Assignment 1

By

Rachael Joan Dias

0651897

Addressed to,

Professor Brian Srivastava

Trent University

AMOD-5410H-A-2019GW-PTBO Big Data

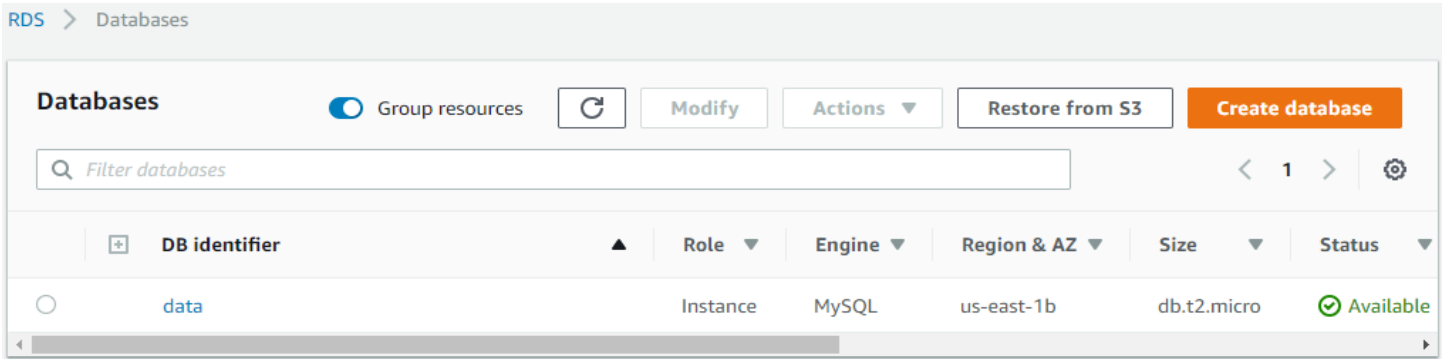
Objective:

The main goal of this assignment is to demonstrate data gathering from 2 different source using APIs, storage and retrieval on a MySQL database. In order to demonstrate this, I extracted tweets containing information related to cryptocurrencies which I was then able to link to the actual market price of the cryptocurrencies.

Streaming tweets related to cryptocurrencies such as Bitcoin, Ripple, Ethereum etc. were collected using Twitter's streaming API and stored on a MySQL database on a Linux server on AWS. Similarly, from another API (coinmarketcap.com) data indicating the latest price and ranking of the cryptocurrencies were stored on the AWS MySQL database.

Step 1: Database Creation

I first created a database "data" on AWS as you see from the below screenshot.



Step 2: Table Structures/Schemas

In the next step, I created tables for the data from Twitter and coinmarketcap.com. Below is a screenshot of the DDL statements used to create the tables. "Tweets" for twitter data and "crypto" for coinmarketcap.com data.

```
1  CREATE DATABASE IF NOT EXISTS data;
2  USE data;
3
4  CREATE TABLE tweets (
5  id int(11) NOT NULL AUTO_INCREMENT,
6  tweet_id varchar(250) DEFAULT NULL,
7  name varchar(128) DEFAULT NULL,
8  screen_name varchar(128) DEFAULT NULL,
9  created_at timestamp NULL DEFAULT NULL,
10 followers int(11),
11 friends int(11),
12 text text,
13 PRIMARY KEY (id)
14 ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;
15
16 CREATE TABLE crypto (
17 id int(11) NOT NULL AUTO_INCREMENT,
18 name varchar(250) DEFAULT NULL,
19 rank int(11) DEFAULT NULL,
20 price_usd DOUBLE(16,2) DEFAULT NULL,
21 date_time timestamp NULL DEFAULT NULL,
22 PRIMARY KEY (id)
23 ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;
```

Validation of the table structures

```
25 •  describe tweets;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Cont

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
tweet_id	varchar(250)	YES		NULL	
name	varchar(128)	YES		NULL	
screen_name	varchar(128)	YES		NULL	
created_at	timestamp	YES		NULL	
followers	int(11)	YES		NULL	
friends	int(11)	YES		NULL	
text	text	YES		NULL	

```
25 •  describe crypto;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Co

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(250)	YES		NULL	
rank	int(11)	YES		NULL	
price_usd	double(16,2)	YES		NULL	
date_time	timestamp	YES		NULL	

Step 3: Data Gathering

Twitter Data:

Streaming tweets from twitter containing words such as #Bitcoin, #Ripple, #Ethereum etc. were collected and inserted into a MySQL database using below Python code

```
#Importing required libraries
from __future__ import print_function
import tweepy
import json
import MySQLdb
from dateutil import parser

#Setting the search words, the code will grab tweets that contain below hashtags
WORDS = ['#bitcoin','#XRP','ripple','#ethereum','#BitcoinCash','#EOS','#tether','#litecoin','#stellar','#TRON','#bitcoin SV']

#Defining twitter API keys and tokens
CONSUMER_KEY = "oO7yjGF0DOx4U4OlbeEy2xirm"
CONSUMER_SECRET = "AHp8XYwqwqvRyclgM0EaC6FXzQ9R0uN8L3RzmQUQgnvq6VMwjN"
ACCESS_TOKEN = "1064669303230861312-XM3VQPT6QycJEPJ1mZB0fDvKdmHoTB"
ACCESS_TOKEN_SECRET = "KC6SBkblOXJnulVAIvKwLmzurHXnJq9KW3LKlvYJMZLes"

#Defining database details hostname, user, password and database
HOST = "data.cep8edprhf64.us-east-1.rds.amazonaws.com"
USER = "dias_r"
PASSWD = "dIAs_777"
DATABASE = "data"
```

```
#Below function will select 'created_at', 'text', 'screen_name','tweet_id', 'followers' and 'friends'
#Store it in MySQL database
def store_data(created_at, text, name, screen_name, tweet_id, followers, friends):
    db=MySQLdb.connect(host=HOST, user=USER, passwd=PASSWD, db=DATABASE, charset="utf8")
    cursor = db.cursor()
    insert_query = "INSERT INTO tweets (tweet_id, name, screen_name, created_at, text, followers, friends) VALUES (%s, %s, %s, %s
    cursor.execute(insert_query, (tweet_id, name, screen_name, created_at, text, followers, friends))
    db.commit()
    cursor.close()
    db.close()
    return
```

```
#Class provided by the tweepy library to access Twitter Streaming API
class StreamListener(tweepy.StreamListener):

      def on_connect(self):
         # print message once connected to the API
         print("You are now connected to the streaming API.")

      def on_error(self, status_code):
         #In case an error occurs, display the error / status code
         print('An Error has occured: ' + repr(status_code))
         return False

      def on_data(self, data):
         #Connects to the database and stores the tweets
         try:
            # Decode the JSON from Twitter
             datajson = json.loads(data)

             #Extract data from the Tweet
             text = datajson['text']
             name = datajson['user']['name']
             screen_name = datajson['user']['screen_name']
             tweet_id = datajson['id']
             created_at = parser.parse(datajson['created_at'])
             followers = datajson['user']['followers_count']
             friends = datajson['user']['friends_count']

             #print out a message to the screen that we have collected a tweet
             print("Tweet collected at " + str(created_at))
```

```
            #print out a message to the screen that we have collected a tweet
            print("Tweet collected at " + str(created_at))

            #insert the data into the MySQL database
            store_data(created_at, text, name, screen_name, tweet_id, followers, friends)

        except Exception as e:
           print(e)

auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth.set_access_token(ACCESS_TOKEN, ACCESS_TOKEN_SECRET)

#Set up the listener. The 'wait_on_rate_limit=True' is needed to help with Twitter API rate limiting.
listener = StreamListener(api=tweepy.API(wait_on_rate_limit=True))
streamer = tweepy.Stream(auth=auth, listener=listener)
print("Tracking: " + str(WORDS))
streamer.filter(track=WORDS,languages=["en"]) #filter English tweets based on search words
```

The above Python script was executed for a couple of minutes and tweets were loaded into the “tweets” table as you can see from the below screenshot.

```
28 •  SELECT * FROM tweets;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	tweet_id	name	screen_name	created_at	followers	friends	text
1	1090080003616923648	Brad Mines	popthunder	2019-01-29 02:51:40	814	3382	RT @RonnieMoas: #bitcoin forecasts #crypto ...
2	1090080005114216448	Keyno neile	KeynoNeile	2019-01-29 02:51:40	90	261	RT @CommonEnemyInc: ✓ Instant Payments ...
3	1090080022348607488	Crypto Bravehearts	Master91Buil...	2019-01-29 02:51:44	1198	452	Target 1 hit
4	1090080025620041728	Crypto Catherine	FemmeForte11	2019-01-29 02:51:45	473	1199	RT @adoudble212: Excellenet and succint #Hol...
5	1090080042431008768	rafael anibal	Anibal81Ra	2019-01-29 02:51:49	35	0	RT @eth_dex: EthereumDEX AIRDROP + 2 ETH...
6	1090080047371902976	Thriller Crypto	ThrillerPodcast	2019-01-29 02:51:50	1232	3552	NYSE Arca Files Paperwork for Bitwise Bitcoin ETF
7	1090080063779786752	kardam	kardam2015	2019-01-29 02:51:54	19	24	RT @StellarOrg: Hello 2019! The January Mon...
8	1090080062672683008	Cheds Trading [Cancer Figh...	BigCheds	2019-01-29 02:51:54	70787	86	$BTC #Bitcoin 12 hour - Tweezer with light volu...
9	1090080077822545921	mari	mari81991382	2019-01-29 02:51:57	80	450	RT @digifinex: Ukraine to Legalize Cryptocurren...
10	1090080079303065600	Adriana Isabel	airm29	2019-01-29 02:51:58	96	341	RT @digifinex: Bitcoin may fall below $8,000 as ...

Please see attached script for twitter: twitter.ipynb

Coinmarketcap:

Coinmarketcap.com keeps track of the latest value of a cryptocurrency, I extracted the top 10 cryptocurrencies and noted time when they were collected. The code below collects data through the API and inserts the records into the “crypto” table created on the AWS MySQL database.

```
#import the required libraries
import urllib.request
import json
import requests
import pymysql
import datetime

#Connect to MySQL database
con=pymysql.connect(host='data.cep8edprhf64.us-east-1.rds.amazonaws.com', user='dias_r', passwd='dIAs_777',db='data');
cursor=con.cursor()
url='https://api.coinmarketcap.com/v1/ticker/?limit=10'; #API url

response=urllib.request.urlopen(url).read() #connecting to API url to get response

#create json object
json_obj=json.loads(response.decode('utf-8'))
print(json_obj)
now=datetime.datetime.now()
```

```
#create json object
json_obj=json.loads(response.decode('utf-8'))
print(json_obj)
now=datetime.datetime.now()

#Loop through json object and insert top 10 cryptocurrencies
for l in range(0,10):
    print("id",json_obj[l]['id'])
    print("name",json_obj[l]['name'])
    print("rank",json_obj[l]['rank'])
    print("price_usd",json_obj[l]['price_usd'])
    print(now);
    #insert data into MySQL database
    cursor.execute("INSERT INTO crypto (id, name, rank, price_usd, date_time) VALUES (%s,%s,%s,%s,%s)",
                   (json_obj[l]['id'],json_obj[l]['name'],json_obj[l]['rank'],json_obj[l]['price_usd'],now))

#Commit data
    con.commit()
con.close()
```

Please see attached script for coinmarketcap.com: coinmarket.ipynb

Step 4: Testing select queries on both the tables "tweets" and "crypto"

The tweets table contains 8 columns the first column "id" is the primary key which is generated while streaming tweets and loading them into the database.

Selecting the tweets which contain "TRON" the text column contains the tweet message using the below query we can filter out tweets based on certain keywords. Only 3 people tweeted about the TRON cryptocurrency

```
1  SELECT * FROM tweets where text like '%TRON%';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	tweet_id	name	screen_name	created_at	followers	friends	text
22	1090080231141068800	"RB3" - Digital Asset Addict	RobertBagley	2019-01-29 02:52:34	846	2014	RT @Trontonium:
26	1090080277240504320	Jignesh Beladiya	jigneshon	2019-01-29 02:52:45	1287	2078	RT @justinsuntron: I would like to explore the p...
47	1090080455918043136	TRX MEXICO	uriostegui1519	2019-01-29 02:53:28	208	315	RT @TronColony: This in itself will be a huge ac...

From the data collected there are only 3 users which have more than 10000 followers

```
2  SELECT * FROM tweets where followers>=10000;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	tweet_id	name	screen_name	created_at	followers	friends	text
8	1090080062672683008	Cheds Trading [Cancer Figh...	BigCheds	2019-01-29 02:51:54	70787	86	$BTC #Bitcoin 12 hour - Tweezer with light volu...
16	1090080145052962816	Groovy Hooman Tokens/ BA...	groovyhooman	2019-01-29 02:52:13	42400	2337	Buy GHT token here: https://t.co/uoWP8fzKD8 ...
33	1090080337332502528	Obi	aaaamhim	2019-01-29 02:52:59	17207	15852	$NSPX 006?! #BIOTECH STOCK CANCER PATE...

Only one user has more than 10000 friends

```
3  SELECT * FROM tweets where friends>=10000;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	tweet_id	name	screen_name	created_at	followers	friends	text
33	1090080337332502528	Obi	aaaamhim	2019-01-29 02:52:59	17207	15852	$NSPX 006?! #BIOTECH STOCK CANCER PATE...

Below is a list of users that tweeted more than once from the data gathered

```
6  select name, count(*) FROM tweets group by name having count(*)>1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	count(*)
	4
Adriana Isabel	2
Alejandra♣ »♥Dj17♥«	3
Alejandro Rodriguez	2
Anyerson1575	2
Block Watcher	4
boshra ali	2
Brian	2

User Remi Vee has the greatest number of friends and followers

```
8 • select * FROM tweets group by name order by followers,friends desc;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	tweet_id	name	screen_name	created_at	followers	friends	text
193	1090416413473411072	Remi Vee	Remi_Vladuc...	2019-01-30 01:08:26	160122	138215	Naval: Killer App of Crypto is Socialism, Venezue...

Below is the list of the top 10 cryptocurrencies as you can see Bitcoin has the highest price followed Bitcoin Cash and Ethereum. The rank column indicates the popularity of the cryptocurrency.

```
8 • select * from crypto order by price_usd desc;
```

Result Grid | Filter Rows: | Edit: | Ex

id	name	rank	price_usd	date_time
1	Bitcoin	1	3462.84	2019-01-28 21:50:16
6	Bitcoin Cash	6	111.11	2019-01-28 21:50:16
3	Ethereum	3	105.78	2019-01-28 21:50:16
10	Bitcoin SV	10	64.18	2019-01-28 21:50:16
7	Litecoin	7	31.07	2019-01-28 21:50:16
5	EOS	5	2.25	2019-01-28 21:50:16
4	Tether	4	1.01	2019-01-28 21:50:16
2	XRP	2	0.29	2019-01-28 21:50:16
9	Stellar	9	0.09	2019-01-28 21:50:16
8	TRON	8	0.03	2019-01-28 21:50:16
NULL	NULL	NULL	NULL	NULL

Step 5: Basic functions (MIN, MAX, AVG)

Finding the maximum, minimum and average number of followers

```
8 • SELECT max(followers),min(followers),avg(followers) FROM tweets;
9 • SELECT max(friends),min(friends),avg(friends) FROM tweets;
10
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

max(followers)	min(followers)	avg(followers)
160122	0	4850.6126

Finding the maximum, minimum and average number of friends

```
9 • SELECT max(friends),min(friends),avg(friends) FROM tweets;
10
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

max(friends)	min(friends)	avg(friends)
138215	0	2197.8018

Step 6: JOINS (INNER, LEFT, RIGHT, FULL)

In order to link the "tweets" and "crypto" table I created a new table "tweets2" which indicates the crypto currency that was tweeted

```
47  CREATE TABLE tweets2 (
48  id int(11) NOT NULL AUTO_INCREMENT,
49  tweet_id varchar(250) DEFAULT NULL,
50  name varchar(128) DEFAULT NULL,
51  screen_name varchar(128) DEFAULT NULL,
52  created_at timestamp NULL DEFAULT NULL,
53  followers int(11),
54  friends int(11),
55  text text,
56  currency varchar(128) DEFAULT NULL,
57  PRIMARY KEY (id)
58  ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;
59
60  describe tweets2;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
tweet_id	varchar(250)	YES		NULL	
name	varchar(128)	YES		NULL	
screen_name	varchar(128)	YES		NULL	
created_at	timestamp	YES		NULL	
followers	int(11)	YES		NULL	
friends	int(11)	YES		NULL	
text	text	YES		NULL	
currency	varchar(128)	YES		NULL	

Now the currency column will contain the currency depending on the currency tweeted. Below query was executed to identify and insert the currency based on tweets.

```
62  insert into tweets2
63  select id,tweet_id,name,screen_name,created_at,followers,friends,text,
64  case
65   when text like '%bitcoin%' THEN "Bitcoin"
66   when text like '%ripple%' THEN "Ripple"
67   when text like '%XRP%' THEN "Ripple"
68   when text like '%ethereum%' THEN "Ethereum"
69   when text like '%bitcoin cash%' THEN "Bitcoin Cash"
70   when text like '%EOS%' THEN "EOS"
71   when text like '%tether%' THEN "Tether"
72   when text like '%THETA%' THEN "Tether"
73   when text like '%litecoin%' THEN "Litecoin"
74   when text like '%stellar%' THEN "Stellar"
75   when text like '%TRON%' THEN "TRON"
76   when text like '%Bitcoin SV%' THEN "Bitcoin SV"
77   else NULL
78   END AS currency
79  FROM tweets;
```

sult Grid | Filter Rows: | Export: | Wrap Cell Content:

id	tweet_id	name	screen_name	created_at	follower:	friends	text	currency
1	1090080...	Brad Mines	popthunder	2019-01-29 ...	814	3382	RT @RonnieMoas: #bitcoin forecasts #crypto ...	Bitcoin
2	1090080...	Keyno n...	KeynoNeile	2019-01-29 ...	90	261	RT @CommonEnemyInc: ✓ Instant Payments ...	NULL
3	1090080...	Crypto ...	Master91Buil...	2019-01-29 ...	1198	452	Target 1 hit	NULL
4	1090080...	Crypto ...	FemmeForte11	2019-01-29 ...	473	1199	RT @adoudble212: Excellenet and succint #Hol...	Ripple
5	1090080...	rafael a...	Anibal81Ra	2019-01-29 ...	35	0	RT @eth_dex: EthereumDEX AIRDROP + 2 ETH...	Ethereum
6	1090080...	Thriller ...	ThrillerPodcast	2019-01-29 ...	1232	3552	NYSE Arca Files Paperwork for Bitwise Bitcoin ETF	Bitcoin
7	1090080...	kardam	kardam2015	2019-01-29 ...	19	24	RT @StellarOrg: Hello 2019! The January Mon...	Stellar
8	1090080...	Cheds T...	BigCheds	2019-01-29 ...	70787	86	$BTC #Bitcoin 12 hour - Tweezer with light volu...	Bitcoin

Inner Join: Inner join tweets which contain same currencies in the crypto table. All the matching rows between "tweets2" and "crypto" are returned in other words if the currency column of "tweets2" matches the name of the "crypto" table those tweets are returned.

```
83  #INNER JOIN
84  select text,currency,b.name,b.price_usd,created_at,date_time
85  FROM tweets2 a inner join crypto b on a.currency=b.name;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

text	currency	name	price_usd	created_at	date_time
RT @RonnieMoas: #bitcoin forecasts #crypto #cryptocurrency ...	Bitcoin	Bitcoin	3462.84	2019-01-29 02:51:40	2019-01-28 21:50:16
RT @eth_dex: EthereumDEX AIRDROP + 2 ETH GIVEAWAYS Et...	Ethereum	Ethereum	105.78	2019-01-29 02:51:49	2019-01-28 21:50:16
NYSE Arca Files Paperwork for Bitwise Bitcoin ETF	Bitcoin	Bitcoin	3462.84	2019-01-29 02:51:50	2019-01-28 21:50:16
RT @StellarOrg: Hello 2019! The January Monthly Roundup Ne...	Stellar	Stellar	0.09	2019-01-29 02:51:54	2019-01-28 21:50:16
$BTC #Bitcoin 12 hour - Tweezer with light volume - 1 hour BB pi...	Bitcoin	Bitcoin	3462.84	2019-01-29 02:51:54	2019-01-28 21:50:16
RT @digifinex: Ukraine to Legalize Cryptocurrencies by law #Ukr...	Bitcoin	Bitcoin	3462.84	2019-01-29 02:51:57	2019-01-28 21:50:16
RT @digifinex: Bitcoin may fall below $8,000 as crypto confab fa...	Bitcoin	Bitcoin	3462.84	2019-01-29 02:51:58	2019-01-28 21:50:16
The Hardware Bitcoin Wallet. Get Trezor now for only 89 EUR ht...	Bitcoin	Bitcoin	3462.84	2019-01-29 02:52:02	2019-01-28 21:50:16
RT @criscyborg: Awesome meeting @johnkim77 today! Who o...	Litecoin	Litecoin	31.07	2019-01-29 02:52:06	2019-01-28 21:50:16

Left Join: To demonstrate a left join I pulled tweets related to currencies such as Binance and Cardano. These currencies are not present in the crypto table since they are not in the top 10. The query will return all the rows present in the left table "tweets2" even if they are not present in the "crypto" table. As we can see from the below screenshot for Cardano and Binance the text column and currency columns are returned from the "tweets2" table while all the columns of the crypto table are populated with Null values.

```
91  #LEFT JOIN
92  select text,currency,b.name,b.price_usd,created_at,date_time
93  FROM tweets2 a LEFT join crypto b on a.currency=b.name;
94
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

text	currency	name	price_usd	created_at	date_time
RT @PLAY_GOC: A work in progress... Player unlockables, team...	NULL	NULL	NULL	2019-01-30 02:17:55	NULL
Give the Soba Home detox kit they need it now https://t.co/4IN...	NULL	NULL	NULL	2019-01-30 02:18:01	NULL
RT @CHalexov2016: 7h and 7min left for voting for $xtz #tezos...	Binance	NULL	NULL	2019-01-30 02:18:01	NULL
If you have #marketing skills, https://t.co/e09nHXDkuF is lookin...	NULL	NULL	NULL	2019-01-30 02:18:05	NULL
Binance RSI + STOCH 15 min strategy. Currently RSI and STOC...	Binance	NULL	NULL	2019-01-30 02:18:57	NULL
RT @IOHK_Charles: Fake Stake got nothing on Cardano https://...	Cardano	NULL	NULL	2019-01-30 02:19:06	NULL
RT @CHalexov2016: 7h and 7min left for voting for $xtz #tezos...	Binance	NULL	NULL	2019-01-30 02:19:11	NULL
RT @CHalexov2016: 7h and 7min left for voting for $xtz #tezos...	Binance	NULL	NULL	2019-01-30 02:19:35	NULL

Right Join: Like the left join query the result of this query returned all the records from the right table "crypto" and only records that matched from the left table "tweets2". We can see that none of the tweets contained Tether, Bitcoin Cash and Bitcoin SV hence those values were populated with Null for the left table.

```
95  #RIGHT JOIN
96  select text,currency,b.name,b.price_usd,created_at,date_time
97  FROM tweets2 a RIGHT join crypto b on a.currency=b.name;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

text	currency	name	price_usd	created_at	date_time
RT @DiepSanh: David E Rutter, CEO of R3 is also at Paris Fintec...	XRP	XRP	0.29	2019-01-30 02:17:44	2019-01-28 21:50:16
RT @TecStats: Developer Mark Friedenbach just announce way ...	Bitcoin	Bitcoin	3462.84	2019-01-30 02:17:44	2019-01-28 21:50:16
RT @BitTorrent: Don't miss out on Jordy Berson's keynote speec...	TRON	TRON	0.03	2019-01-30 02:17:51	2019-01-28 21:50:16
RT @Btcexpertindia: Good news from Iran #IRAN LIFTS BITCOI...	Bitcoin	Bitcoin	3462.84	2019-01-30 02:17:52	2019-01-28 21:50:16
RT @mishalederman: Facts: #TRON ICO raised $58,097,999 #...	TRON	TRON	0.03	2019-01-30 02:17:57	2019-01-28 21:50:16
RT @Gamecom_Korea: Play #CryptoSpin on #Tron everyday!! ...	TRON	TRON	0.03	2019-01-30 02:17:58	2019-01-28 21:50:16
RT @utorrent: Don't miss out on Jordy Berson's keynote speech ...	TRON	TRON	0.03	2019-01-30 02:17:59	2019-01-28 21:50:16
RT @TronWeekly: New York DFS allows buying #Bitcoin with deb...	Bitcoin	Bitcoin	3462.84	2019-01-30 02:17:59	2019-01-28 21:50:16
Ripple Price Analysis: XRP's Previous Support Now Resistance ht...	Bitcoin	Bitcoin	3462.84	2019-01-30 02:18:01	2019-01-28 21:50:16
Major Exchanges Join Forces to Combat Money Laundering http...	Bitcoin	Bitcoin	3462.84	2019-01-30 02:18:02	2019-01-28 21:50:16
RT @_alterdice: EOS Price Analysis: A Break Out Imminent in IO...	EOS	EOS	2.25	2019-01-30 02:19:04	2019-01-28 21:50:16
@ChicoCrypto @binance @Tronfoundation @BitTorrent @crypt...	TRON	TRON	0.03	2019-01-30 02:19:25	2019-01-28 21:50:16
Ethereum $ETH price: $107.12 We checked! Binance registratio...	Ethereum	Ethereum	105.78	2019-01-30 02:19:27	2019-01-28 21:50:16
We now accept @ethereum also for travel bookings in over 200 ...	Ethereum	Ethereum	105.78	2019-01-30 02:20:04	2019-01-28 21:50:16
NULL	NULL	Tether	1.01	NULL	2019-01-28 21:50:16
NULL	NULL	Bitcoin Cash	111.11	NULL	2019-01-28 21:50:16
NULL	NULL	Bitcoin SV	64.18	NULL	2019-01-28 21:50:16

Conclusion: To summarize, I was able to extract tweets which contained keywords related to cryptocurrencies and load them into a MySQL table "tweets". I also extracted data from coinmarketcap.com which publishes the latest value of a cryptocurrency and inserted the top 10 cryptocurrencies into the table "crypto". Since it was difficult to link the cryptocurrencies with the tweets because neither of the table have much in common other than the keywords, I created a new table "tweets2". Tweets2 identifies tweets based on the cryptocurrency in the tweet, I then joined the tables based on the cryptocurrency tweeted.