

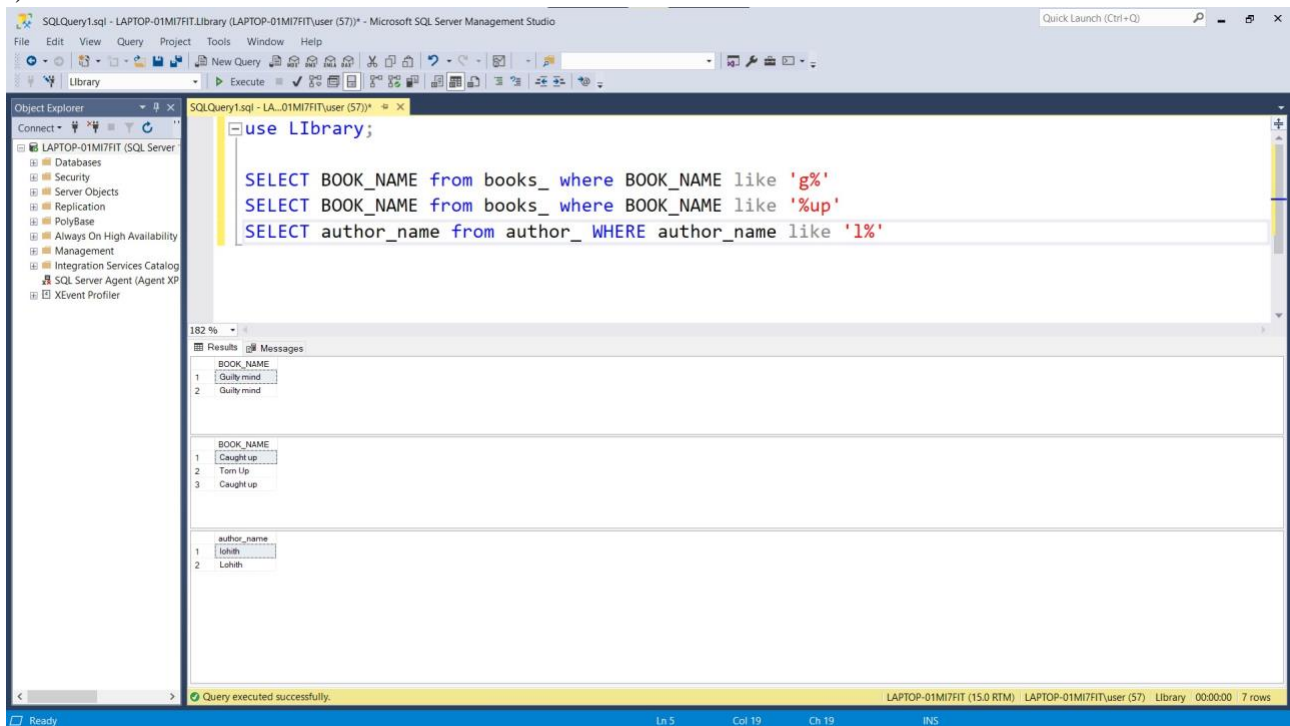
DBMS LAB ASSIGNMENT-5

Name:-BOKKA SRIKAR

Roll no:-19BCS024

1) Illustrate logical ANY, ALL and LIKE operator- the queries should be relevant to your respective databases 3 queries for each operator. One query explaining the difference between ANY and ALL

i) LIKE :-



The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The query window contains the following SQL code:

```
use Library;  
  
SELECT BOOK_NAME from books_ where BOOK_NAME like 'g%'  
SELECT BOOK_NAME from books_ where BOOK_NAME like '%up'  
SELECT author_name from author_ WHERE author_name like '1%'
```

The Results pane displays the output for each query:

BOOK_NAME
Guilty mind
Guilty mind

BOOK_NAME
Caught up
Tom Up
Caught up

author_name
Lohith
Lohith

The status bar at the bottom indicates "Query executed successfully." and "7 rows".

~

ii) ALL :-

The screenshot displays two instances of Microsoft SQL Server Management Studio. The top instance shows a query that filters books by the number of issues and then selects authors with IDs greater than the maximum number of issues. The bottom instance shows a query that selects authors whose IDs are greater than the maximum ID from a subset of books (where ID < 6).

Top Screenshot:

```

SELECT no_of_issues from books_
where no_of_issues > all(
select no_of_issues from books_ WHERE author_id>14 )

SELECT author_id,author_name from
author_ WHERE author_id > all(
select author_id from books_)

```

Results:

no_of_issues
12
23
45
67
34
24
75
29
13
12
12
13

author_id	author_name
16	Priya
17	Vishwa
18	Geetha

Query executed successfully. LAPTOP-01M7FIT (15.0 RTM) LAPTOP-01M7FIT\user (57) Library 00:00:00 15 rows

Bottom Screenshot:

```

SELECT author_id,author_name from
author_ WHERE author_id > all(
select author_id from books_ where author_id<6)

```

Results:

author_id	author_name
6	chandra
7	lohit
8	Bhanupriya
9	Lohith
10	Koushik
11	Sohail
12	Greeshma
13	Malikarjun
14	Pavan
15	Suman
16	Priya
17	Vishwa
18	Geetha

Query executed successfully. LAPTOP-01M7FIT (15.0 RTM) LAPTOP-01M7FIT\user (57) Library 00:00:00 13 rows

iii) ANY :-

SQLQuery1.sql - LA_01MI7FIT(user (57))

```

SELECT no_of_issues from books_
where no_of_issues > ANY(
select no_of_issues from books_ WHERE author_id>15 )

SELECT author_id,BOOK_NAME from books_
WHERE author_id > ANY(
select author_id from books_ WHERE published_year>2015)

```

Results

author_id	BOOK_NAME
8	Close my eyes
7	My Band
8	Fantasy Girl
15	No Flaws
10	Drop It
8	Close my eyes
7	My Band
8	Fantasy Girl
9	No Flaws
10	Drop It
9	ALONE
9	SOLO

Query executed successfully. LAPTOP-01MI7FIT (15.0 RTM) LAPTOP-01MI7FIT(user (57)) Library 00:00:00 12 rows

SQLQuery1.sql - LA_01MI7FIT(user (57))

```

SELECT author_id,BOOK_NAME from books_
WHERE author_id > ANY(
select author_id from books_ where bought_year>2019)

```

Results

author_id	BOOK_NAME
8	Close my eyes
8	Fantasy Girl
15	No Flaws
10	Drop It
8	Close my eyes
8	Fantasy Girl
9	No Flaws
10	Drop It
9	ALONE
9	SOLO

Query executed successfully. LAPTOP-01MI7FIT (15.0 RTM) LAPTOP-01MI7FIT(user (57)) Library 00:00:00 10 rows

~ iv) ANY and ALL comparison

:-

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the server structure. The central query editor contains the following SQL query:

```
select author_id from books_
where author_id > all(
select author_id from books_ where author_id < 4 )
```

The Results pane at the bottom shows the output of the query, which is a single column named 'author_id' with 18 rows of data. The status bar at the bottom indicates 'Query executed successfully.' and '18 rows'.

author_id
4
8
5
6
7
8
15
10
4
8
5
6
7
8
15
9
10
9
9

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the server structure. The central query editor contains the following SQL query:

```
select author_id from books_
where author_id > any(
select author_id from books_ where author_id < 4 )
```

The Results pane at the bottom shows the output of the query, which is a single column named 'author_id' with 19 rows of data. The status bar at the bottom indicates 'Query executed successfully.' and '19 rows'.

author_id
4
2
8
5
6
7
8
15
10
4
8
5
6
7
8
15
9
10
9
9

2). One query for each Aggregate function.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'LAPTOP-01M7FIT' (SQL Server 15.0.2000). The main query window contains the following SQL code:

```
use Library

select sum(PRICE) from books_
select avg(PRICE) from books_ where author_id > 2
select count(no_of_issues) from books_ where author_id > 4
select max(PRICE) from books_ where author_id > 6
select min(no_of_issues) from books_ where author_id > 8
```

The Results pane shows the output of the queries:

(No column name)
18599
(No column name)
1002
(No column name)
14
(No column name)
890
(No column name)
1

The status bar at the bottom indicates 'Query executed successfully.' and 'LAPTOP-01M7FIT (15.0 RTM) LAPTOP-01M7FIT(user (61)) Library 00:00:00 5 rows'.

~

3)

Illustrate the usage of order by, group by and having clause (2 queries for each case)

i)ORDER BY :-

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'LAPTOP-01M7FIT' (SQL Server 15.0.2000). The main query window contains the following SQL code:

```
use Library

select BOOK_NAME,published_year from books_ order by published_year desc
```

The Results pane shows the output of the query:

BOOK_NAME	published_year
Leave me alone	2016
Guilty mind	2015
Close my eyes	2014
Tom Up	2013
Guilty mind	2013
Drop It	2013
Drop It	2012
Caught up	2012
In the Dark	2011
Natural Disaster	2010
My Band	2010
Leave me alone	2010
No Flaws	2010
No Flaws	2002
Fantasy Girl	2001
Close my eyes	1998
My Band	1996
Fantasy Girl	1996
Caught up	1994

The status bar at the bottom indicates 'Query executed successfully.' and 'LAPTOP-01M7FIT (15.0 RTM) LAPTOP-01M7FIT(user (66)) Library 00:00:00 19 rows'.

~

SQLQuery3.sql - LAPTOP-01M7FIT.Library (LAPTOP-01M7FIT\user (66)) - Microsoft SQL Server Management Studio

Object Explorer: LAPTOP-01M7FIT (SQL Server 15.0.2000.)

Query: `use Library`
`select BOOK_NAME, PRICE from books_order by PRICE`

Results:

	BOOK_NAME	PRICE
1	Close my eyes	160
2	Close my eyes	160
3	Drop It	450
4	Drop It	450
5	In the Dark	567
6	No Flaws	679
7	No Flaws	679
8	Fantasy Girl	799
9	Fantasy Girl	799
10	My Band	890
11	My Band	890
12	Leave me alone	988
13	Leave me alone	988
14	Natural Disaster	1000
15	Tom Up	1000
16	Caught up	1070
17	Caught up	1070
18	Guilty mind	2980
19	Guilty mind	2980

Query executed successfully. LAPTOP-01M7FIT (15.0 RTM) LAPTOP-01M7FIT\user (66) Library 00:00:00 19 rows

ii) GROUP BY :-

SQLQuery6.sql - LAPTOP-01M7FIT.Library (LAPTOP-01M7FIT\user (70)) - Microsoft SQL Server Management Studio

Object Explorer: LAPTOP-01M7FIT (SQL Server 15.0.2000.)

Query: `select bought_year from books_group by bought_year`
`select no_of_issues from books_group by no_of_issues`

Results:

	bought_year
1	2000
2	2004
3	2012
4	2013
5	2014
6	2015
7	2016

	no_of_issues
1	1
2	4
3	5
4	6
5	9
6	11
7	12
8	13
9	23
10	24
11	29
12	34
13	45
14	67
15	75

Query executed successfully. LAPTOP-01M7FIT (15.0 RTM) LAPTOP-01M7FIT\user (70) Library 00:00:00 23 rows

iii) HAVING :-

SQLQuery6.sql - LAPTOP-01M7FIT.Library (LAPTOP-01M7FIT/user (70)) - Microsoft SQL Server Management Studio

Object Explorer: LAPTOP-01M7FIT (SQL Server 15.0.2000)

Query:

```
select sum(PRICE) from books_ having avg(PRICE)>1000
select sum(no_of_issues) from books_ having max(no_of_issues)>10
```

Results:

(No column name)
396

Messages:

1 (No column name)

Query executed successfully.

Ready In 15 Col 53 Ch 53 INS

4) Use Aggregate function with group by and having

SQLQuery6.sql - LAPTOP-01M7FIT.Library (LAPTOP-01M7FIT/user (70)) - Microsoft SQL Server Management Studio

Object Explorer: LAPTOP-01M7FIT (SQL Server 15.0.2000)

Query:

```
select bought_year, count(author_id) as no_of_books_bought from books_ group by bought_year
select published_year, count(PRICE) as no_of_books_published from books_ group by published_year
```

Results:

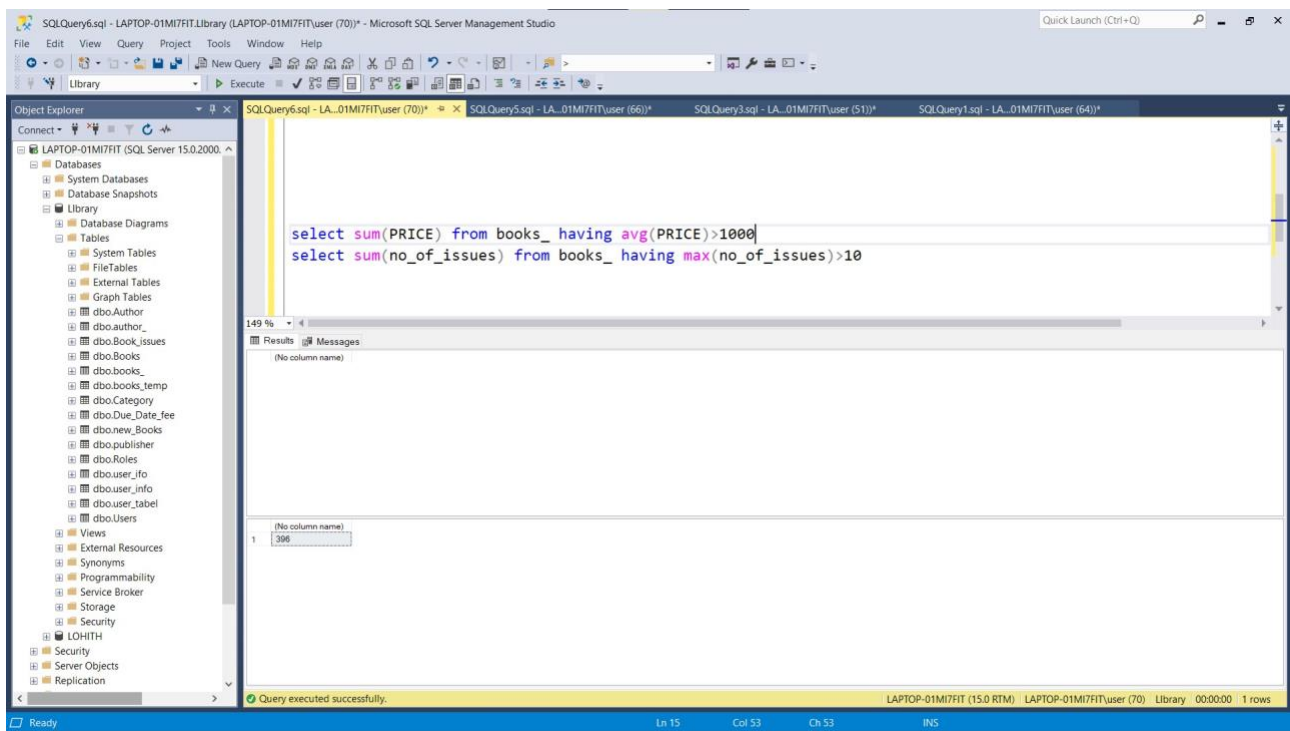
bought_year	no_of_books_bought
2000	1
2004	1
2012	8
2013	1
2014	2
2015	2
2016	2
2020	2

published_year	no_of_books_published
1994	1
1996	2
1998	1
2001	1
2002	1
2010	4
2011	1
2012	2
2013	3
2014	1
2015	1
2016	1

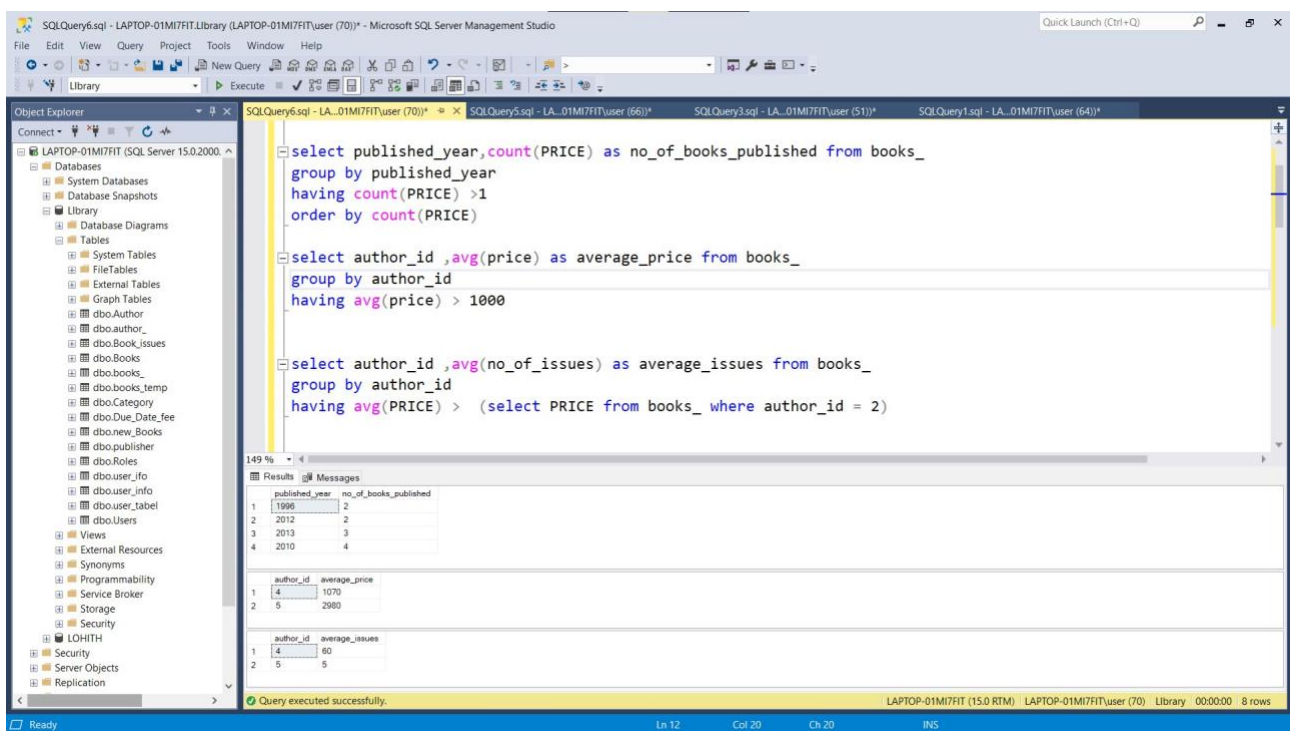
Messages:

Query executed successfully.

Ready In 5 Col 1 Ch 1 INS



5) Write at least 3 nested queries using order by, group by and having clause.



6) Illustrate the Usage of Except, Exists, Not Exists, Union, Intersection

i) EXISTS and NOT EXISTS :-

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'LAPTOP-01M7FIT (SQL Serv...'. The main query window contains the following SQL code:

```
SELECT *
FROM books_
WHERE EXISTS (SELECT author_id FROM books_ WHERE author_id = 3);

SELECT *
FROM books_
WHERE not EXISTS (SELECT author_id FROM books_ WHERE author_id = 3);
```

The Results pane shows the output of the query, displaying a table with columns: ID, BOOK_NAME, author_id, PRICE, barcode_, category, no_of_issues, published_year, and bought_year. The table contains 22 rows of data.

ID	BOOK_NAME	author_id	PRICE	barcode_	category	no_of_issues	published_year	bought_year
1	Natural Disaster	1	1000	2938476	209	12	2010	2012
2	In the Dark	1	567	96567	405	23	2011	2012
3	Caught up	4	1070	89456	708	45	2012	2013
4	Tom Up	2	1000	44567	560	67	2013	2014
5	Close my eyes	8	160	967467	769	34	2014	2014
6	Guilty mind	5	2980	7665246	156	9	2015	2016
7	Leave me alone	6	968	875342346	134	6	2016	2016
8	My Band	7	890	764563	122	5	2010	2012
9	Fantasy Girl	8	799	432653	342	4	2001	2012
10	No Flaws	15	679	45653	564	11	2002	2004
11	Drop It	10	420	456746	908	24	2012	2012
12	Caught up	4	1070	456358	879	75	1994	2012
13	Close my eyes	8	160	87657	657	29	1998	2000
14	Guilty mind	5	2980	456354	123	1	2013	2015
15	Leave me alone	6	968	8563425	234	13	2010	2012
16	My Band	7	890	65476	323	12	1996	2020
17	Fantasy Girl	8	799	79676	567	12	1996	2020
20	No Flaws	9	679	345476	343	13	2010	2012
22	Drop It	10	450	456358	919	1	2013	2015

ii) EXCEPT :-

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'LAPTOP-01M7FIT (SQL Serv...'. The main query window contains the following SQL code:

```
select author_name from author_ except (select author_name from author_ where no_of_books>2)
```

The Results pane shows the output of the query, displaying a table with columns: author_name. The table contains 4 rows of data.

author_name
sharda
mail
rio
so_hail

~ iii) UNION and INTERSECT

:-

SQLQuery6.sql - LAPTOP-01M7FIT.Library (LAPTOP-01M7FIT\user (70))*

```

select author_name from author_ where author_id in
((select author_id from books_ where author_id > 2)
union
(select author_id from author_ where author_id < 2))

select author_name from author_ where author_id in
((select author_id from books_ where author_id > 2)
intersect
(select author_id from author_ where author_id < 2))

```

Results

author_name
greenhine
no
maili
chandra
lohihi
Bhanupriya
Lohih
Koushik
Suman

Query executed successfully.

~

7) INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN- 3 queries for each instance

I) JOIN :-

SQLQuery1.sql - LAPTOP-01M7FIT.Library (LAPTOP-01M7FIT\user (57))*

```

SELECT BOOK_NAME,books_.author_id from books_
join author_ on author_.author_id = books_.no_of_issues

```

Results

BOOK_NAME	author_id
Natural Disaster	1
Guilty mind	5
Leave me alone	6
My Band	7
Fantasy Girl	8
No Flaws	15
Guilty mind	5
Leave me alone	6
My Band	7
Fantasy Girl	8
No Flaws	9
Drop It	10
ALONE	9
SOLO	9

Query executed successfully.

~

SQLQuery1.sql - LAPTOP-01M7FIT.Library (LAPTOP-01M7FIT\user (57)) - Microsoft SQL Server Management Studio

Object Explorer: LAPTOP-01M7FIT (SQL Server) > Databases > Books_ > Tables > books_

```

SELECT books_.author_id ,COUNT(BOOK_NAME) from books_
JOIN author_ on author_.author_id = books_.author_id GROUP by books_.author_id

```

Results (No column name):

author_id	Count
1	2
2	1
3	4
4	2
5	2
6	2
7	2
8	4
9	3
10	2
15	1

Query executed successfully. LAPTOP-01M7FIT (15.0 RTM) LAPTOP-01M7FIT\user (57) Library 00:00:00 10 rows

SQLQuery1.sql - LAPTOP-01M7FIT.Library (LAPTOP-01M7FIT\user (57)) - Microsoft SQL Server Management Studio

Object Explorer: LAPTOP-01M7FIT (SQL Server) > Databases > Books_ > Tables > books_

```

SELECT BOOK_NAME,COUNT(BOOK_NAME) from books_
JOIN author_ on author_.author_id = books_.author_id GROUP by BOOK_NAME

```

Results (No column name):

BOOK_NAME	Count
ALONE	1
Caught up	2
Close my eyes	2
Drop It	2
Fantasy Girl	2
Guilty mind	2
In the Dark	1
Leave me alone	2
My Band	2
Natural Disaster	1
No Flaws	2
SOLO	1
Torn Up	1

Query executed successfully. LAPTOP-01M7FIT (15.0 RTM) LAPTOP-01M7FIT\user (57) Library 00:00:00 13 rows

ii) RIGHT JOIN :-

SQLQuery1.sql - LAPTOP-01MI7FIT.Library (LAPTOP-01MI7FIT/user (57)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer

Connect LAPTOP-01MI7FIT (SQL Server)

Databases Security Server Objects Replication PolyBase Always On High Availability Management Integration Services Catalog SQL Server Agent (Agent XP) XEvent Profiler

SQLQuery1.sql - LA_01MI7FIT/user (57))

```
SELECT PRICE,avg(PRICE) from books_
right JOIN author_ on author_.author_id = books_.author_id GROUP by PRICE
```

Results Messages

PRICE (No column name)
1 NULL NULL
2 160 160
3 450 450
4 567 567
5 679 679
6 799 799
7 890 890
8 988 988
9 1000 1000
10 1070 1070
11 2980 2980
12 10001 10001

Query executed successfully.

LAPTOP-01MI7FIT (15.0 RTM) LAPTOP-01MI7FIT/user (57) Library 00:00:00 12 rows

SQLQuery1.sql - LAPTOP-01MI7FIT.Library (LAPTOP-01MI7FIT/user (57)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer

Connect LAPTOP-01MI7FIT (SQL Server)

Databases Security Server Objects Replication PolyBase Always On High Availability Management Integration Services Catalog SQL Server Agent (Agent XP) XEvent Profiler

SQLQuery1.sql - LA_01MI7FIT/user (57))

```
SELECT author_.author_name,books_.BOOK_NAME from books_
right JOIN author_ on author_.author_id= books_.author_id
```

Results Messages

author_name	BOOK_NAME
1 greeshma	Natural Disaster
2 greeshma	In the Dark
3 koushik	Torn Up
4 so_hail	NULL
5 rio	Caught up
6 rio	Caught up
7 mali	Guilty mind
8 mali	Guilty mind
9 chandra	Leave me alone
10 chandra	Leave me alone
11 lohith	My Band
12 lohith	My Band
13 Bhanupriya	Close my eyes
14 Bhanupriya	Fantasy Girl
15 Bhanupriya	Close my eyes
16 Bhanupriya	Fantasy Girl
17 Lohith	No Flaws
18 Lohith	ALONE
19 Lohith	SOLO
20 Koushik	Drop It
21 Koushik	Drop It
22 Sohail	NULL
23 Greeshma	NULL
24 Malikarjun	NULL
25 Pavan	NULL
26 Suman	No Flaws
27 Priya	NULL
28 Vishwa	NULL
29 Geetha	NULL

Query executed successfully.

LAPTOP-01MI7FIT (15.0 RTM) LAPTOP-01MI7FIT/user (57) Library 00:00:00 29 rows

SQLQuery1.sql - LAPTOP-01M17FIT.Library (LAPTOP-01M17FIT\user (57)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Library

Object Explorer

Connect LAPTOP-01M17FIT (SQL Server)

Databases Security Server Objects Replication PolyBase Always On High Availability Management Integration Services Catalog SQL Server Agent (Agent XP) XEvent Profiler

SQLQuery1.sql - LA_01M17FIT\user (57))

```

SELECT author_.author_name, COUNT(BOOK_NAME) FROM books_
RIGHT JOIN author_ ON author_.author_id = books_.author_id GROUP BY author_.author_name

```

Results Messages

author_name (No column name)

1	Bhanupriya	4
2	chandra	2
3	Geeha	0
4	greeshma	2
5	Koushik	3
6	lohih	5
7	naili	2
8	Malikarjun	0
9	Pavan	0
10	Piya	0
11	rio	2
12	so_hail	0
13	Sohail	0
14	Suman	1
15	Vishwa	0

Query executed successfully.

LAPTOP-01M17FIT (15.0 RTM) LAPTOP-01M17FIT\user (57) Library 00:00:00 15 rows

Ready Ln 116 Col 88 Ch 88 INS

~ iii) LEFT JOIN :-

SQLQuery1.sql - LAPTOP-01M17FIT.Library (LAPTOP-01M17FIT\user (57)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Library

Object Explorer

Connect LAPTOP-01M17FIT (SQL Server)

Databases Security Server Objects Replication PolyBase Always On High Availability Management Integration Services Catalog SQL Server Agent (Agent XP) XEvent Profiler

SQLQuery1.sql - LA_01M17FIT\user (57))

```

SELECT BOOK_NAME, COUNT(BOOK_NAME) FROM books_
LEFT JOIN author_ ON author_.author_id = books_.author_id GROUP BY BOOK_NAME

```

Results Messages

BOOK_NAME (No column name)

1	ALONE	1
2	Caught up	2
3	Close my eyes	2
4	Drop It	2
5	Fantasy Girl	2
6	Guilty mind	2
7	In the Dark	1
8	Leave me alone	2
9	My Band	2
10	Natural Disaster	1
11	No Flaws	2
12	SOLO	1
13	Torn Up	1

Query executed successfully.

LAPTOP-01M17FIT (15.0 RTM) LAPTOP-01M17FIT\user (57) Library 00:00:00 13 rows

Ready Ln 129 Col 1 Ch 1 INS

~

SQLQuery1.sql - LAPTOP-01M17FIT.Library (LAPTOP-01M17FIT\user (57)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Library

Object Explorer

Connect LAPTOP-01M17FIT (SQL Server)

- Databases
- Security
- Server Objects
- Replication
- PolyBase
- Always On High Availability
- Management
- Integration Services Catalog
- SQL Server Agent (Agent XP)
- XEvent Profiler

```
SELECT author_.author_name,books_.BOOK_NAME from books_
left JOIN author_ on author_.author_id= books_.author_id
```

165 %

Results Messages

	author_name	BOOK_NAME
1	greeshma	Natural Disaster
2	greeshma	In the Dark
3	rio	Caught up
4	koushik	Torn Up
5	Bhanupriya	Close my eyes
6	malli	Guilty mind
7	chandra	Leave me alone
8	lohith	My Band
9	Bhanupriya	Fantasy Girl
10	Suman	No Flaws
11	Koushik	Drop It
12	rio	Caught up
13	Bhanupriya	Close my eyes
14	malli	Guilty mind
15	chandra	Leave me alone
16	lohith	My Band
17	Bhanupriya	Fantasy Girl
18	Lohith	No Flaws
19	Koushik	Drop It
20	Lohith	ALONE
21	Lohith	SOLO

Query executed successfully.

LAPTOP-01M17FIT (15.0 RTM) LAPTOP-01M17FIT\user (57) Library 00:00:00 21 rows

Ready In 139 Col 1 Ch 1 INS

~

SQLQuery1.sql - LAPTOP-01M7FIT.Library (LAPTOP-01M7FITUser (57)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Library Execute New Query

Object Explorer

- Connect
- LAPTOP-01M7FIT (SQL Server)
 - Databases
 - Security
 - Server Objects
 - Replication
 - PolyBase
 - Always On High Availability
 - Management
 - Integration Services Catalog
 - SQL Server Agent (Agent XP)
 - XEvent Profiler

SQLQuery1.sql - LA...01M7FITUser (57) *

```
SELECT author_.author_name, count(BOOK_NAME) from books_
left JOIN author_ on author_.author_id= books_.author_id group by author_.author_name
having avg(books_.PRICE) > 100
```

165 %

Results Messages

	author_name	(No column name)
1	Shanupriya	4
2	chandra	2
3	greeshma	2
4	Koushik	3
5	Lalith	5
6	mali	2
7	no	2
8	Suman	1

Query executed successfully.

LAPTOP-01M7FIT (15.0 RTM) LAPTOP-01M7FITUser (57) Library 00:00:00 8 rows

Ready Ln 147 Col 15 Ch 15 INS