

Syntaxs Diff

Oracle 可以迁移 PostgreSQL(或 HgDB), 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 SELECT ROWNUM, t.* FROM tab t ROWNUM<x	ROW_NUMBER() OVER() 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-ossp")
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	NVL(a, b)	COALESCE(a, b)

不为空，则显示 a 参数的值。		
INSTR 函数	<code>instr('str1','str2')</code>	<code>strpos('str1','str2')</code>
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 CONNECT BY 语句	WITH RECURSIVE 语句替换
固定值列	<code>SELECT '1' AS COL1</code>	<code>SELECT CAST('1' AS TEXT) AS COL1</code>
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: <code>select xx name from t</code>	需要加 as, 比如 <code>select xx as name from t</code>
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	<code>TO_CHAR(COL,FMT)</code> , 格式化字符串可以为空	<code>TO_CHAR(COL1,'FM999999')</code> , 9 的个数为字段长度, 详见: https://www.postgresql.org/docs/current/static/functions-formatting.html
TO_NUMBER	<code>TO_NUMBER(COL,FMT)</code> , 格式化字符串可以为空	<code>TO_NUMBER(COL1,'999999')</code> , 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''	<code>LENGTH('')</code> 为 NULL	<code>LENGTH('')</code> 为 0
NULL 和''	<code>TO_DATE('','YYYYMMDD')</code> 为空	<code>TO_DATE('','YYYYMMDD')</code> 为 0001-01-01 BC
NULL 和''	<code>TO_NUMBER('',1)</code> 为 NULL	<code>TO_NUMBER('',1)</code> , 报错
NULL 和''	<code>INSERT INTO TEST(VALUE4)VALUES('')</code>	<code>INSERT INTO TEST(VALUE4)VALUES('')</code>
	<code>[Result]VALUE4=NULL</code> (注: VALUE3 字段为数值类型)	<code>VALUE4=NULL</code>
NULL 和''	<code>INSERT INTO TEST(VALUE4)VALUES('')</code>	<code>INSERT INTO TEST(VALUE4)VALUES('')</code>
	<code>[Result]VALUE4=NULL</code> (注: VALUE3 字段为字符类型)	<code>VALUE4=''</code>
NULL 和''	<code>INSERT INTO TEST(VALUE4)VALUES(TO_DATE('','YYYYMMDD'))</code>	<code>INSERT INTO TEST(VALUE6)VALUES(TO_DATE('','YYYYMMDD'))</code>
	<code>[Result]VALUE4=NULL</code> (注: VALUE3 字段为时间类型)	<code>[Result]VALUE6=0001-01-01 BC</code>

ADD_MONTHS	ADD_MONTHS (DATE, INT)	<pre>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)</pre>
LAST_DAY	LAST_DAY (DATE)	<pre>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;</pre>
MONTHS_BETWEEN	MONTHS_BETWEEN (DATE, DATE)	<pre>RETURNS NUMERIC AS ' SELECT (extract(year from age(d1,d2))*12 + extract(month from age(d1,d2)))::integer' LANGUAGE 'sql';</pre>
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