James Hippler
Helen Stockman
CS 340-400
Final Project
Due: December 03, 2017

Hospital Database (Final Project)

Outline

We will be creating a database that represents the various entities of a hospital. Hospitals require many distinct entities to operate, affording the opportunity to incorporate additional levels of complex and meaningful relationships into the database. The hospital environment is compelling because of the extent to which administrators rely on data to make informed and timely decisions regarding patient health care.

Our database will represent the relationship between human, organizational, and medical entities.

Database Outline

Patients

Each patient has the attributes patient_id (which is a unique identifier), first name, middle name, last name, date of birth, phone number, and insurance provider.

A patient may have at most one address, which will be referenced by a foreign key address_id. In some cases patients may not have an address, for example an urban hospital that serves a large homeless population. If a patients has more than one address, only one will be recorded as we only need one primary mailing address.

Patients also may have up to one doctor, which will be referenced by a foreign key doctor_id. In cases where the patient has not been triaged, they may not have a doctor yet. In cases of a complex diagnosis where they may have a team of doctors, only one will be considered the primary.

Patients may have up to one department, which will be referenced by a foreign key department_id. In cases where the patient has not been triaged, they may not have a department yet. A patient can only be treated and located in at most one department.

Patients also may have up to one diagnosis, which will be referenced by a foreign key diagnosis_id. In cases where the patient is new, they may not have a diagnosis yet. In cases of a complex diagnosis where there many be many distinct diagnoses, only one will be considered the primary.

James Hippler
Helen Stockman
CS 340-400
Final Project
Due: December 03, 2017

Doctors

Each doctor has the attributes eid (employee id- which is a unique identifier), first name, middle name, last name, and phone number.

A doctor must have exactly one address, which will be referenced by a foreign key address_id. If a doctor has more than one address, only one will be recorded as we only need one primary mailing address.

A doctor may have many patients. In cases where a doctor is new or in training they may not yet have any patients assigned. Most often though, a doctor may several patients in their care at any one time.

A doctor must belong to exactly one department, which will be reference by a foreign key department_id. A doctor must be hired into a department, and may only work for that department.

A doctor may have between 0 and many specialties.

Departments

Each department has the attributes did (department id- which is a unique identifier) and a name.

A department must have at least one doctor, and most often has several.

A department may have between 0 and many patients.

Specialty

Each specialty has the attributes sid (specialty id which is a unique identifier) and a name.

A specialty may be shared by between 0 and many doctors.

Diagnosis

Each diagnosis has the attributes diid (diagnosis id-which is a unique identifier), a name, and a severity.

A diagnosis may be shared by between 0 and many patients.

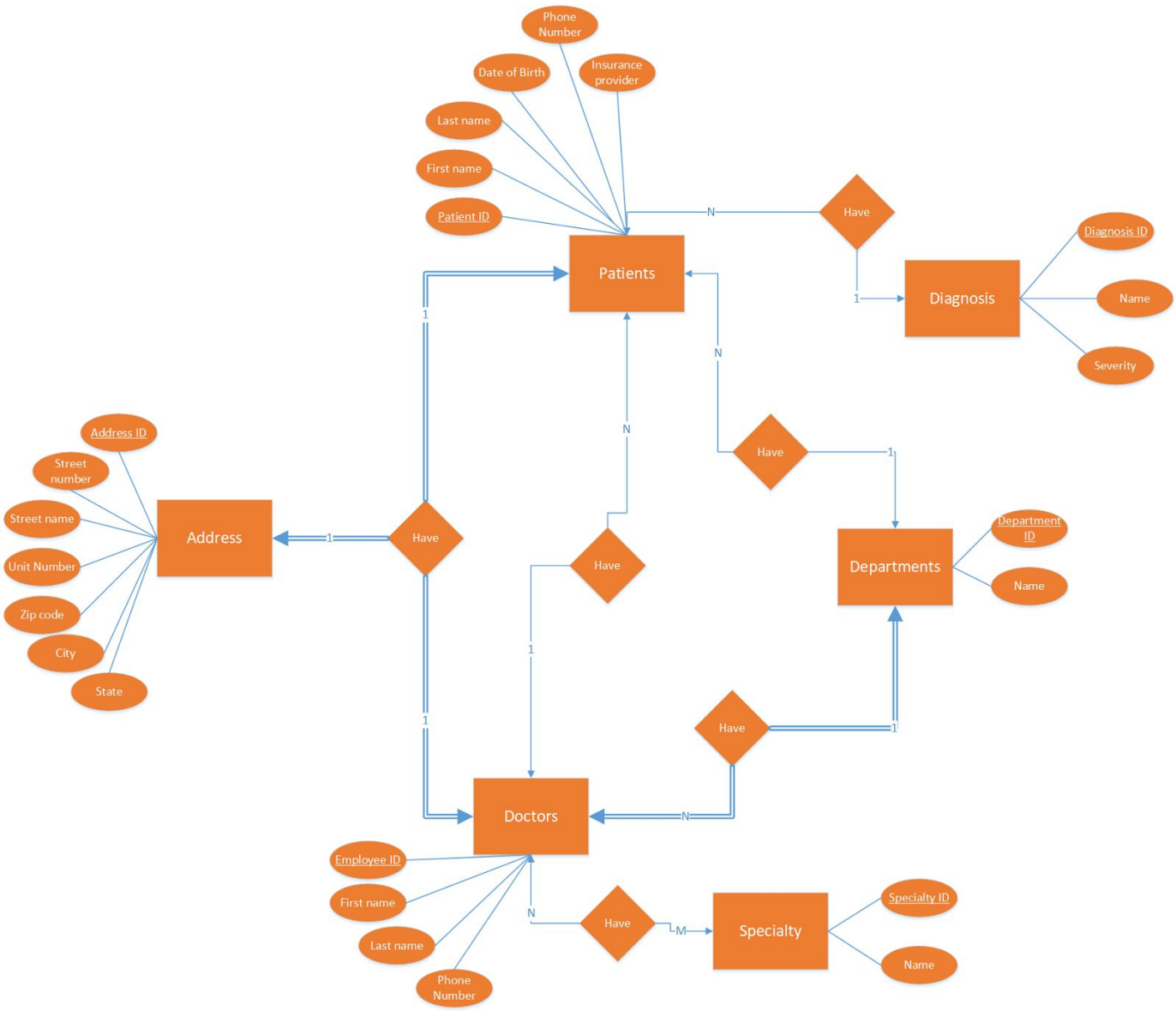
James Hippler
Helen Stockman
CS 340-400
Final Project
Due: December 03, 2017

Address

Each address has the attributes aid (address id-which is a unique identifier), a street number, a street name, a unit number, a zip code, a city, and a state.

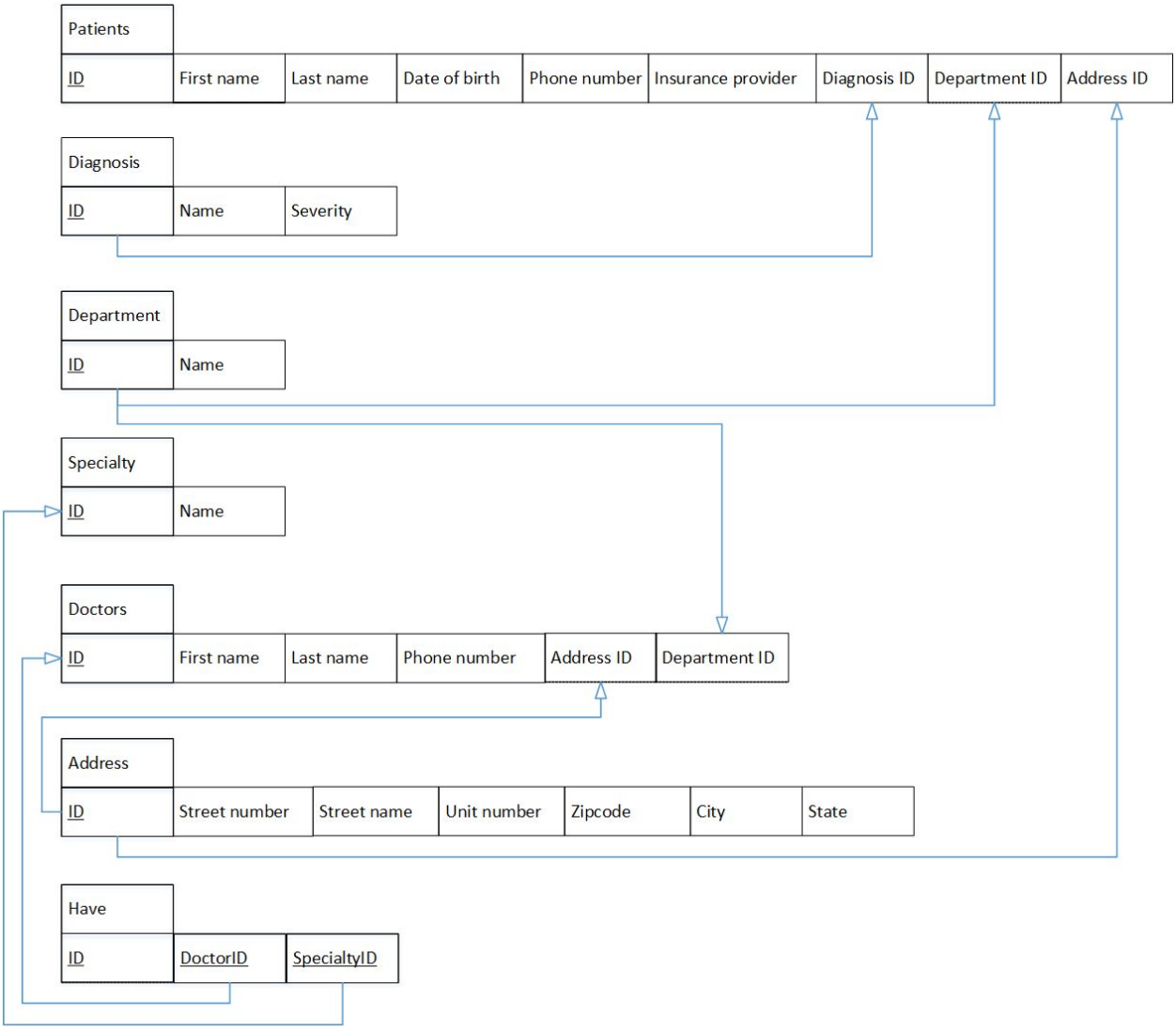
An address must belong to exactly one patient or one doctor.

ER Diagram



James Hippler
Helen Stockman
CS 340-400
Final Project
Due: December 03, 2017

Schema



Data Definition Queries

```
CREATE TABLE address (
    aid INT NOT NULL AUTO_INCREMENT,
    streetNum INT,
    streetName VARCHAR(255),
    unit INT,
    zip INT,
    city VARCHAR(255),
    state CHAR(2),
```

```
    PRIMARY KEY (aid)
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;

CREATE TABLE diagnosis (
  diid INT NOT NULL AUTO_INCREMENT,
  name VARCHAR(255) NOT NULL,
  severity INT NOT NULL,
  PRIMARY KEY (diid)
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;

CREATE TABLE specialty (
  sid INT NOT NULL AUTO_INCREMENT,
  name VARCHAR(255) NOT NULL,
  PRIMARY KEY (sid)
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;

CREATE TABLE department (
  did INT NOT NULL AUTO_INCREMENT,
  name VARCHAR(255) NOT NULL,
  PRIMARY KEY (did)
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;

CREATE TABLE doctors (
    eid INT NOT NULL AUTO_INCREMENT,
    first_name VARCHAR(255) NOT NULL,
    last_name VARCHAR(255) NOT NULL,
    middle_name VARCHAR(255),
    phone CHAR(10) NOT NULL,
    address_id INT NOT NULL,
    department_id INT NOT NULL,
    PRIMARY KEY (eid),
    FOREIGN KEY (address_id) REFERENCES address (aid),
    FOREIGN KEY (department_id) REFERENCES department (did)
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;

CREATE TABLE patients (
    pid INT NOT NULL AUTO_INCREMENT,
    first_name VARCHAR(255) NOT NULL,
    last_name VARCHAR(255) NOT NULL,
    middle_name VARCHAR(255),
    dob DATE,
    phone CHAR(10) NOT NULL,
```

James Hippler
Helen Stockman
CS 340-400
Final Project
Due: December 03, 2017

```
    insurance VARCHAR(255),
    address_id INT,
    doctor_id INT,
    department_id INT,
    diagnosis_id INT,
    PRIMARY KEY (pid),
    FOREIGN KEY (address_id) REFERENCES address (aid),
    FOREIGN KEY (doctor_id) REFERENCES doctors (eid) ON DELETE SET NULL,
    FOREIGN KEY (department_id) REFERENCES department (did),
    FOREIGN KEY (diagnosis_id) REFERENCES diagnosis (diid)
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;

CREATE TABLE doctor_specialty (
    dsid INT NOT NULL AUTO_INCREMENT,
    eid INT NOT NULL,
    sid INT NOT NULL,
    PRIMARY KEY (dsid),
    FOREIGN KEY (eid) REFERENCES doctors(eid) ON DELETE CASCADE,
    FOREIGN KEY (sid) REFERENCES specialty(sid) ON DELETE CASCADE
)ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;
```

Data Manipulation Queries

Search all doctors with a certain specialty

```
SELECT d.first_name, d.last_name, s.name
FROM doctors d
JOIN doctor_specialty ds ON d.eid = ds.eid
JOIN specialty s ON ds.sid = s.sid
WHERE s.name = [specialtyNameInput]
```

Search all patients who live in a certain zipcode

```
SELECT p.first_name, p.last_name, a.zip
FROM patients p
JOIN address a ON p.address_id = a.aid
WHERE a.zip = [zipInput]
```

James Hippler
Helen Stockman
CS 340-400
Final Project
Due: December 03, 2017

Search all patients who have a diagnosis severity greater than a certain number

```
SELECT p.first_name, p.last_name, d.severity
FROM patients p
JOIN diagnosis d ON p.diagnosis_id = d.diid
WHERE d.severity > [severityInput]
```

Search all patients who have a diagnosis severity lower than a certain number

```
SELECT p.first_name, p.last_name, d.severity
FROM patients p
JOIN diagnosis d ON p.diagnosis_id = d.diid
WHERE d.severity < [severityInput]
```

Get the count of all doctors in a certain department

```
SELECT department.name AS ‘Department_Name’, COUNT(*) AS ‘Doctors_Staffed’
FROM doctors
JOIN department ON doctors.department_id = department.did
WHERE department.did = [departmentInput]
```

Get the count of all patients in a certain department

```
SELECT department.name AS ‘Department_Name’, COUNT(*) AS ‘Patients_Admitted’
FROM patients
JOIN department ON patients.department_id = department.did
WHERE department.did = [departmentInput]
```