



第九届中国数据库技术大会
DATABASE TECHNOLOGY CONFERENCE CHINA 2018

PostgreSQL

51风控系统背后的利器

张泽鹏

DTCC
2018

2018.05.10 – 12 北京国际会议中心



IT168.com

ChinaUnix

ITPUB

个人简介

- 2015年加入51信用卡
- 担任架构师
- 负责金融信贷相关系统
 - 风控
 - 信审
 - 数据支持
 -



AGENDA

- 风控业务概览
- 风控系统概览
 - 快照数据
 - 维度数据
 - 名单数据
 - 任务调度



风控业务概览

DTCC
2018

数领先机 智赢未来 (9)

IT168.com

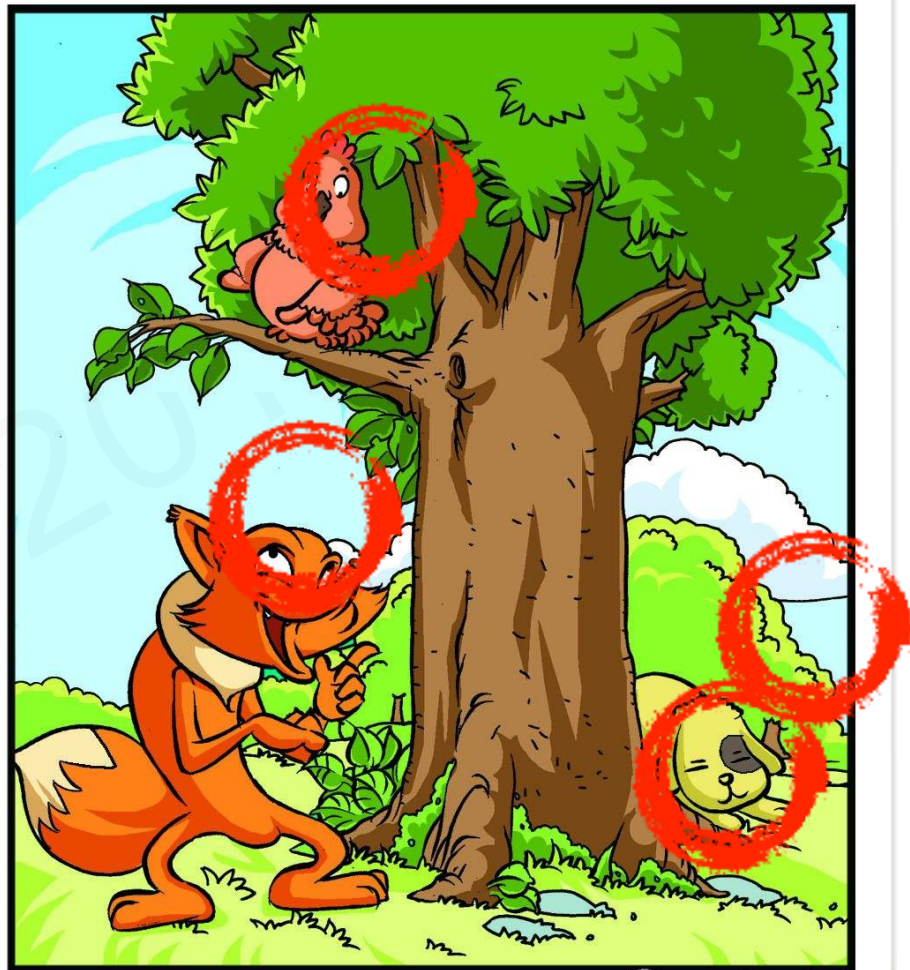
ChinaUnix

ITPUB

风控业务概览

.....





举个例子

- 符合以下条件的用户通过，否则拒绝：
 - 男性客户：年龄在22到55岁之间；
 - 女性客户：年龄在18到65岁之间。

风控系统概览

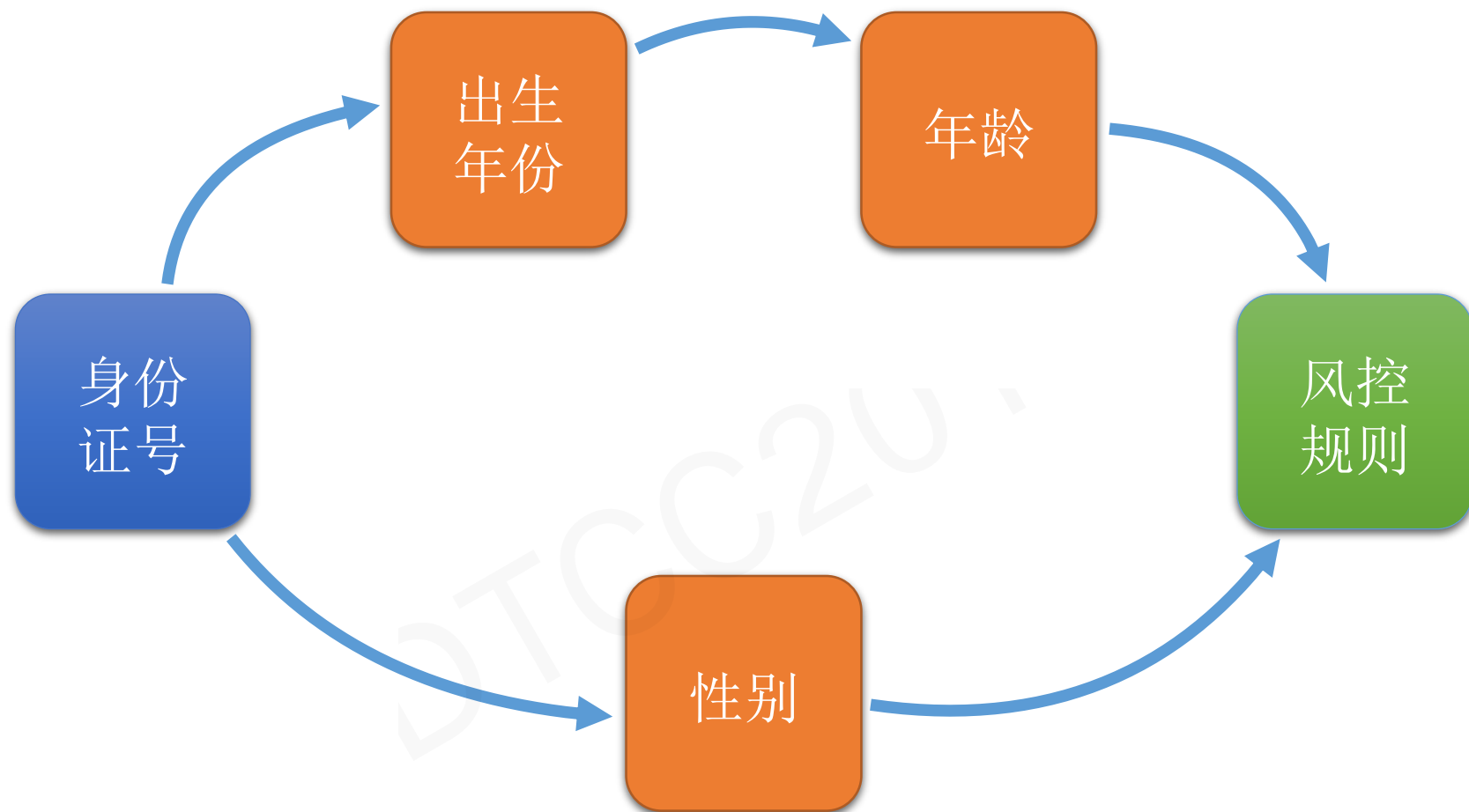
DTCC
2018

数领先机 智赢未来 (9)

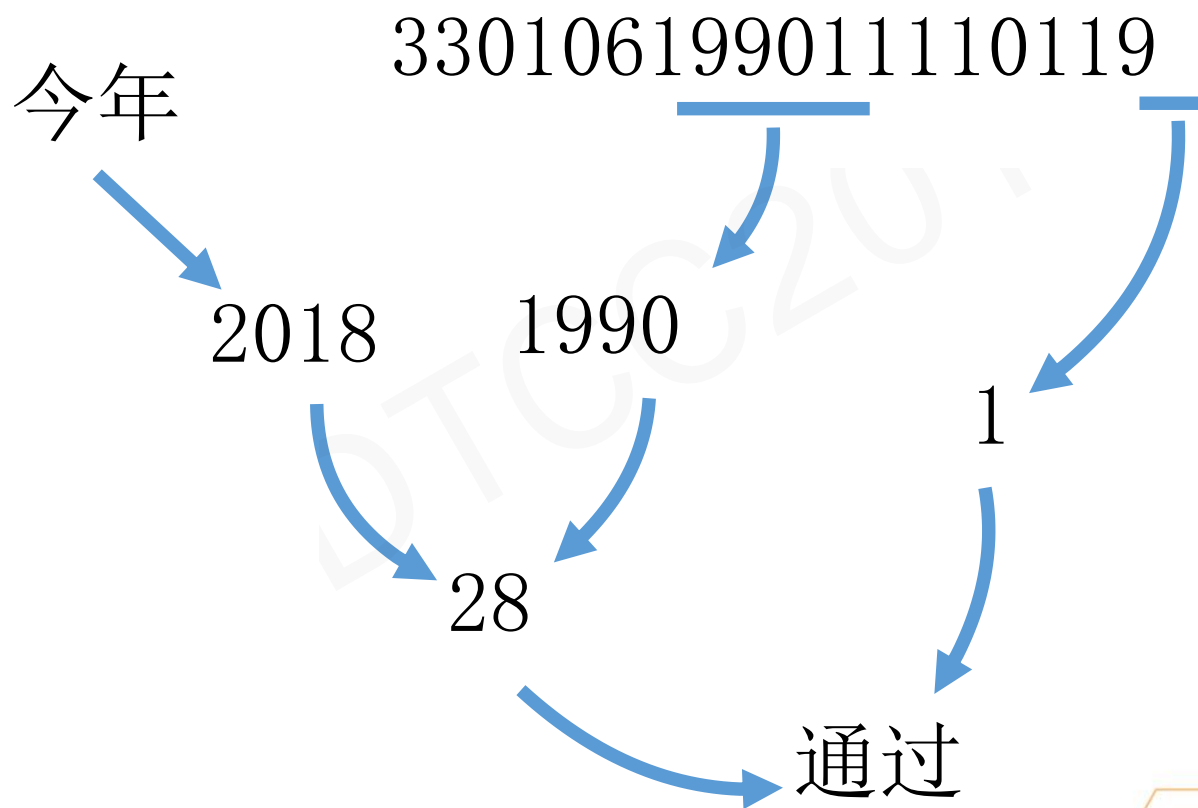
IT168.com

ChinaUnix

ITPUB



举个例子



风控系统概览

工作簿2

Q 在工作表中搜索

开始 插入 页面布局 公式 数据 审阅 视图

fx =IF(OR(AND(B5 = 1, 22 <= B4, B4 <= 55), AND(B5 = 0, 18 <= B4, B4 <= 65)), 1, 0)

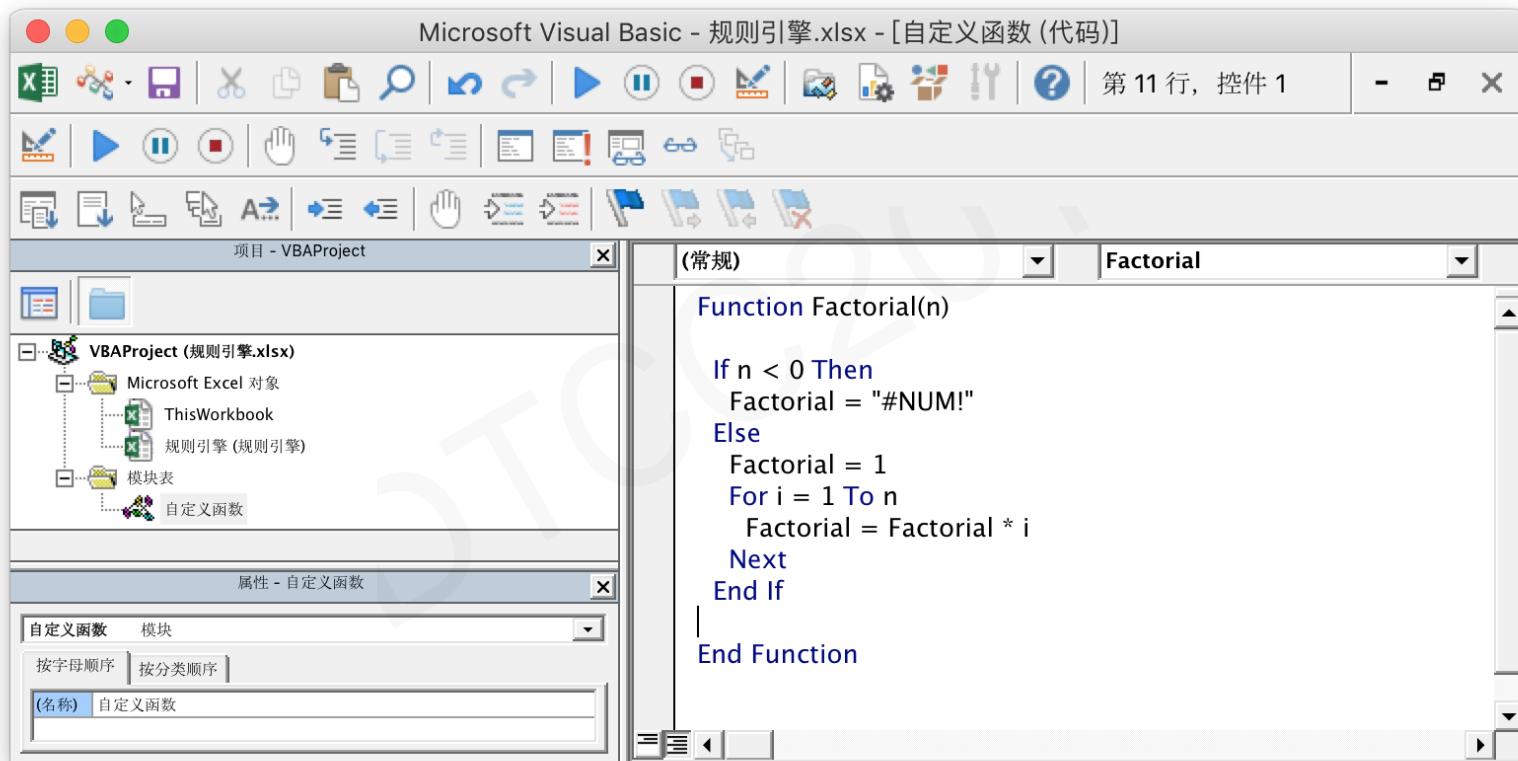
	A	B	C
	名称	值	描述
1	id_card	330106199011110119	身份证号
2	birth_year	1990	出生年份：身份证7-10位
3	age	28	年龄：当前年份-出生年份
4	gender	1	性别：0-女；1-男
5	is_accept	=IF(OR(AND(B5 = 1, 22 <= B4, B4 <= 55), AND(B5 = 0, 18 <= B4, B4 <= 65)), 1, 0)	是否接受：1通过；0拒绝 * 男性年龄22-55之间 * 女性年龄18-65之间
6			
7			

工作表1

编辑

300%

风控系统概览



风控系统概览

运算

清洗

特征

模型

规则

任务调度

存储

快照

维度

名单

快照系统存储

DTCC
2018

数领先机 智赢未来 (9)

IT168.com

ChinaUnix

ITPUB

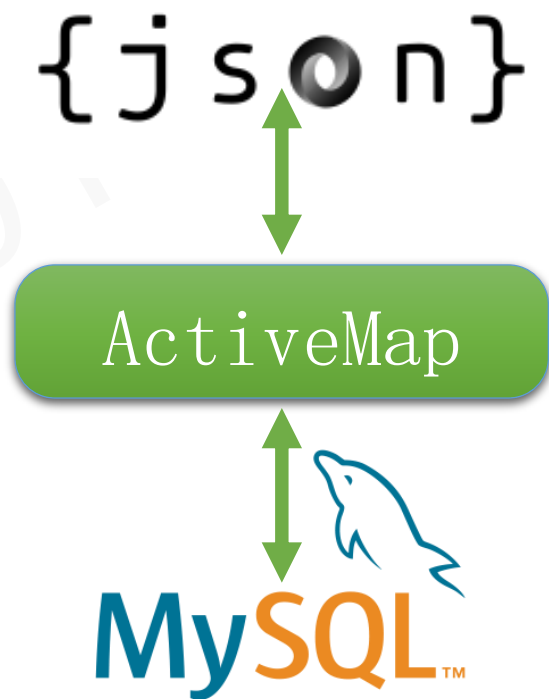
方案一：MONGODB

- 成也文档 {json}
- 败也文档
 - NoSQL
 - SAS



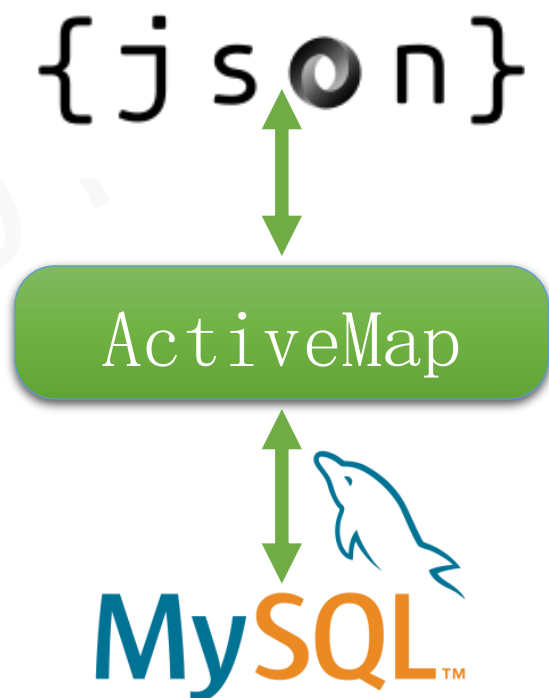
方案二：MYSQL

- inputs
- features
 - id
 - addresses
 - feature_id
- rules



方案二：MYSQL

- 痛点
 - 大表添加索引
 - 大表新增字段



方案三：POSTGRESQL

➤ 解决

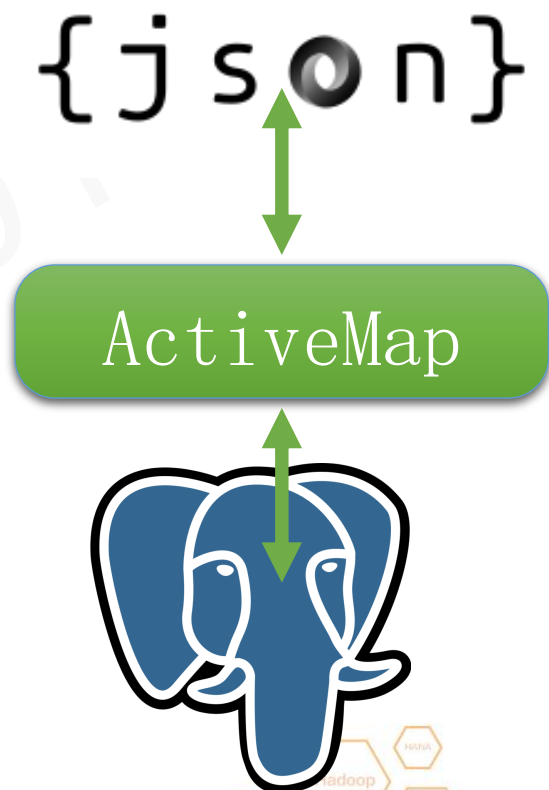
➤ 大表添加索引

- create index concurrently

➤ 并发构建索引

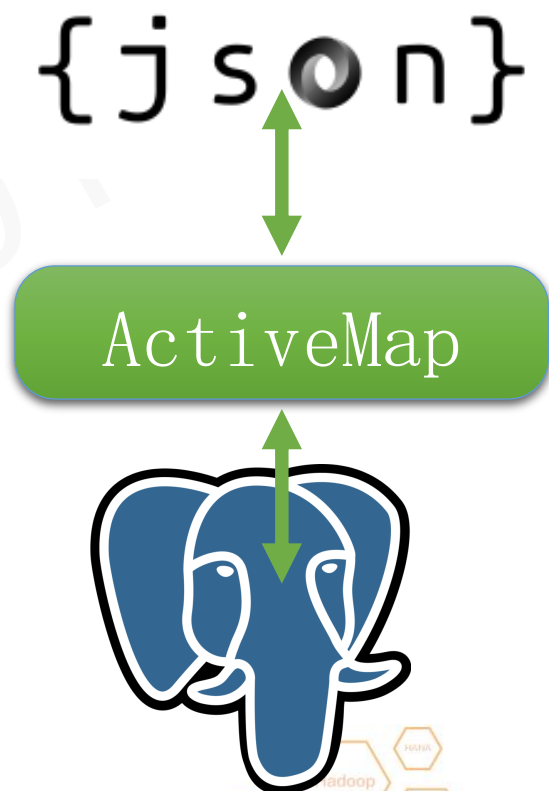
➤ 大表新增字段

- 无默认值瞬间完成



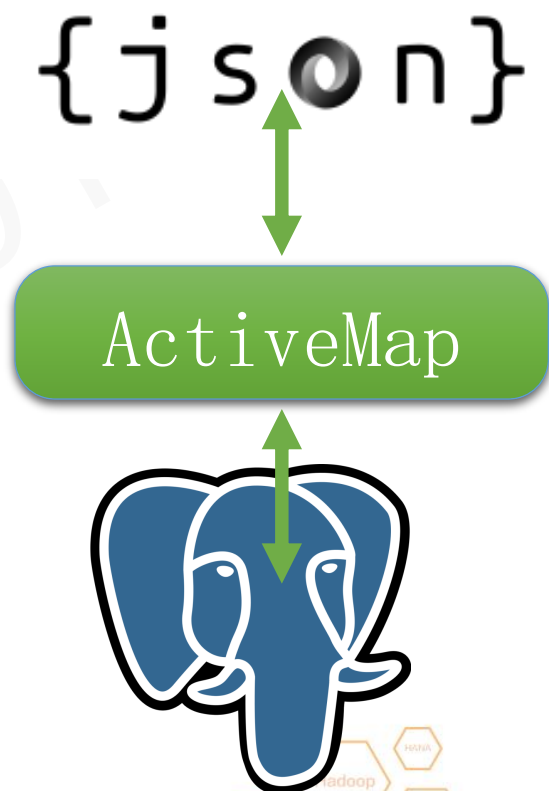
方案三：POSTGRESQL

- 1分钟收到8000笔申请
- 每笔产生约20万记录
- 单事务最长超过20秒
- 大量事务排队并超时
- TPS突破50000



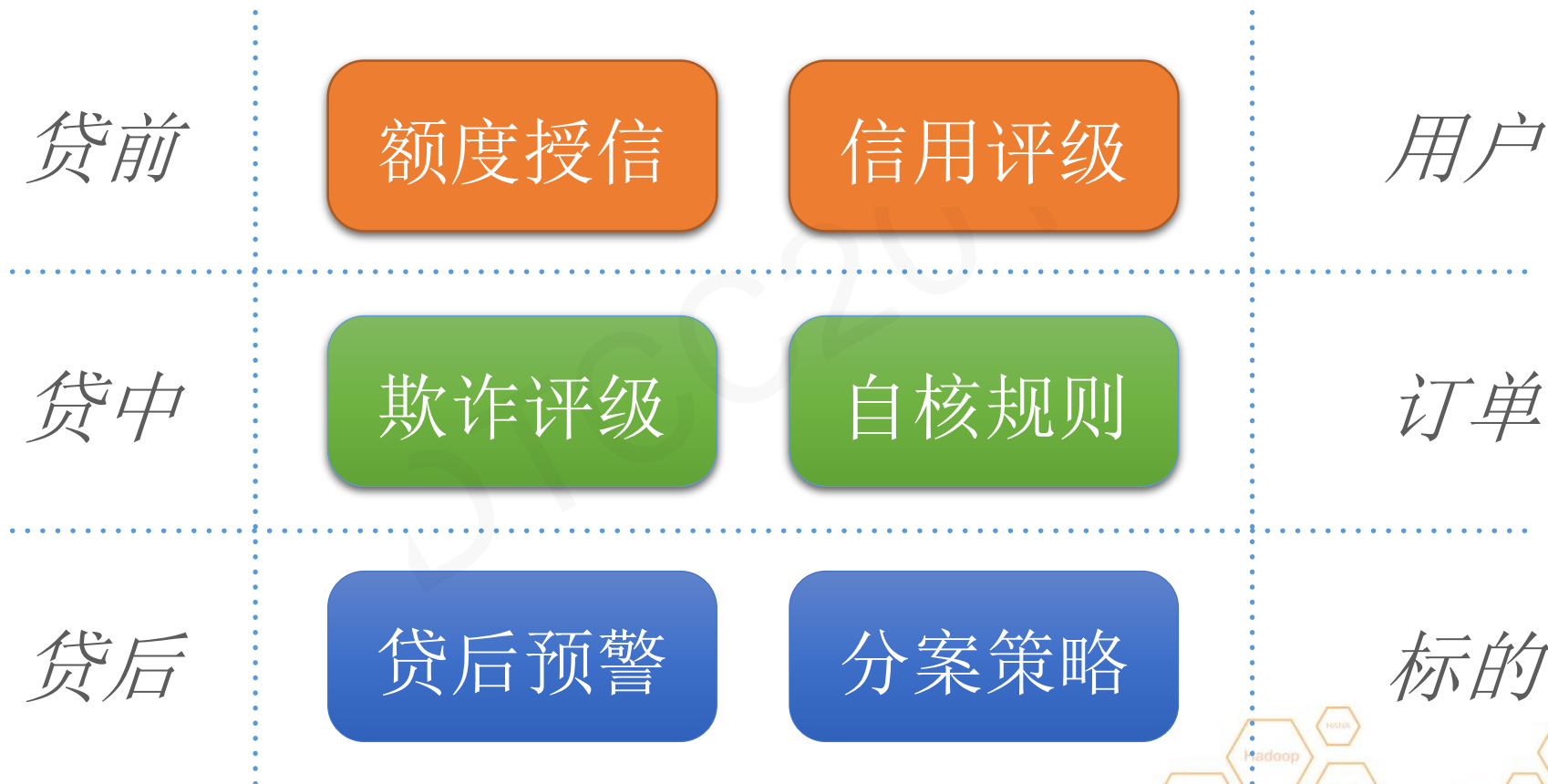
方案三：POSTGRESQL

- 优化应用端任务调度
- JSONB
- Hive中解析JSONB



维度数据存储

维度数据存储



组合爆炸

➤ 维度

- orders

- users

➤ 快照

- inputs

- features

- rules

- orders_inputs

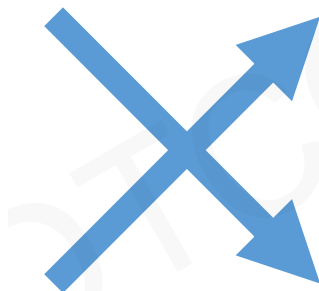
- orders_features

- orders_rules

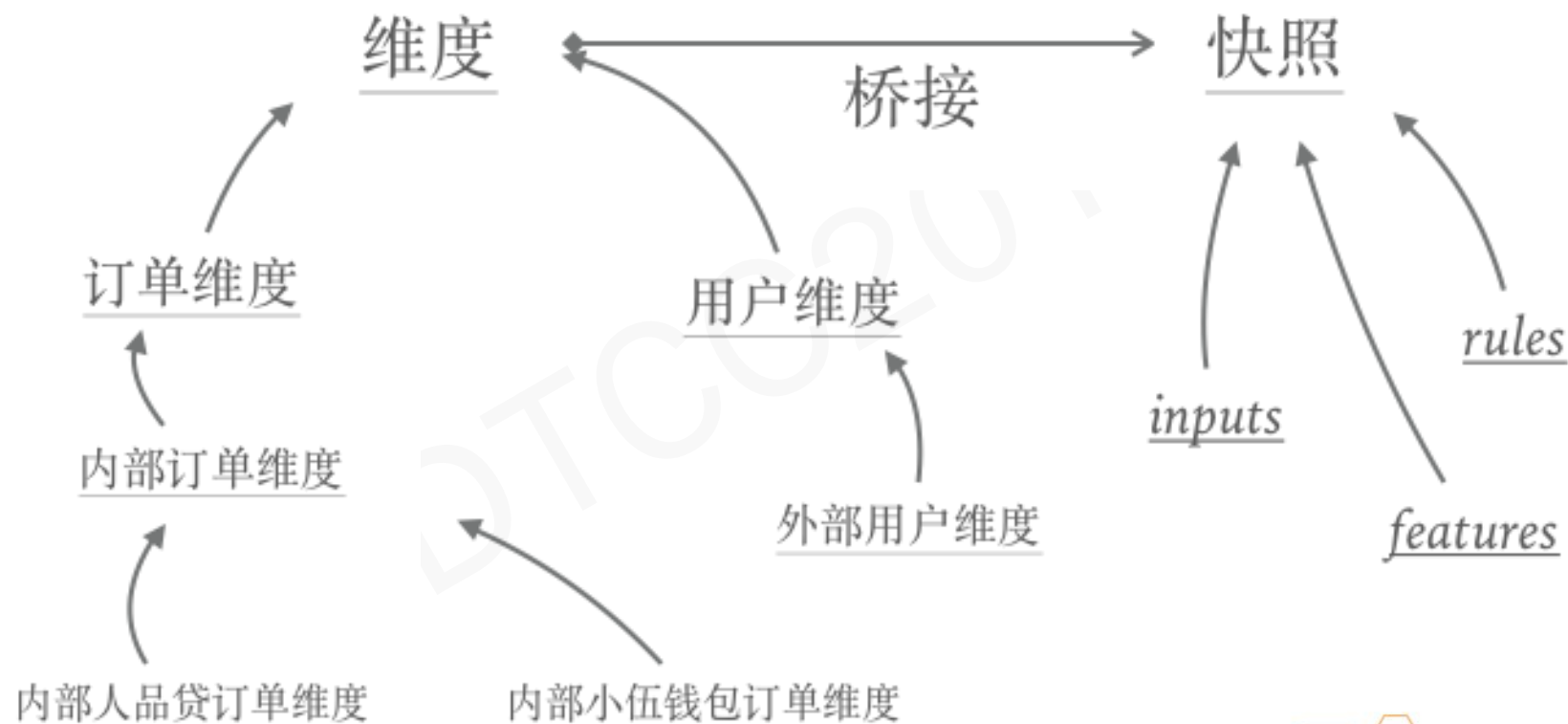
- users_inputs

- users_features

- users_rules



桥接模式



方案一：多表JOIN

- parents

- id

- children

- parent_id

- foo

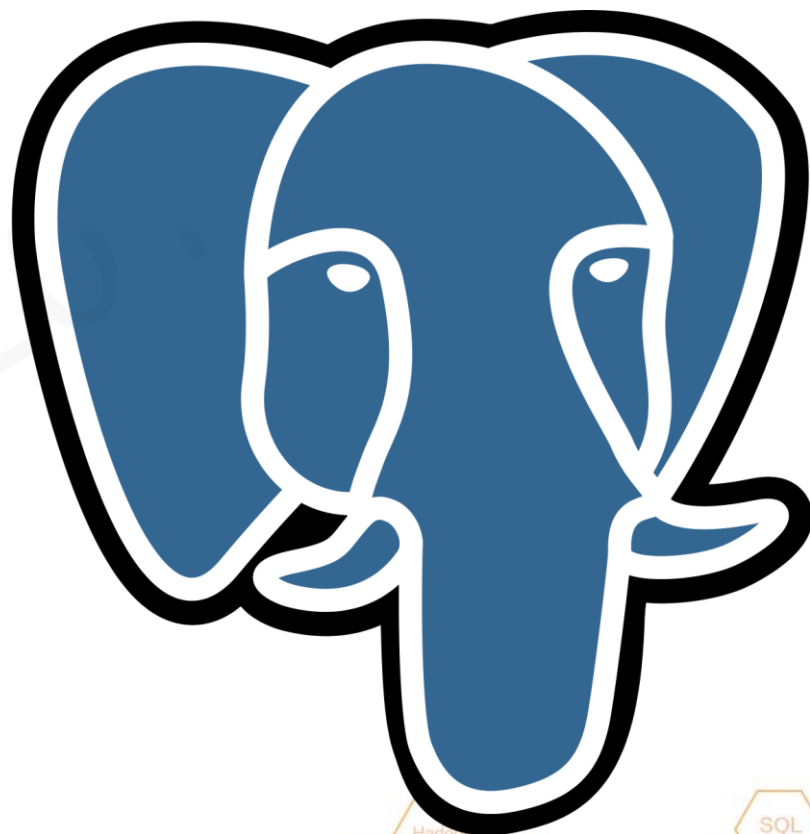
- 通过外键关联

- 父表保存公共字段

- 子表保存扩展字段

方案二：POSTGRESQL继承表

- 子表继承父表的字段
- 父表可查询子表数据
- 通过约束排除无关表
 - `constraint_exclusion`
- 与父表共享一个序列



方案：POSTGRESQL继承表

.....

- `create table parents (id serial primary key, type int, name text);`
- `create table children1 (primary key (id), check (type = 1)) inherits (parents);`
- `create table children2 (primary key (id), check (type = 2)) inherits (parents);`
- `select * from parents where type = 1;`



名单系统存储

方案一：资源文件

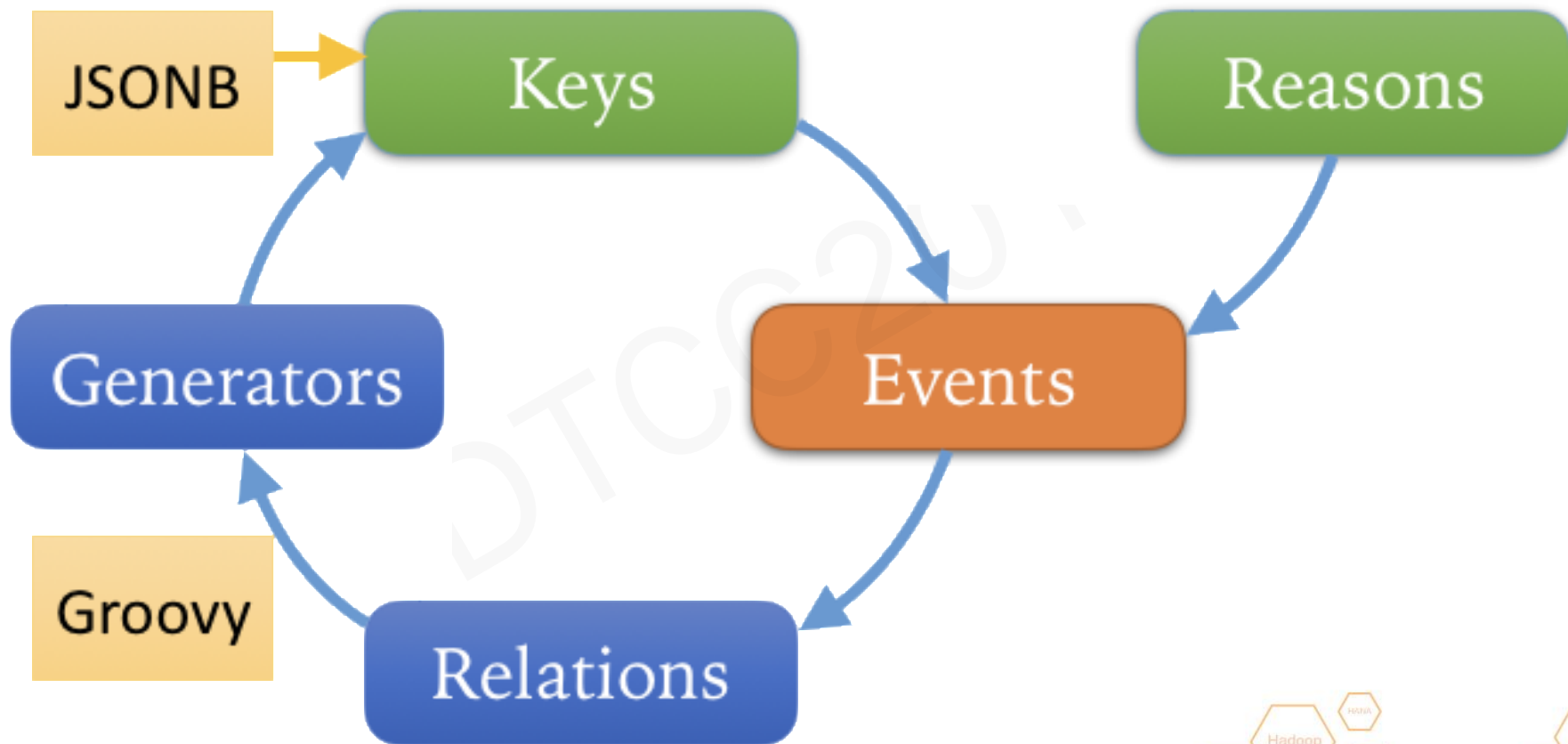
- 名单作为资源文件打包进每个Java项目中
- 系统启动时将数据加载到内存
- 生成HashSet
- 查询=检查Key是否在集合中
- 问题：多个项目都在引用，维护困难



方案二：REDIS

- 将名单批量导入Redis集群
- 查询：检查Key是否在Redis中
- 新需求：模糊查询功能
 - 假设330106199011110119在黑名单
 - 查询3301061990111*****返回“击中”

方案三：POSTGRESQL



方案三：POSTGRESQL

.....

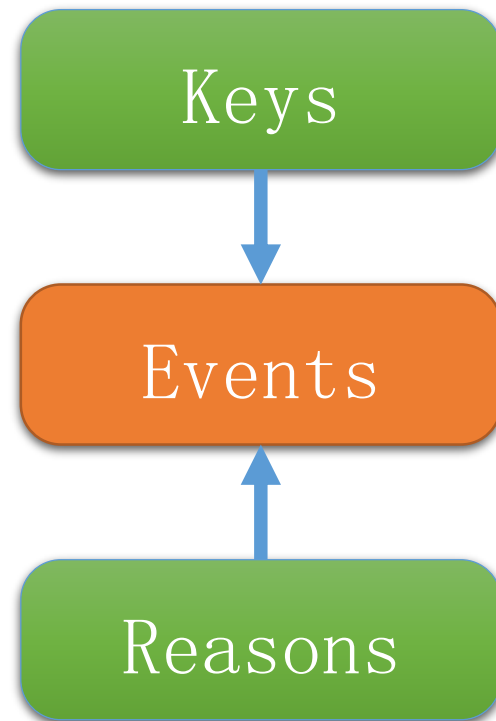
- 每次新增Groovy Generator时，需全量回溯所有Events中的Keys，来生成新的Key
- 无法用COPY来批量导入Keys，因为衍生的Key依赖应用端用Groovy生成，直接导入数据库不会自动生成衍生Keys
- Groovy Generator性能不高



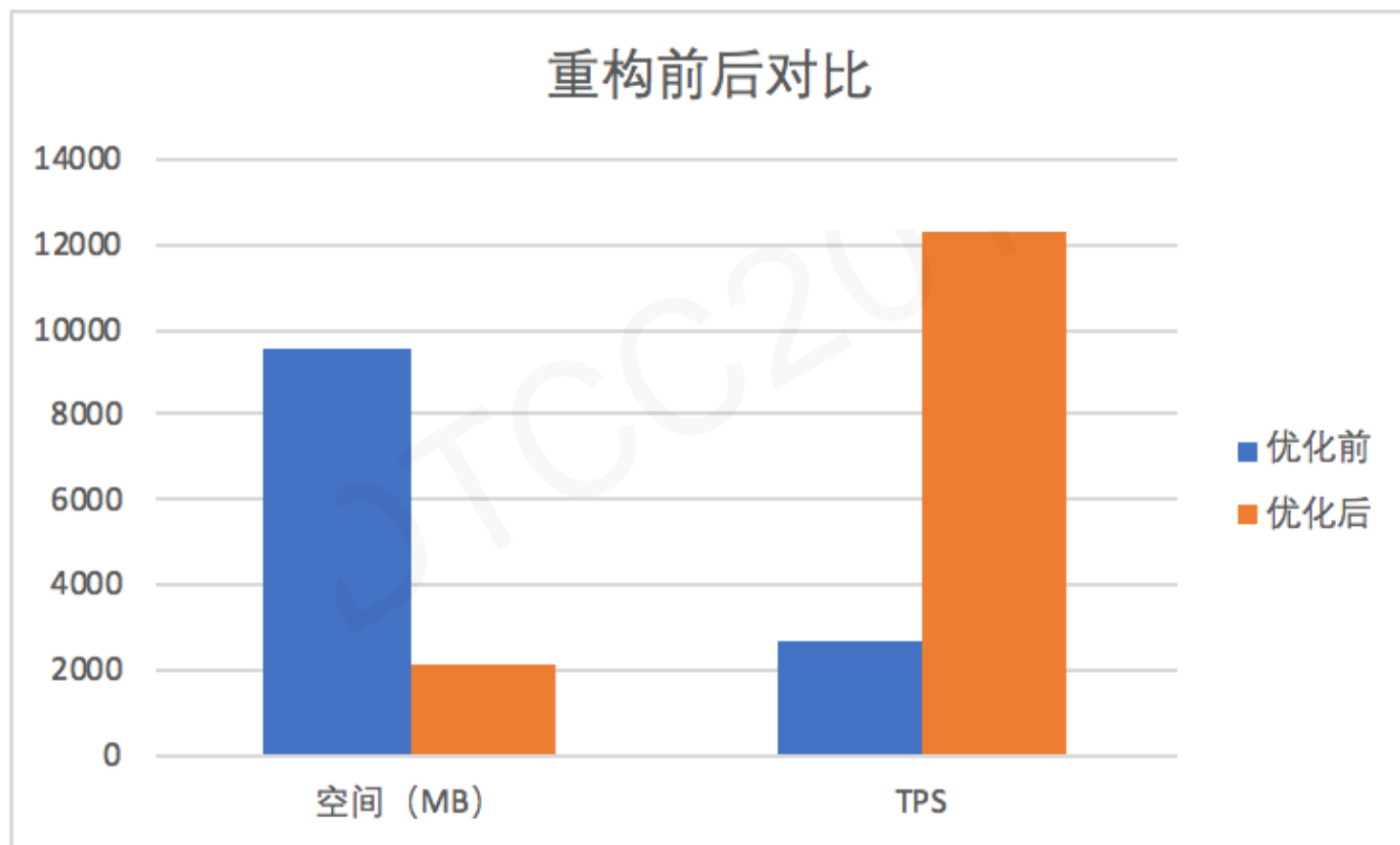
方案四：POSTGRESQL

- UDF
- 表达式索引

```
create index on keys (id_card_mask(context));
select
  *
from
  keys
where
  id_card_mask(context) = '3301061990111*****';
```



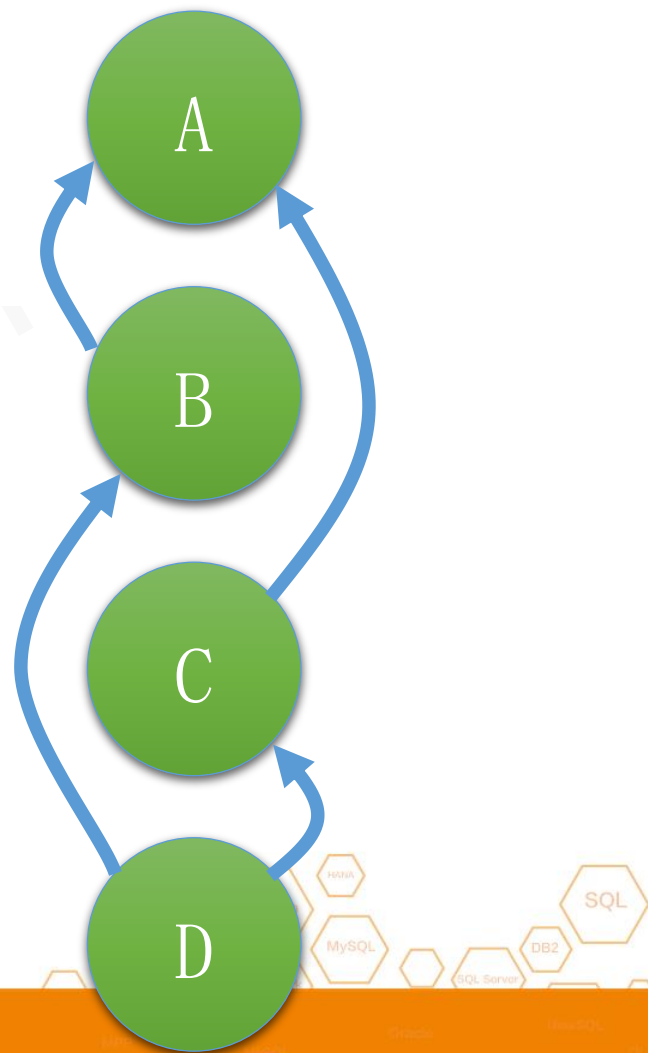
最终效果



任务调度系统

依赖关系

id	dep_id
D	C
D	B
C	A
B	A



方案一：POSTGRESQL多表JOIN

- 每个任务对应一张任务队列表
- 用status字段表示任务当前状态
 - submitted: 待处理
 - completed: 已完成
- 任务的依赖通过JOIN依赖任务的队列

queue_a

id	status
1	submitted

queue_b

id	status
1	submitted

queue_c

id	status
1	submitted


queue_d

id	status
1	submitted

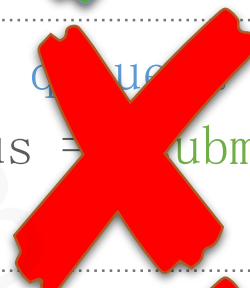
方案一：任务依赖的队列状态

任务名称	队列名称	依赖状态
A	queue_a	submitted
B	queue_a	completed
B	queue_b	submitted
C	queue_a	completed
C	queue_c	submitted
D	queue_b	completed
D	queue_c	completed
D	queue_d	submitted


(A) `select queue_a.id from queue_a where queue_a.status = 'submitted';`



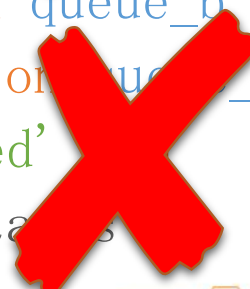
(B) `select queue_b.id from queue_b inner join queue_a on queue_b.id = queue_a.id where queue_b.status = 'submitted' and queue_a.status = 'completed';`



(C) `select queue_c.id from queue_c inner join queue_a on queue_c.id = queue_a.id where queue_c.status = 'submitted' and queue_a.status = 'completed';`



(D) `select queue_d.id from queue_d inner join queue_b on queue_d.id = queue_b.id inner join queue_c on queue_d.id = queue_c.id where queue_d.status = 'submitted' queue_b.status = 'completed' and queue_c.status = 'completed';`



queue_a

id	status
1	completed

queue_b

id	status
1	submitted

queue_c

id	status
1	submitted

queue_d

id	status
1	submitted

(A) `select queue_a.id from queue_a where queue_a.status = 'submitted';`

(B) `select queue_b.id from queue_b inner join queue_a on queue_b.id = queue_a.id where queue_b.status = 'submitted' and queue_a.status = 'completed';`

(C) `select queue_c.id from queue_c inner join queue_a on queue_c.id = queue_a.id where queue_c.status = 'submitted' and queue_a.status = 'completed';`

(D) `select queue_d.id from queue_d inner join queue_b on queue_d.id = queue_b.id inner join queue_c on queue_d.id = queue_c.id where queue_d.status = 'submitted' queue_b.status = 'completed' and queue_c.status = 'completed';`



queue_a

id	status
1	completed

queue_b

id	status
1	completed

queue_c

id	status
1	completed

queue_d

id	status
1	submitted

(A) select queue_a.id from queue_a where queue_a.status = 'submitted';

(B) select queue_b.id from queue_b inner join queue_a on queue_b.id = queue_a.id where queue_b.status = 'submitted' and queue_a.status = 'completed';

(C) select queue_c.id from queue_c inner join queue_a on queue_c.id = queue_a.id where queue_c.status = 'submitted' and queue_a.status = 'completed';

(D) select queue_d.id from queue_d inner join queue_b on queue_d.id = queue_b.id inner join queue_c on queue_d.id = queue_c.id where queue_d.status = 'submitted' and queue_b.status = 'completed' and queue_c.status = 'completed';



queue_a

id	status
1	completed

queue_b

id	status
1	completed

queue_c

id	status
1	completed

queue_d

id	status
1	completed

方案一：POSTGRESQL多表JOIN

- 原理简单：任何关系型数据库都能实现
- 条件索引：只索引待处理的任务

```
create index on queues (id)  
where status = 'submitted';
```

方案一：问题

- 必须将 submitted 的队列表放在 inner join 最前面的位置
 - 否则优化器无法正确选中
- 依赖膨胀：超过40张表inner join
 - 查询性能下降，最长耗时730ms



方案二：POSTGRESQL倒排索引

- 用两个数组字段取代原来任务队列表
 - submits int[]
 - completes int[]
- 数组字段创建GIN倒排索引

方案二：POSTGRESQL倒排索引

.....

queues		
id	submits	completes
1	{A, B, C, D}	{}

(A) select id from queues where submits
@> ' {A} ' ;



(B) select id from queues where submits
@> ' {B} ' and completes @> ' {A} ' ;



(C) select id from queues where submits
@> ' {C} ' and completes @> ' {A} ' ;



(D) select id from queues where submits
@> ' {D} ' and completes @> ' {B, C} ' ;



方案二：POSTGRESQL倒排索引

.....

queues		
id	submits	completes
1	{B, C, D}	{A}

(A) select id from queues where submits
@> ' {A} ' ;



(B) select id from queues where submits
@> ' {B} ' and completes @> ' {A} ' ;



(C) select id from queues where submits
@> ' {C} ' and completes @> ' {A} ' ;



(D) select id from queues where submits
@> ' {D} ' and completes @> ' {B} ' ;



方案二：POSTGRESQL倒排索引

.....

queues		
id	submits	completes
1	{D}	{A, B, C}

(A) select id from queues where submits
@> ' {A} ' ;



(B) select id from queues where submits
@> ' {B} ' and completes @> ' {A} ' ;



(C) select id from queues where submits
@> ' {C} ' and completes @> ' {A} ' ;



(D) select id from queues where submits
@> ' {D} ' and completes @> ' {B} ' ;

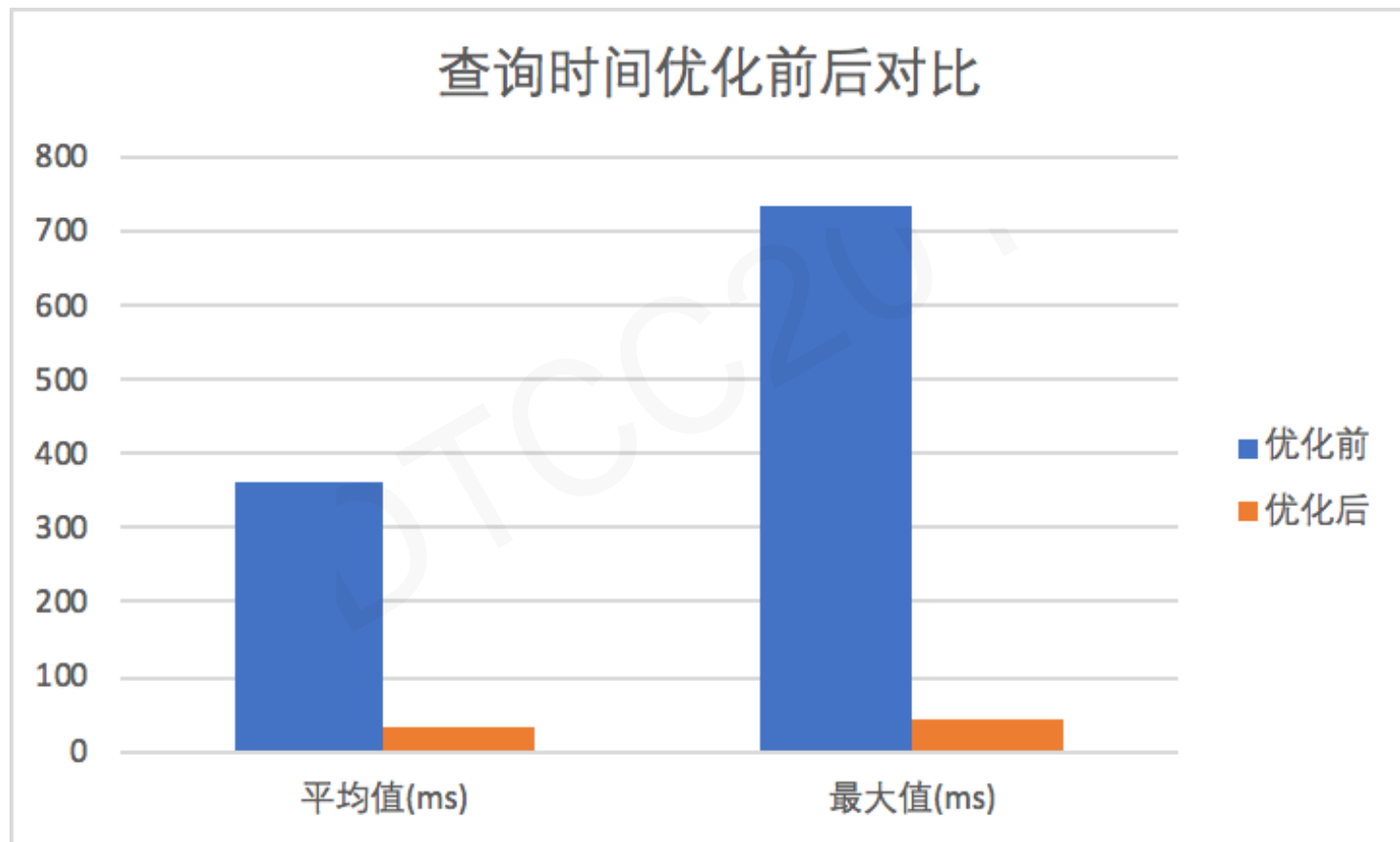


方案二：POSTGRESQL倒排索引

.....

queues		
id	submits	completes
1	{}	{A, B, C, D}

最终效果



总结

POSTGRESQL的独孤九剑

1. 瞬间添加无默认值新列
2. 并发创建索引
3. JSONB类型支持
4. 继承表
5. 条件索引
6. UDF
7. 表达式索引
8. 数组类型
9. 倒排索引

THANKS





BigDate

Spark

MySQL

DB2

SQL Server

NewSQL

Oracle

NoSQL

MPP

OLAP



讲师申请

联系电话（微信号）：18612470168

关注“ITPUB”更多
技术干货等你来拿~

与百度外卖、京东、魅族等先后合作系列分享活动



让学习更简单

微学堂是以ChinaUnix、ITPUB所组建的微信群为载体，定期邀请嘉宾对热点话题、技术难题、新产品发布等进行移动端的在线直播活动。

截至目前，累计举办活动期数60+，参与人次40000+。

ITPUB学院

ITPUB学院是盛拓传媒IT168企业事业部（ITPUB）旗下
企业级在线学习咨询平台
历经18年技术社区平台发展
汇聚5000万技术用户
紧随企业一线IT技术需求
打造全方式技术培训与技术咨询服务
提供包括企业应用方案培训咨询（包括企业内训）
个人实战技能培训（包括认证培训）
在内的全方位IT技术培训咨询服务

ITPUB学院讲师均来自于企业
一些工程师、架构师、技术经理和CTO
大会演讲专家1800+
社区版主和博客专家500+

培训特色

无限次免费播放
随时随地在线观看
碎片化时间集中学习
聚焦知识点详细解读
讲师在线答疑
强大的技术人脉圈

八大课程体系

基础架构设计与建设
大数据平台
应用架构设计与开发
系统运维与数据库
传统企业数字化转型
人工智能
区块链
移动开发与SEO



联系我们

联系人：黄老师
电话：010-59127187
邮箱：edu@itpub.net
网址：edu.itpub.net
培训微信号：18500940168