CPSC 304 Project Cover Page

Milestone #: 2

Date: 01/03/24

Group Number: 60

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Sangita Dutta	29933983	sdutta06	sangita7dutta@gmail.com
Delsther James Edralin	26279729	dedralin	dedralin@student.ubc.ca
Aman Johal	40801152	amanj21	ajohal21@gmail.com

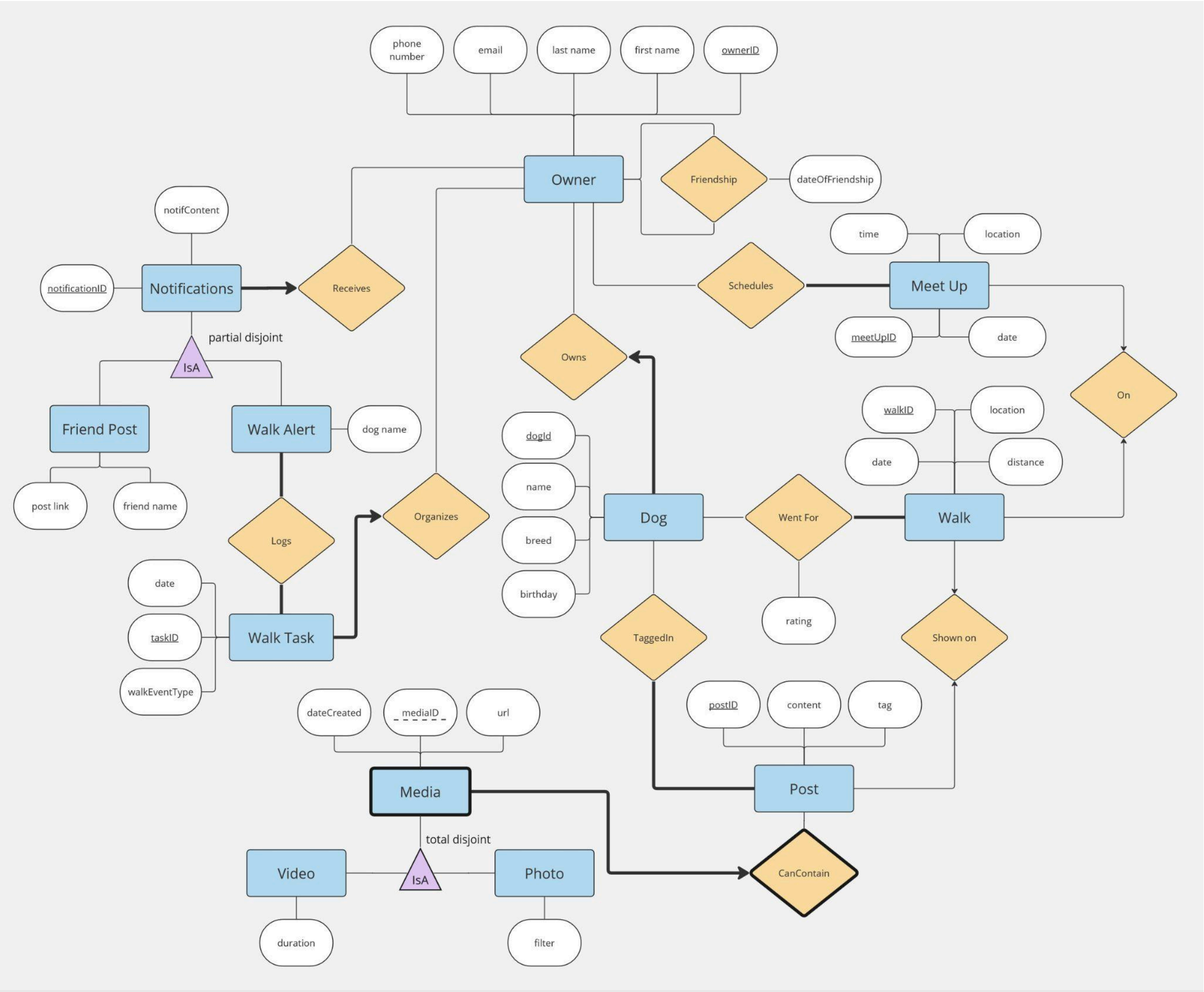
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address,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

2. Project Description

The project is a social platform that manages outdoor, dog walking activities. The domain of this project focuses on “pet wellbeing”, which relates to pet care, recreation, and health. This mainly caters for people that are dog owners and want to guarantee their dogs to receive consistent stimulation by regular walks.

3.a ER Diagram



3.b ER Diagram Revision Notes

- Removed the Friend entity, and its relationship with the Owner entity (Owner-ConnectedWith-Friend) to create a Unary relationship between two owners.
- Added a MeetUp entity to track MeetUps between multiple owners. They can also participate in the same walk.
- Added constraints to the ISA entities.

- Rearranged the Media ISA entity, provided proper attributes, relationship with the Post entity, and the superclass' primary key.

4. Schema

Entity Name	**Table Definition**
Ower	Owner(ownerID: INTEGER, firstName: VARCHAR, lastName: VARCHAR, email: VARCHAR, phoneNumber: INTEGER) firstName NOT NULL email UNIQUE NOT NULL phoneNumber UNIQUE PRIMARY KEY: ownerID CANDIDATE KEY: {email} // avoids multiple Owners for one email
Friendship	Friendship(**ownerID1**: INTEGER, **ownerID2**: INTEGER, dateOfFriendship: DATE) dateOfFriendship NOT NULL PRIMARY KEY: {ownerID1, ownerID2} FOREIGN KEY: ownerID1 REFERENCES Owner(ownerID1) FOREIGN KEY: ownerID2 REFERENCES Owner(ownerID2)
Receives_Notifications	Receives_Notifications(notificationsID: INTEGER, notifContent: VARCHAR, **ownerID**: INTEGER) notifContent NOT NULL ownerID NOT NULL PRIMARY KEY: notificationID FOREIGN KEY: ownerID REFERENCES Owner CANDIDATE KEY: {notifContent, ownerID} // avoids repeating notifs
Friend Post	FriendPost(**notificationID**: INTEGER, postLink: VARCHAR, friendName: VARCHAR) postLink NOT NULL friendName NOT NULL PRIMARY KEY: notificationID FOREIGN KEY: notificationID REFERENCES Receives_Notifications
Walk Alert	WalkAlert(**notificationID**: INTEGER, dogName: VARCHAR) dogName NOT NULL

	PRIMARY KEY: notificationID FOREIGN KEY: notificationID REFERENCES Receives_Notifications
Logs (WalkAlert-Organizes_WalkTask relationship)	Logs(**notificationID:** INTEGER, **taskID**: INTEGER) PRIMARY KEY: (notificationID, taskID) FOREIGN KEY: notificationID references Notifications FOREIGN KEY: taskID references Organizes_WalkTask
Organizes_WalkTask	Organizes_WalkTask(taskID: INTEGER, date: DATE, walkEventType: ENUM, **ownerID**: INTEGER) ownerID NOT NULL PRIMARY KEY: taskID FOREIGN KEY: ownerID REFERENCES Owner
Owns_Dog	Owns_Dog(dogID: INTEGER, name: VARCHAR, breed: VARCHAR, birthday: DATE, **ownerID**: INTEGER) name NOT NULL ownerID NOT NULL PRIMARY KEY: (dogID) FOREIGN KEY: ownerID REFERENCES Owner CANDIDATE KEY: {name, breed, ownerID}, {name, birthday, ownerID} // we assume that no owner has two or more dogs, that has the same breed or birthday, with the same name
Walk	Walk(walkID: INTEGER, location: VARCHAR, date: DATE, distance: FLOAT) location NOT NULL PRIMARY KEY: walkID
WentFor (Dog-Walk relationship)	WentFor(**dogID**: INTEGER, **walkID**: INTEGER, rating: INTEGER) PRIMARY KEY: {dogID, walkID} FOREIGN KEYS: dogID REFERENCES Dog walkID REFERENCES Walk
Post_Walk	Post_Walk(postID: INTEGER, content: VARCHAR, tag: VARCHAR, **walkID**: INTEGER) walkID UNIQUE PRIMARY KEY: postID FOREIGN KEY: walkID REFERENCES Walk

On_MeetUp	On_MeetUp(meetUpID: INTEGER, time: INTEGER, location: VARCHAR, date: DATE, **walkID**: INTEGER) walkID UNIQUE PRIMARY KEY: meetUpID FOREIGN KEY: walkID CANDIDATE KEY: {walkID, date, location, time} // avoids multiple meetups for a walk
Schedules (Owner-On_ MeetUp relationship)	Schedules(**meetUpID**: INTEGER, **ownerID**: INTEGER) PRIMARY KEY: {meetUpID, ownerID} FOREIGN KEYS: meetUpID REFERENCES On_MeetUp ownerID REFERENCES Owner

Post_Media (Weak Entity)	Post_Media(**postID**: INTEGER, mediaID: INTEGER, dateCreated: DATE, url: VARCHAR)
	
	url UNIQUE NOT NULL // avoids bloating of same media content
	
	PRIMARY KEY: {postID, mediaID}
	FOREIGN KEY: postID REFERENCES Walk_Post ON DELETE CASCADE

Video	Video(**mediaID**: INTEGER, duration: INTEGER) PRIMARY KEY: mediaID FOREIGN KEY: mediaID REFERENCES Post_Media
Photo	Photo(**mediaID**: INTEGER, filter: VARCHAR) PRIMARY KEY: mediaID FOREIGN KEY: mediaID REFERENCES Post_Media
TaggedIn (Dog-Post relationship)	TaggedIn(**dogID**: INTEGER, **postID**: INTEGER) PRIMARY KEY: {dogID, postID} FOREIGN KEYS: dogID REFERENCES Dog postID REFERENCES Post_Media

5. Functional Dependencies

Entity	**Functional Dependencies**
Owner	ownerID -> email, firstName, lastName email -> phoneNumber, ownerID //One account per ID, one account per email, one account per phone#
Friendship	ownerID1, ownerID2 -> dateOfFriendship
Receives_Notifications	notificationID -> ownerID, notifContent ownerID, notifContent -> notificationID
FriendPost	notificationID -> postLink postLink -> friendName
WalkAlert	notificationID -> dogName
Logs (WalkAlert-Organizes_Walk Task relationship)	No FDs
Organizes_WalkTask	taskID -> ownerID ownerID, taskID -> date, walkEventType
Owns_Dog	dogID -> ownerID, name, breed, birthday name, breed, ownerID -> dogID, birthday name, birthday, ownerID -> dogID, breed
Walk	walkID -> location walkID, location -> date, distance
WentFor (Dog-Walk relationship)	dogID, walkID -> rating
Post_Walk	postID -> walkID, content, tag postID, tag -> content
On_MeetUp	meetUpID -> time, location, date, walkID walkID, time, location, date -> meetUpID

Schedules (Owner-On_MeetUp relationship)	No FDs
Post_Media (Weak Entity)	postID, mediaID -> dateCreated, url url -> dateCreated, mediaID
Video	mediaID -> duration
Photo	mediaID -> filter
TaggedIn (Dog-Post_Walk relationship)	No FDs

6. Normalization

Entity	Normalized Relations
Owner	ownerID -> email, firstName, lastName email -> phoneNumber, ownerID Minimal cover: ownerID -> email ownerID -> firstName ownerID -> lastName email -> phoneNumber email -> ownerID Minimal keys: {ownerID}, {email} All FDs are in 3NF and BCNF. Owner_1(ownerID, email) Owner_2(**ownerID**, firstName) Owner_3(**ownerID**, lastName) Owner_4(**email**, phoneNumber) Removed: Owner_5(email, ownerID) Foreign Key References: Owner_2 to Owner_4's ownerID references to Owner_1

	Owner 4’s email references to Owner_1
Friendship	ownerID1, ownerID2 -> dateOfFriendship Minimal key: {ownerID1, ownerID2} Already in 3NF and BCNF. Friendship(**ownerID1**, **ownerID2**, dateOfFriendship) Foreign Key References: ownerID1 references to Owner_1(ownerID1) ownerID2 references to Owner_1(ownerID2)
Receives_Notifications	Already in 3NF and BCNF. Receives_Notifications(notificationID, **ownerID**, notifContent) Foreign Key References: ownerID references to Owner_1
FriendPost	notificationID -> postLink postLink -> friendName // removes repeating links Minimal key: {notificationID} notificationID -> postLink is already in 3NF and BCNF. FriendPost_1(notificationID, postLink) postLink -> friendName violates BCNF. Decompose. FriendPost_2(postLink, friendName) FriendPost_1(**notificationID**, postLink) FriendPost_2(**postLink**, friendName) Foreign Key References: FriendPost_1’s notificationID references to Receives_Notifications FriendPost_2’s postLink references to FriendPost_1
Walk Alert	notificationID -> dogName Minimal key: {notificationID}

	Already in BCNF and 3NF. WalkAlert(**notificationID**, dogName) Foreign Key References: notificationID references to Receives_Notifications
Logs (WalkAlert-Organizes_Walk Task relationship)	No FDs. Logs(**notificationID**, **taskID**) Foreign Key References: notificationID references to Receives_Notifications taskID references to Organizes_WalkTask
Organizes_WalkTask	taskID -> ownerID ownerID, taskID -> date, walkEventType Minimal key: {taskID} Already in BCNF. Organizes_WalkTask(taskID, **ownerID**, date, walkEventType) ownerID references to Owner_1
Owns_Dog	dogID -> ownerID, name, breed, birthday name, breed, ownerID -> dogID, birthday name, birthday, ownerID -> dogID, breed Minimal key: {dogID} Candidate keys: {name, breed, ownerID}, {name, birthday, ownerID} Minimal cover: dogID -> ownerID dogID -> name dogID ->breed dogID -> birthday name, breed, ownerID -> dogID name, birthday, ownerID -> dogID All FDs are in BCNF and 3NF.

	Owns_Dog_1(name, breed, **ownerID**, dogID) Owns_Dog_2(**name**, birthday, **ownerID**, **dogID**) Removed: Owns_Dog_3(dogID, ownerID) Owns_Dog_4(dogID, name) Owns_Dog_3(dogID, breed) Foreign Key References: Owns_Dog_1's ownerID references to Owner_1 Owns_Dog_2's ownerID references to Owner_1 Owns_Dog_2's name references to Owns_Dog_1 Owns_Dog_2's dogID references to Owns_Dog_1
Walk	walkID -> location walkID, location -> date, distance Minimal key: {walkID} Minimal cover: walkID -> location walkID -> date walkID -> distance Already in BCNF and 3NF. Walk_1(walkID, location) Walk_2(**walkID**, date) Walk_3(**walkID**, distance) Foreign Key References: Walk_2's walkID references to Walk_1 Walk_3's walkID references to Walk_1
WentFor (Dog-Walk relationship)	dogID, walkID -> rating Minimal key: {dogID, walkID} Already in BCNF and 3NF.

	WentFor(**dogID**, **walkID**, rating) Foreign Key References: dogID references to Owns_Dog_1 walkID references to Walk_1
Post_Walk	postID -> walkID, content, tag postID, tag -> content Minimal key: {postID} Minimal cover: postID -> walkID postID -> content postID -> tag Already in BCNF and 3NF. Post_Walk_1(postID, **walkID**) Post_Walk_2(**postID**, content) Post_Walk_3(**postID**, tag) Foreign Key References: Post_Walk_1's walkID references Walk_1 Post_Walk_2's postID references Post_Walk_1 Post_Walk_3's postID references Post_Walk_1
On_MeetUp	meetUpID -> time, location, date, walkID walkID, time, location, date -> meetUpID Minimal key: {meetUpID} Minimal cover: meetUpID -> time meetUpID -> location meetUpID -> date meetUpID -> walkID walkID, time, location, date -> meetUpID Already in BCNF and 3NF.

	On_MeetUp(**walkID**, time, location, date, meetUpID)
	
	Removed:
	On_MeetUp_1(meetUpID, time)
	On_MeetUp_2(meetUpID, location)
	On_MeetUp_3(meetUpID, date)
	On_MeetUp_4(meetUpID, walkID)
	
	Foreign Key References:
	walkID references Walk_1

Schedules (Owner-On_MeetUp relationship)	No FDs. Schedules(**meetUpID**, **ownerID**) Foreign Key References: meetUpID references On_MeetUp ownerID references Owner_1
Post_Media (Weak Entity)	postID, mediaID -> dateCreated, url url -> dateCreated, mediaID // removes repeating urls Minimal key: {postID} Minimal cover: postID, mediaID -> url url -> dateCreated url -> mediaID Post_Media_1(**postID**, mediaID, url) Post_Media_2(**url**, dateCreated) Removed: Post_Media_3(url, mediaID) Foreign Key References: Post_Media_1's postID references Post_Walk_1 Post_Media_2's url references Post_Media_1
Video	mediaID -> duration Minimal key: {mediaID}

	Already in BCNF and 3NF. Video(**mediaID**, duration) Foreign Key References: mediaID references Post_Media_1
Photo	mediaID -> filter Minimal key: {mediaID} Already in BCNF and 3NF. Photo(**mediaID**, filter) Foreign Key References: mediaID references Post_Media_1
TaggedIn (Dog-Post_Walk relationship)	No FDs. TaggedIn(**dogID**, **postID**) Foreign Key References: dogID references Owns_Dog_1 postID references Post_Walk_1

7. SQL DDL Statements

Owner	CREATE TABLE Owner_1 (ownerID INTEGER PRIMARY KEY, email VARCHAR UNIQUE NOT NULL,); CREATE TABLE Owner_2 (ownerID INTEGER PRIMARY KEY, firstName VARCHAR NOT NULL, FOREIGN KEY ownerID REFERENCES Owner_1 ON DELETE CASCADE); CREATE TABLE Owner_3 (ownerID INTEGER PRIMARY KEY, lastName VARCHAR NOT NULL, FOREIGN KEY ownerID REFERENCES Owner_1 ON DELETE CASCADE); CREATE TABLE Owner_4 (email INTEGER PRIMARY KEY, phoneNumber VARCHAR UNIQUE, FOREIGN KEY email REFERENCES Owner_1 ON DELETE CASCADE);
Friendship	CREATE TABLE Friendship(ownerID1 INTEGER, ownerID2 INTEGER, dateOfFriendship DATE NOT NULL, PRIMARY KEY (ownerID1, ownerID2) FOREIGN KEY ownerID1 REFERENCES Owner_1(ownerID1), FOREIGN KEY ownerID2 REFERENCES Owner_1(ownerID2));
Receives_Notifications	CREATE TABLE Receives_Notifications(notificationID INTEGER PRIMARY KEY, ownerID INTEGER NOT NULL, notifContent VARCHAR, FOREIGN KEY ownerID REFERENCES Owner_1 ON DELETE NO ACTION ON UPDATE CASCADE);

FriendPost	CREATE TABLE FriendPost_1(notificationID INTEGER PRIMARY KEY, postLink VARCHAR UNIQUE NOT NULL, FOREIGN KEY notificationID REFERENCES Receives_Notifications ON DELETE CASCADE); CREATE TABLE FriendPost_2(postLink PRIMARY KEY, friendName VARCHAR NOT NULL, postLink REFERENCES FriendPost_1 ON DELETE CASCADE);
WalkAlert	CREATE TABLE WalkAlert (notificationID PRIMARY KEY, dogName VARCHAR NOT NULL, FOREIGN KEY notificationID REFERENCES Receives_Notifications ON DELETE CASCADE);
Logs	CREATE TABLE Logs (notificationID INTEGER, taskID INTEGER, PRIMARY KEY (notificationID, taskID), FOREIGN KEY notificationID REFERENCES Receives_ Notifications, FOREIGN KEY taskID REFERENCES Organizes_WalkTask);
Organizes_WalkTask	CREATE TABLE Organizes_WalkTask (taskID INTEGER PRIMARY KEY, ownerID INTEGER NOT NULL, date DATE, walkEventType ENUM, FOREIGN KEY ownerID REFERENCES Owner_1 ON DELETE NO ACTION ON UPDATE CASCADE);
Owns_Dog	CREATE TABLE Owns_Dog_1(dogID INTEGER PRIMARY KEY, name VARCHAR NOT NULL, breed VARCHAR, ownerID INTEGER NOT NULL, UNIQUE (name, ownerID, breed), FOREIGN KEY ownerID References Owner_1 ON DELETE NO ACTION

	ON UPDATE CASCADE); CREATE TABLE Owns_Dog_2(dogID INTEGER PRIMARY KEY, name VARCHAR NOT NULL, birthday DATE, ownerID INTEGER NOT NULL, UNIQUE (name, ownerID, birthday), FOREIGN KEY ownerID References Owner_1 ON DELETE NO ACTION ON UPDATE CASCADE, FOREIGN KEY dogID References Owns_Dog_1 ON DELETE CASCADE, FOREIGN KEY name References Owns_Dog_1);
Walk	CREATE TABLE Walk_1(walkID INTEGER PRIMARY KEY, location VARCHAR NOT NULL,); CREATE TABLE Walk_2(walkID INTEGER PRIMARY KEY date DATE, FOREIGN KEY walkID REFERENCES Walk_1 ON DELETE CASCADE); CREATE TABLE Walk_3(walkID INTEGER PRIMARY KEY, distance FLOAT, FOREIGN KEY walkID REFERENCES Walk_1 ON DELETE CASCADE);
WentFor	CREATE TABLE WentFor(dogID INTEGER, walkID INTEGER, rating VARCHAR, PRIMARY KEY (dogID, walkID), FOREIGN KEY dogID REFERENCES Owns_Dog_1, FOREIGN KEY walkID REFERENCES Walk_1);
Post_Walk	CREATE TABLE Post_Walk_1(postID INTEGER PRIMARY KEY,

	walkID INTEGER, FOREIGN KEY walkID REFERENCES Walk_1); CREATE TABLE Post_Walk_2(postID INTEGER PRIMARY KEY, content VARCHAR, FOREIGN KEY postID REFERENCES Post_Walk_1 ON DELETE CASCADE); CREATE TABLE Post_Walk_3(postID INTEGER PRIMARY KEY, tag VARCHAR, FOREIGN KEY postID REFERENCES Post_Walk_1 ON DELETE CASCADE);
On_MeetUp	CREATE TABLE On_MeetUp(meetUpID INTEGER PRIMARY KEY, walkID INTEGER, time DATETIME, location VARCHAR, date DATE, FOREIGN KEY walkID REFERENCES Walk_1 ON DELETE NO ACTION ON UPDATE CASCADE);
Schedules	CREATE TABLE Schedules(meetUpID INTEGER, ownerID INTEGER, PRIMARY KEY(meetUPID, ownerID), FOREIGN KEY meetUpID REFERENCES On_MeetUp, FOREIGN KEY ownerID REFERENCES Owner_1 ON DELETE NO ACTION ON UPDATE CASCADE);
Post_Media	CREATE TABLE Post_Media_1(postID INTEGER, mediaID INTEGER, url VARCHAR UNIQUE NOT NULL, PRIMARY KEY (postID, mediaID) FOREIGN KEY postID REFERENCES Post_Walk_1 ON DELETE CASCADE);

	CREATE TABLE Post_Media_2(url VARCHAR PRIMARY KEY, dateCreated DATE, FOREIGN KEY url REFERENCES Post_Media_1 ON DELETE CASCADE);
Video	CREATE TABLE Video(mediaID INTEGER PRIMARY KEY, duration TIME, FOREIGN KEY mediaID REFERENCES Post_Media_1 ON DELETE CASCADE);
Photo	CREATE TABLE Photo(mediaID INTEGER PRIMARY KEY, filter VARCHAR, FOREIGN KEY mediaID REFERENCES Post_Media_1 ON DELETE CASCADE);
TaggedIn	CREATE TABLE TaggedIn(dogID INTEGER, postID INTEGER, PRIMARY KEY (dogID, postID), FOREIGN KEY dogID REFERENCES Owns_Dog_1, FOREIGN KEY postID REFERENCES Post_Walk_1);

8. Tuples

Owner_1	

1	INSERT
	INTO Owner_1(ownerID, email)
	VALUES (1, ‘ajohal21@gmail.com’)

2	INSERT
	INTO Owner_1(ownerID, email)
	VALUES (2, ‘edralin@gmail.com’)

3	INSERT
	INTO Owner_1(ownerID, email)
	VALUES (3, ‘sangita@gmail.com’)

4	INSERT
	INTO Owner_1(ownerID, email)
	VALUES (4, ‘spiderman@gmail.com’)

5	INSERT
	INTO Owner_1(ownerID, email)
	VALUES (5, ‘tswift@ubc.ca’)

Owner_2	

1	INSERT
	INTO Owner_2(ownerID, firstName)
	VALUES (1, ‘Aman’)

2	INSERT
	INTO Owner_2(ownerID, firstName)
	VALUES (2, ‘James’)

3	INSERT
	INTO Owner_2(ownerID, firstName)
	VALUES (3, ‘Sangita’)

4	INSERT
	INTO Owner_2(ownerID, firstName)
	VALUES (4, ‘Peter’)

5	INSERT
	INTO Owner_2(ownerID, firstName)
	VALUES (5, ‘Taylor’)

Owner_3	
1	INSERT

	INTO Owner_3(ownerID, lastName) VALUES (5, ‘Johal’)

2	INSERT
	INTO Owner_3(ownerID, lastName)
	VALUES (5, ‘Edralin’)

3	INSERT
	INTO Owner_3(ownerID, lastName)
	VALUES (5, ‘Dutta’)

4	INSERT
	INTO Owner_3(ownerID, lastName)
	VALUES (5, ‘Parkerr’)

5	INSERT
	INTO Owner_3(ownerID, lastName)
	VALUES (5, ‘Swift’)

Owner_4	

1	INSERT
	INTO Owner_4(email, phoneNumber)
	VALUES (‘ajohal21@gmail.com’, 6049103824)

2	INSERT
	INTO Owner_4(email, phoneNumber)
	VALUES (‘edralin@gmail.com’, 7785559403)

3	INSERT
	INTO Owner_4(email, phoneNumber)
	VALUES (‘dutta@gmail.com’, 6044921020)

4	INSERT
	INTO Owner_4(email, phoneNumber)
	VALUES (‘parker@gmail.com’, 4390456831)

5	INSERT
	INTO Owner_4(email, phoneNumber)
	VALUES (‘tswift@ubc.ca’, 6042904822)

Friendship	

1	INSERT
	INTO Friendship(ownerID1, ownerID2, dateOfFrienship)
	VALUES (1,2, ‘2022-09-09’)

2	INSERT
	INTO Friendship(ownerID1, ownerID2, dateOfFrienship)
	VALUES (1,3, ‘2022-10-01’)

3	INSERT
	INTO Friendship(ownerID1, ownerID2, dateOfFrienship)
	VALUES (5,3, ‘2024-01-11’)

4	INSERT
	INTO Friendship(ownerID1, ownerID2, dateOfFrienship)
	VALUES (2,4, ‘2023-07-09’)

5	INSERT
	INTO Friendship(ownerID1, ownerID2, dateOfFrienship)
	VALUES (1,5, ‘2024-02-10’)

Receives_Notifications	

1	INSERT
	INTO Receives_Notification(notificationsID, notifContent, ownerID)
	VALUES (1, ‘New Post from James Edralin’, 1)

2	INSERT
	INTO Receives_Notification(notificationsID, notifContent, ownerID)
	VALUES (2, ‘New Post’, 4)

3	INSERT
	INTO Receives_Notification(notificationsID, notifContent, ownerID)
	VALUES (3, ‘New Post from Aman Johal’, 5)

4	INSERT
	INTO Receives_Notification(notificationsID, notifContent, ownerID)
	VALUES (4, ‘New Post from Peter Parker’, 5)

5	INSERT
	INTO Receives_Notification(notificationsID, notifContent, ownerID)
	VALUES (5, ‘New Post from Taylor Swift’, 2)

6	INSERT INTO Receives_Notification(notificationsID, notifContent, ownerID)

	VALUES (6, ‘Walk Alert’, 2)

7	INSERT
	INTO Receives_Notification(notificationsID, notifContent, ownerID)
	VALUES (6, ‘Walk Alert for May’, 5)

8	INSERT
	INTO Receives_Notification(notificationsID, notifContent, ownerID)
	VALUES (7, ‘Walk Alert for Coda, 4)

9	INSERT
	INTO Receives_Notification(notificationsID, notifContent, ownerID)
	VALUES (8, ‘Walk Alert’, 1)

10	INSERT
	INTO Receives_Notification(notificationsID, notifContent, ownerID)
	VALUES (8, ‘Walk Alert for Murphy’, 3)

11	INSERT
	INTO Receives_Notification(notificationsID, notifContent, ownerID)
	VALUES (11, ‘Normal Notification’, 3)

FriendPost_1	

1	INSERT
	INTO FriendPost_1(notificationID, postLink)
	VALUES (1, ‘www.example.com/example1’)

2	INSERT
	INTO FriendPost_1(notificationID, postLink)
	VALUES (2, ‘www.example.com/example2’)

3	INSERT
	INTO FriendPost_1(notificationID, postLink)
	VALUES (3, ‘www.example.com/example3’)

4	INSERT
	INTO FriendPost_1(notificationID, postLink)
	VALUES (4, ‘www.example.com/example4’)

5	INSERT
	INTO FriendPost_1(notificationID, postLink)
	VALUES (5, ‘www.example.com/example5’)

FriendPost_2	

1	INSERT
	INTO FriendPost_2(postLink, friendName)
	VALUES (‘www.example.com/example1’, ‘James Edralin’)

2	INSERT
	INTO FriendPost_2(postLink, friendName)
	VALUES (‘www.example.com/example2’, ‘Aman Johal’)

3	INSERT
	INTO FriendPost_2(postLink, friendName)
	VALUES (‘www.example.com/example3’, ‘Aman Johal’)

4	INSERT
	INTO FriendPost_2(postLink, friendName)
	VALUES (‘www.example.com/example4’, ‘Peter Parker’)

5	INSERT
	INTO FriendPost_2(postLink, friendName)
	VALUES (‘www.example.com/example5’, ‘Taylor Swift')

Walk_Alert	

1	INSERT
	INTO WalkAlert(notificationID, dogName)
	VALUES (6, ‘Eugene’)

2	INSERT
	INTO WalkAlert(notificationID, dogName)
	VALUES (7, ‘May’)

3	INSERT
	INTO WalkAlert(notificationID, dogName)
	VALUES (8, ‘Coda’)

4	INSERT
	INTO WalkAlert(notificationID, dogName)
	VALUES (9, ‘Kobe’)

5	INSERT
	INTO WalkAlert(notificationID, dogName)
	VALUES (10, ‘Murphy’)

Logs

1	INSERT
	INTO Logs(notificationID, taskID)
	VALUES (6, 1)

2	INSERT
	INTO Logs(notificationID, taskID)
	VALUES (7, 2)

3	INSERT
	INTO Logs(notificationID, taskID)
	VALUES (9, 3)

4	INSERT
	INTO Logs(notificationID, taskID)
	VALUES (8, 5)

5	INSERT
	INTO Logs(notificationID, taskID)
	VALUES (10, 4)

Organizes_WalkTask	

1	INSERT
	INTO Organizes_WalkTask(taskID, date, walkEventType, ownerID)
	VALUES (1, ‘2024-02-14’, ‘Run’, 2)

2	INSERT
	INTO Organizes_WalkTask(taskID, date, walkEventType, ownerID)
	VALUES (2, ‘2024-01-31’, ‘Walk’, 5)

3	INSERT
	INTO Organizes_WalkTask(taskID, date, walkEventType, ownerID)
	VALUES (3, ‘2024-02-22’, ‘Walk’, 3)

4	INSERT
	INTO Organizes_WalkTask(taskID, date, walkEventType, ownerID)
	VALUES (4, ‘2023-12-25’, ‘Run’, 1)

5	INSERT
	INTO Organizes_WalkTask(taskID, date, walkEventType, ownerID)
	VALUES (5, ‘2024-01-16’, ‘Run’, 4)

6	INSERT
	INTO Organizes_WalkTask(taskID, date, walkEventType, ownerID)
	VALUES (6, ‘2024-01-16’, ‘Run’, 4)

7	INSERT

	INTO Organizes_WalkTask(taskID, date, walkEventType, ownerID) VALUES (6, NULL, NULL, 4)

Owns_Dog_1	

1	INSERT
	INTO Owns_Dog_1(name, breed, ownerID, dogID)
	VALUES (‘Eugene’, ‘Whoodle’, 1, 1)

2	INSERT
	INTO Owns_Dog_1(name, breed, ownerID, dogID)
	VALUES (‘May’, NULL, 2, 2)

3	INSERT
	INTO Owns_Dog_1(name, breed, ownerID, dogID)
	VALUES (‘Kobe, ‘Pug’, 3, 3)

4	INSERT
	INTO Owns_Dog_1(name, breed, ownerID, dogID)
	VALUES (‘Kobe’, ‘Mutt’, 4, 4)

5	INSERT
	INTO Owns_Dog_1(name, breed, ownerID, dogID)
	VALUES (‘Murphy’, ‘Boykin Spaniel’, 1, 5)

Owns_Dog_2	

1	INSERT
	INTO Owns_Dog_2(name, birthday, ownerID, dogID)
	VALUES (‘Eugene’, ‘2022-05-12’, 1, 1)

2	INSERT
	INTO Owns_Dog_2(name, birthday, ownerID, dogID)
	VALUES (‘May’, NULL, 2,2)

3	INSERT
	INTO Owns_Dog_2(name, birthday, ownerID, dogID)
	VALUES (‘Kobe’, ‘2021-09-23’, 3, 3)

4	INSERT
	INTO Owns_Dog_2(name, birthday, ownerID, dogID)
	VALUES (‘Kobe’, ‘2021-09-23’, 4, 4)

5	INSERT INTO Owns_Dog_2(name, birthday, ownerID, dogID)

	VALUES (‘Murphy’, ‘2023-04-04’, 1, 5)

Walk_1	
1	INSERT INTO Walk_1(walkID, location) Values (1, ‘Victoria Park’)
2	INSERT INTO Walk_1(walkID, location) Values (2, ‘Pacific Spirit Park’)
3	INSERT INTO Walk_1(walkID, location) Values (3, ‘ABC Park’)
4	INSERT INTO Walk_1(walkID, location) Values (4, ‘The Local Pond’)
5	INSERT INTO Walk_1(walkID, location) Values (5, ‘Around The Block’)
Walk_2	
1	INSERT INTO Walk_2(walkID,date) Values (1, ‘2024-02-28’)
2	INSERT INTO Walk_2(walkID,date) Values (2, NULL)
3	INSERT INTO Walk_2(walkID,date) Values (3, ‘2024-02-11’)
4	INSERT INTO Walk_2(walkID,date) Values (4, ‘2024-02-11’)
5	INSERT INTO Walk_2(walkID,date)

	Values (5, ‘2024-02-26’)
Walk_3	
1	INSERT INTO Walk_3(walkID, distance) Values (1, 2.5)
2	INSERT INTO Walk_3(walkID, distance) Values (2, NULL)
3	INSERT INTO Walk_3(walkID, distance) Values (3, 1)
4	INSERT INTO Walk_3(walkID, distance) Values (4, 1)
5	INSERT INTO Walk_3(walkID, distance) Values (5, 0.9)

WentFor	

1	INSERT
	INTO WentFor(dogId, walkID, rating)
	VALUES (1, 3, 5)

2	INSERT
	INTO WentFor(dogId, walkID, rating)
	VALUES (1, 1, 5)

3	INSERT
	INTO WentFor(dogId, walkID, rating)
	VALUES (5, 2, 2)

4	INSERT
	INTO WentFor(dogId, walkID, rating)
	VALUES (4, 4, NULL)

5	INSERT INTO WentFor(dogId, walkID, rating)

	VALUES (3, 5, 1)

Post_Walk_1	

1	INSERT
	INTO Post_Walk_1(postID, walkID)
	VALUES (1, 2)

2	INSERT
	INTO Post_Walk_1(postID, walkID)
	VALUES (3, 4)

3	INSERT
	INTO Post_Walk_1(postID, walkID)
	VALUES (42, 1)

4	INSERT
	INTO Post_Walk_1(postID, walkID)
	VALUES (1123, 3)

5	INSERT
	INTO Post_Walk_1(postID, walkID)
	VALUES (234231, 5)

6	INSERT
	INTO Post_Walk_1(postID, walkID)
	VALUES (231, 5)

7	INSERT
	INTO Post_Walk_1(postID, walkID)
	VALUES (232, NULL)

Post_Walk_2	
1	INSERT INTO Post_Walk_2(postID, content) VALUES (1, ‘Today was a good day’,)

2	INSERT
	INTO Post_Walk_3(postID, content)
	VALUES (3, ‘Raining Day’)

3	INSERT
	INTO Post_Walk_3(postID, content)
	VALUES (42, ‘Chilled at Park’)

4	INSERT
	INTO Post_Walk_3(postID, content)
	VALUES (1123, ‘Pond Walk’)

5	INSERT
	INTO Post_Walk_3(postID, content)
	VALUES (234231, ‘Chased a duck’)

6	INSERT
	INTO Post_Walk_3(postID, content)
	VALUES (231, NULL)

7	INSERT
	INTO Post_Walk_3(postID, content)
	VALUES (232, ‘Raining Day’)

Post_Walk_3	

1	INSERT
	INTO Post_Walk_3(postID, tag)
	VALUES (1, ‘#SawASquirrel')

2	INSERT
	INTO Post_Walk_3(postID, tag)
	VALUES (3, ‘#SunOutBumOut’)

3	INSERT
	INTO Post_Walk_3(postID, tag)
	VALUES (42, ‘#ChillDay’)

4	INSERT
	INTO Post_Walk_3(postID, tag)
	VALUES (1123, ‘ #FYP’)

5	INSERT
	INTO Post_Walk_3(postID, tag)
	VALUES (234231, ‘#Whoooo’)

6	INSERT
	INTO Post_Walk_3(postID, tag)
	VALUES (231, NULL)

5	INSERT
	INTO Post_Walk_3(postID, tag)
	VALUES (232, ‘#Whoooo’)

On_MeetUp

1	INSERT
	INTO On_MeetUp(meetUpID, time, location, date, walkID)
	VALUES (1, ‘12:00p.m.’, ‘Berczy Park’, ‘Feb-20-2024’, 2)

2	INSERT
	INTO On_MeetUp(meetUpID, time, location, date, walkID)
	VALUES (2, ‘NULL, ‘Centennial Station’, ‘Feb-22-2028’, 5)

3	INSERT
	INTO On_MeetUp(meetUpID, time, location, date, walkID))
	VALUES (3, ‘2:30p.m.’, NULL, ‘Mar-02-2024’, 1)

4	INSERT
	INTO On_MeetUp(meetUpID, time, location, date, walkID)
	VALUES (4, ‘1:40p.m.’, ‘Markville Mall’, NULL, 3)

5	INSERT
	INTO On_MeetUp(meetUpID, time, location, date, walkID)
	VALUES (5, ‘12:23p.m.’, ‘DavidLam Park’, ‘Apr-20-2025’, NULL)

6	INSERT
	INTO On_MeetUp(meetUpID, time, location, date, walkID)
	VALUES (6, NULL, NULL, NULL, NULL)

Schedules	

1	INSERT
	INTO Schedules(meetUpID, ownerID)
	VALUES (2, 3)

2	INSERT
	INTO Schedules(meetUpID, ownerID)
	VALUES (4, 1)

3	INSERT
	INTO Schedules(meetUpID, ownerID)
	VALUES (5, 5)

4	INSERT
	INTO Schedules(meetUpID, ownerID)
	VALUES (1, 2)

5	INSERT
	INTO Schedules(meetUpID, ownerID)
	VALUES (3, 4)

6	INSERT

	INTO Schedules(meetUpID, ownerID) VALUES (6, 1)

Post_Media_1	

1	INSERT
	INTO Post_Media_1(postID, mediaID, url)
	VALUES (132323, 43213, ‘123.png’)

2	INSERT
	INTO Post_Media_1(postID, mediaID, url)
	VALUES (35232, 34234, ‘hello_world.png’)

3	INSERT
	INTO Post_Media_1(postID, mediaID, url)
	VALUES (56456456, 2123123, ‘hi.mov’)

4	INSERT
	INTO Post_Media_1(postID, mediaID, url)
	VALUES (542324, 12341324, ‘456.png’)

5	INSERT
	INTO Post_Media_1(postID, mediaID, url)
	VALUES (657657234, 1234, ‘another_dog.png’)

Post_Media_2	

1	INSERT
	INTO Post_Media_2(url, dateCreated)
	VALUES (‘123.png’, Feb-20-2023)

2	INSERT
	INTO Post_Media_2(url, dateCreated)
	VALUES (‘hello_world.png’, Feb-23-2033)

3	INSERT
	INTO Post_Media_2(url, dateCreated)
	VALUES (‘hi.mov’, Jan-20-2043)

4	INSERT
	INTO Post_Media_2(url, dateCreated)
	VALUES (‘456.png’, Jan-20-2043)

5	INSERT INTO Post_Media_2(url, dateCreated)

	VALUES (‘another_dog.png’, NULL)

Video	

1	INSERT
	INTO Video(mediaID, duration)
	VALUES (1, 01:22)

2	INSERT
	INTO Video(mediaID, duration)
	VALUES (2, 01:22)

3	INSERT
	INTO Video(mediaID, duration)
	VALUES (3, NULL)

4	INSERT
	INTO Video(mediaID, duration)
	VALUES (4, 00:13)

5	INSERT
	INTO Video(mediaID, duration)
	VALUES (5, 02:28)

Photo	

1	INSERT
	INTO Video(mediaID, filter)
	VALUES (1, sepia)

2	INSERT
	INTO Video(mediaID, filter)
	VALUES (2, original)

3	INSERT
	INTO Video(mediaID, filter)
	VALUES (3, grayscale)

4	INSERT
	INTO Video(mediaID, filter)
	VALUES (4, NULL)

5	INSERT INTO Video(mediaID, filter)

	VALUES (5, original)

TaggedIn	

1	INSERT
	INTO TaggedIn(dogID, postID)
	VALUES (1, 1)

2	INSERT
	INTO TaggedIn(dogID, postID)
	VALUES (2, 2)

3	INSERT
	INTO TaggedIn(dogID, postID)
	VALUES (2, 1)

4	INSERT
	INTO TaggedIn(dogID, postID)
	VALUES (4, 5)

5	INSERT
	INTO TaggedIn(dogID, postID)
	VALUES (3, 3)