Oracle Cloud Infrastructure

# New Free Tier

oracle.com/gbtour

## Always Free

**Services you can use for unlimited time**

+

## 30-Day Free Trial

**Free credits you can use for more services**

ORACLE

# How to Find Patterns in Your Data With SQL

**Chris Saxon,** @ChrisRSaxon **&** @SQLDaily

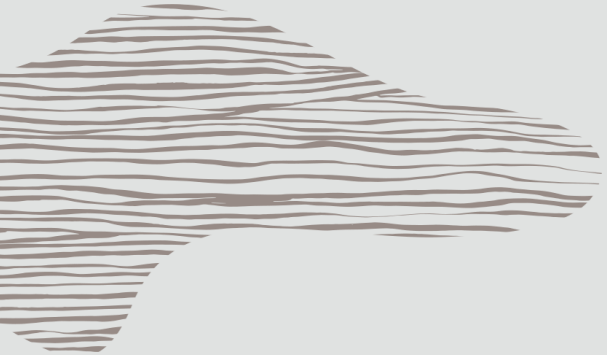
blogs.oracle.com/sql

youtube.com/c/TheMagicofSQL

asktom.oracle.com


Am I Training
Regularly?
Am I Improving?
Can Beat My PB?
2
km
Hamilton Park

ORACLE



# How to Find Patterns in Your Data With SQL

**Chris Saxon, @ChrisRSaxon & @SQLDaily**

blogs.oracle.com/sql

youtube.com/c/TheMagicofSQL

asktom.oracle.com

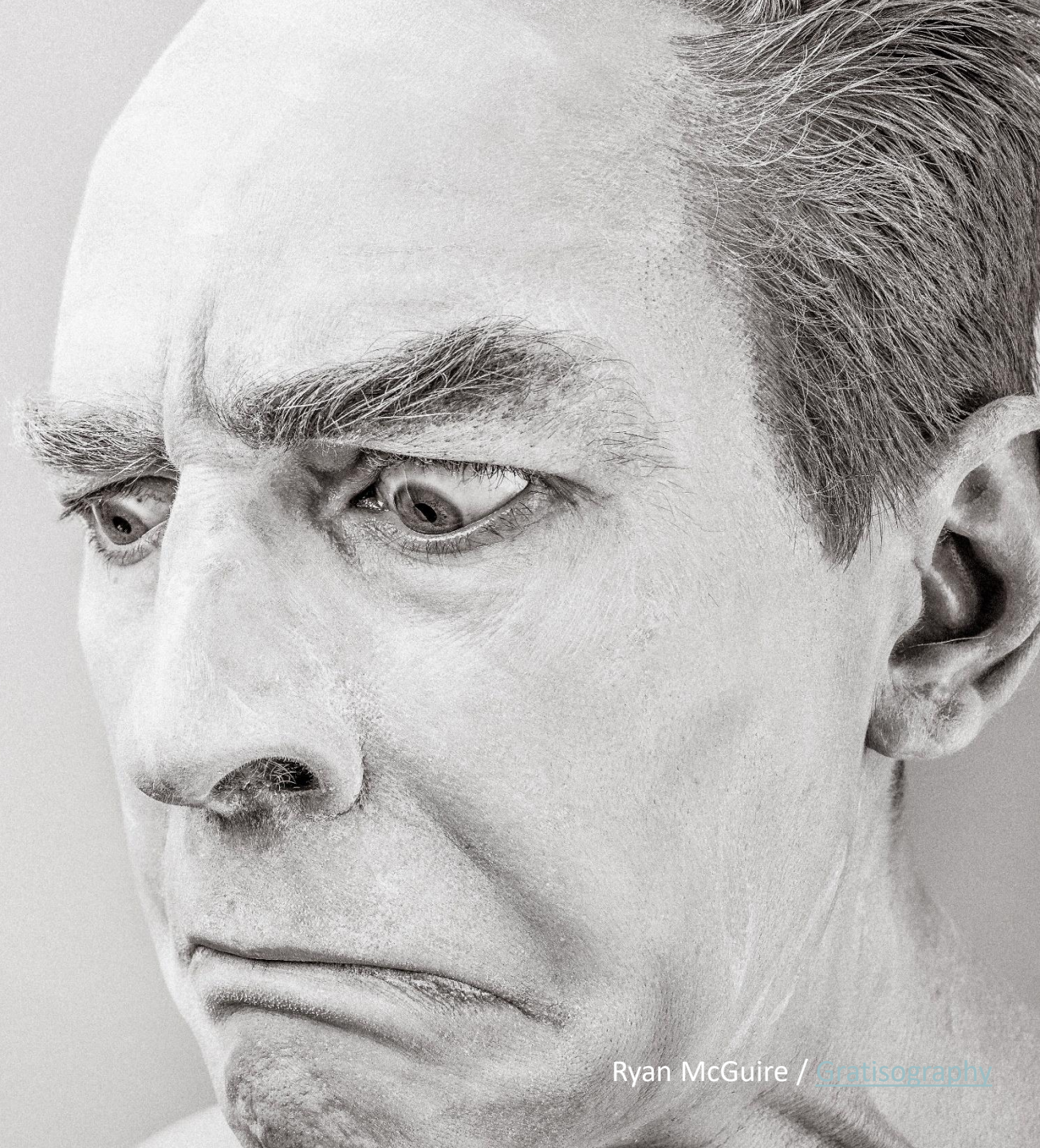
## Safe Harbor

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

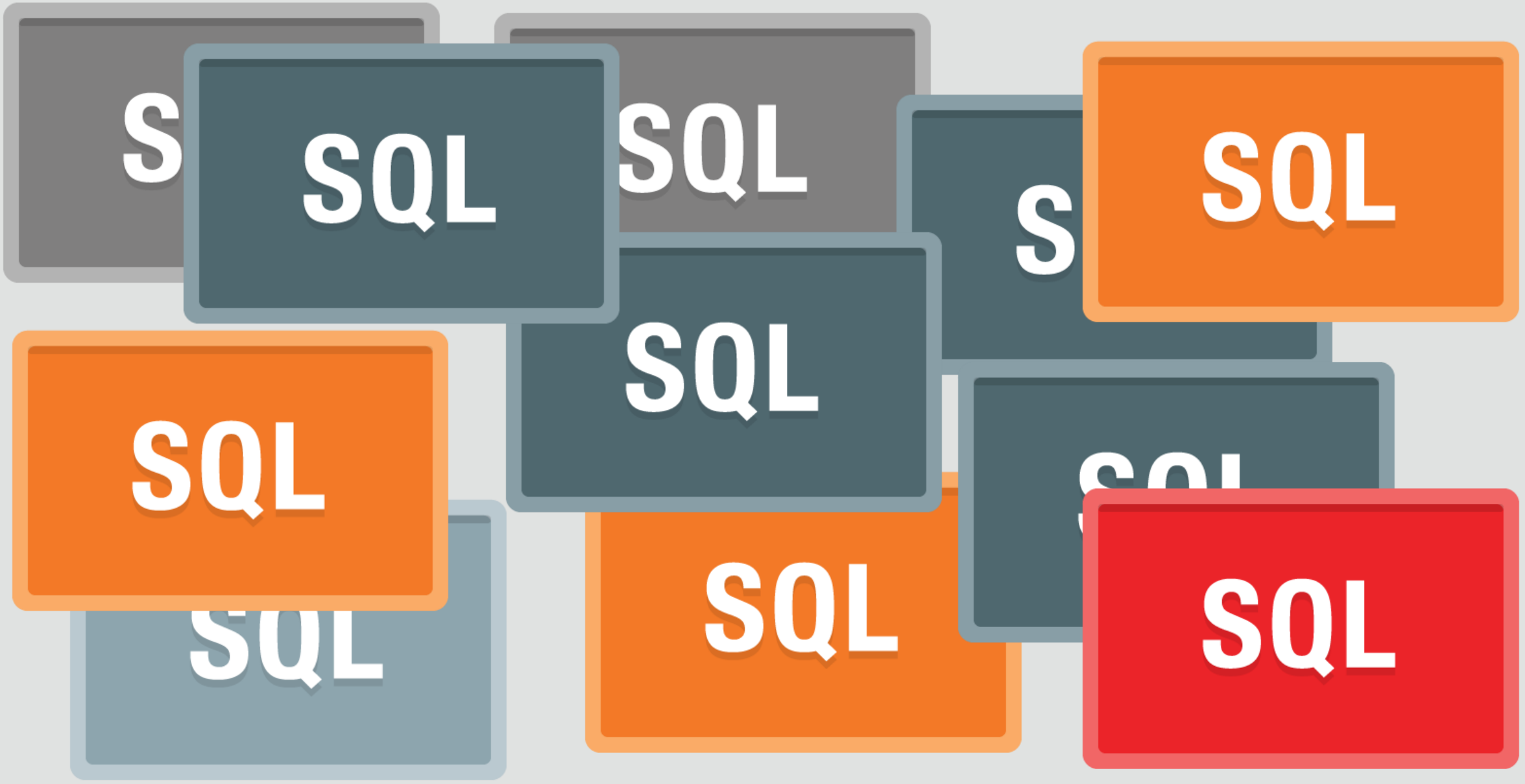
Statements in this presentation relating to Oracle's future plans, expectations, beliefs, intentions and prospects are "forward-looking statements" and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle's Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading "Risk Factors." These filings are available on the SEC's website or on Oracle's website at http://www.oracle.com/investor. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.

This presentation contains <regular expressions>!

# I thought this was about SQL!



Ryan McGuire / Gratisography


SQL
SQL
SQL
SQL
SQL
SQL
SQL
SQL

*  => zero or more matches

\+  => one or more matches

{n,m}  => N through M matches
(either optional)

# Regular Expressions: Say What?

Alex Nuijten

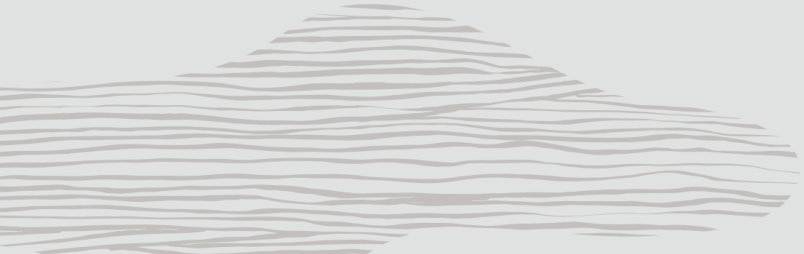
Next session!


Am I running every day?
Ryan McGuire / Gratisography

| RUN_DATE | TIME_IN_S | DISTANCE_IN_KM |
|---|---|---|
| 01 Jan 2018 | 310 | 1 |
| 02 Jan 2018 | 1,600 | 5 |
| 03 Jan 2018 | 3,580 | 11 |
| 06 Jan 2018 | 1,550 | 5 |
| 07 Jan 2018 | 300 | 1 |
| 10 Jan 2018 | 280 | 1 |
| 13 Jan 2018 | 1,530 | 5 |
| 14 Jan 2018 | 295 | 1 |
| 15 Jan 2018 | 292 | 1 |

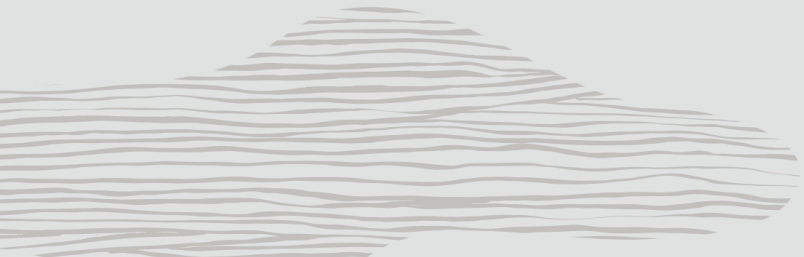
| RUN_DATE | | TIME_IN_S | DISTANCE_IN_KM |
|---|---|---|---|
| 01 Jan 2018 | #1 | 310 | 1 |
| 02 Jan 2018 | | 1,600 | 5 |
| 03 Jan 2018 | | 3,580 | 11 |
| 06 Jan 2018 | #2 | 1,550 | 5 |
| 07 Jan 2018 | | 300 | 1 |
| 10 Jan 2018 | #3 | 280 | 1 |
| 13 Jan 2018 | #4 | 1,530 | 5 |
| 14 Jan 2018 | | 295 | 1 |
| 15 Jan 2018 | | 292 | 1 |

# How I know if rows are consecutive?



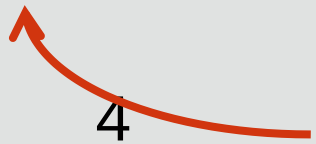
current value = previous value + 1

```
lag ( run_date ) over
  ( order by run_date )
```



Get the previous row's date

| RUN_DATE | RN | TIME_IN_S | DISTANCE_IN_KM |
|---|---|---|---|
| 01 Jan 2018 | 1 | 310 | 1 |
| 02 Jan 2018 | 2 | 1,600 | 5 |
| 03 Jan 2018 | 3 | 3,580 | 11 |
| 06 Jan 2018 | 4 | 1,550 | 5 |
| 07 Jan 2018 | 5 | 300 | 1 |
| 10 Jan 2018 | 6 | 280 | 1 |
| 13 Jan 2018 | 7 | 1,530 | 5 |
| 14 Jan 2018 | 8 | 295 | 1 |
| 15 Jan 2018 | 9 | 292 | 1 |



consecutive
=>
constant gap

| RUN_DATE | RN | TIME_IN_S | DISTANCE_IN_KM |
|---|---|---|---|
| 01 Jan 2018 | 1 | 310 | 1 |
| 02 Jan 2018 | 2 | 1,600 | 5 |
| 03 Jan 2018 | 3 | 3,580 | 11 |
| 06 Jan 2018 | 4 | 1,550 | 5 |
| 07 Jan 2018 | 5 | 300 | 1 |
| 10 Jan 2018 | 6 | 280 | 1 |
| 13 Jan 2018 | 7 | 1,530 | 5 |
| 14 Jan 2018 | 8 | 295 | 1 |
| 15 Jan 2018 | 9 | 292 | 1 |

| RUN_DATE | | RN | RUN_DATE - RN | TIME_IN_S | DISTANCE_IN_KM |
|---|---|---|---|---|---|
| 01 Jan 2018 | - | 1 | 31 Dec 2017 | 310 | 1 |
| 02 Jan 2018 | - | 2 | 31 Dec 2017 | 1,600 | 5 |
| 03 Jan 2018 | - | 3 | 31 Dec 2017 | 3,580 | 11 |
| 06 Jan 2018 | - | 4 | 02 Jan 2018 | 1,550 | 5 |
| 07 Jan 2018 | - | 5 | 02 Jan 2018 | 300 | 1 |
| 10 Jan 2018 | - | 6 | 04 Jan 2018 | 280 | 1 |
| 13 Jan 2018 | - | 7 | 06 Jan 2018 | 1,530 | 5 |
| 14 Jan 2018 | - | 8 | 06 Jan 2018 | 295 | 1 |
| 15 Jan 2018 | - | 9 | 06 Jan 2018 | 292 | 1 |

| RUN_DATE | | RN | RUN_DATE - RN | TIME_IN_S | DISTANCE_IN_KM |
|---|---|---|---|---|---|
| 01 Jan 2018 | - | 1 | 31 Dec 2017 | 310 | 1 |
| 02 Jan 2018 | - | 2 | 31 Dec 2017 | 1,600 | 5 |
| 03 Jan 2018 | - | 3 | 31 Dec 2017 | 3,580 | 11 |
| 06 Jan 2018 | - | 4 | 02 Jan 2018 | 1,550 | 5 |
| 07 Jan 2018 | - | 5 | 02 Jan 2018 | 300 | 1 |
| 10 Jan 2018 | - | 6 | 04 Jan 2018 | 280 | 1 |
| 13 Jan 2018 | - | 7 | 06 Jan 2018 | 1,530 | 5 |
| 14 Jan 2018 | - | 8 | 06 Jan 2018 | 295 | 1 |
| 15 Jan 2018 | - | 9 | 06 Jan 2018 | 292 | 1 |

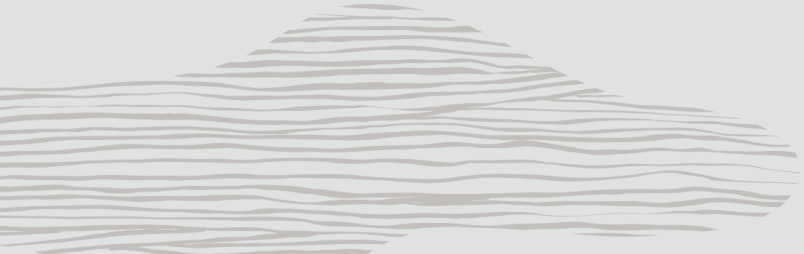
# Tabibitosan Method

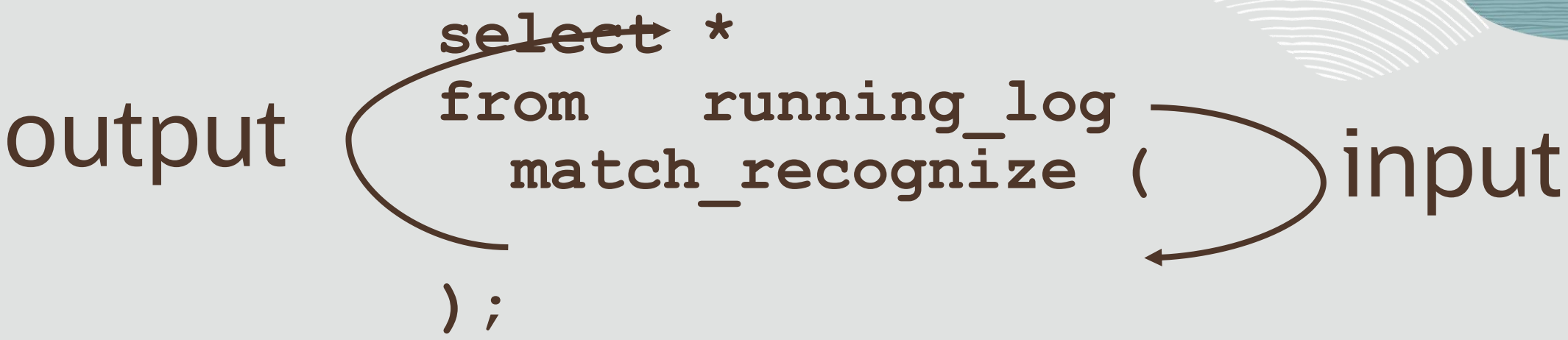
```
row_number ()
   over ( order by run_date )
```

```
run_date -
row_number ()
   over ( order by run_date ) grp
```

```
with grps as (
   select run_date ,
          run_date -
          row_number ()
            over ( order by run_date ) grp
   from   running_log r
)
   select min ( run_date ), count (*)
   from   grps
   group  by grp
```

# 12c Pattern Matching




output
select *
from running_log
match_recognize (
);
input

| RUN_DATE | TIME_IN_S | DISTANCE_IN_KM |
|---|---|---|
| 01 Jan 2018 | 310 | 1 |
| 02 Jan 2018 | 1,600 | 5 |
| 03 Jan 2018 | 3,580 | 11 |
| 06 Jan 2018 | 1,550 | 5 |
| 07 Jan 2018 | 300 | 1 |
| 10 Jan 2018 | 280 | 1 |
| 13 Jan 2018 | 1,530 | 5 |
| 14 Jan 2018 | 295 | 1 |
| 15 Jan 2018 | 292 | 1 |

| RUN_DATE | TIME_IN_S | DISTANCE_IN_KM |
|---|---|---|
| 01 Jan 2018 | 310 | 1 |
| 02 Jan 2018 | 1,600 | 5 |
| 03 Jan 2018 | 3,580 | 11 |
| 06 Jan 2018 | 1,550 | 5 |
| 07 Jan 2018 | 300 | 1 |
| 10 Jan 2018 | 280 | 1 |
| 13 Jan 2018 | 1,530 | 5 |
| 14 Jan 2018 | 295 | 1 |
| 15 Jan 2018 | 292 | 1 |

this = prev + 1

this = prev + 3

| RUN_DATE | TIME_IN_S | DISTANCE_IN_KM |
|---|---|---|
| 01 Jan 2018 | 310 | 1 |
| 02 Jan 2018 | 1,600 | 5 |
| 03 Jan 2018 | 3,580 | 11 |
| 06 Jan 2018 | 1,550 | 5 |
| 07 Jan 2018 | 300 | 1 |
| 10 Jan 2018 | 280 | 1 |
| 13 Jan 2018 | 1,530 | 5 |
| 14 Jan 2018 | 295 | 1 |
| 15 Jan 2018 | 292 | 1 |

this = prev + 1

this = prev + 3

this ≠ prev + 1

current value = previous value + 1

```
define
   consecutive as
      run_date = prev ( run_date ) + 1
```

```
pattern ( init consecutive* )
define
   consecutive as
      run_date = prev ( run_date ) + 1
```

Undefined =>
"Always true"

> 0 matches

```
pattern ( init consecutive* )
define
   consecutive as
      run_date = prev ( run_date ) + 1
```

| RUN_DATE | VARIABLE | TIME_IN_S | DISTANCE_IN_KM |
|---|---|---|---|
| 01 Jan 2018 | INIT | 310 | 1 |
| 02 Jan 2018 | CONSECUTIVE | 1,600 | 5 |
| 03 Jan 2018 | CONSECUTIVE | 3,580 | 11 |
| | | | |
| 06 Jan 2018 | INIT | 1,550 | 5 |
| 07 Jan 2018 | CONSECUTIVE | 300 | 1 |
| | | | |
| 10 Jan 2018 | INIT | 280 | 1 |
| | | | |
| 13 Jan 2018 | INIT | 1,530 | 5 |
| 14 Jan 2018 | CONSECUTIVE | 295 | 1 |
| 15 Jan 2018 | CONSECUTIVE | 292 | 1 |

```
pattern ( init consecutive* )
define
   consecutive as
      run_date = prev ( run_date ) + 1
```

```
order by run_date


pattern ( init consecutive* )
define
   consecutive as
      run_date = prev ( run_date ) + 1
```

```
match_recognize (
   order by run_date
   measures
      first ( run_date ) as start_date,
      count (*) as days
   pattern ( init consecutive* )
   define
      consecutive as
         run_date = prev ( run_date ) + 1
);
```

First row in group

How many consecutive rows?

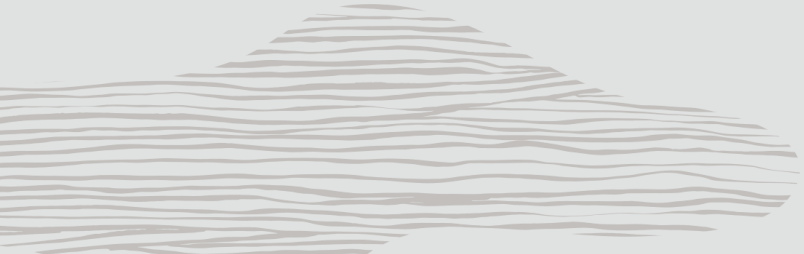
| START_DATE | DAYS |
|---|---|
| 01 Jan 2018 | 3 |
| 06 Jan 2018 | 2 |
| 10 Jan 2018 | 1 |
| 13 Jan 2018 | 3 |


8i*
tabibitosan
12c
pattern
matching
~speed
So which is
better?
Pixabay

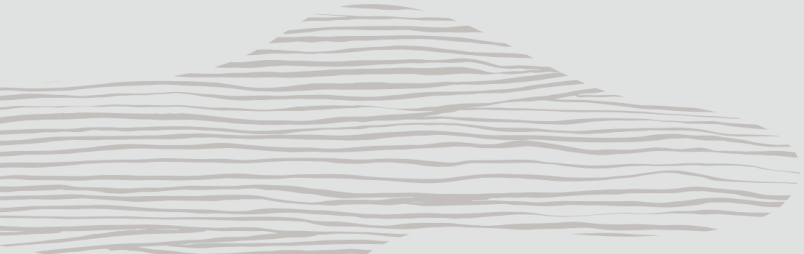

Am I running >= 3 times/week?
Pixabay

| RUN_DATE | | TIME_IN_S | DISTANCE_IN_KM |
|---|---|---|---|
| 01 Jan 2018 | #1 | 310 | 1 |
| 02 Jan 2018 | | 1,600 | 5 |
| 03 Jan 2018 | | 3,580 | 11 |
| 06 Jan 2018 | | 1,550 | 5 |
| 07 Jan 2018 | | 300 | 1 |
| 10 Jan 2018 | #2 | 280 | 1 |
| 13 Jan 2018 | | 1,530 | 5 |
| 14 Jan 2018 | | 295 | 1 |
| 15 Jan 2018 | #3 | 292 | 1 |

# How I know if runs are in the same week?



# latest Monday = prev latest Monday



```
trunc ( run_date , 'iw' )
```

Return the start of the ISO week…

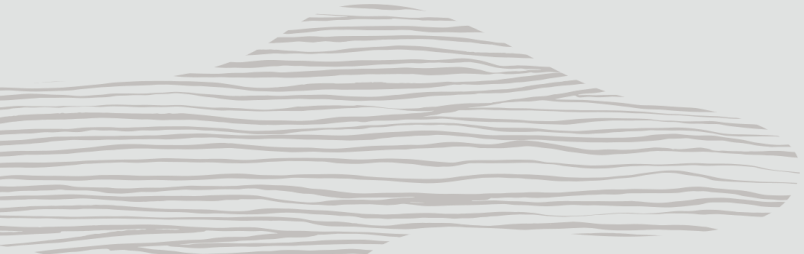
…Monday!

| RUN_DATE | TRUNC(RUN_DATE, 'IW') | TIME_IN_S | DISTANCE_IN_KM |
|---|---|---|---|
| 01 Jan 2018 | 01 Jan 2018 | 310 | 1 |
| 02 Jan 2018 | 01 Jan 2018 | 1,600 | 5 |
| 03 Jan 2018 | 01 Jan 2018 | 3,580 | 11 |
| 06 Jan 2018 | 01 Jan 2018 | 1,550 | 5 |
| 07 Jan 2018 | 01 Jan 2018 | 300 | 1 |
| 10 Jan 2018 | 08 Jan 2018 | 280 | 1 |
| 13 Jan 2018 | 08 Jan 2018 | 1,530 | 5 |
| 14 Jan 2018 | 08 Jan 2018 | 295 | 1 |
| 15 Jan 2018 | 15 Jan 2018 | 292 | 1 |

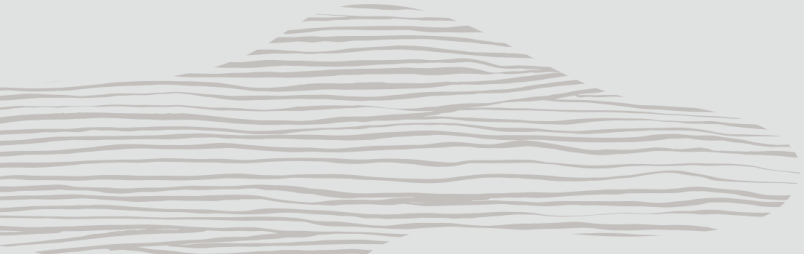
```
select trunc ( run_date , 'iw' ),
       count(*)
from   running_log
group  by trunc ( run_date , 'iw' )
```

```
select trunc ( run_date , 'iw' ),
       count(*)
from   running_log
group  by trunc ( run_date , 'iw' )
having count (*) >= 3
```

# 12c Pattern Matching



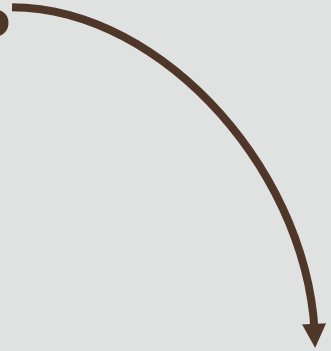
# latest Monday = prev latest Monday



```
define
   same_week as
      trunc ( run_date, 'iw' ) =
         prev ( trunc ( run_date, 'iw' ) )
```

```
pattern ( init same_week* )
define
   same_week as
     trunc ( run_date, 'iw' ) =
       prev ( trunc ( run_date, 'iw' ) )
```

Two or more matches



```
pattern ( init same_week {2, } )
define
   same_week as
     trunc ( run_date, 'iw' ) =
       prev ( trunc ( run_date, 'iw' ) )
```

```
match_recognize (
   order by run_date
   measures
      first ( run_date ) as start_date,
      count (*) as days
   pattern ( init same_week {2, } )
   define
      same_week as
         trunc ( run_date, 'iw' ) =
            prev ( trunc ( run_date, 'iw' ) )
);
```

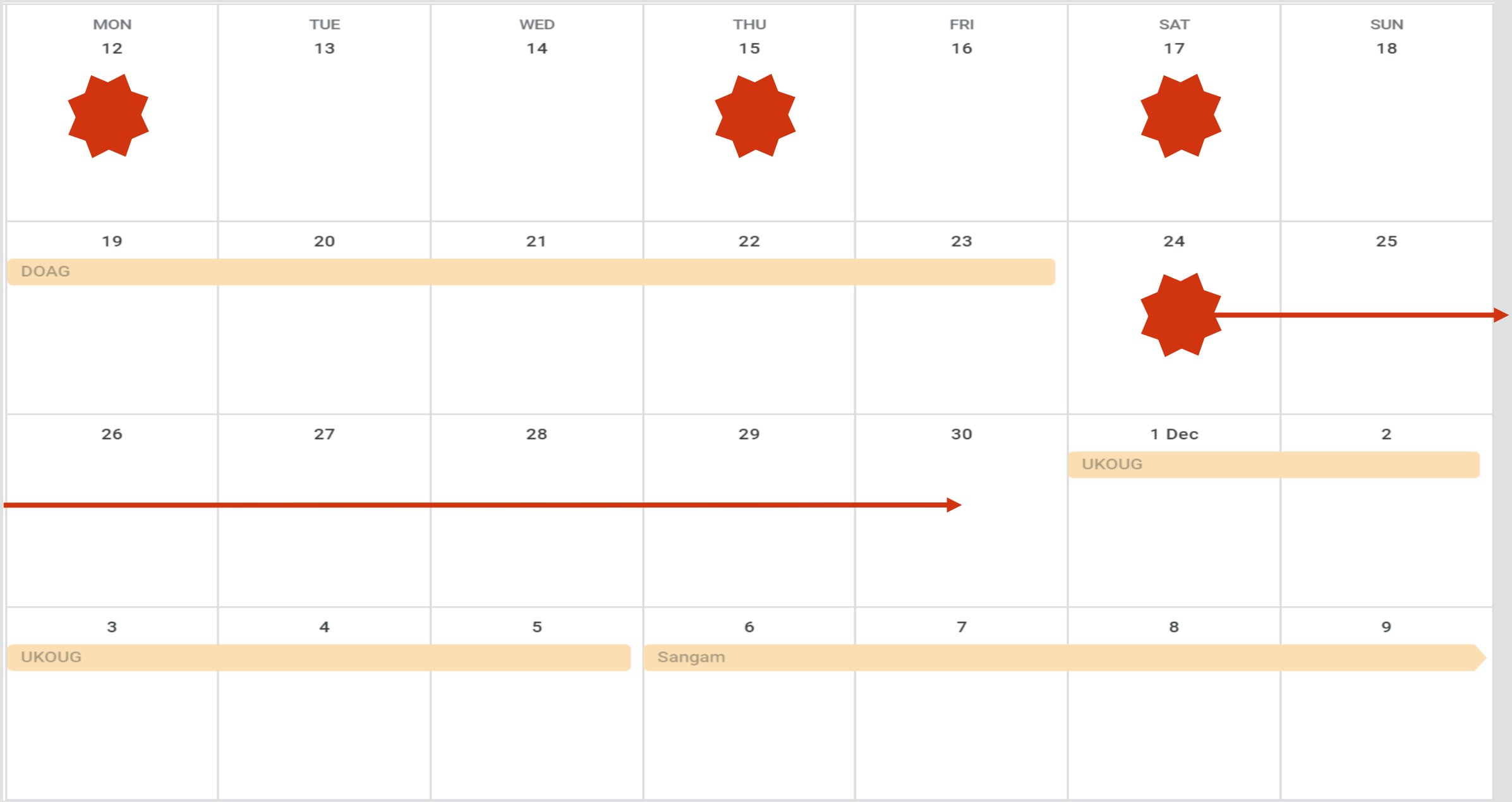
| START_DATE | DAYS |
| --- | --- |
| 01 Jan 2018 | 5 |
| 08 Jan 2018 | 3 |

```
match_recognize (
   order by run_date
   measures
      first ( run_date ) as start_date,
      count (*) as days
   pattern ( init same_week {2, } )
   define
      same_week as
         trunc ( run_date, 'iw' ) =
            prev ( trunc ( run_date, 'iw' ) )
);
```

```
match_recognize (
   order by run_date
   measures
      first ( run_date ) as start_date,
      count (*) as days
   pattern ( init consecutive* )
   define
      consecutive as
         run_date = prev ( run_date ) + 1
);
```

| MON | TUE | WED | THU | FRI | SAT | SUN |
|---|---|---|---|---|---|---|
| 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| DOAG | | | | | | |
| 26 | 27 | 28 | 29 | 30 | 1 Dec | 2 |
| | | | | | UKOUG | |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| UKOUG | | | Sangam | | | |

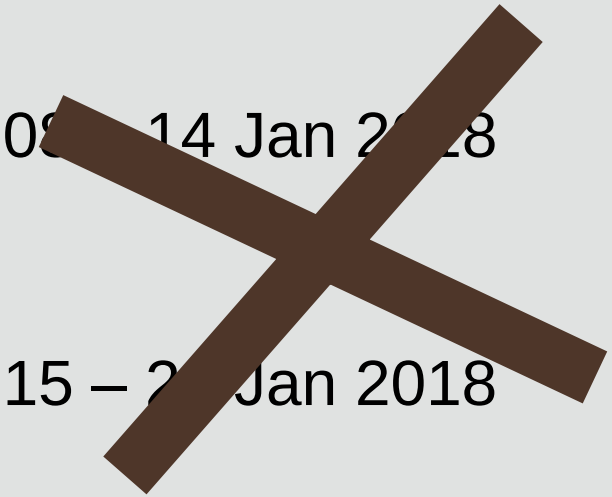

Am I running >=3
times in 7 days?
Pixabay


MON
TUE
WED
THU
FRI
SAT
SUN
12
13
14
15
16
17
18
19
20
21
22
23
24
25
DOAG
26
27
28
29
30
1 Dec
2
UKOUG
3
4
5
6
7
8
9
UKOUG
Sangam

| RUN_DATE | TIME_IN_S | DISTANCE_IN_KM |
|---|---|---|
| 01 Jan 2018 | 310 | 1 |
| 02 Jan 2018 | 1,600 | 5 |
| 03 Jan 2018 | 3,580 | 11 |
| 06 Jan 2018 | 1,550 | 5 |
| 07 Jan 2018 | 300 | 1 |
| 10 Jan 2018 | 280 | 1 |
| 13 Jan 2018 | 1,530 | 5 |
| 14 Jan 2018 | 295 | 1 |
| 15 Jan 2018 | 292 | 1 |

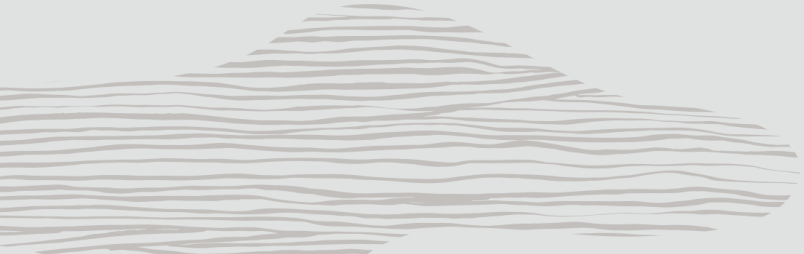
| RUN_DATE | | TIME_IN_S | DISTANCE_IN_KM |
|---|---|---|---|
| 01 Jan 2018 | #1 | 310 | 1 |
| 02 Jan 2018 | | 1,600 | 5 |
| 03 Jan 2018 | | 3,580 | 11 |
| 06 Jan 2018 | | 1,550 | 5 |
| 07 Jan 2018 | | 300 | 1 |
| 10 Jan 2018 | #2 | 280 | 1 |
| 13 Jan 2018 | | 1,530 | 5 |
| 14 Jan 2018 | | 295 | 1 |
| 15 Jan 2018 | | 292 | 1 |

| RUN_DATE | | TIME_IN_S | DISTANCE_IN_KM |
|---|---|---|---|
| 01 Jan 2018 | 01 – 07 Jan 2018 | 310 | 1 |
| 02 Jan 2018 | | 1,600 | 5 |
| 03 Jan 2018 | | 3,580 | 11 |
| 06 Jan 2018 | | 1,550 | 5 |
| 07 Jan 2018 | | 300 | 1 |
| 10 Jan 2018 | 08 – 14 Jan 2018 | 280 | 1 |
| 13 Jan 2018 | | 1,530 | 5 |
| 14 Jan 2018 | | 295 | 1 |
| 15 Jan 2018 | 15 – 21 Jan 2018 | 292 | 1 |

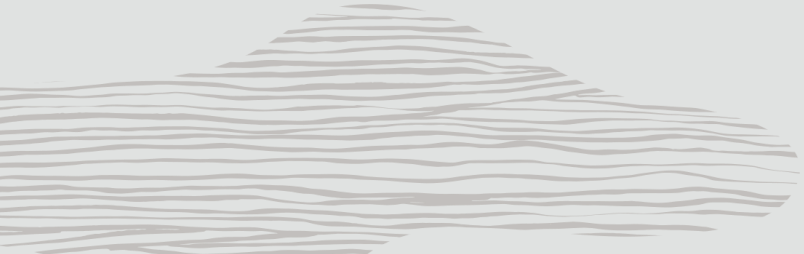


| RUN_DATE | | TIME_IN_S | DISTANCE_IN_KM |
|---|---|---|---|
| 01 Jan 2018 | 01 – 07 Jan 2018 | 310 | 1 |
| 02 Jan 2018 | | 1,600 | 5 |
| 03 Jan 2018 | | 3,580 | 11 |
| 06 Jan 2018 | | 1,550 | 5 |
| 07 Jan 2018 | | 300 | 1 |
| | | | |
| 10 Jan 2018 | 10 – 16 Jan 2018 | 280 | 1 |
| 13 Jan 2018 | | 1,530 | 5 |
| 14 Jan 2018 | | 295 | 1 |
| 15 Jan 2018 | | 292 | 1 |

current day < first day + 7

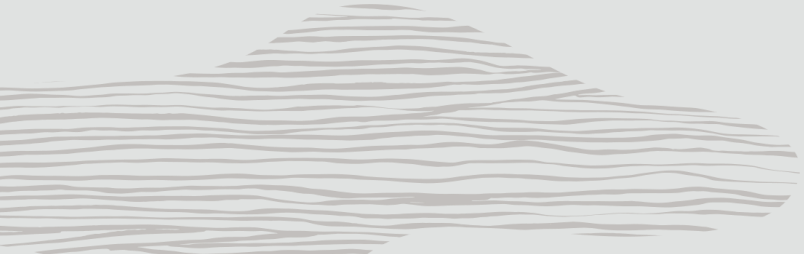


# 11.2 Recursive With



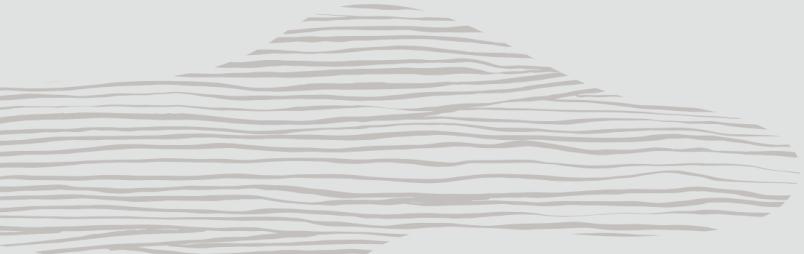
```
with rws as (
  select r.*, row_number() over ( order by run_date ) rn
  from   running_log r
), within_7 (
  run_date, time_in_s, distance_in_km, rn, grp_start
) as (
  select run_date, time_in_s, distance_in_km,
         rn, run_date grp_start
  from   rws where rn = 1
  union all
  select r.run_date, r.time_in_s, r.distance_in_km, r.rn,
         case
           when r.run_date < w.grp_start + 7 then grp_start
           else r.run_date
         end grp_start
  from   within_7 w join rws r on w.rn + 1 = r.rn
)
  select grp, w.* from within_7 w
```

# 10g Model

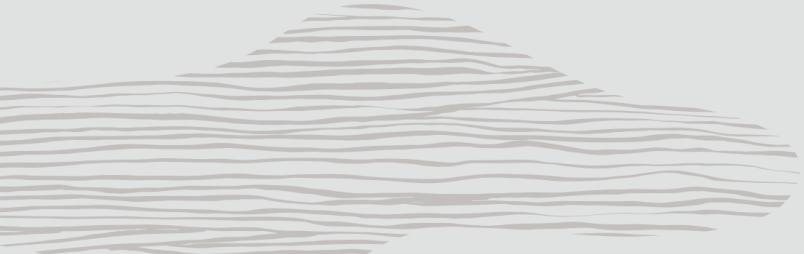


```
select * from running_log
model
  dimension by ( row_number() over ( order by run_date ) rn )
  measures ( run_date, 1 grp, run_date grp_start )
  rules (
    grp_start[1] = run_date[cv()],
    grp_start[any] =
      case
        when run_date[cv()] < grp_start[cv()-1] + 7 then
          grp_start[cv() - 1]
        else run_date[cv()]
      end ,
    grp[any] =
      case
        when run_date[cv()] < grp_start[cv()-1] + 7 then
          grp[cv() - 1]
        else nvl(grp[cv() - 1] + 1, 1)
      end
  );
```

# 12c Pattern Matching



current day < first day + 7



```
define
   within7 as
      run_date < first ( run_date ) + 7
```

```
pattern ( within7 {3, } )
define
   within7 as
      run_date < first ( run_date ) + 7
```

```
match_recognize (
   order by run_date
   measures
      first ( run_date ) as start_date,
      count (*) as days
   pattern ( within7 {3, } )
   define
      within7 as
         run_date < first ( run_date ) + 7
);
```

| START_DATE | DAYS |
|---|---|
| 01 Jan 2018 | 5 |
| 10 Jan 2018 | 4 |


Am I getting
faster?
stocksnap.io

current time < prev time

```
define
   faster as
      time_in_s < prev ( time_in_s )
```

```
pattern ( slower faster* )
define
   faster as
      time_in_s < prev ( time_in_s )
```

```
match_recognize (
   order by run_date
   measures
      classifier () as faster

   pattern ( slower faster* )
   define
      faster as
         time_in_s < prev ( time_in_s )
);
```

FASTER

---

SLOWER

SLOWER

FASTER

FASTER

```
match_recognize (
   order by run_date
   measures
      classifier () as faster
   one row per match
   pattern ( slower faster* )
   define
      faster as
         time_in_s < prev ( time_in_s )
);
```

```
match_recognize (
   order by run_date
   measures
      classifier () as faster
   all rows per match
   pattern ( slower faster* )
   define
      faster as
         time_in_s < prev ( time_in_s )
);
```

| RUN_DATE | FASTER | TIME_IN_S | DISTANCE_IN_KM |
|---|---|---|---|
| 01 Jan 2018 | SLOWER | 310 | 1 |
| 02 Jan 2018 | SLOWER | 1,600 | 5 |
| 03 Jan 2018 | SLOWER | 3,580 | 11 |
| 06 Jan 2018 | FASTER | 1,550 | 5 |
| 07 Jan 2018 | FASTER | 300 | 1 |
| 10 Jan 2018 | FASTER | 280 | 1 |
| 13 Jan 2018 | SLOWER | 1,530 | 5 |
| 14 Jan 2018 | FASTER | 295 | 1 |
| 15 Jan 2018 | FASTER | 292 | 1 |

| RUN_DATE | FASTER | TIME_IN_S | DISTANCE_IN_KM |
|---|---|---|---|
| 01 Jan 2018 | SLOWER | 310 | 1 |
| 02 Jan 2018 | SLOWER | 1,600 | 5 |
| 03 Jan 2018 | SLOWER | 3,580 | 11 |
| 06 Jan 2018 | FASTER | 1,550 | 5 |
| 07 Jan 2018 | FASTER | 300 | 1 |
| 10 Jan 2018 | FASTER | **280** | 1 |
| 13 Jan 2018 | SLOWER | 1,530 | 5 |
| 14 Jan 2018 | ~~FASTER~~ SLOWER! | **295** | 1 |
| 15 Jan 2018 | FASTER | 292 | 1 |

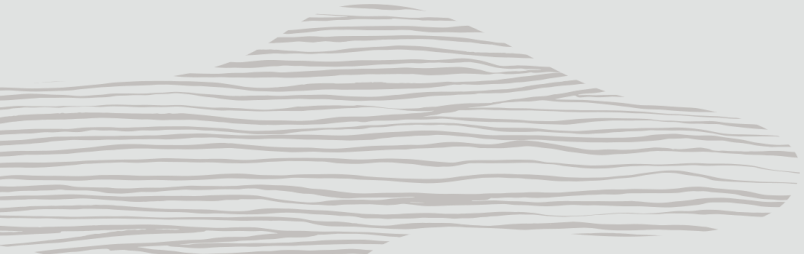
| RUN_DATE | TIME_IN_S | DISTANCE_IN_KM |
|---|---|---|
| 01 Jan 2018 | 310 | 1 |
| 07 Jan 2018 | 300 | 1 |
| 10 Jan 2018 | 280 | 1 |
| 14 Jan 2018 | 295 | 1 |
| 15 Jan 2018 | 292 | 1 |
| 02 Jan 2018 | 1,600 | 5 |
| 06 Jan 2018 | 1,550 | 5 |
| 13 Jan 2018 | 1,530 | 5 |
| 03 Jan 2018 | 3,580 | 11 |

```
match_recognize (
   partition by distance_in_km
   order by run_date
   measures
      classifier () as faster
   all rows per match
   pattern ( slower faster* )
   define
      faster as
         time_in_s < prev ( time_in_s )
);
```

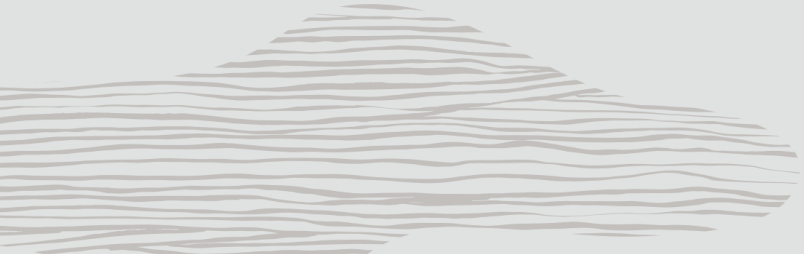
| RUN_DATE | FASTER | TIME_IN_S | DISTANCE_IN_KM |
|---|---|---|---|
| 01 Jan 2018 | SLOWER | 310 | 1 |
| 07 Jan 2018 | FASTER | 300 | 1 |
| 10 Jan 2018 | FASTER | **280** | 1 |
| 14 Jan 2018 | SLOWER | **295** | 1 |
| 15 Jan 2018 | FASTER | 292 | 1 |
| 02 Jan 2018 | SLOWER | 1,600 | 5 |
| 06 Jan 2018 | FASTER | 1,550 | 5 |
| 13 Jan 2018 | FASTER | 1,530 | 5 |
| 03 Jan 2018 | SLOWER | 3,580 | 11 |


# Can I run 10k in < 50 minutes?

Is my average pace < 300 s/km for runs with a total distance <= 10 km



cumulative dist <= 10 km



```
define
   ten_k as
      sum ( distince_in_km ) <= 10
```

Returns the running total

```
pattern ( ten_k+ )
define
   ten_k as
      sum ( distince_in_km ) <= 10
```

```
match_recognize (
   order by run_date
   measures
      first ( run_date ) as strt ,
      round ( avg ( time_in_s /
        distance_in_km ), 2 ) as mean_pace,
      sum ( distance_in_km ) as dist

   pattern ( ten_k+ )
   define
      ten_k as
        sum ( distince_in_km ) <= 10
);
```

| STRT | MEAN_PACE | DIST |
|---|---|---|
| 01 Jan 2018 | 315.00 | 6 |
| 06 Jan 2018 | 296.67 | 7 |
| 13 Jan 2018 | 297.67 | 7 |

# Where's my 11 km run?

***any*** runs cumulative dist < 10
and
***one*** run cumulative dist >= 10

```
pattern (                                  )
```

```
pattern ( under_10k* over_10k )
```

```
   pattern ( under_10k* over_10k )
   define
      under_10k as
         sum ( distance_in_km ) < 10,
      over_10k as
         sum ( distance_in_km ) >= 10
);
```

Includes under_10k values

```
match_recognize (
   order by run_date
   measures
      first ( run_date ) as strt ,
      round ( avg ( time_in_s /
        distance_in_km ), 2 ) as mean_pace
      sum ( distance_in_km )  as dist

   pattern ( under_10k* over_10k )
   define
      under_10k as
        sum ( distance_in_km ) < 10,
      over_10k as
        sum ( distance_in_km ) >= 10
);
```

| STRT | MEAN_PACE | DIST |
|---|---|---|
| **01 Jan 2018** | **318.48** | **17** |
| 06 Jan 2018 | 299.00 | 12 |

Hmmm....

```
match_recognize (
   order by run_date
   measures
      first ( run_date ) as strt ,
      round ( avg ( time_in_s /
        distance_in_km ), 2 ) as mean_pace
      sum ( distance_in_km )  as dist
   after match skip past last row
   pattern ( under_10k* over_10k )
   define
      under_10k as
        sum ( distance_in_km ) < 10,
      over_10k as
        sum ( distance_in_km ) >= 10
);
```

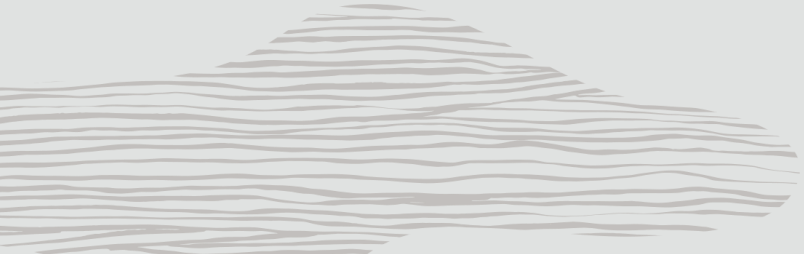
```
match_recognize (
   order by run_date
   measures
      first ( run_date ) as strt ,
      round ( avg ( time_in_s /
        distance_in_km ), 2 ) as mean_pace
      sum ( distance_in_km )  as dist
   after match skip to next row
   pattern ( under_10k* over_10k )
   define
      under_10k as
        sum ( distance_in_km ) < 10,
      over_10k as
        sum ( distance_in_km ) >= 10
);
```

| STRT | MEAN_PACE | DIST |
|---|---|---|
| 01 Jan 2018 | 318.48 | 17 |
| 02 Jan 2018 | 322.73 | 16 |
| **03 Jan 2018** | **325.45** | **11** |
| 06 Jan 2018 | 299.00 | 12 |

00:48:19

Photo by Doruk Yemenici on Unsplash

# What About Query Performance?

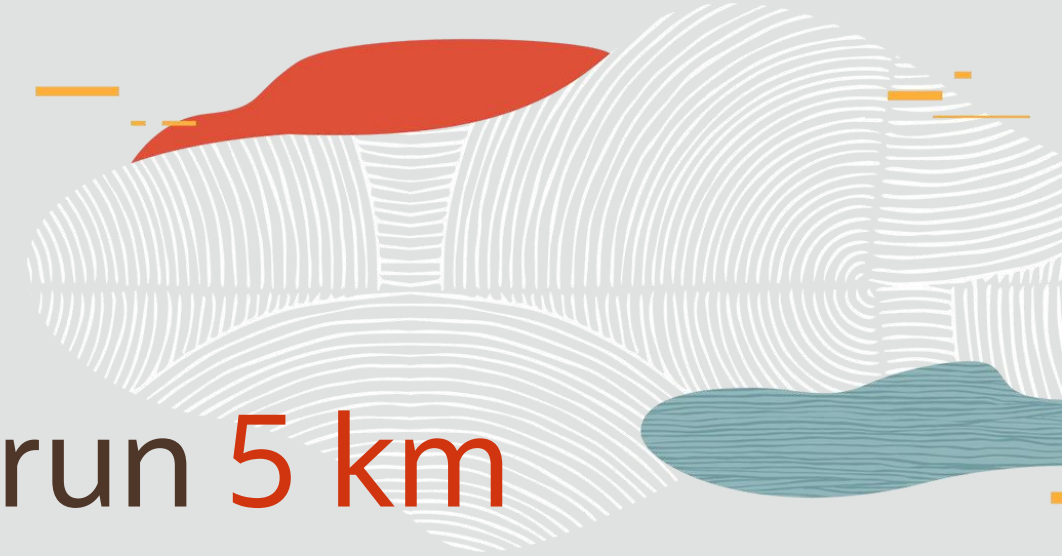


Non-deterministic

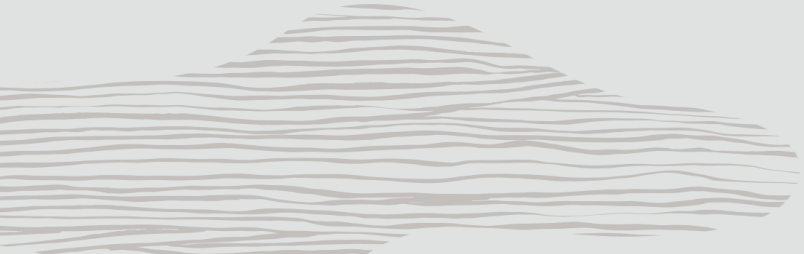
**MATCH RECOGNIZE SORT**

MATCH RECOGNIZE SORT
DETERMINISTIC FINITE AUTO


Pixabay



How often did I run 5 km
Followed by 2+ 1 km runs
Within 7 days?



```
pattern ( five_km one_km {2,} )
```

```
pattern ( five_km one_km {2,} )
define
   five_km as distance_in_km = 5,
```

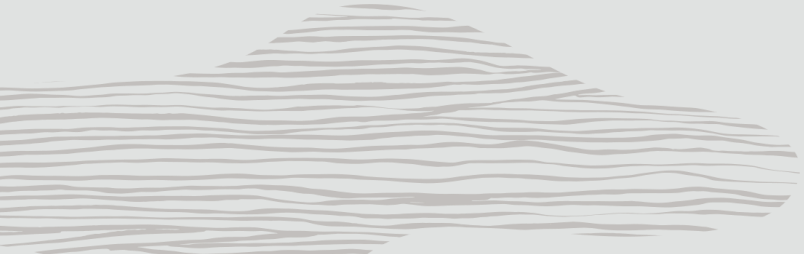
```
pattern ( five_km one_km {2,} )
define
   five_km as distance_in_km = 5,
   one_km as distance_in_km = 1
```

```
pattern ( five_km one_km {2,} )
define
   five_km as distance_in_km = 5,
   one_km as distance_in_km = 1
     and run_date < first ( run_date ) + 7
```

```
match_recognize (
   order by run_date
   measures
      first ( run_date ) as start_date,
      count (*) as total_runs
   pattern ( five_km one_km {2,} )
   define
      five_km as distance_in_km = 5,
      one_km as distance_in_km = 1
         and run_date < first ( run_date ) + 7
);
```

| START_DATE | TOTAL_RUNS |
| --- | --- |
| 06 Jan 2018 | 3 |
| 13 Jan 2018 | 3 |

# Why would I want to do that?!




Pixabay

## Row Pattern Matching Use Cases

**Fraud Analytics**
2+ $1 trx between acts
1 $10,000 trx in 7 days

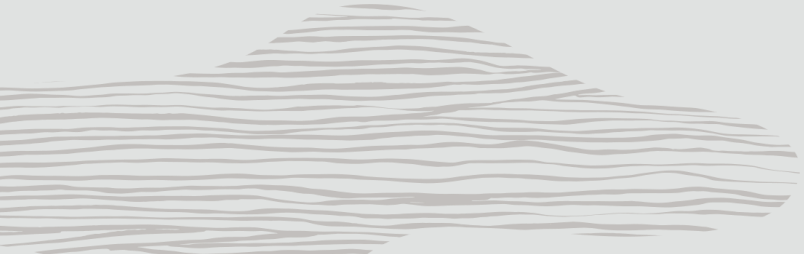
**Stock Market Trends**
Price rose 3 days
Then fell 3 days

**Customer Retention**
2+ orders/month for years
Max 2 orders past 6 mths

**Date Ranges**
Finding gaps & overlaps


How do
I debug it?
Gratisography

# (Regular) [exprsion]+ are easy to *missteak*



SAVE & SHARE

save regex ctrl+s

FLAVOR

- pcre (php) ✔
- javascript
- python
- golang

TOOLS

- code generator
- regex debugger

REGULAR EXPRESSION

2 matches, 27 steps (~0ms)

/ (5)(1){2,} / gmx

TEST STRING

SWITCH TO UNIT TESTS

1510511511



SUBSTITUTION

EXPLANATION

/ (5)(1){2,} / gmx

**1st Capturing Group** (5)

5 matches the character 5 literally (case sensitive)

**2nd Capturing Group** (1){2,}

{2,} **Quantifier** — Matches between 2 and unlimited times, as many times as possible, giving back as needed (greedy)

A repeated capturing group will only capture the last iteration. Put a capturing group around the repeated group to capture all iterations or use a non-capturing group instead if you're not interested in the data

MATCH INFORMATION

Match 1

| | | |
|---|---|---|
| Full match | 4-7 | `511` |
| Group 1. | 4-5 | `5` |
| Group 2. | 6-7 | `1` |

Match 2

| | | |
|---|---|---|
| Full match | 7-10 | `511` |
| Group 1. | 7-8 | `5` |
| Group 2. | 9-10 | `1` |

QUICK REFERENCE

Search reference

- all tokens
- common tokens ✔
- general tokens
- anchors
- meta sequences
- quantifiers

| | |
|---|---|
| A single character of: a, ... | [abc] |
| A character except: a, ... | [^abc] |
| A character in the range... | [a-z] |
| A character not in the r... | [^a-z] |
| A character in the ra... | [a-zA-Z] |
| Any single character | . |
| Any whitespace character | \s |

classifier

=> Which variable matched?

classifier

=> Which variable matched?

match_number

=> Which group is this?

classifier

=> Which variable matched?

match_number

=> Which group is this?

all rows per match

classifier

=> Which variable matched?

match_number

=> Which group is this?

all rows per match with unmatched rows

=> Show me everything!

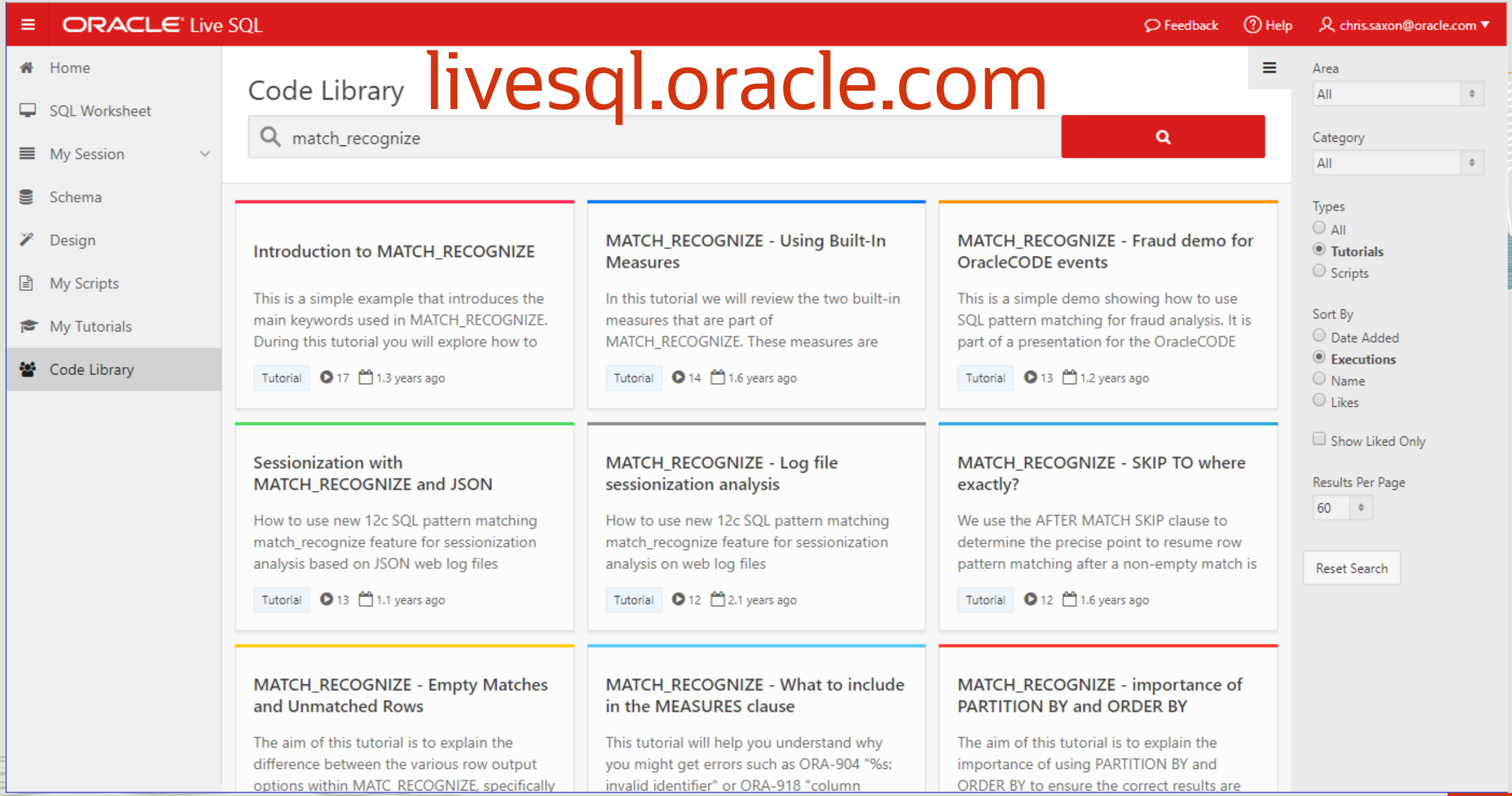
```
match_recognize (
   order by run_date
   measures
      classifier () as var,
      match_number () as grp
   all rows per match with unmatched rows
   pattern ( five_km one_km {2,} )
   define
      five_km as distance_in_km = 5,
      one_km as distance_in_km = 1
         and run_date < first ( run_date ) + 7
);
```

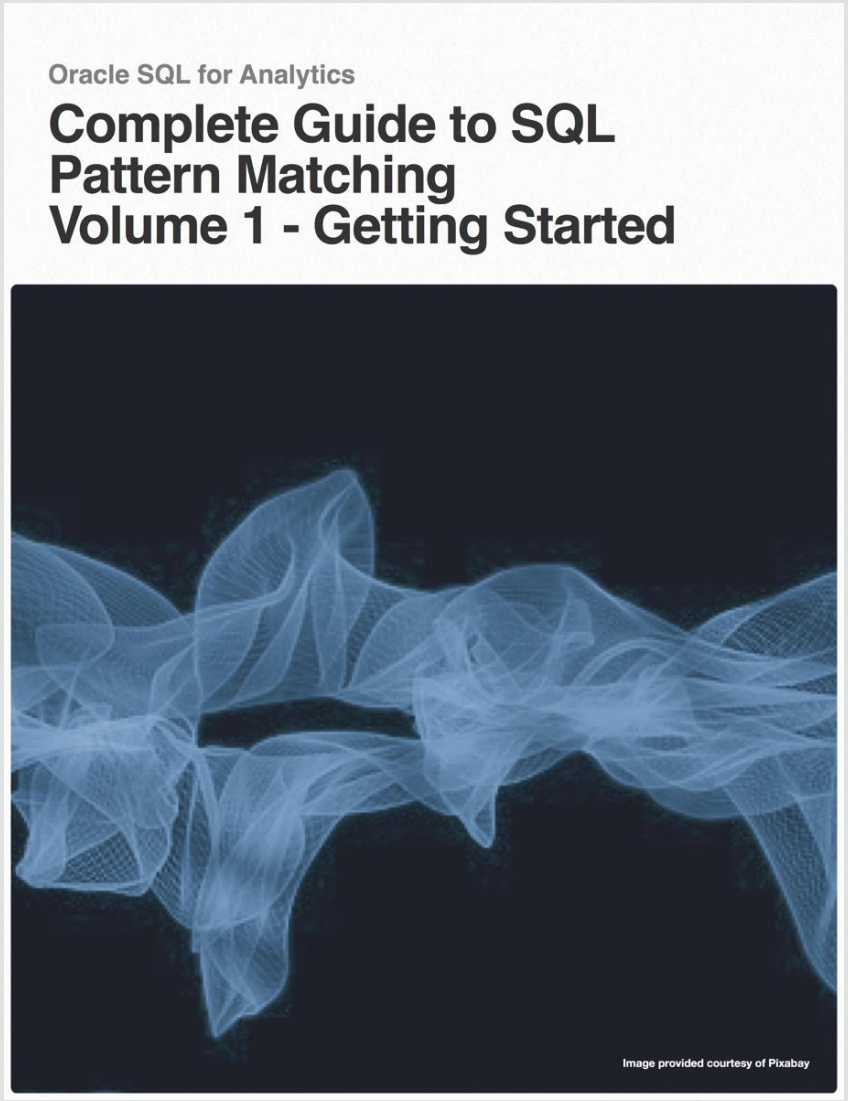
| RUN_DATE | VAR | GRP | TIME_IN_S | DISTANCE_IN_KM |
|---|---|---|---|---|
| 01 Jan 2018 | | | 310 | 1 |
| 02 Jan 2018 | | | 1,600 | 5 |
| 03 Jan 2018 | | | 3,580 | 10 |
| 06 Jan 2018 | FIVE_KM | 1 | 1,550 | 5 |
| 07 Jan 2018 | ONE_KM | 1 | 300 | 1 |
| 10 Jan 2018 | ONE_KM | 1 | 280 | 1 |
| 13 Jan 2018 | FIVE_KM | 2 | 1,530 | 5 |
| 14 Jan 2018 | ONE_KM | 2 | 295 | 1 |
| 15 Jan 2018 | ONE_KM | 2 | 292 | 1 |

| RUN_DATE | VAR | GRP | TIME_IN_S | DISTANCE_IN_KM |
|---|---|---|---|---|
| 01 Jan 2018 | | | 310 | 1 |
| **02 Jan 2018** | | | **1,600** | **5** |
| 03 Jan 2018 | | | 3,580 | 10 |
| | | | | |
| 06 Jan 2018 | FIVE_KM | 1 | 1,550 | 5 |
| 07 Jan 2018 | ONE_KM | 1 | 300 | 1 |
| 10 Jan 2018 | ONE_KM | 1 | 280 | 1 |
| | | | | |
| 13 Jan 2018 | FIVE_KM | 2 | 1,530 | 5 |
| 14 Jan 2018 | ONE_KM | 2 | 295 | 1 |
| 15 Jan 2018 | ONE_KM | 2 | 292 | 1 |


Want more?

Pixabay


ORACLE Live SQL
Feedback
Help
chris.saxon@oracle.com
Home
SQL Worksheet
My Session
Schema
Design
My Scripts
My Tutorials
Code Library
Code Library
livesql.oracle.com
match_recognize
Introduction to MATCH_RECOGNIZE
This is a simple example that introduces the main keywords used in MATCH_RECOGNIZE. During this tutorial you will explore how to
Tutorial 17 1.3 years ago
MATCH_RECOGNIZE - Using Built-In Measures
In this tutorial we will review the two built-in measures that are part of MATCH_RECOGNIZE. These measures are
Tutorial 14 1.6 years ago
MATCH_RECOGNIZE - Fraud demo for OracleCODE events
This is a simple demo showing how to use SQL pattern matching for fraud analysis. It is part of a presentation for the OracleCODE
Tutorial 13 1.2 years ago
Sessionization with MATCH_RECOGNIZE and JSON
How to use new 12c SQL pattern matching match_recognize feature for sessionization analysis based on JSON web log files
Tutorial 13 1.1 years ago
MATCH_RECOGNIZE - Log file sessionization analysis
How to use new 12c SQL pattern matching match_recognize feature for sessionization analysis on web log files
Tutorial 12 2.1 years ago
MATCH_RECOGNIZE - SKIP TO where exactly?
We use the AFTER MATCH SKIP clause to determine the precise point to resume row pattern matching after a non-empty match is
Tutorial 12 1.6 years ago
MATCH_RECOGNIZE - Empty Matches and Unmatched Rows
The aim of this tutorial is to explain the difference between the various row output options within MATC_RECOGNIZE, specifically
MATCH_RECOGNIZE - What to include in the MEASURES clause
This tutorial will help you understand why you might get errors such as ORA-904 "%s: invalid identifier" or ORA-918 "column
MATCH_RECOGNIZE - importance of PARTITION BY and ORDER BY
The aim of this tutorial is to explain the importance of using PARTITION BY and ORDER BY to ensure the correct results are
Area
All
Category
All
Types
All
Tutorials
Scripts
Sort By
Date Added
Executions
Name
Likes
Show Liked Only
Results Per Page
60
Reset Search





Keith Laker
Analytic SQL PM

iTunes & PDF

FREE!

**SQL for Data Warehousing and Analytics**
https://oracle-big-data.blogspot.co.uk


oracle-big-data.blogspot.co.uk
Thank You!
#MakeDataGreatAgain
Ryan McGuire / Gratisography