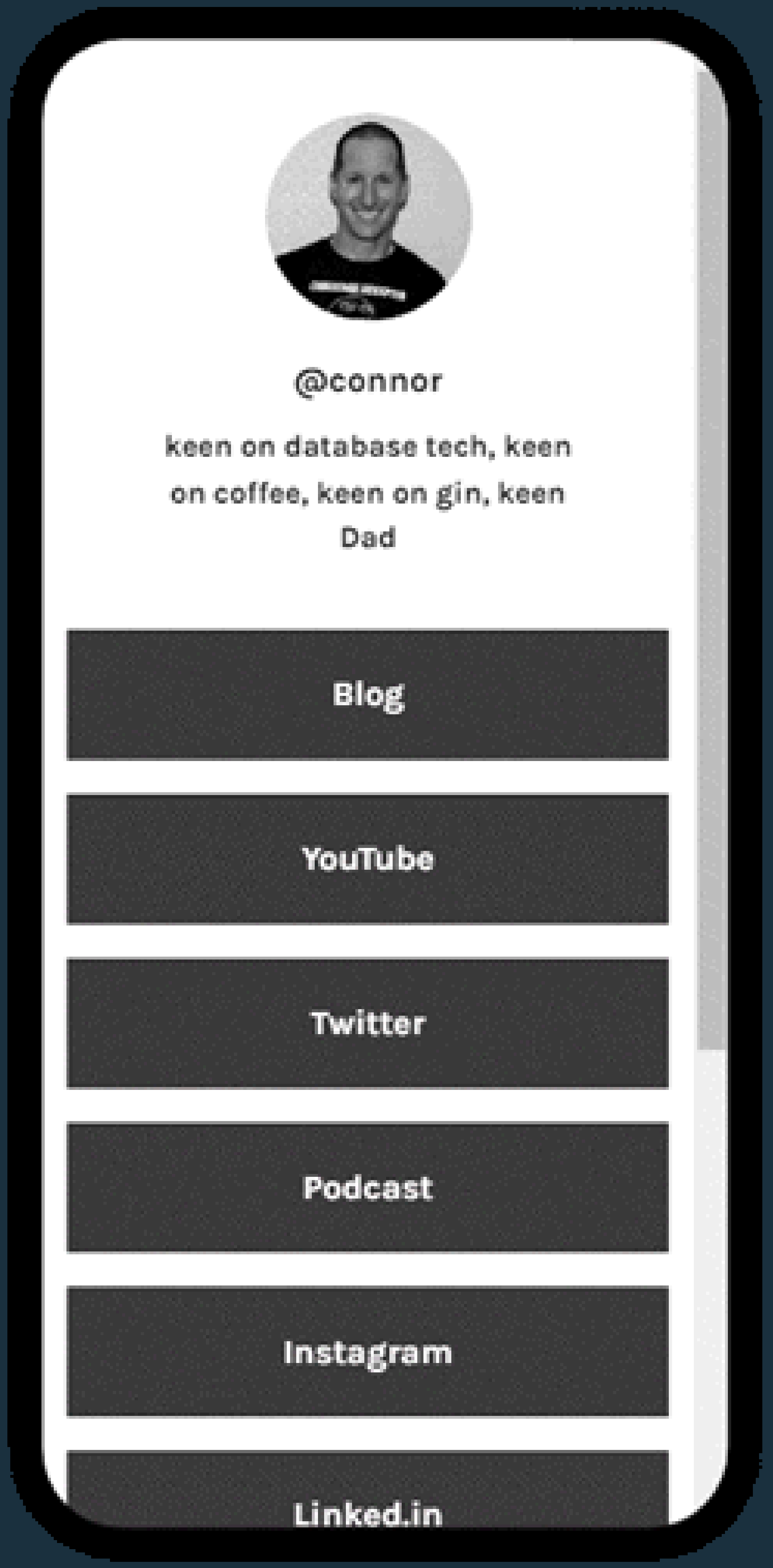
ORACLE
CloudWorld

Futuristic SQL

Making the Impossible Possible

Connor McDonald
Database Advocate

Getting in touch is easy…



@connor

keen on database tech, keen on coffee, keen on gin, keen Dad

Blog

YouTube

Twitter

Podcast

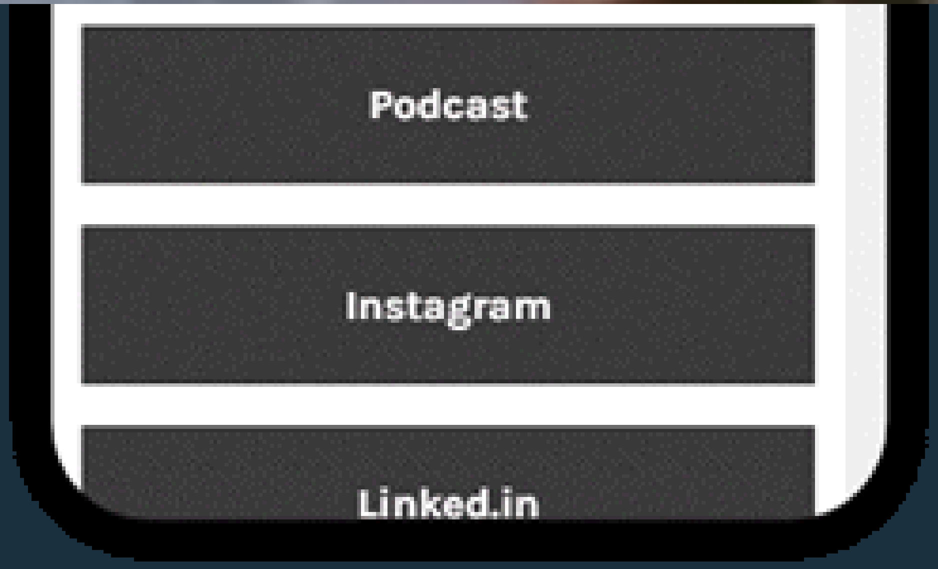
Instagram

Linked.in

@connor_mc_d

https://linktr.ee/connor

Getting in touch is easy...



Podcast

Instagram

Linked.in

@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)
```

```
     or    z = substr(s,mod(ind-1,9)-8+lp*9,1)
     or    z = substr(s,mod(trunc((ind-1)/3),3)*3
                        +trunc((ind-1)/27)*27+lp
                        +trunc((lp-1)/3)*6,1)
   )
),
result as (
  select s
  from x
  where ind = 0 )
select
  regexp_replace(substr(s,(idx-1)*9+1,9),
             '(...)(...)(...)',
             '\1|\2|\3')||
  case when mod(idx,3)=0 then chr(10)||rpad('-',11,'-') end  soln
from result,
    ( select level idx
      from dual
      connect by level <= 9 )
```

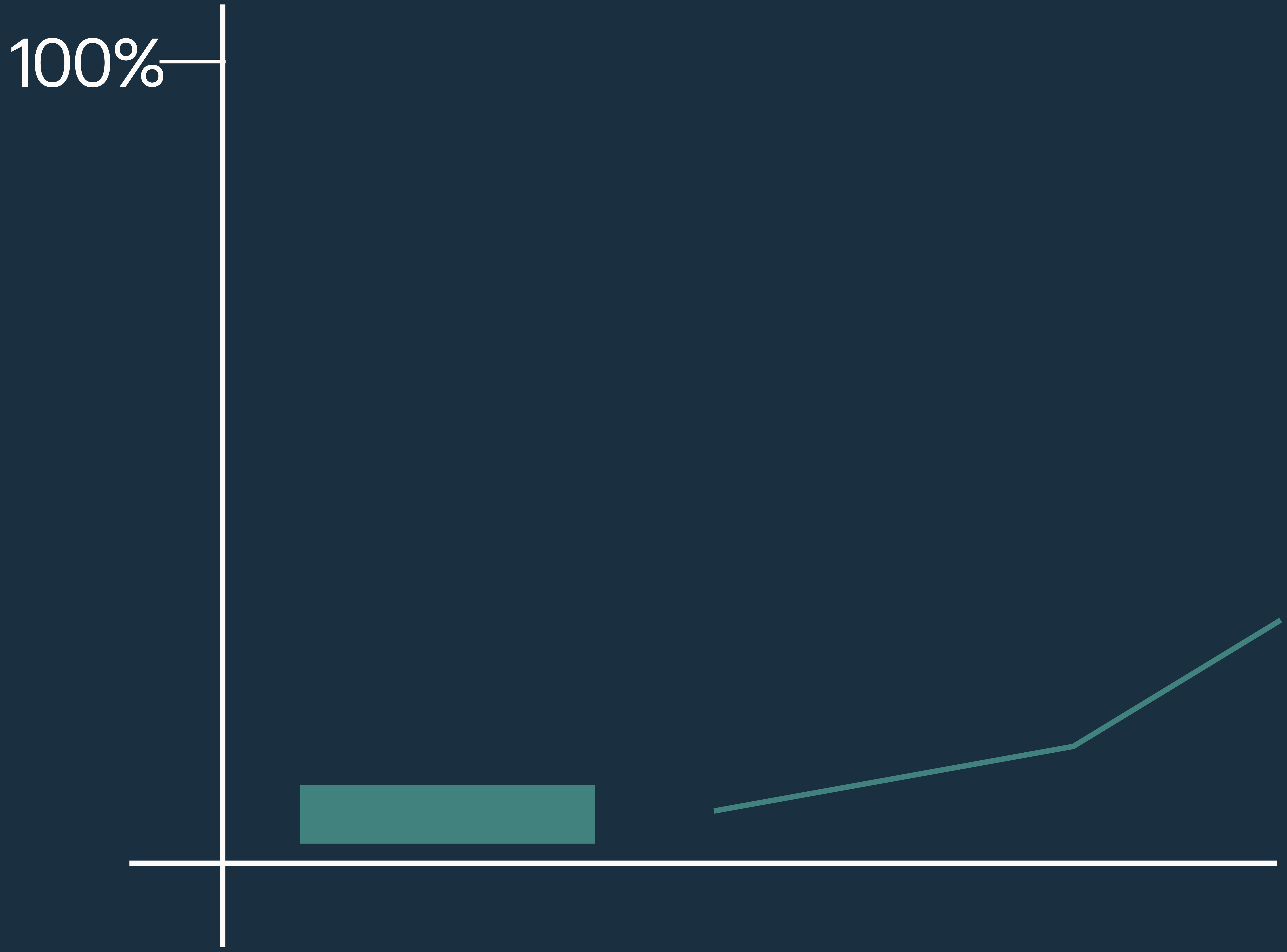
Ack: Anton Scheffer,
technology.amis.nl

```
SQL> variable board varchar2(1000)
SQL> begin :board :=
  2
  3
  4
  5
  6
  7
  8
  9
 10
 11
 12
 13
 14  end;
```

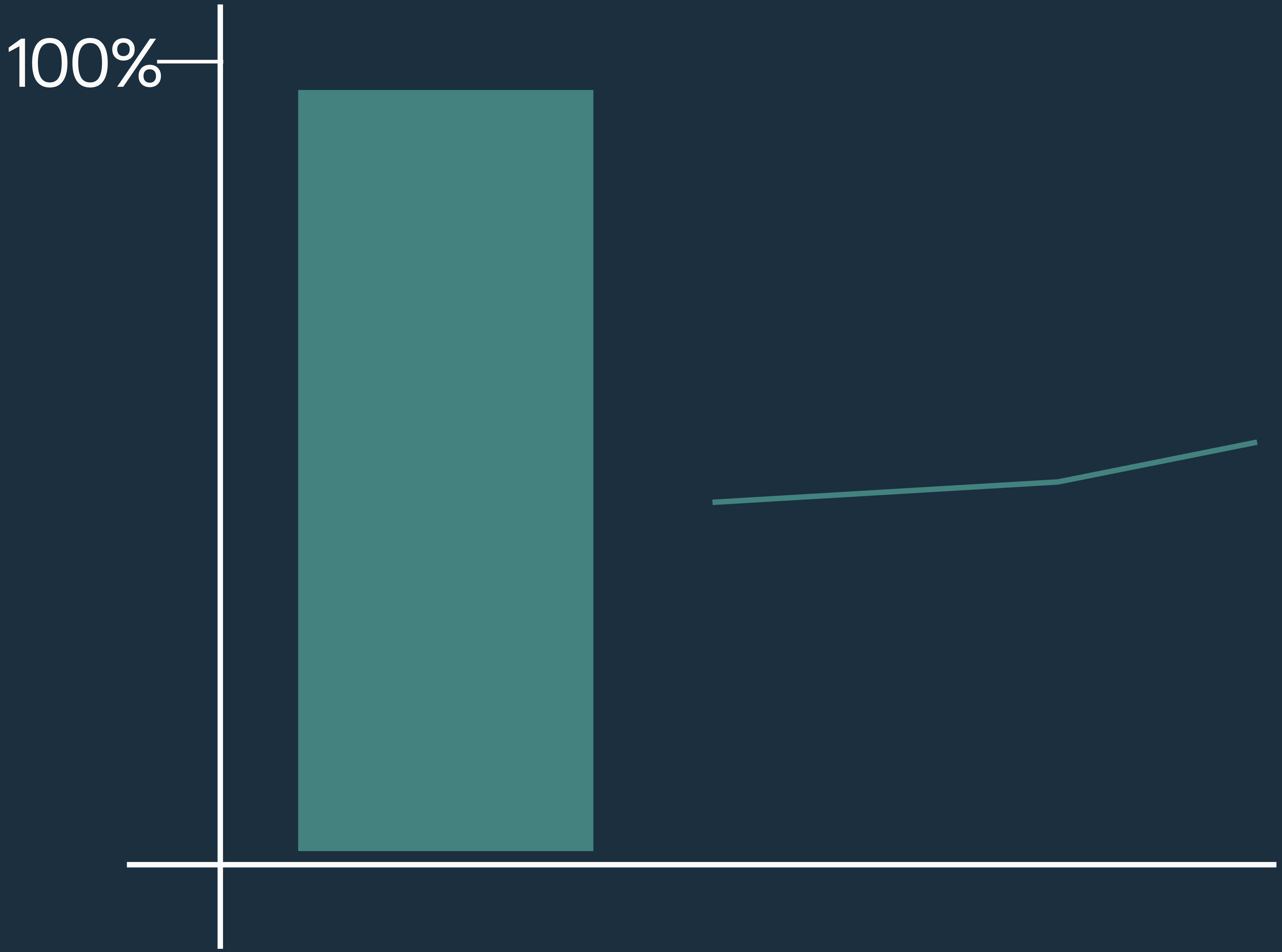
5	3		7					
6			1	9	5			
	9	8					6	
8				6	1			3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

```
SQL> variable board varchar2(1000)
SQL> begin :board :=
  2  '53.|.7.|...
  3   6..|195|...
  4   .98|...|.6.
  5   -----------
  6   8..|.61|..3
  7   4..|8.3|..1
  8   7..|.2.|..6
  9   -----------
 10   .6.|...|28.
 11   ...|419|..5
 12   ...|.8.|.79
 13  ';
 14  end;
```

```
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
-----------
```



% of developers that will need to solve Sudoku as part of their job



% of developers that need to get real stuff done

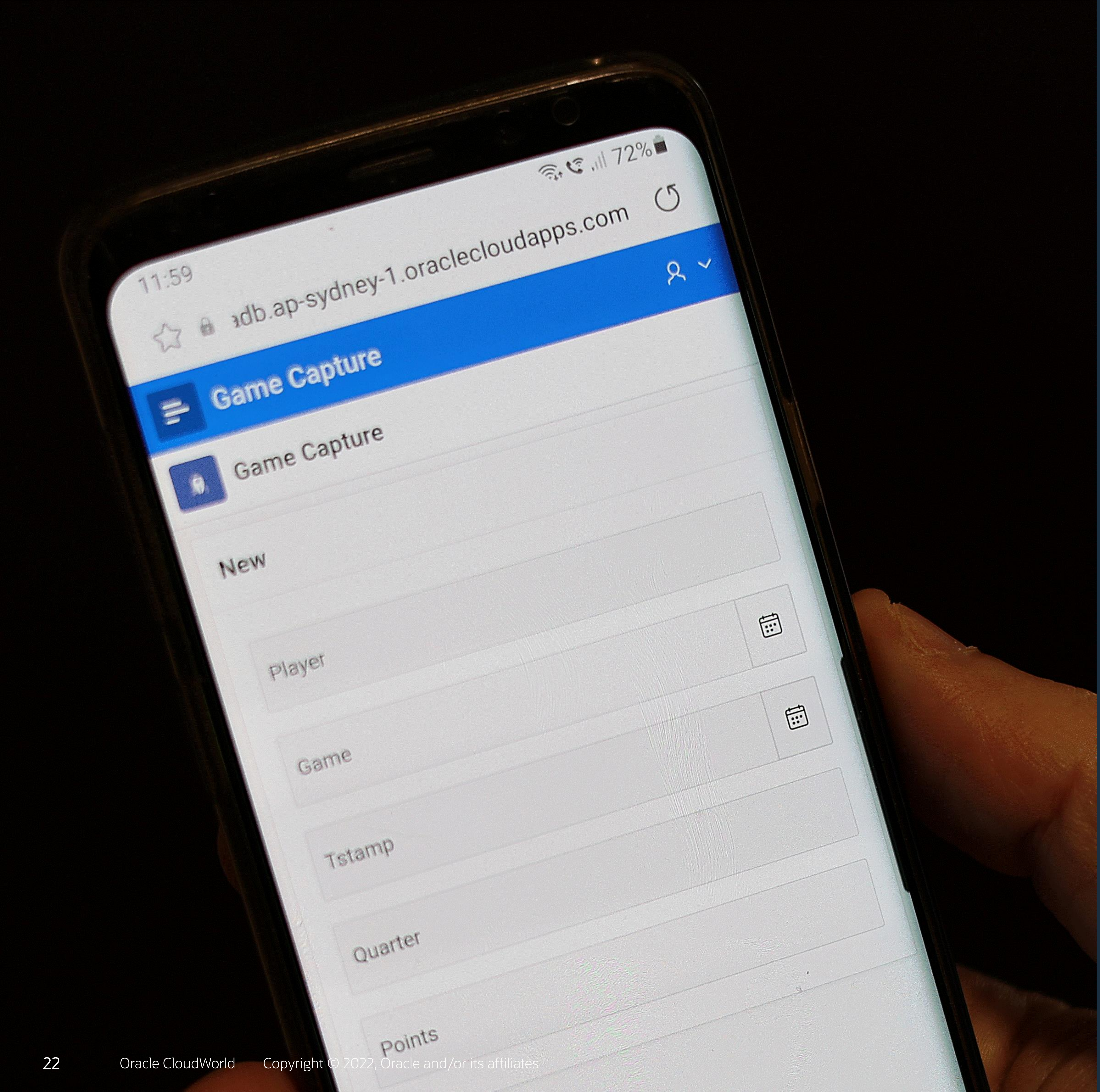
real stuff

My typical weekend

My typical weekend

… SQL edition

Son #1



11:59 72%

adb.ap-sydney-1.oraclecloudapps.com

Game Capture

Game Capture

New

Player

Game

Tstamp

Quarter

Points

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 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 12:33:05 Max                 3
         1 12:33:08 Campbell            1
         1 12:33:22 Campbell            1
...
```

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

   QUART
--------

...
```

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

   QUARTER SUM(POINTS)
---------- -----------
         1          43
         2          19
         3          33
         4          24
```

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

   QUART
--------

...
```

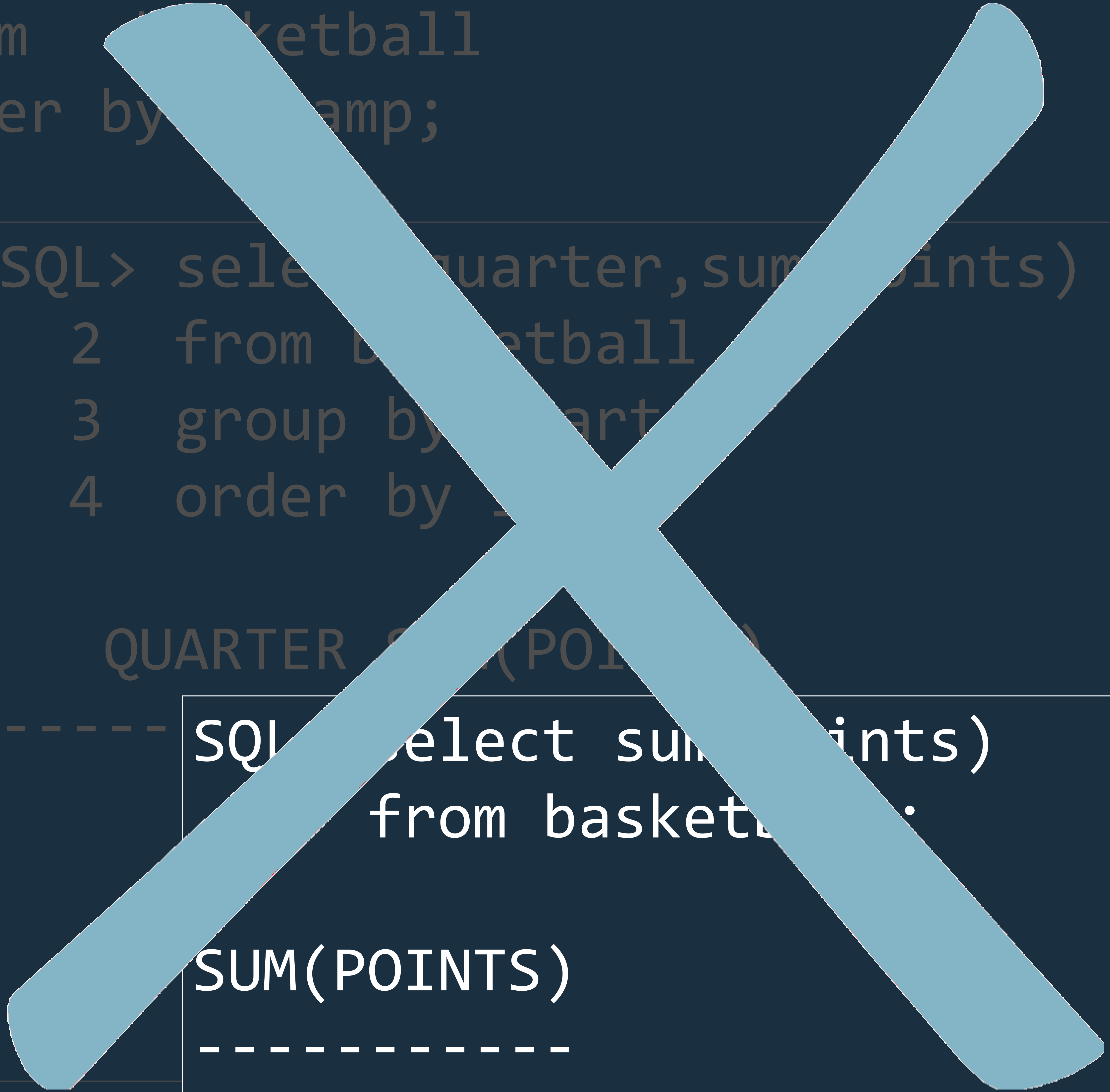
```
SQL> select quarter,sum(points)
  2  from basketball
  3  group by quarter
  4  order by 1;

   QUARTER SUM(POINTS)
-----
```

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

SUM(POINTS)
-----------
        119
```

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

   QUART
--------

...
```

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

   QUARTER SUM(POINTS)
 -----
```

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

SUM(POINTS)
-----------
        119
```



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

   QUART
--------

...
```

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

   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 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
...
```

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

   QUARTER
----------
         1
         1
         1
         1
         1
         1
         1
         1
...
```

```
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          33
         4          24
                   119
   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          33
         4          24
                   119
```

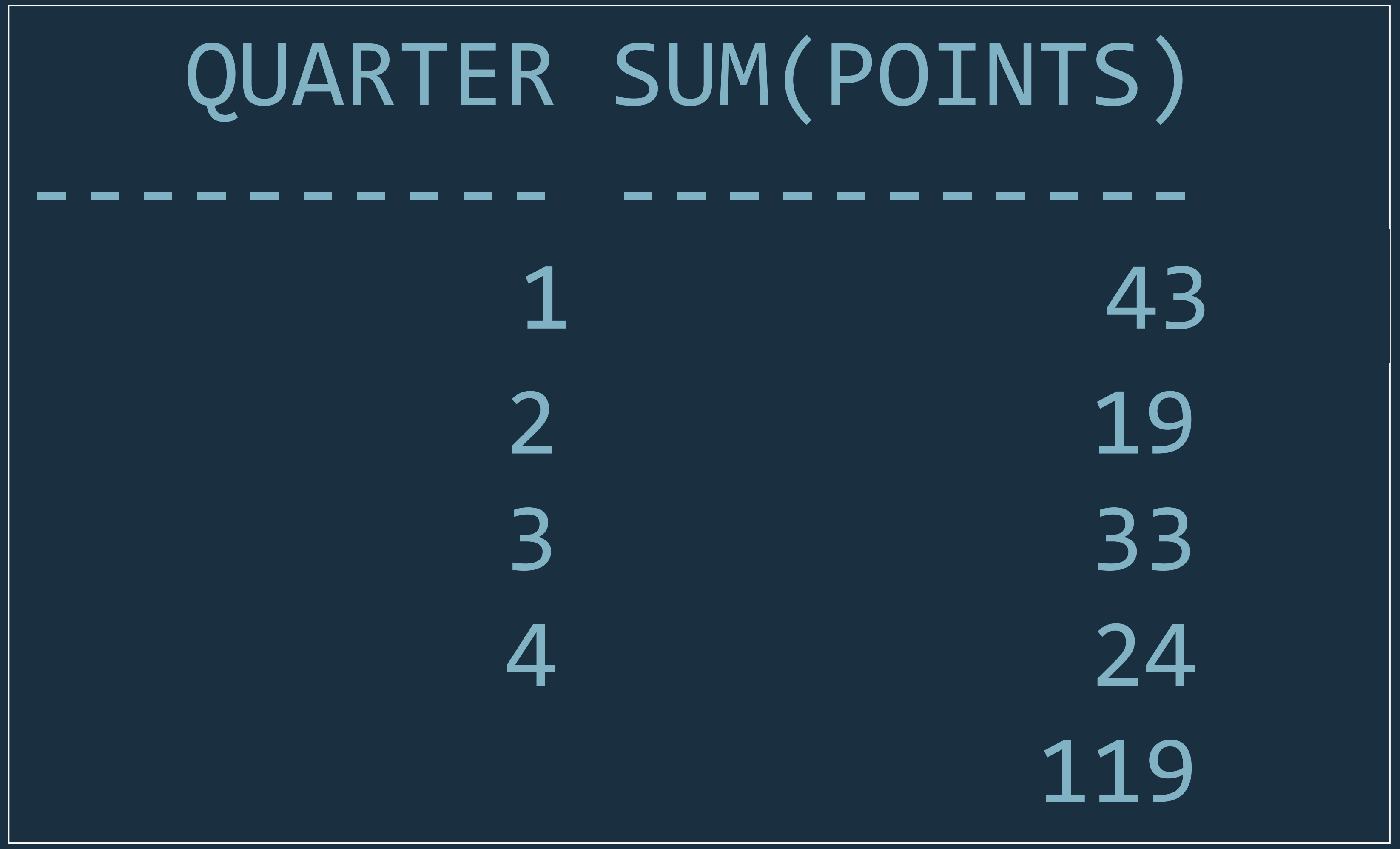
```
1 Campbell            1
```

```
   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          33
         4          24
                   119
```

```
         1 Campbell            1
         1 Robbie              3
```

```
   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          33
         4          24
                   119
```

```
         1 Campbell            1
         1 Robbie              3
         1 Zack                2
```

```
   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           33
         4           24
                    119
```

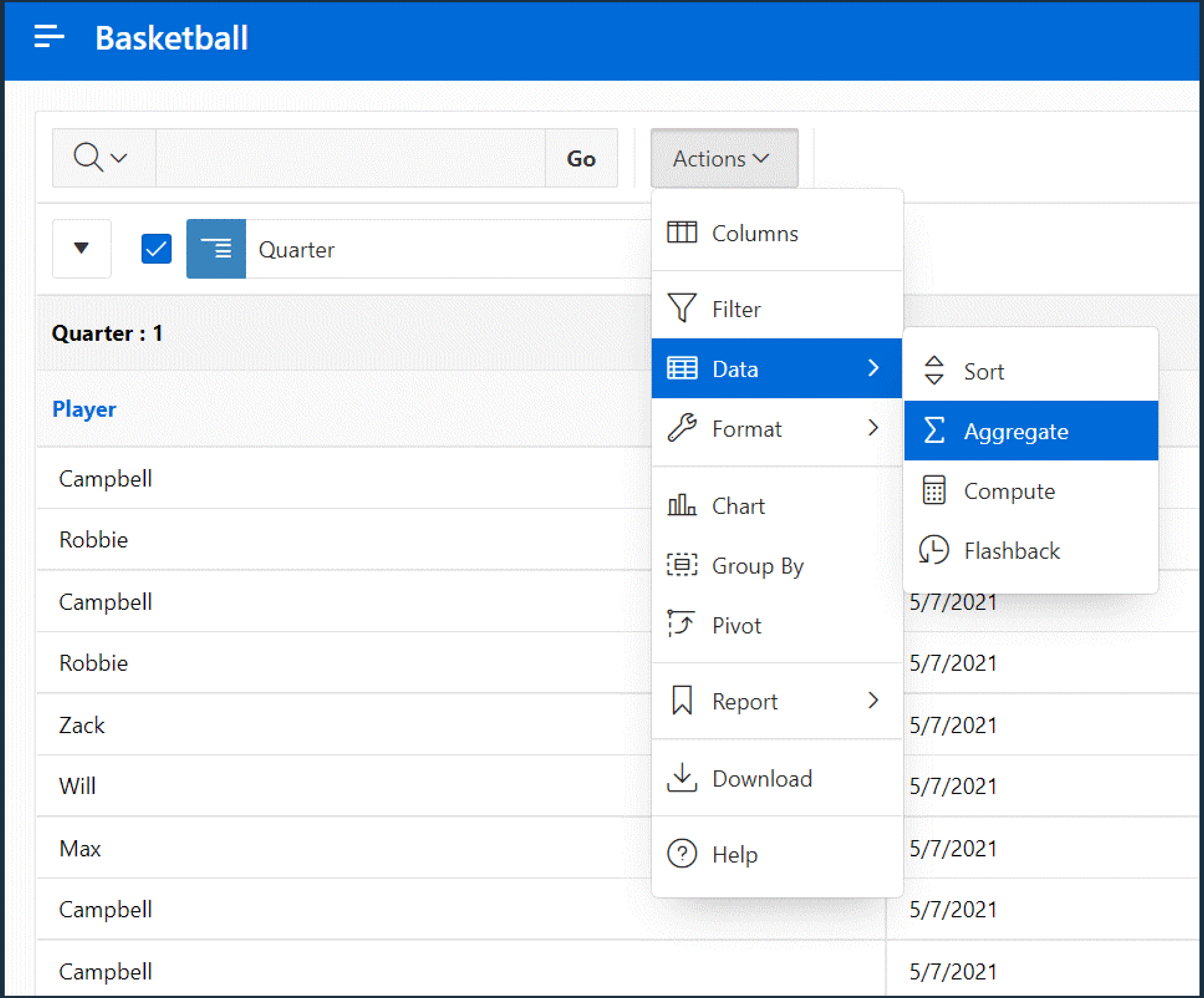
```
         1 Campbell            1
         1 Robbie              3
         1 Zack                2
                  1          43
```

```
   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          33
         4          24
                   119
```

```
1 Campbell            1
1 Robbie              3
1 Zack                2
        1        43
2 Max                 3
```

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



Quarter : 1

Player	
Campbell	
Robbie	
Campbell	5/7/2021
Robbie	5/7/2021
Zack	5/7/2021
Will	5/7/2021
Max	5/7/2021
Campbell	5/7/2021
Campbell	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         nvl2(rownum,max(player),null) player,
  3         nvl2(rownum,max(tstamp),null) tstamp,
  4         sum(points)
  5  from   basketball
  6  group by rollup(quarter,rownum)
  7  order by quarter,tstamp;

   QUARTER PLAYER     TSTAMP   SUM(POINTS)
---------- ---------- -------- -----------
         1 Campbell   12:30:12           1
         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
```

```
SQL> select quarter,
  2         nvl2(rownum,max(player),null) player,
  3         nvl2(rownum,max(tstamp),null) tstamp,
  4         sum(points)
  5  from   basketball
  6  group by rollup(quarter,rownum)
  7  order by quarter,tstamp;

   QUARTER PLAYER     TSTAMP   SUM(POINTS)
---------- ---------- -------- -----------
         1 Campbell   12:30:12           1
         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
```

```
SQL> select quarter,
  2         nvl2(rownum,max(player),null) player,
  3         nvl2(rownum,max(tstamp),null) tstamp,
  4         sum(points)
  5  from   basketball
  6  group by rollup(quarter,rownum)
  7  order by quarter,tstamp;

   QUARTER PLAYER     TSTAMP   SUM(POINTS)
---------- ---------- -------- -----------
         1 Campbell   12:30:12           1
         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
```

```
SQL> select quarter,
  2         nvl2(rownum,max(player),null) player,
  3         nvl2(rownum,max(tstamp),null) tstamp,
  4         sum(points)
  5  from   basketball
  6  group by rollup(quarter,rownum)
  7  order by quarter,tstamp;

   QUARTER PLAYER     TSTAMP   SUM(POINTS)
---------- ---------- -------- -----------
         1 Campbell   12:30:12           1
         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
```

```
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
```

```
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
```

```
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
```

```
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
```

```
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
```

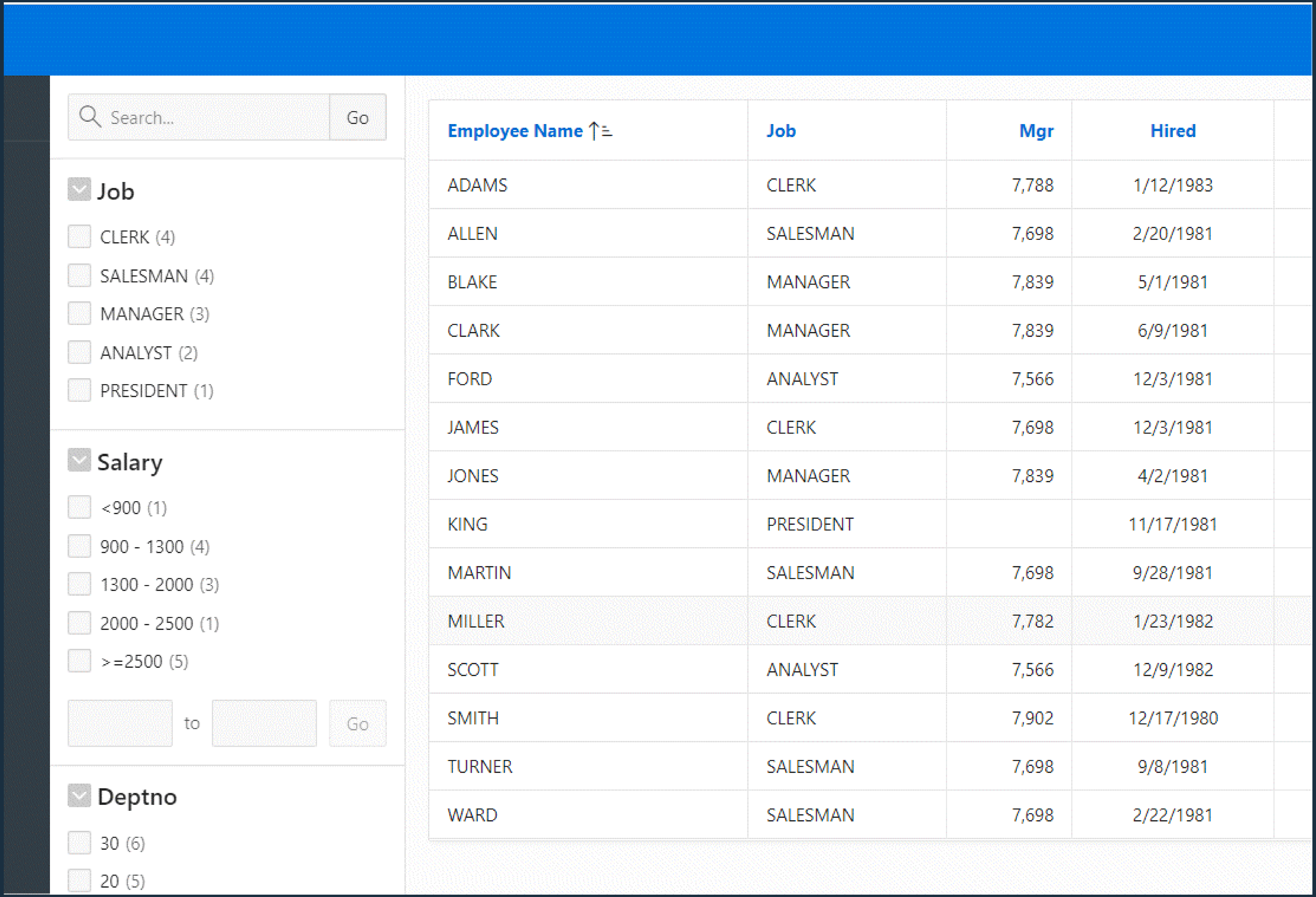
totally customisable

```
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
```



Job

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

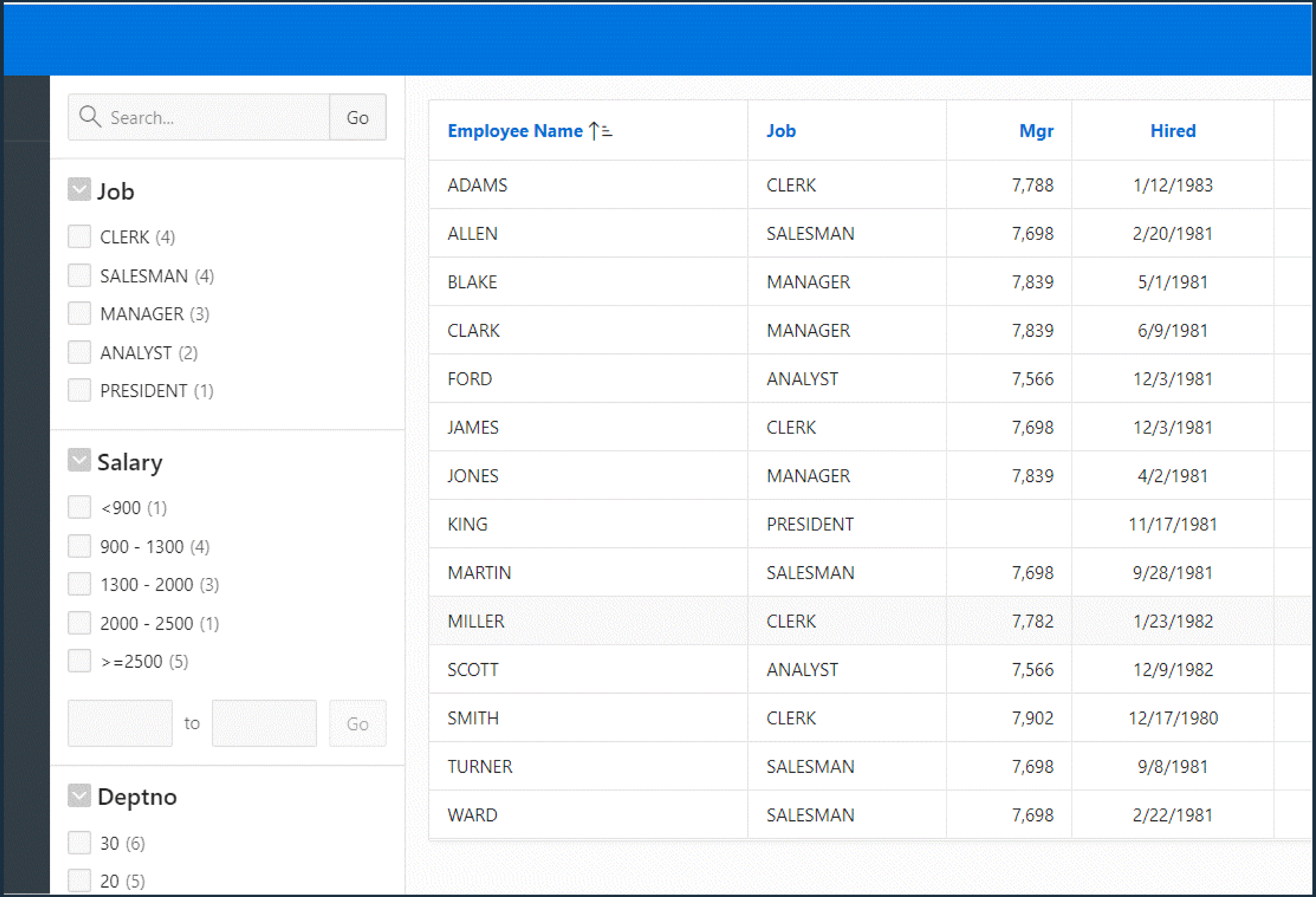
Salary

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

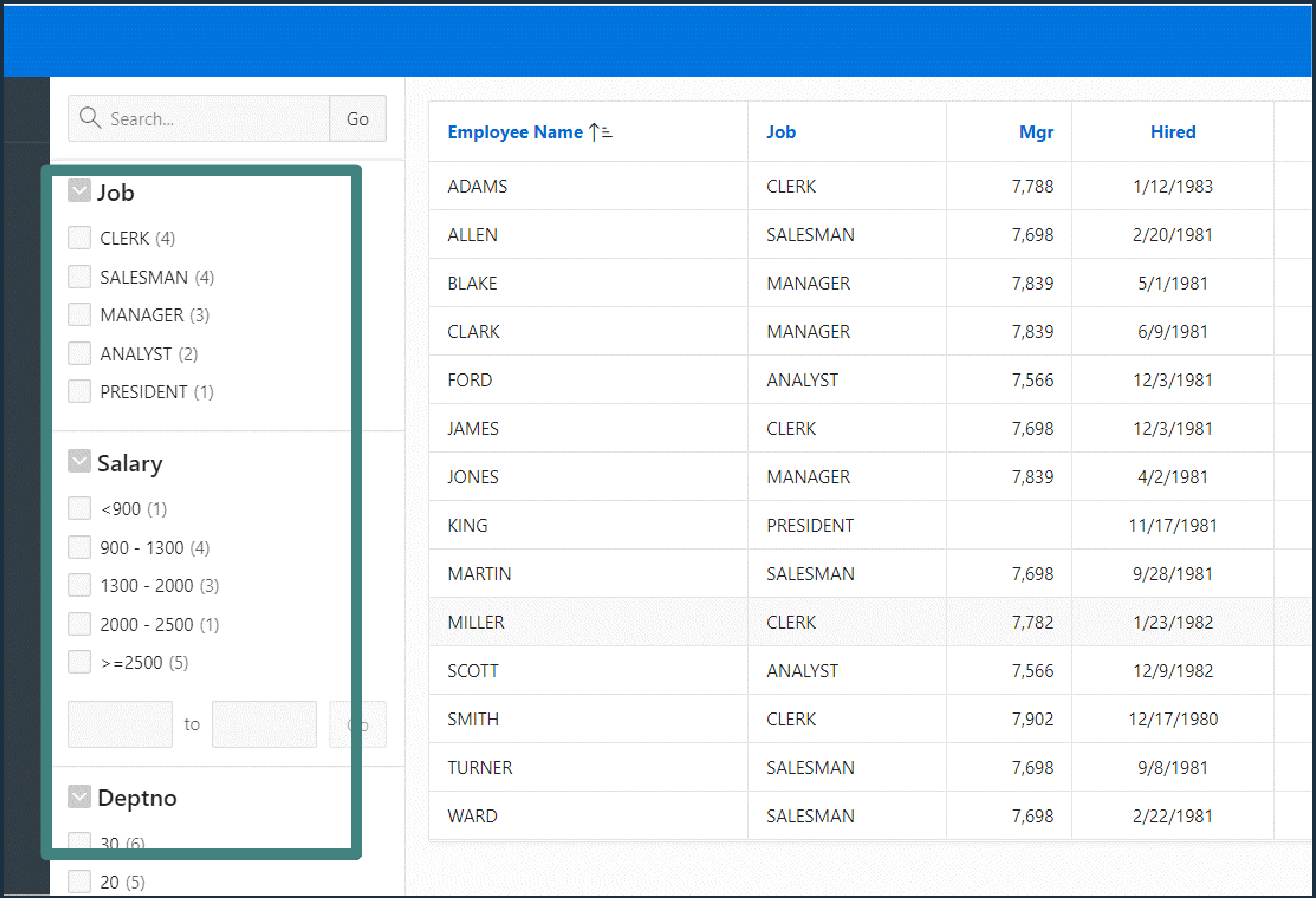
Deptno

- □ 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



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



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)

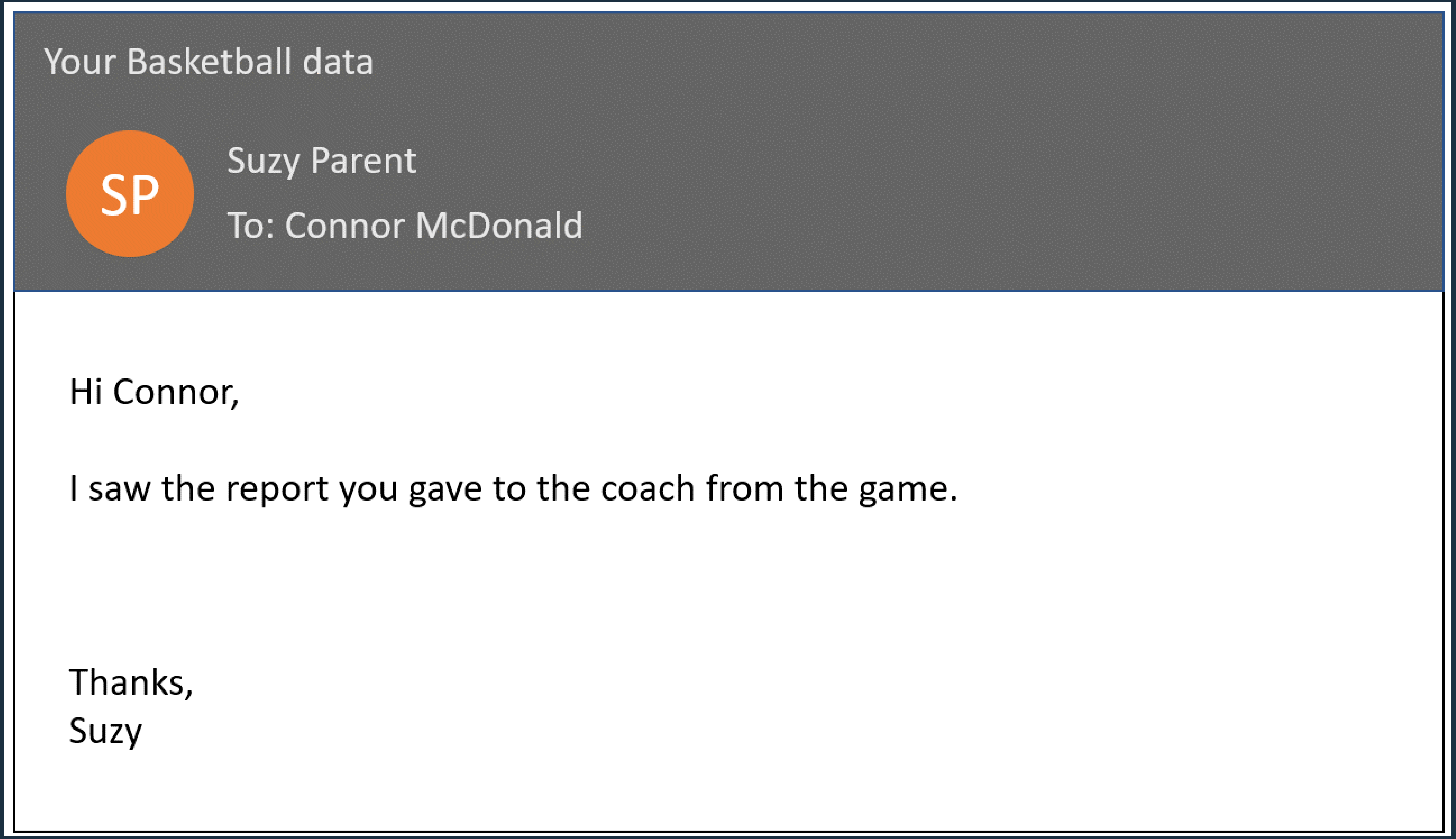
to Go

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

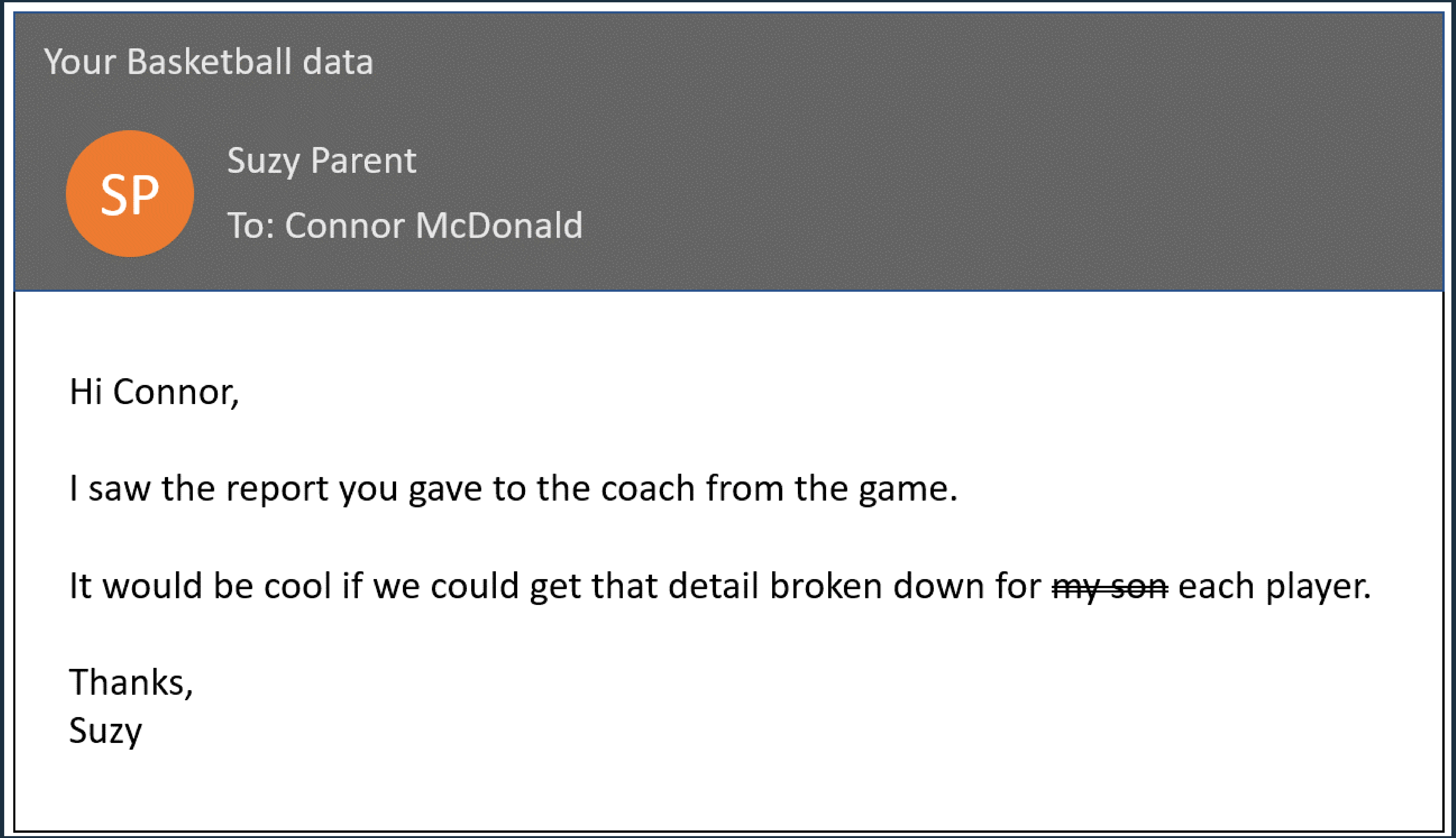
Suzy Parent

To: Connor McDonald

Hi Connor,

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

Thanks,
Suzy



Your Basketball data

Suzy Parent

To: Connor McDonald

Hi Connor,

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

It would be cool if we could get that detail broken down for ~~my son~~ each player.

Thanks,
Suzy

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

       QTR
----------
         1
         2
         3
         4
```

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

       QTR
----------
         1
         2
         3
         4
```

```
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
```

```
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)
     from
       ( select quarter, player, sum(points) pts
         from basketball
         group by quarter, player ) b
     partition by (b.player)
     right outer join
       ( select rownum qtr from dual connect by level <= 4 ) q
     on ( q.qtr = b.quarter )
     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
```

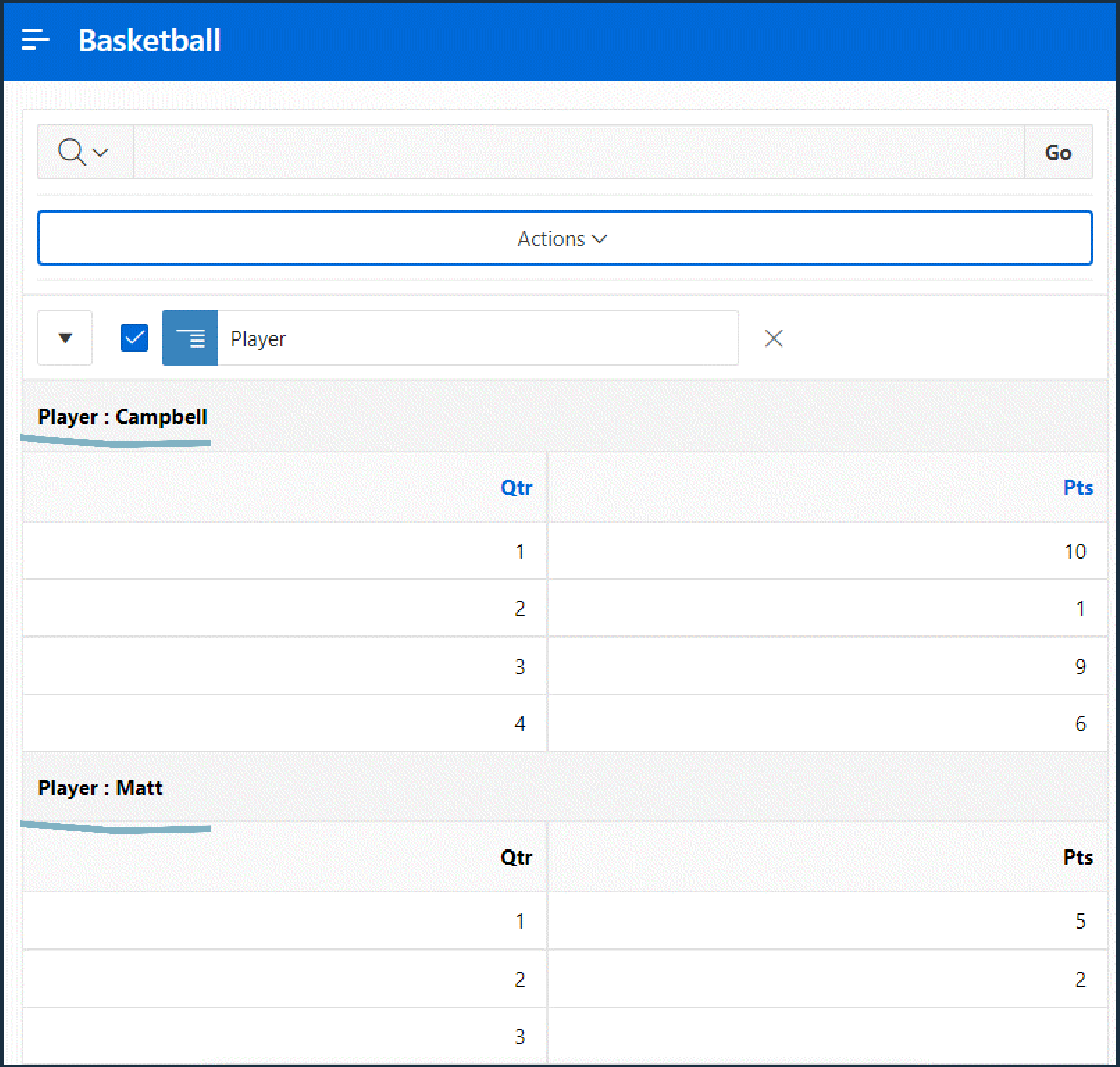
```
SQL> select qtr, player, nvl(pts,0)
  2  from
  3    ( select quarter, player, sum(points) pts
  4      from basketball
  5      group by quarter, player ) b
  6  partition by (b.player)
  7  right outer join
  8    ( select rownum qtr from dual connect by level <= 4 ) q
  9  on ( q.qtr = b.quarter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

```
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
```

```
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
```

```
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
```

```
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
```

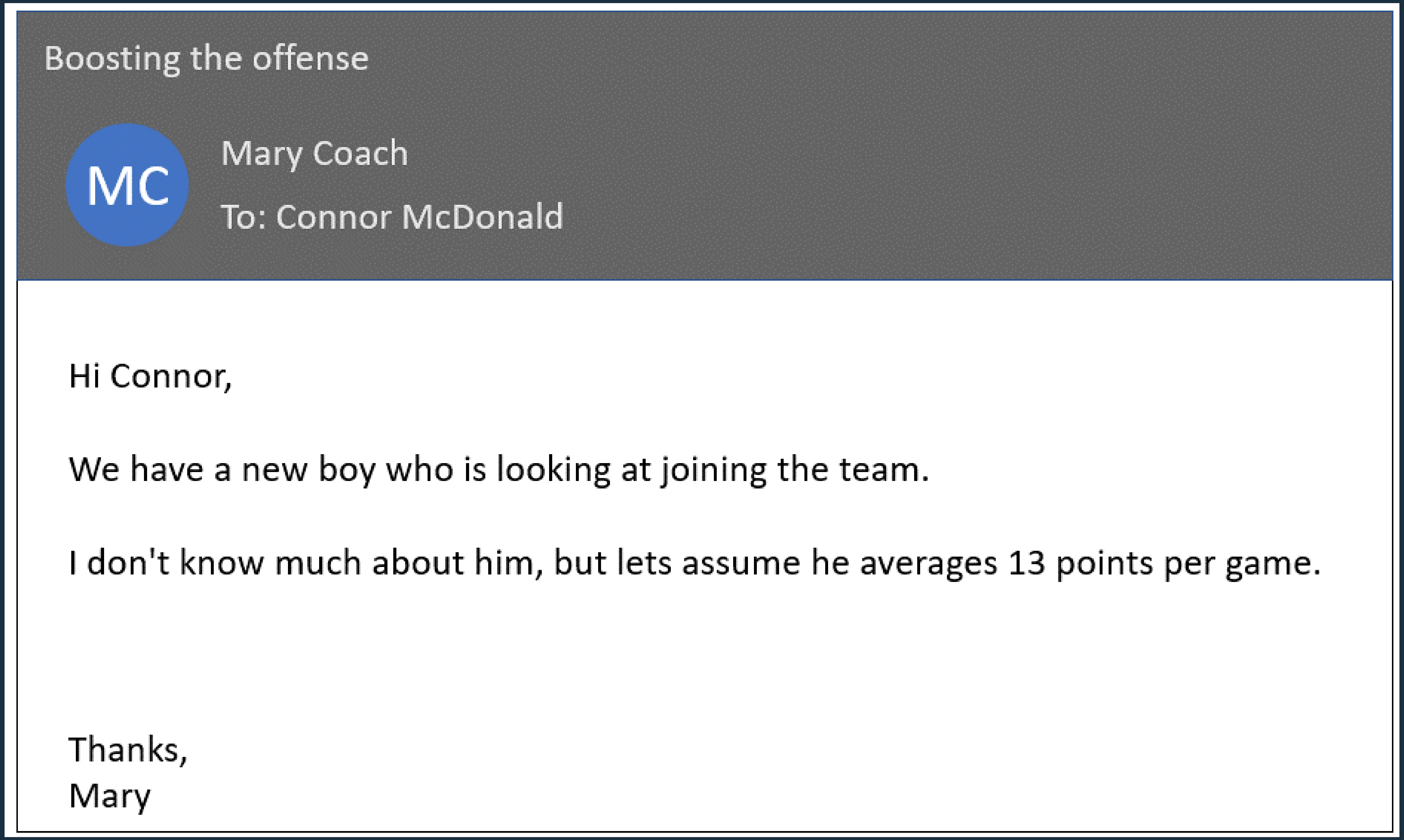


Player : Campbell

Qtr	Pts
1	10
2	1
3	9
4	6

Player : Matt

Qtr	Pts
1	5
2	2
3	



Boosting the offense

Mary Coach

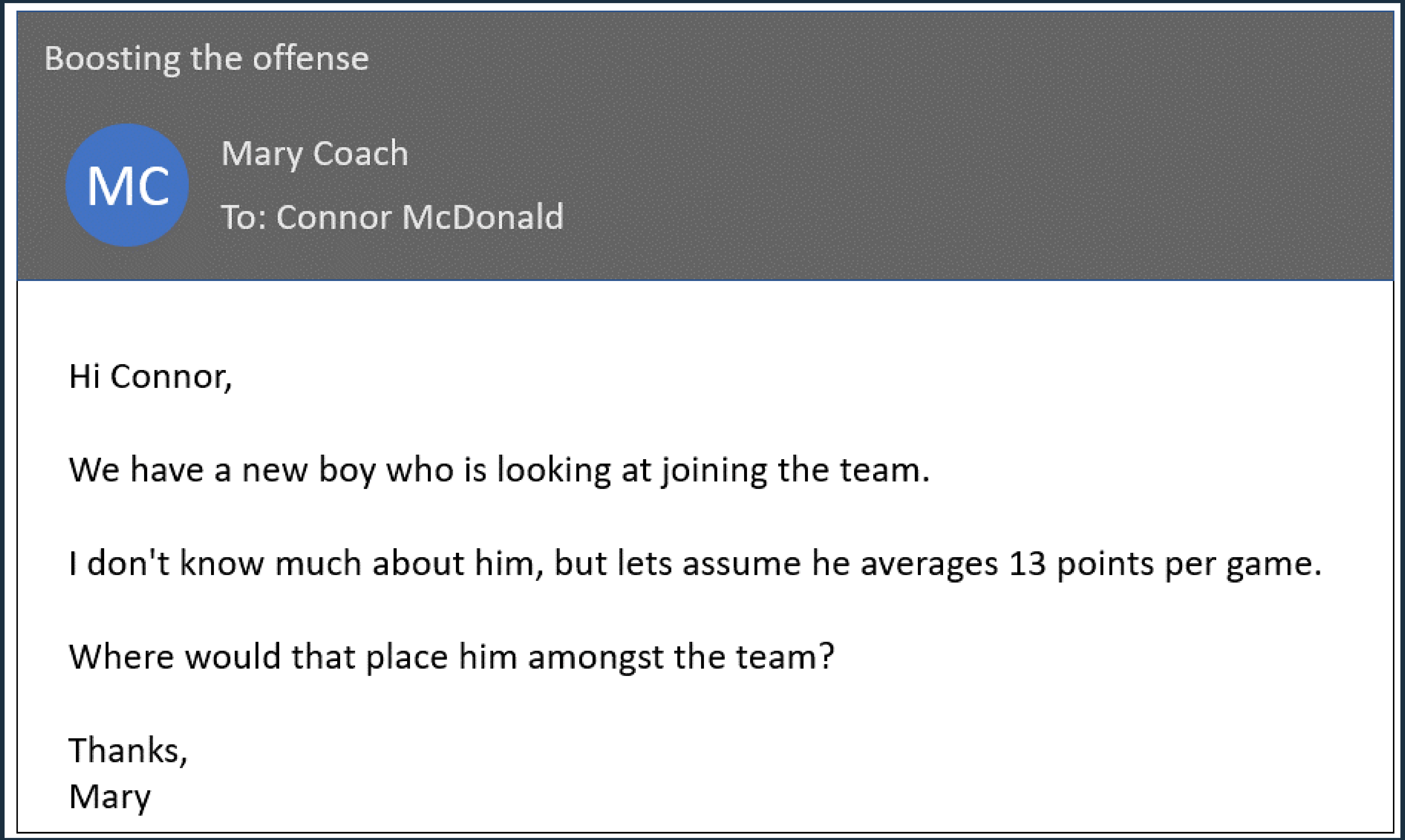
To: Connor McDonald

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



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.

Where would that place him amongst the team?

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
```

```
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
```

```
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 * from aust_rules
  2  order by 1,2;

   PLAYER#    QUARTER POSITION        GOALS NAME
---------- ---------- ---------- ---------- --------
         1          1 Forward             0 Liam
         1          2 Forward             0 Liam
         1          3 Forward             1 Liam
         1          4 Forward             0 Liam
         2          1 Forward             0 Noah
         2          2 Forward             1 Noah
         2          3 Forward             1 Noah
         2          4 Forward             1 Noah
         3          1 Centre              0 Oliver
         3          2 Centre              1 Oliver
         3          3 Centre              0 Oliver
         3          4 Centre              0 Oliver
         4          1 Defence             1 James
         4          2 Defence             0 James
         4          3 Defence             1 James
         4          4 Defence             1 James
...
...
```

"Find the highest goals scored in a single quarter by each player."

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

NAME                   HI
-------------- ----------
Aden                    1
Alex                    1
Ben                     0
Daniel                  1
David                   0
Ethan                   0
Henry                   0
Jack                    1
James                   1
John                    1
Joseph                  1
...
...
```

"...and what position were they playing when they scored them?"

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

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

```
SQL> select name, position, max(goals) hi
  2  from aust_rules
  3  group by name
  4  order by 1;
```

```
SQL> select name, position, max(goals) hi
  2  from aust_rules
  3  group by name
  4  order by 1;

select name, position, 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(goals) as hi,
  3         max(position) keep ( dense_rank last order by goals ) as hi_pos
  4  from aust_rules
  5  group by name
  6  order by 1;

NAME                    HI HI_POS
-------------- ---------- --------------------
Aden                     1 Centre
Alex                     1 Defence
Ben                      0 Forward
Daniel                   1 Forward
David                    0 Forward
Ethan                    0 Defence
Henry                    0 Centre
Jack                     1 Defence
James                    1 Defence
John                     1 Defence
Joseph                   1 Defence
...
```

```
SQL> select name,
  2         max(goals) as hi,
  3         max(position) keep ( dense_rank last order by goals ) as hi_pos
  4  from aust_rules
  5  group by name
  6  order by 1;

NAME                   HI HI_POS
-------------- ---------- --------------------
Aden                    1 Centre
Alex                    1 Defence
Ben                     0 Forward
Daniel                  1 Forward
David                   0 Forward
Ethan                   0 Defence
Henry                   0 Centre
Jack                    1 Defence
James                   1 Defence
John                    1 Defence
Joseph                  1 Defence
...
```

```
SQL> select name,
  2         max(goals) as hi,
  3         min(position) keep ( dense_rank last order by goals ) as hi_pos
  4  from aust_rules
  5  group by name
  6  order by 1;

NAME                   HI HI_POS
-------------- ---------- --------------------
Aden                    1 Centre
Alex                    1 Defence
Ben                     0 Forward
Daniel                  1 Forward
David                   0 Forward
Ethan                   0 Defence
Henry                   0 Centre
Jack                    1 Defence
James                   1 Defence
John                    1 Defence
Joseph                  1 Defence
...
```

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

NAME                   HI HI_POS
-------------- ---------- --------------------
Aden                    1 Centre
Alex                    1 Defence
Ben                     0 Forward
Daniel                  1 Forward
David                   0 Forward
Ethan                   0 Defence
Henry                   0 Centre
Jack                    1 Defence
James                   1 Defence
John                    1 Defence
Joseph                  1 Defence
...
```

As requirements get more complex …

… SQL gets more complex

"Find the total goals scored by each player,
then with the average of these totals,
list those players who got above that average."

*"Find the total goals scored by each player,
then with the average of these totals,
list those players who got above that average."*

SQL ?

common table expressions

WITH clause

```
select quarter, sum(goals)
from   aust_rules
group by quarter
```

```
WITH subtotals AS
(
  select quarter, sum(goals)
  from   aust_rules
  group by quarter
)
select * from subtotals;
```

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

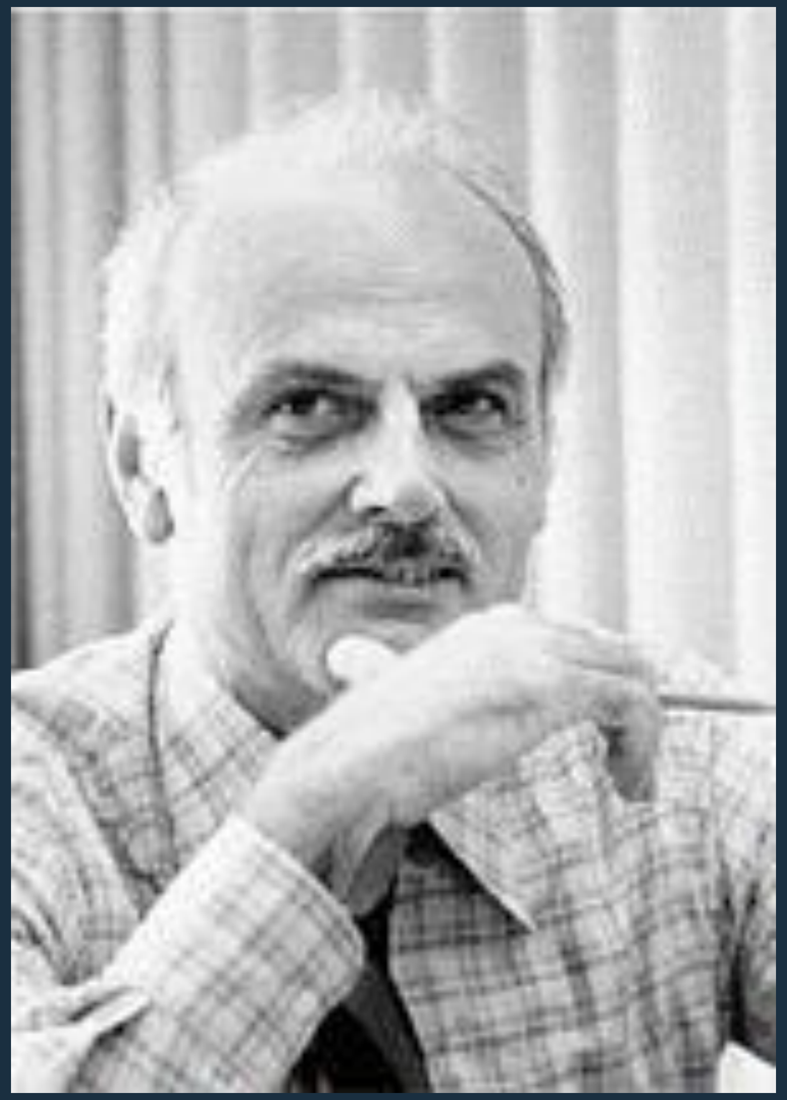
great mental model for developers

relational is a rigorous model …

relational is the dominant model …

relational … can be 🤯

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."

Image: https://en.wikipedia.org/wiki/File:Edgar_F_Codd.jpg

HUH?

procedural approach to relational

step by step

"First, find the total goals scored by each player…

```
SQL> with
  2   player_total as
  3  ( select player#, sum(goals) goals_per_player
  4     from aust_rules
  5     group by player#
  6  ),
```

"First, find the total goals scored by each player...

"... then with the average of these totals...

```
avg_goals as
( select avg(goals_per_player) avg_goals
  from player_total
)
```

"... then with the average of these totals...

"... now list those players who got above that average."

```
select *
from player_total,
     avg_goals
where goals_per_player > avg_goals
order by 1;
```

"... now list those players who got above that average."

```
SQL> with
  2   player_total as
  3  ( select player#, sum(goals) goals_per_player
  4    from aust_rules
  5    group by player#
  6  ),
  7  avg_goals as
  8  ( select avg(goals_per_player) avg_goals
  9    from player_total
 10  )
 11  select *
 12  from player_total,
 13       avg_goals
 14  where goals_per_player > avg_goals
 15  order by 1;

   PLAYER# GOALS_PER_PLAYER  AVG_GOALS
---------- ---------------- ----------
         2                3       1.12
         4                3       1.12
         5                2       1.12
```

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
partition by (b.player)
right outer join
  ( select rownum qtr from dual connect by level <= 4 ) q
on ( q.qtr = b.quarter )
```

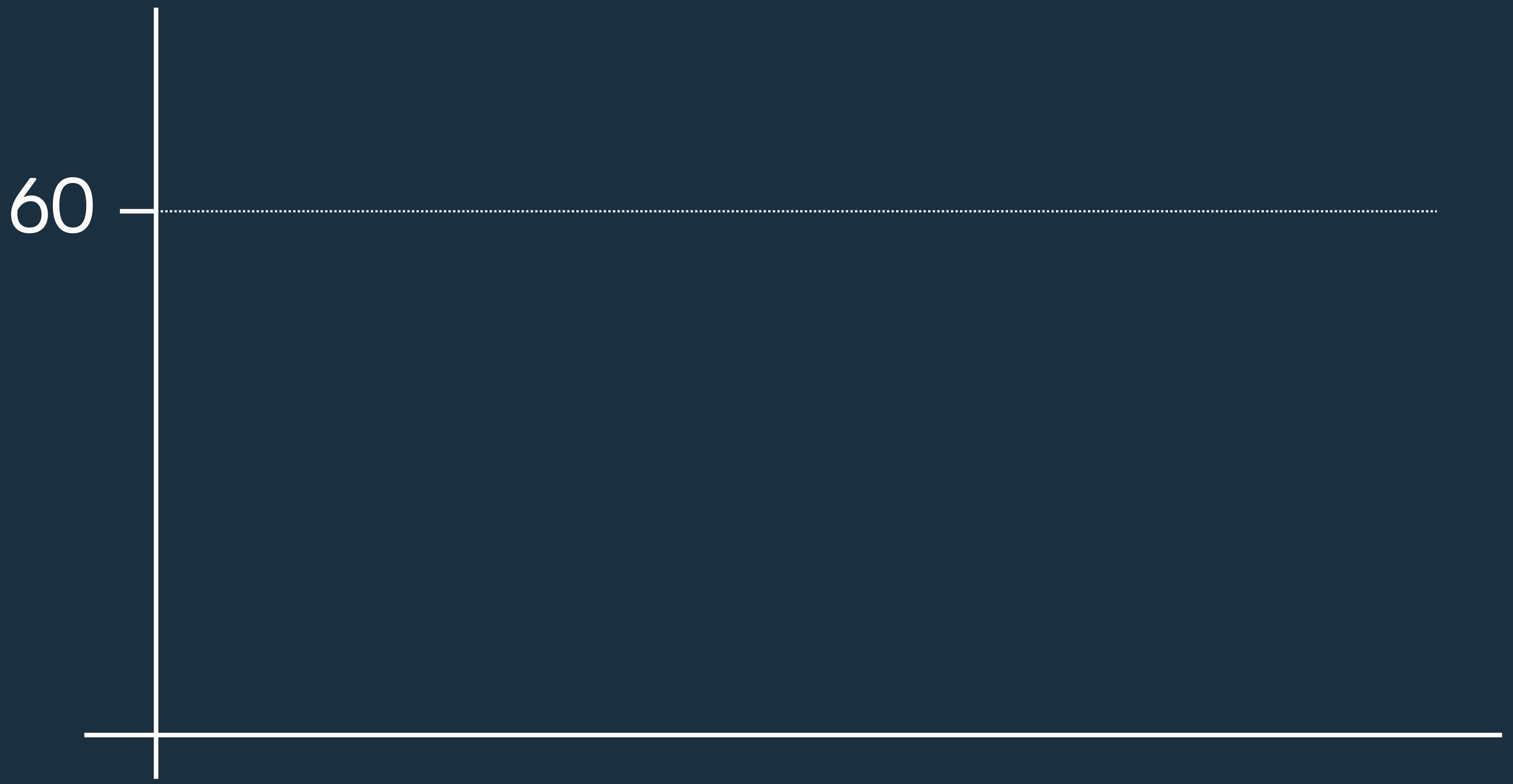
```
SQL> with raw_data as (
       select qtr, player, pts
       from
         ( select quarter, player, sum(points) pts
           from basketball
           group by quarter, player ) b
       partition by (b.player)
       right outer join
         ( select rownum qtr from dual connect by level <= 4 ) q
       on ( q.qtr = b.quarter )
     )
     select
       json_arrayagg(
           json_object(key player value pts )
           order by qtr ) as results
     from raw_data;
```

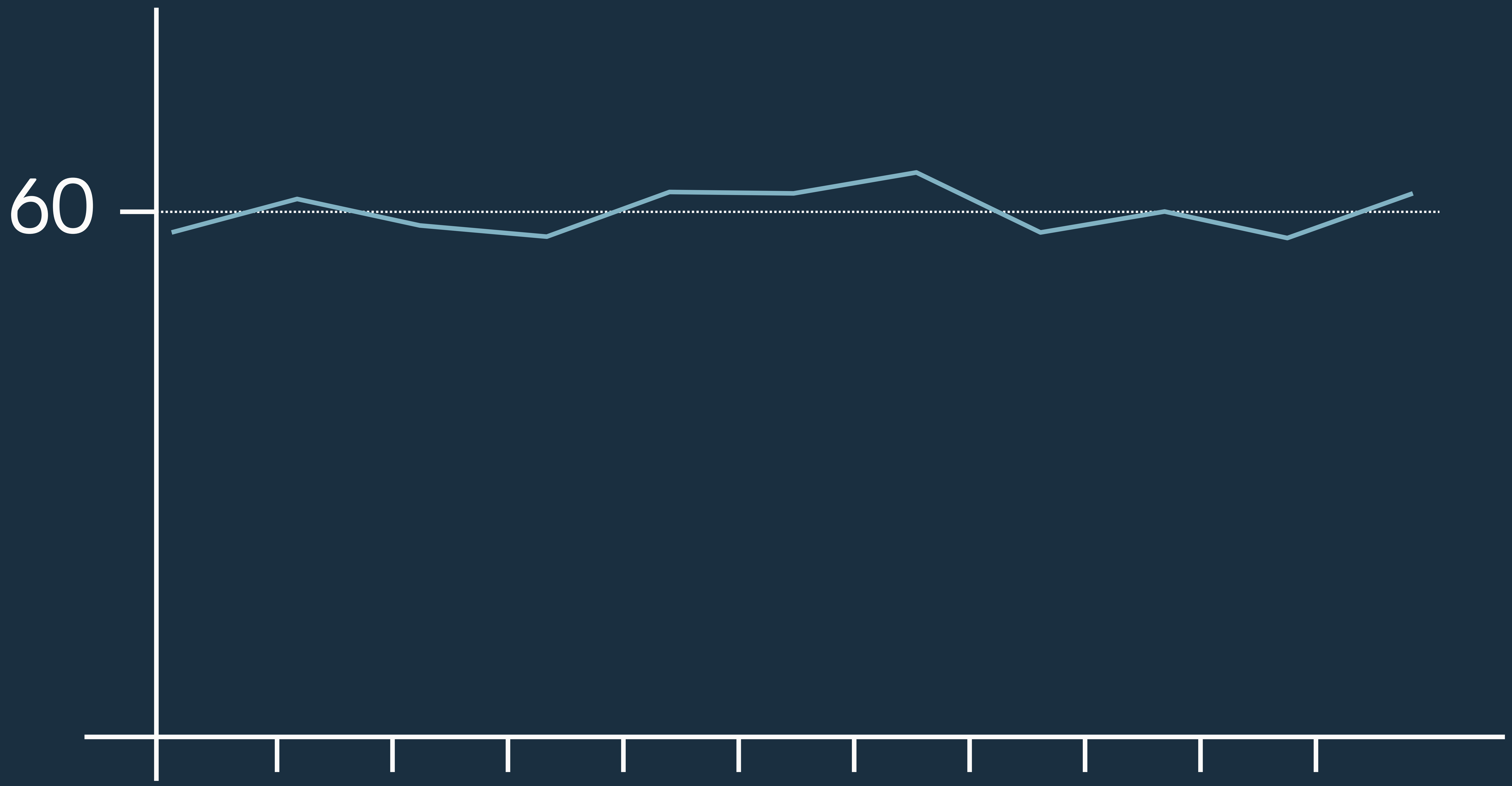
```
SQL> with raw_data as (
  2    select qtr, player, pts
  3    from
  4      ( select quarter, player, sum(points) pts
  5        from basketball
  6        group by quarter, player ) b
  7    partition by (b.player)
  8    right outer join
  9      ( select rownum qtr from dual connect by level <= 4 ) q
 10    on ( q.qtr = b.quarter )
 11  )
 12  select
 13    json_arrayagg(
 14        json_object(key player value pts )
 15        order by qtr ) as results
 16  from raw_data;
```

```
SQL> with raw_data as (
       select qtr, player, pts
       from
         ( select quarter, player, sum(points) pts
           from basketball
           group by quarter, player ) b
       partition by (b.pl
       right outer join
         ( select rownum
       on ( q.qtr = b.qua
     )
     select
       json_arrayagg(
           json_object(ke
           order by qtr )
     from raw_data;
```

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

Some "me" time


60


60

```
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
```

"bring the tempo to 60 seconds"

```
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       60.2
12-JUL-22          4       61.4
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 *
  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       60.2
12-JUL-22          4       61.4
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 *
  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       60.2
12-JUL-22          4       61.4
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 *
  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       60.2
12-JUL-22          4       61.4
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 *
  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       60.2
12-JUL-22          4       61.4
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
```

```
-1.3
-0.3
+0.2
+1.4
====
 0.0
```

60

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

Table altered.
```

```
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
--------- ---------- ---------- ---------- ----------
```

```
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
--------- ---------- ---------- ---------- ----------
```

```
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
```

```
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"
```

```
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"
```

```
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
```

```
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"
```

```
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
```

```
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"
```

```
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"
```

```
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
```

```
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
```

```
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" ☺
```

```
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
```

```
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
```

```
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`

```
bad_lap* zero
```

```
zero as sum(delta) = 0
```

```
pattern (bad_lap* zero)
define zero as sum(delta) = 0
```

```
pattern (bad_lap* zero)
define zero as sum(delta) = 0
```

```
SQL> select * from swimming
     match_recognize (
       partition by sess order by lap
       measures classifier() pattern, sum(delta) as run_tot
       all rows per match
       pattern (bad_lap* zero)
       define 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
```

```
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
```

```
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
```

```
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
```

```
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

SQL> select avg(ela) from swimming;

  AVG(ELA)
----------
     59.93
```

```
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
```

60

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

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
```

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

SESS            LAP PATTERN        RUN_TOT        ELA
--------- ----- ---------- ---------- ----------
12-JUL-22
12-JUL-22
12-JUL-22
12-JUL-22
```

```
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               .5       58.6
12-JUL-22      9 BAD_LAP               .1       59.6
12-JUL-22     10 ZERO                   0       59.9
```

```
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 BAD_LAP           1.3       58.7
12-JUL-22
12-JUL-22
12-JUL-22
```

```
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-
12-JUL-
12-JUL-
12-JUL-
12-JUL-
```

```
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             .5       58.6
12-JUL-22     9 BAD_LAP             .1       59.6
12-JUL-22    10 ZERO                 0       59.9
```

```
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
12-JUL-22
12-JUL-22
12-JUL-22
```

```
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-
12-JUL-
12-JUL-
12-JUL-
12-JUL-
12-JUL-
```

```
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
12-JUL-22
12-JUL-22
12-JUL-22
```

```
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

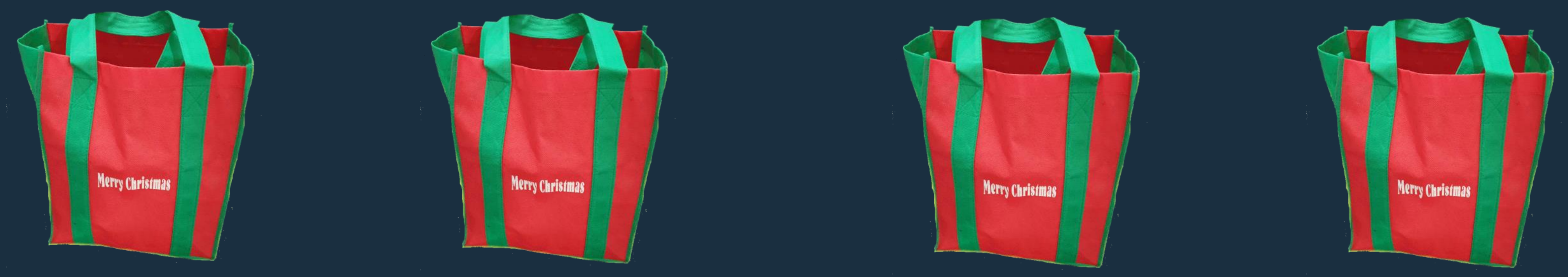
Boys need ^ feeding!

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                    290
ketchup                     540
eggs                        800
detergent                   950
deodrant                    220
```



```
SQL> select * from shopping;

ITEM                     WEIGHT
-------------------- ----------
milk
bread
dogfood
biscuits
soda
gin
apples
bananas
carrots
steak
icecream
butter
honey
vegemite
ketchup
eggs
detergent
deodrant
```

650
220
800
450
1200
490
650
950
250
240
1000
290
1500
Merry Christmas
Merry Christmas
Merry Christmas
Merry Christmas

SQL can do this too !

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

define
```

I have 4 bags

(matching my as yet unknown rules)

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

define
```

I have 4 bags

(matching my as yet unknown rules)

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

define
```

I have 4 bags

(matching my as yet unknown rules)

use a bag if …

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

define
```

I have 4 bags

(matching my as yet unknown rules)

use a bag if ...

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

define
```

I have 4 bags

(matching my as yet unknown rules)

use a bag if …

first item in the bag, or

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

define
   bag1 as count(bag1.*) = 1 or
```

I have 4 bags

(matching my as yet unknown rules)

use a 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
        <= least(sum(bag2.weight),sum(bag3.weight),sum(bag4.weight))
```

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

I have 4 bags

(matching my as yet unknown rules)

use a 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
        <= least(sum(bag2.weight),sum(bag3.weight),sum(bag4.weight))
```

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

I have 4 bags

(matching my as yet unknown rules)

use a 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
        <= least(sum(bag2.weight),sum(bag3.weight),sum(bag4.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 then 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
    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  )
```

```
SQL> with portions as
  2  (
  3   select *
  4   from shopping
  5   match_recognize (
  6     order by weight desc
  7     measures
        ...
 25  )
 26  select
 27    bag#,
 28    listagg(item,',') within group ( order by item ) as items,
 29    sum(weight)/1000 kg
 30  from portions
 31  group by bag#;
```

```
SQL> with portions as
  2  (
  3   select *
  4   from shopping
  5   match_recognize (
  6     order by weight desc
  7     measures
        ...
 25  )
 26  select
 27    bag#,
 28    listagg(item,',') within group ( order by item ) as items,
 29    sum(weight)/1000 kg
 30  from portions
 31  group by bag#;
```

```
SQL> with portions as
     (
      select *
      from shopping
      match_recognize (
        order by weight desc
        measures
        ...
     )
     se
     
     
     
     fr
     gr
```

```
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
     function pack_and_carry(p_tab dbms_tf.table_t, p_bags int)
         return clob sql_macro is
       l_sql clob;
       l_bag varchar2(1000);
       l_sum varchar2(4000);
       l_pattern varchar2(4000);
     begin
      for i in 1 .. p_bags loop
        l_bag := l_bag || 'bag'||i||'|';
        l_sum := l_sum || replace('sum(bag@.weight) bag@,','@',i)||chr(10);
        if i < p_bags then
          if i < p_bags-1 then
            l_pattern := l_pattern ||
            replace(',bag@ as count(bag@.*)=1 or sum(bag@.weight)-bag@.weight <= least(','@',i);
          else
            l_pattern := l_pattern ||
            replace(',bag@ as count(bag@.*)=1 or sum(bag@.weight)-bag@.weight <= ','@',i);
          end if;
          for j in i+1 .. p_bags loop
            l_pattern := l_pattern ||replace('sum(bag@.weight),','@',j);
          end loop;
          l_pattern := rtrim(l_pattern,',')||')'||chr(10);
        end if;
      end loop;
```

```
  l_sql := q'{
select * from p_tab
match_recognize (
 order by weight desc
 measures
    classifier() bag#,
    ~~~
all rows per match
pattern ( (###)* )
define $$$}';

  l_sql := replace(l_sql,'###',rtrim(l_bag,'|'));
  l_sql := replace(l_sql,'~~~',rtrim(l_sum,','||chr(10)));
  l_sql := replace(l_sql,'$$$',ltrim(l_pattern,','));
  return l_sql;
end;
/

Function created.
```

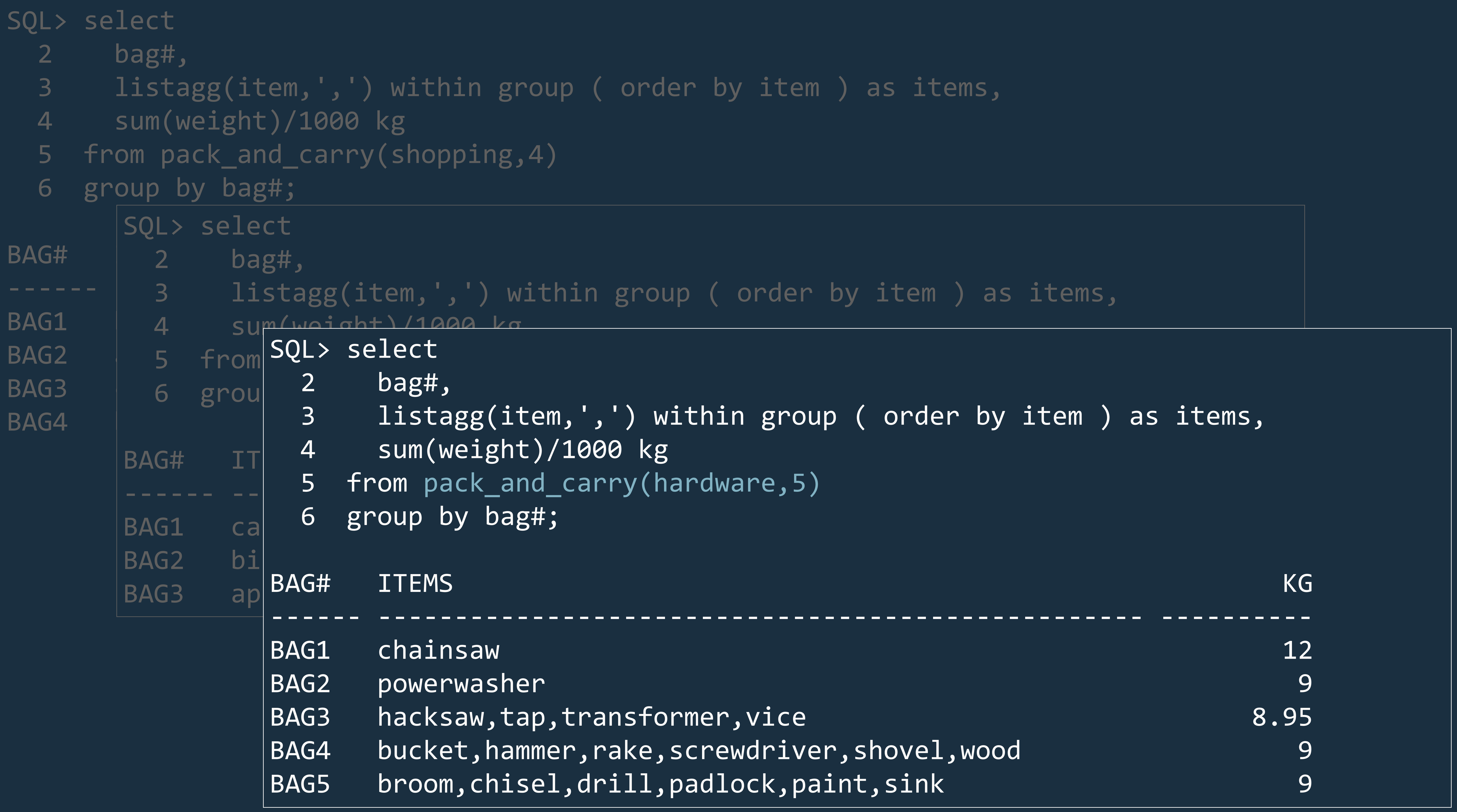
```
SQL> select
  2    bag#,
  3    listagg(item,',') within group ( order by item ) as items,
  4    sum(weight)/1000 kg
  5  from pack_and_carry(shopping,4)
  6  group by bag#;

BAG#   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
```

```
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 pack_and_carry(shopping,3)
  6  group by bag#;

BAG#   ITEMS                                               KG
------ ---------------------------------------- ----------
BAG1   carrots,deodrant,detergent,gin,honey,vegemite     4.83
BAG2   biscuits,bread,butter,eggs,milk,soda              4.65
BAG3   apples,bananas,dogfood,icecream,ketchup,steak     4.67
```

```
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 pack_and_carry(shopping,3)
  6  group by bag#;

BAG#   ITEMS                                                      KG
------ -------------------------------------------------- ----------
BAG1   carrots,deodrant,detergent,gin,honey,vegemite            4.83
BAG2   biscuits,bread,butter,eggs,milk,soda                     4.65
BAG3   apples,bananas,dogfood,icecream,ketchup,steak            4.67
```



```
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
```

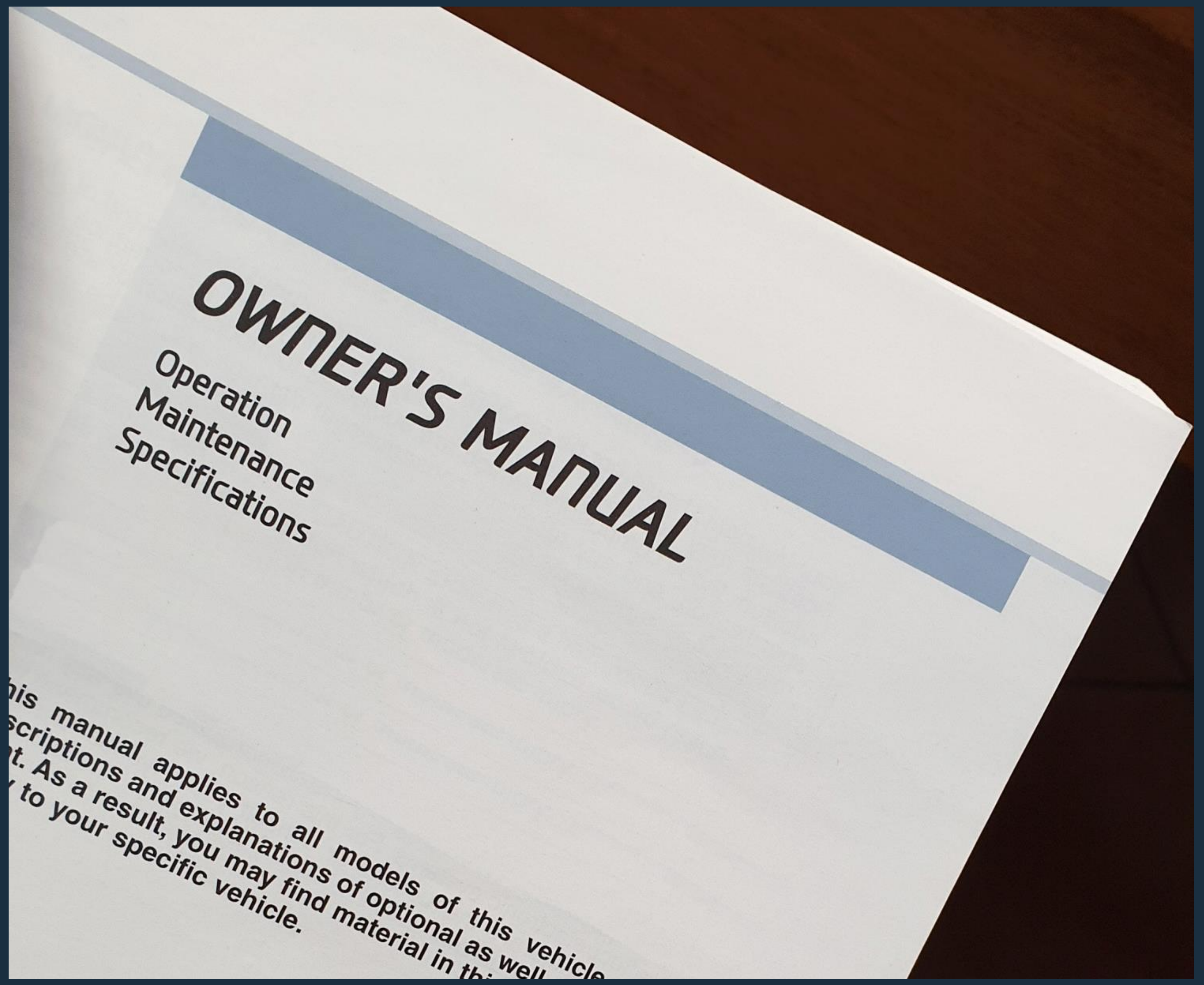
```
SQL> select
  2    bag#,
  3    listagg(item,',') within group ( order by item ) as items,
  4    sum(weight)/1000 kg
  5  from
  6  grou

BAG#   IT
------ --
BAG1   ca
BAG2   bi
BAG3   ap
```

```
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
```

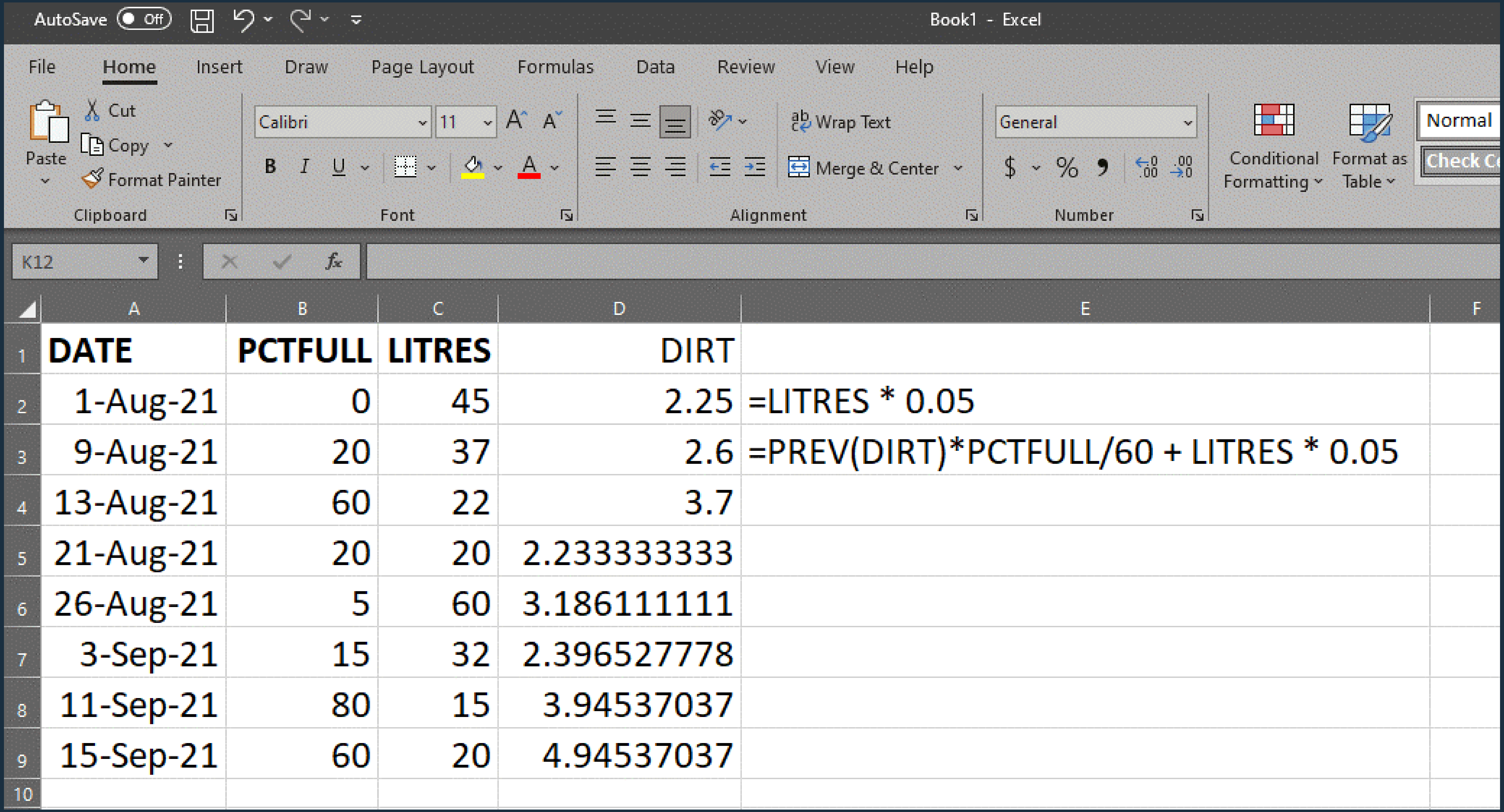


OWNER'S MANUAL

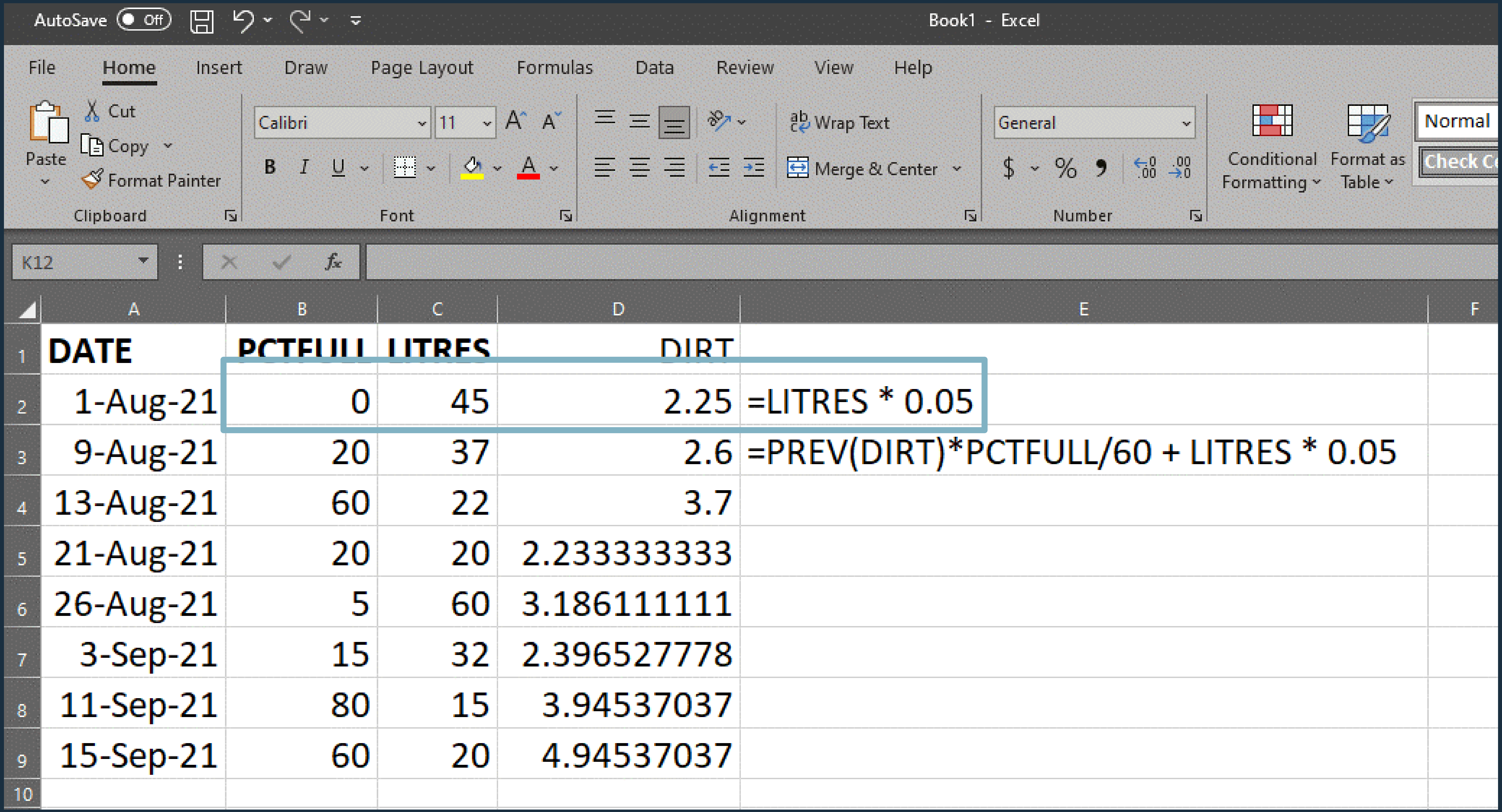
Operation
Maintenance
Specifications

is manual applies to all models of this vehicle
scriptions and explanations of optional as well
t. As a result, you may find material in thi
r to your specific vehicle.

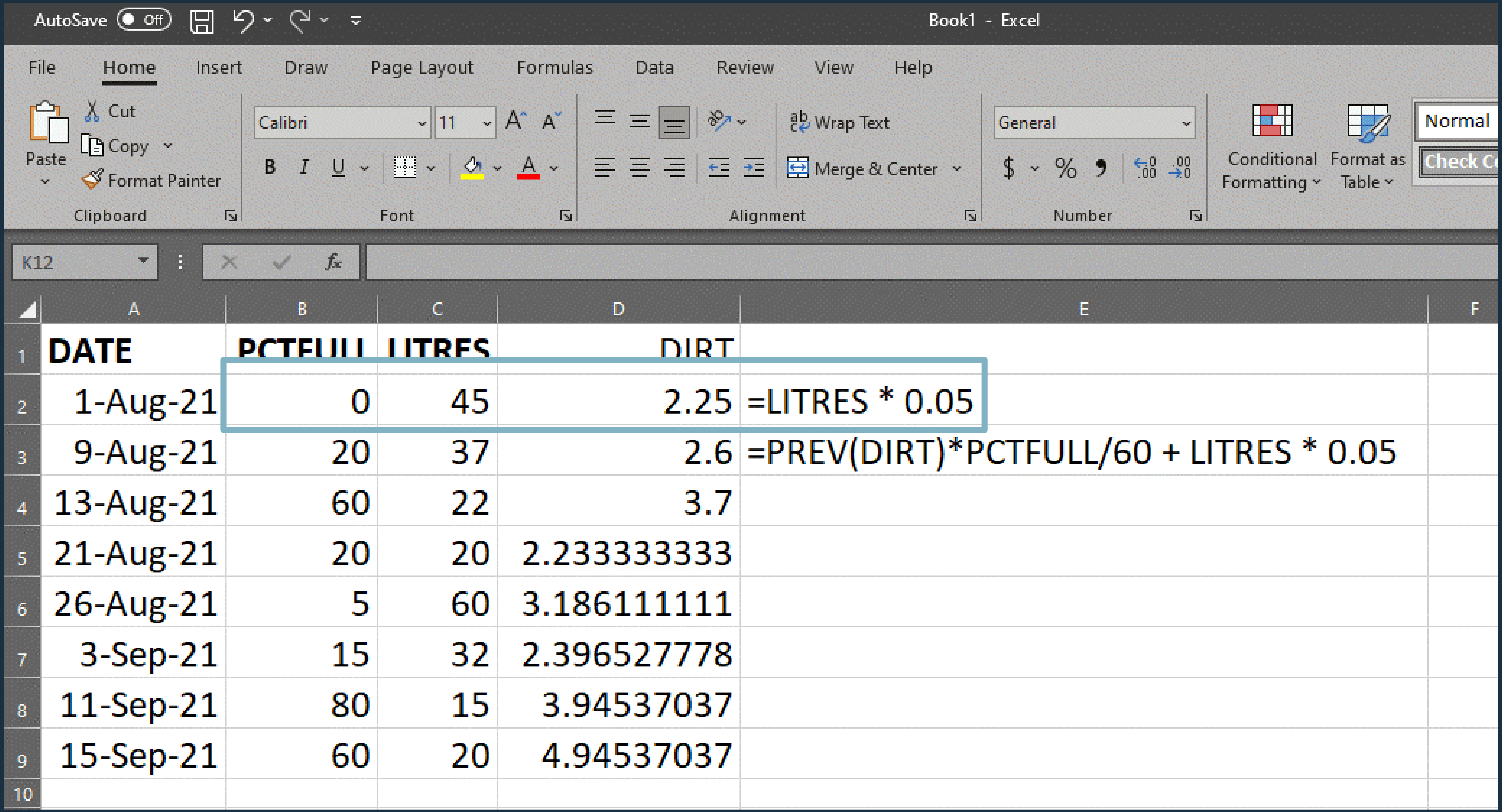
"Clean fuel"



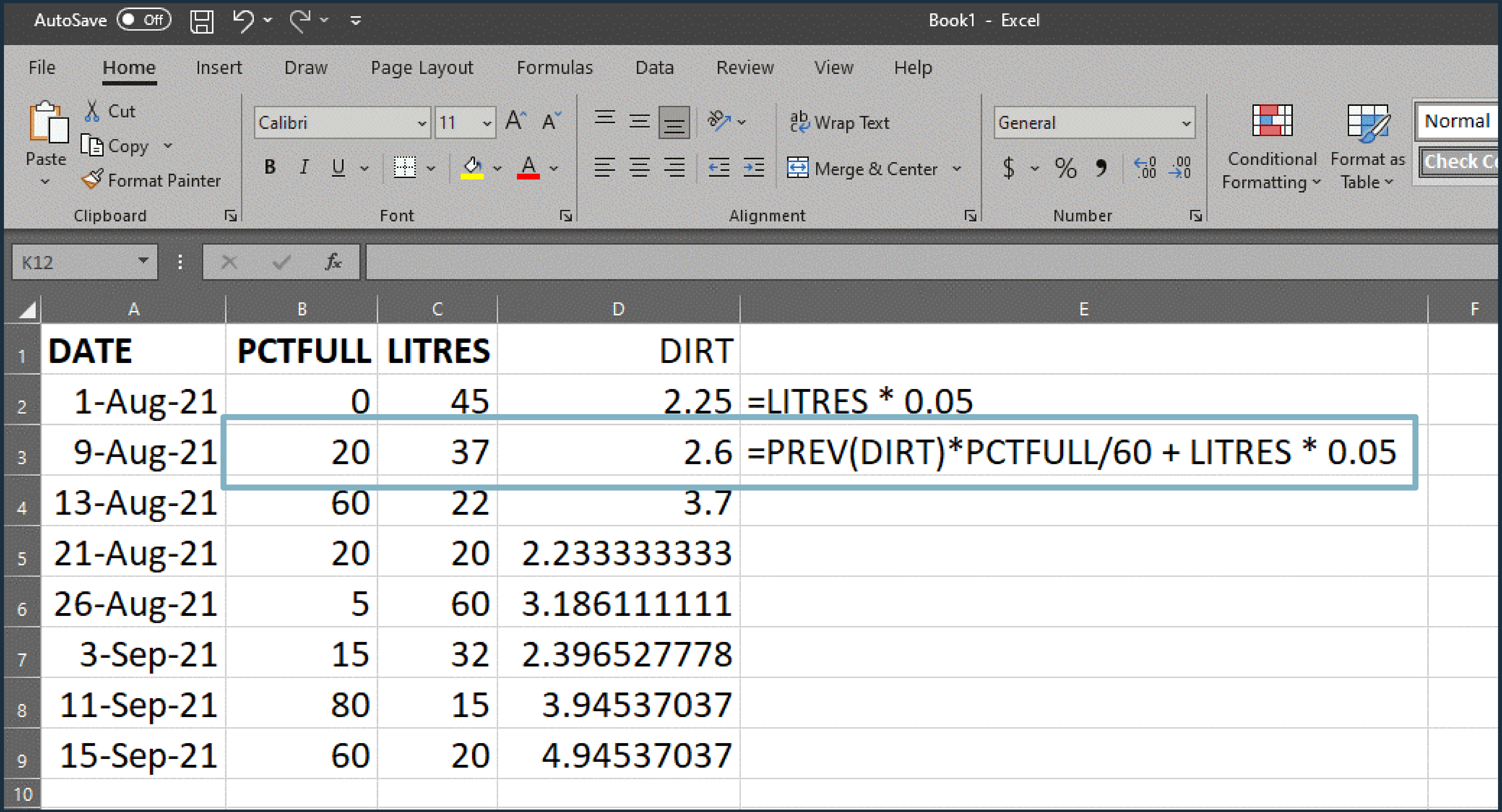
	A	B	C	D	E
1	**DATE**	**PCTFULL**	**LITRES**	DIRT	
2	1-Aug-21	0	45	2.25	=LITRES * 0.05
3	9-Aug-21	20	37	2.6	=PREV(DIRT)*PCTFULL/60 + LITRES * 0.05
4	13-Aug-21	60	22	3.7	
5	21-Aug-21	20	20	2.233333333	
6	26-Aug-21	5	60	3.186111111	
7	3-Sep-21	15	32	2.396527778	
8	11-Sep-21	80	15	3.94537037	
9	15-Sep-21	60	20	4.94537037	
10					



	A	B	C	D	E
1	DATE	PCTFULL	LITRES	DIRT	
2	1-Aug-21	0	45	2.25	=LITRES * 0.05
3	9-Aug-21	20	37	2.6	=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	
7	3-Sep-21	15	32	2.396527778	
8	11-Sep-21	80	15	3.94537037	
9	15-Sep-21	60	20	4.94537037	
10					



	A	B	C	D	E
1	DATE	PCTFULL	LITRES	DIRT	
2	1-Aug-21	0	45	2.25	=LITRES * 0.05
3	9-Aug-21	20	37	2.6	=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	
7	3-Sep-21	15	32	2.396527778	
8	11-Sep-21	80	15	3.94537037	
9	15-Sep-21	60	20	4.94537037	
10					



	A	B	C	D	E
1	DATE	PCTFULL	LITRES	DIRT	
2	1-Aug-21	0	45	2.25	=LITRES * 0.05
3	9-Aug-21	20	37	2.6	=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	
7	3-Sep-21	15	32	2.396527778	
8	11-Sep-21	80	15	3.94537037	
9	15-Sep-21	60	20	4.94537037	
10					

```
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
```

```
SQL> select *
  2  from car_fuel;

DTE             PCTFULL     LITRES          DIRT
--------- ---------- ----------     ---------
01-AUG-21          0         45          2.25
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> select *
  2  from car_fuel;

DTE          PCTFULL     LITRES          DIRT
--------- ---------- ----------     ---------
01-AUG-21          0         45          2.25
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> select *
  2  from car_fuel;

DTE          PCTFULL     LITRES          DIRT
--------- ---------- ----------     ---------
01-AUG-21          0         45          2.25
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> 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> select
  2    car_fuel.*,
  3    row_number() over (order by dte ) as seq
  4  from car_fuel;

DTE          PCTFULL     LITRES        SEQ
--------- ---------- ---------- ----------
01-AUG-21          0         45          1
09-AUG-21         20         37          2
13-AUG-21         60         22          3
21-AUG-21         20         20          4
26-AUG-21          5         60          5
03-SEP-21         15         32          6
11-SEP-21         80         15          7
15-SEP-21         60         20          8
```

```
SQL> with t as
  2  ( select
  3      car_fuel.*,
  4      row_number() over (order by dte ) as seq
  5    from car_fuel
  6  ),
SQL> with t as
  2  ( select
  3      car_fuel.*,
  4      row_number() over (order by dte ) as seq
  5    from car_fuel
  6  ),
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  (
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  (
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
```

```
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
```

```
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;
```

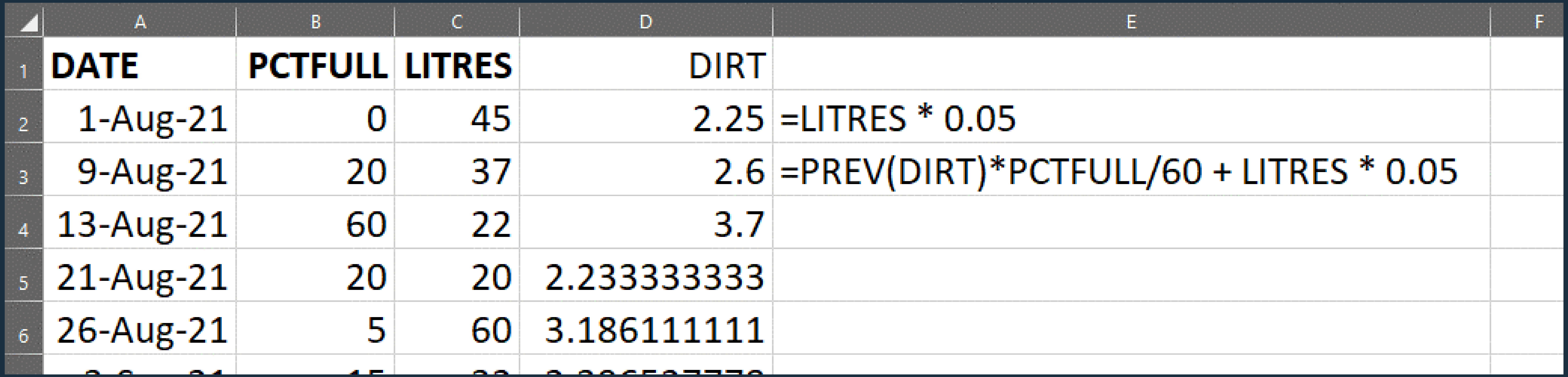
```
SQL> with t as
  2  ( select
  3      car_fuel.*,
  4      row_number() over (order by dte ) as seq
  5    from car_fuel
  6  ),
  7  results(dte, pctfull, litres, dirt ,seq) as
  8  (
  9    select dte, pctfull, litres, litres*0.05 dirt, seq
 10    from t
 11    where seq = 1
 12    union all
 13    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." ☹

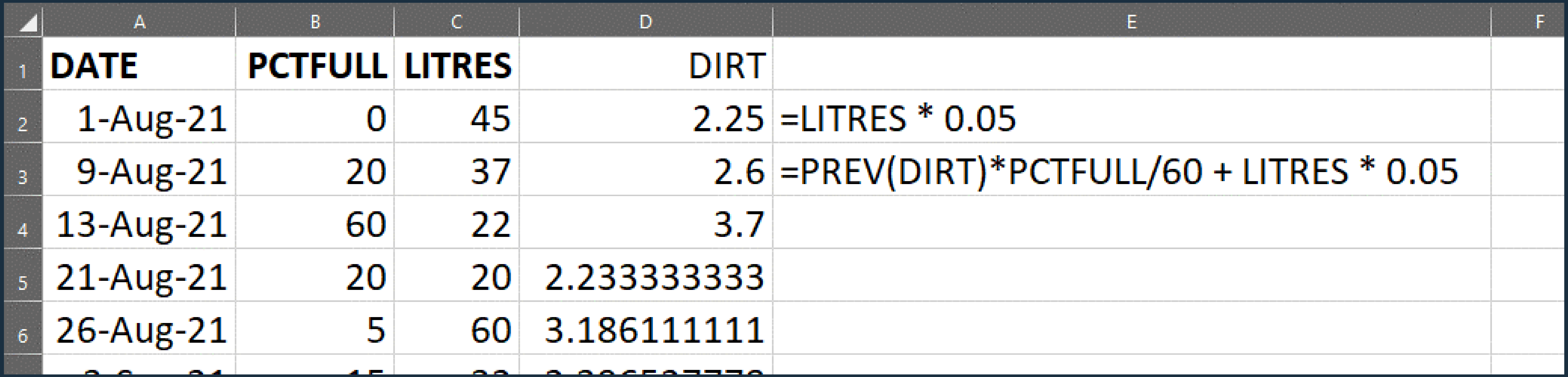
cell formula expressions in SQL

```
SQL> select dte, pctfull, litres, dirt
  2  from car_fuel
  3  model
```



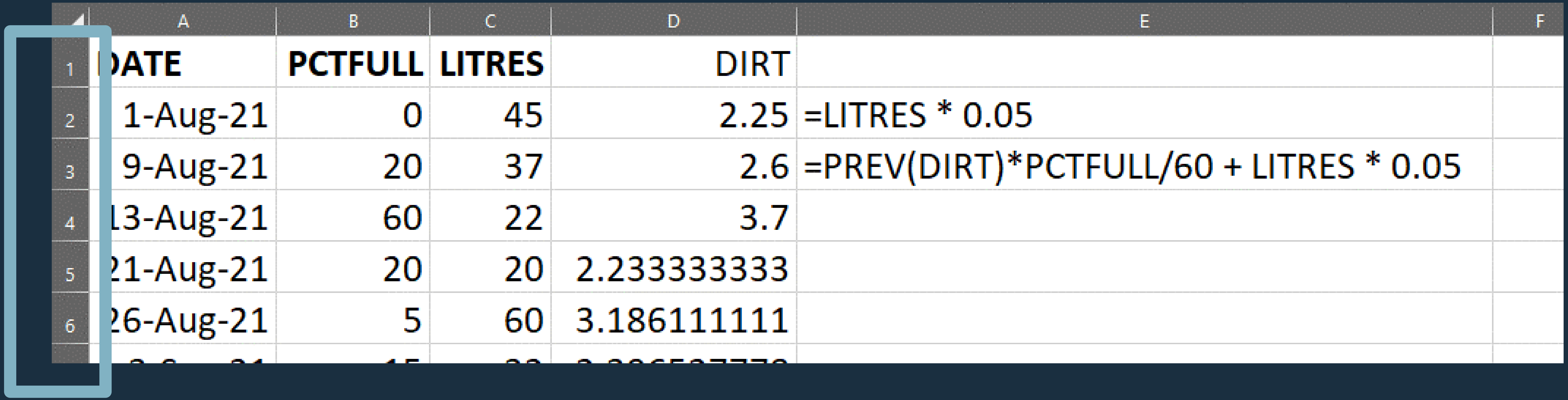
	A	B	C	D	E	F
1	**DATE**	**PCTFULL**	**LITRES**	DIRT		
2	1-Aug-21	0	45	2.25	=LITRES * 0.05	
3	9-Aug-21	20	37	2.6	=PREV(DIRT)*PCTFULL/60 + LITRES * 0.05	
4	13-Aug-21	60	22	3.7		
5	21-Aug-21	20	20	2.233333333		
6	26-Aug-21	5	60	3.186111111		

```
SQL> select dte, pctfull, litres, dirt
  2  from car_fuel
  3  model
```



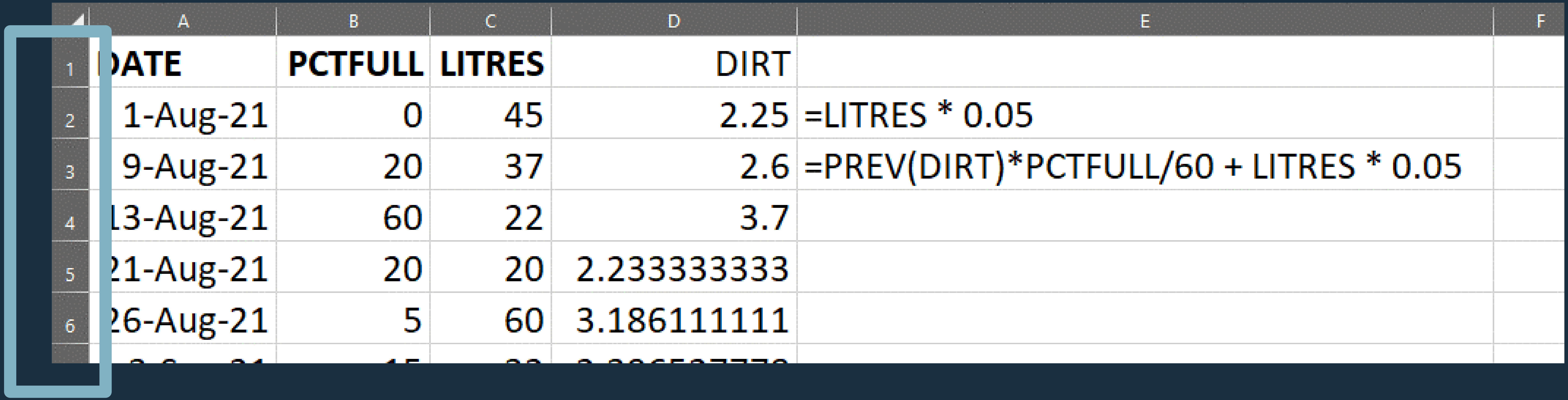
	A	B	C	D	E	F
1	DATE	PCTFULL	LITRES	DIRT		
2	1-Aug-21	0	45	2.25	=LITRES * 0.05	
3	9-Aug-21	20	37	2.6	=PREV(DIRT)*PCTFULL/60 + LITRES * 0.05	
4	13-Aug-21	60	22	3.7		
5	21-Aug-21	20	20	2.233333333		
6	26-Aug-21	5	60	3.186111111		

```
SQL> select dte, pctfull, litres, dirt
  2  from car_fuel
  3  model
  4  dimension by ( row_number() over(order by dte) seq )
```



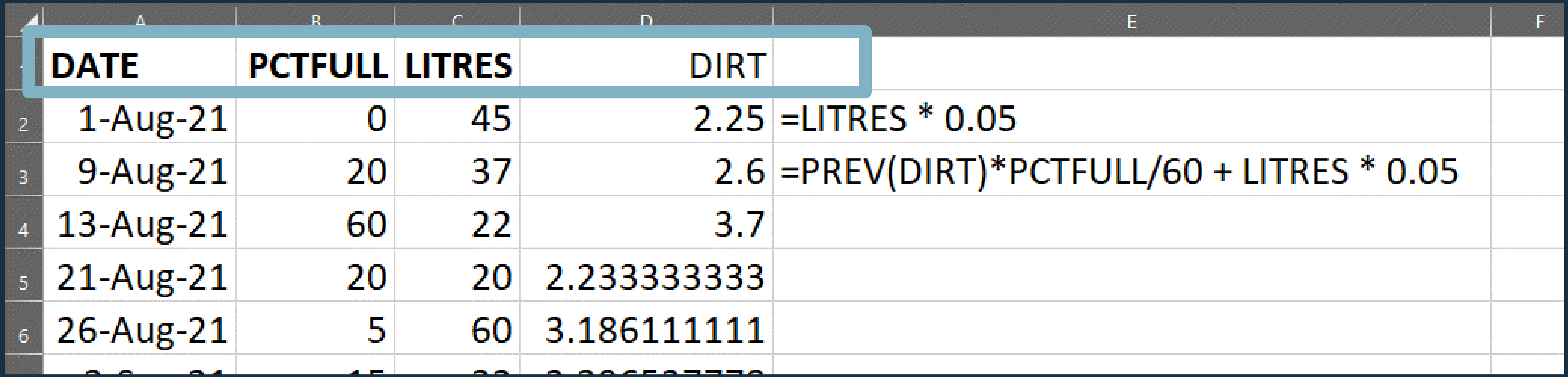
	A	B	C	D	E	F
1	DATE	PCTFULL	LITRES	DIRT		
2	1-Aug-21	0	45	2.25	=LITRES * 0.05	
3	9-Aug-21	20	37	2.6	=PREV(DIRT)*PCTFULL/60 + LITRES * 0.05	
4	13-Aug-21	60	22	3.7		
5	21-Aug-21	20	20	2.233333333		
6	26-Aug-21	5	60	3.186111111		

```
SQL> select dte, pctfull, litres, dirt
  2  from car_fuel
  3  model
  4  dimension by ( row_number() over(order by dte) seq )
```



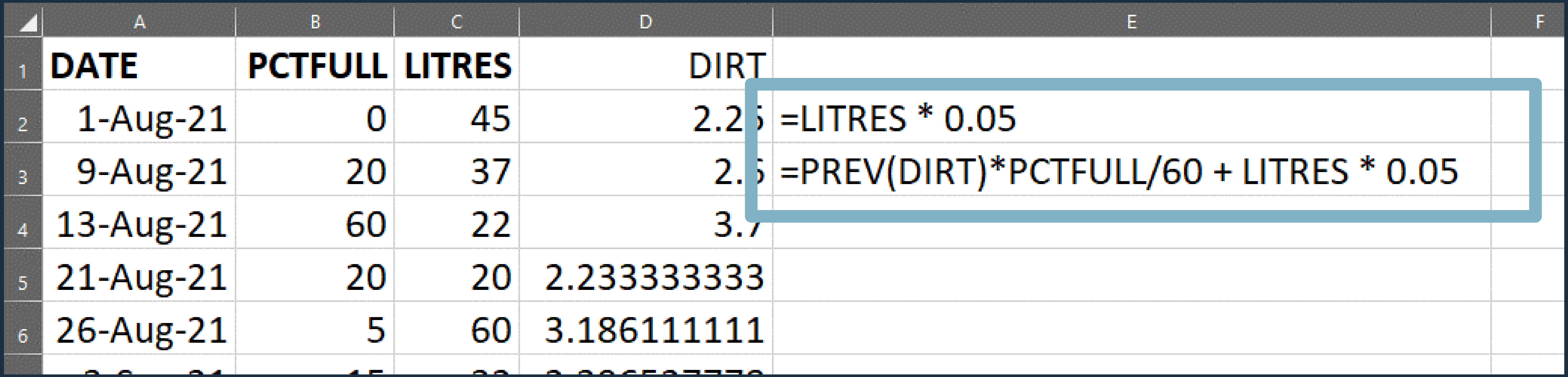
	A	B	C	D	E	F
1	**DATE**	**PCTFULL**	**LITRES**	DIRT		
2	1-Aug-21	0	45	2.25	=LITRES * 0.05	
3	9-Aug-21	20	37	2.6	=PREV(DIRT)*PCTFULL/60 + LITRES * 0.05	
4	13-Aug-21	60	22	3.7		
5	21-Aug-21	20	20	2.233333333		
6	26-Aug-21	5	60	3.186111111		

```
SQL> select dte, pctfull, litres, dirt
  2  from car_fuel
  3  model
  4  dimension by ( row_number() over(order by dte) seq )
  5  measures( dte, pctfull,litres, 0 dirt )
```



	A	B	C	D	E	F
1	DATE	PCTFULL	LITRES	DIRT		
2	1-Aug-21	0	45	2.25	=LITRES * 0.05	
3	9-Aug-21	20	37	2.6	=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
     from car_fuel
     model
     dimension by ( row_number() over(order by dte) seq )
     measures( dte, pctfull,litres, 0 dirt )
       rules(
          dirt[any] order by seq =
             presentnnv(dirt[cv()-1],dirt[cv()-1],0) *
             pctfull[cv()]/60 +
             litres[cv()]*0.05
             );
```



	A	B	C	D	E	F
1	DATE	PCTFULL	LITRES	DIRT		
2	1-Aug-21	0	45	2.25	=LITRES * 0.05	
3	9-Aug-21	20	37	2.6	=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( 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          );
```

```
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
  9
 10
 11
```

```
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
```

wrap up

Futuristic SQL

Intelligent SQL

- grouping sets
- row_number() + analytics
- hypothetical analytics
- partitioned outer join
- WITH clause
- KEEP clause
- MODEL clause
- recursive WITH
- MATCH_RECOGNIZE
- SQL Macros

• grouping sets	23 years
• row_number() + analytics	23 years
• hypothetical analytics	23 years
• partitioned outer join	
• WITH clause	
• KEEP clause	
• MODEL clause	
• recursive WITH	
• MATCH_RECOGNIZE	
• SQL Macros	

• grouping sets	23 years
• row_number() + analytics	23 years
• hypothetical analytics	23 years
• partitioned outer join	
• WITH clause	
• KEEP clause	
• MODEL clause	
• recursive WITH	
• MATCH_RECOGNIZE	
• SQL Macros	

• grouping sets	23 years
• row_number() + analytics	23 years
• hypothetical analytics	23 years
• partitioned outer join	20 years
• WITH clause	20 years
• KEEP clause	20 years
• MODEL clause	
• recursive WITH	
• MATCH_RECOGNIZE	
• SQL Macros	

• grouping sets	23 years
• row_number() + analytics	23 years
• hypothetical analytics	23 years
• partitioned outer join	20 years
• WITH clause	20 years
• KEEP clause	20 years
• MODEL clause	
• recursive WITH	
• MATCH_RECOGNIZE	
• SQL Macros	

• grouping sets	23 years
• row_number() + analytics	23 years
• hypothetical analytics	23 years
• partitioned outer join	20 years
• WITH clause	20 years
• KEEP clause	20 years
• MODEL clause	18 years
• recursive WITH	
• MATCH_RECOGNIZE	
• SQL Macros	

grouping sets	23 years
row_number() + analytics	23 years
hypothetical analytics	23 years
partitioned outer join	20 years
WITH clause	20 years
KEEP clause	20 years
MODEL clause	18 years
recursive WITH	
MATCH_RECOGNIZE	
SQL Macros	

grouping sets	23 years
row_number() + analytics	23 years
hypothetical analytics	23 years
partitioned outer join	20 years
WITH clause	20 years
KEEP clause	20 years
MODEL clause	18 years
recursive WITH	12 years
MATCH_RECOGNIZE	
SQL Macros	

• grouping sets	23 years
• row_number() + analytics	23 years
• hypothetical analytics	23 years
• partitioned outer join	20 years
• WITH clause	20 years
• KEEP clause	20 years
• MODEL clause	18 years
• recursive WITH	12 years
• MATCH_RECOGNIZE	7 years
• SQL Macros	

grouping sets	23 years
row_number() + analytics	23 years
hypothetical analytics	23 years
partitioned outer join	20 years
WITH clause	20 years
KEEP clause	20 years
MODEL clause	18 years
recursive WITH	12 years
MATCH_RECOGNIZE	7 years
SQL Macros	

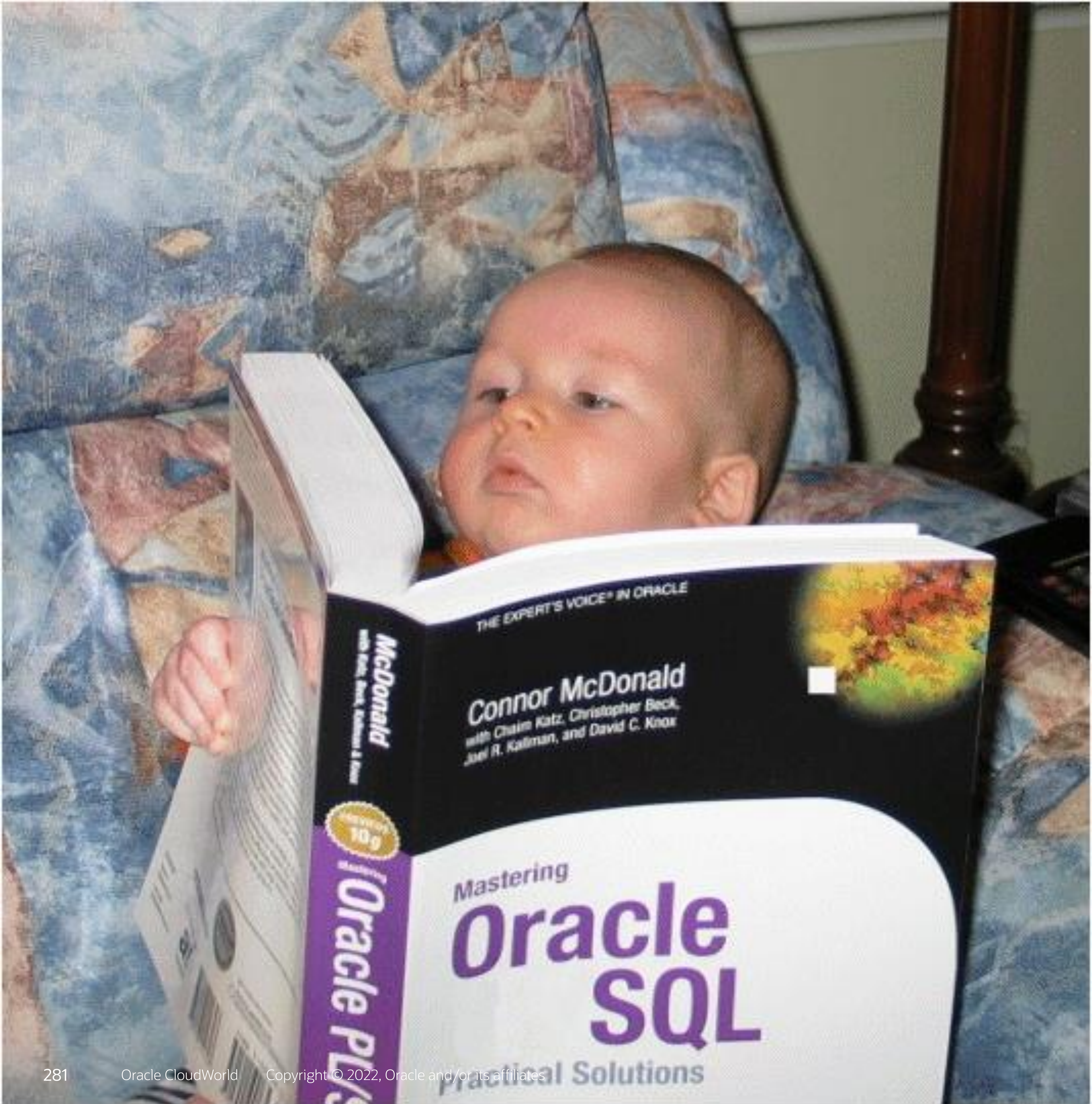
• grouping sets	23 years
• row_number() + analytics	23 years
• hypothetical analytics	23 years
• partitioned outer join	20 years
• WITH clause	20 years
• KEEP clause	20 years
• MODEL clause	18 years
• recursive WITH	12 years
• MATCH_RECOGNIZE	7 years
• SQL Macros	2 years

robust, powerful

you write less code

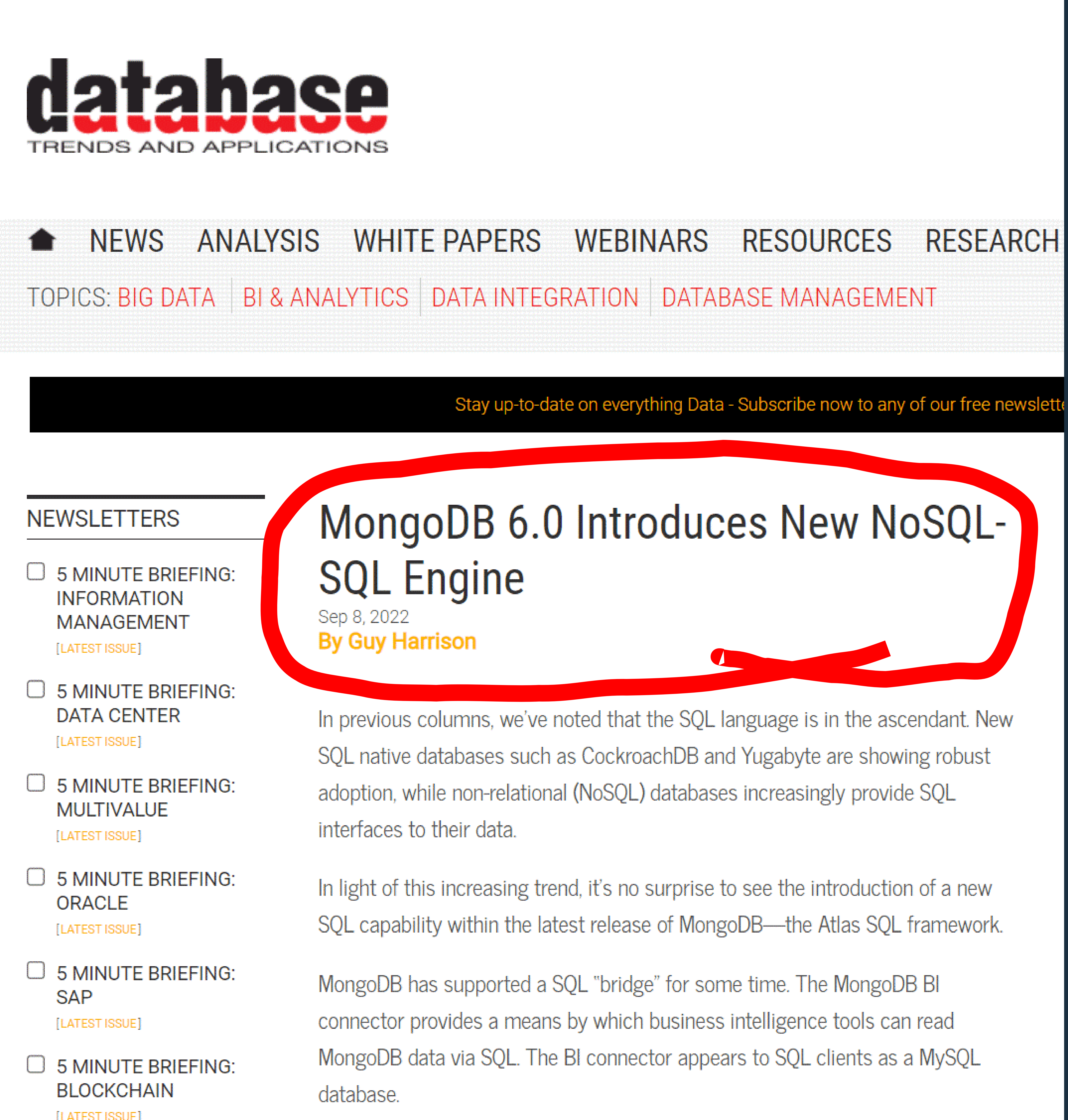
faster, scalable apps

never too early to start


THE EXPERT'S VOICE® IN ORACLE
Connor McDonald
with Chaim Katz, Christopher Beck, Joel R. Kallman, and David C. Knox
Mastering
Oracle
SQL
Practical Solutions
McDonald
Oracle PL/S

Son #1 ☺

never too late to start ☺



database
TRENDS AND APPLICATIONS

NEWS ANALYSIS WHITE PAPERS WEBINARS RESOURCES RESEARCH

TOPICS: BIG DATA | BI & ANALYTICS | DATA INTEGRATION | DATABASE MANAGEMENT

Stay up-to-date on everything Data - Subscribe now to any of our free newslette

NEWSLETTERS

□ 5 MINUTE BRIEFING: INFORMATION MANAGEMENT [LATEST ISSUE]

□ 5 MINUTE BRIEFING: DATA CENTER [LATEST ISSUE]

□ 5 MINUTE BRIEFING: MULTIVALUE [LATEST ISSUE]

□ 5 MINUTE BRIEFING: ORACLE [LATEST ISSUE]

□ 5 MINUTE BRIEFING: SAP [LATEST ISSUE]

□ 5 MINUTE BRIEFING: BLOCKCHAIN [LATEST ISSUE]

MongoDB 6.0 Introduces New NoSQL-SQL Engine

Sep 8, 2022

By Guy Harrison

In previous columns, we've noted that the SQL language is in the ascendant. New SQL native databases such as CockroachDB and Yugabyte are showing robust adoption, while non-relational (NoSQL) databases increasingly provide SQL interfaces to their data.

In light of this increasing trend, it's no surprise to see the introduction of a new SQL capability within the latest release of MongoDB—the Atlas SQL framework.

MongoDB has supported a SQL "bridge" for some time. The MongoDB BI connector provides a means by which business intelligence tools can read MongoDB data via SQL. The BI connector appears to SQL clients as a MySQL database.

prodigal ☺ Son

your turn!

- livelabs.oracle.com

  (hands on labs)

- bit.ly/database4dev

  (SQL beginners)

- livesql.oracle.com

  (SQL scratchpad)

- developer.oracle.com/free

  (so much great free stuff)

ORACLE
CloudWorld

Let's connect!

Connor McDonald, Database Advocate
https://linktr.ee/connor