Seminar:

Oracle Database 12c
New Features for Developers

Presenter:

Oren Nakdimon
www.db-oriented.com
✉ oren@db-oriented.com
☎ 054-4393763
@DBoriented

Who Am I?

- Chronology by "Oracle years"

When		What	Where
Oracle 6/7	1991-1997	Developer	IAF
Oracle 8	1997-1998	Server group manager	Golden Screens
Oracle 8i/9i	1998-2003	DBA group manager	TELEknowledge
Oracle 10g/11g	2004-2011	VP R&D and Israel Site Manager	Olista

- Today: dbORIENTED
  - Freelance database expert
    - Database Administration
    - Development
    - Courses

> ***"An expert is a person who has made all the mistakes that can be made in a very narrow field"***
>
> **Niels Bohr (1885-1962)**

Agenda

- Schema-level new features
- Optimizer new features
- PL/SQL new features
- Easier migration to Oracle
- SQL new features
- Utilities new features
- Transaction Guard
- ILM (information lifecycle management)
- Partitioning new features

One Thing Before We Start…

Known Bugs in 12.1

- http://docs.oracle.com/cd/E16655_01/readmes.121/e17908/toc.htm#READM160

Probably not the most important new feature

Last Login

Last Login

```
C:\Users\orenn>sqlplus

SQL*Plus: Release 12.1.0.1.0 Production on Sat Nov 9 08:24:09 2013

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Enter user-name: hr
Enter password:
Last Successful login time: Fri Nov 08 2013 17:26:27 +02:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL>
```

Schema Level New Features

Invisible Columns

Invisible Columns

- Invisible columns are invisible, unless explicitly specified
- For example, they will not appear in:
  - SELECT * FROM…
  - INSERT with no column list
  - %ROWTYPE

Invisible Columns

- May be created as invisible
- May be modified from visible to invisible and vice versa
- When an invisible column becomes visible, it appears as the last column of the table

Invisible Columns

- In SQL*Plus
  - SET COLINVISIBLE controls inclusion or exclusion of invisible columns in DESCRIBE
- Invisible columns can be also:
  - Virtual
  - Partitioning key
- Cannot contain invisible columns:
  - External tables
  - Temporary tables
  - Clustered tables

Schema Level New Features

Column Defaults

Column Defaults – Before 12c

- No Sequence/AutoNumber/Identity
- Nothing like NVL(value,DEFAULT)
- Adding an optional column with default is an offline operation

Column Defaults in 12c – Sequence

- A column can be associated with a sequence

Column Defaults in 12c – On Null

- A default can be defined for explicit Null values
- The column is implicitly defined as NOT NULL

Column Defaults in 12c – Identity

- A column can be defined as “identity”
  - Implicit sequence
  - Implicit NOT NULL
- GENERATED…
  - [always] as identity
  - by default as identity
  - by default on null as identity
- You need the CREATE SEQUENCE privilege

Column Defaults in 12c – Identity

- Configuring the implicit sequence
  - Like in CREATE SEQUENCE
  - START WITH LIMIT VALUE
- Restrictions
  - Only for numeric data types
  - Maximum one identity column per table
  - Non-identity column cannot be modified to identity column
  - CTAS ignores the identity definition

Column Defaults in 12c – New Optional Column

- As of 11g, adding a mandatory column with default is a meta-data only operation:
  - Fast
  - Online
  - No space
  - No redo
  - No undo
- As of 12c, the same is true also for optional columns

Schema Level New Features

Extended Strings

Extended Strings

- By default, maximum length of:
  - VARCHAR2 and NVARCHAR2 columns is 4KB
  - RAW columns is 2KB
- If MAX_STRING_SIZE = EXTENDED, it is 32KB
- Implemented like LOBs
- SQL string functions can return 32KB

Extended Strings – Enabling

- MAX_STRING_SIZE can be changed from STANDARD to EXTENDED, but not from EXTENDED to STANDARD
- COMPATIBLE must be set to 12.0.0.0 or higher
- Can be done only in UPGRADE mode
- The script utl32k.sql must be executed
- Objects may be invalidated
- Instructions: http://docs.oracle.com/cd/E16655_01/server.121/e17615/refrn10321.htm

Schema Level New Features

Multiple Indexes on the Same Column List

Invisible Indexes (11g)

- An invisible index is:
  - Maintained by DML
  - Invisible to the optimizer
    - Unless optimizer_use_invisible_indexes is true

Multiple Indexes on the Same Column List

- Before 12c
  - Impossible to define several indexes on the same column list
- In 12c
  - It is possible, as long as
    - Only one of the indexes is visible
    - The indexes are different somehow; for example:
      - Unique vs. non-unique
      - B*tree vs. bitmap
      - Partitioned vs. non-partitioned

Schema Level New Features

Session Level Sequences

Session Level Sequences

- A new type of sequence
- Returns a range of numbers which is unique within a session
- Non persistent

Schema Level New Features

Temporary Undo

Temporary Undo

- Undo for temporary tables can be stored in the temporary tablespace
- As a result:
  - No redo for undo of DML on temporary tables
  - Increased “real” undo retention
  - Ability to use temporary tables for read and write in read-only Active Data Guard
- TEMP_UNDO_ENABLED = true | false

Optimizer New Features

Histograms

Histograms – Before 12c

- Frequency histogram
  - "All values created equal"
- Height balanced histogram
  - May miss popular values
- Up to 254 buckets

Histograms – in 12c

1. Frequency histogram
   - When #(distinct values) ≤ #buckets
2. Top frequency histogram
   - When #(dominant distinct values) ≤ #buckets and estimate_percent=AUTO_SAMPLE_SIZE
3. Hybrid histogram
   - When estimate_percent=AUTO_SAMPLE_SIZE
   - Values do not cross bucket boundaries
   - "Endpoint repeat count" is saved
4. Height-balanced histogram

- Up to 2048 buckets (default is still 254)

Optimizer New Features

Online Statistics Gathering

Online Statistics Gathering

- Index statistics are gathered during index creation/rebuild (for many versions now)
- In 12c, table and column statistics are also gathered automatically: during direct path load into empty objects
  - Empty table
  - Empty partition (the partition should be explicitly specified)

Online Statistics Gathering

- Can be disabled at statement level:

  /*+ NO_GATHER_OPTIMIZER_STATISTICS */

- Histogram and index statistics are not gathered as part of this
- To gather only them:
  - dbms_stats.gather_table_stats with options=>'GATHER AUTO'

Optimizer New Features

Dynamic Statistics

Dynamic Statistics (formerly Sampling)

- New level: 11
- Enables the optimizer to automatically decide to use dynamic statistics for any SQL statement
- Results are persisted in the cache

Optimizer New Features

Global Temporary Tables

Global Temporary Tables

- Structure – shared
- Data – private (session level)
- Statistics gathering
  - Before 12c: shared
  - In 12c: either session-level (the default) or shared
    - GLOBAL_TEMP_TABLE_STATS = SESSION | SHARED

Optimizer New Features

Concurrent Execution of Union/All

Concurrent Execution of Union/All

- Before 12c
  - Set operators are processed in a sequential manner
- In 12c
  - Concurrent execution of UNION is possible (if at least one of the branches is considered being processed in parallel)
  - If OPTIMIZER_FEATURES_ENABLED < 12.1 use /*+ PQ_CONCURRENT_UNION */
  - Disable concurrency by /*+ NO_PQ_CONCURRENT_UNION */

PL/SQL New Features

White Lists

PL/SQL White Lists

- The ACCESSIBLE BY clause can be added to packages, procedures, functions and types to specify which objects are able to reference the PL/SQL object directly
  - In packages – only at the top-level of the spec
- ACCESSIBLE BY:
  - Object name
  - Object type and object name
- The white list objects do not have to exist

PL/SQL New Features

Invoker's Rights

Granting Roles to Program Units

- Roles can be granted to program units
- It allows different units of the same owner having different privileges
- Note: the unit's owner must be granted these roles itself (but they can be disabled)

BEQUEATH

- The BEQUEATH clause specifies whether functions referenced in the view are executed using the view invoker's rights or the view definer's rights
- Name resolution within the view is still handled using the view owner's schema
- Privilege checking for the view is still done using the view owner's privileges

INHERIT PRIVILEGES

- When executing an invoker's rights procedure, the owner gets access to the invoker's privileges
- In 12c, the owner must be granted INHERIT PRIVILEGES on the invoker (or INHERIT ANY PRIVILEGES) to achieve this
- By default PUBLIC has INHERIT PRIVILEGES on all newly created and upgraded user accounts

Function Result Cache

- Before 12c only definer's rights functions could be RESULT_CACHE-ed
- In 12c, invoker's rights functions can be RESULT_CACHE-ed too
  - The current user becomes part of the cache lookup key

PL/SQL New Features

SQL Text Expansion

SQL Text Expansion

- DBMS_UTILITY.EXPAND_SQL_TEXT reveals the actual SQL executed for a given query
- For example:
  - Views are replaced by their underlying queries
  - VPD driven conditions are added to the query

PL/SQL New Features

Introspection

Introspection – Before 12c

- Functions in DBMS_UTILITY:
  - FORMAT_CALL_STACK
  - FORMAT_ERROR_STACK
  - FORMAT_ERROR_BACKTRACE
- One big string
- Up to 2000 bytes
- Coarse-grained resolution

Introspection – in 12c

- Added: the UTL_CALL_STACK package
  - Advantages
    - Modeled access
    - Fine-grained resolution
  - Functions for
    - Various types of depth information
    - Owner
    - Unit
    - Subprogram
    - Line number
    - Edition

PL/SQL New Features

Binding

Binding PL/SQL-Only Data Types

- Before 12c
  - Impossible to bind PL/SQL-only data types to SQL statements
    - Boolean
    - Package-level record
    - Package-level collection
- In 12c
  - It is possible

Binding PL/SQL-Only Data Types

- Restrictions
  - Associative array – index by string is not supported
  - Function cannot return PL/SQL-only data type to SQL
  - A Boolean literal cannot be an argument in a static SQL

PL/SQL New Features

Select from Collections

Select from Collection

- Before 12c
  - Impossible to select from package-level collections
- In 12c
  - It is possible

Easier Migration to Oracle

Implicit Result Sets

Implicit Result Sets

- Before 12c, result sets were retuned explicitly, using cursor variables
- In 12c it is possible also to return result sets implicitly:
  - dbms_sql.return_result
- SQL*Plus 12 automatically displays the implicit results

Easier Migration to Oracle

SQL Translation Framework

SQL Translation Framework

- SQL Translation Framework translates SQL statements of a client program from a non-Oracle dialect into the Oracle dialect
- Can also be used to substitute an Oracle SQL statement with another Oracle statement to address a semantic or a performance issue
- The name of the translator is specified at connect time
  - Database service attribute
  - ALTER SESSION SET sql_translation_profile and set event 10601

SQL New Features

Row Limiting

Row Limiting

- The Row Limiting clause is added to the end of SELECT statement
  - FETCH FIRST n ROWS WITH TIES
  - FETCH FIRST n ROWS ONLY
  - FETCH FIRST p PERCENT ROWS
  - OFFSET m ROWS FETCH NEXT n ROWS

SQL New Features

PL/SQL in the WITH Clause

PL/SQL in the WITH Clause

- Before 12c, the WITH clause included only subquery factoring
- In 12c, it can include also PL/SQL declarations
  - Functions, that can be used in the query
  - Procedures, that can be used in the functions
- Name resolution
  - statement-level function names have precedence over schema-level stored functions

PL/SQL in the WITH Clause

- Not yet supported in static SQL in PL/SQL
- If used as subquery, the top-level statement must be hinted with WITH_PLSQL (or be itself a SELECT with PL/SQL declaration)
- Considerations for statement-level vs. schema-level functions
  - Ad-hoc vs. common functionality
  - Performance
    - Consider also PRAGMA UDF

SQL New Features

Pattern Matching

Pattern Matching

- Basic SQL – row-level visibility
- Analytic Functions – window-level visibility, when the window is strictly defined
- Pattern Matching – enhanced analysis of row sequences

Pattern Matching

- MATCH_RECOGNIZE
  - Partition the data set into non-overlapping subsets
  - Sort each subset
  - Define variables, and conditions to map rows to variables
  - Define pattern to seek
  - Define measures
  - ONE ROW | ALL ROWS per match
  - AFTER MATCH SKIP…

Pattern Matching

- The MEASURES clause may include:
  - FIRST, LAST
  - MATCH_NUMBER()
  - CLASSIFIER()
  - COUNT, SUM, AVG, MIN, MAX
  - FINAL | RUNNING

Pattern Matching

- The PATTERN clause contains a regular expression and may use the operators:
  - * for 0 or more iterations
  - + for 1 or more iterations
  - ? for 0 or 1 iteration
  - {n} for n iteraions
  - ? as suffix for reluctant quantifier (rather than greedy)
  - And much more: http://docs.oracle.com/cd/E16655_01/server.121/e17749/pattern.htm#CACDIGBA

SQL New Features

Truncate Cascade

Truncate Cascade

- Before 12c
  - You cannot truncate the parent table of an enabled foreign key constraint
  - You cannot truncate the parent table of a reference-partitioned table
- In 12c
  - Yes, you can (assuming the foreign key is defined with ON DELETE CASCADE)

SQL New Features

New Join Syntax

New Join Syntax

- CROSS APPLY
- OUTER APPLY
- LATERAL inline view
- Multi-table “left outer join” with Oracle syntax (+)

SQL New Features

Online Operations

Online Operations

- In 12c more operations can be executed online:
  - DROP INDEX ONLINE
  - DROP CONSTRAINT ONLINE
  - SET UNUSED COLUMN ONLINE
  - ALTER INDEX UNUSABLE ONLINE
  - ALTER INDEX [VISIBLE | INVISIBLE]
  - ALTER TABLE MOVE SUB/PARTITION ONLINE

Utilities New Features

Data Pump Enhancements

Data Pump Enhancements

- impdp
  - TRANSFORM=DISABLE_ARCHIVE_LOGGING
  - TRANSFORM=TABLE_COMPRESSION_CLAUSE
- expdp
  - views_as_tables=viewname:tablename
- LOGTIME = all

Utilities New Features

SQL*Loader Express Mode

SQL*Loader Express Mode

- No need to create a control file
- A log file is generated for future use, including:
  - Control file
  - CREATE EXTERNAL TABLE statement
  - INSERT statement
- Data types of all table columns should be numeric, string or datetime

SQL*Loader Express Mode

- Defaults
  - File name: <tableName>.dat, in current directory
  - Record delimiter: newline
  - Field delimiter: comma
  - No enclosures

SQL*Loader Express Mode

- Mandatory command line parameter
  - TABLE
- Some optional command line parameters
  - DATA (up to 6 names, wildcards supported)
  - TERMINATED_BY
  - CHARACTERSET
  - CSV = WITH_EMBEDDED
  - OPTIONALLY_ENCLOSED_BY
  - DATE_FORMAT
  - DEGREE_OF_PARALLELISM
  - DIRECT

Transaction Guard

Transaction Guard

- A generic tool for applications to use for at-most-once execution in case of outages
- Applications use logical transaction ID to determine the outcome of the last transaction open in a database session following an outage

Transaction Guard

- Get the LTXID in the exception handler
  - JDBC: OracleConnection.getLogicalTransactionId
  - ODP.Net: OracleConnection.LogicalTransactionId
- Check the transaction outcome
  - JDBC: call DBMS_APP_CONT.GET_LTXID_OUTCOME
  - ODP.Net: OracleConnection.GetLogicalTransactionStatus

Configuration for Transaction Guard

- Use an application service (not the default services) for all database work
- Set parameters for the service:
  - COMMIT_OUTCOME = TRUE
  - RETENTION_TIMEOUT = <retention-value>
- GRANT EXECUTE ON DBMS_APP_CONT TO user;
- See http://docs.oracle.com/cd/E16655_01/appdev.121/e17620/adfns_trans_idemp_guard.htm#ADFNS332

Information Lifecycle Management

Temporal Validity

Temporal Validity

- Enables to track time periods for real world validity or effectivity
- Valid times can be set by users/application for data
- Data can be selected by a specific valid time (or range)

Temporal Validity

- A valid time period consists of two date/time columns (start/end)
- The columns can be created explicitly or implicitly
- The table is defined with PERIOD FOR

Temporal Validity

- Statement level visibility control
  - SELECT … AS OF PERIOD FOR
  - SELECT … VERSIONS PERIOD FOR
- Session level visibility control
  - DBMS_FLASHBACK_ARCHIVE.enable_at_valid_time
    - ALL
    - CURRENT
    - ASOF

Information Lifecycle Management

In-Database Archiving

In-Database Archiving

- Enables to archive rows within a table by marking them as “inactive”
- Inactive rows are still there, but are not visible to the application (or visible, when we want)

In-Database Archiving

- The table should be defined as ROW ARCHIVAL
- A hidden column is created: ORA_ARCHIVE_STATE
- The session level parameter ROW ARCHIVAL VISIBILITY controls if archived rows are visible (ALL) or not (ACTIVE)

Information Lifecycle Management

Heat Map

Heat Map

- Heat Map provides data access tracking, like:
  - Last read time
  - Last update time
  - Last full table scan
  - Last index scan
- Enable/disable:
  - alter system set HEAT_MAP = on | off (default is off)

Information Lifecycle Management

Automatic Data Optimization

Automatic Data Optimization

- Automatic data movement between storage tiers
- Automatic data compression
- Based on the Heat Map
- ADO policies can be defined in row / segment / tablespace levels

Partitioning New Features

Partitioning Schemes

Partitioning Schemes

- Range (8)
- Hash (8i)
- List (9i)
- Composite
  - Range-Hash (8i)
  - Range-List (9i)
  - The other combinations (11g)

Partitioning Schemes – Cont.

- Reference (11g)
- Interval (11g)
- Reference + Interval (12c)

Partitioning New Features

Asynchronous Global Index Maintenance

Async Global Index Maintenance

- DROP/TRUNCATE PARTITION
  - Before 12c: either fast operation or keeping global indexes usable
  - In 12c: both fast operation and keeping global indexes usable
- New data dictionary column ORPHANED_ENTRIES
- Orphaned index entries are:
  - Not deleted as part of the operation
  - Not reused (usually)

Async Global Index Maintenance

- Cleaning out the orphaned entries:
  - Alter index rebuild
  - Alter index coalesce cleanup
  - DBMS_PART.CLEANUP_GIDX
  - PMO_DEFERRED_GIDX_MAINT_JOB

Partitioning New Features

Truncate/Exchange Cascade

Truncate/Exchange Partition

- Before 12c
  - You cannot truncate/exchange partitions of the parent table of a reference-partitioned table
- In 12c
  - Yes, you can (assuming the foreign key is defined with ON DELETE CASCADE)

Partitioning New Features

Multiple (Sub)Partitions in Single DDL Operations

Multiple Partitions in Single DDL Operations

- Add (range, list)
- Drop (range, list)
- Merge (range, list)
- Split (range, list)
- Truncate (range, list, hash, ref)

Partitioning New Features

Partial Indexes

Partial Indexes

- In 12c it is possible to create an index on a subset of partitions
- Supported for both local and global indexes
- The optimizer is aware of it
- INDEXING ON | INDEXING OFF
  - At table level
  - At partition level
- INDEXING PARTIAL
  - At index level
- New data dictionary column INDEXING

שבוע אורקל 2013

להתראות

בשבוע אורקל הבא

16-20 בנובמבר 2014

מכללת אל-תק + JOHN BRYCE תלמד אי-טק. זה עובד!

ORACLE

dbORIENTED