

Write Less Code With More Oracle 12c New Features

Oren Nakdimon

www.db-oriented.com

✉ oren@db-oriented.com

☎ +972-54-4393763

🐦 @DBoriented

WHO AM I?

A CHRONOLOGY BY "ORACLE YEARS"

Where: IAF
When: Oracle 6/7 [1991-1997]
What: Developer


Where: Golden Screens
When: Oracle 8 [1997-1998]
What: Server Group Manager

Where: TELEknowledge
When: Oracle 8i/9i [1998-2003]
What: DBA Group Manager

Where: Olista
When: Oracle 10g/11g [2004-2011]
What: VP R&D + Israel Site Manager

Where: **dbORIENTED**
When: Oracle 11g/12c [2011-]
What: Freelance Consultant

Lectures and Courses

Where:  **moovit**
When: 2015-
What: Database Expert



@DBORIENTED

Oren Nakdimon @DBoriented

#oracle #tip

USER_TAB_COLS columns.

USER_TAB_COLS

- in 11g: hidden
- in 12c: system

RETWEETS	LIKES
2	5

11:23 AM - 28 Sep 2015

Oren Nakdimon @DBoriented

default prefix length in COMPRESS with no value:

- non-unique index = # of columns
- unique index = # of columns - 1

Oren Nakdimon @DBoriented

cardinality of *column* + 0 = constant

≠

cardinality of *column* - 0 = constant

[db-oriented.com/2015/09/18/whe ...](http://db-oriented.com/2015/09/18/whe...)

#oracle CBO #SpecialCase

Oren Nakdimon @DBoriented

ALTER INDEX [IN]VISIBLE is a

operation in #oracle 12c (was of

No need, and impossible, to spec

ONLINE keyword.

Can you tell what is loaded by

#oracle Data Pump Import to constraint


violation?

Some #trickery helps ;)

[db-oriented.com/2014/07/19/imp ...](http://db-oriented.com/2014/07/19/imp...)

HTTP://DB-ORIENTED.COM

עברית



Blog
About
Courses
Presentations
Contact

Write Less with More – Part 8 (PL/SQL in the WITH Clause)


🕒 November 14, 2015
👤 Oren Nakdimon
💬 2 Replies

This post is part 8 of the **Write Less with More** series of blog posts, focusing on new features of Oracle 12c that allow us developers to **write less** than in previous versions in order to achieve the same functionality. Each part is dedicated for one new feature that solves some common development task. For more details, including the setup of the examples used throughout the series and the list of development tasks that drive the series, please see [part 1](#).

All the Series Parts

- Part 1** – write less configuration with **SQL*Loader Express Mode**
- Part 2** – write less application code with **Identity Columns**
- Part 3** – write less application code with **In-Database Archiving**
- Part 4** – write less application code with **Temporal Validity**
- Part 5** – write less code in SQL with **Row Limiting**
- Part 6** – write less “inappropriately located” code with **SELECT FROM Package-Level Collection Types**
- Part 7** – write less code in SQL with **Lateral Inline Views**
- Part 8** – write less “inappropriately located” code with **PL/SQL in the WITH Clause**

Task #8



ORACLE[®]

ACE Associate

UPCOMING EVENTS

March 3, 2016: I will give two presentations in **OUG Ireland 2016:**

How to Upgrade Your Application with no Downtime (and no Additional Costs!)

Write Less (code) with More (Oracle 12c New Features)

RECENT POSTS

✍ Write Less with More – Part 8 (PL/SQL in the WITH Clause)
November 14, 2015

MORE “WRITE LESS WITH MORE”

Download this presentation from
db-oriented.com/presentations

Read the blog post series
db-oriented.com/category/writelesswithmore/

Read a summary of this presentation in the article in Oracle Scene Issue 58



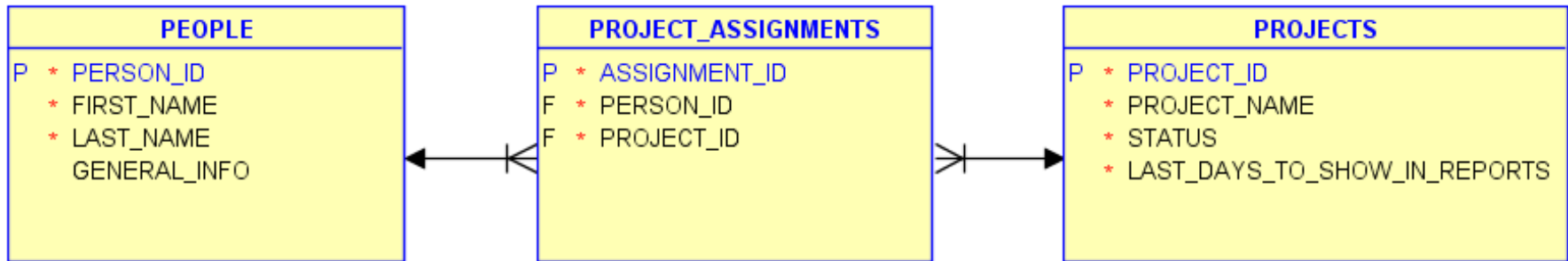
AGENDA

- Go over some common use cases
- For each one
 - A pre-12c solution
 - A 12c solution, that allows us to write less
- We'll see examples for features that allow writing less...
 - ...configuration
 - ...application code
 - ...code in SQL statements
 - ...“inappropriately-located” code

THE USUAL WARNING

- Before you decide to apply any of the presented features in production, make sure:
 - to thoroughly test them for your applications and environments
 - that you have the required license to use them

MEET THE DEMO TABLES





mobile +972 54 439 3763 . tel/fax +972 4 988 7216
P.O. Box 145 Zurit 2010400 . email oren@db-oriented.com
www.db-oriented.com

My To Do List

1. Fill the PEOPLE table from a text file

2. Set a unique number to ASSIGNMENT_ID (implicitly) for every new record

3. Delete obsolete projects (but keep their assignment history)

4. Add a "validity period" to PROJECT_ASSIGNMENTS

5. Write a query to get project assignments with pagination

Oren Nakdimon Oracle Database Expert
Database Administration | Development | Courses

1/2

2/2

LOADING DATA FROM FILES

1. Fill the PEOPLE table from a text file

○ Before 12c

- SQL*Loader
 - Requires a control file
- External tables
 - Require a “control file” within the CREATE TABLE statement
 - Then:

```
INSERT INTO <"real" table> ...  
SELECT FROM <external table>... ;
```

12c

SQL*LOADER EXPRESS MODE

- No need to create a control file
- Defaults
 - File name: <tableName>.dat, in current directory
 - Record delimiter: newline
 - Field delimiter: comma
 - No enclosures



@ldr

12c

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

12c

SQL*LOADER EXPRESS MODE

- 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



mobile +972 54 439 3763 . tel/fax +972 4 988 7216
P.O. Box 145 Zurit 2010400 . email oren@db-oriented.com
www.db-oriented.com

My To Do List

1. Fill the PEOPLE table from a text file

2. Set a unique number to ASSIGNMENT_ID (implicitly) for every new record

3. Delete obsolete projects (but keep their assignment history)

4. Add a "validity period" to PROJECT_ASSIGNMENTS

5. Write a query to get project assignments with pagination

Oren Nakdimon Oracle Database Expert
Database Administration | Development | Courses

1/2

2/2

AUTO-INCREMENT COLUMNS

2. Set a unique number to ASSIGNMENT_ID (implicitly) for every new record

○ Before 12c

- A sequence + a BEFORE INSERT trigger

```
CREATE SEQUENCE project_assignments_seq;
```

```
--  
-- first option: ignore the input even if supplied ←  
--  
CREATE TRIGGER project_assignments_bir_tr  
  BEFORE INSERT ON project_assignments  
  FOR EACH ROW  
  
BEGIN  
  :new.assignment_id := project_assignments_seq.nextval;  
END;
```

AUTO-INCREMENT COLUMNS

2. Set a unique number to ASSIGNMENT_ID (implicitly) for every new record

○ Before 12c

- A sequence + a BEFORE INSERT trigger

```
CREATE SEQUENCE project_assignments_seq;
```

```
--  
-- second option: only if input not supplied  
--  
CREATE TRIGGER project_assignments_bir_tr  
  BEFORE INSERT ON project_assignments  
  FOR EACH ROW  
  WHEN (new.assignment_id IS NULL)  
BEGIN  
  :new.assignment_id := project_assignments_seq.nextval;  
END;
```


12c

IDENTITY COLUMNS

- 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

@idn1

12c

IDENTITY COLUMNS

- 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

@idn2



mobile +972 54 439 3763 . tel/fax +972 4 988 7216
P.O. Box 145 Zurit 2010400 . email oren@db-oriented.com
www.db-oriented.com

My To Do List

1. Fill the PEOPLE table from a text file

2. Set a unique number to ASSIGNMENT_ID (implicitly) for every new record

3. Delete obsolete projects (but keep their assignment history)

4. Add a "validity period" to PROJECT_ASSIGNMENTS

5. Write a query to get project assignments with pagination

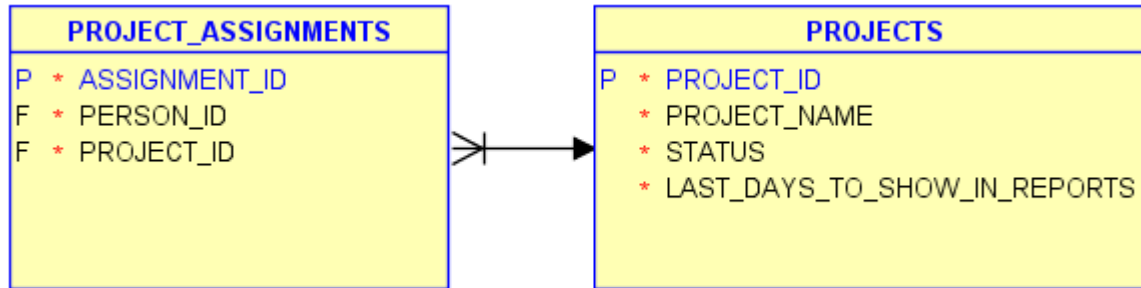
Oren Nakdimon Oracle Database Expert
Database Administration | Development | Courses

1/2

2/2

LOGICAL DELETION OF RECORDS

3. Delete obsolete projects (but keep their assignment history)



LOGICAL DELETION OF RECORDS

3. Delete obsolete projects (but keep their assignment history)

○ Before 12c

- Add an IS_DELETED column
- Use a view to hide “deleted” records

```
ALTER TABLE projects ADD  
  is_deleted NUMBER(1) DEFAULT 0 NOT NULL  
  CHECK (is_deleted IN (0,1));
```

```
RENAME projects TO all_projects;
```

```
CREATE VIEW projects AS  
  SELECT *  
  FROM all_projects  
  WHERE is_deleted=0;
```

12c

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)

@idar

12c

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)



mobile +972 54 439 3763 . tel/fax +972 4 988 7216
P.O. Box 145 Zurit 2010400 . email oren@db-oriented.com
www.db-oriented.com

My To Do List

1. Fill the PEOPLE table from a text file
2. Set a unique number to ASSIGNMENT_ID (implicitly) for every new record
3. Delete obsolete projects (but keep their assignment history)
4. Add a "validity period" to PROJECT_ASSIGNMENTS
5. Write a query to get project assignments with pagination

Oren Nakdimon Oracle Database Expert
Database Administration | Development | Courses

1/2

2/2

VALIDITY PERIODS

4. Add a "validity period" to
PROJECT_ASSIGNMENTS

○ Before 12c

- Add PERIOD_START and PERIOD_END columns
- Add conditions to queries to filter only valid periods

12c

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)

12c

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



@vld

12c

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

THERE ARE TWO SIDES TO EVERY COIN

In Database
Archiving

Temporal
Validity

- Hidden columns
- Hidden predicates

```
SELECT * FROM PROJECTS ;  
where most of the rows are "archived"
```

- "Hidden" = "Easy to Forget"
- "Hidden" \neq "Can be Ignored"

```
DELETE  
    PROJECT_ASSIGNMENTS ;  
where  
    enable_at_valid_time  
    is on
```

- And session-level control is session-level control



mobile +972 54 439 3763 . tel/fax +972 4 988 7216
P.O. Box 145 Zurit 2010400 . email oren@db-oriented.com
www.db-oriented.com

My To Do List

1. Fill the PEOPLE table from a text file
2. Set a unique number to ASSIGNMENT_ID (implicitly) for every new record
3. Delete obsolete projects (but keep their assignment history)
4. Add a "validity period" to PROJECT_ASSIGNMENTS
5. Write a query to get project assignments with pagination

Oren Nakdimon Oracle Database Expert
Database Administration | Development | Courses

1/2

2/2

TOP-N AND PAGINATION QUERIES

5. Write a query to get project assignments with pagination

○ Before 12c

- Use inline views and ROWNUM or ROW_NUMBER()
- Queries become quite cumbersome

TOP-N AND PAGINATION QUERIES

5. Write a query to get project assignments with pagination

- All the records
- Top 8 records
- Records 5-8

```
SELECT project_id,  
       person_id,  
       assignment_id,  
       validity_period_start,  
       validity_period_end  
FROM (  
  SELECT x.*  
        ,rownum row_num  
  FROM (  
    SELECT project_id,  
           person_id,  
           assignment_id,  
           validity_period_start,  
           validity_period_end  
    FROM project_assignments  
    ORDER BY project_id, person_id  
  ) x WHERE rownum <= 8  
 ) WHERE row_num > 4;
```


12c

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

@topn



mobile +972 54 439 3763 . tel/fax +972 4 988 7216
P.O. Box 145 Zurit 2010400 . email oren@db-oriented.com
www.db-oriented.com

My To Do List

6. Write a procedure to update the status of multiple projects

7. Write a query that shows the number of assignments per project in the last days

8. Write a query that shows all the people with a valid date in their GENERAL_INFO column

6. Write a procedure to update the status of multiple projects

- Write a stored procedure that
 - Gets a collection parameter of “project updates”, each one with:
 - PROJECT_ID
 - UPDATE_TIME
 - STATUS
 - Updates the PROJECTS table with the latest status of each project

PROJECT_ID	STATUS
1	2 3
2	3 2
3	4 4
4	2

PROJECT ID	UPDATE TIME	STATUS
1	10:00	2
1	14:00	3
2	09:00	1
3	09:00	4
2	08:00	3
3	08:00	2
2	10:00	2

6. Write a procedure to update the status of multiple projects

```
CREATE OR REPLACE PACKAGE projects_dl AS
```

```
    TYPE proj_update_t IS RECORD (  
        project_id projects.project_id%TYPE,  
        update_time DATE,  
        status      projects.status%TYPE);
```

```
    TYPE proj_update_tt IS TABLE OF proj_update_t;
```

```
    PROCEDURE update_status (  
        i_proj_update_list IN proj_update_tt  
    );
```

```
END projects_dl;
```

6. Write a procedure to update the status of multiple projects

```
CREATE OR REPLACE PACKAGE BODY projects_d1 AS
  PROCEDURE update_status(
    i_proj_update_list IN proj_update_tt) IS
  BEGIN
    MERGE INTO projects p
    USING (
      SELECT project_id,
             MAX(status) keep(dense_rank LAST
                              ORDER BY update_time) latest_status
      FROM   TABLE(i_proj_update_list)
      GROUP BY project_id
    ) i
    ON (p.project_id = i.project_id)
    WHEN MATCHED THEN UPDATE
      SET p.status = i.latest_status;
  END update_status;
END projects_d1;
```

SELECT FROM COLLECTION VARIABLES

○ Before 12c

- Impossible to SELECT FROM package-level collections

```
PL/SQL: SQL Statement ignored
```

```
ORA-22905:
```

```
cannot access rows from a non-nested table item
```

```
PLS-00642:
```

```
local collection types not allowed in SQL statements
```

SELECT FROM COLLECTION VARIABLES

○ Before 12c

- Schema-level collection types must be created and used, even if needed only in the scope of a single package

```
CREATE OR REPLACE TYPE proj_update_t AS OBJECT (  
    project_id    INTEGER,  
    update_time   DATE,  
    status        INTEGER)
```

```
CREATE TYPE proj_update_tt AS TABLE OF proj_update_t
```

```
CREATE OR REPLACE PACKAGE projects_d1 AS  
    PROCEDURE update_status(  
        i_proj_update_list IN proj_update_tt  
    );  
END projects_d1;
```

12c

SELECT FROM COLLECTION VARIABLES

- Now it is possible to SELECT FROM package-level collections



mobile +972 54 439 3763 . tel/fax +972 4 988 7216
P.O. Box 145 Zurit 2010400 . email oren@db-oriented.com
www.db-oriented.com

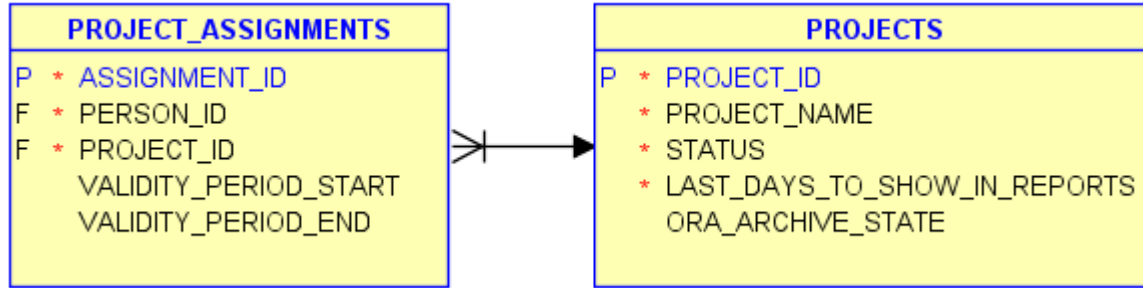
My To Do List

6. Write a procedure to update the status of multiple projects

7. Write a query that shows the number of assignments per project in the last days

8. Write a query that shows all the people with a valid date in their GENERAL_INFO column

7. Write a query that shows the number of assignments per project in the last days



PROJECT NAME	DATE	NUM OF ASSIGNMENTS
Project A	02/03/2016	3
Project A	03/03/2016	3
Project B	01/03/2016	5
Project B	02/03/2016	4
Project B	03/03/2016	2
Project C	29/02/2016	1
Project C	01/03/2016	0
Project C	02/03/2016	0
Project C	03/03/2016	1

PROJECT NAME	LAST DAYS TO SHOW IN REPORTS
Project A	2
Project B	3
Project C	4

@ltrl

12c

LATERAL INLINE VIEWS

- An inline view in a query can be defined as LATERAL
- It allows it to refer to other tables that appear to its left in the FROM clause



mobile +972 54 439 3763 . tel/fax +972 4 988 7216
P.O. Box 145 Zurit 2010400 . email oren@db-oriented.com
www.db-oriented.com

My To Do List

6. Write a procedure to update the status of multiple projects

7. Write a query that shows the number of assignments per project in the last days

8. Write a query that shows all the people with a valid date in their GENERAL_INFO column

CALLING PL/SQL FROM SQL

8. Write a query that shows all the people with a valid date in their GENERAL_INFO column

○ Before 12c

- A PL/SQL function must be stored in the database
- And it must be public (either standalone or exposed in a package spec)

CALLING PL/SQL FROM SQL

8. Write a query that shows all the people with a valid date in their GENERAL_INFO column

○ Before 12c

```
CREATE OR REPLACE FUNCTION is_date(i_info IN VARCHAR2)
RETURN NUMBER AS
    l_date DATE;
BEGIN
    IF i_info IS NULL THEN
        RETURN 0;
    ELSE
        l_date := to_date(i_info, 'dd/mm/yyyy');
        RETURN 1;
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        RETURN 0;
END;
```

```
SELECT * FROM people WHERE is_date(general_info) = 1;
```

12c

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



@with

12c

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



mobile +972 54 439 3763 . tel/fax +972 4 988 7216
P.O. Box 145 Zurit 2010400 . email oren@db-oriented.com
www.db-oriented.com

My To Do List

1. Fill the PEOPLE table from a text file
2. Set a unique number to ASSIGNMENT_ID (implicitly) for every new record
3. Delete obsolete projects (but keep their assignment history)
4. Add a "validity period" to PROJECT_ASSIGNMENTS
5. Write a query to get project assignments with pagination

Oren Nakdimon Oracle Database Expert
Database Administration | Development | Courses



mobile +972 54 439 3763 . tel/fax +972 4 988 7216
P.O. Box 145 Zurit 2010400 . email oren@db-oriented.com
www.db-oriented.com

My To Do List

6. Write a procedure to update the status of multiple projects
7. Write a query that shows the number of assignments per project in the last days
8. Write a query that shows all the people with a valid date in their GENERAL_INFO column

Oren Nakdimon Oracle Database Expert
Database Administration | Development | Courses

THANK YOU 😊

Oren Nakdimon

www.db-oriented.com

✉ oren@db-oriented.com

☎ +972-54-4393763

🐦 @DBoriented