[PostgreSQL](http://blog.chinaunix.net/uid-15145533-id-2775821.html)

[9.0新特性介绍：唯一约束的增强](http://blog.chinaunix.net/uid-15145533-id-2775821.html)

分类：

2010-08-22 10:49:50

    在PostgreSQL9.0提供了一个新特性： Deferrable unique constraints, now permit mass updates to unique keys，基本意思是，对于可延迟的唯一约束，允许集中更新后再统一做约束检查。在PostgreSQL9.0以前的版本中，只有外键约束可以定义为deferrable，其它的都不行，而PostgreSQL9.0中，可为deferrable的约束类型还增加了UNIQUE, PRIMARY KEY, EXCLUDE。

    在PostgreSQL9.0版本之前，在执行DML语句时，唯一约束是不能deferable的，所以每更新一行，就要检查一次唯一性约束。所以，如果有一张表上有一个唯一字段叫id，内容是按顺序增长的，执行一条SQL语句： update t set id=id+1，这条SQL就会报错。实际上这条SQL语句如果更新完所有数据，是没有唯一性冲突的。这条SQL在oracle数据库中是可以正常执行完的，也不会报错。

    而PostgreSQL9.0之后，当创建的约束为“Deferable”后，不论你用set constraints all immediate，还是set constraints all deferrd时，这条SQL都不会了报错了，原因是当表上的唯一约束为“Deferable”后，当执行DML语句做批量更新时，是整个更新完后，才开始做唯一性检查，而不是每更新完一行后就检查。

见下面的实验：

环境为PostgreSQL9.0 beta4，

先不加deferable参数，约束默认是no deferable的：

postgres=# create table t4 (id int primary key,name varchar(40));

NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "t4\_pkey" for table "t4"

CREATE TABLE

postgres=# insert into t4 values(1,'1111111111111111');

INSERT 0 1

postgres=# insert into t4 values(2,'2222222222222222');

INSERT 0 1

postgres=#

postgres=# update t4 set id=id+1;

ERROR:  duplicate key value violates unique constraint "t4\_pkey"

DETAIL:  Key (id)=(2) already exists.

STATEMENT:  update t4 set id=id+1;

ERROR:  duplicate key value violates unique constraint "t4\_pkey"

DETAIL:  Key (id)=(2) already exists.

默认约束是no deferable的，这时可以看到SQL执行出错。

执行set constraints all deferred:

postgres=# begin;

BEGIN

postgres=# set constraints all deferred;

SET CONSTRAINTS

postgres=# update t4 set id=id+1;

ERROR:  duplicate key value violates unique constraint "t4\_pkey"

DETAIL:  Key (id)=(2) already exists.

STATEMENT:  update t4 set id=id+1;

ERROR:  duplicate key value violates unique constraint "t4\_pkey"

DETAIL:  Key (id)=(2) already exists.

postgres=# end;

ROLLBACK

可以看出即使设置了constraints为deferred，由于建表时约束为no deferred的，也是就约束是不能延迟的检查，执行时，即使设置constraints为deferred时，也不生效。

下面我们把约束建成deferable的：

postgres=# drop table t4;

DROP TABLE

postgres=# create table t4 (id int primary key DEFERRABLE,name varchar(40));

NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "t4\_pkey" for table "t4"

CREATE TABLE

postgres=# insert into t4 values(1,'1111111111111111');

INSERT 0 1

postgres=# insert into t4 values(2,'2222222222222222');

INSERT 0 1

postgres=# update t4 set id=id+1;

UPDATE 2

可以看到执行成功。

同样，我们在执行时，把constraints设置为immediate时，也可以执行成功：

postgres=# begin;

BEGIN

postgres=# set constraints all IMMEDIATE;

SET CONSTRAINTS

postgres=# update t4 set id=id+1;

UPDATE 2

postgres=# end;

COMMIT

如果执行时设置constraints为deferred时，然后执行一个会导致唯一约束冲突的更新，可以看到到提交时，才检查约束：

postgres=# begin;

BEGIN

postgres=# set constraints all deferred;

SET CONSTRAINTS

postgres=# update t4 set id=2 where id=1;

UPDATE 1

postgres=# end;

ERROR:  duplicate key value violates unique constraint "t4\_pk"

DETAIL:  Key (id)=(2) already exists.

STATEMENT:  end;

ERROR:  duplicate key value violates unique constraint "t4\_pk"

DETAIL:  Key (id)=(2) already exists.

# [**数据库备份策略**](https://www.cnblogs.com/hellojesson/p/8376030.html)

odoo默认的数据库为postgresql数据库，

PG是个非常强大的数据库，也是未来的一个趋势。

对于odoo的数据备份，odoo提供了自己的备份方式，

1. 从前台页面。输入odoo应用访问地址，例如：192.168.1.188:8069/web/database/manager 可以直接选择自己需要的数据库，进行手动复制，备份，和删除。

2. odoo应用社区也提供了一些第三方备份的模块，感兴趣的可以自行搜索。

3. 从数据库级别，即：按照常规数据库的备份恢复来操作。

    这里需要注意一点，对于单实例的非SAAS架构的odoo程序，完全可以按照数据库的常规方式去处理。

    但是，对于SAAS架构的odoo应用程序，就需要注意自己备份的数据库ID，恢复数据库的时候是否还能和主应用程序SAAS模块的管理ID相对应！

备份策略待更新！

# [**数据库 对比、同步工具推荐**](https://www.cnblogs.com/hellojesson/p/9205705.html)

1. dbForge Data Compare for PostgreSQL
2. PostgreSQL Data Sync

# [**Postgresql 字符串操作函数**](https://www.cnblogs.com/hellojesson/p/8044095.html)

样例测试：

update property set memorial\_no = btrim(memorial\_no, ' ') where memorial\_no like ' %'

或：update property set memorial\_no = trim(both ' ' from memorial\_no) where memorial\_no like ' %'

btrim()方法为删除字符串两边的某类字符，可以同时指定多个，在上面为' '，意为空格

trim()方法可以实现所有btrim()能实现的功能，事实上btrim()即为删除两边的某类字符串，trim()可以只指定某一边，当然也可以为两边

现把Postgresql的字符串操作函数罗列在以，以便日方使用

函数：string || string　  
说明：String concatenation 字符串连接操作  
例子：'Post' || 'greSQL'　＝　PostgreSQL

函数：string || non-string or non-string || string  
说明：String concatenation with one non-string input 字符串与非字符串类型进行连接操作  
例子：'Value: ' || 42 = Value: 42

函数：bit\_length(string)  
说明：Number of bits in string 计算字符串的位数  
例子：bit\_length('jose') = 32

函数：char\_length(string) or character\_length(string)  
说明：Number of characters in string 计算字符串中字符个数  
例子：char\_length('jose') = 4

函数：lower(string)  
说明：Convert string to lower case 转换字符串为小写  
例子：bit\_length('jose') = 32

函数：octet\_length(string)  
说明：Number of bytes in string 计算字符串的字节数  
例子：octet\_length('jose') = 4

函数：overlay(string placing string from int [for int])  
说明：Replace substring 替换字符串中任意长度的子字串为新字符串  
例子：overlay('Txxxxas' placing 'hom' from 2 for 4) = 4

函数：position(substring in string)  
说明：Location of specified substring 子串在一字符串中的位置  
例子：position('om' in 'Thomas') = 3

函数：substring(string [from int] [for int])  
说明：Extract substring 截取任意长度的子字符串  
例子：substring('Thomas' from 2 for 3) = hom

函数：substring(string from pattern)  
说明：Extract substring matching POSIX regular expression. See Section 9.7 for more information on pattern matching. 利用正则表达式对一字符串进行任意长度的字串的截取  
例子：substring('Thomas' from '...$') = mas

函数：substring(string from pattern for escape)  
说明：Extract substring matching SQL regular expression. See Section 9.7 for more information on pattern matching. 利于正则表达式对某类字符进行删除，以得到子字符串  
例子：trim(both 'x' from 'xTomxx') = Tom

函数：trim([leading | trailing | both] [characters] from string)   
说明：Remove the longest string containing only the characters (a space by default) from the start/end/both ends of the string 去除尽可能长开始，结束或者两边的某类字符，默认为去除空白字符，当然可以自己指定，可同时指定多个要删除的字符串  
例子：trim(both 'x' from 'xTomxx') = Tom

函数：upper(string)  
说明：Convert string to uppercase 将字符串转换为大写  
例子：upper('tom') = TOM

函数：ascii(string)  
说明：ASCII code of the first character of the argument. For UTF8 returns the Unicode code point of the character. For other multibyte encodings. the argument must be a strictly ASCII character. 得到某一个字符的Assii值  
例子：ascii('x') = 120

函数：btrim(string text [, characters text])  
说明：Remove the longest string consisting only of characters in characters (a space by default) from the start and end of string 去除字符串两边的所有指定的字符，可同时指定多个字符  
例子：btrim('xyxtrimyyx', 'xy') = trim

函数：chr(int)  
说明：Character with the given code. For UTF8 the argument is treated as a Unicode code point. For other multibyte encodings the argument must designate a strictly ASCII character. The NULL (0) character is not allowed because text data types cannot store such bytes. 得到某ACSII值对应的字符  
例子：chr(65) = A

函数：convert(string bytea, src\_encoding name, dest\_encoding name)   
说明：Convert string to dest\_encoding. The original encoding is specified by src\_encoding. The string must be valid in this encoding. Conversions can be defined by CREATE CONVERSION. Also there are some predefined conversions. See Table 9-7 for available conversions. 转换字符串编码，指定源编码与目标编码  
例子：convert('text\_in\_utf8', 'UTF8', 'LATIN1') = text\_in\_utf8 represented in ISO 8859-1 encoding

函数：convert\_from(string bytea, src\_encoding name)   
说明：Convert string to the database encoding. The original encoding is specified by src\_encoding. The string must be valid in this encoding. 转换字符串编码，自己要指定源编码,目标编码默认为数据库指定编码，  
例子：convert\_from('text\_in\_utf8', 'UTF8') = text\_in\_utf8 represented in the current database encoding

函数：convert\_to(string text, dest\_encoding name)   
说明：Convert string to dest\_encoding.转换字符串编码，源编码默认为数据库指定编码，自己要指定目标编码,  
例子：convert\_to('some text', 'UTF8') = some text represented in the UTF8 encoding

函数：decode(string text, type text)   
说明：Decode binary data from string previously encoded with encode. Parameter type is same as in encode. 对字符串按指定的类型进行解码  
例子：decode('MTIzAAE=', 'base64') = 123\000\001

函数：encode(data bytea, type text)   
说明：Encode binary data to different representation. Supported types are: base64, hex, escape. Escape merely outputs null bytes as \000 and doubles backslashes. 与decode相反，对字符串按指定类型进行编码  
例子：encode(E'123\\000\\001', 'base64') = MTIzAAE=

函数：initcap(string)  
说明：Convert the first letter of each word to uppercase and the rest to lowercase. Words are sequences of alphanumeric characters separated by non-alphanumeric characters. 将字符串所有的单词进行格式化，首字母大写，其它为小写  
例子：initcap('hi THOMAS') = Hi Thomas

函数：length(string)  
说明：Number of characters in string 讲算字符串长度  
例子：length('jose') = 4

函数：length(stringbytea, encoding name )  
说明：Number of characters in string in the given encoding. The string must be valid in this encoding. 计算字符串长度，指定字符串使用的编码  
例子：length('jose', 'UTF8') = 4

函数：lpad(string text, length int [, fill text])   
说明：Fill up the string to length length by prepending the characters fill (a space by default). If the string is already longer than length then it is truncated (on the right). 对字符串左边进行某类字符自动填充，即不足某一长度，则在左边自动补上指定的字符串，直至达到指定长度，可同时指定多个自动填充的字符  
例子：lpad('hi', 5, 'xy') = xyxhi

函数：ltrim(string text [, characters text])   
说明：Remove the longest string containing only characters from characters (a space by default) from the start of string 删除字符串左边某一些的字符，可以时指定多个要删除的字符  
例子：trim

函数：md5(string)  
说明：Calculates the MD5 hash of string, returning the result in hexadecimal 将字符串进行md5编码  
例子：md5('abc') = 900150983cd24fb0 d6963f7d28e17f72

函数：pg\_client\_encoding()  
说明：Current client encoding name 得到pg客户端编码  
例子：pg\_client\_encoding() = SQL\_ASCII

函数：quote\_ident(string text)  
说明：Return the given string suitably quoted to be used as an identifier in an SQL statement string. Quotes are added only if necessary (i.e., if the string contains non-identifier characters or would be case-folded). Embedded quotes are properly doubled. 对某一字符串加上两引号  
例子：quote\_ident('Foo bar') = "Foo bar"

函数：quote\_literal(string text)  
说明：Return the given string suitably quoted to be used as a string literal in an SQL statement string. Embedded single-quotes and backslashes are properly doubled. 对字符串里两边加上单引号，如果字符串里面出现sql编码的单个单引号，则会被表达成两个单引号  
例子：quote\_literal('O\'Reilly') = 'O''Reilly'

函数：quote\_literal(value anyelement)  
说明：Coerce the given value to text and then quote it as a literal. Embedded single-quotes and backslashes are properly doubled. 将一数值转换为字符串，并为其两边加上单引号，如果数值中间出现了单引号，也会被表示成两个单引号  
例子：quote\_literal(42.5) = '42.5'

函数：regexp\_matches(string text, pattern text [, flags text])  
说明：Return all captured substrings resulting from matching a POSIX regular expression against the string. See Section 9.7.3 for more information. 对字符串按正则表达式进行匹配，如果存在则会在结果数组中表示出来  
例子：regexp\_matches('foobarbequebaz', '(bar)(beque)') = {bar,beque}

函数：regexp\_replace(string text, pattern text, replacement text [, flags text])  
说明：Replace substring(s) matching a POSIX regular expression. See Section 9.7.3 for more information. 利用正则表达式对字符串进行替换  
例子：regexp\_replace('Thomas', '.[mN]a.', 'M') = ThM

函数：regexp\_split\_to\_array(string text, pattern text [, flags text ])  
说明：Split string using a POSIX regular expression as the delimiter. See Section 9.7.3 for more information. 利用正则表达式将字符串分割成数组  
例子：regexp\_split\_to\_array('hello world', E'\\s+') = {hello,world}

函数：regexp\_split\_to\_table(string text, pattern text [, flags text])  
说明：Split string using a POSIX regular expression as the delimiter. See Section 9.7.3 for more information. 利用正则表达式将字符串分割成表格  
例子：regexp\_split\_to\_table('hello world', E'\\s+') =   
hello  
world  
(2 rows)

函数：repeat(string text, number int)  
说明：Repeat string the specified number of times 重复字符串一指定次数  
例子：repeat('Pg', 4) = PgPgPgPg

函数：replace(string text, from text, to text)  
说明：Replace all occurrences in string of substring from with substring to 将字符的某一子串替换成另一子串  
例子：('abcdefabcdef', 'cd', 'XX') = abXXefabXXef

函数：rpad(string text, length int [, fill text])   
说明：Fill up the string to length length by appending the characters fill (a space by default). If the string is already longer than length then it is truncated. 对字符串进行填充，填充内容为指定的字符串  
例子：rpad('hi', 5, 'xy') = hixyx

函数：rtrim(string text [, characters text])  
说明：Remove the longest string containing only characters from characters (a space by default) from the end of string   
去除字符串右边指定的字符  
例子：rtrim('trimxxxx', 'x') = trim

函数：split\_part(string text, delimiter text, field int)  
说明：Split string on delimiter and return the given field (counting from one)  对字符串按指定子串进行分割，并返回指定的数值位置的值  
例子：split\_part([mailto:&amp;apos;abc~@~def~@~ghi&amp;apos;](mailto:%26amp;amp;apos;abc~@~def~@~ghi%26amp;amp;apos;), [mailto:&amp;apos;~@~&amp;apos;](mailto:%26amp;amp;apos;~@~%26amp;amp;apos;), 2) = def

函数：strpos(string, substring)  
说明：Location of specified substring (same as position(substring in string), but note the reversed argument order) 指定字符串在目标字符串的位置  
例子：strpos('high', 'ig') = 2

函数：substr(string, from [, count])  
说明：Extract substring (same as substring(string from from for count)) 截取子串  
例子：substr('alphabet', 3, 2) = ph

函数：to\_ascii(string text [, encoding text])  
说明：Convert string to ASCII from another encoding (only supports conversion from LATIN1, LATIN2, LATIN9, and WIN1250 encodings) 将字符串转换成ascii编码字符串  
例子：to\_ascii('Karel') = Karel

函数：to\_hex(number int or bigint)  
说明：Convert number to its equivalent hexadecimal representation 　对数值进行十六进制编码  
例子：to\_hex(2147483647) = 7fffffff

函数：translate(string text, from text, to text)   
说明：Any character in string that matches a character in the from set is replaced by the corresponding character in the to set 将字符串中某些匹配的字符替换成指定字符串，目标字符与源字符都可以同时指定多个  
例子：translate('12345', '14', 'ax') = a23x5

https://www.cnblogs.com/stephen-liu74/archive/2012/05/02/2294071.html

http://www.jsjtt.com/shujuku/postgresql/29.html

# [**postgresql重新启动、状态查看**](https://www.cnblogs.com/hellojesson/p/7149421.html)

场景描述：

当遇到数据库不能正常访问的时候，我们首先想到的是，查看相关的告警日志，一般先查看系统的日志，然后查看数据库的日志，Linux平台下，postgresql的日志文件存放目录在如下路径：

test@odoo:/var/log/postgresql$ pwd

/var/log/postgresql

将日志的信息实时打印输出：

test@odoo:/var/log/postgresql$ tail -f postgresql-9.5-main.log

如果需要重启数据库

利用psql启动**数据库**

**首先切换到postgres用户下，**

[postgres@test~]$ pg\_ctl start

查看系统中运行的postgres进程

#ps -ef | grep postgres

连接postgresql数据库

#psql -h 127.0.0.1 -d postgres -U postgres

停止postgresql数据库实例

#pg\_ctl stop

#ps -ef | grep postgres

启动服务器最简单的方法是像下面这样：

$ postgres -D /usr/local/pgsql/data

最好在后台启动postgres，使用下面的 Unix shell 语法：

$ postgres -D /usr/local/pgsql/data >logfile 2>&1 &

要做一次快速关闭：

$ kill -INT `head -1 /usr/local/pgsql/data/postmaster.pid`

如果启动失败提示一下信息：

LOG: could not bind IPv4 socket: Address already in use

HINT: Is another postmaster already running on port 5432? If not, wait a few seconds and retry.

FATAL: could not create TCP/IP listen socket

解决方法：

查看端口在用

netstat -lanp|grep 5432

ps -ef|grep postgres

498 673 1 0 10:50 ? 00:00:00/usr/bin/postgres -D /var/lib/cloudera-scm-server-db/data

kill进程  
kill -9 673

重启启动服务

service start

service status

 其他参考：

http://blog.csdn.net/Dracotianlong/article/details/7828515

# [**数据库默认端口**](https://www.cnblogs.com/hellojesson/p/6432149.html)

Oracle 默认端口：1521

Mysql 默认端口： 3306

PostgreSQL默认端口：5432

Redis默认端口：6379

Mogondb默认端口：27017

DB2 默认端口号为：5000