

Running head: EZ GYM DATABASE DESIGN

EZ Gym Database Design  
Zailyn Tamayo and Jaime D. Rodriguez  
National University

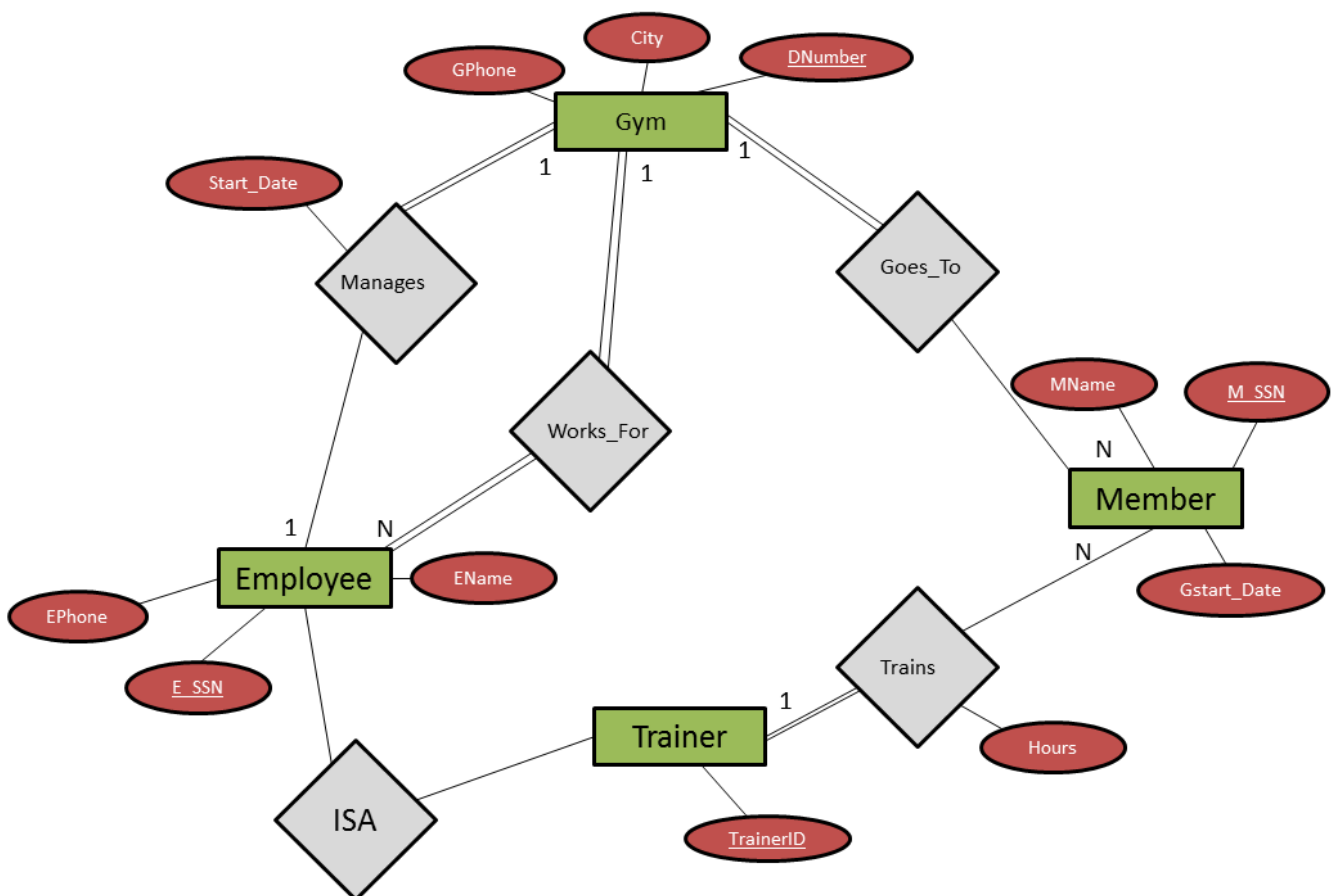
## Introduction

Our database system designed is for a gym company called EZ Gym which has two locations. The Entity Relationship Diagram has four Entities and five Relations. Our first entity is the Gym, which then breaks down into Employees and Members. From there, Employees can also be Trainers who train members.

The purpose of this database design is to map out the Relational Data Base System for good record and transaction keeping on the business and avoid redundancy with storage and to maximize performance during the execution of queries.

## The Entity Relation Diagram

### ER Diagram



The business rules are as follows. The EZ Gym Corporation owns two gym locations. Each gym location needs to have a manager. One employee can only manage one gym. A gym can have several employees but each employee can work for only one gym. One gym can have several members but each member can only go to the gym at which they are members. The gym needs members in order to stay open; however, members do not need to go to the gym because they pay for membership regardless of attendance. Some employees are trainers. A trainer needs to have at least one member to train and can train multiple members, but a member does not need to have a trainer.

The Relational Schema:

Gym (Dnumber, Dphone, City)

Employee (Essn, Ename, EPhone, Dnumber)

Manages (Start\_date, Dnumber, Essn)

Member (Mssn, Mname, Gstart\_date, Dnumber)

Trainer (Trainer\_ID, Essn)

Trains (Hours, Mssn, Trainer\_ID)

Gym

Dnumber	DPhone	City
001	310-828-7566	Westchester
002	909-355-9812	Chino Hills

Employee

Essn	Ename	Ephone	Dnumber
626-87-5403	Christopher Rowen	619-585-4987	001
405-22-5458	Sheryl Watts	818-546-1238	001
605-40-3214	Shawn Ellis	213-548-7546	002
304-45-7540	Consuelo Munoz	909-725-7299	002
244-88-0125	Brandon Long	951-265-4407	001
118-44-9376	Jessica Black	909-398-8064	002

Manages

Start_date	Dnumber	Essn
02/15/2010	001	626-87-5403
07/31/2013	002	304-45-7540

Members

Mssn	Mname	Gstart_date	Dnumber
111-22-3333	Jane Smith	01/04/2013	001
123-44-5677	Dan Mason	03/10/2014	002
456-99-0824	Diana Cain	11/16/2012	002
986-88-1155	John Woo	12/02/2012	001
777-35-2690	Nancy Garcia	08/09/2013	001
842-12-6789	Zack Snow	09/12/2013	002

Trainer

Trainer_ID	Essn
T001	405-22-5458
T002	605-40-3214

Trains

Hours	Mssn	Trainer_ID
10	456-99-0824	T002
2	777-35-2690	T001
18	842-12-6789	T002

Our normalization process included rigorous testing ensuring that there was no redundancy data unless otherwise absolutely necessary for query optimization. When our data for the relational schemas are in second normal form, full functional dependency would occur if we deleted an attribute and that dependency did not hold (Elmasri, 2011).

However, now that our database is normalized to 3NF or has become a transitive dependency, there now exists a set of attributes that are neither a candidate key nor a subset of any key. Through the decomposition of Employee schema we can get the Manages table. However, during the creation process we broke down the Employee table into Third normal form. That is, the Manages relationship was created with Start\_date, Dnumber, and Essn as its attributes and its primary key is the composite key Dnumber and ESSN.

Table Name: Gym



Table Name: Manages

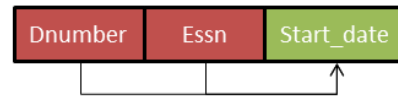


Table Name: Employee



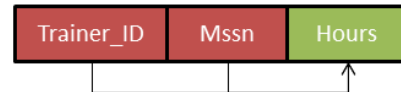
Table Name: Trainer



Table Name: Member



Table Name: Trains



The CREATE TABLE statements for the database:

```
CREATE TABLE gym (  
    dnumber char(3) primary key,  
    dphone char(10),  
    city varchar2(20),  
);
```

```
CREATE TABLE employee (  
    essn char(9) not null,  
    ename varchar2(32),  
    ephone char(10),  
    dnumber char(3) not null references gym(dnumber),  
    primary key (essn, dnumber));
```

```
CREATE TABLE manages (  
    start_date date,  
    dnumber char(3) not null references gym(dnumber),  
    essn char(9) not null references employee(essn),  
    primary key (dnumber, essn));
```

```
CREATE TABLE members (  
    mssn char(9) not null,  
    mname varchar2(32),  
    gstart_date date,  
    dnumber char(3) not null references gym(dnumber),  
    primary key (mssn, dnumber));
```

```
CREATE TABLE trainer (
    trainer_id char(4),
    essn char(9) not null references employee(essn),
    primary key (trainer_id, essn));
```

```
CREATE TABLE trains (
    hours decimal(4,1),
    mssn char(9) not null references members(mssn),
    trainer_id char(4) not null references trainer(trainer_id),
    primary key (mssn, trainer_id));
```

The INSERT statements for the database:

```
INSERT INTO gym (dnumber, dphone, city) VALUES ('001', '3108287566', 'Westchester');
```

```
INSERT INTO gym (dnumber, dphone, city) VALUES ('002', '9093559812', 'Chino Hills');
```

```
INSERT INTO employee (essn, ename, ephone, dnumber) VALUES ('626875403', 'Christopher Rowen', '6195854987', '001');
```

```
INSERT INTO employee (essn, ename, ephone, dnumber) VALUES ('405225458', 'Sheryl Watts', '8185461238', '001');
```

```
INSERT INTO employee (essn, ename, ephone, dnumber) VALUES ('605403214', 'Shawn Ellis', '2135487546', '002');
```

```
INSERT INTO employee (essn, ename, ephone, dnumber) VALUES ('304457540', 'Consuelo Munoz', '9097257299', '002');
```

```
INSERT INTO employee (essn, ename, ephone, dnumber) VALUES ('244880125', 'Brandon Long', '9512654407', '001');
```

```
INSERT INTO employee (essn, ename, ephone, dnumber) VALUES ('118449376', 'Jessica Black', '9093988064', '002');
```

```
--
```

```
INSERT INTO manages (start_date, dnumber, essn) VALUES (parsedatetime('15-02-2010 00:00:00.00', 'dd-MM-yyyy hh:mm:ss.SS'), '001', '626875403');
```

```
INSERT INTO manages (start_date, dnumber, essn) VALUES (parsedatetime('31-07-2013 00:00:00.00', 'dd-MM-yyyy hh:mm:ss.SS'), '002', '304457540');
```

--

```
INSERT INTO members (mssn, mname, gstart_date, dnumber) VALUES ('111223333', 'Jane Smith', parsedatetime('04-01-2013 00:00:00.00', 'dd-MM-yyyy hh:mm:ss.SS'),'001');
```

```
INSERT INTO members (mssn, mname, gstart_date, dnumber) VALUES ('123445677', 'Dan Mason', parsedatetime('10-03-2014 00:00:00.00', 'dd-MM-yyyy hh:mm:ss.SS'),'002');
```

```
INSERT INTO members (mssn, mname, gstart_date, dnumber) VALUES ('456990824', 'Diana Cain', parsedatetime('16-11-2012 00:00:00.00', 'dd-MM-yyyy hh:mm:ss.SS'),'002');
```

```
INSERT INTO members (mssn, mname, gstart_date, dnumber) VALUES ('986881155', 'John Woo', parsedatetime('02-12-2012 00:00:00.00', 'dd-MM-yyyy hh:mm:ss.SS'),'001');
```

```
INSERT INTO members (mssn, mname, gstart_date, dnumber) VALUES ('777352690', 'Nancy Garcia', parsedatetime('09-08-2013 00:00:00.00', 'dd-MM-yyyy hh:mm:ss.SS'),'001');
```

```
INSERT INTO members (mssn, mname, gstart_date, dnumber) VALUES ('842126789', 'Zack Snow', parsedatetime('12-09-2013 00:00:00.00', 'dd-MM-yyyy hh:mm:ss.SS'),'002');
```

--

```
INSERT INTO trainer (trainer_id, essn) VALUES ('T001', '405225458');
```

```
INSERT INTO trainer (trainer_id, essn) VALUES ('T002', '605403214');
```

--

```
INSERT INTO trains (hours, mssn, trainer_id) VALUES ('10', '456990824', 'T002');
```

```
INSERT INTO trains (hours, mssn, trainer_id) VALUES ('2', '777352690', 'T001');
```

```
INSERT INTO trains (hours, mssn, trainer_id) VALUES ('18', '842126789', 'T002');
```

### Queries for Validation

1. List names and gym number of employees who are not managers or trainers.

```
SELECT ename, dnumber
FROM Employee e
WHERE NOT EXISTS (SELECT essn FROM manages m WHERE e.essn = m.essn)
AND NOT EXISTS (SELECT essn FROM trainer t WHERE e.essn = t.essn);
```

2. List the member name and gym number of members who do not receive training.

```
SELECT m.mname, m.dnumber
FROM members m
WHERE NOT EXISTS (SELECT mssn FROM trains t WHERE m.mssn = t.mssn);
```



3. List the start date, name, phone number and gym number of all managers.

```
SELECT start_date, ename, ephone, e.dnumber  
FROM employee e, manages m  
WHERE e.essn = m.essn;
```

4. List the sum of hours that members have trained at Gym 002.

```
SELECT SUM(hours)  
FROM trains t, members m, gym g  
WHERE t.mssn = m.mssn AND m.dnumber = g.dnumber AND m.dnumber = 002;
```

### Conclusion

The purpose of this project was to design a database; we chose to design a database for a made-up gym corporation called EZ Gym. First we started by creating an Entity Relation Diagram, drawing out the entities and relations first; then we added the attributes and cardinalities. For this step, we learned to think about primary keys and also about what attributes were necessary in order for our database to be useful. Another lesson learned was that, if we took the time to make a detailed and well-designed ERD, the process of converting the ERD to a relational schema would be much simpler.

The SQL part of the project was educational through the practice of writing the script. The project overall was a very positive experience and with the finalization of this paper, we are left with a sense of accomplishment at the fact that we have thought up, designed, created and placed into work a database.

### References

Elmasri, R. (2011). *Fundamentals of Database Systems* (6<sup>th</sup> ed.) Boston: Pearson.