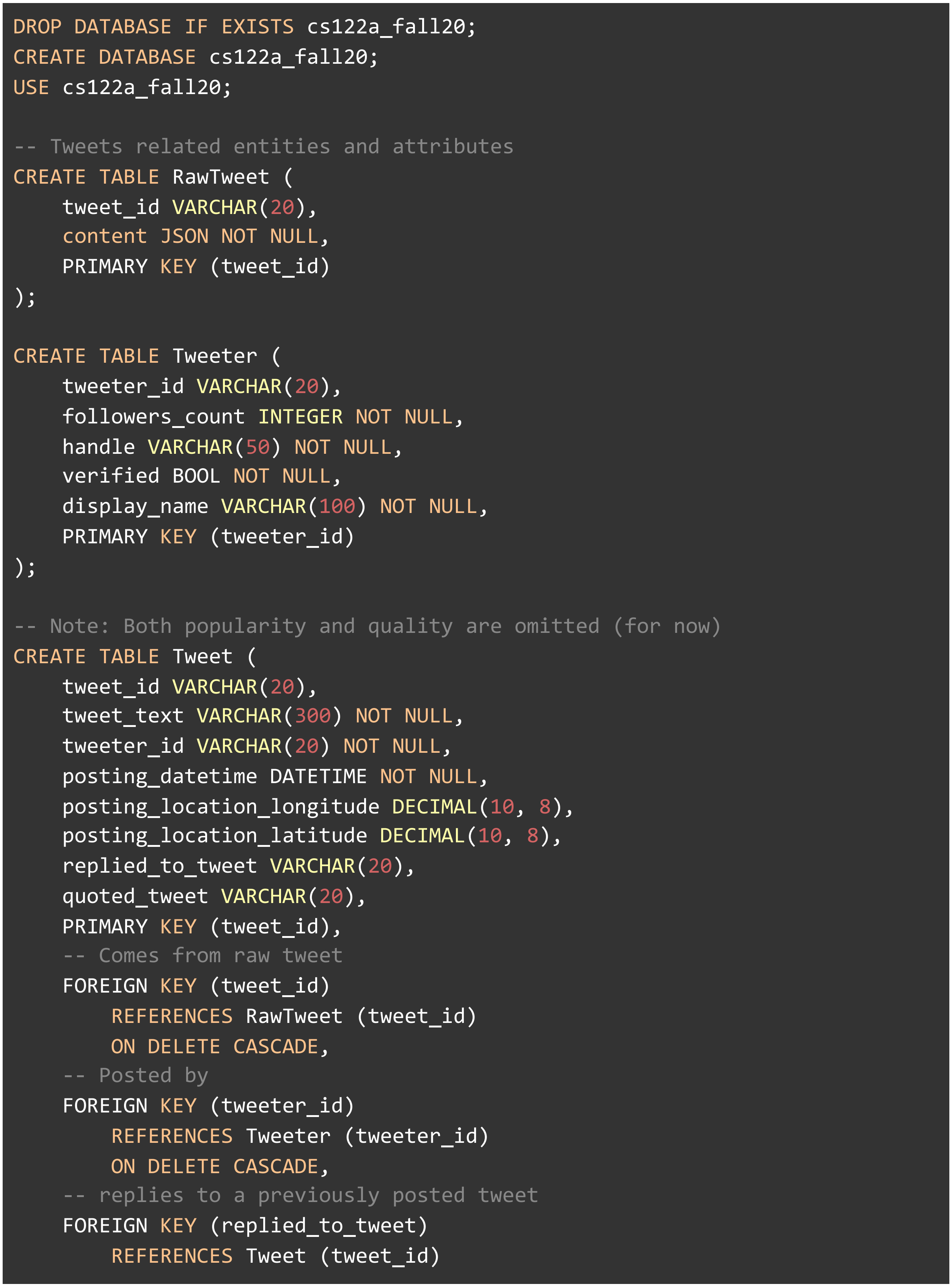
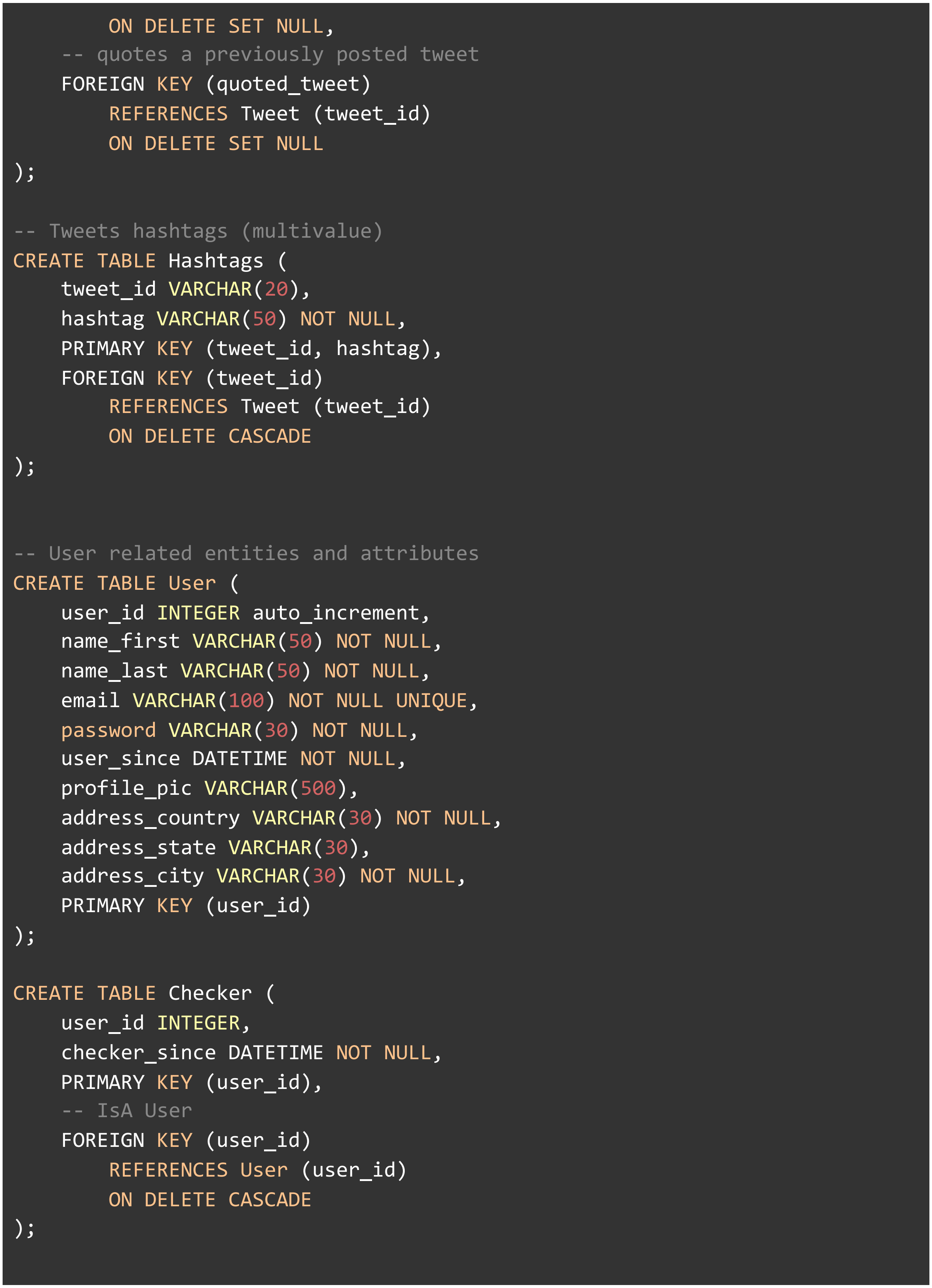


```
DROP DATABASE IF EXISTS cs122a_fall20;
CREATE DATABASE cs122a_fall20;
USE cs122a_fall20;

-- Tweets related entities and attributes
CREATE TABLE RawTweet (
    tweet_id VARCHAR(20),
    content JSON NOT NULL,
    PRIMARY KEY (tweet_id)
);

CREATE TABLE Tweeter (
    tweeter_id VARCHAR(20),
    followers_count INTEGER NOT NULL,
    handle VARCHAR(50) NOT NULL,
    verified BOOL NOT NULL,
    display_name VARCHAR(100) NOT NULL,
    PRIMARY KEY (tweeter_id)
);

-- Note: Both popularity and quality are omitted (for now)
CREATE TABLE Tweet (
    tweet_id VARCHAR(20),
    tweet_text VARCHAR(300) NOT NULL,
    tweeter_id VARCHAR(20) NOT NULL,
    posting_datetime DATETIME NOT NULL,
    posting_location_longitude DECIMAL(10, 8),
    posting_location_latitude DECIMAL(10, 8),
    replied_to_tweet VARCHAR(20),
    quoted_tweet VARCHAR(20),
    PRIMARY KEY (tweet_id),
    -- Comes from raw tweet
    FOREIGN KEY (tweet_id)
        REFERENCES RawTweet (tweet_id)
        ON DELETE CASCADE,
    -- Posted by
    FOREIGN KEY (tweeter_id)
        REFERENCES Tweeter (tweeter_id)
        ON DELETE CASCADE,
    -- replies to a previously posted tweet
    FOREIGN KEY (replied_to_tweet)
        REFERENCES Tweet (tweet_id)
```



```sql
        ON DELETE SET NULL,
    -- quotes a previously posted tweet
    FOREIGN KEY (quoted_tweet)
        REFERENCES Tweet (tweet_id)
        ON DELETE SET NULL
);

-- Tweets hashtags (multivalue)
CREATE TABLE Hashtags (
    tweet_id VARCHAR(20),
    hashtag VARCHAR(50) NOT NULL,
    PRIMARY KEY (tweet_id, hashtag),
    FOREIGN KEY (tweet_id)
        REFERENCES Tweet (tweet_id)
        ON DELETE CASCADE
);


-- User related entities and attributes
CREATE TABLE User (
    user_id INTEGER auto_increment,
    name_first VARCHAR(50) NOT NULL,
    name_last VARCHAR(50) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    password VARCHAR(30) NOT NULL,
    user_since DATETIME NOT NULL,
    profile_pic VARCHAR(500),
    address_country VARCHAR(30) NOT NULL,
    address_state VARCHAR(30),
    address_city VARCHAR(30) NOT NULL,
    PRIMARY KEY (user_id)
);

CREATE TABLE Checker (
    user_id INTEGER,
    checker_since DATETIME NOT NULL,
    PRIMARY KEY (user_id),
    -- IsA User
    FOREIGN KEY (user_id)
        REFERENCES User (user_id)
        ON DELETE CASCADE
);
```

```
-- Checker phone numbers (multivalue)
CREATE TABLE Phone (
    user_id INTEGER,
    -- type
    kind ENUM('HOME', 'OFFICE', 'MOBILE') NOT NULL,
    number VARCHAR(20),
    PRIMARY KEY (user_id, number),
    FOREIGN KEY (user_id)
        REFERENCES Checker (user_id)
        ON DELETE CASCADE
);

-- Checker expertise (multivalue)
CREATE TABLE Expertise (
    user_id INTEGER,
    domain VARCHAR(30),
    PRIMARY KEY (user_id, domain),
    FOREIGN KEY (user_id)
        REFERENCES Checker (user_id)
        ON DELETE CASCADE
);


-- Fact checking entities
CREATE TABLE Evidence (
    ev_id INTEGER auto_increment,
    url VARCHAR(500) NOT NULL,
    PRIMARY KEY (ev_id)
);

CREATE TABLE Verification (
    ver_id INTEGER auto_increment,
    comment VARCHAR(500) NOT NULL,
    verified_on DATETIME NOT NULL,
    user_id INTEGER NOT NULL,
    tweet_id VARCHAR(20) NOT NULL,
    PRIMARY KEY (ver_id),
    -- A verification BY a checker
    FOREIGN KEY (user_id)
        REFERENCES Checker (user_id)
        ON DELETE CASCADE,
    -- A verification OF a tweet
    FOREIGN KEY (tweet_id)
```

```
        REFERENCES Tweet (tweet_id)
        ON DELETE CASCADE
);

-- -------------------
-- Relationships
-- -------------------

-- User <--from--> Evidence
CREATE TABLE EvidenceFrom(
    user_id INTEGER,
    ev_id INTEGER,
    PRIMARY KEY (user_id, ev_id),
    FOREIGN KEY (user_id)
        REFERENCES User (user_id)
        ON DELETE CASCADE,
    FOREIGN KEY (ev_id)
        REFERENCES Evidence (ev_id)
        ON DELETE CASCADE
);

-- Evidence <--about--> Tweet
CREATE TABLE About(
    tweet_id VARCHAR(20),
    ev_id INTEGER,
    PRIMARY KEY (tweet_id, ev_id),
    FOREIGN KEY (tweet_id)
        REFERENCES Tweet (tweet_id)
        ON DELETE CASCADE,
    FOREIGN KEY (ev_id)
        REFERENCES Evidence (ev_id)
        ON DELETE CASCADE
);

-- Evidence <--using--> Verification
CREATE TABLE VerifiedUsing(
    ver_id INTEGER,
    ev_id INTEGER,
    PRIMARY KEY (ver_id, ev_id),
    FOREIGN KEY (ver_id)
        REFERENCES Verification (ver_id)
        ON DELETE CASCADE,
    FOREIGN KEY (ev_id)
```

```
        REFERENCES Evidence (ev_id)
        ON DELETE CASCADE
);
```