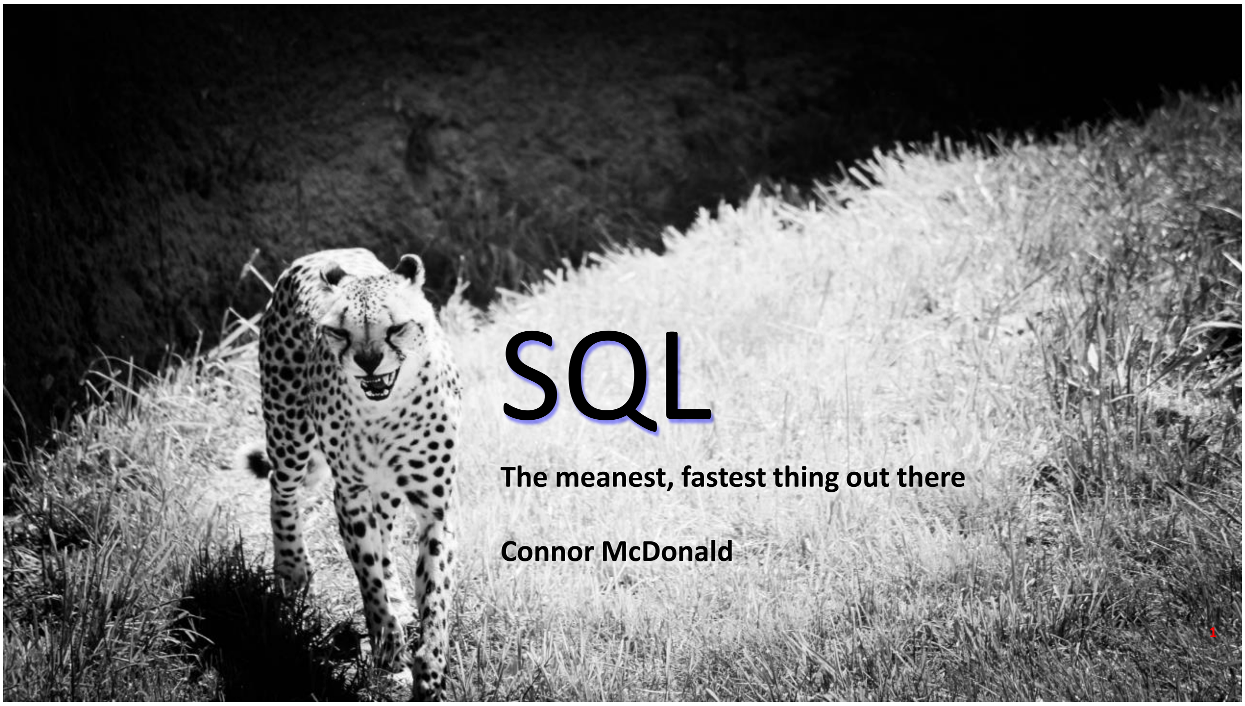


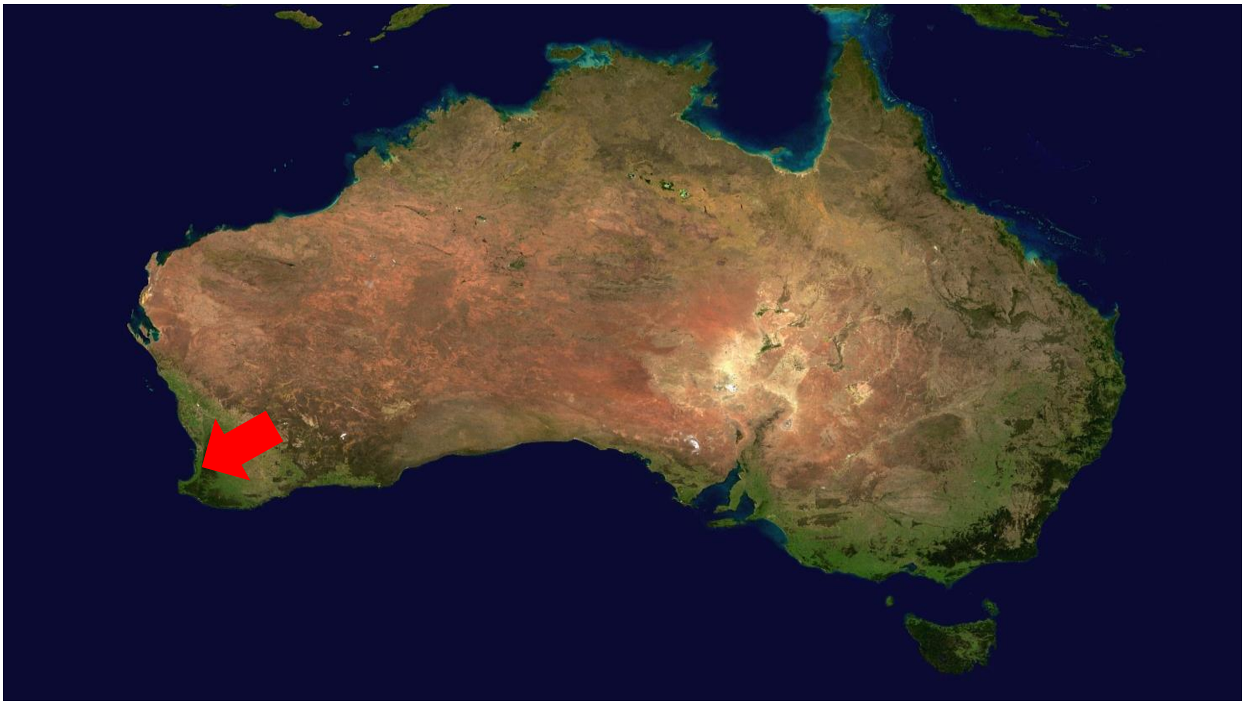
SQL

The meanest, fastest thing out there

Connor McDonald

Connor McDonald

3



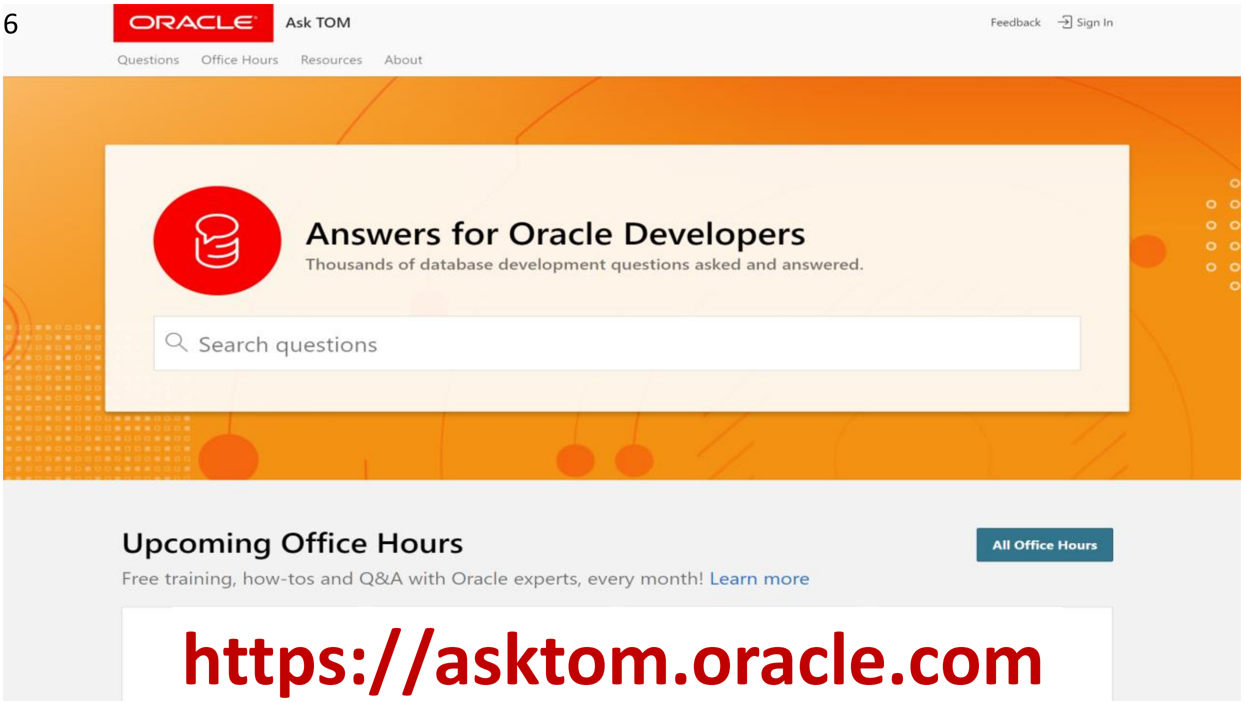
Typical speaker ego slide

Follow and Subscribe !

blog	connor-mcdonald.com
youtube	tinyurl.com/connor-tube
twitter	@connor_mc_d

ORACLE

6



ORACLE Ask TOM

Feedback Sign In

Questions Office Hours Resources About

Answers for Oracle Developers

Thousands of database development questions asked and answered.

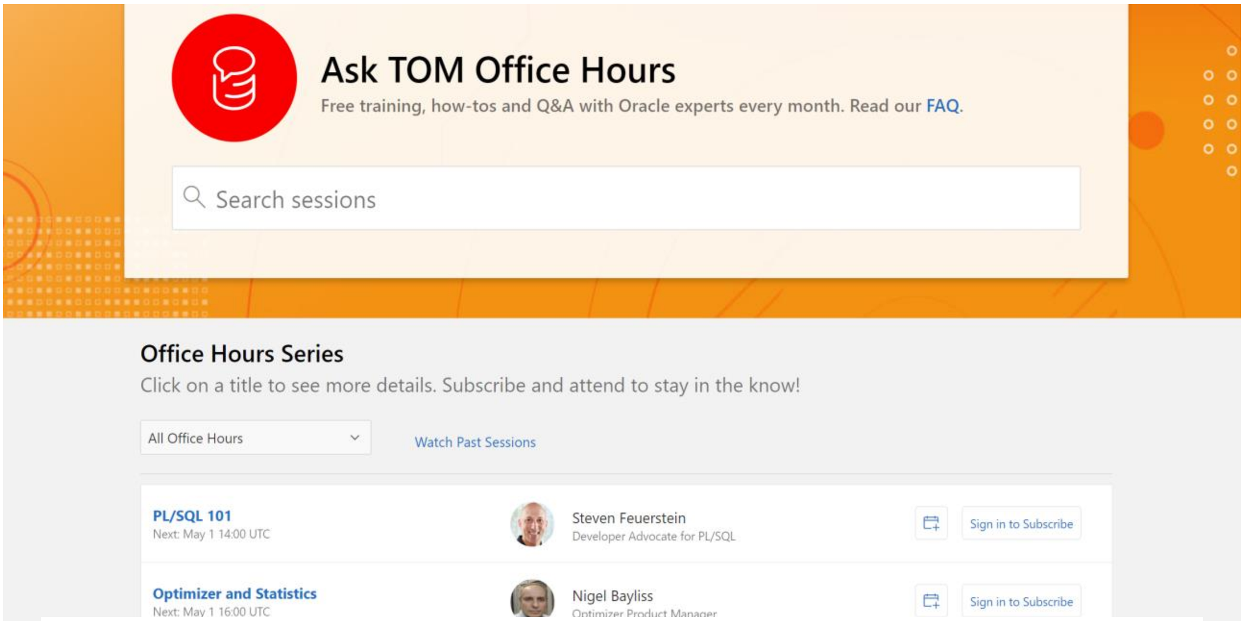
Search questions

Upcoming Office Hours

All Office Hours

Free training, how-tos and Q&A with Oracle experts, every month! Learn more

https://asktom.oracle.com



Ask TOM Office Hours

Free training, how-tos and Q&A with Oracle experts every month. Read our FAQ.

Office Hours Series

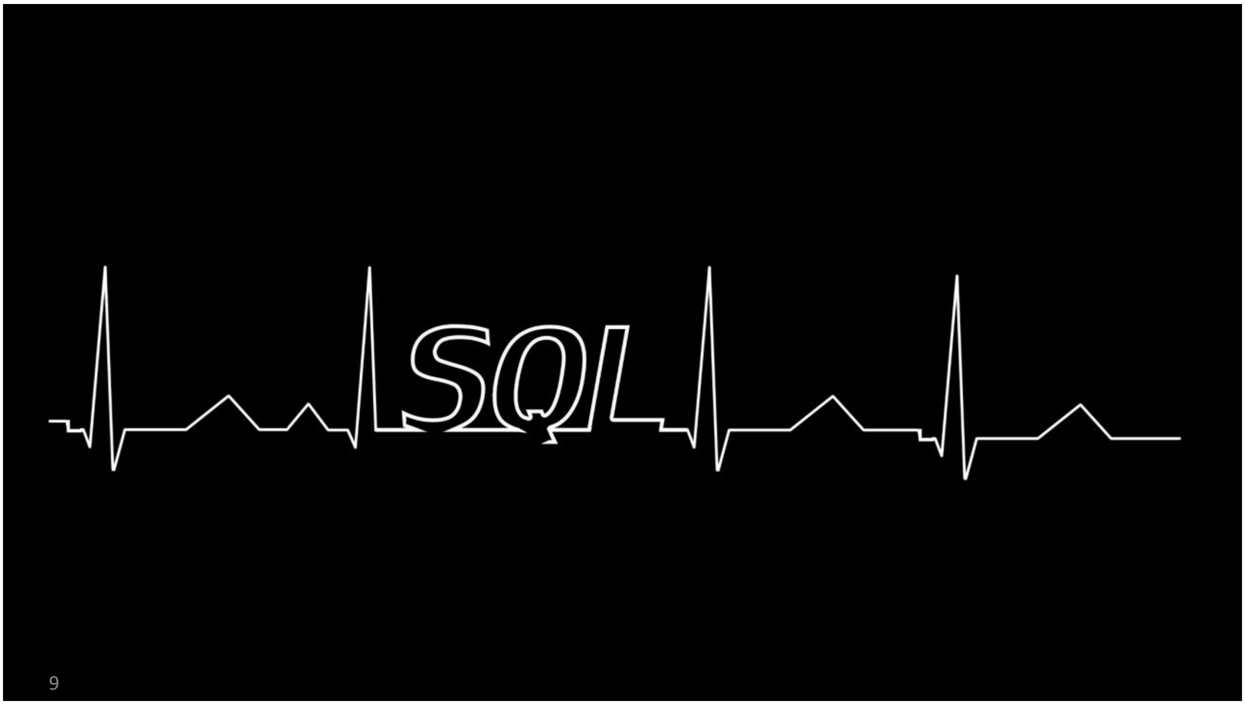
Click on a title to see more details. Subscribe and attend to stay in the know!

PL/SQL 101 Next: May 1 14:00 UTC	Steven Feuerstein Developer Advocate for PL/SQL		Sign in to Subscribe
Optimizer and Statistics Next: May 1 16:00 UTC	Nigel Bayliss Optimizer Product Manager		Sign in to Subscribe

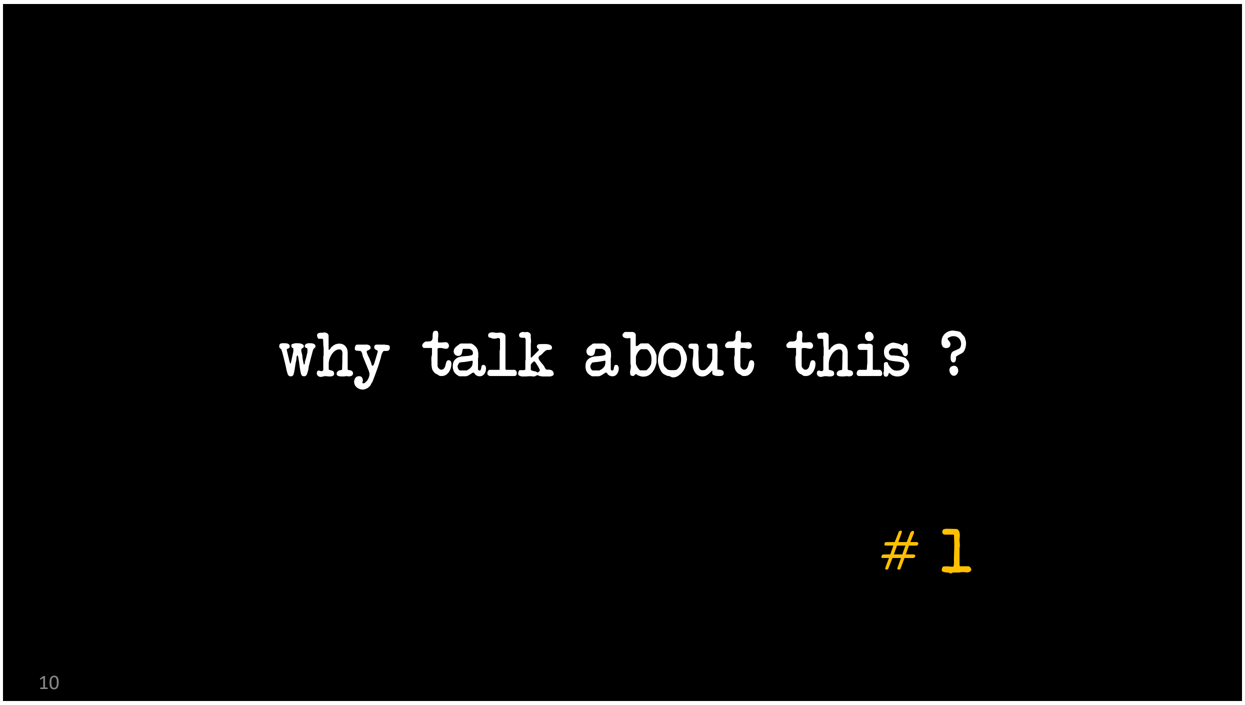
https://asktom.oracle.com/officehours

80 hours of free access / Q&A

since Feb



9



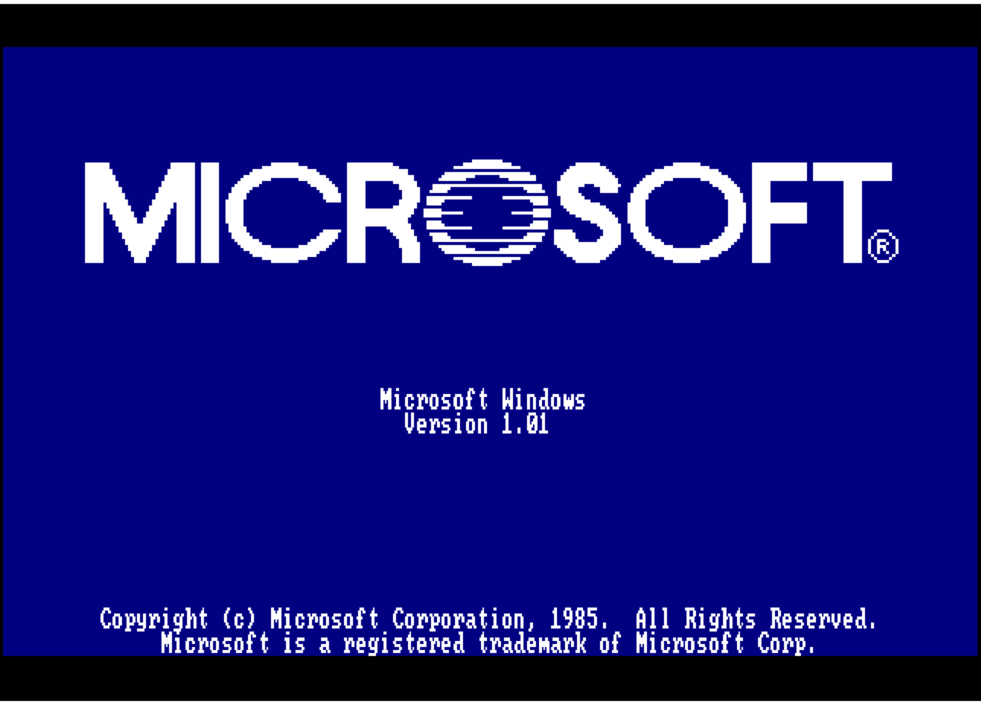
why talk about this ?

1

10

after all …

11



Microsoft Windows
Version 1.01

Copyright (c) Microsoft Corporation, 1985. All Rights Reserved.
Microsoft is a registered trademark of Microsoft Corp.

12

~~NoSQL~~

non relational

“….the complexity of dealing with a non-ACID data store in every part of our business logic would be too great, and there was simply no way our business could function without SQL queries.”

Google

“[Facebook] started in the Hadoop world. We are now bringing in relational to enhance that. … [we] realized that using the wrong technology for certain kinds of problems can be difficult.”

facebook

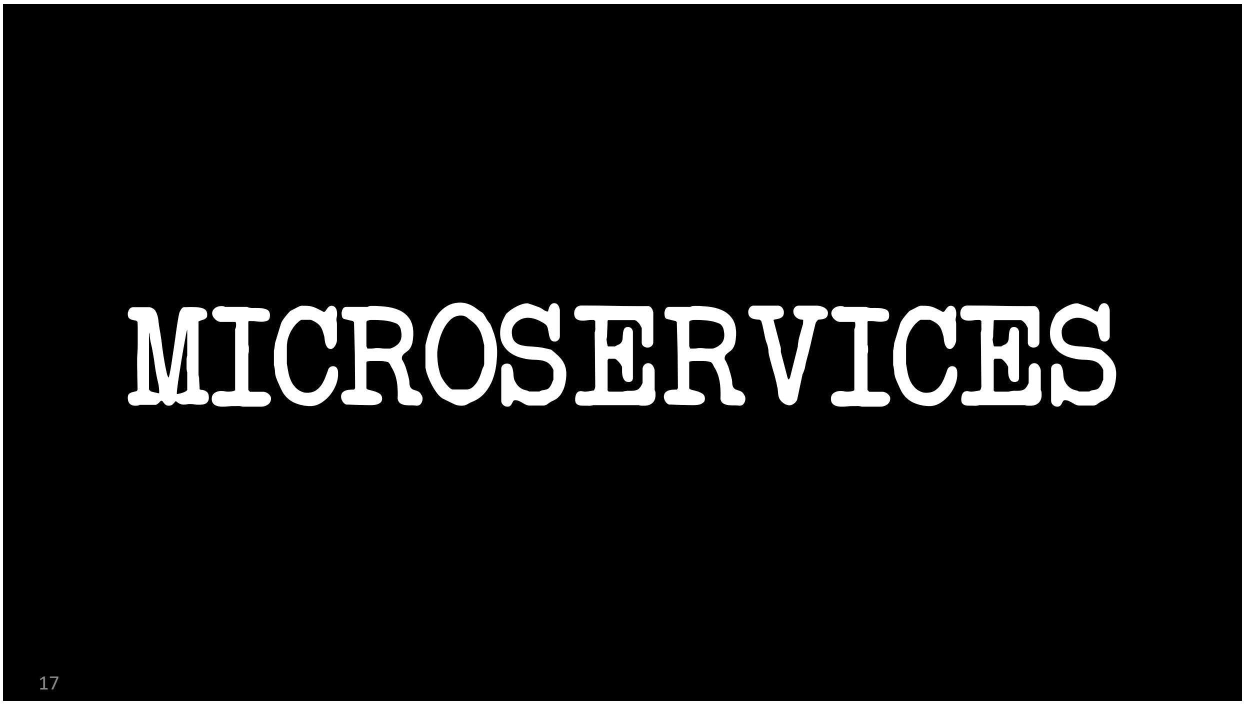
why talk about this

2

15

developers love cool stuff

16

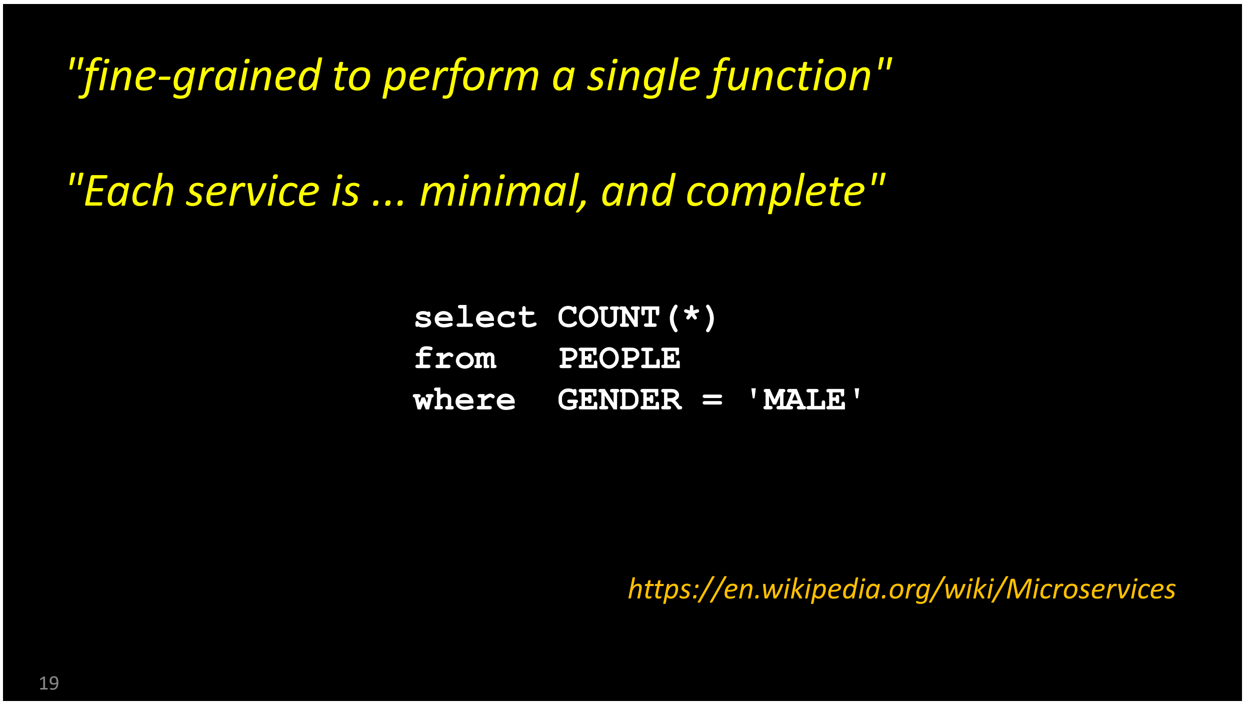


MICROSERVICES

17

SQL is microservices !

18



"fine-grained to perform a single function"

"Each service is ... minimal, and complete"

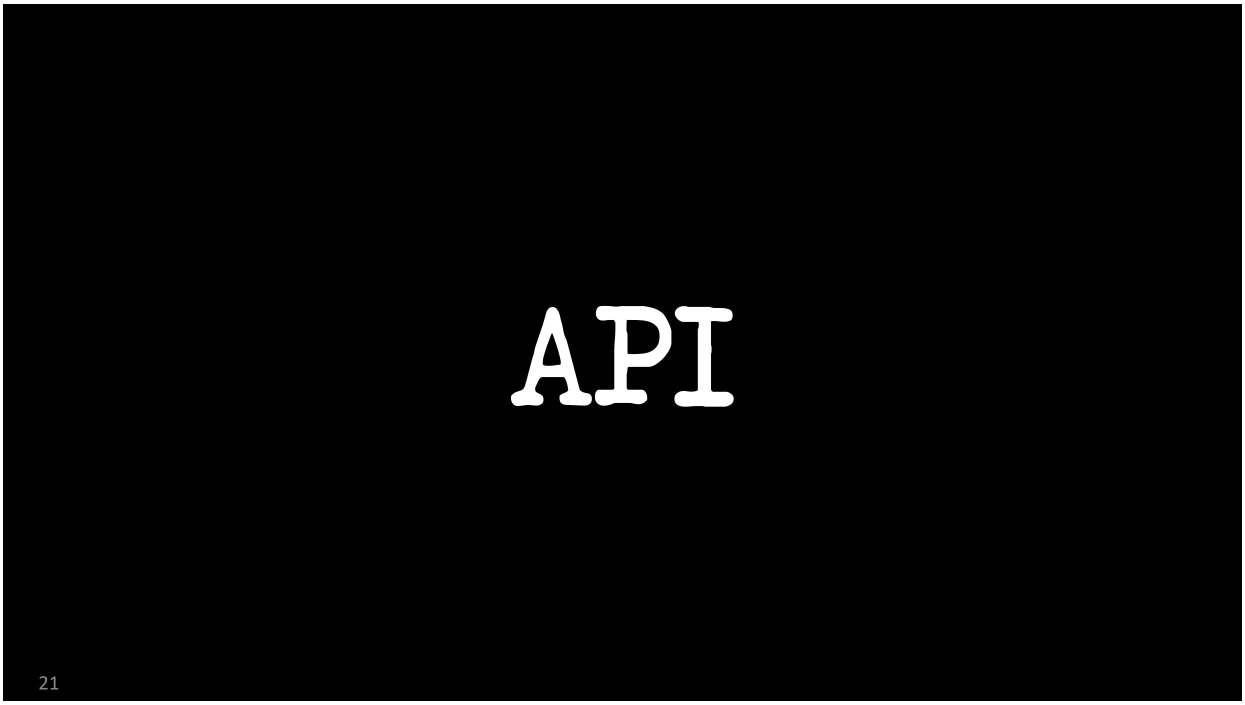
```
select COUNT(*)
from   PEOPLE
where  GENDER = 'MALE'
```

https://en.wikipedia.org/wiki/Microservices

19

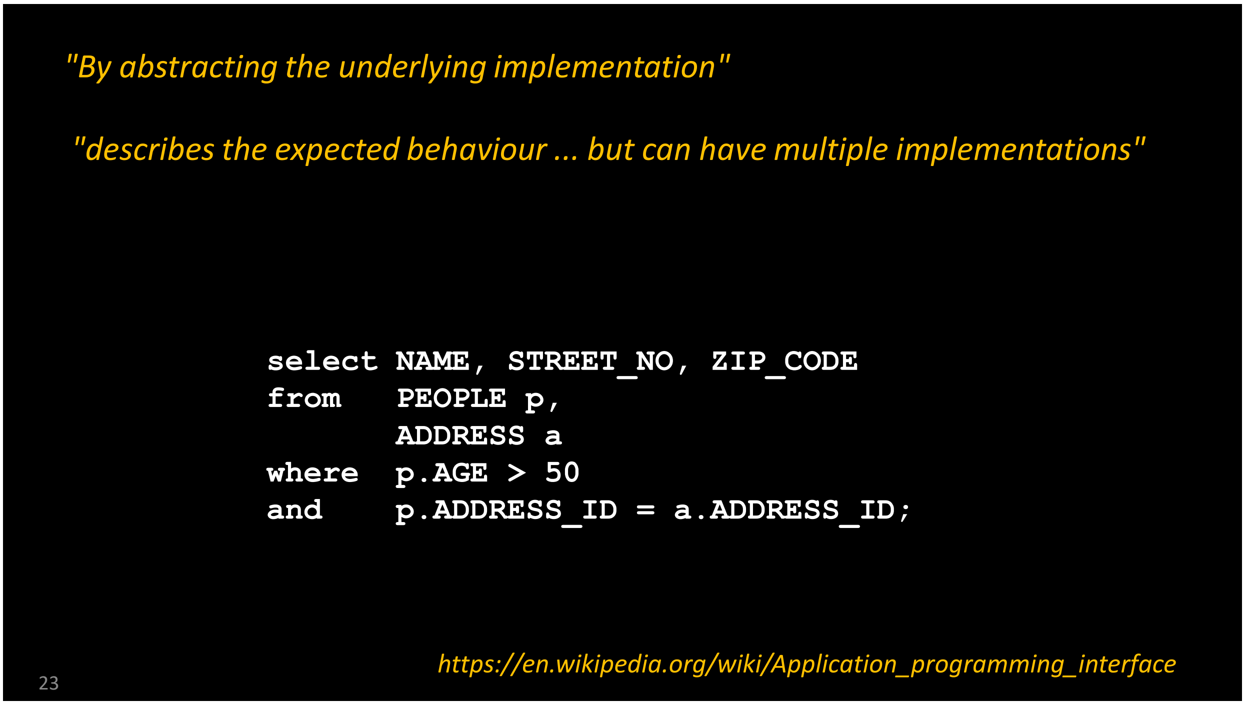
even cooler stuff

20



API

SQL is entirely APIs !



"By abstracting the underlying implementation"

"describes the expected behaviour ... but can have multiple implementations"

```
select NAME, STREET_NO, ZIP_CODE
from   PEOPLE p,
       ADDRESS a
where  p.AGE > 50
and    p.ADDRESS_ID = a.ADDRESS_ID;
```

https://en.wikipedia.org/wiki/Application_programming_interface

23

key point

24

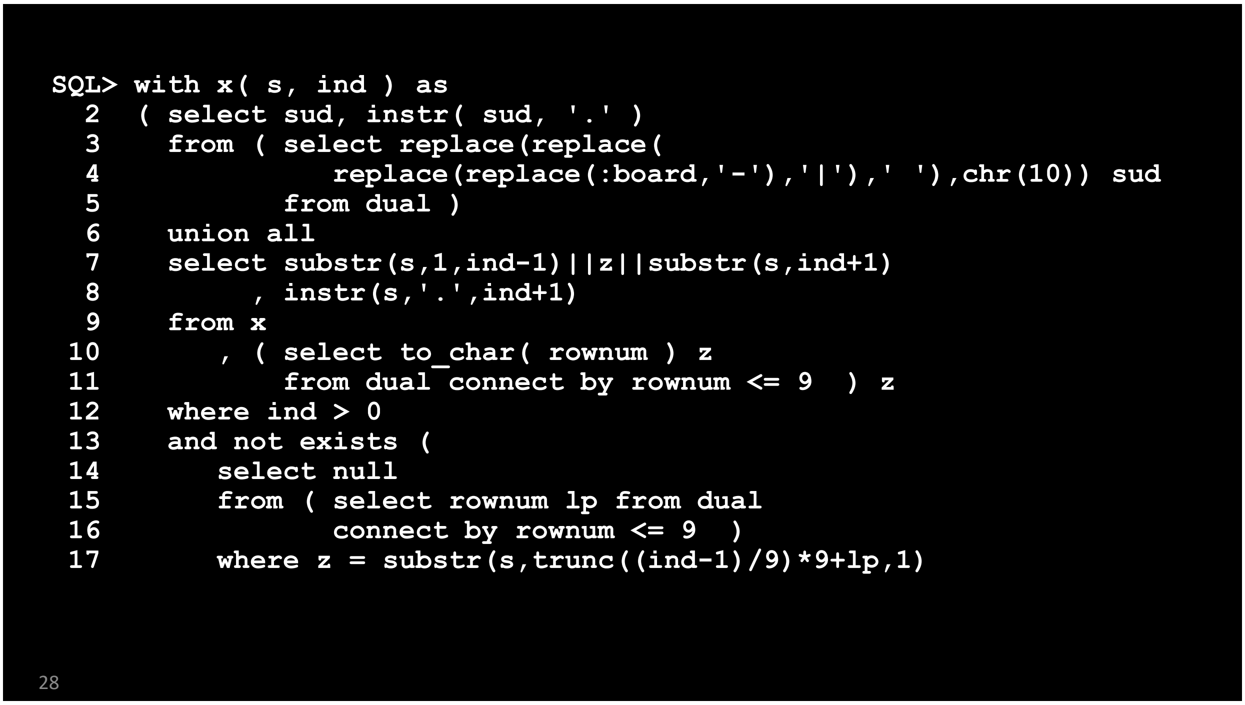
this session is **not** about …

25

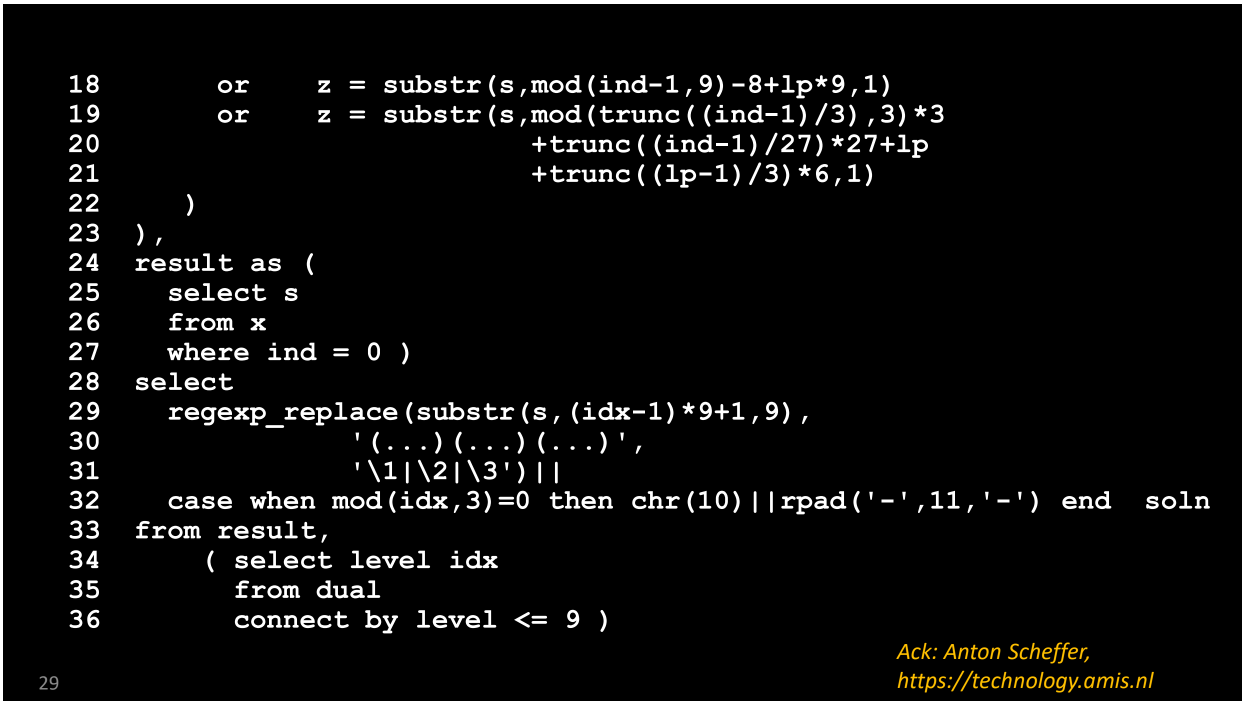
being a ~~smart~~-ass

26

we can do anything …

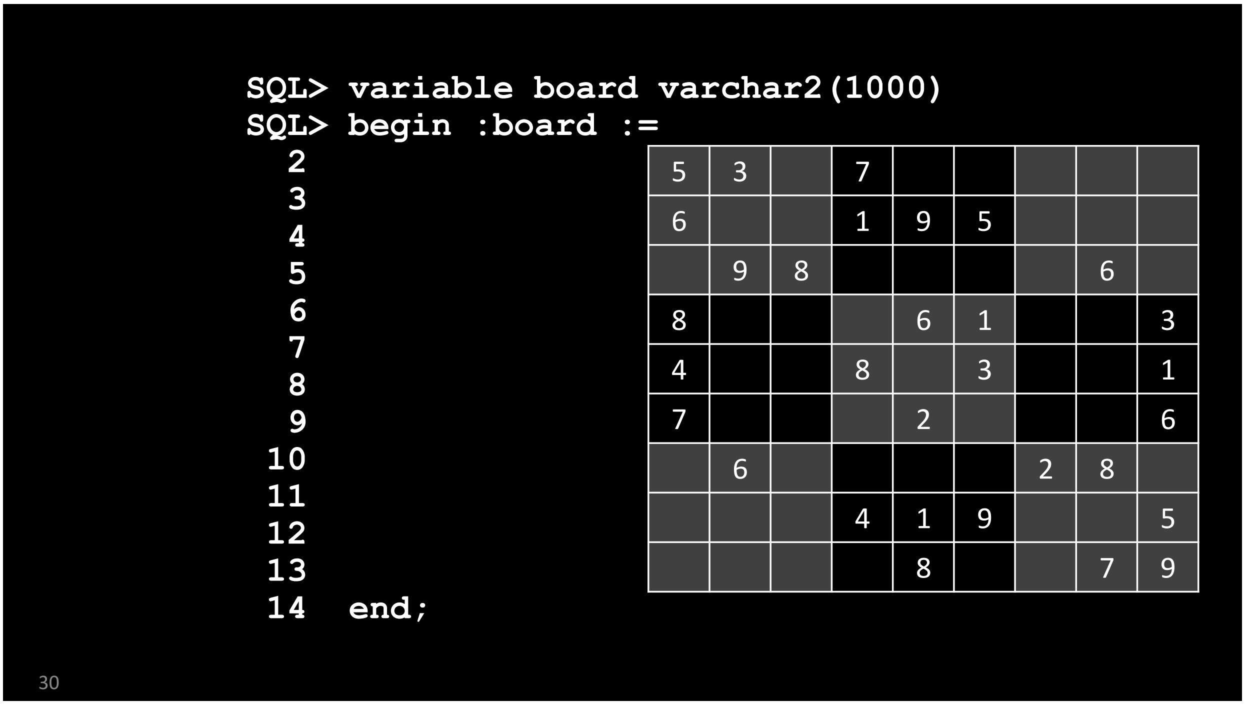


```
SQL> with x( s, ind ) as
  2  ( select sud, instr( sud, '.' )
  3    from ( select replace(replace(
  4                replace(replace(:board,'-'),'|'),' '),chr(10)) sud
  5           from dual )
  6    union all
  7    select substr(s,1,ind-1)||z||substr(s,ind+1)
  8         , instr(s,'.',ind+1)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



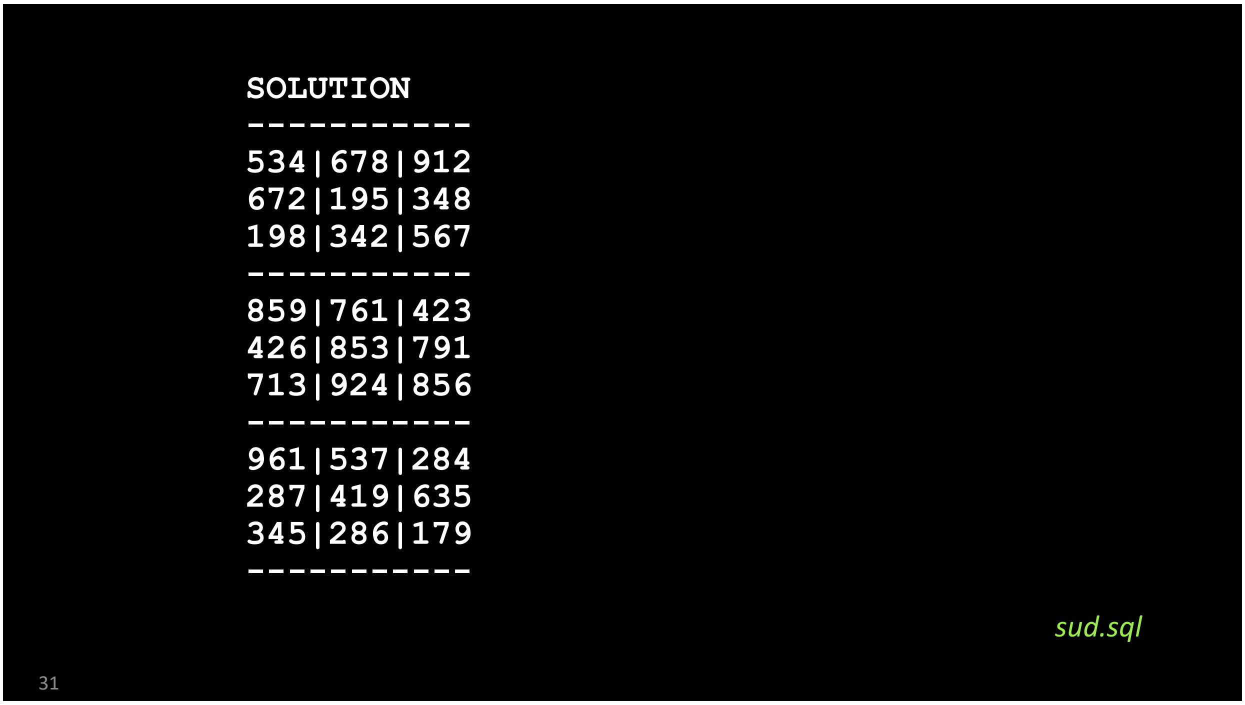
```
18       or    z = substr(s,mod(ind-1,9)-8+lp*9,1)
19       or    z = substr(s,mod(trunc((ind-1)/3),3)*3
20                      +trunc((ind-1)/27)*27+lp
21                      +trunc((lp-1)/3)*6,1)
22     )
23  ),
24  result as (
25    select s
26    from x
27    where ind = 0 )
28  select
29 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

Ack: Anton Scheffer,
https://technology.amis.nl



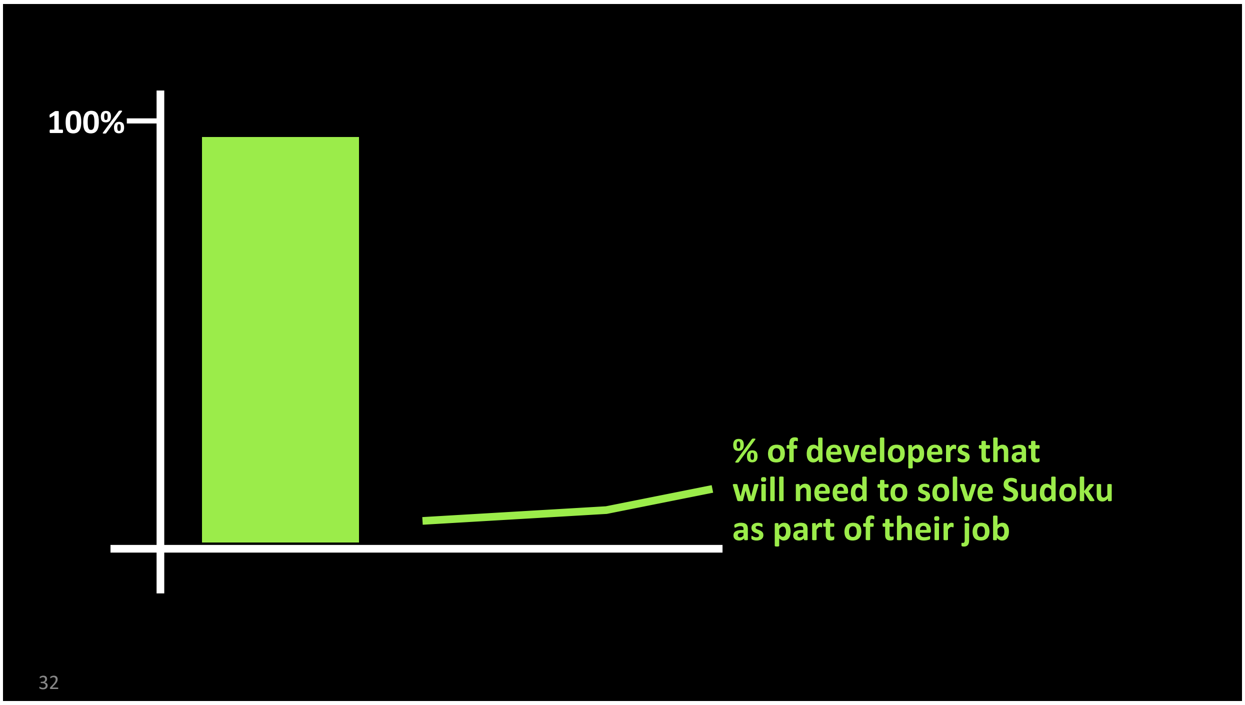
```
SQL> variable board varchar2(1000)
SQL> begin :board :=
  2
  3
  4
  5
  6
  7
  8
  9
 10
 11
 12
 13
 14  end;
```

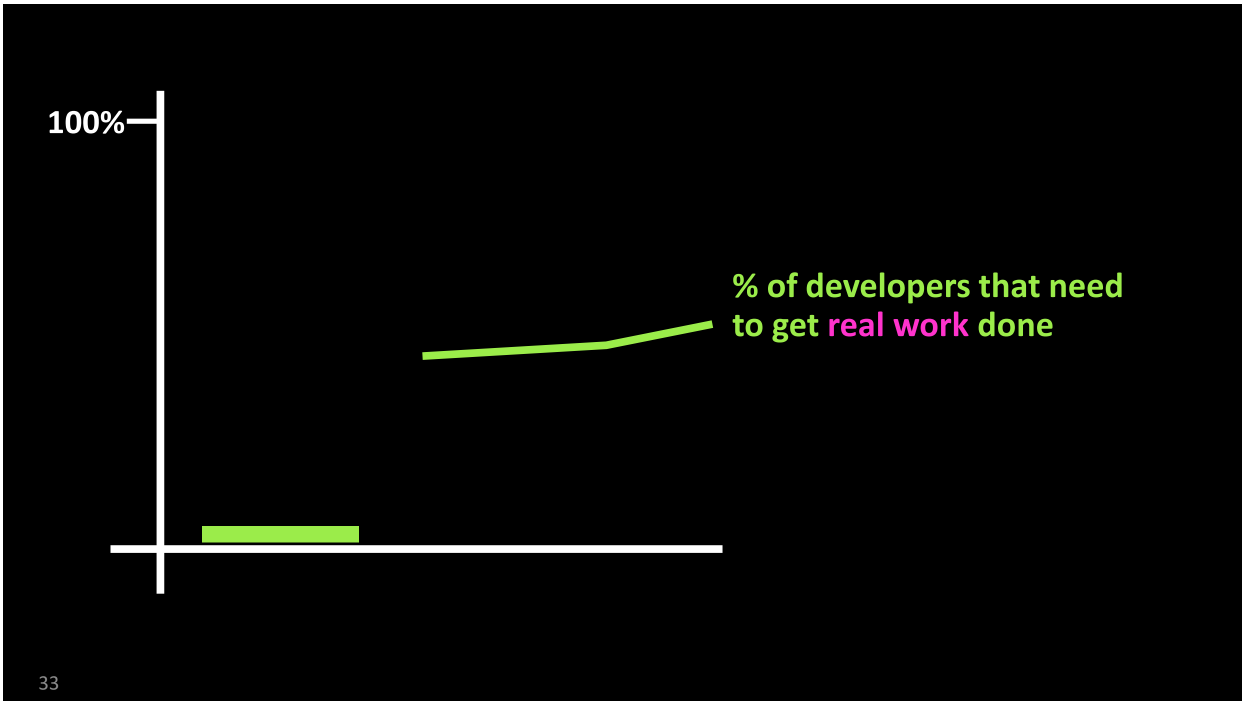
5	3		7					
6			1	9	5			
	9	8					6	
8				6	1			3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9



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

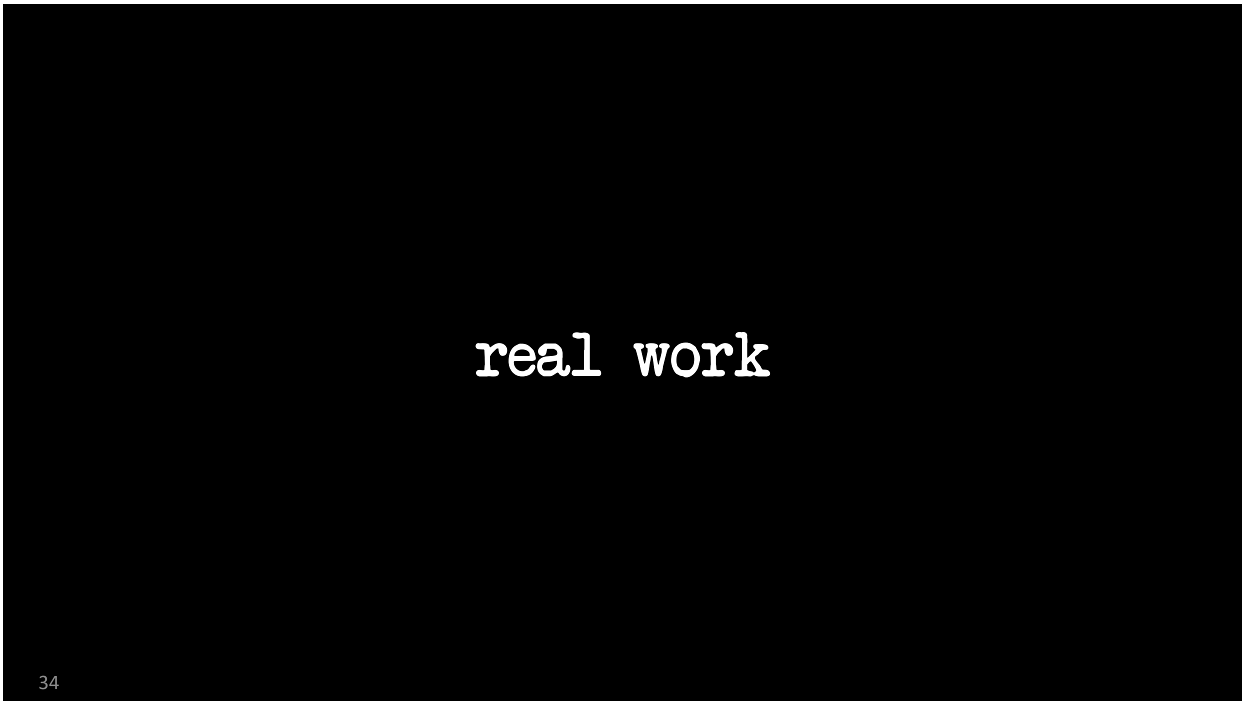
sud.sql



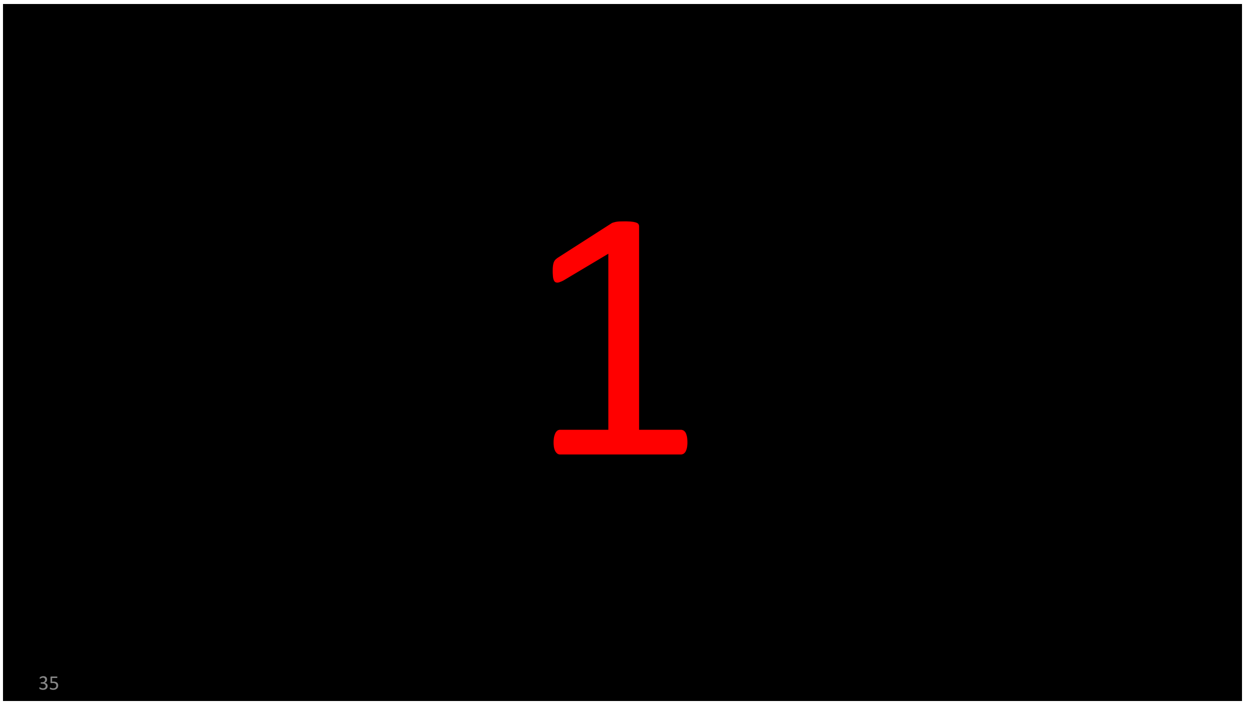


% of developers that need to get real work done

33


real work

34



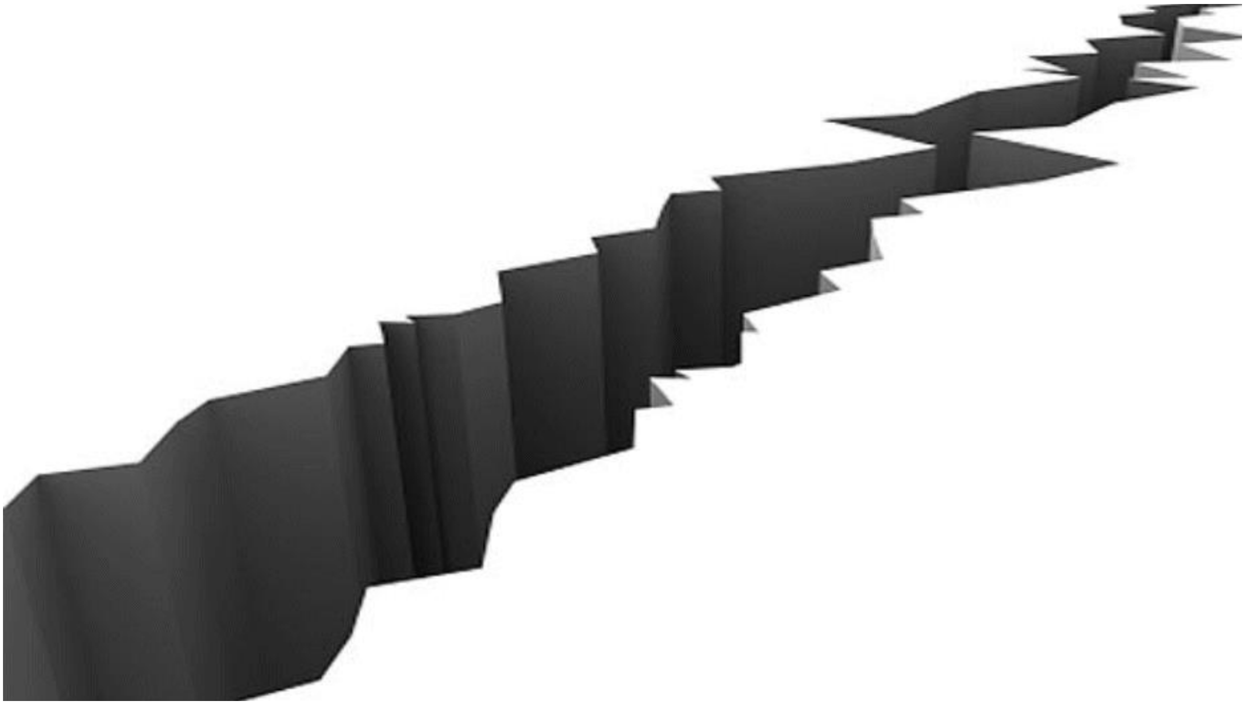
1

35

some controversy…

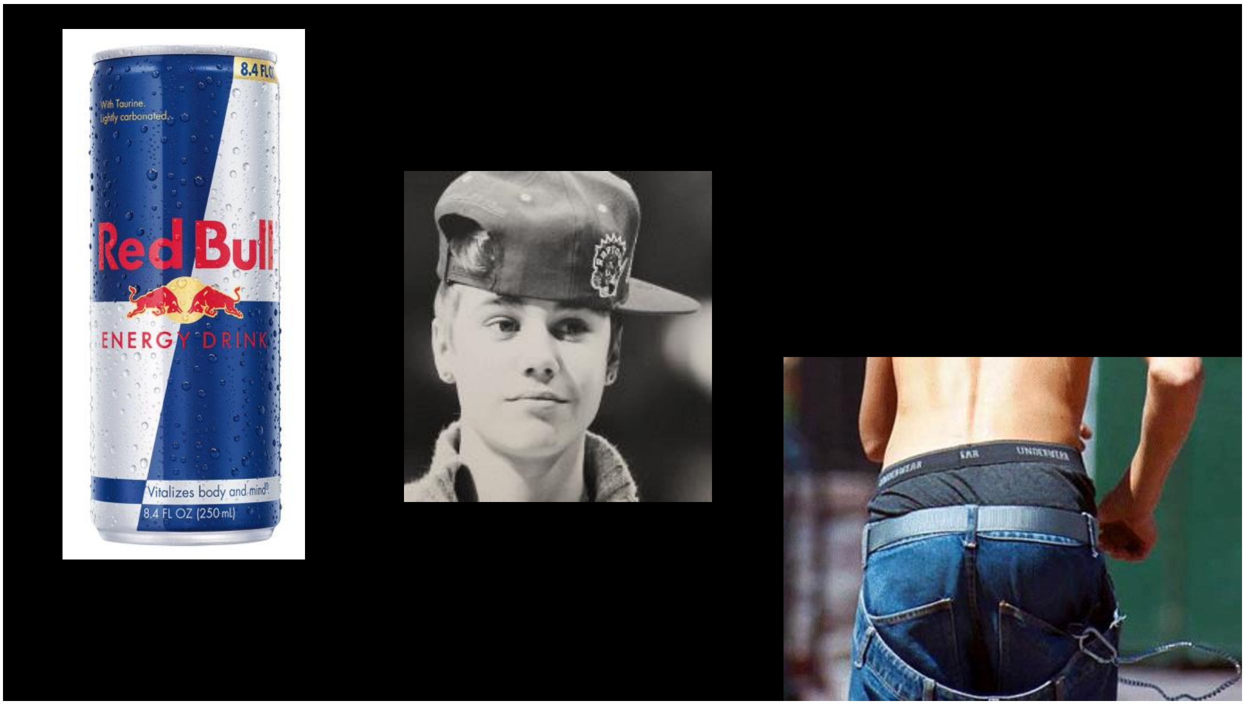
36

37



SHERRY
Nº17

39


8.4 FL OZ
With Taurine.
Lightly carbonated.
Red Bull
ENERGY DRINK
Vitalizes body and mind.
8.4 FL OZ (250 ml)

DBA friendship is important

oc00.sql

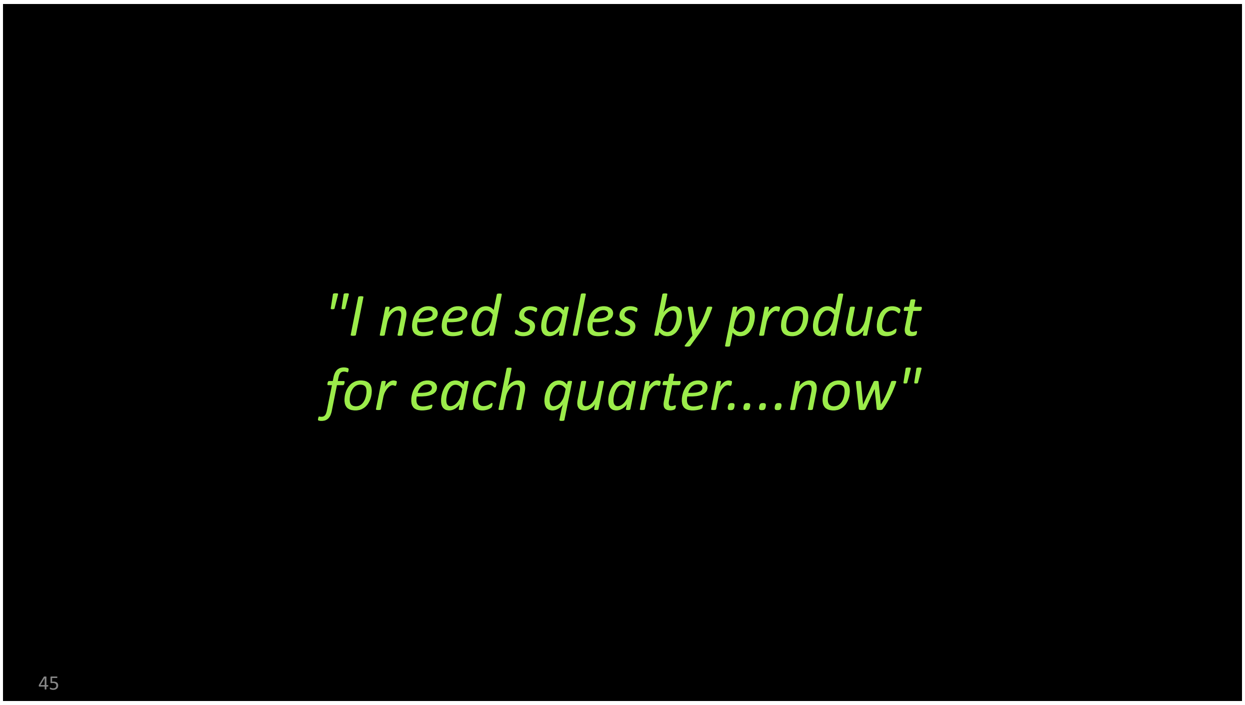
2

transposition

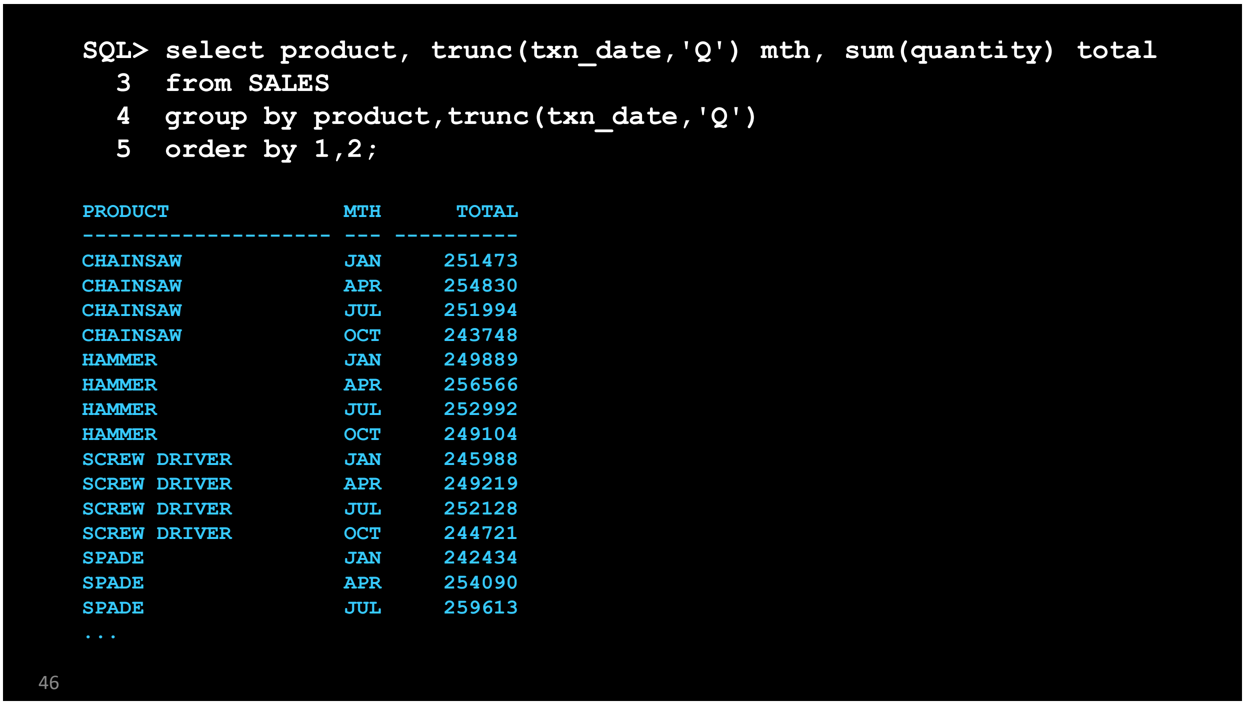
43

rows to columns, colums to rows

44



"I need sales by product for each quarter....now"

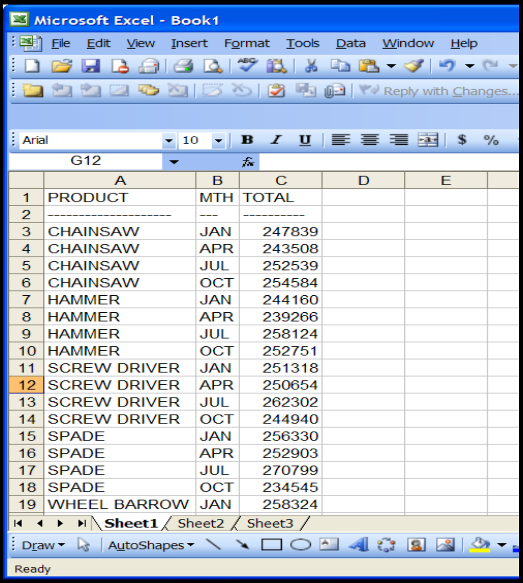


```
SQL> select product, trunc(txn_date,'Q') mth, sum(quantity) total
  3  from SALES
  4  group by product,trunc(txn_date,'Q')
  5  order by 1,2;

PRODUCT              MTH      TOTAL
-------------------- --- ----------
CHAINSAW             JAN     251473
CHAINSAW             APR     254830
CHAINSAW             JUL     251994
CHAINSAW             OCT     243748
HAMMER               JAN     249889
HAMMER               APR     256566
HAMMER               JUL     252992
HAMMER               OCT     249104
SCREW DRIVER         JAN     245988
SCREW DRIVER         APR     249219
SCREW DRIVER         JUL     252128
SCREW DRIVER         OCT     244721
SPADE                JAN     242434
SPADE                APR     254090
SPADE                JUL     259613
...
```

47

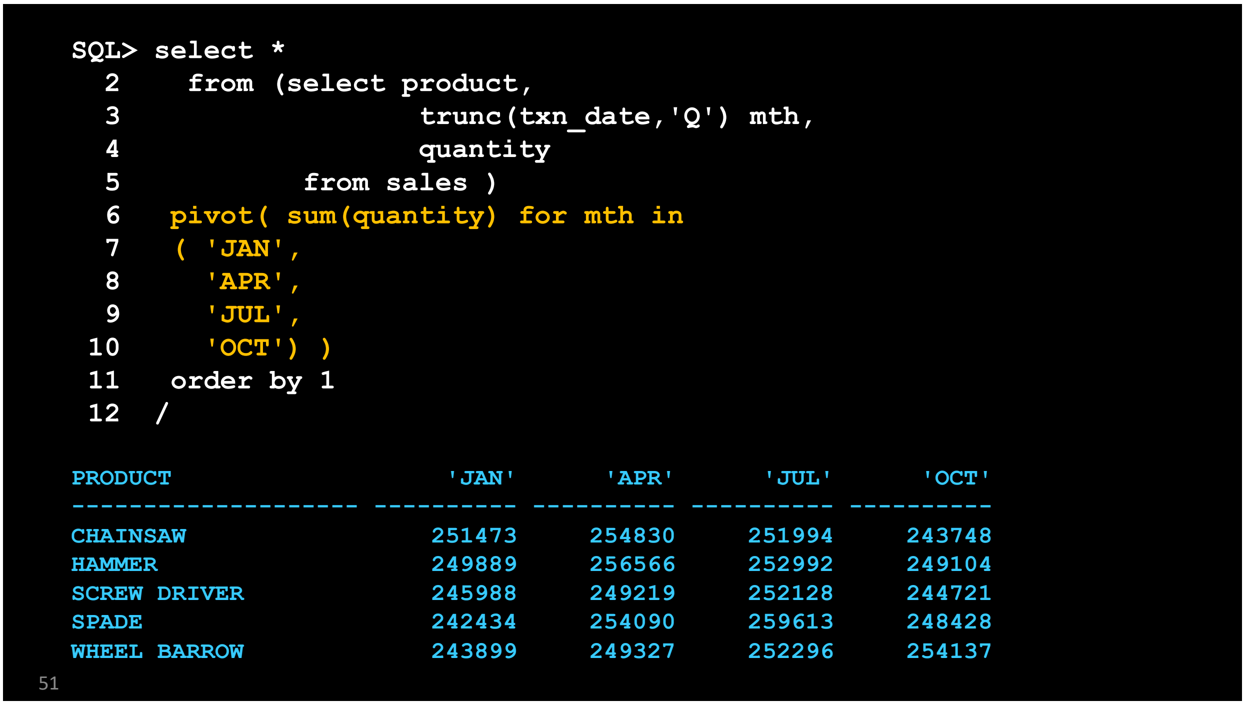
"That’s wrong!...surely you know I wanted it ACROSS the page"

48

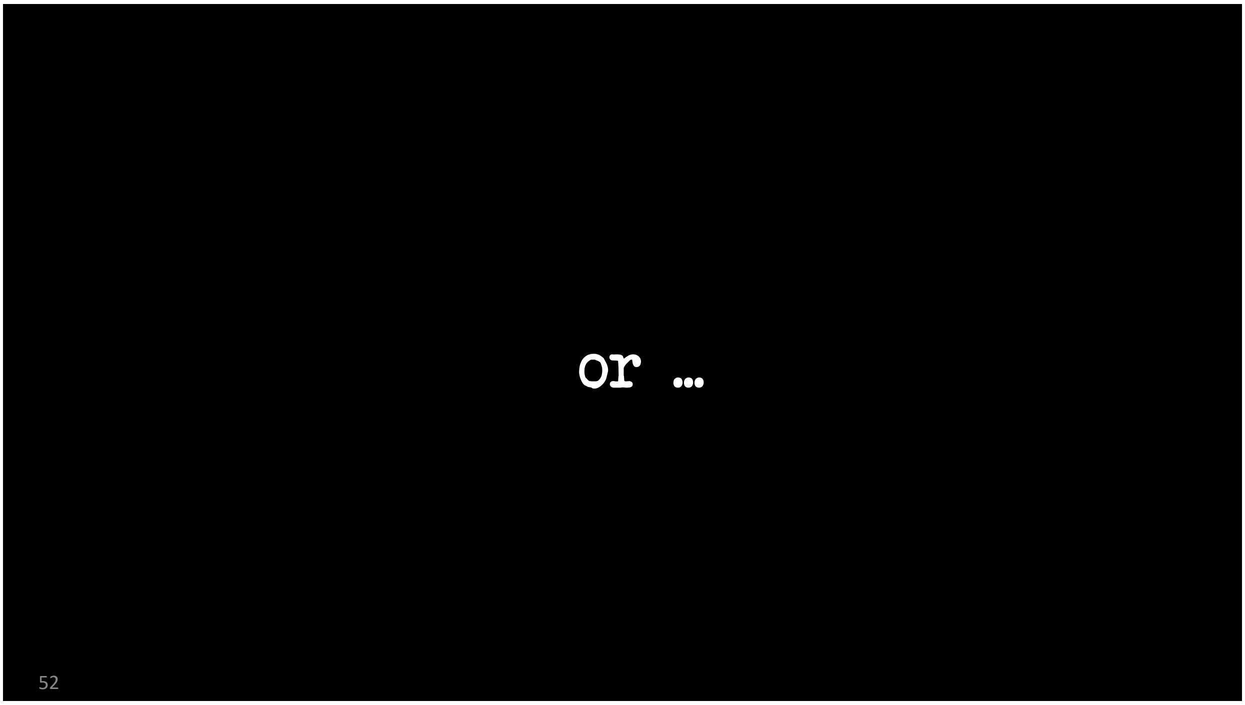


	A	B	C	D	E
1	PRODUCT	MTH	TOTAL		
2	---------------------	---	----------		
3	CHAINSAW	JAN	247839		
4	CHAINSAW	APR	243508		
5	CHAINSAW	JUL	252539		
6	CHAINSAW	OCT	254584		
7	HAMMER	JAN	244160		
8	HAMMER	APR	239266		
9	HAMMER	JUL	258124		
10	HAMMER	OCT	252751		
11	SCREW DRIVER	JAN	251318		
12	SCREW DRIVER	APR	250654		
13	SCREW DRIVER	JUL	262302		
14	SCREW DRIVER	OCT	244940		
15	SPADE	JAN	256330		
16	SPADE	APR	252903		
17	SPADE	JUL	270799		
18	SPADE	OCT	234545		
19	WHEEL BARROW	JAN	258324		

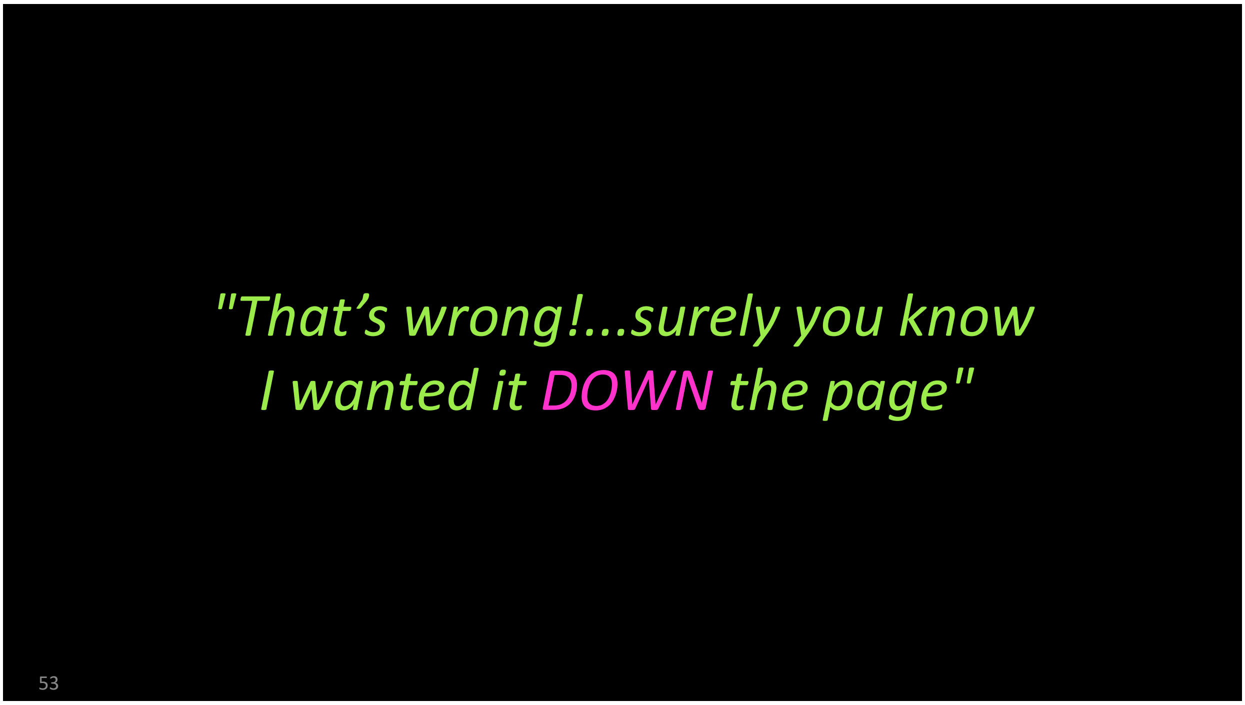
pivot clause



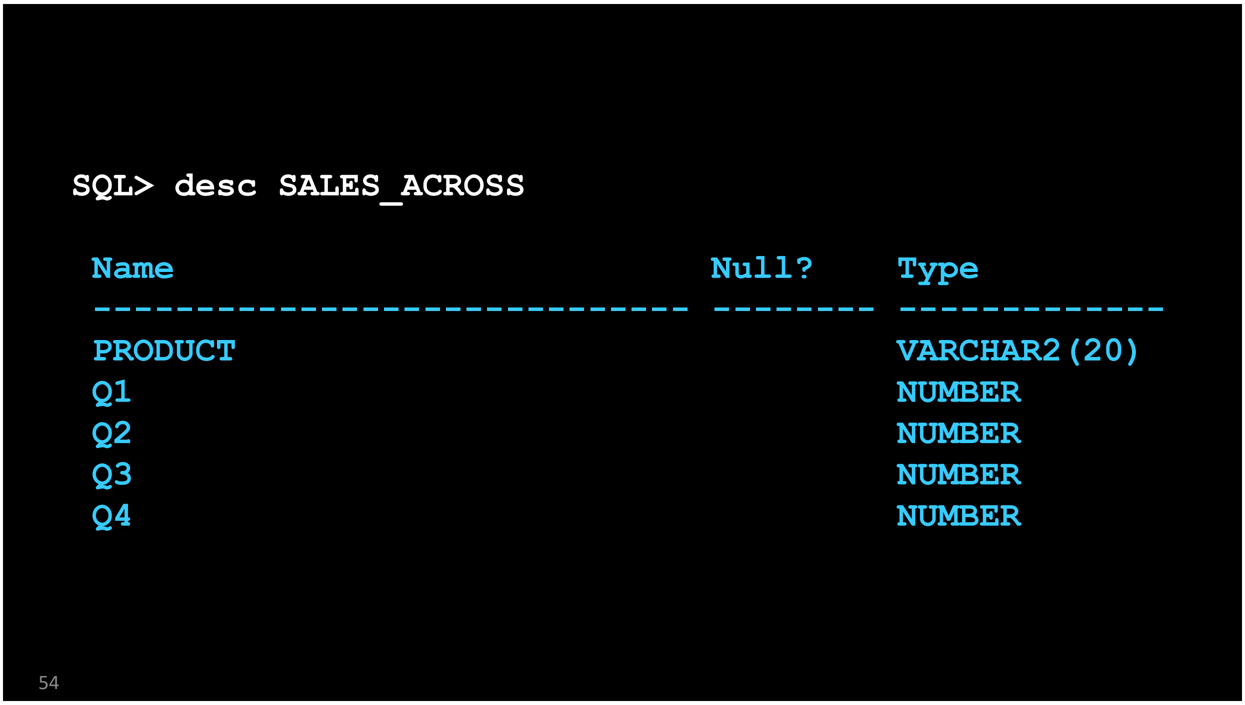
```
SQL> select *
  2    from (select product,
  3                  trunc(txn_date,'Q') mth,
  4                  quantity
  5           from sales )
  6   pivot( sum(quantity) for mth in
  7   ( 'JAN',
  8     'APR',
  9     'JUL',
 10     'OCT') )
 11   order by 1
 12  /

PRODUCT                   'JAN'      'APR'      'JUL'      'OCT'
-------------------- ---------- ---------- ---------- ----------
CHAINSAW                 251473     254830     251994     243748
HAMMER                   249889     256566     252992     249104
SCREW DRIVER             245988     249219     252128     244721
SPADE                    242434     254090     259613     248428
WHEEL BARROW             243899     249327     252296     254137
```



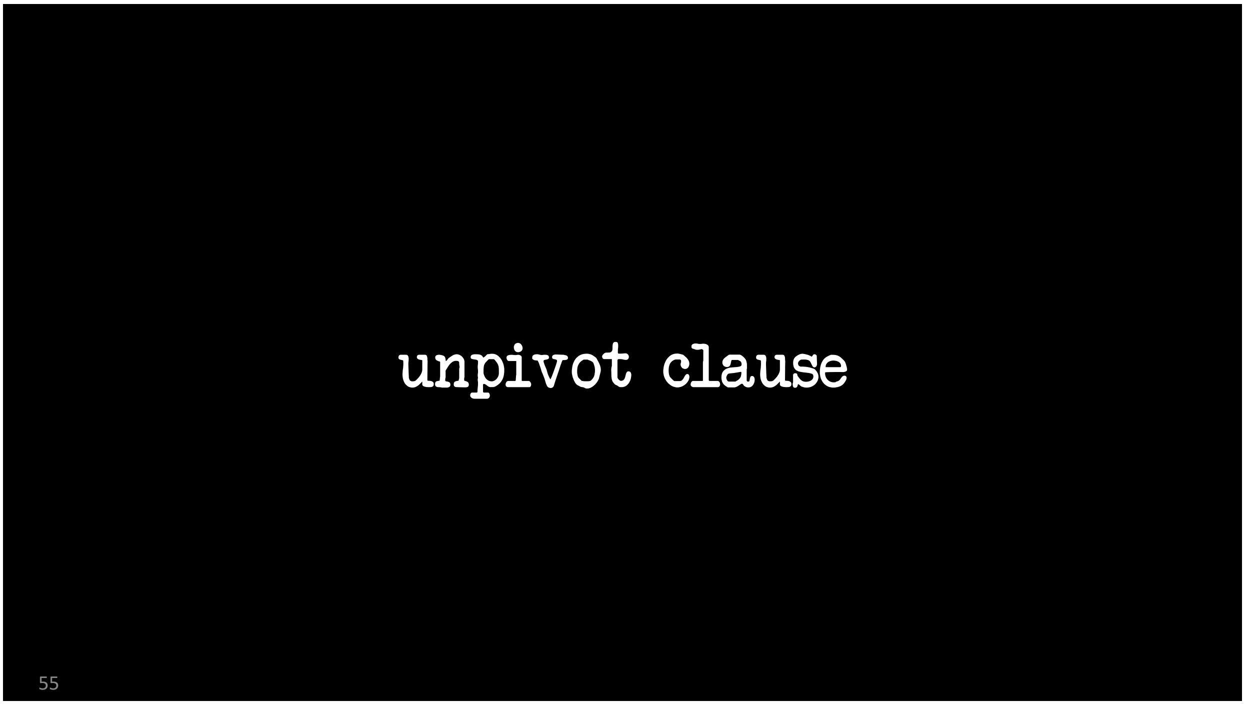
or …



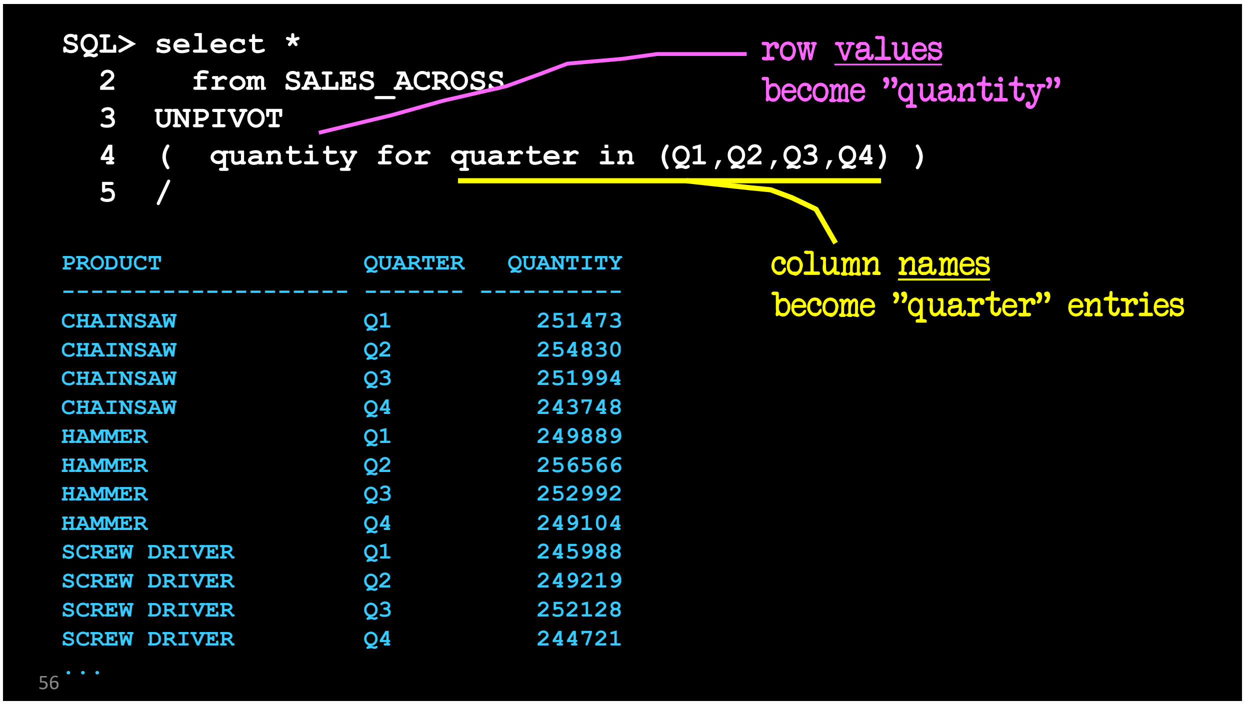
"That's wrong!...surely you know
I wanted it DOWN the page"



```
SQL> desc SALES_ACROSS

 Name                            Null?    Type
 ------------------------------- -------- -------------
 PRODUCT                                  VARCHAR2(20)
 Q1                                       NUMBER
 Q2                                       NUMBER
 Q3                                       NUMBER
 Q4                                       NUMBER
```



unpivot clause

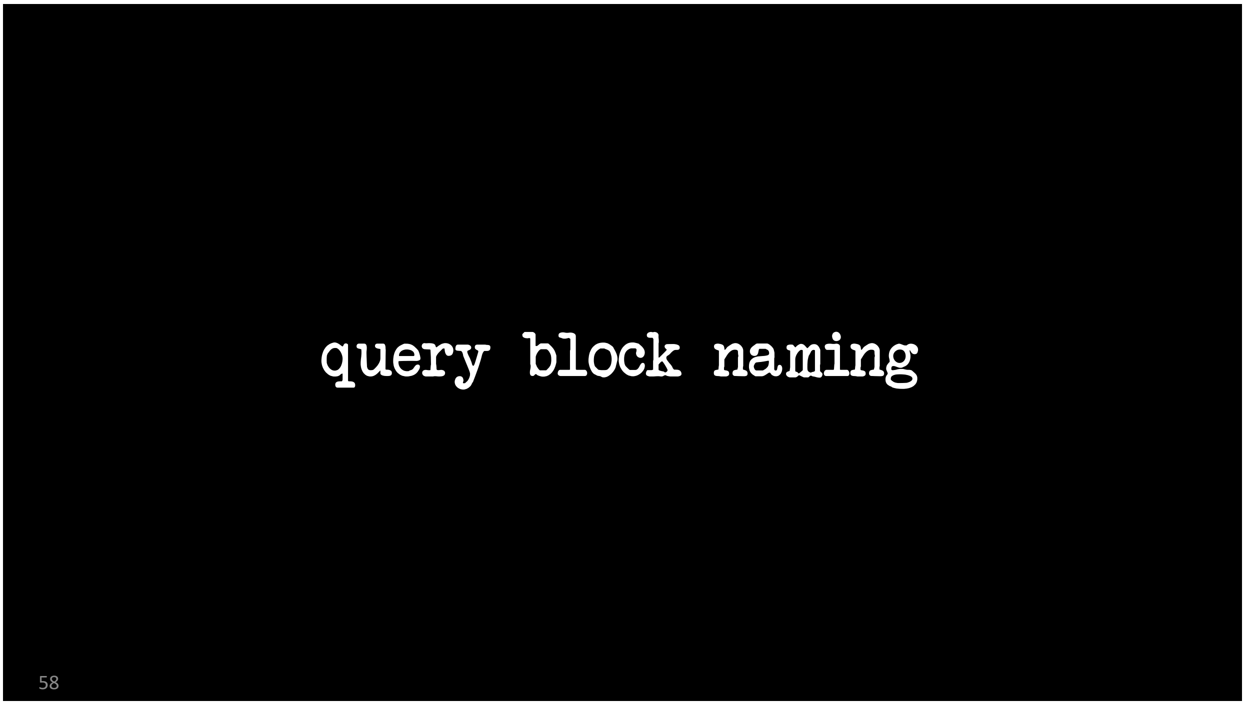


```
SQL> select *
  2    from SALES_ACROSS
  3  UNPIVOT
  4  (  quantity for quarter in (Q1,Q2,Q3,Q4)  )
  5  /

PRODUCT              QUARTER   QUANTITY
-------------------- ------- ----------
CHAINSAW             Q1          251473
CHAINSAW             Q2          254830
CHAINSAW             Q3          251994
CHAINSAW             Q4          243748
HAMMER               Q1          249889
HAMMER               Q2          256566
HAMMER               Q3          252992
HAMMER               Q4          249104
SCREW DRIVER         Q1          245988
SCREW DRIVER         Q2          249219
SCREW DRIVER         Q3          252128
SCREW DRIVER         Q4          244721
...
```

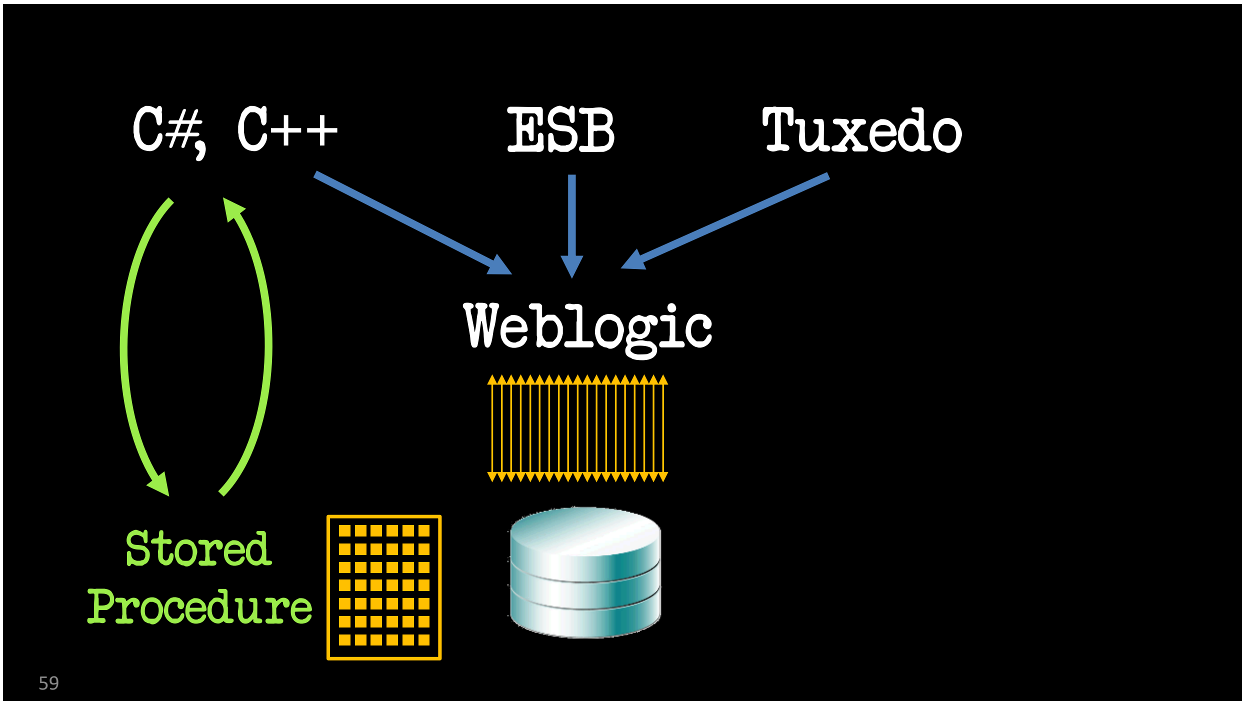
row values
become "quantity"

column names
become "quarter" entries

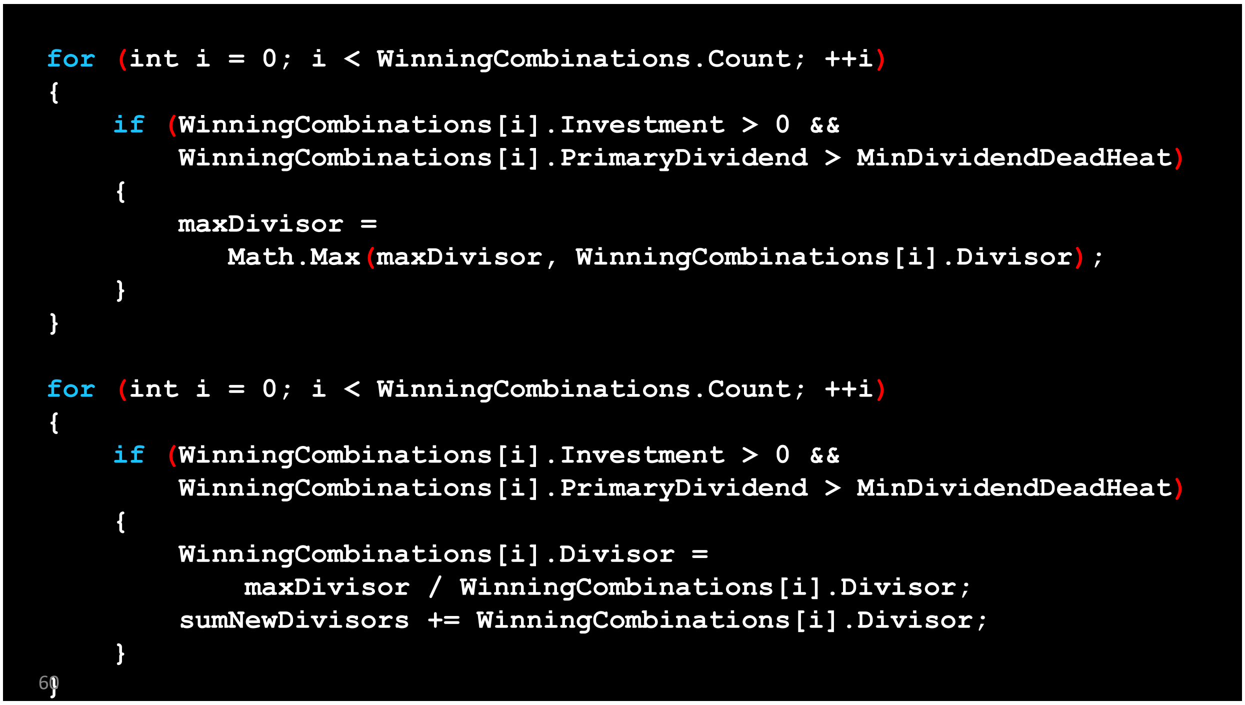
3



query block naming



59

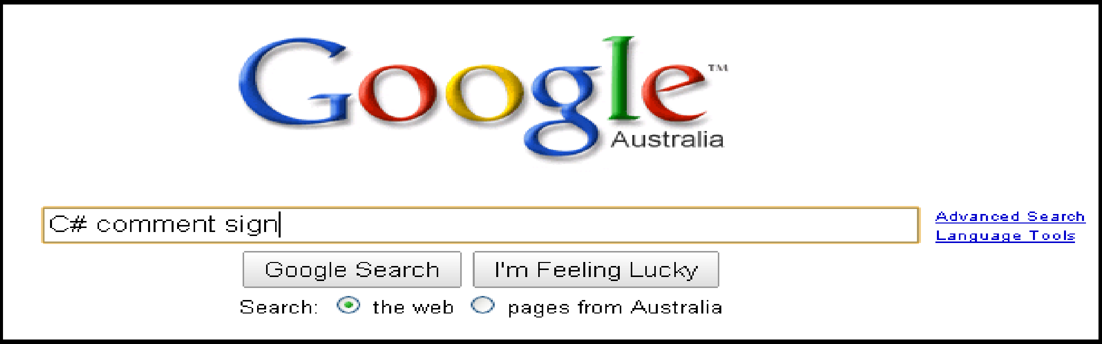


```
for (int i = 0; i < WinningCombinations.Count; ++i)
{
    if (WinningCombinations[i].Investment > 0 &&
        WinningCombinations[i].PrimaryDividend > MinDividendDeadHeat)
    {
        maxDivisor =
           Math.Max(maxDivisor, WinningCombinations[i].Divisor);
    }
}

for (int i = 0; i < WinningCombinations.Count; ++i)
{
    if (WinningCombinations[i].Investment > 0 &&
        WinningCombinations[i].PrimaryDividend > MinDividendDeadHeat)
    {
        WinningCombinations[i].Divisor =
            maxDivisor / WinningCombinations[i].Divisor;
        sumNewDivisors += WinningCombinations[i].Divisor;
    }
}
```

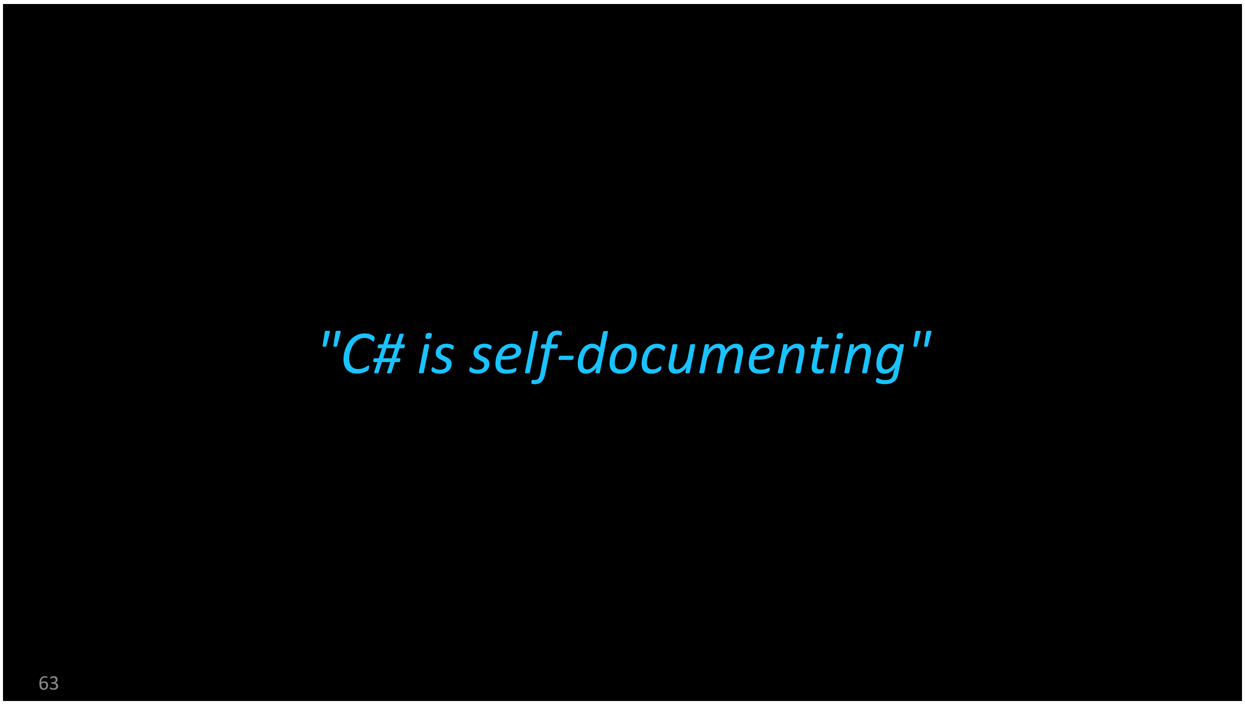
60

no comments

61

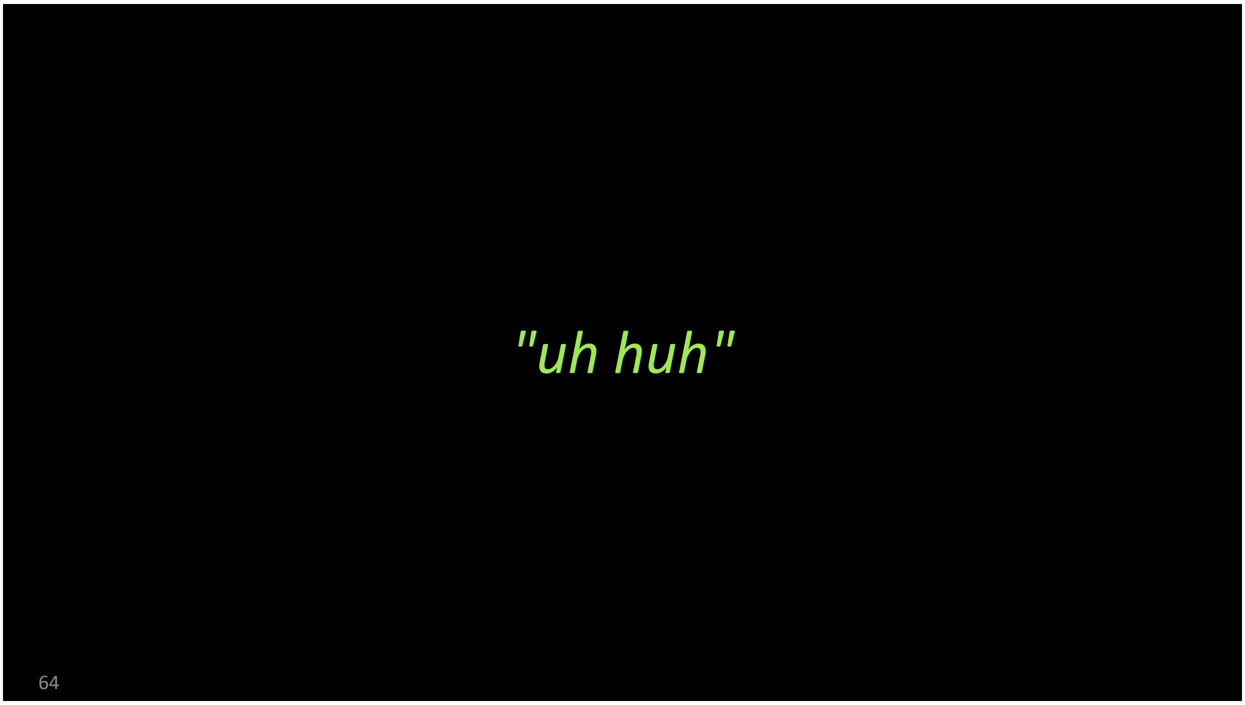

C# comment sign

62



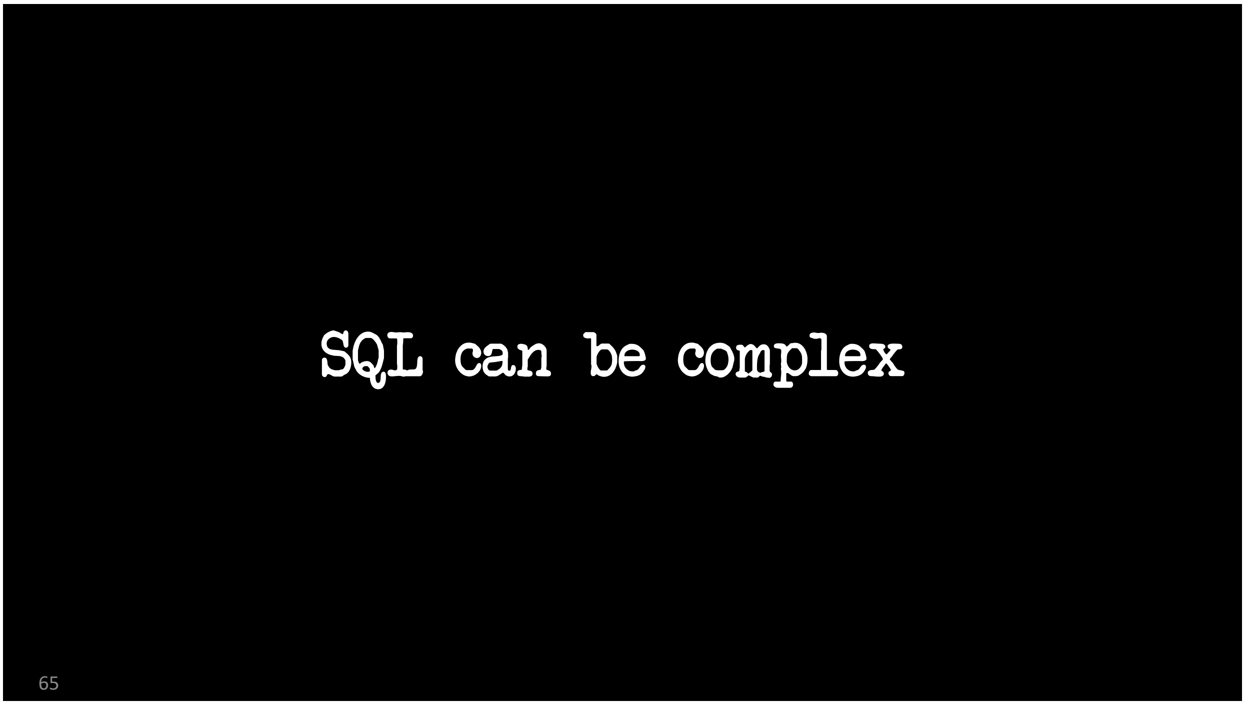
"C# is self-documenting"

63



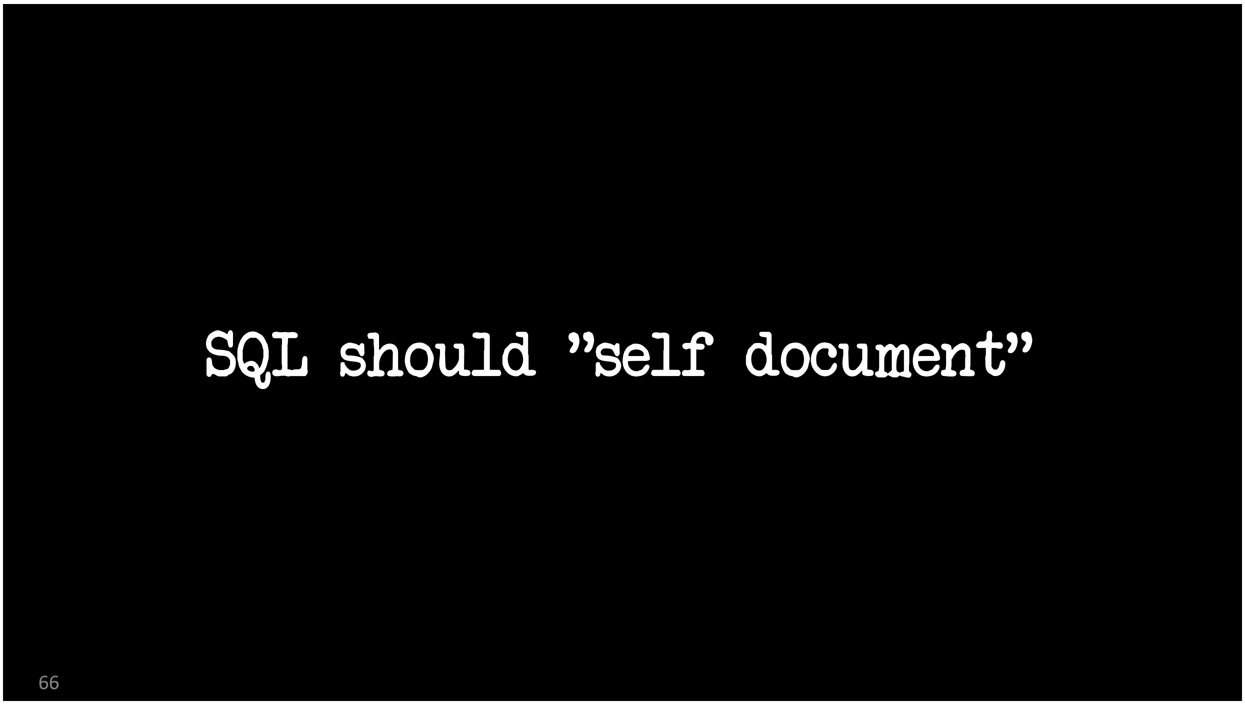
"uh huh"

64



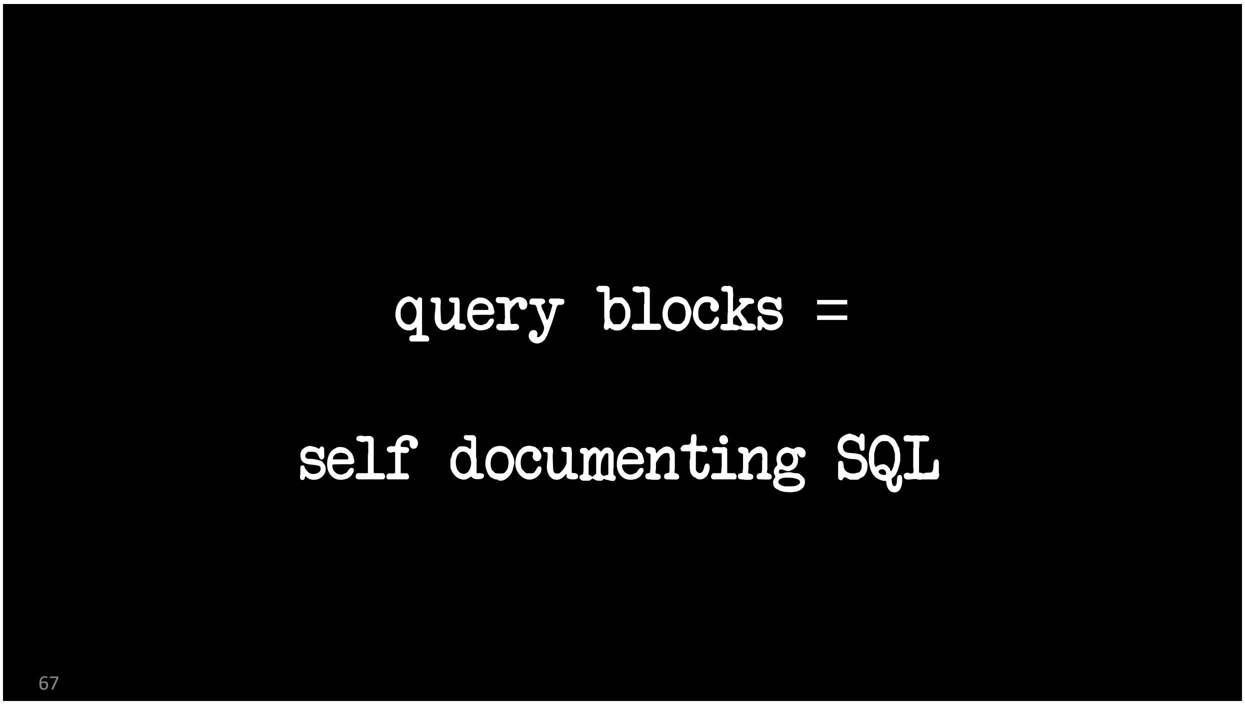
SQL can be complex

65



SQL should ”self document”

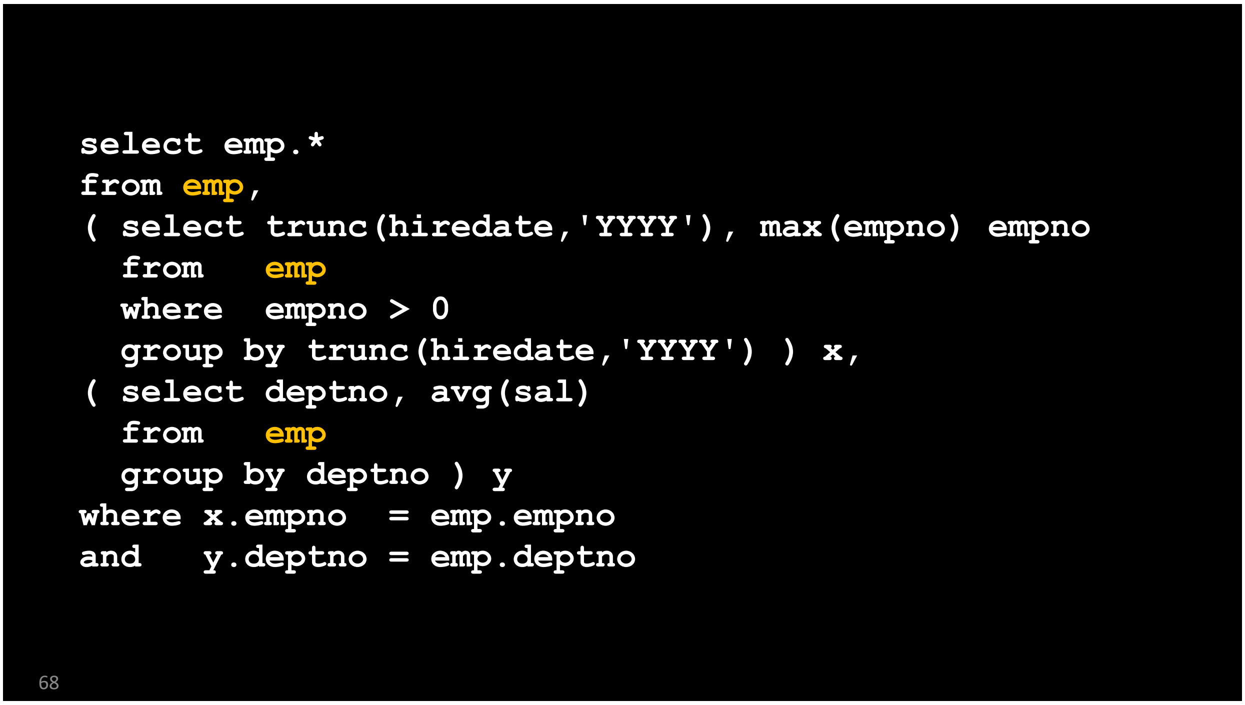
66



query blocks =

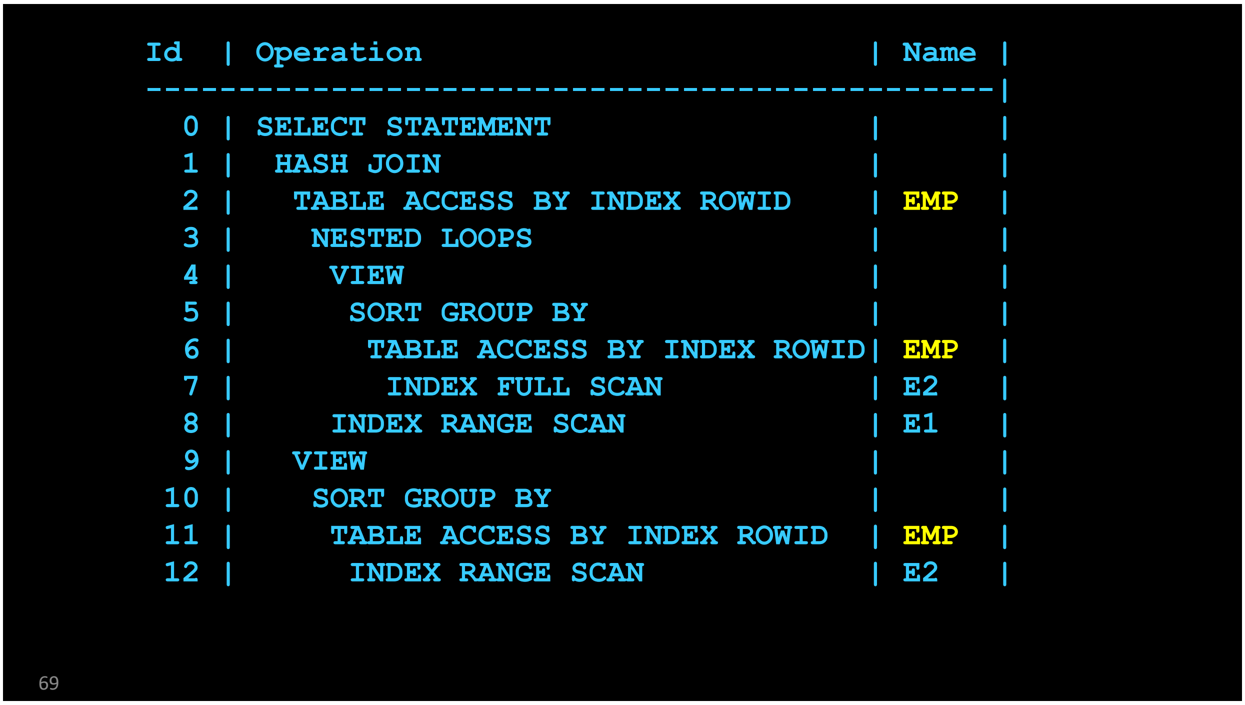
self documenting SQL

67



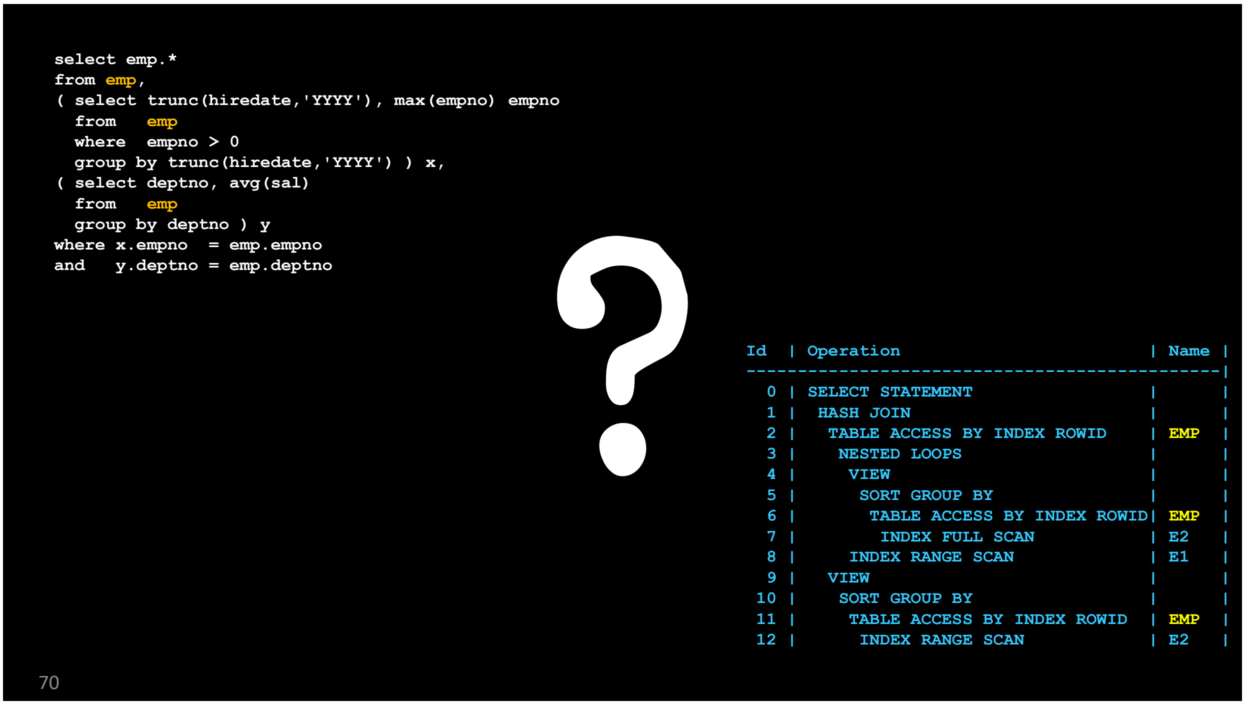
```
select emp.*
from emp,
( select trunc(hiredate,'YYYY'), max(empno) empno
  from   emp
  where  empno > 0
  group by trunc(hiredate,'YYYY') ) x,
( select deptno, avg(sal)
  from   emp
  group by deptno ) y
where x.empno  = emp.empno
and   y.deptno = emp.deptno
```

68



```
Id  | Operation                           | Name |
------------------------------------------------|
  0 | SELECT STATEMENT                    |      |
  1 |  HASH JOIN                          |      |
  2 |   TABLE ACCESS BY INDEX ROWID       | EMP  |
  3 |    NESTED LOOPS                     |      |
  4 |     VIEW                            |      |
  5 |      SORT GROUP BY                  |      |
  6 |       TABLE ACCESS BY INDEX ROWID| EMP  |
  7 |        INDEX FULL SCAN              | E2   |
  8 |     INDEX RANGE SCAN                | E1   |
  9 |   VIEW                              |      |
 10 |    SORT GROUP BY                    |      |
 11 |     TABLE ACCESS BY INDEX ROWID     | EMP  |
 12 |      INDEX RANGE SCAN               | E2   |
```

69

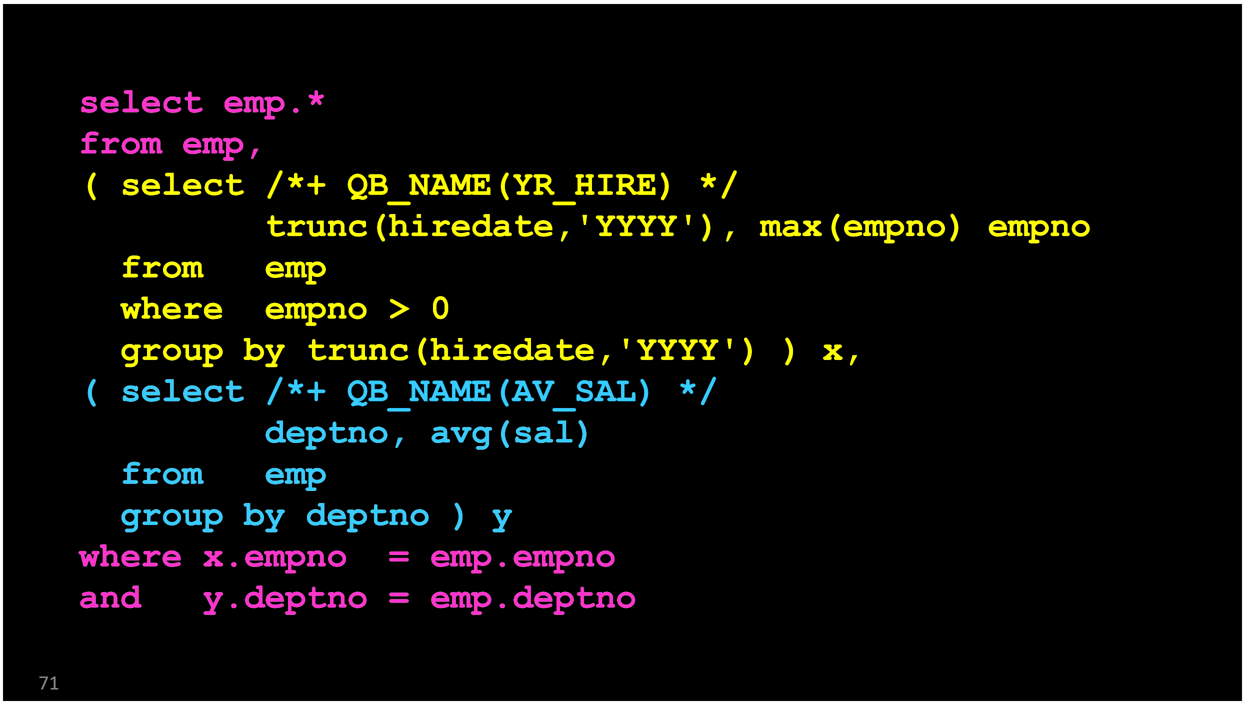


```
select emp.*
from emp,
( select trunc(hiredate,'YYYY'), max(empno) empno
  from   emp
  where  empno > 0
  group by trunc(hiredate,'YYYY') ) x,
( select deptno, avg(sal)
  from   emp
  group by deptno ) y
where x.empno  = emp.empno
and   y.deptno = emp.deptno
```

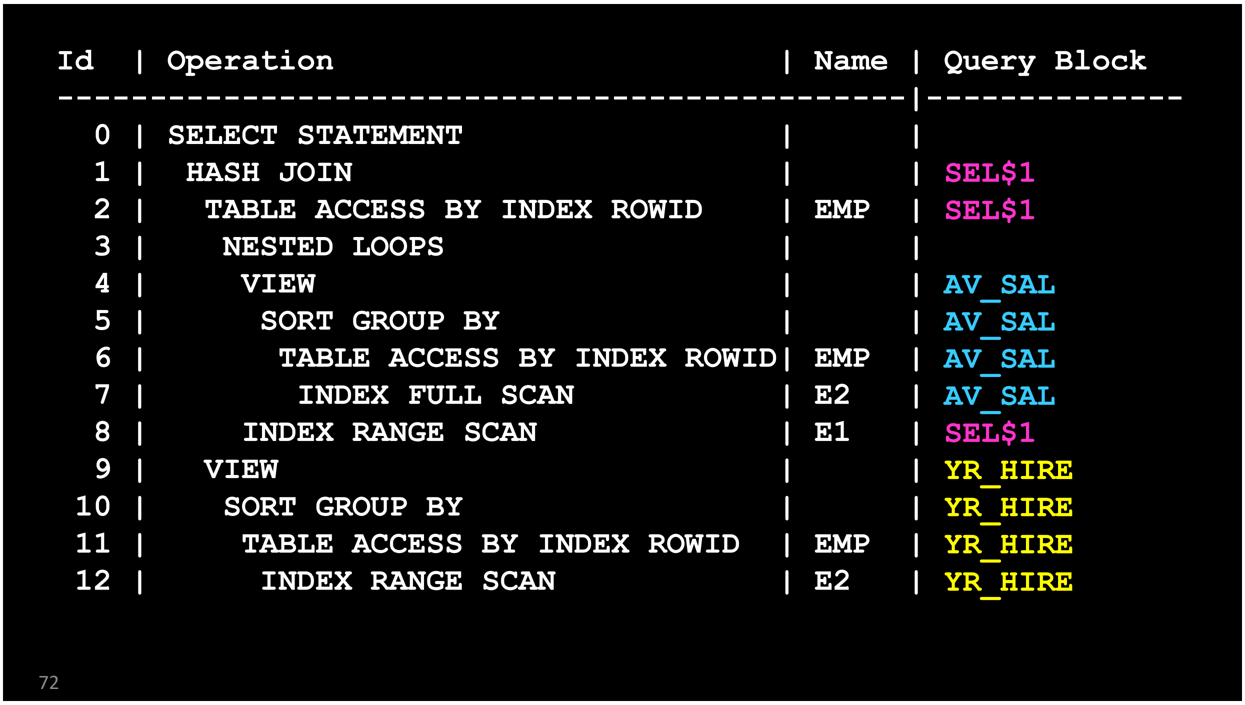
?

```
Id  | Operation                           | Name |
------------------------------------------------|
  0 | SELECT STATEMENT                    |      |
  1 |  HASH JOIN                          |      |
  2 |   TABLE ACCESS BY INDEX ROWID       | EMP  |
  3 |    NESTED LOOPS                     |      |
  4 |     VIEW                            |      |
  5 |      SORT GROUP BY                  |      |
  6 |       TABLE ACCESS BY INDEX ROWID| EMP  |
  7 |        INDEX FULL SCAN              | E2   |
  8 |     INDEX RANGE SCAN                | E1   |
  9 |   VIEW                              |      |
 10 |    SORT GROUP BY                    |      |
 11 |     TABLE ACCESS BY INDEX ROWID     | EMP  |
 12 |      INDEX RANGE SCAN               | E2   |
```

70



```
select emp.*
from emp,
( select /*+ QB_NAME(YR_HIRE) */
         trunc(hiredate,'YYYY'), max(empno) empno
  from   emp
  where  empno > 0
  group by trunc(hiredate,'YYYY') ) x,
( select /*+ QB_NAME(AV_SAL) */
         deptno, avg(sal)
  from   emp
  group by deptno ) y
where x.empno  = emp.empno
and   y.deptno = emp.deptno
```



```
Id  | Operation                         | Name | Query Block
----------------------------------------|--------------
  0 | SELECT STATEMENT                  |      |
  1 |  HASH JOIN                        |      | SEL$1
  2 |   TABLE ACCESS BY INDEX ROWID     | EMP  | SEL$1
  3 |    NESTED LOOPS                   |      |
  4 |     VIEW                          |      | AV_SAL
  5 |      SORT GROUP BY                |      | AV_SAL
  6 |       TABLE ACCESS BY INDEX ROWID| EMP  | AV_SAL
  7 |        INDEX FULL SCAN            | E2   | AV_SAL
  8 |     INDEX RANGE SCAN              | E1   | SEL$1
  9 |   VIEW                            |      | YR_HIRE
 10 |    SORT GROUP BY                  |      | YR_HIRE
 11 |     TABLE ACCESS BY INDEX ROWID   | EMP  | YR_HIRE
 12 |      INDEX RANGE SCAN             | E2   | YR_HIRE
```

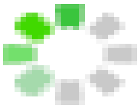
4

73

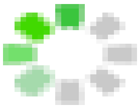
error logging

74

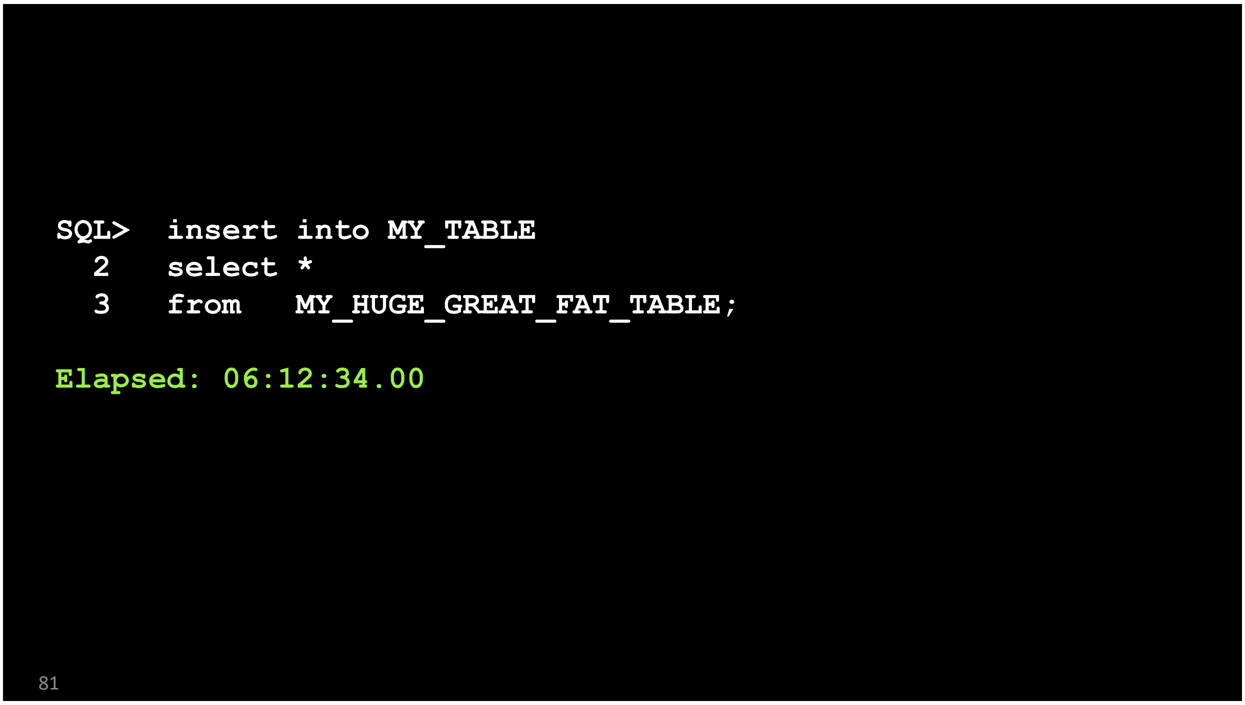
```
SQL>  insert into MY_TABLE
  2   select *
  3   from   MY_HUGE_GREAT_FAT_TABLE;
```



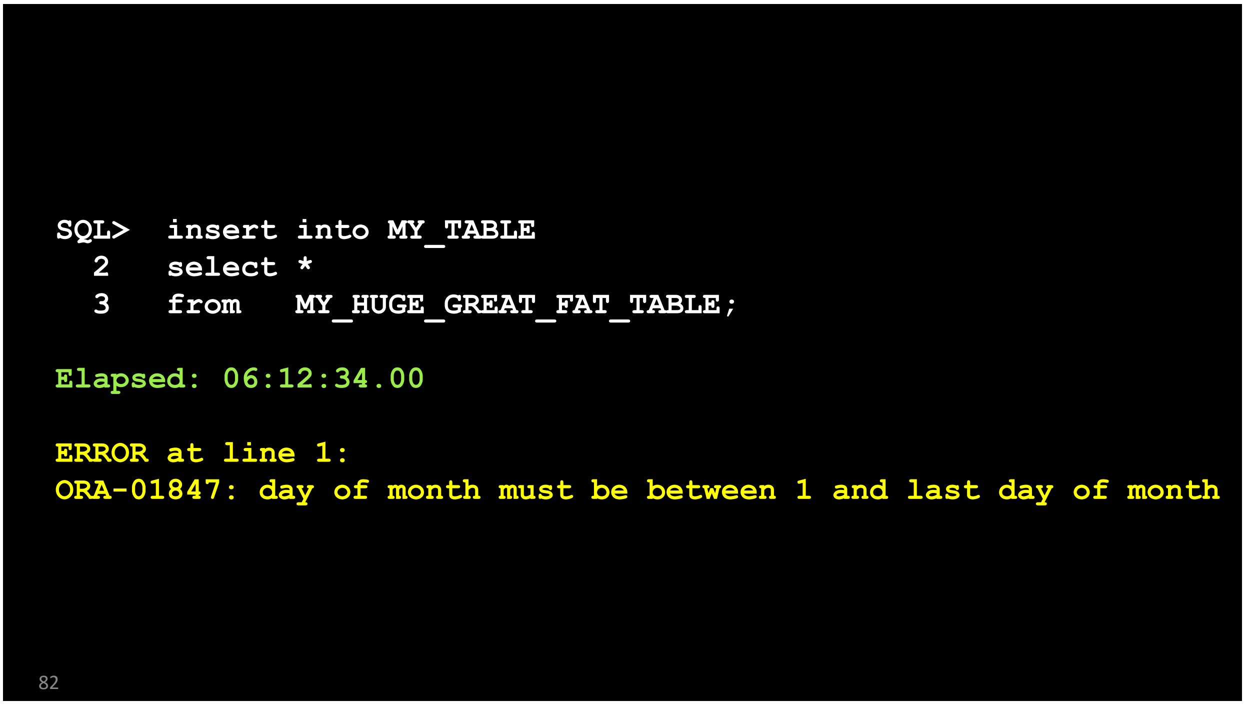
79



80



```
SQL>  insert into MY_TABLE
  2   select *
  3   from   MY_HUGE_GREAT_FAT_TABLE;

Elapsed: 06:12:34.00
```

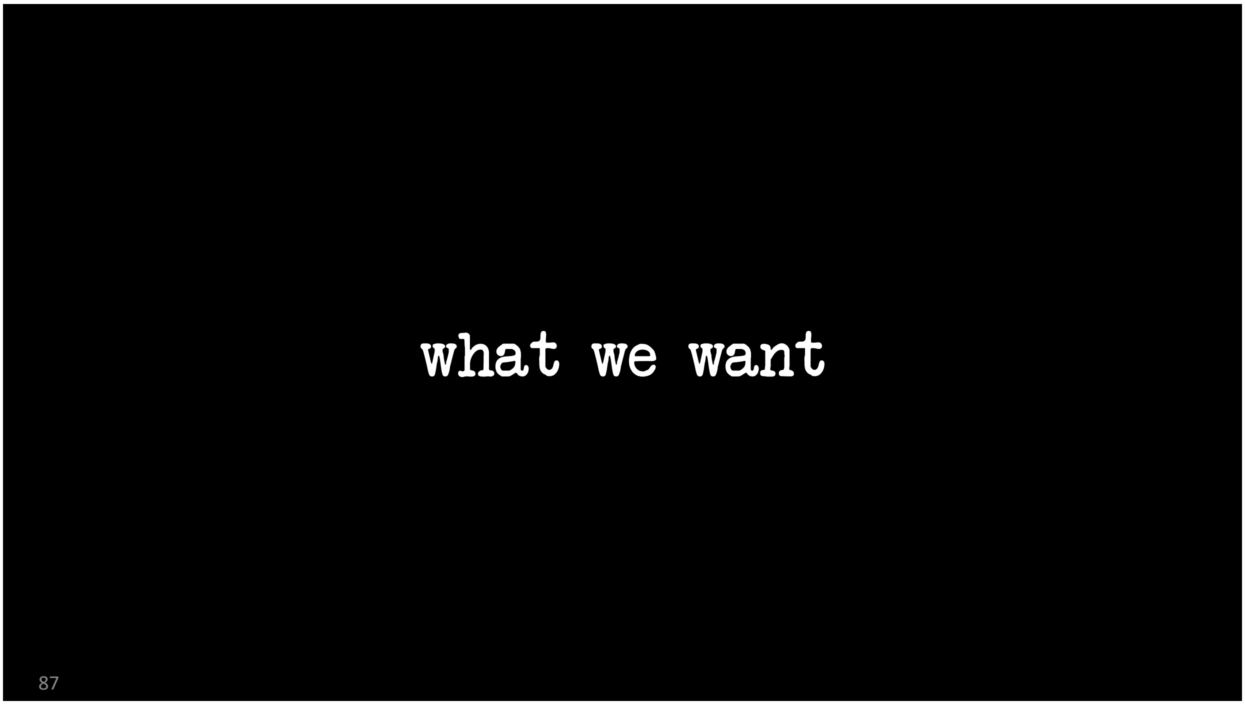


```
SQL>  insert into MY_TABLE
  2   select *
  3   from   MY_HUGE_GREAT_FAT_TABLE;

Elapsed: 06:12:34.00

ERROR at line 1:
ORA-01847: day of month must be between 1 and last day of month
```

NOoooooooo !

and then we do this :)

```
SQL>  select count(*) from MY_TABLE;

  COUNT(*)
----------
         0
```

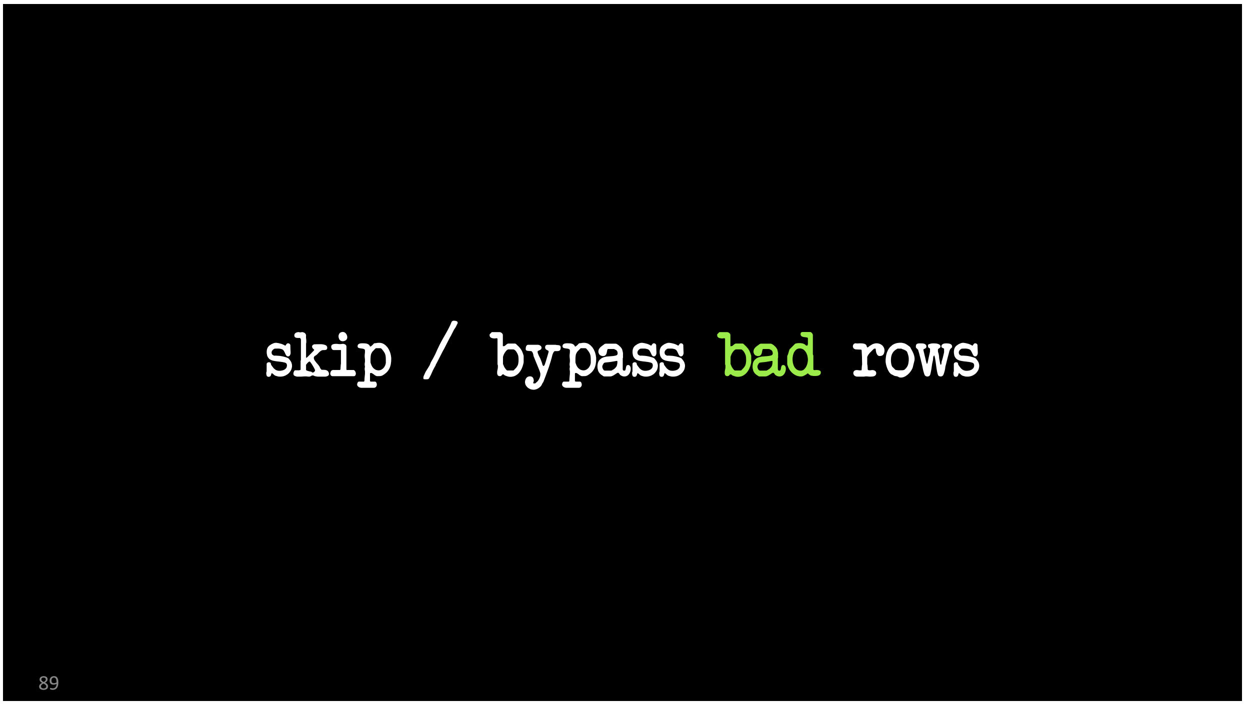


what we want

87

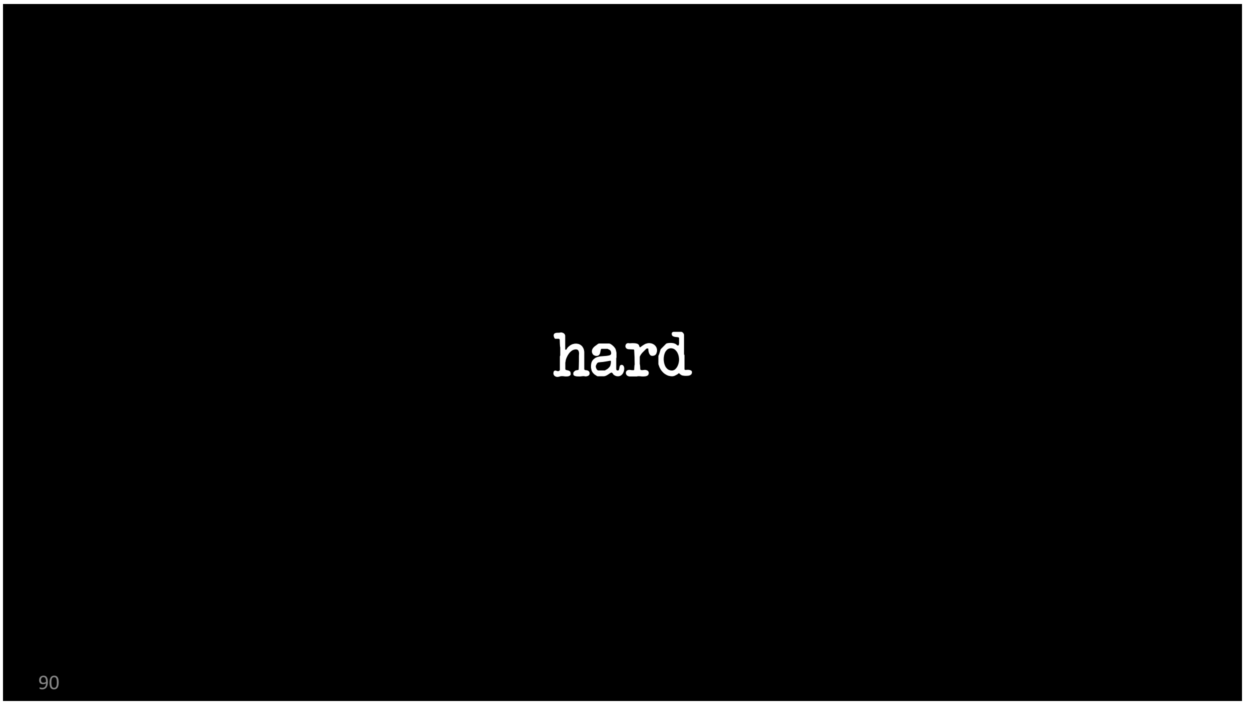
keep successful rows

88



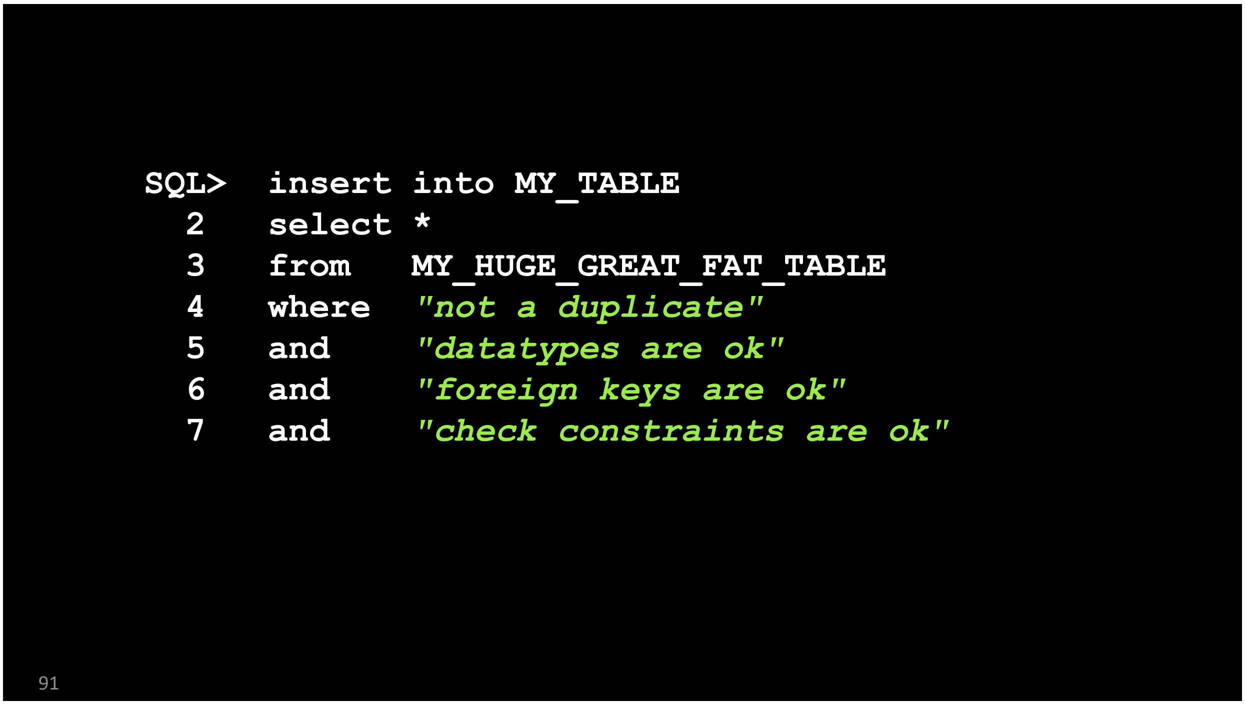
skip / bypass bad rows

89

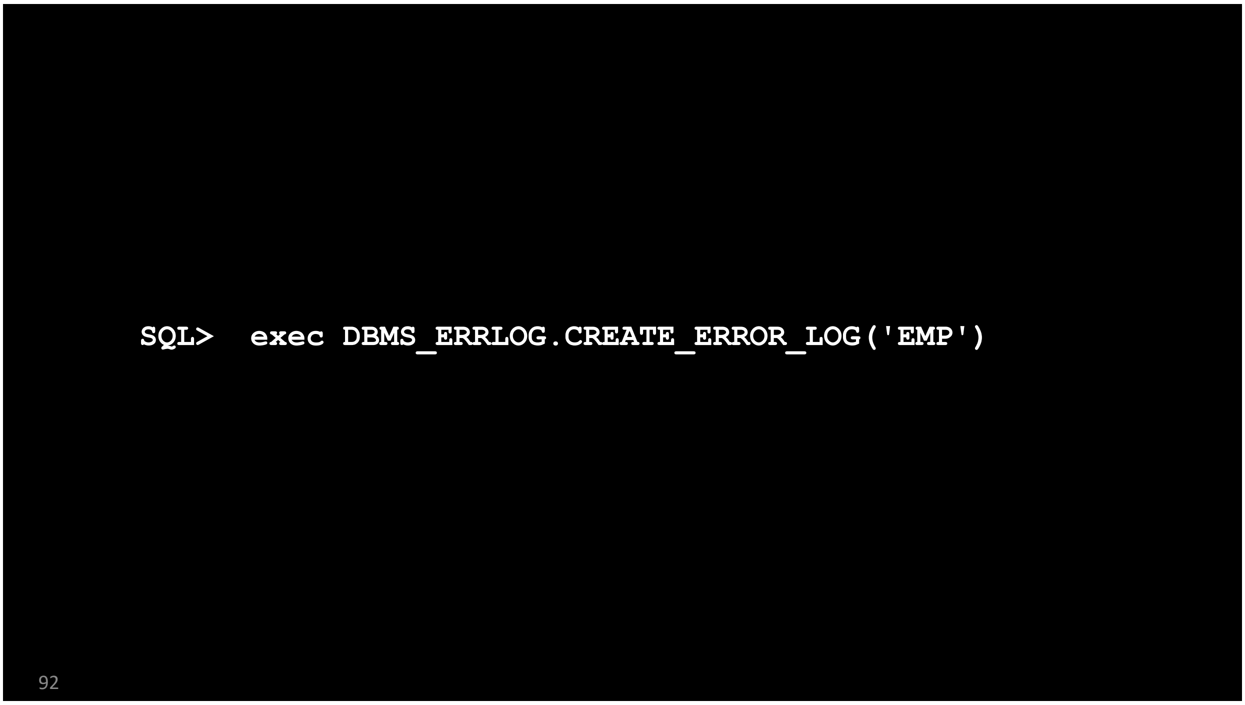


hard

90

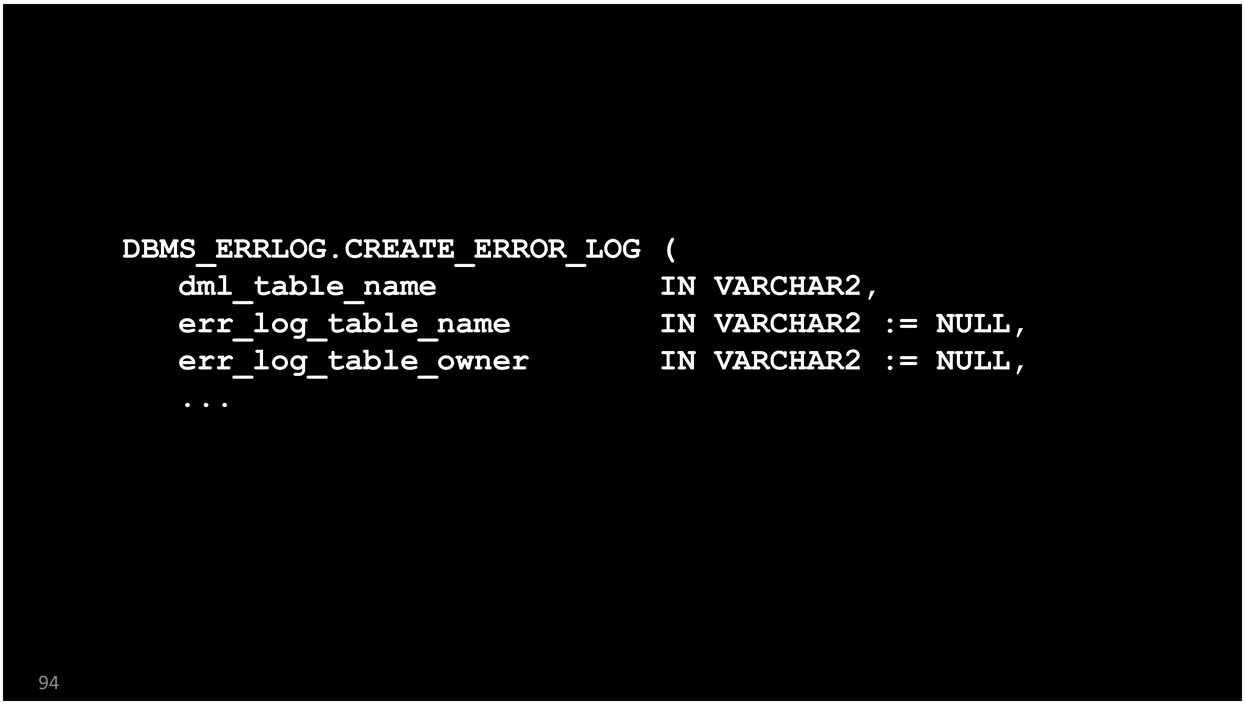


```
SQL>  insert into MY_TABLE
  2   select *
  3   from   MY_HUGE_GREAT_FAT_TABLE
  4   where  "not a duplicate"
  5   and    "datatypes are ok"
  6   and    "foreign keys are ok"
  7   and    "check constraints are ok"
```

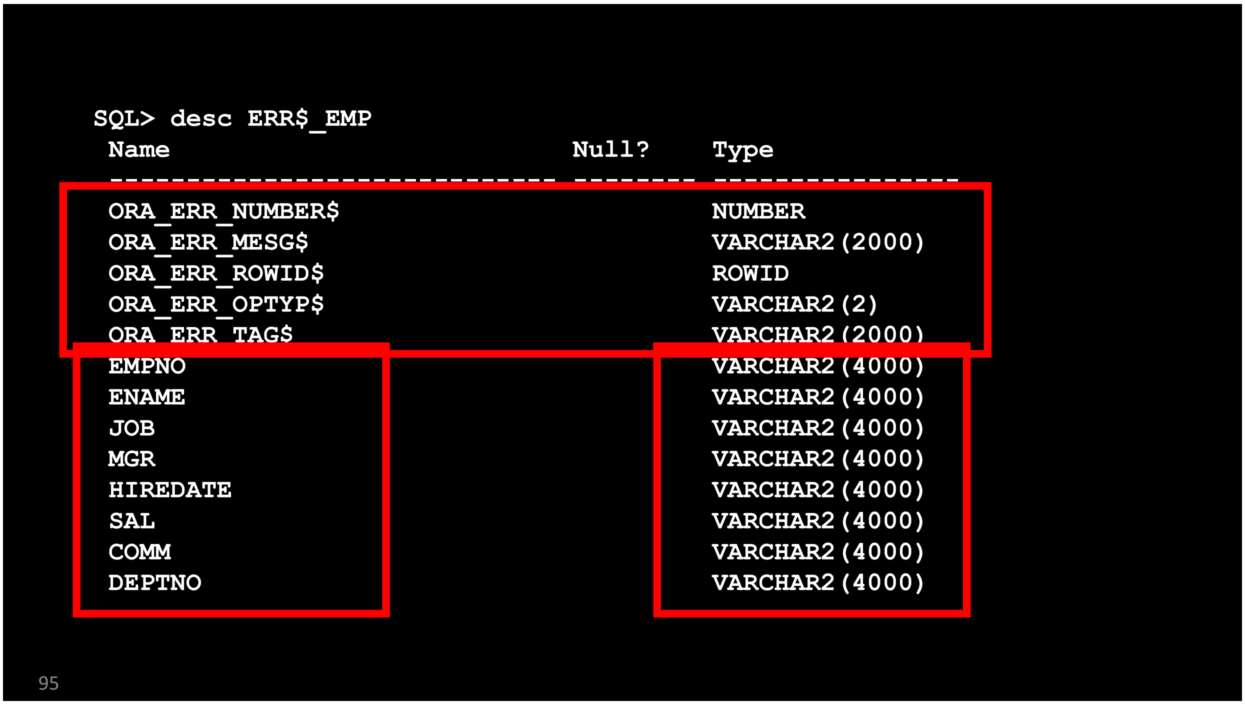


```
SQL>  exec DBMS_ERRLOG.CREATE_ERROR_LOG('EMP')
```

ERR$_EMP



```
DBMS_ERRLOG.CREATE_ERROR_LOG (
   dml_table_name            IN VARCHAR2,
   err_log_table_name        IN VARCHAR2 := NULL,
   err_log_table_owner       IN VARCHAR2 := NULL,
   ...
```



```
SQL> desc ERR$_EMP
 Name                          Null?    Type
 ----------------------------- -------- ----------------
 ORA_ERR_NUMBER$                        NUMBER
 ORA_ERR_MESG$                          VARCHAR2(2000)
 ORA_ERR_ROWID$                         ROWID
 ORA_ERR_OPTYP$                         VARCHAR2(2)
 ORA_ERR_TAG$                           VARCHAR2(2000)
 EMPNO                                  VARCHAR2(4000)
 ENAME                                  VARCHAR2(4000)
 JOB                                    VARCHAR2(4000)
 MGR                                    VARCHAR2(4000)
 HIREDATE                               VARCHAR2(4000)
 SAL                                    VARCHAR2(4000)
 COMM                                   VARCHAR2(4000)
 DEPTNO                                 VARCHAR2(4000)
```

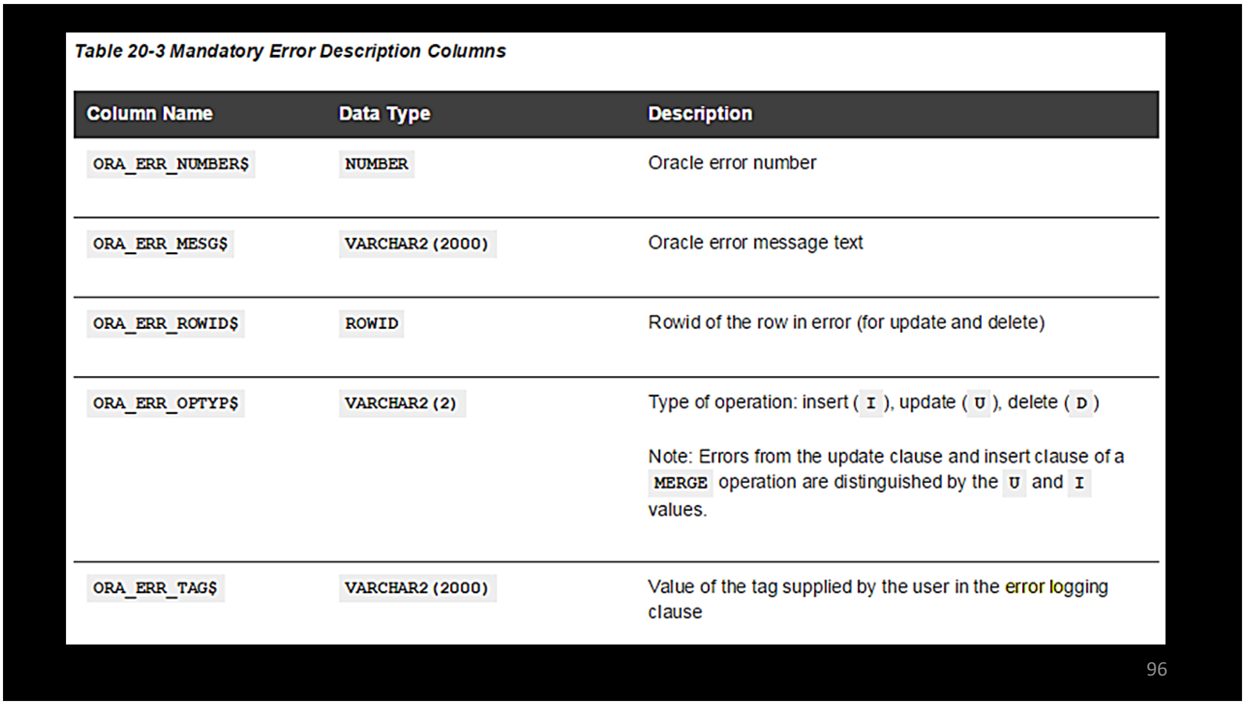
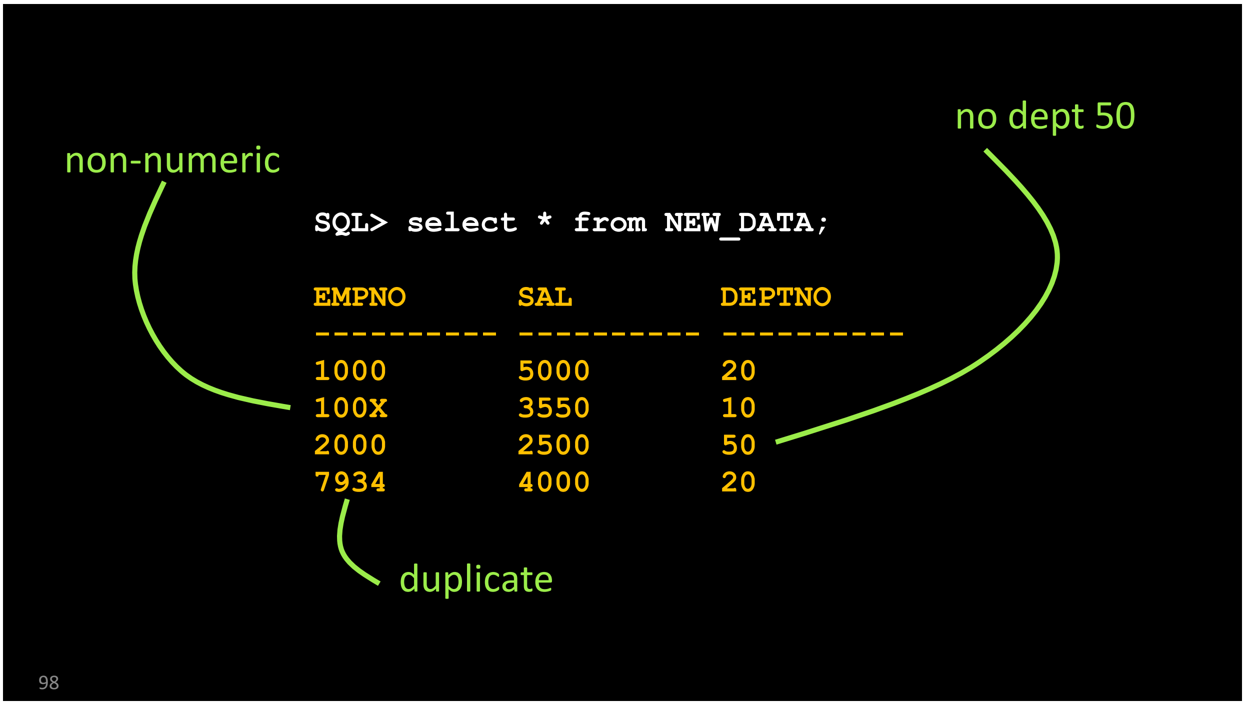


Table 20-3 Mandatory Error Description Columns

Column Name	Data Type	Description
ORA_ERR_NUMBER$	NUMBER	Oracle error number
ORA_ERR_MESG$	VARCHAR2(2000)	Oracle error message text
ORA_ERR_ROWID$	ROWID	Rowid of the row in error (for update and delete)
ORA_ERR_OPTYP$	VARCHAR2(2)	Type of operation: insert (I), update (U), delete (D) Note: Errors from the update clause and insert clause of a MERGE operation are distinguished by the U and I values.
ORA_ERR_TAG$	VARCHAR2(2000)	Value of the tag supplied by the user in the error logging clause

example

97



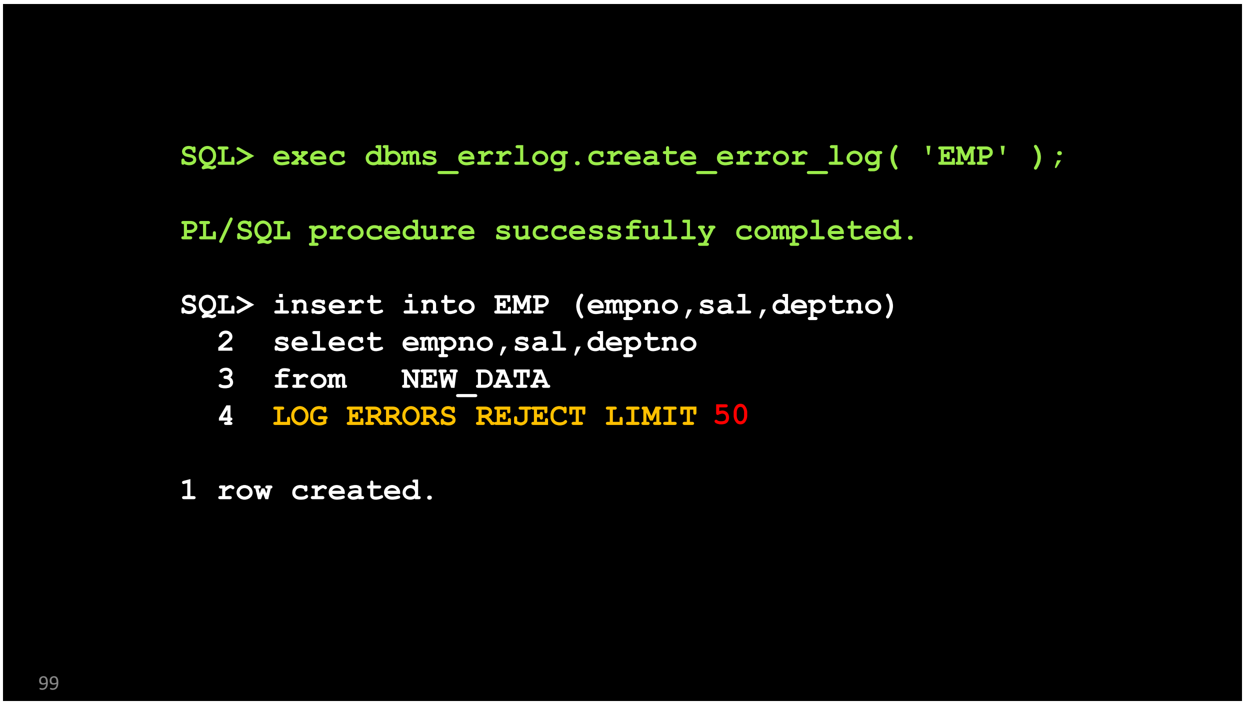
non-numeric

no dept 50

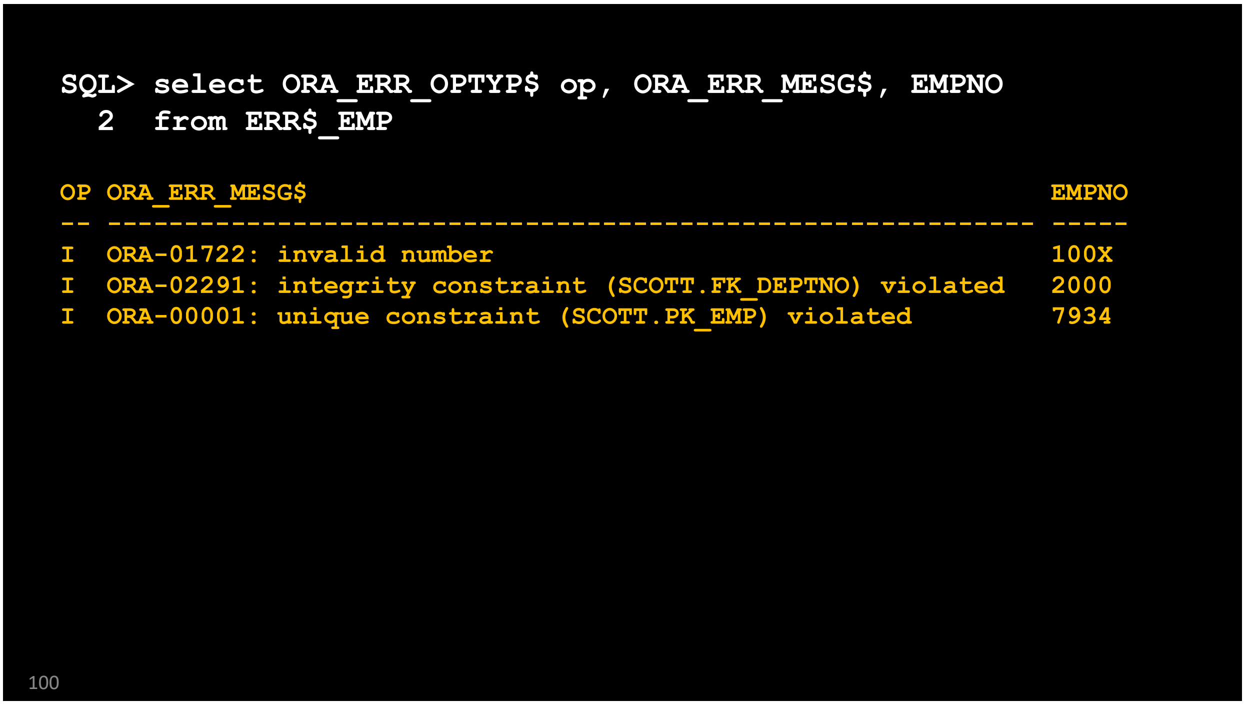
```
SQL> select * from NEW_DATA;

EMPNO      SAL        DEPTNO
---------- ---------- ----------
1000       5000       20
100X       3550       10
2000       2500       50
7934       4000       20
```

duplicate

98



```
SQL> exec dbms_errlog.create_error_log( 'EMP' );

PL/SQL procedure successfully completed.

SQL> insert into EMP (empno,sal,deptno)
  2  select empno,sal,deptno
  3  from   NEW_DATA
  4  LOG ERRORS REJECT LIMIT 50

1 row created.
```



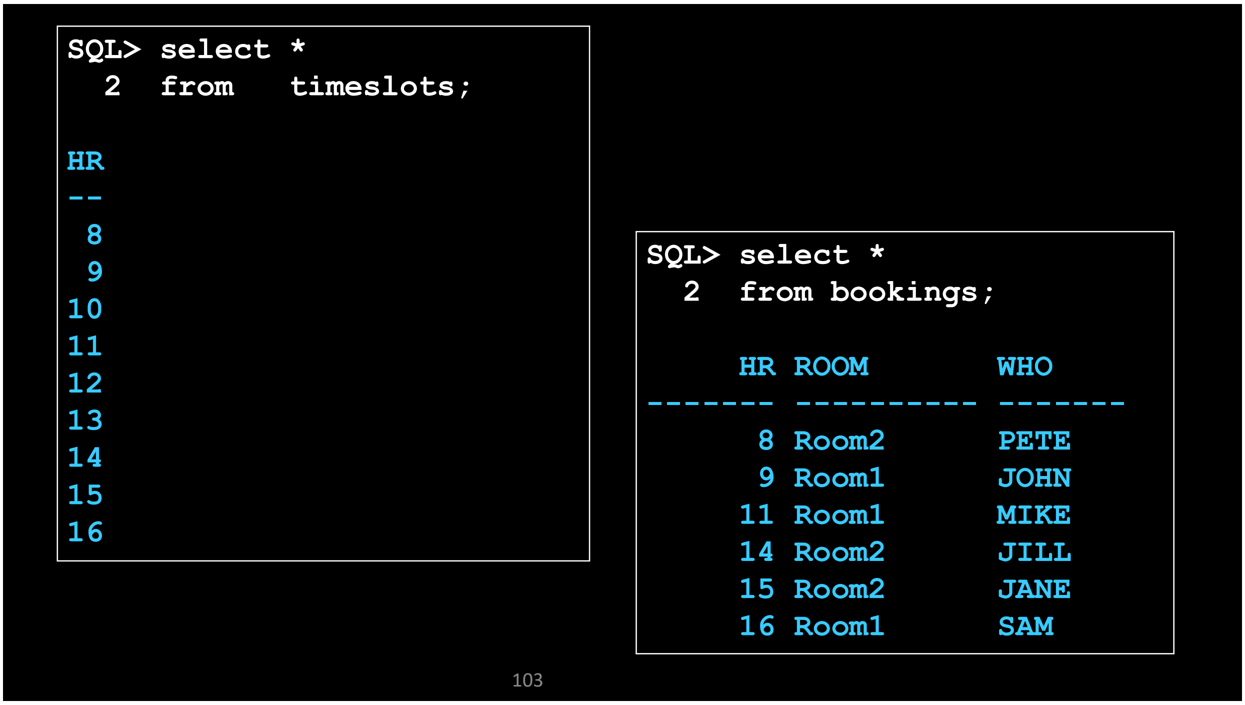
```
SQL> select ORA_ERR_OPTYP$ op, ORA_ERR_MESG$, EMPNO
  2  from ERR$_EMP

OP ORA_ERR_MESG$                                                EMPNO
-- ------------------------------------------------------------ -----
I  ORA-01722: invalid number                                    100X
I  ORA-02291: integrity constraint (SCOTT.FK_DEPTNO) violated   2000
I  ORA-00001: unique constraint (SCOTT.PK_EMP) violated         7934
```

5

101

partitioned outer join

102

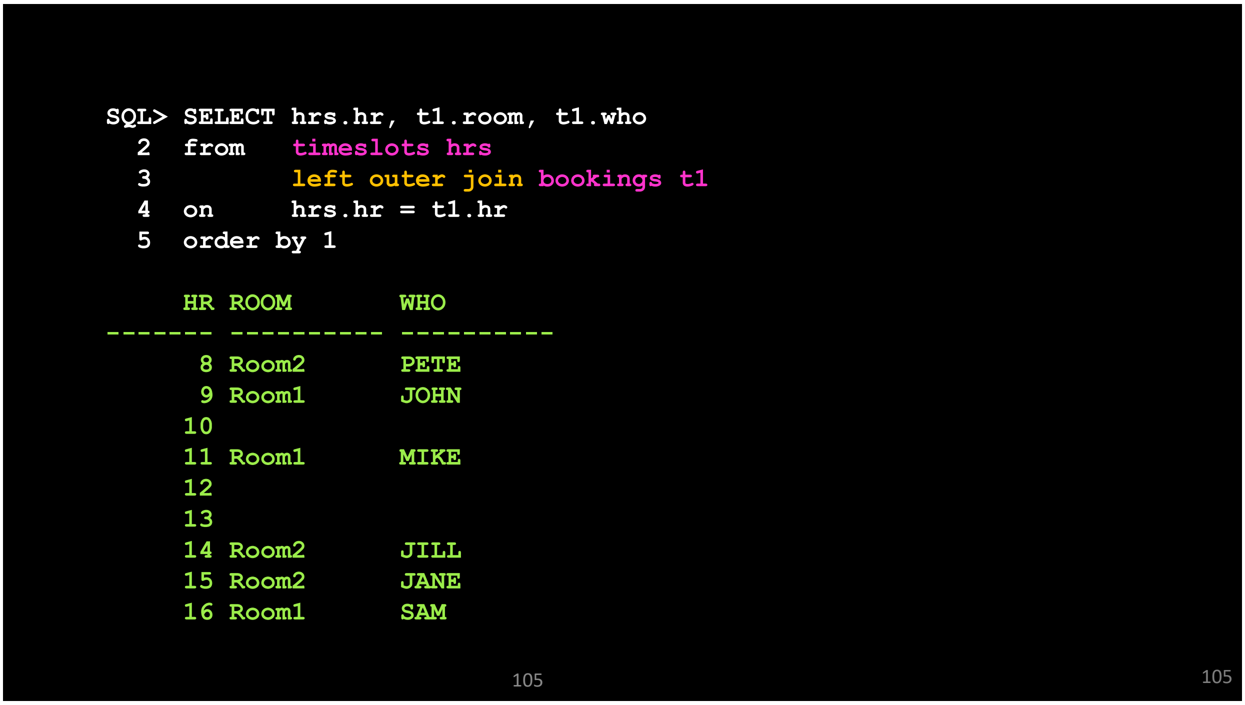


```
SQL> select *
  2  from   timeslots;

HR
--
 8
 9
10
11
12
13
14
15
16
```

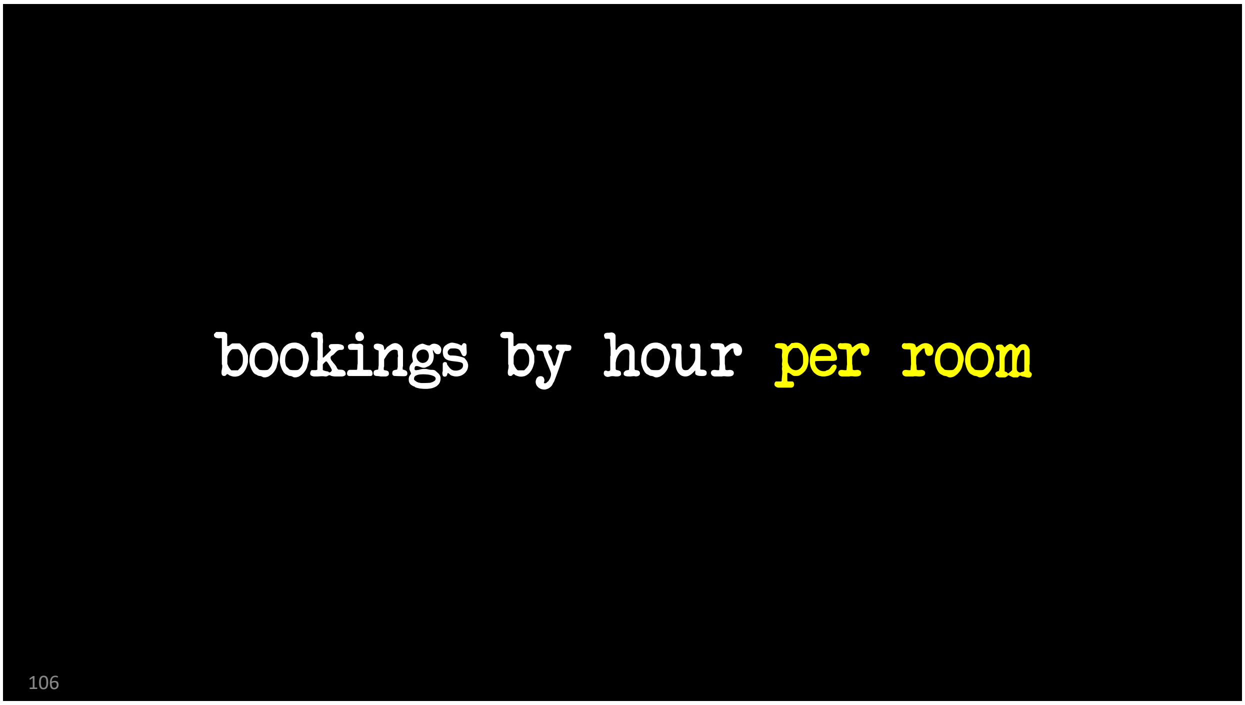
```
SQL> select *
  2  from bookings;

     HR ROOM       WHO
------- ---------- -------
      8 Room2      PETE
      9 Room1      JOHN
     11 Room1      MIKE
     14 Room2      JILL
     15 Room2      JANE
     16 Room1      SAM
```

bookings by hour

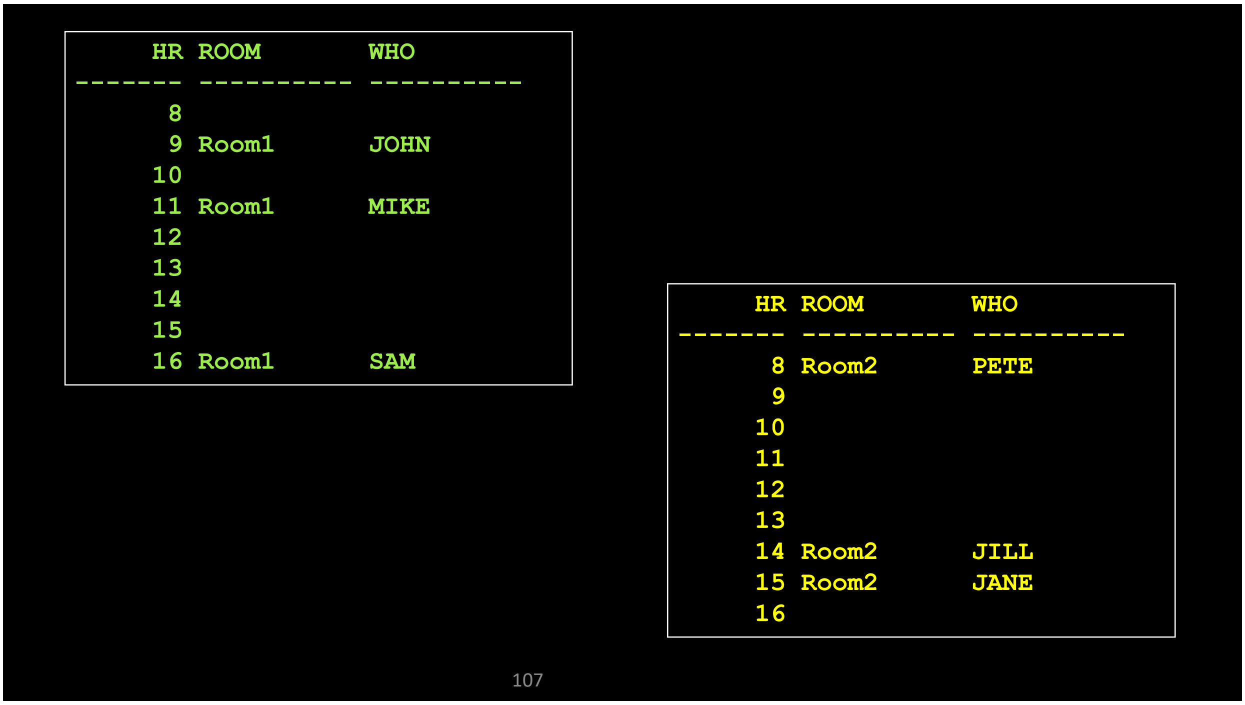
conventional outer join



```
SQL> SELECT hrs.hr, t1.room, t1.who
  2  from   timeslots hrs
  3         left outer join bookings t1
  4  on     hrs.hr = t1.hr
  5  order by 1

     HR ROOM       WHO
------- ---------- ----------
      8 Room2      PETE
      9 Room1      JOHN
     10
     11 Room1      MIKE
     12
     13
     14 Room2      JILL
     15 Room2      JANE
     16 Room1      SAM
```

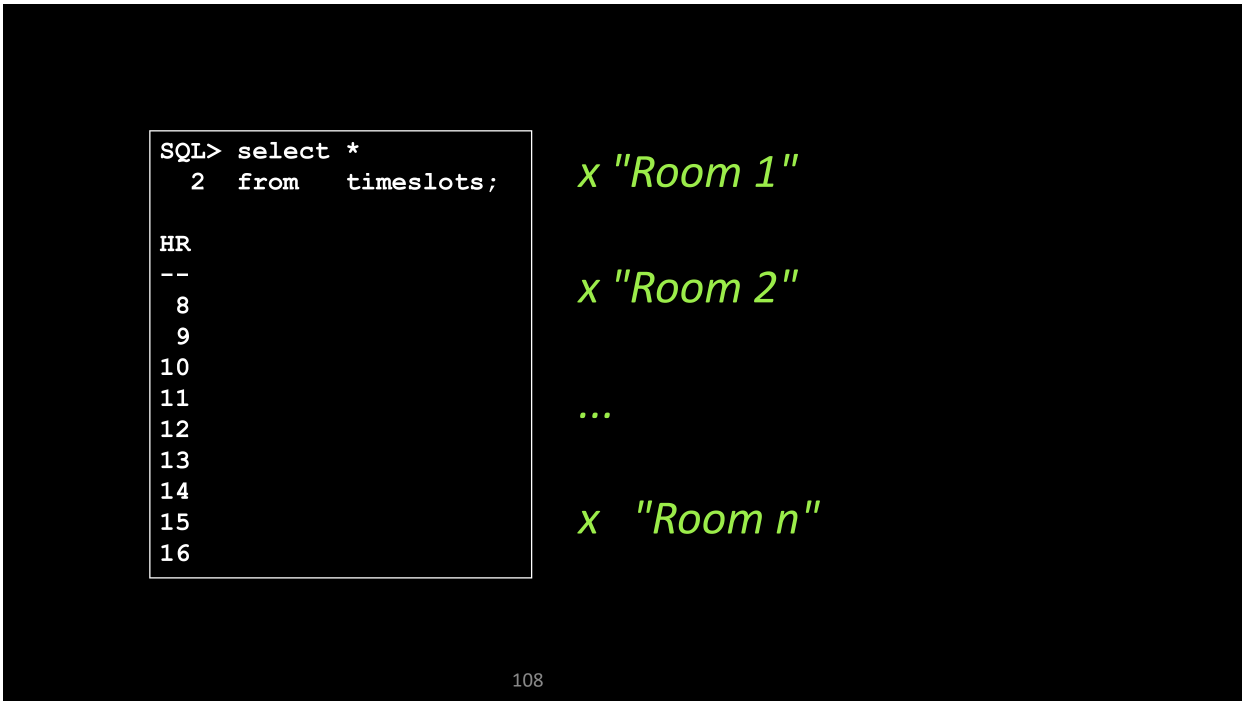


bookings by hour per room



```
      HR ROOM       WHO
------- ---------- ----------
      8
      9 Room1      JOHN
     10
     11 Room1      MIKE
     12
     13
     14
     15
     16 Room1      SAM
```

```
      HR ROOM       WHO
------- ---------- ----------
      8 Room2      PETE
      9
     10
     11
     12
     13
     14 Room2      JILL
     15 Room2      JANE
     16
```



```
SQL> select *
  2  from   timeslots;

HR
--
 8
 9
10
11
12
13
14
15
16
```

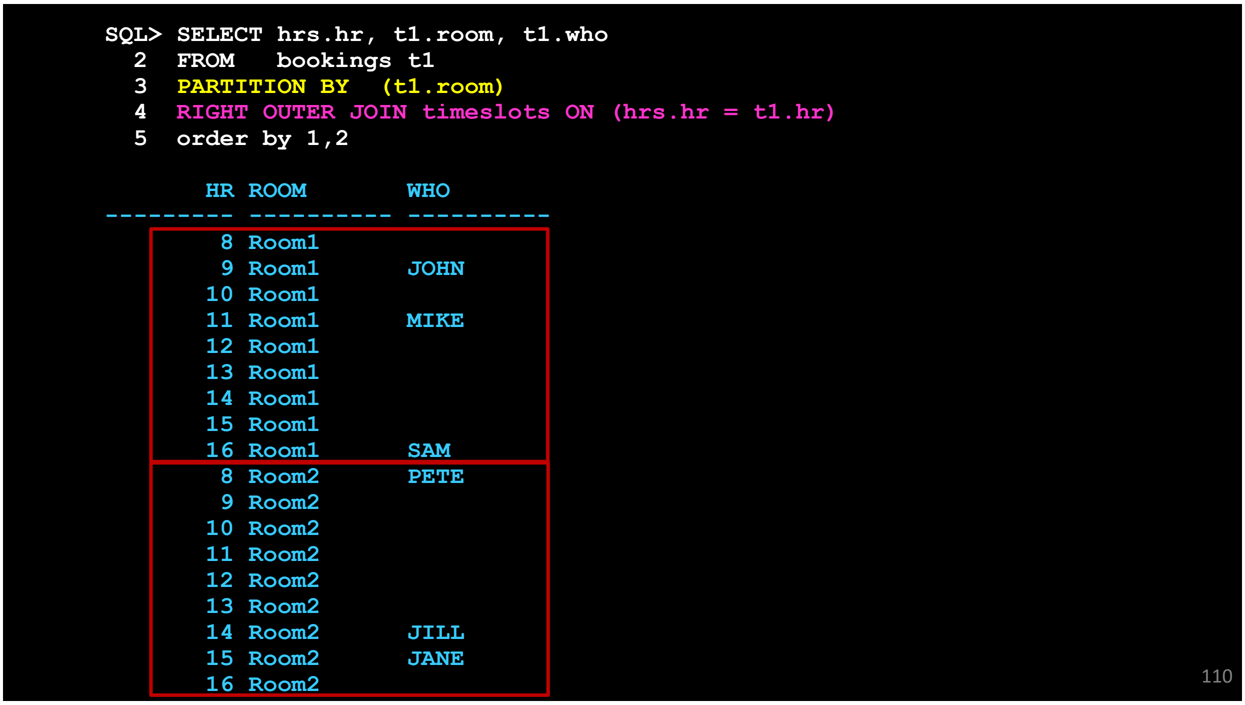
x "Room 1"

x "Room 2"

…

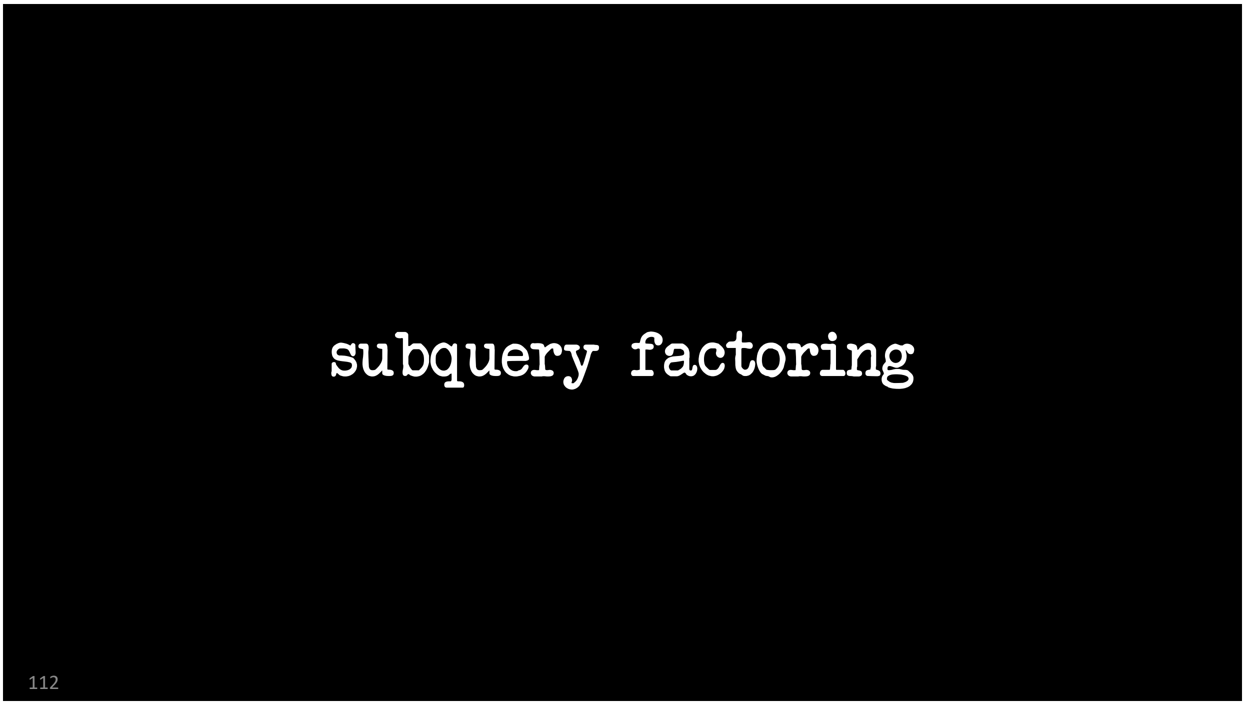
x "Room n"

partitioned outer join



```
SQL> SELECT hrs.hr, t1.room, t1.who
  2  FROM   bookings t1
  3  PARTITION BY  (t1.room)
  4  RIGHT OUTER JOIN timeslots ON (hrs.hr = t1.hr)
  5  order by 1,2

        HR ROOM       WHO
---------- ---------- ----------
         8 Room1
         9 Room1      JOHN
        10 Room1
        11 Room1      MIKE
        12 Room1
        13 Room1
        14 Room1
        15 Room1
        16 Room1      SAM
         8 Room2      PETE
         9 Room2
        10 Room2
        11 Room2
        12 Room2
        13 Room2
        14 Room2      JILL
        15 Room2      JANE
        16 Room2
```

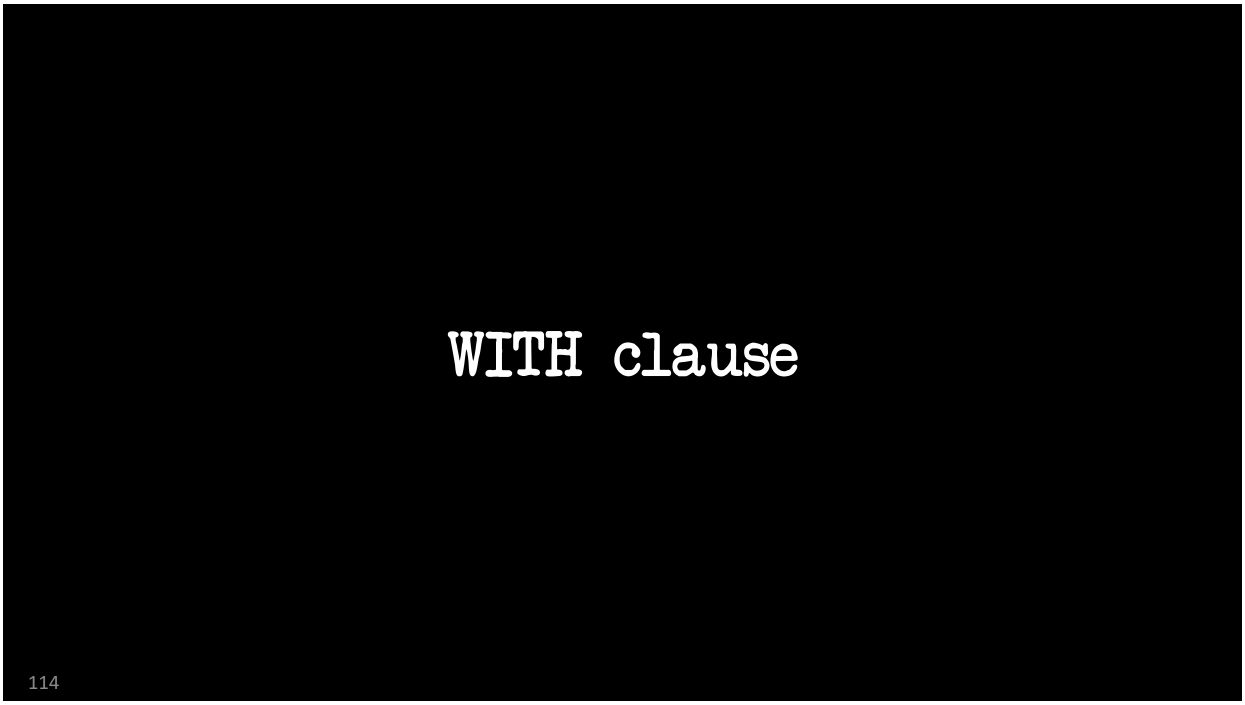
6



subquery factoring

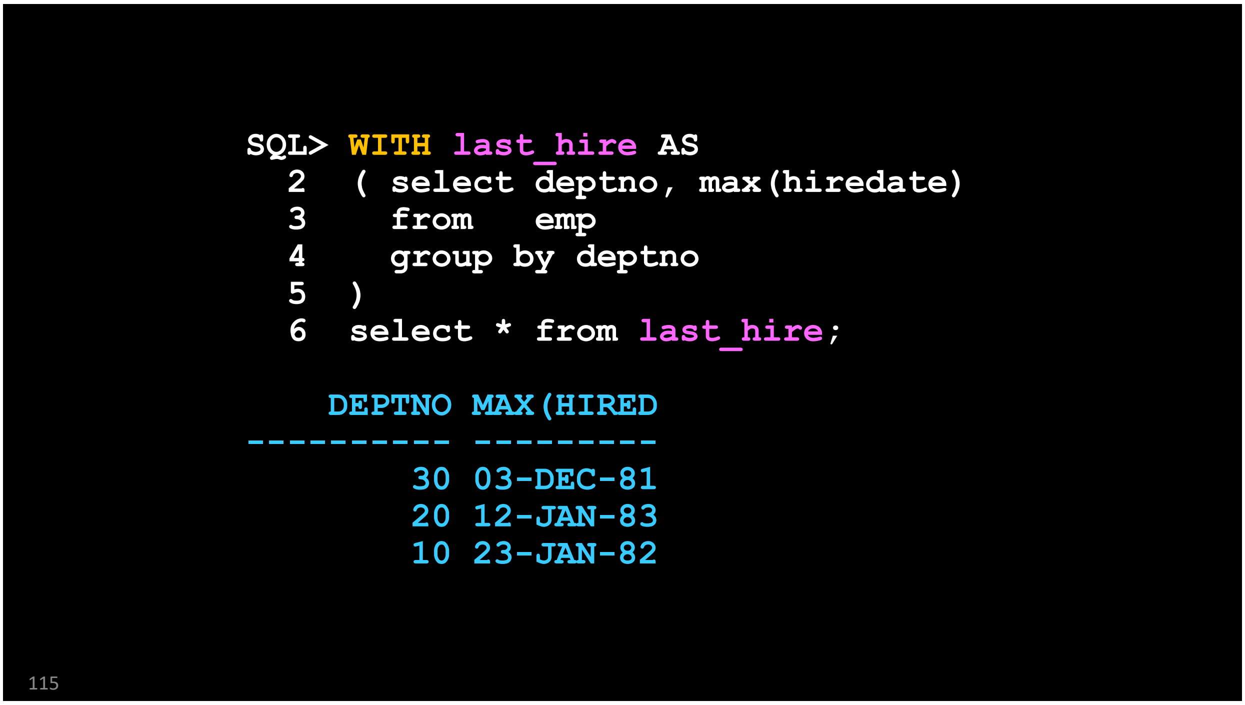
common table expressions

113

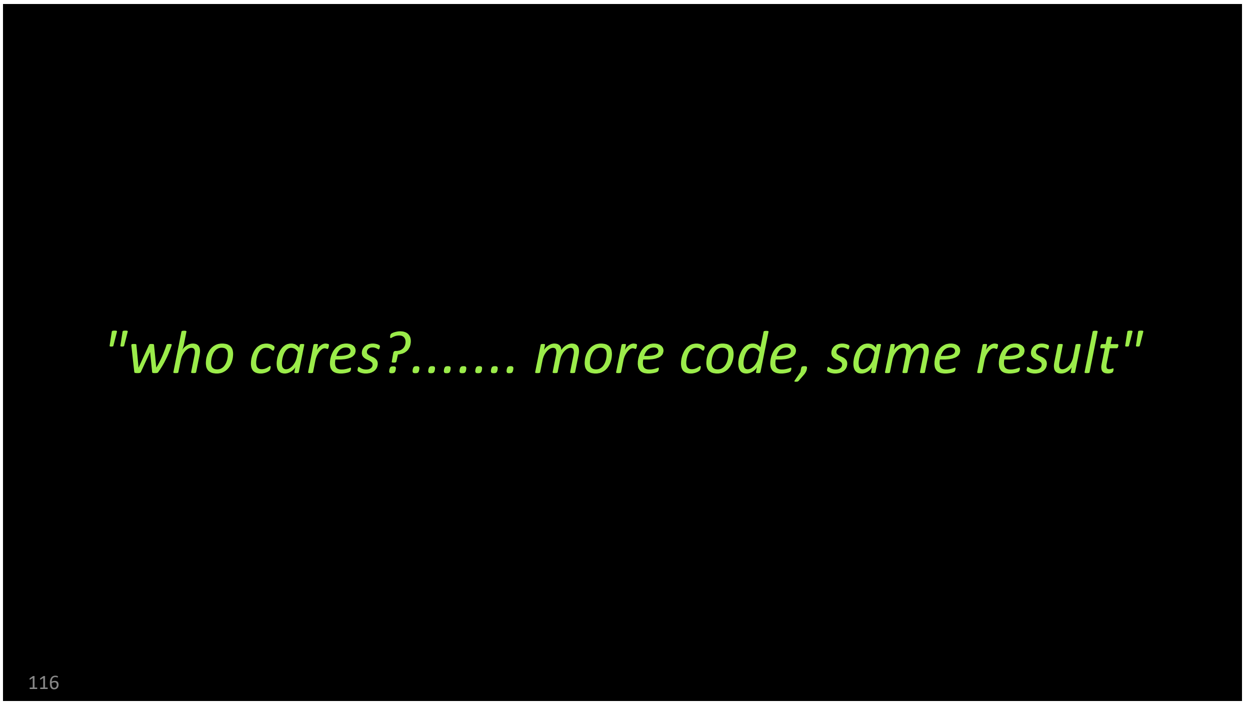


WITH clause

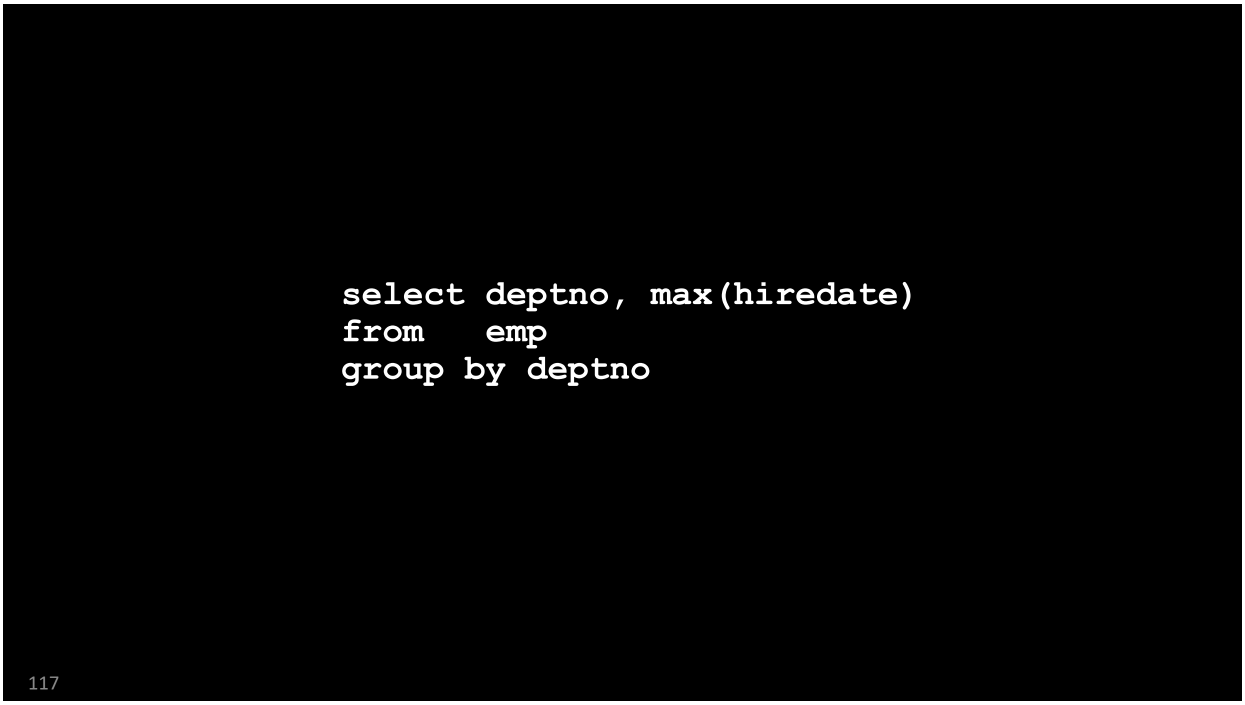
114



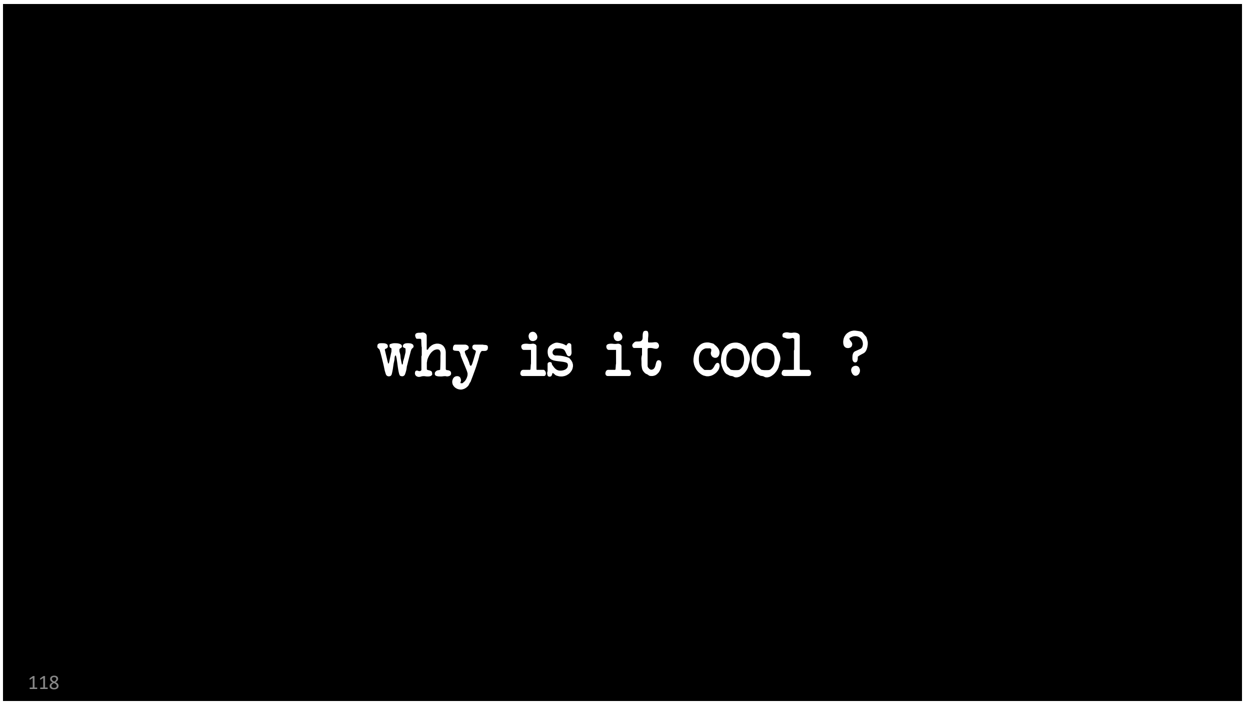
```
SQL> WITH last_hire AS
  2  ( select deptno, max(hiredate)
  3    from   emp
  4    group by deptno
  5  )
  6  select * from last_hire;

    DEPTNO MAX(HIRED
---------- ---------
        30 03-DEC-81
        20 12-JAN-83
        10 23-JAN-82
```



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



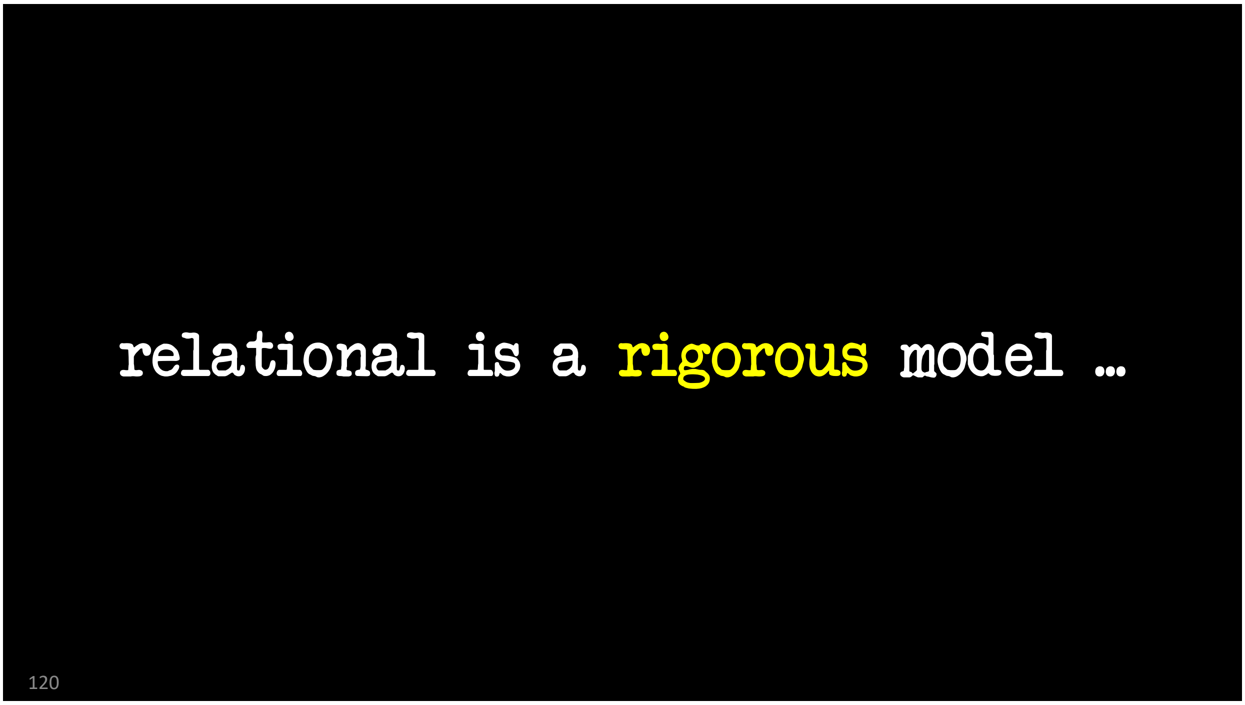
```
select deptno, max(hiredate)
from   emp
group by deptno
```



why is it cool ?

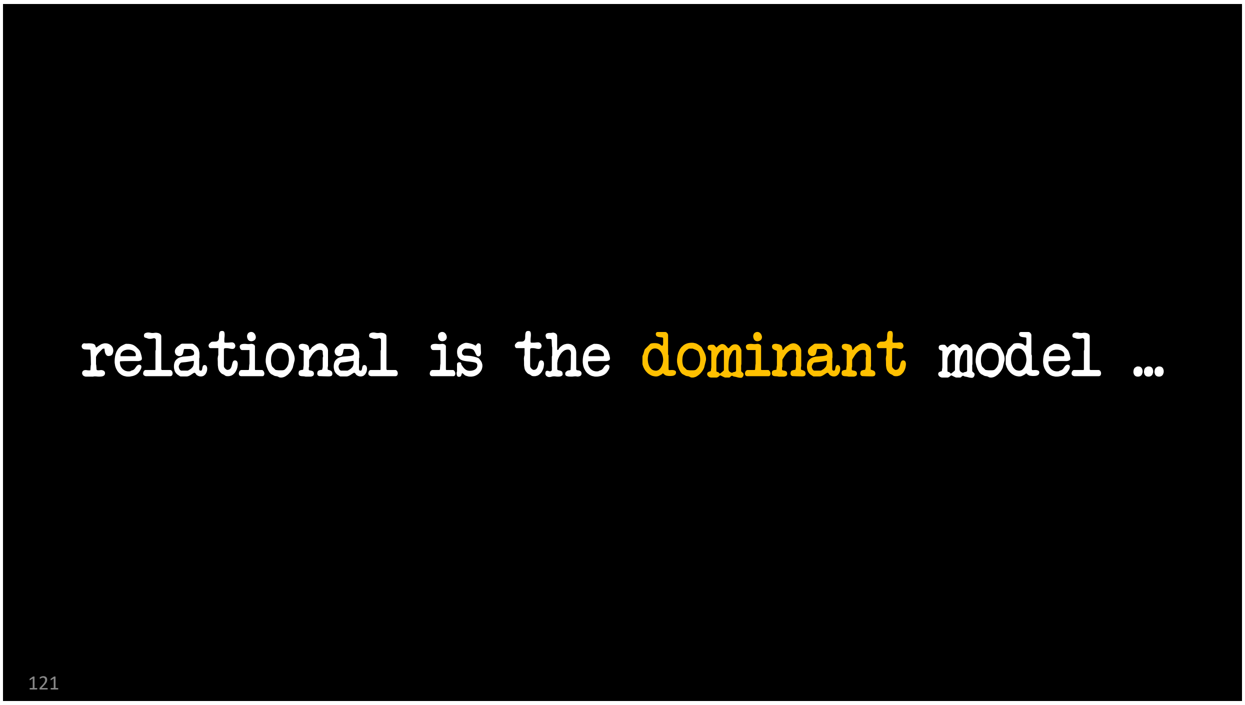
good solution metaphor

119

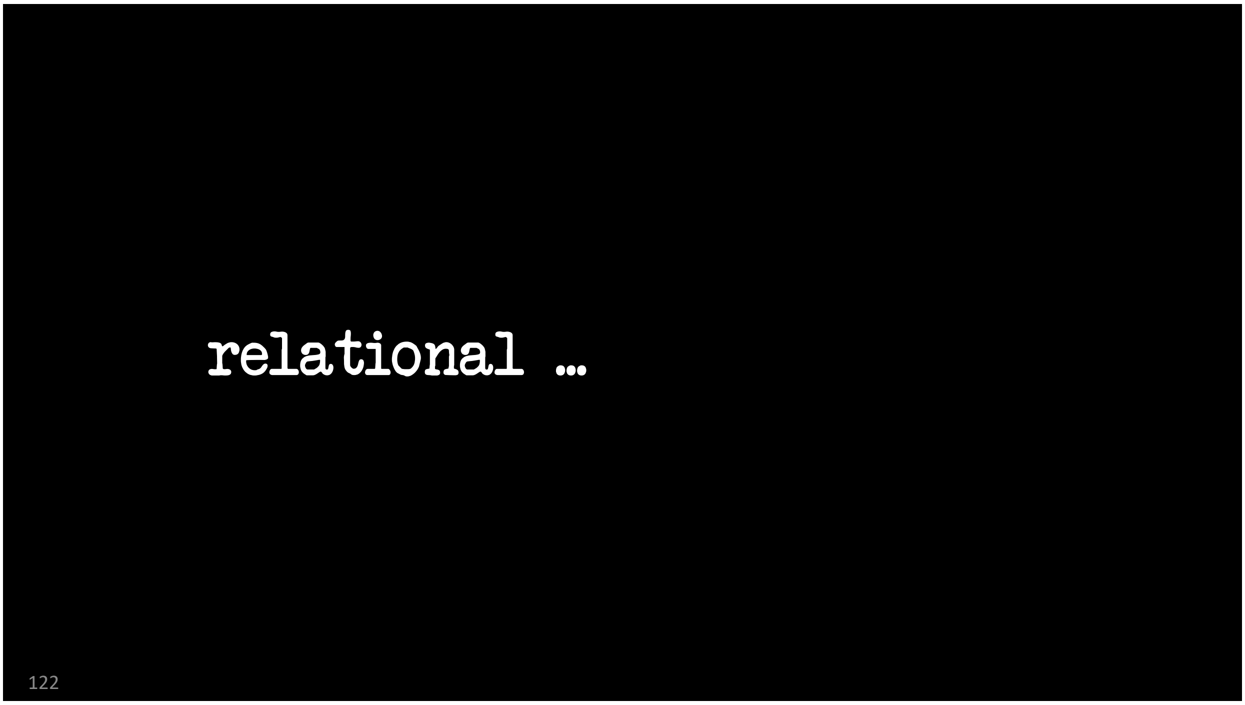


relational is a rigorous model …

120



relational is the **dominant** model …

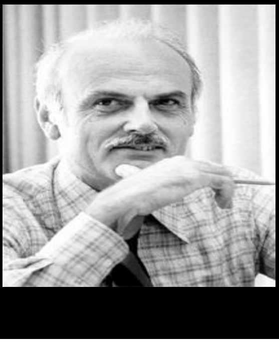


relational …

not our fault

123

Codd & Date



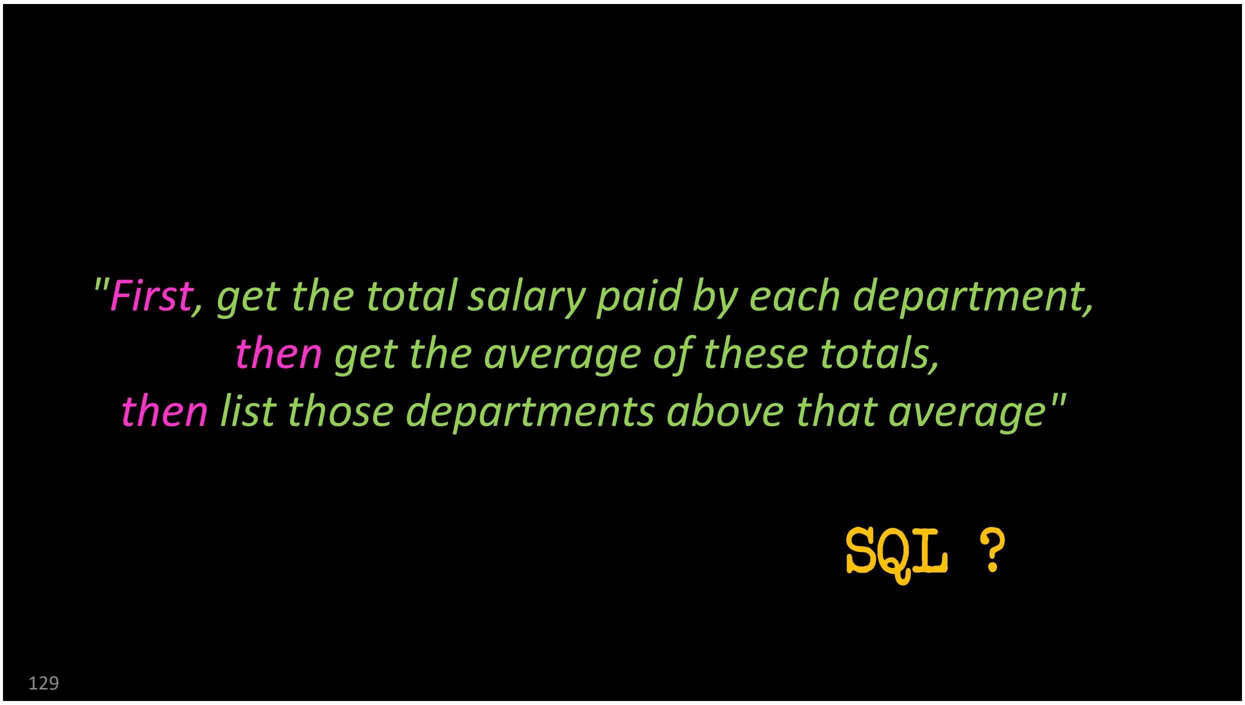
124

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

?

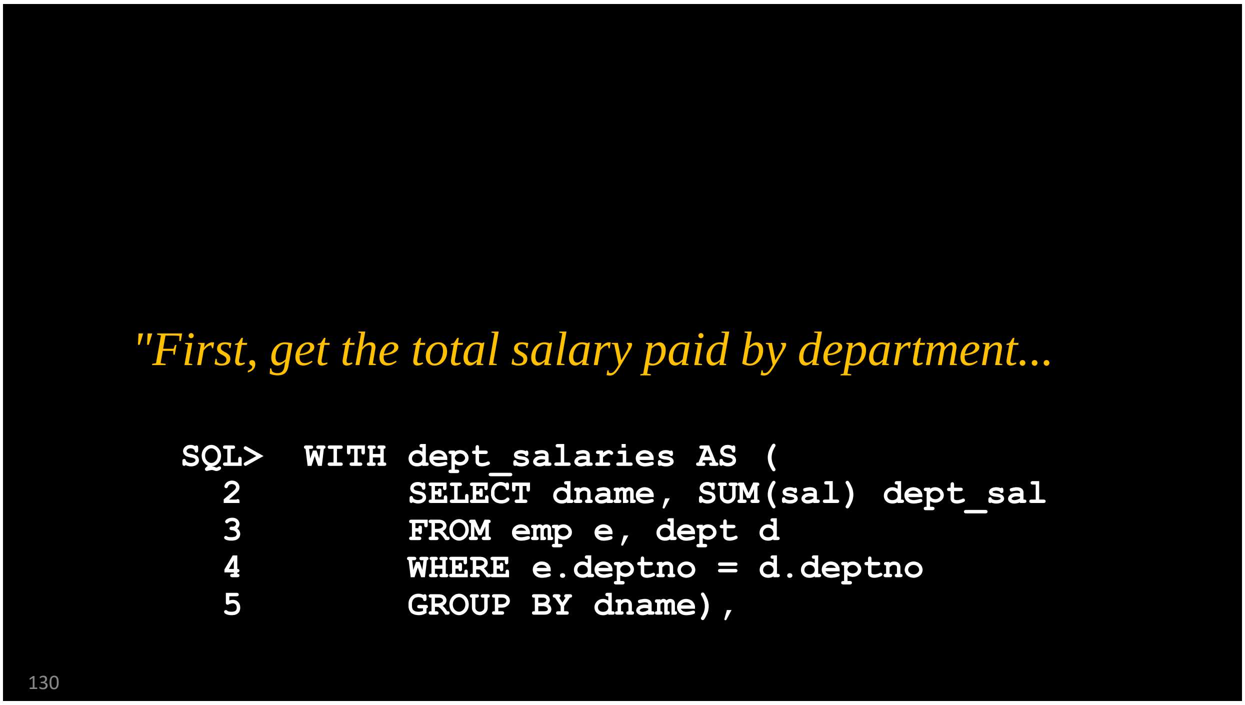
procedural world

step by step



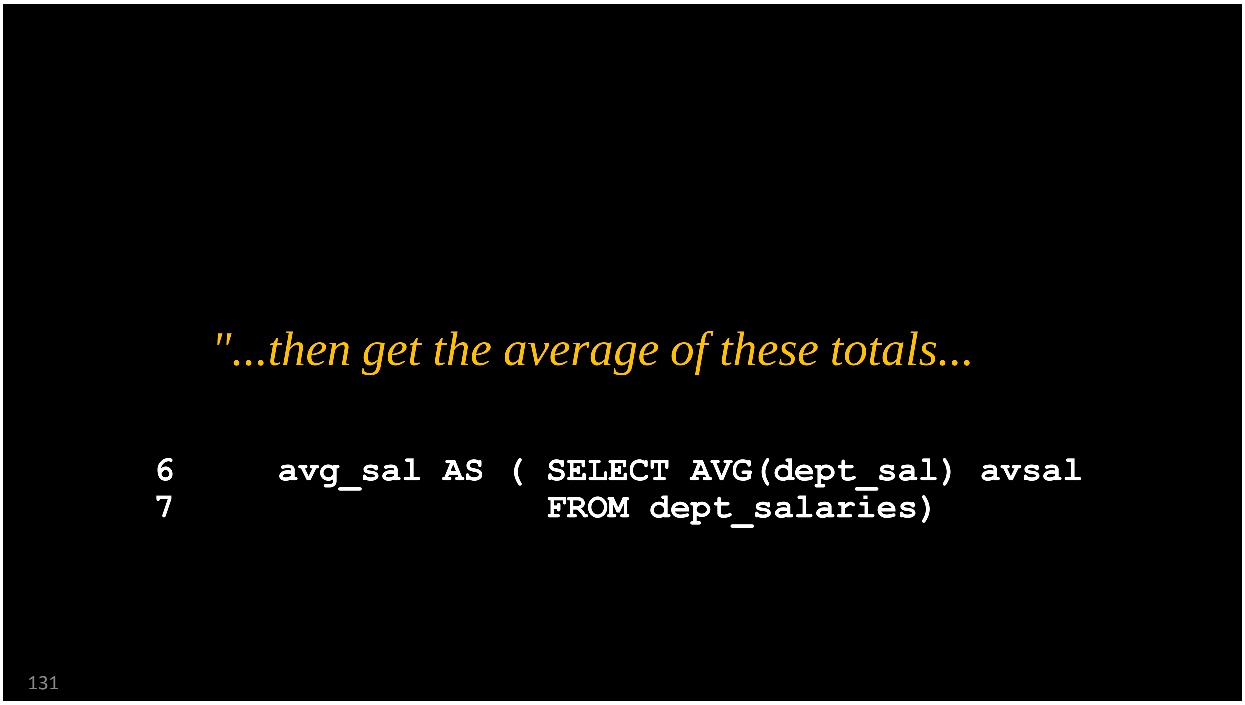
"First, get the total salary paid by each department,
then get the average of these totals,
then list those departments above that average"

SQL ?



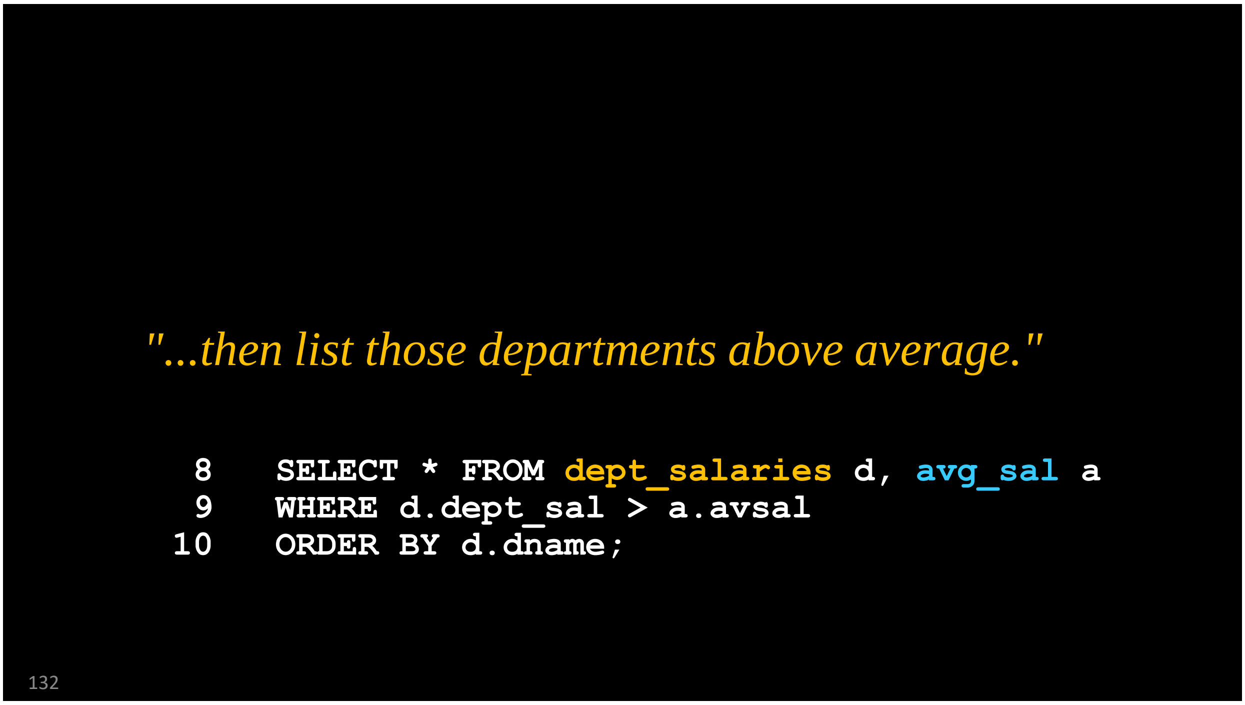
"First, get the total salary paid by department...

```
SQL>  WITH dept_salaries AS (
  2          SELECT dname, SUM(sal) dept_sal
  3          FROM emp e, dept d
  4          WHERE e.deptno = d.deptno
  5          GROUP BY dname),
```



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

```
avg_sal AS ( SELECT AVG(dept_sal) avsal
             FROM dept_salaries)
```

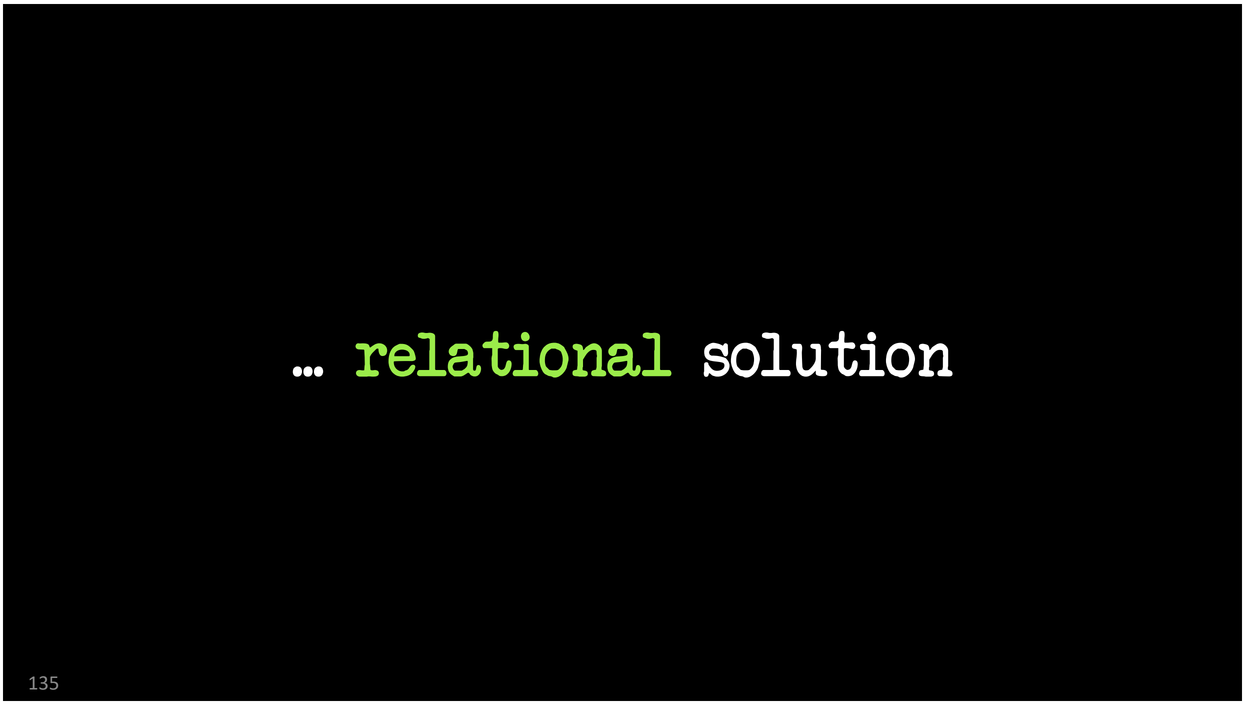


"...then list those departments above average."

```
SELECT * FROM dept_salaries d, avg_sal a
WHERE d.dept_sal > a.avsal
ORDER BY d.dname;
```

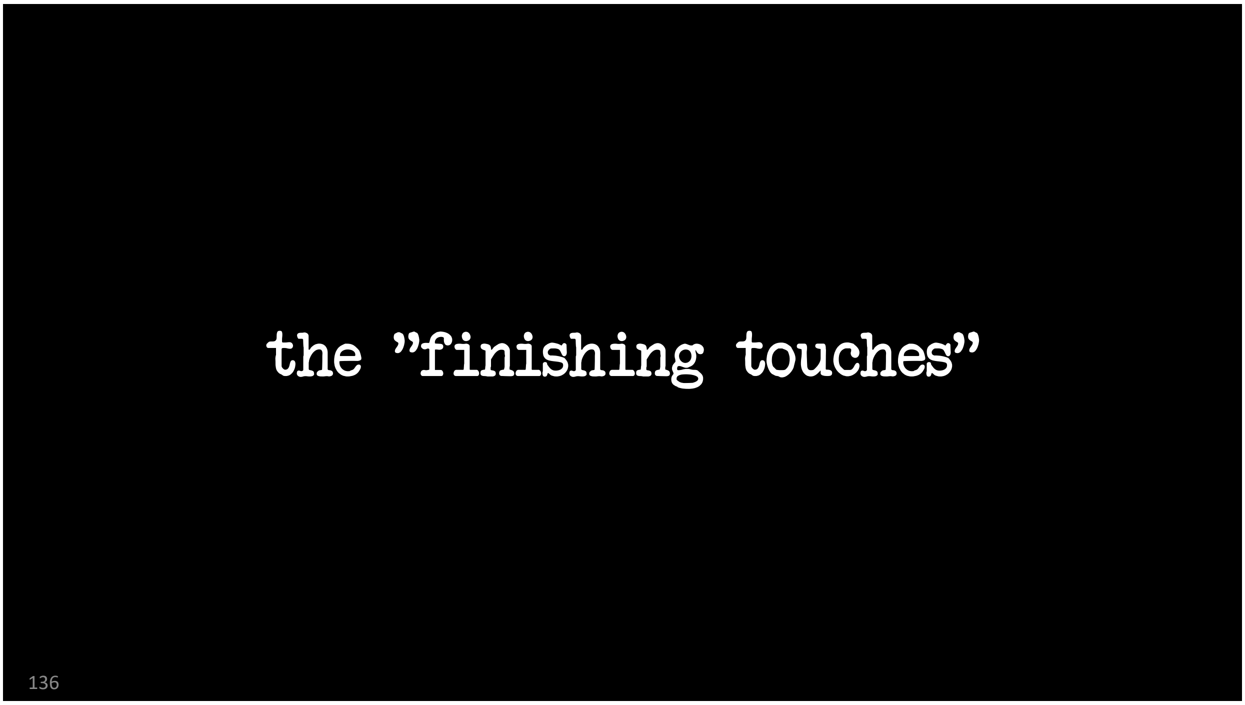
```
SQL>  WITH dept_salaries AS (
  2          SELECT dname, SUM(sal) dept_sal
  3          FROM emp e, dept d
  4          WHERE e.deptno = d.deptno
  5          GROUP BY dname),

  6      avg_sal AS ( SELECT AVG(dept_sal) avsal
  7                   FROM dept_salaries)

  8   SELECT * FROM dept_salaries d, avg_sal a
  9   WHERE d.dept_sal > a.avsal
 10   ORDER BY d.dname;
```

programmer's approach....



… relational solution

135

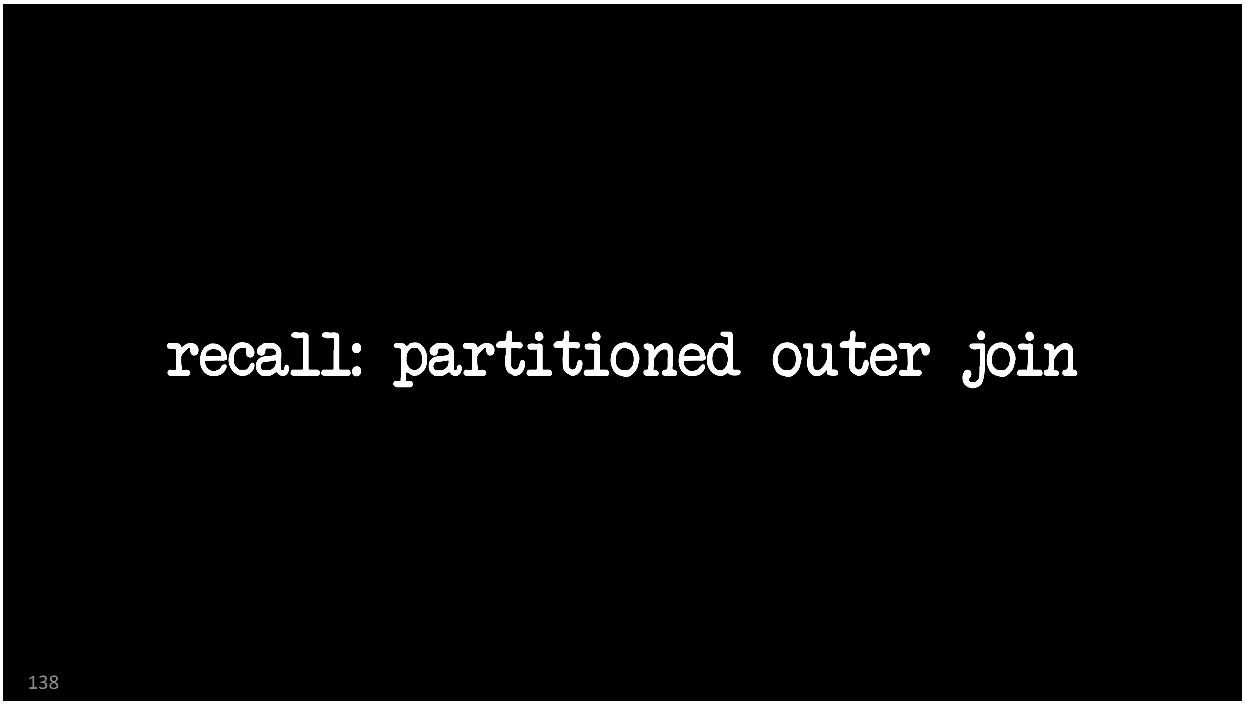


the ”finishing touches”

136

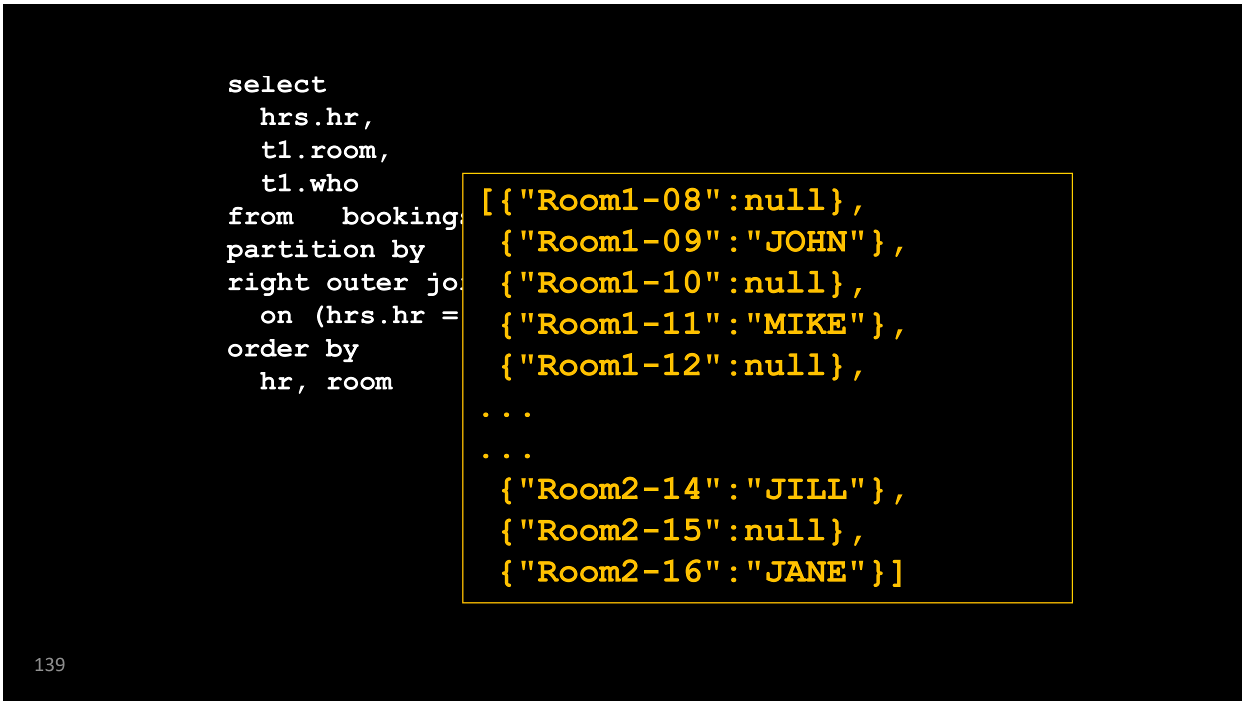
everyone loves JSON

137



recall: partitioned outer join

138



```
select
  hrs.hr,
  t1.room,
  t1.who
from   booking
partition by
right outer jo
  on (hrs.hr =
order by
  hr, room
```

```
[{"Room1-08":null},
 {"Room1-09":"JOHN"},
 {"Room1-10":null},
 {"Room1-11":"MIKE"},
 {"Room1-12":null},
...
...
 {"Room2-14":"JILL"},
 {"Room2-15":null},
 {"Room2-16":"JANE"}]
```

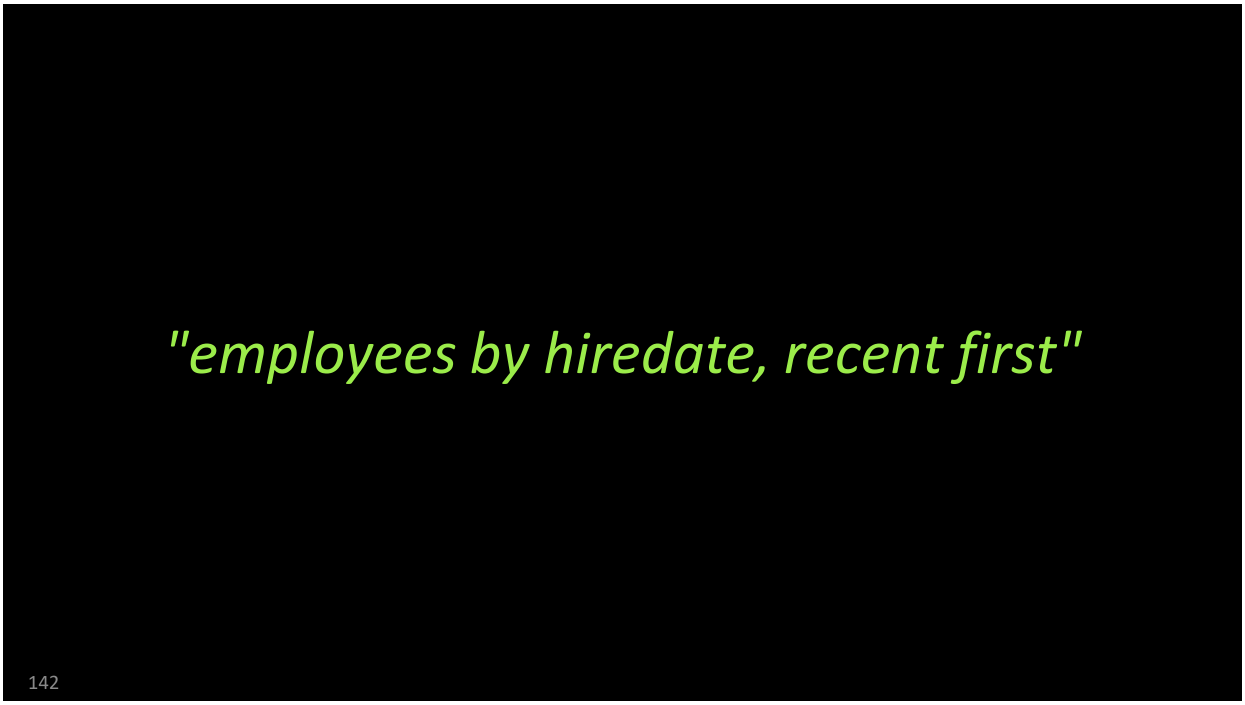
139

7

140

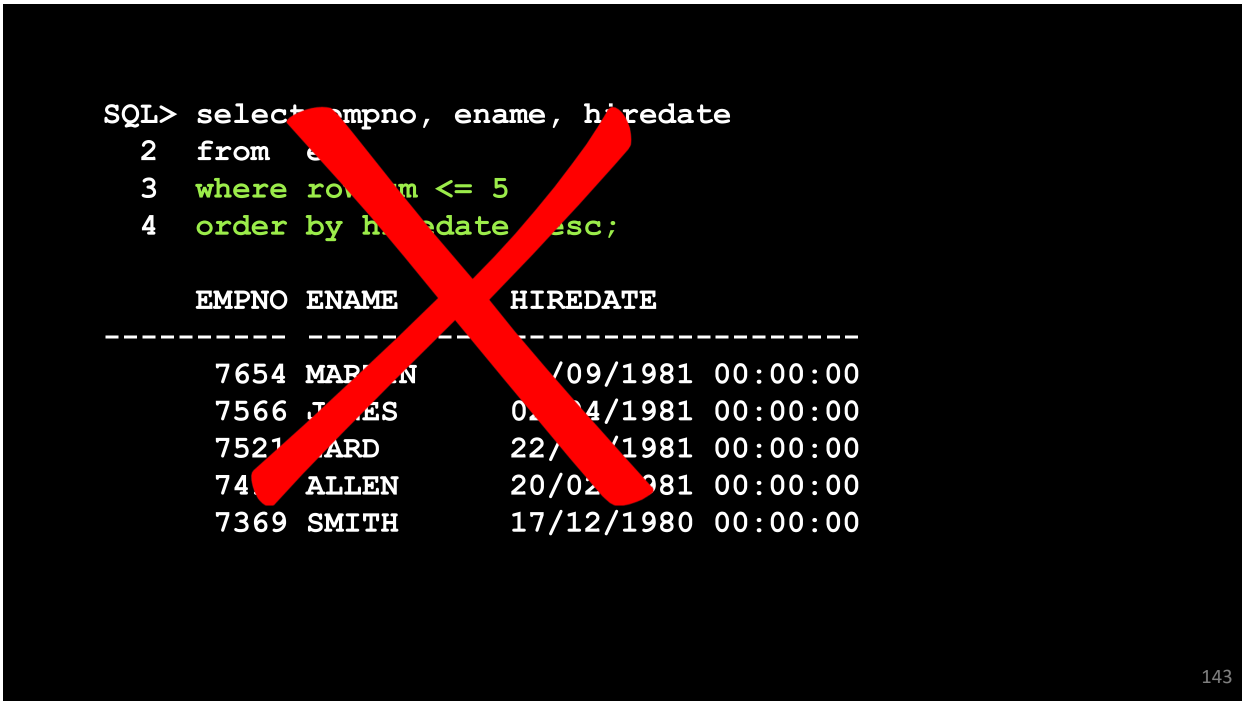
pagination

141



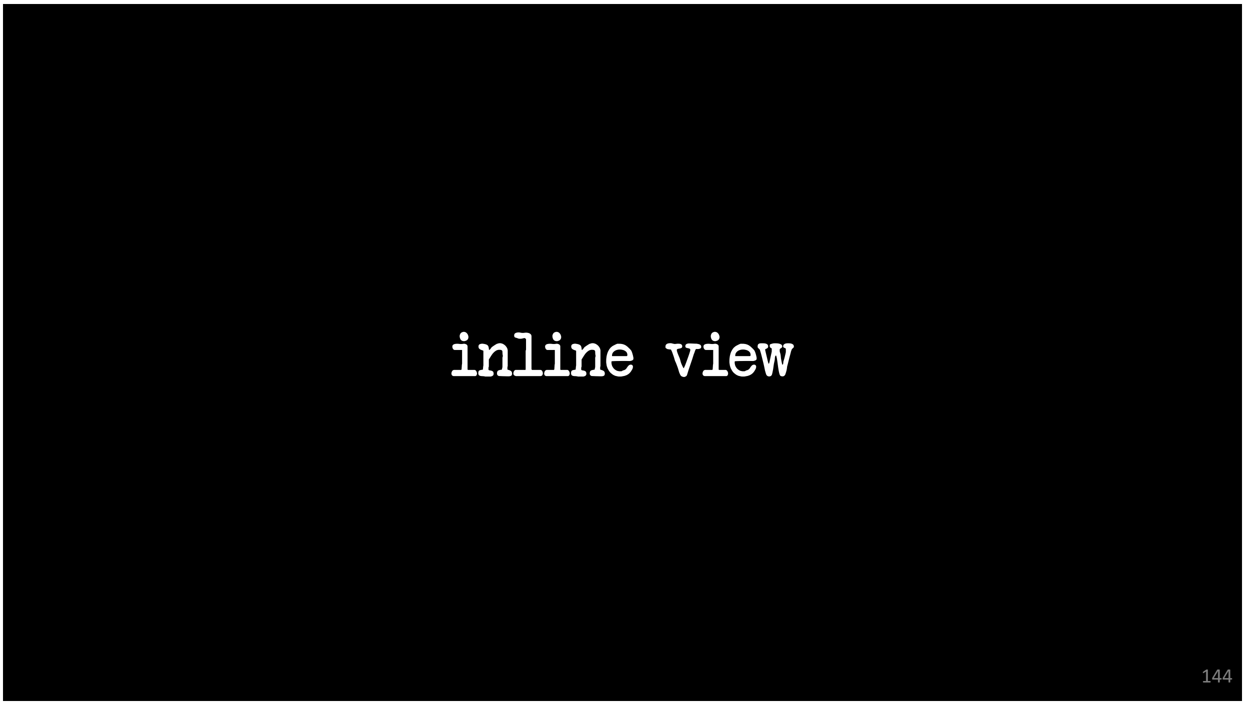
"employees by hiredate, recent first"

142



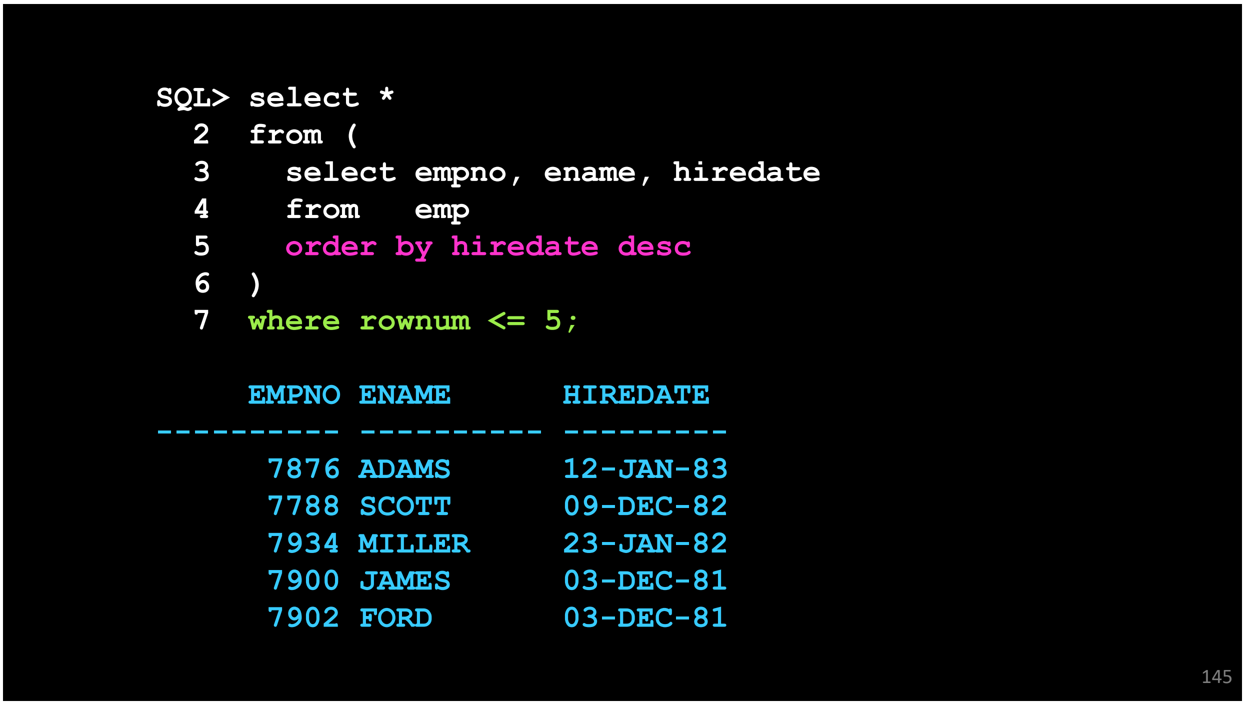
```
SQL> select empno, ename, hiredate
  2  from   emp
  3  where rownum <= 5
  4  order by hiredate desc;

     EMPNO ENAME      HIREDATE
---------- ---------- -------------------
      7654 MARTIN     28/09/1981 00:00:00
      7566 JONES      02/04/1981 00:00:00
      7521 WARD       22/02/1981 00:00:00
      7499 ALLEN      20/02/1981 00:00:00
      7369 SMITH      17/12/1980 00:00:00
```

143

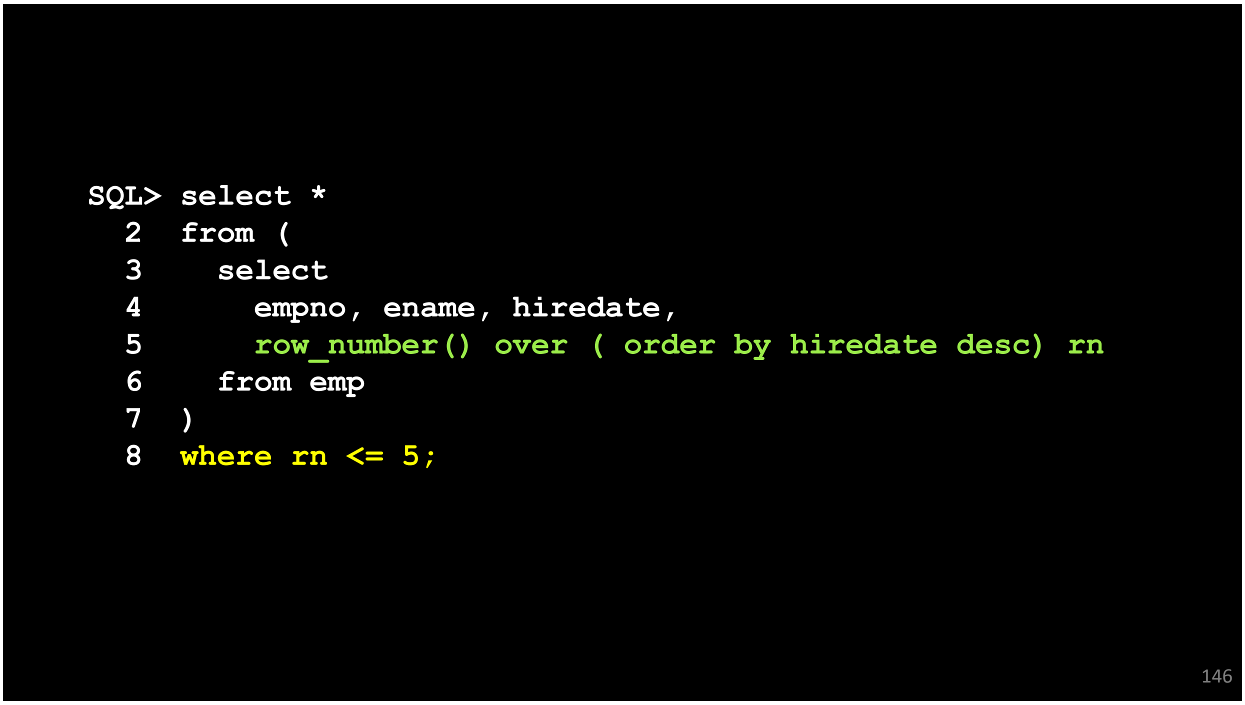


inline view

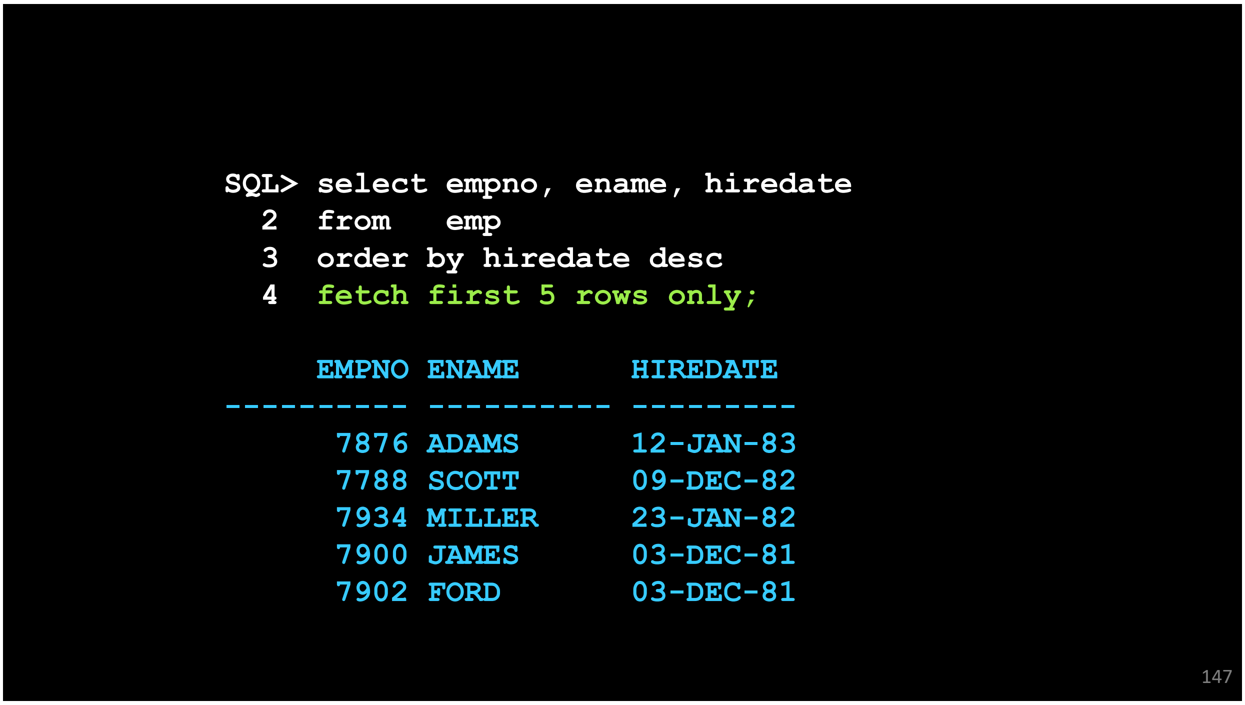
144



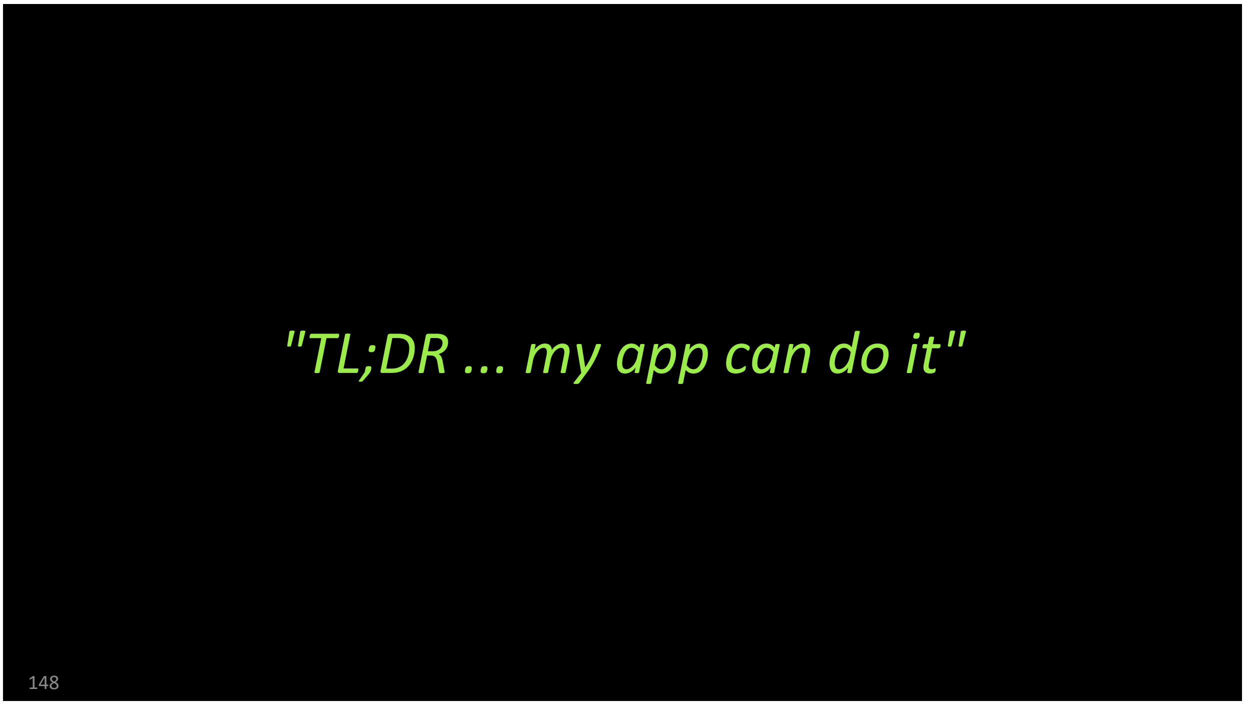
```
SQL> select *
  2  from (
  3    select empno, ename, hiredate
  4    from   emp
  5    order by hiredate desc
  6  )
  7  where rownum <= 5;

     EMPNO ENAME      HIREDATE
---------- ---------- ---------
      7876 ADAMS      12-JAN-83
      7788 SCOTT      09-DEC-82
      7934 MILLER     23-JAN-82
      7900 JAMES      03-DEC-81
      7902 FORD       03-DEC-81
```



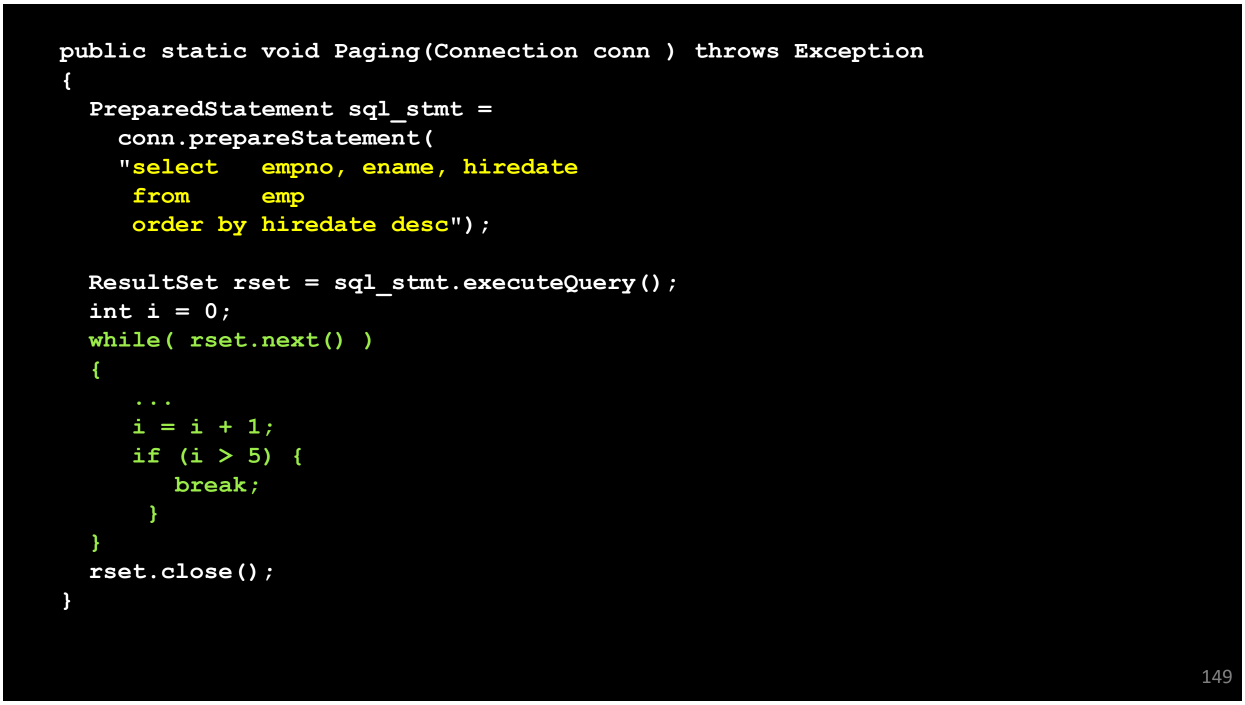
```
SQL> select *
  2  from (
  3    select
  4      empno, ename, hiredate,
  5      row_number() over ( order by hiredate desc) rn
  6    from emp
  7  )
  8  where rn <= 5;
```



```
SQL> select empno, ename, hiredate
  2  from   emp
  3  order by hiredate desc
  4  fetch first 5 rows only;

     EMPNO ENAME      HIREDATE
---------- ---------- ---------
      7876 ADAMS      12-JAN-83
      7788 SCOTT      09-DEC-82
      7934 MILLER     23-JAN-82
      7900 JAMES      03-DEC-81
      7902 FORD       03-DEC-81
```

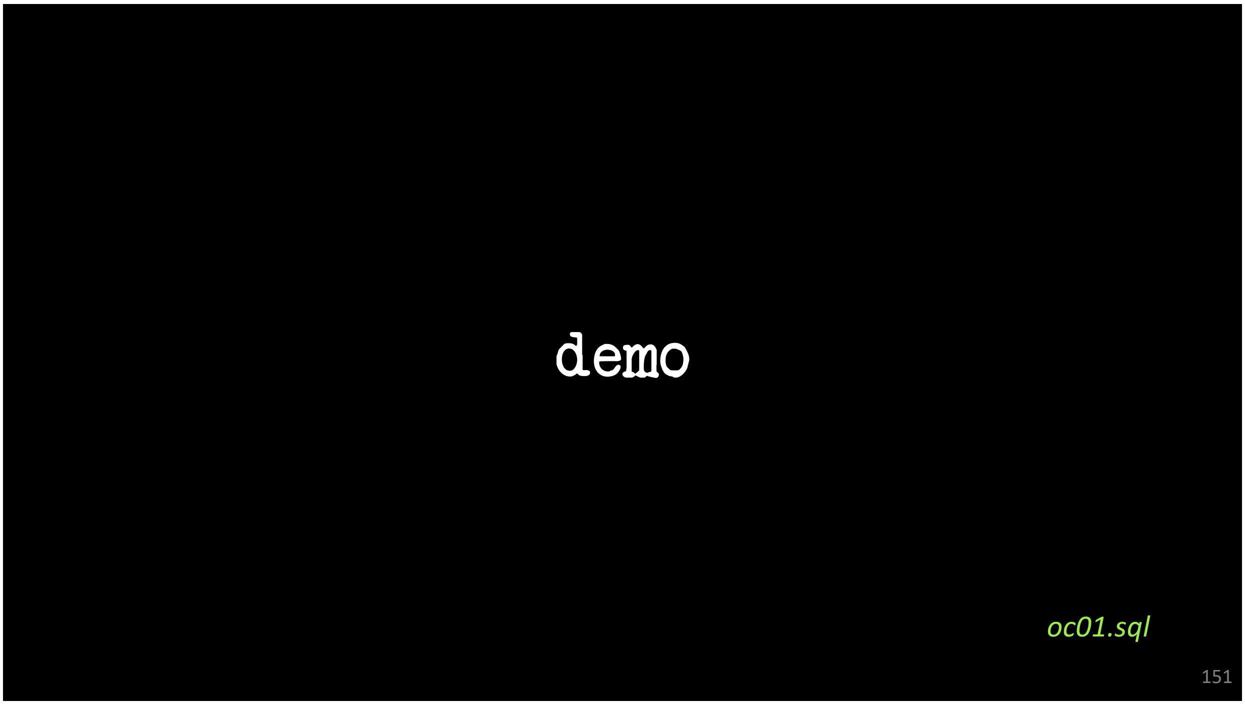


"TL;DR ... my app can do it"



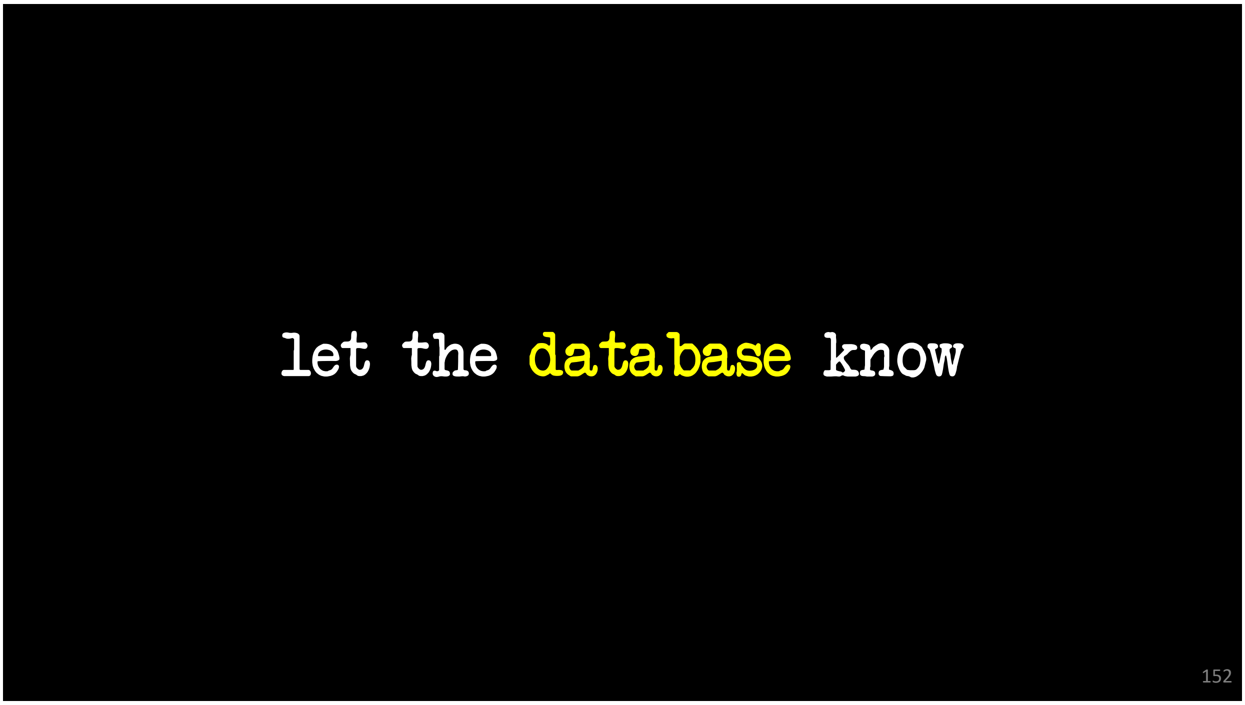
```
public static void Paging(Connection conn ) throws Exception
{
  PreparedStatement sql_stmt =
    conn.prepareStatement(
    "select   empno, ename, hiredate
     from     emp
     order by hiredate desc");

  ResultSet rset = sql_stmt.executeQuery();
  int i = 0;
  while( rset.next() )
  {
     ...
     i = i + 1;
     if (i > 5) {
        break;
      }
  }
  rset.close();
}
```

149

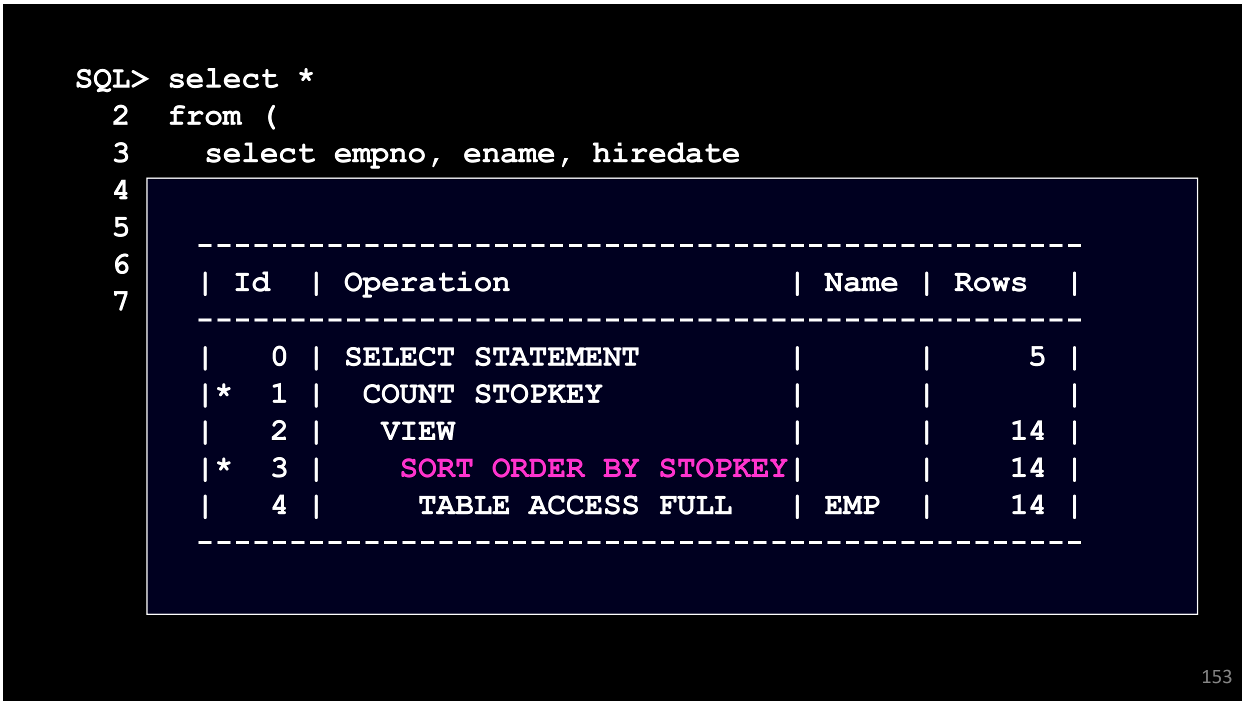


demo

oc01.sql

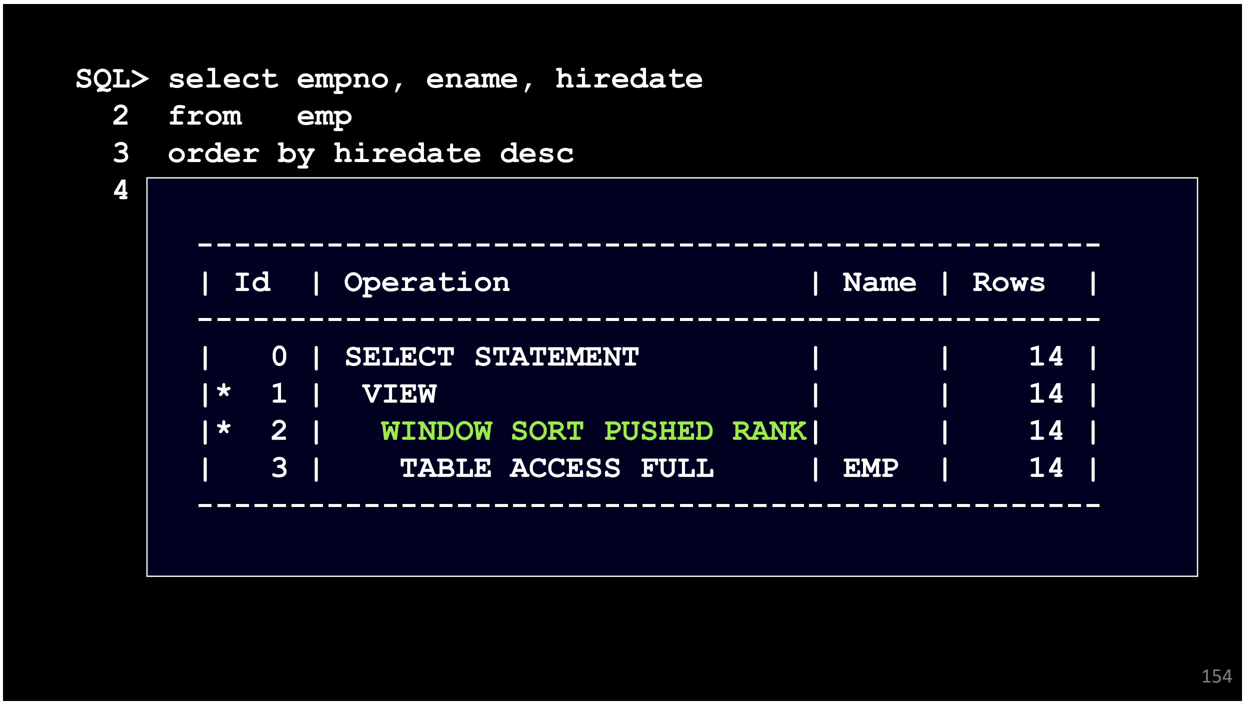


let the database know



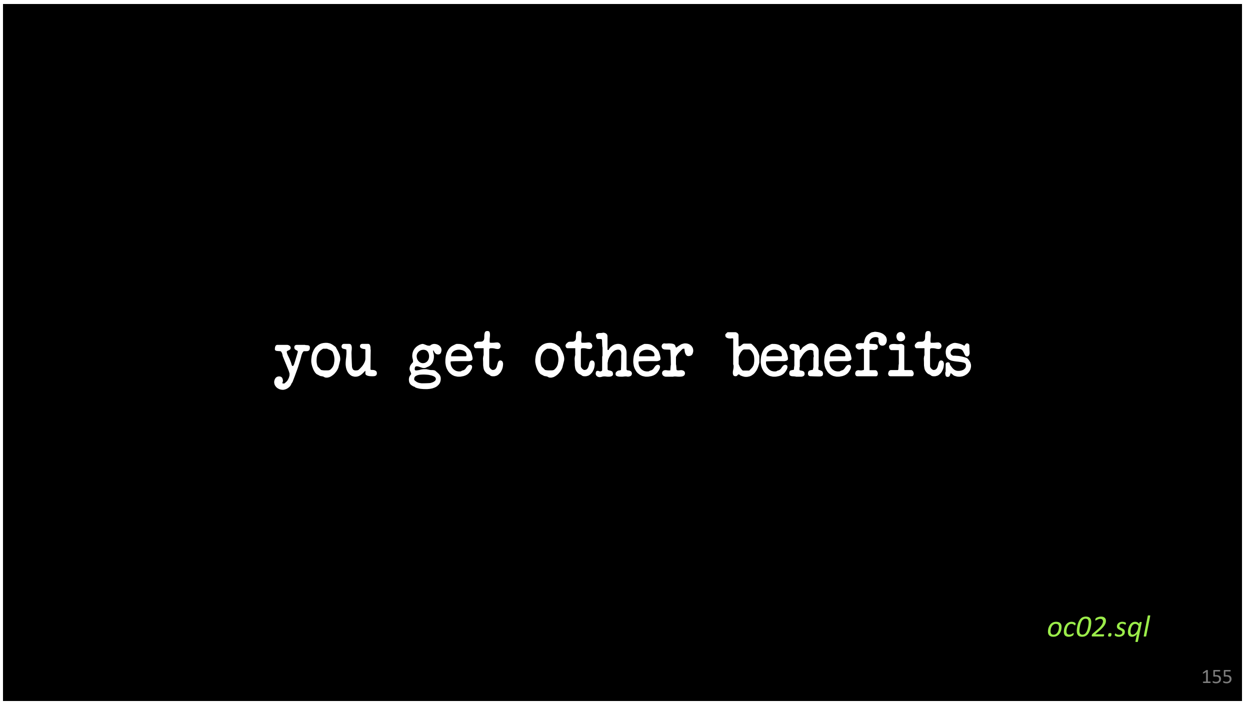
```
SQL> select *
  2  from (
  3    select empno, ename, hiredate
  4
  5
  6
  7

   ------------------------------------------------
   | Id  | Operation               | Name | Rows  |
   ------------------------------------------------
   |   0 | SELECT STATEMENT        |      |     5 |
   |*  1 |  COUNT STOPKEY          |      |       |
   |   2 |   VIEW                  |      |    14 |
   |*  3 |    SORT ORDER BY STOPKEY|      |    14 |
   |   4 |     TABLE ACCESS FULL   | EMP  |    14 |
   ------------------------------------------------
```

153



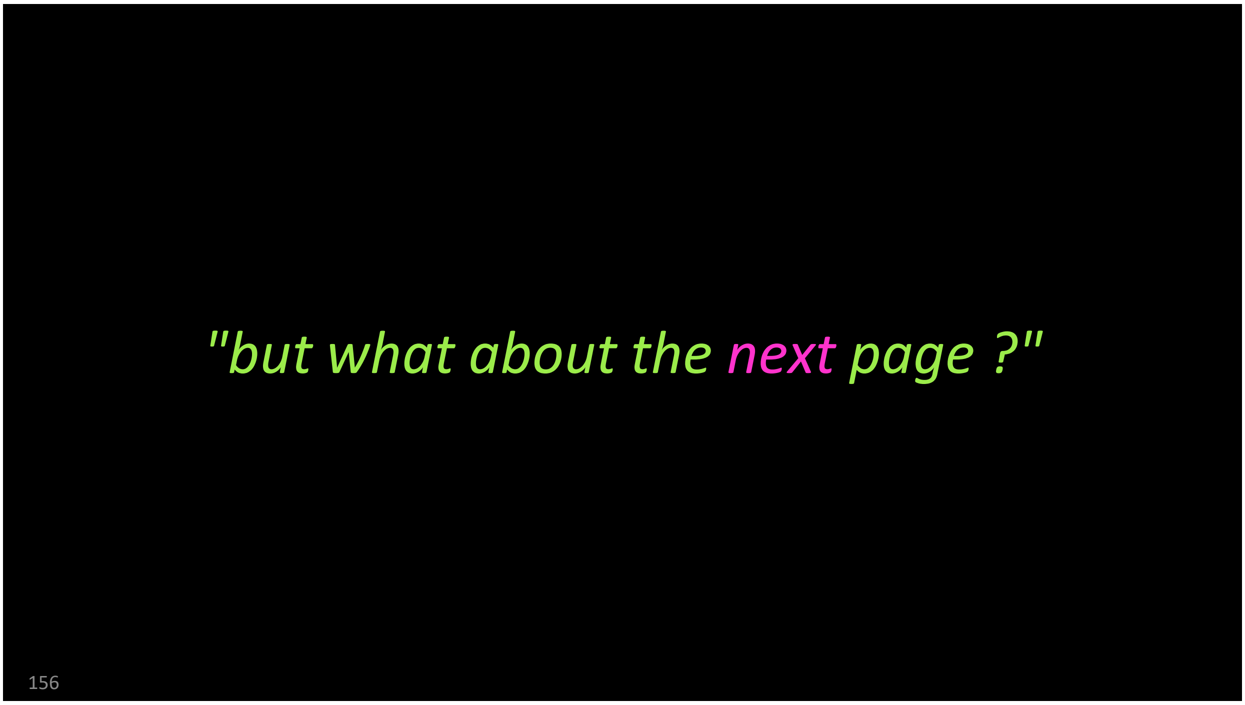
```
SQL> select empno, ename, hiredate
  2  from   emp
  3  order by hiredate desc
  4

   -------------------------------------------------
   | Id  | Operation                | Name | Rows  |
   -------------------------------------------------
   |   0 | SELECT STATEMENT         |      |    14 |
   |*  1 |  VIEW                    |      |    14 |
   |*  2 |   WINDOW SORT PUSHED RANK|      |    14 |
   |   3 |    TABLE ACCESS FULL     | EMP  |    14 |
   -------------------------------------------------
```

154



you get other benefits

oc02.sql



"but what about the next page ?"

please

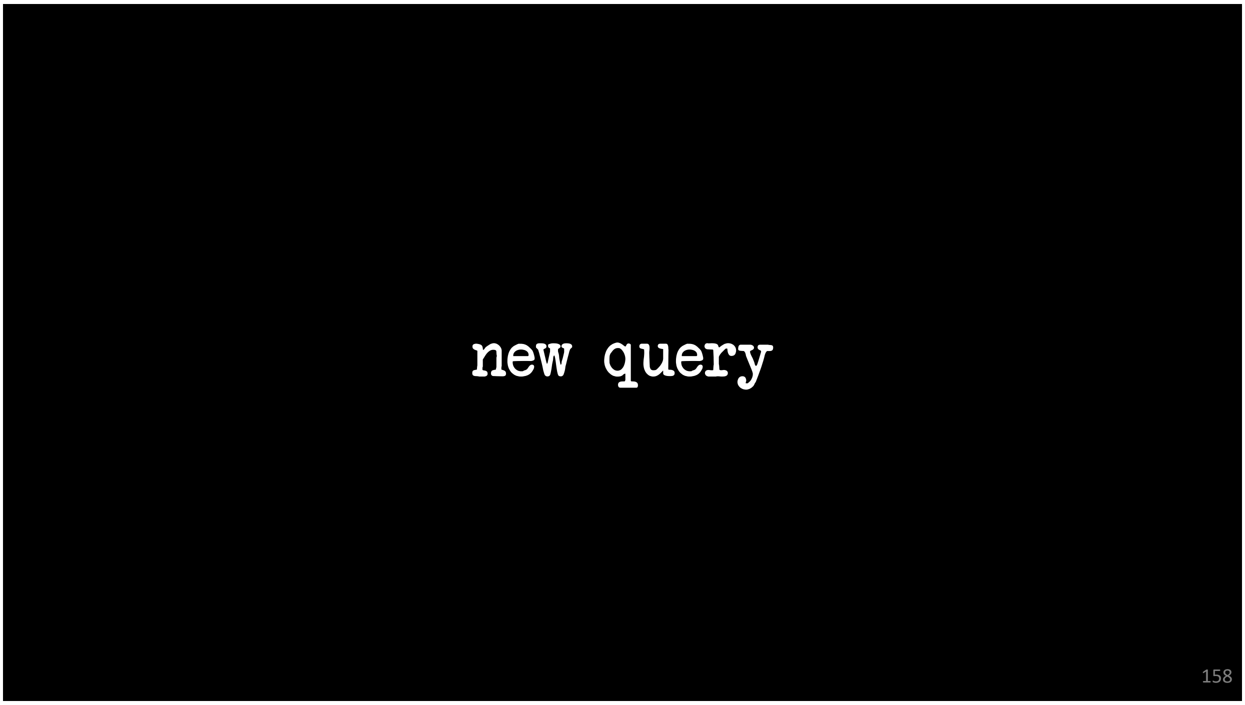
forget

the

next

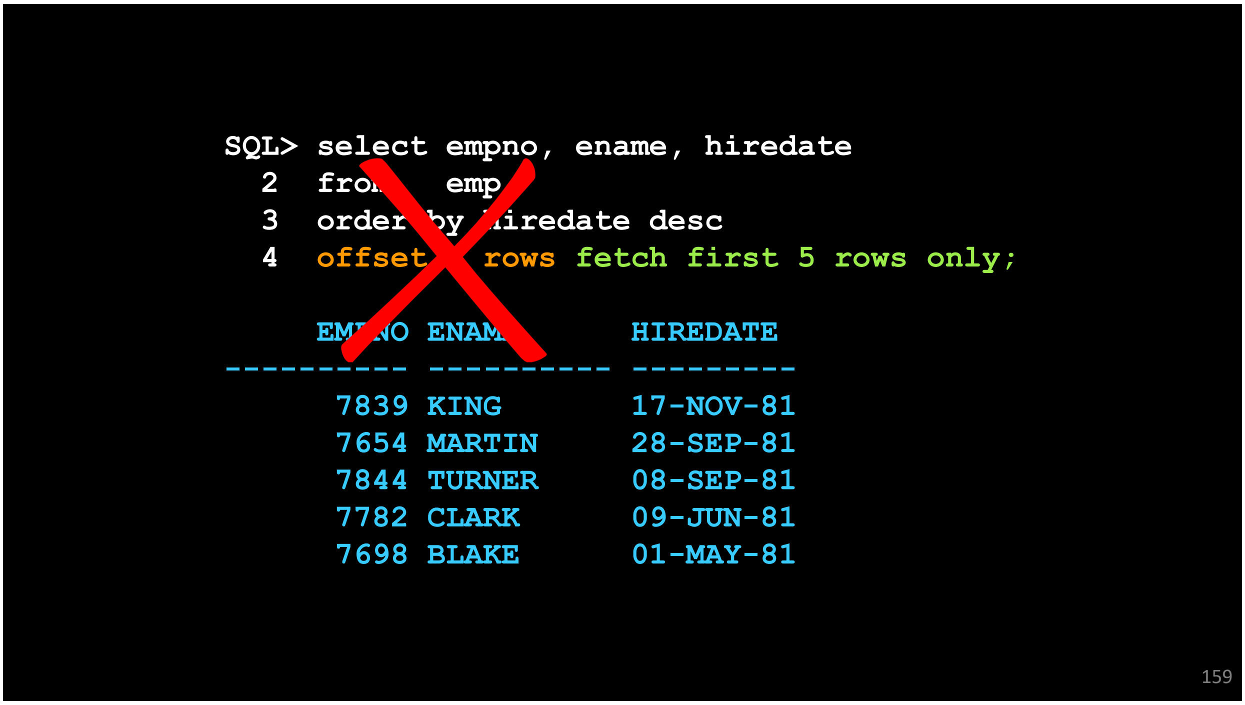
page

157



new query

158

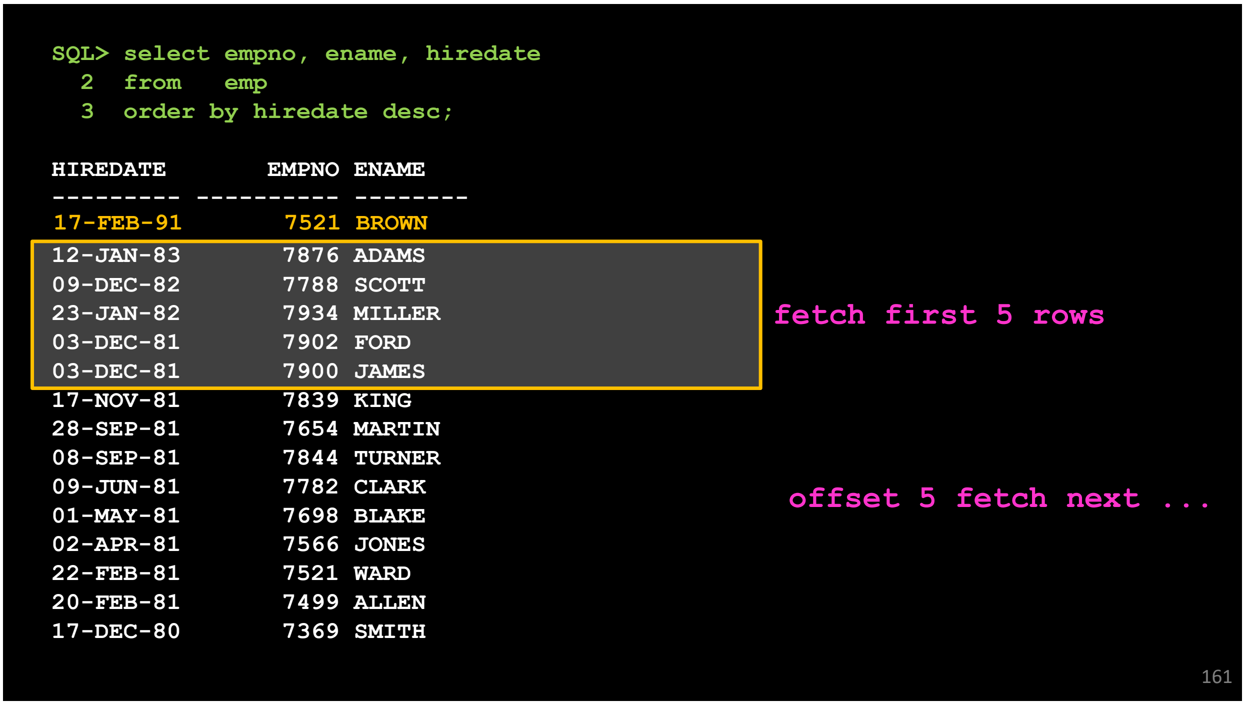


```
SQL> select empno, ename, hiredate
  2  from   emp
  3  order by hiredate desc
  4  offset   rows fetch first 5 rows only;

     EMPNO ENAME      HIREDATE
---------- ---------- ---------
      7839 KING       17-NOV-81
      7654 MARTIN     28-SEP-81
      7844 TURNER     08-SEP-81
      7782 CLARK      09-JUN-81
      7698 BLAKE      01-MAY-81
```

159

forget the OFFSET clause

160

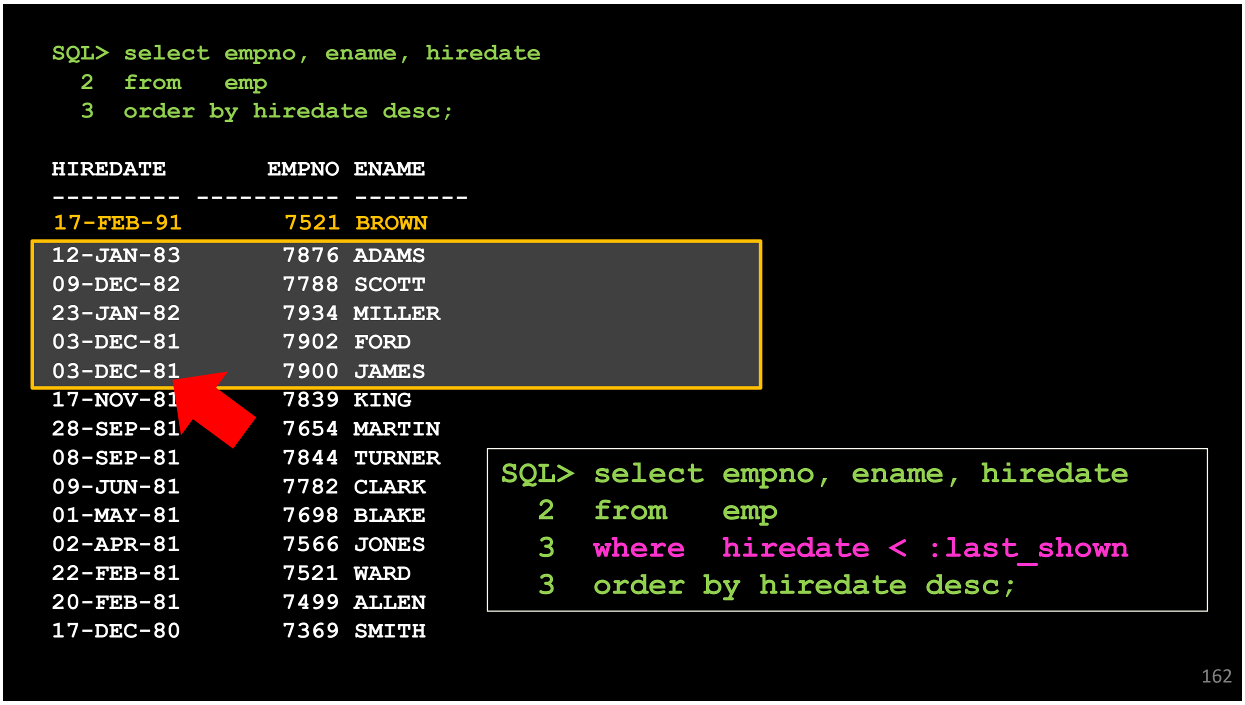


```
SQL> select empno, ename, hiredate
  2  from   emp
  3  order by hiredate desc;

HIREDATE        EMPNO ENAME
--------- ---------- --------
17-FEB-91        7521 BROWN
12-JAN-83        7876 ADAMS
09-DEC-82        7788 SCOTT
23-JAN-82        7934 MILLER
03-DEC-81        7902 FORD
03-DEC-81        7900 JAMES
17-NOV-81        7839 KING
28-SEP-81        7654 MARTIN
08-SEP-81        7844 TURNER
09-JUN-81        7782 CLARK
01-MAY-81        7698 BLAKE
02-APR-81        7566 JONES
22-FEB-81        7521 WARD
20-FEB-81        7499 ALLEN
17-DEC-80        7369 SMITH
```

fetch first 5 rows

offset 5 fetch next ...

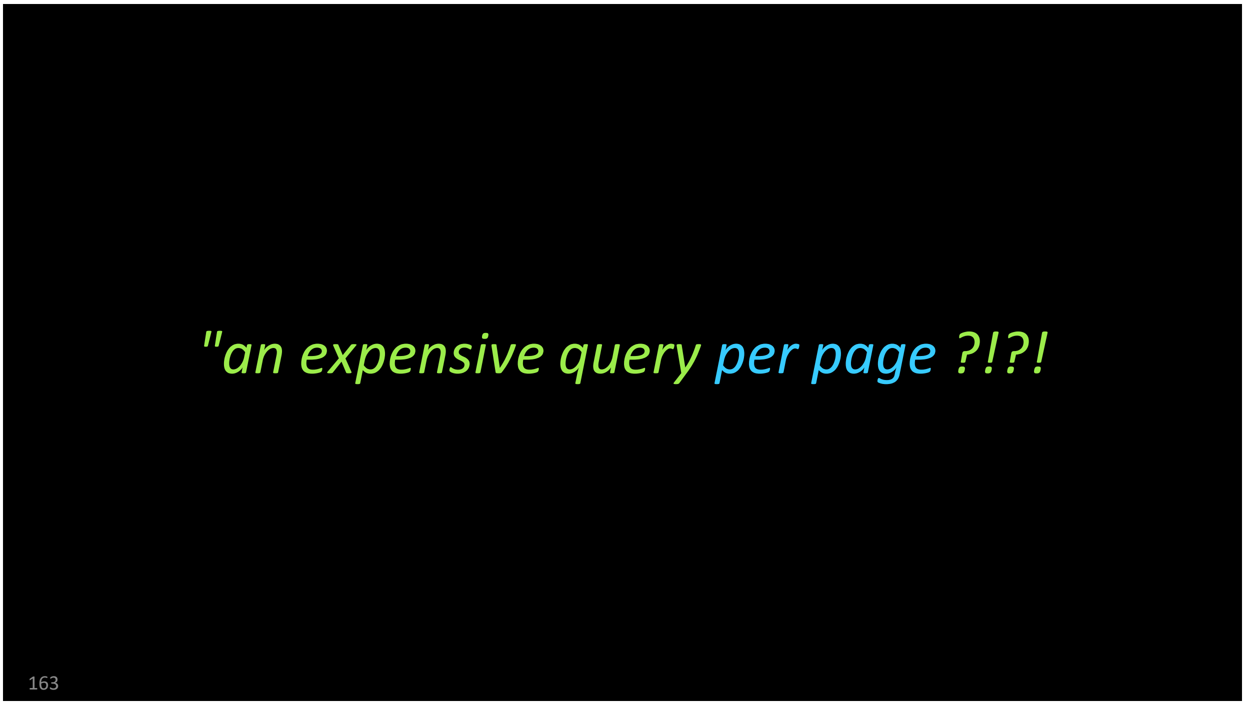
161



```
SQL> select empno, ename, hiredate
  2  from   emp
  3  order by hiredate desc;

HIREDATE        EMPNO ENAME
--------- ---------- --------
17-FEB-91        7521 BROWN
12-JAN-83        7876 ADAMS
09-DEC-82        7788 SCOTT
23-JAN-82        7934 MILLER
03-DEC-81        7902 FORD
03-DEC-81        7900 JAMES
17-NOV-81        7839 KING
28-SEP-81        7654 MARTIN
08-SEP-81        7844 TURNER
09-JUN-81        7782 CLARK
01-MAY-81        7698 BLAKE
02-APR-81        7566 JONES
22-FEB-81        7521 WARD
20-FEB-81        7499 ALLEN
17-DEC-80        7369 SMITH
```

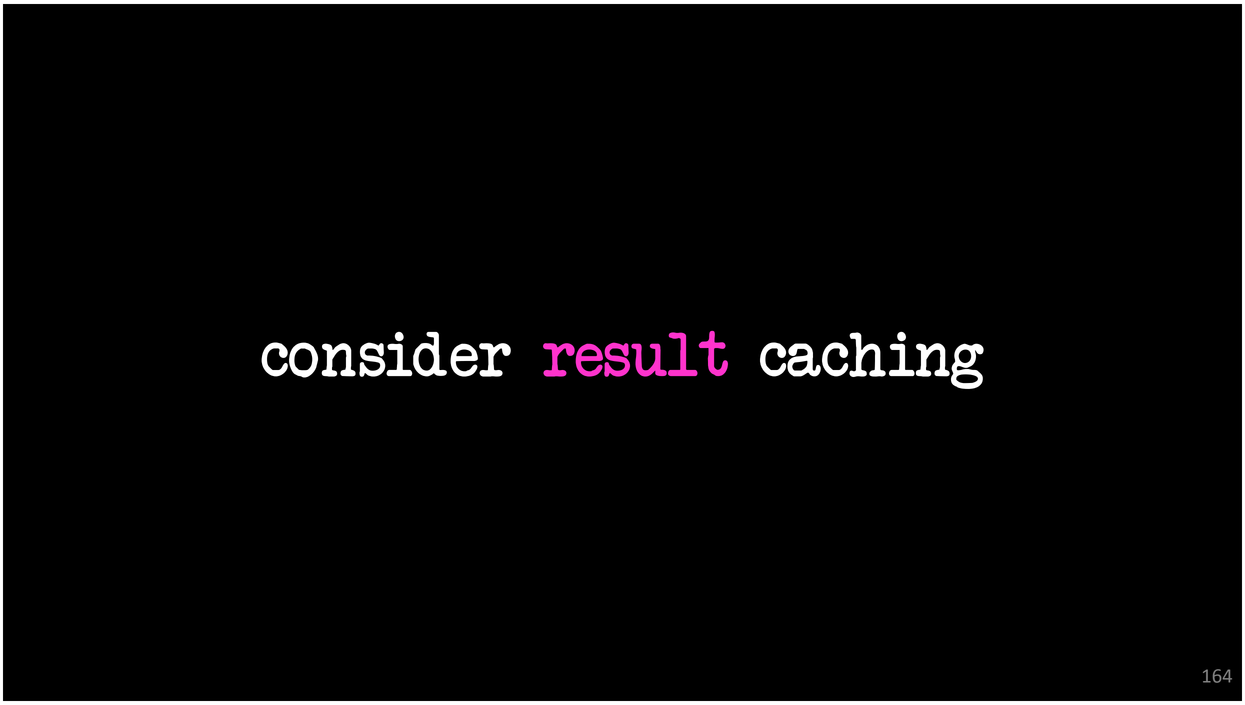
```
SQL> select empno, ename, hiredate
  2  from   emp
  3  where  hiredate < :last_shown
  3  order by hiredate desc;
```

162



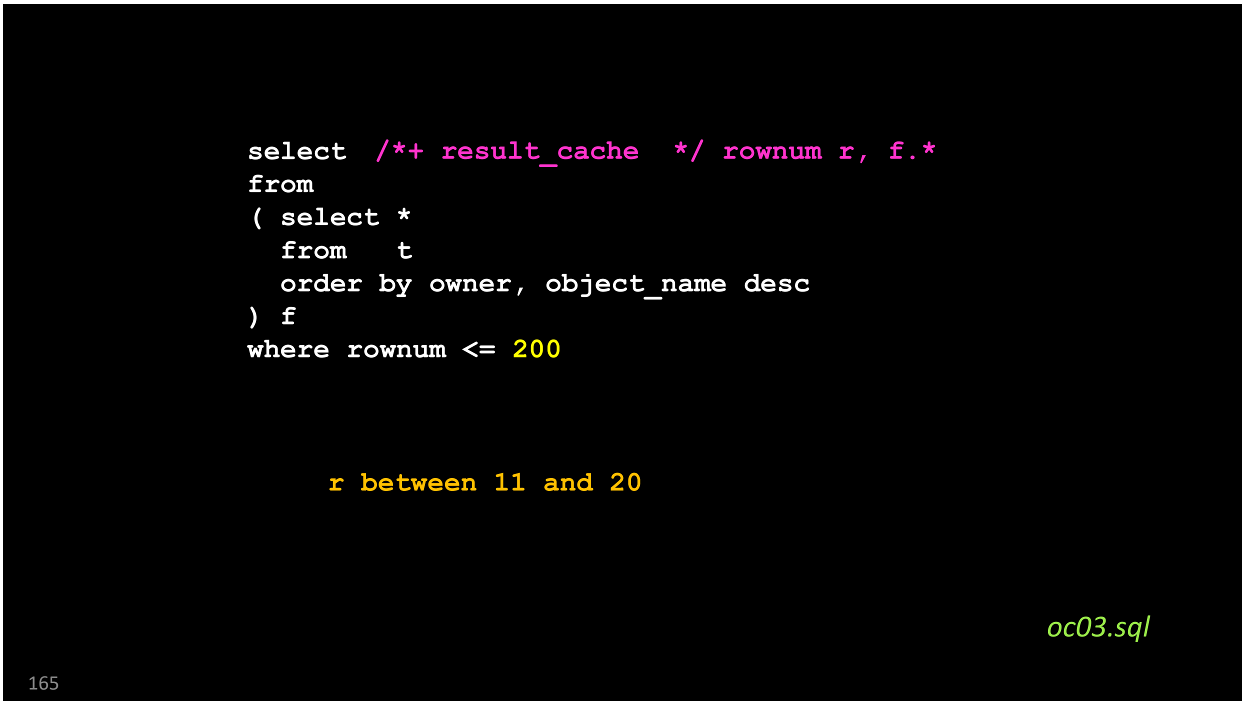
"an expensive query per page ?!?!

163



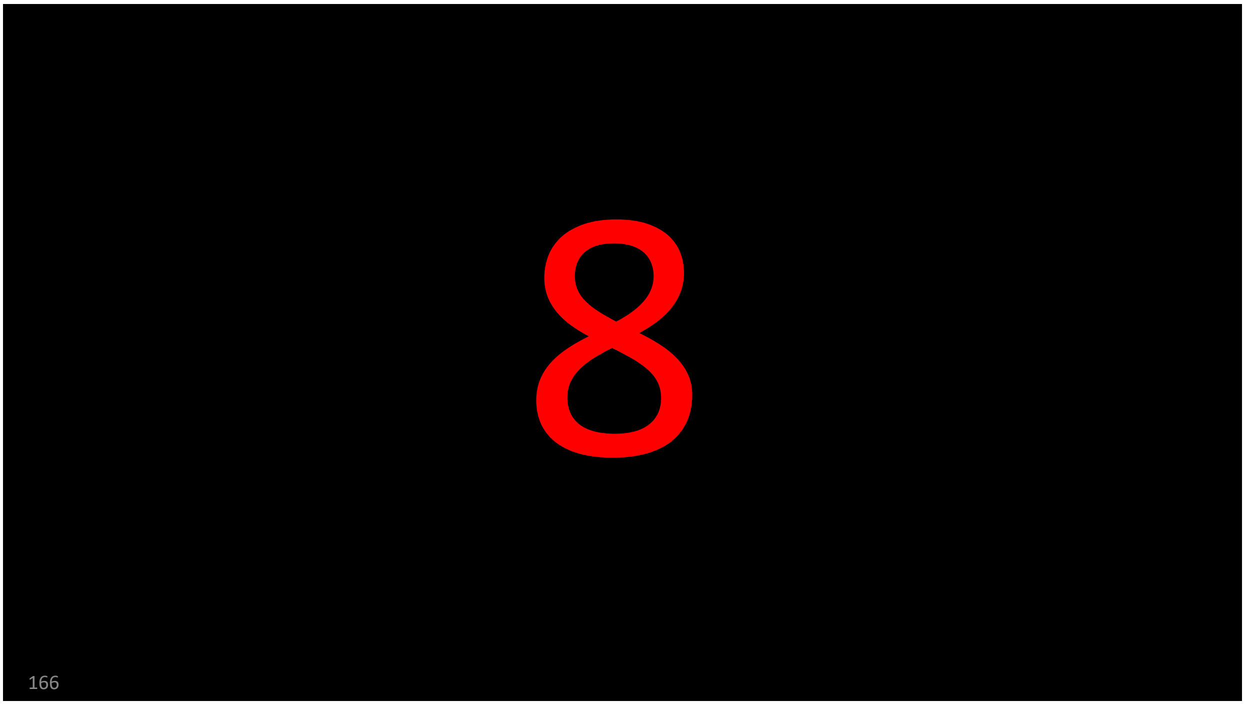
consider result caching

164

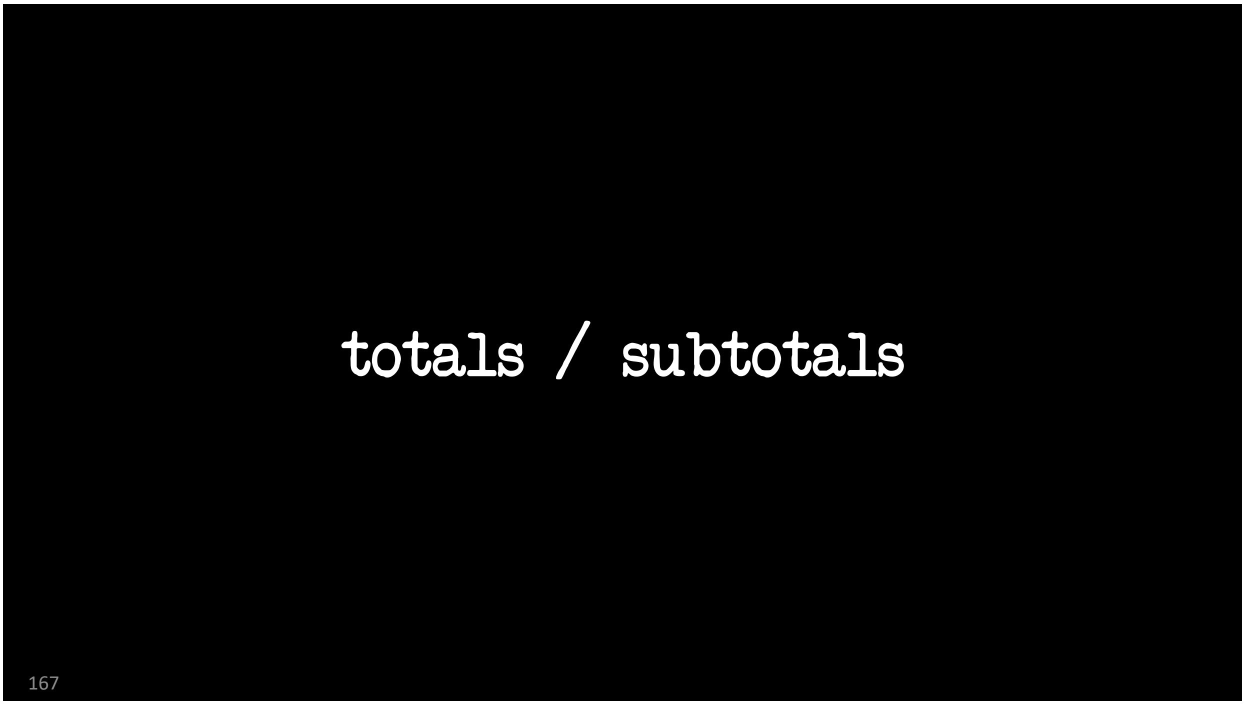


```
select  /*+ result_cache  */ rownum r, f.*
from
( select *
  from   t
  order by owner, object_name desc
) f
where rownum <= 200


     r between 11 and 20
```

oc03.sql



8

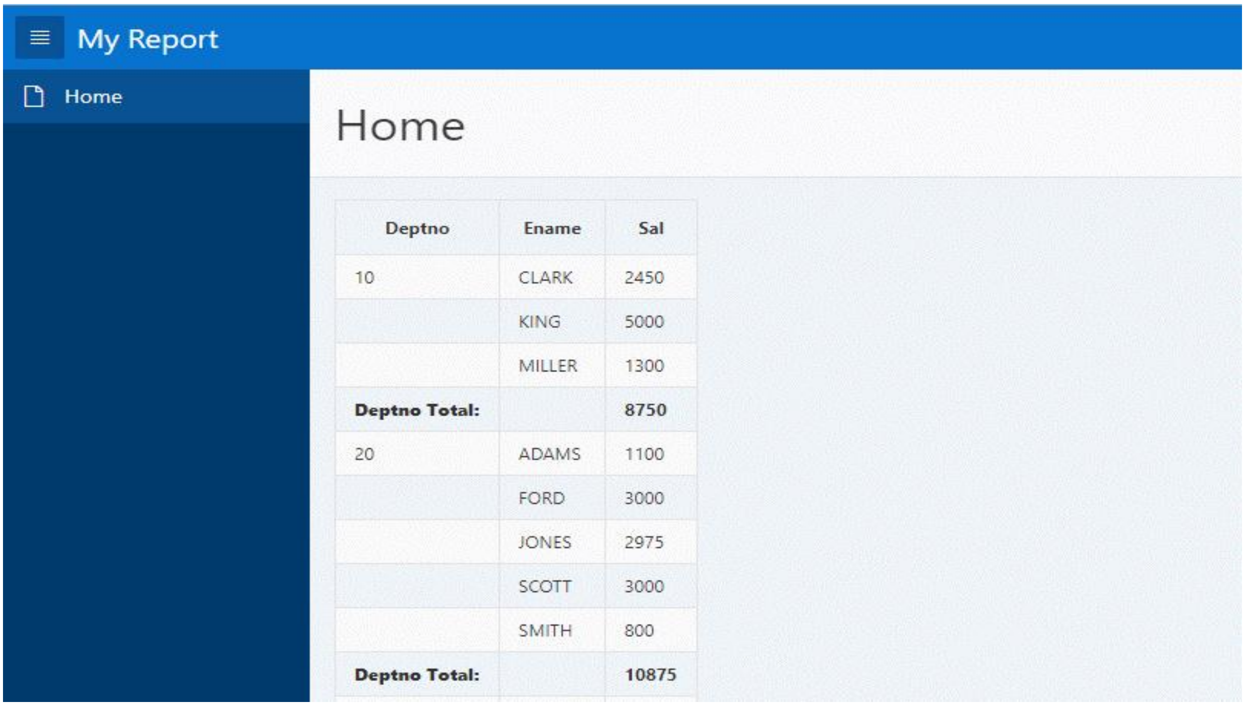


totals / subtotals

167

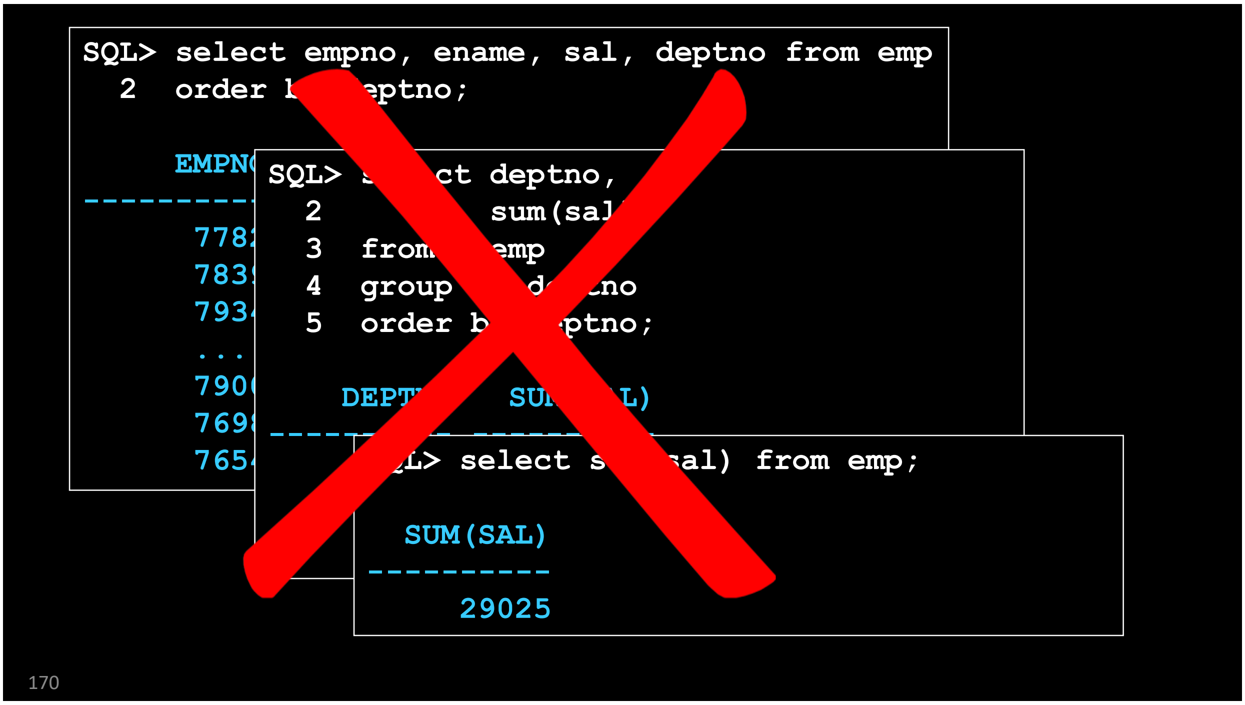
*"Employee salary list,
plus department total,
plus grand total"*

168



Home

Deptno	Ename	Sal
10	CLARK	2450
	KING	5000
	MILLER	1300
Deptno Total:		**8750**
20	ADAMS	1100
	FORD	3000
	JONES	2975
	SCOTT	3000
	SMITH	800
Deptno Total:		**10875**



```
SQL> select empno, ename, sal, deptno from emp
  2  order l   eptno;

     EMPN
----------
      778
      783
      793
      . . .
      790
      769
      765
```

```
SQL> s    ct deptno,
  2          sum(sal
  3  from    emp
  4  group    d   no
  5  order b    ptno;

    DEPT      SU    L)
--------- ------
```

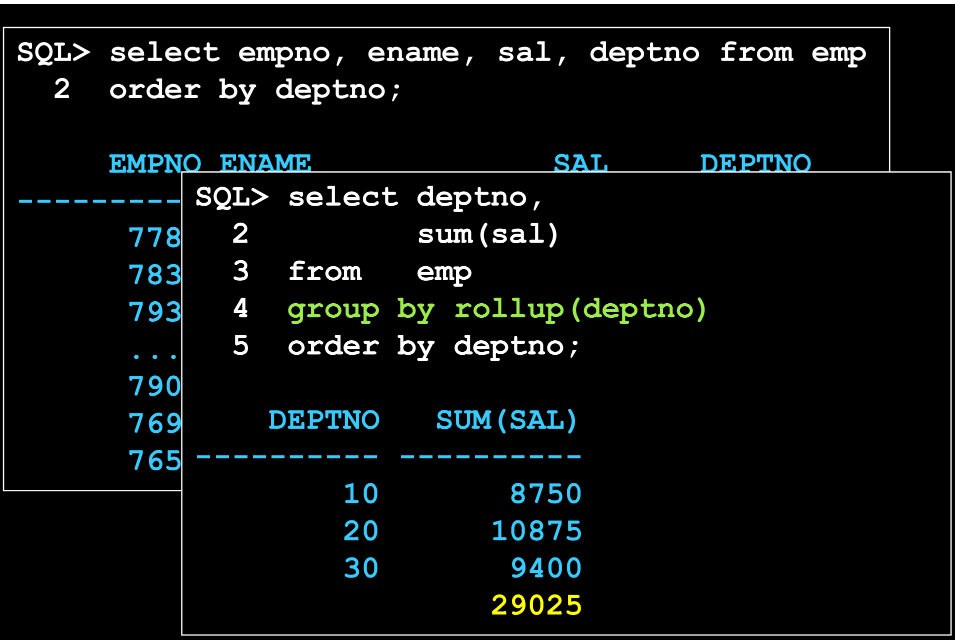
```
  L> select s    sal) from emp;

  SUM(SAL)
----------
     29025
```

170

from 3 to 2

rollup

171



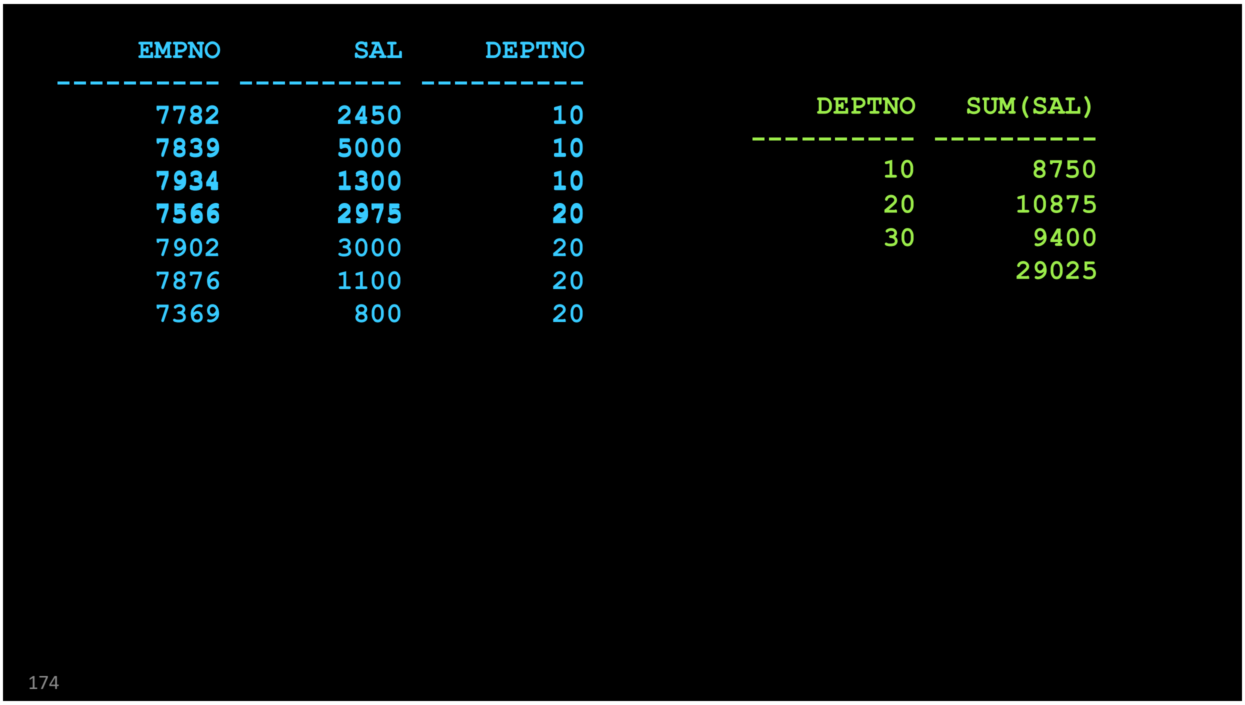
```
SQL> select empno, ename, sal, deptno from emp
  2  order by deptno;

     EMPNO ENAME                     SAL      DEPTNO
----------
       778
       783
       793
       ...
       790
       769
       765
```

```
SQL> select deptno,
  2         sum(sal)
  3  from   emp
  4  group by rollup(deptno)
  5  order by deptno;

    DEPTNO   SUM(SAL)
---------- ----------
        10       8750
        20      10875
        30       9400
                29025
```

172

still messy…

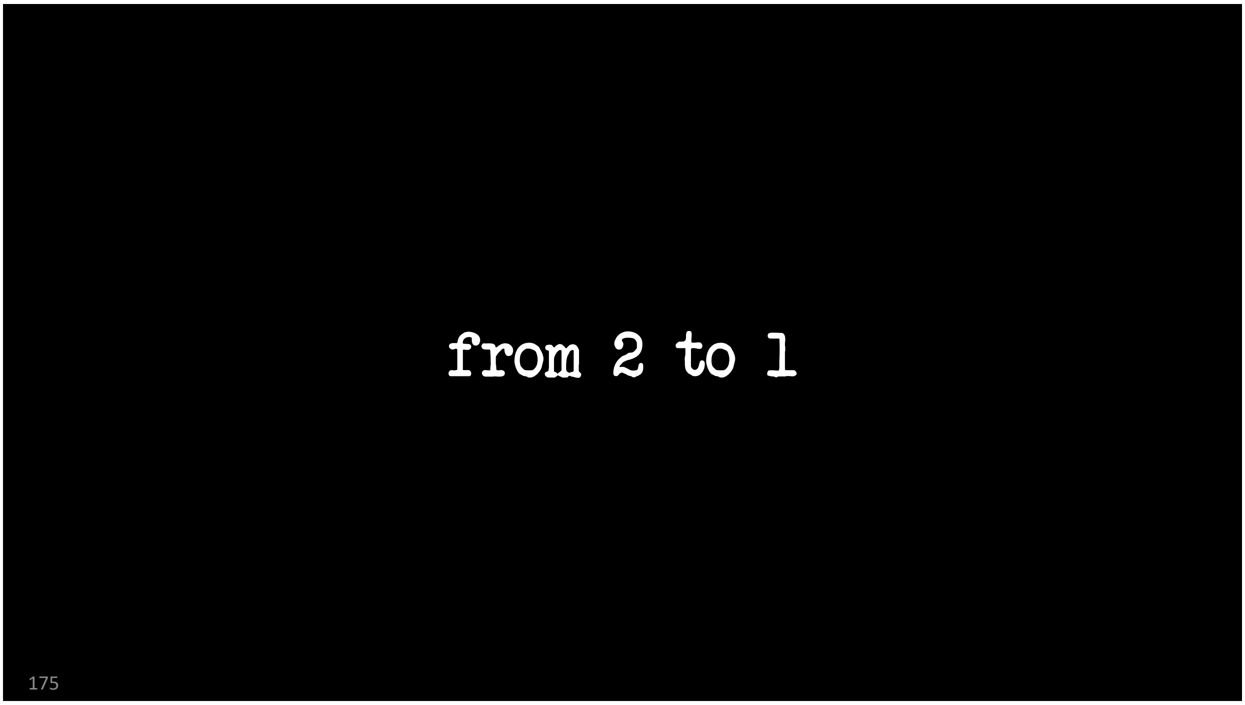
173



```
     EMPNO        SAL     DEPTNO
---------- ---------- ----------
      7782       2450         10
      7839       5000         10
      7934       1300         10
      7566       2975         20
      7902       3000         20
      7876       1100         20
      7369        800         20
```

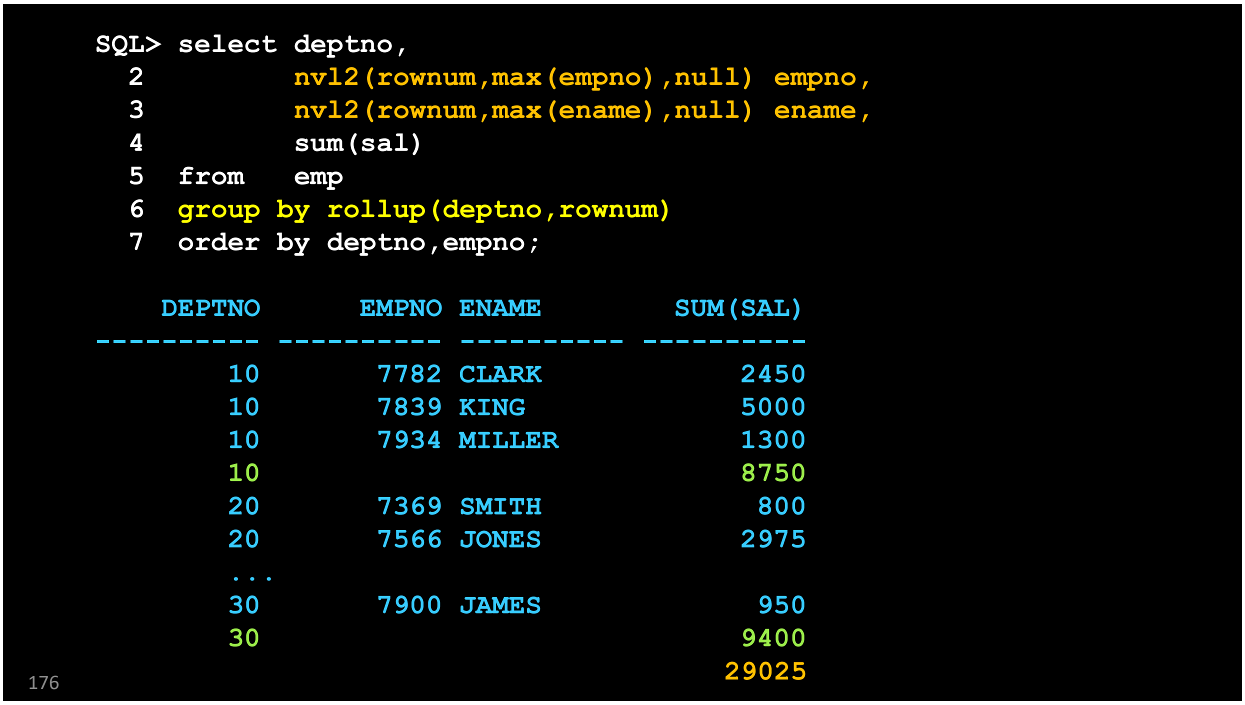
```
    DEPTNO   SUM(SAL)
---------- ----------
        10       8750
        20      10875
        30       9400
                29025
```

174



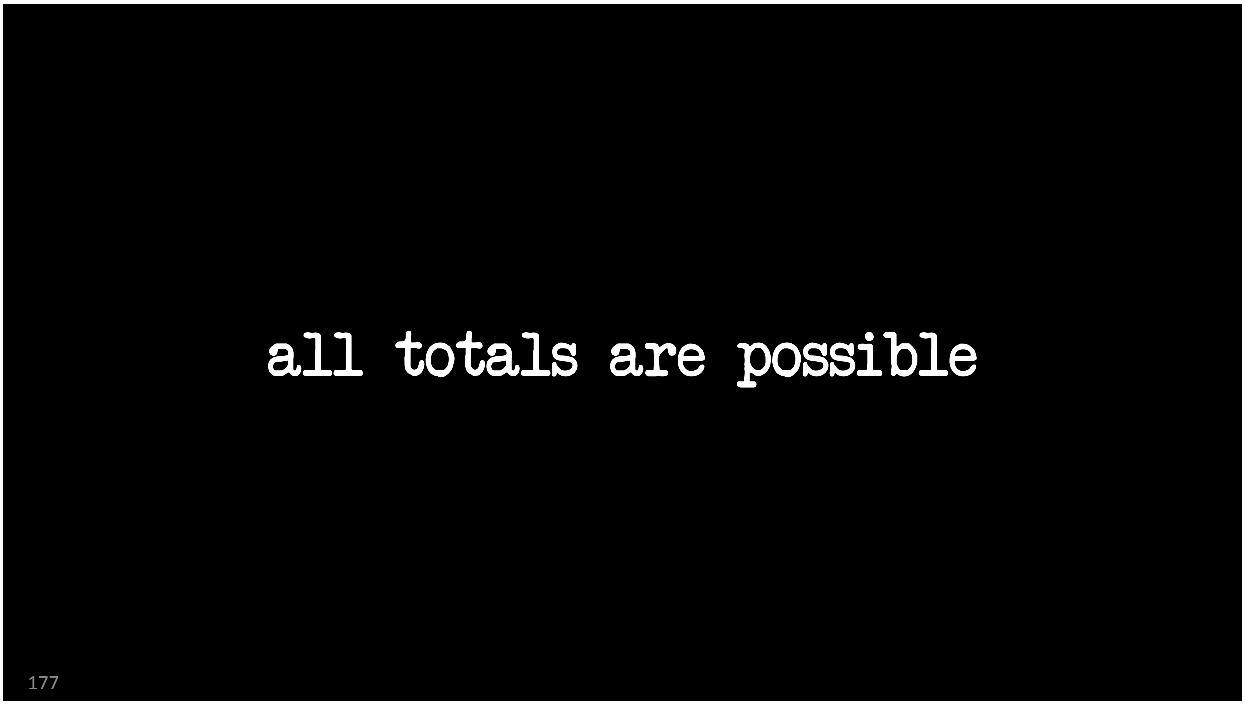
from 2 to 1

175



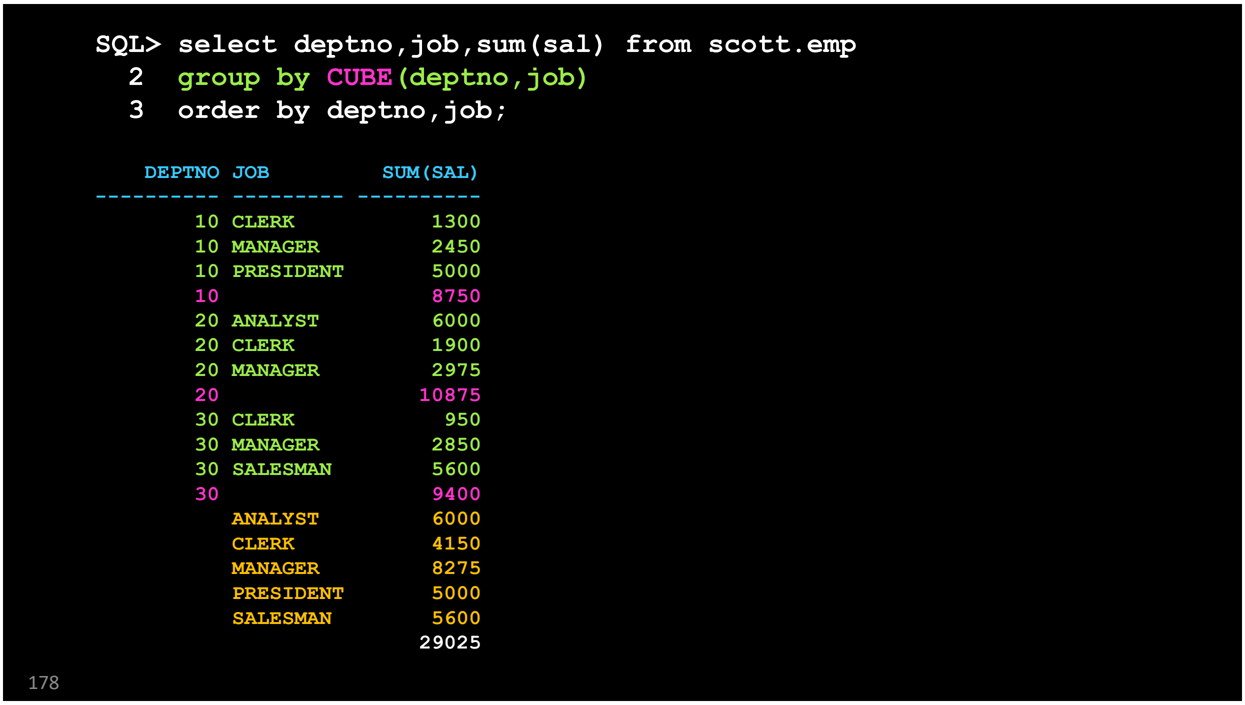
```
SQL> select deptno,
  2          nvl2(rownum,max(empno),null) empno,
  3          nvl2(rownum,max(ename),null) ename,
  4          sum(sal)
  5  from    emp
  6  group by rollup(deptno,rownum)
  7  order by deptno,empno;

    DEPTNO      EMPNO ENAME        SUM(SAL)
---------- ---------- ---------- ----------
        10       7782 CLARK            2450
        10       7839 KING             5000
        10       7934 MILLER           1300
        10                             8750
        20       7369 SMITH             800
        20       7566 JONES            2975
        ...
        30       7900 JAMES             950
        30                             9400
                                      29025
```

176



all totals are possible

177

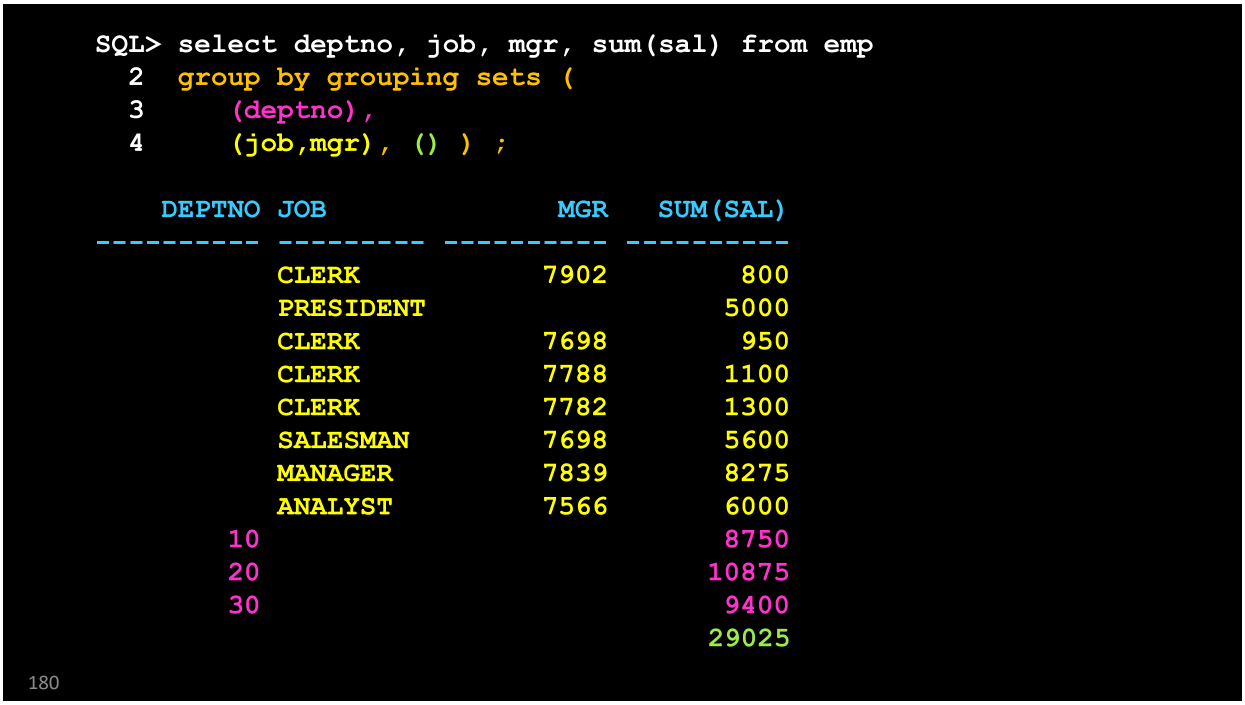


```
SQL> select deptno,job,sum(sal) from scott.emp
  2  group by CUBE(deptno,job)
  3  order by deptno,job;

    DEPTNO JOB         SUM(SAL)
---------- --------- ----------
        10 CLERK           1300
        10 MANAGER         2450
        10 PRESIDENT       5000
        10                 8750
        20 ANALYST         6000
        20 CLERK           1900
        20 MANAGER         2975
        20                10875
        30 CLERK            950
        30 MANAGER         2850
        30 SALESMAN        5600
        30                 9400
           ANALYST         6000
           CLERK           4150
           MANAGER         8275
           PRESIDENT       5000
           SALESMAN        5600
                          29025
```

178

totally customisable

179

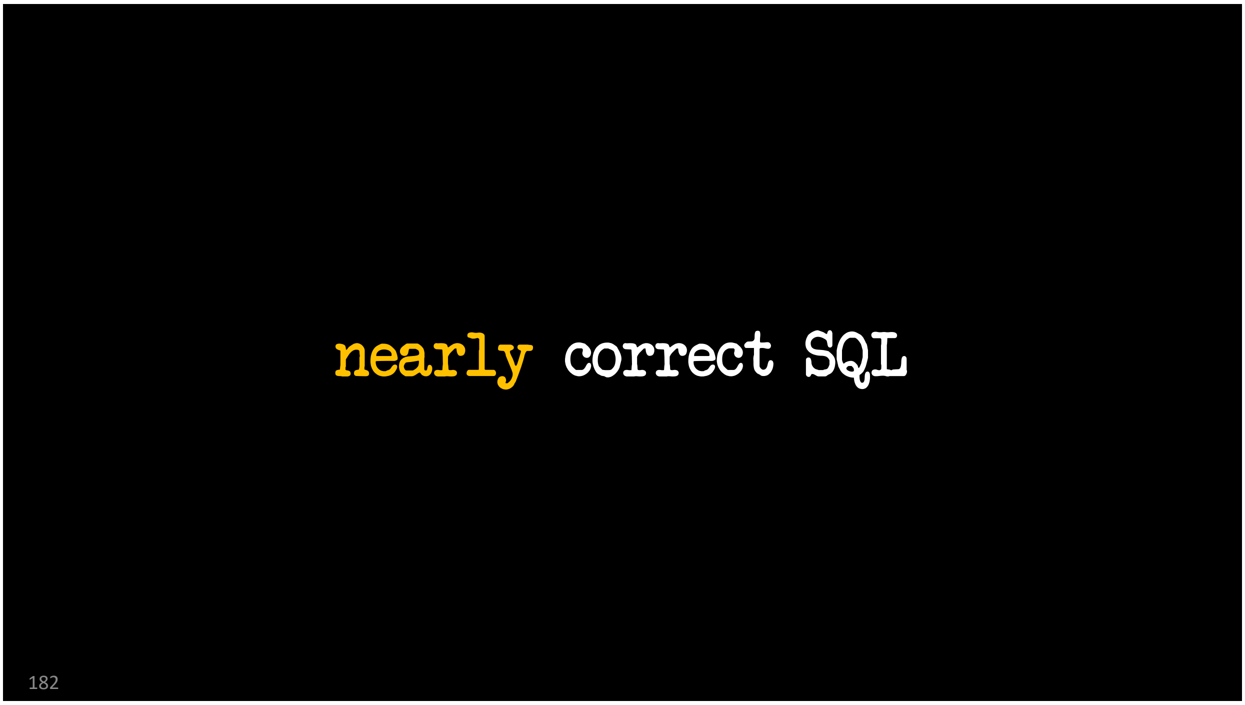


```
SQL> select deptno, job, mgr, sum(sal) from emp
  2  group by grouping sets (
  3      (deptno),
  4      (job,mgr), () ) ;

    DEPTNO JOB              MGR   SUM(SAL)
---------- --------- ---------- ----------
           CLERK           7902        800
           PRESIDENT                  5000
           CLERK           7698        950
           CLERK           7788       1100
           CLERK           7782       1300
           SALESMAN        7698       5600
           MANAGER         7839       8275
           ANALYST         7566       6000
        10                            8750
        20                           10875
        30                            9400
                                     29025
```

180

9

181



nearly correct SQL

182

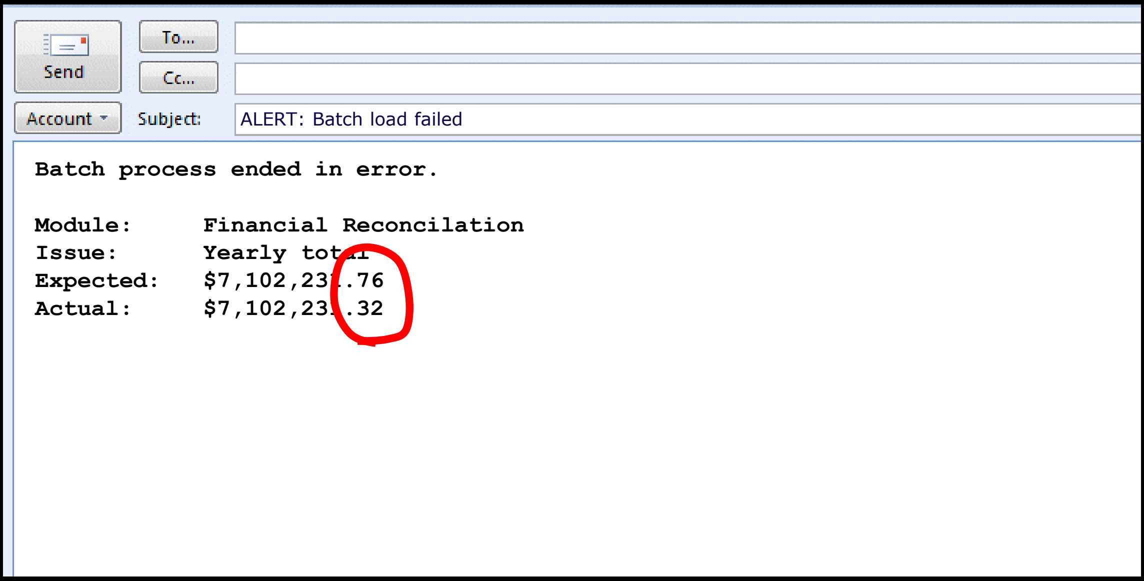
?

background

I hate being woken up …

185

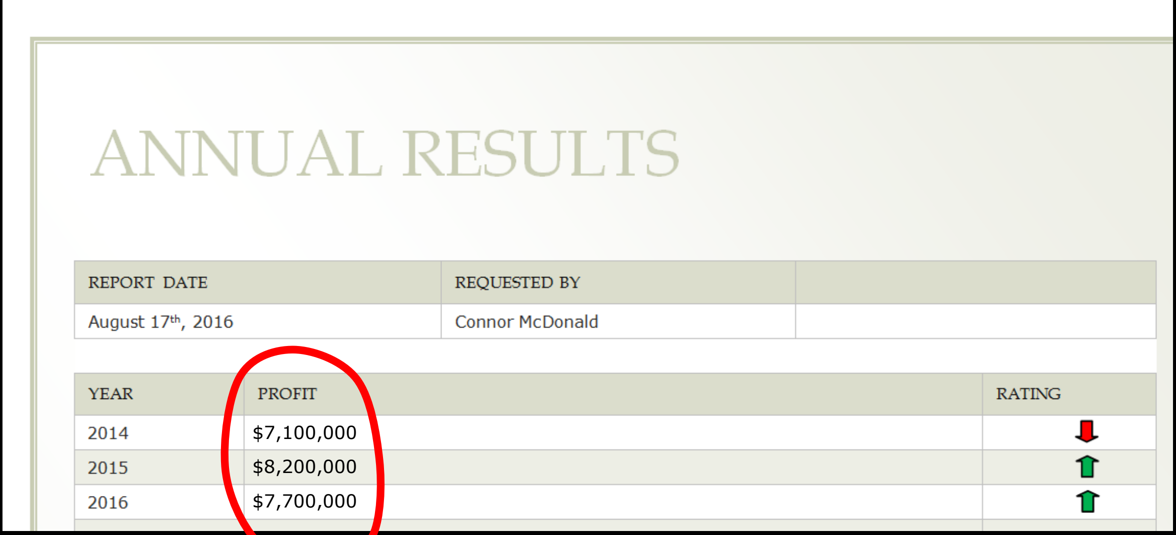
186



Subject: ALERT: Batch load failed

```
Batch process ended in error.

Module:     Financial Reconcilation
Issue:      Yearly total
Expected:   $7,102,231.76
Actual:     $7,102,231.32
```

187

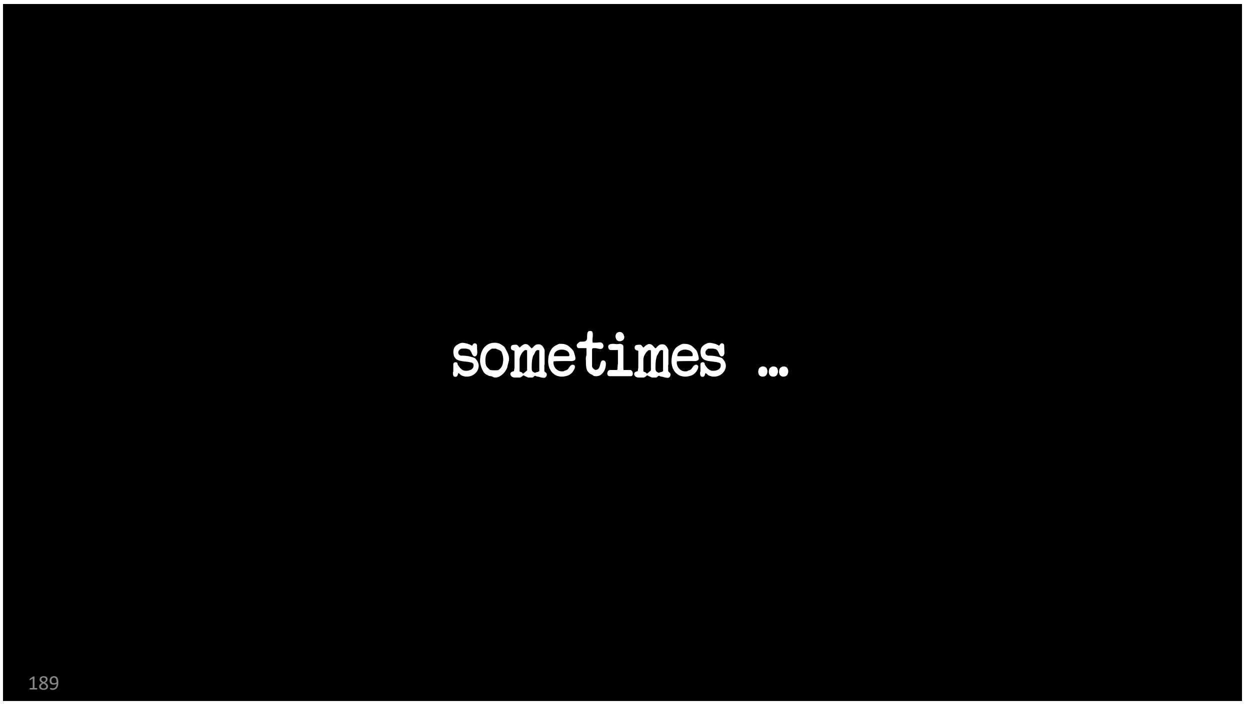


ANNUAL RESULTS

REPORT DATE	REQUESTED BY	
August 17th, 2016	Connor McDonald	

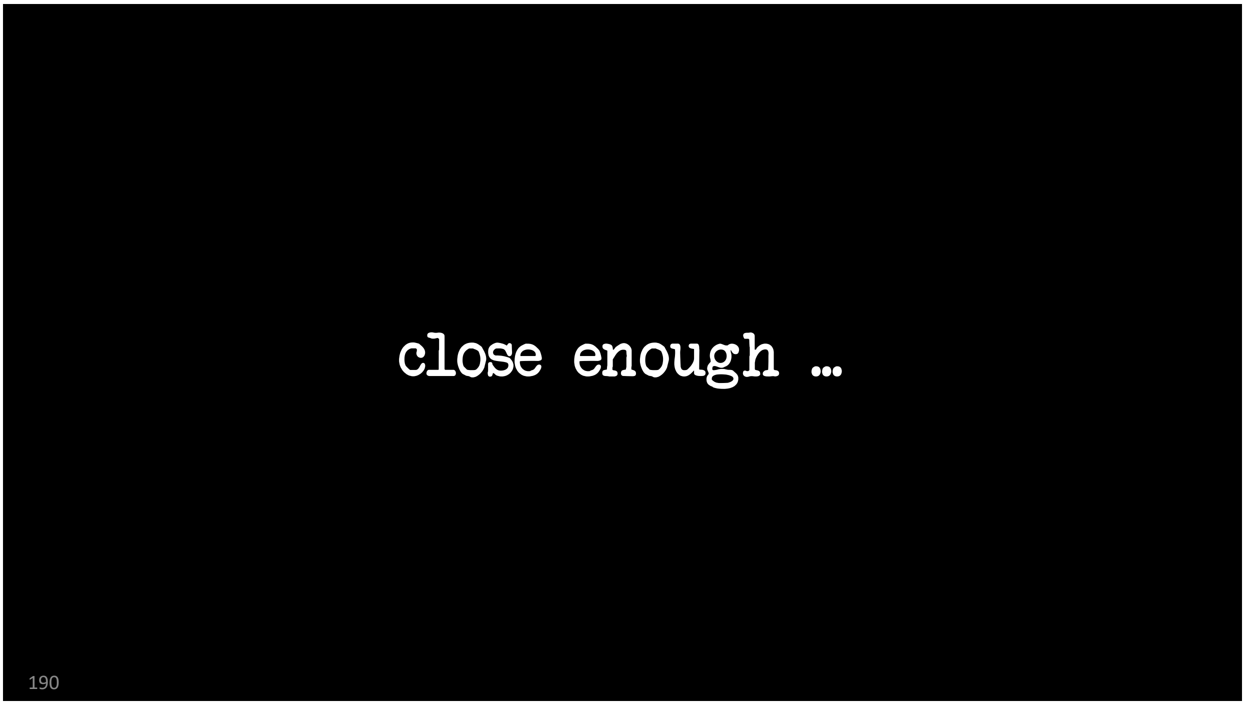
YEAR	PROFIT	RATING
2014	$7,100,000	⬇
2015	$8,200,000	⬆
2016	$7,700,000	⬆

188



sometimes …

189

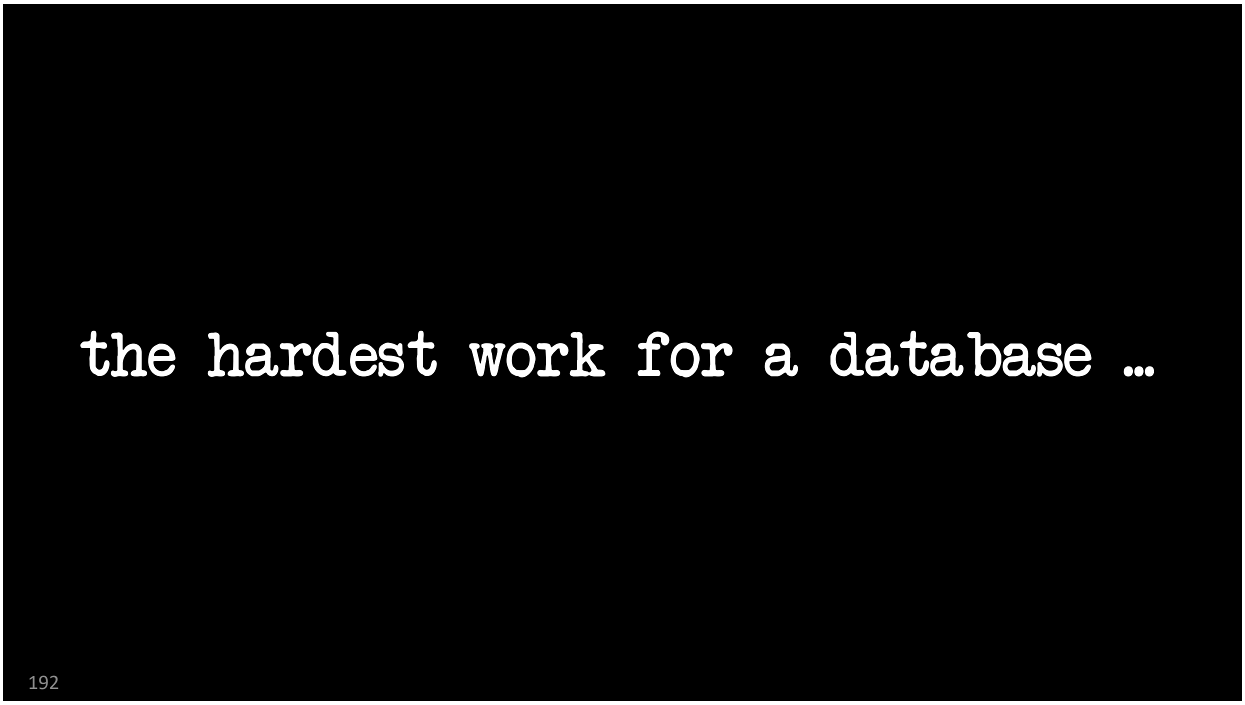


close enough …

190

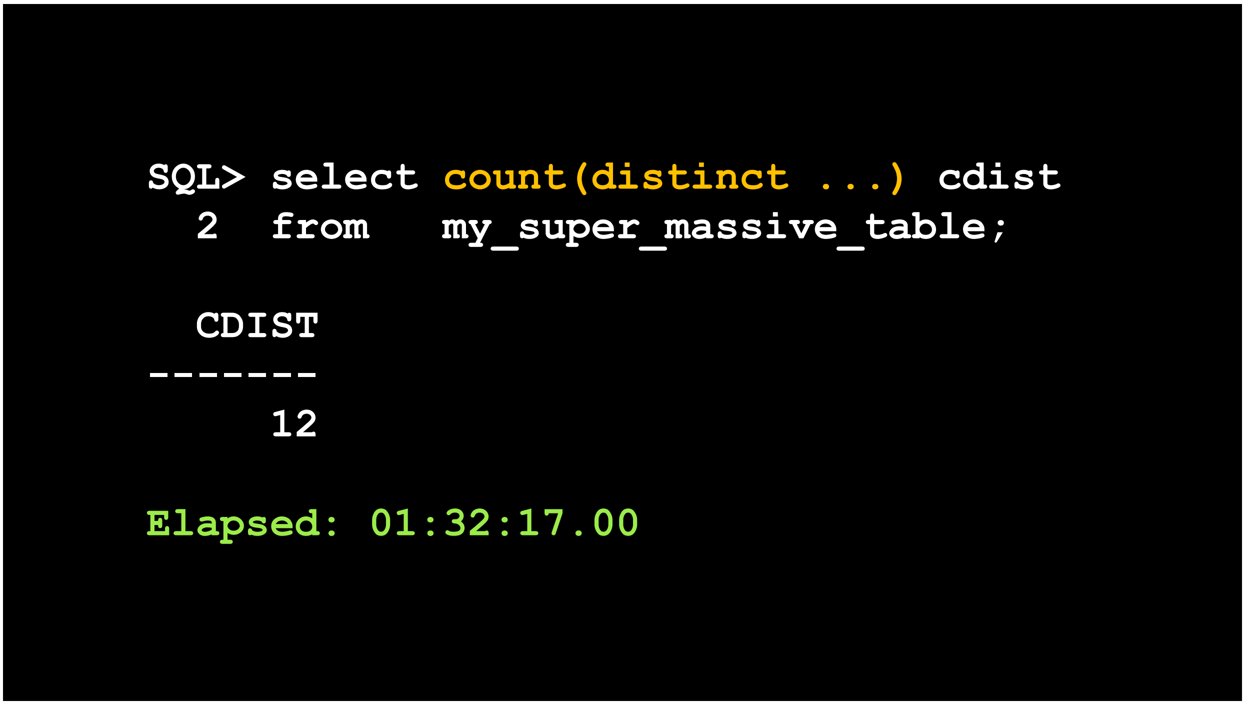
… is good enough

191



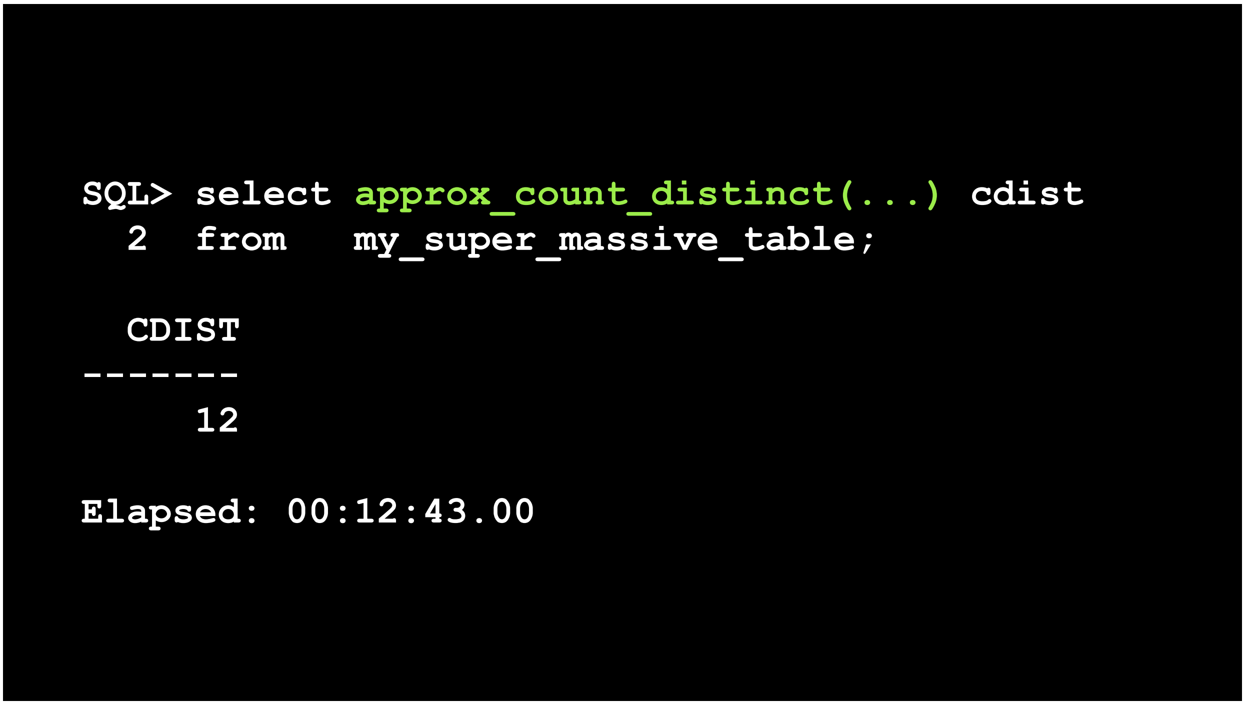
the hardest work for a database …

192



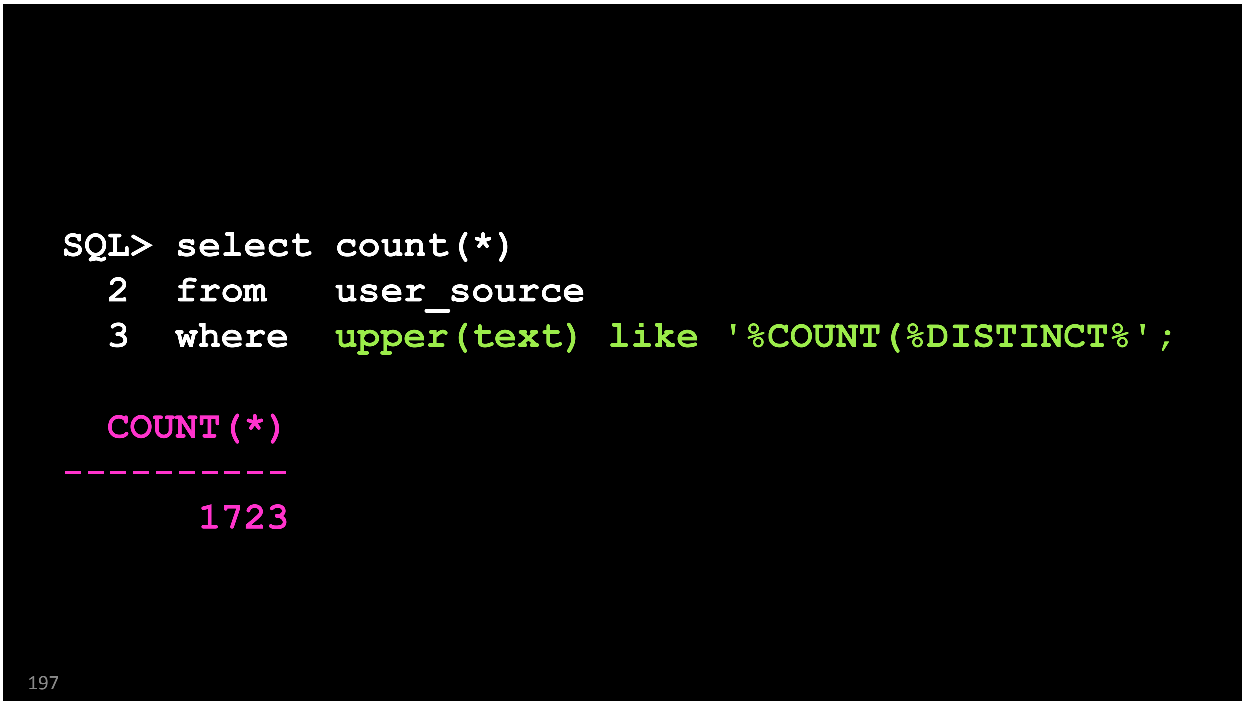
```
SQL> select count(distinct ...) cdist
  2  from   my_super_massive_table;

  CDIST
-------
     12

Elapsed: 01:32:17.00
```

12.1



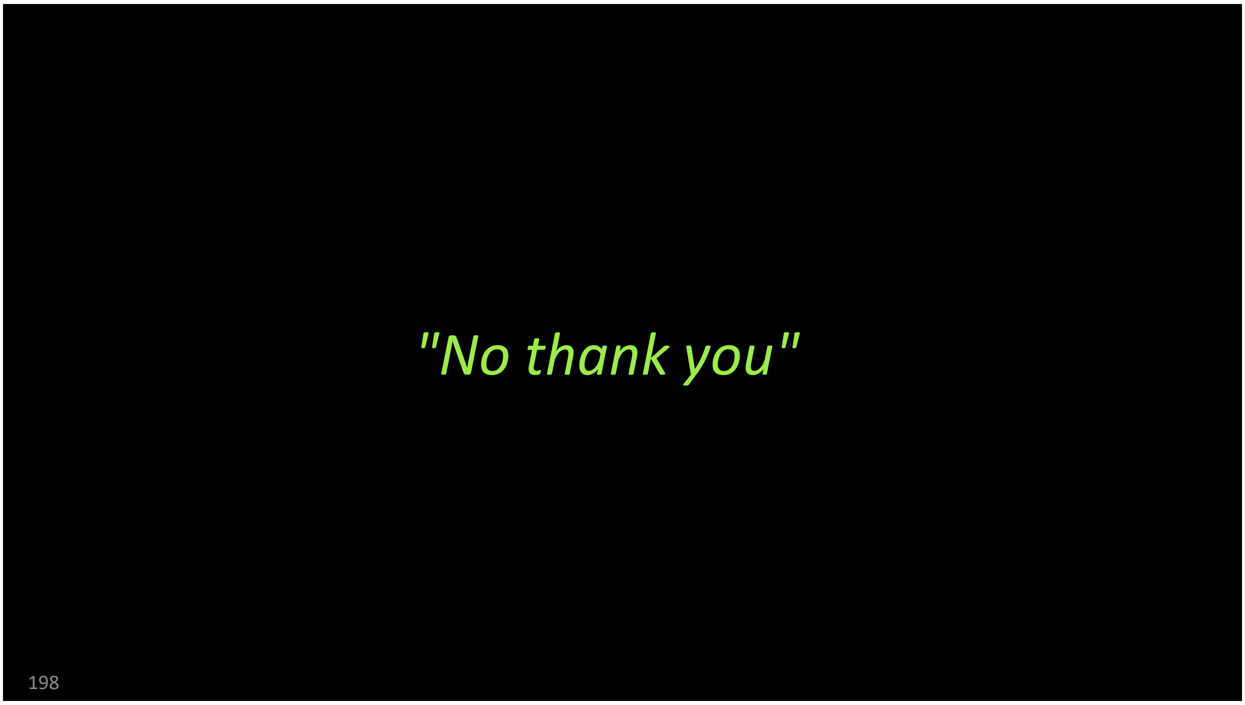
```
SQL> select approx_count_distinct(...) cdist
  2  from   my_super_massive_table;

  CDIST
-------
     12

Elapsed: 00:12:43.00
```

existing code



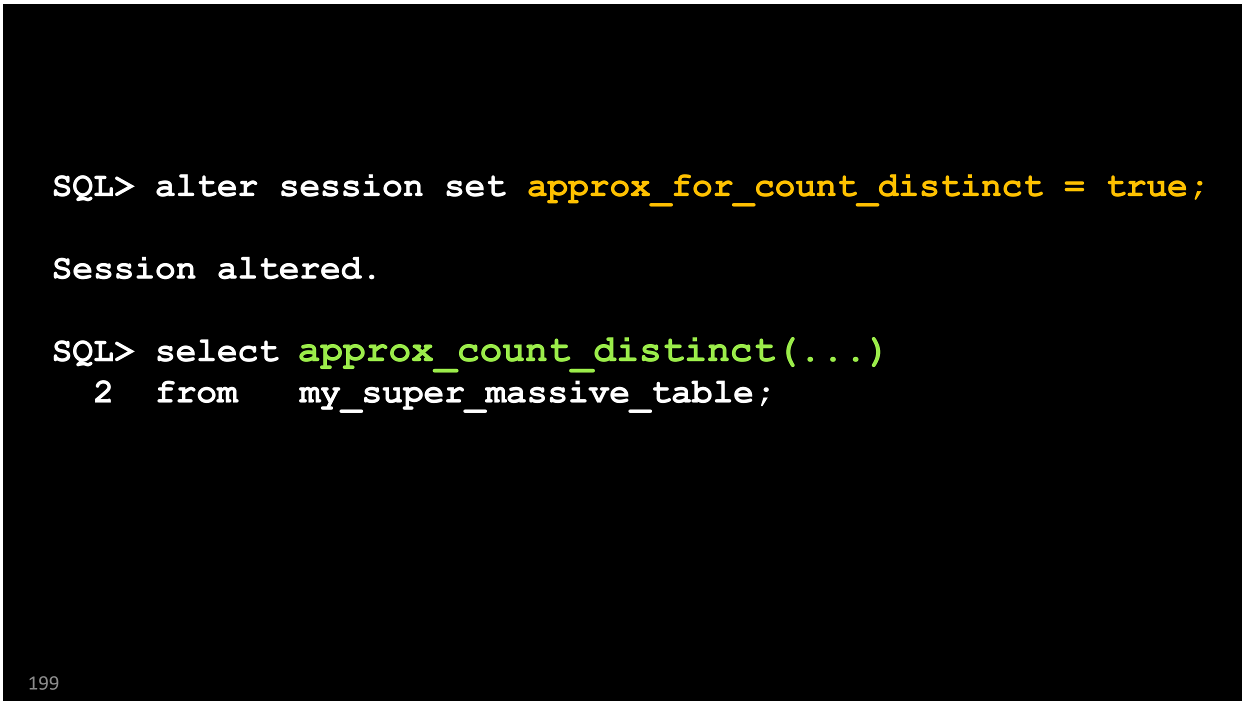
```
SQL> select count(*)
  2  from   user_source
  3  where  upper(text) like '%COUNT(%DISTINCT%';

  COUNT(*)
----------
      1723
```

197

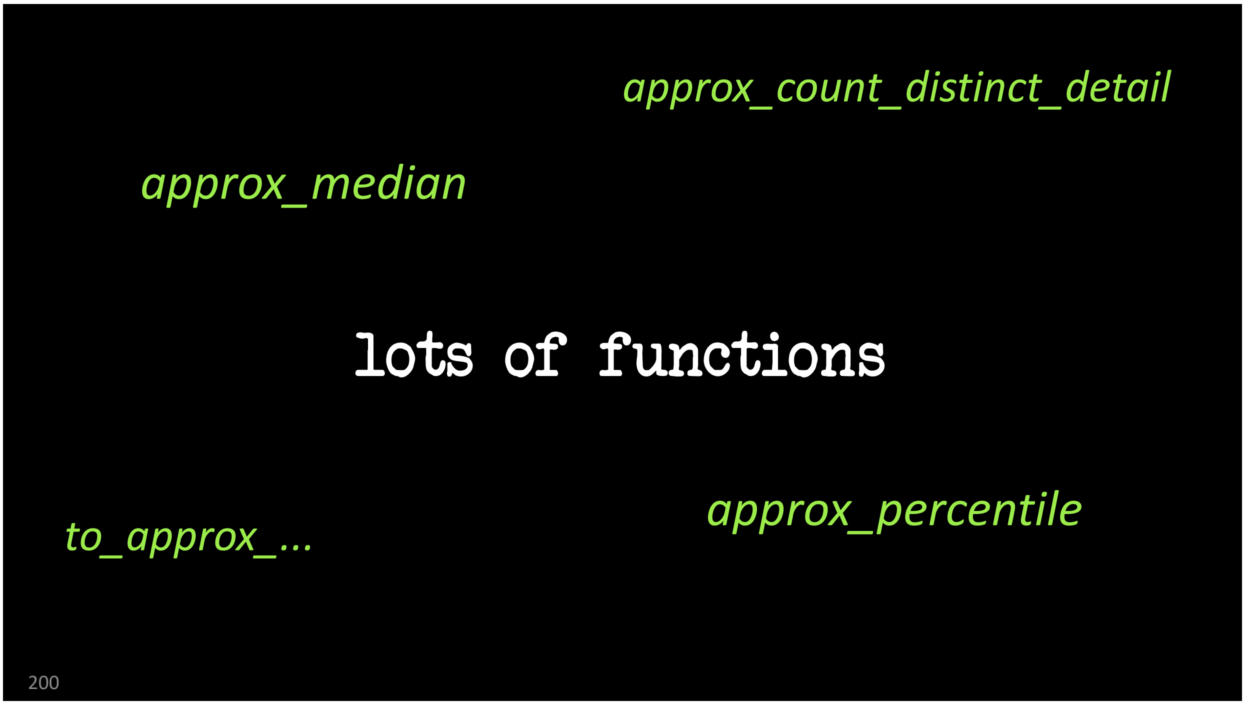


"No thank you"

198



```
SQL> alter session set approx_for_count_distinct = true;

Session altered.

SQL> select approx_count_distinct(...)
  2  from   my_super_massive_table;
```



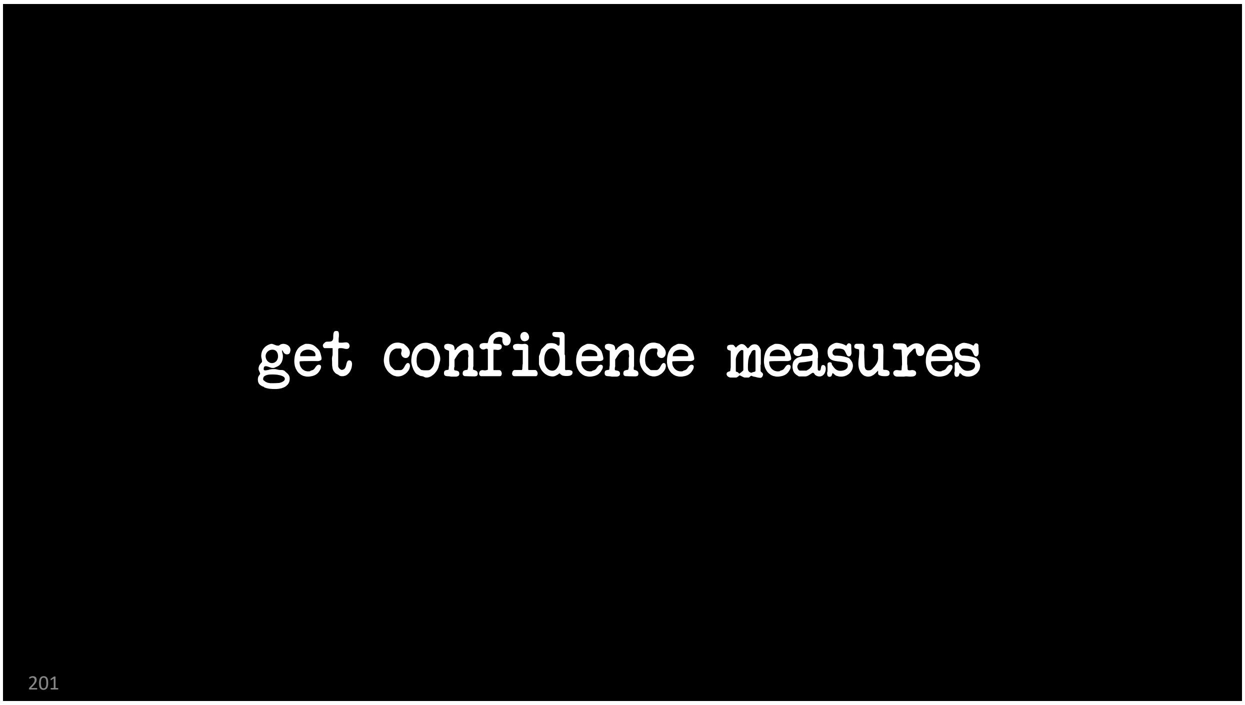
approx_count_distinct_detail

approx_median

lots of functions

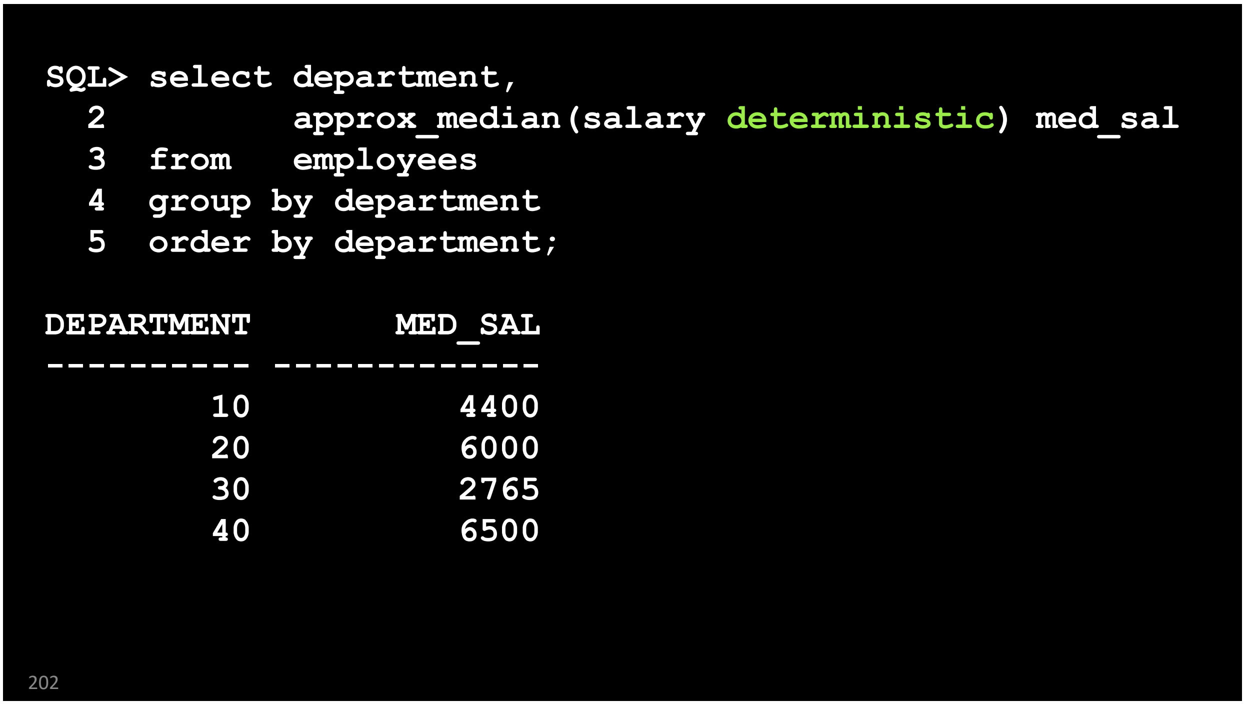
to_approx_...

approx_percentile



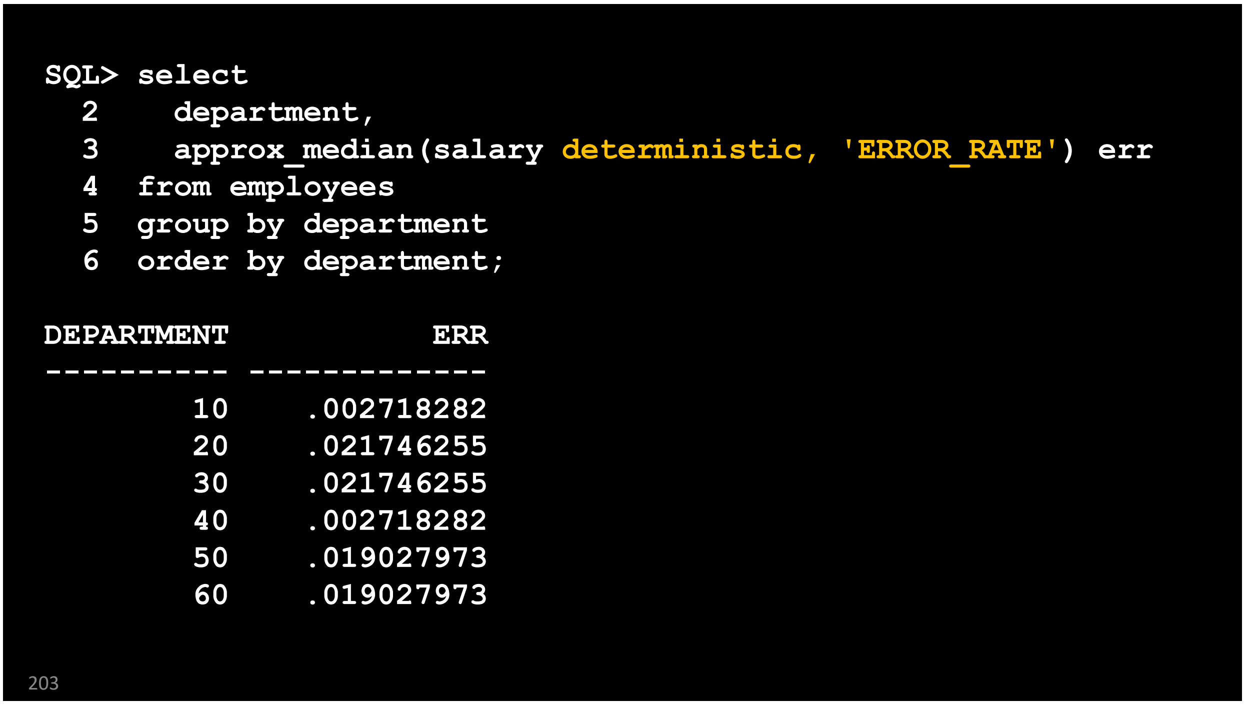
get confidence measures

201



```
SQL> select department,
  2         approx_median(salary deterministic) med_sal
  3  from   employees
  4  group by department
  5  order by department;

DEPARTMENT       MED_SAL
---------- -------------
        10          4400
        20          6000
        30          2765
        40          6500
```

202



```
SQL> select
  2    department,
  3    approx_median(salary deterministic, 'ERROR_RATE') err
  4  from employees
  5  group by department
  6  order by department;

DEPARTMENT          ERR
---------- ------------
        10   .002718282
        20   .021746255
        30   .021746255
        40   .002718282
        50   .019027973
        60   .019027973
```

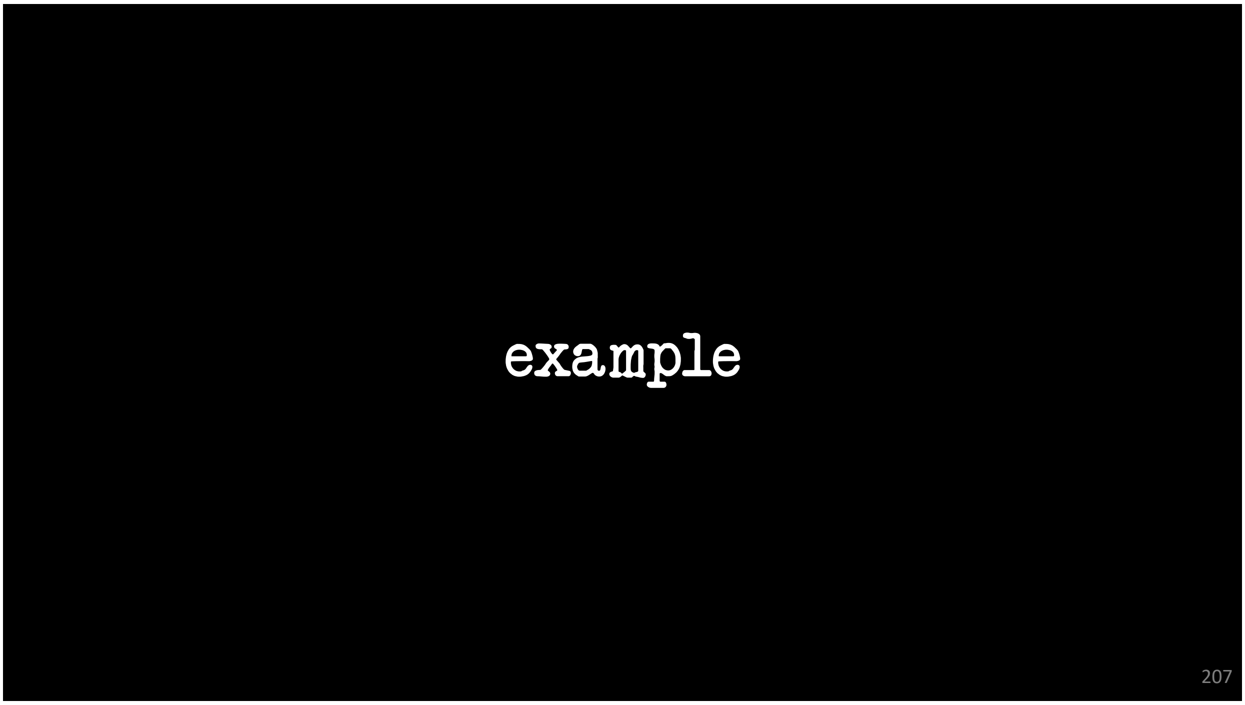
last

205

”talking” to your database

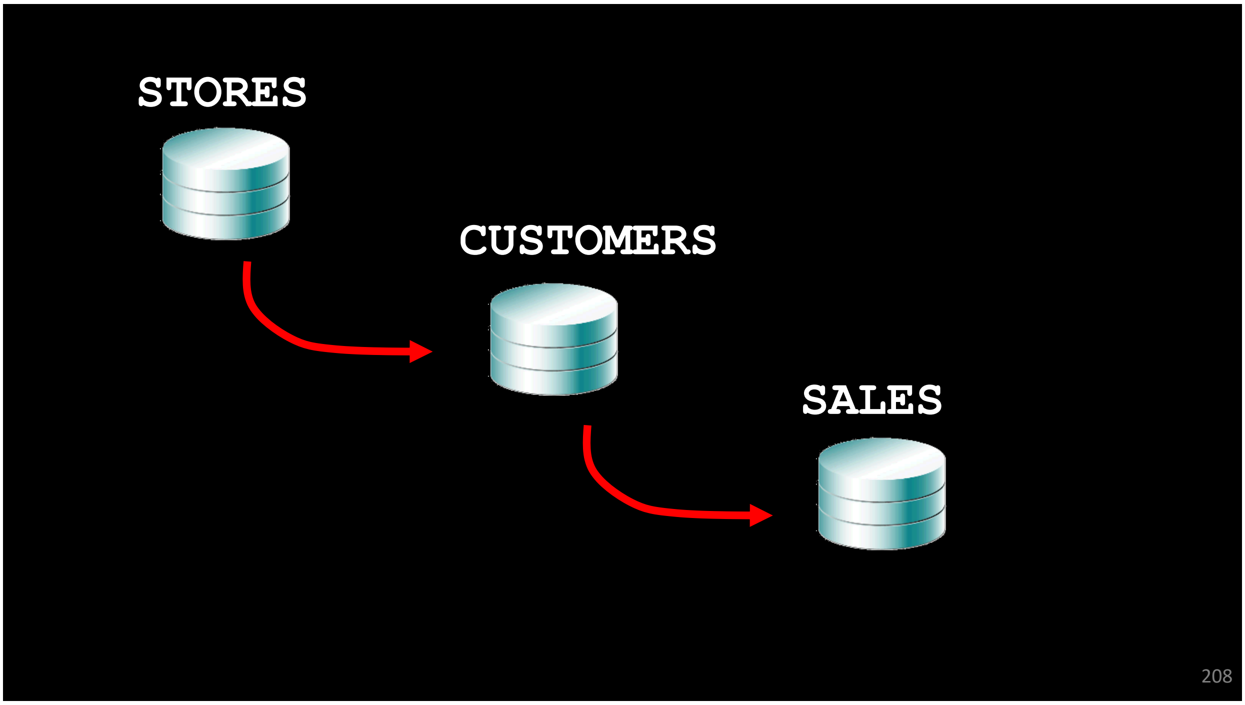
... makes it faster

206



example

207

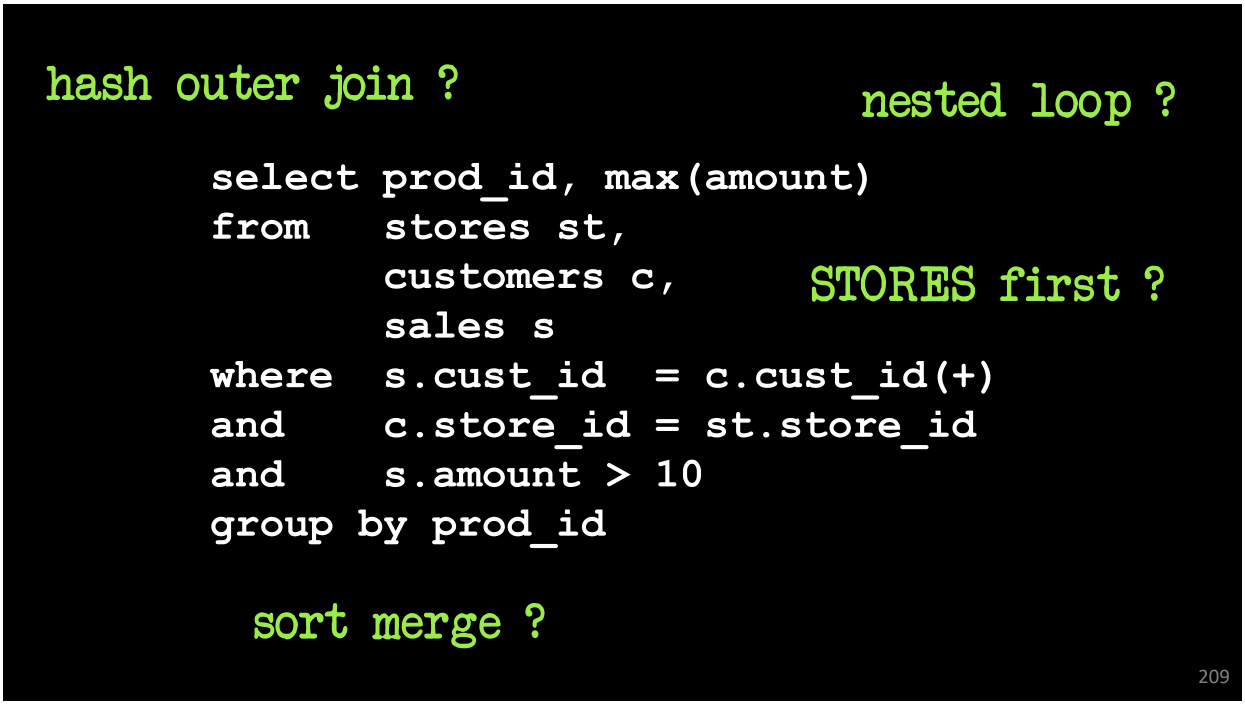


STORES

CUSTOMERS

SALES

208



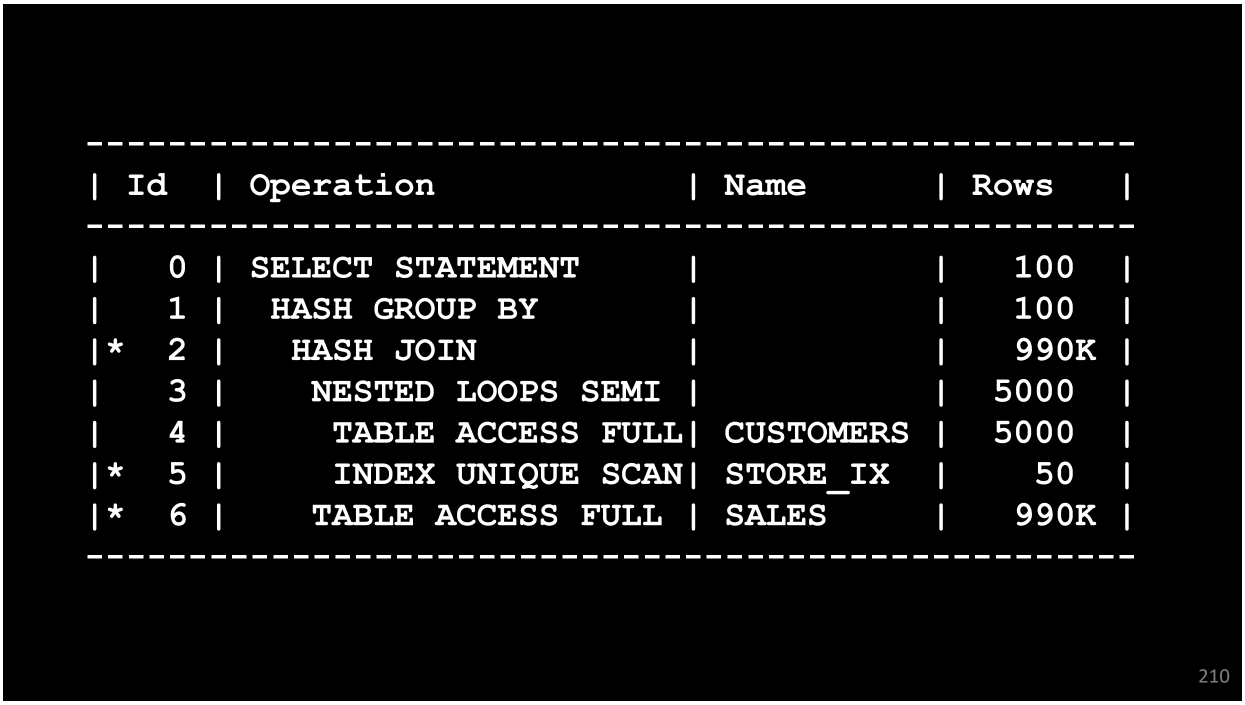
hash outer join ?

nested loop ?

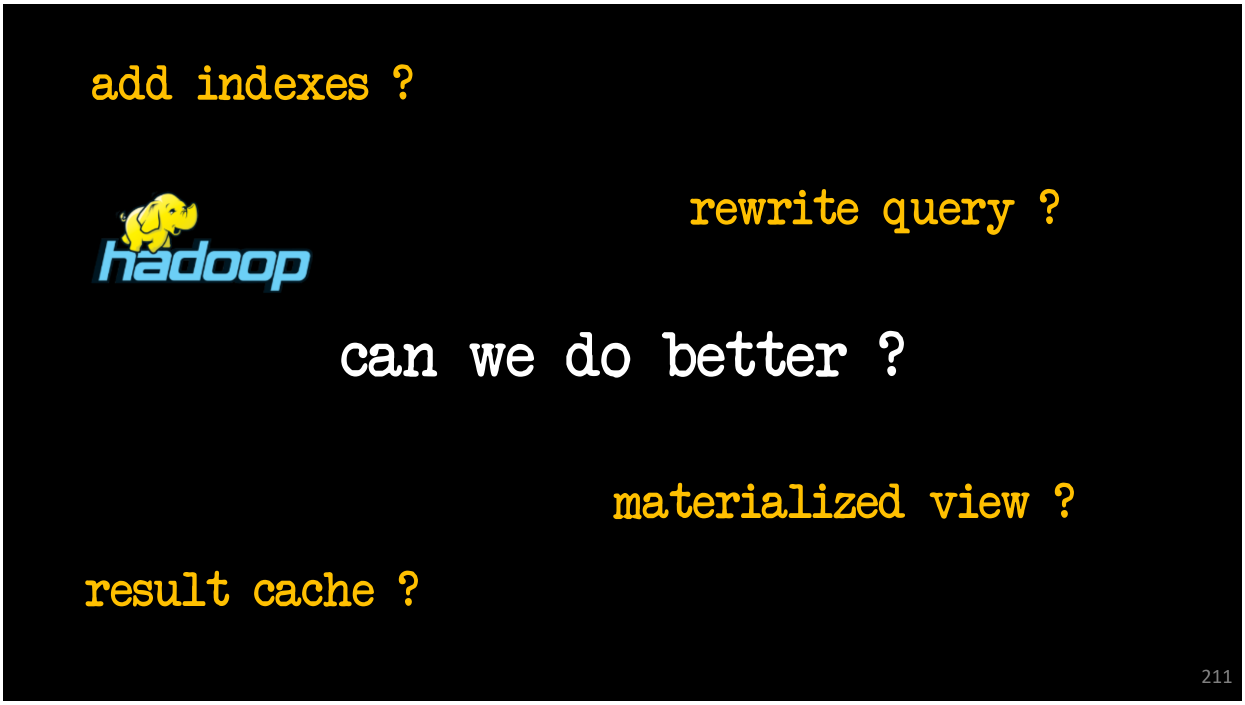
```
select prod_id, max(amount)
from   stores st,
       customers c,
       sales s
where  s.cust_id  = c.cust_id(+)
and    c.store_id = st.store_id
and    s.amount > 10
group by prod_id
```

STORES first ?

sort merge ?



```
---------------------------------------------------
| Id  | Operation             | Name       | Rows   |
---------------------------------------------------
|   0 | SELECT STATEMENT      |            |   100  |
|   1 |  HASH GROUP BY        |            |   100  |
|*  2 |   HASH JOIN           |            |   990K |
|   3 |    NESTED LOOPS SEMI  |            |  5000  |
|   4 |     TABLE ACCESS FULL| CUSTOMERS |  5000  |
|*  5 |     INDEX UNIQUE SCAN| STORE_IX   |    50  |
|*  6 |    TABLE ACCESS FULL | SALES      |   990K |
---------------------------------------------------
```



add indexes ?

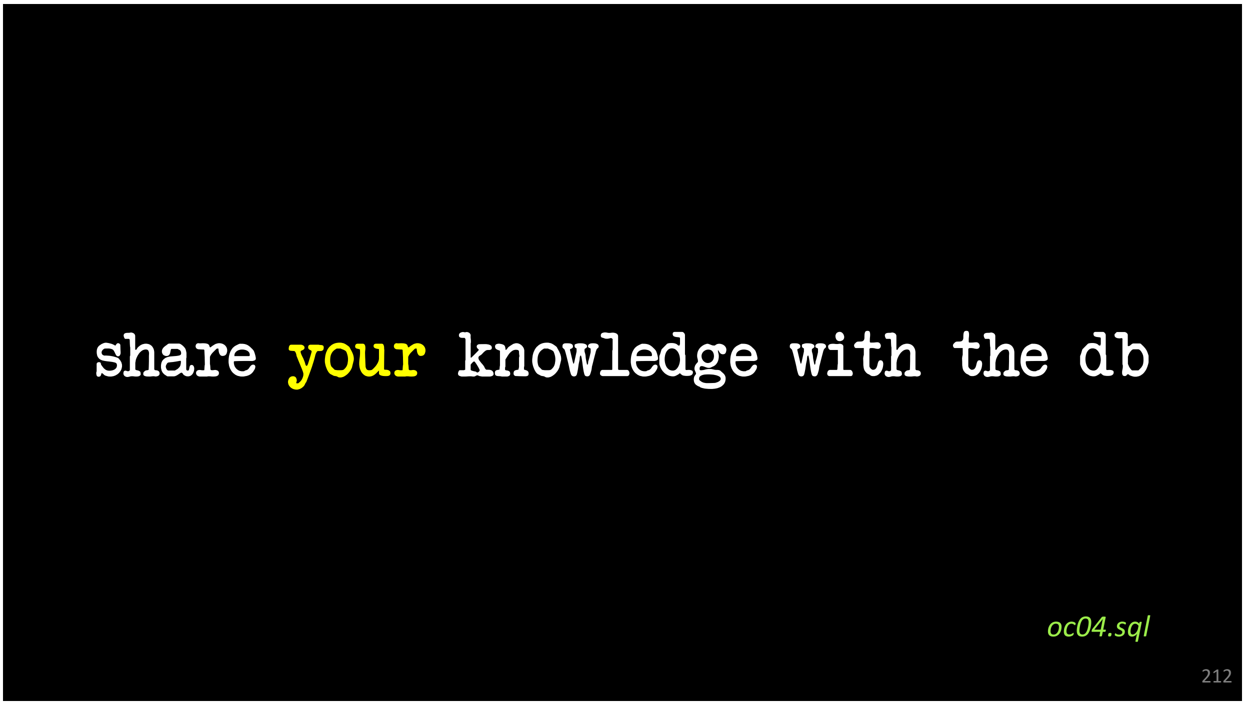
hadoop

rewrite query ?

can we do better ?

materialized view ?

result cache ?



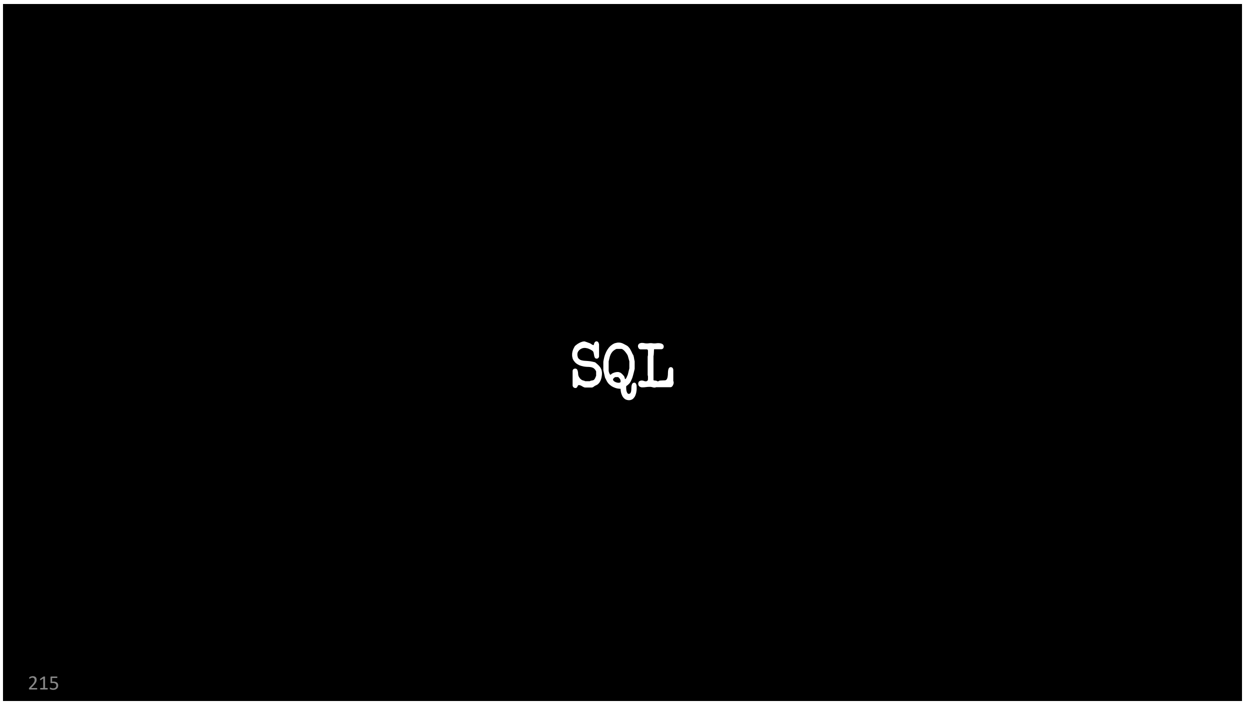
share your knowledge with the db

oc04.sql

wrap up

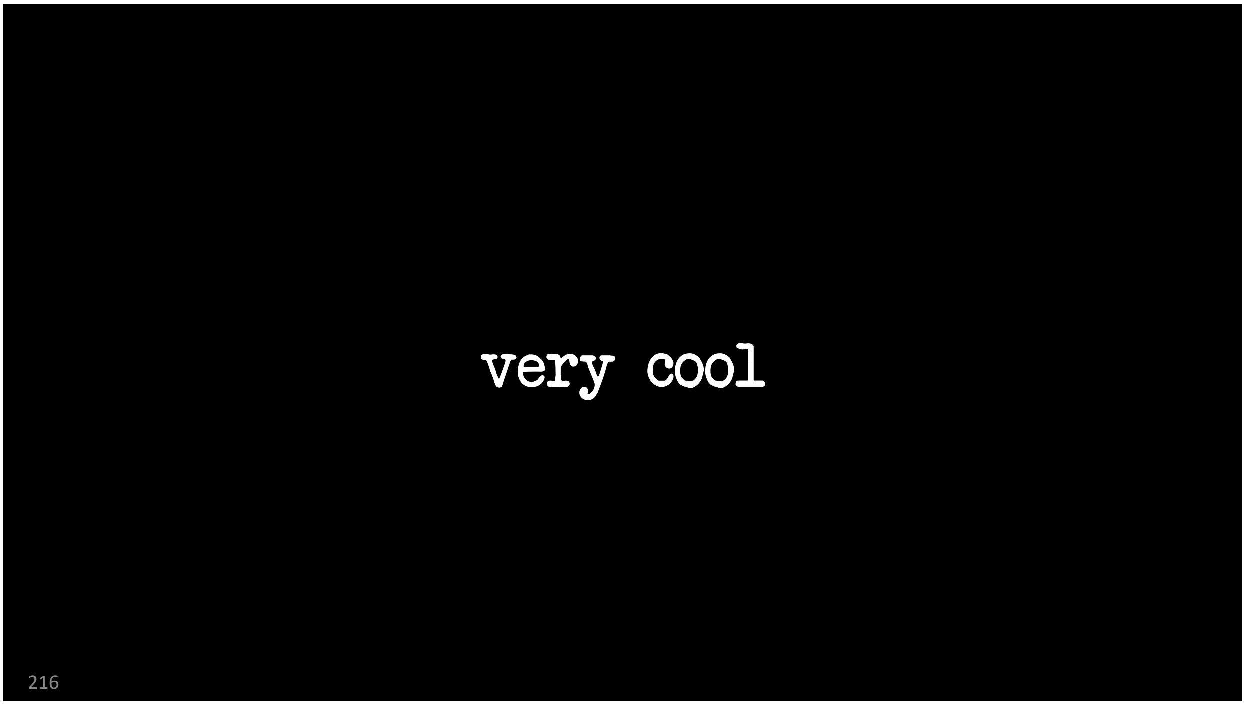
213

214



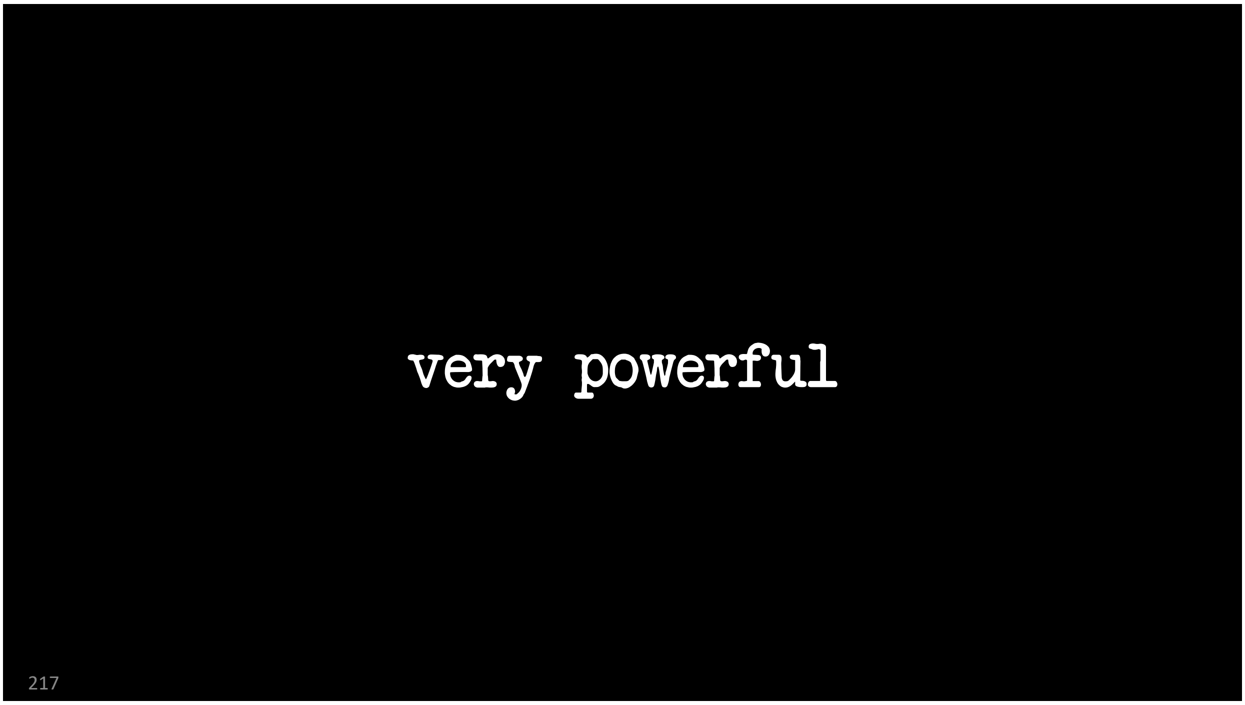
SQL

215



very cool

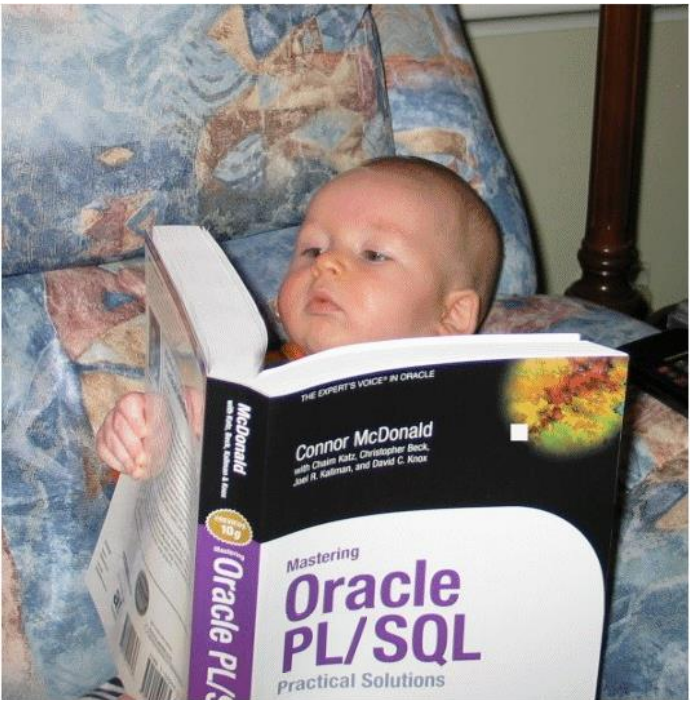
216



very powerful

less code

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
PL/SQL
Practical Solutions
McDonald
Oracle PL/S

Enjoy the conference !!!

Follow and Subscribe !

blog	connor-mcdonald.com
youtube	tinyurl.com/connor-tube
twitter	@connor_mc_d