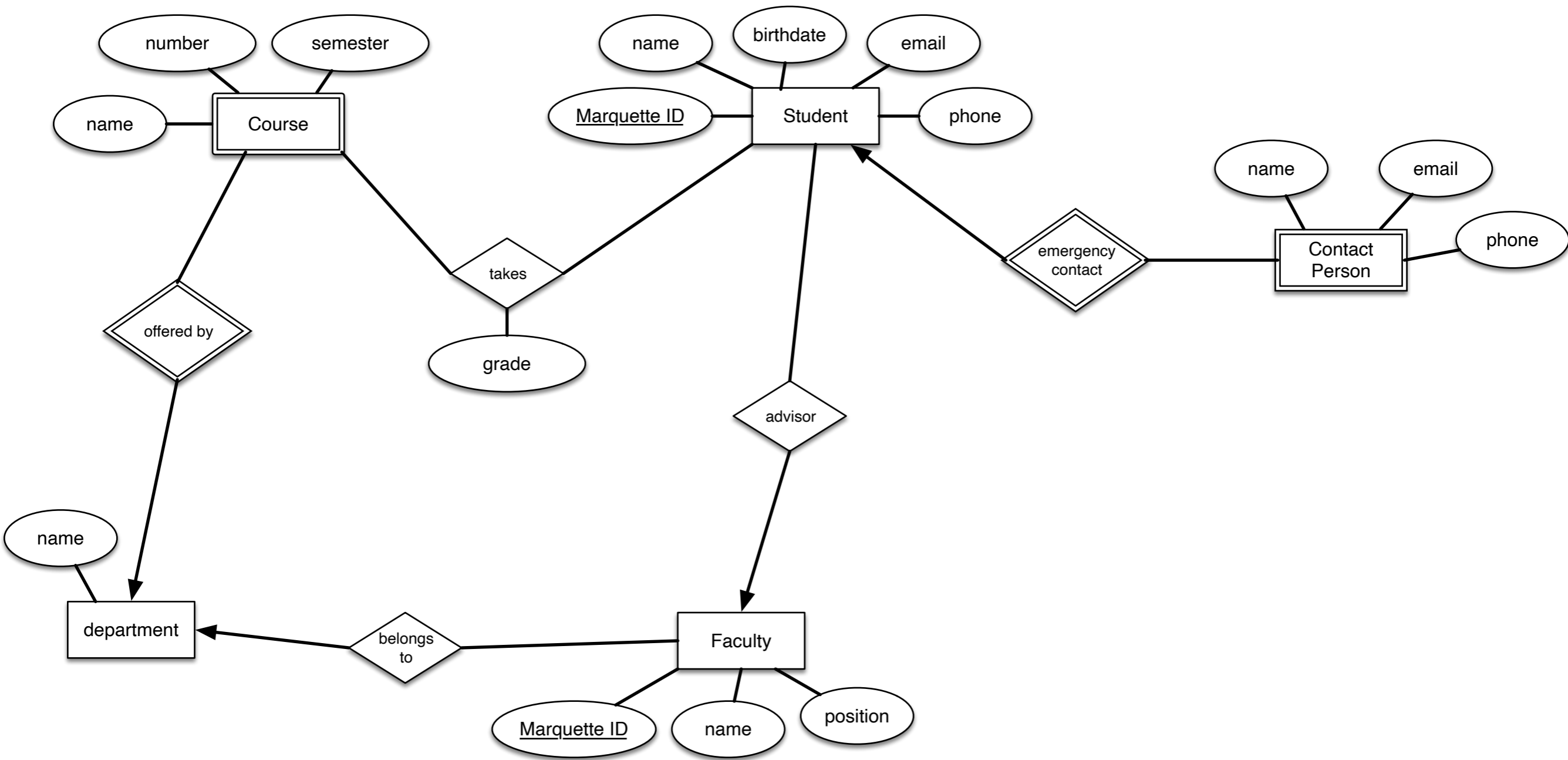


Repetition

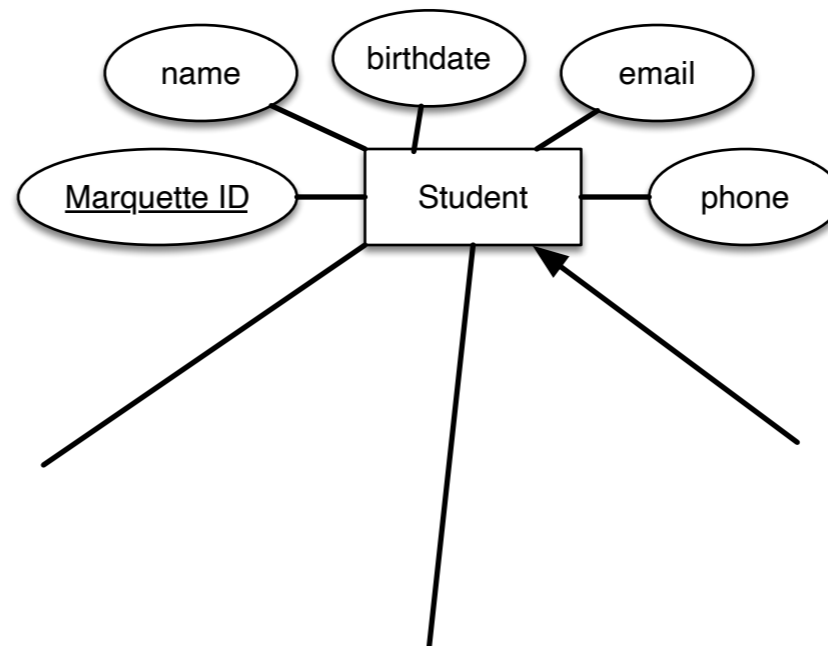
From E/R to Database

E/R Diagram



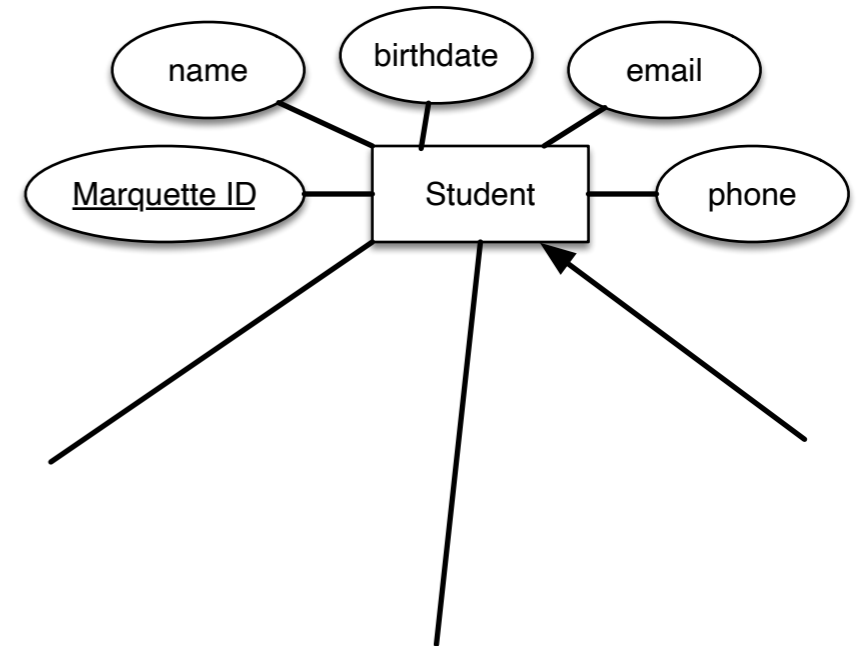
Schemata

- Create a table for an entity
- Attributes are properties
- `student (MID, name, birthdate, email, phone)`



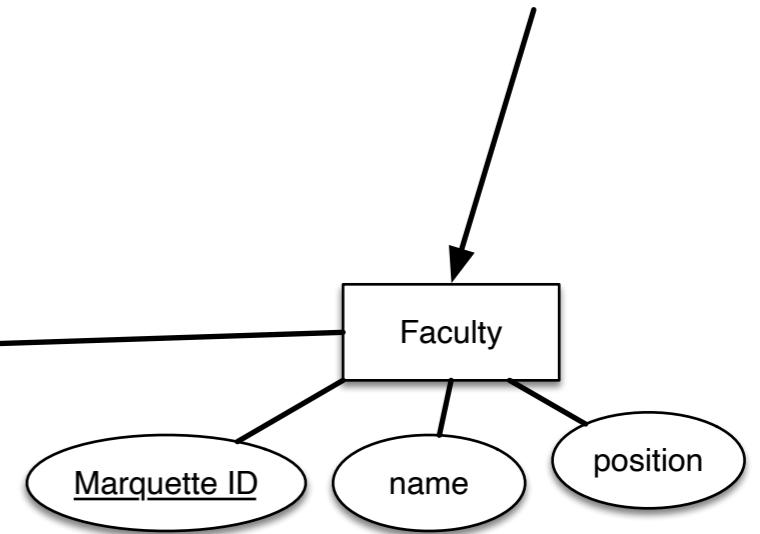
SQL Table

```
CREATE DATABASE marquette;  
USE marquette;  
  
CREATE TABLE student (  
    MID CHAR(12),  
    birthdate DATE,  
    email CHAR(64) NOT NULL,  
    name VARCHAR(64) NOT NULL,  
    PRIMARY KEY (MID)  
);
```

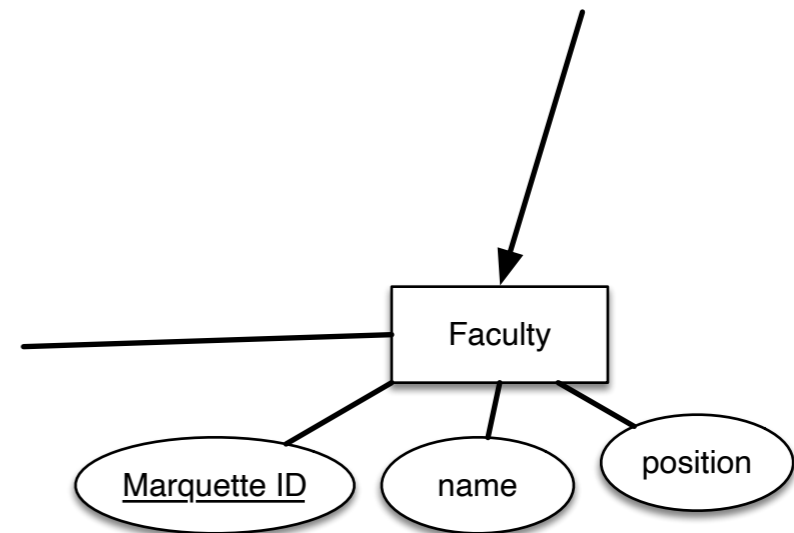


Schemata

- `faculty(MID, name, position)`



SQL Table

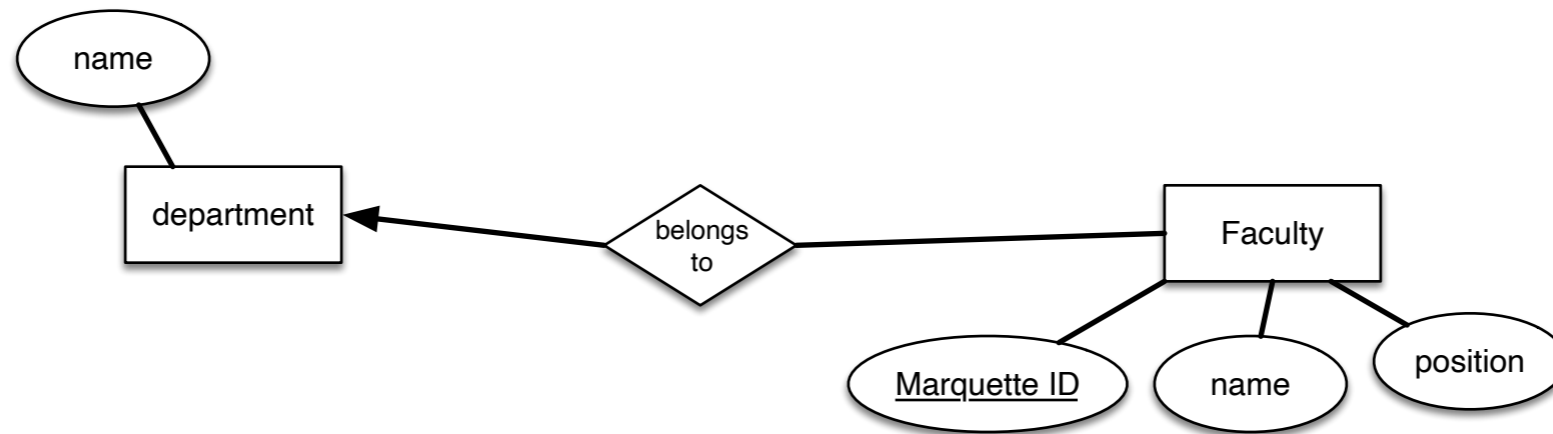


```
CREATE TABLE faculty (  
    MID CHAR(12),  
    name VARCHAR(64) NOT NULL,  
    position VARCHAR(32) NOT NULL  
);
```

Schemata

- Relationships:
 - Create a table for the relationship set.
 - Add all primary keys of the participating entity sets as fields of the table.
 - Add a field for each attribute of the relationship.
Declare a primary key using the key fields from the source entity set only.
 - Declare foreign key constraints for all the fields from the source and target entity sets.

Schemata



department (name)

faculty (MID, name, position)

belongsTo (MID, departmentName)

SQL Table

```
CREATE TABLE department (  
    name VARCHAR(32),  
    PRIMARY KEY (name)  
);
```

```
CREATE TABLE belongsTo (  
    MID CHAR(12),  
    departmentName VARCHAR(32) NOT NULL,  
    PRIMARY KEY (MID),  
    CONSTRAINT fk_fac_depname  
    FOREIGN KEY (departmentName)  
        REFERENCES department (name)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
);
```

SQL Table

- Modeling many-to-one relationships is a bit iffy.
 - To make sure that all faculty have a department, we can integrate the department table into the faculty table.
 - This can create problems if we need departments again

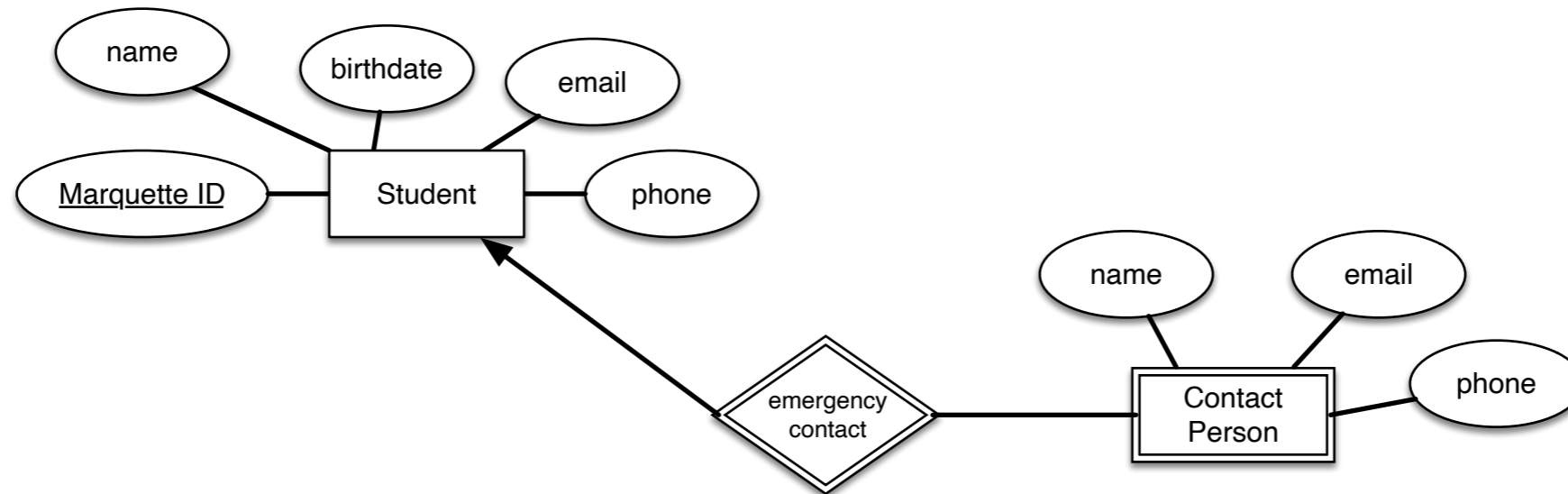
SQL Table

```
CREATE TABLE facultyAlt (  
    MID CHAR(12),  
    name VARCHAR(64) NOT NULL,  
    position VARCHAR(32) NOT NULL,  
    departmentName VARCHAR(32) NOT NULL,  
    PRIMARY KEY(MID)  
);
```

Schemata

- Weak entities:
 - Create a table for the weak entity
 - Make each attribute of the weak entity set a field of the table
 - Add fields for the primary key attributes of the identifying owner
 - Declare a foreign key constraint on these identifying owner fields
 - Automatically delete any tuples in the table for which there are no owners

Schemata

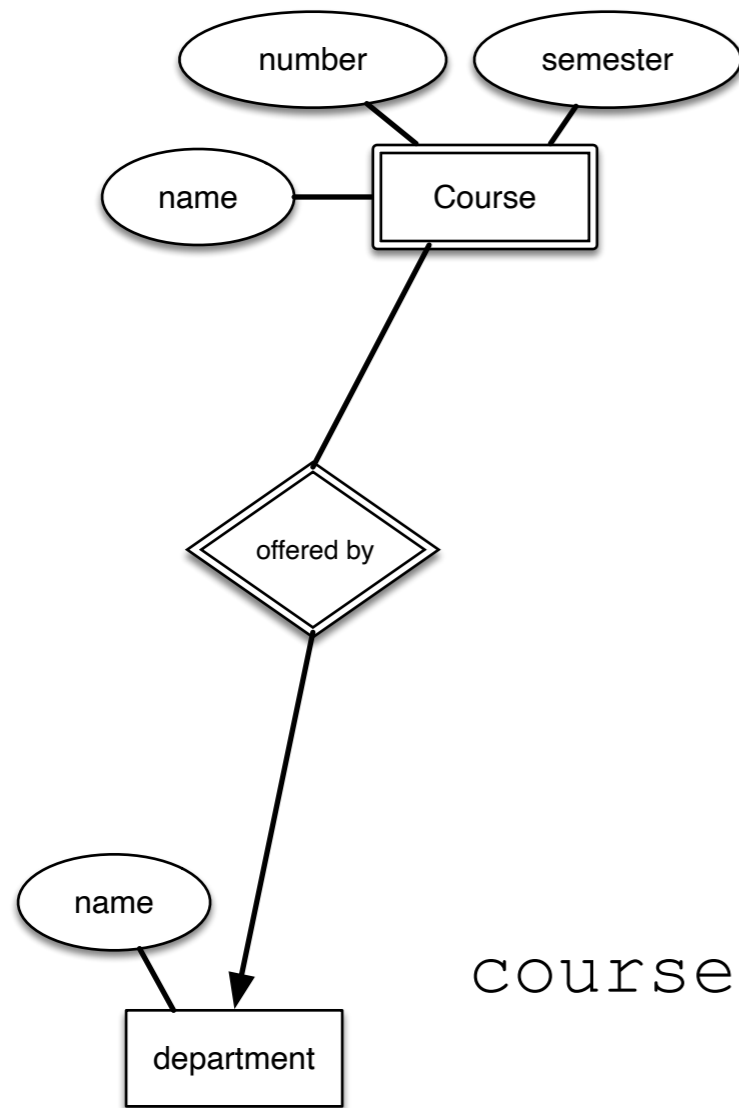


```
contactPerson(name, email, phone, studentMID)
```

SQL Table

```
CREATE TABLE contactPerson (  
    name VARCHAR(32),  
    phone CHAR(15),  
    email VARCHAR(64),  
    studentMID CHAR(12),  
    PRIMARY KEY (name),  
    CONSTRAINT fk_cp_st FOREIGN KEY (studentMID)  
        REFERENCES student (MID)  
        ON DELETE CASCADE  
);
```

Schemata

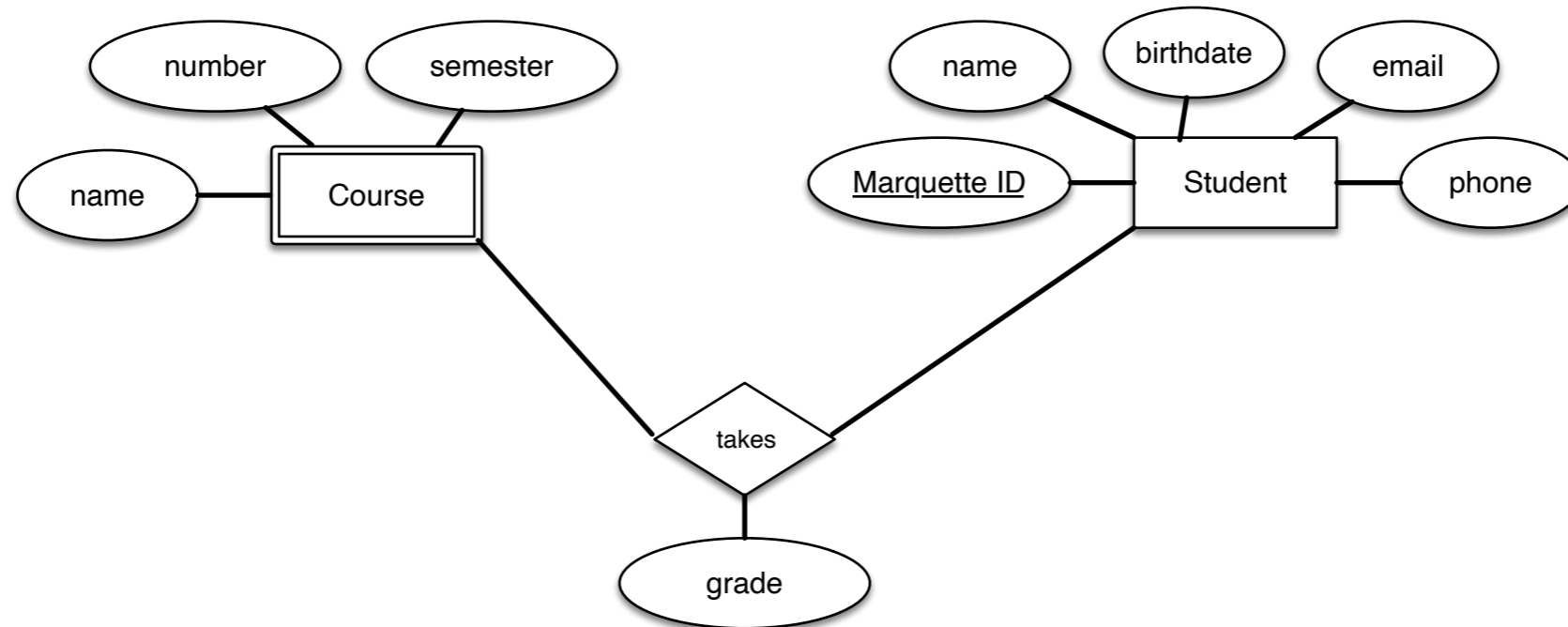


`courses (deptName, number, semester, name)`

SQL Table

```
CREATE TABLE courses (  
    name VARCHAR(64) NOT NULL,  
    number CHAR(3),  
    deptName VARCHAR(32),  
    semester CHAR(5),  
    PRIMARY KEY (deptName , number , semester),  
    CONSTRAINT fk_cs_dpt FOREIGN KEY (deptName)  
        REFERENCES department (name)  
        ON DELETE CASCADE  
);
```

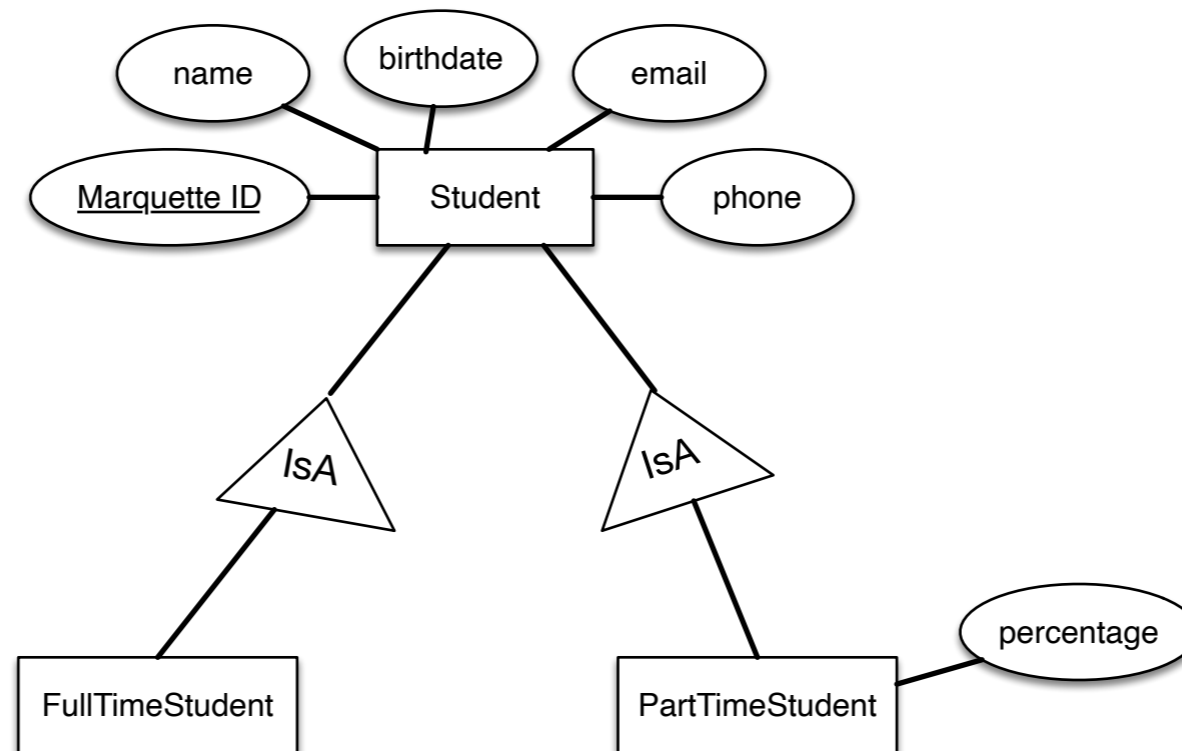

Schemata



takes (MID, deptName, number, semester, grade)

Is A Relationships

- There are many ways to deal with IsA relationship
 - Assume we have now part-time students



Schemata

- We create one table for the super-category and add tables for each sub-category with the additional attributes
- This solution only needs one more table

```
partTimeStudent (MID, percentage)
```

- We declare a foreign key constraint to ensure that the entry of a subtable corresponds to a table in the super

SQL Table

```
CREATE TABLE partTimeStudent (  
    MID CHAR(6),  
    percentage DECIMAL(2, 2),  
    PRIMARY KEY (MID),  
    FOREIGN KEY (MID)  
        REFERENCES student (MID)  
);
```