Other_ID | Other_Name | Other_Cost | Member_ID | Employee_ID
Bill
Bill_Number | BillAmount | Bill_DatePaid | Member_ID


Branch
Branch_Id
int
Branch_Name
varchar(20)
Branch_Contact
int
Branch_Address(Street, City, Zipcode)
varchar(20)
Connects
Reception
Appointment_Number
int
Customer_Name
varchar(20)
Customer_DOB
date
Customer_Appointment_date
date
Cutomer_Appointment_Time
varchar(20)
Connects to mandatory
Employee
Employee_Id
int
Employee_name
varchar(20)
Employee_description
varchar(100)
gets id through
Membership
Member_Id
int
Membership_type
varchar(20)
Membership_cost
int
Members
Contacts
o
Member goes into
Member goes into
Bill
New User
SSN
int
Name
varchar(20)
User_Age
int
User_Address(Street, city, zipcode)
varchar(20)
Contact
int
Email
varchar(30)
General Fitness
Fitness_ID
int
Fitness_Name
varchar(20)
Fitness_Cost
int
Specific Fitness
Specific_Id
int
Specific_Name
varchar(20)
Specific_Cost
int
Other Program
Other_Id
int
Other_Name
varchar(20)
Other_Cost
int
Store_Orders
Order_ID
int
Order_Cost
int
Purchase & Pay
Bill
Bill_Number
int
BillAmount
int
Bill_DatePaid
date
From
Store
Product_id
varchar(3)
Product_Category
varchar(3)
Product_Description)
varchar(30)
Price
int

Data Dictionary

1. **New User**
    - SSN int NOT NULL,
    - Name varchar(30) NOT NULL,
    - User_Age int NOT NULL,
    - Street varchar(30) NOT NULL,
    - City varchar(30) NOT NULL,
    - Zipcode int NOT NULL,
    - Contact int NOT NULL,
    - Email varchar(30) NOT NULL,
    - PRIMARY KEY (SSN)
    - FOREIGN KEY(Zipcode)

2. **Branch**
    - Branch_ID int NOT NULL,
    - Branch_Name varchar(30) NOT NULL,
    - Branch_Contact int NOT NULL,
    - Street varchar(30) NOT NULL,
    - City varchar(30) NOT NULL,
    - Zipcode int NOT NULL,
    - PRIMARY KEY (Branch_ID)
    - FOREIGN KEY(Zipcode)

3. **Reception**
    - Appointment_Number int NOT NULL,
    - Customer_Name varchar(30) NOT NULL,
    - Customer_DOB date NOT NULL,
    - Customer_Appointment_Date date NOT NULL,
    - Customer_Appointment_Time varchar(30) NOT NULL,
    - PRIMARY KEY (Appointment_Number)

4. **Employee**
    - Employee_ID int NOT NULL,
    - Employee_Name varchar(30) NOT NULL,
    - Employee_description varchar(30) NOT NULL,
    - PRIMARY KEY(Employee_ID)

5. **Membership**
    - Member_Id int NOT NULL,

- Employee_ID int NOT NULL,
- Appointment_Number int NOT NULL,
- Membership_type varchar(30) NOT NULL,
- Membership_cost int NOT NULL,
- PRIMARY KEY (Member_Id)
- FOREIGN KEY(Employe_Id)

6. **General Fitness**
    - Fitness_ID int NOT NULL
    - Fitness_Name varchar(30) NOT NULL
    - Fitness_Cost int NOT NULL
    - PRIMARY KEY(Fitness_ID)
    - FOREIGN KEY(Employe_Id)

7. **Specific Fitness**
    - Specific_ID int NOT NULL
    - Specific_Name varchar(30) NOT NULL
    - Specific_Cost int NOT NULL
    - PRIMARY KEY(Specific_ID)
    - FOREIGN KEY(Employe_Id)

8. **Other Program**
    - Other_ID int NOT NULL
    - Other_Name varchar(30) NOT NULL
    - Other_Cost int NOT NULL
    - PRIMARY KEY(Other_ID)
    - FOREIGN KEY(Employe_Id)

9. **Store**
    - Product_Id varchar(3) Not Null,
    - Product_Category varchar(3) Not Null,
    - Product_Description varchar(30) Not Null,
    - Price int Not Null,
    - PRIMARY KEY (Product_ID)

10. **Store Orders**
    - Order_ID int NOT NULL,
    - Order_Cost int NOT NULL,
    - PRIMARY KEY (Order_ID)
    - FOREIGN KEY(Product_Id)

11. Bill

- Bill_Number int NOT NULL
- BillAmount decimal(7,2) NOT NULL
- Bill_DatePaid date NOT NULL
- PRIMARY KEY(Bill_Number)
- FOREIGN KEY(Member_Id)

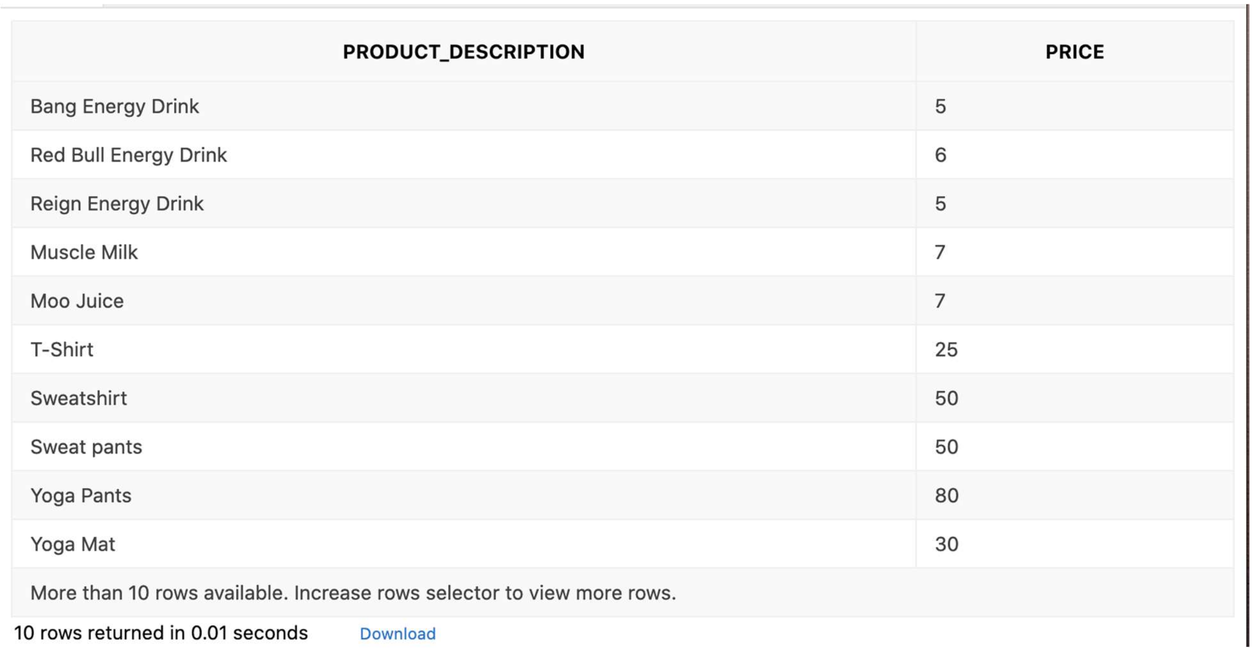
Data Definition Language

```
Set Escape On

DROP TABLE New_User;
Create Table New_User(
SSN int NOT NULL,
Name varchar(30) NOT NULL,
User_Age int NOT NULL,
Street varchar(30) NOT NULL,
City varchar(30) NOT NULL,
Zipcode int NOT NULL,
Contact int NOT NULL,
Email varchar(30) NOT NULL,
PRIMARY KEY (SSN)
);
INSERT INTO New_User VALUES(141124312, 'Bill Gate', 38, '145 Main Street ','Ames',
50014, 5155987640,'billgates@micrsoft.com');
INSERT INTO New_User VALUES(146324395, 'Bob Lee', 58, '422 Oak Lane ','Ames', 50014,
5155155151,'fighterbob@gmail.com');

INSERT INTO New_User VALUES(144194312, 'John Ryan ', 23, '231 Big Street ','Des
Moines', 50013, 8172653450,'Jryan@yahoo.co.au');
INSERT INTO New_User VALUES(136324395, 'Kate Lee', 57, '422 Oak Lane ','Des Moines',
50013, 9876541234,'dojolee@yahoo.co.in');

INSERT INTO New_User VALUES(141126543, 'Joe Man', 81, '313 Tree Road ','Cedar Rapids',
50010, 2349876547,'manJoe@gmail.com');
INSERT INTO New_User VALUES(123456789, 'Joe Lee', 58, '421 Ok Lane ','Cedar Rapids',
50010, 7658900142,'leejoe@gmail.com');


DROP TABLE Branch;
CREATE TABLE Branch(
Branch_ID int NOT NULL,
Branch_Name varchar(30) NOT NULL,
Branch_Contact int NOT NULL,
Street varchar(30) NOT NULL,
City varchar(30) NOT NULL,
Zipcode int NOT NULL,
PRIMARY KEY (Branch_ID)
);
INSERT INTO Branch VALUES(3411, 'Ames Branch', 5154421972, '147 Oak Lane ','Ames', 50014);
INSERT INTO Branch VALUES(3412, 'Des Monies Branch', 5155291971, '212 Big Lane ','Des Moines', 50013);
INSERT INTO Branch VALUES(3413, 'Cedar Rapids Branch', 5157121951, '861 Small Lane ','Cedar Rapids', 50010);


DROP TABLE Reception;
CREATE TABLE Reception(
Appointment_Number int NOT NULL,
Branch_ID int NOT NULL,
Customer_Name varchar(30) NOT NULL,
Customer_DOB date NOT NULL,
Customer_Appointment_Date date NOT NULL,
Customer_Appointment_Time varchar(30) NOT NULL,
PRIMARY KEY (Appointment_Number)
);
INSERT INTO Reception VALUES(001,3411 ,'Bill Gate', '17-Nov-1972', '04-Feb-2021 ','12:30PM');
INSERT INTO Reception VALUES(002,3411,'Bob Lee', '14-Dec-1982', '05-Feb-2021 ','12:40PM');
INSERT INTO Reception VALUES(003,3412,'John Ryan', '14-Nov-1999', '03-Feb-2021 ','11:30PM');
INSERT INTO Reception VALUES(004,3412,'Kate Lee', '24-Aug-1989', '02-Feb-2021 ','12:45PM');
INSERT INTO Reception VALUES(005,3413,'Joe Man', '18-Oct-1989', '11-Feb-2021 ','12:45PM');
INSERT INTO Reception VALUES(006,3413,'Joe Lee', '16-Jan-1989', '21-Feb-2021 ','12:45PM');

DROP TABLE Employee;
CREATE TABLE Employee(
Employee_ID int NOT NULL,
Employee_Name varchar(30) NOT NULL,
Employee_description varchar(60) NOT NULL,
Employee_contact varchar(30) NOT NULL,
PRIMARY KEY (Employee_ID)
```

```
);
INSERT INTO Employee VALUES(191442, 'Dwayne Johnson', 'Personal
Trainer','djohn@fitlife.com');
INSERT INTO Employee VALUES(111432, 'Wendy Smith', 'Personal
Trainer','wsmith@fitlife.com');
INSERT INTO Employee VALUES(181441, 'Carl Bob', 'Receptionist/Lifestyle
Specialist','cbob@fitlife.com');
INSERT INTO Employee VALUES(171442, 'Tony Tiger', 'Yoga Instructor','ttiger@fitlife.com');
Insert INTO Employee VALUES(123456, 'Issac Newton', 'Swimming
Instructor','inewton@fitlife.com');
Insert INTO Employee VALUES(123457, 'Bob Marley', 'Personal
Trainer','bmarley@fitlife.com');
Insert INTO Employee VALUES(123458, 'Nicola Tesla', 'Swimming
Instructor','ntesla@fitlife.com');
Insert INTO Employee VALUES(985724, 'Trevor Scott', 'Personal Trainer','scotty@fitlife.com');
Insert INTO Employee VALUES(123443, 'Ali Baba', 'Manager','alibaba@fitlife.com');
Insert INTO Employee VALUES(435794, 'Jaden Smith', 'Assistant
Manager','jsmith@fitlife.com');
Insert INTO Employee VALUES(342343, 'Trevor Noah', 'Yoga Instructor','tnoah@fitlife.com');
Insert INTO Employee VALUES(757656, 'Jimmy Fallon', 'Personal
Trainer','jfallon@fitlife.com');
Insert INTO Employee VALUES(634566, 'Stephen Colbert', 'Receptionist/Lifestyle
Specialist','scolbert@fitlife.com');



DROP TABLE Membership;
CREATE TABLE Membership(
Member_Id int NOT NULL,
Employee_ID int NOT NULL,
Appointment_Number int NOT NULL,
Membership_type varchar(30) NOT NULL,
Membership_cost int NOT NULL,
PRIMARY KEY (Member_Id)
);
INSERT INTO Membership VALUES(921471, 181441,001,'Monthly', 50);
INSERT INTO Membership VALUES(751421, 181441,002,'Annual', 500);
INSERT INTO Membership VALUES(123455, 181441,003,'Annual', 500);
INSERT INTO Membership VALUES(264552, 634566,004,'Monthly', 50);
INSERT INTO Membership VALUES(345678, 634566,005,'Annual', 500);
INSERT INTO Membership VALUES(245236, 634566,006,'Monthly', 50);


DROP TABLE General_Fitness;
CREATE TABLE General_Fitness(
Fitness_ID int NOT NULL,
Member_Id int NOT NULL,
Fitness_Name varchar(30) NOT NULL,
Fitness_Cost int NOT NULL,
Employee_ID int NOT NULL,
PRIMARY KEY (Fitness_ID)
);
INSERT INTO General_Fitness VALUES(123, 921471,'General Fitness', 50,171442);
INSERT INTO General_Fitness VALUES(124, 751421,'General Fitness', 50,111432);


DROP TABLE Specific_Fitness;
CREATE TABLE Specific_Fitness(
Specific_ID int NOT NULL,
Member_Id int NOT NULL,
Specific_Name varchar(30) NOT NULL,
Specific_Cost int NOT NULL,
Employee_ID int NOT NULL,
PRIMARY KEY (Specific_ID)
);
INSERT INTO Specific_Fitness VALUES(234, 123455, 'Specific Fitness-Endurance',
50,985724);
INSERT INTO Specific_Fitness VALUES(235, 264552, 'Specific Fitness-Strength', 50,191442);


DROP TABLE Other_Program;
CREATE TABLE Other_Program(
Other_ID int NOT NULL,
Member_Id int NOT NULL,
Other_Name varchar(30) NOT NULL,
Other_Cost int NOT NULL,
Employee_ID int NOT NULL,
PRIMARY KEY (Other_ID)
);
INSERT INTO Other_Program VALUES(300, 345678,'Yoga', 25,171442);
INSERT INTO Other_Program VALUES(301, 245236,'Swimming', 100,123456);


Drop Table Store;
CREATE TABLE Store(
Product_Id varchar(3) Not Null,
Product_Category varchar(3) Not Null,
Product_Description varchar(30) Not Null,
Price int Not Null,
PRIMARY KEY (Product_ID)
);


Insert into Store VALUES('ED1','BEV','Bang Energy Drink',5);
Insert into Store VALUES('ED2','BEV','Red Bull Energy Drink',6);
Insert into Store VALUES('ED3','BEV','Reign Energy Drink',5);
Insert into Store VALUES('ED4','BEV','Muscle Milk',7);
Insert into Store VALUES('ED5','BEV','Moo Juice',7);
Insert into Store VALUES('ME1','MER','T-Shirt',25);
Insert into Store VALUES('ME2','MER','Sweatshirt',50);
Insert into Store VALUES('ME3','MER','Sweat pants',50);
Insert into Store VALUES('ME4','MER','Yoga Pants',80);
Insert into Store VALUES('ME5','MER','Yoga Mat',30);
Insert into Store VALUES('ME6','MER','Water Bottle',20);
Insert into Store VALUES('ME7','MER','Blender Bottle',20);
Insert into Store VALUES('ME8','MER','Swimming Clothes',60);
Insert into Store VALUES('SU1','SUP','Supplement Muscle Powder',40);
Insert into Store VALUES('SU2','SUP','Supplement Vitamin Powder',40);
Insert into Store VALUES('SU3','SUP','Supplement Multivitamin Tablet',40);


DROP TABLE Store_Orders;
CREATE TABLE Store_Orders(
Order_ID int NOT NULL,
Member_Id int NOT NULL,
Product_Id varchar(3) NOT NULL,
Order_Cost int NOT NULL,
PRIMARY KEY (Order_ID)
);
Insert Into Store_Orders VALUES(001,921471,'ED1',5);
Insert Into Store_Orders VALUES(002,921471,'ED5',7);
Insert Into Store_Orders VALUES(003,921471,'ME2',50);
Insert Into Store_Orders VALUES(004,751421,'ED2',6);
Insert Into Store_Orders VALUES(005,751421,'ME1',25);
Insert Into Store_Orders VALUES(006,751421,'ED2',6);
Insert Into Store_Orders VALUES(007,751421,'SU2',40);
Insert Into Store_Orders VALUES(008,264552,'SU2',40);
Insert Into Store_Orders VALUES(009,264552,'SU3',40);
Insert Into Store_Orders VALUES(010,245236,'ME5',30);


DROP TABLE Bill;
CREATE TABLE Bill(
Bill_Number int NOT NULL,
Member_Id int NOT NULL,
BillAmount decimal(7,2) NOT NULL,
Bill_DatePaid date NOT NULL,
PRIMARY KEY (Bill_Number)
);
Insert Into Bill VALUES(001,921471,162.00,'04-Feb-2021');
Insert Into Bill VALUES(002,751421,587.00,'10-Jan-2022');
Insert Into Bill VALUES(003,264552,660.00,'07-Feb-2022');
Insert Into Bill VALUES(004,264552,100.00,'04-Mar-2021');
Insert Into Bill VALUES(005,345678,525.00,'10-Jan-2022');
Insert Into Bill VALUES(006,245236,150.00,'07-Feb-2021');

COMMIT;
```

Data Manipulation Language and Reports

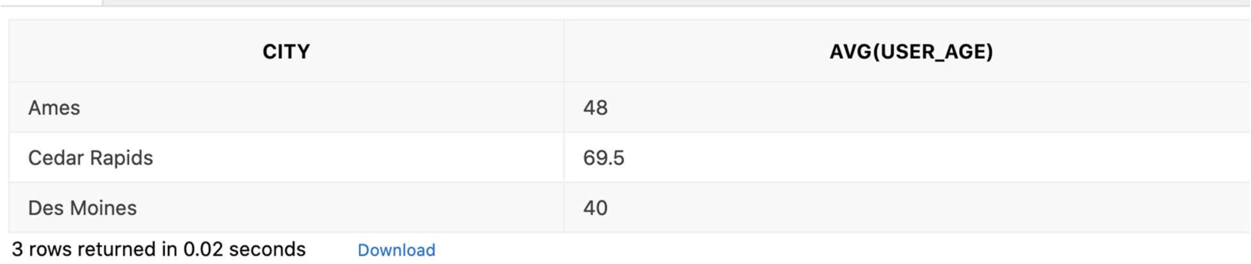
1. select product_description, price
   from store;
   (**Displays the product name and its price to the customer**)



PRODUCT_DESCRIPTION	PRICE
Bang Energy Drink	5
Red Bull Energy Drink	6
Reign Energy Drink	5
Muscle Milk	7
Moo Juice	7
T-Shirt	25
Sweatshirt	50
Sweat pants	50
Yoga Pants	80
Yoga Mat	30
More than 10 rows available. Increase rows selector to view more rows.	

10 rows returned in 0.01 seconds

2. select new_user.city, avg(user_age)
   from new_user, branch
   where new_user.city = branch.city
   group by new_user.city
   order by city;
   (**Displays the average age of all users within every individual branch's city.**)



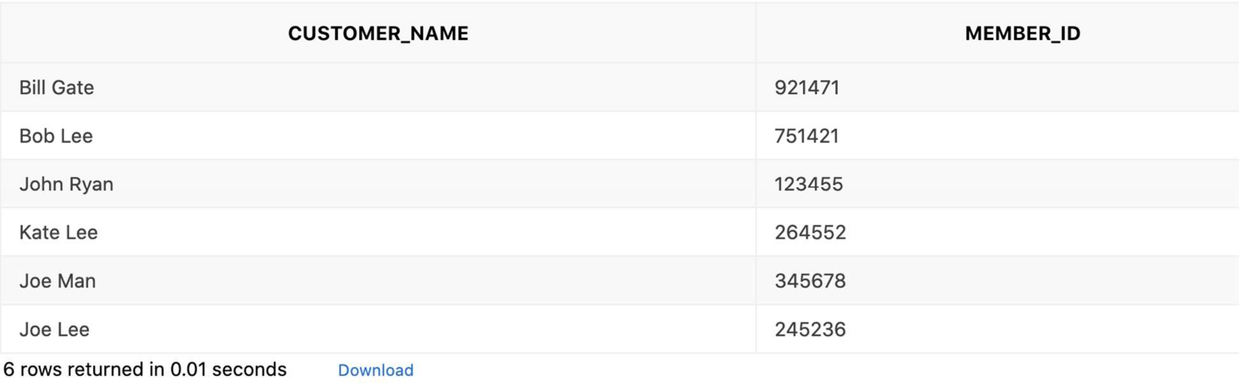
CITY	AVG(USER_AGE)
Ames	48
Cedar Rapids	69.5
Des Moines	40

3 rows returned in 0.02 seconds

3. select reception.customer_name, membership.member_id
   from membership, employee, reception

```
where membership.employee_id = employee.employee_id and employee_description
like '%Lifestyle%' and
membership.appointment_number=reception.appointment_number;
```

(**Displays the name and id of every member that has an assigned Lifestyle specialist**)



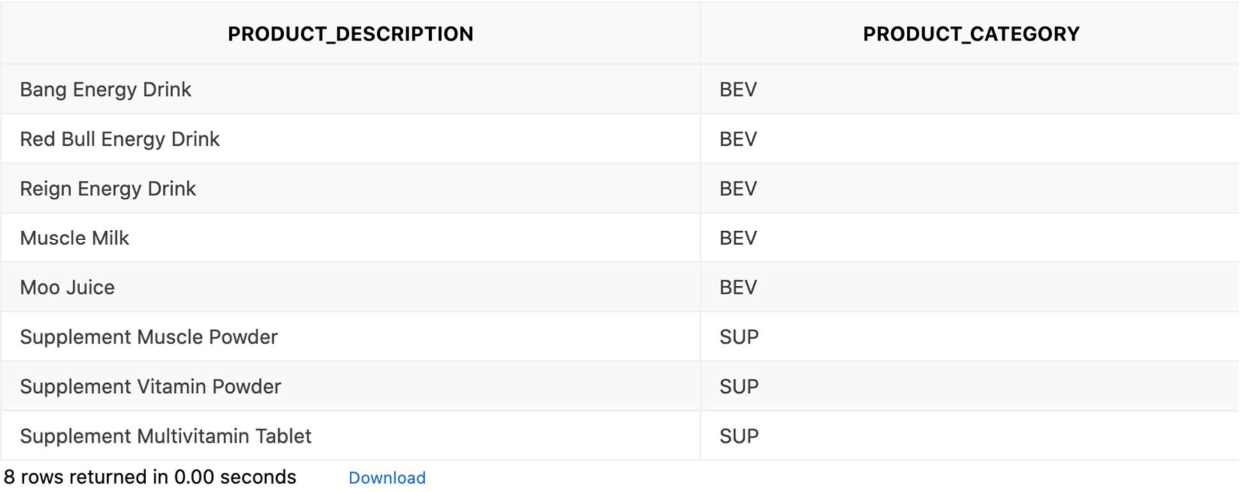
CUSTOMER_NAME	MEMBER_ID
Bill Gate	921471
Bob Lee	751421
John Ryan	123455
Kate Lee	264552
Joe Man	345678
Joe Lee	245236

6 rows returned in 0.01 seconds Download

4.

```
select product_description, product_category
from store
where product_category = 'BEV' or product_category = 'SUP';
```

(**Displays the consumable products that the store has to offer along with their respective category tag**)



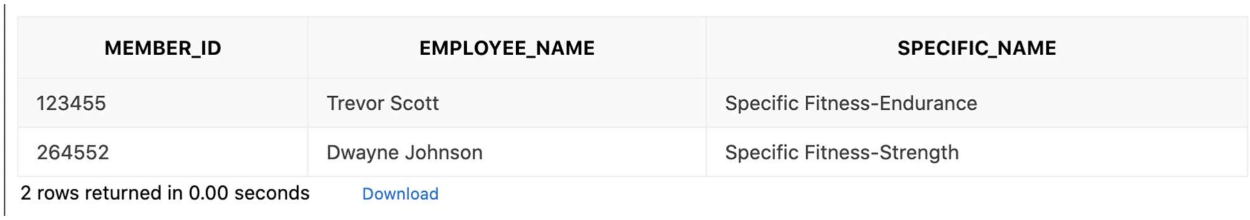
PRODUCT_DESCRIPTION	PRODUCT_CATEGORY
Bang Energy Drink	BEV
Red Bull Energy Drink	BEV
Reign Energy Drink	BEV
Muscle Milk	BEV
Moo Juice	BEV
Supplement Muscle Powder	SUP
Supplement Vitamin Powder	SUP
Supplement Multivitamin Tablet	SUP

8 rows returned in 0.00 seconds Download

5.

```
select membership.member_id, employee_name, specific_name
from employee, specific_fitness, membership
where employee.employee_id = specific_fitness.employee_id
and membership.member_id = specific_fitness.member_id;
```

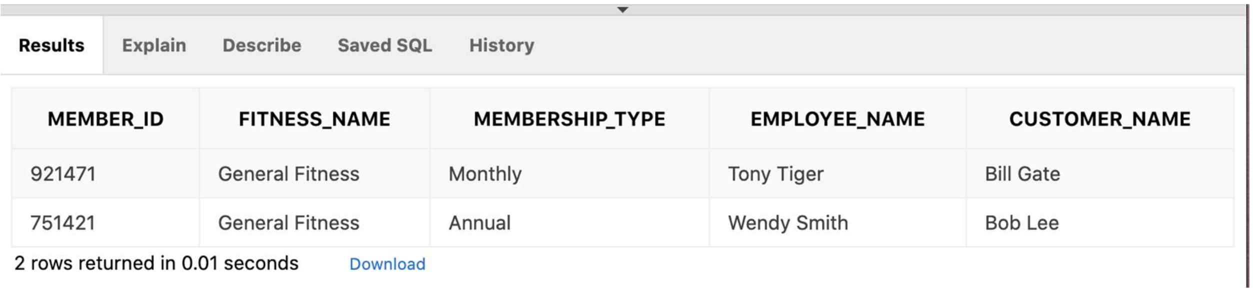
(**Displays the member ID of the assigned specific fitness employee, the employee name, and the name of their respective specific fitness program**)



MEMBER_ID	EMPLOYEE_NAME	SPECIFIC_NAME
123455	Trevor Scott	Specific Fitness-Endurance
264552	Dwayne Johnson	Specific Fitness-Strength

2 rows returned in 0.00 seconds

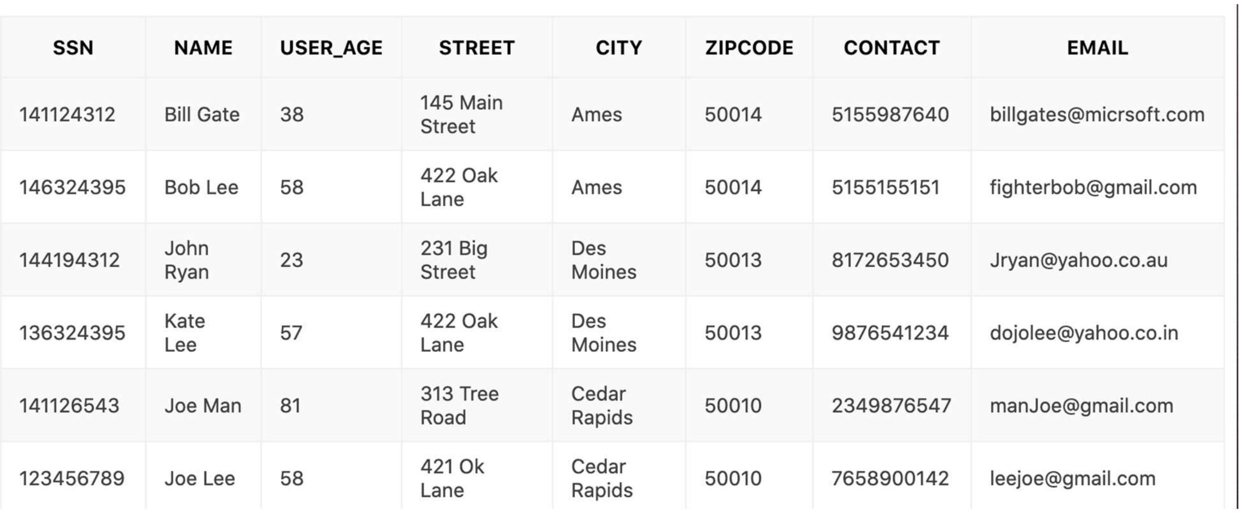
6. select general_fitness.member_id, general_fitness.fitness_name,membership.membership_type, employee_name, reception.customer_name from general_fitness, membership, employee, reception where general_fitness.member_id=membership.member_id and employee.employee_id=general_fitness.employee_id and membership.appointment_number=reception.appointment_number;
   (**Displays the names of members in the general fitness program along with names of assigned employee, member id and customer membership type**)



MEMBER_ID	FITNESS_NAME	MEMBERSHIP_TYPE	EMPLOYEE_NAME	CUSTOMER_NAME
921471	General Fitness	Monthly	Tony Tiger	Bill Gate
751421	General Fitness	Annual	Wendy Smith	Bob Lee

2 rows returned in 0.01 seconds

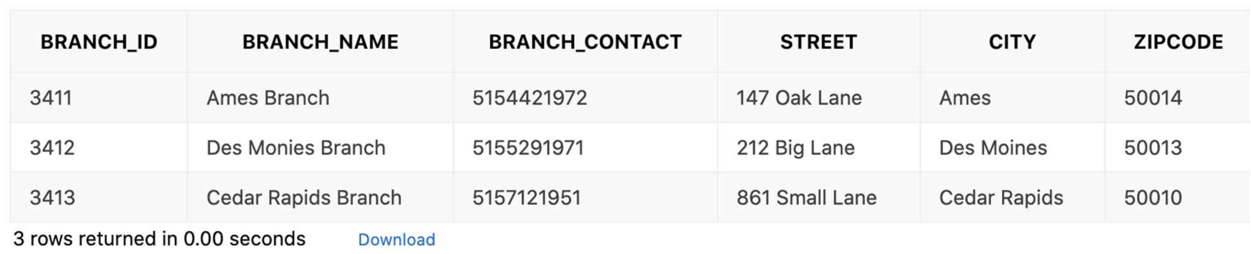
7. select *
   from new_user;
   (**Displays a report all new user information that have planned to enroll into the gym**)



SSN	NAME	USER_AGE	STREET	CITY	ZIPCODE	CONTACT	EMAIL
141124312	Bill Gate	38	145 Main Street	Ames	50014	5155987640	billgates@micrsoft.com
146324395	Bob Lee	58	422 Oak Lane	Ames	50014	5155155151	fighterbob@gmail.com
144194312	John Ryan	23	231 Big Street	Des Moines	50013	8172653450	Jryan@yahoo.co.au
136324395	Kate Lee	57	422 Oak Lane	Des Moines	50013	9876541234	dojolee@yahoo.co.in
141126543	Joe Man	81	313 Tree Road	Cedar Rapids	50010	2349876547	manJoe@gmail.com
123456789	Joe Lee	58	421 Ok Lane	Cedar Rapids	50010	7658900142	leejoe@gmail.com

8. select *
   from branch;

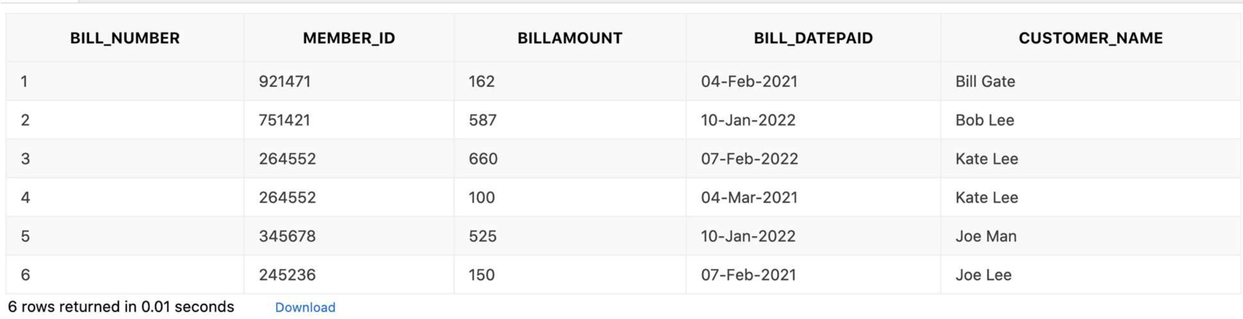
(Displays information about all the branches that exist in the state that users will use)



BRANCH_ID	BRANCH_NAME	BRANCH_CONTACT	STREET	CITY	ZIPCODE
3411	Ames Branch	5154421972	147 Oak Lane	Ames	50014
3412	Des Monies Branch	5155291971	212 Big Lane	Des Moines	50013
3413	Cedar Rapids Branch	5157121951	861 Small Lane	Cedar Rapids	50010

3 rows returned in 0.00 seconds

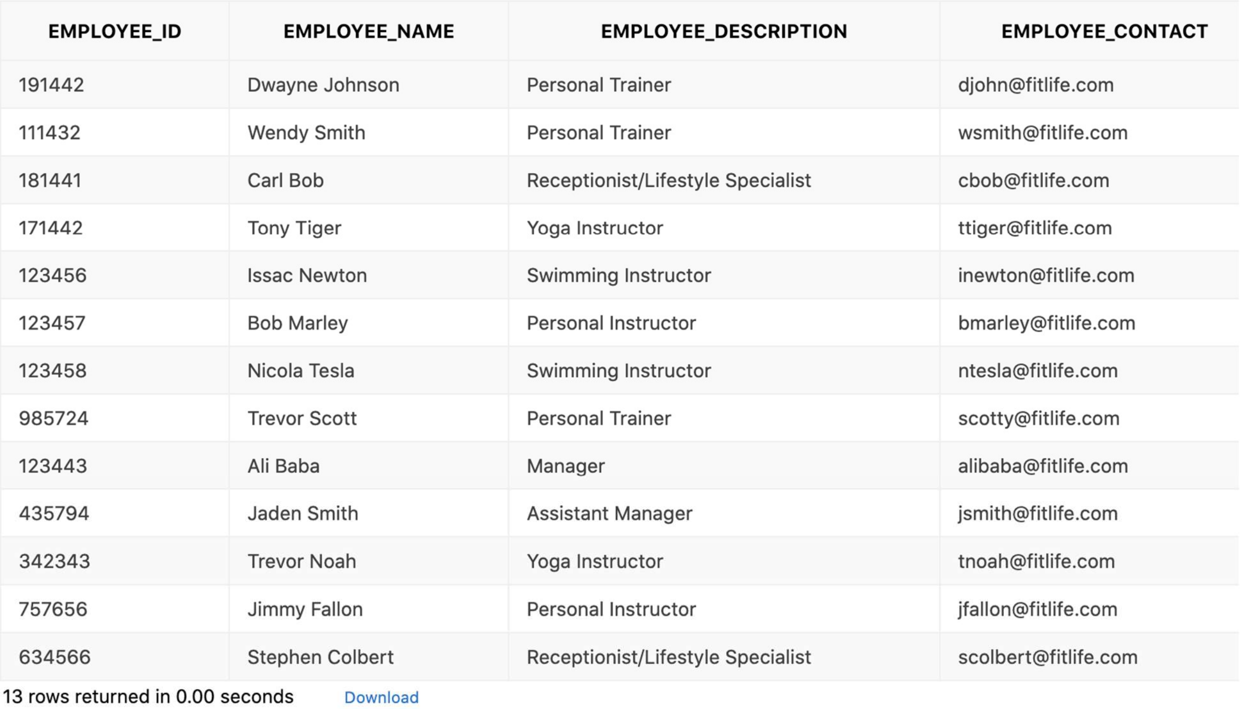
9. select bill.bill_number, bill.member_id, bill.billamount, bill.bill_datepaid, reception.customer_name from bill, membership, reception
   where bill.member_id=membership.member_id and
   membership.appointment_number=reception.appointment_number;
   (Displays information about the bills paid by the users.)



BILL_NUMBER	MEMBER_ID	BILLAMOUNT	BILL_DATEPAID	CUSTOMER_NAME
1	921471	162	04-Feb-2021	Bill Gate
2	751421	587	10-Jan-2022	Bob Lee
3	264552	660	07-Feb-2022	Kate Lee
4	264552	100	04-Mar-2021	Kate Lee
5	345678	525	10-Jan-2022	Joe Man
6	245236	150	07-Feb-2021	Joe Lee

6 rows returned in 0.01 seconds

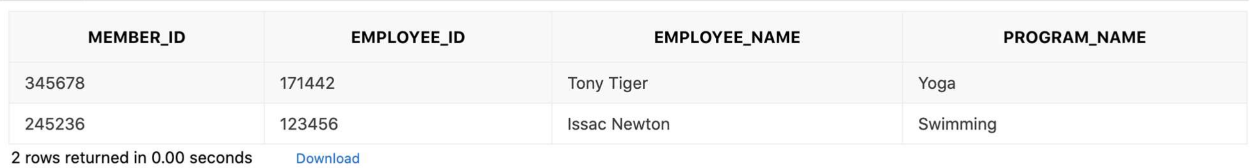
10. select * from employee;
    (Displays a report about the employees at fitlife)



EMPLOYEE_ID	EMPLOYEE_NAME	EMPLOYEE_DESCRIPTION	EMPLOYEE_CONTACT
191442	Dwayne Johnson	Personal Trainer	djohn@fitlife.com
111432	Wendy Smith	Personal Trainer	wsmith@fitlife.com
181441	Carl Bob	Receptionist/Lifestyle Specialist	cbob@fitlife.com
171442	Tony Tiger	Yoga Instructor	ttiger@fitlife.com
123456	Issac Newton	Swimming Instructor	inewton@fitlife.com
123457	Bob Marley	Personal Instructor	bmarley@fitlife.com
123458	Nicola Tesla	Swimming Instructor	ntesla@fitlife.com
985724	Trevor Scott	Personal Trainer	scotty@fitlife.com
123443	Ali Baba	Manager	alibaba@fitlife.com
435794	Jaden Smith	Assistant Manager	jsmith@fitlife.com
342343	Trevor Noah	Yoga Instructor	tnoah@fitlife.com
757656	Jimmy Fallon	Personal Instructor	jfallon@fitlife.com
634566	Stephen Colbert	Receptionist/Lifestyle Specialist	scolbert@fitlife.com

13 rows returned in 0.00 seconds Download

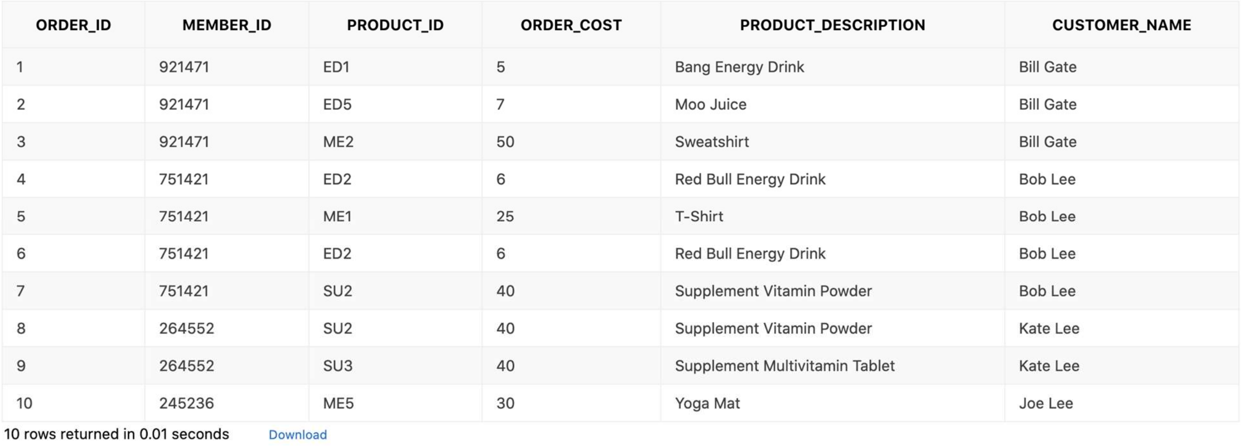
11. select other_program.member_id, other_program.employee_id, employee_name, other_name as program_name
from other_program, membership, employee
where other_program.member_id = membership.member_id
and other_program.employee_id = employee.employee_id;
(**Displays the member ID, employee ID, name of the employee, and name of the program so the user can see relevant information about their program if they're registered to "other" program**)



MEMBER_ID	EMPLOYEE_ID	EMPLOYEE_NAME	PROGRAM_NAME
345678	171442	Tony Tiger	Yoga
245236	123456	Issac Newton	Swimming

2 rows returned in 0.00 seconds Download

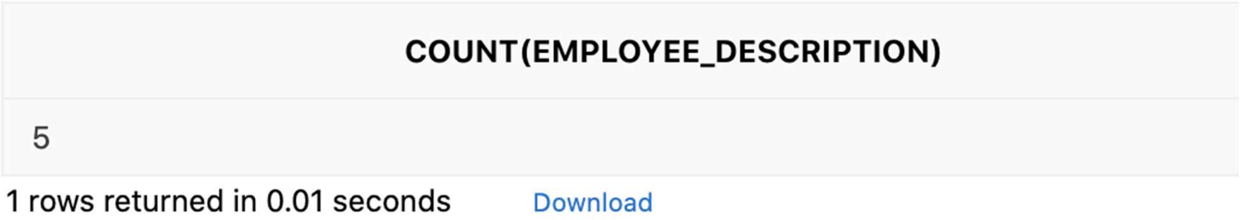
12. select s.order_id, s.member_id, s.product_id, s.order_cost, a.product_description, c.customer_name from store_orders s, store a,membership b, reception c
where c.appointment_number=b.appointment_number and b.member_id=s.member_id and s.product_id=a.product_id;
(Sales report from fitlife inc store orders)



ORDER_ID	MEMBER_ID	PRODUCT_ID	ORDER_COST	PRODUCT_DESCRIPTION	CUSTOMER_NAME
1	921471	ED1	5	Bang Energy Drink	Bill Gate
2	921471	ED5	7	Moo Juice	Bill Gate
3	921471	ME2	50	Sweatshirt	Bill Gate
4	751421	ED2	6	Red Bull Energy Drink	Bob Lee
5	751421	ME1	25	T-Shirt	Bob Lee
6	751421	ED2	6	Red Bull Energy Drink	Bob Lee
7	751421	SU2	40	Supplement Vitamin Powder	Bob Lee
8	264552	SU2	40	Supplement Vitamin Powder	Kate Lee
9	264552	SU3	40	Supplement Multivitamin Tablet	Kate Lee
10	245236	ME5	30	Yoga Mat	Joe Lee

10 rows returned in 0.01 seconds Download

13. select count(employee_description) from employee
    where employee_description='Personal Trainer';
    (**Displays count of employees with the job description 'Personal Trainer'**)



COUNT(EMPLOYEE_DESCRIPTION)
5

1 rows returned in 0.01 seconds Download