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

);