



北京大学
PEKING UNIVERSITY

翰林81
HIGH GO

PG 社区贡献

开源开发实践-第十一周

David & Cary

AGENDA

目录

- 准备工作
- 补丁审查
- 补丁提交
- 博客贡献
- 总结



```
##### OPERATOR CLASSES #####
Mirror Tool

class MirrorX(bpy.types.Operator):
    """This adds an X mirror to the selected object"""
    bl_idname = "object.mirror_mirror_x"
    bl_label = "Mirror X"

    classmethod
    def poll(cls, context):
        return context.active_object is not None

    def execute(self, context):
        # Selection at the end - add back the deselected mirror modifier object
        mirror_ob.select = 1
        modifier_ob.select = 1
        bpy.context.scene.objects.active = modifier_ob
        print("Selected" + str(modifier_ob)) # modifier ob is the active ob
        mirror_ob.select = 0
        scene = bpy.context.selected_objects[0]
        bpy.data.objects[scene.name].select = 1
    except:
        print("please select exactly two objects, the last one gets the mirror")
```

准备工作

此章节描述了成为一个国际PG社区贡献者的前期准备工作。



PG官方账户注册

参与社区贡献的第一步就是在PostgreSQL官方网站
<https://www.postgresql.org> 注册一个PG账户，我们建议使用
瀚高的邮件地址进行注册，以便社区了解我们公司的贡献。

有了PG官方账户，您可以：

- 社区捐款
- 订阅PG相关的电子邮件列表
- 参与培训活动
- 发布PG博客
- 参与代码审核
- 等等。。。



北京大学
PEKING UNIVERSITY

瀚高
HIGH GO

点击your account注册官方账户



New to PostgreSQL?

PostgreSQL is a powerful, open source object-relational database system with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.

There is a wealth of information to be found describing how to **install** and **use** PostgreSQL through the **official documentation**. The PostgreSQL community provides many helpful places to become familiar with the technology, discover how it works, and find career opportunities. Reach out to the community **here**.

[Read More](#)

Latest Releases

2020-05-21 - PostgreSQL 13 Beta 1 Released!

The PostgreSQL Global Development Group announces that the **first beta release** of PostgreSQL 13 is now **available for download**. This release contains **previews of all features** that will be available in the final release of PostgreSQL 13, though some details of the release could change before then.

In the spirit of the open source PostgreSQL community, we strongly encourage you to **test the new features** of PostgreSQL 13 in your systems to help us eliminate any bugs or other issues that may exist. While we do not advise you to run PostgreSQL 13 Beta 1 in your production environments, we encourage you to find ways to run your

PG电子邮件列表订阅

- 有了PG账户，第一件事情就是先透过这个链接<https://lists.postgresql.org/manage/> 订阅PG的邮件列表。
- PG社区提供了很多不同类别的邮件列表平台给PG爱好者参与讨论。
- 我建议至少订阅以下2个电子邮件列表，以接收最新的技术讨论，新闻和错误报告：
- PG开发群列表 `pgsql-hackers` 是最活跃列表之一，也是这次培训最常用来与社区交流的沟通渠道。

错误提交列表 pgsql-bugs@lists.postgresql.org

PG开发群列表 pgsql-hackers@lists.postgresql.org



北京大学
PEKING UNIVERSITY

清华大学
HIGH GO



PostgreSQL

The world's most advanced
open source database.

Your subscriptions

这里显示所有订阅的邮件列表

You are currently subscribed to the following lists.

cary.huang@highgo.ca	Mail delivery	Actions			
pgsql-admin@lists.postgresql.org	Enabled	View archives	Edit subscription	Unsubscribe	Send test mail
pgsql-announce@lists.postgresql.org	Enabled	View archives	Edit subscription	Unsubscribe	Send test mail
pgsql-bugs@lists.postgresql.org	Enabled	View archives	Edit subscription	Unsubscribe	Send test mail
pgsql-general@lists.postgresql.org	Enabled	View archives	Edit subscription	Unsubscribe	Send test mail
pgsql-hackers@lists.postgresql.org	Enabled	View archives	Edit subscription	Unsubscribe	Send test mail

Subscribe

Note 1: Please ensure you read the [Archive Policy](#) before posting to the lists.

Note 2: Please do not subscribe to mailing lists using e-mail accounts protected by mail-back anti-spam systems. These are extremely annoying to the list maintainers and other members, and you may be automatically unsubscribed.

Subscribe to a mailing list

选择邮件列表

List [pgsql-advocacy \(pgsql-advocacy\)](#)

Address [cary.huang@highgo.ca](#)

[Subscribe](#)

Manage email addresses

点击Subscribe完成订阅

You can register multiple email addresses, for example to use them for different mailing lists. You can also blacklist your own email address from the lists. In this case, if you accidentally send an email from this address or if somebody includes it in an email to the list, it will be denied from posting, as a way to avoid leaking the address to the list.

Email address	Status	Flags	Actions
cary.huang@highgo.ca	Confirmed		Blacklist

Add new address

To add a new email address, fill out the fields below. An email will be sent to the address to verify that you are in control of it.

Address

Confirm address

Add email address

电子邮件签名

- 电子邮件是与社区沟通最常用到的渠道，拥有一个好的电子签名可以让人们更加认识你个人或是你的公司，达到一点点宣传的效果
- 请为您自己创建一个电子签名，您可以使用右边的样例或是自己设计的签名模板
- 建议使用英文的名字以及职位



北京大学
PEKING UNIVERSITY

瀚高
HIGH GO

模板：

[英文名字] [姓氏]

[英文职位]

[公司名（如果适用）]

[电子邮件地址]

[公司/个人网站]

样例：

Cary Huang

Senior Software Developer
HighGo Software Inc. (Canada)
cary.huang@highgo.ca
www.highgo.ca



了解您的专业知识及贡献方向

PostgreSQL是一个庞大的数据库系统，极为少数人理解PG的每个组件及围绕它构建的系统。
在做出贡献之前，我建议先根据您的专业知识来缩小组件的范围

以下是PostgreSQL讨论最多的组件列表，选择2到3个组件研究对于一个新的社区贡献者是很好的起点。

- | | |
|---|---------------------------|
| • Security | 安全 |
| • Logical & physical replication | 逻辑&物理数据复制 |
| • Storage Optimization | 存储优化 |
| • Backup and recovery | 备份和恢复 |
| • Query processing (parser, analyzer, optimizer, planner, executor) | 语句处理（解析器，分析器，计划器，优化器，执行器） |
| • Table space | 表空间 |
| • Stored procedure and stored function | 存储提交和函数 |
| • Master and standby architecture | 主备架构 |
| • Scaling and high availability | 拓展及高可用 |
| • FDW machinery | 外部数据包装器 |



了解您的专业知识及贡献方向 - Cont.

- Postgres FDW Postgres自带的外部数据包装器插件
- Horizontal scaling / sharding 水平拓展/分片
- Table partitioning 表分区
- Postgres build and cross platform support 构建和跨平台支持
- Connection pooling 连接池
- Connector support 连接器支持
- PG tools (pgadmin, database migration tool) PG工具 (pgadmin, 数据库迁移工具)
- Vacuum 数据真空
- Write ahead log (WAL) WAL日志
- Optimizer and executor 优化器和执行器
- TOAST table Toast表
- Performance tuning 数据库性能调优
- Indexing and search algorithm 索引及搜索算法
- Data type support 数据类型支持
- Database snapshot 数据库快照
- Transaction management 数据库事务管理



```

    #deselection at the end - add back the deselected mirror modifier object
    mirror_ob.select= 1
    modifier_ob.select=1
    bpy.context.scene.objects.active = modifier_ob
    print("Selected" + str(modifier_ob)) # modifier ob is the active ob
    #mirror_ob.select = 0
    $one = bpy.context.selected_objects[0]
    #bpy.data.objects[one.name].select = 1
except:
    print("please select exactly two objects, the last one gets the mirror")

----- OPERATOR CLASSES -----
#Mirror Tool

class MirrorX(bpy.types.Operator):
    """This adds an X mirror to the selected object"""
    bl_idname = "object.mirror_mirror_x"
    bl_label = "Mirror X"

    @classmethod
    def poll(cls, context):
        return context.active_object is not None
```

补丁审查

此章节描述了如何参与国际社区补丁审查





利用commitfest 平台做补丁审查

- PostgreSQL官方的补丁审查都是在commitfest网站 (<https://commitfest.postgresql.org/>) 维护的。这其实是一个PG开发群邮件列表 (pgsql-hackers) 邮件的集合。
- Commitfest 上的每一条记录其实就是一个链接到某一个含有补丁附件的开发群邮件线程
- Commitfest 上的每一条记录还带有状态信息，显示某补丁进入官方PostgreSQL源代码的状态。
- 在这里，我们可以看到其他贡献者提交的补丁以及过去的讨论并参与审核

Home / Commitfest 2020-07 / Fix false "ERROR: subtransaction logged without previous top-level txn record" alert / Log in

Fix false "ERROR: subtransaction logged without previous top-level txn record" alert 选择评论或反馈

Edit Comment/Review Change Status

Title	Fix false "ERROR: subtransaction logged without previous top-level txn record" alert	补丁概述
Topic	Bug Fixes	补丁类型
Created	2019-12-17 15:02:06	提交日期
Last modified	2020-04-08 15:21:04 (2 months ago)	
Latest email	2020-03-04 13:29:44 (3 months, 1 week ago)	
Status	2020-07: Needs review 2020-03: Moved to next CF 2020-01: Moved to next CF	补丁审核状态
Target version		点此注册成为审核员
Authors	Arseny Sher (sher-ars)	作者
Reviewers		Become reviewer
Committer		
Links		
Emails	ERROR: subtransaction logged without previous top-level txn record First at 2019-06-10 21:08:46 by "Hsu, John" <hsuchen@amazon.com> Latest at 2020-03-04 13:29:44 by Arseny Sher <a.sher@postgresspro.ru> Latest attachment (0002-Stop-demanding-that-top-xact-must-be-seen-before-sub-HEAD_12_patch) at 2020-02-12 08:12:23 from Amit Kapila <amit.kapila16@gmail.com>	链接到开发群邮件
History		补丁下载

When	Who	What
2020-04-08 15:21:04	David Steele (dsteele)	Closed in commitfest 2020-03 with status: Moved to next CF
2020-04-04 02:41:22	Álvaro Herrera (alvherre)	Changed topic to Bug Fixes
2020-02-01 12:54:36	Tomas Vondra (fuzzyc)	Closed in commitfest 2020-01 with status: Moved to next CF
2020-01-30 10:13:23	Maurizio Sambali (maurizios)	Posted comment with messageid <158037920370.742.6543064318169114591.pgcf@coridan.postgresql.org>
2019-12-17 15:07:21	Arseny Sher (sher-ars)	Changed authors to Arseny Sher (sher-ars)

Edit Comment/Review Change Status

历史记录



补丁选择指南

所有提交的补丁又分为下面几类：

- | | |
|--------------------------|-------|
| • Bug fixes | Bug修复 |
| • Clients | 客户端 |
| • Code comments | 代码注释 |
| • Documentation | 文档 |
| • Miscellaneous | 杂项 |
| • Monitoring & Control | 监控 |
| • Performance | 性能 |
| • Procedural Languages | 程序语言 |
| • Refactoring | 重构 |
| • Replication & Recovery | 复制和恢复 |
| • Security | 安全性 |
| • Server Features | 服务器功能 |
| • SQL commands | SQL命令 |
| • System Administration | 系统管理 |

- 作为新的贡献者，您可以从客户端、代码注释、文档、系统管理和SQL命令类别中的补丁开始，
- 因为这些类别解决的问题更多的是关于前端工具而不是核心逻辑。这些类别的补丁作为新的贡献者相对容易。
- 随着您对Postgres以及后端的工作方式越来越有经验，您可以开始查看这些类别的补丁。



补丁选择指南 - Cont.

- 要查看的另一个参数是“Num cfs”，这代表这个补丁经历了多少轮的commitfest。每一轮commitfest会持续一个月的时间
- 如果Num cfs太大，则意味着该补丁已经讨论了很长时间，审核该补丁可能需要花很多时间了解大家讨论的重点。作为新的贡献者，选择“Num cfs”较低的补丁往往会花费较少的时间来审核。
- 您应该选择处于“Needs review”状态的补丁。
- 一个补丁允许多个审核员（reviewers）注册，已经有审核员的补丁您也一样可以参与
- 发布审查之后，别忘了点击“become reviewer”来注册您自己成为正式的补丁审核员。当这个补丁被提交了，审查员的名字也会被写在代码提交的描述里。切记！

Home / Commitfest 2020-07 / Activity log / Log in

Commitfest 2020-07

Search/filter Shortcuts New patch 提交新补丁

Status summary: Needs review: 159. Waiting on Author: 16. Ready for Committer: 12. Committed: 14. Withdrawn: 6. Rejected: 1. Total: 208.

Active patches

Patch	补丁名	Status	Ver	作者	审核员	提交者	Num cfs	Latest activity	Latest mail
Bug Fixes									
COPY FREEZE, PD_VISIBLE_ALL and VM status		Needs review		Pavan Deolasee (pavan), Ibrar Ahmed (ibrar)	Darafei Praliaskouski (komzpa)		3	2020-05-10 21:54	2020-03-25 14:37
Spurious "apparent wraparound" via SimpleLruTruncate() rounding (data loss)		Needs review	stable	Noah Misch (nmisch)			7	2020-04-08 15:19	2020-05-25 07:00
Fix unique join costings		Needs review	stable	David Rowley (davidrowley)	Tom Lane (tgl)		7	2020-04-08 15:19	2020-04-24 06:26
standby recovery fails when re-replaying due to missing directory which was removed in previous replay.		Waiting on Author	stable	Kyotaro Horiguchi (horiguti), Paul Guo (paulguo)	Fuji Masao (masaofuji), Anastasia Lubennikova (lubennikovaav)		6	2020-04-08 13:00	2020-03-25 05:52
pg_upgrade fails with non-standard ACL		Needs review		Anastasia Lubennikova (lubennikovaav), Arthur Zakirov (a.zakirov)	Grigory Smolkin (g.smolkin.postgrespro.ru)		5	2020-04-08 15:20	2020-06-12 07:08
Invalid permission check in pg_stats for functional indexes		Waiting on Author	13	Pierre Ducroquet (pinararf)	Kuntal Ghosh (kuntal_ghosh)		5	2020-04-08 13:01	2020-03-25 14:52
Fix base backup checksum verification for random or zero page headers		Waiting on Author		Michael Banck (mbanck)			4	2020-04-08 13:04	2020-04-06 21:15
Report all I/O errors in buffile.c		Needs review		Thomas Munro (macdice)	Ibrar Ahmed (ibrar)		3	2020-04-07 01:51	2020-06-09 02:32
Fix false "ERROR: subtransaction logged without previous top-level txn record" alert		Needs review		Arseny Sher (sher-ars)			3	2020-04-08 15:21	2020-03-04 13:29
Fix to_char() behaviour with 'WW-D' pattern		Needs review		Mark Lorenz (s-man42)	Cleysson Lima (cleyssonlima), Nitin Jadhav (nitinjadhav11)		3	2020-04-08 15:21	2020-04-04 16:39
DROP OWNED CASCADE vs Temp tables		Waiting on Author	stable	Álvaro Herrera (alherre)	Tom Lane (tgl), ahsan hadi (ahsan.hadi)		2	2020-04-08 14:31	2020-05-06 17:02
more ALTER ... DEPENDS ON EXTENSION fixes		Waiting on Author		Álvaro Herrera (alherre)	Ibrar Ahmed (ibrar), ahsan hadi (ahsan.hadi)		2	2020-04-08 13:06	2020-04-20 17:55
Fix PGXS so built control files work with VPATH builds		Needs review		Craig Ringer (ringerc)			2	2020-05-26 02:15	2020-04-09 03:54



如何透过commitfest审查补丁

- 一旦选择了一个补丁，您需要把他下载然后把它打进PostgreSQL的Master或某个版本分支，然后做一轮测试，也可以审查代码并提供意见。
- 准备好您的反馈后就可以提交了。您可以选择comment（评论）或是review（反馈），两者都要求您提供简短的评论。对于反馈选项，您将需要提供regression测试结果。
- 提交评论时，除了您的评论外，还需要填写以下内容：
 - Make installcheck-world 回归测试
 - Implements feature 实现功能
 - Spec compliant 符合规格
 - Documentation 文档
- 提交了以后，您的反馈将自动发给订阅[pgsql-hackers](https://www.postgresql.org/lists/pgsql-hackers)邮件列表里所有人。

Home / Commitfest 2020-07 / View patch / Add review

/ Logged in as cary (edit profile | log out)

Add review

Note! This form will generate an email to the public mailinglist pgsql-hackers@lists.postgresql.org, with sender set to cary.huang@highgo.ca! Please ensure that the email settings for your domain (DKIM, SPF) allow emails from external sources.

In response to:

From Ibrar Ahmed <ibrar.ahmad@gmail.com> at 2020-03-25T14:37:22

make installcheck-world:

☐ Tested ☐ Passed

回归测试?

Implements feature:

☐ Tested ☐ Passed

实现功能?

Spec compliant:

☐ Tested ☐ Passed

符合规格?

Documentation:

☐ Tested ☐ Passed

文档?

Message:

评论写在这里

New status:

☐ Needs review

不要改动状态

提交

审查commitfest之外的补丁

- Commitfest上其实并未列出所有提交给社区的补丁。实际上，大多数补丁首先会发送到[pgsql-hackers](https://www.postgresql.org/list/pgsql-hackers/)电子邮件列表来直接进行社区讨论，并在后期阶段创建一个commitfest条目。
- 所以，我们也可以通过浏览[pgsql-hackers](https://www.postgresql.org/list/pgsql-hackers/)电子邮件封存（<https://www.postgresql.org/list/pgsql-hackers/>）并搜索“patch”，“poc”，“wip”，“proposal”等关键字来找到合适的补丁进行审查
- 请注意，以这种方式查看补丁并没有像commitfest页面中那样的GUI界面，因此，多数情况下，审查是通过简单地写回给作者的电子邮件说明您对他/她的补丁的看法和观点来完成的。



北京大学
PEKING UNIVERSITY

清华
HIGH GO

pgsql-hackers

The PostgreSQL developers team lives here. Discussion of current development issues, problems and bugs, and proposed new features. If your question cannot be answered by people in the other lists, and it is likely that only a developer will know the answer, you may re-post your question in this list. You must try elsewhere first!

Search the Archives

在这里放关键字
搜索邮件封存

(enter a message-id to go directly to that message)

Advanced Search

View Archives

2020		
June 2020	View Archives	Download mbox
May 2020	View Archives	Download mbox
April 2020	View Archives	Download mbox
March 2020	View Archives	Download mbox
February 2020	View Archives	Download mbox
January 2020	View Archives	Download mbox
2019		
December 2019	View Archives	Download mbox
November 2019	View Archives	Download mbox
October 2019	View Archives	Download mbox
September 2019	View Archives	Download mbox
August 2019	View Archives	Download mbox
July 2019	View Archives	Download mbox
June 2019	View Archives	Download mbox
May 2019	View Archives	Download mbox
April 2019	View Archives	Download mbox
March 2019	View Archives	Download mbox
February 2019	View Archives	Download mbox
January 2019	View Archives	Download mbox
2018		
December 2018	View Archives	Download mbox
November 2018	View Archives	Download mbox

或是按月
慢慢浏览



```
        #deselection at the end - add back the deselected mirror modifier object
        mirror_ob.select= 1
        modifier_ob.select=1
        bpy.context.scene.objects.active = modifier_ob
        print("Selected" + str(modifier_ob)) # modifier ob is the active ob
        #mirror_ob.select = 0
        $one = bpy.context.selected_objects[0]
        #bpy.data.objects[$one.name].select = 1
    except:
        print("please select exactly two objects, the last one gets the mirror")

----- OPERATOR CLASSES -----
#Mirror Tool

class MirrorX(bpy.types.Operator):
    """This adds an X mirror to the selected object"""
    bl_idname = "object.mirror_mirror_x"
    bl_label = "Mirror X"

    @classmethod
    def poll(cls, context):
        return context.active_object is not None
```

补丁提交

此章节描述了如何透过补丁提交进行贡献





发现问题并修复

- 贡献错误补丁贡献的过程中**最具挑战性**的部分可能是找到一个可以马上修复的错误。
- 一种方法是浏览pgsql-bugs封存 (<https://www.postgresql.org/list/pgsql-bugs/>), 查看人们正在讨论的问题类型, 可能有一些可以解决的问题。讨论中的主题很可能不是很直接了当, 但还是值得一看。
- 如果在PostgreSQL文档页面中发现错字或描述错误, 你也可以提供补丁来修复文档。
- 除了**错误修复**补丁, 你也可以提交**新功能**补丁
- 浏览Postgres Wiki上的TODO页面: <https://wiki.postgresql.org/wiki/TODO>。此页面包含已发布一段时间的待办事项列表。某些待办事项功能仍未实现。您可能会找到一些可以轻松完成的功能, 并提供补丁给社区



补丁生成及提交

- 一旦找到了要修复的错误，建议在官方的Postgres源代码的master分支上修复此错误
<https://github.com/postgres/postgres/>
- 生成错误修复补丁可以使用“git diff”命令生成补丁文件。补丁文件名建议使用以下的格式
[版本]_[补丁序号]_[补]-[丁]-[的]-[描]-[述]. patch

例如：

- `git diff file1.c file2.c file3.c > v1_0001_uri-encoding-improvement.patch`

如果你的改动比较大，你可以考虑分多个补丁提交，以方便社区审核

例如：

- `git diff file1.c file2.c file3.c > v1_0001_add-key-management-system.patch`
- `git diff file4.c file5.c file6.c > v1_0002_documentation.patch`
- `git diff file7.c file8.c file9.c > v1_0003_regression-test.patch`






补丁生成及提交 - Cont.

- 准备好错误补丁了以后，建议找另一为软件背景的人员帮你**审核该补丁**，以获得一些反馈。
David和Gary可以帮你们做审核。
- 改进以后，您就可以撰写一个新的电子邮件，其中包含您的补丁的说明。例如：
 - 它修复了什么？
 - 如何验证此修复？
 - 修复是基于什么概念或是测试？
 - 为什么需要这个补丁？
 - 这个补丁会不会影响PostgreSQL的正常运作？
- 邮件准备好了以后，就可以把它发送到PG开发群（pgsql-hackers@lists.postgresql.org）来开启一个新的社区讨论。
- 参与社区讨论通常是以**英文**来做交流

提交补丁到Commitfest

- 经过几轮讨论，您会开始感觉社区是否对此补丁感到兴趣。社区很有可能会提出意见且要求您修改你的补丁。
- 当您的补丁变得更加可靠时，社区中的一些高级成员可能会要求您将补丁发布在[commitFest](https://www.postgresql.org/commitfest/)上来吸引其他社区成员做下一步的审查。如果没有，您也可以询问你的补丁合不合适发布在Commitfest上
- 发布到commitfest上之前，您必须去[pgsql-hackers](https://www.postgresql.org/list/pgsql-hackers/)电子邮件封存(<https://www.postgresql.org/list/pgsql-hackers/>)，搜索你发起的电子邮件并拷贝它的thread id (线程号)
- 然后再回到commitfest的首页，点击“new patch”来提交一个新的commitfest记录



21st May 2020: PostgreSQL 13 Beta 1 Released!

Search term:

List: 邮件列表选pgsql-hackers

Date: 用PG提供的搜索过滤器来找到您的邮件

Sort By:

Results 21-40 of 41.

Result pages: [Prev](#) [1](#) [2](#) [3](#) [Next](#)

21. Include sequence relation support in logical replication [0.10]
From Cary Huang <cary.huang@highgo.com> on 2020-03-24 16:19:21
From the PG logical replication documentation, I see that there is a listed limitation
<https://www.postgresql.org/message-id/1710ed7e13b.cd7177461430746.3372264562543607781@highgo.ca>

22. Re: Include sequence relation support in logical replication [0.20]
From Andres Freund <andres@anarazel.de> on 2020-03-25T00:44:39.
On 2020-03-24 16:19:21 -0700, Cary Huang wrote: > Hi > > > From the PG
<https://www.postgresql.org/message-id/20200325004439.wa6s4eg54qm3qyfs@alap3.anarazel.de>

Home / Commitfest 2020-07 / New patch / Logged in as cary (edit profile | log out)

New patch

Description:

Topic: 选择一个适合的归类 线程号贴到这

Specify thread message id:
 Find thread

Create patch

点击这里提交

编码标准及其他资源



北京大学
PEKING UNIVERSITY

翰林
HIGH GO

- PostgreSQL有严格的编码准则，每个开发人员都应遵
<https://www.postgresql.org/docs/current/source.html>

- 开发人员常见问题解
https://wiki.postgresql.org/wiki/Developer_FAQ

- PG测试
<https://www.postgresql.org/developer/testing/>

- PG 编码
<https://www.postgresql.org/developer/coding/>

- PG开发人员文档
<https://www.postgresql.org/docs/devel/>

- 提交补丁
https://wiki.postgresql.org/wiki/Submitting_a_Patch

- 编码准则
<https://www.postgresql.org/docs/devel/source-format.html>

- 服务器上的错误报告
<https://www.postgresql.org/docs/devel/error-message-reporting.html>
<https://www.postgresql.org/docs/devel/source-conventions.html>



北京大学
PEKING UNIVERSITY

瀚高
HIGH GO

补丁审核和 贡献演示



PostgreSQL



```
        #deselection at the end - add back the deselected mirror modifier object
        mirror_ob.select= 1
        modifier_ob.select=1
        bpy.context.scene.objects.active = modifier_ob
        print("Selected" + str(modifier_ob)) # modifier ob is the active ob
        #mirror_ob.select = 0
        #one = bpy.context.selected_objects[0]
        #bpy.data.objects[one.name].select = 1
    except:
        print("please select exactly two objects, the last one gets the mirror")

----- OPERATOR CLASSES -----
#Mirror Tool

class MirrorX(bpy.types.Operator):
    """This adds an X mirror to the selected object"""
    bl_idname = "object.mirror_mirror_x"
    bl_label = "Mirror X"

    @classmethod
    def poll(cls, context):
        return context.active_object is not None
```

博客贡献

此章节描述了如何透过博客发布来对国际社区进行贡献





北京大学
PEKING UNIVERSITY

瀚高
HIGH GO

个人博客网站 样例



PostgreSQL

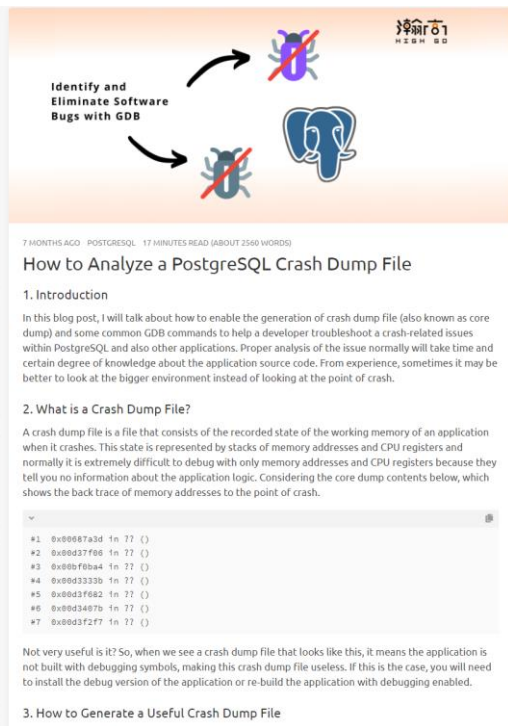
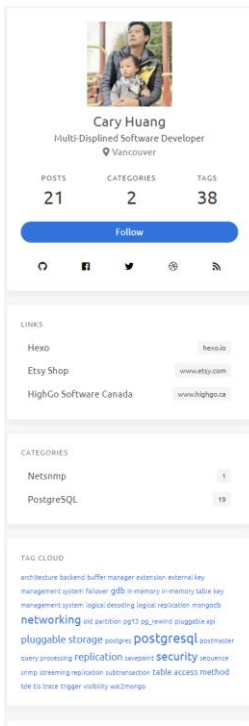


发布PostgreSQL博客规定

- 博客应以PostgreSQL社区账户的名义提交
- 与PostgreSQL有关的所有博客都非常感激-技术性和非技术性
- 所有帖子均应为英文。
- 所有博客均应由发布者名称标识。昵称可以与此结合使用。没有名字的博客是不允许的
- 禁止在帖子中发表煽动性言论。不允许进行人身攻击
- 禁止在博客发布广告
- 提交的博客在发布前会被PG社区审核

博客写什么？

- 你可以写关于PG的任何事情，包含PG操作教程，工作原理，你目前PG的工作心得。
- 或是你可以把博客当成你的笔记，学到了什么新知识就可以用博客的形式把它记录下来（像是gdb，加密，或密钥管理理论基础等等。。。）
- 可以不需要把一个博客写的很专业或死板，你可以用第一人称视角和你平时说话的语气去描述一个事情
- 利用博客网站慢慢建立你的个人声誉



知识版权



北京大学
PEKING UNIVERSITY

翰林
HIGH GO

- 如果你的博客引用到了其他博客，文章，论文，或是图片，切记一定要在博客里清楚的表示来源
- 博客网站是公开的。当你注册进PG社区的博客系统，世界各地的人都很有机会看到你的发文来进一步认识你。
- 你不想给大家一个印象就是你的博客有抄袭的嫌疑
- 切记，要清楚表示资料来源





架设个人博客网站

- 目前市面上有很多可以架设个人博客的平台，免费以及付费的都有
- 个人博客平台需要支持RSS atom feed 的功能，这个feedURL链接需要被注册到PG社区里面的
- 你可以使用wordpress
- 或是使用github自带的免费的web hosting + hexo 静态页面生成工具
- 或是任何国内主流的博客平台
- PG博客通常都是用一种叫做markdown（.md）的格式编写的
<https://www.markdownguide.org/cheat-sheet/>



Wordpress博客发布

- 使用wordpress来做个人博客网站相对容易很多。你不需要懂很多web development的知识。但要收费。。。
- 在管理页面左方的选项选择新增一个新的博客（Posts）并给与它一个吸引注意的标题。
- 然后就可以使用wordpress自带的编辑工具撰写一篇有意义的博客

WordPress 5.4.2 is available! Please notify the site administrator.

Dashboard

Click posts then add new to create a new blog

MonsterInsights Overview Report

Sessions	Pageviews
2995	3624
↑ 100% vs. Previous 30 Days	↑ 100% vs. Previous 30 Days

Avg. Session ...	Bounce Rate
46s	88.31%
↑ 100% vs. Previous 30 Days	↑ 100% vs. Previous 30 Days

Pro Tip: Get Page Insights to see important metrics for individual posts / pages in WordPress. [Upgrade to Pro »](#)

Top Posts/Pages

New vs. Returning Visitors

Device Breakdown

See All Reports [Go to the Analytics Dashboard](#)

Most Viewed Posts

Hits	Post Title
11550 hits	The difference in five modes in the AES encryption algorithm
5389 hits	Streaming Replication Setup in PG12 – How to Do it Right
5074 hits	Stored Procedures in PG 11 – Better late than never
3804 hits	Optimizing SQL – Step 1: EXPLAIN in PostgreSQL – Part 1
3218 hits	The Transparent data encryption in PostgreSQL
3186 hits	Have An Eye On Locks Of PostgreSQL
3182 hits	Can you gain performance with Pgpool-II as a load balancer?
2951 hits	Logical Replication Between PostgreSQL and MongoDB
2700 hits	The performance test on the AES modes
2582 hits	Replication Failover with pg_rewind in PG12

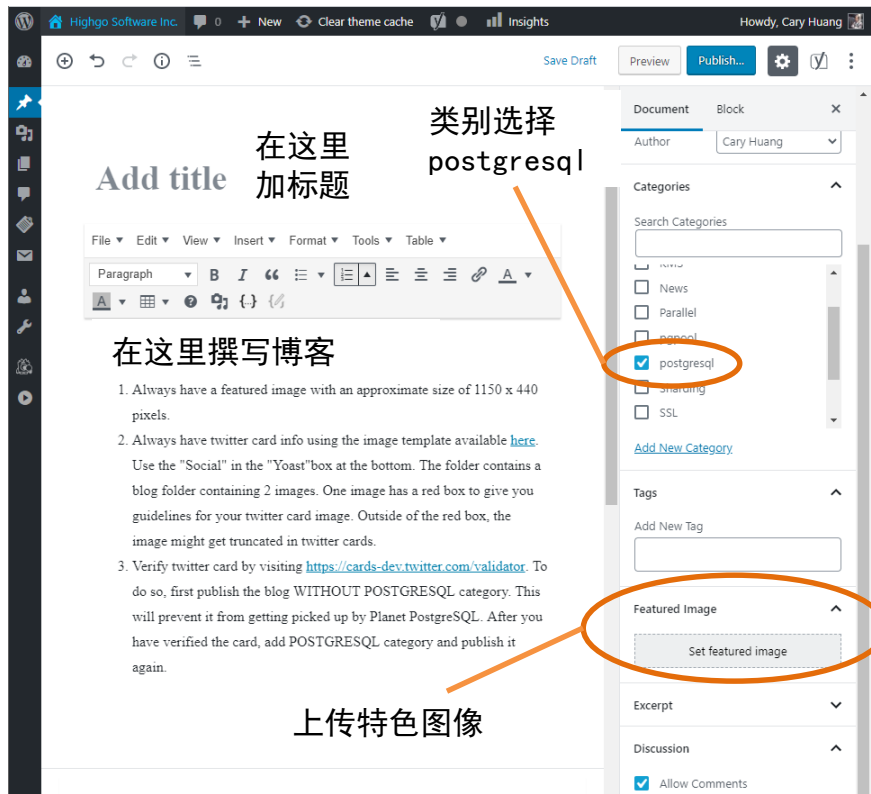
Wordpress博客发布 - cont.



北京大学
PEKING UNIVERSITY

清华
HIGH GO

- 博客类别请选择postgresql
- 在右侧选项栏上, 有一个添加**特色图像** (featured image) 的选项, 请务必上传一个和你的博客内容相关的图像。
- 你可以使用canva.com 来做一个简单的特色图像





北京大学
PEKING UNIVERSITY

瀚高
HIGH GO

架设免费github space博客网站



PostgreSQL

Planet PostgreSQL博客网站



北京大学
PEKING UNIVERSITY

翰林
HIGH GO

- PG官方博客分享网站叫做 Planet PostgreSQL。可以透过此链接访问 (<https://planet.postgresql.org/>)
- 这个网站利用个人或企业的博客网站的 feedURL 来自动抓取您在发布的帖子。所以只要在 Planet PostgreSQL注册了feedURL之后，您在您的博客页面发布的帖子都会自动出现在 Planet PostgreSQL.

注册feedURL



北京大学
PEKING UNIVERSITY

清华
HIGH GO

在Planet PostgreSQL 网站点击右下角的 add

The screenshot shows the Planet PostgreSQL website. The main content area features several articles, including "How to use AdaBoost Machine Learning model with ZUDA - PostgreSQL and Orange (Part 6)", "Repartitioning with logical replication in PostgreSQL 13", "Control Runaway Postgres Queries With Statement Timeout", "Controlling Connection Parameters Using Environment Variables", and "Composite type performance issues in PostgreSQL". The sidebar on the right contains sections for "Top posters", "Top teams", "Feeds", "Planet", and "Contact". The "Planet" section is highlighted with an orange circle, showing a link to "Policy for being listed on Planet PostgreSQL" and a link to "Add your blog to Planet PostgreSQL".

planet.postgresql.org

Apps simply design inc. cornex highgo trading supplier services shokunin design co. amazon stores

PostgreSQL The world's most enhanced open source database

How to use AdaBoost Machine Learning model with ZUDA - PostgreSQL and Orange (Part 6)

Posted by [Benoit Dierker](#) in [Data Science](#) on 2020-09-10 at 17:41

This article gives a step by step guide to utilizing Machine Learning capabilities with ZUDA. In this article, we will use examples of Animals to predict whether they are Mammals, Birds, Fish or Insects. Software versions We are going to use ZUDA version 11.6-1 to implement the Machine Learning model. This version 11.6-1 combines: PostgreSQL [...]

Repartitioning with logical replication in PostgreSQL 13

Posted by [Stefan Posaty](#) in [Database](#) on 2020-09-10 at 17:41


So, you have a partitioned table. And you want to change your mind. Re-partitioning is "easy" if you can take downtime -- just create a new table with a new name and copy all the data over. But what if we want to try to do it without downtime? Logical replication enhancements in PostgreSQL 13 brings us some new options for this!

But first a disclaimer -- this is definitely not pretty! And does not take into consideration things like foreign keys and similar. But sometimes a quick hack can be the best hack.

So let's go!

Control Runaway Postgres Queries With Statement Timeout

Posted by [Core Academic](#) in [Crunchy Data](#) on 2020-09-10 at 16:08



Most queries against a database are short lived. Whether you're inserting a new record or querying for a list of upcoming tasks for a user, you're not typically aggregating millions of records or sending back thousands of rows to the end user. A typical short lived query in Postgres can easily be accomplished in a few milliseconds or less. For the typical application, this means a [well tuned production PostgreSQL](#) database is capable of easily running thousands or up to hundreds of thousands of queries per second on a beefy instance.

Controlling Connection Parameters Using Environment Variables

Posted by [Benoit Dierker](#) in [EnterpriseDB](#) on 2020-09-10 at 17:41

[libpq](#) is the PostgreSQL connection library used by almost every non-JDBC application. It allows many connection parameters, which can be specified on the command line or embedded in applications:

```
$ psql -h myhost.com -d mydb
```

[Continue Reading >](#)

Composite type performance issues in PostgreSQL

Posted by [Benoit Dierker](#) in [Database](#) on 2020-09-10 at 17:41

PostgreSQL is a really powerful database and offers many features to make SQL even more powerful. One of these impressive things is the concept of a composite data type. In PostgreSQL a column can be a fairly complex thing. This is especially important if you want to work with server side stored procedures or functions. However, there are some details people are usually not aware of when making use of stored procedures and composite types.

Top posters

Number of posts in the past month

- [Bruce Momjian](#) (EnterpriseDB) - 24
- [Andreas Jähr](#) (Scherbaum) - 9
- [Hans-Juergen Schoenig](#) (Cybertec) - 7
- [Luca Ferrari](#) - 6
- [Dennis Patel](#) - 5
- [Hubert 'doppe' Lubaczewski](#) - 4
- [Stefan Posaty](#) (Crunchy Data) - 4
- [Paul Ramsey](#) (Crunchy Data) - 4
- [Pavel Stehule](#) - 4
- [Laurent Abbe](#) (Cybertec) - 4

Top teams

Number of posts in the past month

- [EnterpriseDB](#) - 32
- [Cybertec](#) - 15
- [Crunchy Data](#) - 14
- [ZooQuadrant](#) - 14
- [Hypnos Software](#) - 11
- [Percona](#) - 4
- [Seralinius](#) - 2
- [PostGIS](#) - 2
- [Crunchy Data](#) - 2
- [Datto](#) - 2

Feeds

- [All feeds \(full entries\)](#)
- [All feeds \(short entries\)](#)

Planet

[Policy for being listed on Planet PostgreSQL.](#)
[Add your blog to Planet PostgreSQL.](#)
[List of all subscribed blogs.](#)
[Manage your registration.](#)

Contact

Get in touch with the Planet PostgreSQL administrators at [planet at postgresql.org](#).

注册feedURL - cont.



北京大学
PEKING UNIVERSITY

翰林
HIGH GO

点击 registration 注册



The world's most advanced
open source database.

Adding your blog to Planet PostgreSQL

Adding your blog to Planet PostgreSQL is simple, and we encourage all people who blog about PostgreSQL to do it. Just follow the steps below:

1. Make sure your blog follows the [policy](#).
2. If you don't have one already, [sign up](#) for a PostgreSQL Community Account (same as is used for example for the PostgreSQL [wiki](#)).
3. Go to the [registration](#) website, and add your blog.
4. Wait for the blog to get approved by the [moderators](#).
5. Wait up to 15 minutes for syndication of your blog to begin.

注册feedURL - cont.



北京大学
PEKING UNIVERSITY

翰林
HIGH GO

根据以下方针来填写

	Content内容
Feedurl	这里填上你得feedURL https://caryhuang.github.io/atom.xml
Team	如果没有，可以不选
Twitteruser	如果您有一个Twitter用户名，请在此处输入
Authorfilter	可以暂时留空



注册feedURL - cont.

注册完后，您的feedURL会被Planet PostgreSQL审核，完毕后状态应该会是 approved and working。如果不是，请联系Planet 的管理员。Approved 了之后，您以前和后来发布的博客都将会自动出现在Planet PostgreSQL 上。

Your blogs

You have the following blog(s) registered:

Name	Approved	Feed URL/Blog URL	Status	Operations
David Zhang	Yes	Feed: https://www.highgo.ca/author/david.z/feed/?category_name=postgresql Blog: https://www.highgo.ca Last http get: 2020-03-19 17:36:46	Approved and working	<button>Edit</button>

Register blog

Note that your blog will have to be approved before it appears on the planet.

Register new blog

PG Weekly 和 DB Weekly

- PG Weekly (PostgreSQL每周新闻)
 - <https://postgresweekly.com/issues/408> 这里订阅
- DB Weekly (数据库每周新闻)
 - <https://dbweekly.com/> 这里订阅
- 这两个订阅服务每周都会去planet Postgres 里面找寻有趣或是重要的博客文章
- 这意味着你发布到planet Postgres里的博客有机会会被选中并发布在这两个订阅服务里面。
- 被选中对你个人来说是很大的成就啊！
- HighGo每周都会把这两个订阅服务翻译成中文。有兴趣可以在这查看
 - <https://grantzhou.github.io/>



北京大学
PEKING UNIVERSITY

高戈
HIGH GO

数据库每周新闻 2021-5-28

数据库每周新闻#356 - 2021年5月28日

数据库每周新闻

备注：英文原文地址



immudb v1.0: 具有不可变历史记录的“防篡改”数据库

添加了SQL支持，因此创建者现在吹嘘 immudb 是“第一个具有关系功能的不可变数据库”。用 Go 编写，并获得 Apache 许可。

Codenotary Inc

Amazon Redshift ML 现在普遍可用

Redshift 是 Amazon 的云数据仓库，它允许您使用 SQL 查询和组合潜在的 EB（如果您有美元！）的结构化和半结构化数据……并且ML扩展更进一步训练机器学习模型以从数据仓库中的数据中生成见解。更好的是，您可以使用 SQL 完成所有这些工作。

Danilo Poccia (AWS)

比较 Fauna 和 DynamoDB: 功能和定价

Fauna 和 DynamoDB 都是无服务器数据库，但它们的设计目标、架构和用例非常不同。这篇文章讨论了它们在哪里发光，它们在哪里不发光，以及它们如何提供根本不同的价值主张。

Fauna sponsor

Postgres 14 的早期观察: 性能和监控改进

上周我们展示了Postgres 14 beta 1 版本。现在人们开始仔细研究它，看看潜在的改进在哪里。小而稳定且可靠的改进似乎已成为日常工作。

Lukas Fittl



总结

```

    #deselect at the end - add back the deselected mirror modifier object
    mirror_ob.select= 1
    modifier_ob.select=1
    bpy.context.scene.objects.active = modifier_ob
    print("Selected" + str(modifier_ob)) # modifier ob is the active ob
    #mirror_ob.select = 0
    $one = bpy.context.selected_objects[0]
    #bpy.data.objects[one.name].select = 1
except:
    print("please select exactly two objects, the last one gets the modifier unless its not a mod")

----- OPERATOR CLASSES -----
#Mirror Tool

class MirrorX(bpy.types.Operator):
    """This adds an X mirror to the selected object"""
    bl_idname = "object.mirror_mirror_x"
    bl_label = "Mirror X"

    @classmethod
    def poll(cls, context):
        return context.active_object is not None
```



总结



北京大学
PEKING UNIVERSITY

翰林
HIGH GO

- 我们提到了许多可以对PG社区贡献的方法，包含：
 - 博客发布
 - 补丁审查
 - 功能建议
 - 会议讨论
 - 错误修复
- 对于一个新的贡献者来说，与社区中成千上万的人进行交流刚开始可能很艰巨，但是只要您能够坚持自己的立场并进行讨论且慢慢习惯于这种方式，贡献过程将变得更加顺利。
- 沟通的方式往往决定了一个贡献者能不能和社区好好的交流，我在这汇总了一些沟通准则，希望可以在沟通上帮到大家。
- 和社区打交道是一个漫长的旅程，但最终结果肯定会是正面的，加油！



如何有效的和社区成员沟通

- 不要猜

- 社区中的人们技术水平很高。如果您不确定某事，请澄清一下。我相信社区会很高兴帮您澄清。

- 提供证明

- 要在讨论中立足，您需要做好准备。这意味着，您可能需要准备支持材料，例如测试报告或研究论文，以提高您在讨论中的发言的可信度。

- 别害羞

- 参与一个拥有成千上万名成员的社区可能非常令人生畏。如果您有正确的观点，请不要害怕分享。记住，他们不知道你知道什么；他们不知道你不知道什么。所以就大胆去做吧。

- 使用简单语言而不是复杂语言

- 每天都有很多社区电子邮件在来回发布。如果您可以使用简洁明了的语言，这个比长而复杂的邮件更有效地传递信息，因为实际上没有多少人会花时间完全阅读所有的邮件。

- 如果英文是个问题。。。

- 如果英语表达是一个问题，或是您不确定是否以英语足够清楚地传达了信息，你可以随时将您的消息发送到cary.huang@highgo.ca来帮助您进行修改及审核。

瀚高
HIGH GO



融知与行 瀚且高远
THANKS