



Never Been a Joiner

Sierra SQL for the SQL Challenged

Nathan James



Getting Started

Required software

- Sierra
- Windows
 - Access does not run on Mac OS
- Microsoft Access
 - Access 2016 is used for this presentation
- pgAdmin 4
 - Get it at <https://www.pgadmin.org/download/>
 - Runs in a browser window
- What about pgAdmin 3?
 - Included a visual query editor
 - Editor was very hard to use (in my opinion)



Required reading

- User Manual section: Sierra Direct SQL Access
- https://csdirect.iii.com/sierrahelp/Default.htm#ssql/ssql_direct_access.html
- Getting Started - Setting Up SQL Client Connections has info needed for pgAdmin



#IUG2020



Grant Permission to Sierra SQL Access Application

- Do this in Sierra Admin
 - <https://<your.sierra.server>/sierra/admin>
 - Login
 - Click Authorizations and Authentication
 - Click Applications
 - Edit Sierra SQL Access
 - Add login to be used for queries as an assigned user



Sierra SQL Access Application Permission

Home | Authorizations and Authentication Admin You are signed in as Nathan James (lnj) | [more](#) ▼

[Users](#)

Applications

[Permissions](#)

[Workflows](#)

[User Groups](#)

[< back to Applications](#)

Sierra SQL Access

[SAVE](#) [Discard changes](#)

Assigned users (6) [remove all](#)

Filter list by typing here

Angelic Sauk
Cyntaxs Acc
Dawn Perry
Nathan Jame
Unique SQL
mike curtis (n

Available users (700) [add all](#)

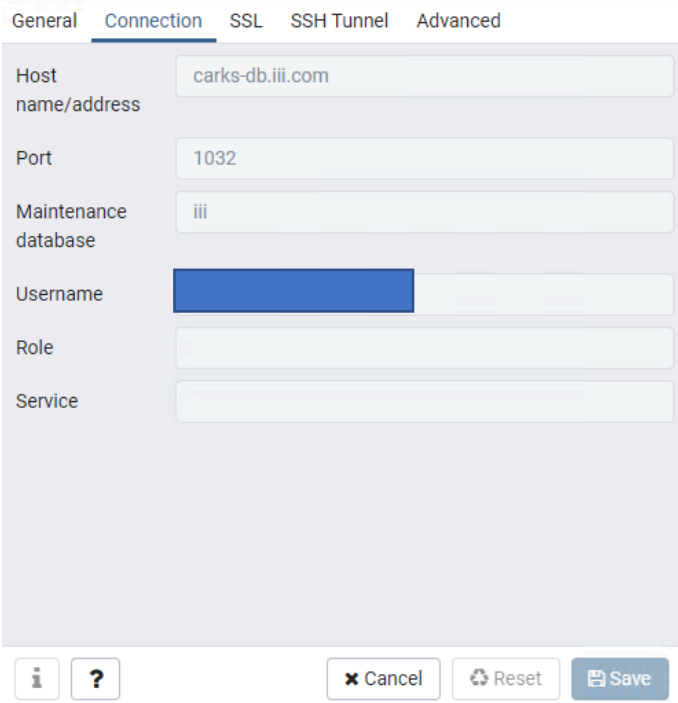
Filter list by typing here

(webpe)
(webwm)
(webth)
(temil3)
(temil4)
(temil5)
(serstaff)
(temil1)
(temil2)



Setting up server in pgAdmin 4

- Set up pgAdmin 4 for your Sierra Database Server
 - Use a login with Sierra SQL Access permission
 - Server connection properties should look like the following
 - Host name: your Sierra database server
 - Port: 1032
 - Maintenance database: iii
 - Username: a user with SQL Access permission



The screenshot shows the 'Connection' tab in pgAdmin 4. The fields are filled with the following values:

Field	Value
Host name/address	cars-db.iii.com
Port	1032
Maintenance database	iii
Username	[Redacted]
Role	
Service	

At the bottom, there are buttons for 'Cancel', 'Reset', and 'Save', along with information and help icons.



Access: Reconstructing sierra_view

Start with sierra_view table definitions

- sierra_view is used when querying via Sierra SQL Access
- Table definitions are in Sierra DNA
 - Sierra DNA is at <https://techdocs.iii.com>
- How to find the table(s) you need?
 - Ask an expert (if you know one)
 - Fixed-length fields are in individual tables by record type
 - All variable-length fields are in one large table
- For this presentation we'll use
 - sierra_view.fine
 - sierra_view.patron_record_address
 - sierra_view.varfield_view



Recreate the table and columns in Access

- Use Create Menu – Table Design Tool
- Name table sv + table name from Sierra DNA - svpatron_record_address
 - This will be replaced with sierra_view. in pgAdmin
- Add each column/field name as it appears in Sierra DNA
- Set the Data Type for each column/field to be similar to the Sierra table (optional)
 - This isn't necessary since we aren't going to populate these tables with data
- Don't panic!
 - Will share my latest sierra_view Access database on request
 - Disclaimer: I only add tables as needed so it is incomplete and may have errors!
 - Yet Another Disclaimer: I'm not a SQL expert!
 - If something doesn't work, I'll try my best to assist
 - Thus far my focus has been on patron, item, and circulation data queries



sierra_view.patron_record_address in Sierra DNA

patron_record_address

Each row of patron_record_address contains the address data associated with a patron record.

Always read this carefully!

Column	Data Type	Not NULL?	Comment
id	bigint	false	System-generated sequential ID.
patron_record_id	bigint	false	Foreign key to patron_record.
patron_record_address_type_id	bigint	false	Foreign key to patron_record_address_type.
display_order	int	false	Integer to manage the display order of a list.
addr1	varchar	false	The first line of the patron address.
addr2	varchar	false	The second line of the patron address.
addr3	varchar	false	The third line of the patron address.
village	varchar	false	The village of the patron address. (Used only at UK sites.)
city	varchar	false	The city of the patron address.
region	varchar	false	The region of the patron address. (US State or Canadian Province)
postal_code	varchar	false	The postal code of the patron address.
country	varchar	false	The country of the patron address.



svpatron_record_address in Access

The screenshot displays the Microsoft Access interface. On the left, the 'All Access Objects' pane shows a list of tables, with 'svpatron_record_address' selected. The main window shows the table structure for 'svpatron_record_address' with the following fields:

Field Name	Data Type	Description (Optional)
id	Number	
patron_record_id	Number	
patron_record_address_type	Number	
display_order	Number	
addr1	Short Text	
addr2	Short Text	
addr3	Short Text	
village	Short Text	
city	Short Text	
region	Short Text	
postal_code	Short Text	
country	Short Text	

Below the table structure, the 'Field Properties' task pane is open for the 'id' field, showing the following properties:

Property	Value
Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	0
Validation Rule	
Validation Text	
Required	No
Indexed	Yes (Duplicates OK)
Text Align	General

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.



Comparing the tables

patron_record_address

Each row of patron_record_address contains the addr

Column	Data Type
id	bigint
patron_record_id	bigint
patron_record_address_type_id	bigint
display_order	int
addr1	varchar
addr2	varchar
addr3	varchar
village	varchar
city	varchar
region	varchar
postal_code	varchar
country	varchar

svpatron_record_address	
Field Name	Data Type
id	Number
patron_record_id	Number
patron_record_address_type_id	Number
display_order	Number
addr1	Short Text
addr2	Short Text
addr3	Short Text
village	Short Text
city	Short Text
region	Short Text
postal_code	Short Text
country	Short Text

Table name begins with sv





Access: Creating a Query

A simple example: Fine data with patron zip code

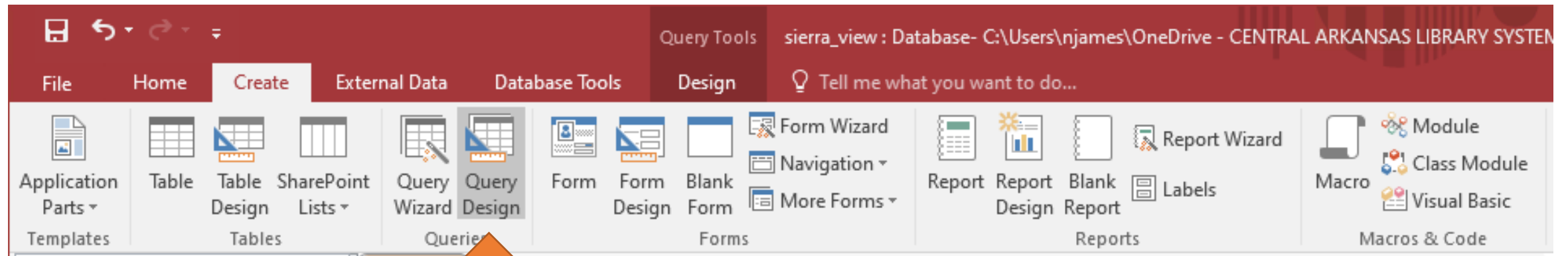
- Problem: How to do a geographic analysis of fines in our service area
- Two sierra_view tables are needed
 - sierra_view.fine
 - sierra_view.patron_record_address
- What field do these tables have in common?
 - Look them up in Sierra DNA
 - Find a common column/field
 - Both have this column/field – patron_record_id
 - For every row in sierra_view.fine we want to match the patron_record_id to the patron_record_id in sierra_view.patron_record_address in order to get the zip code



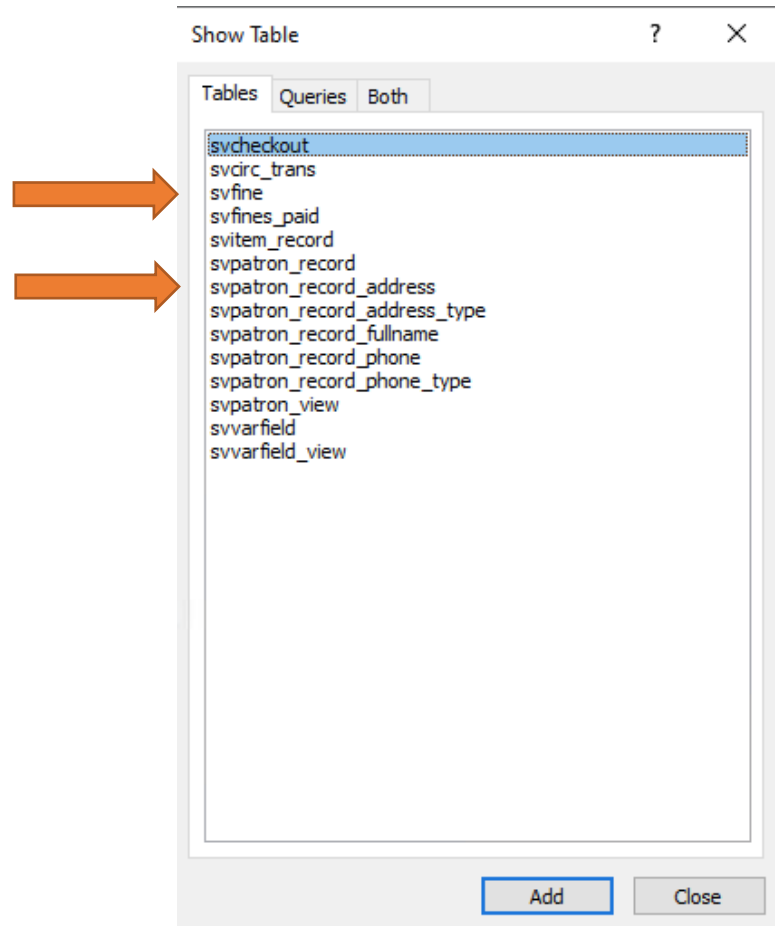
Do this in Access

- Create Menu - Query Design tool
- Add the tables you need (or create new ones first)
- Drag the common field from one table and drop it on the same field in other table(s)
 - This links the tables using a JOIN
- Double-click each field you want in your output
 - Add any criteria or sort options in the table at bottom of tool
- Switch to SQL View and copy the SQL statement for use in pgAdmin

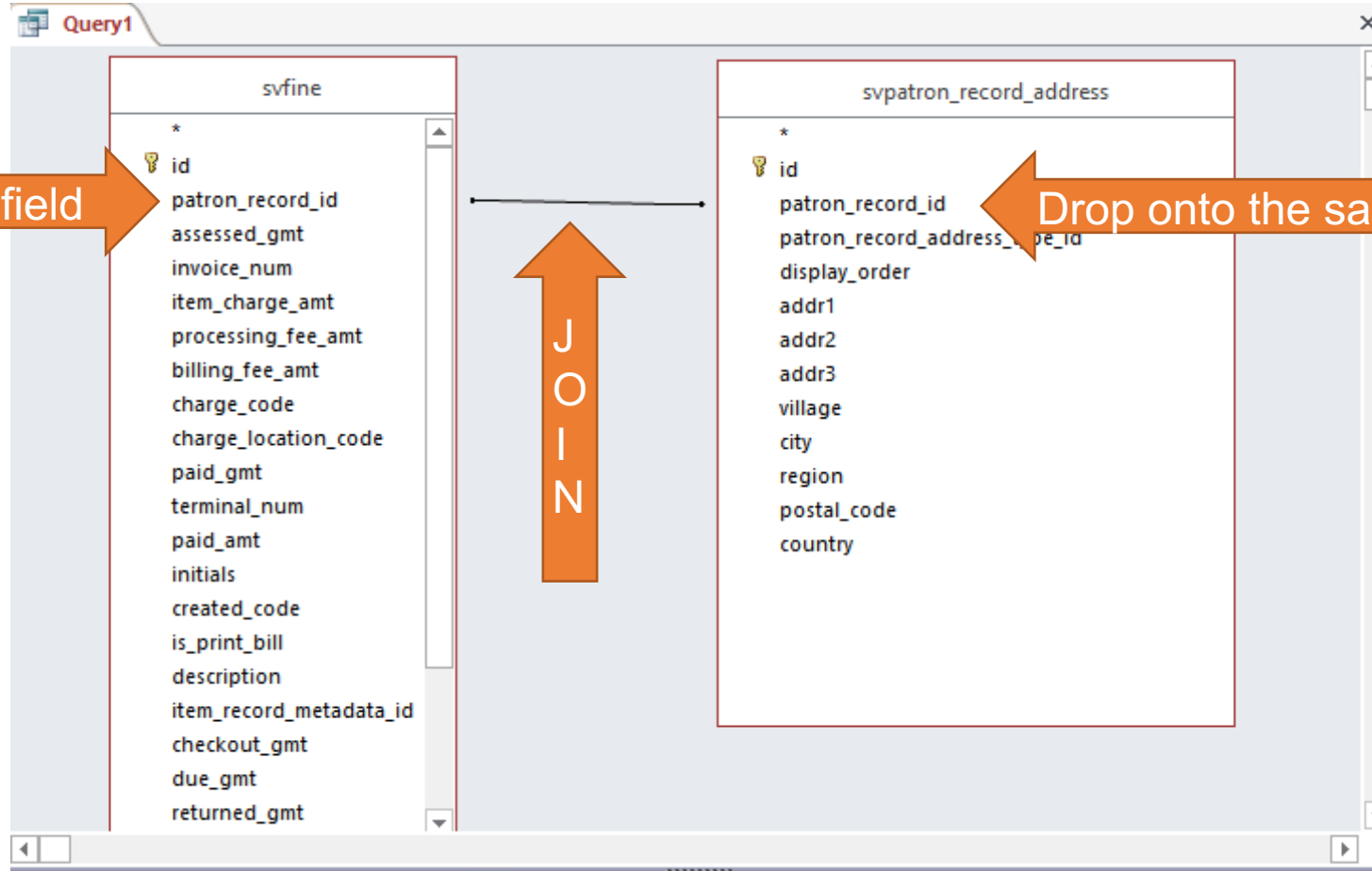
Create Menu - Query Design tool



Add tables to use



Join the tables: drag and drop patron_record_id



Double-click the fields for output (or for limiting criteria)

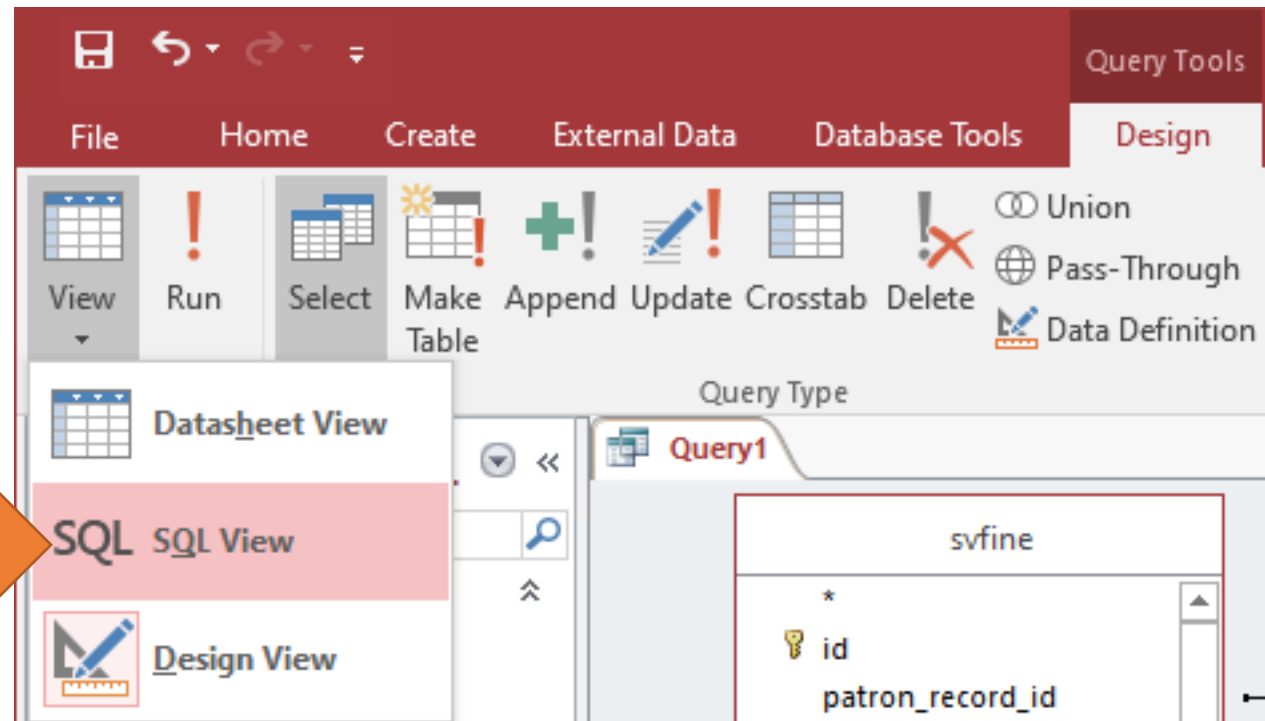
The screenshot shows the Microsoft Access Query Design view for a query named 'Query1'. The design grid shows a join between the 'svfine' and 'svpatron_record_address' tables on the 'patron_record_id' field. The 'svfine' table fields are: id, patron_record_id, assessed_gmt, invoice_num, item_charge_amt, processing_fee_amt, billing_fee_amt, charge_code, charge_location_code, paid_gmt, terminal_num, paid_amt, initials, created_code, is_print_bill, description, item_record_metadata_id, checkout_gmt, due_gmt, and returned_gmt. The 'svpatron_record_address' table fields are: id, patron_record_id, patron_record_address_type_id, display_order, addr1, addr2, addr3, village, city, region, postal_code, and country. The 'Field List' pane on the left shows the 'svfine' table selected. The 'Property Sheet' on the right shows the 'Query Properties' for 'Query1'. The 'General' tab is active, showing various query settings.

Fields for output and criteria

Field:	patron_record_id	item_charge_amt	processing_fee_amt	billing_fee_amt	postal_code	
Table:	svfine	svfine	svfine	svfine	svpatron_record_add	
Sort:						
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:						
or:						

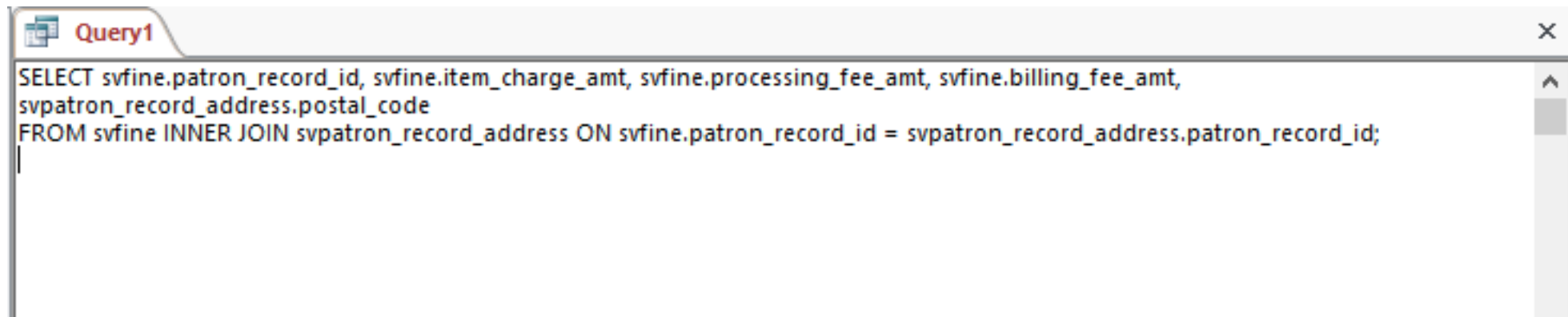


Switch to Design Menu – View - SQL View



View option – SQL View

Copy and use in pgAdmin



```
SELECT svfine.patron_record_id, svfine.item_charge_amt, svfine.processing_fee_amt, svfine.billing_fee_amt,  
svpatron_record_address.postal_code  
FROM svfine INNER JOIN svpatron_record_address ON svfine.patron_record_id = svpatron_record_address.patron_record_id;
```





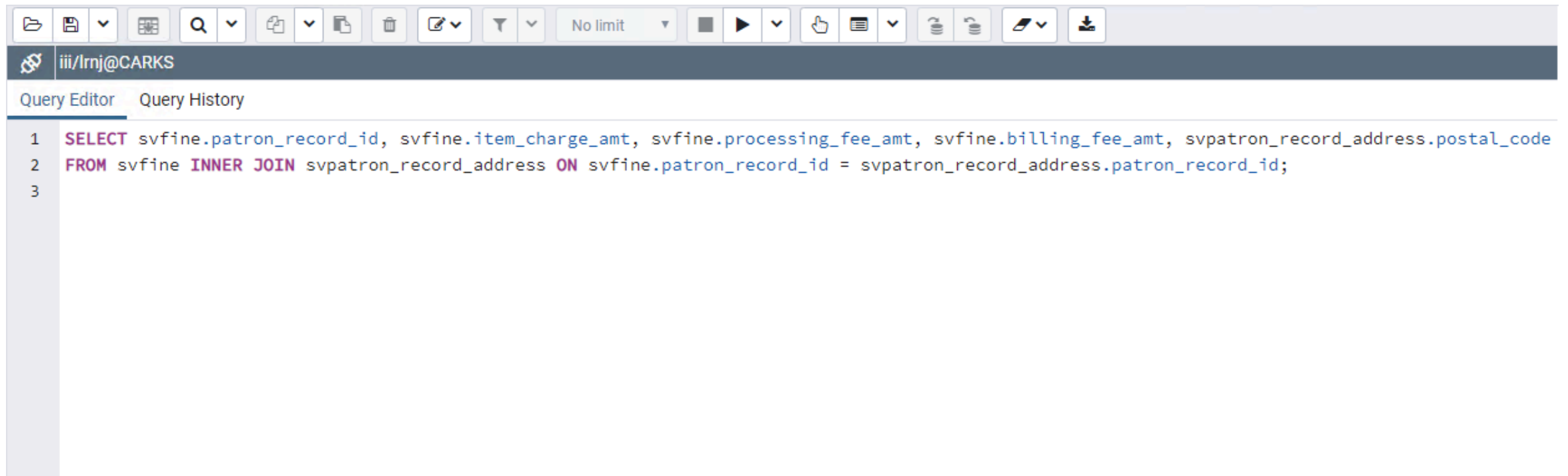
pgAdmin4: Running the query

Do this in pgAdmin

- Connect to your Sierra server
- Open the Query Tool
- Paste the SQL statement into the Query Tool panel
- Replace all sv with sierra_view.
 - Be sure to put the period (.) at the end of sierra_view.
- Run the query
- Export to csv for further work (or work with the data another way)
 - I often include record_id fields so I can look up a few examples in Sierra to confirm my query worked as intended
 - You will need to use the id2reckey() function to get usable Sierra record numbers



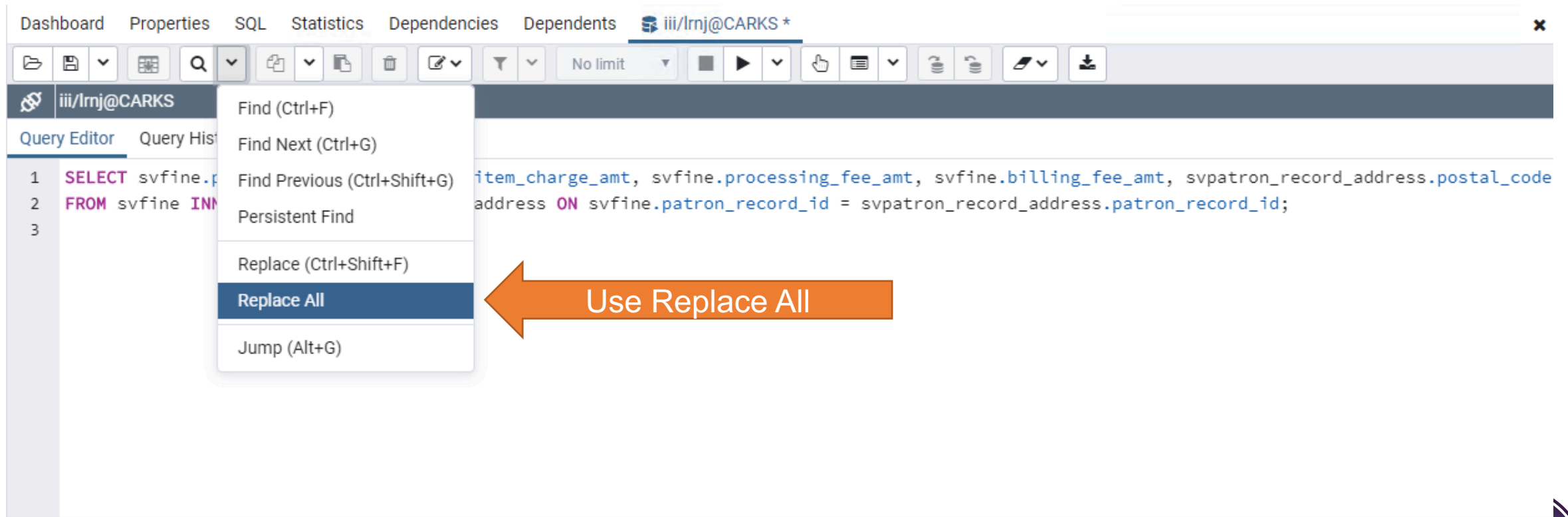
Paste query into Query Tool panel



The screenshot shows a web-based Query Tool interface. At the top is a toolbar with various icons for file operations, search, and execution. Below the toolbar is a dark header bar with the text "iii/lrnj@CARKS". Underneath is a tabbed interface with "Query Editor" and "Query History". The "Query Editor" tab is active, displaying a SQL query on a light blue background. The query is as follows:

```
1 SELECT svfine.patron_record_id, svfine.item_charge_amt, svfine.processing_fee_amt, svfine.billing_fee_amt, svpatron_record_address.postal_code
2 FROM svfine INNER JOIN svpatron_record_address ON svfine.patron_record_id = svpatron_record_address.patron_record_id;
3
```

Replace all sv with sierra_view.



Replace all sv with sierra_view.

Dashboard Properties SQL Statistics Dependencies Dependents **iii/lrnj@CARKS ***

iii/lrnj@CARKS

Query Editor Query History

Replace all: **sv**

2 **FROM** svfine **INNER JOIN** svpatron_record_address **ON** svfine.patron_record_id = svpatron_record_address.patron_record_id;

3

Replace all sv

Replace all sv with sierra_view.

Dashboard Properties SQL Statistics Dependencies Dependents iii/lnj@CARKS *

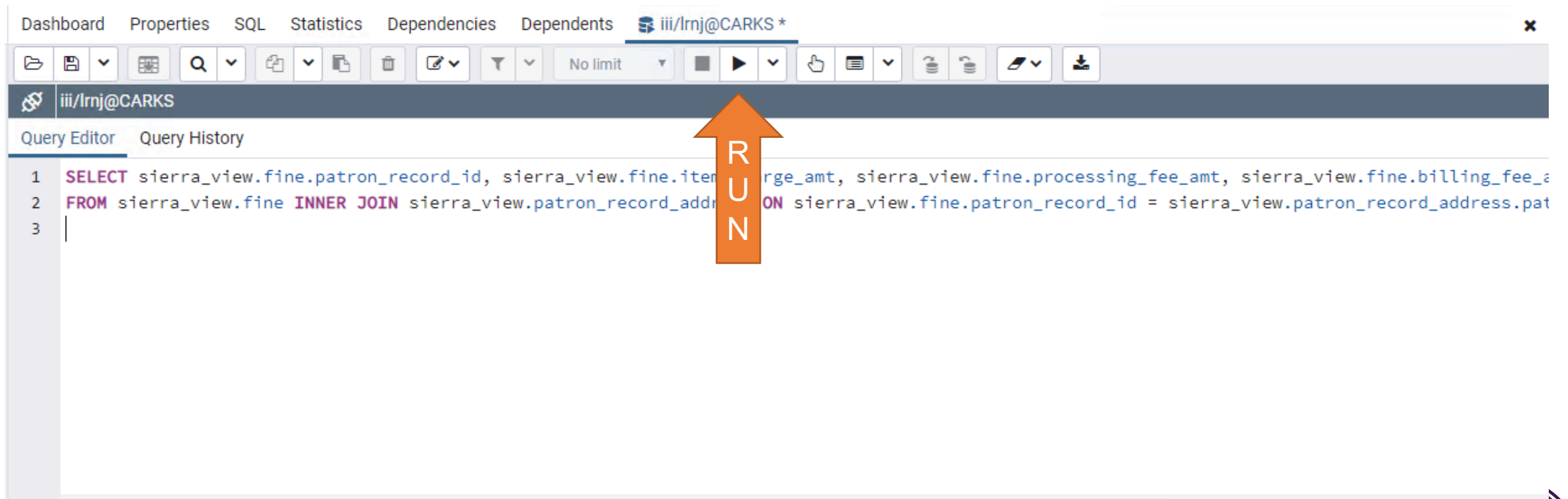
Query Editor Query History

With: sierra_view. **With sierra_view.**

```
2 FROM svfine INNER JOIN svpatron_record_address ON svfine.patron_record_id = svpatron_record_address.patron_record_id;  
3
```



Now the query is ready to run



The screenshot shows a database query editor interface. At the top, there are tabs for Dashboard, Properties, SQL, Statistics, Dependencies, and Dependents. The current tab is 'iii/Irnj@CARKS *'. Below the tabs is a toolbar with various icons, including a play button (execute). A large orange arrow with the word 'RUN' written vertically on it points to the play button. The main area is the 'Query Editor', which contains the following SQL query:

```
1 SELECT sierra_view.fine.patron_record_id, sierra_view.fine.item_id, sierra_view.fine.item_barcode, sierra_view.fine.item_charge_amt, sierra_view.fine.processing_fee_amt, sierra_view.fine.billing_fee_amt,
2 FROM sierra_view.fine INNER JOIN sierra_view.patron_record_address ON sierra_view.fine.patron_record_id = sierra_view.patron_record_address.patron_record_id
3
```



If all is well, you'll get data

Data Output		Explain	Messages	Notifications	
	patron_record_id bigint	item_charge_amt numeric (30,6)	processing_fee_amt numeric (30,6)	billing_fee_amt numeric (30,6)	postal_code character varying (100)
1	481037712386	0.700000	0.000000	0.000000	72204
2	481037712386	0.700000	0.000000	0.000000	72204
3	481037712386			0.000000	72204
4	481037700821	0.700000	0.000000	0.000000	72227
5	481037712386	0.400000	0.000000	0.000000	72204
6	481037629342	0.200000	0.000000	0.000000	72205
7	481037809265	2.200000	0.000000	0.000000	72113
8	481037669949	0.100000	0.000000	0.000000	72076
9	481037627162	0.100000	0.000000	0.000000	72204
10	481037820345	0.600000	0.000000	0.000000	72204

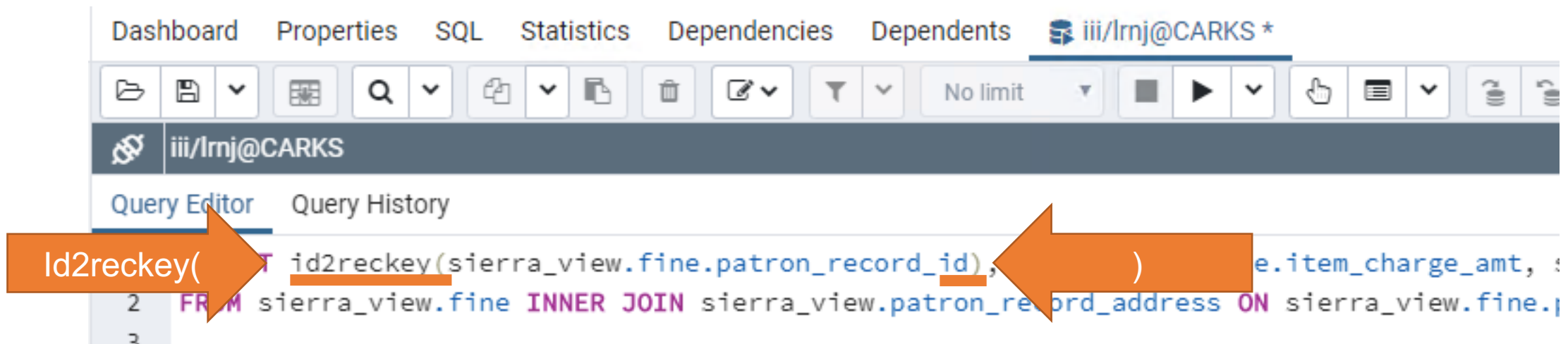
Record_ids aren't record numbers!
Use id2reckey() to convert

Record_ids aren't record numbers!
Use id2reckey() to convert



Add id2reckey() around record_id fields

- Find any record_id field in the SELECT statement
- Add id2reckey and an open parenthesis before it, and a close parenthesis after it



Id2reckey() converts record_ids to Sierra record numbers

Data Output		Explain	Messages	Notifications	
	id2reckey character varying	item_charge_amt numeric (30,6)	processing_fee_amt numeric (30,6)	billing_fee_amt numeric (30,6)	postal_code character varying (100)
1	p1375234	0.700000	0.000000	0.000000	72204
2	p1375234	0.700000	0.000000	0.000000	72204
3	p1375234			0.000000	72204
4	p1363669	0.700000	0.000000	0.000000	72227
5	p1375234	0.400000	0.000000	0.000000	72204
6	p1292190	0.200000	0.000000	0.000000	72205
7	p1472113	2.200000	0.000000	0.000000	72113
8	p1332797	0.100000	0.000000	0.000000	72076
9	p1290010	0.100000	0.000000	0.000000	72204
10	p1483103	0.600000	0.000000	0.000000	72204

Id2reckey() changed record_id to Sierra record number





Conditions and variable-length fields

Adding conditions

- Sometimes you don't need all of the data in a table, for example,
 - Only need certain types of fines
 - Only need fields/columns with specific data
 - Required when working with variable-length fields from `sierra_view.varfield_view`
- Two examples
 - Rows where a field/column has a specific code or codes
 - Finding rows where a field/column begins with specific text

Limit to specific values in fields/columns

- Use Criteria row in Access query editor
- Sierra_view.fine has a field/column for charge_code
 - This indicates the Sierra fine type
 - “Overdue” fines use codes 2, 4, and 6
- Add codes to the Criteria row for the field to match
 - Use single quotes around codes
- Use LIKE in Criteria row to match character strings
- Match text at beginning of a field
 - Use single quotes around text to match
 - LIKE ‘matchtext%’
 - % means “match one or more characters”

CODE	DEFINITION
1	manual charge
2	overdue
3	replacement
4	adjustment (OVERDUEX)
5	lost book
6	overdue renewed
7	rental
8	rental adjustment (RENTALX)
9	debit
a	notice
b	credit card
p	program (i.e., Program Registration)



Adding conditions: matching codes

Field:	patron record id	item charge amt	processing fee a	billing fee amt	postal code	description	charge code
Table:	svfine	svfine	svfine	svfine	svpatron record	svfine	svfine
Sort:							
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:							
or:							
							'2'
							'4'
							'6'

To match codes, enclose the code in single quotes

```
SELECT svfine.patron_record_id, svfine.item_charge_amt, svfine.processing_fee_amt,
svfine.billing_fee_amt, svpatron_record_address.postal_code, svfine.description, svfine.charge_code
FROM svfine INNER JOIN svpatron_record_address ON svfine.patron_record_id =
svpatron_record_address.patron_record_id
WHERE (((svfine.charge_code)='2')) OR (((svfine.charge_code)='4')) OR
(((svfine.charge_code)='6'));
```

Matching beginning of a field for text fields

- Needed to find all fines where the description began with 'Non-resident" which I shortened to 'Non%'
- The % will match any number of characters after the Non

[illegible]

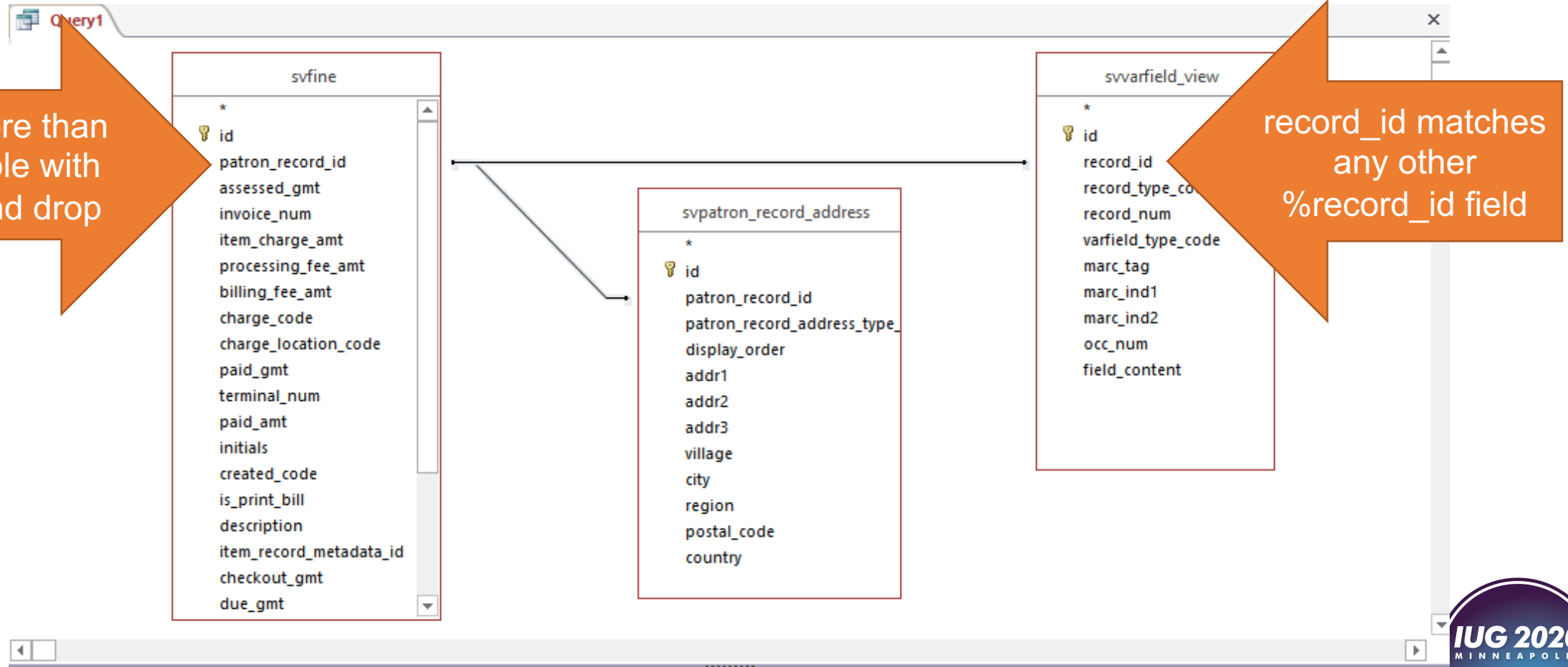
```
SELECT svfine.patron_record_id, svfine.item_charge_amt, svfine.processing_fee_amt,  
svfine.billing_fee_amt, svpatron_record_address.postal_code, svfine.description, svfine.charge_code  
FROM svfine INNER JOIN svpatron_record_address ON svfine.patron_record_id =  
svpatron_record_address.patron_record_id  
WHERE (((svfine.description) Like 'Non%'));
```

Adding variable-length fields with sierra_view.varfield_view

- All variable-length fields are in one table: sierra_view.varfield_view
- record_id
 - Matches record_id fields/columns from other tables like patron_record_id
- record_type_code
 - Use to limit to a specific record type
 - Example: p = patron
- varfield_type_code
 - Use to limit to specific variable-length field(s)
 - Use field group tag
 - Example: z is the field group tag for email address in our patron record
- field_content
 - This contains the data from the variable-length field



Adding email address from sierra_view.varfield_view



Limiting to patron email fields from sierra_view.varfield_view

- Limit to record_type_code = p (patron)
- Limit to varfield_type_code = z (patron email address field group tag in our Sierra database)
- Uncheck the “show” checkbox for any fields you don’t need to see in the output
- Don’t forget to add field_content or you won’t see any data!

Field:	item_charge_amt	processing_fee_amt	billing_fee_amt	postal_code	record_type_code	varfield_type_code	field_content	description
Table:	svfine	svfine	svfine	svpatron_record_ac	svvarfield_view	svvarfield_view	svvarfield_view	svfine
Sort:								
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:					'p'	'z'		Like 'Non%'
or:								



Output fines with email addresses from varfield_view

Data Output Explain Messages Notifications

	item_charge_amt numeric (30,6)	processing_fee_amt numeric (30,6)	billing_fee_amt numeric (30,6)	postal_code character varying (100)	field_content character varying (20001)	description character varying (100)
1	54.000000	0.000000	0.000000	72106	[REDACTED]@yahoo.com	Non-Resident 1 YEAR
2	54.000000	0.000000	0.000000	72114	[REDACTED].com	Non-Resident 1 YEAR
3	60.000000	0.000000	0.000000	71901	[REDACTED]ail.com	NonResident Card Fee
4	60.000000	0.000000	0.000000	72143	[REDACTED]o.com	NonResident Card Fee
5	60.000000	0.000000	0.000000	49460	[REDACTED]e7@gmail.com	NonResident Card Fee
6	60.000000	0.000000	0.000000	72032	[REDACTED]@gmail.com	NonResident Card Fee
7	60.000000	0.000000	0.000000	71909	[REDACTED]sbcglobal.net	NonResident Card Fee
8	60.000000	0.000000	0.000000	72632	[REDACTED]om	NonResident Card Fee
9	60.000000	0.000000	0.000000	71901	[REDACTED]gmail.com	NonResident Card Fee



#IUG2020





The wishlist

How could this be improved?

- No more manual sierra_view reconstruction
 - Automatically import table structure into Access by connecting Access to Sierra directly
- No more copy and paste
 - No need to use two applications
 - Run queries directly from Access
- An alternative to Access
 - Free would be good
 - Mac OS support would be good

Thank you!

Questions?



#IUG2020

