New York Oracle Users Group

Oracle9i New Features and Oracle8i Key Features

By Richard J. Niemiec, TUSC (www.tusc.com)
(Thanks: Sanjay, Dave, Jon, Randy, Bob, Ray, Ron, Ian & Pete)
Audience Knowledge

• Oracle8.0 Experience?
• Oracle8i Experience?
• Oracle9i Experience?
Overview

• Logging in … “sysdba”
• Memory & Key Init.ora Changes
• Indexing Arsenal
  – Function-based index
  – Index merges
  – Bitmap join index
  – Skip scan index
• Merge (Upsert) Command
• External Tables
Overview (cont.)

• Partitions
  - Range (Oracle8 which is version 8.0)
  - Hash (Oracle8i which is 8.1+)
  - Composite (Oracle8i)
  - List (Oracle9i which is 9.0.1+)

• Summary
Presentation
Goals/ Non-Goals

• Goals
  - Target Key Areas of Oracle9i
  - Mention some closely tied Oracle8i key areas
  - Target tips that are most useful

• Non-Goals
  - Learn ALL aspects of Oracle9i
Logging In - “sysdba”
Logging In

SQL> select * from v$version;

BANNER
Oracle9i Enterprise Edition Release 9.0.1.0.0 - Production
PL/SQL Release 9.0.1.0.0 - Production
TNS for Solaris: Version 9.0.1.0.0 - Production

Release 9.0.1.0.0

Version #  New Features #  Maint. #  Generic Patchset  Platform Patchset
$ sqlplus sys/ manager

ERROR:
ORA-28009: connection to sys should be as sysdba or sysoper

Enter user-name: sys as sysdba
Enter password:

Connected to: Oracle9i Enterprise Edition Release 9.0.1.0.0 - Production
With the Partitioning option
JServer Release 9.0.1.0.0 - Production

SQL >
The DBA Role does not include the sysdba or sysoper privileges... these are needed to start/shutdown. The security is OS Authentication or Password file (orapwd).

**SYSDBA (sys schema):**
- Shutdown, Startup, Archivelog, Recovery
- Alter Database (Open, Mount, Backup, change character set)

**SYSPER (public schema):**
- Shutdown, Startup, Archivelog, Recovery
- Alter Database (Open, Mount, Backup)
Logging In - “sysdba”

$ sqlplus " sys as sysdba"

SQL*Plus: Release 9.0.1.0.0 - Production on Wed Aug 8 10:31:59
(c) Copyright 2001 Oracle Corporation. All rights reserved.

Enter password: manager

Connected to:
Oracle9i Enterprise Edition Release 9.0.1.0.0 - Production
With the Partitioning option
JServer Release 9.0.1.0.0 – Production
SQL>
$ sqlplus sys/ manager as givemethesyntaxplease
Usage: SQLPLUS [ [<option>] [<logon>] [<start>] ]
   <logon> ::= <username>[/ <password>][@ <connect_string>] | / | / NO LOG
   <start> ::= @ <filename>[.<ext>] [<parameter> ...]
"-H" displays the SQL*Plus version banner and usage syntax
"-V" displays the SQL*Plus version banner
"-M <o>" uses HTML markup options <o>
"-R <n>" uses restricted mode <n>
"-S" uses silent mode

$
Logging In - “sysdba”

$ sqlplus " sys/ manager as sysdba"

SQL*Plus: Release 9.0.1.0.0 - Production on Wed Aug 8 10:32:42
(c) Copyright 2001 Oracle Corporation. All rights reserved.

Connected to:
Oracle9i Enterprise Edition Release 9.0.1.0.0 - Production
With the Partitioning option
JServer Release 9.0.1.0.0 - Production

SQL> !ps -ef | grep sqlplus
oracle  9425  9371  1 10:32:42 pts/ 3   0:00 sqlplus sys/ manager as sysdba
oracle  9427  9425  0 10:32:52 pts/ 3   0:00 grep sqlplus

www.tusc.com 12
Logging In - “sysdba”
O/ S Privilege is the key!

$ sqlplus "scott/ tiger as sysdba"

SQL*Plus: Release 8.1.7.0.0 - Production on Wed Aug 8 18:01:21 2001
(c) Copyright 2000 Oracle Corporation. All rights reserved.

Connected to:
Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production

SQL> shutdown abort
ORACLE instance shut down.

Also no more default passwords & locked accounts...
Memory Tuning Issues & Key Init.ora Parameters
Init.ora Parameters
Documented & Total

Oracle6  Oracle7  Oracle8i  Oracle9i

Documented
Total
The Init.ora over the years
Documented & Undocumented

<table>
<thead>
<tr>
<th>Version</th>
<th>Doc.</th>
<th>Undoc.</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>111</td>
<td>19</td>
<td>130</td>
</tr>
<tr>
<td>7</td>
<td>117</td>
<td>68</td>
<td>185</td>
</tr>
<tr>
<td>8</td>
<td>193</td>
<td>119</td>
<td>312</td>
</tr>
<tr>
<td>8i</td>
<td>203</td>
<td>301</td>
<td>504</td>
</tr>
<tr>
<td>9i</td>
<td>251</td>
<td>436</td>
<td>687</td>
</tr>
</tbody>
</table>
System Global Area - SGA

![Diagram showing SGA Size (M) and OS Memory for Oracle 5 through Oracle 8]
64-Bit advancement of Directly addressable memory

<table>
<thead>
<tr>
<th>Address Direct</th>
<th>Indirect/ Extended</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Bit:</td>
<td>16 (4 Bit)</td>
</tr>
<tr>
<td>8 Bit:</td>
<td>256 (8 Bit)</td>
</tr>
<tr>
<td>16 Bit:</td>
<td>65,536 (16 Bit)</td>
</tr>
<tr>
<td>32 Bit:</td>
<td>4,294,967,296 (32 Bit)</td>
</tr>
<tr>
<td>64 Bit:</td>
<td>18,446,744,073,709,551,616</td>
</tr>
</tbody>
</table>

When the hardware physically implements the theoretical possibilities of 64-Bit, things will dramatically change.... ...moving from 32 bit to 64 bit will be like moving from 4 bit to 32 bit or like moving from 1971 to 2000 overnight.
System Global Area - SGA

<table>
<thead>
<tr>
<th></th>
<th>Oracle6</th>
<th>Oracle7</th>
<th>Oracle8</th>
<th>Oracle9</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGA Size (M)</td>
<td></td>
<td></td>
<td></td>
<td>100000</td>
</tr>
<tr>
<td>OS Memory</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Init.ora Parameter Changes:
SGA_MAX_SIZE

- **Dynamic Memory allocation/resizing**
  - INIT.ora changes are immediate (no shutdown)
  - Set an SGA_MAX_SIZE:
    - _ksm_granule_size=4M (if <128M)
    - _ksm_granule_size=16M (if >=128M)
  - SGA_MAX_SIZE NOT dynamic!!
- Altered Sizes must be multiples of a granule size and may not exceed SGA_MAX_SIZE
- OEM to tune Database, App. Svr & OS
DB_CACHE_SIZE / DB_BLOCK_BUFFERS

- DB_BLOCK_BUFFERS was the way to set the cache... don’t use it even though it’s compatible.
- DB_BLOCK_BUFFERS = 0 (default) in Oracle9i
- _DB_BLOCK_BUFFERS = DB_CACHE_SIZE/DB_BLOCK_SIZE at startup
- If you set DB_BLOCK_BUFFERS... then
  - _DB_BLOCK_BUFFERS=DB_BLOCK_BUFFERS
  - DB_CACHE_SIZE/DB_BLOCK_SIZE=_DB_BLOCK_BUFFERS
Init.ora Values

```
select a.ksppinm, b.ksppstv1
from x$ksppi a, x$ksppcv b
where a.indx = b.indx
and (ksppinm like '%cache_size%'
  or ksppinm in ('shared_pool_size', 'large_pool_size',
                 'db_block_size', 'sga_max_size', 'db_block_buffers',
                 '_db_block_buffers'))
order by a.ksppinm;
```
# Init.ora Values

<table>
<thead>
<tr>
<th>KSPPINM</th>
<th>KSPPSTVL</th>
</tr>
</thead>
<tbody>
<tr>
<td>_db_block_buffers</td>
<td>2006</td>
</tr>
<tr>
<td>db_16k_cache_size</td>
<td>0</td>
</tr>
<tr>
<td>db_2k_cache_size</td>
<td>0</td>
</tr>
<tr>
<td>db_32k_cache_size</td>
<td>0</td>
</tr>
<tr>
<td>db_4k_cache_size</td>
<td>0</td>
</tr>
<tr>
<td>db_8k_cache_size</td>
<td>0</td>
</tr>
<tr>
<td>db_block_buffers</td>
<td>0</td>
</tr>
<tr>
<td>db_block_size</td>
<td>8192</td>
</tr>
<tr>
<td>db_cache_size</td>
<td>16777216</td>
</tr>
</tbody>
</table>
## Init.ora Values

<table>
<thead>
<tr>
<th>K S P P I N M</th>
<th>K S P P S T V L</th>
</tr>
</thead>
<tbody>
<tr>
<td>db_keep_cache_size</td>
<td>0</td>
</tr>
<tr>
<td>db_recycle_cache_size</td>
<td>0</td>
</tr>
<tr>
<td>large_pool_size</td>
<td>1048576</td>
</tr>
<tr>
<td>sga_max_size</td>
<td>185369592</td>
</tr>
<tr>
<td>shared_pool_size</td>
<td>67108864</td>
</tr>
</tbody>
</table>
DB_CACHE_SIZE

- **Upon Startup [K/M/G]:**
  - `DB_CACHE_SIZE=16M` (assume there’s enough in `SGA_MAX_SIZE` to accommodate any cache set here)
- `ALTER SYSTEM SET DB_CACHE_SIZE=32M`
  - Sets it to 32M (sets it to 33554432)
- `ALTER SYSTEM SET DB_CACHE_SIZE=17M`
  - Sets it to 32M (sets it to next highest multiple)
- `ALTER SYSTEM SET DB_CACHE_SIZE=33M`
  - Sets it to 48M (again sets it to next highest multiple of original size)
- `ALTER SYSTEM SET DB_CACHE_SIZE=2M`
  - Sets it to 16M (sets it to next highest/ but at minimum of initial size)
Initsid.ora & spfileSID.ora

- To create use: `CREATE SPFILE FROM PFILE;`
- `*.db_2k_cache_size=16777216`
- `*.db_block_size=8192`
- `*.db_cache_size=16777216`
- `*.db_name='dev2'`
- `*.shared_pool_size=33554432`
- `*.undo_management='AUTO'`
- `*.undo_tablespace='UNDOTBS'`

- This file changes dynamically as I change things so that the next STARTUP has all of my "alter system set ... " changes.
OS Memory Issues

alter system set db_cache_size=32M
*
ERROR at line 1:
ORA-02097: parameter cannot be modified specified value is invalid
ORA-00384: Insufficient memory to grow cache

Elapsed: 00:00:00.00
SQL> !oerr ora 384
00384, 00000, "Insufficient memory to grow cache"
// *Cause: The system could not allocate sufficient memory to grow the
// cache to the specified size.
// *Action: Attempt a smaller increase in the value of the parameter.
OS Memory Issues

shared_pool_size  67108864 (64M)

SQL> alter system set shared_pool_size = 80M;
shared_pool_size  83886080 (84M)

Shutdown and startup
(The SGA and the SHARED_POOL_SIZE both up by 16M)

SQL> alter system set shared_pool_size = 32M;
shared_pool_size  33554432 (32M)
OS Memory Issues

SQL> alter system set shared_pool_size = 16M;

<table>
<thead>
<tr>
<th>Time</th>
<th>%usr</th>
<th>%sys</th>
<th>%wio</th>
<th>%idle</th>
</tr>
</thead>
<tbody>
<tr>
<td>21:09:03</td>
<td>99</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21:09:05</td>
<td>99</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21:09:07</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Average 100 0 0 0

(shutdown abort was next... flush first didn’t help)
Multiple Block Sizes

Must be a power of 2. Can’t set `db_nK_cache_size` where `db_block_size = nK`. You only set the others.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>db_2k_cache_size</code></td>
<td>0</td>
</tr>
<tr>
<td><code>db_4k_cache_size</code></td>
<td>0</td>
</tr>
<tr>
<td><code>db_8k_cache_size</code></td>
<td>0</td>
</tr>
<tr>
<td><code>db_8k_cache_size</code> (must stay as 0 if 8K as below)</td>
<td>0</td>
</tr>
<tr>
<td><code>db_16k_cache_size</code></td>
<td>0</td>
</tr>
<tr>
<td><code>db_32k_cache_size</code></td>
<td>0</td>
</tr>
<tr>
<td><code>db_block_size</code></td>
<td>8192</td>
</tr>
<tr>
<td><code>db_cache_size</code></td>
<td>16777216</td>
</tr>
</tbody>
</table>
Multiple Block Sizes

alter system set db_2k_cache_size=4M;
(multiples of original granule size... 16M in this case)

```
db_2k_cache_size               16777216 (16M)
```

Create tablespace rich_block_2k
Datafile ‘rich_block_2k.dbf’
Size 100M blocksize 2k;
alter system set db_32k_cache_size=4m;
SQL>
alter system set db_32k_cache_size=4m
*

ERROR at line 1:
ORA-02097: parameter cannot be modified because specified value is
ORA-00382: 32768 not a valid block size, valid range [2048..16384]
Large Block Size - Bottlenecks in UPDATE

- With larger blocks, many rows reside in one block
- Updates cause many before images to be created
  - Readers with read consistency suffer
- To ensure row level locking, transaction areas are allocated in every block
  - If there is not enough free space for TRX info, the lock escalates to block level lock
- Oracle9i multiple block sizes will allow large size for data warehouse and small block for transactional.
Use Bind Variables / Latch Issue

• If you don’t use bind variables in your application code you usually end up with latch contention with the shared_pool and library_caches latches. Oracle9i promises a lock for every block.

• This latch wait time can be reduced by changing just the top couple of executed statements that were using literal SQL instead of bind variables.

• Oracle 8i Release 2 (8.1.6) has an auto-conversion of literals into bind variables… Oracle9i extends this slightly…

```
CURSOR_SHARING = FORCE
```

(Default is EXACT; 8iR2)
Cursor Sharing - Not Sharing

declare
temp varchar2(30);
begin
   SELECT ename INTO temp
   FROM rich
   WHERE ename = 'SMITH';
   SELECT ename INTO temp
   FROM rich
   WHERE ename = 'JONES';
end;
Cursor Sharing - Not Sharing

```
SELECT SQL_TEXT
FROM V$SQLAREA
WHERE SQL_TEXT LIKE 'SELECT ENAME%';

SQL_TEXT
-------------------------------------
SELECT ENAME   FROM RICH  WHERE ENAME = 'JONES'
SELECT ENAME   FROM RICH  WHERE ENAME = 'SMITH'

Note: PL/SQL converted to uppercase AND trims spaces AND carriage returns are trimmed.
```
On 8.1.5 or lower

If you have unfixable bind variable issues in v$sqlarea:

• A shared pool of 50M seems to work well
• A shared pool of 100M seems to bog down when x$ksmsp exceeds around 200,000 records.
• A shared pool of over 200M comes to a standstill when x$ksmsp gets to around 580,000 records.
  – The CPU goes to 0-2% with each query
  – Heavy duty I/O

Keep the Shared_pool_size at 100M or lower (50M if possible).
Cursor Sharing - 8.1.6 / 9i

If v$sqlarea looks like this:
select empno from rich778 where empno = 451572
select empno from rich778 where empno = 451573
select empno from rich778 where empno = 451574
select empno from rich778 where empno = 451575
select empno from rich778 where empno = 451576

Use cursor_sharing=force (sqlarea goes to this):
select empno from rich778 where empno = :SYS_B_0
Oracle 9i - Cursor_sharing

Oracle9i Enterprise Edition Release 9.0.1.0.0 - Production
(SIMILAR takes into account the statistics as well...)

SQL> alter session set cursor_sharing=SIMILAR;
System altered.

Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production
SQL> alter session set cursor_sharing=SIMILAR;

ERROR: ORA-00096: invalid value SIMILAR for parameter
cursor_sharing, must be from among EXACT, FORCE
Important Init.ora’s

- Setting `db_file_multiblock_read_count` high helps read more in one I/O... but, too high can cause Oracle’s optimizer to do MORE full table scans.

Fix/Override as of 8.05+ (force index use!):

- `OPTIMIZER_INDEX_COST_ADJ = 10 (1-14)`
- If an index-based operation, Oracle will adjust the cost by a factor of `OPTIMIZER_INDEX_COST_ADJ/100` before choosing the lowest cost option.
Important V8 Init.ora’s

**DB_KEEP_CACHE_SIZE[9i]** (BUFFER_POOL_KEEP) - How many buffers to have for pinned objects that you need - V8+ ONLY

**DB_RECYCLE_CACHE_SIZE [9i]** (BUFFER_POOL_RECYCLE)  
- How many buffers to have for new stuff that will get pushed out - V8+ ONLY

```
Create table states (state_abbrev varchar2(2), desc varchar2(25))
Storage (buffer_pool keep);
Alter table state_list storage (buffer_pool recycle);
```

**DB_BLOCK_HASH_BUCKETS** – Set to next prime number higher than 2 x DB_BLOCK_BUFFERS (does correct in 9i)
Important Oracle9i Init.ora’s

In 8i:  _UNNEST_SUBQUERY= FALSE (Default)
In 9i:  _UNNEST_SUBQUERY= TRUE (Default)

• Fix potential Oracle9i performance problems by setting this parameter back to False. This will cause the same plans as in Oracle 8.1.7 to be used.

_UNNEST_SUBQUERY= FALSE (IN THE INIT.ORA)   ...OR

SQL> Alter session set _unnest_subquery=false;
(Thanks Jon V.)
LOOK OUT! (Works in 8+/9i)

```sql
select name, value, isdefault, isses_modifiable, issys_modifiable
from v$parameter
where issys_modifiable <> 'FALSE'
  or isses_modifiable <> 'FALSE'
order by name;
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>VALUE</th>
<th>ISDEFAULT</th>
<th>ISSSES</th>
<th>ISSYS</th>
</tr>
</thead>
<tbody>
<tr>
<td>optimizer_mode</td>
<td>CHOOSE</td>
<td>TRUE</td>
<td>TRUE</td>
<td>FALSE</td>
</tr>
<tr>
<td>sort_area_size</td>
<td>65536</td>
<td>TRUE</td>
<td>TRUE</td>
<td>DEFERRED</td>
</tr>
</tbody>
</table>

Conn scott/tiger
Alter session set sort_area_size=1000000000;
Session altered.
V$ Scripts

PARAMETERS

• Undocumented Parameters (89-Be Careful)

```
select indx, ksppinm
from x$ksppi
where substr(ksppinm,1,1) = '_'
```

<table>
<thead>
<tr>
<th>INDEX</th>
<th>KSPPINM</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>_debug_sga</td>
</tr>
<tr>
<td>32</td>
<td>_trace_buffers_per_process</td>
</tr>
<tr>
<td>33</td>
<td>_trace_block_size</td>
</tr>
<tr>
<td>34</td>
<td>_trace_archive_start</td>
</tr>
<tr>
<td>38</td>
<td>_trace_archive_dest</td>
</tr>
<tr>
<td>39</td>
<td>_trace_file_size</td>
</tr>
<tr>
<td>94</td>
<td>_log_buffers_debug</td>
</tr>
<tr>
<td>99</td>
<td>_disable_logging</td>
</tr>
<tr>
<td>115</td>
<td>_allow_resetlogs_corruption</td>
</tr>
<tr>
<td>126</td>
<td>_log_space_errors</td>
</tr>
<tr>
<td>135</td>
<td>_corrupted_rollback_segments</td>
</tr>
<tr>
<td>179</td>
<td>_init_sql_file</td>
</tr>
<tr>
<td>239</td>
<td>_oracle_trace_events</td>
</tr>
<tr>
<td>240</td>
<td>_oracle_trace_facility_version</td>
</tr>
</tbody>
</table>
# V$ Views over the years

<table>
<thead>
<tr>
<th>Version</th>
<th>V$ Views</th>
<th>X$ Tables</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>23</td>
<td>? (35)</td>
</tr>
<tr>
<td>7</td>
<td>72</td>
<td>126</td>
</tr>
<tr>
<td>8</td>
<td>132</td>
<td>200</td>
</tr>
<tr>
<td>8i</td>
<td>185</td>
<td>271</td>
</tr>
<tr>
<td>9i</td>
<td>227</td>
<td>352</td>
</tr>
</tbody>
</table>
Indexing Arsenal
- Function-Based Indexes - 8i
- Index Merges - 8i
- Bitmap Join Index - 9i
- Skip Scan Index - 9i
The 80/20 Rule is Back ...

- When "Optimizer" finds a query to retrieve less than 20% of the rows, the optimizer will USUALLY choose to drive the query with an index if one exists.
The 80/20 Rule (Short Answer)

• If there are a small number of rows returned by a query... **Use an index**

• If there are a Large number of rows returned by a query... **Do NOT use an index** or suppress a current index.

• Depends on MANY THINGS:
  - Type of index (bitmap is 60/40 rule)
  - Data distribution / `db_file_multiblock_read_count`
  - Clustering factor
Oracle8i Key Index Features

The Function-based Index
• Allows you to index a function
• Saves you from suppressing indexes with functions
• Requires the CBO (Cost-based Optimizer)

The Index Merge
• Can merge 2 indexes instead of a table access
• Requires the CBO
Oracle 8.1
Function Based Index

```
select count(*)
from sample
where ratio(balance,limit) > .5;
```

Elapsed time: 20.1 minutes
Oracle 8.1
Function Based Index

cREATE INDEX ration_idx ON
sample(ratio(balance, limit));

SELECT count(*)
FROM sample
WHERE ratio(balance, limit) > .5;

Elapse time: 7 seconds!!!
Oracle 8.1
Index Merge

- Merge Separate Indexes / Avoids going to the table
- Only using CBO (Cost Based Optimizer)

The following are statistics based on 1,000,000 records. The table is 210 MB.

```sql
create index year_idx on test2 (year);
create index state_idx on test2 (state);
```
Oracle 8.1
Index Merge

```
select /*+ rule index(test2) */ state, year
from test2
where year = '1972'
and state = MA
```

SELECT STATEMENT Optimizer=HINT: RULE
TABLE ACCESS (BY INDEX ROWID) OF 'TEST2'
INDEX (RANGE SCAN) OF 'STATE_IDX' (NON-UNIQUE)

Elapsed time 23.50 seconds
select /*+ index_join(test2 year_idx state_idx) */

    state, year
from test2
where year = '1972'
and state = MA

SELECT STATEMENT Optimizer=CHOOSE
    VIEW OF 'index$_join$_001'
    HASH JOIN
    INDEX (RANGE SCAN) OF 'YEAR_IDX' (NON-UNIQUE)
    INDEX (RANGE SCAN) OF 'STATE_IDX' (NON-UNIQUE)

Elapsed time 4.76 seconds
Oracle9i Key Index Features

Bitmap Join Index
• Allows a single index to span two tables
• Requires the use of a unique constraint
• Can give substantial performance gains

Skip Scan Index
• Allows the use of the 2\textsuperscript{nd} part of an index
• Can also be a source of problems if not careful
Bitmap Join Index:

CREATE TABLE EMP1
 AS SELECT * FROM SCOTT.EMP;

CREATE TABLE DEPT1
 AS SELECT * FROM SCOTT.DEPT;

ALTER TABLE DEPT1
 ADD CONSTRAINT DEPT_CONSTR1 UNIQUE (DEPTNO);
Bitmap Join Index
Small Tables... .

CREATE BITMAP INDEX EMPDEPT_IDX
ON EMP1(DEPT1.DEPTNO)
FROM EMP1, DEPT1
WHERE EMP1.DEPTNO = DEPT1.DEPTNO
/

Bitmap Join Index
Small Tables...

```sql
SELECT /*+ INDEX(EMP1 EMPDEPT_IDX) */ COUNT(*)
FROM EMP1, DEPT1
WHERE EMP1.DEPTNO = DEPT1.DEPTNO;

COUNT(*)
-----------------
    14

Elapsed: 00:00:00.67
```
Bitmap Join Index

Execution Plan

0  SELECT STATEMENT Optimizer=CHOOSE
1  0  SORT (AGGREGATE)
2  1  BITMAP CONVERSION (COUNT)
3  2  BITMAP INDEX (FULL SCAN) OF 'EMPDEPT_IDX'
EMP5/EMP6 have 2 million rows each with empno indexes on them.

```sql
alter table emp5
add constraint emp5_constr unique (empno);
```

```sql
create bitmap index emp5_j6
on emp6(emp5.empno)
from emp5,emp6
where emp5.empno=emp6.empno;
```

Index created.
Elapsed: 00:02:29.91
1  select count(*)
  2  from emp5, emp6
  3* where emp5.empno=emp6.empno

COUNT(*)
---------
2005007

Elapsed: 00:01:07.18
Execution Plan

0  SELECT STATEMENT Optimizer=CHOOSE
1  0  SORT (AGGREGATE)
2  1  NESTED LOOPS
3  2  TABLE ACCESS (FULL) OF 'EMP6'
4  2  INDEX (RANGE SCAN) OF 'EMP5I_EMPNO' (NON-UNIQUE)

Statistics

6026820 consistent gets
7760 physical reads
FORCE THE USE OF THE BITMAP JOIN INDEX:

1. select /*+ index(emp6 emp5_j6) */ count(*)
2. from emp5, emp6
3. where emp5.empno = emp6.empno

COUNT(*)

2005007

Elapsed: 00:00:00.87
Bitmap Join Index - 10,000 Times Faster

Execution Plan

<table>
<thead>
<tr>
<th>Step</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SELECT STATEMENT Optimizer=CHOOSE</td>
</tr>
<tr>
<td>1</td>
<td>SORT (AGGREGATE)</td>
</tr>
<tr>
<td>2</td>
<td>BITMAP CONVERSION (COUNT)</td>
</tr>
<tr>
<td>3</td>
<td>BITMAP INDEX (FULL SCAN) OF 'EMP5_J6'</td>
</tr>
</tbody>
</table>

Statistics

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consistent gets</td>
<td>970</td>
</tr>
<tr>
<td>Physical reads</td>
<td>967</td>
</tr>
</tbody>
</table>
Choosing a Driving Table

**Hint:** ORDERED

Using the Ordered Hint will force the optimizer to use the table listed first in the FROM clause as the driving table. If possible...
Basic Join Theory...

\[
\text{select count(*) from emp5 where empno=7900;}
\]

\underline{COUNT(*)}
1

\[
\text{select count(*) from emp6 where empno=7900;}
\]

\underline{COUNT(*)}
143214
Large result first
(emp6 = 143,214 Rows)

```
1  select /*+ ordered */ count(*)
2  from emp6, emp5
3  where emp5.empno=emp6.empno
4  and emp6.empno=7900
5* and emp5.empno=7900;

_COUNT(*)
143214

Elapsed: 00:00:05.28
```
Large result first  
(emp6 = 143,214 Rows)  

Execution Plan  
----------------------------------------------------------
0      SELECT STATEMENT Optimizer=CHOOSE
1    0   SORT (AGGREGATE)
2    1     NESTED LOOPS (Cost=1 Card=1 Bytes=26)
3    2       INDEX (RANGE SCAN) OF 'EMP6I_EMPNO'
4    2       INDEX (RANGE SCAN) OF 'EMP5I_EMPNO'

Statistics  
----------------------------------------------------------
429925  consistent gets
0 physical reads
Small result first (emp5 = 1 Row)

1. select /*+ ordered*/ count(*)
   2. from emp5, emp6
   3. where emp5.empno=emp6.empno
   4. and emp6.empno=7900
   5. and emp5.empno=7900;

COUNT(*)

-----------

143214

Elapsed: 00:00:00.77
Small result first (emp5 = 1 Row)

Execution Plan

0      SELECT STATEMENT Optimizer=CHOOSE
1    0   SORT (AGGREGATE)
2    1     NESTED LOOPS (Cost=2 Card=1 Bytes=26)
3    2       INDEX (RANGE SCAN) OF 'EMP5I_EMPNO'
4    2       INDEX (RANGE SCAN) OF 'EMP6I_EMPNO'

Statistics

286  consistent gets
0   physical reads
Analyze the table... Optimizer decides to use the join index...

analyze the table and leave it to the optimizer...

```sql
1  select count(*)
2  from emp5, emp6
3  where emp5.empno=emp6.empno
4  and emp6.empno=7900
5* and emp5.empno=7900;
```

**COUNT(*)**

143214

Elapsed: 00:00:06.60
Analyze the table... optimizer decides to use the join index

Execution Plan
----------------------------------------------------------
0 SELECT STATEMENT Optimizer=CHOOSE
1 0  SORT (AGGREGATE)
2 1  TABLE ACCESS (BY INDEX ROWID) OF 'EMP6'
3 2  BITMAP CONVERSION (TO ROWIDS)
4 3  BITMAP INDEX (SINGLE VALUE) OF 'EMP5_J6'

Statistics
11867 consistent gets
10501 physical reads
Better NOT to use Bitmap Join Index
Force emp5 first instead!!

1. select /*+ ordered */ count(*)
2. from emp5, emp6
3. where emp5.empno=emp6.empno
4. and emp6.empno=7900
5. and emp5.empno=7900;

COUNT(*)
143214

Elapsed: 00:00:00.77
Bitmap Join Index - 1000 x faster
Better to force the emp5 index...

Execution Plan

----------------------------------------------------------

0  SELECT STATEMENT Optimizer=CHOOSE
1  0  SORT (AGGREGATE)
2  1  NESTED LOOPS (Cost=302 Card=10230 Bytes=81840)
3  2  INDEX (RANGE SCAN) OF 'EMP5I_EMPNO'
4  2  INDEX (RANGE SCAN) OF 'EMP6I_EMPNO'

Statistics
286  consistent gets
0  physical reads
Oracle9i Skip/Scan Index

The Skip/Scan Index:
- Allows you to scan the index instead of the table
- Saves you from doing a full table scan
- Create index year_state_idx on test2(year, state);
Oracle9i Skip/Scan Index

- Skips the first part of the index (YEAR)
- Scans the second part of the index (STATE)

SELECT COUNT(*)
FROM TEST2
WHERE STATE = ‘AL’
### Oracle9i Skip/ Scan Index

```sql
SQL> desc emp5
Name | Null? | Type
------------------------- -------- ------------------
EMPNO |        | NUMBER(15)       
ENAME |        | VARCHAR2(10)     
JOB   |        | VARCHAR2(9)      
MGR   |        | NUMBER(4)        
HIREDATE |    | DATE             
SAL   |        | NUMBER(7,2)      
COMM  |        | NUMBER(7,2)      
DEPTNO|        | NUMBER(2)        
```
Oracle9i Skip/Scan Index

SQL> create index skip1 on emp5(job,empno);

create index skip1 on emp5(job,empno)
  *
ERROR at line 1:
ORA-01652: unable to extend temp segment by 128 in tablespace TEMP

SQL> alter session set sort_area_size=300000000;
Session altered

SQL> create index skip1 on emp5(job,empno);
Index created.
Oracle9i Skip/Scan Index

```
select count(*)
from emp5
where empno=7900;
Elapsed: 00:00:00.12 (Result is a single row...not displayed)
```

**Execution Plan**

0  SELECT STATEMENT Optimizer=CHOOSE (Cost=2 Card=1 Bytes=5)
1  0  SORT (AGGREGATE)
2  1  INDEX (RANGE SCAN) OF 'EMP5I_EMPNO' (NON-UNIQUE)

**Statistics**

3  consistent gets
0  physical reads
Oracle9i Skip/Scan Index

```sql
SELECT /*+ no_index(emp5 emp5i_empno) */ COUNT(*)
FROM emp5
WHERE empno = 7900;
Elapsed: 00:00:03.13 (Result is a single row...not displayed)
```

**Execution Plan**

0. SELECT STATEMENT Optimizer=CHOOSE (Cost=4 Card=1 Bytes=5)
1. 0. SORT (AGGREGATE)
2. 1. INDEX (FAST FULL SCAN) OF 'SKIP1' (NON-UNIQUE)

**Statistics**

6826 consistent gets
6819 physical reads
Oracle9i Skip/Scan Index

```
select /*+ index(emp5 skip1) */ count(*)
from emp5
where empno = 7900;
Elapsed: 00:00:00.56

Execution Plan
0  SELECT STATEMENT Optimizer=CHOOSE (Cost=6 Card=1 Bytes=5)
1   0  SORT (AGGREGATE)
2   1  INDEX (SKIP SCAN) OF 'SKIP1' (NON-UNIQUE)

Statistics
21  consistent gets
17  physical reads
```
Oracle9i Skip/ Scan Index

```
select /*+ full(emp5) */ count(*)
from emp5
where empno = 7900;
Elapsed: 00:00:05.42
```

**Execution Plan**

1. SELECT STATEMENT Optimizer=CHOOSE (Cost=1856 Card=1 Bytes=5)
2. 0  TABLE ACCESS (FULL) OF 'EMP5' (Cost=1856 Card=1 Bytes=5)

**Statistics**

- 12231 consistent gets
- 12227 physical reads
Merge (Upsert)
Merge (Upsert):

The new MERGE command does the following:

- **UPDATE** the record if it already exists
- **INSERT** the record if it doesn’t exist
- Often known as an “upsert,” the actual command is MERGE
Merge (Upsert):

SELECT * 
FROM DEPTBIG;

<table>
<thead>
<tr>
<th>DEPTNO</th>
<th>DNAME</th>
<th>LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>40</td>
<td>RESEARCH</td>
<td>BOSTON</td>
</tr>
</tbody>
</table>
Merge (Upsert):

```
SELECT * 
FROM DEPT2;
```

<table>
<thead>
<tr>
<th>DEPTNO</th>
<th>DNAME</th>
<th>LOC</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>OPERATIONS</td>
<td>CHICAGO</td>
<td>(to update)</td>
</tr>
<tr>
<td>50</td>
<td>OPERATIONS</td>
<td>CHICAGO</td>
<td>(to insert)</td>
</tr>
</tbody>
</table>
Merge (Upsert):

merge into deptbig a
using (select deptno, dname, loc from dept2) b
on (a.deptno = b.deptno)
when matched then
update set a.loc = b.loc
when not matched then
insert (a.deptno, a.dname, a.loc)
values (b.deptno, b.dname, b.loc);

2 rows merged.
**Merge (Upsert):**

```sql
SELECT * 
FROM DEPTBIG;
```

<table>
<thead>
<tr>
<th>DEPTNO</th>
<th>DNAME</th>
<th>LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>40</td>
<td>RESEARCH</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>50</td>
<td>OPERATIONS</td>
<td>CHICAGO</td>
</tr>
</tbody>
</table>
Merge (Upsert):

merge into deptbig a
using (select deptno, dname, loc from dept2) b
on (a.deptno = b.deptno)
when matched then
update set a.loc = b.loc,
a.dname = b.dname
when not matched then
insert (a.deptno, a.dname, a.loc)
values (b.deptno, b.dname, b.loc);

2 rows merged.
# Merge (Upsert):

**SELECT** * 
**FROM** DEPTBIG;

<table>
<thead>
<tr>
<th>DEPTNO</th>
<th>DNAME</th>
<th>LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>ACCOUNTING</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>20</td>
<td>RESEARCH</td>
<td>DALLAS</td>
</tr>
<tr>
<td>30</td>
<td>SALES</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>40</td>
<td>OPERATIONS</td>
<td>CHICAGO</td>
</tr>
<tr>
<td>50</td>
<td>OPERATIONS</td>
<td>CHICAGO</td>
</tr>
</tbody>
</table>

*new record* (was updated)
Partitioning and Partition-wise Joins
Partitioning

• Tables can be split into thousands of pieces.
  • Using partition tables and indexes
  • Only a subset of the data is queried
  • All of the data COULD be queried
  • Leads to enhanced performance of large tables
  • Partitioned views was the precursor to this
  • Data Warehouses can be tuned greatly!
  • Re-orgs can be done on a partition level

• Examples on the following slide.
CREATE TABLE DEPT
(DEPTNO NUMBER(2),
DEPT_NAME VARCHAR2(30))
PARTITION BY RANGE(DEPTNO)
(PARTITION D1 VALUES LESS THAN (10) TABLESPACE DEPT1,
PARTITION D2 VALUES LESS THAN (20) TABLESPACE DEPT2,
PARTITION D3 VALUES LESS THAN (MAXVALUE) TABLESPACE DEPT3);

INSERT INTO DEPT VALUES (1, ‘DEPT 1’);
INSERT INTO DEPT VALUES (7, ‘DEPT 7’);
INSERT INTO DEPT VALUES (10, ‘DEPT 10’);
INSERT INTO DEPT VALUES (15, ‘DEPT 15’);
INSERT INTO DEPT VALUES (22, ‘DEPT 22’);
Range Partitioning (Multi-Column)

create table cust_sales (  
acct_no number(5),  
cust_name char(30),  
sale_day integer not null,  
sale_mth integer not null,  
sale_yr integer not null)  
partition by range (sale_yr, sale_mth, sale_day)  
  partition cust_sales_q1 values less than (1998, 04, 01) tablespace users1,  
  partition cust_sales_q2 values less than (1998, 07, 01) tablespace users2,  
  partition cust_sales_q3 values less than (1998, 10, 01) tablespace users3,  
  partition cust_sales_q4 values less than (1999, 01, 01) tablespace users4);
Hash Partitioning (Multi-Column)

create table cust_sales_hash(
cust_no number(5),
cust_name char(30),
sale_day integer not null,
sale_mth integer not null,
sale_yr integer not null)
partition by hash (cust_no)
partitions 4
store in (users1, users2, users3, users4);
CREATE TABLE test5 (data_item INTEGER, length_of_item INTEGER, 
storage_type VARCHAR(30), owning_dept NUMBER, 
storage_date DATE) PARTITION BY RANGE (storage_date) SUBPARTITION BY 
HASH(data_item) SUBPARTITIONS 4 
STORE IN (data_tbs1, data_tbs2, 
data_tbs3, data_tbs4) (PARTITION q1_1999 VALUES LESS 
THAN (TO_DATE('01-apr-1999', 'dd-mon-yyyy')), PARTITION q2_1999 
VALUES LESS THAN (TO_DATE('01-jul-1999', 'dd-mon-yyyy')), 
PARTITION q3_1999 VALUES LESS THAN (TO_DATE('01-oct-1999', 'dd-mon-yyyy')) 
(SUBPARTITION q3_1999_s1 TABLESPACE data_tbs1, 
SUBPARTITION q3_1999_s2 TABLESPACE data_tbs2), 
PARTITION q4_1999 VALUES LESS THAN (TO_DATE('01-jan-2000', 'dd-mon-yyyy')) 
SUBPARTITIONS 8 
STORE IN (q4_tbs1, q4_tbs2, q4_tbs3, q4_tbs4, 
q4_tbs5, q4_tbs6, q4_tbs7, q4_tbs8), PARTITION q1_2000 
VALUES LESS THAN (TO_DATE('01-apr-2000', 'dd-mon-yyyy'));
Partition Joins - Oracle8i

- Prior to 8i, Oracle would join partitioned tables by joining the entire tables together.
- In Oracle9i, tables can now be joined by partitions if they are equi-partitioned tables (Tables partitioned using the same partition key and same partition break points)
create table dept_part
(deptno number(2),
dname varchar2(14),
loc varchar2(13))
partition by list (dname)
(partition d1_east values ('BOSTON', 'NEW YORK'),
partition d2_west values ('SAN FRANCISCO', 'LOS ANGELES'),
partition d3_south values ('ATLANTA', 'DALLAS'),
partition d4_north values ('CHICAGO', 'DETROIT'));

Table created
External Tables

SQL> spool emp4.dat
  1  select empno||','||ename ||','|| job||','|| deptno
  2* from scott.emp;

empout file contents:
7369,SMITH,CLERK,20
7499,ALLEN,SALES MAN,30
7521,WARD,SALES MAN,30
7566,JONES,MANAGER,20
7654,MARTIN,SALES MAN,30
...
14 rows selected
External Tables

createdirectory rich_new as '/u01/home/oracle/rich';
Directory created.

1 create table emp_external4
   (empno char(4),
    ename char(10),
    job char(9),
    deptno char(2))
organization external
7 (type oracle_loader)
8 default directory rich_new
... more
External Tables

access parameters
(fields terminated by ',
(empno,
ename,
job,
deptno))
* location ('emp4.dat'));

Table created.
Elapsed: 00:00:00.12
## External Tables

```sql
SQL> desc emp_external4

<table>
<thead>
<tr>
<th>Name</th>
<th>Null?</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPNO</td>
<td></td>
<td>CHAR(4)</td>
</tr>
<tr>
<td>ENAME</td>
<td></td>
<td>CHAR(10)</td>
</tr>
<tr>
<td>JOB</td>
<td></td>
<td>CHAR(9)</td>
</tr>
<tr>
<td>DEPTNO</td>
<td></td>
<td>CHAR(2)</td>
</tr>
</tbody>
</table>
```
External Tables - Errors early on...

SQL> select * from emp_external
   2 /
select * from emp_external
*
ERROR at line 1:
ORA-29913: error in executing ODCIEXTTABLEOPEN callout
ORA-29400: data cartridge error
KUP-00554: error encountered while parsing input commands
KUP-01005: syntax error: found "badfile": expecting one of: "encoded, exit, (,
ltrim, lrtrim, ldrtrim, missing notrim optionally, rtrim"
KUP-01007: at line 2 column 2
ORA-06512: at "SYS.ORACLE_LOADER", line 14
ORA-06512: at line 1
### External Tables - SUCCESS

```sql
select * from emp_external4;
```

<table>
<thead>
<tr>
<th>EMPNO</th>
<th>ENAME</th>
<th>JOB</th>
<th>DEPTNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>7369</td>
<td>SMITH</td>
<td>CLERK</td>
<td>20</td>
</tr>
<tr>
<td>7499</td>
<td>ALLEN</td>
<td>SALESMAN</td>
<td>30</td>
</tr>
<tr>
<td>7521</td>
<td>WARD</td>
<td>SALESMAN</td>
<td>30</td>
</tr>
</tbody>
</table>

....

**Execution Plan**

0  SELECT STATEMENT Optimizer=CHOOSE

1 0  EXTERNAL TABLE ACCESS (FULL) OF 'EMP_EXTERNAL4'
External Tables -
DML is not Allowed

INSERT INTO EMP_EXTERNAL...
    *
ERROR at line 1:
ORA-30657: operation not supported on external organized table

CREATE INDEX emp_ei ON emp_external(deptno)
    *
ERROR at line 1:
ORA-30657: operation not supported on external organized table
External Tables - Bigger SUCCESS

$ wc -l emp4.dat
   200020  200020  4400400 emp4.dat

$ ls -l emp4.dat
-rw-r-xr-x  1 oracle   oinstall  4400400 Aug  9  06:31 emp4.dat

1* select count(*) from emp_external4;

COUNT(*)
----------
 200020

Elapsed: 00:00:00.63
External Tables - Bigger SUCCESS

1* select count(*) from emp_external4
2  where empno=7900;

COUNT(*)
  20

Elapsed: 00:00:00.82
Execution Plan

----------------------------------------------------------
0  SELECT STATEMENT  Optimizer=CHOOSE
1  0  SORT (AGGREGATE)
2  1  EXTERNAL TABLE ACCESS (FULL) OF 'EMP_EXTERNAL4'
External Tables - Bigger SUCCESS

```sql
1  create table emp_external5
  2  (empno char(4),
  3  ename char(10),
  4  job char(9),
  5  deptno char(2))
  6  organization external
...
15* location ('emp5.dat'));

SQL> select count(*) from emp_external5;
COUNT(*)
200020
```
External Tables - SUCCESS
Joining 4M+ rows in under 3 Secs.

1. select a.empno, b.job, a.job
2. from emp_external4 a, emp_external5 b
3. where a.empno = b.empno
4. and a.empno = 7900
5. and b.empno = 7900;

400 rows selected.

Elapsed: 00:00:02.46
External Tables - SUCCESS
Joining 4M+ rows in under 3 Secs.

Execution Plan

----------------------------------------------------------
0  SELECT STATEMENT Optimizer=CHOOSE
1  0  MERGE JOIN
2  1  SORT (JOIN)
3  2  EXTERNAL TABLE ACCESS (FULL) OF 'EMPEXTERNAL5'
4  1  SORT (JOIN)
5  4  EXTERNAL TABLE ACCESS (FULL) OF 'EMPEXTERNAL4'
External Tables - SUCCESS
Can even use a HINT!

SQL> 1
  1  select /*+ use_hash(a) */ a.empno, b.job, a.job
  2  from emp_external4 a, emp_external5 b
  3  where a.empno = b.empno
  4  and a.empno = 7900
  5* and b.empno = 7900;

400 rows selected.

Elapsed: 00:00:02.65
External Tables - SUCCESS
Can even use a HINT!

Execution Plan

0  SELECT STATEMENT Optimizer=CHOOSE
   (Cost=33 Card=81 Bytes=275  4)
1  0  HASH JOIN (Cost=33 Card=81 Bytes=2754)
2  1  EXTERNAL TABLE ACCESS (FULL) OF 'EMP_EXTERNAL4'
3  1  EXTERNAL TABLE ACCESS (FULL) OF 'EMP_EXTERNAL5'
Overview

• Logging in … “sysdba” 
• Memory & Key Init.ora Changes
• Indexing Arsenal
  – Function-based index
  – Index merges
  – Bitmap join index
  – Skip scan index
• Merge (Upsert) Command
• External Tables
Overview (cont.)

• Partitions
  - Range (Oracle8 which is version 8.0)
  - Hash (Oracle8i which is 8.1+)
  - Composite (Oracle8i)
  - List (Oracle9i which is 9.0.1+)

• Summary
# Impact Tuning with Oracle8i

<table>
<thead>
<tr>
<th>Option</th>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partitions</td>
<td>120 sec - 310M</td>
<td>0.43 sec - 200k</td>
</tr>
<tr>
<td>Partitions / Tuned</td>
<td>120 sec - 310M</td>
<td>0.01 sec - 8k</td>
</tr>
<tr>
<td>Parallel Query (20 Proc.)</td>
<td>230 sec</td>
<td>18 sec</td>
</tr>
<tr>
<td>Function-Based Index</td>
<td>1206 sec - 3G</td>
<td>7 sec - 8k</td>
</tr>
<tr>
<td>Materialized View</td>
<td>28 sec</td>
<td>3 sec</td>
</tr>
<tr>
<td>Cursor_Sharing</td>
<td>240 sec</td>
<td>0.01 sec</td>
</tr>
<tr>
<td>Truncate</td>
<td>510 sec / 8G</td>
<td>0.40 sec / 32k</td>
</tr>
<tr>
<td>Driving Table</td>
<td>900 sec</td>
<td>1 second</td>
</tr>
<tr>
<td>SGA Sizing</td>
<td>30 sec</td>
<td>0.01 sec</td>
</tr>
<tr>
<td>750,000 Query Mix</td>
<td>5.1 T / 540 hours</td>
<td>9 G / 23 hours</td>
</tr>
</tbody>
</table>
Hourly Processing (After implementing 8i Features):
Before: 6,800M and 43.2 minutes
After: 12 M and 1.8 minutes

99.8% Less Data Accessed
96.8% Time Reduction
Enjoy the Day!

Call with questions: (800) 755-TUSC; niemiecr@tusc.com
www.tusc.com
For More Information

- www.tusc.com
- Oracle Performance Tuning Tips & Techniques, Richard J. Niemiec; Oracle Press
- Oracle PL/SQL Tips and Techniques, Joseph P. Trezzo; Oracle Press
- Oracle Application Server Web Toolkit Reference, Bradley D. Brown; Oracle Press
Copyright Information

• Neither TUSC, the IOUG-A or the author guarantee this document to be error-free. Please provide comments/ questions to niemiecr@tusc.com.

• TUSC © 2000. This document cannot be reproduced without expressed written consent from an officer of TUSC