

# Predict Crypto Database Quick Start Guide

*Ricky Esclapon - [riccardo.esclapon@colorado.edu](mailto:riccardo.esclapon@colorado.edu)*

*2020-01-01*



# Contents

<b>1</b>	<b>Overview</b>	<b>5</b>
<b>2</b>	<b>Interacting with the DB</b>	<b>7</b>
2.1	Benefits . . . . .	7
2.2	“Metabase” explained . . . . .	7
2.3	Logging In . . . . .	8
<b>3</b>	<b>Useful Tables</b>	<b>11</b>
3.1	Browse Data . . . . .	11
3.2	Useful tables . . . . .	14
<b>4</b>	<b>Usage Guide</b>	<b>17</b>
4.1	Pulling Most Recent Data . . . . .	17
4.2	Data Exploration . . . . .	17
4.3	Answer Questions . . . . .	25
4.4	Creating dashboards . . . . .	25
<b>5</b>	<b>Documentation Usage</b>	<b>27</b>
5.1	pkDummy and pkey . . . . .	27
5.2	Navigating the Metabase interface . . . . .	27
5.3	Using x-rays to explore a table . . . . .	27
<b>6</b>	<b>Additional Tips</b>	<b>29</b>
6.1	Creating Variables for queries . . . . .	29
6.2	Embedding anything created on Metabase . . . . .	29
6.3	Creating e-mail triggers . . . . .	29
6.4	Using the Shrimpy-Python Library . . . . .	29



# Chapter 1

## Overview



This is a quick start guide for the Predict Crypto DataBase which should provide the support you need to interact with the database and pull data. Everything you need to know will be outlined in this document and you can use the sidebar on the left to review the following sections:

1. Overview- This section.
2. Interacting with the DB- Instructions around accessing the *Metabase* environment that will allow you to interact with the database, which is hosted on the website PredictCryptoDB.com
3. Useful Tables- A review of some of the more useful/interesting tables you can find within the database and an overview of the best places to get started.
4. Usage Guide- This guide explores some of the functionality found within the website/Metabase environment.
5. Documentation Usage- An overview of the documentation that is available through the Metabase environment and how to use it to answer questions you may have around where the data is sourced from as well as a complete data dictionary for every table.
6. Additional Tips- Some additional notes around using the environment to its full potential. This section goes over things like pulling the most recent data, creating dashboards, embedding anything you create as an iframe within a website or blog post, creating e-mail triggers and using a Python package to execute trades.



## Chapter 2

# Interacting with the DB

First off, why the heck is there a website? If there's a database why wouldn't you just access it through a SQL editor like SSMS or MySQL? Let's start with the benefits of having this environment readily available to access through a website before jumping into how to start interacting with the database.

### 2.1 Benefits

- Giving access to the database can be challenging, and having a centralized server that can be accessed through a website allows us to analyze the data without worrying about individual IP addresses being allowed access and other issues we would run into relating to authentication without this type of environment to work out of.
- There is no setup required on your part to start working with the database and as soon as your account is setup you can start writing SQL queries through the website and log into the website at any time.
- You can access the Metabase interface to interact with the database from any device, like a tablet or smartphone.
- It works as a shared environment where we can all collaborate and see each other's dashboards, sql queries, visualizations etc... This is similar to the way a Tableau Online website would work within an organization.
- Makes creating dashboards to visualize the data extremely simple. Those dashboards can then be shared and be embedded, and the data can be refreshed on a schedule in a matter of clicks.

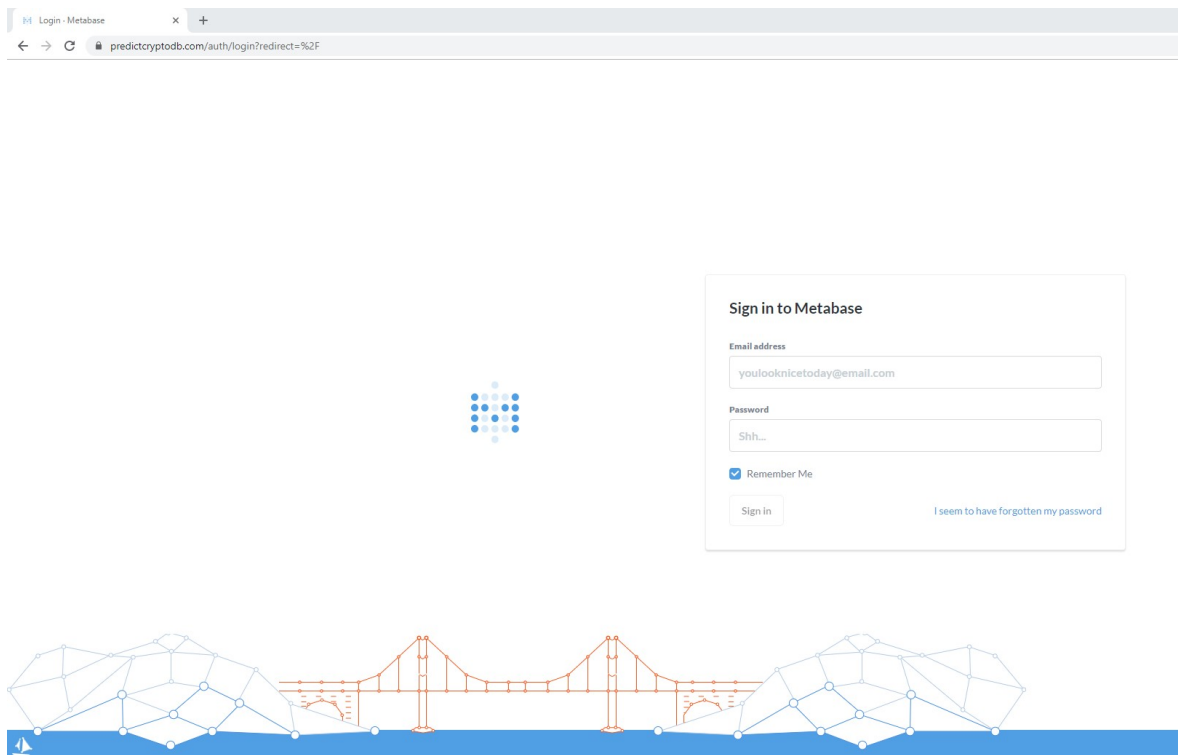
### 2.2 “Metabase” explained

First, let me clarify on what I mean by *Metabase* and how this all works.

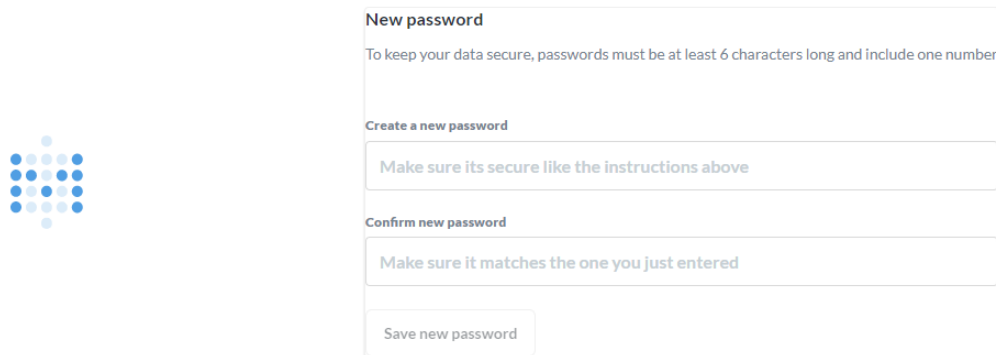
1. Metabase is the name of the open source software that we are using to interact with the database.
2. Using the docker container available to download on Metabase's website, I stood up a server-type environment in AWS that hosts the Metabase instance that is connected to the database.
3. The AWS environment running the Metabase instance can be connected to by opening your web browser and navigating to the https secured website [predictcryptodb.com](https://predictcryptodb.com). From here, you will use the set of credentials that you created after clicking on the invitation from the e-mail you received. If you need an invite to create a set of credentials, e-mail me at [riccardo.esclapon@colorado.edu](mailto:riccardo.esclapon@colorado.edu).

## 2.3 Logging In

1. To get started, the first thing you will want to do every time is go to the URL `predictcryptodb.com`. You should see a login page:



- If you are having issues accessing the website, try spelling the url including ***https://***`predictcryptodb.com`.
- Before you can login, you need to configure your account through the e-mail you should have received. If you did not receive the e-mail, check your spam folder first:
  - Configure your password and you will be able to use your e-mail and password to login in the future:



2. Once you have logged in, you should see a page that looks like this:



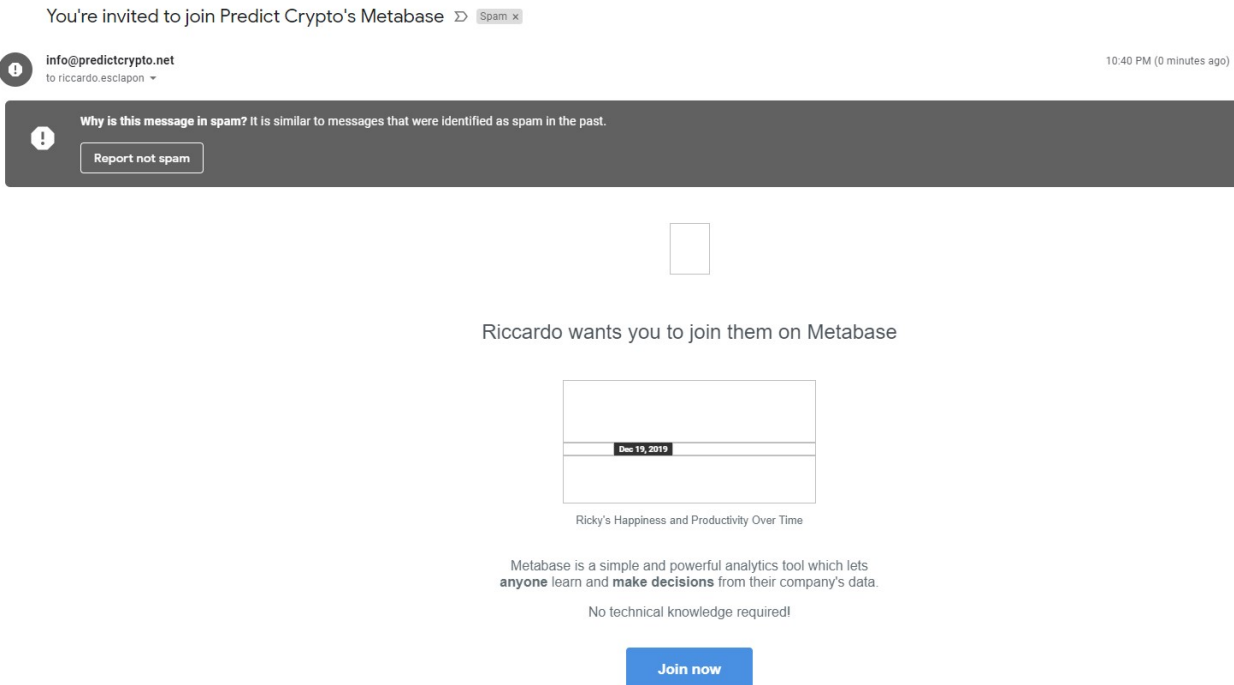
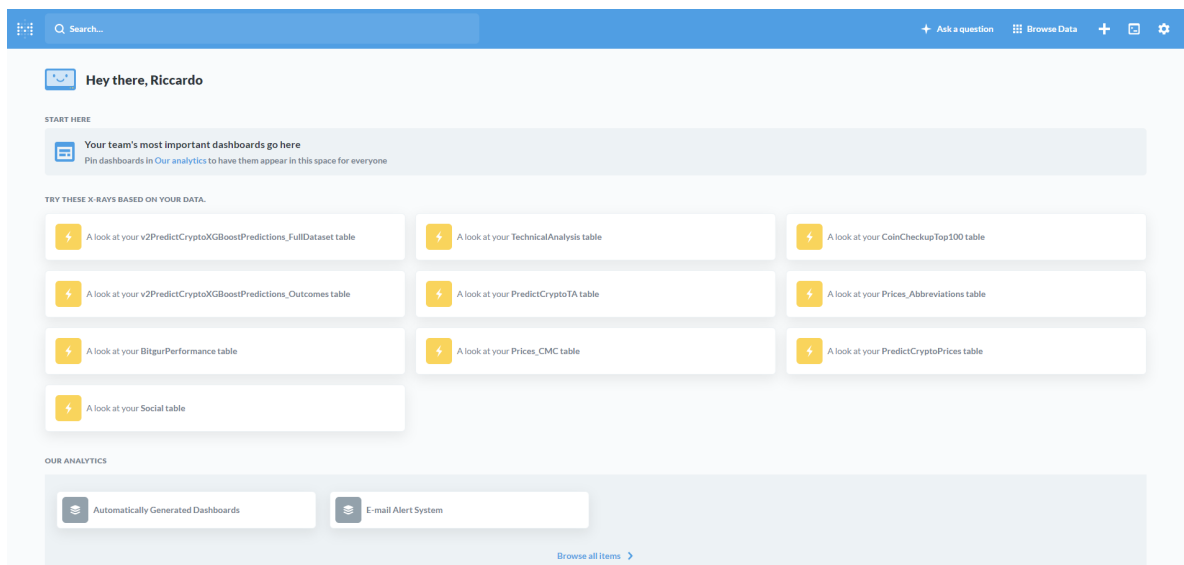


Figure 2.1: *If you need the e-mail to setup your login, e-mail me*



In the next section we will talk about some good tables to get started on, and getting more comfortable navigating the layout of the website.

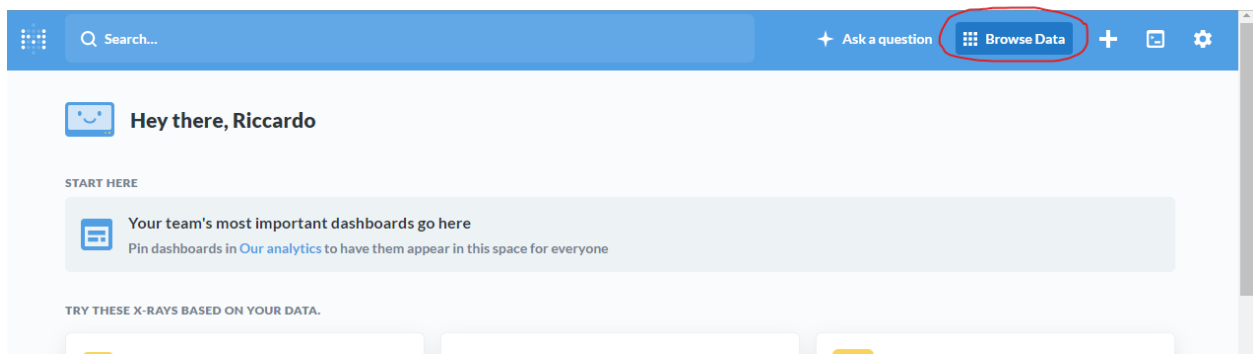


## Chapter 3

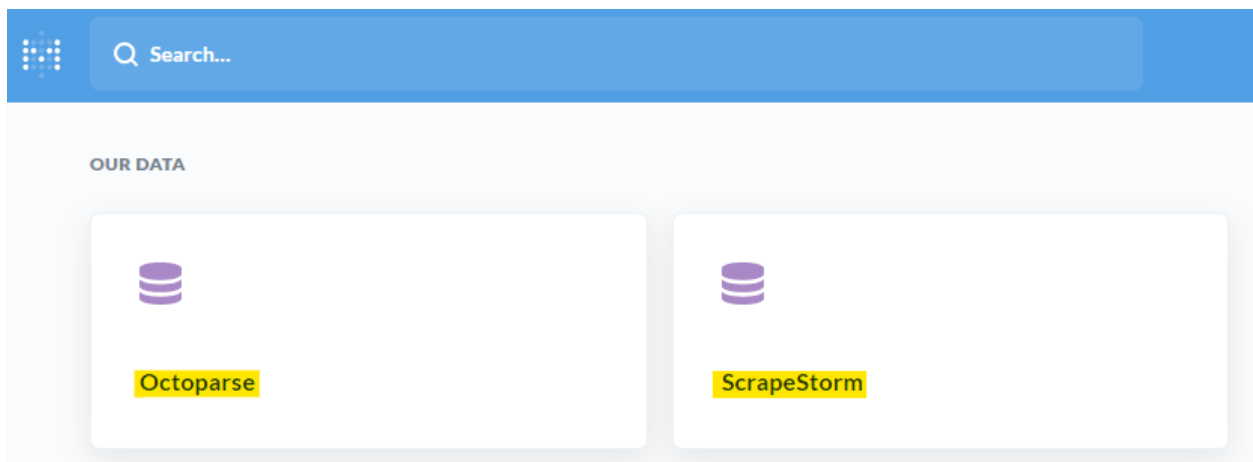
# Useful Tables

### 3.1 Browse Data

Let's start by clicking on the **Browse Data** tab in the top right of the Metabase environment:



Here, you will see two different options:



- **Octoparse** is the schema that is associated with data I have collected by using the *Octoparse* web scraping software.
- Conversely, **ScrapeStorm** is the schema associated with data collected using the *ScrapeStorm* web

scraping software.

- Web scraping has its challenges in terms of stability, so I built some additional resilience by using two different tools that work independently of each other and do similar things (and in some cases collect the same data). Although not a perfect solution, having both up and running means we can usually fill the gaps that might arise in each tool respectively.
  - The web scrapers that are run through Octoparse run on their servers in the cloud, which are very stable but have had some issues here and there in the past.
  - The web scrapers that are run through ScrapeStorm run on a local machine on my end. I have a computer in the cloud that runs 24/7 but even after upgrading the hardware on it ScrapeStorm kept crashing, so this currently runs on my powerful desktop computer that is always on at home. When these run, the data flows in almost immediately, meaning if it's 4:05PM and you pulled the latest hour of data (which would be equivalent to the `max(**pkDummy**)`, more to come on that later on)
- Here is the size in MB of both the Octoparse database and the ScrapeStorm db as of the last time this

Show 10 entries Search:

	DB name	Size in MB	Today
1	Octoparse	34274.21875	2020-01-02
2	ScrapeStorm	828.625	2020-01-02

Showing 1 to 2 of 2 entries Previous 1 Next

document was refreshed (updated daily):

**Back in MetaBase, let's click on the option that says Octoparse:**

Now you should see the tables that are contained within the Octoparse schema. By hovering over each table, you will see three options appear, which will be better explained in the next section about *Documentation Usage*. In the screenshot the mouse is hovering over the *i* symbol for the Bitgur table:

OUR DATA > OCTOPARSE Learn about our data

Bitgur	BitgurPerformance	CoinCheckupTop100
CoinToBuyCountries	concatApp	concatAppFour
joinTarget	joinTargetFour	Last7Days_Prices
mysql_Bitgur	NewPrices	NewPricesFour
newsCheck	PredictCryptoPrices	PredictCryptoSocial

Data is collected hourly from https://bitgur.com/perf. The first available data for this table is 2018-11-11. Collects data up to around rank 1,500.

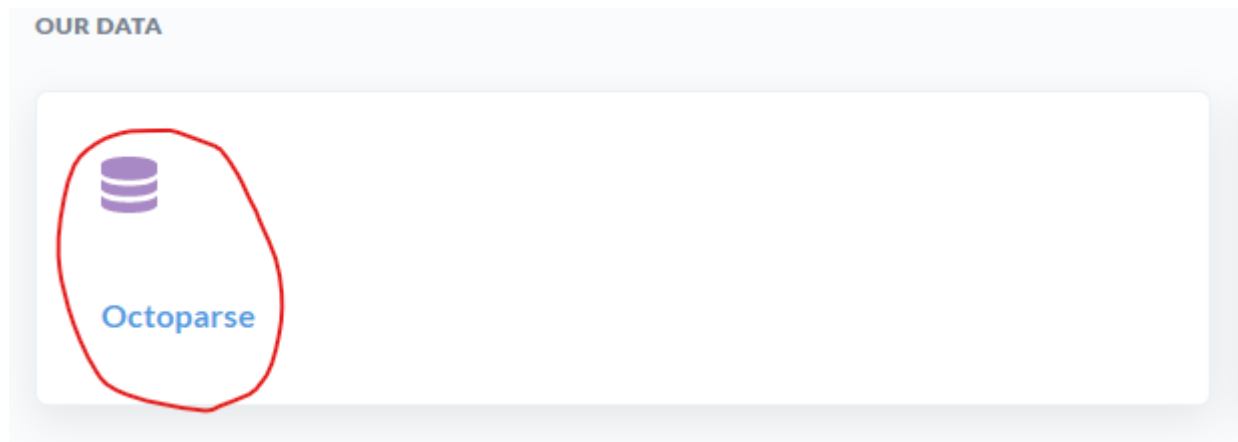
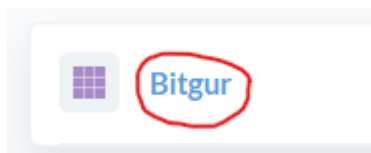


Figure 3.1: I would recommend starting here because this was the first/original database and will have more historical data compared to ScrapeStorm, which I got up and running much later

For now, let's go ahead and click on the name of the table **Bitgur**:



After clicking on the table name, you should see some example data show up. This shows the first 2,000 rows of data found in the table:

Octoparse • Bitgur

Save Filter + Summarize

key	Name	Symbol	BITEX	Date	DateTime	Index	pkDummy	Price (\$)	Rank	Volatility
2018-11-12 03ETH	Ethereum	ETH	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	209.13	2	17.36
2018-11-12 03XRP	XRP	XRP	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	0.5	3	23.09
2018-11-12 03BCH	Bitcoin Cash	BCH	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	324.01	4	66.11
2018-11-12 03XLM	Stellar	XLM	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	0.28	5	62.98
2018-11-12 03EOS	EOS	EOS	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	5.32	6	22.96
2018-11-12 03LTC	Litecoin	LTC	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	50.70	7	40.87
2018-11-12 03ADA	Cardano	ADA	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	0.076	8	23.02
2018-11-12 03XMR	Monero	XMR	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	103.70	9	31.6
2018-11-12 03DASH	DASH	DASH	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	159.40	12	28.24
2018-11-12 03NEO	NEO	NEO	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	15.63	15	36.43
2018-11-12 03XEM	NEM	XEM	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	0.092	17	21.58
2018-11-12 03BTG	Bitcoin Gold	BTG	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	29.66	23	190.97
2018-11-12 03OMG	OmiseGo	OMG	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	3.18	24	47.27
2018-11-12 03DOGE	Dogecoin	DOGE	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	0.0032	26	42.01
2018-11-12 03ZNL	Zilliqa	ZIL	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	0.034	31	33.15
2018-11-12 03AE	Aeternity	AE	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	1.14	32	23.74
2018-11-12 03BTS	Bitshares	BTS	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	0.094	35	22.8
2018-11-12 03ICX	ICON	ICX	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	0.58	39	49.1
2018-11-12 03DGB	DigiByte	DGB	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	0.02	40	45.84
2018-11-12 03GNT	Golem	GNT	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	0.16	51	31.83
2018-11-12 03SNT	Status	SNT	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	0.036	59	23.91
2018-11-12 03KCS	KuCoin Sh...	KCS	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	1.11	65	18.45
2018-11-12 03IOST	Internet of...	IOST	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	0.012	69	26.44
2018-11-12 03ELF	elf	ELF	56.02	November 12, 2018, 12:00 AM	November 12, 2018, 3:02 A...	21.34	2018-11-12-03	0.31	76	27.95

Visualization Settings

Showing first 2,000 rows

In the next section we will walk through some of the functionality associated with the things circled in red in the screenshot above using the Bitgur table as an example.

## 3.2 Useful tables

Symbol <chr>	Price <dbl>	
BTC	7257.300000	
ETH	132.870000	
XRP	0.194140	

For the previews below, keep an eye out for a button to show more columns:

### 3.2.1 Tables in Octoparse db

#### 3.2.1.1 Bitgur

- Source: <https://bitgur.com/>
- Data collected since: 2018-11-11

#### 3.2.1.2 BitgurPerformance

- This table is very similar to the `Bitgur` table, but this one includes some fields that summarize the short-term performance of the cryptocurrency in the short-term relative to when the data was collected.
- The normal `Bitgur` table would have the prices fields, but not information about the performance over the previous 1h, 3h, etc..
- Source: <https://bitgur.com/performance>
- Data collected since: 2019-11-05

#### 3.2.1.3 ADD MORE TABLES

### 3.2.2 Tables in ScrapeStorm db

#### 3.2.2.1 ScrapeStorm.ShrimpyPrices

When using data from a table like the `BitgurPerformance`, the data is aggregated at a global level. Meaning, the Bitgur website connects to a certain number of exchanges, and calculates things like the overall Market Capitalization (\$), the 24h volume, etc. When it comes to the price, the website will average things across exchanges to get to the current price, which makes sense, but in my experience this is not as precise as it needs to be in order to programmatically trade on the cryptocurrency markets.

+ Therefore, a better approach is to collect the prices directly through the individual exchanges. Becau

- Source: <https://www.shrimpy.io/exchange>

#### 3.2.2.2 ScrapesStorm.ShrimpyPricesBTC

This table is exactly like the previous one `ShrimpyPrices`, but this one shows prices in BTC instead of being in USD

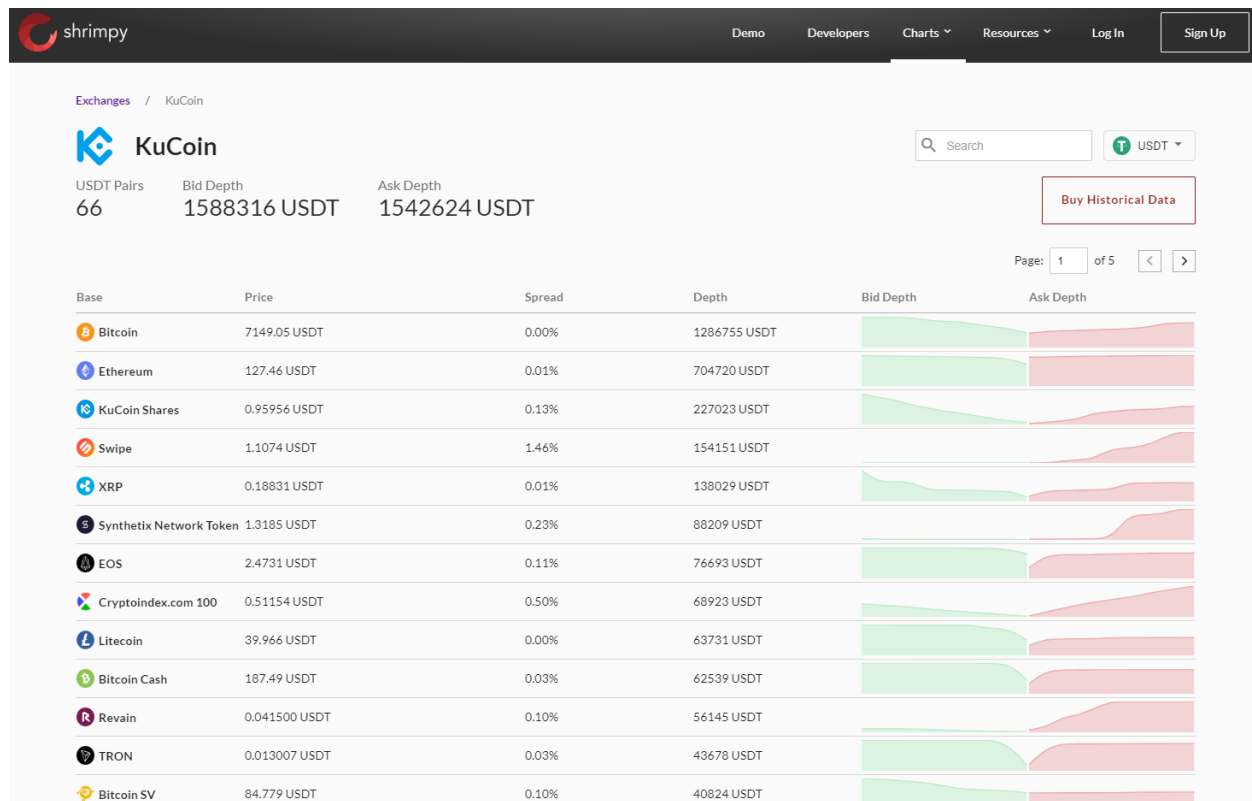


Figure 3.2: Example for the KuCoin exchange

### 3.2.3 Number of rows by table

Database	Table Name	Rows
ScrapeStorm	ShrimpyPricesBTC	570220
ScrapeStorm	Messari-2019-09-26	829532
ScrapeStorm	BitgurBasic	450889
ScrapeStorm	ShrimpyPrices	299491
ScrapeStorm	KuCoinPrices	38820
ScrapeStorm	Symbols	817056
Octoparse	v2PredictCryptoXGBoostPredictions_Outcomes	14135169
Octoparse	BitgurPerformance	3030001
Octoparse	Bitgur	23138495
Octoparse	TechnicalAnalysis	2191923
Octoparse	CoinCheckupTop100	681092
Octoparse	AYX191231dabf06dfe9758fb1a5477	17764
Octoparse	v2PredictCryptoXGBoostPredictions	54262
Octoparse	v2PredictCryptoXGBoostPredictions_FullDataset	54414





## Chapter 4

# Usage Guide

### 4.1 Pulling Most Recent Data

Let's say you went ahead and trained a predictive model using the data found in the database and now you want to test making some *semi-live* predictions, how would you go about pulling the latest data to make predictions on?

- pkDummy
- pkey

Make a note that can't do this between different schemas because the Octoparse one is in UTC and the ScrapeStorm one in local time. Only real effective way I found do to that is through Alteryx tools.

### 4.2 Data Exploration

Let's explore some of the functionality found in Metabase using the **Bitgur** table results page we got to at the end of the previous section.

- The first thing I want to point out, is that you can click on each field to open a menu with an assortment of options based on the data type. For example, if we click on the **Price** field, these are the options we are given:

Index	pkDummy	Price (\$)	Rank	Volatility
21.34	2018-11-12 03			
21.34	2018-11-12 03			
21.34	2018-11-12 03			
21.34	2018-11-12 03			
21.34	2018-11-12 03			
21.34	2018-11-12 03			
21.34	2018-11-12 03			
21.34	2018-11-12 03	103.70	9	31.6
21.34	2018-11-12 03	150.10	10	32.34

- The Date field is a different data types and offers different options. For example we could sort the data by the **Date** field **Descending**:

BITEX	Date	DateTime
56.02		er 12, 2018, 3:02 A...
56.02		er 12, 2018, 3:02 A...
56.02		er 12, 2018, 3:02 A...
56.02		er 12, 2018, 3:02 A...
56.02		er 12, 2018, 3:02 A...
56.02		er 12, 2018, 3:02 A...
56.02		er 12, 2018, 3:02 A...
56.02	November 12, 2018 12:00 AM	November 12, 2018 3:02 A...

- \* Which would return the first 2,000 rows again but this time sorted showing the largest values first (meaning the latest date):

- Next, let's click on the purple **Filter** button in the top right of the page:

## Octoparse • Bitgur

pkey	Name	Symbol	BITEX	Date	DateTime	Index	pkDummy	Price (\$)	Rank	Volatility
2019-12-20 00BTC	Bitcoin	BTC	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	7,087.80	1	57.7
2019-12-20 00ETH	Ethereum	ETH	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	126.10	2	46.5
2019-12-20 00XRP	XRP	XRP	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	0.19	3	49.49
2019-12-20 00USDT	Tether	USDT	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	1	4	20.53
2019-12-20 00BCH	Bitcoin Cash	BCH	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	185.45	5	50.34
2019-12-20 00LTC	Litecoin	LTC	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	39.48	6	53

Figure 4.1: If the Bitgur table is not being very responsive, try using the BitgurPerformance table instead, which should load faster (but has less historical data, this table is described in a section below)

Octoparse • Bitgur

Save

Filter

+ Summarize

pkey	Name	Symbol	BITEX	Date	DateTime	Index	pkDummy	Price(\$)	Rank	Volatility
2019-12-20 00BTC	Bitcoin	BTC	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	7,087.80	1	57.7
2019-12-20 00ETH	Ethereum	ETH	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	126.10	2	46.5
2019-12-20 00XRP	XRP	XRP	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	0.19	3	49.49
2019-12-20 00USDT	Tether	USDT	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	1	4	20.53
2019-12-20 00BCH	Bitcoin Cash	BCH	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	185.45	5	50.34
2019-12-20 00LTC	Litecoin	LTC	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	39.48	6	53

– Now you should see the following sidebar pop-up:

\* Let's do a quick example filtering based on the **DateTime** field:

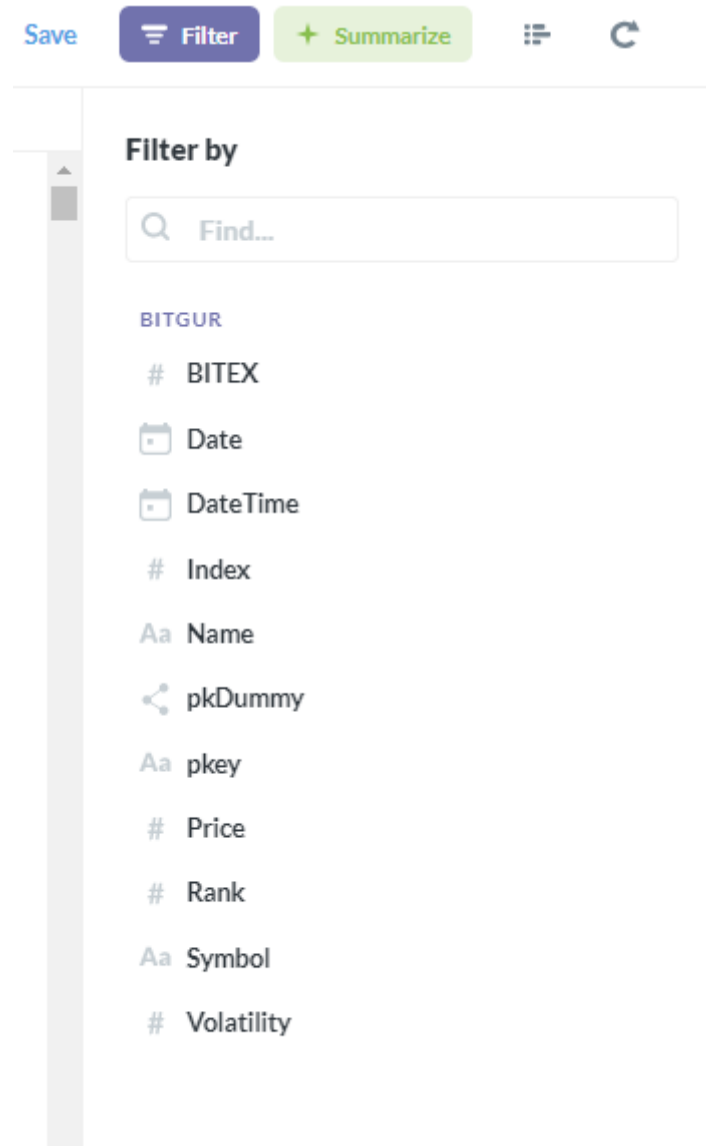


Figure 4.2: Notice how the icons next to the field names correspond to their data types

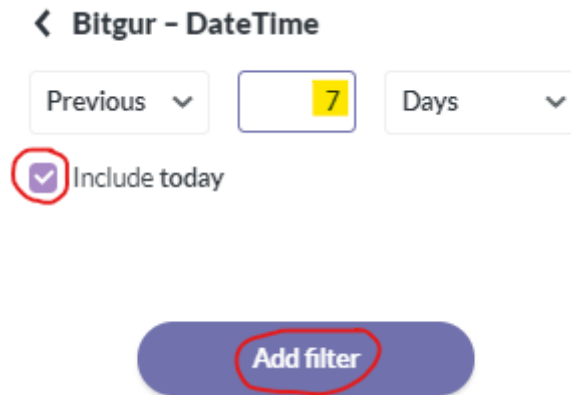
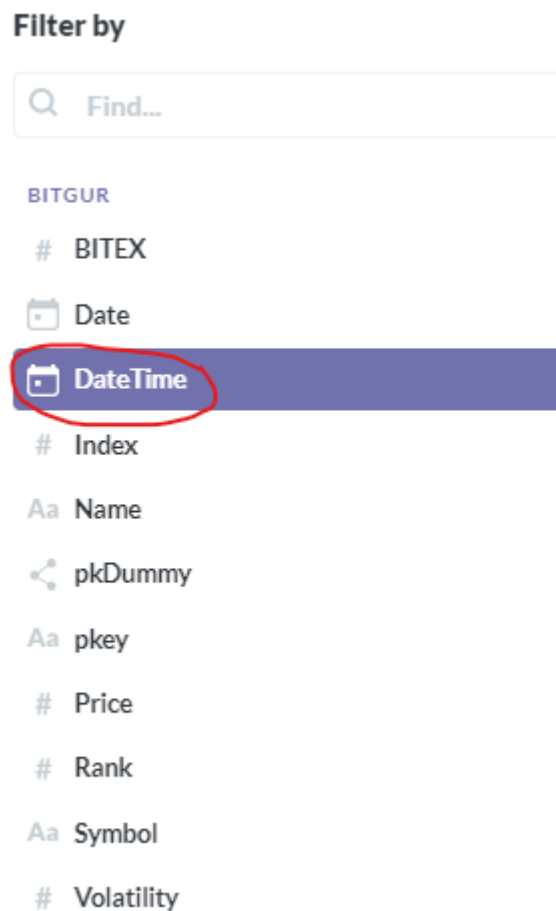


Figure 4.3: *please be patient as this loads or do the same with a different table that contains less historical data*



- Let's add a filter to only select data where the **DateTime** field is from the last 7 days:
- Now that we have filtered the data down to the subset we are interested in, let's adjust the look of the table using the **Settings** option in the bottom left:

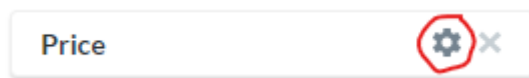
Octoparse • Bitgur

DateTime Previous 7 Days X

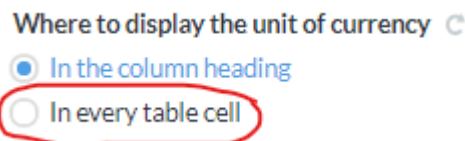
pkey	Name	Symbol	BITEX	Date	DateTime	Index	pkDummy	Price (\$)	Rank	Volatility
2019-12-20 00BTC	Bitcoin	BTC	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	7,087.80	1	57.7
2019-12-20 00ETH	Ethereum	ETH	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	126.10	2	46.5
2019-12-20 00XRP	XRP	XRP	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	0.19	3	49.49
2019-12-20 00USDT	Tether	USDT	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	1	4	20.53
2019-12-20 00BCH	Bitcoin Cash	BCH	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	185.45	5	50.34
2019-12-20 00LTC	Litecoin	LTC	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	39.48	6	53
2019-12-20 00EOS	EOS	EOS	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	2.43	7	59.52
2019-12-20 00BNB	Binance Coin	BNB	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	13.12	8	45.35
2019-12-20 00BSV	Bitcoin SV	BSV	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	87.33	9	109.85
2019-12-20 00XTZ	Tezos	XTZ	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	1.51	10	69.62
2019-12-20 00PAX	Paxos Stan...	PAX	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	1.00	31	7.51
2019-12-20 00ZEC	ZCash	ZEC	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	28.55	32	47.57
2019-12-20 00SNX	Synthetic ...	SNX	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	1.35	33	66.52
2019-12-20 00FTT	FTX Token	FTT	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	2.03	34	54.22
2019-12-20 00DCR	Decred	DCR	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	17.08	35	92.27
2019-12-20 00BAT	Basic Atten...	BAT	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	0.16	36	73
2019-12-20 00XLM	Stellar	XLM	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	0.045	11	94.99
2019-12-20 00ADA	Cardano	ADA	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	0.034	12	48.65
2019-12-20 00LEO	UNUSSED ...	LEO	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	0.86	13	82.18
2019-12-20 00TRX	TRON	TRX	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	0.013	14	39.18
2019-12-20 00XMR	Monero	XMR	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	2019-12-20 00	47.55	15	64.47

Visualization Settings Visualization

- After clicking on the **Settings** button, you should see a sidebar show up on the left. First, let's edit the settings for the **Price** field:



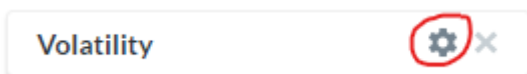
- \* And let's set the currency to be displayed with every cell instead of in parenthesis in the column name:



- Now the currency field will look like this for each row:

Price
\$7,087.80
\$126.10
\$0.19

- Let's also do a quick edit to the **Volatility** field settings:




- \* Enable the setting to **Show a mini bar chart**:

### < Volatility

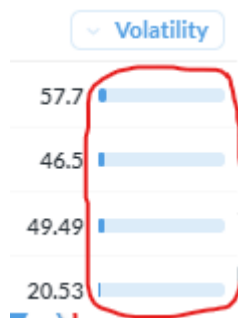
Column title

Volatility

Show a mini bar chart 



- \* Now each row should have a bar that is filled based on the current value compared to the minimum and maximum values for that field for the current subset of the data:



- If we sort the data by **Volatility** in *descending* order, we can see the row that sets the bar for the maximum for the rest of the rows:

Name	Symbol	BITEX	Date	DateTime	Index	Price	Rank	Volatility
Budbo	BUBO	52.72	December 13, 2019, 12:00 AM	December 13, 2019, 12:03 AM	19.53	\$9.76	999.37	

- \* Now let's also add some conditional formatting to the **Volatility** field, which will color each row when a certain criteria is met. Go back to the **Settings** page of the table, and this time click on **Conditional Formatting**:

### < Table options

Columns

Conditional Formatting

### Conditional formatting

You can add rules to make the cells in this table change color if they meet certain conditions.

+ Add a rule

- Now we can create a rule to highlight rows with a value for **Volatility** > 50:

### < Table options

Columns

Conditional Formatting

Which columns should be affected?

Volatility



Formatting style

☒ Single color

☐ Color range

When a cell in this column...

is greater than



50

...turn its background this color:



Highlight the whole row



Update rule

- The table's rows should now be highlighted when the criteria is met:

Name	Symbol	BITEX	Date	DateTime	Index	Price	Rank	Volatility
Bitcoin	BTC	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	\$7,087.80	1	57.7
Ethereum	ETH	50.34	December 20, 2019, 12:00 AM	December 20, 2019, 12:01 AM	56.66	\$126.10	2	46.5

## 4.2.1 Exporting Data

(make a note about confidentiality of the data)



## 4.3 Answer Questions

## 4.4 Creating dashboards



## Chapter 5

# Documentation Usage

Everything you have visibility to within the Metabase environment is documented. Meaning, every table you see has a description which includes the URL of where the data is collected (as well as the web scraping methodology where informative), and every field within those tables has also been documented on [predictcryptodb.com](http://predictcryptodb.com). Every table you have visibility to also has an e-mail alert system that alerts me if the data stops flowing through on an hourly (or daily for some) basis. If anything becomes broken I will remove the visibility of that table within Metabase, so if a table is found within the Metabase environment you can assume the web scraper for that table is still functioning correctly and that the data is uploaded to the table on an automated schedule. I think Metabase could do a better job at making the documentation for the different tables more intuitive to find, so hopefully this guide can provide some clarity where Metabase might be lacking.

### 5.1 pkDummy and pkey

### 5.2 Navigating the Metabase interface

### 5.3 Using x-rays to explore a table

- caveat that the database doesn't have the greatest resources that a big company would be able to provide and how this database costs around \$220 a month to maintain vs. what a company would spend

(could really use a section on explaining the way pkDummy, pkey work and how they follow two different time zones, show Alteryx trick to deal with the trickiness of UTC not having daylight savings and our local times having it)



## Chapter 6

# Additional Tips

### 6.1 Creating Variables for queries

- Within new question, click on the “x” symbol to open sidebar on the side and learn how to use variables:
  - `{{variable_name}}` creates a variable in this SQL template called “variable\_name”. Variables can be given types in the side panel, which changes their behavior. All variable types other than “Field Filter” will automatically cause a filter widget to be placed on this question; with Field Filters, this is optional. When this filter widget is filled in, that value replaces the variable in the SQL template.

### 6.2 Embedding anything created on Metabase

### 6.3 Creating e-mail triggers

#### 6.3.1 Pulses

### 6.4 Using the Shrimpy-Python Library

This section is a bit more *expert level* and I don’t really expect anyone to go down this route, but I figured it was worth having a small section on using the Shrimpy software for trading cryptocurrencies by using their Python package. This is a particularly good way to execute trades on the cryptocurrency markets because rather than having to connect to and manage each API individually, you can connect up to 16 exchanges to your Shrimpy account and programmatically trade on every one of them using the same set of functions in Python.

```
## Warning: Closing open result sets
## [1] TRUE
```