SCQL: 多方安全分析语言, Secure Collaborative Query Language

# SCQL概览

SCQL属于隐私计算BI范围，

允许多个互不信任的参与方在不泄露各自隐私的条件下进行**联合数据分析**

（BI是Business Intelligence，把数据转化为信息从而支持决策）

## 背景

有这样一些SQL语句

SELECT … FROM … JOIN … ON …

GROUP BY …

如果数据分布在不同参与方（多方数据分析的场景），那么不泄露隐私的进行统计就比较困难。

目前多方数据分析的两种技术路线：

TEE SQL (数据持有方把自己的数据加密后上传到可信执行环境内。能保障数据的安全)

MPC SQL (数据不用出域，每个数据持有方部署计算节点，用mpc协议进行计算)

方案对比：

TEE计算效率很高；但是数据出域、信任根在硬件提供商(如intel)

MPC原始数据不会泄露，但是计算效率很慢、规模有限

## SCQL概述

一个MPC SQL解决方案

系统组件：(一个中心化方案)

1. SCDB: 部署在**可信第三方**，负责将用户提交的query翻译成密态执行图，并下发。

2. SCQLEngine: 部署在数据参与方，负责协同其他参与方一起完成计算。

SCQL特征：

1. 是一个**半诚实**安全模型
2. 支持2方及以上的参与方进行计算。
3. 容易上手，提供MySQL兼容的sql方言，不懂mpc但是熟悉sql就能写
4. 支持丰富的SQL语法和算子。
5. 可以实用的性能
6. 提供列级别的数据可用授权控制CCL
7. 支持多种密态协议（semi2k，cheetah，aby3）
8. 支持多种数据源接入，如MySQL，csv等

## 应用场景

场景一，医疗数据研究。分析多家医院的医疗数据

数据分布特点为水平切分

可以通过UNION算子集合多家数据

场景二，联合精准营销。获取一些机构的交叉客户群，并进行用户画像

数据分布特点为垂直切分

先JOIN获取交叉客群，然后进行用户画像，做一些分组聚合统计

场景三，智能保险理赔。这是一个在线场景，保险机构接入医疗机构的数据

数据分布特点为垂直切分

# SCQL CCL

## CCL用途

SQL的语法比较灵活，用户可以恶意构造query获取数据

当前，业内多方数据分析的安全保障模型：

Query执行之前需要所有参与方进行审核确认。

审核需要分析query泄露数据的风险。

等待所有参与方都通过，周期比较长。

基于CCL的多方数据分析的安全保障模型：

数据持有方首先设置自己数据如何使用的限制。

通过sql检查的query会有审核员进行再次的确认，减轻审核员的负担

## CCL定义

全称是column control list，是一个三元组：

<某一列数据，某个参与方，约束条件> => <src\_column, dest\_party, constraint>

Constraint的一个案例：PLAINTEXT\_AFTER\_AGGREGATE，聚合后bob才能看到

如果不满足ccl约束，那么一定不安全

满足了ccl约束，也不一定安全

## CCL constraint

目前有六种约束：

PLAINTEXT 允许任何形式的披露

PLAINTEXT After join 作为inner join的key，在join后可以明文披露。

PLAINTEXT After group by 分组后，可以明文披露

PLAINTEXT After aggregate 经过聚合操作后的结果可以明文披露

PLAINTEXT After compare 经过比较操作之后的结果可以披露

Encrypted only 除了count以外，不会披露

# SCQL架构

## 具体架构

用户 -> trusted third party

* Party I (DB + SCQL Engine)

trusted third party的内部结构：

* Session manager 校验身份信息
* Parser
* Planner (DDL+DCL去对应的executor，DQL继续往下)
* Translator 翻译之前检查CCL，通过后继续执行（会自动选择最佳协议去执行）
* Graph Optimizer
* Graph Splitter 每个参与方看到自己的子图

## 执行流程

1. 用户发起DQL给SCDB
2. SCDB下发密态执行图
3. 数据提供方读取数据
4. SCQL Engine执行MPC计算协议
5. SCQL Engine上报计算结果
6. SCDB返回执行结果