

Brianna Griffin

MAT 265 - Assignment Week 3

Resources: [library search](#) [item types](#) [meta data: SPL](#)

---

### Part 1:

For the first query, I am interested in finding data points about books about wars. I will go more in depth as I write the queries, but to begin I collected itemtype, title, check out date, and subject on books that had the key word “war” in their subject description. This was just to get a small gist of the data.

★ Note: I filtered the cout column to be YEAR(cout) >= 2006 because I noticed some books with a checkout date of 1970 and wanted to disregard any data anomalous in my query.

### Query:

```
SELECT
  B.itemtype, B.title, B.cout, A.subject AS book_subject
FROM
  spl_2016.subject AS A
  INNER JOIN
  spl_2016.inraw AS B ON A.bibNumber = B.bibNumber
WHERE
  A.subject LIKE '%war %'
  AND B.itemtype LIKE '%bk'
  AND YEAR(B.cout) >= 2006
```

LIMIT 500;

[Output](#) : (1)

Furthermore, I added some more lines to the query in order to find out more about this selection of data. I did so by now looking at the month that some of these books were checked out during the year 2022. I wanted to see at which war books people are interested in the year 2022.

### Query:

```
SELECT
  CONCAT(MONTH(cout), '-', YEAR(cout)) AS date_CO, title, book_subject, itemtype
FROM
  (SELECT
    B.itemtype, B.title, B.cout, A.subject AS book_subject
  FROM
    spl_2016.subject AS A
    INNER JOIN spl_2016.inraw AS B ON A.bibNumber = B.bibNumber
  WHERE
    A.subject LIKE '%war %'
```

```

AND B.itemtype LIKE '%bk'
AND YEAR(B.COUT) = 2022) sub
GROUP BY title , date_CO, book_subject, itemtype
ORDER BY date_CO ASC
LIMIT 500;
output \(2\)

```

Next, I aggregated the war books by year. Hence, we are now able to see how many times a certain war book was checked out each year since 2006.

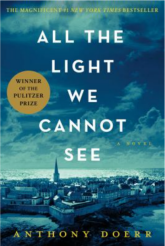
Query:

```

SELECT
  DISTINCT(bibNumber), title, YEAR(cout) as YEARCO, COUNT(cout) AS NumOfCO,
  book_subject, itemtype
FROM
  (SELECT
    B.itemtype, A.bibNumber, B.title, B.cout, A.subject AS book_subject
  FROM
    spl_2016.subject AS A
  INNER JOIN spl_2016.inraw AS B ON A.bibNumber = B.bibNumber
  WHERE
    A.subject LIKE '%war %'
    AND B.itemtype LIKE '%bk'
    AND YEAR(B.COUT) >= 2006) sub
GROUP BY title , bibNumber, YEARCO, book_subject, itemtype
ORDER BY NumOfCo DESC
LIMIT 500;
output \(3\)

```

← Search results



**All the Light We Cannot See**  
a Novel  
Doerr, Anthony, 1973-  
★★★★★ (5,800 ratings) ☆☆☆☆ Rate this

Book, 2014   Downloadable Audiobook   eBook +2

Available   All copies in use

"Marie Laure lives with her father in Paris within walking distance of the Museum of Natural History where he works as the master of the locks (there are thousands of locks in the museum). When she is six, she goes blind, and her father builds her a model of their neighborhood, every house, every manhole, so she can memorize it with her fingers and navigate the real streets with her feet and cane. When the Germans occupy Paris, father and daughter flee to Saint-Malo on the Brittany coast, where Marie-Laure's agoraphobic great uncle lives in a tall, narrow house by the sea wall. In another world in Germany, an orphan boy, Werner, grows up with his younger sister, Jutta, both enchanted by a crude radio Werner finds. He becomes a master at building and fixing radios, a talent that wins him a place

[Read an excerpt](#)

**All copies in use**  
30 copies 136 on hold

Availability by location >

[Place hold](#)

For Later

(fiction book during WW2)

I found this all interesting, and was able to practice with some more complex query writing skills like sub queries and inner-joins. However, I want to narrow down the data that I am gathering as

the prior queries were very broad. Some of the outgoing data were comic books about war while some were nonfiction, historical books. It was very vast. Hence, I will look into books about the civil war.

Query:

```
SELECT
  title, YEAR(cout) as YEARCO, COUNT(cout) AS NumOfCO, book_subject, itemtype
FROM
  (SELECT
    B.itemtype, B.title, B.cout, A.subject AS book_subject
  FROM
    spl_2016.subject AS A
  INNER JOIN spl_2016.inraw AS B ON A.bibNumber = B.bibNumber
  WHERE
    A.subject LIKE '%war %'
      AND A.subject LIKE '%civil%'
    AND B.itemtype LIKE '%bk'
    AND YEAR(B.COUT) >= 2006) sub
GROUP BY title , YEARCO, book_subject, itemtype
ORDER BY NumOfCO DESC;
output\(4\)(2642 Rows) < Max
```

Following, I refined the search to look at books dealing with wars and America.

Query:

```
SELECT
  title, YEAR(cout) as YEARCO, COUNT(cout) AS NumOfCO, book_subject, itemtype
FROM
  (SELECT
    B.itemtype, B.title, B.cout, A.subject AS book_subject
  FROM
    spl_2016.subject AS A
  INNER JOIN spl_2016.inraw AS B ON A.bibNumber = B.bibNumber
  WHERE
    A.subject LIKE '%war %'
      AND A.subject LIKE '%america%'
    AND B.itemtype LIKE '%bk'
    AND YEAR(B.COUT) >= 2006) sub
GROUP BY title , YEARCO, book_subject, itemtype
ORDER BY NumofCO DESC;
output \(5\)(15401 Rows)
```

Overall, this sequence of queries was a good exercise for me in learning more about the data set and wrangling the data to find the columns that I want.

---

## Part 2:

Moving on to a secondary part, I would like to explore the different item types being checked out at the SPL. I am interested in who is going to the library for musical/listening purposes. In order to explore this, I began by looking at the different item types. I wrote down which were for CDs, song books and records keeping note of this for my following queries.

The first query that I wrote is very broad and simple: pulling up itemNumber, itemType, title, checkout date, and bib number for CDs, song books, and records at the SPL.

### Query:

```
SELECT
  A.itemNumber, A.itemType, C.title, C.cout, B.bibnumber
FROM
  spl_2016.itemType AS A
  INNER JOIN
  spl_2016.itemToBib AS B ON A.itemNumber = B.itemNumber
  INNER JOIN
  spl_2016.inraw AS C ON B.bibNumber = C.bibNumber
WHERE
  A.itemtype = 'accd' OR A.itemtype = 'arcd' OR A.itemtype = 'jccd' OR A.itemtype = 'jrzd' --
  CDS
  OR A.itemtype = 'jcmus' OR A.itemtype = 'jrmus' -- song books
  OR A.itemtype = 'acrec' OR A.itemtype = 'arrec' OR A.itemtype = 'jrec' -- records
  AND YEAR(C.cout) >= 2006;
```

### Output (6)

Next, I wanted to look at CD's individually grouped by the titles. We are able to see which CD was popular with the data aggregated by year.

### Query:

```
SELECT title, YEAR(cout) as year_CO, COUNT(cout) AS num_CO, itemType, bibNumber
-- add bibNumber because some had the same title
-- example 'Greatest Hits' for multiple artists
FROM spl_2016.inraw
WHERE
  itemtype = 'accd' OR itemtype = 'arcd' OR itemtype = 'jccd' OR itemtype = 'jrzd' -- CDS
  AND YEAR(cout) >= 2006
```

and subj is not null

GROUP BY title, year\_CO, itemType, bibNumber

ORDER BY num\_CO DESC;

### [Output\(7\)](#)

The screenshot shows three search results for music CDs from the Seattle Public Library. Each result includes a cover image, the artist/group name, a star rating, and options to view availability and place a hold. The results are: 1. Fleet Foxes (Musical group), Music CD, 2008, 3 copies available. 2. The Fame Monster (Lady Gaga), Music CD, 2009, 2 copies available. 3. Contra (Vampire Weekend (Musical group)), Music CD, 2010, 4 copies available.

Then, I ran a query to see the top 100 CD's checked out since 2006. In other words, it shows the CD's with the most checkouts since 2006. (The same as last w/o year)

### Query:

```
SELECT title, COUNT(cout) AS num_CO, itemType, bibNumber
```

```
FROM spl_2016.inraw
```

```
WHERE
```

```
    itemType = 'accd' OR itemType = 'arcd' OR itemType = 'jccd' OR itemType = 'jrzd' -- CDS
```

```
    AND YEAR(cout) >= 2006
```

```
GROUP BY title, itemType, bibNumber
```

```
ORDER BY num_CO DESC
```

```
LIMIT 100;
```

### [Output\(8\)](#)

I did the same for [song books](#). (Top 100 song books with most checkouts since 2006)

### Query:

```
SELECT title, COUNT(cout) AS num_CO, itemType, bibNumber
```

```
FROM spl_2016.inraw
```

```
WHERE
  itemtype = 'jcmus' OR itemtype = 'jrmus' -- song books
  AND YEAR(cout) >= 2006
GROUP BY title, itemType, bibNumber
ORDER BY num_CO DESC
LIMIT 100;
Output\(9\)
```

Also, for [records](#). (Top 100 records with the most checkouts since 2006)

```
Query:
SELECT title, COUNT(cout) AS num_CO, itemType, bibNumber
FROM spl_2016.inraw
WHERE
  itemtype = 'acrec' OR itemtype = 'arrec' OR itemtype = 'jcrec' -- records
  AND YEAR(cout) >= 2006
  and subj is not null
GROUP BY title, itemType, bibNumber
ORDER BY num_CO DESC
LIMIT 100;
Output \(10\)
```

---

---

### [Part 3:](#)

Next, I wanted to explore autobiographies/biographies/memoirs specifically for those involved with sports. I will begin by querying data that have 'biography' in their subject description as a starting point.

```
Query:
SELECT * FROM
spl_2016.subject
WHERE subject LIKE '%biography%'
LIMIT 500;
OUTPUT \(11\)
```

Next, I will output the Dewey class as well. By the Dewey classification biographies are in the 920s range, but after looking at the output we see that this is not the case for all of the data points. Therefore, utilizing dewey class is not a good way to wrangle the data that we want.

```
Query:
SELECT B.title, A.subject, B.deweyClass, B.bibNumber
```

```

FROM spl_2016.subject AS A
INNER JOIN
spl_2016.inraw AS B
ON A.bibNumber = B.bibNumber
WHERE A.subject LIKE '%biography%' OR A.subject LIKE '%autobiography%' OR A.subject
LIKE '%memoior%'
AND B.itemtype LIKE '%ck'
LIMIT 500;
OUTPUT \(12\)

```

Therefore, I will not call the dewey class anymore since it is irrelevant to what I am trying to search for within the data. I will replace it with itemType. (Most are books)

Query:

```

SELECT B.title, A.subject, B.itemType, B.bibNumber
FROM spl_2016.subject AS A
INNER JOIN
spl_2016.inraw AS B
ON A.bibNumber = B.bibNumber
WHERE A.subject LIKE '%biography%' OR A.subject LIKE '%autobiography%' OR A.subject
LIKE '%memoior%'
LIMIT 500;
output\(13\)

```

Now, I will look at biographies about tennis. Specifically looking for items that are books as well.

Query:

```

SELECT DISTINCT
  (bibNumber), title, subject, itemType
FROM
  (SELECT
    B.title, A.subject, B.itemType, B.bibNumber
  FROM
    spl_2016.subject AS A
  INNER JOIN spl_2016.inraw AS B ON A.bibNumber = B.bibNumber
  WHERE
    A.subject LIKE '%biography%'
    OR A.subject LIKE '%autobiography%'
    OR A.subject LIKE '%memoior%') sub
WHERE
  subject LIKE '%tennis%'

```

AND itemType LIKE '%bk';

#### [Output \(14\)](#)

We can do the same for other sports like softball: (must less books here!)

#### Query:

```
SELECT DISTINCT
  (bibNumber), title, subject, itemType
FROM
  (SELECT
    B.title, A.subject, B.itemType, B.bibNumber
  FROM
    spl_2016.subject AS A
  INNER JOIN spl_2016.inraw AS B ON A.bibNumber = B.bibNumber
  WHERE
    A.subject LIKE '%biography%'
    OR A.subject LIKE '%autobiography%'
    OR A.subject LIKE '%memoior%') sub
WHERE
  subject LIKE '%softball%'
  AND itemType LIKE '%bk';
```

#### [output\(15\)](#)

Now, I made a table that shows the amount of biographies/autobiographies/memoirs per sport.

#### Query:

```
SELECT *
FROM(
SELECT COUNT(DISTINCT(bibNumber)) as num_of_books,
CASE
  WHEN sub LIKE '%tennis%' AND itemtype LIKE '%bk' THEN 'tennis'
  WHEN sub LIKE '%softball%' AND itemtype LIKE '%bk' THEN 'softball'
  WHEN sub LIKE '%football%' AND itemtype LIKE '%bk' THEN 'football'
  WHEN sub LIKE '%baseball%' AND itemtype LIKE '%bk' THEN 'baseball'
  WHEN sub LIKE '%volleyball%' AND itemtype LIKE '%bk' THEN 'volleyball'
  WHEN sub LIKE '%basketball%' AND itemtype LIKE '%bk' THEN 'basketball'
  WHEN sub LIKE '%golf%' AND itemtype LIKE '%bk' THEN 'golf'
  WHEN sub LIKE '%rugy%' AND itemtype LIKE '%bk' THEN 'rugy'
  WHEN sub LIKE '%surf%' AND itemtype LIKE '%bk' THEN 'surf'
  WHEN sub LIKE '%dance%' AND itemtype LIKE '%bk' THEN 'dance'
  ELSE 'NA'
END AS sport_cat
```



```

FROM
(SELECT B.bibNumber, A.subject AS sub, B.itemtype
FROM spl_2016.subject as A
INNER JOIN spl_2016.inraw AS B
ON A.bibNumber = B.bibNumber
WHERE A.subject LIKE '%biography%' OR A.subject LIKE '%autobiography%' OR A.subject
LIKE '%memoior%') sub
GROUP BY 2)sub2
WHERE sport_cat != 'NA'
order by num_of_books desc;

```

[Output \(16\)](#)

Lastly, you can see all of the book titles, subjects, and itemtypes grouped by their sport categorization in a large table of biographies, autobiographies, and memoirs.

Query:

```

SELECT *
FROM
(SELECT DISTINCT(A.BibNumber),
CASE
    WHEN subject LIKE '%tennis%' AND B.itemtype LIKE '%bk' THEN 'tennis'
    WHEN subject LIKE '%softball%' AND B.itemtype LIKE '%bk' THEN 'softball'
    WHEN subject LIKE '%football%' AND B.itemtype LIKE '%bk' THEN 'football'
    WHEN subject LIKE '%baseball%' AND B.itemtype LIKE '%bk' THEN 'baseball'
    WHEN subject LIKE '%volleyball%' AND B.itemtype LIKE '%bk' THEN 'volleyball'
    WHEN subject LIKE '%basketball%' AND B.itemtype LIKE '%bk' THEN 'basketball'
    WHEN subject LIKE '%golf%' AND B.itemtype LIKE '%bk' THEN 'golf'
    WHEN subject LIKE '%rugy%' AND B.itemtype LIKE '%bk' THEN 'rugy'
    WHEN subject LIKE '%surf%' AND B.itemtype LIKE '%bk' THEN 'surf'
    WHEN subject LIKE '%dance%' AND B.itemtype LIKE '%bk' THEN 'dance'
    ELSE 'NA'
END AS sport_cat,
B.title,
A.subject AS Sub, B.itemType
FROM spl_2016.subject AS A
INNER JOIN
spl_2016.inraw AS B
ON A.bibNumber = B.bibNumber
WHERE A.subject LIKE '%biography%' OR A.subject LIKE '%autobiography%' OR A.subject
LIKE '%memoior%') sub
WHERE sport_cat != 'NA'

```

GROUP BY 1, 2, 3, 4, 5

ORDER BY 2

[output \(17\)](#)

---