|  |
| --- |
| Oregon state university |
| CS340 Project |
| Community Garden Management System |
|  |
| **Chavis Sabin and Adina Edwards** |
| **5/31/2016** |

|  |
| --- |
|  |

Contents

[Outline 1](#_Toc452482729)

[Database Outline Description 2](#_Toc452482730)

[ER Diagram 3](#_Toc452482731)

[Database Schema 4](#_Toc452482732)

[Table Creation Queries 4](#_Toc452482733)

[Add Queries 7](#_Toc452482734)

[Delete Queries 8](#_Toc452482735)

[Update Queries 8](#_Toc452482736)

[Search Query 8](#_Toc452482737)

[Picture of Unicorn **Error! Bookmark not defined.**](#_Toc452482738)

# Outline

For our CS 340 project we will build a database for managing community gardens. The database will store data for the community gardens around townas well as the individual plots that make up each community garden, who is assigned to each plot, and what plants they contain. The main purpose for creating this database is so one individual can manage all community gardens within a giving area.

The database can act as a contact list in case the owner of a plot needs to be contacted for an emergency. It can also track how many types of plants are grown in the city. Depending on the impact the community gardens have on the area, the data can be used to organize local exchanges of vegetables between community gardeners or to set up a farmers market using vegetables grown and the community gardens.

# Database Outline Description

As a team, Chavis Sabin and Adina Edwards propose to design and build a database which holds information about community gardens. Entity tables can include:

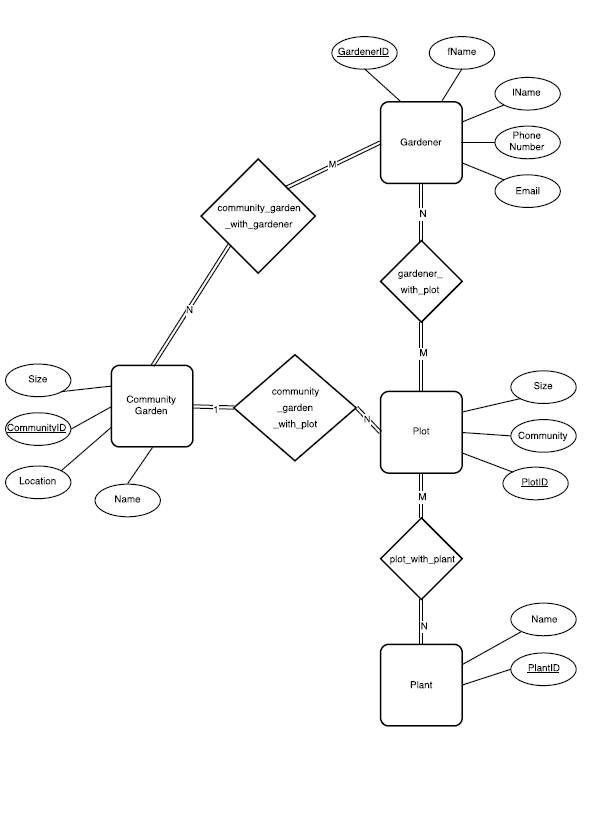
1. Community garden
2. Garden plot
3. Gardener
4. Plant

These entities include different types of relationships:

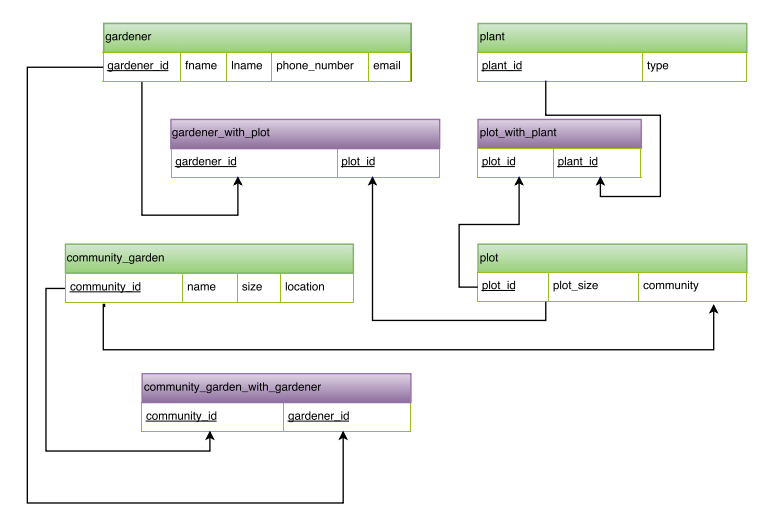
Rules for each entity include:

1. One community garden is composed of two or more plots. This is a one-to-many relationship.
   1. Community Garden has at least one or more plot.
   2. A plot has exactly one Community Garden
2. Multiple gardeners can work on a single plot. For example, a family can tend a single plot together. This is also a many-to-one relationship.
   1. Plots must have 1 or more gardeners
   2. Gardeners have at least 1 plot
3. A gardener can have plots in multiple community gardens and a single community garden can have many gardeners. This is a many-to-many relationship.
   1. A Community Garden has 1 or more Gardeners
   2. A Gardener has at least 1 or more Community Garden
4. A single plot can have multiple types of plants, and a single type of plant can be grown in many different plots. This is many-to-many relationship.
   1. A Plot must have 1 or more plants
   2. A Plant has at least 1 plot

# ER Diagram



# Database Schema



# Table Creation Queries

1. Gardener

-- gardener:

-- gardener\_id - an auto incrementing integer which is the primary key, cannot be null

-- fname - a varchar with a maximum length of 255 characters, cannot be null

-- lname - a varchar with a maximum length of 255 characters, cannot be null

-- phone\_number - integer

-- email - a varchar with a maximum length of 255 characters

CREATE TABLE gardener(

gardener\_id int(11) NOT NULL AUTO\_INCREMENT,

fname varchar(255) NOT NULL,

lname varchar(255) NOT NULL,

phone\_number int(11),

email varchar(255),

PRIMARY KEY(gardener\_id)

)ENGINE=InnoDB;

2. Plant

-- plant:

-- plant\_id - an auto incrementing integer which is the primary key, cannot be null

-- type - a varchar of maximum length 255, cannot be null

-- TODO: should this table actually be "plant\_type" and we have another table that is "plant"?

-- Alternatively, we could chance "type" to "name" and have a table of plant names

CREATE TABLE plant(

plant\_id int(11) NOT NULL AUTO\_INCREMENT,

name varchar(255) NOT NULL,

PRIMARY KEY(plant\_id)

)ENGINE=InnoDB;

3. Community Garden

-- community\_garden

-- community\_id - an auto incrementing integer which is the primary key, cannot be null

-- size - an integer

-- location - a varchar of maximum length 255

CREATE TABLE community\_garden(

community\_id int(11) NOT NULL AUTO\_INCREMENT,

name varchar(255) NOT NULL,

size int(11),

location varchar(255) NOT NULL,

PRIMARY KEY (community\_id)

)ENGINE=InnoDB;

4. Plot

-- plot

-- plot\_id - an auto incrementing integer which is the primary key, cannot be null

-- plot\_size - an integer

-- community - a varchar of maximum length 255, unique key, cannot be null, reference to community\_garden

CREATE TABLE plot(

plot\_id int(11) NOT NULL AUTO\_INCREMENT,

plot\_size int(11),

community int(11),

PRIMARY KEY(plot\_id),

FOREIGN KEY (community) REFERENCES community\_garden(community\_id)

ON DELETE SET NULL

ON UPDATE CASCADE

)ENGINE=InnoDB;

5. Gardener with plot

-- gardener\_with\_plot:

-- gardener\_id - an integer references gardener

-- plot\_id - an integer references plot

-- The primary key is a combination of gardener\_id and plot\_id

CREATE TABLE gardener\_with\_plot(

gardener\_id int(11),

plot\_id int(11),

FOREIGN KEY (plot\_id) REFERENCES plot(plot\_id)

ON DELETE CASCADE,

FOREIGN KEY (gardener\_id) REFERENCES gardener(gardener\_id)

ON DELETE CASCADE,

PRIMARY KEY (gardener\_id,plot\_id)

)ENGINE=InnoDB;

6. Plot with Plant

-- plot\_with\_plant:

-- plot\_id - an integer references plot

-- plant\_id - an integer references plant

-- The primary key is a combination of plot\_id and plant\_id

CREATE TABLE plot\_with\_plant(

plot\_id int(11),

plant\_id int(11),

FOREIGN KEY (plot\_id) REFERENCES plot(plot\_id)

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY (plant\_id) REFERENCES plant(plant\_id)

ON DELETE CASCADE

ON UPDATE CASCADE,

PRIMARY KEY (plot\_id,plant\_id)

)ENGINE=InnoDB;

6. Community Garden with Gardener

-- community\_garden\_with\_gardener:

-- community\_id - an integer references community\_garden

-- gardener\_id - an integer references gardener

-- The primary key is a combination of community\_id and gardener\_id

CREATE TABLE community\_garden\_with\_gardener(

community\_id int(11),

gardener\_id int(11),

FOREIGN KEY (community\_id) REFERENCES community\_garden(community\_id)

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY (gardener\_id) REFERENCES gardener(gardener\_id)

ON DELETE CASCADE

ON UPDATE CASCADE,

PRIMARY KEY (community\_id,gardener\_id)

)ENGINE=InnoDB;

# Add Queries

1. Add data to community\_garden

INSERT INTO community\_garden(name, size, location) values ([name], [size], [location]);

For Example:

INSERT INTO community\_garden(name, size, location) values

("Beautiful Strolls", "75", "Tualatin Hills");

1. Add data to gardener

INSERT INTO gardener(fname, lname, phone\_number, email) values ([fname], [lname], [phone\_number], [email]);

For Example:

INSERT INTO gardener(fname, lname, phone\_number, email) values

("Nyota","Uhura","3238611234","nuhura@enterprise.fed");

1. Add data to plot

INSERT INTO plot(plot\_size, community) values ([plot\_size], [community\_id]);

For Example:

INSERT INTO plot(plot\_size, community) values

("10", (SELECT community\_id FROM community\_garden WHERE location = "Wilsonville"));

1. Add data to plant

INSERT INTO plant(name) values ([name]);

For Example

INSERT INTO plant(name) values ("Snow Pea"), ("Butter Head Lettuce"), ("Strawberry");

1. Add data to gardener\_with\_plot

INSERT INTO gardener\_with\_plot(gardener\_id, plot\_id) values ([gardener\_id],[plot\_id]);

For Example:

INSERT INTO gardener\_with\_plot(gardener\_id, plot\_id) values

((SELECT g.gardener\_id FROM gardener g WHERE g.fname = "Jean-Luc" AND g.lname = "Picard"), "1");

1. Add data to plot\_with\_plant

INSERT INTO plot\_with\_plant(plot\_id, plant\_id) values ([plot\_id], [plant\_id]);

For Example:

INSERT INTO plot\_with\_plant(plot\_id, plant\_id) values

(“1”, (SELECT plant\_id FROM plant WHERE name = “Snow Pea”);

1. Add data to community\_garden\_with\_gardener

INSERT INTO community\_garden\_with\_gardener(community\_id, gardener\_id) values

([community\_id],[ gardener\_id]);

For Example:

INSERT INTO community\_garden\_with\_gardener(community\_id, gardener\_id) values

((SELECT community\_id FROM community\_garden WHERE location = "Wilsonville"),(SELECT gardener\_id FROM gardener WHERE fname = "Jean-Luc" AND lname = "Picard"));

# Delete Queries

DELETE FROM {$table} WHERE {$table\_id} = {$id}

# Update Queries

UPDATE plant SET name='{$plant\_name[$n]}' WHERE plant\_id='{$plant\_id[$n]}';

# Search Query

SELECT cg.name, cg.location FROM community\_garden cg

INNER JOIN community\_garden\_with\_gardener cgwg ON cg.community\_id = cgwg.community\_id

INNER JOIN gardener g ON cgwg.gardener\_id = g.gardener\_id

INNER JOIN gardener\_with\_plot gwp ON g.gardener\_id = gwp.gardener\_id

INNER JOIN plot po ON gwp.plot\_id = po.plot\_id

INNER JOIN plot\_with\_plant pwp ON po.plot\_id = pwp.plot\_id

INNER JOIN plant pa ON pwp.plant\_id = pa.plant\_id

WHERE pa.name = '{$query}'

ORDER BY cg.location;