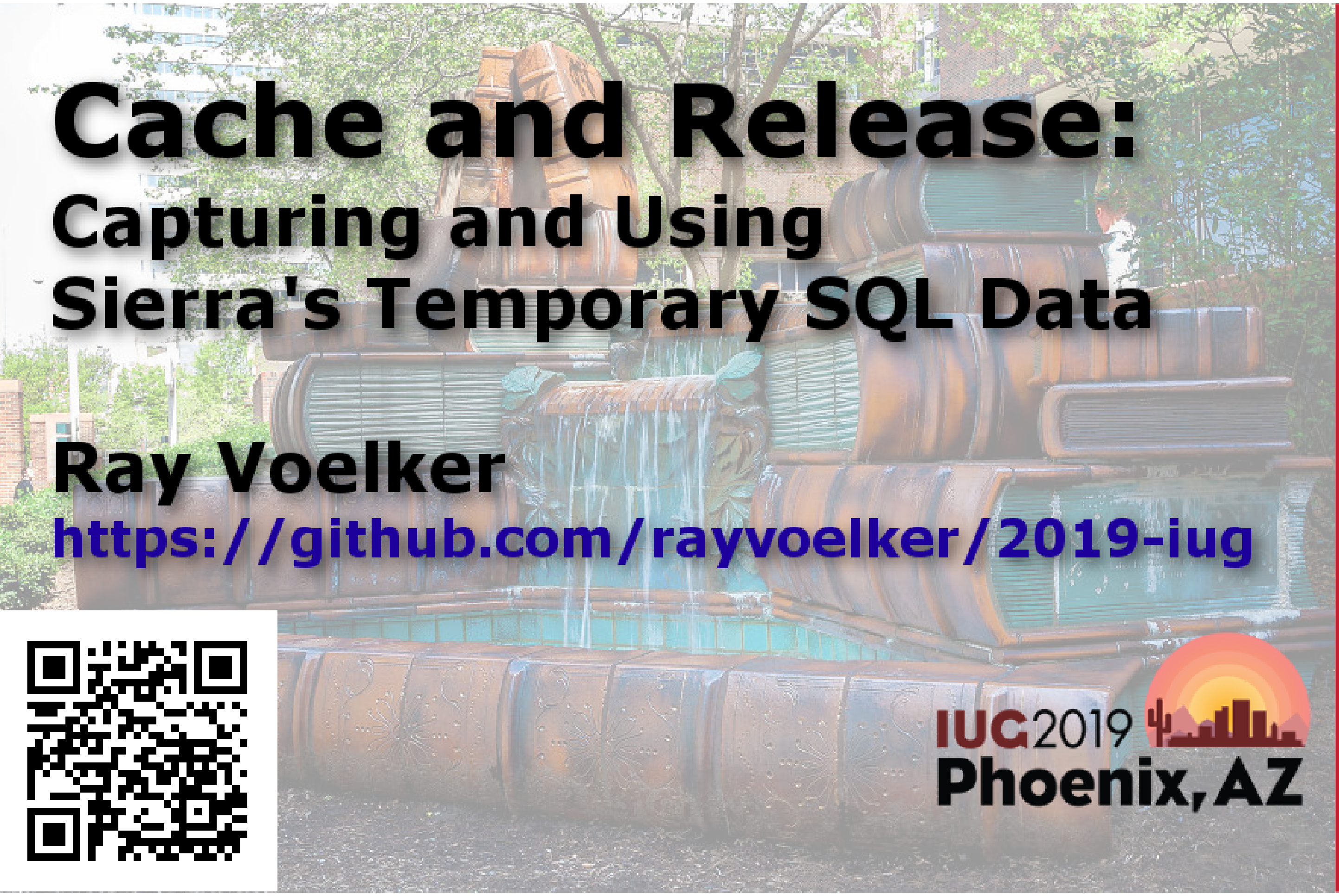
Cache and Release:
Capturing and Using Sierra's Temporary SQL Data

Ray Voelker
https://github.com/rayvoelker/2019-iug

IUG2019 Phoenix, AZ

THE PUBLIC LIBRARY of Cincinnati and Hamilton County



QUICK BACKGROUND

- Sierra’s Direct SQL Access feature allows us to:
  - Quickly and efficiently target and extract real-time data from the Sierra ILS.
  - Organize data in logical and useful ways

QUICK BACKGROUND (CONT.)

- Why save / preserve data from Sierra?
  1. Data transformations and integrations for specific use cases
     wikipedia.org/wiki/Data_transformation
     wikipedia.org/wiki/Data_integration
     - For example, adding geocoding information to patron address data:
       wikipedia.org/wiki/Geocoding

QUICK BACKGROUND (CONT.)

- Why save / preserve data from Sierra? (cont.)

  2. Cache Sierra data:
     - For use in an application instead of running an "expensive query" to deliver content
     - For use in an application where holding onto data which may otherwise be destroyed or transformed by the Sierra application itself

UNDERSTANDING SIERRA'S DATABASE VIEWS AND DATA

- Some data in Sierra stays more static (think of a "receipt", or log of transactions)

circ_trans

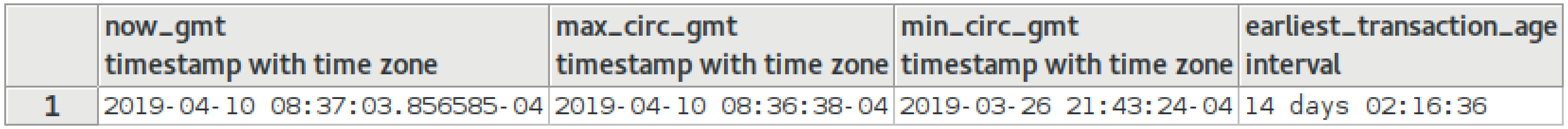
Each row of circ_trans contains information about a circulation transaction.

Column	Data Type	Not NULL?	Comment
id	int	false	System-generated sequential ID.
transaction_gmt	timestamptz	false	Transaction date in UNIX format.
application_name	varchar	false	The name of the program that generated the transaction. Valid program names are: • circ (includes transactions made using PC Circ) • circa (for transactions written by selfcheckwebserver and in-house use [transaction codes 'u' and 's'], which use webpac to execute transactions.) • milcirc • milmyselfcheck • readreq • selfcheck
source_code	varchar	false	The transaction source. Possible values are: • local • INN-Reach • ILL
op_code	varchar	false	Type of transaction: o = i = checkin

UNDERSTANDING SIERRA’S DATABASE VIEWS AND DATA (CONT.)

- Circulation transactions are created in the database table and remain static
- Rows are deleted from the table after a certain period of time (2 weeks is the default, but this can be extended by iii upon request)

```sql
-- get some info about our circ_trans dates
SELECT
NOW()::TIMESTAMP WITH TIME ZONE as now_gmt,
MAX(c.transaction_gmt)::TIMESTAMP WITH TIME ZONE AS max,
MIN(c.transaction_gmt)::TIMESTAMP WITH TIME ZONE AS min,
AGE(MIN(c.transaction_gmt)) AS earliest_transaction_age

FROM
sierra_view.circ_trans as c
```



	now_gmt timestamp with time zone	max_circ_gmt timestamp with time zone	min_circ_gmt timestamp with time zone	earliest_transaction_age interval
1	2019-04-10 08:37:03.856585-04	2019-04-10 08:36:38-04	2019-03-26 21:43:24-04	14 days 02:16:36

UNDERSTANDING SIERRA’S DATABASE VIEWS AND DATA (CONT.)

- Other data is more variable or is a direct representation that describes a particular state of a record or process in the ILS.

hold

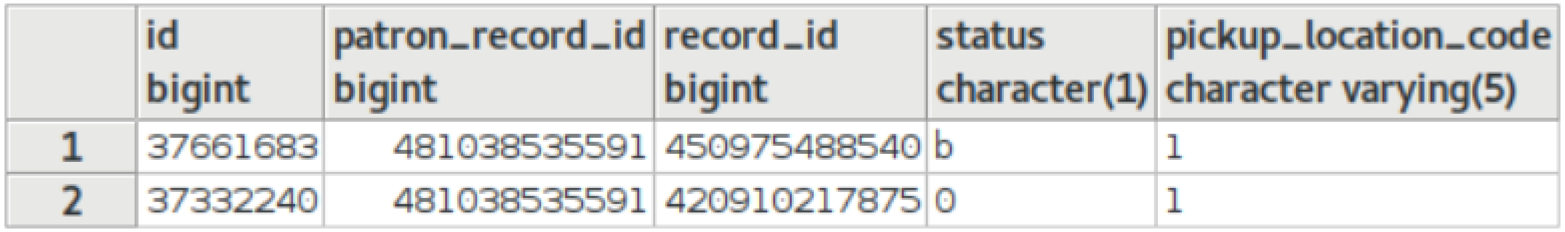
Each row of hold describes a bibliographic, item, or volume hold.

Column	Data Type	Not NULL?	Comment
id	bigint	false	System-generated sequential ID.
patron_record_id	bigint	false	Foreign key to patron_record.
record_id	bigint	false	Foreign key to record.
placed_gmt	timestamp	false	Date the hold was placed.
is_frozen	boolean	false	Specifies whether the hold is frozen (suspended).
delay_days	int	false	Stores the "not wanted before" date as a number of days after the date the hold was placed. The maximum value is "180". If a "not wanted before" date was not specified, the value is '0'.
location_code	varchar	false	For bib or volume-level holds, the branch location from which to fill the hold, if the hold is set for 'Limit to Location'. Does not apply to item-level holds (blank).
expires_gmt	timestamp	false	"Not needed after" date.
status	char	false	Hold status.

UNDERSTANDING SIERRA'S DATABASE VIEWS AND DATA (CONT.)

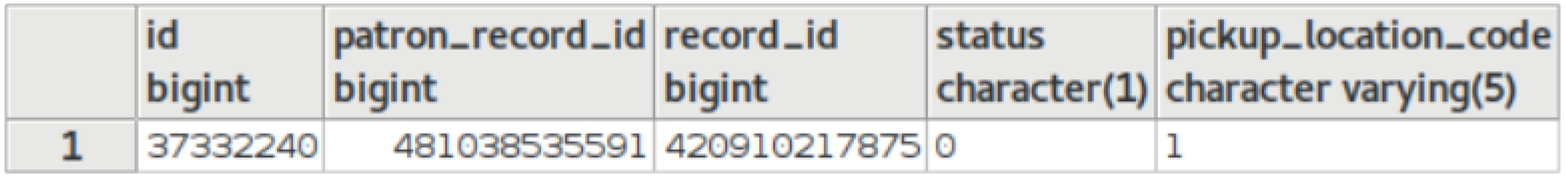
- The state of the hold is defined in the Sierra database
- Data changes depending on the state or status of the hold, and is then removed from the database when the hold is deleted, filled or expires

```sql
-- this will select Ray Voelker's hold information from the
-- Sierra SQL database
SELECT
h.id,
h.patron_record_id,
h.record_id,
h.status,
h.pickup_location_code

FROM
sierra_view.hold as h

WHERE
h.patron_record_id = 481038535591;
```



	id bigint	patron_record_id bigint	record_id bigint	status character(1)	pickup_location_code character varying(5)
1	37661683	481038535591	450975488540	b	1
2	37332240	481038535591	420910217875	0	1

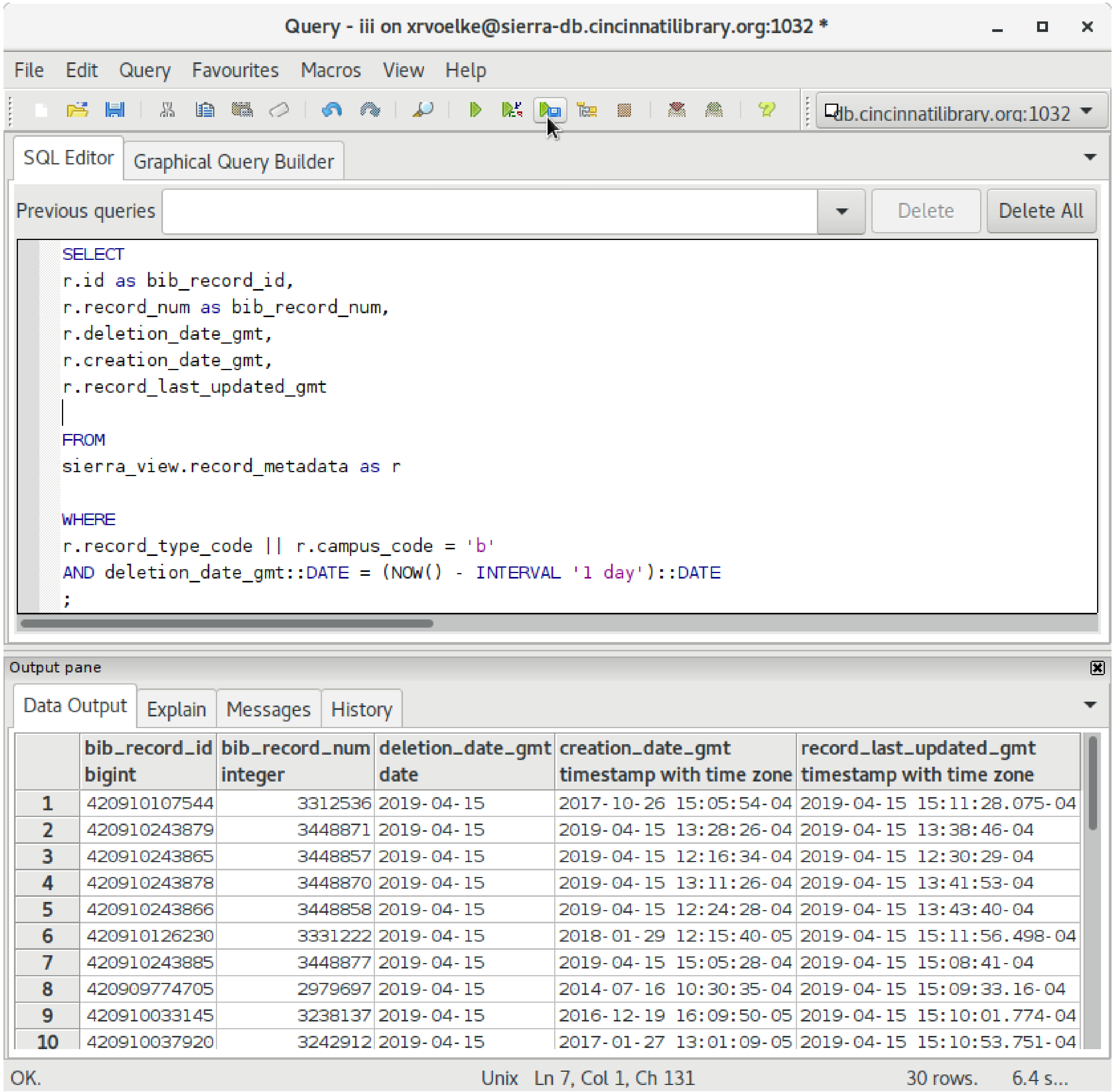
I checkout the item, and the hold data goes away after a second query.



	id bigint	patron_record_id bigint	record_id bigint	status character(1)	pickup_location_code character varying(5)
1	37332240	481038535591	420910217875	0	1

HOW TO CACHE / TRANSFORM / PRESERVE DATA FROM SIERRA?

- No shortage of options!
  - pgAdmin is a popular choice for a desktop client
    www.pgadmin.org
  - "Execute query, write result to file"
    Creates a .csv file from the results



```
SELECT
r.id as bib_record_id,
r.record_num as bib_record_num,
r.deletion_date_gmt,
r.creation_date_gmt,
r.record_last_updated_gmt

FROM
sierra_view.record_metadata as r

WHERE
r.record_type_code || r.campus_code = 'b'
AND deletion_date_gmt::DATE = (NOW() - INTERVAL '1 day')::DATE
;
```

	bib_record_id bigint	bib_record_num integer	deletion_date_gmt date	creation_date_gmt timestamp with time zone	record_last_updated_gmt timestamp with time zone
1	420910107544	3312536	2019-04-15	2017-10-26 15:05:54-04	2019-04-15 15:11:28.075-04
2	420910243879	3448871	2019-04-15	2019-04-15 13:28:26-04	2019-04-15 13:38:46-04
3	420910243865	3448857	2019-04-15	2019-04-15 12:16:34-04	2019-04-15 12:30:29-04
4	420910243878	3448870	2019-04-15	2019-04-15 13:11:26-04	2019-04-15 13:41:53-04
5	420910243866	3448858	2019-04-15	2019-04-15 12:24:28-04	2019-04-15 13:43:40-04
6	420910126230	3331222	2019-04-15	2018-01-29 12:15:40-05	2019-04-15 15:11:56.498-04
7	420910243885	3448877	2019-04-15	2019-04-15 15:05:28-04	2019-04-15 15:08:41-04
8	420909774705	2979697	2019-04-15	2014-07-16 10:30:35-04	2019-04-15 15:09:33.16-04
9	420910033145	3238137	2019-04-15	2016-12-19 16:09:50-05	2019-04-15 15:10:01.774-04
10	420910037920	3242912	2019-04-15	2017-01-27 13:01:09-05	2019-04-15 15:10:53.751-04

OK. Unix Ln 7, Col 1, Ch 131 30 rows. 6.4 s...

HOW TO CACHE / TRANSFORM / PRESERVE DATA FROM SIERRA? (CONT.)

- Many programming languages provide access to PostgreSQL via their libraries:
  - php-pgsql: PHP PostgreSQL driver www.php.net/manual/en/book.pgsql.php
  - psycopg2: Python PostgreSQL adapter initd.org/psycopg/docs/

HOW TO CACHE / TRANSFORM / PRESERVE DATA FROM SIERRA? (CONT.)

- My method consists of the following overview:
  1. Use Python to connect to Sierra’s database
  2. Issue SQL statement on Sierra's database to target and compile the data for extraction

HOW TO CACHE / TRANSFORM / PRESERVE DATA FROM SIERRA? (CONT.)

- My method consists of the following overview (cont.):

  3. Export result data to either a .csv file, and/or directly into a SQLite database
     - .csv files are easy to later load into an SQLite database, spreadsheet, or other data warehouse tool

HOW TO CACHE / TRANSFORM / PRESERVE DATA FROM SIERRA? (CONT.)

SQLite Database: sqlite.org

"SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine."

HOW TO CACHE / TRANSFORM / PRESERVE DATA FROM SIERRA? (CONT.)

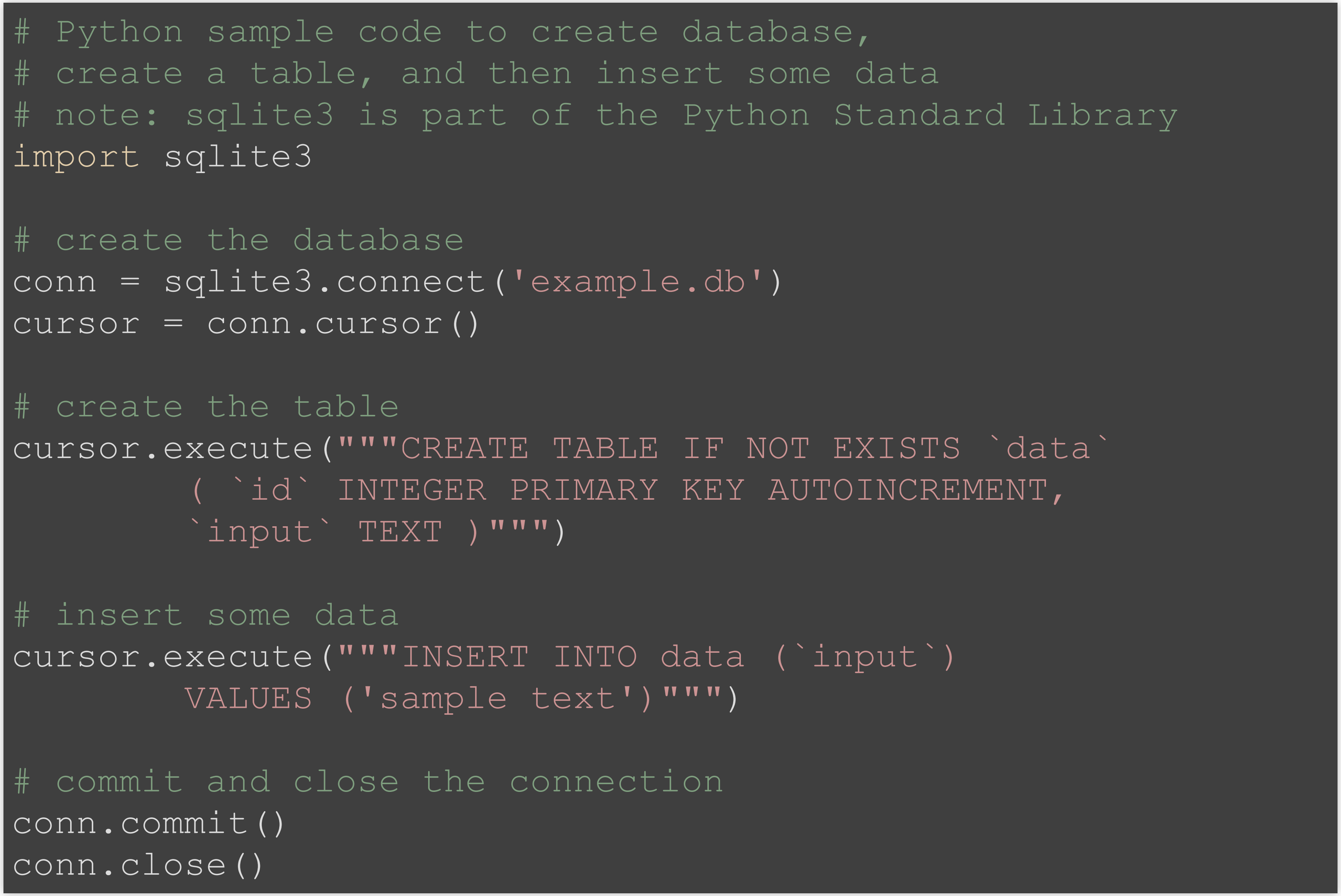
SQLite Database (cont.)

- SQLite database engine is a great tool for caching data:
  1. Stores and organizes large amounts of data quickly and efficiently
  2. You don’t have to set up and maintain a server (data is portable; entire database is contained in a single, cross-platform file)

HOW TO CACHE / TRANSFORM / PRESERVE DATA FROM SIERRA? (CONT.)

SQLite Database (cont.)

3. Ability and flexibility to build SQL queries and applications that directly use the data that you've collected
4. Has a useful desktop tool: sqlitebrowser.org
5. It's included in the Python Standard Library!



```python
# Python sample code to create database,
# create a table, and then insert some data
# note: sqlite3 is part of the Python Standard Library
import sqlite3

# create the database
conn = sqlite3.connect('example.db')
cursor = conn.cursor()

# create the table
cursor.execute("""CREATE TABLE IF NOT EXISTS `data`
        ( `id` INTEGER PRIMARY KEY AUTOINCREMENT,
        `input` TEXT )""")

# insert some data
cursor.execute("""INSERT INTO data (`input`)
        VALUES ('sample text')""")

# commit and close the connection
conn.commit()
conn.close()
```

EXAMPLES:

1. **Collection Analysis / Circulation Data Analysis**
    - intended for use with the CollectionHQ service, but can be adapted for local use
2. **Patron Savings Calculator**
    - intended to display information on the Encore patron account information page about how much money has been "saved" by using the library

EXAMPLES: (CONT.)

3. **Hold Shelf Delivery Matrix Report**
   - intended to produce a spreadsheet report of items for hold shelf locations (ready for patron pickup) and from where they originated
4. **Mapping Geo Data from Patron Address Data and Circulation Transaction Data**
   - intended to take geocoded patron address data and produce a visualization by plotting it on a map

EXAMPLE 1

COLLECTION ANALYSIS / CIRCULATION DATA ANALYSIS

- github.com/plch/collection-analysis
- This application was built for the purpose of automatically preserving and sending data from the Sierra database to the CollectionHQ service via FTP

COLLECTION ANALYSIS / CIRCULATION DATA ANALYSIS (CONT.)

- Three main groups of data are targeted for export:
  1. Bibliographic Record Data:
     temp_table-bib_data.sql
     Fields exported include: bib record number, control numbers (var fields tagged 'o'), isbn, call numbers, and a few other metadata fields related to the bib record

COLLECTION ANALYSIS / CIRCULATION DATA ANALYSIS (CONT.)

- Three main groups of data are targeted for export: (cont.)
  2. Item Record Data:
     temp_table-item_data.sql
     Fields exported include: item record number, bib record number, circulation information, price, and a few other metadata fields related to the item record

COLLECTION ANALYSIS / CIRCULATION DATA ANALYSIS (CONT.)

- Three main groups of data are targeted for export: (cont.)

  3. Hold Data:

     temp_table-hold_data.sql

     Weekly snapshot of bib-level holds. It is organized by title (bib record number), and then each title has a list of holds with related metadata for each hold (patron number, pickup location, created date, expiration date, etc)

COLLECTION ANALYSIS / CIRCULATION DATA ANALYSIS (CONT.)

- Overview of the process:
  1. Create and open .csv files to hold data output and/or open local database (if it’s useful to place data into local SQLite database)
  2. Connect to remote Sierra database, and create the temporary tables that will be used for exporting

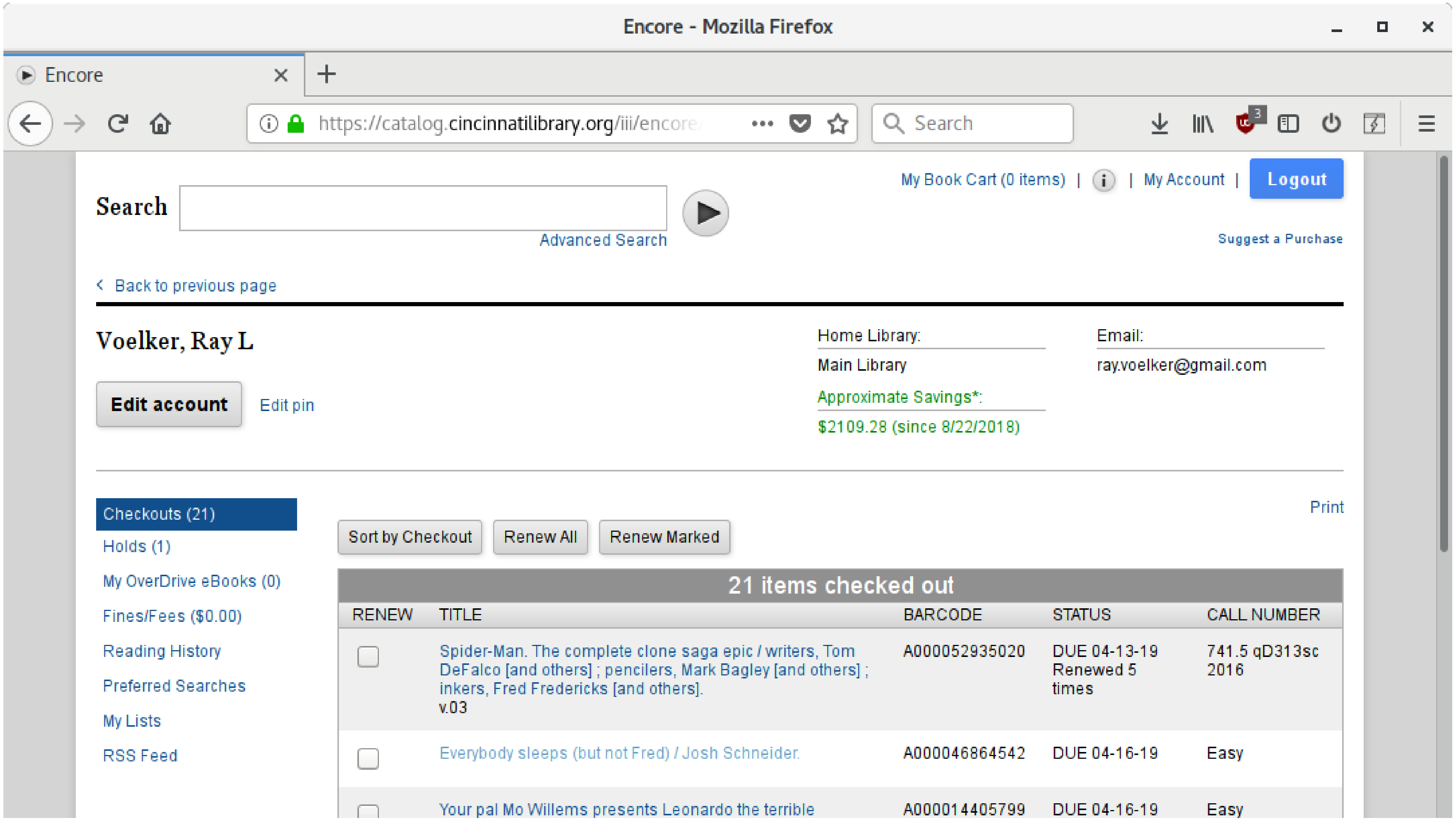
COLLECTION ANALYSIS / CIRCULATION DATA ANALYSIS (CONT.)

- Overview of the process (cont.):
  3. Generate data output from the Sierra database temporary tables, and write the output to a .csv file and/or to a local SQLite database
  4. Send data via FTP
- full export script can be found in the public github repository for this project

EXAMPLE 2

PATRON SAVINGS CALCULATOR

- github.com/plch/patron-savings-calculator
- This application was designed to work with the iii discovery layer Encore's account detail page, to display how much a patron has "saved" by borrowing from the library



Encore - Mozilla Firefox

Encore

https://catalog.cincinnatilibrary.org/iii/encore/

My Book Cart (0 items) | | My Account | Logout

Search

Advanced Search

Suggest a Purchase

< Back to previous page

Voelker, Ray L

Edit account   Edit pin

Home Library:	Email:
Main Library	ray.voelker@gmail.com
Approximate Savings*:	
$2109.28 (since 8/22/2018)	

Checkouts (21)
Holds (1)
My OverDrive eBooks (0)
Fines/Fees ($0.00)
Reading History
Preferred Searches
My Lists
RSS Feed

Print

Sort by Checkout   Renew All   Renew Marked

21 items checked out

RENEW	TITLE	BARCODE	STATUS	CALL NUMBER
☐	Spider-Man. The complete clone saga epic / writers, Tom DeFalco [and others] ; pencilers, Mark Bagley [and others] ; inkers, Fred Fredericks [and others]. v.03	A000052935020	DUE 04-13-19 Renewed 5 times	741.5 qD313sc 2016
☐	Everybody sleeps (but not Fred) / Josh Schneider.	A000046864542	DUE 04-16-19	Easy
☐	Your pal Mo Willems presents Leonardo the terrible	A000014405799	DUE 04-16-19	Easy

PATRON SAVINGS CALCULATOR (CONT.)

- Cached "savings" information is based on the patron record number and can be output in multiple formats (based on this application's custom URL endpoints):
    - JSON

      /api/v1/patron_savings/2198439

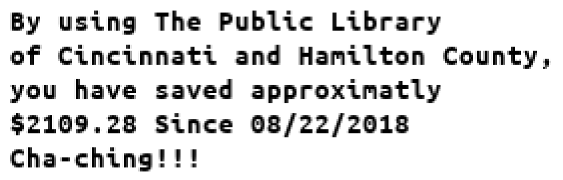
```
{
        "count_titles": 118,
        "min_date_epoch": 1534957500,
        "patron_record_num": 2198439,
        "total_savings": 2109.2799999999997
}
```

PATRON SAVINGS CALCULATOR (CONT.)

- Cached “savings” information is based on the patron record number and can be output in multiple formats (based on this application's custom URL endpoints):
  - PNG /api/v1/patron_savings/img/2198439

PATRON SAVINGS CALCULATOR (CONT.)

- Cached “savings” information is based on the patron record number and can be output in multiple formats (based on this application's custom URL endpoints):
  - PNG /api/v1/patron_savings/img/2198439



```
By using The Public Library
of Cincinnati and Hamilton County,
you have saved approximatly
$2109.28 Since 08/22/2018
Cha-ching!!!
```

PATRON SAVINGS CALCULATOR (CONT.)

- About this very simple RESTful API
  - Written as a Python / Flask application:
    - github.com/plch/patron-savings-calculator/blob/master/app.py
    - www.palletsprojects.com/p/flask

PATRON SAVINGS CALCULATOR (CONT.)

- About this very simple RESTful API (cont)
  - Hosted on an Apache Web Server via the WSGI module
  - A good tutorial on how this can be set up can be found here: www.digitalocean.com/community/tutorials/how-to-deploy-a-flask-application-on-an-ubuntu-vps

PATRON SAVINGS CALCULATOR (CONT.)

- About this very simple RESTful API (cont)
  - Draws data from the SQLite database that has been caching relevant data
  - SQLite database, `patron_savings.db` is updated frequently (every 5 minutes) via a Python update script: github.com/plch/patron-savings-calculator/blob/master/update.py

PATRON SAVINGS CALCULATOR (CONT.)

About the `update.py` script:

- Script starts by looking at the last entry it received from the Sierra database
- A query is constructed to extract relevant data from the Sierra database that is more recent than that last entry in the local database
- The local database is updated with the fresh data from the Sierra database

PATRON SAVINGS CALCULATOR (CONT.)

A note about privacy / protection / obfuscation of this data:

- No title information is saved in the local database, other than a hashed bib record id
- Hashed bib record id is stored to avoid duplicating the price information when a title is checked out multiple times, and to differentiate titles from one another in the local database

PATRON SAVINGS CALCULATOR (CONT.)

A note about obfuscation with this data (cont.):

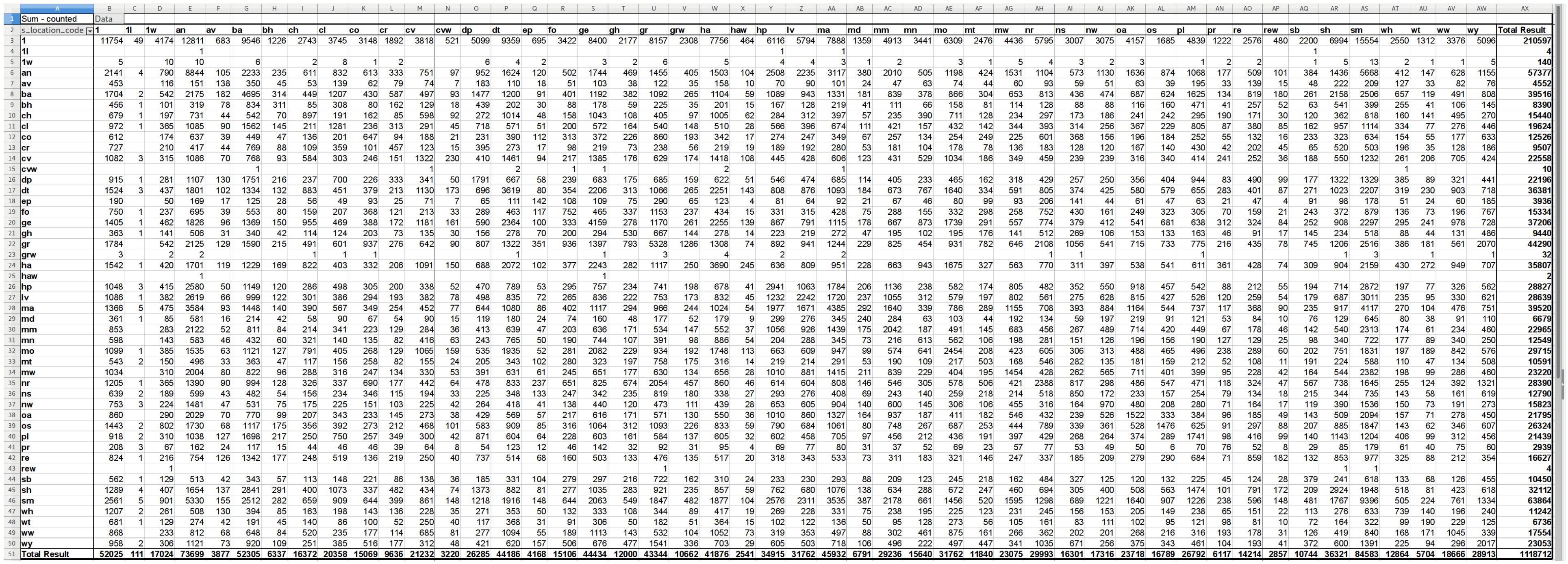
- No patron information, other than the patron record number, is stored in the local database
- Only price and number of titles checked out are surfaced via the RESTful API

EXAMPLE 3

HOLD SHELF DELIVERY MATRIX REPORT

- github.com/plch/plch-holds-shelf
- The purpose of this application is to create a spreadsheet that displays items delivered to a hold shelf location, and from what location they came

HOLD SHELF DELIVERY MATRIX REPORT (CONT.)



Sum - counted	Data																																																
s_location_code	1	1l	1w	an	av	ba	bh	ch	cl	co	cr	cv	cvw	dp	dt	ep	fo	ge	gh	gr	grw	ha	haw	hp	lv	ma	md	mm	mn	mo	mt	mw	nr	ns	nw	oa	os	pl	pr	re	rew	sb	sh	sm	wh	wt	ww	wy	Total Result
1	11754	49	4174	12811	683	9546	1226	2743	3745	3148	1892	3818	521	5099	9359	695	3422	8400	2177	8157	2308	7756	464	6116	5794	7888	1359	4913	3441	6309	2476	4436	5795	3007	3075	4157	1685	4839	1222	2576	480	2200	6994	15554	2550	1312	3376	5096	210597
1l				1																				1		1																1							4
1w	5		10	10		6		2	8	1	2			6	4	2		3	2	6		5		4	4	3	1	2		3	1	5	4	3	2	3		1	2	2		1	5	13	2	1	1	5	140
an	2141	4	790	8844	105	2233	235	611	832	613	333	751	97	952	1624	120	502	1744	469	1455	405	1503	104	2508	2235	3117	380	2010	505	1198	424	1531	1104	573	1130	1636	874	1068	177	509	101	384	1436	5668	412	147	628	1155	57377
av	453		116	151	138	350	45	53	139	62	79	74	7	183	110	18	51	103	38	122	35	158	10	70	90	101	24	47	63	74	44	60	93	59	51	63	39	195	33	139	15	48	222	209	127	33	82	76	4552
ba	1704	2	542	2175	182	4695	314	449	1207	430	587	497	93	1477	1200	91	401	1192	382	1092	265	1104	59	1089	943	1331	181	839	378	866	304	653	813	436	474	687	624	1625	134	819	180	261	2158	2506	657	119	491	808	39516
bh	456	1	101	319	78	834	311	85	308	80	162	129	18	439	202	30	88	178	59	225	35	201	15	167	128	219	41	111	66	158	81	114	128	88	88	116	160	471	41	257	52	63	541	399	255	41	106	145	8390
ch	679	1	197	731	44	542	70	897	191	162	85	598	92	272	1014	48	158	1043	108	405	97	1005	62	284	312	397	57	235	390	711	128	234	297	173	186	241	242	295	190	171	30	120	362	818	160	141	495	270	15440
cl	972	1	365	1085	90	1562	145	211	1281	236	313	291	45	718	571	51	200	572	164	540	148	510	28	566	396	674	111	421	157	432	142	344	393	314	256	367	229	805	87	380	85	162	957	1114	334	77	276	446	19624
co	612		174	637	39	449	47	136	201	647	94	188	21	231	390	112	313	372	226	860	193	342	17	274	247	349	67	257	134	254	249	225	601	368	156	196	184	252	55	132	16	233	323	634	154	55	177	633	12526
cr	727		210	417	44	769	88	109	359	101	457	123	15	395	273	17	98	219	73	238	56	219	19	189	192	280	53	181	104	178	78	136	183	128	120	167	140	430	42	202	45	65	520	503	196	35	128	186	9507
cv	1082	3	315	1086	70	768	93	584	303	246	151	1322	230	410	1461	94	217	1385	176	629	174	1418	108	445	428	606	123	431	529	1034	186	349	459	239	239	316	340	414	241	252	36	188	550	1232	261	206	705	424	22558
cvw						1						1			2		1	1				2				1																			1				10
dp	915	1	281	1107	130	1751	216	237	700	226	333	341	50	1791	667	58	239	683	175	685	159	622	51	546	474	685	114	405	233	465	162	318	429	257	250	356	404	944	83	490	99	177	1322	1329	385	89	321	441	22196
dt	1524	3	437	1801	102	1334	132	883	451	379	213	1130	173	696	3619	80	354	2206	313	1066	265	2251	143	808	876	1093	184	673	767	1640	334	591	805	374	425	580	579	655	283	401	87	271	1023	2207	319	230	903	718	36381
ep	190		50	169	17	125	28	56	49	93	25	71	7	65	111	142	108	109	75	290	65	123	4	81	64	92	21	67	46	80	99	93	206	141	44	61	47	63	21	47	4	91	98	178	51	24	60	185	3936
fo	750	1	237	695	39	553	80	159	207	368	121	213	33	289	463	117	752	465	337	1153	237	434	15	331	315	428	75	288	155	332	298	258	752	430	161	249	323	305	70	159	21	243	372	879	136	73	196	767	15334
ge	1405	1	462	1826	96	1369	150	955	469	388	172	1181	161	590	2364	100	333	4159	278	1170	261	2255	139	867	791	1115	178	667	873	1739	291	557	774	379	412	541	681	638	312	324	84	252	908	2297	295	241	978	728	37206
gh	363	1	141	506	31	340	42	114	124	203	73	135	30	156	278	70	200	294	530	667	144	278	14	223	219	272	47	195	102	195	176	141	512	269	106	153	133	163	46	91	17	145	234	518	88	44	131	486	9440
gr	1784		542	2125	129	1590	215	491	601	937	276	642	90	807	1322	351	936	1397	793	5328	1286	1308	74	892	941	1244	229	825	454	931	782	646	2108	1056	541	715	733	775	216	435	78	745	1206	2516	386	181	561	2070	44290
grw	3		2	2				1	1	1					1			1		3		4		2		2							1	1				1					1	3		1		1	32
ha	1542	1	420	1701	119	1229	169	822	403	332	206	1091	150	688	2072	102	377	2243	282	1117	250	3690	245	636	809	951	228	663	943	1675	327	563	770	311	397	538	541	611	361	428	74	309	904	2159	430	272	949	707	35807
haw				1														1																															2
hp	1048	3	415	2580	50	1149	120	286	498	305	200	338	52	470	789	53	295	757	234	741	198	678	41	2941	1063	1784	206	1136	238	582	174	805	482	352	550	918	457	542	88	212	55	194	714	2872	197	77	326	562	28827
lv	1086	1	382	2619	66	999	122	301	386	294	193	382	78	498	835	72	265	836	222	753	173	832	45	1232	2242	1720	237	1055	312	579	197	802	561	275	628	815	427	526	120	259	54	179	687	3011	235	95	330	621	28639
ma	1366	5	475	3584	93	1448	140	390	567	349	254	452	77	644	1080	86	402	1117	294	966	244	1024	54	1977	1671	4385	292	1640	339	786	289	1155	708	393	884	1164	544	737	117	368	90	235	917	4117	270	104	476	751	39520
md	361	1	85	581	16	214	42	58	90	67	54	90	15	119	180	24	74	160	48	177	52	179	9	299	276	345	240	284	63	103	44	192	134	59	197	219	91	121	53	84	10	76	129	645	80	38	91	110	6679
mm	853		283	2122	52	811	84	214	341	223	129	284	36	413	639	47	203	636	171	534	147	552	37	1056	926	1439	175	2042	187	491	145	683	456	267	489	714	420	449	67	178	46	142	540	2313	174	61	234	460	22965
mn	598		143	583	46	432	60	321	140	135	82	416	63	243	765	50	190	744	107	391	98	886	54	204	288	345	73	216	613	562	106	198	281	151	126	196	156	190	127	129	25	98	340	722	177	89	340	250	12549
mo	1099	1	385	1535	63	1121	127	791	405	268	129	1065	159	535	1935	52	281	2082	229	934	192	1748	113	663	609	947	99	574	641	2454	208	423	605	306	313	488	465	496	238	289	60	202	751	1831	197	189	842	576	29715
mt	543	2	150	496	33	363	47	117	156	258	82	155	24	205	343	102	280	323	197	758	175	316	14	219	214	291	53	190	109	217	503	168	546	282	135	181	159	212	52	108	11	191	224	588	110	47	134	508	10591
mw	1034		310	2004	80	822	96	288	316	247	134	330	53	391	631	61	245	651	177	630	134	656	28	1010	881	1415	211	839	229	404	195	1454	428	262	565	711	401	399	95	228	42	164	544	2382	198	99	286	460	23220
nr	1205	1	365	1390	90	994	128	326	337	690	177	442	64	478	833	237	651	825	674	2054	457	860	46	614	604	808	146	546	305	578	506	421	2388	817	298	486	547	471	118	324	47	567	738	1645	255	124	392	1321	28390
ns	639	2	189	599	43	482	54	156	234	346	115	194	33	225	348	133	247	342	235	819	180	338	27	293	276	408	69	243	140	259	218	214	518	850	172	233	157	254	79	134	18	215	344	735	143	58	161	619	12790
nw	753	3	224	1481	47	531	75	175	225	151	103	225	42	264	418	41	138	440	120	473	111	439	28	653	605	904	140	600	145	306	106	455	316	164	970	480	208	280	71	164	17	119	390	1536	150	73	191	273	15823
oa	860		290	2029	70	770	99	207	343	233	145	273	38	429	569	57	217	616	171	571	130	550	36	1010	860	1327	164	937	187	411	182	546	432	239	526	1522	333	384	96	185	49	143	509	2094	157	71	278	450	21795
os	1443	2	802	1730	68	1117	175	356	392	273	212	468	101	583	909	85	316	1064	312	1093	226	833	59	790	684	1061	80	748	267	687	253	444	789	339	361	528	1476	625	91	297	88	207	885	1847	143	62	346	607	26324
pl	918	2	310	1038	127	1698	217	250	750	257	349	300	42	871	604	64	228	603	161	584	137	605	32	602	458	705	97	456	212	436	191	397	429	268	264	374	289	1741	98	416	99	140	1143	1204	406	99	312	456	21439
pr	208	3	67	162	24	117	15	44	46	46	39	64	8	54	123	12	46	142	32	92	31	95	4	69	77	80	31	37	52	69	23	57	77	53	49	50	6	70	76	52	8	29	85	179	61	40	75	60	2939
re	824	1	216	754	126	1342	177	248	519	136	219	250	40	737	514	68	160	503	133	476	135	517	20	318	343	533	73	311	183	321	146	247	337	185	209	279	290	684	71	859	182	132	853	977	325	88	212	354	16627
rew			1																	1																							1	1					4
sb	562	1	129	513	42	343	57	113	148	221	86	138	36	185	331	104	279	297	216	722	162	310	24	233	230	293	88	209	123	245	218	162	484	327	125	120	132	225	45	124	28	379	241	618	133	68	126	455	10450
sh	1289	4	407	1654	137	2841	291	400	1073	337	482	434	74	1373	882	81	277	1035	283	921	235	857	59	762	680	1076	138	634	288	672	247	460	694	305	400	508	563	1474	101	791	172	209	2924	1948	518	81	423	618	32112
sm	2561	5	901	5330	155	2512	282	659	909	644	399	861	148	1218	1916	148	644	2063	549	1847	482	1877	104	2576	2311	3535	387	2178	661	1456	520	1595	1298	689	1221	1640	907	1226	238	596	148	481	1767	9396	505	224	761	1334	63864
wh	1207	2	261	508	130	394	85	163	198	143	136	228	35	271	353	50	132	333	108	344	89	417	19	269	228	331	75	238	195	225	123	231	245	156	153	205	149	238	65	151	22	113	276	633	739	140	196	240	11242
wt	681	1	129	274	42	191	45	140	86	100	52	250	40	117	368	31	91	306	50	182	51	364	15	102	122	136	50	95	128	273	56	105	161	83	111	102	95	121	98	81	10	72	164	322	99	190	229	125	6736
ww	868		233	812	68	648	84	520	235	177	114	685	81	277	1094	55	189	1113	143	532	104	1052	73	319	353	497	88	302	461	875	161	266	362	202	201	268	216	316	193	178	31	126	419	840	168	171	1045	339	17554
wy	958	2	306	1121	73	920	109	251	385	516	177	312	48	421	620	157	506	676	477	1541	336	703	29	605	503	718	106	496	222	497	447	341	1035	671	256	375	343	461	104	193	41	372	600	1391	225	94	296	2017	23053
Total Result	**52025**	**111**	**17024**	**73699**	**3877**	**52305**	**6337**	**16372**	**20358**	**15069**	**9636**	**21232**	**3220**	**26285**	**44186**	**4168**	**15106**	**44434**	**12000**	**43344**	**10662**	**41876**	**2541**	**34915**	**31762**	**45932**	**6791**	**29236**	**15640**	**31762**	**11840**	**23075**	**29993**	**16301**	**17316**	**23718**	**16789**	**26792**	**6117**	**14214**	**2857**	**10744**	**36321**	**84583**	**12864**	**5704**	**18666**	**28913**	**1118712**

HOLD SHELF DELIVERY MATRIX REPORT (CONT.)

SELECT FROM `**sierra_view.hold**` WHERE `**status**` =

code	**definition**
"b"	Bib hold ready for pickup
"j"	Volume hold ready for pickup
"i"	Item hold ready for pickup

HOLD SHELF DELIVERY MATRIX REPORT (CONT.)

- Source location:
  - `checkin_statistics_group_code_num` found in `sierra_view.item_record` table view

HOLD SHELF DELIVERY MATRIX REPORT (CONT.)

- Destination location:
    - `pickup_location_code` found in `sierra_view.hold`

HOLD SHELF DELIVERY MATRIX REPORT (CONT.)

- Note: the source value of `location_code` comes from `sierra_view.statistic_group_myuser` using `checkin_statistics_group_code_num` effectively giving us the pickup location from the stat group code num

HOLD SHELF DELIVERY MATRIX REPORT (CONT.)

- Do not insert duplicate rows of hold data...
  - We could build a complicated set of comparisons of the remotely selected data to our local data ...
  - Or, we could let the databases do all the work!
  - Create a hash of the entire hold row, use that value as the unique primary key in the local database table

HOLD SHELF DELIVERY MATRIX REPORT (CONT.)

Local SQLite table creation (simplified)

```sql
--  local SQLite table creation (simplified) ...
CREATE TABLE IF NOT EXISTS "data" (
       `hash_row` TEXT UNIQUE PRIMARY KEY
       -- more columns created below ...
);
```

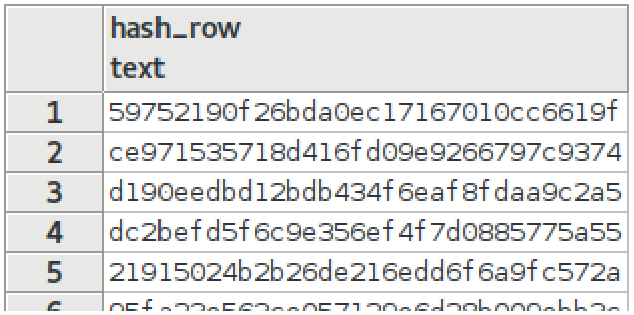
HOLD SHELF DELIVERY MATRIX REPORT (CONT.)

Remote Sierra database query (simplified)

```sql
-- remote Sierra database query (simplified) ...
SELECT
MD5(CAST((h.*) AS TEXT)) AS hash_row
-- more columns of data selected below
FROM
sierra_view.hold AS h
WHERE
h.status IN(
        'b', 'j', 'i'
);
```

HOLD SHELF DELIVERY MATRIX REPORT (CONT.)

Remote Sierra database query (simplified) output:



	hash_row text
1	59752190f26bda0ec17167010cc6619f
2	ce971535718d416fd09e9266797c9374
3	d190eedbd12bdb434f6eaf8fdaa9c2a5
4	dc2befd5f6c9e356ef4f7d0885775a55
5	21915024b2b26de216edd6f6a9fc572a
6	[illegible]

HOLD SHELF DELIVERY MATRIX REPORT (CONT.)

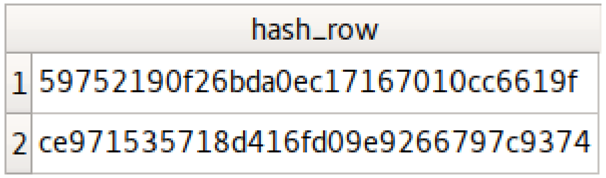
Local SQLite table inserting the retrieved data (simplified):

```
INSERT OR IGNORE INTO data (
        hash_row
)
VALUES ('59752190f26bda0ec17167010cc6619f'),
       ('59752190f26bda0ec17167010cc6619f'),
       ('ce971535718d416fd09e9266797c9374');

SELECT * FROM DATA;
```

HOLD SHELF DELIVERY MATRIX REPORT (CONT.)

Local SQLite table inserting the retrieved data (simplified):

```
INSERT OR IGNORE INTO data (
        hash_row
)
VALUES ('59752190f26bda0ec17167010cc6619f'),
       ('59752190f26bda0ec17167010cc6619f'),
       ('ce971535718d416fd09e9266797c9374');

SELECT * FROM DATA;
```



	hash_row
1	59752190f26bda0ec17167010cc6619f
2	ce971535718d416fd09e9266797c9374

HOLD SHELF DELIVERY MATRIX REPORT (CONT.)

- Overview of the process: (this should look familiar)
  1. Connect to our local database, and create table if it doesn’t exist. Also, establish connection to the remote Sierra database
  2. Query the Sierra ‘hold’ table for rows that have status of ‘i’, ‘j’, or ‘b’; this indicates that there is a item hold, volume hold, or bib hold ready for pickup

HOLD SHELF DELIVERY MATRIX REPORT (CONT.)

- Overview of the process: (cont.)
  3. Insert retrieved rows (or ignore duplicate rows as explained previously) to the local SQLite database, then close all connections
  4. Set the update of local data to happen frequently (every 5 minutes via CRON is a good method for doing this)

HOLD SHELF DELIVERY MATRIX REPORT (CONT.)

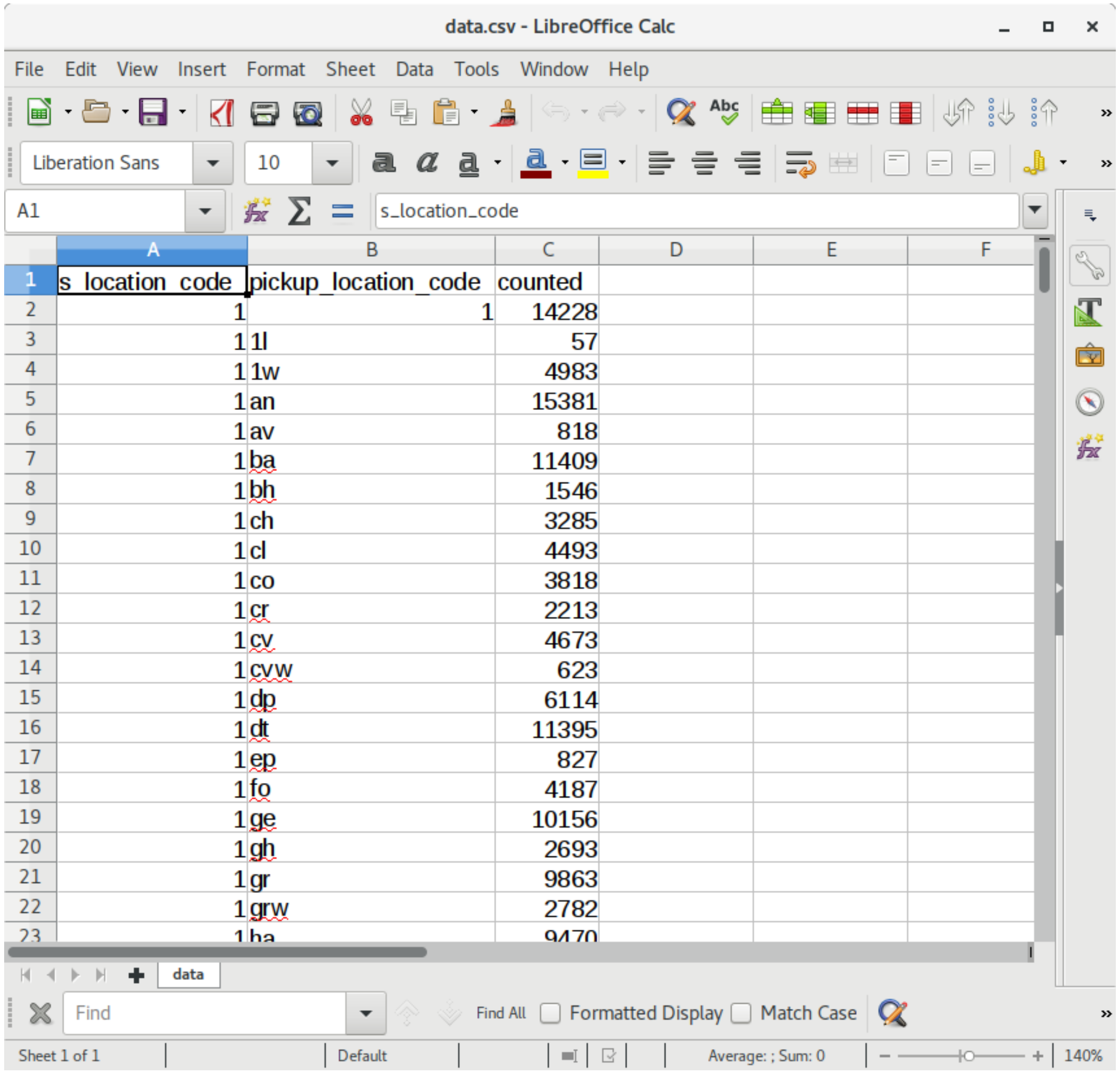
- Exporting / Producing the report
  1. Query the local SQLite database, then export the results to a .csv file:
     github.com/plch/plch-holds-shelf/blob/master/export_csv.sh
     github.com/plch/plch-holds-shelf/blob/master/export.sql

HOLD SHELF DELIVERY MATRIX REPORT (CONT.)

- Exporting / Producing the report (cont.)

  2. Import the .csv file into LibreOffice Calc (or Excel) and perform a pivot on the data:

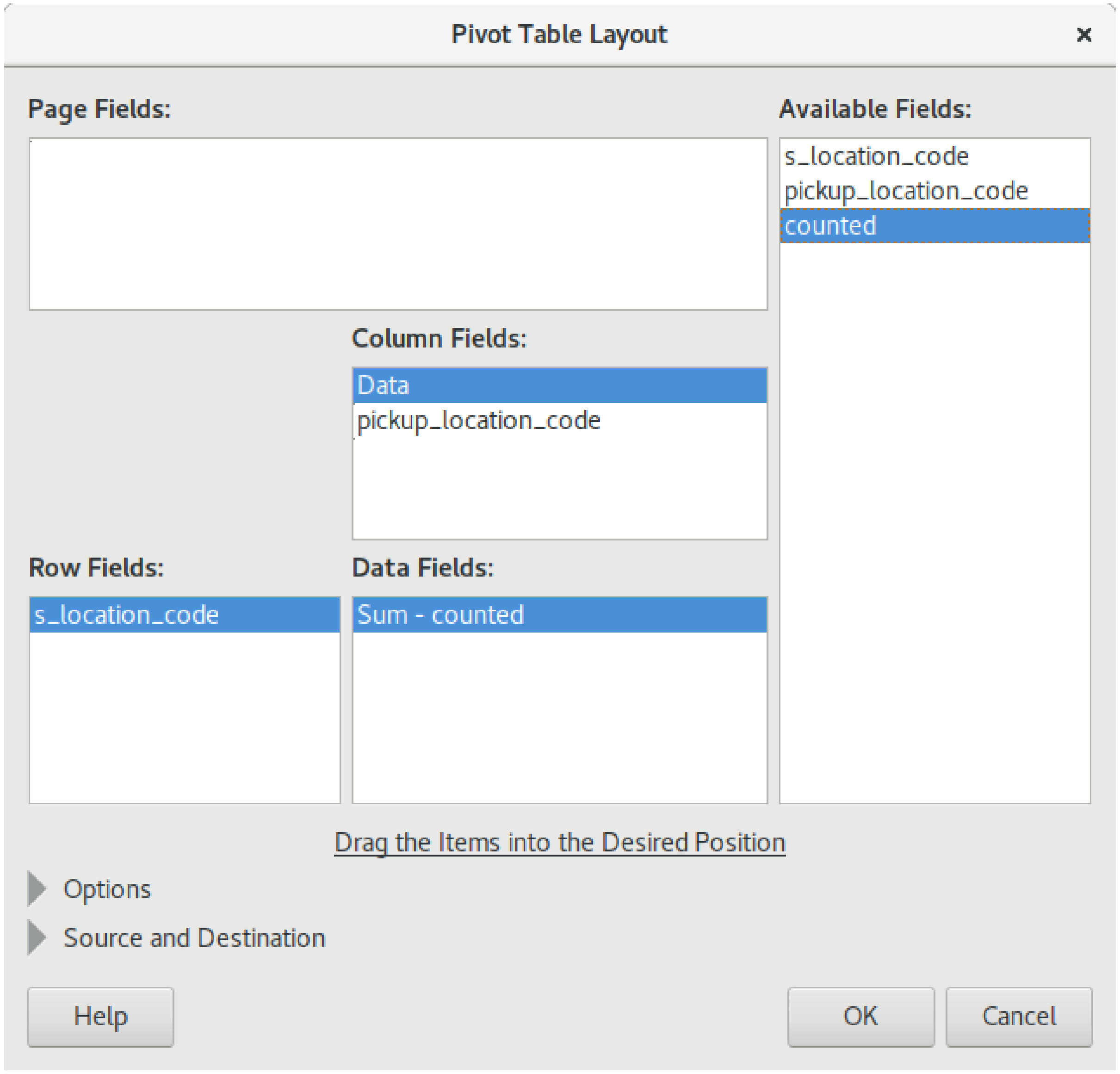
- Exporting / Producing the report (cont.)



	A	B	C
1	s_location_code	pickup_location_code	counted
2	1	1	14228
3	1	1l	57
4	1	1w	4983
5	1	an	15381
6	1	av	818
7	1	ba	11409
8	1	bh	1546
9	1	ch	3285
10	1	cl	4493
11	1	co	3818
12	1	cr	2213
13	1	cv	4673
14	1	cvw	623
15	1	dp	6114
16	1	dt	11395
17	1	ep	827
18	1	fo	4187
19	1	ge	10156
20	1	gh	2693
21	1	gr	9863
22	1	grw	2782
23	1	ha	9470

HOLD SHELF DELIVERY MATRIX REPORT (CONT.)

• Exporting / Producing the report (cont.)


Pivot Table Layout

Page Fields:

Available Fields:
s_location_code
pickup_location_code
counted

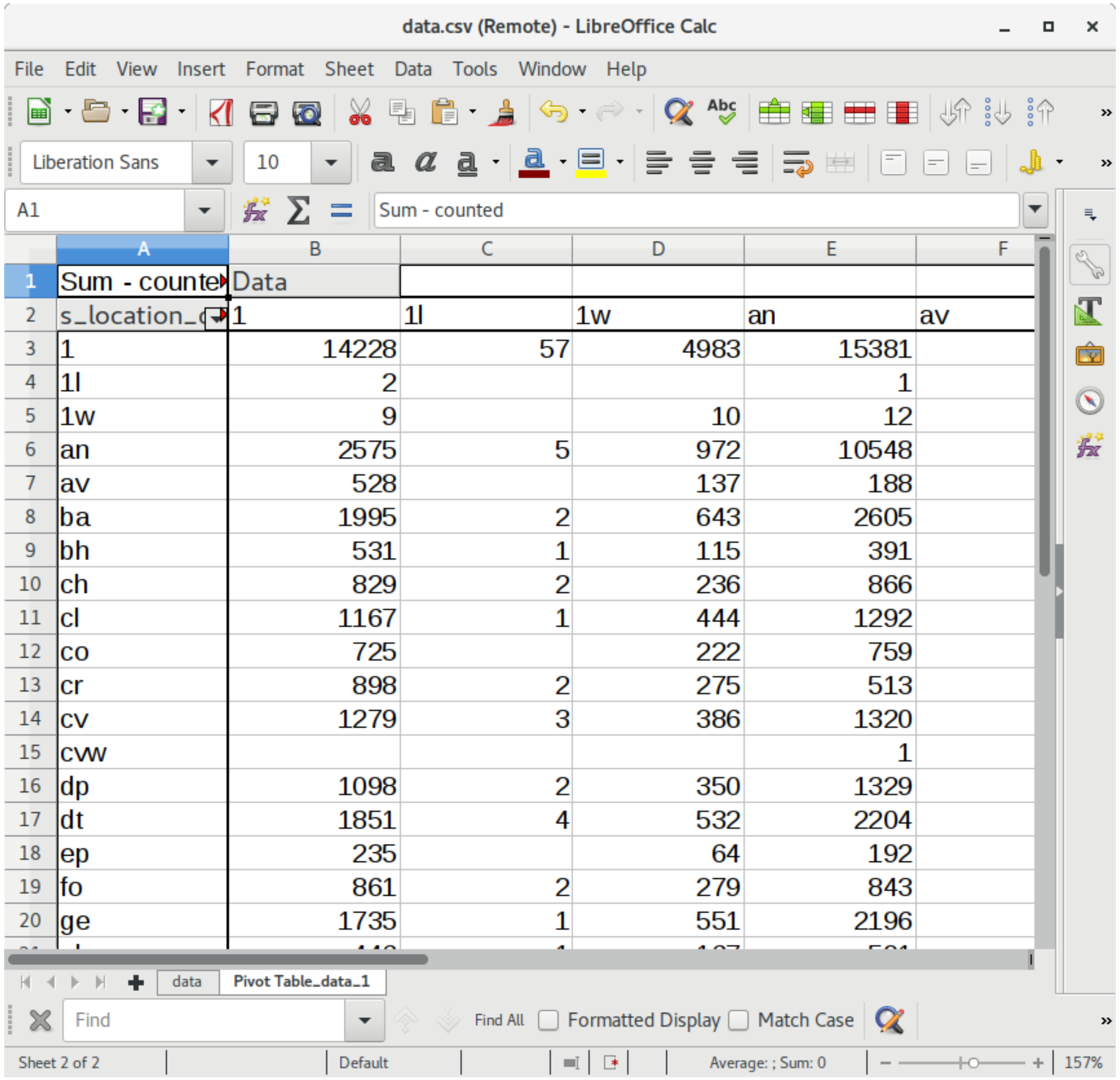
Column Fields:
Data
pickup_location_code

Row Fields:
s_location_code

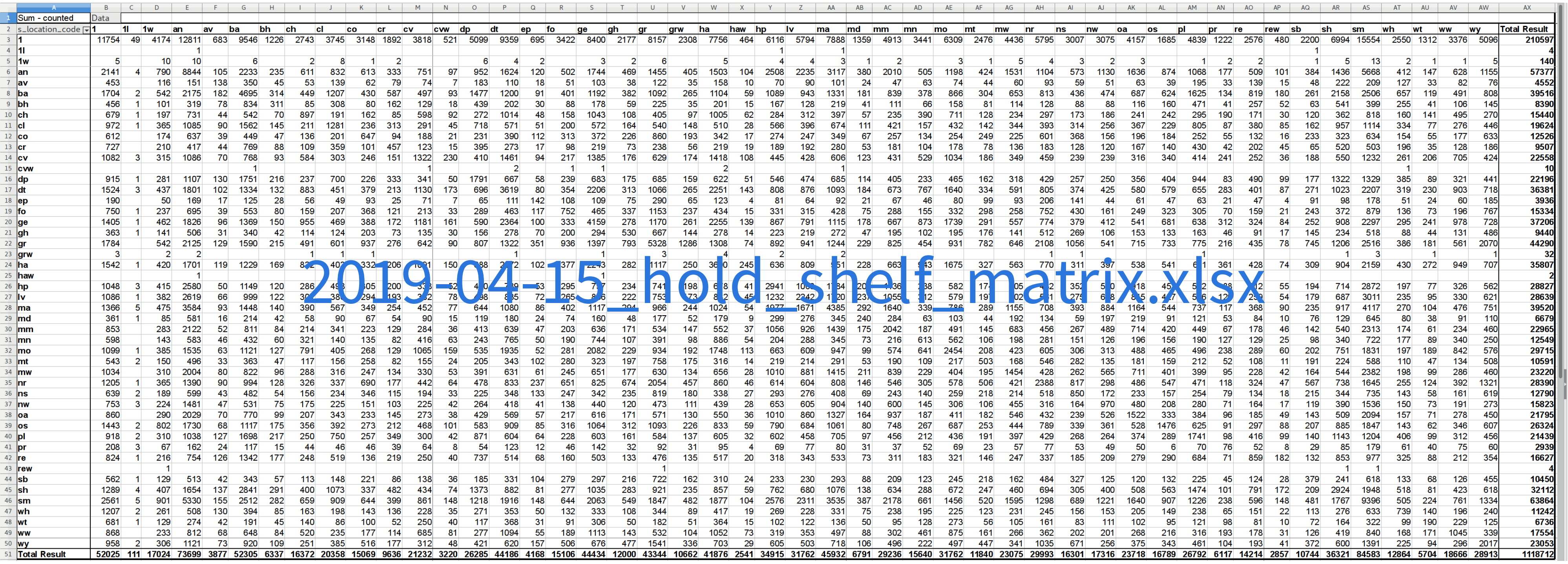
Data Fields:
Sum - counted

Drag the Items into the Desired Position

- Exporting / Producing the report (cont.)



	A	B	C	D	E	F
1	Sum - counte	Data				
2	s_location_c	1	1l	1w	an	av
3	1	14228	57	4983	15381	
4	1l	2			1	
5	1w	9		10	12	
6	an	2575	5	972	10548	
7	av	528		137	188	
8	ba	1995	2	643	2605	
9	bh	531	1	115	391	
10	ch	829	2	236	866	
11	cl	1167	1	444	1292	
12	co	725		222	759	
13	cr	898	2	275	513	
14	cv	1279	3	386	1320	
15	cvw				1	
16	dp	1098	2	350	1329	
17	dt	1851	4	532	2204	
18	ep	235		64	192	
19	fo	861	2	279	843	
20	ge	1735	1	551	2196	



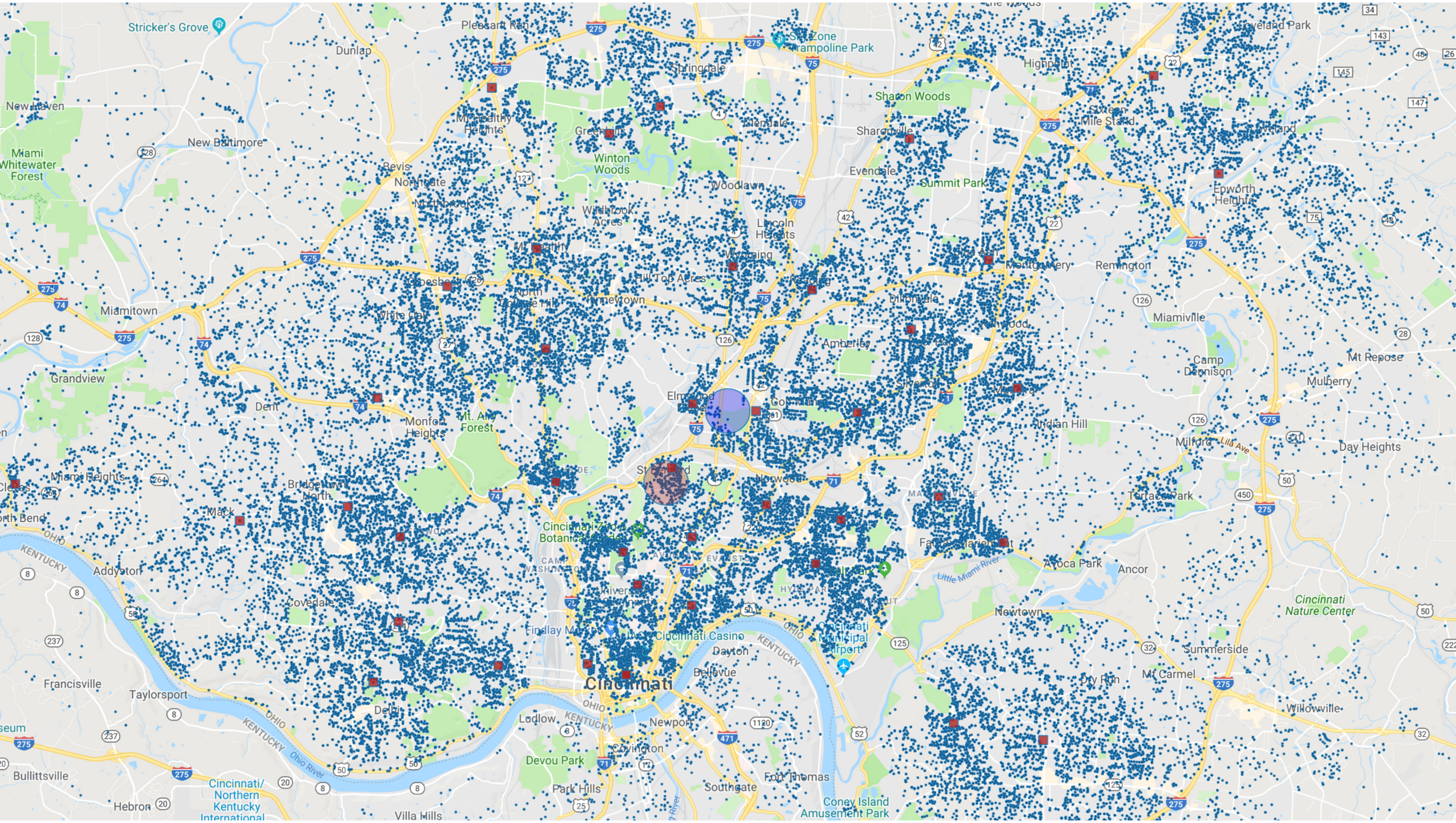
2019-04-15_hold_shelf_matrix.xlsx

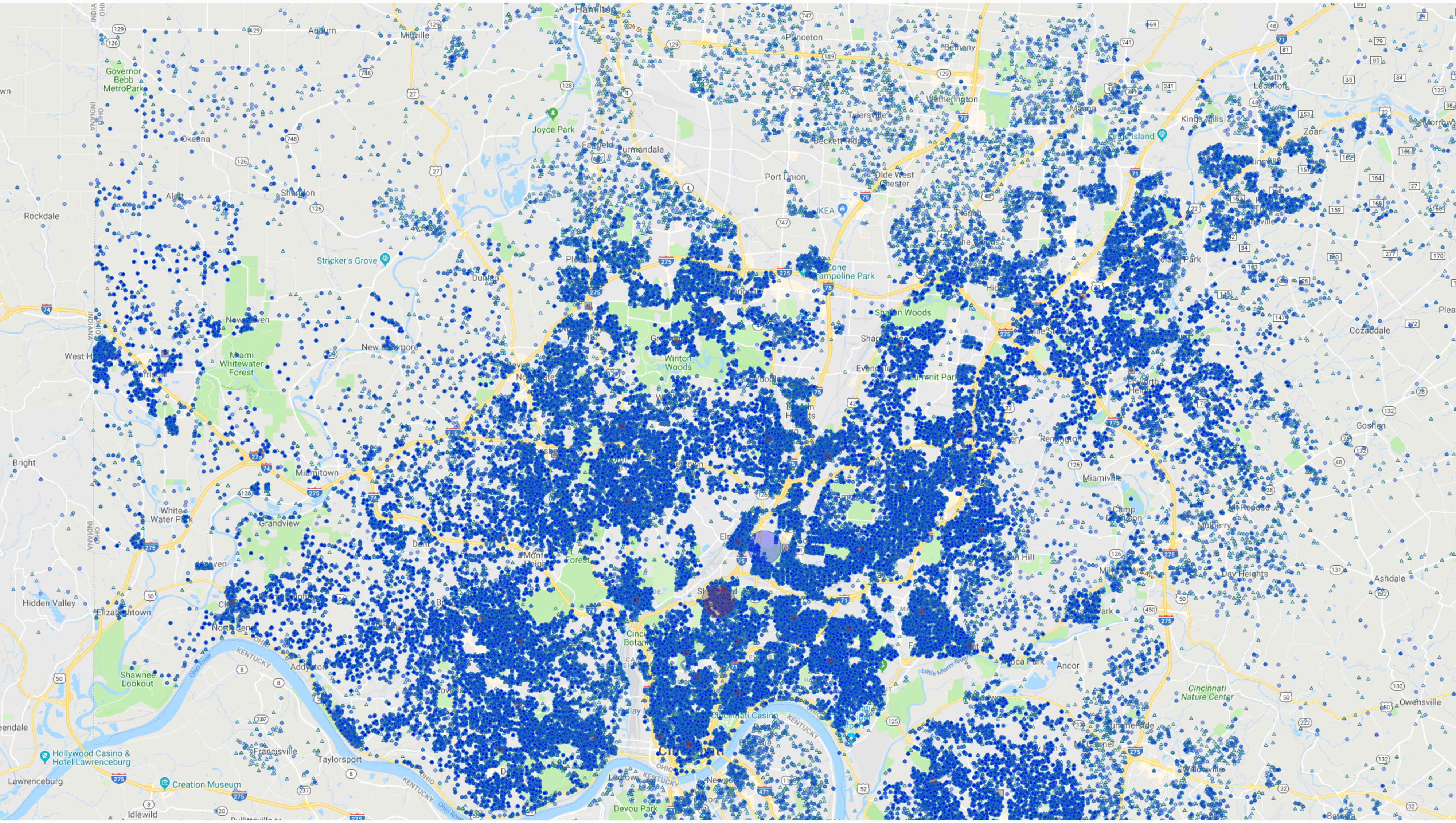
Sum - counted	Data																																																
s_location_code	1	1l	1w	an	av	ba	bh	ch	cl	co	cr	cv	cvw	dp	dt	ep	fo	ge	gh	gr	grw	ha	haw	hp	lv	ma	md	mm	mn	mo	mt	mw	nr	ns	nw	oa	os	pl	pr	re	rew	sb	sh	sm	wh	wt	ww	wy	Total Result
1	11754	49	4174	12811	683	9546	1226	2743	3745	3148	1892	3818	521	5099	9359	695	3422	8400	2177	8157	2308	7756	464	6116	5794	7888	1359	4913	3441	6309	2476	4436	5795	3007	3075	4157	1685	4839	1222	2576	480	2200	6994	15554	2550	1312	3376	5096	210597
1l				1																				1		1																1							4
1w	5		10	10		6		2	8	1	2			6	4	2		3	2	6		5		4	4	3	1	2		3	1	5	4	3	2	3		1	2	2		1	5	13	2	1	1	5	140
an	2141	4	790	8844	105	2233	235	611	832	613	333	751	97	952	1624	120	502	1744	469	1455	405	1503	104	2508	2235	3117	380	2010	505	1198	424	1531	1104	573	1130	1636	874	1068	177	509	101	384	1436	5668	412	147	628	1155	57377
av	453		116	151	138	350	45	53	139	62	79	74	7	183	110	18	51	103	38	122	35	158	10	70	90	101	24	47	63	74	44	60	93	59	51	63	39	195	33	139	15	48	222	209	127	33	82	76	4552
ba	1704	2	542	2175	182	4695	314	449	1207	430	587	497	93	1477	1200	91	401	1192	382	1092	265	1104	59	1089	943	1331	181	839	378	866	304	653	813	436	474	687	624	1625	134	819	180	261	2158	2506	657	119	491	808	39516
bh	456	1	101	319	78	834	311	85	308	80	162	129	18	439	202	30	88	178	59	225	35	201	15	167	128	219	41	111	66	158	81	114	128	88	88	116	160	471	41	257	52	63	541	399	255	41	106	145	8390
ch	679	1	197	731	44	542	70	897	191	162	85	598	92	272	1014	48	158	1043	108	405	97	1005	62	284	312	397	57	235	390	711	128	234	297	173	186	241	242	295	190	171	30	120	362	818	160	141	495	270	15440
cl	972	1	365	1085	90	1562	145	211	1281	236	313	291	45	718	571	51	200	572	164	540	148	510	28	566	396	674	111	421	157	432	142	344	393	314	256	367	229	805	87	380	85	162	957	1114	334	77	276	446	19624
co	612		174	637	39	449	47	136	201	647	94	188	21	231	390	112	313	372	226	860	193	342	17	274	247	349	67	257	134	254	249	225	601	368	156	196	184	252	55	132	16	233	323	634	154	55	177	633	12526
cr	727		210	417	44	769	88	109	359	101	457	123	15	395	273	17	98	219	73	238	56	219	19	189	192	280	53	181	104	178	78	136	183	128	120	167	140	430	42	202	45	65	520	503	196	35	128	186	9507
cv	1082	3	315	1086	70	768	93	584	303	246	151	1322	230	410	1461	94	217	1385	176	629	174	1418	108	445	428	606	123	431	529	1034	186	349	459	239	239	316	340	414	241	252	36	188	550	1232	261	206	705	424	22558
cvw						1						1			2		1	1				2				1																			1				10
dp	915	1	281	1107	130	1751	216	237	700	226	333	341	50	1791	667	58	239	683	175	685	159	622	51	546	474	685	114	405	233	465	162	318	429	257	250	356	404	944	83	490	99	177	1322	1329	385	89	321	441	22196
dt	1524	3	437	1801	102	1334	132	883	451	379	213	1130	173	696	3619	80	354	2206	313	1066	265	2251	143	808	876	1093	184	673	767	1640	334	591	805	374	425	580	579	655	283	401	87	271	1023	2207	319	230	903	718	36381
ep	190		50	169	17	125	28	56	49	93	25	71	7	65	111	142	108	109	75	290	65	123	4	81	64	92	21	67	46	80	99	93	206	141	44	61	47	63	21	47	4	91	98	178	51	24	60	185	3936
fo	750	1	237	695	39	553	80	159	207	368	121	213	33	289	463	117	752	465	337	1153	237	434	15	331	315	428	75	288	155	332	298	258	752	430	161	249	323	305	70	159	21	243	372	879	136	73	196	767	15334
ge	1405	1	462	1826	96	1369	150	955	469	388	172	1181	161	590	2364	100	333	4159	278	1170	261	2255	139	867	791	1115	178	667	873	1739	291	557	774	379	412	541	681	638	312	324	84	252	908	2297	295	241	978	728	37206
gh	363	1	141	506	31	340	42	114	124	203	73	135	30	156	278	70	200	294	530	667	144	278	14	223	219	272	47	195	102	195	176	141	512	269	106	153	133	163	46	91	17	145	234	518	88	44	131	486	9440
gr	1784		542	2125	129	1590	215	491	601	937	276	642	90	807	1322	351	936	1397	793	5328	1286	1308	74	892	941	1244	229	825	454	931	782	646	2108	1056	541	715	733	775	216	435	78	745	1206	2516	386	181	561	2070	44290
grw	3		2	2				1	1	1					1			1		3		4		2		2							1	1				1					1	3		1		1	32
ha	1542	1	420	1701	119	1229	169	[illegible]	[illegible]	332	206	[illegible]	150	[illegible]	[illegible]	102	[illegible]	[illegible]	282	1117	250	[illegible]	245	636	809	[illegible]	228	663	[illegible]	1675	327	563	770	311	397	538	541	[illegible]	361	428	74	309	904	2159	430	272	949	707	35807
haw				1														1																															2
hp	1048	3	415	2580	50	1149	120	286	[illegible]	[illegible]	200	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	295	[illegible]	234	741	198	[illegible]	41	2941	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	582	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	55	194	714	2872	197	77	326	562	28827
lv	1086	1	382	2619	66	999	122	[illegible]	[illegible]	294	193	[illegible]	78	[illegible]	[illegible]	72	[illegible]	[illegible]	222	753	[illegible]	[illegible]	[illegible]	1232	[illegible]	[illegible]	[illegible]	1055	[illegible]	579	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	54	179	687	3011	235	95	330	621	28639
ma	1366	5	475	3584	93	1448	140	390	567	349	254	452	77	644	1080	86	402	1117	294	966	244	1024	54	[illegible]	1671	4385	292	1640	339	[illegible]	289	1155	708	393	884	1164	544	737	117	368	90	235	917	4117	270	104	476	751	39520
md	361	1	85	581	16	214	42	58	90	67	54	90	15	119	180	24	74	160	48	177	52	179	9	299	276	345	240	284	63	103	44	192	134	59	197	219	91	121	53	84	10	76	129	645	80	38	91	110	6679
mm	853		283	2122	52	811	84	214	341	223	129	284	36	413	639	47	203	636	171	534	147	552	37	1056	926	1439	175	2042	187	491	145	683	456	267	489	714	420	449	67	178	46	142	540	2313	174	61	234	460	22965
mn	598		143	583	46	432	60	321	140	135	82	416	63	243	765	50	190	744	107	391	98	886	54	204	288	345	73	216	613	562	106	198	281	151	126	196	156	190	127	129	25	98	340	722	177	89	340	250	12549
mo	1099	1	385	1535	63	1121	127	791	405	268	129	1065	159	535	1935	52	281	2082	229	934	192	1748	113	663	609	947	99	574	641	2454	208	423	605	306	313	488	465	496	238	289	60	202	751	1831	197	189	842	576	29715
mt	543	2	150	496	33	363	47	117	156	258	82	155	24	205	343	102	280	323	197	758	175	316	14	219	214	291	53	190	109	217	503	168	546	282	135	181	159	212	52	108	11	191	224	588	110	47	134	508	10591
mw	1034		310	2004	80	822	96	288	316	247	134	330	53	391	631	61	245	651	177	630	134	656	28	1010	881	1415	211	839	229	404	195	1454	428	262	565	711	401	399	95	228	42	164	544	2382	198	99	286	460	23220
nr	1205	1	365	1390	90	994	128	326	337	690	177	442	64	478	833	237	651	825	674	2054	457	860	46	614	604	808	146	546	305	578	506	421	2388	817	298	486	547	471	118	324	47	567	738	1645	255	124	392	1321	28390
ns	639	2	189	599	43	482	54	156	234	346	115	194	33	225	348	133	247	342	235	819	180	338	27	293	276	408	69	243	140	259	218	214	518	850	172	233	157	254	79	134	18	215	344	735	143	58	161	619	12790
nw	753	3	224	1481	47	531	75	175	225	151	103	225	42	264	418	41	138	440	120	473	111	439	28	653	605	904	140	600	145	306	106	455	316	164	970	480	208	280	71	164	17	119	390	1536	150	73	191	273	15823
oa	860		290	2029	70	770	99	207	343	233	145	273	38	429	569	57	217	616	171	571	130	550	36	1010	860	1327	164	937	187	411	182	546	432	239	526	1522	333	384	96	185	49	143	509	2094	157	71	278	450	21795
os	1443	2	802	1730	68	1117	175	356	392	273	212	468	101	583	909	85	316	1064	312	1093	226	833	59	790	684	1061	80	748	267	687	253	444	789	339	361	528	1476	625	91	297	88	207	885	1847	143	62	346	607	26324
pl	918	2	310	1038	127	1698	217	250	750	257	349	300	42	871	604	64	228	603	161	584	137	605	32	602	458	705	97	456	212	436	191	397	429	268	264	374	289	1741	98	416	99	140	1143	1204	406	99	312	456	21439
pr	208	3	67	162	24	117	15	44	46	46	39	64	8	54	123	12	46	142	32	92	31	95	4	69	77	80	31	37	52	69	23	57	77	53	49	50	6	70	76	52	8	29	85	179	61	40	75	60	2939
re	824	1	216	754	126	1342	177	248	519	136	219	250	40	737	514	68	160	503	133	476	135	517	20	318	343	533	73	311	183	321	146	247	337	185	209	279	290	684	71	859	182	132	853	977	325	88	212	354	16627
rew			1																	1																							1	1					4
sb	562	1	129	513	42	343	57	113	148	221	86	138	36	185	331	104	279	297	216	722	162	310	24	233	230	293	88	209	123	245	218	162	484	327	125	120	132	225	45	124	28	379	241	618	133	68	126	455	10450
sh	1289	4	407	1654	137	2841	291	400	1073	337	482	434	74	1373	882	81	277	1035	283	921	235	857	59	762	680	1076	138	634	288	672	247	460	694	305	400	508	563	1474	101	791	172	209	2924	1948	518	81	423	618	32112
sm	2561	5	901	5330	155	2512	282	659	909	644	399	861	148	1218	1916	148	644	2063	549	1847	482	1877	104	2576	2311	3535	387	2178	661	1456	520	1595	1298	689	1221	1640	907	1226	238	596	148	481	1767	9396	505	224	761	1334	63864
wh	1207	2	261	508	130	394	85	163	198	143	136	228	35	271	353	50	132	333	108	344	89	417	19	269	228	331	75	238	195	225	123	231	245	156	153	205	149	238	65	151	22	113	276	633	739	140	196	240	11242
wt	681	1	129	274	42	191	45	140	86	100	52	250	40	117	368	31	91	306	50	182	51	364	15	102	122	136	50	95	128	273	56	105	161	83	111	102	95	121	98	81	10	72	164	322	99	190	229	125	6736
ww	868		233	812	68	648	84	520	235	177	114	685	81	277	1094	55	189	1113	143	532	104	1052	73	319	353	497	88	302	461	875	161	266	362	202	201	268	216	316	193	178	31	126	419	840	168	171	1045	339	17554
wy	958	2	306	1121	73	920	109	251	385	516	177	312	48	421	620	157	506	676	477	1541	336	703	29	605	503	718	106	496	222	497	447	341	1035	671	256	375	343	461	104	193	41	372	600	1391	225	94	296	2017	23053
Total Result	**52025**	**111**	**17024**	**73699**	**3877**	**52305**	**6337**	**16372**	**20358**	**15069**	**9636**	**21232**	**3220**	**26285**	**44186**	**4168**	**15106**	**44434**	**12000**	**43344**	**10662**	**41876**	**2541**	**34915**	**31762**	**45932**	**6791**	**29236**	**15640**	**31762**	**11840**	**23075**	**29993**	**16301**	**17316**	**23718**	**16789**	**26792**	**6117**	**14214**	**2857**	**10744**	**36321**	**84583**	**12864**	**5704**	**18666**	**28913**	**1118712**

EXAMPLE 4

MAPPING GEO DATA FROM PATRON ADDRESS DATA AND CIRCULATION TRANSACTION DATA

- The purpose of this process is to plot patron locations and branch locations on a map based on latitude / longitude coordinates derived from mailing address data
- This is a work in progress!




Hamilton
Millville
Auburn
Princeton
Bethany
Governor Bebb MetroPark
Okeana
Joyce Park
Fairfield
Tylersville
Wetherington
Mason
Kings Mills
South Lebanon
Zoar
Kings Island
Beckett Ridge
Olde West Chester
Port Union
IKEA
Shandon
Alert
Rockdale
Stricker's Grove
Dunlap
Sharon Woods
Winton Woods
New Haven
New Baltimore
Miami Whitewater Forest
West Harrison
Evendale
Summit Park
Remington
Cozaddale
Goshen
Miamitown
Bright
White Water Park
Grandview
Miamiville
Camp Dennison
Mulberry
Mt Repose
Day Heights
Ashdale
Hidden Valley
Elizabethtown
North Bend
Addyston
Shawnee Lookout
Hollywood Casino & Hotel Lawrenceburg
Lawrenceburg
Creation Museum
Francisville
Taylorsport
Idlewild
Cincinnati
Cincinnati Nature Center
Owensville
Ancor
Newport
Ludlow
Devou Park
Hard Rock Cincinnati Casino
Little Miami River
Ohio River
KENTUCKY
OHIO
INDIANA

MAPPING GEO DATA (CONT.)

- Cached data is contained in an SQLite database
    - Circulation data
      (weekly export of `sierra_view.circ_trans` table view)
    - Patron data
      (weekly export of relevant patron information from multiple table views)

MAPPING GEO DATA (CONT.)

- Geocoding Patron Street Addresses:
  - Patron address data (**patron_record_id**, **street number**, **street name**, **city**, **zip**) are exported to a .csv file

MAPPING GEO DATA (CONT.)

- Possible Geocoding Services:
  - Census.gov
    www.census.gov/data/developers/data-sets/Geoco
    services.html
  - Google
    developers.google.com/maps/documentation/java
  - SmartyStreets
    smartystreets.com/products/list

MAPPING GEO DATA (CONT.)

- SmartyStreets has some very user-friendly services for bulk upload / download of address data for verification and geocoding
- SmartyStreets may be able to offer a discounted / free service to libraries that allow for bulk verification / geocoding as well as on-the-fly verification and auto-corrected address inputs for things such as web input forms

MAPPING GEO DATA (CONT.)

- SmartyStreets list service returns .csv data back with relevant address information, identified by the unique ID (`patron_record_id`) that was provided
- We may easily load this into the local SQLite database with the import csv feature (from the GUI)

MAPPING GEO DATA (CONT.)

- There are very good data analysis / visualization tools available for Python:
  - PyViz
    pyviz.org
  - A Conda metapackage "pyviz"
  - Makes data visualization in Python easier to use

jupyter_notebook_example.pdf