Student Name: Joowon Suh
Student ID:44414081

SQL DDLs for Entities and their supporting tables

```
CREATE TABLE User(
userid INTEGER,
name_first VARCHAR(50),
name_last VARCHAR(50),
email VARCHAR(50),
password VARCHAR(10),
user_since DATE,
profile_pic_url VARCHAR(200),
address_country VARCHAR(100),
address_state VARCHAR(100),
address_city VARCHAR(100),
PRIMARY KEY (userid));

CREATE TABLE User_Checker(
userid INTEGER,
name_first VARCHAR(50),
name_last VARCHAR(50),
email VARCHAR(50),
password VARCHAR(10),
user_since DATE,
profile_pic_url VARCHAR(200),
address_country VARCHAR(100),
address_state VARCHAR(100),
address_city VARCHAR(100),
checkersince DATE,
PRIMARY KEY (userid),
FOREIGN KEY (userid) REFERENCES User ON DELETE CASCADE);

CREATE TABLE Checker_expertise(
userid INTEGER,
expertise VARCHAR(100),
PRIMARY KEY (userid, expertise),
FOREIGN KEY (userid) REFERENCES User_Checker ON DELETE CASCADE);

CREATE TABLE Checker_phone(
userid INTEGER,
phone_type ENUM('home', 'office', 'mobile'),
phone_number VARCHAR(14),
PRIMARY KEY (userid, phone_type, phone_number),
FOREIGN KEY (userid) REFERENCES User_Checker ON DELETE CASCADE);
```

```
CREATE TABLE verification(
verid INTEGER,
comment VARCHAR(200),
verifiedon DATE,
PRIMARY KEY (verid));

CREATE TABLE Evidence(
evid INTEGER,
url VARCHAR(50),
PRIMARY KEY (evid));

CREATE TABLE RawTweet(
tweetid INTEGER,
content VARCHAR(200),
PRIMARY KEY (tweetid));

CREATE TABLE Tweet(
tweet_id INTEGER,
tweettext VARCHAR(200),
quote_id INTEGER,
quoted_times INTEGER,
reply_id INTEGER,
replied_times INTEGER,
evidences INTEGER,
PRIMARY KEY (tweeterid),
FOREIGN KEY (quote_id) REFERENCES Tweet,
FOREIGN KEY (reply_id) REFERENCES Tweet);

CREATE VIEW TweetView(tweet_id, tweeettext, popularity, quality)
AS SELECT T.tweet_id, T.tweettext, 0.4*(T.quoted_times)*0.6*(T.replied_times),
T.evidences
FROM Tweet T;

CREATE TABLE Tweet_hashtags(
tweet_id INTEGER,
hashtags VARCHAR(50),
PRIMARY KEY (tweet_id,hashtags),
FOREIGN KEY (tweet_id) REFERENCES Tweet);

CREATE TABLE Tweeter(
tweeterid INTEGER,
followers_count INTEGER,
handle VARCHAR(50),
verified BOOL,
display_name VARCHAR(20),
PRIMARY KEY (tweetid) );
```

SQL DDLs for Relationships

```
CREATE TABLE comes_from(
tweetid INTEGER,
tweet_id INTEGER NOT NULL,
PRIMARY KEY (tweeid, tweet_id),
FOREIGN KEY (tweetid) REFERENCES RawTweet,
FOREIGN KEY (tweet_id) REFERENCES Tweet);

CREATE TABLE posts(
tweeterid INTEGER NOT NULL,
tweet_id INTEGER NOT NULL,
posting_datetime DATETIME,
posting_location_longitude INTEGER,
posting_location_latitude INTEGER,
PRIMARY KEY (tweet_id),
FOREIGN KEY (tweet_id )REFERENCES Tweet,
FOREIGN KEY (tweeterid )REFERENCES Tweeter);

CREATE TABLE EvidenceFrom(
userid INTEGER,
evid INTEGER,
PRIMARY KEY(userid, evid),
FOREIGN KEY (userid) REFERENCES User,
FOREIGN KEY (evid) REFERENCES evidence
);
CREATE TABLE about(
evid INTEGER NOT NULL,
tweet_id INTEGER,
PRIMARY KEY (evid, tweet_id),
FOREIGN KEY (evid) REFERENCES evid,
FOREIGN KEY(tweet_id) REFERENCES Tweet
);
CREATE TABLE VerifiedUsing(
verid INTEGER NOT NULL,
evid INTEGER,
PRIMARY KEY (evid, verid),
FOREIGN KEY(evid) REFERENCES evid,
FOREIGN KEY(verid) REFERENCES verification
);
CREATE TABLE VerifiedOf(
verid INTEGER NOT NULL,
tweet_id INTEGER,
PRIMARY KEY (verid),
FOREIGN KEY(tweet_id) REFERENCES tweet,
FOREIGN KEY(verid) REFERENCES verification
);

CREATE TABLE VerifiedBy(
verid INTEGER NOT NULL,
userid INTEGER,
PRIMARY KEY (verid),
FOREIGN KEY(userid) REFERENCES User_Checker,
FOREIGN KEY(verid) REFERENCES verification
);
```