SQL Queries for database setup:

I used the following SQL queries to create the tables with their respective constraints. Given below is how I achieved the table creation in MS SQL Server Management Studio.

```
DROP TABLE IF EXISTS business
CREATE TABLE business
( business_id char(22) PRIMARY
KEY, name varchar(60) NOT NULL,
address varchar(75), city
varchar(30) NOT NULL, postal_code
varchar(7),
stars decimal(2,1) CHECK (stars >= 1 AND stars <= 5), review_count
int DEFAULT 0 CHECK (review_count >= 0)
);

DROP TABLE IF EXISTS user_yelp
CREATE TABLE user_yelp
( user_id char(22) PRIMARY
KEY, name varchar(35) NOT
NULL,
review_count int DEFAULT 0 CHECK(review_count >=0),
yelping_since DATETIME DEFAULT GETDATE() NOT NULL,
useful int DEFAULT 0 CHECK (useful >= 0), funny int
DEFAULT 0 CHECK (funny >=0), cool int DEFAULT 0
CHECK (cool >=0), fans int DEFAULT 0 CHECK (fans >=
0),
average_stars decimal(3,2) CHECK (average_stars >= 1 AND average_stars <= 5) );

DROP TABLE IF EXISTS checkin
CREATE TABLE checkin
( checkin_id INT PRIMARY KEY,
business_id CHAR(22) NOT NULL,
    date DATETIME NOT NULL DEFAULT GETDATE(),
    FOREIGN KEY (business_id) REFERENCES business(business_id) ON DELETE NO ACTION ON
UPDATE NO ACTION
);

DROP TABLE IF EXISTS tip
CREATE TABLE tip
(
tip_id INT PRIMARY KEY, user_id CHAR(22)
NOT NULL, business_id CHAR(22) NOT NULL,
date DATETIME NOT NULL DEFAULT GETDATE(),
compliment_count INT DEFAULT 0 CHECK (compliment_count >= 0),
FOREIGN KEY (user_id) REFERENCES user_yelp(user_id) ON DELETE NO ACTION ON UPDATE NO
ACTION,
FOREIGN KEY (business_id) REFERENCES business(business_id) ON DELETE NO ACTION ON UPDATE
NO ACTION
);

DROP TABLE IF EXISTS friendship
CREATE TABLE friendship
```

```
( user_id CHAR(22),
friend CHAR(22),
    PRIMARY KEY (user_id, friend),
    FOREIGN KEY (user_id) REFERENCES user_yelp(user_id) ON DELETE NO ACTION ON UPDATE NO
ACTION,
    FOREIGN KEY (friend) REFERENCES user_yelp(user_id) ON DELETE NO ACTION ON UPDATE NO
ACTION
);

DROP TABLE IF EXISTS review
CREATE TABLE review
(      review_id CHAR(22) PRIMARY
KEY,      user_id CHAR(22) NOT NULL,
business_id CHAR(22) NOT NULL,
    stars INT NOT NULL CHECK (stars >= 1 AND stars <= 5),
useful INT DEFAULT 0 CHECK (useful >= 0),      funny INT
DEFAULT 0 CHECK (funny >= 0),      cool INT DEFAULT 0
CHECK (cool >= 0),      date DATETIME DEFAULT GETDATE(),
    FOREIGN KEY (user_id) REFERENCES user_yelp(user_id) ON DELETE NO ACTION ON UPDATE NO
ACTION,
    FOREIGN KEY (business_id) REFERENCES business(business_id) ON DELETE NO ACTION ON
UPDATE NO ACTION
);
```

Importing JSON data into SQL tables:

The insertion into the tables from the files was done using the BULK INSERT method, I followed an insertion order of the following: USER_YELP -> BUSINESS -> CHECKIN -> TIP -> REVIEW -> FRIENDSHIP. This was done because tables like tip, review and others involved foreign key references from the tables user_yelp and business. Hence, the tables user_yelp and business had to be constructed first in order to hold the foreign key relation and to execute successful referencing between tables in the database.

```
BULK INSERT dbo.user_yelp FROM 'd:\userdata\asa_assn3\user_yelp.csv' WITH
(FIELDTERMINATOR = ',', ROWTERMINATOR = '\n', FIRSTROW = 2);

BULK INSERT dbo.business FROM 'd:\userdata\asa_assn3\business.csv' WITH (FIELDTERMINATOR
= ',', ROWTERMINATOR = '\n', FIRSTROW = 2);

BULK INSERT dbo.checkin FROM 'd:\userdata\asa_assn3\checkin.csv' WITH (FIELDTERMINATOR =
',', ROWTERMINATOR = '\n', FIRSTROW = 2);

BULK INSERT dbo.tip FROM 'd:\userdata\asa_assn3\tip.csv' WITH (FIELDTERMINATOR = ',',
ROWTERMINATOR = '\n', FIRSTROW = 2);

BULK INSERT dbo.review FROM 'd:\userdata\asa_assn3\review.csv' WITH (FIELDTERMINATOR =
',', ROWTERMINATOR = '\n', FIRSTROW = 2);

BULK INSERT dbo.friendship FROM 'd:\userdata\asa_assn3\friendship.csv' WITH
(FIELDTERMINATOR = ',', ROWTERMINATOR = '\n', FIRSTROW = 2);
```