

Date-August 26,2022

Name-Divya Kirtikumar Patel

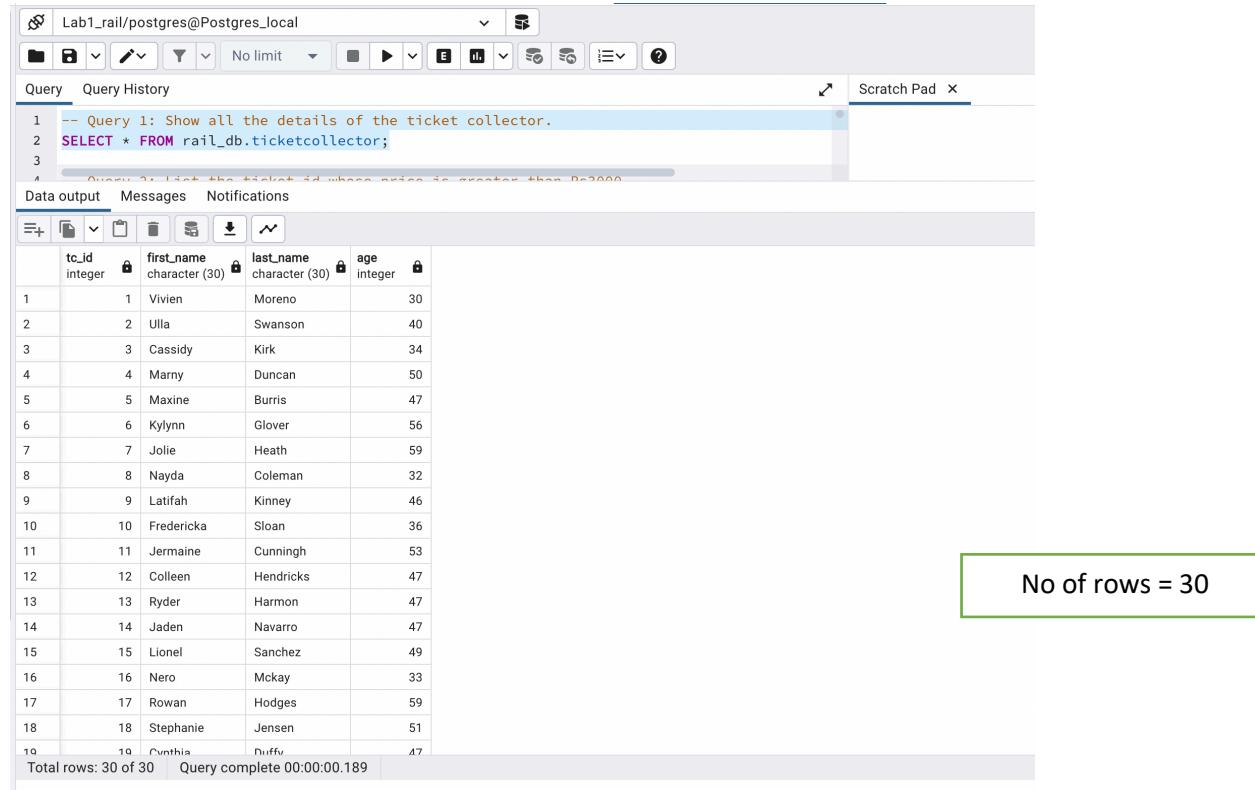
Student ID-202001420

Lab Group- 6      Section - 8

**Query 1: Show all the details of the ticket collector.**

SQL Query –

SELECT \* FROM rail\_db.ticketcollector;



```
-- Query 1: Show all the details of the ticket collector.
SELECT * FROM rail_db.ticketcollector;
```

tc_id	first_name	last_name	age
1	Vivien	Moreno	30
2	Ulla	Swanson	40
3	Cassidy	Kirk	34
4	Marny	Duncan	50
5	Maxine	Burris	47
6	Kylynn	Glover	56
7	Jolie	Heath	59
8	Nayda	Coleman	32
9	Latifah	Kinney	46
10	Fredericka	Sloan	36
11	Jermaine	Cunningh	53
12	Colleen	Hendricks	47
13	Ryder	Harmon	47
14	Jaden	Navarro	47
15	Lionel	Sanchez	49
16	Nero	Mckay	33
17	Rowan	Hodges	59
18	Stephanie	Jensen	51
19	Cynthia	Duffy	47

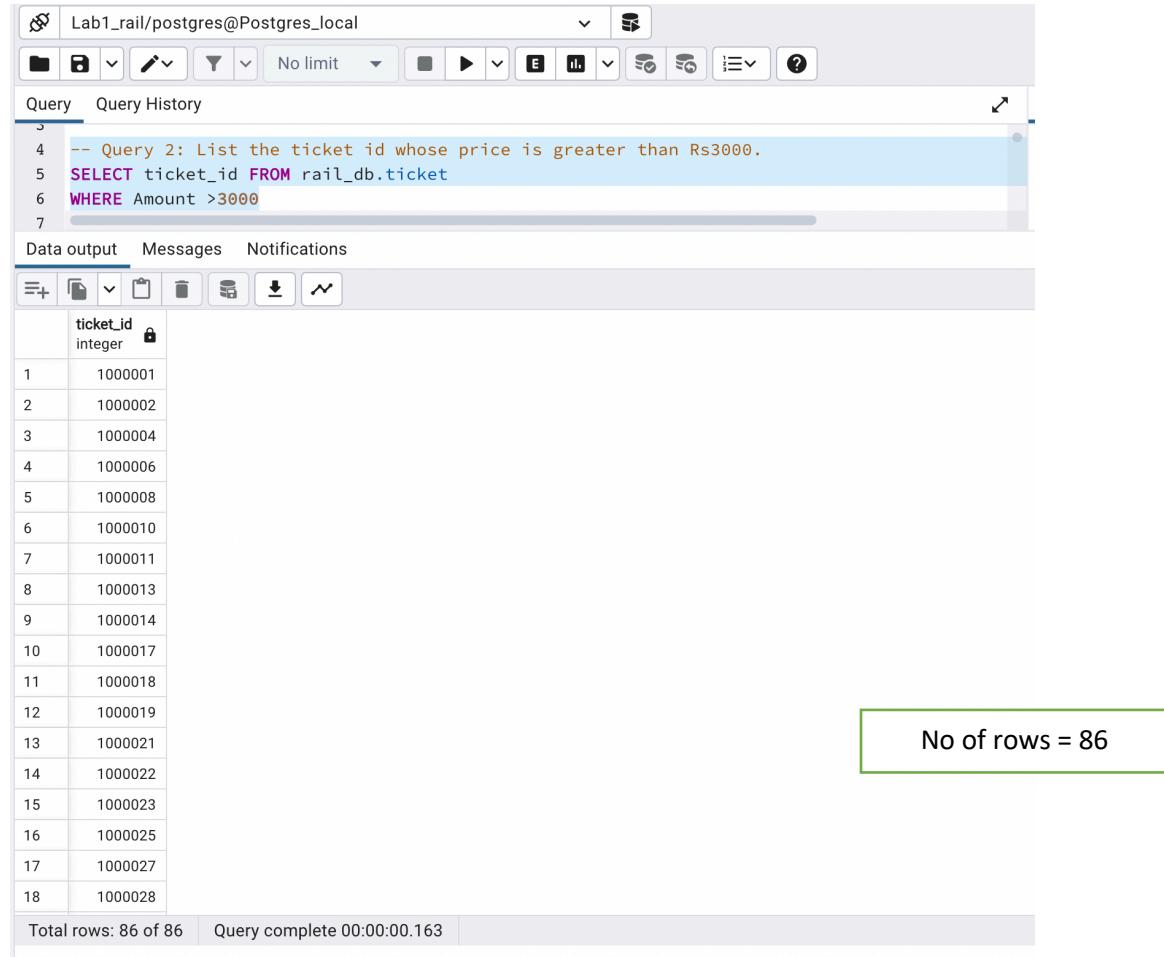
No of rows = 30

**Query 2: List the ticket id whose price is greater than Rs3000.**

SQL Query -

SELECT ticket\_id FROM rail\_db.ticket

WHERE Amount &gt;3000



```
-- Query 2: List the ticket id whose price is greater than Rs3000.
SELECT ticket_id FROM rail_db.ticket
WHERE Amount >3000;
```

ticket_id
1000001
1000002
1000004
1000006
1000008
1000010
1000011
1000013
1000014
1000017
1000018
1000019
1000021
1000022
1000023
1000025
1000027
1000028

No of rows = 86

### Query 3: Show all the food items that are not available.

Ans. :  
SELECT item FROM rail\_db.pantry  
where availability = 'No';

```
7
8 -- Query 3: Show all the food items that are not available.
9 SELECT item FROM rail_db.pantry
10 where availability = 'No';
11
```

item
sosa
pizza
dal chawal
dosa
dhokla
thali

No of rows = 6

### Query 4: Count the number of female passengers.

Ans. :  
SELECT count(gender) FROM rail\_db.passengerdetails  
WHERE gender = 'female';

```
12 -- Query 4: Count the number of female passengers.
13 SELECT count(gender) FROM rail_db.passengerdetails as count
14 WHERE gender = 'female';
15
16 -- Query 5: Find out the smallest age of passenger travelling.
```

count
134

No of rows = 1

### Query 5: Find out the smallest age of passenger travelling.

Ans. :  
SELECT MIN(age) as minimum FROM rail\_db.passengerdetails;

```
14 WHERE gender = 'female',
15
16 -- Query 5: Find out the smallest age of passenger travelling.
17 SELECT MIN(age) as minimum FROM rail_db.passengerdetails;
18
19 -- Query 6: Find out the eldest tc among all the ticket collectors.
```

minimum
1

No of rows = 1

### Query 6: Find out the eldest tc among all the ticket collectors.

Ans. :  
SELECT max(age) as eldest FROM rail\_db.ticketcollector;

```
1 -- Query 6: Find out the eldest tc among all the ticket collectors.
2 SELECT * FROM rail_db.ticketcollector where (Age = (SELECT MAX(Age) FROM rail_db.ticketcollector));
```

tc_id	first_name	last_name	age
7	Jolie	Heath	59
17	Rowan	Hodges	59
22	Charles	Rhodes	59

No of rows = 3

### Query 7: Write a query to print all the food items being served on the train.

Ans. : SELECT DISTINCT item from rail\_db.pantry;

The screenshot shows the pgAdmin interface with the following details:

- Connection:** Lab1\_rail/postgres@Postgres\_local
- Toolbar:** Includes icons for file operations, search, and various database management functions.
- Query Editor:** Shows the query: `-- Query 7: Write a query to print all the food items being served on the train.  
SELECT DISTINCT item from rail_db.pantry;`
- Data Output:** A table with 10 rows labeled 1 through 10, each containing a food item name. The columns are labeled "item" and "character (30)".

	item character (30)
1	rajma chawal
2	upma
3	dhokla
4	sandwich
5	burger
6	thali
7	pizza
8	sosa
9	dosa
10	dal chawal
- Message:** A green box in the bottom right corner states "No of rows = 10".

### Query 8: Print the name of trains where numb of seats available is greater than 5.

Ans. : SELECT train\_name FROM rail\_db.train

WHERE seats\_available>5;

The screenshot shows the pgAdmin interface with the following details:

- Connection:** Lab1\_rail/postgres@Postgres\_local
- Toolbar:** Includes icons for file operations, search, and various database management functions.
- Query Editor:** Shows the query: `-- Query 8: Print the name of trains where numb of seats available is greater than 5  
SELECT train_name FROM rail_db.train  
WHERE seats_available>5;`
- Data Output:** A table with 77 rows labeled 1 through 77, each containing a train name. The columns are labeled "train\_name" and "character (50)".

	train_name character (50)
1	Suspendisse ...
2	di luctus lobor...
3	fermentum m...
4	faucibus orci
5	dui. Cras pelle...
6	aliquet
7	Sed congue elit
8	imperdiet erat
9	sit et
10	ultricies sem ...
11	nunc ac
12	accumsan co...
13	aliqui eros
14	pede
15	egestas bland...
16	nulla vulputat...
17	in lobortis tell...
18	...nd...
- Message:** A green box in the bottom right corner states "No of rows = 77".
- Footer:** Shows "Total rows: 77 of 77" and "Query complete 00:00:00.094".

### Query 9: Print the number of rows present in the passenger table.

Ans. : SELECT count(passenger\_id) FROM rail\_db.passengerdetails;

The screenshot shows the pgAdmin interface with the following details:

- Connection:** Lab1\_rail/postgres@Postgres\_local
- Toolbar:** Includes icons for file operations, search, filters, and database management.
- Query Bar:** Shows the query text:

```
-- Query 9: Print the number of rows present in the passenger table
SELECT count(passenger_id) FROM rail_db.passengerdetails;
```
- Data Output:** A table with one row:

count	bigint
1	250
- Note:** A green box highlights the value "250" with the text "No of rows = 1".

### Query 10: Find the ticket id of the costliest ticket.

Ans. : SELECT ticket\_id FROM rail\_db.ticket

where amount = (select(max(amount)) from rail\_db.ticket );

The screenshot shows the pgAdmin interface with the following details:

- Connection:** Lab1\_rail/postgres@Postgres\_local
- Toolbar:** Includes icons for file operations, search, filters, and database management.
- Query Bar:** Shows the query text:

```
-- Query 10: Find the ticket id of the costliest ticket.
SELECT ticket_id FROM rail_db.ticket
where amount = (select(max(amount)) from rail_db.ticket );
```
- Data Output:** A table with one row:

ticket_id	integer
1	1000087
- Note:** A green box highlights the value "1000087" with the text "No of rows = 1".

### Query 11: Show the second highest priced ticket\_id.

Ans. :

```
SELECT ticket_id FROM rail_db.ticket
```

```
ORDER BY amount DESC
```

```
LIMIT 1
```

```
OFFSET 1;
```

The screenshot shows the pgAdmin interface with the following details:

- Connection:** Lab1\_rail/postgres@Postgres\_local
- Toolbar:** Includes icons for file operations, search, filters, and database management.
- Query Bar:** Shows the query text:

```
-- Query 11: Show the second highest priced ticket_id
SELECT ticket_id FROM rail_db.ticket
ORDER BY amount DESC
LIMIT 1
OFFSET 1;
```
- Data Output:** A table with one row:

ticket_id	integer
1	1000039
- Note:** A green box highlights the value "1000039" with the text "No of rows = 1".

### Query 12: Print the name of passengers whose ticket is confirmed.

Ans. : SELECT name FROM rail\_db.passengerdetails

WHERE reservation\_status = 'confirm';

The screenshot shows the DBeaver interface with the following details:

- Toolbar:** Includes icons for file operations, search, and various database functions.
- Query Bar:** Shows the query number (41) and the SQL code:

```
-- Query 12: Print the name of passengers whose ticket is confirmed
SELECT name FROM rail_db.passengerdetails
WHERE reservation_status = 'confirm';
```
- Data Output:** A table with the column "name" (character(50)). The data consists of 17 rows numbered 1 to 17, listing names such as Serena, Philip, Levi, Ina, Beau, Giselle, Ralph, Sade, Grady, Zena, Rama, Orlando, Elaine, Kim, Reuben, Samson, and Tanisha.
- Message Box:** A green-bordered box contains the text "No of rows = 122".
- Status Bar:** Shows "Total rows: 122 of 122" and "Query complete 00:00:00.151".

### Query 13: Print all the names of stations.

Ans. : SELECT name FROM rail\_db.station

The screenshot shows the DBeaver interface with the following details:

- Toolbar:** Includes icons for file operations, search, and various database functions.
- Query Bar:** Shows the query number (46) and the SQL code:

```
-- Query 13: Print all the names of stations.
SELECT name FROM rail_db.station
```
- Data Output:** A table with the column "name" (character(50)). The data consists of 17 rows numbered 1 to 17, listing station codes such as S4Z 2X3, L3X 6S5, A4M 3T4, P0Y 3J5, I5X 8I5, Q7Y 6J9, Q4G 1K5, F6G 2F1, G6S 7E7, Y6W 2R8, E9D 2R7, I7D 4S7, D2L 3R2, I7U 105, P2D 8B8, I7U 5J4, and E6K 2B1.
- Message Box:** A green-bordered box contains the text "No of rows = 100".
- Status Bar:** Shows "Total rows: 100 of 100" and "Query complete 00:00:00.168".

**Query 14: Print the name of all the ticket collectors who are below the age of 40.**

**Ans. :** SELECT first\_name,last\_name FROM rail\_db.ticketcollector

WHERE age<40

The screenshot shows a PostgreSQL query editor interface. The top bar includes a connection dropdown (Lab1\_rail/postgres@Postgres\_local), a toolbar with various icons, and a status bar indicating 'No limit' and other connection details. The main area has tabs for 'Query' and 'Query History'. The query text is as follows:

```
48  SELECT name FROM rail_db.station
49
50  -- Query 14: Print the name of all the ticket collectors who are below the age of 40.
51  SELECT first_name,last_name FROM rail_db.ticketcollector
52  WHERE age<40
53
```

The results table has columns 'first\_name' and 'last\_name'. The data is:

	first_name	last_name
1	Vivien	Moreno
2	Cassidy	Kirk
3	Nayda	Coleman
4	Fredericka	Sloan
5	Nero	Mckay
6	Steven	Swanson
7	Beck	Andrews

A green box highlights the text 'No of rows = 7'.

**Query 15: Write a query to print the number of foods that were available & their mode of payment was online.**

**Ans. :** SELECT \* FROM rail\_db.pantry

WHERE availability='Yes' and payment\_mode='online'

The screenshot shows a PostgreSQL query editor interface. The top bar includes a connection dropdown (Lab1\_rail/postgres@Postgres\_local), a toolbar with various icons, and a status bar indicating 'No limit' and other connection details. The main area has tabs for 'Query' and 'Query History'. The query text is as follows:

```
178  -- Query 15: Write a query to print the number of foods that were available & their mode of payment was online.
179  SELECT COUNT(*) FROM pantry
180  WHERE availability='Yes' and payment_mode='online'
181
```

The results table has a single column 'count'. The data is:

	count
1	2

A green box highlights the text 'No of rows = 1'.