

DBMS LAB ASSIGNMENT-5

Name:- MATHANGI SRAVAN

Roll no:-19BCS068

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 Management Studio interface. The query editor 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 'l%'
```

The Results pane displays the output of these queries:

BOOK_NAME
1 Guilty mind
2 Guilty mind

BOOK_NAME
1 Caught up
2 Tom Up
3 Caught up

author_name
1 lohith
2 Lohith

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

~ ii) ALL

:-

The first screenshot shows a SQL query in the query editor:

```
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_)
```

The results pane shows two tables:

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

author_id	author_name
16	Piya
17	Vishwa
18	Geetha

The second screenshot shows a different SQL query:

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

The results pane shows a single table:

author_id	author_name
6	Shandha
7	Ishith
8	Bhanupriya
9	Lohith
10	Kaushik
11	Sohail
12	Greeshma
13	Malikarjun
14	Prasen
15	Suman
16	Piya
17	Vishwa
18	Geetha

~

iii) ANY :-

SQLQuery1.sql - LAPTOP-01M7FITLibrary (LAPTOP-01M7FITUser (S7)) - Microsoft SQL Server Management Studio

```

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)

```

165 %

Results Messages

no_of_issues

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

Query executed successfully.

LAPTOP-01M7FIT (15.0 RTM) LAPTOP-01M7FITUser (S7) Library 00:00:00 12 rows

~

SQLQuery1.sql - LAPTOP-01M7FITLibrary (LAPTOP-01M7FITUser (S7)) - Microsoft SQL Server Management Studio

```

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

```

165 %

Results Messages

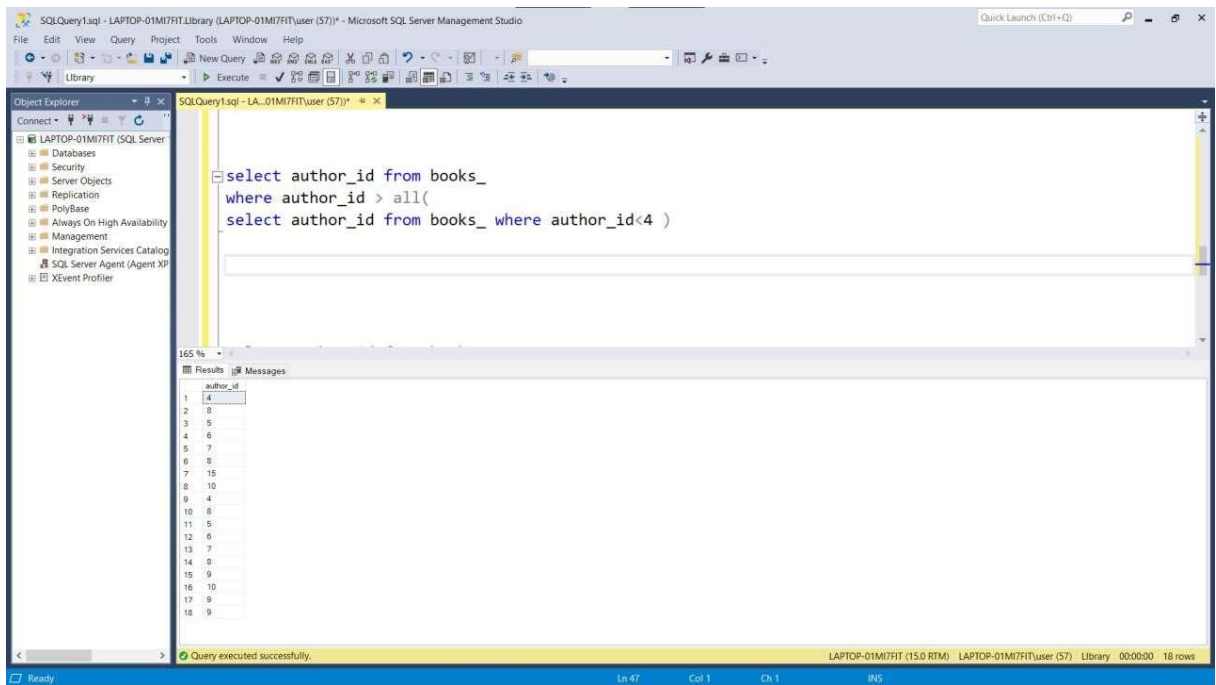
author_id	BOOK_NAME
8	Close my eyes
8	Fantasy Girl
15	No Flares
10	Drop It
8	Close my eyes
8	Fantasy Girl
9	No Flares
10	Drop It
9	ALONE
9	SOLO

Query executed successfully.

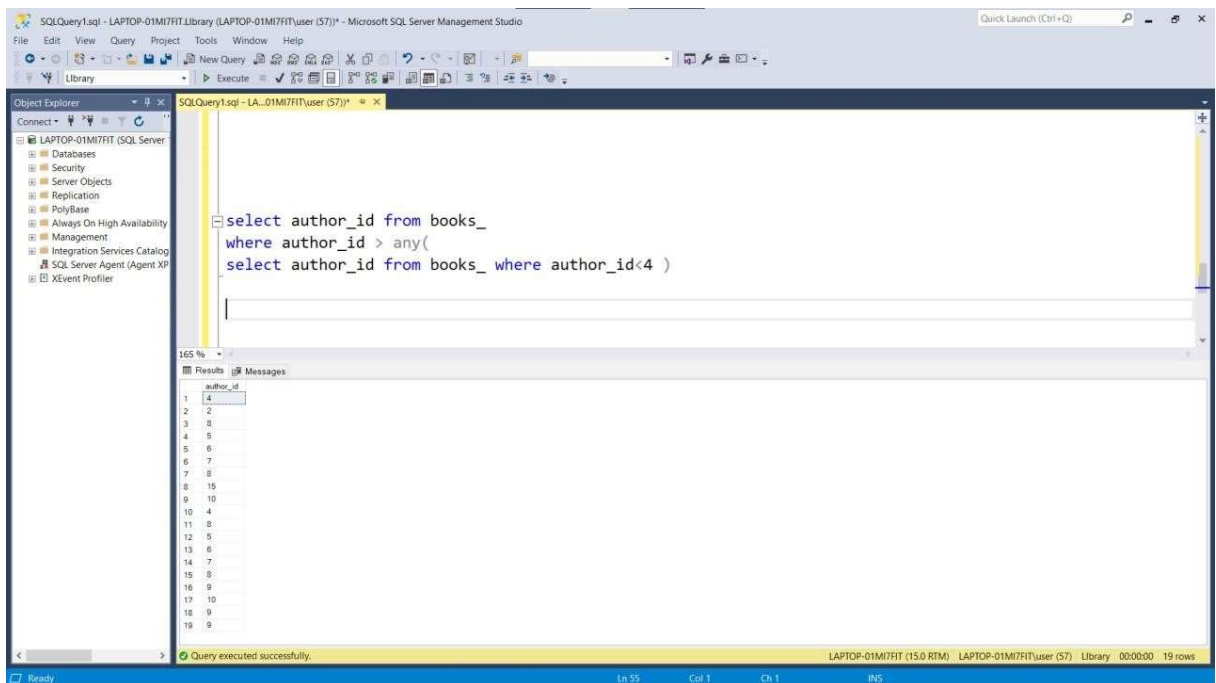
LAPTOP-01M7FIT (15.0 RTM) LAPTOP-01M7FITUser (S7) Library 00:00:00 10 rows

~ iv) ANY and ALL comparison

:-



~



~

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-01M7HT' (SQL Server 15.0.2000). The 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 these queries:

Query	Result
select sum(PRICE) from books_	18599
select avg(PRICE) from books_ where author_id > 2	1002
select count(no_of_issues) from books_ where author_id > 4	14
select max(PRICE) from books_ where author_id > 6	890
select min(no_of_issues) from books_ where author_id > 8	1

The status bar at the bottom indicates 'Query executed successfully.' and 'LAPTOP-01M7HT (15.0 RTM) LAPTOP-01M7HTUser (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 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 this query, sorted by published_year in descending order:

BOOK_NAME	published_year
Leave me alone	2016
Guilty mind	2015
Close my eyes	2014
Turn 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-01M7HT (15.0 RTM) LAPTOP-01M7HTUser (66) Library 00:00:00 19 rows'.

~

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The 'Object Explorer' on the left displays the database structure. The 'Query' window in the center contains the following SQL query:

```
use Library
select BOOK_NAME,PRICE from books_ order by PRICE
```

The 'Results' pane on the right displays the output of the query, showing 19 rows of data. The status bar at the bottom indicates 'Query executed successfully.' and '19 rows'.

BOOK_NAME	PRICE
Close my eyes	160
Close my eyes	160
Drop It	450
Drop It	450
In the Dark	567
No Flaws	679
No Flaws	679
Lease me alone	799
Lease me alone	799
My Band	890
My Band	890
Lease me alone	908
Lease me alone	908
Natural Disaster	1000
Turn Up	1000
Caught up	1070
Caught up	1070
Quality mind	2060
Quality mind	2060

ii)GROUP BY :-

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The 'Query' window in the center contains the following SQL queries:

```
select bought_year from books_ group by bought_year
select no_of_issues from books_ group by no_of_issues
```

The 'Results' pane on the right displays the output of the queries, showing 15 rows of data. The status bar at the bottom indicates 'Query executed successfully.' and '23 rows'.

bought_year
2000
2004
2012
2013
2014
2015
2016

no_of_issues
1
4
5
6
9
11
12
13
23
24
29
34
45
67
75

iii) HAVING :-

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

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

The Results pane at the bottom shows the output of the query, which is a single row with the value 398.

(No column name)
398

The status bar at the bottom indicates "Query executed successfully." and "LAPTOP-01M7FIT (15.0 RTM) LAPTOP-01M7FIT/user (70) Library 00:00:00 1 rows".

4) Use Aggregate function with group by and having

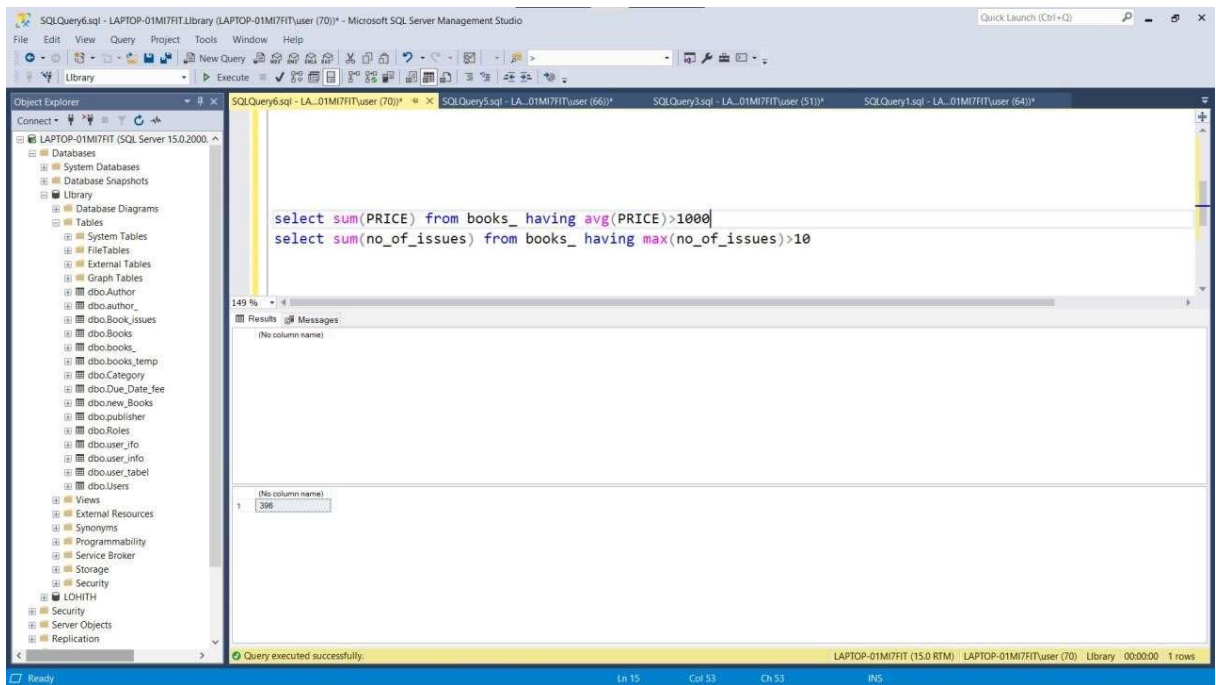
The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure. The central query window contains the following SQL code:

```
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
```

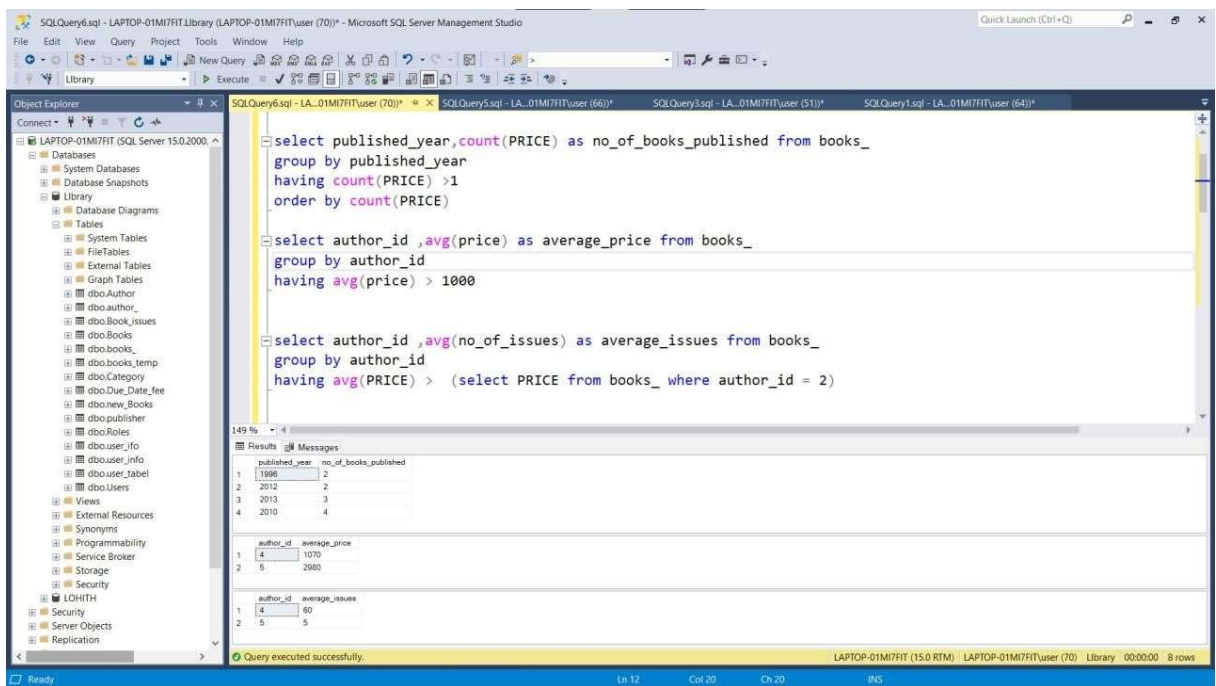
The Results pane at the bottom shows the output of the query, which is a table with two columns: bought_year and no_of_books_bought. The table contains 8 rows of data.

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

The status bar at the bottom indicates "Query executed successfully." and "LAPTOP-01M7FIT (15.0 RTM) LAPTOP-01M7FIT/user (70) Library 00:00:00 20 rows".



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 Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'LAPTOP-01M7FIT (SQL Serv)'. The central 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 at the bottom shows the execution results for the first query. The status bar indicates 'Query executed successfully.' and '19 rows'.

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	95957	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	988	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	450	456748	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	988	8563425	234	13	2010	2012
16	My Band	7	890	65476	323	12	1996	2020
17	Fantasy Girl	8	799	79678	567	12	1996	2020
18	No Flaws	9	679	345476	343	13	2010	2012
19	Drop It	10	450	456358	919	1	2013	2015

~ ii) EXCEPT

:-

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'LAPTOP-01M7FIT (SQL Serv)'. The central 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 at the bottom shows the execution results for the query. The status bar indicates 'Query executed successfully.' and '4 rows'.

author_name
chundia
mai
no
so_hail

~ iii) UNION and INTERSECT

:-

SQLQuery6.sql - LAPTOP-01M7FIT Library (LAPTOP-01M7FITUser (70)) - Microsoft SQL Server Management Studio

```

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
1
2
3
4
5
6
7
8
9

author_name

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-01M7FITUser (57)) - Microsoft SQL Server Management Studio

```

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

```

Results

BOOK_NAME	author_id
1	1
2	5
3	6
4	7
5	8
6	15
7	5
8	6
9	7
10	8
11	9
12	10
13	9
14	9

Query executed successfully.

~

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

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

The Results pane at the bottom shows the output of the query, which is a table with two columns: author_id and an unlabeled column for the count. The data is as follows:

author_id	(No column name)
1	2
2	1
3	4
4	2
5	2
6	2
7	2
8	4
9	3
10	2
15	1

The status bar at the bottom indicates "Query executed successfully." and "10 rows".

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

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

The Results pane at the bottom shows the output of the query, which is a table with two columns: BOOK_NAME and an unlabeled column for the count. The data is as follows:

BOOK_NAME	(No column name)
ALONE	1
Caught up	2
Close my eyes	2
Drop It	2
Fantasy Girl	2
Guiltymind	2
In the Dark	1
Leave me alone	2
My Band	2
Natural Disaster	1
No Flaws	2
SOLD	1
Tom Up	1

The status bar at the bottom indicates "Query executed successfully." and "13 rows".

ii) RIGHT JOIN :-

SQLQuery1.sql - LAPTOP-01M7F7IT\Library (LAPTOP-01M7F7IT\User (57)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Library

Object Explorer

Connect

LAPTOP-01M7F7IT (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...01M7F7IT\User (57))

```

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

```

165 %

Results Messages

PRICE (No column name)
1
2
3
4
5
6
7
8
9
10
11
12

Query executed successfully.

LAPTOP-01M7F7IT (15.0 RTM) LAPTOP-01M7F7IT\User (57) Library 00:00:00 12 rows

Ready In 99 Col 1 Ch 1 INS

SQLQuery1.sql - LAPTOP-01M7F7IT\Library (LAPTOP-01M7F7IT\User (57)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Library

Object Explorer

Connect

LAPTOP-01M7F7IT (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...01M7F7IT\User (57))

```

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

```

165 %

Results Messages

author_name	BOOK_NAME
greeshma	Natural Disaster
greeshma	In the Dark
koushik	Tom Up
so_hai	NULL
no	Caught up
no	Caught up
maili	Gully mind
maili	Gully mind
chandra	Leave me alone
chandra	Leave me alone
lohit	My Band
lohit	My Band
Bhanupriya	Close my eyes
Bhanupriya	Fantasy Girl
Bhanupriya	Close my eyes
Bhanupriya	Fantasy Girl
Lohith	No Flaws
Lohith	ALONE
Lohith	SOLD
Koushik	Drop It
Koushik	Drop It
Sohail	NULL
Greeshma	NULL
Malikarjun	NULL
Pavan	NULL
Suman	No Flaws
Priya	NULL
Vishwa	NULL
Geetha	NULL

Query executed successfully.

LAPTOP-01M7F7IT (15.0 RTM) LAPTOP-01M7F7IT\User (57) Library 00:00:00 29 rows

Ready In 103 Col 1 Ch 1 INS

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

```
SELECT author_.author_name, count(BOOK_NAME) from books_
right JOIN author_ on author_.author_id= books_.author_id group by author_.author_name
```

The Results pane at the bottom displays the output of the query, showing 15 rows of data. The status bar at the bottom indicates "Query executed successfully." and "15 rows".

author_name	count
1. Bhanupriya	4
2. chandni	2
3. Geetha	0
4. greeshma	2
5. Koushik	3
6. Ishith	5
7. mali	2
8. Malikaagun	0
9. Pavan	0
10. Priya	0
11. no	2
12. no_hall	0
13. Sohail	0
14. Suman	1
15. Vishwa	0

~ iii) LEFT JOIN :-

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

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

The Results pane at the bottom displays the output of the query, showing 13 rows of data. The status bar at the bottom indicates "Query executed successfully." and "13 rows".

BOOK_NAME	count
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. S.O.D.	1
13. Torn Up	1

~

SQLQuery1.sql - LAPTOP-01M7F7IT\Library (LAPTOP-01M7F7IT\User (57)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Library

Object Explorer

Connect

LAPTOP-01M7F7IT (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 - LAPTOP-01M7F7IT\User (57)

```
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	no	Caught up
4	Koushik	Turn Up
5	Bhanupriya	Close my eyes
6	maili	Guilty mind
7	chandra	Leave me alone
8	lohith	My Band
9	Bhanupriya	Fantasy Girl
10	Suman	No Flaws
11	Koushik	Drop It
12	no	Caught up
13	Bhanupriya	Close my eyes
14	maili	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-01M7F7IT (15.0 RTM) LAPTOP-01M7F7IT\User (57) Library 00:00:00 21 rows

Ready In 139 Col 1 Ch 1 INS

~

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

File Edit View Query Project Tools Window Help

Library Execute

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

```
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
```

Results Messages

	author_name	(No column name)
1	Bhanupriya	4
2	chandra	2
3	greshma	2
4	Koushik	3
5	Lohith	5
6	mali	2
7	ro	2
8	Suman	1

Query executed successfully.

LAPTOP-01M7FIT (15.0 RTM) | LAPTOP-01M7FIT\User (57) | Library | 00:00:00 | 8 rows

Ready | Ln 147 | Col 15 | Ch 15 | RS