Task 0: Setup

Before you click the Start Lab button
Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

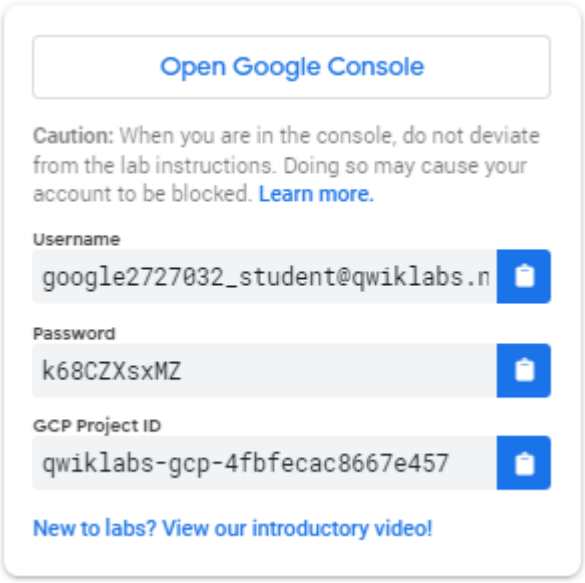
This Qwiklabs hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

What you need
To complete this lab, you need:

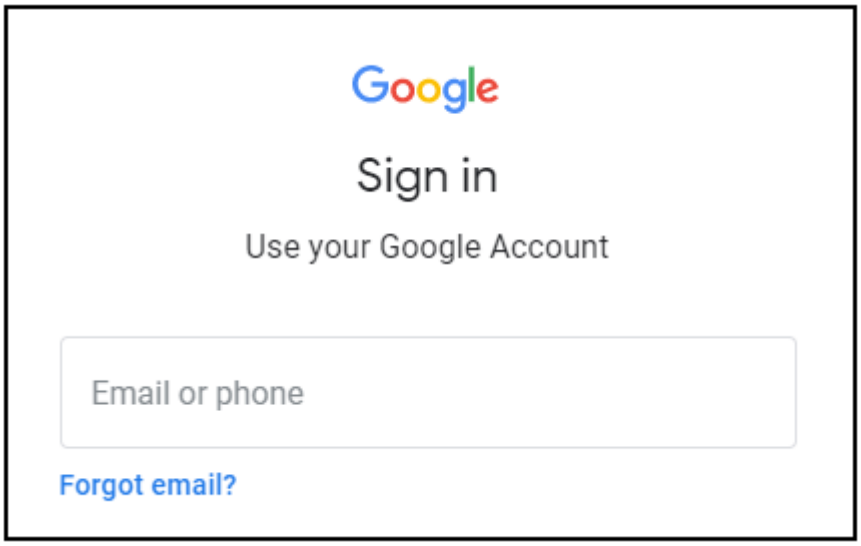
•Access to a standard internet browser (Chrome browser recommended).
•Time to complete the lab.
Note: If you already have your own personal Google Cloud account or project, do not use it for this lab.

Note: If you are using a Chrome OS device, open an Incognito window to run this lab.

How to start your lab and sign in to the Google Cloud Console
1.Click the **Start Lab** button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is a panel populated with the temporary credentials that you must use for this lab.

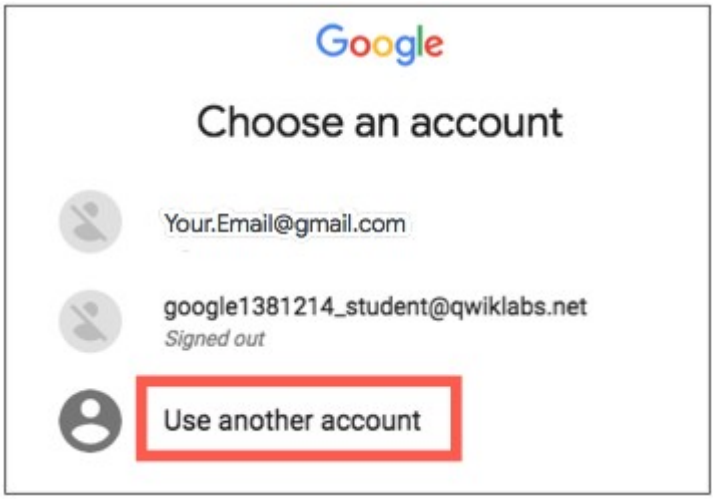


2.Copy the username, and then click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Sign in** page.



Tip: Open the tabs in separate windows, side-by-side.

Choose an account page, click **Use Another**



Account.

3.In the **Sign in** page, paste the username that you copied from the Connection Details panel. Then copy and paste the password.

Important: You must use the credentials from the Connection Details panel. Do not use your Qwiklabs credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

4.Click through the subsequent pages:

- •Accept the terms and conditions.
- •Do not add recovery options or two-factor authentication (because this is a temporary account).
- •Do not sign up for free trials.

After a few moments, the Cloud Console opens in this tab.

Note: You can view the menu with a list of Google Cloud Products and Services by clicking the **Navigation menu** at the top-left.

Task 1: View the cryptocurrencies in the public dataset

1. Open **Navigation Menu > BigQuery**.

2. Click **+ ADD DATA > Explore public datasets** .

3. In **Search for solutions**, type bitcoin and press **enter**.

4. Click **Bitcoin Cash Cryptocurrency Dataset**.

5. Click **VIEW DATASET**.

A new tab will open with BigQuery, and you should be on the bigquery-public-data:crypto_bitcoin_cash dataset.

All the public datasets are visible.

6. In **Type to search**, type crypto.
Only the public datasets starting with crypto will be displayed. Expand the datasets so you can see they all share the same structure. This makes performing queries across the different cryptocurrencies easy as the tables, views, and fields are identical in each cryptocurrency dataset.

Task 2: Perform a simple query

In this task you will view the famous 10,000 bitcoin pizza purchase transaction. You can read about what happened.

1.Copy and paste this query into the query window and then press **Run**.

```
SELECT * FROM `bigquery-public-data.crypto_bitcoin.transactions` as transactions WHERE transactions.hash = 'a1075db55d416d3ca199f55b6084e2115b9345e16c5cf302fc80e9d5fbf5d48d'content_copy
```

You can see the raw data returned.

Points to note:

•Values for input, output and fee are in satoshis. The satoshi is currently the smallest unit of the bitcoin currency recorded on the block chain. It is a one hundred millionth of a single bitcoin (0.00000001 BTC).
•The sent amount is via multiple inputs all coming from the same address.
•The output amount is sent as a single transaction to the one address.

Task 3: Validate the data

In this task you will check that you can access the cryptocurrency datasets by performing simple validation queries on two cryptocurrencies.

Double-entry book query of Bitcoin Cash

To motivate an initial exploration of these new datasets, let’s start with a simple example, comparing the way to query both payments and receipts across multiple cryptocurrencies. This comparison is the simplest way to verify that a cryptocurrency is operating as intended, and at least operationally, is a mathematically correct store of value.

1.Copy and paste this query into the query window and then press **Run**.

```
-- SQL source from https://cloud.google.com/blog/products/data-analytics/introducing-six-new-cryptocurrencies-in-bigquery-public-datasets-and-how-to-analyze-them
WITH double_entry_book AS (
    -- debits
    SELECT
      array_to_string(inputs.addresses, ",") as address
    , inputs.type
    , -inputs.value as value
    FROM `bigquery-public-data.crypto_bitcoin.inputs` as inputs
    UNION ALL
    -- credits
    SELECT
      array_to_string(outputs.addresses, ",") as address
    , outputs.type
    , outputs.value as value
    FROM `bigquery-public-data.crypto_bitcoin.outputs` as outputs
)
SELECT
    address
,   type
,   sum(value) as balance
FROM double_entry_book
GROUP BY 1,2
ORDER BY balance DESC
LIMIT 100content_copy
```

Verify the Bitcoin values returned are accurate

2.In another browser tab open the website https://www.blockchain.com/search? It will say Oops! - this is ok.

3.From the **All Blockchains** dropdown, under Mainnet select **Bitcoin** .

4.Copy the top address returned in BigQuery results.

5.Paste into the search box and click **Search**.

6.Check the final balance returned - it should be the same as the balance listed in BigQuery results for that address.

NOTE: If it is out, that it will either be due to a rounding difference in the adding (small fraction), or that the 24 hour delay in the dataset is the issue.

Double-entry book query of Dogecoin

7.Modify the query to use dogecoin, and run the query.

Verify the Dogecoin values returned are accurate

8.In another browser tab open the website https://dogechain.info/

9.Copy the top address returned in BigQuery results.

10.Paste into the search box and click search.

11.Note the balance returned, it should be the same as the balance listed in BigQuery results for that address.

NOTE: If it is out, that it will either be due to a rounding difference in the adding (small fraction), or that the 24 hour delay in the dataset is the issue.

Note that the only difference between them is the name of the data location.

You can swap in Bitcoin Cash, Dash, Litecoin, and Zcash in a similar fashion.

Task 4: Plot the Gini coefficient for cryptocurrency

Beyond quality control and auditing applications, presenting cryptocurrency in a traditional format enables integration with other financial data management systems. As an example, let's consider a common economic measure, the Gini Coefficient. In the field of macroeconomics, the Gini Coefficient is a member of a family of econometric measures of wealth inequality. Values range between 0.0 and 1.0, with completely distributed wealth (all members have the same amount) mapping to a value of 0.0 and completely accumulated wealth (one member has everything) mapping to 1.0.

Typically, the Gini Coefficient is estimated for a specific country's economy based on data sampling or imputation. For crypto-economies, we have complete transparency of the data at the highest possible resolution.

In addition to data transparency, one of the purported benefits of cryptocurrencies is that they allow the implementation of money to more closely resemble the implementation of digital information. It follows that a fully digitized money network will come to resemble the internet, with reduced transactional friction and fewer barriers that impede capital flow. Frequently, implicit in this narrative is that capital will distribute more equally. But we don't always observe that particular outcome, and the crypto-assets presented here display a broad spectrum of distribution patterns over time. You can read more about using the Gini coefficient to reason about crypto-economic network performance in Quantifying Decentralization.

To set a baseline to interpret our findings, consider how resources are distributed in traditional, non-crypto economies. According to a World Bank analysis in 2013, recent Gini coefficients for world economies have a mean value of 39.6 (with a standard deviation of 9.6).

Create the query

In this next query you will calculate the gini coefficient of dash, on a week-by-week basis. This query will take about a minute to execute. Once you have the data you can easily visualize the graph and compare it with the one generated for the source article.

1.Copy and paste this query into the query window and then press **Run**.

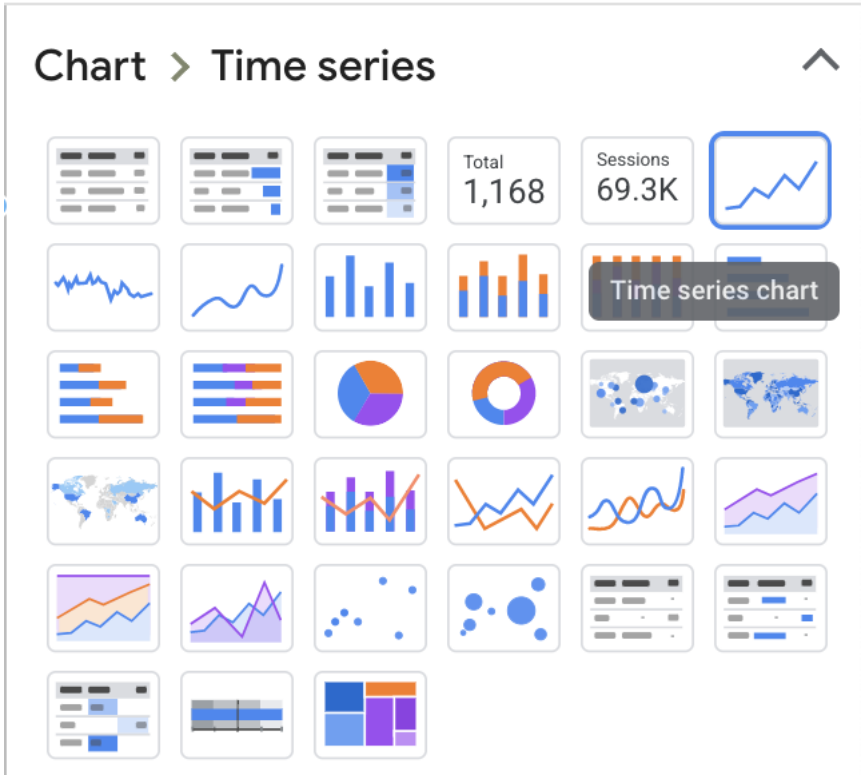
```
-- SQL source from https://gist.github.com/allenday/1500cc268f24ae89b7adfc25c74967b0
WITH double_entry_book AS (
    -- debits
    SELECT
      array_to_string(inputs.addresses, ",") as address
    , inputs.type
    , -inputs.value as value
    , block_timestamp
    FROM `bigquery-public-data.crypto_dash.inputs` as inputs
    UNION ALL
    -- credits
    SELECT
      array_to_string(outputs.addresses, ",") as address
    , outputs.type
    , outputs.value as value
    , block_timestamp
    FROM `bigquery-public-data.crypto_dash.outputs` as outputs
)
,double_entry_book_by_date as (
    select
        date(block_timestamp) as date,
        address,
        sum(value / POWER(10,0)) as value
    from double_entry_book
    group by address, date
)
,daily_balances_with_gaps as (
    select
```

```
            address,
            date,
            sum(value) over (partition by address order by date) as balance,
            lead(date, 1, current_date()) over (partition by address order by date) as next_date
            from double_entry_book_by_date
)
,calendar as (
      select date from unnest(generate_date_array('2009-01-12', current_date())) as date
)
,daily_balances as (
      select address, calendar.date, balance
      from daily_balances_with_gaps
      join calendar on daily_balances_with_gaps.date <= calendar.date and calendar.date <
daily_balances_with_gaps.next_date
)
,supply as (
      select
            date,
            sum(balance) as daily_supply
      from daily_balances
      group by date
)
,ranked_daily_balances as (
      select
            daily_balances.date,
            balance,
            row_number() over (partition by daily_balances.date order by balance desc) as rank
      from daily_balances
      join supply on daily_balances.date = supply.date
      where safe_divide(balance, daily_supply) >= 0.0001
      ORDER BY safe_divide(balance, daily_supply) DESC
)
select
      date,
      -- (1 − 2B) https://en.wikipedia.org/wiki/Gini_coefficient
      1 - 2 * sum((balance * (rank - 1) + balance / 2)) / count(*) / sum(balance) as gini
from ranked_daily_balances
group by date
order by date asccontent_copy
```

Use Data Studio to visualize the query

2.In the Query Results ribbon, click **EXPLORE DATA > Explore with Data Studio** and authorize BigQuery.

3.Click **Time Series Chart**.



4.Change the metric to **gini**.

5.Compare the graph with the original source article graph (reproduced below). In the chart below you are looking for the brown line. Starting in December 2019, see how it's changed since then.

Generate the gini coefficient for litecoin

6.Perform the same steps as you did for the Dash cryptocurrency to get the gini coefficient, but this time perform it for Litecoin.

Task 5: Explore two famous cryptocurrency events

There is a blog post on the Blockchain site that describes a number of memorable events in bitcoin. You will perform two queries on the bitcoin public dataset to retrieve the data in a couple of those events.

November 22, 2013: This bitcoin transaction sparks mystery and speculation

In the fall of 2013, a 194,993 bitcoin transaction hit the network, which caused many to wonder who was behind this very large transaction. The value at the time of the transaction in USD was over $149 million. CoinDesk wrote "unsurprisingly, a transaction of that size has prompted the bitcoin community to do some analysis and detective work. The transaction involved a large number of sending addresses, with some of them from blocks mined in February 2010 or even earlier, prompting excited speculation they might be from Satoshi Nakamoto, bitcoin’s absent (and likely pseudonymous) founder."

1.Write a SQL query to find the transaction_hash for transfer of 194993 bitcoins (BTC) and write that data to a table called 51 in the dataset lab. Below is SQL to help you get started:

```
CREATE OR REPLACE TABLE lab.51 (transaction_hash STRING) as
SELECT -- write the rest of the select statement (remember to use where) ...content_copy
```

This will require you to understand the structure of the **transactions** table, or you can use the **outputs** view. Your query must return the transaction hash, other fields are not important.

Hint: The unique amount makes this easy to query for that value. You will need to convert the value to satoshis.

Below is

After all this time, what is the balance of the address that purchased the two pizzas for 10,000 BTC?

2.Write a SQL query to write the balance for the account that paid 10,000 BTC for the two pizzas in 2010, into the table 52 in the lab dataset.

The output must have the balance for the address.

Hint: Modify the query from task 3 to select only rows with the purchaser address (using a WHERE clause). To get the purchaser address you can perform the query from task 2 and copy the inputs address (you do not need to include that query as part of the solution for this activity).

Here is some SQL to start you off:

```
CREATE OR REPLACE TABLE lab.52 (balance NUMERIC) as
WITH double_entry_book AS (
    -- debits
    SELECT
      array_to_string(inputs.addresses, ",") as address
    , -inputs.value as value
    FROM `bigquery-public-data.crypto_bitcoin.inputs` as inputs
    UNION ALL
    -- credits
```

```
    SELECT
      array_to_string(outputs.addresses, ",") as address
    , outputs.value as value
    FROM `bigquery-public-data.crypto_bitcoin.outputs` as outputs
)
SELECT -- write the rest of the select statement (remember to use where) ...content_copy
```

Answer:

Task - 5.1 : Store the transaction hash of the large mystery transfer of 194993 BTC in the table 51 inside the lab dataset:

```sql
CREATE OR REPLACE TABLE lab.51 (transaction_hash STRING) as
SELECT transaction_id FROM `bigquery-public-data.bitcoin_blockchain.transactions` , UNNEST( outputs ) as outputs
where outputs.output_satoshis = 19499300000000
```

SQL

Task - 5.2 : Store the balance of the pizza purchase address in the table 52 inside the lab dataset:

```sql
-- SQL source from https://cloud.google.com/blog/product...
CREATE OR REPLACE TABLE lab.52 (balance NUMERIC) as
WITH double_entry_book AS (
  -- debits
  SELECT
   array_to_string(inputs.addresses, ",") as address
  , -inputs.value as value
  FROM `bigquery-public-data.crypto_bitcoin.inputs` as inputs
  UNION ALL
  -- credits
  SELECT
   array_to_string(outputs.addresses, ",") as address
  , outputs.value as value
```

```
14   FROM `bigquery-public-data.crypto_bitcoin.outputs` as outputs
15
16)
17SELECT
18sum(value) as balance
19FROM double_entry_book
20where address = "1XPTgDRhN8RFnzniWCddobD9iKZatrvH4"
```

Congratulations!

Google Cloud Training & Certification

...helps you make the most of Google Cloud technologies. Our classes include technical skills and best practices to help you get up to speed quickly and continue your learning journey. We offer fundamental to advanced level training, with on-demand, live, and virtual options to suit your busy schedule. Certifications help you validate and prove your skill and expertise in Google Cloud technologies.

Manual Last Updated: February 01, 2021

Lab Last Tested: February 01, 2021