

# SQL Exercises

*from*  
Data Science FGA Digitalent Kominfo

X  
Binar Academy

---

By: Siti Fatimatuzzahro



<https://www.linkedin.com/in/sitifatimatuzzahro/>



[sitifatim0010@gmail.com](mailto:sitifatim0010@gmail.com)



# Tools

Tools that used for solving the exercises are Google BigQuery, Table Plus, and Jupyter Notebook. I used different tools because there exercise that i can't solve with one tool only so that i need other tools to solve that.



# Exercise 1 – Chapter 1 Practice 1: Live Session

## Exercise Instructions:

### Preview Dataset

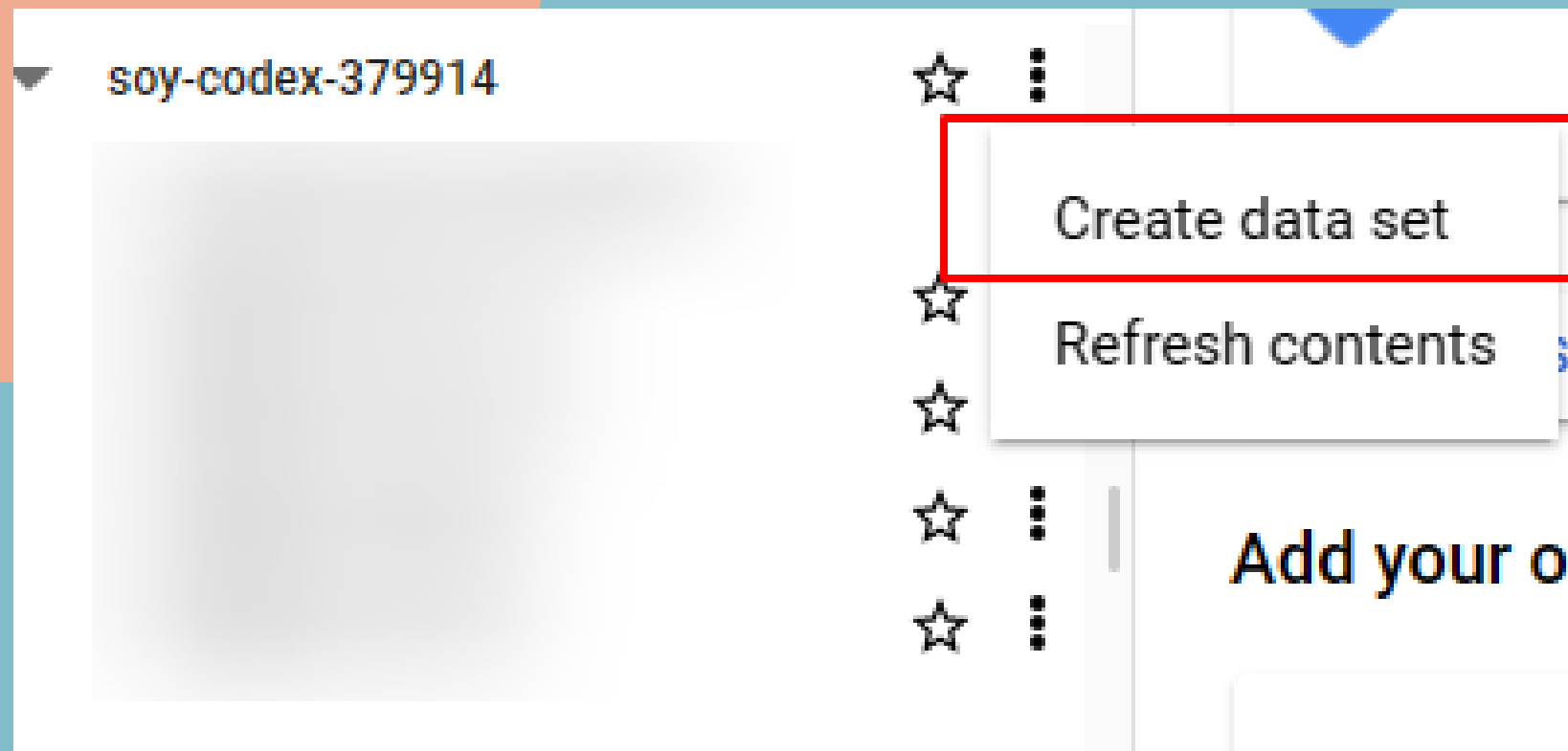
bulan	tahun	jenis	kode_trayek	trayek	jumlah_penumpang
Januari	2019	BRT	1	KORIDOR 1	2223257
Januari	2019	BRT	2	KORIDOR 2	781605
Januari	2019	BRT	3	KORIDOR 3	919.2
Januari	2019	BRT	4	KORIDOR 4	669417
Januari	2019	BRT	5	KORIDOR 5	890939
Januari	2019	BRT	6	KORIDOR 6	976411
Januari	2019	BRT	7	KORIDOR 7	872467
Januari	2019	BRT	8	KORIDOR 8	911731
Januari	2019	BRT	9	KORIDOR 9	1408962
Januari	2019	BRT	10	KORIDOR 10	737123
Januari	2019	BRT	11	KORIDOR 11	267869
Januari	2019	BRT	12	KORIDOR 12	208326
Januari	2019	BRT	13	KORIDOR 13	654831
Januari	2019	ANGKUTAN UMUM	1A	BALAIKOTA - PIK	93208
Januari	2019	ANGKUTAN UMUM	1B	ST. PALMERAH - TC	62726
Januari	2019	ANGKUTAN UMUM	1F	ST. PALMERAH - SE	33011
Januari	2019	ANGKUTAN UMUM	1C	BLOK M - PESANGG	44429
Januari	2019	ANGKUTAN UMUM	1E	BLOK M - PONDOK	94933
Januari	2019	ANGKUTAN UMUM	1H	TANAH ABANG - GC	66742

1. Download and import data into BigQuery
2. Answer this questions:
  - What are the types of Transjakarta transportation and routes used by the community in 2019?
  - When does Transjakarta transit receive the highest number of passengers? From the top 5 ridership data are they in the same month?
  - How many routes are there that start with ST?



# Import Dataset to BigQuery

1. Go to project and create new data set
2. Fill the data set id
3. Set the region (it can be other region too not limited to asia)
4. Click create data set button



### Create data set

Project ID  
soy-codex-379914 [CHANGE](#)

Data set ID \*   
Letters, numbers and underscores allowed

Location type [?](#)

☒ Region  
Specifying a region provides dataset colocation with other GCP services

☐ Multi-region  
Letting BigQuery select a region within a group of regions provides higher quota limits

Region \*  [▼](#)

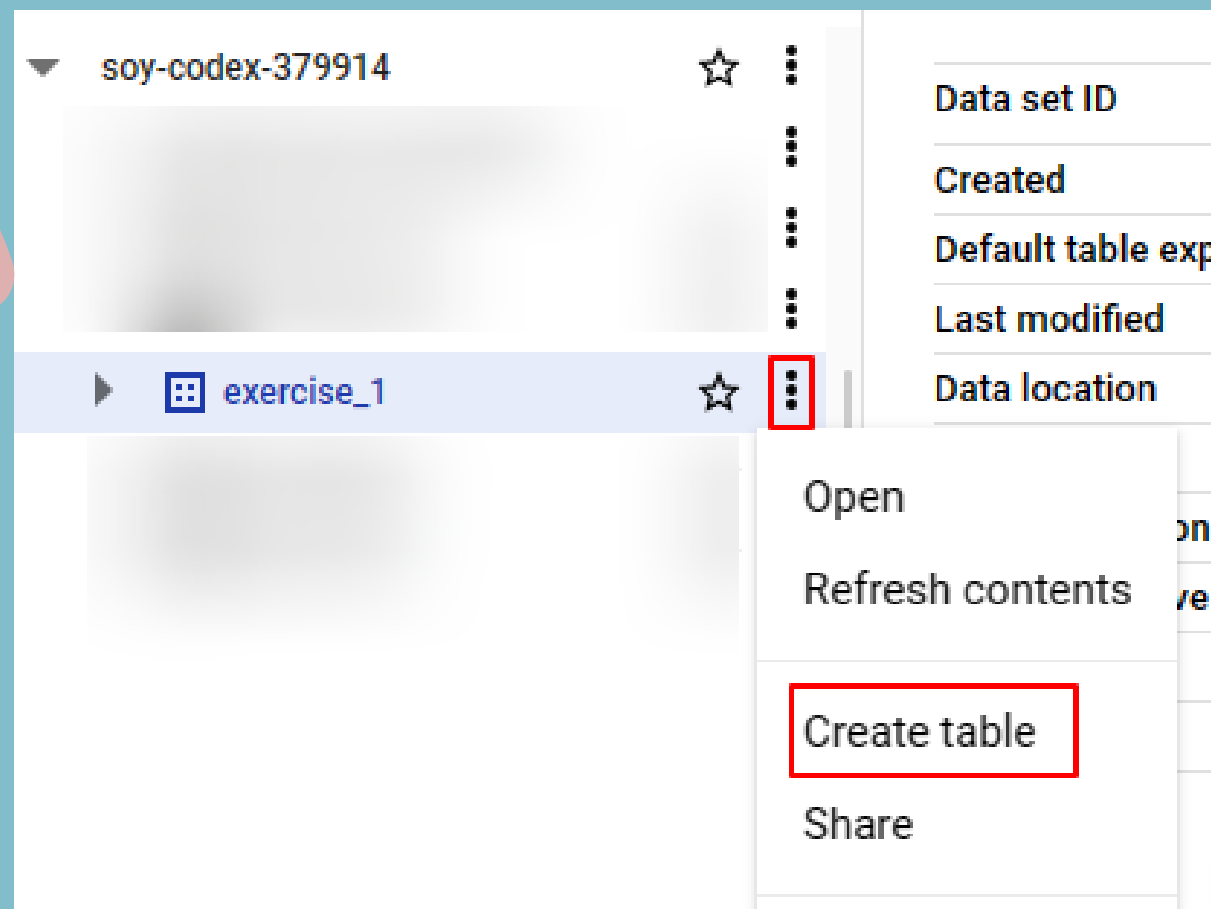
Default table expiry

☐ Enable table expiry [?](#)

Advanced options [▼](#)

[CREATE DATA SET](#) [CANCEL](#)

5. Click right side of data set and choose create table
6. Change source to upload
7. Upload the dataset
8. Fill the table name
9. Select the Auto Detection
10. Click create table



This screenshot shows the 'Create table' dialog box in the Google Cloud console. The dialog is divided into several sections:

- Source:**
  - Create table from:** A dropdown menu with 'Upload' selected (highlighted with a red box).
  - Select file \*:** A text input field containing 'Exercise1.csv' (highlighted with a red box). To the right are 'X' and 'BROWSE' buttons.
  - File format:** A dropdown menu with 'CSV' selected.
- Destination:**
  - Project \*:** A text input field containing 'soy-codex-379914' with a 'BROWSE' button to the right.
  - Data set \*:** A text input field containing 'exercise\_1'.
  - Table \*:** A text input field containing 'tabel\_exercise1' (highlighted with a red box). Below it, a note states: 'Unicode letters, marks, numbers, connectors, dashes or spaces allowed.'
  - Table type:** A dropdown menu with 'Native table' selected.
- Schema:**
  - Auto-detect:** A checkbox that is checked (highlighted with a red box).
  - A message box states: 'Schema will be automatically generated.'
- Partition and cluster settings:**
  - Partitioning:** A dropdown menu with 'No partitioning' selected.
  - Clustering order:** A text input field. Below it, a note states: 'Clustering order determines the sort order of the data. Clustering can be used on both partitioned and non-partitioned tables.'
- Advanced options:** A section that is currently collapsed.

At the bottom of the dialog, there are two buttons: 'CREATE TABLE' (highlighted with a red box) and 'CANCEL'.

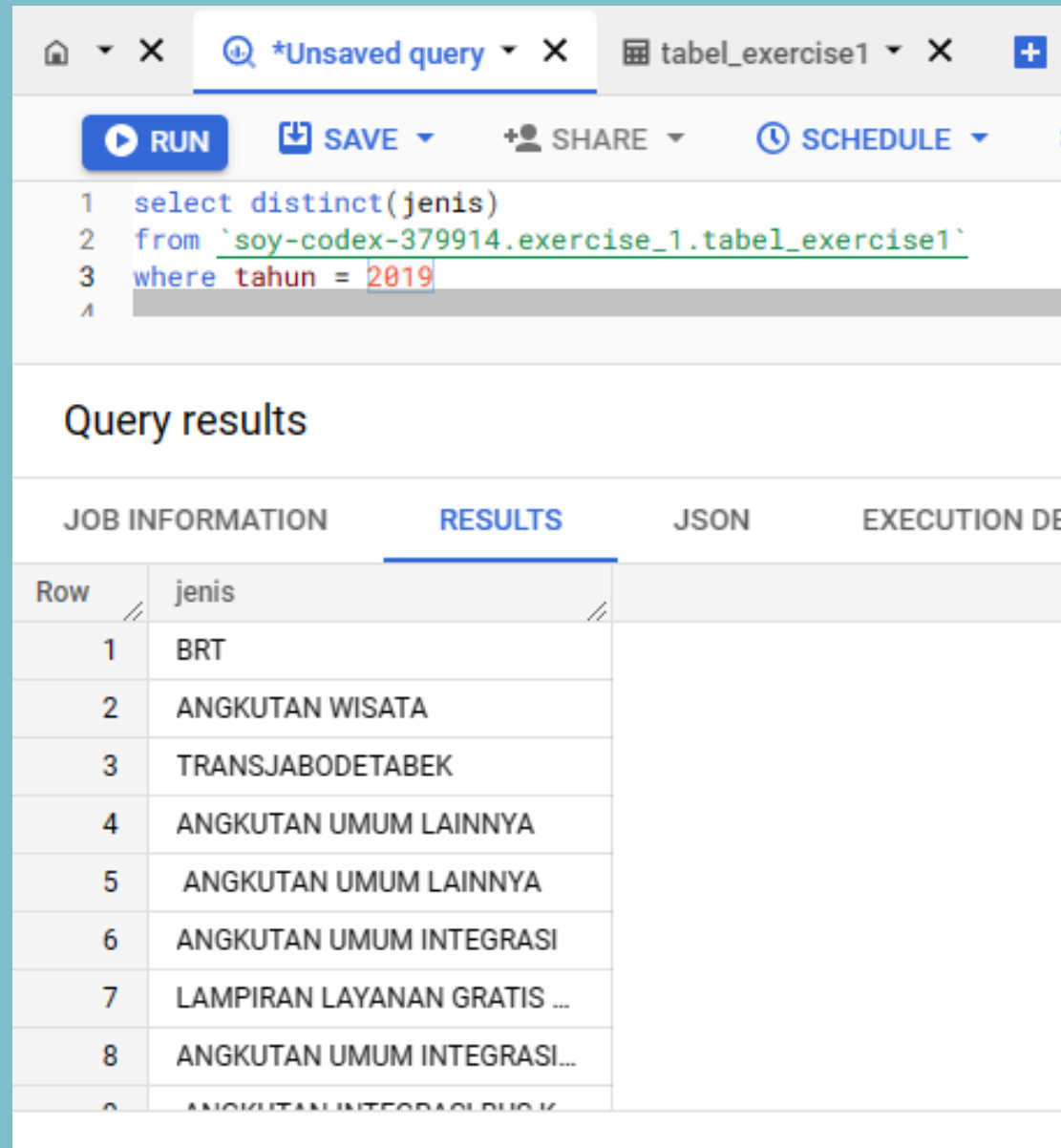
# preview dataset after successful import

tabel_exercise1							
SCHEMA DETAILS PREVIEW LINEAGE							
Row	bulan	tahun	jenis	kode_trayek	trayek	jumlah	
1	Januari	2019	BRT	1	KORIDOR 1	22	
2	Februari	2019	BRT	1	KORIDOR 1	19	
3	Maret	2019	BRT	1	KORIDOR 1	23	
4	April	2019	BRT	1	KORIDOR 1 (BLOK M - KOTA)	22	
5	Mei	2019	BRT	1	KORIDOR 1 (BLOK M - KOTA)	2.0	
6	Juni	2019	BRT	1	KORIDOR 1 (BLOK M - KOTA)	2.0	
7	Juli	2019	BRT	1	KORIDOR 1 (BLOK M - KOTA)	2.6	
8	Agustus	2019	BRT	1	KORIDOR 1 (BLOK M - KOTA)	2.4	
9	September	2019	BRT	1	KORIDOR 1 (BLOK M - KOTA)	2.4	
10	Oktober	2019	BRT	1	KORIDOR 1 (BLOK M - KOTA)	2.4	
11	November	2019	BRT	1	KORIDOR 1 (BLOK M - KOTA)	2.5	
12	Desember	2019	BRT	1	KORIDOR 1 (BLOK M - KOTA)	2,6	
13	Januari	2019	BRT	2	KORIDOR 2	20	



## Query results:

- What are the types of Transjakarta transportation and routes used by the community in 2019?

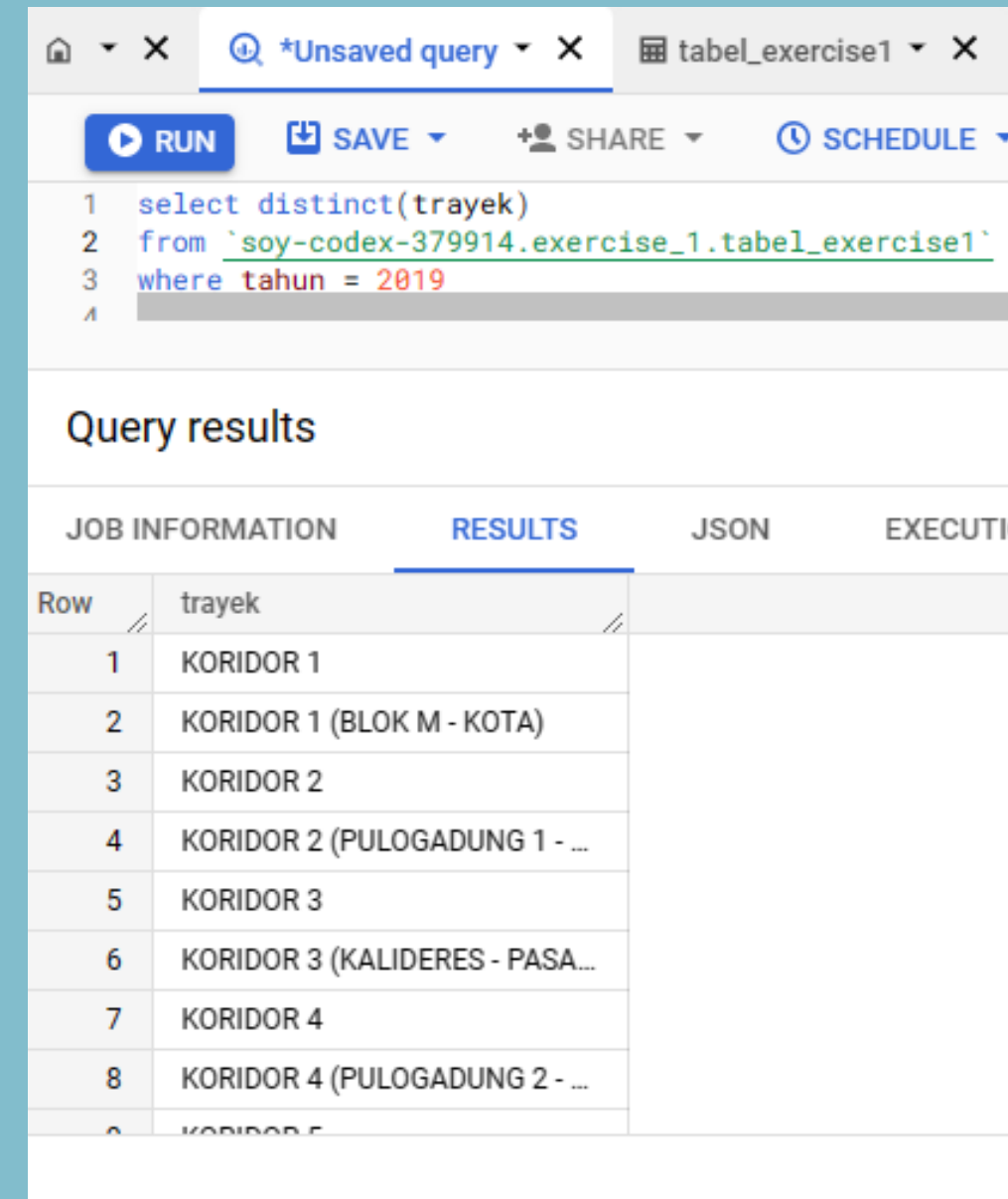


The screenshot shows a SQL query interface with a query editor at the top containing the following SQL code:

```
1 select distinct(jenis)
2 from `soy-codex-379914.exercise_1.tabel_exercise1`
3 where tahun = 2019
```

Below the query editor, the 'Query results' section is displayed with a tabbed interface. The 'RESULTS' tab is active, showing a table with the following data:

Row	jenis
1	BRT
2	ANGKUTAN WISATA
3	TRANSJABODETABEK
4	ANGKUTAN UMUM LAINNYA
5	ANGKUTAN UMUM LAINNYA
6	ANGKUTAN UMUM INTEGRASI
7	LAMPIRAN LAYANAN GRATIS ...
8	ANGKUTAN UMUM INTEGRASI...
9	ANGKUTAN INTEGRASI BUK...



The screenshot shows a SQL query interface with a query editor at the top containing the following SQL code:

```
1 select distinct(trayek)
2 from `soy-codex-379914.exercise_1.tabel_exercise1`
3 where tahun = 2019
```

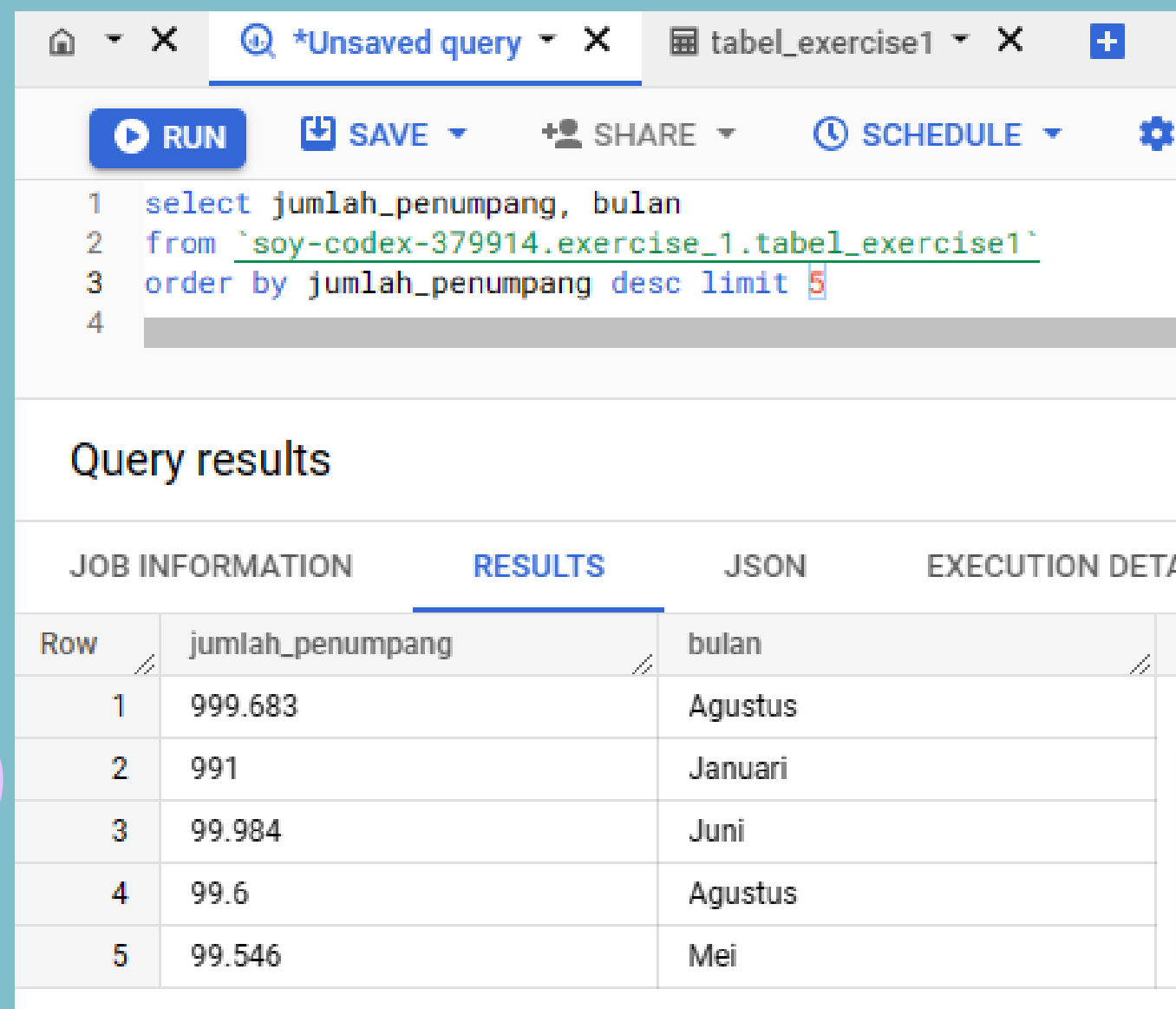
Below the query editor, the 'Query results' section is displayed with a tabbed interface. The 'RESULTS' tab is active, showing a table with the following data:

Row	trayek
1	KORIDOR 1
2	KORIDOR 1 (BLOK M - KOTA)
3	KORIDOR 2
4	KORIDOR 2 (PULOGADUNG 1 - ...
5	KORIDOR 3
6	KORIDOR 3 (KALIDERES - PASA...
7	KORIDOR 4
8	KORIDOR 4 (PULOGADUNG 2 - ...
9	KORIDOR 5



## Query results:

- When does Transjakarta transit receive the highest number of passengers? From the top 5 ridership data are they in the same month?
- How many routes are there that start with ST?

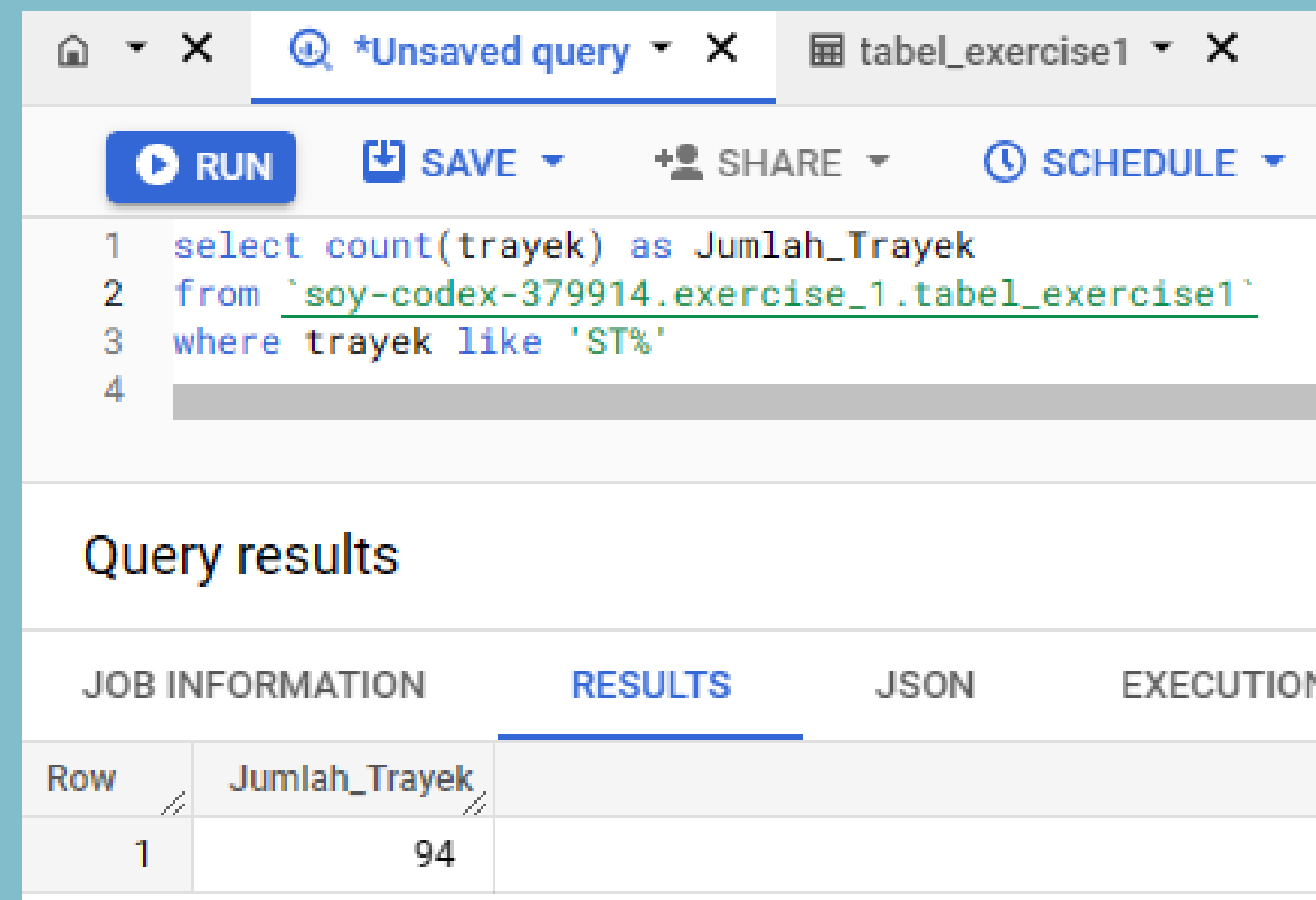


The screenshot shows a SQL query editor with a query to select the top 5 months by passenger count. The query is as follows:

```
1 select jumlah_penumpang, bulan
2 from `soy-codex-379914.exercise_1.tabel_exercise1`
3 order by jumlah_penumpang desc limit 5
4
```

Below the query editor, the "Query results" section is displayed, showing a table with two columns: "jumlah\_penumpang" and "bulan". The results are as follows:

Row	jumlah_penumpang	bulan
1	999.683	Agustus
2	991	Januari
3	99.984	Juni
4	99.6	Agustus
5	99.546	Mei



The screenshot shows a SQL query editor with a query to count the number of routes starting with 'ST'. The query is as follows:

```
1 select count(trayek) as Jumlah_Trayek
2 from `soy-codex-379914.exercise_1.tabel_exercise1`
3 where trayek like 'ST%'
4
```

Below the query editor, the "Query results" section is displayed, showing a table with two columns: "Jumlah\_Trayek". The results are as follows:

Row	Jumlah_Trayek
1	94

## Exercise 1 Conclusions:

1. The dataset seems not clean enough because in query number 1 result, it still gave duplicate Transjakarta types although the query included distinct in it.
2. The order of the displayed data in number 2 query result is strange because the number of passengers data type is a string so the order is done from the largest number in the string (9) not from the actual number of passengers in integer. Converting the string data type to integer is also not possible because some of the data is inconsistent such as 99.6 and 2,031,443. Then some also don't make sense because passengers (humans) can't be comma-valued.
3. To get more accurate query result, cleansing data set is needed.

# Exercise 2 – Chapter 1 Practice 2: Offline

## Exercise Instructions:

Organize unstructured data from the following news content into structured data

link: Ramalan BMKG Cuaca Hari Ini di Jawa & Bali: Surabaya Cerah Berawan, Yogya Berawan (kontan.co.id)

### Ramalan BMKG Cuaca Hari Ini di Jawa & Bali: Surabaya Cerah Berawan, Yogya Berawan

Rabu, 10 Agustus 2022 | 05:40 WIB Reporter: SS. Kurniawan



#### Serang

- Pagi hari: cerah berawan
- Siang hari: berawan
- Malam hari: berawan
- Dini hari: berawan
- Suhu: 24-33 derajat Celcius

Baca Juga: [Belakangan Cuaca Panas di Indonesia, Ini Penyebabnya Menurut BMKG](#)

#### Jakarta

- Pagi hari: cerah berawan
- Siang hari: berawan
- Malam hari: berawan
- Dini hari: cerah berawan
- Suhu: 25-30 derajat Celcius

# Query for creating and insert value into table:

```
SQL Query X ramalan_cuaca
1 CREATE TABLE ramalan_cuaca(
2     kota VARCHAR(20) NOT NULL AUTO_INCREMENT,
3     pagi VARCHAR(10) NOT NULL,
4     siang VARCHAR(10) NOT NULL,
5     dini VARCHAR(10) NOT NULL,
6     suhu VARCHAR(7) NOT NULL
7 )
8
```

SQL Query X ramalan\_cuaca

1 DESCRIBE ramalan\_cuaca;

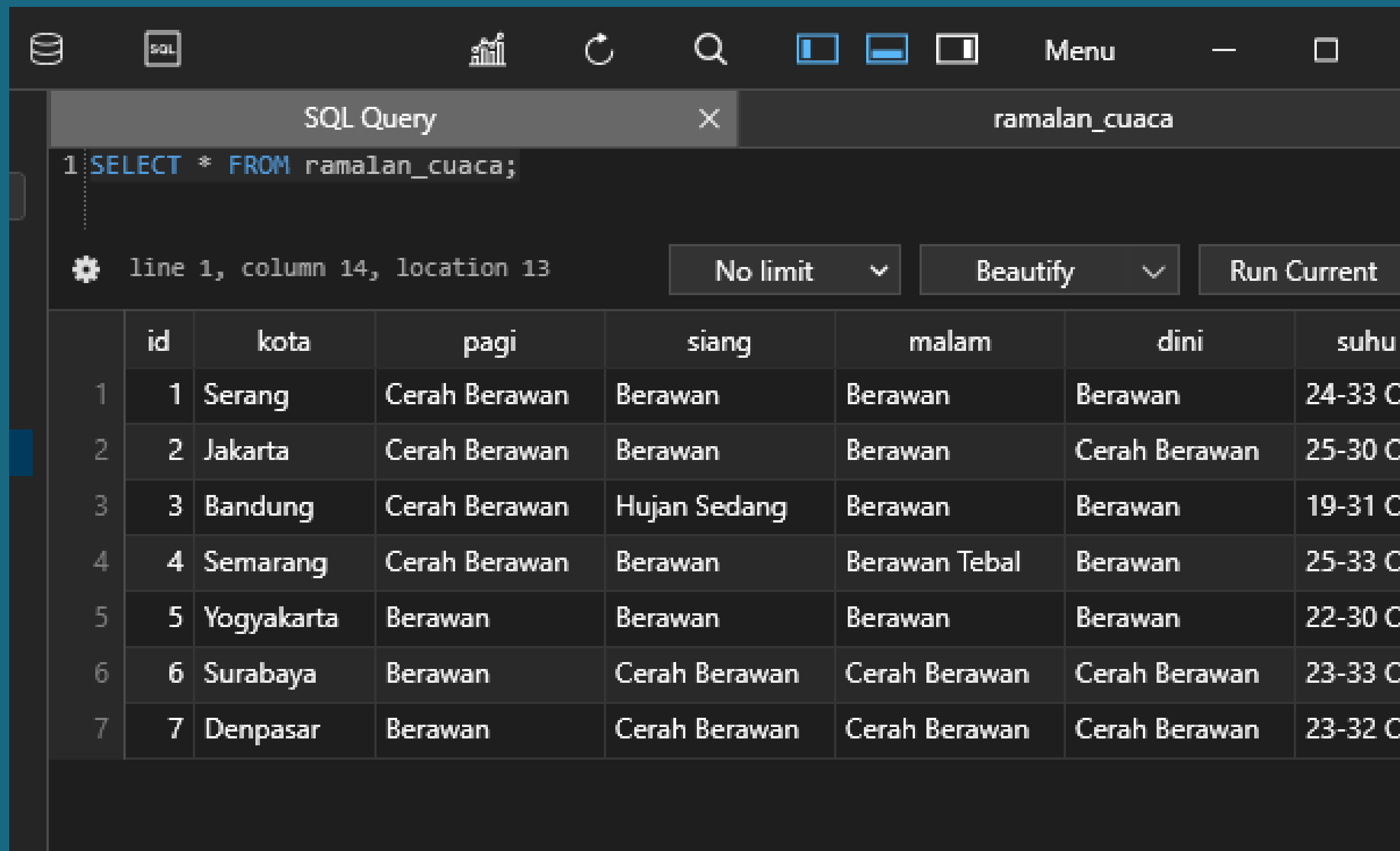
line 1, column 24, location 23 No limit Beautify

	Field	Type	Null	Key	Default	Extra
1	id	int(11)	NO	PRI	NULL	auto_increment
2	kota	varcha...	NO	EM...	NULL	EMPTY
3	pagi	varcha...	NO	EM...	NULL	EMPTY
4	siang	varcha...	NO	EM...	NULL	EMPTY
5	malam	varcha...	NO	EM...	NULL	EMPTY
6	dini	varcha...	NO	EM...	NULL	EMPTY
7	suhu	varcha...	NO	EM...	NULL	EMPTY

```
SQL Query X ramalan_cuaca
1 INSERT INTO ramalan_cuaca(kota, pagi, siang, malam, dini, suhu)
2 VALUES ('Serang', 'Cerah Berawan', 'Berawan', 'Berawan', 'Berawan', '24-33 C'),
3 ('Jakarta', 'Cerah Berawan', 'Berawan', 'Berawan', 'Cerah Berawan', '25-30 C'),
4 ('Bandung', 'Cerah Berawan', 'Hujan Sedang', 'Berawan', 'Berawan', '19-31 C'),
5 ('Semarang', 'Cerah Berawan', 'Berawan', 'Berawan Tebal', 'Berawan', '25-33 C'),
6 ('Yogyakarta', 'Berawan', 'Berawan', 'Berawan', 'Berawan', '22-30 C'),
7 ('Surabaya', 'Berawan', 'Cerah Berawan', 'Cerah Berawan', 'Cerah Berawan', '23-33 C'),
8 ('Denpasar', 'Berawan', 'Cerah Berawan', 'Cerah Berawan', 'Cerah Berawan', '23-32 C')
9
10
```



## Result table:



The screenshot shows a SQL query editor interface. At the top, there is a toolbar with icons for database, SQL, charts, refresh, search, and window management. Below the toolbar, the editor has a tab labeled 'SQL Query' and a dropdown menu showing 'ramalan\_cuaca'. The SQL query entered is 'SELECT \* FROM ramalan\_cuaca;'. Below the query, there is a status bar indicating 'line 1, column 14, location 13' and buttons for 'No limit', 'Beautify', and 'Run Current'. The result of the query is displayed as a table with 8 columns: 'id', 'kota', 'pagi', 'siang', 'malam', 'dini', and 'suhu'. The table contains 7 rows of data.

	id	kota	pagi	siang	malam	dini	suhu
1	1	Serang	Cerah Berawan	Berawan	Berawan	Berawan	24-33 C
2	2	Jakarta	Cerah Berawan	Berawan	Berawan	Cerah Berawan	25-30 C
3	3	Bandung	Cerah Berawan	Hujan Sedang	Berawan	Berawan	19-31 C
4	4	Semarang	Cerah Berawan	Berawan	Berawan Tebal	Berawan	25-33 C
5	5	Yogyakarta	Berawan	Berawan	Berawan	Berawan	22-30 C
6	6	Surabaya	Berawan	Cerah Berawan	Cerah Berawan	Cerah Berawan	23-33 C
7	7	Denpasar	Berawan	Cerah Berawan	Cerah Berawan	Cerah Berawan	23-32 C

# Exercise 2 Conclusions:

Unstructured data are data that aren't stored in a structured database format. In this exercise case, the data were all in text format like this:

Serang

- Pagi hari: cerah berawan
- Siang hari: berawan
- Malam hari: berawan
- Dini hari: berawan
- Suhu: 24-33 derajat Celcius

For converting unstructured data to structured data, the data can be change into database format like this:

		id	kota	pagi	siang	malam	dini	suhu
	1	1	Serang	Cerah Berawan	Berawan	Berawan	Berawan	24-33 C
	2	2	Jakarta	Cerah Berawan	Berawan	Berawan	Cerah Berawan	25-30 C
	3	3	Bandung	Cerah Berawan	Hujan Sedang	Berawan	Berawan	19-31 C
	4	4	Semarang	Cerah Berawan	Berawan	Berawan Tebal	Berawan	25-33 C
	5	5	Yogyakarta	Berawan	Berawan	Berawan	Berawan	22-30 C
	6	6	Surabaya	Berawan	Cerah Berawan	Cerah Berawan	Cerah Berawan	23-33 C
	7	7	Denpasar	Berawan	Cerah Berawan	Cerah Berawan	Cerah Berawan	23-32 C



# Exercise 3 – Chapter 1 Practice 3: Live Session

## Exercise Instructions:

1. Download the 2019 Transjakarta passenger data -link: Data Penumpang Transjakarta Tahun 2019 - Open Data Jakarta
2. Import csv data into BigQuery and answer this question:
  - How many total transjakarta passenger records were recorded in 2019?
  - To clarify the data entry, change the vehicle type = BRT to Bus Rapid Transit in the 2019 Transjakarta bus passenger data!



# Data source preview:



[Data](#) [Organisasi](#) [Topik](#) [Visualisasi](#) [Infografis](#) [Tentang](#)

[Home](#) / [Organisasi](#) / [Dinas Perhubungan](#) / [Data Penumpang Transjakarta Tahun 2019](#)



**Dinas Perhubungan**

Mewujudkan Jakarta Baru melalui penyediaan layanan transportasi yang handal, modern, dan berdaya saing internasional, dengan angkutan publik sebagai layanan utama

Kumpulan Data

### Data Penumpang Transjakarta Tahun 2019

Data Penumpang Transjakarta tahun 2019, disajikan dalam bentuk data bulanan

variabel data terdiri dari :

1. jenis : jenis angkutan
2. kode\_trayek : kode trayek
3. trayek : trayek
4. jumlah\_penumpang : jumlah Penumpang

Tag :

angkutan umum

angkutan umum masal

busway

dinas perhubungan

KPI 2019

kpi

penumpang

penumpang transjakarta

perhubungan

transjakarta

Metadata :

- Terakhir Diperbarui : 05 Mei 2020
- Dibuat : 01 Februari 2019
- Sumber : PT. Transportasi Jakarta
- Frekuensi Penerbitan : 1 Bulan Sekali
- Tahun : 2019
- Cakupan : Provinsi
- Penyajian : Bulanan
- Kontak : lodishub@gmail.com
- Rujukan :
- Lisensi : [Creative Commons Attribution](#)

## Data dan Sumber Data :



**Data Penumpang Bus Transjakarta Januari 2019**

Data Penumpang Bus Transjakarta Januari 2019

[Lihat Data](#) [JSON](#) [Unduh Data](#)



**Data Penumpang Bus Transjakarta Februari 2019**

Data Penumpang Bus Transjakarta Februari 2019

[Lihat Data](#) [JSON](#) [Unduh Data](#)



**Data Penumpang Bus Transjakarta Maret 2019**

Data Penumpang Bus Transjakarta Maret 2019

[Lihat Data](#) [JSON](#) [Unduh Data](#)



**Data Penumpang Bus Transjakarta April 2019**

Data Penumpang Bus Transjakarta April 2019

[Lihat Data](#) [JSON](#) [Unduh Data](#)



**Data Penumpang Bus Transjakarta Mei 2019**

Data Penumpang Bus Transjakarta Mei 2019

[Lihat Data](#) [JSON](#) [Unduh Data](#)



**Data Penumpang Bus Transjakarta Juni 2019**

Data Penumpang Bus Transjakarta Juni 2019

[Lihat Data](#) [JSON](#) [Unduh Data](#)



**Data Penumpang Bus Transjakarta Juli 2019**

Data Penumpang Bus Transjakarta Juli 2019

[Lihat Data](#) [JSON](#) [Unduh Data](#)



**Data Penumpang Bus Transjakarta Agustus 2019**

Data Penumpang Bus Transjakarta Agustus 2019

[Lihat Data](#) [JSON](#) [Unduh Data](#)



**Data Penumpang Bus Transjakarta September 2019**

Data Penumpang Bus Transjakarta September 2019

[Lihat Data](#) [JSON](#) [Unduh Data](#)



**Data Penumpang Bus Transjakarta Oktober 2019**

Data Penumpang Bus Transjakarta Oktober Tahun 2019

[Lihat Data](#) [JSON](#) [Unduh Data](#)



**Data Penumpang Bus Transjakarta November 2019**

Data Penumpang Bus Transjakarta November 2019

[Lihat Data](#) [JSON](#) [Unduh Data](#)



**Data Penumpang Bus Transjakarta Desember 2019**

Data Penumpang Bus Transjakarta Desember 2019

[Lihat Data](#) [JSON](#) [Unduh Data](#)

From webpage, it can be seen there are totsl 12 file data csv separated by month

Before jump to the query tasks, the dataset need treatment because it not clean yet as it shows in the preview, that BigQuery failed to detect field name. In order to make the dataset ready to use, here to do:

- Cleansing jumlah\_penumpang value
- Change the value into INT type data value

data-penumpang-bus-tra... 019

data-penumpa... QUERY

SCHEMADETAILSPREVIEWLINE

Filter Enter property name or value

<input type="checkbox"/>	Field name	Type	Mode
<input type="checkbox"/>	<a href="#">string_field_0</a>	STRING	NULLABLE
<input type="checkbox"/>	<a href="#">string_field_1</a>	STRING	NULLABLE
<input type="checkbox"/>	<a href="#">string_field_2</a>	STRING	NULLABLE
<input type="checkbox"/>	<a href="#">string_field_3</a>	STRING	NULLABLE

data-penumpang-bus-tra... 019

data-penumpa... QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMADDETAILSPREVIEWLINEAGE

Row	string_field_0	string_field_1	string_field_2	string_field_3
1	BRT	1	KORIDOR 1	2.223.257
2	BRT	2	KORIDOR 2	781.605
3	BRT	3	KORIDOR 3	919.2
4	BRT	4	KORIDOR 4	669.417

Cleaning jumlah\_penumpang value that is erasing dot (.) between value because if it not be erased, when program trying to change the value data type into INT it will failed. After the dot erased, the value in jumlah\_penumpang column is still in OBJECT/STRING data type, so it need to change to INT.

```
jmp = ds['jumlah_penumpang']
```

```
jmp[:10]
```

0	2.223.257
1	781.605
2	919.2
3	669.417
4	890.939
5	976.411
6	872.467
7	911.731
8	1.408.962
9	737.123

Name: jumlah\_penumpang, dtype: object

```
jmpP = jmp.str.replace('.', '')
```

```
jmpP[:10]
```

0	2223257
1	781605
2	9192
3	669417
4	890939
5	976411
6	872467
7	911731
8	1408962
9	737123

Name: jumlah\_penumpang, dtype: object

After changing the jumlah\_penumpang value into INT, it can be seen that jumlah\_penumpang data type is now INT.

```
#change data to int
new_jmp = [int(i) for i in jmp]
```

```
new_jmp[:5]
```

```
[2223257, 781605, 9192, 669417, 890939]
```

```
ds.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 160 entries, 0 to 159
```

```
Data columns (total 4 columns):
```

#	Column	Non-Null Count	Dtype
0	jenis	160 non-null	object
1	kode_trayek	128 non-null	object
2	trayek	160 non-null	object
3	jumlah_penumpang	160 non-null	int64

januari				
SCHEMA				
Filter Enter property name or value				
<input type="checkbox"/>	Field name	Type	Mode	
<input type="checkbox"/>	<a href="#">jenis</a>	STRING	NULLABLE	
<input type="checkbox"/>	<a href="#">kode_trayek</a>	STRING	NULLABLE	
<input type="checkbox"/>	<a href="#">trayek</a>	STRING	NULLABLE	
<input type="checkbox"/>	<a href="#">jumlah_penumpang</a>	INTEGER	NULLABLE	

Now, the datasets are ready to use and BigQuery also not failed to detect field name like before

## Query result:

- How many total transjakarta passenger records were recorded in 2019?

```
1 select sum(total) from (SELECT sum(a.jumlah_penumpang) as total
2 FROM `soy-codex-379914.penumpang.januari` a
3 UNION ALL
4 SELECT sum(b.jumlah_penumpang) as total
5 FROM `soy-codex-379914.penumpang.februari` b
6 UNION ALL
7 SELECT sum(c.jumlah_penumpang) as total
8 FROM `soy-codex-379914.penumpang.maret` c
9 UNION ALL
10 SELECT sum(d.jumlah_penumpang) as total
11 FROM `soy-codex-379914.penumpang.april` d
12 UNION ALL
13 SELECT sum(e.jumlah_penumpang) as total
14 FROM `soy-codex-379914.penumpang.mey` e
15 UNION ALL
16 SELECT sum(f.jumlah_penumpang) as total
17 FROM `soy-codex-379914.penumpang.juni` f
18 UNION ALL
19 SELECT sum(g.jumlah_penumpang) as total
20 FROM `soy-codex-379914.penumpang.juli` g
21 UNION ALL
22 SELECT sum(h.jumlah_penumpang) as total
23 FROM `soy-codex-379914.penumpang.agustus` h
24 UNION ALL
25 SELECT sum(i.jumlah_penumpang) as total
26 FROM `soy-codex-379914.penumpang.september` i
```

Query results			
JOB INFORMATION		RESULTS	JSON
Row	Total_Passenger_2019		
1	252910421		



## Exercise Instructions:

- To clarify the data entry, change the vehicle type = BRT to Bus Rapid Transit in the 2019 Transjakarta bus passenger data!

```
1 select nj as jenis_lama, replace(nj, 'BRT', 'Bus Rapid Transfer') as jenis_baru
2 from ((SELECT a.jenis as nj
3 FROM `soy-codex-379914.penumpang.januari` a
4 UNION ALL
5 SELECT b.jenis as nj
6 FROM `soy-codex-379914.penumpang.februari` b
7 UNION ALL
8 SELECT c.jenis as nj
9 FROM `soy-codex-379914.penumpang.maret` c
10 UNION ALL
11 SELECT d.jenis as nj
12 FROM `soy-codex-379914.penumpang.april` d
13 UNION ALL
14 SELECT e.jenis as nj
15 FROM `soy-codex-379914.penumpang.mey` e
16 UNION ALL
17 SELECT f.jenis as nj
18 FROM `soy-codex-379914.penumpang.juni` f
19 UNION ALL
20 SELECT g.jenis as nj
21 FROM `soy-codex-379914.penumpang.juli` g
22 UNION ALL
23 SELECT h.jenis as nj
```

Query results				
JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	jenis_lama	jenis_baru		
1	BRT	Bus Rapid Transfer		
2	BRT	Bus Rapid Transfer		
3	BRT	Bus Rapid Transfer		
4	BRT	Bus Rapid Transfer		



## Exercise 3 Conclusions:

1. In this exercise i learn how to cleaning dataset before ready to use.
2. Because there are 12 table (12 data set) in total, so rather than merger all of them when i did cleaning data, i try to challenge my self to used UNION to merge these all and sub-query to solve that and it succesful.



# Thank You

Contact me:



<https://www.linkedin.com/in/sitifatimatuzzahro/>



[sitifatimoo10@gmail.com](mailto:sitifatimoo10@gmail.com)