

WHERE Clause

WHERE Clause Review

- Use WHERE to identify a subset of data which meets a certain condition (filtering condition)
 - DBMS goes through each row and applies SQL action on the rows that meet the WHERE condition.

```
SELECT * FROM <table_name>  
WHERE <filtering condition>
```

WHERE Clause Review

- Filtering condition can include:
 - Original columns
 - Computed columns

```
SELECT * FROM Registration  
WHERE Actual_Tuition * 1.1 > 500
```

WHERE Clause: Comparison

- Compare equations

=

<>

!=

>

>=

<

<=



Logical Operator

- AND/OR/NOT
 - Tuition > 500 AND Sales < 200: Both conditions must be met
 - Tuition > 500 OR Sales < 200: Any one of the conditions is met
 - NOT 100 > 200

```
SELECT * FROM Registration
WHERE Actual_Tuition * 1.1 > 500
AND Student_Name = 'Luke Skywalker'
```



AND vs OR

- Common English vs Logical Operation

I want to find students in Data Science department and Computer Science department.

```
Student_Department = 'Data Science'
```

OR

```
Student_Department = 'Computer Science'
```

WHERE Clause: IN

- Syntax:

```
WHERE <column_computation> IN (<value_list>)
```

- Example:

```
WHERE Instructor_Name IN ('Yoda', 'Obi-wan Kenobi')
```

- Each string should be enclosed by single quote, not the whole list:

```
('Yoda, Obi-wan Kenobi') ❌
```

WHERE Clause: BETWEEN...AND

- BETWEEN ... AND

```
WHERE <column_computation>  
BETWEEN <lower_value> AND <upper_value>
```

- Inclusive of lower bound and upper bound
- Lower bound must be lower than upper bound

```
SELECT * FROM Registration  
WHERE Actual_Tuition BETWEEN 200 AND 400
```


WHERE Clause: String

String Comparison

- String comparison is character by character:
 - Compare first character first
 - Compare next character if previous one equals.

I	n	t	r	o	d	u	c	t	I	o	n
I	n	t	r	o	d	u	c	t	o	r	y

- Character comparison is based on ASCII

String Comparison

- Character comparison is based on ASCII

(“ASCII”, *Wikipedia*)

$0 < 1 < \dots < 9$

$<$

$A < B < \dots < Z$

$<$

$a < b < \dots < z$

USASCII code chart

b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				
b7 b6 b5					b4 b3 b2 b1 b0					b7 b6 b5 b4 b3 b2 b1 b0				

String Comparison

- Can use logical operator and BETWEEN...AND too

```
WHERE Student_Name BETWEEN 'A' AND 'F'
```

-- This will include:

```
'A', 'Bxxx', 'Ezzzzzzzzzzzz', 'F'
```

-- But not including 'F1'

String vs Number

- Some DBMS (such as PostgreSQL) can do implicit type conversion. It can interpret the following as a numeric comparison:

```
Actual_Tuition > '650'
```

- Don't trust it!
- DBMS can't properly interpret it if you can't properly write it.
 - Is '090' greater than '10'?

String Pattern Matching

- Looking for patterns:
 - Someone's name starting with Luke: 'Luke' + an unknown number of characters
 - Any SQL course: Starting with an unknown number of characters (can be 0), followed by the word 'SQL', followed by an unknown number of characters (can be 0).

WHERE Clause: LIKE

- LIKE: String pattern contains wildcard matching
 - % for multiple character wildcard
 - _ for single character wildcard

```
WHERE <column_or_computation>  
LIKE <String_Pattern>
```

WHERE Clause: LIKE with %

Starting with S	<code>'S%'</code>
Ending with t	<code>'%t'</code>
Starting with S and ending with t	<code>'S%t'</code>
Contains the pattern SQL	<code>'%SQL%'</code>
Contains the word SQL, regardless of case	<code>UPPER (Course_Name) LIKE '%SQL%'</code>

WHERE Clause: LIKE with _

Starting with one any character but continues with tatement	'_tatement'
Last char unknown but starting with SQ	'SQ_'
Last four chars unknown but starting with State	'State____'

WHERE Clause: LIKE

- Difference between `'_'` and `'%'`
 - `'_'` means there MUST be a character there. Can not be empty.
 - `'State____': 'State SQL', 'Statement', 'State XYZ'`
 - `'%'` means there may be nothing, or one, or many chars.
 - `'State%%%%': 'State', 'States', 'State ', 'Statement'`

PostgreSQL Behavior

- For PostgreSQL installed on Windows with default locale, the “>”, “>=”, “<”, “<=” and “BETWEEN...AND” comparisons for VARCHAR is case insensitive.
 - ‘L’ > ‘h’ will return TRUE
- All other comparisons are case sensitive
- All comparisons on PostgreSQL installed on Linux is case sensitive.

PostgreSQL Behavior

It is a PostgreSQL behavior, not SQL standard.

It can be changed, but usually is done by administrator, not data scientists.

Use C locale or C collation to enable standard SQL behavior.

WHERE Clause: Subquery

Subquery

- How to use the result of a query in another query's IN clause?
 - Run one query, write down the output, and type in another query
 - Put the first query in parenthesis and place it in the second query

```
WHERE Instructor_Name IN (  
    SELECT Instructor_Name FROM Instructor  
    WHERE Instructor_Affiliation = 'Jedi' )
```

Result of Subquery

- The result of Subqueries:
 - Vector (list), or
 - Single value using aggregation functions (SUM, AVG, e.g.) or
 - Single value using ANY/ALL prefix
- All three can be used in IN clause
- Single valued subquery can be used in normal WHERE clause, such as comparison

Using Aggregation in Subquery

- Single value using aggregation functions (SUM, AVG, e.g.)

```
WHERE Actual_Tuition > (  
    SELECT AVG(Actual_Tuition) FROM Registration  
    WHERE Course_Name = 'Introduction to the Force')
```

```
WHERE Actual_Tuition = (  
    SELECT MAX(Actual_Tuition) FROM Registration  
    WHERE Course_Name LIKE '%SQL%')
```


Using ALL/ANY Prefix in Subquery

- Single value using ALL/ANY prefix

```
WHERE Actual_Tuition > ALL (  
    SELECT Actual_Tuition FROM Registration  
    WHERE Course_Name = 'Introduction to the Force')
```

```
WHERE Actual_Tuition < ANY (  
    SELECT Actual_Tuition FROM Registration  
    WHERE Course_Name LIKE '%SQL%')
```

Aggregation vs ALL/ANY

- > MAX is equivalent to > ALL

```
WHERE Actual_Tuition > (  
    SELECT MAX(Actual_Tuition) FROM Registration  
    WHERE Course_Name = 'Introduction to the Force')
```

```
WHERE Actual_Tuition > ALL (  
    SELECT Actual_Tuition FROM Registration  
    WHERE Course_Name = 'Introduction to the Force')
```

Aggregation vs ALL/ANY

- Similarly:

> ALL is equivalent to > MAX ()

> ANY is equivalent to > MIN ()

< ALL is equivalent to < MIN ()

< ANY is equivalent to < MAX ()

SQL:

Intermediate vs Basic

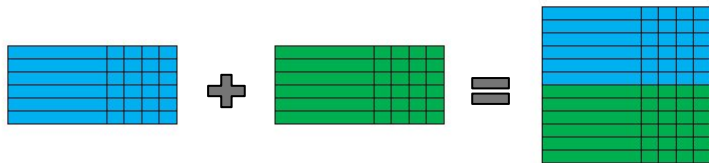
Basic vs Intermediate

- So far, we have learned the data access of relations -- Basic SQL.
- We will start learning the manipulation of dataset(s) -- Intermediate SQL
 - Join between relations
 - Aggregate relations
 - Filter based on relations
 - Treat everything as relation

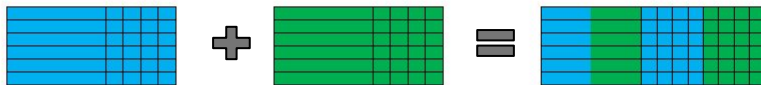
Dataset Operation

Data Blending

- Vertical Blending
 - Set operations



- Horizontal Blending
 - Join



UNION

- UNION: Link two SELECT statements with UNION key word
 - Data in either of the datasets, e.g., students registered in 2018 or students registered in 2019

```
SELECT <column_list> FROM <table_1>  
UNION  
SELECT <column_list> FROM <table_2>
```

- Column lists must be the same

UNION ALL

- UNION will only provide unique results.
- To show all occurrences, use UNION ALL, which will keep repeating rows

```
SELECT <column_list> FROM <table_1>  
UNION ALL  
SELECT <column_list> FROM <table_2>
```

- Column lists must match each other

INTERSECT

- INTERSECT
 - Data in both of the datasets, e.g., students registered in both 2018 and 2019

```
SELECT <column_list> FROM <table_1>  
INTERSECT  
SELECT <column_list> FROM <table_2>
```

- Column lists must be the same

EXCEPT

- EXCEPT
 - Data in one dataset but not the other, e.g., students registered in 2019 but not Student registered in 2018

```
SELECT <column_list> FROM <table_1>  
EXCEPT  
SELECT <column_list> FROM <table_2>
```

- Column lists must be the same
- EXCEPT is called MINUS in some other DBMS

Set Operations

- Putting two datasets with same structures together

