



ORACLE®

Designing and Developing Highly Scalable Applications with the Oracle Database

Mark Townsend
VP, Database Product Management
Server Technologies, Oracle



Background

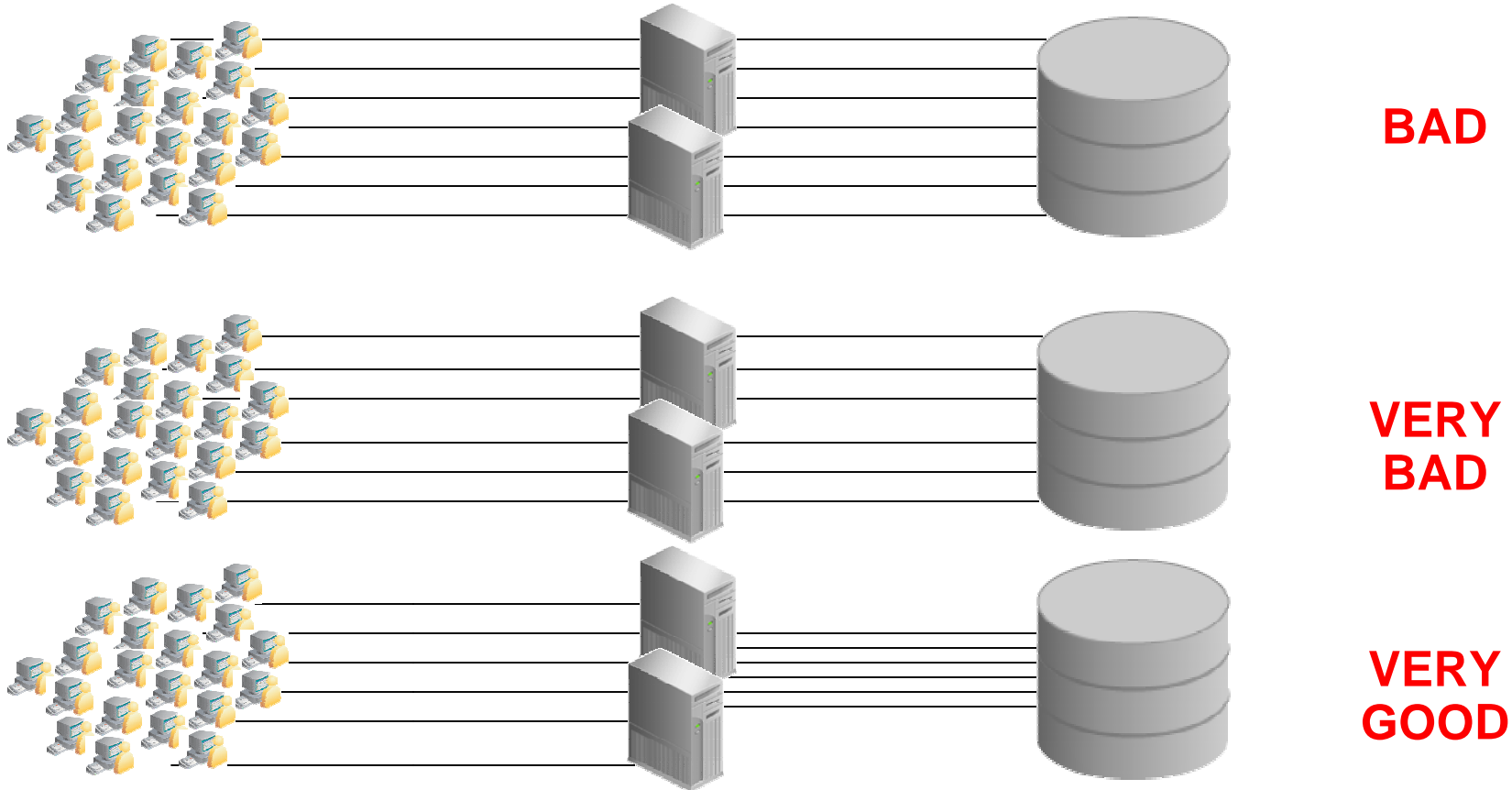
- **Information from Thomas Kyte (AskTom), Andrew Holdsworth (Real World Performance Team)**
- **Rules of Engagement:**
- **Do not take anything as a given**
 - **Ask for an explanation**
 - **Test it for yourself**
- **Test Properly**
 - **On an exact copy of the data (Importing stats is OK but a poor second)**
 - **The same number of users as in production**
- **Test and Optimize for the worst case, not the best**
 - **Optimize for the spikes in workload or scalability, not the average**
- **Yes – this is extremely difficult to do – more later**



Improving performance and scalability against Oracle Databases

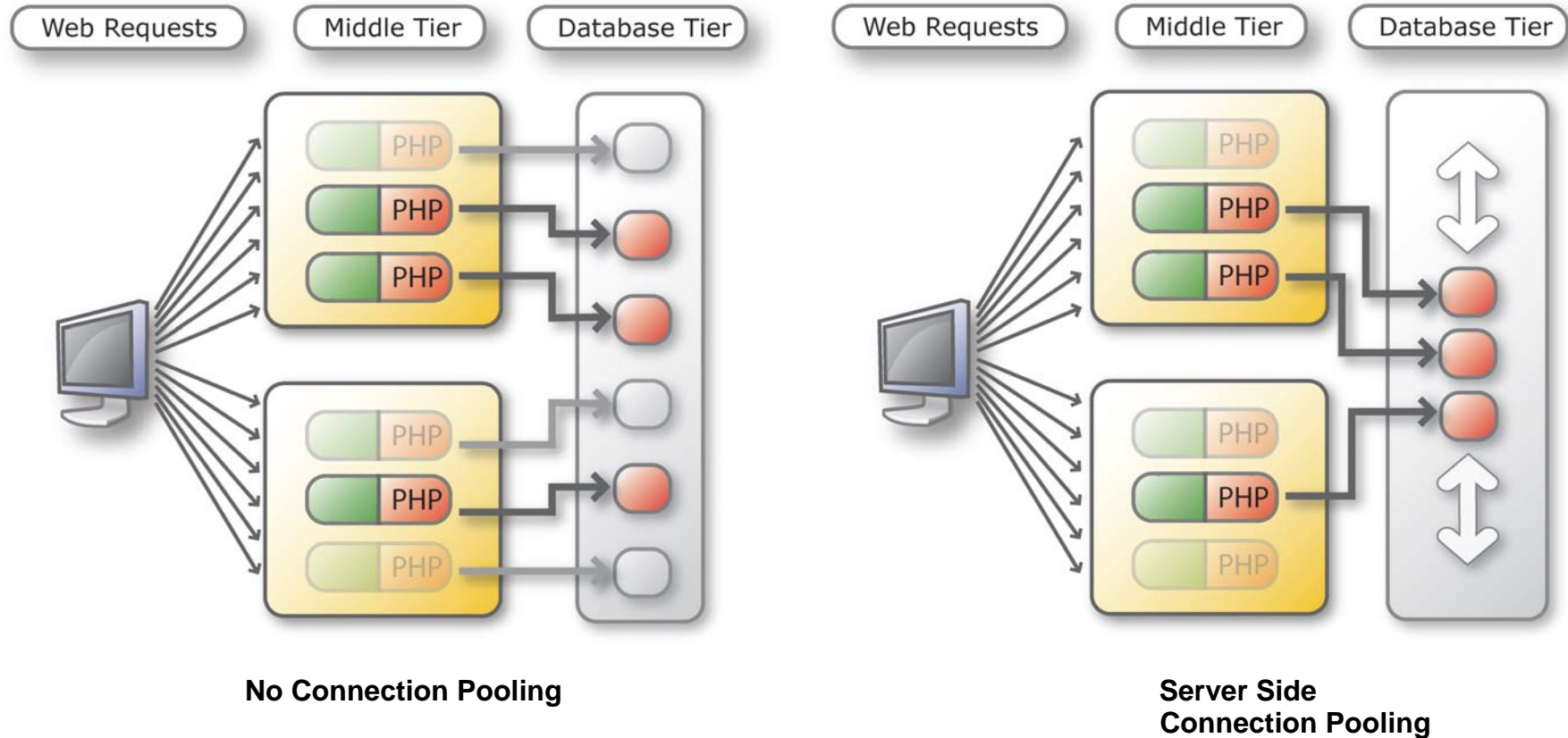
- Manage Connections
- Use Bind Variables
- Divide and Conquer with Partitioning
- Effective Indexing
- Inserting Large Amounts of Data
- Deleting Large Amounts of Data
- You will need lots of Storage
- Testing, testing and more testing

Manage Connections



- Do not connect all browser users
- Do not connect, do something, disconnect
- Use Connection Pools - #1 advantage of the middle tier

Managing Connections – New in 11g



- **Connection Pooling supported in .Net, Java and OCI**
 - All above for 10g and earlier
- **Oracle Database 11g adds support for PHP Connection Pooling**

Bind Variables – Use them

```
query =  
'select *  
  from t  
  where x = ' || x || '  
  And y = ' || y
```


Execute it

**Less Code, Bad
Performance, Doesn't Scale**

```
query =  
'select *  
  from t  
  where x = ?  
  And y = ?'
```

Prepare it
Bind x
Bind y
Execute it
Close it

**More Code, Better
Performance, Scales Well**



Bind Variables – Performance and Scalability – Parse1

```
set timing on
begin
    for i in 1 .. 10000
    loop
        execute immediate 'insert into t(x,y) values
('||i||','X")';
    end loop;
end;
/
```

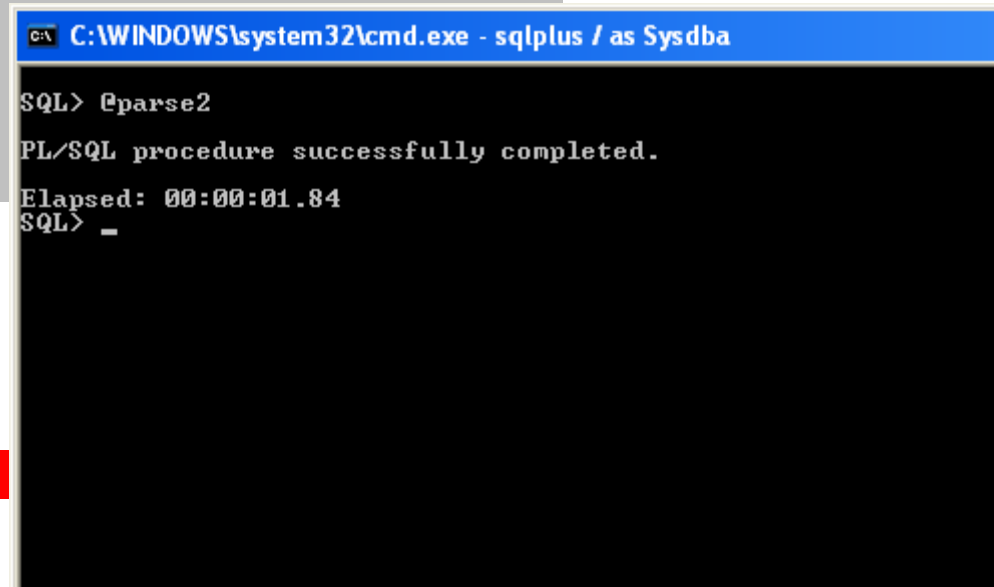


```
C:\WINDOWS\system32\cmd.exe - sqlplus / as Sysdba
```

```
SQL> @parse1
PL/SQL procedure successfully completed.
Elapsed: 00:00:29.41
SQL> _
```

Bind Variables – Performance and Scalability

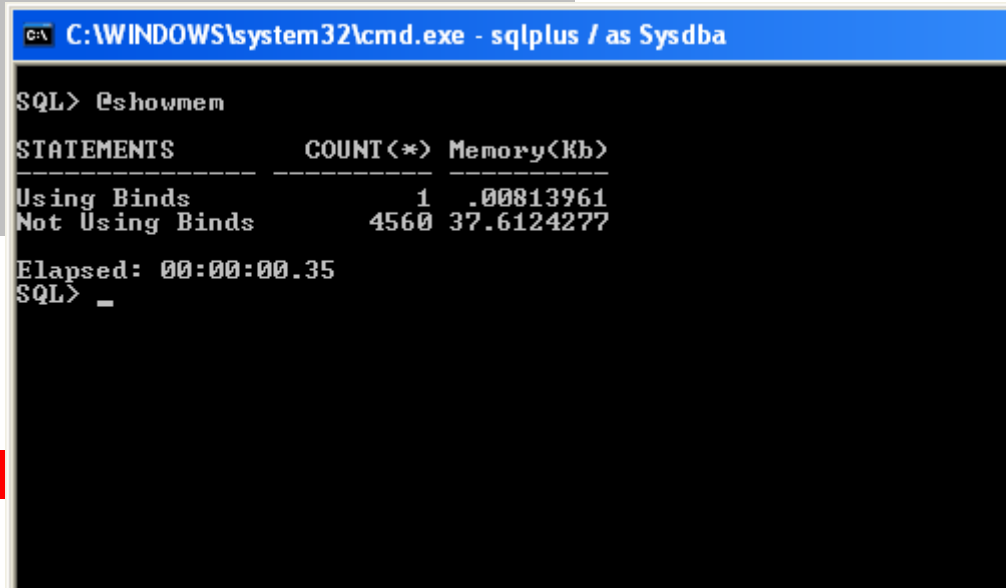
```
set timing on
begin
for i in 1 .. 10000
    loop
        execute immediate 'insert into t(x,y) values
(:i,"X")'
        using i;
    end loop;
end;
/
```



```
C:\WINDOWS\system32\cmd.exe - sqlplus / as Sysdba
SQL> @parse2
PL/SQL procedure successfully completed.
Elapsed: 00:00:01.84
SQL> _
```

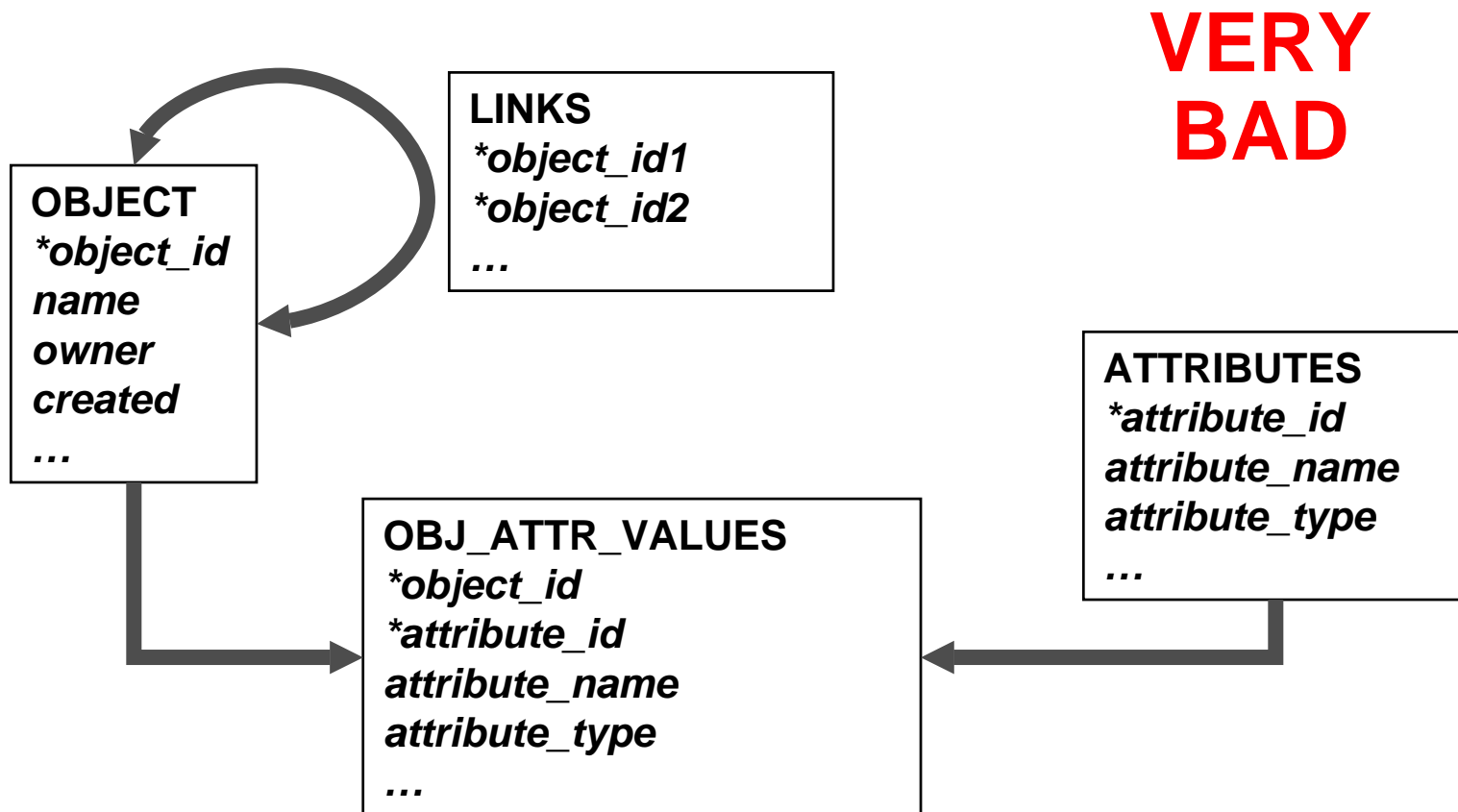
Bind Variables – Performance and Scalability

```
select
case when instr( sql_text, ':' ) > 0
      then 'Using Binds'
      else 'Not Using Binds'
end Statements,
count(*),
sum(sharable_mem)/(1024*1024) "Memory(Kb)"
from v$sql where sql_text like insert into t(x,y) values (%'
group by
case when instr( sql_text, ':' ) > 0
      then 'Using Binds'
      else 'Not Using Binds'
end
/
```

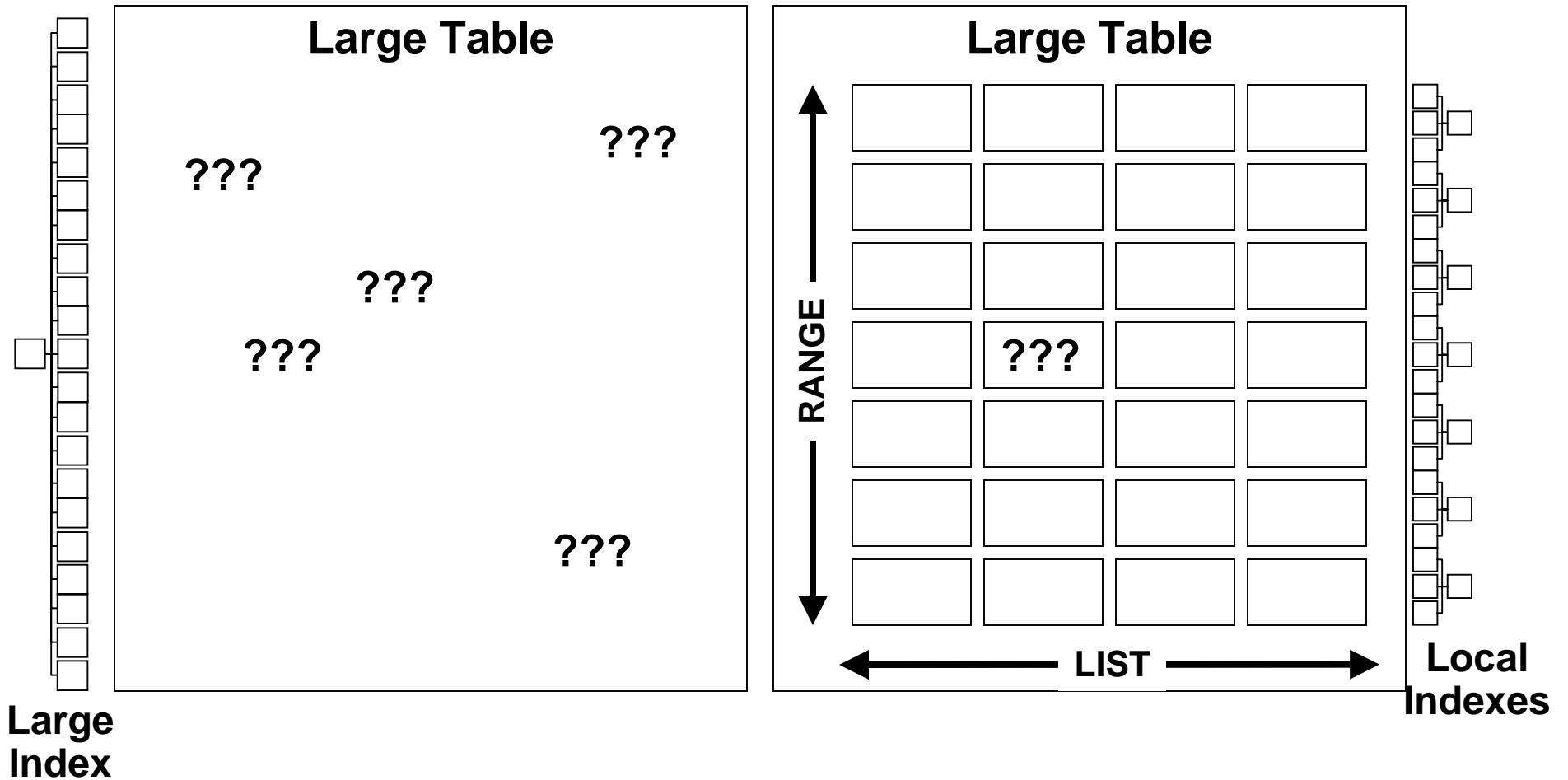


```
C:\WINDOWS\system32\cmd.exe - sqlplus / as Sysdba
SQL> @showmem
STATEMENTS          COUNT(*)  Memory(Kb)
-----
Using Binds          1         .00813961
Not Using Binds     4560      37.6124277
Elapsed: 00:00:00.35
SQL> _
```

How many tables do you really need ?

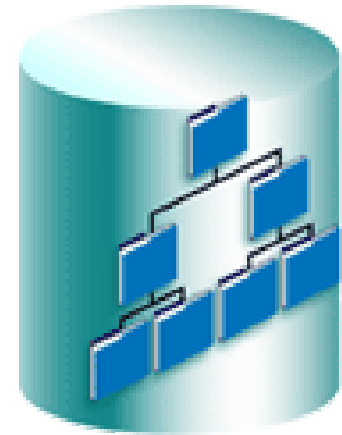


Divide and Conquer with Partitioning



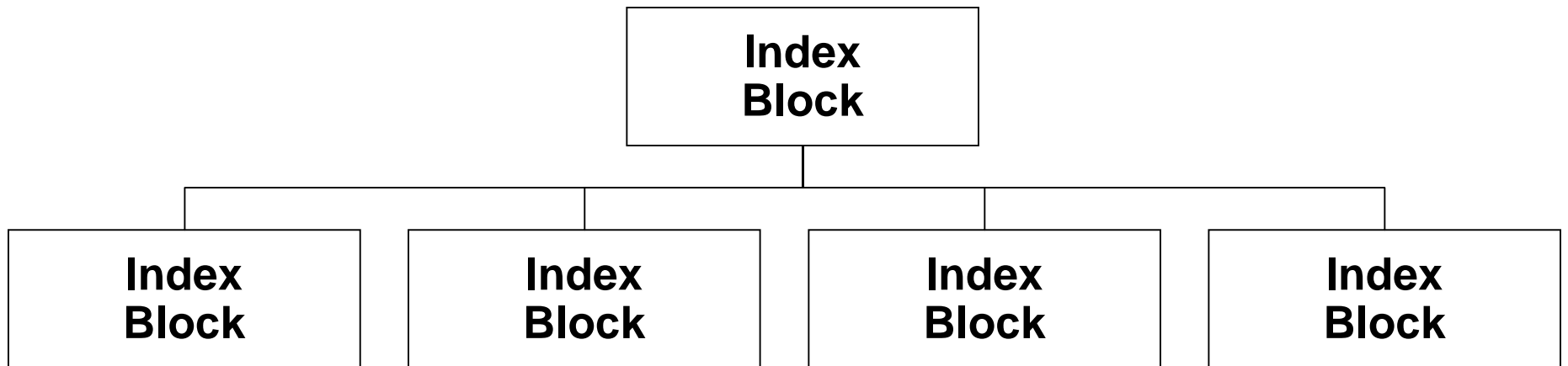
The best IO is the one you don't do – Indexes

- Indexes help you avoid I/O
- BTree indexes based on the columns in the predicate work very well for access
- Indexes also have a insert cost – roughly 3 times the cost per insert per index
- Concatenate index columns in the order of selectivity
- One index on (C1,C2,C3, C4) is better than three indexes on (C1, C2), (C2, C3), (C1, C4)

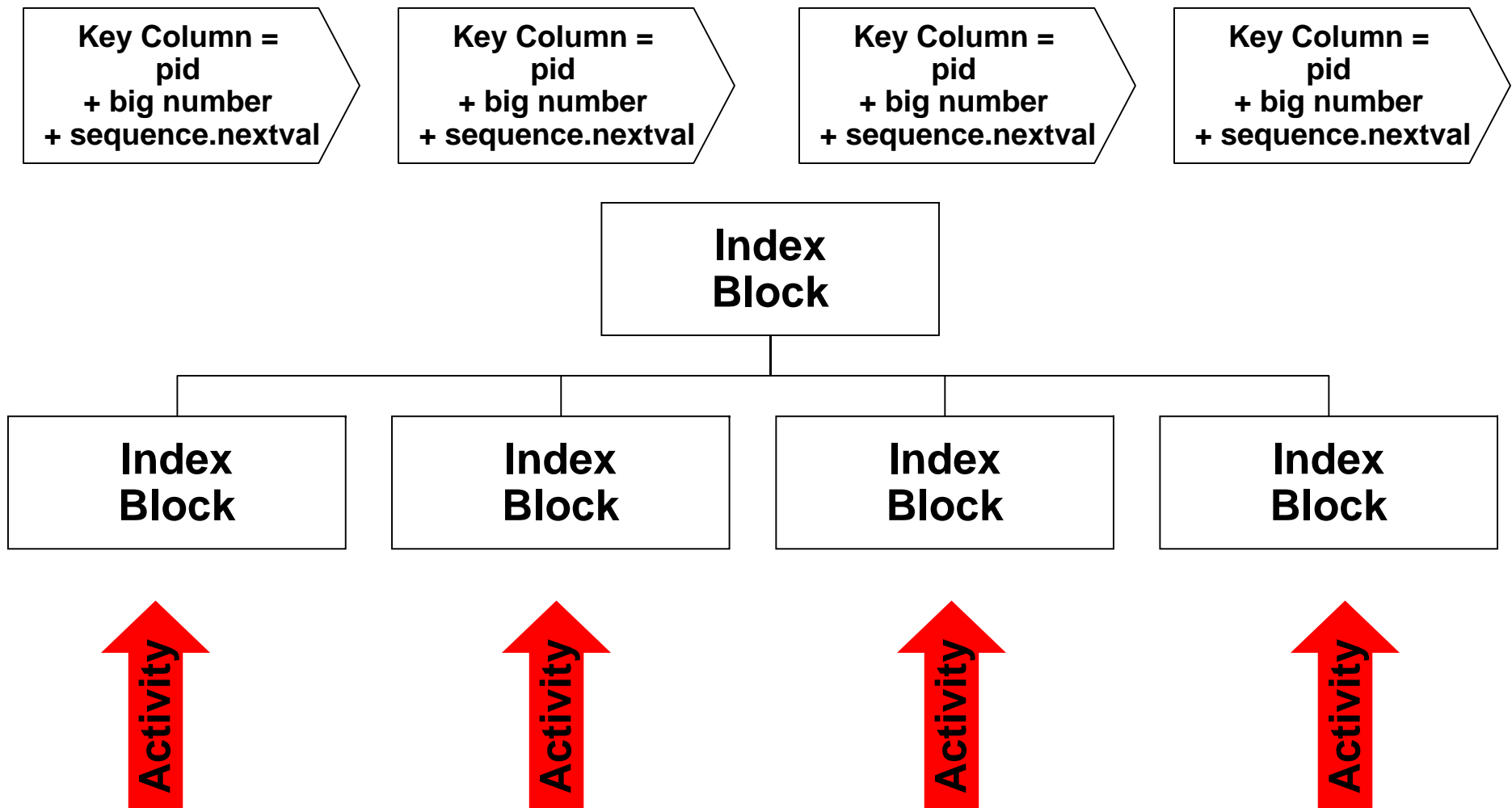


Unique Indexes – Use a generated key to spread I/O

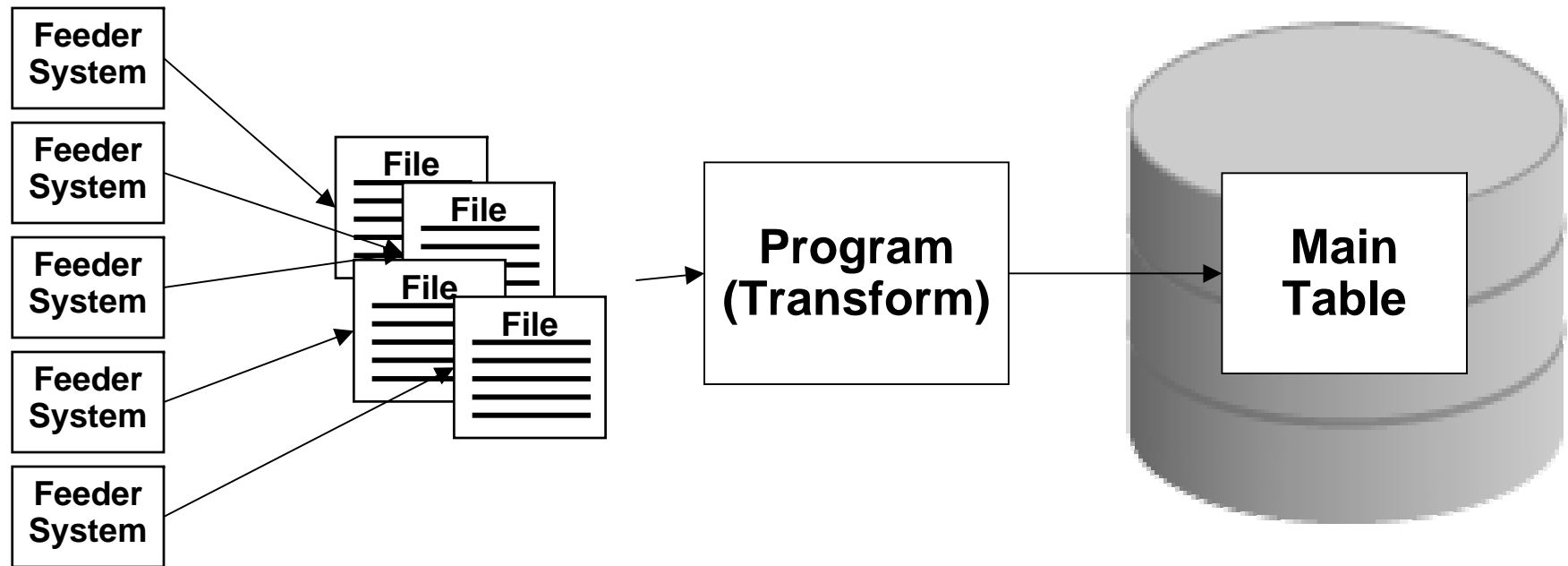
Key Column = `sequence.nextval`



Unique Indexes – Use a generated key to spread I/O

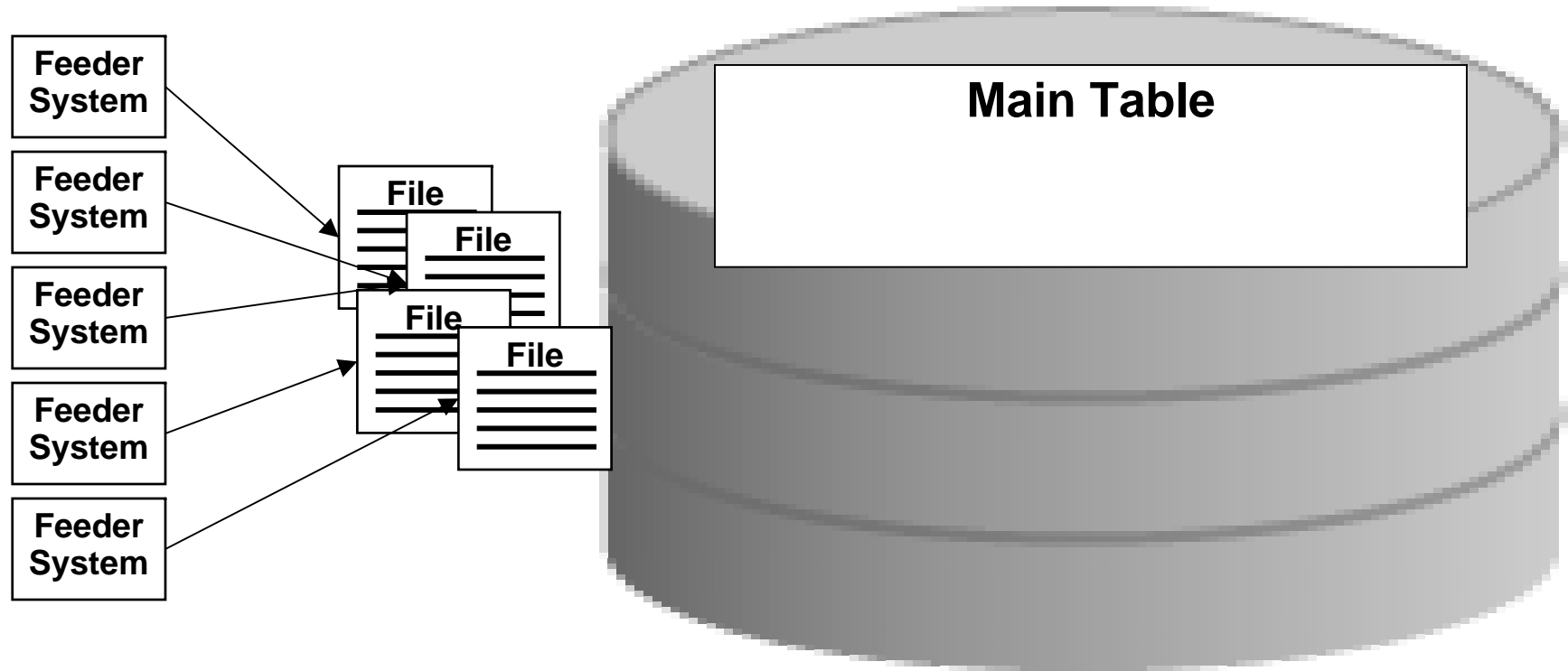


Inserting large amounts of data

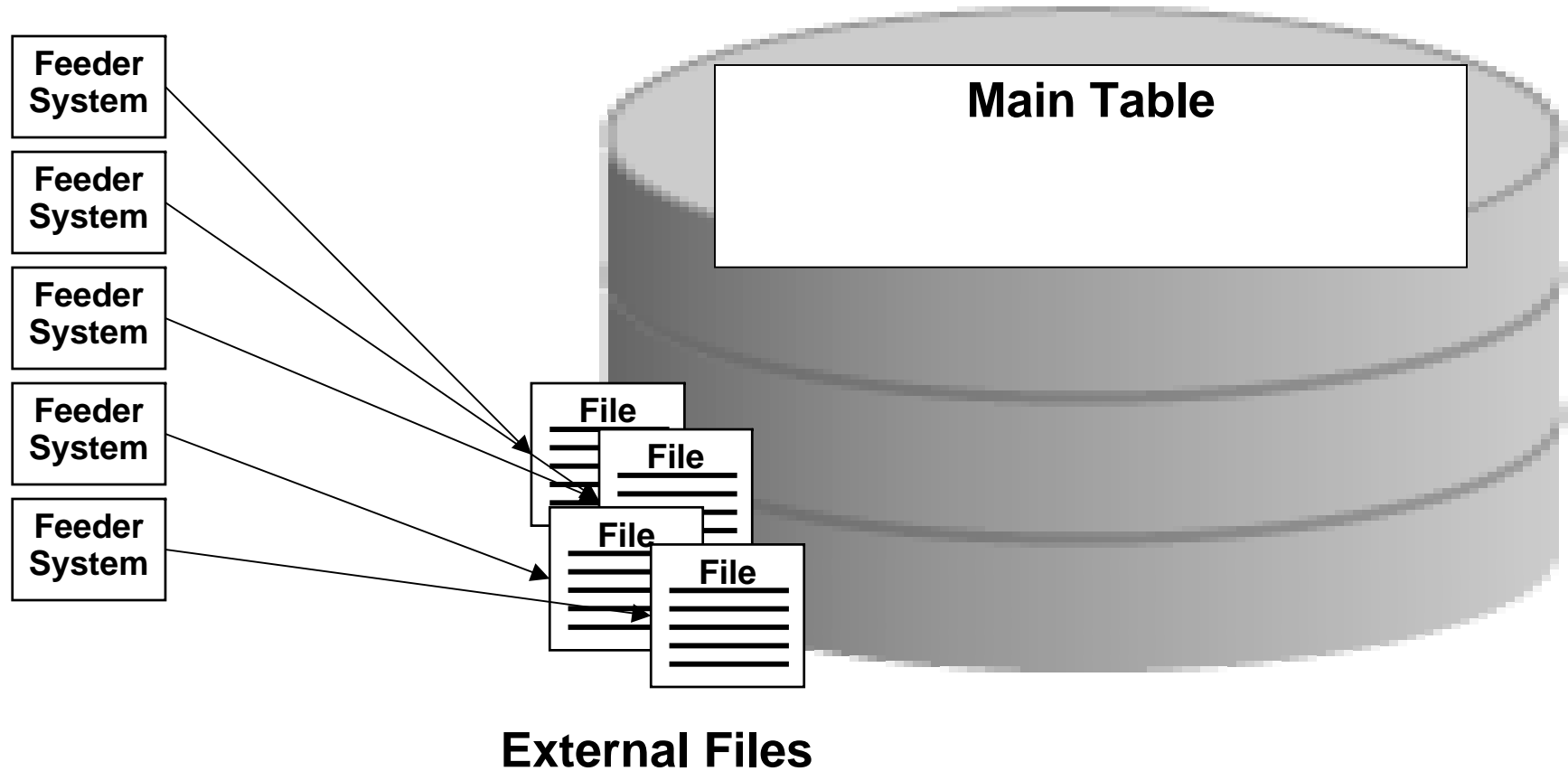


- Index Maintenance
- Statistics Maintenance
- Logging

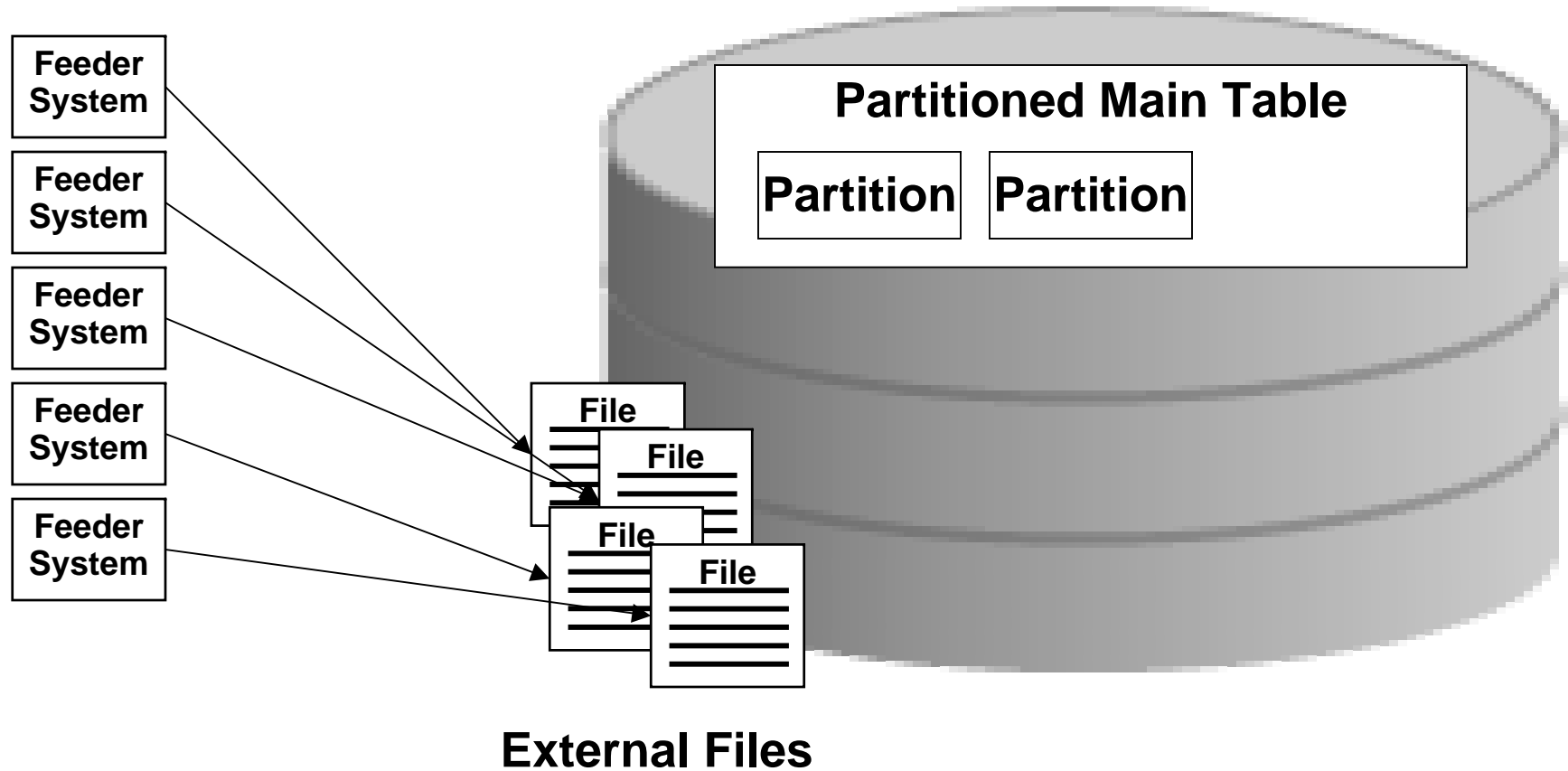
Inserting large amounts of data



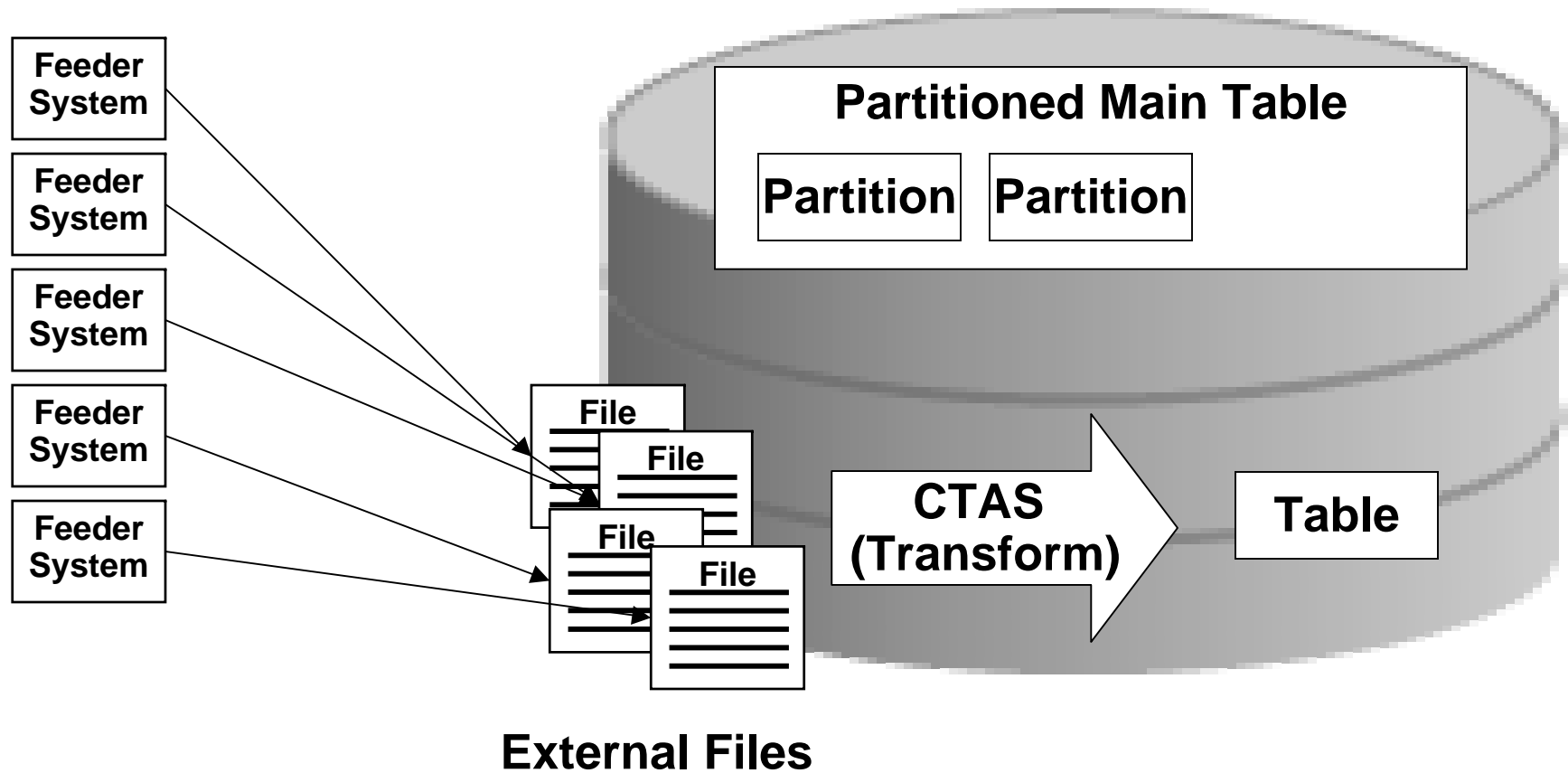
Inserting large amounts of data



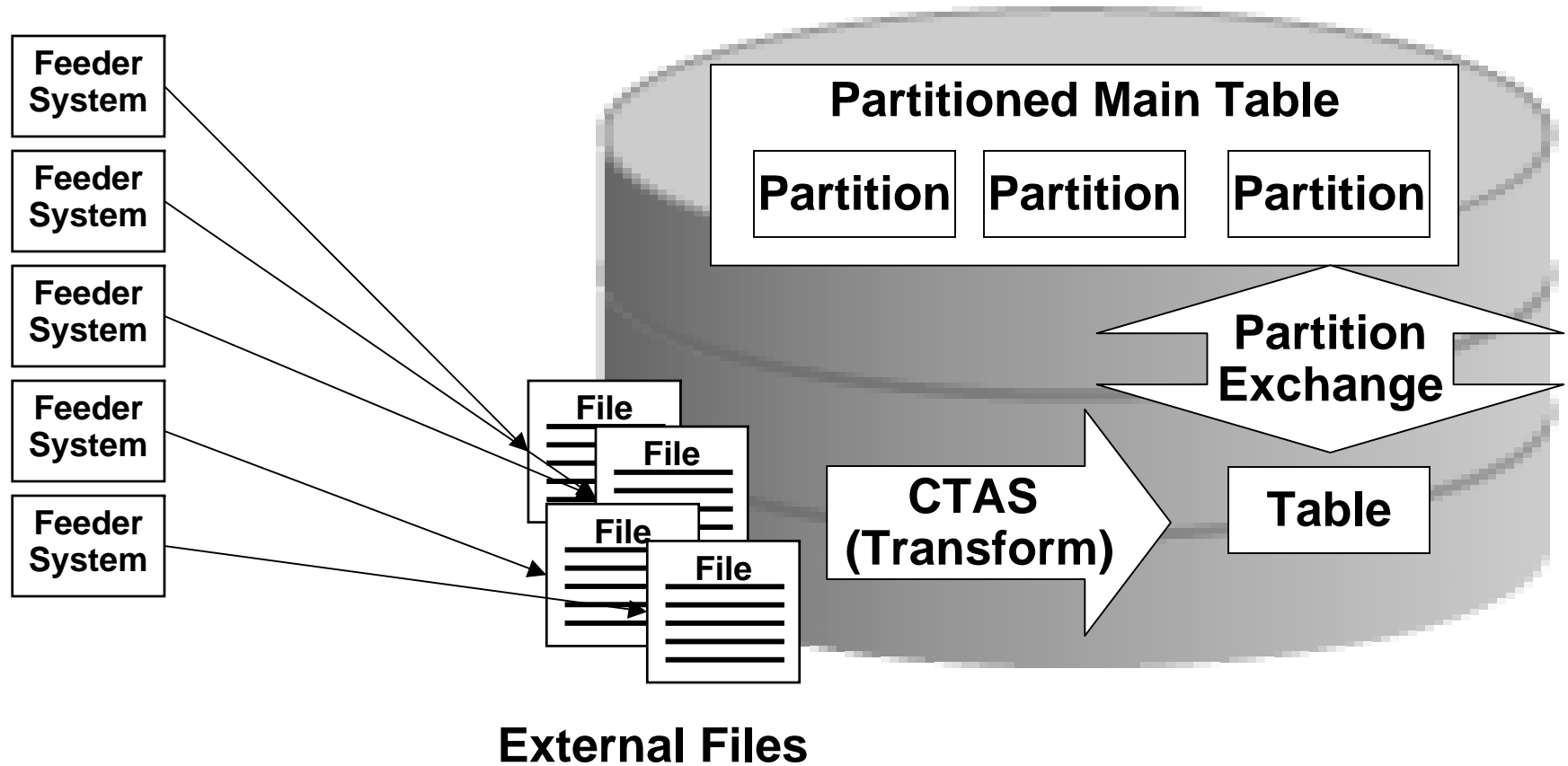
Inserting large amounts of data



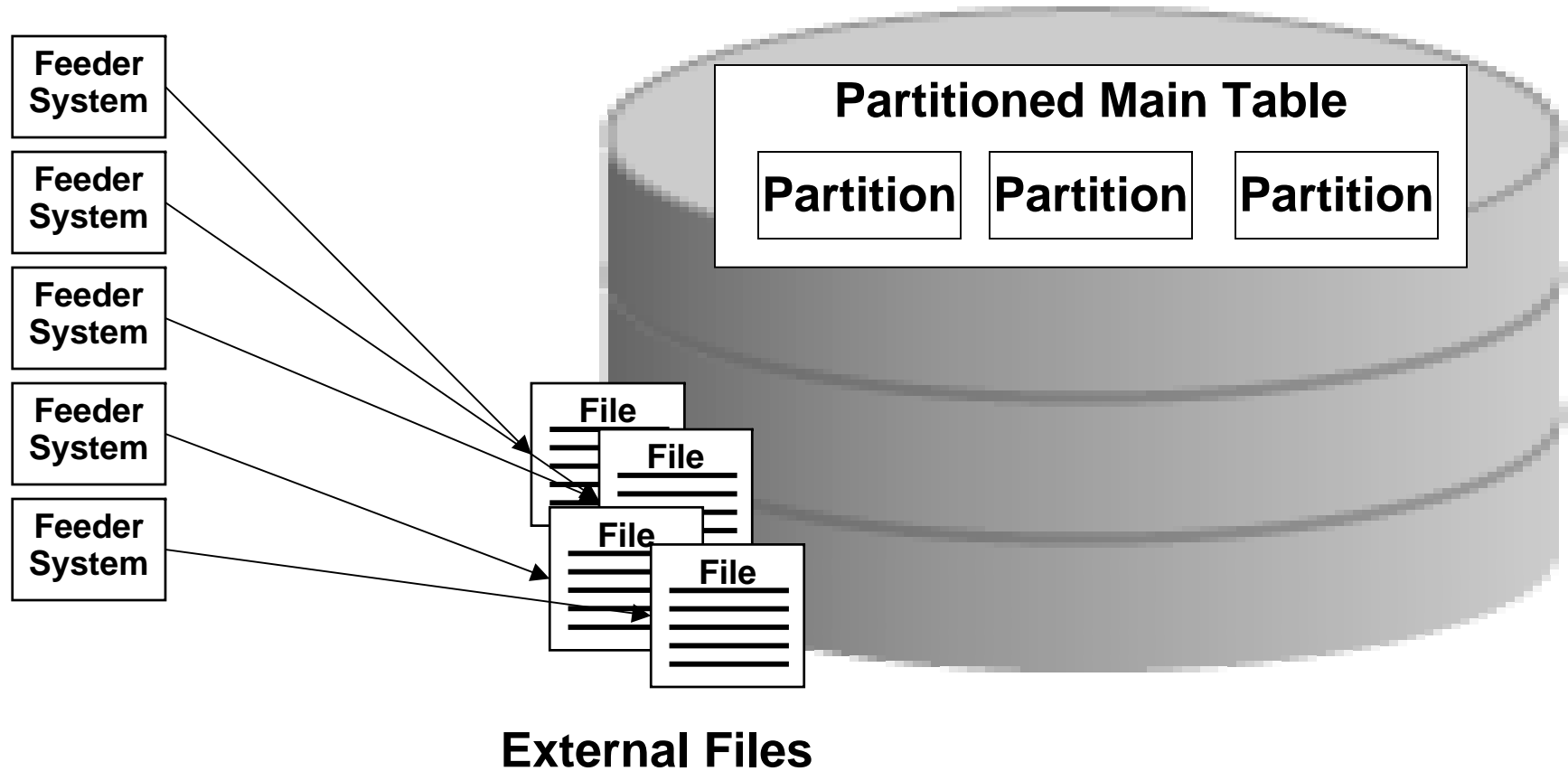
Inserting large amounts of data



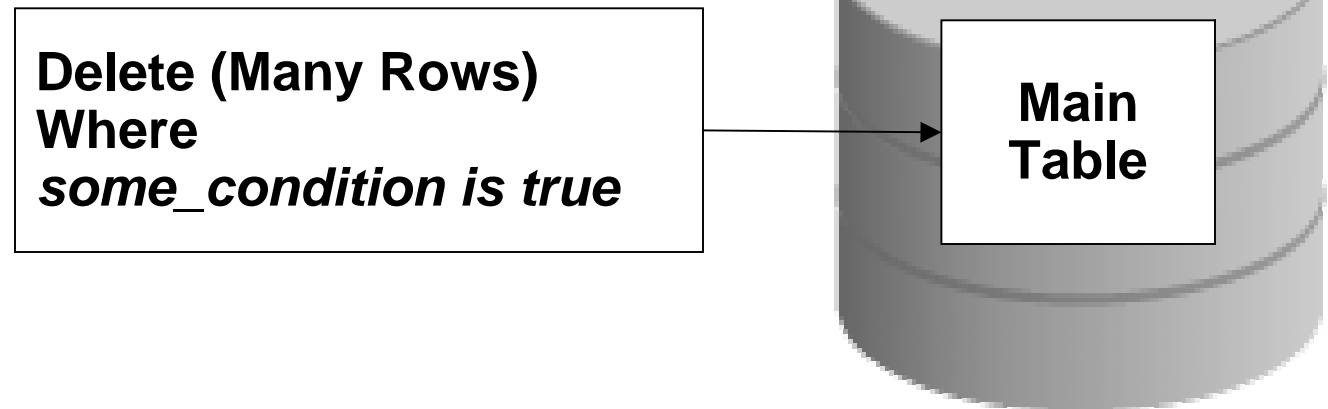
Inserting large amounts of data



Inserting large amounts of data

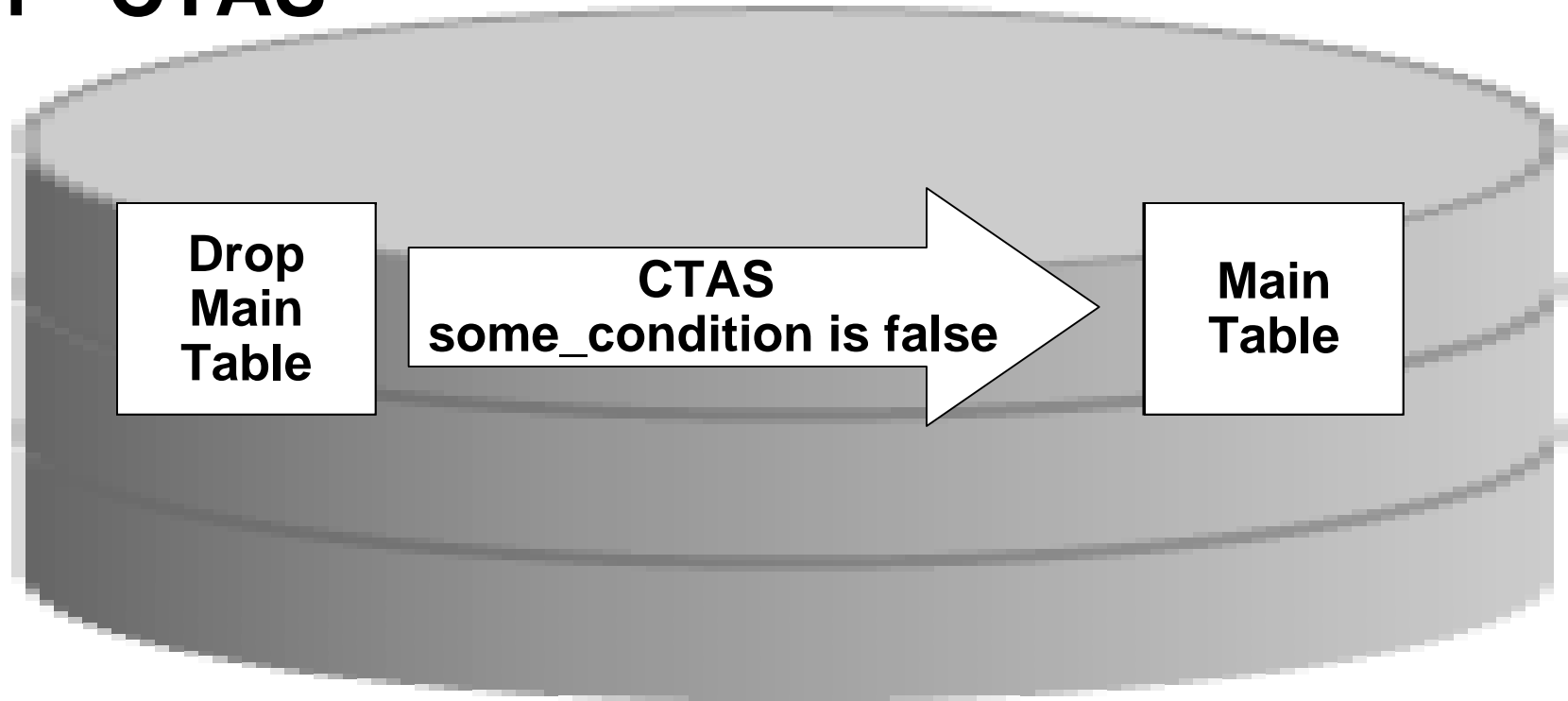


Deleting large amounts of data



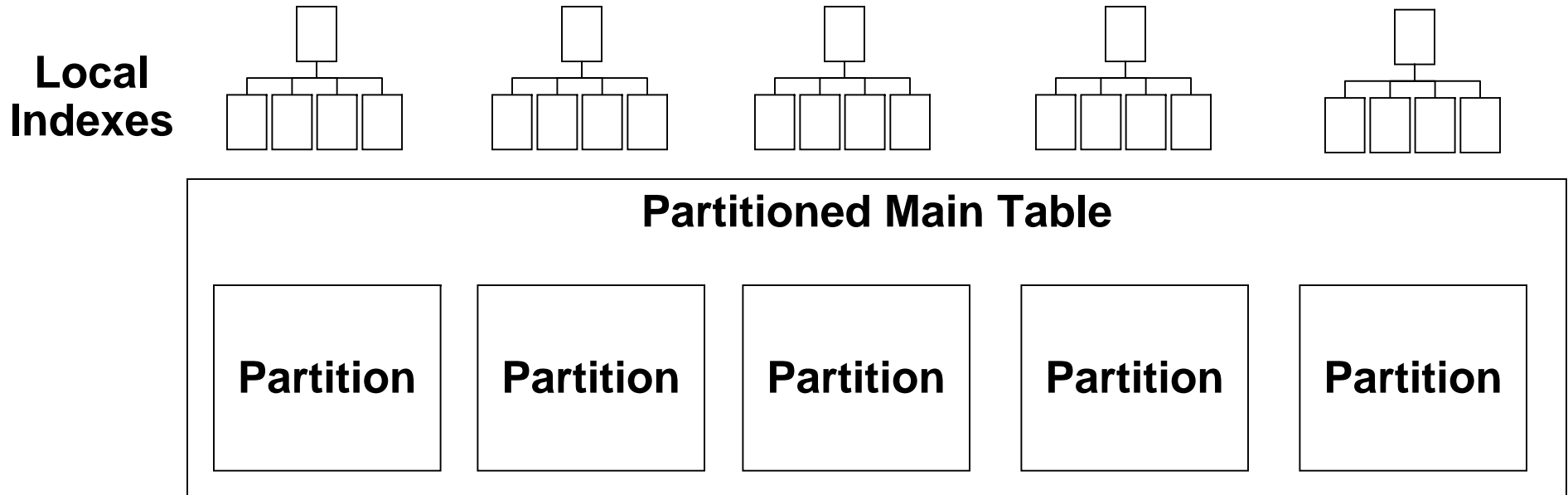
- **Index Maintenance**
- **Statistics Maintenance**
- **Logging**

Deleting large amounts of data – Trick 1 - CTAS

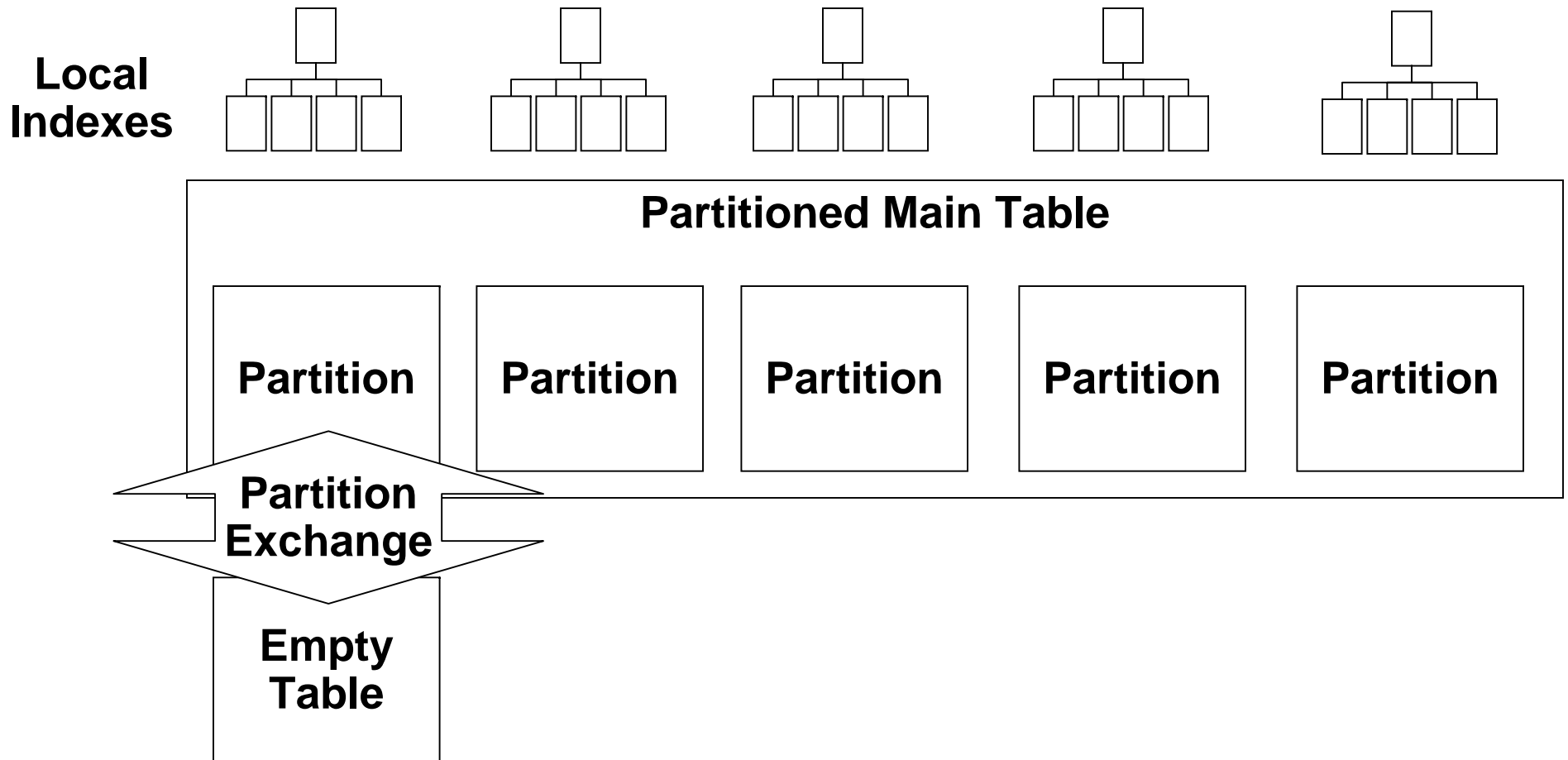


- Less redo log, data is reloaded
- Indexes need to be rebuilt
- OK for small tables, not too good for large tables

Deleting large amounts of data – Trick 2 – Partition Exchange

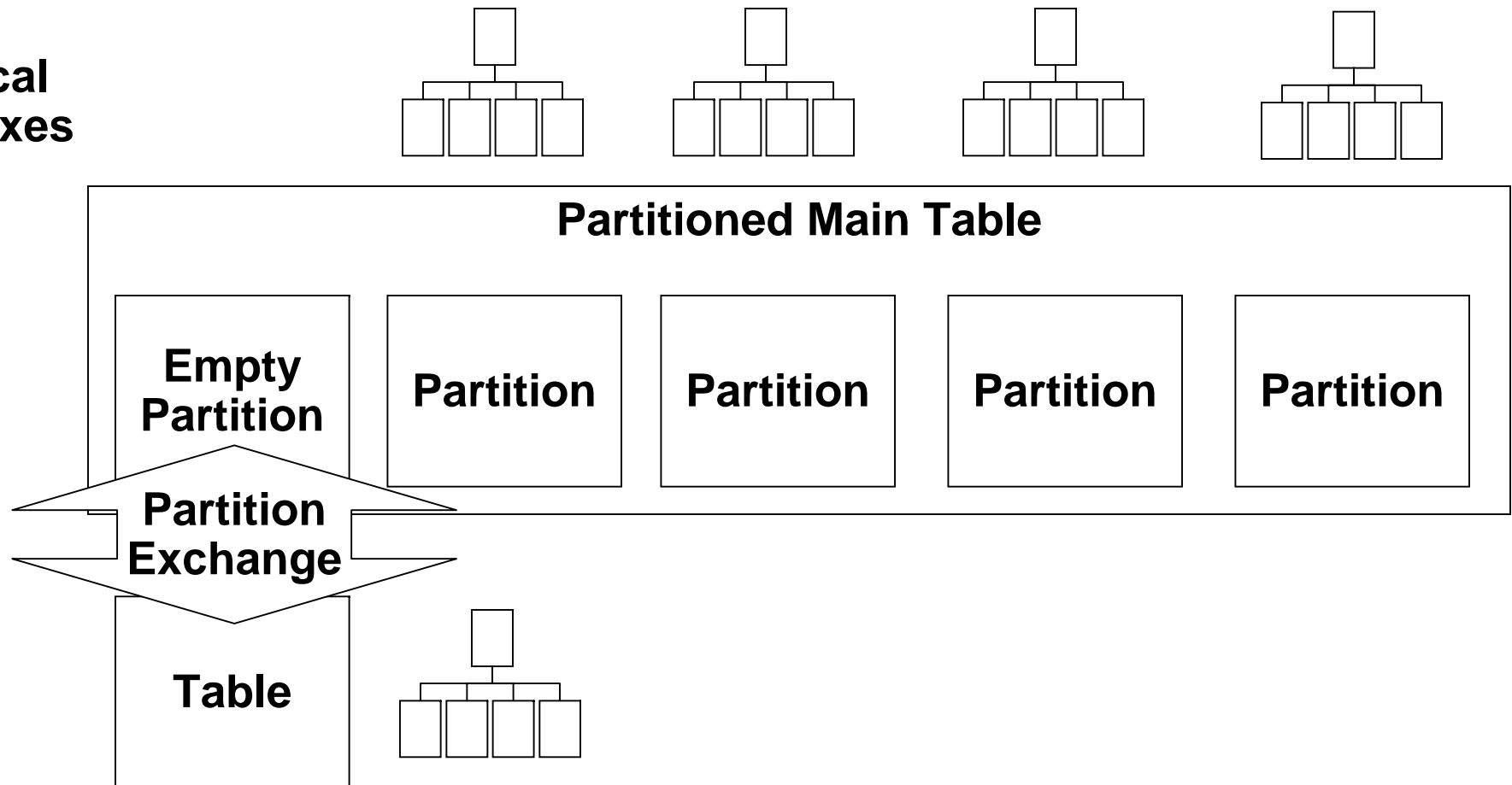


Deleting large amounts of data – Trick 2 – Partition Exchange



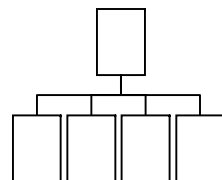
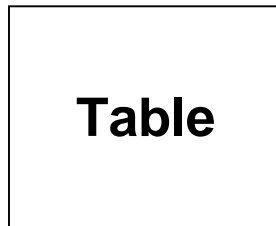
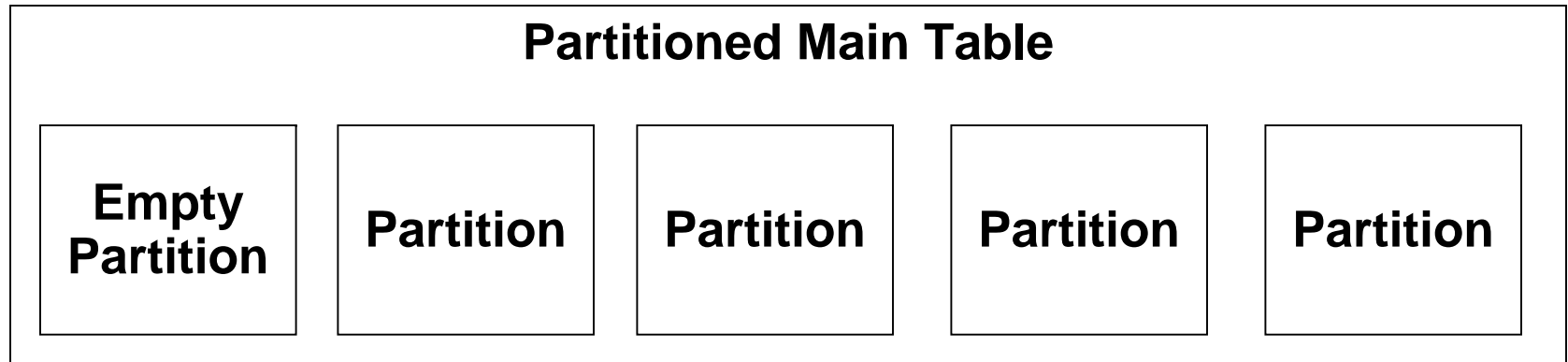
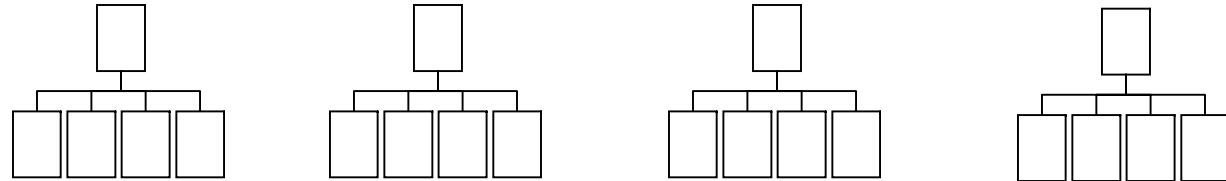
Deleting large amounts of data – Trick 2 – Partition Exchange

Local
Indexes



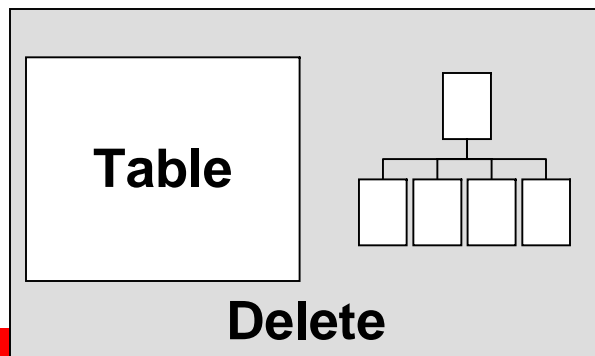
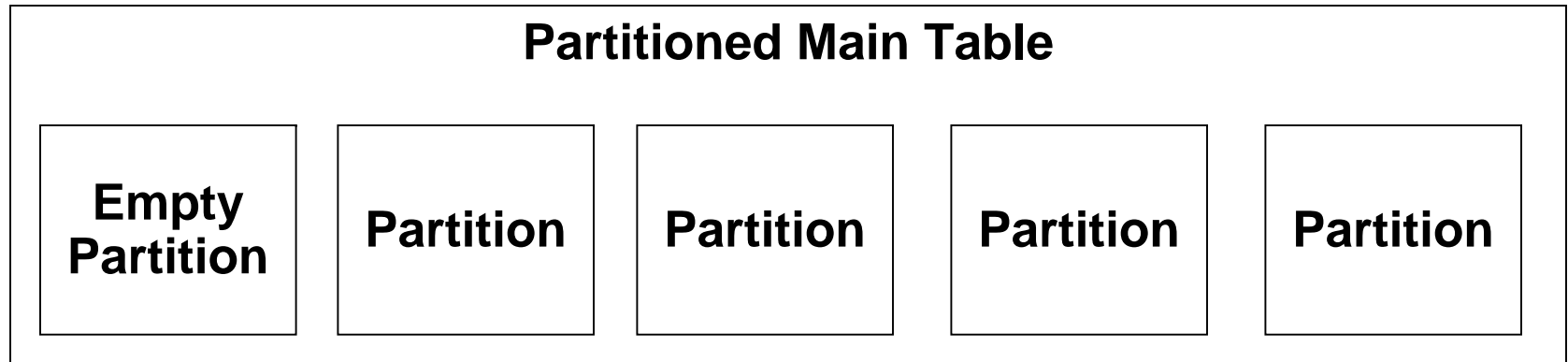
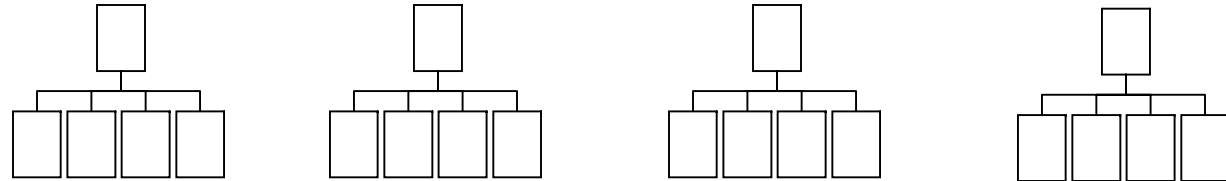
Deleting large amounts of data – Trick 2 – Partition Exchange

Local
Indexes



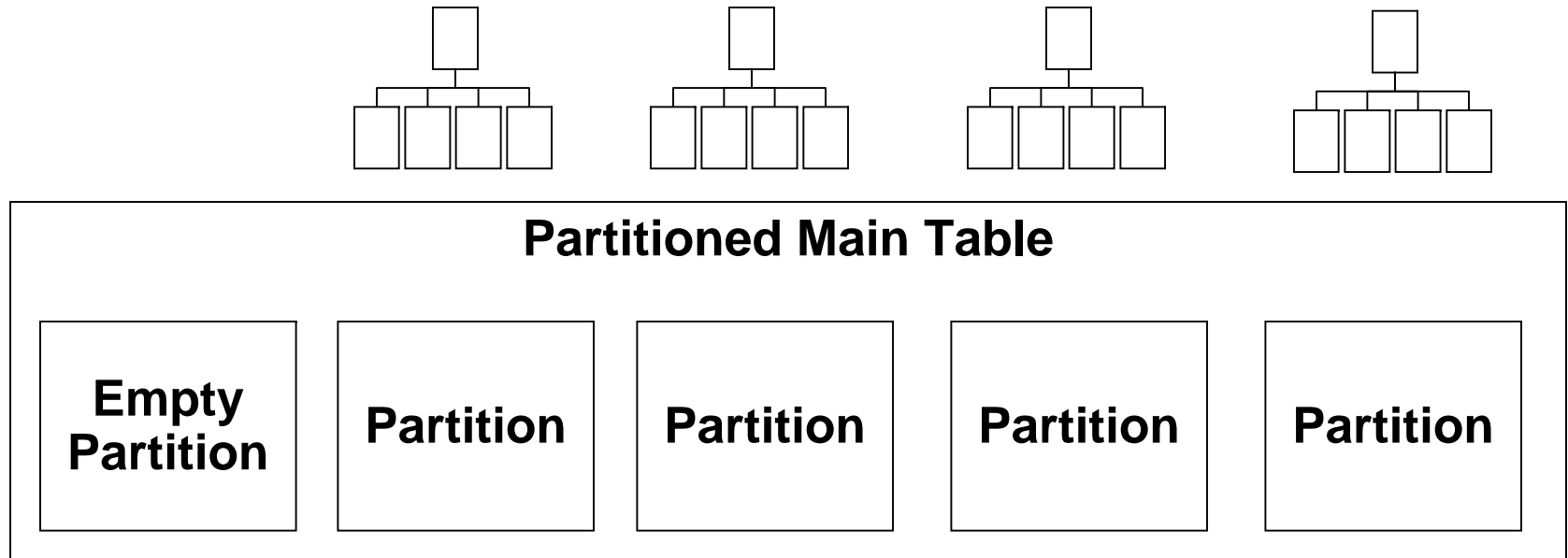
Deleting large amounts of data – Trick 2 – Partition Exchange

Local
Indexes



Deleting large amounts of data – Trick 2 – Partition Exchange

Local
Indexes



- Very quick – Exchange Partition is a dictionary option
- Indexes and statistics maintained
- Can backup deleted data before dropping it



Choosing a Partitioning Strategy and Partitioning Key

- There may be a conflict between the best partition column for delete, and the best partition column for performance
 - I.e transaction date vs. post date
- You may need to find an alternate.
- Take advantage of Composite Partitioning capabilities
 - Oracle Database 10g and earlier
 - Range, Range-Hash, Range-List
 - Oracle Database 11g
 - Range-Range, List-Range, List-List, List-Hash



Best Statistics Management Procedure for OLTP Environments

- Goal - get plan stability and same plans as in test environment (where you have validated them). The same plan will run millions of times
- In an OLTP environment be very careful with changes in stats as the plans can change. Test plan changes in test before they get to development
- Consider locking the statistics so that they can never change (using DBMS_STATS)
- Avoid histograms



Best Statistics Management Procedure for DSS Environments

- **Goal – get the statistics to represent the data as much as possible. A plan may only run once, but needs to be as good as possible. However you want to minimize the gathering of statistics. Techniques to minimize include:**
- **Use the DBMS_STATS package as opposed to the ANALYZE command**
- **Do not use COMPUTE statistics for 100% of the rows.**
- **Use ESTIMATE, infrequently**
- **Global level statistics are less expensive than partition level statistics on a partitioned table. If you do not need specific partition level optimizations, global level statistics will probably suffice.**



Storage Recommendations

- Balance the entire stack for the workload – CPUs to I/O bandwidth. People typically do not size their storage correctly.
- For OLTP - the key metric is I/O ops per second because of the random I/O.
 - Assume 5 I/O ops per transaction
 - 1,000 txns per second = 5,000 I/O ops per second
 - Assume 30 I/O ops per disk
 - Calculation is 1,000 txns per second = 5,000 I/O ops per second = $5,000/30$ disks = 166 disks
 - Use the additional disk space for online backups, archive redo logs etc.



Storage Recommendations

- For DW - the key metric is I/O bandwidth because of the table scans
 - As a rule of thumb, for each CPU (or core) you will need to sustain about 200 Mb/Sec
 - You need to size the HBA, switches and arrays downstream to matches this throughput so it is kept in balance.
 - Once again the cost of the storage becomes the big budget item.



Improving performance and scalability against Oracle Databases

- Manage Connections
- Use Bind Variables
- Divide and Conquer with Partitioning
- Effective Indexing
- Inserting Large Amounts of Data
- Deleting Large Amounts of Data
- You will need lots of Storage

Test, test and test again



Alfred North Whitehead:
Principia Mathematica

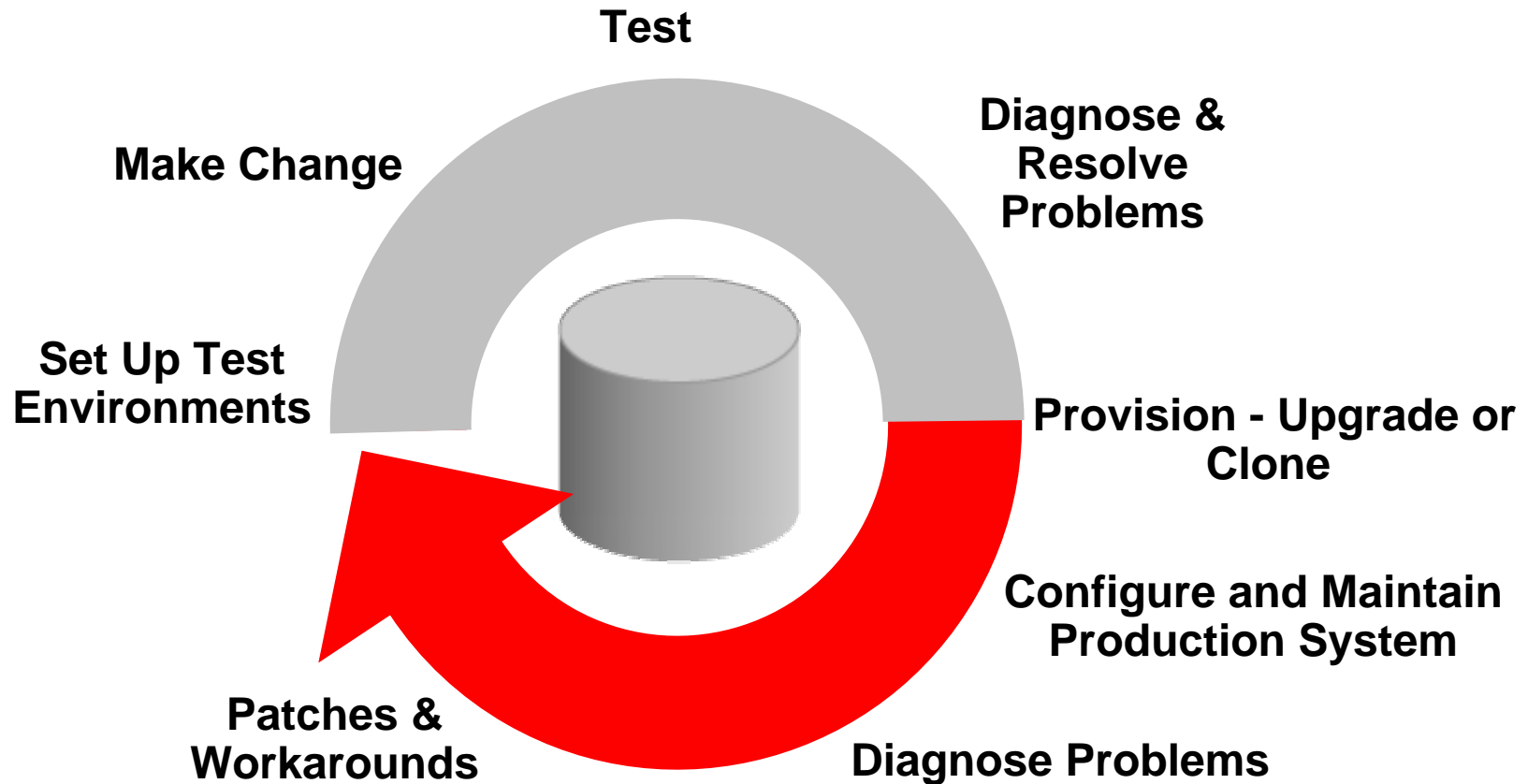
“The art of progress is to preserve order amid change and to preserve change amid order.”



**Oracle Database 11g will focus on
helping customers
preserve order amid change**



Lifecycle of Change Management

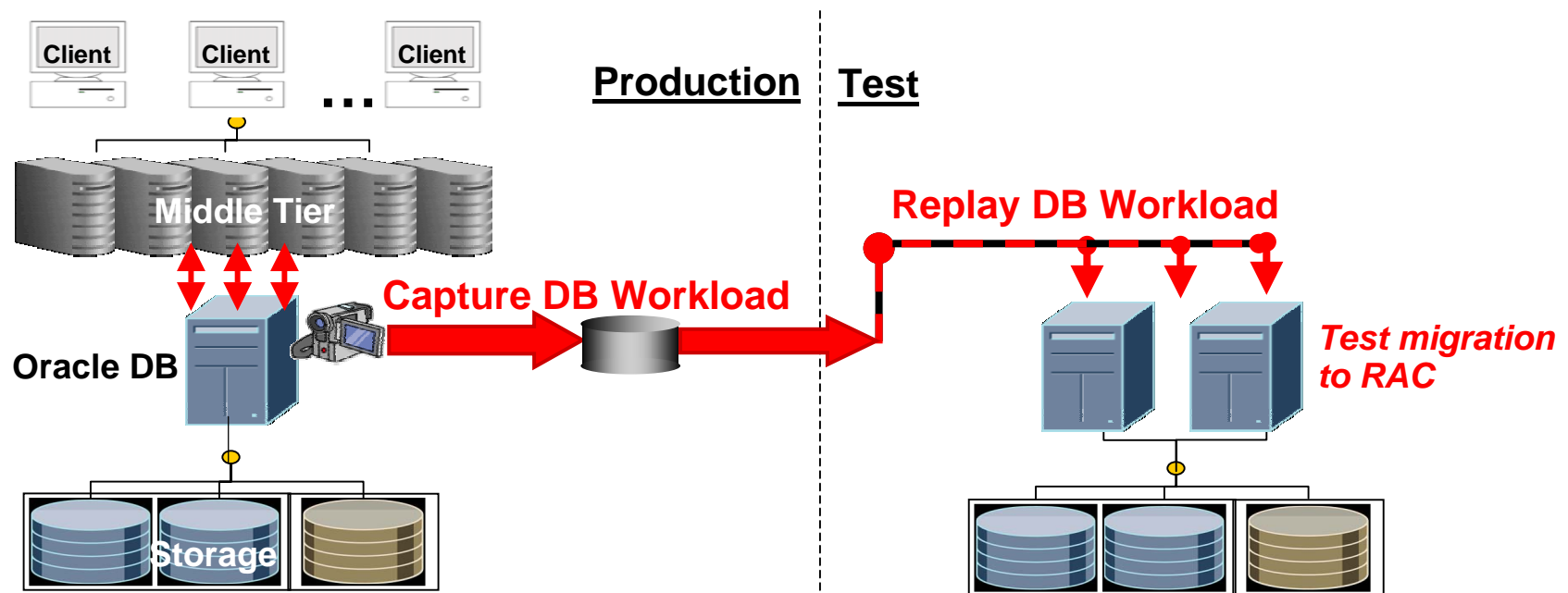


Preserve Order Amid Change

Realistic Testing with Database Replay

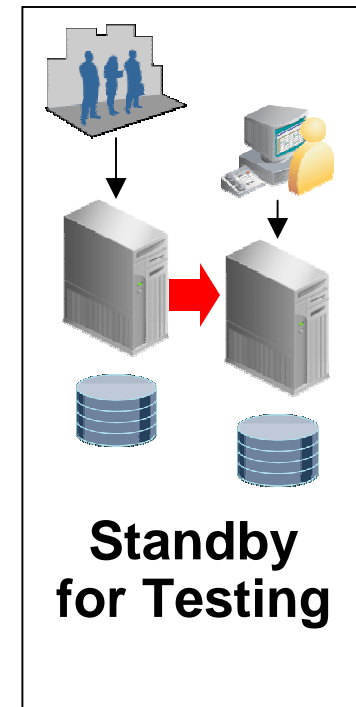
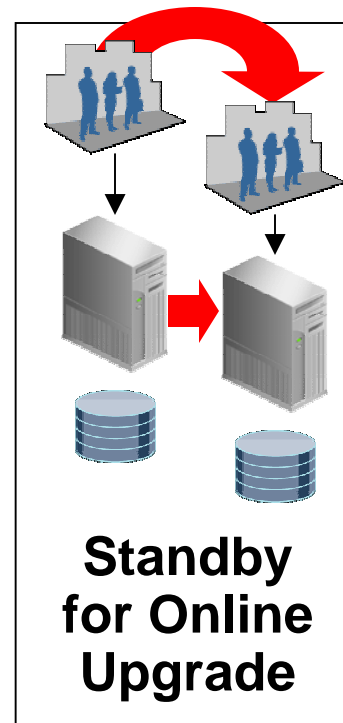
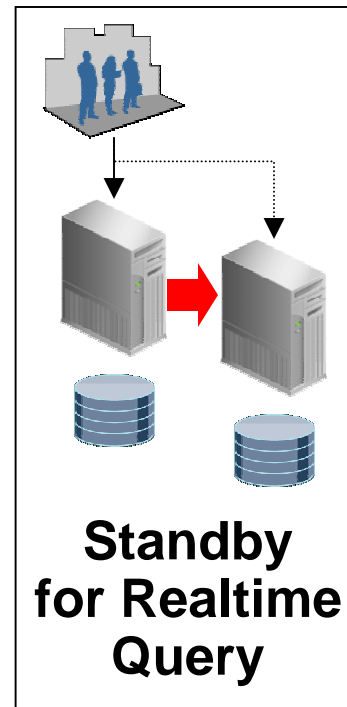
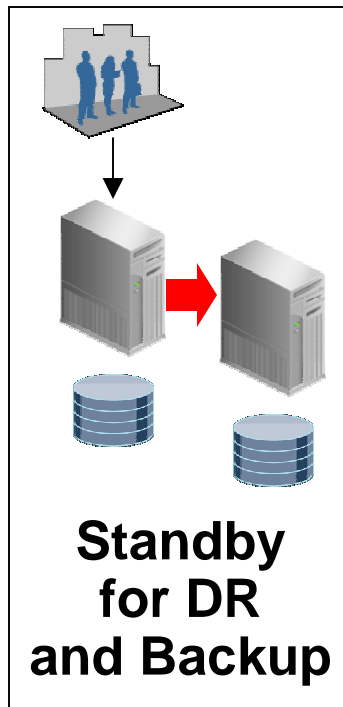
ORACLE[®] 11^g
DATABASE

- Recreate actual production database workload in test environment
- Capture workload in production including critical concurrency
- Replay workload in test with production timing
- Analyze & fix issues before production

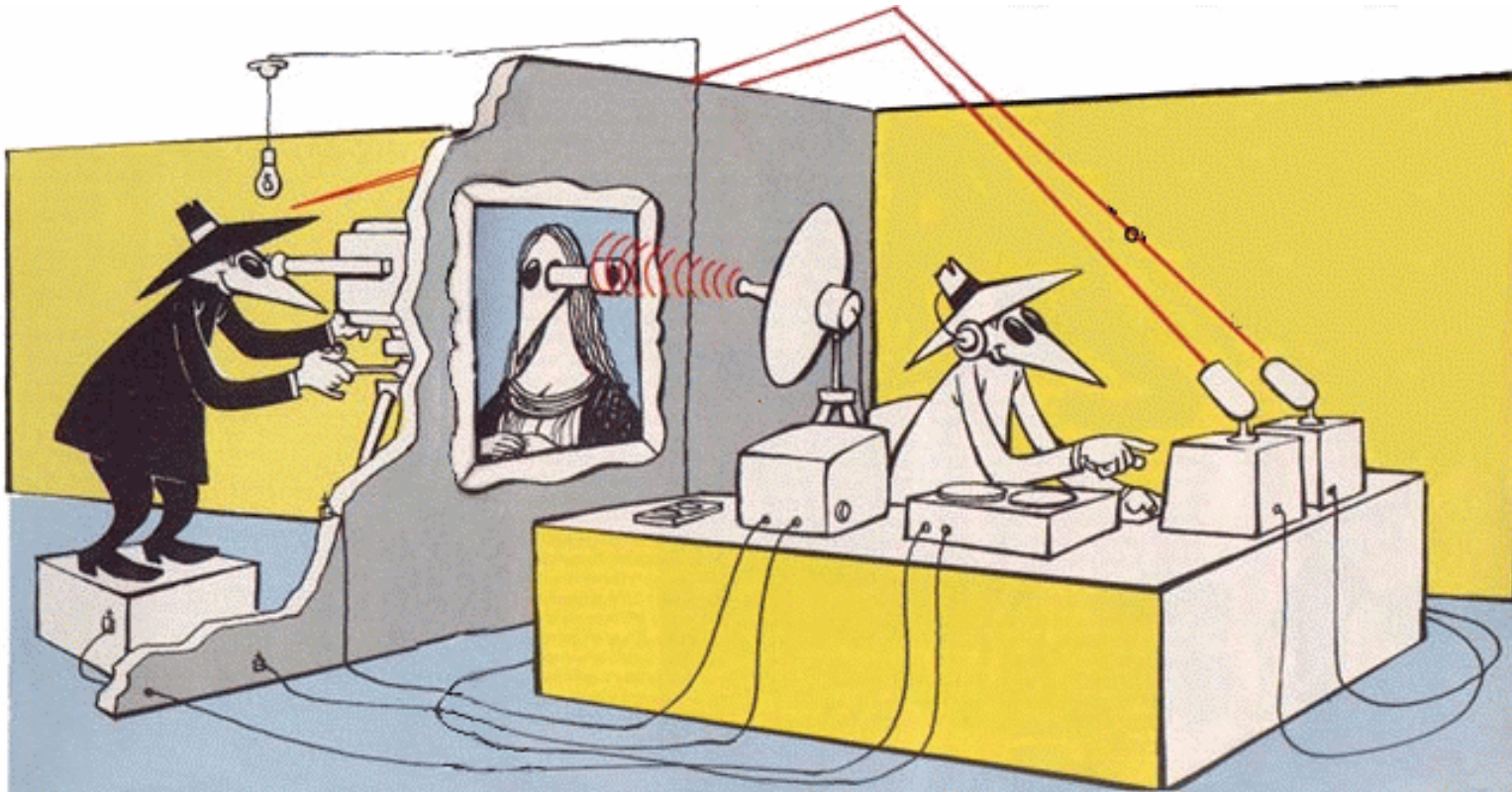


ORACLE[®]

Unlocking the Value of Standby DBs



DBA vs Developer





DBA vs Developer

- Don't outlaw features – none are 100% evil, none are 100% good. They are all tools
 - Views
 - Stored procedures
 - Any feature added after version 6
 - No feature can be used until its N versions old – this is software, not wine
- On the other hand, don't get sucked into 'feature obsession'
 - XML is so cool, lets do everything in XML



Developers

- Don't even try to work around the DBA
 - You'll be instantly labeled a subversive, loose cannon, dangerous
- Don't assume the DBA's are working against you when they say no
 - They may well have valid reasons for not wanting you to do nologging, or insert /*+ append */
- Do ask the DBA's to tell you why – it's only fair
- Do make sure you know what you are talking about
 - Use the scientific method
 - Or you risk losing credibility



DBAs

- DBA's – consider the developer as someone to teach/educate
- Everyone – back up your policy decisions and procedures with factual evidence.
 - “I heard it was slow”
 - “I've heard it is buggy”
 - “When your cache hit ratio is above 99%, you can put your feet up on the desk, well done”
 - “Export and Import all objects with more than one extent”



Everyone

- Back up your decisions and procedures with factual evidence.
 - “I heard it was slow”
 - “I’ve heard it is buggy”
 - “When your cache hit ratio is above 99%, you can put your feet up on the desk, well done”
 - “Export and Import all objects with more than one extent”