String Functions and Pivoting Data

Relevel
by Unacademy

Assignment Discussion- Previous Class

Assignment Question

Use mode.com:
Tables:
tutorial.crunchbase_companies_clean_date
tutorial.crunchbase_acquisitions_clean_date

Write a query that counts the number of companies acquired within 1st year, 2nd year, and 3rd year of being founded (in 3 separate columns). Include a column for total companies acquired as well. Group by category and limit to only rows with a founding date.

Assignment Solution Link

Relevel
by Unacademy

What are String Functions?

These are the functions that are primarily utilized for string manipulation. Multiple built-in SQL string functions make it easier for us to find and alter string values.

A few examples of string manipulation are:

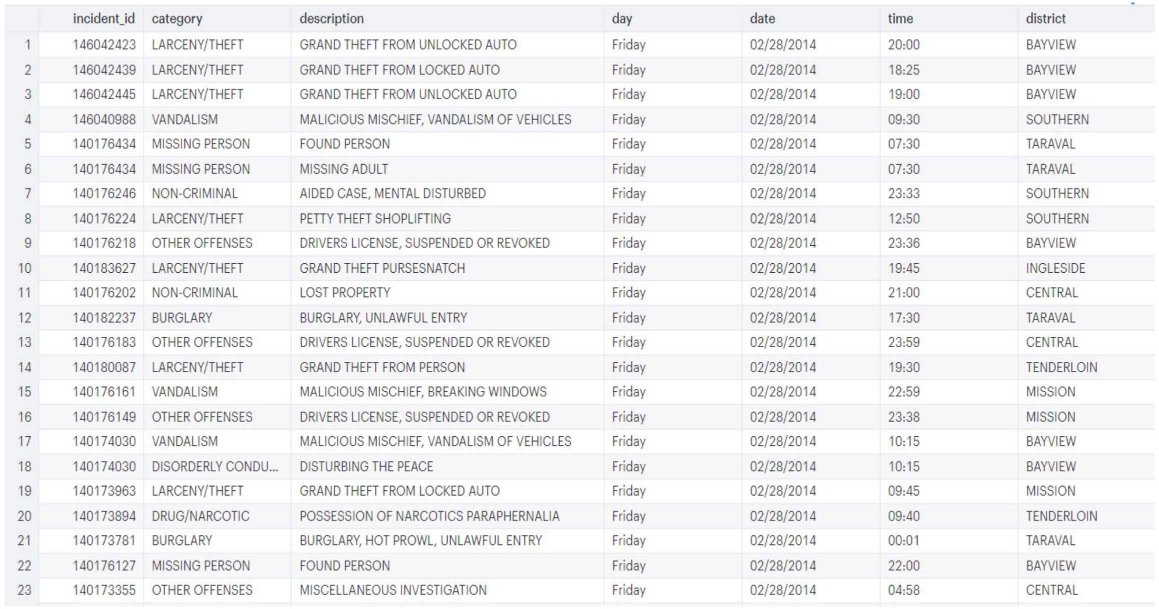
- Removing blanks from a string
- Find the position of a character/word in a string
- Finding the length of a string

Learning through an example

Relevel by Unacademy

We will use the below-mentioned data (incidents) to understand the string functions:

heather.sfpd_incidents



	incident_id	category	description	day	date	time	district
1	146042423	LARCENY/THEFT	GRAND THEFT FROM UNLOCKED AUTO	Friday	02/28/2014	20:00	BAYVIEW
2	146042439	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Friday	02/28/2014	18:25	BAYVIEW
3	146042445	LARCENY/THEFT	GRAND THEFT FROM UNLOCKED AUTO	Friday	02/28/2014	19:00	BAYVIEW
4	146040988	VANDALISM	MALICIOUS MISCHIEF, VANDALISM OF VEHICLES	Friday	02/28/2014	09:30	SOUTHERN
5	140176434	MISSING PERSON	FOUND PERSON	Friday	02/28/2014	07:30	TARAVAL
6	140176434	MISSING PERSON	MISSING ADULT	Friday	02/28/2014	07:30	TARAVAL
7	140176246	NON-CRIMINAL	AIDED CASE, MENTAL DISTURBED	Friday	02/28/2014	23:33	SOUTHERN
8	140176224	LARCENY/THEFT	PETTY THEFT SHOPLIFTING	Friday	02/28/2014	12:50	SOUTHERN
9	140176218	OTHER OFFENSES	DRIVERS LICENSE, SUSPENDED OR REVOKED	Friday	02/28/2014	23:36	BAYVIEW
10	140183627	LARCENY/THEFT	GRAND THEFT PURSESNATCH	Friday	02/28/2014	19:45	INGLESIDE
11	140176202	NON-CRIMINAL	LOST PROPERTY	Friday	02/28/2014	21:00	CENTRAL
12	140182237	BURGLARY	BURGLARY, UNLAWFUL ENTRY	Friday	02/28/2014	17:30	TARAVAL
13	140176183	OTHER OFFENSES	DRIVERS LICENSE, SUSPENDED OR REVOKED	Friday	02/28/2014	23:59	CENTRAL
14	140180087	LARCENY/THEFT	GRAND THEFT FROM PERSON	Friday	02/28/2014	19:30	TENDERLOIN
15	140176161	VANDALISM	MALICIOUS MISCHIEF, BREAKING WINDOWS	Friday	02/28/2014	22:59	MISSION
16	140176149	OTHER OFFENSES	DRIVERS LICENSE, SUSPENDED OR REVOKED	Friday	02/28/2014	23:38	MISSION
17	140174030	VANDALISM	MALICIOUS MISCHIEF, VANDALISM OF VEHICLES	Friday	02/28/2014	10:15	BAYVIEW
18	140174030	DISORDERLY CONDU...	DISTURBING THE PEACE	Friday	02/28/2014	10:15	BAYVIEW
19	140173963	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Friday	02/28/2014	09:45	MISSION
20	140173894	DRUG/NARCOTIC	POSSESSION OF NARCOTICS PARAPHERNALIA	Friday	02/28/2014	09:40	TENDERLOIN
21	140173781	BURGLARY	BURGLARY, HOT PROWL, UNLAWFUL ENTRY	Friday	02/28/2014	00:01	TARAVAL
22	140176127	MISSING PERSON	FOUND PERSON	Friday	02/28/2014	22:00	BAYVIEW
23	140173355	OTHER OFFENSES	MISCELLANEOUS INVESTIGATION	Friday	02/28/2014	04:58	CENTRAL

LEFT Function

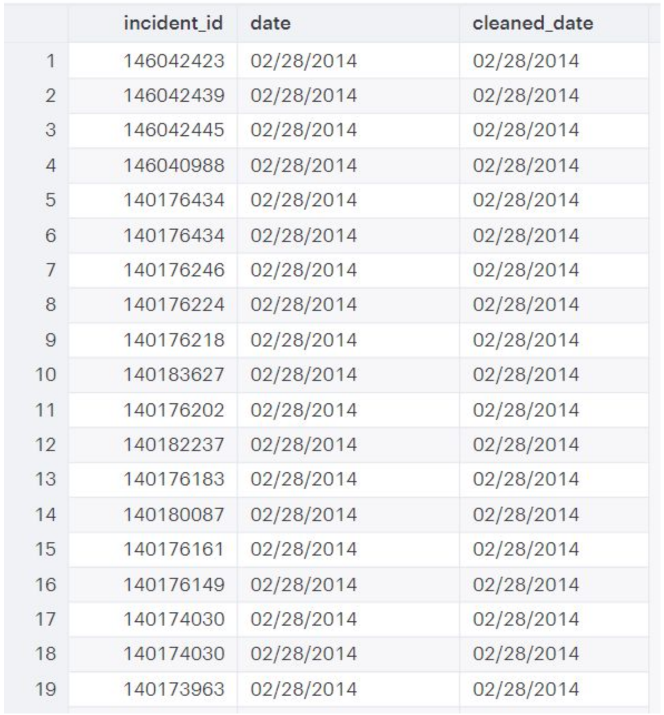
It is used to pull a certain number of characters from the left side of a string and present them as a separate string.

Syntax

LEFT(string, number of characters)

LEFT Function – An Example

```
SELECT
        incident_id,
        date,
        LEFT(date, 10) AS cleaned_date
FROM
         heather.sfpd_incidents
```



	incident_id	date	cleaned_date
1	146042423	02/28/2014	02/28/2014
2	146042439	02/28/2014	02/28/2014
3	146042445	02/28/2014	02/28/2014
4	146040988	02/28/2014	02/28/2014
5	140176434	02/28/2014	02/28/2014
6	140176434	02/28/2014	02/28/2014
7	140176246	02/28/2014	02/28/2014
8	140176224	02/28/2014	02/28/2014
9	140176218	02/28/2014	02/28/2014
10	140183627	02/28/2014	02/28/2014
11	140176202	02/28/2014	02/28/2014
12	140182237	02/28/2014	02/28/2014
13	140176183	02/28/2014	02/28/2014
14	140180087	02/28/2014	02/28/2014
15	140176161	02/28/2014	02/28/2014
16	140176149	02/28/2014	02/28/2014
17	140174030	02/28/2014	02/28/2014
18	140174030	02/28/2014	02/28/2014
19	140173963	02/28/2014	02/28/2014

RIGHT Function

It is used to pull a certain number of characters from the right side of a string and present them as a separate string.

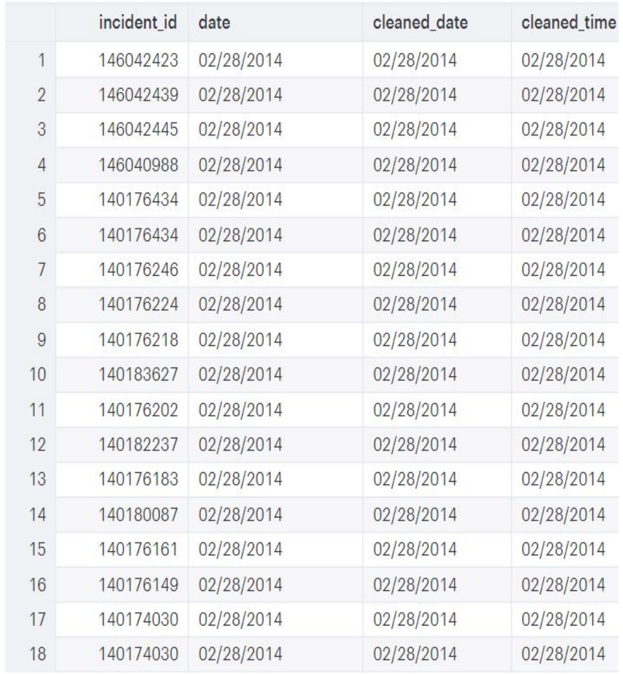
Syntax

RIGHT(string, number of characters)

Relevel
by Unacademy

RIGHT Function – An Example

```
SELECT
        incident_id,
        date,
        LEFT(date, 10) AS cleaned_date,
        RIGHT(date, 17) AS cleaned_time
FROM heather.sfpd_incidents
```



	incident_id	date	cleaned_date	cleaned_time
1	146042423	02/28/2014	02/28/2014	02/28/2014
2	146042439	02/28/2014	02/28/2014	02/28/2014
3	146042445	02/28/2014	02/28/2014	02/28/2014
4	146040988	02/28/2014	02/28/2014	02/28/2014
5	140176434	02/28/2014	02/28/2014	02/28/2014
6	140176434	02/28/2014	02/28/2014	02/28/2014
7	140176246	02/28/2014	02/28/2014	02/28/2014
8	140176224	02/28/2014	02/28/2014	02/28/2014
9	140176218	02/28/2014	02/28/2014	02/28/2014
10	140183627	02/28/2014	02/28/2014	02/28/2014
11	140176202	02/28/2014	02/28/2014	02/28/2014
12	140182237	02/28/2014	02/28/2014	02/28/2014
13	140176183	02/28/2014	02/28/2014	02/28/2014
14	140180087	02/28/2014	02/28/2014	02/28/2014
15	140176161	02/28/2014	02/28/2014	02/28/2014
16	140176149	02/28/2014	02/28/2014	02/28/2014
17	140174030	02/28/2014	02/28/2014	02/28/2014
18	140174030	02/28/2014	02/28/2014	02/28/2014

Length Function

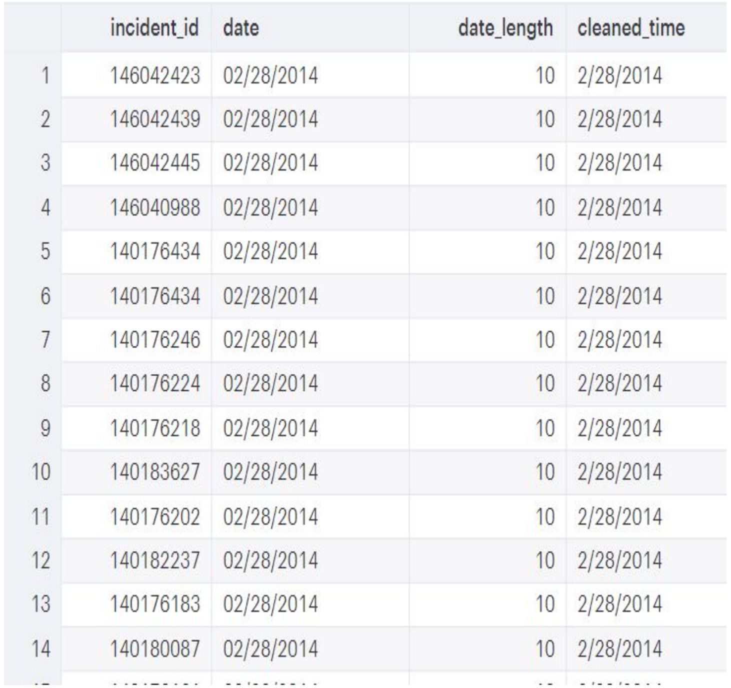
It returns the length of a string.

Syntax

```
LENGTH(string)
```

Length Function – An Example

```
SELECT
        incident_id,
        date,
        LENGTH(date) AS date_length,
        RIGHT(date, LENGTH(date) - 11) AS cleaned_time
FROM heather.sfpd_incidents
```



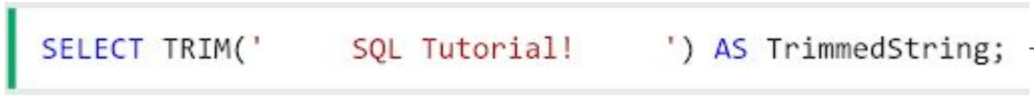
	incident_id	date	date_length	cleaned_time
1	146042423	02/28/2014	10	2/28/2014
2	146042439	02/28/2014	10	2/28/2014
3	146042445	02/28/2014	10	2/28/2014
4	146040988	02/28/2014	10	2/28/2014
5	140176434	02/28/2014	10	2/28/2014
6	140176434	02/28/2014	10	2/28/2014
7	140176246	02/28/2014	10	2/28/2014
8	140176224	02/28/2014	10	2/28/2014
9	140176218	02/28/2014	10	2/28/2014
10	140183627	02/28/2014	10	2/28/2014
11	140176202	02/28/2014	10	2/28/2014
12	140182237	02/28/2014	10	2/28/2014
13	140176183	02/28/2014	10	2/28/2014
14	140180087	02/28/2014	10	2/28/2014

TRIM Function

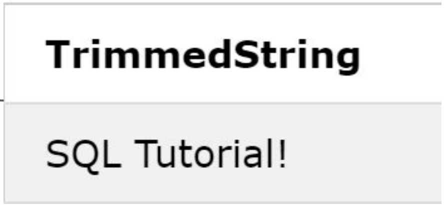
The TRIM() function removes the space character OR other specified characters from the start or end of a string.

Syntax

TRIM([characters FROM]string)



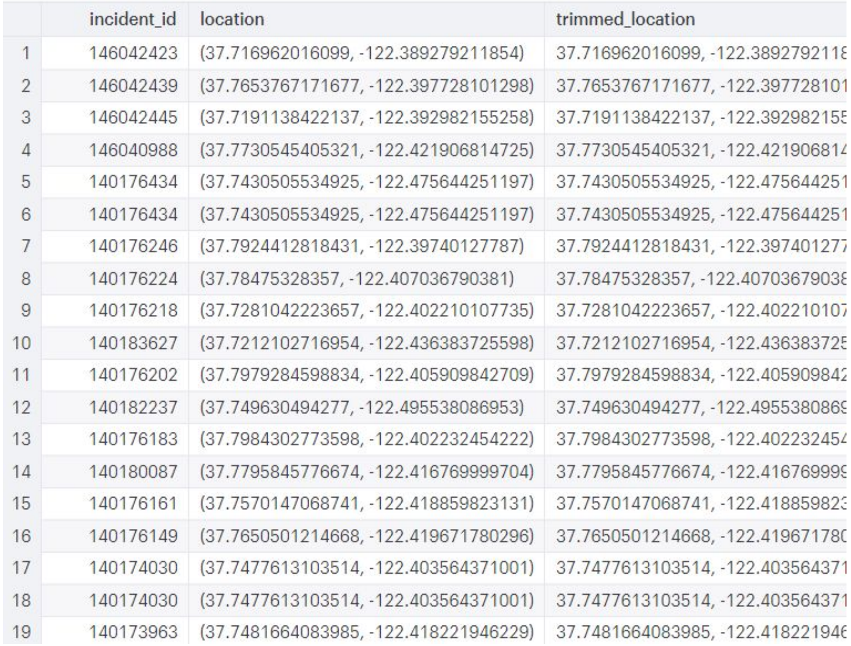
```
SELECT TRIM('      SQL Tutorial!      ') AS TrimmedString;
```



TrimmedString
SQL Tutorial!

TRIM Function – An Example

```
SELECT
        incident_id,
        location,
        TRIM('()'FROM location) AS trimmed_location
FROM
         heather.sfpd_incidents
```



	incident_id	location	trimmed_location
1	146042423	(37.716962016099, -122.389279211854)	37.716962016099, -122.3892792118
2	146042439	(37.7653767171677, -122.397728101298)	37.7653767171677, -122.397728101
3	146042445	(37.7191138422137, -122.392982155258)	37.7191138422137, -122.392982155
4	146040988	(37.7730545405321, -122.421906814725)	37.7730545405321, -122.421906814
5	140176434	(37.7430505534925, -122.475644251197)	37.7430505534925, -122.475644251
6	140176434	(37.7430505534925, -122.475644251197)	37.7430505534925, -122.475644251
7	140176246	(37.7924412818431, -122.39740127787)	37.7924412818431, -122.397401277
8	140176224	(37.78475328357, -122.407036790381)	37.78475328357, -122.4070367903
9	140176218	(37.7281042223657, -122.402210107735)	37.7281042223657, -122.402210107
10	140183627	(37.7212102716954, -122.436383725598)	37.7212102716954, -122.436383725
11	140176202	(37.7979284598834, -122.405909842709)	37.7979284598834, -122.405909842
12	140182237	(37.749630494277, -122.495538086953)	37.749630494277, -122.4955380869
13	140176183	(37.7984302773598, -122.402232454222)	37.7984302773598, -122.402232454
14	140180087	(37.7795845776674, -122.416769999704)	37.7795845776674, -122.416769999
15	140176161	(37.7570147068741, -122.418859823131)	37.7570147068741, -122.418859823
16	140176149	(37.7650501214668, -122.419671780296)	37.7650501214668, -122.419671780
17	140174030	(37.7477613103514, -122.403564371001)	37.7477613103514, -122.403564371
18	140174030	(37.7477613103514, -122.403564371001)	37.7477613103514, -122.403564371
19	140173963	(37.7481664083985, -122.418221946229)	37.7481664083985, -122.418221946

LTRIM and RTRIM Function

The LTRIM() function is used to remove trailing blanks (blank on the left sides).

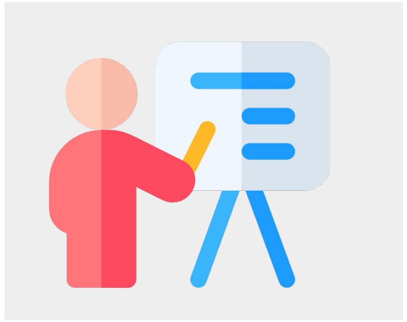
The RTRIM() function is used to remove leading trailing blanks (blank on the right sides).

Instructions for practice questions

- We will use mode.com for these questions
- We will use tutorial.sf_crime_incidents_2014_01 database

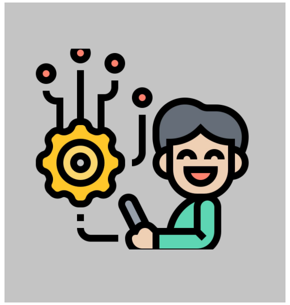
Practice Question

Write a query that separates the `location` field into separate fields for latitude and longitude. You can compare your results against the actual `lat` and `lon` fields in the table.

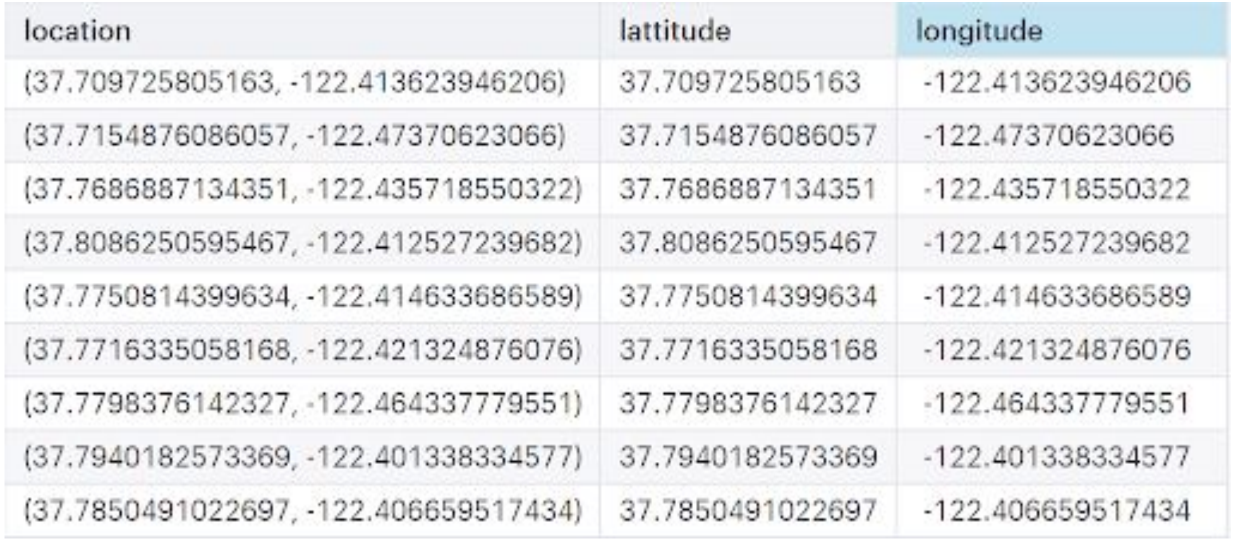


Solution

```
SELECT

        location,

        TRIM(leading '(' FROM LEFT(location, POSITION(',' IN location) - 1)) AS lattitude,

        TRIM(trailing ')' FROM RIGHT(location, LENGTH(location) - POSITION(',' IN location) ) ) AS longitude

 FROM

tutorial.sf_crime_incidents_2014_01
```



Solution



location	lattitude	longitude
(37.709725805163, -122.413623946206)	37.709725805163	-122.413623946206
(37.7154876086057, -122.47370623066)	37.7154876086057	-122.47370623066
(37.7686887134351, -122.435718550322)	37.7686887134351	-122.435718550322
(37.8086250595467, -122.412527239682)	37.8086250595467	-122.412527239682
(37.7750814399634, -122.414633686589)	37.7750814399634	-122.414633686589
(37.7716335058168, -122.421324876076)	37.7716335058168	-122.421324876076
(37.7798376142327, -122.464337779551)	37.7798376142327	-122.464337779551
(37.7940182573369, -122.401338334577)	37.7940182573369	-122.401338334577
(37.7850491022697, -122.406659517434)	37.7850491022697	-122.406659517434

Position Function

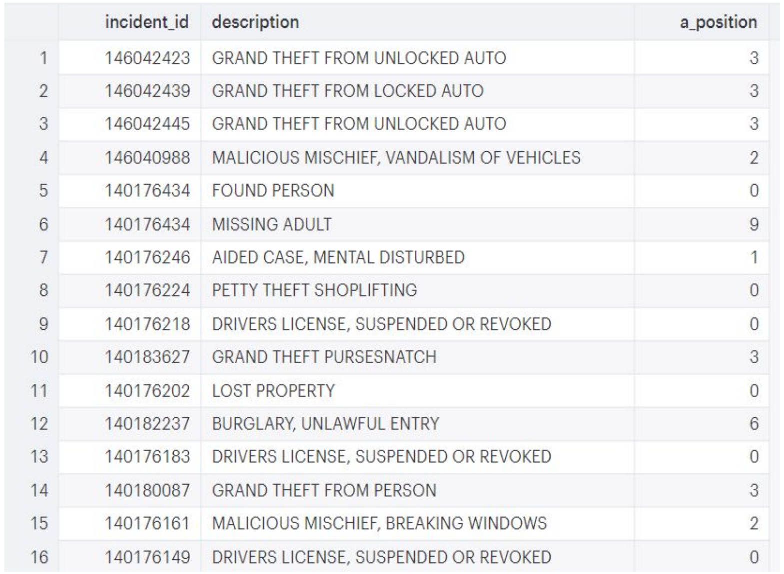
POSITION allows you to specify a substring, then returns a numerical value equal to the character number (counting from left) where that substring first appears in the target string.

Syntax

POSITION(substring IN string)

Position Function – An Example

```
SELECT
        incident_id,
        description,
        POSITION('A' IN description) AS a_position
FROM  heather.sfpd_incidents
```



	incident_id	description	a_position
1	146042423	GRAND THEFT FROM UNLOCKED AUTO	3
2	146042439	GRAND THEFT FROM LOCKED AUTO	3
3	146042445	GRAND THEFT FROM UNLOCKED AUTO	3
4	146040988	MALICIOUS MISCHIEF, VANDALISM OF VEHICLES	2
5	140176434	FOUND PERSON	0
6	140176434	MISSING ADULT	9
7	140176246	AIDED CASE, MENTAL DISTURBED	1
8	140176224	PETTY THEFT SHOPLIFTING	0
9	140176218	DRIVERS LICENSE, SUSPENDED OR REVOKED	0
10	140183627	GRAND THEFT PURSESNATCH	3
11	140176202	LOST PROPERTY	0
12	140182237	BURGLARY, UNLAWFUL ENTRY	6
13	140176183	DRIVERS LICENSE, SUSPENDED OR REVOKED	0
14	140180087	GRAND THEFT FROM PERSON	3
15	140176161	MALICIOUS MISCHIEF, BREAKING WINDOWS	2
16	140176149	DRIVERS LICENSE, SUSPENDED OR REVOKED	0

SUBSTR Function

SUBSTR allows you to specify a substring, then returns a numerical value equal to the character number (counting from left) where that substring first appears in the target string.

Syntax

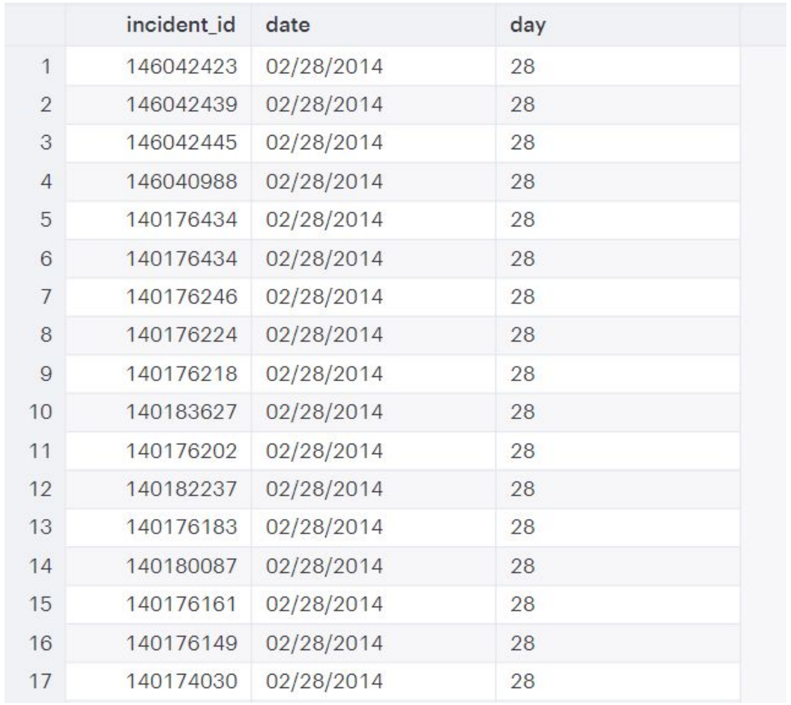
SUBSTRING(string, start, length)

Parameter Values

- String: Required. The string to extract from.
- Start: Required. The start position. The first position in string is 1.
- Length: Required. The number of characters to extract. Must be a positive number.

SUBSTR Function – An Example

```
SELECT
        incident_id,
        date,
        SUBSTR(date, 4, 2) AS day
FROM  heather.sfpd_incidents
```

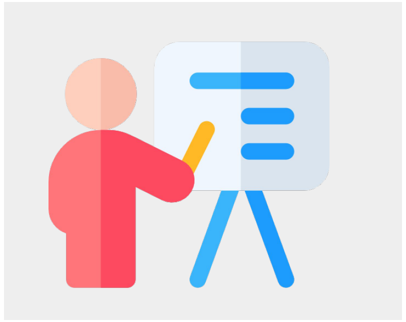


	incident_id	date	day
1	146042423	02/28/2014	28
2	146042439	02/28/2014	28
3	146042445	02/28/2014	28
4	146040988	02/28/2014	28
5	140176434	02/28/2014	28
6	140176434	02/28/2014	28
7	140176246	02/28/2014	28
8	140176224	02/28/2014	28
9	140176218	02/28/2014	28
10	140183627	02/28/2014	28
11	140176202	02/28/2014	28
12	140182237	02/28/2014	28
13	140176183	02/28/2014	28
14	140180087	02/28/2014	28
15	140176161	02/28/2014	28
16	140176149	02/28/2014	28
17	140174030	02/28/2014	28

Relevel
by Unacademy

Practice Question

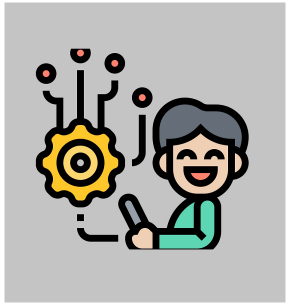
Write a query that creates a date column formatted by YYYY-MM-DD.



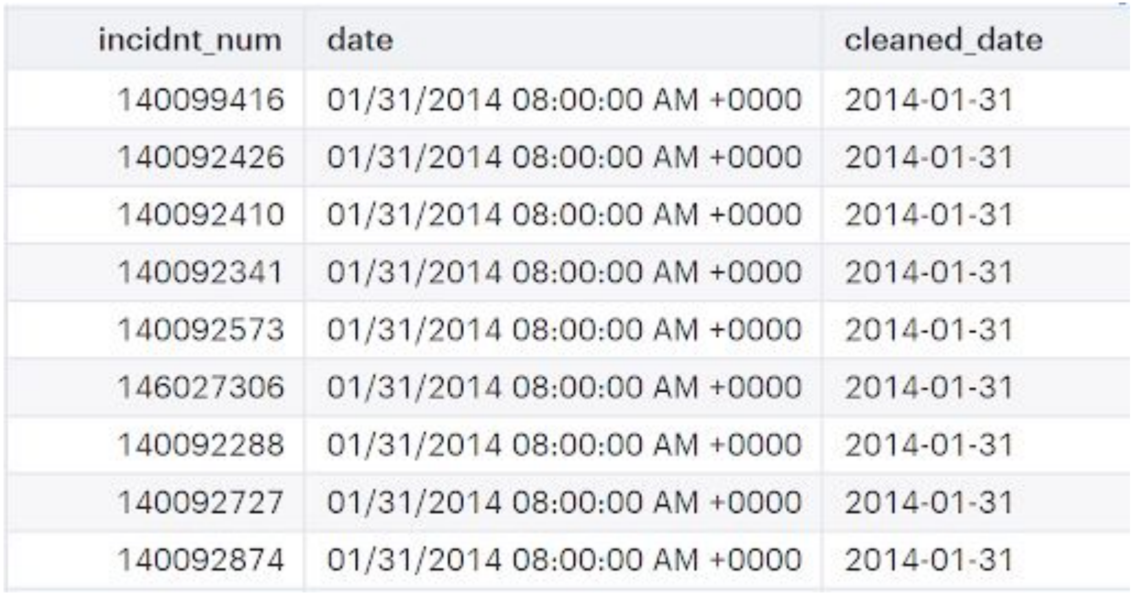
Relevel
by Unacademy

Solution

```
SELECT
        incidnt_num,
        date,
        SUBSTR(date, 7, 4) || '-' || LEFT(date, 2) || '-' || SUBSTR(date, 4, 2) AS cleaned_date
FROM
        tutorial.sf_crime_incidents_2014_01
```



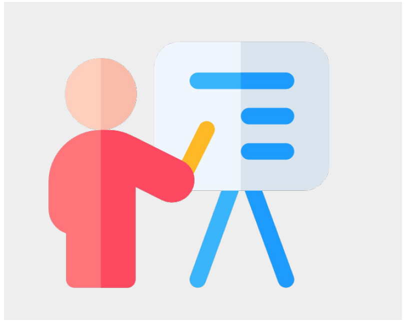
Solution



incidnt_num	date	cleaned_date
140099416	01/31/2014 08:00:00 AM +0000	2014-01-31
140092426	01/31/2014 08:00:00 AM +0000	2014-01-31
140092410	01/31/2014 08:00:00 AM +0000	2014-01-31
140092341	01/31/2014 08:00:00 AM +0000	2014-01-31
140092573	01/31/2014 08:00:00 AM +0000	2014-01-31
146027306	01/31/2014 08:00:00 AM +0000	2014-01-31
140092288	01/31/2014 08:00:00 AM +0000	2014-01-31
140092727	01/31/2014 08:00:00 AM +0000	2014-01-31
140092874	01/31/2014 08:00:00 AM +0000	2014-01-31

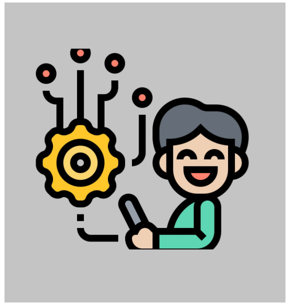
Practice Question

Write a query that creates an accurate timestamp using the date and time columns in tutorial.sf_crime_incidents_2014_01. Include a field that is exactly one week later as well.

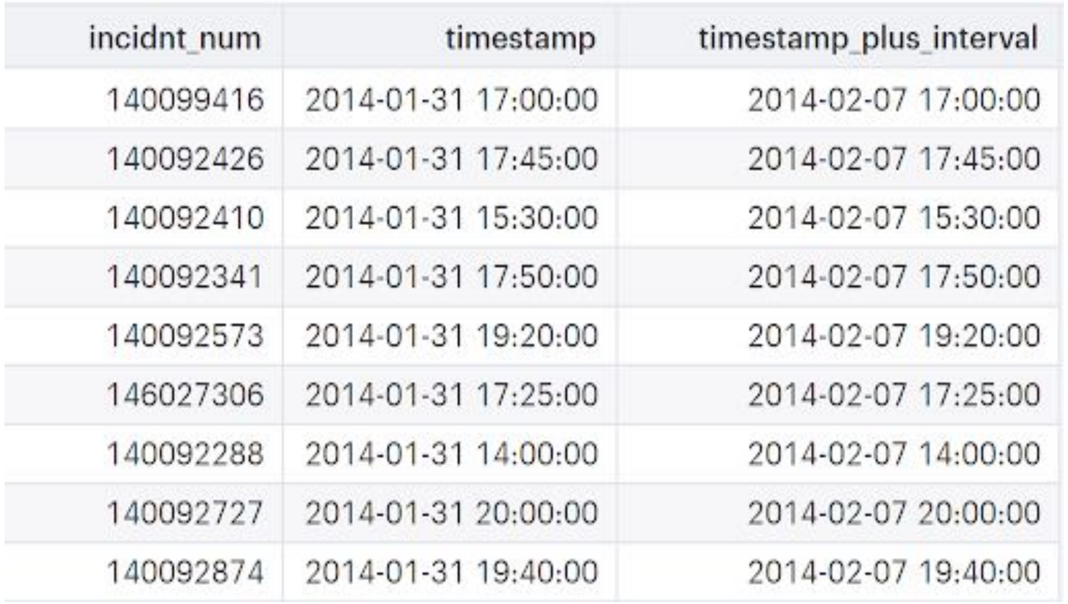


Solution

```
SELECT

         incidnt_num,

     (SUBSTR(date, 7, 4) || '-' || LEFT(date, 2) ||

      '-' || SUBSTR(date, 4, 2) || ' ' || time || ':00')::timestamp AS timestamp,

     (SUBSTR(date, 7, 4) || '-' || LEFT(date, 2) ||

      '-' || SUBSTR(date, 4, 2) || ' ' || time || ':00')::timestamp

      + INTERVAL '1 week' AS timestamp_plus_interval

 FROM tutorial.sf_crime_incidents_2014_01
```



Solution



incidnt_num	timestamp	timestamp_plus_interval
140099416	2014-01-31 17:00:00	2014-02-07 17:00:00
140092426	2014-01-31 17:45:00	2014-02-07 17:45:00
140092410	2014-01-31 15:30:00	2014-02-07 15:30:00
140092341	2014-01-31 17:50:00	2014-02-07 17:50:00
140092573	2014-01-31 19:20:00	2014-02-07 19:20:00
146027306	2014-01-31 17:25:00	2014-02-07 17:25:00
140092288	2014-01-31 14:00:00	2014-02-07 14:00:00
140092727	2014-01-31 20:00:00	2014-02-07 20:00:00
140092874	2014-01-31 19:40:00	2014-02-07 19:40:00

CONCAT Function

The CONCAT() function adds two or more strings together.

Syntax

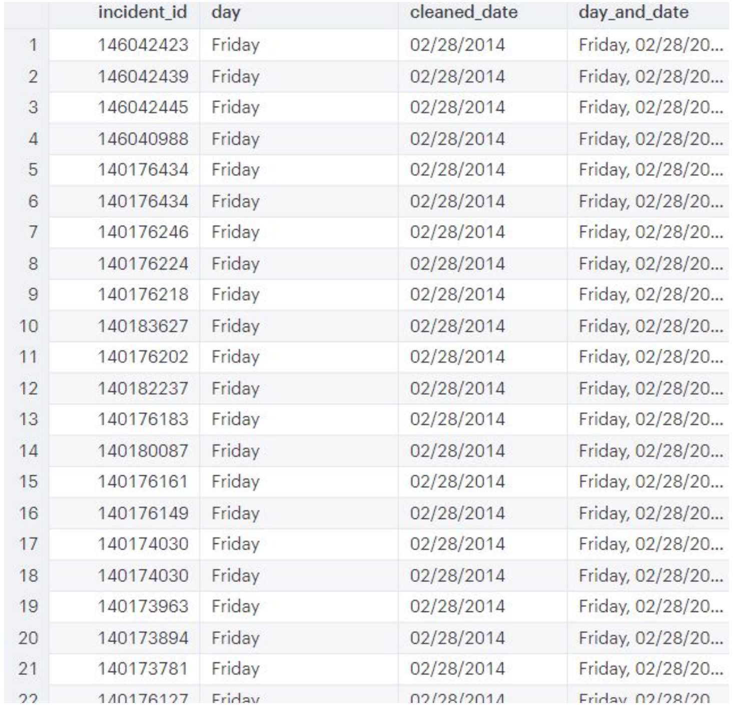
CONCAT(string1, string2, …., string_n)

Parameter Values

- string1, string2, string_n: Required. The strings to add together.

CONCAT Function – An Example

```
SELECT
        incident_id,
        day,
       LEFT(date, 10) AS cleaned_date,
       CONCAT(day, ', ', LEFT(date, 10)) AS day_and_date
FROM heather.sfpd_incidents
```

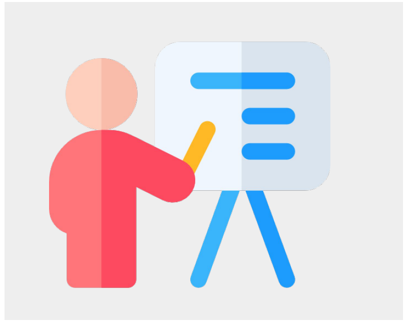


	incident_id	day	cleaned_date	day_and_date
1	146042423	Friday	02/28/2014	Friday, 02/28/20...
2	146042439	Friday	02/28/2014	Friday, 02/28/20...
3	146042445	Friday	02/28/2014	Friday, 02/28/20...
4	146040988	Friday	02/28/2014	Friday, 02/28/20...
5	140176434	Friday	02/28/2014	Friday, 02/28/20...
6	140176434	Friday	02/28/2014	Friday, 02/28/20...
7	140176246	Friday	02/28/2014	Friday, 02/28/20...
8	140176224	Friday	02/28/2014	Friday, 02/28/20...
9	140176218	Friday	02/28/2014	Friday, 02/28/20...
10	140183627	Friday	02/28/2014	Friday, 02/28/20...
11	140176202	Friday	02/28/2014	Friday, 02/28/20...
12	140182237	Friday	02/28/2014	Friday, 02/28/20...
13	140176183	Friday	02/28/2014	Friday, 02/28/20...
14	140180087	Friday	02/28/2014	Friday, 02/28/20...
15	140176161	Friday	02/28/2014	Friday, 02/28/20...
16	140176149	Friday	02/28/2014	Friday, 02/28/20...
17	140174030	Friday	02/28/2014	Friday, 02/28/20...
18	140174030	Friday	02/28/2014	Friday, 02/28/20...
19	140173963	Friday	02/28/2014	Friday, 02/28/20...
20	140173894	Friday	02/28/2014	Friday, 02/28/20...
21	140173781	Friday	02/28/2014	Friday, 02/28/20...
22	140176127	Friday	02/28/2014	Friday, 02/28/20

Practice Question

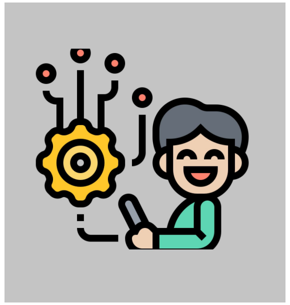
Concatenate the lat and lon fields to form a field that is equivalent to the location field

(Note that the answer will have a different decimal precision).

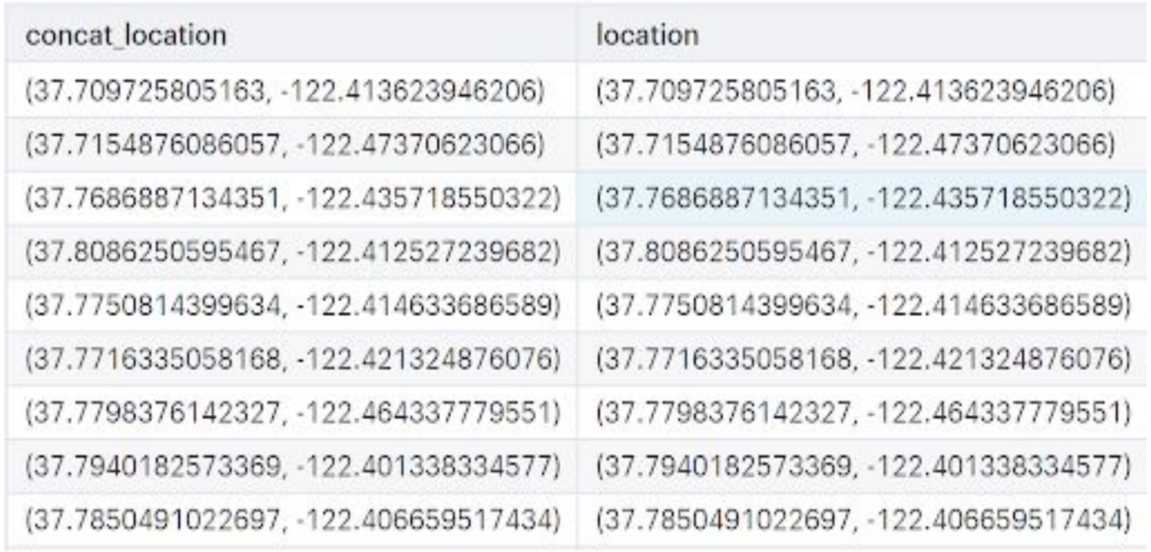


Solution

```
SELECT

  CONCAT('(', lat, ', ', lon, ')') AS concat_location,

  location

FROM

  tutorial.sf_crime_incidents_2014_01
```



Solution



concat_location	location
(37.709725805163, -122.413623946206)	(37.709725805163, -122.413623946206)
(37.7154876086057, -122.47370623066)	(37.7154876086057, -122.47370623066)
(37.7686887134351, -122.435718550322)	(37.7686887134351, -122.435718550322)
(37.8086250595467, -122.412527239682)	(37.8086250595467, -122.412527239682)
(37.7750814399634, -122.414633686589)	(37.7750814399634, -122.414633686589)
(37.7716335058168, -122.421324876076)	(37.7716335058168, -122.421324876076)
(37.7798376142327, -122.464337779551)	(37.7798376142327, -122.464337779551)
(37.7940182573369, -122.401338334577)	(37.7940182573369, -122.401338334577)
(37.7850491022697, -122.406659517434)	(37.7850491022697, -122.406659517434)

UPPER() Function

The UPPER() function converts a string to the upper-case.

Syntax

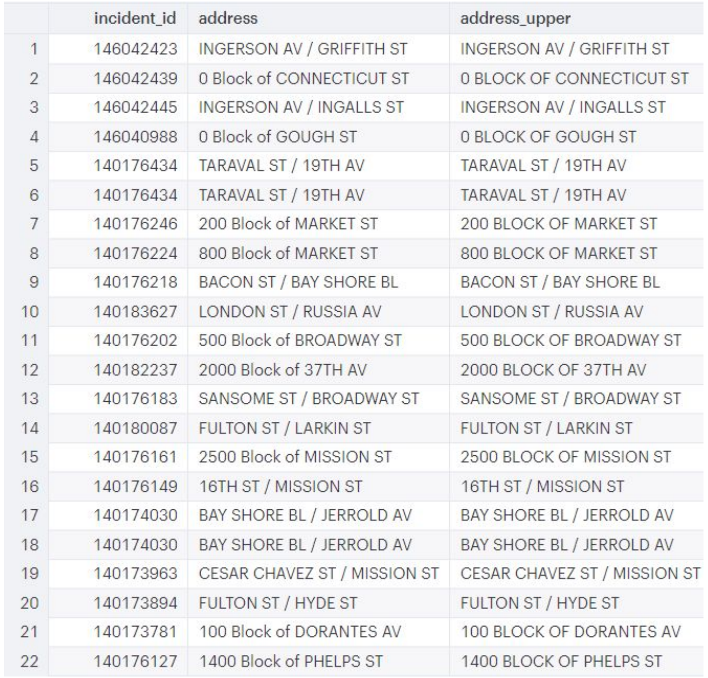
UPPER(text)

Parameter Values

- Text : Required. The string to convert.

UPPER Function – An Example

```
SELECT
        incident_id,
        address,
        UPPER(address) AS address_upper
FROM
        heather.sfpd_incidents
```



	incident_id	address	address_upper
1	146042423	INGERSON AV / GRIFFITH ST	INGERSON AV / GRIFFITH ST
2	146042439	0 Block of CONNECTICUT ST	0 BLOCK OF CONNECTICUT ST
3	146042445	INGERSON AV / INGALLS ST	INGERSON AV / INGALLS ST
4	146040988	0 Block of GOUGH ST	0 BLOCK OF GOUGH ST
5	140176434	TARAVAL ST / 19TH AV	TARAVAL ST / 19TH AV
6	140176434	TARAVAL ST / 19TH AV	TARAVAL ST / 19TH AV
7	140176246	200 Block of MARKET ST	200 BLOCK OF MARKET ST
8	140176224	800 Block of MARKET ST	800 BLOCK OF MARKET ST
9	140176218	BACON ST / BAY SHORE BL	BACON ST / BAY SHORE BL
10	140183627	LONDON ST / RUSSIA AV	LONDON ST / RUSSIA AV
11	140176202	500 Block of BROADWAY ST	500 BLOCK OF BROADWAY ST
12	140182237	2000 Block of 37TH AV	2000 BLOCK OF 37TH AV
13	140176183	SANSOME ST / BROADWAY ST	SANSOME ST / BROADWAY ST
14	140180087	FULTON ST / LARKIN ST	FULTON ST / LARKIN ST
15	140176161	2500 Block of MISSION ST	2500 BLOCK OF MISSION ST
16	140176149	16TH ST / MISSION ST	16TH ST / MISSION ST
17	140174030	BAY SHORE BL / JERROLD AV	BAY SHORE BL / JERROLD AV
18	140174030	BAY SHORE BL / JERROLD AV	BAY SHORE BL / JERROLD AV
19	140173963	CESAR CHAVEZ ST / MISSION ST	CESAR CHAVEZ ST / MISSION ST
20	140173894	FULTON ST / HYDE ST	FULTON ST / HYDE ST
21	140173781	100 Block of DORANTES AV	100 BLOCK OF DORANTES AV
22	140176127	1400 Block of PHELPS ST	1400 BLOCK OF PHELPS ST

LOWER() Function

The LOWER() function converts a string to the lower-case.

Syntax

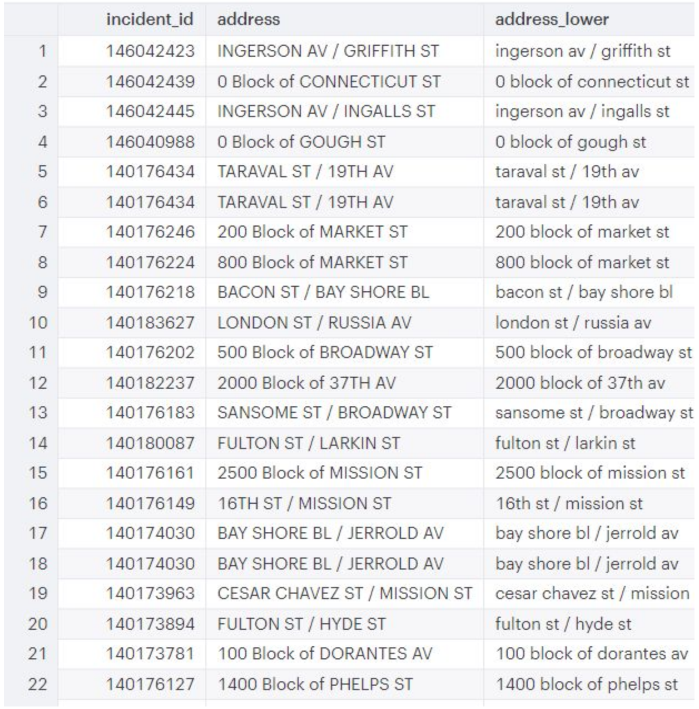
LOWER(text)

Parameter Values

- Text : Required. The string to convert.

LOWER Function – An Example

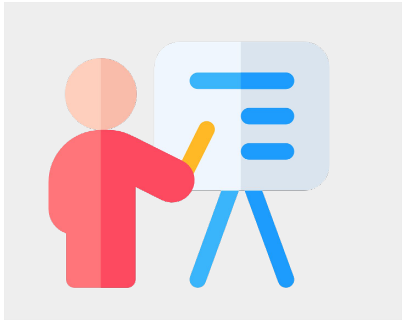
```
SELECT
        incident_id,
        address,
        LOWER(address) AS address_lower
FROM
        heather.sfpd_incidents
```



	incident_id	address	address_lower
1	146042423	INGERSON AV / GRIFFITH ST	ingerson av / griffith st
2	146042439	0 Block of CONNECTICUT ST	0 block of connecticut st
3	146042445	INGERSON AV / INGALLS ST	ingerson av / ingalls st
4	146040988	0 Block of GOUGH ST	0 block of gough st
5	140176434	TARAVAL ST / 19TH AV	taraval st / 19th av
6	140176434	TARAVAL ST / 19TH AV	taraval st / 19th av
7	140176246	200 Block of MARKET ST	200 block of market st
8	140176224	800 Block of MARKET ST	800 block of market st
9	140176218	BACON ST / BAY SHORE BL	bacon st / bay shore bl
10	140183627	LONDON ST / RUSSIA AV	london st / russia av
11	140176202	500 Block of BROADWAY ST	500 block of broadway st
12	140182237	2000 Block of 37TH AV	2000 block of 37th av
13	140176183	SANSOME ST / BROADWAY ST	sansome st / broadway st
14	140180087	FULTON ST / LARKIN ST	fulton st / larkin st
15	140176161	2500 Block of MISSION ST	2500 block of mission st
16	140176149	16TH ST / MISSION ST	16th st / mission st
17	140174030	BAY SHORE BL / JERROLD AV	bay shore bl / jerrold av
18	140174030	BAY SHORE BL / JERROLD AV	bay shore bl / jerrold av
19	140173963	CESAR CHAVEZ ST / MISSION ST	cesar chavez st / mission
20	140173894	FULTON ST / HYDE ST	fulton st / hyde st
21	140173781	100 Block of DORANTES AV	100 block of dorantes av
22	140176127	1400 Block of PHELPS ST	1400 block of phelps st

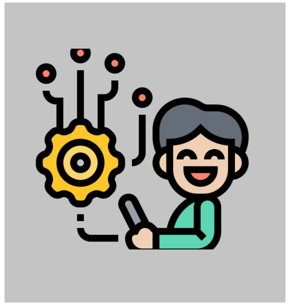
Practice Question

Write a query that returns the `category` field, but with the first letter capitalized and the rest of the letters in lower-case.

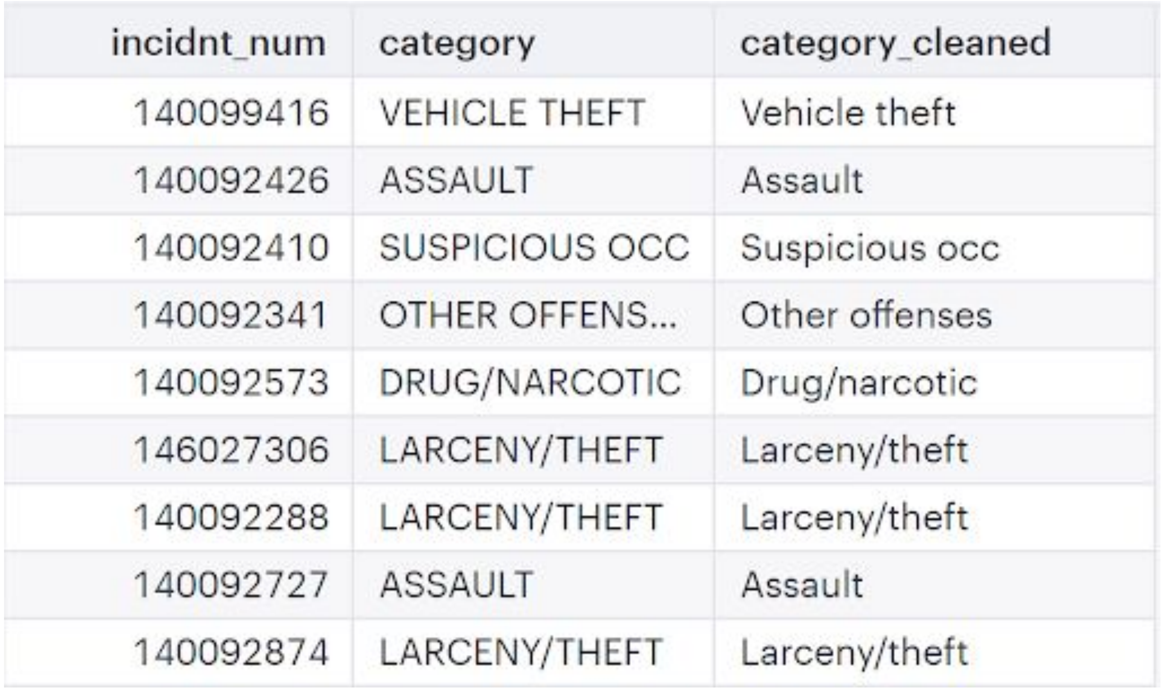


Solution

```
SELECT

    incidnt_num,

    category,

    UPPER(LEFT(category, 1)) || LOWER(RIGHT(category, LENGTH(category) - 1)) AS category_cleaned

FROM

    tutorial.sf_crime_incidents_2014_01
```



Solution



incidnt_num	category	category_cleaned
140099416	VEHICLE THEFT	Vehicle theft
140092426	ASSAULT	Assault
140092410	SUSPICIOUS OCC	Suspicious occ
140092341	OTHER OFFENS...	Other offenses
140092573	DRUG/NARCOTIC	Drug/narcotic
146027306	LARCENY/THEFT	Larceny/theft
140092288	LARCENY/THEFT	Larceny/theft
140092727	ASSAULT	Assault
140092874	LARCENY/THEFT	Larceny/theft

COALESCE() Function

The COALESCE() function returns the first non-null value in a list.

Syntax

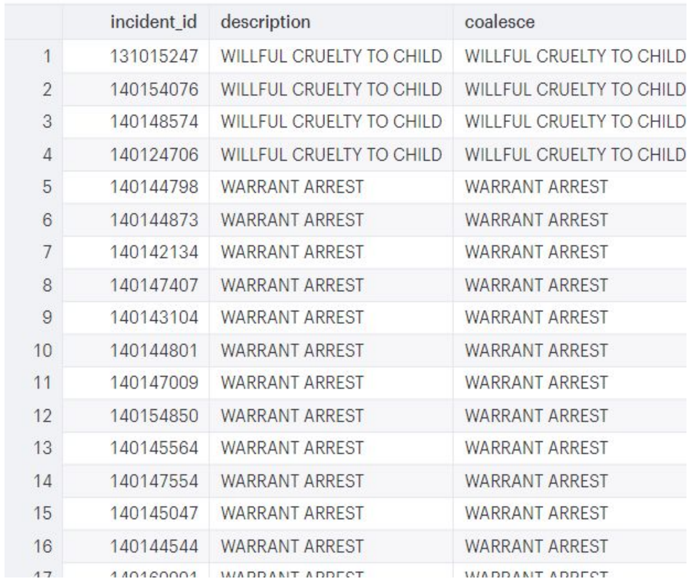
COALESCE(val1, val2, ..., val_n)

Parameter Values

- Val1, val2, val3 : Required. The string to collate.

COALESCE() Function – An Example

```
SELECT
        incident_id,
        description,
        COALESCE(descript, 'No Description')
FROM
        Heather.sfpd_incidents ORDER BY description DESC
```



	incident_id	description	coalesce
1	131015247	WILLFUL CRUELTY TO CHILD	WILLFUL CRUELTY TO CHILD
2	140154076	WILLFUL CRUELTY TO CHILD	WILLFUL CRUELTY TO CHILD
3	140148574	WILLFUL CRUELTY TO CHILD	WILLFUL CRUELTY TO CHILD
4	140124706	WILLFUL CRUELTY TO CHILD	WILLFUL CRUELTY TO CHILD
5	140144798	WARRANT ARREST	WARRANT ARREST
6	140144873	WARRANT ARREST	WARRANT ARREST
7	140142134	WARRANT ARREST	WARRANT ARREST
8	140147407	WARRANT ARREST	WARRANT ARREST
9	140143104	WARRANT ARREST	WARRANT ARREST
10	140144801	WARRANT ARREST	WARRANT ARREST
11	140147009	WARRANT ARREST	WARRANT ARREST
12	140154850	WARRANT ARREST	WARRANT ARREST
13	140145564	WARRANT ARREST	WARRANT ARREST
14	140147554	WARRANT ARREST	WARRANT ARREST
15	140145047	WARRANT ARREST	WARRANT ARREST
16	140144544	WARRANT ARREST	WARRANT ARREST
17	[illegible]	WARRANT ARREST	WARRANT ARREST

Pivoting Data in SQL

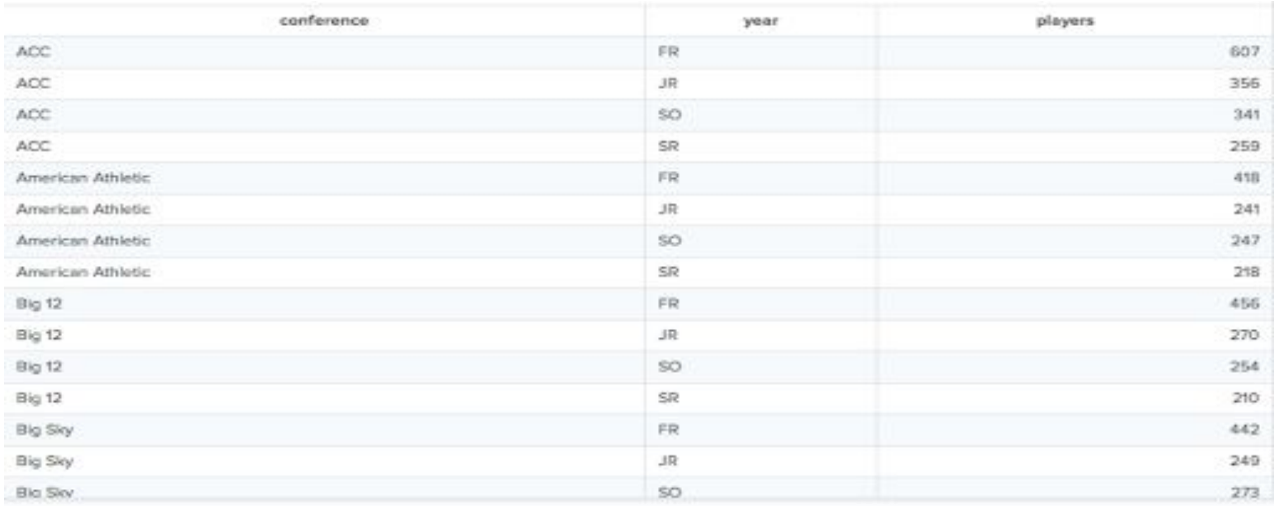
Under this topic, we will learn about two pivots:

- Pivoting rows to columns
- Pivoting columns to rows

Pivoting Rows to Columns

This lesson will teach you how to take data that is formatted for analysis and pivot it for presentation or charting.

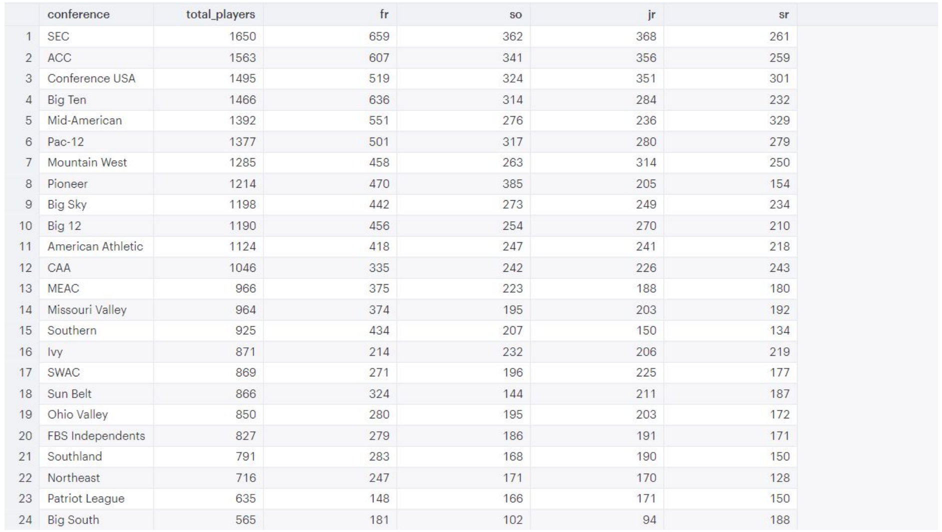
We'll take a dataset that looks like this:



conference	year	players
ACC	FR	607
ACC	JR	356
ACC	SO	341
ACC	SR	259
American Athletic	FR	418
American Athletic	JR	241
American Athletic	SO	247
American Athletic	SR	218
Big 12	FR	456
Big 12	JR	270
Big 12	SO	254
Big 12	SR	210
Big Sky	FR	442
Big Sky	JR	249
Big Sky	SO	273

Pivoting Rows to Columns

And make it look like this:



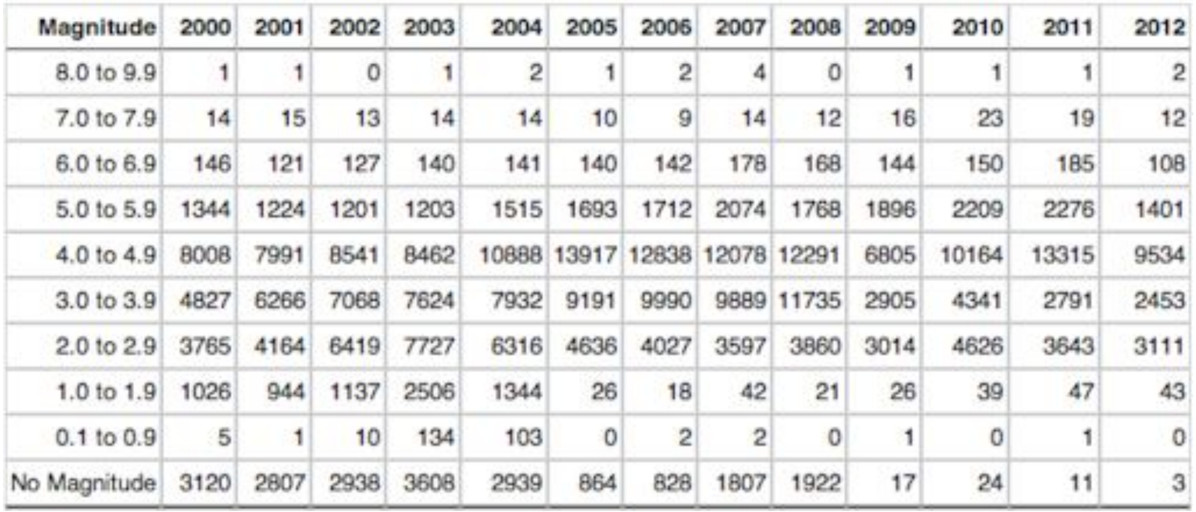
	conference	total_players	fr	so	jr	sr
1	SEC	1650	659	362	368	261
2	ACC	1563	607	341	356	259
3	Conference USA	1495	519	324	351	301
4	Big Ten	1466	636	314	284	232
5	Mid-American	1392	551	276	236	329
6	Pac-12	1377	501	317	280	279
7	Mountain West	1285	458	263	314	250
8	Pioneer	1214	470	385	205	154
9	Big Sky	1198	442	273	249	234
10	Big 12	1190	456	254	270	210
11	American Athletic	1124	418	247	241	218
12	CAA	1046	335	242	226	243
13	MEAC	966	375	223	188	180
14	Missouri Valley	964	374	195	203	192
15	Southern	925	434	207	150	134
16	Ivy	871	214	232	206	219
17	SWAC	869	271	196	225	177
18	Sun Belt	866	324	144	211	187
19	Ohio Valley	850	280	195	203	172
20	FBS Independents	827	279	186	191	171
21	Southland	791	283	168	190	150
22	Northeast	716	247	171	170	128
23	Patriot League	635	148	166	171	150
24	Big South	565	181	102	94	188

Pivoting Rows to Columns - Query

```
SELECT
        conference,
        SUM(players) AS total_players,
        SUM(CASE WHEN year = 'FR' THEN players ELSE NULL END) AS fr,
        SUM(CASE WHEN year = 'SO' THEN players ELSE NULL END) AS so,
         SUM(CASE WHEN year = 'JR' THEN players ELSE NULL END) AS jr,
        SUM(CASE WHEN year = 'SR' THEN players ELSE NULL END) AS sr
FROM (
        SELECT
                teams.conference AS conference, players.year,
                COUNT(1) AS players
        FROM
                benn.college_football_players players JOIN benn.college_football_teams teams
        ON teams.school_name = players.school_name GROUP BY 1,2
) sub
 GROUP BY 1 ORDER BY 2 DESC
```

Pivoting columns to Rows

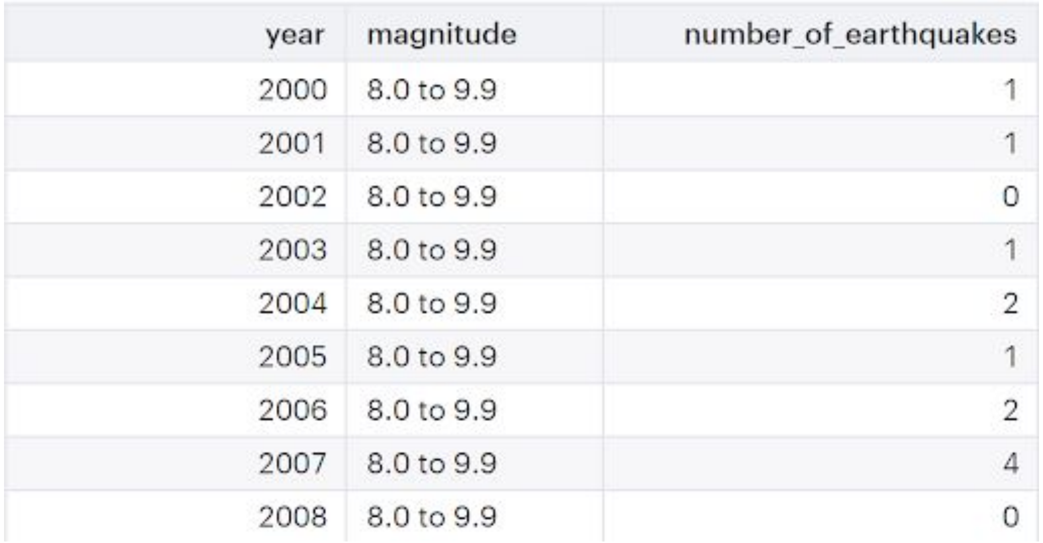
This lesson will teach you how to take data that is formatted for analysis and pivot it for presentation or charting.
We'll take a dataset that looks like this:



Magnitude	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012
8.0 to 9.9	1	1	0	1	2	1	2	4	0	1	1	1	2
7.0 to 7.9	14	15	13	14	14	10	9	14	12	16	23	19	12
6.0 to 6.9	146	121	127	140	141	140	142	178	168	144	150	185	108
5.0 to 5.9	1344	1224	1201	1203	1515	1693	1712	2074	1768	1896	2209	2276	1401
4.0 to 4.9	8008	7991	8541	8462	10888	13917	12838	12078	12291	6805	10164	13315	9534
3.0 to 3.9	4827	6266	7068	7624	7932	9191	9990	9889	11735	2905	4341	2791	2453
2.0 to 2.9	3765	4164	6419	7727	6316	4636	4027	3597	3860	3014	4626	3643	3111
1.0 to 1.9	1026	944	1137	2506	1344	26	18	42	21	26	39	47	43
0.1 to 0.9	5	1	10	134	103	0	2	2	0	1	0	1	0
No Magnitude	3120	2807	2938	3608	2939	864	828	1807	1922	17	24	11	3

Pivoting columns to Rows

And make it look like this:



year	magnitude	number_of_earthquakes
2000	8.0 to 9.9	1
2001	8.0 to 9.9	1
2002	8.0 to 9.9	0
2003	8.0 to 9.9	1
2004	8.0 to 9.9	2
2005	8.0 to 9.9	1
2006	8.0 to 9.9	2
2007	8.0 to 9.9	4
2008	8.0 to 9.9	0

Pivoting Columns to Rows - Query

```
SELECT
        years.*,
        earthquakes.magnitude,
        CASE year
                WHEN 2000 THEN year_2000
                WHEN 2001 THEN year_2001
                WHEN 2002 THEN year_2002
                WHEN 2003 THEN year_2003
                WHEN 2004 THEN year_2004
                WHEN 2005 THEN year_2005
                WHEN 2006 THEN year_2006
                WHEN 2007 THEN year_2007
                WHEN 2008 THEN year_2008
                WHEN 2009 THEN year_2009
                WHEN 2010 THEN year_2010
                WHEN 2011 THEN year_2011
                WHEN 2012 THEN year_2012
                ELSE NULL END         AS number_of_earthquakes   FROM tutorial.worldwide_earthquakes earthquakes CROSS
JOIN   (                SELECT   year                      FROM   (VALUES   (2000),(2001),(2002),(2003),(2004),(2005),(2006),
(2007),(2008),(2009),(2010),(2011),(2012)) v(year)     ) years
```

Relevel
by Unacademy

Thank You