

让所有人不再为 SQL 问题头疼

全方位 SQL 质量管理平台 SQLE

爱可生开源社区主理人 管长龙

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?



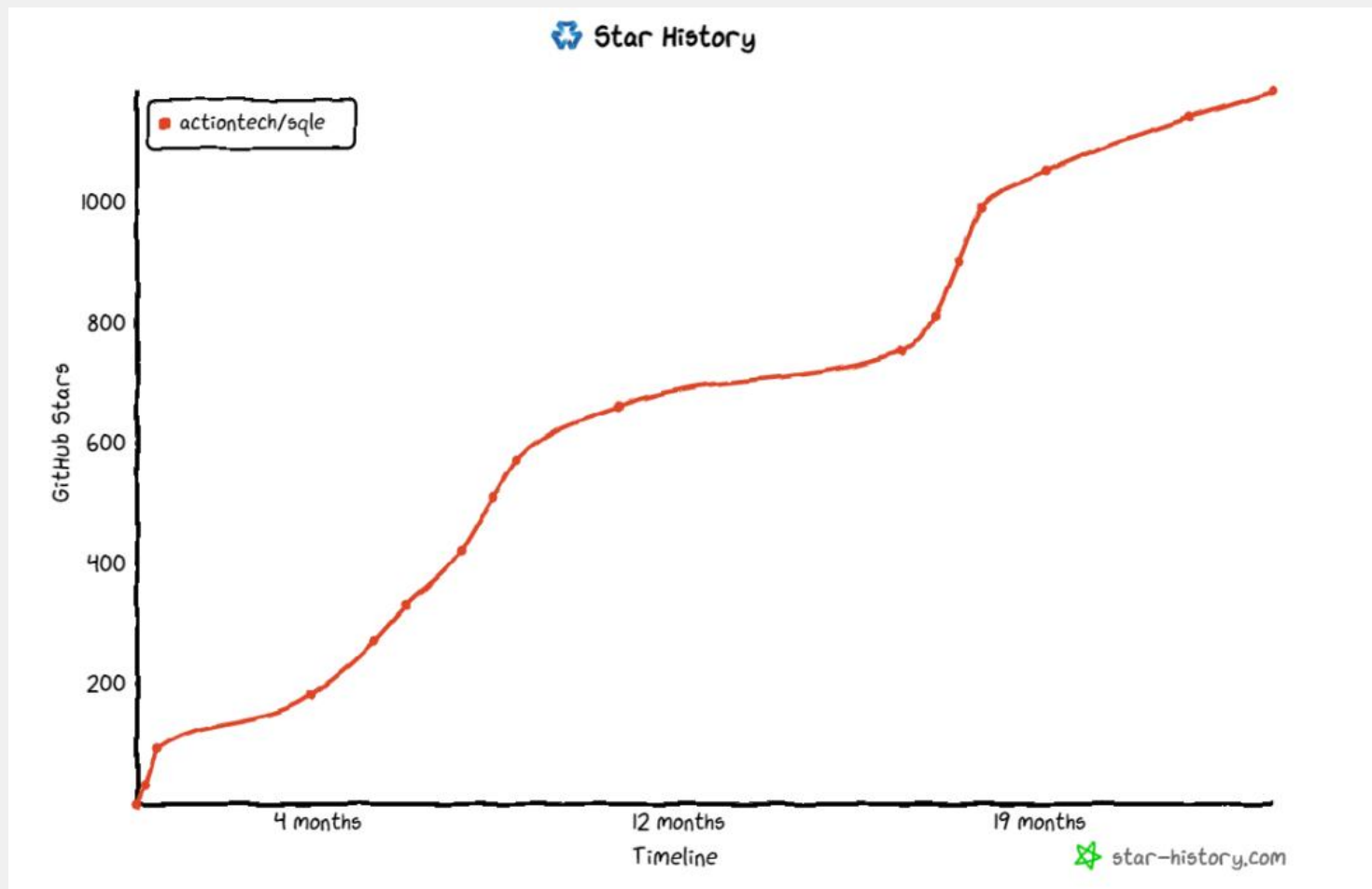
SQLE 是一款全方位的 SQL 质量管理平台，覆盖开发至生产环境的 **SQL 审核和管理**。支持主流的**开源、商业、国产数据库**，为开发和运维提供流程自动化能力，提升上线效率，提高数据质量。

什么是 SQLE?

2021 年 10 月 24 日开源。

过去的两年多，**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+**。



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;
```

建三张表，分别使用字符集 **utf8** 和 **utf8mb4**

```
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;
```

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>
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

← 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

SQL审核案例

mysql_audit_template

项目: default

数据源类型: MySQL

请输入规则关键词搜索



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



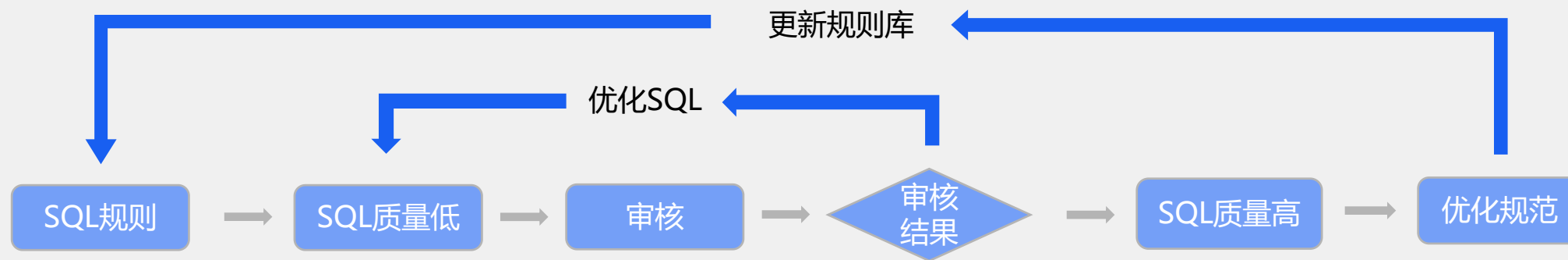
使用联合索引时，必须使用联合索引的首字段

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 质量正反馈

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)

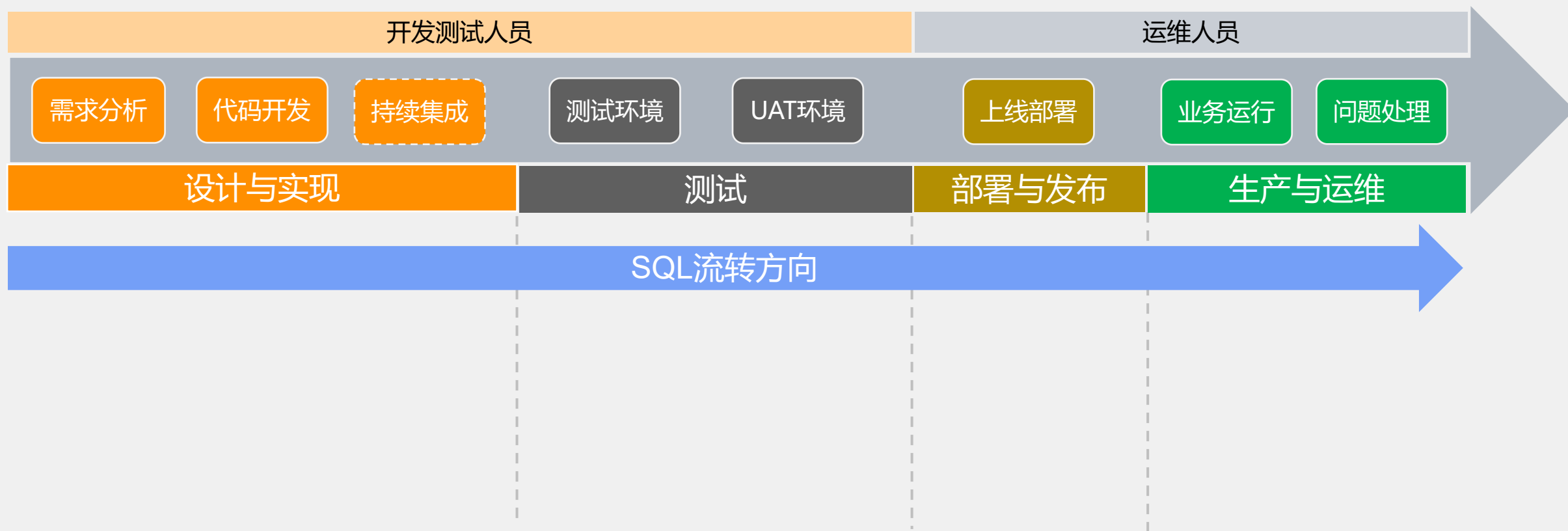
质量如何量化:

- 评分
- 审核报告
- 统计报表

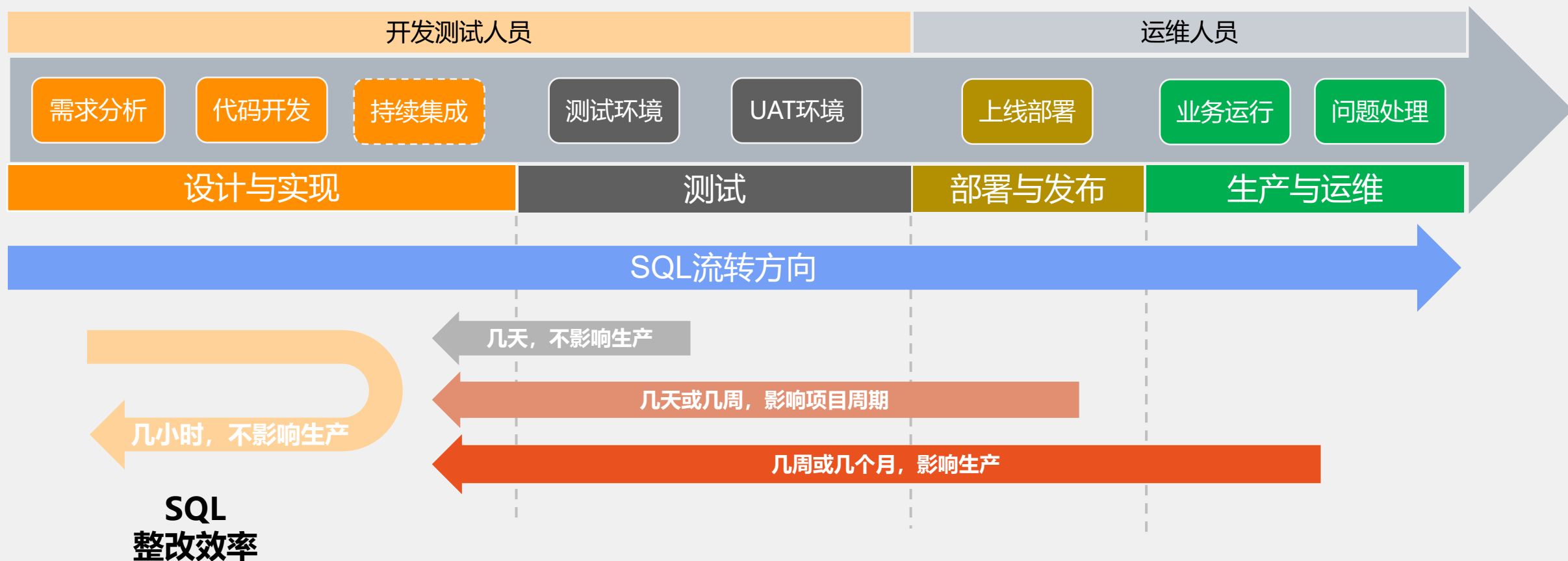
问题如何优化:

- 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

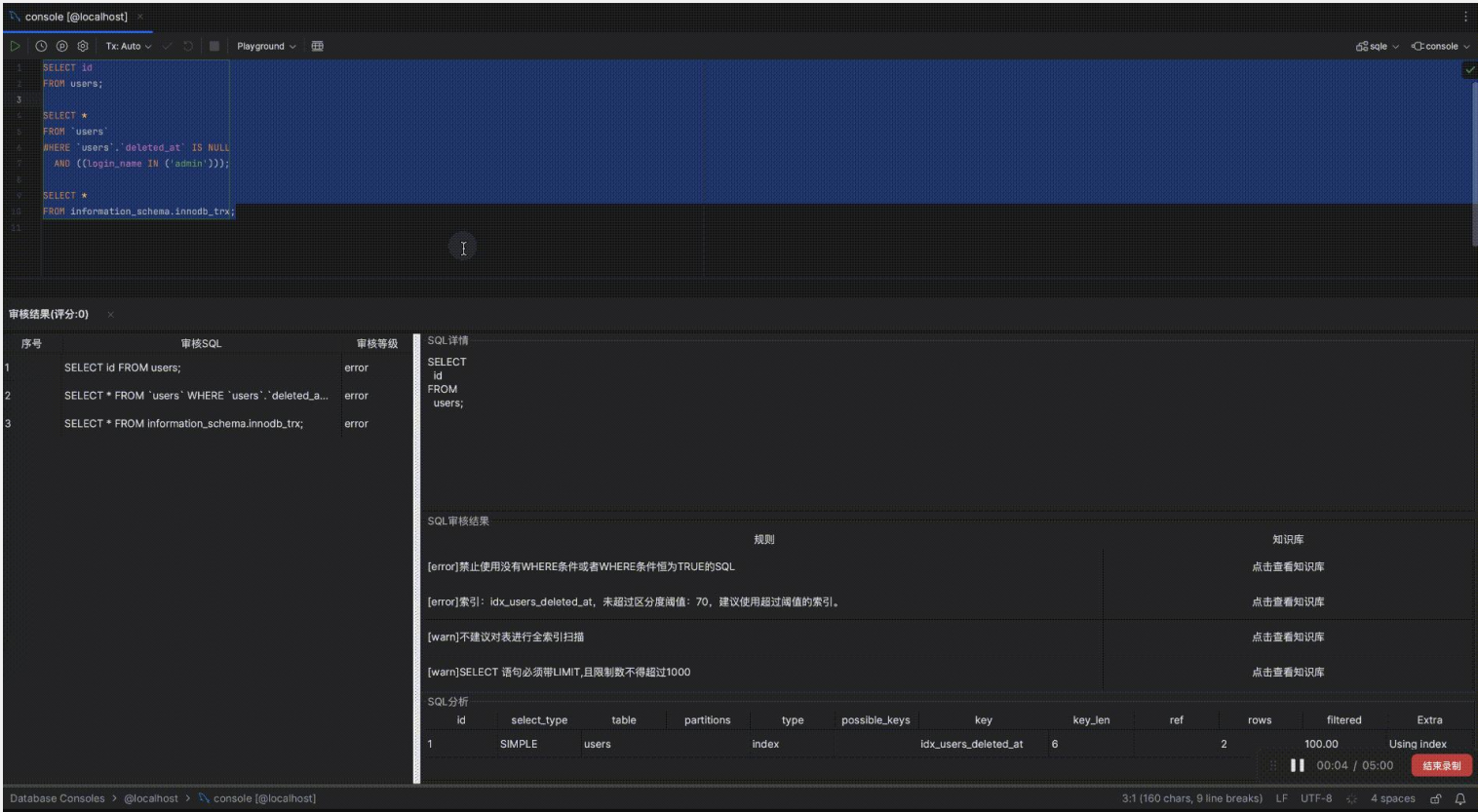


问题越早解决成本越低!

不同阶段对应的审核手段



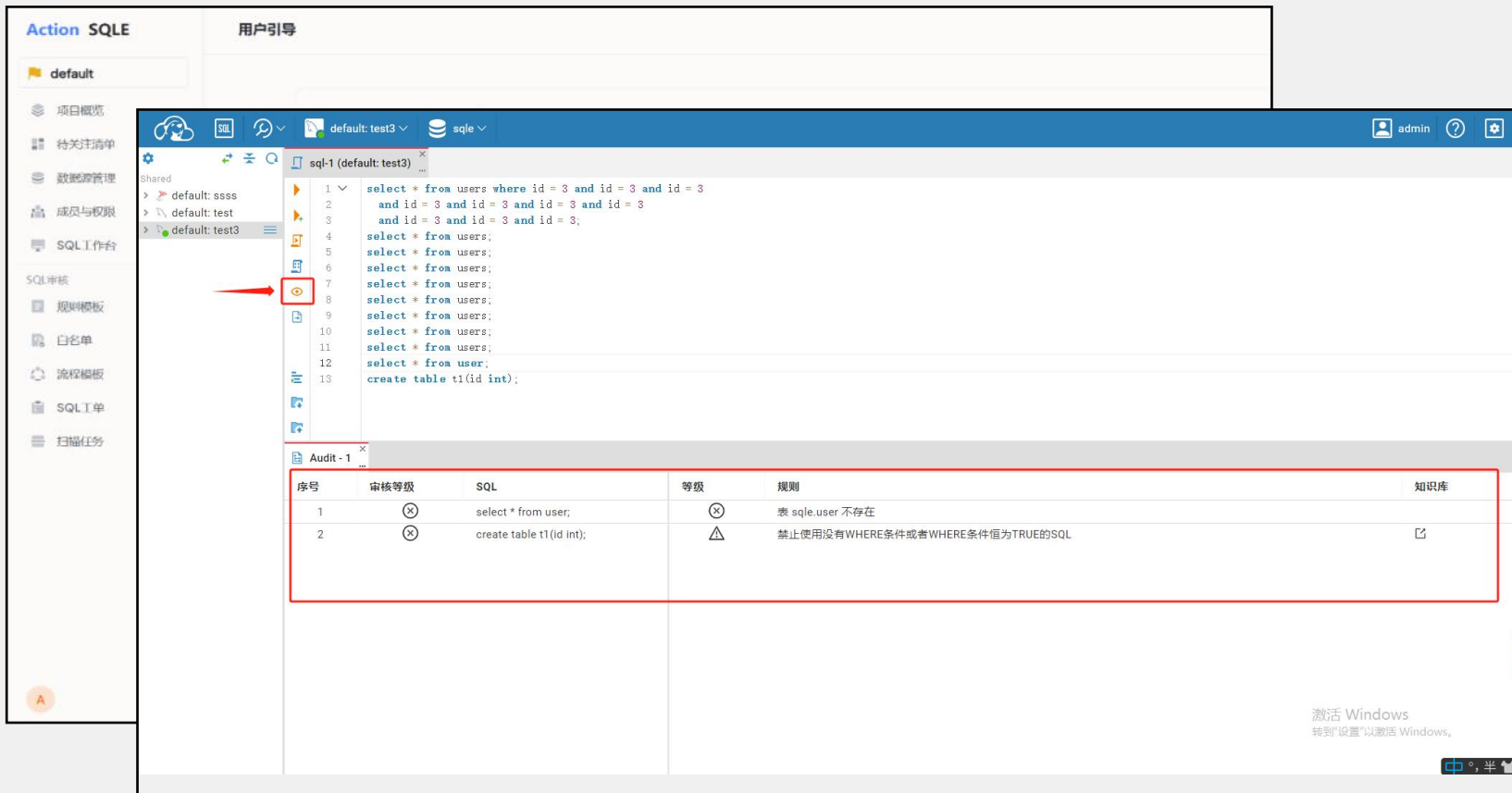
审核前置：集成 IDE 插件



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



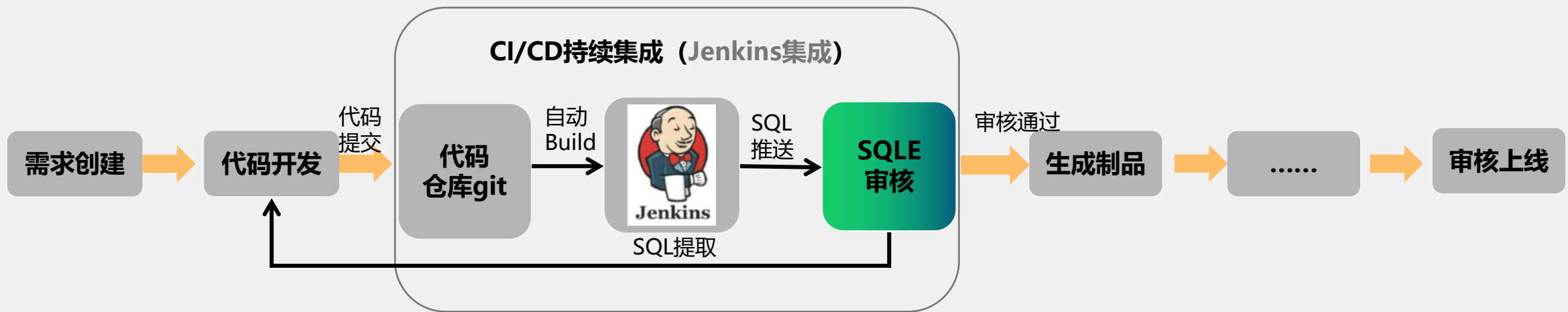
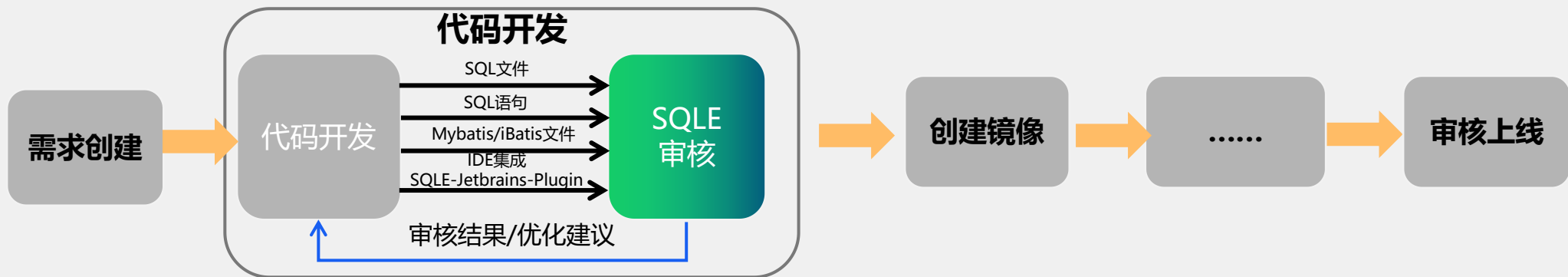
SQL 与 CloudBeaver 集成

用户权限体系集成

双系统统一登录

所有的 SQL 操作通过 SQL 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

db1Schema

#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

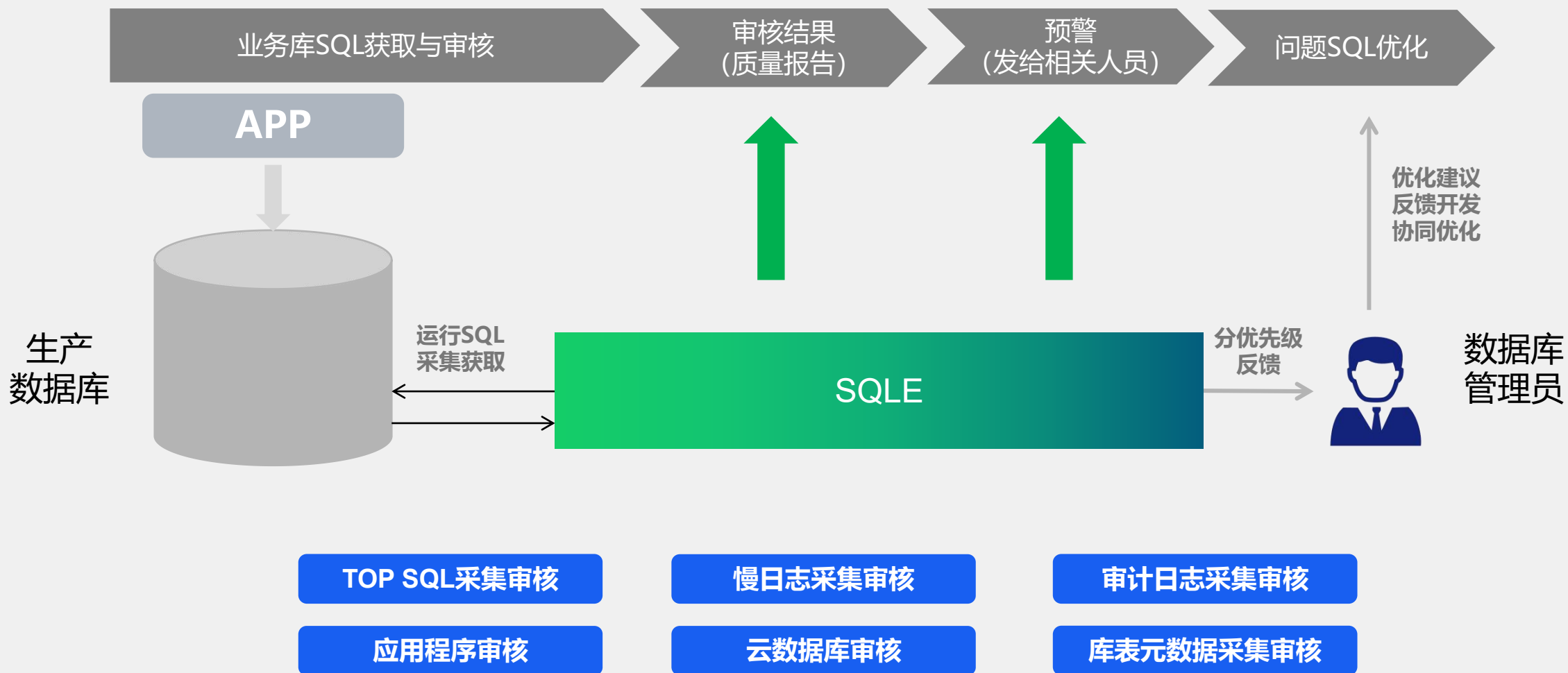
上线成功: 1

上线失败: 0

正在上线: 0

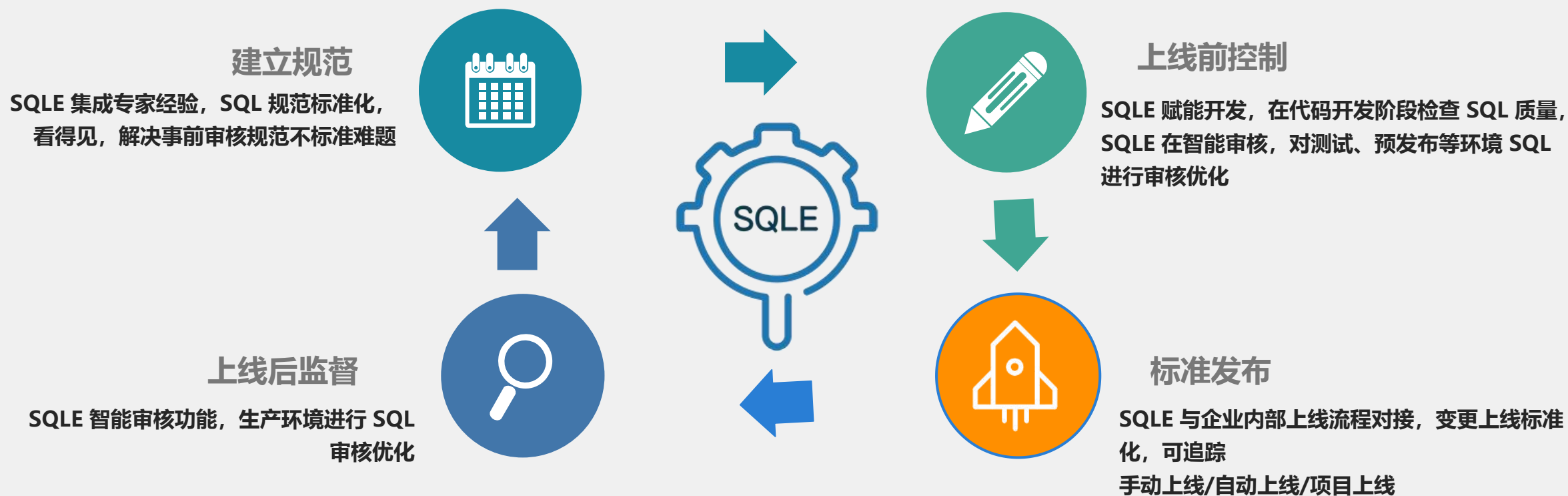
1. 立即/定时执行
2. 执行中止
3. 无锁发布（Online DDL）
4. 备份回滚

事后审核：持续监控，发现问题



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



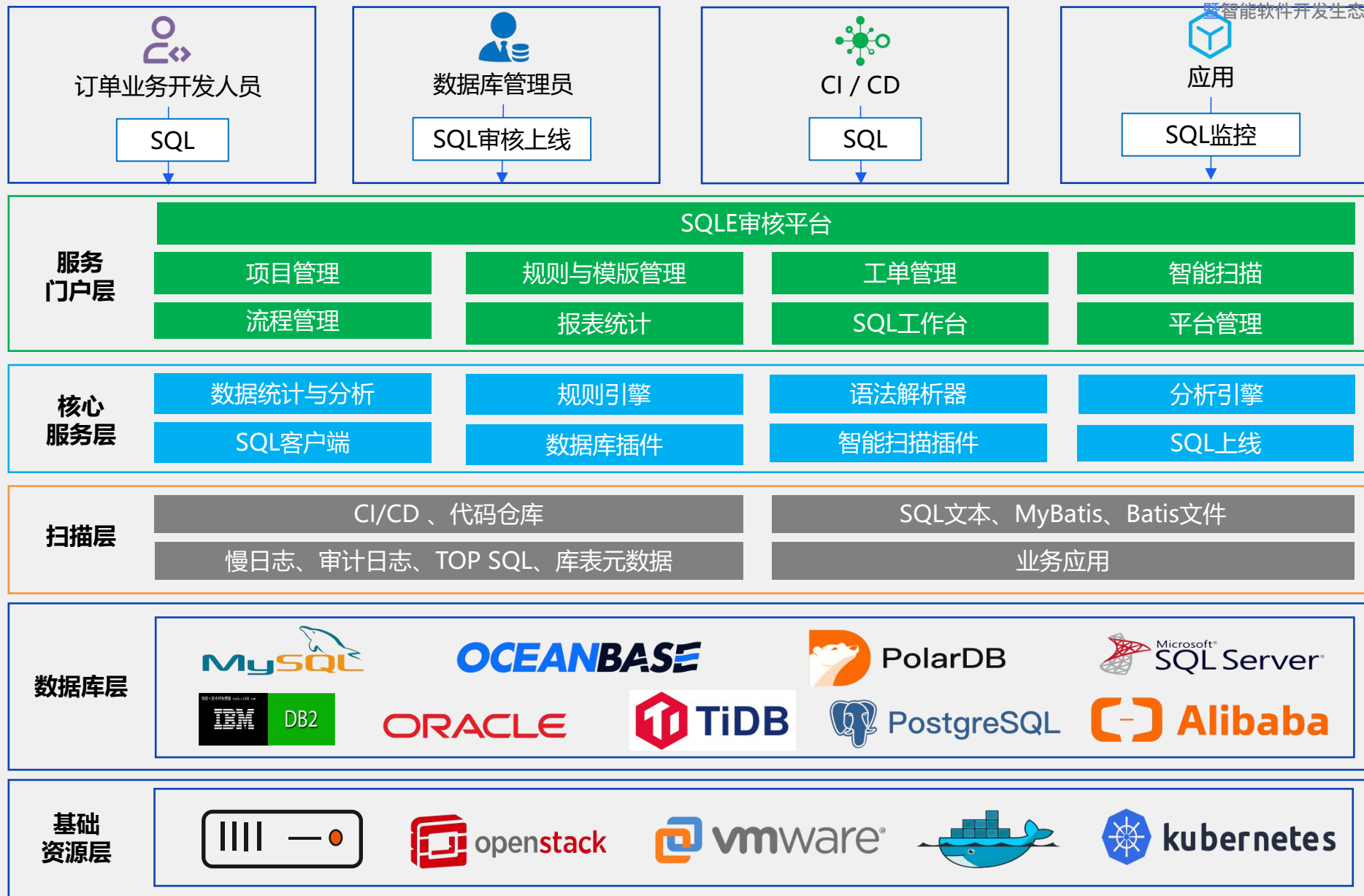
方案优势： **规范标准化**，建 SQL 规范
审核流程化，自动化，审核效率高，落地推进快
质量可跟踪，随着业务发展渐进式提升 SQL 质量，提升效果可量化

产品架构

灵活高效



- ✓ 灵活应用与开发、测试、运维变更、生产运维等审核场景，规则灵活按需调整满足不同业务场景
- ✓ 平台自动分钟级完成审核，审核效率高。



开箱即用

SQLE主要版本

社区版

免费（永久）

快速部署

适用于MySQL基础审核场景

- ✓ 支持纳管MySQL数据源
- ✓ 支持资源权限隔离
- ✓ 集成CloudBeaver在线查询
- ✓ 提供专业SQL审核能力
- ✓ 支持多方式采集MySQL数据
- ✓ 无实例限制

专业版

免费（永久）

立即申请

适用于多种数据源类型体验场景

- ✓ 支持纳管10多种主流数据源
- ✓ 支持资源权限隔离
- ✓ 集成CloudBeaver在线查询
- ✓ 提供专业SQL审核能力
- ✓ 支持多方式采集各数据源数据
- ✓ 最大实例个数20

企业版

定制

联系我们

适用于私有云用户定制场景

- ✓ 支持纳管10多种主流数据源
- ✓ 支持资源权限隔离
- ✓ 集成CloudBeaver在线查询
- ✓ 提供专业SQL审核能力
- ✓ 支持多方式采集各数据源数据
- ✓ 无实例限制
- ✓ 多维度智能统计
- ✓ 个性化定制

● 地址

SQL 社区版：<http://demo.sql.actionsky.com/>

SQL 企业版：<http://demo.sql.actionsky.com:8889/login?target=/>

● 登录用户

user	password
admin	admin

● 测试 MySQL

variable	value
host	20.20.20.3
port	3306
user	root
pass	test



注意事项:

1. 该服务仅用于在线功能体验，请勿在生产环境使用；
2. 该测试服务数据会定期清理。

2024 展望



SQL 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

社区支持

微信技术交流群

欢迎加入微信群进行交流，反馈使用中遇到的问题或者使用建议。

进群方式：添加管理员微信 ActionOpenSource

爱可生社区公众号

关注爱可生社区公众号，获取最新 SQLE 动态包括新功能清单与产品规划。



管理员微信



爱可生社区公众号

相关链接

产品官网: <https://www.actionsky.com/sqlc>

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

社区版代码库: <https://github.com/actiontech/sqlc>



Thank you