Combining columns

CLEANING DATA IN POSTGRESQL DATABASES

SQL

Darryl Reeves, Ph.D.
Industry Assistant Professor, New York University

datacamp

Combining columns (an example)

Concatenation

```
name                        |  boro  | building |      street      | zip_code | ...
----------------------------+--------+----------+------------------+----------+-----
...                         | ...    | ...      | ...              | ...      | ...
DARBAR'S CHICKEN & RIBS     | Queens | 12609    | LIBERTY AVE      | 11419    | ...
F & J PINE RESTAURANT       | Bronx  | 1913     | BRONXDALE AVENUE | 10462    | ...
EL RINCONCITO DE LOS SABORES | Queens | 13933    | 89TH AVE         | 11435    | ...
DON NICO'S                  | Queens | 9014     | 161ST ST         | 11432    | ...
ASTORIA PIZZA               | Queens | 3204B    | 30TH AVE         | 11102    | ...
...                         | ...    | ...      | ...              | ...      | ...
```

```
Restaurant Name
Street Address
Boro, NY Zipcode
```

Joining values with CONCAT()

- `CONCAT(string1 [, string2, string3, ...])`
- `CONCAT('data', 'cleaning', 'is', 'fun')` → `datacleaningisfun`
- `CONCAT('data', ' ', 'cleaning', ' ', 'is', ' ', 'fun')` → `data cleaning is fun`

Joining values with CONCAT()

```sql
SELECT
  CONCAT(
      name, E'\n',
      building, ' ', street, E'\n',
      boro, ', NY ', zip_code
  ) AS mailing_address
FROM
  restaurant_inspection;
```

```
mailing_address
---------------------------
DARBAR'S CHICKEN & RIBS     +
12609 LIBERTY AVE           +
Queens, NY 11419
F & J PINE RESTAURANT       +
1913 BRONXDALE AVENUE       +
Bronx, NY 10462
EL RINCONCITO DE LOS SABORES+
13933 89TH AVE              +
Queens, NY 11435
DON NICO'S                  +
9014 161ST ST               +
Queens, NY 11432
ASTORIA PIZZA               +
3204B 30TH AVE              +
Queens, NY 11102
```

Joining values with CONCAT()

```
name                    |  boro     | building |      street                | zip_code | ...
------------------------+-----------+----------+----------------------------+----------+-----
...                     | ...       | ...      | ...                        | ...      | ...
IRVING FARMS            | Queens    |          | CENTRAL TERMINAL BUILDING  | 11371    | ...
DON PEPIS DELICATESSEN  | Manhattan |          | AMTRAK LEVEL               | 10001    | ...
DUNKIN'                 | Queens    |          | CENTRAL TERMINAL BLDG      | 11371    | ...
                        | Queens    | 17111    | JAMAICA AVE                | 11432    | ...
                        | Brooklyn  | 1489     | FULTON STREET              | 11216    | ...
...                     | ...       | ...      | ...                        | ...      | ...
```

Joining values with CONCAT()

```sql
SELECT
  CONCAT(
      name, E'\n',
      building, ' ', street, E'\n',
      boro, ', NY ', zip_code
  ) AS mailing_address
FROM
  restaurant_inspection;
```

```
     mailing_address
---------------------------
IRVING FARMS              +
 CENTRAL TERMINAL BUILDING+
Queens, NY 11371
DON PEPIS DELICATESSEN    +
 AMTRAK LEVEL             +
Manhattan, NY 10001
DUNKIN'                   +
 CENTRAL TERMINAL BLDG    +
Queens, NY 11371
                          +
17111 JAMAICA AVE         +
Queens, NY 11432
                          +
1489 FULTON STREET        +
Brooklyn, NY 11216
```

Joining values with ||

- `string1 || string2 [ || string3 || ...]`

```
SELECT 'data' || ' ' || 'cleaning' || ' ' || 'is' || ' ' || 'fun';
```

```
data cleaning is fun
```

`NULL` valued arguments → `NULL` value

```
SELECT
  name || E'\n' ||
  building || ' ' || street || E'\n'
  || boro || ', NY ' || zip_code AS mailing_address
FROM
  restaurant_inspection
```

Joining values with ||

```
name              |    mailing_address
------------------+-----------------------
SCHNIPPERS        | SCHNIPPERS           +
                  | 570 LEXINGTON AVENUE +
                  | Manhattan, NY 10022
ATOMIC WINGS      |
WING LING         | WING LING            +
                  | 159B EAST  170 STREET+
                  | Bronx, NY 10452
JUAN VALDEZ CAFE | JUAN VALDEZ CAFE     +
                  | 140 EAST   57 STREET +
                  | Manhattan, NY 10022
FULTON GRAND      | FULTON GRAND         +
                  | 1011 FULTON STREET   +
                  | Brooklyn, NY 11238
```

Let's practice!

CLEANING DATA IN POSTGRESQL DATABASES

Splitting column data

CLEANING DATA IN POSTGRESQL DATABASES

SQL

Darryl Reeves, Ph.D.
Industry Assistant Professor, New York University

datacamp

Splitting columns

```
camis    | inspection_date |                    violation                    | ...
---------+-----------------+-------------------------------------------------+-----
...      | ...             | ...                                             | ...
50038736 | 03/29/2018      | 09B Thawing procedures                          | ...
50033304 | 12/18/2019      | 02B Hot food item not held at or above 140° ... | ...
50081658 | 12/13/2018      | 06F Wiping cloths soiled or not stored in sa... | ...
50033733 | 02/12/2019      | 10B Plumbing not properly installed or maint... | ...
40559634 | 08/22/2017      | 04N Filth flies or food/refuse/sewage-associ... | ...
...      | ...             | ...                                             | ...
```

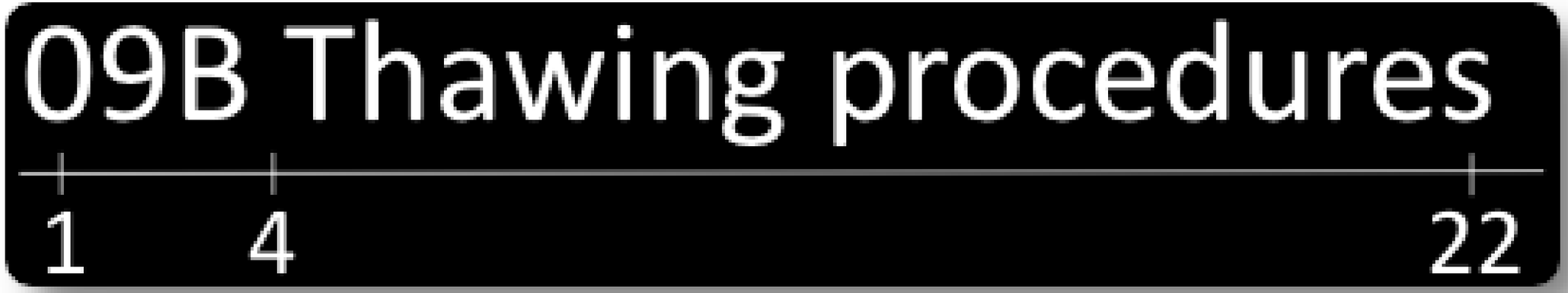
Finding substring starting position with STRPOS()

```
STRPOS(source_string, search_string)
```


```
SELECT
  STRPOS('09B Thawing procedures', ' ');
```

```
4
```

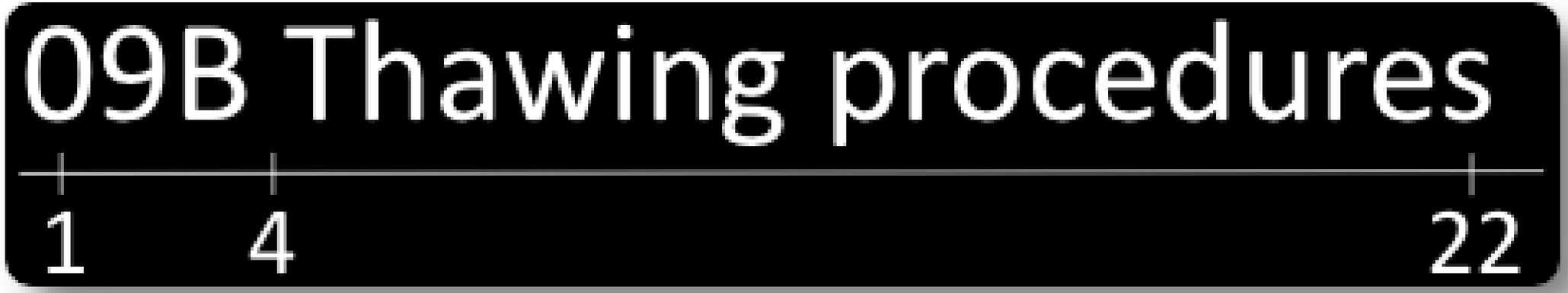
Finding substring starting position with STRPOS()




```
SELECT
  STRPOS('09B Thawing procedures', '?');
```

```
0
```

Finding substring starting position with STRPOS()




```
SELECT
  STRPOS('09B Thawing procedures', ' ');
```

```
4
```

Extracting a substring using SUBSTRING()

```
SUBSTRING(source_string FROM start_pos FOR num_chars)
```

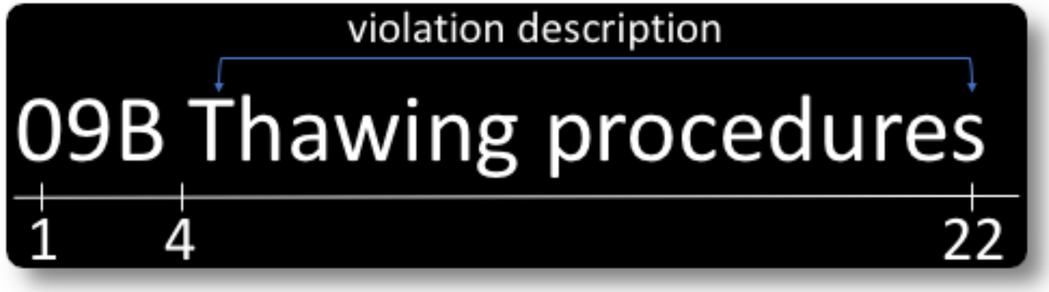
Extracting a substring using SUBSTRING()

`SUBSTRING('Homerun' FROM 1 FOR 4)` → `Home`


```
SELECT
  SUBSTRING(
      '09B Thawing procedures'
      FROM 1
      FOR STRPOS('09B Thawing procedures', ' ') - 1
  );
```

```
09B
```

Extracting a substring using SUBSTRING()



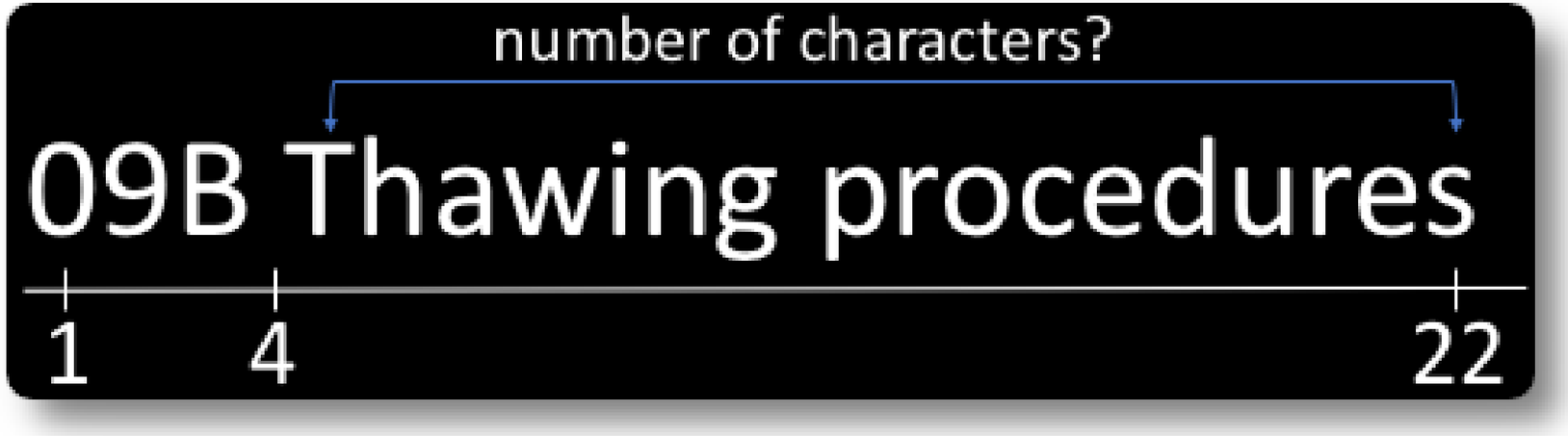
Requirements:

- Violation description start position
- Number of characters in the description

```
SELECT
  STRPOS('09B Thawing procedures', ' ') + 1;
```

```
5
```

Calculating the length of a string with LENGTH()



`LENGTH(string)` → `INTEGER`

```
SELECT LENGTH('hello')
```

```
5
```

Calculating the length of a string with LENGTH()

`LENGTH('09B Thawing procedures')` → `22`

`STRPOS('09B Thawing procedures', ' ')` → `4`

`LENGTH('09B Thawing procedures') - STRPOS('09B Thawing procedures', ' ')` → `18`

`LENGTH('Thawing procedures')` → `18`

Calculating the length of a string with LENGTH()

```
SELECT
    LENGTH('09B Thawing procedures') -
    STRPOS('09B Thawing procedures', ' ');
```

```
18
```

Putting the pieces together

```
SELECT
    SUBSTRING(
      '09B Thawing procedures'
      FROM
        STRPOS('09B Thawing procedures', ' ')
        + 1
      FOR
        LENGTH('09B Thawing procedures')
        - STRPOS('09B Thawing procedures', ' ')
    );
```

```
Thawing procedures
```

Splitting the violation column

```sql
SELECT
    camis,
    inspection_date,

    SUBSTRING(
      violation
      FROM 1
      FOR STRPOS(violation, ' ') - 1
    ) AS violation_code,

    SUBSTRING(
      violation
      FROM STRPOS(violation, ' ') + 1
      FOR LENGTH(violation) - STRPOS(violation, ' ')
    ) AS violation_description
FROM
    restaurant_inspection;
```

Splitting the violation column

```
camis    | inspection_date |                    violation                    | ...
---------+-----------------+-------------------------------------------------+-----
...      | ...             | ...                                             | ...
50038736 | 03/29/2018      | 09B Thawing procedures                          | ...
50033304 | 12/18/2019      | 02B Hot food item not held at or above 140° ... | ...
50081658 | 12/13/2018      | 06F Wiping cloths soiled or not stored in sa... | ...
50033733 | 02/12/2019      | 10B Plumbing not properly installed or maint... | ...
40559634 | 08/22/2017      | 04N Filth flies or food/refuse/sewage-associ... | ...
...      | ...             | ...                                             | ...
```

Splitting the violation column

```
camis    | inspection_date | violation_code |            violation_description            | ...
---------+-----------------+----------------+--------------------------------------------+-----
...      | ...             | ...            | ...                                        | ...
50038736 | 03/29/2018      | 09B            | Thawing procedures                         | ...
50033304 | 12/18/2019      | 02B            | Hot food item not held at or above 140° ... | ...
50081658 | 12/13/2018      | 06F            | Wiping cloths soiled or not stored in sa... | ...
50033733 | 02/12/2019      | 10B            | Plumbing not properly installed or maint... | ...
40559634 | 08/22/2017      | 04N            | Filth flies or food/refuse/sewage-associ... | ...
...      | ...             | ...            | ...                                        | ...
```

Let's practice!

CLEANING DATA IN POSTGRESQL DATABASES

Splitting data with delimiters

CLEANING DATA IN POSTGRESQL DATABASES

SQL

Darryl Reeves, Ph.D.
Industry Assistant Professor, New York University

datacamp

Splitting data into columns

```
camis    |         name          |               inspection_type                | ...
---------+-----------------------+----------------------------------------------+-----
...      | ...                   | ...                                          | ...
50084922 | JUICE POINT           | Cycle Inspection / Re-inspection             | ...
50075375 | ATOMIC WINGS          | Administrative Miscellaneous / Re-inspection | ...
50048685 | KENNEDY FRIED CHICKEN | Cycle Inspection / Re-inspection             | ...
50058910 | HUNGER PANG           | Pre-permit (Operational) / Re-inspection     | ...
50047834 | SUBWAY                | Smoke-Free Air Act / Re-inspection           | ...
...      | ...                   | ...                                          | ...
```

Value delimiter: `' / '`

```
sub_inspection_type          | count
-----------------------------+-------
Reopening Inspection         |    56
Re-inspection                |  1333
Initial Inspection           |  3488
Second Compliance Inspection |     2
Compliance Inspection        |    27
```

Splitting data into columns

```
camis    |         name          |               inspection_type                | ...
---------+-----------------------+----------------------------------------------+-----
...      | ...                   | ...                                          | ...
50084922 | JUICE POINT           | Cycle Inspection / Re-inspection             | ...
50075375 | ATOMIC WINGS          | Administrative Miscellaneous / Re-inspection | ...
50048685 | KENNEDY FRIED CHICKEN | Cycle Inspection / Re-inspection             | ...
50058910 | HUNGER PANG           | Pre-permit (Operational) / Re-inspection     | ...
50047834 | SUBWAY                | Smoke-Free Air Act / Re-inspection           | ...
...      | ...                   | ...                                          | ...
```

```
 camis   |         name          |     main_inspection_type     | sub_inspection_type | ...
---------+-----------------------+------------------------------+---------------------+-----
...      | ...                   | ...                          | ...                 | ...
50084922 | JUICE POINT           | Cycle Inspection             | Re-inspection       | ...
50075375 | ATOMIC WINGS          | Administrative Miscellaneous | Re-inspection       | ...
50048685 | KENNEDY FRIED CHICKEN | Cycle Inspection             | Re-inspection       | ...
50058910 | HUNGER PANG           | Pre-permit (Operational)     | Re-inspection       | ...
50047834 | SUBWAY                | Smoke-Free Air Act           | Re-inspection       | ...
...      | ...                   | ...                          | ...                 | ...
```

Splitting strings using SPLIT_PART()

- `SPLIT_PART(source_string, delimiter_string, part_number)`

```
SELECT
  SPLIT_PART('Cycle Inspection / Re-inspection', ' / ', 1);
```

```
Cycle Inspection
```

```
SELECT
  SPLIT_PART('Cycle Inspection / Re-inspection', ' / ', 2);
```

```
Re-inspection
```

Splitting strings using SPLIT_PART()

```sql
SELECT
  camis,
  name,
  SPLIT_PART(inspection_type, ' / ', 1) AS main_inspection_type,
  SPLIT_PART(inspection_type, ' / ', 2) AS sub_inspection_type
FROM
  restaurant_inspection;
```

```
camis    |         name          |       main_inspection_type    | sub_inspection_type | ...
---------+-----------------------+-------------------------------+---------------------+-----
...      | ...                   | ...                           | ...                 | ...
50084922 | JUICE POINT           | Cycle Inspection              | Re-inspection       | ...
50075375 | ATOMIC WINGS          | Administrative Miscellaneous  | Re-inspection       | ...
50048685 | KENNEDY FRIED CHICKEN | Cycle Inspection              | Re-inspection       | ...
50058910 | HUNGER PANG           | Pre-permit (Operational)      | Re-inspection       | ...
50047834 | SUBWAY                | Smoke-Free Air Act            | Re-inspection       | ...
...      | ...                   | ...                           | ...                 | ...
```

Splitting data into rows

```
 camis   |         name          | cuisine_description | ...
---------+-----------------------+---------------------+-----
...      | ...                   | ...                 | ...
50066768 | FIRST LAMB SHABU      | Chinese             | ...
41450971 | GIOVANNI'S RESTAURANT | Pizza/Italian       | ...
41628459 | KFC                   | Chicken             | ...
50043003 | BANGIA                | Korean              | ...
41418978 | BAGEL EXPRESS III     | Bagels/Pretzels     | ...
...      | ...                   | ...                 | ...
```

```
camis   |         name          | cuisine_description | ...
-------+-----------------------+---------------------+-----
...      | ...                   | ...                 | ...
50066768 | FIRST LAMB SHABU      | Chinese             | ...
41450971 | GIOVANNI'S RESTAURANT | Pizza               | ...
41450971 | GIOVANNI'S RESTAURANT | Italian             | ...
41628459 | KFC                   | Chicken             | ...
50043003 | BANGIA                | Korean              | ...
41418978 | BAGEL EXPRESS III     | Bagels              | ...
41418978 | BAGEL EXPRESS III     | Pretzels            | ...
...      | ...                   | ...                 | ...
```

Splitting data with REGEXP_SPLIT_TO_TABLE()

`REGEXP_SPLIT_TO_TABLE(source, pattern)`

```
SELECT REGEXP_SPLIT_TO_TABLE('Pizza/Italian', '/');
```

```
Pizza
Italian
```

Splitting data with REGEXP_SPLIT_TO_TABLE()

```sql
SELECT
  camis,
  name,
  REGEXP_SPLIT_TO_TABLE(cuisine_description, '/') AS cuisine_description,
  ...
FROM
  restaurant_inspection;
```

```
camis    |         name          | cuisine_description | ...
---------+-----------------------+---------------------+-----
...      | ...                   | ...                 | ...
50066768 | FIRST LAMB SHABU      | Chinese             | ...
41450971 | GIOVANNI'S RESTAURANT | Pizza               | ...
41450971 | GIOVANNI'S RESTAURANT | Italian             | ...
41628459 | KFC                   | Chicken             | ...
50043003 | BANGIA                | Korean              | ...
41418978 | BAGEL EXPRESS III     | Bagels              | ...
41418978 | BAGEL EXPRESS III     | Pretzels            | ...
...      | ...                   | ...                 | ...
```

Enumerating the resulting rows

```
cuisine_num |  camis   |         name          | cuisine_description | ...
------------+----------+-----------------------+---------------------+-----
...         | ...      | ...                   | ...                 | ...
1           | 41418978 | BAGEL EXPRESS III     | Bagels              | ...
2           | 41418978 | BAGEL EXPRESS III     | Pretzels            | ...
1           | 41450971 | GIOVANNI'S RESTAURANT | Pizza               | ...
2           | 41450971 | GIOVANNI'S RESTAURANT | Italian             | ...
1           | 41628459 | KFC                   | Chicken             | ...
1           | 50043003 | BANGIA                | Korean              | ...
1           | 50066768 | FIRST LAMB SHABU      | Chinese             | ...
...         | ...      | ...                   | ...                 | ...
```

`ROW_NUMBER() OVER()`

`PARTITION BY col1, col2, ...`

`ORDER BY colA, colB, ...`

Enumerating the resulting rows

```sql
SELECT
  ROW_NUMBER() OVER (
    PARTITION BY
      -- group columns for numbering
      camis,
      name
    ORDER BY
      -- set ordering of results
      camis,
      name
  ) AS cuisine_num,
  *
FROM (
  SELECT
    camis,
    name,
    REGEXP_SPLIT_TO_TABLE(cuisine_description, '/')
      AS cuisine_description
FROM
  restaurant_inspection;
```

```
cuisine_num |  camis   |         name          | cuisine_description | ...
------------+----------+-----------------------+---------------------+-----
...         | ...      | ...                   | ...                 | ...
1           | 41418978 | BAGEL EXPRESS III     | Bagels              | ...
2           | 41418978 | BAGEL EXPRESS III     | Pretzels            | ...
1           | 41450971 | GIOVANNI'S RESTAURANT | Pizza               | ...
2           | 41450971 | GIOVANNI'S RESTAURANT | Italian             | ...
1           | 41628459 | KFC                   | Chicken             | ...
1           | 50043003 | BANGIA                | Korean              | ...
1           | 50066768 | FIRST LAMB SHABU      | Chinese             | ...
...         | ...      | ...                   | ...                 | ...
```

Let's practice!

CLEANING DATA IN POSTGRESQL DATABASES

Creating pivot tables

CLEANING DATA IN POSTGRESQL DATABASES

SQL

Darryl Reeves, Ph.D.
Assistant Professor, Long Island University - Brooklyn

datacamp

Multiple category records

```
              name                 |             inspection_type             | grade | ...
-----------------------------------+-----------------------------------------+-------+-----
...                                | ...                                     | ...   | ...
EMPANADAS MONUMENTAL               | Cycle Inspection / Re-inspection        | B     | ...
ALPHONSO'S PIZZERIA & TRATTORIA    | Cycle Inspection / Initial Inspection   | A     | ...
THE SPARROW TAVERN                 | Cycle Inspection / Initial Inspection   | A     | ...
BURGER KING                        | Cycle Inspection / Re-inspection        | A     | ...
ASTORIA PIZZA                      | Cycle Inspection / Re-inspection        | B     | ...
...                                | ...                                     | ...   | ...
```

Accessing inspection grades by type

```sql
SELECT
    inspection_type,
    grade,
    COUNT(*)
FROM
    restaurant_inspection
WHERE
    grade IS NOT NULL
GROUP BY
    inspection_type,
    grade
ORDER BY
    inspection_type,
    grade;
```

Aggregated inspection results by type

```
                  inspection_type                  | grade | count
-----------------------------------------------------+-------+-------
Cycle Inspection / Initial Inspection              | A     |  1063
Cycle Inspection / Re-inspection                   | A     |   723
Cycle Inspection / Re-inspection                   | B     |   270
Cycle Inspection / Re-inspection                   | C     |    93
Cycle Inspection / Re-inspection                   | Z     |    29
Cycle Inspection / Reopening Inspection            | C     |     8
Cycle Inspection / Reopening Inspection            | P     |    26
Cycle Inspection / Reopening Inspection            | Z     |     3
Pre-permit (Non-operational) / Initial Inspection | N     |     4
Pre-permit (Operational) / Initial Inspection      | A     |   119
Pre-permit (Operational) / Initial Inspection      | N     |    17
Pre-permit (Operational) / Re-inspection           | A     |    79
Pre-permit (Operational) / Re-inspection           | B     |    49
Pre-permit (Operational) / Re-inspection           | C     |    13
Pre-permit (Operational) / Re-inspection           | Z     |     9
Pre-permit (Operational) / Reopening Inspection    | C     |     3
Pre-permit (Operational) / Reopening Inspection    | P     |     3
Pre-permit (Operational) / Reopening Inspection    | Z     |     1
```

Changing (pivoting) data orientation

inspection_type	A	B	C	N	P	Z
Cycle Inspection / Re-inspection	723	270	93	0	0	29
Cycle Inspection / Initial Inspection	1063	0	0	0	0	0
Pre-permit (Operational) / Reopening Inspection	0	0	3	0	3	1
Cycle Inspection / Reopening Inspection	0	0	8	0	26	3
Pre-permit (Non-operational) / Initial Inspection	0	0	0	4	0	0
Pre-permit (Operational) / Initial Inspection	119	0	0	17	0	0
Pre-permit (Operational) / Re-inspection	79	49	13	0	0	9

The FILTER clause

- Applies an aggregation over a subset of records
- Subset of records determined by accompanying `WHERE` clause
- Used in the `SELECT` list of a query

The FILTER clause

- Example: `AVG(qty_sold) FILTER (WHERE qty_sold > 1)`
- Format: `AGG_FUNC(expression) FILTER (WHERE condition)`
  - `AGG_FUNC()` - aggregate function

The pivot table query

```
SELECT
    summary_column,
    AGG(agg_column) FILTER (WHERE agg_column = PIVOT_VALUE_1) AS "pivot_column_1",
    AGG(agg_column) FILTER (WHERE agg_column = PIVOT_VALUE_2) AS "pivot_column_2",
    ...
    AGG(agg_column) FILTER (WHERE agg_column = PIVOT_VALUE_N) AS "pivot_column_N"
FROM
    source_table
GROUP BY
    summary_column;
```

The pivot table output

summary_column	pivot_column_1	pivot_column_2	...	pivot_column_N
summary_val_1	agg result for PV1	agg result for PV2		agg value for PVN
summary_val_2	agg result for PV1	agg result for PV2		agg value for PVN
...	...			...
summary_val_M	agg result for PV1	agg result for PV2	...	agg value for PVN

Pivoting restaurant inspection data

```sql
SELECT
    inspection_type,
    COUNT(grade) FILTER (WHERE grade = 'A') AS "A",
    COUNT(grade) FILTER (WHERE grade = 'B') AS "B",
    COUNT(grade) FILTER (WHERE grade = 'C') AS "C",
    COUNT(grade) FILTER (WHERE grade = 'N') AS "N",
    COUNT(grade) FILTER (WHERE grade = 'P') AS "P",
    COUNT(grade) FILTER (WHERE grade = 'Z') AS "Z"
FROM
    restaurant_inspections
WHERE
    grade IS NOT NULL
GROUP BY
    inspection_type;
```

Pivot table output for inspection data

inspection_type	A	B	C	N	P	Z
Cycle Inspection / Re-inspection	723	270	93	0	0	29
Cycle Inspection / Initial Inspection	1063	0	0	0	0	0
Pre-permit (Operational) / Reopening Inspection	0	0	3	0	3	1
Cycle Inspection / Reopening Inspection	0	0	8	0	26	3
Pre-permit (Non-operational) / Initial Inspection	0	0	0	4	0	0
Pre-permit (Operational) / Initial Inspection	119	0	0	17	0	0
Pre-permit (Operational) / Re-inspection	79	49	13	0	0	9

Let's practice!

CLEANING DATA IN POSTGRESQL DATABASES

Course wrap-up

CLEANING DATA IN POSTGRESQL DATABASES

SQL

Darryl Reeves, Ph.D.
Industry Assistant Professor, New York University

datacamp

Course content

Chapter 1: Data cleaning basics

Chapter 2: Missing, duplicate, and invalid data

Chapter 3: Converting data

Chapter 4: Transforming data

Onward!

- Functions for Manipulating Data in PostgreSQL
- Reporting in SQL
- PostgreSQL Summary Stats and Window Functions
- Exploratory Data Analysis in SQL

Congratulations!

CLEANING DATA IN POSTGRESQL DATABASES