

数据库DevOps最佳实践

周振兴

NineData联合创始人 & 技术副总裁



关于我：周振兴



2021~至今

联合创始人 & 技术副总裁



2009~2021

资深数据库专家（曾用花名“苏普”）

PolarDB 产品管理总监

淘宝DBA，去IOE核心DBA



2021~至今

Oracle ACE（MySQL方向）



2013、2022

《高性能MySQL》

第三、四版译者

个人博客：<https://www.orczhou.com>



“三多” VS “一少”

数据库种类/数量多

2022年10月，DB-Engines上共计397数据库



- 分布式数据库技术/产品快速发展
- 国产化进展加速
- 实时分析型数据库发展迅速
- 向量化发展迅速
- 变更稳定性管理变得困难、工具平台繁多

云/IDC环境多



- 不同厂商的云原生数据库不同
- 网络方案也会变得复杂

开发人员多



- 数据库变更发布变得频繁
- 团队/业务多元化



DBA 人数少

* <https://octoverse.github.com/2022/>

数据库研发协同挑战



老板说这周四
就要发布



老板说再出稳
定性问题就卷
铺盖走人

- 没权限
- 不了解规范
- SQL开发
- 需求紧急
- 流程慢
- 用户体验差
- Java
- mybatis
- Python

我要发布数据库DDL!
我要导出生产数据做分析!
我要订正线上Bug导致的错误数据!
我要导出电话号码做营销!
我的数据库为什么运行这么慢?
为什么一定要有ID主键!
.....

- SQL审核
- 性能优化
- 审计日志
- 变更窗口
- 备份恢复
- 规范审核
- 故障排查
- 监控告警



开发者



DBA/OPS

如何防止“删库跑路”、“误操作”

无意：误操作 / Bug

有心：删库跑路

```
UPDATE TABLE t SET c=1 WHERE ...
```

```
DROP DATABASE order_and_pay;
```



.....

是谁/什么时候做的操作?

如何快速恢复?

以后如何避免? 能不能事前阻断?

当前数据库开发工具的挑战

传统解决方案



云厂商产品



直连数据库带来数据安全 隐患与业务稳定性

- 离职与转岗，无法便捷的回收权限
- 执行低效率 SQL，带来稳定性问题
- 数据库账号难以管理，带来数据泄露问题



数据库变更随意 影响业务稳定性

- 无法保障核心数据库稳定，影响业务
- 所有的变更都走 DBA，影响业务发展效率
- 生产环境变更凌晨执行，DBA 效率较低



库/表结构定义混乱 影响业务可扩展性

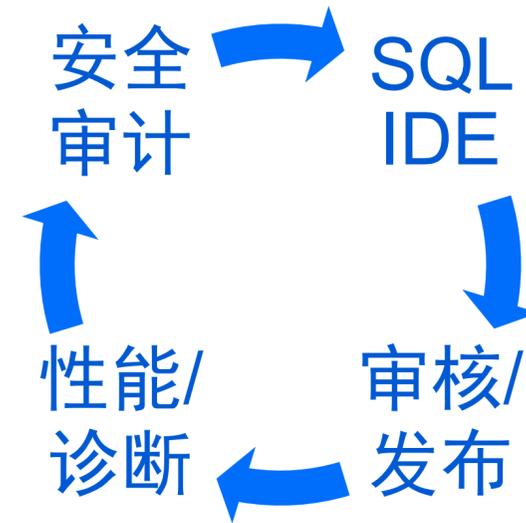
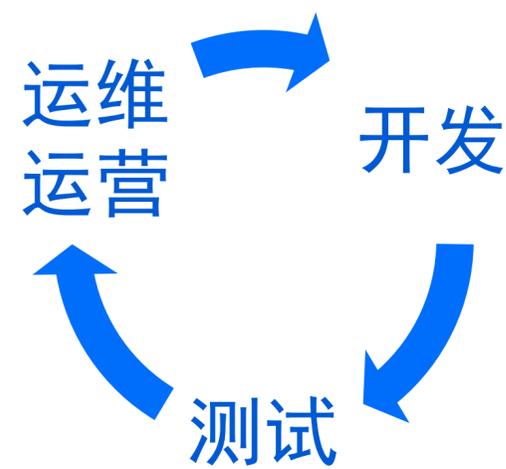
- 字段定义不规范，扩展性较低
- 索引定义混乱，影响性能与业务
- 缺乏注释等，影响扩展性



数据库种类、平台多样， 无法统一管理

- 多种不同的数据库，工具混杂
- 多云、多 IDC 访问管理复杂
- RDMBS、NoSQL 工具不一致

DBA/产品：赋能研发，让研发具备DBA能力



数据库 Devops, 端到端数据库开发与治理解决方案

高效
协作

SQL 开发规范

审批流程

版本管理

CI/CD 集成

Development

数据库设计

数据查询

数据变更

导入导出

SQL 诊断优化

Test

离线 SQL 审核

在线 SQL 审核

全量 SQL 审核

增量 SQL 审核

Operations

变更

历史数据归档

数据追踪

慢 SQL 治理

数据
安全

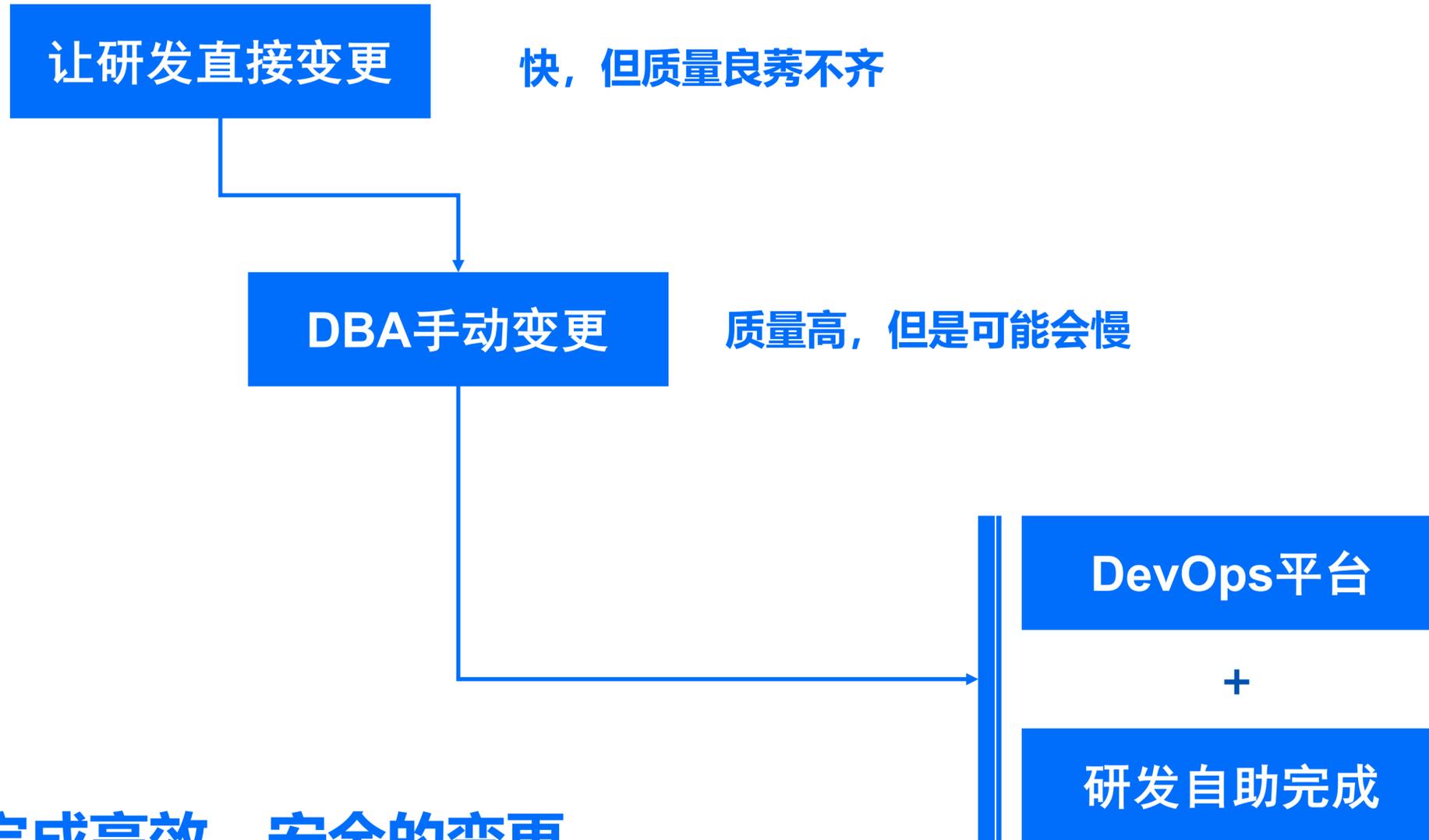
细粒度权限管控

安全规范约束

敏感数据保护

操作审计

赋能研发，让研发具备DBA能力



如何完成高效、安全的变更



结构规范

- 字符集默认utf8mb4
- 默认自增主键
- 默认字段定义为not null
- INT 默认带unsigned

索引合理

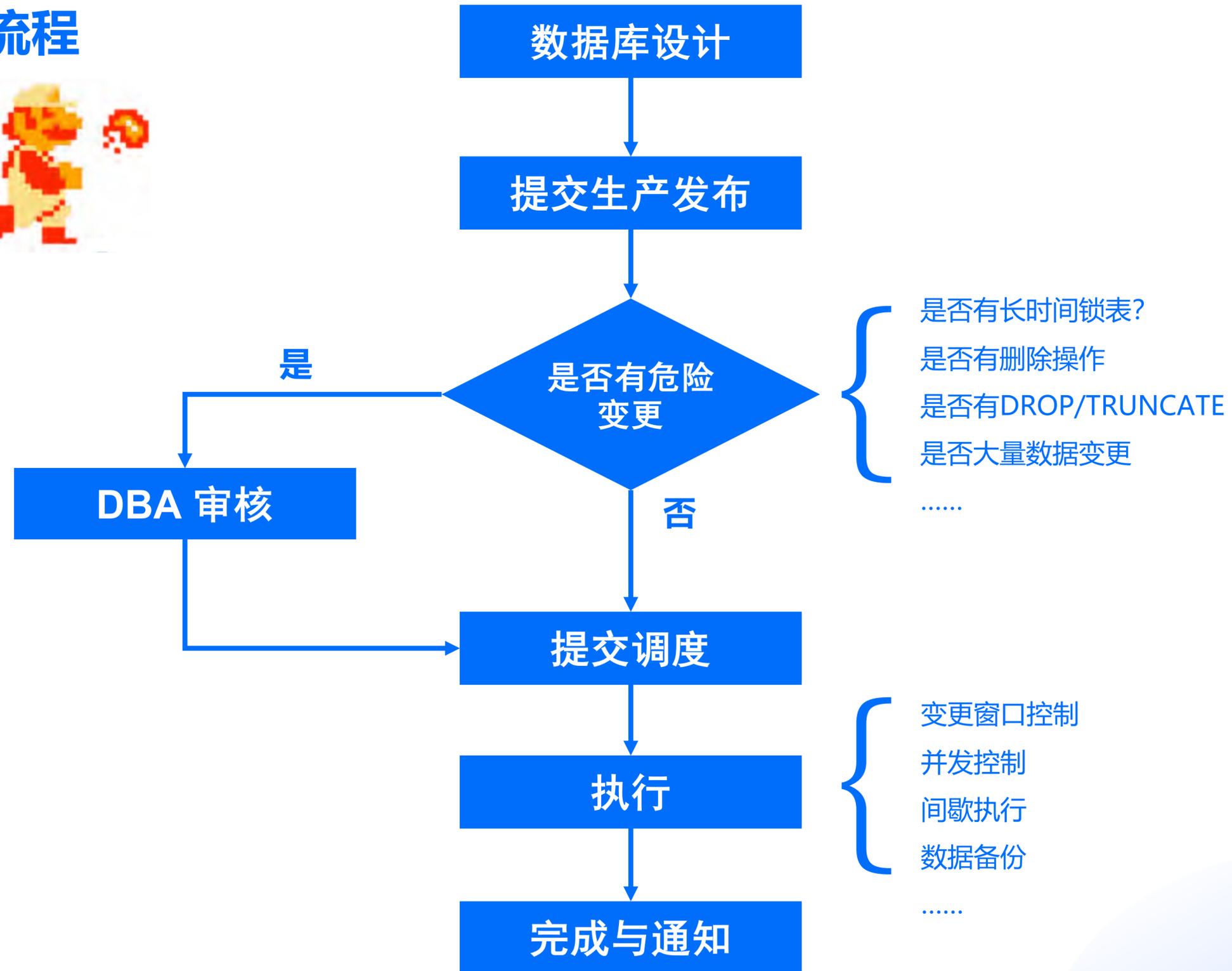
- 索引命名规范
- 不过多使用单列索引
- 字符尽量定义前缀

SQL高效

- 尽量使用索引
- 尽量使用覆盖索引
- 避免WHERE中的隐式转换
- 索引列避免计算

高效，但依旧保障数据库的稳定性

一次具体的变更流程



数据变更与回滚

保护数据变更的安全



t-backup st-culyh3mcpk1s 执行成功

周振兴 创建于 2023-08-07 10:26:56(UTC +08:00) 数据源: \.01-ninedata-sample-datasource

错误处理: 执行出错后, 终止任务 备份失败处理: 备份失败后, 继续任务

执行时间: 2023-08-07 10:27:34(UTC +08:00) 备注: t_007 backup

数据备份: ● 已完成 [下载](#)

提交任务 2023-08-07 规范预审 2023-08-07 提交审批 2023-08-07 自动通过

执行结果 规范预审 变更 SQL [全部 SQL](#) [出错 SQL](#)

行号	Query	详情
1	DELETE FROM t_007 WHERE id in (1,3);	影响行数: 2
2	ALTER TABLE t_007 ADD INDEX IND_NICK(nick);	影响行数: 0

共 2 条 < 1 > 10 条/页

操作历史

操作时间	操作人	操作项	详情
2023-08-07 10:27:39(UTC +08:00)	NineData	执行成功	-
2023-08-07 10:27:31(UTC +08:00)	NineData	开始执行	-
2023-08-07 10:27:31(UTC +08:00)	NineData	备份成功	开始执行 SQL
2023-08-07 10:27:21(UTC +08:00)	NineData	开始备份	-
2023-08-07 10:27:21(UTC +08:00)	周振兴	执行	Execute immediately
2023-08-07 10:26:57(UTC +08:00)	NineData	自动通过	根据规范, 允许直接提交不需要审批
2023-08-07 10:26:56(UTC +08:00)	周振兴	提交申请	-

```
CREATE TABLE t_007 (id int,nick varchar(32));
INSERT INTO `t_007` VALUES (1,'Tom'),(2,'Jerry'),(3,'foo'),(4,'bar');
```

```
DELETE FROM t_007 WHERE id in (1,3);
ALTER TABLE t_007 ADD INDEX IND_NICK(nick);
```

```
/*
select * from t_007 WHERE id in (1,3);
*/
-- `t_007`

REPLACE INTO `t_007` (`id`,`nick`) VALUES (1,'Tom');
REPLACE INTO `t_007` (`id`,`nick`) VALUES (3,'foo');
/*
show create table `t_007`;
*/
CREATE TABLE `t_007` (
  `id` int DEFAULT NULL,
  `nick` varchar(32) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Online DML

大表，不锁表的DML变更



```
DELETE FROM t_orders WHERE created_time < "2024-01-01"
```

数亿的数据如何安全删除？

- ✓ 避免“长事务”影响
- ✓ 有效控制短时系统压力
- ✓ 避免短时日志空间暴增
- ✓ 自动拆分、自动执行

* 规则名称

* 支持数据库
 MySQL

* DML 语句扫描行数风险阈值 ?

* 是否开启 OnlineDML

* OnlineDML 分批大小(行数)

* OnlineDML Sleep 时间(s) ?

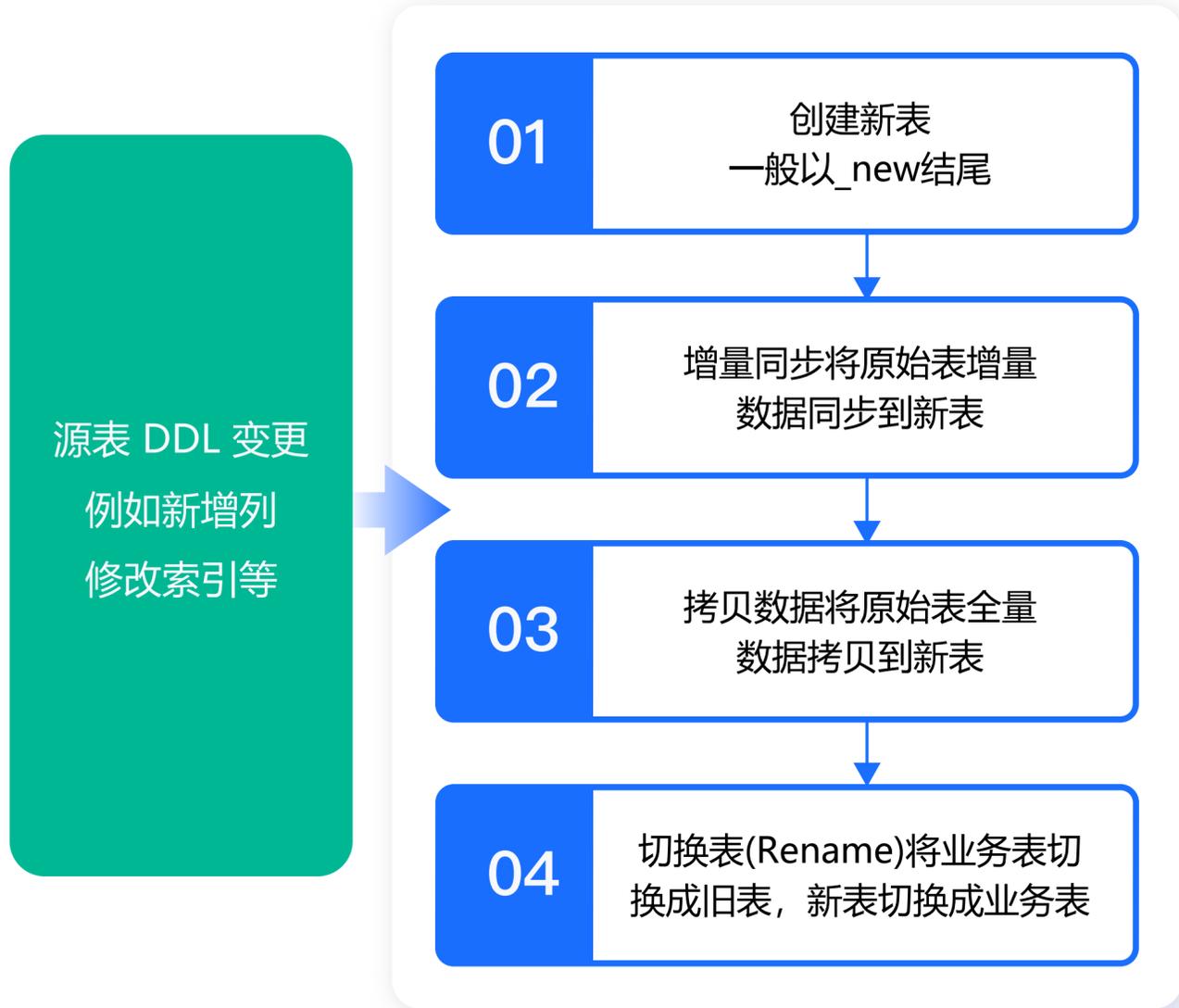
Online DDL

大表，不锁表的DDL变更



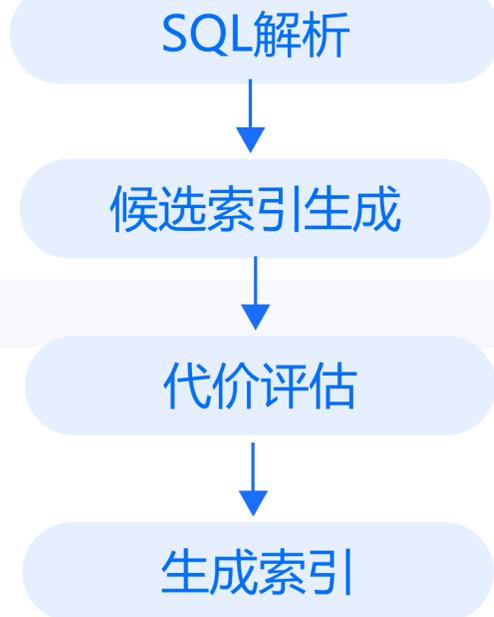
```
ALTER TABLE t_01 modify column c_int(archar(32))
```

- ✓ 大表变更，对业务零影响
- ✓ 并发控制，保障稳定
- ✓ 进度可观察
- ✓ 无需触发器，无额外压力



提供自动化SQL优化建议

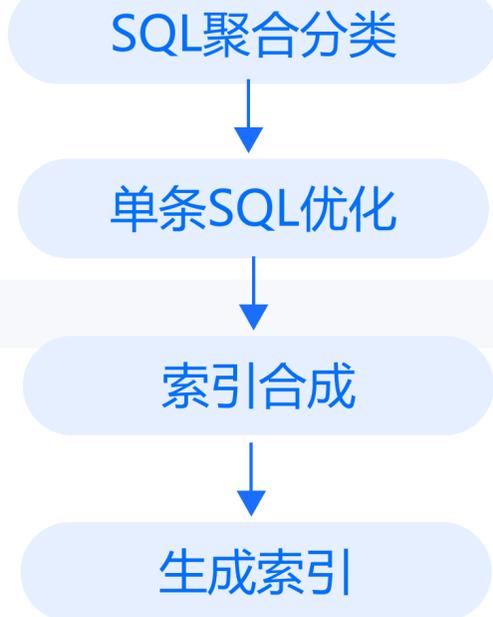
单条SQL优化流程



```

SELECT *
FROM t1
WHERE time_created >= '2017-11-25'
AND consuming_time > 1000
ORDER BY consuming_time DESC;
  
```

局部SQL优化流程



```

IX1(time_created)
IX2(time_created, consuming_time)
IX3(consuming_time)
IX4(consuming_time, time_created)
  
```

SQL诊断建议

详情

规范审核 使用默认生产环境规范

- SELECT 指定业务需要的字段，避免返回多余字段造成性能下降和资源浪费
- SELECT 未指定 WHERE 条件，会造成全表扫描，建议指定 WHERE 条件进行高效检索

索引推荐

- lineitem 新增索引 1 个
 - 待选索引: create index index_l_shipdate on tpch_new.lineitem(l_shipdate,l_quantity)tpch_new.lineitem(l_shipdate)
 - 索引择优: 根据最左索引前缀相同进行合并, tpch_new.lineitem(l_shipdate,l_quantity)

SQL 语句 默认选择 SQL 模板中

```

10 行 144 B
1 SELECT
2   users.user_id,
3   users.username,
4   orders.order_id,
5   orders.order_date,
6 FROM
7   users
8 WHERE
9   users.registration_date >= '2023-01-01'
10  AND products.category = 'Electronics'
  
```

执行计划 元数据

id	select_type	table	type	possible_keys	key	key_len	ref	rows	extra
	SIMPLE	employees	ALL	PRIMARY	null	0	null	897892	null
	SIMPLE	employees	ref	PRIMARY,emp_no	PRIMARY	4	employees.employees.emp_no	1	null
	SIMPLE	titles	ref	PRIMARY,emp_no	PRIMARY	4	employees.employees.emp_no	1	null
	SIMPLE	salaries	ref	PRIMARY,emp_no	PRIMARY	4	employees.employees.emp_no	1	null
	SIMPLE	salaries	ref	PRIMARY,emp_no	PRIMARY	4	employees.employees.emp_no	1	null

关闭

数据库DevOps平台：高效安全的数据库开发平台

AIGC + 企业级最佳实践，定义数据库开发新范式

面向个人

AIGC 助力
数据库开发 Copilot 替代传统开发
工具 (Navicat、Dbeaver)

智能开发 IDE

- 智能补全
- 数据查询
- 数据变更
- 数据导入
- 数据导出

表设计/编辑

- 全列类型
- 选项支持
- 全 GUI 支持
- 实时 SQL 脚本
- DDL 规范

智能 AI 辅助

- 自然语言转 SQL
- 智能索引推荐
- 智能 SQL 优化
- 代码自动生成

个人效率

面向企业

数据库开发新范式
企业数据库开发最佳实践

组织权限

- SSO
- 自定义角色
- 数据源权限
- 多渠道消息

规范流程

- Schema 规范
- DML & 查询规范
- 自定义环境规范
- 自定义审批流程

变更发布

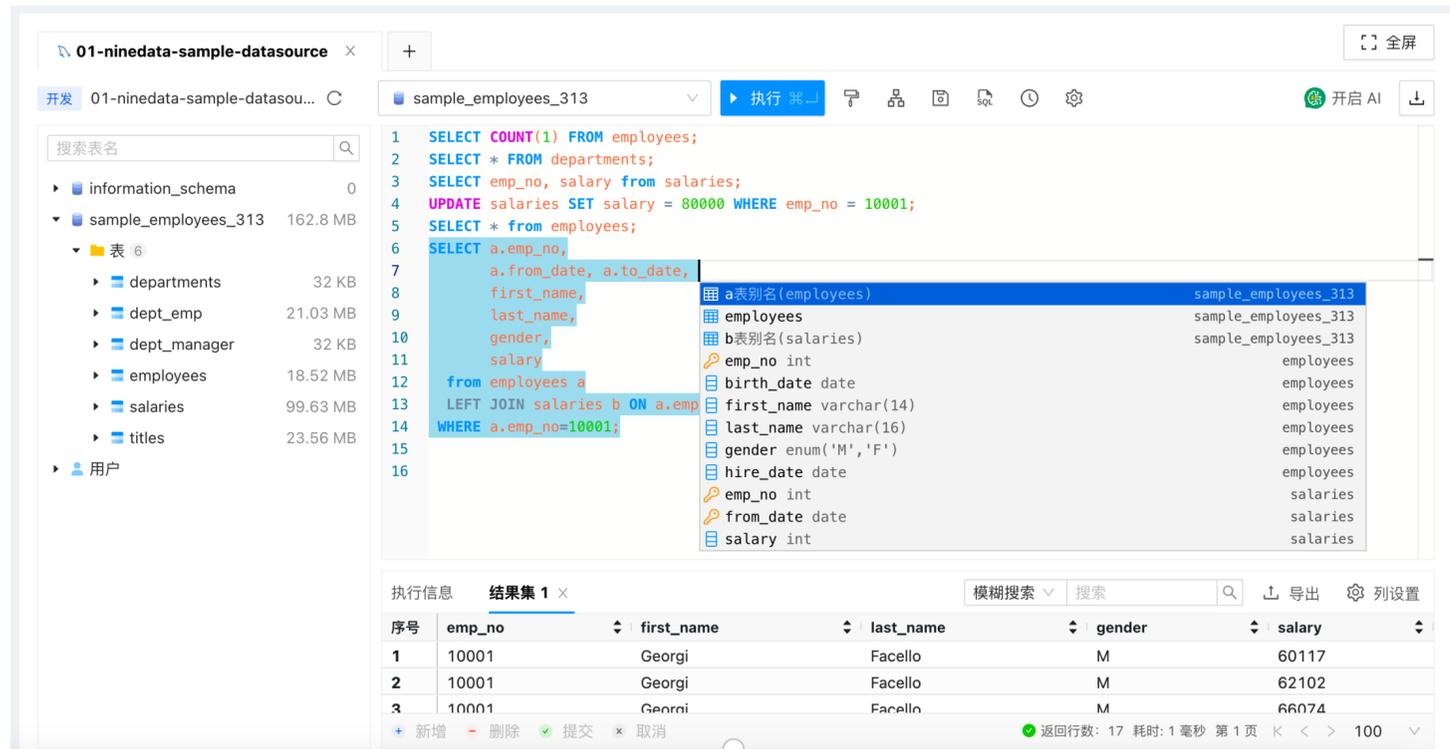
- OnlineDDL
- 变更备份
- 变更回滚

敏感数据管理

- 自动发现
- 多种发现规则
- 手动配置
- 多种遮掩规则

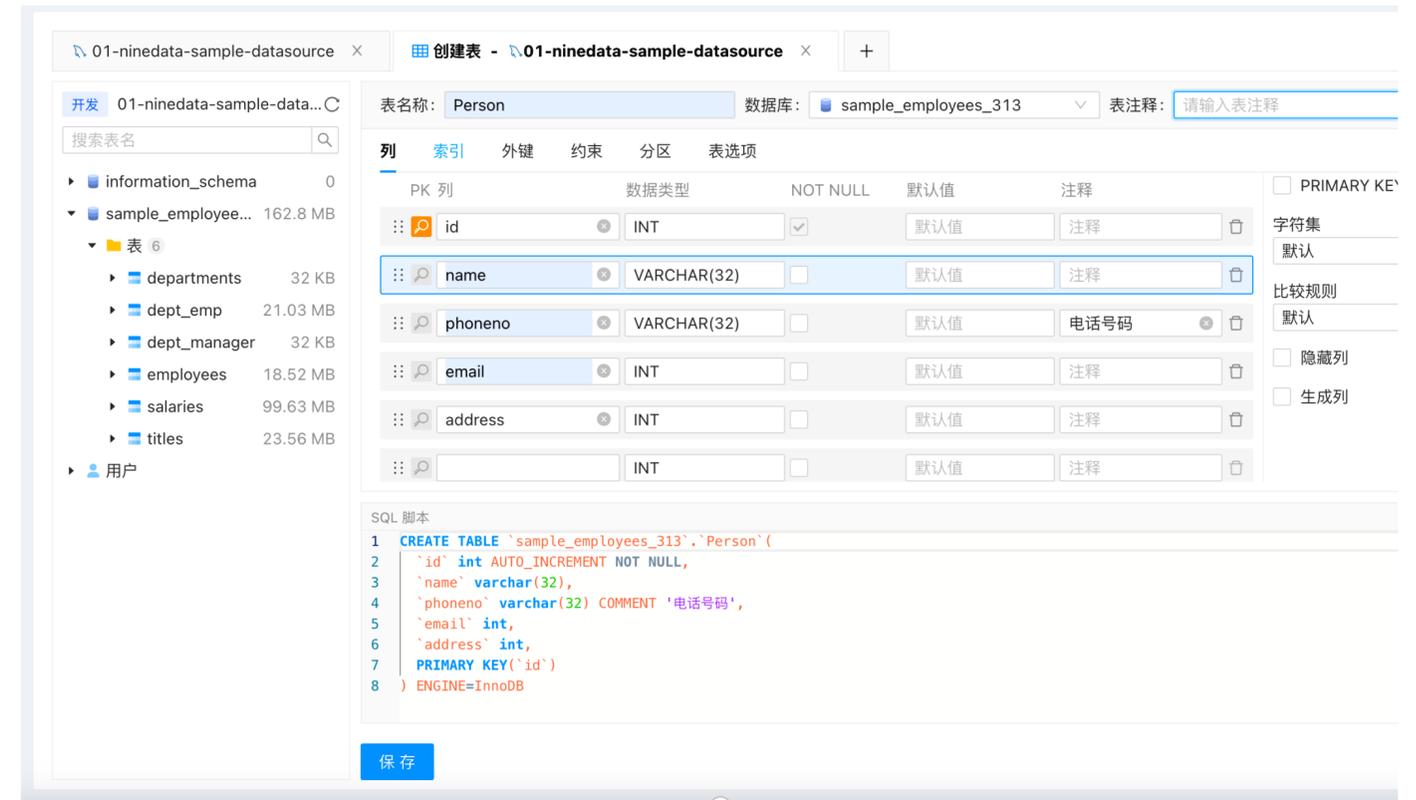
团队效率

首先，它是一个生产力工具



- 智能语法补全
- 支持导航树遍历对象
- 常用SQL保持
- 查询历史查看
- 结果集编辑

- 数据导出
- 执行计划查看
- 支持用户、角色与权限管理
- 支持全量SQL执行审计
- 支持敏感数据管理



- 支持完整字段类型，包括常用字段、空间类型、TEXT、BLOB类型等
- 支持完整的字段类型选项，包括默认值、隐藏列、生成列、字符集、SRID等
- 支持多种类型索引创建与修改，包括普通索引、唯一索引、空间索引等
- 完整的约束/外键/分区/其他表选项支持支持

一个平台，多种数据库



ORACLE



PolarDB

GaussDB



腾讯分布式数据库



丰富的关系型数据库支持，包括

开源数据库

商业数据库

分布式数据库

云原生数据库



文档数据库

内存数据库

分析型数据库

消息队列

HTAP数据库

规范、流程来保障生产环境的稳定

默认生产环境规范

结构
查询与更新
SQL 任务与窗口

规则级别	规则名称	规则类型	状态	支持数据库	描述	操作
必须改进	表变更必须要 Online 执行	表	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
必须改进	表要有主键	表	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
建议改进	限制 VARCHAR 类型字段长度	列	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
建议改进	限制唯一索引名格式	索引	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
建议改进	表需要包含某些列	表	<input type="checkbox"/>	<input type="checkbox"/>		
建议改进	限制表不能使用外键	表	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
建议改进	限制单个索引包含列的个数	索引	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
建议改进	限制主键包含列的个数	索引	<input type="checkbox"/>	<input type="checkbox"/>		
建议改进	字段名不能是关键字	列	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
建议改进	限制自增列为无符号	列	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

是企业数据库开发规范

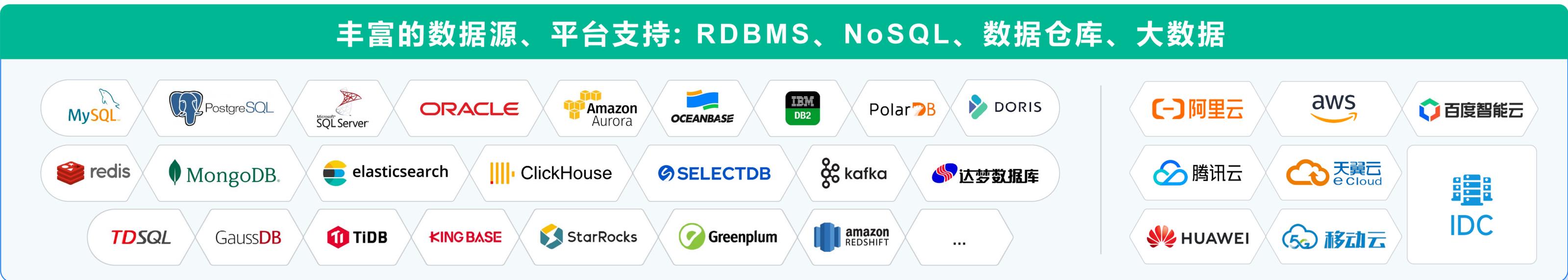
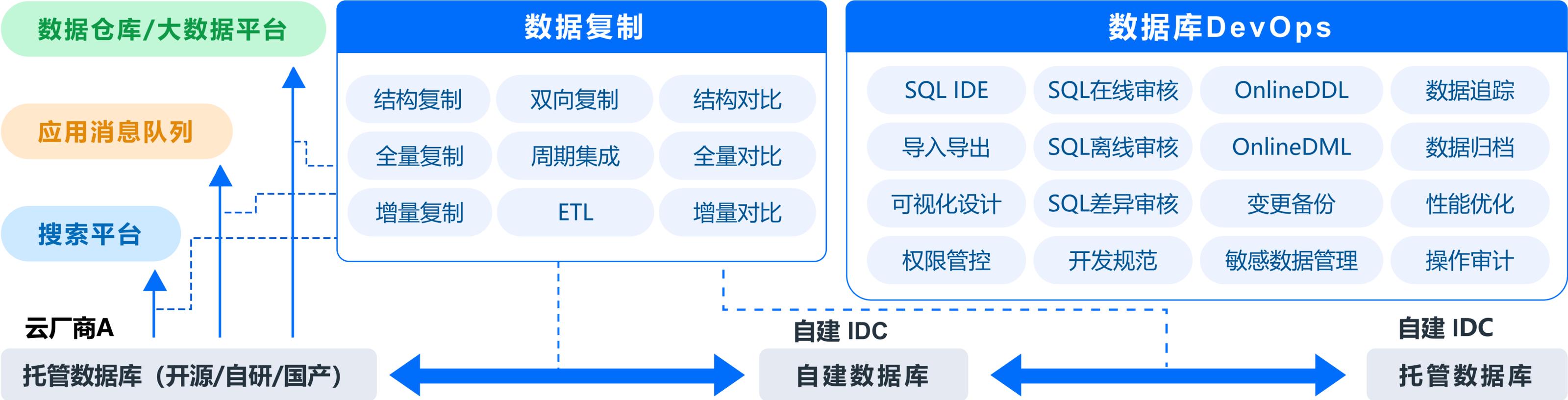
是企业数据库流程规范

保障企业数据库安全、稳定与可扩展

其他问题

- 大数据量的DDL
- 敏感数据保护
- 日常数据导入导出
- 数据归档
- 如何与企业内的身份认证统一
- 批量的数据源、团队权限如何管理
- 变更出错后，数据如何回滚
- 如何利用大模型的能力

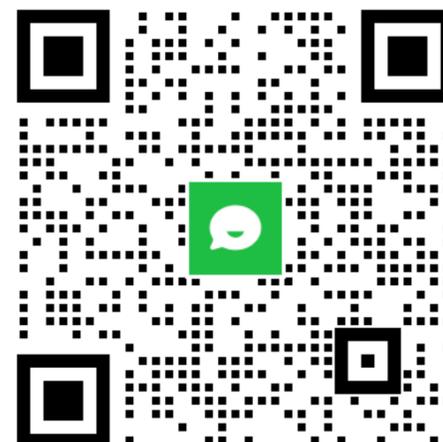
NineData云原生智能数据管理平台



www.ninedata.cloud



关注NineData



扫码立刻获取支持