

Assignment-1(SQL)

Name:Madugundu Suresh (M.Suresh -G8 DS)

1. Create Schema Library

2. Create tables

Books: BookID - Pk,BookName,AuthorName,Genre,pages

Customers : CustomerId - PK, CustomerName, address

BooksBorrowed:SlNo,BookID - FK,CustomerID - FK,DaysBookRetained

Cost: MaxPages, Cost/day

The screenshot shows a SQL Server Enterprise Manager window with a script editor on the left and an 'Action Output' pane at the bottom. The script contains the following SQL commands:

```
1 • create schema Library; /*Q1*/
2 • use Library;
3 /*Q2*/
4 • create table Books(
5   BookID int primary key,
6   BookName varchar(100),
7   AuthorName varchar(100),
8   Genre varchar(100),
9   pages int);
10 • create table Customers(
11   CustomerId int primary key,
12   CustomerName varchar(100),
13   address varchar(100));
14 • create table BooksBorrowed(
15   SlNo int,
16   BookID int,
17   CustomerID int,
18   DaysBookRetained int,
19   foreign key(BookID) references Books(BookID),
20   foreign key(CustomerID) references Customers(CustomerID)
21 );
```

The 'Action Output' pane displays the results of the executed commands:

#	Time	Action	Message	Duration / Fetch
✓ 1	23:06:07	create schema Library	1 row(s) affected	0.00074 sec
✓ 2	23:06:13	use Library	0 row(s) affected	0.00035 sec
✓ 3	23:06:20	create table Books(BookID int primary key, BookName varchar...	0 row(s) affected	0.599 sec

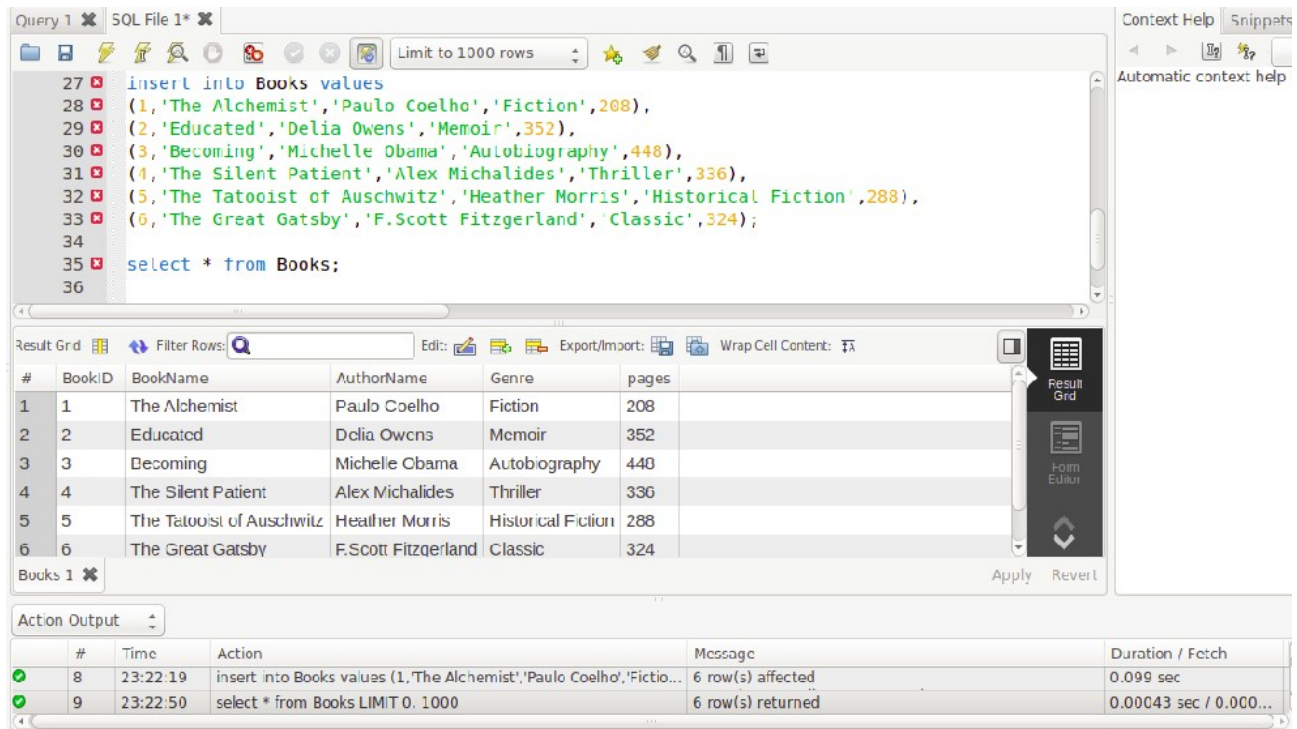
The screenshot shows the continuation of the SQL script execution. The script editor shows the following commands:

```
21 );
22 • create table Cost(
23   MaxPages int,
24   costperday int
25 );
26
```

The 'Action Output' pane displays the results, including an error for the 'BooksBorrowed' table:

#	Time	Action	Message	Duration / Fetch
✓ 3	23:06:20	create table Books(BookID int primary key, BookName varchar...	0 row(s) affected	0.599 sec
✓ 4	23:06:28	create table Customers(CustomerId int primary key, Custom...	0 row(s) affected	0.401 sec
✓ 5	23:06:42	create table BooksBorrowed(SlNo int, BookID int, CustomerID i...	0 row(s) affected	0.489 sec
✗ 6	23:06:49	create table BooksBorrowed(SlNo int, BookID int, CustomerID i...	Error Code: 1050. Table 'BooksBorrowed' already exists	0.0012 sec
✓ 7	23:06:58	create table Cost(MaxPages int, costperday int)	0 row(s) affected	0.410 sec

3. Google Search for books along with authors, genre and pages. Insert atleast 5 authors and 5 different genres into Books Table. Make sure you have atleast 30 records.



The screenshot shows a SQL query window with the following code:

```

27 insert into Books values
28 (1,'The Alchemist','Paulo Coelho','Fiction',208),
29 (2,'Educated','Delia Owens','Memoir',352),
30 (3,'Becoming','Michelle Obama','Autobiography',448),
31 (4,'The Silent Patient','Alex Michalides','Thriller',336),
32 (5,'The Tattooist of Auschwitz','Heather Morris','Historical Fiction',288),
33 (6,'The Great Gatsby','F.Scott Fitzgerald','Classic',324);
34
35 select * from Books;
36

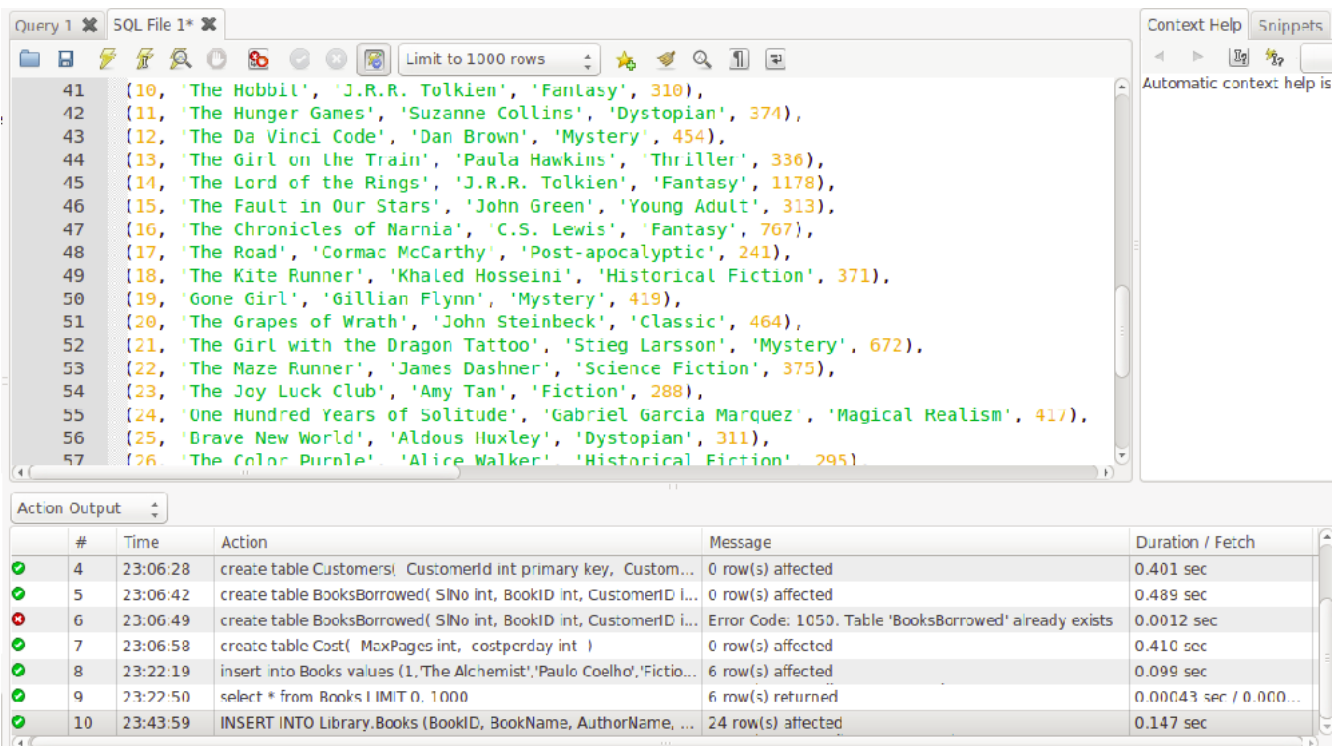
```

Below the query window, the 'Result Grid' shows the following data:

#	BookID	BookName	AuthorName	Genre	pages
1	1	The Alchemist	Paulo Coelho	Fiction	208
2	2	Educated	Delia Owens	Memoir	352
3	3	Becoming	Michelle Obama	Autobiography	440
4	4	The Silent Patient	Alex Michalides	Thriller	336
5	5	The Tattooist of Auschwitz	Heather Morris	Historical Fiction	288
6	6	The Great Gatsby	F.Scott Fitzgerald	Classic	324

The 'Action Output' window shows the following log:

#	Time	Action	Message	Duration / Fetch
8	23:22:19	insert into Books values (1,'The Alchemist','Paulo Coelho','Fiction',208), (2,'Educated','Delia Owens','Memoir',352), (3,'Becoming','Michelle Obama','Autobiography',448), (4,'The Silent Patient','Alex Michalides','Thriller',336), (5,'The Tattooist of Auschwitz','Heather Morris','Historical Fiction',288), (6,'The Great Gatsby','F.Scott Fitzgerald','Classic',324);	6 row(s) affected	0.099 sec
9	23:22:50	select * from Books LIMIT 0. 1000	6 row(s) returned	0.00043 sec / 0.000...



The screenshot shows a SQL query window with the following code:

```

41 (10,'The Hobbit','J.R.R. Tolkien','Fantasy',310),
42 (11,'The Hunger Games','Suzanne Collins','Dystopian',374),
43 (12,'The Da Vinci Code','Dan Brown','Mystery',454),
44 (13,'The Girl on the Train','Paula Hawkins','Thriller',336),
45 (14,'The Lord of the Rings','J.R.R. Tolkien','Fantasy',1178),
46 (15,'The Fault in Our Stars','John Green','Young Adult',313),
47 (16,'The Chronicles of Narnia','C.S. Lewis','Fantasy',767),
48 (17,'The Road','Cormac McCarthy','Post-apocalyptic',241),
49 (18,'The Kite Runner','Khaled Hosseini','Historical Fiction',371),
50 (19,'Gone Girl','Gillian Flynn','Mystery',419),
51 (20,'The Grapes of Wrath','John Steinbeck','Classic',464),
52 (21,'The Girl with the Dragon Tattoo','Stieg Larsson','Mystery',672),
53 (22,'The Maze Runner','James Dashner','Science Fiction',375),
54 (23,'The Joy Luck Club','Amy Tan','Fiction',288),
55 (24,'One Hundred Years of Solitude','Gabriel Garcia Marquez','Magical Realism',417),
56 (25,'Brave New World','Aldous Huxley','Dystopian',311),
57 (26,'The Color Purple','Alice Walker','Historical Fiction',295);

```

Below the query window, the 'Action Output' window shows the following log:

#	Time	Action	Message	Duration / Fetch
4	23:06:28	create table Customers(Customerid int primary key, CustomerName varchar(50), Address varchar(100), City varchar(50), State varchar(50), Zip varchar(10));	0 row(s) affected	0.401 sec
5	23:06:42	create table BooksBorrowed(SIdo int, BookID int, CustomerID int, BorrowDate datetime, ReturnDate datetime);	0 row(s) affected	0.489 sec
6	23:06:49	create table BooksBorrowed(SIdo int, BookID int, CustomerID int, BorrowDate datetime, ReturnDate datetime);	Error Code: 1050. Table 'BooksBorrowed' already exists	0.0012 sec
7	23:06:58	create table Cost(MaxPages int, costperday int);	0 row(s) affected	0.410 sec
8	23:22:19	insert into Books values (1,'The Alchemist','Paulo Coelho','Fiction',208), (2,'Educated','Delia Owens','Memoir',352), (3,'Becoming','Michelle Obama','Autobiography',448), (4,'The Silent Patient','Alex Michalides','Thriller',336), (5,'The Tattooist of Auschwitz','Heather Morris','Historical Fiction',288), (6,'The Great Gatsby','F.Scott Fitzgerald','Classic',324);	6 row(s) affected	0.099 sec
9	23:22:50	select * from Books LIMIT 0. 1000	6 row(s) returned	0.00043 sec / 0.000...
10	23:43:59	INSERT INTO Library.Books (BookID, BookName, AuthorName, Genre, pages) VALUES (10,'The Hobbit','J.R.R. Tolkien','Fantasy',310), (11,'The Hunger Games','Suzanne Collins','Dystopian',374), (12,'The Da Vinci Code','Dan Brown','Mystery',454), (13,'The Girl on the Train','Paula Hawkins','Thriller',336), (14,'The Lord of the Rings','J.R.R. Tolkien','Fantasy',1178), (15,'The Fault in Our Stars','John Green','Young Adult',313), (16,'The Chronicles of Narnia','C.S. Lewis','Fantasy',767), (17,'The Road','Cormac McCarthy','Post-apocalyptic',241), (18,'The Kite Runner','Khaled Hosseini','Historical Fiction',371), (19,'Gone Girl','Gillian Flynn','Mystery',419), (20,'The Grapes of Wrath','John Steinbeck','Classic',464), (21,'The Girl with the Dragon Tattoo','Stieg Larsson','Mystery',672), (22,'The Maze Runner','James Dashner','Science Fiction',375), (23,'The Joy Luck Club','Amy Tan','Fiction',288), (24,'One Hundred Years of Solitude','Gabriel Garcia Marquez','Magical Realism',417), (25,'Brave New World','Aldous Huxley','Dystopian',311), (26,'The Color Purple','Alice Walker','Historical Fiction',295);	24 row(s) affected	0.147 sec

Query 1 SQL File 1*

Result Grid

#	BookID	BookName	AuthorName	Genre	pages
21	21	The Girl with the Drago...	Stieg Larsson	Mystery	672
22	22	The Maze Runner	James Dashner	Science Fiction	375
23	23	The Joy Luck Club	Amy Tan	Fiction	288
24	24	One Hundred Years of ...	Gabriel Garcia ...	Magical Realism	417
25	25	Brave New World	Aldous Huxley	Dystopian	311
26	26	The Color Purple	Alice Walker	Historical Fiction	295
27	27	The Handmaid's Tale	Margaret Atwood	Dystopian	311
28	28	The Godfather	Mario Puzo	Crime	448
29	29	Wuthering Heights	Emily Bronte	Gothic	464
30	30	The Picture of Dorian ...	Oscar Wilde	Gothic	254
*	NULL	NULL	NULL	NULL	NULL

Books 2

Action Output

#	Time	Action	Message	Duration / Fetch
5	23:06:42	create table BooksBorrowed(SNo int, BookID int, CustomerID int ...	0 row(s) affected	0.489 sec
6	23:06:49	create table BooksBorrowed(SNo int, BookID int, CustomerID int ...	Error Code: 1050. Table 'BooksBorrowed' already exists	0.0012 sec
7	23:06:58	create table Cost(MaxPages int, costperday int)	0 row(s) affected	0.410 sec
8	23:22:19	insert into Books values (1,'The Alchemist','Paulo Coelho','Fictio...	6 row(s) affected	0.099 sec
9	23:22:50	select * from Books LIMIT 0, 1000	6 row(s) returned	0.00043 sec / 0.000...
10	23:43:59	INSERT INTO Library.Books (BookID, BookName, AuthorName, ...	24 row(s) affected	0.147 sec
11	23:44:33	select * from Books LIMIT 0, 1000	30 row(s) returned	0.00079 sec / 0.000...

4. Insert 15 customers into customer table by providing appropriate details

Query 1 SQL File 1*

Limit to 1000 rows

```

63 select * from Books;
64
65
66 insert into Customers values
67 (1,'Suresh','Kurnool'),
68 (2,'Kethan','Chittor'),
69 (3,'Dhanush','Chittor'),
70 (4,'Narayanaswamy','Kadapa'),
71 (5,'K.Ramudu','Tirupathi'),
72 (6,'Madhu Kiran',''),
73 (7,'Lahari','Chennai'),
74 (8,'Shiva','Hyderabad'),
75 (9,'Lavanya','Ongole'),
76 (10,'Lakshmi','Nuzvid'),
77 (11,'Dharmateja','Madanapalli'),
78 (12,'Dhanasekaran','Chennai'),
79 (13,'Sowrya sree','Kalikiri'),
80 (14,'BHargav','Kadapa'),
81 (15,'Lakshmipriya','Nellore');
82
83
84

```

Action Output

#	Time	Action	Message	Duration / Fetch
12	23:54:47	insert into Customers values (1,'Suresh','Kurnool'), (2,'Kethan',...	15 row(s) affected Records: 15 Duplicates: 0 Warnings: 0	0.081 sec

Query 1 x SQL File 1* x

Limit to 1000 rows

Result Grid

#	CustomerId	CustomerName	address
1	1	Suresh	Kumool
2	2	Kethari	Chitor
3	3	Dhanush	Chitor
4	4	Narayanaswamy	Kadapa
5	5	K.Ramudu	Trupathi
6	6	Madhu Kiran	Mumbai
7	7	Lahari	Chennai
8	8	Shiva	Hyderabad
9	9	Lavanya	Ongole
10	10	Lakshmi	Nuzvid
11	11	Dharmateja	Madena...
12	12	Dhanasekaran	Chennai
13	13	Sowrya sree	Kalikiri
14	14	BHargav	Kadapa
15	15	Lakshmipriya	Nellore

Customers 4 x

Apply Revert

Action Output

#	Time	Action	Message	Duration / Fetch
14	23:56:47	update Customers set address='Mumbai' where CustomerID=6	1 row(s) affected	0.186 sec
15	23:56:54	select * from Customers LIMIT 0, 1000	15 row(s) returned	0.00076 sec / 0.000...

5. Using BookId in Books Table and CustomerID in Customer Table insert data into BooksBorrowed table to have atleast 50 records.

Query 1 x assignment* x

Limit to 1000 rows

```

79 (13,'Sowrya sree','Kalikiri'),
80 (14,'BHargav','Kadapa'),
81 (15,'Lakshmipriya','Nellore');
82
83 • update Customers set address='Mumbai' where CustomerID=6;
84 • select * from Customers;
85
86 ✖ insert into BooksBorrowed values
87 (1,1, 1, 7),
88 (2,2, 3, 5),
89 (3,3, 5, 10),
90 (4,4, 7, 14),
91 (5,5, 9, 7),
92 (6,6, 11, 21),
93 (7,7, 13, 14),
94 (8,8, 15, 7),
95 (9,9, 2, 10),
96 (10,10, 4, 14),
97 (11,11, 6, 21),
98 (12,12, 8, 7),
99 (13,13,10,10),
100 (14,14,12,14),
101 (15,15,14,7),

```

Action Output

#	Time	Action	Message	Duration / Fetch
3	20:37:12	insert into BooksBorrowed values (1,1, 1, 7), (2,2, 3, 5), (3,3, ...	50 row(s) affected	0.199 sec

Query 1 assignment*

Result Grid Filter Rows: Export: Wrap Cell Content: Read Only

#	SINo	BookID	CustomerID	DaysBookRetained
1	1	1	1	7
2	2	2	3	5
3	3	3	5	10
4	4	4	7	14
5	5	5	9	7
6	6	6	11	21
7	7	7	13	14
8	8	8	15	7
9	9	9	2	10
10	10	10	4	14
11	11	11	6	21
12	12	12	8	7
13	13	13	10	10
14	14	14	12	14
15	15	15	14	7
16	16	16	3	7

BooksBorrowed 1

Action Output

#	Time	Action	Message	Duration / Fetch
4	20:37:43	select * from BooksBorrowed LIMIT 0, 1000	50 row(s) returned	0.00044 sec / 0.000...

Result Grid Filter Rows: Export: Wrap Cell Content: Read Only

#	SINo	BookID	CustomerID	DaysBookRetained
35	5	9	7	
36	6	11	21	
37	7	13	14	
38	8	15	7	
39	9	2	10	
40	10	4	14	
41	11	11	11	
42	19	12	12	
43	10	10	10	
44	9	9	9	
45	26	14	1	
46	6	12	12	
47	8	8	9	
48	7	12	19	
49	8	9	8	
50	10	13	12	

BooksBorrowed 1

Action Output

#	Time	Action	Message	Duration / Fetch
4	20:37:43	select * from BooksBorrowed LIMIT 0, 1000	50 row(s) returned	0.00044 sec / 0.000...

6. Retrieve the total number of books in each genre.

Query 1 assignment* Limit to 1000 rows

```

137
138 • select * from BooksBorrowed;
139
140 • select Genre, count(*) from Books group by Genre;
141
142

```

Result Grid Filter Rows: Export: Wrap Cell Content: Read Only

#	Genre	count(*)
1	Autobiography	1
2	Classic	3
3	Coming-of-age	1
4	Critique	1
5	Dystopian	4
6	Fantasy	3
7	Fiction	2
8	Gothic	2
9	Historical Fiction	3

Result 2

Action Output

#	Time	Action	Message	Duration / Fetch
5	20:43:21	select Genre, count(*) from Books group by Genre LIMIT 0, 1000	16 row(s) returned	0.058 sec / 0.00001...

Query 1 assignment*

Limit to 1000 rows

```

137
138 • select * from BooksBorrowed;
139
140 • select Genre, count(*) from Books group by Genre;
141
142

```

Context Help Snippets
Automatic context help

Result Grid

#	Genre	count(*)
8	Gothic	2
9	Historical Fiction	3
10	Magical Realism	1
11	Memoir	1
12	Mystery	3
13	Post-apocalyptic	1
14	Science Fiction	1
15	Thriller	2
16	Young Adult	1

Result 2 Read Only

Action Output

#	Time	Action	Message	Duration / Fetch
5	20:43:21	select Genre, count(*) from Books group by Genre LIMIT 0, 1000	16 row(s) returned	0.058 sec / 0.00001...

7. Retrieve total number of books borrowed.

Query 1 assignment*

Limit to 1000 rows

```

143
144
145 • select count(*) as Total_NO_OF_Books_Borrowed from BooksBorrowed;
146
147

```

Context Help Snippets
Automatic context help

Result Grid

#	Total_NO_OF_Books_Borrowed
1	50

Result 3 Read Only

Action Output

#	Time	Action	Message	Duration / Fetch
6	20:49:15	select count(*) as Total_NO_OF_Books_Borrowed from BooksBo...	1 row(s) returned	0.00045 sec / 0.000...

8. retrieve names of books borrowed without repetition

The screenshot shows a database query tool interface. The query editor at the top contains the following SQL query:

```
select distinct BookName from Books as b inner join  
BooksBorrowed as B where b.BookID=B.BookID;
```

The results are displayed in a table with the following data:

#	BookName
1	The Alchemist
2	Educated
3	Becoming
4	The Silent Patient
5	The Tattooist of Auschwitz
6	The Great Gatsby
7	To Kill a Mockingbird
8	1984
9	The Catcher in the Rye
10	The Hobbit

The Action Output pane at the bottom shows the following message:

#	Time	Action	Message	Duration / Fetch
7	20:52:13	select distinct BookName from Books as b inner join BooksBorr...	25 row(s) returned	0.00094 sec / 0.000...

The screenshot shows a database query tool interface. The query editor at the top contains the following SQL query:

```
select distinct BookName from Books as b inner join  
BooksBorrowed as B where b.BookID=B.BookID;
```

The results are displayed in a table with the following data:

#	BookName
16	The Chronicles of Narnia
17	The Road
18	The Kite Runner
19	Gone Girl
20	The Grapes of Wrath
21	One Hundred Years of ...
22	Brave New World
23	The Color Purple
24	The Handmaid's Tale
25	Wuthering Heights

The Action Output pane at the bottom shows the following message:

#	Time	Action	Message	Duration / Fetch
7	20:52:13	select distinct BookName from Books as b inner join BooksBorr...	25 row(s) returned	0.00094 sec / 0.000...

9. Retrieve the customerName, BookName for books borrowed.

Query 1 assignment*

```
149
150 select CustomerName,BookName from Customers C inner join BooksBorrowed BB on C.CustomerId=
151 inner join Books B on BB.BookID=B.BookID;
152
153
154
```

Result Grid

#	CustomerName	BookName
1	Suresh	The Alchemist
2	Suresh	The Alchemist
3	Kethan	The Catcher in the Rye
4	Kethan	The Catcher in the Rye
5	Dhanush	Educated
6	Dhanush	The Chronicles of Narnia
7	Dhanush	Educated
8	Narayanaswamy	The Hobbit
9	Narayanaswamy	The Hobbit
10	K.Ramudu	Becoming

Action Output

#	Time	Action	Message	Duration / Fetch
8	20:59:20	select CustomerName,BookName from Customers C inner join ...	50 row(s) returned	0.00093 sec / 0.000...

Query 1 assignment*

```
149
150 select CustomerName,BookName from Customers C inner join BooksBorrowed BB on C.CustomerId=
151 inner join Books B on BB.BookID=B.BookID;
152
153
154
```

Result Grid

#	CustomerName	BookName
11	K.Ramudu	The Road
12	K.Ramudu	Becoming
13	Madhu Kiran	The Hunger Games
14	Lahari	The Silent Patient
15	Lahari	The Kite Runner
16	Lahari	The Silent Patient
17	Shiva	The Da Vinci Code
18	Shiva	1984
19	Lavanya	The Tattooist of Auschwitz
20	Lavanya	Gone Girl

Action Output

#	Time	Action	Message	Duration / Fetch
8	20:59:20	select CustomerName,BookName from Customers C inner join ...	50 row(s) returned	0.00093 sec / 0.000...

Query 1 assignment*

Limit to 1000 rows

```

149
150 select CustomerName,BookName from Customers C inner join BooksBorrowed BB on C.CustomerId=
151 inner join Books B on BB.BookID=B.BookID;
152
153
154

```

Result Grid

#	CustomerName	BookName
22	Lavanya	The Talooist of Auschwitz
23	Lavanya	The Catcher in the Rye
24	Lavanya	1984
25	Lakshmi	The Girl on the Train
25	Lakshmi	The Hobbit
27	Dharmateja	The Great Gatsby
28	Dharmateja	The Grapes of Wrath
29	Dharmateja	One Hundred Years of...
30	Dharmateja	The Great Gatsby
31	Dharmateja	The Hunger Games

Result 5

Action Output

#	Time	Action	Message	Duration / Fetch
8	20:59:20	select CustomerName,BookName from Customers C inner join ...	50 row(s) returned	0.00093 sec / 0.000...

Query 1 assignment*

Limit to 1000 rows

```

149
150 select CustomerName,BookName from Customers C inner join BooksBorrowed BB on C.CustomerId=
151 inner join Books B on BB.BookID=B.BookID;
152
153
154

```

Result Grid

#	CustomerName	BookName
31	Dharmateja	The Hunger Games
32	Dhanasekaran	The Lord of the Rings
33	Dhanasekaran	The Handmaid's Tale
34	Dhanasekaran	The Da Vinci Code
35	Dhanasekaran	The Great Gatsby
36	Dhanasekaran	To Kill a Mockingbird
37	Dhanasekaran	Gone Girl
38	Dhanasekaran	The Great Gatsby
39	Dhanasekaran	To Kill a Mockingbird
40	Sowrya srcc	To Kill a Mockingbird

Result 5

Action Output

#	Time	Action	Message	Duration / Fetch
8	20:59:20	select CustomerName,BookName from Customers C inner join ...	50 row(s) returned	0.00093 sec / 0.000...

Query 1 assignment*

```

149
150 select CustomerName,BookName from Customers C inner join BooksBorrowed BB on C.CustomerId=
151 inner join Books B on BB.BookID=B.BookID;
152
153
154

```

Result Grid

#	CustomerName	BookName
41	Sowrya sree	Wuthering Heights
42	Sowrya sree	Brave New World
43	Sowrya sree	The Hobbit
44	Sowrya sree	To Kill a Mockingbird
45	Sowrya sree	The Hobbit
46	DI Hargav	The Fault in Our Stars
47	BHargav	The Color Purple
48	BHargav	The Color Purple
49	Lakshmipriya	1984
50	Lakshmipriya	1984

Result: 5

Action Output

#	Time	Action	Message	Duration / Fetch
8	20:59:20	select CustomerName,BookName from Customers C inner join ...	50 row(s) returned	0.00093 sec / 0.000...

10. Insert below data into Cost table 100pages -10 rs, 200pages - 15rs, 300 pages - 30rs, 1000 pages -50Rs

Query 1 assignment*

```

151
152 insert into Cost values
153 (100,10),
154 (200,15),
155 (300,30),
156 (1000,50);
157
158 select * from Cost;
159
160
161

```

Result Grid

#	MaxPages	costperday
1	100	10
2	200	15
3	300	30
4	1000	50

Cost 6

Action Output

#	Time	Action	Message	Duration / Fetch
10	21:04:23	insert into Cost values (100,10), (200,15), (300,30), (1000,50)	4 row(s) affected	0.087 sec
11	21:04:38	select * from Cost LIMIT 0, 1000	4 row(s) returned	0.00068 sec / 0.000...

11. total Earnings by Library.

The screenshot shows a database query editor with a SQL query to calculate total earnings by library. The query is as follows:

```
SELECT SUM(CostPerDay * DaysBookRetained) AS Total_Earnings
FROM BooksBorrowed BB INNER JOIN Books B ON BB.BookID = B.BookID
INNER JOIN Cost ON pages <= MaxPages;
```

The result grid shows a single row with the total earnings of 31000.

#	Total_Earnings
1	31000

The action output shows the execution of the query, with a message indicating that 1 row(s) returned.

#	Time	Action	Message	Duration / Fetch
19	21:26:03	select B.pages,RR.BookID,RR.DaysBookRetained from Books B i...	50 row(s) returned	0.00070 sec / 0.000...
20	21:33:30	SELECT SUM(CostPerDay * DaysBookRetained) AS Total_Earn...	1 row(s) returned	0.00080 sec / 0.000...

12. Amount paid by each customer. Rank the customers based in amount paid.

The screenshot shows a database query editor with a SQL query to calculate total earnings by customer and rank them. The query is as follows:

```
SELECT CustomerName,SUM(CostPerDay * DaysBookRetained) AS Total_Earnings
FROM Customers C INNER JOIN BooksBorrowed BB ON C.CustomerId=BB.CustomerID
INNER JOIN Books B ON BB.BookID = B.BookID
INNER JOIN Cost ON pages <= MaxPages GROUP BY C.CustomerID order by Total_Earnings DESC;
```

The result grid shows the total earnings for each customer, ranked by amount paid.

#	CustomerName	Total_Earnings
1	Dhanasekaran	6390
2	Sowrya sree	4440
3	Lavanya	3690
4	Dharmateja	3550
5	Iahari	2100
6	K.Ramudu	1800
7	Kethan	1600
8	Narayanaswamy	1400
9	Suresh	1120

The action output shows the execution of the query, with a message indicating that 15 row(s) returned.

#	Time	Action	Message	Duration / Fetch
27	22:20:21	SELECT CustomerName,SUM(CostPerDay * DaysBookRetained)...	15 row(s) returned	0.0012 sec / 0.0000...

Limit to 1000 rows

```

174
175
176 • SELECT CustomerName,SUM(CostPerDay * DaysBookRetained) AS Total_Earnings
177 FROM Customers C INNER JOIN BooksBorrowed BB ON C.CustomerId=BB.CustomerID
178 INNER JOIN Books B ON BB.BookID = B.BookID
179 INNER JOIN Cost ON pages <= MaxPages GROUP BY C.CustomerID order by Total_Earnings DESC;
180

```

Result Grid

#	CustomerName	Total_Earnings
7	Kethan	1600
8	Narayanaswamy	1400
9	Suresh	1120
10	Madhu Kiran	1050
11	Lakshmi	1000
12	Dhanush	850
13	Shiva	800
14	LakshmiPriya	700
15	BHargav	510

Result: 14

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 27	22:20:21	SELECT CustomerName,SUM(CostPerDay * DaysBookRetained)...	15 row(s) returned	0.0012 sec / 0.0000...

12. Amount paid by each customer. Rank the customers based in amount paid.

MySql

```

1
2 SELECT CustomerName, SUM(CostPerDay * DaysBookRetained)
3 AS Total_Earnings,RANK() OVER
4 (ORDER BY SUM(CostPerDay * DaysBookRetained) DESC) AS Rnk
5 FROM Customers C
6 INNER JOIN BooksBorrowed BB ON
7 C.CustomerId = BB.CustomerID
8 INNER JOIN Books B ON BB.BookID = B.BookID
9 INNER JOIN Cost ON B.Pages <= Cost.MaxPages
10 GROUP BY C.CustomerID
11 ORDER BY Total_Earnings DESC;
12

```

Run Code

Output

CustomerName	Total_Earnings	Rnk
Dhanasekaran	6390	1
Sowrya sree	4440	2
Lavanya	3690	3
Dharmateja	3550	4
Lahari	2100	5
K.Ramudu	1800	6
Kethan	1600	7
Narayanaswamy	1400	8
Suresh	1120	9
Madhu Kiran	1050	10
Lakshmi	1000	11
Dhanush	850	12
Shiva	800	13
Lakshmipriya	700	14
BHargav	510	15