

消失的海岸线

永久域名 <http://eternal1025.javaeye.com>

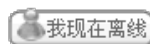


eternal1025

浏览: 16762 次

性别:

来自: 北京



[详细资料](#)
[留言簿](#)

搜索本博客

最近访客
 [>>更多访客](#)



博客分类

- [全部博客 \(51\)](#)
- [Java相关 \(10\)](#)
- [数据库 \(20\)](#)
- [JavaScript相关 \(7\)](#)
- [jsp相关 \(6\)](#)
- [2010年工作项目 \(2\)](#)
- [GIS相关 \(1\)](#)
- [CSS \(1\)](#)
- [JQuery学习 \(2\)](#)

其他分类

- [我的收藏 \(0\)](#)
- [我的论坛主题贴 \(0\)](#)
- [我的所有论坛贴 \(0\)](#)

2009-04-02

[java 类型转换](#)
|
[oracle主键自增](#)

Oracle join

Java代码

- 条件连接(join)
- T1 { [INNER] | { LEFT | RIGHT | FULL } [OUTER] } JOIN T2 ON boolean_expression
- T1 { [INNER] | { LEFT | RIGHT | FULL } [OUTER] } JOIN T2 USING (join column list)
- T1 NATURAL { [INNER] | { LEFT | RIGHT | FULL } [OUTER] } JOIN T2
- INNER 和 OUTER 对所有连接 (join) 类型都是可选的. INNER 是缺省; LEFT, RIGHT, 和 FULL 隐舍外连接.
-
- 连接条件在ON或USING子句里声明, 或者用关键字NATURAL隐舍地声明. 连接条件判断来自两个源表 中的那些行是"匹配"的, 这些我们将在下面详细解释.
-
- ON子句是最常见的连接条件的类型: 它接收一个和WHERE子句里用的一样的 布尔值表达式. 如果两个分别来自T1和T2的行在ON表达式上运算的 结果为真, 那么它们就算是匹配的行.
-
- USING是缩写的概念: 它接收一个用逗号分隔的字段名字列表, 这些字段必须是连接表共有的, 最终形成一个连接条件, 表示 这些字段对必须相同. 最后, JOIN USING 的输出会为每一对相等 的输入字段输出一个字段, 后面跟着来自各个表的所有其它字段. 因此, USING (a, b, c) 等效于 ON (t1.a = t2.a AND t1.b = t2.b AND t1.c = t2.c) 只不过是如果使用了ON, 那么在结果里 a, b, 和 c字段每个都会有两个, 而用USING的时候每个字段就只会一个.
-
- 最后, NATURAL 是USING 的缩写形式: 它形成一个 USING 列表, 该列表由那些在两个表里都出现了的字段名字组成. 和USING一样, 这些字段只在输出表里出现一次.
-
- 条件JOIN的可能类型是:
-
- INNER JOIN
- 对于 T1 的每一行 R1, 生成的连接表都有一行对应 T2 中的 每一个满足和 R1 的连接条件的行.
-
- LEFT OUTER JOIN
- 首先, 执行一次内连接. 然后, 为 T1 里那些和 T2 里任何一行都不满足连接条件的行返回一个连接行, 同时该连接行里对应 T2 的列用空值补齐. 因此, 生成的连接表里无条件地包含来自 T1 里的每一行至少 一个副本.
-
- RIGHT OUTER JOIN
- 首先, 执行一次内连接. 然后, 为 T2 里那些和 T1 里任何一行都不满足连接条件的行返回一个连接行, 同时该连接行里对应 T1 的列用空值补齐. 因此, 生成的连接表里无条件地包含来自 T2 里的每一行.
-
- FULL OUTER JOIN
- 首先, 执行一次内连接. 然后, 为 T1 里那些和 T2 里任何一行都不满足连接条件的行返回一个连接行, 同时该连接行里对应 T2 的列用空值补齐. 同样, 为 T2 里那些和 T1 里的任何行都不满足连接条件的 行返回一个连接行, 该行里对应 T1 的列用空值补齐.
-
- 如果 T1 和 T2 有一个或者都是可以连接(join)的表, 那么所有类型的连接都可以串在一起或嵌套在一起. 你可以在 JOIN子句周围使用圆括弧来控制连接顺序, 如果没有圆括弧, 那么JOIN子句是从左向右嵌套的.
-
- 为了解释这些问题, 假设我们有一个表 t1
-
- ```

num | name
-----+-----
1 | a
2 | b
3 | c

```
- 和 t2

[我的精华良好贴](#) (0)

最近加入圈子

▪ [EXT](#)

存档

▪ [2010-07](#) (5)

▪ [2010-06](#) (4)

▪ [2010-05](#) (1)

▪ [更多存档...](#)

最新评论

▪ [经常遇到一些java.sql.SQL...](#)

invalid column name 无效列名

cannot insert N ...

-- by [javaAlpha](#)

评论排行榜

▪ [hibernate createSQLQuery](#)

▪ [JavaScript中取到form的值的方  
法](#)

▪ [滚动条](#)

▪ [关于js的验证](#)

▪ [@后面的String](#)



[\[什么是RSS?\]](#)

```
40.
41. num | value
42. -----+-----
43. 1 | xxx
44. 3 | yyy
45. 5 | zzz
46. 然后用不同的连接方式可以获得各种结果:
47.
48. => SELECT * FROM t1 CROSS JOIN t2;
49. num | name | num | value
50. -----+-----+-----+-----
51. 1 | a | 1 | xxx
52. 1 | a | 3 | yyy
53. 1 | a | 5 | zzz
54. 2 | b | 1 | xxx
55. 2 | b | 3 | yyy
56. 2 | b | 5 | zzz
57. 3 | c | 1 | xxx
58. 3 | c | 3 | yyy
59. 3 | c | 5 | zzz
60. (9 rows)
61.
62. => SELECT * FROM t1 INNER JOIN t2 ON t1.num = t2.num;
63. num | name | num | value
64. -----+-----+-----+-----
65. 1 | a | 1 | xxx
66. 3 | c | 3 | yyy
67. (2 rows)
68.
69. => SELECT * FROM t1 INNER JOIN t2 USING (num);
70. num | name | value
71. -----+-----+-----
72. 1 | a | xxx
73. 3 | c | yyy
74. (2 rows)
75.
76. => SELECT * FROM t1 NATURAL INNER JOIN t2;
77. num | name | value
78. -----+-----+-----
79. 1 | a | xxx
80. 3 | c | yyy
81. (2 rows)
82.
83. => SELECT * FROM t1 LEFT JOIN t2 ON t1.num = t2.num;
84. num | name | num | value
85. -----+-----+-----+-----
86. 1 | a | 1 | xxx
87. 2 | b | |
88. 3 | c | 3 | yyy
89. (3 rows)
90.
91. => SELECT * FROM t1 LEFT JOIN t2 USING (num);
92. num | name | value
93. -----+-----+-----
94. 1 | a | xxx
95. 2 | b |
96. 3 | c | yyy
97. (3 rows)
98.
99. => SELECT * FROM t1 RIGHT JOIN t2 ON t1.num = t2.num;
100. num | name | num | value
101. -----+-----+-----+-----
102. 1 | a | 1 | xxx
103. 3 | c | 3 | yyy
104. | | 5 | zzz
```

```
105. (3 rows)
106.
107. => SELECT * FROM t1 FULL JOIN t2 ON t1.num = t2.num;
108. num | name | num | value
109. -----+-----+-----+-----
110. 1 | a | 1 | xxx
111. 2 | b | |
112. 3 | c | 3 | yyy
113. | | 5 | zzz
114. (4 rows)
```

用 ON 声明的连接条件也可以包含与连接不直接相关 的条件。这种功能可能对某些查询很有用，但是需要我们仔细想想。 比如：

```
117.
118. => SELECT * FROM t1 LEFT JOIN t2 ON t1.num = t2.num AND t2.value = 'xxx';
119. num | name | num | value
120. -----+-----+-----+-----
121. 1 | a | 1 | xxx
122. 2 | b | |
123. 3 | c | |
124. (3 rows)
```

[Download 20day Free trial](#)  
UML + Mind Map = \$ 50 per year Smooth modeling with astah\*  
[www.change-vision.com](http://www.change-vision.com)



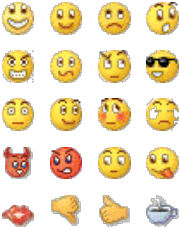
[java 类型转换](#) | [oracle主键自增](#)

19:02 | [浏览 \(544\)](#) | [评论 \(0\)](#) | 分类: [数据库](#) | [相关推荐](#)

评论

发表评论

表情图标



字体颜色:  字体大小:  对齐:

提示: 选择您需要装饰的文字, 按上列按钮即可添加上相应的标签

---

声明：JavaEye 文章版权属于作者，受法律保护。没有作者书面许可不得转载。若作者同意转载，必须以超链接形式标明文章原始出处和作者。

© 2003-2010 JavaEye.com. All rights reserved. 上海炯耐计算机软件有限公司 [ 沪ICP备05023328号 ]