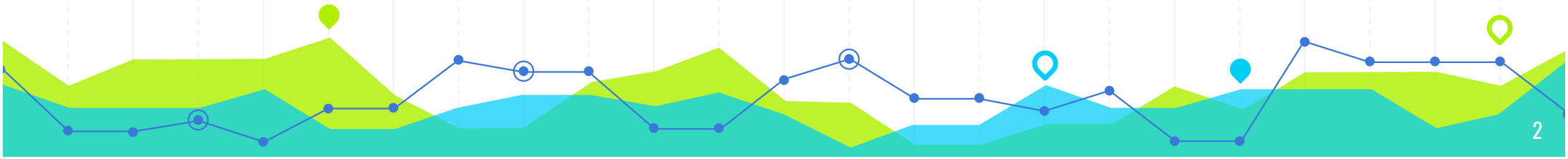
Lab 6: SQL functions

Character functions

- Character functions are functions that transform strings in SQL into different formats than the way they stored in the table.
- You use character functions mainly to compare, join, search, and extract a segment of a string or a value in a column.
- Several character functions are available in SQL programmers.

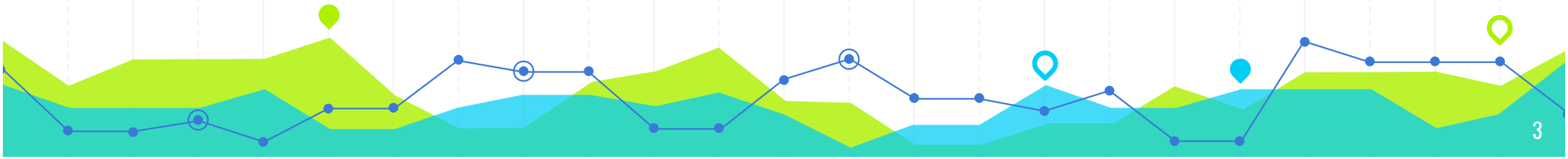


Concatenation functions

- Concatenation is the process of combining two strings into one.
- **CONCAT**(String1, String2, ...)

Example

- **SELECT CONCAT**('CSS225 ', 'IS ', 'EASY') **AS** ConcatenationResult;
- **SELECT CONCAT**(name, ' come from ', dept_name, ' department.') **FROM** instructor;



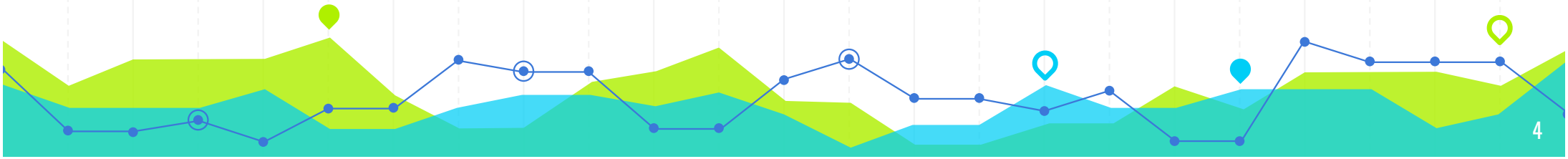
Case of data functions

Upper case function

- **UPPER**(String1), **UCASE**(String1)

Example

- **SELECT UPPER**('css225 ') **AS** UpperResult;
- **SELECT UCASE**(name) **FROM** instructor;



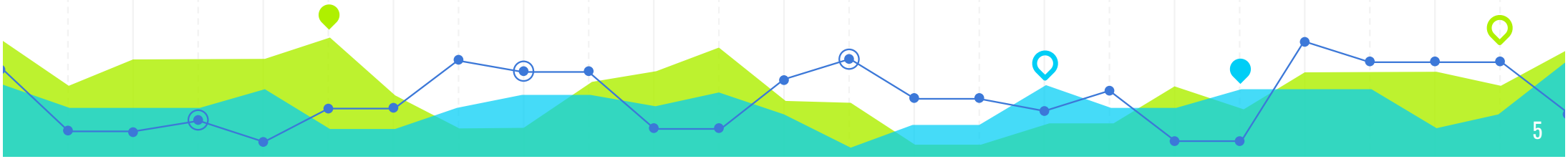
Case of data functions

Lower case function

- **LOWER**(String1), **LCASE**(String1)

Example

- **SELECT LOWER**('CSS225 ') **AS** LowerResult;
- **SELECT LCASE**(name) **FROM** instructor;



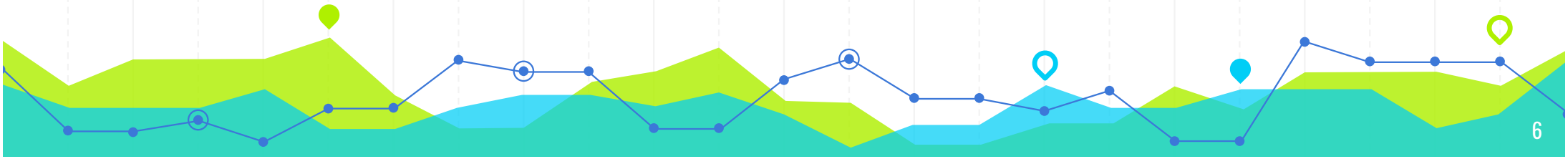
Substring functions

Syntax

SUBSTR(String1, Starting_position, Length)

Example

- **SELECT SUBSTR**('CSS225 is easy', 1, 6) **AS** Result;
- **SELECT SUBSTR**(name, 1, 3) **FROM** instructor;



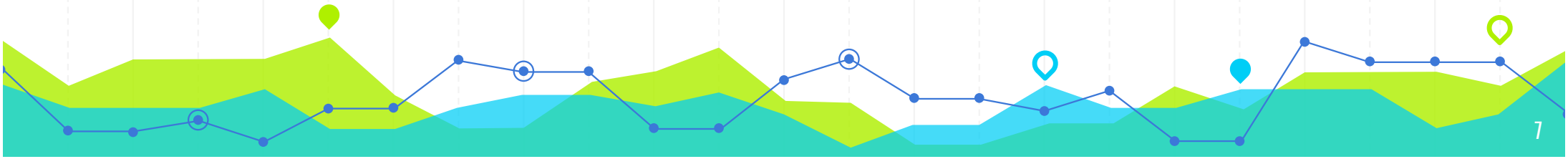
Replace functions

Syntax

REPLACE(String, oldString, newString)

Example

- **SELECT REPLACE**('CSS225 is easy', '225', '222') **AS** Result;
- **SELECT REPLACE**(dept_name, '.', '') **FROM** instructor;



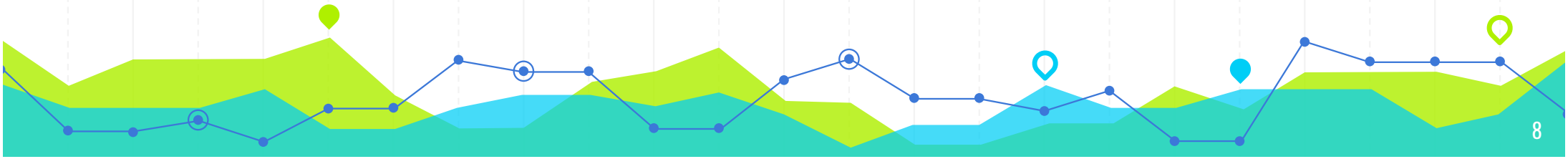
Trim functions

Syntax

TRIM(String), **LTRI**M(String), **RTRIM**(String)

Example

- **SELECT TRIM**(’ CSS225 is easy ’) **AS** Result;
- **SELECT LTRIM**(’ CSS225 is easy ’) **AS** Result;
- **SELECT RTRIM**(’ CSS225 is easy ’) **AS** Result;



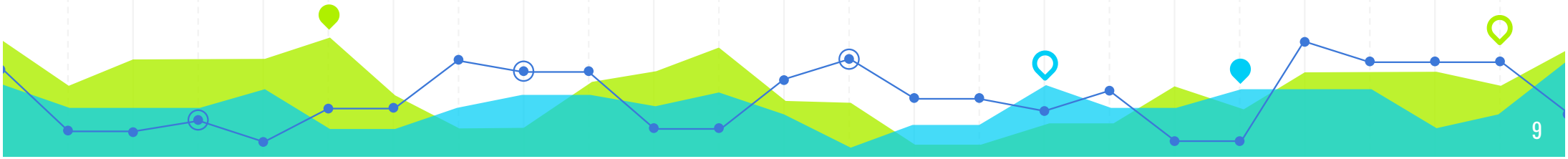
Miscellaneous functions

LENGTH function

LENGTH(String)

Example

- **SELECT LENGTH**('CSS225 is easy') **AS** Result;
- **SELECT LENGTH**(' CSS225 is easy ') **AS** Result;



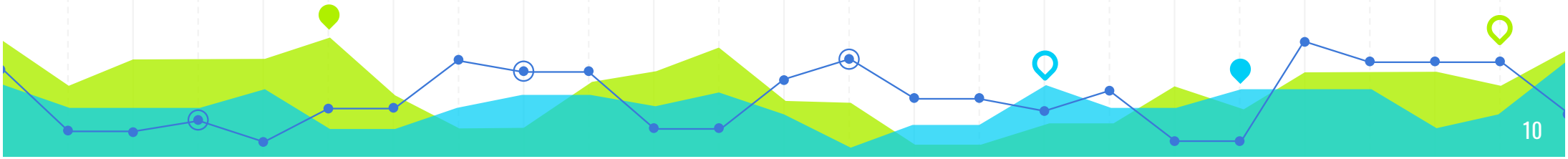
Miscellaneous functions

COALESCE function

- The COLALESCE function replaces NULL values within the result set.

Example

- **SELECT COALESCE**(address2,'none') new_address **FROM** address;
- **SELECT** address, **COALESCE**(address2,'none') new_address **FROM** address;

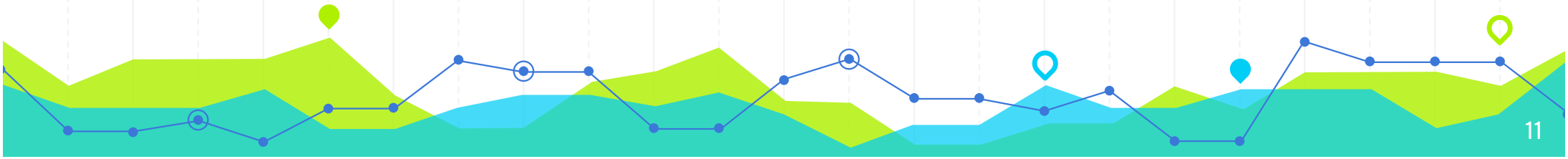


Miscellaneous functions

- The padding function adds characters or spaces to a string.
  LPAD(String, total_length,character)
  RPAD(String, total_length,character)

Example

- **SELECT** LPAD('CSS225 is easy', 25,'.') **AS** Result;
- **SELECT** RPAD('CSS225 is easy', 25,' ') **AS** Result;



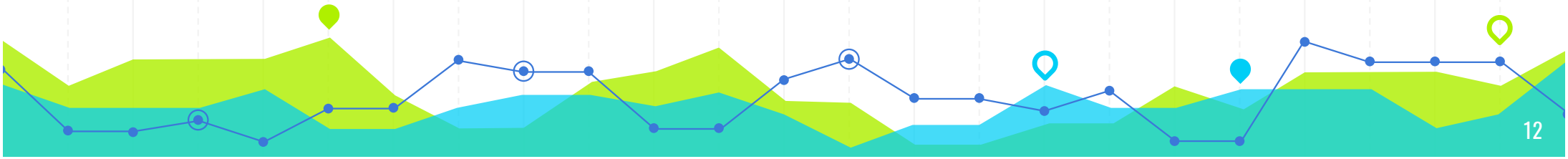
Miscellaneous functions

- The ASCII function returns the ASCII representation of the leftmost character of a string.

  ASCII(String)

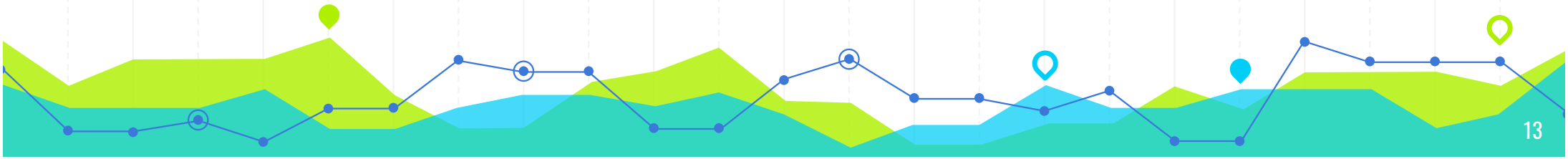
Example

- **SELECT ASCII**('C') **AS** Result;
- **SELECT ASCII**('CSS225 is easy', 25,' ') **AS** Result;



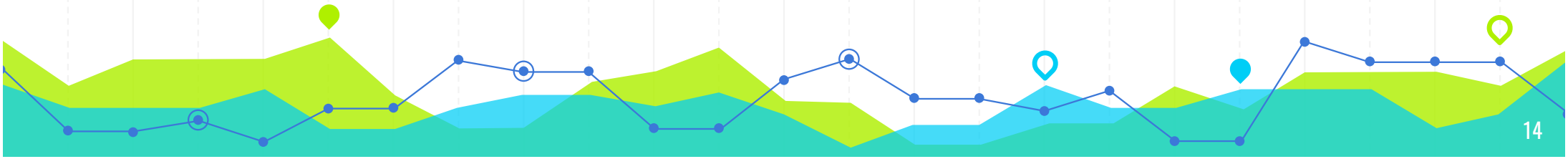
Mathematical functions

- Absolute value – **ABS**()
- Rounding – **ROUND**()
- Square root – **SQRT**()
- Sign value – **SIGN**()
- Power – **POWER**()
- Ceiling and floor values – **CEIL**(), **FLOOR**()
- Exponential values – **EXP**()
- Modulo - **MOD**()
- **SIN**, **COS**, **TAN**



Date and Time Functions

MySQL Date and Time data types	
DATETIME	YYYY-MM-DD HH:MM:SS
DATE	YYYY-MM-DD
TIMESTAMP	YYYYMMDDHHSSMM
TIME	HH:MM:SS
YEAR	YYYY



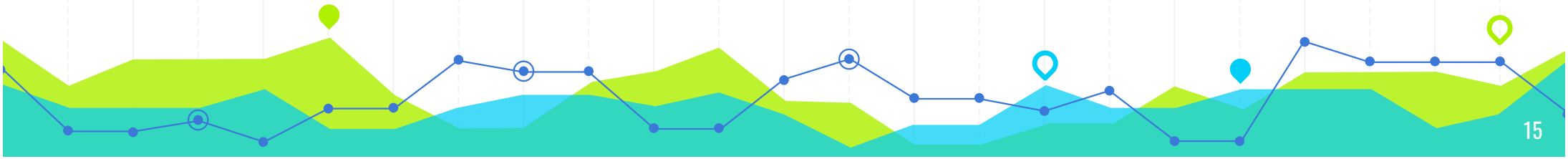
Date and Time Functions

- If you enter a date in a format other than the Year-Month-Day format then it might work, but it won't be storing them as expect!
- Changing the format of the date using the DATE_FORMAT() function.

  Syntax

  DATE_FORMAT(date,format)

  Example
- **SELECT DISTINCT**(**DATE_FORMAT**(RENTAL_DATE, '%D %b %Y')) **FROM** RENTAL;



Date and Time Functions

Specifier	Description
%a	Abbreviated weekday name (Sun..Sat)
%b	Abbreviated month name (Jan..Dec)
%c	Month, numeric (0..12)
%D	Day of the month with English suffix (0th, 1st, 2nd, 3rd, …)
%d	Day of the month, numeric (00..31)
%e	Day of the month, numeric (0..31)
%f	Microseconds (000000..999999)
%H	Hour (00..23)
%h	Hour (01..12)
%I	Hour (01..12)
%i	Minutes, numeric (00..59)
%j	Day of year (001..366)
%k	Hour (0..23)
%l	Hour (1..12)
%M	Month name (January..December)
%m	Month, numeric (00..12)
%p	AM or PM
%r	Time, 12-hour (hh:mm:ss followed by AM or PM)
%S	Seconds (00..59)
%s	Seconds (00..59)

Date and Time Functions

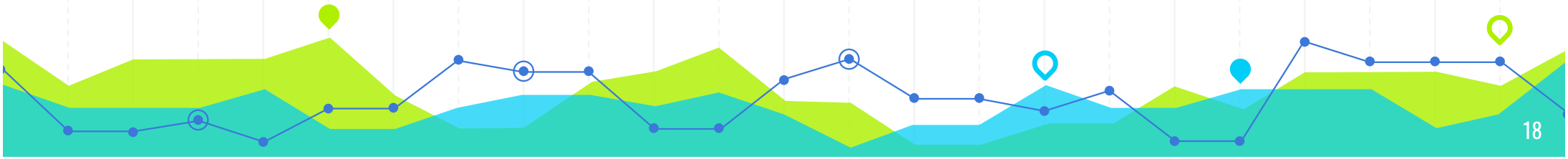
Specifier	Description
%T	Time, 24-hour (hh:mm:ss)
%U	Week (00..53), where Sunday is the first day of the week
%u	Week (00..53), where Monday is the first day of the week
%V	Week (01..53), where Sunday is the first day of the week; used with %X
%v	Week (01..53), where Monday is the first day of the week; used with %x
%W	Weekday name (Sunday..Saturday)
%w	Day of the week (0=Sunday..6=Saturday)
%X	Year for the week where Sunday is the first day of the week, numeric, four digits; used with %V
%x	Year for the week, where Monday is the first day of the week, numeric, four digits; used with %v
%Y	Year, numeric, four digits
%y	Year, numeric (two digits)
%%	A literal '%' character
%x	x, for any 'x' not listed above

Date and Time Functions

- **CURRENT DATE** and **CURRENT TIME**
- The **CURRENT_DATE** function returns today's date while the **CURRENT_TIME** function returns the current time.
- The **CURDATE**() function equal to the **CURRENT_DATE** function.
- The **CURRENT_TIMESTAMP**() function returns the current date and time.

Example

- **SELECT CURRENT_DATE**(), **CURDATE**(), **CURRENT_TIME**(), **CURRENT_TIMESTAMP**() ;



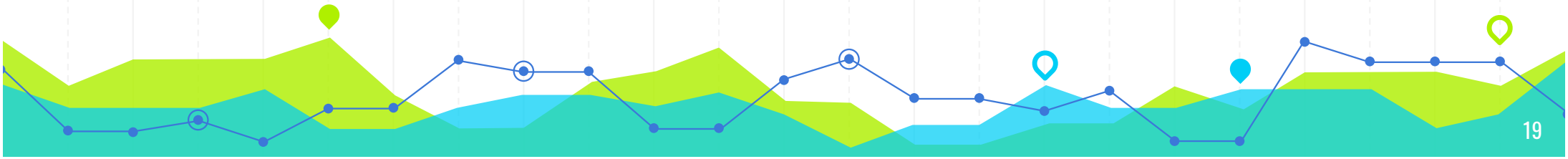
Date and Time Functions

MONTH, DAYOFMONTH and YEAR

- The syntax of each function is as follows:
  - **DAYOFMONTH(**date**)** returns the day of the month for date.
  - **MONTH(**date**)** returns the month for date.
  - **YEAR(**date**)** returns the year for date.

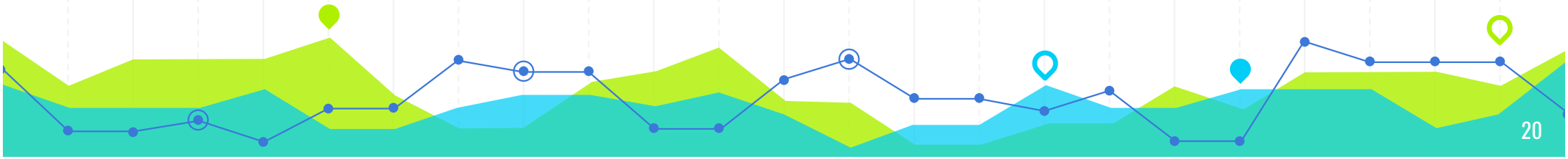
Example

- **SELECT DAYOFMONTH**(RENTAL_DATE) **AS** DAY, **MONTH**(RENTAL_DATE) **AS** MONTH, **YEAR**(RENTAL_DATE) **AS** YEAR **FROM** RENTAL;



Date and Time Functions

- The **DATEDIFF** function subtracts two dates and returns a value in days from one date to the other.
- **Ex.** Calculates the number of days between the 1st January 2022 and the 25th December 2018.
- **SELECT DATEDIFF**('2018-12-25','2022-01-01');



Date and Time Functions

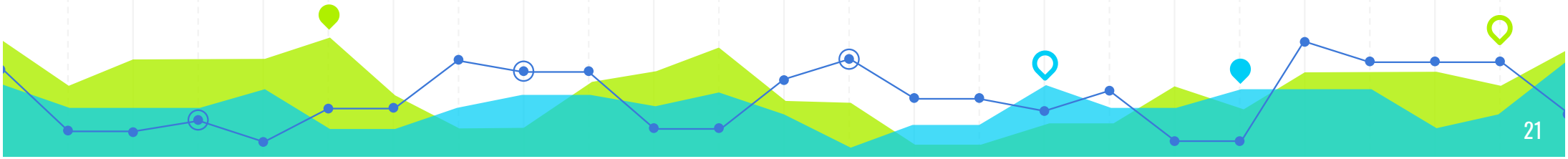
DATE_ADD and DATE_SUB

- The **DATE_ADD** and **DATE_SUB** functions both perform date arithmetic of two dates.

- Syntax

  DATE_ADD(date,**INTERVAL** value unit)
  DATE_SUB(date,**INTERVAL** value unit)

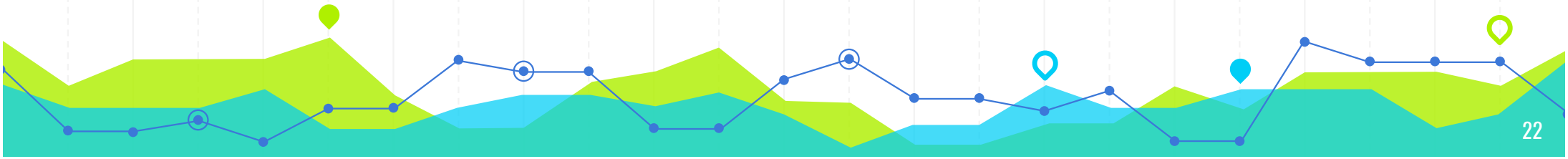
  Example
- **SELECT DATE_ADD**('2020-01-01', **INTERVAL** 11 MONTH);
- **SELECT DATE_SUB**('2020-10-30', **INTERVAL** 10 DAY);



Date and Time Functions

- **LAST_DAY** returns the date of the last day of the month given in a date.
- Syntax

  LAST_DAY(date_value).
- **Ex.** Query which lists all sales transactions that were made in the last 20 days of a month:
- **SELECT** * **FROM** RENTAL **WHERE** RENTAL_DATE >= **LAST_DAY**(RENTAL_DATE)-20;

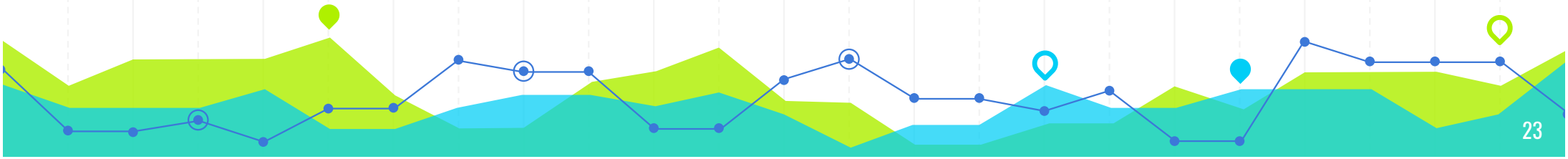


Conversion Functions

- Conversion functions allow you to take a value of a given data type and convert it to the equivalent value in another data type.

Example

- **SELECT** 10 + ‘10’ ;
- **SELECT CAST**(10 **AS** CHAR);



Conversion Functions

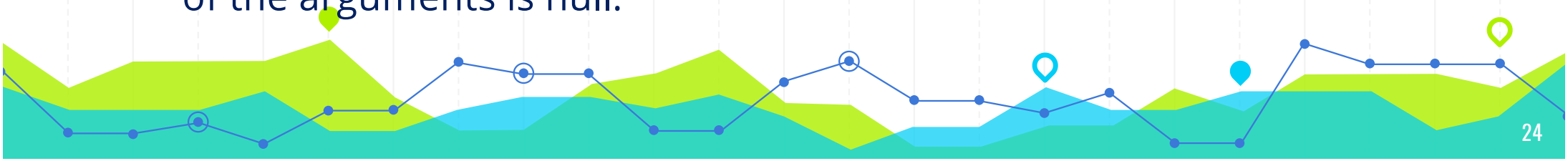
IFNULL

- The **IFNULL** function lets you substitute a value when a null value is encountered in the results of a query.

- Syntax

  IFNULL(expr1,expr2)

- If expr1 is not NULL, **IFNULL**() returns expr1; otherwise, it returns expr2.
- It is useful for avoiding errors caused by incorrect calculation when one of the arguments is null.



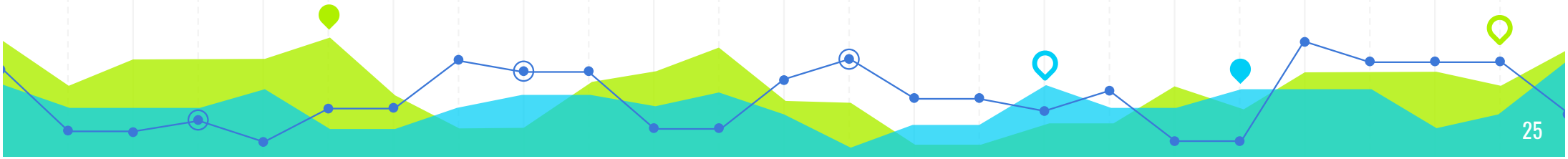
Conversion Functions

CASE

- The **CASE** function compares an attribute or expression with a series of values and returns an associated value or a default value if no match is found.

Syntax (first version)

```
CASE value
WHEN [compare_value] THEN result
[WHEN [compare_value] THEN result ...]
[ELSE result]
END;
```



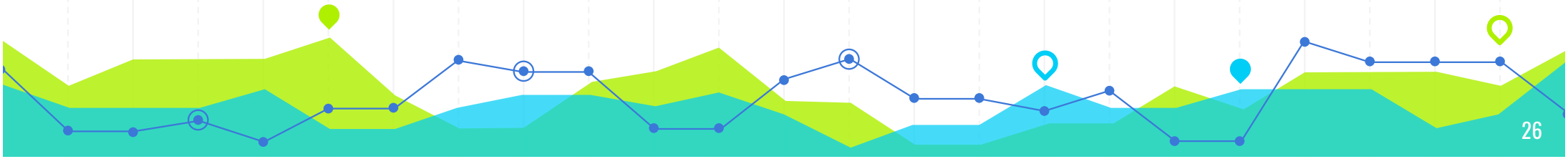
Conversion Functions

CASE

Syntax (second version)

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

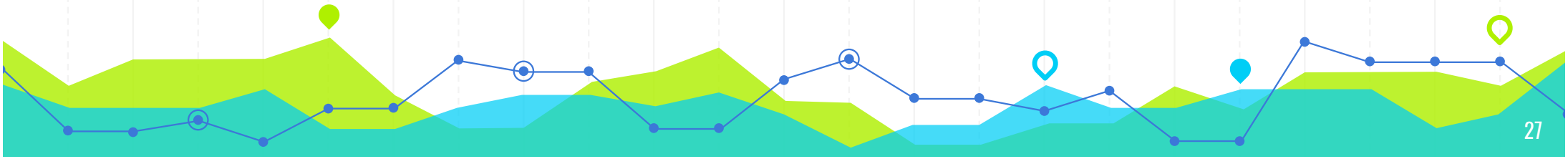
- The second version returns the result for the first condition that is true.
- If there was no matching result value, the result after ELSE is returned, or NULL if there is no ELSE part.



Conversion Functions

Example

```
SELECT *
FROM film
ORDER BY
(CASE
WHEN rating IS NULL THEN length
ELSE release_year
END);
```



Homework

Use sakila database

- Display the all the rental duration of the films and change them to 7.
- Write a query to show the first 3 letters of the name of the films and convert to upper case.
- Calculate your own age in a query.
- Write query to determine today's Julian date.
- Try to display today with five different formats.