



# SQL 开源之夏 宣讲介绍

管长龙/爱可生开源社区主理人





# 目录

- 01 社区介绍  
一个有深度的数据库开源社区
- 02 什么是问题 SQL?  
我们的项目是为了解决什么问题?
- 03 什么是 SQLE?  
全方位 SQL 质量管理平台
- 04 开源之夏课题介绍  
活动导师及课题内容

# 社区介绍

一个有深度的数据库开源社区



# 一个有 深度 的数据库开源社区



- 成立于 2017 年，爱可生公司发起
- 开源多款数据库工具
- 免费提供数据库技术服务
- 输出数据库技术内容近千篇
- 助力数据库从业人员的职业发展

## 企业级数据库工具及服务

- SQLE - 全流程 SQL 质量管理平台
- DBLE - MySQL 分布式中间件
- DTLE - 数据传输组件
- .....



## 高质量的技术内容

- 内容涉及 MySQL、Redis、MongoDB、OceanBase 等国内外知名开源数据库产品
- 来自全国 150+ 位数据库专家的技术内容
- 《大智小技》系列书籍

## 丰富的社区活动

- 行业峰会、Meetup、线上直播等
- 《雍正荐书》、《主理人荐书》



# 一家受益于开源的数据库公司

## 丰富的开源数据库解决方案经验

爱可生成立于 2003 年，拥有 15 年以上的大型系统平台运维经验，位列全球 TOP10 规模的系统建设及运维经验、1000+运维项目经验。

## 国内最早一批 MySQL 解决方案服务商。

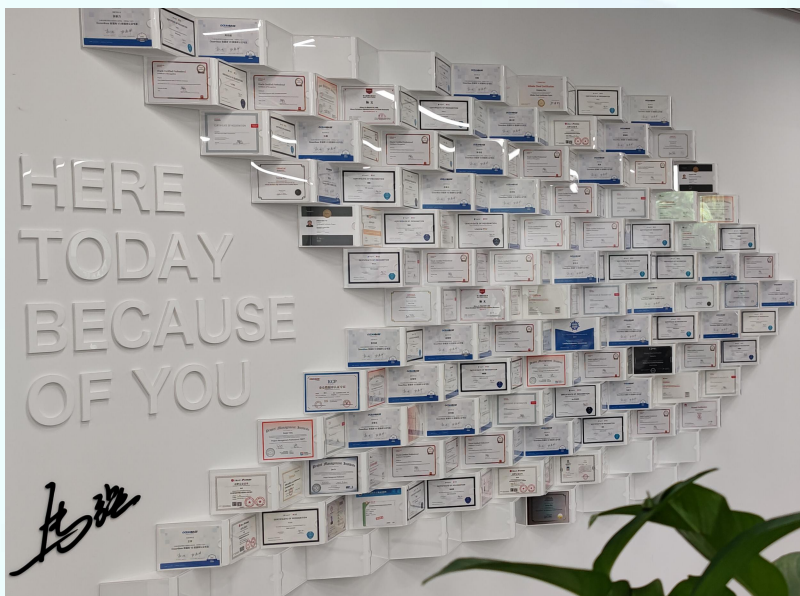
MySQL AB/SUN/Oracle 公司的白金合作伙伴。  
OceanBase 生态合作伙伴

## 专业的开源数据库专家团队

拥有 MySQL、Redis、MongoDB、TiDB、OceanBase、Oracle 等认证专家 50+

## 数据库产品

ActionDB (OceanBase商业发行版)  
云树系列数据库管理平台  
SQL 质量管理平台 SQLE  
向量数据库 TensorDB





# 什么是问题 SQL?

我们的项目是为了解决什么问题?

# 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?



# 一个案例

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;
```



# 一个案例

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

# 一个案例

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

# 什么是 SQLE?

全方位 SQL 质量管理平台



# 全流程 SQL 质量管理平台 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。



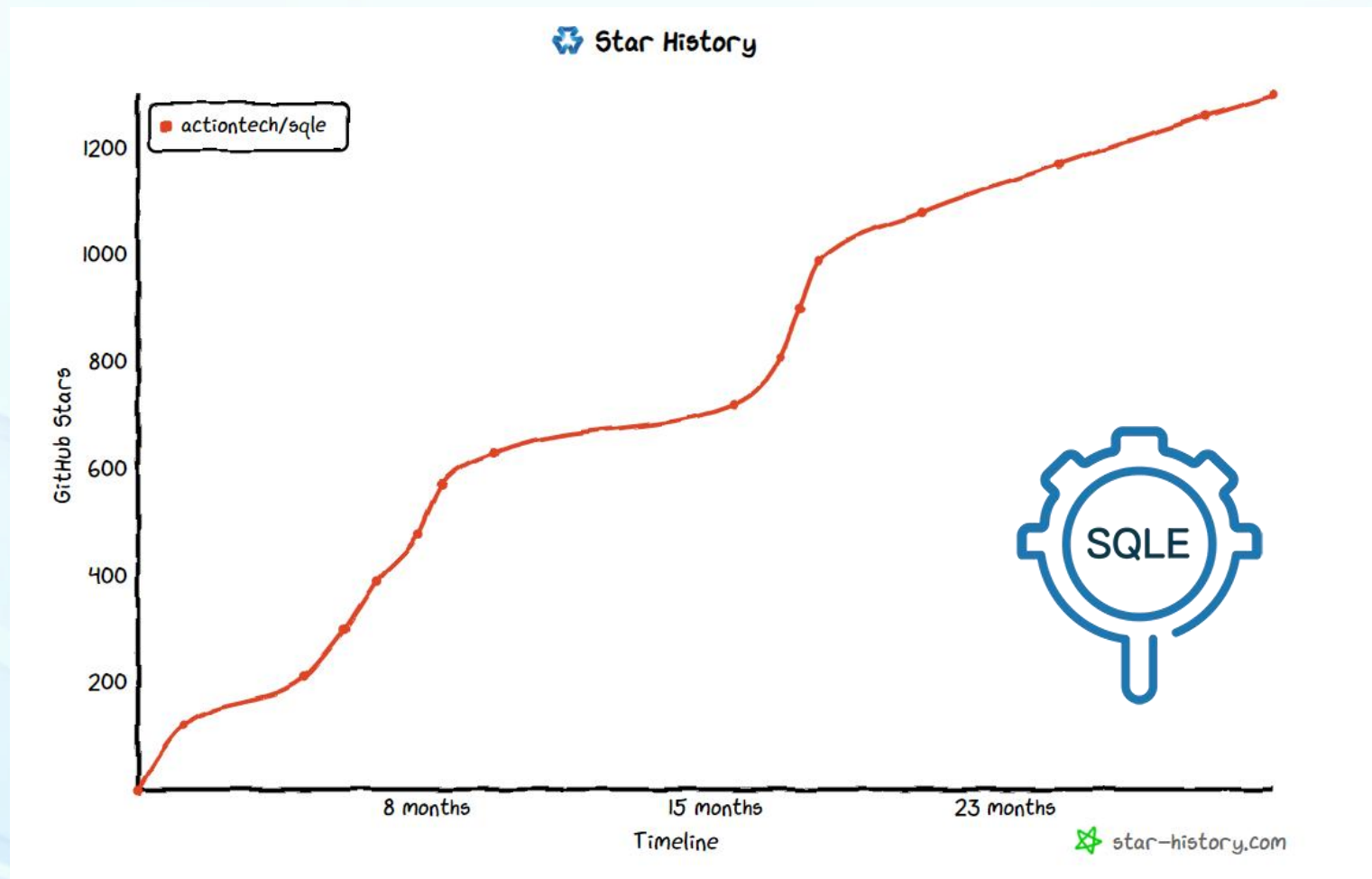
# 全流程 SQL 质量管理平台 SQLE

2021 年 10 月 24 日开源。

过去的两年多，**每月**发布版本：

- 一共发布 **27** 个正式版本
- 新增大功能 **70+**
- 提交 commits **6K+**
- 处理完成 issue&pr **1.4K+**
- 累计收获 Star **1.3K**

\* 数据统计时间 20240417




# SQLE 如何解决上面的案例

案例：表字符集不一致导致部分场景下索引失效

[← 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	<code>create table `t_utf8` ( `id` int(11) NOT NULL AUTO_INCREMENT, `name` varchar(20) DEFAULT NU...</code>		添加说明	分析
2	<code>create table `t_utf8mb4` ( `id` int(11) NOT NULL AUTO_INCREMENT, `name` varchar(20) DEFAUL...</code>		添加说明	分析
3	<code>create table `t_utf8mb4_2` ( `id` int(11) NOT NULL AUTO_INCREMENT, `name` varchar(20) DEFA...</code>		添加说明	分析
4	<code>SELECT * FROM t_utf8 as t1 LEFT JOIN t_utf8mb4 as t2 ON t1.code=t2.code WHERE t2.name ...</code>	 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	<code>SELECT * FROM t_utf8mb4_2 AS t1 LEFT JOIN t_utf8mb4 AS t2 ON t1.code=t2.code WHERE t...</code>	 审核通过	添加说明	分析

# 完善规则规范是做好 SQL 质量管理的第一步

目前支持的数据库/数据源类型

社区版

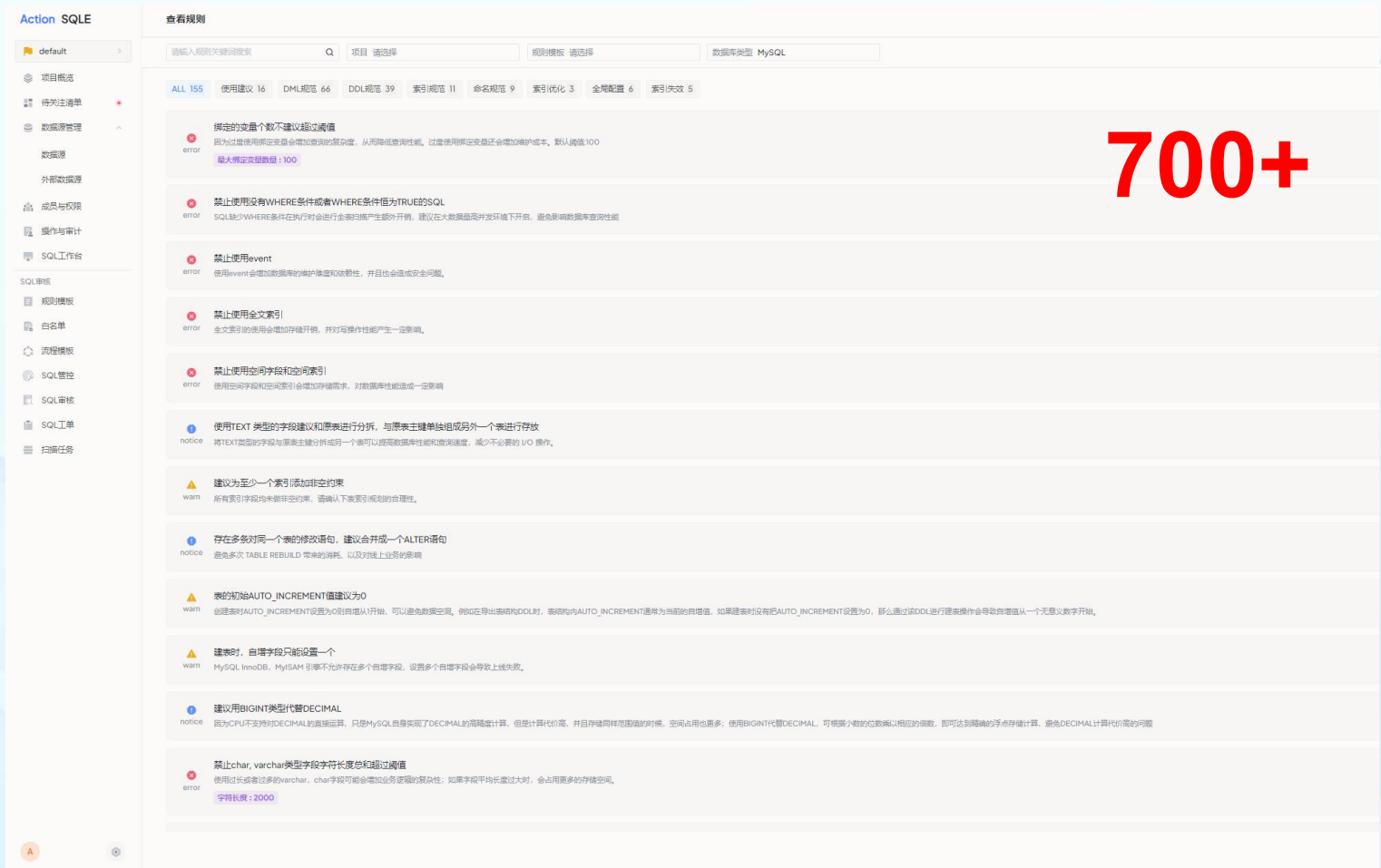
• MySQL

专业版

企业版

- MySQL
- Oracle
- SQL Server
- PostgreSQL
- DB2
- OceanBase
- TiDB
- TDSQL for InnoDB
- Mycat
- DM (达梦)
- GoldenDB
- (持续增加中)

SQL  
规则库

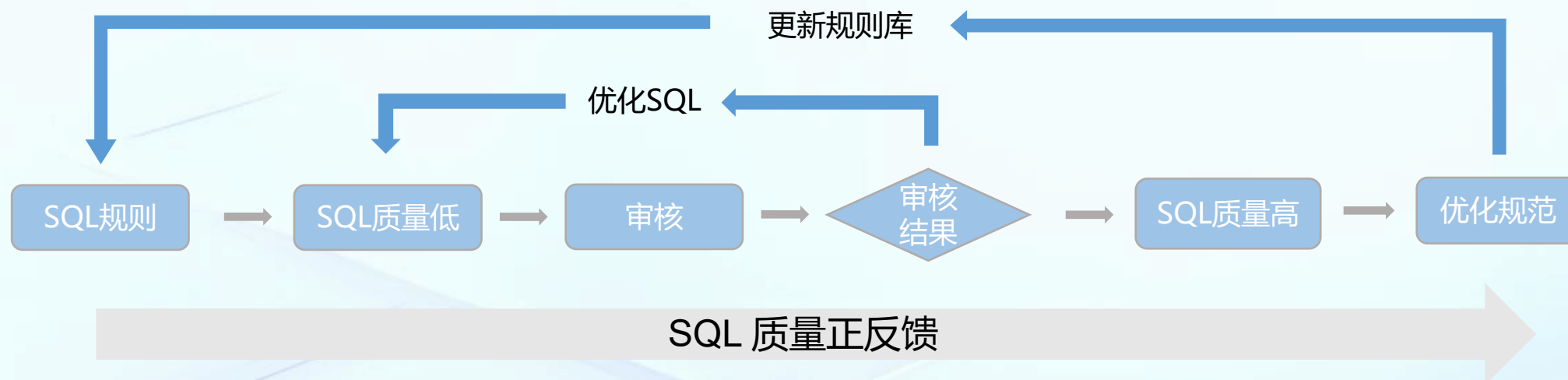


The screenshot shows the 'Action SQLC' interface for managing SQL rules. On the left is a sidebar with navigation options: '项目概述', '待关注清单', '数据源管理', '数据源', '外部数据源', '成员与权限', '操作与审计', 'SQL工作台', 'SQL审核', '规则模板', '白名单', '流程模板', 'SQL管控', 'SQL审核', 'SQL工单', and '扫描任务'. The main area is titled '查看规则' and includes a search bar, a dropdown for '项目', and a dropdown for '规则模板'. Below this, there's a summary bar showing 'ALL 155' rules, categorized by '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', and '索引失效: 5'. The list of rules includes:

- 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条件值为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。建议在大数据量库并开发环境下开启, 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复杂性, 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空间, 并对写操作性能产生一定影响。
- 禁止使用空字段和空索引** (error): 使用空字段和空索引会增加存储空间, 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拆, 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): 所有索引字段均做非空的约束, 请确认下索引规则的合理性。
- 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。例如在导出数据到DOL时, 表结构内AUTO\_INCREMENT通常会被当前的自增值。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的近似值计算, 但是计算代价高, 并且存储精度范围值的时候, 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指数, 即可达到精确的浮点存储计算, 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+

# 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:

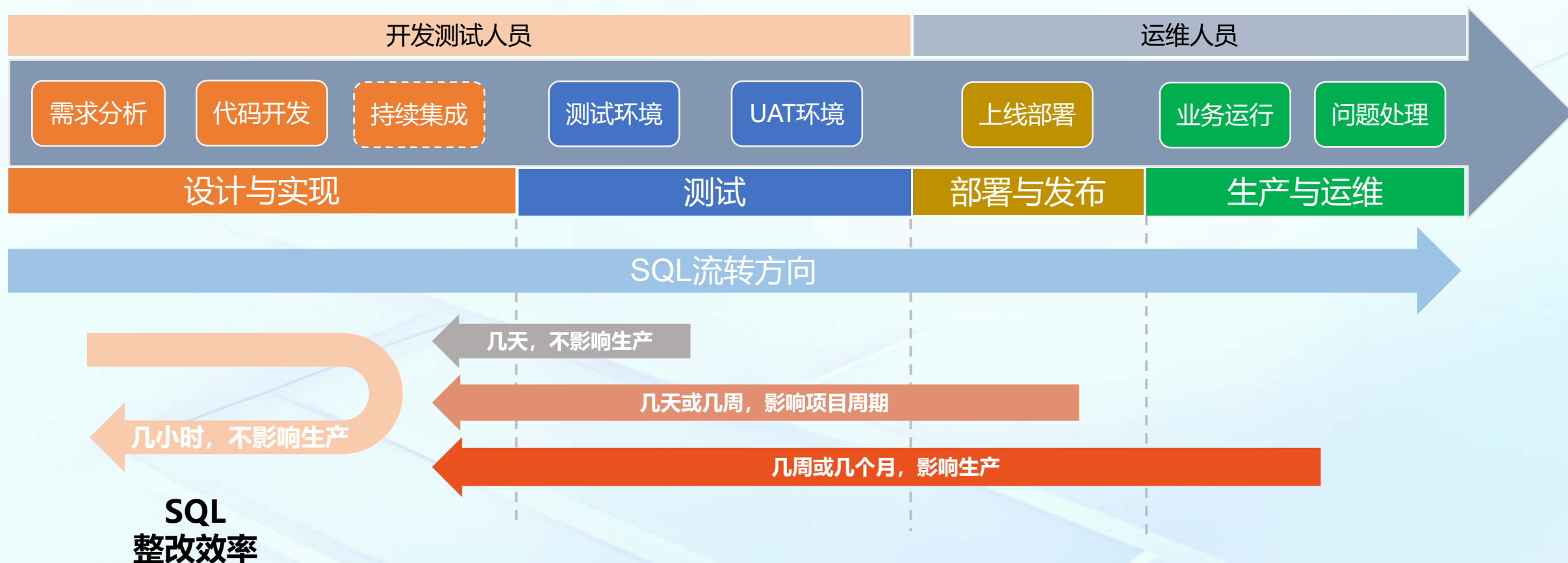
- 评分
- 审核报告
- 统计报表

## 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



# 在软件生命周期中的 SQL



问题越早解决成本越低！

# 不同阶段对应的审核手段





# SQLLE 在线体验

## ● 地址

SQLLE 社区版: <http://demo.sqlle.actionsky.com/>

SQLLE 企业版: <http://demo.sqlle.actionsky.com:8889/>

## ● 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。



# 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



# 开源之夏项目介绍

项目导师及项目介绍

# 项目介绍

## 项目名称

新增 SQL 审核规则 “MySQL 自增字段  
禁止手动插值”

## 项目支持报名语言

仅中文(请使用中文输入项目所有信息)

## 项目导师

孙健<sunjian@actionsky.com>

## 项目综合难度

基础

## 技术领域标签

• MySQL • Database • Relational Database • DevOps

## 编程语言标签

• Go • SQL

## 项目产出要求

1. 参考当前 SQLE 审核规则的设计模式完成规则的添加； 2. 参考当前 SQLE 规则测试用例完成单侧的添加； 3. 代码需接受复审并合并到 SQLE 开源仓库主分支。

## 项目技术要求

1. 熟悉 Go 语言； 2. 了解任一关系型数据库的使用； 3. 了解或简单使用过 Docker； 4. 了解编译原理（加分项）。

# 项目介绍

SQLE 是一个 SQL 质量管理工具，帮助研发规范 SQL 编写，持续集成更多 SQL 审核规则是 SQLE 的重要目标。

本次将对 MySQL 数据库新增审核规则“自增字段禁止手动插值”。  
例如：

对于如下的表格，建表语句如下：

```
1 create table test1 (  
2     id INT AUTO_INCREMENT PRIMARY KEY ,  
3     name VARCHAR(255)  
4 );
```

在数据库设计和开发中，自增主键是常见的设计模式，由数据库生成自增主键 ID。手动赋值可能会造成数据空洞，主键冲突。自动生成简化了程序设计复杂性。

Bad SQL:

```
1 insert into  
2     t1 (id, name)  
3 values  
4     (1, "tim");
```

Good SQL:

```
1 insert into  
2     t1 (name)  
3 values  
4     ("tim");
```

审核规则实现效果



# 加入社区

## 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>

# Thanks

