Table definitions are as follows:

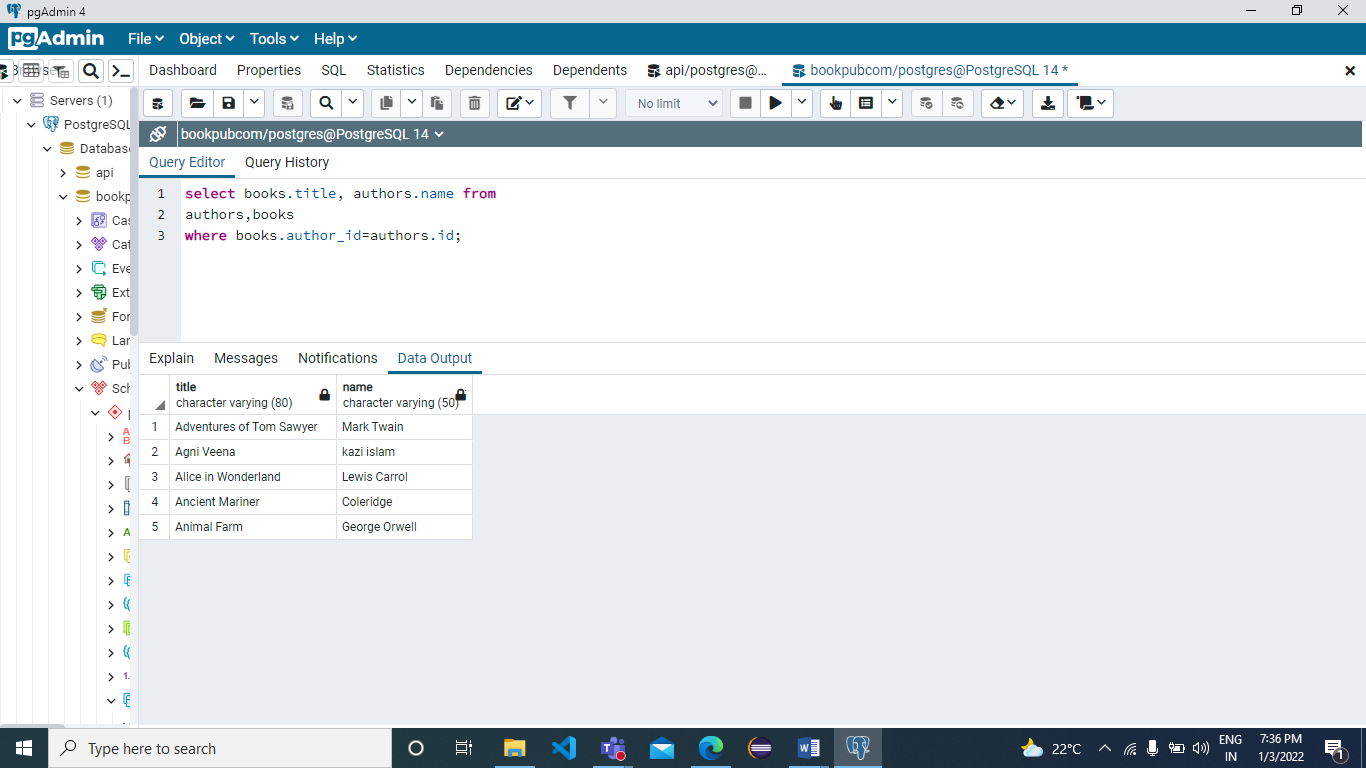
|  |  |
| --- | --- |
| Authors | id SERIAL, name varchar(50) |
| Books | Id SERIAL, title varchar(100), author\_id |
| Reviews | Id SERIAL, rating INTEGER, reviewer\_id INTEGER, book\_is  Note: reviewer\_id is nothing but authors id |

Write a query that will join together these two tables.

1. For each book, print the title of the book and name of the author.

Ans:

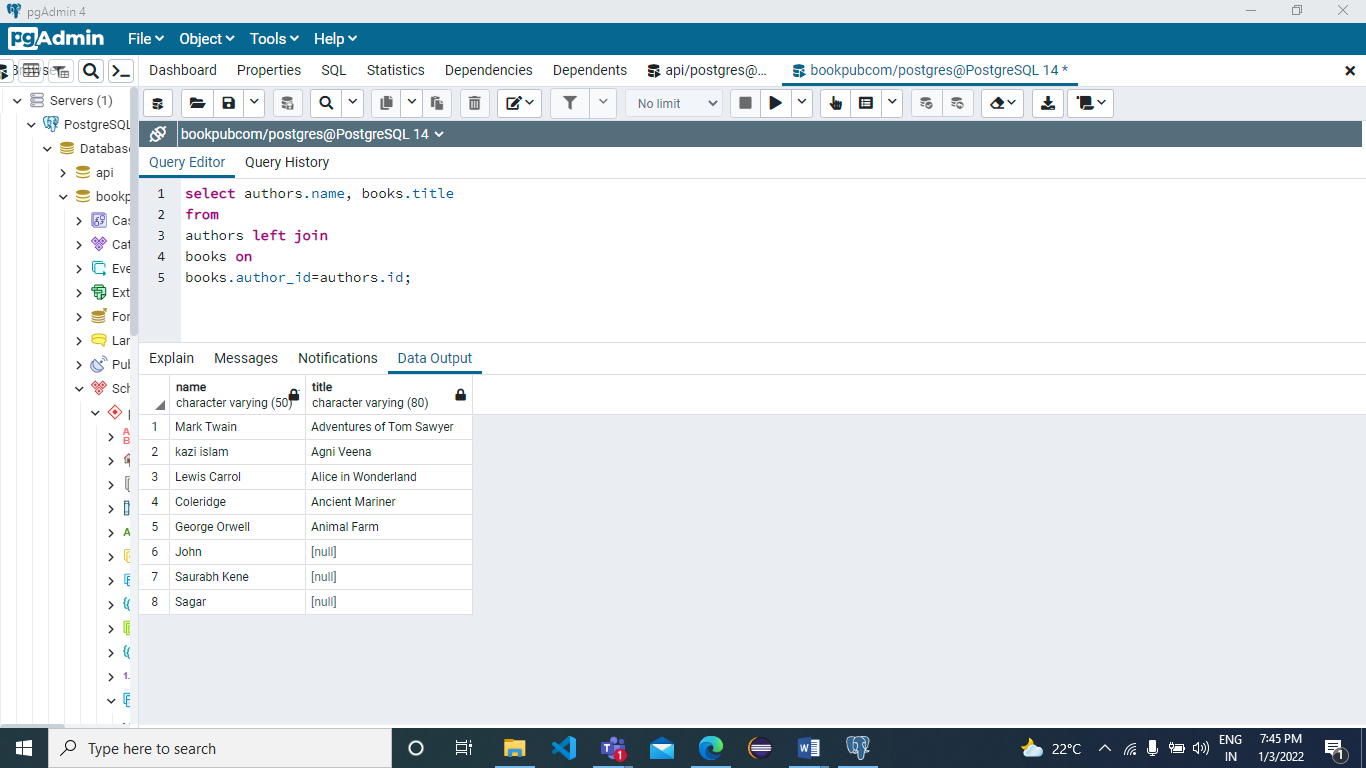
select books.title,authors.name from authors, books where books.author\_id=authors.id;



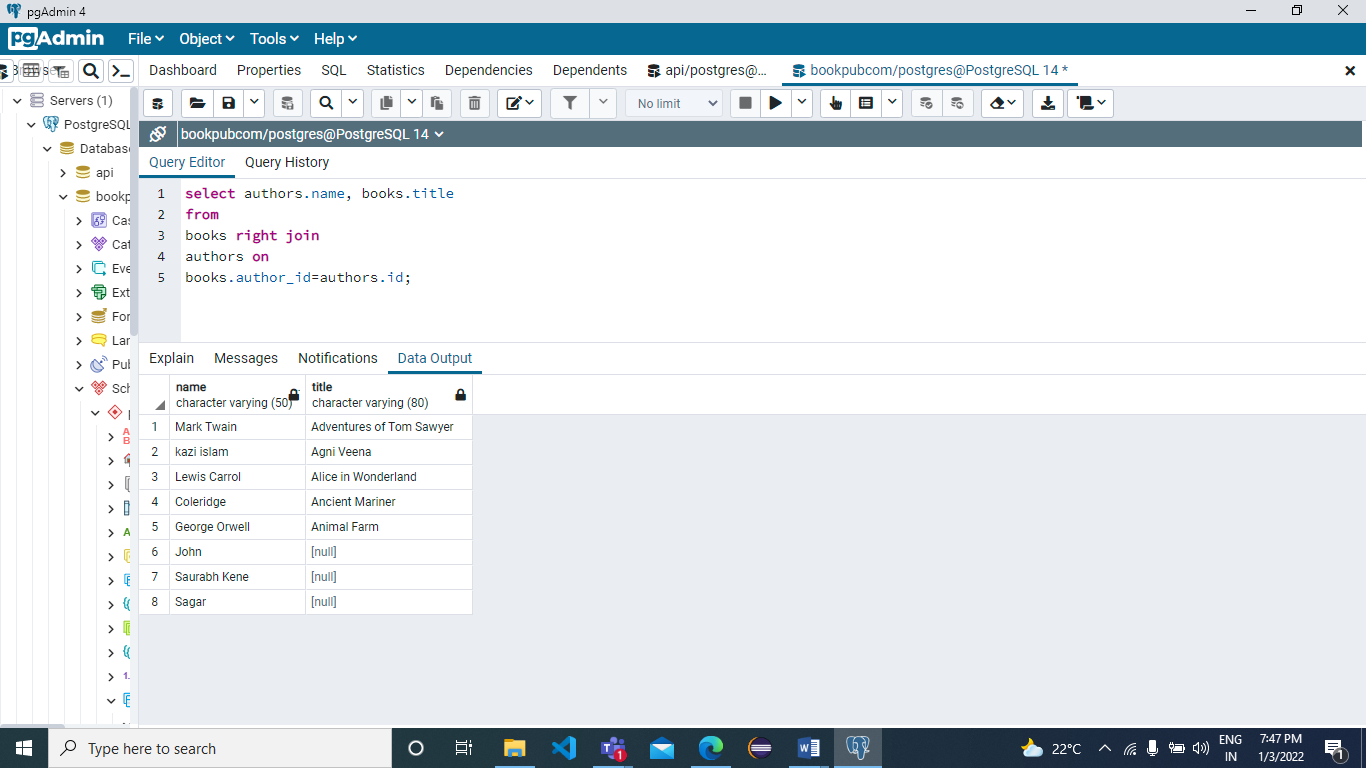
1. Return title of each book, along with the name of the author. All authors should be included, even if they don’t have a book associated with them.

**(You need to provide two possible solutions using Joins)**

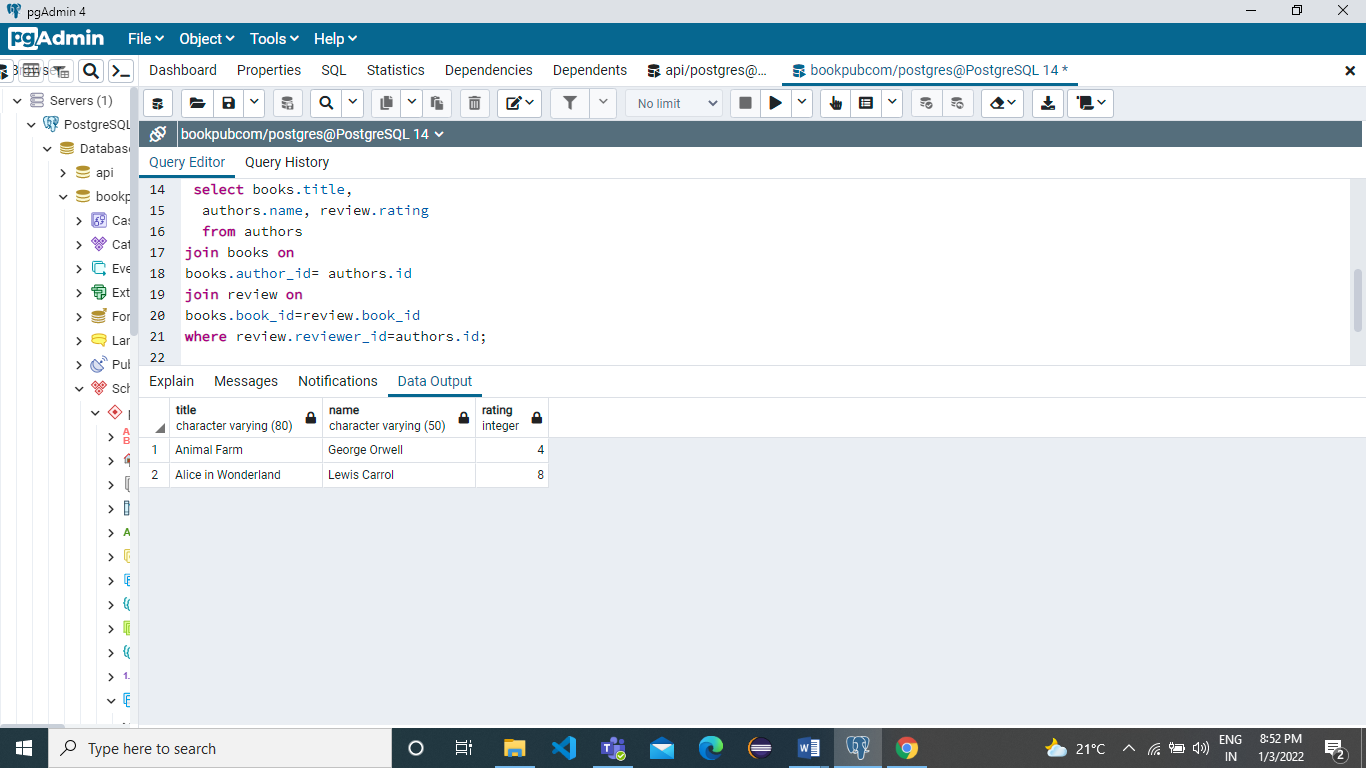
**Ans:- LEFT JOIN USED**



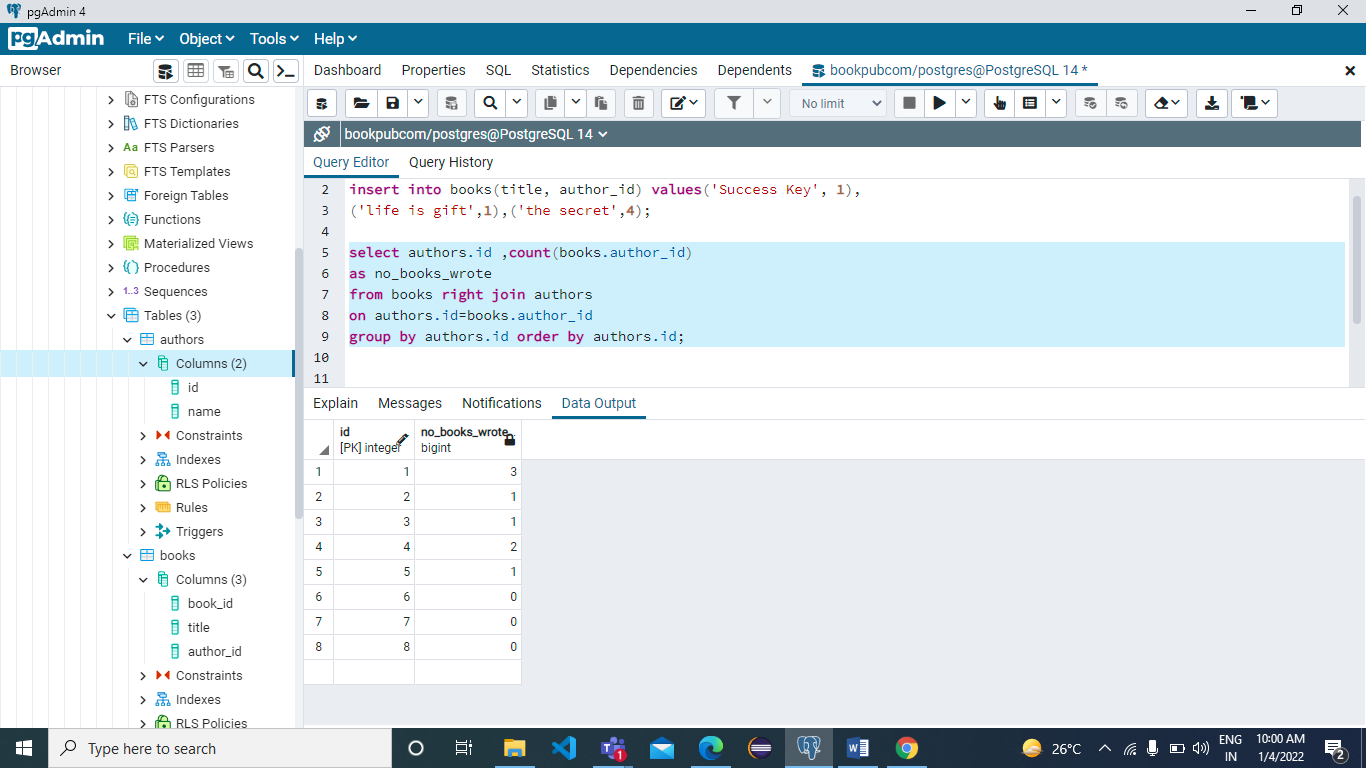
**ANS:- RIGHT JOIN USED**



1. Return title of each book, along with the name of the author, and the rating of a review. Only show rows where the author of the book is also the author of the review.



1. Prints an author’s id and the number of books they have authored.

**Ans:** select authors.id ,count(books.author\_id) as no\_books\_wrote from books right join authorson authors.id=books.author\_id group by authors.id order by authors.id;

1. Print an author’s name and the number of books they have authored.

**Ans: select authors.id , authors.name,**

**count(books.author\_id) as books\_wrote**

**from authors left join books**

**on authors.id = books.author\_id**

**group by authors.id order by authors.id;**

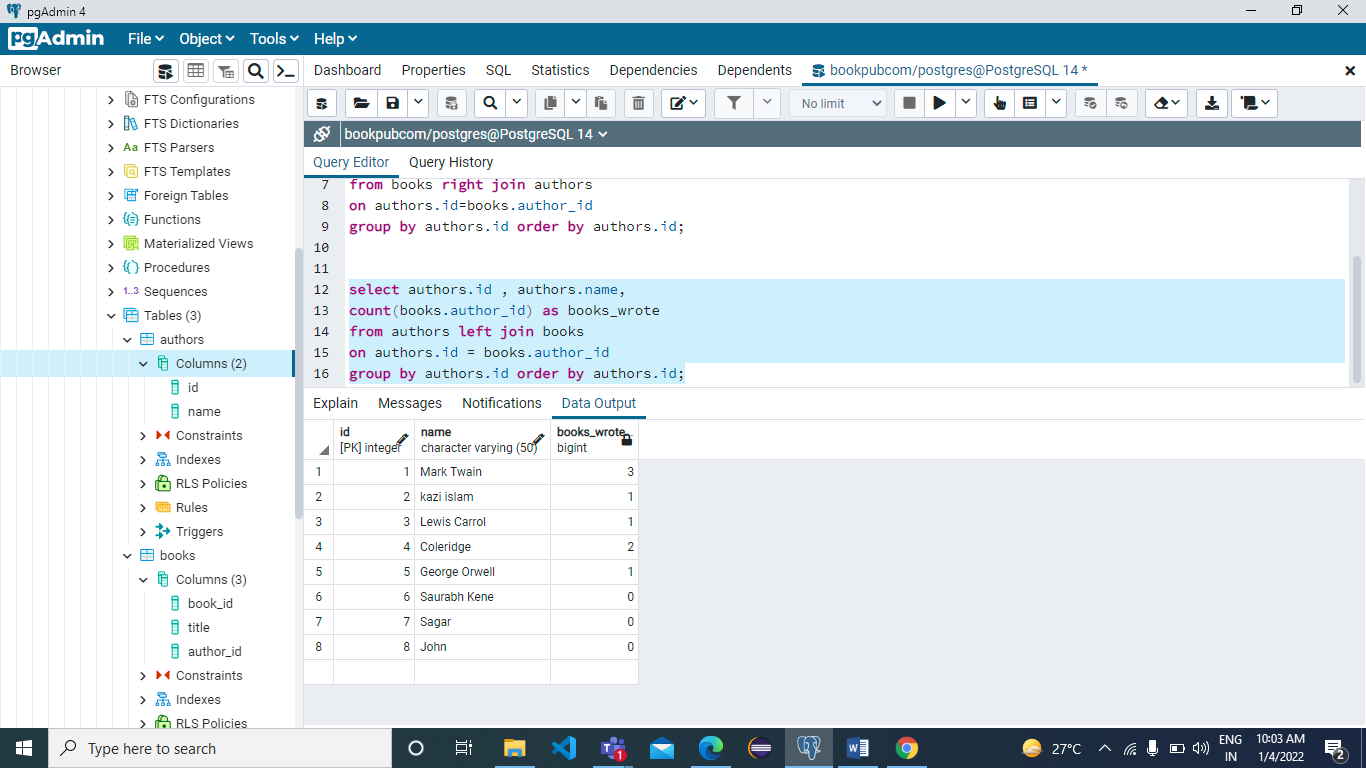


Table details are as follows:

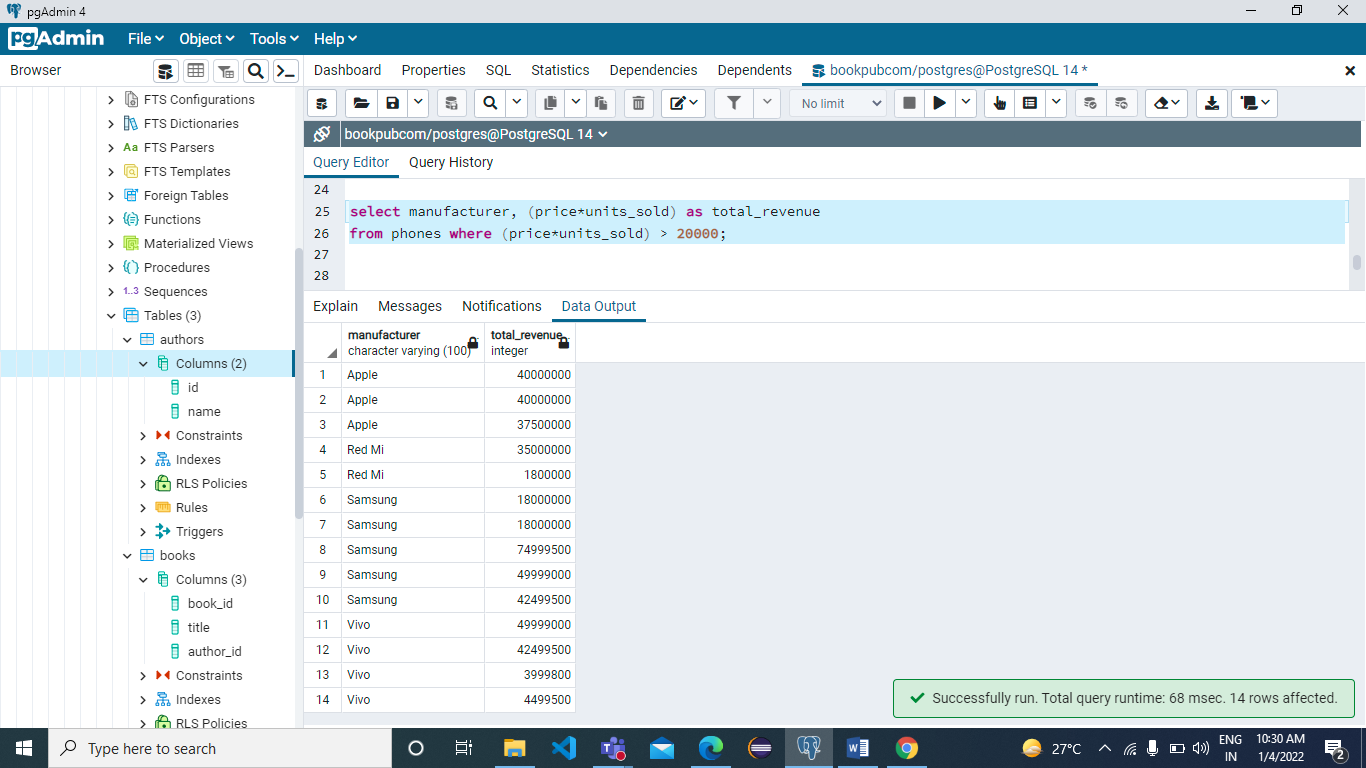
|  |  |
| --- | --- |
| Phones | name Varchar(50), manufacturer varchar(100), price INTEGER, units\_sold INTEGER |

1. Write a query that prints the name of manufacturers and total revenue (price \* units\_sold) for all phones. Only print the manufacturers who have revenue greater than 2,000,00 for all phones they sold.

ANS:

select manufacturer, (price\*units\_sold) as total\_revenue

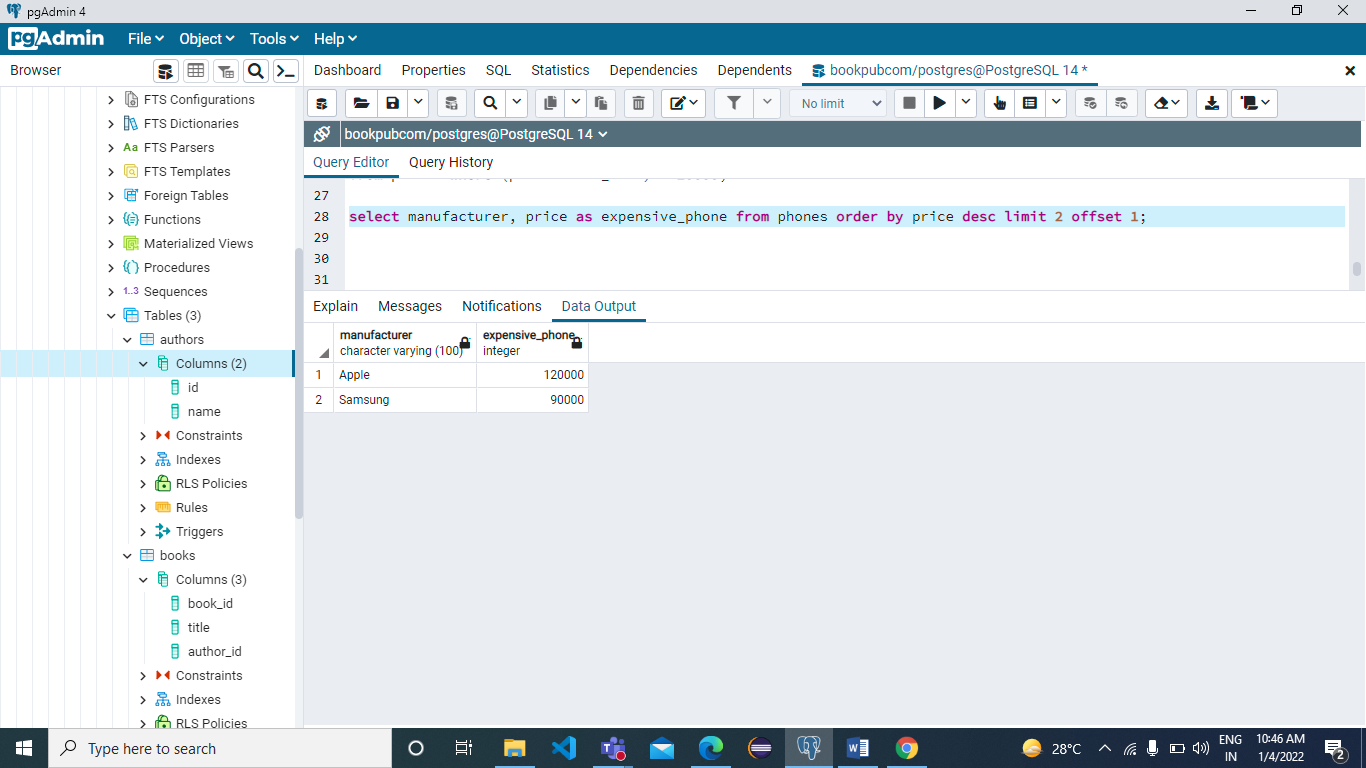
from phones where (price\*units\_sold) > 20000;



1. Write a query that shows the names of only the second and third most expensive phones.

Ans:

select manufacturer, price as expensive\_phone from phones order by price desc limit 2 offset 1;



1. Write a query that will print the manufacturer of phones where the phone’s price is less than 170. Also print all manufaturer that have created more than two phones.

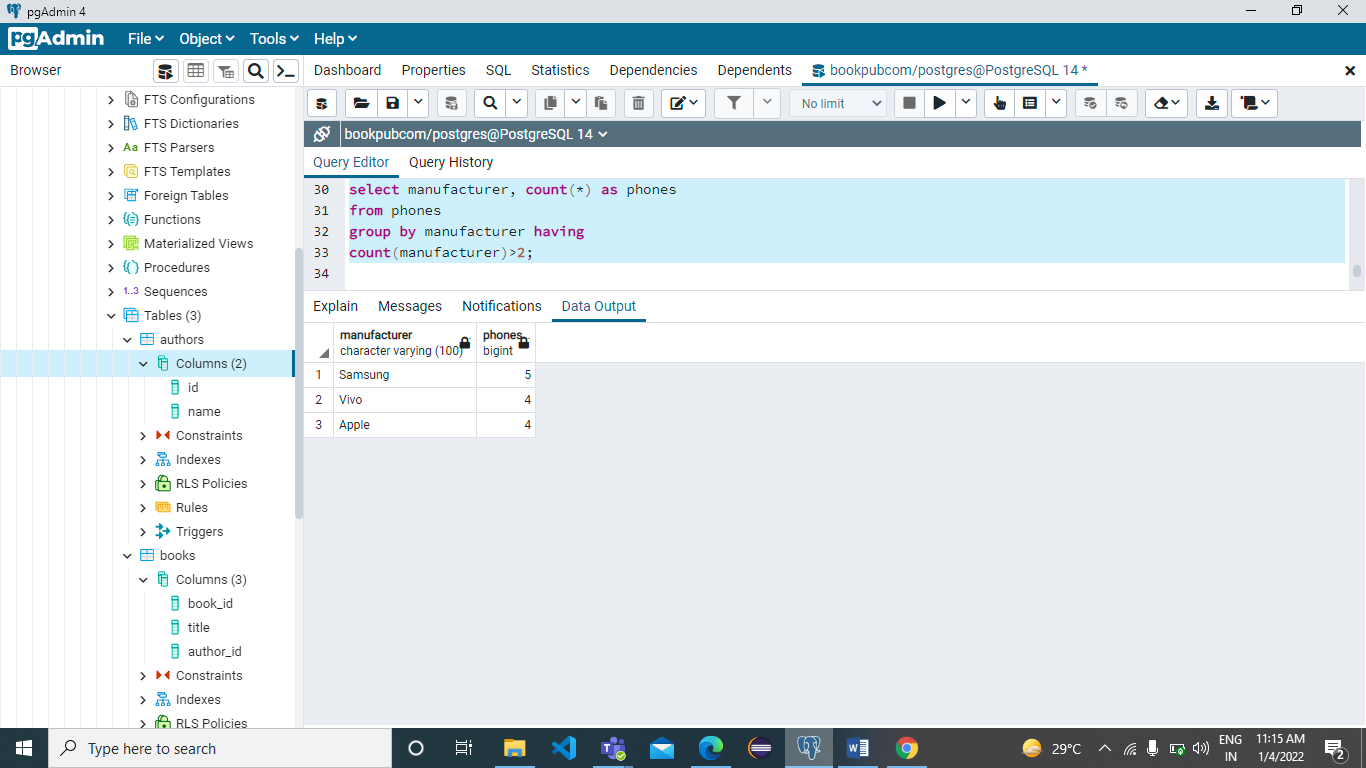
Ans:

select manufacturer, count(\*) as phones

from phones

group by manufacturer having

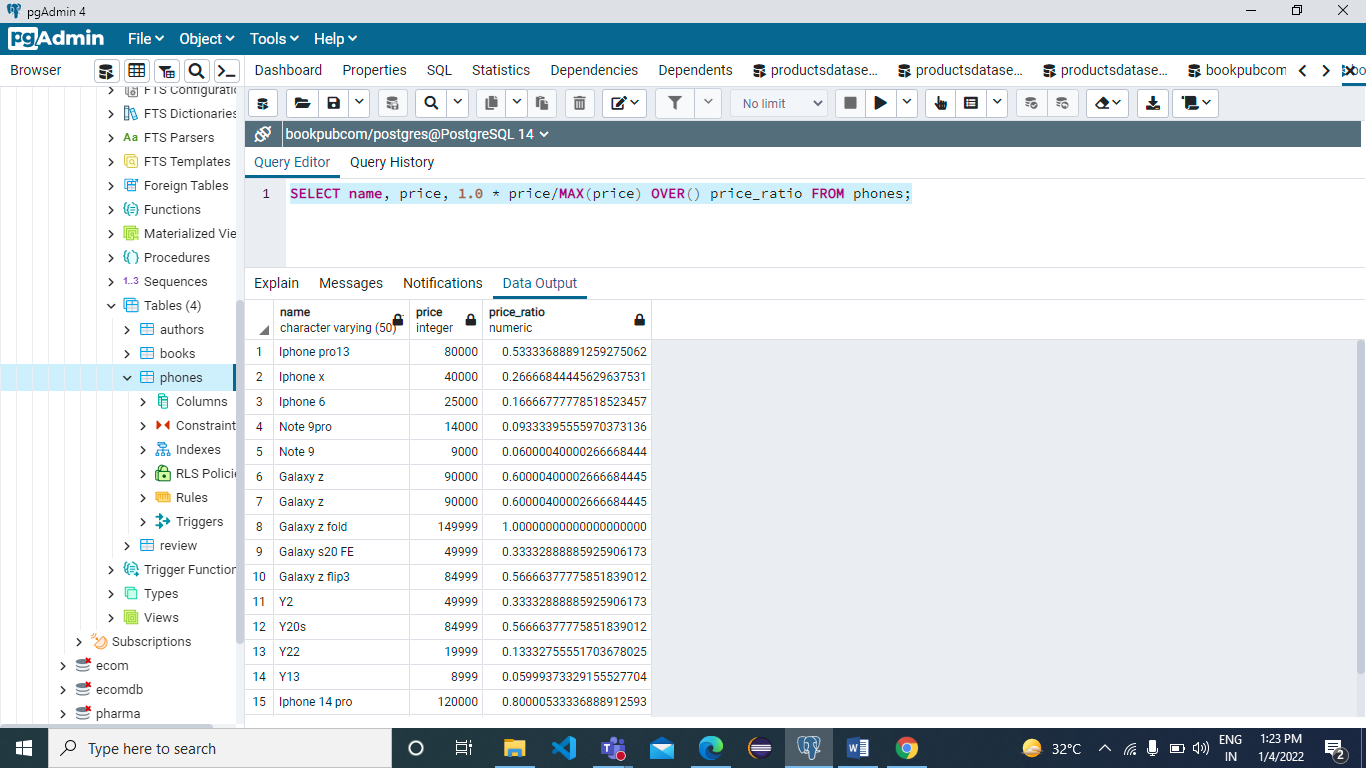
count(manufacturer)>2;



1. Write a query that prints the name and price for each phone. In addition, print out the ratio of the phones price against max of all prices. Rename this third column to price\_ratio.

Ans:

select name, price, 1.0 \* price/max(price) over() price\_ratio from phones;

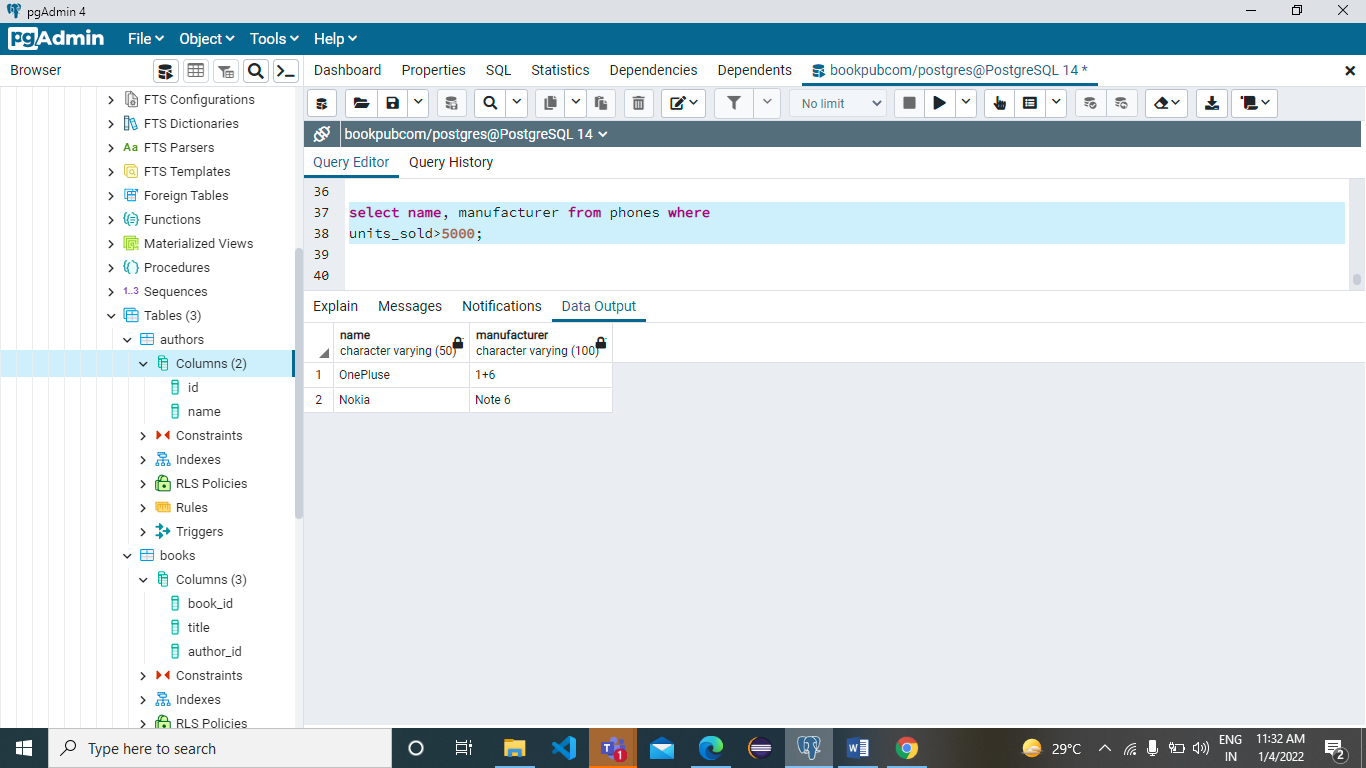


1. Write a query that will print the name and price of all phones that sold greater than 5000 units.

Ans:-

select name, manufacturer from phones where

units\_sold>5000;



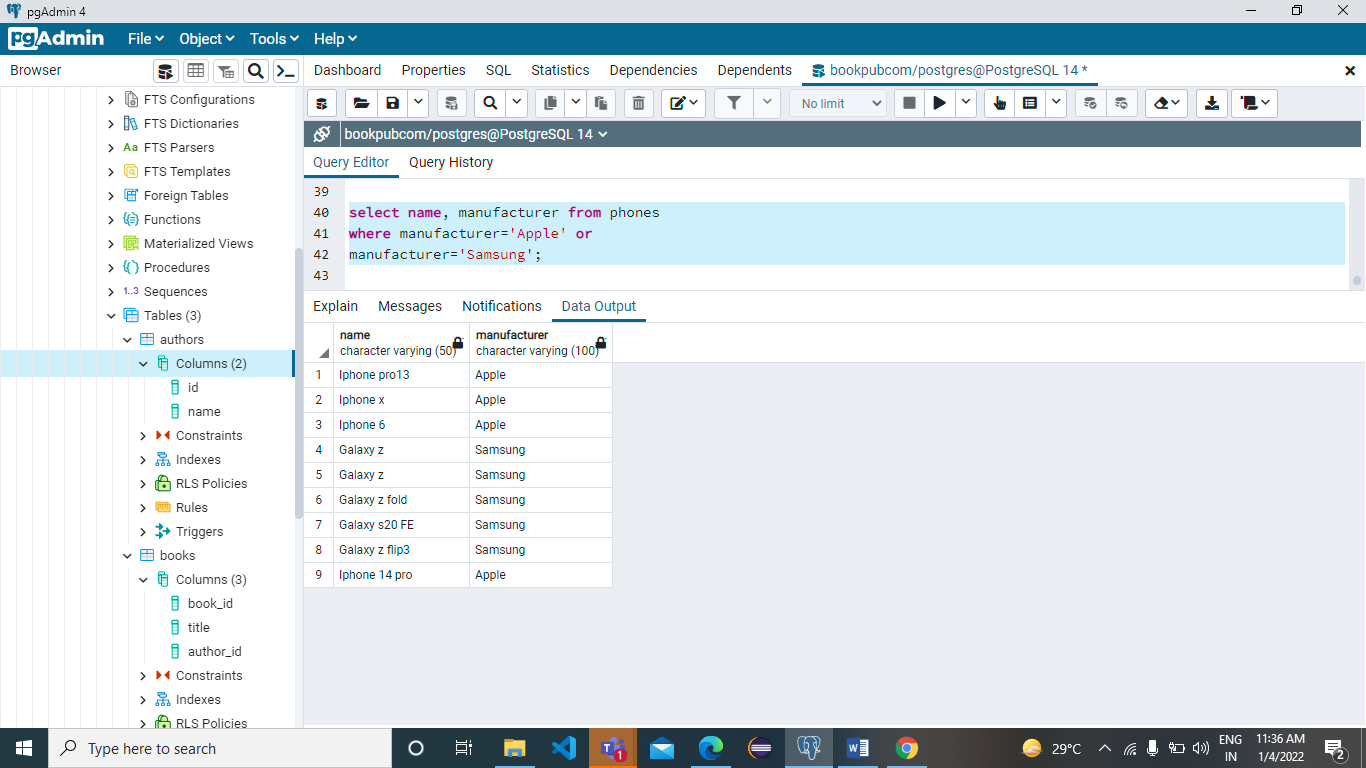
1. Write a query that will select the name and manufacturer for all phones created by Apple or Samsung.

Ans:-

select name, manufacturer from phones

where manufacturer='Apple' or

manufacturer='Samsung';

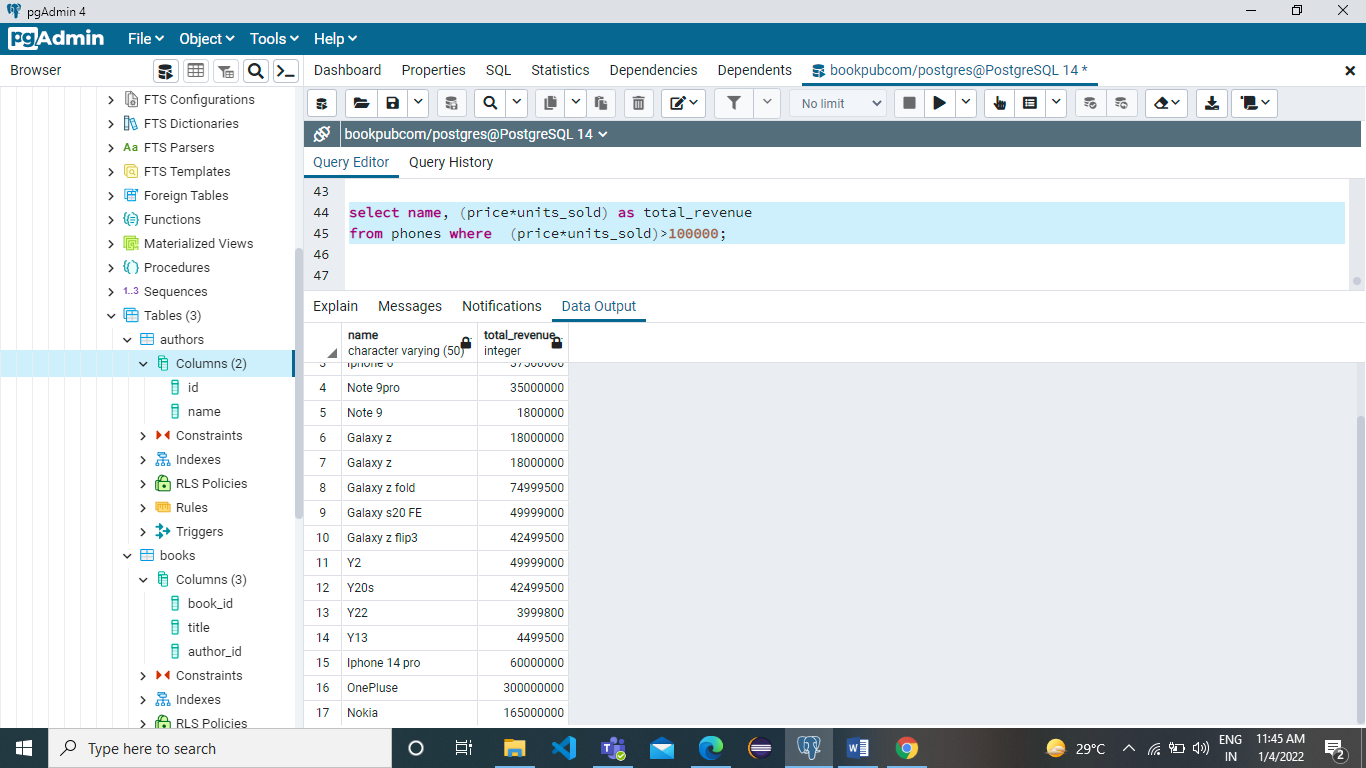


1. Write a query that will print the name and total\_revenue of all phones with a total\_revenue greater than 1,000,00.

Ans:

select name, (price\*units\_sold) as total\_revenue

from phones where (price\*units\_sold)>100000;



Create a new dataset from the shared product-orders dataset sheet.

Write the following queries:

1. Print the number of paid and unpaid orders.

Ans:

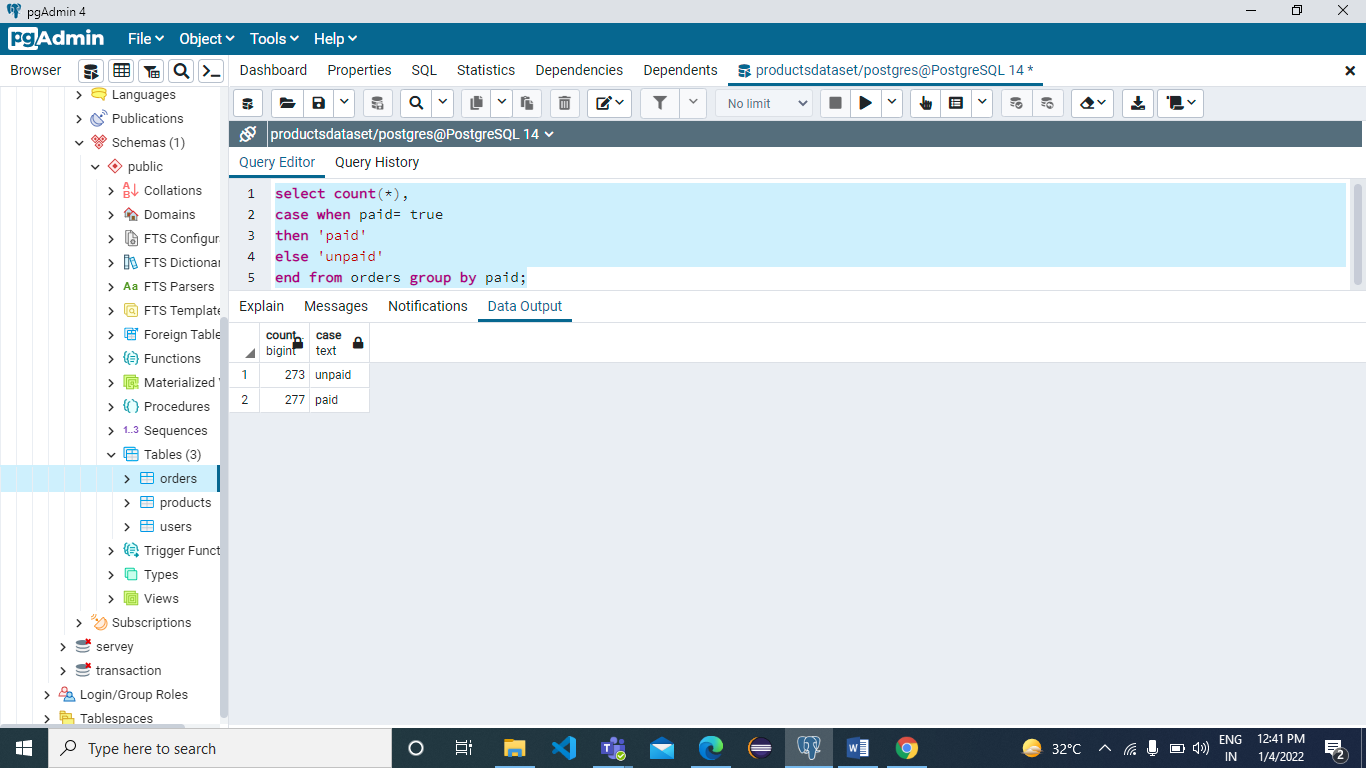
select count(\*),

case when paid= true

then 'paid'

else 'unpaid'

end from orders group by paid;



1. Print the first\_name and last\_name of each user along with the whether they have paid for their order.

Ans:-

select first\_name, last\_name, orders.paid from users

left JOIN orders ON orders.id = users.id;

