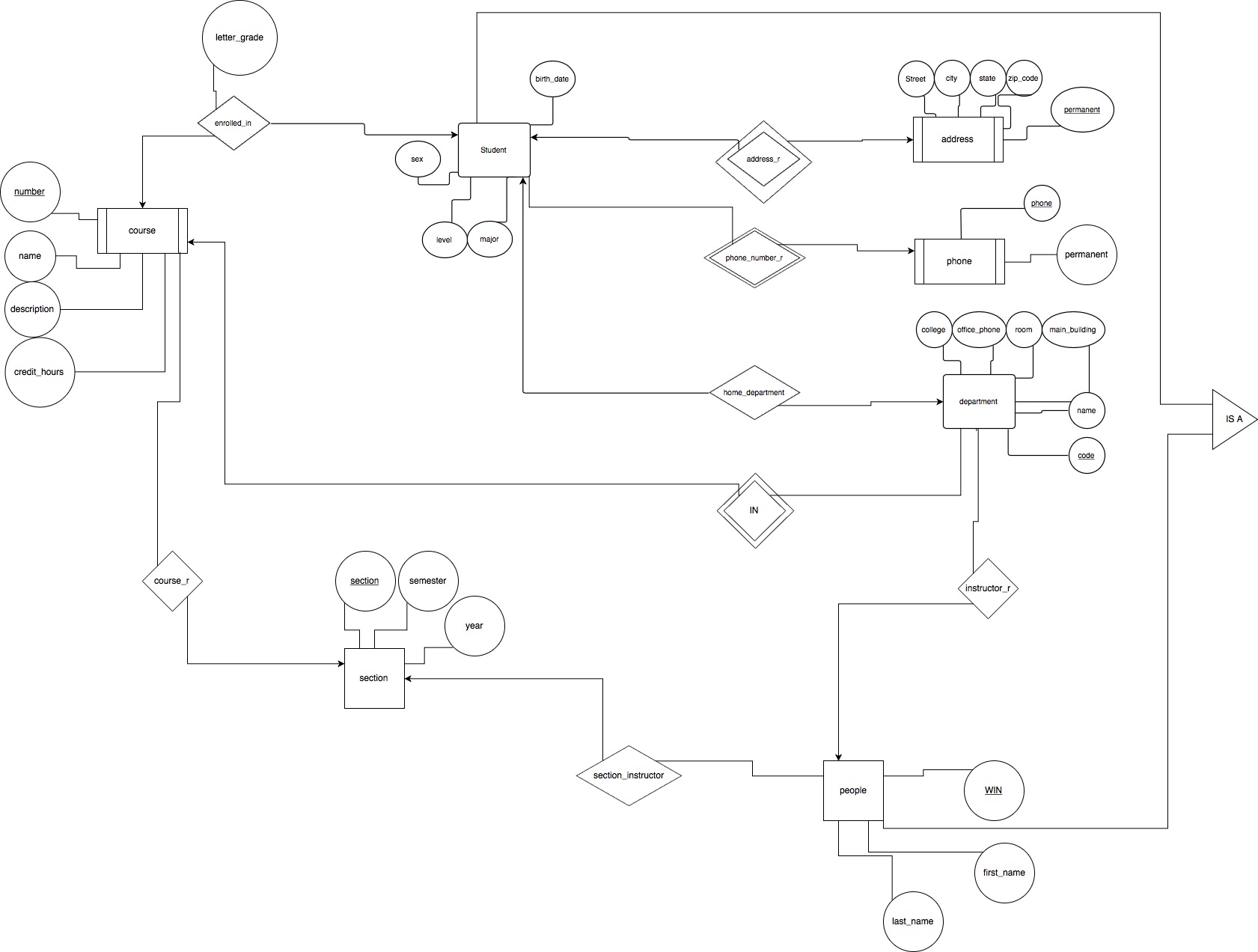
**Part 1 (20 points): Database Design**

**\*\* I understand the arrows can be confusing, so I wrote a description of the relations \*\***



**ER Diagram Guide**

\*\*Starting with student and moving clockwise around the diagram\*\*

address\_r (Student to Address)

many-to-many

1. Student must have one permanent address and one non-permanent address

2. An address could have multiple students at it

phone\_number\_r (Student to Phone)

one to many

1. One student can have many phones

2. A phone can only be used by one student

home\_department (Student to department)

many to many

1. many students to one department

2. many departments for a student (double major)

Student IS A people

instructor\_r (department to people)

many to one

1. a department can have many instructors

2. an instructor can only work in one department

section\_instruct (people to section)

many to one

1. an instructor can teach multiple sections

2. a section only has one instructor

course\_r (section to course)

many to one

1. a section can only have one course

2. a course can have many sections

course IN department

many to many

1. many courses in one department

2. the same course id can be in many departments

enrolled\_in (course to student)

many to many

1. a course can have many students

2. a student can be in many courses

**Convert the E-R diagram to a relational schema.**

CREATE TABLE people (

WIN int,

f\_name varchar(100),

l\_name varchar(100),

PRIMARY KEY(WIN)

);

CREATE TABLE student (

WIN int,

birthdate date,

sex bit,

level varchar(100),

major varchar(4),

department varchar(4),

PRIMARY KEY(WIN),

FOREIGN KEY(department) REFERENCES department(code),

FOREIGN KEY(WIN) REFERENCES people(WIN)

);

CREATE TABLE address (

WIN int,

street varchar(100),

city varchar(100),

state varchar(2),

zipcode int(9),

permanent bit,

PRIMARY KEY(WIN, permanent),

FOREIGN KEY(WIN) REFERENCES student(WIN)

);

CREATE TABLE phone (

phone varchar(20),

permanent bit,

WIN int,

PRIMARY KEY(WIN, permanent),

FOREIGN KEY(WIN) REFERENCES student(WIN)

);

CREATE TABLE department (

college varchar(100),

office\_phone varchar(100),

room varchar(20),

main\_building varchar(100),

name varchar(100),

code varchar(4),

PRIMARY KEY(code)

);

CREATE TABLE instructor (

WIN int,

department varchar(4),

PRIMARY KEY(WIN),

FOREIGN KEY(department) REFERENCES department(code),

FOREIGN KEY(WIN) REFERENCES people(WIN)

);

CREATE TABLE course (

number int,

name varchar(100),

description varchar(300),

credit int(1),

department varchar(4),

PRIMARY KEY(number, department),

FOREIGN KEY(department) REFERENCES department(code),

);

CREATE TABLE enrolled\_in (

WIN int,

grade varchar(2),

course\_number int,

PRIMARY KEY(WIN, course\_number),

FOREIGN KEY(WIN) REFERENCES student(WIN),

FOREIGN KEY(course\_number) REFERENCES course(number)

);

CREATE TABLE section (

section int,

semester varchar(10),

year int(4),

course int,

instructor int,

PRIMARY KEY(section),

FOREIGN KEY(course) REFERENCES course(number),

FOREIGN KEY(instructor) REFERENCES instructor(WIN)

);

CREATE TABLE address\_r (

WIN int,

permanent bit,

address char(20),

PRIMARY KEY(WIN, permanent),

FOREIGN KEY(WIN) REFERENCES student(WIN)

);

CREATE TABLE phone\_number\_r (

WIN int,

permanent bit,

phone char(20),

PRIMARY KEY(WIN, permanent),

FOREIGN KEY(WIN) REFERENCES student(WIN)

);

CREATE TABLE home\_department(

WIN int,

department char(4),

PRIMARY KEY(WIN, department),

FOREIGN KEY(WIN) REFERENCES student(WIN),

FOREIGN KEY(department) REFERENCES department(code)

);

CREATE TABLE course\_department (

course int,

department char(4),

PRIMARY KEY(course, department)

FOREIGN KEY(course) REFERENCES course(number),

FOREIGN KEY(department) REFERENCES department(code)

);

CREATE TABLE instruct\_r (

WIN int,

department char(4),

PRIMARY KEY(WIN),

FOREIGN KEY(WIN) REFERENCES people(WIN)

);

CREATE TABLE section\_instructor (

WIN int,

section int,

PRIMARY KEY(section),

FOREIGN KEY(section) REFERENCES section(section),

FOREIGN KEY(WIN) REFERENCES people(WIN)

);

CREATE TABLE course\_r (

section int,

course int,

PRIMARY KEY(section),

FOREIGN KEY(section) REFERENCES section(section)

);

**Part 2 (10 points): Relational schema normalization**

1. C -> D, C -> A, B -> C

candidate key - B

Normal Form - 2 NF

R1(B, C) and R2(C, D, A)

1. B -> C, D -> A

candidate key - B and D

Normal Form - 1 NF

R1(B, C) and R2(D, A)

1. ABC -> B, BC -> D, A -> C

candidate key - B, C, and D

Normal Form - 3 NF

No BCNF decomposition

1. A -> B, BC -> D, A -> C

candidate key - A

Normal Form - 2 NF

R1(A, B, C) and R2(BC, D)

1. AB -> C, AB -> D, C -> A, D -> B

candidate key – AB, CD, BC

Normal Form - 1 NF

R1(AB, C, D)