Mapping Relationships to Databases

One-To-One Relationships

In a one-to-one relationship, one object in one class is related to at most one object in another class. In Figure 1, each object of FatClass references one object in SkinnyClass.

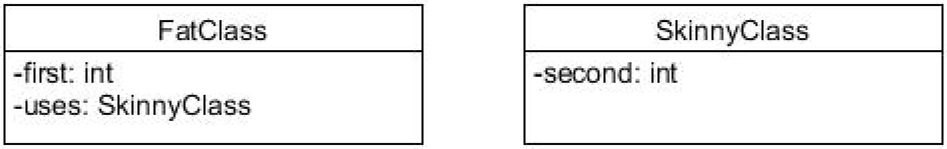

Figure 1: One-To-One Relationship in a Class Diagram

The ERD reflecting this relationship would be:

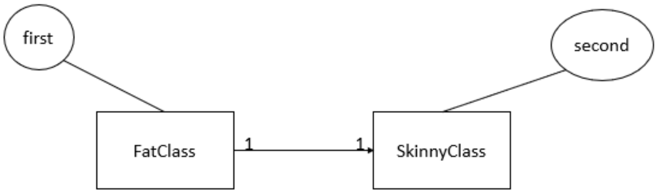

Figure 2: One-To-One Relationship in an ER Diagram

Coding a one-to-one relationship in a database only requires that the table representing the referring class has a column that contains an ID of an object in the referenced class.

```
CREATE TABLE SkinnyClass
          (SkinnyClassID int NOT NULL AUTO_INCREMENT,second int),
          PRIMARY KEY SkinnyClassID;
CREATE TABLE FatClass
          (FatClassID int NOT NULL AUTO_INCREMENT,first int),
          PRIMARY KEY FatClassID,
          FOREIGN KEY Uses REFERENCES SkinnyClass(SkinnyClassID);
```

One-To-Many Relationships

In a one-to-many relationship, one object in one class is related to zero or more objects in another class. In Figure 3, each object of FatClass references many objects in SkinnyClass.

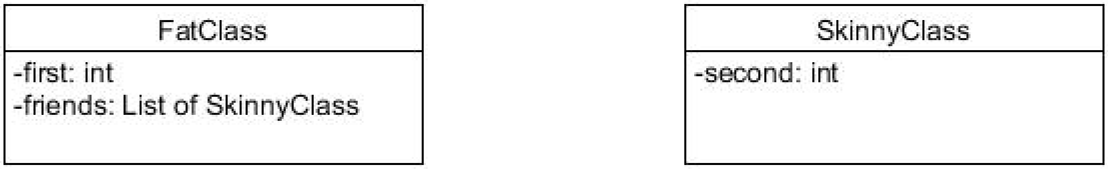

Figure 3: One-To-Many Relationship in a Class Diagram

The ERD reflecting this relationship would be:

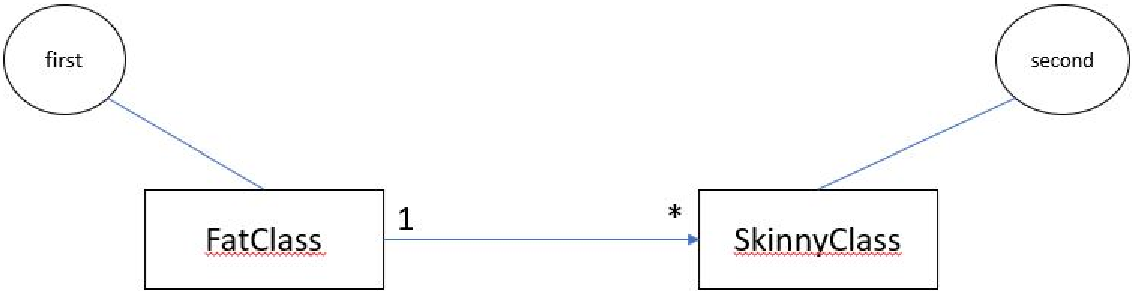

Figure 4: One-To-Many Relationship in an ER Diagram

Coding a one-to-many relationship in a database requires that the table representing the ”many” side of the relationship has a column that contains an ID of an object in the ”one” side of the relationship.

```
CREATE TABLE SkinnyClass
        (SkinnyClassID int NOT NULL AUTO_INCREMENT, second int),
        PRIMARY KEY SkinnyClassID,
        FOREIGN KEY Uses REFERENCES FatClass(FatClassID);
CREATE TABLE FatClass
        (FatClassID int NOT NULL AUTO_INCREMENT, first int),
        PRIMARY KEY FatClassID;
```

Many-To-Many Relationships

In a many-to-many relationship, one object in class A is related to zero or more objects in class B and *vice versa*. The classic example for this is students and courses in a university. Every student can have multiple courses and every course can have multiple classes In Figure 5, each object of FatClass references many objects in SkinnyClass and *vice versa*.

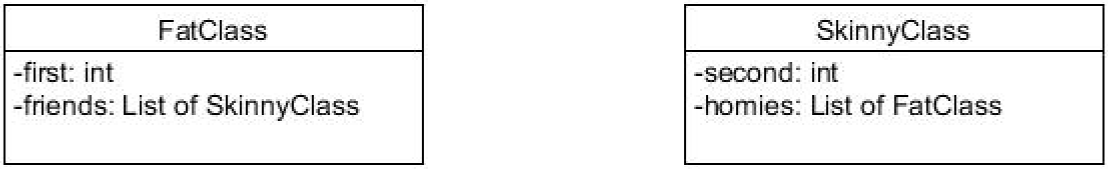

Figure 5: Many-To-Many Relationship in a Class Diagram

The ERD reflecting this relationship would be:

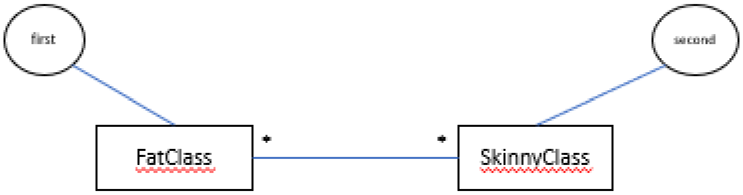

Figure 6: One-To-Many Relationship in an ER Diagram

Coding a many-to-many relationship in a database requires an extra table. This table contains two columns: one for an ID of each class. Querying this table can retrieve the relationship from either side.

```
CREATE TABLE SkinnyClass
          (SkinnyClassID int NOT NULL AUTO_INCREMENT,second int),
         PRIMARY KEY SkinnyClassID;
CREATE TABLE FatClass
          (FatClassID int NOT NULL AUTO_INCREMENT, first int),
         PRIMARY KEY FatClassID;
CREATE TABLE FatAndSkinny
     SkinnyClassID int NOT NULL,
          FatClassID int NOT NULL,
         FOREIGN KEY (SkinnyClassID) REFERENCES SkinnyClass(SkinnyClassID),
         FOREIGN KEY (FatClassID) REFERENCES FatClass(FatClassID),
         UNIQUE (SkinnyClassID, FatClassID)
```