

- Functional and Grammar differences between PostgreSQL and Oracle:

\* Transactions

Oracle database always uses transactions but in PostgreSQL you have to activate that. In Oracle, the transaction starts when executing any statement and ends when COMMIT statement executed. In PostgreSQL, transaction starts when execute BEGIN and end when COMMIT statement executed.

\* Dual Table

In Oracle FROM clause is mandatory for every SELECT statement so Oracle database uses DUAL table for SELECT statement where table name is not required. In PostgreSQL, FROM clause is not mandatory so DUAL table is not necessary.

\* SYSDATE

Oracle's SYSDATE function returns date and time. The behavior of SYSDATE function is different in different places. PostgreSQL does not have any function corresponding to SYSDATE function.

\* TO\_DATE(two argument)

Oracle's TO\_DATE function return DATE type value(year, month, day, hour, minute, second) while PostgreSQL's TO\_DATE(two\_argument) return DATE type value(year,month,day). The solution for this incompatibility is to convert TO\_DATE() to TO\_TIMESTAMP().

\* SYNONYM

CREATE SYNONYM is not supported in PostgreSQL. In Oracle CREATE SYNONYM is used to access remote objects while in PostgreSQL we can use SET search\_path to include the remote definition.

\* Behavior of Empty String and NULL

In Oracle, empty strings and NULL values in string context are the same. The concatenation of NULL and string obtain string as a result. In PostgreSQL the concatenation result is null in this case. In Oracle IS NULL operator is used to check whether string is empty or not but in PostgreSQL result is FALSE for empty string and TRUE for NULL.

\* Sequences

There is a slight difference in the syntax of sequence in Oracle and PostgreSQL.

Oracle:       Sequence\_name.nextval  
PostgreSQL: Nextval('sequence\_name')

\* Subquery in FROM

PostgreSQL requires a sub-SELECT surrounded by parentheses, and an alias must be provided for it. The alias is not mandatory for Oracle.

Oracle:       SELECT \* FROM (SELECT \* FROM table\_a)

PostgreSQL: SELECT \* FROM (SELECT \* FROM table\_a) AS foo

\* SUBSTR

The behavior of SUBSTR function in Oracle and PostgreSQL is different. The SUBSTR function works in PostgreSQL without error but returns a different result. This difference can cause application bugs.

\* Other Oracle built-in functions compatibility?

ROWNUM, ROWID, NVL, DECODE, TO\_CHAR(), DBMS\_OUTPUT, UTL\_FILE

\* External Coupling + (Outer Joins)

Oracle uses + operator for left and right join but PostgreSQL does not use it.

\* Database Links

Oracle:       SELECT proname, prosrc FROM pg\_proc@mydb  
                  WHERE proname LIKE 'bytea%';

PostgreSQL: SELECT \* FROM dblink('dbname=mydb',  
                                  'select proname,  
                                  prosrc from pg\_proc')  
          AS t1(proname name, prosrc text)  
          WHERE proname LIKE 'bytea%';

\* No concept of packages of stored procedures in PostgreSQL