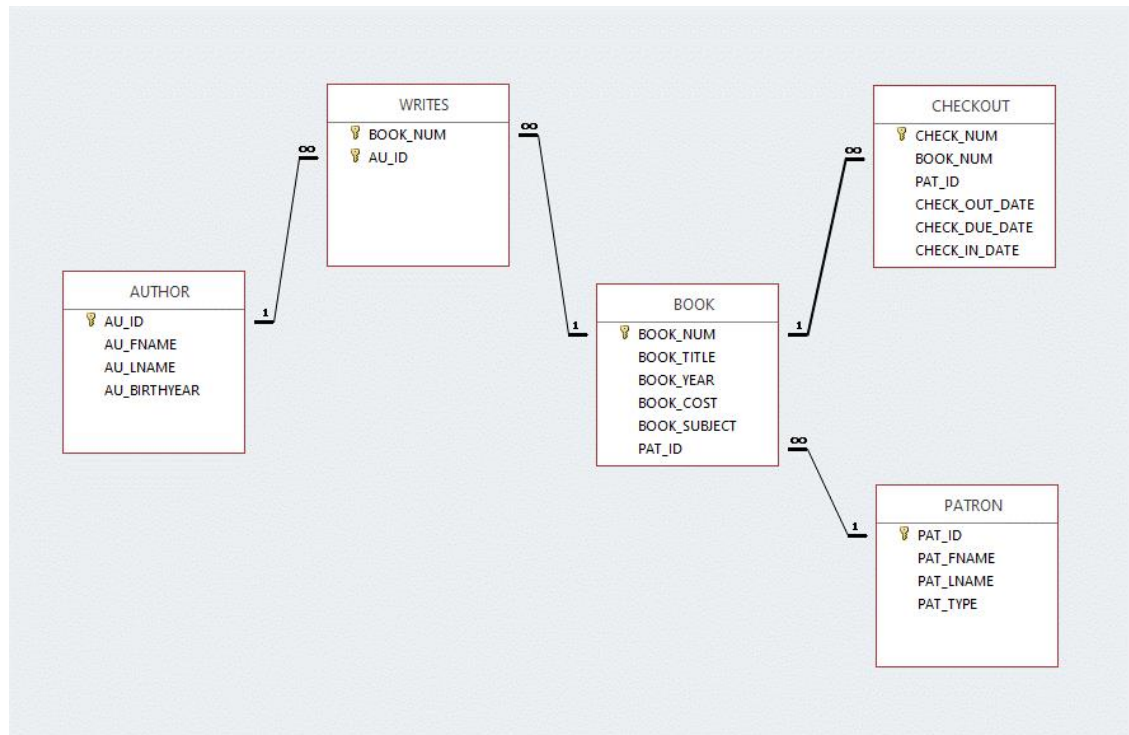


## Team: GOD (Granter of Data)

Team Member: Sohal Patel, Gabriel Harper, Jose Andre, Carrie Doucette, Janelle Hall

I.



## II.

56.

SELECT

BOOK\_TITLE,  
BOOK\_COST,  
BOOK\_YEAR

FROM

BOOK

ORDER BY

BOOK\_TITLE;

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT
2     BOOK_TITLE,
3     BOOK_COST,
4     BOOK_YEAR
5 FROM
6     BOOK
7 ORDER BY
8     BOOK_TITLE;
9
```

91 %

Results Messages

	BOOK_TITLE	BOOK_COST	BOOK_YEAR
1	Beginner's Guide to JAVA	59.95	2014
2	Beyond the Database Veil	69.95	2016
3	C# in Middleware Deployment	59.95	2015
4	Capture the Cloud	69.95	2016
5	Cloud-based Mobile Applications	69.95	2015
6	Coding Style for Maintenance	49.95	2017
7	Conceptual Programming	59.95	2015
8	Database in the Cloud	79.95	2014
9	DATABASES in Theory	129.95	2015
10	iOS Programming	79.95	2015
11	J++ in Mobile Apps	49.95	2015
12	JAVA First Steps	49.95	2015
13	Mastering the database environment	89.95	2015
14	Reengineering the Middle Tier	89.95	2016
15	Shining Through the Cloud: Sun Programming	109.95	2016
16	Starlight Applications	69.95	2016
17	The Golden Road to Platform independence	119.95	2016
18	Thoughts on Revitalizing Ruby	59.95	2016
19	Virtual Programming for Virtual Environments	79.95	2016
20	What You Always Wanted to Know About Dat...	49.95	2016

57.

SELECT

PATRON.PAT\_FNAME,  
PATRON.PAT\_LNAME

FROM

PATRON

ORDER BY

UPPER(PATRON.PAT\_LNAME),  
UPPER(PATRON.PAT\_FNAME);

SQLQuery1.sql - US...USER-PC\User (56)) \* X

```
1 SELECT
2   PATRON.PAT_FNAME,
3   PATRON.PAT_LNAME
4 FROM
5   PATRON
6 ORDER BY
7   UPPER(PATRON.PAT_LNAME),
8   UPPER(PATRON.PAT_FNAME);
```

91 %

Results Messages

	PAT_FNAME	PAT_LNAME
1	Vera	Alvarado
2	Holly	Anthony
3	Cedric	Baldwin
4	Cory	Barry
5	Nadine	Blair
6	Erika	Bowen
7	Gerald	Burke
8	Ollie	Cantrell
9	robert	carter
10	Keith	Cooley
11	Alicia	Dickson
12	Maureen	Downs
13	Thomas	Duran
14	Jewel	England
15	Wilfred	Fuller
16	Jerald	Gaines
17	Claire	Gomez
18	Homer	Goodman
19	Desiree	Harrington
20	Jose	Hays
21	Allen	Home
22	Helena	Hughes
23	Elton	Inwin
24	Jan	Joyce
25	Zach	Kelly
26	Roberto	Kennedy
27	Marina	King
28	Mario	King

Query executed successfully.

USER-PC (14.0 RTM) | USER-PC\User (56) | Ch07\_Fact | 00:00:00 | 50 rows

58.

SELECT

```
CHECKOUT.CHECK_NUM,  
CHECKOUT.CHECK_OUT_DATE,  
CHECKOUT.CHECK_DUE_DATE
```

FROM

```
CHECKOUT
```

ORDER BY

```
CHECKOUT.CHECK_NUM;
```

SQLQuery1.sql - US...USER-PC\User (56)) \* X

```
1 SELECT  
2   CHECKOUT.CHECK_NUM,  
3   CHECKOUT.CHECK_OUT_DATE,  
4   CHECKOUT.CHECK_DUE_DATE  
5 FROM  
6   CHECKOUT  
7 ORDER BY  
8   CHECKOUT.CHECK_NUM;  
9
```

91 %

Results Messages

	CHECK_NUM	CHECK_OUT_DATE	CHECK_DUE_DATE
1	91001	2017-03-31 00:00:00.000	2017-04-14 00:00:00.000
2	91002	2017-03-31 00:00:00.000	2017-04-07 00:00:00.000
3	91003	2017-03-31 00:00:00.000	2017-04-14 00:00:00.000
4	91004	2017-03-31 00:00:00.000	2017-04-14 00:00:00.000
5	91005	2017-03-31 00:00:00.000	2017-04-07 00:00:00.000
6	91006	2017-04-05 00:00:00.000	2017-04-12 00:00:00.000
7	91007	2017-04-05 00:00:00.000	2017-04-12 00:00:00.000
8	91008	2017-04-05 00:00:00.000	2017-04-12 00:00:00.000
9	91009	2017-04-05 00:00:00.000	2017-04-19 00:00:00.000
10	91010	2017-04-05 00:00:00.000	2017-04-19 00:00:00.000
11	91011	2017-04-05 00:00:00.000	2017-04-12 00:00:00.000
12	91012	2017-04-08 00:00:00.000	2017-04-15 00:00:00.000
13	91013	2017-04-10 00:00:00.000	2017-04-24 00:00:00.000
14	91014	2017-04-11 00:00:00.000	2017-04-18 00:00:00.000
15	91015	2017-04-11 00:00:00.000	2017-04-18 00:00:00.000
16	91016	2017-04-13 00:00:00.000	2017-04-27 00:00:00.000
17	91017	2017-04-14 00:00:00.000	2017-04-21 00:00:00.000
18	91018	2017-04-14 00:00:00.000	2017-04-28 00:00:00.000
19	91019	2017-04-14 00:00:00.000	2017-04-21 00:00:00.000
20	91020	2017-04-16 00:00:00.000	2017-04-23 00:00:00.000
21	91021	2017-04-16 00:00:00.000	2017-04-23 00:00:00.000
22	91022	2017-04-16 00:00:00.000	2017-04-23 00:00:00.000
23	91023	2017-04-16 00:00:00.000	2017-04-23 00:00:00.000
24	91024	2017-04-21 00:00:00.000	2017-04-28 00:00:00.000
25	91025	2017-04-21 00:00:00.000	2017-04-28 00:00:00.000

Query executed successfully.

USER-PC (14.0 RTM) | USER-PC\User (56) | Ch07\_Fact | 00:00:00 | 68 rows

59.

SELECT

```
BOOK.BOOK_NUM,  
BOOK.BOOK_TITLE AS TITLE,  
BOOK.BOOK_SUBJECT AS 'Subject of Book'
```

FROM

```
BOOK
```

ORDER BY

```
BOOK.BOOK_NUM;
```

SQLQuery1.sql - US...USER-PC\User (56))  
1 SELECT  
2 BOOK.BOOK\_NUM,  
3 BOOK.BOOK\_TITLE AS TITLE,  
4 BOOK.BOOK\_SUBJECT AS 'Subject of Book'  
5 FROM  
6 BOOK  
7 ORDER BY  
8 BOOK.BOOK\_NUM;  
9

91 %

Results Messages

	BOOK_NUM	TITLE	Subject of Book
1	5235	Beginner's Guide to JAVA	Programming
2	5236	Database in the Cloud	Cloud
3	5237	Mastering the database environment	Database
4	5238	Conceptual Programming	Programming
5	5239	J++ in Mobile Apps	Programming
6	5240	iOS Programming	Programming
7	5241	JAVA First Steps	Programming
8	5242	C# in Middleware Deployment	Middleware
9	5243	DATABASES in Theory	Database
10	5244	Cloud-based Mobile Applications	Cloud
11	5245	The Golden Road to Platform independence	Middleware
12	5246	Capture the Cloud	Cloud
13	5247	Shining Through the Cloud: Sun Programming	Programming
14	5248	What You Always Wanted to Know About Database, B...	Database
15	5249	Starlight Applications	Cloud
16	5250	Reengineering the Middle Tier	Middleware
17	5251	Thoughts on Revitalizing Ruby	Programming
18	5252	Beyond the Database Veil	Database
19	5253	Virtual Programming for Virtual Environments	Programming
20	5254	Coding Style for Maintenance	Programming

Query executed successfully. USER-PC (14.0 RTM) USER-PC\User (56) Ch07\_Fact 00:00:00 20 rows

60.

```
SELECT DISTINCT
    BOOK.BOOK_YEAR
FROM
    BOOK
ORDER BY
    BOOK.BOOK_YEAR;
```

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT DISTINCT
2     BOOK.BOOK_YEAR
3 FROM
4     BOOK
5 ORDER BY
6     BOOK.BOOK_YEAR;
7
```

91 %

Results Messages

	BOOK_YEAR
1	2014
2	2015
3	2016
4	2017

61.

```
SELECT
    BOOK_SUBJECT
FROM
    BOOK
GROUP BY
    BOOK_SUBJECT;
```

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT
2     BOOK_SUBJECT
3 FROM
4     BOOK
5 GROUP BY
6     BOOK_SUBJECT;
7
```

91 %

Results Messages

	BOOK_SUBJECT
1	Cloud
2	Database
3	Middleware
4	Programming

62.

SELECT

```
BOOK.BOOK_NUM,  
BOOK.BOOK_TITLE,  
BOOK.BOOK_COST AS 'Replacement Cost'
```

FROM

BOOK

ORDER BY BOOK.BOOK\_NUM;

The screenshot shows a SQL Server Enterprise Manager window with a query editor and a results pane. The query editor contains the following SQL code:

```
1 SELECT  
2     BOOK.BOOK_NUM,  
3     BOOK.BOOK_TITLE,  
4     BOOK.BOOK_COST AS 'Replacement Cost'  
5 FROM  
6     BOOK  
7  
8 ORDER BY BOOK.BOOK_NUM;  
9
```

The results pane shows a table with 20 rows. The columns are BOOK\_NUM, BOOK\_TITLE, and Replacement Cost. The first row is highlighted.

	BOOK_NUM	BOOK_TITLE	Replacement Cost
1	5235	Beginner's Guide to JAVA	59.95
2	5236	Database in the Cloud	79.95
3	5237	Mastering the database environment	89.95
4	5238	Conceptual Programming	59.95
5	5239	J++ in Mobile Apps	49.95
6	5240	iOS Programming	79.95
7	5241	JAVA First Steps	49.95
8	5242	C# in Middleware Deployment	59.95
9	5243	DATABASES in Theory	129.95
10	5244	Cloud-based Mobile Applications	69.95
11	5245	The Golden Road to Platform independence	119.95
12	5246	Capture the Cloud	69.95
13	5247	Shining Through the Cloud: Sun Programming	109.95
14	5248	What You Always Wanted to Know About Database, B...	49.95
15	5249	Starlight Applications	69.95
16	5250	Reengineering the Middle Tier	89.95
17	5251	Thoughts on Revitalizing Ruby	59.95
18	5252	Beyond the Database Veil	69.95
19	5253	Virtual Programming for Virtual Environments	79.95
20	5254	Coding Style for Maintenance	49.95

Query executed successfully. USER-PC (14.0 RTM) | USER-PC\User (56) | Ch07\_Fact | 00:00:00 | 20 rows

63.

SELECT

```
CHECKOUT.CHECK_NUM,  
CHECKOUT.BOOK_NUM,  
CHECKOUT.PAT_ID,  
CHECKOUT.CHECK_OUT_DATE,  
CHECKOUT.CHECK_DUE_DATE
```

FROM

```
CHECKOUT
```

ORDER BY

```
CHECKOUT.CHECK_OUT_DATE DESC,  
CHECKOUT.CHECK_NUM;
```

SQLQuery1.sql - US...USER-PC\User (56) \* X

```
1 SELECT  
2   CHECKOUT.CHECK_NUM,  
3   CHECKOUT.BOOK_NUM,  
4   CHECKOUT.PAT_ID,  
5   CHECKOUT.CHECK_OUT_DATE,  
6   CHECKOUT.CHECK_DUE_DATE  
7 FROM  
8   CHECKOUT  
9 ORDER BY  
10  CHECKOUT.CHECK_OUT_DATE DESC,  
11  CHECKOUT.CHECK_NUM;  
12
```

91 %

Results Messages

	CHECK_NUM	BOOK_NUM	PAT_ID	CHECK_OUT_DATE	CHECK_DUE_DATE
1	91067	5252	1229	2017-05-24 00:00:00.000	2017-05-31 00:00:00.000
2	91068	5238	1229	2017-05-24 00:00:00.000	2017-05-31 00:00:00.000
3	91066	5242	1228	2017-05-19 00:00:00.000	2017-05-26 00:00:00.000
4	91064	5236	1183	2017-05-17 00:00:00.000	2017-05-31 00:00:00.000
5	91065	5244	1210	2017-05-17 00:00:00.000	2017-05-24 00:00:00.000
6	91060	5235	1209	2017-05-15 00:00:00.000	2017-05-22 00:00:00.000
7	91061	5246	1172	2017-05-15 00:00:00.000	2017-05-22 00:00:00.000
8	91062	5254	1223	2017-05-15 00:00:00.000	2017-05-22 00:00:00.000
9	91063	5243	1223	2017-05-15 00:00:00.000	2017-05-22 00:00:00.000
10	91066	5254	1224	2017-05-10 00:00:00.000	2017-05-17 00:00:00.000
11	91057	5238	1224	2017-05-10 00:00:00.000	2017-05-17 00:00:00.000
12	91058	5252	1171	2017-05-10 00:00:00.000	2017-05-17 00:00:00.000
13	91059	5249	1207	2017-05-10 00:00:00.000	2017-05-17 00:00:00.000
14	91053	5240	1212	2017-05-09 00:00:00.000	2017-05-16 00:00:00.000
15	91054	5236	1221	2017-05-09 00:00:00.000	2017-05-16 00:00:00.000
16	91055	5246	1221	2017-05-09 00:00:00.000	2017-05-16 00:00:00.000
17	91060	5236	1220	2017-05-08 00:00:00.000	2017-05-15 00:00:00.000
18	91051	5237	1222	2017-05-08 00:00:00.000	2017-05-15 00:00:00.000
19	91052	5236	1213	2017-05-08 00:00:00.000	2017-05-15 00:00:00.000
20	91048	5249	1229	2017-05-04 00:00:00.000	2017-05-11 00:00:00.000
21	91049	5240	1214	2017-05-04 00:00:00.000	2017-05-11 00:00:00.000
22	91044	5248	1219	2017-04-30 00:00:00.000	2017-05-07 00:00:00.000
23	91045	5242	1210	2017-04-30 00:00:00.000	2017-05-07 00:00:00.000
24	91046	5235	1225	2017-04-30 00:00:00.000	2017-05-07 00:00:00.000
25	91047	5236	1218	2017-04-30 00:00:00.000	2017-05-07 00:00:00.000

Query executed successfully.

USER-PC (14.0 RTM) USER-PC\User (56) Ch07\_Fact 00:00:00 68 rows



64.

SELECT

```
BOOK.BOOK_TITLE,  
BOOK.BOOK_YEAR,  
BOOK.BOOK_SUBJECT
```

FROM

```
BOOK
```

ORDER BY

```
BOOK.BOOK_SUBJECT,  
BOOK.BOOK_YEAR DESC,  
BOOK.BOOK_TITLE;
```

SQLQuery1.sql - US...USER-PC\User (56))\* X

```
1 SELECT  
2   BOOK.BOOK_TITLE,  
3   BOOK.BOOK_YEAR,  
4   BOOK.BOOK_SUBJECT  
5 FROM  
6   BOOK  
7 ORDER BY  
8   BOOK.BOOK_SUBJECT,  
9   BOOK.BOOK_YEAR DESC,  
10  BOOK.BOOK_TITLE;  
11
```

91 %

Results Messages

	BOOK_TITLE	BOOK_YEAR	BOOK_SUBJECT
1	Capture the Cloud	2016	Cloud
2	Starlight Applications	2016	Cloud
3	Cloud-based Mobile Applications	2015	Cloud
4	Database in the Cloud	2014	Cloud
5	Beyond the Database Veil	2016	Database
6	What You Always Wanted to Know About Database, B...	2016	Database
7	DATABASES in Theory	2015	Database
8	Mastering the database environment	2015	Database
9	Reengineering the Middle Tier	2016	Middleware
10	The Golden Road to Platform independence	2016	Middleware
11	C# in Middleware Deployment	2015	Middleware
12	Coding Style for Maintenance	2017	Programming
13	Shining Through the Cloud: Sun Programming	2016	Programming
14	Thoughts on Revitalizing Ruby	2016	Programming
15	Virtual Programming for Virtual Environments	2016	Programming
16	Conceptual Programming	2015	Programming
17	iOS Programming	2015	Programming
18	J++ in Mobile Apps	2015	Programming
19	JAVA First Steps	2015	Programming
20	Beginner's Guide to JAVA	2014	Programming

Query executed successfully.

USER-PC (14.0 RTM) | USER-PC\User (56) | Ch07\_Fact | 00:00:00 | 20 rows

65.

SELECT

BOOK.BOOK\_NUM,  
BOOK.BOOK\_TITLE,  
BOOK.BOOK\_COST

FROM

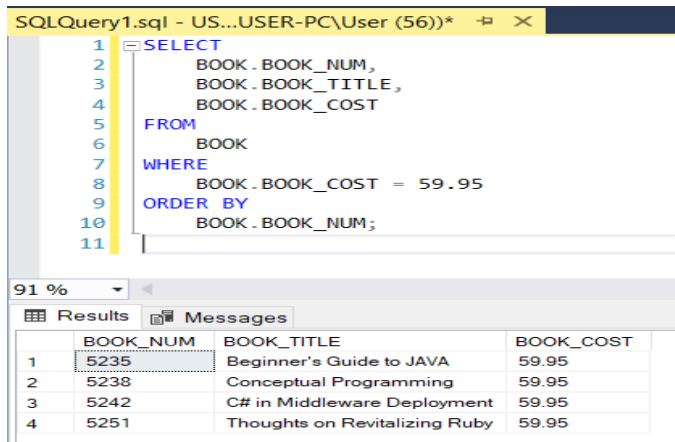
BOOK

WHERE

BOOK.BOOK\_COST = 59.95

ORDER BY

BOOK.BOOK\_NUM;



The screenshot shows a SQL Query Editor window titled "SQLQuery1.sql - US...USER-PC\User (56))". The query is as follows:

```
1 SELECT
2     BOOK.BOOK_NUM,
3     BOOK.BOOK_TITLE,
4     BOOK.BOOK_COST
5 FROM
6     BOOK
7 WHERE
8     BOOK.BOOK_COST = 59.95
9 ORDER BY
10    BOOK.BOOK_NUM;
11
```

The Results tab is active, displaying a table with 4 rows and 3 columns: BOOK\_NUM, BOOK\_TITLE, and BOOK\_COST.

	BOOK_NUM	BOOK_TITLE	BOOK_COST
1	5235	Beginner's Guide to JAVA	59.95
2	5238	Conceptual Programming	59.95
3	5242	C# in Middleware Deployment	59.95
4	5251	Thoughts on Revitalizing Ruby	59.95

66.

SELECT

BOOK.BOOK\_NUM,  
BOOK.BOOK\_TITLE,  
BOOK.BOOK\_COST

FROM

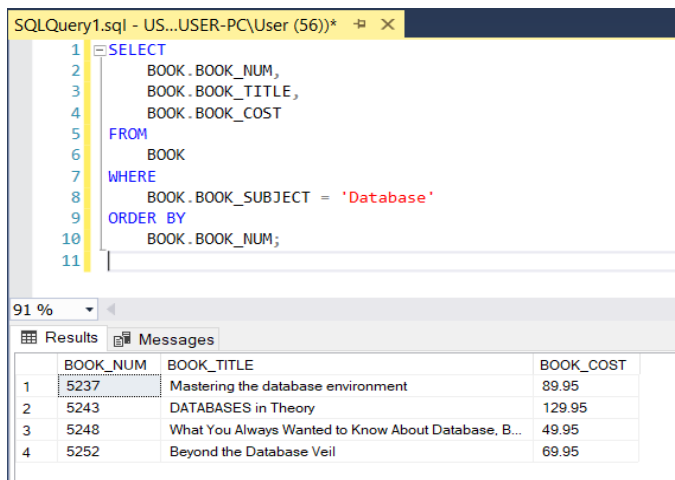
BOOK

WHERE

BOOK.BOOK\_SUBJECT = 'Database'

ORDER BY

BOOK.BOOK\_NUM;



The screenshot shows a SQL Query Editor window titled "SQLQuery1.sql - US...USER-PC\User (56))". The query is as follows:

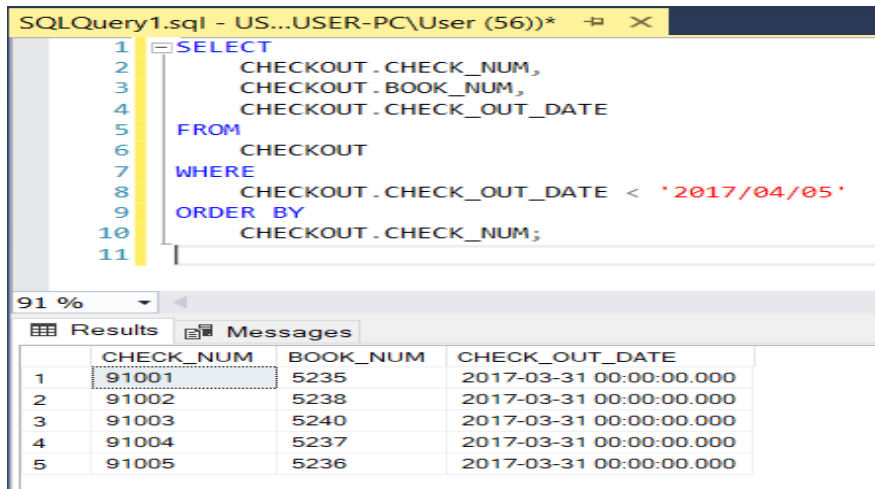
```
1 SELECT
2     BOOK.BOOK_NUM,
3     BOOK.BOOK_TITLE,
4     BOOK.BOOK_COST
5 FROM
6     BOOK
7 WHERE
8     BOOK.BOOK_SUBJECT = 'Database'
9 ORDER BY
10    BOOK.BOOK_NUM;
11
```

The Results tab is active, displaying a table with 4 rows and 3 columns: BOOK\_NUM, BOOK\_TITLE, and BOOK\_COST.

	BOOK_NUM	BOOK_TITLE	BOOK_COST
1	5237	Mastering the database environment	89.95
2	5243	DATABASES in Theory	129.95
3	5248	What You Always Wanted to Know About Database, B...	49.95
4	5252	Beyond the Database Veil	69.95

67.

```
SELECT
    CHECKOUT.CHECK_NUM,
    CHECKOUT.BOOK_NUM,
    CHECKOUT.CHECK_OUT_DATE
FROM
    CHECKOUT
WHERE
    CHECKOUT.CHECK_OUT_DATE < '2017/04/05'
ORDER BY
    CHECKOUT.CHECK_NUM;
```



The screenshot shows a SQL Query Editor window titled "SQLQuery1.sql - US...USER-PC\User (56))\*". The query is as follows:

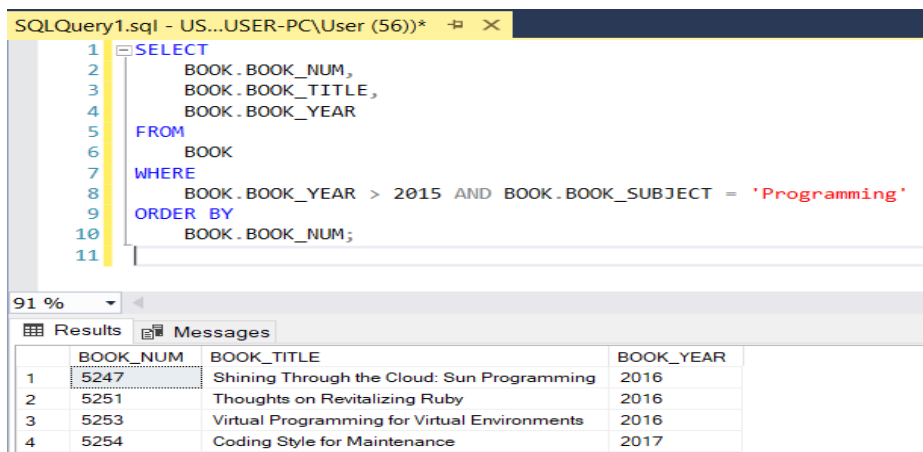
```
1 SELECT
2     CHECKOUT.CHECK_NUM,
3     CHECKOUT.BOOK_NUM,
4     CHECKOUT.CHECK_OUT_DATE
5 FROM
6     CHECKOUT
7 WHERE
8     CHECKOUT.CHECK_OUT_DATE < '2017/04/05'
9 ORDER BY
10    CHECKOUT.CHECK_NUM;
11
```

The results pane shows 5 rows of data:

	CHECK_NUM	BOOK_NUM	CHECK_OUT_DATE
1	91001	5235	2017-03-31 00:00:00.000
2	91002	5238	2017-03-31 00:00:00.000
3	91003	5240	2017-03-31 00:00:00.000
4	91004	5237	2017-03-31 00:00:00.000
5	91005	5236	2017-03-31 00:00:00.000

68.

```
SELECT
    BOOK.BOOK_NUM,
    BOOK.BOOK_TITLE,
    BOOK.BOOK_YEAR
FROM
    BOOK
WHERE
    BOOK.BOOK_YEAR > 2015 AND BOOK.BOOK_SUBJECT = 'Programming'
ORDER BY
    BOOK.BOOK_NUM;
```



The screenshot shows a SQL Query Editor window titled "SQLQuery1.sql - US...USER-PC\User (56))\*". The query is as follows:

```
1 SELECT
2     BOOK.BOOK_NUM,
3     BOOK.BOOK_TITLE,
4     BOOK.BOOK_YEAR
5 FROM
6     BOOK
7 WHERE
8     BOOK.BOOK_YEAR > 2015 AND BOOK.BOOK_SUBJECT = 'Programming'
9 ORDER BY
10    BOOK.BOOK_NUM;
11
```

The results pane shows 4 rows of data:

	BOOK_NUM	BOOK_TITLE	BOOK_YEAR
1	5247	Shining Through the Cloud: Sun Programming	2016
2	5251	Thoughts on Revitalizing Ruby	2016
3	5253	Virtual Programming for Virtual Environments	2016
4	5254	Coding Style for Maintenance	2017

69.

```
SELECT
    BOOK.BOOK_NUM,
    BOOK.BOOK_TITLE,
    BOOK.BOOK_SUBJECT,
    BOOK.BOOK_COST
FROM
    BOOK
WHERE
    BOOK.BOOK_SUBJECT = 'Middleware'
    AND BOOK.BOOK_COST >= 70
    OR BOOK.BOOK_SUBJECT = 'Cloud'
    AND BOOK.BOOK_COST >= 70
ORDER BY
    BOOK.BOOK_NUM;
```

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT
2     BOOK.BOOK_NUM,
3     BOOK.BOOK_TITLE,
4     BOOK.BOOK_SUBJECT,
5     BOOK.BOOK_COST
6 FROM
7     BOOK
8 WHERE
9     BOOK.BOOK_SUBJECT = 'Middleware'
10    AND BOOK.BOOK_COST >= 70
11    OR BOOK.BOOK_SUBJECT = 'Cloud'
12    AND BOOK.BOOK_COST >= 70
13 ORDER BY
14     BOOK.BOOK_NUM;
15
```

91 %

Results Messages

	BOOK_NUM	BOOK_TITLE	BOOK_SUBJECT	BOOK_COST
1	5236	Database in the Cloud	Cloud	79.95
2	5245	The Golden Road to Platform independence	Middleware	119.95
3	5250	Reengineering the Middle Tier	Middleware	89.95

70.

SELECT

AUTHOR.AU\_ID,  
AUTHOR.AU\_FNAME,  
AUTHOR.AU\_LNAME,  
AUTHOR.AU\_BIRTHYEAR

FROM

AUTHOR

WHERE

AUTHOR.AU\_BIRTHYEAR BETWEEN 1980 AND 1989

ORDER BY AUTHOR.AU\_ID;

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT
2     AUTHOR.AU_ID,
3     AUTHOR.AU_FNAME,
4     AUTHOR.AU_LNAME,
5     AUTHOR.AU_BIRTHYEAR
6 FROM
7     AUTHOR
8 WHERE
9     AUTHOR.AU_BIRTHYEAR BETWEEN 1980 AND 1989
10 ORDER BY AUTHOR.AU_ID;
11
```

91 %

Results Messages

	AU_ID	AU_FNAME	AU_LNAME	AU_BIRTHYEAR
1	218	Rachel	Beatney	1983
2	383	Neal	Walsh	1980
3	394	Robert	Lake	1982
4	438	Perry	Pearson	1986
5	460	Connie	Paulsen	1983
6	581	Manish	Aggerwal	1984
7	603	Julia	Palca	1988

71.

SELECT

BOOK.BOOK\_NUM,  
BOOK.BOOK\_TITLE,  
BOOK.BOOK\_SUBJECT

FROM

BOOK

WHERE

BOOK.BOOK\_TITLE LIKE '%Database%'

ORDER BY

BOOK.BOOK\_NUM;

The screenshot shows a SQL Server Enterprise Manager window titled "SQLQuery1.sql - US...USER-PC\User (56))". The query editor displays the following SQL query:

```
1 SELECT
2     BOOK.BOOK_NUM,
3     BOOK.BOOK_TITLE,
4     BOOK.BOOK_SUBJECT
5 FROM
6     BOOK
7 WHERE
8     BOOK.BOOK_TITLE LIKE '%Database%'
9 ORDER BY
10    BOOK.BOOK_NUM;
11
```

Below the query editor, the "Results" tab is active, showing a table with 5 rows and 4 columns. The columns are BOOK\_NUM, BOOK\_TITLE, and BOOK\_SUBJECT. The first row is highlighted with a dotted border.

	BOOK_NUM	BOOK_TITLE	BOOK_SUBJECT
1	5236	Database in the Cloud	Cloud
2	5237	Mastering the database environment	Database
3	5243	DATABASES in Theory	Database
4	5248	What You Always Wanted to Know About Database, B...	Database
5	5252	Beyond the Database Veil	Database

72.

SELECT

PATRON.PAT\_ID,  
PATRON.PAT\_FNAME,  
PATRON.PAT\_LNAME

FROM

PATRON

WHERE

PATRON.PAT\_TYPE = 'Student'

ORDER BY

PATRON.PAT\_ID;

SQLQuery1.sql - US...USER-PC\User (56)) \* X

```
1 SELECT
2     PATRON.PAT_ID,
3     PATRON.PAT_FNAME,
4     PATRON.PAT_LNAME
5 FROM
6     PATRON
7 WHERE
8     PATRON.PAT_TYPE = 'Student'
9 ORDER BY
10    PATRON.PAT_ID;
```

91 %

Results Messages

	PAT_ID	PAT_FNAME	PAT_LNAME
1	1166	Vera	Alvarado
2	1171	Peggy	Marsh
3	1172	Tony	Miles
4	1174	Betsy	Malone
5	1180	Nadine	Blair
6	1181	Allen	Horne
7	1182	Jamal	Melendez
8	1184	Jimmie	Love
9	1185	Sandra	Yang
10	1200	Lorenzo	Torres
11	1201	Shelby	Noble
12	1202	Holly	Anthony
13	1203	Tyler	Pope
14	1204	Thomas	Duran
15	1205	Claire	Gomez
16	1207	Iva	Ramos
17	1208	Ollie	Cantrell
18	1209	Rena	Mathis
19	1210	Keith	Cooley
20	1211	Jerald	Gaines
21	1212	Iva	McClain

Query executed successfully. USER-PC (14.0 RTM) | USER-PC\User (56) | Ch07\_Fact | 00:00:00 44 rows

73.

SELECT

PATRON.PAT\_ID,  
PATRON.PAT\_FNAME,  
PATRON.PAT\_LNAME,  
PATRON.PAT\_TYPE

FROM

PATRON

WHERE

PATRON.PAT\_LNAME LIKE 'C%'

ORDER BY

PATRON.PAT\_ID;

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT
2     PATRON.PAT_ID,
3     PATRON.PAT_FNAME,
4     PATRON.PAT_LNAME,
5     PATRON.PAT_TYPE
6 FROM
7     PATRON
8 WHERE
9     PATRON.PAT_LNAME LIKE 'C%'
10 ORDER BY
11     PATRON.PAT_ID;
12
```

91 %

Results Messages

	PAT_ID	PAT_FNAME	PAT_LNAME	PAT_TYPE
1	1160	robert	carter	Faculty
2	1208	Ollie	Cantrell	Student
3	1210	Keith	Cooley	STUdent



74.

SELECT

AUTHOR.AU\_ID,  
AUTHOR.AU\_FNAME,  
AUTHOR.AU\_LNAME

FROM

AUTHOR

WHERE

AUTHOR.AU\_BIRTHYEAR IS NULL

ORDER BY

AUTHOR.AU\_ID;

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT
2     AUTHOR.AU_ID,
3     AUTHOR.AU_FNAME,
4     AUTHOR.AU_LNAME
5 FROM
6     AUTHOR
7 WHERE
8     AUTHOR.AU_BIRTHYEAR IS NULL
9 ORDER BY
10    AUTHOR.AU_ID;
11
```

91 %

Results Messages

	AU_ID	AU_FNAME	AU_LNAME
1	229	Carmine	Salvadore
2	262	Xia	Chiang
3	559	Rachel	McGill

75.

SELECT

AUTHOR.AU\_ID,  
AUTHOR.AU\_FNAME,  
AUTHOR.AU\_LNAME

FROM

AUTHOR

WHERE

AUTHOR.AU\_BIRTHYEAR IS NOT NULL

ORDER BY

AUTHOR.AU\_ID;

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT
2     AUTHOR.AU_ID,
3     AUTHOR.AU_FNAME,
4     AUTHOR.AU_LNAME
5 FROM
6     AUTHOR
7 WHERE
8     AUTHOR.AU_BIRTHYEAR IS NOT NULL
9 ORDER BY
10    AUTHOR.AU_ID;
11
```

91 %

Results Messages

	AU_ID	AU_FNAME	AU_LNAME
1	185	Benson	Reeves
2	218	Rachel	Beatney
3	251	Hugo	Bruer
4	273	Reba	Durante
5	284	Trina	Tankersly
6	383	Neal	Walsh
7	394	Robert	Lake
8	438	Perry	Pearson
9	460	Connie	Paulsen
10	581	Manish	Aggerwal
11	592	Lawrence	Sheel
12	603	Julia	Palca

76.

SELECT

```
CHECKOUT.CHECK_NUM,  
CHECKOUT.BOOK_NUM,  
CHECKOUT.PAT_ID,  
CHECKOUT.CHECK_OUT_DATE,  
CHECKOUT.CHECK_DUE_DATE
```

FROM

```
CHECKOUT
```

WHERE

```
CHECKOUT.CHECK_IN_DATE IS NULL
```

ORDER BY

```
CHECKOUT.BOOK_NUM;
```

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT  
2     CHECKOUT.CHECK_NUM,  
3     CHECKOUT.BOOK_NUM,  
4     CHECKOUT.PAT_ID,  
5     CHECKOUT.CHECK_OUT_DATE,  
6     CHECKOUT.CHECK_DUE_DATE  
7 FROM  
8     CHECKOUT  
9 WHERE  
10    CHECKOUT.CHECK_IN_DATE IS NULL  
11 ORDER BY  
12    CHECKOUT.BOOK_NUM;  
13
```

91 %



Results



Messages

	CHECK_NUM	BOOK_NUM	PAT_ID	CHECK_OUT_DATE	CHECK_DUE_DATE
1	91068	5238	1229	2017-05-24 00:00:00.000	2017-05-31 00:00:00.000
2	91053	5240	1212	2017-05-09 00:00:00.000	2017-05-16 00:00:00.000
3	91066	5242	1228	2017-05-19 00:00:00.000	2017-05-26 00:00:00.000
4	91061	5246	1172	2017-05-15 00:00:00.000	2017-05-22 00:00:00.000
5	91059	5249	1207	2017-05-10 00:00:00.000	2017-05-17 00:00:00.000
6	91067	5252	1229	2017-05-24 00:00:00.000	2017-05-31 00:00:00.000

77.

SELECT

```
AUTHOR.AU_ID,  
AUTHOR.AU_FNAME,  
AUTHOR.AU_LNAME,  
AUTHOR.AU_BIRTHYEAR
```

FROM

```
AUTHOR
```

ORDER BY

```
AUTHOR.AU_BIRTHYEAR DESC,  
AUTHOR.AU_LNAME;
```

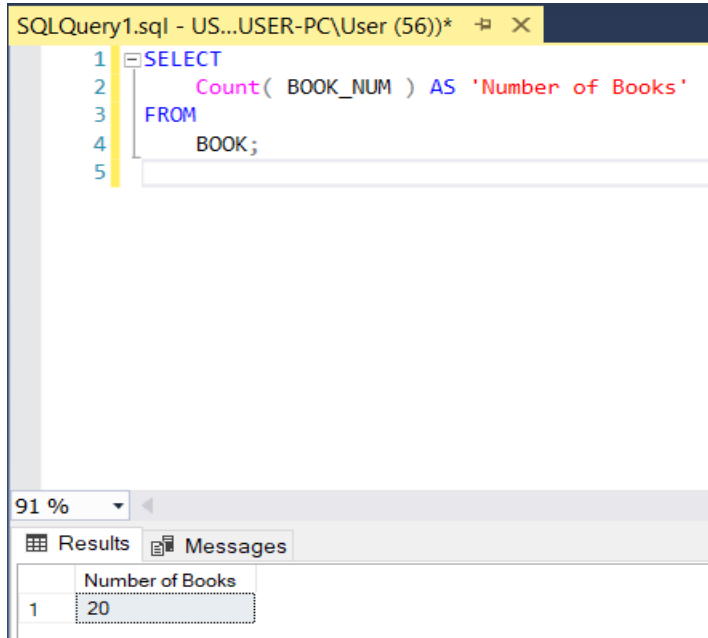
SQLQuery1.sql - US...USER-PC\User (56))*				
1	SELECT			
2		AUTHOR.AU_ID,		
3		AUTHOR.AU_FNAME,		
4		AUTHOR.AU_LNAME,		
5		AUTHOR.AU_BIRTHYEAR		
6	FROM			
7		AUTHOR		
8	ORDER BY			
9		AUTHOR.AU_BIRTHYEAR DESC,		
10		AUTHOR.AU_LNAME;		
11				

91 %				
Results Messages				
	AU_ID	AU_FNAME	AU_LNAME	AU_BIRTHYEAR
1	185	Benson	Reeves	1990
2	603	Julia	Palca	1988
3	438	Perry	Pearson	1986
4	581	Manish	Aggerwal	1984
5	218	Rachel	Beatney	1983
6	460	Connie	Paulsen	1983
7	394	Robert	Lake	1982
8	383	Neal	Walsh	1980
9	592	Lawrence	Sheel	1976
10	251	Hugo	Bruer	1972
11	273	Reba	Durante	1969
12	284	Trina	Tankersly	1961
13	262	Xia	Chiang	NULL
14	559	Rachel	McGill	NULL
15	229	Carmine	Salvadore	NULL

78.

```
SELECT  
    Count( BOOK_NUM ) AS 'Number of Books'  
FROM  
    BOOK;
```



The screenshot shows a SQL query window titled 'SQLQuery1.sql - US...USER-PC\User (56))\*'. The query is as follows:

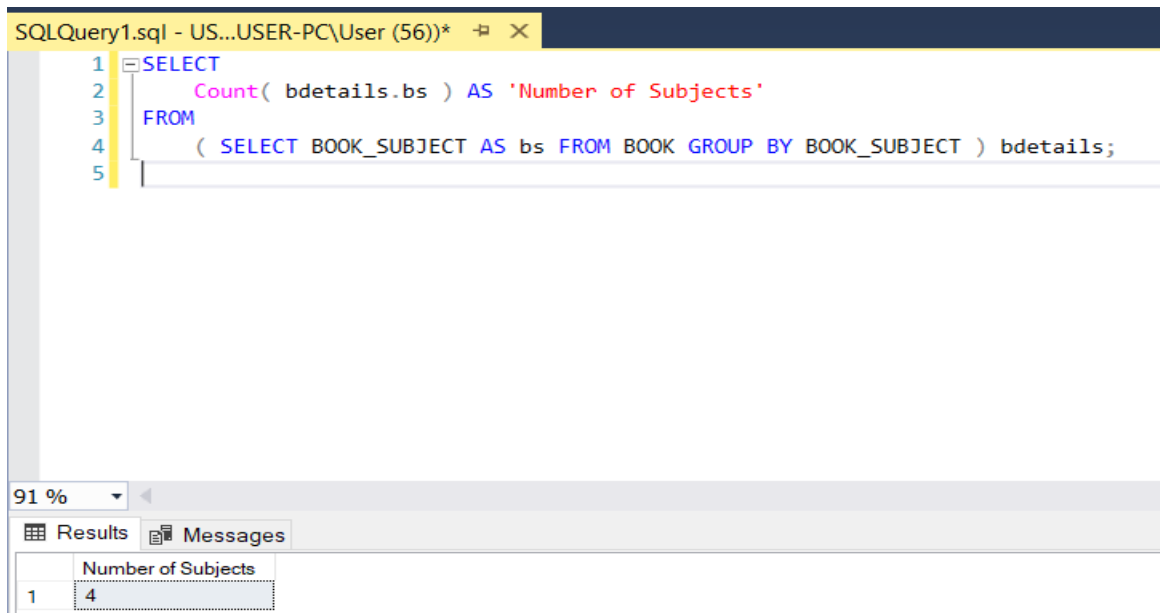
```
1 SELECT  
2     Count( BOOK_NUM ) AS 'Number of Books'  
3 FROM  
4     BOOK;  
5
```

Below the query window, the 'Results' tab is active, displaying a table with one row and one column:

	Number of Books
1	20

79.

```
SELECT  
    Count( bdetails.bs ) AS 'Number of Subjects'  
FROM  
    ( SELECT BOOK_SUBJECT AS bs FROM BOOK GROUP BY BOOK_SUBJECT ) bdetails;
```



The screenshot shows a SQL query window titled 'SQLQuery1.sql - US...USER-PC\User (56))\*'. The query is as follows:

```
1 SELECT  
2     Count( bdetails.bs ) AS 'Number of Subjects'  
3 FROM  
4     ( SELECT BOOK_SUBJECT AS bs FROM BOOK GROUP BY BOOK_SUBJECT ) bdetails;  
5
```

Below the query window, the 'Results' tab is active, displaying a table with one row and one column:

	Number of Subjects
1	4

80.

```
SELECT
    Count( BOOK_NUM ) AS 'Available Books'
FROM
    BOOK
WHERE
    BOOK.PAT_ID IS NULL;
```

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT
2     Count( BOOK_NUM ) AS 'Available Books'
3 FROM
4     BOOK
5 WHERE
6     BOOK.PAT_ID IS NULL;
7
```

91 %

Results Messages

	Available Books
1	14

81.

```
SELECT
    Max( BOOK_COST ) AS 'Most Expensive'
FROM
    BOOK;
```

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT
2     Max( BOOK_COST ) AS 'Most Expensive'
3 FROM
4     BOOK;
5
```

91 %

Results Messages

	Most Expensive
1	129.95

82.

```
SELECT
    Min( BOOK_COST ) AS 'Least Expensive'
FROM
    BOOK;
```

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT
2     Min( BOOK_COST ) AS 'Least Expensive'
3 FROM
4     BOOK;
```

91 %

Results Messages

	Least Expensive
1	49.95

83.

```
SELECT
    Count( pidSet.pid ) AS 'DIFFERENT PATRONS'
FROM
    ( SELECT DISTINCT PAT_ID AS pid FROM CHECKOUT GROUP BY PAT_ID ) AS pidSet;
```

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT
2     Count( pidSet.pid ) AS 'DIFFERENT PATRONS'
3 FROM
4     ( SELECT DISTINCT PAT_ID AS pid FROM CHECKOUT GROUP BY PAT_ID ) AS pidSet;
```

91 %

Results Messages

	DIFFERENT PATRONS
1	33

84.

```
SELECT
    BOOK.BOOK_SUBJECT,
    Count( BOOK.BOOK_NUM ) AS 'Books in Subject'
FROM
    BOOK
GROUP BY
    BOOK.BOOK_SUBJECT,
    BOOK.BOOK_SUBJECT
ORDER BY
    Count( BOOK.BOOK_NUM ) DESC,
    BOOK.BOOK_SUBJECT;
```

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT
2     BOOK.BOOK_SUBJECT,
3     Count( BOOK.BOOK_NUM ) AS 'Books in Subject'
4 FROM
5     BOOK
6 GROUP BY
7     BOOK.BOOK_SUBJECT,
8     BOOK.BOOK_SUBJECT
9 ORDER BY
10    Count( BOOK.BOOK_NUM ) DESC,
11    BOOK.BOOK_SUBJECT;
12
```

91 %

Results Messages

	BOOK_SUBJECT	Books in Subject
1	Programming	9
2	Cloud	4
3	Database	4
4	Middleware	3



85.

```
SELECT
    WRITES.AU_ID,
    Count( WRITES.BOOK_NUM ) AS 'Books Written'
FROM
    WRITES
GROUP BY
    WRITES.AU_ID
ORDER BY
    Count( WRITES.BOOK_NUM ) DESC,
    WRITES.AU_ID;
```

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT
2     WRITES.AU_ID,
3     Count( WRITES.BOOK_NUM ) AS 'Books Written'
4 FROM
5     WRITES
6 GROUP BY
7     WRITES.AU_ID
8 ORDER BY
9     Count( WRITES.BOOK_NUM ) DESC,
10    WRITES.AU_ID;
11
```

91 %

Results Messages

	AU_ID	Books Written
1	262	3
2	460	3
3	185	2
4	229	2
5	251	2
6	383	2
7	394	2
8	559	2
9	218	1
10	273	1
11	284	1
12	438	1
13	581	1
14	592	1
15	603	1

86.

```
SELECT  
    Sum( BOOK_COST ) AS 'Library Value'  
FROM  
    BOOK;
```

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT  
2     Sum( BOOK_COST ) AS 'Library Value'  
3 FROM  
4     BOOK;  
5
```

91 %

Results Messages

	Library Value
1	1499.00

87.

SELECT

```
CHECKOUT.PAT_ID AS PATRON,  
CHECKOUT.BOOK_NUM AS BOOK,  
DATEDIFF(day, CHECKOUT.CHECK_OUT_DATE, CHECKOUT.CHECK_IN_DATE ) AS 'Days Kept'
```

FROM

```
CHECKOUT
```

ORDER BY

```
DATEDIFF(day, CHECKOUT.CHECK_OUT_DATE, CHECKOUT.CHECK_IN_DATE ) DESC, PAT_ID,  
BOOK_NUM;
```

SQLQuery1.sql - US...USER-PC\User (56)\* X

```
1 SELECT  
2 CHECKOUT.PAT_ID AS PATRON,  
3 CHECKOUT.BOOK_NUM AS BOOK,  
4 DATEDIFF(day, CHECKOUT.CHECK_OUT_DATE, CHECKOUT.CHECK_IN_DATE ) AS 'Days Kept'  
5 FROM  
6 CHECKOUT  
7 ORDER BY  
8 DATEDIFF(day, CHECKOUT.CHECK_OUT_DATE, CHECKOUT.CHECK_IN_DATE ) DESC, PAT_ID, BOOK_NUM;  
9
```

91 %

Results Messages

	PATRON	BOOK	Days Kept
1	1160	5240	9
2	1160	5240	9
3	1165	5235	9
4	1183	5236	8
5	1184	5240	8
6	1185	5240	8
7	1202	5236	8
8	1203	5235	8
9	1204	5236	8
10	1207	5242	8
11	1209	5235	8
12	1219	5248	8
13	1222	5240	8
14	1226	5244	8
15	1165	5252	7
16	1185	5254	7
17	1218	5236	7
18	1222	5237	7
19	1228	5237	7
20	1172	5246	6
21	1181	5236	6
22	1220	5235	6
23	1161	5243	5
24	1161	5254	5

Query executed successfully.

USER-PC (14.0 RTM) USER-PC\User (56) Ch07\_Fact 00:00:00 68 rows

88.

SELECT

```
PATRON.PAT_ID,  
CONCAT( PATRON.pat_fname, ' ', PATRON.pat_lname ) AS 'Patron Name',  
PATRON.PAT_TYPE
```

FROM

```
PATRON
```

ORDER BY PATRON.PAT\_ID;

SQLQuery1.sql - US...USER-PC\User (56)\* # X

```
1 SELECT  
2 PATRON.PAT_ID,  
3 CONCAT( PATRON.pat_fname, ' ', PATRON.pat_lname ) AS 'Patron Name',  
4 PATRON.PAT_TYPE  
5 FROM  
6 PATRON  
7 ORDER BY PATRON.PAT_ID;  
8
```

91 %

Results Messages

	PAT_ID	Patron Name	PAT_TYPE
1	1160	robert carter	Faculty
2	1161	Kelsey Koch	Faculty
3	1165	Cedric Baldwin	Faculty
4	1166	Vera Alvarado	Student
5	1167	Alan Martin	FACULTY
6	1170	Cory Barry	faculty
7	1171	Peggy Marsh	STUDENT
8	1172	Tony Miles	STUDENT
9	1174	Betsy Malone	STUDENT
10	1180	Nadine Blair	STUDENT
11	1181	Allen Horne	Student
12	1182	Jamal Melendez	STUDENT
13	1183	Helena Hughes	Faculty
14	1184	Jimmie Love	Student
15	1185	Sandra Yang	student
16	1200	Lorenzo Torres	Student
17	1201	Shelby Noble	Student
18	1202	Holly Anthony	Student
19	1203	Tyler Pope	STUDENT
20	1204	Thomas Duran	Student
21	1205	Claire Gomez	student
22	1207	Iva Ramos	Student
23	1208	Ollie Cantrell	Student
24	1209	Rena Mathis	Student

Query executed successfully.

USER-PC (14.0 RTM) | USER-PC\User (56) | Ch07\_Fact | 00:00:00 | 50 rows

89.

SELECT

```
BOOK.BOOK_NUM,  
CONCAT( BOOK.BOOK_TITLE, ' (', BOOK.BOOK_YEAR, ')' ) AS BOOK,  
BOOK.BOOK_SUBJECT
```

FROM

BOOK

ORDER BY BOOK.BOOK\_NUM;

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT  
2     BOOK.BOOK_NUM,  
3     CONCAT( BOOK.BOOK_TITLE, ' (', BOOK.BOOK_YEAR, ')' ) AS BOOK,  
4     BOOK.BOOK_SUBJECT  
5 FROM  
6     BOOK  
7 ORDER BY BOOK.BOOK_NUM;  
8
```

91 %

Results Messages

	BOOK_NUM	BOOK	BOOK_SUBJECT
1	5235	Beginner's Guide to JAVA (2014)	Programming
2	5236	Database in the Cloud (2014)	Cloud
3	5237	Mastering the database environment (2015)	Database
4	5238	Conceptual Programming (2015)	Programming
5	5239	J++ in Mobile Apps (2015)	Programming
6	5240	iOS Programming (2015)	Programming
7	5241	JAVA First Steps (2015)	Programming
8	5242	C# in Middleware Deployment (2015)	Middleware
9	5243	DATABASES in Theory (2015)	Database
10	5244	Cloud-based Mobile Applications (2015)	Cloud
11	5245	The Golden Road to Platform independence (2016)	Middleware
12	5246	Capture the Cloud (2016)	Cloud
13	5247	Shining Through the Cloud: Sun Programming (2016)	Programming
14	5248	What You Always Wanted to Know About Database, B...	Database
15	5249	Starlight Applications (2016)	Cloud
16	5250	Reengineering the Middle Tier (2016)	Middleware
17	5251	Thoughts on Revitalizing Ruby (2016)	Programming
18	5252	Beyond the Database Veil (2016)	Database
19	5253	Virtual Programming for Virtual Environments (2016)	Programming
20	5254	Coding Style for Maintenance (2017)	Programming

90.

SELECT

```
AUTHOR.AU_LNAME,  
AUTHOR.AU_FNAME,  
WRITES.BOOK_NUM
```

FROM

```
AUTHOR  
JOIN WRITES ON AUTHOR.AU_ID = WRITES.AU_ID
```

ORDER BY

```
AU_LNAME,  
AU_FNAME,  
BOOK_NUM;
```

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT  
2     AUTHOR.AU_LNAME,  
3     AUTHOR.AU_FNAME,  
4     WRITES.BOOK_NUM  
5 FROM  
6     AUTHOR  
7     JOIN WRITES ON AUTHOR.AU_ID = WRITES.AU_ID  
8 ORDER BY  
9     AU_LNAME,  
10    AU_FNAME,  
11    BOOK_NUM;  
12
```

91 %

Results Messages

	AU_LNAME	AU_FNAME	BOOK_NUM
1	Aggerwal	Manish	5242
2	Beatney	Rachel	5240
3	Bruer	Hugo	5243
4	Bruer	Hugo	5246
5	Chiang	Xia	5244
6	Chiang	Xia	5249
7	Chiang	Xia	5252
8	Durante	Reba	5235
9	Lake	Robert	5245
10	Lake	Robert	5247
11	McGill	Rachel	5241
12	McGill	Rachel	5254
13	Palca	Julia	5238
14	Paulsen	Connie	5239
15	Paulsen	Connie	5241
16	Paulsen	Connie	5251
17	Pearson	Perry	5250
18	Reeves	Benson	5237
19	Reeves	Benson	5253
20	Salvadore	Carmine	5239
21	Salvadore	Carmine	5248
22	Sheel	Lawrence	5239
23	Tankersly	Trina	5244
24	Walsh	Neal	5236
25	Walsh	Neal	5250

91.

SELECT

```
WRITES.AU_ID,  
BOOK.BOOK_NUM,  
BOOK.BOOK_TITLE,  
BOOK.BOOK_SUBJECT
```

FROM

```
BOOK  
INNER JOIN WRITES ON BOOK.BOOK_NUM = WRITES.BOOK_NUM
```

ORDER BY

```
BOOK.BOOK_NUM,  
WRITES.AU_ID;
```

SQLQuery1.sql - US...USER-PC\User (56))*				
1	SELECT			
2	WRITES.AU_ID,			
3	BOOK.BOOK_NUM,			
4	BOOK.BOOK_TITLE,			
5	BOOK.BOOK_SUBJECT			
6	FROM			
7	BOOK			
8	INNER JOIN WRITES ON BOOK.BOOK_NUM = WRITES.BOOK_NUM			
9	ORDER BY			
10	BOOK.BOOK_NUM,			
11	WRITES.AU_ID;			
12				
91 %				
Results Messages				
	AU_ID	BOOK_NUM	BOOK_TITLE	BOOK_SUBJECT
1	273	5235	Beginner's Guide to JAVA	Programming
2	383	5236	Database in the Cloud	Cloud
3	185	5237	Mastering the database environment	Database
4	603	5238	Conceptual Programming	Programming
5	229	5239	J++ in Mobile Apps	Programming
6	460	5239	J++ in Mobile Apps	Programming
7	592	5239	J++ in Mobile Apps	Programming
8	218	5240	iOS Programming	Programming
9	460	5241	JAVA First Steps	Programming
10	559	5241	JAVA First Steps	Programming
11	581	5242	C# in Middleware Deployment	Middleware
12	251	5243	DATABASES in Theory	Database
13	262	5244	Cloud-based Mobile Applications	Cloud
14	284	5244	Cloud-based Mobile Applications	Cloud
15	394	5245	The Golden Road to Platform independence	Middleware
16	251	5246	Capture the Cloud	Cloud
17	394	5247	Shining Through the Cloud: Sun Programming	Programming
18	229	5248	What You Always Wanted to Know About Database, B...	Database
19	262	5249	Starlight Applications	Cloud
20	383	5250	Reengineering the Middle Tier	Middleware
21	438	5250	Reengineering the Middle Tier	Middleware
22	460	5251	Thoughts on Revitalizing Ruby	Programming
23	262	5252	Beyond the Database Veil	Database
24	185	5253	Virtual Programming for Virtual Environments	Programming
25	559	5254	Coding Style for Maintenance	Programming

92.

SELECT

```
AUTHOR.AU_LNAME,
AUTHOR.AU_FNAME,
BOOK.BOOK_TITLE,
BOOK.BOOK_COST
```

FROM

```
BOOK
JOIN WRITES ON BOOK.BOOK_NUM = WRITES.BOOK_NUM
JOIN AUTHOR ON AUTHOR.AU_ID = WRITES.AU_ID
```

ORDER BY

```
WRITES.BOOK_NUM,
WRITES.AU_ID;
```

SQLQuery1.sql - US...USER-PC\User (56))\*

```

1 SELECT
2     AUTHOR.AU_LNAME,
3     AUTHOR.AU_FNAME,
4     BOOK.BOOK_TITLE,
5     BOOK.BOOK_COST
6 FROM
7     BOOK
8     JOIN WRITES ON BOOK.BOOK_NUM = WRITES.BOOK_NUM
9     JOIN AUTHOR ON AUTHOR.AU_ID = WRITES.AU_ID
10 ORDER BY
11     WRITES.BOOK_NUM,
12     WRITES.AU_ID;
```

91 %

Results Messages

	AU_LNAME	AU_FNAME	BOOK_TITLE	BOOK_COST
1	Durante	Reba	Beginner's Guide to JAVA	59.95
2	Walsh	Neal	Database in the Cloud	79.95
3	Reeves	Benson	Mastering the database environment	89.95
4	Palca	Julia	Conceptual Programming	59.95
5	Salvadore	Carmine	J++ in Mobile Apps	49.95
6	Paulsen	Connie	J++ in Mobile Apps	49.95
7	Sheel	Lawrence	J++ in Mobile Apps	49.95
8	Beatney	Rachel	iOS Programming	79.95
9	Paulsen	Connie	JAVA First Steps	49.95
10	McGill	Rachel	JAVA First Steps	49.95
11	Aggerwal	Manish	C# in Middleware Deployment	59.95
12	Bruer	Hugo	DATABASES in Theory	129.95
13	Chiang	Xia	Cloud-based Mobile Applications	69.95
14	Tankersly	Trina	Cloud-based Mobile Applications	69.95
15	Lake	Robert	The Golden Road to Platform independence	119.95
16	Bruer	Hugo	Capture the Cloud	69.95
17	Lake	Robert	Shining Through the Cloud: Sun Programming	109.95
18	Salvadore	Carmine	What You Always Wanted to Know About Database, B...	49.95
19	Chiang	Xia	Starlight Applications	69.95
20	Walsh	Neal	Reengineering the Middle Tier	89.95
21	Pearson	Perry	Reengineering the Middle Tier	89.95
22	Paulsen	Connie	Thoughts on Revitalizing Ruby	59.95
23	Chiang	Xia	Beyond the Database Veil	69.95
24	Reeves	Benson	Virtual Programming for Virtual Environments	79.95
25	McGill	Rachel	Coding Style for Maintenance	49.95



93.

SELECT

```
PATRON.PAT_ID,  
BOOK.BOOK_NUM,  
PATRON.PAT_FNAME,  
PATRON.PAT_LNAME,  
BOOK.BOOK_TITLE
```

FROM

```
PATRON  
INNER JOIN BOOK ON PATRON.PAT_ID = BOOK.PAT_ID
```

ORDER BY

```
PATRON.PAT_LNAME,  
BOOK.BOOK_TITLE;
```

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT  
2     PATRON.PAT_ID,  
3     BOOK.BOOK_NUM,  
4     PATRON.PAT_FNAME,  
5     PATRON.PAT_LNAME,  
6     BOOK.BOOK_TITLE  
7 FROM  
8     PATRON  
9     INNER JOIN BOOK ON PATRON.PAT_ID = BOOK.PAT_ID  
10 ORDER BY  
11     PATRON.PAT_LNAME,  
12     BOOK.BOOK_TITLE;  
13
```

91 %

Results Messages

	PAT_ID	BOOK_NUM	PAT_FNAME	PAT_LNAME	BOOK_TITLE
1	1229	5252	Gerald	Burke	Beyond the Database Veil
2	1229	5238	Gerald	Burke	Conceptual Programming
3	1228	5242	Homer	Goodman	C# in Middleware Deployment
4	1212	5240	Iva	McClain	iOS Programming
5	1172	5246	Tony	Miles	Capture the Cloud
6	1207	5249	Iva	Ramos	Starlight Applications

94.

SELECT

```
PATRON.PAT_ID,  
CONCAT( PATRON.PAT_FNAME, ' ', PATRON.PAT_LNAME ) AS NAME,  
PATRON.PAT_TYPE
```

FROM

```
PATRON
```

ORDER BY

```
PATRON.PAT_TYPE,  
PATRON.PAT_LNAME,  
PATRON.PAT_FNAME;
```

SQLQuery1.sql - US...USER-PC\User (56)\*

```
1 SELECT  
2 PATRON.PAT_ID,  
3 CONCAT( PATRON.PAT_FNAME, ' ', PATRON.PAT_LNAME ) AS NAME,  
4 PATRON.PAT_TYPE  
5 FROM  
6 PATRON  
7 ORDER BY  
8 PATRON.PAT_TYPE,  
9 PATRON.PAT_LNAME,  
10 PATRON.PAT_FNAME;  
11
```

91 %

Results Messages

	PAT_ID	NAME	PAT_TYPE
1	1165	Cedric Baldwin	Faculty
2	1170	Cory Barry	faculty
3	1160	robert carter	Faculty
4	1183	Helena Hughes	Faculty
5	1161	Kelsey Koch	Faculty
6	1167	Alan Martin	FACULTY
7	1166	Vera Alvarado	Student
8	1202	Holly Anthony	Student
9	1180	Nadine Blair	STUDENT
10	1238	Erika Bowen	Student
11	1229	Gerald Burke	Student
12	1208	Ollie Cantrell	Student
13	1210	Keith Cooley	STudent
14	1227	Alicia Dickson	Student
15	1215	Maureen Downs	Student
16	1204	Thomas Duran	Student
17	1224	Jewel England	Student
18	1225	Wilfred Fuller	Student
19	1211	Jerald Gaines	Student
20	1205	Claire Gomez	student
21	1228	Homer Goodman	Student
22	1219	Desiree Harrington	Student
23	1223	Jose Hays	Student

Query executed successfully. USER-PC (14.0 RTM) | USER-PC\User (56) | Ch07\_Fact | 00:00:00 | 50 rows

95.

```
SELECT
    CHECKOUT.BOOK_NUM,
    COUNT( CHECKOUT.CHECK_NUM ) AS 'Times Checked Out'
FROM
    CHECKOUT
GROUP BY
    CHECKOUT.BOOK_NUM
ORDER BY
    COUNT( CHECKOUT.CHECK_NUM ) DESC;
```

SQLQuery1.sql - US...USER-PC\User (56)\*

```
1 SELECT
2     CHECKOUT.BOOK_NUM,
3     COUNT( CHECKOUT.CHECK_NUM ) AS 'Times Checked Out'
4 FROM
5     CHECKOUT
6 GROUP BY
7     CHECKOUT.BOOK_NUM
8 ORDER BY
9     COUNT( CHECKOUT.CHECK_NUM ) DESC;
10
```

91 %

Results Messages

	BOOK_NUM	Times Checked Out
1	5236	12
2	5235	9
3	5240	7
4	5238	6
5	5237	5
6	5242	4
7	5244	4
8	5246	4
9	5249	4
10	5252	4
11	5254	4
12	5248	3
13	5243	2

96.

SELECT

AUTHOR.AU\_ID,  
AUTHOR.AU\_FNAME,  
AUTHOR.AU\_LNAME,  
WRITES.BOOK\_NUM,  
BOOK.BOOK\_TITLE

FROM

AUTHOR

INNER JOIN ( BOOK INNER JOIN WRITES ON BOOK.BOOK\_NUM = WRITES.BOOK\_NUM ) ON

AUTHOR.AU\_ID = WRITES.AU\_ID

WHERE

BOOK.BOOK\_SUBJECT = 'Cloud'

ORDER BY

BOOK.BOOK\_TITLE,  
AUTHOR.AU\_LNAME;

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT
2     AUTHOR.AU_ID,
3     AUTHOR.AU_FNAME,
4     AUTHOR.AU_LNAME,
5     WRITES.BOOK_NUM,
6     BOOK.BOOK_TITLE
7 FROM
8     AUTHOR
9     INNER JOIN ( BOOK INNER JOIN WRITES ON BOOK.BOOK_NUM = WRITES.BOOK_NUM ) ON AUTHOR.AU_ID = WRITES.AU_ID
10 WHERE
11     BOOK.BOOK_SUBJECT = 'Cloud'
12 ORDER BY
13     BOOK.BOOK_TITLE,
14     AUTHOR.AU_LNAME;
15
```

91 %

Results

Messages

	AU_ID	AU_FNAME	AU_LNAME	BOOK_NUM	BOOK_TITLE
1	251	Hugo	Bruer	5246	Capture the Cloud
2	262	Xia	Chiang	5244	Cloud-based Mobile Applications
3	284	Trina	Tankersly	5244	Cloud-based Mobile Applications
4	383	Neal	Walsh	5236	Database in the Cloud
5	262	Xia	Chiang	5249	Starlight Applications

97.

SELECT

```
BOOK.BOOK_NUM,  
BOOK.BOOK_TITLE,  
AUTHOR.AU_LNAME,  
AUTHOR.AU_FNAME,  
BOOK.PAT_ID,  
PATRON.PAT_LNAME,  
PATRON.PAT_TYPE
```

FROM

AUTHOR

```
INNER JOIN ( ( PATRON INNER JOIN BOOK ON PATRON.PAT_ID = BOOK.PAT_ID ) INNER JOIN  
WRITES ON BOOK.BOOK_NUM = WRITES.BOOK_NUM ) ON AUTHOR.AU_ID = WRITES.AU_ID
```

ORDER BY

BOOK.BOOK\_TITLE;

```
SQLQuery1.sql - US...USER-PC\User (56)* X  
1 SELECT  
2   BOOK.BOOK_NUM,  
3   BOOK.BOOK_TITLE,  
4   AUTHOR.AU_LNAME,  
5   AUTHOR.AU_FNAME,  
6   BOOK.PAT_ID,  
7   PATRON.PAT_LNAME,  
8   PATRON.PAT_TYPE  
9 FROM  
10  AUTHOR  
11  INNER JOIN ( ( PATRON INNER JOIN BOOK ON PATRON.PAT_ID = BOOK.PAT_ID ) INNER JOIN WRITES ON BOOK.BOOK_NUM = WRITES.BOOK_NUM ) ON AUTHOR.AU_ID = WRITES.AU_ID  
12 ORDER BY  
13   BOOK.BOOK_TITLE;  
14
```

91 %

Results Messages

	BOOK_NUM	BOOK_TITLE	AU_LNAME	AU_FNAME	PAT_ID	PAT_LNAME	PAT_TYPE
1	5252	Beyond the Database Veil	Chiang	Xia	1229	Burke	Student
2	5242	C# in Middleware Deployment	Aggarwal	Manish	1228	Goodman	Student
3	5246	Capture the Cloud	Bruer	Hugo	1172	Miles	STUDENT
4	5238	Conceptual Programming	Palca	Julia	1229	Burke	Student
5	5240	iOS Programming	Beatney	Rachel	1212	McClain	Student
6	5249	Starlight Applications	Chiang	Xia	1207	Ramos	Student

98.

```
SELECT
    BOOK.BOOK_NUM,
    BOOK.BOOK_TITLE,
    Count( CHECKOUT.CHECK_NUM ) AS 'Times Checked Out'
FROM
    BOOK
    LEFT JOIN CHECKOUT ON BOOK.BOOK_NUM = CHECKOUT.BOOK_NUM
GROUP BY
    BOOK.BOOK_NUM,
    BOOK.BOOK_TITLE,
    BOOK.BOOK_TITLE
ORDER BY
    Count( CHECKOUT.CHECK_NUM ) DESC,
    BOOK.BOOK_TITLE;
```

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT
2     BOOK.BOOK_NUM,
3     BOOK.BOOK_TITLE,
4     Count( CHECKOUT.CHECK_NUM ) AS 'Times Checked Out'
5 FROM
6     BOOK
7     LEFT JOIN CHECKOUT ON BOOK.BOOK_NUM = CHECKOUT.BOOK_NUM
8 GROUP BY
9     BOOK.BOOK_NUM,
10    BOOK.BOOK_TITLE,
11    BOOK.BOOK_TITLE
12 ORDER BY
13     Count( CHECKOUT.CHECK_NUM ) DESC,
14     BOOK.BOOK_TITLE;
15
```

91 %

Results

Messages

	BOOK_NUM	BOOK_TITLE	Times Checked Out
1	5236	Database in the Cloud	12
2	5235	Beginner's Guide to JAVA	9
3	5240	iOS Programming	7
4	5238	Conceptual Programming	6
5	5237	Mastering the database environment	5
6	5252	Beyond the Database Veil	4
7	5242	C# in Middleware Deployment	4
8	5246	Capture the Cloud	4
9	5244	Cloud-based Mobile Applications	4
10	5254	Coding Style for Maintenance	4
11	5249	Starlight Applications	4
12	5248	What You Always Wanted to Know About Database, B...	3
13	5243	DATABASES in Theory	2
14	5239	J++ in Mobile Apps	0
15	5241	JAVA First Steps	0
16	5250	Reengineering the Middle Tier	0
17	5247	Shining Through the Cloud: Sun Programming	0
18	5245	The Golden Road to Platform independence	0
19	5251	Thoughts on Revitalizing Ruby	0
20	5253	Virtual Programming for Virtual Environments	0

99.

```
SELECT
    BOOK.BOOK_NUM,
    BOOK.BOOK_TITLE,
    Count( CHECKOUT.CHECK_NUM ) AS 'Times Checked Out'
FROM
    BOOK
    INNER JOIN CHECKOUT ON BOOK.BOOK_NUM = CHECKOUT.BOOK_NUM
GROUP BY
    BOOK.BOOK_NUM,
    BOOK.BOOK_TITLE,
    BOOK.BOOK_TITLE
HAVING
    Count( CHECKOUT.CHECK_NUM ) > 5
ORDER BY
    Count( CHECKOUT.CHECK_NUM ) DESC,
    BOOK.BOOK_TITLE;
```

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT
2     BOOK.BOOK_NUM,
3     BOOK.BOOK_TITLE,
4     Count( CHECKOUT.CHECK_NUM ) AS 'Times Checked Out'
5 FROM
6     BOOK
7     INNER JOIN CHECKOUT ON BOOK.BOOK_NUM = CHECKOUT.BOOK_NUM
8 GROUP BY
9     BOOK.BOOK_NUM,
10    BOOK.BOOK_TITLE,
11    BOOK.BOOK_TITLE
12 HAVING
13     Count( CHECKOUT.CHECK_NUM ) > 5
14 ORDER BY
15     Count( CHECKOUT.CHECK_NUM ) DESC,
16     BOOK.BOOK_TITLE;
17
```

91 %

Results Messages

	BOOK_NUM	BOOK_TITLE	Times Checked Out
1	5236	Database in the Cloud	12
2	5235	Beginner's Guide to JAVA	9
3	5240	iOS Programming	7
4	5238	Conceptual Programming	6

100.

SELECT

```
AUTHOR.AU_ID,  
AUTHOR.AU_LNAME,  
BOOK.BOOK_TITLE,  
CHECKOUT.CHECK_OUT_DATE,  
PATRON.PAT_LNAME
```

FROM

```
AUTHOR  
INNER JOIN (  
    PATRON  
    INNER JOIN (  
        ( BOOK INNER JOIN CHECKOUT ON BOOK.BOOK_NUM = CHECKOUT.BOOK_NUM )  
        INNER JOIN WRITES ON BOOK.BOOK_NUM = WRITES.BOOK_NUM  
    ) ON ( PATRON.PAT_ID = CHECKOUT.PAT_ID )  
    AND ( PATRON.PAT_ID = BOOK.PAT_ID )  
) ON AUTHOR.AU_ID = WRITES.AU_ID
```

WHERE

```
( ( ( AUTHOR.AU_LNAME ) = 'Bruer' ) AND ( ( PATRON.PAT_LNAME ) = 'Miles' ) )
```

ORDER BY

```
CHECKOUT.CHECK_OUT_DATE;
```

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT  
2     AUTHOR.AU_ID,  
3     AUTHOR.AU_LNAME,  
4     BOOK.BOOK_TITLE,  
5     CHECKOUT.CHECK_OUT_DATE,  
6     PATRON.PAT_LNAME  
7 FROM  
8     AUTHOR  
9     INNER JOIN (  
10        PATRON  
11        INNER JOIN (  
12            ( BOOK INNER JOIN CHECKOUT ON BOOK.BOOK_NUM = CHECKOUT.BOOK_NUM )  
13            INNER JOIN WRITES ON BOOK.BOOK_NUM = WRITES.BOOK_NUM  
14        ) ON ( PATRON.PAT_ID = CHECKOUT.PAT_ID )  
15        AND ( PATRON.PAT_ID = BOOK.PAT_ID )  
16    ) ON AUTHOR.AU_ID = WRITES.AU_ID  
17 WHERE  
18     ( ( ( AUTHOR.AU_LNAME ) = 'Bruer' ) AND ( ( PATRON.PAT_LNAME ) = 'Miles' ) )  
19 ORDER BY  
20     CHECKOUT.CHECK_OUT_DATE;  
21
```

91 %

Results Messages

	AU_ID	AU_LNAME	BOOK_TITLE	CHECK_OUT_DATE	PAT_LNAME
1	251	Bruer	Capture the Cloud	2017-04-21 00:00:00.000	Miles
2	251	Bruer	Capture the Cloud	2017-05-15 00:00:00.000	Miles



101.

SELECT

PATRON.PAT\_ID,  
PATRON.PAT\_FNAME,  
PATRON.PAT\_LNAME

FROM

PATRON  
LEFT JOIN CHECKOUT ON PATRON.PAT\_ID = CHECKOUT.PAT\_ID

WHERE

CHECKOUT.CHECK\_NUM IS NULL

ORDER BY

PATRON.PAT\_LNAME,  
PATRON.PAT\_FNAME;

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT
2     PATRON.PAT_ID,
3     PATRON.PAT_FNAME,
4     PATRON.PAT_LNAME
5 FROM
6     PATRON
7     LEFT JOIN CHECKOUT ON PATRON.PAT_ID = CHECKOUT.PAT_ID
8 WHERE
9     CHECKOUT.CHECK_NUM IS NULL
10 ORDER BY
11     PATRON.PAT_LNAME,
12     PATRON.PAT_FNAME;
13
```

91 %

Results Messages

	PAT_ID	PAT_FNAME	PAT_LNAME
1	1166	Vera	Alvarado
2	1180	Nadine	Blair
3	1238	Erika	Bowen
4	1208	Ollie	Cantrell
5	1227	Alicia	Dickson
6	1205	Claire	Gomez
7	1239	Elton	Irwin
8	1240	Jan	Joyce
9	1243	Roberto	Kennedy
10	1242	Mario	King
11	1237	Brandi	Larson
12	1167	Alan	Martin
13	1182	Jamal	Melendez
14	1201	Shelby	Noble
15	1244	Leon	Richmond
16	1200	Lorenzo	Torres
17	1241	Irene	West

102.

SELECT

```
PATRON.PAT_ID,  
PATRON.PAT_LNAME,  
COUNT( CHECK_NUM ) AS 'NUM CHECKOUTS',  
S1.ndb AS 'NUM DIFFERENT BOOKS'
```

FROM

```
(  
  SELECT  
    PAT_ID,  
    COUNT( BOOK_NUM ) AS ndb  
  FROM  
    ( SELECT DISTINCT PAT_ID, BOOK_NUM FROM CHECKOUT ) AS sub2  
  GROUP BY  
    PAT_ID  
  ) AS S1
```

INNER JOIN ( PATRON INNER JOIN CHECKOUT ON PATRON.PAT\_ID = CHECKOUT.PAT\_ID ) ON

S1.PAT\_ID = PATRON.PAT\_ID

GROUP BY

```
PATRON.PAT_ID,  
PATRON.PAT_LNAME,  
S1.ndb
```

HAVING

```
COUNT( CHECK_NUM ) >= 3
```

ORDER BY

```
'NUM DIFFERENT BOOKS' DESC,  
"NUM CHECKOUTS" DESC,  
PATRON.PAT_ID ASC;
```

```
SQLQuery1.sql - US...USER-PC\User (56)*  
1 SELECT  
2 PATRON.PAT_ID,  
3 PATRON.PAT_LNAME,  
4 COUNT( CHECK_NUM ) AS 'NUM CHECKOUTS',  
5 S1.ndb AS 'NUM DIFFERENT BOOKS'  
6 FROM  
7 (  
8   SELECT  
9     PAT_ID,  
10    COUNT( BOOK_NUM ) AS ndb  
11  FROM  
12    ( SELECT DISTINCT PAT_ID, BOOK_NUM FROM CHECKOUT ) AS sub2  
13  GROUP BY  
14    PAT_ID  
15  ) AS S1  
16  INNER JOIN ( PATRON INNER JOIN CHECKOUT ON PATRON.PAT_ID = CHECKOUT.PAT_ID ) ON S1.PAT_ID = PATRON.PAT_ID  
17 GROUP BY  
18 PATRON.PAT_ID,  
19 PATRON.PAT_LNAME,  
20 S1.ndb  
21 HAVING  
22 COUNT( CHECK_NUM ) >= 3  
23 ORDER BY  
24 'NUM DIFFERENT BOOKS' DESC,  
25 "NUM CHECKOUTS" DESC,  
26 PATRON.PAT_ID ASC;  
27
```

Results			
PAT_ID	PAT_LNAME	NUM CHECKOUTS	NUM DIFFERENT BOOKS
1161	Koch	3	3
1165	Baldwin	3	3
1181	Home	3	3
1185	Yang	3	3
1210	Cooley	3	3
1229	Burke	3	3
1160	carter	3	2
1171	Marsh	3	2
1172	Miles	3	2
1207	Ramos	3	2

Query executed successfully.

USER-PC (14.0 RTM) | USER-PC\User (56) | Ch07\_Fact | 00:00:00 | 12 rows

103.

```
SELECT
    ROUND( AVG( DATEDIFF(d, CHECKOUT.CHECK_OUT_DATE, CHECKOUT.CHECK_IN_DATE ) ), 2 )
AS 'Average Days Kept'
FROM
    CHECKOUT;
```

The screenshot shows a SQL Query Editor window titled 'SQLQuery1.sql - US...USER-PC\User (56))'. The query is as follows:

```
1 SELECT
2     ROUND( AVG( DATEDIFF(d, CHECKOUT.CHECK_OUT_DATE, CHECKOUT.CHECK_IN_DATE ) ), 2 ) AS 'Average Days Kept'
3 FROM
4     CHECKOUT;
```

Below the query editor, the 'Results' tab is active, displaying a single row of data:

Average Days Kept
4

104.

```
SELECT
    CHECKOUT.PAT_ID,
    AVG( DATEDIFF(d, CHECKOUT.CHECK_OUT_DATE, CHECKOUT.CHECK_IN_DATE ) ) AS 'Average
Days Kept'
FROM
    CHECKOUT
GROUP BY
    CHECKOUT.PAT_ID
HAVING
    Count( CHECKOUT.CHECK_NUM ) >= 3
ORDER BY
    AVG( DATEDIFF(d, CHECKOUT.CHECK_OUT_DATE, CHECKOUT.CHECK_IN_DATE ) ) DESC;
```

The screenshot shows a SQL Query Editor window titled 'SQLQuery1.sql - US...USER-PC\User (56))'. The query is as follows:

```
1 SELECT
2     CHECKOUT.PAT_ID,
3     AVG( DATEDIFF(d, CHECKOUT.CHECK_OUT_DATE, CHECKOUT.CHECK_IN_DATE ) ) AS 'Average Days Kept'
4 FROM
5     CHECKOUT
6 GROUP BY
7     CHECKOUT.PAT_ID
8 HAVING
9     Count( CHECKOUT.CHECK_NUM ) >= 3
10 ORDER BY
11     AVG( DATEDIFF(d, CHECKOUT.CHECK_OUT_DATE, CHECKOUT.CHECK_IN_DATE ) ) DESC;
```

Below the query editor, the 'Results' tab is active, displaying a list of results:

	PAT_ID	Average Days Kept
1	1160	7
2	1185	6
3	1207	5
4	1209	5
5	1165	5
6	1172	4
7	1183	4
8	1181	3
9	1171	3
10	1161	3
11	1210	2
12	1229	2

105.

SELECT

BOOK.BOOK\_NUM,  
BOOK.BOOK\_TITLE,  
BOOK.BOOK\_COST

FROM

BOOK

WHERE

BOOK.BOOK\_COST = ( SELECT MIN( book\_cost ) FROM book )

ORDER BY

BOOK.BOOK\_NUM;

SQLQuery1.sql - US...USER-PC\User (56))\*

```
1 SELECT
2     BOOK.BOOK_NUM,
3     BOOK.BOOK_TITLE,
4     BOOK.BOOK_COST
5 FROM
6     BOOK
7 WHERE
8     BOOK.BOOK_COST = ( SELECT MIN( book_cost ) FROM book )
9 ORDER BY
10    BOOK.BOOK_NUM;
11
```

91 %



Results



Messages

	BOOK_NUM	BOOK_TITLE	BOOK_COST
1	5239	J++ in Mobile Apps	49.95
2	5241	JAVA First Steps	49.95
3	5248	What You Always Wanted to Know About Database, B...	49.95
4	5254	Coding Style for Maintenance	49.95

106.

SELECT

AUTHOR.AU\_ID,  
AUTHOR.AU\_FNAME,  
AUTHOR.AU\_LNAME

FROM

AUTHOR

WHERE

AUTHOR.AU\_ID NOT IN ( SELECT au\_id FROM writes INNER JOIN book ON writes.book\_num  
= book.book\_num WHERE book\_subject = 'Programming' )

ORDER BY

AUTHOR.AU\_LNAME;

SQLQuery1.sql - US...USER-PC\User (56))\* X

```
1 SELECT
2     AUTHOR.AU_ID,
3     AUTHOR.AU_FNAME,
4     AUTHOR.AU_LNAME
5 FROM
6     AUTHOR
7 WHERE
8     AUTHOR.AU_ID NOT IN ( SELECT au_id FROM writes INNER JOIN book ON writes.book_num = book.book_num WHERE book_subject = 'Programming' )
9 ORDER BY
10    AUTHOR.AU_LNAME;
11
```

91 %

Results Messages

	AU_ID	AU_FNAME	AU_LNAME
1	581	Manish	Aggarwal
2	251	Hugo	Bruer
3	262	Xia	Chiang
4	438	Perry	Pearson
5	284	Trina	Tankersly
6	383	Neal	Walsh

107.

SELECT

```
BOOK.BOOK_NUM,
BOOK.BOOK_TITLE,
BOOK.BOOK_SUBJECT,
ROUND((AVGCOST),2) AS AVGCOST,
ROUND((BOOK.BOOK_COST - AVGCOST),2) AS DIFFERENCE
```

FROM

```
BOOK
INNER JOIN ( SELECT book_subject, round( avg( book_cost ), 2 ) AS avgcost FROM
book GROUP BY book_subject ) AS s1 ON BOOK.BOOK_SUBJECT = s1.book_subject
ORDER BY
BOOK_TITLE;
```

SQLQuery1.sql - US...USER-PC\User (56))\* X

```

1 SELECT
2     BOOK.BOOK_NUM,
3     BOOK.BOOK_TITLE,
4     BOOK.BOOK_SUBJECT,
5     ROUND((AVGCOST),2) AS AVGCOST,
6     ROUND((BOOK.BOOK_COST - AVGCOST),2) AS DIFFERENCE
7 FROM
8     BOOK
9     INNER JOIN ( SELECT book_subject, round( avg( book_cost ), 2 ) AS avgcost FROM book GROUP BY book_subject ) AS s1 ON BOOK.BOOK_SUBJECT = s1.book_subject
10    ORDER BY
11        BOOK_TITLE;
12

```

91 %

Results Messages

	BOOK_NUM	BOOK_TITLE	BOOK_SUBJECT	AVGCOST	DIFFERENCE
1	5235	Beginner's Guide to JAVA	Programming	66.620000	-6.670000
2	5252	Beyond the Database Veil	Database	84.950000	-15.000000
3	5242	C# in Middleware Deployment	Middleware	89.950000	-30.000000
4	5246	Capture the Cloud	Cloud	72.450000	-2.500000
5	5244	Cloud-based Mobile Applications	Cloud	72.450000	-2.500000
6	5254	Coding Style for Maintenance	Programming	66.620000	-16.670000
7	5238	Conceptual Programming	Programming	66.620000	-6.670000
8	5236	Database in the Cloud	Cloud	72.450000	7.500000
9	5243	DATABASES in Theory	Database	84.950000	45.000000
10	5240	iOS Programming	Programming	66.620000	13.330000
11	5239	J++ in Mobile Apps	Programming	66.620000	-16.670000
12	5241	JAVA First Steps	Programming	66.620000	-16.670000
13	5237	Mastering the database environment	Database	84.950000	5.000000
14	5250	Reengineering the Middle Tier	Middleware	89.950000	0.000000
15	5247	Shining Through the Cloud: Sun Programming	Programming	66.620000	43.330000
16	5249	Starlight Applications	Cloud	72.450000	-2.500000
17	5245	The Golden Road to Platform independence	Middleware	89.950000	30.000000
18	5251	Thoughts on Revitalizing Ruby	Programming	66.620000	-6.670000
19	5253	Virtual Programming for Virtual Environments	Programming	66.620000	13.330000
20	5248	What You Always Wanted to Know About Database, B...	Database	84.950000	-35.000000

108.

SELECT

```
BOOK.BOOK_NUM,  
BOOK.BOOK_TITLE,  
BOOK.BOOK_SUBJECT,  
AUTHOR.AU_LNAME,  
s1.numbooks AS 'Num Books by Author'
```

FROM

```
(  
    AUTHOR  
    INNER JOIN ( SELECT au_id, count( book_num ) AS numbooks FROM writes GROUP  
BY au_id ) AS s1 ON AUTHOR.AU_ID = s1.au_id  
)  
INNER JOIN ( BOOK INNER JOIN WRITES ON BOOK.BOOK_NUM = WRITES.BOOK_NUM ) ON  
AUTHOR.AU_ID = WRITES.AU_ID  
WHERE  
    BOOK.BOOK_SUBJECT = 'Cloud'  
ORDER BY  
    BOOK.BOOK_TITLE,  
    AUTHOR.AU_LNAME;
```

SQLQuery1.sql - US...USER-PC\User (56)\*

```
1 SELECT  
2     BOOK.BOOK_NUM,  
3     BOOK.BOOK_TITLE,  
4     BOOK.BOOK_SUBJECT,  
5     AUTHOR.AU_LNAME,  
6     s1.numbooks AS 'Num Books by Author'  
7 FROM  
8     (  
9         AUTHOR  
10        INNER JOIN ( SELECT au_id, count( book_num ) AS numbooks FROM writes GROUP BY au_id ) AS s1 ON AUTHOR.AU_ID = s1.au_id  
11    )  
12    INNER JOIN ( BOOK INNER JOIN WRITES ON BOOK.BOOK_NUM = WRITES.BOOK_NUM ) ON AUTHOR.AU_ID = WRITES.AU_ID  
13 WHERE  
14     BOOK.BOOK_SUBJECT = 'Cloud'  
15 ORDER BY  
16     BOOK.BOOK_TITLE,  
17     AUTHOR.AU_LNAME;  
18
```

91 %

Results Messages

	BOOK_NUM	BOOK_TITLE	BOOK_SUBJECT	AU_LNAME	Num Books by Author
1	5246	Capture the Cloud	Cloud	Bruer	2
2	5244	Cloud-based Mobile Applications	Cloud	Chiang	3
3	5244	Cloud-based Mobile Applications	Cloud	Tankersly	1
4	5236	Database in the Cloud	Cloud	Walsh	2
5	5249	Starlight Applications	Cloud	Chiang	3

109.

SELECT

MIN( AVG.AVG COST ) AS 'Lowest Avg Cost',

MAX( AVG.AVG COST ) AS 'Highest Avg Cost'

FROM

( SELECT book\_subject, round( avg( book\_cost ), 2 ) AS avgcost FROM BOOK GROUP BY  
book\_subject ) AS AVG;

SQLQuery1.sql - US...USER-PC\User (56)\* X

```
1 SELECT
2     MIN( AVG.AVG COST ) AS 'Lowest Avg Cost',
3     MAX( AVG.AVG COST ) AS 'Highest Avg Cost'
4 FROM
5     ( SELECT book_subject, round( avg( book_cost ), 2 ) AS avgcost FROM BOOK GROUP BY book_subject ) AS AVG;
```

91 %

Results Messages

	Lowest Avg Cost	Highest Avg Cost
1	66.620000	89.950000