

Doug Burns

Statistics on Partitioned Objects



Introduction

- Introduction
- Simple Fundamentals
- Statistics on Partitioned Objects
- The Quality/Performance Trade-off
- Aggregation Scenarios
- Alternative Strategies
- Incremental Statistics
- Conclusions and References

Introduction

- Who am I?
- Why am I talking?
- Setting Expectations

Who am I?

- Possibly a question some of us will be asking ourselves at 8:30 am tomorrow after tonight's party
- I am Doug
 - Doug I am
 - Actually I am Douglas
 - ... or, if you're Scottish, Dougie or Doogie
- I'm not from round here
 - You will have probably noticed that already
 - See Twitter @doug_conference for lots of whining about my 21 hour journey

A Bitter Old ~~Drunk~~ Man



A Pioneer



A Sports Fan



A Family Man



A Performance Guy



1986

Zilog Z80A (3.5MHz)

32KB Usable RAM

Yes, Cary, we used profiles!

Why am I talking?

- Partitioned objects are a given when working with large databases
- Maintaining statistics on partitioned objects is one of the primary challenges of the DW designer/developer/DBA
- There are many options that vary between versions but the fundamental challenges are the same
- Trade-off between statistics quality and collection effort
- People keep getting it wrong!

Setting Expectations

- What I will and won't include
 - No Histograms
 - No Sampling Sizes
 - No Indexes
 - No Detail
- Level of depth – paper
- WeDoNotUseDemos
- A lot to get through!
- Questions

Simple Fundamentals

- Introduction
- Simple Fundamentals
- Statistics on Partitioned Objects
- The Quality/Performance Trade-off
- Aggregation Scenarios
- Alternative Strategies
- Incremental Statistics
- Conclusions and References

Cost-Based Optimiser

- The CBO evaluates potential execution plans using
 - Rules and formulae embedded in the code
 - Some control through
 - Configuration parameters
 - Hints
 - Statistics
 - Describing the content of data objects (Object Statistics)
 - e.g. Tables, Indexes, Clusters
 - Describing system characteristics (System Statistics)

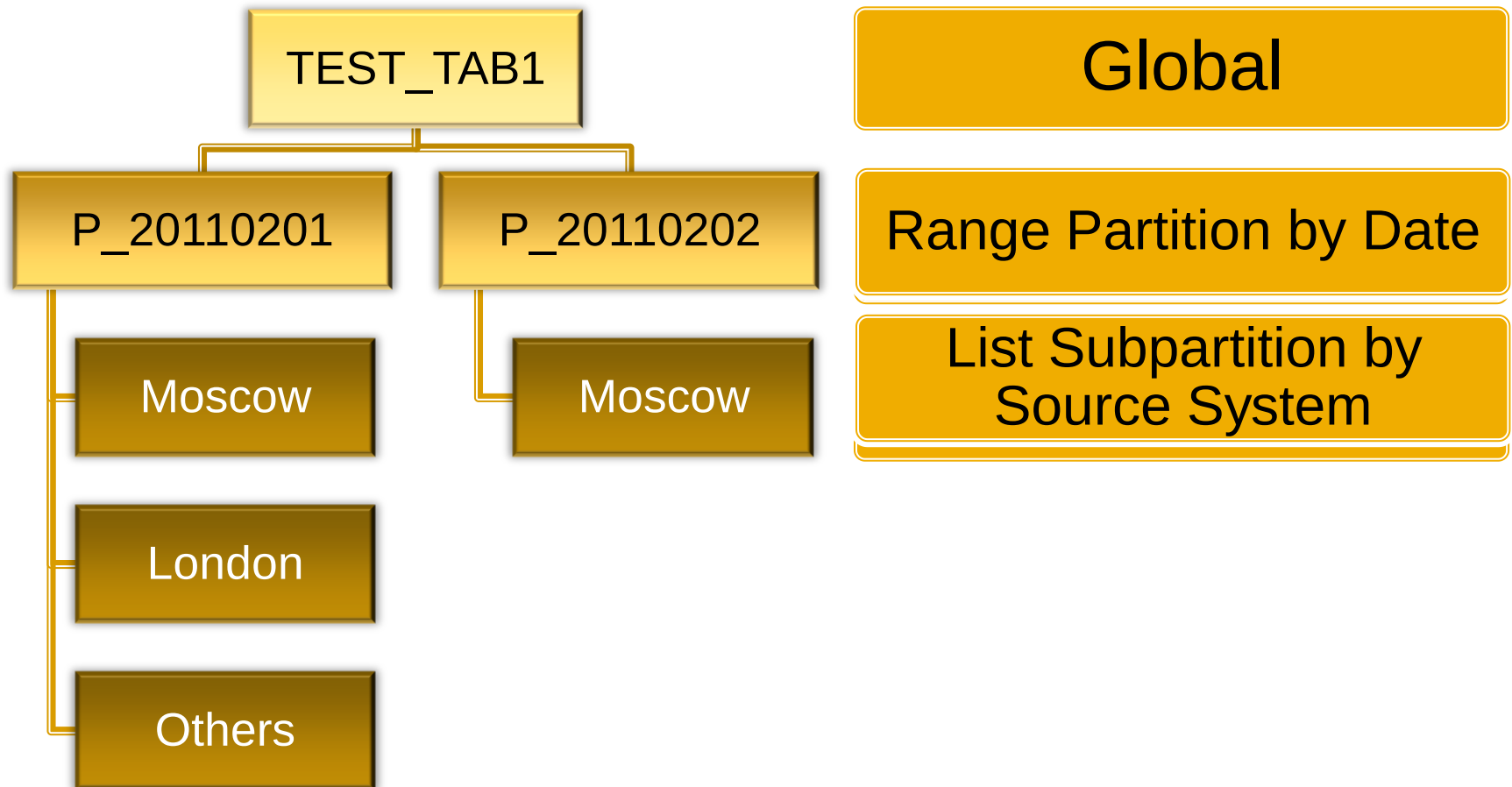
Statistics Quality

- The CBO uses statistics to estimate row source cardinalities
 - How many rows do we expect a specific operation to return
 - Primary driver in selecting the best operations to perform and their order
- Inaccurate or missing statistics are the most common cause of sub-optimal execution plans
- Hard work on designing and implementing appropriate statistics maintenance will pay off across the system

Statistics on Partitioned Objects

- Introduction
- Simple Fundamentals
- **Statistics on Partitioned Objects**
- The Quality/Performance Trade-off
- Aggregation Scenarios
- Alternative Strategies
- Incremental Statistics
- Conclusions and References

Statistics on Partitioned Objects



Statistics at all levels

- Global
 - Describe the entire table or index and all of its underlying partitions and subpartitions *as a whole*
 - Important – **GLOBAL_STATS=YES/NO**
- Partition
 - Describe individual partitions and potentially the underlying subpartitions *as a whole*
 - Important – **GLOBAL_STATS=YES/NO**
- Subpartition
 - Describe individual subpartitions
 - Implicitly, **GLOBAL_STATS=YES**

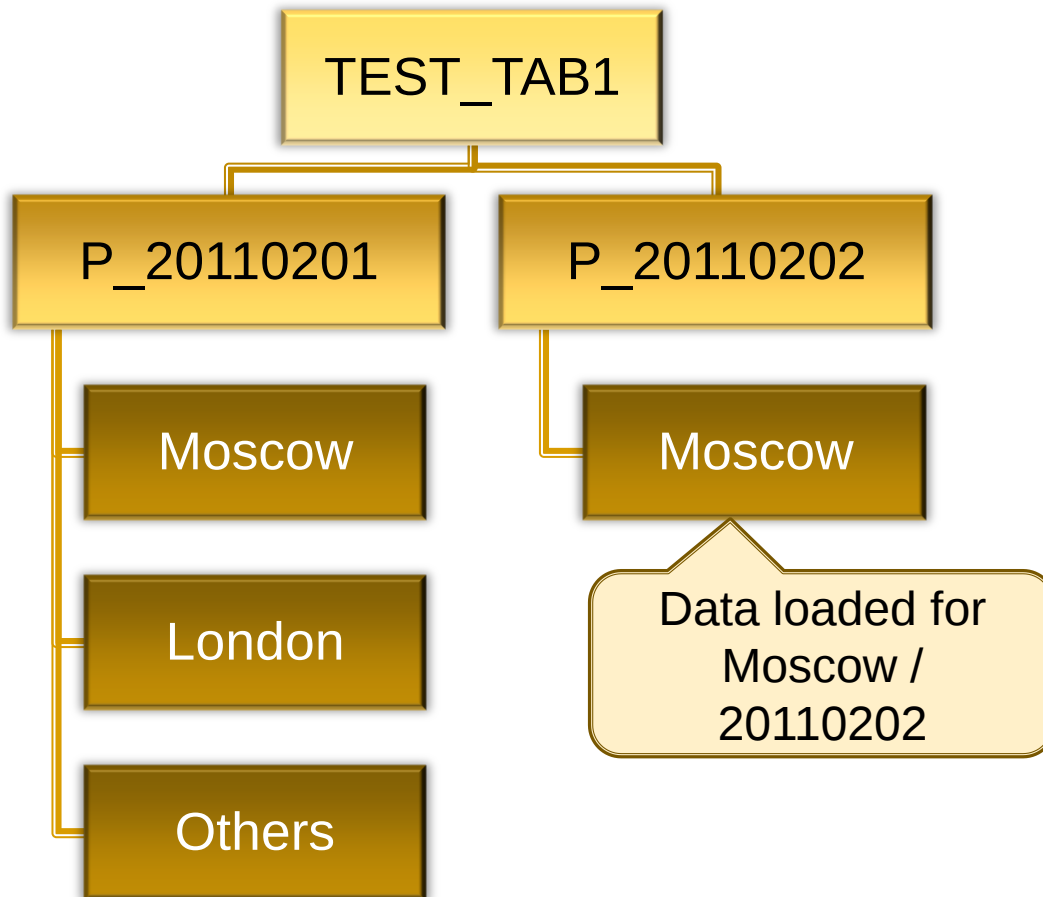
How Statistics Levels are used

- If a statement accesses multiple partitions the CBO will use Global Statistics.
- If a statement is able to limit access to *a single partition*, then the partition statistics can be used.
- If a statement accesses a single subpartition, then subpartition statistics can be used. However, prior to 10.2.0.4, subpartition statistics are rarely used.
- For most applications you will need *both Global and Partition stats* for the CBO to operate effectively

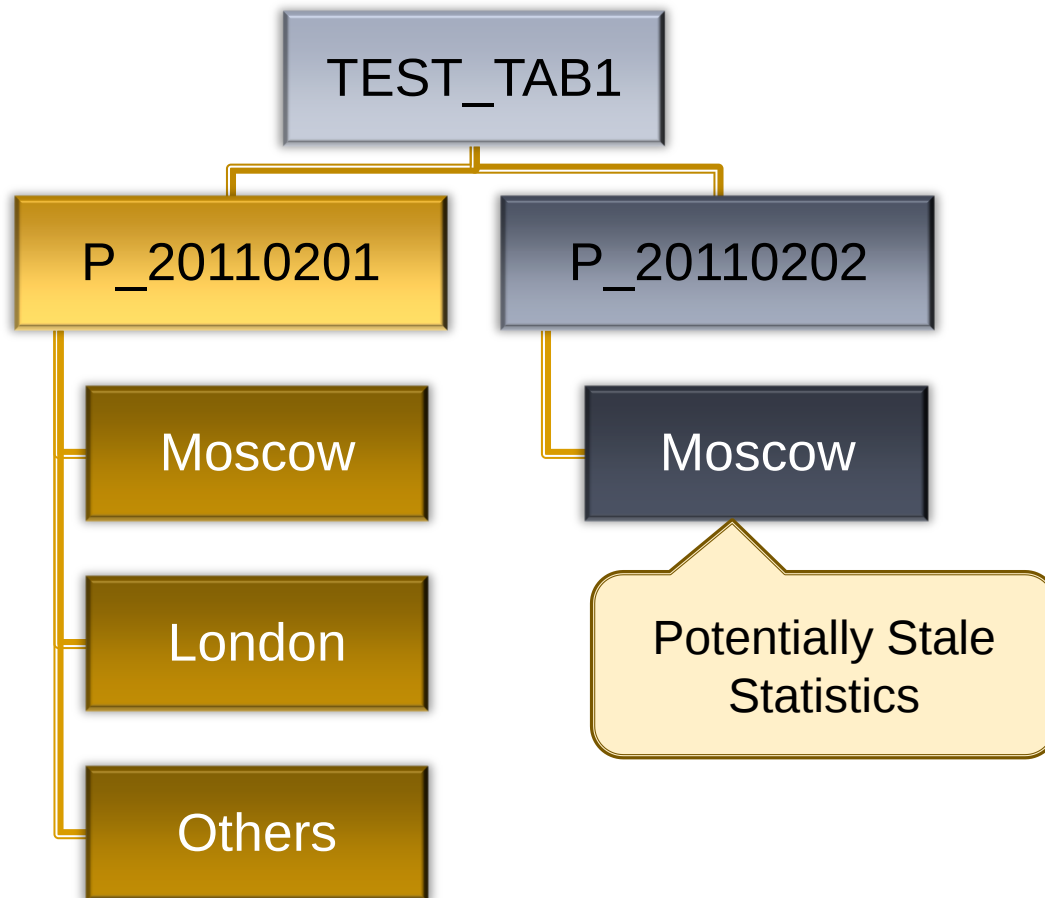
The Quality/Performance Trade-off

- Introduction
- Simple Fundamentals
- Statistics on Partitioned Objects
- The Quality/Performance Trade-off
- Aggregation Scenarios
- Alternative Strategies
- Incremental Statistics
- Conclusions and References

Collecting Global Statistics



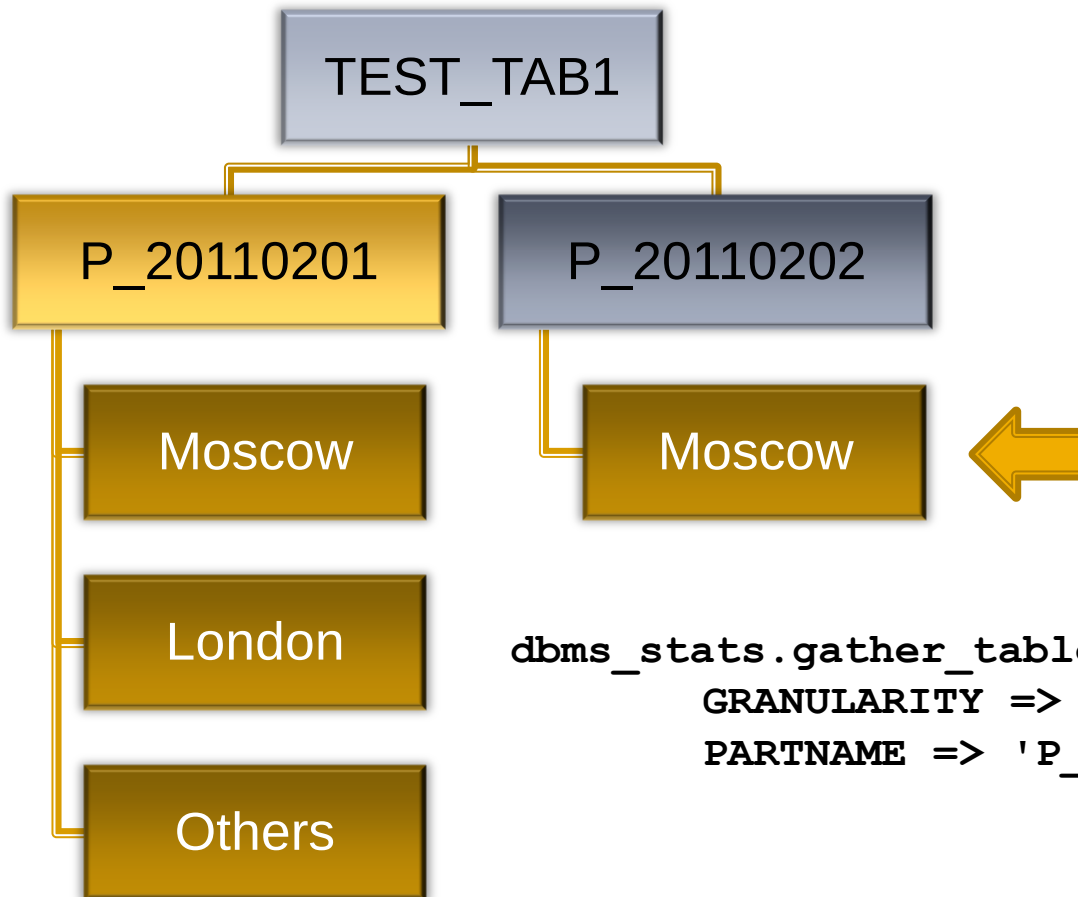
Collecting Global Statistics



GRANULARITY Parameter

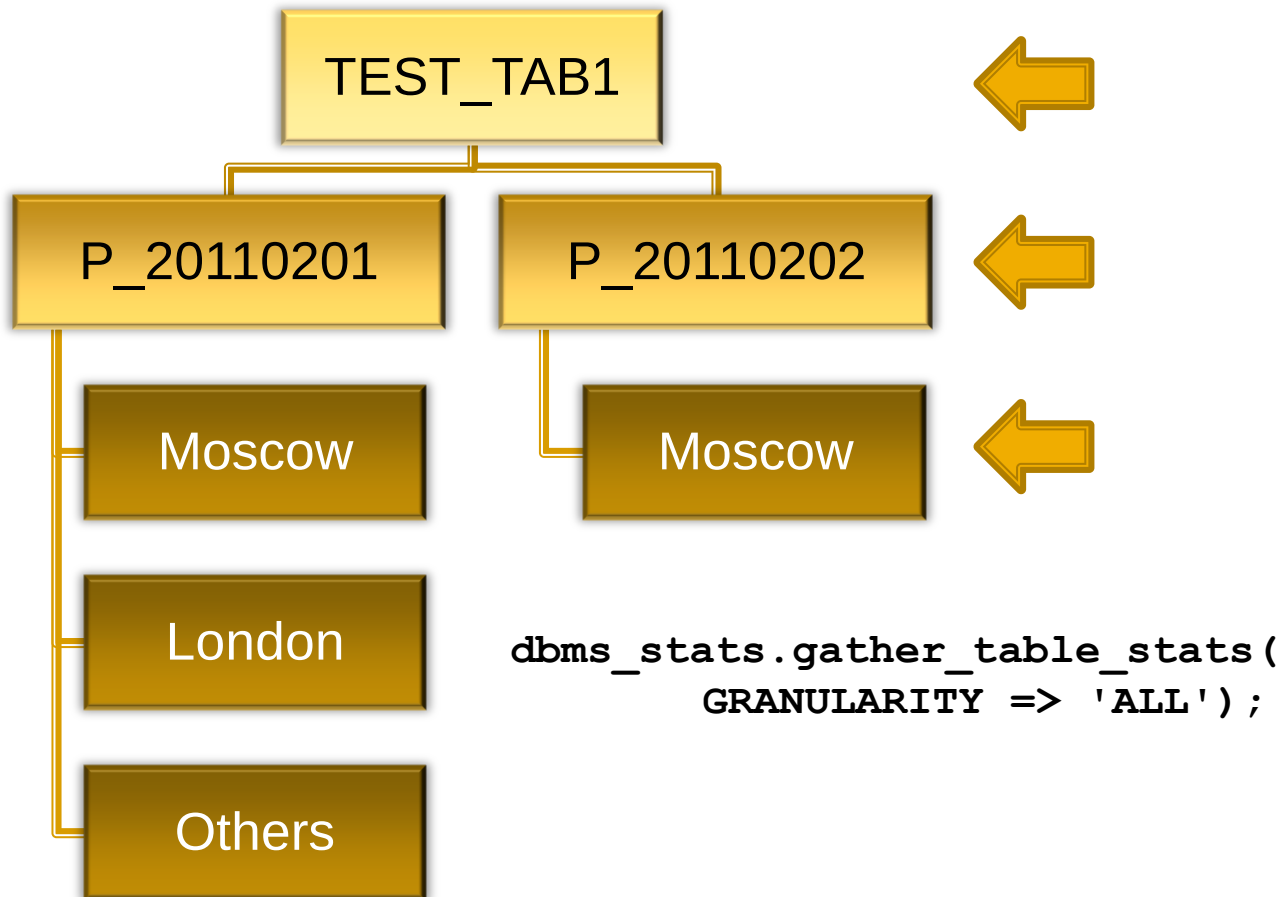
GRANULARITY	Statistics Gathered
ALL	Global, Partition and Subpartition
AUTO	Determines granularity based on partitioning type. This is the default
DEFAULT	Gathers global and partition-level stats. This option is deprecated, and while currently supported, it is included in the documentation for legacy reasons only. You should use 'GLOBAL AND PARTITION' for this functionality.
GLOBAL	Global
GLOBAL AND PARTITION	Global and Partition (but not subpartition) stats
PARTITION	Partition (specify PARTNAME for a specific partition. Default is all partitions.)
SUBPARTITION	Subpartition (specify PARTNAME for a specific subpartition. Default is all subpartitions.)

GRANULARITY => SUBPARTITION

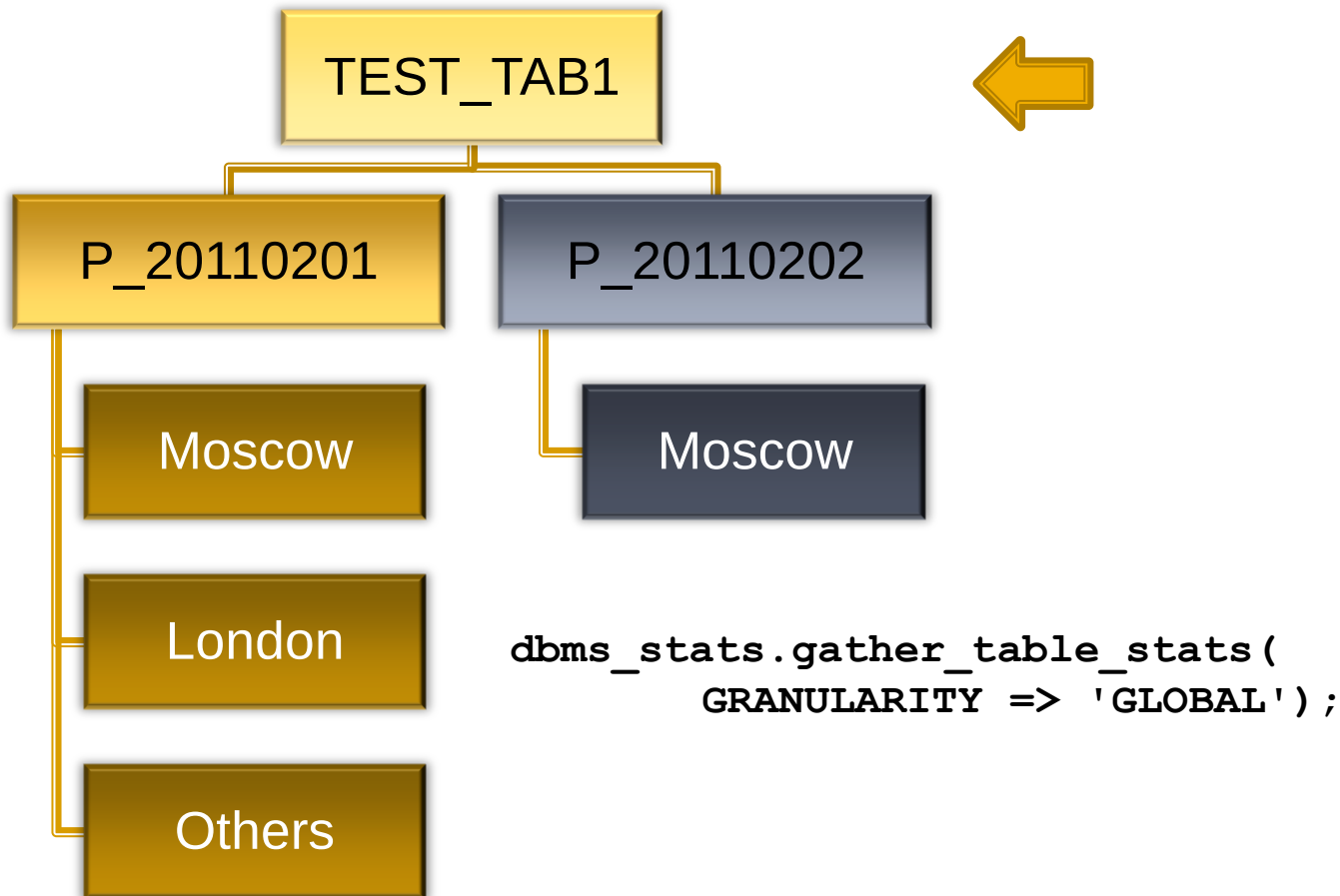


```
dbms_stats.gather_table_stats(  
    GRANULARITY => 'SUBPARTITION',  
    PARTNAME => 'P_20110202_MOSCOW') ;
```

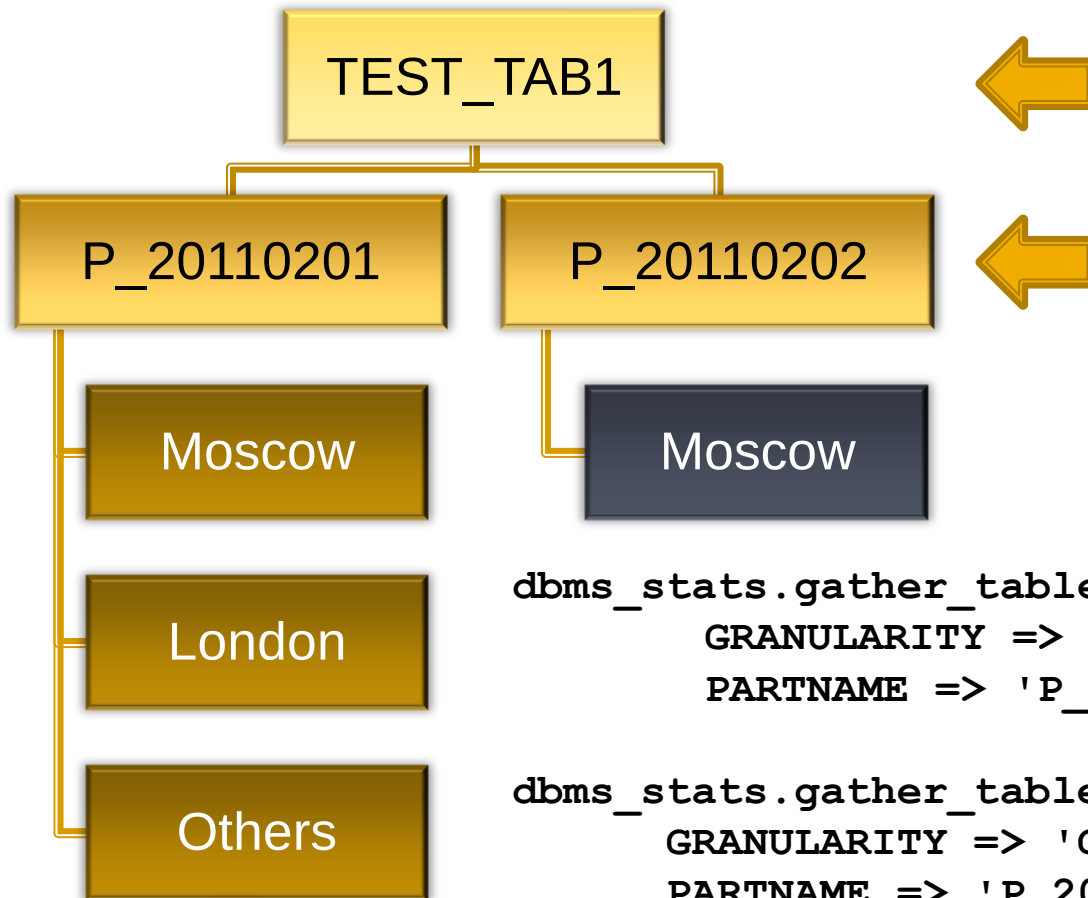
GRANULARITY => ALL



GRANULARITY => GLOBAL



GRANULARITY => DEFAULT



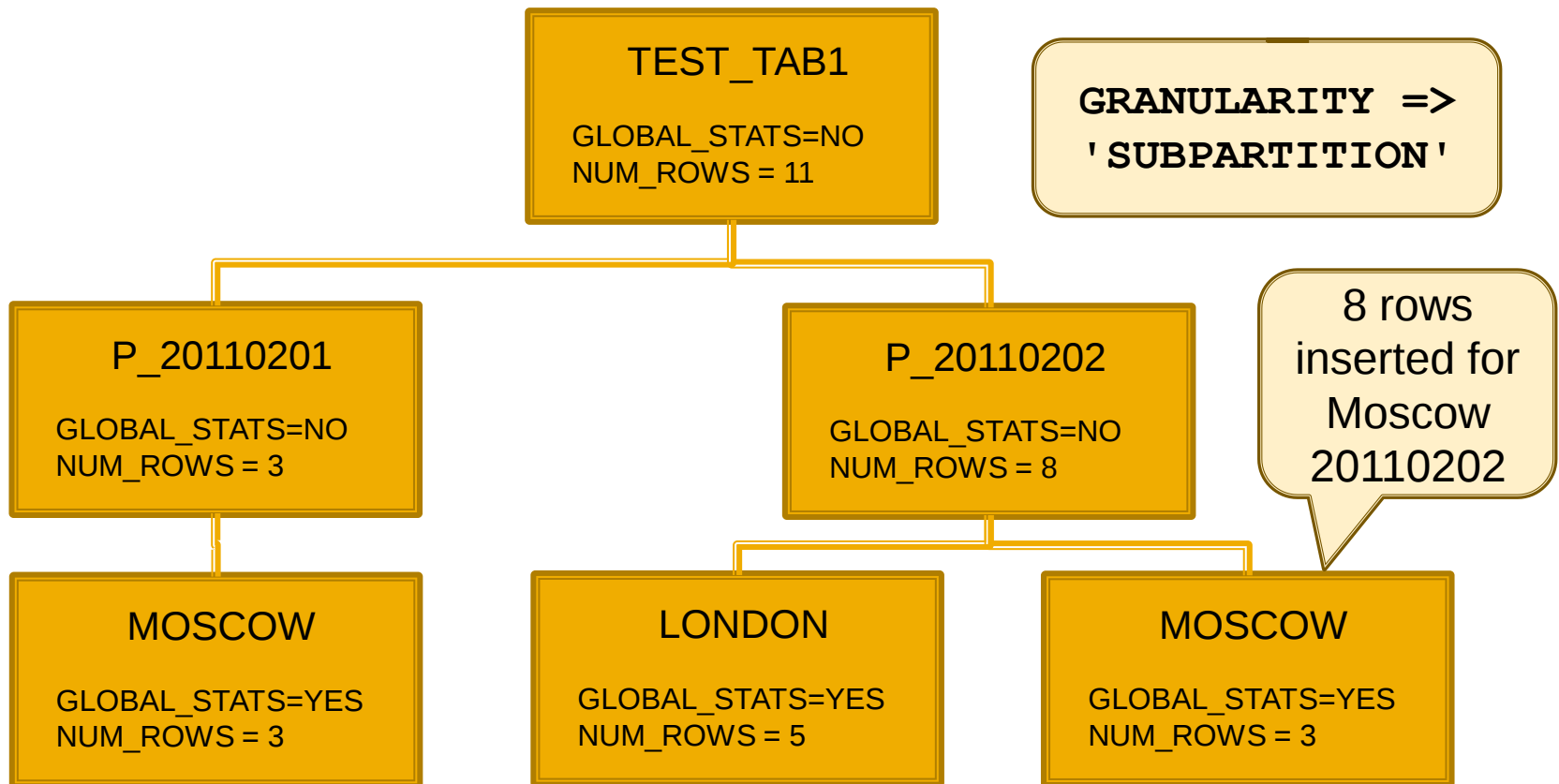
```
dbms_stats.gather_table_stats(  
    GRANULARITY => 'DEFAULT',  
    PARTNAME => 'P_20110202_MOSCOW');
```

```
dbms_stats.gather_table_stats(  
    GRANULARITY => 'GLOBAL AND PARTITION',  
    PARTNAME => 'P_20110202_MOSCOW');
```

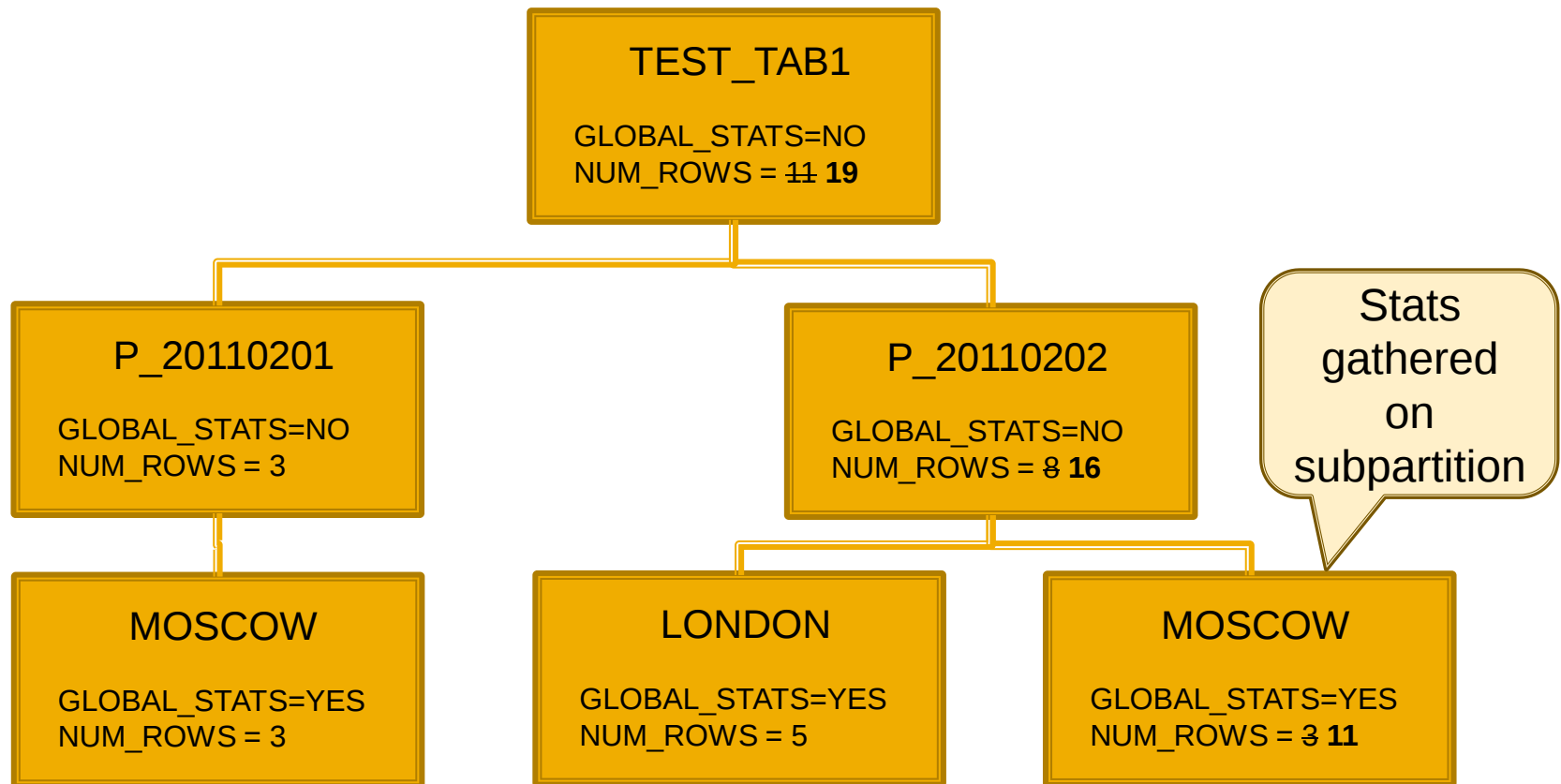
Aggregated Global Statistics

- To address the high cost of collecting Global Stats, Oracle provides another option – Aggregated or Approximate Global Stats
- Only gather stats on the lower levels of the object
 - Partition on partitioned tables
 - Subpartition on composite-partitioned tables
- DBMS_STATS will *aggregate* the underlying statistics to generate *approximate* global statistics at higher levels
- Important – **GLOBAL_STATS=NO**

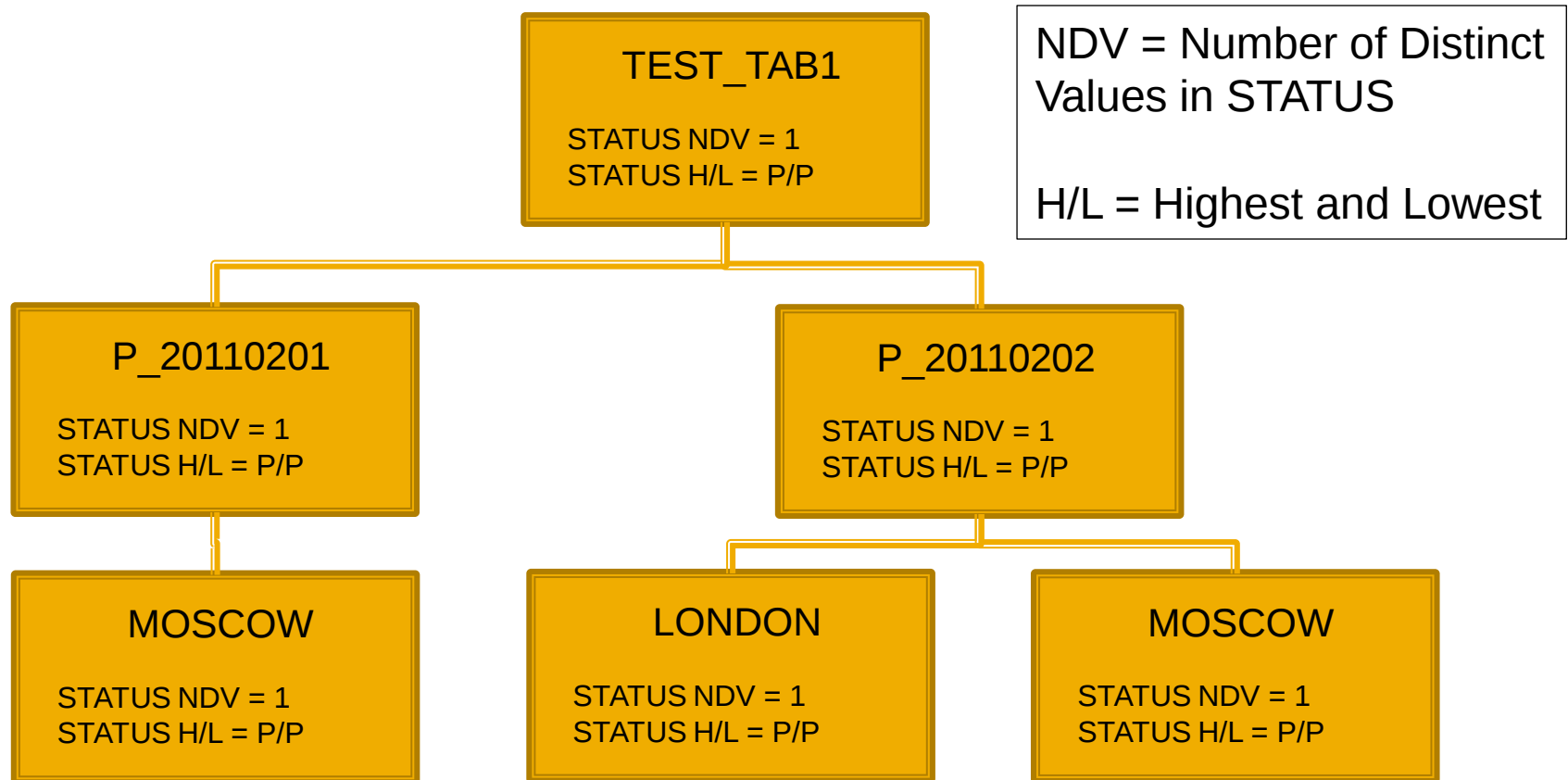
Aggregated Row Counts



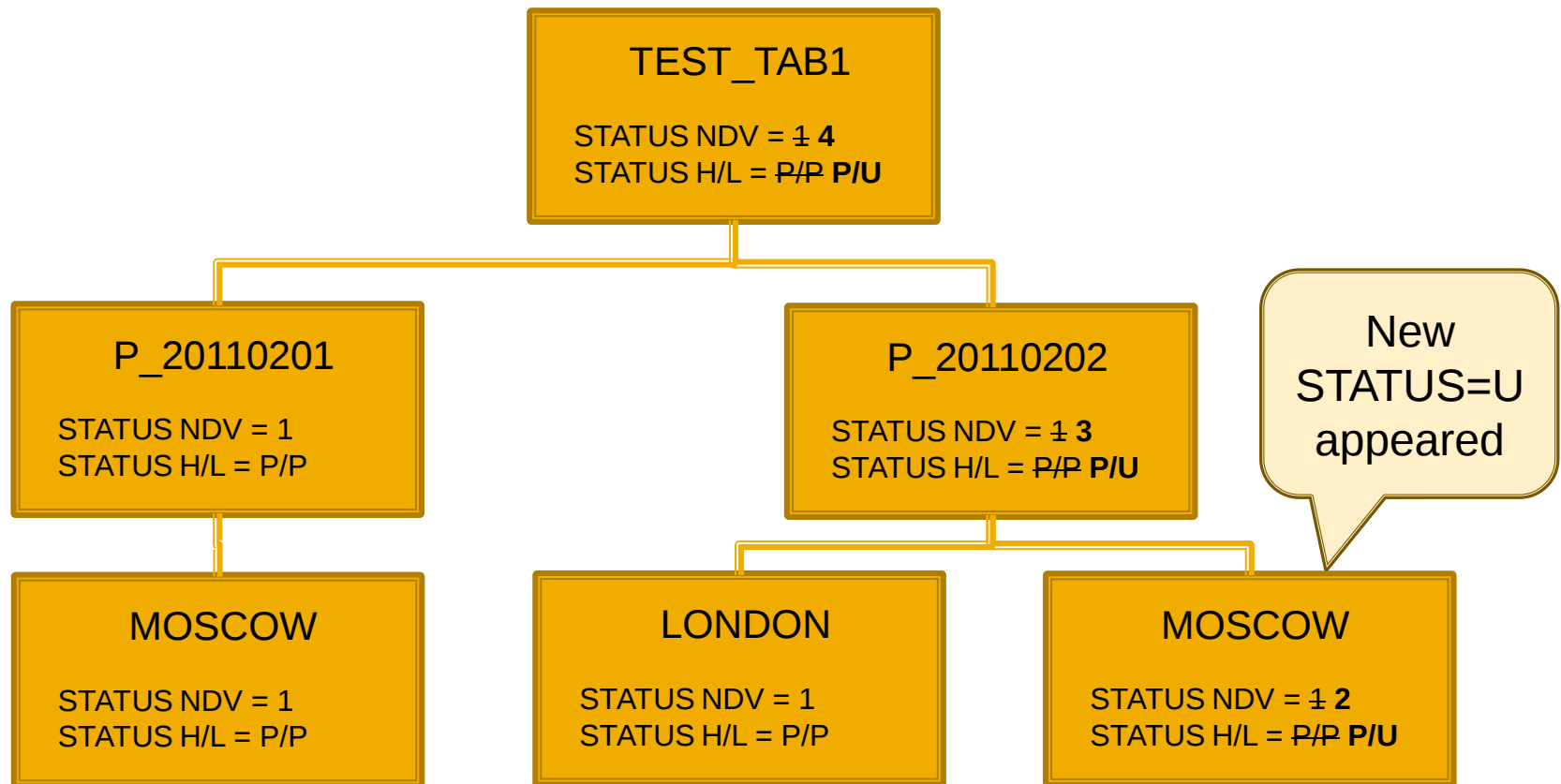
Aggregated Row Counts



Aggregated High/Low and NDVs



Aggregated High/Low and NDVs



Quality/Performance Trade-off

- You have a choice
- Gather True Global Stats
 - More accurate NDVs
 - Requires high-cost full table scan (which will get progressively slower and more expensive as tables grow)
 - Maybe an occasional activity?
- Gather True Partition Stats and Aggregated Global Stats
 - Accurate row counts and column High/Low values
 - Wildly inaccurate NDVs
 - Requires low-cost partition scan activity plus aggregation

Aggregation Scenarios

- Introduction
- Simple Fundamentals
- Statistics on Partitioned Objects
- The Quality/Performance Trade-off
- Aggregation Scenarios
- Alternative Strategies
- Incremental Statistics
- Conclusions and References

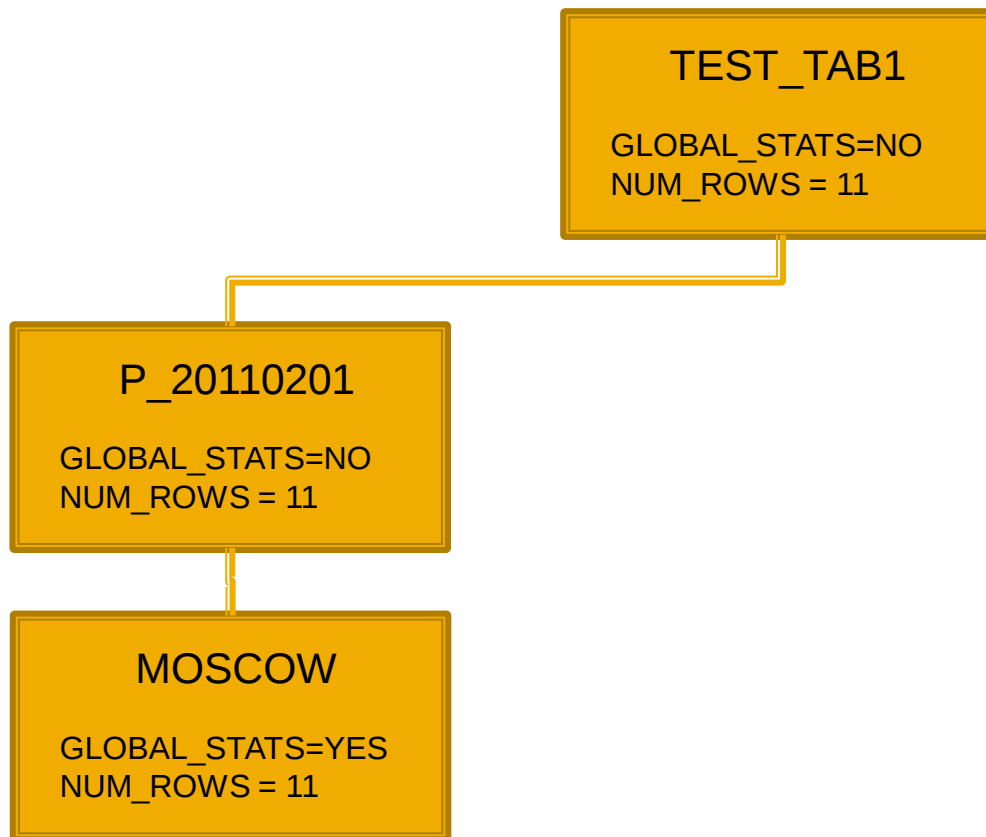
Aggregation Scenarios

- Take care if you decide to use Aggregated Global Stats
- Several implicit rules govern the aggregation process
- I have seen every issue I'm about to describe
 - In the past 18 months
 - Working on systems with people who are usually pretty smart

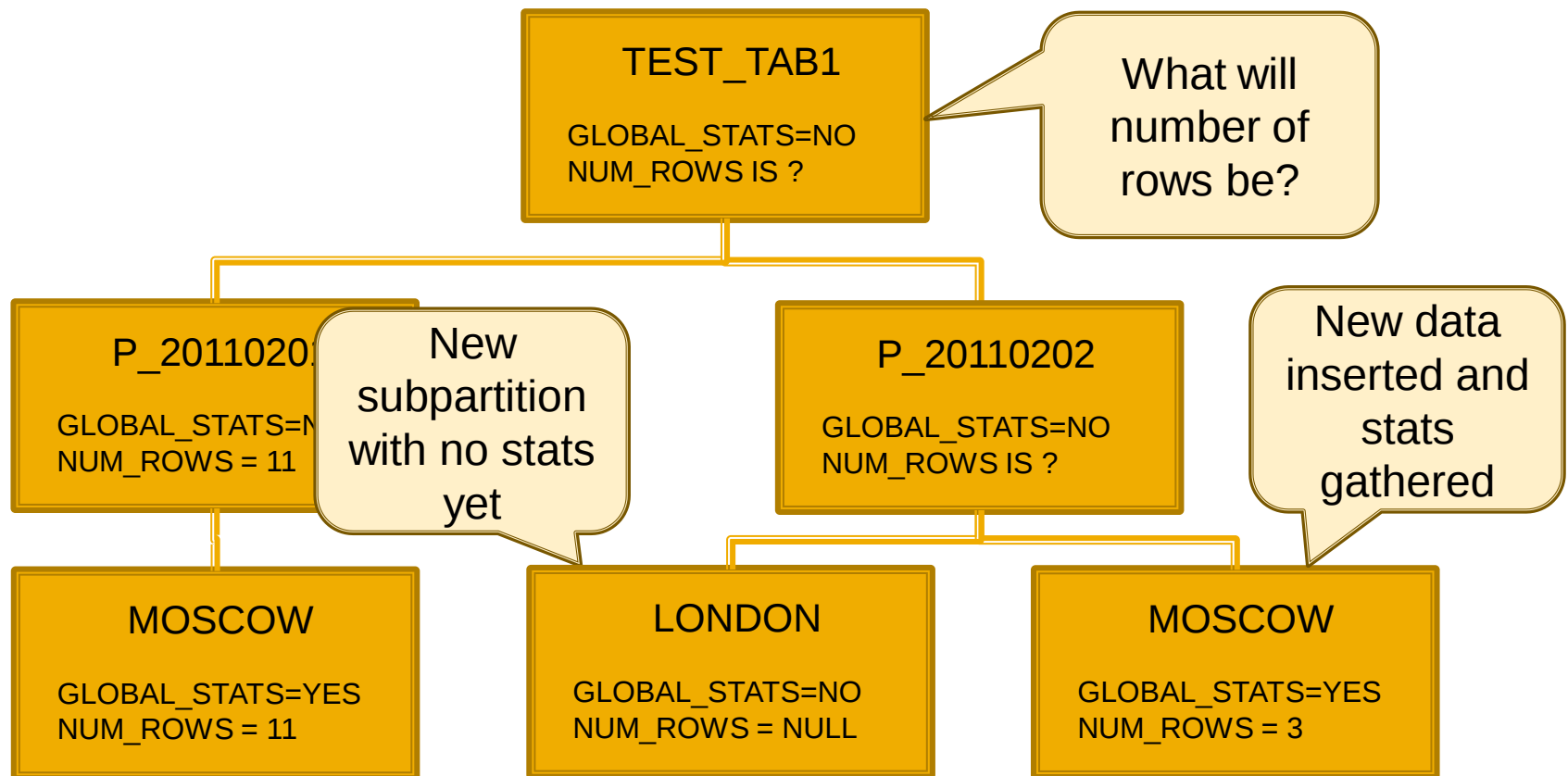
Missing Subpartition Stats

- Scenario 1
 - Aggregated Global Stats at Table-level
 - Subpartition Stats gathered at subpartition-level as part of new subpartition load process
 - Emergency hits when someone tries to INSERT data for which there is no valid subpartition
 - Solution – quickly add a new partition *and gather stats on new subpartition.*

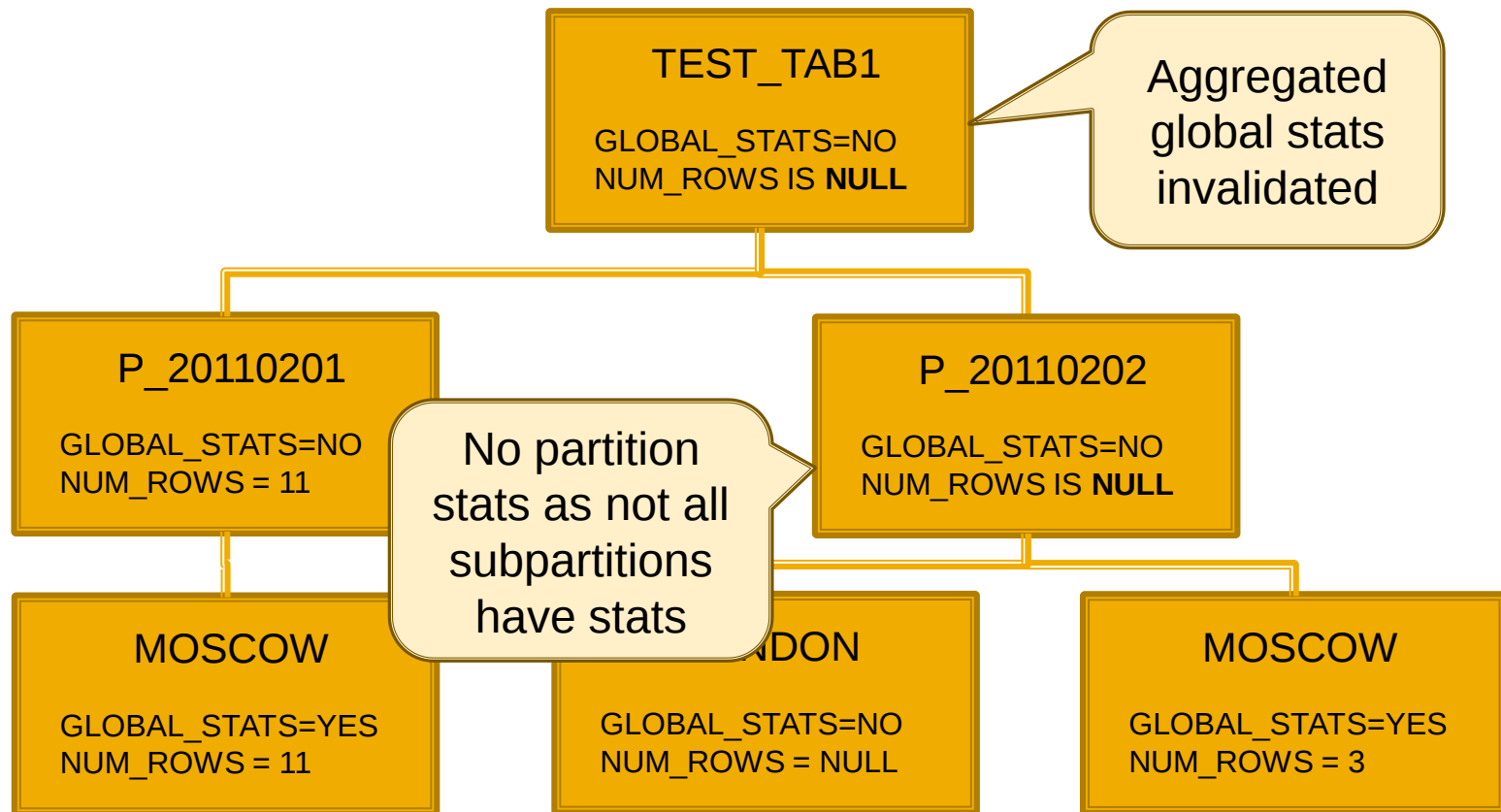
Missing Subpartition Stats



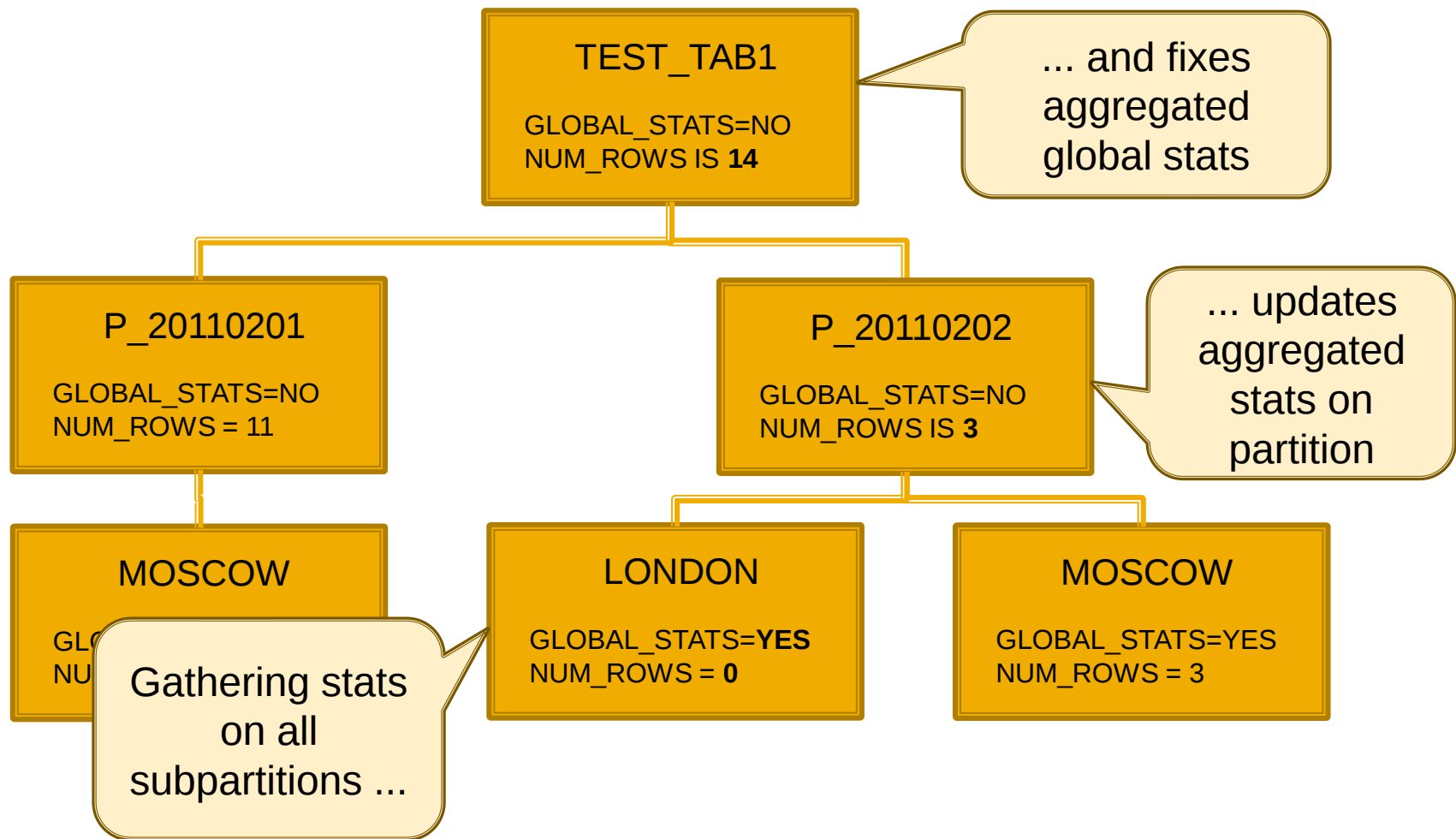
Missing Subpartition Stats



Missing Subpartition Stats



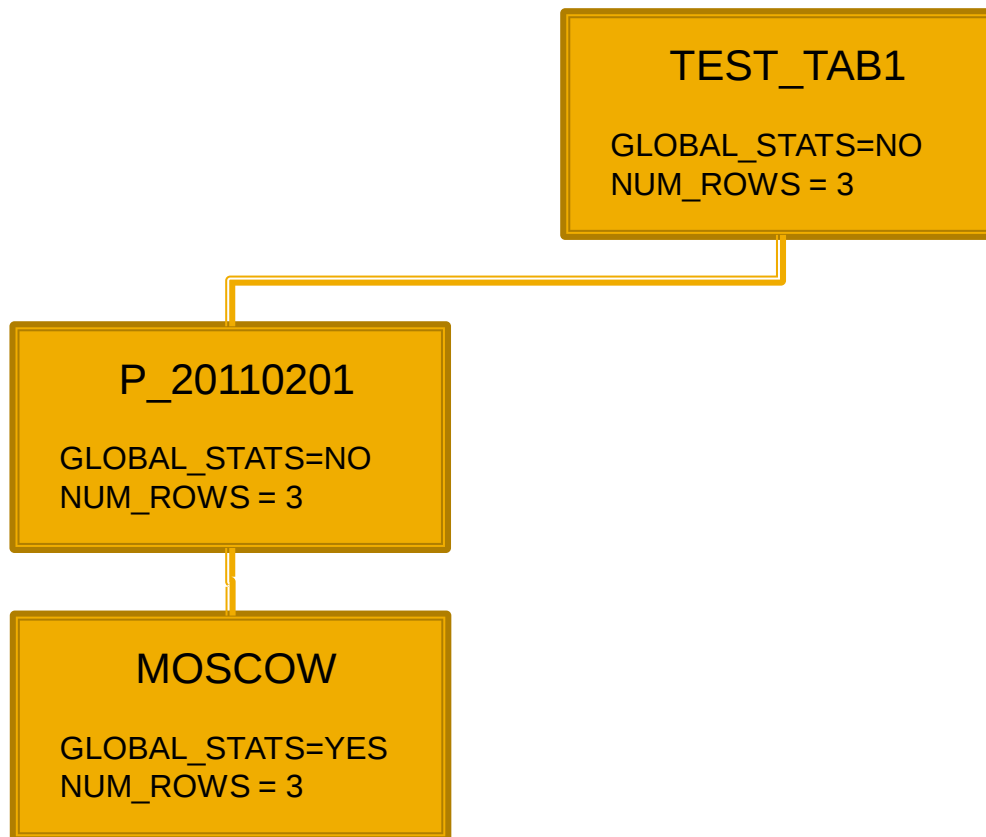
Missing Subpartition Stats



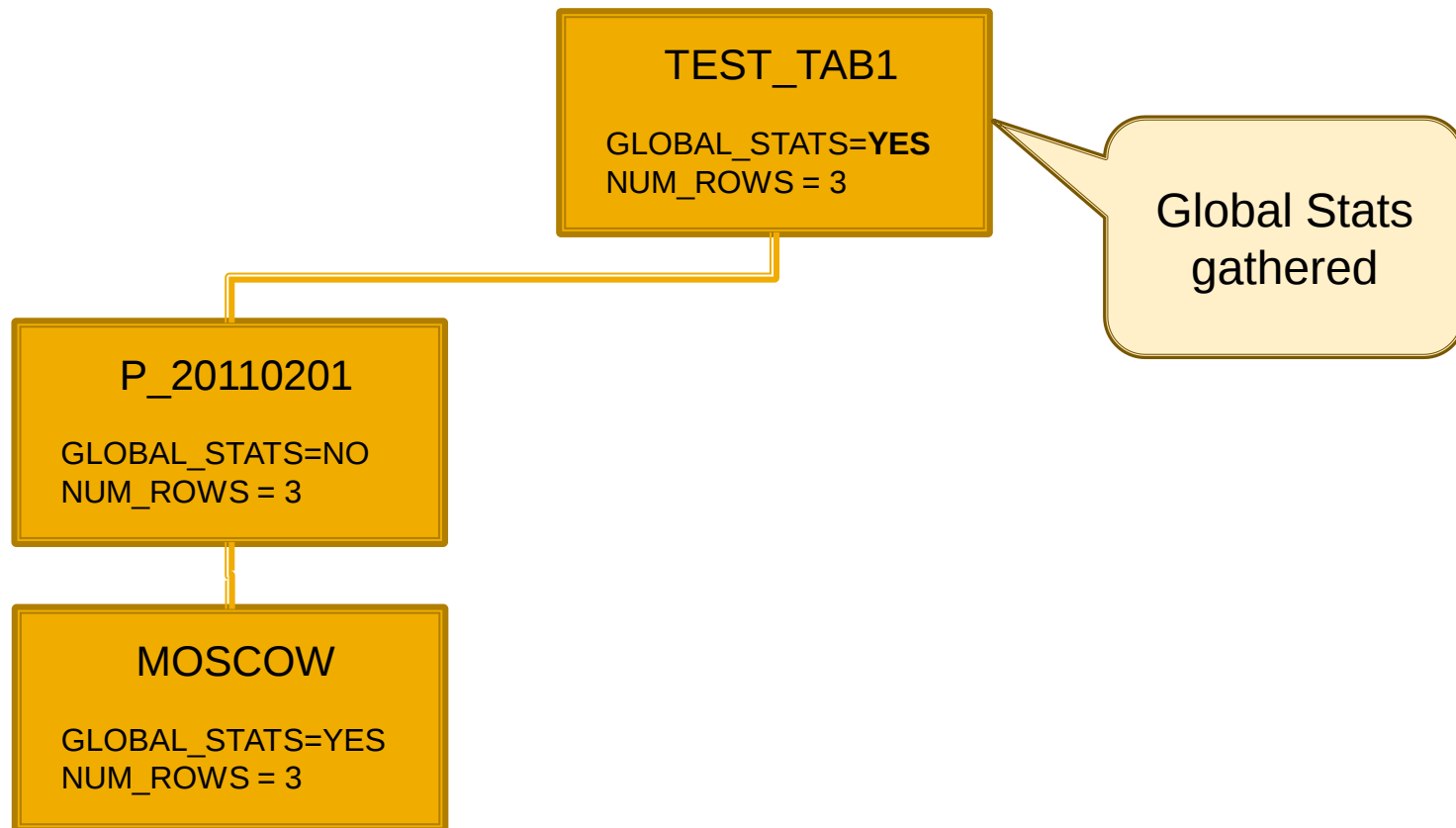
Incorrectly gathered Global Stats

- Scenario 2
 - Aggregated Global Stats at Table-level
 - Partition Stats gathered at Partition-level as part of new partition load process
 - Performance of several queries is horrible and poor NDVs at the Table-level are identified as root cause
 - Solution – Gather Global Stats quickly!

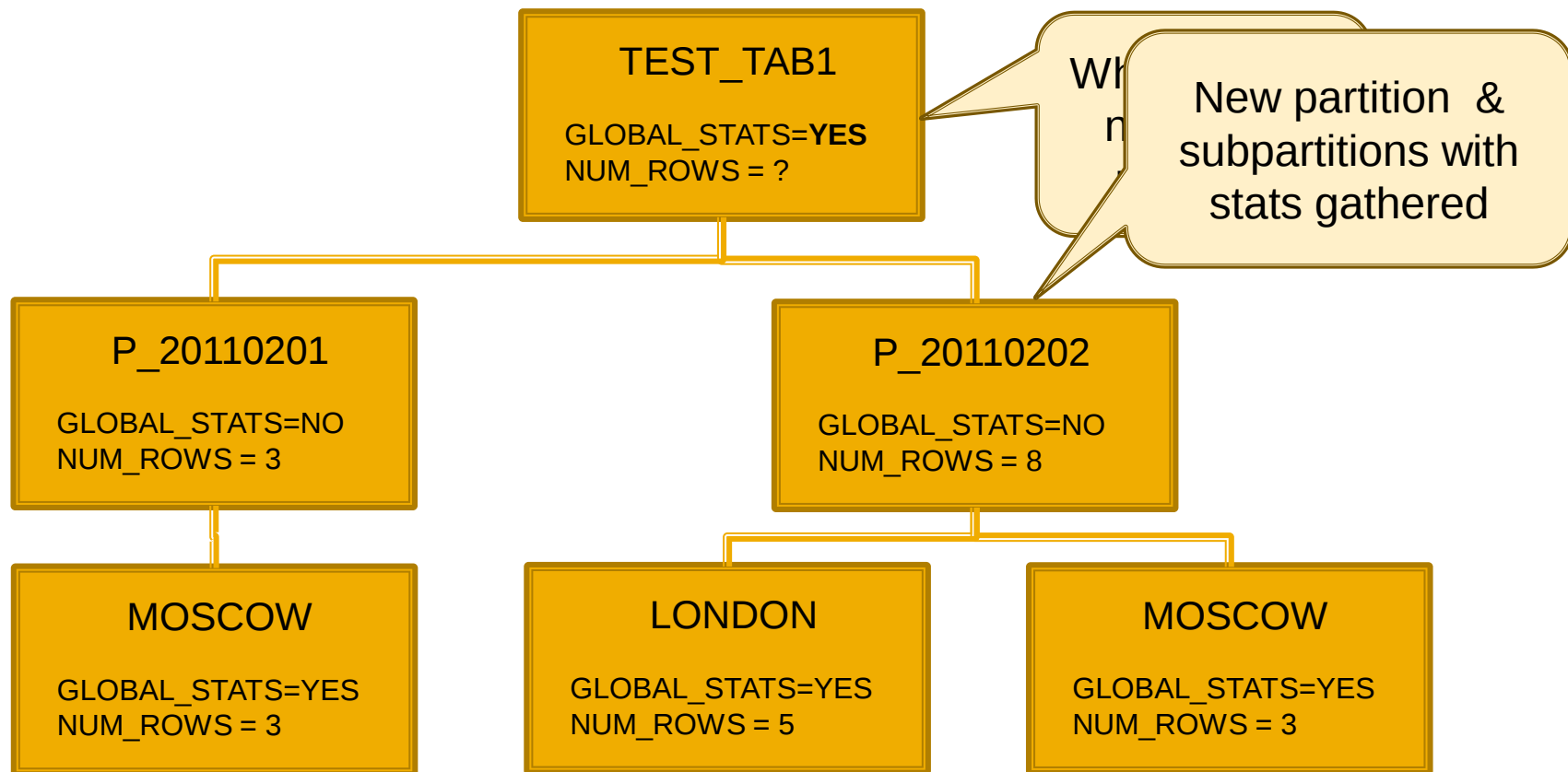
Incorrectly Gathered Global Stats



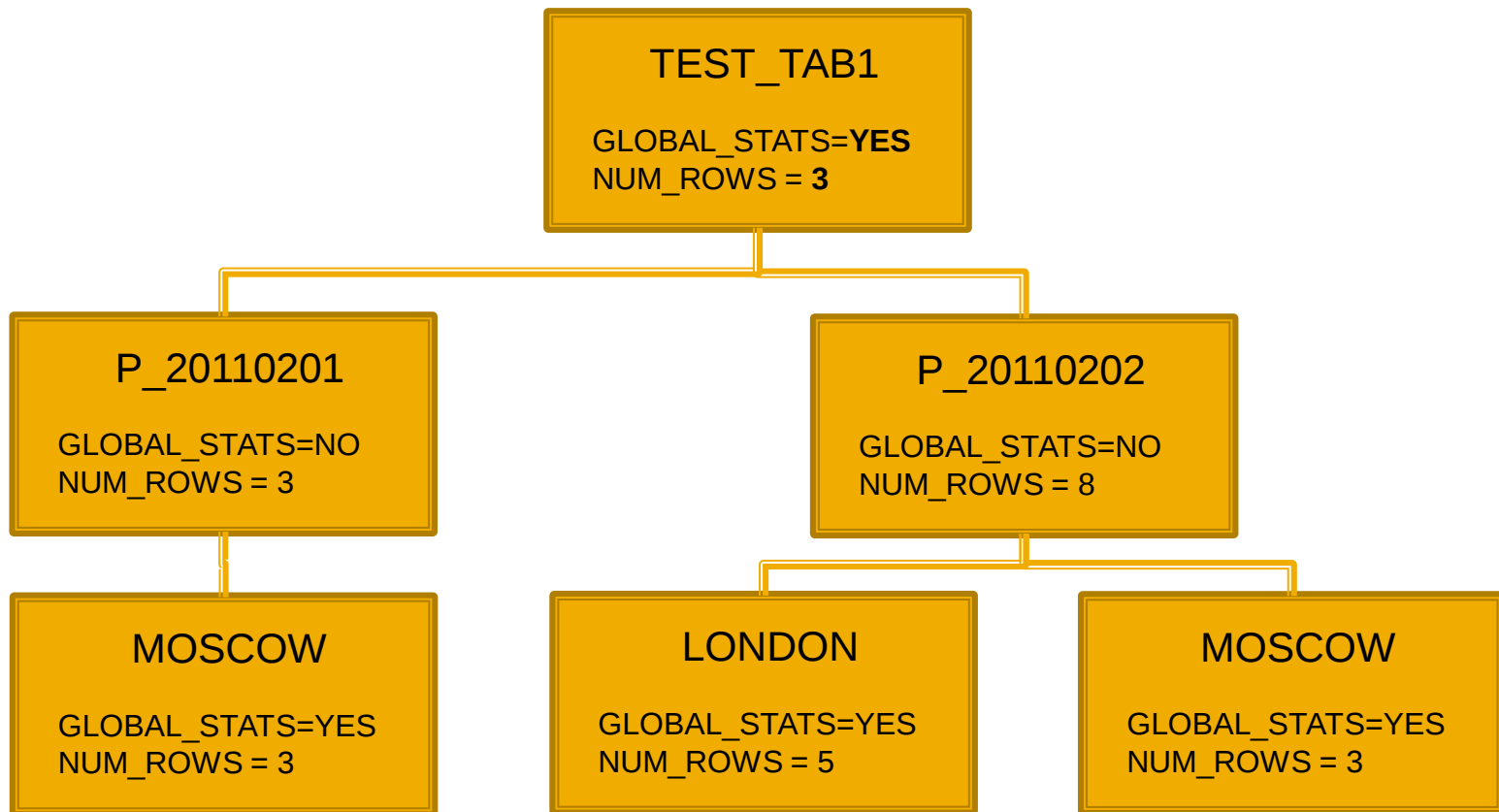
Incorrectly Gathered Global Stats



Incorrectly Gathered Global Stats



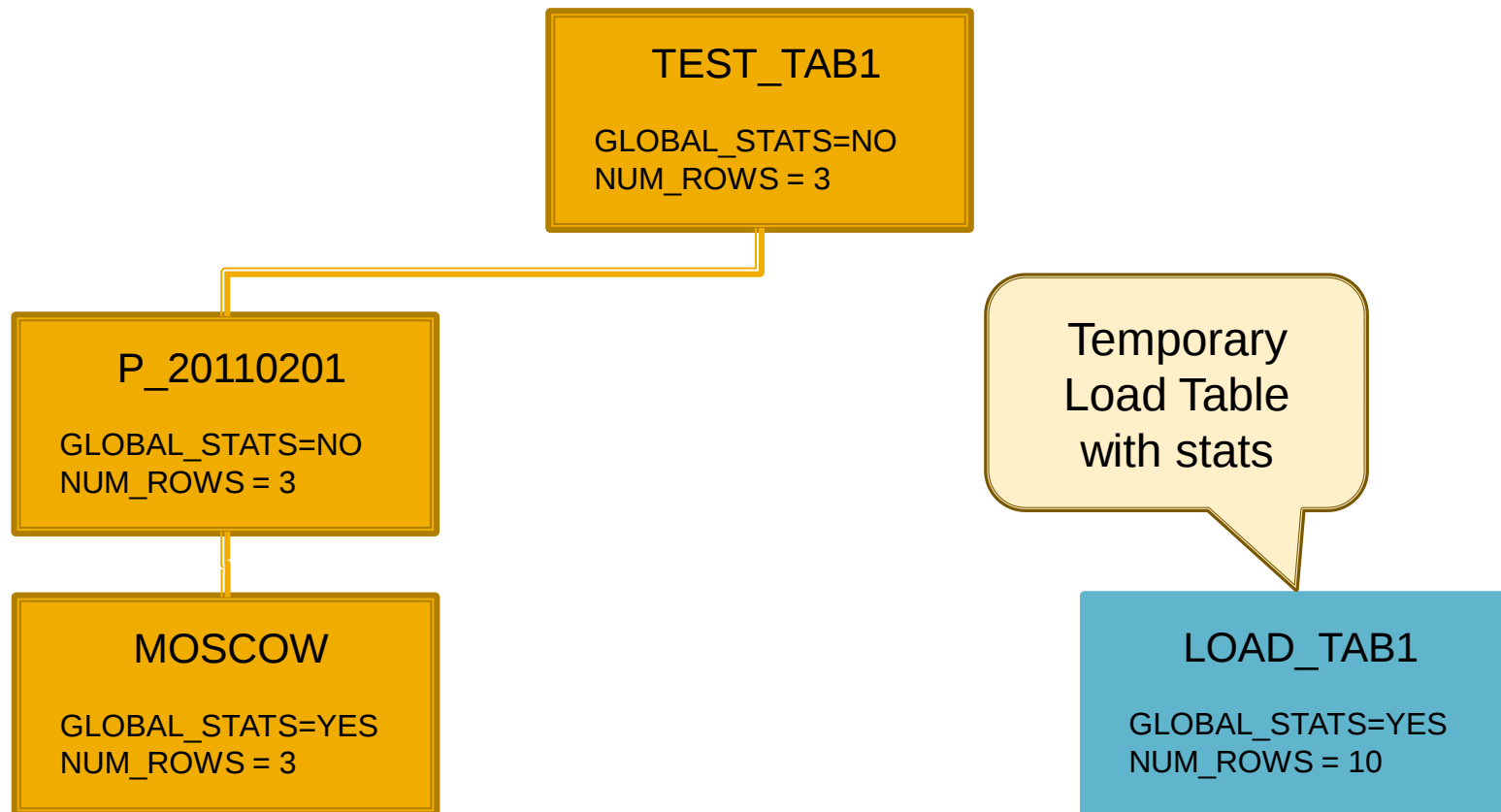
Incorrectly Gathered Global Stats



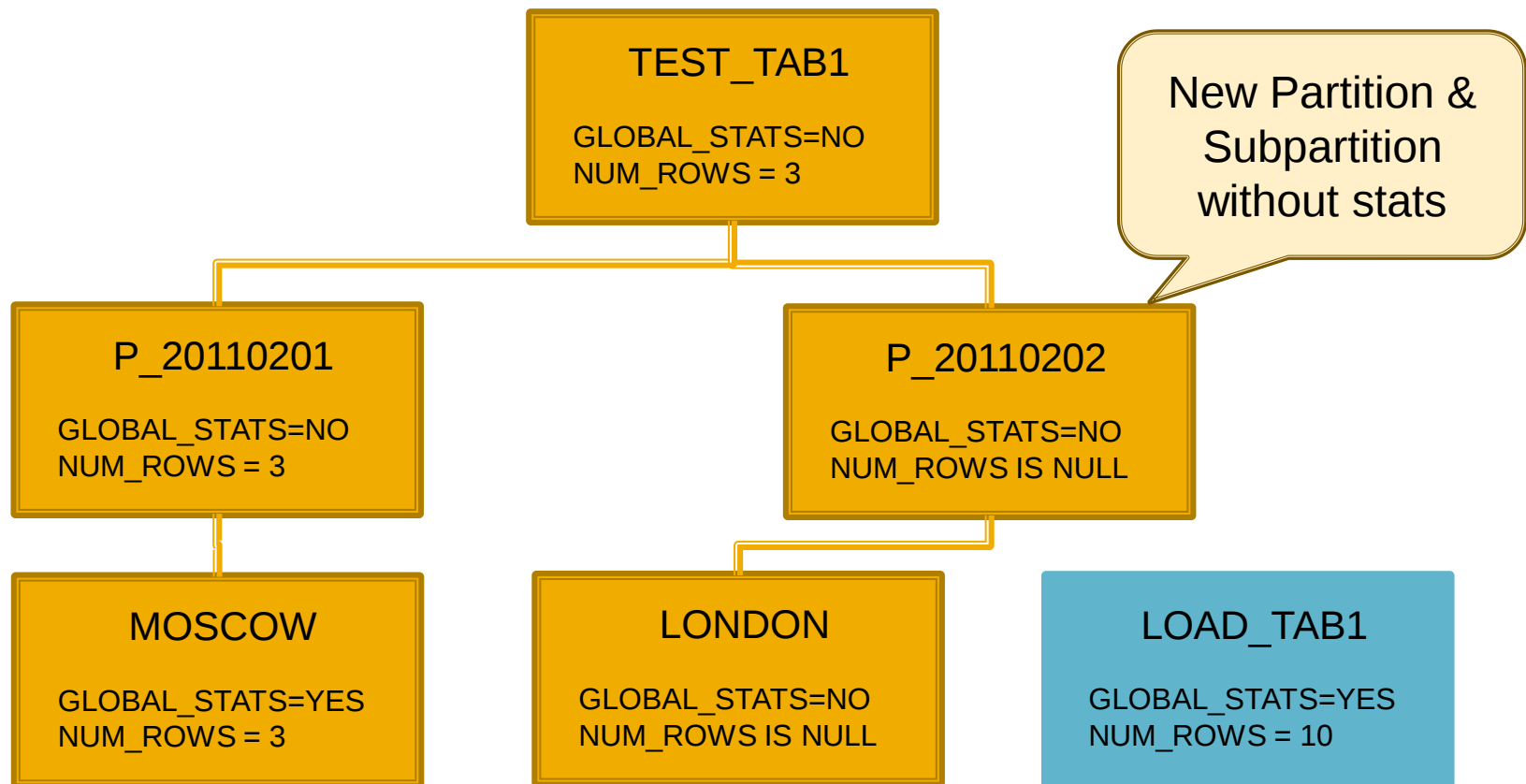
Partition Exchange Issues

- Scenario 3
 - Aggregated Global Stats at Table-level
 - Statistics are gathered on temporary Load Table
 - Load Table is exchanged with partition of target table
 - Objective is to minimise activity on target table and ensure that stats are available on partition immediately on exchange

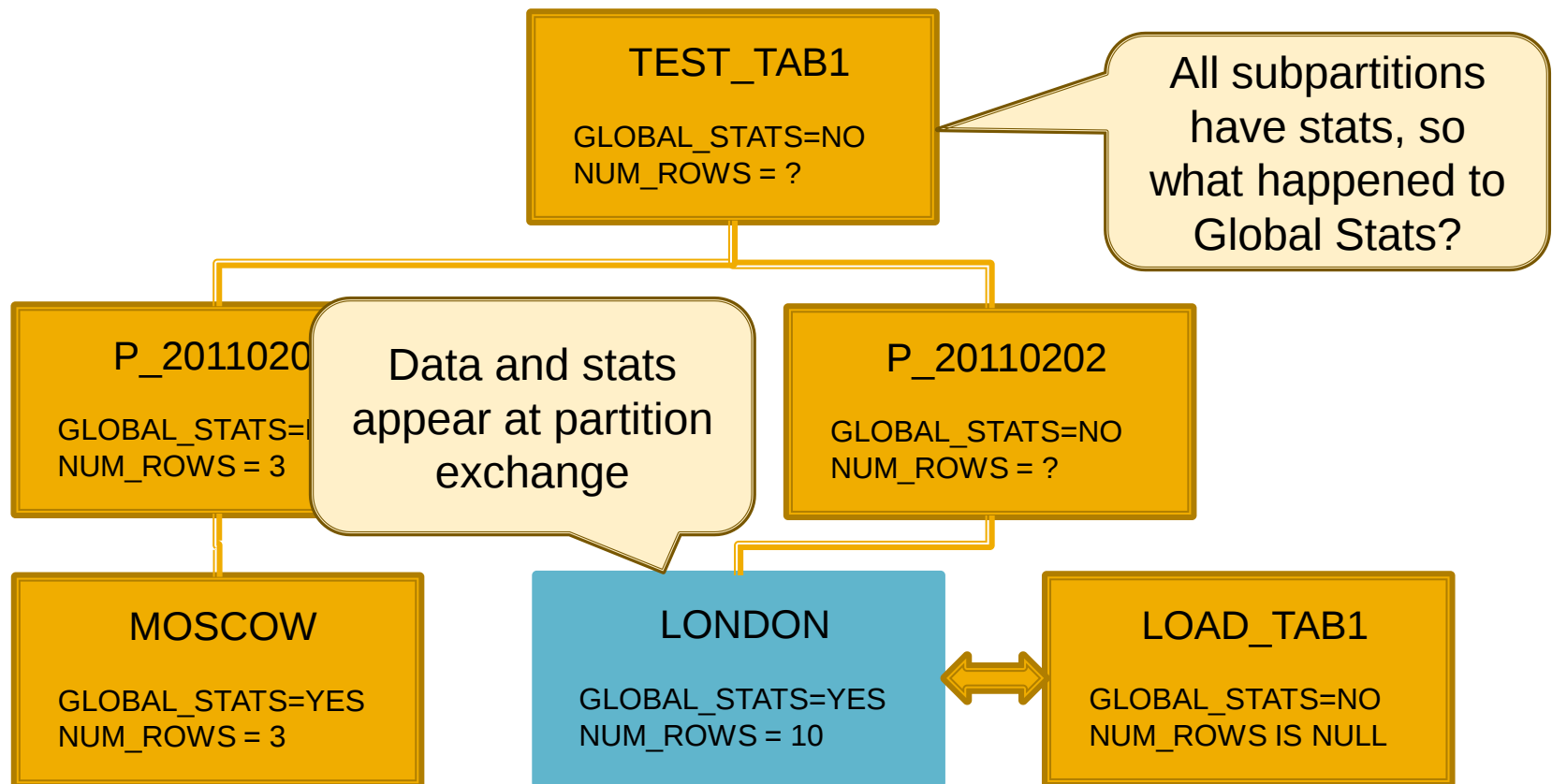
Gather-then-Exchange



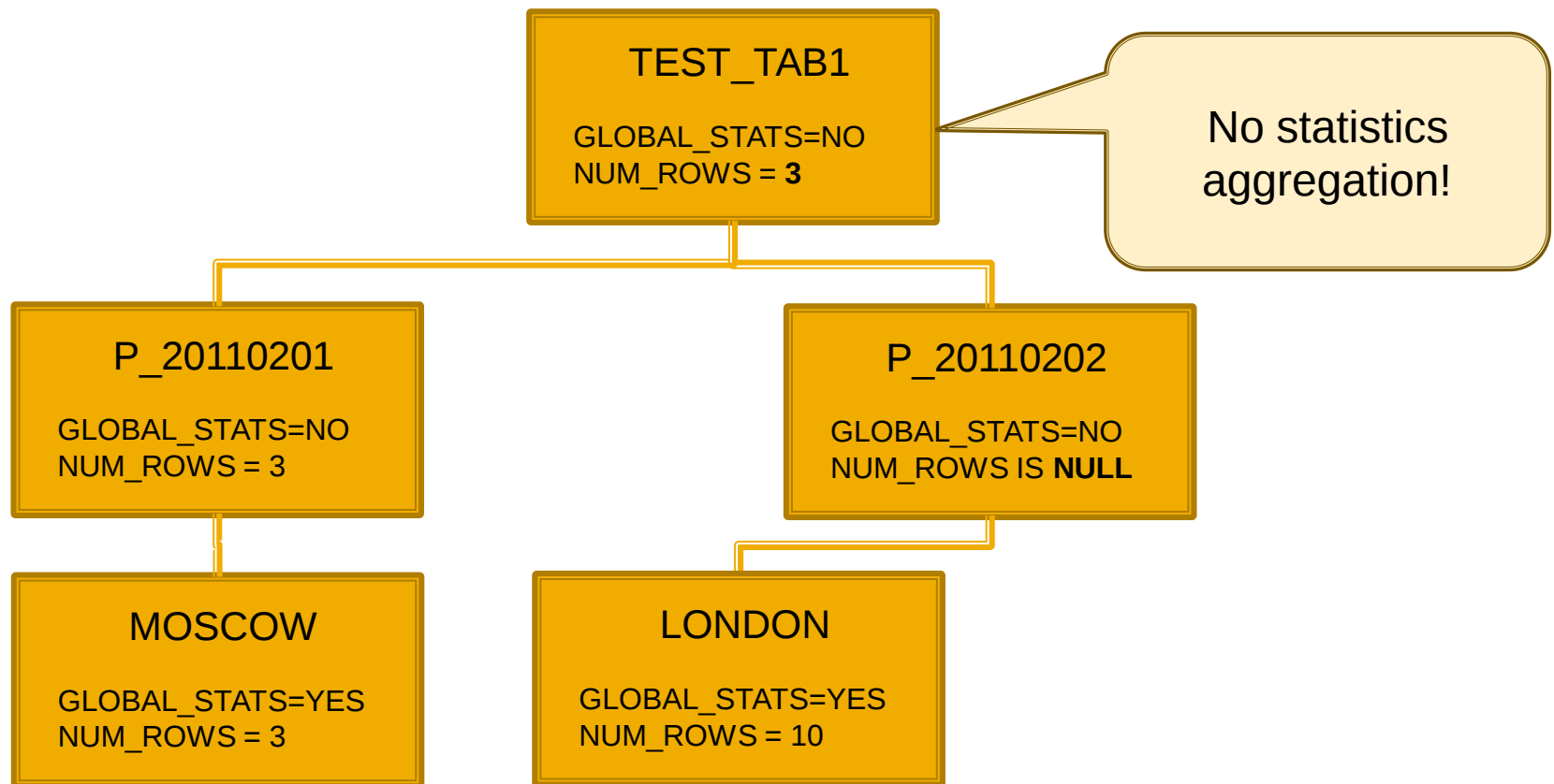
Gather-then-Exchange



Gather-then-Exchange



Gather-then-Exchange



_minimal_stats_aggregation

- Hidden parameter used to minimise the impact of statistics aggregation process
- Default is TRUE which means *minimise* aggregation
- Partition exchange will not trigger the aggregation process!
- Solutions
 - Change hidden parameter – speak to Support
 - Exchange-then-Gather (another good reason for this later)

Aggregated Stats – Summary

- Wildly inaccurate NDVs which *will* impact Execution Plans
- Take care with the aggregation process
- Do not use aggregated statistics unless you really don't have time to gather true Global Stats
- But the problem is, what if your table is so damn big that you can never manage to update those Global Stats?

Alternative Strategies

- Introduction
- Simple Fundamentals
- Statistics on Partitioned Objects
- The Quality/Performance Trade-off
- Aggregation Scenarios
- **Alternative Strategies**
- Incremental Statistics
- Conclusions and References

Dynamic Sampling

- If stats collection is such a nightmare, perhaps we shouldn't bother gathering stats at all?
- Dynamic Sampling could be used
 - Gather no stats manually
 - When statements are parsed, Oracle will execute queries against objects to generate temporary stats on-the-fly
- I would not recommend this as a system-wide strategy
 - What happened when stats were missing in earlier examples!
 - Recurring overhead for every query
 - Either expensive or low quality stats

Setting Statistics

- Gathering stats takes time and resources
- The resulting stats describe your data to help the CBO determine optimal execution plans
- If you know your data well enough to know the appropriate stats, why not just set them manually and avoid the collection overhead?
 - Plenty of appropriate DBMS_STATS procedures
- Not a new idea and discussed in several places on the net (including JL chapter in latest Oak Table book)

Setting Statistics - Summary

■ Positives

- Very fast and low resource method for setting statistics on new partitions
- Potential improvements to plan stability when accessing time-period partitions that are filled over time

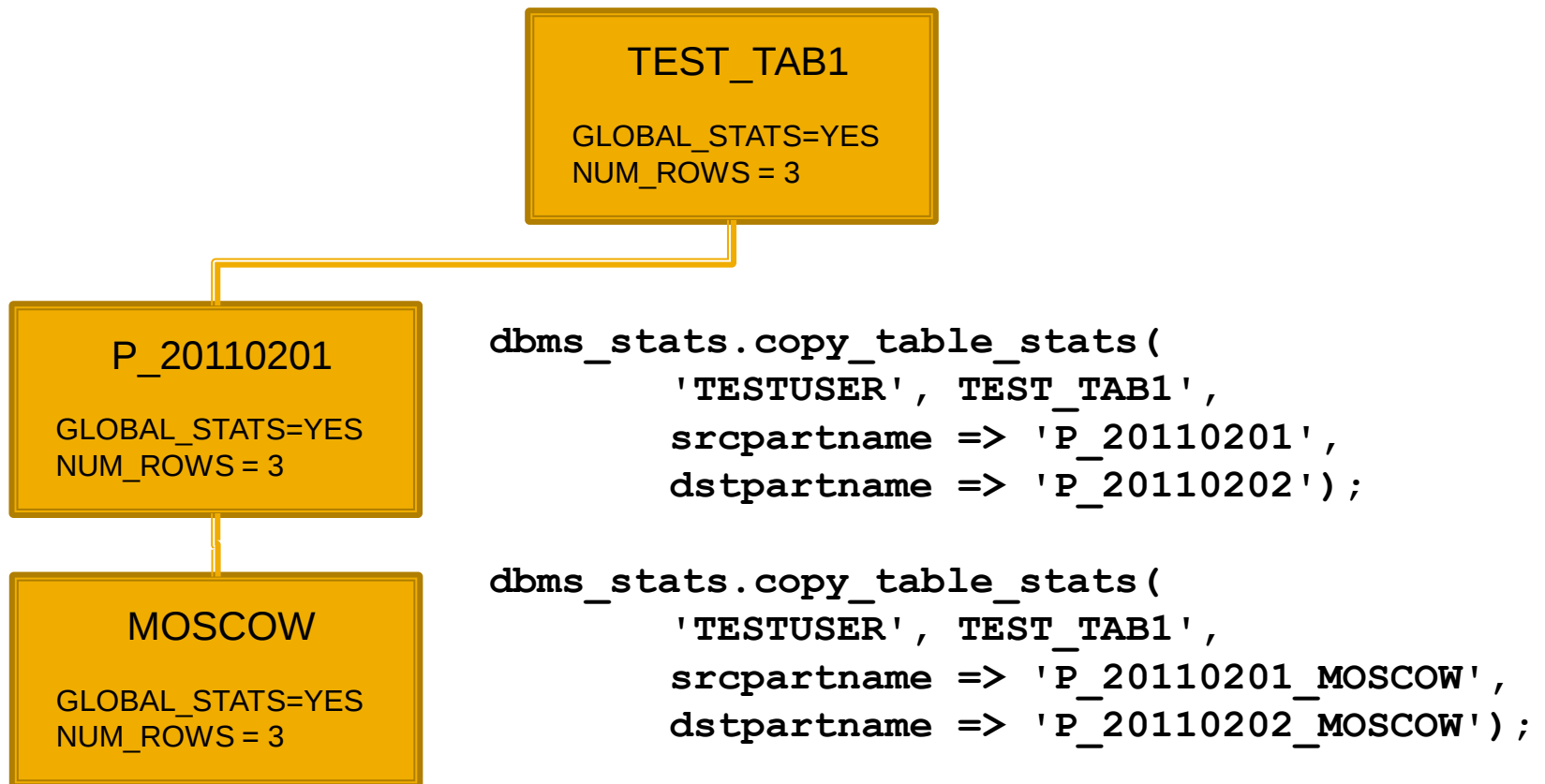
■ Negatives

- You need to know your data well, particularly any time periodicity
- You need to develop your own code implementation
- You could undermine the CBO's ability to use more appropriate execution plans as data changes over time
- Does not eliminate the difficulty in maintaining accurate Global Statistics, although these could be set manually too

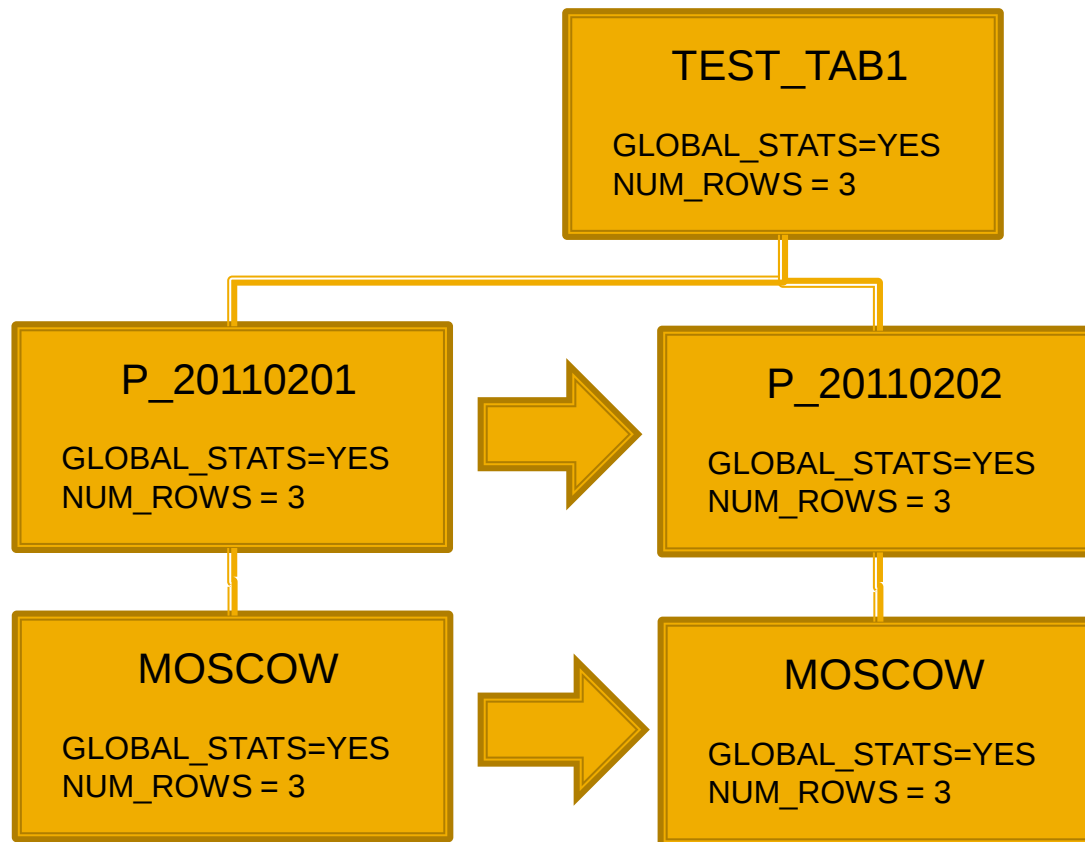
Copying Statistics

- Extending the concept of setting statistics manually
- Instead of trying to work out what the appropriate statistics are for a new partition, copy the statistics from another partition
 - The previous partition – increasing volumes?
 - A golden template partition – plan stability?
 - A prior partition to reflect the periodicity of your data. The second Tuesday from last month, Tuesday from last week, the 8th of last month
- *Supported* from 10.2.0.4

Copying Statistics



Copy Statistics



Copying Statistics – Bug 1

- The previous example doesn't work on an unpatched 10.2.0.4
- When copying stats between partitions on a composite partitioned object (one with subpartitions)

```
SQL> exec dbms_stats.copy_table_stats(ownname => 'TESTUSER',  
    tabname => 'TEST_TAB1', srcpartname => 'P_20110201',  
    dstpartname => 'P_20110202');  
BEGIN dbms_stats.copy_table_stats(ownname => 'TESTUSER',  
    tabname => 'TEST_TAB1', srcpartname => 'P_20110201',  
    dstpartname => 'P_20110202'); END;
```

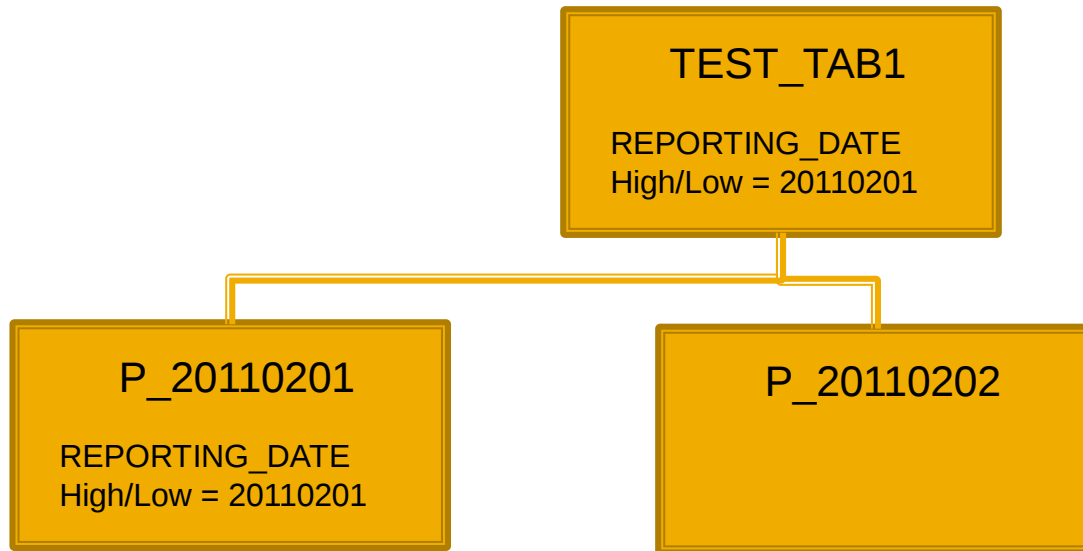
*

```
ERROR at line 1:  
ORA-06533: Subscript beyond count  
ORA-06512: at "SYS.DBMS_STATS", line 17408  
ORA-06512: at line 1
```

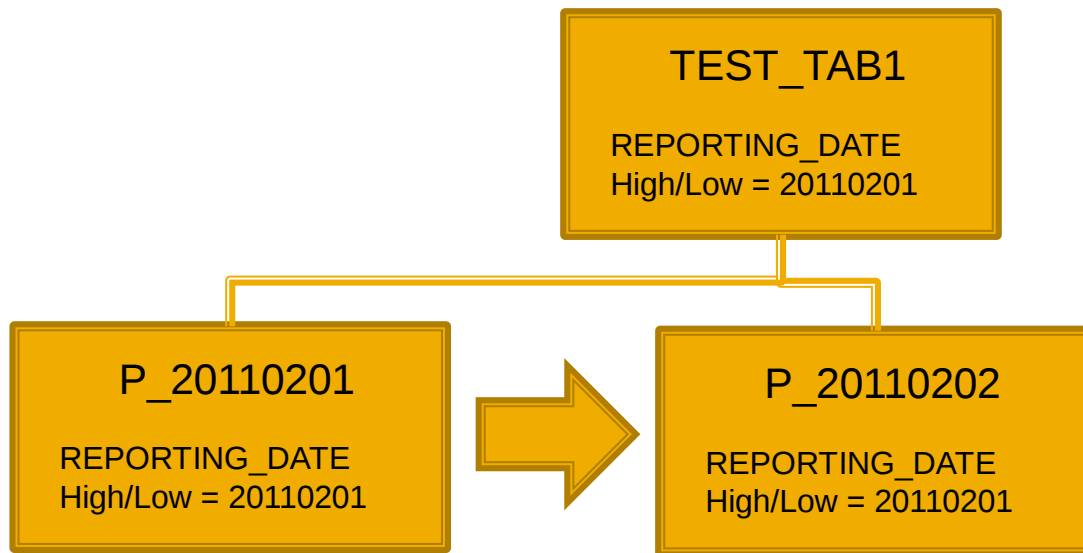
Copying Statistics – Bug 1

- Bug number 8318020
- Merge Label Request 8866627
 - Fixes a variety of stats-related bugs
- Patchset 10.2.0.5
- Upgrade to 11.2.0.2

Copying Statistics – Bug 2



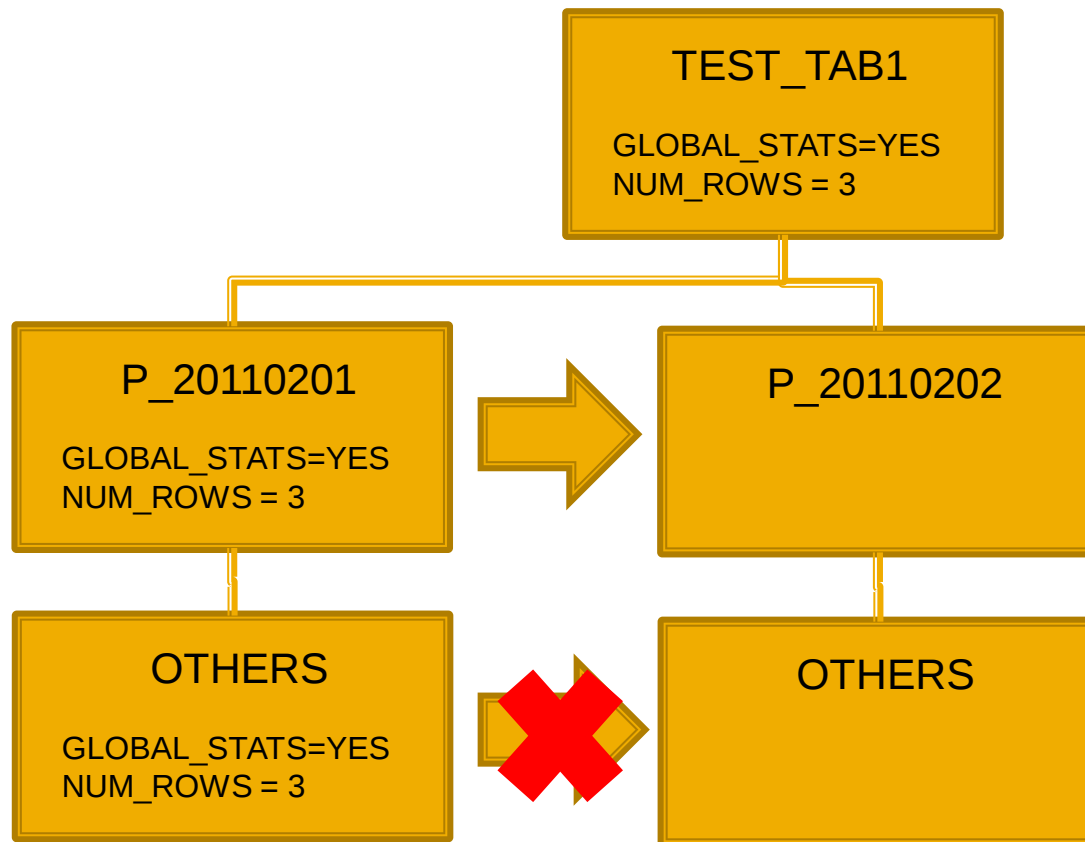
Copying Statistics – Bug 2



Copying Statistics – Bug 2

- We might reasonably expect Oracle to understand the *implicit* High/Low values of a partition key
- Merge Label Request 8866627
- Patchset 10.2.0.5
- Upgrade to 11.2
- The wider issue here is that High/Low values (other than Partition Key columns and NDVs) will simply be copied
 - Are you *sure* that's what you want?

Copying Statistics – Bug 3



Copying Statistics

- ORA-03113 / 07445 while copying list partition statistics
 - Core dump in `qospMinMaxPartCol`
- I initially thought this was because the OTHERS subpartition was the last one I copied stats for
- It is because it is a DEFAULT list subpartition
- Bug number 10268597
 - Still in 10.2.0.5 and 11.2.0.2
 - Marked as fixed in 11.2.0.3 and 12.1.0.0

Copying Statistics - Summary

■ Positives

- Very fast and low resource method for setting statistics on new partitions
- Potential improvements to plan stability when accessing time-period partitions that are filled over time

■ Negatives

- Bugs and related patches although better using 10.2.0.5 or 11.2
- Does not eliminate the difficulty in maintaining accurate Global Statistics.
- Does not work well with composite partitioned tables.
- Does not work in current releases with List Partitioning where there is a DEFAULT partition

APPROX_GLOBAL AND PARTITION

- New 10.2 GRANULARITY option as an alternative to GLOBAL AND PARTITION
- Uses the aggregation process, but can replace gathered global statistics
- If the aggregation process is unavailable, e.g. Because there are missing partition statistics, it falls back to GLOBAL AND PARTITION
- All the same NDV issues with aggregated stats so you should use with occasional Global Stats gather process

Incremental Statistics

- Introduction
- Simple Fundamentals
- Statistics on Partitioned Objects
- The Quality/Performance Trade-off
- Aggregation Scenarios
- Alternative Strategies
- **Incremental Statistics**
- Conclusions and References

Incremental Statistics

- What's the problem with the process for aggregating NDVs?
 - Oracle knows the *number* of distinct values in the other partitions but not *what* those values were
 - This might seem counter-intuitive. Oracle must have known what the values were when stats were gathered.
 - But they are not *stored* anywhere
 - Aggregation is a destructive process
- Incremental Statistics feature tracks the distinct values, stored as synopses
 - Stored in WRI\$_OPTSTAT_SYNOPSIS_HEAD\$ and WRI\$_OPTSTAT_SYNOPSIS\$

Incremental Statistics

- Prerequisites
- INCREMENTAL setting for the partitioned table is TRUE
 - Set using `DBMS_STATS.SET_TABLE_PREFS`
- PUBLISH setting for the partitioned table is TRUE
 - Which is the default setting anyway
- The user specifies (both defaults)
 - `ESTIMATE_PERCENT => AUTO_SAMPLE_SIZE`
 - `GRANULARITY => 'AUTO'`

New Process

- Gather initial statistics using the default settings
 - Oracle will gather statistics at all appropriate levels using one-pass distinct sampling and store initial synopses
- As partitions are added or stats become stale, keep gathering using AUTO granularity and Oracle will
 - Gather missing or stale partition stats
 - Update synopses for those partitions
 - Merge the synopses with synopses for higher levels of the same object, maintaining all Global Stats along the way
- Intelligent and accurate aggregation process

Other Resources

- Amit Poddar's excellent paper and presentation from earlier Hotsos Symposium
- Robin Moffat's blog post
 - Synopses can take a *lot* of space in SYSAUX
 - Aggregation seems hopelessly slow in older releases. Probably because `WRI$_OPTSTAT_SYNOPSIS$` is not partitioned (it is in 11.2.0.2)
- Incremental Stats looks like the solution to our problems
 - If you have the time to gather using defaults

Conclusions and References

- Introduction
- Simple Fundamentals
- Statistics on Partitioned Objects
- The Quality/Performance Trade-off
- Aggregation Scenarios
- Alternative Strategies
- Incremental Statistics
- Conclusions and References

Issues

- Aggregated NDVs are very low quality
- DBMS_STATS will only update aggregated stats when stats have been gathered appropriately on all underlying structures
- DBMS_STATS will never overwrite properly gathered Global Stats with aggregated results
 - Unless you use 'APPROX_GLOBAL AND PARTITION'
 - APPROX_GLOBAL stats otherwise suffer from the same problems as any other aggregated stats
 - If aggregation fails because of missing partition stats, you will suddenly be using GLOBAL AND PARTITION

Issues

- Dynamic Sampling is almost certainly not the answer to your problems
- The default setting of `_minimal_stats` aggregation implies that you should normally use exchange-then-gather
- If you are using Incremental Stats you must use exchange-then-gather anyway

Suggestions

- Try the Oracle default options first, particularly 11.2 and up
- If you do not have time to gather using the default granularity, gather the best statistics you can as data is loaded and gather proper global statistics later
- DBMS_STATS is constantly evolving so you should try to be on the latest patchsets with all relevant one-off patches applied
- Checking stats means checking all levels, including
 - GLOBAL_STATS column
 - NUM_DISTINCT and High/Low Values

Suggestions

- Design a strategy
- Develop any surrounding code
- Stick to the strategy
- Always gather stats using the wrapper code
- Lock and unlock stats programmatically to prevent human errors ruining the strategy

Additional References

- Optimiser Development Group blog
- Greg Rahn's blog
- Amit Poddar's Paper
- Jonathan Lewis chapter in latest Oak Table book
- Lots of others in references section of paper

Doug Burns
doughburns@yahoo.com
<http://oracledoug.com/stats.docx>

Statistics on Partitioned Objects

