

This is a database of sample social media that allows users to upload posts, stories and list items for sale. Users can add location to things they post and also add songs to stories. Users can also make comments to the other users' posts and stories. Users can also create and join message groups. Users can also create advertisements and show them to other users.

The diagram is a complex Entity-Relationship (ER) model for a social media system. It features the following components:

- Entities (Rectangles):**
  - Users** (Central entity, thick border): Attributes include ID (PK), Username, Display name, Birthday, Zodiac, Bio, Password hash, Password salt, Is Admin, Role, and various relationship roles (Follower, Following, Blocker, Blocked).
  - Advertisement**: Attributes include ID (PK), Image URL, Click URL, and Remaining views.
  - Ad showing** (Thick border): Attributes include ID (PK), Time, and Clicked.
  - Message** (Thick border): Attributes include ID (PK), Text, and Time.
  - Message Group**: Attributes include ID (PK) and Name.
  - Locations**: Attributes include ID (PK), Name, City, Country, Altitude, Latitude, and Longitude.
  - Media** (Thick border): Attributes include ID (PK), URL, Index, and Caption.
  - Posts**: Attributes include ID (PK), Time, and Caption.
  - Comment** (Thick border): Attributes include ID (PK) and Text.
  - Stories**: Attributes include ID (PK), URL, and Expires.
  - Song**: Attributes include ID (PK), URL, Artist, and Title.
  - Listings**: Attributes include Title and Price.
  - Product Categories**: Attributes include ID (PK) and Name.
  - Story Reply**: A special case of Message.
- Relationships (Diamonds):**
  - Shown to**: Connects Advertisement and Ad showing.
  - Creates**: Connects Users and Advertisement.
  - Follows**: Connects Users (Follower) and Users (Following).
  - Blocks**: Connects Users (Blocker) and Users (Blocked).
  - Comments**: Connects Users and Comment.
  - Comments on** (Thick border): Connects Comment and Posts.
  - Likes**: Connects Users and Posts.
  - Uploads**: Connects Users and Posts.
  - Is at**: Connects Users and Locations.
  - Includes**: Connects Posts and Media.
  - Replies to**: Connects Story Reply and Stories.
  - Includes**: Connects Stories and Song.
  - Offers**: Connects Listings and Product Categories.
- Other Relationships and Constraints:**
  - Is in**: Connects Message Group and Message.
  - IsA** (Disjoint Partial): Connects Message and Story Reply.

- As requested by the TA, the “Shown to” relationship of the “Ad showing” weak entity was bolded to properly indicate that this relationship is what defines the ad showing.
- For locations, coordinates was replaced with latitude, longitude, and altitude to take better advantage of Oracle’s built in data types and avoid having to write conversions.
- Added city and country to Locations specifically so we could create a functional dependency.

- Added birthday and zodiac to Users to create a functional dependency.

## Relational model (not normalized)

Users(

```
UserID: CHAR(16),
Username: VARCHAR2(32),
DisplayName: VARCHAR2(32),
Bio: VARCHAR2(4000),
PasswordHash: CHAR(256),
PasswordSalt: CHAR(256),
IsAdmin: INTEGER,
IsAt: CHAR(16),
Birthday: DATE,
Zodiac: INTEGER
```

)

- UserID is the primary key. It has been renamed from the ER diagram because ID is not sufficiently descriptive. The same renaming has been applied to all other entities.
- Username, PasswordHash, PasswordSalt, and isAdmin not null.
- Username and PasswordSalt must be unique and are candidate keys.
- IsAt is a foreign key referring to Locations(LocationID). On delete set null.

MessageGroupMembers(

```
MessageGroup: CHAR(16),
User: CHAR(16)
```

)

- Renamed from “Is in” on the ER diagram to be more descriptive and avoid naming collisions.
- (MessageGroup, User) is the primary key
- MessageGroup is a foreign key referring to MessageGroups(MessageGroupID). On delete cascade.
- User is a foreign key referring to Users(UserID). On delete cascade.

AdRoles(User: CHAR(16), Advertisement: CHAR(16), Role: INTEGER)

- Renamed from “Creates” on the ER diagram to be more descriptive and avoid naming collisions.
- (User, Advertisement) is the primary key
- Role not null
- User is a foreign key referring to Users(UserID). On delete cascade.
- Advertisement is a foreign key referring to Advertisements(AdID). On delete cascade.

AdShowing(

```
AdShowingID: CHAR(16),
Time: DATE,
Clicked: INTEGER,
Advertisement: CHAR(16),
User: CHAR(16)
```

)

- (AdShowingID, Advertisement, User) is the primary key
- All attributes not null.
- Advertisement is a foreign key referring to Advertisements(AdID). On delete cascade.
- User is a foreign key referring to Users(UserID). On delete cascade.

UserMedia(User: CHAR(16), Media: CHAR(16), Post: CHAR(16))

- Renamed from Uploads on the ER diagram to be more descriptive.
- (User, Media) is the primary key
- User is a foreign key referring to Users(UserID). On delete cascade.
- (Media, Post) is a foreign key referring to Media(MediaID, Post). On delete cascade.

Likes(User: CHAR(16), Post: CHAR(16))

- (User, Post) is the primary key
- User is a foreign key referring to Users(UserID). On delete cascade.
- Post is a foreign key referring to Posts(PostID). On delete cascade.

UserPosts(User: CHAR(16), Post: CHAR(16), Role: INTEGER)

- Renamed from "Creates" on the ER diagram to be more descriptive and avoid naming collisions.
- (User, Post) is the primary key
- All values not null.
- User is a foreign key referring to Users(UserID). On delete cascade.
- Post is a foreign key referring to Posts(ID). On delete cascade.

Follows(Follower: CHAR(16), Following: CHAR(16))

- (Follower, Following) is the primary key
- Follower and Following are foreign keys referring to Users(UserID). On delete cascade.

Blocks(Blocker: CHAR(16), Blocked: CHAR(16))

- (Blocker, Blocked) is the primary key
- Blocker and Blocked are foreign keys referring to Users(UserID). On delete cascade.

Locations(

LocationID: CHAR(16),  
Name: VARCHAR2(32),  
Latitude: NUMBER(15, 12),  
Longitude: NUMBER(15, 12),  
Altitude: NUMBER(15, 12),  
City: VARCHAR2(256),  
Country: VARCHAR2(256)

)

- LocationID is the primary key
- (Name, Latitude, Longitude, Altitude) is unique and a candidate key.
- LocationID, Name, Latitude, and Longitude not null. (Altitude may be null.)

MessageGroups(MessageGroupID: CHAR(16), Name: VARCHAR2(32))

- MessageGroupID is the primary key

```
Messages(
    MessageID: CHAR(16),
    Time: DATE,
    Text: LONG,
    MessageGroup: CHAR(16),
    Sender: CHAR(16),
    ReplyTo: CHAR(16)
)
```

- (MessageID, MessageGroup) is the primary key
- MessageID, Time, Text, MessageGroup, and Sender not null. (ReplyTo may be null if this is not a story reply.)
- MessageGroup is a foreign key referencing MessageGroups(ID). On delete cascade.
- Sender is a foreign key referencing Users(UserID). On delete set null.
- ReplyTo is a foreign key referencing Posts(PostID). On delete set null. Renamed from "Replies to" on the ER diagram to be grammatically nicer

```
Advertisements(
    AdID: CHAR(16),
    ImageURL: VARCHAR2(512),
    ClickURL: VARCHAR2(512),
    RemainingViews: INTEGER
)
```

- AdID is the primary key. Using AdID instead of AdvertisementID because AdID is far less annoying to type.
- All attributes not null

```
Media(
    MediaID: CHAR(16),
    URL: VARCHAR2(512),
    Post: CHAR(16),
    Index: INTEGER
)
```

- (ID, Post) is the primary key
- (Post, Index) is unique and a candidate key
- All attributes not null
- Post is a foreign key referring to Posts(PostID). On delete cascade.

```
Posts(
    PostID: CHAR(16),
    Time: DATE,
    Caption: VARCHAR2(4000),
    IsAt: CHAR(16),
    Expires: DATE
)
```

- PostID is the primary key
- Time not null. (Caption, IsAt, and Expires may be null.)
- IsAt is a foreign key referencing Locations(LocationID). On delete set null.

```
Listings(
```

```

        PostID: CHAR(16),
        Title: VARCHAR2(32),
        Price: INTEGER,
        Offering: CHAR(16)
    )
    • PostID is the primary key. Using PostID as the name rather than ListingID
      to make it clear that a Listing is a Post.
    • All attributes not null
    • PostID is a foreign key referencing Posts(PostID). On delete cascade.
    • Offering is a foreign key referencing ProductCategories(ProductCategoryID).
      On delete set null.

Comments(
    CommentID: CHAR(16),
    Time: DATE,
    Text: LONG,
    User: CHAR(16),
    Post: CHAR(16)
)
    • (CommentID, Post) is the primary key
    • All attributes not null
    • User is a foreign key referencing Users(UserID). On delete cascade.
    • Post is a foreign key referencing Posts(PostID). On delete cascade.

ProductCategories(
    ProductCategoryID: CHAR(16),
    Name: VARCHAR2(32)
)
    • ID is the primary key
    • All attributes not null

```

## Functional Dependencies

User:

-UserID -> Username, Display Name, Bio, Password Hash, Password Salt, isAdmin,  
 Birthday  
 -Birthday -> Zodiac Sign

Advertisement:

-AdID -> ImageURL, ClickURL, Remaining views

Ad Showing:

-AdShowingID, AdID, UserID -> AdTime, AdClicked

Message Group:

-MessageGroupID -> MessageGroupName

Message:

-MessageGroupID, MessageID -> MessageText, MessageTime

Story Reply:

-MessageGroupID, MessageID -> MessageText, MessageTime

Posts:

-PostID -> Caption, Time

Comment:

-CommentID, PostID -> CommentTime, CommentText

Media:

-MediaID, PostID -> MediaURL, Index

Locations:

-LocationID -> LocationName, Altitude, Longitude, Latitude, City, Country

-Latitude, Longitude -> City, Country

Stories:

-PostID -> Caption, Time, Expires

Song:

-SongID -> SongURL, SongArtist, SongTitle

Listings:

-PostID -> Caption, Time, ListingTitle, ListingPrice, Minimum Offer Percentage, Minimum Offer

-ListingPrice, Minimum Offer Percentage -> Minimum Offer

## Normalization

**\*\*foreign keys are *italic***

User(UserID, Username, Display Name, Bio, Password Hash, Password Salt, isAdmin, *Birthday*)

Zodiac(Birthday, Zodiac Sign)

Advertisement(AdID, ImageURL, ClickURL, Remaining views)

Ad Showing(AdShowingID, AdID, UserID, AdTime, AdClicked)

Message Group(MessageGroupID, MessageGroupName)

Message(MessageGroupID, MessageID, MessageText, MessageTime)

Story Reply(MessageGroupID, MessageID, MessageText, MessageTime)

Posts(PostID, Caption, Time)

Comment(CommentID, PostID, CommentTime, CommentText)

Media(MediaID, PostID, MediaURL, Index)

Locations(LocationID, LocationName, Altitude, *Longitude*, *Latitude*)

Region(Latitude, Longitude, City, Country)

Stories(PostID, Caption, Time, Expires)

Song(SongID, SongURL, SongArtist, SongTitle)

## SQL DDL

```
CREATE TABLE Regions(  
    Latitude NUMBER(15, 12),  
    Longitude NUMBER(15, 12),  
    City VARCHAR2(256),  
    Country VARCHAR2(256),  
    PRIMARY KEY (Latitude, Longitude)  
);  
  
CREATE TABLE Locations(  
    LocationID CHAR(16) PRIMARY KEY,  
    "Name" VARCHAR2(32) NOT NULL,  
    Latitude NUMBER(15, 12) NOT NULL,  
    Longitude NUMBER(15, 12) NOT NULL,  
    Altitude NUMBER(15, 12),  
    City VARCHAR2(256),  
    Country VARCHAR2(256),  
    UNIQUE ("Name", Latitude, Longitude, Altitude),  
    FOREIGN KEY (Latitude, Longitude) REFERENCES Regions(Latitude, Longitude) ON  
DELETE CASCADE  
);  
  
CREATE TABLE Zodiac(  
    Birthday DATE PRIMARY KEY,  
    Zodiac INTEGER  
);  
  
CREATE TABLE Users(  
    UserID CHAR(16) PRIMARY KEY,  
    Username VARCHAR2(32) UNIQUE,  
    DisplayName VARCHAR2(32),  
    Bio VARCHAR2(4000),  
    PasswordHash CHAR(256) NOT NULL,  
    PasswordSalt CHAR(256) UNIQUE NOT NULL,  
    IsAdmin INTEGER NOT NULL,  
    IsAt CHAR(16),  
    Birthday DATE,  
    FOREIGN KEY (IsAt) REFERENCES Locations(LocationID) ON DELETE SET NULL,  
    FOREIGN KEY (Birthday) REFERENCES Zodiac(Birthday)  
);  
  
CREATE TABLE MessageGroups(  
    MessageGroupID CHAR(16) PRIMARY KEY,  
    "Name" VARCHAR2(32)  
);
```

```

CREATE TABLE MessageGroupMembers(
    MessageGroup CHAR(16),
    "User" CHAR(16),
    PRIMARY KEY (MessageGroup, "User"),
    FOREIGN KEY (MessageGroup) REFERENCES MessageGroups(MessageGroupID) ON DELETE
    CASCADE,
    FOREIGN KEY ("User") REFERENCES Users(UserID) ON DELETE CASCADE
);

```

```

CREATE TABLE Advertisements(
    AdID CHAR(16) PRIMARY KEY,
    ImageURL VARCHAR2(512) NOT NULL,
    ClickURL VARCHAR2(512) NOT NULL,
    RemainingViews INTEGER NOT NULL
);

```

```

CREATE TABLE AdRoles(
    "User" CHAR(16),
    Advertisement CHAR(16),
    "Role" INTEGER NOT NULL,
    PRIMARY KEY ("User", Advertisement),
    FOREIGN KEY ("User") REFERENCES Users(UserID) ON DELETE CASCADE,
    FOREIGN KEY (Advertisement) REFERENCES Advertisements(AdID) ON DELETE CASCADE
);

```

```

CREATE TABLE AdShowing(
    AdShowingID CHAR(16),
    "Time" DATE NOT NULL,
    Clicked INTEGER NOT NULL,
    Advertisement CHAR(16),
    "User" CHAR(16),
    PRIMARY KEY (AdShowingID, Advertisement, "User"),
    FOREIGN KEY (Advertisement) REFERENCES Advertisements(AdID) ON DELETE CASCADE,
    FOREIGN KEY ("User") REFERENCES Users(UserID) ON DELETE CASCADE
);

```

```

CREATE TABLE Posts(
    PostID CHAR(16) PRIMARY KEY,
    "Time" DATE NOT NULL,
    Caption VARCHAR2(4000),
    IsAt CHAR(16),
    Expires DATE,
    FOREIGN KEY (IsAt) REFERENCES Locations(LocationID) ON DELETE SET NULL
);

```

```

CREATE TABLE Media(
    MediaID CHAR(16),
    "URL" VARCHAR2(512) NOT NULL,
    Post CHAR(16),
    "Index" INTEGER NOT NULL,
    PRIMARY KEY (MediaID, Post),
    UNIQUE (Post, "Index"),
    FOREIGN KEY (Post) REFERENCES Posts(PostId)
);

```



```

CREATE TABLE UserMedia(
    "User" CHAR(16),
    Media CHAR(16),
    Post CHAR(16),
    PRIMARY KEY ("User", Media),
    FOREIGN KEY ("User") REFERENCES Users(UserID) ON DELETE CASCADE,
    FOREIGN KEY (Media, Post) REFERENCES Media(MediaID, Post) ON DELETE CASCADE
);

```

```

CREATE TABLE Likes(
    "User" CHAR(16),
    Post CHAR(16),
    PRIMARY KEY ("User", Post),
    FOREIGN KEY ("User") REFERENCES Users(UserID) ON DELETE CASCADE,
    FOREIGN KEY (Post) REFERENCES Posts(PostID) ON DELETE CASCADE
);

```

```

CREATE TABLE UserPosts(
    "User" CHAR(16),
    Post CHAR(16),
    "Role" INTEGER NOT NULL,
    PRIMARY KEY ("User", Post),
    FOREIGN KEY ("User") REFERENCES Users(UserID) ON DELETE CASCADE,
    FOREIGN KEY (Post) REFERENCES Posts(PostID) ON DELETE CASCADE
);

```

```

CREATE TABLE Follows(
    Follower CHAR(16),
    "Following" CHAR(16),
    PRIMARY KEY (Follower, "Following"),
    FOREIGN KEY (Follower) REFERENCES Users(UserID) ON DELETE CASCADE,
    FOREIGN KEY ("Following") REFERENCES Users(UserID) ON DELETE CASCADE
);

```

```

CREATE TABLE Blocks(
    Blocker CHAR(16),
    Blocked CHAR(16),
    PRIMARY KEY (Blocker, Blocked),
    FOREIGN KEY (Blocker) REFERENCES Users(UserID) ON DELETE CASCADE,
    FOREIGN KEY (Blocked) REFERENCES Users(UserID) ON DELETE CASCADE
);

```

```

CREATE TABLE Messages(
    MessageID CHAR(16),
    "Time" DATE NOT NULL,
    "Text" LONG NOT NULL,
    MessageGroup CHAR(16),
    Sender CHAR(16) NOT NULL,
    ReplyTo CHAR(16),
    PRIMARY KEY (MessageID, MessageGroup),
    FOREIGN KEY (MessageGroup) REFERENCES MessageGroups(MessageGroupID) ON DELETE
CASCADE,
    FOREIGN KEY (Sender) REFERENCES Users(UserID) ON DELETE SET NULL,
    FOREIGN KEY (ReplyTo) REFERENCES Posts(PostID) ON DELETE SET NULL
);

```

```

CREATE TABLE ProductCategories(
    ProductCategoryID CHAR(16) PRIMARY KEY,
    "Name" VARCHAR2(32) NOT NULL
);

CREATE TABLE Listings(
    PostID CHAR(16) PRIMARY KEY,
    Title VARCHAR2(32) NOT NULL,
    Price INTEGER NOT NULL,
    Offering CHAR(16) NOT NULL,
    FOREIGN KEY (PostID) REFERENCES Posts(PostID) ON DELETE CASCADE,
    FOREIGN KEY (Offering) REFERENCES ProductCategories(ProductCategoryID) ON
DELETE SET NULL
);

CREATE TABLE Comments(
    CommentID CHAR(16),
    "Time" DATE NOT NULL,
    "Text" LONG NOT NULL,
    "User" CHAR(16) NOT NULL,
    Post CHAR(16),
    PRIMARY KEY (CommentID, Post),
    FOREIGN KEY ("User") REFERENCES Users(UserID) ON DELETE CASCADE,
    FOREIGN KEY (Post) REFERENCES Posts(PostID) ON DELETE CASCADE
);

```

## INSERT Statements

```

INSERT INTO Users(ID, Username, DisplayName, Bio, PasswordHash, PasswordSalt,
IsAdmin, IsAt)
VALUES('user_id1', 'johnsmith', 'John Smith', 'Don't worry be happy',
'hashed_password1', 'salt_value1', 0, 'location_id1'),
    ('user_id2', 'catlover123', 'Cindy Hale', 'I love cats!',
'hashed_password2', 'salt_value2', 0, 'location_id2'),
    ('user_id3', 'darboyxx', 'dArk b0y', 'I hate everyone', 'hashed_password3',
'salt_value3', 0, 'location_id3'),
    ('user_id4', 'thatguyatubc', 'That Guy', 'UBC'25', 'hashed_password4',
'salt_value4', 1, 'location_id4'),
    ('user_id5', 'ubc', 'University of British Columbia', 'Tuum est',
'hashed_password5', 'salt_value5', 0, 'location_id5');
INSERT INTO MessageGroupMembers(MessageGroup, User)
VALUES('group_id1', 'user_id1'),
    ('group_id2', 'user_id2'),
    ('group_id3', 'user_id3'),
    ('group_id4', 'user_id4'),
    ('group_id5', 'user_id5');

INSERT INTO AdRoles(User, Advertisement, Role)
VALUES('user_id1', 'ad_id1', 1),
    ('user_id2', 'ad_id2', 1),
    ('user_id3', 'ad_id3', 0),

```

```
    ('user_id4', 'ad_id4', 1),  
    ('user_id5', 'ad_id5', 0);
```

```
INSERT INTO AdShowing(ID, Time, Clicked, Advertisement, User)  
VALUES('showing_id1', TO_DATE('2023-10-16', 'YYYY-MM-DD'), 0, 'ad_id1',  
      'user_id1'),  
      ('showing_id2', TO_DATE('2023-10-16', 'YYYY-MM-DD'), 1, 'ad_id2',  
      'user_id2'),  
      ('showing_id3', TO_DATE('2023-10-16', 'YYYY-MM-DD'), 0, 'ad_id3',  
      'user_id3'),  
      ('showing_id4', TO_DATE('2023-10-16', 'YYYY-MM-DD'), 1, 'ad_id4',  
      'user_id4'),  
      ('showing_id5', TO_DATE('2023-10-16', 'YYYY-MM-DD'), 0, 'ad_id5',  
      'user_id5');
```

```
INSERT INTO UserMedia(User, Media)  
VALUES('user_id1', 'media_id1'),  
      ('user_id2', 'media_id2'),  
      ('user_id3', 'media_id3'),  
      ('user_id4', 'media_id4'),  
      ('user_id5', 'media_id5');
```

```
INSERT INTO Likes(User, Post)  
VALUES('user_id1', 'post_id1'),  
      ('user_id2', 'post_id2'),  
      ('user_id3', 'post_id3'),  
      ('user_id4', 'post_id4'),  
      ('user_id5', 'post_id5');
```

```
INSERT INTO UserPosts(User, Post, Role)  
VALUES('user_id1', 'post_id1', 1),  
      ('user_id2', 'post_id2', 1),  
      ('user_id3', 'post_id3', 0),  
      ('user_id4', 'post_id4', 1),  
      ('user_id5', 'post_id5', 0);
```

```
INSERT INTO Follows(Follower, Following)  
VALUES('user_id1', 'user_id2'),  
      ('user_id2', 'user_id3'),  
      ('user_id3', 'user_id4'),  
      ('user_id4', 'user_id5'),  
      ('user_id5', 'user_id1');
```

```
INSERT INTO Blocks(Blocker, Blocked)  
VALUES('user_id1', 'user_id3'),  
      ('user_id2', 'user_id4'),  
      ('user_id3', 'user_id5'),  
      ('user_id4', 'user_id1'),  
      ('user_id5', 'user_id2');
```

```
INSERT INTO Locations(ID, Name, Lat, Long, Altitude)  
VALUES('location_id1', 'LocationName1', 80, 30, 10),  
      ('location_id2', 'LocationName2', 70, 20, 10),  
      ('location_id3', 'LocationName3', 40, 40, 10),  
      ('location_id4', 'LocationName4', 0, 10, 0),
```

```
('location_id5', 'LocationName5', 50, 60, 20);
```

```
INSERT INTO MessageGroups(ID, Name)
VALUES('group_id1', 'GroupName1'),
      ('group_id2', 'GroupName2'),
      ('group_id3', 'GroupName3'),
      ('group_id4', 'GroupName4'),
      ('group_id5', 'GroupName5');
```

```
INSERT INTO Messages(ID, Time, Text, MessageGroup, Sender, ReplyTo)
VALUES('msg_id1', TO_DATE('2023-10-16', 'YYYY-MM-DD'), 'Hi', 'group_id1',
      'user_id1', 'post_id1'),
      ('msg_id2', TO_DATE('2023-10-16', 'YYYY-MM-DD'), 'Hello', 'group_id2',
      'user_id2', 'post_id2'),
      ('msg_id3', TO_DATE('2023-10-16', 'YYYY-MM-DD'), 'Bye', 'group_id3',
      'user_id3', 'post_id3'),
      ('msg_id4', TO_DATE('2023-10-16', 'YYYY-MM-DD'), 'Hey', 'group_id4',
      'user_id4', 'post_id4'),
      ('msg_id5', TO_DATE('2023-10-16', 'YYYY-MM-DD'), 'LoL', 'group_id5',
      'user_id5', 'post_id5');
```

```
INSERT INTO Advertisements(ID, ImageURL, ClickURL, RemainingViews)
VALUES('ad_id1', 'http://imageurl.com/image1', 'http://clickurl.com/click1', 99),
      ('ad_id2', 'http://imageurl.com/image2', 'http://clickurl.com/click2', 89),
      ('ad_id3', 'http://imageurl.com/image3', 'http://clickurl.com/click3', 79),
      ('ad_id4', 'http://imageurl.com/image4', 'http://clickurl.com/click4', 69),
      ('ad_id5', 'http://imageurl.com/image5', 'http://clickurl.com/click5', 59);
```

```
INSERT INTO Media(ID, URL, Post, Index)
VALUES('media_id1', 'http://mediaurl.com/media1', 'post_id1', 1),
      ('media_id2', 'http://mediaurl.com/media2', 'post_id2', 1),
      ('media_id3', 'http://mediaurl.com/media3', 'post_id3', 1),
      ('media_id4', 'http://mediaurl.com/media4', 'post_id4', 1),
      ('media_id5', 'http://mediaurl.com/media5', 'post_id5', 1);
```

```
INSERT INTO Posts(ID, Time, Caption, IsAt, Expires)
VALUES('post_id1', TO_DATE('2023-10-16', 'YYYY-MM-DD'), 'beautiful day',
      'location_id1', TO_DATE('2023-12-16', 'YYYY-MM-DD')),
      ('post_id2', TO_DATE('2023-10-16', 'YYYY-MM-DD'), 'Friendsgiving',
      'location_id2', TO_DATE('2023-12-16', 'YYYY-MM-DD')),
      ('post_id3', TO_DATE('2023-10-16', 'YYYY-MM-DD'), 'Happy birthday to me',
      'location_id3', TO_DATE('2023-12-16', 'YYYY-MM-DD')),
      ('post_id4', TO_DATE('2023-10-16', 'YYYY-MM-DD'), 'Raincouver',
      'location_id4', TO_DATE('2023-12-16', 'YYYY-MM-DD')),
      ('post_id5', TO_DATE('2023-10-16', 'YYYY-MM-DD'), 'Me and my cats',
      'location_id5', TO_DATE('2023-12-16', 'YYYY-MM-DD'));
```

```
INSERT INTO Listings(ID, Title, Price, Offering)
VALUES('listing_id1', 'EVERYTHING MUST GO', 10, 'product_category_id1'),
      ('listing_id2', 'Brand new car', 10000, 'product_category_id2'),
      ('listing_id3', 'Elon Musk by Walter Isaacson - Pick up at UBC', 10,
      'product_category_id3'),
      ('listing_id4', 'Asus Monitor', 200, 'product_category_id4'),
      ('listing_id5', 'Lululemon Hoodie', 50, 'product_category_id5');
```

```
INSERT INTO Comments(ID, Time, Text, User, Post)
VALUES('comment_id1', TO_DATE('2023-10-16', 'YYYY-MM-DD'), 'Cute cats!',
'user_id1', 'post_id1'),
      ('comment_id2', TO_DATE('2023-10-16', 'YYYY-MM-DD'), 'Brooo', 'user_id2',
'post_id2'),
      ('comment_id3', TO_DATE('2023-10-16', 'YYYY-MM-DD'), 'dudee', 'user_id3',
'post_id3'),
      ('comment_id4', TO_DATE('2023-10-16', 'YYYY-MM-DD'), 'nice', 'user_id4',
'post_id4'),
      ('comment_id5', TO_DATE('2023-10-16', 'YYYY-MM-DD'), 'check your DM',
'user_id5', 'post_id5');
```

```
INSERT INTO ProductCategories(ID, Name)
VALUES('product_category_id1', 'Furniture'),
      ('product_category_id2', 'Vehicles'),
      ('product_category_id3', 'Books'),
      ('product_category_id4', 'Electronics'),
      ('product_category_id5', 'Clothes');
```