

开源秀



让所有人不再为 SQL 问题头疼

全方位 SQL 质量管理平台 —— SQLE

SQLE 开源负责人 孙健



什么是问题SQL?



业务缓慢

业务中断

数据丢失

案例

- 某些查询未加索引，业务加载慢，影响用户体验；
- 业务高峰期、出现高频查询未加索引导致数据库服务器资源占满，整个业务瘫痪；
- 开发者清理测试数据时不小心连到了生产库。

研发 or DBA 背锅？



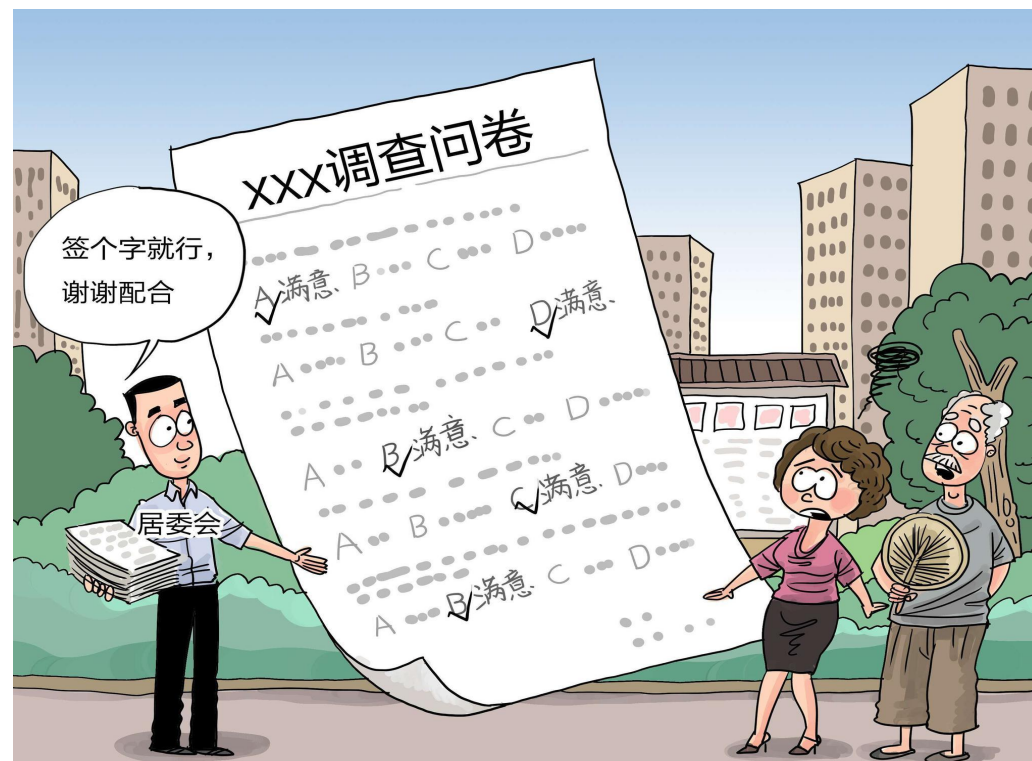
研发会在开发阶段优化SQL吗？



项目进度

开发经验

- 时间紧，赶进度，没时间考虑性能问题优先满足需求；
- 业务量小暂不用考虑性能问题；
- 使用 ORM 框架，未接触实际 SQL。



什么是 SQLE?



SQLE 是一款全方位的 SQL 质量管理平台，覆盖开发至生产环境的 **SQL 审核和管理**。
支持主流的**开源、商业、国产数据库**，为开发和运维提供流程自动化能力，提升上线效率，提高数据质量。



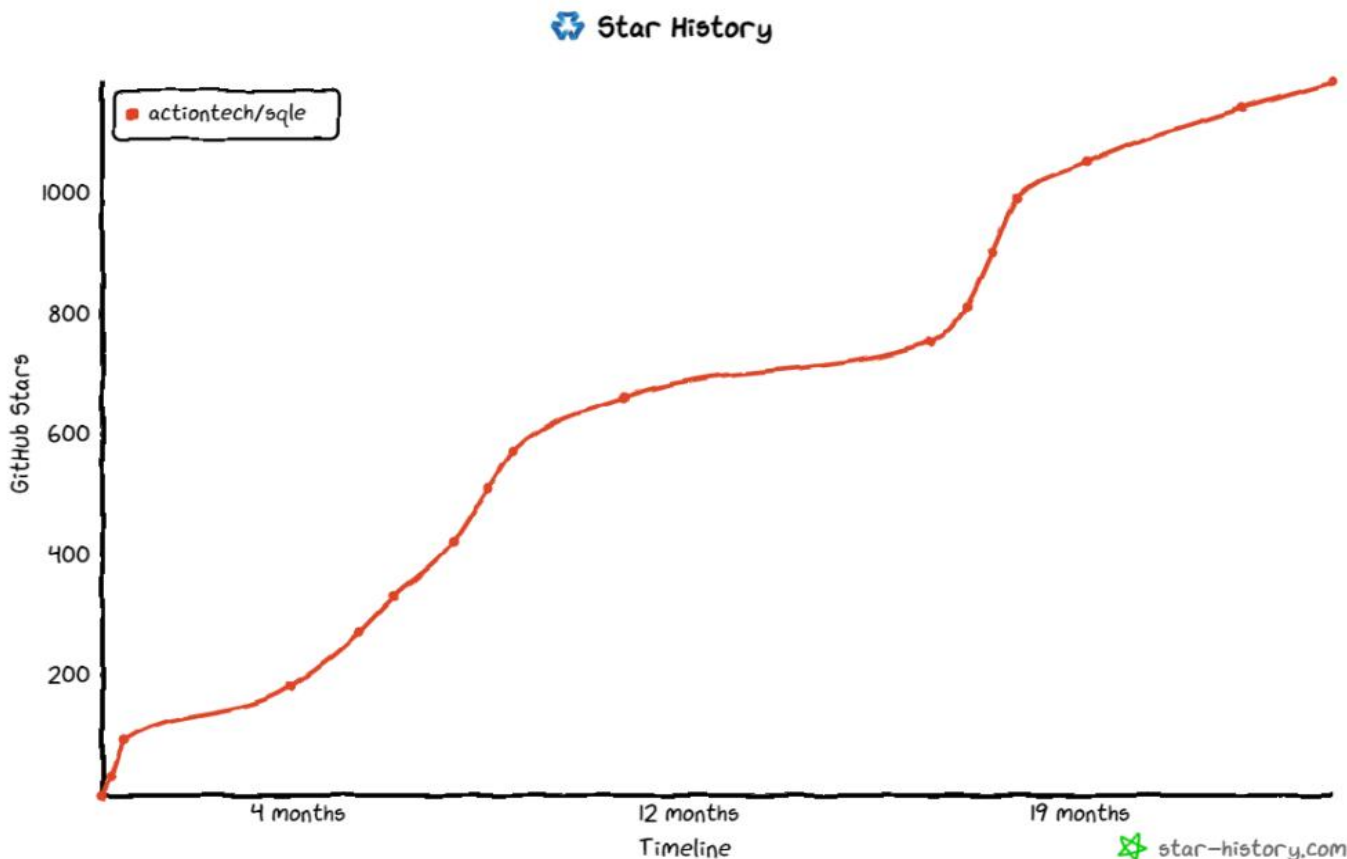
什么是SQLE?



SQLE 于 2021 年 10 月 24 日开源。过去的两年多，SQLE **每月**发布版本：

- 一共发布 **26** 个正式版本
- 新增大功能 **60+**
- 提交 commits **5000+**
- 处理完成 issue&pr **1400+**
- 累计收获 Star **1200+**。

社区多名研发支持。



SQL审核案例



案例：表字符集不一致导致部分场景下索引失效

```
CREATE TABLE `t_utf8` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(20) DEFAULT NULL,  
  `code` varchar(50) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `idx_code` (`code`),  
  KEY `idx_name` (`name`)  
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `t_utf8mb4` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(20) DEFAULT NULL,  
  `code` varchar(50) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `idx_code` (`code`),  
  KEY `idx_name` (`name`)  
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE `t_utf8mb4_2` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(20) DEFAULT NULL,  
  `code` varchar(50) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `idx_code` (`code`),  
  KEY `idx_name` (`name`)  
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4;
```

建三张表，分别使用字符集 **utf8** 和 **utf8mb4**





案例：表字符集不一致导致部分场景下索引失效

```
mysql>
mysql> explain select * from t_utf8 as t1 left join t_utf8mb4 as t2 on t1.code = t2.code where t2.name = 'aaaa_1';
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | t2 | NULL | ref | idx_code,idx_name | idx_name | 83 | const | 8 | 100.00 | NULL |
| 1 | SIMPLE | t1 | NULL | ALL | NULL | NULL | NULL | NULL | 793968 | 100.00 | Using where; Using join buffer (Block Nested Loop) |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)
```

```
mysql>
mysql> explain select * from t_utf8mb4_2 as t1 left join t_utf8mb4 as t2 on t1.code = t2.code where t2.name = 'aaaa_1';
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | t2 | NULL | ref | idx_code,idx_name | idx_name | 83 | const | 8 | 100.00 | Using where |
| 1 | SIMPLE | t1 | NULL | ref | idx_code | idx_code | 203 | test.t2.code | 7 | 100.00 | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)
```

对比执行计划发现当字段的字符集不一致，表关联字段未命中索引

SQL审核案例



案例：表字符集不一致导致部分场景下索引失效

```
mysql> select count(*) from t_utf8 as t1 left join t_utf8mb4 as t2 on t1.code = t2.code where t2.name = 'aaaa_1';
+-----+
| count(*) |
+-----+
|      64 |
+-----+
1 row in set (0.93 sec)

mysql> select count(*) from t_utf8mb4_2 as t1 left join t_utf8mb4 as t2 on t1.code = t2.code where t2.name = 'aaaa_1';
+-----+
| count(*) |
+-----+
|      64 |
+-----+
1 row in set (0.00 sec)
```

每张表插入80万数据，执行时间差异大



SQL审核案例



← 返回工单列表

编辑工单信息 提交工单

测试隐式转换

mysql-1

数据去重 下载

全部等级 普通(Normal) 提示(Notice) 告警(Warning) 错误(Error)

20% 审核通过率 12 审核结果评分 db1 Schema

序号	执行语句	审核结果	说明	操作
1	create table `t_utf8` (`id` int(11) NOT NULL AUTO_INCREMENT, `name` varchar(20) DEFAULT NU...		添加说明	分析
2	create table `t_utf8mb4` (`id` int(11) NOT NULL AUTO_INCREMENT, `name` varchar(20) DEFAUL...		添加说明	分析
3	create table `t_utf8mb4_2` (`id` int(11) NOT NULL AUTO_INCREMENT, `name` varchar(20) DEFA...		添加说明	分析
4	SELECT * FROM t_utf8 as t1 LEFT JOIN t_utf8mb4 as t2 ON t1.code=t2.code WHERE t2.name ...	连接表字段的字符集和排序规则必须一致	添加说明	分析
5	SELECT * FROM t_utf8mb4_2 AS t1 LEFT JOIN t_utf8mb4 AS t2 ON t1.code=t2.code WHERE t...	审核通过	添加说明	分析





mysql_audit_template

项目: default

数据源类型: MySQL

请输入规则关键词搜索

Q

ALL 154

使用建议 16

DML规范 66

DDL规范 39

索引规范 11

命名规范 9

索引优化 3

全局配置 5

索引失效 5

×

连接表字段的字符集和排序规则必须一致

error

连接表字段的字符集和排序规则一致可避免数据不一致和查询错误，确保连接操作正确执行。

×

JOIN字段必须包含索引

error

JOIN字段包含索引可提高连接操作的性能和查询速度。

×

禁止对索引列进行数学运算和使用函数

error

对索引列进行数学运算和使用函数会导致索引失效，从而导致全表扫描，影响查询性能。

×

禁止对联合索引左侧字段进行IN、OR等非等值查询

error

对联合索引左侧字段进行IN、OR等非等值查询会导致联合索引失效

×

使用联合索引时，必须使用联合索引的首字段

error

使用联合索引时，不包含首字段会导致联合索引失效

暂无更多规则

完善规则规范是做好SQL质量管理的第一步



SQL
规则库

Action SQL

default

项目概述

待关注清单

数据源管理

数据源

外部数据源

成员与权限

操作与审计

SQL工作台

SQL审核

规则模板

白名单

流程模板

SQL管控

SQL审核

SQL工单

扫描任务

查看规则

请输入规则关键词搜索

项目 请选择

规则模板 请选择

数据库类型 MySQL

ALL 155

使用建议 16

DDL规范 66

DOL规范 39

索引规范 11

命名规范 9

索引优化 3

全局配置 6

索引失效 5

绑定变量个数不建议超过阈值

error

因为过度使用绑定变量会增加查询的复杂度，从而降低查询性能。过度使用绑定变量还会增加维护成本。默认阈值100

最大绑定变量数: 100

禁止使用没有WHERE条件或者WHERE条件恒为TRUE的SQL

error

SQL缺少WHERE条件在执行时会进行全表扫描产生额外开销，建议在大数据量开发环境下开启，避免影响数据库查询性能

禁止使用event

error

使用event会增加数据库的维护难度和依赖性，并且也会造成安全问题。

禁止使用全文索引

error

全文索引的使用会增加存储开销，并对写操作性能产生一定影响。

禁止使用空间字段和空间索引

error

使用空间字段和空间索引会增加存储需求，对数据库性能造成一定影响

使用TEXT类型的字段建议和原表进行分析，与原表主键单独组成另外一个表进行存放

notice

将TEXT类型的字段与原表主键分拆成另一个表可以提高数据库性能和查询速度，减少不必要的 I/O 操作。

建议为至少一个索引添加非空约束

warn

所有索引字段均未做非空的约束，请确认下该索引规划的有效性。

存在多条对同一个表的修改语句，建议合并成一个ALTER语句

notice

避免多次 TABLE REBUILD 带来的消耗，以及对线上业务的影响

表的初始AUTO_INCREMENT值建议为0

warn

创建表时AUTO_INCREMENT设置为0则自增从1开始，可以避免数据空洞。例如在导出数据库DOL时，表结构内AUTO_INCREMENT通常未设置，如果建表时没有把AUTO_INCREMENT设置为0，那么通过DOL进行建表操作会导致自增值从一个无意义数字开始。

建表时，自增字段只能设置一个

warn

MySQL InnoDB、MyISAM引擎不允许存在多个自增字段，设置多个自增字段会导致上线失败。

建议用BIGINT类型代替DECIMAL

notice

因为CPU不支持对DECIMAL的直接运算，只是MySQL自身实现了DECIMAL的高精度计算，但是计算代价高，并且存储同样范围值的时候，空间占用也更多；使用BIGINT代替DECIMAL，可避免小数的位数乘以相应的倍数，即可达到精确的手点存储计算，避免DECIMAL计算代价高的问题

禁止char、varchar类型字段字符长度总和超过阈值

error

使用过长或者过多的varchar、char字段可能会增加业务逻辑的复杂度；如果字段平均长度过大时，会占用更多的存储空间。

字符长度: 2000

700+

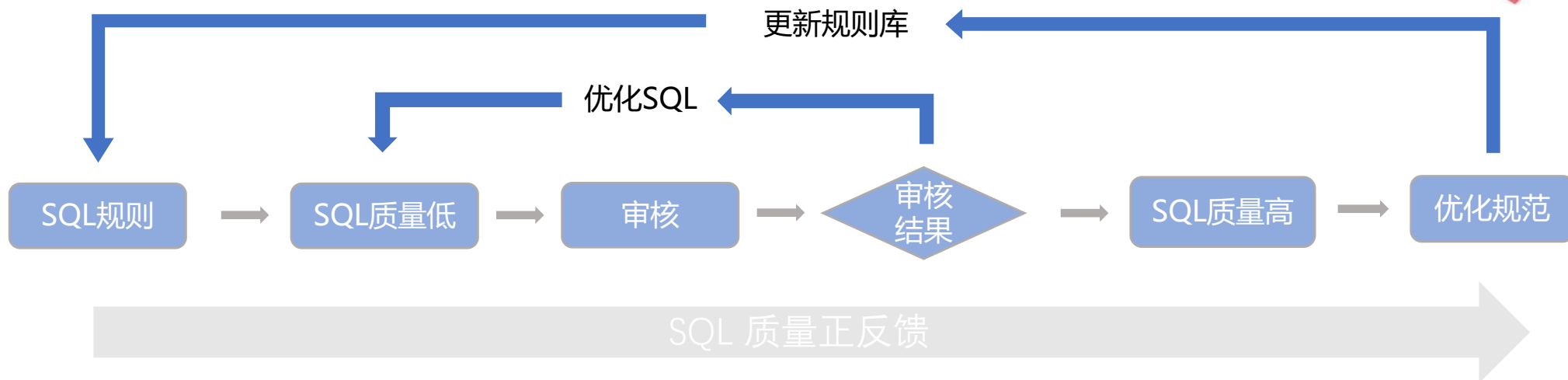
开源秀

IT168

ChinaUnix

ITPUB

完善规则规范是做好SQL质量管理的第一步



如何设计SQL规范:

- 不同公司标准不一致 (可灵活开关, 可定制)
- 审核提示 (分级, 要不要改, 优先级问题)
- 旁路模式 (白名单)

质量如何量化:

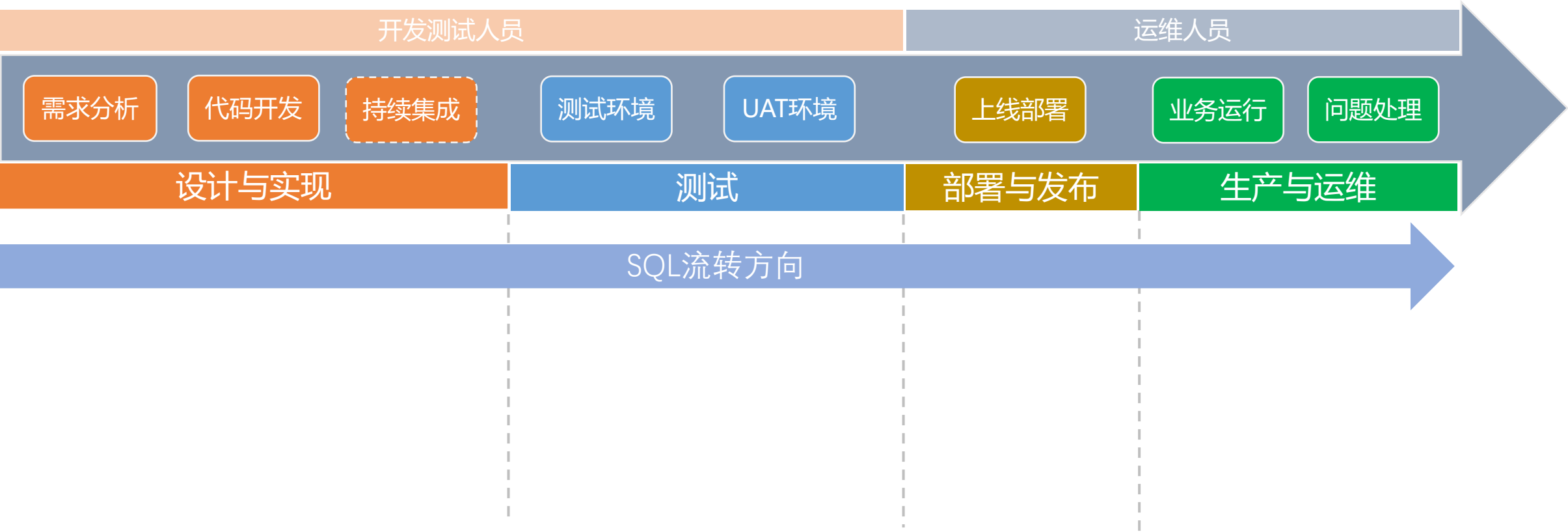
- 评分
- 审核报告
- 统计报表

问题如何优化:

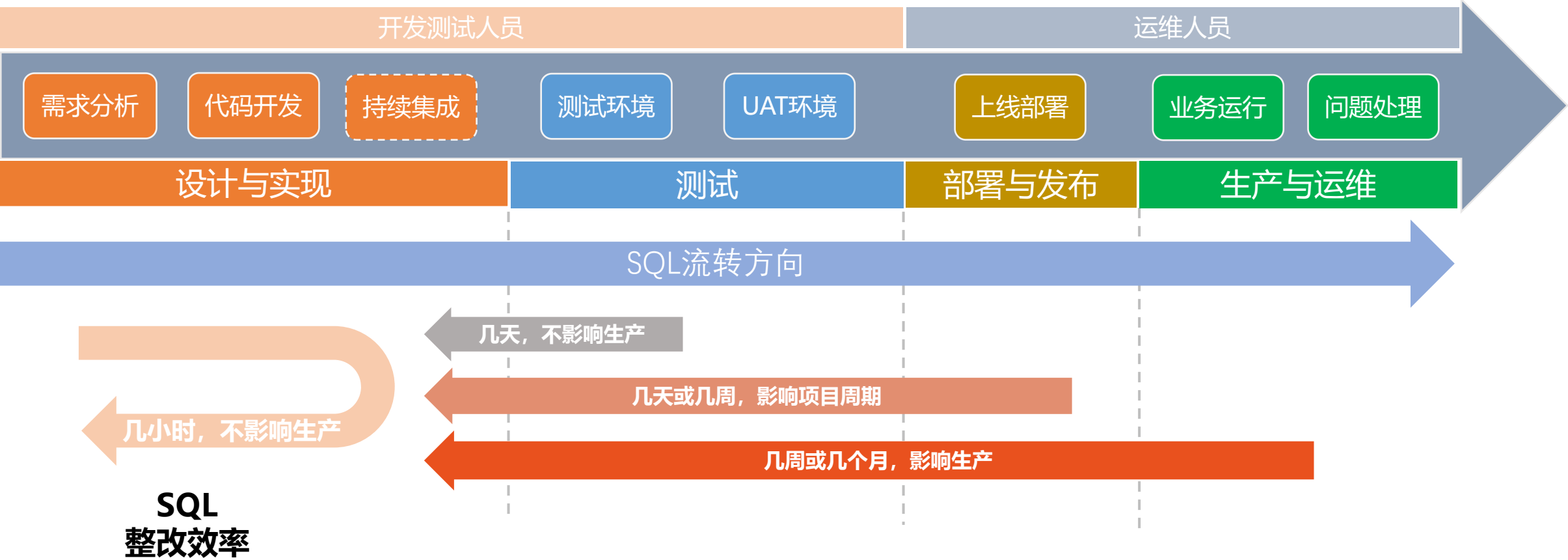
- 修改建议 (知识库)
- 辅助诊断 (SQL分析)
- SQL自动优化
- AI优化



软件生命周期



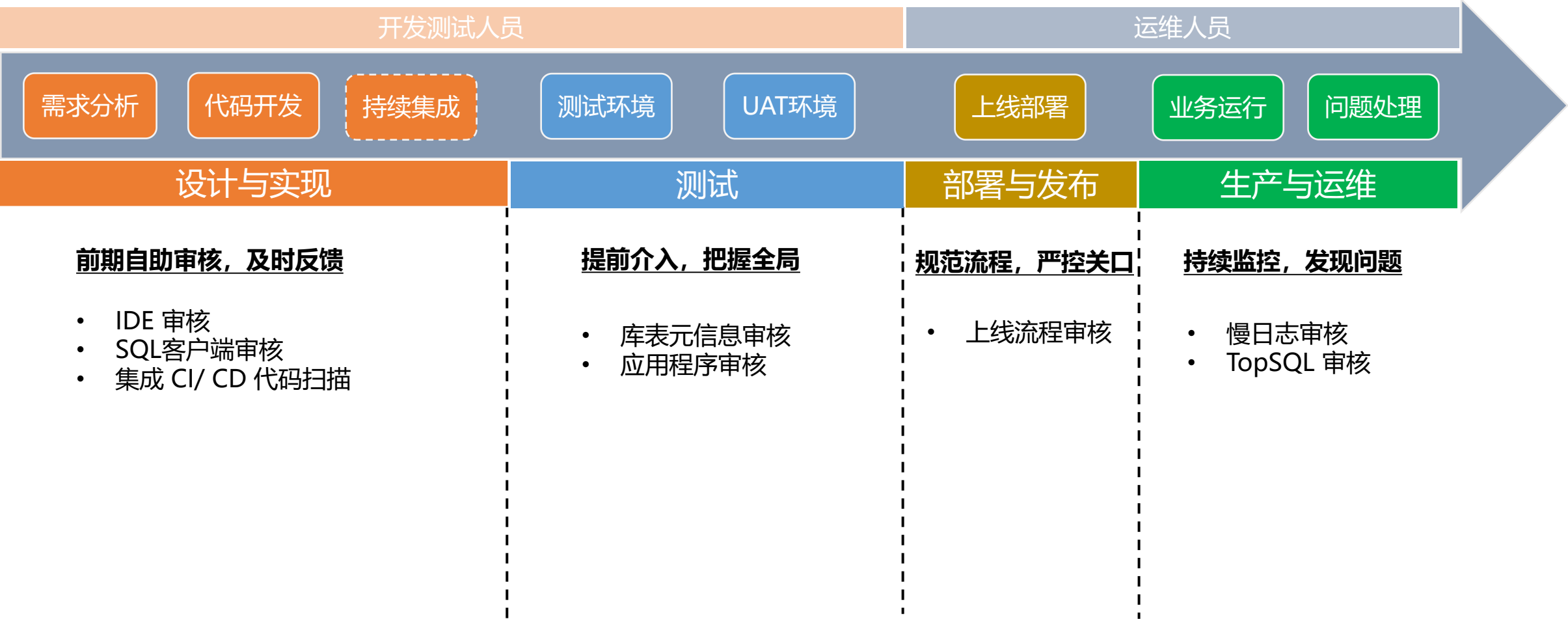
软件生命周期



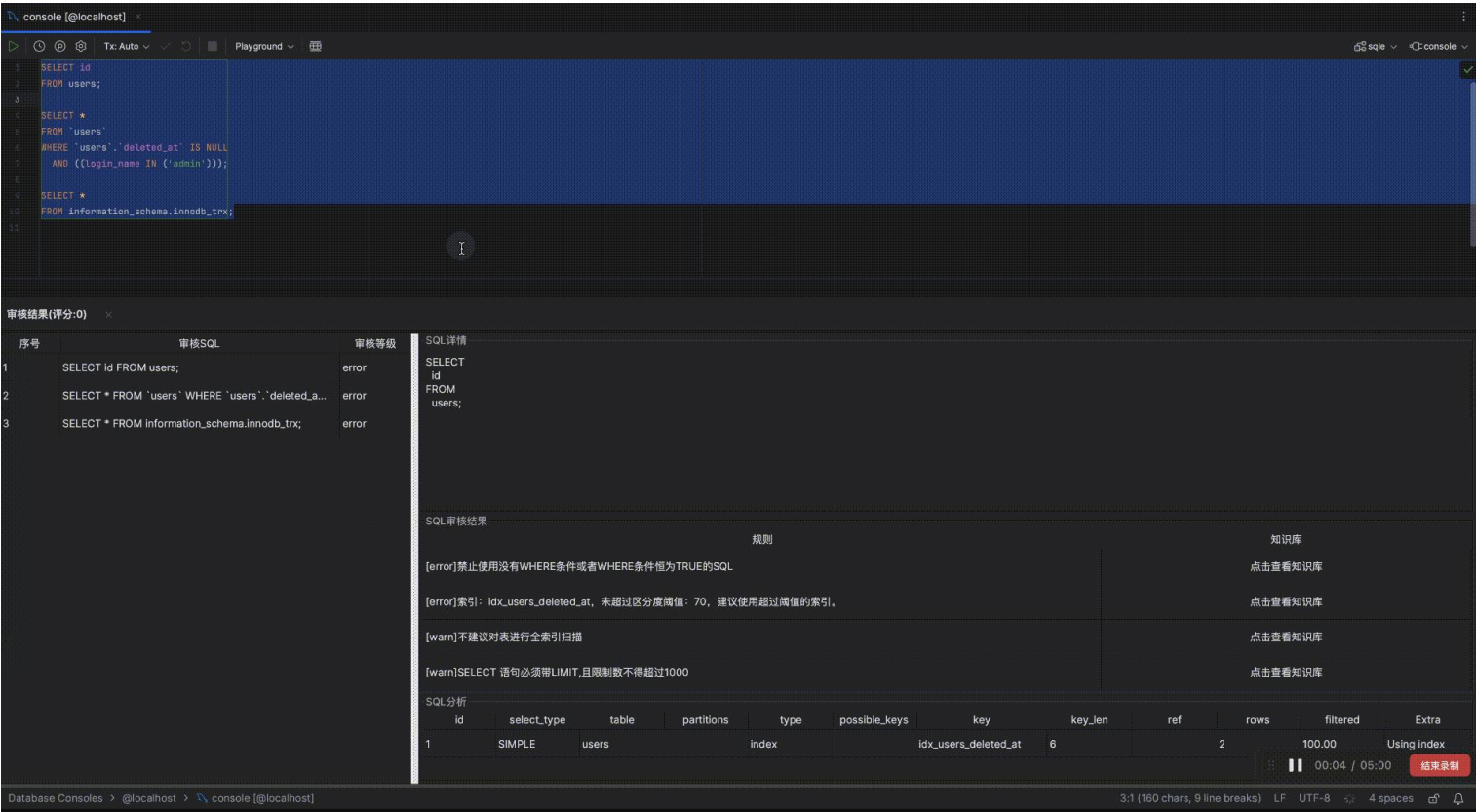
问题越早解决成本越低!



不同阶段对应的审核手段



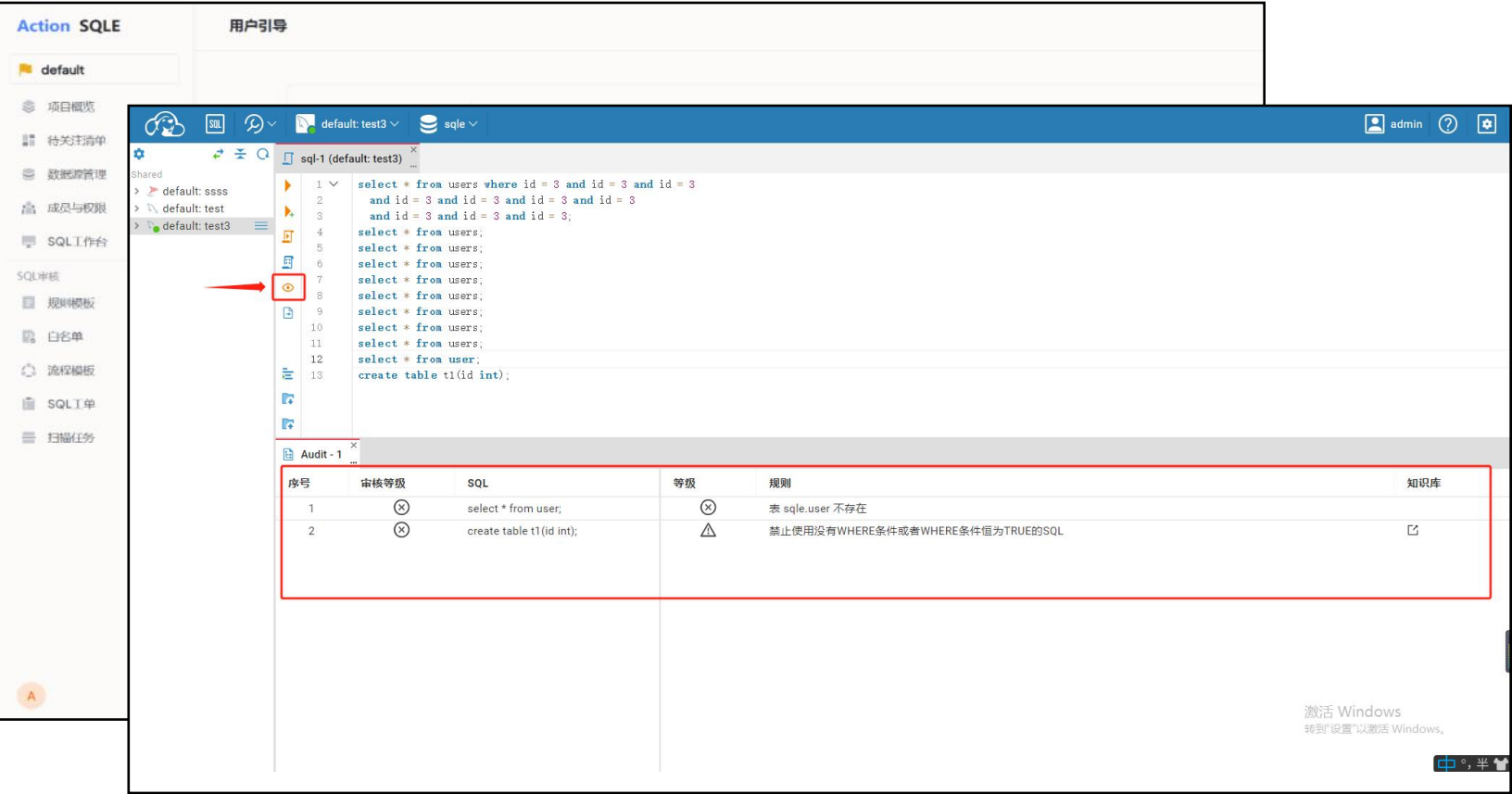
审核前置：集成 IDE 插件



集成 IDE 插件实现代码开发即审核



审核前置：SQL 工作台方便业务开发快速数据检索及了解数据库



SQLE 与 CloudBeaver 集成

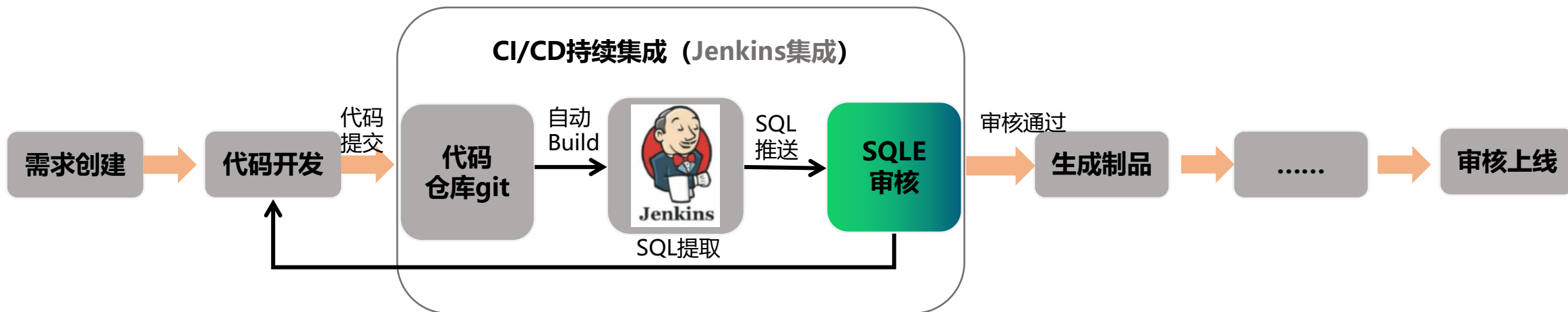
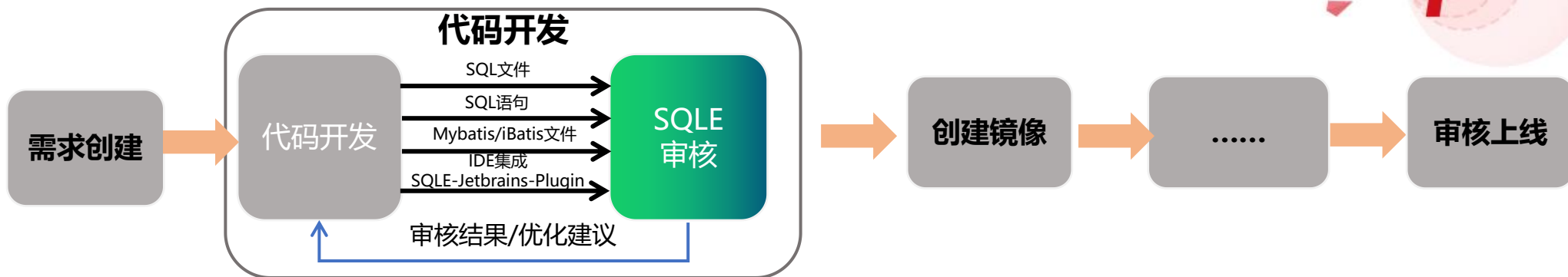
用户权限体系集成

双系统统一登录

所有的 SQL 操作通过 SQLE 审核管控，自动拦截不合规 SQL，提供 SQL 审核建议。



审核前置：集成CI/CD流程，自动审核卡点



- 提升效率：2 个工作日 SQL 审核等待 分钟级 完成
- 赋能开发，拉齐开发能力，推动企业开发规范落地

标准发布：SQL 上线卡点（产品展示）



[← 返回工单列表](#)[重置](#)

审核SQL语句信息

数据源

请选择

请选择数据库 (选填)

[添加数据源](#)[测试数据库连通性](#)

选择相同SQL ☒

当数据源类型相同时，才能使用相同SQL模式

选择SQL语句上传方式

☒ 输入SQL语句

☐ 上传SQL文件

☐ 上传Mybatis的XML文件

☐ 上传ZIP文件

1

```
/* input your sql */
```

[审核](#)[SQL美化](#)

- 多种格式文件
- 多环境发布
- 上线卡点，审核通过才能提交工单



标准发布：SQL上线卡点（产品展示）



← 返回工单列表

上线成功

mysql-1_20240112053302

-

审核结果

概览mysql-1

筛选

数据去重

下载

分页展示

全部状态

准备执行

正在执行

执行成功

执行失败

人工执行

正在中止

中止成功

中止失败

100%
审核通过率

100
审核结果评分

db1
Schema

#1

执行成功

审核通过

复制执行语句

分析

执行语句

回滚语句

审核结果

执行结果

添加说明

工单信息

基本信息

创建人admin

创建时间2024-01-12 17:33:11

工单状态上线成功

工单进度

创建工单admin2024-01-12 17:33:11

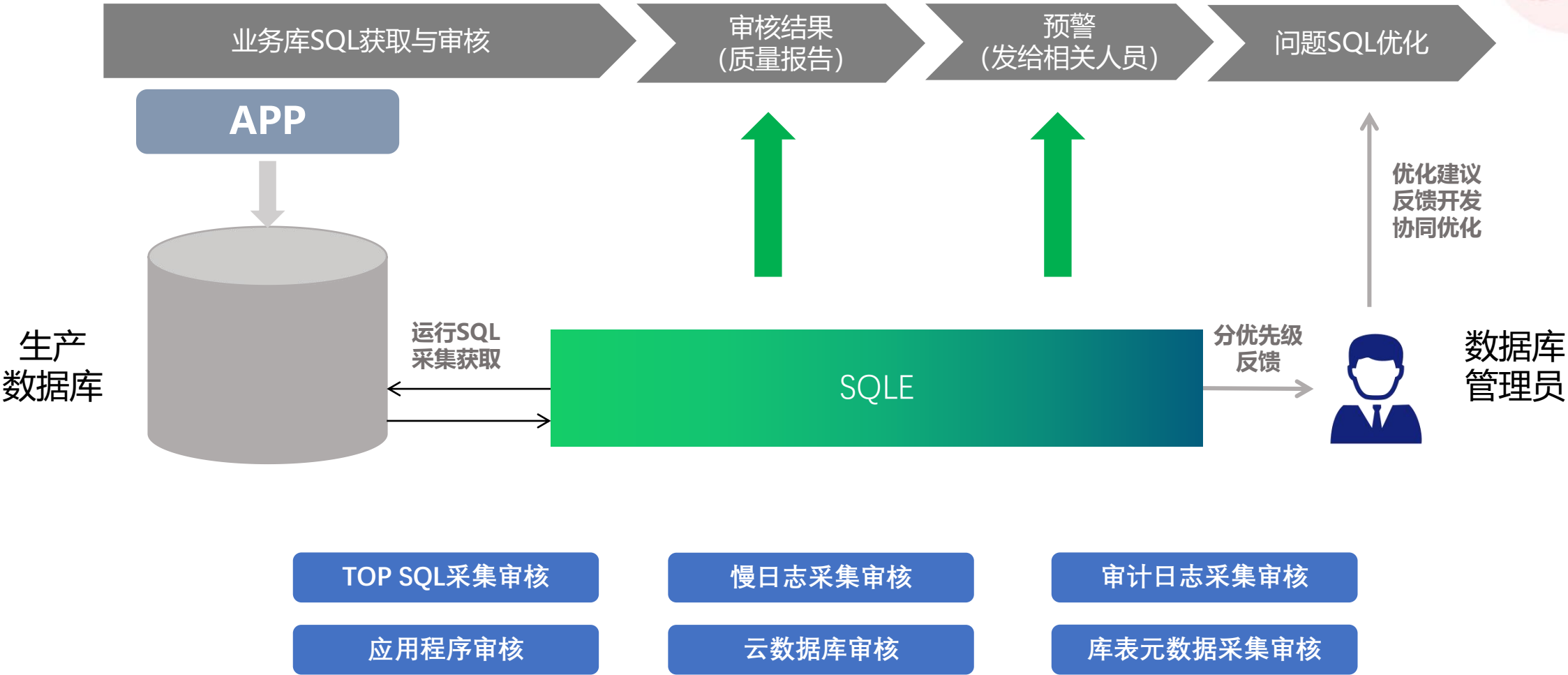
审核工单admin2024-01-14 10:42:38

上线工单
上线成功: 1
上线失败: 0
正在上线: 0

- 1. 立即/定时执行
- 2. 执行中止
- 3. 无锁发布（Online DDL）
- 4. 备份回滚



事后审核：持续监控，发现问题



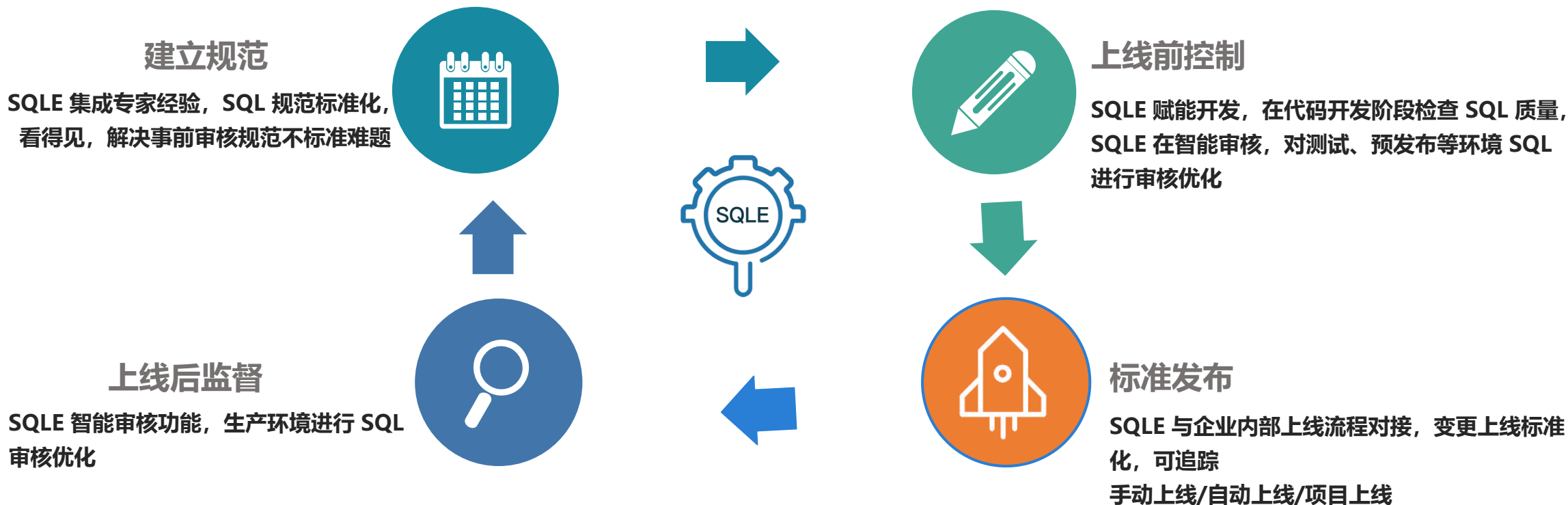
标准发布：SQL上线卡点（产品展示）



SQL管控										导出
7 SQL总数		4 问题SQL数		2 已优化SQL数						
全部 未处理 已解决 已忽略 已人工审核		输入关键字搜索		筛选			与我相关	批量指派	批量解决	批量忽略
							表格设置			
SQL指纹	SQL	来源	审核结果	数据源	负责人	状态	备注	操作		
<input type="checkbox"/> CREATE TABLE `AREA2` (`id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT...	<input type="checkbox"/> CREATE TABLE `area2` (`id` bigint(20) unsigned NOT NULL AUTO_INCREMENT...	SQL审核	×	mysql-1	-	已解决		指派负责人	变更状态	分析
<input type="checkbox"/> CREATE TABLE `area2` (`id` bigint(20) unsigned NOT NULL AUTO_INCREMENT...	<input type="checkbox"/> CREATE TABLE `area2` (`id` bigint(20) unsigned NOT NULL AUTO_INCREMENT...	智能扫描	×	mysql-1	-	未处理		指派负责人	变更状态	分析
<input type="checkbox"/> CREATE TABLE `t1` (`id` varchar(255) DEFAULT NULL) ENGINE=InnoDB DEFAULT...	<input type="checkbox"/> CREATE TABLE `t1` (`id` varchar(255) DEFAULT NULL) ENGINE=InnoDB DEFAULT...	智能扫描	×	mysql-1	A S	已解决		指派负责人	变更状态	分析
<input type="checkbox"/> CREATE TABLE `st1` (`id` bigint(20) unsigned NOT NULL AUTO_INCREMENT, `name`...	<input type="checkbox"/> CREATE TABLE `st1` (`id` bigint(20) unsigned NOT NULL AUTO_INCREMENT, `name`...	智能扫描	×	mysql-1	-	未处理		指派负责人	变更状态	分析
<input type="checkbox"/> CREATE TABLE `area22` (`id` bigint(20) unsigned NOT NULL AUTO_INCREMENT...	<input type="checkbox"/> CREATE TABLE `area22` (`id` bigint(20) unsigned NOT NULL AUTO_INCREMENT...	智能扫描	×	mysql-1	-	已人工审核		指派负责人	变更状态	分析
<input type="checkbox"/> CREATE TABLE `area2` (`id` bigint(20) unsigned NOT NULL AUTO_INCREMENT...	<input type="checkbox"/> CREATE TABLE `area2` (`id` bigint(20) unsigned NOT NULL AUTO_INCREMENT...	智能扫描	×	mysql-1	-	未处理		指派负责人	变更状态	分析
<input type="checkbox"/> CREATE TABLE `area` (`id` bigint(20) unsigned NOT NULL AUTO_INCREMENT COMMENT '主键...	<input type="checkbox"/> CREATE TABLE `area` (`id` bigint(20) unsigned NOT NULL AUTO_INCREMENT COMMENT '主键...	智能扫描	×	mysql-1	-	未处理		指派负责人	变更状态	分析

爱可生 SQL 质量管理方案：建立规范、标准发布、前控后督

——自动化审核平台 SQLE+ 专家服务经验，闭环渐进式提升 SQL 质量



方案优势： **规范标准化**，建 SQL 规范
审核流程化，自动化，审核效率高，落地推进快
质量可跟踪，随着业务发展渐进式提升 SQL 质量，提升效果可量化



SQLLE Roadmap

1. SQL 自动优化

2. SQL 上线流程优化

- 备份回滚

3. 数据库安全

- 动态脱敏
- 数据库权限控制



如何获取?

仓库: <https://github.com/actiontech/sqlc>

文档: <https://actiontech.github.io/sqlc-docs/>

官网: <https://opensource.actionsky.com/sqlc/>

SQLC 技术交流群:
添加管理员微信
ActionOpenSource



了解 SQLC 最新消息
请关注社区订阅号



开源秀