Project:
Code 187

Group Members:
Colt Wilson colt.wilson@sjsu.edu
Bedrya Balema bedrya11@gmail.com
Justin Passanisi passanisijd@yahoo.com
Bryan Garcia bryangarcia831@gmail.com

1) **DROP TABLE IF EXISTS** KillsIn, Kills, GoesTo, Contain, Victim, SerialKiller, Person, Trial, State, Country;

```
CREATE TABLE Country (
        name VARCHAR(52) NOT NULL,
        population INT UNSIGNED NOT NULL,
        PRIMARY KEY (name)
);

CREATE TABLE State (
        name VARCHAR(85) NOT NULL,
        population INT UNSIGNED NOT NULL,
        PRIMARY KEY (name)
);

CREATE TABLE Trial (
        name VARCHAR(75) NOT NULL,
        dateOfTrial DATE NOT NULL,
        defenseName VARCHAR(100),
        prosecutorName VARCHAR(100),
        verdict ENUM('Guilty','NotGuilty','NoContest','Unknown') NOT NULL DEFAULT
'Unknown',
        PRIMARY KEY (name)
);

CREATE TABLE Person (
        name VARCHAR(100) NOT NULL,
        dateOfBirth DATE NOT NULL,
        race VARCHAR(25),
        gender ENUM('M','F','O','Unknown') NOT NULL DEFAULT 'Unknown',
        PRIMARY KEY (name, dateOfBirth)
);

CREATE TABLE SerialKiller (
        name VARCHAR(100) NOT NULL,
        dateOfBirth DATE NOT NULL,
        currentStatus ENUM('NotCaught','Dead','Caught','Unknown') NOT NULL DEFAULT
'Unknown',
        FOREIGN KEY (name, dateOfBirth) REFERENCES Person(name, dateOfBirth)
```

```
        ON DELETE RESTRICT ON UPDATE RESTRICT
);

CREATE TABLE Victim (
        name VARCHAR(100) NOT NULL,
        dateOfBirth DATE NOT NULL,
        dateOfDeath DATE,
        typeOfDeath VARCHAR(100),
        FOREIGN KEY (name, dateOfBirth) REFERENCES Person(name, dateOfBirth)
        ON DELETE RESTRICT ON UPDATE RESTRICT
);

CREATE TABLE Contain (
        countryName VARCHAR(52) NOT NULL,
        stateName VARCHAR(85) NOT NULL,
        FOREIGN KEY (countryName) REFERENCES Country(name)
        ON DELETE RESTRICT ON UPDATE RESTRICT,
        FOREIGN KEY (stateName) REFERENCES State(name)
        ON DELETE RESTRICT ON UPDATE RESTRICT
);

CREATE TABLE GoesTo (
        personName VARCHAR(100) NOT NULL,
        personDateOfBirth DATE NOT NULL,
        trialName VARCHAR(75) NOT NULL,
        FOREIGN KEY (personName, personDateOfBirth ) REFERENCES
Person(name,dateOfBirth)
        ON DELETE RESTRICT ON UPDATE RESTRICT,
        FOREIGN KEY (trialName) REFERENCES Trial(name)
        ON DELETE RESTRICT ON UPDATE RESTRICT
);

CREATE TABLE Kills (
        skName VARCHAR(100) NOT NULL,
        skDateOfBirth DATE NOT NULL,
        victimName VARCHAR(100) NOT NULL,
        victimDateOfBirth DATE NOT NULL,
        FOREIGN KEY (skName, skDateofBirth) REFERENCES Person(name, dateOfBirth)
        ON DELETE RESTRICT ON UPDATE RESTRICT,
```

```
    FOREIGN KEY (victimName, victimDateOfBirth) REFERENCES Person(name,
dateOfBirth)
    ON DELETE RESTRICT ON UPDATE RESTRICT
);

CREATE TABLE KillsIn (
    stateName VARCHAR(85) NOT NULL,
    personName VARCHAR(100) NOT NULL,
    personDateOfBirth DATE NOT NULL,
    FOREIGN KEY (stateName) REFERENCES State(name)
    ON DELETE RESTRICT ON UPDATE RESTRICT,
    FOREIGN KEY (personName, personDateOfBirth) REFERENCES Person(name,
dateOfBirth)
    ON DELETE RESTRICT ON UPDATE RESTRICT
);
```

IF WE WANT TO INCLUDE/CASCADE UPDATE/DELETION OF PARENT TUPLE:

DROP TABLE IF EXISTS KillsIn, Kills, GoesTo, Contain, Victim, SerialKiller, Person, Trial, State, Country;

```
CREATE TABLE Country (
        name VARCHAR(52) NOT NULL,
        population INT UNSIGNED NOT NULL,
        PRIMARY KEY (name)
);

CREATE TABLE State (
        name VARCHAR(85) NOT NULL,
        population INT UNSIGNED NOT NULL,
        PRIMARY KEY (name)
);

CREATE TABLE Trial (
        name VARCHAR(75) NOT NULL,
        dateOfTrial DATE NOT NULL,
        defenseName VARCHAR(100),
        prosecutorName VARCHAR(100),
        verdict ENUM('Guilty','NotGuilty','NoContest','Unknown') NOT NULL DEFAULT
'Unknown',
        PRIMARY KEY (name)
);

CREATE TABLE Person (
        name VARCHAR(100) NOT NULL,
        dateOfBirth DATE NOT NULL,
        race VARCHAR(25),
        gender ENUM('M','F','O','Unknown') NOT NULL DEFAULT 'Unknown',
        PRIMARY KEY (name, dateOfBirth)
);

CREATE TABLE SerialKiller (
        name VARCHAR(100) NOT NULL,
        dateOfBirth DATE NOT NULL,
```

```
        currentStatus ENUM('NotCaught','Dead','Caught','Unknown') NOT NULL DEFAULT
'Unknown',
        FOREIGN KEY (name, dateOfBirth) REFERENCES Person(name, dateOfBirth)
        ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Victim (
        name VARCHAR(100) NOT NULL,
        dateOfBirth DATE NOT NULL,
        dateOfDeath DATE,
        typeOfDeath VARCHAR(100),
        FOREIGN KEY (name, dateOfBirth) REFERENCES Person(name, dateOfBirth)
        ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Contain (
        countryName VARCHAR(52) NOT NULL,
        stateName VARCHAR(85) NOT NULL,
        FOREIGN KEY (countryName) REFERENCES Country(name)
        ON DELETE CASCADE ON UPDATE CASCADE,
        FOREIGN KEY (stateName) REFERENCES State(name)
        ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE GoesTo (
        personName VARCHAR(100) NOT NULL,
        personDateOfBirth DATE NOT NULL,
        trialName VARCHAR(75) NOT NULL,
        FOREIGN KEY (personName, personDateOfBirth ) REFERENCES
Person(name,dateOfBirth)
        ON DELETE CASCADE ON UPDATE CASCADE,
        FOREIGN KEY (trialName) REFERENCES Trial(name)
        ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Kills (
        skName VARCHAR(100) NOT NULL,
        skDateOfBirth DATE NOT NULL,
        victimName VARCHAR(100) NOT NULL,
```

```
    victimDateOfBirth DATE NOT NULL,
    FOREIGN KEY (skName, skDateofBirth) REFERENCES Person(name, dateOfBirth)
    ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (victimName, victimDateOfBirth) REFERENCES Person(name,
dateOfBirth)
    ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE KillsIn (
    stateName VARCHAR(85) NOT NULL,
    personName VARCHAR(100) NOT NULL,
    personDateOfBirth DATE NOT NULL,
    FOREIGN KEY (stateName) REFERENCES State(name)
    ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (personName, personDateOfBirth) REFERENCES Person(name,
dateOfBirth)
    ON DELETE CASCADE ON UPDATE CASCADE
);
```

2) Attached

3) The belief is that our web-based database is powerful enough for the real world. If we wanted to create a more robust database for research or investigations, we would have to import a larger amount of data. Data would also need to be regulated and possibly include extra attributes that refer to patterns or trends that could link serial killers. Right now, our database is simply a fun tool for fans of serial killers. We believe that with the contribution of the public, and our implementation for the customer to insert and update queries into the database, our database would naturally broaden due to public input. With this broadening and progression of serial killer information, our database could become a hub for serial killer's, similar to the way Wikipedia is a hub for general information.

4) We added a "Insert a Serial Killer" feature on our webpage. This allows the user to insert into our database, a serial killer. The "Insert a Serial Killer" feature uses the "insert" function of MySQL. The user has to give full name, birthdate, ethnicity, gender and current status. Our PHP post method checks that the birthdate is in the correct format of 'YYYY-MM-DD'. An incorrect format returns the user an error.

Gender is also controlled by the MySQL database to make sure it is either M, F, O, or Unknown. The database also has a check to make sure that NotCaught, Dead, Caught, or Unknown is the current status of the serial killer to be added.

The last feature we added was a “Check if your name matches a serial killer’s name”. A user can put in any name and we will check our database to see if you have a serial killer’s name!

IMPORTANT TROUBLESHOOTING INFO

- Load the CSV's and make sure that the database you load into is named 'CS157A_Database'.

- Add the entire 'Code187' folder to your ‘../XAMPP/htdocs’ folder.

Colt Wilson:	Worked on #1’s implementation as well as fixing some php/mysql problems that we encountered.
Bryan Garcia:	HTML form to add a serial killer using textboxes and the php method that responds to it.
Bedrya Balema:	HTML coding of the webpage, CSS styling and testing.
Justin Passanisi:	Building and testing queries before and after webpage creation.