第9章　数据库安全概论

南京大学计算机系　黄皓教授

2009 12月

参考文献

- Charles P. Pfleeger, Shari Lawrence Pfleeger. 李毅超等译。信息安全原理与应用。电子工业出版社，2004年7月第1版。
- 刘启原，刘怡，数据库与信息系统的安全，科学出版社，2000年1月，第1版。

内容

- 安全需求
- 可靠性和完整性
- 敏感数据
- 推理
- 多级安全

1. 安全需求

- **数据库系统的基本安全需求** 访问控制、排除欺骗数据、用户鉴别和可靠性。
- **数据库的物理完整性** 数据库中的数据不受停电之类问题的影响，并且人们可以重建被灾难破坏掉的数据库。
- **数据库的逻辑完整性** 保护数据库的结构。例如，有了数据库的逻辑完整性，修改一个域的值不影响其他的域。
- **元素完整性** 每个元素中包含的数据都是正确的。
- **可审计性** 可跟踪谁访问(或修改)了数据库的元素，或者访问(或修改)了什么元素。
- **用户鉴别** 每一个用户都必须鉴别，包括审计跟踪和允许访问特定的数据。
- **访问控制** 用户只能访问被授权的数据，不同的用户有不同的访问模式(如读或写)。
- **可用性** 用户可以访问数据库中的授权数据和一般数据。

(1) 数据库的完整性

- 保证只有授权个体可以执行对数值的更新
- 能被外在的、非法的程序行为或外力(如失火、停电等)破坏
- 定期地备份数据库系统中的所有文件
- 在故障点重建数据库
  - 例如，突然停电时，银行的客户可能正在进行交易，学生正在为自己在线登记课程。
  - 在这些情况下，我们希望可以将系统恢复到一个稳定点而不必强迫用户重做最近的事务。
  - 为了处理这些情形，DBMS必须维护一个事务日志。例如，假如设计一个银行系统，将每一次处理的事务作为一条记录写入日志文件中(电子日志、纸质日志或两者皆有)。
  - 在系统发生故障后，可通过重新装入数据库的后备副本并重新执行“日志”记录所有的事务，而获得用户账户余额的正确值。

(2)元素完整性

- 元素完整性指数据库元素的正确性或准确性。
- DBMS可以利用域检查，确保某个域的所有值在合适的范围之内。
- 访问控制
- 维护更改日志
  - 更改日志保存了数据库的每次修改；
  - 同时记录初始值和更新值。
  - 使用更改日志，数据库管理员可以取消任何不正确的改变。
  - 例如，一个图书馆的罚单被图书管理员错误地输入到Charles W. Robertson的记录中，而不是Charles M. Robertson，这样就表示Charles W. Robertson不具备成为校运动员的资格。一旦发现这个错误，数据库管理员通过日志恢复Charles W. Robertson原始的值，从而更正数据库。

(3)可审计性

- 审计记录
  - 记录可以协助维护数据库的完整性
  - 能在发生故障后让系统了解发生过什么事、何人参加、有何影响。
  - 可以知道用户不断增加对被保护数据的访问；
  - 为了有效地保持完整性，数据库的审计踪迹应该包括对记录、域和数据元素一级的访问。
  - 只记录所有直接访问的日志可能夸大或低估了用户对数据库的实际了解情形。

(4)访问控制

- 数据库的有用性在于集中存储和维护数据。限制访问是这种集中管理形式的职责和优势。
- 数据库管理员指定了哪个用户可以访问哪些数据，分别以视图、关系、域、记录甚至元素级详细说明。
- 推理(inference)
  - 限制推理意味着要禁止一些特定的路径来阻止可能的推理。
  - 系统企图检查访问请求以防止可能的、不受欢迎的推理攻击，这样做实际上降低了DBMS的性能。

(5)用户鉴别

- DBMS和操作系统之间没有可信途径
- DBMS必须对所有接收到的数据持怀疑态度，包括用户户鉴别。因此，DBMS不得不自己进行用户鉴别。

2. 可靠性和完整性

- 数据库完整性
  - 在磁盘驱动器或数据库主索引损坏后，保障整个数据库不受损害。操作系统的完整性控制和恢复过程可以解决数据库完整性所关心的问题。
- 元素完整性
  - 只有授权用户可以写或修改一个特殊的数据元素。通过适当的访问控制 防止非授权用户篡改和破坏数据。
- 元素正确性
  - 只有正确的值才能被写人数据库。检查元素值可以防止插入不适当的值。

(1) 操作系统提供的保护特性

- 数据库文件会被周期性备份
- 操作系统使用标准访问控制工具保护文件，以免文件在正常的执行过程中被外界访问。
- 最后，在正常读写I/O设备时，操作系统将对所有数据进行完整性检查。

(2) 两阶段更新

- 在修改数据的途中计算系统出现故障是一个严峻的问题。
- 两阶段更新。
  - 第一阶段称为意向(Intent)阶段

    DBMS收集了所有执行更新需要的资源。DBMS可能收集数据、创建哑元记录、打开文件、锁定其他用户的访问、计算最终结果；简而言之，做足更新前的一切准备工作，但并没有对数据库做任何改变。因为每个步骤都并未永久地更新数据，所以第一阶段可以重复无数次。如果在第一阶段系统出故障，不会对数据库有任何损害，因为所有的操作步骤都可以等系统恢复时重做。第一阶段的最后事件是提交(committing)包括为数据库做提交标记(commit flag)。提交标记意味着：DBMS通过了不需要撤销的点：提交之后，DBMS将开始执行永久的更新。
  - 第二阶段实现永久更新

    在第二阶段，不能重复执行提交前的任何动作，但是在这个阶段可以根据需要重复执行更新操作。如果在这个阶段系统出现故障，数据库可能包含不完整的数据，但系统可以重新执行第二阶段的所有动作以修复数据。第二阶段结束，数据库更新完毕。

两阶段更新的实例

- 假定数据库中包含某个公司的办公用品清单。公司的中心仓库里储存了纸、笔、文件夹等，以及不同部门申请不同办公用品的请求清单。由于公司有一套内部针对各部门使用办公用品的收费机制，所以，各部门都有办公用品的预算。同时，中心仓库管理员需要监控现有的供应量，以便在现存的供应品不足时定货。

- 假定会计部最先申请，需要50箱文件夹。现有107箱库存，当库存总量低于100时，需要定货。这里列出仓库接收请求时的步骤：

  1. *仓库核对数据库是否有50箱文件夹库存。如果没有，拒绝请求，事务结束。*
  2. *如果库存充足，从数据库中的存货总量中扣除50(107－50=57)。*
  3. *仓库向会计部的办公用品预算(也是在数据库中)收取50箱文件夹的费用。*
  4. *仓库检查剩余的库存(57箱)，查看是否剩余量低于定货界限。如果是，将产生一个定　货提示，并将数据库中的文件夹项标记为“定货中”。*
  5. *准备好交货，把50箱文件夹送到会计部。*

两阶段更新的实例

- 意向：
- 检查数据库中COMMIT-FLAG值。如果该值已经设置，则意向阶段不再执行。所以意向阶段可能停止，或者循环检查COMMIT-FLAG值直到它被清零为止。
  1. 比较库存的文件夹箱数和需求量；如果需求量大于库存，停止。
  2. 计算TCLIPS=ONHAND-REQUISITION。
  3. 获得BUDGET，目前应获得会计部的办公用品预算。计算TBUDGET＝BUDGET－COST． COST是指这50箱文件夹的成本。
  4. 检查TCLIPS是否低于定货界限；如果是，设置TREORDER=TRUE；否则，设置TREORDER=FALSE。
- 提交：
  1. 在数据库中设置COMMIT-FLAG。
  2. 复制TCLIPS到数据库中的CLIPS。
  3. 复制TBUDGE’到数据库中的BUDGET。
  4. 复制TREORDER到数据库中的REORDER。
  5. 准备向会计部发送文件夹的通知。在日志中标明已完成的事务。
  6. 清除COMMIT－FLAG。

(3) 冗余／内在一致性

- 检错与纠错码
  - 一种冗余形式是检错码和纠错码，比如奇偶校验位、汉明(Haming)编码、循环冗余检查。这些编码可应用于单个域、记录或整个数据库。每次在数据库中添加数据项时，将计算并存储适当的检验码；每次检索数据项时，同样将计算检验码并与存储的值相比较。如果两个值不相等，表明DBMS检测到数据库中出现了错误。某些检验码能指出错误的具体位置；另一些则还能指出正确的值应该是什么。检验码提供的信息越多，所占的存储空间越大。
- 影子域
  - 在数据库中可以复制整个属性或记录。如果一个数据是不能再生的，一旦发现了错误，可以立即用它的拷贝来替换。冗余域显然需要大量的存储空间。

(4) 恢复

- 除了这些纠错过程外，DBMS还需要维护用户的访问日志尤其是更改日志。在发生故障后，从后备副本中重新装载数据库，并根据审计日志将数据库恢复到故障前最后一个正确状态。

(5) 并发性 / 一致性

- DBMS把整个查询-更新周期看做一个原子操作。

(6) 监视器

- 监视器是DBMS中负责数据库结构完整性的单元。监视器检查输入值以确保输入值与数据库中的其他部分或特定域的特性保持一致。
- 范围比较
  - 范围比较监视器检测每个新产生的值，确保每个值在可接受的范围内。
- 状态约束
  - 状态约束(state constraint)描述了整个数据库的约束条件。数据库的值决不能违反这些约束。也就是说，如果不满足这些约束，数据库中的一些数值必然会出错。
- 转换约束
  - 状态约束描述了数据库正确的状态。转换约束(transition constraint)则描述了改变数据库之前的必需条件。

3. 敏感数据

- 某些数据库保存了所谓的敏感数据。其工作性质决定了敏感数据(sensitive data)是指不能公开的数据。决定哪个数据项或域为敏感数据依赖于数据库和数据的含义。
- 一些数据库，如公共图书目录库，没有敏感数据；另一些数据库，如与国防相关的数据库，全部是敏感数据。
- 这是最容易处理的两个实例：或者根本没有敏感数据，或者全部都是敏感数据，因为可以对整个数据库进行访问控制。一个人要么是整个数据库的授权用户，要么不能访问数据库，这样的控制可以由操作系统提供。

（1）敏感数据

- 在实际情况中，数据库中的元素常常是部分而不是全部为敏感数据，而且敏感的程度各有不同。

- 例如，大学数据库中可能包含学生的姓名、资助、寝室号、吸毒、性别、停车罚金、种族。
- 这些数据中姓名和寝室号的敏感程度最低；而资助、停车罚金、是否吸毒具有最高敏感性；性别和种族介于两者之间。也就是说，许多人可能有访问姓名的权限，少部分人还可以访问性别、种族，更少部分人可以访问资助、停车罚金或吸毒。

（2）一个数据库的样本数据

Name	Sex	Race	Aid	Fines	Drugs	Dorm
Adams	M	C	5000	45	1	Holmes
Bailey	M	B	0	0	0	Grey
Chin	F	A	3000	20	0	West
Dwitt	M	B	1000	35	3	Grey
Earhart	F	C	2000	95	1	Holmes
Fein	F	C	1000	15	0	West
Groff	M	C	4000	0	3	West
Hill	F	B	5000	10	2	Holmes
Koch	F	C	0	0	1	West
Lin	F	A	0	10	2	Grey
Majors	M	C	2000	0	2	Grey

（3）使数据敏感的几个因素

- 固有的敏感性
  - 因数据值本身具有揭露性而为敏感数据。例如防御导弹的位置，只有一个理发师的城镇中理发师的平均收入。
- 来自敏感源
  - 数据的来源预示了对机密性的要求。例如信息来自一个告密者，如果信息被揭露，将会危及这个告密者的安全。
- 声明的敏感性
  - 数据库管理员或数据所有者声明该数据为敏感数据。例如机密的军事数据或一件艺术品的匿名捐赠人。
- 敏感属性或敏感记录中的部分
  - 在数据库中，整个属性或记录可以归类为敏感的。例如 人事部数据库中的薪水属性或描述一个秘密太空任务的记录。
- 与已经泄露信息相关的敏感性
  - 有些数据在某些数据面前变得敏感。例如，只知道秘密金矿的经度没有什么用处，但是如果既知道经度又知道纬度，就可以查明金矿的位置。

（4）访问决策

- 数据的可用性
  - 一个或多个被请求的元素可能不可访问。例如，如果用户正在更新几个数据域，其他用户对这几个域的访问便被暂时阻塞。
  - 这种阻塞保证了用户不接收错误的信息，比如街道地址已更新而城市和州还未更新，或旧文件中的新代码组件。阻塞通常是暂时的。当执行更新时，一个用户可能不得不阻塞其他用户对几个域或几个记录的访问，以确保其他用户获得的数据具有一致性。
  - 可能导致的拒绝服务攻击

（5）可接受的访问

- 确定什么样的数据是敏感数据
  - **敏感数据** 一个用户请求访问FINES不为0的所有学生的NAME和DORM。这个FINES域为敏感域，不能泄露其准确值，尽管“不为0”只是部分的揭露了其性质。尽管不是明确的给出敏感数据，数据库管理器仍然拒绝了这种访问，理由是不能向未授权用户泄露信息。
  - **敏感数据的平均值** 用户也可以通过不敏感的统计数据推理敏感数据；例如，如果资助的平均值并未泄露个人资助值，数据库管理系统可以认为这个平均值是安全的，可以访问。但是，当仅有一个数据时，平均值无疑是泄露了敏感数据。

（6）保证用户的真实性

- 当允许访问时，可能要考虑数据库用户的某些外在特性。
  - 为了加强安全，数据库管理员可能只允许一个人仅在特定时间段访问数据库，比如上班时间。
  - 可能还会考虑前一个用户的请求；重复请求某个相同的数据或者发出一个穷尽某特定目录所有信息的请求。
  - 有时可以通过组合几个敏感程度低的查询结果揭示一些敏感数据。

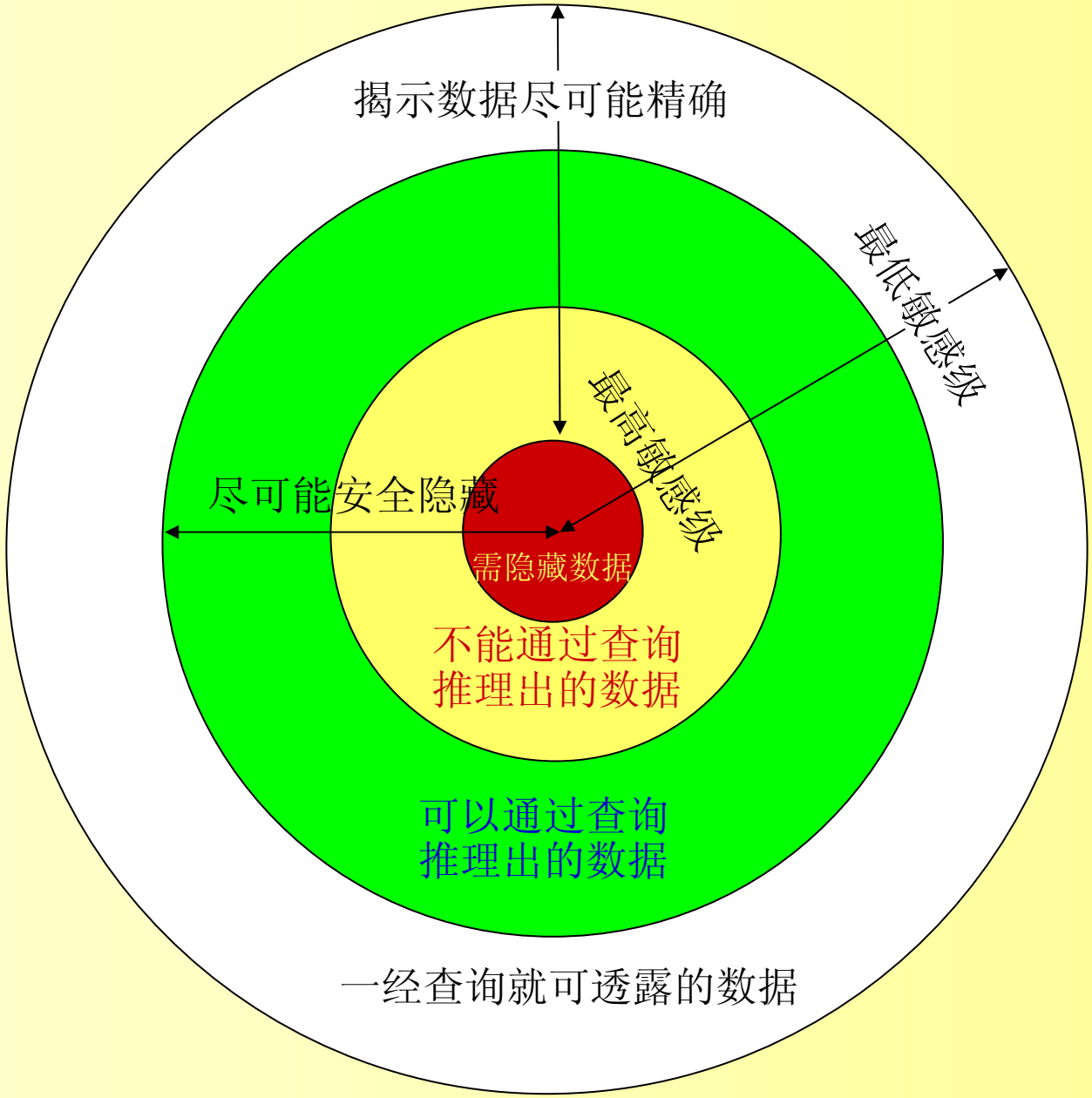
(7) 泄露类型

- 准确数据
  - 最严重的暴露就是泄露了敏感数据的准确值。不够完善的数据库管理器甚至可能意外地送出一些敏感数据。
- 范围
  - 另外一种暴露就是揭露了敏感数据值的范围，比如，指出敏感数据y的值间于值L和值H之间。有时，用户可以通过类似折半查找的逼近技术，先确定是否$L \leqslant y \leqslant H$，然后确定是否$L \leqslant y \leqslant H/2$，如此推理下去，用户可以得到关于y的较精确的值。
  - 另一种情形是，只揭露一个值超过某个总量，比如运动奖金预算或中央情报局情报员的数量，这同样严重违反了数据的安全性。
- 否定结果
  - 有时我们会执行查询确定一个否定的结果。也就是说，我们可以了解z不是y的值。

- 存在性
  - 有时，与数据的具体值无关，数据的存在性本身就是敏感数据。例如，老板不想雇员知道使用长途电话是被监控的。在这种情况下，在职员文件中发现了LONG DISTANCE域就揭露了敏感数据。
- 大概值
  - 居住在宾夕法尼亚大道1600号的政府官员有几个人?(回答：4)
  - 居住在宾夕法尼亚大道1600号的政府官员且“保守党”域是YES的有几个人?(回 答：1)
  - 通过查询，你可以推断总统是保守党党员的可能性是25%。
- 成功的安全策略必须能够同时防止直接揭露和 间接揭露敏感数据。

(8) 安全与精确度

- 从数据的机密性考虑
  - 我们只能透露一些不具敏感性的数据。透露的数据越少越好。
  - 我们因此会拒绝了许多合理的并非揭露性的查询。例如，研究员想要一个所有吸毒学生所在年级的列表，或统计员请求所有男性和女性的薪水表。这类查询不会泄露任何个人的身份。
- 从用户的情况考虑
  - 希望尽可能地显示数据以满足数据库用户的需求。这个目标称为精确度，旨在保护所有敏感数据的同时尽可能多地揭示非敏感数据。
  - 安全与精确度的理想结合要求我们维护完善的机密性与最大的精确性；换句话说，揭示所有的、但只有非敏感数据。


揭示数据尽可能精确
最低敏感级
最高敏感级
尽可能安全隐藏
需隐藏数据
不能通过查询
推理出的数据
可以通过查询
推理出的数据
一经查询就可透露的数据

4. 多级数据库管理系统体系结构

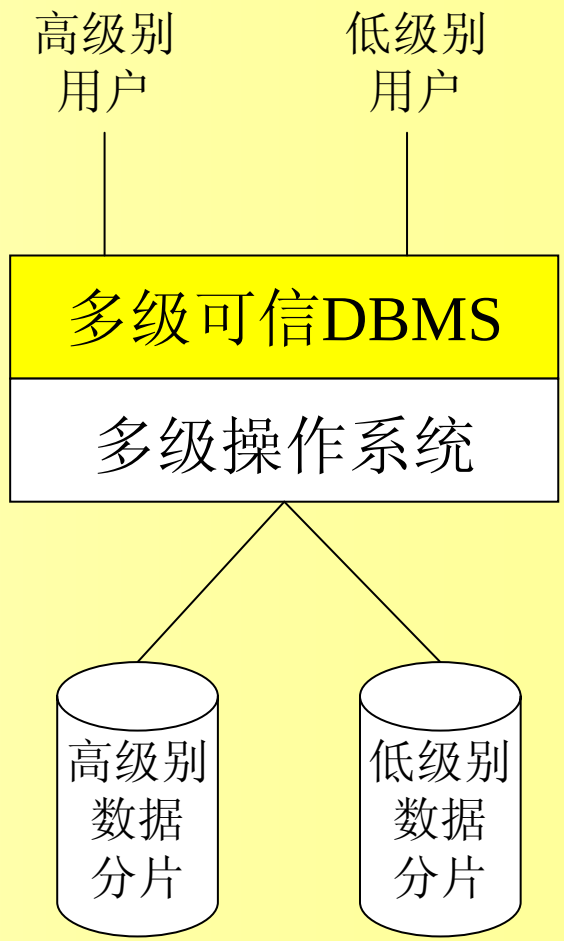
- 高级运行结构
- 可信主体结构
- 完整性锁结构
- 集中式结构
- 分布式结构

高级运行结构

- 它是单级DBMS的一种特殊运行方式。
- 非可信的数据库管理系统运行在安全操作系统的最高安全级，同时所有的数据库用户都工作在最高级。
- 由于所有的用户与数据都是最高安全级别，因此数据的输入/输出需要专门的人员负责数据信息的密级转换与管理。
- 由于这种方式完全依赖于操作系统的。数据库中的数据内容是不分级的。

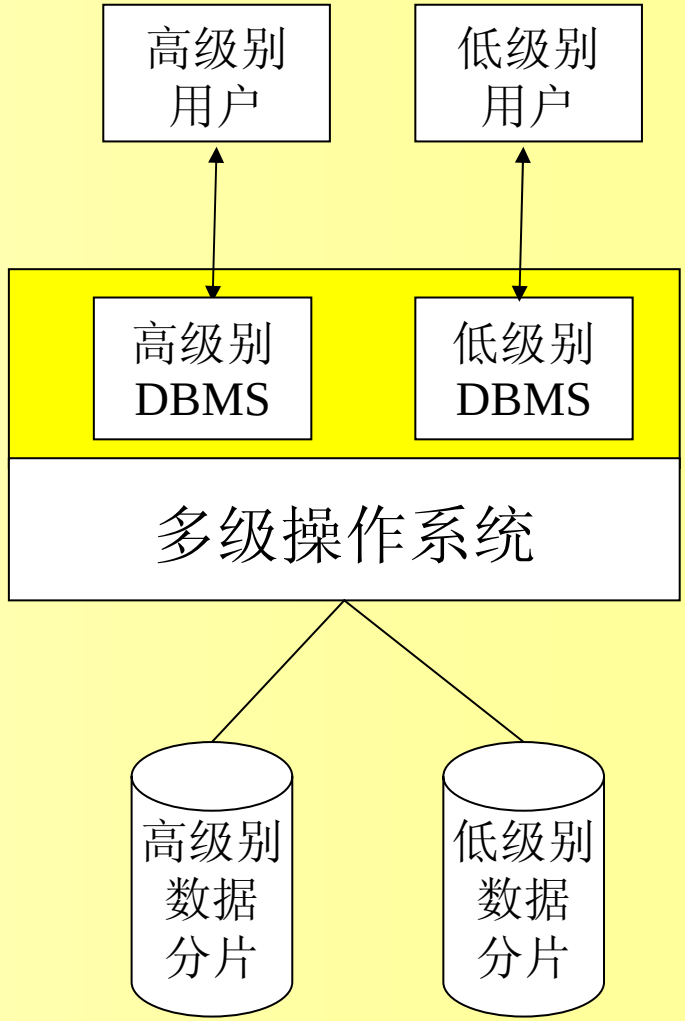
可信主体结构

- 可信主体结构(trusted subject architecture)也是一种单一DBMS的体系结构。
- 数据库管理系统自身是安全可信的，并且所管理的是多级数据库信息。
- 所有的DBMS数据带标记存储。可信DBMS维护并管理数据信息的多级属性。
- 数据库中不同粒度对象的安全级别由可信DBMS维护，对操作系统不可见。



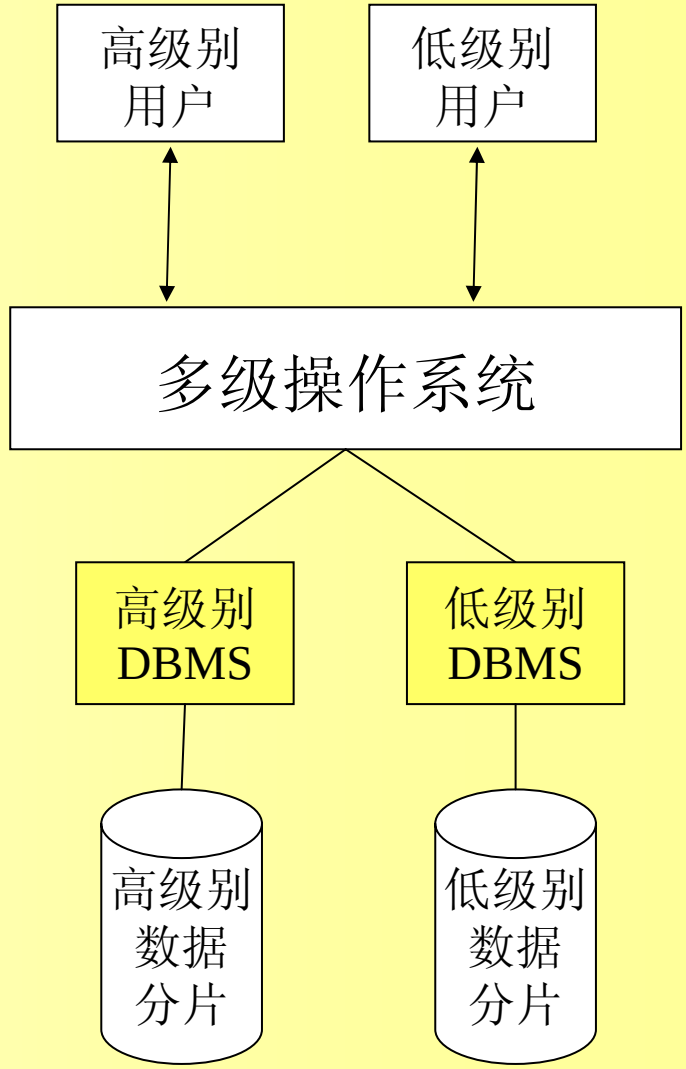
集中式结构

- 数据存储集中。数据库内容按级别分解后，作为操作系统的对象集中保存在安全操作系统中，任何级别的数据在数据库中只保存一份。
- 运行进程集中。不同级别的DBMS实例运行在同一个安全操作系统中。



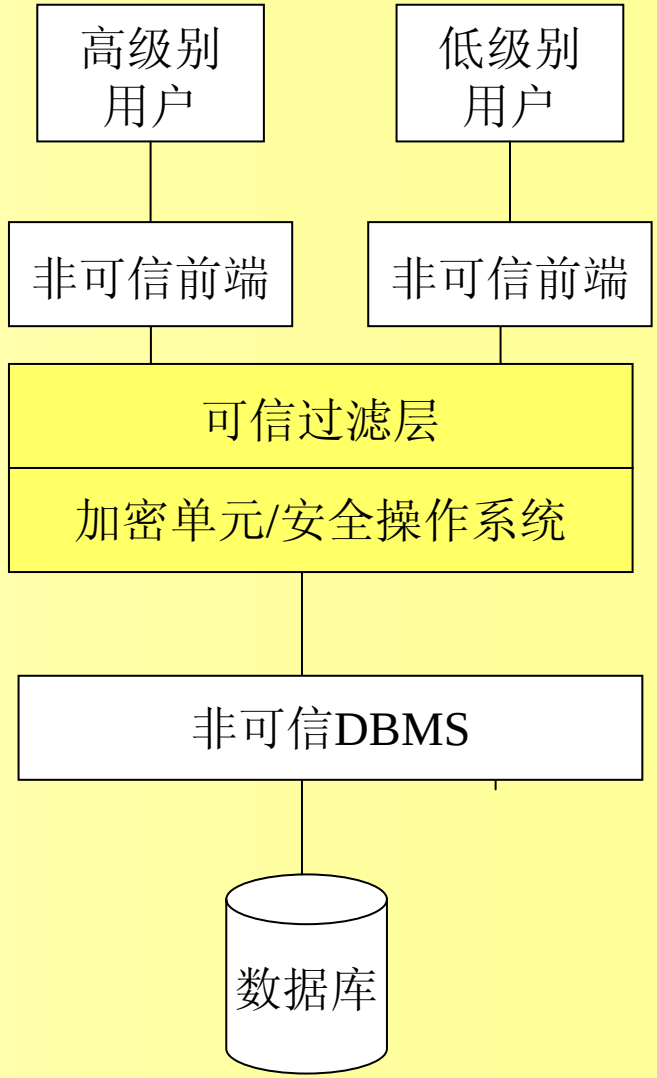
分布式结构

- 数据库中的内容不是集中保存，而是保存在多个站点。每个站点的安全级别不同。某个级别的站点中数据库的内容包括其所支配的所有安全级别的数据。因此，低级数据库中的内容被复制到安全级别支配它的所有高级数据库中。
- 每个站点上运行的DBMS实例只为该级别用户服务。不同级别的用户所连接访问的数据库不同。高级别用户所连接的后端数据库中包括了该级别以及该级别所支配的所有安全级别的数据。



完整性锁结构

- 用户通过非可信前端执行查询前后的处理过程。
- DBMS是非可信的，作为操作系统的高级别主体运行。
- 在非可信的前端与非可信的DBMS之间插入一个基于安全操作系统的可信过滤器，它作为TCB实现安全功能和多级别的保护。



5 推理攻击控制

- 推理问题(inference)
  - 一种通过非敏感数据推断或推导敏感数据的方法。推理问题是数据库安全中一个很微妙的弱点。

推理攻击控制

- 推理问题(inference)
  - 一种通过非敏感数据推断或推导敏感数据的方法。推理问题是数据库安全中一个很微妙的弱点。
- 假定AID, FINES,DRUGS都是敏感数据。

Name	Sex	Race	Aid	Fines	Drugs	Dorm
Adams	M	C	5000	45	1	Holmes
Bailey	M	B	0	0	0	Grey
Chin	F	A	3000	20	0	West
Dwitt	M	B	1000	35	3	Grey
Earhart	F	C	2000	95	1	Holmes
Fein	F	C	1000	15	0	West
Groff	M	C	4000	0	3	West
Hill	F	B	5000	10	2	Holmes
Koch	F	C	0	0	1	West
Lin	F	A	0	10	2	Grey
Majors	M	C	2000	0	2	Grey

(1)直接攻击

- 直接攻击是指用户试图通过直接查询敏感域，根据产生的少量记录决定敏感域的值。最成功的技术是形成一个与数据项精确匹配的查询。
  - List NAME where SEX=M ∧ DRUGS＝1
  - 一个比较隐蔽的查询是

    List NAME where (SEX=M ∧ DRUGS＝1) ∨
    (SEX≠M ∧ SEX≠F) ∨
    (DORM=AYRES)
- “**n个数据项超过k%**”规则，它表明：如果n个数据项代表了超过了k%的报告结果，则结果数据应该被保留而不能公布。

（2）间接攻击

- 另一个规定是只能发布统计数据。美国人口普查局和收集敏感数据的其他机构都使用了这一规定。这些机构禁止查询名字、住址或其他可以泄露个人身份的数据。只能发布统计数据，如计数、总数和平均数等。
- 间接攻击是根据一个或多个中间的统计值推理最后结果。

（3）和

- 通过“和”(SUM)攻击 试图从一个已知的和推理单个值。
- 例如按性别和寝室分类报告学生的资助总额似乎是安全的。报告结果如下所示，它似乎无辜地报告了住在Grey宿舍的女生没有接受资助。

	Holmes	Grey	West	Total
M	5000	3000	4000	12000
F	7000	0	4000	11000
Total	12000	3000	8000	23000

（4）计数

- 计数可以和总数结合起来揭露更多的结果。通常数据库给出这两个统计量以便用户确定平均值。
- 右表显示了学生按性别和寝室分类的计数值。这个表本身并没有泄露敏感数据。
- 然而，与总数表结合起来，就说明了住在Holmes和West的两位男性分别接受了$5000和$4000的资助。子模式NAME和DORM在整个数据库中属于安全级别低的数据，所以我们可以通过子模式获得具体的姓名。

	Holmes	Grey	West	Total
M	1	3	1	5
F	2	1	3	6
Total	3	4	4	11

	Holmes	Grey	West	Total
M	5000	3000	4000	12000
F	7000	0	4000	11000
Total	12000	3000	8000	23000

（5）控制统计推理攻击

- 禁止查询
  - 禁止查询方式就是不提供敏感数据；对敏感数据的查询以不响应的方式拒绝。
- 隐藏
  - 隐藏方式就是提供的结果接近但不是精确的实际数据值。

(5) 控制统计推理攻击

- 有限响应禁止
  - 根据n项k%的规则，不显示所占百分比过大的元素。
  - 然而，仅在显示结果中删除它们还是不够的，仍然有办法推理它们。

	Holmes	Grey	West	Total
M	1	3	1	5
F	2	1	3	6
Total	3	4	4	11

	Holmes	Grey	West	Total
M	-	3	-	5
F	2	-	3	6
Total	3	4	4	11

（5）控制统计推理攻击

- 组合结果

按性别和吸毒值查询的学生人数

SEX	Drug use			
	0	1	2	3
M	1	1	1	2
F	2	2	2	0

通过组合数据抑制推理

SEX	Drug use	
	0 or 1	2 or 3
M	2	3
F	4	2

（6）其他方法

- 随机样本
- 随机数据扰乱
- 查询分析

（7）推理问题的小结

- 禁止明显的敏感数据
  - 采取这个行动相当容易。但常常错误地选择了要禁止的数据，因而降低了数据库的可用性。
- 追踪用户已知的数据
  - 虽然这种方法在揭示数据时可能有最大的安全性，但是实现它的代价十分昂贵。必须为每个用户维护查询信息，即使其中有些用户并不打算访问敏感 数据。此外，这个方法很少考虑到两个用户把已知的结果结合起来以及一个用户以多种身份实现查询的情形。
- 伪装数据
  - 随机数扰乱或舍入可以限制依靠精确值进行逻辑和代数运算的统计攻击。数据库的用户得到的数据是稍微有点不精确或可能不一致的结果。

（8）聚集

- 聚集(aggregation)意味着从较低敏感性的输入构造出敏感数据。
  - 它与推理问题相关。例如只知道一个金矿的经度或纬度对你并没有益处。但是如果你既知道经度又知道纬度，就可以准确地找到金矿位置。
- 考虑警察是如何常常使用聚集技术来调查案件罪犯的
  - 他们确定谁有犯罪的动机、犯罪发生的时间、谁有当时不在场的证据、谁有实施犯罪的能力等。
  - 在通常情况下，警察调查案件是从一大群人开始的，然后逐渐缩小到对个人的调查。但是，如果警官并行工作，一些人调查一系列的嫌疑犯，另一些人分析研究可能的犯罪动机，还有一些人分析具有犯罪能力的人。如果发现某人在这几种调查结果的交集中，那么警察认定这个人是主要犯罪嫌疑人。

(8) 聚集

- 对于一个数据库安全的研究员来说，聚集与推理对安全问题的影响同样重要。
- 目前正使用的一些方法可以抑制推理攻击。
- 但是几乎没有方法被提出来制止聚集。
- 应当研究数据挖掘的保密问题。