# 3 Syntaxs Diff

Oracle可以迁移PostgreSQL(或PG)，是因为二者均为关系型数据库，并且PostgreSQL总体能覆盖Oracle的绝大部分功能;但是Oracle与PG毕竟在设计上是不同的，异构数据库迁移，有许多差异点需要格外注意。

参考：<https://my.oschina.net/liyuj/blog/539303>

<http://www.dataguru.cn/thread-499637-1-1.html>

|  |  |  |
| --- | --- | --- |
| **项目** | **Oracle** | **PostgreSQL** |
| 对象名称大小写 | 创建时不区分大小写且字典表中默认存储大写  (引号引起区分大小写，关键字必须引号引起) | 创建时不区分大小写且字典表中默认存储小写 (引号引起区分大小写，关键字必须引号引起) |
| 表 | TABLE(USER\_TABLES列表会包含雾化视图) | TABLE |
| 视图 | VIEW | VIEW |
| 序列 | SEQNAME.NEXTVAL | NEXTVAL('SEQNAME') |
| 函数 | FUNCTION | FUNCTION |
| 存储过程 | PROCEDURE | FUNCTION |
| 包 | PACKAGE | FUNCTION |
| 触发器 | TRIGGER | TRIGGER |
| 物化视图 | MATERIALIZED VIEW | MATERIALIZED VIEW |
| 同义词 | SYNONYM | VIEW |
| 连接 | DBLINK | FDW |
| ALL\_COL\_COMMENTS  注释 | 通过SELECT \* FROM ALL\_COL\_COMMENTS可以获得列注释信息 | select s.column\_name as COLUMN\_NAME,coalesce(col\_description(c.oid,ordinal\_position) ,s.column\_name) as COMMENTS from information\_schema.columns s,pg\_class c where s.table\_name = 'ac01\_si' and s.table\_name = c.relname and s.table\_schema = current\_schema(); PG需要通过col\_description获得列注释信息 |
|  |  |  |
| DUAL | SELECT 123 FROM DUAL; | 无DUAL直接执行SELECT 123; |
| 当前登录用户 | SELECT USER FROM DUAL | select current\_user; |
| 系统当前时间，常用作default值 | DEFAULT SYSDATE | DEFAULT current\_timestamp  （current\_date、current\_time） |
| ROWNUM用作行号,限制结果集或分页 | ROWNUM | ROW\_NUMBER() OVER() |
| SELECT ROWNUM,t.\* FROM tab t ROWNUM<x | SELECT \* FROM tab LIMIT 5 OFFSET 0 |
| sys\_guid()常用作default值 | DEFAULT sys\_guid() | DEFAULT uuid\_generate\_v4;  （有类似sys\_guid的函数uuid\_generate\_v4，但需要安装，create extension "uuid-oosp"） |
| NVL(a,b)函数，a参数为空那么显示b参数的值，如a参数的值不为空，则显示a参数的值。 | NVL(a,b) | COALESCE(a,b) |
| INSTR函数 | instr('str1','str2') | strpos('str1','str2') |
|  |  |  |
| MINUS求两个查询差集 | MINUS语句 | EXCEPT |
| 外连接 | Oracle可简写为(+) | 用LEFT OUTER JOIN等语句替换 |
| 层次查询 | START WITH语句 | WITH RECURSIVE语句替换 |
| CONNECT BY语句 |
| 固定值列 | SELECT '1' AS COL1 | SELECT CAST('1' AS TEXT) AS COL1 |
| 列(别)名为关键字 | Oracle中比如name，type这样的关键字可以直接作为列的别名，比如：select xx name from t | 需要加as，比如select xx as name from t |
| 子查询别名 | 如果FROM后只有一个子查询，该子查询可以没有别名 | 必须有别名 |
| 修改表字段类型 | 1.如果字段无数据，可直接修改 | 1.如果新类型和原类型兼容，可直接修改 |
| 2.如果有数据且新类型和原类型兼容，也可以直接修改 | 2.如果不兼容，需要使用USING关键字然后提供一个类型转换的表达式 |
| 3.如果不兼容，可通过对原字段改名，然后增加新字段，再通过UPDATE语句对数据进行处理 |  |
| 以下修改未完待续 | | |
| TO\_CHAR | TO\_CHAR(COL,FMT)，格式化字符串可以为空 | TO\_CHAR(COL1,'FM999999')，9的个数为字段长度，详见：https://www.postgresql.org/docs/current/static/functions-formatting.html |
| TO\_NUMBER | TO\_NUMBER(COL,FMT)，格式化字符串可以为空 | TO\_NUMBER(COL1,'999999')，9的个数为字段长度，详见：https://www.postgresql.org/docs/current/static/functions-formatting.html |
| NULL和'' | ORACLE认为''等同于NULL | NULL和''不同 |
| NULL和'' | LENGTH('')为NULL | LENGTH('')为0 |
| NULL和'' | TO\_DATE('','YYYYMMDD')为空 | TO\_DATE('','YYYYMMDD')为0001-01-01 BC |
| NULL和'' | TO\_NUMBER('',1)为NULL | TO\_NUMBER('',1)，报错 |
| NULL和'' | INSERT INTO TEST(VALUE4)VALUES('') | INSERT INTO TEST(VALUE4)VALUES('') |
| [Result]VALUE4=NULL (注：VALUE3字段为数值类型) | VALUE4=NULL |
| NULL和'' | INSERT INTO TEST(VALUE4)VALUES('') | INSERT INTO TEST(VALUE4)VALUES('') |
| [Result]VALUE4=NULL (注：VALUE3字段为字符类型) | VALUE4='' |
| NULL和'' | INSERT INTO TEST(VALUE4)VALUES(TO\_DATE('','YYYYMMDD')) | INSERT INTO TEST(VALUE6)VALUES(TO\_DATE('','YYYYMMDD')) |
| [Result]VALUE4=NULL (注：VALUE3字段为时间类型) | [Result]VALUE6=0001-01-01 BC |
| ADD\_MONTHS | ADD\_MONTHS(DATE,INT) | CREATE FUNCTION add\_months(date, int) RETURNS date AS 'SELECT ($1 +($2::text||'' month'')::interval)::date' LANGUAGE 'sql' 或：SELECT ($1 +($2::text||' month')::interval) |
| LAST\_DAY | LAST\_DAY(DATE) | CREATE OR REPLACE FUNCTION last\_day(date) RETURNS date AS $$ SELECT (date\_trunc('MONTH', $1) + INTERVAL '1 MONTH - 1 day')::date; $$ LANGUAGE 'sql'; 或：SELECT (date\_trunc('MONTH', $1) + interval '1 month - 1 day')::date; |
| MONTHS\_BETWEEN | MONTHS\_BETWEEN(DATE,DATE) | RETURNS NUMERIC AS  'SELECT (extract(year from age(d1,d2))\*12 + extract(month from age(d1,d2)))::integer' LANGUAGE 'sql'; |
| BITAND | BITAND(A,B) | A & B |
| BIN\_ | SELECT BIN\_TO\_NUM(1,0,1,0) AS VALUE1 FROM DUAL | SELECT CAST(B'1010' AS INTEGER) AS VALUE1 |
| 类型自动转换 | Oracle某些情况下支持类型自动转换 | 会出现类型不匹配等错误，需要在Java或者sql中进行类型转换，使类型匹配 |
| DECODE等判断函数 | DECODE() | 用标准的CASE WHEN THEN ELSE END语句替换 |
| SUBSTR函数 | 如果从第一个开始取子串，可以从0开始，也可以从1开始，如果不是第一个开始，则从1开始计数，可以为负值，从字符串结尾计数，用于取最后几位。 | 从1开始计数。如果要取最后几位，可以用RIGHT函数解决。 |
| UPDATE语句列列表 | UPDATE accounts SET (contact\_last\_name, contact\_first\_name) = (SELECT last\_name, first\_name FROM salesmen WHERE salesmen.id = accounts.sales\_id); | 不支持该语法，需要拆分为多个单独的列 |