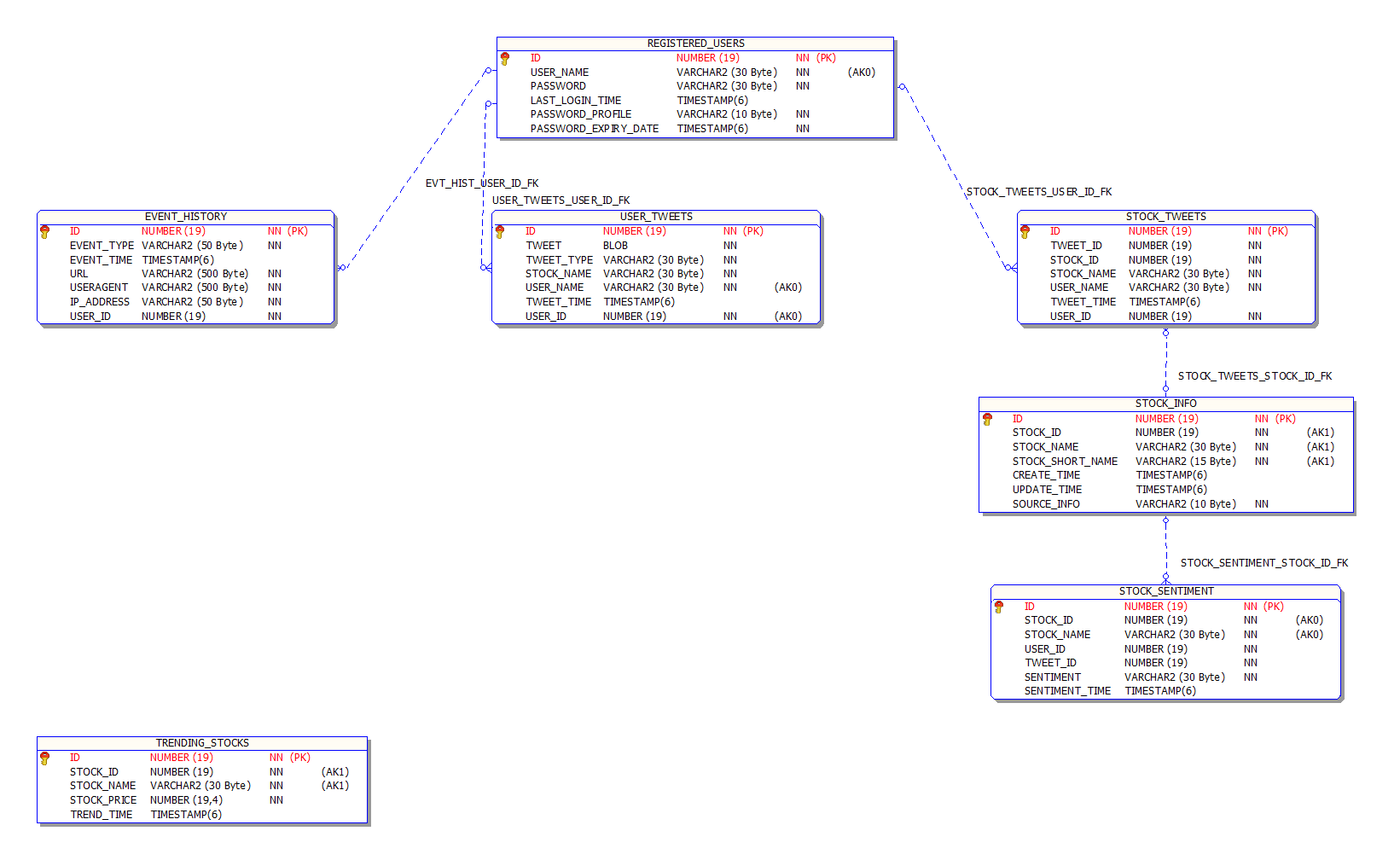
ER Diagram.



Scripts.

CREATE TABLE tsocial.registered\_users

(

ID NUMBER (19) NOT NULL,

USER\_NAME VARCHAR2 (30) NOT NULL,

PASSWORD VARCHAR2 (30) NOT NULL,

LAST\_LOGIN\_TIME TIMESTAMP (6) DEFAULT SYSTIMESTAMP,

PASSWORD\_PROFILE VARCHAR2 (10) NOT NULL,

PASSWORD\_EXPIRY\_DATE TIMESTAMP (6) NOT NULL,

CONSTRAINT reg\_users\_pk PRIMARY KEY (ID),

CONSTRAINT reg\_users\_user\_name\_uk UNIQUE (USER\_NAME)

);

COMMENT ON TABLE tsocial.registered\_users IS

'Table holds the information of all users.';

COMMENT ON COLUMN tsocial.registered\_users.id IS 'Primary Key of the table.';

COMMENT ON COLUMN tsocial.registered\_users.user\_name IS

'User name of the trading social account. It is a unique identifier.';

COMMENT ON COLUMN tsocial.registered\_users.password IS

'Encrypted form of the password.';

COMMENT ON COLUMN tsocial.registered\_users.last\_login\_time IS

'Timestamp of login with a precision of microseconds.';

COMMENT ON COLUMN tsocial.registered\_users.password\_profile IS

'Depending on the values here, a user''s password expiry is controlled. Eg. A value of ''FOREVER'' means a user''s password cannot expire.';

COMMENT ON COLUMN tsocial.registered\_users.password\_expiry\_date IS

'Date of expiry of the password.';

-- USER\_NAME needs a unique key, ID needs to be a PK.

CREATE SEQUENCE tsocial.registered\_users\_seq;

CREATE TABLE tsocial.event\_history

(

ID NUMBER (19) NOT NULL,

EVENT\_TYPE VARCHAR2 (50) NOT NULL,

EVENT\_TIME TIMESTAMP (6) DEFAULT SYSTIMESTAMP,

URL VARCHAR2 (500) NOT NULL,

USERAGENT VARCHAR2 (500) NOT NULL,

IP\_ADDRESS VARCHAR2 (50) NOT NULL,

USER\_ID NUMBER (19) NOT NULL,

CONSTRAINT event\_history\_pk PRIMARY KEY (ID),

CONSTRAINT evt\_hist\_user\_id\_fk FOREIGN KEY

(user\_id)

REFERENCES tsocial.registered\_users (ID)

)

PARTITION BY RANGE (EVENT\_TIME)

INTERVAL ( NUMTOYMINTERVAL (1, 'MONTH') )

(PARTITION P1

VALUES LESS THAN (TO\_DATE ('2000-01-01', 'YYYY-MM-DD')));

COMMENT ON TABLE tsocial.event\_history IS

'Table holds the history of event information of all users.';

COMMENT ON COLUMN tsocial.event\_history.id IS 'Primary Key of the table.';

COMMENT ON COLUMN tsocial.event\_history.event\_time IS 'Timestamp of event.';

COMMENT ON COLUMN tsocial.event\_history.url IS '';

COMMENT ON COLUMN tsocial.event\_history.useragent IS

'Captures the useragent from which end user login is attempted.';

COMMENT ON COLUMN tsocial.event\_history.user\_id IS

'User identifier. Foreign Key to tsocial.registered\_users.id.';

CREATE SEQUENCE tsocial.event\_history\_seq;

CREATE TABLE tsocial.user\_tweets

(

ID NUMBER (19) NOT NULL,

TWEET BLOB NOT NULL,

TWEET\_TYPE VARCHAR2 (30) NOT NULL,

STOCK\_NAME VARCHAR2 (30) NOT NULL,

USER\_NAME VARCHAR2 (30) NOT NULL,

TWEET\_TIME TIMESTAMP (6) DEFAULT SYSTIMESTAMP,

USER\_ID NUMBER (19) NOT NULL,

CONSTRAINT user\_tweets\_pk PRIMARY KEY (ID),

CONSTRAINT user\_tweets\_user\_uk UNIQUE (USER\_ID, USER\_NAME),

CONSTRAINT user\_tweets\_user\_id\_fk FOREIGN KEY

(user\_id)

REFERENCES tsocial.registered\_users (ID)

)

PARTITION BY RANGE (TWEET\_TIME)

INTERVAL ( NUMTOYMINTERVAL (1, 'MONTH') )

(PARTITION P1

VALUES LESS THAN (TO\_DATE ('2000-01-01', 'YYYY-MM-DD')));

COMMENT ON TABLE tsocial.user\_tweets IS

'Table to hold all the tweets of a user.';

COMMENT ON COLUMN tsocial.user\_tweets.id IS 'Primary Key of the table.';

COMMENT ON COLUMN tsocial.user\_tweets.tweet IS

'The content tweeted by the user.';

COMMENT ON COLUMN tsocial.user\_tweets.tweet\_type IS

'The type of tweet of the user.';

COMMENT ON COLUMN tsocial.user\_tweets.stock\_name IS

'The name of the stock tweeted by the user.';

COMMENT ON COLUMN tsocial.user\_tweets.user\_name IS

'User name of the tweeter. Same as tsocial.registered\_users.user\_name.';

COMMENT ON COLUMN tsocial.user\_tweets.tweet\_time IS 'Timestamp of the tweet.';

COMMENT ON COLUMN tsocial.user\_tweets.user\_id IS

'User identifier. Foreign Key to tsocial.registered\_users.id.';

CREATE SEQUENCE tsocial.user\_tweets\_seq;

-- unique key on user\_id and user\_name

CREATE TABLE tsocial.stock\_info

(

ID NUMBER (19) NOT NULL,

STOCK\_ID NUMBER (19) NOT NULL,

STOCK\_NAME VARCHAR2 (30) NOT NULL,

STOCK\_SHORT\_NAME VARCHAR2 (15) NOT NULL,

CREATE\_TIME TIMESTAMP (6) DEFAULT SYSTIMESTAMP,

UPDATE\_TIME TIMESTAMP (6),

SOURCE\_INFO VARCHAR2 (10) NOT NULL,

CONSTRAINT stock\_info\_pk PRIMARY KEY (ID),

CONSTRAINT stock\_info\_uk UNIQUE (stock\_id, stock\_name, stock\_short\_name),

CONSTRAINT stock\_info\_ck CHECK (source\_info IN ('USER', 'SYSTEM'))

);

COMMENT ON TABLE tsocial.stock\_info IS

'Table to hold information about the publicly traded stocks in the market.';

COMMENT ON COLUMN tsocial.stock\_info.id IS 'Primary Key of the table.';

COMMENT ON COLUMN tsocial.stock\_info.stock\_id IS

'The identifier of the stock.';

COMMENT ON COLUMN tsocial.stock\_info.stock\_name IS

'The name of the stock in the market.';

COMMENT ON COLUMN tsocial.stock\_info.stock\_short\_name IS

'The short name of the stock in the market.';

COMMENT ON COLUMN tsocial.stock\_info.create\_time IS

'The timestamp of creation of a new stock entry.';

COMMENT ON COLUMN tsocial.stock\_info.update\_time IS

'The timestamp of the updation of a stock entry.';

COMMENT ON COLUMN tsocial.stock\_info.source\_info IS

'The source of the stock data, i.e, whether USER or SYSTEM generated.';

CREATE SEQUENCE tsocial.stock\_info\_seq;

-- unique constraint on stock\_id, stock\_name, stock\_short\_name

-- check constraint on source\_info.

CREATE TABLE tsocial.stock\_tweets

(

ID NUMBER (19),

TWEET\_ID NUMBER (19) NOT NULL,

STOCK\_ID NUMBER (19) NOT NULL,

STOCK\_NAME VARCHAR2 (30) NOT NULL,

USER\_NAME VARCHAR2 (30) NOT NULL,

TWEET\_TIME TIMESTAMP (6) DEFAULT SYSTIMESTAMP,

USER\_ID NUMBER (19) NOT NULL,

CONSTRAINT stock\_tweets\_pk PRIMARY KEY (ID),

CONSTRAINT stock\_tweets\_stock\_id\_fk FOREIGN KEY

(stock\_id)

REFERENCES tsocial.stock\_info (ID),

CONSTRAINT stock\_tweets\_user\_id\_fk FOREIGN KEY

(user\_id)

REFERENCES tsocial.registered\_users (ID)

)

PARTITION BY RANGE (TWEET\_TIME)

INTERVAL ( NUMTOYMINTERVAL (1, 'MONTH') )

(PARTITION P1

VALUES LESS THAN (TO\_DATE ('2000-01-01', 'YYYY-MM-DD')));

COMMENT ON TABLE tsocial.stock\_tweets IS

'Table to hold all the tweets of a stock.';

COMMENT ON COLUMN tsocial.stock\_tweets.id IS 'Primary Key of the table.';

COMMENT ON COLUMN tsocial.stock\_tweets.tweet\_id IS

'The identifier of the stock tweet.';

COMMENT ON COLUMN tsocial.stock\_tweets.stock\_id IS

'The identifier of the stock. Foreign key to tsocial.stock\_info.id.';

COMMENT ON COLUMN tsocial.stock\_tweets.stock\_name IS 'The name of the stock.';

COMMENT ON COLUMN tsocial.stock\_tweets.user\_name IS

'User name of the tweeter.';

COMMENT ON COLUMN tsocial.stock\_tweets.tweet\_time IS

'Timestamp of the tweet.';

COMMENT ON COLUMN tsocial.stock\_tweets.user\_id IS

'User identifier. Foreign Key to tsocial.registered\_users.id.';

CREATE TABLE tsocial.stock\_sentiment

(

ID NUMBER (19) NOT NULL,

STOCK\_ID NUMBER (19) NOT NULL,

STOCK\_NAME VARCHAR2 (30) NOT NULL,

USER\_ID NUMBER (19) NOT NULL,

TWEET\_ID NUMBER (19) NOT NULL,

SENTIMENT VARCHAR2 (30) NOT NULL,

SENTIMENT\_TIME TIMESTAMP (6) DEFAULT SYSTIMESTAMP,

CONSTRAINT stock\_sentiment\_pk PRIMARY KEY (ID),

CONSTRAINT stock\_sentiment\_stock\_uk UNIQUE (stock\_id, stock\_name),

CONSTRAINT stock\_sentiment\_stock\_id\_fk FOREIGN KEY

(stock\_id)

REFERENCES tsocial.stock\_info (id),

CONSTRAINT stock\_sentiment\_ck CHECK

(sentiment IN ('BEARISH', 'BULLISH', 'NEUTRAL'))

)

PARTITION BY RANGE (SENTIMENT\_TIME)

INTERVAL ( NUMTOYMINTERVAL (1, 'MONTH') )

(PARTITION P1

VALUES LESS THAN (TO\_DATE ('2000-01-01', 'YYYY-MM-DD')));

COMMENT ON TABLE tsocial.stock\_sentiment IS

'Table to hold the sentiments of a stock mapped to a user.';

COMMENT ON COLUMN tsocial.stock\_sentiment.id IS 'Primary Key of the table.';

COMMENT ON COLUMN tsocial.stock\_sentiment.stock\_id IS

'The identifier of the stock.';

COMMENT ON COLUMN tsocial.stock\_sentiment.stock\_name IS

'The name of the stock on which the user expresses his sentiments.';

COMMENT ON COLUMN tsocial.stock\_sentiment.user\_id IS 'User identifier.';

COMMENT ON COLUMN tsocial.stock\_sentiment.tweet\_id IS 'Tweet identifier.';

COMMENT ON COLUMN tsocial.stock\_sentiment.sentiment IS

'Sentiment of the stock, i.e, whether BEARISH, BULLISH or NEUTRAL.';

COMMENT ON COLUMN tsocial.stock\_sentiment.sentiment\_time IS

'Timestamp of the sentiment.';

CREATE SEQUENCE tsocial.stock\_sentiment\_seq;

-- check constraint on sentiment.

-- unique key on?

CREATE TABLE tsocial.trending\_stocks

(

ID NUMBER (19) NOT NULL,

STOCK\_ID NUMBER (19) NOT NULL,

STOCK\_NAME VARCHAR2 (30) NOT NULL,

STOCK\_PRICE NUMBER (19, 4) NOT NULL,

TREND\_TIME TIMESTAMP (6) DEFAULT SYSTIMESTAMP,

CONSTRAINT trending\_stocks\_pk PRIMARY KEY (ID),

CONSTRAINT trending\_stocks\_stock\_uk UNIQUE (STOCK\_ID, STOCK\_NAME)

)

PARTITION BY RANGE (TREND\_TIME)

INTERVAL ( NUMTODSINTERVAL (1, 'DAY') )

(PARTITION P1

VALUES LESS THAN (TO\_DATE ('2000-01-01', 'YYYY-MM-DD')));

COMMENT ON TABLE tsocial.trending\_stocks IS

'Table to hold information about the trending stocks in the market.';

COMMENT ON COLUMN tsocial.trending\_stocks.id IS 'Primary Key of the table.';

COMMENT ON COLUMN tsocial.trending\_stocks.stock\_id IS

'The identifier of the stock.';

COMMENT ON COLUMN tsocial.trending\_stocks.stock\_name IS

'The name of the stock in the market.';

COMMENT ON COLUMN tsocial.trending\_stocks.stock\_price IS

'The realtime price of the stock in the market.';

COMMENT ON COLUMN tsocial.trending\_stocks.trend\_time IS

'The timestamp of the realtime price of the stock in the market.';

CREATE SEQUENCE tsocial.trending\_stocks\_seq;

-- unique constraint on stock\_id, stock\_name