

-- Task :Subqueries and Nested Queries

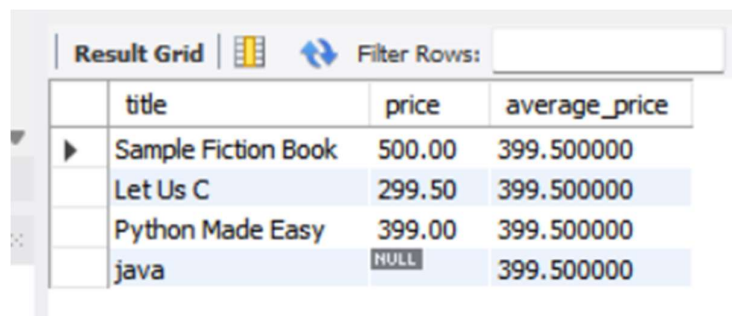
-- Database

USE library_management_system;

-- 1. Scalar Subquery – Show each book with average price for comparison

```
SELECT title, price,  
       (SELECT AVG(price) FROM Books) AS average_price  
FROM Books;
```

Output:-



The screenshot shows a 'Result Grid' window with a 'Filter Rows' input field. The grid contains four rows of data. The first row is 'Sample Fiction Book' with a price of 500.00 and an average price of 399.500000. The second row is 'Let Us C' with a price of 299.50 and an average price of 399.500000. The third row is 'Python Made Easy' with a price of 399.00 and an average price of 399.500000. The fourth row is 'java' with a price of NULL and an average price of 399.500000.

	title	price	average_price
▶	Sample Fiction Book	500.00	399.500000
	Let Us C	299.50	399.500000
	Python Made Easy	399.00	399.500000
	java	NULL	399.500000

-- insert sample data in books and issue table

```
INSERT INTO Books (isbn, title, category, price) VALUES ('FIC123', 'Sample Fiction Book',  
'Fiction', 500);
```

```
INSERT INTO Issue (issue_id, reader_id, isbn, issue_date) VALUES (101, 1, 'FIC123',  
CURDATE());
```

-- 2. Subquery with IN – Readers who issued books in the 'Fiction' category

```
SELECT first_name, last_name
```

```
FROM Readers
```

```
WHERE reader_id IN (
```

```
    SELECT reader_id
```

```
    FROM Issue
```

```
    WHERE isbn IN (
```

```

        SELECT isbn FROM Books WHERE category = 'Fiction'
    )
);

```

Output:-

Result Grid		Filter Rows:	
	first_name	last_name	
▶	Sahil	Verma	

-- 3. Subquery with EXISTS – Readers who have issued at least one book

```

SELECT first_name, last_name
FROM Readers r
WHERE EXISTS (
    SELECT 1
    FROM Issue i
    WHERE i.reader_id = r.reader_id
);

```

Output:-

Result Grid		Filter Rows:	
	first_name	last_name	
▶	Sahil	Verma	
	Meena	Patel	
	Priyanka	Nikam	

-- 4. Correlated Subquery – Show each reader and how many books they issued

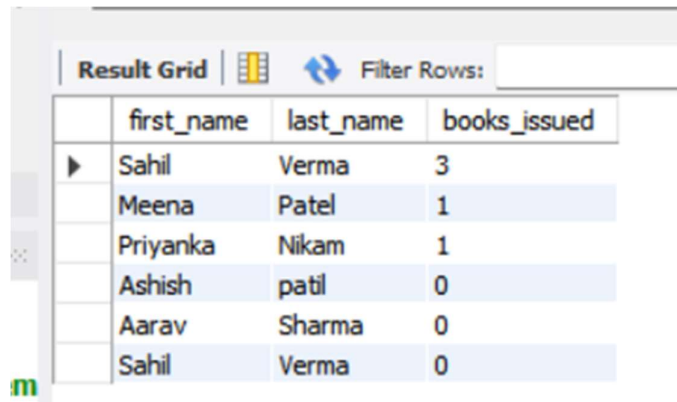
```

SELECT r.first_name, r.last_name,
    (SELECT COUNT(*)
     FROM Issue i
     WHERE i.reader_id = r.reader_id) AS books_issued

```

FROM Readers r;

Output:-



The screenshot shows a 'Result Grid' window with a 'Filter Rows' input field. The grid contains a table with four columns: an empty column, 'first_name', 'last_name', and 'books_issued'. There are seven rows of data. The first row is highlighted with a mouse cursor. A small 'm' icon is visible in the bottom left corner of the window.

	first_name	last_name	books_issued
▶	Sahil	Verma	3
	Meena	Patel	1
	Priyanka	Nikam	1
	Ashish	patil	0
	Aarav	Sharma	0
	Sahil	Verma	0

-- 5. Subquery in FROM (Derived Table) – Categories and their average price

SELECT category, average_price

FROM (

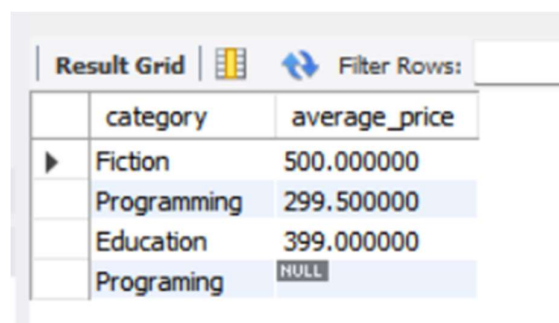
 SELECT category, AVG(price) AS average_price

 FROM Books

 GROUP BY category

) AS category_avg;

Output:-



The screenshot shows a 'Result Grid' window with a 'Filter Rows' input field. The grid contains a table with two columns: 'category' and 'average_price'. There are five rows of data. The first row is highlighted with a mouse cursor. The last row has a 'NULL' value in the 'average_price' column.

	category	average_price
▶	Fiction	500.000000
	Programming	299.500000
	Education	399.000000
	Programing	NULL

-- 6. Subquery with = – Find the most expensive book

SELECT title, price

FROM Books

WHERE price = (SELECT MAX(price) FROM Books);

Output:-

Result Grid

Filter Rows:

	title	price
▶	Sample Fiction Book	500.00