SELECT * FROM

"Top Ten Things I Wish I Had Known About SierraSQL"

Jeremy Goldstein: Minuteman Library Network

Ray Voelker: The Public Library of Cincinnati and Hamilton County

IUG Connections

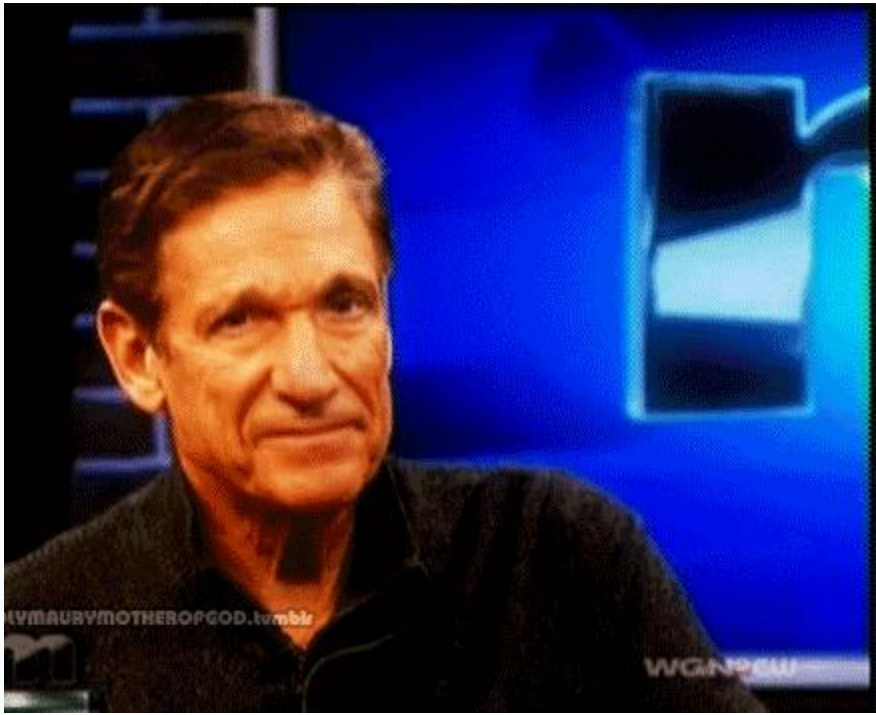
Meet Jeremy and Ray...and Rufus and Audrey

Things I Wish I Had Known About SierraSQL

1. Maury Povich says, "record_metadata ... you *are* the father!"

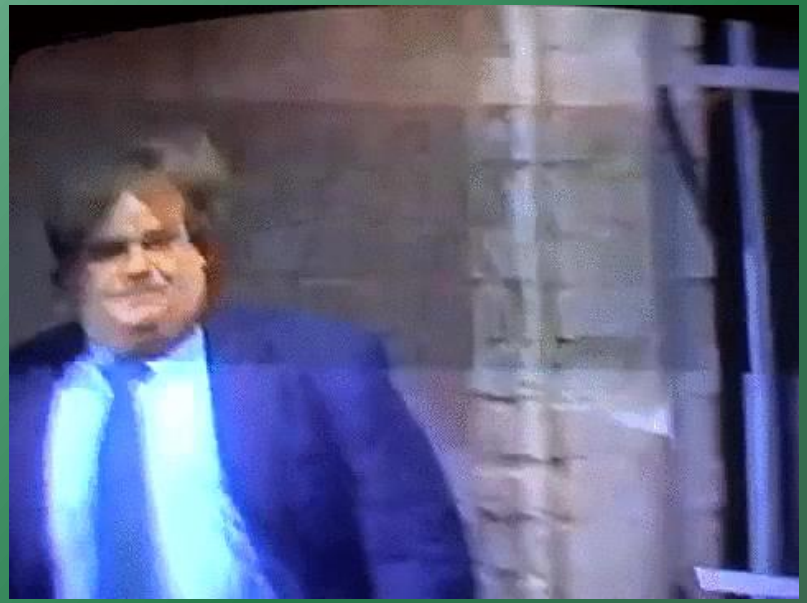
1. Maury Povich says, "record_metadata ... you *are* the father!"

```
CREATE TABLE record_metadata (
        id BIGINT,
        record_type_code TEXT,
        record_num INTEGER,
        creation_date_gmt DATETIME,
        deletion_date_gmt DATE,
        campus_code TEXT,
        agency_code_num INTEGER,
        num_revisions INTEGER,
        record_last_updated_gmt DATETIME,
        previous_last_updated_gmt DATETIME
);
```



Things I Wish I Had Known About SierraSQL

2. reckey? damn near killed 'em!



ID2RECKEY()

Unique Sierra function to convert id fields for various record types to the record numbers used by Sierra

Does not include the check digit

Works for all record types included in the record_metadata table…with one big caveat

```
1 SELECT
2 id,
3 id2reckey(id)||'a' AS record_number
4 FROM
5 sierra_view.order_record
6 LIMIT 100
```

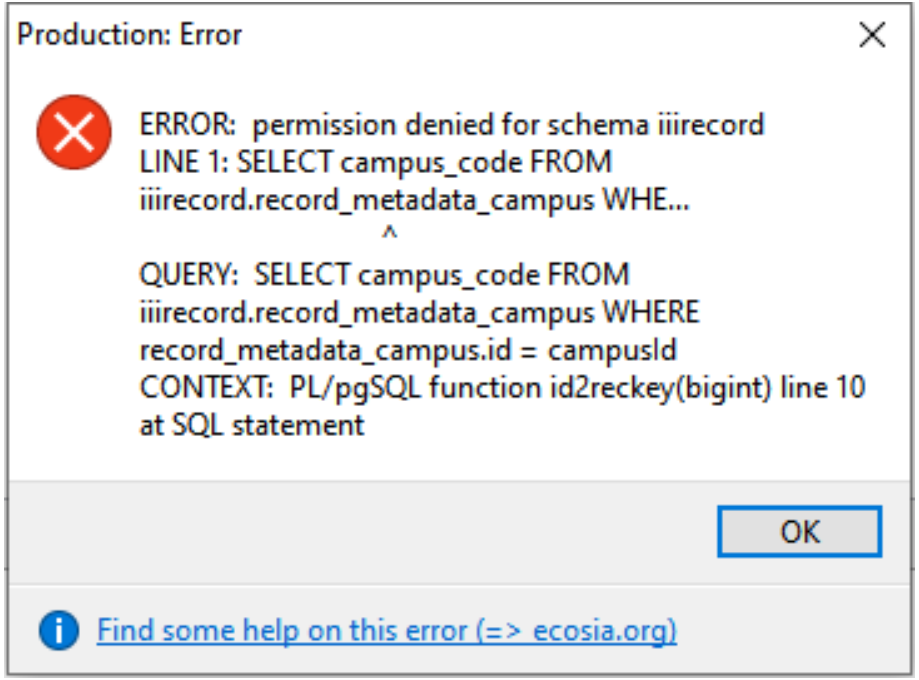
Result #1 (100r × 2c)

id	record_number
476751486385	o10116529a
476751486381	o10116525a
476748767383	o7397527a
476747207535	o5837679a
476747208474	o5838618a
476751489204	o10119348a
476747885708	o6515852a
476749919022	o8549166a
476749919023	o8549167a
476751489177	o10119321a
476748780399	o7410543a
476751489196	o10119340a
476749919043	o8549187a
476749919000	o8549144a
476749919040	o8549184a
476747885790	o6515934a
476747885732	o6515876a

ID2RECKEY() Error

Using this function on record types that can include Virtual or INN Reach records yields the following error

Note the references to a campus_code field



campus_code

Id2reckey works by looking up the id in the record_metadata table and returning both the record_type_code and record_num fields

Table also includes a campus_code field to identify virtual/INNReach records (marked as 'ncip' within Minuteman)

These entries have shorter record numbers than those associated with 'real' records

```sql
SELECT
id,
record_type_code,
record_num,
campus_code
FROM
sierra_view.record_metadata rm
WHERE
campus_code != ''
LIMIT 100
```

record_metadata (100r × 4c)

id	record_type_code	record_num	campus_code
281895883623801	b	118137	ncip
281895883644686	b	139022	ncip
281895883648370	b	142706	ncip
281895883633159	b	127495	ncip
281925948381912	i	105176	ncip
281925948384898	i	108162	ncip
281895883610961	b	105297	ncip
281895883610962	b	105298	ncip
281925948408397	i	131661	ncip

Workaround

Option 1: Don't use id2reckey

```
2  b.id,
3  rm.record_type_code||rm.record_num||'a' AS record_number
4  FROM
5  sierra_view.bib_record b
6  JOIN
7  sierra_view.record_metadata rm
8  ON b.id = rm.id
9  LIMIT 100
```

Result #1 (100r × 2c)

id	record_number
420908897333	b2102325a
420907935051	b1140043a
420908361223	b1566215a
420909636695	b2841687a
420909788414	b2993406a
420909552996	b2757988a
420908145298	b1350290a
420908474228	b1679220a
420909915583	b3120575a
420908250293	b1455285a
420910388001	b3592993a
420910288954	b3493946a
420908910630	b2115622a

Option 2: Filter on campus_code

```
2  b.id,
3  id2reckey(b.id)||'a' AS record_number
4  FROM
5  sierra_view.bib_record b
6  JOIN
7  sierra_view.record_metadata rm
8  ON b.id = rm.id
9  WHERE rm.campus_code = ''
10 LIMIT 100
```

Result #1 (100r × 2c)

id	record_number
420908897333	b2102325a
420907935051	b1140043a
420908361223	b1566215a
420909636695	b2841687a
420909788414	b2993406a
420909552996	b2757988a
420908145298	b1350290a
420908474228	b1679220a
420909915583	b3120575a
420908250293	b1455285a
420910388001	b3592993a
420910288954	b3493946a
420908910630	b2115622a

Things I Wish I Had Known About SierraSQL

3. NULL Relationship Status : Complicated”

NULL Relationship Status : “Complicated”

Consider This Query:

howtosql.cincy.pl/iug2021/sierra_view

```
select
  r.id,
  r.record_type_code,
  r.record_num,
  r.creation_date_gmt,
  r.deletion_date_gmt
from
  record_metadata as r
  join bib_record as b on b.record_id = r.id
where
  r.record_type_code = 'b'
  and r.record_num in (1000001, 1000345, 1001270, 1001323)
```

NULL Relationship Status : “Complicated”

Consider This Query:

howtosql.cincy.pl/iug2021/sierra_view

```
select
  r.id,
  r.record_type_code,
  r.record_num,
  r.creation_date_gmt,
  r.deletion_date_gmt
from
  record_metadata as r
  left join bib_record as b on b.record_id = r.id
where
  r.record_type_code = 'b'
  and r.record_num in (1000001, 1000345, 1001270, 1001323)
```

NULL Relationship Status : “Complicated”

Consider This Query: howtosql.cincy.pl/iug2021/sierra_view …

```sql
select
  r.record_type_code,
  r.record_num,
  p.best_title,
  p.publish_year,
  p.best_title || p.publish_year as title_and_pub_year,
  p.best_title || coalesce(p.publish_year, '') as title_and_pub_year_correct
from
  record_metadata as r
  join bib_record_property as p on p.bib_record_id = r.id
where
  r.record_type_code = 'b'
  and r.campus_code = ''
  and r.record_num in (1000001, 1021602, 3021602, 1403467)
```

Format SQL Run SQL

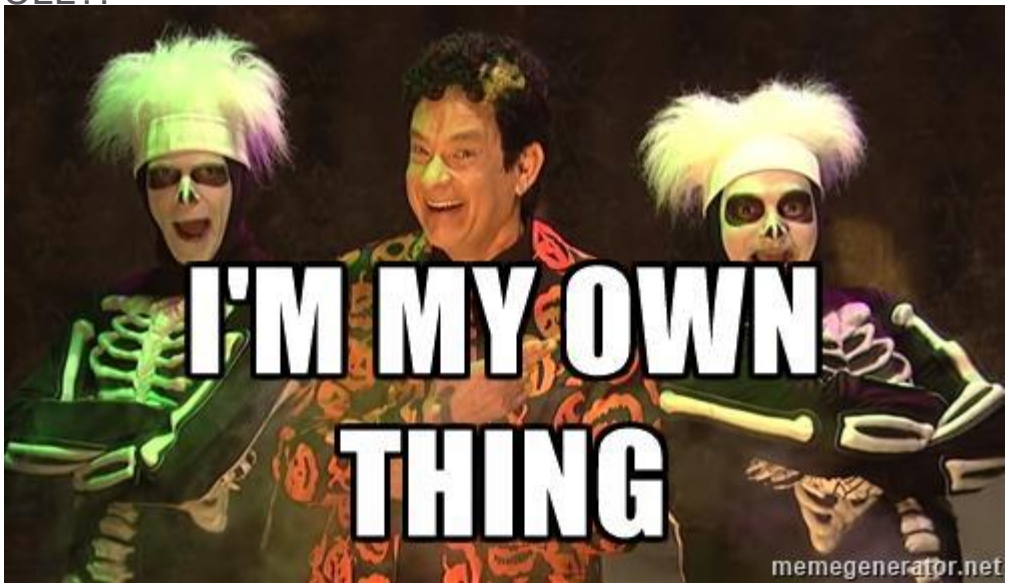
This data as json, CSV

record_type_code	record_num	best_title	publish_year	title_and_pub_year	title_and_pub_year_correct
b	1000001	Water monsters : opposing viewpoints	1991	Water monsters : opposing viewpoints1991	Water monsters : opposing viewpoints1991
b	1021602	TEAPOTS			TEAPOTS
b	1403467	The French revolution; a study in democracy,			The French revolution; a study in democracy,

NULL Relationship Status : “Complicated”

Consider This Query: howtosql.cincy.pl/iug2021/sierra_view …

What are NULL?!




```sql
with data as (
  select
    1 as val
  union all
  select
    2 as val
  union all
  select
    3 as val
  union all
  select
    null as val
)
select
  0 + 1 + 2,
  NULL + 1 + 2,
  count(*),
  sum(val)
from
  data
```

Format SQL | Run SQL

This data as json, CSV

0 + 1 + 2	NULL + 1 + 2	count(*)	sum(val)
3		4	6

Things I Wish I Had Known About SierraSQL

4. Var out, man!

Variable-length Fields

Record

Not checked out
i180205572 Last Updated: 09-15-2021 Created: 09-13-2021 Revisions: 2

ICode 1 (SCAT)	216	Inventory Date	- -
Item Code 2	- -	Checkin Location	0
Item Type	150 J Book	No. of Renewals	0
Price	$27.00	No. of Notices	0
Checkout Date	- - :	Notice Date	- -
Checkout Location	0	Item Use 3	0
Due Date	- -	Recall Date	- -
Patron No.	0	Total Checkouts	0
Last Patron	0	Total Renewals	0
Last Checkin	- - :	Last Checkout Date	- - :

Call No.	092		j636.73 M
Barcode			31619008271793
Staff Initials			LEX, br

b29688681 Last Updated: 09-13-2021 Created: 01-11-2012 Revisions: 18

Language	eng English	Cat. Date	02-08-2012	Bib Code 3	- -
Skip	0	Bib Level	m MONOGRAPH	Country	mnu Minnesota
Location	multi	Material Type	a BOOK		

MARC Leader	##### c a m 2 2 ##### a 4 5 0 0
LOCATIONS	lexj ,wsnj ,wobj

Bib Utility No.	001			704908500
Misc.	003			OCoLC
Misc.	005			20120208085144.0
Misc.	008			110301 s 2012 mnu a j 0 0 1 0 eng
LCCN	010			2011009105
Misc.	019			761089946
Standard No.	020			9781617149900
Standard No.	020			161714990X
Misc.	035			(OCoLC)704908500|z(OCoLC)761089946
Misc.	040			DLC|beng|cDLC|dPNX|dQBX|dSINLB|dMLN
Misc.	042			pcc
Misc.	049			MLNM
Classification No.	050	0	0	SF429.C35|bM38 2012
Classification No.	082	0	0	636.73|222
Classification No.	082	1	4	636.7|214
Author	100	1		Mattern, Joanne,|d1963-
Title	245	1	0	Catahoula leopard dogs /|cJoanne Mattern.
Publication Info.	260			Edina, Minn. :|bABDO Pub. Co.,|cc2012.
Description	300			24 p. :|bcol. ill. ;|c21 cm.
Series	490	1		Dogs. Set IX
Series	490	1		Checkerboard animal library
Note	500			Includes index.
Note	520			Describes the physical and behavior characteristics of the Catahoula leopard dog, and offers advice on training, breeding, feeding, grooming, and health care.

Fields exists for all record types and in SQL exist in a single table….well 3 but we'll get to that

Varfield

Entries from the varfield table

Contains all fields from all record types

Here limited to varfield tag 'm'

Most are bib classification numbers

Highlighted to the right are a pair of patron and item messages

Note there is no field in the table to designate the record type

```
1  SELECT *
2
3  FROM
4  sierra_view.varfield v
5  WHERE
6  v.varfield_type_code = 'm'
7
8  LIMIT 500
```

varfield (500r × 8c)

id	record_id	varfield_type_code	marc_tag	marc_ind1	marc_ind2	occ_num	field_content
177508904	481038196639	m	(NULL)			0	no receipt
177508159	450989421908	m	(NULL)			0	Paper damaged at the bottom
177508767	420910827313	m	050	0	0	0	|aN89.2.R67|bG46 2021
177508856	420910764051	m	050	0	0	0	|aHV1451|b.W457 2021
164580936	420910764051	m	082	0		1	|a362.61|223
177507781	420910856640	m	082	0	0	0	|a551.55209764
177507799	420910856641	m	082	0	0	0	|aB
177508553	420908529632	m	082	0	4	0	|a811.54|bZ8
177508176	450989743157	m	(NULL)			0	2 DVDS
177508219	420910265964	m	050	1	4	1	|aDS201.4|b.R47 1999
177508964	420910773806	m	050	1	4	0	|aPZ7.J429877|bTo 2021

Varfield

Join to record_metadata or a record_table to properly limit entries

```sql
SELECT v.*

FROM
sierra_view.varfield v
JOIN
sierra_view.record_metadata rm
ON
v.record_id = rm.id AND rm.record_type_code = 'b'
WHERE
v.varfield_type_code = 'm'

LIMIT 500
```

varfield (500r × 8c)

id	record_id	varfield_type_code	marc_tag	marc_ind1	marc_ind2	occ_num	field_content
177566123	420910857006	m	050	0	0	0	|aPT8175.D5|bT76 2022
177566124	420910857006	m	082	0	0	1	|a839.813/72|223/eng/20211217
177566146	420910857007	m	050			0	|aPS
177566147	420910857007	m	082			1	|a813
177566102	420910857005	m	050	0	0	0	|aPS3608.A89348|bB49 2021
177566103	420910857005	m	082	0	0	1	|a813/.6|223
177566195	420910839661	m	050	0	0	0	|aPZ7.S6566|bLe 2022
177566196	420910839661	m	082	0	4	1	|a813.6|aFic|223
177566357	420910857013	m	082	0	0	1	|a808/.02023|223
177566356	420910857013	m	050	0	0	0	|aPN151|b.B625 2022
177566324	420910857011	m	050	0	0	0	|aPN842|b.S76 2021
177566325	420910857011	m	082	0	0	1	|a809/.88924|223
177566485	420910857017	m	050	1	4	0	|aHQ755.8|b.R6756 2021
177566486	420910857017	m	082	0	4	1	|a649/.1|223/eng/20210922

Varfield vs. Subfield

```sql
SELECT v.*

FROM
sierra_view.varfield v
JOIN
sierra_view.record_metadata rm
ON
v.record_id = rm.id AND rm.record_type_code = 'b'
WHERE
v.varfield_type_code = 'm'

LIMIT 500
```

varfield (500r × 8c)

id	record_id	varfield_type_code	marc_tag	marc_ind1	marc_ind2	occ_num	field_content
177566123	420910857006	m	050	0	0	0	|aPT8175.D5|bT76 2022
177566124	420910857006	m	082	0	0	1	|a839.813/72|223/eng/20211217
177566146	420910857007	m	050			0	|aPS
177566147	420910857007	m	082			1	|a813
177566102	420910857005	m	050	0	0	0	|aPS3608.A89348|bB49 2021
177566103	420910857005	m	082	0	0	1	|a813/.6|223
177566195	420910839661	m	050	0	0	0	|aPZ7.S6566|bLe 2022
177566196	420910839661	m	082	0	4	1	|a813.6|aFic|223
177566357	420910857013	m	082	0	0	1	|a808/.02023|223
177566356	420910857013	m	050	0	0	0	|aPN151|b.B625 2022
177566324	420910857011	m	050	0	0	0	|aPN842|b.S76 2021
177566325	420910857011	m	082	0	0	1	|a809/.88924|223
177566485	420910857017	m	050	1	4	0	|aHQ755.8|b.R6756 2021
177566486	420910857017	m	082	0	4	1	|a649/.1|223/eng/20210922

```sql
SELECT s.*

FROM
sierra_view.subfield s
JOIN
sierra_view.record_metadata rm
ON
s.record_id = rm.id AND rm.record_type_code = 'b'
WHERE
s.field_type_code = 'm'

LIMIT 500
```

subfield (500r × 10c)

record_id	varfield_id	field_type_code	marc_tag	marc_ind1	marc_ind2	occ_num	display_order	tag	content
420909932312	85	m	050		4	0	0	a	PN1992.77
420909932312	85	m	050		4	0	1	b	.H37 2014
420909932312	86	m	082	0	4	1	0	a	741.45/75
420909932312	86	m	082	0	4	1	1	2	23
420909932321	135	m	050		4	0	0	a	PN1997.2
420909932321	135	m	050		4	0	1	b	.M9 2013
420909932321	136	m	082	0	4	1	0	a	791.43/72
420909932321	136	m	082	0	4	1	1	2	23
420909932642	179	m	050	1	4	0	0	a	PN1997.2
420909932642	179	m	050	1	4	0	1	b	.Y6864 2014
420909932642	180	m	082	0	4	1	0	a	791.4372
420909932642	180	m	082	0	4	1	1	2	23
420909933322	271	m	050	1	4	0	0	a	PN1997.2
420909933322	271	m	050	1	4	0	1	b	.O5844 2014

Note the inconsistent field names:

- varfield_type_code / field_type_code
- field_content / content)

Phrase Entry

Indexed portion of varfield table

Text is normalized, capitalization and special characters removed

Smaller table, faster to parse

Permuted entries included

Lacks MARC field, indicators and delimiters

```sql
SELECT p.*

FROM
sierra_view.phrase_entry p
JOIN
sierra_view.record_metadata rm
ON
p.record_id = rm.id AND rm.record_type_code = 'b'
WHERE
p.varfield_type_code = 'd'

LIMIT 500
```

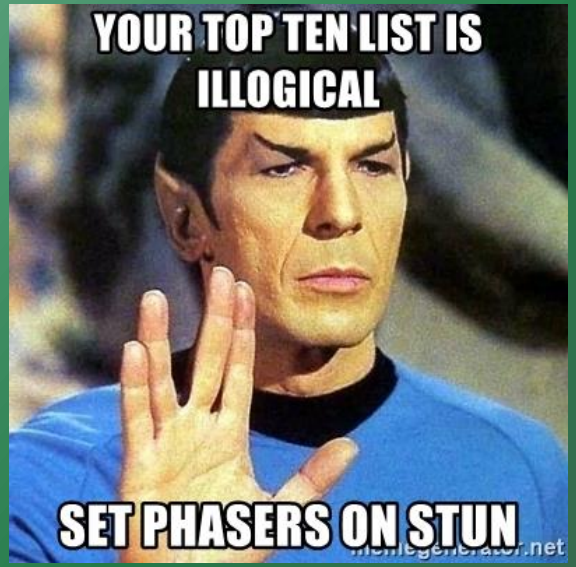
Filter ...
- Columns
- SQL functions
- SQL keywords
- Snippets
- Query history
- Query profile
- Bind parameters

phrase_entry (500r × 17c)

id	record_id	index_tag	varfield_type_code	occurrence	is_permuted	type2	index_entry	insert_title
55226343	420908384256	d	d	1	true	102	sources farm tractors united states history 20th century	tfarm tractors 1916 1925
55226367	420908384266	d	d	2	false	112	germany history 1933 1945	tplotting hitlers death the story c
55226378	420908384286	d	d	0	true	119	alexander i 1801 1825 russia history fiction	twar and peace
55226379	420908384286	d	d	1	true	119	campaigns napoleonic wars 1800 1815 russia fiction	twar and peace
55226380	420908384286	d	d	1	false	119	napoleonic wars 1800 1815 campaigns russia fiction	twar and peace
55226381	420908384286	d	d	0	false	119	russia history alexander i 1801 1825 fiction	twar and peace
55226299	420908384244	d	d	0	false	97	alexander the great 356 b c 323 b c	talexander the great man of acti
55226318	420908384254	d	d	1	false	32	zoo animals pictorial works	t 1 2 3 to the zoo a
55226319	420908384254	d	d	0	false	32	counting juvenile literature	t 1 2 3 to the zoo a
55226373	420908384279	d	d	0	false	108	bemelmans ludwig 1898 1962 criticism and interpret...	tludwig bemelmans
55226374	420908384279	d	d	1	false	108	children books and reading united states history 20th ...	tludwig bemelmans

Things I Wish I Had Known About SierraSQL

5. “Set phrase entries to ‘stun’, er … I mean ‘index’”



5. “Set phrase entries to ‘stun’, er … I mean ‘index’”

For some queries–like this query below that finds the item record that contains a specified item barcode in the item record varfield field content–the amount of time required to complete can be quite awful

```
SELECT
*
FROM
sierra_view.varfield AS v
WHERE
v.varfield_type_code = 'b'
AND v.field_content = 'A000052469475'
-- 1 row(s) fetched - 14.767s
```

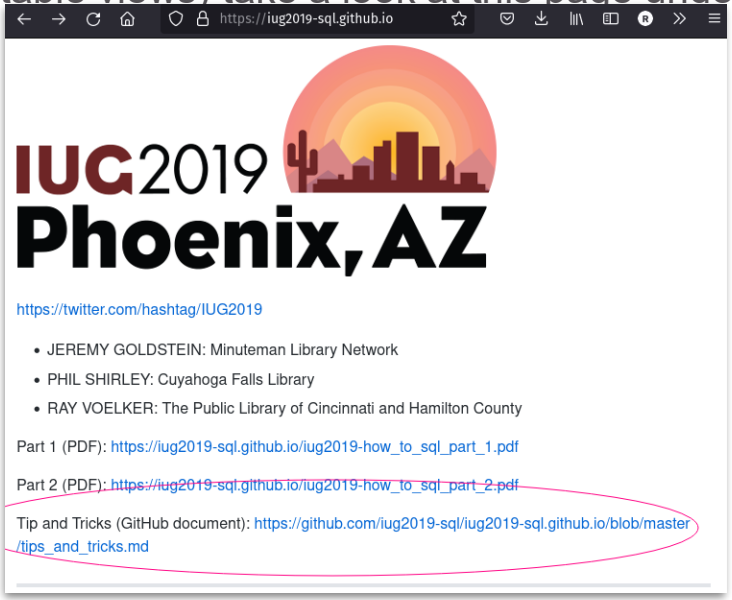
5. “Set phrase entries to ‘stun’, er … I mean ‘index’”

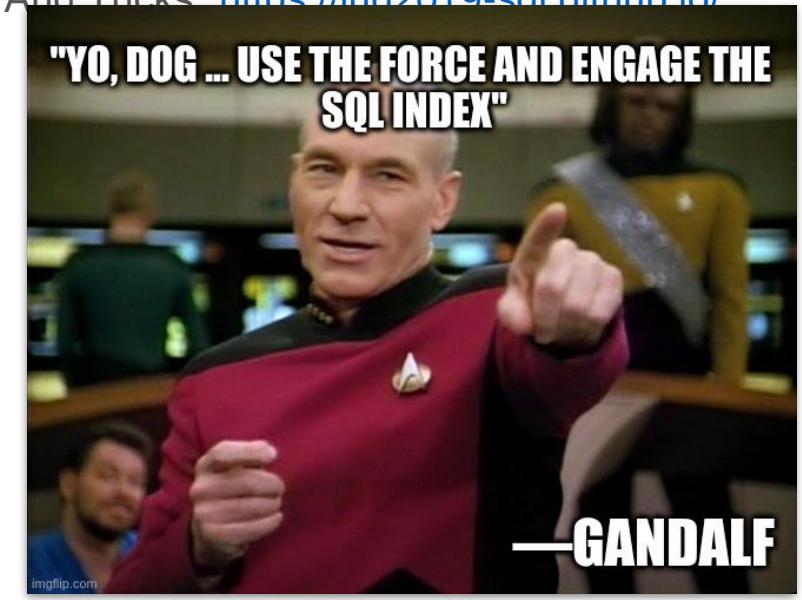
For some queries (like finding the varfield and item record that contains an item barcode) the amount of time required to complete can be quite awful

```
SELECT
*
FROM
sierra_view.phrase_entry AS e
WHERE
e.index_tag || e.index_entry = 'b'
  || lower('A000052469475')
--1 row(s) fetched - 40ms
```

5. “Set phrase entries to ‘stun’, er … I mean ‘index’”

For more information on how to figure out which indexes exist for the tables (via corresponding table views) take a look at this page under “Tips And Tricks” https://iug2019-sql.github.io/





6. A View to A Void

_View Views

Combines fields from record_metadata, record and record_property tables into a single table

```sql
SELECT
rm.record_type_code||rm.record_num||'a' AS record_num,
ip.barcode,
i.location_code

FROM
sierra_view.item_record i
JOIN
sierra_view.record_metadata rm
ON
i.id = rm.id
JOIN
sierra_view.item_record_property ip
ON
i.id = ip.item_record_id
LIMIT 100
```

Result #1 (100r × 3c)

record_num	barcode	location_code
i16774356a	32405005251429	nora
i16774360a	34867007642285	wlmj
i17844924a	31189933335972	camj
i13600385a	31712901127830	br2n
i15129246a	31189931060416	ca6a
i16352229a	31189931834018	camj
i15604662a	34861005141259	bedn

```sql
SELECT
i.record_type_code||i.record_num||'a' AS record_num,
i.barcode,
i.location_code

FROM
sierra_view.item_view i
LIMIT 100
```

Result #1 (100r × 3c)

record_num	barcode	location_code
i1000081a	34867001672098	wlmr
i1000113a	31213004710598	fplh
i1000120a	33018000940596	wwdr
i1000122a	32405001336885	nors
i1000162a	34862002740549	blms
i10795347a	31712012915420	brka
i1000173a	34868000851824	wath
i15861146a	34866010534703	sudn
i12143539a	30308000923673	asha
i1000235a	31155001653885	somr
i9888743a	35957001198522	mayj
i1000317a	32211000423835	actr
i1000323a	34868000272757	wath

Convenience Vs Performance

Convenience comes with a cost in performance time

```sql
SELECT
SUBSTRING(i.location_code,1,3),
COUNT(i.id)

FROM
sierra_view.item_record i

GROUP BY 1
ORDER BY 1
```

- (73r × 2c)

substring	count
	12
ac2	6752
act	168654
ar2	29183
arl	174919
ash	74206
bed	106449
blm	131033
br2	75927
br3	35584
brk	159016

Time to Run:
1 Second

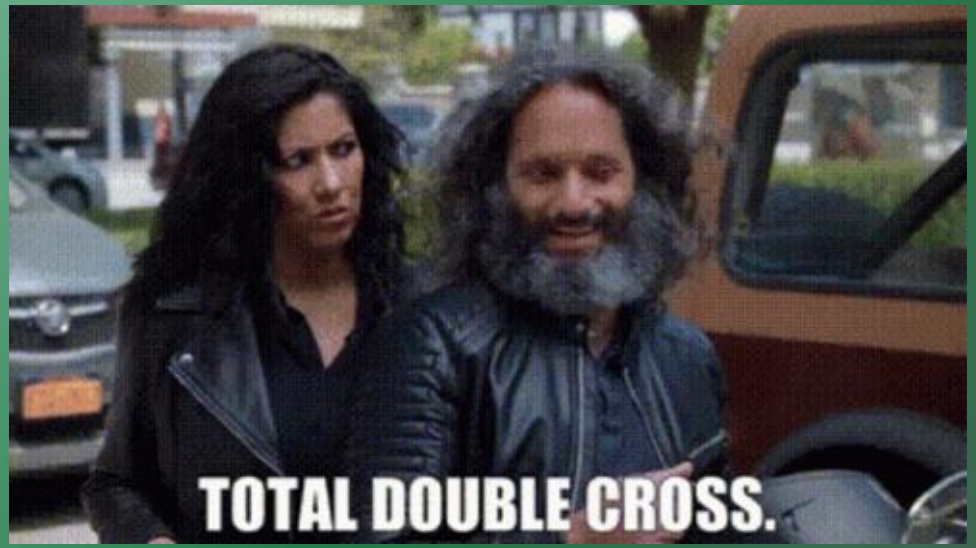
```sql
SELECT
SUBSTRING(i.location_code,1,3),
COUNT(i.id)

FROM
sierra_view.item_view i

GROUP BY 1
ORDER BY 1
```

- (72r × 2c)

substring	count
	12
ac2	6752
act	168654
ar2	29183
arl	174919
ash	74206
bed	106449
blm	131033
br2	75927
br3	35584
brk	159016

Time to Run:
19.359
Seconds

7. My JOIN double crossed me!



7. My JOIN double crossed me!

When using the JOIN clause, you may end up with situations where you're introducing more rows of output than you would expect–specifically in one-to-many types of relationships.

Consider This Query: https://howtosql.cincy.pl/iug2021/sierra_view ...

```sql
select
  r.record_type_code,
  r.record_num,
  v.varfield_type_code,
  v.field_content
from
  record_metadata as r
  join varfield as v on v.record_id = r.id
where
  r.campus_code = ''
  and r.record_type_code = 'i'
  and r.record_num in (1000317, 7043125, 9661321)
  and v.varfield_type_code = 'b'
```

How many rows do we expect in the results?

7. My JOIN double crossed me!

When using the JOIN clause, you may end up with situations where you're introducing more rows of output than you would expect–specifically in one-to-many types of relationships.

Consider This Query: https://howtosql.cincy.pl/iug2021/sierra_view ...

```
1  select
2    r.record_type_code,
3    r.record_num
```

record_type_code	record_num	varfield_type_code	field_content
i	1000317	b	1041621948010
i	7043125	b	A000029114253
i	7043125	b	A000029115458
i	9661321	b	f
i	9661321	b	A000058308248

```
11   and r.record_type_code = 'i'
12   and r.record_num in (1000317, 7043125, 9661321)
13   and v.varfield_type_code = 'b'
```

How many rows do we expect in the results?

7. My JOIN double crossed me!

To avoid producing extra rows, *one* way is to use a "**subquery**"

(note that you can only return a single value when using a subquery in the SELECT clause)

record_type_code	record_num	varfield_type_code	field_content
i	1000317	b	1041621948010
i	7043125	b	A000029115458
i	9661321	b	A000058308248

```
select
  r.record_type_code,
  r.record_num,
  'b' as varfield_type_code,
  (
    -- just grab whatever the first occurrence is
    select
      v.field_content
    from
      varfield as v
    where
      v.varfield_type_code = 'b'
      and v.record_id = r.id
    order by
      v.occ_num
    limit
      1
  ) as field_content
from
  record_metadata as r
where
  r.campus_code = ''
  and r.record_type_code = 'i'
  and r.record_num in (1000317, 7043125, 9661321)
```

8. Putting the fun in funds

Where to find funds

The order_record table does not include fund data

For that you must look at the order_record_cmf table

sierra_view.order_record_cmf

id	order_record_id	display_order	fund_code	copies	location_code
2650093	476759621246	0	00000	1	ca4a
2650092	476759621245	0	00000	1	ca4a
2650091	476759621244	0	00000	1	ca4a
2646203	476748804344	0	00000	1	wata
2645231	476748374051	0	00000	1	ntnn
2645230	476748374050	0	00000	1	ntnn
2645229	476748374049	0	00000	2	ntnn
2642171	476752507872	0	00000	1	conal
2638350	476748373947	0	00000	1	ntnj
2634365	476748373814	0	00000	1	ntnj
2634293	476748373811	0	00000	1	ntnn
2634290	476748373808	0	00000	1	ntnn
2634288	476748373806	0	00000	1	ntnn
2634287	476748373805	0	00000	1	ntnn
2595588	476749174144	0	00000	1	arln

Name: order_record

Definer:

SQL security: Definer

Algorithm
- UNDEFINED
- MERGE
- TEMPTABLE

Check option for updates
- None
- CASCADED
- LOCAL

Select statement:

```
SELECT order_record.id,
    order_record.record_id,
    order_record.accounting_unit_code_num,
    order_record.acq_type_code,
    order_record.catalog_date AS catalog_date_gmt,
    order_record.claim_action_code,
    order_record.code1 AS ocode1,
    order_record.code2 AS ocode2,
    order_record.code3 AS ocode3,
    order_record.code4 AS ocode4,
    order_record.estimated_price,
    order_record.material_type_code AS form_code,
    order_record.order_date AS order_date_gmt,
    order_record.order_note_code,
    order_record.order_type_code,
    order_record.receiving_action_code,
    order_record.received_date AS received_date_gmt,
    order_record.receiving_location_code,
    order_record.billing_location_code,
    order_record.order_status_code,
    order_record.temporary_location_code,
    order_record.vendor_record_code,
    order_record.language_code,
    order_record.blanket_purchase_order_num,
    order_record.country_code,
    order_record.volume_count,
    order_record.fund_allocation_rule AS fund_allocation_rule_code,
    order_record.reopen_text,
    order_record.list_price,
    order_record.list_price_foreign_amt,
    order_record.list_price_discount_amt,
    order_record.list_price_service_charge,
    order_record.is_suppressed,
    order_record.fund_copies_paid
  FROM iiirecord.order_record;
```

IUG Connections

order_record_cmf

These are not our fund codes

sierra_view.order_record_cmf

id	order_record_id	display_order	fund_code	copies	location_code
2650093	476759621246	0	00000	1	ca4a
2650092	476759621245	0	00000	1	ca4a
2650091	476759621244	0	00000	1	ca4a
2646203	476748804344	0	00000	1	wata
2645231	476748374051	0	00000	1	ntnn
2645230	476748374050	0	00000	1	ntnn
2645229	476748374049	0	00000	2	ntnn
2642171	476752507872	0	00000	1	conal
2638350	476748373947	0	00000	1	ntnj
2634365	476748373814	0	00000	1	ntnj
2634293	476748373811	0	00000	1	ntnn
2634290	476748373808	0	00000	1	ntnn
2634288	476748373806	0	00000	1	ntnn
2634287	476748373805	0	00000	1	ntnn
2595588	476749174144	0	00000	1	arln

No fund is shared by these locations

IUG Connections

fund_master and accounting_unit

Text

sierra_view.order_record_cmf

id	order_record_id	display_order	fund_code	copies	location_code
2650093	476759621246	0	00000	1	ca4a
2650092	476759621245	0	00000	1	ca4a
2650091	476759621244	0	00000	1	ca4a
2646203	476748804344	0	00000	1	wata
2645231	476748374051	0	00000	1	ntnn
2645230	476748374050	0	00000	1	ntnn
2645229	476748374049	0	00000	2	ntnn
2642171	476752507872	0	00000	1	conal
2638350	476748373947	0	00000	1	ntnj
2634365	476748373814	0	00000	1	ntnj
2634293	476748373811	0	00000	1	ntnn
2634290	476748373808	0	00000	1	ntnn
2634288	476748373806	0	00000	1	ntnn
2634287	476748373805	0	00000	1	ntnn
2595588	476749174144	0	00000	1	arln

Integer

sierra_view.fund_master

id	accounting_unit_id	code_num	code
1	2	1	af
2	3	1	afab
3	3	2	afcb
4	3	3	afso
5	3	4	arava
6	3	5	aravv
7	3	6	araba
8	3	7	arabb
9	3	8	arabc
10	3	9	arabd
11	3	10	arabe
12	3	11	frchi
13	3	12	arabg

sierra_view.accounting_unit

id	code_num
1	0
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8

IUG Connections

fund_master and accounting_unit

ID not the actual accounting unit

sierra_view.order_record_cmf

id	order_record_id	display_order	fund_code	copies	location_code
2650093	476759621246	0	00000	1	ca4a
2650092	476759621245	0	00000	1	ca4a
2650091	476759621244	0	00000	1	ca4a
2646203	476748804344	0	00000	1	wata
2645231	476748374051	0	00000	1	ntnn
2645230	476748374050	0	00000	1	ntnn
2645229	476748374049	0	00000	2	ntnn
2642171	476752507872	0	00000	1	conal
2638350	476748373947	0	00000	1	ntnj
2634365	476748373814	0	00000	1	ntnj
2634293	476748373811	0	00000	1	ntnn
2634290	476748373808	0	00000	1	ntnn
2634288	476748373806	0	00000	1	ntnn
2634287	476748373805	0	00000	1	ntnn
2595588	476749174144	0	00000	1	arln

sierra_view.fund_master

id	accounting_unit_id	code_num	code
1	2	1	af
2	3	1	afab
3	3	2	afcb
4	3	3	afso
5	3	4	arava
6	3	5	aravv
7	3	6	araba
8	3	7	arabb
9	3	8	arabc
10	3	9	arabd
11	3	10	arabe
12	3	11	frchi
13	3	12	arabg

sierra_view.accounting_unit

id	code_num
1	0
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8

IUG Connections

Open Orders By Fund

```sql
SELECT
fm.code,
SUM(cmf.copies) AS copies_on_order

FROM
sierra_view.order_record o
JOIN
sierra_view.order_record_cmf cmf
ON
o.id = cmf.order_record_id
JOIN
sierra_view.accounting_unit a
ON
o.accounting_unit_code_num = a.code_num AND a.code_num = 12
JOIN
sierra_view.fund_master fm
ON
fm.code_num = cmf.fund_code::INT AND fm.accounting_unit_id = a.id

WHERE
o.order_status_code IN ('o','q','g','d')
GROUP BY 1
```

Result #1 (21r × 2c)

code	copies_on_order
bat	1
bjt	38
blt	5
ert	11
faf	82
fat	596

4 Tables later

Ensure accounting unit between tables match

Cast Fund_code to int so data types match

And we're not done yet

IUG Connections

The multi problem

If an order is for locations, then order_record_cmf add an extra row for the overall total

Can lead to a one-to-many join error

order_record_cmf (47,683r × 7c)

id	order_record_id	display_order	fund_code	copies	location_code	order_record_id
94647	476742871251	0	00031	3	multi	476742871251
94648	476742871251	1	00031	1	brka	476742871251
94649	476742871251	2	00031	1	br2a	476742871251
94650	476742871251	3	00031	1	br3a	476742871251
94657	476742871259	0	00031	3	multi	476742871259
94659	476742871259	1	00031	1	brka	476742871259
94661	476742871259	2	00031	1	br2a	476742871259
94663	476742871259	3	00031	1	br3a	476742871259
94658	476742871262	0	65535	2	multi	476742871262
94660	476742871262	1	00010	1	brkr	476742871262
94662	476742871262	2	00049	1	br2r	476742871262

De-Dupe orders

```sql
SELECT
fm.code,
SUM(cmf.copies) AS copies_on_order

FROM
sierra_view.order_record o
JOIN
sierra_view.order_record_cmf cmf
ON
o.id = cmf.order_record_id AND cmf.location_code != 'multi'
JOIN
sierra_view.accounting_unit a
ON
o.accounting_unit_code_num = a.code_num
JOIN
sierra_view.fund_master fm
ON
fm.code_num = cmf.fund_code::INT AND fm.accounting_unit_id = a.id
LEFT JOIN
sierra_view.order_record_paid op
ON
o.id = op.order_record_id AND o.order_status_code = 'q'

WHERE
o.accounting_unit_code_num = 12
AND o.order_status_code IN ('o','q','g','d')

GROUP BY 1
```

Result #1 (21r × 2c)

code	copies_on_order
bat	1
bjt	38
blt	5
ert	11
faf	82
fat	596
fjf	33
fjt	76
flt	200

Limit to location != 'multi

IUG Connections

The other multi problem

If you need to use the fund table, you'll find that each fund is actually up to three different funds

```sql
SELECT
*
FROM
sierra_view.fund f
ORDER BY f.acct_unit, f.fund_code, f.fund_type
```

fund (6,535r × 11c)

acct_unit	fund_type	fund_code	external_fund	appropriation	expenditure	encumbrance
2	fbal	araba	0	2000000	1672572	247659
2	fprevbal	araba	0	2000000	1584398	322796
2	oldfbal	araba	0	1820000	1910202	0
2	fbal	arabb	0	300000	298538	6691
2	fprevbal	arabb	0	300000	277102	24546
2	oldfbal	arabb	0	300000	300477	0
2	fbal	arabc	0	150000	148632	2805
2	fprevbal	arabc	0	150000	117722	27632
2	oldfbal	arabc	0	150000	150597	0
2	fbal	arabd	0	380000	280869	20939
2	fprevbal	arabd	0	380000	248458	47976
2	oldfbal	arabd	0	380000	374530	0

IUG Connections

Things I Wish I Had Known About SierraSQL

9. Is your item_status_code '-' because I'm checking you out

9. Is your item_status_code '-' because I'm checking you out

```sql
-- show item_status_codes
SELECT
isp.display_order,
isp.code AS item_status_code,
ispn.name AS item_status_name
FROM
sierra_view.item_status_property AS isp
JOIN sierra_view.item_status_property_name AS ispn
  ON ispn.item_status_property_id = isp.id
ORDER BY
isp.display_order
```

	display_order	item_status_code	item_status_name
1	0	!	ON HOLDSHELF
2	1	#	SearchOH/OL RECEIVED
3	2	$	LOST AND PAID
4	3	%	SearchOH/OL RETURNED
5	4	&	SearchOH/OHIOLINK REQUEST
6	5	(	SearchOH/OL PAGED
7	6	)	SearchOH/OL CANCEL'D
8	7	+	RENEWAL PENDING
9	8	-	CHECK SHELVES
10	9	=	RENEWAL DENIED
11	10	@	SearchOH/OL OFFSITE
12	11	_	SearchOH/OL RE-REQUEST
13	12	b	AT BINDERY
14	13	d	ON ORDER
15	14	e	EXCUSED LOSS
16	15	f	DISCARD TO FRIENDS
17	16	g	LONG INTRANSIT
18	17	i	MISSING IN INVENTORY
19	18	l	LOST IN SYMPHONY
20	19	m	MISSING
21	20	n	BILLED
22	21	o	LIBRARY USE ONLY
23	22	p	IN PROCESSING
24	23	r	IN REPAIR
25	24	s	ON SEARCH
26	25	t	IN TRANSIT
27	26	u	ORDER PENDING
28	27	v	ONLINE
29	28	w	WITHDRAWN
30	29	y	UNAVAILABLE
31	30	z	CLMS RETD

Item “checkout” is not a “*status*” … it’s a property of the item defined by the table view, **sierra_view.”checkout”**

9. Is your item_status_code '-' because I'm checking you out

	item_record_num	location_code	item_status_code	checkout_date	due_date	overdue_date
1	11070602	dpalc	w	[NULL]	[NULL]	[NULL]
2	11070604	dpalc	w	[NULL]	[NULL]	[NULL]
3	[illegible]	[illegible]	[illegible]	[NULL]	[NULL]	[NULL]
4	[illegible]	[illegible]	[illegible]	[NULL]	[NULL]	[NULL]
5	[illegible]	[illegible]	[illegible]	[NULL]	[NULL]	[NULL]
6	[illegible]	[illegible]	[illegible]	[NULL]	[NULL]	[NULL]
7	[illegible]	[illegible]	[illegible]	[NULL]	[NULL]	[NULL]
8	[illegible]	[illegible]	[illegible]	2022-03-08	2022-04-19	[NULL]
9	[illegible]	[illegible]	[illegible]	2022-01-20	2022-03-03	2022-03-08
10	[illegible]	[illegible]	[illegible]	[NULL]	[NULL]	[NULL]
11	[illegible]	[illegible]	[illegible]	[NULL]	[NULL]	[NULL]

```
SELECT
  rm.record_num AS item_record_num,
  ir.location_code,
  ir.item_status_code,
  date(c.checkout_gmt) AS checkout_date,
  date(c.due_gmt) AS due_date,
  date(c.overdue_gmt) AS overdue_date
FROM
  sierra_view.item_record AS ir
  JOIN sierra_view.record_metadata AS rm
    ON rm.id = ir.record_id
  LEFT JOIN sierra_view.checkout AS c
    ON c.item_record_id = ir.record_id
WHERE
  ir.location_code = 'dpalc'
```

Use a LEFT JOIN here to make sure you don't exclude non-checked-out items in your query

9. Is your item_status_code '-' because I'm checking you out

Remember, that the aggregate function “count” will count rows containing a non-NULL values!

We can also count other values in the same query… e.g. checkouts that have “overdue” dates.

```
SELECT
ir.location_code,
count(ir.*) AS count_items_total,
count(c.checkout_gmt) AS count_items_checkout,
round(
  (count(c.checkout_gmt) :: NUMERIC / count(ir.*) * 100),
  2
) AS pct_checkout,
count(c.overdue_gmt) AS count_overdue,
round(
  (count(c.overdue_gmt) :: NUMERIC / count(ir.*) * 100),
  2
) AS pct_overdue
FROM
```

	location_code	count_items_total	count_items_checkout	pct_checkout	count_overdue	pct_overdue
1	dpalc	503	137	27.24	32	6.36

```
LEFT JOIN sierra_view.checkout AS c
  ON c.item_record_id = ir.record_id
WHERE
ir.location_code = 'dpalc'
GROUP BY
1
```

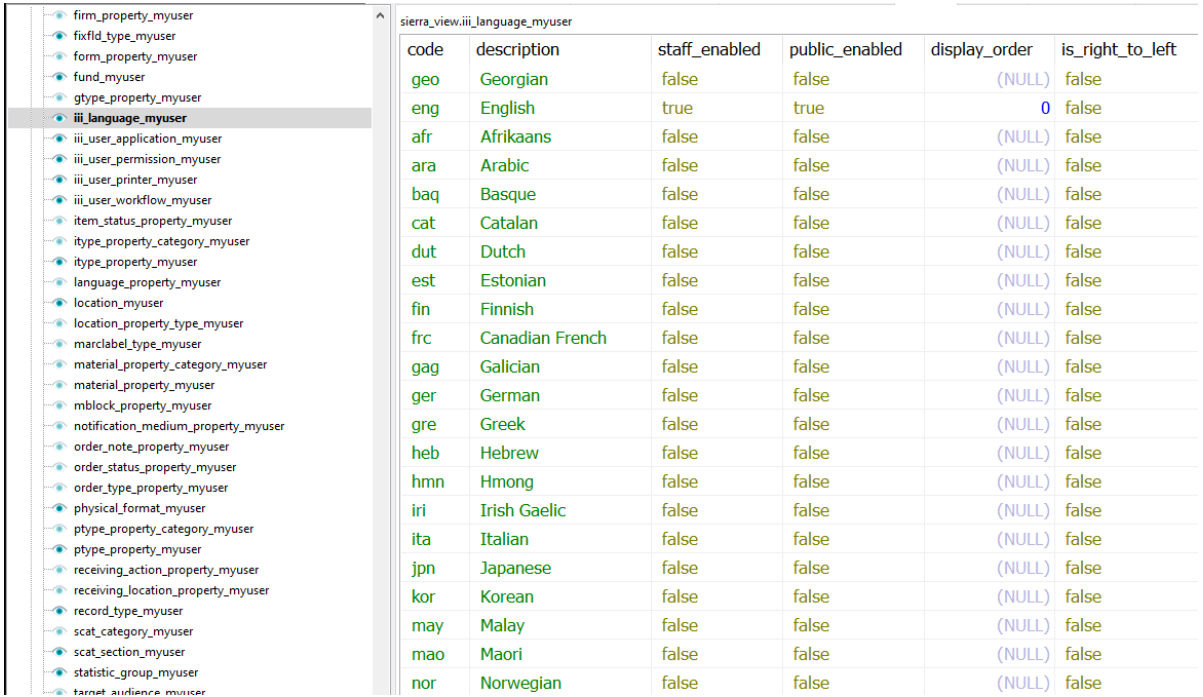
Things I Wish I Had Known About SierraSQL

10. Whose User is it anyway?

_myuser tables

Tables provide translations for fixed field and system codes

The translation you see are dependent on the language settings for your user account



code	description	staff_enabled	public_enabled	display_order	is_right_to_left
geo	Georgian	false	false	(NULL)	false
eng	English	true	true	0	false
afr	Afrikaans	false	false	(NULL)	false
ara	Arabic	false	false	(NULL)	false
baq	Basque	false	false	(NULL)	false
cat	Catalan	false	false	(NULL)	false
dut	Dutch	false	false	(NULL)	false
est	Estonian	false	false	(NULL)	false
fin	Finnish	false	false	(NULL)	false
frc	Canadian French	false	false	(NULL)	false
gag	Galician	false	false	(NULL)	false
ger	German	false	false	(NULL)	false
gre	Greek	false	false	(NULL)	false
heb	Hebrew	false	false	(NULL)	false
hmn	Hmong	false	false	(NULL)	false
iri	Irish Gaelic	false	false	(NULL)	false
ita	Italian	false	false	(NULL)	false
jpn	Japanese	false	false	(NULL)	false
kor	Korean	false	false	(NULL)	false
may	Malay	false	false	(NULL)	false
mao	Maori	false	false	(NULL)	false
nor	Norwegian	false	false	(NULL)	false

_myuser tables

Set Language to Chinese in Admin corner and you see this

*assuming Chinese translations exist in your system

sierra_view.language_property_myuser

code	display_order	is_public	name
heb	(NULL)	true	海地克里奥尔语
iii	(NULL)	false	四川彝语(诺苏语)
xho	(NULL)	false	依地语
swe	(NULL)	true	瑞典语
arg	(NULL)	false	阿拉伯语
est	(NULL)	false	巴斯克语
guj	(NULL)	true	印地语
ara	(NULL)	true	阿拉贡语
pol	(NULL)	true	巴利语
kur	(NULL)	false	吉尔吉斯语
aze	(NULL)	false	白俄罗斯语
glg	(NULL)	false	古吉拉特语
kin	(NULL)	false	朝鲜语、韩语
tib	(NULL)	false	他加禄语
kaz	(NULL)	false	格陵兰语
her	(NULL)	false	匈牙利语
gle	(NULL)	false	瓜拉尼语
arm	(NULL)	true	阿萨姆语
kir	(NULL)	false	卡努里语

IUG Connections

_myuser tables

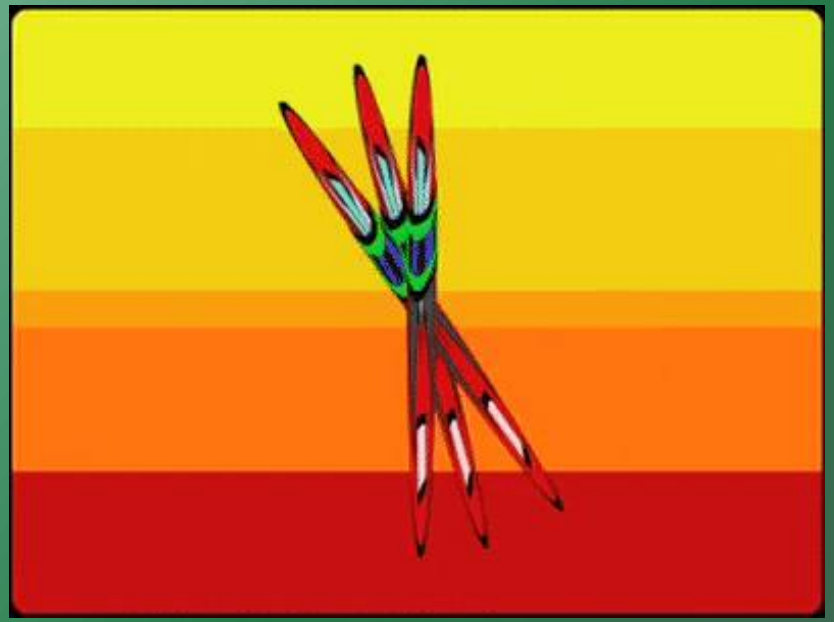
Be sure to set a language, or else all translations will be set to null

This is easy to miss if creating an account solely used for SQL access

sierra_view.language_property_myuser

code	display_order	is_public	name
hin	(NULL)	true	(NULL)
chv	(NULL)	false	(NULL)
cre	(NULL)	false	(NULL)
ave	(NULL)	false	(NULL)
cos	(NULL)	false	(NULL)
aze	(NULL)	false	(NULL)
kua	(NULL)	false	(NULL)
her	(NULL)	false	(NULL)
gle	(NULL)	false	(NULL)
chi	(NULL)	true	(NULL)
lin	(NULL)	false	(NULL)
zul	(NULL)	false	(NULL)
arm	(NULL)	true	(NULL)
sun	(NULL)	false	(NULL)
khm	(NULL)	false	(NULL)
iii	(NULL)	false	(NULL)
tuk	(NULL)	false	(NULL)

11. Everybody to the limit



All Together Now

The circ_trans table only stores

2 weeks of data by default

You can open a ticket with Innovative to Expand that

But that will not take effect retroactively

Thank You

Jeremy Goldstein jgoldstein@minlib.net
Ray Voelker ray.voelker@cincinnatilibrary.org