BigQuery

Introduction to BigQuery

Data Warehouse History

Databases

- Primarily used for transaction processing
- Difficult for managers to analyze data and create reports when the data resides in numerous databases across an organization

Data warehouses

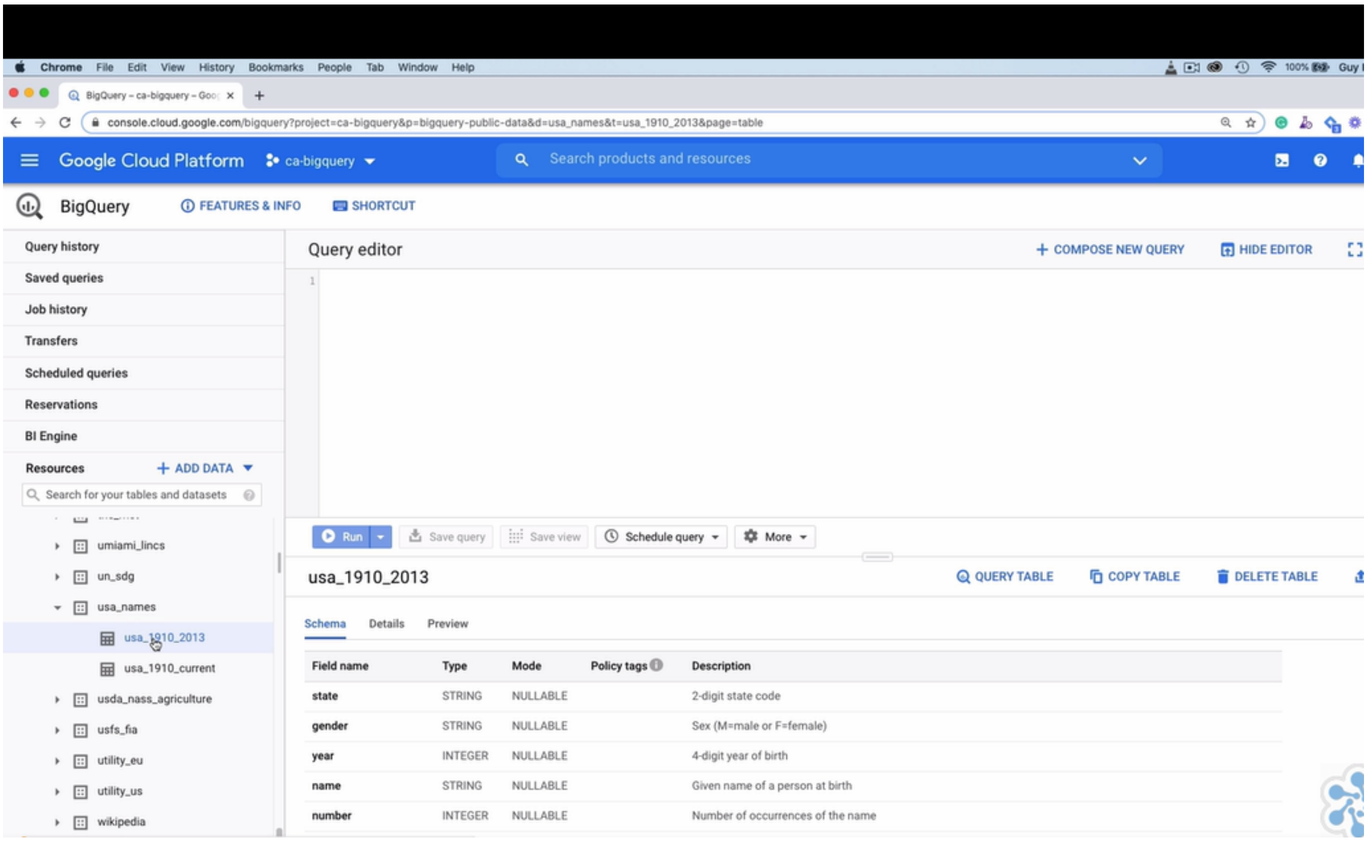
- Collect data from wide variety of sources
- Designed for reporting and data analysis

Why Use BigQuery

- Ease of implementation
  - Building our own is expensive, time-consuming, and difficult to scale
  - With BigQuery, just load data and pay only for what you use
- Speed
  - Processes billions of rows in seconds
  - Handles real-time analysis of streaming data

Running a Query in GCP Console

1. Login to GCP, go to the console, then select **BigQuery** from the menu or type **BigQuery** in the search bar



console.cloud.google.com/bigquery?project=ca-bigquery&p=bigquery-public-data&d=usa_names&t=usa_1910_2013&page=table

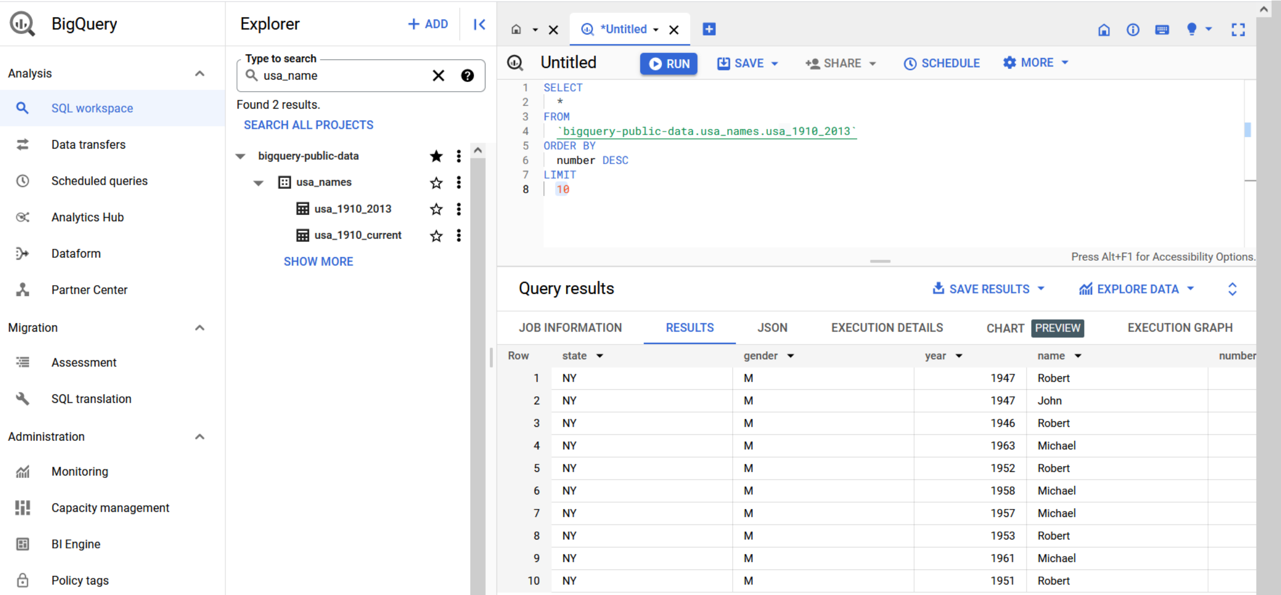
Field name	Type	Mode	Policy tags	Description
state	STRING	NULLABLE		2-digit state code
gender	STRING	NULLABLE		Sex (M=male or F=female)
year	INTEGER	NULLABLE		4-digit year of birth
name	STRING	NULLABLE		Given name of a person at birth
number	INTEGER	NULLABLE		Number of occurrences of the name

2. Click **+ Compose a new query**

3. In the query editor, enter a valid GoogleSQL query, then click **Run**

   - **Query 1**

```
SELECT
  *
FROM
  `bigquery-public-data.usa_names.usa_1910_2013`
ORDER BY
  number DESC
LIMIT
  10
```

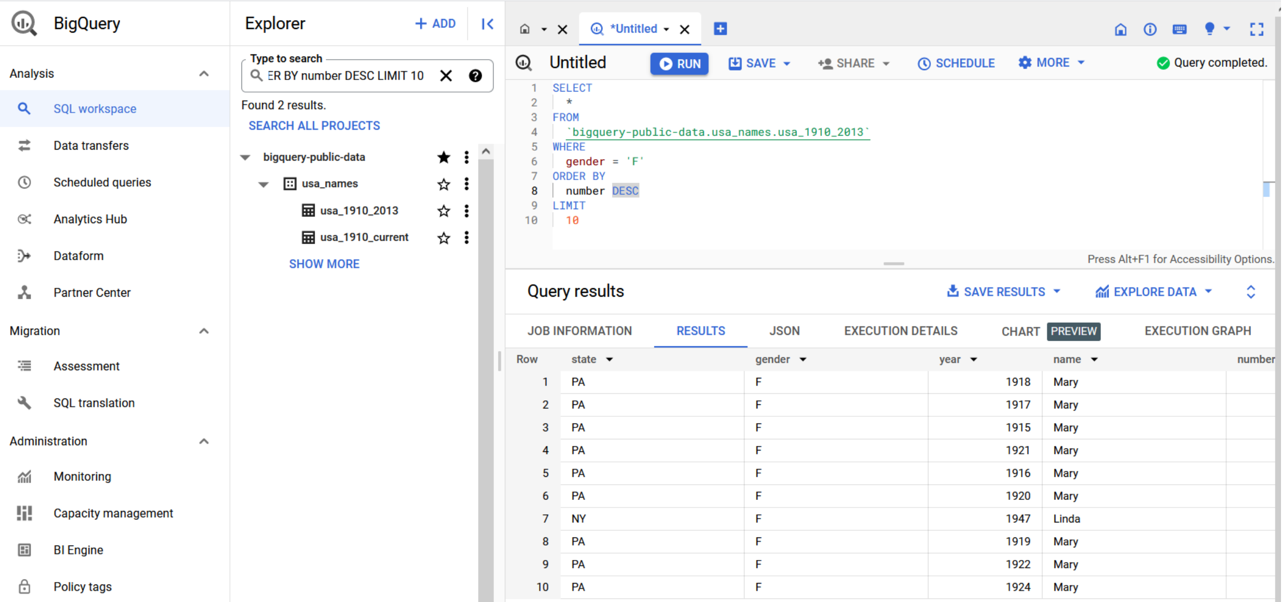


```
SELECT
  *
FROM
  `bigquery-public-data.usa_names.usa_1910_2013`
ORDER BY
  number DESC
LIMIT
  10
```

Row	state	gender	year	name	number
1	NY	M	1947	Robert	
2	NY	M	1947	John	
3	NY	M	1946	Robert	
4	NY	M	1963	Michael	
5	NY	M	1952	Robert	
6	NY	M	1958	Michael	
7	NY	M	1957	Michael	
8	NY	M	1953	Robert	
9	NY	M	1961	Michael	
10	NY	M	1951	Robert	

- **Query 2**

```
SELECT
  *
FROM
  `bigquery-public-data.usa_names.usa_1910_2013`
WHERE
  gender = 'F'
ORDER BY
  number DESC
LIMIT
  10
```

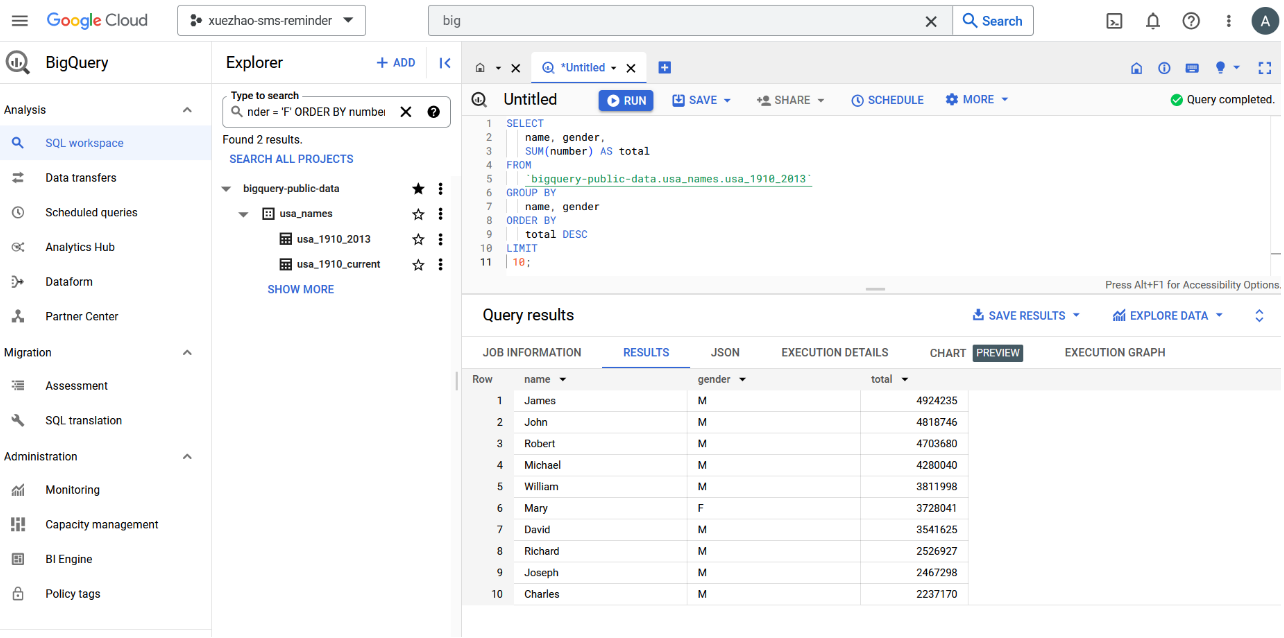


```
SELECT
  *
FROM
  `bigquery-public-data.usa_names.usa_1910_2013`
WHERE
  gender = 'F'
ORDER BY
  number DESC
LIMIT
  10
```

Row	state	gender	year	name	number
1	PA	F	1918	Mary	
2	PA	F	1917	Mary	
3	PA	F	1915	Mary	
4	PA	F	1921	Mary	
5	PA	F	1916	Mary	
6	PA	F	1920	Mary	
7	NY	F	1947	Linda	
8	PA	F	1919	Mary	
9	PA	F	1922	Mary	
10	PA	F	1924	Mary	

- **Query 3** Query the most common names in the United States between the years 1910 and 2013

```
SELECT
   name, gender,
    SUM(number) AS total
FROM
    `bigquery-public-data.usa_names.usa_1910_2013`
GROUP BY
    name, gender
ORDER BY
    total DESC
LIMIT
 10;
```

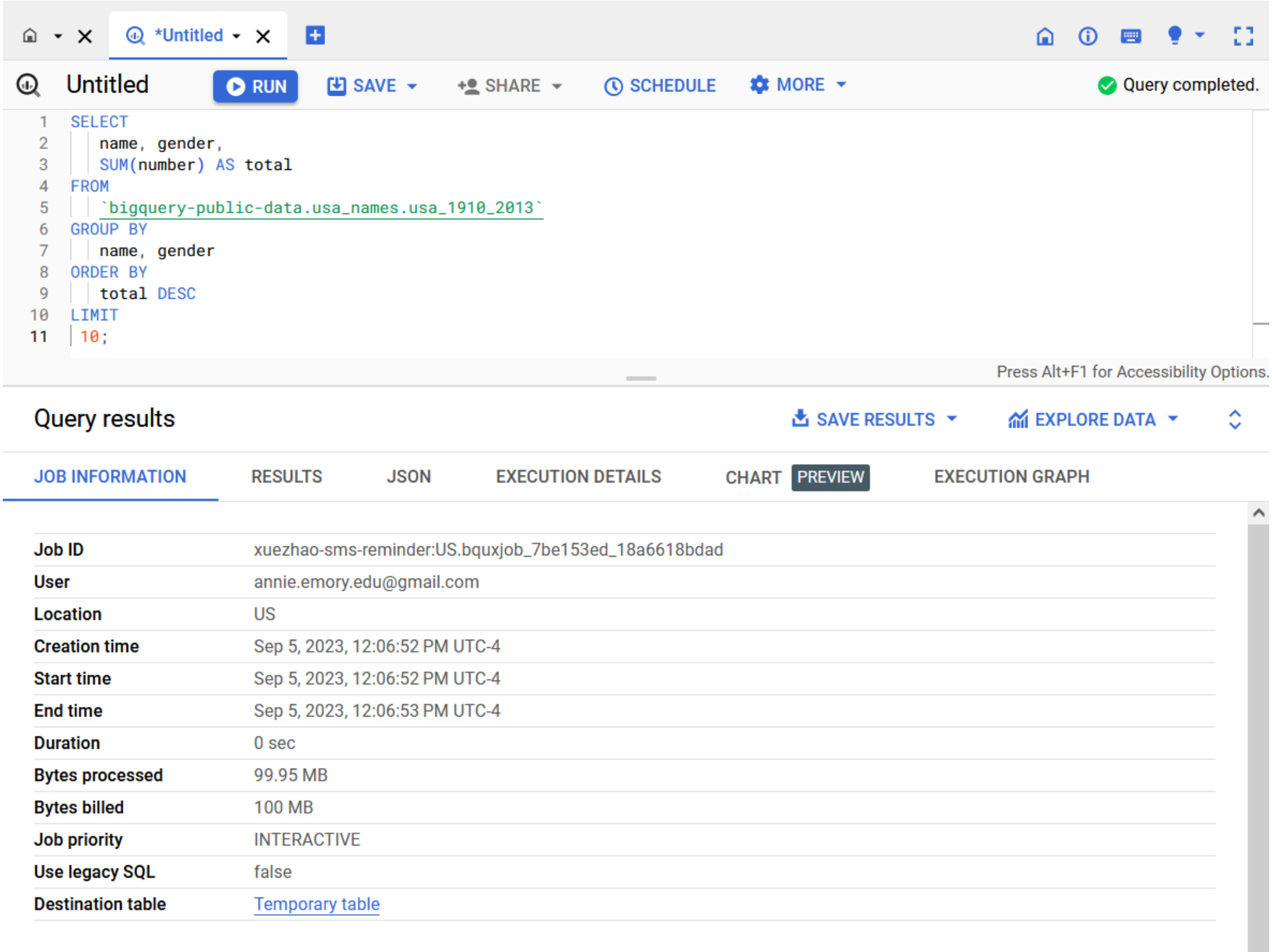


```
SELECT
  name, gender,
  SUM(number) AS total
FROM
  `bigquery-public-data.usa_names.usa_1910_2013`
GROUP BY
  name, gender
ORDER BY
  total DESC
LIMIT
 10;
```

Row	name	gender	total
1	James	M	4924235
2	John	M	4818746
3	Robert	M	4703680
4	Michael	M	4280040
5	William	M	3811998
6	Mary	F	3728041
7	David	M	3541625
8	Richard	M	2526927
9	Joseph	M	2467298
10	Charles	M	2237170

View BigQuery Results

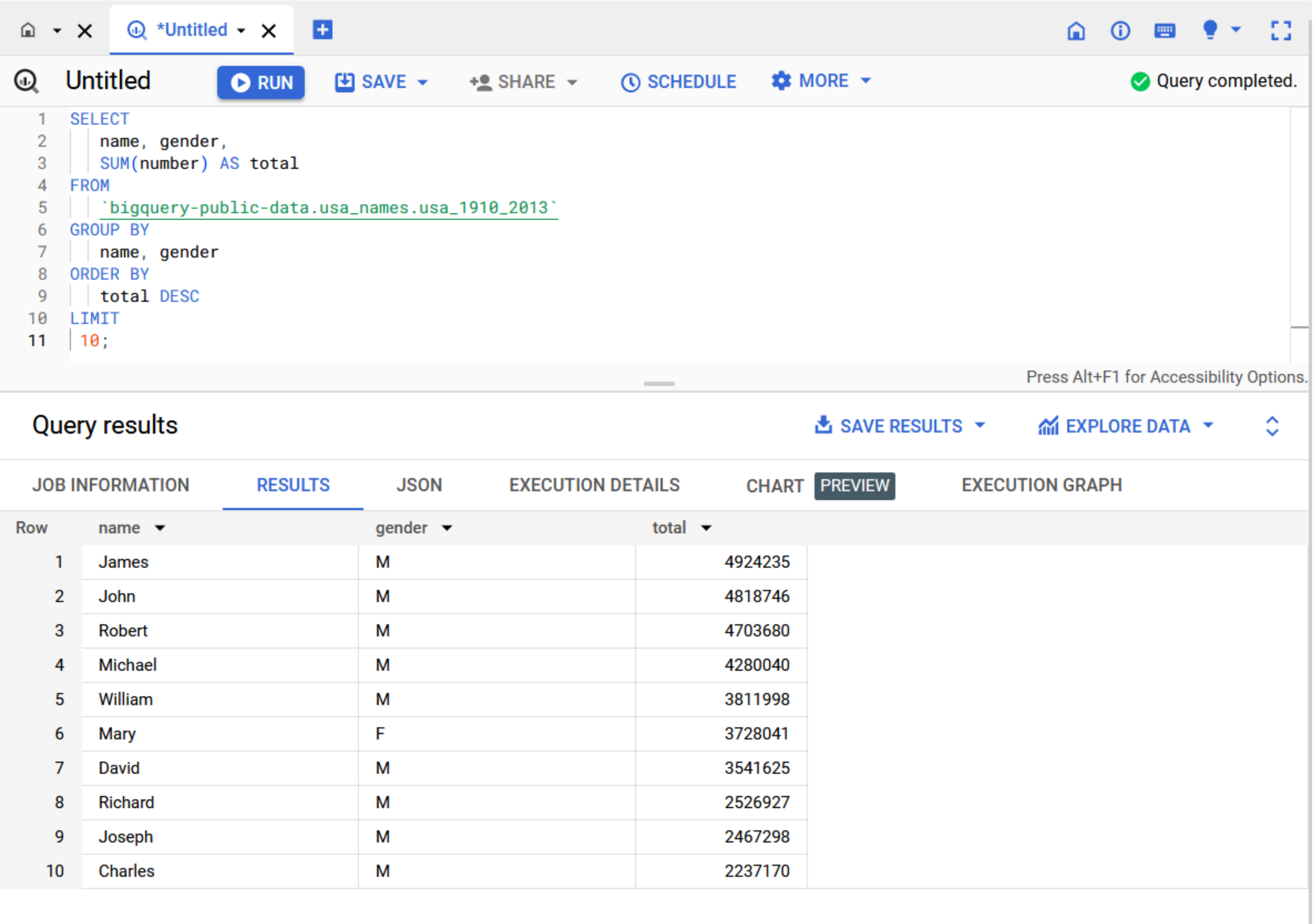
Job Information



```
SELECT
  name, gender,
  SUM(number) AS total
FROM
  `bigquery-public-data.usa_names.usa_1910_2013`
GROUP BY
  name, gender
ORDER BY
  total DESC
LIMIT
 10;
```

Job ID	xuezhao-sms-reminder:US.bquxjob_7be153ed_18a6618bdad
User	annie.emory.edu@gmail.com
Location	US
Creation time	Sep 5, 2023, 12:06:52 PM UTC-4
Start time	Sep 5, 2023, 12:06:52 PM UTC-4
End time	Sep 5, 2023, 12:06:53 PM UTC-4
Duration	0 sec
Bytes processed	99.95 MB
Bytes billed	100 MB
Job priority	INTERACTIVE
Use legacy SQL	false
Destination table	Temporary table

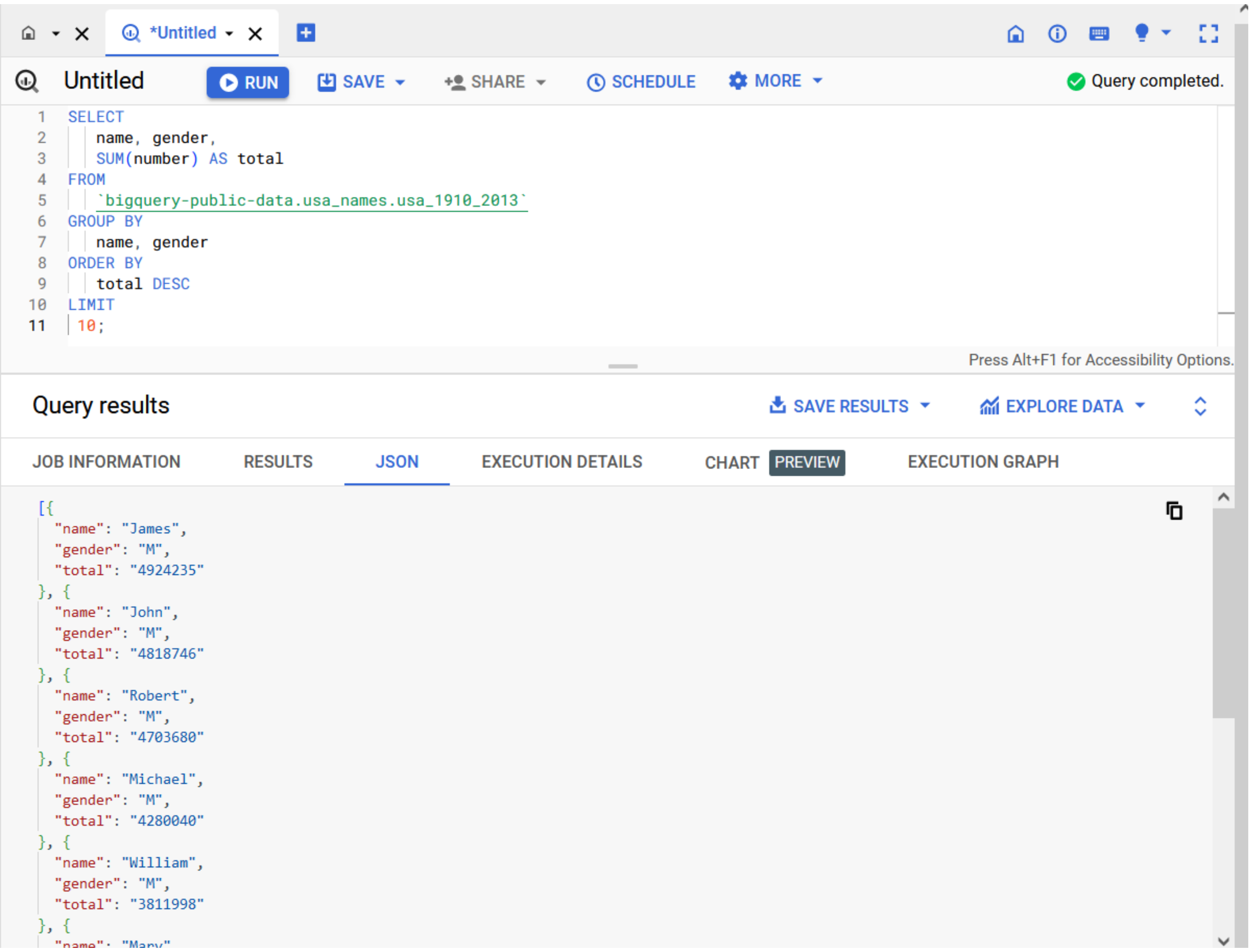
Results



```sql
SELECT
    name, gender,
    SUM(number) AS total
FROM
    `bigquery-public-data.usa_names.usa_1910_2013`
GROUP BY
    name, gender
ORDER BY
    total DESC
LIMIT
    10;
```

Row	name	gender	total
1	James	M	4924235
2	John	M	4818746
3	Robert	M	4703680
4	Michael	M	4280040
5	William	M	3811998
6	Mary	F	3728041
7	David	M	3541625
8	Richard	M	2526927
9	Joseph	M	2467298
10	Charles	M	2237170

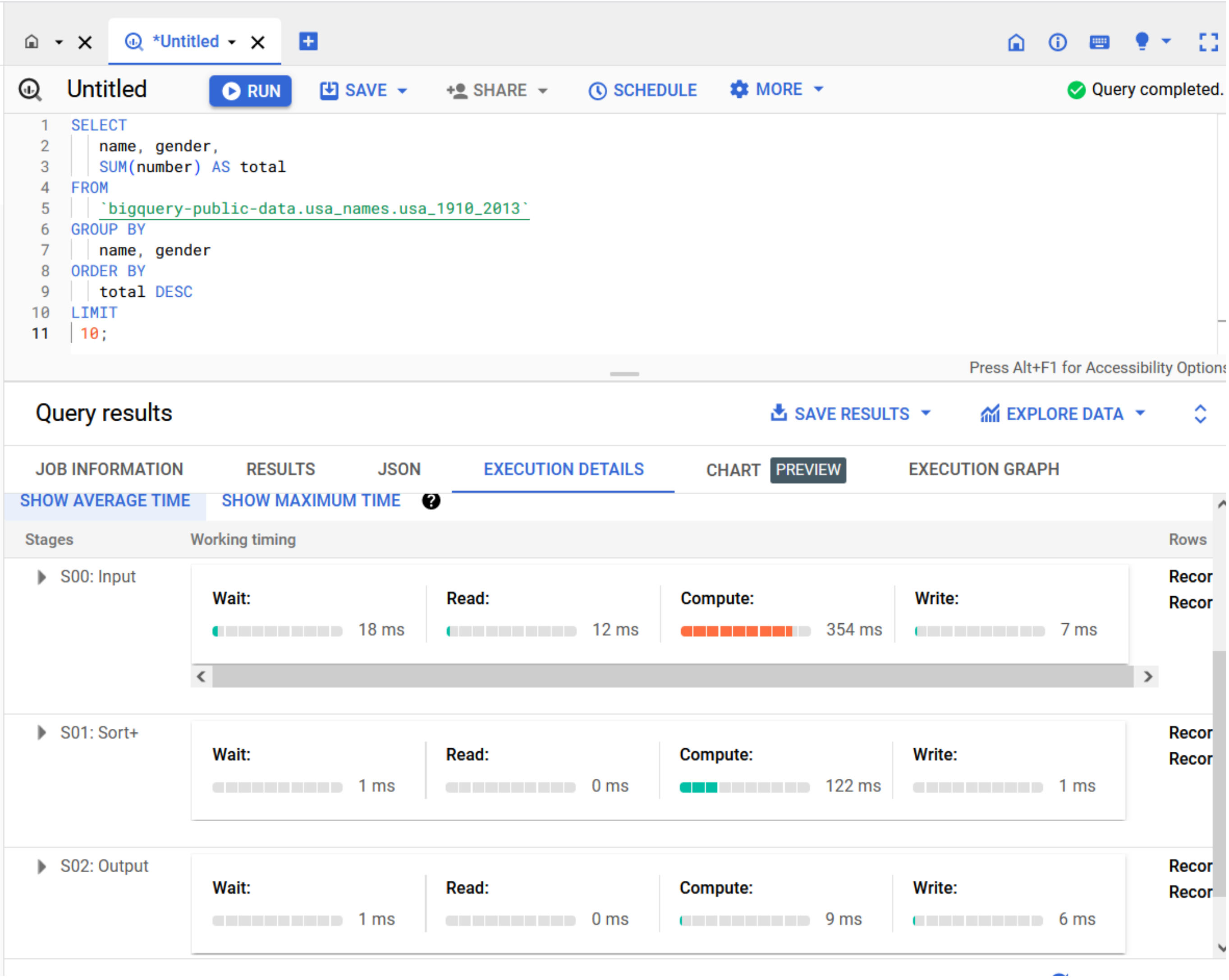
Json results



```sql
SELECT
  name, gender,
  SUM(number) AS total
FROM
  `bigquery-public-data.usa_names.usa_1910_2013`
GROUP BY
  name, gender
ORDER BY
  total DESC
LIMIT
  10;
```

```json
[{
  "name": "James",
  "gender": "M",
  "total": "4924235"
}, {
  "name": "John",
  "gender": "M",
  "total": "4818746"
}, {
  "name": "Robert",
  "gender": "M",
  "total": "4703680"
}, {
  "name": "Michael",
  "gender": "M",
  "total": "4280040"
}, {
  "name": "William",
  "gender": "M",
  "total": "3811998"
}, {
  "name": "Mary"
```

Execution Details



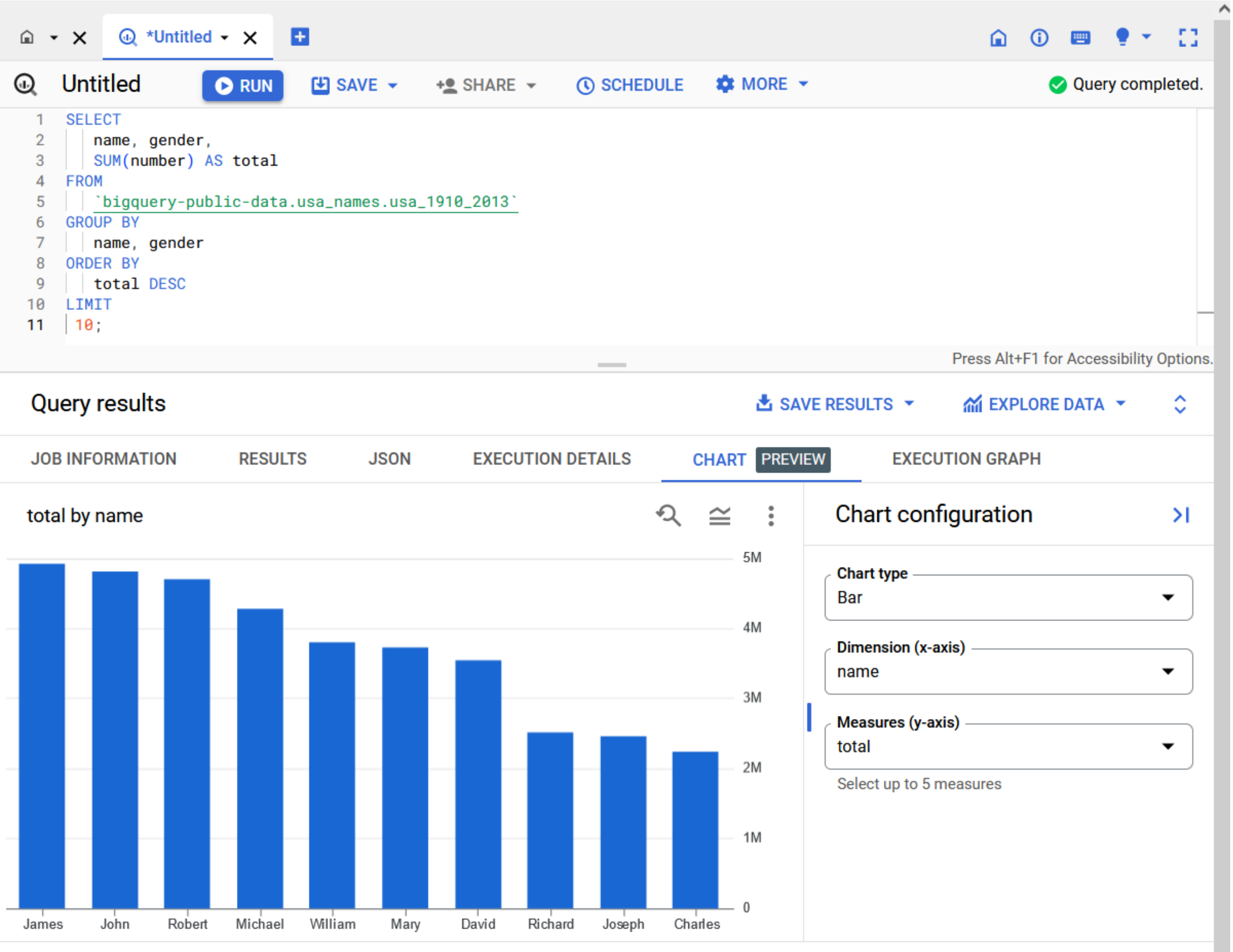
Query completed.

```sql
SELECT
    name, gender,
    SUM(number) AS total
FROM
    `bigquery-public-data.usa_names.usa_1910_2013`
GROUP BY
    name, gender
ORDER BY
    total DESC
LIMIT
  10;
```

Press Alt+F1 for Accessibility Options

Stages	Working timing				Rows
S00: Input	Wait: 18 ms	Read: 12 ms	Compute: 354 ms	Write: 7 ms	Recor Recor
S01: Sort+	Wait: 1 ms	Read: 0 ms	Compute: 122 ms	Write: 1 ms	Recor Recor
S02: Output	Wait: 1 ms	Read: 0 ms	Compute: 9 ms	Write: 6 ms	Recor Recor

Chart Preview



```
SELECT
  name, gender,
  SUM(number) AS total
FROM
  `bigquery-public-data.usa_names.usa_1910_2013`
GROUP BY
  name, gender
ORDER BY
  total DESC
LIMIT
  10;
```

Chart type: Bar

Dimension (x-axis): name

Measures (y-axis): total

Select up to 5 measures

Perform a Dry Run

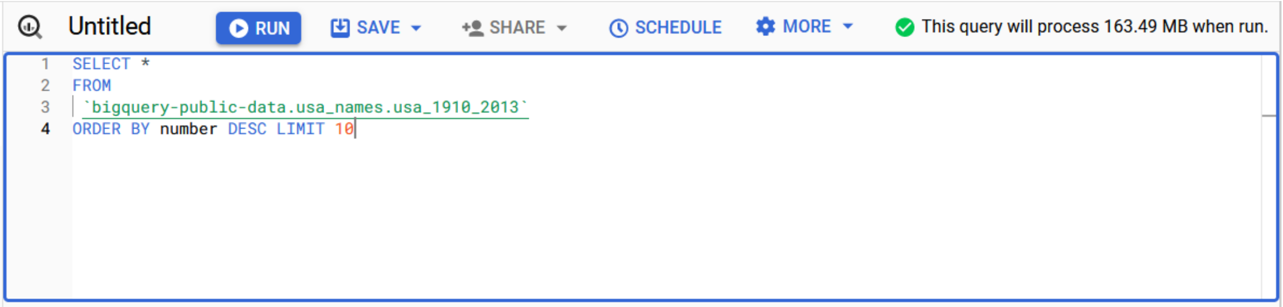
A dry run in BigQuery provides the following information:

- estimate of charges in on-demand mode
- validation of your query
- approximate size and complexity of your query in capacity mode

Dry runs don't use query slots, and you are not charged for performing a dry run. You can use the estimate returned by a dry run to calculate query costs in the pricing calculator.

- Go to the BigQuery page

- Enter your query in the query editor. If the query is valid, then a check mark automatically appears along with the amount of data that the query will process. If the query is invalid,



This query will process 163.49 MB when run.

```
SELECT *
FROM
  `bigquery-public-data.usa_names.usa_1910_2013`
ORDER BY number DESC LIMIT 10
```

BigQuery Pricing

BigQuery pricing has two main components:

- **Compute** (analysis) pricing is the cost to process queries, including SQL queries, user-defined functions, scripts, and certain data manipulation language (DML) and data definition language (DDL) statements.
  - Queries (on-demand) $6.25 per TB, the first 1 TB per month is free
- **Storage** - is the cost to store data that you load into BigQuery
  - $0.02 per GB per month
  - After 90 days with no edits, price drops to $0.01 per GB per month
  - No charges for reading data from storage

Cached or non-cached query results

- If no destination table specified, query results are cached in Temporary table
- Temporary table stays in cache for one day
- If you run a query again within 24 hours, there is no charge
- If you run a query again and specify a destination table to store results, it won't read from cache, and you will be charge.

Run Query from Python

Install Python Client for Google BigQuery

You can install BigQuery Python package by one of options below:

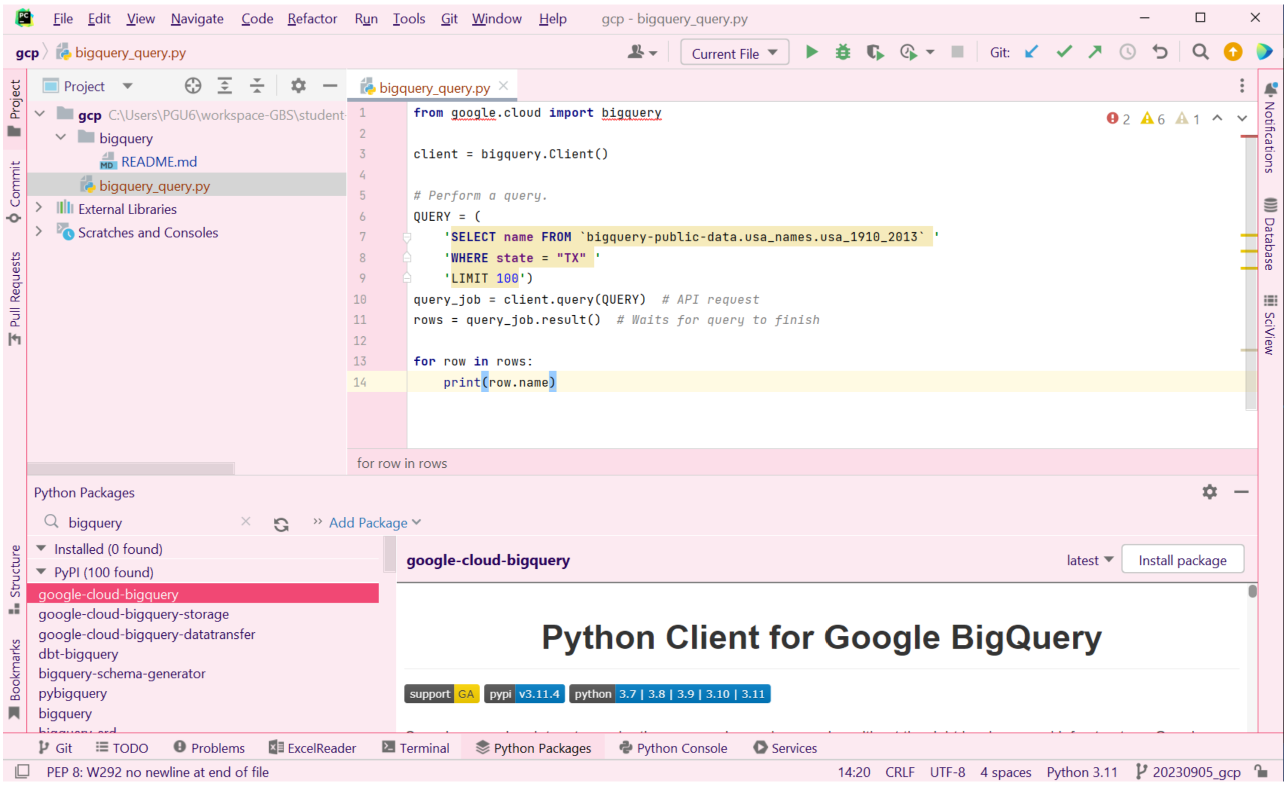
1. Install by PIP

Open a terminal, run the command:

```
pip install google-cloud-bigquery
```

2. Install by PyCharm

- Select **View -> Tool Windows -> Python Packages**
- Type **bigquery** in search field
- Select **google-cloud-bigquery** in the list
- Click **Install package**



```
from google.cloud import bigquery

client = bigquery.Client()

# Perform a query.
QUERY = (
    'SELECT name FROM `bigquery-public-data.usa_names.usa_1910_2013` '
    'WHERE state = "TX" '
    'LIMIT 100')
query_job = client.query(QUERY)  # API request
rows = query_job.result()  # Waits for query to finish

for row in rows:
    print(row.name)
```

PEP 8: W292 no newline at end of file

Python Script

Copy following Python code and save to the file bigquery_query.py

```
from google.cloud import bigquery
from google.oauth2 import service_account

## construct credentials from service account
credentials = service_account.Credentials.from_service_account_file(
    '<service-account.json>')

## construct a BigQuery client object
client = bigquery.Client(credentials=credentials)

# Perform a query.
QUERY = (
    'SELECT name FROM `bigquery-public-data.usa_names.usa_1910_2013` '
    'WHERE state = "TX" '
    'LIMIT 10')
query_job = client.query(QUERY)  # API request
rows = query_job.result()  # Waits for query to finish

for row in rows:
    print(row.name)
```

Important: Replace **<service-account.json>** with the actual full path of service account json file in you local laptop

Run Python Script

```
py bigquery_query.py

or
python bigquery_query.py
```

it outputs 10 names after the run.