

Student Name: yiting wang  
Student ID: 13853462

### **SQL DDLs for Entities and their supporting tables**

```
DROP DATABASE IF EXISTS cs122a_hw2_entities;  
CREATE DATABASE cs122a_hw2_entities;  
USE cs122a_hw2_entities;
```

```
CREATE TABLE Users (  
    user_id INTEGER,  
    name_first VARCHAR(15) NOT NULL,  
    name_last VARCHAR(15) NOT NULL,  
    email VARCHAR(20) NOT NULL,  
    password VARCHAR(40) NOT NULL,  
    user_since DATE NOT NULL,  
    profile_pic_url VARCHAR(40),  
    address_country VARCHAR(40) NOT NULL ,  
    address_state VARCHAR(40) NOT NULL,  
    address_city VARCHAR(40) NOT NULL,  
    PRIMARY KEY (user_id)  
);
```

```
CREATE TABLE RawTweet (  
    tweet_id INTEGER,  
    content VARCHAR(200) NOT NULL,  
    PRIMARY KEY (tweet_id)  
);
```

```
CREATE TABLE Tweeter (  
    tweeter_id INTEGER,  
    followers_count INTEGER NOT NULL,  
    display_name VARCHAR(20) NOT NULL,  
    verified BOOL NOT NULL,  
    handle INTEGER NOT NULL,  
    PRIMARY KEY (tweeter_id)  
);
```

```
CREATE TABLE Evidence (  
    ev_id INTEGER,  
    url VARCHAR(40) NOT NULL,  
    PRIMARY KEY (ev_id)  
);
```

```
CREATE TABLE Checker (  
    user_id INTEGER,  
    checker_since DATE NOT NULL,  
    PRIMARY KEY (user_id),  
    FOREIGN KEY (user_id)  
        REFERENCES Users (user_id)  
        ON DELETE CASCADE  
);
```

```
CREATE TABLE Checker_Expertise (  
    user_id INTEGER,  
    expertise VARCHAR(20),  
    PRIMARY KEY (user_id),  
    FOREIGN KEY (user_id)  
        REFERENCES Users (user_id)  
        ON DELETE CASCADE  
);
```

```
CREATE TABLE Checker_Phone (  
    user_id INTEGER,  
    number VARCHAR(20),  
    type ENUM("HOME","OFFICE","MOBILE"),  
    PRIMARY KEY (user_id,number),  
    FOREIGN KEY (user_id)  
        REFERENCES Users (user_id)  
        ON DELETE CASCADE  
);
```

-- by,of relationship is folded here

```
CREATE TABLE Verification (  
    ver_id INTEGER,  
    user_id INTEGER,  
    tweet_id INTEGER,  
    comment VARCHAR(200),  
    verified_on DATE NOT NULL,  
    PRIMARY KEY (ver_id),  
    FOREIGN KEY (user_id)
```

```

        REFERENCES Users (user_id)
        ON DELETE CASCADE,
    FOREIGN KEY (tweet_id)
        REFERENCES RawTweet (tweet_id)
);

-- fold post, comes_from, of,quotes, replies_to, relationship
CREATE TABLE Tweet (
    tweet_id INTEGER,
    tweeter_id INTEGER ,
    posting_datetime DATE NOT NULL,
    posting_location_longitude DECIMAL(3,2),
    posting_location_latitude DECIMAL(3,2),
    tweet_text VARCHAR(200) NOT NULL,
    popularity DECIMAL(10,2) NOT NULL,
    quality INTEGER NOT NULL,
    quoted_Tweet_id INTEGER,
    quote_id INTEGER ,
    replied_tweet_id INTEGER,
    reply_id INTEGER,
    PRIMARY KEY (tweet_id),
    FOREIGN KEY (tweet_id)
        REFERENCES Rawtweet (tweet_id),
    FOREIGN KEY (tweeter_id)
        REFERENCES Tweeter (tweeter_id),
    FOREIGN KEY (quoted_Tweet_id)
        REFERENCES RawTweet (tweet_id),
    FOREIGN KEY (quote_id)
        REFERENCES Tweet (quoted_Tweet_id)
        ON DELETE SET NULL,
    FOREIGN KEY (replied_tweet_id)
        REFERENCES RawTweet (tweet_id),
    FOREIGN KEY (reply_id)
        REFERENCES Tweet (replied_tweet_id)
        ON DELETE SET NULL
);

CREATE TABLE Tweet_hash_tags (
    tweet_id INTEGER,
    hash_tags VARCHAR(20),
    PRIMARY KEY (tweet_id,hash_tags),
    FOREIGN KEY (tweet_id)
        REFERENCES Rawtweet (tweet_id)
);

```

## SQL DDLs for Relationships

```
CREATE TABLE About (  
    tweet_id INTEGER,  
    ev_id INTEGER,  
    PRIMARY KEY (tweet_id,ev_id),  
    FOREIGN KEY (tweet_id)  
        REFERENCES RawTweet (tweet_id),  
    FOREIGN KEY (ev_id)  
        REFERENCES Evidence (ev_id)  
);
```

```
CREATE TABLE VerifiedUsing(  
    ver_id INTEGER,  
    ev_id INTEGER,  
    PRIMARY KEY (ver_id,ev_id),  
    FOREIGN KEY (ver_id)  
        REFERENCES Verification (ver_id),  
    FOREIGN KEY (ev_id)  
        REFERENCES Evidence (ev_id)  
);
```

```
CREATE TABLE EvidenceFrom (  
    user_id INTEGER,  
    ev_id INTEGER,  
    PRIMARY KEY (user_id,ev_id),  
    FOREIGN KEY (user_id)  
        REFERENCES Users (user_id)  
        ON DELETE CASCADE,  
    FOREIGN KEY (ev_id)  
        REFERENCES Evidence (ev_id)  
);
```