Oracle 12c: Gee Whiz Features

Arup Nanda

Longtime Oracle DBA
Agenda

• Key useful features; not all.
• 12 Features
• Some demo
1. Pluggable Database

Application 1
DB1
User SIEBEL

Application 2
DB2
User SIEBEL

Application 3
DB3
User SIEBEL

Application 1
Application 2
Application 3
DBX
User SIEBEL
DBA_USERS

```
SELECT NAME
FROM USER$
WHERE CON_ID = ...
```

- PDB1: CON_ID=2
- PDB2: CON_ID=3
- PDB3: CON_ID=4
2. RMAN

• Sysbackup role
  – grant sysbackup to rmanuser identified by rmanuser;
  – rman target='rmanuser/rmanuser as sysbackup'

• SQL in RMAN
  – No need to put #SQL ‘command’
  – Select, even describe

• Recovering over the network
  – Great for data guard environments
**Table Recovery**

```
RMAN> recover table scott.accounts:p1
   2> until scn 123456
   3> auxiliary destination '+DG1';
```

- Creates a table called ACCOUNTS_P1
- Remap Table Name
  ```
  4> remap table arup.accounts:p1:newaccs;
  ```
- Remap Tablespace
  ```
  4> remap tablespace users:accdata;
  ```
- Datapump Dump
  ```
  4> datapump destination '/tmp' dump file 'acc.dmp'
  5> notableimport;
  ```
Cross-platform Database Migration

Full backup

Incremental backup

Final Inc

Oracle 12c: Gee Whiz Features
3. Data Guard

- Global Temporary Tables in ADG
  - You can use DML on GTTs
  - Uses temporary undo (in temp tablespace)

- Sequences in ADG
  - Select x.nextval from dual
  - Pulls the <cache> amount of numbers for SGA

- Realtime cascade

- Data Guard Broker
  - One command role transition
  - Resumable switchover
4. MV Refresh

```sql
dbms_mview.refresh(
    ...
    out_of_place => true
)
```
5. Partitioning: Asynch Global Index

```
alter table
drop t
partition part3
update global indexes;
A scheduler job
pmo_deferred_gidx _maint_job cleans up
```
5. Partitioning: Partial Index

```sql
SQL> alter table ptabl modify partition p1 indexing on;

SQL> alter table ptabl modify partition p2 indexing off;

SQL> create index in_g2_ptabl on ptabl (c1) global indexing partial;
```
6. Automatic Data Optimization

- **Enable Heat Map**
  ```sql
  alter system set heat_map = on scope=both;
  ```

- **Add Data Movement Policy**
  ```sql
  alter table exchange_rate ilm add policy tier to midterm_ts;
  ```

- **Add Compression Policy**
  ```sql
  alter table exchange_rate ilm add policy row store
  compress advanced segment
  after 7 days
  of no modification;
  ```
7. Schema Management

- **DDL Logging**
  - Enable `alter system set enable_ddl_logging=true;`
  - The logs are written in
    ```
    C:\oracle\diag\rdbms\anl2\anl2\log\ddl
    ```
  - In XML format

```xml
<msg time='2013-08-30T20:29:36.635-04:00' org_id='oracle' comp_id='rdbms'
  msg_id='opiexe:4181:29461b3730' type='UNKNOWN' group='diag_adl'
  level='16' host_id='STARUPNANT420B'
  host_addr='fe80::58b8:d0b2:f7c9:3147%27'
  version='1'>
  <txt> create table t11 (col1 number) </txt>
</msg>

<msg time='2013-08-30T20:32:56.719-04:00' org_id='oracle' comp_id='rdbms'
  msg_id='opiexe:4181:29461b3730' type='UNKNOWN' group='diag_adl'
  level='16' host_id='STARUPNANT420B'
  host_addr='fe80::58b8:d0b2:f7c9:3147%27'>
  <txt> drop table t11 </txt>
</msg>
```
Multiple Indexes

```sql
SQL> create table t3 (col1 number, col2 number);
Table created.

SQL> create index in_t3 on t3(col1);
Index created.

SQL> create index in_t3_02 on t3(col1);
create index in_t3_02 on t3(col1)
  *
ERROR at line 1:
ORA-01408: such column list already indexed

SQL> create bitmap index in_t3_02 on t3(col1) invisible;
Index created.
```

Rules
- Different types: b-tree/ bitmap
- Unique/nonUnique
- Only one is visible at a time
Invisible Column

```sql
SQL> create table t4 (col1 number, col2 number invisible);

SQL> desc t4
Name Null? Type
----- ----- -----  
COL1 NUMBER

SQL> insert into t4 values (1);
1 row created.

SQL> select * from t4;
    COL1
----------
      1

SQL> select col1, col2 from t4;
  COL1   COL2
-------- -------
     1

SQL> insert into t4 (col1, col2) values (2,2);
1 row created.

SQL> set colinvisible on

SQL> desc t4
Name Null? Type
----- -------- -----
COL1     NUMBER  
COL2 (INVISIBLE) NUMBER

SQL> create index in_t4 on t4(col2);
Index created.
```
Default Values

SQL> create table t5 (col1 number, col2 number default on null 0);
Table created.

SQL> desc t5
   Name                  Null? Type
   ----------- --------
   COL1          NUMBER
   COL2              NOT NULL NUMBER

SQL> insert into t5 values (1, null);
SQL> insert into t5 values (2,2);

SQL> select * from t5;
   COL1  COL2
--------------------
    1      0
    2      2

SQL> create table t6 (col1 number generated always as identity);

SQL> create table t7 (col1 number generated always as identity (start with 1000 increment by 10));

SQL> insert into t6 values (1);
* ERROR at line 1:
ORA-32795: cannot insert into a generated always identity column

SQL> create table t9 (col1 number, col2 number generated by default as identity);
SQL> insert into t9 values (9,9);
SQL> insert into t9 values (10,default);
SQL> insert into t9 (col1) values (11);
SQL> select * from t9;
   COL1  COL2
--------------------
     9      9
    10      2
    11      3
8. SQL and PL/SQL

- Repetitive operations
- Stored Function and Procedure

```sql
CALC_INT (  
    P_principal  
    P_int_rate  
  )

Returns Number
```
with function calc_int (  
    p_principal in number,  
    p_int_rate  in number  
)  
return number  
as  
begin  
   return p_principal * (1+p_int_rate/100);  
end;  
select  
  balance,  
  calc_int(balance,10) int  
from ptab1;
Top-N Query

• First 10, second 10 rows, etc.

```
select ... from (select ... from ... order by ...) where rownum <= 10
```

• 12c way:

```
select *
from sales_fact
order by year, week, country, region, product
fetch first 10 rows only;
```

• Next 10 rows

  - offset 10 rows fetch first 10 rows only
  - offset 10 rows fetch first 0.1 percent rows only
  - offset 10 rows fetch first 0.1 percent rows with ties
• Session Sequences
  – Values visible only in the session
  – Not persistent

```
SQL> create sequence sessseq session;
SQL> create sequence globseq global;
SQL> select globseq.nextval from dual;
3
SQL> select sessseq.nextval from dual;
1
```
create view v1 as select * from t1;

select * from v1;

SQL> var o clob
SQL> begin
    2       dbms_utility.expand_sql_text ( 'select * from v1', :o);
    3  end;
    4  /

SQL> print o

SELECT "A1"."COL2" "COL2" FROM (SELECT "A2"."COL2" "COL2" FROM ARUP."T1" "A2")
9. Administration

- PGA Size Limit
  - `pga_aggregate_target` is merely a target
  - `pga_aggregate_limit` limits PGA memory consumption
  - Greater of
    - 2 GB
    - 2 X `pga_aggregate_target`
    - 3 MB X processes

- Error:
  - `ORA-00028: your session has been killed`
  - `ORA-04036: PGA memory used by the instance exceeds PGA_AGGREGATE_LIMIT`
  - PGA memory used by the instance exceeds `PGA_AGGREGATE_LIMIT` of 4000 MB

- Want Old Behavior?
  - Set `pga_aggregate_limit` to 0
• Adaptive Execution Plans
  – Estimate can change mid-execution
• Incremental Stats
  – Stats of partitions
10. Online Activities

• Online Datafile Move
  – No need to put tablespace to read only
    ```sql
    ALTER DATABASE MOVE DATAFILE 'old1.dbf' TO 'new1.dbf';
    ```
  – Queries and DML can continue
  – Great for ASM conversions
  – Use REUSE at the end, if you want to overwrite
  – If you want to keep the old one, use KEEP at the end
• Move partitions online
  – alter table t1 move p1 to
tablespace ts1 update indexes online
  – DML allowed as usual

• Online DDLs

drop index i1 online;
alter table t1 drop constraint c1 online;
alter table t1 set unused column col1 online;
alter index i1 unusable online;
alter index i1 [in]visible;
11. Replay Consolidation

Capture workload → files → Replay workload

DB1

Capture workload → files → Replay workload

DB2
11. Replay Consolidation

Capture workload → files → Replay workload

DB1 → DB1

Capture workload → files → Replay workload

DB2 → DB2
Workload Characteristics #1

Time

Load

Capacity

DB1

DB2

Oracle 12c: Gee Whiz Features
Workload Characteristics #2

![Graph showing load and capacity over time for DB1 and DB2.]

- Time
- Capacity
- Load
- DB1
- DB2
Database Replay Consolidation

- Capture from multiple databases
- Replay all the capture files against one database

![Graph showing capacity and load over time for DB1 and DB2 with a combined result line.](image)
12. Miscellaneous

- **Temporary Undo**
  - Put the undo information in TEMP tablespace
  - `alter session set temp_undo_enabled=true;`
  - `V$TEMPUNDOSTAT` view gives the details

- **VARCHAR2** is now 32676 bytes
  - Param `MAX_STRING_SIZE` should be set to EXTENDED
  - DB must be in upgrade mode
  - Irreversible

- **Inline Stats Collection**
  - CTAS, Insert Into Select From will collect stats
Thank You!

My Blog: arup.blogspot.com
My Tweeter: arupnanda