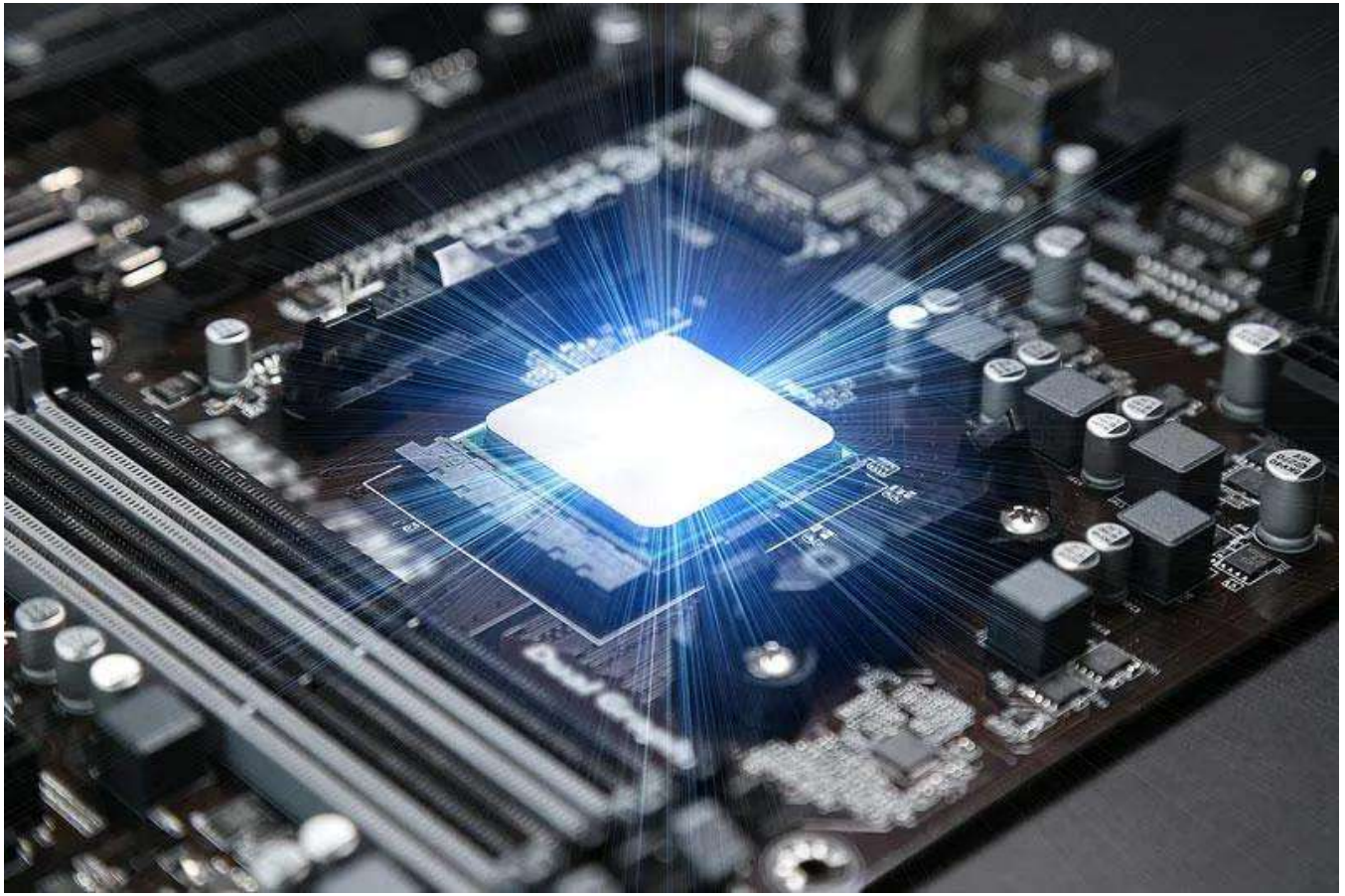
中山大學
SUN YAT-SEN UNIVERSITY



2021级《数据库原理与应用》第6周

2024.4.3

中山大学数学学院 黄志洪

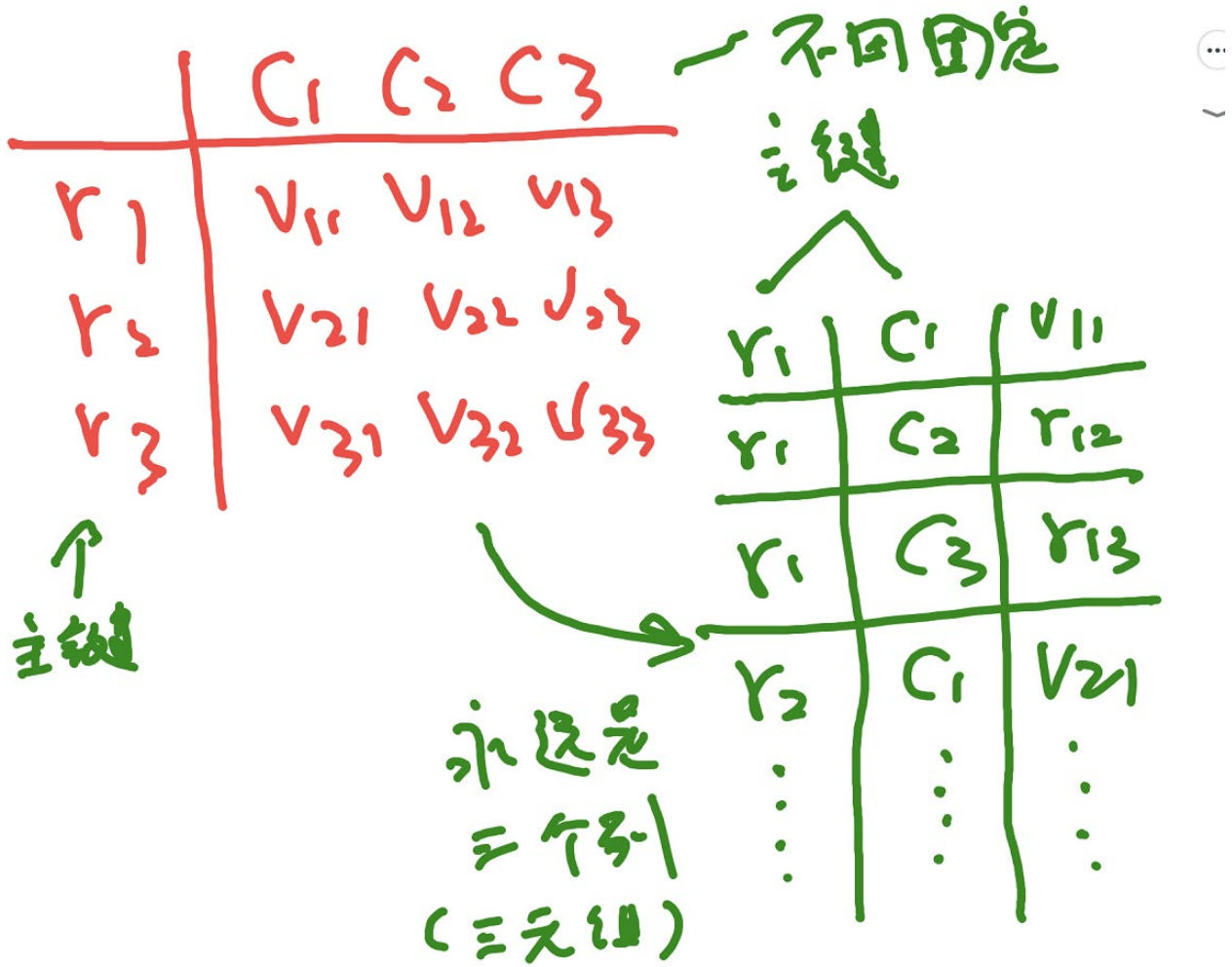
窄表的魔术：从关系型数据库的窘境说起

- 关系型数据库的建模需要有**固定的模式**，也就是先要把列固定下来才可以建表，但有些场景下，列是很难被固定的。
- 例子：设备资产记录表，记录公司里不同的设备，包括家具，电视，电脑，传真机等。每种设备都要记录本身的属性，不同设备具有不同属性，而且也无法预测将来会增加什么新设备
- 在使用常见的宽表的情况下，无法固定在一张表例，要么针对每种设备建立一张新表，要不不断给表不断增加新列。无论哪种方法，我们都无法固定一个可交付的应用给用户，用户需要长期雇佣DBA维护系统，代码也处于永远在修改的状态

思路

- 一切表都可以转变成窄表，进一步地可以转变为两列式的统一格式的窄表，于是k-v型数据库就产生了
- 为什么需要分布式数据库？（因为数量很大的时候如果使用超级计算机成本太高）k-v型数据库特别适合做成分布式数据库。分布式数据库需要满足哪些特性？（C一致性，A高可用性，P分区容忍性）
- CAP定律：C、A、P三者不可能同时具备（一个简单的证明），由此可知像Oracle这样的数据库是不能在保持原有全部功能的情况下做成分布式集群的。NOSQL数据库一般都放弃某种特性，通常是把C变弱

变形记：用窄表解决问题



2024.4.3

转变为k-v形式的表

表 abc

主键	C_1	C_2	C_3
K_1	V_{11}	V_{12}	V_{13}
K_2	V_{21}	V_{22}	V_{23}
K_3	V_{31}	V_{32}	V_{33}
K_4	V_{41}	V_{42}	V_{43}

当为K-V

key	value
表名.主键.列名	值
abc.k1.c1	V11
abc.k3.c3	V33
abc.k4.c2	V42

一切表都可以转变为K-V形式

故K-V数据库也可以区分表

bigtable中的time stamp用于保留多个版本数据，也可用物理删除也可以以以上操作

Google BigTable的思想

- 存放所有数据的一张大表
- 每行由rowkey，columnkey，timestamp，string组成
- String的内容由应用解释
- 把所有DML操作都归结为insert
- 行数非常多的窄表设计非常适合各种数据，包括结构，半结构，非结构化数据，可以通过分布式哈希算法映射到分布式集群
- Group by归结为map-reduce操作（根据这个思想后来进化出hadoop）
- 不支持join，但可以把join转化为一系列key value查询操作（以emp，dept表为例），但join操作肯定变得不方便了
- Hbase，cassandra是Big Table思想的开源实现（BigTable是Google内部使用的技术并不对外公开）

Key-Value数据库（K-V db）

- 把现实世界的所有操作都归结为输入Key，查出Value
- 是极简的数据库形式
- 适合做成分布式集群数据库
- 擅长OLTP，不擅长OLAP

大数据

- 什么是大数据？任何简单的场景，只要量上去了，都会变得非常复杂
- 为什么需要分布式集群（以提供可线性增长的计算力和存储能力）而不是巨型计算机？

C：Consistency

- 在分布式系统中，一致性（Consistency）是指多副本（Replications）问题中的数据一致性。即访问所有的节点得到的数据应该是一样的
- https://mp.weixin.qq.com/s?__biz=MzI0MTU0ODQwMQ==&mid=2247485966&idx=1&sn=06281fb2e5bb3089dd2f897dd24b09c5&chksm=e908a710de7f2e06bbd7fdc91347890b3bd3768d0ee5a366b1b7d5912f85069503a19dab7ea0&scene=27

C：Consistency

- **强一致性 与 弱一致性**：强一致性（线性一致性）即复制是同步的，弱一致性：即复制是异步的
- **最终一致性**不保证在任意时刻任意节点上的同一份数据都是相同的，但是随着时间的迁移，不同节点上的同一份数据总是在向趋同的方向变化。最终两个字用得很微妙，因为从写入主库到反映至从库之间的延迟，可能仅仅是几分之一秒，也可能是几个小时。简单说，就是在一段时间后，节点间的数据会最终达到一致状态。
- https://mp.weixin.qq.com/s?__biz=MzI0MTU0ODQwMQ==&mid=2247485966&idx=1&sn=06281fb2e5bb3089dd2f897dd24b09c5&chksm=e908a710de7f2e06bbd7fdc91347890b3bd3768d0ee5a366b1b7d5912f85069503a19dab7ea0&scene=27

C：Consistency

- CAP定律中的“C”指的是“强一致性”
- 由于CAP定律说明CAP不能同时满足，通常把C从强一致性放宽到弱一致性

A：Availability

- 所有的节点都保持高可用性，即用户（很高概率地）至少能从分布式集群一个节点访问到数据。注意，这里的高可用还包括不能出现延迟，比如如果节点B由于等待数据同步而阻塞请求，那么节点B就不满足高可用性。也就是说，任何没有发生故障的服务必须在有限的时间内返回合理的结果集。

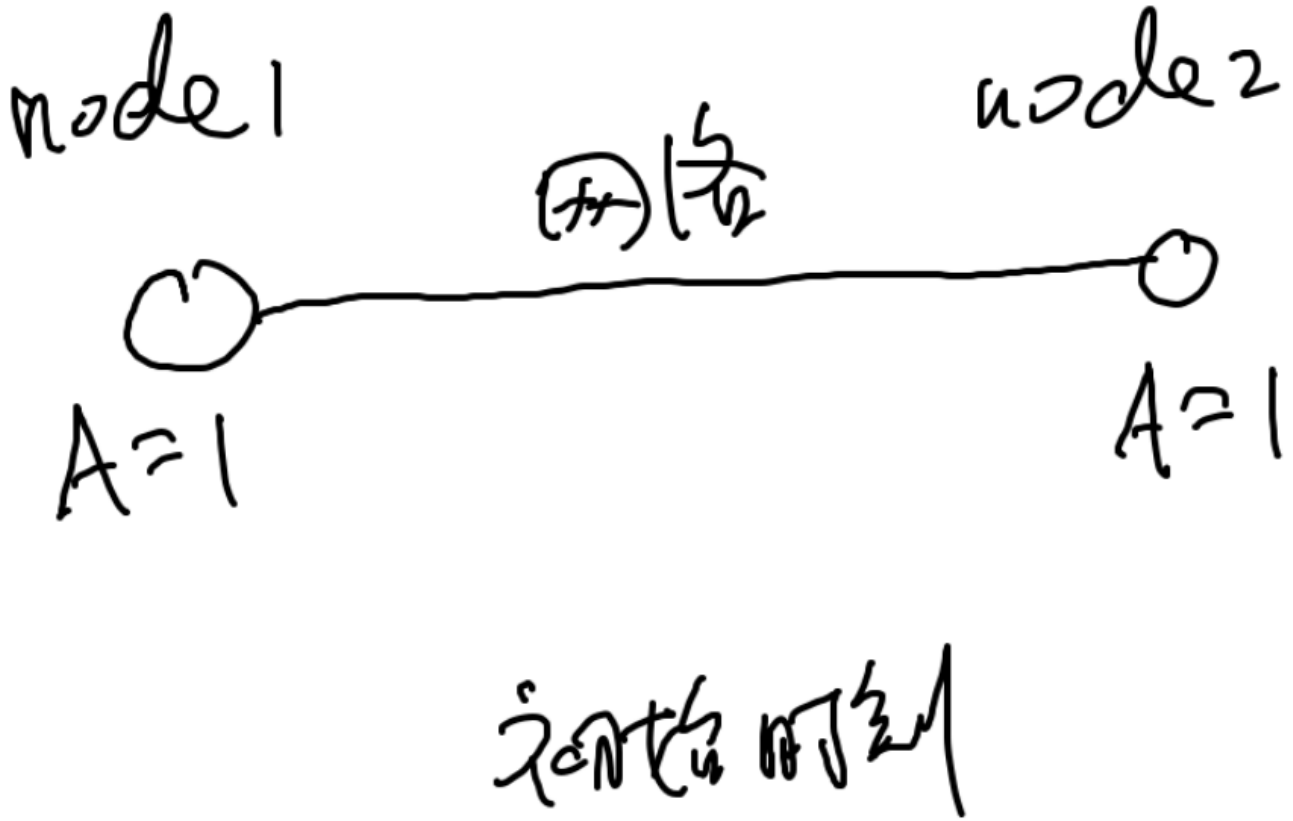
P：Partiton tolerance

- 即**分区容忍性**，这里的分区是指网络意义上的分区，每个“区”通常指延迟很低，高带宽高可靠性的网络（以保证一致性的实现，否则只能通过阻塞保证一致性了），例如（同一机房里的）局域网。由于（跨机房的）网络是不可靠的，所有节点之间很可能出现无法通讯的情况，在节点不能通信时，要保证系统可以继续正常服务。
- 以实际效果而言，分区相当于对通信的时限要求。系统如果不能在时限内达成数据一致性，就意味着发生了分区的情况，必须就当前操作在C和A之间做出选择

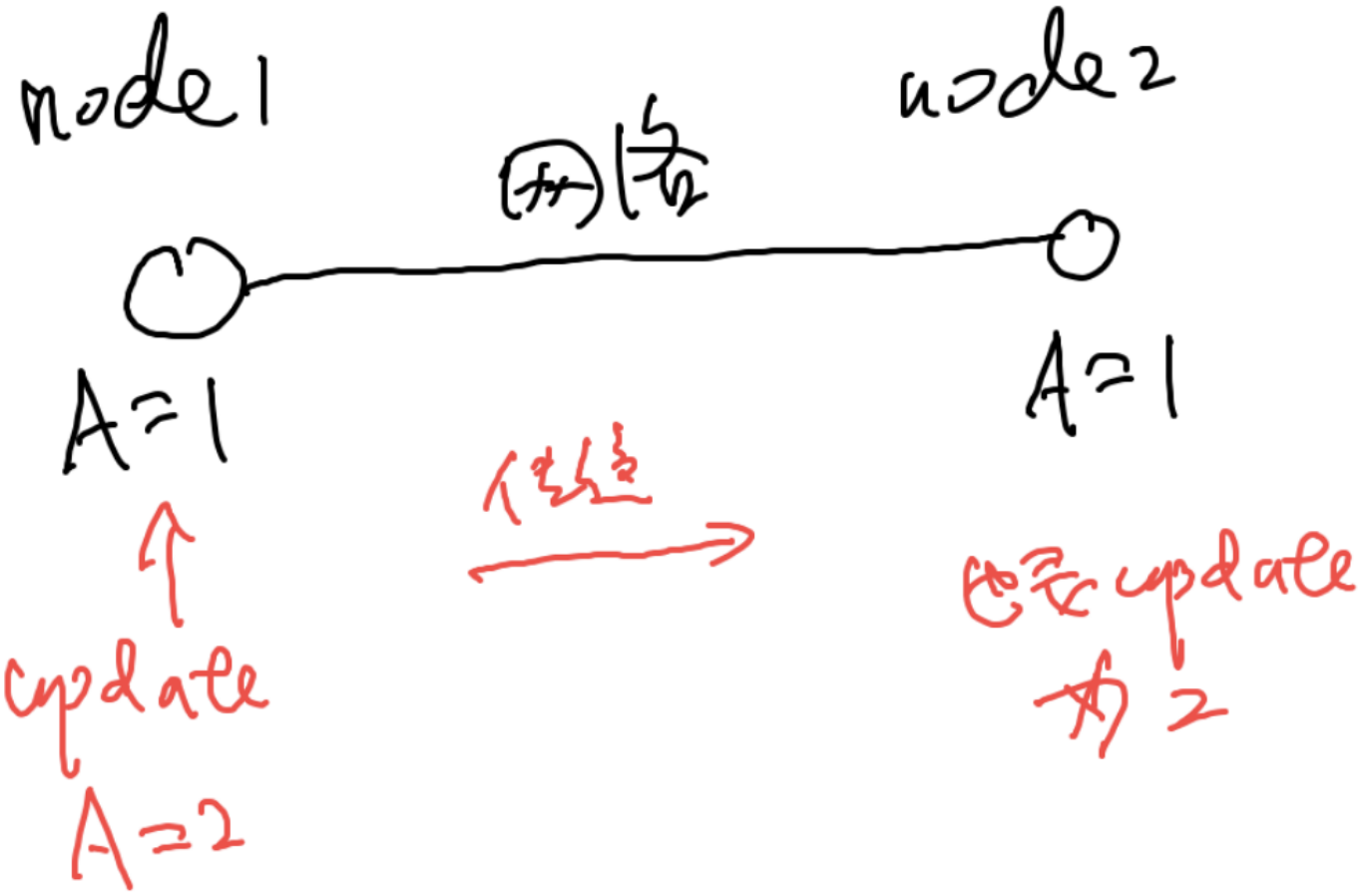
CAP定律

- CAP（Consistency,Availability,Patition tolerance）理论论述的是在任何分布式系统中，只可能满足一致性，可用性及分区容忍性三者中的两者，不可能全部都满足。所以不用花时间精力在如何满足所有三者上面。
- CAP理论无疑是导致技术趋势由关系数据库系统向NoSQL系统转变的最重要原因。
- http://www.julianbrowne.com/article/viewer/brewers-cap-theorem

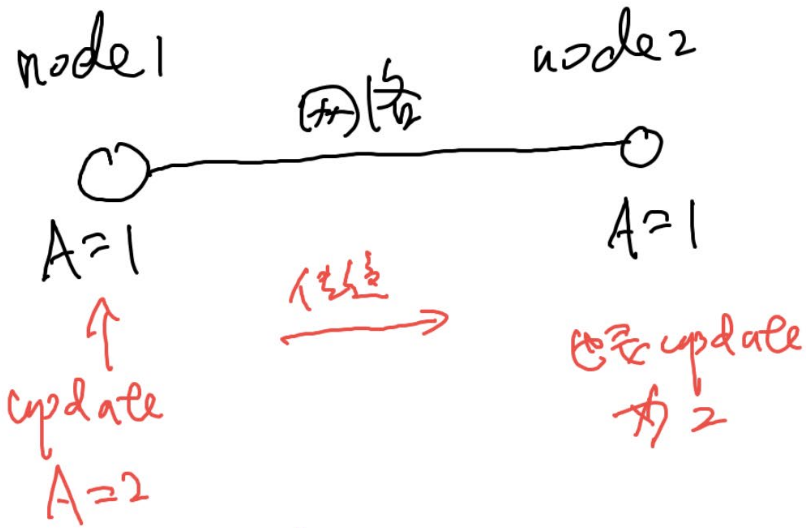
图解C、A、P概念及CAP定律证明思路



初始状况



保证CP，则不满足A



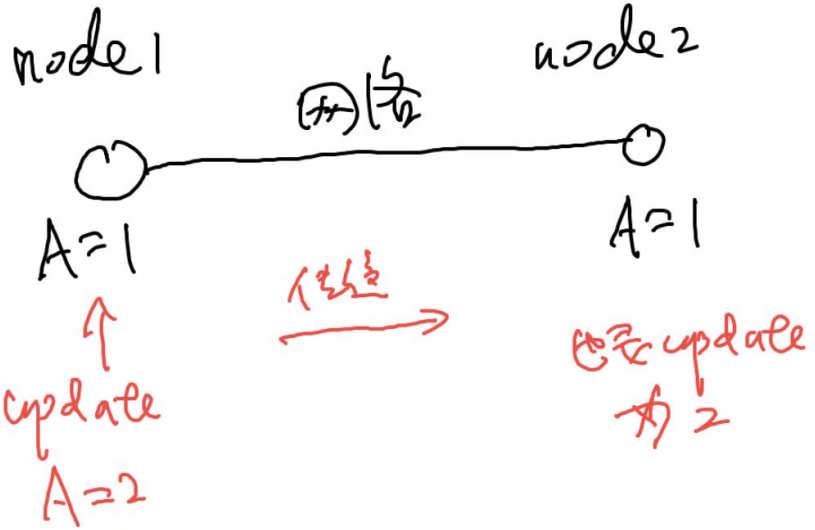
保证C和P的情况

C: 数据要复制给node2

P: 容忍分区错误. 网络可能中断. 为了保障一致性. node2 必须要等到网络恢复.

于是 A 无法保证

保证AP，不满足C



保证A和P.

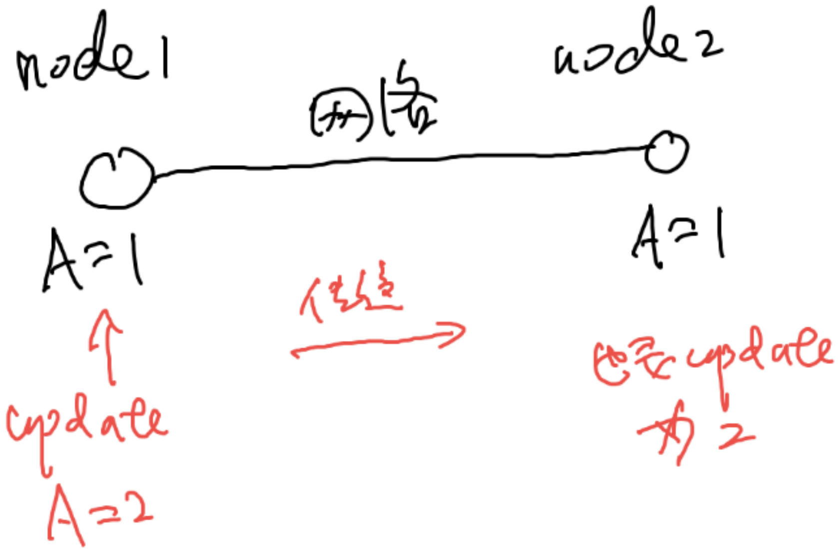
A: node1和node2都必须有响应的时间。否则

P: 网络不可靠. node1的更新未必及时到达node2.

于是 node1 与 node2 可能有不一致的

C无法保证

保证AC，不满足P

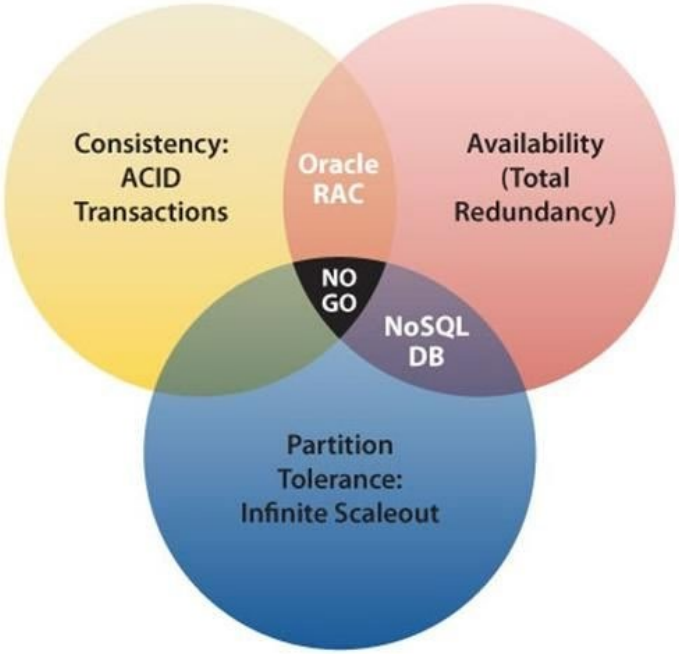


保证A和C
A: node1与node2 都能访问一个
C: node1与node2 数据一致.
这要求 node1与node2 在高度可靠的网络
必须在同一个"区"，于是P不能满足

2024.4.3

NoSQL运动

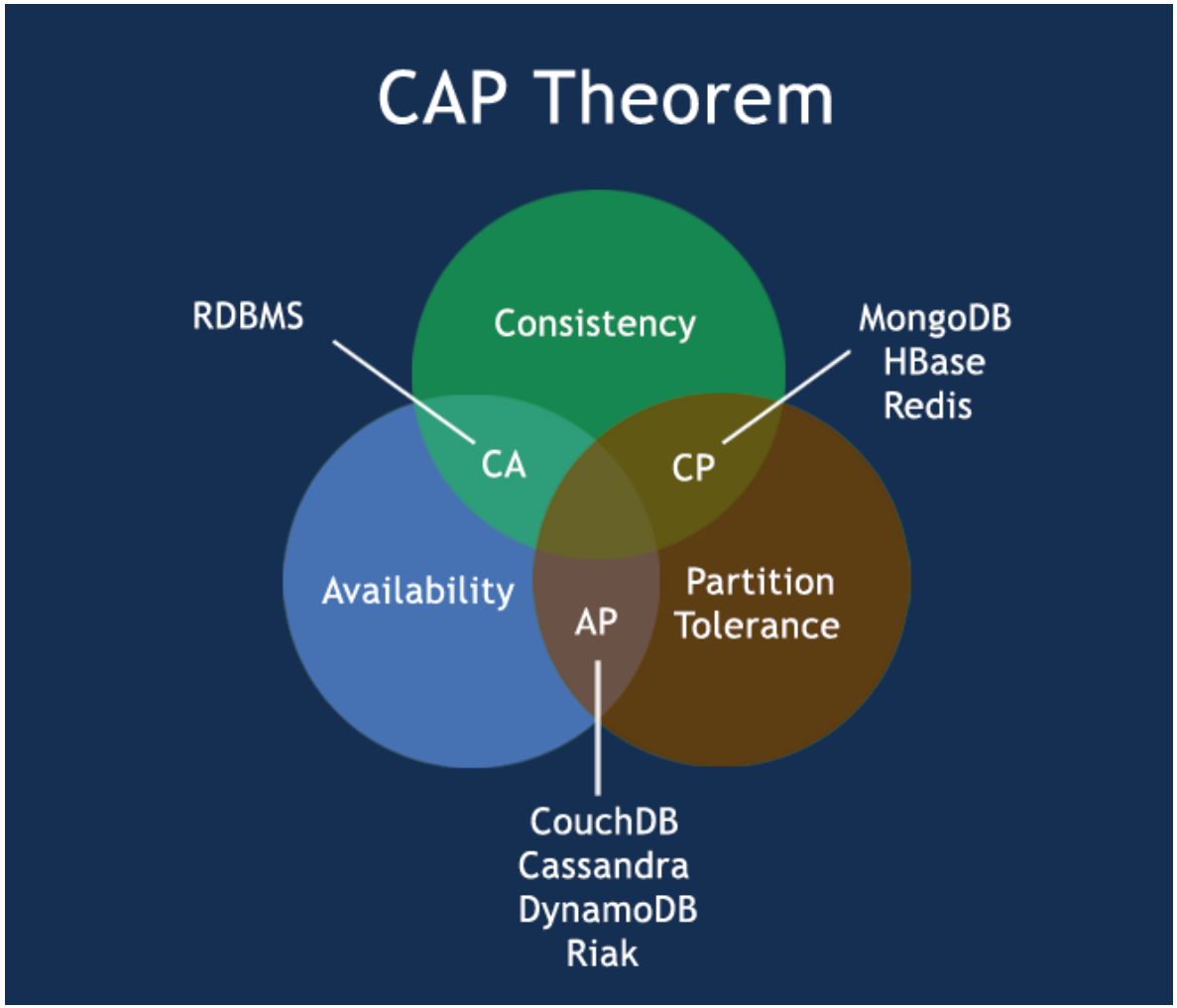
- NoSQL=Not Only SQL
- 大部分为开源软件
- 大部分支持分布式集群
- 在特定的场景下，能解决关系型数据库的某些弱点
- 缺乏统一的解决方案
- K-v数据库是最常见的NoSQL数据库，是BigTable的简化版



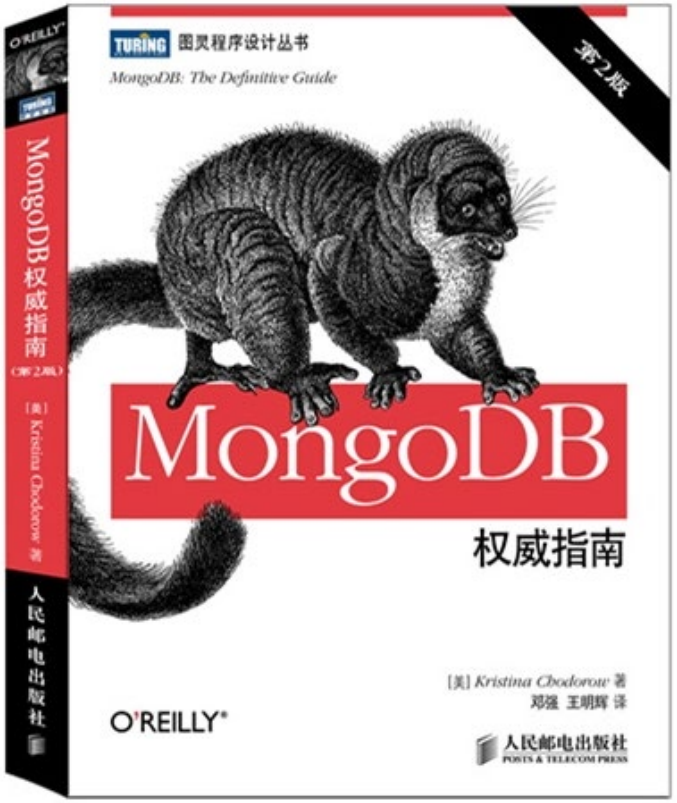
NoSQL数据库家族

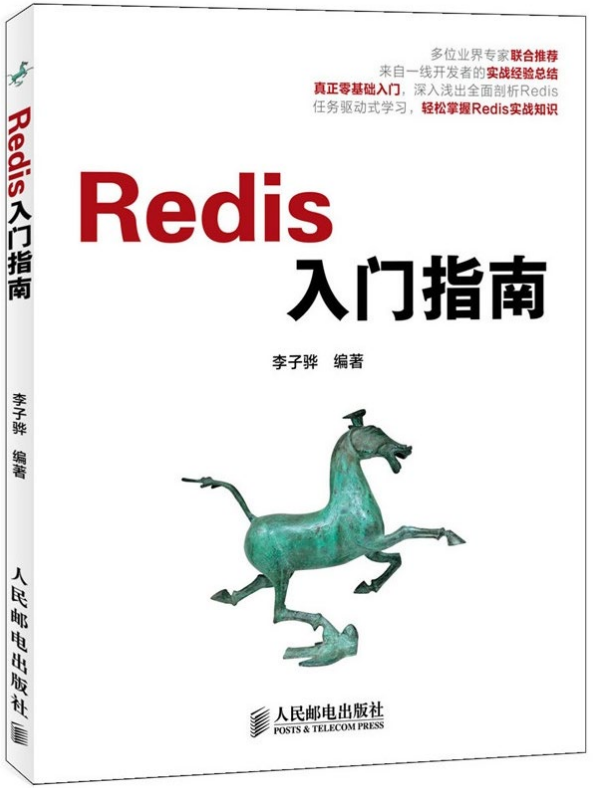
- 多达100多种，还在迅速增加
- 键值（key-value）数据库（例如redis/memcache）
- 面向文档的数据库（例如mongodb）
- 面向列的数据库（例如hbase/hadoop，Cassandra）
- 面向图的数据库（例如neo4j）
- NewSQL与集群型关系数据库（MySQL/MyCAT，PostgreSQL/Greenplum/华为高斯）

各种NoSQL与它们满足的CAP特性

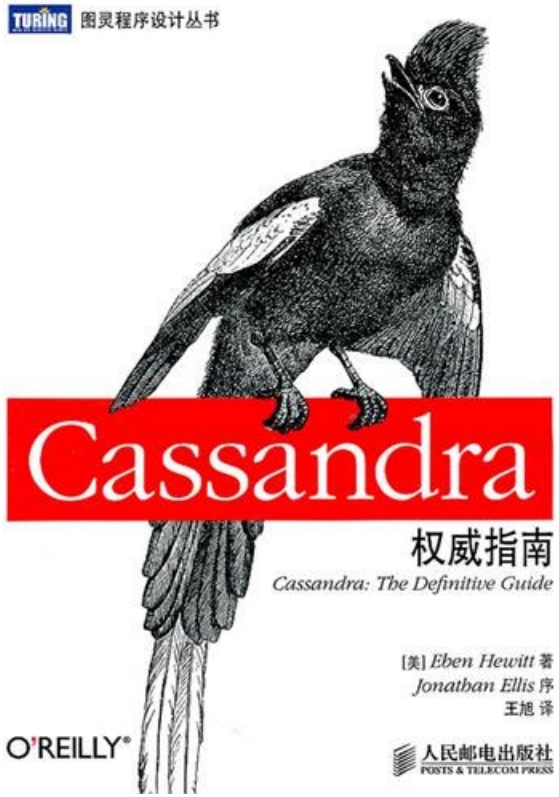


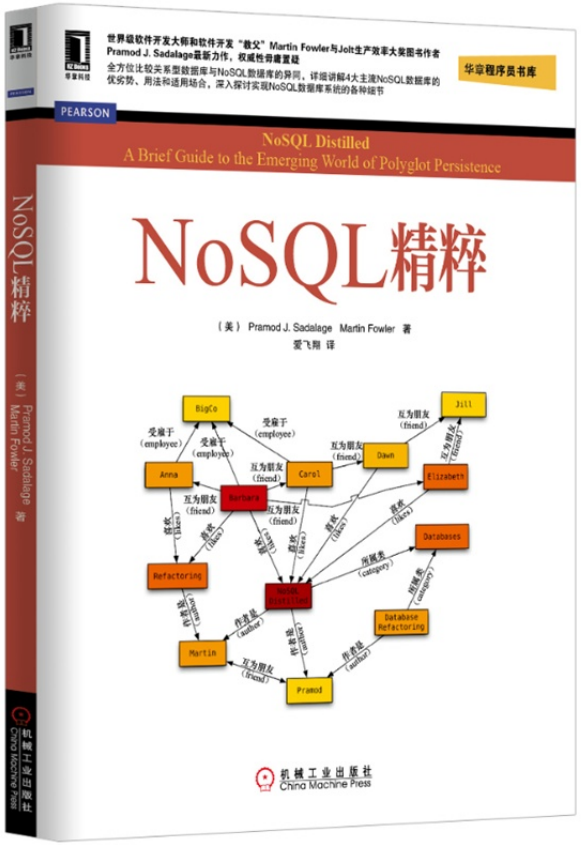
参考书


TURING 图灵程序设计丛书
MongoDB: The Definitive Guide
第2版
MongoDB
权威指南
[美] Kristina Chodorow 著
邓强 王明辉 译
O'REILLY
人民邮电出版社
POSTS & TELECOM PRESS


多位业界专家联合推荐
来自一线开发者的实战经验总结
真正零基础入门，深入浅出全面剖析Redis
任务驱动式学习，轻松掌握Redis实战知识
Redis
入门指南
李子骅 编著
人民邮电出版社
POSTS & TELECOM PRESS

参考书


TURING 图灵程序设计丛书
Cassandra
权威指南
Cassandra: The Definitive Guide
[美] Eben Hewitt 著
Jonathan Ellis 序
王旭 译
O'REILLY
人民邮电出版社
POSTS & TELECOM PRESS


世界级软件开发大师和软件开发"教父"Martin Fowler与Jolt生产效率大奖图书作者Pramod J. Sadalage最新力作，权威性毋庸置疑
全方位比较关系型数据库与NoSQL数据库的异同，详细讲解4大主流NoSQL数据库的优劣势、用法和适用场合，深入探讨实现NoSQL数据库系统的各种细节
华章程序员书库
PEARSON
NoSQL Distilled
A Brief Guide to the Emerging World of Polyglot Persistence
NoSQL精粹
NoSQL精粹
（美）Pramod J. Sadalage Martin Fowler 著
爱飞翔 译
机械工业出版社
China Machine Press

2024.4.3

作业答案：整合三张表

```
SQL> select * from t1;

I           G
- ----------
A          70
B          80
C          75
D          90

SQL> select * from t2;

I           G
- ----------
B          70
D          50
E          60
```

```
SQL> select * from t3;

I           G
- ----------
A          90
B          90
E          55
F          93
```

整合两个表

```
SQL> select * from t1 full outer join t2 on t1.id=t2.id;

I          G I          G
- ---------- - ----------
A          70
B          80 B          70
C          75
D          90 D          50
              E          60
```

```
SQL> select nvl(t1.id,t2.id),t1.g g1,t2.g g2 from t1 full outer join t2 on t1.id=t2.id;

N         G1         G2
- ---------- ----------
A          70
B          80         70
C          75
D          90         50
E                     60
```

作业答案

```
SQL> with
  2  ttt as (select nvl(t1.id,t2.id) C,t1.g g1,t2.g g2 from t1 full outer join t2 on t1.id=t2.id)
  3  select nvl(ttt.C,t3.id) CC,ttt.G1 G1,ttt.G2 G2,t3.G G3 from  ttt full outer join t3 on ttt.C=t3.id
  4  order by CC;

C         G1         G2         G3
- ---------- ---------- ----------
A         70                    90
B         80         70         90
C         75
D         90         50
E                    60         55
F                               93

已选择6行。
```

夹带知识点：with语句

- 用于结构化select语句，使查询逻辑更加清晰易懂，能使用with则尽量使用
- **代码的可读性可维护性使衡量质量的最重要指标之一**
- **为企业确立代码书写规范，同时也为自己确立一个**
- **为什么强调“一条语句完成”？（不产生临时表，发挥SQL优化器作用）**

```
with
ttt as (select nvl(t1.id,t2.id) C,t1.g g1,t2.g g2 from t1 full outer join t2
on t1.id=t2.id)
select nvl(ttt.C,t3.id) CC,ttt.G1 G1,ttt.G2 G2,t3.G G3 from  ttt full
outer join t3 on ttt.C=t3.id
order by CC;
```

作业答案

- 列出平均总收入（工资+提成）最高的部门名称

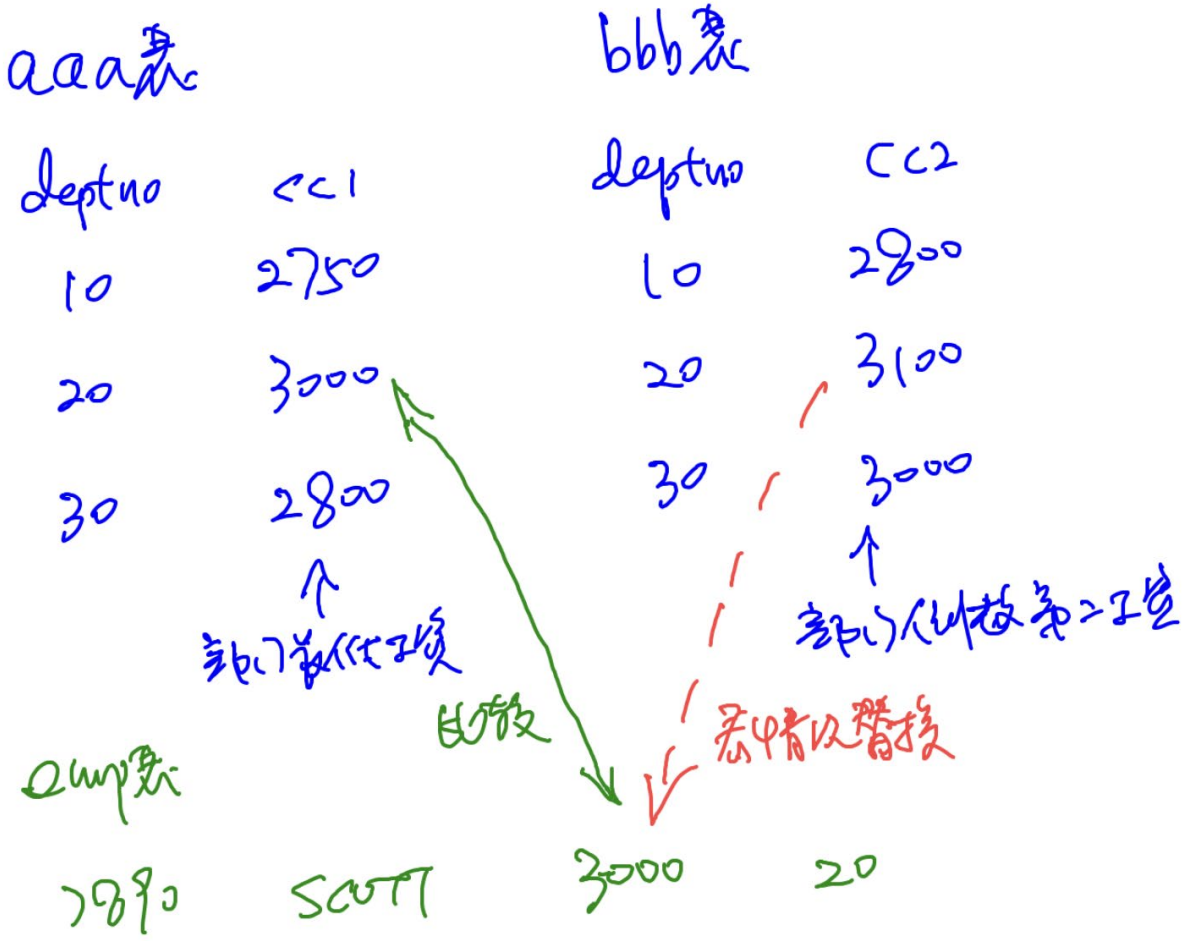
```
SQL> with aaa as --列出部门名称，总收入，并按总收入降序排序
  2  (select dname,avg(sal+nvl(comm,0)) cc from emp,dept where emp.deptno=dept.deptno group by dname order by cc
desc)
  3  select dname from aaa where rownum=1    --找出平均值最高的部门
  4  ;

DNAME
--------------
ACCOUNTING
```

With模块化思维法

```
with
aaa as (select deptno,min(sal) cc1 from emp group by deptno),  /*各部门最低工资*/
bbb as (select deptno,min(sal) cc2 from emp /*各部门倒数第二的工资*/
        where sal<>(select cc1 from aaa where deptno=emp.deptno)
        group by deptno)
update emp set sal=(select cc2 from bbb where deptno=emp.deptno)
where sal=(select cc1 from aaa where deptno=emp.deptno);
```

注意这不是Oracle的合法代码（Oracle只支持select语句前用with定义CTE公共临时表表达式）

思路解释


aaa表
deptno
cc1
10
2750
20
3000
30
2800
bbb表
deptno
cc2
10
2800
20
3100
30
3000
emp表
7890
SCOTT
3000
20

With模块化思维法

- 把上述with语句中的模块名用具体代码替换掉，得到合法代码

```
update emp set sal=(select cc2 from(select deptno,min(sal) cc2 from emp
where sal<>(select cc1 from  (select deptno,min(sal) cc1 from emp group by
deptno)
where deptno=emp.deptno)
group by deptno) where deptno=emp.deptno)
where sal=(select cc3 from (select deptno,min(sal) cc3 from emp group by
deptno) where deptno=emp.deptno);
```

把with部分塞到update语句中

```
update emp set sal=(with
aaa as (select deptno,min(sal) cc1 from emp group by deptno),  /*每个部门最低工资
*/
bbb as (select deptno,min(sal) cc2 from emp /*每个部门倒数第二的工资*/
        where sal<>(select cc1 from aaa where deptno=emp.deptno)
        group by deptno)
select cc2 from bbb where deptno=emp.deptno)
where sal=(with
ddd as (select deptno,min(sal) cc1 from emp group by deptno)  /*每个部门最低工资
*/
select cc1 from ddd where deptno=emp.deptno);
```

运行结果

```
SQL> update emp set sal=(select cc2 from(select deptno,min(sal) cc2 from emp
  2              where sal<>(select cc1 from  (select deptno,min(sal) cc1 from emp group by deptno) where deptno
=emp.deptno)
  3              group by deptno) where deptno=emp.deptno)
  4  where sal=(select cc3 from (select deptno,min(sal) cc3 from emp group by deptno) where deptno=emp.deptno);

已更新4行。
```

```
SQL> update emp set sal=(with
  2  aaa as (select deptno,min(sal) cc1 from emp group by deptno),  /*每个部门最低工资*/
  3  bbb as (select deptno,min(sal) cc2 from emp /*每个部门倒数第二的工资*/
  4               where sal<>(select cc1 from aaa where deptno=emp.deptno)
  5               group by deptno)
  6  select cc2 from bbb where deptno=emp.deptno)
  7  where sal=(with
  8  ddd as (select deptno,min(sal) cc1 from emp group by deptno)  /*每个部门最低工资*/
  9  select cc1 from ddd where deptno=emp.deptno);

已更新3行。
```

注释语句

- 增加代码可读性，凡是能加上注释的地方都习惯性加上注释，不光为了自己以后阅读方便，也可以赢得别人赞赏

```
with aaa as --列出部门名称，总收入，并按总收入降序排序
(select dname,avg(sal+nvl(comm,0)) cc from emp,dept where
emp.deptno=dept.deptno group by dname order by cc desc)
select dname from aaa where rownum=1  --找出平均值最高的部门
;
```

注释语句

```
with aaa as   /*列出部门名称，总收入，并按总收入降序排序*/
(select dname,avg(sal+nvl(comm,0)) cc from emp,dept where
emp.deptno=dept.deptno group by dname order by cc desc)
select dname from aaa where rownum=1     /*找出平均值最高的部
门*/;
```

层次查询

```
SQL> select level,empno,ename,job,mgr from emp
  2  start with empno=7839
  3  connect by prior empno=mgr;

     LEVEL      EMPNO ENAME      JOB              MGR
---------- ---------- ---------- --------- ----------
         1       7839 KING       PRESIDENT
         2       7566 JONES      MANAGER         7839
         3       7902 FORD       ANALYST         7566
         4       7369 SMITH      CLERK           7902
         2       7698 BLAKE      MANAGER         7839
         3       7499 ALLEN      SALESMAN        7698
         3       7521 WARD       SALESMAN        7698
         3       7654 MARTIN     SALESMAN        7698
         3       7844 TURNER     SALESMAN        7698
         3       7900 JAMES      CLERK           7698
         2       7782 CLARK      MANAGER         7839
         3       7934 MILLER     CLERK           7782

已选择12行。
```

层次查询的关键元素

- Level：伪列函数，标注属于树状结构中第几层
- Start with：指定树的起点（根节点）
- Connect by：指定层级上下级关系，prior条件后的次序指出了是自上往下，还是自底往上?

测试：从底往上

```
SQL> select level,empno,ename,job,mgr from emp
  2  start with empno=7839
  3  connect by prior mgr=empno;

     LEVEL      EMPNO ENAME      JOB              MGR
---------- ---------- ---------- --------- ----------
         1       7839 KING       PRESIDENT

SQL> select level,empno,ename,job,mgr from emp
  2  start with empno=7369
  3  connect by prior mgr=empno;

     LEVEL      EMPNO ENAME      JOB              MGR
---------- ---------- ---------- --------- ----------
         1       7369 SMITH      CLERK           7902
         2       7902 FORD       ANALYST         7566
         3       7566 JONES      MANAGER         7839
         4       7839 KING       PRESIDENT
```

测试：换个起点

```
SQL> select level,empno,ename,job,mgr from emp
  2  start with empno=7698
  3  connect by  prior empno=mgr;

     LEVEL      EMPNO ENAME      JOB              MGR
---------- ---------- ---------- --------- ----------
         1       7698 BLAKE      MANAGER         7839
         2       7499 ALLEN      SALESMAN        7698
         2       7521 WARD       SALESMAN        7698
         2       7654 MARTIN     SALESMAN        7698
         2       7844 TURNER     SALESMAN        7698
         2       7900 JAMES      CLERK           7698

已选择6行。
```

测试：更像树状的格式

```
column mychart format A12

select lpad(ename,length(ename)+(level*2)-2,'_') as mychart
from emp
start with ename='KING'
connect by  prior empno=mgr;
```

测试

```
SQL> column mychart format A12
SQL> select lpad(ename,length(ename)+(level*2)-2,'_') as mychart
  2  from emp
  3  start with ename='KING'
  4  connect by  prior empno=mgr;

MYCHART
------------
KING
__JONES
____FORD
______SMITH
__BLAKE
____ALLEN
____WARD
____MARTIN
____TURNER
____JAMES
__CLARK
____MILLER

已选择12行。
```

夹带知识点：SQLplus的column命令

- 很多时候sql语句输出的列宽度不合适（比如非常宽^_^）而影响查看，都需要用到这个命令来更改select语句中指定列的宽度和标题。可简写column为col
- 用法1：修改列宽度

```
column c1 format a20          --将列c1（字符型）显示最大宽度调整为20个字符
column c1 format 9999999  --将列c1（num型）显示最大宽度调整为7个字符
```

用法2：修改列标题

```
column c1 heading c2          --将c1的列名输出为c2
```

删除树的分枝

- 例如删除掉BLAKE带头的分枝，用where会产生怪异结果

```
SQL> column mychart format A12
SQL> select lpad(ename,length(ename)+(level*2)-2,'_') as mychart
  2  from emp where ename<>'BLAKE'
  3  start with ename='KING'
  4  connect by  prior empno=mgr;

MYCHART
------------
KING
__JONES
____FORD
______SMITH
____ALLEN
____WARD
____MARTIN
____TURNER
____JAMES
__CLARK
____MILLER
```

测试：条件加在connect by 子句

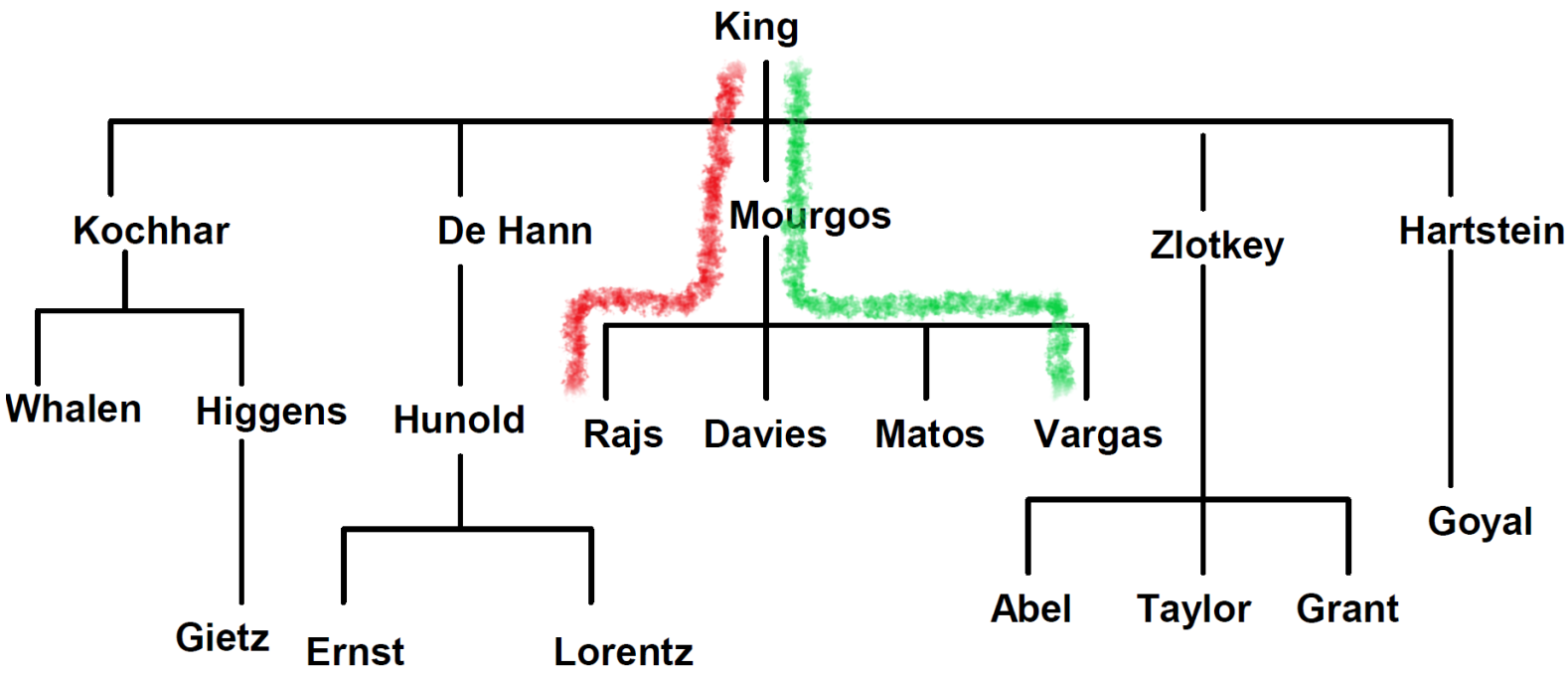
```
SQL> column mychart format A12
SQL> select lpad(ename,length(ename)+(level*2)-2,'_') as mychart
  2  from emp start with ename='KING'
  3  connect by  prior empno=mgr and ename<>'BLAKE';

MYCHART
------------
KING
__JONES
____FORD
______SMITH
__CLARK
____MILLER

已选择6行。
```

问题

- 假设每个人都只能和直接上司或直接下属交流，求任意两人间交流信息需要经过的最小中间节点数。
- 能否用宽表展示结果？

图解与提示

- 假设每个人都只能和直接上司或直接下属交流，求任意两人间交流信息需要经过的最小中间节点数

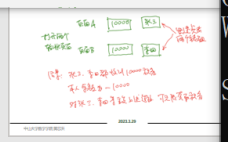


手动加锁：Select...for update语句

```
连接到:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> select * from emp where ename='KING' for update;

     EMPNO ENAM
---------- ----
    DEPTNO
----------
      7839 KING
        10


SQL>
```

SQL Plus

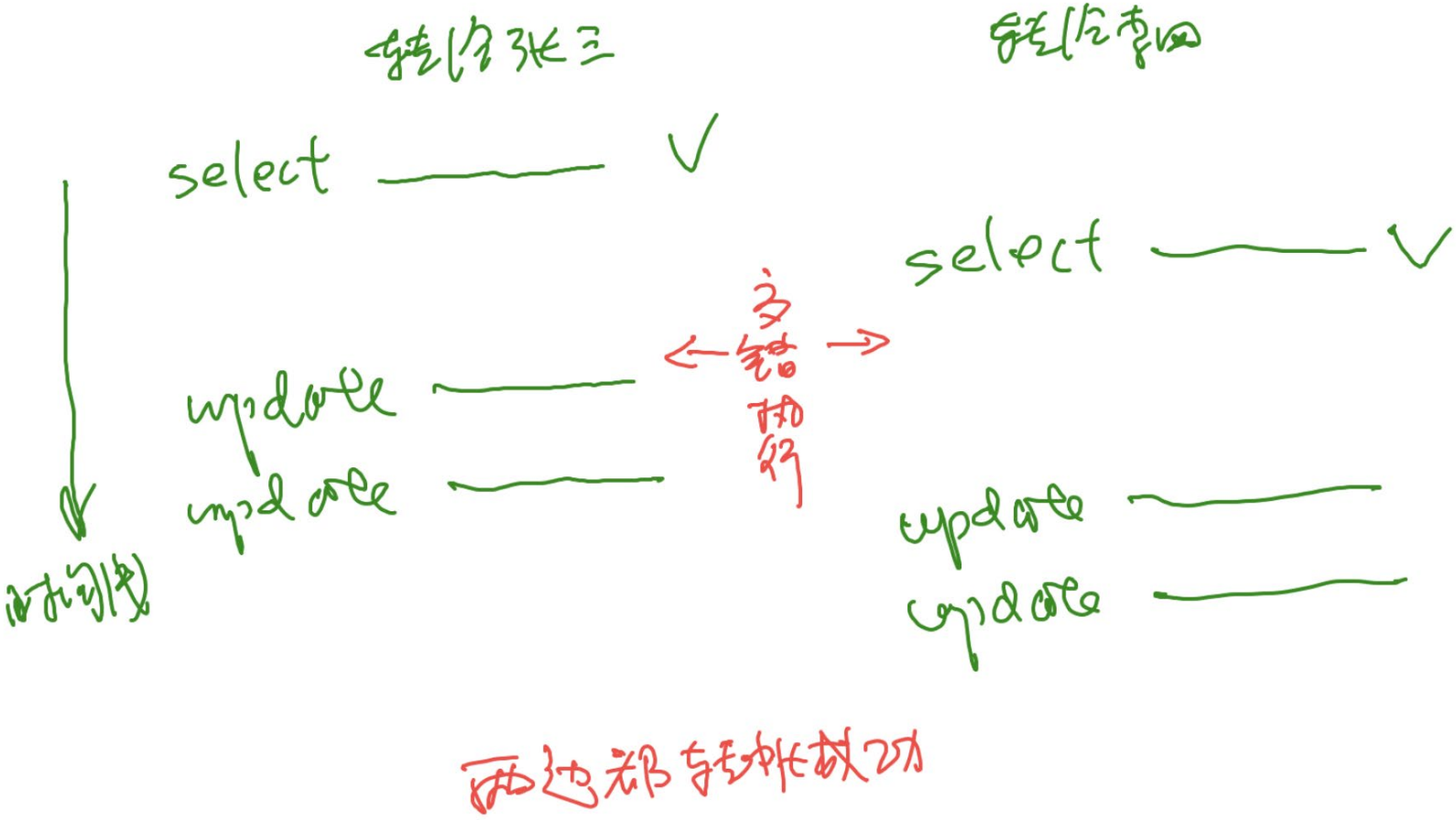
```
SQL*Plus: Release 11.2.0.1.0 Production on 星期三 3月 29 11:57:56 2023

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

请输入用户名:  scott
输入口令:

连接到:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> update emp set sal=5500 where ename='KING';
```



4

5

作业答案


转给张三
转给李四
select
select
update
update
update
update
时间线
互锁
两边都转账成功

作业答案

- 给出第6周课程幻灯片第3页中提到的“ITPUB论坛盗币事件”的解决方案

答：改为单一update语句，用where条件判断账号内是否有足够的钱，如果update了0行，说明没有足够的钱，如果update了1行，说明账户内有足够的钱同时已经扣除，这时可以继续运行下一条update给目标账号加钱

中山大學
SUN YAT-SEN UNIVERSITY

Thanks

FAQ时间