

崇鹏豪-258430418-项目申请书

项目名称: 基于 uuid-ossp 的 sys_guid 函数开发项目申请书

项目导师: 牛世继<niushiji@highgo.com>

申请人: 崇鹏豪

邮箱: cph13000@gmail.com

日期: 2025.06.05

一、项目背景

- (一) 背景描述
- (二) 项目基本需求
- (三) 项目相关仓库
- (四) 参考资料

二、细节分析

三、任务规划

- (一) 开发阶段 (07 月 01 日 - 08 月 15 日)
- (二) 优化阶段 (08 月 15 日 - 09 月 1 日)
- (三) 交付阶段 (09 月 1 日 - 09 月 30 日)

五、关于我

附件:

项目名称: 基于 uuid-ossp 的 sys_guid 函数开发项目申请书

项目导师: 牛世继<niushiji@highgo.com>

申请人: 崇鹏豪

邮箱: cph13000@gmail.com

一、项目背景

(一) 背景描述

IvorySQL 是基于 PostgreSQL 的开源数据库，旨在支持多种 Oracle 兼容特性。目前，IvorySQL 缺少数据库级别唯一的sys_guid函数，而 PostgreSQL 内核提供的gen_random_uuid()基于伪随机数生成 UUID，无法完全满足 Oracle 兼容性需求。因此，需要基于uuid-ossp或uuid-e2fs库开发符合要求的sys_guid函数，确保唯一性和性能。

(二) 项目基本需求

1. PG内核提供一个类似sys_guid()的函数，即基于伪随机数的gen_random_uuid ()。
2. 当前缺少sys_guid函数。需要基于uuid-ossp/e2fs开发这样一个函数。
3. 通过修改插件 uuid-ossp 的代码 contrib/uuid-ossp/uuid-ossp.c 实现sys_guid()函数。如果系统有uuid-ossp，就使用uuid_make()，否则看有没有uuid-e2fs，如果有的话，就调用 uuid_generate_random()，否则就调用 arc4random()。同时修改代码使得IvorySQL能够自动载入uuid-ossp插件。
4. 新增sys_guid函数，兼容uuid-ossp、uuid-e2fs及内置随机算法。实现插件自动加载，提升用户体验。

(三) 项目相关仓库

- 代码仓库地址: <http://github.com/IvorySQL/IvorySQL>

(四) 参考资料

- 参考资料: <https://docs.ivorysql.org/cn/ivorysql-doc/v4.4/v4.4/6>

二、细节分析

(1) 函数需求设计

修改插件uuid-ossp的代码contrib/uuid-ossp/uuid-ossp.c，实现sys_guid()函数，逻辑如下：

- 优先检测系统是否安装 uuid-ossp 库：

若存在，调用uuid_make()生成 UUID（基于时间或随机算法，具体需结合 Oracle 的 sys_guid 逻辑）。

- 若 uuid-ossp 不可用，检测 uuid-e2fs 库：

若存在，调用uuid_generate_random()生成随机 UUID。

- 若以上库均不可用：

调用 PostgreSQL 内置的arc4random()生成随机数，并拼接为 UUID 格式。

(2) 插件自动加载

修改 IvorySQL 配置，确保uuid-ossp插件在数据库启动时自动载入，避免用户手动干预。具体通过修改postgresql.conf中的shared_preload_libraries参数或插件初始化脚本实现。

(3) 代码实现细节

- 在uuid-ossp.c中新增sys_guid函数定义，参考现有uuid_generate_v*函数的实现逻辑，结合条件编译处理不同库的兼容性。
- 确保 UUID 生成逻辑符合 Oracle sys_guid 的语义（如时间戳 + 随机数组合，或纯随机唯一值），可参考 Oracle 官方文档或社区实现方案。
- 添加单元测试用例，验证函数在不同库环境下的输出唯一性、性能及兼容性。

三、任务规划

(一) 开发阶段 (07 月 01 日 – 08 月 15 日)

1. 阶段一：需求分析与设计

- 深入分析项目需求，明确脚本的功能模块和流程。
- 调研 Oracle sys_guid函数特性，分析 IvorySQL 现有 UUID 机制，确定技术方案。

2. 阶段二：核心功能开发

- 配置uuid-ossp和uuid-e2fs开发环境，熟悉 PostgreSQL 插件开发流程。
- 实现sys_guid函数逻辑，处理多库兼容条件，完成插件自动加载配置。

3. 阶段三：单元测试与验证

- 编写单元测试，验证函数唯一性、性能及异常场景，优化代码逻辑。
- 撰写开发文档，提交 PR 至 IvorySQL 的 GitHub 仓库，配合社区评审修改。

(二) 优化阶段 (08月15日 – 09月1日)

- 根据测试结果和用户反馈，对实现进行优化和改进，提高性能和易用性。
- 持续学习IvorySQL并完善相关文档。

(三) 交付阶段 (09月1日 – 09月30日)

- 将最终的代码和文档提交到项目相关仓库，完成项目交付。
- 提供后续的技术支持，及时解决用户在使用脚本过程中遇到的问题，尽可能帮助社区解决一些问题，尽自己的绵薄之力。

五、关于我

申请人崇鹏豪，熟悉 Bash 脚本语言和常见的 Linux 命令。目前计算机研究生一年级在读，任 OERV 社区前端实习小组组长，有丰富的git使用经验。具备独立完成本项目的能力。相信通过本次项目的开发，参与到IvorySQL 的建设中来，之后能持续为 IvorySQL 贡献自己的力量。

附件：

本人简历： [崇鹏豪-全栈工程师.pdf](#)