Notes, abstract

The presentation shows the use of “thick DB”, with lots of PL/SQL.

This can be both good and bad.

From an AWR of a badly performing system we show the symptoms, and how the “root cause” was identified. Via the diagnosis and some SQL*Plus-screenshots we show the possible fixes and their limitations.

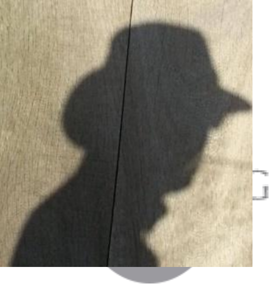
After this presentation, attendee will be able to make better-informed decisions on the use of PL/SQL, notably the use of functions.

Notes: more on 3 x solutions.

Notes: digging…

Notes: clipart uphill.. ?

Notes:



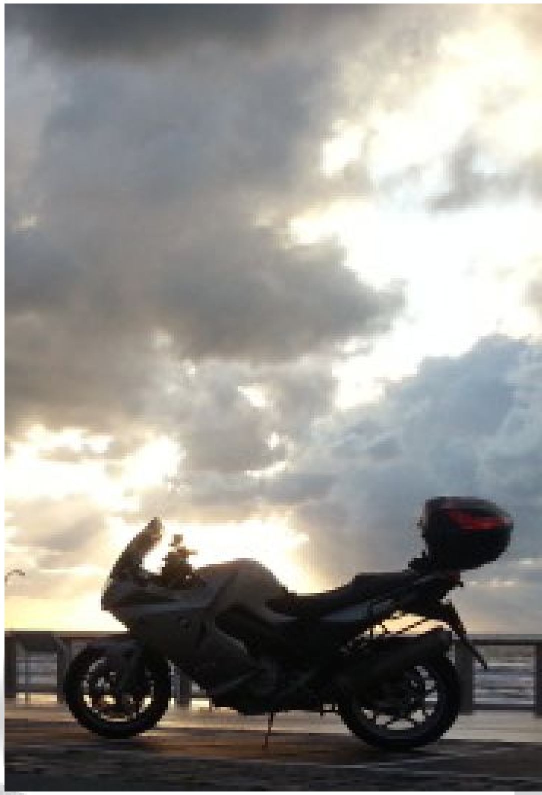
Piet de Visser
PDVBV

PGDAY'17 RUSSIA
КОНФЕРЕНЦИЯ ПО БАЗАМ ДАННЫХ

The Brilliant Concept

A DB-centered architecture, what could go wrong.

PDVBV – The Simple (oracle) DBA



PDVBV

Favorite Quotes: "The Limitation shows the master" (Goethe), "Simplicity is not a luxury, it is a necessity. Unfortunately, "Complex' solutions sell better. (EW Dijkstra).

Logo Cloud

- Shell
- Philips
- ING bank
- Nokia
- (dutch gov)
- Insinger, BN
- Etihad
- NHS
- BT
- Claritas, Niels
- Unilever
- Exxon
- GE

Don’t waste time on Self-Inflation… but Hey, this was such a cool Idea (from a marketing guy)…
Logos of my major customers over time. If you want your logo here: Hire me.

What does it look like..

PDVBV

Couldn't resist... after this changing room, not allowed to take pictures anymore..
For travel pictures from Asia: later...

Agenda (approx 45 min)

History (why this…)

Investigate (read AWR)

Application Architecture… (layers!)

PL/SQL-Functions (concept, code)

3 (three) Fixes (good, bad?)

More to it, time.. (Others…)

10 min Discussion (Do Challenge!)

9:41 AM
Monday, March 7
Search Events
All
My Schedule
Suggested
9:00 PM
9:00 Drinks & Mingle
9:30 Brian Goldman presents...
10:00 PM
10:00 Michael Pikos & Maurice...
11:00 PM
11:15 Michael Pikos & Maurice...
11:30 Michael Pikos & Maurice...
Tuesday, May 23

Agenda. Why, what, how, comparisons.
30+ slides, 1min/slide.

Why this topic ...   Dev (no-Dev-Ops)!

Slow screens, Slow Reports, even Slow MVs

KIWI… Moved to Exadata: _Only_ +/- 3x Faster…

Users, Operations:

Management : RCA…
    Root Cause Analysis
    # cat /etc/RCA

Start by observing …

PDVBV

System had moved to Exadata, but was "only" 3x faster, and in some cases not at all...
Note: I am part of a "IT Repsonse Team, we go listen and help... (5 MS/sharepoint ppl, 1 Oracle DBA...)

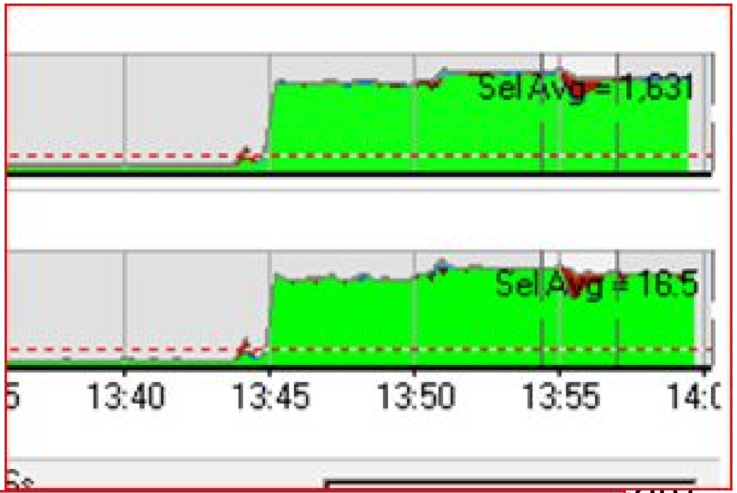
4SYNERGY

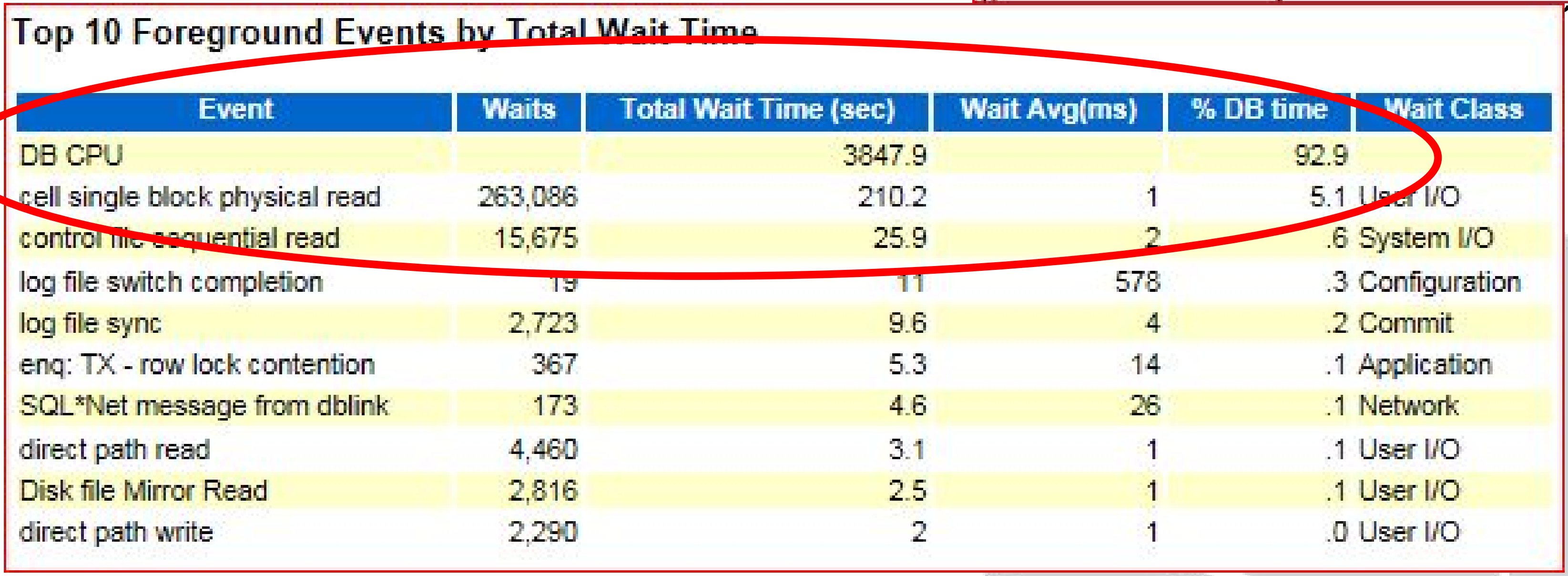
PGDAY'17 RUSSIA

Investigate…

Start by observing..

- **Use OEM (and Lab128 !)**
- **Isolate + run Test-cases**
- **Extract AWR reports**





Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Wait Avg(ms)	% DB time	Wait Class
DB CPU		3847.9		92.9	
cell single block physical read	263,086	210.2	1	5.1	User I/O
control file sequential read	15,675	25.9	2	.6	System I/O
log file switch completion	19	11	578	.3	Configuration
log file sync	2,723	9.6	4	.2	Commit
enq: TX - row lock contention	367	5.3	14	.1	Application
SQL*Net message from dblink	173	4.6	26	.1	Network
direct path read	4,460	3.1	1	.1	User I/O
Disk file Mirror Read	2,816	2.5	1	.1	User I/O
direct path write	2,290	2	1	.0	User I/O

System was only using CPU – green graph from lab128 is typical "test",
AWR was a 30min report – 92% of activity is CPU… "average" only 2 sessions active. During test … 16 sess.

Findings.....

AWR and Lab128 show:

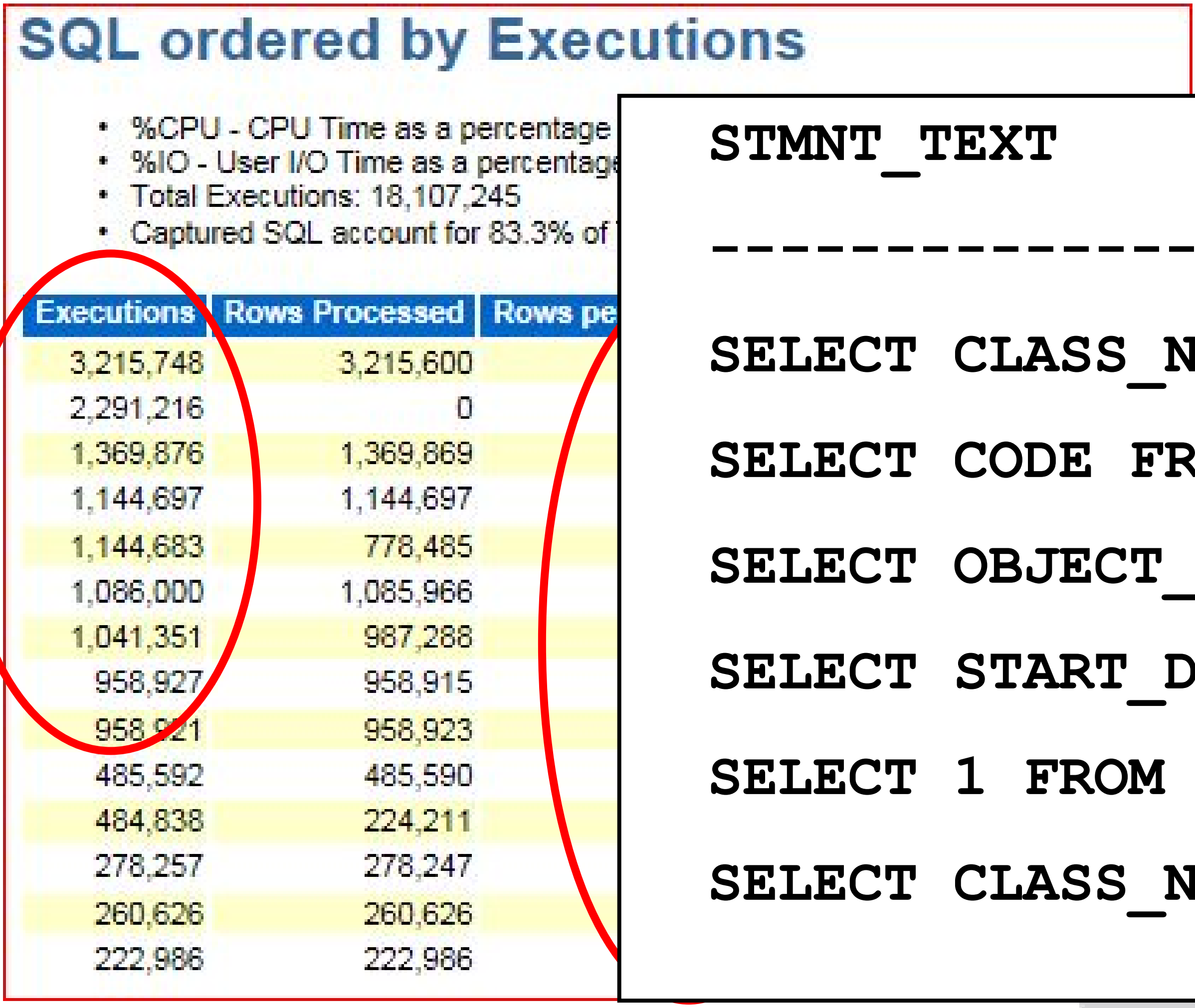
- CPU only (hence 3x gain from Exa)
- Very high nr of "Executes"
- (and Several SQLs at same-frequency)

PDVBV

System was only using CPU,... high nr of "executes (25.000, open AWR....)
(and relatively low nr of user-calls, no chattyness) => this indicates lots of PL/SQL activity (we have a start)

Findings..... (zoom in)

Most executes seem to query same "Objects"



SQL ordered by Executions

- %CPU - CPU Time as a percentage
- %IO - User I/O Time as a percentage
- Total Executions: 18,107,245
- Captured SQL account for 83.3% of

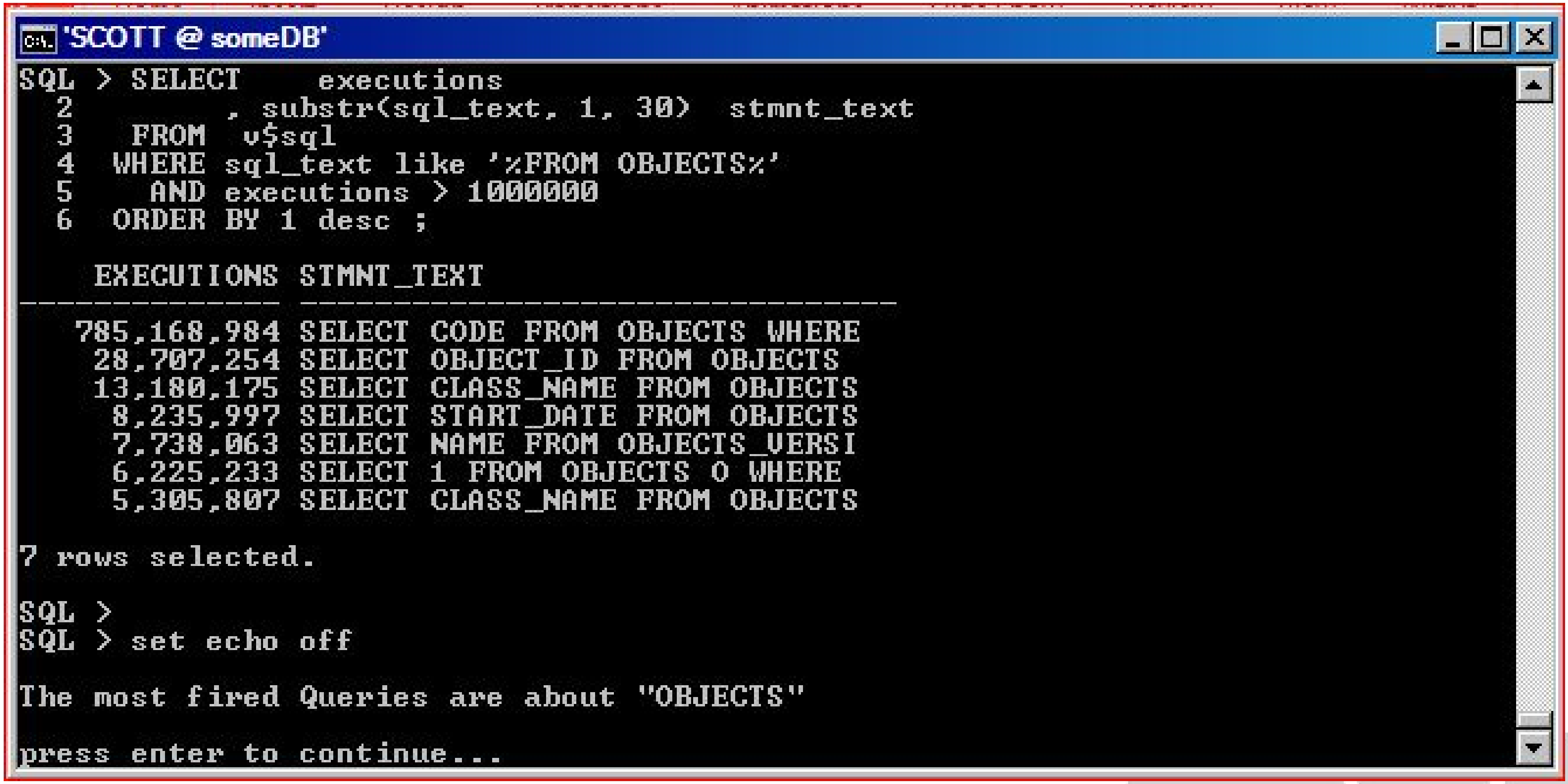
Executions	Rows Processed	Rows pe
3,215,748	3,215,600	
2,291,216	0	
1,369,876	1,369,869	
1,144,697	1,144,697	
1,144,683	778,485	
1,086,000	1,085,966	
1,041,351	987,288	
958,927	958,915	
958,921	958,923	
485,592	485,590	
484,838	224,211	
278,257	278,247	
260,626	260,626	
222,986	222,986	

```
STMNT_TEXT
------------------------------
SELECT CLASS_NAME FROM OBJECTS
SELECT CODE FROM OBJECTS WHERE
SELECT OBJECT_ID FROM OBJECTS
SELECT START_DATE FROM OBJECTS
SELECT 1 FROM OBJECTS O WHERE
SELECT CLASS_NAME FROM OBJECTS
```

PDVBV

System was only using CPU, high nr of "executes", and most stmtnts returned 1-row, very often..
Some stmts had "same nr of executes" . (total: 10M queries/ 30min = 5.5 per milisec..? Yes!)

What about this "OBJECTS" 1/2



'SCOTT @ someDB'

```
SQL > SELECT     executions
  2        , substr(sql_text, 1, 30)  stmnt_text
  3   FROM   v$sql
  4  WHERE sql_text like '%FROM OBJECTS%'
  5    AND executions > 1000000
  6  ORDER BY 1 desc ;

    EXECUTIONS STMNT_TEXT
-------------- ------------------------------
   785,168,984 SELECT CODE FROM OBJECTS WHERE
    28,707,254 SELECT OBJECT_ID FROM OBJECTS
    13,180,175 SELECT CLASS_NAME FROM OBJECTS
     8,235,997 SELECT START_DATE FROM OBJECTS
     7,738,063 SELECT NAME FROM OBJECTS_VERSI
     6,225,233 SELECT 1 FROM OBJECTS O WHERE
     5,305,807 SELECT CLASS_NAME FROM OBJECTS

7 rows selected.

SQL >
SQL > set echo off

The most fired Queries are about "OBJECTS"

press enter to continue...
```

- Highest Freq SQL is on "OBJECTS"...

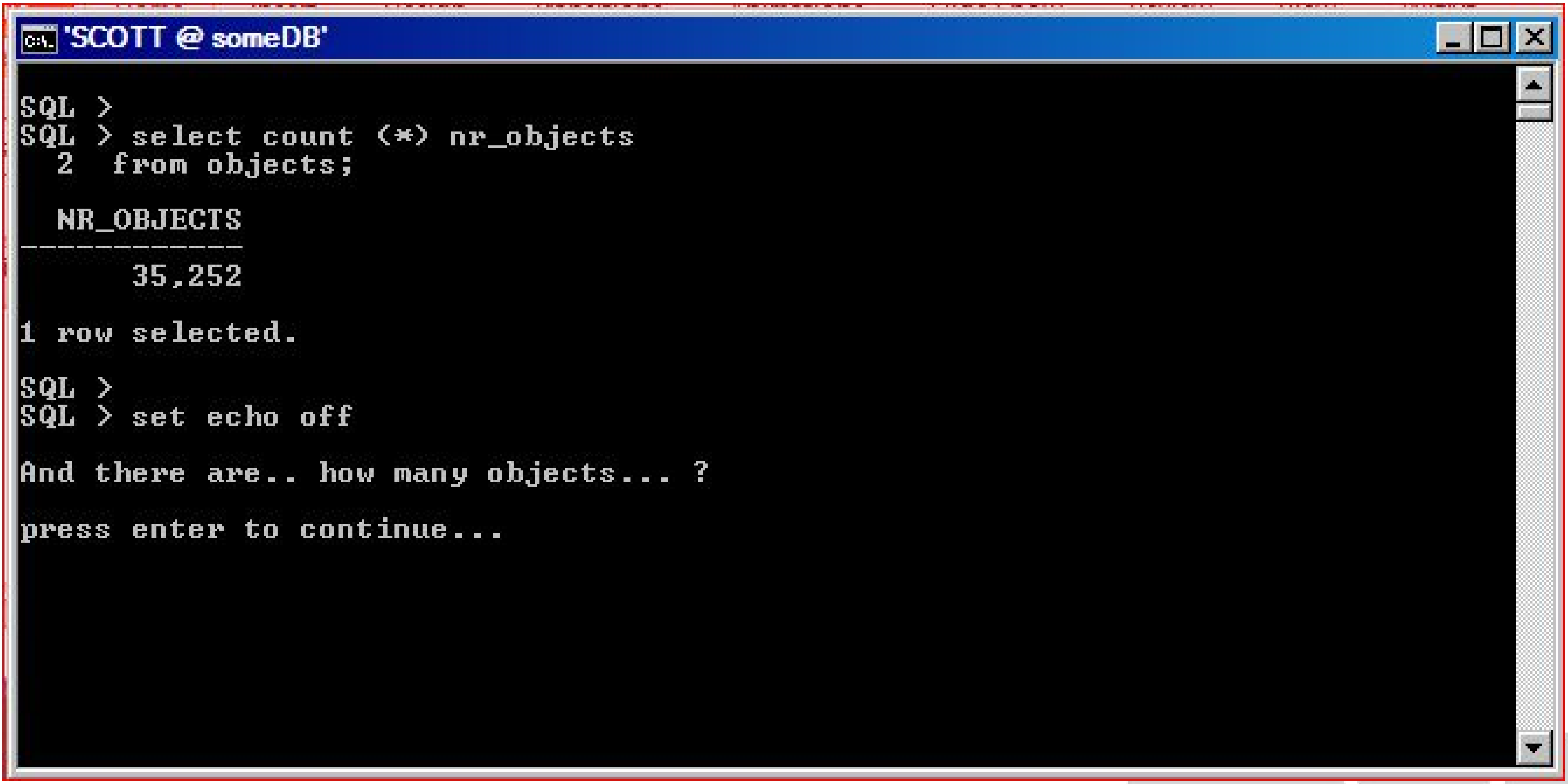
PDVBV

Investigate the "objects" thing...
Central to the app, hence lots of dependencies.

4SYNERGY

PGDAY'17 RUSSIA

How many "OBJECTS"   2/2



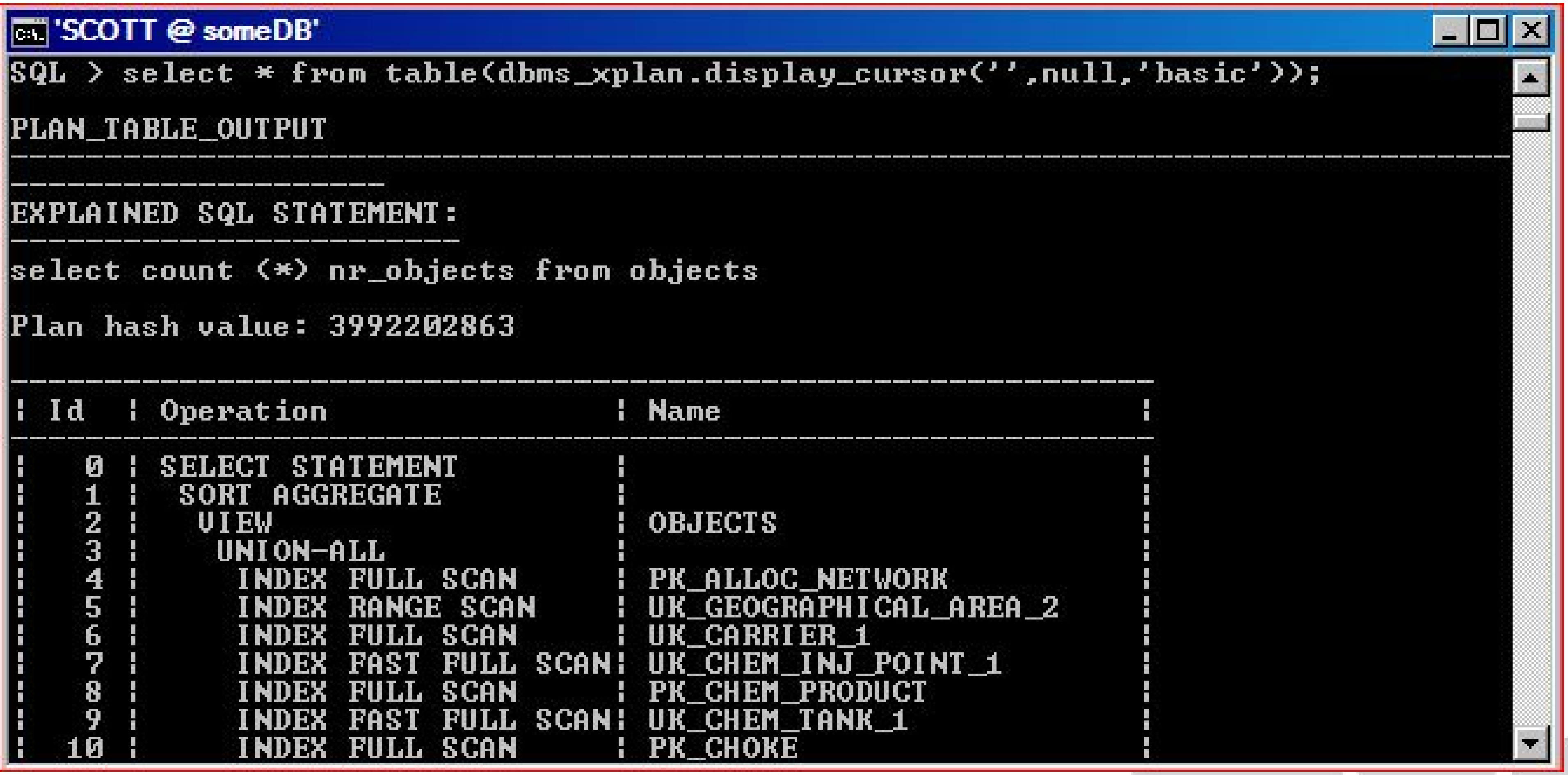
'SCOTT @ someDB'

```
SQL >
SQL > select count (*) nr_objects
  2  from objects;

  NR_OBJECTS
------------
      35,252

1 row selected.

SQL >
SQL > set echo off

And there are.. how many objects... ?

press enter to continue...
```

Only 35252 records in "OBJECTS" ...

So there are 32 Thousand records in a set that is queried 10+M times...
Tell me more...

Explain “OBJECTS” 1/4

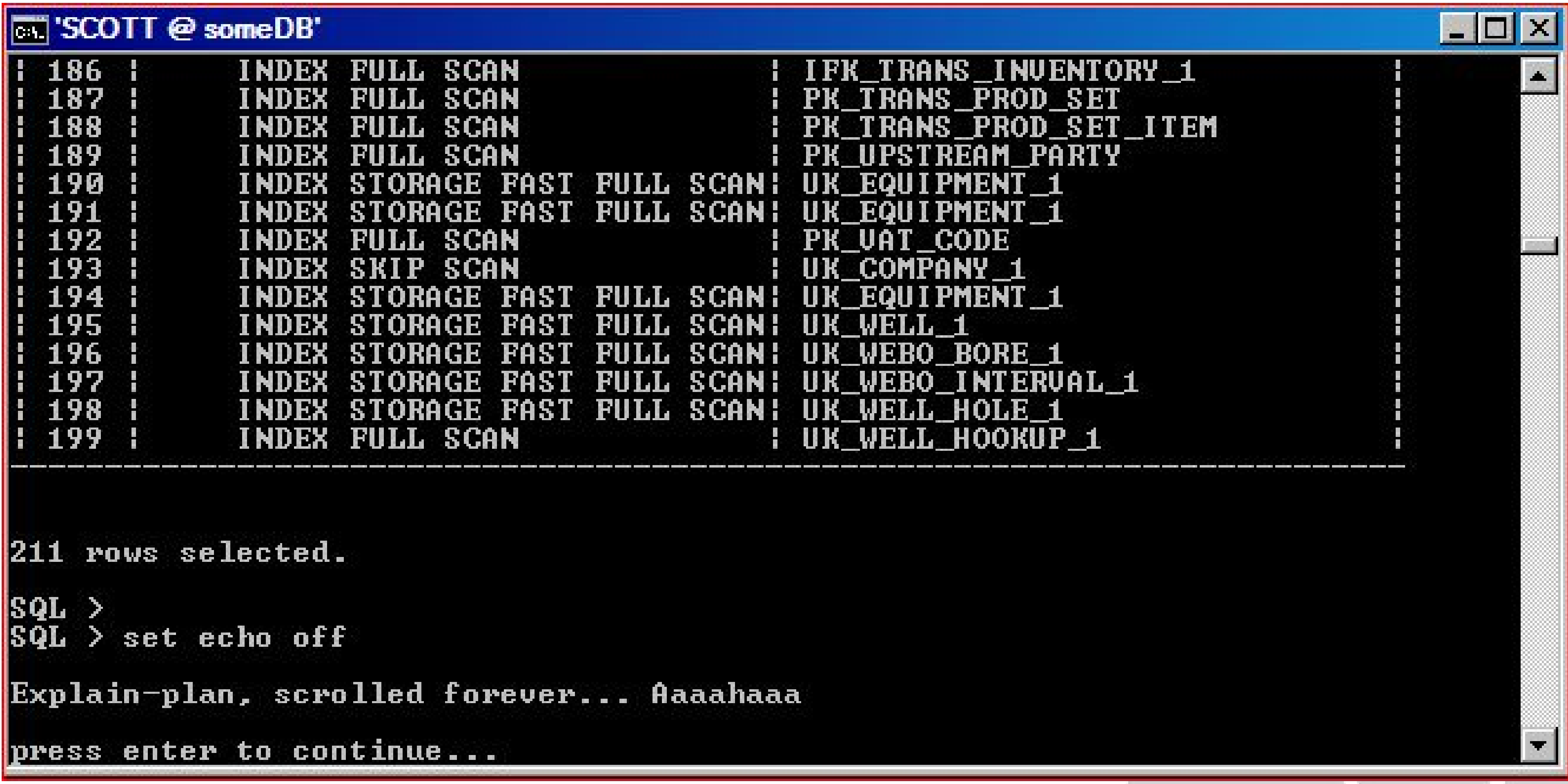

'SCOTT @ someDB'

```
SQL > select * from table(dbms_xplan.display_cursor('',null,'basic'));

PLAN_TABLE_OUTPUT
-------------------------------------------------------------------------------------
--------------------
EXPLAINED SQL STATEMENT:
------------------------
select count (*) nr_objects from objects

Plan hash value: 3992202863

-------------------------------------------------------------
| Id  | Operation                 | Name                      |
-------------------------------------------------------------
|   0 | SELECT STATEMENT          |                           |
|   1 |  SORT AGGREGATE           |                           |
|   2 |   VIEW                    | OBJECTS                   |
|   3 |    UNION-ALL              |                           |
|   4 |     INDEX FULL SCAN       | PK_ALLOC_NETWORK          |
|   5 |     INDEX RANGE SCAN      | UK_GEOGRAPHICAL_AREA_2    |
|   6 |     INDEX FULL SCAN       | UK_CARRIER_1              |
|   7 |     INDEX FAST FULL SCAN| UK_CHEM_INJ_POINT_1         |
|   8 |     INDEX FULL SCAN       | PK_CHEM_PRODUCT           |
|   9 |     INDEX FAST FULL SCAN| UK_CHEM_TANK_1              |
|  10 |     INDEX FULL SCAN       | PK_CHOKE                  |
```

xplan... “OBJECTS” is a VIEW!

Check a count(*) query on objects,
Surprise ...?

Explain "OBJECTS" 2/4


'SCOTT @ someDB'

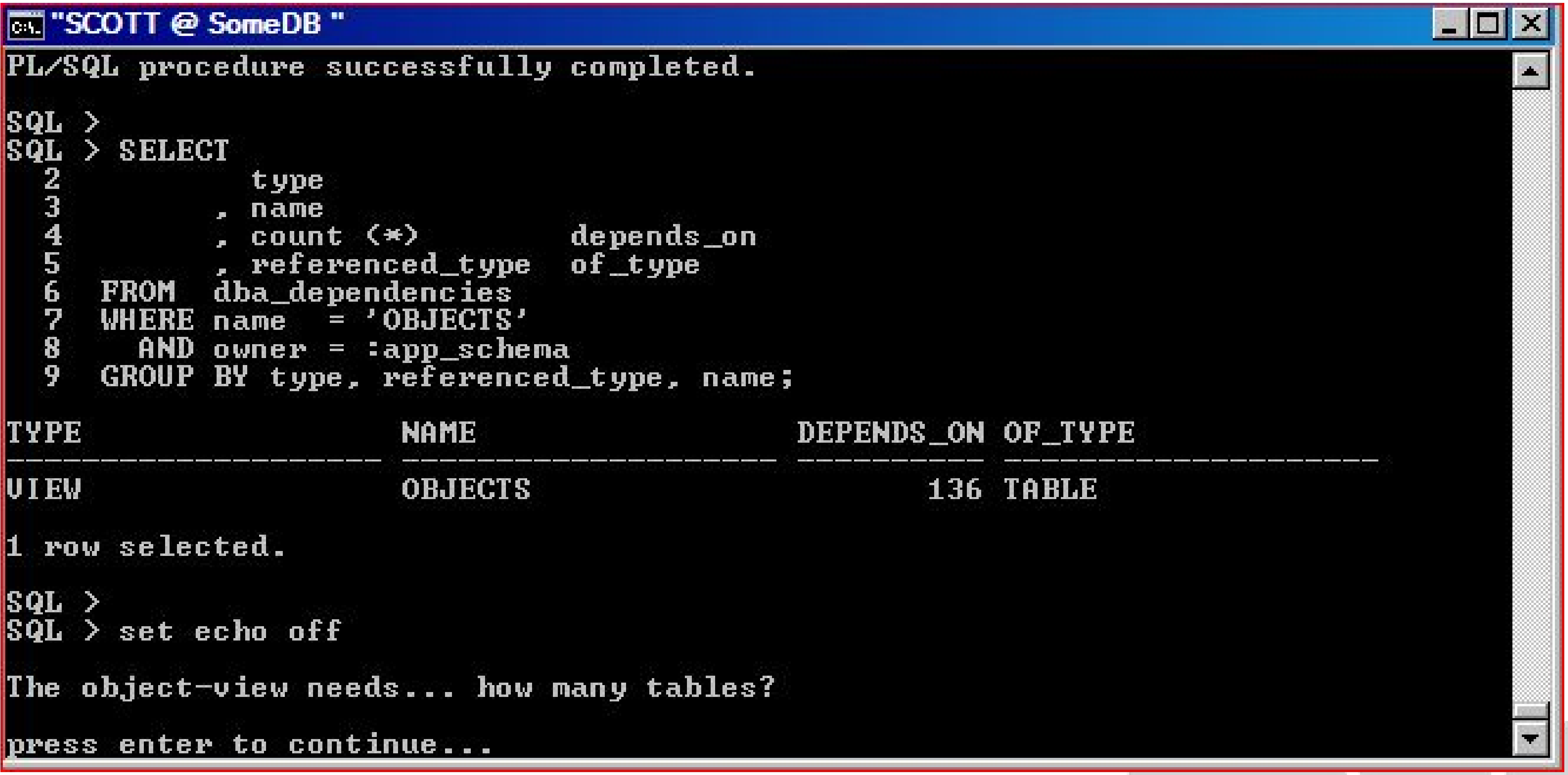
```
| 186 |      INDEX FULL SCAN                | IFK_TRANS_INVENTORY_1             |
| 187 |      INDEX FULL SCAN                | PK_TRANS_PROD_SET                 |
| 188 |      INDEX FULL SCAN                | PK_TRANS_PROD_SET_ITEM            |
| 189 |      INDEX FULL SCAN                | PK_UPSTREAM_PARTY                 |
| 190 |      INDEX STORAGE FAST FULL SCAN| UK_EQUIPMENT_1                    |
| 191 |      INDEX STORAGE FAST FULL SCAN| UK_EQUIPMENT_1                    |
| 192 |      INDEX FULL SCAN                | PK_VAT_CODE                       |
| 193 |      INDEX SKIP SCAN                | UK_COMPANY_1                      |
| 194 |      INDEX STORAGE FAST FULL SCAN| UK_EQUIPMENT_1                    |
| 195 |      INDEX STORAGE FAST FULL SCAN| UK_WELL_1                         |
| 196 |      INDEX STORAGE FAST FULL SCAN| UK_WEBO_BORE_1                    |
| 197 |      INDEX STORAGE FAST FULL SCAN| UK_WEBO_INTERVAL_1                |
| 198 |      INDEX STORAGE FAST FULL SCAN| UK_WELL_HOLE_1                    |
| 199 |      INDEX FULL SCAN                | UK_WELL_HOOKUP_1                  |
--------------------------------------------------------------------------------


211 rows selected.

SQL >
SQL > set echo off

Explain-plan, scrolled forever... Aaaahaaa

press enter to continue...
```

xplan... scrolled forever...

PDVBV

So there are 32 Thousand records in a set that is queried 10+M times...
I can smell a solution for his one already...

Dependencies of “OBJECTS” 3/4



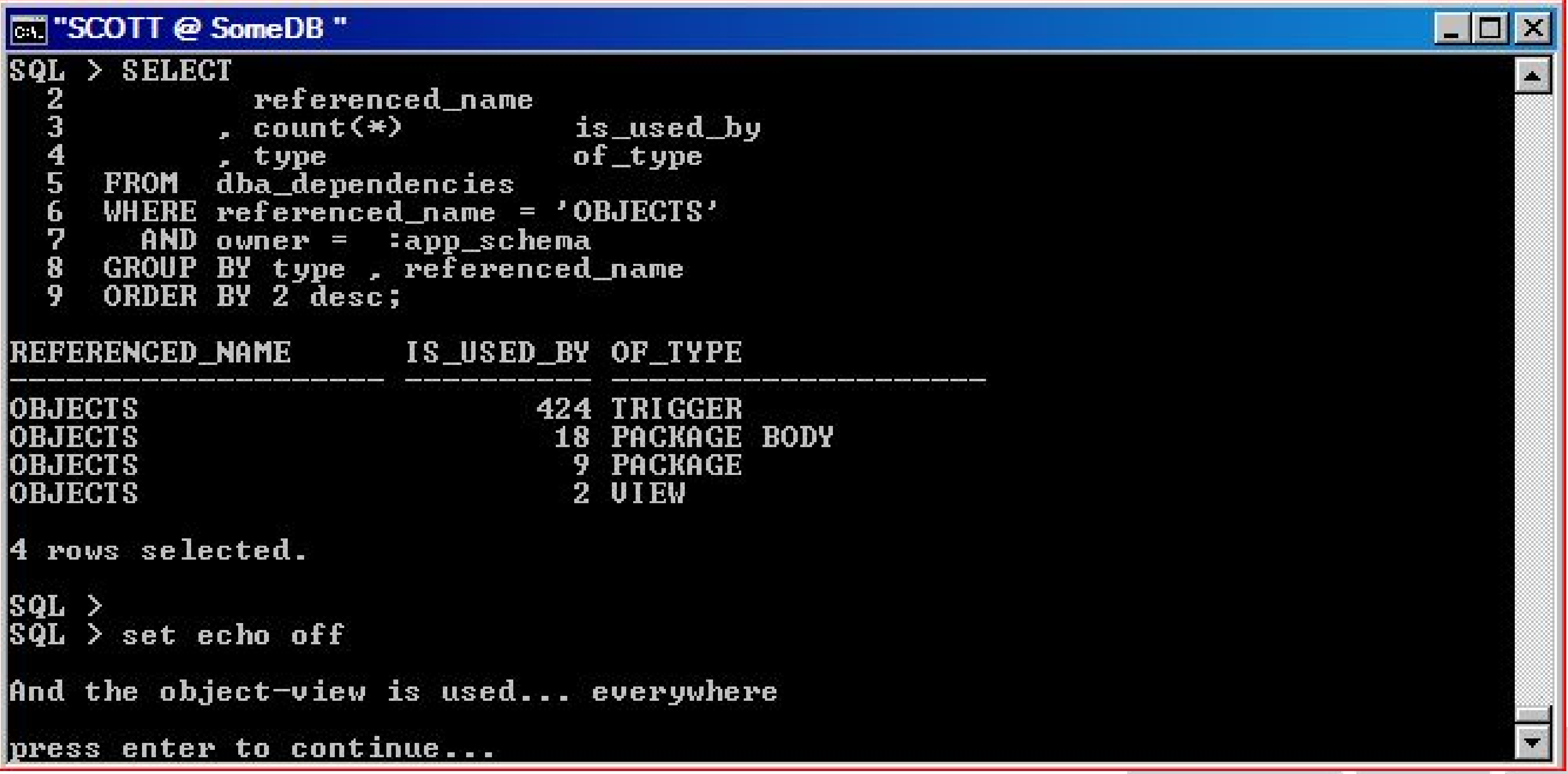
```
PL/SQL procedure successfully completed.

SQL >
SQL > SELECT
  2           type
  3         , name
  4         , count (*)        depends_on
  5         , referenced_type  of_type
  6  FROM  dba_dependencies
  7  WHERE name   = 'OBJECTS'
  8    AND owner = :app_schema
  9  GROUP BY type, referenced_type, name;

TYPE                 NAME                 DEPENDS_ON OF_TYPE
-------------------- -------------------- ---------- --------------------
VIEW                 OBJECTS                     136 TABLE

1 row selected.

SQL >
SQL > set echo off

The object-view needs... how many tables?

press enter to continue...
```

- This View covers …. 136 tables?

PDVBV

This “objects” thing depens on 136 tables
That is a lot of dependencies…

Dependencies of "OBJECTS" 4/4

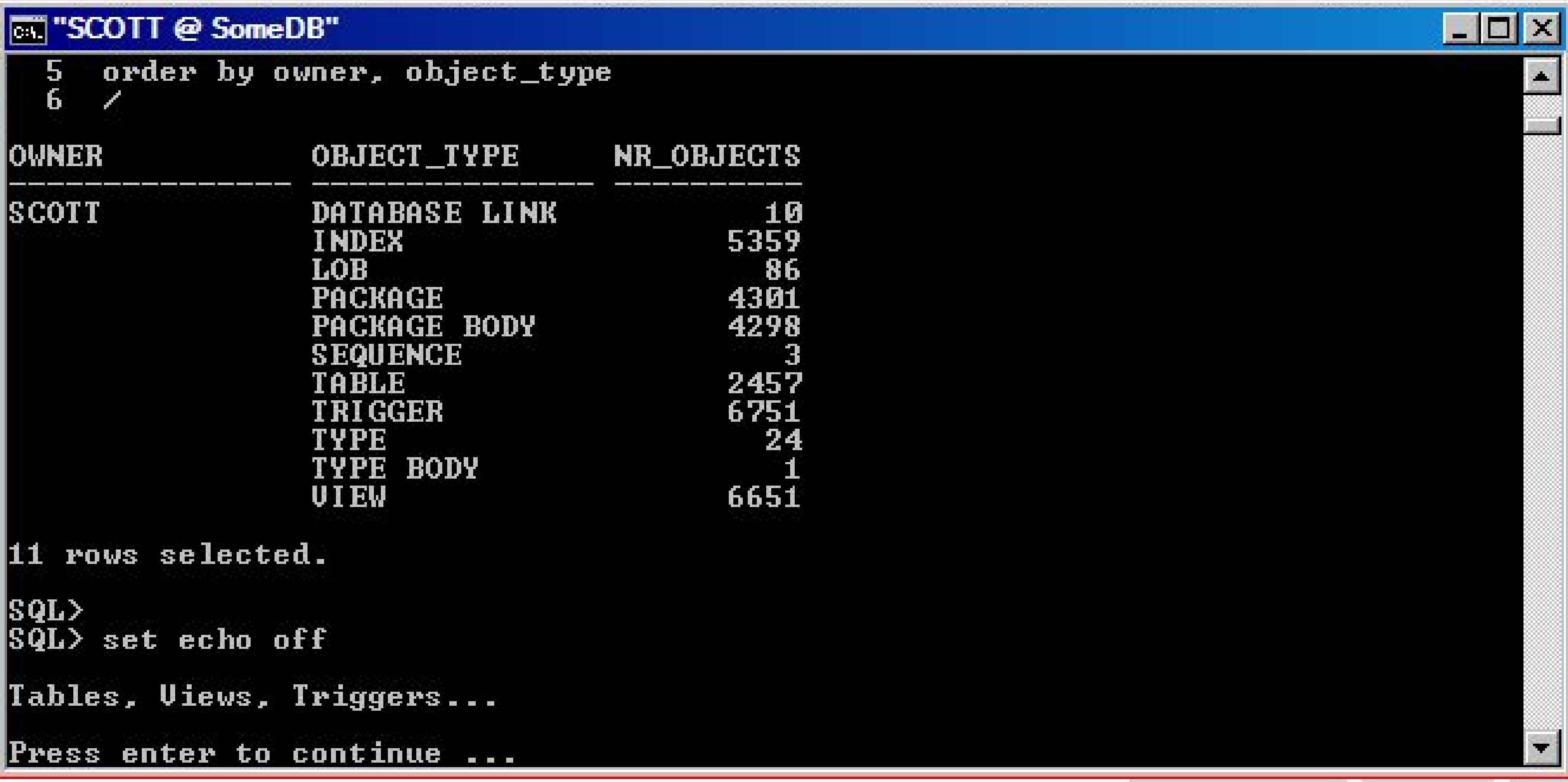


```
SQL > SELECT
  2           referenced_name
  3         , count(*)          is_used_by
  4         , type              of_type
  5   FROM  dba_dependencies
  6   WHERE referenced_name = 'OBJECTS'
  7     AND owner =  :app_schema
  8   GROUP BY type , referenced_name
  9   ORDER BY 2 desc;

REFERENCED_NAME      IS_USED_BY OF_TYPE
-------------------- ---------- --------------------
OBJECTS                     424 TRIGGER
OBJECTS                      18 PACKAGE BODY
OBJECTS                       9 PACKAGE
OBJECTS                       2 VIEW

4 rows selected.

SQL >
SQL > set echo off

And the object-view is used... everywhere

press enter to continue...
```

- **And it is used....in triggers and packages...**

And 450 other items refer to this "objects" ... Everyone referto this view..
So much for dependencies: this "objects" view is central to the application.

Inspect the Schema…   ./.


"SCOTT @ SomeDB"

```
  5  order by owner, object_type
  6  /

OWNER                OBJECT_TYPE          NR_OBJECTS
-------------------- -------------------- ----------
SCOTT                DATABASE LINK                10
                     INDEX                      5359
                     LOB                          86
                     PACKAGE                    4301
                     PACKAGE BODY               4298
                     SEQUENCE                      3
                     TABLE                      2457
                     TRIGGER                    6751
                     TYPE                         24
                     TYPE BODY                     1
                     VIEW                       6651

11 rows selected.

SQL>
SQL> set echo off

Tables, Views, Triggers...

Press enter to continue ...
```

- **1000s of schema-objects…**

PDVBV

The total schema is also quite large.. 2400 tables, 4300 packages, 6000 views 6000 triggers.. .
This thing is complex! Ooops..

So far.. : CPU Busy on Objects ./.

- We know the DB is busy on CPU…
  - Top SQL: Retrieving “objects”
  - 35000 rows in view over 136 tables

- SQL… 10M sql-executes in 1800sec.
  - Most SQL returns 0 or 1 row.

  “On average”: each object is retrieved only ... 10x/min

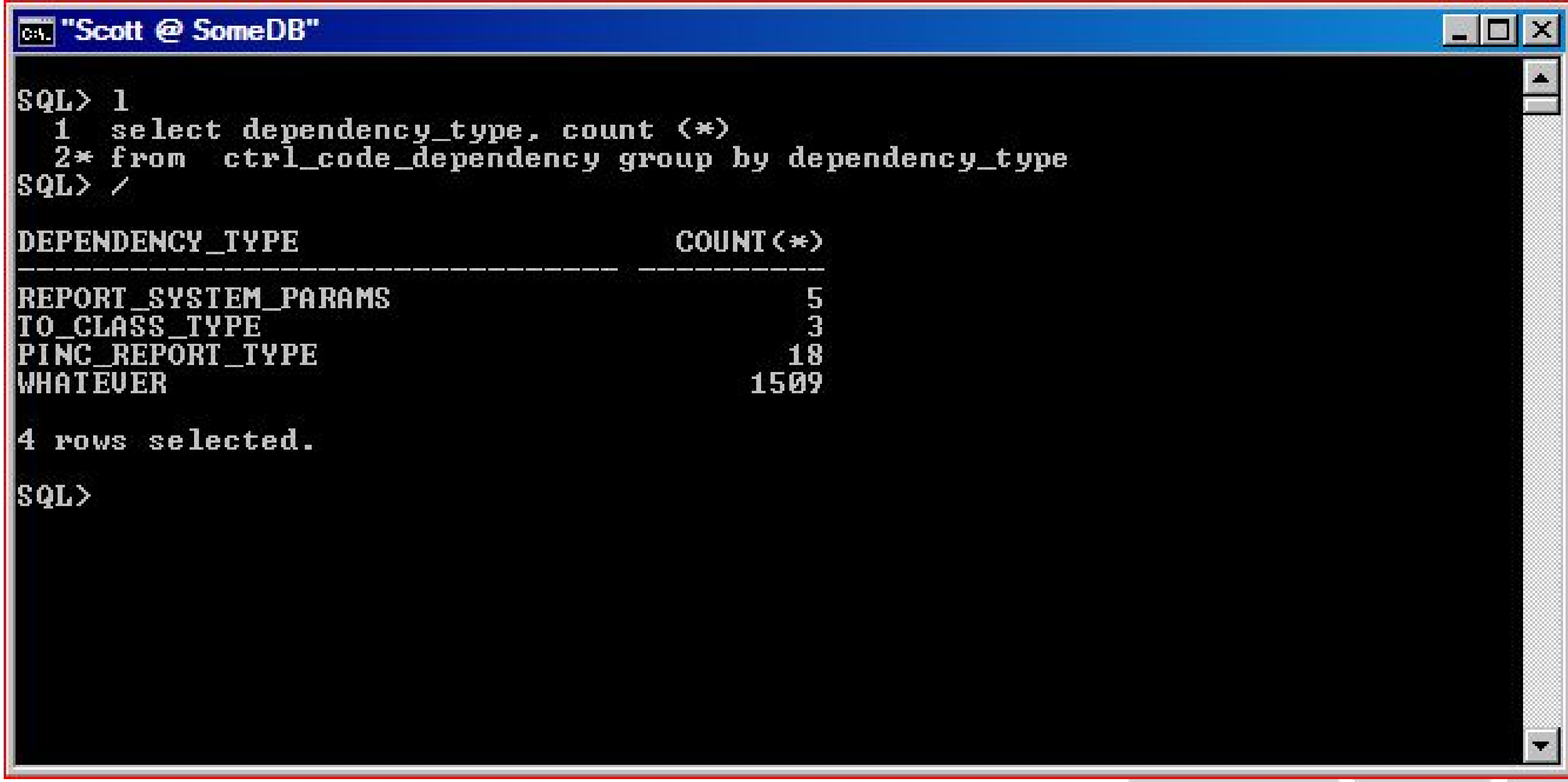
  Average fetch is only 154 “gets” (logica

- How to Fix…?
  - keep digging…

PDVBV

We know it is “Busy”… Why… “this is the Application”..
Focus on fix (and cannot change the app much)

What you sometimes find…


"Scott @ SomeDB"

```
SQL> l
  1  select dependency_type, count (*)
  2* from  ctrl_code_dependency group by dependency_type
SQL> /

DEPENDENCY_TYPE                    COUNT(*)
-------------------------------- ----------
REPORT_SYSTEM_PARAMS                      5
TO_CLASS_TYPE                             3
PINC_REPORT_TYPE                         18
WHATEVER                               1509

4 rows selected.

SQL>
```

There are 5 params, 3+18 types, and 1509 of ..Whatever?

PDVBV

What happens when a team of Dev is developing on some "generic" model

Views -> Functions -> View -> Tbls ./.

- Top to bottom first…

- “Rich” application, uses Views to define “things”
  - “generated code”, including INSTEAD-OF triggers

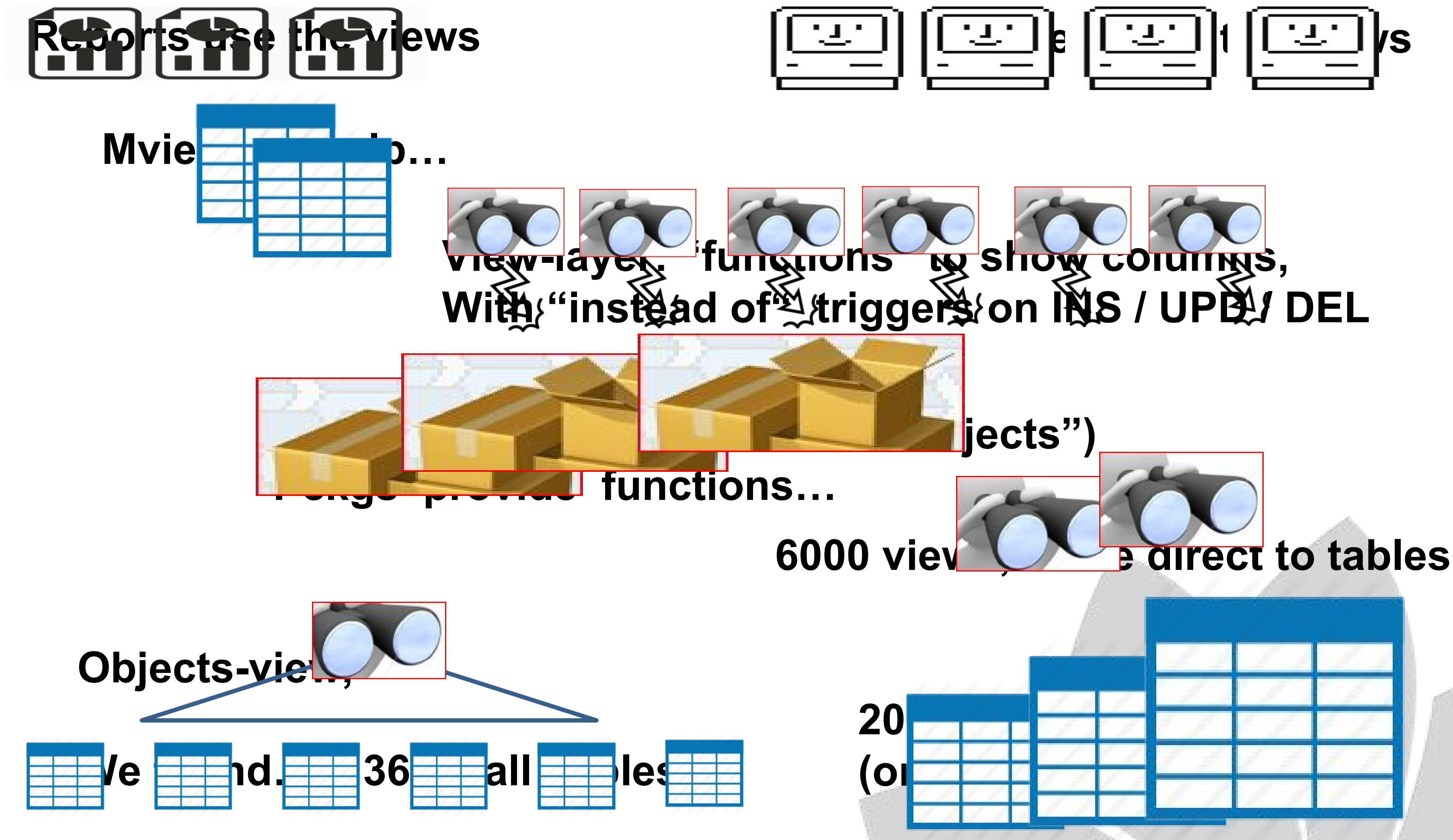
- The Views use Functions (mostly in Pckgs)…
  - both in Select and in Where (and in joins)
  - Get_name_of_mything (thing_id) returns varchar2…

- Functions query “Objects”
  - Objects is a view… (of 136 tables)

- Let me try explain…

Show that the object-view created with best of intentions… Generic model, very Flexible, very “Rich” in functionality. This application is “deployed” differently per client, per instance.

Many layers between User and Data.


Reports use the views
Mvie o…
View-layer: "functions" to show columns,
With "instead of" triggers on INS / UPD / DEL
jects")
Pckgs provide functions…
6000 view e direct to tables
Objects-view,
We nd. 36 all les
20
(or

PDVBV

Show that the object-view created with best of intentions... Generic model, very Flexible, very "Rich" in functionality. This application is "deployed" differently per client, per instance.

Root Cause (we think): Views + pkgs

```
Create View RichView as (
Select
   pkg.get_attrib_f1 (id) as atrrib1
   Pkg.get_attrib_f2 (id) as attrib2
    .. Etc..
From SomeTable [, MoreTables ]
[ whereclause, some with functions]
);
```

Note: Generic, and potentially Flexible system…

(Add instead-of triggers.. Make it more Flexible still..)

Views defiend using packaged-functions… flexible, generic, potentially very "rich" in functionality.
And it worked fine in testing…

Now join + filter using those views… ./.

```
Select  v1.attrib1, v2.attrrib2,  etc …..
From    Richview1 v1
   ,  Richview2 v2
Where
   v1.attrib1 = v2.attrib1
And   v1.attrib2 > :x
And   v2.attrib2 = :y
And  .. More…
```

Columns … Cause function calls

Joins… Cause Function-calls

Where-filters… Cause Function calls.

Now start using those views… and they start calling functions..
Often the same function with the same arguments.. (room for invstigation + optimization, later)

Played with Xplan + Auto-trace-Stats

```
Select
     pkg.get_attrib_f1 (id) as column1
From    SomeTable
Where   Key = :arg1
```

- If you “Explain” this:
  - index + table, probably … 3 Gets
  - Looks very Efficient…
- But if you autotrace it… (1000s gets…)
- Now imagine doing Aggregates, on views…

Initial investigations using “explain button” were misleading.. Dev: “All queries using indexes….
But the AWR, the Lab128, and the Autotrace-stats told us: there is effort hidden in Functions.

Solutions… (multiple) ./.

PDVBV

So here I was... Fix IT! (boss: make it So, Make it Go !)

Solutions… 1st: Cache?

- Data will Change…
  - Especially when system is “used” during critical periods…
- DIY: Cache in Array ? … No!
  - Needs even more “Code”. Risky.
  - Irregular responses if cache needs refreshing.
  - And .. This defies the “ACID” property of the database ..
- Function result cache ? … Perfect use case.
  - Doesnt work… ??? WTF ??
  - Workaround (by Peter Swier):
    - Multiple Views, and cleverly search them 1 by 1…
  - (needs separate ppt.. Ask me @ coffee…)

Cache was complicated for this set of data, and function-resultcache did not work at first.. The “object” had too many dependencies…

Solutions… 2nd: "Eliminate" ?

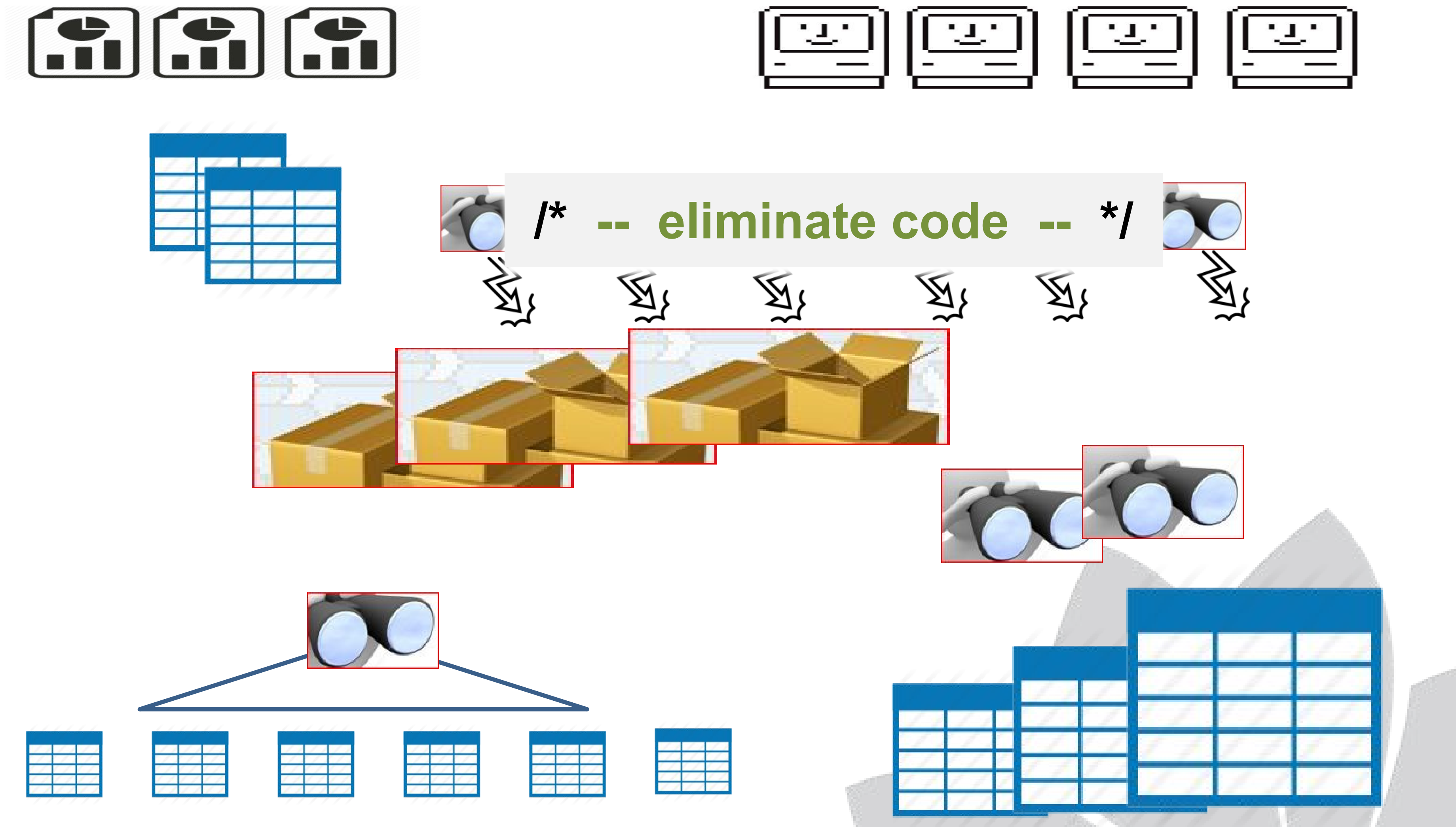
- We "removed" some code (we asked the Dev team)
  - Examination of "top" SQL and "problem" components.
  - "Generated code" - much of it seemed irrelevant.
  - Removed columns from views and reports.
  - Create "clone-views" with only the necessary columns…
- Old Fashioned /* out comment unused items */

- The result was:
  - Less calls… (less executes, less cpu !)
  - Faster reports and screens (eliminated some Mviews!)

- Possible Maintenance work in Future!

Elimination: heavy work, just don't do it..
The fastest SQL is the one you don't do..

Developers /* eliminate code */ …



Once we got a "knowledgable" developer on the reports.. They started to Fly!
Trick was to /* eliminate */ unused columns from queries.. That reduced the calls to fuctions !

4SYNERGY

PGDAY'17 RUSSIA

Solutions… 3rd: Bypass ?

- более умное решение
- Go find “Data” directly…

- For “heavy” reports,
- For critical, or high-frequency views/functions…

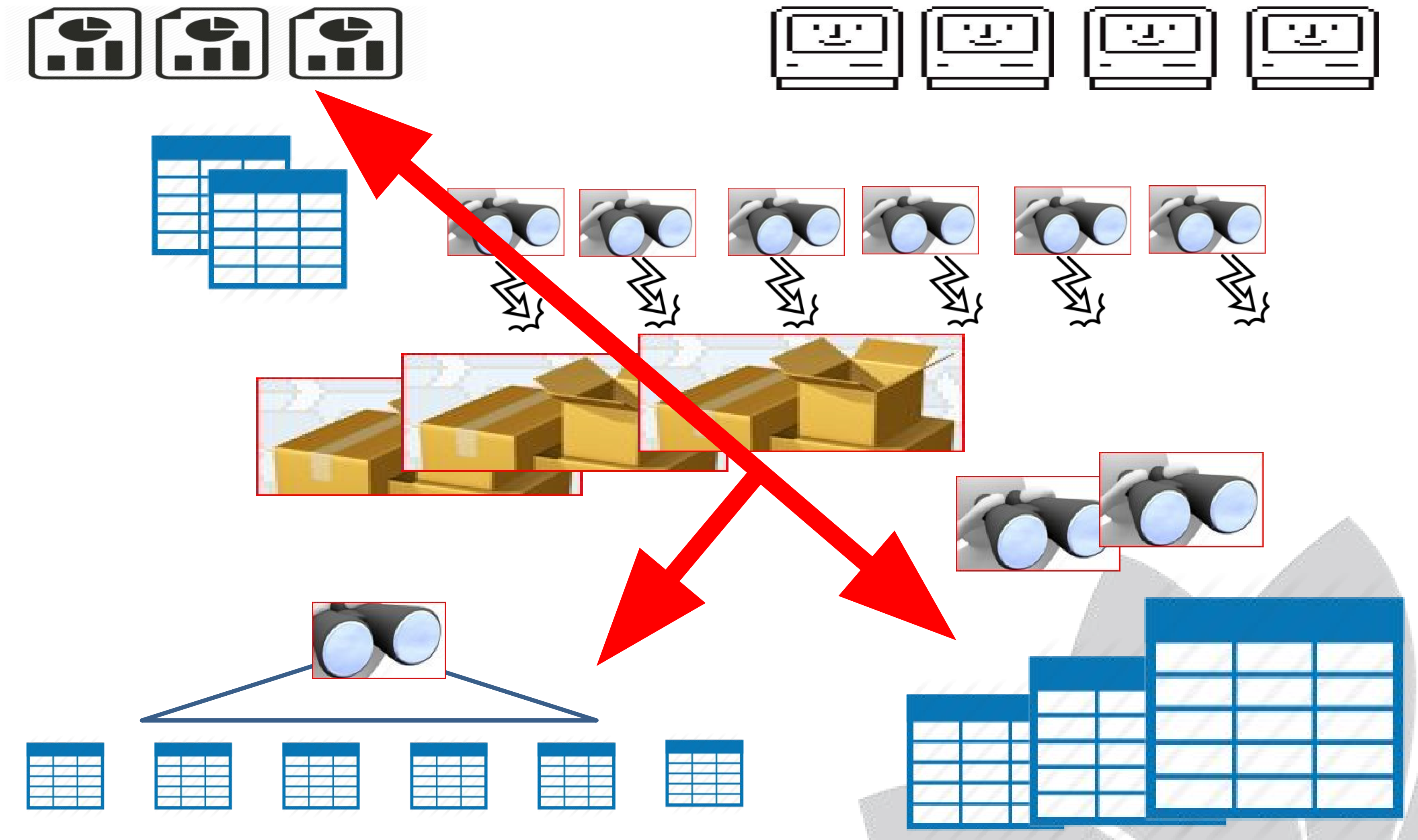
- Re-Code “logic” to bypass as much as possible

- Expensive
- Hard-coded…
- Maintenance!

PDVBV

The smarter solution…
Show list of multiple solutions: direct-query, pre-query, function-cache..

Developers to Bypass layers…



PDVBV

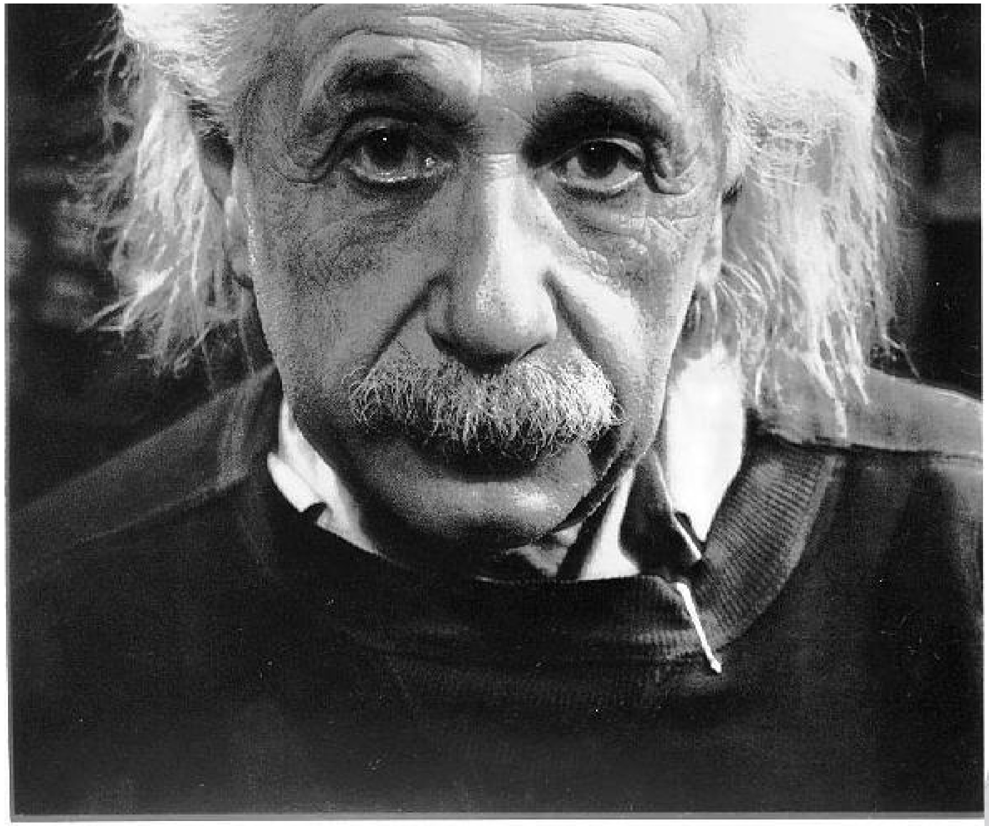
Once we got a “knowledgable” developer on the reports.. They started to Fly!

Summary, tips.

- 3 solutions: Cache / Eliminate / Bypass…
- Cannot re-write total app (yet) …
  - Some clever, and useful functionality in there…
- Don’t want to criticize a “good concept”
  - Good intentions (Respect), Very Clever Architect (Respect)
- Test, Test Early, Test Realistic (volume!)
- Think about usage (do not rely on KIWI)
- Monitor usage… (it was never intended to…)
- Evaluate… (are we still doing the right thing?)

I cannot solve world-hunger all alone. I also respect the “designers” for coming up with well-meaning, very clever, and very rich functionality… 12yrs later it doesn’t work out well..

He got it ...



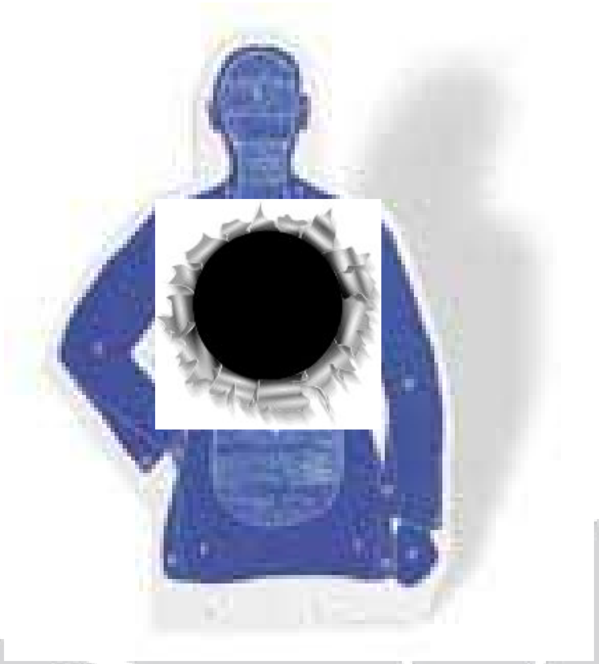
“If you can't explain it simply, you don't understand it well enough”

As Simple as Possible, but not too simple
Simplicity is a Requirement – but Comlexity just sells better (EWD).

4SYNERGY

Quick Q & A (3 min ;-) 3 .. 2 .. 1 .. Zero

- Questions ?
- Reactions ?
- Experiences from the audience ?

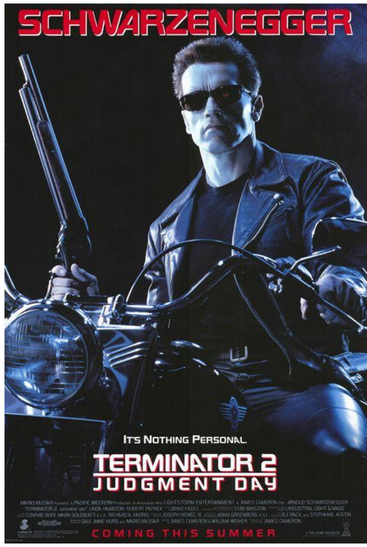


PDVBV

Question and Answer time. Discussion welcome (what about that Razor?)
Teach me something: Tell me where you do NOT AGREE.

Fixes, if needed…

- “Elimination”: don’t run the component.
  - Best option!
- “Optimization”: make it faster.
  - Realistic option (hopefully)
- “Containment” : run the item less frequent.
  - (= Worst option; It Will Be Back!
- Do-Nothing (KIWI) :
  - IF… you are confident about workload and hardware.
  - Dynanic-SQL…? Hmm; Single Threaded work…? Never!



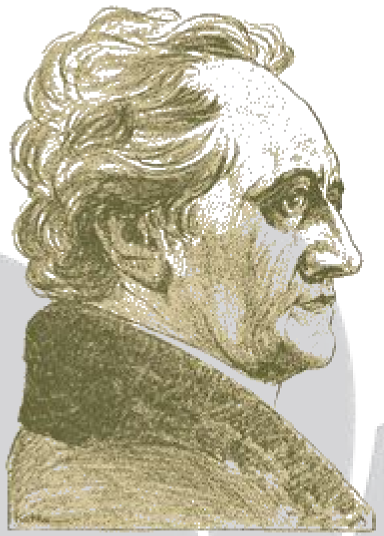
If and How to fix will depend on your situation, but you basically have those options.
And nowadays, a lot of problems get “killed by Iron”. Capacity is becoming cheaper all the time.

Don’t Take my word for it… 2/2

SimpleOracleDba . Blogspot . com (my ramblings)
And do some investigation yourself …

Homework: Check your team !
- knowledge
- procedures
- Exercise

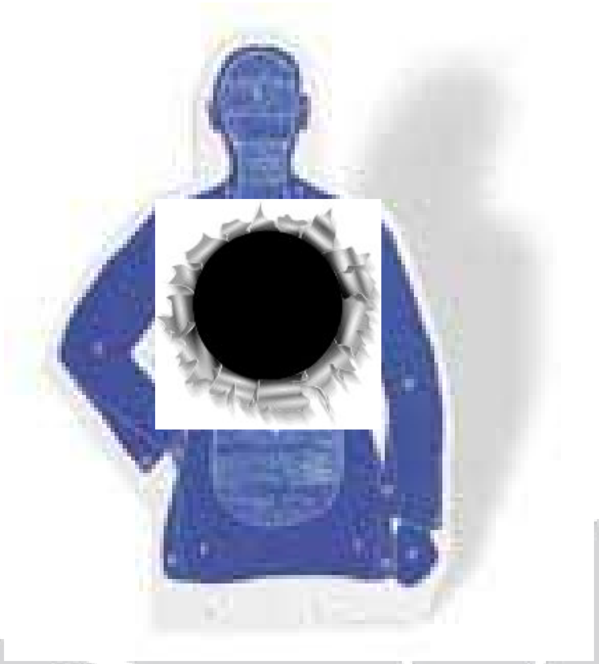
And keep it Simple!
Goethe: Limitation shows the Master.



Majority of times, I have been WRONG.
So go see for yourself – but don’t complicate life. “In der Beschrankung zeight sich der Meister”

Quick Q & A (3 min ;-) 3 .. 2 .. 1 .. Zero

- Questions ?
- Reactions ?
- Experiences from the audience ?



PDVBV

Question and Answer time. Discussion welcome (what about that Razor?)
Teach me something: Tell me where you do NOT AGREE.

Many layers between User and Data.

Reports use the views

Screens use the views

Mviews to help…

View-layer: “functions” to show columns,
With “instead of“ triggers on INS / UPD / DEL

4200 packages (18 depend on “objects”)
Pckgs provide functions…

6000 views, some direct to tables

Objects-view,

2000 other Tables
(only a few in use)

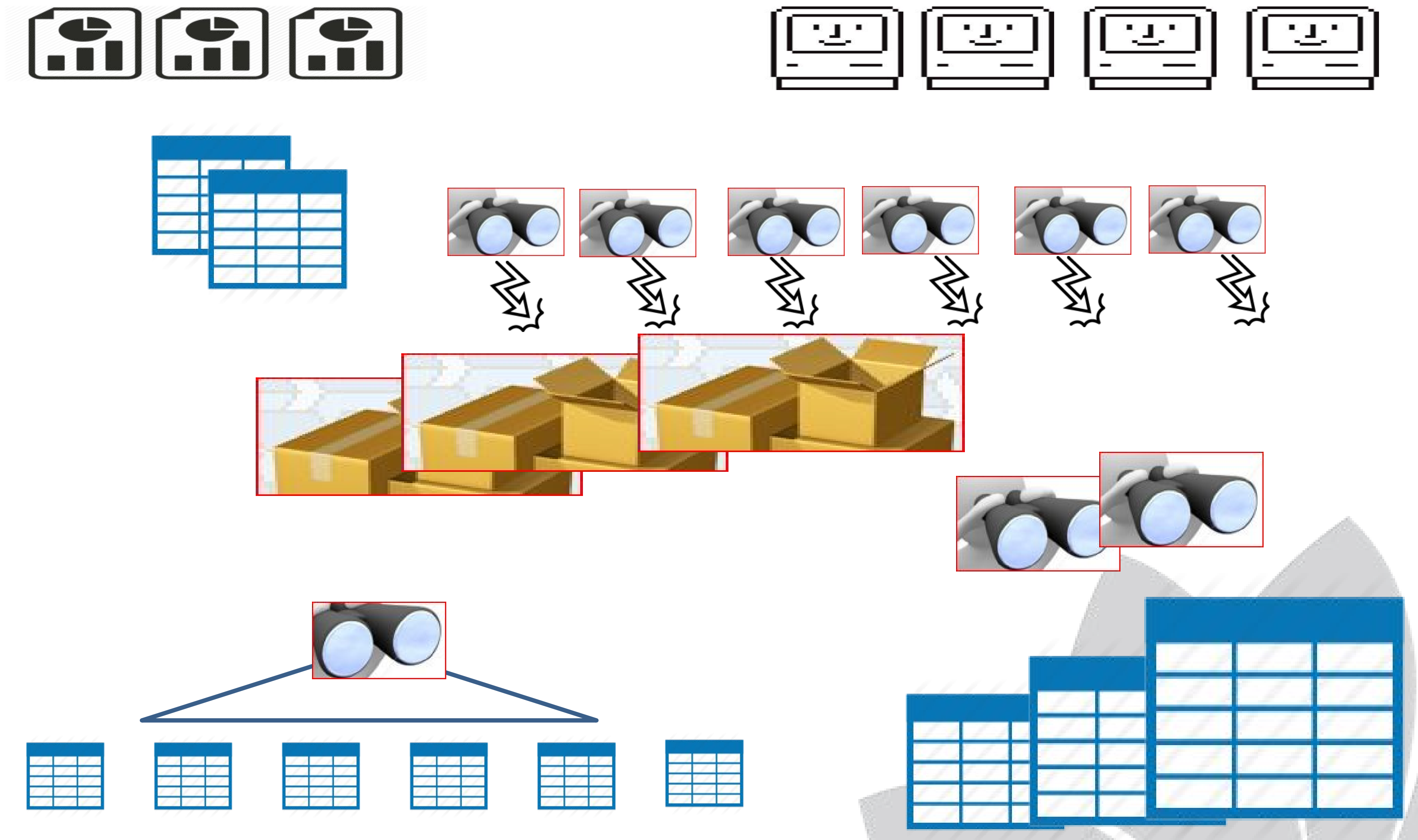
We found… 136 small Tables

Show that the object-view created with best of intentions... Generic model, very Flexible, very “Rich” in functionality. This application is “deployed” differently per client, per instance.

4SYNERGY

PGDAY'17 RUSSIA

Many layers between User and Data.



PDVBV

Show that the object-view created with best of intentions… Generic model, very Flexible, very “Rich” in functionality. This application is “deployed” differently per client, per instance.

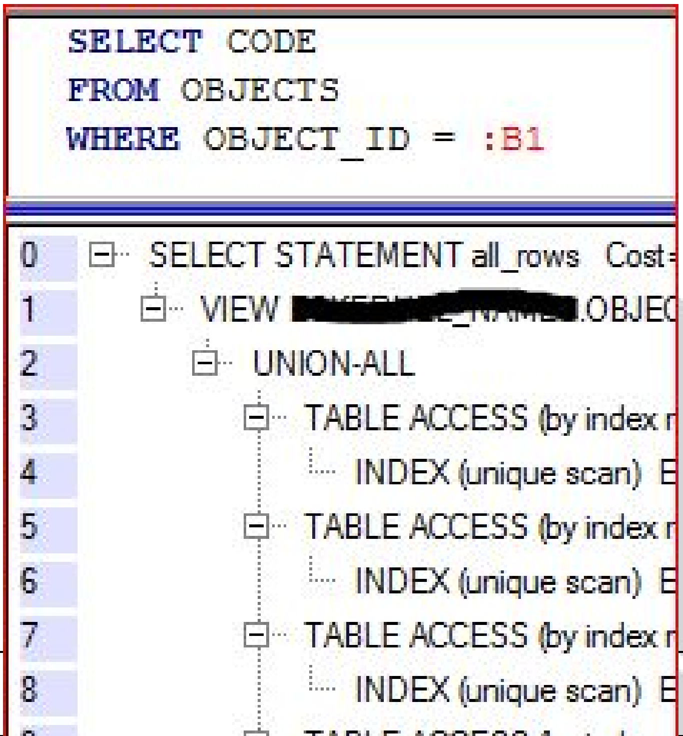
Findings.....

AWR and Lab128 show:

- **CPU only (hence 3x gain from Exa)**
- **Very high nr of “Executes”**
- **(and Several SQLs at same-frequency)**

Image: beard..

Most executed SQL:



```
SELECT CODE
FROM OBJECTS
WHERE OBJECT_ID = :B1
```

```
0  SELECT STATEMENT all_rows  Cost
1    VIEW [redacted].OBJE
2      UNION-ALL
3        TABLE ACCESS (by index r
4          INDEX (unique scan)  E
5        TABLE ACCESS (by index r
6          INDEX (unique scan)  E
7        TABLE ACCESS (by index r
8          INDEX (unique scan)  E
```

System was only using CPU, high nr of “executes”,
Some stmts had “same nr of executes” ..

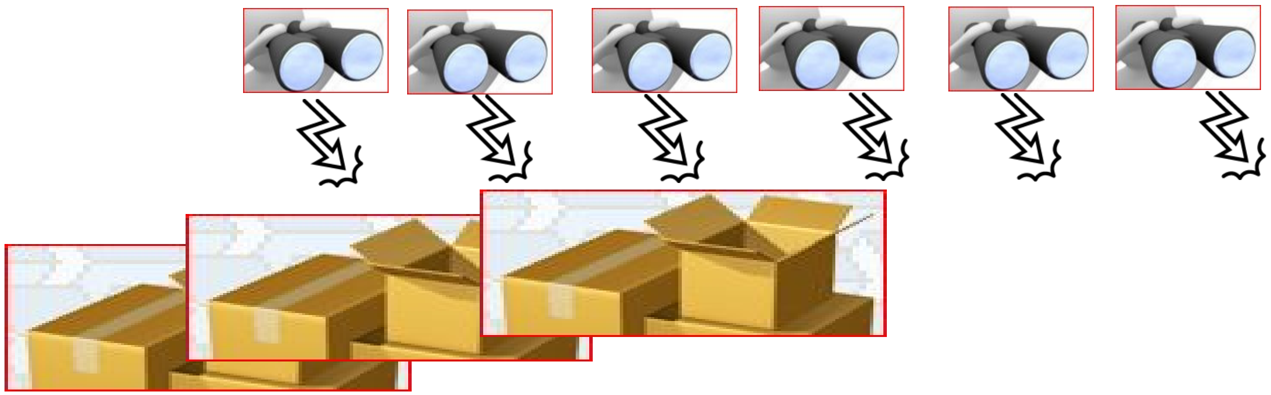
4SYNERGY

PGDAY'17 RUSSIA

Quick Q & A (3 min ;-) 3 .. 2 .. 1 .. Zero

PDVBV

Question and Answer time. Discussion welcome (what about that Razor?)
Teach me something: Tell me where you do NOT AGREE.



PDVBV