

关系代数 and 数据库的问题与展望

创新技术推动应用进步

主讲人:蒋步星

RAQSOFT
RAQSOFT
RAQSOFT

目录

contents



数据计算与关系代数



关联查询



结构化数据计算



大数据计算问题



库外计算引擎



数据仓库与云数据库



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

知道解法写出代码也不容易

利率变动情况下房贷还剩多少本金

程序员在做什么

将解决问题的思路翻译成计算机可执行的精确的形式语言的过程

小学生解应用题：把解答翻译成四则运算

代码为什么难写

用于记录解法的形式语言与思维相差太远，不能直接描述思路

翻译问题解法的难度经常远远超过解决问题本身

形式语言的代数体系



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

形式语言采用的代数体系是关键因素

严重影响数据处理的能力和效率

代数体系

数据类型及其上的运算规则，确保封闭性和逻辑自治

不够好的代数体系：罗马数字、汇编语言

```
mov ax,3  
mov bx,5  
mul bx,7  
add ax,bx
```

$N=3+5*7$



结构化数据仍是现代数据分析计算最重要的内容

数据计算和分析的目的是记录和寻找事物之间的关联，而事物是由其属性决定的，技术上表现为结构化数据

当前结构化数据计算的代数体系主要是关系代数

数据类型：简单属性的对象集合

运算：交并差、过滤/补充、分组聚合、连接

关系数据库和SQL



计算是数据库的主要目标

理论基础是关系代数

形式化语言是SQL

很多大数据方案也将实现SQL作为目标

SQL有较大改善余地

描述计算难度大：*连续上涨3天以上的股票*

实施运算性能低：*从1亿记录中取前10名*

原因

不提倡分步运算

缺乏离散性

集合化不彻底

无序集合

关联定义太简单



目录

contents



数据计算与关系代数



关联查询



结构化数据计算



大数据计算问题



库外计算引擎



数据仓库与云数据库



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

关联查询



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

SQL处理单表简单查询问题不大

有意义的查询经常是多表关联

SQL中表关联用JOIN运算实现

关系代数对JOIN运算的定义过于简单通用

笛卡尔积后再过滤

多表查询的理解和编写难度大



外键引用



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

查询目标：中国经理的美国员工

```
1 SELECT A.* FROM 员工表 A,部门表 B,员工表 C
2     WHERE A.国籍='美国'
3     AND C.国籍='中国'
4     AND A.部门=B.编号
5     AND B.经理=C.编号
```

```
1 SELECT * FROM 员工表
2     WHERE 国籍='美国' AND 部门.经理.国籍='中国'
```

员工表

编号

姓名

国籍

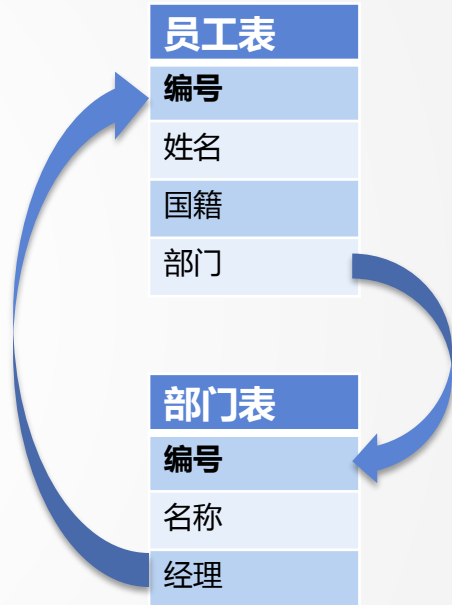
部门

部门表

编号

名称

经理



同维表对齐



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

查询目标：所有员工的收入

```
1 SELECT 员工表.姓名,员工表.工资+经理表.津贴
2       FROM 员工表
3       LEFT JOIN 经理表 ON 员工表.编码=经理表.编号
```

```
1 SELECT 姓名,工资+津贴 FROM 员工表
```

员工表

编号

姓名

性别

工资

经理表

编号

津贴



主子表对齐



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

查询目标：订单及其客户与金额

```
1 SELECT 订单表.编号, 订单表.客户, SUM(订单明细.价格)
2       FROM 订单表
3     LEFT JOIN 订单明细 ON 订单表.编码=订单明细.编号
4     GROUP BY 订单表.编号, 订单表.客户
```

```
1 SELECT 编号, 客户, 订单明细.SUM(价格) FROM 订单表
```

订单表

编号

客户

日期

...

订单明细

编号

序号

产品

价格



汇总后对齐



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

查询目标：按日期统计合同额、回款额和库存额

```
1 SELECT T1.日期,T1.金额,T2.金额 FROM
2 (SELECT 日期, SUM(金额) 金额 FROM 合同表 GROUP BY 日期 ) T1,
3 (SELECT 日期, SUM(金额) 金额 FROM 回款表 GROUP BY 日期 ) T2 ,
4 (SELECT 日期, SUM(金额) 金额 FROM 库存表 GROUP BY 日期 ) T3
5 WHERE T1.日期 = T2.日期 AND T2.日期=T3.日期
```

```
1 SELECT 合同表.SUM(金额),回款表.SUM(金额),库存表.金额 ON 日期
   FROM 合同表 BY 日期 JOIN 回款表 BY 日期 JOIN 库存表 BY 日期
```

合同表

编号

日期

客户

金额

回款表

序号

日期

来源

金额

库存表

序号

日期

数量

金额



混合情况



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

查询目标：按地区统计销售员人数和合同额

```
1 SELECT T1.地区,T1.数量,T2.金额 FROM
2 (SELECT 地区,COUNT(1) 数量 FROM 销售员 GROUP BY 地区 ) T1,
3 (SELECT 客户表.地区 地区,SUM(合同.金额) 金额 FROM 客户表,合同表
4 WHERE 客户表.编号=合同表.客户 GROUP BY 客户表.地区 ) T2
5 WHERE T1.地区 = T2.地区
```

```
1 SELECT 销售员.count(1),合同表.sum(金额) ON 地区
   FROM 销售员 BY 地区 JOIN 合同表 BY 客户.地区
```

合同表

编号

日期

客户

金额

销售员

编号

姓名

地区

客户表

编号

名称

地区



区分JOIN！



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

1

外键引用

外键属性化

2

同维主子表对齐

同维主子表等同化

3

汇总对齐

按维度自动对齐



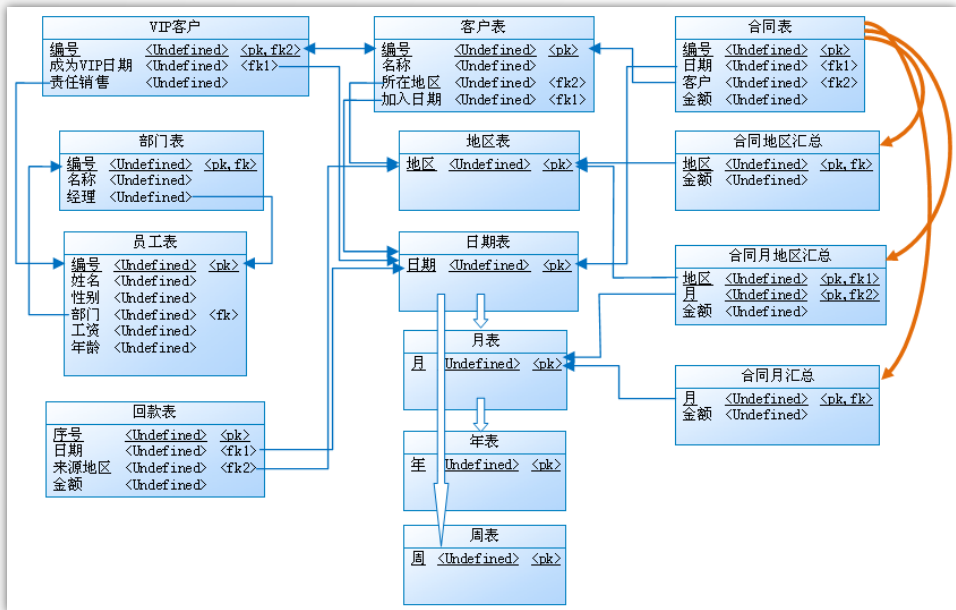
低耦合的数据模型



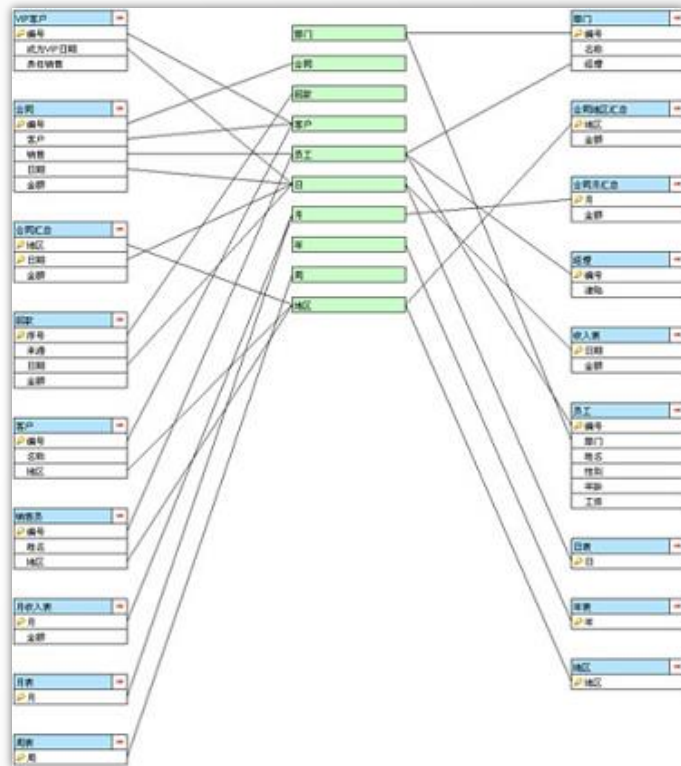
中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

网状结构的E-R图



线状结构的维关系图



关联查询

上周 新增用户数, 10月 北京 合同额 回款额, ...

1. 关键词与字段匹配, 拼SELECT和WHERE

2. 寻找表间关联路径, 拼FROM和JOIN

SQL的JOIN机制下第2步非常困难

各种关联关系混在一起分不清

改变JOIN设计后关联清晰简单

外键、同维/主子表都变成了“单表”

汇总合并只要找到共同维度去对齐

本月消费超过20元的用户

```
SELECT T_1.PRODUCT_NO "用户",  
T_1_1.HF0004 "消费" FROM  
DM_AUTORPT_BASE_INFO T_1 LEFT  
JOIN DM_AUTORPT_HIS_INDEX T_1_1  
ON T_1.PRODUCT_NO =  
T_1_1.PRODUCT_NO AND T_1.YM =  
T_1_1.YM WHERE  
(MONTH(T_1.OP_TIME) =  
MONTH(CURRENT_TIMESTAMP)) AND  
(T_1_1.HF0004 > 20)
```

其它优势



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

降低出错率

避免无业务意义的完全叉乘

避免无关维度对齐

避免非维度随意分组

提高性能

维表内存化、序号化、多冗余

同维表主子表同序归并

汇总数据建设指导



目录

contents



数据计算与关系代数



关联查询



结构化数据计算



大数据计算问题



库外计算引擎



数据仓库与云数据库



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术



结构化数据计算



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
资深IT专家

1

集合化

2

离散性



3

更彻底的集合化

4

有序计算



集合化



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

集合化

结构化数据总是批量形式

SQL是集合化的语言

WHERE, ORDER BY, GROUP
INTERSECT, UNION, MINUS

集合运算库

lambda语法

动态数据结构



离散性



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

离散性

集合成员可以游离在集合外存在

独立运算或及其它游离成员再组合新集合运算

SQL缺乏离散性

只有单行记录的表，没有游离记录

计算结果总是新产生，和原记录无关



计算任务：张三和李四的年龄差与工资差

1	SELECT (SELECT age FROM employee WHERE name= '张三')
2	- (SELECT age FROM employee WHERE name= '李四')
3	FROM dual
4	SELECT (SELECT salary FROM employee WHERE name= '张三')
5	- (SELECT salary FROM employee WHERE name='李四')
6	FROM dual

	A
1	=employee.select@1(name=="张三")
2	=employee.select@1(name=="李四")
3	=A1.age-A2.age
4	=A1.salary-A2.salary

指定条件更新数据

计算任务：业绩在前10%的代理商再奖励5%业绩

	A	
1	=agent.sort(amount:-1).to(agent.len()*0.1)	取前10%
2	=A1.run(amount=amount*1.05)	奖励销售额

引用式外键

计算任务：选出北京地区的交易记录

	A	
1	>交易记录.switch(地区,地区表:编号)	建立外键引用
2	=交易记录.select(地区.名称=="北京")	用外键引用记录的字段过滤

更彻底的集合化



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

彻底的集合化需要离散性的支持
允许游离成员组成新集合

分组子集
有序聚合

逆分组
主子表

.....



分组子集（一）



计算任务：找出总分500分以上的学生各科成绩记录

```
1 WITH T AS
2 (SELECT 学生 FROM 成绩表 GROUP BY 学生 HAVING SUM(成绩)> 500)
3 SELECT TT.* FROM T LEFT JOIN 成绩表 TT on T.学生=TT.学生
```

SQL没有游离记录构成的显式集合，无法保持分组结果，必须强迫聚合有离散性后才能还原分组运算的本来面目

	A
1	=成绩表.group(学生).select(~.sum(成绩)>=500).conj()



分组子集（二）



计算任务：用户在最后一次登录前三天内的登录次数

```
1 WITH T AS
2     (SELECT uid,max(logtime) last FROM 登录表 GROUP BY uid)
3 SELECT T. uid,T.last,count(TT.logtime)
4     FROM T LEFT JOIN 登录表 TT ON T.uid=TT.uid
5     WHERE T.last-TT.logtime<=3 GROUP BY T.uid,T.last
```

针对分组子集的聚合运算较复杂时难以用简单聚合式写出
保留分组子集再结合分步计算则很容易

	A
1	=登录表. group(uid;~.max(logtime):last,~.count(interval(logtime,last)<=3):num)



有序聚合（一）



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

计算任务：列出用户首次登录的记录

```
1 SELECT * FROM  
2     (SELECT RANK() OVER(PARTITION BY uid ORDER BY logtime) rk, T.* FROM 登录表 T) TT  
3 WHERE TT.rk=1;
```

聚合运算不一定总是SUM/COUNT这些，还可以理解为取按某个次序成员有离散性时可以简单针对分组子集实施有序聚合

	A
1	=登录表.group(uid).(~.maxp(logtime))
	A
1	=登录表.sort(logtime:-1).group@1(uid)



有序聚合（二）



计算任务：列出每个用户最近一次登录间隔

```
1 WITH TT AS
2     (SELECT RANK() OVER(PARTITION BY uid ORDER BY logtime DESC) rk, T.* FROM 登录表 T)
3 SELECT uid,(SELECT TT.logtime FROM TT where TT.uid=TTT.uid and TT.rk=1)
4         -(SELET TT.logtim FROM TT WHERE TT.uid=TTT.uid and TT.rk=2) 间隔
5 FROM 登录表 TTT GROUP BY uid
```

聚合函数返回值不一定是单值，也可以返回一个集合

彻底的集合化后很容易针对分组子集实施返回集合的聚合运算

	A	
1	=登录表.groups(uid;top(2,-logtime))	最后2个登录记录
2	=A1.new(uid,#2(1).logtime-#2(2).logtime:间隔)	计算间隔



关系代数延用了数学上的无序集合概念

早期SQL要生成序号后JOIN才能实现有限的有序计算

计算股票的涨幅

```
1 WITH T AS
2   (SELECT rownum 序号,交易日,收盘价
3    FROM ( SELECT * FROM 股票 ORDER BY 交易日) )
4 SELECT T1.交易日,T1.收盘价-T2.收盘价
5 FROM T T1 JOIN T T2 ON T1.序号=T2.序号+1
```

SQL2003标准增加了窗口函数方便生成序号并引用相邻成员

计算股票的涨幅（用窗口函数）

```
1 SELECT 交易日,收盘价-LAG(收盘价) OVER (ORDER BY 交易日)
2 FROM 股票
```

有序计算



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

有序计算是集合化与离散性的结合物

运算不仅与数据本身有关，还和数据所在集合的位置有关

跨行引用

有序分组

位置利用



跨行引用（一）



中生代技术
FRESHMAN TECHNOLOGY



计算任务：选出比上月销售量和销售额均高出10%的产品销售记录

```
1 WITH T AS
2   (SELECT 销售量/LAG(销售量) OVER(PARTITION BY 产品 ORDER BY 月份) r1
3   (SELECT 销售额/LAG(销售额) OVER(PARTITION BY 产品 ORDER BY 月份) r2,
4   A.*, FROM 销售表 A)
5 SELECT * FROM T WHERE r1>1.1 AND r2>1.1
```

窗口函数必须使用子查询才能实现跨行计算，多个跨行项要分别使用窗口函数
集合有序后可直接提供跨行引用的方案

	A	
1	=销售表.sort(产品,月份)	
2	=A1.select(if(产品==产品[-1],销售量/销售量[-1])>1.1 && 销售额/销售额[-1])>1.1)	



跨行引用（二）



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

计算任务：计算每月前后各一个月的销售额移动平均值

```
1 WITH B AS
2     (SELECT LAG(销售额) OVER (ORDER BY 月份) f1, LEAD(销售额) OVER (ORDER BY 月份) f2, A.* FROM 销售表 A)
3 SELECT 月份,销售额,
4     (NVL(f1,0)+NVL(f2,0)+销售额)/(DECODE(f1,NULL,0,1)+DECODE(f2,NULL,0,1)+1) 移动平均
5 FROM B
```

窗口函数只有简单的跨行引用，涉及集合要用成员去拼

有序集合上可提供跨行集合引用方案

	A	
1	=销售表.sort(月份).derive(销售额{-1,1}.avg()).移动平均)	



有序分组（一）



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

计算任务：一批婴儿中连续出生的同性别人数超过5人的有多少组

```
1 SELECT COUNT(*) FROM
2   (SELECT 改变次数 FROM
3     (SELECT SUM(改变标志) OVER ( ORDER BY 生日) 改变次数 FROM
4       (SELECT CASE WHEN 性别=LAG(性别) OVER (ORDER BY 生日) THEN 0 ELSE 1 END 改变标志 FROM 婴儿
5        表) )
6     GROUP BY 改变次数 HAVING COUNT(*)>=5)
```

分组可能和次序相关，不只是等值分组

有序集合上可以定义与次序有关的分组，考察值变化时即产生新组

	A	
1	=婴儿表.sort(生日).group@o(性别).count(~.len())>=5)	



有序分组（二）



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

计算任务：一支股票最长连续上涨了多少天

```
1 SELECT max(连续日数) FROM
2     (SELECT count(*) 连续日数 FROM
3         (SELECT SUM(涨跌标志) OVER ( ORDER BY 交易日) 不涨日数 FROM
4             ( SELECT 交易日,
5                 CASE WHEN 收盘价>LAG(收盘价) OVER( ORDER BY 交易日 THEN 0 ELSE 1 END 涨跌标
6                 志
7             FROM 股票 ))
9         GROUP BY 不涨日数)
```

另一种和次序有关的分组，条件成立时产生新组

	A	
1	=股票.sort(交易日).group@i(收盘价<收盘价[-1]).max(~.len())	



位置利用（一）



计算任务：计算一组商品价格的中位数

```
1 WITH N AS (SELECT COUNT(!) FROM 价格表)
2 SELECT AVERAGE(价格) FROM
3     (SELECT 价格,ROW_NUMBER() OVER (ORDER BY 价格) r FROM 价格表 ) T
4     WHERE r=TRUNC((N+1)/2) OR r=TRUNC(N/2)+1)
```

SQL的无序集合需要人为制造出序号，不分步运算加剧了困难

有序集合可以直接用序号访问成员

	A
1	=价格表.sort(价格).([(价格表.len()+1)\2,价格表.len()\2+1]).avg()



位置利用（二）



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

计算任务：某股票股价最高的那三天的平均涨幅

```
1 SELECT AVG(涨幅) FROM
2   ( SELECT 交易日, 收盘价-LAG(收盘价) OVER ( ORDER BY 交易日) 涨幅 FROM 股价表
3     WHERE 交易日 IN
4           (SELECT 交易日 FROM
5             (SELECT 交易日, ROW_NUMBER() OVER(ORDER BY 收盘价 DESC) 排名 FROM 股票)
6           WHERE 排名<=3 )
```

无序集合不能利用位置访问相邻成员，计算量增大

有序集合可以提供丰富的按位置访问机制

	A	
1	=股票.sort(交易日)	
2	=A1.calc(A1.ptop(3,-收盘价),收盘价-收盘价[-1]).avg()	



结构化数据计算总结



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

离散性与集合化的有效结合

集合化是批量计算的基本能力

离散计算也不可或缺

离散性支持更彻底的集合化

离散性产生有序集合运算

离散数据集模型

=

集合运算

+

游离成员

=>

彻底集合化/有序集合



目录

contents



数据计算与关系代数



关联查询



结构化数据计算



大数据计算问题



库外计算引擎



数据仓库与云数据库



中生代技术
FRESHMAN TECHNOLOGY

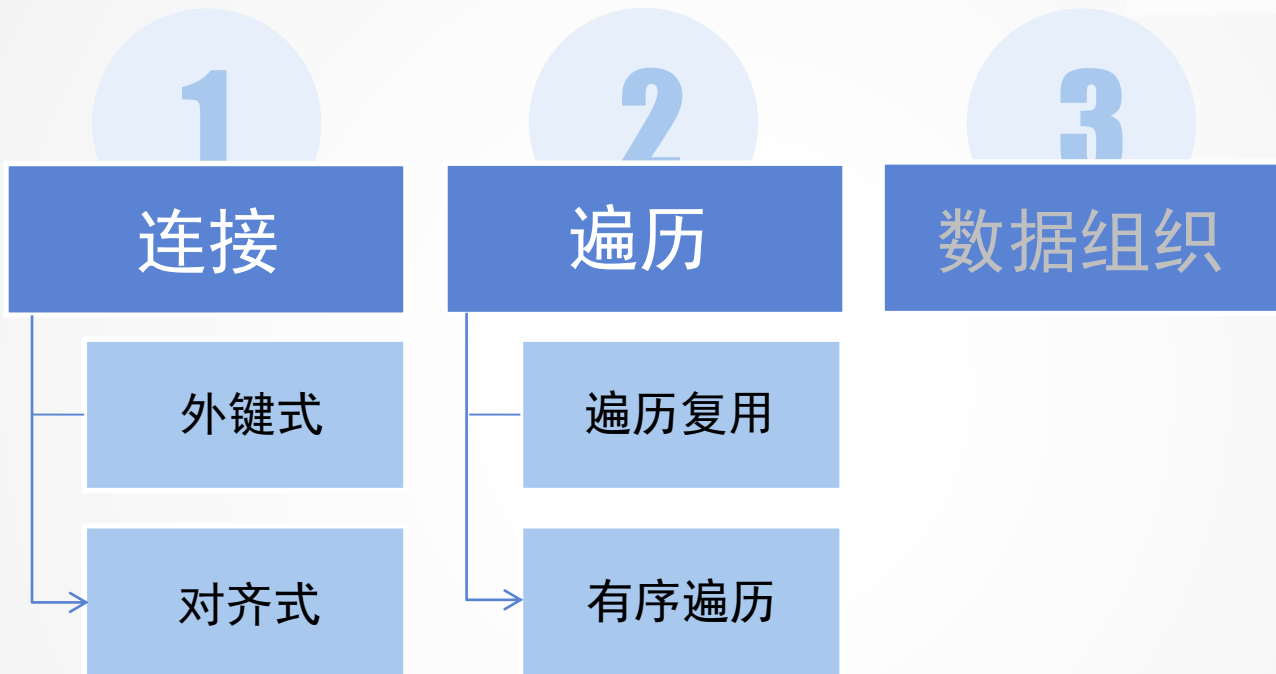
IT大咖说
不止于技术

大数据计算问题



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术



外键指针化



外键是随机小量频繁访问

内存指针查找大幅提高性能

SQL缺乏离散性，不能提供指针引用

				Java指针连接	Oracle
	A		单表无连接	0.57s	0.623s
1	=file("Products.txt").import()	读入商品列表	五表外键连接	2.3s	5.1s
2	=file("Sales.txt").import()	读入销售记录			
3	> A2.switch(productid,A1:id)	建立指针式连接，把商品编号转换成指针			
4	=A2.sum(quantity*productid.price)	计算销售金额，用指针方式引用商品单价			

序号化相当于外存指针化

	A	
1	=file("Products.txt").import()	读入商品列表
2	=file("Sales.txt").cursor()	根据已序号化的销售记录建立游标
3	=A2.switch(productid,A1:#)	用序号定位建立连接指针，准备遍历
4	=A3.groups(;sum(quantity*productid.price))	计算结果

SQL无序集合，事先序号化无效

有序归并



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

对齐式连接可以先排序后变成有序归并

同维表、主子表

	A	
1	=file("Order.txt").cursor@t()	订单游标，按订单id排序
2	=file("Detail.txt").cursor@t()	订单明细游标，也按订单id排序
3	=join@x(A1:O,id;A2:D,id)	有序归并连接，仍返回游标
4	=A3.groups(O.area;sum(D.amount))	按地区分组汇总金额，地区字段在主表中，金额字段在明细子表中

追加数据的再排序也仍然是低成本的归并计算

易于分段并行

SQL的无序集合机制不能保存有序结果



小维表复制存入节点内存

大维表分段存入节点内存

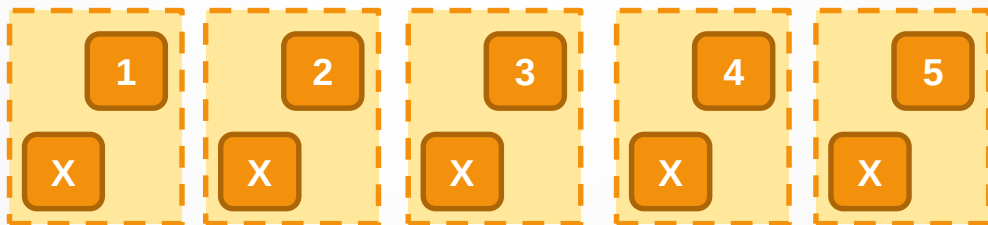
同维主子表分段存入节点外存

JOIN总结

外键式：小维表、大维表

对齐式：有序归并

避免低效率Hash分段算法



外存计算优化方向是减少访问量

可复用的遍历减少外存访问量

	A	
1	=file("data.txt").cursor()	
2	=channel().groups(;count(1))	配置同步计算
3	> A1.push(A2)	绑定
4	=A1.sortx(key)	排序，遍历过程中处理绑定计算
5	=A2.result().#1	取出绑定计算的结果，即总记录数
6	=A4.skip((A5-1)\2).fetch@x(2-A5%2).avg(key)	取出中位数记录并计算中位数

多分组方案

SQL语法体系不支持一次遍历返回多个结果

有序游标



复杂处理需要读出数据库实施

有序游标技术有效减少查找和遍历数量

	A	
1	=file("userlog.txt").cursor@t()	原始数据游标
2	=A1.sortx(id)	按id排序并返回游标
3	>file("userlog.dat").export@z(A2;id)	写成按id分段的二进制文件

	A	B	C
1	=file("userlog.dat")		对id有序的数据文件
2	fork 4	=A1.cursor@b(;A2:4)	分段时保证同一组id不会被拆开
3		for B2;id	... 后续计算语法相同
4			...

关系数据库的游标没有按序取出的功能

目录

contents



数据计算与关系代数



关联查询



结构化数据计算



大数据计算问题



库外计算引擎



数据仓库与云数据库



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

- ▶ 数据库计算的封闭性
- ▶ 库外计算不能实施
- ▶ 跨库计算困难
- ▶ 库外计算引擎
- ▶ 开放性
- ▶ 可集成，计算无处不在

减少存储过程



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

▶ 存储过程的目的

- ▶ 数据整理
- ▶ 呈现准备

▶ 存储过程的问题

- ▶ 应用内与应用间耦合
- ▶ 安全性与易管理性

▶ 库外计算替代存储过程



冗余中间表



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

中间表的由来

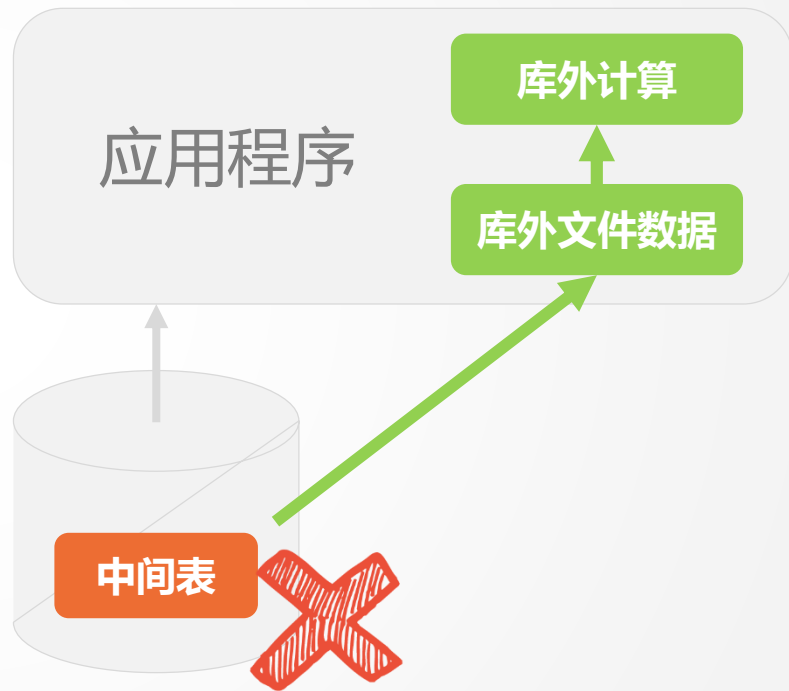
- ▶ 运算复杂或数据量大
- ▶ 再次计算的能力

中间表的问题

- ▶ 数量众多占用数据库资源
- ▶ 线性结构导致管理困难

库外计算将中间数据外置

- ▶ 计算不依赖于数据库，其它能力不需要
- ▶ 绑定应用、树状结构、易于管理



多样性数据



中生代技术
FRESHMAN TECHNOLOGY

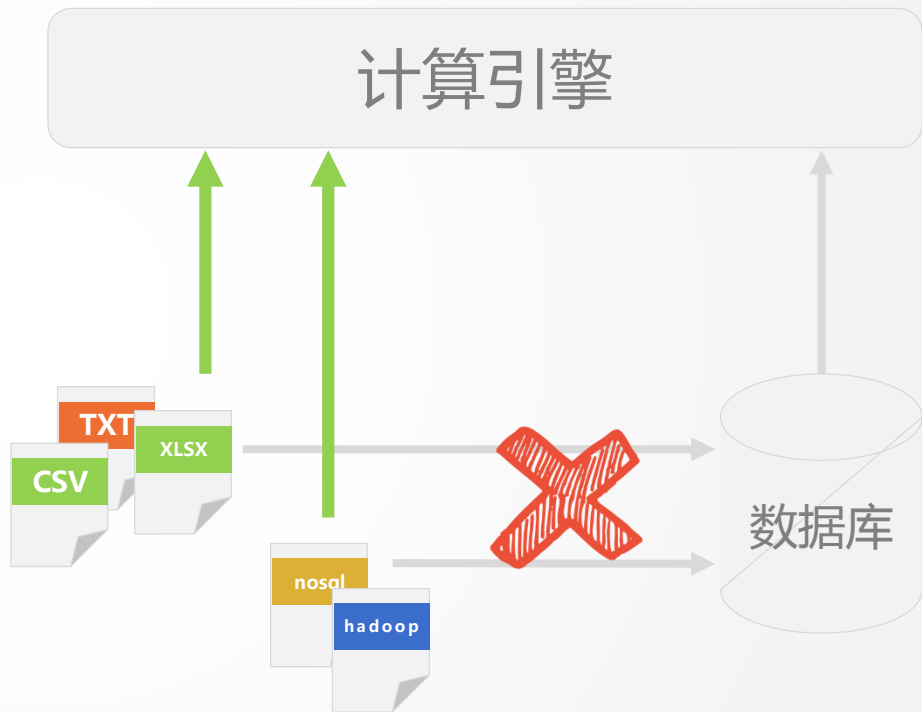
IT大咖说
不止于技术

多样性数据普遍存在

- ▶ txt, csv, xlsx
- ▶ nosql, hadoop
- ▶ json, xml

直接计算多样性数据

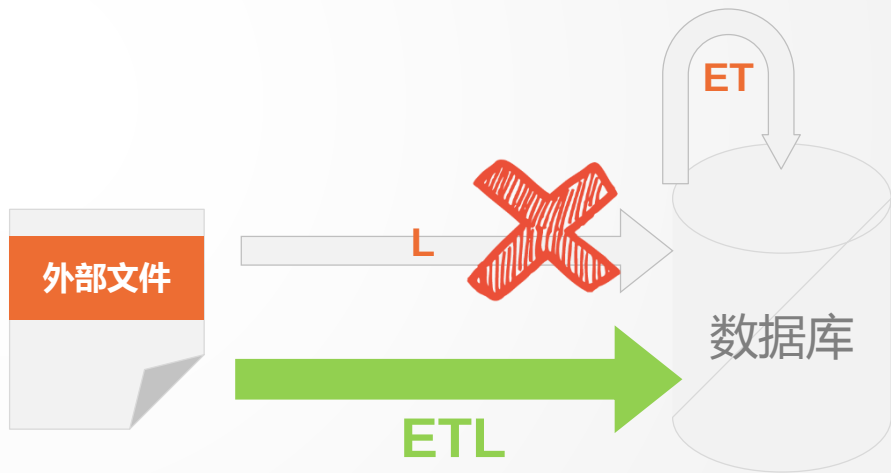
- ▶ 不需要建设专门的数据库及转入工作
- ▶ 结构简单、实时性更好



▶ 缺乏库外计算能力扰乱ETL过程

- ▶ ETL ? ELT ? LET ?
- ▶ 加大数据库负担

▶ 库外计算实现合理的ETL



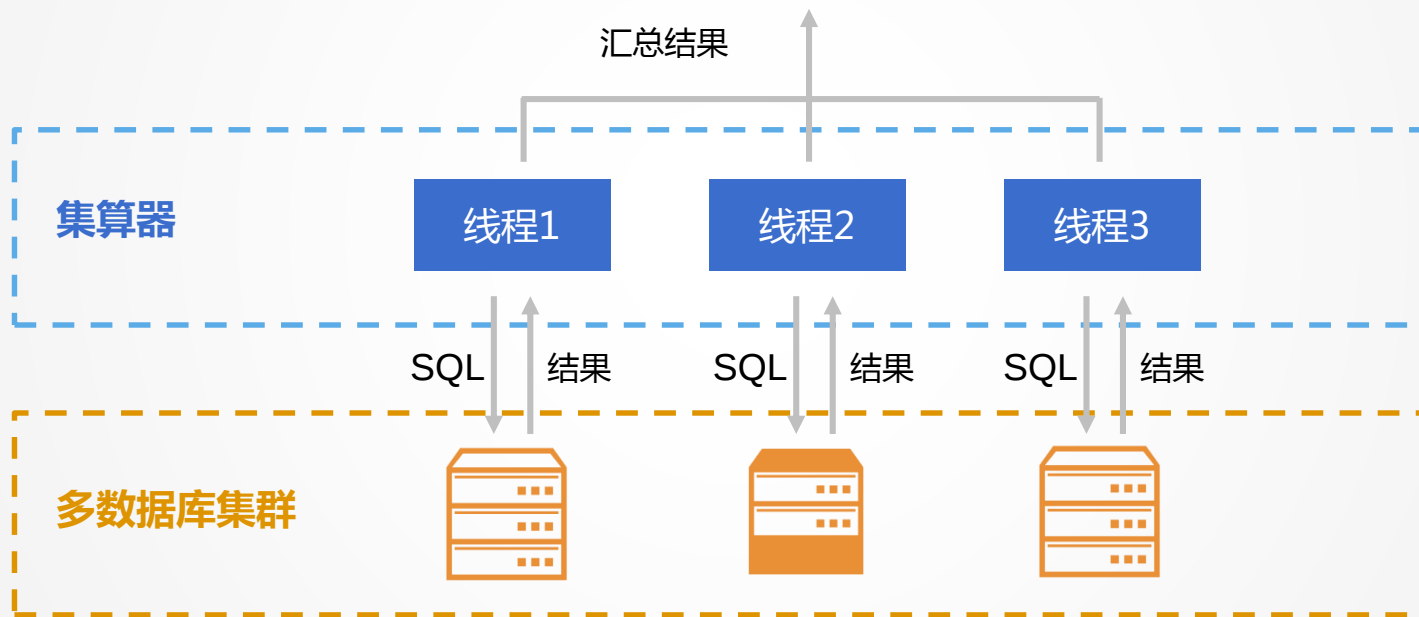
跨库集群



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

► 异构数据库集群



T+0查询报表



中生代技术
FRESHMAN TECHNOLOGY

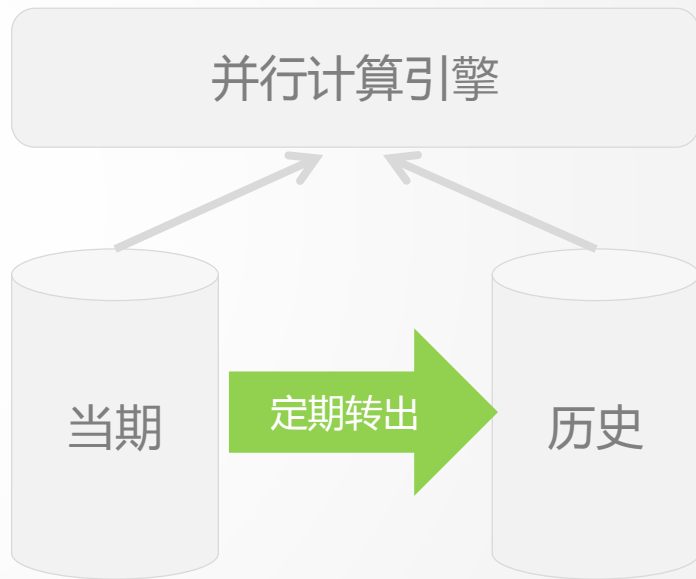
IT大咖说
不止于技术

▶ T+0问题

- ▶ 交易一致性要求关系数据库
- ▶ 历史与当期同库，数据量太大
- ▶ 历史与当期异库，跨库计算困难

▶ 库外计算实现并行跨库计算

- ▶ 历史数据还可文件化



▶ 数据中心的特征与要求

- ▶ 数据库群
- ▶ 多样性数据源
- ▶ 服务式接口
- ▶ 访问受控
- ▶ 数据脱敏

库外计算引擎实现数据库中心访问层



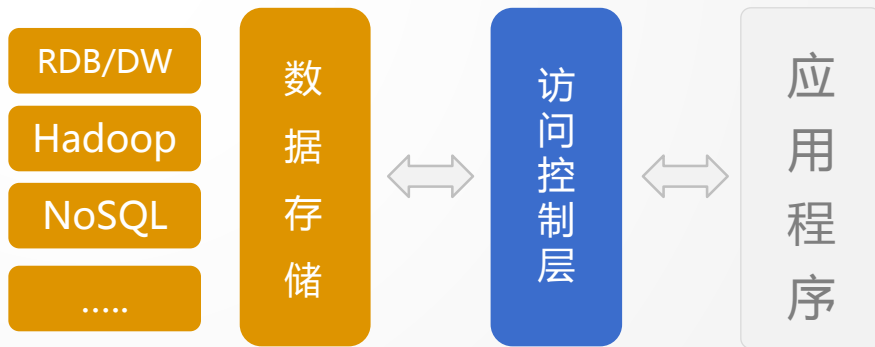
计算能力



编程能力



集成能力



目录

contents



数据计算与关系代数



关联查询



结构化数据计算



大数据计算问题



库外计算引擎



数据仓库与云数据库



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

数据库与数据仓库的不同



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

数据仓库/OLAP	数据库/OLTP
大数据	单任务小数据量
低并发	高并发
无一致性	强一致性
强计算	弱计算
可预先整理	多样性

关系数据库把两种不同需求融合到一起并不合适



解决前述问题

关联问题

离散性

彻底集合化

有序计算支持

提倡分步计算

中间结果再复用

避免子查询，降低难度

数据组织



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

组合表

多数据结构混存

同维表与主子表统一

消除对齐式连接

强数据结构，有模式机制

集合数据类型

北京		面积	人口	GDP	...
	海淀				
	朝阳				
	...				
上海					
	浦东				
	...				
...					

订单		客户	地区	日期	...
	订单明细	产品	数量	单价	...



重整主键



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

主键序号化

外存指针式外键，高速引用

外存游离记录表示，离散性

性能优化

分组结果自然对齐有序

维排序可用基数排序法

避免连接中的Hash计算

北京		1
	海淀	1,1
	朝阳	1,2
	...	
上海		2
	浦东	2,1
	...	
...		



多样性需求

同类数据的数据结构不完全相同

业务、空间与时间多样性

确保计算能力

多样性数据可以一起计算

当前技术

关系数据库不支持多样性

nosql牺牲计算能力

多样性与高性能的矛盾



局域性

数据分区域管理和操作

高性能：域内数据的高速寻址

封闭性：操作本域数据时不会影响和被域外数据影响

局域性的特征

域边界由业务决定

多层次

局域性强化多样性

当前技术

关系数据库和nosql都缺乏局域性



一致性



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

一致性

软一致性（写一致性 / 读一致性）

硬一致性

低成本一致性的实现

业务粒度：数据访问粒度由业务决定

关系数据库采用技术粒度且缺乏局域性，一致性实现方案成本高

nosql几乎都没有一致性



润乾【华东】技术交流群



- ✓ 润乾数据产品技术咨询/支持
- ✓ 技术资料共享
- ✓ 线下技术活动交流
- ✓



润乾
RAQSOFT

创新技术 推动应用进步

www.raqsoft.com.cn

谢谢大家！