University of British Columbia, Vancouver **Group: 93**
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: Oct. 1st, 2025

Group Number: 93

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Simon Mattern	85839439	m7r9u	SimonMattern03@gmail.com
Gavin Krebbers	47192521	d4o5r	gkrebbers5@gmail.com
Adrian Leung	52615242	f8e6e	adrian.leung4228@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

Project Description:

Our goal for this project is to create a citizen science application, specifically focusing on tracking and documenting the distribution of marine flora and fauna species. Not only will the application assist in scientific research, such as ecological surveys and documentation of rare species, but it will also provide an outlet for hobbyists to explore, record and collaborate about the dynamic aquatic landscape.

Relational Schema:

user(
email: VARCHAR(255) (PK),
username: VARCHAR(100) NOT NULL,
password: VARCHAR(255) NOT NULL
)
professional(
email: VARCHAR(255) (PK) (FK),
username: VARCHAR(100),
certification: VARCHAR(255),
degree: VARCHAR(255),
specialization: VARCHAR(255)
)
project_user(
projectID: INT (PK) (FK),
email: VARCHAR(255) (PK) (FK)
)
project_professional(
email: VARCHAR(255) (PK) (FK),
projectID: INT (PK) (FK)
)
project(
projectID: INT (PK),
name: VARCHAR(255),
description: TEXT
)
project_observation(
projectID: INT (PK), (FK),
observationID: INT (PK), (FK)
)

observation(
observationID: INT (PK),
longitude: DECIMAL(9,6),
latitude: DECIMAL(9,6),
date: DATE,
quantity: INT,
notes: TEXT,
meanLongitude: DECIMAL(9,6) NOT NULL (FK),
meanLatitude: DECIMAL(9,6) NOT NULL (FK),
scientificName: VARCHAR(255) NOT NULL (FK),
email: VARCHAR(255) NOT NULL, (FK)
professionalEmail: VARCHAR(255) (FK),
dateConfirmed: DATE
)
media(
observationID: INT (PK) (FK),
mediaID: INT (PK),
URL: VARCHAR(500),
mediaType: VARCHAR(50)
)
location(
meanLongitude: DECIMAL(9,6) (PK),
meanLatitude: DECIMAL(9,6) (PK),
name: VARCHAR(255)
)
groupChat_user(
email: VARCHAR(255) (PK) (FK),
ID: INT (PK) (FK)
)
groupChat(
ID: INT (PK),
name: VARCHAR(255),
createdAt: TIMESTAMP
)
message(
ID: INT (PK),
data: TEXT,
time_sent: TIMESTAMP,
group_chat_id: INT NOT NULL (FK),
email: VARCHAR(255) NOT NULL (FK)

)
species(
 scientificName: VARCHAR(255) (PK),
 commonName: VARCHAR(255),
 description: TEXT,
 genus: VARCHAR(255) NOT NULL (FK)
)
taxonomy(
 genus: VARCHAR(255) (PK),
 order: VARCHAR(255)(Not Null),
 family: VARCHAR(255)(Not Null),
 class: VARCHAR(255)(Not Null),
 phylum: VARCHAR(255)(Not Null),
 kingdom: VARCHAR(255)(Not Null)
)

Functional Dependencies:

User:
email -> username, password

Professional:
email -> username, certification, degree, specialization

Project:
projectID -> name, description

Observation:
observationID -> longitude, latitude, date, quantity, notes, meanLongitude, meanLatitude, scientificName, email, professionalEmail, dateConfirmed

Media:
(observationID, mediaID) -> URL, mediaType

Location:
(meanLongitude, meanLatitude) -> name

GroupChat:
ID -> name, created_at

Messages:
ID -> data, timeSent, groupChatID, email

Species:
scientificName -> commonName, description, genus

Taxonomy:
genus -> order, family, class, phylum, kingdom
Order -> family, class, phylum, kingdom
Family -> class, phylum, kingdom
Class -> phylum, kingdom
phylum -> kingdom

Minimal Cover & Normalization of FD:

User:
email -> username
email -> password

Because there is only one functional dependency, the only simplification for this schema is to convert it into standard form. Because the primary key's closure includes all attributes and there are no other candidate keys or FD, this entity is in BCNF

Professional:
email -> username
email -> certification
email -> degree
email -> specialization

Because there is only one functional dependency, the only simplification for this schema is to convert it into standard form. Because the primary key's closure includes all attributes and there are no other candidate keys or FD, this entity is in BCNF

Project:
projectID -> name
projectID -> description

Because there is only one functional dependency, the only simplification for this schema is to convert it into standard form. Because the primary key's closure includes all attributes and there are no other candidate keys or FD, this entity is in BCNF

Observation:
observationID -> longitude
observationID -> latitude
observationID -> date
observationID -> quantity
observationID -> notes
observationID -> meanLongitude

observationID -> meanLatitude
observationID -> scientificName
observationID -> email
observationID -> professionalEmail
observationID -> dateConfirmed

Because there is only one functional dependency, the only simplification for this schema is to convert it into standard form. Because the primary key’s closure includes all attributes and there are no other candidate keys or FD, this entity is in BCNF

Media:
observationID, mediaID -> URL
observationID, mediaID -> mediaType

Because there is only one functional dependency, this schema is first converted into standard form then we can minimize the LHS. However, there are no other FDs to let us minimize the LHS, so it cannot be simplified. Because the primary key’s closure includes all attributes and there are no other candidate keys or FD, this entity is in BCNF

Location:
meanLongitude, meanLatitude -> name

Because there is only one functional dependency, this schema is first converted into standard form then we can minimize the LHS. However, there are no other FDs to let us minimize the LHS, so it cannot be simplified. Because the primary key’s closure includes all attributes and there are no other candidate keys or FD, this entity is in BCNF

GroupChat:
ID -> name
ID -> created_at

Because there is only one functional dependency, the only simplification for this schema is to convert it into standard form. Because the primary key’s closure includes all attributes and there are no other candidate keys or FD, this entity is in BCNF

Messages:
ID -> data
ID -> timeSent
ID -> groupChatID
ID -> email

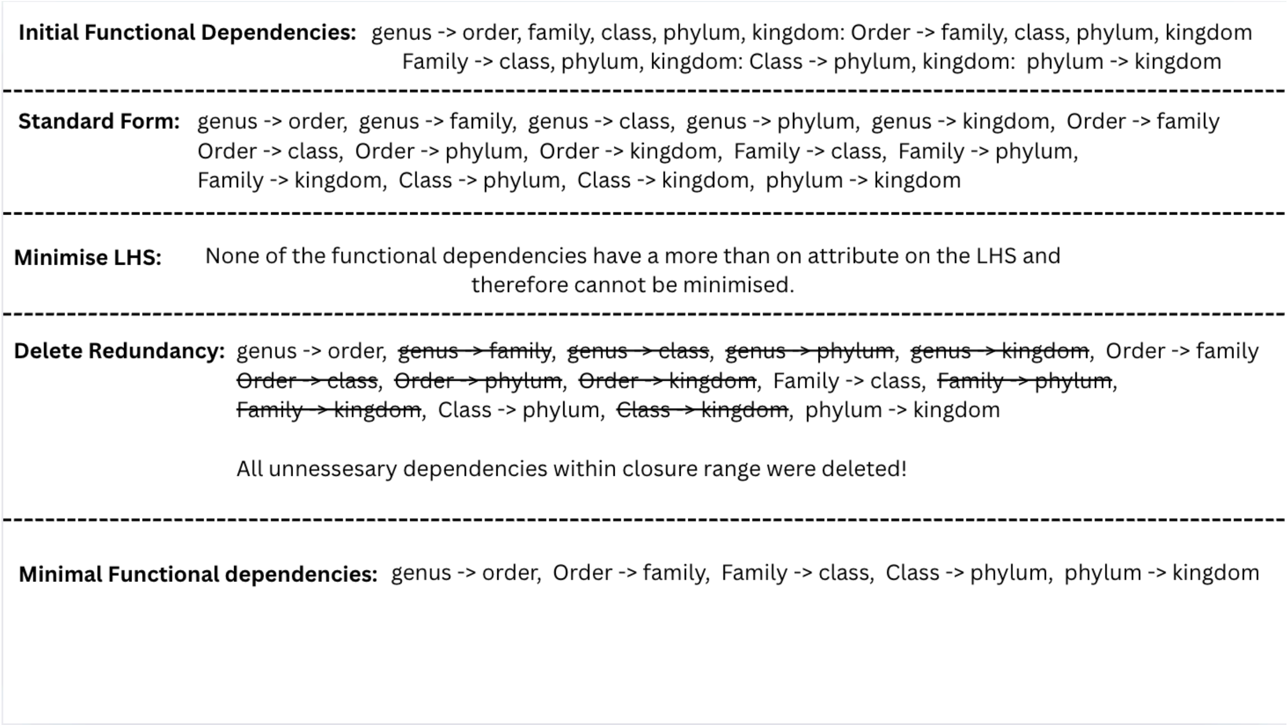
Because there is only one functional dependency, the only simplification for this schema is to convert it into standard form. Because the primary key’s closure includes all attributes and there are no other candidate keys or FD, this entity is in BCNF

Species:

scientificName -> commonName
scientificName -> description
scientificName -> genus

Because there is only one functional dependency, the only simplification for this schema is to convert it into standard form. Because the primary key's closure includes all attributes and there are no other candidate keys or FD, this entity is in BCNF

Taxonomy:



Initial Functional Dependencies: genus -> order, family, class, phylum, kingdom: Order -> family, class, phylum, kingdom
Family -> class, phylum, kingdom: Class -> phylum, kingdom: phylum -> kingdom

Standard Form: genus -> order, genus -> family, genus -> class, genus -> phylum, genus -> kingdom, Order -> family
Order -> class, Order -> phylum, Order -> kingdom, Family -> class, Family -> phylum,
Family -> kingdom, Class -> phylum, Class -> kingdom, phylum -> kingdom

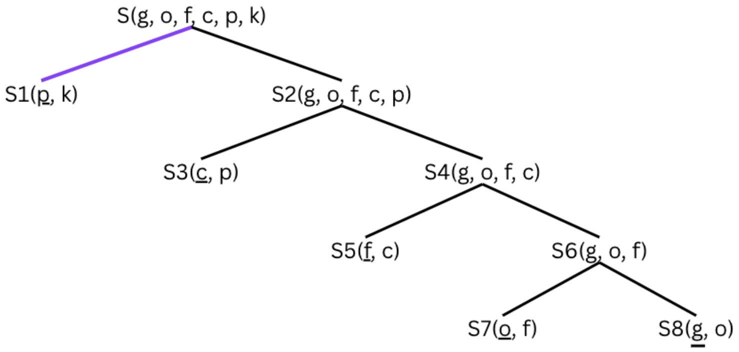
Minimise LHS: None of the functional dependencies have a more than on attribute on the LHS and therefore cannot be minimised.

Delete Redundancy: genus -> order, ~~genus -> family~~, ~~genus -> class~~, ~~genus -> phylum~~, ~~genus -> kingdom~~, Order -> family
~~Order -> class~~, ~~Order -> phylum~~, ~~Order -> kingdom~~, Family -> class, ~~Family -> phylum~~,
~~Family -> kingdom~~, Class -> phylum, ~~Class -> kingdom~~, phylum -> kingdom

All unnessesary dependencies within closure range were deleted!

Minimal Functional dependencies: genus -> order, Order -> family, Family -> class, Class -> phylum, phylum -> kingdom

Minimal Functional dependencies: genus -> order, Order -> family, Family -> class, Class -> phylum, phylum -> kingdom



The new Relations are now in BCNF. This is because the key of every relation (underlined on the LHS) closure includes every single attribute in its set using the minimised functional dependencies. There are no candidate keys or additional functional dependencies outside of a superkey that allow us to derive any attributes in the sets. All this comes together to create a set of new relations that are normalized into BCNF.

New Schema for Taxonomy:

Genus(

genus: VARCHAR(255) (PK)

order: VARCHAR(255) (FK) (Not Null)

)

Order(

order: VARCHAR(255) (PK)

family: VARCHAR(255) (FK) (Not Null)

)

Family(

family: VARCHAR(255) (PK)

Class: VARCHAR(255) (FK) (Not Null)

)

Class(

class: VARCHAR(255) (PK)

phylum: VARCHAR(255) (FK) (Not Null)

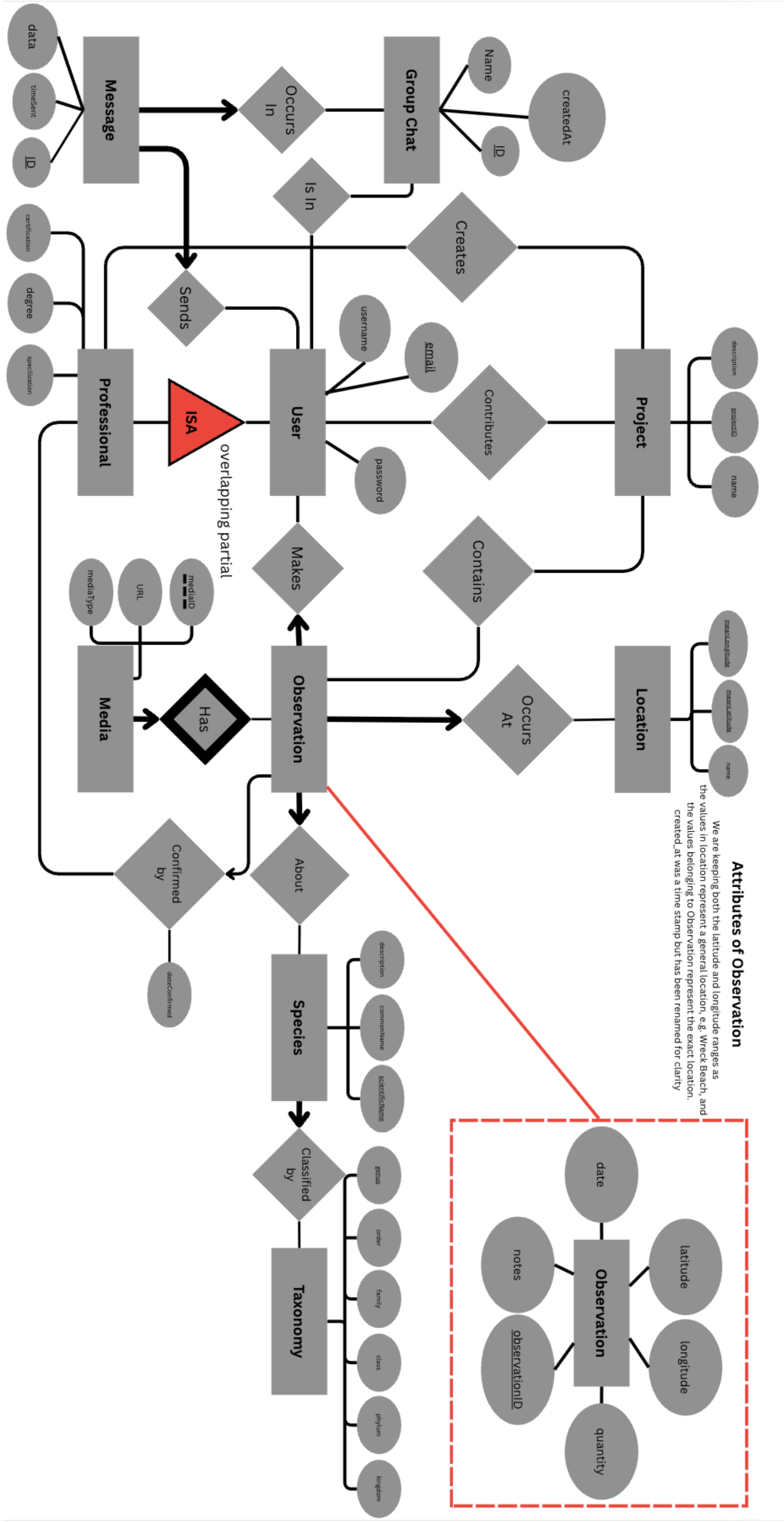
)

Phylum(

phylum: VARCHAR(255) (PK)

Kingdom: VARCHAR(255) (Not Null)

)



Attributes of Observation

We are keeping both the latitude and longitude ranges as the values in location represent a general location, e.g. Wreck Beach, and the values belonging to Observation represent the exact location. created_at was a time stamp but has been renamed for clarity

ER Diagram Changes:

- The ISA relation constraints now listed (Overlapping, Partial) for clarity. Since there is only a single child entity, the Overlap to show that an user can be a user and a professional at the same time and Partial to show that a user does not need to be a professional, they can exist as a user alone.
- Weak entity now listed (Media) with the correct diagram notation. We meant to indicate this in our original diagram but failed to use the correct notation.
- Description added for the Observation entity outlining constraints. This was primarily to show that we have chosen to maintain the coordinate information (lat and long) in both observation and location entities. While these names are similar, they represent separate important information. Within the location entity, coordinate ranges are indicated to show that central coordinate of a generalized region (ex. Wreck beach, point Nemo, Great Barrier Reef). The location information in the observation represents the exact location, to a factor of ten meters. This info is important when researchers need both rough and exact locations and the names of the attributes have been changed to reflect these factors.
- All attributes changed into upper camel case for consistency with dependency diagrams and code.
- organismType attribute removed from Taxonomy entity due to redundancy. The same information is presented in the “kingdom” attribute of Taxonomy.

```
     CREATE TABLE user (
       email VARCHAR(255) PRIMARY KEY,
       username VARCHAR(255) NOT NULL,
       password VARCHAR(255) NOT NULL
       );

       CREATE TABLE professional (
       email VARCHAR(255) PRIMARY KEY,
       degree VARCHAR(255),
       certification VARCHAR(255),
       specialization VARCHAR(255),
        -- If the user is deleted, the professional should be too
       FOREIGN KEY (email) REFERENCES user(email) ON DELETE CASCADE
       );

       CREATE TABLE project (
       projectID INTEGER PRIMARY KEY AUTOINCREMENT,
       name VARCHAR(255),
       description TEXT
       );

       CREATE TABLE project_user (
       projectID INTEGER,
       email VARCHAR(255),
       PRIMARY KEY (email, projectID),
          -- If a project is deleted, all relations between the user and the deleted project
should be deleted
       FOREIGN KEY (projectID) REFERENCES project(projectID) ON DELETE CASCADE,
          -- If a user is deleted, all relations between the project and the deleted user should
        be deleted
       FOREIGN KEY (email) REFERENCES user(email) ON DELETE CASCADE
       );
```

```
	CREATE TABLE project_professional (
	projectID INTEGER,
	email VARCHAR(255),
	PRIMARY KEY (email, projectID),
	 -- If a project is deleted, all relations between the project and the deleted project should be deleted
	FOREIGN KEY (projectID) REFERENCES project(projectID) ON DELETE CASCADE,

	 -- If a professional is deleted, all relations between the project and the deleted professional should be deleted
	FOREIGN KEY (email) REFERENCES professional(email) ON DELETE CASCADE
	);

	CREATE TABLE project_observation (
	projectID INTEGER,
	observationID INTEGER,
	PRIMARY KEY (observationID, projectID),

	 -- If a project is deleted, all relations between the project and the deleted observation should be deleted
	FOREIGN KEY (projectID) REFERENCES project(projectID) ON DELETE CASCADE,
	 -- If an observation is deleted, all relations between the project and the deleted observation should be deleted
	FOREIGN KEY (observationID) REFERENCES observation(observationID) ON DELETE CASCADE
	);

	CREATE TABLE observation(
	observationID INTEGER PRIMARY KEY AUTOINCREMENT,
	longitude DECIMAL(9,6),
	latitude DECIMAL(9,6),
	date DATE,
	quantity INTEGER,
	notes TEXT,
	meanLongitude DECIMAL(9,6) NOT NULL,
	meanLatitude DECIMAL(9,6) NOT NULL,
	scientificName VARCHAR(255) NOT NULL,
	email VARCHAR(255) NOT NULL,
	professionalEmail VARCHAR(255),
	dateConfirmed DATE,
	FOREIGN KEY (meanLongitude, meanLatitude) REFERENCES location(meanLongitude, meanLatitude),
	FOREIGN KEY (scientificName) REFERENCES species(scientificName),
-- If a user is deleted that so should its observations
	FOREIGN KEY (email) REFERENCES user(email) ON DELETE CASCADE
```

```
         FOREIGN KEY (professionalEmail) REFERENCES professional(email)
        );

        CREATE TABLE media(
        observationID INTEGER,
        mediaID INTEGER,
        URL VARCHAR(500),
        mediaType VARCHAR(50),
        PRIMARY KEY (observationID, mediaID),
         -- of course, if the observation is deleted, the media must also be deleted
        FOREIGN KEY (observationID) REFERENCES observation(observationID) ON DELETE
CASCADE
        );

        CREATE TABLE location(
        meanLongitude DECIMAL(9,6),
        meanLatitude DECIMAL(9,6),
        name VARCHAR(255),
        PRIMARY KEY (meanLongitude, meanLatitude)
        );

        CREATE TABLE groupChat_user(
        email VARCHAR(255),
        ID INTEGER,
         -- -- If a group chat is deleted, all relations between the group chat and the deleted
user should be deleted
        FOREIGN KEY (email) REFERENCES user(email) ON DELETE CASCADE,
-- If a user is deleted, all relations between the user and the deleted group chat should be
deleted
        FOREIGN KEY (ID) references groupChat(ID) ON DELETE CASCADE,
        PRIMARY KEY (email, ID)
        );

        CREATE TABLE groupChat(
        ID INTEGER PRIMARY KEY AUTOINCREMENT,
        name VARCHAR(255),
        created_at TIMESTAMP
        );

        CREATE TABLE message(
        ID INTEGER PRIMARY KEY AUTOINCREMENT,
        data TEXT,
        time_sent TIMESTAMP,
        group_chat_id INTEGER NOT NULL,
        email VARCHAR(255) NOT NULL,
         -- As a message can only exist within a group chat if the group chat is deleted the message must also be deleted
         -- Messages also must have a user who sent them so if the user is deleted so must the message be
         FOREIGN KEY (group_chat_id) REFERENCES groupChat(ID) ON DELETE CASCADE,
        FOREIGN KEY (email) REFERENCES user(email) ON DELETE CASCADE
        );

        CREATE TABLE phylum (
        phylum VARCHAR(255) PRIMARY KEY,
        kingdom VARCHAR(255) NOT NULL
        );

        CREATE TABLE class (
        class VARCHAR(255) PRIMARY KEY,
        phylum VARCHAR(255) NOT NULL,
        FOREIGN KEY (phylum) REFERENCES phylum(phylum)
        );

        CREATE TABLE family (
        family VARCHAR(255) PRIMARY KEY,
        class VARCHAR(255) NOT NULL,
        FOREIGN KEY (class) REFERENCES class(class)
        );

        CREATE TABLE `order` (
        `order` VARCHAR(255) PRIMARY KEY,
        family VARCHAR(255)NOT NULL,
        FOREIGN KEY (family) REFERENCES family(family)
        );

        CREATE TABLE genus (
        genus VARCHAR(255) PRIMARY KEY,
        `order` VARCHAR(255) NOT NULL,
        FOREIGN KEY (`order`) REFERENCES `order`(`order`)
        );

        CREATE TABLE species(
        scientificName VARCHAR(255) PRIMARY KEY,
        commonName VARCHAR(255),
        description TEXT,
        genus VARCHAR(255) NOT NULL,
        FOREIGN KEY (genus) REFERENCES genus(genus)
        );

        INSERT INTO genus (genus, `order`)
```

```
VALUES ('Macrocystis', 'Laminariales'),
    ('Ulva', 'Ulvales'),
    ('Orcinus', 'Artiodactyla'),
    ('Enteroctopus', 'Octopoda'),
    ('Glaucus', 'Nudibranchia');

INSERT INTO `order` (`order`, family)
VALUES ('Laminariales', 'Laminariaceae'),
    ('Ulvales', 'Ulvaceae'),
    ('Artiodactyla', 'Delphinidae'),
    ('Octopoda', 'Enteroctopodidae'),
    ('Nudibranchia', 'Glaucidae');

INSERT INTO family (family, class)
VALUES ('Laminariaceae', 'Phaeophyceae'),
    ('Ulvaceae', 'Ulvophyceae'),
    ('Delphinidae', 'Mammalia'),
    ('Enteroctopodidae', 'Cephalopoda'),
    ('Glaucidae', 'Gastropoda');

INSERT INTO class (class, phylum)
VALUES ('Phaeophyceae', 'Ochrophyta'),
    ('Ulvophyceae', 'Chlorophyta'),
    ('Mammalia', 'Chordata'),
    ('Cephalopoda', 'Mollusca'),
    ('Gastropoda', 'Mollusca');

INSERT INTO phylum (phylum, kingdom)
VALUES ('Ochrophyta', 'Chromista'),
    ('Chlorophyta', 'Plantae'),
    ('Chordata', 'Animalia'),
    ('Mollusca', 'Animalia'),
    ('Rhodophyta', 'Plantae');

INSERT INTO species (scientificName, commonName, description, genus)
VALUES ('Macrocystis pyrifera', 'Giant Kelp', 'found this stuff everywhere, fills me with rage', 'Macrocystis'),
    ('Ulva intestinalis', 'Sea lettuce', 'green, translucent, skinny tubes', 'Ulva'),
    ('Ulva lactuca', 'Sea lettuce', 'green, translucent, sheets', 'Ulva'),
    ('Orcinus orca', 'Orca', 'still mad I can’t hunt these', 'Orcinus'),
    ('Enteroctopus dofleini', 'Giant Pacific Octopus', 'Red, 2m in diameter, depth of 30m, hiding in rocks', 'Enteroctopus');

INSERT INTO location (meanLongitude, meanLatitude, name)
VALUES ('49.245173', '-125.257978', 'Vancouver Island'),
    ('49.267324', '-123.263471', 'Wreck Beach'),
            ('49.330532', '-124.290541', 'EnglishMan River Estuary'),
            ('49.405967', '-123.469368', 'Plumper Cove'),
            ('49.451951', '-123.326787', 'Halkett Bay');

        INSERT INTO user (email, username, password)
        VALUES ('johnSmith@gmail.com', 'jsmithy', '1234'),
            ('johnDoe@gmail.com', 'johnDoe', '12j3h1k2h3'),
            ('gavinKrebbers@gmail.com', 'GBoy', 'ilovepasswords'),
            ('janeDoe@gmail.com', 'jDoe', '1234'),
            ('randomUser@gmail.com', 'jsmithy', '123N*!(2342d)'),
            ('Simon@gmail.com', 'Simon', 'IlikeResearch'),
            ('rachelResearch@gmail.com', 'rachelResearch', 'IlikeResearch'),
            ('robertResearch@gmail.com', 'robertResearch', 'IlikeResearch'),
            ('raoulResearch@gmail.com', 'raoulResearch', 'IlikeResearch'),
            ('rdanielResearch@gmail.com', 'rdanielResearch', 'IlikeResearch'),
            ('rowanResearch@gmail.com', 'rowanResearch', 'kjdssp');

        INSERT INTO professional (email, degree, certification, specialization)
        VALUES ('rachelResearch@gmail.com', 'Bachelors in Science', NULL, 'Fish'),
            ('robertResearch@gmail.com', NULL, 'Data science from BCIT', 'Data Science'),
            ('raoulResearch@gmail.com', 'Bachelors in Underwater Basket Weaving', NULL,
'Basket weaving'),
            ('rdanielResearch@gmail.com', 'PHD in computer science', NULL, NULL),
            ('Simon@gmail.com', 'Bachelors in Biology', NULL, 'Kelp'),
            ('rowanResearch@gmail.com', NULL, 'Fish Xrays', NULL);

        INSERT INTO project (name, description)
        VALUES ('Clown Fish at Wreck Beach', 'This is a project tracking spottings of clown fish
at Wreck beach near UBC'),
            ('Coral Health Monitoring', 'Volenteers document coral reef health and
biodiversity across vancouver island'),
            ('Seastar documentation survey', 'A project tracking the population of  seastars
and how many are effected by disease'),
            ('Tidepool biodiversity', 'Participants photograph and document different species
found in local tidepools'),
            ('Kelp forest monitoring', 'Citizen monitor kelp forests along the coast of
vancouver islands');

        INSERT INTO project_user (projectID, email)
        VALUES
        (1, 'johnDoe@gmail.com'),
        (1, 'johnSmith@gmail.com'),
        (3, 'janeDoe@gmail.com'),
        (3, 'gavinKrebbers@gmail.com'),
        (5, 'johnSmith@gmail.com'),
        (5, 'johnDoe@gmail.com'),
        (5, 'gavinKrebbers@gmail'),
        (5, 'janeDoe@gmail.com'),
        (5, 'randomUser@gmail.com');

        INSERT INTO project_professional (projectID, email)
        VALUES
        (1, 'Simon@gmail.com'),
        (1, 'rachelResearch@gmail.com'),
        (2, 'robertResearch@gmail.com'),
        (2, 'raoulResearch@gmail.com'),
        (3, 'rdanielResearch@gmail.com'),
        (3, 'rowanResearch@gmail.com'),
        (4, 'Simon@gmail.com'),
        (5, 'raoulResearch@gmail.com'),
        (5, 'rdanielResearch@gmail.com');


        INSERT INTO observation (longitude, latitude, date, quantity, notes, meanLongitude,
meanLatitude, scientificName, email, professionalEmail, dateConfirmed)
        VALUES
        ('49.267100', '-123.263500', '2025-10-10', 3, 'Observed several kelp fronds near
shore.', '49.267324', '-123.263471', 'Macrocystis pyrifera', 'johnSmith@gmail.com',
'Simon@gmail.com', '2025-10-11'),
        ('49.330600', '-124.290500', '2025-10-12', 1, 'Single kelp spotted in shallow waters.',
'49.330532', '-124.290541', 'Macrocystis pyrifera', 'johnDoe@gmail.com',
'rachelResearch@gmail.com', '2025-10-13'),
        ('49.405900', '-123.469300', '2025-09-30', 5, 'Healthy kelp growth observed in cove.',
'49.405967', '-123.469368', 'Macrocystis pyrifera', 'gavinKrebbers@gmail.com', NULL, NULL),
        ('49.451900', '-123.326700', '2025-10-01', 2, 'Sparse kelp patches noted.', '49.451951',
'-123.326787', 'Macrocystis pyrifera', 'janeDoe@gmail.com', 'robertResearch@gmail.com',
'2025-10-02'),
        ('49.245100', '-125.257900', '2025-09-28', 8, 'Large kelp forest area thriving.',
'49.245173', '-125.257978', 'Macrocystis pyrifera', 'randomUser@gmail.com',
'Simon@gmail.com', '2025-09-29'),
        ('49.267200', '-123.263480', '2025-10-14', 4, 'Observed kelp growing around sea
anemones.', '49.267324', '-123.263471', 'Macrocystis pyrifera', 'johnDoe@gmail.com',
'rachelResearch@gmail.com', '2025-10-15'),
        ('49.330550', '-124.290510', '2025-10-16', 2, 'Small kelp patches near coral
structures.', '49.330532', '-124.290541', 'Macrocystis pyrifera', 'johnSmith@gmail.com',
NULL, NULL),
        ('49.245120', '-125.257950', '2025-10-17', 6, 'Multiple young kelp fronds sighted in
kelp forest region.', '49.245173', '-125.257978', 'Macrocystis pyrifera',
'gavinKrebbers@gmail.com', 'raoulResearch@gmail.com', '2025-10-18'),
        ('49.405950', '-123.469350', '2025-10-02', 5, 'Several kelp plants found on rocks
during low tide.', '49.405967', '-123.469368', 'Macrocystis pyrifera', 'janeDoe@gmail.com',
'rowanResearch@gmail.com', '2025-10-03'),
        ('49.451930', '-123.326760', '2025-10-04', 3, 'Kelp observed with signs of stress or
damage.', '49.451951', '-123.326787', 'Macrocystis pyrifera', 'johnSmith@gmail.com',
'robertResearch@gmail.com', '2025-10-05'),
        ('49.330520', '-124.290520', '2025-10-06', 7, 'Healthy kelp population noted around
reef area.', '49.330532', '-124.290541', 'Macrocystis pyrifera', 'gavinKrebbers@gmail.com',
'Simon@gmail.com', '2025-10-07');

        INSERT INTO project_observation (projectID, observationID)
        VALUES (5, 1),(5, 2),(5, 3),(5, 4),(5, 5),(1, 6),(1, 7),(1, 8),(3, 9),(3, 10),(3, 11);

        INSERT INTO media (observationID, mediaID, URL, mediaType)
        VALUES
        (1, 1, 'https://example.com/media/kelp_shore_1.jpg', 'image'),
        (1, 2, 'https://example.com/media/kelp_shore_2.mp4', 'video'),
        (2, 1, 'https://example.com/media/kelp_shallow_1.jpg', 'image'),
        (3, 1, 'https://example.com/media/kelp_cove_1.jpg', 'image'),
        (3, 2, 'https://example.com/media/kelp_cove_2.jpg', 'image'),
        (4, 1, 'https://example.com/media/kelp_sparse_1.jpg', 'image'),
        (5, 1, 'https://example.com/media/kelp_forest_1.jpg', 'image'),
        (6, 1, 'https://example.com/media/kelp_anemones_1.jpg', 'image'),
        (7, 1, 'https://example.com/media/kelp_coral_1.jpg', 'image'),
        (8, 1, 'https://example.com/media/kelp_young_1.jpg', 'image'),
        (9, 1, 'https://example.com/media/kelp_rocks_1.jpg', 'image'),
        (10, 1, 'https://example.com/media/kelp_stress_1.jpg', 'image'),
        (11, 1, 'https://example.com/media/kelp_reef_1.jpg', 'image');

        INSERT INTO groupChat (id, `name`, created_at)
        VALUES (0, 'the fish boys', '2025-10-18 14:30:00'),
            (1, 'the pufferfish boys', '2022-10-18 6:30:00'),
            (2, 'the coral boys', '1-10-18 14:30:00'),
            (3, 'empty groupchat', '2025-11-18 14:30:00'),
            (4, 'dead groupchat because two people in the chat got into a heated argument',
'2025-10-18 14:30:00');

        INSERT INTO groupChat_user (email, id)
        VALUES ('Simon@gmail.com', 0),
            ('Simon@gmail.com', 1),
            ('rachelResearch@gmail.com', 1),
            ('randomUser@gmail.com', 0),
            ('robertResearch@gmail.com', 2);

        INSERT INTO message (id, data, time_sent, group_chat_id, email)
        VALUES
        (0, 'hi nice to meet you', '2025-10-18 14:30:00', 0, 'Simon@gmail.com'),
        (1, 'hi not nice to meet you', '2025-10-18 14:30:01', 0, 'rachelResearch@gmail.com'),
        (2, 'aslkdjfalsdkfjlaskjfdlsajfalksjdfasdflkjsakdjflawoeifjwoefjoewfjoewijfoewijf', '2025-10-19 16:36:02', 0, 'johnSmith@gmail.com'),
        (3, 'asdf', '2022-10-18 14:30:00', 0, 'johnDoe@gmail.com'),
        (4, 'I love clownfish, this is the clownfish group', '2025-10-18 14:30:00', 1, 'rachelResearch@gmail.com'),
        (5, 'hello coral porject peopel ', '2025-10-18 15:00:00', 2, 'robertResearch@gmail.com'),
        (6, 'wow the tide sure did change today ', '2025-10-19 09:45:00', 3, 'janeDoe@gmail.com'),
        (7, 'i swa so much kelp you wont beleive it ', '2025-10-19 10:15:00', 4, 'gavinKrebbers@gmail.com'),
        (8, 'i like petting sturgeons', '2025-10-19 11:00:00', 0, 'Simon@gmail.com'),
        (9, 'i found 3 star fish', '2025-10-19 11:30:00', 3, 'johnSmith@gmail.com');
```

AI Acknowledgments:

There were **no AI tools used** for this milestone of the project. This declaration includes the process of brainstorming, researching, writing, diagramming and editing.