

## GROUP – 7

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 :-

```
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%'
```

BOOK_NAME
Guilty mind
Guilty mind

BOOK_NAME
Caught up
Tom Up
Caught up

author_name
Lohith
Lohith

Query executed successfully.

~

ii) ALL :-

```
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_)
```

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.

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

```

165 %

Results Messages

	author_id	author_name
1	6	chandra
2	7	lohih
3	8	Bhanupriya
4	9	Lohih
5	10	Koushik
6	11	Sohail
7	12	Greeshma
8	13	Malikarjun
9	14	Pavan
10	15	Suman
11	16	Priya
12	17	Vishwa
13	18	Geetha

Query executed successfully.

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

Ready Ln 21 Col 10 Ch 10 INS

iii) ANY :-

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 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
1	8
2	7
3	8
4	15
5	10
6	8
7	7
8	8
9	9
10	10
11	9
12	9

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

Query executed successfully.

LAPTOP-01M17FIT (15.0 RTM) LAPTOP-01M17FIT(user (57)) Library 00:00:00 12 rows

Ready Ln 31 Col 18 Ch 18 INS

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

Object Explorer: LAPTOP-01M17FIT (SQL Server) > Databases > Books

```

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
5	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.

iv) ANY and ALL comparison :-

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

Object Explorer: LAPTOP-01M17FIT (SQL Server) > Databases > Books

```

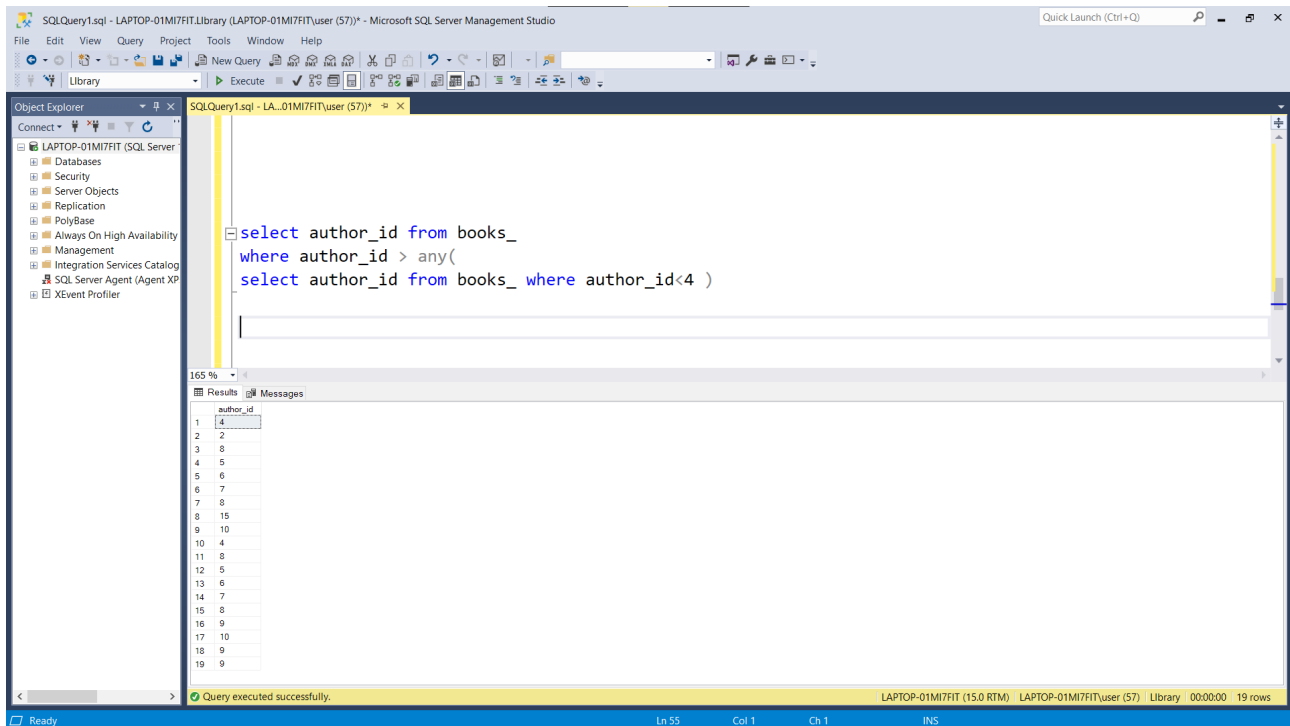
select author_id from books_
where author_id > all(
select author_id from books_ where author_id<4 )

```

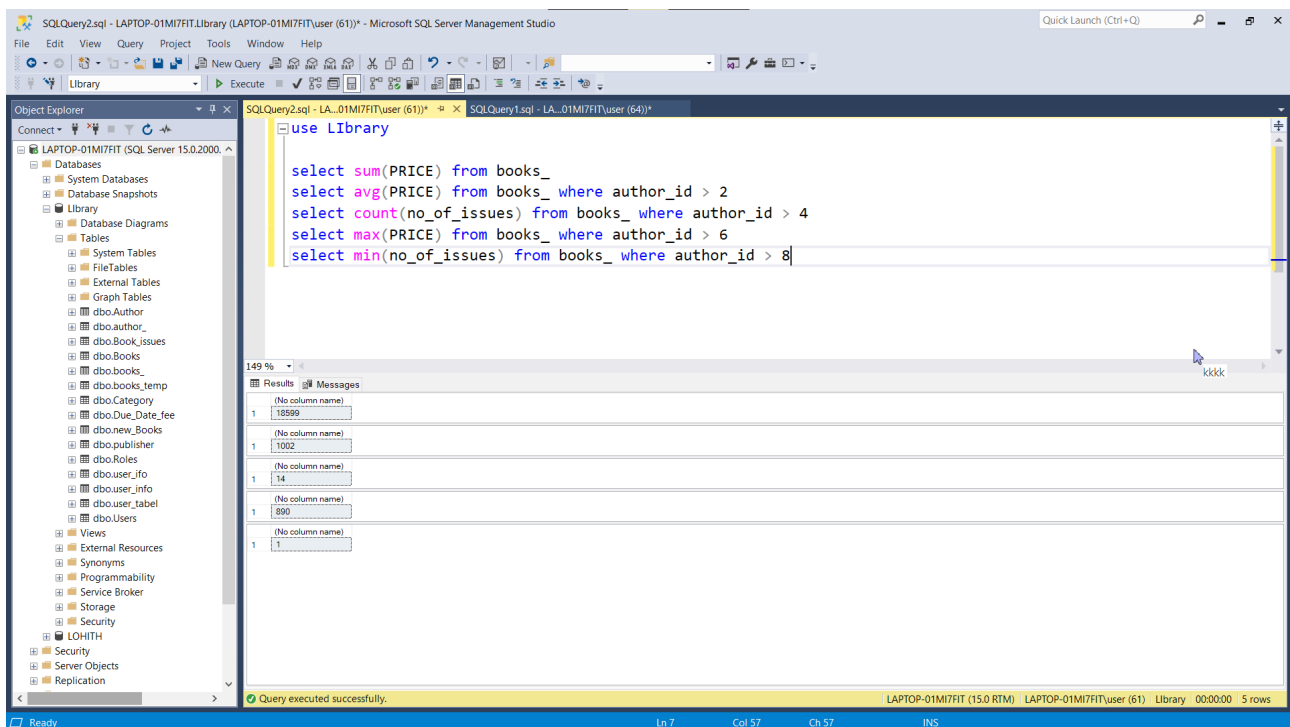
Results:

author_id
4
8
5
6
7
8
15
10
4
8
11
6
7
8
9
10
9
9

Query executed successfully.

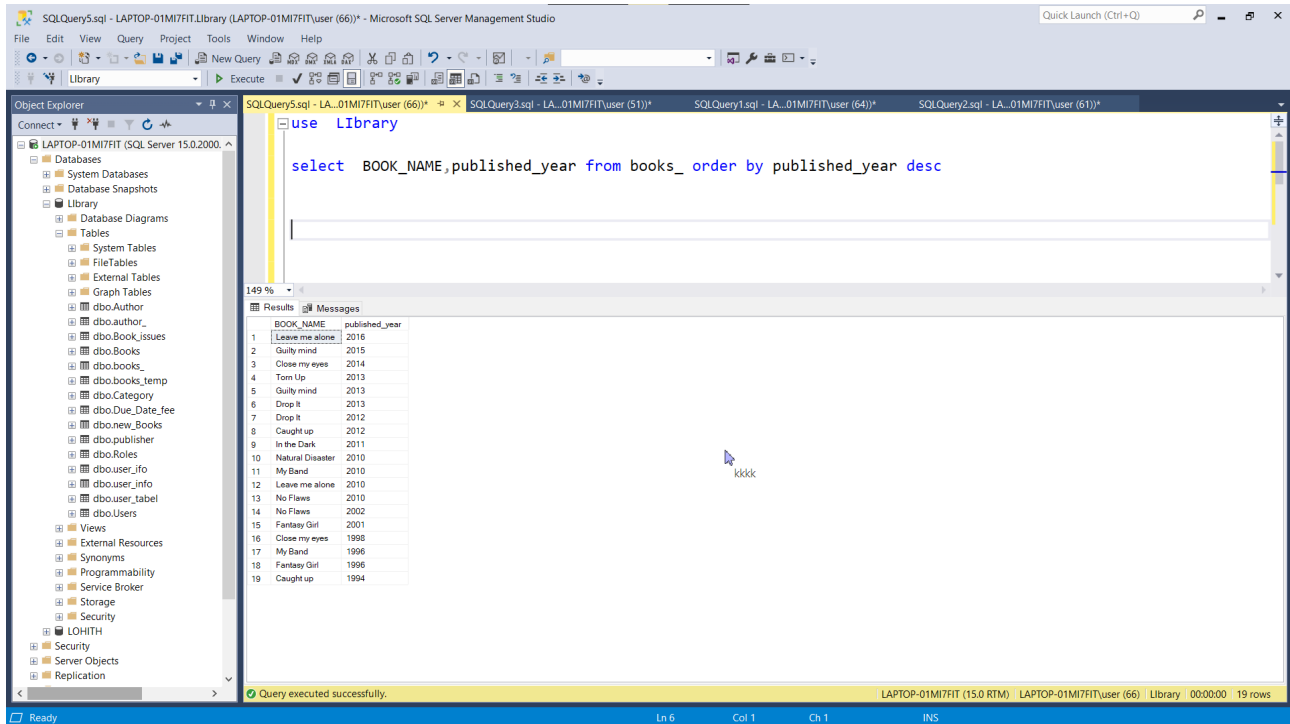


2). One query for each Aggregate function.



3)  
Illustrate the usage of order by, group by and having clause (2 queries for each case)

i)ORDER BY :-



SQLQuery5.sql - LAPTOP-01M17FIT.Library (LAPTOP-01M17FIT(user (66))) - Microsoft SQL Server Management Studio

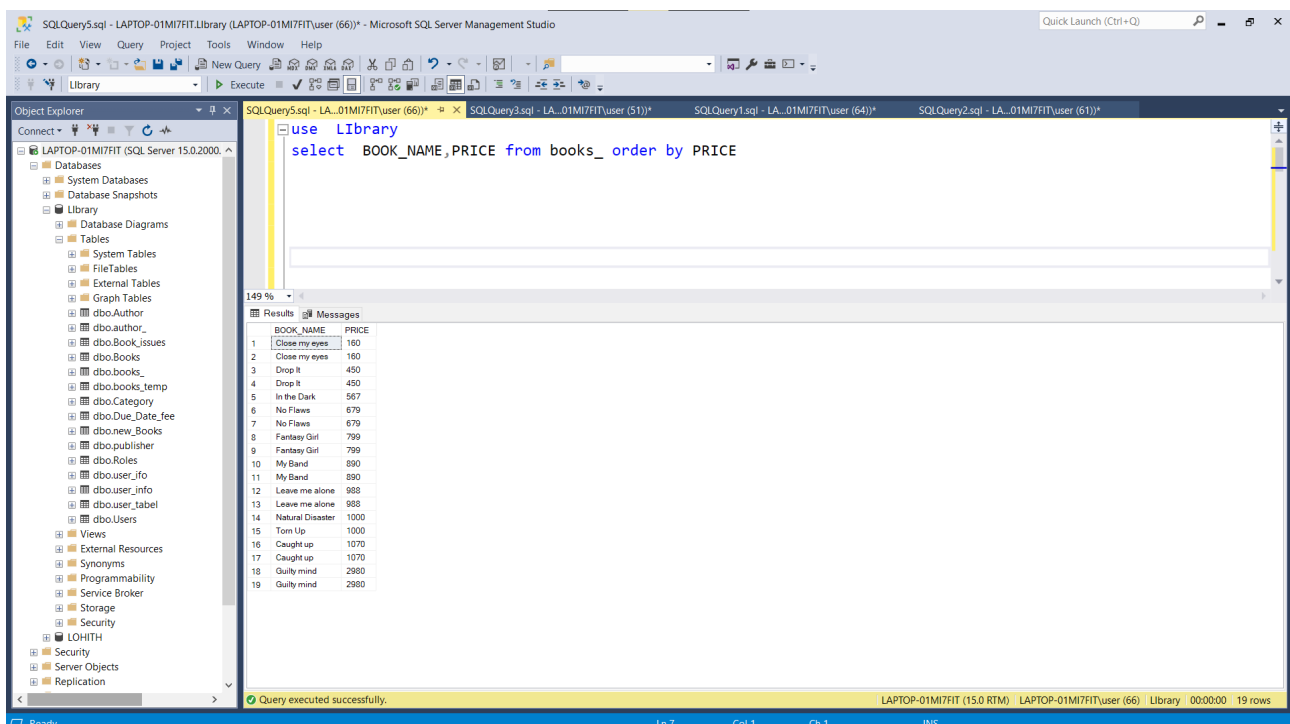
use Library

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

Results

	BOOK_NAME	published_year
1	Leave me alone	2016
2	Guilty mind	2015
3	Close my eyes	2014
4	Tom Up	2013
5	Guilty mind	2013
6	Drop It	2013
7	Drop It	2012
8	Caught up	2012
9	In the Dark	2011
10	Natural Disaster	2010
11	My Band	2010
12	Leave me alone	2010
13	No Flaws	2010
14	No Flaws	2002
15	Fantasy Girl	2001
16	Close my eyes	1998
17	My Band	1996
18	Fantasy Girl	1996
19	Caught up	1994

Query executed successfully.



SQLQuery5.sql - LAPTOP-01M17FIT.Library (LAPTOP-01M17FIT(user (66))) - Microsoft SQL Server Management Studio

use Library

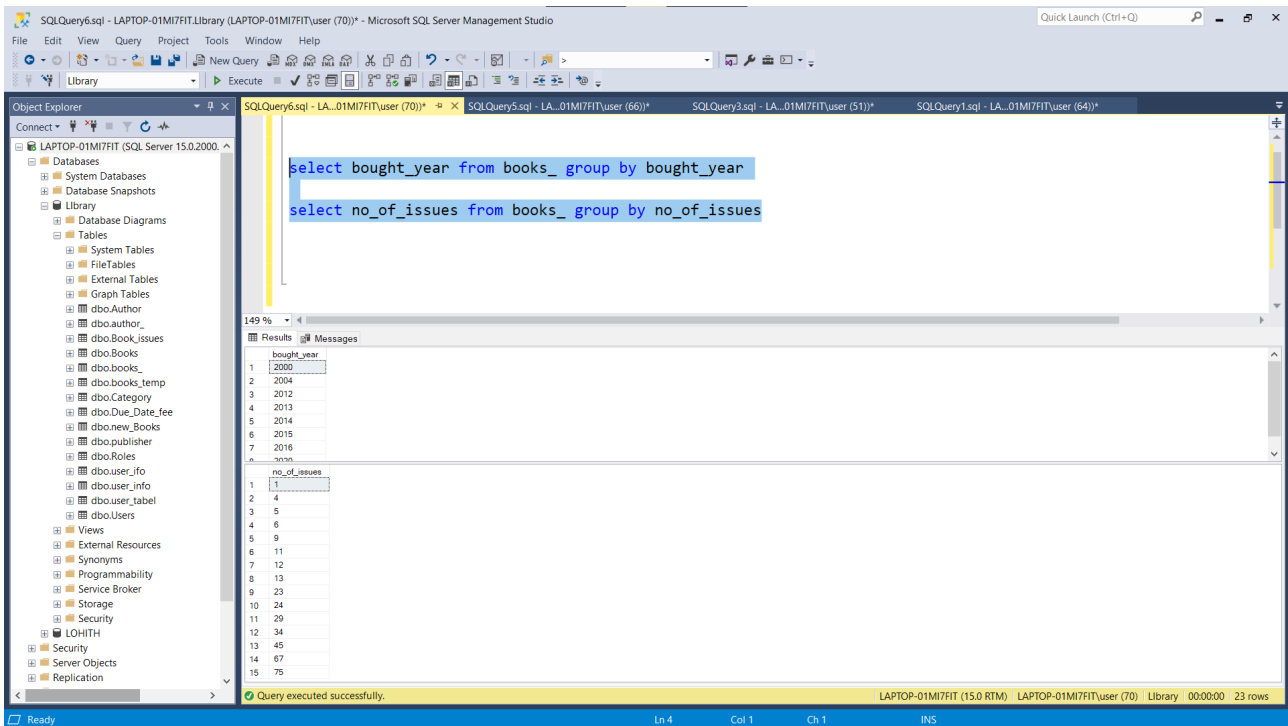
```
select BOOK_NAME,PRICE from books_ order by PRICE
```

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	980
13	Leave me alone	980
14	Natural Disaster	1000
15	Tom Up	1000
16	Caught up	1070
17	Caught up	1070
18	Guilty mind	2900
19	Guilty mind	2900

Query executed successfully.

## ii) GROUP 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 two 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 shows the output of the first query, which is a list of years:

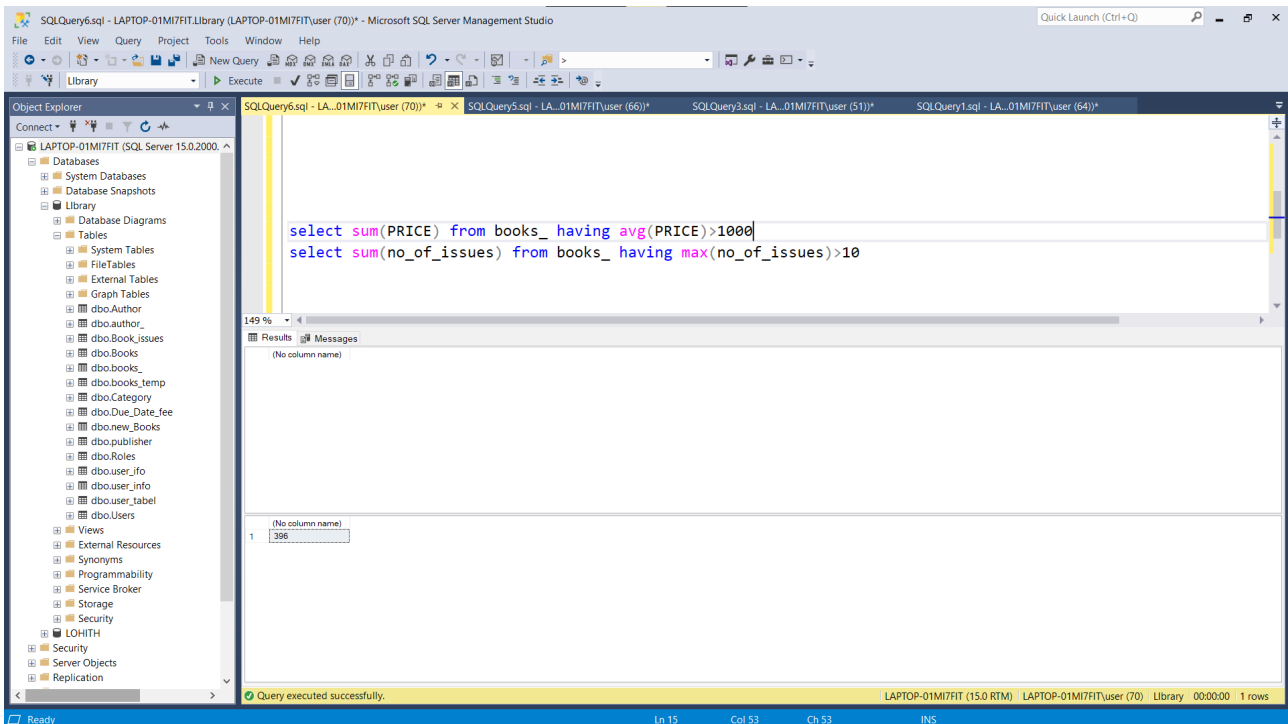
bought_year
2000
2004
2012
2013
2014
2015
2016

The Results pane also shows the output of the second query, which is a list of issue counts:

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

The status bar at the bottom indicates 'Query executed successfully.' and 'LAPTOP-01M7FIT (15.0 RTM) LAPTOP-01M7FIT(user (70)) Library 00:00:00 23 rows'.

## iii) HAVING :-



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 two SQL queries:

```
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 shows the output of the first query, which is a single row with a sum of prices:

(No column name)
396

The Results pane also shows the output of the second query, which is a single row with a sum of issue counts:

(No column name)
396

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 Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'LAPTOP-01MI7FIT (SQL Server 15.0.2000.0)'. The main query window contains two SQL queries. The first query uses 'group by' and 'count' to aggregate data from the 'books\_' table. The second query uses 'group by' and 'count' to aggregate data from the 'books\_' table. The Results pane shows the output of the first query, displaying two tables: one for books bought by year and one for books published by year.

```
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
```

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

Query executed successfully.

~

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'LAPTOP-01MI7FIT (SQL Server 15.0.2000.0)'. The main query window contains two SQL queries. The first query uses 'having' and 'avg' to filter books based on their average price. The second query uses 'having' and 'max' to filter books based on their maximum number of issues. The Results pane shows the output of the first query, displaying a single row with the sum of prices for books having an average price greater than 1000.

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

(No column name)
396

Query executed successfully.

~

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

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'LAPTOP-01M17FIT'. The main query window contains three SQL queries:

```

-- Query 1
select published_year, count(PRICE) as no_of_books_published from books_
group by published_year
having count(PRICE) > 1
order by count(PRICE)

-- Query 2
select author_id , avg(price) as average_price from books_
group by author_id
having avg(price) > 1000

-- Query 3
select author_id , avg(no_of_issues) as average_issues from books_
group by author_id
having avg(PRICE) > (select PRICE from books_ where author_id = 2)
  
```

The Results pane shows the output of the first query:

published_year	no_of_books_published
1998	2
2012	2
2013	3
2010	4

The Results pane also shows the output of the second query:

author_id	average_price
4	1070
5	2980

The Results pane also shows the output of the third query:

author_id	average_issues
4	60
5	5

The status bar at the bottom indicates 'Query executed successfully.' and 'LAPTOP-01M17FIT (15.0 RTM) LAPTOP-01M17FIT(user (70)) Library 00:00:00 8 rows'.

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-01M17FIT'. The main query window contains two SQL queries:

```

-- Query 1
SELECT *
FROM books_
WHERE EXISTS (SELECT author_id FROM books_ WHERE author_id = 3);

-- Query 2
SELECT *
FROM books_
WHERE NOT EXISTS (SELECT author_id FROM books_ WHERE author_id = 3);
  
```

The Results pane shows the output of the first query:

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	98567	405	23	2011	2012
3	Caught up	4	1070	89456	708	45	2012	2013
4	Torn Up	2	1000	44567	560	67	2013	2014
5	Close my eyes	8	160	967487	769	34	2014	2014
6	Guiltymind	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	456398	879	75	1994	2012
13	Close my eyes	8	160	87657	657	29	1996	2000
14	Guiltymind	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	79676	567	12	1996	2020
18	No Flaws	9	679	345476	343	13	2010	2012
19	Drop It	10	450	456358	919	1	2013	2015

The Results pane also shows the output of the second query:

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	98567	405	23	2011	2012
3	Caught up	4	1070	89456	708	45	2012	2013
4	Torn Up	2	1000	44567	560	67	2013	2014
5	Close my eyes	8	160	967487	769	34	2014	2014
6	Guiltymind	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	456398	879	75	1994	2012
13	Close my eyes	8	160	87657	657	29	1996	2000
14	Guiltymind	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	79676	567	12	1996	2020
18	No Flaws	9	679	345476	343	13	2010	2012
19	Drop It	10	450	456358	919	1	2013	2015

The status bar at the bottom indicates 'Query executed successfully.' and 'LAPTOP-01M17FIT (15.0 RTM) LAPTOP-01M17FIT(user (70)) Library 00:00:00 19 rows'.



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 at the bottom shows the output of the query:

author_name
1 chandra
2 maili
3 no
4 so_hail

The status bar at the bottom indicates 'Query executed successfully.' and 'LAPTOP-01M7FIT (15.0 RTM) LAPTOP-01M7FIT\user (70) Library 00:00:00 4 rows'.

~

iii) UNION and INTERSECT :-

The screenshot shows the Microsoft SQL Server Management Studio interface. The main query window contains the following SQL code:

```
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))
```

The Results pane at the bottom shows the output of the first query (UNION):

author_name
1 greeshma
2 no
3 maili
4 chandra
5 lohith
6 Bhanupriya
7 Lohith
8 Koushik
9 Suman

The status bar at the bottom indicates 'Query executed successfully.' and 'LAPTOP-01M7FIT (15.0 RTM) LAPTOP-01M7FIT\user (70) Library 00:00:00 9 rows'.

~

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

I) JOIN :-

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL query:

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

The Results pane displays the following data:

	BOOK_NAME	author_id
1	Natural Disaster	1
2	Guilty mind	5
3	Leave me alone	6
4	My Band	7
5	Fantasy Girl	8
6	No Flaws	15
7	Guilty mind	5
8	Leave me alone	6
9	My Band	7
10	Fantasy Girl	8
11	No Flaws	9
12	Drop It	10
13	ALONE	9
14	SOLO	9

The status bar at the bottom indicates the query was executed successfully, showing 14 rows.

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL query:

```
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 displays the following data:

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

The status bar at the bottom indicates the query was executed successfully, showing 10 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 BOOK_NAME, COUNT(BOOK_NAME) from books_
JOIN author_ on author_.author_id = books_.author_id GROUP by BOOK_NAME

```

165 %

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	Guiltly 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	Tun Up	1

Query executed successfully.

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

Ready Ln 82 Col 1 Ch 1 INS

~

ii) RIGHT 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 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	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-01M17FIT (15.0 RTM) LAPTOP-01M17FIT\user (57) Library 00:00:00 12 rows

Ready Ln 99 Col 1 Ch 1 INS

~

SQLQuery1.sql - LAPTOP-01M17FIT.Library (LAPTOP-01M17FITuser (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...01M17FITuser (57))

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

Results

	author_name	BOOK_NAME
1	greeshma	Natural Disaster
2	greeshma	In the Dark
3	koushik	Tom 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	Mailikarjun	NULL
25	Pavan	NULL
26	Suman	No Flaws
27	Pritya	NULL
28	Vishwa	NULL
29	Geetha	NULL

Query executed successfully.

LAPTOP-01M17FIT (15.0 RTM) LAPTOP-01M17FITuser (57) Library 00:00:00 29 rows

Ready Ln 103 Col 1 Ch 1 INS

SQLQuery1.sql - LAPTOP-01M17FIT.Library (LAPTOP-01M17FITuser (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...01M17FITuser (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

	author_name	(No column name)
1	Bhanupriya	4
2	chandra	2
3	Geetha	0
4	greeshma	2
5	Koushik	3
6	lohith	5
7	mali	2
8	Mailikarjun	0
9	Pavan	0
10	Pritya	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-01M17FITuser (57) Library 00:00:00 15 rows

Ready Ln 116 Col 88 Ch 88 INS

### iii) LEFT JOIN :-

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
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 displays the output of the query:

	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	Tom Up	1

The status bar at the bottom indicates the query was executed successfully.

~

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

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

The Results pane displays the output of the query:

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

The status bar at the bottom indicates the query was executed successfully.

~

SQLQuery1.sql - LAPTOP-01MI7FIT.Library (LAPTOP-01MI7FIT\user (S7)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Library

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

```
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	Bhanupriya	4
2	chandra	2
3	greeshma	2
4	Koushik	3
5	Lohith	5
6	maili	2
7	no	2
8	Suman	1

Query executed successfully.

LAPTOP-01MI7FIT (15.0 RTM) LAPTOP-01MI7FIT\user (S7) Library 00:00:00 8 rows

Ready Ln 147 Col 15 Ch 15 INS