Bertelsmann Tech Scholarship

30.08 Video: CONCAT -

CONCAT

COMBINES VALUES FROM SEVERAL COLUMNS INTO ONE COLUMN

```
SELECT first_name,
       last_name,
       CONCAT(first_name,' ',last_name) AS full_name,
       first_name || ' ' || last_name AS full_name_alt
  FROM demo.customer_data
```

- Piping || // pipe character
- Will combine cols together across rows, 1st & last names stored in separate cols can combined together to create a full name: CONCAT(first_name, ' ', last_name) or with piping as first_name || ' ' || last_name.

30.09 Quiz: CONCAT

1. Each company in the accounts table wants to create an email address for each primary_poc. The email address should be the first name of the primary_poc . last name primary_poc @ company name .com.

```
WITH t1 AS (SELECT LEFT(primary_poc,     STRPOS(primary_poc, ' ') -1 ) first_name,
                   RIGHT(primary_poc, LENGTH(primary_poc) - STRPOS(primary_poc, ' ')) last_name, name
            FROM accounts)
SELECT first_name, last_name, CONCAT(first_name, '.', last_name, '@', name, '.com')
FROM t1;
```

2. You may have noticed that in the previous solution some of the company names include spaces, which will certainly not work in an email address. See if you can create an email address that will work by removing all of the spaces in the account name, but otherwise your solution should be just as in question 1. Some helpful documentation is here.

```
WITH t1 AS (SELECT LEFT(primary_poc,     STRPOS(primary_poc, ' ') -1 ) first_name,
                   RIGHT(primary_poc, LENGTH(primary_poc) - STRPOS(primary_poc, ' ')) last_name, name
            FROM accounts)
SELECT first_name, last_name, CONCAT(first_name, '.', last_name, '@', REPLACE(name, ' ', ''), '.com')
FROM t1;
```

3. We would also like to create an initial password, which they will change after their first log in. The first password will be the first letter of the primary_poc's first name (lowercase), then the last letter of their first name (lowercase), the first letter of their last name (lowercase), the last letter of their last name (lowercase), the number of letters in their first name, the number of letters in their last name, and then the name of the company they are working with, all capitalized with no spaces.

```
WITH t1 AS (SELECT LEFT(primary_poc,     STRPOS(primary_poc, ' ') -1 ) first_name,
                   RIGHT(primary_poc, LENGTH(primary_poc) - STRPOS(primary_poc, ' ')) last_name, name
            FROM accounts)
SELECT first_name, last_name, CONCAT(first_name, '.', last_name, '@', name, '.com'),
       LEFT(LOWER(first_name), 1) || RIGHT(LOWER(first_name), 1) || LEFT(LOWER(last_name), 1) ||
       RIGHT(LOWER(last_name), 1) || LENGTH(first_name) || LENGTH(last_name) || REPLACE(UPPER(name), ' ', '')
FROM t1;
```

30.11 Video: CAST

DATE_PART('month', TO_DATE(month, 'month'))

CAST

ALLOWS US TO CHANGE COLUMNS FROM ONE DATA TYPE TO ANOTHER

BOTH 'CAST' AND '::' ALLOW FOR THE CONVERTING OF ONE DATA TYPE TO ANOTHER

LEFT, RIGHT, OR SUBSTRING AUTOMATICALLY CAST DATA TO A STRING DATA TYPE

- additional functionality for work w/ dates > TO_DATE > CAST > Casting with `::`

DATE_PART('month', TO_DATE(month, 'month')) changed a month name into number associated with that particular month.

- change a string to a date using CAST. CAST change column types.

CAST(date_column AS DATE), or can use date_column::DATE

* LEFT, RIGHT, and TRIM are all used to select only certain elements of strings, but using them to select elements of a number or date will treat them as strings for the purpose of the function.

- TRIM used to remove characters from the beginning and end of a string, or unwanted spaces at the beginning or end of a row that often happen with data being moved from Excel or other storage systems.

30.12 Quiz: CAST

`4.` Write a query to change the date into the correct SQL date format.

You will need to use at least **SUBSTR** and **CONCAT** to perform this operation.

`5.` Once you have created a column in the correct format, use either `CAST` or `::` to convert this to a date.

`4.`

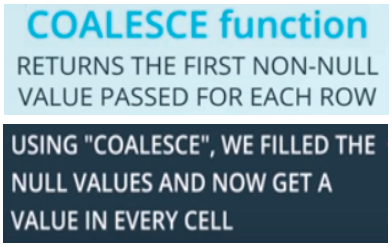
```
SELECT date orig_date,
(SUBSTR(date, 7, 4) || '-' || LEFT(date, 2) || '-' || SUBSTR(date, 4, 2)) new_date
FROM sf_crime_data;
```

`5.`

```
SELECT date orig_date,
(SUBSTR(date, 7, 4) || '-' || LEFT(date, 2) || '-' || SUBSTR(date, 4, 2))::DATE new_date
FROM sf_crime_data;
```

30.14 Video: COALESCE > returns the first non-NULL value passed for each row.
Hence why the video used no_poc if the value in the row was NULL.

```
SELECT COUNT(primary_poc) AS regular_count,
       COUNT(COALESCE(primary_poc, 'no POC')) AS modified_count
  FROM demo.accounts
```



30.15

30.17 Video + Text: Recap
- Learn from what you have done in solving previous problems to solve new problem.
* SQL functions > https://www.w3schools.com/sql/
* Using SQL String Functions to Clean Data > https://mode.com/sql-tutorial/sql-string-functions-for-cleaning/

```
SELECT *,
       DATE_PART('month',TO_DATE(month, 'month')) AS clean_month,
       year || '-' || DATE_PART('month',TO_DATE(month, 'month')) || '-' || day AS concatenated_date,
       CAST(year || '-' || DATE_PART('month',TO_DATE(month, 'month')) || '-' || day AS date) AS formatted_date,
       (year || '-' || DATE_PART('month',TO_DATE(month, 'month')) || '-' || day)::date AS formatted_date_alt
  FROM demo.ad_clicks
```