New Features in Database 11g for Oracle Developers

11.1.0.6 - 11.2.0.1

presentation for:

19 April, 2010

Daniel A. Morgan | damorgan11g@gmail.com | www.morganslibrary.org

Introduction

- Daniel Morgan – damorgan11g@gmail.com
- Oracle Ace Director
- University of Washington, retired
- The Morgan of Morgan's Library on the web
  - www.morganslibrary.org/library.html
- Board of Directors Western Washington Oracle Users Group
- Member: UK Oracle Users Group
- Frequent speaker . . . . . .

- Oracle since version 6
- 11g beta test site

Morgan's Library

UNIVERSITY OF WASHINGTON

WWOUG

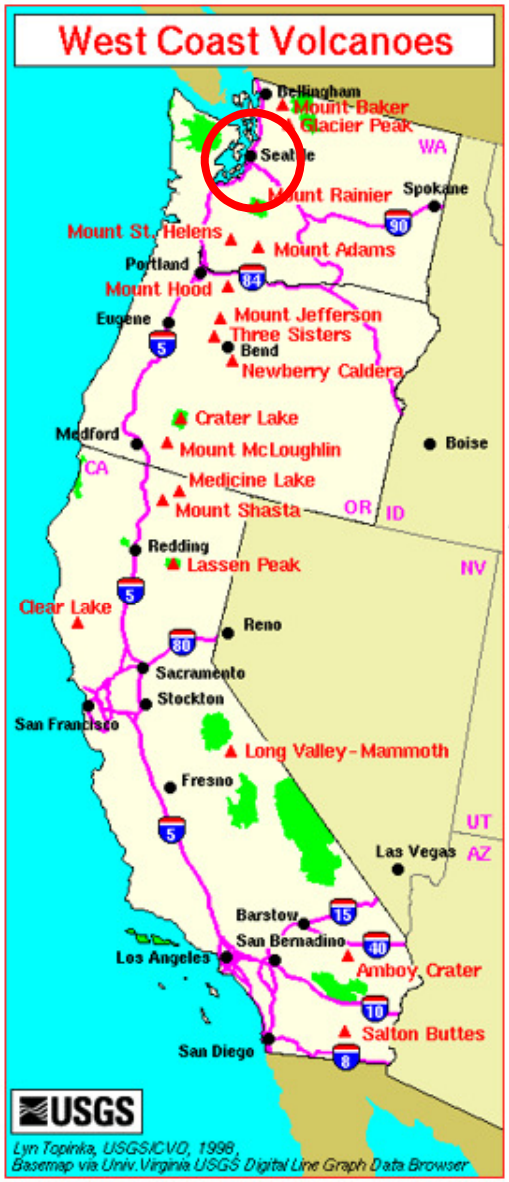
UKOUG

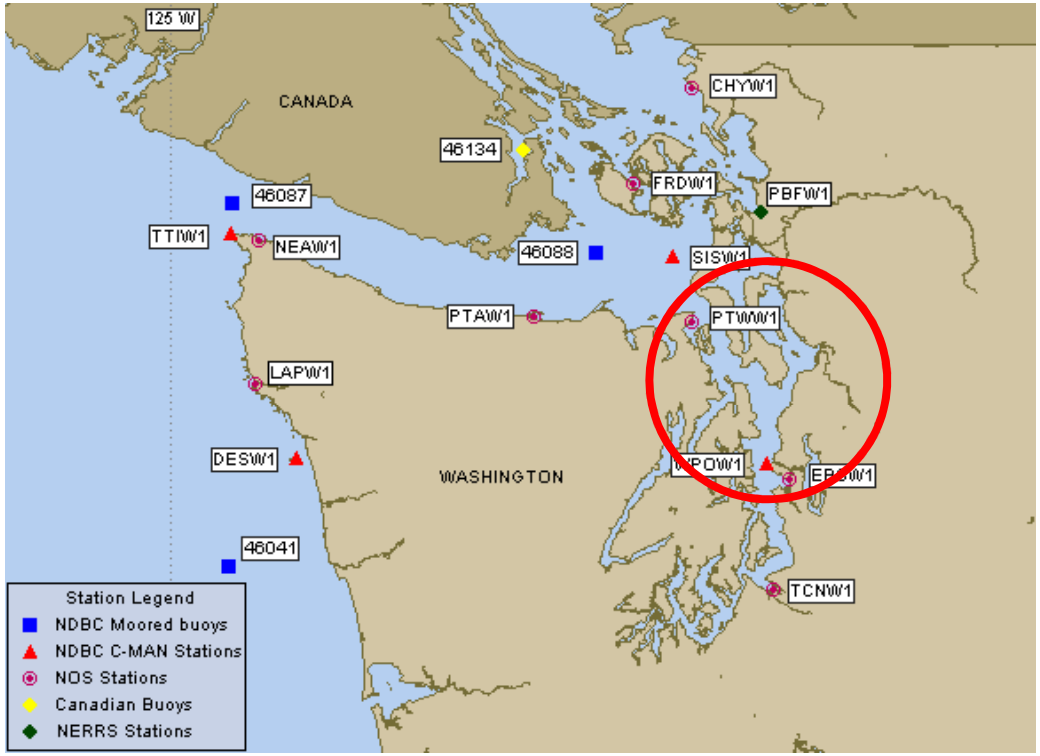
affiliate of OAUG oracle applications users group

Official Beta Site ORACLE DATABASE 11g

cd $MORGAN_HOME

cd $MORGAN_HOME

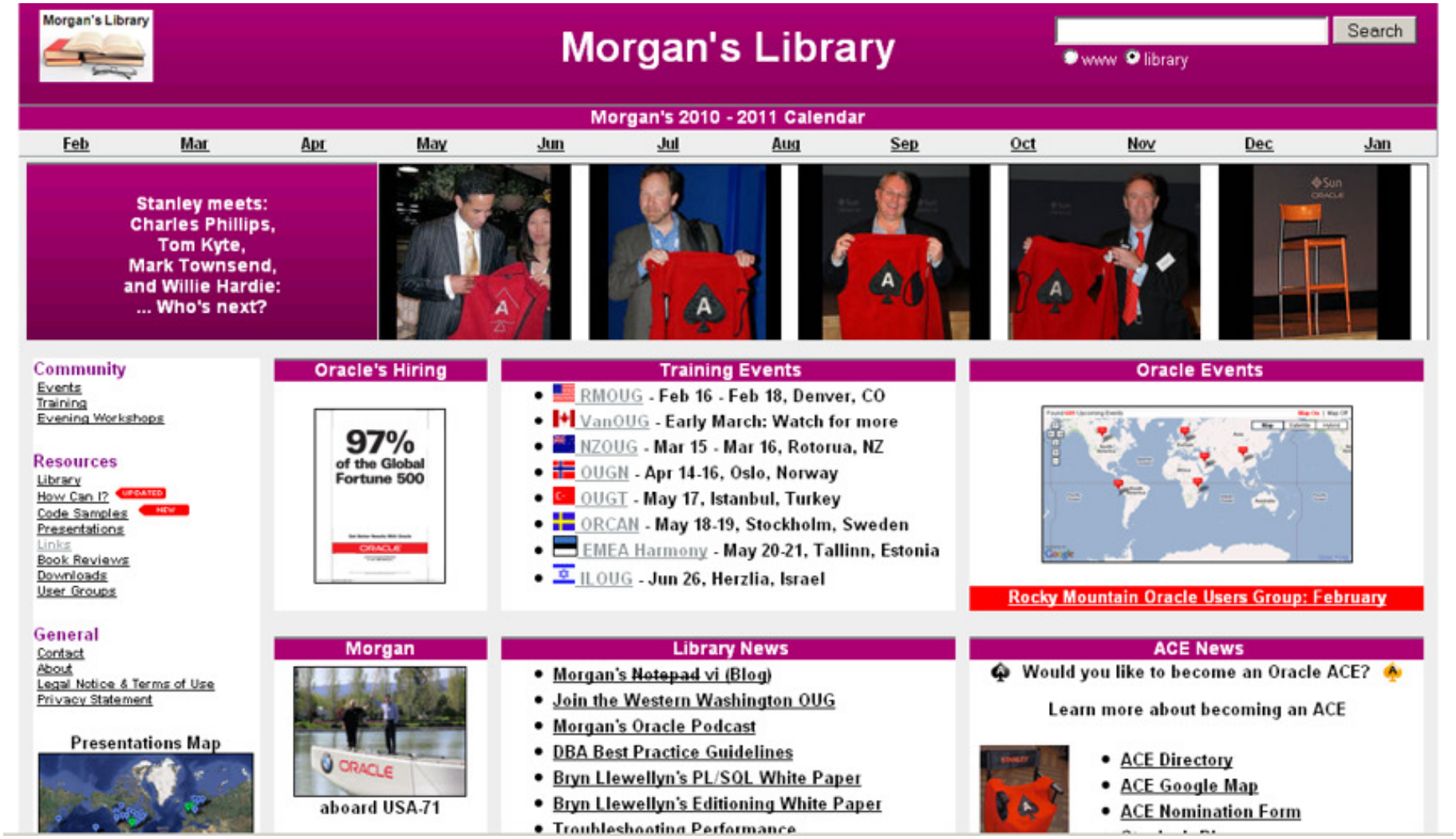




cd $MORGAN_HOME

Part of the Pacific Ring of Fire

Morgan's Library: www.morganslibrary.org



Sometime what is old is new

```
DECLARE
 SUBTYPE flagtype IS PLS_INTEGER RANGE 0..1;
 x flagtype;
BEGIN
  BEGIN
    x := 0;
    dbms_output.put_line('Success: ' || TO_CHAR(x));
  EXCEPTION
    WHEN others THEN
      dbms_output.put_line('Can not assign 0 to flagtype');
  END;

  BEGIN
    x := 1;
    dbms_output.put_line('Success: ' || TO_CHAR(x));
  EXCEPTION
    WHEN others THEN
      dbms_output.put_line('Can not assign 1 to flagtype');
  END;
  ....
END;
/
```

How Can I #11

```
BEGIN
  SELECT COUNT(*)
  INTO i
  FROM gv$backup_corruption;

  IF i <> 0 THEN
    dbms_system.ksdfls;
    dbms_system.ksdddt;
    dbms_system.ksdwrt(2,'ORA-20341: GV$BACKUP_CORRUPTION contains '|| TO_CHAR(i)||' corruption
records');
  END IF;

  SELECT COUNT(*)
  INTO j
  FROM gv$copy_corruption;

  IF j <> 0 THEN
    dbms_system.ksdfls;
    dbms_system.ksdddt;
    dbms_system.ksdwrt(2,'ORA-20342: GV$COPY_CORRUPTION contains ' || TO_CHAR(j)||' corruption
records');
  END IF;

  SELECT COUNT(*)
  INTO k
  FROM gv$database_block_corruption;

  IF k <> 0 THEN
    dbms_system.ksdfls;
    dbms_system.ksdddt;
    dbms_system.ksdwrt(2,'ORA-20343: GV$DATABASE_BLOCK_CORRUPTION contains '||TO_CHAR(k)||'
corruption records');
```

OpenWorld Unconference

Oracle OpenWorld Unconference

Monday - Oct 12

2pm

Overlook I: Fundamentals of Performance (Oracle ACE Director Cary Millsap)
Overlook II: Oracle Scheduler: what front end tools are available: bring your own tools and demonstrate ! (Ronald Rood, Ciber)
Overlook III: Hudson community meet up

Tuesday - Oct 13

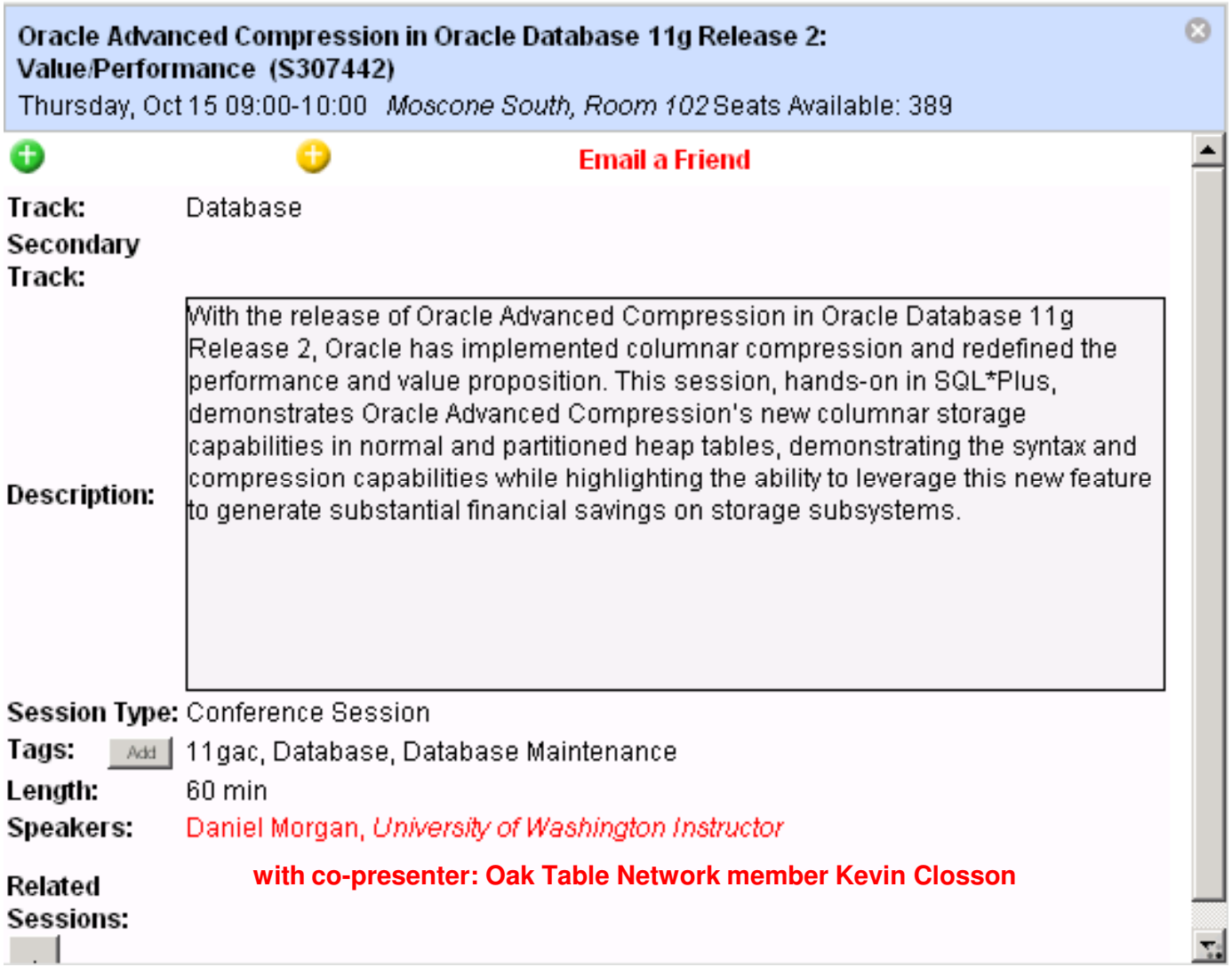
9am

Overlook I: What's New in Eleven ... Dot Two (that Oracle won't be talking about) presented by Oracle ACE Director Daniel Morgan
Overlook II: Creating Platform as a Service (PaaS) environments with WebLogic Server: Best Practices and Stories from the Trenches. (Bill Dettelback)
Overlook III: Reserved for onsite signups

10am

Overlook I: Oracle Indexes: Q & A Discussion With Oracle ACE Director Richard Foote
Overlook II: Experiences on Java programming within the databases, tips & tricks, tools, open source libraries and more. (Oracle ACE Marcelo F. Ochoa)
Overlook III: Reserved for onsite signups

Oracle OpenWorld 2009



Oracle Advanced Compression in Oracle Database 11g Release 2: Value/Performance (S307442)

Thursday, Oct 15 09:00-10:00 *Moscone South, Room 102* Seats Available: 389

Email a Friend

Track: Database

Secondary Track:

Description: With the release of Oracle Advanced Compression in Oracle Database 11g Release 2, Oracle has implemented columnar compression and redefined the performance and value proposition. This session, hands-on in SQL*Plus, demonstrates Oracle Advanced Compression's new columnar storage capabilities in normal and partitioned heap tables, demonstrating the syntax and compression capabilities while highlighting the ability to leverage this new feature to generate substantial financial savings on storage subsystems.

Session Type: Conference Session

Tags: Add 11gac, Database, Database Maintenance

Length: 60 min

Speakers: Daniel Morgan, *University of Washington Instructor*

with co-presenter: Oak Table Network member Kevin Closson

Related Sessions:

The Intersection

With Academia

- Bloom Filters
  - A space-efficient probabilistic data structure that is used to test whether an element is a member of a set. False positives are possible, but false negatives are not. The more elements that are added to the set, the larger the probability of false positives.

```
-------------------------------------------------------------------------------
| Id  | Operation                               | Name      | Pstart| Pstop |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                        |           |       |       |
|   1 |  PX COORDINATOR                         |           |       |       |
|   2 |   PX SEND QC (RANDOM)                   | :TQ10002  |       |       |
|*  3 |    FILTER                               |           |       |       |
|   4 |     HASH GROUP BY                       |           |       |       |
|   5 |      PX RECEIVE                         |           |       |       |
|   6 |       PX SEND HASH                      | :TQ10001  |       |       |
|   7 |        HASH GROUP BY                    |           |       |       |
|*  8 |         HASH JOIN                       |           |       |       |
|   9 |          PART JOIN FILTER CREATE        | :BF0000   |       |       |
|  10 |           PX RECEIVE                    |           |       |       |
|  11 |            PX SEND PARTITION (KEY)      | :TQ10000  |       |       |
|  12 |             PX BLOCK ITERATOR           |           |       |       |
|  13 |              TABLE ACCESS FULL          | CUSTOMERS |       |       |
|  14 |          PX PARTITION HASH JOIN-FILTER  |           |:BF0000|:BF0000|
|* 15 |           TABLE ACCESS FULL             | SALES     |:BF0000|:BF0000|
-------------------------------------------------------------------------------
```

The Intersection with Academia

- Hash tables (aka hash map)
  - A data structure that uses a hash function to map identifiers or keys (e.g., person names) to associated values (e.g., their telephone numbers). A hash function is used to transform the key into an index (the hash) of an array element (the slot or bucket) where the corresponding value is to be sought.

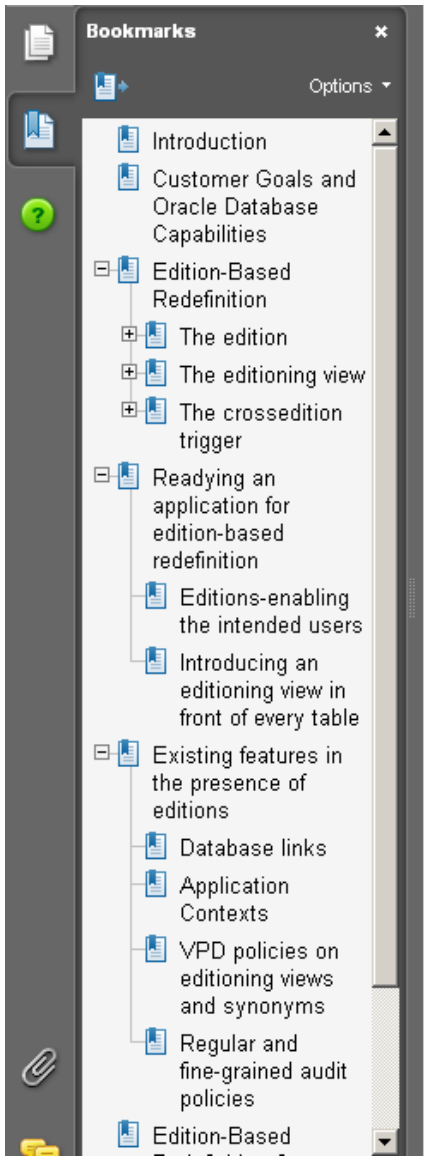
The Intersection with Academia

- Latches
  - Latches are simple, low-level serialization mechanisms that coordinate multiuser access to shared data structures, objects, and files. Latches protect shared memory resources from corruption when accessed by multiple processes. Specifically, latches protect data structures from the following situations:
    - Concurrent modification by multiple sessions
    - Being read by one session while being modified by another session
    - Deallocation (aging out) of memory while being accessed

The Intersection with Academia

- Mutexes
  - A mutual exclusion object (mutex) is a low-level mechanism that prevents an object in memory from aging out or from being corrupted when accessed by concurrent processes. A mutex is similar to a latch, but whereas a latch typically protects a group of objects, a mutex protects a single object. Mutexes provide several benefits:
    - Reduced possibility of contention
    - Because a latch protects multiple objects, it can become a bottleneck when processes attempt to access any of these objects concurrently. By serializing access to an individual object rather than a group, a mutex increases availability
    - A mutex consumes less memory than a latch
    - When in shared mode, a mutex permits concurrent reference by multiple sessions

Bryn Llewellyn's White Papers

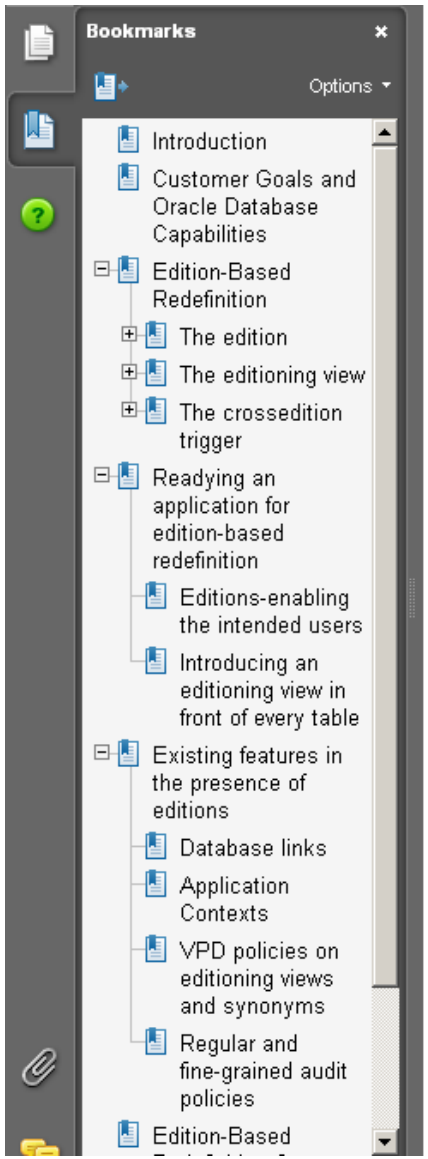


An Oracle White Paper
July 2009

Edition-Based Redefinition

a new capability in Oracle Database 11g Release 2
to support online application upgrade

Bryn Llewellyn's White Papers



Bryn started off doing image analysis and pattern recognition at Oxford University (programming in FORTRAN!) and then worked in Oslo, first at the Norwegian Computing Center, then in a startup, and then at the Norwegian Institute for Public Health. In Norway, Bryn programmed in Simula (its inventors were his close colleagues) a language recognized as the first object-oriented programming language and the inspiration for Smalltalk and C++.

An Oracle White Paper
July 2009

Edition-Based Redefinition

a new capability in Oracle Database 11g Release 2
to support online application upgrade

Bryn Llewellyn's White Papers

Library News

- Morgan's ~~Notepad~~ vi (Blog)
- Join the Western Washington OUG
- Morgan's Oracle Podcast
- DBA Best Practice Guidelines
- Bryn Llewellyn's PL/SQL White Paper
- Bryn Llewellyn's Editioning White Paper
- Troubleshooting Performance

Doing SQL from PL/SQL: Best and Worst Practices

An Oracle White Paper
September 2008

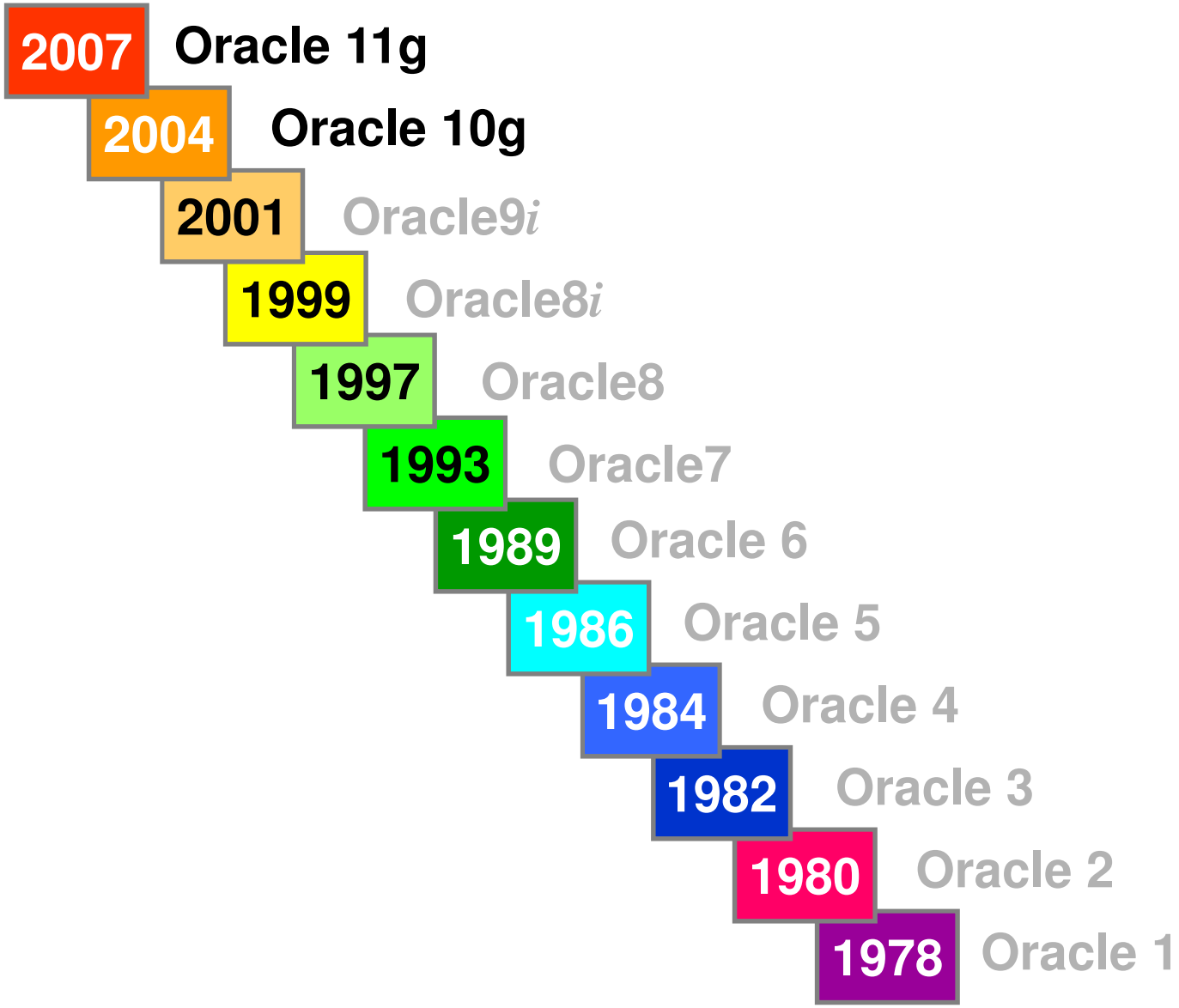
PL/SQL-specific syntax.

```
-- Code 3
<<b>>declare
  Some_Value t.PK%type := 42;
  The_Result t%rowtype;
begin
  select    a.*
  into      b.The_Result
  from      t a
  where     a.PK = b.Some_Value;

  The_Result.v1 := 'New text';

  update  t a
  set     row = b.The_Result
  where   a.PK = b.The_Result.PK;

  The_Result.PK := -Some_Value;
  insert  into t
  values  The_Result;
end;
```

Resolution of names in embedded SQL statements.

New in 11g for Developers

Oracle Server History



New Features for Developers

- Oracle Database introduced a large number of new features. Among the most notable:
  - ADR (UDUMP and BDUMP are gone)
  - Security
    - Alter Database Link
    - Case Sensitive Passwords
    - Execute Grant on Directories
  - Triggers
    - Compound Triggers
    - Follows Clause
  - Partitioning Options
    - Partition by Interval, Reference, System, Virtual Column
    - Many new composite partitioning options
  - Storage Options
    - Advanced Compression
    - SecureFile Tablespaces (in 11.2 also known as DBFS)

New in11gR1

- Segments
  - Invisible Indexes
  - Nosegment Tables
  - Read Only Tables
  - Virtual Columns
- SQL Features
  - New Analytic Functions
  - New XML Functions
  - Pivot and Unpivot Operators
  - Recursive Common Table Expression (WITH statement queries)
  - Regular Expression Enhancements
- PL/SQL Features
  - Continue Statement
  - Edition Based Redefinition
  - Improved Native Compilation & SIMPLE_INTEGER Data Type
  - New PL/SQL Warnings
  - Pragma Inline
  - Skip Locked Syntax in SELECT FOR UPDATE

New in11gR1

- Many new packages, objects, and parameters in
  - DBMS_CONNECTION_POOL — Connection Pool Management
  - DBMS_CUBE — Cube Materialized Views
  - DBMS_DATAPUMP — Legacy Mode
  - DBMS_FLASHBACK — Transaction Backout
  - DBMS_NETWORK_ACL_ADMIN — Access Control Lists
  - DBMS_RESULT_CACHE — Result Cache
  - DBMS_SCHEDULER — File Watcher
  - DBMS_WORKLOAD_CAPTURE — Workload Capture
  - DBMS_WORKLOAD_REPLAY — Workload Replay
  - DBMS_XPLAN — Explain Plan Reports

Execute for Directory Objects

- In 10g we granted READ and/or WRITE
- But this also allowed executing the ORACLE_LOADER access driver
- Now only a user that has been given EXECUTE access to the directory object is allowed to run programs in it

```
CREATE DIRECTORY stage on /home/oracle/stage

GRANT read ON stage;

GRANT write ON stage;

GRANT execute ON stage;
```

SQL

- New Analytic Functions
  - LISTAGG
  - NTH_VALUE
  - RESPECT or IGNORE NULLS

```
SELECT department_id "Dept.",
LISTAGG(last_name, '; ') WITHIN GROUP (ORDER BY hire_date) "Employees"
FROM employees
GROUP BY department_id;

SELECT prod_id, channel_id, MIN(amount_sold),
NTH_VALUE(MIN(amount_sold), 2)
  OVER (PARTITION BY prod_id ORDER BY channel_id ROWS
  BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) NV
FROM sales
WHERE prod_id BETWEEN 13 AND 16
GROUP BY prod_id, channel_id;

FIRST_VALUE(<expression> [<RESPECT | IGNORE> NULLS]) OVER (<analytic clause>)
```

SQL

- IGNORE_ROW_ON_DUPKEY_INDEX Hint
  - To ignore collisions and insert the rows that do not collide with existing rows, put the hint in the INSERT statement
- DATABASE_ROLE constant for SYS_CONTEXT
  - PRIMARY
  - PHYSICAL STANDBY
  - LOGICAL STANDBY
  - SNAPSHOT STANDBY

```
SELECT sys_context('USERENV', 'DATABASE_ROLE') FROM dual;
```

SQL

- Recursive Queries
  - CONNECT BY
- Recursive Common Table Expressions (WITH)

```
WITH <alias> AS
  (subquery_sql_statement)
  SEARCH <BREADTH | DEPTH>
  FIRST BY <column_name> [ASC | DESC] [NULLS FIRST | NULLS LAST]
  SET <ordering_column>
  CYCLE (alias) SET <cycle_mark_alias>
  TO <cycle_value> DEFAULT <no_cycle_value>
SELECT <column_name_list>
FROM <alias_one>, <alias_two>
WHERE <join_condition>;
```

PL/SQL

- New NO_DATA_NEEDED Predefined Exception
  - ORA-06548: For parallel access and pipelined table functions
    The caller of a pipelined function does not need more rows to be produced by the pipelined function.
- Improved fine grained dependency tracking
- Warnings
  - Severe
    - 5018 - omitted optional AUTHID clause
    - 5019 - deprecated language element
    - 5020 - parameter name must be identified
  - Informative
    - 6016 - native code generation turned off (size/time)
    - 6017 - operation will raise an exception
    - 6018 - an infinity or NaN value computed or used
  - Performance
    - None

Edition Based Redefinition "EBR"

Why Do We Need EBR?

- Application upgrades need to:
  - Not perturb users
  - Not corrupt data
  - Reflect all pre-upgrade transactions
  - Seamlessly roll changes forward or backward

  - To be safe
  - To be secure
  - To be fully supported by Oracle
  - No cost included in EE, SE, and SE1

What is EBR?

- A revolutionary new capability
  - Code changes are installed in the privacy of an edition
- Editionable object types
  - Synonyms
  - Views
  - PL/SQL objects of all kinds
- New kinds of object
  - Edition
    - dba_editions, dba_edition_comments
  - Editioning View
    - dba_editioning_views, dba_editioning_views_ae, dba_editioning_view_cols
  - Crossedition Trigger
    - dba_triggers

Three New Object Types

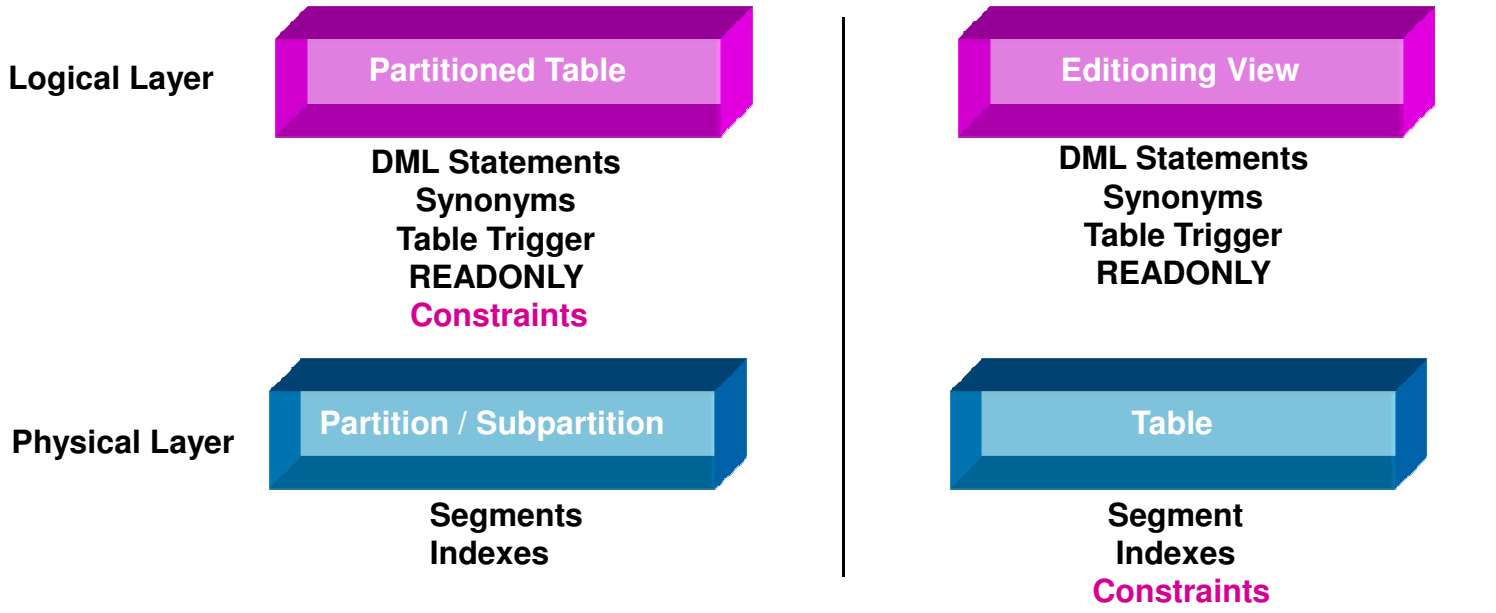
- Edition
  - All pre-upgrade editionable objects are part of a parent edition
  - New editions inherit (by pointer) editionable objects from the parent edition
  - All post-edition editionable objects are part of a child edition
- Editioning View
  - Exposes a different projection of a table into each edition
  - Allows each edition to see only its own columns
  - Data changes are made safely by writing only to new columns or new tables not seen by the old edition
  - Allows different "table" triggers to fire in each edition
- Crossedition Trigger
  - Propagates data changes made by the parent edition into the child edition's columns, or (in hot-rollover) *vice-versa (FORWARD and REVERSE)*.

What is an Edition?

- A nonschema object, uniquely, identified by only its name
- Like another non-schema object, the directory, is listed in DBA_OBJECTS as owned by SYS but has no owner
- Every database from 11.2 onwards, whether brand new or the result of an upgrade from an earlier version, non-negotiably, has at least one edition
- The default edition name is ORA$BASE
- Every foreground database session, at every moment throughout its lifetime, non-negotiably, uses a single edition
- A new edition must be the child of an existing edition
- A child edition is all that is required if an upgrade involves only synonyms, views, and PL/SQL objects

What is an Editioning View?

- A view that you may think of a partitioned table that can only have a single partition
  - Both must present all data "as is" ... no filters, no joins, no functions, no operators, no group by no having no order by no distinct no concatenation no ... no ... no ... no ....
  - Your only choice is which columns to select



If you can not do it in partitioning a table you can not do it in an editioning view

Advanced and Hybrid Columnar Compression

Hybrid Columnar Compression on Exadata V2

Warehouse Compression

- 10x average storage savings
- 10x reduction in Scan IO

Optimized for Speed

Archive Compression

- 15x average storage savings
  - **Up to 70x on some data**
- Some access overhead
- For cold or historical data

Optimized for Space

Totally application transparent

How Traditional Compression Works

- A slightly oversimplified "how it works"
  1. Oracle examines full blocks for duplicates
  2. Creates a symbol that is stored in the block header
  3. Rewrites the block substituting the symbol for the values it represents
- Compression is performed at the block level

City	State	Postal Code
Hot Springs National Park	AR	71901
Hot Springs National Park	AR	71902
Hot Springs National Park	AR	71903
Hot Springs National Park	AR	71913

128 btyes

City	State	Postal Code
Hot Springs National Park	AR	71901
"	"	"02
"	"	"03
"	"	"13

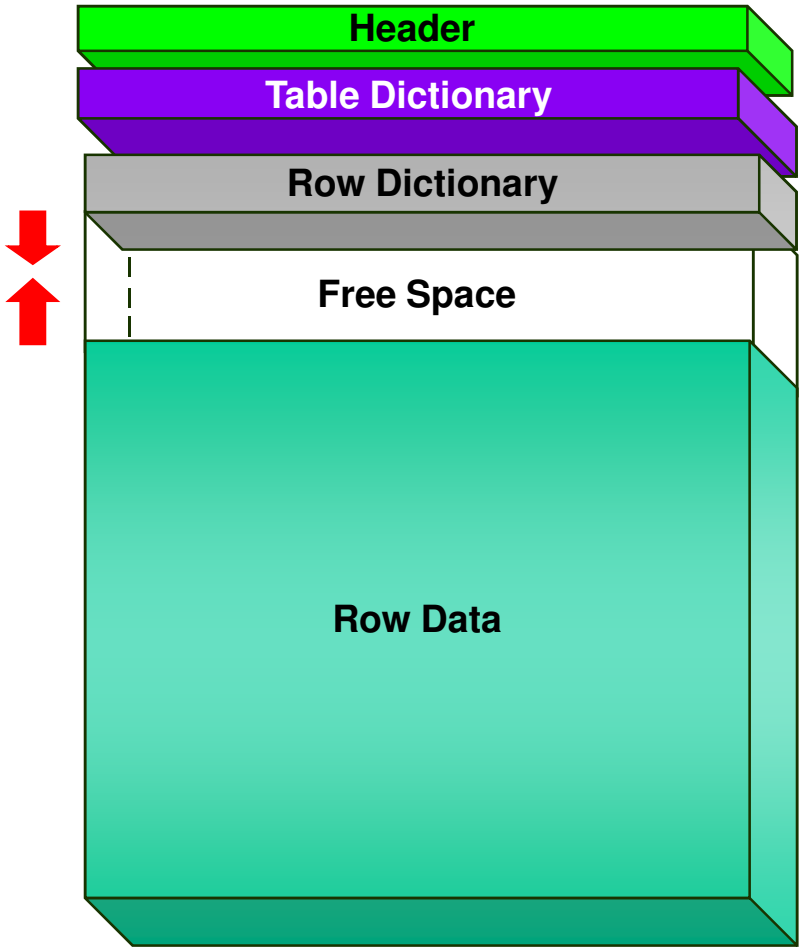
38 btyes

9.2 Data Segment Compression

- Heap Organized Tables
- Materialized Views

```
CREATE TABLE reg_tab AS
SELECT *
FROM dba_tables;

CREATE TABLE COMPRESS comp_tab AS
SELECT *
FROM dba_tables;

exec dbms_stats.gather_table_stats(USER, 'REG_TAB');
exec dbms_stats.gather_table_stats(USER, 'COMP_TAB');

SELECT table_name, blocks
FROM user_tables
WHERE table_name LIKE '%TAB';

SELECT table_name, blocks FROM user_tables WHERE table_name LIKE '%TAB';

TABLE_NAME                       BLOCKS
------------------------------ ----------
REG_TAB                             109
COMP_TAB                             20
```

Database Block Anatomy



General block information

Table info in clusters

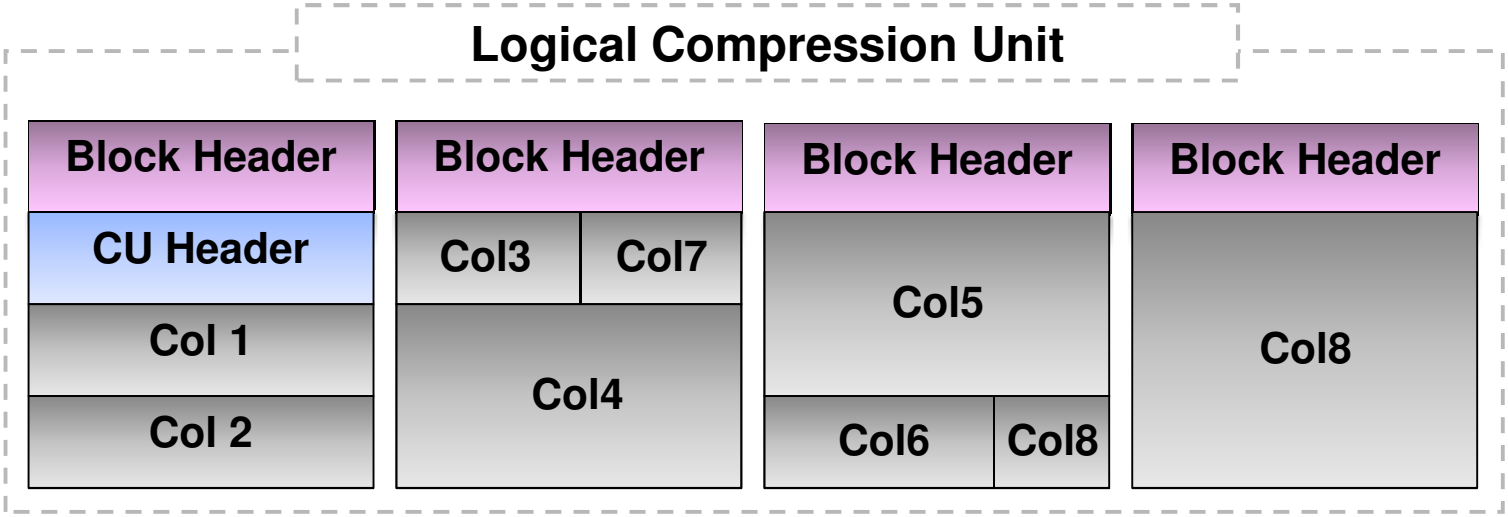
Row info in block (2 bytes/row)

Used when a new row is inserted or an existing row is updated

Segment data (table or index)

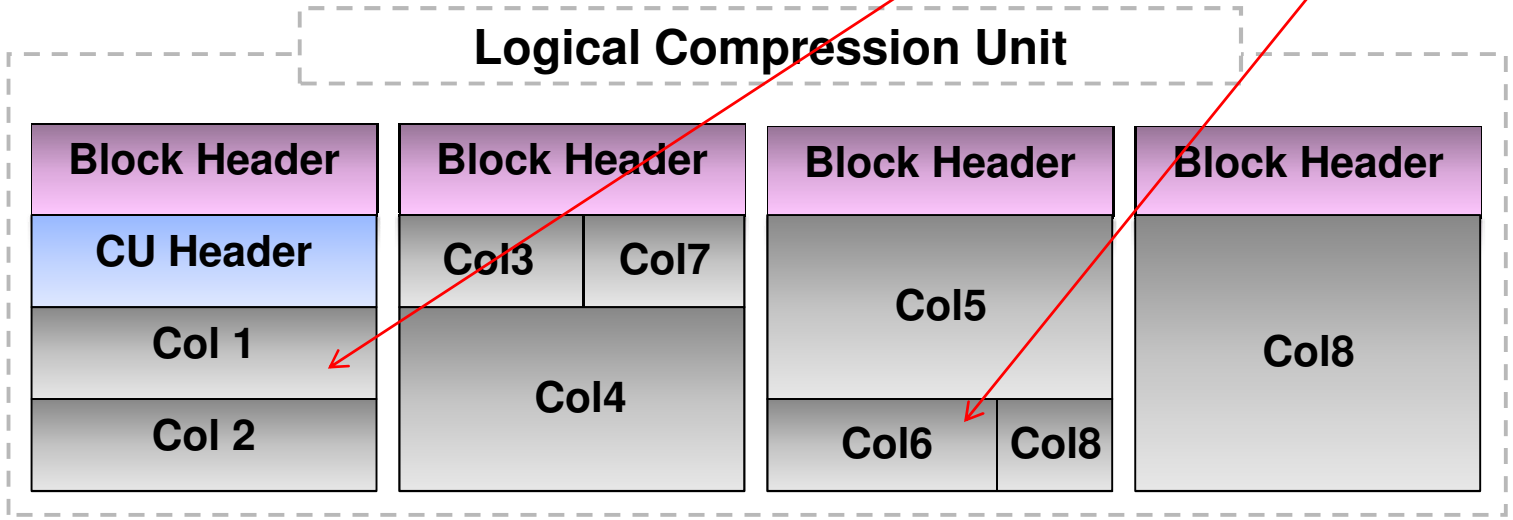
Logical Compression Unit

- Tables are organized into Compression Units (CU)
  - CU's are logical structure spanning multiple database blocks
    - Typically 32K - (4 x 8K block size)
  - Data is organized by column during data load
  - Each column is compressed separately
  - Column organization brings similar values close together



Logical Compression Unit

```
CREATE TABLE demo (
person_id      NUMBER(10),
first_name     VARCHAR2(20),
mid_initial    VARCHAR2(4),
last_name      VARCHAR2(35),
date_of_birth  DATE,
hire_date      DATE,
status         VARCHAR2(5),
comments       VARCHAR2(500));
```



Resources

- http://tahiti.oracle.com
- http://otn.oracle.com
  - downloads
  - forums
- http://www.morganslibrary.org
  - library
  - how can I?
  - code samples
  - links

Questions

ERROR at line 1:

ORA-00028: your session has been killed

All demos at morganslibrary.org

- **Library**
- **How Can I?**

damorgan11g@gmail.com

Thank you