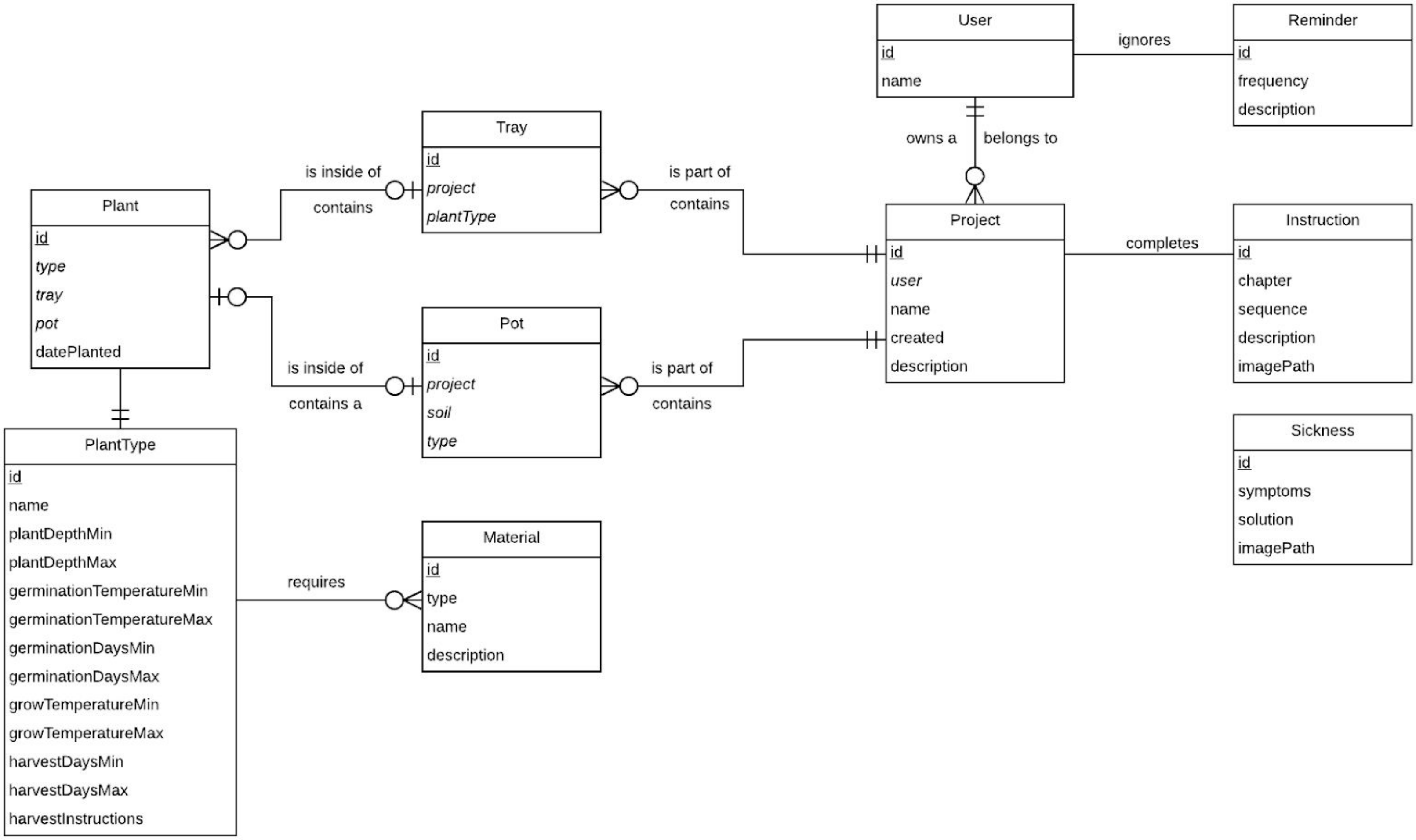
ISTE 432.01

Exercise 01 - KGA Database

Ellie Parobek
Lorenzo Romero

ER Diagram


User
id
name
ignores
Reminder
id
frequency
description
owns a
belongs to
Tray
id
project
plantType
is inside of
contains
is part of
contains
Plant
id
type
tray
pot
datePlanted
Project
id
user
name
created
description
completes
Instruction
id
chapter
sequence
description
imagePath
Pot
id
project
soil
type
is inside of
contains a
is part of
contains
PlantType
id
name
plantDepthMin
plantDepthMax
germinationTemperatureMin
germinationTemperatureMax
germinationDaysMin
germinationDaysMax
growTemperatureMin
growTemperatureMax
harvestDaysMin
harvestDaysMax
harvestInstructions
requires
Material
id
type
name
description
Sickness
id
symptoms
solution
imagePath

Functional Dependencies

{ User(id) } → { name }

{ Reminder(id) } → { frequency, description }

{ Instruction(id) } → { chapter, sequence, description, imagePath }
{ Instruction(chapter), Instruction(sequence) } → { id, description, imagePath }

{ Sickness(id) } → { symptoms, solution, imagePath }

{ PlantType(id) } → { name, plantDepthMin, plantDepthMax, germinationTemperatureMin, germinationTemperatureMax, germinationDaysMin, germinationDaysMax, growTemeratureMin, growTemperatureMax, harvestDaysMin, harvestDaysMax, harvestInstructions }
{ PlantType(name) } → { id, plantDepthMin, plantDepthMax, germinationTemperatureMin, germinationTemperatureMax, germinationDaysMin, germinationDaysMax, growTemeratureMin, growTemperatureMax, harvestDaysMin, harvestDaysMax, harvestInstructions }

{ Material(id) } → { type, name, description }
{ Material(type), Material(name) } → { id, description }

{ Project(id) } → { user, name, created, description }
{ Project(user), Project(name) } → { id, created, description }

{ Tray(id) } → { project, plantType }

{ Pot(id) } → { project, soil, type }

Relational Notation

User (id, name)

Project (id, *user*, created, description)
Project(user) must exist in User(id)

PlantType (id, name, plantDepthMin, plantDepthMax, germinationTemperatureMin, germinationTemperatureMax, germinationDaysMin, germinationDaysMax, growTemeratureMin, growTemperatureMax, harvestDaysMin, harvestDaysMax, harvestInstructions)

Material (id, type, name, description)

Tray (id, *project, plantType*)
Tray(project) must exist in Project(id)
Tray(plantType) must exist in PlantType(id)

Pot (id, *project, soil, type*)
Pot(project) must exist in project(id)
Pot(soil) must exist in Material(id)
Pot(type) must exist in Material(id)

Plant (id, *type, tray, pot,* datePlanted)
Plant(type) must exist in PlantType(id)
Plant(tray) must exist in Tray(id)
Plant(pot) must exist in Pot(id)

Reminder (id, frequency, description)

Instruction (id, chapter, sequence, description, imagePath)

Sickness (id, symptoms, solution, imagePath)

Narrative

Our database was designed to minimize redundant tables and entries while allowing the features of the original application to be implemented without any, or very little, modification.

While the application's outline implies that there will be only a singular user of this application per installation, a table for different 'Users' was created to allow expansion. The information collected about a user is kept to a minimal, no details about the user are needed for any other functions.

A collection of vegetables a user wants to grow will be stored in a 'Project'. The project will be tied to one and only one user and will record when it was created. The user may append a description about what the project is.

There are a variety of different materials related to gardening, including pots, trays, and soil types. A generic 'Material' entity exists to track all the materials the application supports. These materials have a 'type' attribute that specified what they are (e.g., pot, tray, soil), a user-friendly display name, and an optional description.

The different seed and vegetable types outlined in the specifications are directly linked to one another—a vegetable has one and only one seed, and a seed can only grow into one kind of vegetable. To avoid having attributes for the same entity scattered across two different tables, a single table 'PlantType' was created that contains information about both seed and grown vegetable. It is not necessary for the user to directly select a seed, since it is understood that their finished plant will come from one; the user will only select what type of vegetable/plant they wish to grow. Per the outline, a plant type may list materials that are 'required' for growth; these materials are linked to the 'Material' table.

Upon research, the seeds of a plant are first placed into a tray along with many other seeds. There is only one type of seed per tray. A 'Tray' is linked directly to a project, and has one and only one plant type specified. In practice, a tray will be associated with any number of plants of the same type.

Once a seed has matured into a seedling, it is transferred to its own pot. A 'Pot' is linked directly to a project, and contains only one plant. A pot also has one type of soil, and one type of actual pot. The soil and pot types are linked to the 'Material' table.

An actual plant being grown as part of a project is represented by the 'Plant' table. A plant is of only one type, and can be in either a tray or pot, but not both. The project a plant belongs to will be derived from its tray or pot membership. A plant records the date it was planted; that information is used, along with the germination, growth, and harvest time stored in the plant type and current system time, to determine the number of days before each (aforementioned) gardening milestone.

There are a series of generic instructions for gardening that are sorted into 'chapters'. These instruction steps are stored in the 'Instruction' table, along with their chapter, and sequence within each chapter. These instructions act as a stand-alone 'book', but can also be checked off as part of a project. The date an instruction is checked off can be used to create a histogram, or progress par, showing progress on the project.

The outline describes a database of sicknesses, which will give users 'solutions' to certain problems encountered when gardening. Since this is a stand-alone reference that is accessed by a discrete button in the interface, the 'Sickness' table exists as a stand-alone entity storing the text content of each sickness.

The outline also describes reminders that the user will be given at certain intervals. Since these are largely stand-alone, they too are in a stand-alone entity containing their frequency (listed in seconds) and the message they should display. Since there is an option for users to disable specific reminders, a table exists that will show that a user has set a reminder to be ignored; when the application attempts to show reminders, it will reference this table and abort showing reminders the user has marked as disabled.

SQLite3 Statements

Database Creation

```
CREATE TABLE User (
    id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    name TEXT NOT NULL
);

/* A plantable vegetable */
/* Depth values are stored in mm, temperature values stored in centigrade, harvest time stored in days */
CREATE TABLE PlantType (
    id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    name TEXT NOT NULL,
    plantDepthMin INTEGER NOT NULL,
    plantDepthMax INTEGER NOT NULL,
    germinationTemperatureMin REAL NOT NULL,
    germinationTemperatureMax INTEGER NOT NULL,
    germinationDaysMin INTEGER NOT NULL,
    germinationDaysMax INTEGER NOT NULL,
    growTemperatureMin INTEGER NOT NULL,
    growTemperatureMax INTEGER NOT NULL,
    harvestDaysMin INTEGER NOT NULL,
    harvestDaysMax INTEGER NOT NULL,
    harvestInstructions TEXT NOT NULL,
    UNIQUE(name)
);

/*  */
CREATE TABLE Material (
    id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    type TEXT NOT NULL,
    name TEXT NOT NULL,
    description TEXT NOT NULL,
    UNIQUE(type, name)
);

/* List of required materials for planting */
/* This is just for the user to know what they need */
CREATE TABLE PlantType_Material (
    plantType INTEGER NOT NULL,
    material INTEGER NOT NULL,
    FOREIGN KEY (plantType) REFERENCES PlantType(id) ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (material) REFERENCES Material(id) ON UPDATE CASCADE ON DELETE CASCADE
);

/* A user's project */
CREATE TABLE Project (
    id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    user INTEGER NOT NULL,
    name TEXT NOT NULL,
    created DATETIME NOT NULL,
    description TEXT NOT NULL,
    FOREIGN KEY (user) REFERENCES User(id) ON UPDATE CASCADE ON DELETE CASCADE,
    UNIQUE (user, name)
);

/* A plant being grown as part of this project */
/* Tray -OR- Pot defined, but should not be in both at the same time */
CREATE TABLE Plant (
    id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    plantType INTEGER NOT NULL,
    tray INTEGER DEFAULT NULL,
    pot INTEGER DEFAULT NULL,
    datePlanted DATETIME NOT NULL,
    FOREIGN KEY (plantType) REFERENCES PlantType(id) ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (tray) REFERENCES Tray(id) ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (pot) REFERENCES Pot(id) ON UPDATE CASCADE ON DELETE CASCADE
);

/* A tray containing seedlings */
CREATE TABLE Tray (
    id INTEGER NOT NULL,
    project INTEGER NOT NULL,
    plantType INTEGER NOT NULL,
    FOREIGN KEY (project) REFERENCES Project(id) ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (plantType) REFERENCES PlantType(id) ON UPDATE CASCADE ON DELETE CASCADE
);

/* A pot containing a single plant */
/* Has a type (material) */
/* Has a soil type (material) */
CREATE TABLE Pot (
    id INTEGER NOT NULL,
    project INTEGER NOT NULL,
    soil INTEGER NOT NULL,
    type INTEGER NOT NULL,
    FOREIGN KEY (project) REFERENCES Project(id) ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (soil) REFERENCES Material(id) ON UPDATE CASCADE,
    FOREIGN KEY (type) REFERENCES Material(id) ON UPDATE CASCADE
);

/* An ailment that can fall upon a plant */
CREATE TABLE Sickness (
    id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    name TEXT NOT NULL,
    symptoms TEXT NOT NULL,
    solution TEXT NOT NULL,
    imagePath TEXT DEFAULT NULL
);

/* Reference guide for gardening */
CREATE TABLE Instruction (
    id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    chapter INTEGER NOT NULL,
    sequence INTEGER NOT NULL,
    description TEXT NOT NULL,
    imagePath DEFAULT NULL,
    UNIQUE (chapter, sequence)
);

/* Tracks steps that have been marked as completed in a project */
```

```
CREATE TABLE Project_InstructionCompleted (
    project INTEGER NOT NULL,
    instruction INTEGER NOT NULL,
    dateCompleted DATETIME NOT NULL,
    FOREIGN KEY (project) REFERENCES Project(id) ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (instruction) REFERENCES Instruction(id) ON UPDATE CASCADE ON DELETE CASCADE,
    UNIQUE (project, instruction)
);

/* A reminder for the user */
CREATE TABLE Reminder (
    id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    interval INTEGER NOT NULL,
    description TEXT NOT NULL
);

/* Indicates a user does not want to receive the specified notification */
CREATE TABLE User_IgnoreReminder (
    user INTEGER NOT NULL,
    reminder INTEGER NOT NULL
);
```

Database Population

```
INSERT INTO User VALUES
    (1, 'Robert Tables'),
    (2, 'Ellie Parobek'),
    (3, 'Larry Romero');

INSERT INTO PlantType VALUES
    (1, "Broccoli", 6, 13, 10, 29, 7, 10, 18, 24, 78, 98, "Start Harvesting when the flower heads form, but before the plants start to yellow.  You can cut the main head of the broccoli when it’s 3-4 inches long in diameter and 4 inches high.  Finally cut the main stem 6 inches below the head or 1/2 inches above where the roots begin to split.  Dry and Refrigerate or Freeze.  Fresh for up to two weeks."),
    (2, "Bean", 25, 25, 18, 29, 4, 10, 21, 27, 45, 60, "Snap beans.  harvesting daily when beans are pencil size in diameter encourages continual production.  Refrigerate for 1-1.5 weeks.  Do not wash until ready to eat them.
Dry Beans: should remain on bush until leaves have withered and fallen."),
    (3, "Onion", 6, 13, 18, 29, 4, 12, 13, 24, 80, 120, "When onions start to mature, the tops become yellow and begin to fall over. Loosen the soil to encourage drying, and after a few days turn them up and let them cure on dry ground.  When tops are brown, pull the onions.  Be sure to harvest in late summer, before cool weather.");

INSERT INTO Material VALUES
    (1, 'soil', "Black Gold Potting Soil", ""),
    (2, 'soil', "Organic Seed Starting Potting Soil", ""),
    (3, 'tray', "Tray", "A container for your seedlings"),
    (4, 'tray', "Tray Cover", "A cover for your seedling container"),
    (5, 'pot', "GoPro5 Gallon", "I imagine this is 5 gallons..."),
    (6, 'pot', "Plant Warrior", "It kills the plants");

INSERT INTO PlantType_Material VALUES
    (1, 1),
    (1, 3),
    (1, 4),
    (1, 5),
    (2, 1),
    (2, 3),
    (2, 4),
    (2, 5),
    (3, 1),
    (3, 3),
    (3, 4),
    (3, 5);

INSERT INTO Project VALUES
    (1, 2, "A depressing garden", "2019-01-24", "No signs of life..."),
    (2, 3, "A slightly more lively garden", "2019-01-25", ""),
    (3, 1, "Robert's Garden", "2019-01-21", "");

INSERT INTO Plant VALUES
    (1, 1, 1, NULL, "2019-01-24"),
    (2, 1, 1, NULL, "2019-01-24"),
    (3, 2, 2, NULL, "2019-01-24"),
    (4, 1, NULL, 1, "2019-01-24"),
    (5, 2, NULL, 2, "2019-01-24"),
    (6, 2, NULL, 3, "2019-01-24");

INSERT INTO Tray VALUES
    (1, 1, 1),
    (2, 1, 2),
    (3, 2, 3);

INSERT INTO Pot VALUES
    (1, 1, 2, 5),
    (2, 2, 1, 6),
    (3, 2, 1, 6);

INSERT INTO Sickness VALUES
    (1, "Damping-Off", "A plant disease occurring in excessively damp conditions, in particular the collapse and death of a young
seedling as a result of a fungal infection",
                "Watering seedlings with an antifungal solution can help protect seedlings.  You can also simmer mashed up garlic, and
pour it over the seedlings once the water has cooled", "damping.gif"),
    (2, "Too Little Water", "Leaves are turning brown around the edges, curling up or turning translucent they may be suffering from
dehydration.  Stick a finger 1 inch into soil.  If it feels dry, the plant requires watering",
                "Stick finter 1 inch into the plant's soil.  If it feels dry, your plant requires watering pick up smaller plants, before and after
watering, and take not of their weight", "dehydration.gif"),
    (3, "Lack of Sunshine", "Plants deprived of light have tiny leaves and yellow color because they don't form chlorophyll.  When
plants lack sunlight, leaves begin to change colors from green to brown and yellow.  Leaves will start to blister leaving tan, brown, or
warty growths forming where the blisters were.",
                "Check the area the plants are in.  Make sure they have access to direct sunlight.  if you cannot get quality sunlight,
consider making or buying a grow light for the plants.", NULL);

INSERT INTO Instruction VALUES
    (1, 1, 1, "Place your seed tray into the bottom tray.", NULL),
    (2, 1, 2, "Make your labels.  Get creative.", NULL),
    (3, 1, 3, "Fill the container about 3/4 of the way with your organic seed soil (or within .5 inches from the top).  Wet the soil so it's
moist like firm mud.  Be careful to avoid overfilling it.", NULL);

INSERT INTO Project_InstructionCompleted VALUES
    (1, 1, "2018-01-24"),
    (1, 2, "2018-01-24"),
    (1, 3, "2018-01-24");

INSERT INTO Reminder VALUES
    (1, 86400, "Have you watered your plants today?"),
    (2, 86400, "Have you checked your plants leaves today?"),
    (3, 86400, "Are any of your plants ready to harvest?");

INSERT INTO User_IgnoreReminder VALUES
    (3, 1),
    (3, 2),
    (3, 3);
```