



# 开飞机的老张

写给未来的信

[首页](#) [归档](#) [标签](#) [关于](#)

## PostgreSQL与Oracle的语法差异总结

· 2019-07-07 ·



# PostgreSQL



PostgreSQL  
the world's most advanced open source database

为了降低成本、顺应开源化潮流，公司从Oracle转向了PostgreSQL。由于是连平台一块换了，不存在迁移的问题，所以不需要使用orafce、Ora2pg之类的转换迁移工具，只需要了解语法差异，转换比较平稳。

PG目前是数据库榜单上的探花，坐观榜眼MySQL和状元Oracle，学习一下还是很有价值的！

## JDBC连接

- Oracle的JDBC连接字符串：

db.url=jdbc:oracle:thin:@192.168.1.1:1521:ORCL

- PostgreSQL的JDBC连接字符串：

db.url=jdbc:postgresql:@192.168.1.1:5432/database

## 数据类型

Oracle	PostgreSQL
--------	------------

Oracle	PostgreSQL
Oracle varchar2	PostgreSQL varchar
number	numeric
date	timestamp/date/time
不支持boolean	支持boolean
null	null

## 数据类型转换

Oracle和PostgreSQL都支持to\_char()、to\_number()、to\_date()的函数进行字符类型、数值类型和日期类型的转换。

Oracle在数据类型转换方面比较智能，能够在SQL中自动将数据转换为字段对应的类型，例如：

```
create tableA (name varchar2(36), score integer);  
-- '13' 被自动转换为13  
update tableA set score='13' where name='张三';
```

PostgreSQL在类似场景下就比较严格：

```
create tableA (name varchar(36), score int);  
-- '13' 被转换为13, ::是PG特有的简化语法，等价于标准SQL的CAST('13' AS INT)  
update tableA set score='13'::int where name='张三';
```

## 基本语法

函数	Oracle	PostgreSQL
字符串连接		concat()
'a'    null	'a'	null
系统当前时间	sysdate	now(),current_timestamp,current_date,current_time
对时间截取	trunc()	date_trunc()
判空操作	nvl()	coalesce()
条件判断	decode()	case...when...then...else...end
dual伪表	支持	select now();
外联	(+)	left join,right join

GOTO	支持	不支持
procedure	支持	改为function
package	支持	改为function

## MERGE INTO和on conflict() do

Oracle和PostgreSQL增删改查语法基本一致，有一点不同：

Oracle有强大的MERGE INTO：

```
MERGE INTO tableA AT a
USING tableB AS b
ON a.id=b.id
WHEN MATCHED--匹配则更新
THEN UPDATE SET a.field=b.field
WHEN NOT MATCHED--无匹配则插入
THEN INSERT VALUES (b.id,b.field)
WHEN NOT MATCHED BY SOURCE--源表不存在则删除目标表
THEN DELETE;
```

PostgreSQL可以使用on conflict() do:

```
insert into tableA
select id,field from tableB

on conflict (id)
do update set field = exclude.field
```

## 查询语句

1. 查询表中最新n条数据

Oracle用rownum:

写法一:

```
select t.* from (select * from tableA order by create_time desc) t where rownum <=
n;
```

写法二:

```
select * from(select t.*, row_number() over(order by create_time desc) rn from tableA t) where rn <=n;
```

上述写法一为通用常规写法; 写法二可以对分组后数据排序, 分组语句写在over()中

postgresql用limit:

```
select * from tableA order by create_time desc limit n;
```

注意: limit必须用于 order by 之后

## 2. 子查询

Oracle子查询用括号包起来即可, 无需加别名 (Alias) :

```
select name from (select id,name from tableA) ;
```

postgresql子查询要求比较严格, 必须具有别名:

```
select a.name from (select id,name from tableA) a;
```

## 事务

Oracle 中, 通过 commit/rollback来实现事务提交或回滚。结构类似于:

```
begin
    select ...
    update ...
```

```
.
    delete ...
    commit;
exception
    when others then
        rollback;
end;
```

PostgreSQL 实际上把每个 SQL 语句当做在一个事务中执行来看待。

如果你没有发出BEGIN命令，那么每个独立的语句都被一个隐含的BEGIN 和(如果成功的话)COMMIT包围。

一组包围在BEGIN和COMMIT 之间的语句有时候被称做事务块。例如：

```
BEGIN;
UPDATE accounts SET balance = balance - 100.00
    WHERE name = 'Alice';
SAVEPOINT my_savepoint;
UPDATE accounts SET balance = balance + 100.00
    WHERE name = 'Bob';
-- 呀！加错钱了，应该用 Wally 的账号
ROLLBACK TO my_savepoint;
UPDATE accounts SET balance = balance + 100.00
    WHERE name = 'Wally';
COMMIT;
```

在 PL/pgSQL中，也提供了类似于Oracle 的 Begin、End及ExceptionCode处理机制。他们之间的差异不明显。事实上，PL/SQL（Oracle数据库操作语言）与PL/pgSQL是高度相似的，这让procedure在Oracle与 Postgresql之间迁移变得极为便捷。

下一篇

## PostgreSQL的行转列和列转行

0 comment

Anonymous ▾



Leave a comment

[Markdown is supported](#)

Login with GitHub

Preview

Be the first person to leave a comment!



京公网安备 11010802032021号

[Email](#) | [RSS](#)