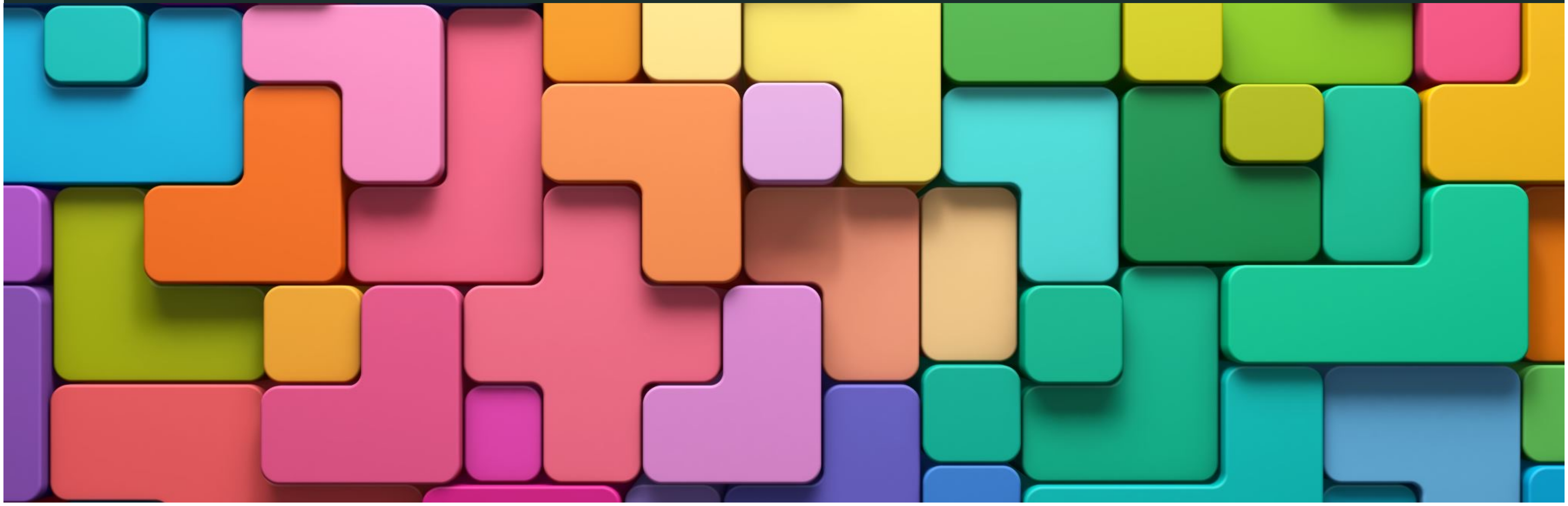
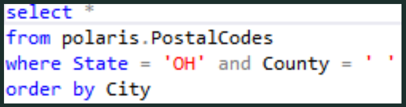
Beginner's SQL

My journey from confused to confident



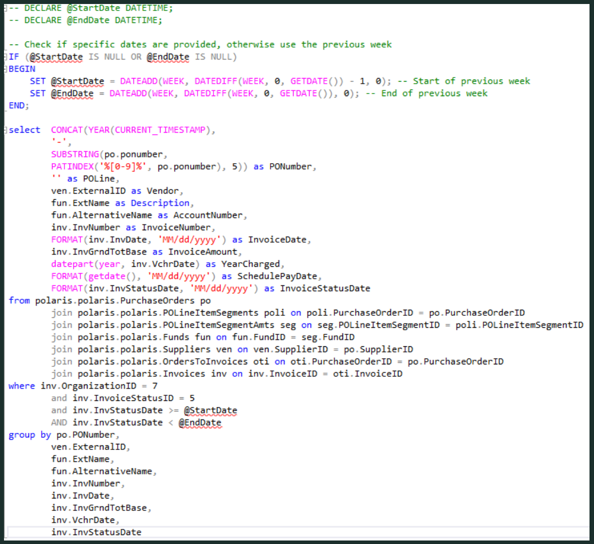
My SQL Evolution

Summer 2021



```sql
select *
from polaris.PostalCodes
where State = 'OH' and County = ' '
order by City
```

Summer 2023



```sql
-- DECLARE @StartDate DATETIME;
-- DECLARE @EndDate DATETIME;

-- Check if specific dates are provided, otherwise use the previous week
IF (@StartDate IS NULL OR @EndDate IS NULL)
BEGIN
    SET @StartDate = DATEADD(WEEK, DATEDIFF(WEEK, 0, GETDATE()) - 1, 0); -- Start of previous week
    SET @EndDate = DATEADD(WEEK, DATEDIFF(WEEK, 0, GETDATE()), 0); -- End of previous week
END;

select  CONCAT(YEAR(CURRENT_TIMESTAMP),
        '-',
        SUBSTRING(po.ponumber,
        PATINDEX('%[0-9]%', po.ponumber), 5)) as PONumber,
        '' as POLine,
        ven.ExternalID as Vendor,
        fun.ExtName as Description,
        fun.AlternativeName as AccountNumber,
        inv.InvNumber as InvoiceNumber,
        FORMAT(inv.InvDate, 'MM/dd/yyyy') as InvoiceDate,
        inv.InvGrndTotBase as InvoiceAmount,
        datepart(year, inv.VchrDate) as YearCharged,
        FORMAT(getdate(), 'MM/dd/yyyy') as SchedulePayDate,
        FORMAT(inv.InvStatusDate, 'MM/dd/yyyy') as InvoiceStatusDate
from polaris.polaris.PurchaseOrders po
        join polaris.polaris.POLineItemSegments poli on poli.PurchaseOrderID = po.PurchaseOrderID
        join polaris.polaris.POLineItemSegmentAmts seg on seg.POLineItemSegmentID = poli.POLineItemSegmentID
        join polaris.polaris.Funds fun on fun.FundID = seg.FundID
        join polaris.polaris.Suppliers ven on ven.SupplierID = po.SupplierID
        join polaris.polaris.OrdersToInvoices oti on oti.PurchaseOrderID = po.PurchaseOrderID
        join polaris.polaris.Invoices inv on inv.InvoiceID = oti.InvoiceID
where inv.OrganizationID = 7
        and inv.InvoiceStatusID = 5
        and inv.InvStatusDate >= @StartDate
        AND inv.InvStatusDate < @EndDate
group by po.PONumber,
        ven.ExternalID,
        fun.ExtName,
        fun.AlternativeName,
        inv.InvNumber,
        inv.InvDate,
        inv.InvGrndTotBase,
        inv.VchrDate,
        inv.InvStatusDate
```

Tools you will need:

Microsoft SQL Server Management Studio

→ Microsoft SSMS is a commonly used free interface for communicating with a database

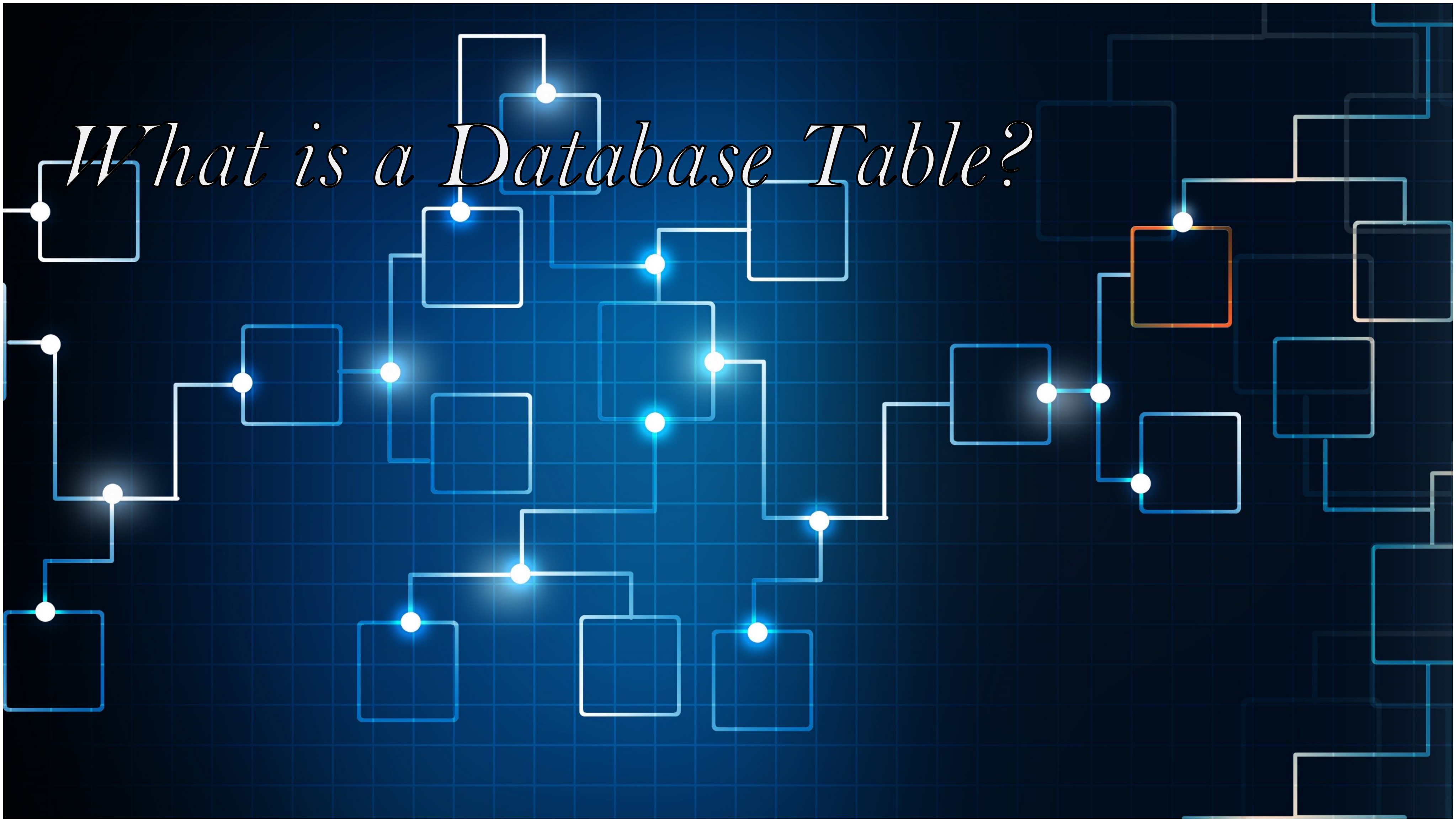
→ https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16

What is SQL?

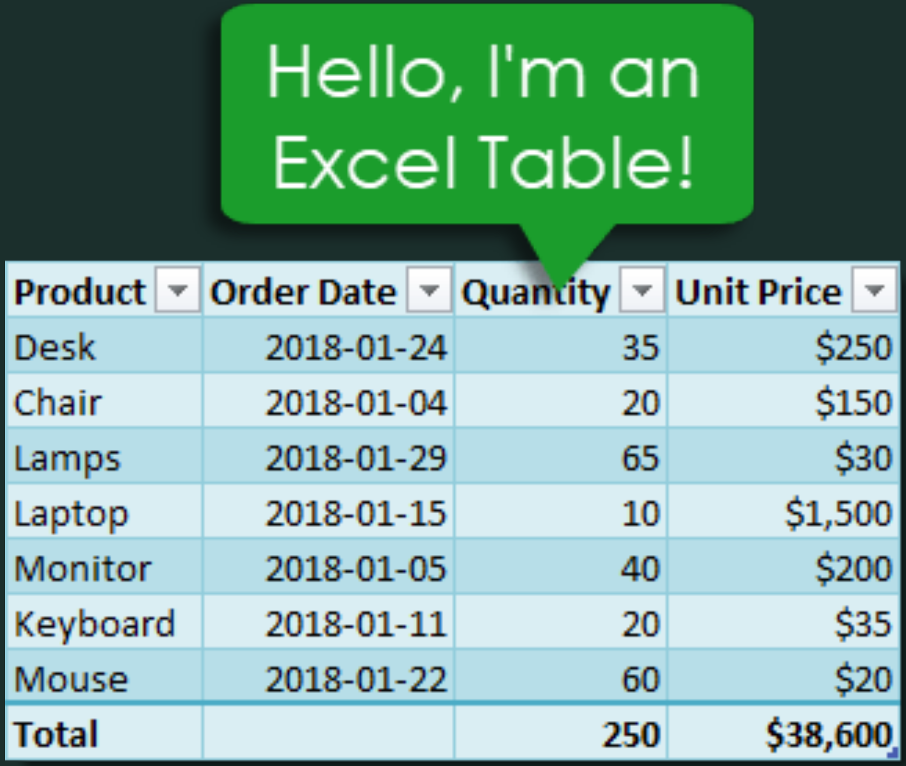
Structured Query Language

- → Used to talk with databases
- → The standard for relational database management systems

What is a Database Table?

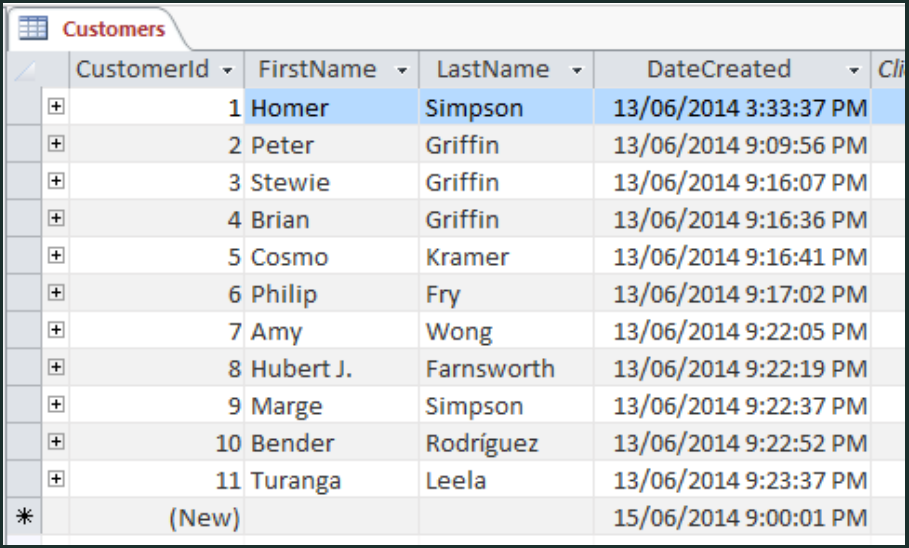


Excel



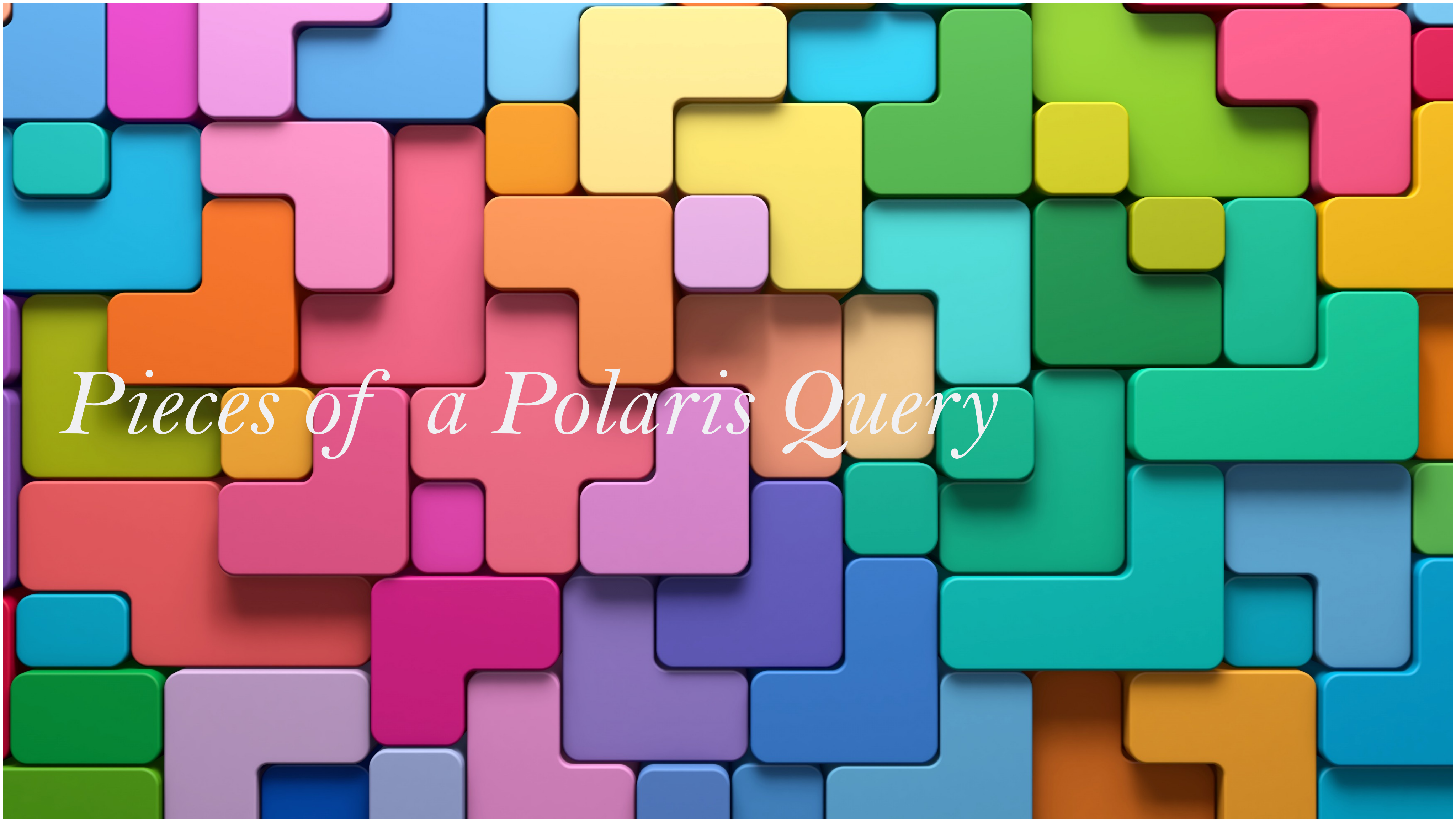
Product	Order Date	Quantity	Unit Price
Desk	2018-01-24	35	$250
Chair	2018-01-04	20	$150
Lamps	2018-01-29	65	$30
Laptop	2018-01-15	10	$1,500
Monitor	2018-01-05	40	$200
Keyboard	2018-01-11	20	$35
Mouse	2018-01-22	60	$20
Total		**250**	**$38,600**

Database Table



Customers

	CustomerId	FirstName	LastName	DateCreated	Cli
⊞	1	Homer	Simpson	13/06/2014 3:33:37 PM	
⊞	2	Peter	Griffin	13/06/2014 9:09:56 PM	
⊞	3	Stewie	Griffin	13/06/2014 9:16:07 PM	
⊞	4	Brian	Griffin	13/06/2014 9:16:36 PM	
⊞	5	Cosmo	Kramer	13/06/2014 9:16:41 PM	
⊞	6	Philip	Fry	13/06/2014 9:17:02 PM	
⊞	7	Amy	Wong	13/06/2014 9:22:05 PM	
⊞	8	Hubert J.	Farnsworth	13/06/2014 9:22:19 PM	
⊞	9	Marge	Simpson	13/06/2014 9:22:37 PM	
⊞	10	Bender	Rodríguez	13/06/2014 9:22:52 PM	
⊞	11	Turanga	Leela	13/06/2014 9:23:37 PM	
*	(New)			15/06/2014 9:00:01 PM	



Pieces of a Polaris Query

*SELECT **

Nothing bad ever happens with SELECT – you will not break anything!

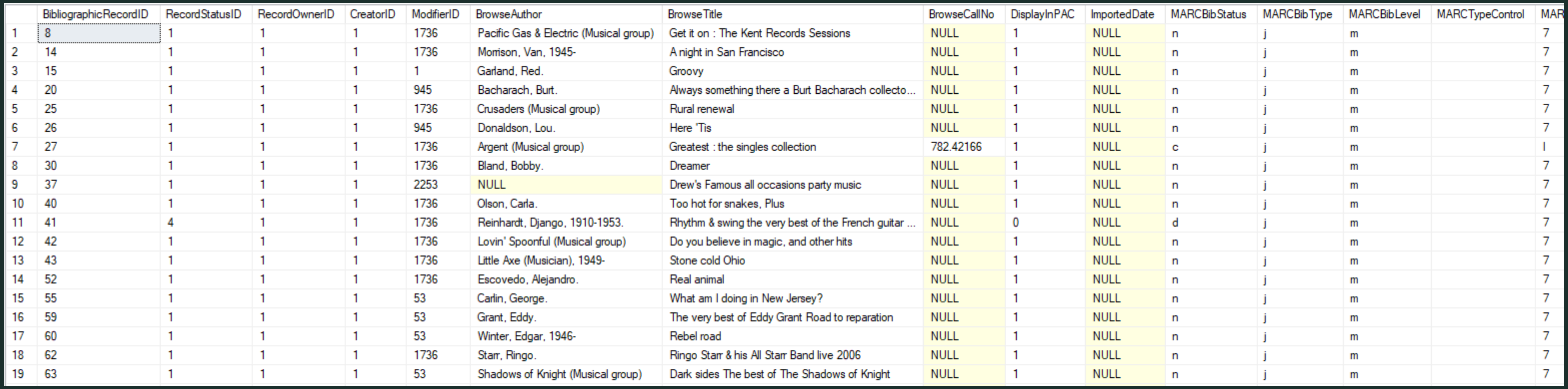
* Is a WILDCARD character which means “Give me everything”

FROM Polaris.Polaris.

FROM gives your database management system directions – where are you trying to look

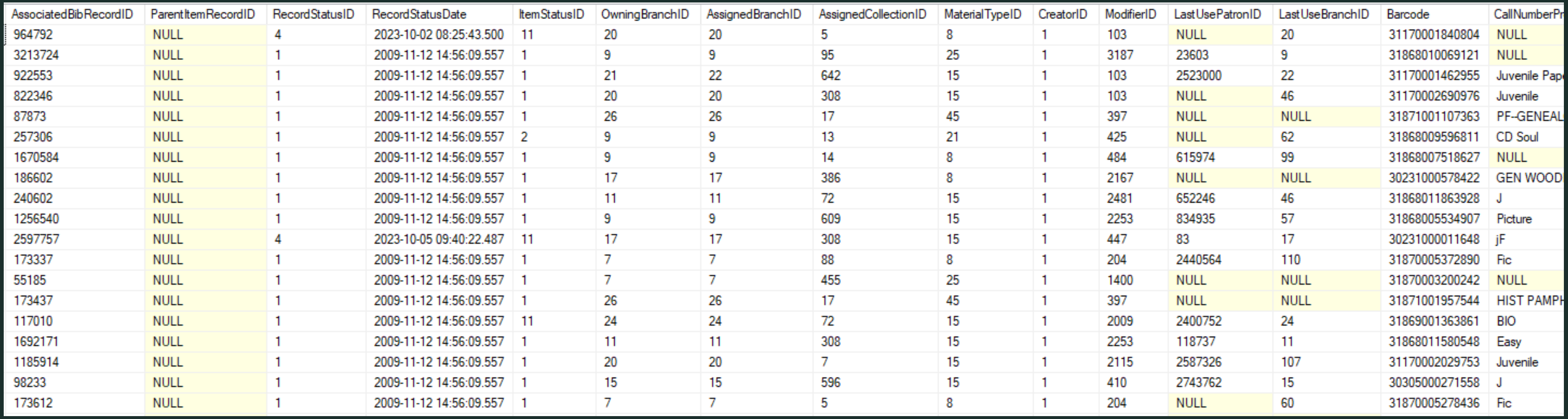
In the Polaris database environment, with a few exceptions, directions start with Polaris.Polaris.

BibliographicRecords



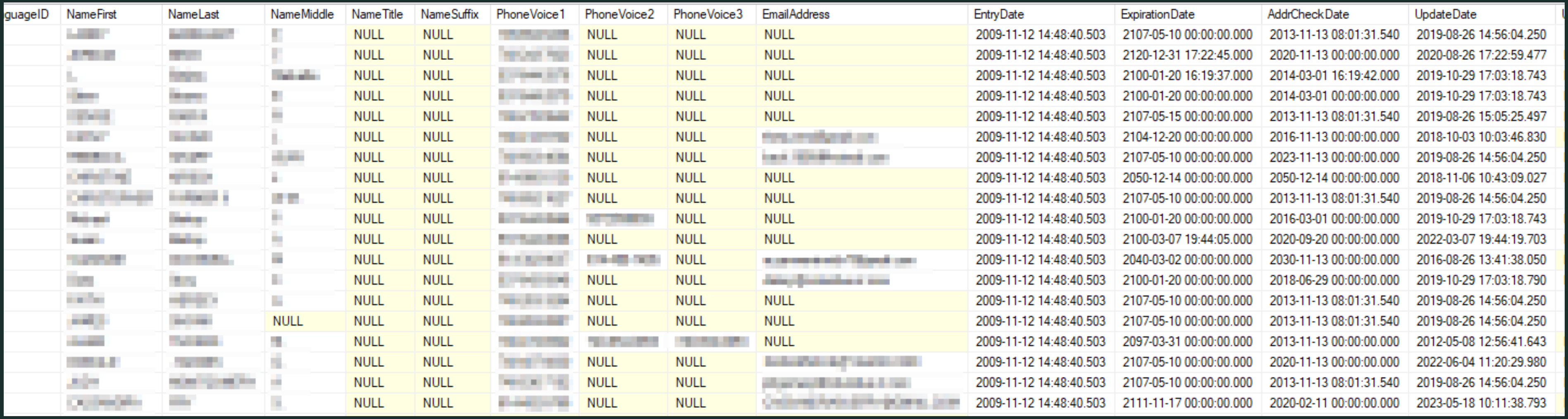
	BibliographicRecordID	RecordStatusID	RecordOwnerID	CreatorID	ModifierID	BrowseAuthor	BrowseTitle	BrowseCallNo	DisplayInPAC	ImportedDate	MARCBibStatus	MARCBibType	MARCBibLevel	MARCTypeControl	MAR
1	8	1	1	1	1736	Pacific Gas & Electric (Musical group)	Get it on : The Kent Records Sessions	NULL	1	NULL	n	j	m		7
2	14	1	1	1	1736	Morrison, Van, 1945-	A night in San Francisco	NULL	1	NULL	n	j	m		7
3	15	1	1	1	1	Garland, Red.	Groovy	NULL	1	NULL	n	j	m		7
4	20	1	1	1	945	Bacharach, Burt.	Always something there a Burt Bacharach collecto...	NULL	1	NULL	n	j	m		7
5	25	1	1	1	1736	Crusaders (Musical group)	Rural renewal	NULL	1	NULL	n	j	m		7
6	26	1	1	1	945	Donaldson, Lou.	Here 'Tis	NULL	1	NULL	n	j	m		7
7	27	1	1	1	1736	Argent (Musical group)	Greatest : the singles collection	782.42166	1	NULL	c	j	m		l
8	30	1	1	1	1736	Bland, Bobby.	Dreamer	NULL	1	NULL	n	j	m		7
9	37	1	1	1	2253	NULL	Drew's Famous all occasions party music	NULL	1	NULL	n	j	m		7
10	40	1	1	1	1736	Olson, Carla.	Too hot for snakes, Plus	NULL	1	NULL	n	j	m		7
11	41	4	1	1	1736	Reinhardt, Django, 1910-1953.	Rhythm & swing the very best of the French guitar ...	NULL	0	NULL	d	j	m		7
12	42	1	1	1	1736	Lovin' Spoonful (Musical group)	Do you believe in magic, and other hits	NULL	1	NULL	n	j	m		7
13	43	1	1	1	1736	Little Axe (Musician), 1949-	Stone cold Ohio	NULL	1	NULL	n	j	m		7
14	52	1	1	1	1736	Escovedo, Alejandro.	Real animal	NULL	1	NULL	n	j	m		7
15	55	1	1	1	53	Carlin, George.	What am I doing in New Jersey?	NULL	1	NULL	n	j	m		7
16	59	1	1	1	53	Grant, Eddy.	The very best of Eddy Grant Road to reparation	NULL	1	NULL	n	j	m		7
17	60	1	1	1	53	Winter, Edgar, 1946-	Rebel road	NULL	1	NULL	n	j	m		7
18	62	1	1	1	1736	Starr, Ringo.	Ringo Starr & his All Starr Band live 2006	NULL	1	NULL	n	j	m		7
19	63	1	1	1	53	Shadows of Knight (Musical group)	Dark sides The best of The Shadows of Knight	NULL	1	NULL	n	j	m		7

ItemRecords



AssociatedBibRecordID	ParentItemRecordID	RecordStatusID	RecordStatusDate	ItemStatusID	OwningBranchID	AssignedBranchID	AssignedCollectionID	MaterialTypeID	CreatorID	ModifierID	LastUsePatronID	LastUseBranchID	Barcode	CallNumberPr
964792	NULL	4	2023-10-02 08:25:43.500	11	20	20	5	8	1	103	NULL	20	31170001840804	NULL
3213724	NULL	1	2009-11-12 14:56:09.557	1	9	9	95	25	1	3187	23603	9	31868010069121	NULL
922553	NULL	1	2009-11-12 14:56:09.557	1	21	22	642	15	1	103	2523000	22	31170001462955	Juvenile Pap
822346	NULL	1	2009-11-12 14:56:09.557	1	20	20	308	15	1	103	NULL	46	31170002690976	Juvenile
87873	NULL	1	2009-11-12 14:56:09.557	1	26	26	17	45	1	397	NULL	NULL	31871001107363	PF--GENEAL
257306	NULL	1	2009-11-12 14:56:09.557	2	9	9	13	21	1	425	NULL	62	31868009596811	CD Soul
1670584	NULL	1	2009-11-12 14:56:09.557	1	9	9	14	8	1	484	615974	99	31868007518627	NULL
186602	NULL	1	2009-11-12 14:56:09.557	1	17	17	386	8	1	2167	NULL	NULL	30231000578422	GEN WOOD
240602	NULL	1	2009-11-12 14:56:09.557	1	11	11	72	15	1	2481	652246	46	31868011863928	J
1256540	NULL	1	2009-11-12 14:56:09.557	1	9	9	609	15	1	2253	834935	57	31868005534907	Picture
2597757	NULL	4	2023-10-05 09:40:22.487	11	17	17	308	15	1	447	83	17	30231000011648	jF
173337	NULL	1	2009-11-12 14:56:09.557	1	7	7	88	8	1	204	2440564	110	31870005372890	Fic
55185	NULL	1	2009-11-12 14:56:09.557	1	7	7	455	25	1	1400	NULL	NULL	31870003200242	NULL
173437	NULL	1	2009-11-12 14:56:09.557	1	26	26	17	45	1	397	NULL	NULL	31871001957544	HIST PAMPH
117010	NULL	1	2009-11-12 14:56:09.557	11	24	24	72	15	1	2009	2400752	24	31869001363861	BIO
1692171	NULL	1	2009-11-12 14:56:09.557	1	11	11	308	15	1	2253	118737	11	31868011580548	Easy
1185914	NULL	1	2009-11-12 14:56:09.557	1	20	20	7	15	1	2115	2587326	107	31170002029753	Juvenile
98233	NULL	1	2009-11-12 14:56:09.557	1	15	15	596	15	1	410	2743762	15	30305000271558	J
173612	NULL	1	2009-11-12 14:56:09.557	1	7	7	5	8	1	204	NULL	60	31870005278436	Fic

PatronRegistration



guageID	NameFirst	NameLast	NameMiddle	NameTitle	NameSuffix	PhoneVoice1	PhoneVoice2	PhoneVoice3	EmailAddress	EntryDate	ExpirationDate	AddrCheckDate	UpdateDate
	[redacted]	[redacted]	[redacted]	NULL	NULL	[redacted]	NULL	NULL	NULL	2009-11-12 14:48:40.503	2107-05-10 00:00:00.000	2013-11-13 08:01:31.540	2019-08-26 14:56:04.250
	[redacted]	[redacted]	[redacted]	NULL	NULL	[redacted]	NULL	NULL	NULL	2009-11-12 14:48:40.503	2120-12-31 17:22:45.000	2020-11-13 00:00:00.000	2020-08-26 17:22:59.477
	[redacted]	[redacted]	[redacted]	NULL	NULL	[redacted]	NULL	NULL	NULL	2009-11-12 14:48:40.503	2100-01-20 16:19:37.000	2014-03-01 16:19:42.000	2019-10-29 17:03:18.743
	[redacted]	[redacted]	[redacted]	NULL	NULL	[redacted]	NULL	NULL	NULL	2009-11-12 14:48:40.503	2100-01-20 00:00:00.000	2014-03-01 00:00:00.000	2019-10-29 17:03:18.743
	[redacted]	[redacted]	[redacted]	NULL	NULL	[redacted]	NULL	NULL	NULL	2009-11-12 14:48:40.503	2107-05-15 00:00:00.000	2013-11-13 08:01:31.540	2019-08-26 15:05:25.497
	[redacted]	[redacted]	[redacted]	NULL	NULL	[redacted]	NULL	NULL	[redacted]	2009-11-12 14:48:40.503	2104-12-20 00:00:00.000	2016-11-13 00:00:00.000	2018-10-03 10:03:46.830
	[redacted]	[redacted]	[redacted]	NULL	NULL	[redacted]	NULL	NULL	[redacted]	2009-11-12 14:48:40.503	2107-05-10 00:00:00.000	2023-11-13 00:00:00.000	2019-08-26 14:56:04.250
	[redacted]	[redacted]	[redacted]	NULL	NULL	[redacted]	NULL	NULL	NULL	2009-11-12 14:48:40.503	2050-12-14 00:00:00.000	2050-12-14 00:00:00.000	2018-11-06 10:43:09.027
	[redacted]	[redacted]	[redacted]	NULL	NULL	[redacted]	NULL	NULL	NULL	2009-11-12 14:48:40.503	2107-05-10 00:00:00.000	2013-11-13 08:01:31.540	2019-08-26 14:56:04.250
	[redacted]	[redacted]	[redacted]	NULL	NULL	[redacted]	[redacted]	NULL	NULL	2009-11-12 14:48:40.503	2100-01-20 00:00:00.000	2016-03-01 00:00:00.000	2019-10-29 17:03:18.743
	[redacted]	[redacted]	[redacted]	NULL	NULL	[redacted]	NULL	NULL	NULL	2009-11-12 14:48:40.503	2100-03-07 19:44:05.000	2020-09-20 00:00:00.000	2022-03-07 19:44:19.703
	[redacted]	[redacted]	[redacted]	NULL	NULL	[redacted]	[redacted]	NULL	[redacted]	2009-11-12 14:48:40.503	2040-03-02 00:00:00.000	2030-11-13 00:00:00.000	2016-08-26 13:41:38.050
	[redacted]	[redacted]	[redacted]	NULL	NULL	[redacted]	NULL	NULL	[redacted]	2009-11-12 14:48:40.503	2100-01-20 00:00:00.000	2018-06-29 00:00:00.000	2019-10-29 17:03:18.790
	[redacted]	[redacted]	[redacted]	NULL	NULL	[redacted]	NULL	NULL	NULL	2009-11-12 14:48:40.503	2107-05-10 00:00:00.000	2013-11-13 08:01:31.540	2019-08-26 14:56:04.250
	[redacted]	[redacted]	NULL	NULL	NULL	[redacted]	NULL	NULL	NULL	2009-11-12 14:48:40.503	2107-05-10 00:00:00.000	2013-11-13 08:01:31.540	2019-08-26 14:56:04.250
	[redacted]	[redacted]	[redacted]	NULL	NULL	[redacted]	[redacted]	[redacted]	NULL	2009-11-12 14:48:40.503	2097-03-31 00:00:00.000	2013-11-13 00:00:00.000	2012-05-08 12:56:41.643
	[redacted]	[redacted]	[redacted]	NULL	NULL	[redacted]	NULL	NULL	[redacted]	2009-11-12 14:48:40.503	2107-05-10 00:00:00.000	2020-11-13 00:00:00.000	2022-06-04 11:20:29.980
	[redacted]	[redacted]	[redacted]	NULL	NULL	[redacted]	NULL	NULL	[redacted]	2009-11-12 14:48:40.503	2107-05-10 00:00:00.000	2013-11-13 08:01:31.540	2019-08-26 14:56:04.250
	[redacted]	[redacted]	[redacted]	NULL	NULL	[redacted]	NULL	NULL	[redacted]	2009-11-12 14:48:40.503	2111-11-17 00:00:00.000	2020-02-11 00:00:00.000	2023-05-18 10:11:38.793

What about filtering results?

WHERE

IDs

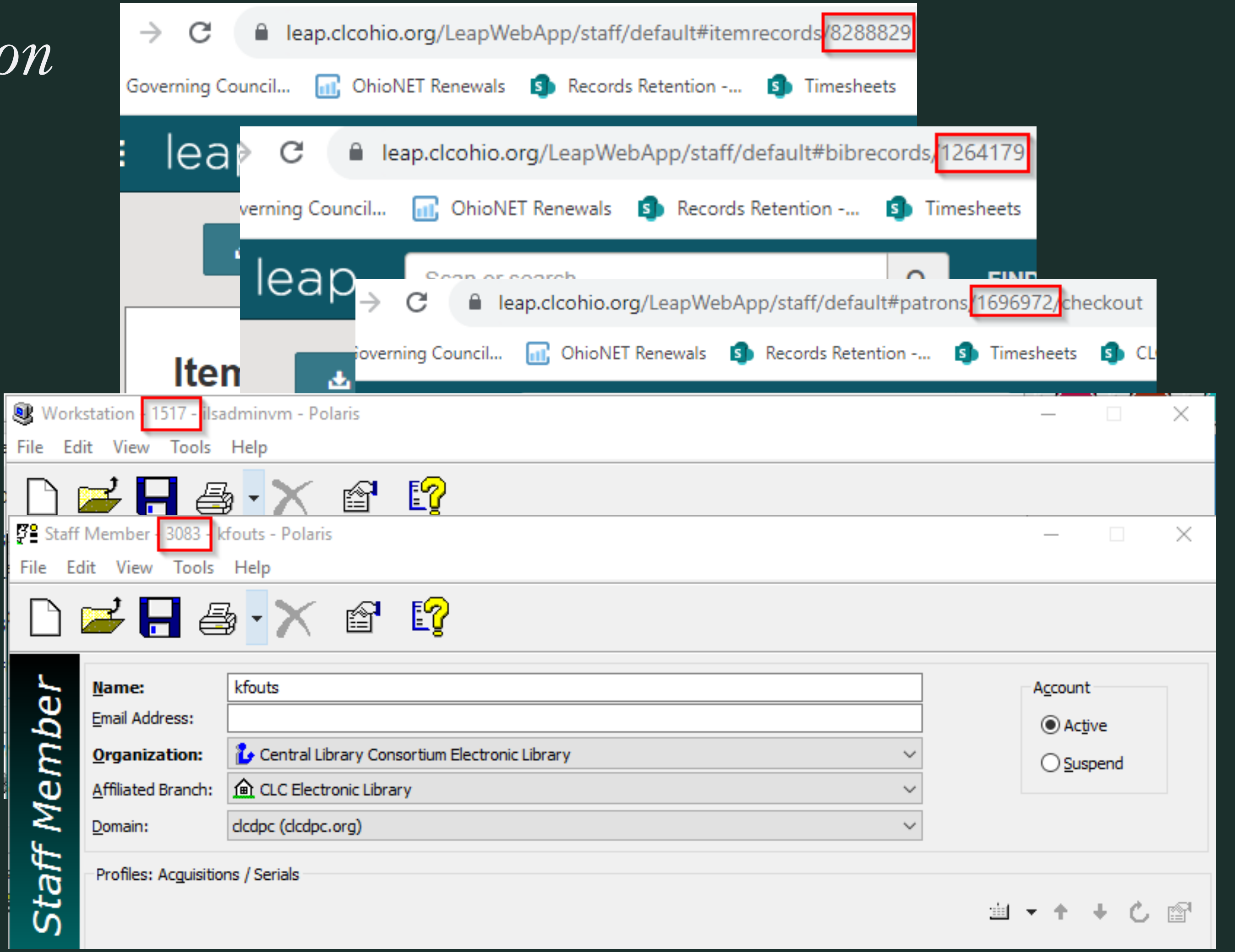
Arrays

Nested Queries

IDs Please

Polaris SQL is built on IDs

- ItemRecordID
- BibliographicRecordID
- PatronRecordID
- WorkstationID
- PolarisUserID
- BranchID
  - If you are a CLC member, we know your branch ID
  - If you are NOT a CLC member, ask your System Administrator or Site Manager


leap.clcohio.org/LeapWebApp/staff/default#itemrecords/8288829

leap.clcohio.org/LeapWebApp/staff/default#bibrecords/1264179

leap.clcohio.org/LeapWebApp/staff/default#patrons/1696972/checkout

Workstation - 1517 - ilsadminvm - Polaris

Staff Member - 3083 - kfouts - Polaris

Name: kfouts
Email Address:
Organization: Central Library Consortium Electronic Library
Affiliated Branch: CLC Electronic Library
Domain: clcdpc (clcdpc.org)
Account: [✓ Active] Suspend
Profiles: Acquisitions / Serials

ARRAYS

Why Arrays?

If you want to look at multiple results at the same time, an array allows you to input all the datapoints instead of running the query multiple times

→ **Make sure all your qualifiers are inside parentheses**

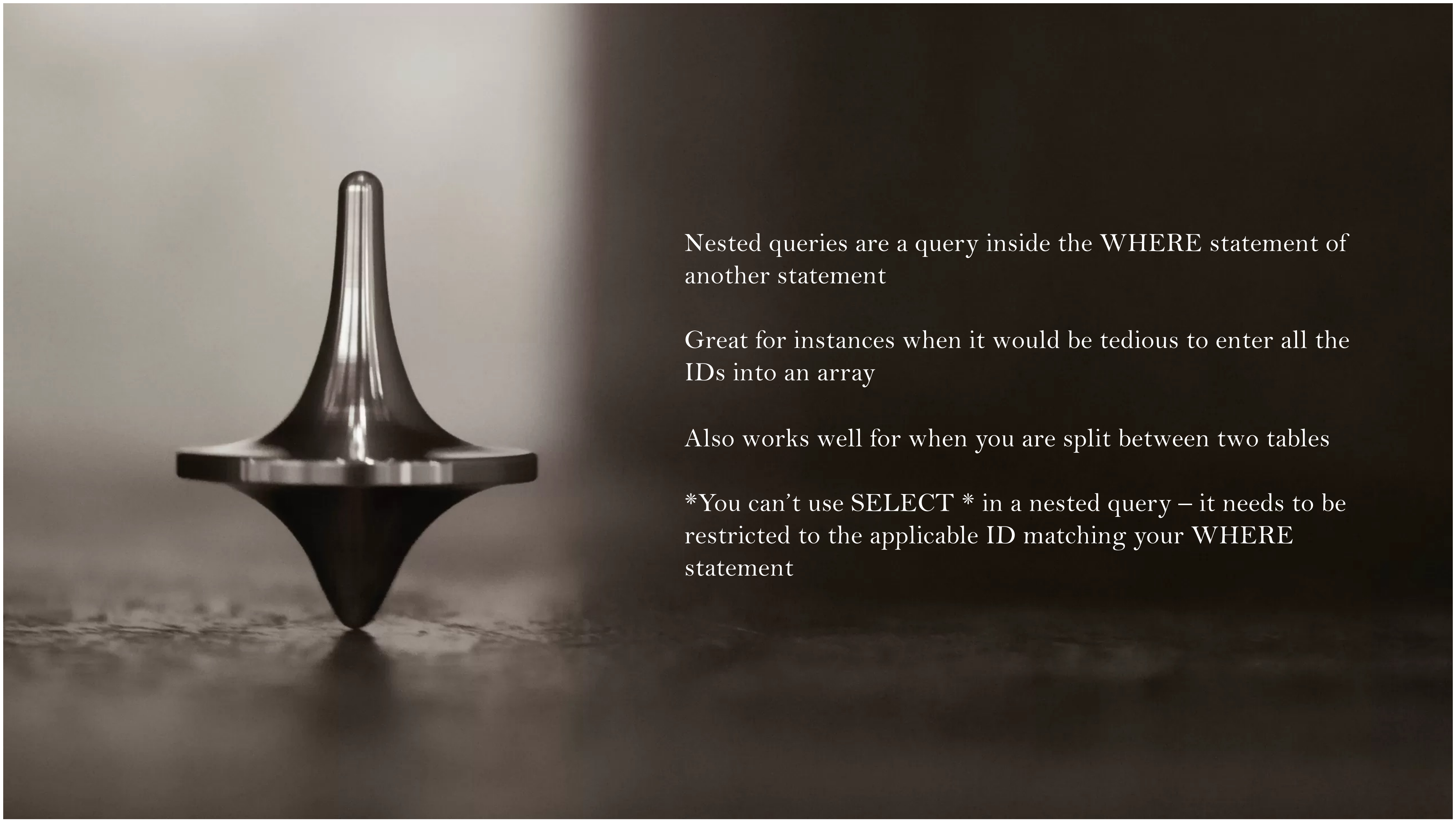
→ **You can put as many qualifiers are you want in the array**

```
SELECT *
FROM Polaris.Polaris.ItemRecords
WHERE ItemRecordID = 3
```

```
SELECT *
FROM Polaris.Polaris.ItemRecords
WHERE ItemRecordID = 4
```

```
SELECT *
FROM Polaris.Polaris.ItemRecords
WHERE ItemRecordID IN (3,4)
```

NESTED QUERIES



Nested queries are a query inside the WHERE statement of another statement

Great for instances when it would be tedious to enter all the IDs into an array

Also works well for when you are split between two tables

*You can’t use SELECT * in a nested query – it needs to be restricted to the applicable ID matching your WHERE statement

```
SELECT *

FROM Polaris.Polaris.Patrons

WHERE PatronID IN (SELECT PatronID
FROM Polaris.Polaris.PatronRegistration
WHERE EmailAddress IS NULL)
```

This nested query is the equivalent of manually typing in all 10,000 (or more!) IDs

Remember to test your nested query separately first to make sure it does what you want

Data Types

There are multiple data types in SQL databases. The most important are

- Numeric
  - Can be whole numbers or decimals
- Strings
  - Text based fields
    - Ex., Kalee, Columbus, Book titles
  - Sometimes strings can look like numbers, i.e. a phone number: 123-456-7890
- Binary
  - "Yes" or "No", typically represented as 1 and 0
- DateTime
  - The date including the time down to the nearest millisecond
    - ex. 2019-10-29 17:03:18.790

Operators

= vs != vs < vs > vs AND vs OR vs IN vs LIKE vs NOT IN vs NOT LIKE

These help narrow down your results to show only the specific things you want to see

= vs !=
Equals vs Not Equals

- = is used when you want your results to exactly match
- In my experience, it works best with numeric data types, but you can use = with all data types
- For example, you are looking for a specific Bibliographic Record:

```
SELECT *
FROM Polaris.Polaris.BibliographicRecords
WHERE BibliographicRecordID = 150160
```

- != works best when you want to exclude anything that doesn't exactly match
- If you would like to see all item records except those that are On-Order:

```
SELECT *
FROM Polaris.Polaris.ItemRecords
WHERE ItemStatusID != 13
```

< vs >
Greater Than vs Less Than

- The < and > operators work well with Numeric and DateTime datatypes.
- For example, you want to find patrons who owe more than $100:

```
SELECT *
FROM Polaris.Polaris.Patrons
WHERE ChargesAmount > 100
```

- Or all patrons registered in the month of September:

```
SELECT *
FROM Polaris.Polaris.PatronRegistration
WHERE RegistrationDate > ‘2023-09-01’ AND RegistrationDate < ‘2023-10-01’
```

AND

- → Great for when you want to match things with two or more points of criteria.
- → AND is used to link other operators together
- → Example, items that are checked in at your branch that have never circulated but are circulating items

```
SELECT *
FROM Polaris.Polaris.ItemRecords
WHERE ItemStatusID = 1 AND CheckInBranchID = 20 AND LifetimeCircCount = 0
AND NonCirculating = 0
```

This is an example of a BINARY datatype. The 0 means "No" as in these items are NOT flagged non-circulating

OR vs IN

- OR and IN can be used when you want results that match either condition specified.
  - IN is sometimes considered the "shorthand" for OR. They work with all datatypes
- If you want to see all items in Lost, Missing, or Withdrawn status

```
SELECT *
FROM Polaris.Polaris.ItemRecords
WHERE ItemStatusID = 7 OR ItemStatusID = 10 OR ItemStatusID = 11
```

And

```
SELECT *
FROM Polaris.Polaris.ItemRecords
WHERE ItemStatusID IN (7,10,11)
```

Return the same results!

LIKE

- LIKE is used most frequently with WILDCARD characters % and _
  - These are helpful for when you want to match part of a datatype
  - A% says to return anything that starts with an A and then has any number of characters following
  - A_ says to return anything that starts with an A and then has a single character afterwards
- For example, to get patrons that have 'School' as any part of their registered name:

```
SELECT *
FROM Polaris.Polaris.PatronRegistration
WHERE NameFirst LIKE '%school%' OR NameLast LIKE '%school%'
```

Here we're using OR in a way we couldn't replace it with IN since we want to return 'school' in either the first or last name

NOT IN

- → NOT IN is used when you want to see everything except things with specific qualifications.
- → If you want to see all items except those in Lost, Missing, or Withdrawn status

```
SELECT *
FROM Polaris.Polaris.ItemRecords
WHERE ItemStatusID NOT IN (7,10,11)
```

NOT LIKE

- NOT LIKE is used when you want to return results for anything that isn't like the word or phrase, usually when it is easier to exclude than include.
- For example, say you wanted to return all registered patrons who DON'T live in Columbus, but you also need to account for misspellings or shorthand:

```
SELECT *
FROM Polaris.Polaris.ViewPatronAddresses
WHERE City NOT LIKE 'Col%'
```

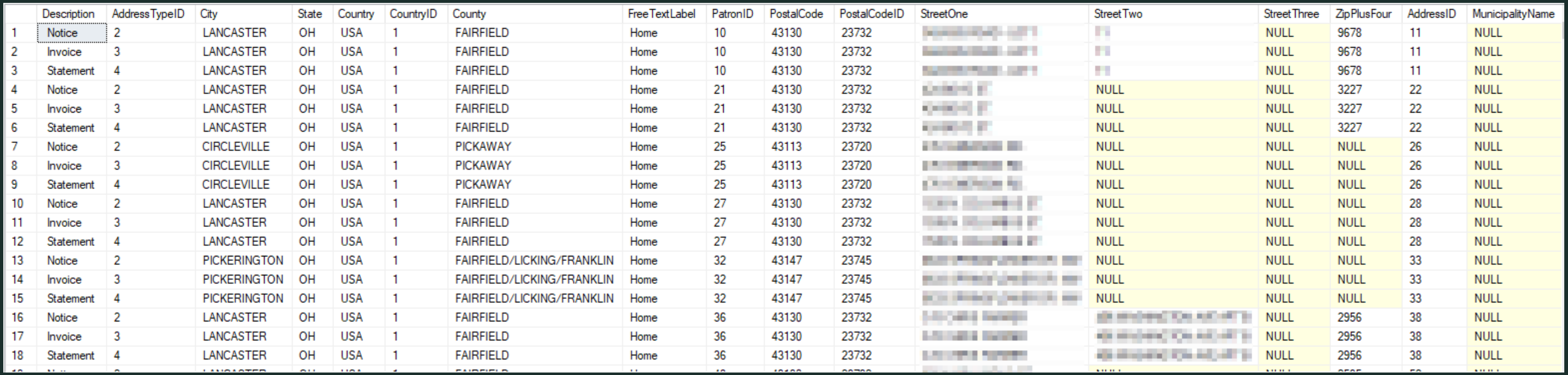
This is a very simplistic example and wouldn't return great results, especially for a multi-branch system

Joins

Sometimes, one table doesn't have all the information you need

In these cases, you need to bridge the gap with a JOIN

Let's take the ViewPatronAddresses example from earlier. The results of that query look like this:



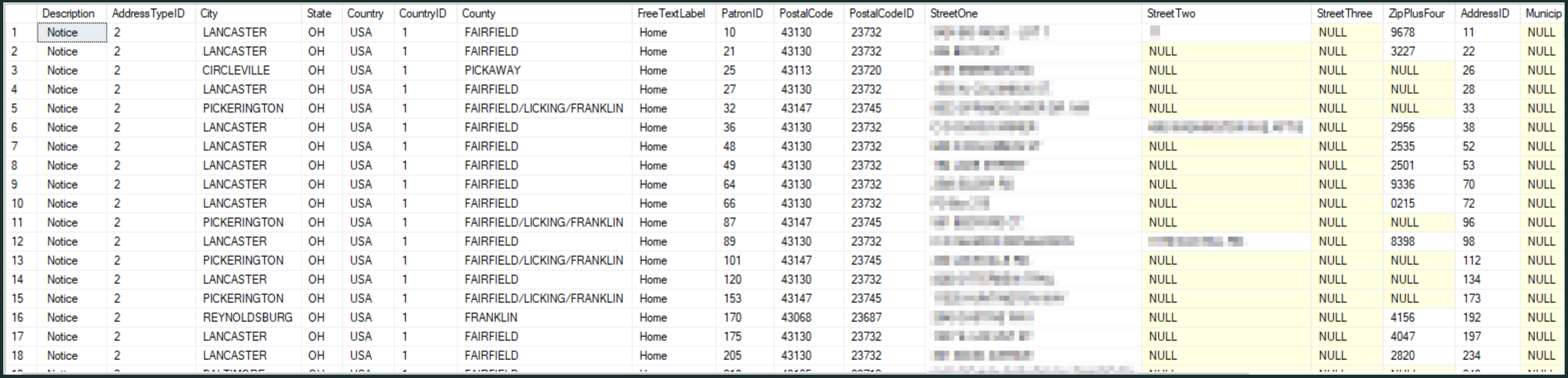
	Description	AddressTypeID	City	State	Country	CountryID	County	FreeTextLabel	PatronID	PostalCode	PostalCodeID	StreetOne	StreetTwo	StreetThree	ZipPlusFour	AddressID	MunicipalityName
1	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	10	43130	23732	[redacted]	[redacted]	NULL	9678	11	NULL
2	Invoice	3	LANCASTER	OH	USA	1	FAIRFIELD	Home	10	43130	23732	[redacted]	[redacted]	NULL	9678	11	NULL
3	Statement	4	LANCASTER	OH	USA	1	FAIRFIELD	Home	10	43130	23732	[redacted]	[redacted]	NULL	9678	11	NULL
4	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	21	43130	23732	[redacted]	NULL	NULL	3227	22	NULL
5	Invoice	3	LANCASTER	OH	USA	1	FAIRFIELD	Home	21	43130	23732	[redacted]	NULL	NULL	3227	22	NULL
6	Statement	4	LANCASTER	OH	USA	1	FAIRFIELD	Home	21	43130	23732	[redacted]	NULL	NULL	3227	22	NULL
7	Notice	2	CIRCLEVILLE	OH	USA	1	PICKAWAY	Home	25	43113	23720	[redacted]	NULL	NULL	NULL	26	NULL
8	Invoice	3	CIRCLEVILLE	OH	USA	1	PICKAWAY	Home	25	43113	23720	[redacted]	NULL	NULL	NULL	26	NULL
9	Statement	4	CIRCLEVILLE	OH	USA	1	PICKAWAY	Home	25	43113	23720	[redacted]	NULL	NULL	NULL	26	NULL
10	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	27	43130	23732	[redacted]	NULL	NULL	NULL	28	NULL
11	Invoice	3	LANCASTER	OH	USA	1	FAIRFIELD	Home	27	43130	23732	[redacted]	NULL	NULL	NULL	28	NULL
12	Statement	4	LANCASTER	OH	USA	1	FAIRFIELD	Home	27	43130	23732	[redacted]	NULL	NULL	NULL	28	NULL
13	Notice	2	PICKERINGTON	OH	USA	1	FAIRFIELD/LICKING/FRANKLIN	Home	32	43147	23745	[redacted]	NULL	NULL	NULL	33	NULL
14	Invoice	3	PICKERINGTON	OH	USA	1	FAIRFIELD/LICKING/FRANKLIN	Home	32	43147	23745	[redacted]	NULL	NULL	NULL	33	NULL
15	Statement	4	PICKERINGTON	OH	USA	1	FAIRFIELD/LICKING/FRANKLIN	Home	32	43147	23745	[redacted]	NULL	NULL	NULL	33	NULL
16	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	36	43130	23732	[redacted]	[redacted]	NULL	2956	38	NULL
17	Invoice	3	LANCASTER	OH	USA	1	FAIRFIELD	Home	36	43130	23732	[redacted]	[redacted]	NULL	2956	38	NULL
18	Statement	4	LANCASTER	OH	USA	1	FAIRFIELD	Home	36	43130	23732	[redacted]	[redacted]	NULL	2956	38	NULL

The biggest problem is we only have the PatronID and if we wanted to see additional information about the patron, it is not here.

There are also THREE addresses per patron which really clutters up the results.

Let’s clean up the query. First, let’s only return the Notice address type for each patron

```
SELECT *
FROM Polaris.Polaris.ViewPatronAddresses
WHERE City NOT LIKE ‘Col%' AND AddressTypeID = 2
```

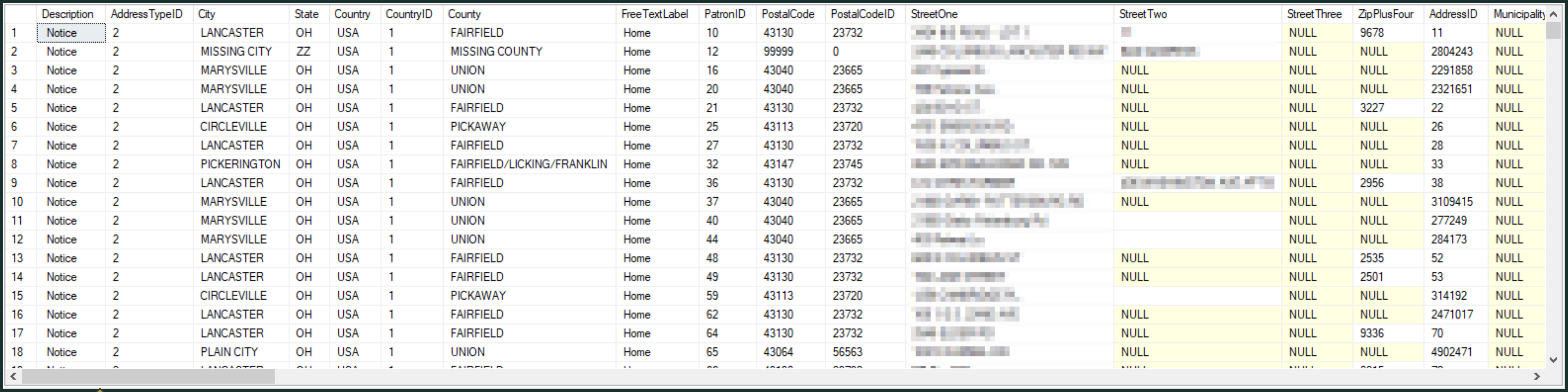


	Description	AddressTypeID	City	State	Country	CountryID	County	FreeTextLabel	PatronID	PostalCode	PostalCodeID	StreetOne	StreetTwo	StreetThree	ZipPlusFour	AddressID	Municip
1	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	10	43130	23732	[redacted]	[redacted]	NULL	9678	11	NULL
2	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	21	43130	23732	[redacted]	NULL	NULL	3227	22	NULL
3	Notice	2	CIRCLEVILLE	OH	USA	1	PICKAWAY	Home	25	43113	23720	[redacted]	NULL	NULL	NULL	26	NULL
4	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	27	43130	23732	[redacted]	NULL	NULL	NULL	28	NULL
5	Notice	2	PICKERINGTON	OH	USA	1	FAIRFIELD/LICKING/FRANKLIN	Home	32	43147	23745	[redacted]	NULL	NULL	NULL	33	NULL
6	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	36	43130	23732	[redacted]	[redacted]	NULL	2956	38	NULL
7	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	48	43130	23732	[redacted]	NULL	NULL	2535	52	NULL
8	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	49	43130	23732	[redacted]	NULL	NULL	2501	53	NULL
9	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	64	43130	23732	[redacted]	NULL	NULL	9336	70	NULL
10	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	66	43130	23732	[redacted]	NULL	NULL	0215	72	NULL
11	Notice	2	PICKERINGTON	OH	USA	1	FAIRFIELD/LICKING/FRANKLIN	Home	87	43147	23745	[redacted]	NULL	NULL	NULL	96	NULL
12	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	89	43130	23732	[redacted]	[redacted]	NULL	8398	98	NULL
13	Notice	2	PICKERINGTON	OH	USA	1	FAIRFIELD/LICKING/FRANKLIN	Home	101	43147	23745	[redacted]	NULL	NULL	NULL	112	NULL
14	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	120	43130	23732	[redacted]	NULL	NULL	NULL	134	NULL
15	Notice	2	PICKERINGTON	OH	USA	1	FAIRFIELD/LICKING/FRANKLIN	Home	153	43147	23745	[redacted]	NULL	NULL	NULL	173	NULL
16	Notice	2	REYNOLDSBURG	OH	USA	1	FRANKLIN	Home	170	43068	23687	[redacted]	NULL	NULL	4156	192	NULL
17	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	175	43130	23732	[redacted]	NULL	NULL	4047	197	NULL
18	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	205	43130	23732	[redacted]	NULL	NULL	2820	234	NULL

Now let's link it to the PatronRegistration table

```
SELECT *

FROM Polaris.Polaris.ViewPatronAddresses vpa

JOIN Polaris.Polaris.PatronRegistration pr on pr.PatronID = vpa.PatronID

WHERE City NOT LIKE 'Columbus' AND AddressTypeID = 2
```

These are shorthand abbreviations to let the database management system know which tables you are referencing. When working with two or more tables, you need to reference each table and these are easiest for humans.

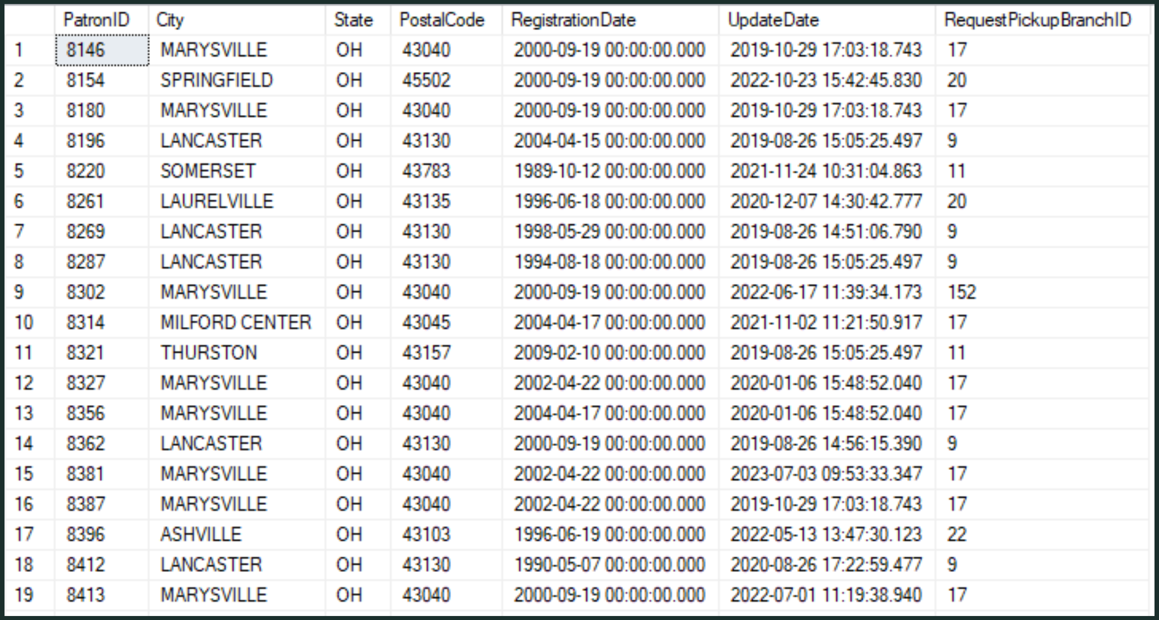


	Description	AddressTypeID	City	State	Country	CountryID	County	FreeTextLabel	PatronID	PostalCode	PostalCodeID	StreetOne	StreetTwo	StreetThree	ZipPlusFour	AddressID	Municipality
1	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	10	43130	23732	[redacted]	[redacted]	NULL	9678	11	NULL
2	Notice	2	MISSING CITY	ZZ	USA	1	MISSING COUNTY	Home	12	99999	0	[redacted]	[redacted]	NULL	NULL	2804243	NULL
3	Notice	2	MARYSVILLE	OH	USA	1	UNION	Home	16	43040	23665	[redacted]	NULL	NULL	NULL	2291858	NULL
4	Notice	2	MARYSVILLE	OH	USA	1	UNION	Home	20	43040	23665	[redacted]	NULL	NULL	NULL	2321651	NULL
5	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	21	43130	23732	[redacted]	NULL	NULL	3227	22	NULL
6	Notice	2	CIRCLEVILLE	OH	USA	1	PICKAWAY	Home	25	43113	23720	[redacted]	NULL	NULL	NULL	26	NULL
7	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	27	43130	23732	[redacted]	NULL	NULL	NULL	28	NULL
8	Notice	2	PICKERINGTON	OH	USA	1	FAIRFIELD/LICKING/FRANKLIN	Home	32	43147	23745	[redacted]	NULL	NULL	NULL	33	NULL
9	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	36	43130	23732	[redacted]	[redacted]	NULL	2956	38	NULL
10	Notice	2	MARYSVILLE	OH	USA	1	UNION	Home	37	43040	23665	[redacted]	NULL	NULL	NULL	3109415	NULL
11	Notice	2	MARYSVILLE	OH	USA	1	UNION	Home	40	43040	23665	[redacted]		NULL	NULL	277249	NULL
12	Notice	2	MARYSVILLE	OH	USA	1	UNION	Home	44	43040	23665	[redacted]		NULL	NULL	284173	NULL
13	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	48	43130	23732	[redacted]	NULL	NULL	2535	52	NULL
14	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	49	43130	23732	[redacted]	NULL	NULL	2501	53	NULL
15	Notice	2	CIRCLEVILLE	OH	USA	1	PICKAWAY	Home	59	43113	23720	[redacted]		NULL	NULL	314192	NULL
16	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	62	43130	23732	[redacted]	NULL	NULL	NULL	2471017	NULL
17	Notice	2	LANCASTER	OH	USA	1	FAIRFIELD	Home	64	43130	23732	[redacted]	NULL	NULL	9336	70	NULL
18	Notice	2	PLAIN CITY	OH	USA	1	UNION	Home	65	43064	56563	[redacted]	NULL	NULL	NULL	4902471	NULL

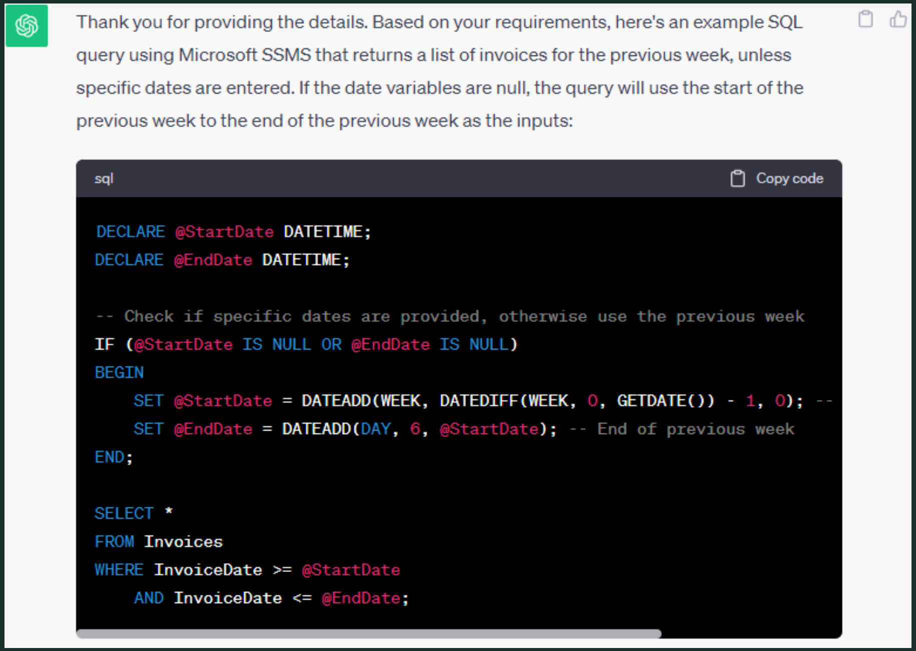
That is too much horizontal scroll. Let’s par it down to something useable

```
SELECT vpa.PatronID, vpa.City, vpa.State, vpa.PostalCode, pr.RegistrationDate,
pr.UpdateDate, pr.RequestPickupBranchID

FROM Polaris.Polaris.ViewPatronAddresses vpa

JOIN Polaris.Polaris.PatronRegistration pr on pr.PatronID = vpa.PatronID

WHERE City NOT LIKE 'Columbus' AND AddressTypeID = 2
```

We also put the abbreviations on the column names to let the system know exactly which columns to return



	PatronID	City	State	PostalCode	RegistrationDate	UpdateDate	RequestPickupBranchID
1	8146	MARYSVILLE	OH	43040	2000-09-19 00:00:00.000	2019-10-29 17:03:18.743	17
2	8154	SPRINGFIELD	OH	45502	2000-09-19 00:00:00.000	2022-10-23 15:42:45.830	20
3	8180	MARYSVILLE	OH	43040	2000-09-19 00:00:00.000	2019-10-29 17:03:18.743	17
4	8196	LANCASTER	OH	43130	2004-04-15 00:00:00.000	2019-08-26 15:05:25.497	9
5	8220	SOMERSET	OH	43783	1989-10-12 00:00:00.000	2021-11-24 10:31:04.863	11
6	8261	LAURELVILLE	OH	43135	1996-06-18 00:00:00.000	2020-12-07 14:30:42.777	20
7	8269	LANCASTER	OH	43130	1998-05-29 00:00:00.000	2019-08-26 14:51:06.790	9
8	8287	LANCASTER	OH	43130	1994-08-18 00:00:00.000	2019-08-26 15:05:25.497	9
9	8302	MARYSVILLE	OH	43040	2000-09-19 00:00:00.000	2022-06-17 11:39:34.173	152
10	8314	MILFORD CENTER	OH	43045	2004-04-17 00:00:00.000	2021-11-02 11:21:50.917	17
11	8321	THURSTON	OH	43157	2009-02-10 00:00:00.000	2019-08-26 15:05:25.497	11
12	8327	MARYSVILLE	OH	43040	2002-04-22 00:00:00.000	2020-01-06 15:48:52.040	17
13	8356	MARYSVILLE	OH	43040	2004-04-17 00:00:00.000	2020-01-06 15:48:52.040	17
14	8362	LANCASTER	OH	43130	2000-09-19 00:00:00.000	2019-08-26 14:56:15.390	9
15	8381	MARYSVILLE	OH	43040	2002-04-22 00:00:00.000	2023-07-03 09:53:33.347	17
16	8387	MARYSVILLE	OH	43040	2002-04-22 00:00:00.000	2019-10-29 17:03:18.743	17
17	8396	ASHVILLE	OH	43103	1996-06-19 00:00:00.000	2022-05-13 13:47:30.123	22
18	8412	LANCASTER	OH	43130	1990-05-07 00:00:00.000	2020-08-26 17:22:59.477	9
19	8413	MARYSVILLE	OH	43040	2000-09-19 00:00:00.000	2022-07-01 11:19:38.940	17



Thank you for providing the details. Based on your requirements, here's an example SQL query using Microsoft SSMS that returns a list of invoices for the previous week, unless specific dates are entered. If the date variables are null, the query will use the start of the previous week to the end of the previous week as the inputs:

```sql
DECLARE @StartDate DATETIME;
DECLARE @EndDate DATETIME;

-- Check if specific dates are provided, otherwise use the previous week
IF (@StartDate IS NULL OR @EndDate IS NULL)
BEGIN
    SET @StartDate = DATEADD(WEEK, DATEDIFF(WEEK, 0, GETDATE()) - 1, 0); --
    SET @EndDate = DATEADD(DAY, 6, @StartDate); -- End of previous week
END;

SELECT *
FROM Invoices
WHERE InvoiceDate >= @StartDate
    AND InvoiceDate <= @EndDate;
```

ChatGPT Knows It All

ChatGPT can help you…

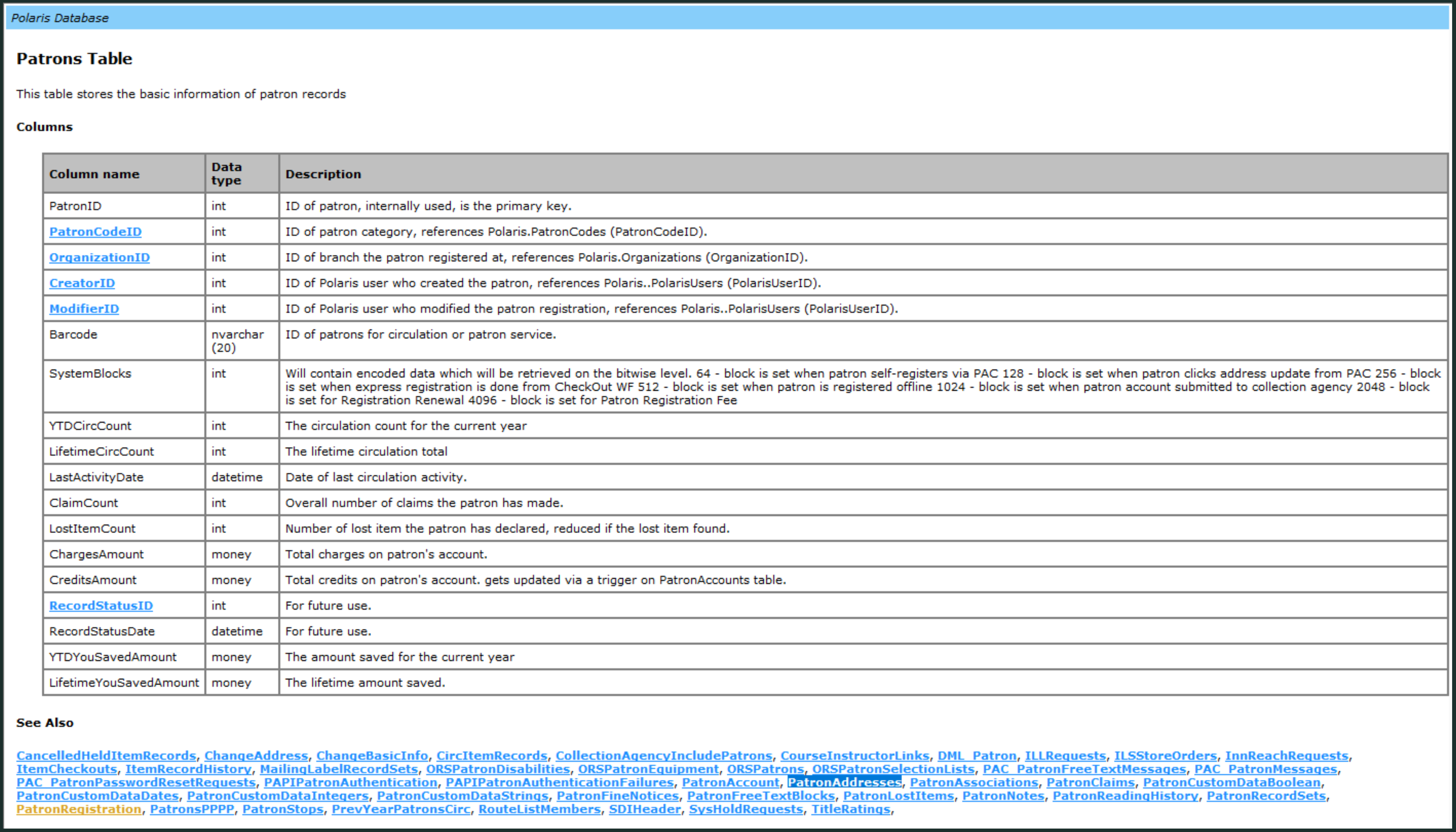
- → Understand SQL when you are confused
- → Work out some tricky syntax
- → Format your SQL so other people can have an easier time reading it
- → Understand SQL someone else has written

ChatGPT cannot help you…

→ Know which table to use in the Polaris database

→ Know what column names are in the Polaris database

Polaris Database Repository (it will help you)



Polaris Database

Patrons Table

This table stores the basic information of patron records

Columns

Column name	Data type	Description
PatronID	int	ID of patron, internally used, is the primary key.
PatronCodeID	int	ID of patron category, references Polaris.PatronCodes (PatronCodeID).
OrganizationID	int	ID of branch the patron registered at, references Polaris.Organizations (OrganizationID).
CreatorID	int	ID of Polaris user who created the patron, references Polaris..PolarisUsers (PolarisUserID).
ModifierID	int	ID of Polaris user who modified the patron registration, references Polaris..PolarisUsers (PolarisUserID).
Barcode	nvarchar (20)	ID of patrons for circulation or patron service.
SystemBlocks	int	Will contain encoded data which will be retrieved on the bitwise level. 64 - block is set when patron self-registers via PAC 128 - block is set when patron clicks address update from PAC 256 - block is set when express registration is done from CheckOut WF 512 - block is set when patron is registered offline 1024 - block is set when patron account submitted to collection agency 2048 - block is set for Registration Renewal 4096 - block is set for Patron Registration Fee
YTDCircCount	int	The circulation count for the current year
LifetimeCircCount	int	The lifetime circulation total
LastActivityDate	datetime	Date of last circulation activity.
ClaimCount	int	Overall number of claims the patron has made.
LostItemCount	int	Number of lost item the patron has declared, reduced if the lost item found.
ChargesAmount	money	Total charges on patron's account.
CreditsAmount	money	Total credits on patron's account. gets updated via a trigger on PatronAccounts table.
RecordStatusID	int	For future use.
RecordStatusDate	datetime	For future use.
YTDYouSavedAmount	money	The amount saved for the current year
LifetimeYouSavedAmount	money	The lifetime amount saved.

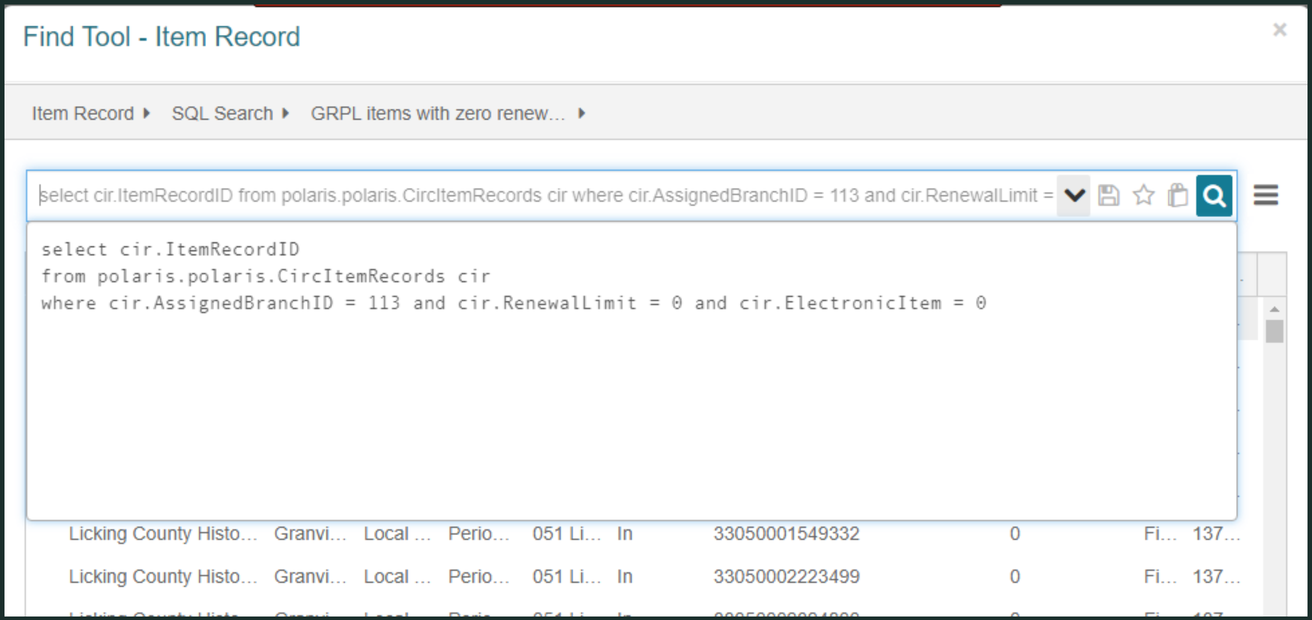
See Also

CancelledHeldItemRecords, ChangeAddress, ChangeBasicInfo, CircItemRecords, CollectionAgencyIncludePatrons, CourseInstructorLinks, DML_Patron, ILLRequests, ILSStoreOrders, InnReachRequests, ItemCheckouts, ItemRecordHistory, MailingLabelRecordSets, ORSPatronDisabilities, ORSPatronEquipment, ORSPatrons, ORSPatronSelectionLists, PAC_PatronFreeTextMessages, PAC_PatronMessages, PAC_PatronPasswordResetRequests, PAPIPatronAuthentication, PAPIPatronAuthenticationFailures, PatronAccount, PatronAddresses, PatronAssociations, PatronClaims, PatronCustomDataBoolean, PatronCustomDataDates, PatronCustomDataIntegers, PatronCustomDataStrings, PatronFineNotices, PatronFreeTextBlocks, PatronLostItems, PatronNotes, PatronReadingHistory, PatronRecordSets, PatronRegistration, PatronsPPPP, PatronStops, PrevYearPatronsCirc, RouteListMembers, SDIHeader, SysHoldRequests, TitleRatings,

Find Tool Applications

→ Did you know you can save SQL to the Polaris Find Tool for anyone to use?

→ All you have to do is replace SELECT * with SELECT [insert appropriate ID type here]



select cir.ItemRecordID from polaris.polaris.CircItemRecords cir where cir.AssignedBranchID = 113 and cir.RenewalLimit =

```
select cir.ItemRecordID
from polaris.polaris.CircItemRecords cir
where cir.AssignedBranchID = 113 and cir.RenewalLimit = 0 and cir.ElectronicItem = 0
```

Licking County Histo…	Granvi…	Local …	Perio…	051 Li…	In	33050001549332	0	Fi…	137…
Licking County Histo…	Granvi…	Local …	Perio…	051 Li…	In	33050002223499	0	Fi…	137…

Other Resources

- https://www.w3schools.com/sql/default.asp
  - W3 Schools online resource describing SQL commands and how to use them
- https://forum.innovativeusers.org/c/clearinghouse-repository/polaris-repository/22
  - Shared SQL reports from other Polaris libraries
- IUG Discord
  - Helpful group with members who can help you troubleshoot
- https://iii.rightanswers.com/portal/app/portlets/results/viewsolution.jsp?solutionid=230406121425783
  - Polaris Database Repository (slide 44)
  - Requires a support log in, if you don't have one, contact either CLC or your Innovative rep