系统架构设计师

DESIGNER:王川林

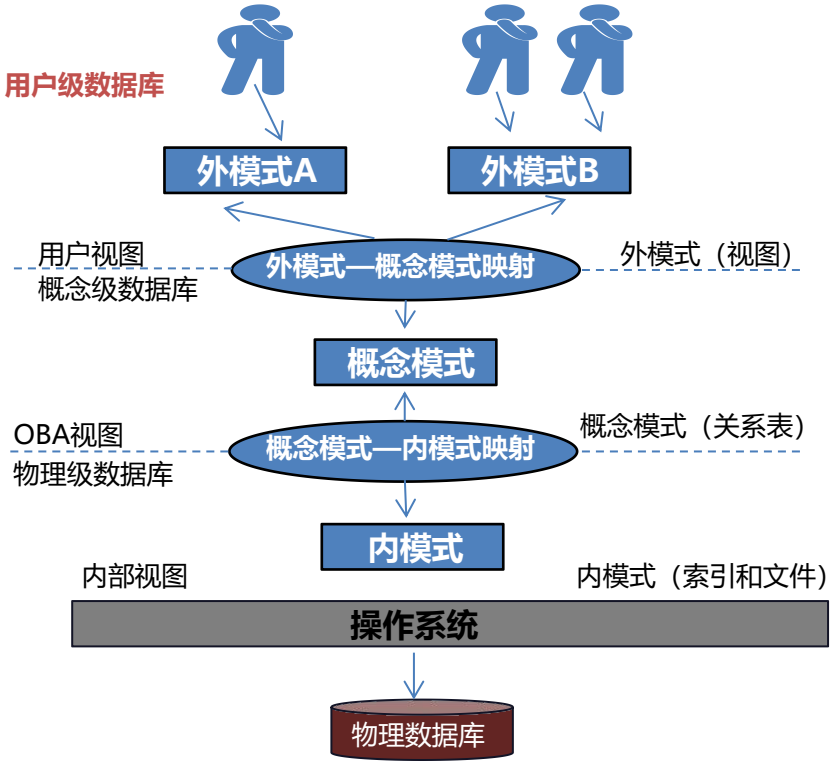
数据库系统

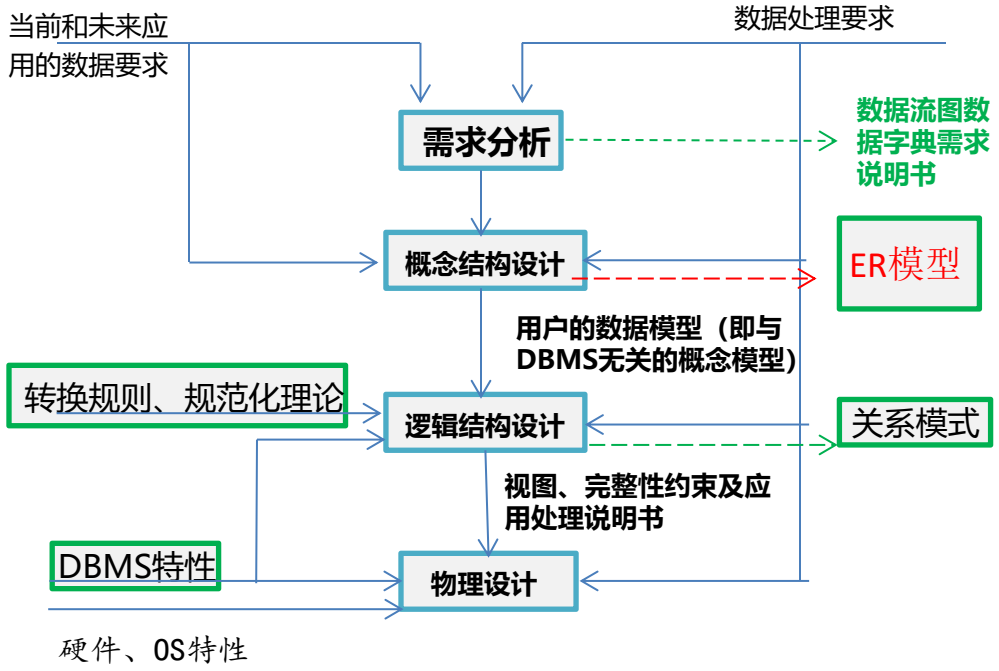
DESIGNER:

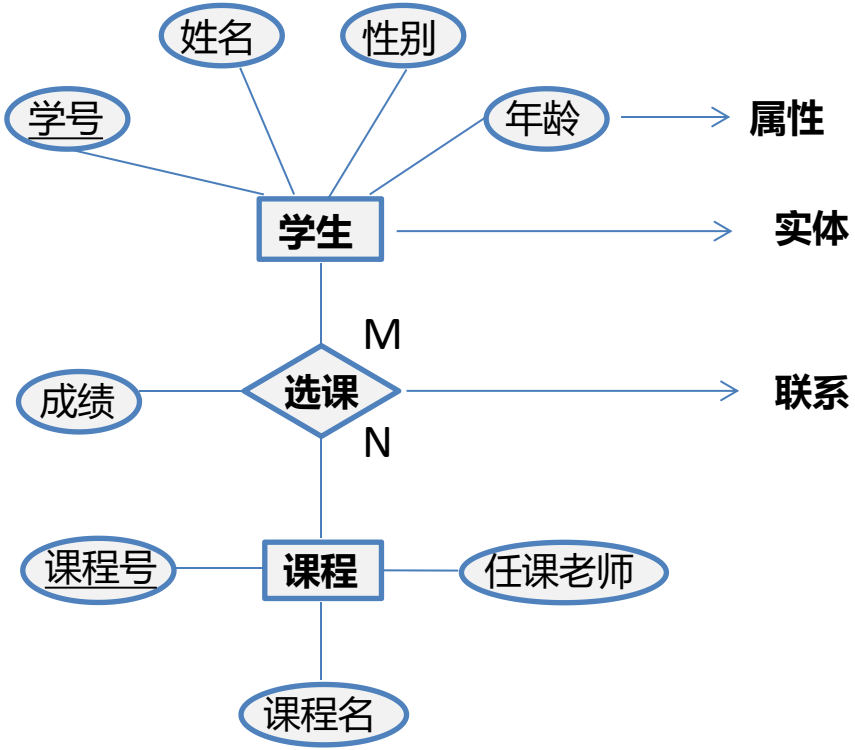
课程内容提要

- 数据库模式 （☆☆☆☆）
- ER模型 （☆）
- 关系代数 （☆☆☆）
- 元组演算 （☆）
- 规范化理论 （☆☆☆☆☆）
- 并发控制 （☆）
- 数据库完整性约束 （☆☆）
- 分布式数据库 （☆☆☆）
- 数据仓库与数据挖掘 （☆☆☆）

三级模式—两层映射


用户级数据库
外模式A
外模式B
用户视图
概念级数据库
外模式—概念模式映射
外模式（视图）
概念模式
OBA视图
物理级数据库
概念模式—内模式映射
概念模式（关系表）
内模式
内部视图
内模式（索引和文件）
操作系统
物理数据库


当前和未来应用的数据要求
数据处理要求
需求分析
数据流图数据字典需求说明书
概念结构设计
ER模型
用户的数据模型（即与DBMS无关的概念模型）
转换规则、规范化理论
逻辑结构设计
关系模式
视图、完整性约束及应用处理说明书
DBMS特性
物理设计
硬件、OS特性


姓名
性别
学号
年龄
属性
学生
实体
M
选课
成绩
联系
N
课程号
课程
任课老师
课程名

集成的方法：

- 多个局部E-R图一次集成。
- 逐步集成，用累加的方式一次集成两个局部E-R。

集成产生的冲突及解决办法：

- 属性冲突：包括属性域冲突和属性取值冲突。
- 命名冲突：包括同名异义和异名同义。
- 结构冲突：包括同一对象在不同应用中具有不同的抽象，以及同一实体在不同局部E-R图
- 中所包含的属性个数和属性排列次序不完全相同。

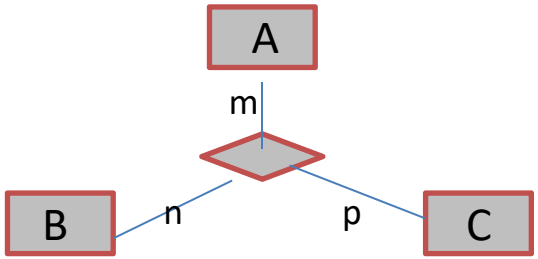
- 一个实体型转换为一个关系模式

  1：1联系　　1：n联系　　m：n联系

- 三个以上实体间的一个多元联系

- 在数据库逻辑结构的设计中，将E-R模型转换为关系模型应遵循相关原则。对于三个不同实体集合它们之间的多对多联系m：n：p，最少可转换为___个关系模式

  A.2　　B.3　　C.4　　D.5



DESIGNER:

关系代数

- 并
- 交
- 差
- 笛卡尔积
- 投影
- 选择
- 联接

关系S1

Sno	Sname	Sdept
No0001	Mary	IS
No0003	Candy	IS
No0004	Jam	IS

关系S2

Sno	Sname	Sdept
No0001	Mary	IS
No0008	Katter	IS
No0021	Tom	IS

s1∩S2（交）

Sno	Sname	Sdept
No0001	Mary	IS

S1 - S2 （差）

Sno	Sname	Sdept
No0003	Candy	IS
No0004	Jam	IS

S1 U S2 （并）

Sno	Sname	Sdept
No0001	Mary	IS
No0003	Candy	IS
No0004	Jam	IS
No0008	Katter	IS
No0021	Tom	IS

DESIGNER:

关系代数

关系S1

Sno	Sname	Sdept
No0001	Mary	IS
No0003	Candy	IS
No0004	Jam	IS

关系S2

Sno	Sname	Sdept
No0001	Mary	IS
No0008	Katter	IS
No0021	Tom	IS

S1 x S2 （笛卡尔积）

Sno	Sname	Sdept	Sno	Sname	Sdepr
No0001	Mary	IS	No0001	Mary	IS
No0001	Mary	IS	No0008	Katter	IS
No0001	Mary	IS	No0021	Tom	IS
No0003	Candy	IS	No0001	Mary	IS
No0003	Candy	IS	No0008	Katter	IS
No0003	Candy	IS	No0021	Tom	IS
No0004	Jam	IS	No0001	Mary	IS
No0004	Jam	IS	No0008	Katter	IS
No0004	Jam	IS	No0021	Tom	IS

投影

Sno	Sname
No0001	Mary
No0003	Candy
No0004	Jam

选择

Sno	Sname	Sdept
No0003	Candy	IS

关系S1		
Sno	Sname	Sdept
No0001	Mary	IS
No0003	Candy	IS
No0004	Jam	IS

关系S2	
Sno	Age
No0001	23
No0008	21
No0021	22

S1 ▷◁S2			
Sno	Sname	Sdept	Age
No0001	Mary	IS	23

规范化理论—价值与用途

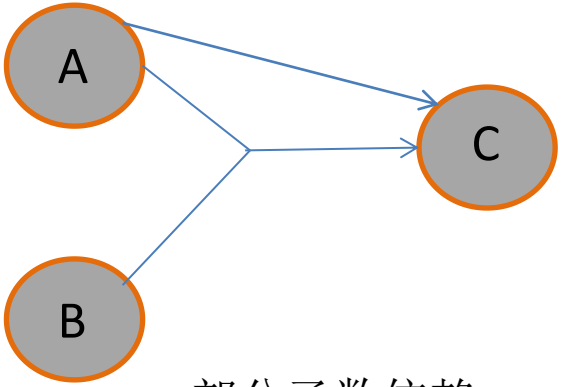
非规范化的关系模式，可能存在的问题包括：数据冗余、更新异常、插入异常、删除异常

SNO	SName	DNO	DNAME	LOCATION
S01	张三	D01	计算机系	1号楼
S02	李四	D01	计算机系	1号楼
S03	王五	D01	计算机系	1号楼
S04	赵六	D02	信息系	2号楼
…	…	…	…	…

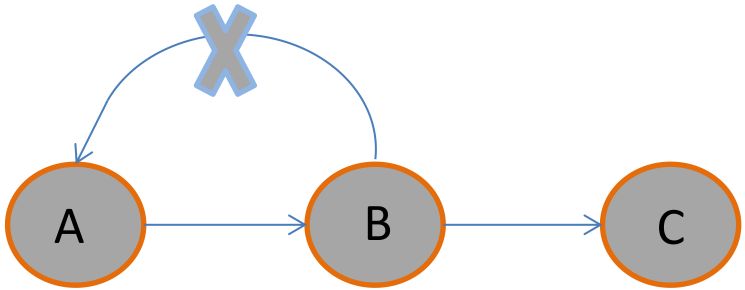
DESIGNER:

规范化理论—函数依赖

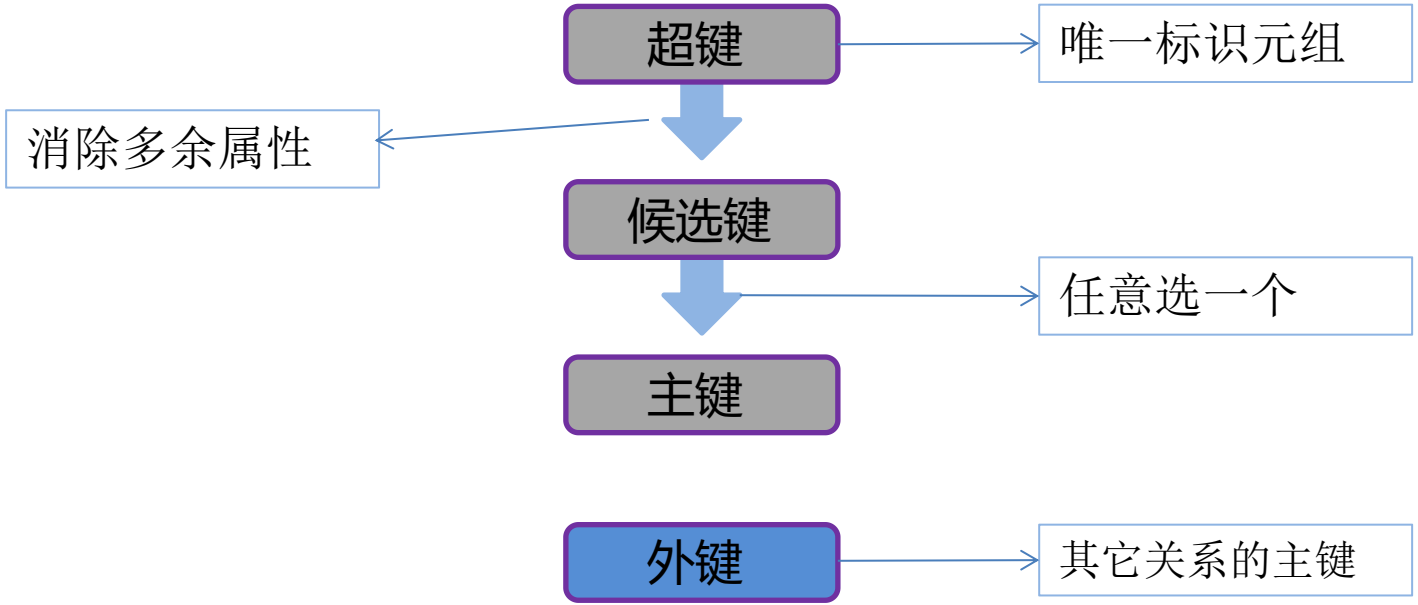
设R（U）是属性U上的一个关系模式，X和Y是U的子集，r为R的任一关系，如果对于r中的任意两个元组u，v，只要有u[Y]=v[Y]，则称X函数决定Y，或称Y函数依赖于X，记为X→Y。



部分函数依赖



传递函数依赖


超键
唯一标识元组
消除多余属性
候选键
任意选一个
主键
外键
其它关系的主键

DESIGNER:

规范化理论—求候选键

- 将关系模式的函数依赖关系用“有向图”的方式表示
- 找入度为0的属性，并以为该属性集合为起点，尝试遍历有向图，若能正常遍历图中所有结点，则该属性集即为关系模式的候选键
- 若入度为0的属性集不能遍历图中所有结点，则需要尝试性的将一些中间点（既有入度，也有出度的结点）并入入度0的属性集中，直至该集合能遍历所有结点，集合为候选键

DESIGNER:

规范化理论—求候选键实例

例1：给定关系R（A1，A2，A3，A4）上的函数依赖集F=｛A1→A2，A3→A2， A2→A3 ，A2→A4｝，R的候选关键字为____。

A.A1　　B.A1A3

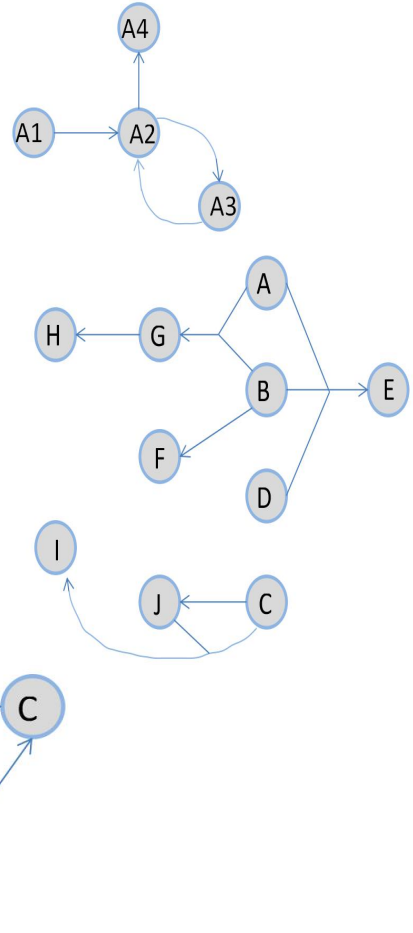
C.A1A3A4　　D.A1A2A3

例2：关系模式P（A，B，C，D，E，F，G，H，I，J）满足下列杉树依赖：FD=｛ABD→E，AB→G，B→F，C→J，CJ→I，G→H｝，求候选码?

例3：关系R（A，B，C）满足下列函数依赖：F｛B→C，B→A，A→BC｝，关系R的候选关键字为___。

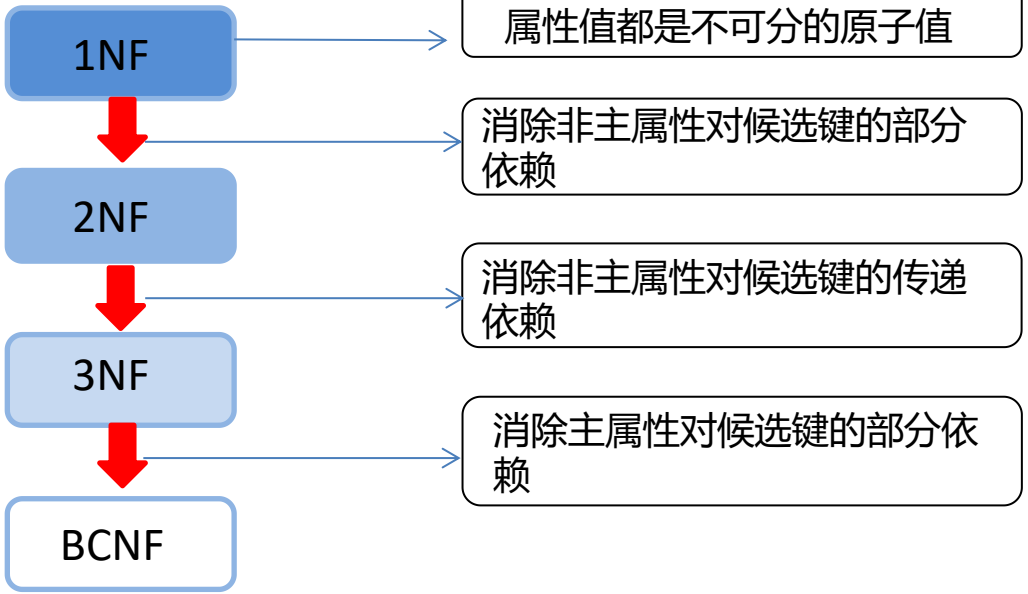
A.AB　　B.A和B

C.A和BC　　D.AC和AB



插入异常、删除异常、数据冗余

逐步优化，以解决



DESIGNER:

规范化理论—第一范式

第一范式（1NF）：在关系模式R中，当且仅当所有域只包含原子值，即每个属性都是不可再分的数据项，则称关系模式R是第一范式。

例如：关系模式R（系名称，高级职称人数）是否满足1NF，如果不满足，应如何调整?

系名称	高级职称人数	
	教授	副教授
计算机系	6	10
电子系	3	5

例：设有职工实体 Employee （职工号，姓名，性别，年龄，通信地址，家庭成员），其中通信地址记录了邮编、省、市、街道信息：家庭成员记录了职工的亲属的姓名。职工实体中的通信地址是一个（1）属性;为了将数据库模式设计的更合理，对于家庭成员属性（2）。

（1）A.简单　　B.复合　　C.多值　　D.派生

（2）A.可以不作任何处理直接记录亲属的姓名
B.只允许记录一个亲属的姓名
C.需要对职工实体设置若干个亲属姓名字段
D.应该将职工的亲属的姓名加上职工号设计成为一个独立的实体

DESIGNER:

规范化理论—第二范式

第二范式（2NF）：当且仅当实体E是第一范式（1NF），且每一个非主属性完全依赖主键（不存在部分依赖）时，则称实体E是第二范式。

思考题：请思考该关系模式会存在哪些问题（从数据冗余、更新异常、插入异常、删除异常这几个方面来考虑），解决方案是什么?

SNO	CNO	GRADE	GREDIT
S01	C01	75	4
S02	C01	92	4
S03	C01	87	4
S04	C01	55	4
S01	C02	87	2
S02	C02	95	2
S01	C03	94	5
…	…	…	…

DESIGNER:

规范化理论—第三范式

第三范式（3NF）：当且仅当实体E是第二范式（2NF），且E中没有非主属性传递依赖于码时，则称实体E是第三范式。

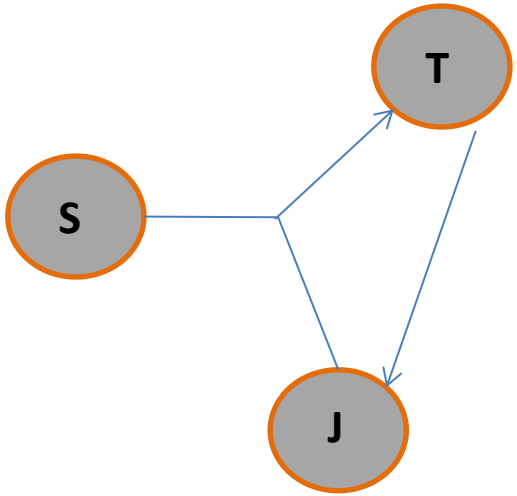
思考题：请思考该关系模式会存在哪些问题（从数据冗余、更新异常、插入异常、删除异常这几个方面来考虑），解决方案是什么?

SNO	SName	DNO	DNAME	LOCATION
S01	张三	D01	计算机系	1号楼
S02	李四	D01	计算机系	1号楼
S03	王五	D01	计算机系	1号楼
S04	赵六	D02	信息系	2号楼
…	…	…	…	…

规范化理论—BC范式

BC范式（BCNF）：设R是一个关系模式，F是它的依赖集，R属于BCNF当且仅当其中F中每个依赖的决定因素必定包含R的某个候选码。

例：关系模式STJ（S，T，J）中，S表示学生，T表示老师，J表示课程。每一老师只教一门课程。每门课程有若干老师，某一学生选定某门课程，就对应一个固定老师。



DESIGNER:

例题讲解

某公司的部门（部门号、部门名、负责人、电话）、商品（商品号、商品名称、单价、库存量）和职工（职工号，姓名，住址）三个实体之间的关系如表1、表2和表3所示。假设每个部门有一位负责人和一部电话，但有若干名员工；每种商品只能由一个部门负责销售。部门关系不属于第三范式的原因是（1）。如果用户要求得到表4所示的结果，需要（2），并增加关系模式（3）。

A.没有消除非主属性对码的部分函数依赖如：部门名→负责人
B.没有消除非主属性对码 部分函数依赖如：负责人→电话
C.只消除了非主属性对码的部分函数依赖而未消除传递函数依赖
D.没有消除非主属性对码的部分函数依赖和传递函数依赖

A.修改表1的结构，在表1中增加一个职工号
B.修改表2的结构，在表2中增加一个职工号
C.修改表2的结构，在表2中增加一个部门号
D.修改表3的结构，在表3中增加一个部门号

A.销售（职工号，商品号，日期，数量）
B.销售（职工号，商品名称，商品号，数量）
C.销售（职工号，部门号，日期，数量）
D.销售（职工号，部门号，商品号，日期）

1

部门号	部门名	负责人	电话
001	家电部	E002	1001
002	百货部	E026	1002
003	视频部	E030	1003

2

商品号	商品名称	单价	库存量
30023	微机	4800	27
30024	打印机	1650	7
...	...	...	...
30101	毛巾	10	106
30102	牙刷	38	288
...	...	...	...

3

职工号	姓名	住址
E001	王军	南京路
E002	李晓斌	淮海路
E021	杨烨	江西路
E026	田波	西藏路
E028	李晓斌	西藏路
E029	刘丽华	淮海路
E030	李彬彬	唐山路
E031	胡慧芬	昆明路
...	...	...

4

职工号	姓名	部门名	月销售额
E001	王军	家电部	528900
E002	李晓斌	家电部	368000
E021	杨烨	百货部	12500
E028	李晓斌	百货部	82500
E031	胡慧芬	食品部	282608
...	...	...	

DESIGNER:

规范化理论—无损分解

保持函数依赖分解

设数据库模式ρ={R1，R2，...，RK}是关系模式R的一个分解，F是R上的函数依赖集，ρ中每个模式Ri上的FD集是Fi。如果{F1，F2，...，Fk}与F是等价的（即相互逻辑蕴涵），那么称分解ρ保持FD。

无损分解

什么是有损，什么又是无损?
有损：不能还原。　　无损：还可以还原。

无损联接分解：指将一个关系模式分解成若干个关系模式后，通过自然联接和投影灯运算仍能还原到原来的关系模式

DESIGNER:

规范化理论—无损分解

思考题：

有关系模式：成绩（学号，姓名，课程号，课程名，分数）
函数依赖：学号→姓名，课程号→课程名，（学号，课程号）→分数
若将其分解为：

成绩（学号，课程号，分数）
学生（学号，姓名）
课程（课程号，课程名）

请思考该分解是否为无损分解?

由于有：学号→姓名，所以：
成绩（学号，课程号，分数，姓名）
由于有：课程号→课程名，所以：
成绩（学号，课程号，分数，姓名，课程名）

DESIGNER:

规范化理论—无损分解

将一个具有函数依赖：学号→姓名，课程号→课程名，（学号，课程号）→分数的关系模式：成绩（学号，姓名，课程号，课程名，分数），分解为：成绩（学号，课程号，分数）；学生（学号，姓名）；课程（课程号，课程名）。

初始表如下：

	学号	姓名	课程号	课程名	分数
成绩	√	×	√	×	√
学生	√	√	×	×	×
课程	×	×	√	√	×

DESIGNER:

规范化理论—无损分解

根据学号→姓名，对上表进行处理，将×改成符号√；然后考虑课程号→课程名，将×改为√，得下表：

	学号	姓名	课程号	课程名	分数
成绩	√	√	√	√	√
学生	√	√	×	×	×
课程	×	×	√	√	×

从上图中可以看出，第1行已全部为√，因此本次R分解是无损联接分解。

DESIGNER:

规范化理论—无损分解

定理：如果R的分解为ρ=｛R_1，R_2｝，F为R所满足的函数依赖集合，分解ρ具有无损联接性的充分必要条件是：

$R_1 \cap R_2 \rightarrow (R_1 - R_2)$

或 $R_1 \cap R_2 \rightarrow (R_2 - R_1)$

其中，$R_1 \cap R_2$表示模式的交，为R_1与R_2中公共属性组成，$R_1 - R_2$表示模式的差集，$R_1 - R_2$表示R_1中去除R_1和R_2的公共属性所组成。当模式R分解成两个关系模式R_1和R_2时，如果R_1与R_2的公共属性能函数决定R_1中或R_2中的其它属性，这样的分解就具有无损联接性。

例：设R=ABC，F=｛A→B｝则分解ρ1=｛R_1（AB），（AC）｝与分解ρ2=｛R_1（AB），R_3（BC）｝是否都为无损分解?

$R_1 \cap R_2 = A$　　$R_1 - R_2 = B$　　$R_2 - R_1 = C$

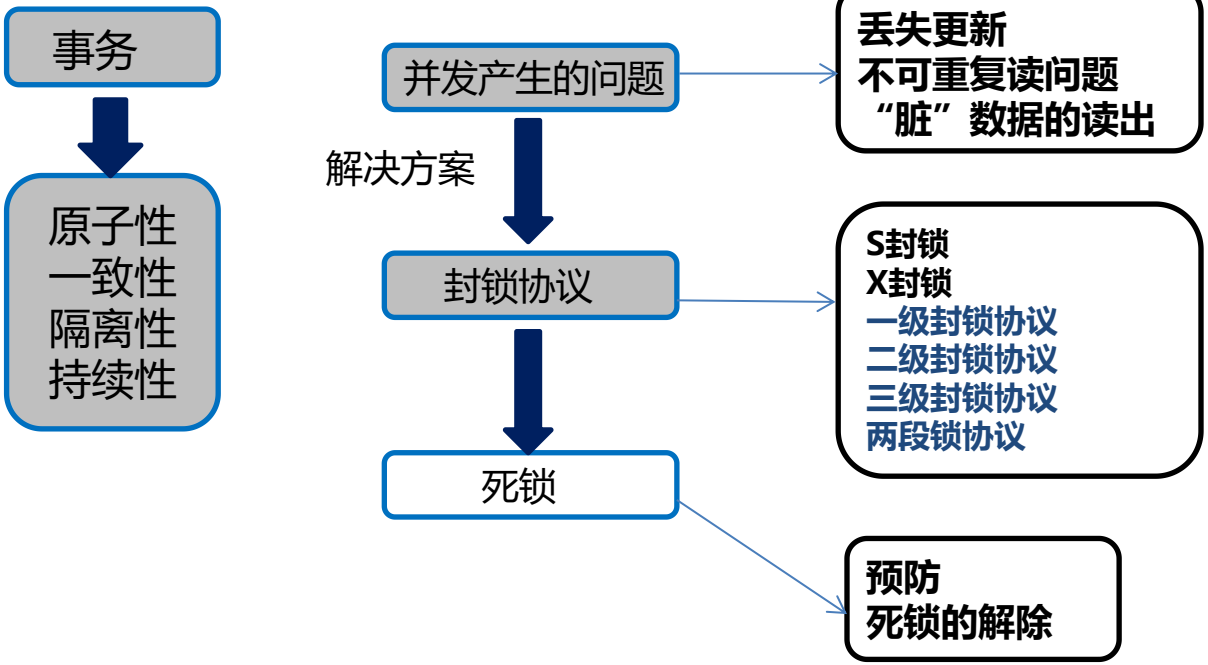
A→B或A→C

$R_1 \cap R_3 = B$　　$R_1 - R_3 = A$　　$R_3 - R_1 = C$

B→A或B→C。

DESIGNER:

并发控制—基本概念



DESIGNER:

并发控制—存在的问题示例

丢失更新

T1	T2
①读A=10 ② ③A=A-5写回④	读A=10 A=A-8写回

不可重复读

T1	T2
①读A=20 读B=30 求和=50 ② ③读A=70 读B=30 求和=100 （验算不对）	读A=20 A←A+50 写A=70

读“脏”数据

T1	T2
①读A=20 A←A+50 写回70 ② ③ROLLBACK A恢复为20	读A=70

DESIGNER:

并发控制—封锁协议

- 一级封锁协议。事务T在修改数据T之前必须先对其加X锁，直到事务结束才释放。可防止丢失修改
- 二级封锁协议。一级封锁协议加上事务T在读取数据R之前先对其加S锁，读完后即可释放S锁。可防止丢失修改，还可防止读“脏”数据
- 三级封锁协议。一级封锁协议加上事务T在读取数据T之前先对其加S锁，直到事务结束才释放。可防止丢失修改、防止读“脏”数据与防止数据重复读
- 两段锁协议。可串行化的。可能发生死锁

DESIGNER:

数据库完整性约束

实体完整性约束
参照完整性约束
用户自定义完整性约束

触发器

措施	说明
用户标识和鉴定	最外层的安全保护措施，可以使用用户账户、口令及随机数检验等方式
存取控制	对用户进行授权，包括操作类型（如查找、插入、删除、修改、等动作）和数据对象（主要是数据范围）的权限。
密码存储和传输	对远程终端信息用密码传输
视图和保护	对视图进行授权
审计	使用一个专用文件或数据库，自动将用户对数据库的所有操作纪律下来

DESIGNER:
数据备份

- 冷备份也称为静态备份，是将数据库正常关闭，在停止状态下，将数据库的文件全部备份（复制）下来
- 热备份也称为动态备份，是利用备份软件，在数据库正常运行的状态下，将数据库中的数据文件备份出来。

优缺点 / 备份方式	优点	缺点
冷备份	非常快速的备份方法（只需要复制文件）：容易归档（简单复制即可）；容易恢复到某个时间点上（只需将文件再复制回去）；能与归档方法相结合，做数据库“最佳状态”的恢复；低度维护，高度安全	单独使用时，只能提供到某一时间点上的恢复；在实施备份的全过程中，数据库必须要作备份而不能做其他工作；若磁盘空间有限只能复制到磁带等其他外部存储设备上，速度会很慢；不能按表或按用户恢复
热备份	可在表空间或数据库文件级备份，备份的时间短；备份时数据库仍可使用；可达到秒级恢复（恢复到某一时间点上）；可对几乎所有数据库实体做恢复；恢复是快速的	不能出错，否则后果严重；若热备份不成功所得结果不可用于时间点的恢复；因难于维护，所以要特别小心，不允许“以失败告终”

DESIGNER:
数据备份

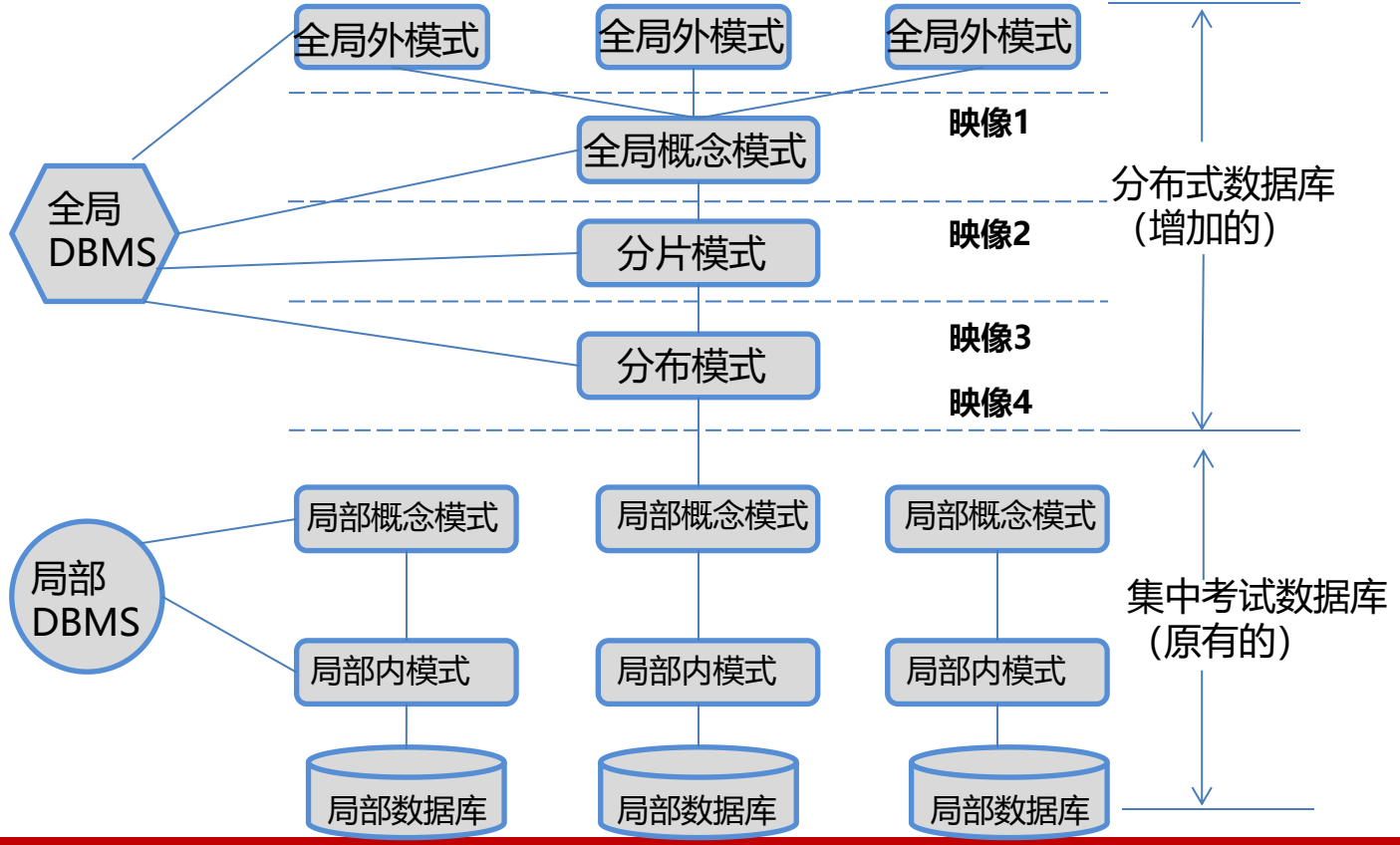
- 完全备份：备份所有数据
- 差量备份：仅备份上一次完全备份之后变化的数据
- 增量备份：备份上一次备份之后变化的数据

日志文件：事务日志是针对数据库改变所做的记录，它可以记录针对数据库的任何操作，并将记录结果保存在独立的文件中

故障关系	故障原因	解决方法
事务本身的可预期故障	本身逻辑	在程序中预先设置Rollback语句
事务本身的不可预期故障	算术溢出、违反存储保护	由DBMS的恢复子系统通过日志，撤销事务对数据库的修改，回退到事务初始状态
系统故障	系统停止运转	通常使用检查点法
介质故障	外存被破坏	一般使用日志重做业务

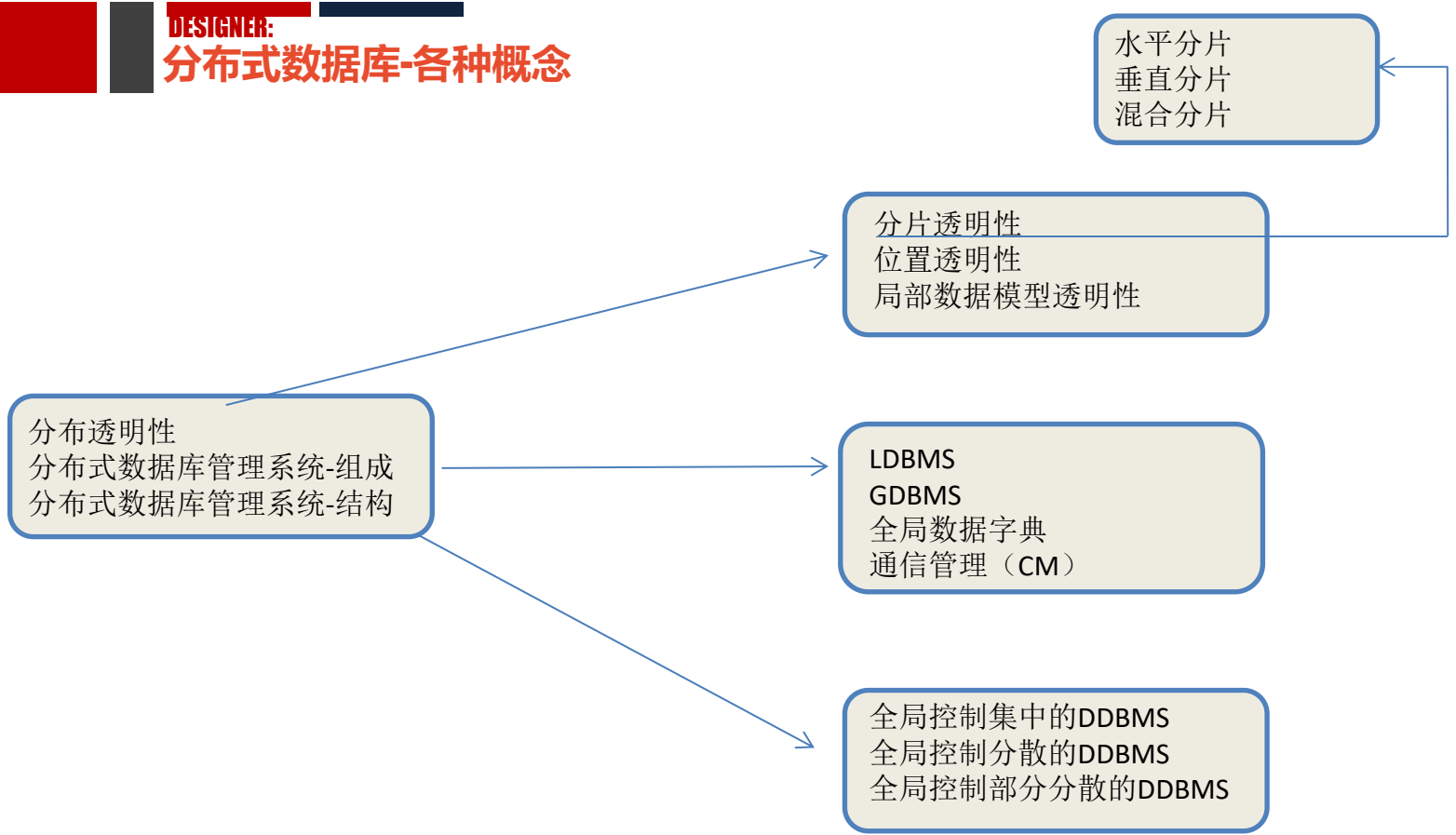
DESIGNER:

分布式数据库-体系结构



DESIGNER:

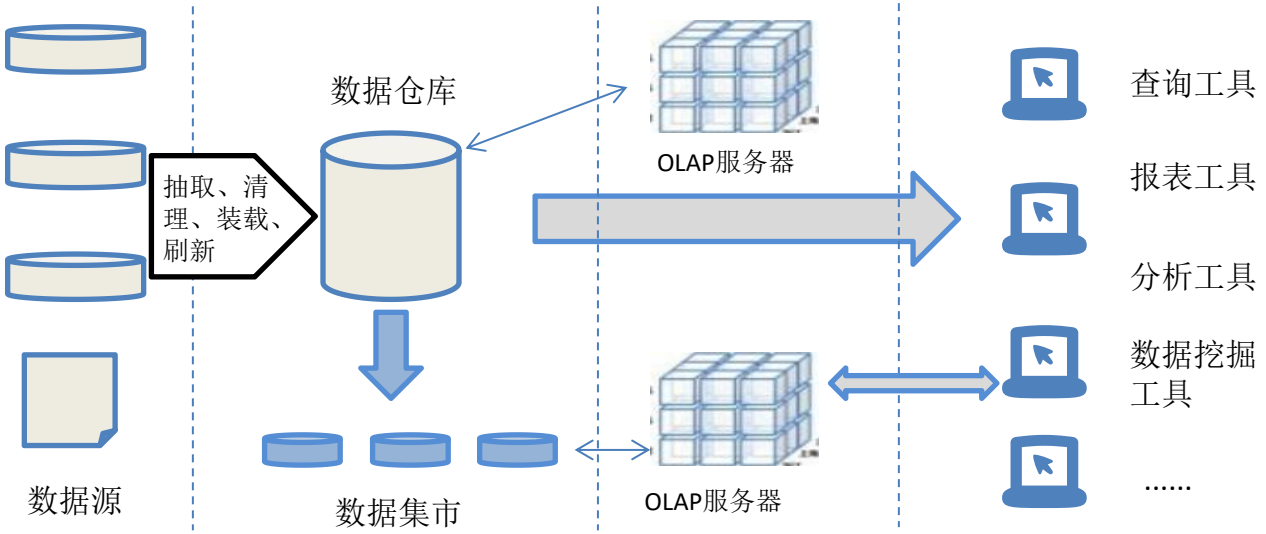
分布式数据库-各种概念



DESIGNER:

数据仓库与数据挖掘

面向主题
集成的
相对稳定的（非易失的）
反映历史变化（随着时间变化）



DESIGNER:

联邦数据库

联邦数据库系统（FDBS）是一个彼此协作却又相互独立的成员数据库（CDBS）的集合，它将成员数据库系统按不同程度进行集成，对该系统整体提供控制和协同操作的软件叫做联邦数据库管理系统（FDBMS）

联邦数据库特征

分布性

异构性

自治性

透明性

联邦数据库分类

紧耦合

松耦合

NoSQL（Not-only SQL）：随着互联网web2.0网站的兴起，传统的关系数据库在应付web2.0网站，特别是超大规模和高并发的SNS类型的web2.0纯动态网站已经显得力不从心了，暴露了很多难以克服的问题，而非关系型的数据库则由于其本身的特点得到了非常迅速的发展

	关系数据库模式	NoSQL模式
并发支持	支持并发、效率低	并发性能高
存储与查询	关系表方式存储、SQL查询	海量数据存储、查询效率高
扩展方式	向上扩展	向外扩展
索引方式	B树、哈希等	键值索引
应用领域	面向通用领域	特定应用领域

- 成熟度不够，大量关键特性有待实现
- 开源数据库产品的支持力度有限
- 数据挖掘与商务智能支持不足，现有的产品无法直接使用NoSQL数据库
- NoSQL数据库专家较少，大部分都处于学习阶段

- SQL+NoSQL = MoreSQL/NewSQL

- Redis、MongoDB、Flare、Cassandra、CouchDB、Oracle NoSQL Database、Tokyo Cabinet

DESIGNER:
反规范化

- **由于规范化会使表不断的拆分，从而导致数据表过多。这样虽然减少了数据冗余，提高了增、删、改的速度 但会增加查询的工作量。系统需要进行多次连接，才能进行查询操作，使得系统的效率大大的下降**

- **技术手段**

  增加派生性冗余列
  增加冗余列
  重新组表
  分割表

DESIGNER:

内存数据

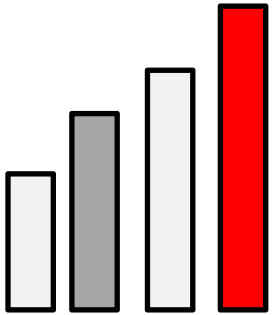
- 内存数据库抛弃了磁盘数据管理的传统方式，基于全部数据都在内存中重新设计了体系结构，并且在数据缓存、快速算法、并行操作方面也进行了相应的改进，所以数据处理速度比传统数据库的数据处理速度要快很多，一般都在10倍以上。内存数据库的最大特点是其“主拷贝”或“工作版本”常驻内存，即活动事务只与实时内存数据库的内存拷贝打交道

- 常见的内存数据库包括：Redis、eXtremeDB、TT、FastDB、SQLite、Microsoft SQL Server Compact 、 MySQL的MEMORY存储引擎等。

DESIGNER:

何谓大数据

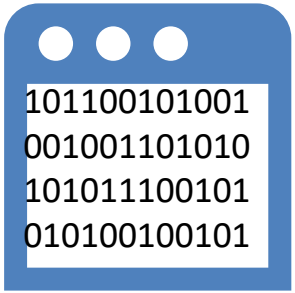
4V?



数据量
Volume

速度
Velocity

多样性
Variety



值
Value

比较维度	传统数据	大数据
数据量	GB或TB级	PB级或以上
数据分析需求	现有数据的分析与检测	深度分析（关联分析、回归分析）
硬件平台	高端服务器	集群平台

★ **大数据处理系统应该具有的重要特征**

高度可扩展性

高性能

支持异构环境

较短的分析延迟

易用且开放的接口

较低成本

向下兼容性