

[首页](#)[博客](#)[专栏](#)[学院](#)[社区](#)[技术栈](#)[发博](#)[登录](#)[注册](#)[首页](#) > [数据库](#) > [PostgreSQL](#) > [Oracle转换Postgres](#)

Oracle转换Postgres

[翻译](#)[PostgreSQL](#)

作者: yzs87

时间: 2019-10-16 21:42:25



868



0

Oracle 转换 Postgres

1、前提

首先需要对Oracle 和 PostgreSQL 的 SQL 都比较熟悉。对其理解的越详细就越具有优势，本文帮助读者迅速理解这两类 SQL 的区别是什么。

如果因ACS/pg 而需要将 Oracle 移植到 PG ，那么就需要熟悉 AOLserver Tcl ，尤其是 SOLserver 的 API 。本文，主要讨论：

Oracle 10g 到 11g （大多数可以适用到 8i ）

Oracle 12c 某些方面会有不同，但是迁移更加便捷

PostgreSQL 8.4 ，甚至适用更早版本。

2、事务

Oracle 这个数据库会使用事务，那么 PostgreSQL 也需要激活事务。多个 DML 语句组成一个代码片段，而这些语句不会立即提交，那么就需要使用 BEGIN 语句开启一个事务，然后将这些语句包含在 BEGIN 这个块中。Oracle 和 PG 中 ROLLBACK 和 COMMIT 、SAVEPOINT 的语义相同。Oracle 的隔离级别，PostgreSQL 中也有。大多数情况下 PG 的隔离级别（读已提交）就已满足需求。

3、语法差异



yzs87

专家姓名：闫宗帅

专注于MySQL、Aerospike、PostgreSQL数据库

注册时间：2017-10-03

博文量	访问量
106	85313

最新文章

PostgreSQL异常重启postmaster.pid...

PMDK优势

非易失性内存技术及数据库

持久内存编程

PG 中有少数语法不同但功能相同 SQL 。 ACS/pg 会自动进行转换，只有大部分函数不同，需要手工进行转换。这个工作由 db_sql_prep 来完成。

函数

Oracle 有超过 250 个内置单行函数和不止 50 个聚合函数，详情查看：https://wiki.postgresql.org/wiki/Oracle_Functions

。

Sysdate

Oracle 使用 sysdate 函数获取当前日期和时间（以服务器的时区为准）。 Postgres 使用 'now'::timestamp 作为当前事务启动的日期和时间。 ACS/pg 将这个包装成 sysdate() 函数。

ACS/pg 还包括 Tcl 过程，即 db_sysdate 。因此：

```
set now [database_to_tcl_string $db "select sysdate from dual"]
```

应该变成：

```
set now [database_to_tcl_string $db "select [db_sysdate] from dual"]
```

Dual 表

Oracle 的 SELECT 中实际不需要表名的地方可以使用表 DUAL ， 因为 Oracle 中的 FROM 子句是必须的。 Postgresql 中可以将 FROM 子句丢弃。可以在 postgres 中创建一个视图作为这个表从而消除上述问题。这样就可以在不干扰 Postgres 的解析器情况下兼容 Oracle 的 SQL 。迁移过程中，尽可能去掉 “ FROM DUAL ” 子句。因为和 join 进行 join 比较奇怪。

ROWNUM 和 ROWID

Oracle 的虚拟列 ROWNUM ： 在执行 ORDER BY 前读取数据时分配一个数值。很多场景下可以使用 ROW_NUMBER() OVER(ORDER BY...) 替代。但是使用序列进行模拟时可能会使性能慢些。

PostgreSQL TID及tuple slot

PG11新特性解读：新增非空默认值字...

360°全方位比较PostgreSQL和MySQL

图解PostgreSQL--buffer的分配

RDMA网络下重思数据库高可用

NVM作为主存上对数据库管理系统的...

Oracle 的虚拟列 ROWID ：表行的物理地址，以 base64 编码。应用中可以使用该列临时缓存行地址，使第二次访问时更加便捷。 Postgres 的 ctid 起同样的作用。

序列

Oracle 的序列语法是 `sequence_name.nextval` 。

Postgres 的序列语法是 `nextval('sequence_name')` 。

Tcl 中，获取写一个序列值可以抽象为调用 `[db_sequence_nextval $db sequence_name]` 。如果需要在复杂的 SQL 语句中使用序列值，可以使用 `[db_sequence_nextval_sql sequence_name]` 。

解码

Oracle 的解码函数使用方法： `decode(expr, search, result [, search, result...] [, default])`

为了评估这个表达式，Oracle 一个一个地比较 `expr` 和 `search` 值。如果 `expr` 等于 `search` ， Oracle 返回对应的 `result` 。如果没有找到匹配值，返回 `default` 或者 `null` 。

Postgres 没有这样的结构，但是可以使用下面格式替代：

```
CASE WHEN expr THEN expr [...] ELSE expr END
```

例如： `CASE WHEN c1 = 1 THEN 'match' ELSE 'no match' END` ，返回第一个为真的谓词对应的表达式。

DECODE 和 CASE 的模拟方式有一点不同： `DECODE(x,NULL,'null','else')` ，如果 `x` 为 `NULL` 则返回 `NULL` ；而 `CASE x WHEN NULL THEN 'null' ELSE 'else' END` ，则返回 `'else'` 的 `result` 。 Oracle 同样。

NVL

Oracle 还有其他便捷函数： `NVL` 。如果不为 `NULL` ， `NVL` 返回第一个参数，否则返回第二个参数： `start_date := NVL(hire_date, SYSDATE);` 。如果 `hire_date` 为 `NULL` ，则前面的语句会返回 `SYSDATE` 。 Postgres 和 Oracle 有一个函数以更普遍的方式执行同样的行为： `coalesce(expr1, expr2, expr3,...)` ，返回第一个非 `NULL` 表达式。

FROM 中子查询

Postgresql 中子查询需要使用括号包含，并提供一个别名。 Oracle 中不需要别名：

Oracle : SELECT * FROM (SELECT * FROM table_a)

Postgresql : SELECT * FROM (SELECT * FROM table_a) AS foo

4、功能差异

Postgresql 并不具备 Oracle 所有功能。 ACS/pg 通过指定的方案解决这些限制。虽然 postgres 具备大部分功能，但是一些特性还需要等待其新版本发布。

Outer joins

Oracle 老版本 9i 之前， outer join :

```
SELECT a.field1, b.field2
```

```
FROM a, b
```

```
WHERE a.item_id = b.item_id(+)
```

(+) 表示，如果表 b 中没有匹配的 item_id 值，匹配会继续下去，会作为一个空行进行匹配。 Postgresql 和 Oracle 9i 及之前版本：

```
SELECT a.field1, b.field2
```

```
FROM a
```

```
LEFT OUTER JOIN b
```

```
ON a.item_id = b.item_id;
```

只有汇聚值从outer joined 表中提取时，也可能不使用 join 。如果原始查询：

```
SELECT a.field1, sum (b.field2)

FROM a, b

WHERE a.item_id = b.item_id (+)

GROUP BY a.field1
```

Postgres 的查询: `SELECT a.field1, b_sum_field2_by_item_id (a.item_id) FROM a` , 此时可以定义函数:

```
CREATE FUNCTION b_sum_field2_by_item_id (integer)

RETURNS integer

AS '

DECLARE

    v_item_id alias for $1;

BEGIN

    RETURN sum(field2) FROM b WHERE item_id = v_item_id;

END;

' language 'plpgsql';
```

Oracle 9i 开始将支持 SQL 99 的 `outer join` 语法。但是一些程序员仍然使用旧语法，所以这篇文章显得有意义。

`CONNECT BY`

Postgres 不支持 `connect by` 语句。可以使用 `WITH RECURSIVE` 替代。由于 `WITH RECURSIVE` 是图灵完毕的，因此很容易将 `CONNECT BY` 语句转换成 `WITH RECURSIVE` 。有时还可以将 `CONNECT BY` 当做一个简单的 `iterator` :

```
SELECT ... FROM DUAL CONNECT BY rownum <=10
```

等价于：

```
SELECT ... FROM generate_series(...)
```

NO_DATA_FOUND and TOO_MANY_ROWS

默认情况下PL/pgsql 禁止使用此异常。当需要在存储的 PLpgSQL 代码中进行单行检查时，需要在所有 SELECT 中的任何关键字 INTO 之后添加关键字 STRICT 。

5、数据类型

Postgres 严格遵从 SQL 表中，而 Oracle 由于历史原因，会有自己特有的方式，尤其是数据类型方面。

空字符串与NULL

Oracle 中， strings() 空和 NULL 在字符串内容中相同。可以将 NULL 和一个字符串连接起来作为结果。但是在 postgres 中，这种情况得到的结果是 NULL 。 Oracle 中需要使用 IS NULL 操作符来检测字符串是否为空。 Postgres 中，对于空字符串得到的结果是 FALSE ， 而 NULL 得到的是 TRUE 。当从 Oracle 向 postgres 转换时，需要分析字符代码，分离出 NULL 和空字符串。

Numeric 类型

Oracle 中经常使用 NUMBER 数据类型， PG 中对应的数据类型是 DECIMAL 或者 NUMERIC 。 PG 中的 numbers 限制（小数点前到 131072 位，小数点后 16383 位）比 Oracle 高，内部存储方式相同。 Oracle 的 FLOAT 在 PG 中是 REAL ， DOUBLE 是 DOUBLE PRECISION 。

Date and Time

Oracle 中的 DATE 包含 data 和 time 。很多中情况下，使用 PG 中的 TIMESTAMP 就足够了。由于 date 只包含秒、分、小时、天、月和年，所以一些情况下不是精确的结果。没有几分钟、没有夏令时、没有时区。 Oracle 的 TIMESTAMP 和 PG 类似。

Oracle 只有 INTERVAL YEAR TO MONTH and INTERVAL DAY TO SECOND ， 因此 PG 可以直接使用。

CLOBs

PG 以 TEXT 的形式对 CLOB 有不错的支持。

BLOBs

PG 对二进制大对象支持非常差。因为不能使用 pg_dump 进行 dump 所以不适合在 24/7 环境中使用。利用大对象的数据库进行备份时，需要将数据库关闭，然后直接备份数据目录。

Don Baccus 修改了 SOLserver 的 PG 驱动，通过编码 / 解码二进制文件，从而支持二进制大对象。数据库在运行时进行 dump ，这些结果对象可以用来保证一致性，从而在备份时不需要中断服务。

为了绕过PG 对元组大小对于一个块的限制，驱动程序将编码的数据分成 8K 大小的块。 PG 将在 2000 年夏天对大对象进行大修。因此，只实现了 ACS 使用的 BLOB 功能。

为了使用BLOB 驱动扩展，首先需要创建一个表，其 lob 列定义为 interger 类型，再创建一个触发器 on_lob_ref 。例如：

```
create table my_table (  
  
    my_key integer primary key,  
  
    lob integer references lobs,  
  
    my_other_data some_type -- etc  
  
);
```

创建一个触发器my_table_lob_trig ，在 insert 或 delete 或 update 前触发：

```
set lob [database_to_tcl_string $db "select empty_lob()"]
```

```
ns_db dml $db "begin"
```

```
ns_db dml $db "update my_table set lob = $lob where my_key = $my_key"
```

```
ns_pg blob_dml_file $db $lob $tmp_filename
```

```
ns_db dml $db "end"
```

主要，调用时需将其包装在一个事务中，即使此时没有进行update 。

```
set lob [database_to_tcl_string $db "select lob from my_table
```

```
where my_key = $my_key"]
```

```
ns_pg blob_write $db $lob
```

6、其他工具

Ispirer MnMTK：自动迁移整个数据库 schema 并将 Oracle 数据转换成 PG 的数据的工具集。

Full Convert：将 Oracle 转换成 PG，每秒 100K 个记录。

Oracle to Postgres data migration and sync：每 4-5 分钟转换 1M 个记录。基于触发器的数据库同步方法和并行双向同步方式可帮助轻松地管理数据。

ESF Database Migration Toolkit：直连 Oracle 和 PG，迁移表结构、数据、索引、主键、外键、内容等。

Orafce：兼容 Oracle 的函数。比如 date 函数（next_day,last_day,trunc,round 等）、字符串函数、一些包 DBMS_ALERT, DBMS_OUTPUT, UTL_FILE, DBMS_PIPE 等。

Ora2pg：Perl 脚本，兼容 schema。连接 Oracle，提取结构，产生 SQL 语句然后加载到 PG。

Oracle to postgres：不使用 ODBC 和其他中间件。转换表结构、数据、索引、主键和外键。

ora_migrator：PL/pgSQL 扩展，充分利用 Oracle 的 Foreign Data Wrapper。

7、原文

https://wiki.postgresql.org/wiki/Oracle_to_Postgres_Conversion

来自“ITPUB博客”，链接：<http://blog.itpub.net/31493717/viewspace-2660284/>，如需转载，请注明出处，否则将追究法律责任。

👍 点赞 | 0

☆ 收藏 | 0

分享到：



上一篇： XLOG段文件跳号现象分析

下一篇： NVM作为主存上对数据库管理系统的影响



全部评论



[支持我们](#) | [作者招募](#) | [用户协议](#) | [FAQ](#) | [Contact Us](#) | [站长统计](#)



北京盛拓优讯信息技术有限公司. 版权所有 京ICP备09055130号-4 北京市公安局海淀分局网监中心备案编号: 11010802021510

广播电视节目制作经营许可证(京)字第1234号 中国互联网协会会员