University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: October 20, 2023

Group Number: 25

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Ziven Anderson	57820847	v1a2s	zivenanderson@msn.com
Emilie Ma	62226386	h4i7d	kewbish@gmail.com
Youssef Saleh	74023656	z0g7v	ymseam@hotmail.com

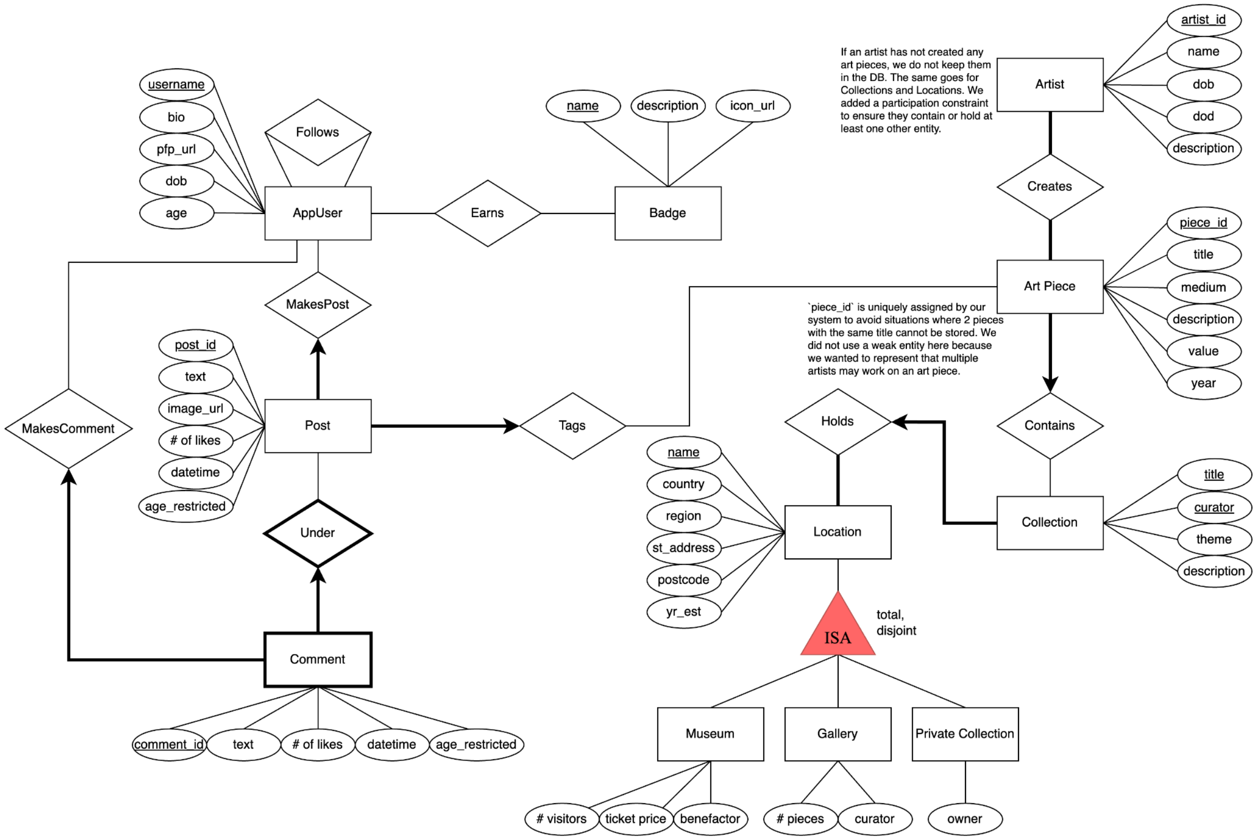
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address,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

Our project is to create a social media website where users can post, discuss, and learn about art in various museums, galleries, and private collections around the world. There will be a typical social media dynamic of users, following, posts, comments, badges, etc as well as a structure around artworks, artists, and the places and collections art is displayed in. The idea is users will be able to experience the app in two ways - one in a more interactive way, viewing a curated feed of content the people they follow have posted about various artworks, and another experience where they can search various parameters about artists, museums, and artworks to gather a collection of information and insights others have posted about them.

ER Diagram



If an artist has not created any art pieces, we do not keep them in the DB. The same goes for Collections and Locations. We added a participation constraint to ensure they contain or hold at least one other entity.

`piece_id` is uniquely assigned by our system to avoid situations where 2 pieces with the same title cannot be stored. We did not use a weak entity here because we wanted to represent that multiple artists may work on an art piece.

ER Diagram Changelog:

1. Badge: Earns changed to many to many because we've now decided multiple AppUsers can earn the same Badge, as opposed to having a separate Badge row for each instance that a AppUser earns. However, they may not necessarily have earned a Badge. Removed badge_id as individual badges should have unique names.
2. Art Piece: Added `year` attribute for more information.
3. Artist: Changed name from Artists to Artist.

4. Location: Added `name`, which will serve as the PK, because we assume Museums / Galleries have unique names. Added `region`, `st_address`, `postcode` for more granularity in location.
5. AppUser: Renamed 'User' to 'AppUser' to avoid conflict w/ Oracle reserved keywords. Added `dob`, `age` to support our 'age_restricted' functionality.
6. Post and Comment: Added `age_restricted` to support discussion of art that may not be appropriate for a general audience.
7. Added Makes relation between AppUser and Comment, to support linking between a Comment and the AppUser that created it.

Relational Schema

AppUser(*username:* char[30], *bio:* char[120], *pfp_url:* char[100], *dob:* date, *age:* integer)

- Primary key: {username}
- Candidate keys: {username}
- pfp_url, dob, age are NOT NULL
- pfp_url has a default value of a link to a default image
- FDs: {username}→{bio, pfp_url, dob, age}, {dob}→{age}

Follows(***follower****:* char[30], ***followee****:* char[30])

- Primary key: {follower, followee}
- Candidate keys: {follower, followee}
- Foreign keys: {follower} (references AppUser), {followee} (references AppUser)
- Require that the follower and followee are not the same person

Post(*post_id:* integer, *text:* char[120], *image_url:* char[100], *num_likes:* integer, *datetime:* datetime, *age_restricted:* number[1], ***username****:* char[30], ***piece_id****:* integer)

- Primary key: {post_id}
- Candidate keys: {post_id}, {image_url}, {username, datetime}
- Foreign keys: {username} (references AppUser), {piece_id} (references ArtPiece)
- image_url is UNIQUE, {username, datetime} is UNIQUE
- num_likes, text, username, piece_id, datetime, age_restricted are NOT NULL
- num_likes and age_restricted default to 0
- FDs: {post_id} → {text, image_url, num_likes, datetime, age_restricted, username, piece_id}, {image_url}→{text, post_id, num_likes, datetime, age_restricted, username, piece_id}, {username, datetime}→{text, post_id, image_url, num_likes, age_restricted, piece_id}

Comment(*comment_id:* integer, *text:* char[120], *num_likes:* integer, *age_restricted:* number[1], *datetime:* datetime, ***username:*** char[30], ***post_id:*** integer)

- Primary key: {comment_id, post_id}
- Candidate keys: {comment_id, post_id}
- Foreign keys: {post_id} (references Post), {username} (references AppUser)
- text, num_likes, datetime, age_restricted, username are NOT NULL
- num_likes and age_restricted default to 0
- FDs: {comment_id, post_id}→{text, num_likes, age_restricted, datetime, username}

Badge(*name:* char[30], *description:* char[120], *icon_url:* char[100])

- Primary key: {name}
- Candidate keys: {name}, {icon_url}, {description}
- name, icon_url, description are UNIQUE and NOT NULL
- FDs: {name} → {icon_url, description}, {icon_url}→{name, description}, {description}→{name, icon_url}

Earns(***username:*** char[30], ***badge_name:*** char[30])

- Primary key: {username, badge_name}
- Candidate keys: {username, badge_name}
- Foreign keys: {username} (references AppUser), {badge_name} (references Badge)

Artist(*artist_id:* integer, *name:* char[30], *dob:* date, *dod:* date, *description:* char[120])

- Primary key: {artist_id}
- Candidate keys: {artist_id}
- name is NOT NULL
- FDs: {artist_id} → {name, dob, dod, description}

Creates(***artist_id:*** integer, ***piece_id:*** integer)

- Primary key: {artist_id, piece_id}
- Candidate keys: {artist_id, piece_id}
- Foreign keys: {artist_id} (references Artist), {piece_id} (references ArtPiece)
- Should have a total participation constraint, but this cannot be represented in SQL DDL without assertions

ArtPiece(*piece_id:* integer, *title:* char[50], *medium:* char[30], *description:* char[120], *value:* integer, *year*: integer, ***collection_title****:* char[50], ***collection_curator****:* char[30])

- Primary key: {piece_id}
- Candidate keys: {piece_id}
- Foreign keys: {collection_title, collection_curator} (references Collection)
- collection_title, collection_curator, title are NOT NULL
- FDs: {piece_id} → {title, medium, description, value, year, collection_title, collection_curator}

Collection(*title:* char[50], *curator:* char[30], *theme:* char[50], *description:* char[120], ***location_name:*** char[50])

- Primary key: {title, curator}
- Candidate key: {title, curator}
- Foreign keys: {location_name} (references Location)
- location_name is NOT NULL
- FDs: {title, curator}→{theme, description, location_name}

Location(*name:* char[50], *country:* char[30], *region*: char[30], *city*: char[30], *st_address:* char[50], *postcode*: char[10], *yr_est:* integer)

- Primary key: {name}
- Candidate keys: {name} [in some museum-rich cities, there are multiple art locations (museums/galleries/private collections) located in the same building. Hence, the address fields are not candidate keys.]
- country, region, city, st_address, postcode, yr_est are NOT NULL
- FDs: {name} → {country, region, city, st_address, postcode, yr_est}, {postcode} → {city, region}
- Should have total participation in a Holds relationship with Collection, but this cannot be represented in SQL DDL without assertions

Museum(***name***: char[50], *num_visitors:* integer, *ticket_price:* integer, *benefactor:* char[30])

- Primary key: {name}
- Candidate keys: {name}
- Foreign keys: {name} (references Location)
- num_visitors, ticket_price are NOT NULL
- FDs: {name} → {num_visitors, ticket_price, benefactor}

Gallery(***name***: char[50], *num_pieces:* integer, *curator:* char[30])

- Primary key: {name}
- Candidate keys: {name}
- Foreign keys: {name} (references Location)
- num_pieces, curator are NOT NULL
- FDs: {name} → {num_pieces, curator}

PrivateCollection(***name***: char[50], *owner:* char[30])

- Primary key: {name}
- Candidate keys: {name}
- Foreign keys: {name} (references Location)
- owner is NOT NULL
- FDs: {name} → {owner}

BCNF Normalization

Location and AppUser are the only relations with FDs that do not have a superkey as the LHS (all other FDs have PKs or CKs as their LHS, so all other relations are already in BCNF). Therefore, these are the only 2 tables requiring normalization — the schemas of all the other relations remain the same.

Location decomposition:

- We have a non-PK FD: {postcode} → {city, region}.
- postcode+ = {postcode, city, region}, so postcode is not a candidate key/superkey, so this FD violates BCNF.
- For ease of normalization, rename the schema to Location(I, CI, R, C, S, P, Y, N)
- Take P→CR and decompose the relation into Location(I, CI, S, P, Y, N) and Postcode(P, C, R)
- I is a candidate key for Location because its closure over Location is I+ = {I, CI, S, P, Y, N}, so it is a key
- P is a candidate key for Postcode because P+ = {P, C, R}

Location(*name:* char[50], *country:* char[30], *st_address:* char[50], ***postcode***: char[30], *yr_est:* integer)

- Primary key: {name}
- Candidate keys: {name}
- Foreign keys: {postcode} (references Postcode)

Postcode(*postcode*: char[10], *region*: char[30], *city*: char[30])

- Primary key: {postcode}
- Candidate keys: {postcode}

AppUser decomposition:

- We have a non-PK FD: {dob}→{age}

- dob+ = {dob, age}, so dob is not a candidate key/superkey, so this FD violates BCNF.
- We'll decompose on this FD to get: AppUser(username, bio, pfp_url, dob} and AppUserAge{dob, age}
- As before, username is still a key for AppUser, because username+ = {username, bio, pfp_url, dob}, so this satisfies BCNF.
- dob is now a key for AppUserAge, so this is now in BCNF form:

AppUser(*username:* char[30], *bio:* char[120], *pfp_url:* char[100], ***dob:*** date)

- Primary key: {username}
- Candidate keys: {username}
- Foreign keys: {dob} (references AppUserAge)
- pfp_url, dob are NOT NULL
- pfp_url has a default value of a link to a default image
- FDs: {username}→{bio, pfp_url, dob}

AppUserAge(*dob*: date, *age:* integer)

- Primary key: {dob}
- Candidate keys: {dob}
- age is NOT NULL
- FDs: {dob}→{age}

Museum(***name***: char[50], *num_visitors:* integer, *ticket_price:* integer, *benefactor:* char[30])

- Primary key: {name}
- Candidate keys: {name}
- Foreign keys: {name} (references Location)
- num_visitors, ticket_price are NOT NULL
- FDs: {name} → {num_visitors, ticket_price, benefactor}

Gallery(***name***: char[50], *num_pieces:* integer, *curator:* char[30])

- Primary key: {name}
- Candidate keys: {name}
- Foreign keys: {name} (references Location)
- num_pieces, curator are NOT NULL
- FDs: {name} → {num_pieces, curator}

PrivateCollection(***name***: char[50], *owner:* char[30])

- Primary key: {name}
- Candidate keys: {name}
- Foreign keys: {name} (references Location)
- owner is NOT NULL
- FDs: {name} → {owner}

Follows(***follower****:* char[30], ***followee****:* char[30])

- Primary key: {follower, followee}
- Candidate keys: {follower, followee}
- Foreign keys: {follower} (references AppUser), {followee} (references AppUser)
- Require that the follower and followee are not the same person

Post(*post_id:* integer, *text:* char[120], *image_url:* char[100], *num_likes:* integer, *datetime:* datetime, *age_restricted:* number[1], ***username****:* char[30], ***piece_id****:* integer)

- Primary key: {post_id}
- Candidate keys: {post_id}, {image_url}, {username, datetime}
- Foreign keys: {username} (references AppUser), {piece_id} (references ArtPiece)
- image_url is UNIQUE, {username, datetime} is UNIQUE
- num_likes, text, username, piece_id, datetime, age_restricted are NOT NULL
- num_likes and age_restricted default to 0
- FDs: {post_id} → {text, image_url, num_likes, datetime, age_restricted, username, piece_id}, {image_url}→{text, post_id, num_likes, datetime, age_restricted, username, piece_id}, {username, datetime}→{text, post_id, image_url, num_likes, age_restricted, piece_id}

Comment(*comment_id:* integer, *text:* char[120], *num_likes:* integer, *age_restricted:* number[1], *datetime:* datetime, ***username**:* char[30], ***post_id**:* integer)

- Primary key: {comment_id, post_id}
- Candidate keys: {comment_id, post_id}
- Foreign keys: {post_id} (references Post), {username} (references AppUser)
- text, num_likes, datetime, age_restricted, username are NOT NULL
- num_likes and age_restricted default to 0
- FDs: {comment_id, post_id}→{text, num_likes, age_restricted, datetime, username}

Badge(*name:* char[30], *description:* char[120], *icon_url:* char[100])

- Primary key: {name}
- Candidate keys: {name}, {icon_url}, {description}
- name, icon_url, description are UNIQUE and NOT NULL
- FDs: {name} → {icon_url, description}, {icon_url}→{name, description}, {description}→{name, icon_url}

Earns(***username**:* char[30], ***badge_name**:* char[30])

- Primary key: {username, badge_name}
- Candidate keys: {username, badge_name}
- Foreign keys: {username} (references AppUser), {badge_name} (references Badge)

Artist(*artist_id:* integer, *name:* char[30], *dob:* date, *dod:* date, *description:* char[120])

- Primary key: {artist_id}
- Candidate keys: {artist_id}
- name is NOT NULL
- FDs: {artist_id} → {name, dob, dod, description}

Creates(***artist_id**:* integer, ***piece_id**:* integer)

- Primary key: {artist_id, piece_id}
- Candidate keys: {artist_id, piece_id}
- Foreign keys: {artist_id} (references Artist), {piece_id} (references ArtPiece)
- Should have a total participation constraint, but this cannot be represented in SQL DDL without assertions

ArtPiece(*piece_id*: integer, *title:* char[50], *medium:* char[30], *description:* char[120], *value:* integer, *year*: integer, ***collection_title***: char[50], ***collection_curator***: char[30])

- ➔ Primary key: {piece_id}
- ➔ Candidate keys: {piece_id}
- ➔ Foreign keys: {collection_title, collection_curator} (references Collection)
- ➔ collection_title, collection_curator, title are NOT NULL
- ➔ FDs: {piece_id} → {title, medium, description, value, year, collection_title, collection_curator}

Collection(*title*: char[50], *curator*: char[30], *theme:* char[50], *description:* char[120], ***location_name:*** char[50])

- ➔ Primary key: {title, curator}
- ➔ Candidate key: {title, curator}
- ➔ Foreign keys: {location_name} (references Location)
- ➔ location_name is NOT NULL
- ➔ FDs: {title, curator}→{theme, description, location_name}

DDL Statements

```
-- Oracle defaults to the equivalent of ON DELETE NO ACTION for
foreign keys, so that is not specified

CREATE TABLE AppUser
(
username VARCHAR(30) PRIMARY KEY,
bio VARCHAR(120),
pfp_url VARCHAR(100) DEFAULT
'https://placehold.co/400x400/blue/white?text=pfp' NOT NULL,
dob DATE NOT NULL,
FOREIGN KEY (dob) REFERENCES AppUserAge (dob) -- should also have
ON UPDATE NO ACTION
);

CREATE TABLE AppUserAge
(
dob DATE PRIMARY KEY,
age int NOT NULL
);

CREATE TABLE Follows
```

```
(follower VARCHAR(30),
 followee VARCHAR(30),
 FOREIGN KEY (follower) REFERENCES AppUser(username) ON DELETE
CASCADE, -- should also have ON UPDATE CASCADE
 FOREIGN KEY (followee) REFERENCES AppUser(username) ON DELETE
CASCADE, -- should also have ON UPDATE CASCADE
 PRIMARY KEY (follower, followee)
);

CREATE TABLE Post
(post_id int PRIMARY KEY,
 text VARCHAR(120) NOT NULL,
 image_url VARCHAR(100) UNIQUE,
 num_likes int DEFAULT 0 NOT NULL,
 datetime DATE NOT NULL,
 age_restricted NUMBER(1) DEFAULT 0 NOT NULL,
 username VARCHAR(30) NOT NULL,
 piece_id int NOT NULL,
 UNIQUE(username, datetime),
 FOREIGN KEY (username) REFERENCES AppUser(username) ON DELETE
CASCADE, -- should also have ON UPDATE CASCADE
 FOREIGN KEY (piece_id) REFERENCES ArtPiece(piece_id) -- should
also have ON UPDATE CASCADE
);

CREATE TABLE CommentPost
(comment_id int,
 text VARCHAR(120) NOT NULL,
 num_likes int DEFAULT 0  NOT NULL,
 datetime DATE NOT NULL,
 age_restricted NUMBER(1) DEFAULT 0 NOT NULL,
 username VARCHAR(30) NOT NULL,
 post_id int,
 PRIMARY KEY (comment_id, post_id),
 FOREIGN KEY (post_id) REFERENCES Post(post_id),
 FOREIGN KEY (username) REFERENCES AppUser(username) ON DELETE
CASCADE -- should also have ON UPDATE CASCADE
);

CREATE TABLE Badge
```

```
(name VARCHAR(30) PRIMARY KEY,
 description VARCHAR(120) UNIQUE NOT NULL,
 icon_url VARCHAR(100) UNIQUE NOT NULL
);

CREATE TABLE Earns
(username VARCHAR(30),
badge_name VARCHAR(30),
FOREIGN KEY (username) REFERENCES AppUser(username) ON DELETE
CASCADE, -- should also have ON UPDATE CASCADE
FOREIGN KEY (badge_name) REFERENCES Badge(name), -- should also
have ON UPDATE CASCADE,
PRIMARY KEY (username, badge_name)
);

CREATE TABLE Artist
(
artist_id int PRIMARY KEY,
name VARCHAR(30) NOT NULL,
dob DATE,
dod DATE,
description VARCHAR(120)
);

CREATE TABLE Creates
(artist_id int,
 piece_id  int,
 FOREIGN KEY (artist_id) REFERENCES Artist(artist_id), -- should
also have ON UPDATE CASCADE,
 FOREIGN KEY (piece_id) REFERENCES ArtPiece(piece_id), -- should
also have ON UPDATE CASCADE
 PRIMARY KEY (artist_id, piece_id)
);

CREATE TABLE ArtPiece
(piece_id int PRIMARY KEY,
 title VARCHAR(50) NOT NULL,
 medium VARCHAR(30),
 description VARCHAR(120),
 value int,
 year int,
 collection_title VARCHAR(50) NOT NULL,
 collection_curator VARCHAR(30) NOT NULL,
 FOREIGN KEY (collection_title, collection_curator) REFERENCES
Collection(title, curator) -- should also have ON UPDATE CASCADE
);

CREATE TABLE Collection
(
title VARCHAR(50),
curator VARCHAR(30),
theme VARCHAR(50),
description VARCHAR(120),
location_name VARCHAR(50) NOT NULL,
PRIMARY KEY (title, curator),
FOREIGN KEY (location_name) REFERENCES Location (name) -- should
also have ON UPDATE CASCADE
);

CREATE TABLE Location
(name VARCHAR(50),
 country VARCHAR(30) NOT NULL,
 st_address VARCHAR(50) NOT NULL,
 postcode VARCHAR(10) NOT NULL,
 yr_est int NOT NULL,
 FOREIGN KEY (postcode) REFERENCES Postcode(postcode), -- should
also have ON UPDATE CASCADE,
 PRIMARY KEY (name)
);

CREATE TABLE Postcode
(postcode VARCHAR(10),
 region VARCHAR(30) NOT NULL,
 city VARCHAR(30) NOT NULL,
 PRIMARY KEY(postcode)
);

CREATE TABLE Museum
(
name VARCHAR(50),
num_visitors int NOT NULL,
ticket_price int NOT NULL,
benefactor VARCHAR(30),
FOREIGN KEY (name) REFERENCES Location (name) ON DELETE NO
ACTION, -- should also have ON UPDATE CASCADE
PRIMARY KEY(name)
);

CREATE TABLE Gallery
(
name VARCHAR(50),
num_pieces int NOT NULL,
curator VARCHAR(30),
FOREIGN KEY (name) REFERENCES Location (name) ON DELETE NO
ACTION, -- should also have ON UPDATE CASCADE
PRIMARY KEY(name)
);

CREATE TABLE PrivateCollection
(
name VARCHAR(50),
owner VARCHAR(30) NOT NULL,
FOREIGN KEY (name) REFERENCES Location (name) ON DELETE NO
ACTION, -- should also have ON UPDATE CASCADE
PRIMARY KEY(name)
);
```

Insert Statements

```
INSERT INTO AppUser VALUES('nicole_lloyd', 'I like reviewing
     art', 'https://placehold.co/400x400/orange/white?text=pfp',
     TO_DATE('January 1 2000', 'Month DD YYYY'));
INSERT INTO AppUser VALUES('johnnie_banks', 'I like reviewing
     art', 'https://placehold.co/400x400/orange/white?text=pfp',
     TO_DATE('January 2 2000', 'Month DD YYYY'));
INSERT INTO AppUser VALUES('anon_art_lover', 'I like reviewing
     art', 'https://placehold.co/400x400/orange/white?text=pfp',
     TO_DATE('January 3 2000', 'Month DD YYYY'));
```

```
INSERT INTO AppUser VALUES('taylor_brewer', 'I like reviewing
     art', 'https://placehold.co/400x400/orange/white?text=pfp',
     TO_DATE('January 4 2000', 'Month DD YYYY'));
INSERT INTO AppUser VALUES('julia_mccoy', 'I like reviewing
     art', 'https://placehold.co/400x400/orange/white?text=pfp',
     TO_DATE('January 5 2000', 'Month DD YYYY'));
–
INSERT INTO AppUserAge VALUES(TO_DATE('January 1 2000', 'Month
     DD YYYY'), 23);
INSERT INTO AppUserAge VALUES(TO_DATE('January 2 2000', 'Month
     DD YYYY'), 23);
INSERT INTO AppUserAge VALUES(TO_DATE('January 3 2000', 'Month
     DD YYYY'), 23);
INSERT INTO AppUserAge VALUES(TO_DATE('January 4 2000', 'Month
     DD YYYY'), 23);
INSERT INTO AppUserAge VALUES(TO_DATE('January 5 2000', 'Month
     DD YYYY'), 23);
–
INSERT INTO Follows VALUES ('nicole_lloyd', 'johnnie_banks');
INSERT INTO Follows VALUES ('nicole_lloyd', 'julia_mccoy');
INSERT INTO Follows VALUES ('julia_mccoy', 'nicole_lloyd');
INSERT INTO Follows VALUES ('johnnie_banks', 'anon_art_lover');
INSERT INTO Follows VALUES ('taylor_brewer', 'anon_art_lover');
-
INSERT INTO Post VALUES (1, 'so orange!',
'https://placehold.co/600x400/orange/white', 5,
TO_DATE('2023-10-18 23:00:00', 'YYYY-MM-DD HH24:MI:SS'), 0,
'julia_mccoy', 1);
INSERT INTO Post VALUES (2, 'so red!',
     'https://placehold.co/600x400/red/white', 2,
     TO_DATE('2023-10-18 23:00:01', 'YYYY-MM-DD HH24:MI:SS'), 0,
     'johnnie_banks', 2);
INSERT INTO Post VALUES (3, 'so green!',
     'https://placehold.co/600x400/green/white', 7,
     TO_DATE('2023-10-18 23:00:02', 'YYYY-MM-DD HH24:MI:SS'), 0,
     'anon_art_lover', 3);
INSERT INTO Post VALUES (4, 'so blue!',
     'https://placehold.co/600x400/blue/white', 9,
     TO_DATE('2023-10-18 23:00:03', 'YYYY-MM-DD HH24:MI:SS'), 0,
     'nicole_lloyd', 4);
```

```
INSERT INTO Post VALUES (5, 'too colorful to be natural!',
     'https://placehold.co/600x400/blue/orange', 9,
     TO_DATE('2023-10-18 23:00:04', 'YYYY-MM-DD HH24:MI:SS'), 1,
     'nicole_lloyd', 5);
-
INSERT INTO CommentPost VALUES (1, 'so insightful!', 1,
     TO_DATE('2023-10-18 23:30:03', 'YYYY-MM-DD HH24:MI:SS'), 0,
     'anon_art_lover', 5);
INSERT INTO CommentPost VALUES (2, 'agreed!', 0,
     TO_DATE('2023-10-18 23:30:07', 'YYYY-MM-DD HH24:MI:SS'), 0,
     'johnnie_banks', 5);
INSERT INTO CommentPost VALUES (1, 'im so smart!', 0,
     TO_DATE('2023-10-18 23:05:07', 'YYYY-MM-DD HH24:MI:SS'), 0,
     'johnnie_banks', 2);
INSERT INTO CommentPost VALUES (2, 'on your own post?
     seriously?', 1, TO_DATE('2023-10-18 23:05:07', 'YYYY-MM-DD
     HH24:MI:SS'), 0, 'julia_mccoy', 2);
INSERT INTO CommentPost VALUES (3, 'lame', 1,
     TO_DATE('2023-10-18 23:05:17', 'YYYY-MM-DD HH24:MI:SS'), 0,
     'nicole_lloyd', 2);
-
INSERT INTO Badge VALUES ('Explorer', 'Visit 25 different art
     locations.',
     'https://placehold.co/400x400/green/white?text=explorer');
INSERT INTO Badge VALUES ('Connoisseur', 'Make 100 posts.',
     'https://placehold.co/400x400/blue/white?text=connoisseur')
     ;
INSERT INTO Badge VALUES ('Collector', 'Make 10 posts from one
     art location.',
     'https://placehold.co/400x400/orange/white?text=collector')
     ;
INSERT INTO Badge VALUES ('Expert', 'Make 500 posts.',
     'https://placehold.co/400x400/orange/white?text=expert');
INSERT INTO Badge VALUES ('Enthusiast', 'Make 25 posts.',
     'https://placehold.co/400x400/orange/white?text=enthusiast'
     );
-
INSERT INTO Earns VALUES ('taylor_brewer', 'Explorer');
INSERT INTO Earns VALUES ('nicole_lloyd', 'Connoisseur');
INSERT INTO Earns VALUES ('anon_art_lover', 'Collector');
INSERT INTO Earns VALUES ('johnnie_banks', 'Expert');
INSERT INTO Earns VALUES ('julia_mccoy', 'Enthusiast');
-
INSERT INTO Artist VALUES(1, 'Leonardo Da Vinci', TO_DATE('April
     15 1452', 'Month DD YYYY'), TO_DATE('May 2 1519', 'Month DD
     YYYY'), 'Many-talented Italian Renaissance man.');

INSERT INTO Artist VALUES(2, 'Vincent van Gogh', TO_DATE('March
     30 1853', 'Month DD YYYY'), TO_DATE('July 29 1890', 'Month
     DD YYYY'), 'Dutch post-impressionist painter known for his
     expressive use of color and brushstrokes.');

INSERT INTO Artist VALUES(3, 'Pablo Picasso', TO_DATE('October
     25 1881', 'Month DD YYYY'), TO_DATE('April 8 1973', 'Month
     DD YYYY'), 'Spanish painter, sculptor, and co-founder of
     Cubism.');

INSERT INTO Artist VALUES(4, 'Frida Kahlo', TO_DATE('July 6
     1907', 'Month DD YYYY'), TO_DATE('July 13 1954', 'Month DD
     YYYY'), 'Mexican painter and feminist icon who depicted her
     personal and political struggles in self-portraits.');
INSERT INTO Artist VALUES(5, 'Andy Warhol', TO_DATE('August 6
1928', 'Month DD YYYY'), TO_DATE('February 22 1987', 'Month DD
YYYY'), 'American artist and leader of the Pop Art movement who
used mass-produced images and everyday objects in his works.');
-
INSERT INTO Creates VALUES (1, 1);
INSERT INTO Creates VALUES (2, 2);
INSERT INTO Creates VALUES (3, 3);
INSERT INTO Creates VALUES (4, 4);
INSERT INTO Creates VALUES (5, 5);
-
INSERT INTO ArtPiece VALUES (1, 'Orange Overload', 'Digital
     Art', 'orange art. go figure.', 1000000000, 2023, 'Colors',
     'Sir Art Si');
INSERT INTO ArtPiece VALUES (2, 'Red Canvas', 'Digital Art',
     'red art. go figure.', 1000000000, 2023, 'Colors', 'Sir Art
     Si');
INSERT INTO ArtPiece VALUES (3, 'Green Canvas', 'Digital Art',
     'green art. go figure.', 1000000000, 2023, 'Colors', 'Sir
     Art Si');
```

```
INSERT INTO ArtPiece VALUES (4, 'Blue', 'Digital Art', 'blue
     art. go figure.', 1000000000, 2023, 'Colors', 'Sir Art
     Si');
INSERT INTO ArtPiece VALUES (5, 'Orange & Blue Chaos', 'Digital
     Art', 'colourful art. go figure.', 1000000000, 2023,
     'Colors', 'Sir Art Si');
-
INSERT INTO Collection VALUES('Colors', 'Sir Art Si', 'Colors',
     'A collection of colors', 'Vancouver's Secret Art
     Gallery');
INSERT INTO Collection VALUES('Another Collection', 'Sir Art
     Si', 'Colors', 'Another collection of colors', 'Vancouver's
     Secret Art Gallery');
INSERT INTO Collection VALUES('Untitled Art', 'Sir Art Si',
     'Colors', 'Too good for titles', 'Vancouver's Secret Art
     Gallery');
INSERT INTO Collection VALUES('Travelling Palettes', 'Sir Art
     Si', 'Colors', 'A collection of colors', 'Vancouver's
     Secret Art Gallery');
INSERT INTO Collection VALUES('Colors', 'John Doe', 'Colors', 'A
     collection of colors', 'Vancouver's Secret Art Gallery');
-
INSERT INTO Location VALUES ('Vancouver's Secret Art Gallery',
     'Canada', '1 Artists Way', 'V6K 2G2', 2010);
INSERT INTO Location VALUES ('Museum of Modern Vancouver Art',
     'Canada', '3170 Oak St', 'V6H 2L2', 1957);
INSERT INTO Location VALUES ('Musée des Capucines', 'France',
     '17 Boulevard des Capucines', '75002', 2012);
INSERT INTO Location VALUES ('Museum of German Art', 'Germany',
     'Zehntsteinweg 22', '79110', 2012);
INSERT INTO Location VALUES ('Le Centre d'Art', 'France', '104
     Rue Vieille du Temple', '75006', 1740);
INSERT INTO Location VALUES ('Galerie des Pamplemousses',
     'France', '20 Rue Marmontel', '75015', 2016);
INSERT INTO Location VALUES ('Galerie des Jardins', 'France',
     '20 Rue Marmontel', '75015', 2010);
INSERT INTO Location VALUES ('Galerie du Sacre Coeur', 'France',
     '20 Rue Marmontel', '75015', 2008);
INSERT INTO Location VALUES ('Galerie des Fleurs', 'France', '20
     Rue Marmontel', '75015', 2015);
```

```
INSERT INTO Location VALUES ('Galerie et Magasin des Peintures
     Jolies', 'France', '20 Rue Marmontel', '75015', 2015);
INSERT INTO Location VALUES ('Clarke Private Collection',
     'Canada', '1337 Seymour Street', 'V6R 2H8', 2005);
INSERT INTO Location VALUES ('Girard Private Portfolio',
     'Canada', '620 Clark Dr', 'V5L 3H8', 1999);
INSERT INTO Location VALUES ('Wong Family Collection', 'Canada',
     '5390 Grimmer St', 'V5H 2H2', 2014);
INSERT INTO Location VALUES ('Tremblay Private Collection',
     'Canada', '777 Hornby St', 'V6Z 1S4', 2003);
INSERT INTO Location VALUES ('Stewart Family Portfolio',
     'Canada', '4012 Hastings St', 'V5C 2J1', 2020);
-
INSERT INTO Postcode VALUES ('V6K 2G2', 'Vancouver', 'British
     Columbia');
INSERT INTO Postcode VALUES ('V6H 2L2', 'Vancouver', 'British
     Columbia');
INSERT INTO Postcode VALUES ('75002', 'Paris', 'Île-de-France');
INSERT INTO Postcode VALUES ('79110', 'Munich', 'Upper
     Bavaria');
INSERT INTO Postcode VALUES ('75006', 'Paris', 'Île-de-France');
INSERT INTO Postcode VALUES ('75015', 'Paris', 'Île-de-France');
INSERT INTO Postcode VALUES ('V6R 2H8', 'Victoria', 'British
     Columbia');
INSERT INTO Postcode VALUES ('V5L 3H8', 'West Vancouver',
     'British Columbia');
INSERT INTO Postcode VALUES ('V5H 2H2', 'Port Coquitlam',
     'British Columbia');
INSERT INTO Postcode VALUES ('V6Z 1S4', 'Burnaby', 'British
     Columbia');
INSERT INTO Postcode VALUES ('V5C 2J1', 'Richmond', 'British
     Columbia');
-
INSERT INTO Museum VALUES ('Vancouver's Secret Art Gallery',
     10000, 10, 'Vancouver Artists Foundation');
INSERT INTO Museum VALUES ('Museum of Modern Vancouver Art',
     12000, 10, 'Guild of Canadian Artists');
INSERT INTO Museum VALUES ('Musée des Capucines', 400, 10, 'Les
     Amateurs d'Art');
```

```
INSERT INTO Museum VALUES ('Museum of German Art', 15000, 10,
     'German Art Lovers Foundation');
INSERT INTO Museum VALUES ('Le Centre d'Art', 6000, 10, 'La
     Fondation des Artistes');
-
INSERT INTO Gallery VALUES('Galerie des Pamplemousses', 10,
     'Emma Melgares');
INSERT INTO Gallery VALUES('Galerie des Jardins', 12, 'Emma
     Melgares');
INSERT INTO Gallery VALUES('Galerie du Sacre Coeur', 52, 'Anais
     Chalbot');
INSERT INTO Gallery VALUES('Galerie des Fleurs', 5, 'Leo Jean');
INSERT INTO Gallery VALUES('Galerie et Magasin des Peintures
     Jolies', 17, 'Alice Boyer');
-
INSERT INTO PrivateCollection Values('Clarke Private
     Collection', 'Isaac Clarke');
INSERT INTO PrivateCollection Values('Girard Private Portfolio',
     'Ava Girard');
INSERT INTO PrivateCollection Values('Wong Family Collection',
     'Scarlett Wong');
INSERT INTO PrivateCollection Values('Tremblay Private
     Collection', 'Noah Tremblay');
INSERT INTO PrivateCollection Values('Stewart Family Portfolio',
     'Luca Stewart');
```