***SUBMITTED BY: - PARTH PARASHAR***

***PSU ID: -923928157***

## ***Activity- 5.1 Natural Join***

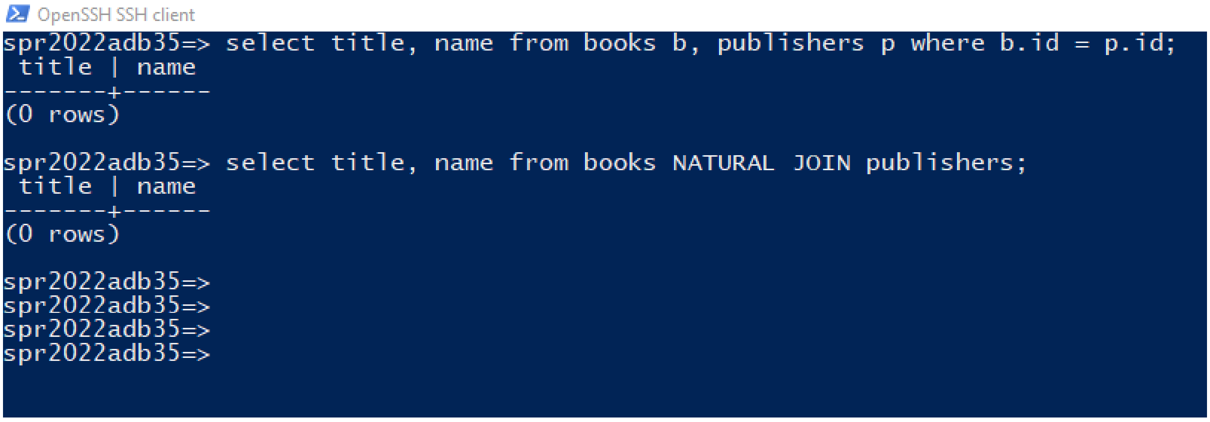
**Problem statement: -** Rewrite the SQL query without using natural join

Select title, name from books NATURAL JOIN publishers;

**Solution: -** The equivalent query for the problem is given below

select title, name from books NATURAL JOIN publishers;

The screenshot below provides the result for the same



```
spr2022adb35=> select title, name from books b, publishers p where b.id = p.id;
 title | name
-------+------
(0 rows)

spr2022adb35=> select title, name from books NATURAL JOIN publishers;
 title | name
-------+------
(0 rows)

spr2022adb35=>
spr2022adb35=>
spr2022adb35=>
spr2022adb35=>
```

## ***Activity-5.2 Union Compatibility***

**Problem Statement: -**

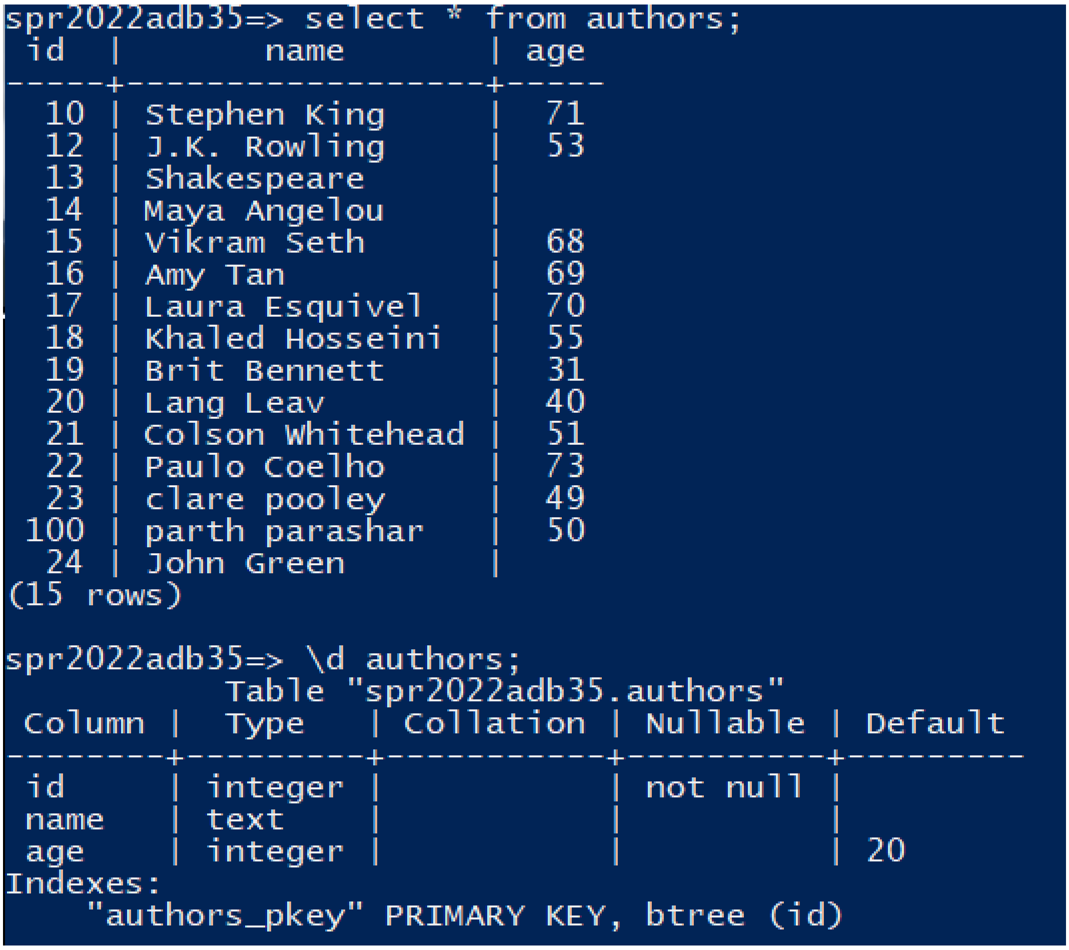
Create a union compatible table with Authors

Write Relational Algebra expression to find union of authors and cities

Describe the result of relational algebra expression

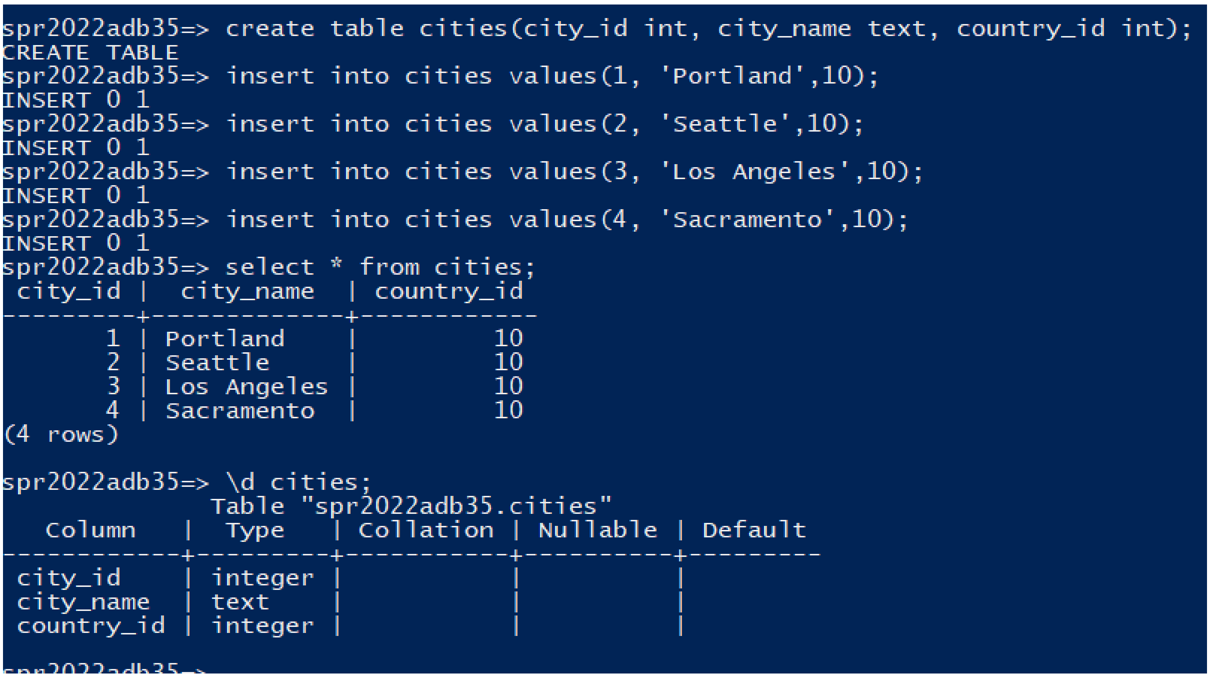
**Solution: -**

The structure and contents of the authors table is given below: -



```
spr2022adb35=> select * from authors;
 id  |        name        | age
-----+--------------------+-----
  10 | Stephen King       |  71
  12 | J.K. Rowling       |  53
  13 | Shakespeare        |
  14 | Maya Angelou       |
  15 | Vikram Seth        |  68
  16 | Amy Tan            |  69
  17 | Laura Esquivel     |  70
  18 | Khaled Hosseini    |  55
  19 | Brit Bennett       |  31
  20 | Lang Leav          |  40
  21 | Colson Whitehead   |  51
  22 | Paulo Coelho       |  73
  23 | clare pooley       |  49
 100 | parth parashar     |  50
  24 | John Green         |
(15 rows)

spr2022adb35=> \d authors;
           Table "spr2022adb35.authors"
 Column |  Type   | Collation | Nullable | Default
--------+---------+-----------+----------+---------
 id     | integer |           | not null |
 name   | text    |           |          |
 age    | integer |           |          | 20
Indexes:
    "authors_pkey" PRIMARY KEY, btree (id)
```

The new table with structure and contents is given below: -



```
spr2022adb35=> create table cities(city_id int, city_name text, country_id int);
CREATE TABLE
spr2022adb35=> insert into cities values(1, 'Portland',10);
INSERT 0 1
spr2022adb35=> insert into cities values(2, 'Seattle',10);
INSERT 0 1
spr2022adb35=> insert into cities values(3, 'Los Angeles',10);
INSERT 0 1
spr2022adb35=> insert into cities values(4, 'Sacramento',10);
INSERT 0 1
spr2022adb35=> select * from cities;
 city_id |  city_name  | country_id
---------+-------------+------------
       1 | Portland    |         10
       2 | Seattle     |         10
       3 | Los Angeles |         10
       4 | Sacramento  |         10
(4 rows)

spr2022adb35=> \d cities;
              Table "spr2022adb35.cities"
   Column    |  Type   | Collation | Nullable | Default
-------------+---------+-----------+----------+---------
 city_id     | integer |           |          |
 city_name   | text    |           |          |
 country_id  | integer |           |          |

spr2022adb35=>
```

These tables are both union compatible as both have three columns and both have same data types.

The relational Algebra expression is given below: -

Authors U Cities

The result will not make much sense. The only thing it will do is interlink the authors' tables attributes with the attributes of the Cities tables.

## ***Activity-5.3 Union and Union All Activity: -***

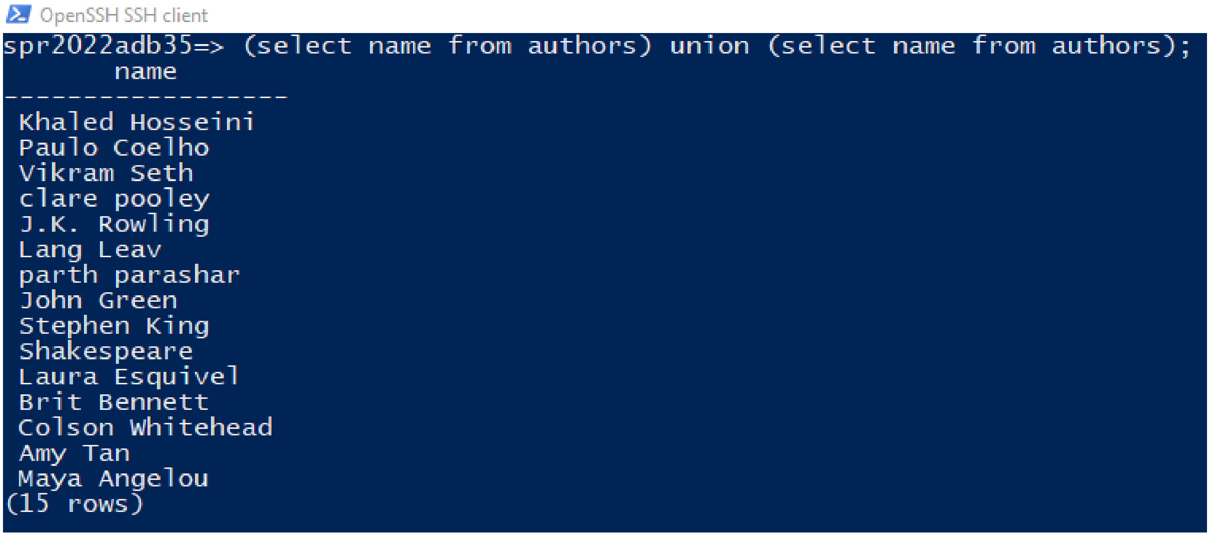
**Problem Statement: -**

1) SQL query for authors union author
2) Rewrite 1 with union all
3) Difference between the answers

**Answer: -**

1) (Select name from authors) union (select name from authors);

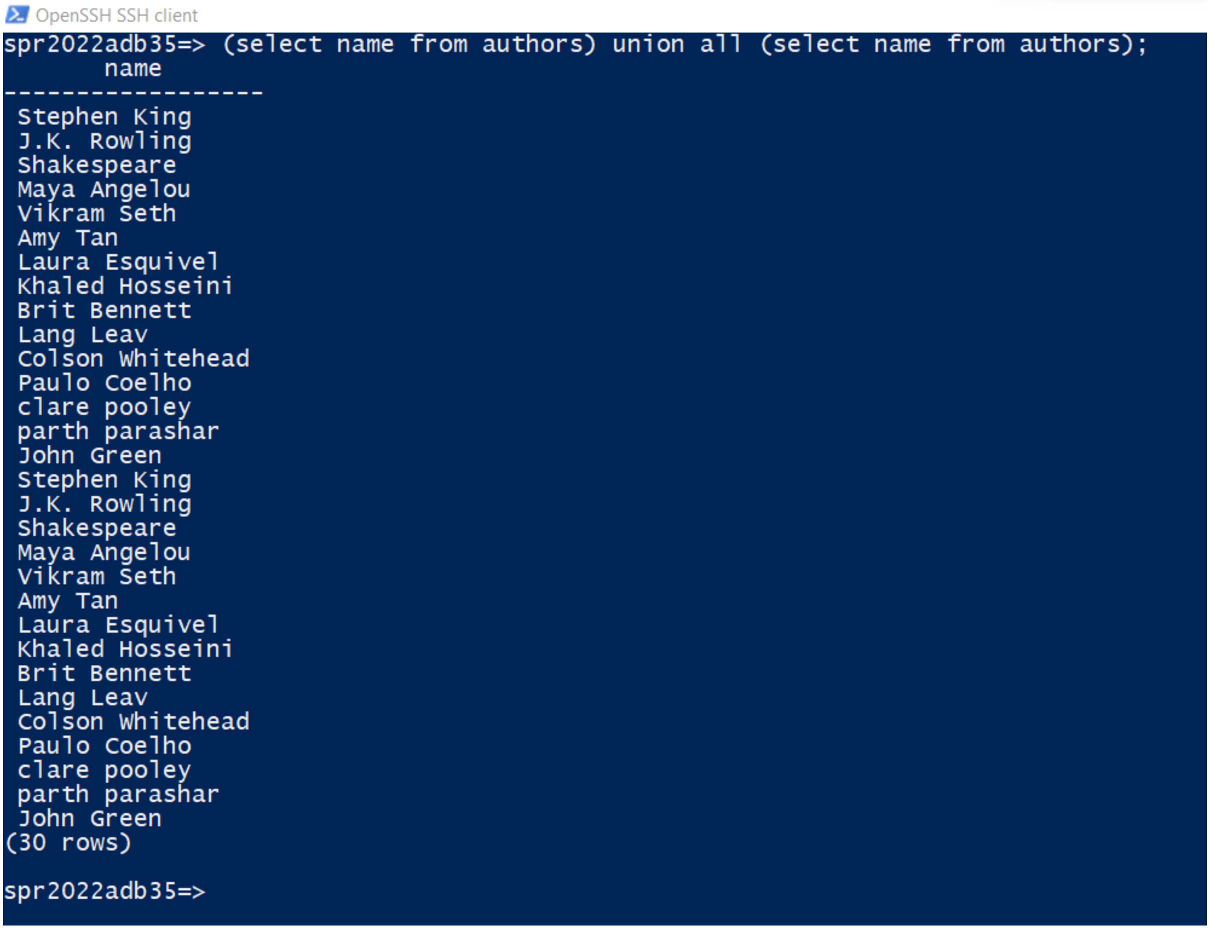
The output of this query is given in the screenshot provided below: -



```
spr2022adb35=> (select name from authors) union (select name from authors);
       name
------------------
 Khaled Hosseini
 Paulo Coelho
 Vikram Seth
 clare pooley
 J.K. Rowling
 Lang Leav
 parth parashar
 John Green
 Stephen King
 Shakespeare
 Laura Esquivel
 Brit Bennett
 Colson Whitehead
 Amy Tan
 Maya Angelou
(15 rows)
```

2) (Select name from authors) union all (select name from authors);

The output for this query is given in the screenshot provided below: -



```
spr2022adb35=> (select name from authors) union all (select name from authors);
       name
------------------
 Stephen King
 J.K. Rowling
 Shakespeare
 Maya Angelou
 Vikram Seth
 Amy Tan
 Laura Esquivel
 Khaled Hosseini
 Brit Bennett
 Lang Leav
 Colson Whitehead
 Paulo Coelho
 clare pooley
 parth parashar
 John Green
 Stephen King
 J.K. Rowling
 Shakespeare
 Maya Angelou
 Vikram Seth
 Amy Tan
 Laura Esquivel
 Khaled Hosseini
 Brit Bennett
 Lang Leav
 Colson Whitehead
 Paulo Coelho
 clare pooley
 parth parashar
 John Green
(30 rows)

spr2022adb35=>
```

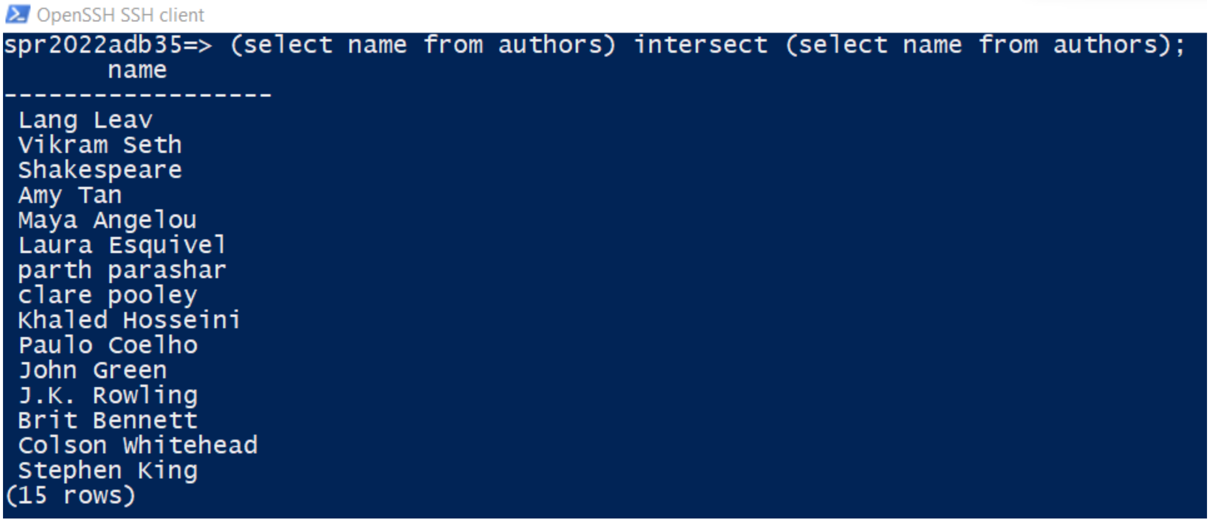
3) The difference between both the answers is that Union does not include any duplicates where the result generated from union all can have duplicate values. Therefore, the number of rows in the result of union all is more than that of union.

**Challenge: -**

**A) Intersect**

(Select name from authors) intersect (select name from authors);

The screenshot for the same is provided below: -

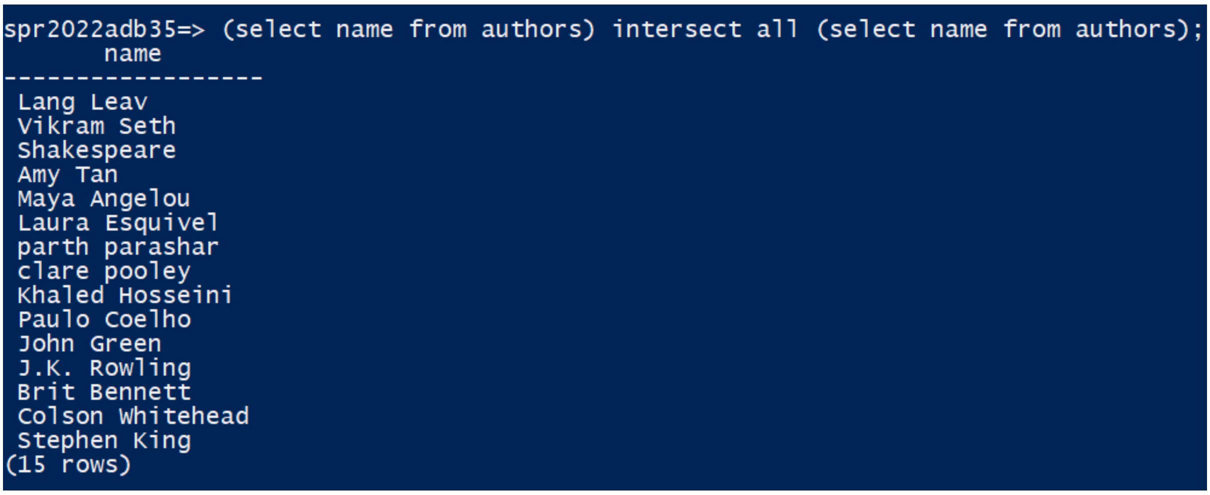


```
spr2022adb35=> (select name from authors) intersect (select name from authors);
       name
------------------
 Lang Leav
 Vikram Seth
 Shakespeare
 Amy Tan
 Maya Angelou
 Laura Esquivel
 parth parashar
 clare pooley
 Khaled Hosseini
 Paulo Coelho
 John Green
 J.K. Rowling
 Brit Bennett
 Colson Whitehead
 Stephen King
(15 rows)
```

### B) Intersect all

(select name from authors) intersect all (select name from authors);

The screenshot for this query is given below: -

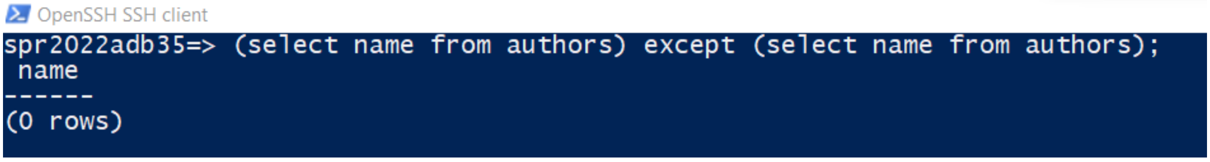


```
spr2022adb35=> (select name from authors) intersect all (select name from authors);
       name
------------------
 Lang Leav
 Vikram Seth
 Shakespeare
 Amy Tan
 Maya Angelou
 Laura Esquivel
 parth parashar
 clare pooley
 Khaled Hosseini
 Paulo Coelho
 John Green
 J.K. Rowling
 Brit Bennett
 Colson Whitehead
 Stephen King
(15 rows)
```

### C) Except

(select name from authors) except (select name from authors);

The screenshot for this query is given below: -

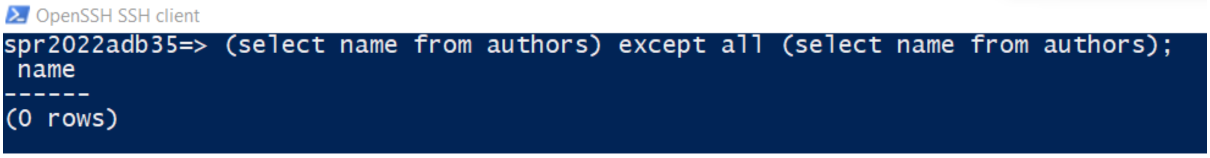


```
spr2022adb35=> (select name from authors) except (select name from authors);
 name
------
(0 rows)
```

## D) Except All

(select name from authors) except all (select name from authors);

The screenshot for this query is given below: -



```
spr2022adb35=> (select name from authors) except all (select name from authors);
 name
------
(0 rows)
```

## *ACTIVITY-6.1.A Outer Join Activity*

**Problem Statement: -** Write SQL queries for the following: -

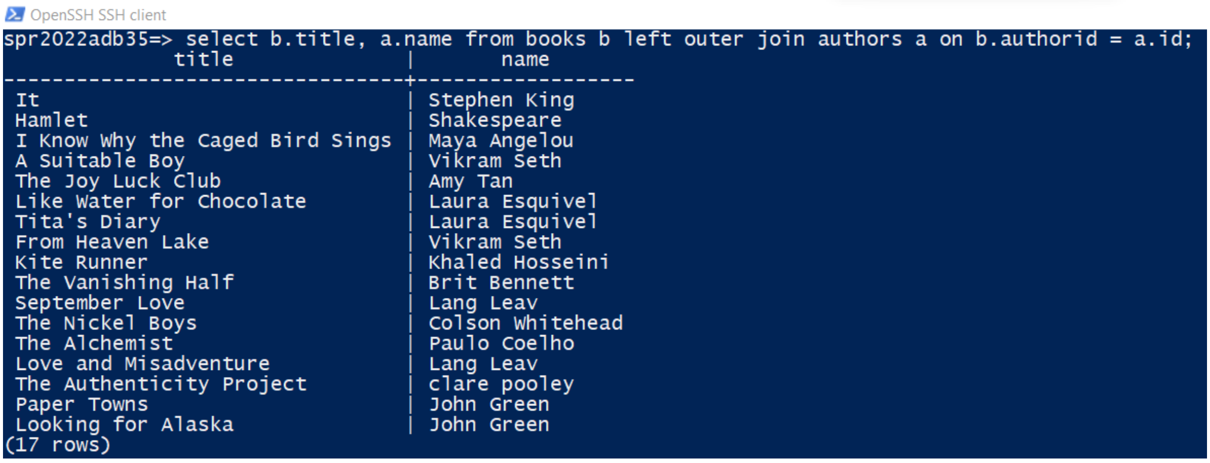
1) To find the titles of the books and the names of their authors which includes the books that do not have an author
2) To list the titles of the books and the names of their authors including books without an author and authors without a book

**Solution: -**

1) The query for this problem is: -

select b.title, a.name from books b left outer join authors a on b.authorid = a.id;

The screenshot for the result of this query is attached below: -

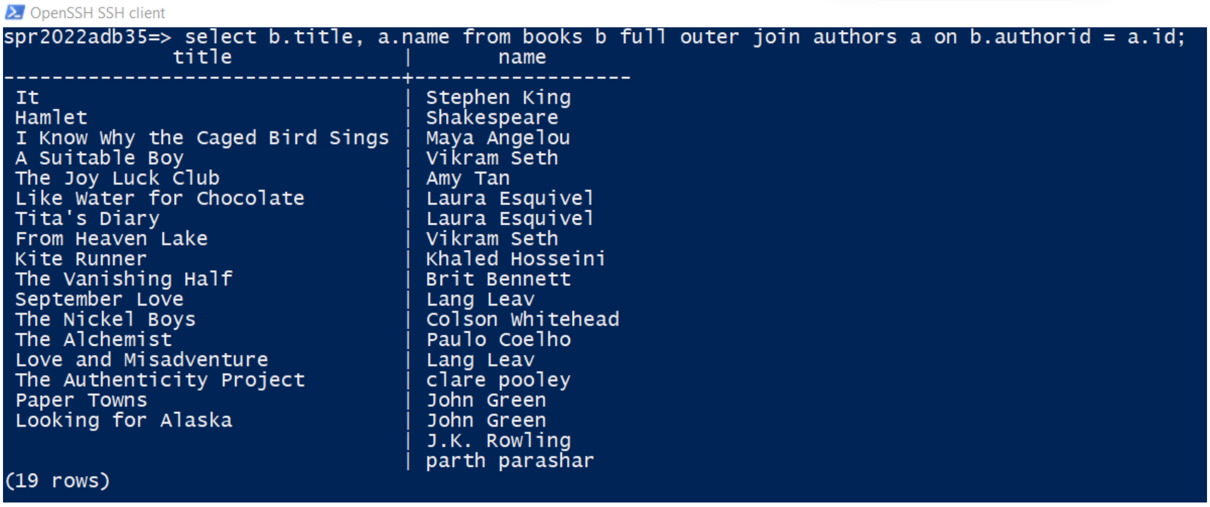


```
spr2022adb35=> select b.title, a.name from books b left outer join authors a on b.authorid = a.id;
              title               |       name
----------------------------------+------------------
 It                               | Stephen King
 Hamlet                           | Shakespeare
 I Know Why the Caged Bird Sings  | Maya Angelou
 A Suitable Boy                   | Vikram Seth
 The Joy Luck Club                | Amy Tan
 Like Water for Chocolate         | Laura Esquivel
 Tita's Diary                     | Laura Esquivel
 From Heaven Lake                 | Vikram Seth
 Kite Runner                      | Khaled Hosseini
 The Vanishing Half               | Brit Bennett
 September Love                   | Lang Leav
 The Nickel Boys                  | Colson Whitehead
 The Alchemist                    | Paulo Coelho
 Love and Misadventure            | Lang Leav
 The Authenticity Project         | clare pooley
 Paper Towns                      | John Green
 Looking for Alaska               | John Green
(17 rows)
```

2) The query for this problem is: -

select b.title, a.name from books b full outer join authors a on b.authorid = a.id;

The screenshot containing the result for this query is given below: -



```
spr2022adb35=> select b.title, a.name from books b full outer join authors a on b.authorid = a.id;
              title              |       name
---------------------------------+------------------
 It                              | Stephen King
 Hamlet                          | Shakespeare
 I Know Why the Caged Bird Sings | Maya Angelou
 A Suitable Boy                  | Vikram Seth
 The Joy Luck Club               | Amy Tan
 Like Water for Chocolate        | Laura Esquivel
 Tita's Diary                    | Laura Esquivel
 From Heaven Lake                | Vikram Seth
 Kite Runner                     | Khaled Hosseini
 The Vanishing Half              | Brit Bennett
 September Love                  | Lang Leav
 The Nickel Boys                 | Colson Whitehead
 The Alchemist                   | Paulo Coelho
 Love and Misadventure           | Lang Leav
 The Authenticity Project        | clare pooley
 Paper Towns                     | John Green
 Looking for Alaska              | John Green
                                 | J.K. Rowling
                                 | parth parashar
(19 rows)
```

## *ACTIVITY-6.1.B Outer Join Activity-II*

**Problem statement: -**

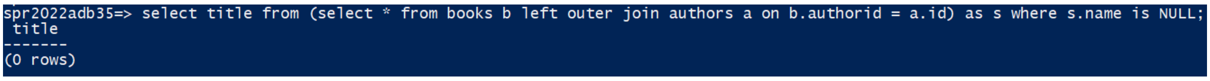
Write a query to find titles of books that do not have an author

**Solution: -**

The query that will be used is: -

select title from (select * from books b left outer join authors a on b.authorid = a.id) as s where s.name is NULL;

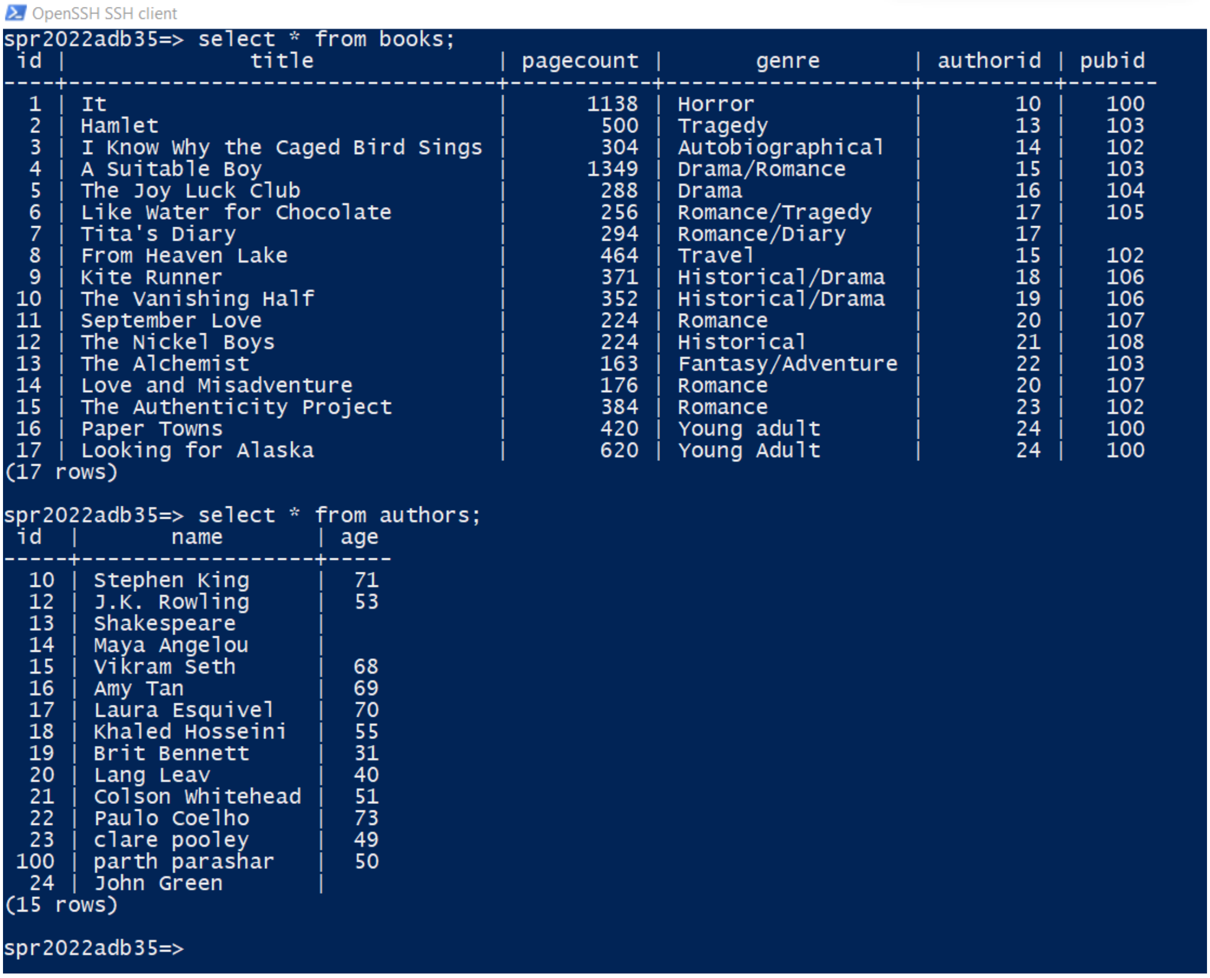
The screenshot contains the result of this query



```
spr2022adb35=> select title from (select * from books b left outer join authors a on b.authorid = a.id) as s where s.name is NULL;
 title
-------
(0 rows)
```

As we can see from the screenshot that the result of this query is 0 rows.

This is because each title has an author which can be confirmed by doing a select * on both books and authors tables.



```
spr2022adb35=> select * from books;
 id |              title               | pagecount |       genre        | authorid | pubid
----+----------------------------------+-----------+--------------------+----------+-------
  1 | It                               |      1138 | Horror             |       10 |   100
  2 | Hamlet                           |       500 | Tragedy            |       13 |   103
  3 | I Know Why the Caged Bird Sings  |       304 | Autobiographical   |       14 |   102
  4 | A Suitable Boy                   |      1349 | Drama/Romance      |       15 |   103
  5 | The Joy Luck Club                |       288 | Drama              |       16 |   104
  6 | Like Water for Chocolate         |       256 | Romance/Tragedy    |       17 |   105
  7 | Tita's Diary                     |       294 | Romance/Diary      |       17 |
  8 | From Heaven Lake                 |       464 | Travel             |       15 |   102
  9 | Kite Runner                      |       371 | Historical/Drama   |       18 |   106
 10 | The Vanishing Half               |       352 | Historical/Drama   |       19 |   106
 11 | September Love                   |       224 | Romance            |       20 |   107
 12 | The Nickel Boys                  |       224 | Historical         |       21 |   108
 13 | The Alchemist                    |       163 | Fantasy/Adventure  |       22 |   103
 14 | Love and Misadventure            |       176 | Romance            |       20 |   107
 15 | The Authenticity Project         |       384 | Romance            |       23 |   102
 16 | Paper Towns                      |       420 | Young adult        |       24 |   100
 17 | Looking for Alaska               |       620 | Young Adult        |       24 |   100
(17 rows)

spr2022adb35=> select * from authors;
 id  |       name       | age
-----+------------------+-----
  10 | Stephen King     |  71
  12 | J.K. Rowling     |  53
  13 | Shakespeare      |
  14 | Maya Angelou     |
  15 | Vikram Seth      |  68
  16 | Amy Tan          |  69
  17 | Laura Esquivel   |  70
  18 | Khaled Hosseini  |  55
  19 | Brit Bennett     |  31
  20 | Lang Leav        |  40
  21 | Colson Whitehead |  51
  22 | Paulo Coelho     |  73
  23 | clare pooley     |  49
 100 | parth parashar   |  50
  24 | John Green       |
(15 rows)

spr2022adb35=>
```

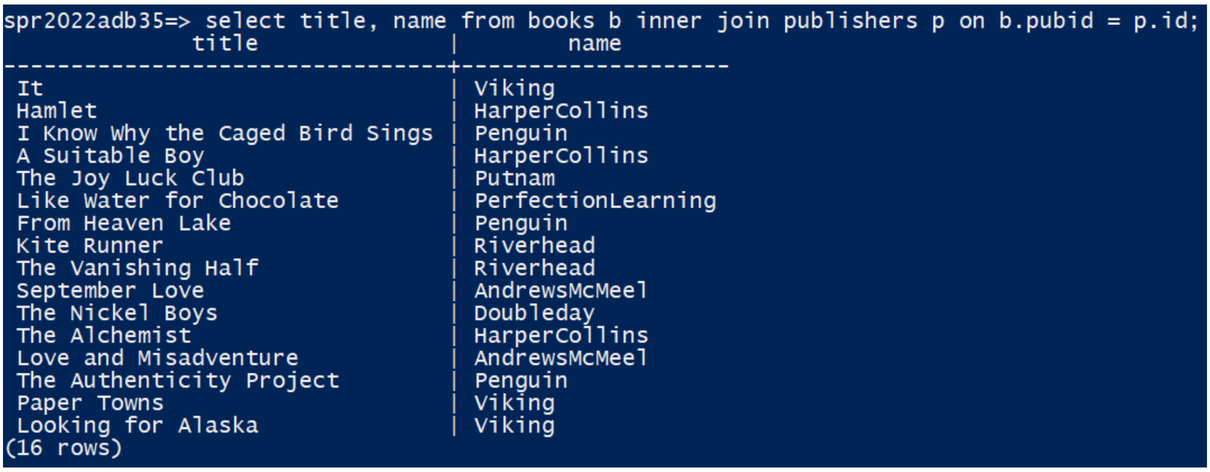
## ***Activity-6.1.C Join on Null Attributes***

**Problem Statement: -**

1) Run the query
2) What happens when pubid is null

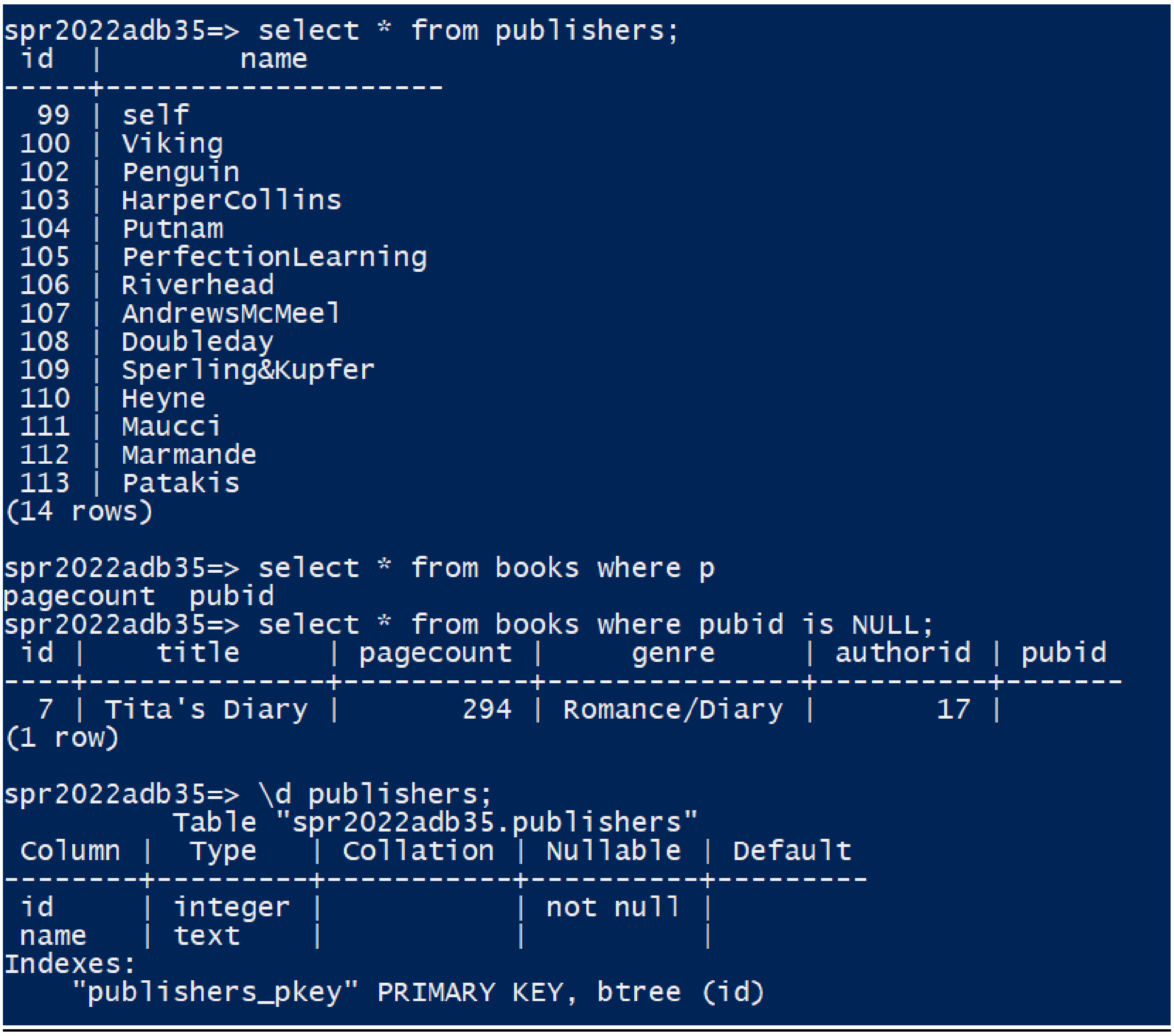
**Solution: -**

1) Select title, name from books b inner join publishers p on b.pubid = p.id;



```
spr2022adb35=> select title, name from books b inner join publishers p on b.pubid = p.id;
              title               |        name
----------------------------------+--------------------
 It                               | Viking
 Hamlet                           | HarperCollins
 I Know Why the Caged Bird Sings  | Penguin
 A Suitable Boy                   | HarperCollins
 The Joy Luck Club                | Putnam
 Like Water for Chocolate         | PerfectionLearning
 From Heaven Lake                 | Penguin
 Kite Runner                      | Riverhead
 The Vanishing Half               | Riverhead
 September Love                   | AndrewsMcMeel
 The Nickel Boys                  | Doubleday
 The Alchemist                    | HarperCollins
 Love and Misadventure            | AndrewsMcMeel
 The Authenticity Project         | Penguin
 Paper Towns                      | Viking
 Looking for Alaska               | Viking
(16 rows)
```

2) When pubid is null, then that book is not printed in the result because publisher id is the primary key for the publishers table and therefore, it cannot have null values. Since publishers' Id cannot be null for publishers table, therefore, the equality condition will not hold and hence it will not print it in the result.
   Also
   Null compared with null will give null.
   This is also given in the screenshot below: -



```
spr2022adb35=> select * from publishers;
 id  |        name
-----+--------------------
  99 | self
 100 | Viking
 102 | Penguin
 103 | HarperCollins
 104 | Putnam
 105 | PerfectionLearning
 106 | Riverhead
 107 | AndrewsMcMeel
 108 | Doubleday
 109 | Sperling&Kupfer
 110 | Heyne
 111 | Maucci
 112 | Marmande
 113 | Patakis
(14 rows)

spr2022adb35=> select * from books where p
pagecount  pubid
spr2022adb35=> select * from books where pubid is NULL;
 id |     title      | pagecount |     genre      | authorid | pubid
----+----------------+-----------+----------------+----------+-------
  7 | Tita's Diary   |       294 | Romance/Diary  |       17 |
(1 row)

spr2022adb35=> \d publishers;
          Table "spr2022adb35.publishers"
 Column |  Type   | Collation | Nullable | Default
--------+---------+-----------+----------+---------
 id     | integer |           | not null |
 name   | text    |           |          |
Indexes:
    "publishers_pkey" PRIMARY KEY, btree (id)
```

## ***Activity-6.2.A Group By/Having Clause***

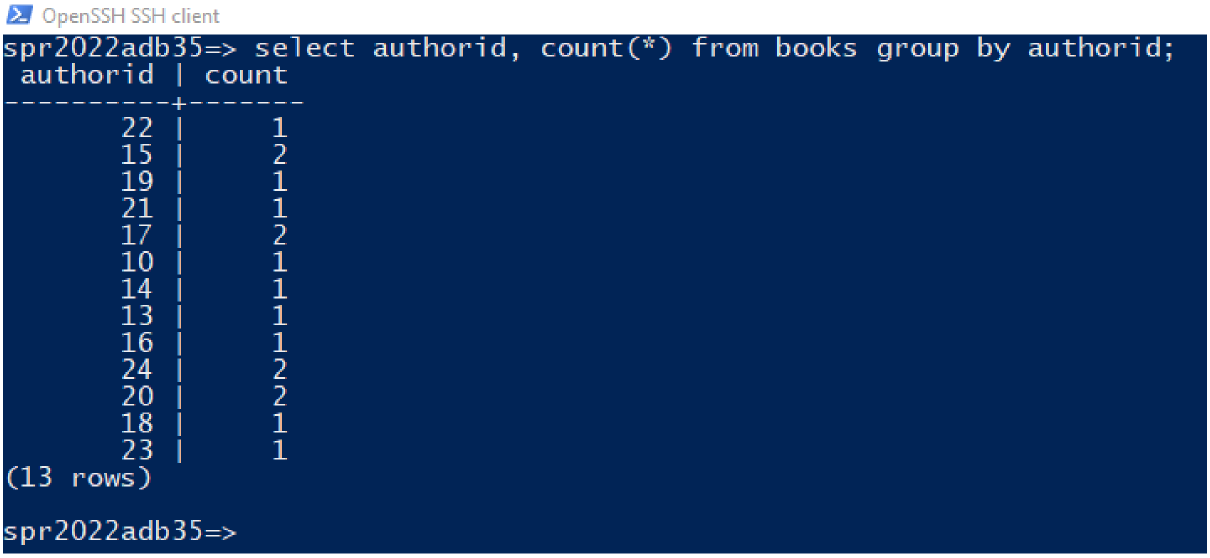
**Problem Statement: -**

Find the number of books by each author

1) Return author id and count
2) Return author name

**Solution: -**

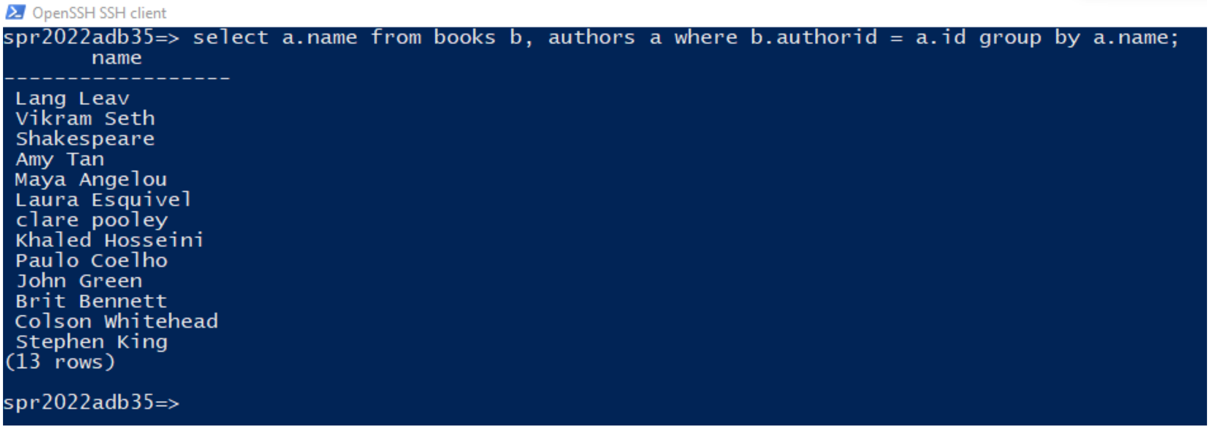
1) select authorid, count(*) from books group by authorid;
   The screenshot depicting the result of this query is given below: -



```
spr2022adb35=> select authorid, count(*) from books group by authorid;
 authorid | count
----------+-------
       22 |     1
       15 |     2
       19 |     1
       21 |     1
       17 |     2
       10 |     1
       14 |     1
       13 |     1
       16 |     1
       24 |     2
       20 |     2
       18 |     1
       23 |     1
(13 rows)

spr2022adb35=>
```

2) select a.name from books b, authors a where b.authorid = a.id group by a.name;

The screenshot depicting the result of this query is given below: -



```
spr2022adb35=> select a.name from books b, authors a where b.authorid = a.id group by a.name;
      name
------------------
 Lang Leav
 Vikram Seth
 Shakespeare
 Amy Tan
 Maya Angelou
 Laura Esquivel
 clare pooley
 Khaled Hosseini
 Paulo Coelho
 John Green
 Brit Bennett
 Colson Whitehead
 Stephen King
(13 rows)

spr2022adb35=>
```

## ***ACTIVITY 6.2.B- Group By/ Having clause***

**Problem statement: -**

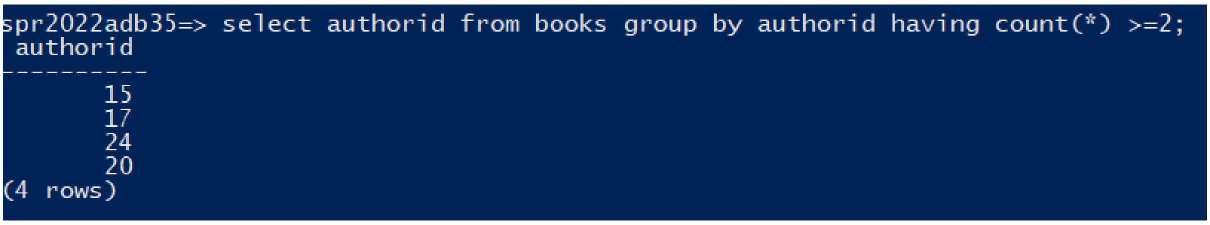
Find the authors that have written two or more books

1) return authored and count
2) return author name

**Solution: -**

1) select authorid from books group by authorid having count(*) >=2;

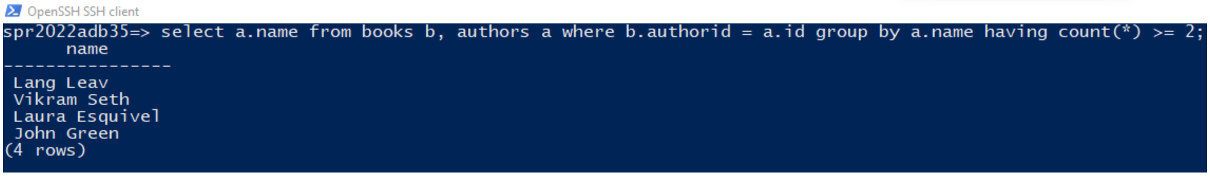
The screenshot depicting the result of this query is given below: -



```
spr2022adb35=> select authorid from books group by authorid having count(*) >=2;
 authorid
----------
       15
       17
       24
       20
(4 rows)
```

2) select a.name from books b, authors a where b.authorid = a.id group by a.name having count(*) >= 2;

The screenshot depicting the result of this query is given below: -



```
spr2022adb35=> select a.name from books b, authors a where b.authorid = a.id group by a.name having count(*) >= 2;
      name
---------------
 Lang Leav
 Vikram Seth
 Laura Esquivel
 John Green
(4 rows)
```

## ***ACTIVITY 6.2.C – Group by and nulls***

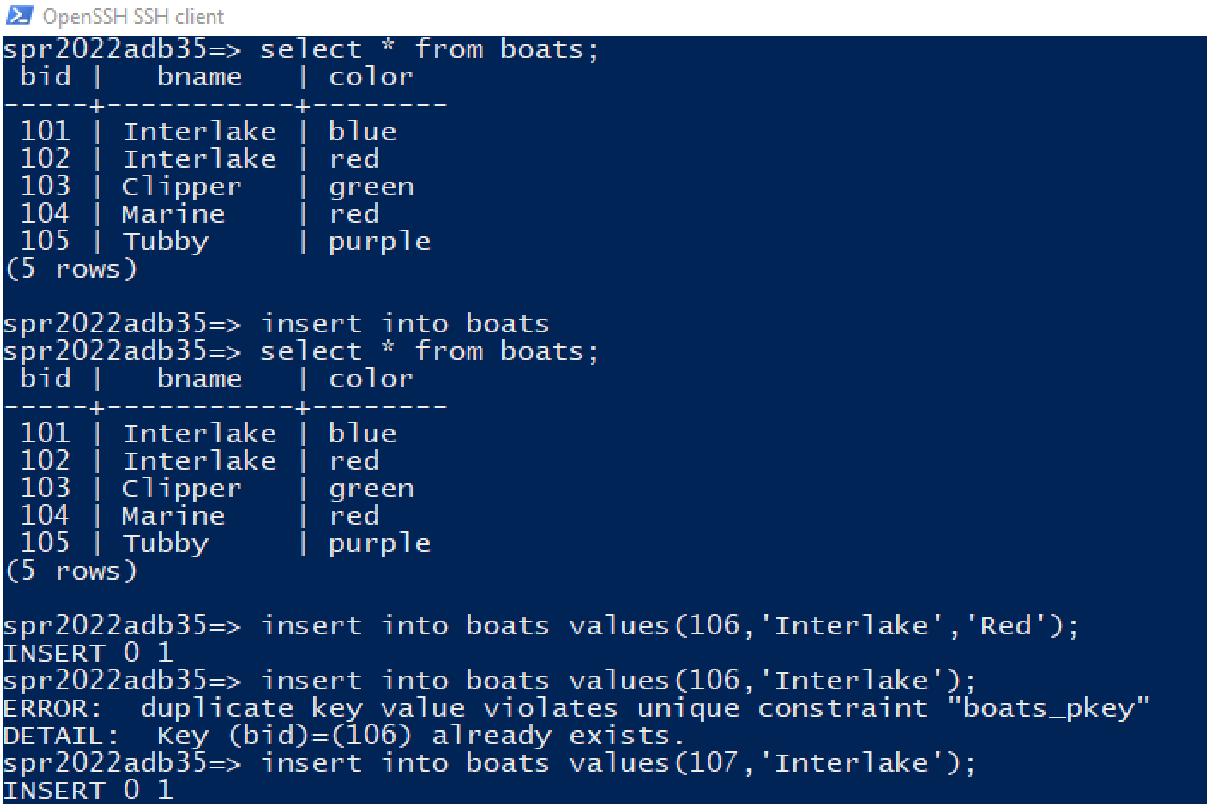
**Problem statement: -**

Run the following queries and print their result after inserting tuples in the boats database.

**Solution: -**

**Step-1: - Insertion into the boats table**

The following screenshot contains the queries used to insert tuples into the boats table

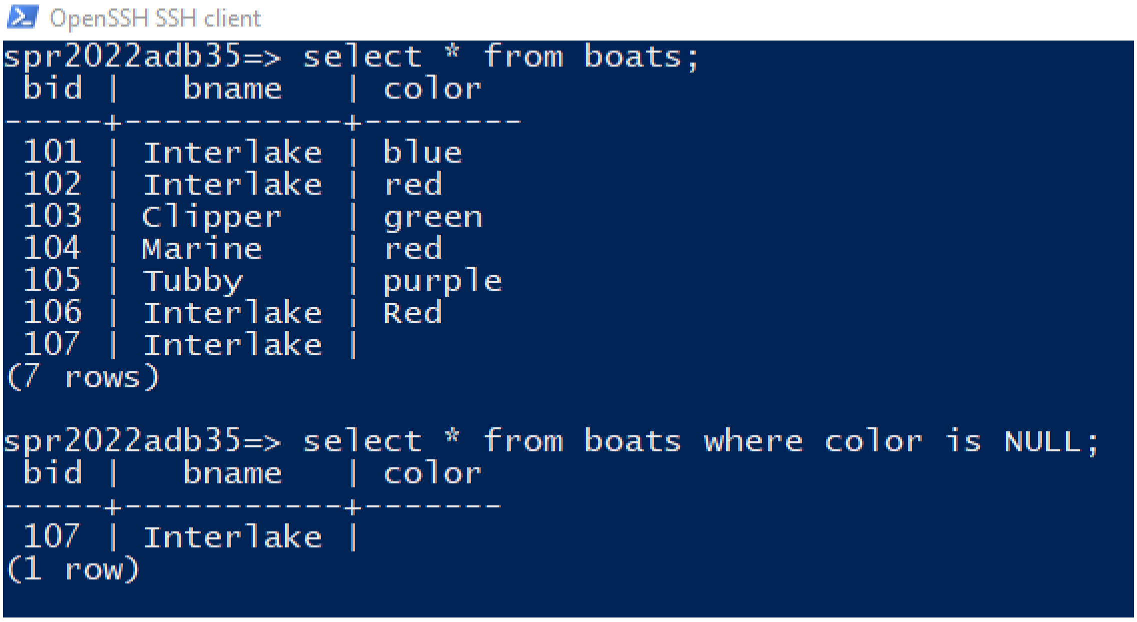


```
spr2022adb35=> select * from boats;
 bid |   bname    | color
-----+------------+--------
 101 | Interlake  | blue
 102 | Interlake  | red
 103 | Clipper    | green
 104 | Marine     | red
 105 | Tubby      | purple
(5 rows)

spr2022adb35=> insert into boats
spr2022adb35=> select * from boats;
 bid |   bname    | color
-----+------------+--------
 101 | Interlake  | blue
 102 | Interlake  | red
 103 | Clipper    | green
 104 | Marine     | red
 105 | Tubby      | purple
(5 rows)

spr2022adb35=> insert into boats values(106,'Interlake','Red');
INSERT 0 1
spr2022adb35=> insert into boats values(106,'Interlake');
ERROR:  duplicate key value violates unique constraint "boats_pkey"
DETAIL:  Key (bid)=(106) already exists.
spr2022adb35=> insert into boats values(107,'Interlake');
INSERT 0 1
```

Here, we can see that the query-2 provided for inserting into the boats table has an error. This is because in the slides, bid column's values is repeated. Since bid is a primary key, it cannot be the same for two tuples. Also, the value of colour is null for the entry with bid of 107.

**Step-2: - Verifying the insertion**

The insertion can be verified by printing all the values present in the boats table as given in the screenshot below: -

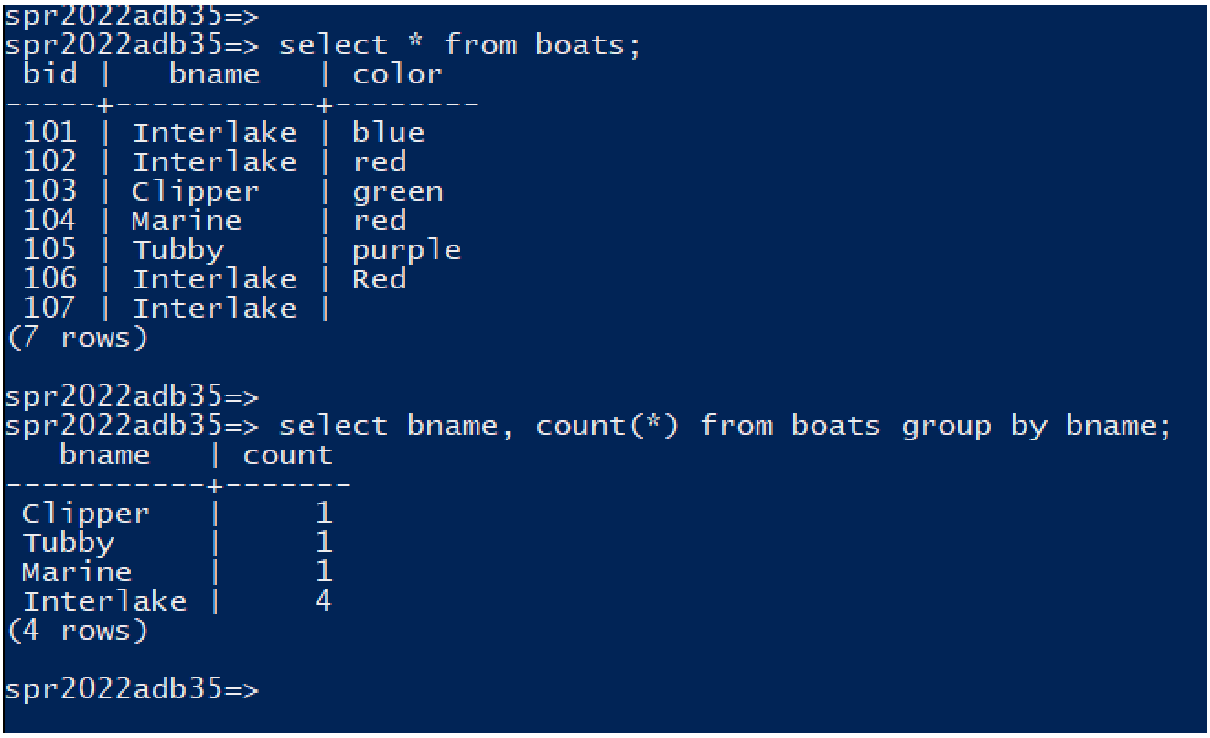


```
spr2022adb35=> select * from boats;
 bid |   bname    | color
-----+------------+--------
 101 | Interlake | blue
 102 | Interlake | red
 103 | Clipper   | green
 104 | Marine    | red
 105 | Tubby     | purple
 106 | Interlake | Red
 107 | Interlake |
(7 rows)

spr2022adb35=> select * from boats where color is NULL;
 bid |   bname    | color
-----+------------+-------
 107 | Interlake |
(1 row)
```

**Step-3: -** Running the queries given and printing the results along with comments

1) select bname, count(*) from boats group by bname;

The result of this query is given in the screenshot below: -

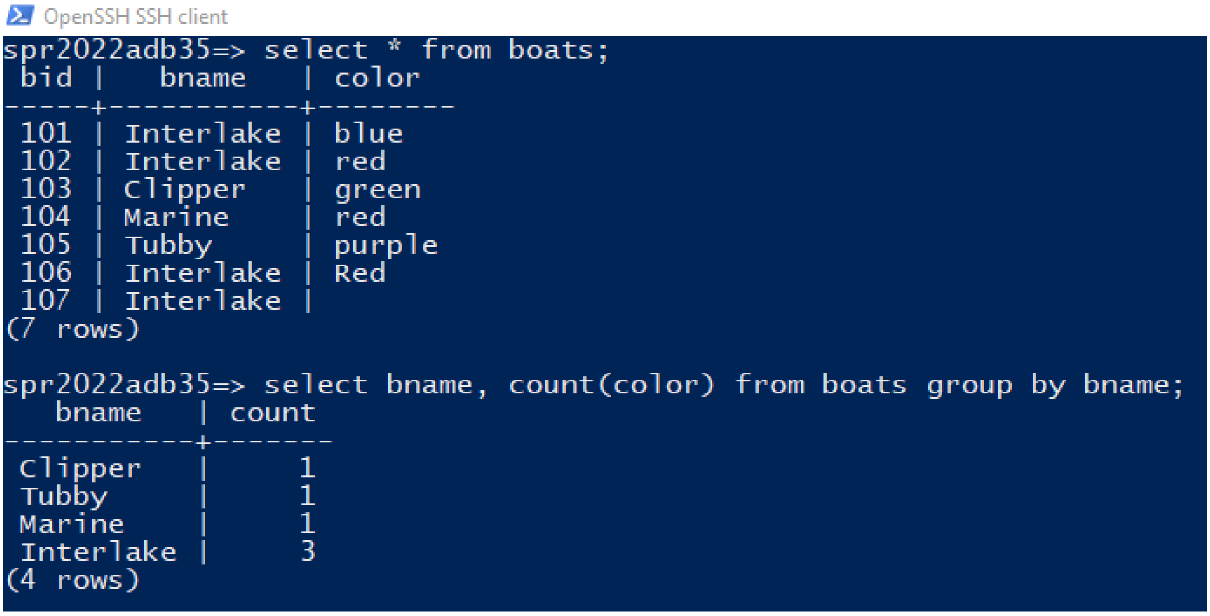


```
spr2022adb35=>
spr2022adb35=> select * from boats;
 bid |   bname    | color
-----+------------+--------
 101 | Interlake | blue
 102 | Interlake | red
 103 | Clipper   | green
 104 | Marine    | red
 105 | Tubby     | purple
 106 | Interlake | Red
 107 | Interlake |
(7 rows)

spr2022adb35=>
spr2022adb35=> select bname, count(*) from boats group by bname;
   bname    | count
------------+-------
 Clipper    |     1
 Tubby      |     1
 Marine     |     1
 Interlake |     4
(4 rows)

spr2022adb35=>
```

This is according to my expectation as count is on the group by clause of bname.

2) select bname, count(color) from boats group by bname;

The result of this query is given in the screenshot provided below: -

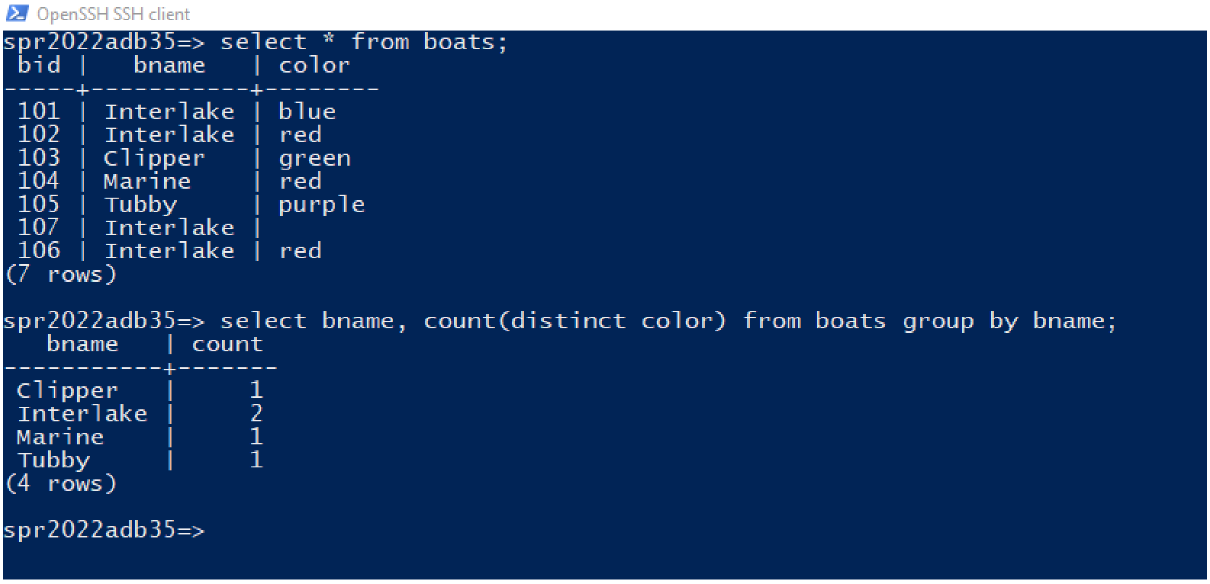


```
spr2022adb35=> select * from boats;
 bid |   bname    | color
-----+-----------+--------
 101 | Interlake | blue
 102 | Interlake | red
 103 | Clipper   | green
 104 | Marine    | red
 105 | Tubby     | purple
 106 | Interlake | Red
 107 | Interlake |
(7 rows)

spr2022adb35=> select bname, count(color) from boats group by bname;
   bname    | count
-----------+-------
 Clipper   |     1
 Tubby     |     1
 Marine    |     1
 Interlake |     3
(4 rows)
```

This is in accordance of my expectations as the count function was on the color and it will not take NULL value. Hence the number of 'Interlake' is 3.

3) select bname, count(distinct color) from boats group by bname;

The result of this query is given in the screenshot provided below: -



```
spr2022adb35=> select * from boats;
 bid |   bname    | color
-----+-----------+--------
 101 | Interlake | blue
 102 | Interlake | red
 103 | Clipper   | green
 104 | Marine    | red
 105 | Tubby     | purple
 107 | Interlake |
 106 | Interlake | red
(7 rows)

spr2022adb35=> select bname, count(distinct color) from boats group by bname;
   bname    | count
-----------+-------
 Clipper   |     1
 Interlake |     2
 Marine    |     1
 Tubby     |     1
(4 rows)

spr2022adb35=>
```

This is in accordance with what I was expecting as it will see distinct colors count.