

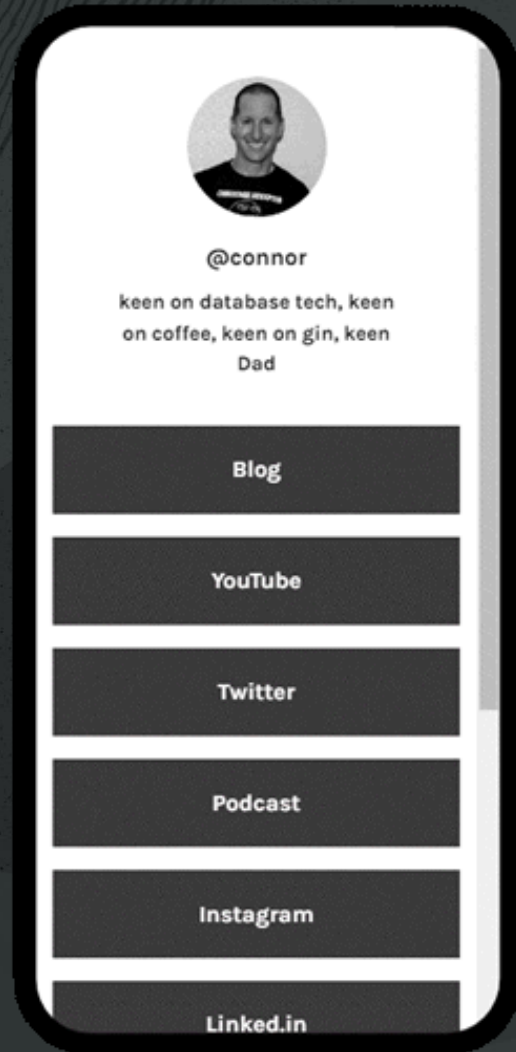
Futuristic SQL

Making the impossible possible

Connor McDonald
Database Advocate



Getting in touch is easy...



@connor_mc_d

<https://linktr.ee/connor>



key point

this session is not about ...

being a smarty ~~pants~~

we can do anything ...

```

SQL> with x( s, ind ) as
  2  ( select sud, instr( sud, '.' )
  3    from ( select replace(replace(
  4              replace(replace(:board, '-'), '|'), ' '), chr(10)) sud
  5              from dual )
  6    union all
  7    select substr(s,1,ind-1)||z||substr(s,ind+1)
  8          , instr(s, '.', ind+1)
  9    from x
 10          , ( select to_char( rownum ) z
 11              from dual connect by rownum <= 9 ) z
 12   where ind > 0
 13   and not exists (
 14     select null
 15     from ( select rownum lp from dual
 16           connect by rownum <= 9 )
 17     where z = substr(s, trunc((ind-1)/9)*9+lp, 1)

```

```

18         or      z = substr(s,mod(ind-1,9)-8+lp*9,1)
19         or      z = substr(s,mod(trunc((ind-1)/3),3)*3
20                               +trunc((ind-1)/27)*27+lp
21                               +trunc((lp-1)/3)*6,1)
22     )
23 ),
24 result as (
25     select s
26     from x
27     where ind = 0 )
28 select
29     regexp_replace(substr(s,(idx-1)*9+1,9),
30                     '(...)(...)(...)',
31                     '\1|\2|\3')||
32     case when mod(idx,3)=0 then chr(10)||rpad('-',11,'-') end soln
33 from result,
34     ( select level idx
35       from dual
36       connect by level <= 9 )

```



```
SQL> variable board varchar2(1000)
```

```
SQL> begin :board :=
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

```
9
```

```
10
```

```
11
```

```
12
```

```
13
```

```
14 end;
```

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | 3 | | 7 | | | | | |
| 6 | | | 1 | 9 | 5 | | | |
| | 9 | 8 | | | | | 6 | |
| 8 | | | | 6 | 1 | | | 3 |
| 4 | | | 8 | | 3 | | | 1 |
| 7 | | | | 2 | | | | 6 |
| | 6 | | | | | 2 | 8 | |
| | | | 4 | 1 | 9 | | | 5 |
| | | | | 8 | | | 7 | 9 |

SOLUTION

534 | 678 | 912

672 | 195 | 348

198 | 342 | 567

859 | 761 | 423

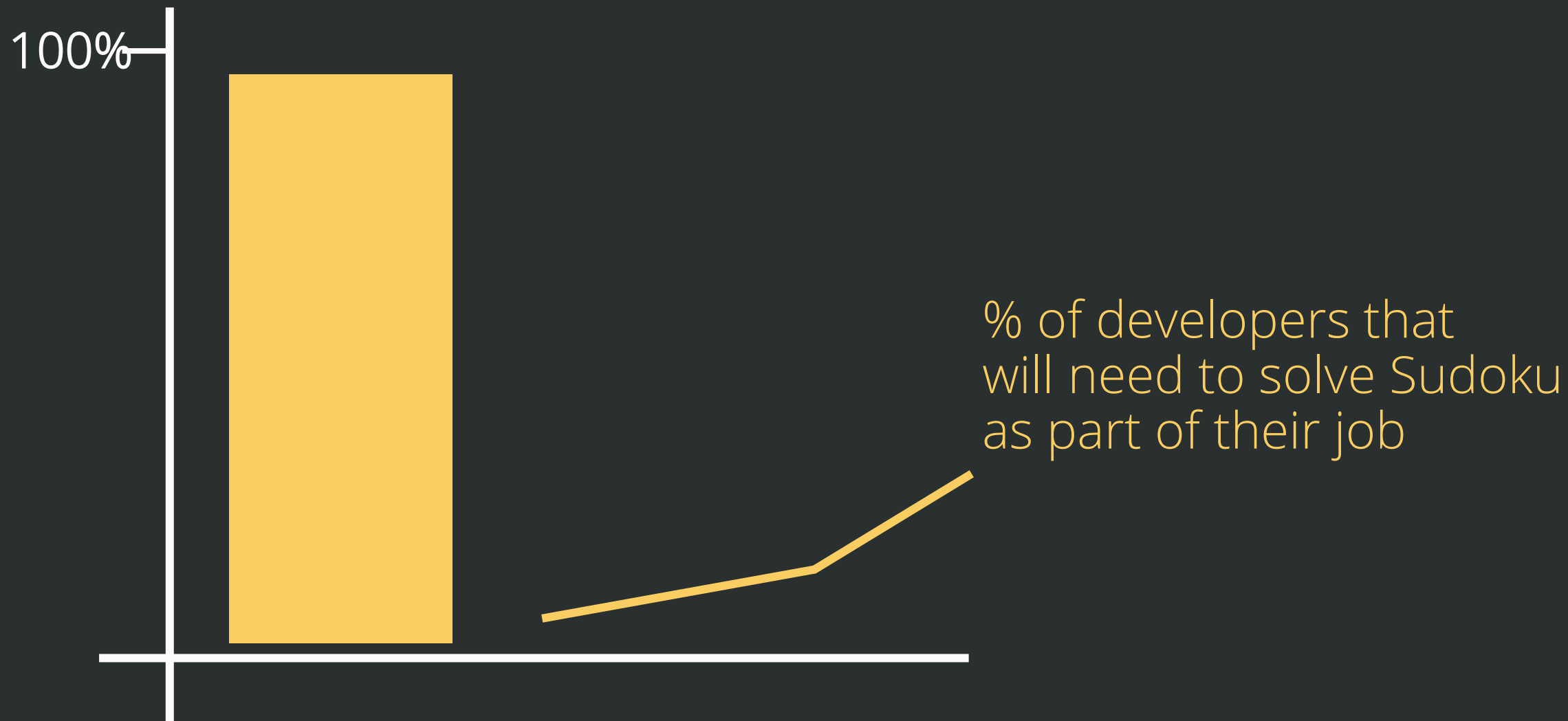
426 | 853 | 791

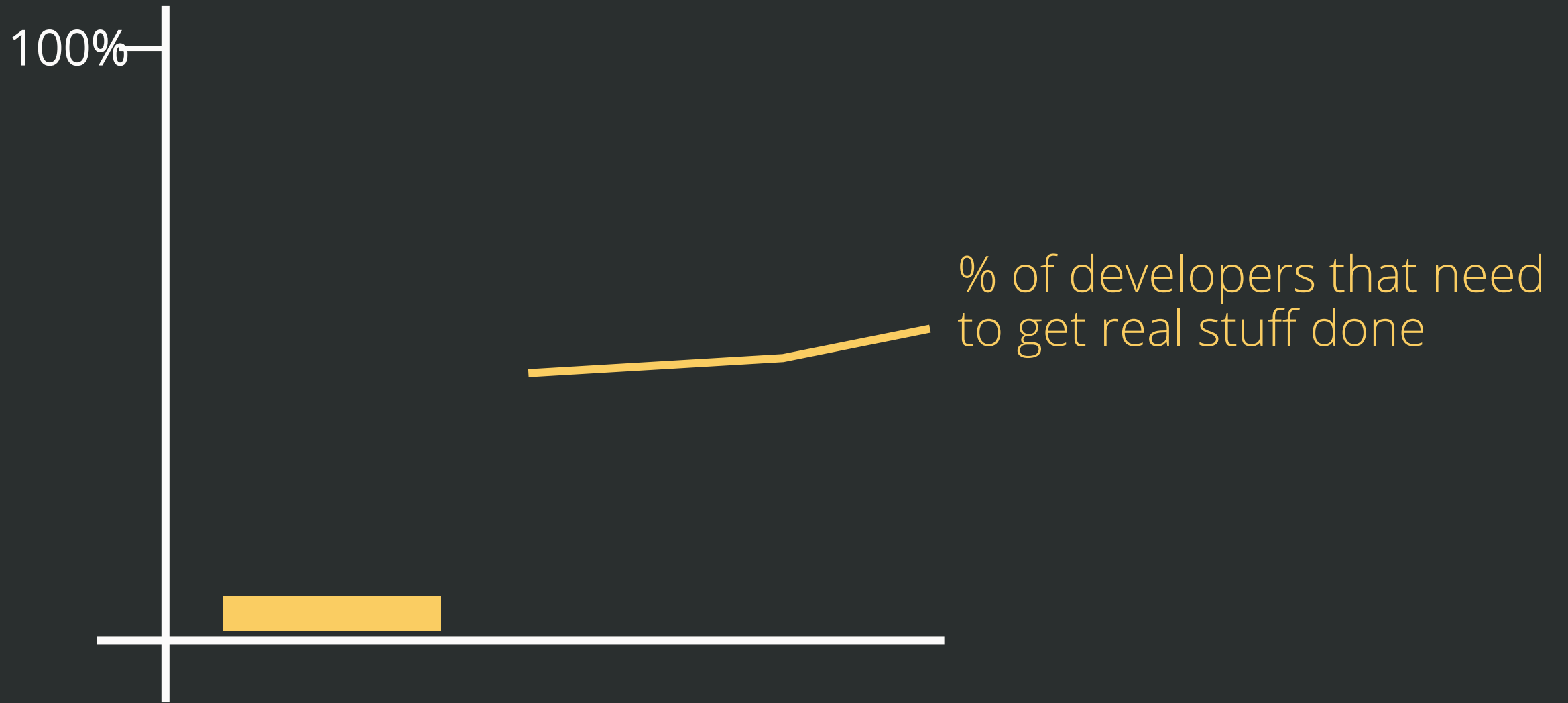
713 | 924 | 856

961 | 537 | 284

287 | 419 | 635

345 | 286 | 179





real stuff

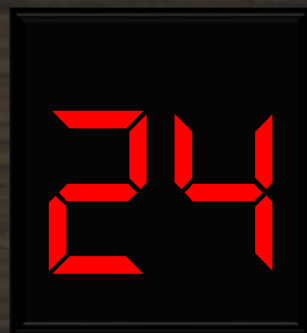
My typical weekend

... SQL edition

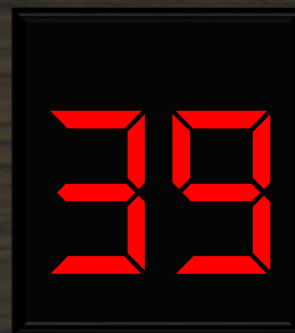


Son #1

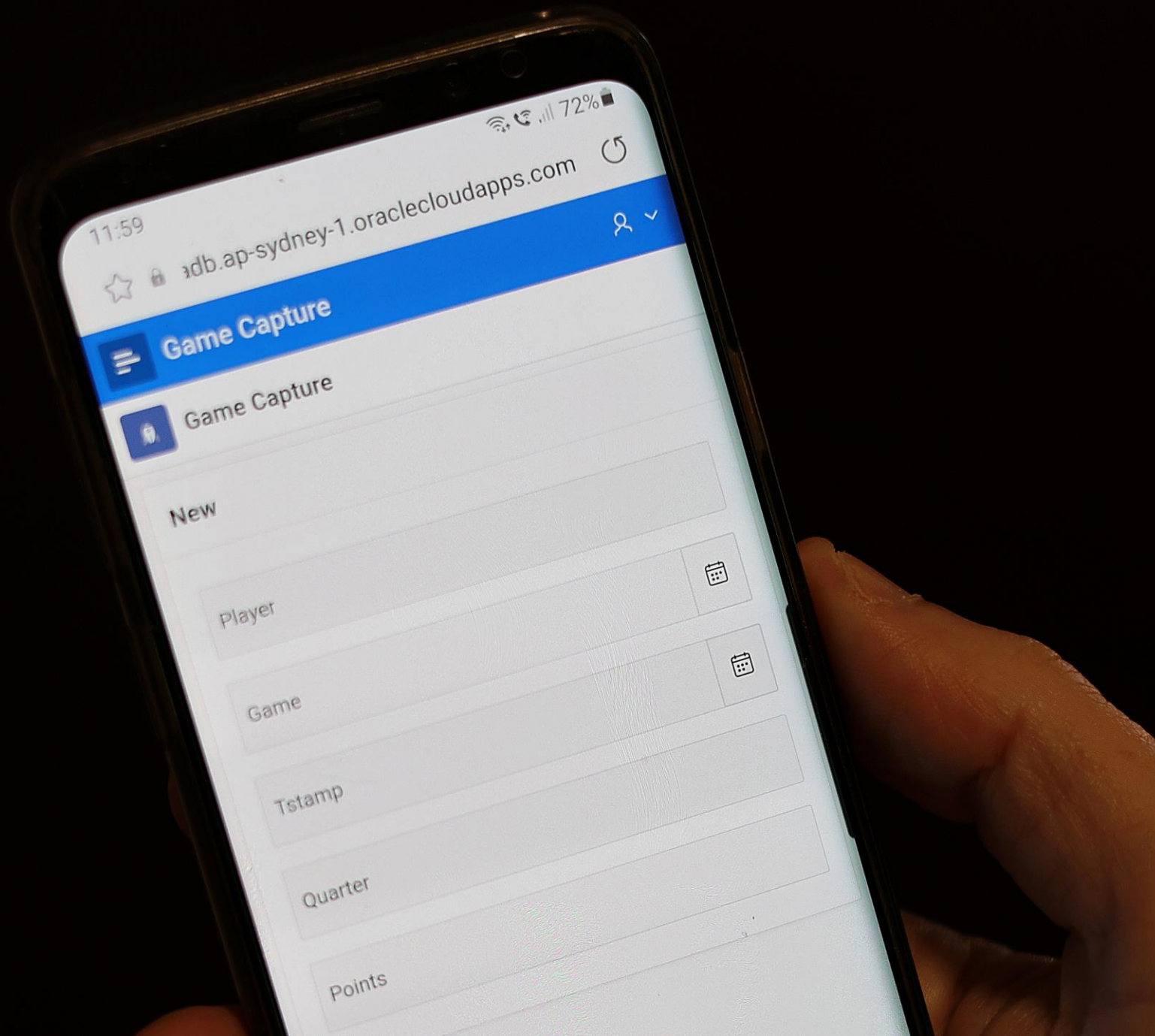




HOME



AWAY



apex.oracle.com



```
SQL> select quarter, tstamp, player, points
2   from   basketball
3   order by tstamp;
```

| QUARTER | TSTAMP | PLAYER | POINTS |
|---------|----------|----------|--------|
| 1 | 12:30:12 | Campbell | 1 |
| 1 | 12:31:57 | Robbie | 3 |
| 1 | 12:32:03 | Zack | 2 |
| 1 | 12:32:08 | Robbie | 1 |
| 1 | 12:32:19 | Robbie | 2 |
| 1 | 12:33:05 | Max | 3 |
| 1 | 12:33:08 | Campbell | 1 |
| 1 | 12:33:22 | Campbell | 1 |
| 1 | 12:33:59 | Campbell | 3 |
| 1 | 12:34:19 | Rory | 3 |
| 1 | 12:35:25 | Campbell | 3 |
| 1 | 12:35:50 | Matt | 2 |
| 1 | 12:35:54 | Robbie | 3 |
| 1 | 12:35:54 | Will | 3 |
| 1 | 12:36:07 | Matt | 3 |

...

...



*"I need the points per player,
plus quarter by quarter totals,
plus the grand total"*

```
SQL> select quarter, tstamp, player, points
2  from basketball
3  order by tstamp;
```

```
QUARTER SQL> select quarter, sum(points)
----- 2  from basketball
3  group by quarter
4  order by
```

```
QUARTER SUM(POINTS)
```

```
----- SQL> select sum(points)
from basketball;
```

```
...
```

```
SUM(POINTS)
```

```
-----
```

```
119
```

from 3 to 2

rollup

```
SQL> select quarter, tstamp, player, points
2   from   basketball
3   order by tstamp;
```

```
QUARTER SQL> select quarter, sum(points)
----- 2   from basketball
3   group by rollup(quarter)
4   order by 1;
```

| QUARTER | SUM (POINTS) |
|---------|--------------|
| ----- | ----- |
| 1 | 43 |
| 2 | 19 |
| 3 | 33 |
| 4 | 24 |
| ... | 119 |

still messy...

| QUARTER | PLAYER | POINTS |
|---------|----------|--------|
| 1 | Campbell | 1 |
| 1 | Robbie | 3 |
| 1 | Zack | 2 |
| 1 | Robbie | 1 |
| ... | | |
| 2 | Max | 3 |

| QUARTER | SUM (POINTS) |
|---------|--------------|
| 1 | 43 |
| 2 | 19 |
| 3 | 23 |
| 4 | 24 |
| | 119 |

"TL;DR ... the app can do this for me"

| | | | |
|--------------------------------|-------------------------------------|--------------------------|---------|
| <input type="text" value="Q"/> | | Go | Actions |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Quarter |
| Quarter : 1 | | | |
| Player | | | |
| Campbell | | | |
| Robbie | | | |
| Campbell | | | |
| Robbie | | | |
| Zack | | | |
| Will | | | |
| Max | | | |
| Campbell | | | |
| Campbell | | | |

Columns

Filter

Data

Format

Chart

Group By

Pivot

Report

Download

Help

Sort

Aggregate

Compute

Flashback

5/7/2021

5/7/2021

5/7/2021

5/7/2021

5/7/2021

5/7/2021

5/7/2021

```
select
    QUARTER,
    PLAYER,
    POINTS,
    sum(POINTS) over (partition by QUARTER),
    count(*) over () as apxws_row_cnt
from (
    select *
    from (
        select PLAYER,
            POINTS,
            QUARTER
        from EMP
    ) r
) r
order by "QUARTER"
```

from 2 to 1

```

SQL> select quarter,
2         nv12(ball_pk,max(player),null) player,
3         nv12(ball_pk,max(timestamp),null) timestamp,
4         sum(points)
5   from   basketball
6  group by rollup(quarter,ball_pk)
7  order by quarter,timestamp;

```

| QUARTER | PLAYER | TSTAMP | SUM(POINTS) |
|---------|----------|----------|-------------|
| ----- | ----- | ----- | ----- |
| 1 | Campbell | 12:30:12 | 10 |
| 1 | Robbie | 12:31:57 | 3 |
| 1 | Zack | 12:32:03 | 2 |
| 1 | | | 43 |
| 2 | Robbie | 13:00:37 | 2 |
| ... | | | |
| 2 | | | 19 |
| 4 | Max | 14:11:54 | 1 |
| 4 | | | 24 |
| | | | 119 |

all totals are possible

```
SQL> select quarter,player,sum(points) from basketball
      2 group by cube(quarter,player);
```

| QUARTER | PLAYER | SUM(POINTS) |
|---------|----------|-------------|
| 1 | Campbell | 10 |
| 1 | Matt | 5 |
| | ... | |
| 1 | Rory | 5 |
| 1 | Will | 3 |
| 1 | Zack | 4 |
| 1 | | 43 |
| | Campbell | 26 |
| | Matt | 12 |
| | ... | |
| | Rory | 15 |
| | Will | 16 |
| | Zack | 15 |
| | | 119 |



totally customisable


```
SQL> select quarter,player,sum(points)
2   from   basketball
3   group by grouping sets (
4       (player),(quarter), () );
```

| QUARTER | PLAYER | SUM(POINTS) |
|---------|----------|-------------|
| ----- | ----- | ----- |
| 1 | | 43 |
| 2 | | 19 |
| 3 | | 33 |
| 4 | | 24 |
| | Campbell | 26 |
| | Matt | 12 |
| | Max | 11 |
| | Robbie | 24 |
| | Rory | 15 |
| | Will | 16 |
| | Zack | 15 |
| | | 119 |

```
SQL> select quarter,player,sum(points)
2   from   basketball
3   group by grouping sets (
4       (player), (quarter), () );
```

| QUARTER | PLAYER | SUM(POINTS) |
|---------|----------|-------------|
| 1 | | 43 |
| 2 | | 19 |
| 3 | | 33 |
| 4 | | 24 |
| | Campbell | 26 |
| | Matt | 12 |
| | Max | 11 |
| | Robbie | 24 |
| | Rory | 15 |
| | Will | 16 |
| | Zack | 15 |
| | | 119 |

```
SQL> select quarter,player,sum(points)
2   from   basketball
3   group by grouping sets (
4       (player), (quarter), () );
```

| QUARTER | PLAYER | SUM(POINTS) |
|---------|----------|-------------|
| 1 | | 43 |
| 2 | | 19 |
| 3 | | 33 |
| 4 | | 24 |
| | Campbell | 26 |
| | Matt | 12 |
| | Max | 11 |
| | Robbie | 24 |
| | Rory | 15 |
| | Will | 16 |
| | Zack | 15 |
| | | 119 |

```
SQL> select quarter,player,sum(points)
2   from   basketball
3   group by grouping sets (
4       (player),(quarter), () );
```

| QUARTER | PLAYER | SUM(POINTS) |
|---------|----------|-------------|
| 1 | | 43 |
| 2 | | 19 |
| 3 | | 33 |
| 4 | | 24 |
| | Campbell | 26 |
| | Matt | 12 |
| | Max | 11 |
| | Robbie | 24 |
| | Rory | 15 |
| | Will | 16 |
| | Zack | 15 |
| | | 119 |

Search...

Go

▼ Job

- ☐ CLERK (4)
- ☐ SALESMAN (4)
- ☐ MANAGER (3)
- ☐ ANALYST (2)
- ☐ PRESIDENT (1)

▼ Salary

- ☐ <900 (1)
- ☐ 900 - 1300 (4)
- ☐ 1300 - 2000 (3)
- ☐ 2000 - 2500 (1)
- ☐ >=2500 (5)

to

Go

▼ Deptno

- ☐ 30 (6)
- ☐ 20 (5)

Employee Name ↑

Job

Mgr

Hired

ADAMS

CLERK

7,788

1/12/1983

ALLEN

SALESMAN

7,698

2/20/1981

BLAKE

MANAGER

7,839

5/1/1981

CLARK

MANAGER

7,839

6/9/1981

FORD

ANALYST

7,566

12/3/1981

JAMES

CLERK

7,698

12/3/1981

JONES

MANAGER

7,839

4/2/1981

KING

PRESIDENT

11/17/1981

MARTIN

SALESMAN

7,698

9/28/1981

MILLER

CLERK

7,782

1/23/1982

SCOTT

ANALYST

7,566

12/9/1982

SMITH

CLERK

7,902

12/17/1980

TURNER

SALESMAN

7,698

9/8/1981

WARD

SALESMAN

7,698

2/22/1981



```

select *
from(
    select
        grouping_id( "JOB", "APX$BUCKET3", "DEPTNO") "APX$GRPID",
        "JOB" "APX$FLTV2",
        count(*) "APX$FLTC2",
        "APX$BUCKET3" "APX$FLTV3",
        count(*) "APX$FLTC3",
        "DEPTNO" "APX$FLTV4",
        count(*) "APX$FLTC4",
        count(*) APX$ALLC
    from(
        select i.*
            from (select "ENAME", "JOB", "SAL", "DEPTNO"
                    from ((select /*+ qb_name(apex$inner) */
                            d."ENAME", d."JOB", d."SAL", d."DEPTNO"
                        from (select x.* from "EMP" x
                                ) d )) i ) i ) )
    group by grouping sets ("JOB", "APX$BUCKET3", "DEPTNO", ())
)

```

Your Basketball data



Suzy Parent

To: Connor McDonald

Hi Connor,

I saw the report you gave to the coach from the game.

Thanks,
Suzy

```
SQL> select rownum qtr
2   from   dual
3   connect by level <= 4;
```

| QTR |
|-----|
| 1 |
| 2 |
| 3 |
| 4 |

```
SQL> select quarter, player, sum(points)
2   from basketball
3   group by quarter, player
4   order by 1,2;
```

| QUARTER | PLAYER | PTS |
|---------|----------|-----|
| 1 | Campbell | 10 |
| 1 | Matt | 5 |
| 1 | Max | 6 |
| 1 | Robbie | 10 |
| 1 | Rory | 5 |
| 1 | Will | 3 |
| 1 | Zack | 4 |
| 2 | Campbell | 1 |
| 2 | Matt | 2 |
| 2 | Max | 2 |
| ... | | |

player results by quarter

conventional outer join

```

SQL> select qtr, player, pts
      2  from
      3      ( select quarter, player, sum(points) pts
      4          from basketball
      5          group by quarter, player ) b
      6  right outer join
      7      ( select rownum qtr from dual connect by level <= 4 ) q
      8  on ( q.qtr = b.quarter )
      9  order by 2,1;

```

| QTR | PLAYER | PTS |
|-----|----------|-----|
| 1 | Campbell | 10 |
| 2 | Campbell | 1 |
| 3 | Campbell | 9 |
| 4 | Campbell | 6 |
| 1 | Matt | 5 |
| 2 | Matt | 2 |
| 4 | Matt | 5 |
| 1 | Max | 6 |
| 3 | Max | 2 |
| 4 | Max | 1 |

the data is sparse

```
SQL> select rownum qtr
2   from   dual
3   connect by level <= 4;
```

| QTR |
|-----|
| 1 |
| 2 |
| 3 |
| 4 |

```
SQL> select quarter, player, sum(points)
2   from basketball
3   group by quarter, player
4   order by 1,2;
```

| QUARTER | PLAYER | PTS |
|---------|----------|-----|
| 1 | Campbell | 10 |
| 1 | Matt | 5 |
| 1 | Max | 6 |
| 1 | Robbie | 10 |
| 1 | Rory | 5 |
| 1 | Will | 3 |
| 1 | Zack | 4 |
| 2 | Campbell | 1 |
| 2 | Matt | 2 |
| 2 | Max | 2 |
| ... | | |

```
SQL> select rownum qtr
      2   from   dual
      3   connect by level <= 4;
```

| QTR |
|-----|
| 1 |
| 2 |
| 3 |
| 4 |

x "Campbell"

x "Max"

...

x "Player n"

partitioned outer join

```

SQL> select qtr, player, nvl(pts,0)
  2   from
  3       ( select quarter, player, sum(points) pts
  4           from basketball
  5           group by quarter, player ) b
  6   partition by (b.player)
  7   right outer join
  8       ( select rownum qtr from dual connect by level <= 4 ) q
  9   on ( q.qtr = b.quarter )
 10  order by 2,1;

```

| QTR | PLAYER | PTS |
|-----|----------|-----|
| 1 | Campbell | 10 |
| 2 | Campbell | 1 |
| 3 | Campbell | 9 |
| 4 | Campbell | 6 |
| 1 | Matt | 5 |
| 2 | Matt | 2 |
| 3 | Matt | 0 |
| 4 | Matt | 5 |
| 1 | Max | 6 |
| 2 | Max | 0 |
| 3 | Max | 2 |
| 4 | Max | 1 |



Go

Actions ▾



Player



Player : Campbell

| Qtr | Pts |
|-----|-----|
| 1 | 10 |
| 2 | 1 |
| 3 | 9 |
| 4 | 6 |

Player : Matt

| Qtr | Pts |
|-----|-----|
| 1 | 5 |
| 2 | 2 |
| 3 | |

Boosting the offense



Mary Coach

To: Connor McDonald

Hi Connor,

We have a new boy who is looking at joining the team.

I don't know much about him, but lets assume he averages 13 points per game.

Thanks,
Mary

```
SQL> select player, sum(points)
      2   from   basketball
      3   group by player;
```

| PLAYER | SUM (POINTS) |
|----------|--------------|
| ----- | ----- |
| Will | 19 |
| Campbell | 23 |
| Robbie | 12 |
| Zack | 18 |
| Rory | 12 |
| Max | 23 |
| Matt | 10 |



hypothetical analytics

```
SQL> select rank(13) within group ( order by pts ) ranking
2   from
3     ( select player, sum(points) pts
4       from   basketball
5       group by player
6     ) ;
```

RANKING

3

*"What if he gets 4 points **per quarter**?"*

```

SQL> select
      2     quarter,
      3     rank(4) within group ( order by pts ) ranking
      4 from
      5     ( select player, quarter, sum(points) pts
      6         from   basketball
      7         group by player, quarter
      8     )
      9 group by quarter
     10 order by 1;

```

| QUARTER | RANKING |
|---------|---------|
| ----- | ----- |
| 1 | 5 |
| 2 | 4 |
| 3 | 4 |
| 4 | 3 |



Son #2


```
SQL> select *
      2  from long_jump
      3  order by 1,2;
```

| COMP# | JUMP | WIND | DISTANCE | NAME |
|-------|------|------|----------|--------|
| 1 | 1 | Head | 6.16 | Liam |
| 1 | 2 | Head | 6.34 | Aden |
| 1 | 3 | Head | 6.07 | Noah |
| 1 | 4 | Head | 5.74 | Ethan |
| 2 | 1 | Head | 5.61 | Noah |
| 2 | 2 | Head | 6.21 | Daniel |
| 2 | 3 | Head | 6.26 | Liam |
| 2 | 4 | Head | 5.98 | Noah |
| 3 | 1 | None | 6.58 | Oliver |
| 3 | 2 | None | 6.45 | Oliver |
| 3 | 3 | None | 6.53 | James |
| 3 | 4 | None | 6.03 | Oliver |
| 4 | 1 | Tail | 5.86 | David |
| 4 | 2 | Tail | 5.85 | James |
| 4 | 3 | Tail | 5.94 | Liam |
| 4 | 4 | Tail | 6.14 | James |

...

...



*"Find the highest jump achieved
by each competitor"*

```
SQL> select name, max(distance) hi
      2  from long_jump
      3  group by name
      4  order by 1;
```

| NAME | HI |
|--------|-------|
| ----- | ----- |
| Aden | 7.16 |
| Alex | 6.79 |
| Ben | 6.19 |
| Daniel | 6.32 |
| David | 6.03 |
| Ethan | 6.38 |
| Henry | 6.56 |
| Jack | 6.67 |
| James | 6.14 |
| John | 6.51 |
| Joseph | 6.49 |
| ... | |
| ... | |



*"...and what was the wind like
when they achieved that?"*

```
SQL> select name, max(distance) hi  
2   from long_jump  
3   group by name  
4   order by 1;
```

```
SQL> select name, wind, max(distance) hi  
2   from long_jump  
3   group by name  
4   order by 1;
```

```
select name, wind, max(goals) hi  
*
```

ERROR at line 1:

ORA-00979: not a GROUP BY expression

KEEP clause

order by 'x' but output 'y'


```

SQL> select name,
2         max(distance) as hi,
3         max(wind) keep ( dense_rank last order by distance ) as hi_pos
4   from long_jump
5  group by name
6  order by 1;

```

| NAME | HI | HI_POS |
|--------|-------|--------|
| ----- | ----- | ----- |
| Aden | 7.16 | None |
| Alex | 6.79 | Tail |
| Ben | 6.19 | Head |
| Daniel | 6.32 | Head |
| David | 6.03 | Head |
| Ethan | 6.38 | Tail |
| Henry | 6.56 | None |
| Jack | 6.67 | Tail |
| James | 6.14 | Tail |
| John | 6.51 | Tail |
| Joseph | 6.49 | Tail |
| Julian | 6.42 | Tail |
| ... | | |

```

SQL> select name,
2         max(distance) as hi,
3         max(wind) keep ( dense_rank last order by distance ) as hi_pos
4   from long_jump
5  group by name
6  order by 1;

```

| NAME | HI | HI_POS |
|--------|-------|--------|
| ----- | ----- | ----- |
| Aden | 7.16 | None |
| Alex | 6.79 | Tail |
| Ben | 6.19 | Head |
| Daniel | 6.32 | Head |
| David | 6.03 | Head |
| Ethan | 6.38 | Tail |
| Henry | 6.56 | None |
| Jack | 6.67 | Tail |
| James | 6.14 | Tail |
| John | 6.51 | Tail |
| Joseph | 6.49 | Tail |
| Julian | 6.42 | Tail |
| ... | | |

```

SQL> select name,
2         max(distance) as hi,
3         max(wind) keep ( dense_rank last order by distance ) as hi_pos
4   from long_jump
5  group by name
6  order by 1;

```

| NAME | HI | HI_POS |
|--------|-------|--------|
| ----- | ----- | ----- |
| Aden | 7.16 | None |
| Alex | 6.79 | Tail |
| Ben | 6.19 | Head |
| Daniel | 6.32 | Head |
| David | 6.03 | Head |
| Ethan | 6.38 | Tail |
| Henry | 6.56 | None |
| Jack | 6.67 | Tail |
| James | 6.14 | Tail |
| John | 6.51 | Tail |
| Joseph | 6.49 | Tail |
| Julian | 6.42 | Tail |
| ... | | |

```
SQL> select name,
2         max(distance) as hi,
3         any_value(wind) keep ( dense_rank last order by distance ) as hi_pos
4   from long_jump
5  group by name
6  order by 1;
```

| NAME | HI | HI_POS |
|--------|-------|--------|
| ----- | ----- | ----- |
| Aden | 7.16 | None |
| Alex | 6.79 | Tail |
| Ben | 6.19 | Head |
| Daniel | 6.32 | Head |
| David | 6.03 | Head |
| Ethan | 6.38 | Tail |
| Henry | 6.56 | None |
| Jack | 6.67 | Tail |
| James | 6.14 | Tail |
| John | 6.51 | Tail |
| Joseph | 6.49 | Tail |
| Julian | 6.42 | Tail |
| ... | | |



as requirements get more complex ...

... SQL gets more complex

*"Find the average distance jumped by each boy,
then with the median of these results,
list those players who got above the median"*

SQL ?

common table expressions

WITH clause

```
select jump, avg(distance)
from long_jump
group by jump
;
```

"Who cares?... more code, same result"

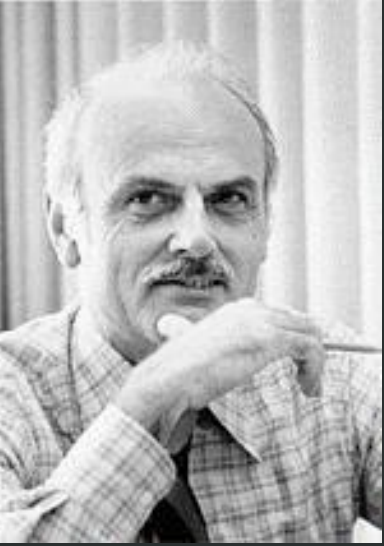
great mental model for developers

relational is a rigorous model ...

relational is the dominant model ...

relational ...

not our fault 😊



"data is represented as mathematical n -ary relations, an n -ary relation being a subset of the Cartesian product of n domains."



НУН?



procedural approach to relational

step by step

```
SQL> with
  2  jump_avg as (
  3      select name, avg(distance) avg_per_jump
  4      from    long_jump
  5      group by name
  6  ),
```

"Find the average distance jumped by each boy..."

```
7  median_jump as
8  ( select median(avg_per_jump) median_jump
9    from jump_avg
10  )
```

"then with the median of these results..."

```
11  select *
12  from jump_avg,
13       median_jump
14  where avg_per_jump > median_jump
15  order by 1;
```

"list those players who got above the median"

```

SQL> with
  2  jump_avg as (
  3      select name, avg(distance) avg_per_jump
  4      from    long_jump
  5      group by name
  6  ),
  7  median_jump as
  8  ( select median(avg_per_jump) median_jump
  9      from jump_avg
10  )
11  select *
12  from jump_avg,
13       median_jump
14  where avg_per_jump > median_jump
15  order by 1;

```

| NAME | AVG_PER_JUMP | MEDIAN_JUMP |
|-------|--------------|-------------|
| ----- | ----- | ----- |
| Aden | 6.89 | 6.15 |
| Alex | 6.33 | 6.15 |
| Ethan | 6.23 | 6.15 |

programmer's approach....

... relational solution

great for code reuse

modern apps need JSON

recall partitioned outer join

```

select qtr, player, pts
from
  ( select quarter, player, sum(points) pts
    from basketball
    group by quarter, player ) b
right outer join
  ( select rownum, player, qtr
    from basketball
    on ( q.qtr = b.qtr

```

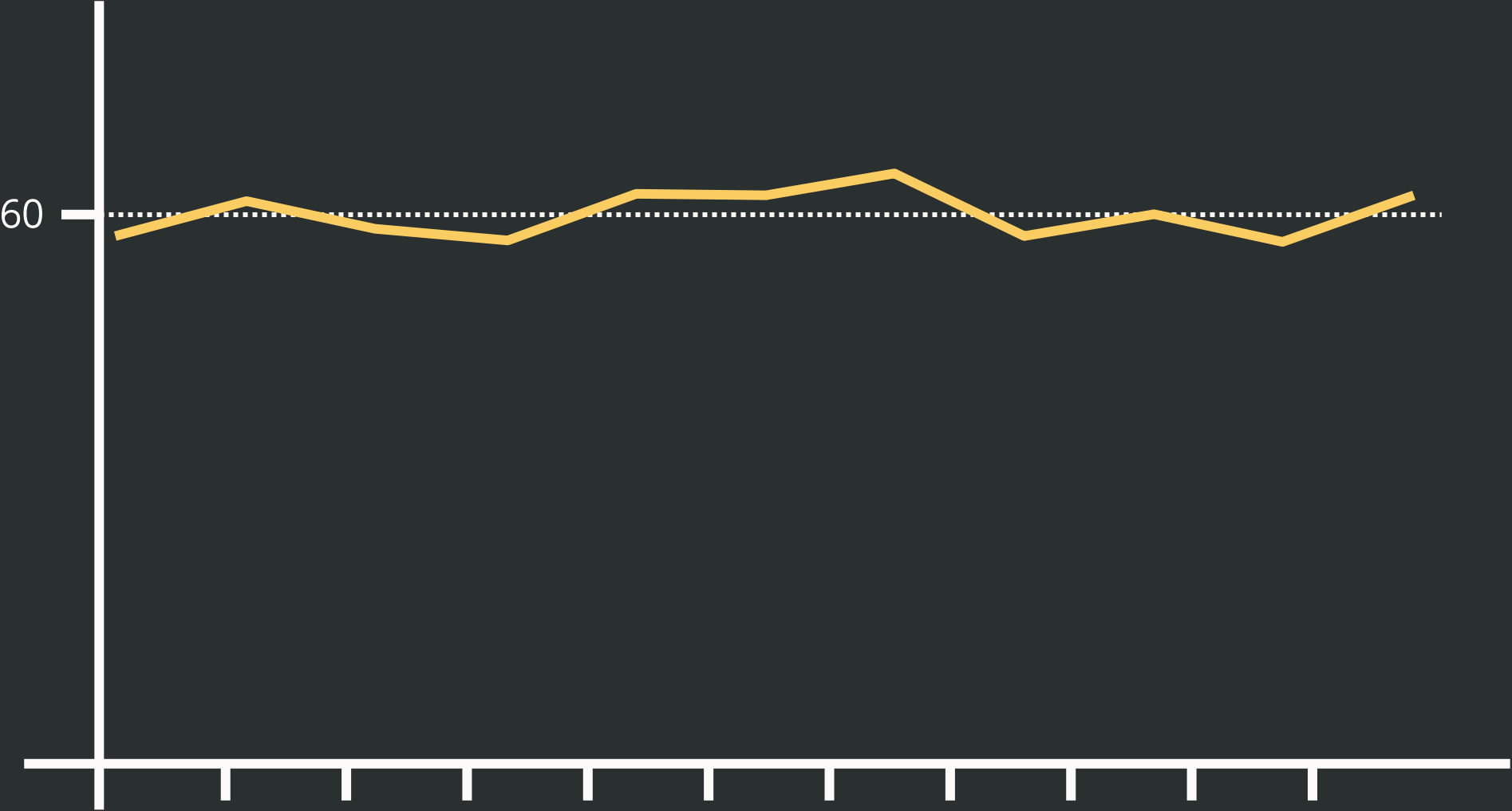
```

    {"Campbell":9},
    {"Zack":6},
    {"Will":4},
    {"Rory":3},
    {"Robbie":11},
    {"Max":6},
    ...
    {"Matt":1}
  ]

```



Some "me" time




```
SQL> select *
      2  from swimming
      3  order by sess,lap;
```

| SESS | LAP | ELA |
|-----------|-----|------|
| 12-JUL-22 | 1 | 58.7 |
| 12-JUL-22 | 2 | 59.7 |
| 12-JUL-22 | 3 | 60.3 |
| 12-JUL-22 | 4 | 61.3 |
| 12-JUL-22 | 5 | 60.7 |
| 12-JUL-22 | 6 | 59.7 |
| 12-JUL-22 | 7 | 60.2 |
| 12-JUL-22 | 8 | 58.6 |
| 12-JUL-22 | 9 | 59.6 |
| 12-JUL-22 | 10 | 59.9 |
| 12-JUL-22 | 11 | 60.4 |
| 12-JUL-22 | 12 | 60.1 |

```
SQL> select avg(ela) from swimming;
```

| AVG (ELA) |
|-----------|
| 59.93 |



"How often am I 'on track'?"

moving average every 3 laps

```

SQL> select
2     s.*,
3     avg(ela) over ( partition by sess order by lap
4                     range between 1 preceding and 1 following
5                     ) as mov_avg
6 from swimming s;

```

| SESS | LAP | ELA | MOV_AVG |
|-----------|-------|-------|---------|
| ----- | ----- | ----- | ----- |
| 12-JUL-22 | 1 | 58.7 | 59.2 |
| 12-JUL-22 | 2 | 59.7 | 59.6 |
| 12-JUL-22 | 3 | 60.3 | 60.4 |
| 12-JUL-22 | 4 | 61.3 | 60.8 |
| 12-JUL-22 | 5 | 60.7 | 60.6 |
| 12-JUL-22 | 6 | 59.7 | 60.2 |
| 12-JUL-22 | 7 | 60.2 | 59.5 |
| 12-JUL-22 | 8 | 58.6 | 59.5 |
| 12-JUL-22 | 9 | 59.6 | 59.4 |
| 12-JUL-22 | 10 | 59.9 | 60.0 |
| 12-JUL-22 | 11 | 60.4 | 60.1 |
| 12-JUL-22 | 12 | 60.1 | 60.3 |

```

SQL> select
      2      s.*,
      3      avg(ela) over ( partition by sess order by lap
      4                          range between 1 preceding and 1 following
      5                          ) as mov_avg
      6  from swimming s;

```

| SESS | LAP | ELA | MOV_AVG |
|-----------|-------|-------|---------|
| ----- | ----- | ----- | ----- |
| 12-JUL-22 | 1 | 58.7 | 59.2 |
| 12-JUL-22 | 2 | 59.7 | 59.6 |
| 12-JUL-22 | 3 | 60.3 | 60.4 |
| 12-JUL-22 | 4 | 61.3 | 60.8 |
| 12-JUL-22 | 5 | 60.7 | 60.6 |
| 12-JUL-22 | 6 | 59.7 | 60.2 |
| 12-JUL-22 | 7 | 60.2 | 59.5 |
| 12-JUL-22 | 8 | 58.6 | 59.5 |
| 12-JUL-22 | 9 | 59.6 | 59.4 |
| 12-JUL-22 | 10 | 59.9 | 60.0 |
| 12-JUL-22 | 11 | 60.4 | 60.1 |
| 12-JUL-22 | 12 | 60.1 | 60.3 |

```

SQL> select
  2     s.*,
  3     avg(ela) over ( partition by sess order by lap
  4                       range between 1 preceding and 1 following
  5                       ) as mov_avg
  6 from swimming s;

```

| SESS | LAP | ELA | MOV_AVG |
|-----------|-------|-------|---------|
| ----- | ----- | ----- | ----- |
| 12-JUL-22 | 1 | 58.7 | 59.2 |
| 12-JUL-22 | 2 | 59.7 | 59.6 |
| 12-JUL-22 | 3 | 60.3 | 60.4 |
| 12-JUL-22 | 4 | 61.3 | 60.8 |
| 12-JUL-22 | 5 | 60.7 | 60.6 |
| 12-JUL-22 | 6 | 59.7 | 60.2 |
| 12-JUL-22 | 7 | 60.2 | 59.5 |
| 12-JUL-22 | 8 | 58.6 | 59.5 |
| 12-JUL-22 | 9 | 59.6 | 59.4 |
| 12-JUL-22 | 10 | 59.9 | 60.0 |
| 12-JUL-22 | 11 | 60.4 | 60.1 |
| 12-JUL-22 | 12 | 60.1 | 60.3 |

```

SQL> select
2     s.*,
3     avg(ela) over ( partition by sess order by lap
4                     range between 1 preceding and 1 following
5                     ) as mov_avg
6 from swimming s;

```

| SESS | LAP | ELA | MOV_AVG |
|-----------|-------|-------|---------|
| ----- | ----- | ----- | ----- |
| 12-JUL-22 | 1 | 58.7 | 59.2 |
| 12-JUL-22 | 2 | 59.7 | 59.6 |
| 12-JUL-22 | 3 | 60.3 | 60.4 |
| 12-JUL-22 | 4 | 61.3 | 60.8 |
| 12-JUL-22 | 5 | 60.7 | 60.6 |
| 12-JUL-22 | 6 | 59.7 | 60.2 |
| 12-JUL-22 | 7 | 60.2 | 59.5 |
| 12-JUL-22 | 8 | 58.6 | 59.5 |
| 12-JUL-22 | 9 | 59.6 | 59.4 |
| 12-JUL-22 | 10 | 59.9 | 60.0 |
| 12-JUL-22 | 11 | 60.4 | 60.1 |
| 12-JUL-22 | 12 | 60.1 | 60.3 |

*"want the **tempo** to be 60 seconds"*


```
SQL> select *
      2  from swimming
      3  order by sess,lap;
```

| | SESS | LAP | ELA | |
|----|-----------|-------|-------|--------|
| | ----- | ----- | ----- | |
| 60 | 12-JUL-22 | 1 | 58.7 | { -1.3 |
| | 12-JUL-22 | 2 | 59.7 | |
| | 12-JUL-22 | 3 | 60.2 | |
| | 12-JUL-22 | 4 | 61.4 | |
| | 12-JUL-22 | 5 | 60.7 | { -0.3 |
| | 12-JUL-22 | 6 | 59.7 | |
| | 12-JUL-22 | 7 | 60.2 | |
| | 12-JUL-22 | 8 | 58.6 | |
| 60 | 12-JUL-22 | 9 | 59.6 | { +0.2 |
| | 12-JUL-22 | 10 | 59.9 | |
| | 12-JUL-22 | 11 | 60.4 | |
| | 12-JUL-22 | 12 | 60.1 | |

+1.4
====
0.0



```
SQL> alter table swimming add delta number(5,2)
      2      generated always as ( ela - 60 );
```

Table altered.

```
SQL> alter table swimming modify delta invisible;
```

Table altered.

```
SQL> select s.*,
2      sum(delta) over ( order by lap ) as run_tot
3  from    swimming s
4  order by sess,lap;
```

| SESS | LAP | ELA | DELTA | RUN_TOT | |
|-----------|-------|-------|-------|---------|----------|
| ----- | ----- | ----- | ----- | ----- | |
| 12-JUL-22 | 1 | 58.7 | -1.3 | -1.3 | "Bad" |
| 12-JUL-22 | 2 | 59.7 | -.3 | -1.6 | "Bad" |
| 12-JUL-22 | 3 | 60.3 | .3 | -1.3 | "Bad" |
| 12-JUL-22 | 4 | 61.3 | 1.3 | 0 | "Zero" 😊 |
| 12-JUL-22 | 5 | 60.7 | .7 | .7 | |
| 12-JUL-22 | 6 | 59.7 | -.3 | .4 | |
| 12-JUL-22 | 7 | 60.2 | .2 | .6 | |

pattern

`bad_lap* zero`

`zero as sum(delta) = 0`

```

SQL> select * from swimming
      2  match_recognize (
      3      partition by sess order by lap
      4      measures classifier() pattern, sum(delta) as run_tot
      5      all rows per match
      6      pattern (bad_lap* zero)
      7      define zero as sum(delta) = 0
      9  );

```

| SESS | LAP | PATTERN | RUN_TOT | ELA |
|-----------|-------|---------|---------|-------|
| ----- | ----- | ----- | ----- | ----- |
| 12-JUL-22 | 1 | BAD_LAP | -1.3 | 58.7 |
| 12-JUL-22 | 2 | BAD_LAP | -1.6 | 59.7 |
| 12-JUL-22 | 3 | BAD_LAP | -1.3 | 60.3 |
| 12-JUL-22 | 4 | ZERO | 0 | 61.3 |
| 12-JUL-22 | 9 | BAD_LAP | -.4 | 59.6 |
| 12-JUL-22 | 10 | BAD_LAP | -.5 | 59.9 |
| 12-JUL-22 | 11 | BAD_LAP | -.1 | 60.4 |
| 12-JUL-22 | 12 | ZERO | 0 | 60.1 |

did you miss it?

```
SQL> select *  
      2  from swimming  
      3  order by sess,lap;
```

| SESS | LAP | ELA |
|-----------|-------|-------|
| ----- | ----- | ----- |
| 12-JUL-22 | 1 | 58.7 |
| 12-JUL-22 | 2 | 59.7 |
| 12-JUL-22 | 3 | 60.3 |
| 12-JUL-22 | 4 | 61.3 |
| 12-JUL-22 | 5 | 60.7 |
| 12-JUL-22 | 6 | 59.7 |
| 12-JUL-22 | 7 | 60.2 |
| 12-JUL-22 | 8 | 58.6 |
| 12-JUL-22 | 9 | 59.6 |
| 12-JUL-22 | 10 | 59.9 |
| 12-JUL-22 | 11 | 60.4 |
| 12-JUL-22 | 12 | 60.1 |

60


```

SQL> select * from swimming
2  match_recognize (
3    partition by sess order by lap
4    measures classifier() pattern, sum(delta) as run_tot
5    all rows per match
6    after match skip to next row
7    pattern (bad_lap* zero)
8    define zero as sum(delta) = 0
9  );

```

| SESS | LAP | PATTERN | RUN_TOT | ELA |
|-----------|-----|---------|---------|------|
| 12-JUL-22 | 1 | ZERO | 0 | 59.7 |
| 12-JUL-22 | 2 | BAD_LAP | -.3 | 59.7 |
| 12-JUL-22 | 3 | BAD_LAP | 0 | 60.3 |
| 12-JUL-22 | 4 | BAD_LAP | 1.3 | 61.3 |
| 12-JUL-22 | 5 | BAD_LAP | 2 | 60.7 |
| 12-JUL-22 | 6 | BAD_LAP | 1.7 | 59.7 |
| 12-JUL-22 | 7 | BAD_LAP | 1.9 | 60.2 |
| 12-JUL-22 | 8 | BAD_LAP | 1.5 | 59.7 |
| 12-JUL-22 | 9 | BAD_LAP | -.4 | 59.6 |
| 12-JUL-22 | 10 | BAD_LAP | -.5 | 59.9 |
| 12-JUL-22 | 11 | BAD_LAP | -.1 | 60.4 |
| 12-JUL-22 | 12 | ZERO | 0 | 60.1 |



lots of
Boys need feeding!



Merry Christmas

"Divide the shopping equally into 4 bags"

```
SQL> select * from shopping;
```

| ITEM | WEIGHT |
|-----------|--------|
| ----- | ----- |
| milk | 1000 |
| bread | 650 |
| dogfood | 490 |
| biscuits | 250 |
| soda | 1500 |
| gin | 2100 |
| apples | 900 |
| bananas | 1200 |
| carrots | 650 |
| steak | 550 |
| icecream | 1240 |
| butter | 450 |
| honey | 370 |
| vegemite | 540 |
| ketchup | 290 |
| eggs | 800 |
| detergent | 950 |
| deodrant | 220 |

```
SQL> select * from shopping;
```

| ITEM | WEIGHT |
|-----------|--------|
| ----- | ----- |
| milk | 1000 |
| bread | 650 |
| dogfood | 490 |
| biscuits | 250 |
| soda | 1500 |
| gin | 2100 |
| apples | 900 |
| bananas | 1200 |
| carrots | 650 |
| steak | 550 |
| icecream | 1240 |
| butter | 450 |
| honey | 370 |
| vegemite | 540 |
| ketchup | 290 |
| eggs | 800 |
| detergent | 950 |
| deodrant | 220 |



SQL can do this too !

I have 4 bags

(matching my as yet unknown rules)

use this bag if ...

first item in the bag, or

```
pattern ( (bag1|bag2|bag3|bag4)* )
```

```
define
```

```
    bag1 as count(bag1.*) = 1 or  
        sum(bag1.weight)-bag1.weight
```

```
, bag2 as count(bag2.*) = 1 or  
    sum(bag2.weight)-bag2.weight  
    <= least(sum(bag3.weight), sum(bag4.weight))
```

```
, bag3 as count(bag3.*) = 1 or  
    sum(bag3.weight)-bag3.weight  
    <= sum(bag4.weight)
```

*my bag (before this item) has
less than the other bags*


```
SQL> select *
  2  from shopping
  3  match_recognize (
  4    order by weight desc
  5    measures
  6      classifier() bag#,
  7      sum(bag1.weight) bag1,
  8      sum(bag2.weight) bag2,
  9      sum(bag3.weight) bag3,
 10      sum(bag4.weight) bag4
 11  all rows per match
 12  pattern ( (bag1|bag2|bag3|bag4)* )
 13  define
 14    bag1 as count(bag1.*) = 1 or
 15      sum(bag1.weight)-bag1.weight <=
 16        least(sum(bag2.weight),sum(bag3.weight),sum(bag4.weight))
 17  , bag2 as count(bag2.*) = 1 or
 18      sum(bag2.weight)-bag2.weight <=
 19        least(sum(bag3.weight),sum(bag4.weight))
 20  , bag3 as count(bag3.*) = 1 or
 21      sum(bag3.weight)-bag3.weight <= sum(bag4.weight)
 22  );
```

| WEIGHT | BAG# | BAG1 | BAG2 | BAG3 | BAG4 | ITEM |
|--------|------|------|------|------|------|-----------|
| 2100 | BAG1 | 2100 | | | | gin |
| 1500 | BAG2 | 2100 | 1500 | | | soda |
| 1240 | BAG3 | 2100 | 1500 | 1240 | | icecream |
| 1200 | BAG4 | 2100 | 1500 | 1240 | 1200 | bananas |
| 1000 | BAG4 | 2100 | 1500 | 1240 | 2200 | milk |
| 950 | BAG3 | 2100 | 1500 | 2190 | 2200 | detergent |
| 900 | BAG2 | 2100 | 2400 | 2190 | 2200 | apples |
| 800 | BAG1 | 2900 | 2400 | 2190 | 2200 | eggs |
| 650 | BAG3 | 2900 | 2400 | 2840 | 2200 | carrots |
| 650 | BAG4 | 2900 | 2400 | 2840 | 2850 | bread |
| 550 | BAG2 | 2900 | 2950 | 2840 | 2850 | steak |
| 540 | BAG3 | 2900 | 2950 | 3380 | 2850 | vegemite |
| 490 | BAG4 | 2900 | 2950 | 3380 | 3340 | dogfood |
| 450 | BAG1 | 3350 | 2950 | 3380 | 3340 | butter |
| 370 | BAG2 | 3350 | 3320 | 3380 | 3340 | honey |
| 290 | BAG2 | 3350 | 3610 | 3380 | 3340 | ketchup |
| 250 | BAG4 | 3350 | 3610 | 3380 | 3590 | biscuits |
| 220 | BAG1 | 3570 | 3610 | 3380 | 3590 | deodrant |

```

SQL> with portions as
2   (
3     select *
4     from shopping
5     match_recognize (
6       order by weight desc
7       measures
      ...
25  )

```

| BAG# | ITEMS | KG |
|-------|-------------------------------------|-------|
| ----- | ----- | ----- |
| BAG1 | butter,deodrant,eggs,gin | 3.57 |
| BAG2 | apples,honey,ketchup,soda,steak | 3.61 |
| BAG3 | carrots,detergent,icecream,vegemite | 3.38 |
| BAG4 | bananas,biscuits,bread,dogfood,milk | 3.59 |

*"You said the code would be **easy!**"*

"But I have 3 bags not 4 bags!"

"But I buy hardware not food!"

SQL macros

SQL ... that writes/changes SQL !


```

SQL> create or replace
  2  function pack_and_carry(p_tab dbms_tf.table_t, p_bags int)
  3      return clob sql_macro is
  4      l_sql clob;
  5      l_bag varchar2(1000);
  6      l_sum varchar2(4000);
  7      l_pattern varchar2(4000);
  8  begin
  9      for i in 1 .. p_bags loop
10          l_bag := l_bag || 'bag' || i || ' ';
11          l_sum := l_sum || replace('sum(bag@.weight) bag@,', '@', i) || chr(10);
12          if i < p_bags then
13              if i < p_bags-1 then
14                  l_pattern := l_pattern ||
15                      replace(',bag@ as count(bag@.*)=1 or sum(bag@.weight)-bag@.weight <= least(', '@', i);
16              else
17                  l_pattern := l_pattern ||
18                      replace(',bag@ as count(bag@.*)=1 or sum(bag@.weight)-bag@.weight <= ', '@', i);
19              end if;
20              for j in i+1 .. p_bags loop
21                  l_pattern := l_pattern || replace('sum(bag@.weight)', '@', j);
22              end loop;
23              l_pattern := rtrim(l_pattern, ',') || ' ' || chr(10);
24          end if;
25      end loop;

```

```

SQL> create or replace
  2  function pack_and_carry(p_tab dbms_tf.table_t, p_bags int)
  3      return clob sql_macro is
  4      l_sql clob;
  5      l_bag varchar2(1000);
  6      l_sum varchar2(4000);
  7      l_pattern varchar2(4000);
  8  begin
  9      for i in 1 .. p_bags loop
10          l_bag := l_bag || 'bag' || i || ' ';
11          l_sum := l_sum || replace('sum(bag@.weight) bag@,', '@', i) || chr(10);
12          if i < p_bags then
13              if i < p_bags-1 then
14                  l_pattern := l_pattern ||
15                      replace(',bag@ as count(bag@.*)=1 or sum(bag@.weight)-bag@.weight <= least(', '@', i);
16              else
17                  l_pattern := l_pattern ||
18                      replace(',bag@ as count(bag@.*)=1 or sum(bag@.weight)-bag@.weight <= ', '@', i);
19              end if;
20              for j in i+1 .. p_bags loop
21                  l_pattern := l_pattern || replace('sum(bag@.weight)', '@', j);
22              end loop;
23              l_pattern := rtrim(l_pattern, ',') || ' ' || chr(10);
24          end if;
25      end loop;

```

```
26     l_sql := q'{
27 select * from p_tab
28 match_recognize (
29     order by weight desc
30     measures
31         classifier() bag#,
32         ~~~
33 all rows per match
34 pattern ( (###)* )
35 define $$$}';
36
37     l_sql := replace(l_sql, '###', rtrim(l_bag, '|'));
38     l_sql := replace(l_sql, '~~~', rtrim(l_sum, ', ' || chr(10)));
39     l_sql := replace(l_sql, '$$$', ltrim(l_pattern, ', '));
40     return l_sql;
41 end;
42 /
```

Function created.

```
SQL> select
2   bag#,
3   listagg(item,',') within group ( order by item ) as items,
4   sum(weight)/1000 kg
5 from pack_and_carry(shopping,4)
6 group by bag#;
```

```
BAG#
-----
BAG1
BAG2
BAG3
BAG4

SQL> select
2   bag#,
3   listagg(item,',') within group ( order by item ) as items,
```

```
4   sum(weight)/1000 kg
5 from
6 gro
```

```
SQL> select
2   bag#,
3   listagg(item,',') within group ( order by item ) as items,
4   sum(weight)/1000 kg
5 from pack_and_carry(hardware,5)
6 group by bag#;
```

| BAG# | ITEMS | KG |
|------|--|------|
| BAG1 | chainsaw | 12 |
| BAG2 | powerwasher | 9 |
| BAG3 | hacksaw,tap,transformer,vice | 8.95 |
| BAG4 | bucket,hammer,rake,screwdriver,shovel,wood | 9 |
| BAG5 | broom,chisel,drill,padlock,paint,sink | 9 |



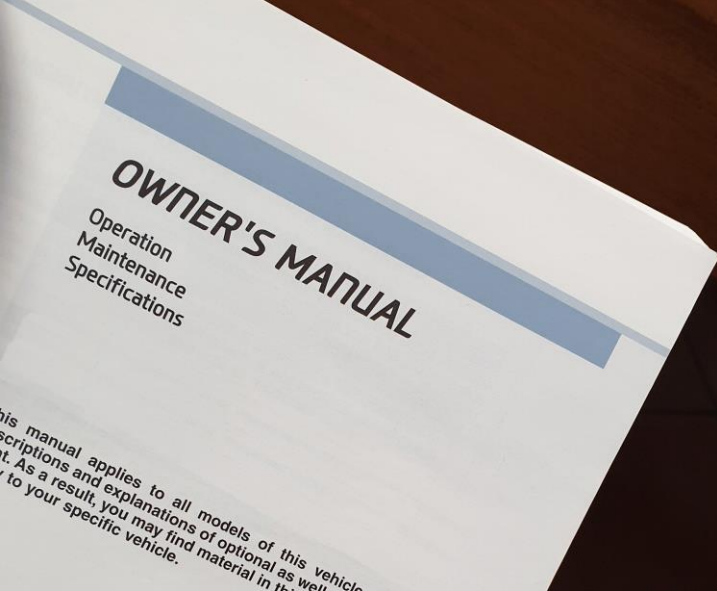


weekend is over ☹️

"Refuel on the discount days \$\$\$"

```
SQL> select *  
      2  from car_fuel;
```

| DTE | PCTFULL | LITRES |
|-----------|---------|--------|
| ----- | ----- | ----- |
| 01-AUG-21 | 0 | 45 |
| 09-AUG-21 | 20 | 37 |
| 13-AUG-21 | 60 | 22 |
| 21-AUG-21 | 20 | 20 |
| 26-AUG-21 | 5 | 60 |
| 03-SEP-21 | 15 | 32 |
| 11-SEP-21 | 80 | 15 |
| 15-SEP-21 | 60 | 20 |



"Clean fuel"

AutoSave Off Book1 - Excel

FileHomeInsertDrawPage LayoutFormulasDataReviewViewHelp

Cut

Copy

Format Painter

Clipboard

Calibri

11

A

A

B

I

U



```
SQL> select *  
      2  from car_fuel;
```

| DTE | PCTFULL | LITRES | DIRT |
|-----------|---------|--------|-------|
| ----- | ----- | ----- | ----- |
| 01-AUG-21 | 0 | 45 | 2.25 |
| 09-AUG-21 | 20 | 37 | 2.6 |
| 13-AUG-21 | 60 | 22 | |
| 21-AUG-21 | 20 | 20 | |
| 26-AUG-21 | 5 | 60 | |
| 03-SEP-21 | 15 | 32 | |
| 11-SEP-21 | 80 | 15 | |
| 15-SEP-21 | 60 | 20 | |



recursion in SQL

```
SQL> select
      2      car_fuel.*,
      3      row_number() over (order by dte ) as seq
      4  from car_fuel;
```

| DTE | PCTFULL | LITRES |
|-----------|---------|--------|
| ----- | ----- | ----- |
| 01-AUG-21 | 0 | 45 |
| 09-AUG-21 | 20 | 37 |
| 13-AUG-21 | 60 | 22 |
| 21-AUG-21 | 20 | 20 |
| 26-AUG-21 | 5 | 60 |
| 03-SEP-21 | 15 | 32 |
| 11-SEP-21 | 80 | 15 |
| 15-SEP-21 | 60 | 20 |

```
SQL> with t as
  2   ( select
  3       car_fuel.*,
  4       row_number() over (order by dte ) as seq
  5   from car_fuel
  6   ),
  7   results(dte, pctfull, litres, dirt ,seq) as
  8   (
  9       select dte, pctfull, litres, litres*0.05 dirt, seq
10   from t
11   where seq = 1
12   union all
13   select t.dte, t.pctfull, t.litres,
14       results.dirt * t.pctfull/60 + t.litres*0.05 , t.seq
15   from t, results
16   where t.seq - 1 = results.seq
17   )
18   select * from results
19   order by seq;
```

```

SQL> with t as
      2  ( select
          ...
      17  )
      18  select * from results
      19  order by seq;

```

| DTE | PCTFULL | LITRES | DIRT |
|-----------|---------|--------|------------|
| ----- | ----- | ----- | ----- |
| 01-AUG-21 | 0 | 45 | 2.25 |
| 09-AUG-21 | 20 | 37 | 2.6 |
| 13-AUG-21 | 60 | 22 | 3.7 |
| 21-AUG-21 | 20 | 20 | 2.23333333 |
| 26-AUG-21 | 5 | 60 | 3.18611111 |
| 03-SEP-21 | 15 | 32 | 2.39652778 |
| 11-SEP-21 | 80 | 15 | 3.94537037 |
| 15-SEP-21 | 60 | 20 | 4.94537037 |

"But I like Excel formulas" 😞

cell formula expressions in SQL


```

SQL> select dte, pctfull, litres, dirt
2  from car_fuel
3  model
4  dimension by ( row_number() over(order by dte) seq )
5  measures( dte, pctfull, litres, 0 dirt )
6  rules(
7      dirt[any] order by seq =
8      presentnnv(dirt[cv()-1], dirt[cv()-1], 0) *
9      pctfull[cv()]/60 +
10     litres[cv()]*0.05
11     );

```

| | A | B | C | D | E | F |
|---|-----------|---------|--------|-------------|--|---|
| | DATE | PCTFULL | LITRES | DIRT | | |
| 2 | 1-Aug-21 | 0 | 45 | 2.25 | =LITRES * 0.05 | |
| 3 | 9-Aug-21 | 20 | 37 | 2.5 | =PREV(DIRT)*PCTFULL/60 + LITRES * 0.05 | |
| 4 | 13-Aug-21 | 60 | 22 | 3.7 | | |
| 5 | 21-Aug-21 | 20 | 20 | 2.233333333 | | |
| 6 | 26-Aug-21 | 5 | 60 | 3.186111111 | | |

```
SQL> select dte, pctfull, litres, dirt
2  from car_fuel
3  model
4  dimension by ( row_number() over(order by dte) seq )
5  measures( 0 dirt, pctfull, litres, dte )
6  rules(
7      dirt[any] order by seq =
```

| | pre | DTE | PCTFULL | LITRES | DIRT |
|----|-----|-----------|---------|--------|-------------|
| 9 | pct | ----- | ----- | ----- | ----- |
| 10 | lit | 01-AUG-21 | 0 | 45 | 2.25 |
| 11 |) ; | 09-AUG-21 | 20 | 37 | 2.6 |
| | | 13-AUG-21 | 60 | 22 | 3.7 |
| | | 21-AUG-21 | 20 | 20 | 2.233333333 |
| | | 26-AUG-21 | 5 | 60 | 3.186111111 |
| | | 03-SEP-21 | 15 | 32 | 2.39652778 |
| | | 11-SEP-21 | 80 | 15 | 3.94537037 |
| | | 15-SEP-21 | 60 | 20 | 4.94537037 |



wrap up

Intelligent SQL

| | |
|----------------------------|----------|
| • grouping sets | 24 years |
| • row_number() + analytics | 24 years |
| • hypothetical analytics | 24 years |
| • partitioned outer join | 21 years |
| • WITH clause | 21 years |
| • KEEP clause | 21 years |
| • MODEL clause | 19 years |
| • recursive WITH | 13 years |
| • MATCH_RECOGNIZE | 8 years |
| • SQL Macros | 3 years |

robust, cool, powerful

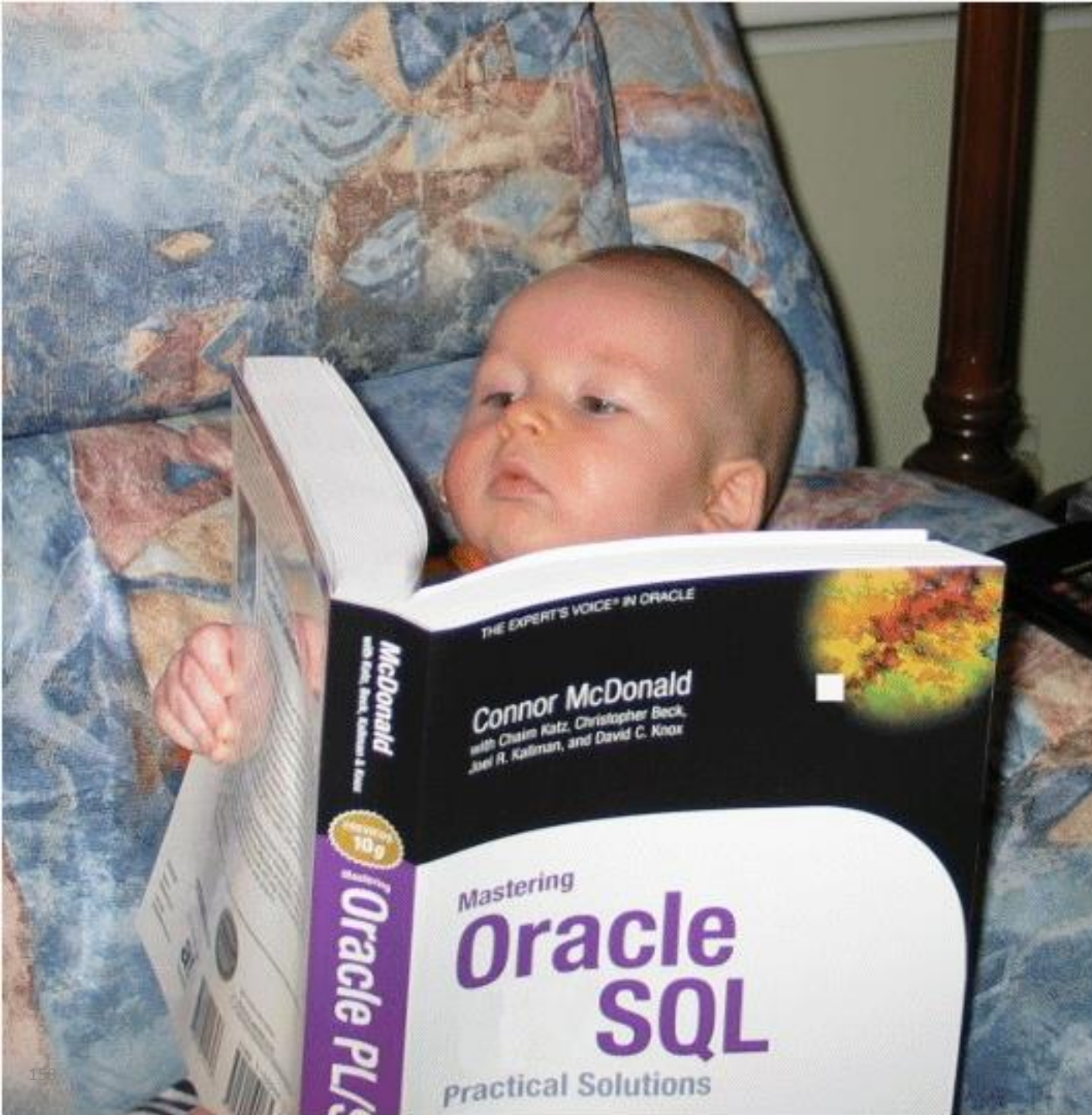
less code



faster, scalable apps

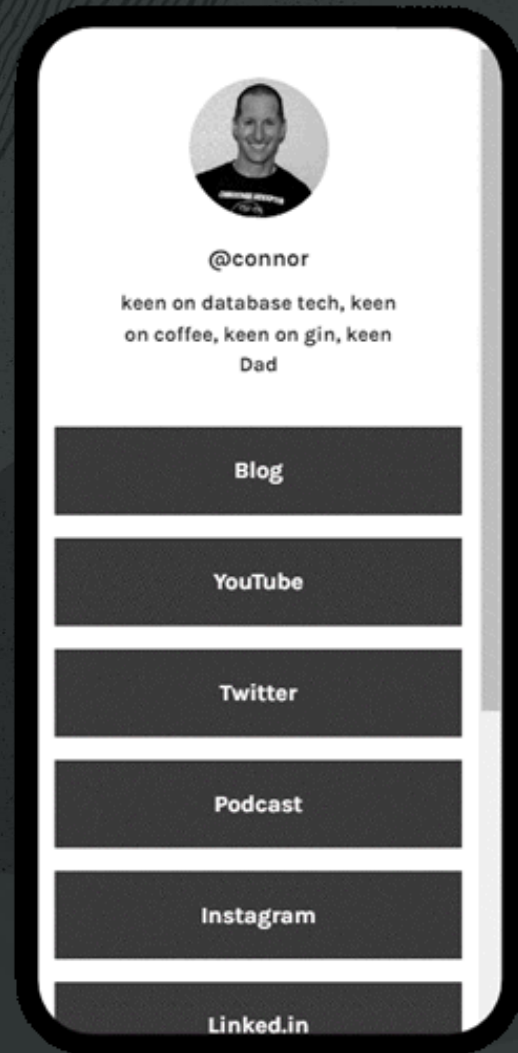


never too early to start



Son #1 😊

Stay in touch!



@connor_mc_d

<https://linktr.ee/connor>

