Oracle9i New Features for DBA's

Presented at NYOUG
September 23, 2003

Dave Anderson
dave@skillbuilders.com

SKILLBUILDERS

Goals

- Brief introduction to 9i "DBA" features
- Grasp *understanding of purpose* of feature
- We'll be moving fast – lots to cover!
  - We'll see some examples where possible
  - Not a lot of time for
    - Pros / Cons
    - Performance

Agenda

- Server Parameter Files
- SGA Management
  - Parameters
  - Multiple blksize support
  - Dynamic sizing
  - Keep & Recycle Caches
- Auto Undo Mgmt
- Resumable Space Mgmt
- New online redefinition support
- Enhancements to Logminer
- RMAN Enhancements
- Performance & Tuning Enhancements
- New Index Stuff
- Partitioning Enhancements
- Flashback Query
- External Tables
- Security Enhancements
- Deprecated & Desupport Features

Server Parameter Files

SKILLBUILDERS

Introduction to SPFILE

- SPFILE = Server Parameter File
- Similar to "init.ora" parameter file
  - Contains initialization parameter values
- Differences from parameter file:
  - Binary format
  - SPFILE always resides on database server
    - Parameter file resides on remote machine performing STARTUP
  - Supports persistent parameter changes
  - Auto-backup with RMAN

Creating SPFILE

- Created with the CREATE SPFILE command
  - Input is text parameter file
  - Output is binary SPFILE

```
SQL> connect dave/dave as sysdba
Connected.
SQL> shutdown
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> create spfile from pfile;

File created.
```

Create spfile when DB is shutdown

Create text pfile from spfile too

```
SQL>
SQL> create pfile = 'c:\temp\initdave.ora' from spfile;

File created.
```

Using SPFILE

- STARTUP command with no parameters

```
STARTUP
```

- Uses 1st file found in search sequence: `$ORACLE_HOME\database`:

  1. Checks for `spfile$ORACLE_SID.ora`
  2. Checks for `spfile.ora`
  3. Checks for a text file called `init$ORACLE_SID.ora`

- The first file found is used

Persistent Parameters

- SPFILE provides persistent parameter initialization
- Change persists even after shutdown

```
ALTER SYSTEM
SET shared_pool_size=75M
COMMENT='changed on 06/10/2002'
SCOPE=BOTH;
```

- SCOPE=BOTH changes memory and SPFILE
  - Only valid for dynamic parameters
- Optional COMMENT can be seen in v$spparameter

SGA Management

Multiple database block size support

Dynamic SGA Memory

Changes to the KEEP and RECYCLE caches

SKILLBUILDERS

Dynamic Memory…

- SGA_MAX_SIZE
  - Total SGA limit in bytes
  - Static parameter
  - Enables the dynamic resizing of SGA areas
  - Reserves memory on Windows
  - Dynamically allocated on Solaris
- DB_CACHE_SIZE
  - Replaces DB_BLOCK_BUFFERS
  - Specify in bytes

…Dynamic Memory

- Buffer cache, sub-caches and shared pool can be dynamically resized

```
ALTER SYSTEM
SET DB_CACHE_SIZE = 48M
COMMENT='Changed on 05/22/2002'
SCOPE=BOTH;
```

- Memory allocated and freed in "granules"
  - Rounding can occur
- Granule size determined by total SGA size
- Query `v$sga_dynamic_components` to see current size

Multiple Blocksize Support

- Assigned with the CREATE TABLESPACE command

```
CREATE TABLESPACE hr_data_tbs
BLOCKSIZE 16K
DATAFILE 'e:\oradata\prod\hr_data_tbs.dbf'
SIZE 500M;
```

- COMPATIBLE parameter must be set to 9.0 or above
- Must define subcaches in parameter file

```
DB_2K_CACHE_SIZE=16000000
DB_16K_CACHE_SIZE=32000000
```

KEEP & RECYCLE Caches

- `DB_KEEP_CACHE_SIZE`
  - Replaces `buffer_pool_keep`
  - Now *not* part of standard buffer cache
- `DB_RECYCLE_CACHE_SIZE`
  - Replaces `buffer_pool_recycle`
  - Now *not* part of standard buffer cache

Automatic Undo Management

SKILLBUILDERS

Introducing AUM…

- AUM is “automatic undo management”
  - Uses Undo Tablespace instead of rollback segments
- Use of rollback segments is “manual undo management mode”
- Undo has not gone away
  - Rollback segments have

…Introducing AUM

- Easier to administer
  - Less management headaches
    - Number of RBS's?
    - Size of RBS's?
    - Maximum Extents?
    - Optimal Size?
- Provides control over retention time
  - Specify number of seconds to retain undo
  - Can reduce "snapshot too old" errors

Implementing AUM…

- Create an “undo tablespace”
- Use the new UNDO clause on CREATE TABLESPACE

```
CREATE UNDO TABLESPACE skillbuilders_undo
DATAFILE
'c:\oracle9i\oradata\test\skillbuilders_undo01.dbf'
SIZE 100M AUTOEXTEND ON;
```

…Implementing AUM

- Add AUM-related parameters
  - UNDO_MANAGEMENT = AUTO
  - UNDO_TABLESPACE = skillbuilders_undo
  - UNDO_RETENTION = 3600
  - UNDO_SUPPRESS_ERRORS = TRUE
- New dictionary views
  - V$UNDOSTAT
  - DBA_UNDO_EXTENTS

Resumable Space Management

SKILLBUILDERS

Introduction to RSM

- Traps space allocation failures
  - No rollback
  - No restarting job from beginning
- RSM *suspends* a failed transaction
  - Allows DBA to correct
  - Automatically restarts
- Transaction can be suspended and resumed multiple times during execution

Errors Handled

- Unable to extend segment
  - ORA-1650, ORA-1653, ORA-1654
- Max extents reached
  - ORA-1628, ORA-1631, ORA-1632
- Exceeding a tablespace quota
  - ORA-1536

Enabling RSM

- Done at the session level
- Consider `ON LOGON` trigger
- Use `ALTER SESSION` or `DBMS_RESUMABLE`
- Enable RSM, suspend transaction for 3 hours:

```
alter session enable resumable timeout 10800
name 'Update of hr table';
```

```
SQL> alter session disable resumable;
Session altered.
```

- Must have `RESUMABLE` system privilege

AFTER SUSPEND Trigger

```
CREATE OR REPLACE TRIGGER resumable_trans
AFTER SUSPEND ON DATABASE
DECLARE
.
.
.
END;
```

- Automate responses to suspend conditions
- Email DBA
- Change default `TIMEOUT`
- Log suspension events

RSM: Final Thoughts

- Can monitor via DBA_RESUMABLE, Alert Log
- Utility support
  - SQL*Loader and Import

New Online Operations

DBMS_REDEFINITION

More Online Index Maintenance

ANALYZE VALIDATE STRUCTURE

SKILLBUILDERS

Online Table Redefinition

- Online Table Operations Available
  - Modify storage parameters
  - Move to a different tablespace
  - Add or drop partitioning support
  - Recreate table to reduce fragmentation
- Implemented via the `DBMS_REDEFINITION` package

DBMS_REDEFINITION

- Five procedures
  - can_redef_table
  - start_redef_table
  - sync_interim_table
  - finish_redef_table
  - abort_redef_table

```
SQL> EXECUTE DBMS_REDEFINITION.START_REDEF_TABLE(USER,'t','it', -
>         'c1 c1, c2+10 c2', -
>         dbms_redefinition.cons_use_pk)

PL/SQL procedure successfully completed.
```

Online Index Operations…

- "Online" index rebuild first available in Oracle8i
  - B-tree indexes only
- Waits for exclusive access

SESSION 1

```
SQL> update emp set sal = 2000;
14 rows updated.
```

SESSION 2

```
SQL> create index ix3 on prod_dba.emp
(hiredate) online;
```

SESSION2 is now blocked.

SESSION 1

```
SQL> commit;
Commit complete.
```

Once the commit from the first user has completed, the create index completes.

…Online Index Operations

- Oracle9i now supports
  - Reverse Key, Key Compressed, Function-based
  - Secondary Indexes on Index-organized tables (IOTs)
- But unfortunately no BITMAP support

ANALYZE - ONLINE Clause

- Used with the VALIDATE STRUCTURE option

```
ANALYZE TABLE orders VALIDATE STRUCTURE ONLINE;
```

- Validation can run while DML operations are executing against the object
- Some performance impact
- OFFLINE clause prevents DML from executing during validation

LogMiner

Command-Line Logminer

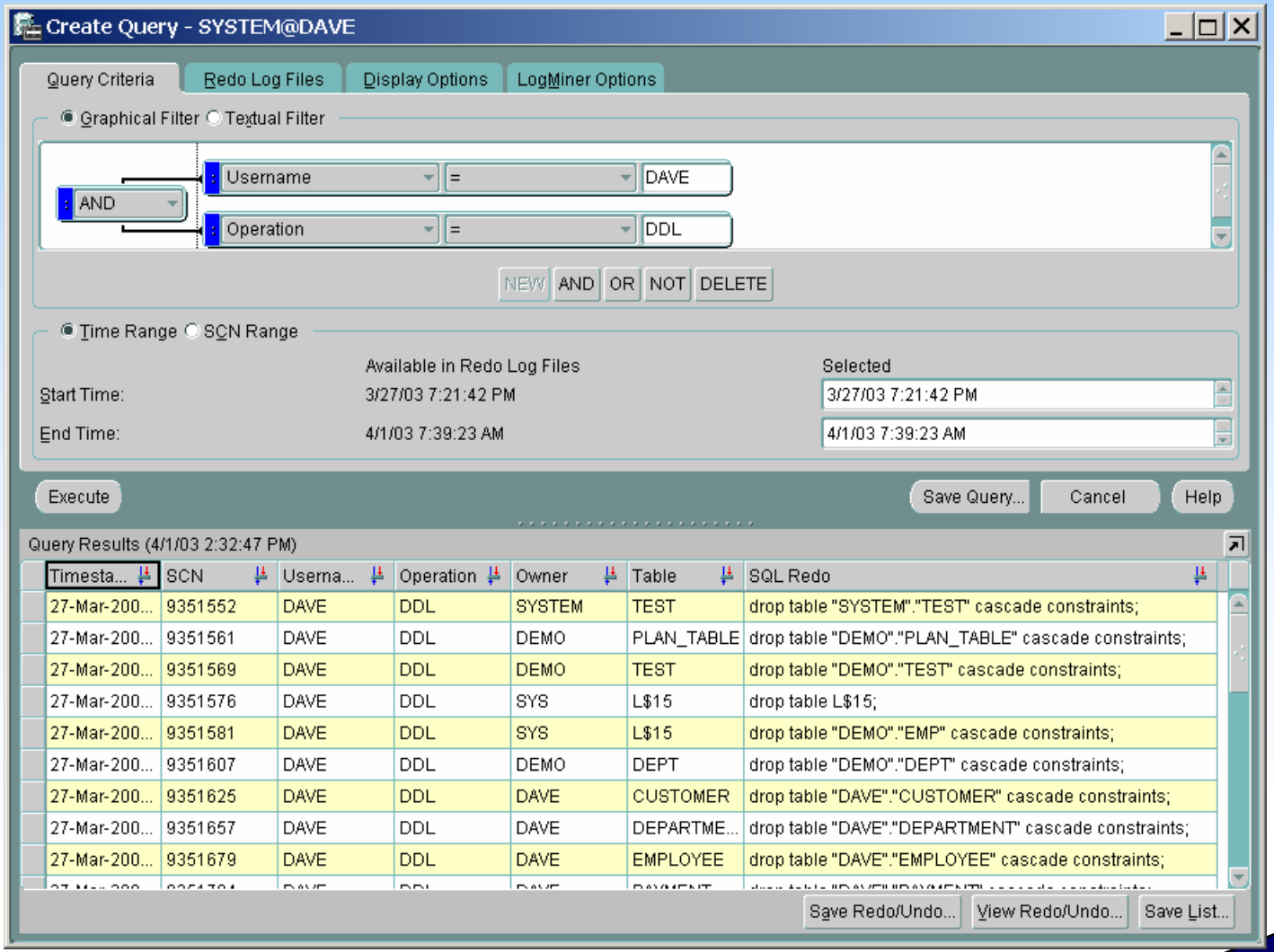
- Utility to read redo log(s)
- Sample results from V$LOGMNR_CONTENTS

```
SELECT username, operation, sql_redo, sql_undo
        FROM v$logmnr_contents;
USERNAME                       OPERATION
------------------------------ --------------------------------
SQL_REDO                       SQL_UNDO
------------------------------ --------------------------------
APP_DEVELOPER                  UPDATE

update "APP_DEVELOPER"."CUSTOMER_NAME" set "LAST_NAME" =
'Smith' where "ID" = '10' and "LAST_NAME" = 'Jones' and ROWID
= 'AAAHEgAAIAAAAASAAA';

update "APP_DEVELOPER"."CUSTOMER_NAME" set "LAST_NAME" =
'Jones' where "ID" = '10' and "LAST_NAME" = 'Smith' and ROWID
= 'AAAHEgAAIAAAAASAAA';
```

REDO

UNDO

9i LogMiner Viewer



More 9i Features

- Support for more object types and data types
  - Clustered tables
  - Chained and migrated rows
  - LOB and LONG datatypes
  - Direct loads
  - Scalar object types
  - Data definition statements (DDL)
  - Parallel data manipulation language (DML)

Introduction to Recovery Manager

SKILLBUILDERS

9i R1 Features…

- AUTOBACKUP of control file
  - Automatic backup to known name and location
  - Recover even if control file and RMAN catalog are lost
- Restartable backups
  - Restart a long running backup after crash
- Simpler command syntax
  - Eliminate RUN blocks in some cases
- Stored channel settings
  - Configure persistent channels and channel settings
  - Eliminates need to constantly reallocate channel(s)

…9i R1 Features

- Recover blocks
  - Block Media Recovery
- Backup and Restore Optimization
  - Skip datafiles that have not changed
- Smarter backup of archive logs
  - Now automatically switches and backs up online logs during backup of database
  - Good for backups while database is open

9i R2 Features

- Backup / restore of SPFILE
- Automatic space management for archive logs
- Database structural changes cause auto backup of controlfile / SPFILE
- `DUPLICATE` command can
  - Exclude tablespaces
  - Create duplicate DB at non-current time

Performance and Tuning Enhancements

SKILLBUILDERS

What's New for Tuning?

- STATISTICS_LEVEL Parameter
- Advisories
- Dynamic SGA
- Precise Query Statistics and Plans WITH `V$SQL_PLAN` and `V$SQL_PLAN_STATISTICS`
- System Statistics - I/O and CPU time
- Dynamic Sampling
- New Cursor sharing features
- Table Compression
- Automatic Segment Space Management
- More….
- Let's take a look at a few of these features…

Statistics Level...

- Initialization parameter
- Controls level of performance statistics collected

```
STATISTICS_LEVEL=BASIC|TYPICAL|ALL
```

- `BASIC` – No statistics collected
- `TYPICAL` – Timed statistics and Advisory statistics
  - e.g. `V$DB_CACHE_ADVICE`
- `ALL` – `TYPICAL` plus timed OS and row source (plan) execution statistics
  - Expensive
  - Better to use TYPICAL and SQL_TRACE=TRUE at session level when tuning

...Statistics Level

```
  1  alter system
  2  set statistics_level = typical
  3* scope = memory
SQL> /

System altered.

SQL> select statistics_name, system_status, statistics_view_name
  2  from v$statistics_level;

STATISTICS_NAME                SYSTEM_S STATISTICS_VIEW_NAME
------------------------------ -------- ----------------------
Buffer Cache Advice            ENABLED  V$DB_CACHE_ADVICE
MTTR Advice                    ENABLED  V$MTTR_TARGET_ADVICE
Timed Statistics               ENABLED
Timed OS Statistics            DISABLED
Segment Level Statistics       ENABLED  V$SEGSTAT
PGA Advice                     ENABLED  V$PGA_TARGET_ADVICE
Plan Execution Statistics      DISABLED V$SQL_PLAN_STATISTICS
Shared Pool Advice             ENABLED  V$SHARED_POOL_ADVICE
```

Shared Pool Advisory

- What's the affect of changing shared pool size?
- Must set STATISTICS_LEVEL=TYPICAL or ALL
- Run database under normal conditions

```
SQL> select SHARED_POOL_SIZE_FOR_ESTIMATE  ,SHARED_POOL_SIZE_FACTOR
  2  ,ESTD_LC_SIZE  ,ESTD_LC_MEMORY_OBJECTS
  3  ,ESTD_LC_TIME_SAVED ,ESTD_LC_TIME_SAVED_FACTOR
  4  ,ESTD_LC_MEMORY_OBJECT_HITS
  5  From v$shared_pool_advice;

      SFE         SF    LC_SIZE LC_MEM_OBJ TIME_SAVED TIME_SAVED_F MEM_OBJ_HITS
--------- ---------- ---------- ---------- ---------- ------------ ------------
        8      .6667          2        701        678            1        55283
       12          1          2        701        678            1        55283
       16     1.3333          2        701        678            1        55283
       20     1.6667          2        701        678            1        55283
       24          2          2        701        678            1        55283
```

Buffer Cache Advisory

- Ask database for buffer cache sizing advice
  - STATISTICS_LEVEL=TYPICAL
  - Run database under "normal" workload
- Reads if cache size was increased? Decreased?

```
select SIZE_FOR_ESTIMATE ,SIZE_FACTOR
      ,BUFFERS_FOR_ESTIMATE
      ,ESTD_PHYSICAL_READ_FACTOR
      ,ESTD_PHYSICAL_READS
from v$db_cache_advice;

SIZEMB SIZE_FACTOR #BUFFERS EST_P_READ_FACTOR EST_P_READS
------ ----------- -------- ----------------- -----------
     4        .1667      500              4.72       4,463
     8        .3333    1,000              1.08       1,019
    12           .5    1,500              1.00         945
```

System Statistics...

- Release 1 feature, provides collection and storing of I/O and CPU statistics
- Used by optimizer when weighing candidate plans
- Factors in estimated CPU and I/O cost

```
-- Collect statistics with in manual start/stop mode
exec DBMS_STATS.GATHER_SYSTEM_STATS(gathering_mode=> 'START')

-- Stop statistics collection:
exec DBMS_STATS.GATHER_SYSTEM_STATS(gathering_mode=> 'STOP')
```

…System Statistics

```
  1  select pname, pval1, pval2
  2* from sys.aux_stats$
SQL> /

PNAME                     PVAL1 PVAL2
--------------- ---------- ---------------
STATUS                          COMPLETED
DSTART                          03-26-2003 14:40
DSTOP                           03-26-2003 14:44
FLAGS                        1
SREADTIM                23.192
MREADTIM                37.892
CPUSPEED                   226
MBRC                        15
MAXTHR                   44032
SLAVETHR                    -1
```

Also can use DBMS_STATS.GET_SYSTEM_STATS

Table Compression

- Table compression saves space, lowers I/O counts

```
SQL> create table t as select * from dba_objects;
Table created.

SQL> select bytes, blocks from user_segments
  2  where segment_name = 'T';

     BYTES     BLOCKS
---------- ----------
   4194304        512

SQL> create table t2 COMPRESS as select * from dba_objects;
Table created.

SQL> select bytes, blocks from user_segments
  2  where segment_name = 'T2';

     BYTES     BLOCKS
---------- ----------
   2097152        256
```

48% less blocks

Direct path INSERT

Create table as select

SQL*Loader direct path loads

Alter table Move

Precise Query Stats

- V$SQL_PLAN
  - Shows actual plan used
- V$SQL_PLAN_STATISTICS
  - Plan runtime statistics

```
SQL> select /*test5*/ *
  2  from t where c2 = 1176748918;

        C1         C2
---------- ---------
 107974537 117674
```

Query
v$sql_plan_statistics

```
ID OPERATION                    OPTIONS    OBJECT_NAME
-- -------------------- --------- ------------
  1  TABLE ACCESS                FULL       T
```

```
EXEC    LOGI_IO  PHYS_IO ELAPSED_TIME
------- ------- -------- ------------
      4     356        0        39221
```

More Perf & Tuning Stuff

- Dynamic Sampling
  - Sample data at parse time to get statistics
  - See `optimizer_dynamic_sampling` parameter
- PGA Management
  - Automatic PGA allocation
  - `WORKAREA_SIZE_POLICY=AUTO`
  - Also see PGA Advisory
- Auto Segment Space Management
  - Bitmap tracks free space on block
  - Obsoletes `PCTUSED`, `FREELIST` and `FREELIST GROUPS`
- New Cursor Sharing option
  - `CURSOR_SHARING = SIMILAR`

More Perf & Tuning Stuff

- Forced Rewrite
  - Force the rewrite of queries to use materialized views

```
SQL> alter session set query_rewrite_enabled = force;

Session altered.
```

- R2 supports parallel DML on non-partitioned tables
- Optimize for *specified number* of rows
  - See `FIRST_ROWS_n` initialization parameter and hint
- Bind Variable Peeking
  - interrogate the bind variables for values during hard parse

More Perf & Tuning Stuff

- Resource Manager
  - Stop long-running queries *before they start*
  - See the `MAX_EST_EXEC_TIME` directive
- Outline Editing
  - Update (DML) stored outlines
- More new hints
  - R1: `NL_AJ, NL_SJ, FACT, NO_FACT`
  - R2: `EXPAND_GSET_TO_UNION`
- STATSPACK
  - Introduced with 8i, but now `utlbstat/utlestat` deprecated
- `FAST_START_MTTR_TARGET`
  - Tune Recovery
- RBO going away
  - 9.2 last release to support RBO

Indexes

Bitmap Join Indexes

- Bitmap index for the join of two or more tables
- Intended for data warehouse
  - Fact to Dimension table join
  - `DIMENSION.PK = FACT.FK`
- Pre-computes the join operation
- Reduce, sometimes eliminate join during query
- Bitmap structure is space efficient

Creating Bitmap Join Index

```
CREATE BITMAP INDEX emp_dept_bidx ON emp( d.dname )
FROM emp e, dept d
WHERE e.deptno = d.deptno;
```

- Given these tables:

```
1 Jones     10
2 Daswani   30
3 Yamauchi  20
4 Anderson  40
5 Belke     10
```

```
10 Sales
20 IT
30 Shipping
40 HR
```

- Bitmap created:

Sales	IT	Shipping	HR
1	0	0	0
0	0	1	0
0	1	0	0
0	0	0	1
1	0	0	0

Using the Index

```
  1  select count(*)
  2  from emp e, dept d
  3  where e.deptno = d.deptno
  4* and d.dname = 'SALES'
SQL> /

  COUNT(*)
----------
         6
Execution Plan
----------------------------------------------------------
0   SELECT STATEMENT Optimizer=CHOOSE (Cost=1 Card=1 Bytes=13
1 0 SORT (AGGREGATE)
2 1 BITMAP CONVERSION (COUNT)
3 2 BITMAP INDEX (SINGLE VALUE) OF 'EMP_DEPT_BIDX'
```

No table access, No join

Data Dictionary Impact

- *_INDEXES
  - New Column
    - JOIN_INDEX
- *_IND_COLUMNS
  - TABLE_NAME column indicates the dimension table used
- *_JOIN_IND_COLUMNS
  - New view in Oracle9i

Index Skip Scan…

- Prior to 9i index usage required leading column
- Skip Scan access path uses index without leading column in predicate
- Requires the use of CBO
- See `PLAN_TABLE` and `V$SQL_PLAN`
  - `OPERATION = INDEX`
  - `OPTIONS = SKIP SCAN`

…Index Skip Scan

- Given:

```
create index it on t(firstname, lastname);
```

Tests showed hint needed,

See Metalink Document 212391.1

```
SQL> select /*+ index(t it) */   *
  2  from t where lastname = 'Anderson' ;

2500 rows selected.

Execution Plan
----------------------------------------------------------
   0      SELECT STATEMENT Optimizer=CHOOSE (Cost=2498 Card=2500 Bytes
          =207500)

   1    0   TABLE ACCESS (BY INDEX ROWID) OF 'T' (Cost=2498 Card=2500
          Bytes=207500)

   2    1     INDEX (SKIP SCAN) OF 'IT' (NON-UNIQUE) (Cost=1092 Card=12)
```

More Index Enhancements

- Index Use Monitoring

```
ALTER INDEX bix MONITORING USAGE;
```

- IOTs and Bitmap Indexes
- `DROP CONSTRAINT…KEEP INDEX`
- `CREATE INDEX` on the `CREATE TABLE` statement

Partitioning Enhancements

SKILLBUILDERS

List Partitioning

- `PARTITION BY LIST (column)`
  - Specifies a single column to partition on
- `VALUES(value1,value2,…,valuex)`
  - Identify the column values for that partition

```
create table comedians
(comedian_no  number
,lastname     varchar2(20)
,firstname    varchar2(15)  )
partition by list (firstname)
(partition THREE_STOOGES values('MO', 'LARRY', 'CURLY'),
 partition MARX_BROTHERS values('HARPO', 'CHICO', 'ZEPPO'),
 partition ABBOTT_AND_COSTELLO   values('LOU', 'XYZ') );
```

Composite Range-List Partitioning

- Two level hierarchy
- Partition by RANGE first
- Subpartition by LIST

```
PARTITION BY RANGE (region)
SUBPARTITION BY LIST (state)
(PARTITION new_england VALUES LESS THAN (2)
 (SUBPARTITION ct       VALUES ('CT'),
  SUBPARTITION ma       VALUES ('MA'),
. . .
```

Other Partitioning Features

- Subpartition Templates
  - Easier coding
- `UPDATE GLOBAL INDEXES` when performing table maintenance with `ALTER TABLE`
- IOTs now support HASH partitioning

Flashback Query

SKILLBUILDERS

Concepts

- Execute SELECT as if it was being run in the past
  - Database restore not necessary
  - “Self-service error correction”

```
SQL> select lastname
  2  from customer as of timestamp
     (SYSTIMESTAMP - INTERVAL '1' DAY);

LASTNAME
--------------------
Jones
Smith
```

SYSTIMESTAMP returns the current timestamp of the server.

“INTERVAL '1' DAY” is an interval literal

Flashback Window

- Window depends on UNDO_RETENTION parameter
  - For example, retain 12 hours of UNDO
    - `undo_retention = 43200`
  - OEM will calculate space requirements
    - Retention * undo blocks per second * blocksize
- Should use 9i automatic UNDO management
  - `undo_management = AUTO`
  - Have control over retention time

Possibilities

- Specify specific date and time

```
select lastname
from customer AS OF timestamp
to_timestamp('2003-01-08 05:30:00', 'YYYY-MM-DD HH:MI:SS')
```

- Compare incarnations of a table

```
select lastname
from employee
minus
select lastname
from employee as of timestamp
  to_timestamp('2003-03-09 05:30:00', 'YYYY-MM-DD HH:MI:SS');
```

More Possibilities…

- Recreate table

```
  1  create table new_customer as
  2  select *
  3* from customer as of timestamp (systimestamp - interval
                                     '30' minute)
SQL> /
```

- Session-level flashback

```
exec DBMS_FLASHBACK.ENABLE_AT_TIME (SYSDATE - (1/24))
select * FROM customer;
select * from ord;
exec DBMS_FLASHBACK.DISABLE
```

…Even More Possibilities

- Create view

```
create or replace view old_customer as
select lastname from customer
AS OF TIMESTAMP (SYSTIMESTAMP - INTERVAL '1' DAY);
```

Final Thoughts on Flashback

- Cannot flashback across DDL time
- No DML or DDL while in flashback mode
- New object privilege called FLASHBACK
- System privilege FLASHBACK ANY TABLE

External Tables

SKILLBUILDERS

Concepts…

- *Query* flat files on the file system
  - No updates
  - No indexes
- Provides a good Extraction, Transformation and Load (ETL) tool
- Many SQL query operations supported
  - `SELECT`, Join, source for `MERGE`, `INSERT`, multi-table `INSERT`, Views

…Concepts

- ORACLE_LOADER Access Driver is supplied
  - It is really SQL*Loader
  - Most options available
- External tables are a way to access SQL*Loader
  - W/O going to OS command line

Create a Directory

- First create Oracle `DIRECTORY`
- Directory is alias for file system directory
- Contains OS file(s)

DBA_DIRECTORIES records all directories

```
SQL> create directory alert as 'C:\oracle\admin\dave\bdump';
Directory created.

SQL> create directory external_logs as 'C:\Oracle_logs\'
Directory created.

SQL> grant read on directory external_tables to dave;
Grant succeeded.

SQL> grant write on directory external_logs to dave;
Grant succeeded.
```

Create the Table

```
SQL> CREATE TABLE alert_log_ext
  2    (detail_line VARCHAR2(255))
  3    ORGANIZATION EXTERNAL
  4    (
  5     TYPE oracle_loader
  6     DEFAULT DIRECTORY alert
  7     ACCESS PARAMETERS
  8       (
  9        RECORDS DELIMITED BY NEWLINE
 10        NOBADFILE
 11        LOGFILE external_logs:'alert_log.log'
 12        NODISCARDFILE
 13        FIELDS
 14          MISSING FIELD VALUES ARE NULL
 15          ( detail_line char(255) )
 16        )
 17     LOCATION('alert_dave.log')
 18    )
 19    NOPARALLEL;
 Table created.
```

Query

```
select * from alert_log_ext
where detail_line like '%ORA-%'
/

create table alert_log
as select * from alert_log_ext
/

insert /*+ append */ into alert_log
select * from alert_log_ext;

create table temp as
select upper(detail_line) as detail_line
from alert_log_ext
/
```

Query external table

Load external table into a permanent table

Load with direct path INSERT

Load with SQL functions

Final Notes on Flashback…

- Parallel load is supported on fixed width files
- Auto-Generate DDL & Access Parameters

```
sqlldr scott/tiger ulcase1 EXTERNAL_TABLE=GENERATE_ONLY
```

- Gather statistics:

```
SQL> exec dbms_stats.gather_table_stats('system',
   'student_emails_ext')


PL/SQL procedure successfully completed.
```

…Final Notes on Flashback

- Combine with *pipelined functions* to create powerful ETL functions
- New dictionary views
  - *_EXTERNAL_TABLES
  - *_EXTERNAL_LOCATIONS
- Hints work, e.g. USE_MERGE
- Can export structure, not data
- Can ALTER access parameter, location

9i Table Function Enhancement

Security Enhancements

SKILLBUILDERS

GRANT ANY OBJECT PRIVILEGE

- DBA can GRANT/REVOKE on other users objects

```
SQL> connect system/dave@laptop
Connected.

SQL> grant grant any object privilege to dave;
Grant succeeded.

SQL> connect dave/dave@laptop
Connected.

SQL> grant select on system.t to scott;
Grant succeeded.
```

Security Enhancements…

- `SELECT ANY DICTIONARY` Privilege
  - Permits query of data dictionary
  - W/O need to grant DBA role or SYSDBA
- Supply password for `SYS/SYSTEM` on `CREATE DATABASE`
- Audit `SYS, SYSDBA` and `SYSOPER` operations
- Initial userids are all locked and expired at database creation time

…Security Enhancements

- Significant Enhancements to:
  - Fine Grained Access Control / Virtual Private DB
  - n-Tier Proxy Authentication (JDBC support)
  - Fine Grained Auditing
  - Label Security
  - Oracle Policy Manager
  - Data encryption

Deprecated and Desupported Features

SKILLBUILDERS

Deprecated Features

- `ANALYZE` command to collect statistics
- `init.ora` parameters
- Export / Import `INCREMENTAL` functionality
- `LONG, LONG RAW` data types
- `bstat` / `estat` scripts

ANALYZE Command

- ANALYZE is deprecated
- Use DBMS_STATS package instead

```
begin
  dbms_stats.gather_schema_stats(user,
          cascade=>TRUE,
          method_opt=> 'FOR ALL INDEXED COLUMNS',
          options=>'GATHER STALE',
          objlist=>olist);
. . .
end;
```

Desupported Features

- Server Manager
- `CONNECT INTERNAL`

Data Dictionary Views

- Deprecated
  - DBA_SNAPSHOT_LOGS
    - Use DBA_MVIEW_LOGS instead
- Views added
  - DBA_UNDO_EXTENTS
    - Commit information about each extent in UNDO tablespaces
  - V$UNDOSTAT
    - Statistics for monitoring and maintenance of undo tablespace sizing problems
- Columns added to pre-9i views
  - BLOCKSIZE to DBA_TABLESPACES and V$DATAFILE
    - Multiple blocksize support

Wrapping Up

Oracle Managed Files

- Oracle Managed Files
  - Provides default location, name and size
  - Deletes OS file when object dropped
  - Datafiles, online redo logs, and control files
- OMF is optional
  - “Normal” file creation techniques still available

```
CREATE TABLESPACE part_tbs1;
CREATE TABLESPACE product_tbs2 datafile size 5M;
```

```
DROP TABLESPACE product_tbs1;
DROP TABLESPACE product_tbs3 INCLUDING CONTENTS AND DATAFILES;
```

SQL Features

- MERGE Statement
  - Update and Insert in one statement
- Multi-Table INSERT
  - Up to 127 tables
- ANSI Joins
- Subquery Factoring
- CONNECT BY Extensions
- New Functions
- Enhanced LOB support
- Scrollable Cursors
- Object inheritance and evolution
- New RENAME support
- DEFAULT keyword for UPDATE and INSERT

PL/SQL Enhancements

- ANSI CASE Statement
- Associative Arrays
- Record-Based DML
- Multi-Level Collections
- Pipelined Functions
- Native Compilation
- LOB Support
- DBMS_METADATA
- UTL_FILE

Datetime Data

- 3 new datetime datatypes
- `TIMESTAMP`
  - `DATE` with fractional seconds
- `TIMESTAMP WITH TIME ZONE`
  - `TIMESTAMP` with timezone preserved
- `TIMESTAMP WITH LOCAL TIME ZONE`
  - `TIMESTAMP` with datetime converted to sessions time zone
- Also, `INTERVAL` type available
  - Time period

Year, Month, Day
Hour, Minute Second, Fractional second

Hour displacement from UTC
Minute displacement
Time Zone region name and abbreviation

Datetime Functions

- Conversion functions
  - `TO_TIMESTAMP`
  - `TO_TIMESTAMP_TZ`
- Date functions
  - `CURRENT_DATE` (9i)
    - Current date and time of the *session*
    - No time zone
  - SYSDATE (not new)
    - Current date and time of the *database*
    - No time zone
- Timestamp functions
  - `LOCALTIMESTAMP`
    - Client datetime in TIMESTAMP format
    - *No time zone*
  - `CURRENT_TIMESTAMP`
    - Client datetime *with time zone*
  - `SYSTIMESTAMP`
    - Server OS datetime *with time zone*

XML in the Database

- 9i Release 2 introduces XML DB
- XMLType provides native XML storage
  - XPATH searches with SQL
  - XSL Transformations
  - OLAP Functions
  - Indexing for performance
  - Piecewise UPDATE support
- XML Repository
  - WebDAV, FTP protocol support

Huge Features

- DataGuard
  - New name for Standby Database feature
  - Release 2 supports *Logical* Standby database
- Real Application Clusters
  - New Oracle Parallel Server

Streams

- Streams
  - New data and event sharing technology
  - Accesses redo log
    - Binary redo into LCR's
    - LCR's into SQL
  - Next generation
    - Advanced Replication
    - Logical Standby
      - Support for heterogeneous environments
    - Data Warehouse load

The End

- Questions?
- Thank You for attending!
- www.skillbuilders.com
- Call Dave at 888.803.5607