University of British Columbia, Vancouver

Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: Milestone 2

Date: July 16, 2024

Group Number: Project Group 3

Name	Student Number	CS Alias (userid)	Preferred E-mail Address
Jeffrey Ho	26034066	g1k2b	jho1211@student.ubc.ca
Michelle Lei	11357167	j1r2b	michellejslei@gmail.com
Frederick Sunstrum	42266379	k0l3e	fr.sunstrum@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

University of British Columbia, Vancouver

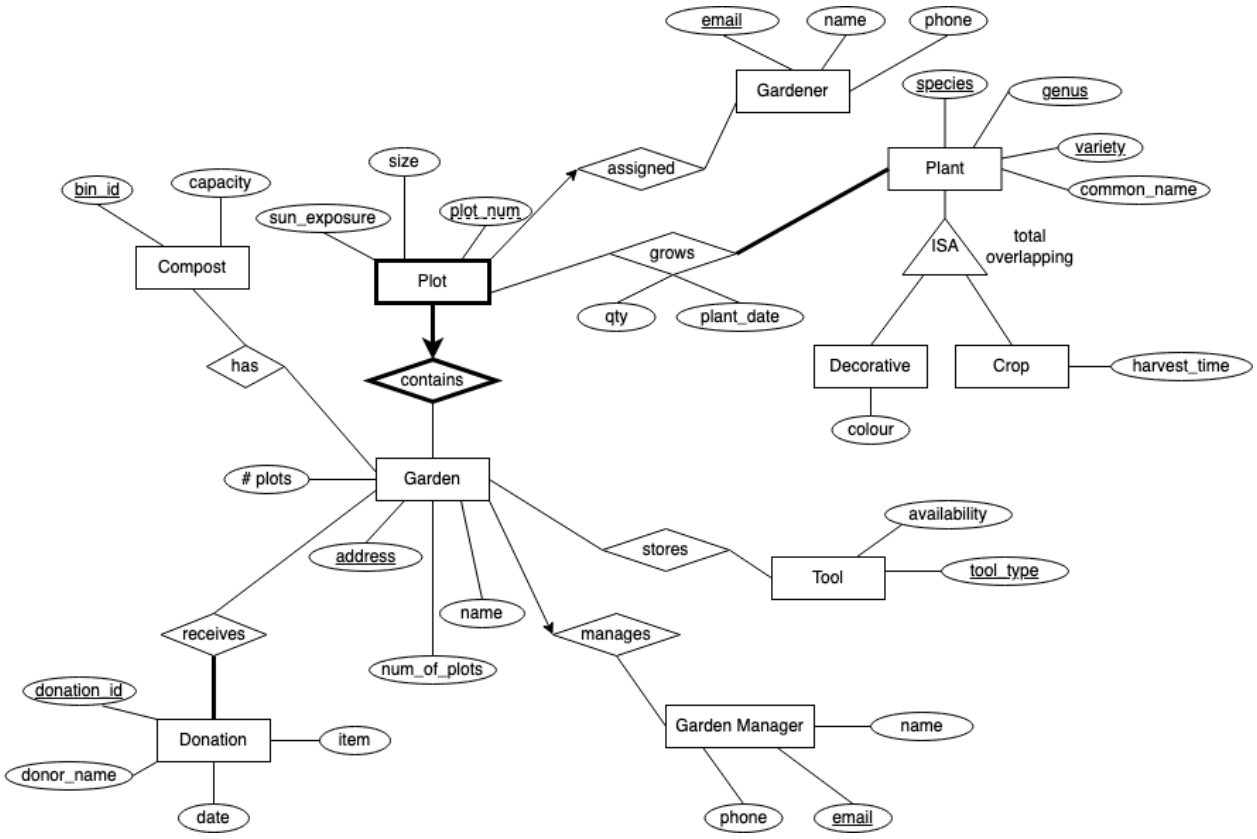
Department of Computer Science

Project Summary

This database management system will be used by a municipality to store information about various community gardens throughout the city so that they can better organize these community gardens. Each garden will be managed by different community organizations and they will record the plants that are seeded in each plot and the gardeners who are assigned to each plot. There will also be donations of tools and plants which are recorded so that the garden members will know what is available for use.

ER Diagram

The ER diagram was changed from the original one sent in Milestone 1 based on the feedback received. The ISA constraints were changed from (total, disjoint) to (total, overlapping). Also, the ISA hierarchy was changed from Species -> (Plant and Seed) to Plant -> (Decorative and Crop) entities.



University of British Columbia, Vancouver

Department of Computer Science

Schema

The schema derived from your ER diagram (above). For the translation of the ER diagram to the relational model, follow the same instructions as in your lectures. The process should be reasonably straightforward. For each table:
a. List the table definition (e.g., Table1(attr1: domain1, attr2: domain2, ...)). Make sure to include the domains for each attribute.
b. Specify the primary key (PK), candidate key, (CK) foreign keys (FK), and other constraints (e.g., not null, unique, etc.) that the table must maintain.

PRIMARY KEY
FOREIGN KEY

```
Receives(donation_id: INTEGER, garden_address: VARCHAR(20))

Donation(donation_id: INTEGER,
         donor_name: VARCHAR(20),
         date: DATE,
         item: VARCHAR(20))
   • Candidate key: {donor_name, date, item}

Stores(garden_address: VARCHAR(20), tool_type: VARCHAR(20))

Tool(tool_type: VARCHAR(20), availability: CHAR(1))

GardenManager(name: VARCHAR(20), phone: CHAR(12), email: VARCHAR(20))
   • Candidate key: {name, phone}
   • name is NOT NULL

GardenManages(address: VARCHAR(20),
              num_of_plots: INTEGER,
              garden_name: VARCHAR(20),
              manager_email: VARCHAR(20))
   • garden_name is UNIQUE and NOT NULL

Gardener(email: VARCHAR(20), phone: CHAR(12), name: VARCHAR(20))
   • Candidate key: {name, phone}
   • name is NOT NULL

GardenerPlot(garden_address: VARCHAR(20),
```

University of British Columbia, Vancouver

Department of Computer Science

```
            gardener_email: VARCHAR(20),
            plot_num: INTEGER,
            sun_exposure: VARCHAR(20),
            plot_size: INTEGER)

Compost(bin_id: INTEGER, capacity: INTEGER)

HasCompost(bin_id: INTEGER, garden_address: VARCHAR(20))

Plant(species: VARCHAR(20),
      genus: VARCHAR(20),
      variety: VARCHAR(20),
      common_name: VARCHAR(20),
      colour: VARCHAR(20),
      harvest_time: INTEGER)
   • common_name is NOT NULL

Grows(species: VARCHAR(20),
      genus: VARCHAR(20),
      variety: VARCHAR(20),
      plot_num: INTEGER,
      garden_address: VARCHAR(20),
      qty: INTEGER,
      plant_date: DATE)
```

Functional Dependencies

PKs and CKs are considered functional dependencies and should be included in the list of FDs. You do not need to include trivial FDs such as A→A.

Note: In your list of FDs, there must be some kind of valid FD other those identified by a PK or CK. If you observe that no relations have FDs other than the PK and CK(s), then you will have to intentionally add some (meaningful) attributes to show valid FDs. We want you to get a good normalization exercise. Your design must go through a normalization process. You do not need to have a non-PK/CK FD for each relation but be reasonable. If your TA feels that some non-PK/CK FDs have been omitted, your grade will be adjusted accordingly.

Receives

- There are no FDs

Donation

- donation_id → donor_name, date, item
- donor_name, date, item → donation_id

Stores

- There are no FDs

Tool

- tool_type → availability

GardenManager

- email → name, phone
- name, phone → email
- phone → email, name

GardenManages

- address → garden_name, num_of_plots
- garden_name → manager_email
- garden_name, num_of_plots → manager_email, address

GardenerPlot

- garden_address, plot_num → gardener_email, sun_exposure, plot_size
- garden_address, gardener_email → plot_num, sun_exposure, plot_size
- gardener_email → garden_address
- gardener_email, plot_num → garden_address, plot_size, sun_exposure

Gardener

- gardener_email → gardener_phone, gardener_name
- gardener_phone, gardener_name → gardener_email

Compost

- bin_id → capacity

HasCompost

- There are no FDs

Plant

- species, genus, variety → common_name, colour, harvest_time
- common_name → colour, harvest_time

Grows

- garden_address, plot_num, species, genus, variety → qty, plant_date

University of British Columbia, Vancouver

Department of Computer Science

Normalization

Legend:
PRIMARY KEY
FOREIGN KEY

Receives is in 3NF - no change
`Receives(donation_id: INTEGER, garden_address: VARCHAR(20))`
Donation is in 3NF - no change

`Donation(donation_id: INTEGER,`
`donor_name: VARCHAR(20),`
`date: DATE,`
`item: VARCHAR(20))`
• Candidate key: {donor_name, date, item}

Stores is in 3NF - no change
`Stores(garden_address: VARCHAR(20), tool_type: VARCHAR(20))`
Tool is in 3NF - no change
`Tool(tool_type: VARCHAR(20), availability: CHAR(1))`
GardenManger is in 3NF - no change
`GardenManager(name: VARCHAR(20), phone: CHAR(12), email: VARCHAR(20))` • Candidate key: {name, phone} • name is NOT NULL
GardenManages is not in 3NF (violated by garden_name → manager_email)
• address → garden_name, num_of_plots • garden_name → manager_email • garden_name, num_of_plots → manager_email, address The LHS of the FD garden_name → manager_email, garden_name doesn't form a superkey so it doesn't satisfy BCNF. Also, it doesn't satisfy 3NF because manager_email is not part of a key.

Left	Middle	Right
garden_name	address, num_of_plots	manager_email

The candidate keys are: {address} and {garden_name, num_of_plots}

Find the minimal cover:
Step 1: Standard Form

(1) address → garden_name
(2) address → num_of_plots

(3) garden_name → manager_email

(4) garden_name, num_of_plots → manager_email
(5) garden_name, num_of_plots → address

Step 2: Minimize LHS of each FD
FD (4) can be minimized because garden_name alone will give you manager_email.
Therefore, we have:

(1) address → garden_name
(2) address → num_of_plots

(3) garden_name → manager_email

(4) garden_name → manager_email
(5) garden_name, num_of_plots → address

Step 3: Delete redundant FDs
(3) and (4) are the same, so we can remove (4) to yield the minimal cover:

address → garden_name
address → num_of_plots
garden_name → manager_email
garden_name, num_of_plots → address

Decompose into 3NF using synthesis:

R1(address, garden_name)
R2(address, num_of_plots)
R3(garden_name, manager_email)

University of British Columbia, Vancouver

Department of Computer Science

R4(garden_name, num_of_plots, address)

The initial decomposition includes a key for R, so we do not need to add the key separately. R1 is a subset of R4, so we can remove R1 to avoid redundancy. Therefore, we are left with the final relations:

GardenNumPlots(address, num_of_plots)
GardenManages(**garden_name, manager_email**)
GardenInfo(address, garden_name, num_of_plots)

Gardener is in 3NF - no change

Gardener(email: VARCHAR(20), phone: CHAR(12), name: VARCHAR(20))

- Candidate key: {name, phone}
- name is NOT NULL

GardenerPlot is in 3NF - no change

```
GardenerPlot(garden_address: VARCHAR(20),
             gardener_email: VARCHAR(20),
             plot_num: INTEGER,
             sun_exposure: VARCHAR(20),
             plot_size: INTEGER)
```

Compost is in 3NF - no change

Compost(bin_id: INTEGER, capacity: INTEGER)

HasCompost is in 3NF - no change

HasCompost(**bin_id**: INTEGER, **garden_address**: VARCHAR(20))

Plant is not in 3NF (violated by common_name → colour, harvest_time)

- species, genus, variety → common_name, colour, harvest_time
- common_name → colour, harvest_time

The LHS of the FD common_name → colour, harvest_time doesn't form a superkey so it doesn't satisfy BCNF. Also, it doesn't satisfy 3NF because colour and harvest_time are not part of a key.

Left	**Middle**	**Right**

species, genus, variety	common_name	colour, harvest_time

The candidate key is {species, genus, variety}

Find the minimal cover:
Step 1: Standard Form

(1) species, genus, variety → common_name
(2) species, genus, variety → colour
(3) species, genus, variety → harvest_time

(4) common_name → colour
(5) common_name → harvest_time

Step 2: Minimize LHS of each FD
No LHS of any FDs can be minimized

Step 3: Delete redundant FDs
(2) can be removed since (4) yields colour and (1) yields common_name. Similarly, (3) can be removed since (5) yields harvest_time and (1) yields common_name. Deletion of redundant FDs yield the minimum cover:

species, genus, variety → common_name
common_name → colour
common_name → harvest_time

Decompose into 3NF using synthesis:

R1(species, genus, variety, common_name)
R2(common_name, colour)
R3(common_name, harvest_time)

The initial decomposition includes a key for R, so we do not need to add the key separately. The final relations after decomposition into 3NF are as follows:

```
PlantInfo(species, genus, variety, common_name)
   ● common_name is NOT NULL
PlantColour(common_name, colour)
PlantHarvest(common_name, harvest_time)
```

Grows is in 3NF - no change

University of British Columbia, Vancouver

Department of Computer Science

```
Grows(species: VARCHAR(20),
      genus: VARCHAR(20),
      variety: VARCHAR(20),
      plot_num: INTEGER,
      garden_address: VARCHAR(20),
      qty: INTEGER,
      plant_date: DATE)
```

University of British Columbia, Vancouver

Department of Computer Science

SQL DDL

The SQL DDL statements required to create all the tables from item #6. The statements should use the appropriate foreign keys, primary keys, UNIQUE constraints, etc.
Unless you know that you will always have exactly x characters for a given character, it is better to use the VARCHAR data type as opposed to a CHAR(Y). For example, UBC courses always use four characters to represent which department offers a course. In that case, you will want to use CHAR(4) for the department attribute in your SQL DDL statement. If you are trying to represent the name of a UBC course, you will want to use VARCHAR as the number of characters in a course name can vary greatly.

```
CREATE TABLE Receives(
   donation_id INTEGER,
   garden_address VARCHAR(20),
   PRIMARY KEY (donation_id, garden_address),
   FOREIGN KEY (donation_id) REFERENCES Donation,
   FOREIGN KEY (garden_address) REFERENCES
      GardenNumPlots(address)
);

CREATE TABLE Donation(
   donation_id INTEGER PRIMARY KEY,
   donor_name VARCHAR(20),
   date DATE,
   item VARCHAR(20),
   UNIQUE (donor_name, date, item)
);


CREATE TABLE Stores(
   garden_address VARCHAR(20),
   tool_type VARCHAR(20),
   PRIMARY KEY (garden_address, tool_type),
   FOREIGN KEY (garden_address) REFERENCES
      GardenNumPlots(address),
   FOREIGN KEY (tool_type) REFERENCES Tool
);

CREATE TABLE Tool(
   tool_type VARCHAR(20) PRIMARY KEY,
```

```
    availability CHAR(1)
);

CREATE TABLE GardenManager(
    name VARCHAR(20) NOT NULL,
    phone CHAR(12),
    email VARCHAR(20) PRIMARY KEY,
    UNIQUE (name, phone)
);

CREATE TABLE GardenNumPlots(
    address VARCHAR(20) PRIMARY KEY,
    num_of_plots INTEGER
);

CREATE TABLE GardenInfo(
    address VARCHAR(20),
    garden_name VARCHAR(20) NOT NULL UNIQUE,
    num_of_plots INTEGER,
    PRIMARY KEY (garden_name, num_of_plots)
);

CREATE TABLE GardenManages(
    garden_name VARCHAR(20) PRIMARY KEY,
    manager_email VARCHAR(20),
    FOREIGN KEY (garden_name) REFERENCES GardenInfo,
    FOREIGN KEY (manager_email) REFERENCES
        GardenManager(email)
);

CREATE TABLE Gardener(
    email VARCHAR(20) PRIMARY KEY,
    phone CHAR(12),
    name VARCHAR(20) NOT NULL,
    UNIQUE (name, phone)
);

CREATE TABLE GardenerPlot(
    garden_address VARCHAR(20),
    gardener_email VARCHAR(20),
```

```
    plot_num INTEGER,
    sun_exposure VARCHAR(20),
    plot_size INTEGER,
    PRIMARY KEY (garden_address, plot_num),
    FOREIGN KEY (garden_address) REFERENCES
       GardenNumPlots(address)
       ON DELETE CASCADE
       ON UPDATE CASCADE,
    FOREIGN KEY (gardener_email) REFERENCES
       Gardener(email)
       ON DELETE SET NULL
       ON UPDATE CASCADE
);

CREATE TABLE Compost(
    bin_id INTEGER PRIMARY KEY,
    capacity INTEGER
);

CREATE TABLE HasCompost(
    bin_id INTEGER,
    garden_address VARCHAR(20),
    PRIMARY KEY (bin_id, garden_address),
    FOREIGN KEY (bin_id) REFERENCES Compost,
    FOREIGN KEY (garden_address) REFERENCES
       GardenNumPlotsManages(address)
);

CREATE TABLE PlantInfo(
    species VARCHAR(20),
    genus VARCHAR(20),
    variety VARCHAR(20),
    common_name VARCHAR(20) NOT NULL,
    PRIMARY KEY (species, genus, variety)
);

CREATE TABLE PlantColour(
    common_name VARCHAR(20) PRIMARY KEY,
    colour VARCHAR(20),
    FOREIGN KEY (common_name) REFERENCES PlantInfo
```

```
      ON DELETE CASCADE
      ON UPDATE CASCADE
);

CREATE TABLE PlantHarvest(
   common_name VARCHAR(20) PRIMARY KEY,
   harvest_time INTEGER,
   FOREIGN KEY (common_name) REFERENCES PlantInfo
      ON DELETE CASCADE
      ON UPDATE CASCADE
);

CREATE TABLE Grows(
   species VARCHAR(20),
   genus VARCHAR(20),
   variety VARCHAR(20),
   plot_num INTEGER,
   garden_address VARCHAR(20),
   qty INTEGER,
   plant_date DATE,
   PRIMARY KEY (species, genus, variety, plot_num, garden_address),
   FOREIGN KEY (species, genus, variety) REFERENCES
      PlantInfo(species, genus, variety)
      ON UPDATE CASCADE,
   FOREIGN KEY (plot_num) REFERENCES GardenerPlot,
   FOREIGN KEY (garden_address) REFERENCES
      GardenNumPlotsManages(address)
);
```

INSERT Statements

INSERT statements to populate each table with at least 5 tuples. You will likely want to have more than 5 tuples so that you can have meaningful queries later.

```
INSERT INTO Receives (donation_id, garden_address) VALUES (1, '123 Elm
St');
INSERT INTO Receives (donation_id, garden_address) VALUES (2, '456 Oak
St');
```

```
INSERT INTO Receives (donation_id, garden_address) VALUES (3, '789 Pine
St');
INSERT INTO Receives (donation_id, garden_address) VALUES (4, '101 Maple
St');
INSERT INTO Receives (donation_id, garden_address) VALUES (5, '202 Birch
St');

INSERT INTO Donation (donation_id, donor_name, date, item) VALUES (1,
'Alice', '2024-07-01', 'Orange Seeds');
INSERT INTO Donation (donation_id, donor_name, date, item) VALUES (2,
'Bob', '2024-07-05', 'Shovel');
INSERT INTO Donation (donation_id, donor_name, date, item) VALUES (3,
'Charlie', '2024-07-10', 'Carrot Seeds');
INSERT INTO Donation (donation_id, donor_name, date, item) VALUES (4,
'Diana', '2024-07-15', 'Grape Tomato Seeds');
INSERT INTO Donation (donation_id, donor_name, date, item) VALUES (5,
'Eve', '2024-07-20', 'Lawn Chair');

INSERT INTO Stores (garden_address, tool_type) VALUES ('123 Elm St',
'Shovel');
INSERT INTO Stores (garden_address, tool_type) VALUES ('123 Elm St',
'Hoe');
INSERT INTO Stores (garden_address, tool_type) VALUES ('202 Birch St',
'Shovel');
INSERT INTO Stores (garden_address, tool_type) VALUES ('789 Pine St',
'Trowel');
INSERT INTO Stores (garden_address, tool_type) VALUES ('101 Maple St',
'Rake');

INSERT INTO Tool (tool_type, availability) VALUES ('Shovel', 'Y');
INSERT INTO Tool (tool_type, availability) VALUES ('Rake', 'N');
INSERT INTO Tool (tool_type, availability) VALUES ('Hoe', 'Y');
INSERT INTO Tool (tool_type, availability) VALUES ('Pruner', 'N');
INSERT INTO Tool (tool_type, availability) VALUES ('Trowel', 'Y');

INSERT INTO GardenManager (name, phone, email) VALUES ('John Doe',
'123-456-7890', 'john@vancouver.com');
INSERT INTO GardenManager (name, phone, email) VALUES ('Jane Smith',
'234-567-8901', 'jane@burnaby.com');
INSERT INTO GardenManager (name, phone, email) VALUES ('Emily Davis',
```

```
'345-678-9012', 'emily@gmail.com');
INSERT INTO GardenManager (name, phone, email) VALUES ('Michael Brown',
'456-789-0123', 'michael@gmail.com');
INSERT INTO GardenManager (name, phone, email) VALUES ('Sarah Wilson',
'567-890-1234', 'sarah@burnaby.com');
INSERT INTO GardenNumPlots (address, num_of_plots) VALUES ('123 Elm St',
10);
INSERT INTO GardenNumPlots (address, num_of_plots) VALUES ('456 Oak St',
15);
INSERT INTO GardenNumPlots (address, num_of_plots) VALUES ('789 Pine St',
20);
INSERT INTO GardenNumPlots (address, num_of_plots) VALUES ('101 Maple St',
25);
INSERT INTO GardenNumPlots (address, num_of_plots) VALUES ('202 Birch St',
30);

INSERT INTO GardenInfo (address, garden_name, num_of_plots) VALUES ('123
Elm St', 'Elm Garden', 10);
INSERT INTO GardenInfo (address, garden_name, num_of_plots) VALUES ('456
Oak St', 'Oak Garden', 15);
INSERT INTO GardenInfo (address, garden_name, num_of_plots) VALUES ('789
Pine St', 'Pine Garden', 20);
INSERT INTO GardenInfo (address, garden_name, num_of_plots) VALUES ('101
Maple St', 'Maple Garden', 25);
INSERT INTO GardenInfo (address, garden_name, num_of_plots) VALUES ('202
Birch St', 'Birch Garden', 30);

INSERT INTO GardenManages (garden_name, manager_email) VALUES ('Elm
Garden', 'john@vancouver.com');
INSERT INTO GardenManages (garden_name, manager_email) VALUES ('Oak
Garden', 'jane@burnaby.com');
INSERT INTO GardenManages (garden_name, manager_email) VALUES ('Pine
Garden', 'emily@gmail.com');
INSERT INTO GardenManages (garden_name, manager_email) VALUES ('Maple
Garden', 'michael@gmail.com');
INSERT INTO GardenManages (garden_name, manager_email) VALUES ('Birch
Garden', 'sarah@burnaby.com');

INSERT INTO Gardener (email, phone, name) VALUES ('johndoe@gmail.com',
'604-123-4567', 'John Doe');
```

University of British Columbia, Vancouver

Department of Computer Science

```
INSERT INTO Gardener (email, phone, name) VALUES ('janedoe@gmail.com',
'778-321-1234', 'Jane Doe');
INSERT INTO Gardener (email, phone, name) VALUES ('marysmith@outlook.com',
'627-000-1111', 'Mary Smith');
INSERT INTO Gardener (email, phone, name) VALUES
('maria_rodriguez@yahoo.com', '778-900-8888', 'Maria Rodriguez');
INSERT INTO Gardener (email, phone, name) VALUES ('jamesj@gmail.com',
'604-444-4444', 'James Johnson');

INSERT INTO GardenerPlot (garden_address, gardener_email, plot_num,
sun_exposure, plot_size) VALUES ('123 Elm St', 'johndoe@gmail.com', 1,
'full sun', 5);
INSERT INTO GardenerPlot (garden_address, gardener_email, plot_num,
sun_exposure, plot_size) VALUES ('123 Elm St', 'janedoe@gmail.com', 2,
'part sun', 5);
INSERT INTO GardenerPlot (garden_address, gardener_email, plot_num,
sun_exposure, plot_size) VALUES ('456 Oak St', 'marysmith@outlook.com', 12,
'part shade', 3);
INSERT INTO GardenerPlot (garden_address, gardener_email, plot_num,
sun_exposure, plot_size) VALUES ('456 Oak St', 'marysmith@outlook.com', 13,
'part sun', 3);
INSERT INTO GardenerPlot (garden_address, gardener_email, plot_num,
sun_exposure, plot_size) VALUES ('789 Pine St', 'jamesj@gmail.com', 41,
'full shade', 4);

INSERT INTO Compost (bin_id, capacity) VALUES (123, 7);
INSERT INTO Compost (bin_id, capacity) VALUES (001, NULL);
INSERT INTO Compost (bin_id, capacity) VALUES (1004, 10);
INSERT INTO Compost (bin_id, capacity) VALUES (22, NULL);
INSERT INTO Compost (bin_id, capacity) VALUES (2098, 12);

INSERT INTO HasCompost (bin_id, garden_address) VALUES (123, '123 Elm St');
INSERT INTO HasCompost (bin_id, garden_address) VALUES (001, '456 Oak St');
INSERT INTO HasCompost (bin_id, garden_address) VALUES (1004, '789 Pine
St');
INSERT INTO HasCompost (bin_id, garden_address) VALUES (22, '101 Maple
St');
INSERT INTO HasCompost (bin_id, garden_address) VALUES (2098, '202 Birch
St');
```

University of British Columbia, Vancouver

Department of Computer Science

```
INSERT INTO PlantInfo (species, genus, variety, common_name) VALUES
('Cucumis sativus', 'Cucumis', 'pickling', 'Cucumber');
INSERT INTO PlantInfo (species, genus, variety, common_name) VALUES
('Lycopersicon esculentum', 'Lycopersicon', 'roma', 'Tomato');
INSERT INTO PlantInfo (species, genus, variety, common_name) VALUES
('Brassica oleracea', 'Brassica', 'capitata', 'Cabbage');
INSERT INTO PlantInfo (species, genus, variety, common_name) VALUES
('Daucus carota', 'Daucus', 'imperator', 'Carrot');
INSERT INTO PlantInfo (species, genus, variety, common_name) VALUES
('Capsicum annuum', 'Capsicum', 'bell', 'Pepper');
INSERT INTO PlantInfo (species, genus, variety, common_name) VALUES ('Rosa
chinensis', 'Rosa', 'Angel Wings');

INSERT INTO PlantColour (common_name, colour) VALUES ('Cucumber', 'Green');
INSERT INTO PlantColour (common_name, colour) VALUES ('Tomato', 'Red');
INSERT INTO PlantColour (common_name, colour) VALUES ('Cabbage', 'Green');
INSERT INTO PlantColour (common_name, colour) VALUES ('Carrot', 'Orange');
INSERT INTO PlantColour (common_name, colour) VALUES ('Pepper', NULL);
INSERT INTO PlantColour (common_name, colour) VALUES ('Angel Wings',
'Pink');

INSERT INTO PlantHarvest (common_name, harvest_time) VALUES ('Cucumber',
60);
INSERT INTO PlantHarvest (common_name, harvest_time) VALUES ('Tomato', 75);
INSERT INTO PlantHarvest (common_name, harvest_time) VALUES ('Cabbage',
85);
INSERT INTO PlantHarvest (common_name, harvest_time) VALUES ('Carrot', 70);
INSERT INTO PlantHarvest (common_name, harvest_time) VALUES ('Pepper', 80);
INSERT INTO PlantHarvest (common_name, harvest_time) VALUES ('Angel Wings',
NULL);

INSERT INTO Grows (species, genus, variety, plot_num, garden_address, qty,
plant_date) VALUES ('Lycopersicon esculentum', 'Lycopersicon', 'roma', 1,
'123 Elm St', 4, TO_DATE('2024-05-24', 'YYYY-MM-DD'))
INSERT INTO Grows (species, genus, variety, plot_num, garden_address, qty,
plant_date) VALUES ('Capsicum annuum', 'Capsicum', 'bell', 2, '456 Oak St',
6, TO_DATE('2024-06-23', 'YYYY-MM-DD'))
INSERT INTO Grows (species, genus, variety, plot_num, garden_address, qty,
plant_date) VALUES ('Capsicum annuum', 'Capsicum', 'bell', 3, '456 Oak St',
7, TO_DATE('2024-05-31', 'YYYY-MM-DD'))
```

University of British Columbia, Vancouver

Department of Computer Science

```
INSERT INTO Grows (species, genus, variety, plot_num, garden_address, qty,
plant_date) VALUES ('Rosa chinensis', 'Rosa', 'angel wings', 4, '123 Elm
St', 11, TO_DATE('2024-08-13', 'YYYY-MM-DD'))
INSERT INTO Grows (species, genus, variety, plot_num, garden_address, qty,
plant_date) VALUES ('Lycopersicon esculentum', 'Lycopersicon', 'roma', 5,
'789 Pine St', 20, TO_DATE('2024-04-23', 'YYYY-MM-DD'))
```