OREGON STATE UNIVERSITY

CS340 Project

Community Garden Management System

Chavis Sabin and Adina Edwards

5/31/2016

Contents

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 town as 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:

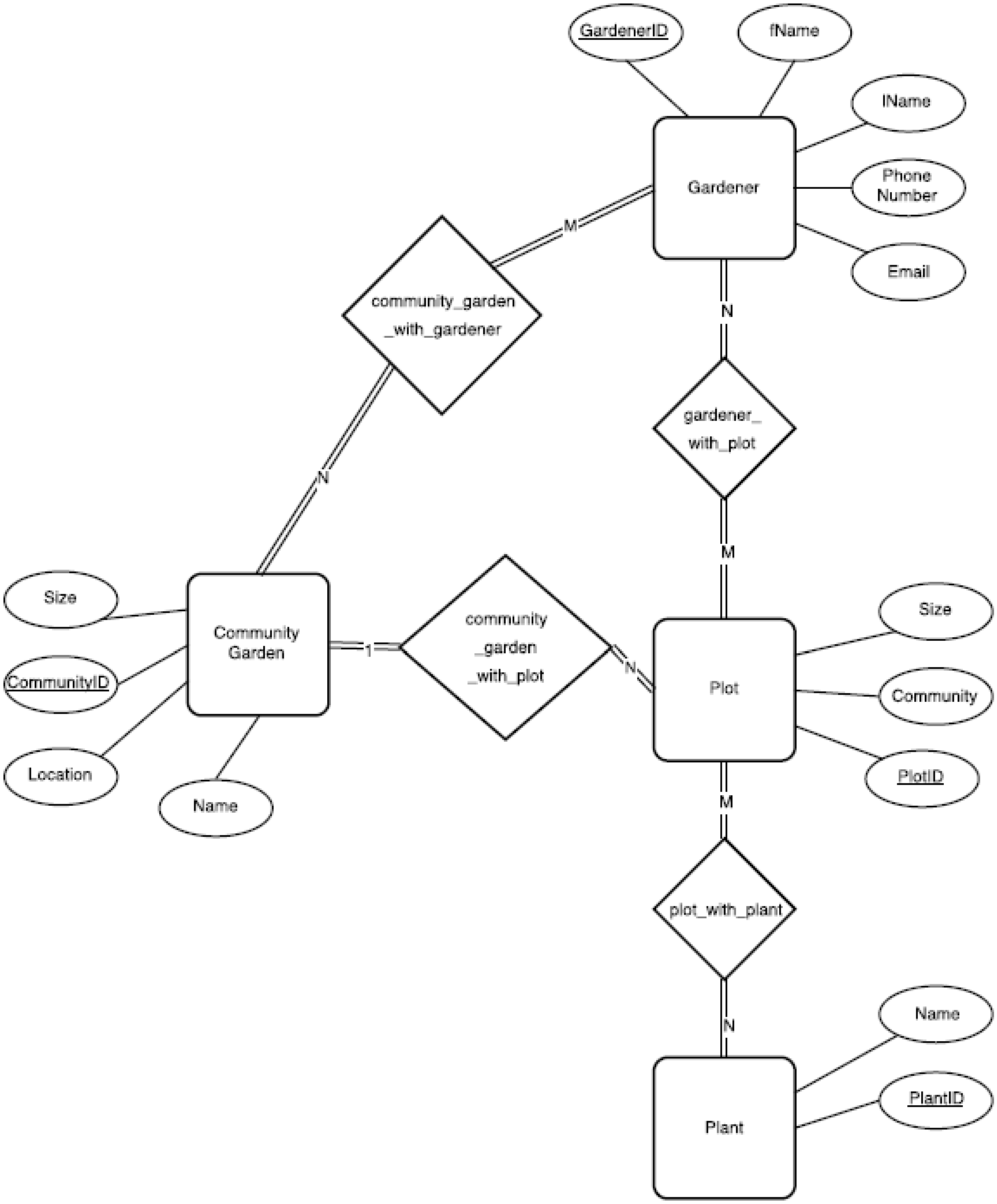
1. One community garden is composed of two or more plots. This is a one-to-many relationship.
   a. Community Garden has at least one or more plot.
   b. A plot has exactly one Community Garden

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.
   a. Plots must have 1 or more gardeners
   b. 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.
   a. A Community Garden has 1 or more Gardeners
   b. 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.
   a. A Plot must have 1 or more plants
   b. A Plant has at least 1 plot

ER Diagram


GardenerID
fName
lName
Gardener
Phone Number
Email
M
community_garden _with_gardener
N
gardener_ with_plot
N
M
Size
CommunityID
Location
Community Garden
Name
1
community _garden _with_plot
N
Plot
Size
Community
PlotID
M
plot_with_plant
N
Name
PlantID
Plant

Database Schema

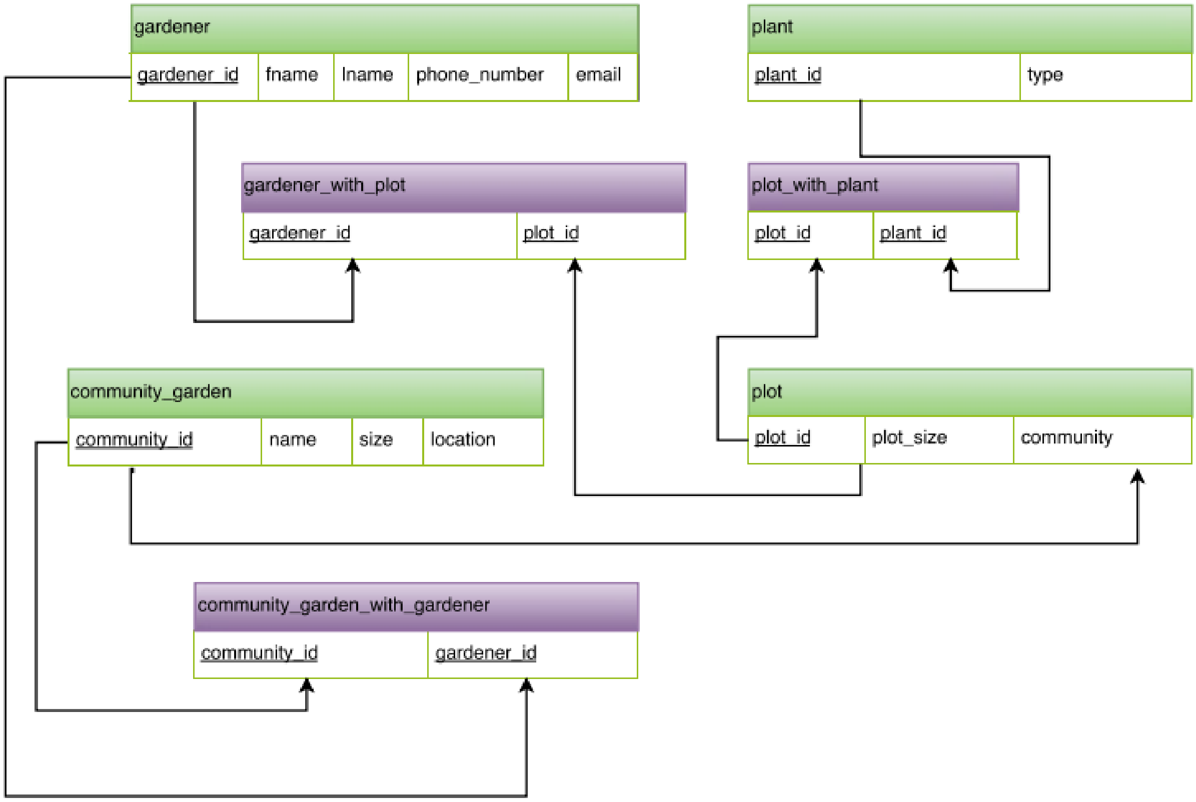


Table Creation Queries

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");
   ```

2. 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");
   ```

3. 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"));
   ```

4. Add data to plant

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

   For Example

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

5. 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");
```

6. 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”);
   ```

7. 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;