ACTIVITY 7.1: - Subqueries with =

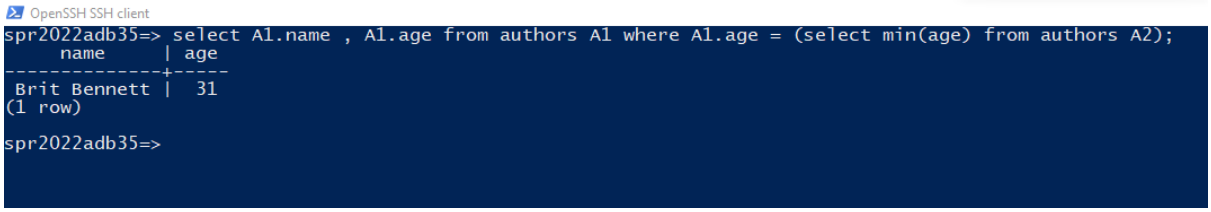
Problem statement: -

a) Write a SQL query to find the names and ages of the youngest authors
b) Try inserting another author with the same youngest age (31) and rerun the query

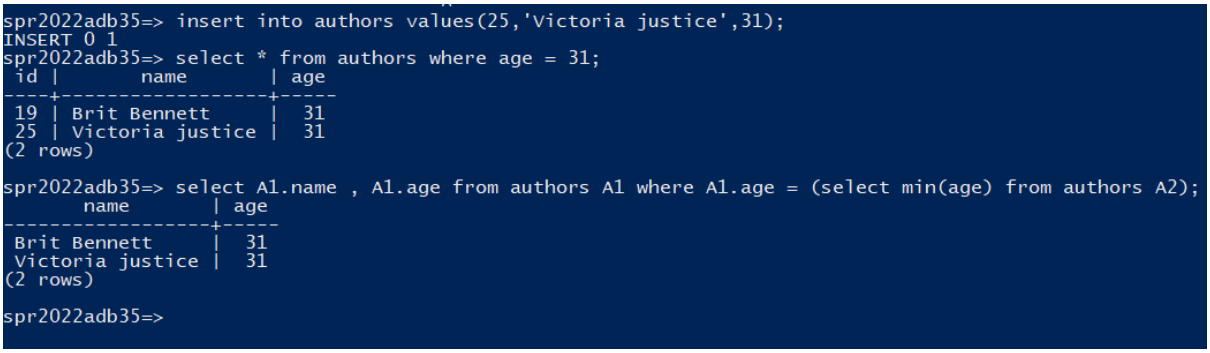
Solution: -

a) The query and the result is provided in the screenshot below: -

select A1.name , A1.age from authors A1 where A1.age = (select min(age) from authors A2);



b) When we try to insert another author with the same youngest age, then the query in a) returns two rows as given below: -



ACTIVITY 7.2: -Subqueries with IN

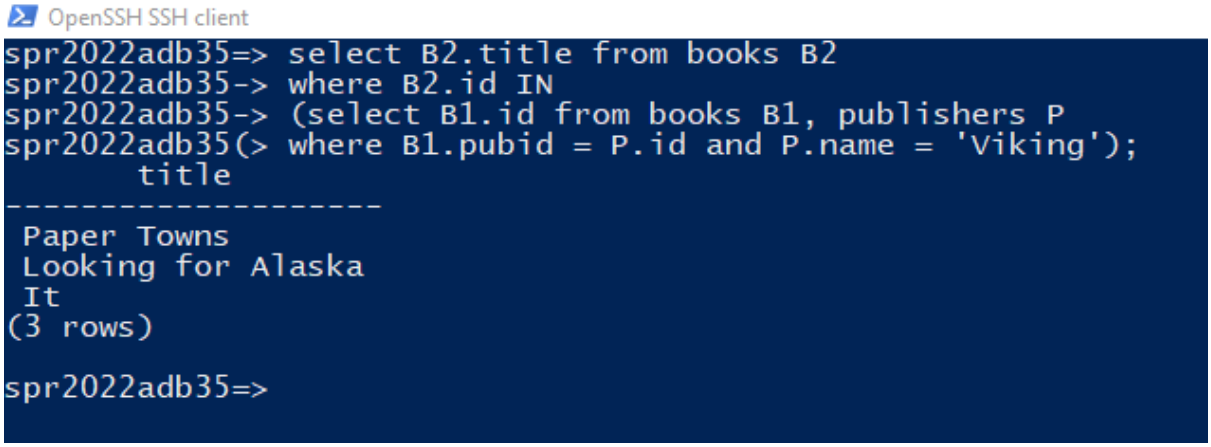
Problem statement: -

a) Find the titles of books published by Viking. Use a subquery.
b) Find the authors that have a book with a title that begins with the letter (use Like 'I%' to find titles that begin with I).
c) Redo 1 and 2 as joins

Solution: -

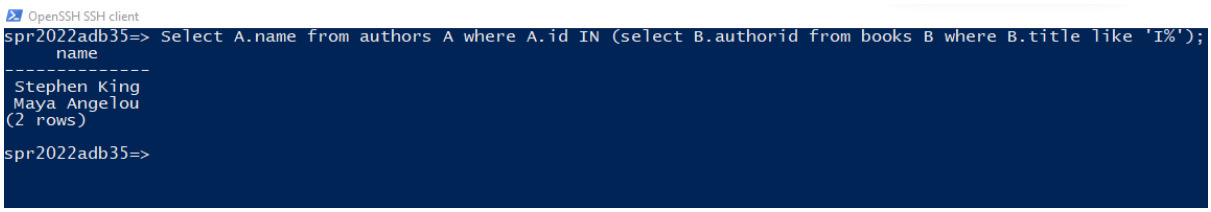
a) The query to solve this problem along with the results is given below: -

select B2.title from books B2 where B2.id IN (select B1.id from books B1, publishers P where B1.pubid = P.id and P.name = 'Viking');



b) The query to solve this problem along with the result is given below: -

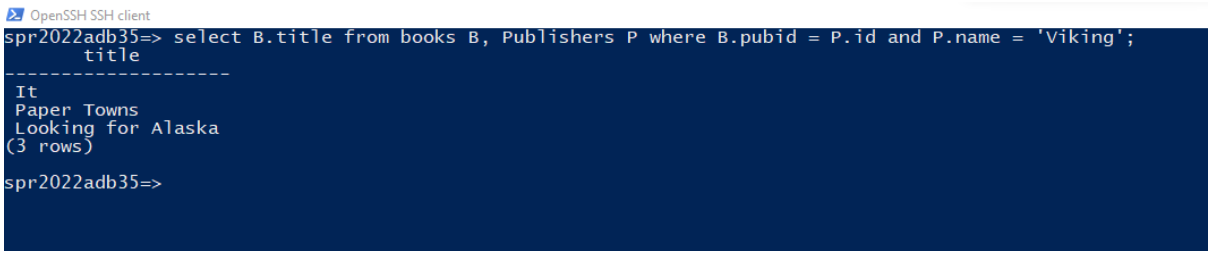
Select A.name from authors A where A.id IN (select B.authorid from books B where B.title like 'I%');



c) Writing a) and b) using joins, we get the following queries along with their results.

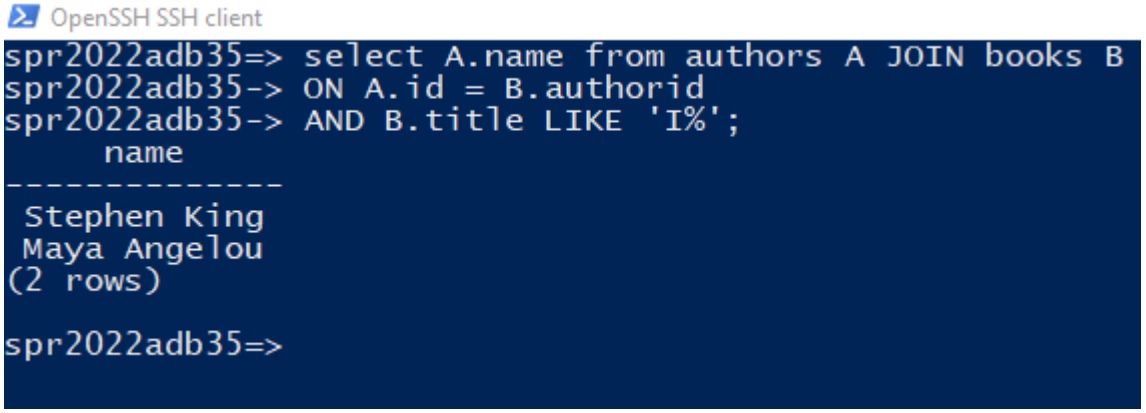
For a), we have the following: -

select B.title from books B, Publishers P where B.pubid = P.id and P.name = 'Viking';



For b), we have the following: -

select A.name from authors A JOIN books B ON A.id = B.authorid AND B.title LIKE 'I%';



ACTIVITY 7.3: - A mix of queries

Problem statement: -

9 queries written in order given in activity screen

Solution: -

a) The query and the screenshot of the result is given below: -

select A.name, P.name , count(*) as numbooks from authors A, books B, publishers P where A.id = B.authorid and B.pubid = P.id group by (A.name,P.name);

OpenSSH SSH client

```
spr2022adb35=> select A.name, P.name , count(*) as numbooks
spr2022adb35-> from authors A, books B, publishers P
spr2022adb35-> where A.id = B.authorid and B.pubid = P.id
spr2022adb35-> group by (A.name,P.name);
       name        |         name         | numbooks
-------------------+----------------------+----------
 Colson Whitehead  | Doubleday            |        1
 Lang Leav         | AndrewsMcMeel        |        2
 Stephen King      | Viking               |        1
 Amy Tan           | Putnam               |        1
 Vikram Seth       | HarperCollins        |        1
 Maya Angelou      | Penguin              |        1
 Shakespeare       | HarperCollins        |        1
 Brit Bennett      | Riverhead            |        1
 Vikram Seth       | Penguin              |        1
 Laura Esquivel    | PerfectionLearning   |        1
 Khaled Hosseini   | Riverhead            |        1
 Paulo Coelho      | HarperCollins        |        1
 clare pooley      | Penguin              |        1
 John Green        | Viking               |        2
(14 rows)

spr2022adb35=>
```

b) select A.name from authors A, books B, publishers P where A.id = B.authorid and B.pubid=P.id and P.name = 'Putnam';

OpenSSH SSH client

```
spr2022adb35=> select A.name from authors A, books B, publishers P
where A.id = B.authorid and B.pubid=P.id
and P.name = 'Putnam';
  name
---------
 Amy Tan
(1 row)

spr2022adb35=>
```

c) select A.name from authors A,books B where A.id = B.authorid group by A.name having count(*) > 1

OpenSSH SSH client

```
spr2022adb35=> select A.name from authors A,
books B where A.id = B.authorid
group by A.name having count(*) > 1
;
      name
----------------
 Lang Leav
 Vikram Seth
 Laura Esquivel
 John Green
(4 rows)

spr2022adb35=>
```

d) select A.name from authors A, books B, publishers P where A.id = B.authorid and B.pubid = P.id and P.name = ‘AndrewsMcMeal’ group by A.name having count(*) > 1;

```
spr2022adb35=>
spr2022adb35=>
spr2022adb35=> select A.name from
spr2022adb35-> authors A, books B, publishers P
spr2022adb35-> where A.id = B.authorid and
spr2022adb35-> B.pubid = P.id and
spr2022adb35-> P.name = 'AndrewsMcMeel'
spr2022adb35-> group by A.name
spr2022adb35-> having count(*) > 1;
   name
-----------
 Lang Leav
(1 row)

spr2022adb35=>
```

e) (select P.name from publishers P) except (select P.name from publishers P, books B where P.id = B.pubid);

OpenSSH SSH client

```
spr2022adb35=> (select P.name from publishers P )
spr2022adb35-> except
spr2022adb35-> (select P.name from publishers P, books B where P.id = B.pubid);
      name
----------------
 Patakis
 Marmande
 Maucci
 Heyne
 self
 Sperling&Kupfer
(6 rows)

spr2022adb35=>
```

f) (select P .name from publishers P)
except
(select P.name from publishers P , books B
where P.id = B.pubid)
union
(select A.name from authors A)
except
(select A.name from authors A, books B
where A.id = B.authorid);

```
OpenSSH SSH client
spr2022adb35=> (select P .name from publishers P)
spr2022adb35-> except
spr2022adb35-> (select P.name from publishers P , books B
spr2022adb35(> where P.id = B.pubid)
spr2022adb35-> union
spr2022adb35-> (select A.name from authors A)
spr2022adb35-> except
spr2022adb35-> (select A.name from authors A, books B
spr2022adb35(> where A.id = B.authorid);
       name
-----------------
 Victoria justice
 self
 Patakis
 J.K. Rowling
 Marmande
 parth parashar
 Heyne
 Sperling&Kupfer
 Maucci
(9 rows)

spr2022adb35=>
```

g) select P.name from publishers P, books B
where P.id = B.pubid group by P.id having count(*) =
(select max(count) from
(select count(*) as count from books B group by pubid)
as countbooks);

```
OpenSSH SSH client
spr2022adb35=> select P.name from
spr2022adb35-> publishers P, books B
spr2022adb35-> where P.id = B.pubid
spr2022adb35-> group by P.id having count(*) =
spr2022adb35-> (select max(count) from
spr2022adb35(> (select count(*) as count from books B group by pubid)
spr2022adb35(> as countbooks);
     name
--------------
 HarperCollins
 Penguin
 Viking
(3 rows)

spr2022adb35=>
```

h) select count(*) from authors A, books B, publishers P
where A.id = B.authorid and B.pubid = P.id
and P.name = 'AndrewsMcMeel' group by A.name having count(*) > 1;

OpenSSH SSH client

```
spr2022adb35=> select count(*) from authors A, books B, publishers P
spr2022adb35-> where A.id = B.authorid and B.pubid = P.id
spr2022adb35-> and P.name = 'AndrewsMcMeel' group by
spr2022adb35-> A.name having count(*) > 1;
 count
-------
     2
(1 row)

spr2022adb35=>
```

i) select distinct A2.name, A2.age, P2.name
from authors A2,publishers P2, books B2
where A2.id = B2.authorid and
B2.pubid = P2.id and A2.age = (
select min(age) from authors A1, books B1
where A1.id = B1.authorid and
B1.pubid = B2.pubid);

OpenSSH SSH client

```
spr2022adb35=> select distinct A2.name, A2.age, P2.name
from authors A2,publishers P2, books B2
where A2.id = B2.authorid and
B2.pubid = P2.id and A2.age = (
select min(age) from authors A1, books B1
where A1.id = B1.authorid and
B1.pubid = B2.pubid);
       name        | age |        name
-------------------+-----+--------------------
 Amy Tan           |  69 | Putnam
 Brit Bennett      |  31 | Riverhead
 clare pooley      |  49 | Penguin
 Colson Whitehead  |  51 | Doubleday
 Lang Leav         |  40 | AndrewsMcMeel
 Laura Esquivel    |  70 | PerfectionLearning
 Stephen King      |  71 | Viking
 Vikram Seth       |  68 | HarperCollins
(8 rows)

spr2022adb35=>
```

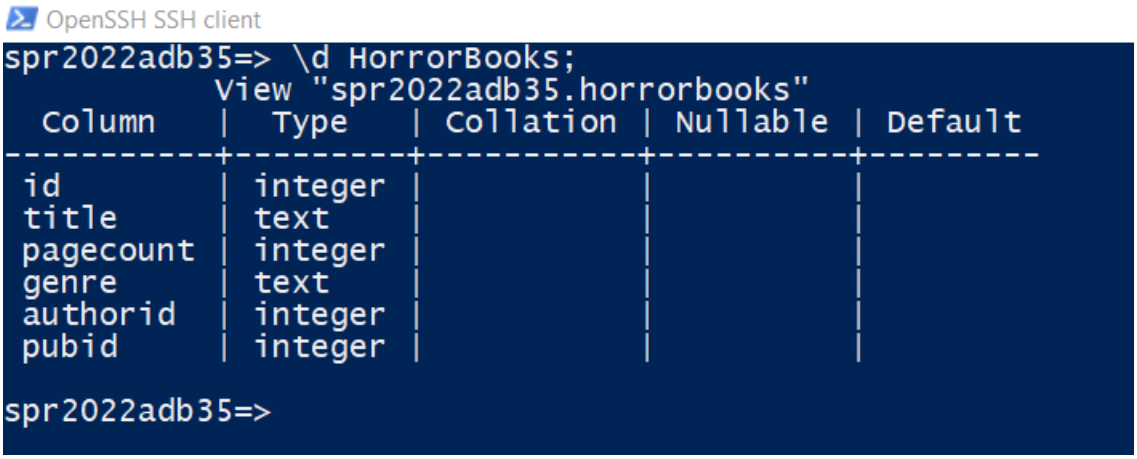
ACTIVITY 8.1 Write Some Views

Problem statement: -

a) Write a view that lists only horror books
b) Write a view that lists books published by viking
c) Which one of the two views can be updated

Solution: -

a) Create view HorrorBooks as
   Select * from books where genre= 'Horror';



b) create view VikingPubs as
   select * from books b, publishers p where
   b.pubid = p.id and p.name= 'Viking';

OpenSSH SSH client

```
spr2022adb35=> Create view VikingPubs as
Select * from  books B, publishers P where
B.pubid = P.id and P.name='Viking';
ERROR:  column "id" specified more than once
spr2022adb35=>
spr2022adb35=>
spr2022adb35=>
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 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=>
```

c) The first view can be updated and the second view cannot be updated. This is because the second view is formed with the joins of two views which makes it impossible to update the view on its own.