

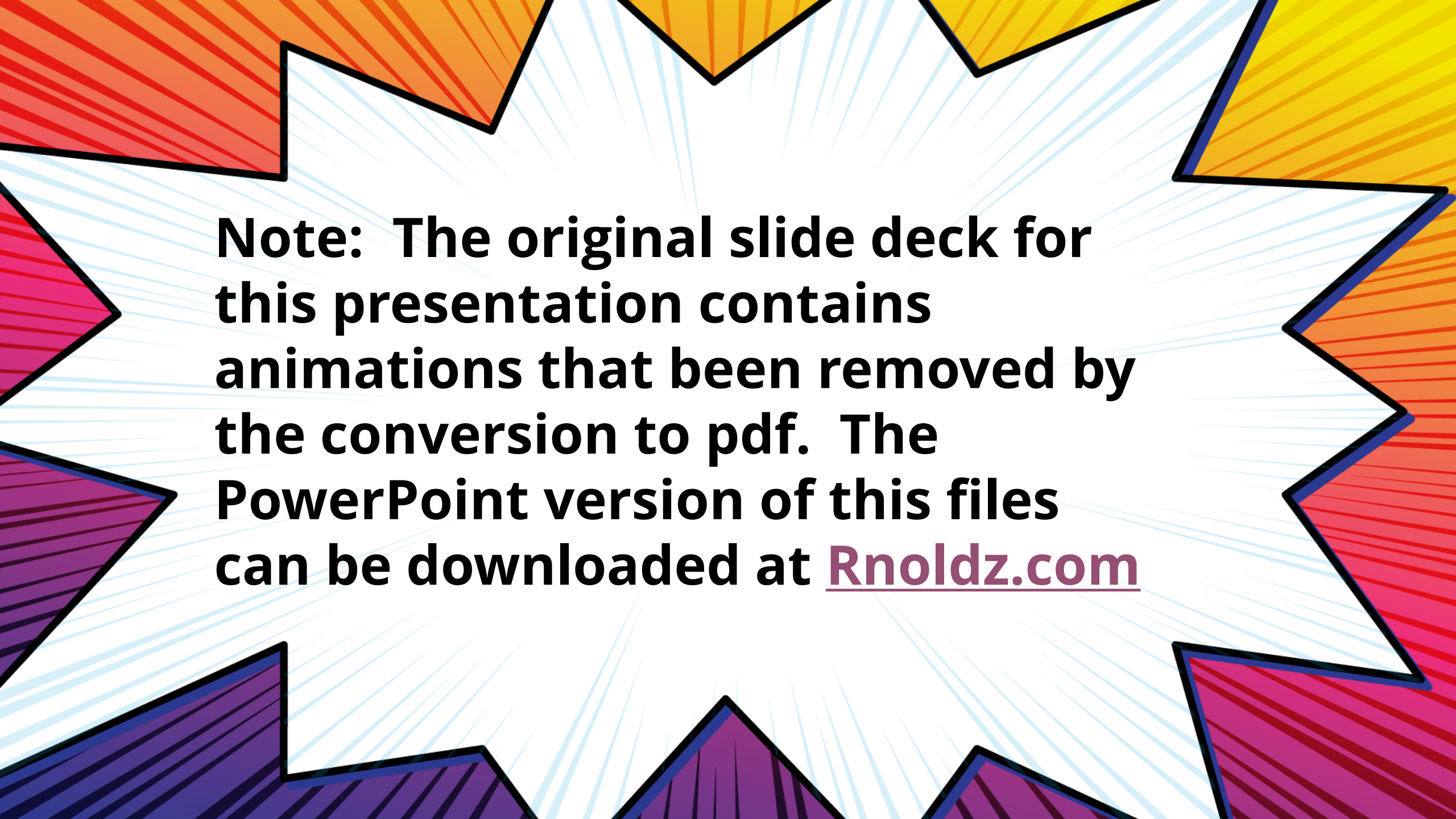
Databases and Queries and SQL, OH MY... Demystified!



Databases and Queries and SQL, OH MY... Demystified!

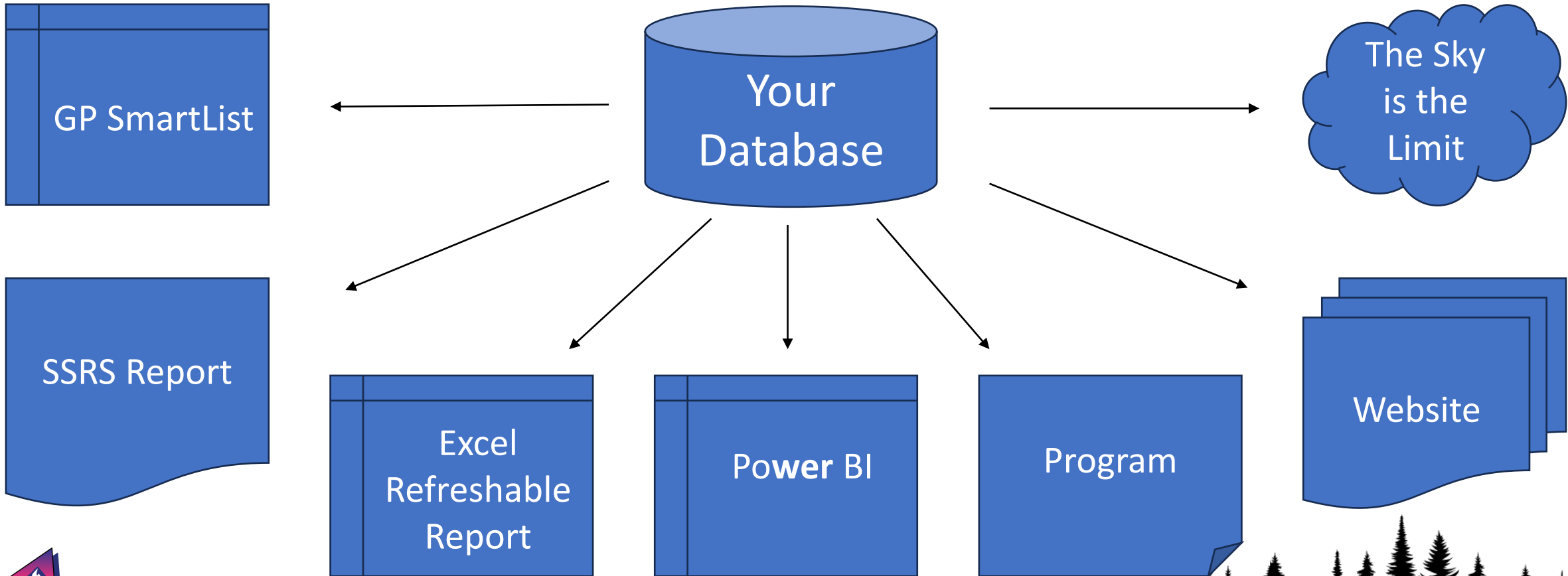
John Arnold
Senior Software Engineer
US Digital Corporation
John_P_Arnold@hotmail.com
Rnoldz.com



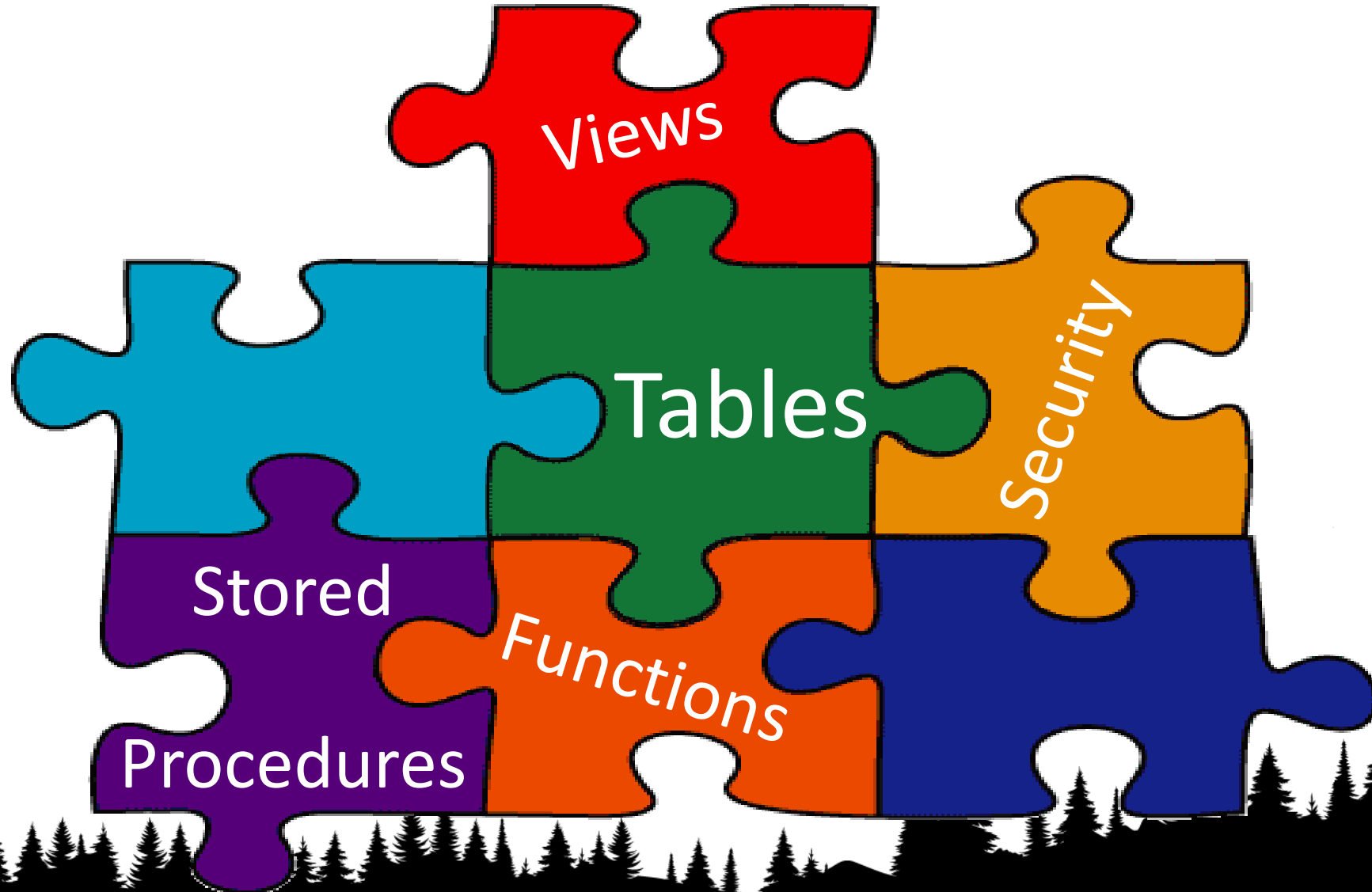


Note: The original slide deck for this presentation contains animations that been removed by the conversion to pdf. The PowerPoint version of this files can be downloaded at Rnoldz.com

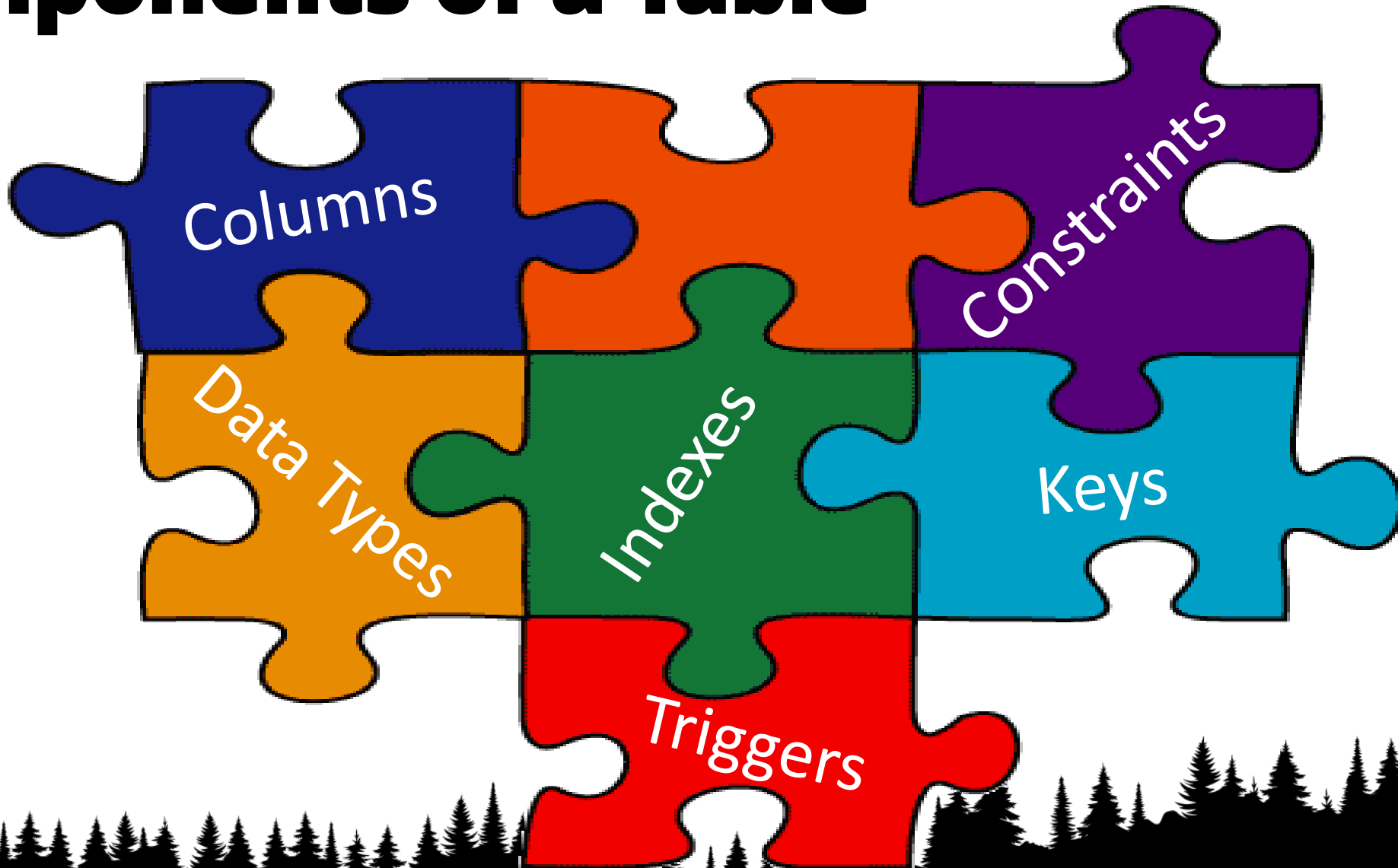
Querying Databases



Components of a Database



Components of a Table



In the old days...

- Every database had its own way of talking
- Dynamics GP originally ran with C-Tree database program – 1993
- 1997, GP was purchased by Microsoft and now runs with MS-SQL
 - C-Tree column names used by MS-SQL version
 - ATYALLOC - Why isn't it QTYALLOC?
- SQL provides a “standard” way to communicate with databases
 - ANSI SQL
 - T-SQL is Microsoft SQL server specific



DML – Data Manipulation Language

- We are going to focus on SELECT statement
- A SELECT statement queries one or more databases and returns data (result set) to the calling application (GP, Excel, SSRS...)
- Simple example

```
SELECT 'Hello World'
```



The **SELECT** Statement

- SELECT - The column(s) of data you want returned
- FROM - The place the data lives.
- JOIN - Optionally specify additional locations to get data
- WHERE - Filter the data returned in the record set
- GROUP BY - Combine multiple records into one record
- HAVING - Filter the combined data
- UNION - Merge two record sets into one
- ORDER BY - Specify the sort order of the rows returned



Example: Sales Report

Request: “I need a GP report that will show our sales orders that are more than \$1,000”

What is really wanted:

Show all Order Numbers, Order Dates, Customer Numbers, Customer's Current Names, and Total Prices

Of non-service Items

Where the Total Order is more than \$1,000

On all non-voided Orders

Work and Historical Orders

Over a specified date range!



Sales Report Source of Data

Tables Needed

- SOP10100 Sales Header Work
- SOP10200 Sales Line Work
- RM00101 Customer Master
- IV00101 Inventory Master
- SOP30200 Sales Header History
- SOP30300 Sales Line History



GP Table and Column Names

How to figure out table and column names in GP

- Wednesday, May 14th @ 1:15 PM – Chris Giesbrecht – “How is my GP data stored? SQL Tables 101”
- Wednesday, May 14th @ 2:45 PM – Amber Bell and John Arnold – “GP, SQL views, and Macros... optimizing your system is easier than ever!”
- VictoriaYudin.com – Everything you want to know about GP tables
- SmartLists – Every GP SmartList is driven by a view with the same name
- GP Add-ins – Give table & field names of objects on the screen
- Google it with Bing – Many blogs and forums list Tables & Columns names

I recommend keeping quick reference of the tables & columns used



Common Data Types

INT	Whole numbers
NUMERIC(19,5)	Fixed position and scale – Floating point numbers
DATETIME	Store dates and times. The time in GP is usually = 0
CHAR(LEN)	Text strings – padded with spaces - used by GP
VARCHAR(LEN)	Text strings – not padded with spaces

Standard installs of SQL Server are case insensitive making 'A' = 'a'

Use RTRIM(ColumnName) to remove trailing spaces



Start the Query

```
SELECT CUSTNMBR, CUSTNAME, SOPNUMBE, DOCDATE FROM SOP10100 AS S  
WHERE SOPTYPE = 2 AND VOIDSTTS = 0 AND DOCDATE BETWEEN '1/1/2023' AND  
'1/1/2025'
```



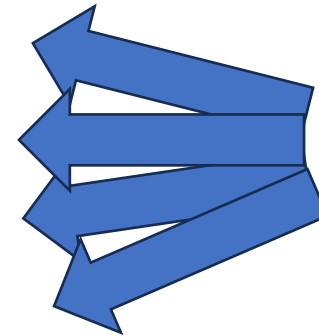
Fix the Query

- A separate line for every item
- Prefix ALL column names with their table name
- Capitalize all reserved words
- Put () around all AND & OR statements
DateSold > '1/1/2024' AND ItemSold = 'Grapes' OR ItemSold = 'Oranges'
(DateSold > '1/1/2024') AND ((ItemSold = 'Grapes') OR (ItemSold = 'Oranges'))
- Don't alias table names
- RTRIM necessary (CHAR) columns
- Format the query so it is consistent and readable



The Fixed Query

```
SELECT  RTRIM(SOP10100.CUSTNMBR) AS CUSTNMBR,  
        RTRIM(SOP10100.CUSTNAME) AS CUSTNAME,  
        RTRIM(SOP10100.SOPNUMBE) AS SOPNUMBE,  
        SOP10100.DOCDATE
```



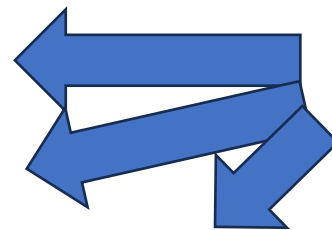
Return these columns

```
FROM    SOP10100
```



From the SOP Header Table

```
WHERE   (SOP10100.SOPTYPE = 2)
```



Filter on these three things

```
AND (SOP10100.VOIDSTTS = 0)
```

```
AND (SOP10100.DOCDATE BETWEEN '1/1/2023' AND '1/1/2025')
```



JOIN – Bringing Tables Together

The JOIN statement allows you to LINK data between two tables

We will focus on three types of joins:

INNER JOIN

LEFT/RIGHT JOIN

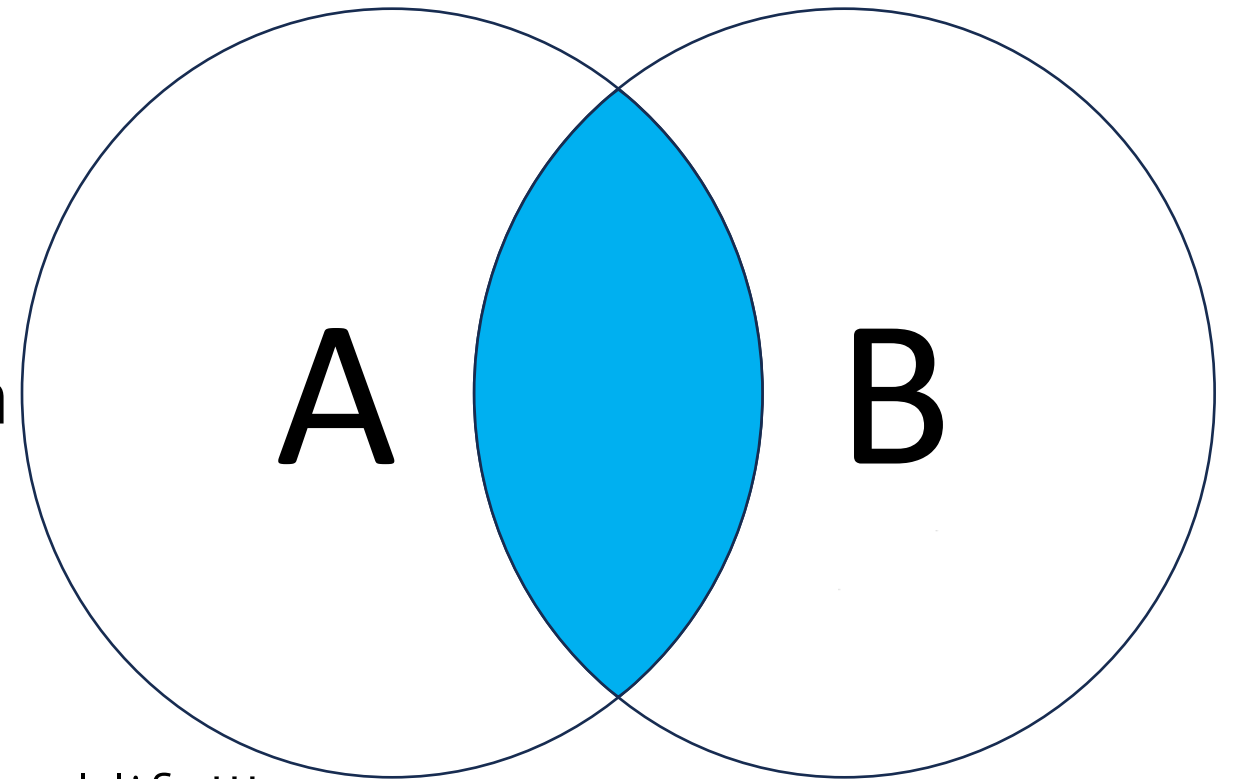
FULL JOIN



JOIN - INNER

FROM A LEFT JOIN B

Return: Everything from Table A
that matches records from
Table B



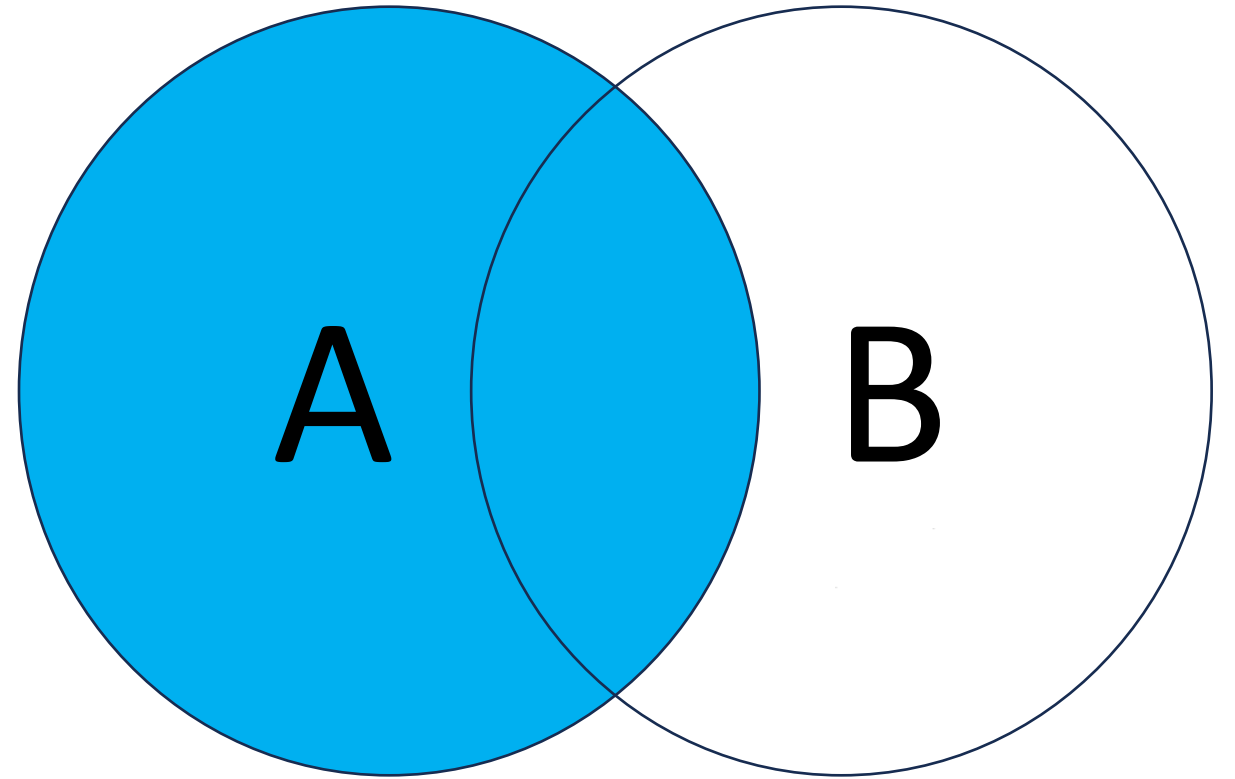
Venn Diagrams... why do I have to
learn this stuff? I'll never use it in real life!!!



LEFT JOIN - OUTER

FROM A LEFT JOIN B

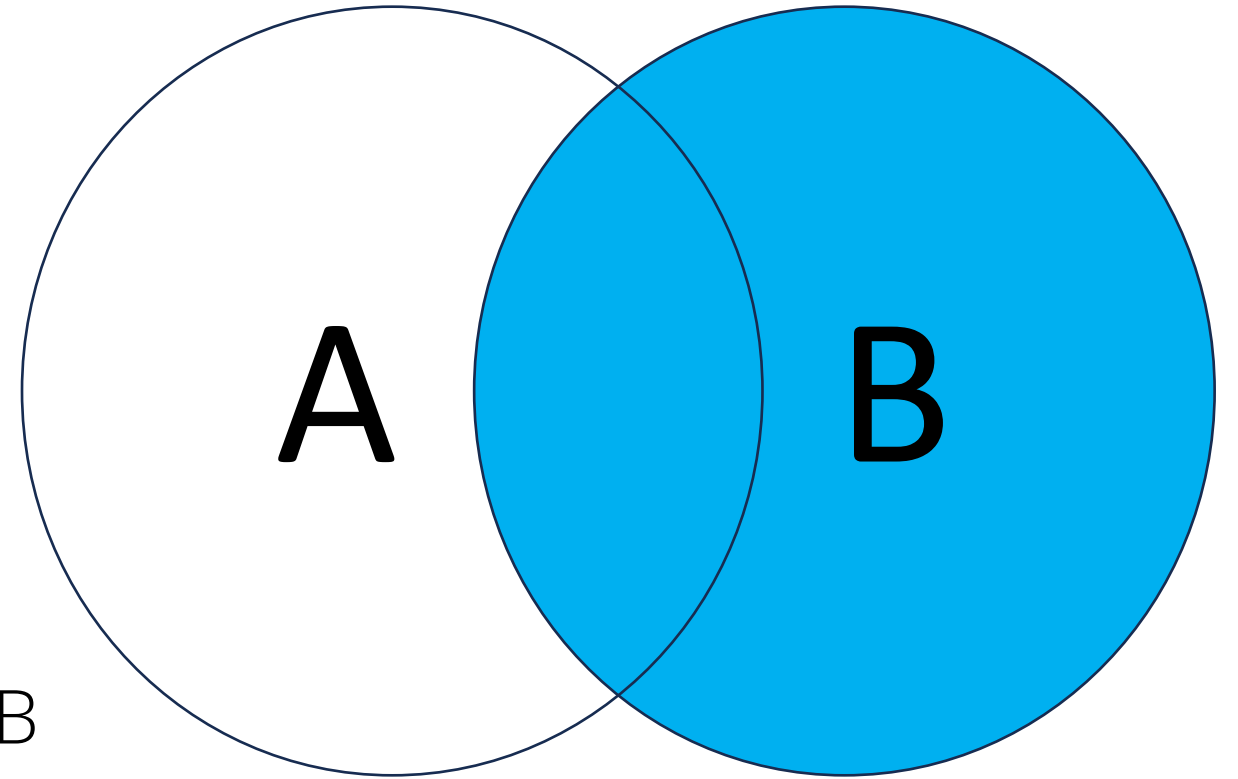
Return: Everything from Table A
and the matching records
from B or NULL



RIGHT JOIN – OUTER – Try to avoid!

FROM A RIGHT JOIN B

Return: Everything from Table B
and the matching records
from A or NULL



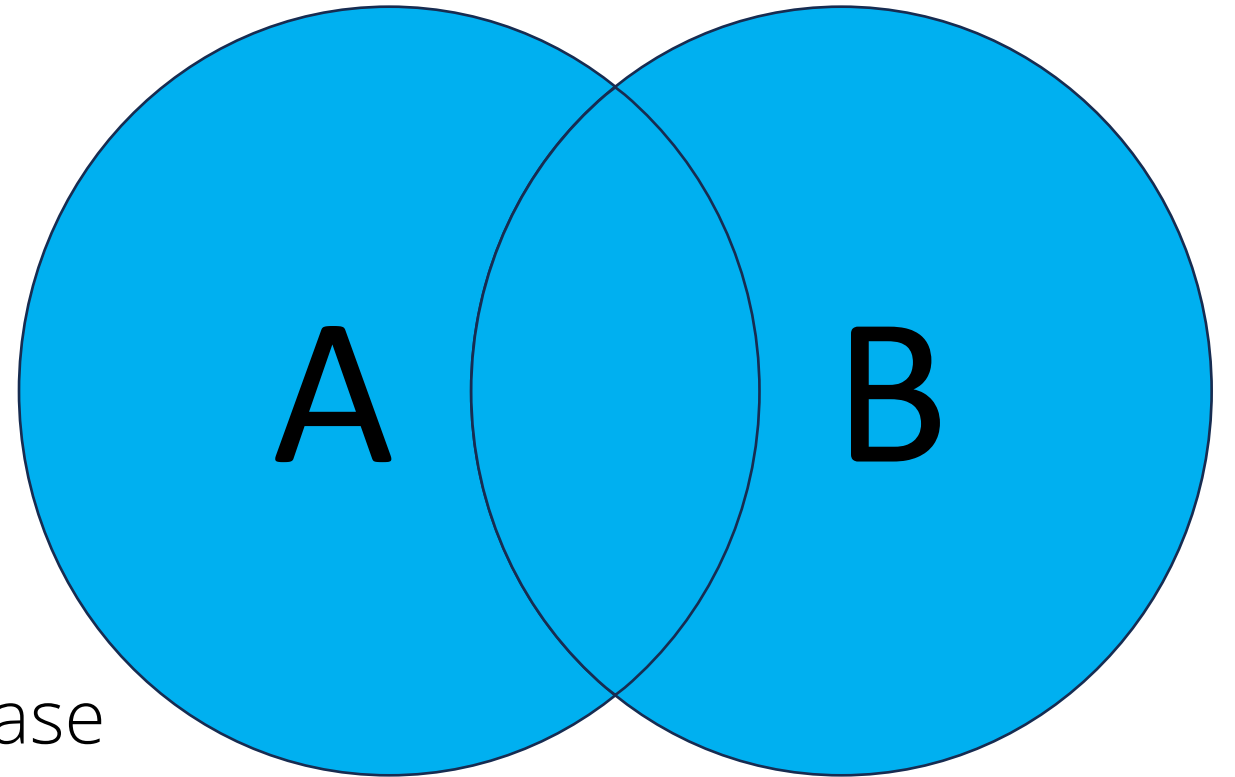
Avoid the RIGHT JOIN, swap A and B
and make it a LEFT JOIN!



FULL JOIN - OUTER

FROM A FULL JOIN B

Return: Everything from Table A
and B and NULL
where no matches exist

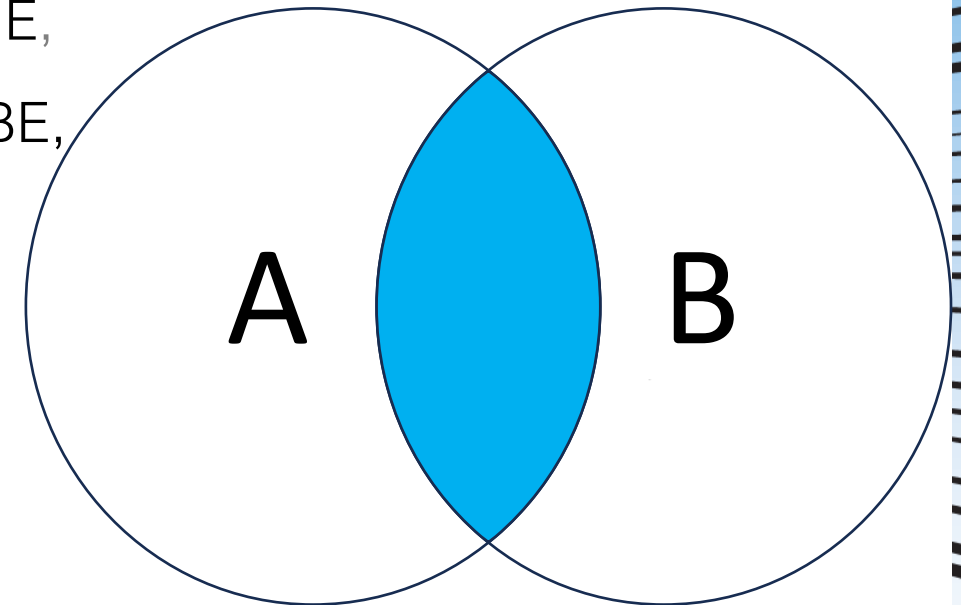


Example – Show me the last purchase
date from all open and historical Purchase Orders



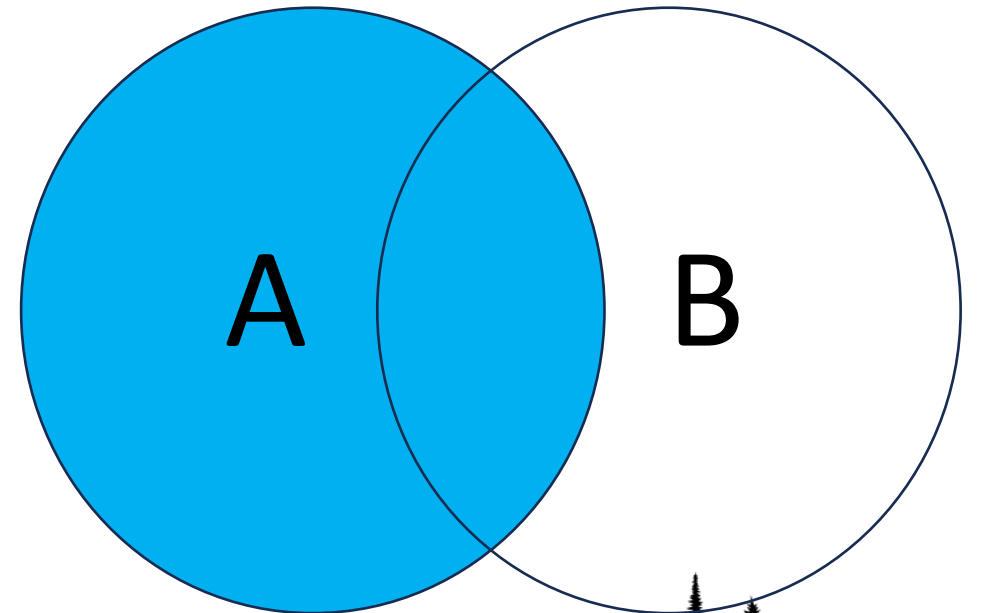
JOIN the Query to the Customer Table

```
SELECT  RTRIM(SOP10100.CUSTNMBR) AS CUSTNMBR,  
        RTRIM(SOP10100.CUSTOMNAME) AS CUSTOMNAME,  
        RTRIM(SOP10100.SOPNUMBE) AS SOPNUMBE,  
        SOP10100.DOCDATE  
FROM    SOP10100  
FROM    RM10100  
JOIN    (SOP10100.SOPTYPE = 2)  
WHERE  ON  RM10100.CUSTNMBR = SOP10100.CUSTNMBR  
WHERE  AND (SOP10100.VOIDSTIS = 0)  
        AND (SOP10100.SOPTYPE = 2)  
        AND (SOP10100.DOCDATE BETWEEN '1/1/2023' AND '1/1/2025')  
        AND (SOP10100.VOIDSTIS = 0)  
        AND (SOP10100.DOCDATE BETWEEN '1/1/2023' AND '1/1/2025')
```



Handle Deleted Customers

```
SELECT    RTRIM(SOP10100.CUSTNMBR) AS CUSTNMBR,  
          RTRIM(ISNULL(RM00101.CUSTNAME, SOP10100.CUSTNAME)) AS CUSTNAME,  
          RTRIM(SOP10100.SOPNUMBE) AS SOPNUMBE,  
          SOP10100.DOCDATE  
FROM      SOP10100  
LEFT JOIN RM10100  
          ON  RM10100.CUSTNMBR = SOP10100.CUSTNMBR  
WHERE     (SOP10100.SOPTYPE = 2)  
          AND (SOP10100.VOIDSTTS = 0)  
          AND (SOP10100.DOCDATE BETWEEN '1/1/2023' AND '1/1/2025')
```



NULL – Unknown Data

- NULL is missing or unknown data
- NULL can appear from missing data in a table or from OUTER joins
- Comparing anything to NULL returns a NULL / False
- $1 = \text{NULL}$ and $1 \neq \text{NULL}$ both fail and both return False
- Use the syntax:
 - ColumnName IS NULL
 - ColumnName IS NOT NULL
- Use the **ISNULL**(TestColumn, ValueIfTestColumnIsNull) function



GROUP BY

- Combine multiple records into one record
- Only columns specified in the GROUP BY or aggregate functions (SUM, COUNT, MIN, MAX, AVERAGE...) can be returned by a SELECT

Order	Date	Item	Quantity
10001	1/2/2024	Oranges	10
10020	1/2/2024	Apples	12
10020	1/5/2024	Oranges	6
10037	1/7/2024	Apples	24

```
SELECT Item,  
SUM(Quantity) AS Total  
FROM Sales  
GROUP BY Item
```

Item	Total
Oranges	16
Apples	36



GROUP BY Continued

Order	Date	Item	Quantity
10001	1/2/2024	Oranges	10
10020	1/2/2024	Apples	12
10020	1/5/2024	Oranges	6
10037	1/7/2024	Apples	24



Item	Total
Oranges	16
Apples	24

- Add the Date column to the SELECT to see when the order date
- You can't:

Column 'Date' is invalid in the select list because it is not contained in either an aggregate function or the GROUP BY clause.

GROUP BY and HAVING example

```
SELECT      SOP10100.SOPNUMBE ,
            SUM(SOP10200.XTNDPRCE) AS OrderTotal
FROM        SOP10100
JOIN        SOP10200
            ON  SOP10200.SOPNUMBE = SOP10100.SOPNUMBE
            AND SOP10200.SOPTYPE = SOP10100.SOPTYPE
WHERE       (SOP10100.SOPTYPE = 2)
            AND (SOP10100.VOIDSTTS = 0)
            AND (SOP10100.DOCDATE BETWEEN '1/1/2023' AND '1/1/2025')
GROUP BY   SOP10100.SOPNUMBE
HAVING     SUM(SOP10200.XTNDPRCE) > 1000
```



UNION

Use the UNION statement to combine work with history tables

The number of columns must match

The data types of each column must match

UNION – performs a DISTINCT and removes duplicate rows

UNION ALL – keeps all rows in the record set, including duplicates

Column names are only needed on the first record set



UNION example

Combine multiple record sets (four) into one record set

SELECT 1 AS Data

UNION

SELECT 2

UNION

SELECT 2

UNION

SELECT 3



Data
1
2
3

VIEWS and Virtual Tables

Tables that are created by a query

Only exist while a query is running

JOINED to like a regular table

Can be useful to simplify GROUP BYs

Virtual Tables operate that same as a view and are interchangeable



VIEW example

```
CREATE VIEW AS GenderTypeView
AS
SELECT      'F' AS Type,
            'Female' AS Description

UNION

SELECT      'M',
            'Male'
```



VIEW example continued

```
SELECT    Person.Name,  
          GenderTypeView.Description  
FROM      Person  
JOIN      GenderTypeView  
ON        GenderTypeView.Type = Person.Type
```



Virtual Table Example

```
SELECT      Person.Name,  
            GenderType.Description  
FROM        Person  
JOIN        (  
            SELECT      'F' AS Type,  
                        'Female' AS Description  
  
            UNION  
  
            SELECT      'M',  
                        'Male'  
            ) AS GenderType  
ON          GenderType.Type = Person.Type
```

SELECT, JOIN, UNION, GROUP BY...

```
SELECT Sales.SOPNUMBE,  
       SUM(Sales.XTNDPRCE) AS OrderTotal  
FROM (  
    SELECT SOP10100.SOPNUMBE,  
           SOP10100.SOPTYPE,  
           SOP10200.ITEMNMBR,  
           SOP10200.XTNDPRCE  
    FROM   SOP10100  
    JOIN   SOP10200  
    ON     SOP10200.SOPNUMBE =  
           SOP10100.SOPNUMBE  
    AND    SOP10200.SOPTYPE =  
           SOP10100.SOPTYPE  
    WHERE  SOP10100.DOCDATE  
           BETWEEN '1/1/2023'  
           AND '1/1/2025'  
    AND    SOP10100.VOIDSTTS = 0  
    UNION ALL
```

```
SELECT SOP30200.SOPNUMBE,  
       SOP30200.SOPTYPE,  
       SOP30300.ITEMNMBR,  
       SOP30300.XTNDPRCE  
FROM   SOP30200  
JOIN   SOP30300  
ON     SOP30300.SOPNUMBE =  
       SOP30200.SOPNUMBE  
AND    SOP30300.SOPTYPE =  
       SOP30200.SOPTYPE  
WHERE  SOP30200.DOCDATE  
       BETWEEN '1/1/2023'  
       AND '1/1/2025'  
AND    SOP30200.VOIDSTTS = 0  
       ) Sales  
JOIN   IV00101  
ON     IV00101.ITEMNMBR = Sales.ITEMNMBR  
WHERE  IV00101.ITEMTYPE <> 4  
GROUP BY Sales.SOPNUMBE  
HAVING SUM(Sales.XTNDPRCE) > 1000
```

The "Final" Query

```

SELECT
Sales.SOPNUMBE AS OrderNumber,
MIN(Sales.DOCDATE) AS OrderDate,
MIN(RTRIM(Sales.CUSTNMBR)) AS CustomerNumber,
MIN(RTRIM(ISNULL(RM00101.CUSTNAME, Sales.CUSTNAME)))
AS CustomerName,
SUM(Sales.XTNDPRCE) AS TotalPrice,
MIN(Source) AS Source
FROM
(

```

```

SELECT SOP10100.SOPNUMBE,
SOP10100.DOCDATE,
SOP10100.CUSTNMBR,
SOP10100.CUSTNAME,
SOP10200.ITEMNMBR,
SOP10200.XTNDPRCE,
'W' AS Source
FROM SOP10100
JOIN SOP10200
ON SOP10200.SOPNUMBE = SOP10100.SOPNUMBE
AND SOP10200.SOPTYPE = SOP10100.SOPTYPE
WHERE (SOP10100.SOPTYPE = 2)
AND (SOP10100.VOIDSTTS = 0)
AND (SOP10100.DOCDATE BETWEEN '1/1/2023'
AND '1/1/2025')

```

UNION ALL

Orders (work and history)

Over a specified
Date Range

On all non-voided

Show all Order Numbers, Order Date, Customer
Number, Customer's Current Names, and Total
Prices

```

SELECT SOP30200.SOPNUMBE,
SOP30200.DOCDATE,
SOP30200.CUSTNMBR
FROM SOP30200
JOIN SOP30300
ON SOP30300.SOPNUMBE = SOP30200.SOPNUMBE
AND SOP30300.SOPTYPE = SOP30200.SOPTYPE
WHERE (SOP30200.SOPTYPE = 2)
AND (SOP30200.VOIDSTTS = 0)
AND (SOP30200.DOCDATE BETWEEN '1/1/2023'
AND '1/1/2025')
) Sales
IV00101
IV00101.ITEMNMBR = Sales.ITEMNMBR
AND Sales.CUSTNMBR =
s.CUSTNMBR
WHERE
GROUP BY Sales.SOPNUMBE
HAVING SUM(Sales.XTNDPRCE) > 1000

```

of Non-Service Items

more than \$1,000

WHERE tips (Table names removed)

WHERE (X=10) OR (X=17)

WHERE X IN (10, 17)

WHERE (X=10) OR (Y=10)

WHERE 10 IN (X, Y)

WHERE (DT >= '1/1/2024')

WHERE DT BETWEEN

AND (DT <= '12/1/2024')

'1/1/2024' AND '12/1/2024'

WHERE ABS(Amount) > 1000

WHERE (Amount < 1000)

OR (Amount > 1000)

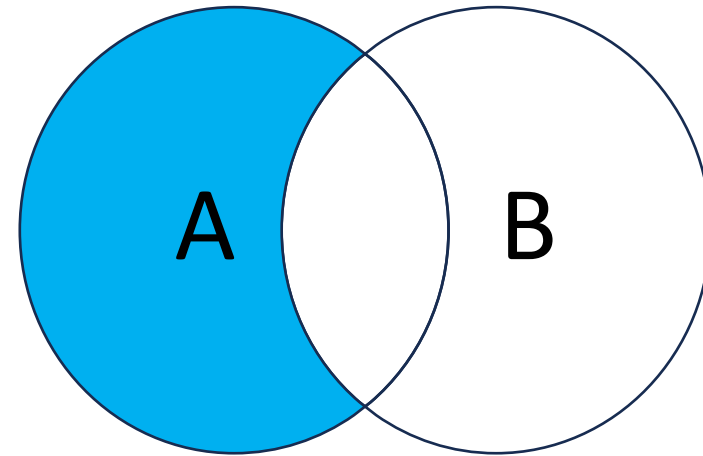
WHERE COL LIKE '_Some%Value%' – Wild card searches

WHERE COL = 'CaseMatters' COLLATE SQL_Latin1_General_CP

-- Case Sensitive Comparisons



WHERE tips



```
SELECT A.X
FROM A
WHERE A.X NOT IN
(
    SELECT B.X
    FROM B
)
```

```
SELECT A.X
FROM B
LEFT JOIN A
    ON B.X = B.X
WHERE B.X IS NULL

ANOTHER Venn diagram?? Really?
```



GROUP BY tips

```
SELECT OrderNumber,  
       Customer,  
       COUNT(*) AS Orders  
FROM   Table1  
GROUP BY OrderNumber,  
         Customer
```

```
SELECT OrderNumber,  
       MIN(Customer) AS  
         Customer,  
       COUNT(*) AS Orders  
FROM   Table1  
GROUP BY OrderNumber
```



HAVING tips

GROUP BY SOPNUMBE

HAVING MIN(CUSTNMBR) <> MAX(CUSTNMBR)

Returns records where CUSTNMBR is not being GROUPed BY and you assume it would have the same value for the entire group, but it doesn't. This example should return 0 rows.



System Views tips

```
SELECT name
FROM sys.tables
WHERE name LIKE 'IVR%5' -- to find the IVR0015 table
```

```
SELECT sys.tables.name
FROM sys.tables
JOIN sys.columns
ON sys.columns.object_id = sys.tables.object_id
WHERE sys.columns.name = 'SOPNUMBE' -- found in 106 tables
```



SSMS tips

Keep your fingers on the keyboard and off the mouse

<https://learn.microsoft.com/en-us/sql/ssms/sql-server-management-studio-keyboard-shortcuts>

Highlight text to only execute that part of the query

Shift + Alt + Enter to make the window full screen

Ctrl + R to hide/show the results window

Alt + ↑ or Alt + ↓ to move the current line or block up or down

Alt key while highlighting to highlight a rectangle of text

Right click on the object window item and Filter the results

Ctrl + T to have query results as searchable text! Ctrl + D for grid



Views vs Stored Procedures

Views

- Pro – Used by SmartLists
- Pro – A View is a Virtual Table
- Pro – Great for reusable query
- Pro – Can hide query complexity
- Con – Can hide query complexity
- Con – Evaluated at run time

Stored Procedures

- Pro – Compiled
- Pro – Can take parameters
- Pro – Code (Variables, Looping...)
- Pro – GREAT source for Excel data
- Con – Compiled, indexes change
- Con – Not supported by SmartLists



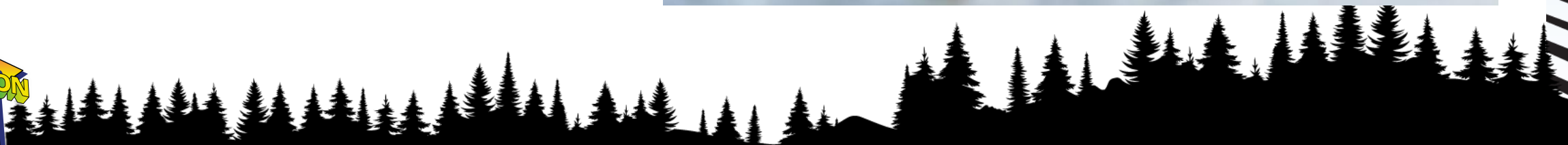
Resources

- GP Views (SmartLists)
- PoorSQL.com (to format SQL code)
- VictoriaYudin.com
- Rnoldz.com
- User Groups
- Dr Google



Go write a query!

Get your hands
dirty and just do it!!!



Thanks for Attending!

Questions or Help?

John_P_Arnold@hotmail.com

