Oracle 12c Top 20 New Features for Developers

Arup Nanda

Longtime Oracle DBA

Agenda

- Top features of Oracle 12c for Developers
- Excludes PL/SQL
  - Covered in a different session
- Of interest to developers and users
  - Not to DBAs

Online DDL

- DDLs do not need lock.
- DML continues as usual
  - drop index i1 online
  - alter index i1 unusable online
  - alter table t1 set unused columns online
  - alter table t1 drop column c1 online
  - alter table t1 move partition p1 online
    - subpartition too

Cascading Truncate

- When you truncate a parent table with child tables, you get:

  ```
  ORA-02266: unique/primary keys in table
  referenced by enabled foreign keys
  ```

- In Oracle 12c, you can use:

  ```
  truncate table <Parent> cascade;
  ```

- Must have defined the FK as ON DELETE CASCADE.

- Otherwise `ORA-14705: unique or primary keys referenced by enabled foreign keys in table will result`

Trunc1.sql

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

topn_first.sql
topn_offset.sql
topn_percent.sql

Arup Nanda

TopN Query Plan

- TopN Plan

```
Rows      Execution Plan

-------   --------------------------------------------------------
      0   SELECT STATEMENT    MODE: ALL_ROWS
      5    SORT (ORDER BY)
      5     VIEW
      5      WINDOW (SORT PUSHED RANK)
 100000       TABLE ACCESS    MODE: ANALYZED (FULL) OF 'ACCOUNTS' (TABLE)
```

- Regular Plan

```
Rows      Execution Plan

-------   --------------------------------------------------------
      0   SELECT STATEMENT    MODE: ALL_ROWS
      5    COUNT (STOPKEY)
      5     VIEW
      5      SORT (ORDER BY STOPKEY)
 100000       TABLE ACCESS    MODE: ANALYZED (FULL) OF 'ACCOUNTS' (TABLE)
```

TopN Restrictions

- If you have a SELECT statement with FOR UPDATE, you can't use it.

- The SELECT statement can't CURRVAL or NEXTVAL of sequences

- If the query of the Materialized Views has this clause, then you can't do an incremental refresh of that MV

Arup Nanda

Bottom-N

```
select round(principal+interest,2) tot_bal,
round((sysdate-created_dt)) age, accno
from accounts
order by 1 desc
offset ((select count(1) from accounts) - 5) rows
fetch next 5 rows only
```

bottomn.sql

Arup Nanda

Session Sequences

- – `select seq1.nextval from dual;`
- – Session Seq: 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
```

Seqg.sql sqqs.sql

Arup Nanda

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

```
<msg time='2013-08-30T20:29:36.635-04:00' org_id='oracle' comp_id='rdbms'
 msg_id='opiexe:4181:2946163730' 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:4181:2946163730' type='UNKNOWN' group='diag_adl'
 level='16' host_id='STARUPNANT420B' host_addr='fe80::58b8:d0b2:f7c9:3147%27'>
 <txt>drop table t11
 </txt>
</msg>
```

View Expansion

```
create view v1 as select * from t1;

select * from v1;

SQL> var o clob
SQL> begin
  2       dbms_utility.expand_sql_text (
  3               'select * from v1',:o);
  4  end;
  5  /

SQL> print o

SELECT "A1"."COL2" "COL2" FROM  (SELECT "A2"."COL2"
"COL2" FROM ARUP."T1" "A2")
```

Exp1.sql

Multiple Indexes

```
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

Multind1.sql

Arup Nanda

Invisible Column

```
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.
```

Invcol1.sql

Invisible Columns, contd.

```
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.
```

Arup Nanda

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
```

Defval1.sql

Identity Column

```
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);
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
```

Idcol1.sql

Longer Varchar2

- VARCHAR2 is now 32676 bytes
  - Param MAX_STRING_SIZE should be set to EXTENDED
  - DB must be in upgrade mode
  - Irreversible
  - CLOB behind the scenes

Outer Join

```
col prod_name format a30
col promo_name format a30
set lines 132 pages 45
set pau on
select prod_name,  promo_name, channel_desc, count(amount_sold) cnt
from sales s, channels h, promotions m, products p
where h.channel_id = s.channel_id (+)
and m.promo_id = s.promo_id (+)
and p.prod_id = s.prod_id (+)
group by prod_name, promo_name, channel_desc
order by prod_name, promo_name, channel_desc
```

Outer Join, contd.

```
PROD_NAME                      PROMO_NAME                      CHANNEL_DESC                CNT
------------------------------ ------------------------------- -------------------- ----------
1.44MB External 3.5" Diskette  NO PROMOTION #                  Catalog                       0
1.44MB External 3.5" Diskette  NO PROMOTION #                  Direct Sales              13655
1.44MB External 3.5" Diskette  NO PROMOTION #                  Internet                   2457
1.44MB External 3.5" Diskette  NO PROMOTION #                  Partners                   6265
1.44MB External 3.5" Diskette  NO PROMOTION #                  Tele Sales                    0
… truncated
```

Cross Apply

```
SELECT …
FROM    T1  ,  T2Collection
 WHERE …
```

Cross Apply, contd 1

```
create or replace type final_acc_int as table of number;

create or replace function get_final_int_tab (
    p_acctype        in accounts.acctype%type )
return final_acc_int is
    l_ret    final_acc_int;
begin
    select
            cast(
                    collect(interest)
                    as final_acc_int
            )
    into l_ret
    from accounts
    where acctype = p_acctype;
    return l_ret;
end;
/
```

Cross Apply, contd. 2

```
select * from
    account_types a
        cross apply
    get_final_int_tab(a.acctype)
where acctype in 'S'
order by column_value
/
```

```
A ACC_DESC    COLUMN_VALUE
- ---------- ------------
… output truncated …
S Savings       9921.56003
S Savings       9943.34883
S Savings       9957.86381
```

Cross.sql

Outer Apply

```
select *
    from account_types a
outer apply
get_final_int_tab(a.acctype)
where acctype in 'T'
order by column_value
/
```

Outer.sql

Lateral Views

- Objective:

```
select acc_desc, interest
from accounts a,
   (select * from account_types t where
t.acctype = a.acctype);
```

- Will fail with
- ORA-00904: "A"."ACCTYPE": invalid identifier

```
select acc_desc, interest
from accounts a,
    lateral (select * from account_types t
where a.acctype = t.acctype)
/
```

Lateral.sql

Arup Nanda

Nologging Import

```
impdp arup/arup tables=ACCOUNTS
transform=DISABLE_ARCHIVE_LOGGING:Y
table_exists_action=append
```

. . imported "ARUP"."ACCOUNTS"
10.67 MB   100000 rows

Job "ARUP"."SYS_IMPORT_TABLE_01" successfully
completed at Mon Oct 21 23:57:09 2013 elapsed 0
00:00:06

Arup Nanda

Match Recognize

```
select *
from sales
match_recognize(
  partition by prod_id
  order by time_id
  measures strt.time_id as start_date,
  last(down.time_id) as bottom_date,
  last(up.time_id) as   end_date,
  sum(amount_sold) as tot_sold
   one row per match
   after match skip to last up
   pattern (strt down+ up+)
   define
     down as down.amount_sold < prev(down.amount_sold),
     up as up.amount_sold > prev(up.amount_sold)
   ) matcher
 where prod_id = 13
 order by matcher.start_date
```

Match Recognize, contd.

```
PROD_ID START_DAT BOTTOM_DA END_DATE   TOT_SOLD
--------- --------- --------- --------- ----------
       13 02-DEC-01 02-DEC-01 02-DEC-01     3752.1
       13 03-DEC-01 03-DEC-01 10-DEC-01    2914.48
       13 10-DEC-01 10-DEC-01 10-DEC-01    2719.28
       13 17-DEC-01 17-DEC-01 17-DEC-01    3000.86
       13 17-DEC-01 17-DEC-01 17-DEC-01     3752.1
       13 20-DEC-01 20-DEC-01 20-DEC-01    2719.28
       13 20-DEC-01 20-DEC-01 20-DEC-01    2719.28
       13 23-DEC-01 23-DEC-01 23-DEC-01    2816.88
       13 23-DEC-01 24-DEC-01 24-DEC-01     3752
       13 23-DEC-01 23-DEC-01 23-DEC-01    2719.
       13 23-DEC-01 23-DEC-01 23-DEC-01    2719.
```

MATCH_RECOGNIZE includes:
– PARTITION Segregate data
– ORDER BY Order with partitions
– MEASURES Define output columns
– AFTER Return single/multiple rows
– PATTERN Define regular expression
– DEFINE Specify expression tags

Arup Nanda

Thank You!

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