SQL Workshop (Dec):

Intermediate SQL – Day 3

UNDERSTANDING FUNCTIONS

- So, you have learned how to retrieve data from a single table, how to join 2 or more tables together to get a bigger source of information.

- Do you know of any other things you can do within SQL?

UNDERSTANDING FUNCTIONS

- Yes, within SQL, you are able to conduct a series of ***data wrangling*, **or data manipulation to get your data in a certain format for reporting purposes, or filtering purposes during data retrieval

- Today, we are going to go through namely a few kinds of functions: *String, Date, Numeric and Other Functions*.

STRING FUNCTIONS

- What are string functions used for?

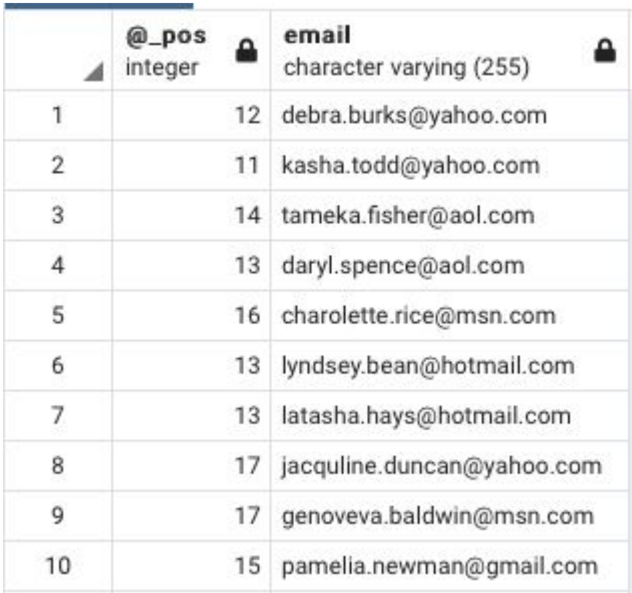
- In cases where you would like to reduce the amount of text (varchar, char) data you are working with be it for a reporting purpose or simply using the data retrieved for filtering/joining purposes.

STRPOS

- Returns starting index of specified *substring* within *string*, or zero if it's not present

- Use to find if there’s a specific word within the column

SELECT STRPOS(email, ‘@’) **AS** "@_pos", email
FROM sales.customers



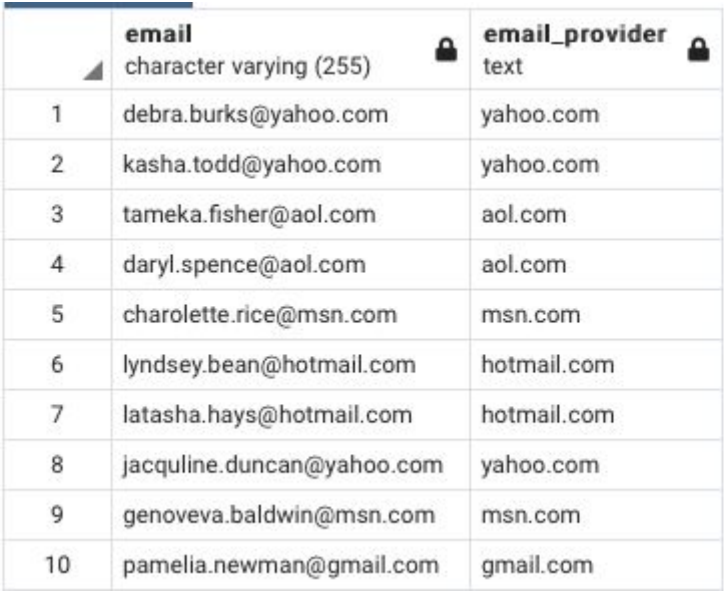
	@_pos integer	email character varying (255)
1	12	debra.burks@yahoo.com
2	11	kasha.todd@yahoo.com
3	14	tameka.fisher@aol.com
4	13	daryl.spence@aol.com
5	16	charolette.rice@msn.com
6	13	lyndsey.bean@hotmail.com
7	13	latasha.hays@hotmail.com
8	17	jacquline.duncan@yahoo.com
9	17	genoveva.baldwin@msn.com
10	15	pamelia.newman@gmail.com

SPLIT_PART (.split(‘.’))

- Splits *string* at occurrences of *delimiter* and returns the *n*'th field (counting from one)

- Breaking up column into smaller chunks of information

```
SELECT email, SPLIT_PART(email,
'@', 2) AS email_provider
FROM sales.customers
```



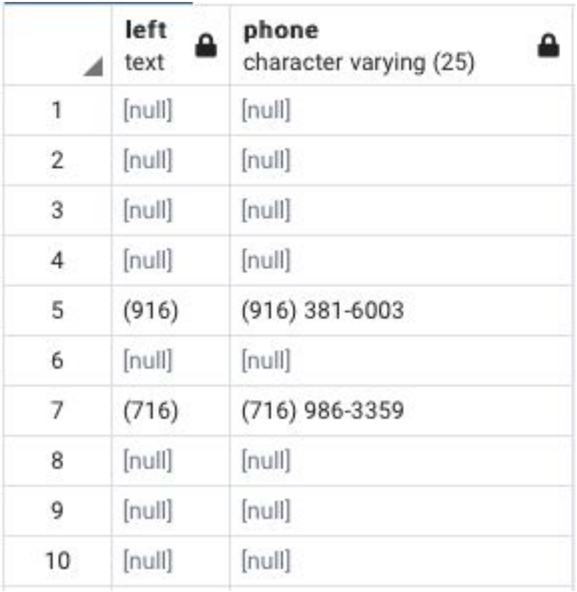
	email character varying (255)	email_provider text
1	debra.burks@yahoo.com	yahoo.com
2	kasha.todd@yahoo.com	yahoo.com
3	tameka.fisher@aol.com	aol.com
4	daryl.spence@aol.com	aol.com
5	charolette.rice@msn.com	msn.com
6	lyndsey.bean@hotmail.com	hotmail.com
7	latasha.hays@hotmail.com	hotmail.com
8	jacquline.duncan@yahoo.com	yahoo.com
9	genoveva.baldwin@msn.com	msn.com
10	pamelia.newman@gmail.com	gmail.com

LEFT

- Returns first *n* characters in the string, or when *n* is negative, returns all but last |*n*| characters.
- Subsetting columns for more detailed information (keep left first)

SELECT LEFT(phone, 5), phone

FROM sales.customers



	left text	phone character varying (25)
1	[null]	[null]
2	[null]	[null]
3	[null]	[null]
4	[null]	[null]
5	(916)	(916) 381-6003
6	[null]	[null]
7	(716)	(716) 986-3359
8	[null]	[null]
9	[null]	[null]
10	[null]	[null]

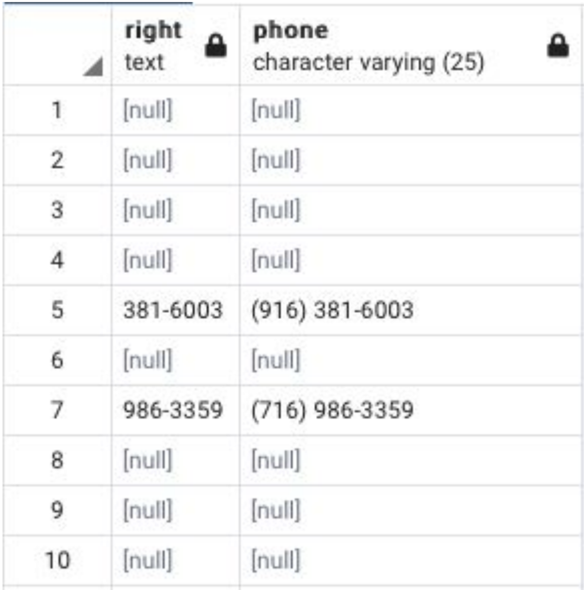
RIGHT

- Returns first *n* characters in the string, or when *n* is negative, returns all but first |*n*| characters.

- Subsetting columns for more detailed information (keep right first)

SELECT RIGHT(phone, 8), phone

FROM sales.customers



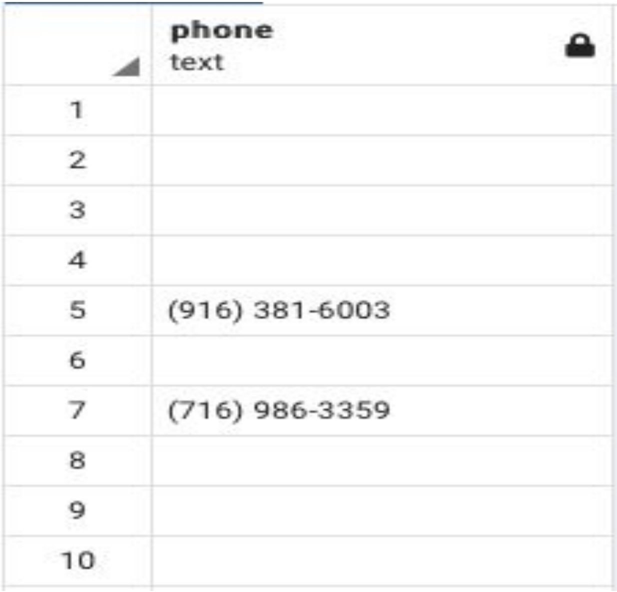
	right text	phone character varying (25)
1	[null]	[null]
2	[null]	[null]
3	[null]	[null]
4	[null]	[null]
5	381-6003	(916) 381-6003
6	[null]	[null]
7	986-3359	(716) 986-3359
8	[null]	[null]
9	[null]	[null]
10	[null]	[null]

CONCAT or w/ +

- Concatenates the text representations of all the arguments. NULL arguments are ignored.

- Join varchar/char columns together to condense information

```
SELECT CONCAT(LEFT(phone, 5), ' ',
RIGHT(phone, 8)) AS phone
FROM sales.customers
```



	phone text
1	
2	
3	
4	
5	(916) 381-6003
6	
7	(716) 986-3359
8	
9	
10	

HANDS-ON

1. Is it true that all emails consists of the last name of the customers? Write a query to check.

2. How to retrieve ‘St. ‘ and ‘Dr. ‘ from the “street” col in CUSTOMERS using RIGHT?

3. If you do not have the “first_name” col, how would retrieve the first_name information using email?

OTHER FUNCTIONS

- Used to deal with numerical data and transform numeric data type into certain format for reporting purposes, or for further data manipulation.

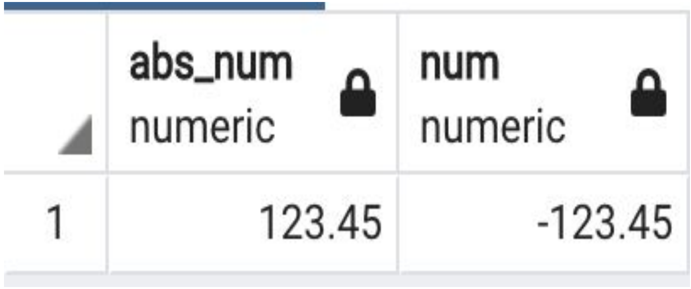
- To manipulate datetime data type to either get back day, month or year to enable reporting or further data manipulation

- Create certain conditions that enables us to transform data based on our needs.

ABS

- Returns the absolute value of a number
  - E.g. |x| = x if x ≥ 0, -x if x < 0

- The returned value will always be positive



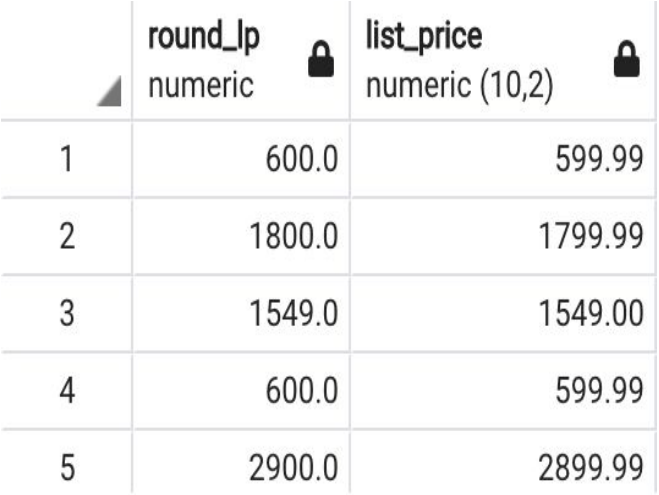
	abs_num numeric 🔒	num numeric 🔒
1	123.45	-123.45

```
SELECT ABS(-123.45) AS abs_num,
-123.45 AS num
```

ROUND

- Returns the value rounded off to the *n*th decimal place
  - ROUND(123.456, 2) -> 123.46

```
SELECT ROUND(list_price, 1) as
round_lp, list_price
FROM sales.order_items
```

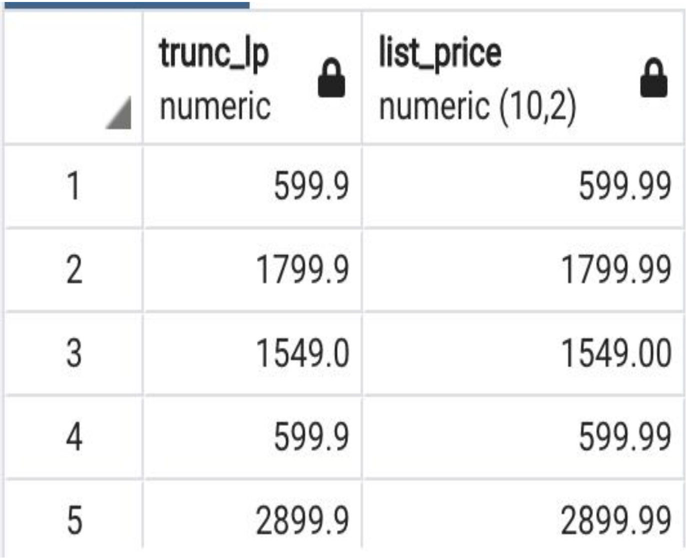


	round_lp numeric	list_price numeric (10,2)
1	600.0	599.99
2	1800.0	1799.99
3	1549.0	1549.00
4	600.0	599.99
5	2900.0	2899.99

TRUNC

- Returns the value truncated to the *n*th decimal place
  - TRUNC(123.456, 2) -> 123.45

```
SELECT TRUNC(list_price) AS
trunc_lp, list_price
FROM sales.order_items
```



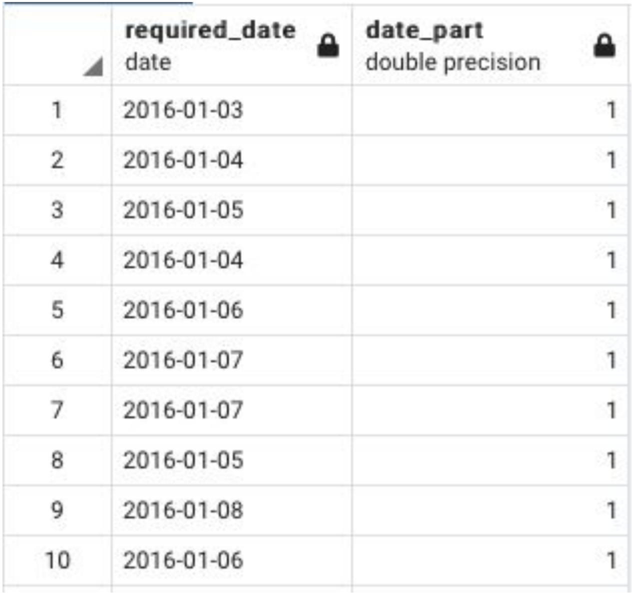
	trunc_lp numeric	list_price numeric (10,2)
1	599.9	599.99
2	1799.9	1799.99
3	1549.0	1549.00
4	599.9	599.99
5	2899.9	2899.99

DATE_PART

- Retrieve specific part of a datetime data type; eg. day, month, year

- date_part(text, interval)

```
SELECT required_date,
DATE_PART('month', required_date)
FROM sales.orders
```



	required_date date	date_part double precision
1	2016-01-03	1
2	2016-01-04	1
3	2016-01-05	1
4	2016-01-04	1
5	2016-01-06	1
6	2016-01-07	1
7	2016-01-07	1
8	2016-01-05	1
9	2016-01-08	1
10	2016-01-06	1

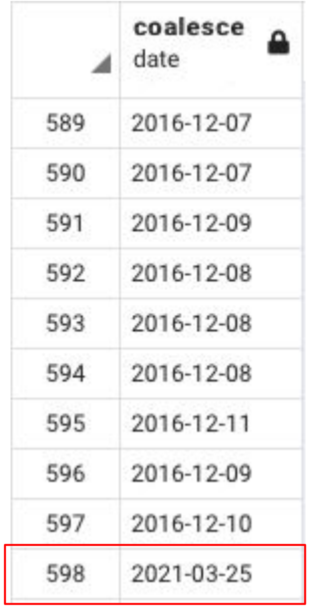
COALESCE

- Replace null value if another value, if null exists. *Must be of same data type

- COALESCE(***value*** [, ...])

SELECT
COALESCE(shipped_date,
current_date)

FROM sales.orders



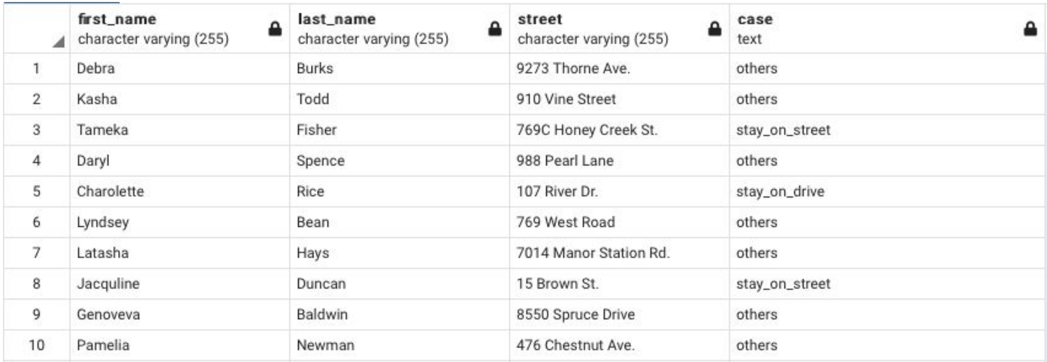
	coalesce date
589	2016-12-07
590	2016-12-07
591	2016-12-09
592	2016-12-08
593	2016-12-08
594	2016-12-08
595	2016-12-11
596	2016-12-09
597	2016-12-10
598	2021-03-25

CASE

- If/Else statement for PSQL.

```
CASE WHEN condition THEN result
    [WHEN ...]
    [ELSE result]
END
```

```
SELECT first_name,
        last_name,
        street,
        CASE
                WHEN RIGHT(street, 4)='St. ' THEN 'stay_on_street'
                WHEN RIGHT(street, 4)='Dr. ' THEN 'stay_on_drive'
                ELSE 'others'
        END
FROM sales.customers
```



	first_name character varying (255)	last_name character varying (255)	street character varying (255)	case text
1	Debra	Burks	9273 Thorne Ave.	others
2	Kasha	Todd	910 Vine Street	others
3	Tameka	Fisher	769C Honey Creek St.	stay_on_street
4	Daryl	Spence	988 Pearl Lane	others
5	Charolette	Rice	107 River Dr.	stay_on_drive
6	Lyndsey	Bean	769 West Road	others
7	Latasha	Hays	7014 Manor Station Rd.	others
8	Jacquline	Duncan	15 Brown St.	stay_on_street
9	Genoveva	Baldwin	8550 Spruce Drive	others
10	Pamelia	Newman	476 Chestnut Ave.	others

HANDS-ON

- Indicate(1/0) the customer with J as the starting letter for the last name using CASE

- Count the number of orders by the month of the required shipping date

FURTHER LEARNING

To further your learning in SQL, I highly recommend the following course on Udemy:

Jose Portilla’s – The Complete SQL Bootcamp 2020: From Zero to Hero:

https://tinyurl.com/yy8bf9ft

END OF DAY 3!

Any questions? Feel free to clarify now.

Or you can reach us at:

Wei Teck (Jensen): wtlow003@suss.edu.sg / https://www.linkedin.com/in/weitecklow/

Jeanette: jeanettekoh001@suss.edu.sg / linkedin.com/in/jeanette-koh-872b64192