

Oracle12c New Features

Dan Hotka

Author/Instructor

Oracle Ace Director 



www.DanHotka.com, LLC

(c) www.danhotka.com LLC.

Any reproduction or copying of this manual without the express written consent of www.danhotka.com LLC is expressly prohibited.

Limitation on Warranty. THERE ARE NO WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT THERETO, INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. PURCHASER SHALL BE SOLELY RESPONSIBLE FOR THE SELECTION, USE, EFFICIENCY AND SUITABILITY OF USE OF INFORMATION CONTAINED HEREIN TO ANY PARTICULAR APPLICATION OR PROBLEM. WWW.DANHOTKA.COM LLC SHALL HAVE NO LIABILITY THEREFOR.

This presentation and/or courseware make reference to Oracle Corp and Quest Software product features that you may or may not be licensed to use at your site (products such as TOAD itself, Advanced Workload Repository, SQL Tuning Advisor, etc). Please consult your contracts and purchase agreements before using any of the product features discussed during this course.

Limitation of Liability. IN NO EVENT SHALL WWW.DANHOTKA.COM LLC BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING, WITHOUT LIMITATION, ANY DAMAGES RELATING TO LOSS OF DATA, AND ANY INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES OR LOST PROFITS, ARISING OUT OF OR IN ANY WAY RELATED TO YOUR USE OF THE INFORMATION CONTAINED IN THIS MANUAL. IN THE EVENT THAT THE FORGOING IS HELD UNINFORCABLE THE PARTIES AGREE THAT WWW.DANHOTKA.COM LLC'S LIABILITY TO YOU HEREUNDER, IF ANY, SHALL IN NO EVENT EXCEED THE FEE PAID BY THE INJURED PARTY FOR THE MANUAL TO WWW.DANHOTKA.COM. LLC.

Dan Hotka
Author/Instructor/Oracle Expert
CEO
DHotka@Earthlink.net
515 279 3361

Dan is a Training Consultant

❖ Dan Hotka

– Oracle ACE Director

- Oracle ACEs and Oracle ACE Directors are known for their strong credentials as Oracle community enthusiasts and advocates, with candidates nominated by anyone in the Oracle Technology and Applications communities.
- www.oracle.com/technology/community/oracle_ace/index.html

– Oracle Authored Expert

- 34 Years in IT – 28 years working with Oracle
- 12 books – hundreds of articles

❖ Register for my quarterly Newsletter

- Full of tips and techniques, downloads, book reviews, etc

❖ DanHotka.Blogspot.com

- I also blog on TOADWorld.com

❖ Training at your Desk!

- Public Web-based training
- Half day lectures with hands-on lab exercises
- Almost like being there!

❖ Flat Fee Training for your company:

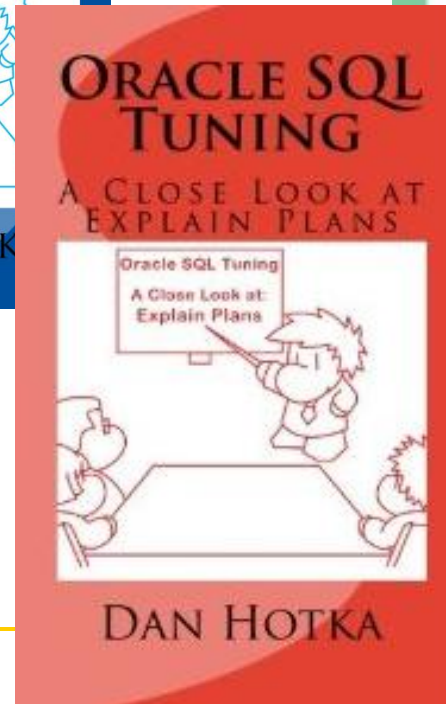
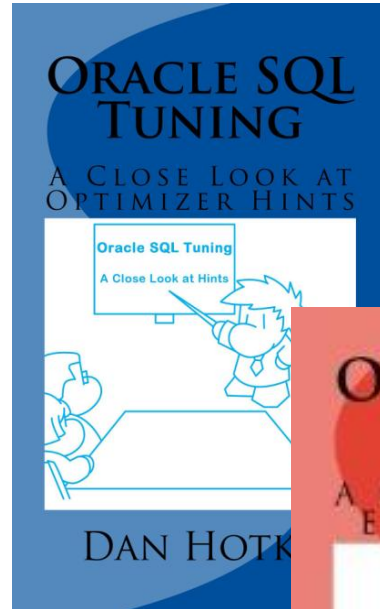
- 1 Course Fee Price
 - Price includes my portable computer lab!
- On-site or over the Web!
- Portable computer lab

❖ Training Courses Include:

- Oracle12 New Features!!!!
- Oracle Advanced PL/SQL
- Oracle SQL Tuning
- TOAD Courses
- APEX – Admin, Developers, Installation
- Oracle OBIEE – Admin & Queries/Reports
- Discoverer, Intro courses, Cross training!

www.Amazon.com

- ❖ Lecture Notes
- ❖ Useful Hints
- ❖ Great info for TOAD users
- ❖ Available in Kindle format too!
- ❖ www.Amazon.com



Additional Reading

- ❖ **Oracle12c New Features**
 - By Robert Freeman;
 - Available in the book store???
 - www.Amazon.com





Agenda

❖ Oracle New Features

- Pluggable Database
- New Table/Index Partitioning
- More New Features
 - Full Transportable Export/Import
 - Temporary Undo
 - Multiple Indexes using same columns
 - Transaction Guard

Agenda

❖ New SQL Syntax

- New Analytical Functions
- New SQL Features
- New Options for Sequences
- Enhanced Join Syntax
- Increased column size limits



Agenda

❖ New PL/SQL Features

- PL/SQL And SQL
- Enhanced DBMS_UTILITY
- New UTL_CALL_STACK
- Enhanced DBMS_SQL
- Enhanced Native SQL





Audience Background

❖ A few questions:

- What Oracle Database are you running?
- When do you plan to go to Oracle12?
- Working with Oracle12 Now?
- What is your Oracle background?
 - Newbee 1 – 3 years 4 – 8 years 8+ years?
- What is your main role?
 - Power User Analyst Developer DBA Management?



#1 Pluggable DB



Pluggable Database

❖ Pluggable Database

– The 'c' in Oracle12

- Container

- Pluggable databases come into the environment as a database that 'contains' other database

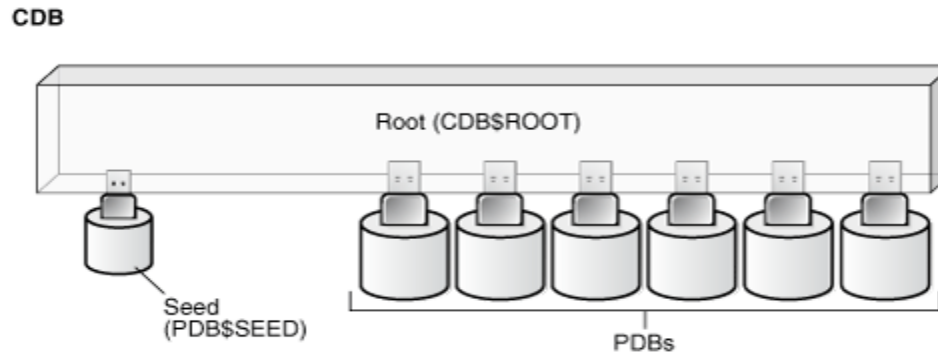
- Consolidate

- Allows for the sharing of resources, backup/recovery schemes, and ease of management of similar application DBs

- Cloud

- This new environment lends itself well for the cloud computing environment

Pluggable Database



❖ How it works:

- Root
 - Contains metadata and common users
- Seed
 - A template for new PDB's
- PDB
 - The actual application DB...just like a non-CDB.



Pluggable Database

❖ How it works:

- 1 set of Oracle executables per container
 - Think of the savings in memory/processing power alone!
- Sharing of features:
 - Container setup features shared across all PDB's automatically
 - Data guard, rac,
 - Resource management (memory, disk, etc)
 - Backup
- Eases management of lots of similar database environments



Pluggable Database

❖ Support

- Supports up to 252 databases per container DB
- Users can share PDB's
 - Called common users
- Can use SQL*Plus, DBCA, Enterprise Mngr Cloud Control, SQL Developer, or Server Control to manage the CDB and PDB's.

Pluggable Database

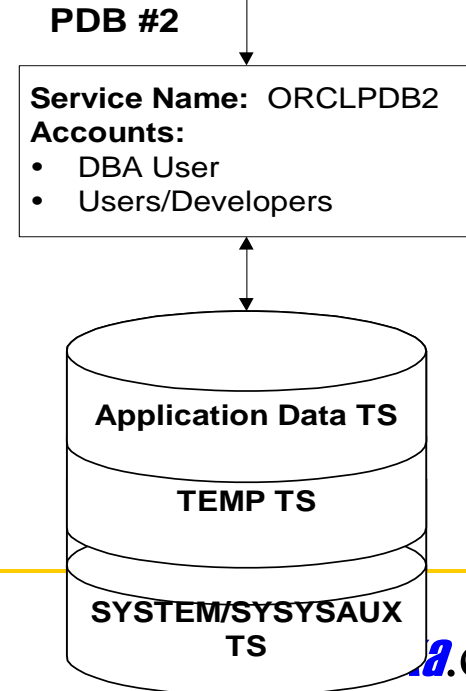
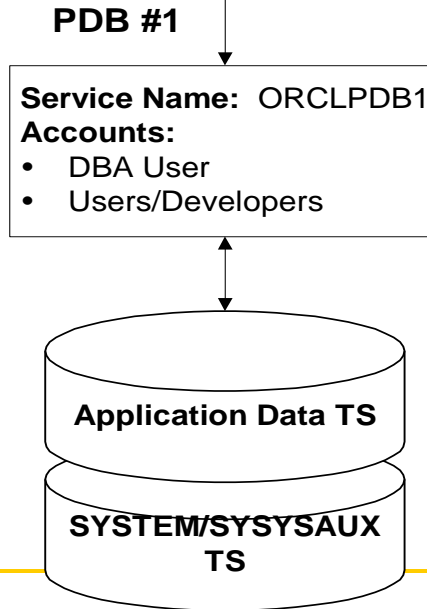
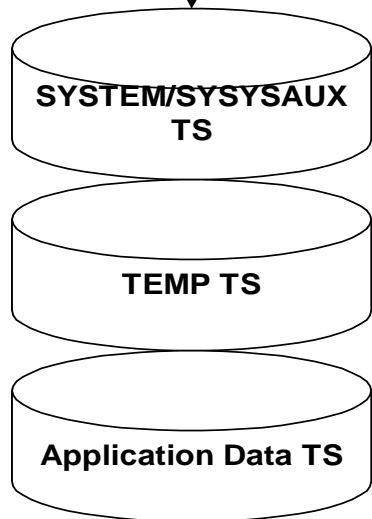
Oracle12 CDB

SID: Orcl
Service Name: ORCL
Container Name: CDB\$ROOT
Accounts:

- Sys
- System
- C## - common users

Alter Session Set container = <service name>;

upto 250
more
PDB's



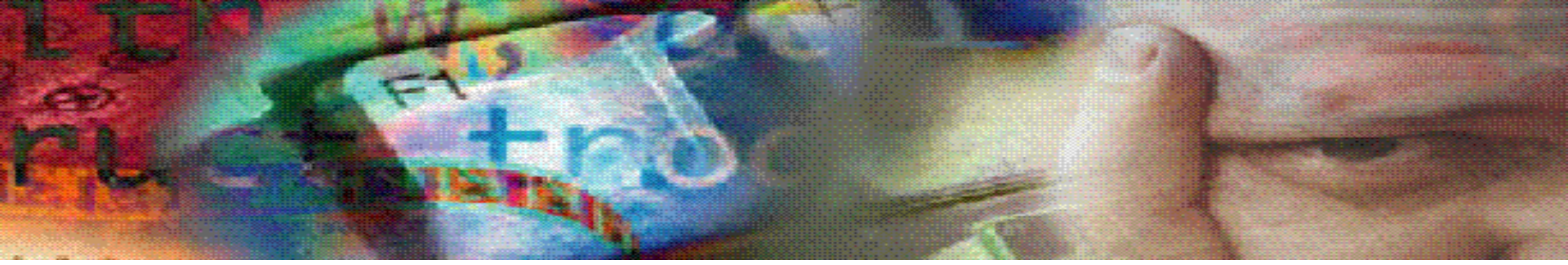


Pluggable Database

❖ Problems

- Same synonym names pointing to different objects
- Same schema names
- Supports Oracle12 DBs only

❖ **Some planning is needed to insure consistency of common names across applications**



Partitioning Enhancements

Partitioning Enhancements

❖ Moving Partitions online

- Can now move partitions and subpartitions online.

❖ Interval-Reference Partitioning

- Interval partitioning allows for automatic partitioning of values outside the range of values defined.
- Reference partitioning has child partitions off the main partition, based on foreign key which becomes the child partition key value
- Prior releases could not create an interval partition for the parent table, this restriction has been lifted



Partitioning Enhancements

❖ Cascade Functionality

- Can now truncate the reference partitioning
- A single command now removes all the old/related data
- Now an all-or-nothing operation

❖ Partition maintenance on multiple partitions

- Alter partition [add, truncate, drop, split, merge]

Partitioning Enhancements

❖ Partial Indexes

- Can now have indexes just on some partitions!
 - Oracle11...had to build the index unuseable then rebuild just the ones you wanted...
- New partition syntax 'indexing on' or 'indexing off' to define which partitions can/can't have indexes

Partitioning Enhancements

❖ Global Index Maintenance

- Drop/truncate/modify partition indexing off ALL make global indexes go 'stale'
- Easy to see now with new column ORPHANED_ENTRIES
 - Views: dba/all/user/ _INDEXES, _IND_PARTITION
 - Yes...has orphaned entries
 - No...does not have orphaned entries
 - N/A...does not apply...ie: local index or non partitioned table

Partitioning Enhancements

❖ Global Index Maintenance

- Automatic scheduled job:
SYS.PMO_DEFERREDGIDX_MAINT_JOB
 - Runs at 2am daily
- Alter index rebuild and alter index coalesce partition cleanup also restores these indexes
- Oracle12 New method
 - Dbms_part.cleanup_didx
 - Uses schema name and table name.



New SQL and PL/SQL Features

Partitioning Enhancements

❖ Moving Partitions online

- Can now move partitions and subpartitions online.

❖ Interval-Reference Partitioning

- Interval partitioning allows for automatic partitioning of values outside the range of values defined.
- Reference partitioning has child partitions off the main partition, based on foreign key which becomes the child partition key value
- Prior releases could not create an interval partition for the parent table, this restriction has been lifted



Partitioning Enhancements

❖ Cascade Functionality

- Can now truncate the reference partitioning
- A single command now removes all the old/related data
- Now an all-or-nothing operation

❖ Partition maintenance on multiple partitions

- Alter partition [add, truncate, drop, split, merge]

Partitioning Enhancements

❖ Partial Indexes

- Can now have indexes just on some partitions!
 - Oracle11...had to build the index unuseable then rebuild just the ones you wanted...
- New partition syntax ‘indexing on’ or ‘indexing off’ to define which partitions can/can’t have indexes

Partitioning Enhancements

❖ Partial Indexes

```
CREATE TABLE PRODUCTS (  
    MFGID NUMBER,  
    PRODUCTID NUMBER,  
    RECEIVED_DATE DATE)  
INDEXING OFF  
PARTITION BY RANGE (RECEIVED_DATE)  
(  
    PARTITION PART_2011 VALUES LESS THAN  
        (TO_DATE('01-JAN-2012', 'DD-MON-YYYY')) INDEXING OFF,  
    PARTITION PART_2012 VALUES LESS THAN  
        (TO_DATE('01-JAN-2013', 'DD-MON-YYYY')) INDEXING ON,  
    PARTITION PART_2013 VALUES LESS THAN  
        (TO_DATE('01-JAN-2014', 'DD-MON-YYYY'))  
)  
ENABLE ROW MOVEMENT;
```

Partitioning Enhancements

❖ Partial Indexes

- Useful scripts to see indexing across partitions

```
SELECT TABLE_NAME, PARTITION_NAME, INDEXING
FROM USER_TAB_PARTITIONS
WHERE TABLE_NAME = 'PRODUCTS'
ORDER BY PARTITION_POSITION;
```

TABLE_NAME	PARTITION_NAME	INDEXING
PRODUCTS	PART_2011	OFF
PRODUCTS	PART_2012	ON
PRODUCTS	PART_2013	OFF

```
SELECT TABLE_NAME, INDEX_NAME, INDEXING
FROM USER_INDEXES
WHERE TABLE_NAME = 'PRODUCTS'
ORDER BY 2 DESC;
```

TABLE_NAME	INDEX_NAME	INDEXING
PRODUCTS	PRODUCTS_PARTIAL	PARTIAL
PRODUCTS	PRODUCTS_FULL	FULL

Partitioning Enhancements

❖ Global Index Maintenance

- Drop/truncate/modify partition indexing off ALL make global indexes go 'stale'
- Easy to see now with new column ORPHANED_ENTRIES
 - Views: dba/all/user/ _INDEXES, _IND_PARTITION
 - Yes...has orphaned entries
 - No...does not have orphaned entries
 - N/A...does not apply...ie: local index or non partitioned table

Partitioning Enhancements

❖ Global Index Maintenance

- Automatic scheduled job:
SYS.PMO_DEFERREDGIDX_MAINT_JOB
 - Runs at 2am daily
- Alter index rebuild and alter index coalesce partition cleanup also restores these indexes
- Oracle12 New method
 - Dbms_part.cleanup_didx
 - Uses schema name and table name.

Partitioning Enhancements

❖ Automatic Global Stats

- Oracle12 gives some control over how partition stats are determined to be stale
- Tables with incremental set to true...ANY DML makes stats stale
- Exec `DBMS_STATS.set_table_prefs(<schema>, <table>, INCREMENTAL_STALENESS, value)`
 - Value = NULL - default behavior
 - `USED_LOCKED_STATS` - stats never go stale
 - `USE_STALE_PERCENT` – stats stale after this % of DML
 - And `stale_percent` can be adjusted using similar syntax



New Limits

❖ Increased column size limits

- IF MAX_STRING_SIZE = EXTENDED
 - Varchar2 – new limit 32K
 - NVarchar2 – new limit 32K
 - RAW – new limit 32K
- MAX_STRING_SIZE = STANDARD (default behavior)
 - Varchar2 –limit 4K
 - NVarchar2 –limit 4K
 - RAW – limit 2K



New Limits

Database Type	Current Table Limit	Current PL/SQL Limit	Oracle12c New Limit
Varchar2	2K	4K	32K
Nvarchar2	2K	4K	32K
Raw	2K	4K	32K



New Limits

❖ New init.ora setting: Max_String_Size

- Set to EXTENDED
 - Allows new limits
- Set to STANDARD
 - Old limits used
- NO Going Back!!!!
- Indexed column max size 6400 Bytes
 - Make sure no indexes are using long varchar columns



New SQL Syntax

❖ New SQL Syntax

- New Analytic SQL Features
- New SQL Features
- New Functions
- New Options for Sequences
- Enhanced Join Syntax
- Increased column size limits



New SQL Syntax

❖ New Analytical SQL Features

- Clustering works with groupings of survey data in BI
- Cluster_Details New Function
 - Returns xml string cluster details for each row selected
- Cluster_Id Function Enhanced
 - Can use either a pre-defined clustering model or perform dynamic clustering
- Cluster_Distance New Function
 - Predicts cluster membership.



New SQL Syntax

❖ New SQL Syntax

- Table Syntax (create or alter)
 - Period_Definition – temporary validity (start and end date) for which each row in the table is valid
 - Visible/Invisible
 - Truncate now has a cascade feature
- Partitioning
 - A plethora of changes – Attend the seminar or class...



New SQL Syntax

❖ New SQL Syntax

– Index Syntax

- Drop index online
- Can set UNUSED online now
- Drop a constraint online as well
- Partitioning options with indexes covered with the Table partitioning new features
- Indexes can be on the same set of columns IF:
 - The indexes are of different types/partitioning/uniqueness
 - ONLY one of the indexes is VISIBLE at any given time



New SQL Syntax

❖ New Options for Sequences

- KEEP and NOKEEP feature added
 - Works with database Replay
 - Controls if NEXTVAL retains its original value (keep feature) or not (nokeep feature)
- Can be used as Table column default values!



New SQL Syntax

❖ New SQL Syntax

– VIEW Enhancements

- Columns can be made visible and invisible as well
- BEQUEATH option to override default Invokers rights

New SQL Syntax

❖ Functions in WITH Clause

- Now can have a PL/SQL function as part of a WITH clause

WITH

```
FUNCTION get_domain(url VARCHAR2) RETURN  
VARCHAR2 IS
```

```
    pos BINARY_INTEGER;
```

```
    len BINARY_INTEGER;
```

```
BEGIN
```

```
    pos := INSTR(url, 'www.');
```

```
    len := INSTR(SUBSTR(url, pos + 4), '.') - 1;
```

```
    RETURN SUBSTR(url, pos + 4, len);
```

```
END;
```

```
SELECT DISTINCT get_domain(catalog_url)
```

```
    FROM product_information;
```

```
/
```

New SQL Syntax

❖ New SELECT Syntax

- Pattern Matching enables patterns to be found across multiple rows

```
SELECT *
FROM Ticker MATCH_RECOGNIZE (
    PARTITION BY symbol
    ORDER BY tstamp
    MEASURES STRT.tstamp AS start_tstamp,
             LAST(DOWN.tstamp) AS bottom_tstamp,
             LAST(UP.tstamp) AS end_tstamp
    ONE ROW PER MATCH
    AFTER MATCH SKIP TO LAST UP
    PATTERN (STRT DOWN+ UP+)
    DEFINE
        DOWN AS DOWN.price < PREV(DOWN.price),
        UP AS UP.price > PREV(UP.price)
    ) MR
ORDER BY MR.symbol, MR.start_tstamp;
```

New SQL Syntax

❖ New SELECT Syntax

- Row_Limiting clause works with Top-N to specify the number or percentage of rows to return
- Does not work with For Update
- Offset is 0 if not defined
 - 0 if negative number
 - Fractions truncated
- WITH TIES works with the order by clause
 - To return the % of rows including matching values of the last item

```
SELECT employee_id, last_name  
FROM employees  
ORDER BY employee_id  
OFFSET 5 ROWS FETCH FIRST 5 ROWS ONLY;
```

```
SELECT employee_id, last_name  
FROM employees  
ORDER BY employee_id  
OFFSET 5 ROWS FETCH NEXT 5 ROWS ONLY;
```

```
SELECT employee_id, last_name, salary  
FROM employees  
ORDER BY salary  
FETCH FIRST 5 PERCENT ROWS ONLY;
```

New SQL Syntax

❖ New Join Syntax

- Cross_Outer_Apply_Clause
 - Allows a variation of the ANSI CROSS JOIN or ANSI LEFT OUTERJOIN with left correlation support.
 - Specify CROSS APPLY to perform a variation of the ANSI Cross Join
 - Specify OUTER APPLY to perform a variation of the ANSI Left Outer Join



New SQL Syntax

```
SELECT d.department_name, v.employee_id, v.last_name  
FROM departments d CROSS APPLY (SELECT * FROM employees e  
WHERE e.department_id = d.department_id) v  
WHERE d.department_name IN ('Marketing', 'Operations', 'Public Relations')  
ORDER BY d.department_name, v.employee_id;
```

```
SELECT d.department_name, v.employee_id, v.last_name  
FROM departments d OUTER APPLY (SELECT * FROM employees e  
WHERE e.department_id = d.department_id) v  
WHERE d.department_name IN ('Marketing', 'Operations', 'Public Relations')  
ORDER by d.department_name, v.employee_id;
```

New SQL Syntax

❖ New Join Syntax

- Lateral clause for Inline Views
 - Allows for columns in the inline view to be accessed!

```
SELECT * FROM employees e, (SELECT * FROM departments d  
WHERE e.department_id = d.department_id);
```

```
ORA-00904: "E"."DEPARTMENT_ID": invalid identifier
```

```
SELECT * FROM employees e, LATERAL(SELECT * FROM  
departments d  
WHERE e.department_id = d.department_id);
```



New SQL Syntax

❖ New Join Syntax

- Partial Join Evaluation
- Goal is to avoid generating duplicate rows
- Replaces DISTINCT operation
- Used with aggregate functions off of UNION clauses
 - S = Semi Join which stops with the first hit
 - NA = null-aware

New SQL Syntax

```
SELECT COUNT (*)
FROM CUSTOMERS C,
WHERE C.CUST_CREDIT_LIMIT > ALL
  (SELECT T.UNIT_COST
   FROM SALES S, COSTS T
   WHERE T.PROD_ID = S.PROD_ID);
```

Oracle11 – runs for minutes

Oracle12 with Null-Aware
Join runs in 7 seconds

```
SELECT COUNT (*)
FROM CUSTOMERS C,
  (SELECT T.UNIT_COST
   FROM SALES S, COSTS T
   WHERE T.PROD_ID S= S.PROD_ID) V ←
WHERE C.CUST_CREDIT_LIMIT NA <= V.UNIT_COST;
```


New SQL Syntax

```
SELECT P.prod_id, S.quantity_sold, S.cust_id
FROM   Sales S, Product P
WHERE  P.prod_list_price > 11 and
       P.prod_id = S.prod.id and
       (S.cust_id is NULL OR
        EXISTS (SELECT 1
                 FROM   Customer C
                 WHERE  C.cust_id = S.cust_id AND
                        C.country_id = 'U.S.'));
```

Another Example:

Oracle11 syntax vs
Oracle12 syntax

```
SELECT P.prod_id, S.quantity_sold, S.cust_id
FROM   Sales S, Product P, Customer C
WHERE  P.prod_list_price > 11 and
       P.prod_id = S.prod.id and
       S.cust_id NS=C.cust_id and ←
       C.country_id = 'U.S.';
```



New SQL Syntax

❖ Increased column size limits

- IF MAX_STRING_SIZE = EXTENDED
 - Varchar2 – new limit 32K
 - NVarchar2 – new limit 32K
 - RAW – new limit 32K
- MAX_STRING_SIZE = STANDARD (default behavior)
 - Varchar2 –limit 4K
 - NVarchar2 –limit 4K
 - RAW – limit 2K



New PL/SQL Features

❖ New PL/SQL Features

- PL/SQL And SQL
- Enhanced DBMS_UTILITY
- New UTL_CALL_STACK
- Enhanced DBMS_SQL
- Enhanced Native SQL

New PL/SQL Features

❖ PL/SQL and SQL

- Same new size limits on Varchar2, Nvarchar2, and Raw
 - Same options to turn it on...
- Bulk Collect supports the Fetch First SQL Features
- PL/SQL Functions that Run Faster in SQL
 - Support for the previously discussed WITH SQL syntax
- Invisible Columns
 - CAN be directly referenced via PL/SQL
 - NOT using '*' syntax...
 - Can make columns visible/invisible too



New PL/SQL Features

- ❖ **Easily another 25 or so packages**
 - Not time to cover all here
 - Coverage for PDB's, etc
- ❖ **DBMS_Utility**
 - Expand_SQL_Text
 - Easily exposes the SQL contained in a view
- ❖ **UTL_Call_Stack**
 - Provides info on currently executing sub programs
 - Info includes name, owner, and line numbers
 - Enhanced Native SQL



New PL/SQL Features

❖ UTL_Call_Stack

- Provides info on currently executing sub programs
 - Info includes name, owner, and line numbers
- Programs include:
 - Backtrace_Depth _Line _Unit
 - Error_Depth _Msg _Number
 - Unit_Line
 - Subprogram



New PL/SQL Features

- ❖ **DBMS_SQL.Parse**

- Now has a schema input parameter

- ❖ **Enhanced Native SQL**

- Also now supports the Fetch First SQL feature

What have we learned?

- ❖ **New Database Features**
- ❖ **New SQL Syntax**
- ❖ **New PL/SQL Features**
- ❖ **Keep in touch**
 - Dan@DanHotka.com

