- List tables in a Dataset.

```
bq ls crypto_data
```

```
  tableId   Type    Labels   Time Partitioning
 --------- ------- -------- -------------------
  markets   TABLE
```

- List the recent executed jobs. This includes both load jobs and queries executed.

```
bq ls -j
```

```
jobId                        Job Type  State     Start Time       Duration
---------------------------  --------  --------  ---------------  --------
bquxjob_767fb332_16625172a52  load      SUCCESS   29 Sep 07:29:27  0:00:10
bquxjob_2a33184c_16625141949  load      SUCCESS   29 Sep 07:26:06  0:00:13
bquxjob_582a116b_16624b3717a  query     SUCCESS   29 Sep 05:41:20  0:00:01
bquxjob_7b18cd73_16624a0f378  query     SUCCESS   29 Sep 05:40:32  0:00:01
```

Loading Data Using the Command-Line bq Utility

The following commands walk through loading a dataset into BigQuery using the bq utility via the terminal:

- Create a new Dataset.

```
bq mk crypto_data_terminal
```

```
Dataset 'secret-country-192905:crypto_data_terminal' successfully
created.
```

- List the datasets to confirm creation of new Dataset.

```
bq ls
```

```
       datasetId
 ----------------------
  crypto_data
  crypto_data_terminal
```

- Load data as a Table into the newly created Dataset. We load the file using the ‘bq load’ command. This command loads data in a new or existing table. In our example, we load the data from the GCS bucket ‘gs://my-test-data/crypto-markets.csv’ into a newly created table named ‘markets_terminal’ with the schema “slug,symbol,name,date, ranknow,open,high,low,close,volume,market,close_ratio,spread”

  ```
  bq load crypto_data_terminal.markets_terminal gs://my-test-data/
  crypto-markets.csv slug,symbol,name,date,ranknow,open,high,low,
  close,volume,market,close_ratio,spread
  ```

- List the tables in the dataset.

  ```
  bq ls crypto_data_terminal

        tableId        Type    Labels   Time Partitioning
   ------------------ ------- -------- -------------------
    markets_terminal   TABLE
  ```

- Examine the table schema.

  ```
  bq show crypto_data_terminal.markets_terminal

  Table secret-country-192905:crypto_data_terminal.markets_terminal

     Last modified            Schema          Total Rows   Total
  Bytes   Expiration   Time Partitioning   Labels
   ----------------- ------------------------ ------------ ---------
  ---- ------------ ------------------- --------
    29 Sep 09:12:24   |- slug: string         498381       52777964
                      |- symbol: string
                      |- name: string
                      |- date: string
                      |- ranknow: string
                      |- open: string
                      |- high: string
                      |- low: string
                      |- close: string
                      |- volume: string
  ```

```
                            |- market: string
                            |- close_ratio: string
                            |- spread: string
```

- Delete a table.

  ```
  bq rm crypto_data_terminal.markets_terminal
  ```

- Delete a Dataset. This command will delete a Dataset with all its containing tables.

  ```
  bq rm -r crypto_data_terminal
  ```

BigQuery SQL

In this section, we'll have an overview of SQL by executing some examples that gives a broad perspective of what can be achieved with SQL. New users who have not used SQL before will benefit from this section. Also, SQL is amazingly easy and intuitive to use that non-technical people like personnel in marketing and sales are experts at this even sometimes more than programmers. It is an expressive declarative language.

BigQuery works with both the standard SQL which supports SQL 2011 standard and the legacy SQL syntax which is a non-standard variant of SQL. However, standard SQL is the preferred query syntax for BigQuery. In experimenting with SQL, we will work with the **census_bureau_international** public dataset. The following queries are available in the chapter notebook of the book repository.

Filtering

The following query selects the fertility rate for each country in the year 2018 from the 'age_specific_fertility_rates' table in the 'census_bureau_international' dataset. The resulting table is arranged in descending order.

```
bq query --use_legacy_sql=false 'SELECT
  country_name AS country,
  total_fertility_rate AS fertility_rate
FROM
  `bigquery-public-data.census_bureau_international.age_specific_fertility_
  rates`
```