

Gdevops

Global DevOps Summit

全球敏捷运维峰会

银行复杂架构下的数据库敏捷运维

演讲人：王鹏冲





王鹏冲

平安银行总行科技运营中心数据库技术总监

个人简介:

16年数据库领域工作经验，历任多家公司数据库技术负责人、团队总监

Oracle 10g OCP, MySQL OCP

2015-2017连续三届PostgreSQL全国大会演讲嘉宾

2018年MongoDB中文社区演讲嘉宾

GOPS、dbaplus、Gdevops等技术社区分享讲师

微信公众号《数据库技术圈》发起人

目录

CONTENT

- 1 概论
- 2 开发
- 3 运维
- 4 展望



01

概论

Database DevOps

银行数据库架构有多复杂？

1	DB类型	10-20种	DB类型不同，人员技能、开发设计、发布部署、配置规范、运维工具、变更方案、自动化流程等，都会有差异
2	实例规模	数以万计	生产、同城、异地、开发、测试、灰度等不同环境的大量实例，带来的是运维管理的复杂度、标准化落地的难度
3	DB版本	新旧并存	新旧版本并存，对补丁管理、安全漏洞、迁移升级等都带来挑战
4	OS类型	Unix、Linux、Windows	不同平台的配置基线、采集监控、报警指标、应急处理等都要有针对性设计
5	开发模式	外购+自研	大量外购商业软件的存在，使得对DB的应用层的优化手段有限，DB重构、去IOE困难重重
6	DB架构	集中式、分布式	分布式架构的数据库运维带来与传统集中式架构不一样的需求与问题，比如审核、灰度发布、批量回滚等。



Compliant Database DevOps

Deliver value quicker while keeping your data safe

数据库怎样
融入DevOps基因

Standardize
team-based development

Protect
& preserve data

Automate
database deployments

Monitor
performance & availability





传统数据库开发、发布方式
存在的问题

Shifting views on the primary drawback of siloed development

What do you consider the **greatest** drawback in traditional, siloed database development?

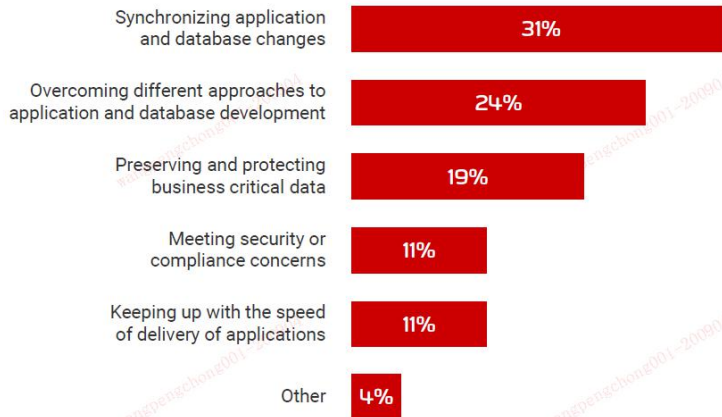




将数据库集成到
DevOps流水线的挑战

Challenges integrating database changes into a DevOps process

What do you consider the greatest challenge in integrating database changes into a DevOps process?

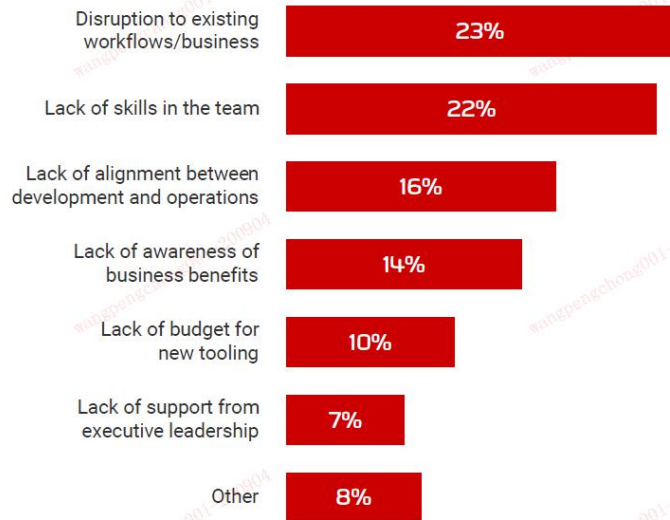




企业实施DevOps的
阻碍因素

Primary obstacles to implementing DevOps

What was or what would be the main obstacle to implementing a DevOps approach in your organization?

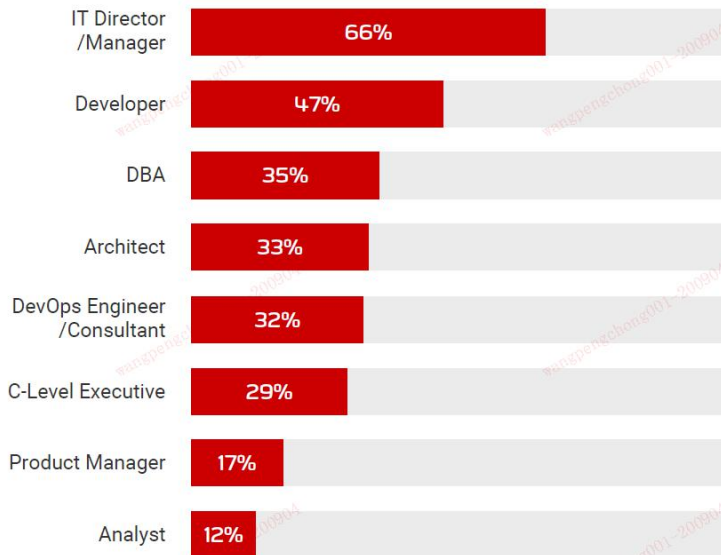




DevOps
谁说了算?

Top decision makers for DevOps implementations

Who's involved in deciding to implement DevOps in your organization?





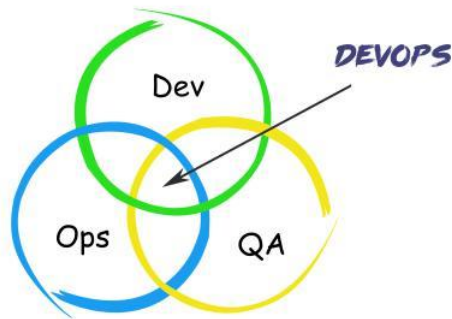
02

DevOps: 面向开发

打造涵盖数据库开发设计、代码审核、
部署测试、生产发布的一条龙生产流水线

以前的发布流程：

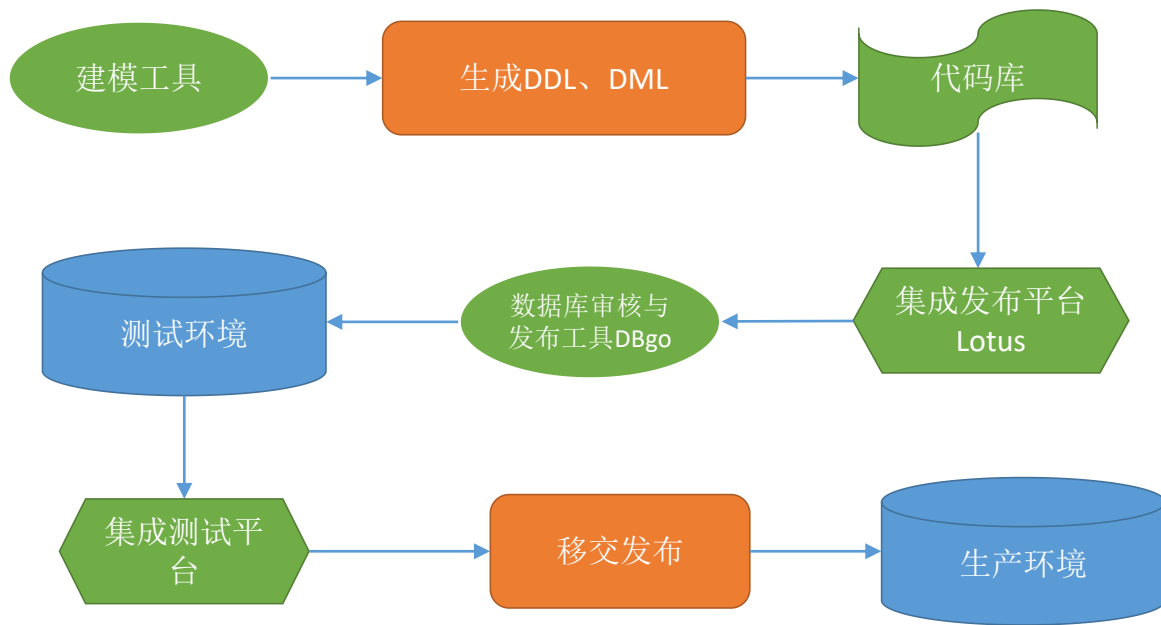
1. 开发人员将应用程序打包，并按顺序汇总并整理数据库发布脚本。线下传递脚本或缺乏审核。
2. DBA拿到数据库发布脚本检查、备份、执行，以完成数据库发版。
3. 部署人员拿到应用部署包，备份、替换，以完成应用程序发版。

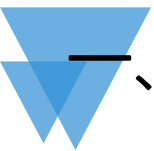


引入DevOps之后的发布流程：

1. 开发人员将应用程序、DB代码在CI工具中完成上传、代码审核、打包。
2. 部署人员拿到部署任务，在发布系统中完成DB发版以及应用包部署。

数据库制品流水线





一、数据建模工具-DDM

问题

缺少统一工具支持

建模与数据标准脱节

数据模型无法共享



目标

快速建模

- 支持逻辑模型和物理模型的可视化建模，快速编辑表，字段，主外键索引，视图等对象，支持模型全生命周期管理

数据标准落地

- 与数据标准库、词根库拉通，可实现数据标准智能发现及数据标准引用，提升建模质量。

模型共享

- 模型可以跨系统、团队共享，支持多方参与数据标准的定义，制定业务规则。

设计驱动开发

- 先设计、再开发，根据设计结果生成数据库部署脚本
- 确保数据库开发规范落地



DDL的现状和问题

当前DDL现状

手工书写

使用excel建模, 手工书写DDL;

DDL校验放在开发环节之后

即将转测试或者即将发版时进行DDL的校验;

DDL相关的工具或平台没有互通

数据库相关的模型设计, 脚本编写, 脚本上传, 脚本执行分别要在多个工具之间进行切换 (Excel/文版编辑器/gitbash/dbgo平台/邮件/PLSQL等)。

导致的问题

质量问题、效率问题

质量参差不齐, 手写效率低;

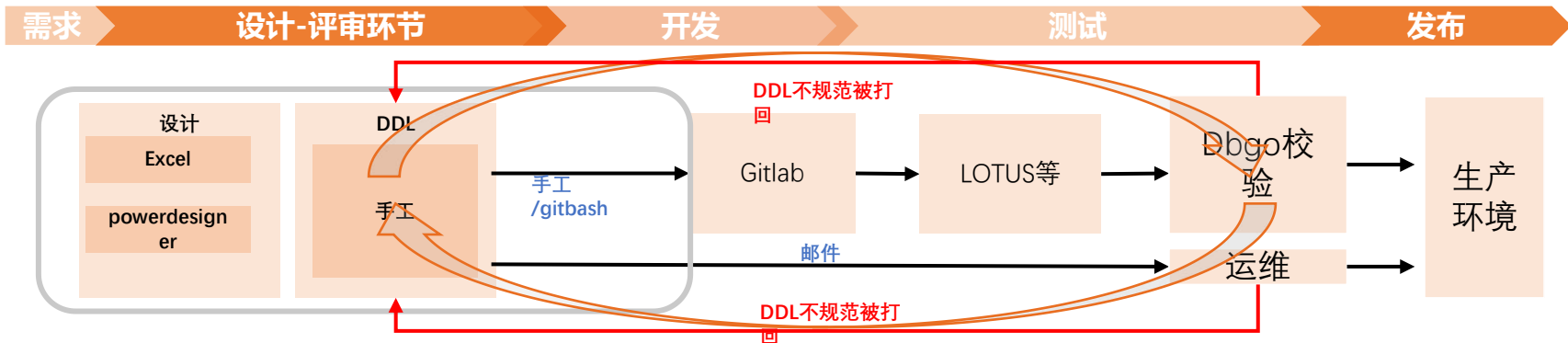
返工风险高, 风险后置

即将转测试或者即将发版时进行校验才发现问题, 一旦发现问题就要返工, 如果涉及代码修改, 返工成本更高;

多工具切换, 效率低

用户的DDL的完成及校验部署需要在多个平台/或工具中切换, 使用复杂, 效率低, 多工具维护复杂, 实时性差。

DDL当前主要流程

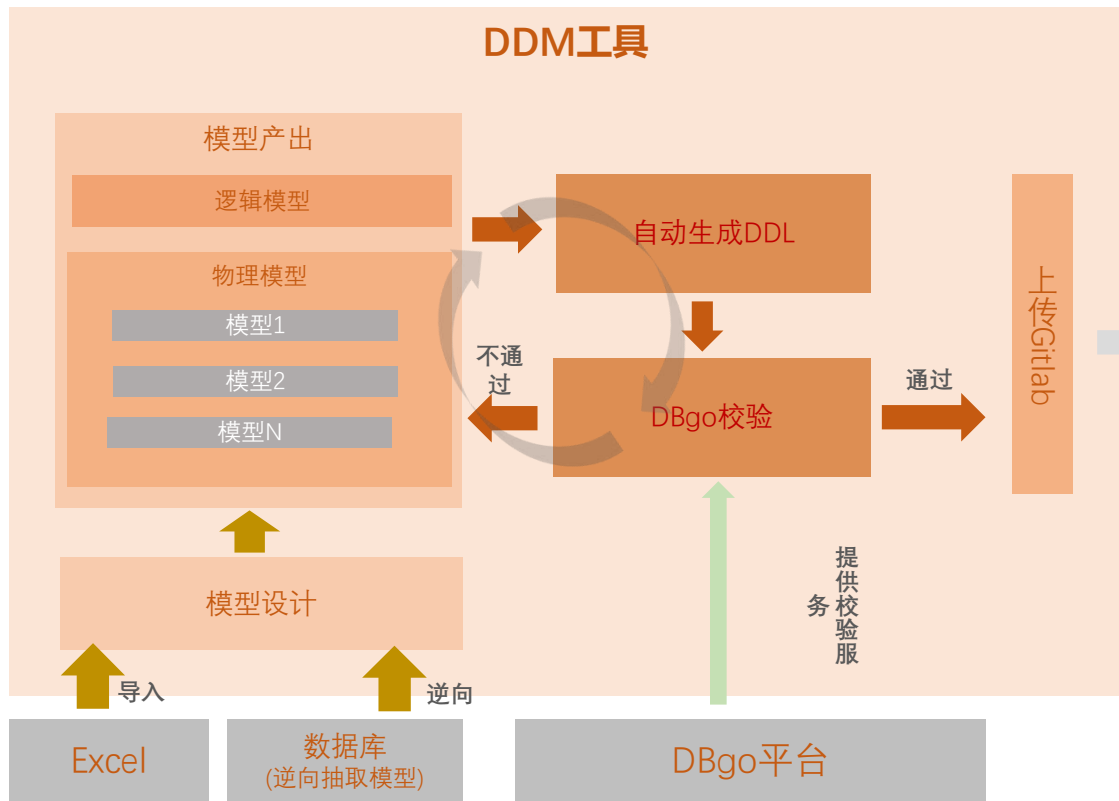




DDM工具方案及收益

需求 → 设计环节 → 开发 → 测试 → 发布

一站式工具



收益

批量生成，高效高质量

工具批量自动一键生成，避免人为疏忽导致的问题。提高效率，提高质量；

提前感知风险，提前解决

与Dbgo平台对接，将DDL规则校验前置，提前察觉问题，提前解决问题；并且达到实时校验实时修改，提高脚本质量，降低投产风险；

一站式开发，减少开发成本

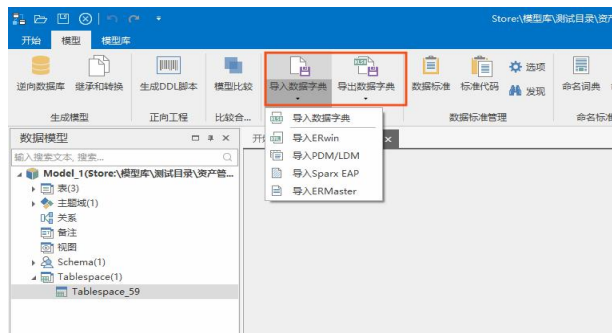
建模，校验，修改，上传Git库，部署（部分开发环境及测试环境）全部都在DDM工具内一站式完成，提高开发效率，降低开发成本。



DDM工具-1 快速建模

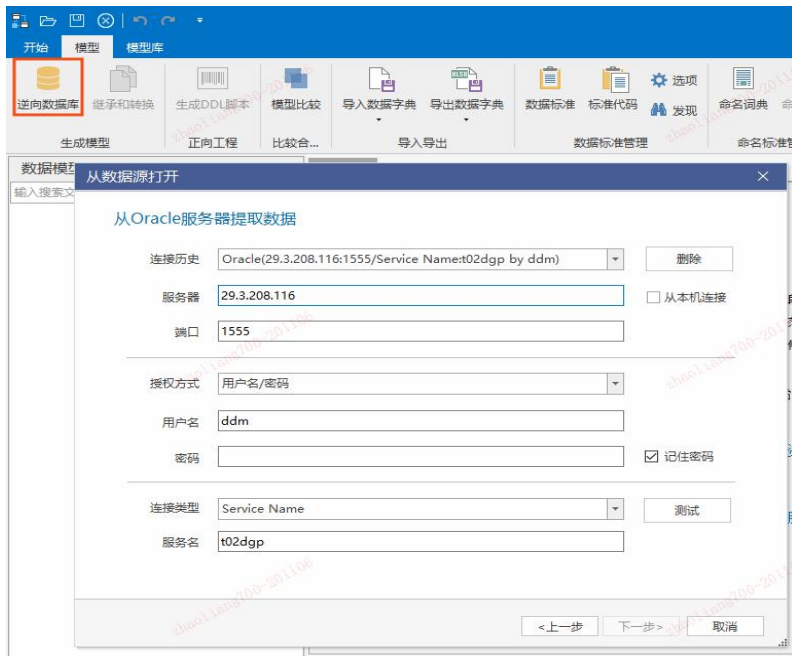
1. Excel导入

- ① 导出模板；
- ② 完善模板；
- ③ 导入数据字典；



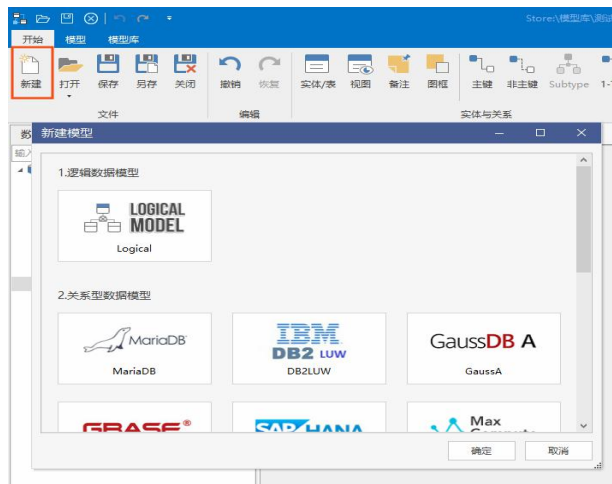
3. 逆向工程

- ① 逆向；
- ② 输入数据库连接串；
- ③ 选择表；



2. 手工新建

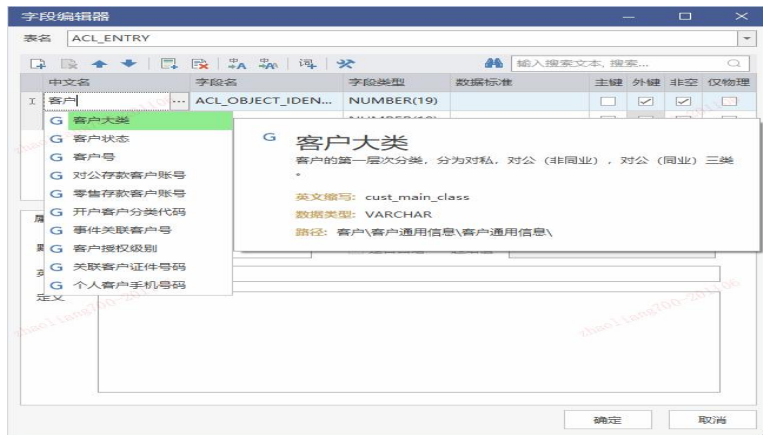
- ① 新建；
- ② 选择数据库类型；





DDM工具-2 模型落标

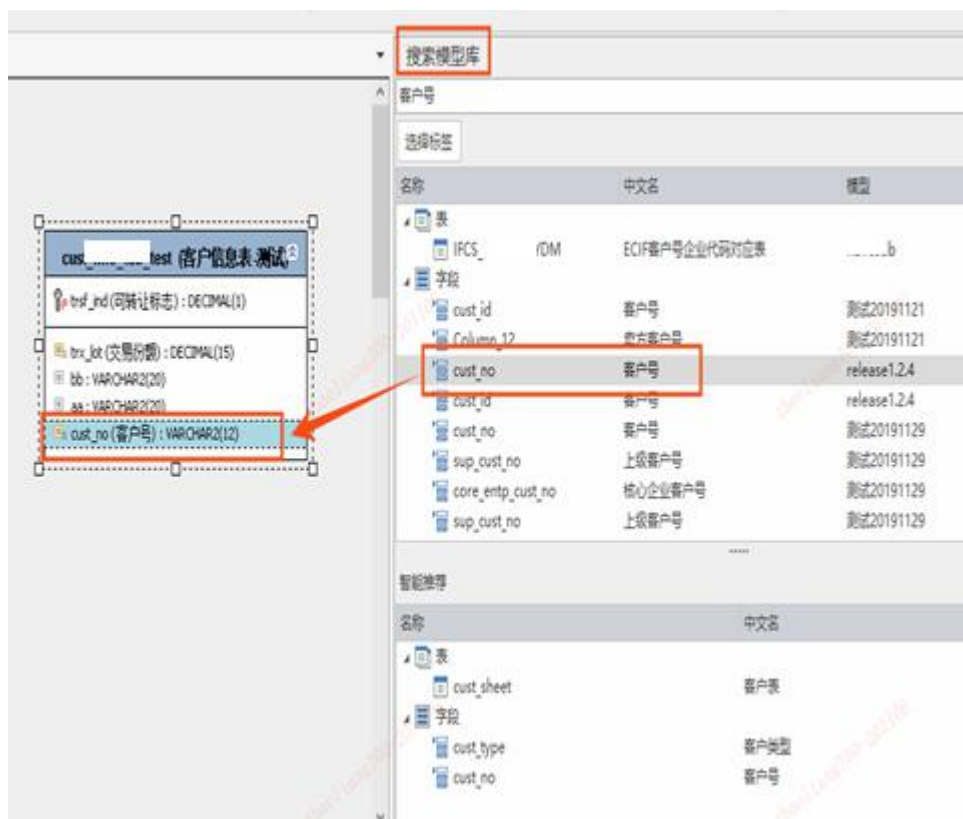
1. 直接引用内置标准库的数据标准



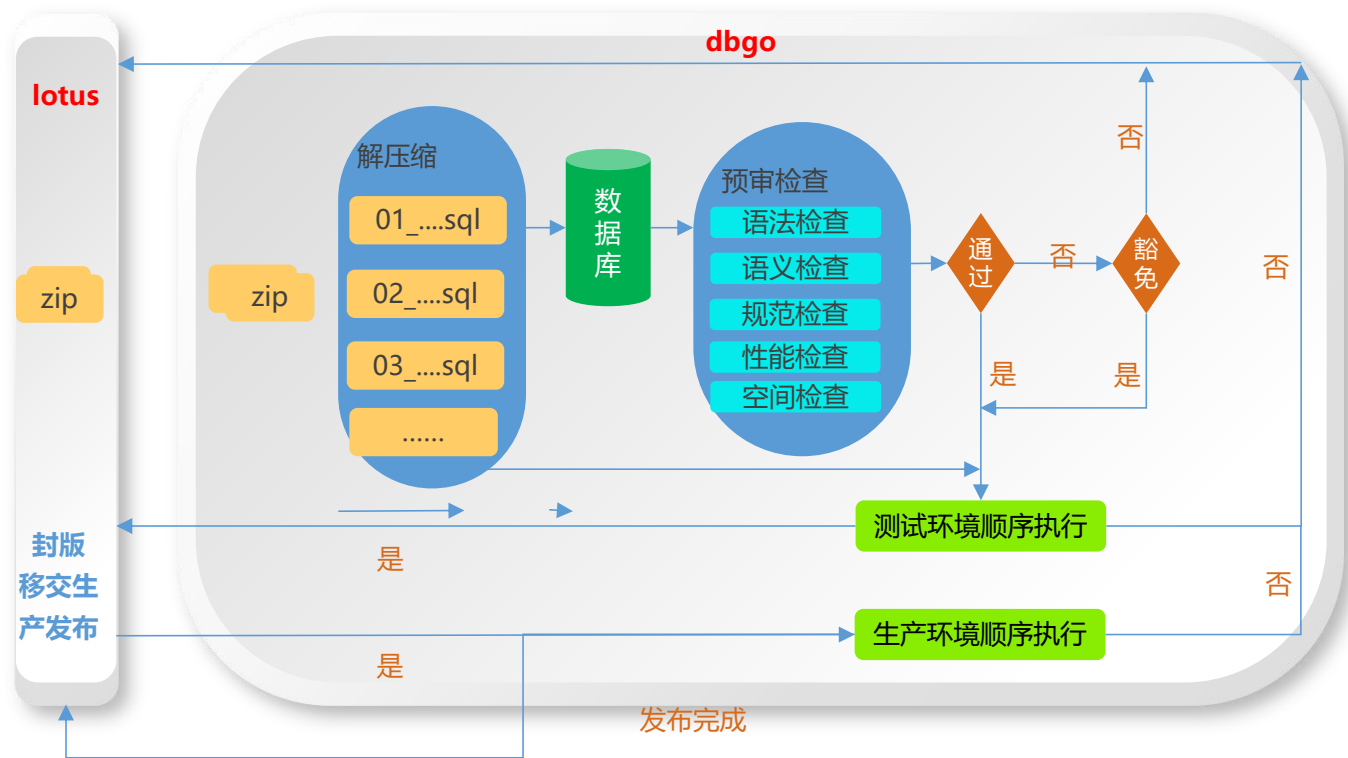
2. 自动使用内置词根进行翻译



3. 直接使用共享模型



二、DB发布工具总览

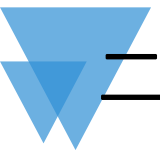


二、数据库发布工具DBgo (1)

背景说明

随着银行业务高速发展，越来越多的系统的上线，数据库版本发布的局限性越来越突出，风险越来越高，构建一个高可用、自动化、安全可控、规则可扩展的数据库脚本发布平台越来越重要。





二、数据库发布工具DBgo (2)

建设目标

01

高可用

平台7*24小时稳定运行

打造7*24小时高可用的自动化部署平台。

02

自动化

任务提交时设定执行时间，到点自动化执行

版本任务可编排，若出现任务按照时间窗口自动执行问题，自动智能回滚。

03

安全可控

用户只可查看或执行与授权的菜单或操作

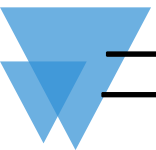
根据权限最小化原则，适当分配各种角色用户的权限，严格控制权限，杜绝安全隐患。

04

可扩展

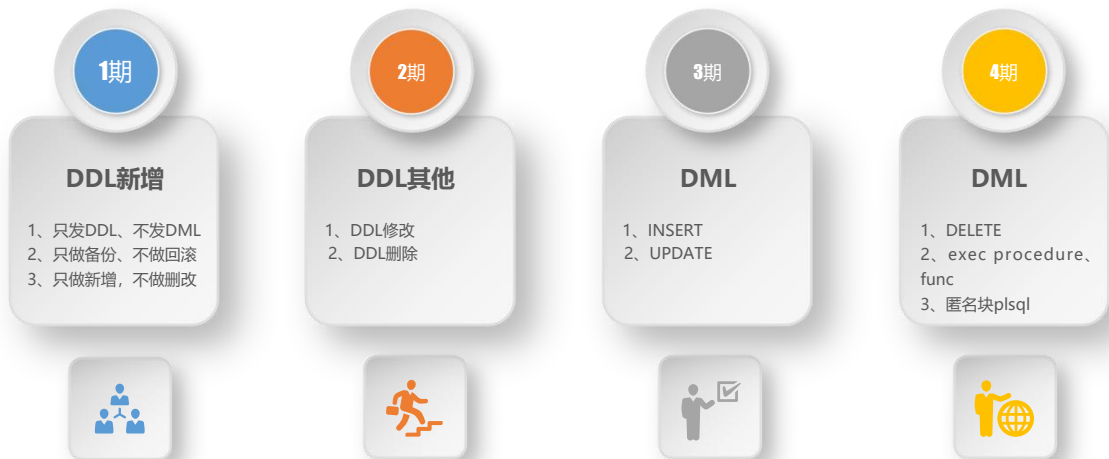
可以根据需要添加或修改响应检验规则

SQL解析规则引擎的规则，可以根据需要动态添加、修改、删除，方便查询。

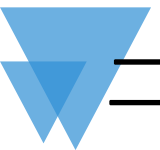


二、数据库发布工具DBgo (3)

分步实现



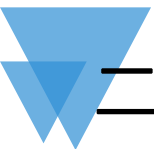
- 1、只发DDL，以及简单的DML
- 2、只做备份，不执行回滚，所有回滚操作都是触发通知到人工判断、执行。
- 3、一期只做新增，不做删、改（比如只做create,或alert * add; 不做alter * drop、alter * modify、rename、drop、truncate)



二、数据库发布工具DBgo (4)

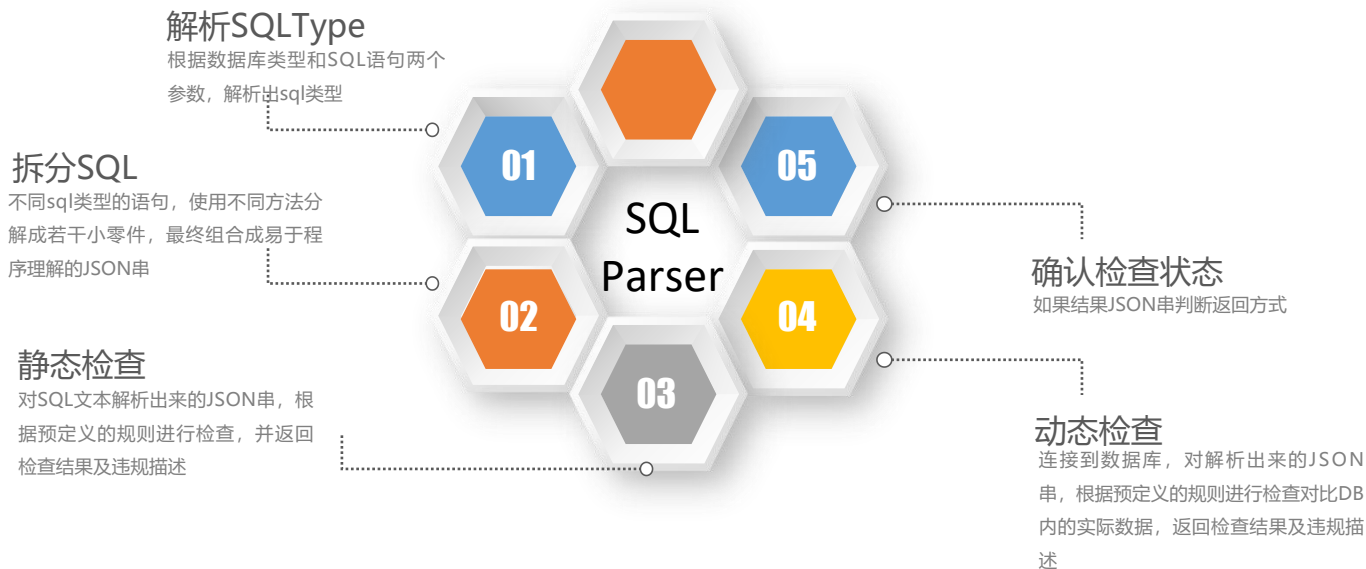
设计要点





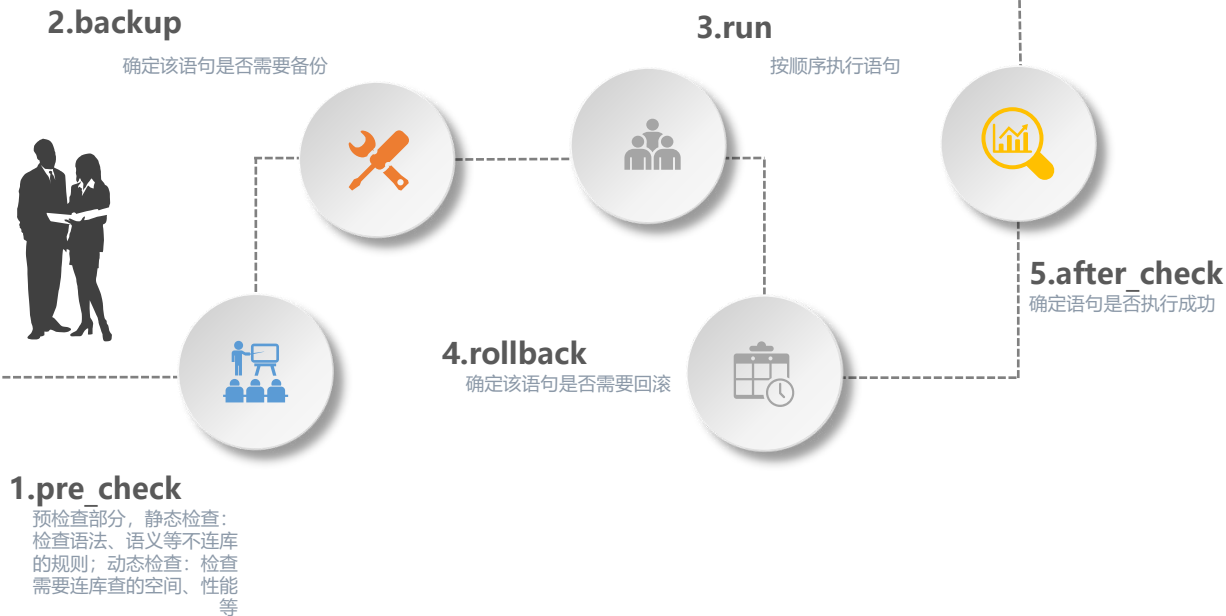
二、数据库发布工具DBgo (5)

工具原理



二、数据库发布工具DBgo (6)

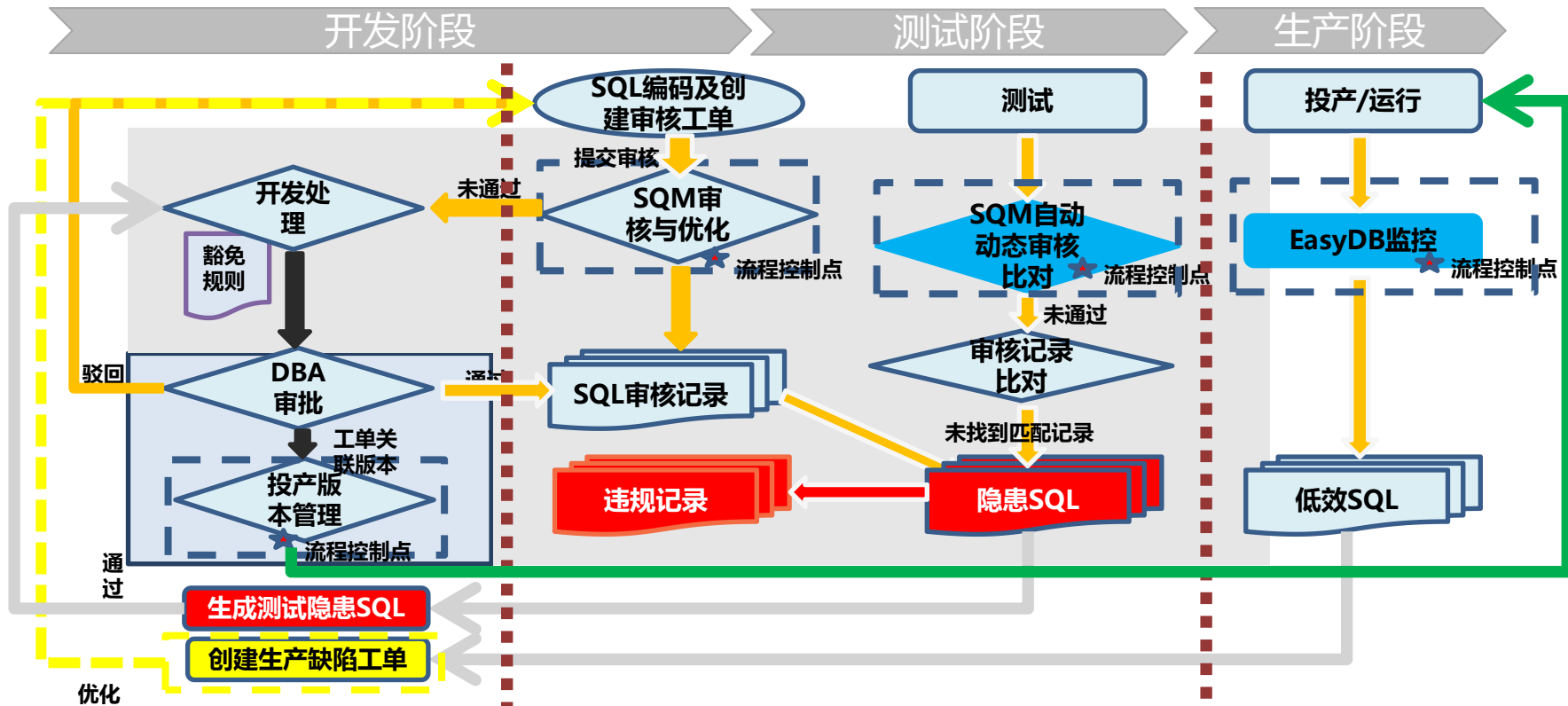
工具流程



二、数据库发布工具DBgo (7)

oracle	create_table	static	precheck	1 create_as_exist	检查create table as select...	直接reject,由dba审核
oracle	create_table	static	precheck	2 owner_exist	检查表名前面是否有属主前缀 CREATE TABLE UOPDATA_T_ABC	若无,则reject
oracle	create_table	static	precheck	3 name_lowerline_exist	检查表名是否至少包含一个_(规范)	若无,则reject
oracle	create_table	static	precheck	4 key_exist	检查主键是否存在 PRIMARY KEY(A),ALTER TABLE ADD CONSTRAINT PRIMRY KEY	如不包含,则reject, 可请dba审核是否豁免
oracle	create_table	dynamic	precheck	5 name_exist_indb	检查库表、同义词、视图或物化视图在库级别是否存在	审核规则样例 对于表: 1、本owner下存在,就reject 2、其他owner下,则提示,进入dba审核 对于同义词、视图或物化视图,存在则reject
oracle	create_table	static	precheck	6 tablespace_exist	检查是否有tablespace语句, 不可包含tablespace	
oracle	create_table	static	precheck	7 initrans6_exist	检查是否有initrans 6	如不是, 则reject
oracle	create_table	static	precheck	8 varchar_over3col	检查多余3个字段以上所有字段varchar/varchar2字段是否一样长	如是, 则reject
oracle	create_table	static	precheck	9 4keepcol_exist	是否包含create_date,create_by,update_date,crete_by字段	若不包含, 则提示
oracle	create_table	static	precheck	10 colnum_over100	字段数量是否超过100	超过, 提示
oracle	create_table	static	precheck	11 varcol_over2w	varchar或varchar2总长度是否超过20000	如超过, 提示
oracle	create_table	static	precheck	12 bloblong_exist	判断是否包含blob,long类型	如包含, 则reject
oracle	create_table	static	precheck	13 clob_over2	clob>=2	如超过2个, 则reject, 可请dba审核是否豁免
oracle	create_table	static	precheck	14 foreign_uniq_check_const_exist	判断是否包含外键、唯一、检查约束	如包含, 则reject,请dba申请是否豁免
oracle	create_table	static	precheck	15 default_null_exist	判断是否包含default " 和default null	如包含, 则reject,请dba申请是否豁免
oracle	create_table	static	precheck	16 if_pk_notnull	判断primary key是否定义成not null	如未定义, 则reject,可请dba审核是否豁免
oracle	create_table	static	precheck	17 pk_name_prefix	主键约束命名规则: pk_表名	如是命名的primary key constraint. 若名称不以pk_开头, 则reject
oracle	create_table	static	precheck	18 range_contain_interval	若是范围分区, 且分区字段是date或timestamp类型, 则必须包含interval,	若无interval, 则reject
oracle	create_table	static	precheck	19 list_contain_default	若是list分区, 则必须包含default分区	若不包含, 则reject
oracle	create_table	static	precheck	20 over_128	若是hash分区, 则必须是2的n次方幂, 且不超过128个	若非n次方, 或超过128个, 则reject
oracle	create_table	static	precheck	21 notnull_set	分区列必须定义成not null	若非, 则reject

三、SQL代码审核流程集成



三、SQM审核规则举例

workflows

audit rules

custom fields

project members

data sources

Oracle

MySQL

审核对象: ☒ 不限 ☐ SQL ☐ DDL

影响程度: ☒ 严重 ☒ 警告 ☒ 提示

审核规则: 使用%模糊搜索

搜索

新增

删除

<input type="checkbox"/>	编号	规则名	规则描述	审核对象	影响程度 (双击修改)	建议	操作
<input type="checkbox"/>	1	select SQL中禁止使用select *	select SQL中禁止使用select *	SQL文本	警告	查看	
<input type="checkbox"/>	2	禁止 DML 语句的 where 条...	禁止 DML 语句的 where 条件中包含恒真...	SQL文本	警告	查看	
<input type="checkbox"/>	3	update / delete SQL中禁止...	update / delete SQL中禁止使用limit子句	SQL文本	警告	查看	
<input type="checkbox"/>	4	update /delete SQL中禁止使...	update /delete SQL中禁止使用order by...	SQL文本	警告	查看	
<input type="checkbox"/>	5	update/delete SQL中必须包...	update/delete SQL中必须包含where子句	SQL文本	严重	查看	
<input type="checkbox"/>	6	where条件中禁止在列上做运...	where条件中禁止在列上做运算或函数操作	SQL文本	严重	查看	
<input type="checkbox"/>	7	where条件中必须使用主键列...	where条件中必须使用主键列或者索引列	SQL文本	警告	查看	
<input type="checkbox"/>	8	insert SQL中必须明确指定插...	insert SQL中必须明确指定插入数据的列...	SQL文本	警告	查看	
<input type="checkbox"/>	9	表join时缺少join条件	表join时必须有join条件	SQL文本	严重	查看	
<input type="checkbox"/>	10	like使用只允许使用右模糊, ...	like使用只允许使用右模糊, 不允许使用...	SQL文本	警告	查看	
<input type="checkbox"/>	11	where条件中or表达式左右两...	where条件中or表达式左右两侧包含的列...	SQL文本	警告	查看	

三、SQM审核工具：支持多种格式的SQL导入

任务卡头信息：任务应用-4, ID: 121-01

操作按钮：修改, 备注, 分配, 删除, 推送DLM, 提交系统审核, 关闭, 完成

详情

类型: Oracle推送 状态: 开放

优先级: 高 分析状态: 新建

数据库信息: cf - 8:1521/oradb

版本号: AB-NTS5.12.1

描述

C !1-01

SQL列表

导入SQL: 添加SQL, 导入Excel, 导入脚本, 导入MyBatis XML文件, 导入MyBatis ZIP文件

创建好工单后，导入将要审核的SQL。可用直接添加SQL语句、excel文件格式、SQL *.sql 文件格式、MyBatis xml格式和MyBatis zip压缩包文件格式导入，可参考模板格式。

三、SQL审核工具：隐患SQL监控

deepplus

概

风险

执行

执行

SQL文本

详情

17个-执行计划里存在低效回表filter

12个-执行计划中存在全表扫描步骤

9个-查询语句缺少where条件

4个-使用select *

4个-查询条件与列类型不一致

2个-在查询列上使用表别名

1个-执行计划中存在索引全扫

返回顶部↑

file:///D:/SQ 其他/风险SQL报告_25 9120 937/index.html

风险SQL清单

基本信息

数据库: 21 12 操作人: 张... 导出时间: 20 16时49分

风险分类

审核规则	规则说明	风险SQL (个)
执行计划里存在低效回表filter 【数据库审核】	低效回表filter指: 索引扫描评估的数据行远大于回表filter的数据行	17
执行计划中存在全表扫描步骤 【数据库审核】	对大表进行全表扫描影响SQL性能, 对小表做高频扫描会影响IO系统	12
查询语句缺少where条件	查询语句缺少where条件, 说明代码中可能存在逻辑缺陷	9
使用select *	使用select *影响SQL解析效率	4
查询条件与列类型不一致 【数据库审核】	在查询列上使用表达式、函数调用, 或者发生隐式转换等	4
在查询列上使用表达式	在查询列上使用表达式, 可能会导致无法使用列上的索引, 影响SQL性能	2
执行计划中存在索引全扫步骤 【数据库审核】	对大表进行索引全扫会影响SQL性能	1

风险SQL

SQL ID	执行用户	上线时间	所属应用	执行次数	平均执行时间	SQL文本
fgrg	i5h	DATA 20	18日 11时21分	--	111275 次	14 毫秒 查看
12n	np67	DATA 20	19日 14时27分	--	2963698 次	0 毫秒 查看
6yg	i7qm	DATA 20	18日 11时21分	--	111127 次	85 毫秒 查看
g6rj	u27	DATA 20	10日 09时08分	--	15622 次	20 毫秒 查看
3hb	mt00	DATA 20	19日 14时25分	--	773472 次	20 毫秒 查看
bmi	ipxtm	DATA 20	19日 14时27分	--	11338 次	40 毫秒 查看
22w	mpp5	DATA 20	19日 14时25分	--	2028 次	22 毫秒 查看
890	5whn	DATA 20	19日 15时47分	--	207 次	4 毫秒 查看
8f1c	gbw	DATA 20	17日 15时29分	--	245 次	40 毫秒 查看

取消选择

物理读 143次

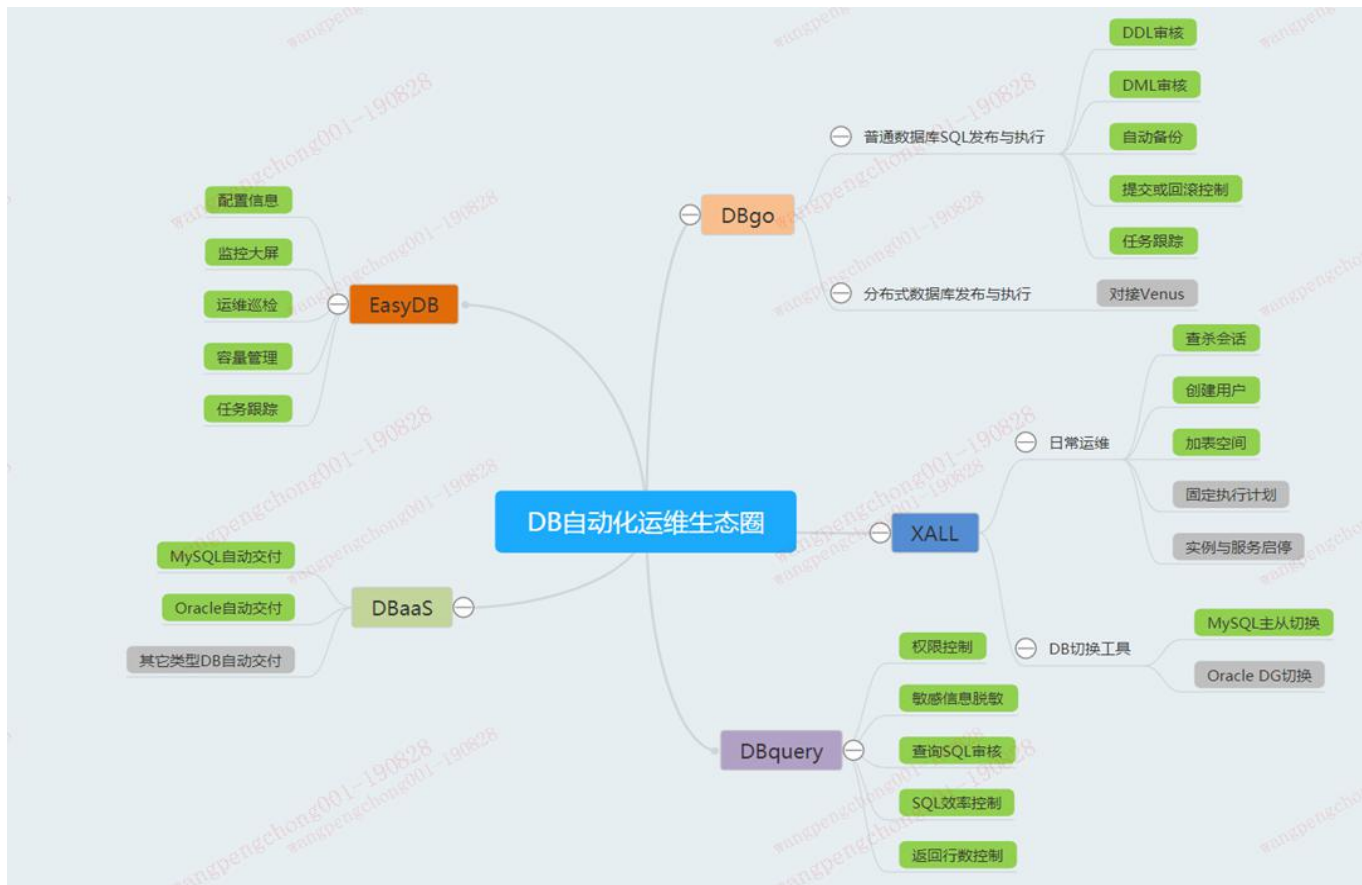
共 1 条

03

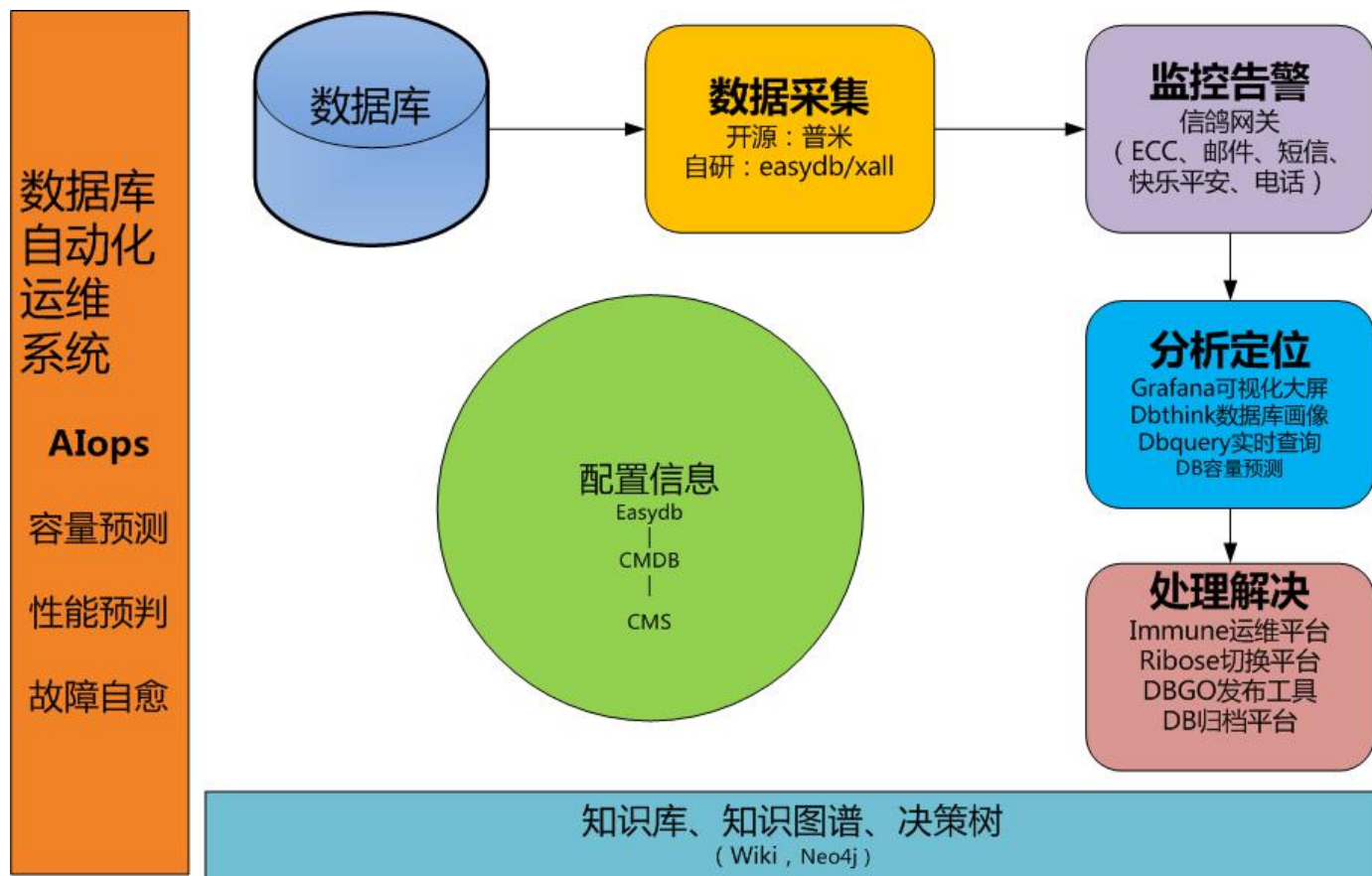
DevOps: 面向运维

打造融合不同DB类型的一站式数据库自动化运维平台

DB自动化运维工具生态圈



DB自动化运维数据架构图





1, 配置信息、监控报警

配置信息是基石, 标准化是基础
监控报警的重点是如何避免遗漏、如何收敛

4, 一站式查询DB

10几种数据库、数千名开发运维人员, 如何满足他们访问DB的日常需求



2、自动化运维

海量数据库的运维, 靠工具

3、自愈-Alops

状态类、容量类的报警, 实现自愈

监控大屏

Easydb

首页

监控大屏

大屏-生产(全部)

大屏-(只含报警)

大屏-(报警/维护)

大屏-生产(开发使用)

大屏-生产(TDSQL)

大屏-同城DG

大屏-远程DG

大屏-测试/开发

配置管理

监控报表

数据库管理监控平台

采集批次: 2021-09-18 10:24

数据库实时监控

搜索

--全部--

搜索

批量维护

服务名	DB类型	IP	端口	OS	子系统	子系统名	环境	连通性	主从	连接数	会话	cpu	维护	违规项
p101	ora		1521	solaris	PROD	PROD	ADG	正常	主	10	10	10	否	
p102	ora		1521	solaris	PROD	PROD	ADG	正常	主	10	10	10	否	
p103	my		3306	linux	PROD	PROD	生产	正常	主	10	10	10	否	
q104	mg		2018	linux	PROD	PROD	生产	正常	主	10	10	10	否	
p1011	ora		1521	linux	PROD	PROD	平台项目	正常	主	10	10	10	否	3
ab1012	my		3306	linux	PROD	PROD	提示系统	正常	主	10	10	10	否	
ab1013	my		3306	linux	PROD	PROD	提示系统	正常	主	10	10	10	否	
ab1014	ora		1521	aix	PROD	PROD	平台	正常	主	10	10	10	否	3
ab1015	ora		1521	aix	PROD	PROD	平台	正常	主	10	10	10	否	
ab1016	ora		1521	aix	PROD	PROD	平台	正常	主	10	10	10	否	
ab1017	ora		1521	aix	PROD	PROD	平台	正常	主	10	10	10	否	
ab1018	my		3306	linux	PROD	PROD	系统	正常	主	10	10	10	否	
ab1019	my		3306	linux	PROD	PROD	系统	正常	主	10	10	10	否	
ab1020	mg		2018	linux	PROD	PROD	平台	正常	主	10	10	10	否	

连通性实时数据

采集频率: 3分钟

	严重	正常	未采集
mysql	1	8	4
redis	0	8	5
tdsql	0	2	1
mongodb	0	1	3
sqlserver	0	1	0
postgre	0	1	0
tidb	0	1	0
hana	0	1	0
db2	0	1	0
app	0	1	9
neo4j	0	1	0
cassandra	0	1	0
kdb	0	1	0
influxdb	0	1	0

敏捷入口

数据库实时监控

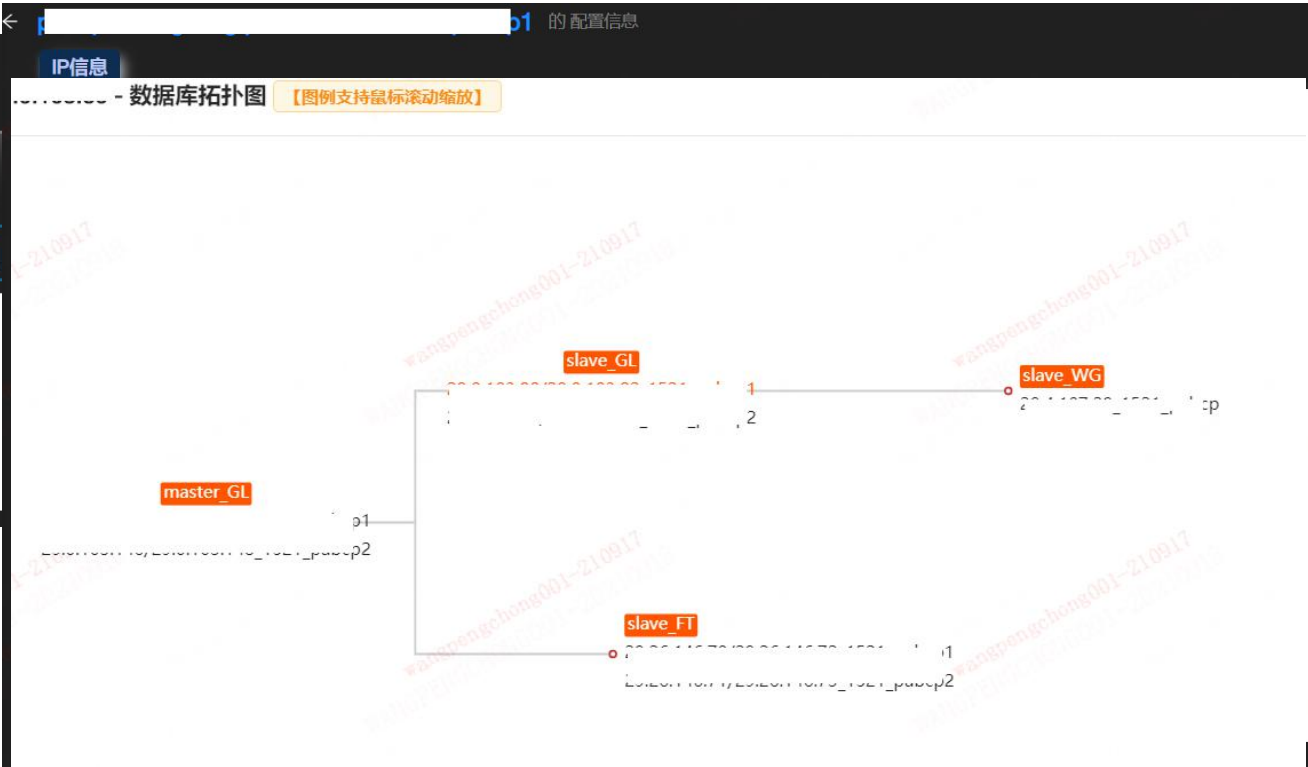
服务名 DB类型

pa 1 ora

pa 2 ora

s my

ql er mg





1, 配置信息、监控报警

配置信息是基石, 标准化是基础
监控报警的重点是如何避免遗漏、如何收敛

4, 一站式查询DB

10几种数据库、数千名开发运维人员, 如何满足他们访问DB的日常需求



2、自动化运维

海量数据库的运维, 靠工具

3、自愈-Alops

状态类、容量类的报警, 实现自愈





Ribose

🏠 首页

🏢 切换场景

MySQL MHA 切换

ORACLE ADG 切换

📅 操作日志

📋 我的任务

🔗 常用链接

🚪 退出

Ribose-MYSQL MHA切换

🔔 📧

1-搜索对象

2-选择系统

3-选择切换模式

4-切换确认

5-结果确认

✔待切换系统

id	clusterid	ip		vip		port	is_master	status	s	cname	arecord		
940	3306	1	1	1	3	3306	1	1	1	2	1	None	online_change failover
941	3306	1	2			3306	0	1	1	2		None	online_change failover

提示:

is_master表示当前的主备状态

Ribose © 2019 Created by DBA



1, 配置信息、监控报警

配置信息是基石, 标准化是基础
监控报警的重点是如何避免遗漏、如何收敛

4, 一站式查询DB

10几种数据库、数千名开发运维人员, 如何满足他们访问DB的日常需求



2、自动化运维

海量数据库的运维, 靠工具

3、自愈-Alops

状态类、容量类的报警, 实现自愈

AI Ops的一小步

背景:

- ✓ 目前对于日常告警的分析及消除, 仍需要人工介入, 存在夜间及节假日可能处理不及时、工具使用不熟练等人为原因造成处理延迟或引发生产故障的风险。
- ✓ 为进一步深化自动化运维, 并逐步向AI Ops转型, 本项目在DBA已积累的日常运维经验知识库基础上, 探索并实践基于决策树模型的故障自愈解决方案。
- ✓ 本项目的主要目标是通过机器学习的方式, 基于运维知识库, 迭代生成数据库常见故障或告警的原因分析及告警消除的决策树, 并通过调用一系列现有的集成工具接口实现故障的自愈操作以及告警的自我消除。

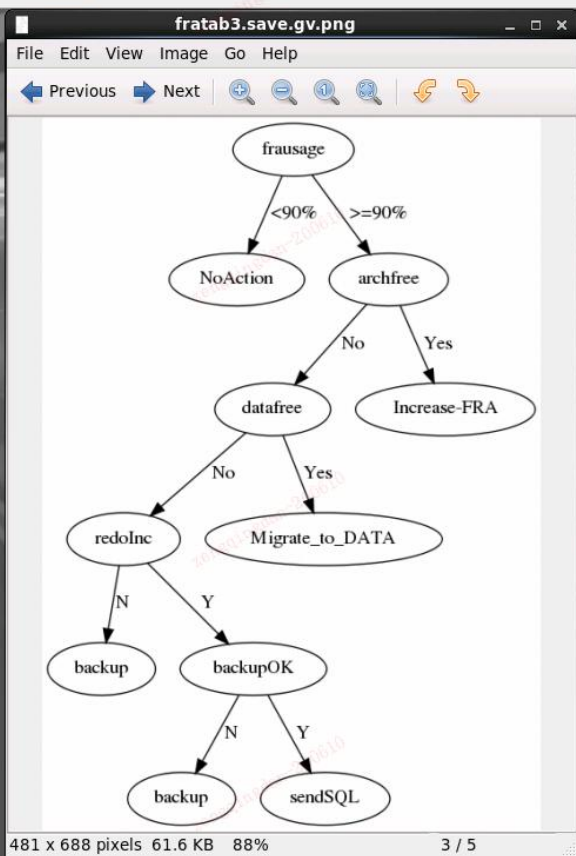
主要过程:

- 1) 根据运维知识库的内容梳理形成常见故障的决策表;
- 2) 导入决策表并通过机器学习迭代生成决策树;
- 3) 对接kafka告警工作流, 获取对应类型的紧急告警, 触发自愈系统;
- 4) 通过对接一系列实时接口获取系统运行状态, 组合生成测试集通过决策树获取解决方案及辅助根因分析;
- 5) 对接变更流程及自动化工具接口, 对告警对象进行干预, 实现告警消除。

以Oracle数据库FRA区的空间使用率报警为例

FRA使用率	ARCH是否有空余空间	DATA是否有空余空间	备份是否成功	是否有大量数据更新	策略
<90%	Yes	No	Y	Y	NoAction
<90%	No	No	Y	N	NoAction
<90%	No	Yes	Y	Y	NoAction
>=90%	Yes	Yes	N	N	Increase-FRA & 调起备份
>=90%	No	Yes	Y	Y	Migrate_to_DATA & 发送具体SQL
>=90%	Yes	Yes	N	N	Increase-FRA
>=90%	No	No	Y	Y	调起备份 & 发送具体SQL

```
oracle@-----:~/aiops
File Edit View Search Terminal Help
8 >=90% No No N Y backup
9 >=90% No No Y Y sendSQL
['NoAction'] [3]
['Increase-FRA' 'Migrate_to_DATA' 'backup' 'sendSQL'] [2 1 3 1]
['Migrate_to_DATA' 'backup' 'sendSQL'] [1 3 1]
['backup' 'sendSQL'] [3 1]
['backup'] [2]
['backup' 'sendSQL'] [1 1]
['backup'] [1]
['sendSQL'] [1]
['Migrate_to_DATA'] [1]
['Increase-FRA'] [2]
决策树:
{'frausage': {'<90%': 'NoAction',
                  '>=90%': {'archfree': {'No': {'datafree': {'No': {'redoInc': {'N':
                  'backup',
                  'Y':
                  {'backupOK': {'N': 'backup',
                  'Y': 'sendSQL'}}}},
                  'Yes': 'Migrate_to_DATA'
                  }},
                  'No': {'datafree': {'No': 'redoInc',
                  'Yes': 'Migrate_to_DATA'
                  }},
                  'Yes': 'Increase-FRA'}}}}}
```





IMMUNE

首页

首页

自愈dashboard

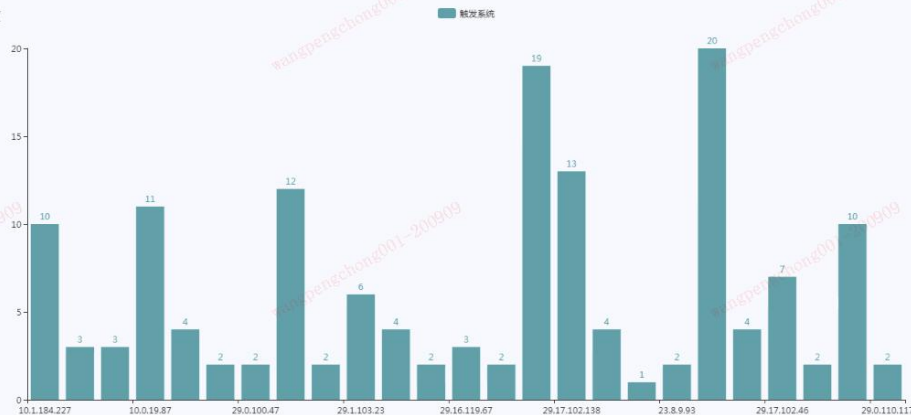
运维场景

操作日志

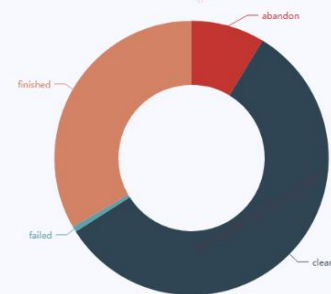
常用链接

退出

系统触发次数



状态分布



每天触发量



IMMUNE

首页

自愈合dashboard

运维场景

操作日志

常用链接

退出

自愈记录

id	listno	intime	status	info	ip	extcol1
157	0000004050	2020-09-09 01:22:24	clear	实际值 75.04 小于参考值 83	1	8-11
156	0000004050	2020-09-09 00:45:51	clear	实际值 82.03 小于参考值 83	1	8-11
155	0000002708	2020-09-08 02:03:19	clear	实际值 65.62 小于参考值 83		
154	0000002708	2020-09-08 01:25:49	clear	实际值 82.02 小于参考值 83		
153	0000000612	2020-09-07 04:35:24	clear	实际值 82.2 小于参考值 83	1	7
152	0000000612	2020-09-07 04:29:46	finished	实际值 83.63 本次采取策略:Increase-FRA 完成 alter system set db_recovery_file_dest_size=565G时间2020-09-07 04:30:01 现有值81.44	1	7
151	0000000612	2020-09-07 04:24:09	clear	实际值 82.74 小于参考值 83	1	7
150	0000000612	2020-09-07 04:18:33	finished	实际值 84.15 本次采取策略:Increase-FRA 完成 alter system set db_recovery_file_dest_size=550G时间2020-09-07 04:18:40 现有值81.91	1	7
149	0000000612	2020-09-07 04:07:20	clear	实际值 81.79 小于参考值 83	1	7
148	0000000612	2020-09-07 04:01:43	finished	实际值 83.45 本次采取策略:Increase-FRA 完成 alter system set db_recovery_file_dest_size=535G时间2020-09-07 04:01:59 现有值81.11	1	7
147	0000000612	2020-09-07 03:50:28	clear	实际值 82.26 小于参考值 83	1	7



1, 配置信息、监控报警

配置信息是基石, 标准化是基础
监控报警的重点是如何避免遗漏、如何收敛

4, 一站式查询DB

10几种数据库、数千名开发运维人员, 如何满足他们访问DB的日常需求



2、自动化运维

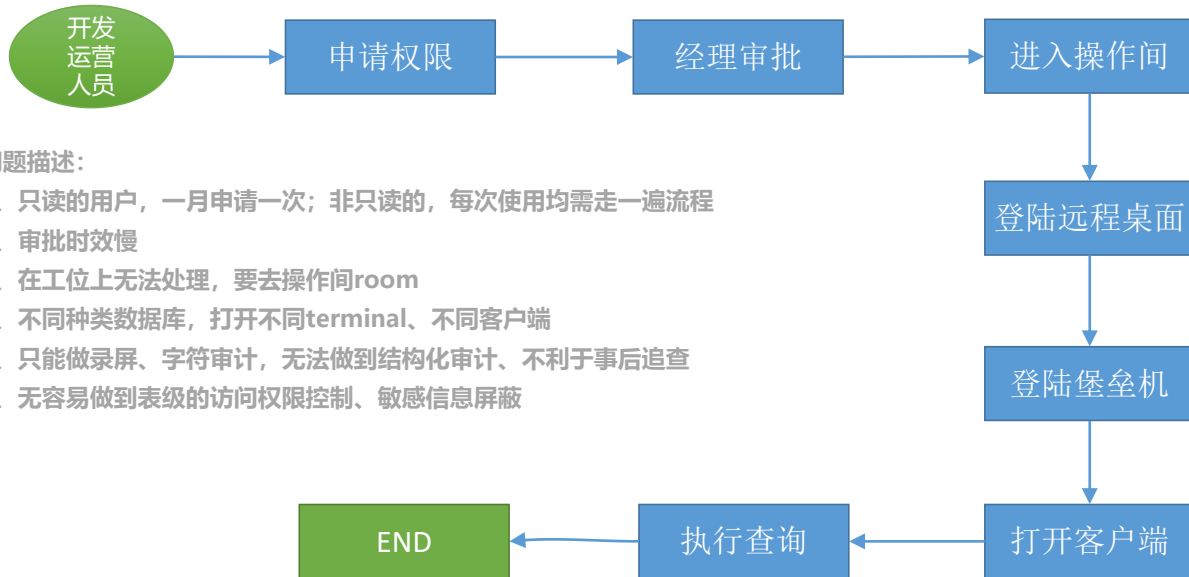
海量数据库的运维, 靠工具

3、自愈-Alops

状态类、容量类的报警, 实现自愈

1

问题背景



问题描述:

- 1、只读的用户，一月申请一次；非只读的，每次使用均需走一遍流程
- 2、审批时效慢
- 3、在工位上无法处理，要去操作间room
- 4、不同种类数据库，打开不同terminal、不同客户端
- 5、只能做录屏、字符审计，无法做到结构化审计、不利于事后追查
- 6、无容易做到表级的访问权限控制、敏感信息屏蔽



2

功能需求

多种类型的库快速切换查询



避免反复提单的工作量



随时查询以方便工作支持



安全审计、操作日志、行为分析



便于分析统计操作趋势等



避免低效sql影响生产性能



一站式的工具



三 查询请加属主，新申请库表权限时，请到TSM搜索dbquery或dbgo看索引提单，地址：ht

您好



首页 / SQL查询 / 选择DB类型

在此选择需要执行的数据库类型 (灰色背景部分尚未开放) :

ORACLE®



TDSQL





查询请加备注，新申请库表权限时，请到ITSM搜索dbquery或dbgo看指引提单，地址：<http://10.10.10.10:8080>

您好 v

首页 / SQL查询 / 选择DB类型 / 查询Oracle

1、选择数据库

设置项目

查询

格式化

Explain

中断

资源窗口

收藏夹

系统对象

表清单

DBf

T_Q

DBf

T_S

DBf

S

DBf

DBf

DBf

DBf

DBf

DBf

DBf

DBf

DBf

DBf

DBf

DBf

DBf

DBf

5、查看表结构、索引信息

3、执行查询SQL

数据库: 2020/09/10 11:15:58; 共 100 行被选择; 耗时 0.049 秒; 执行时间 2020/09/10 11:15:58; 状态 执行成功

RNO	ID	IP	VIP1	DBTYPE	DBPORT	DBINSTAN
1	A3		9	oracle		s1
2	A3	15E05	5	oracle		112
3	A3	5E05	9	oracle		112

4、查看查询结果

2、阅读说明

注意事项

- 1、查询时请选中SQL语句再执行。
- 2、不允许执行非查询(SELECT)类操作，如INSERT、UPDATE、DELETE、TRUNCATE、DROP、CREATE、ALTER等。
- 3、单次查询默认返回记录条数为100条。
- 4、查询对象需带库主前缀。
- 5、SQL语句长度默认限制在3000字符以内



☰ 查询请加备注，新申请表权限时，请到ITSM搜索dbquery或dbgo看指引提单，地址：http

🔗 ? 🗑 您好 wai

首页 / DashBoard

今日访问量
13120次



访问总量：4062128次

今日SQL查询量
12719次



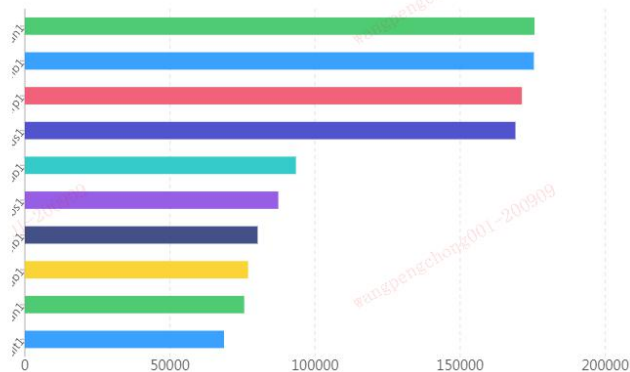
SQL查询总量：3769176次

今日表访问总量
2302张

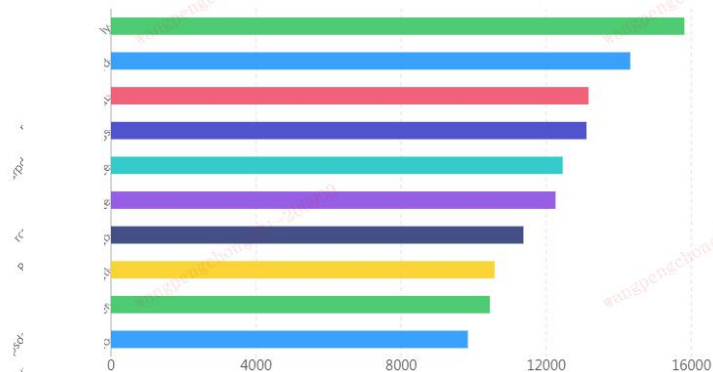


表访问总量：85651张

数据库访问统计



表访问量统计





04

展望

AI + DevOps

1. 运维不做背锅侠

- 变更要有回滚方案
- 生产要有应急预案
- 容量要有预测机制
- 工具要有防呆机制

2. 开发是友军

- 践行DevOps不是一句口号
- 运维要更多、更密切地跟开发并肩战斗





全球敏捷运维峰会

THANK YOU !

