简介，大数据时代的最佳数据库

[加州大学](https://baike.baidu.com/item/%E5%8A%A0%E5%B7%9E%E5%A4%A7%E5%AD%A6)伯克利分校计算机系开发90年代。（这个学校很牛逼，spakr）,mysql 1.0。

postgresql代表着业界的先进潮流，引导着数据库界的开发方向；

大数据时代，互联网厂商基于postgresql 开发的趋势。greenplumn  HAWQ等

起点高，学院派，天生可靠。

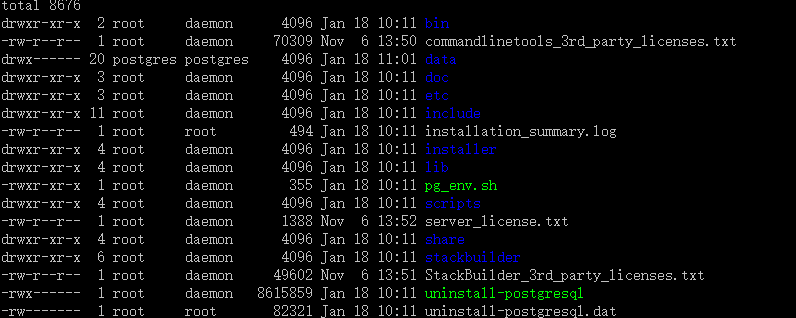
干0 ，google事件和oracle辱华事件后，干O成为政治正确，涉及产业安全。

它拥有超过15年的积极开发和经过验证的架构，在可靠性，数据完整性和正确性方面赢得了良好声誉。它运行在所有主要操作系统上，包括Linux，UNIX（AIX，BSD，HP-UX，macOS，Solaris）和Windows。它完全兼容ACID，完全支持外键，连接，视图，触发器和存储过程（使用多种语言）。它包括大多数SQL：2008数据类型，包括INTEGER，NUMERIC，BOOLEAN，CHAR，VARCHAR，DATE，INTERVAL和TIMESTAMP。它还支持存储二进制大对象，包括图片，声音或视频。它具有用于C / C ++，Java，.Net，Perl，Python，Ruby，Tcl，ODBC等的本机编程接口

注：基于postgresql-10 和 centos 7讲解，postgresql-10选择的是 edb：postgresql-10.linux.run

零、体系结构

目录结构：

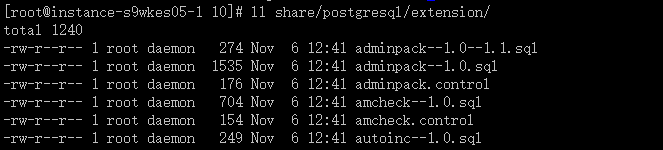


bin  可执行文件  pg\_ctl   pgsql  等

data  数据目录

share  extension目录 plpython,pg\_jieba等插件所在目录

lib   多为c依赖包

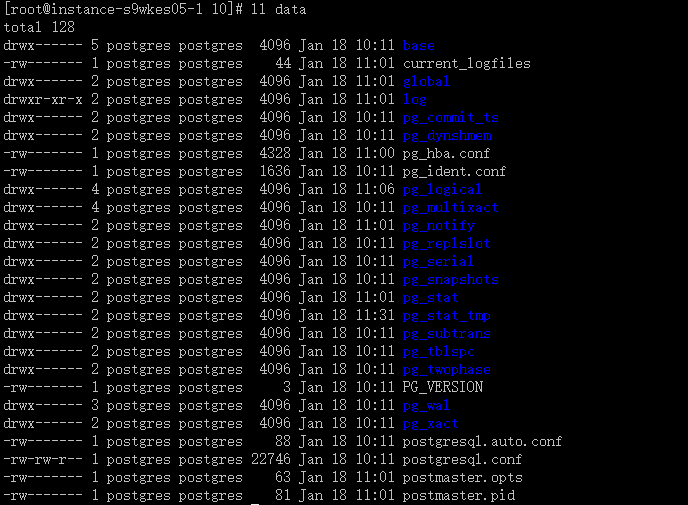


说一下这个share，和以后的extension有关，一个extension约等于 一个 xx.sql+xx.control

这些都是可以自定义开发的

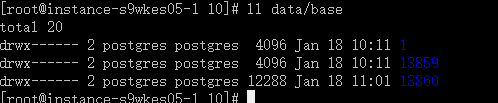
.sql 是安装文件   .control是提供元数据信息

data目录

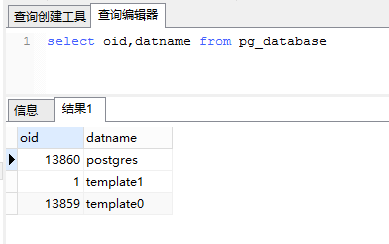


配置文件：pg\_hba.conf  postgresql.conf

base: 默认表空间



表空间下的id就是数据库的oid

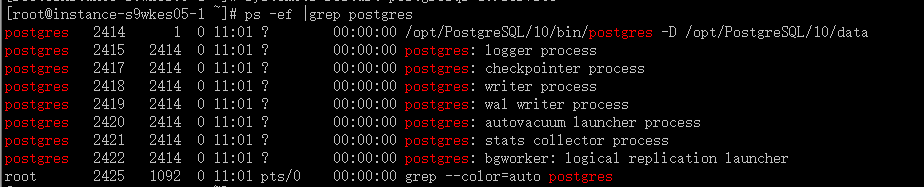


log: 服务日志，记录postgresql运行的信息，由log进程维护

pg\_wal ：预写日志，二进制

pg\_tblsp:自定义表空间的链接

进程结构：



‘

守护进程（有的版本显示的是postmaster），总的来说就是启动数据库的那条命名产生的进程。在本例中 是  2414 。这是数据库服务的起点

守护进程的功能

1 服务的启动和停

2 监听客户端连接，每当一个连接产生，就会fork一个新的postgres 服务连接

3 启动辅助进程（wal checkpointer  loger 等）

...

wal writer ：定期将wal缓冲区的数据写入磁盘（wal预习日志）

stats collectour :统计信息手机进程，提供元数据信息 ，比如pg\_static 表  pg\_stat\_acitivty表的数据就是这个进程在更新

logger process :  将消息和错误信息 写入日志（注意和wal日志的区别）

checkpointer : 检查点，多久刷wal日志到真是数据

autovacuum laucher:自动回收垃圾

对表delete操作时，不会立即删除；更新也不是立即更新，而是新生一条数据，（多版本机制） 空闲 和没有其他业务时删除

writer进程： 将共享内存中的“脏页刷入磁盘”，周期性刷

当往数据库插入和更新时，数据不会马上写到数据文件中；而是在共享内存里；writer周期性把这些数据持久化到磁盘上。周期太快和太慢都有问题

bgworker ：逻辑复制

内存结构

本地内存和共享内存

1 本地内存，是客户端的内存，比如会话级别的临时表

2 共享内存：高频程序的表缓冲、wal日志缓存，进程信息、锁信息，全局统计

一、full-sql +特色sql

with as 语句

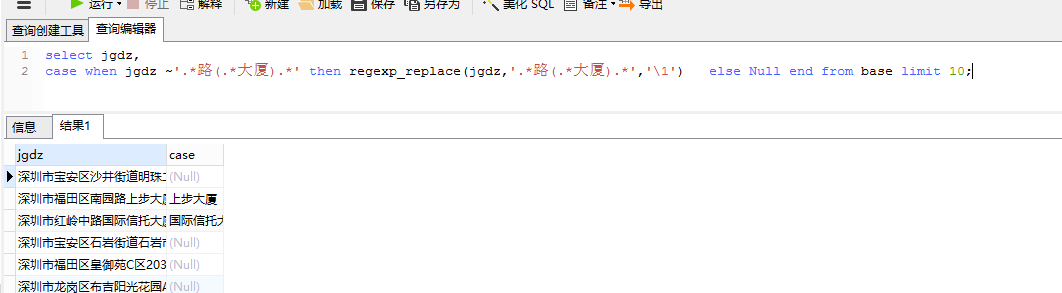
正则

example

select jgdz,

case when jgdz ~'.\*路(.\*大厦).\*' then regexp\_replace(jgdz,'.\*路(.\*大厦).\*','\1')   else Null end from base limit 10;

大厦提取

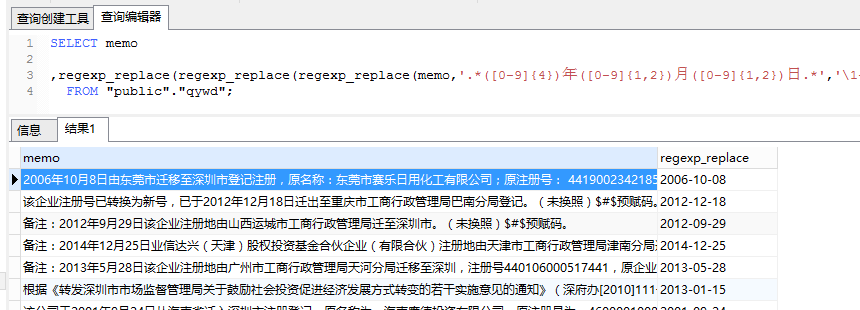


日期提取

SELECT memo

,regexp\_replace(regexp\_replace(regexp\_replace(memo,'.\*([0-9]{4})年([0-9]{1,2})月([0-9]{1,2})日.\*','\1-\2-\3') ,'-([0-9]{1})$','-0\1'),'-([0-9]{1})-','-0\1-')

  FROM "public"."qywd";



substring(string from pattern)函数提供了从字符串中抽取一个匹配 POSIX 正则表达式模式的子字符串的方法。如果没有匹配它返回 NULL ，否则就是文本中匹配模式的那部分。

regexp\_replace(source, pattern, replacement [, flags ])函数提供了将匹配 POSIX 正则表达式模式的子字符串替换为新文本的功能。

regexp\_matches(string, pattern[, flags ])函数返回一个从匹配POSIX正则表达式模式中获取的所有子串结果的text数组。

参数flags是一个可选的text字符串，含有0或者更多单字母标记来改变函数行为。标记g导致查找字符串中的每个匹配，而不仅是第一个，每个匹配返回一行。

regexp\_split\_to\_table(string, pattern[, flags ])函数使用POSIX正则表达式模式作为分隔符，分隔字符串。返回结果为string。。

regexp\_split\_to\_array (string, pattern[, flags ])函数与regexp\_split\_to\_table行为相同，但，返回结果为text数组

distinct on

在主键去重上很有用

filter

窗口函数 row\_number

二、牛逼的数据类型 arrayarray postgres  hstore

array 和  json

no-sql性能支持

三、无限潜力的数据库开发语言（plpythonu 和 plv8）

每个数据库都提供存储过程开发语言，sqlserver的T-SQL  ORACLE的plsql  ,很多年前mysql是不支持变编程的从mysql5开始

postgreql 支持的最彻底,除了自身的psql和pgsql 还支持plpython\java\javsscript\r\tcl

使用plpython3u

pypthon2u和plpython3u，具体要根据编译选项，具体如何编译来的选择；这里选择plpython3u

开发一个接口

create or replace function get\_result(xzqh text,bgdate text,eddate text)

returns text as $$

import json

if xzqh!='all':

    sql="select xzqh,count(\*) as cn from peixun\_exam1   where clrq>='%s' and clrq<'%s' and xzqh='%s'  group by xzqh "%(bgdate,eddate,xzqh)

else:

  sql="select xzqh,count(\*) as cn from peixun\_exam1   where clrq>='%s' and clrq<'%s'   group by xzqh "%(bgdate,eddate)

plpy.notice(sql)

result=plpy.execute(sql)

results=list(result)

a={"code":"200","rows":results}

a=json.dumps(a,ensure\_ascii=False)

return a

$$ language plpython3u;

select get\_result('all','2000-01-01','2018-01-01')

CREATE PROCEDURE generate\_iris\_model (@trained\_model varbinary(max) OUTPUT) AS BEGIN EXEC sp\_execute\_external\_script @language = N'Python', @script = N' import pickle from sklearn.naive\_bayes import GaussianNB GNB = GaussianNB() trained\_model = pickle.dumps(GNB.fit(iris\_data[[0,1,2,3]], iris\_data[[4]].values.ravel())) ' , @input\_data\_1 = N'select "Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width", "SpeciesId" from iris\_data' , @input\_data\_1\_name = N'iris\_data' , @params = N'@trained\_model varbinary(max) OUTPUT' , @trained\_model = @trained\_model OUTPUT; END; GO

<https://pgxn.org/dist/plv8/doc/plv8.html#database-access-via-spi-including-prepared-statements-and-cursors>

使用plv8 编写正则函数

create or replace function test2(x text)

returns text as

$$

if (x==null){return null};

result1=x.match(/[^号路栋]{2,5}大厦/g);

if (result1==null){

        return null;

}

else{

        return result1[0];

}

$$ language plv8;

这些语言是 图林完备的

select test2('深圳市龙岗区龙岗街道南联社区爱南路391号A706-2');

create or replace function get\_result\_plv8(xzqh text)

returns text

as $$

results={}

result=plv8.execute("select jgmc,clrq from peixun\_exam1 where xzqh='"+xzqh +"' limit 10")

plv8.elog(NOTICE,result[1]['jgmc'])

results={"code":"200","rows":result}

results=JSON.stringify(results)

return results

$$ language plv8 ;

select get\_result\_plv8('福田区')

<https://stackoverflow.com/questions/21641722/reusing-pure-python-functions-between-pl-python-functions>  purepython call GD

四、连接一切的外部表(fdw) 中间件特性

把其他数据库、文件系统、网路接口的数据当作自己的表，用自己的语法去查、更新

create server mysql  foreign data wrapper odbc\_fdw

options(dsn 'mysql', encoding 'gbk');

create user mapping for postgres

server mysql

options(odbc\_UID 'root',  odbc\_PWD 'since2015');

import foreign schema mysql

from  server mysql

into fdw

options(

odbc\_DATABASE 'mysql',

table 'mysql',

sql\_query 'select \* from mysql.user'

);