This document describes the relationships between entities involved in the implementation of a database for school system. The purpose of this database is to catalog and track student and teacher information. With this database we can accurately track any student or teacher partaking in school courses, activities, or correctional punishments. By tracking this information, students and parents are able to look up the standing of this child and see any activities the student is participating in or any violations of the code of conduct that the student has performed. School administrators will also be able to accurately track activities of staff members and assign them to contractual obligations as well as acknowledge any clubs/sports being taken on by staff members.

C:\Users\Alex\Dropbox\Junior S2\Pablo-Database Management\project\SchoolDiagramWhite.png

**People lists all people attributes**

CREATE TABLE people (

social integer not null unique,

fName text not null,

lName text not null,

addrID integer not null,

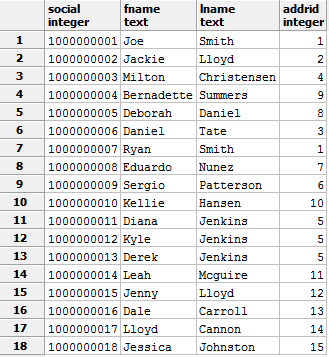
primary key(social),

foreign key(addrID) references address(addrID)

);

**Functional Dependencies**

Social ➔ fName, lName, addrID



**Address lists all addresses in the database**

CREATE TABLE address (

addrID integer not null unique,

strNum integer not null,

street text not null,

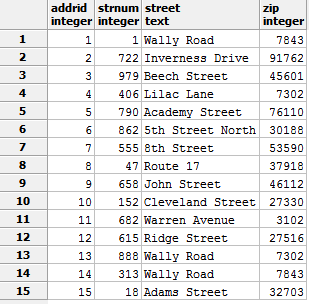
zip integer not null,

primary key(addrID)

);

**Functional Dependencies**

addrID ➔ strNum, street, zip



**Student lists all student attributes**

CREATE TABLE student (

social integer not null unique references people(social),

year text not null,

specNeed text,

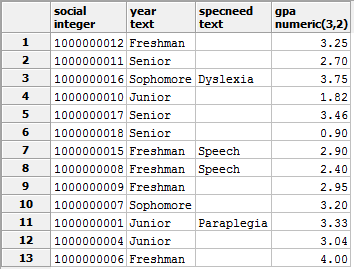
GPA float not null,

primary key(social)

);

**Functional Dependencies**

social ➔ year, specNeed, GPA



**Teacher lists all teacher attributes**

CREATE TABLE teacher (

social integer not null unique references people(social),

dept text not null,

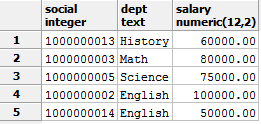
salary float not null,

primary key(social)

);

**Functional Dependencies**

Social ➔ dept, salary



**Class lists all courses and course related information**

CREATE TABLE class (

crsNum integer not null unique,

time time not null,

rmNum integer not null,

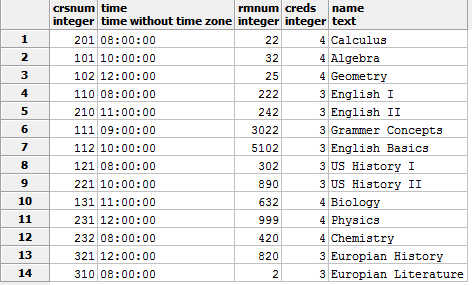
creds integer not null,

name text not null

);

**Functional Dependencies**

crsNum ➔ time, rmNum, creds, name



**Club lists all clubs and sports as well as the time they’re held**

CREATE TABLE club (

clbNum integer not null unique,

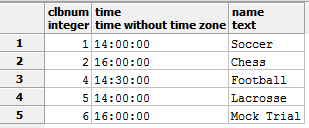
time time not null,

name text not null

);

**Functional Dependencies**

clbNum ➔ time, name



**Detention lists the date of detention and punishment given**

CREATE TABLE detention (

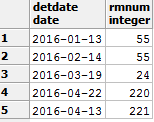
detDate date not null unique,

rmNum integer not null

);

**Functional Dependencies**

detDate ➔ rmNum



**Enrollment lists all students’ courses and grades**

CREATE TABLE enrollment (

social integer not null references student(social),

crsNum integer not null references class(crsNum),

grade integer not null

);

**Functional Dependencies**

(social, crsNum) ➔ grade

**Partic lists all students participating in a club or sport and their position/role**

CREATE TABLE partic (

social integer not null references student(social),

clbNum integer not null references club(clbNum),

position text not null

);

**Functional Dependencies**

(social, clbNum) ➔ position

**Infrac lists all students who violated school rules as well as a description of violation**

CREATE TABLE infrac (

social integer not null references student(social),

detDate date not null references detention(detDate),

blurb text not null

);

**Functional Dependencies**

(social, detDate) ➔ blurb

**Section lists all teachers’ courses**

CREATE TABLE section (

social integer not null references teacher(social),

crsNum integer not null references class(crsNum)

);

**Functional Dependencies**

None

**Coach lists all coaches and their teams**

CREATE TABLE coach (

social integer not null references teacher(social),

clbNum integer not null references club(clbNum)

);

**Functional Dependencies**

None

**Duty lists when teachers are assigned detention duty**

CREATE TABLE duty (

social integer not null references teacher(social),

detDate date not null references detention(detDate)

);

**Functional Dependencies**

None