

SQLE在SQL审核上的探索和实践

上海爱可生信息技术股份有限公司





1 SQLE 对 SQL 审核的理解

2 SQL 质量管控的落地方案

3 SQLE 产品的实践

4 开源现状和规划

01

SQLE对SQL审核的理解

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;
```

创建三张表，分别使用不同的字符集 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

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

解决方案：

- 事后诊断（慢日志等）
- 事前建立规范（统一项目内库表字符集）



■ 无规则规范

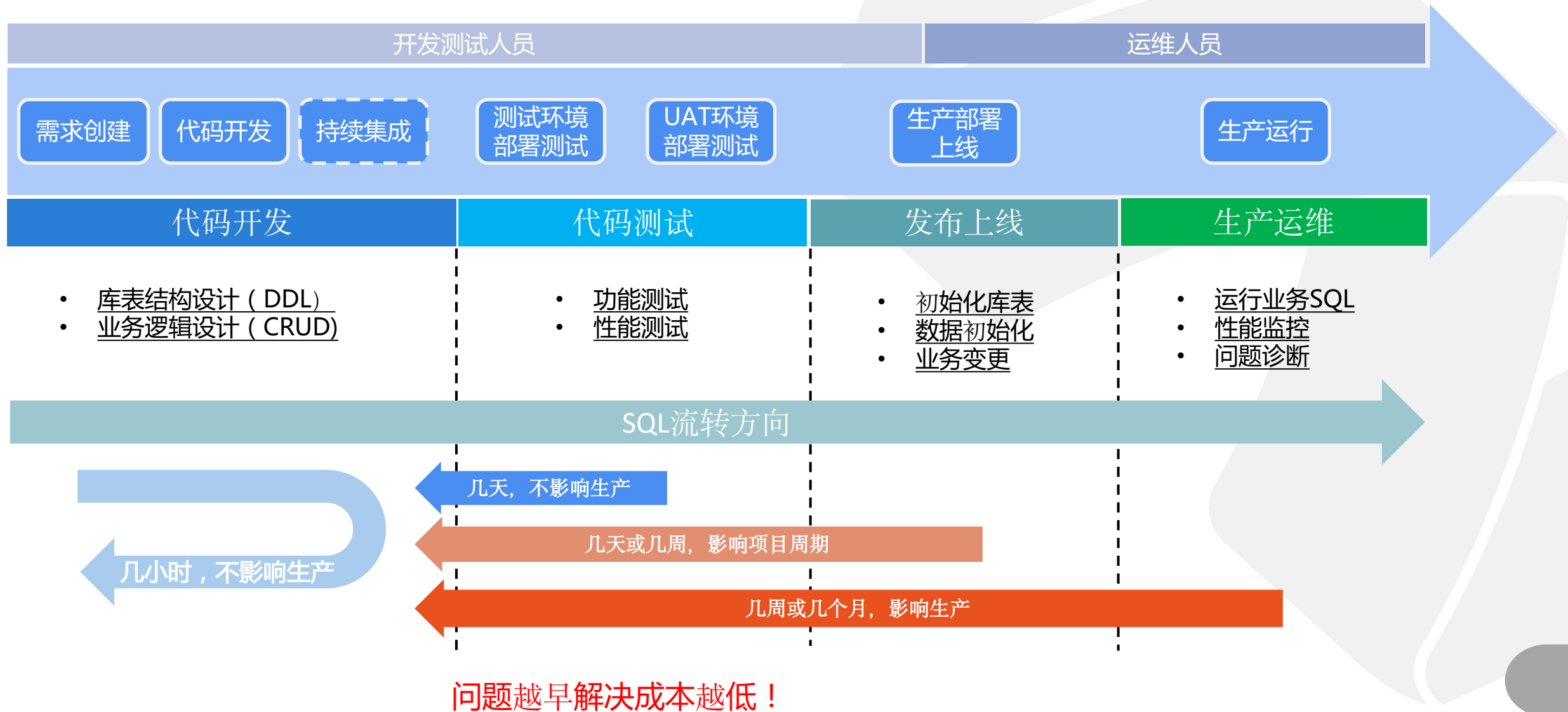
- 问题 SQL 流入生产环境；
- 审核凭经验，如何达成一致；

■ 有规则规范，人工审核

- DBA人工审核，是否会遗漏；
- 随着规模增长，如何平衡DBA人工审核工作量和效率。

推进自动化SQL审核，
建设SQL质量管控平台

SQL开发周期



02

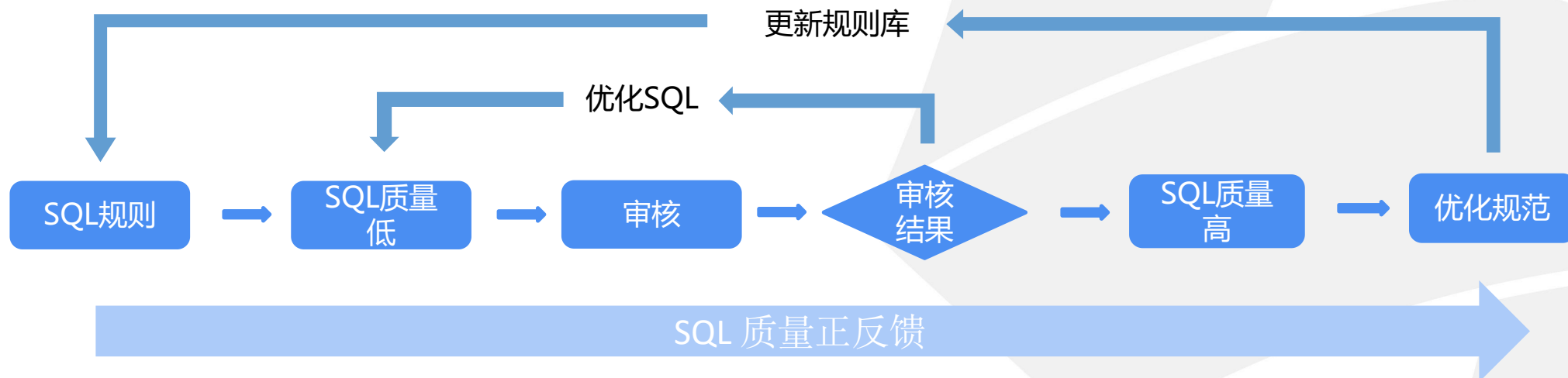
SQL质量管控的落地方案

落地需要考虑的问题

- 如何达到目标（确立目标，提高质量是目标）
- 使用什么方法
- 其他因素



提高SQL质量要考虑的问题



如何设计SQL规范：

- 规范的指定（已有规范标准，行业经验）
- 审核提示（分级，要不要改，优先级问题）
- 修改建议（怎么改）

质量如何量化：

- 评分
- 审核报告
- 统计报表

如何把握规范力度：

- 规则的灵活性
- 旁路模式（白名单）

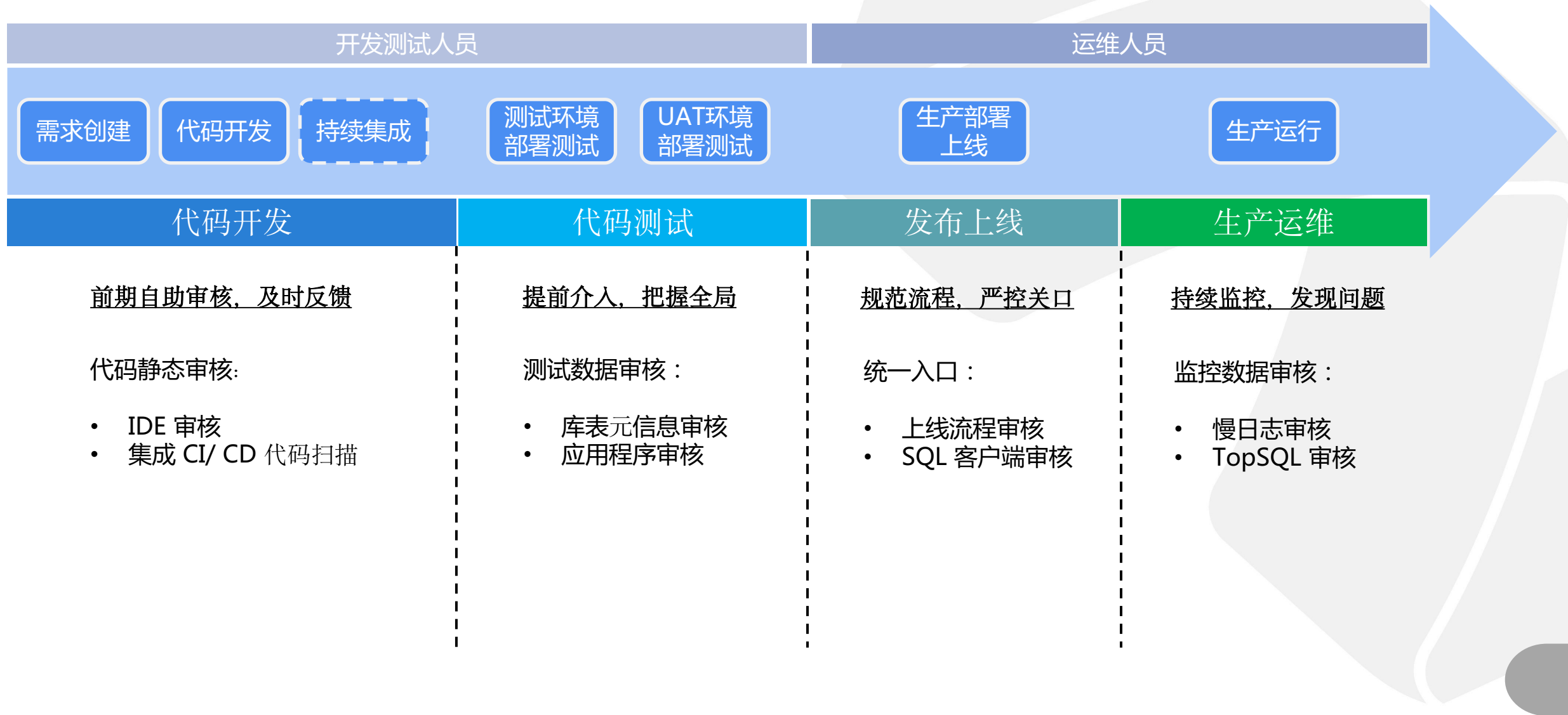
严格
宽松

落地需要考虑的问题

- 如何达到目标（确立目标，提高质量是目标）
- 使用什么方法（找到适合自己的审核模式）
- 其他因素

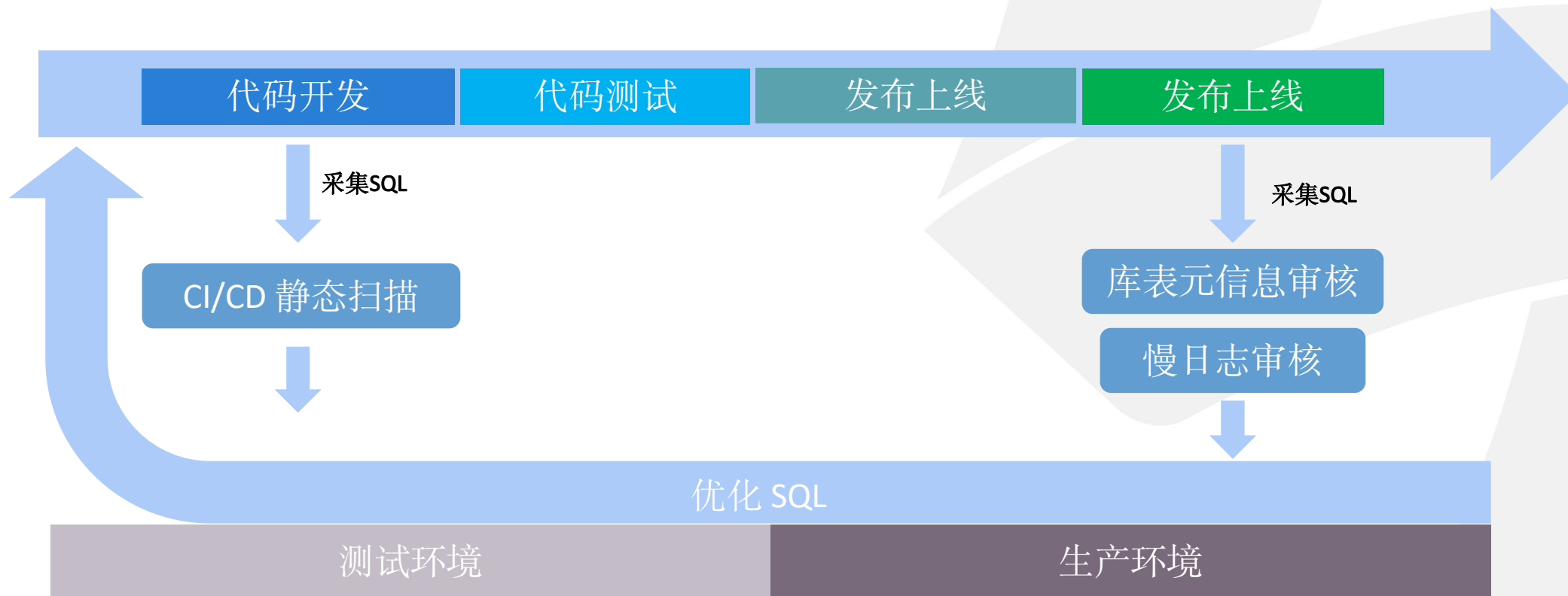


不同阶段对应的审核手段



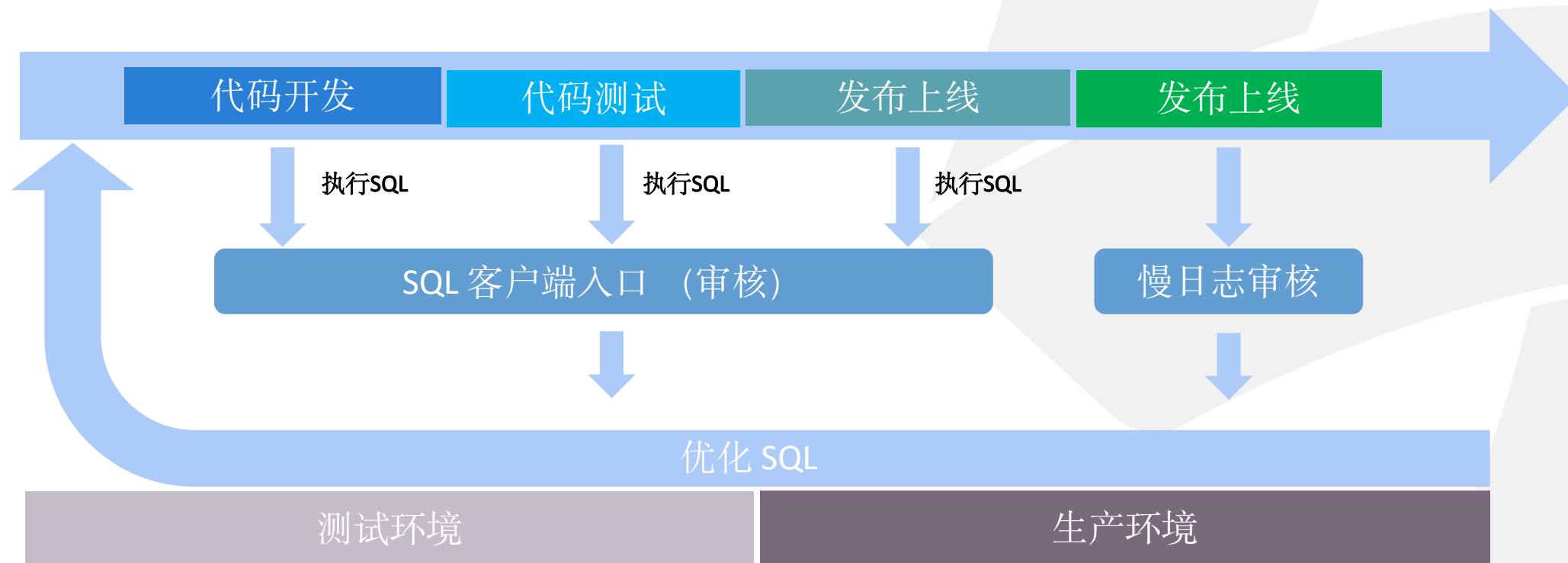
几个例子

快速落地，不改变现有 workflow



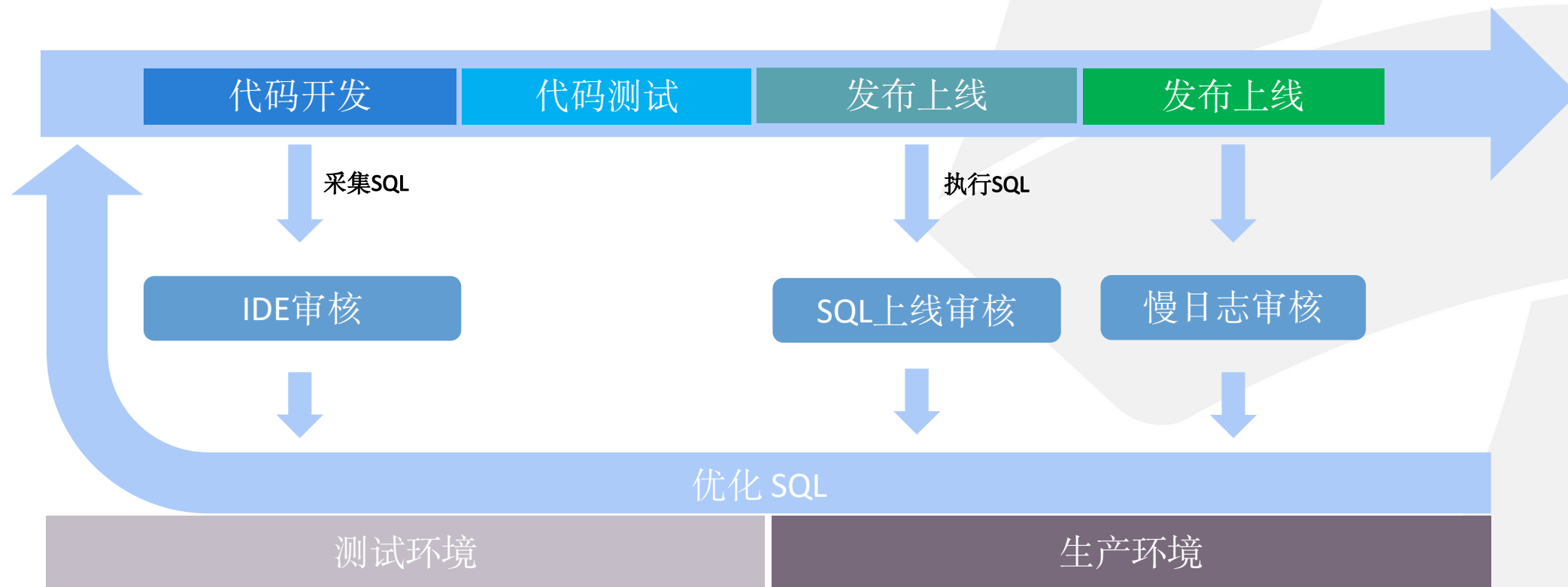
1. 低入侵性

统一SQL执行入口



1. SQL审核覆盖面广（回收数据库权限，统一SQL客户端审核）

改变开发流程



1. 开发自助审核上线提工单，建立上线标准
2. 注重前期的SQL审核反馈



- 如何达到目标（确立目标，提高质量是目标）
- 使用什么方法（找到适合自己的审核模式）
- 其他因素
 - 数据库技术栈现状和规划（数据库类型支持度，后续是否规划）
 - 既有系统的集成（流程对接，用户对接，登录对接，资源对接）

03

SQL实践

SQL规范的设计

SQL集成专家经验，构建SQL规范标准



用户体验



专家经验



信通院

《SQL质量管理平台能力分级要求》



SQL标准规范

SQL
规则库

查看规则

你可以在这里查看所有的审核规则，或者查看某一个数据库启用的所有审核规则

数据源: 请选择

数据库类型: MySQL

全部规则

ALL DDL规范 DML规范 使用建议 全局配置 命名规范 索引优化 索引规范

❗ 禁止使用没有where条件的sql语句或者使用where 1=1等变相没有条件的sql

error

📢 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

⚠ 表的初始AUTO_INCREMENT值不为0

warn

📢 建议用BIGINT类型代替DECIMAL

notice

📢 建议使用规定的数据库排序规则

notice

数据库排序规则: utf8mb4_0900_ai_ci

❗ BLOB 和 TEXT 类型的字段不可指定非 NULL 的默认值

error

📢 不建议使用 BLOB 或 TEXT 类型

notice

❗ BLOB 和 TEXT 类型的字段不建议设置为 NOT NULL

error

专家经验沉淀规范标准

数百家客户经验沉淀

爱可生专家经验沉淀

信通院SQL质量管理标准

社区用户反馈

规则库覆盖广

DDL规范、DML规范

使用建议、全局配置

命名规范、索引设计

审核结果分级展示

error (必须整改)

notice (建议整改)

warn (开发建议)

规则按需自定义，适配不同标准业务诉求（白名单/规则模板）



审核规则模版			
数据库会应用其所绑定的所有规则模板的规则并对其进行SQL审核			
规则模板列表	创建规则模板		
default_mysql 征信业务	应用的数据库	编辑	删除 更多
demo_tem 订单业务		编辑	删除 更多
default_Oracle 风控业务		编辑	删除 更多
default_DB2 贷款业务	应用的数据库	编辑	删除 更多
default_SQL Server 财富管理业务	应用的数据库	编辑	删除 更多
default_PostgreSQL 证券代售业务	应用的数据库	编辑	删除 更多
< 1 > 10条/页			

业务规则模板

根据不同业务特点，定制不同审核模板，适配不同业务需求

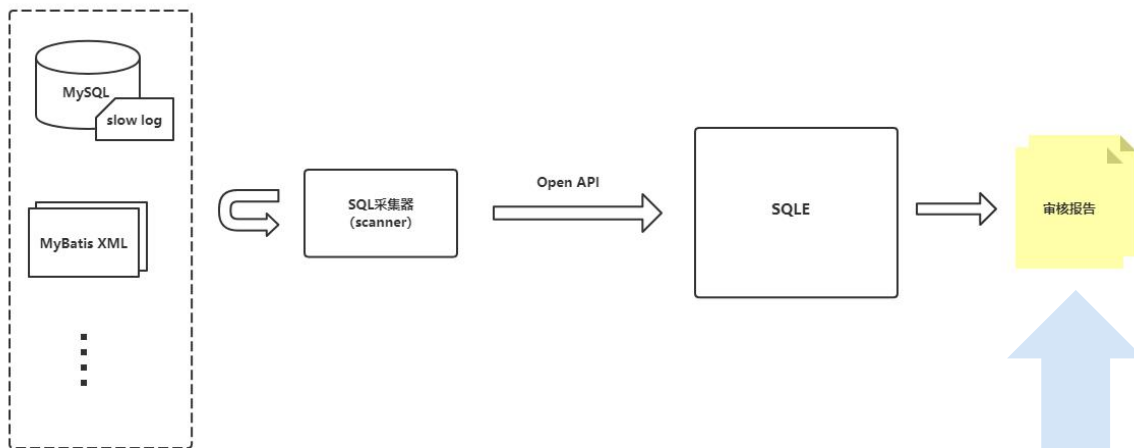
白名单			
您可以在这里添加一些SQL语句，这些SQL语句在进行审核的时候不会触发任何审核规则。			
所有白名单语句	添加白名单		
SQL语句	白名单描述	匹配模式	操作
select user,host from mysql.user;	sql whitelist of mysql username	字符串匹配	编辑 删除
< 1 > 10条/页			

SQL白名单

有些SQL不符合规范，但是必须通过审核上线，比如无法推动的旧业务

智能扫描

智能扫描：多场景的审核模式



任务名称: app1
任务类型: 库表元数据
您可以在这里查看当前智能扫描任务的SQL统计信息, 以及当前任务的审核记录

SQL统计信息 ☐ 立即审核

SQL语句

```
CREATE TABLE `t1` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(50) COLLATE utf8mb4_unicode_ci DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci  
  
CREATE TABLE `t_sync_table` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT 'id',  
  `db_name` varchar(60) COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT '' COMMENT '库名称',  
  `table_name` varchar(60) COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT '' COMMENT '表名称',  
  `is_deleted` tinyint(4) NOT NULL DEFAULT '0' COMMENT '数据状态 (0: 正常/其他: 已删除)',  
  `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',  
  `update_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '更新时间',  
  `create_by` varchar(32) COLLATE utf8mb4_unicode_ci DEFAULT NULL COMMENT '创建人',  
  `update_by` varchar(32) COLLATE utf8mb4_unicode_ci DEFAULT NULL COMMENT '更新人',  
  PRIMARY KEY (`id`) USING BTREE  
) ENGINE=InnoDB AUTO_INCREMENT=112 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci ROW_FORMAT=DYNAMIC COMMENT='登记数据同步表'
```

审核记录 ☐

生成时间2022-10-23 20:16:38
ERROR 审核结果(评分 0 审核通过率--)

< 1 > 10条/页

数据库	场景
	<ul style="list-style-type: none">MyBatis 扫描库表元数据阿里云 RDS 慢日志慢日志应用程序
	<ul style="list-style-type: none">TopSQLMyBatis 扫描
	<ul style="list-style-type: none">TopSQL
	<ul style="list-style-type: none">审计日志

连库审核，抓取TOP SQL，全面分析、展示问题SQL



* 审核任务类型: Oracle TOP SQL

采集周期 (分钟): 60

Top N: 3

V\$SQLAREA中的排序字段: elapsed_time

* 任务审核周期: ☒ 可视化选择 ☐ 手工填写

每 年 在 每月 月的 每天 日且星期 每天 的 每小时 : 每分钟

预览: *****

SQL采集灵活自定义

采集频率、TOP数量自定义

审核周期自定义

当前审核任务SQL语句池

SQL语句	总 执 行 次 数	执行 时间 (s)	CPU消 耗时间 (s)	物理 读	逻辑 读	I/O等 待时 间(s)
<code>begin gsmadmin_internal.dbms_gsm_cloudadmin.syncParameters; end;</code>	9	2.261	2.256	59	990	0
<code>SELECT COUNT(*) FROM SYS.V_\$PARAMETER2 WHERE NAME='_gwm_database_flags'</code>	9	2.243	2.241	9	117	0
<code>select dbms_qopatch.get_opatch_lsinventory() from dual</code>	1	1.52	0.154	470	7817	1.312
<code>INSERT INTO OPATCH_XINV_TAB(XML_INVENTORY) SELECT * FROM OPATCH_XML_INV</code>	1	1.445	0.08	188	3891	1.31

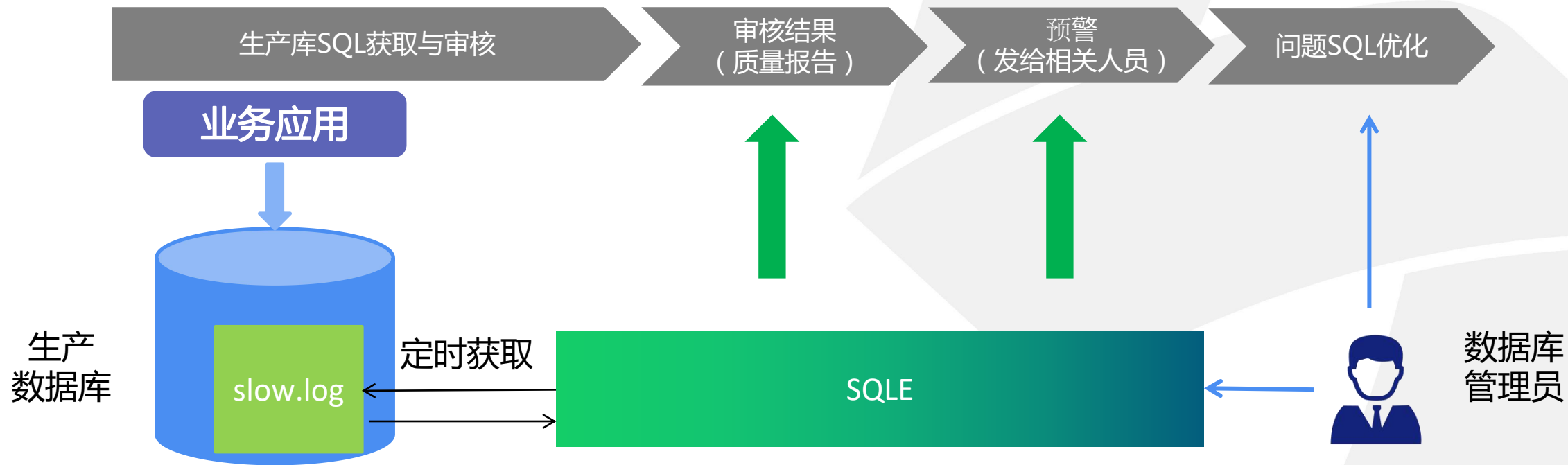
自动排序，资源消耗一目了然

TOP SQL排序展示

执行时间、资源消耗、等待时间、物理读、逻辑读

及时发现资源消耗高的SQL，效率提升80%

连库审核，抓取慢日志，全面分析、展示问题SQL



自动生成质量报告

质量报告 (通过率、评分、优化建议)

支持下载

支持多种告警方式

邮件

webhook

SQL审核上线

标准化上线发布，杜绝问题SQL流向生产



SQL上线发布流程



审核结果 审核通过率100% 审核结果评分100

下载审核报告

下载SQL语句

是否去重 ☐

审核状态: 请选择要搜索的 审核状态

执行状态: 请选择要搜索的 执行状态

审核结果: 请选择要搜索的 审核结果

重置

搜索

序号	审核状态	审核结果	执行语句	执行状态	执行结果	回滚语句 (仅提示, 不支持执行回滚)	说明
1	审核完毕		<code>select name from test where id =1;</code>	准备执行			

<

1

>

10 条/页

工单进度

操作时间:2022-03-21 16:55:14

操作人:admin

admin创建了当前工单

操作时间:2022-03-21 17:45:14

操作人:demo_user

demo_user通过了当前工单的审核

操作时间:2022-03-21 17:45:24

操作人:admin

admin通过了当前工单的审核

操作时间:--

操作人:--

定时上线

立即上线

驳回

只能在运维时间之内执行立即上线 当前数据源的运维时间为: 00: 00-06: 00

发布效率提升

变更SQL一键执行上线，提升发布效率

建立标准化发布流程

所有发布必须通过平台执行，线下人情上线减少100%，问题SQL引起的性能问题减少60%

上线时间自定义

手动实时上线
定时上线
运维时间

流程可编排自定义，无缝衔接内部流程，满足个性化需求



更新审批流程模版

1 基本信息

2 流程

3 结果

设定模版的名称、描述等基本信息

编辑审核流程

变更结果

1 工单发起/工单更新SQL语句

工单被创建，或者工单被驳回后等待修改SQL语句

2 工单审核

审核人在该步骤可以执行 审核通过或驳回 操作

审核人

步骤描述

移除该步骤

3 工单审核

审核人在该步骤可以执行 审核通过或驳回 操作

审核人

步骤描述

移除该步骤

4 执行上线

执行人在该步骤可以执行 执行上线或驳回 操作

执行人

步骤描述

创建/更新 审批流程时请注意以下几点

- 审批流程模版与数据源绑定，如果您添加数据源的时候没有选择审批流程模版，那么该数据源将无法创建工单
- 审批流程结束步骤一定为执行上线且不可移除、不可变更顺序
- 单个流程模版最多只能添加四个审核步骤
- 单个流程模版也可以移除所有审核步骤，即工单创建之后即可点击执行上线
- 单个步骤至少需要添加一个指定人，最多只能添加三个指定人

添加审核步骤

上一步

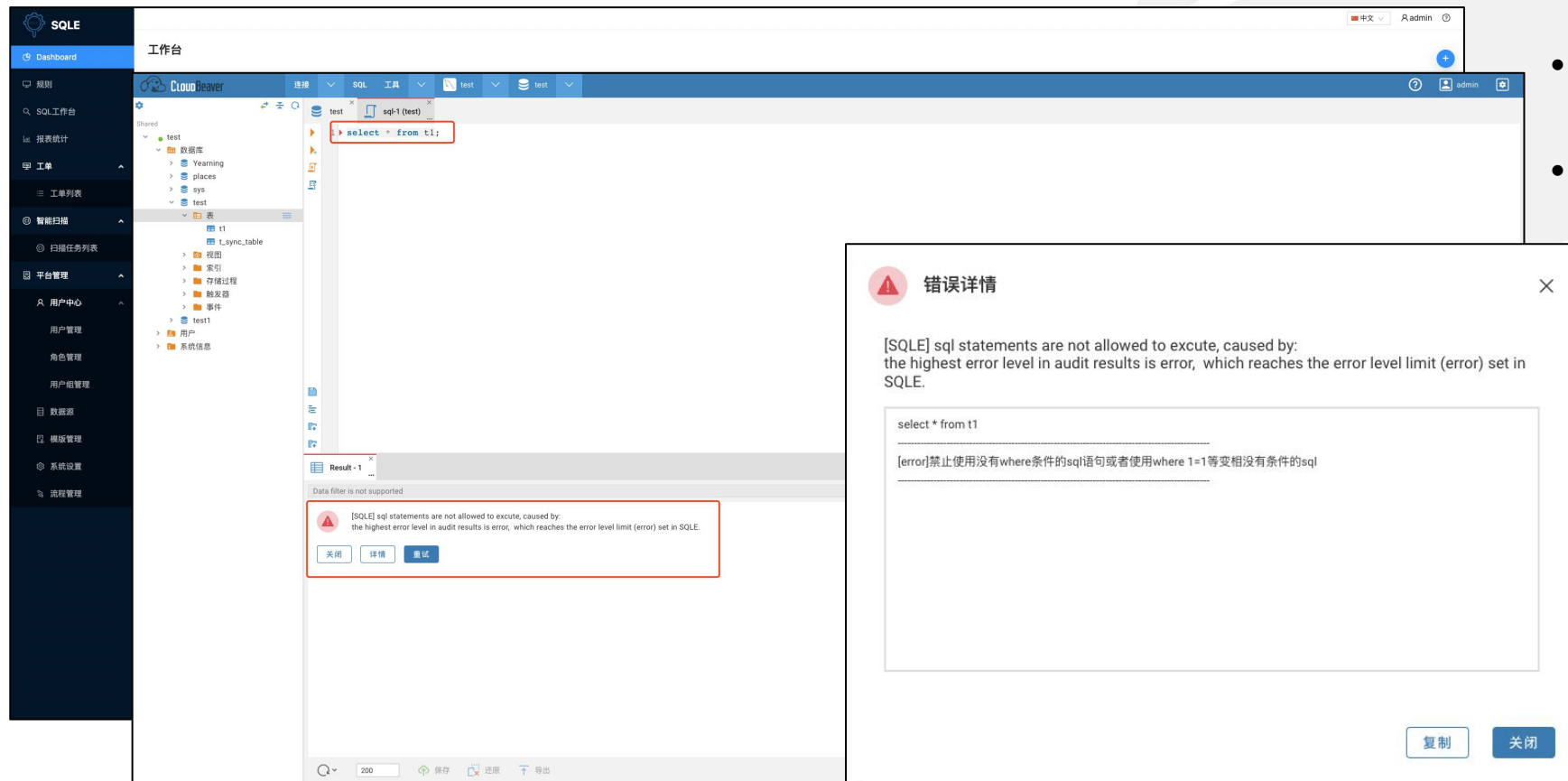
重置

提交

审核与上线流程根据实际场景按需自定义，满足企业内部不同流程管理要求

统一的SQL客户端入口

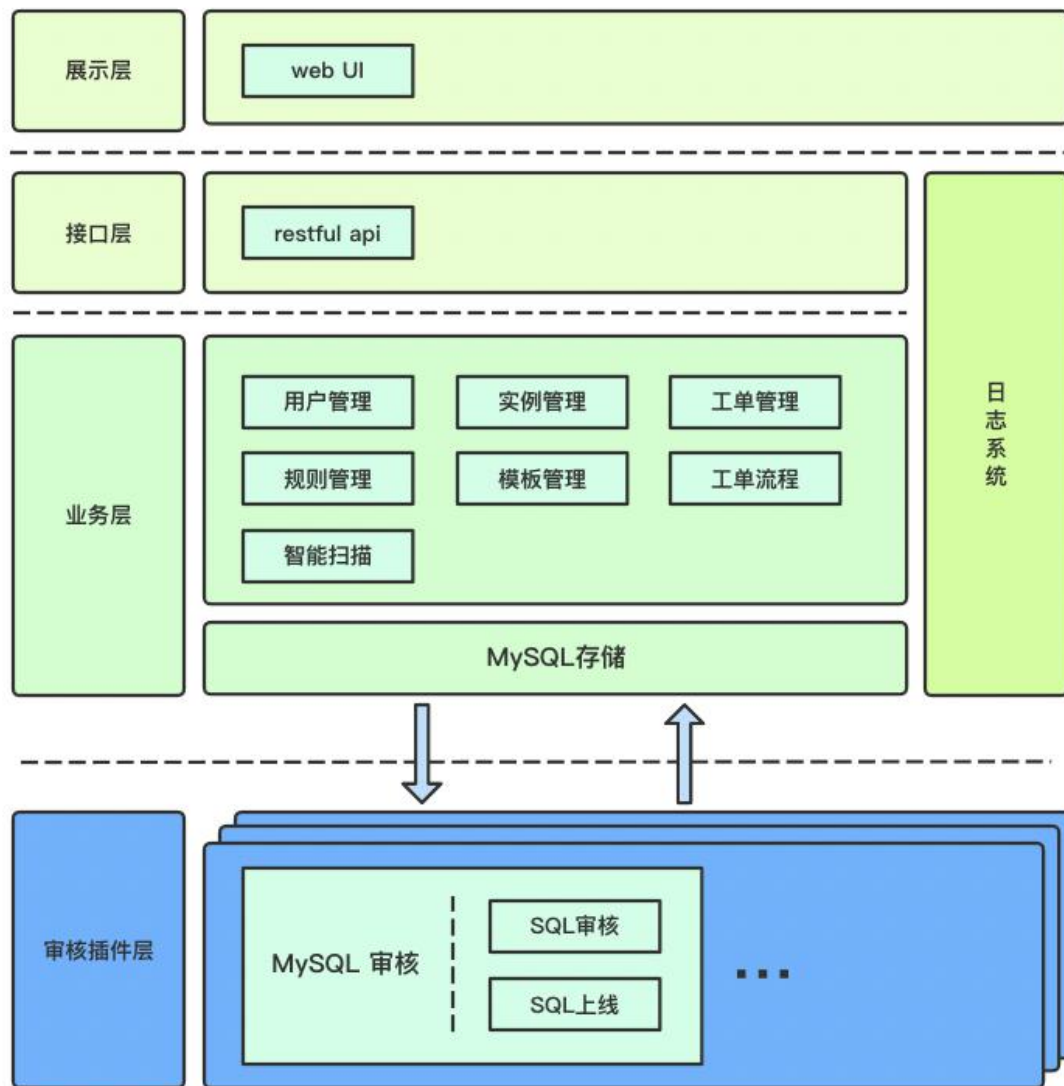
集成 CloudBeaver 作为 SQLE 的 SQL客户端



- 实现数据同步，双系统统一登录；
- 自动拦截不合规SQL

支持多数据库类型

插件化的SQL审核内核



内置的审核插件：



已支持的审核插件：



ORACLE



OCEANBASE

支持OpenAPI 对接

对接- 接入

audit_plan

GET

/v1/audit_plans 获取审核计划信息列表

lock

get audit plan info list

Parameters

Try it out

Name	Description
filter_audit_plan_db_type	filter audit plan db type
string (query)	<input type="text" value="filter_audit_plan_db_type - filter audit plan db"/>
page_index	page index
integer (query)	<input type="text" value="page_index - page index"/>
page_size	size of per page
integer (query)	<input type="text" value="page_size - size of per page"/>

Responses

Response content type application/json

Code	Description
200	OK

Example Value | Model

```
{  "code": 0,  "data": [    {      "audit_plan_cron": "0 */2 * * *",      "audit_plan_db_type": "mysql",      "audit_plan_instance_database": "appl",      "audit_plan_instance_name": "test_mysql",      "audit_plan_name": "audit_for_java_appl",      "audit_plan_token": "it's a JWT Token for scanner"    }  ],  "message": "ok",  "total_nums": 0}
```

POST

/v1/audit_plans 添加审核计划

lock

GET

/v1/audit_plans/{audit_plan_name}/ 获取指定审核计划

lock

DELETE

/v1/audit_plans/{audit_plan_name}/ 删除审核计划

lock

PATCH

/v1/audit_plans/{audit_plan_name}/ 更新审核计划

lock

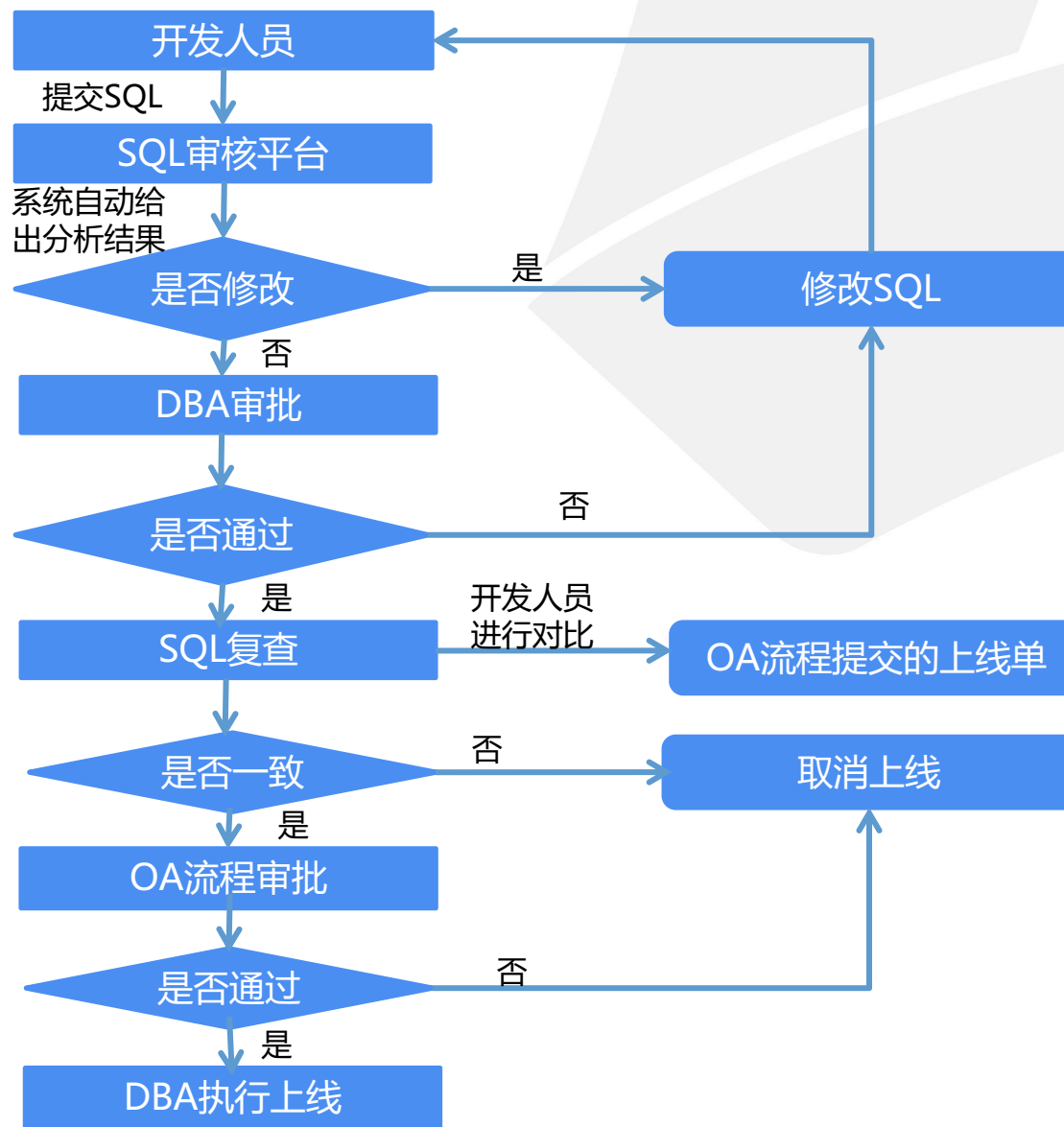
GET

/v1/audit_plans/{audit_plan_name}/report/{audit_plan_report_id}/ 获取指定审核计划CPU 系统资源

lock

标准HTTP API接口可与客户内部流程系统对接

SQL审核流程



功能总结

SQL主要功能特性总结

01. SQL审核规范

- 审核规则自定义（400+条）
- 规则模版
- 审核白名单

02. 多场景审核

- 融合CI/CD，开发环境实时审核，提前修改
- 数据库前置审核
- 数据库后置审核

03. SQL标准化上线流程

- 具备完整的工单审核流程，完善的用户管理机制，DBA、开发、测试均可按需进行功能申请
- Online DDL
- 定时上线



04. 审核管控的SQL操作入口

- 集成审核管控的CloudBeaver作为SQL客户端

05. 支持数据库种类

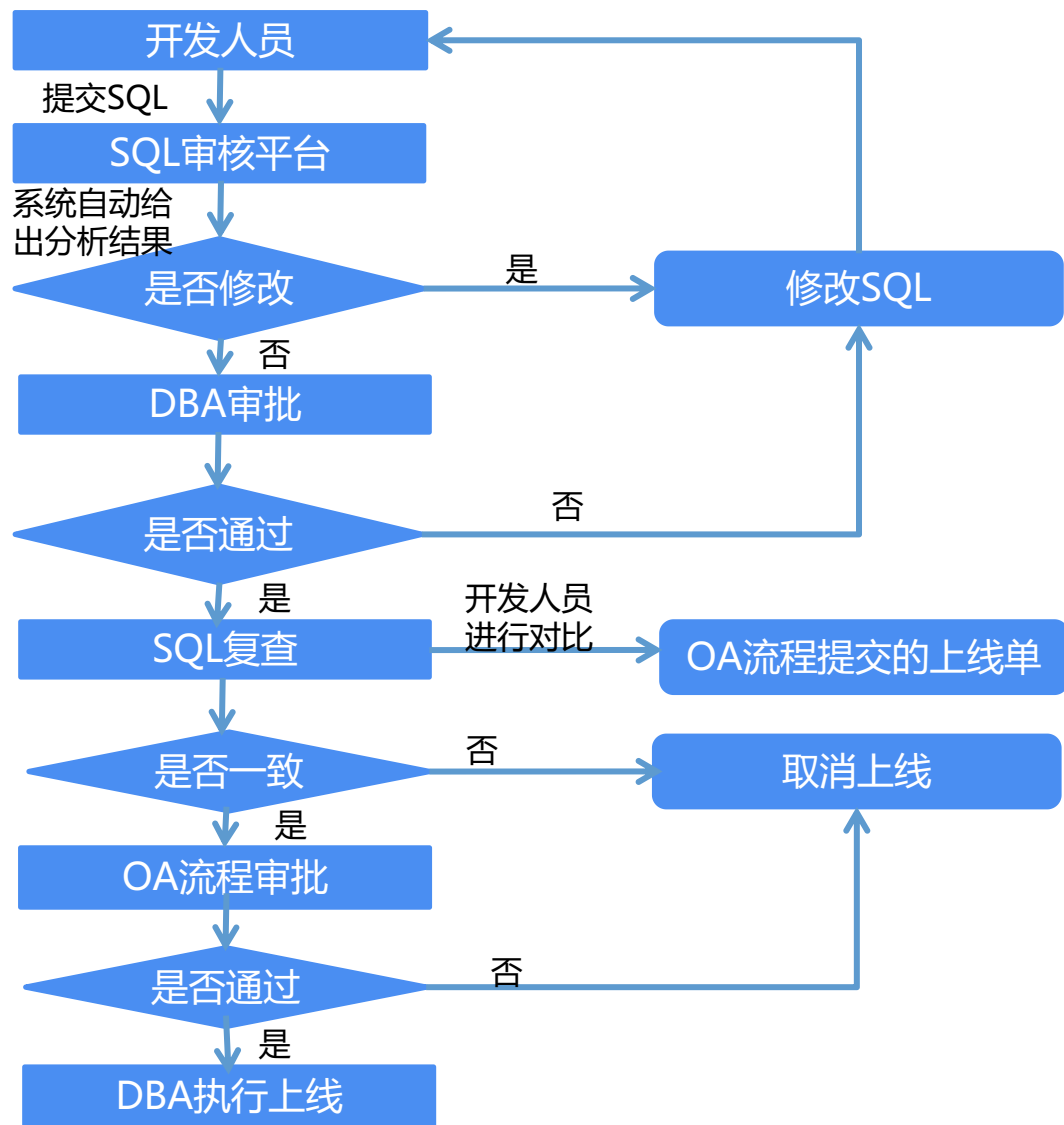
- 内置MySQL审核插件
- 统一接口，可通过插件进行多数据库审核扩展

06. 标准OpenAPI

- 支持流程，用户，审核标准接口对接

项目案例

某股份制银行——轻量



● 项目背景：

该银行的数据中心拥有规模庞大的IT基础设施和种类繁多的线上和开发测试项目，涉及到种类繁多的业务系统，需要一款优秀的SQL审核工具确保SQL审核质量，保障数据库规范落地，推进生产环境安全上线。

● 解决方案：

爱可生提供SQLE解决方案，支持 SQL 审核，上线，回滚，支持审核 MySQL，MyCat，SQL Server。能够支持同时提交多条有关联的 SQL。并在此方案基础上为该行提供如下功能：

- API：提供管理命令接口
- 语法解析器：提供sql语法语义检查
- 备份语句生成器：提供在线回滚语句
- 规则器：提供sql审核的规则
- 优化器：提供sql优化建议

● 方案价值：

- 轻量级SQL审核方案，极大的提升了SQL审核效率，推动核心系统上线稳定运行。
- 性能优化和规则优化专家服务，低效SQL及时优化。

某国有银行——完整

数据库管理员

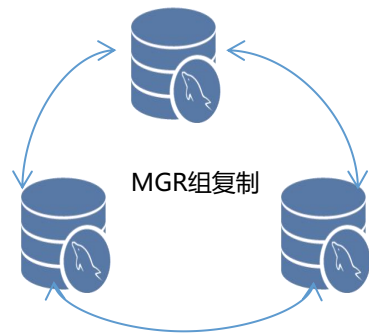


云树DMP

MySQL 8.0 高可用半同步集群



MySQL 8.0 高可用MGR集群



安装部署

配置变更

统一监控

统一告警

备份恢复

资源管理

巡检报告

性能诊断

架构变更

权限管理

SQL审核

用户管理

标准化、统一化运维



...上海A机房



...上海B机房

解决方案

- 提供**MySQL专业化服务**。协助完善了集团内部的MySQL数据库规范基线，确定MySQL的安装标准、服务管理标准、高可用管理标准。
- 提供**云树DMP平台产品**，满足行内开源MySQL统一、标准化运维管理、高效运维的要求，提升了运维效率。
- 提供**SQL审核工具**，与云树DMP平台产品无缝集成，提升上线SQL质量，减小生产性能风险。

方案价值

- 完善MySQL开发规范，提高SQL审核效率。**
- 完善MySQL运维规范体系，提升数据库运维水平。
- 同城双中心建设，满足同城RPO=0，RTO<30S的容灾建设要求。
- 国产服务器、操作系统兼容部署，**满足信创要求。**

04

SQLE开源现状与规划



开源时间

2021年10月24日

开源
一年了

★ Starred 705

Releases 48

🛡️ v1.2210.0 Latest
2 hours ago

+ 47 releases

13 个版本
33+ 新功能

<https://github.com/actiontech/sqlle>



V2.0
版本

Feature:

- 更多类型的数据库
- 更多审核规则
- 更多的审核场景
- 审核规则引擎（自定义规则）
- 项目管理
- 集群模式





技术即服务 Technology as a Service

谢谢观看！