華中科技大學

数据库系统原理

第4章　数据库安全性

学习通，交流

基础篇

第1章 绪论
第2章 关系数据库*3
第3章 标准语言SQL*3
第4章 数据库安全性
第5章 数据库完整性
实验1

设计与应用开发篇

第6章 关系数据理论*2
第7章 数据库设计
实验2
第8章 数据库编程

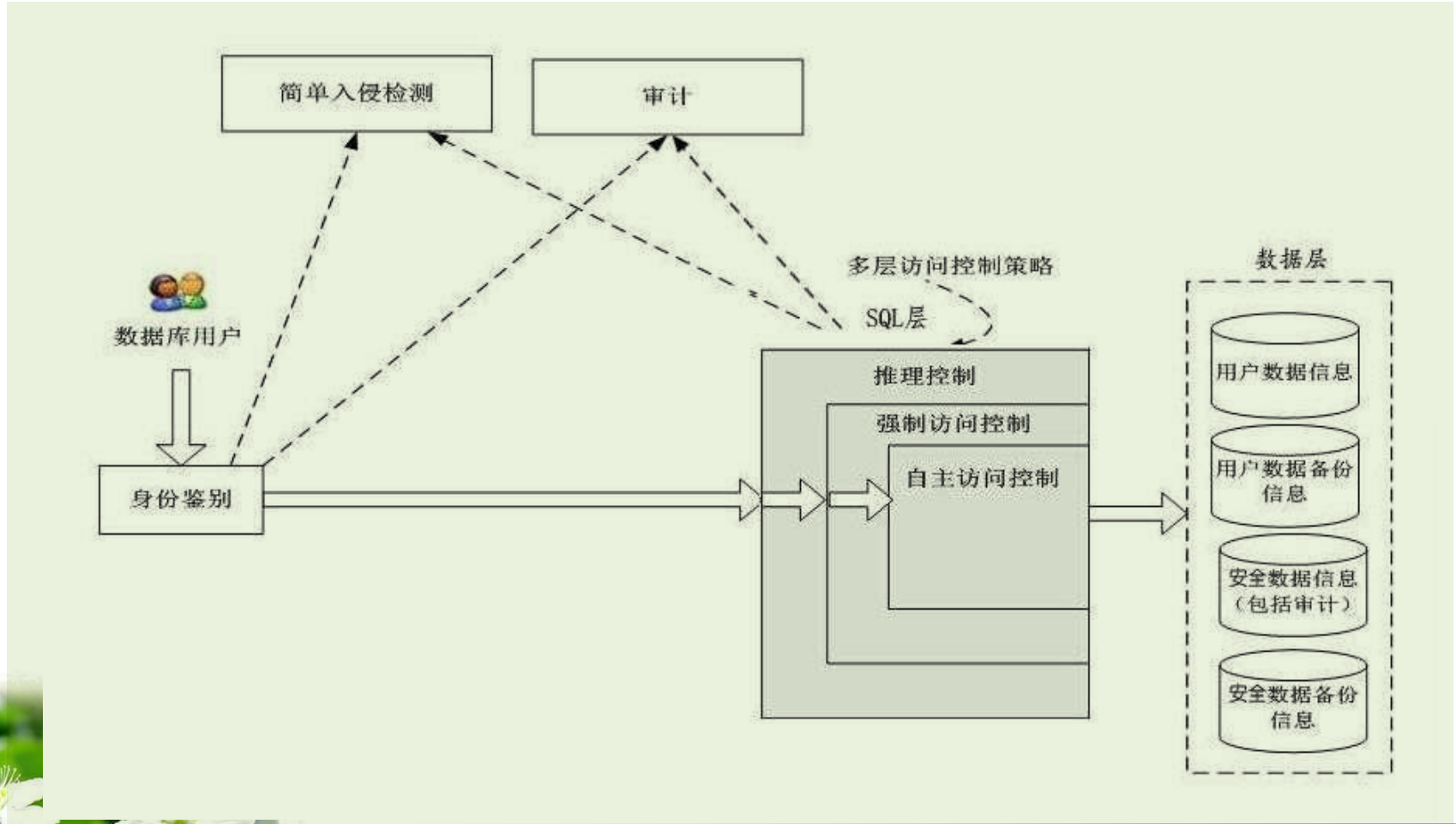
系统篇

第9章 *关系查询处理和优化
实验3
第10章 数据库恢复技术
第11章 并发控制
实验4

内容提要

- ✓ 数据库安全性控制
- ✓ 视图机制
- ✓ 审计（Audit）
- ✓ 数据加密
- ✓ 安全性保护

数据库的安全性控制


简单入侵检测
审计
多层访问控制策略
SQL层
数据层
数据库用户
推理控制
强制访问控制
自主访问控制
身份鉴别
用户数据信息
用户数据备份信息
安全数据信息（包括审计）
安全数据备份信息

安全性控制

自主访问控制DAC

- 自主存取控制是基于存取权限或特权概念及其实现技术与机制，哪个用户对哪个数据对象具有什么样的权限通过授权来说明。
- 用户必须获得适当的权限才可以对所请求的数据其进行相应的操作。

- 自主存取控制（Discretionary Access Control，简称DAC）
  - 用户对不同的数据对象有不同的存取权限
  - 用户还可将其拥有的存取权限转授给其他用户

安全性控制

强制存取控制MAC

- 强制存取控制是：每一个数据对象被（强制地）标以一定的密级，每一个用户也被（强制地）授予某一个级别的许可证。
- ✓ 系统规定只有具有某一许可证级别的用户才能存取某一个密级的数据对象。

- 强制存取控制（Mandatory Access Control，简称 MAC）
  - 对于任意一个对象，只有具有合法许可证的用户才可以存取

安全性控制

❖ 存取控制流程

- 首先，数据库管理系统对提出SQL访问请求的数据库用户进行身份鉴别，防止不可信用户使用系统。
- 然后，在SQL处理层进行自主存取控制和强制存取控制，进一步可以进行推理控制。
- 还可以对用户访问行为和系统关键操作进行审计，对异常用户行为进行简单入侵检测。

安全性控制

- 数据库安全性控制的常用方法
  - 用户标识和鉴定
  - 存取控制
  - 视图
  - 审计
  - 数据加密

4.2.1 用户身份鉴别

4.2.2 存取控制

4.2.3 自主存取控制方法

4.2.4 授权：授予与回收

4.2.5 数据库角色

4.2.6 强制存取控制方法

安全性控制--用户身份鉴别

1.静态口令鉴别

- 静态口令一般由用户自己设定，这些口令是静态不变的

2.动态口令鉴别

- 口令是动态变化的，每次鉴别时均需使用动态产生的新口令登录数据库管理系统，即采用一次一密的方法

3.生物特征鉴别

- 通过生物特征进行认证的技术，生物特征如指纹、虹膜和掌纹等

4.智能卡鉴别

- 智能卡是一种不可复制的硬件，内置集成电路的芯片，具有硬件加密功能

安全性控制--存取控制

- 用户权限定义和合法权检查机制一起组成了数据库管理系统的存取控制子系统
  - DBMS定义用户权限，并将用户权限登记到数据字典中，称做安全规则或授权规则
  - DBMS查找数据字典，进行合法权限检查

- 自主存取控制（Discretionary Access Control，简称DAC）
  - 用户对不同的数据对象有不同的存取权限
  - 用户还可将其拥有的存取权限转授给其他用户

- 强制存取控制（Mandatory Access Control，简称 MAC）
  - 每一个数据对象被标以一定的密级
  - 每一个用户也被授予某一个级别的许可证

安全性控制—自主存取控制

- 通过 SQL 的GRANT 语句和REVOKE 语句实现
- 用户权限组成
  - 数据对象
  - 操作类型
    - ✓ 定义用户存取权限：定义用户可以在哪些数据库对象上进行哪些类型的操作
    - ✓ 定义存取权限称为授权

安全性控制—自主存取控制

对象类型	对象	操 作 类 型
数据库模式	模式	CREATE SCHEMA
	基本表	CREATE TABLE，ALTER TABLE
	视图	CREATE VIEW
	索引	CREATE INDEX
数据	基本表和视图	SELECT，INSERT，UPDATE，DELETE，REFERENCES，ALL PRIVILEGES
	属性列	SELECT，INSERT，UPDATE，REFERENCES，ALL PRIVILEGES

安全性控制—授予

- GRANT语句的一般格式：

```
GRANT <权限>[,<权限>]...
ON <对象类型> <对象名>[,<对象类型> <对象名>]...
TO <用户>[,<用户>]...
[WITH GRANT OPTION];
```

- 语义：将对指定操作对象的指定操作权限授予指定的用户

[例4.1] 把查询Student表权限授给用户U1

```
GRANT   SELECT
ON   TABLE   Student
TO   U1;
```

安全性控制—授予

[例4.2] 把对Student表和Course表的全部权限授予用户U2和U3

```
GRANT ALL PRIVILIGES
ON TABLE Student,Course
TO U2,U3;
```

[例4.3] 把对表SC的查询权限授予所有用户

```
GRANT SELECT ON TABLE SC TO PUBLIC;
```

[例4.4] 把查询Student表和修改学生学号的权限授给用户U4

```
GRANT UPDATE(Sno), SELECT
ON TABLE Student TO U4;
```

- 对属性列的授权时必须明确指出相应属性列名

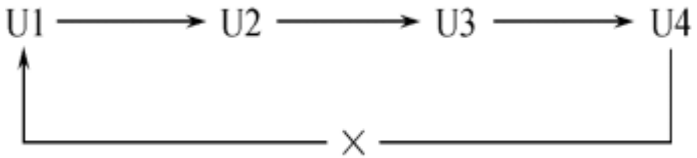
安全性控制—授予

[例4.5] 把对表SC的INSERT权限授予U5用户，并允许他再将此权限授予其他用户

```
GRANT INSERT     ON TABLE SC     TO U5
WITH GRANT OPTION;
```

执行例4.5后，U5不仅拥有了对表SC的INSERT权限， 还可以传播此权限：

[例4.6] GRANT INSERT ON TABLE SC TO U6 WITH GRANT OPTION;



同样，U6还可以将此权限授予U7：

[例4.7] GRANT INSERT ON TABLE SC TO U7;

U7不能再传播此权限。

安全性控制—授予

执行了例4.1~例4.7语句后学生-课程数据库中的用户权限定义表

授权用户名	被授权用户名	数据库对象名	允许的操作类型	能否转授权
DBA	U1	关系Student	SELECT	不能
DBA	U2	关系Student	ALL	不能
DBA	U2	关系Course	ALL	不能
DBA	U3	关系Student	ALL	不能
DBA	U3	关系Course	ALL	不能
DBA	PUBLIC	关系SC	SELECT	不能
DBA	U4	关系Student	SELECT	不能
DBA	U4	属性列 Student.Sno	UPDATE	不能
DBA	U5	关系SC	INSERT	能
U5	U6	关系SC	INSERT	能
U6	U7	关系SC	INSERT	不能

安全性控制—回收

- 授予的权限可以由数据库管理员或其他授权者用REVOKE语句收回
  - REVOKE语句的一般格式为：

```
REVOKE <权限>[,<权限>]...
ON <对象类型> <对象名>[,<对象类型><对象名>]...
FROM <用户>[,<用户>]...[CASCADE | RESTRICT];
```

[例4.8] 把用户U4修改学生学号的权限收回

```
REVOKE UPDATE(Sno)
ON TABLE Student
FROM U4;
```

安全性控制—回收

[例4.9] 收回所有用户对表SC的查询权限

```
REVOKE SELECT   ON TABLE SC
FROM PUBLIC;
```

[例4.10] 把用户U5对SC表的INSERT权限收回

```
REVOKE INSERT   ON TABLE SC
FROM U5 CASCADE ;
```

- 将用户U5的INSERT权限收回的时候应该使用CASCADE
- 如果U6或U7还从其他用户处获得对SC表的INSERT权限，则他们仍具有此权限，系统只收回直接或间接从U5处获得的权限

安全性控制

- 数据库管理员：
  - 拥有所有对象的所有权限
  - 根据实际情况不同的权限授予不同的用户
- 用户：
  - 拥有自己建立的对象的全部的操作权限
  - 可以使用GRANT，把权限授予其他用户
- 被授权的用户
  - 如果具有“继续授权”的许可，可以把获得的权限再授予其他用户
- 所有授予出去的权力都可用REVOKE语句收回

安全性控制

- CREATE USER语句格式

```
CREATE  USER  <username>
[WITH][DBA|RESOURCE|CONNECT];
```

  - CREATE USER不是SQL标准，各个系统的实现相差甚远

- 新创建的数据库用户有三种权限：CONNECT、RESOURCE和DBA
  - 默认拥有CONNECT权限。
  - 拥有CONNECT权限的用户不能创建新用户，不能创建模式，也不能创建基本表，只能登录数据库。
  - 拥有RESOURCE权限的用户能创建基本表和视图，成为所创建对象的属主。但不能创建模式，不能创建新的用户。

安全性控制—数据库角色

- 数据库角色：被命名的一组与数据库操作相关的权限
  - 角色是权限的集合
  - 可以为一组具有相同权限的用户创建一个角色
  - 简化授权的过程

1.角色的创建　CREATE ROLE <角色名>

2.给角色授权　GRANT <权限>[,<权限>]…

ON <对象类型>对象名　TO <角色>[,<角色>]…

3.将一个角色授予其他的角色或用户

GRANT <角色1>[,<角色2>]… TO <角色3>[,<用户1>]…

[WITH ADMIN OPTION]

安全性控制—数据库角色

4.角色权限的收回

REVOKE <权限>[,<权限>]...

ON <对象类型> <对象名>

FROM <角色>[,<角色>]...

- 用户可以回收角色的权限，从而修改角色拥有的权限
- REVOKE执行者是
  - 角色的创建者
  - 拥有在这个（些）角色上的ADMIN OPTION

安全性控制—数据库角色

[例4.11] 通过角色来实现将一组权限授予一个用户。

（1）首先创建一个角色 R1　　CREATE ROLE R1;

（2）然后使用GRANT语句，使角色R1拥有Student表的　SELECT、UPDATE、INSERT权限

```
GRANT SELECT, UPDATE, INSERT
ON TABLE Student  TO R1;
```

（3）将这个角色授予王平，张明，赵玲。使他们具有角色R1所包含的全部权限

```
GRANT R1  TO 王平,张明,赵玲;
```

（4） 可以一次性通过R1来回收王平的这3个权限

```
REVOKE R1  FROM 王平;
```

[例4.12] 角色的权限修改　　GRANT DELETE　ON TABLE Student　TO R1;

[例4.13]　REVOKE SELECT　ON TABLE Student　FROM R1;

安全性控制—强制存取控制

- 强制存取控制（MAC）
  - 是对数据本身进行密级标记，无论数据如何复制，标记与数据是一个不可分的整体，只有符合密级标记要求的用户才可以操纵数据。
  - 实现强制存取控制时要首先实现自主存取控制
- 自主存取控制与强制存取控制共同构成数据库管理系统的安全机制

安全性控制—强制存取控制

- 在强制存取控制中，数据库管理系统所管理的全部实体被分为主体和客体两大类
- 主体是系统中的活动实体
  - 数据库管理系统所管理的实际用户
- 客体是系统中的被动实体，受主体操纵
  - 文件、基本表、索引、视图
- 保证更高程度的安全性
- 适用于对数据有严格而固定密级分类的部门
  - 军事部门、 政府部门

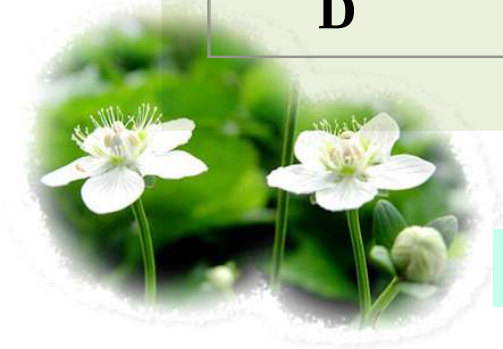
安全性控制—强制存取控制

- 敏感度标记（Label）
  - 对于主体和客体，DBMS为它们每个实例（值）指派一个敏感度标记（Label）
  - 敏感度标记分成若干级别
    - 绝密（Top Secret，TS）
    - 机密（Secret，S）
    - 可信（Confidential，C）
    - 公开（Public，P）
    - TS>=S>=C>=P
- 主体的敏感度标记称为许可证级别（Clearance Level）
- 客体的敏感度标记称为密级（Classification Level）

安全性控制

TCSEC/TDI安全级别划分

安全级别	定　　义
A1	验证设计（Verified Design）
B3	安全域（Security Domains）
B2	结构化保护（Structural Protection）
B1	标记安全保护（Labeled Security Protection）
C2	受控的存取保护（Controlled Access Protection）
C1	自主安全保护（Discretionary Security Protection）
D	最小保护（Minimal Protection）



安全性控制

- C1级
  - 能够实现对用户和数据的分离，进行自主存取控制（DAC），保护或限制用户权限的传播。
  - 现有的商业系统稍作改进即可满足
- C2级
  - 安全产品的最低档次
  - 提供受控的存取保护，将C1级的DAC进一步细化，以个人身份注册负责，并实施审计和资源隔离
  - 典型例子
    - Windows 2000
    - Oracle 7

安全性控制

- B1级
  - 标记安全保护。“安全”（Security）或“可信的”（Trusted）产品。
  - 对系统的数据加以标记，对标记的主体和客体实施强制存取控制（MAC）、审计等安全机制
  - B1级典型例子
    - 操作系统
      - 惠普公司的HP-UX BLS release 9.09+
    - 数据库
      - Oracle公司的Trusted Oracle 7
      - Sybase公司的Secure SQL Server version 11.0.6

安全性控制

- CC评估保证级（EAL）划分

评估保证级	定　　义	TCSEC安全级别（近似相当）
EAL1	功能测试（**functionally tested**）	
EAL2	结构测试（**structurally tested**）	**C1**
EAL3	系统地测试和检查（**methodically tested and checked**）	**C2**
EAL4	系统地设计、测试和复查（**methodically designed， tested， and reviewed**）	**B1**
EAL5	半形式化设计和测试（**semiformally designed and tested**）	**B2**
EAL6	半形式化验证的设计和测试（**semiformally verified design and tested**）	**B3**
EAL7	形式化验证的设计和测试（**formally verified design and tested**）	**A1**

安全性控制—视图机制

- 把要保密的数据对无权存取这些数据的用户隐藏起来，对数据提供一定程度的安全保护

[例4.14] 建立计算机系学生的视图，把对该视图的SELECT权限授于王平，把该视图上的所有操作权限授于张明

先建立计算机系学生的视图CS_Student

```
CREATE VIEW CS_Student    AS
        SELECT  *    FROM   Student    WHERE  Sdept='CS';
```

在视图上进一步定义存取权限

```
GRANT  SELECT    ON  CS_Student    TO 王平;
GRANT ALL PRIVILIGES  ON  CS_Student    TO  张明;
```

安全性控制—审计机制

- 启用一个专用的审计日志（Audit Log）
  - 将用户对数据库的所有操作记录在上面
- 审计员利用审计日志
  - 监控数据库中的各种行为，找出非法存取数据的人、时间和内容
- C2以上安全级别的DBMS必须具有审计功能

安全性控制—审计机制

审计事件

- 服务器事件
  - 审计数据库服务器发生的事件
- 系统权限
  - 对系统拥有的结构或模式对象进行操作的审计
- 语句事件
  - 对SQL语句，如DDL、DML、DQL及DCL语句的审计
- 模式对象事件
  - 对特定模式对象上进行的SELECT或DML操作的审计

- 防止审计员误删审计记录，审计日志必须先转储后删除

- 只允许审计员查阅和转储审计记录，不允许任何用户新增和修改审计记录等

安全性控制—审计机制

AUDIT语句和NOAUDIT语句

- AUDIT语句：设置审计功能
- NOAUDIT语句：取消审计功能

[例4.15] 对修改SC表结构或修改SC表数据的操作进行审计

```
AUDIT ALTER,UPDATE
ON  SC;
```

[例4.16] 取消对SC表的一切审计

```
NOAUDIT  ALTER,UPDATE
ON  SC;
```

安全性控制—数据加密

- 数据加密
  - 防止数据库中数据在存储和传输中失密的有效手段
- 加密的基本思想
  - 根据一定的算法将原始数据—明文（Plain text）变换为不可直接识别的格式—密文（Cipher text）
- 加密方法
  - 存储加密
  - 传输加密

安全性控制—数据加密

- 存储加密
  - 透明存储加密
    - 内核级加密保护方式，对用户完全透明
    - 将数据在写到磁盘时对数据进行加密，授权用户读取数据时再对其进行解密
    - 数据库的应用程序不需要做任何修改，只需在创建表语句中说明需加密的字段即可

    内核级加密方法：性能较好，安全完备性较高
  - 非透明存储加密
    - 通过多个加密函数实现

安全性控制—数据加密

- 传输加密
  - 链路加密
    - 在链路层进行加密
    - 传输信息由报头和报文两部分组成
    - 报文和报头均加密
  - 端到端加密
    - 在发送端加密，接收端解密
    - 只加密报文不加密报头
    - 所需密码设备数量相对较少，容易被非法监听者发现并从中获取敏感信息

安全性控制—数据加密

用户

可信通讯模块

第一步：创建可信连接

第二步：确认通信双方端点的可靠性

第三步：协商加密算法和密钥

第四步：可信传输数据

第五步：关闭可信连接

可信通讯模块

数据库服务器

数据库管理系统可信传输示意图

安全性控制—数据加密

基于安全套接层协议SSL传输方案的实现思路：

（1）确认通信双方端点的可靠性

- 采用基于数字证书的服务器和客户端认证方式
- 通信时均首先向对方提供己方证书，然后使用本地的CA 信任列表和证书撤销列表对接收到的对方证书进行验证

（2）协商加密算法和密钥

- 通信双方协商本次会话的加密算法与密钥

（3）可靠数据传输

- 用某一组特定的密钥进行加密和消息摘要计算，以密文传输
- 接收的时候，需用相同一组特定的密钥进行解密和摘要计算

安全性控制—其他安全性保护

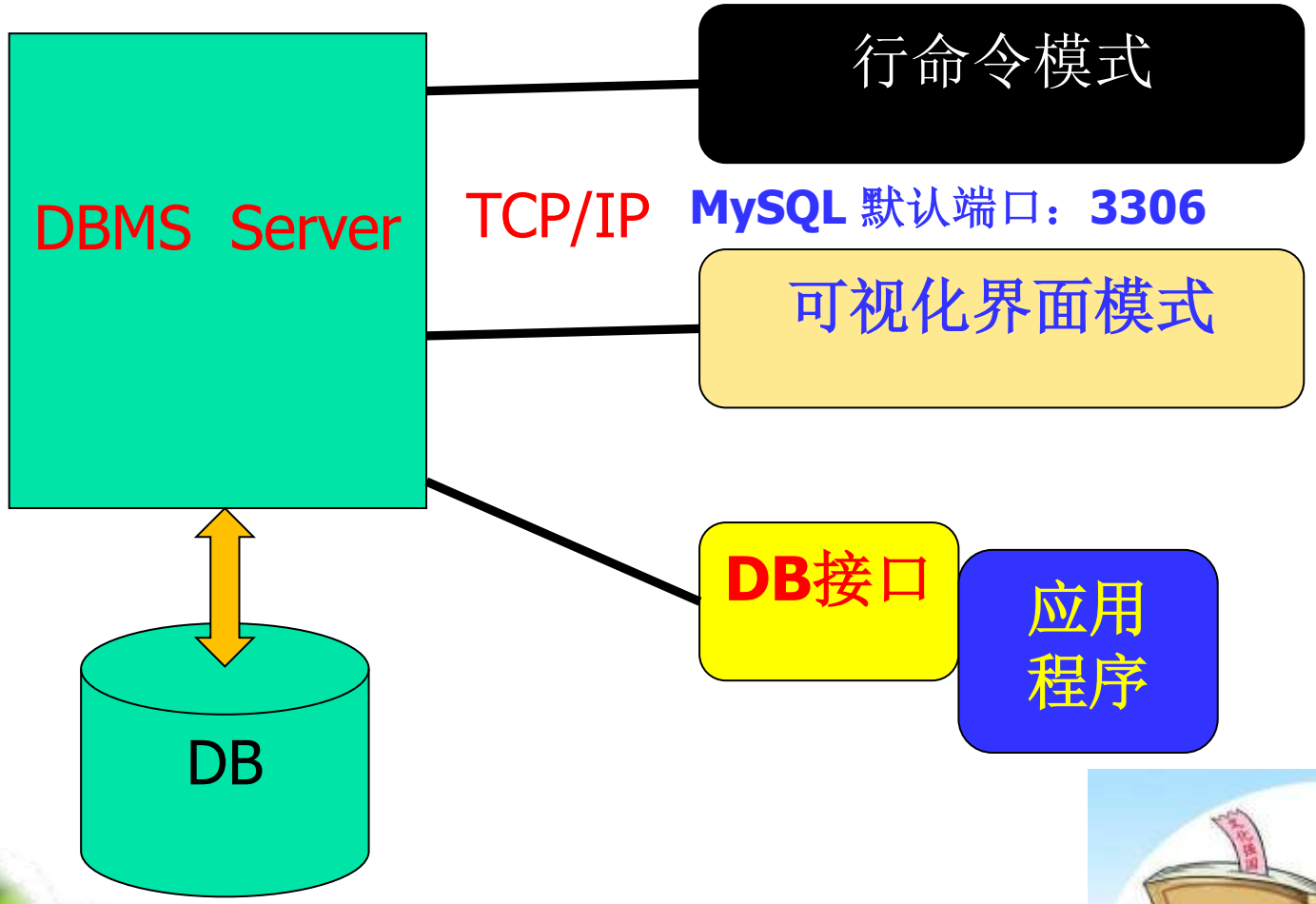
- 推理控制
  - 处理强制存取控制未解决的问题
  - 避免用户利用能够访问的数据推知更高密级的数据
  - 常用方法
    - 基于函数依赖的推理控制
    - 基于敏感关联的推理控制
- 隐蔽信道
  - 处理强制存取控制未解决的问题
- 数据隐私保护
  - 描述个人控制其不愿他人知道或他人不便知道的个人数据的能力
  - 范围很广：数据收集、数据存储、数据处理和数据发布等阶段

小结

- ✓ 用户身份鉴别
- ✓ 存取控制技术：
  - ➢ 自主存取控制
  - ➢ 强制存取控制
- ✓ 视图技术
- ✓ 审计技术
- ✓ 数据加密存储和加密传输

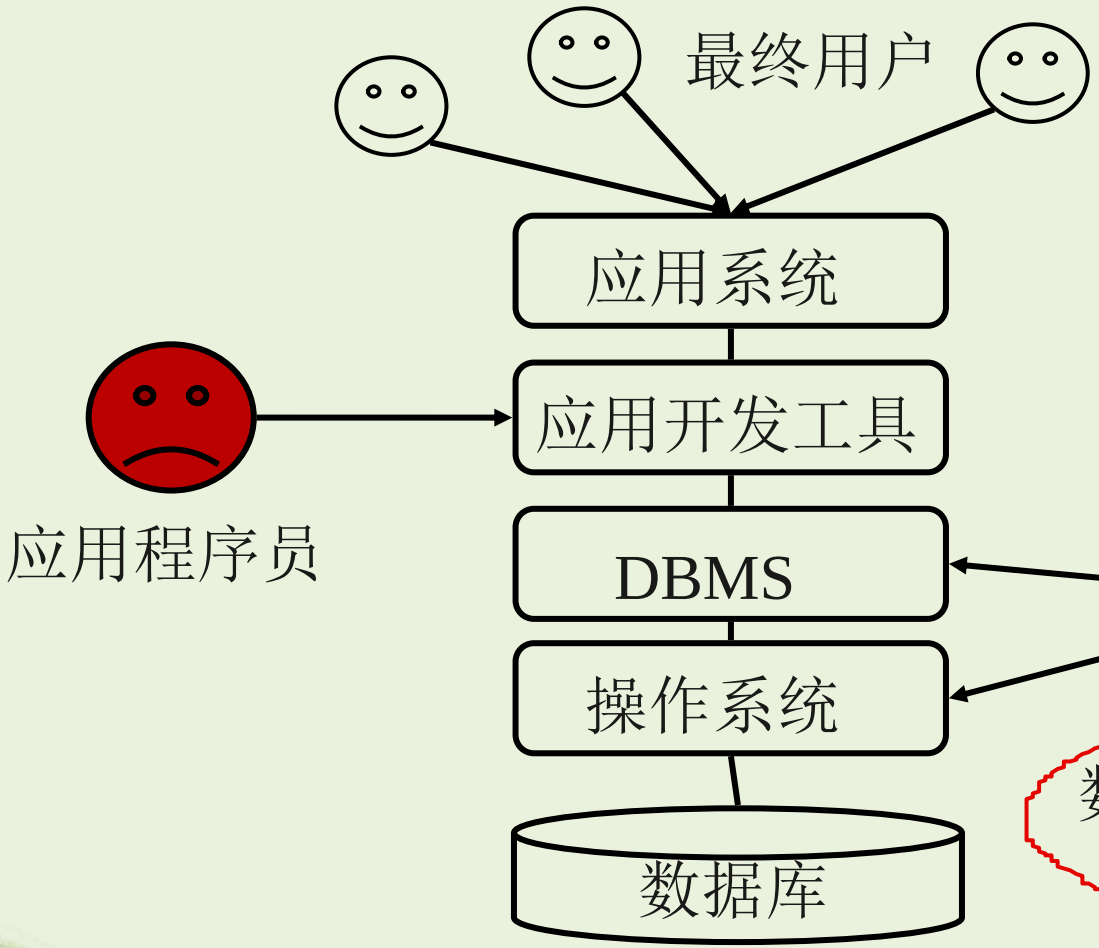
实际应用

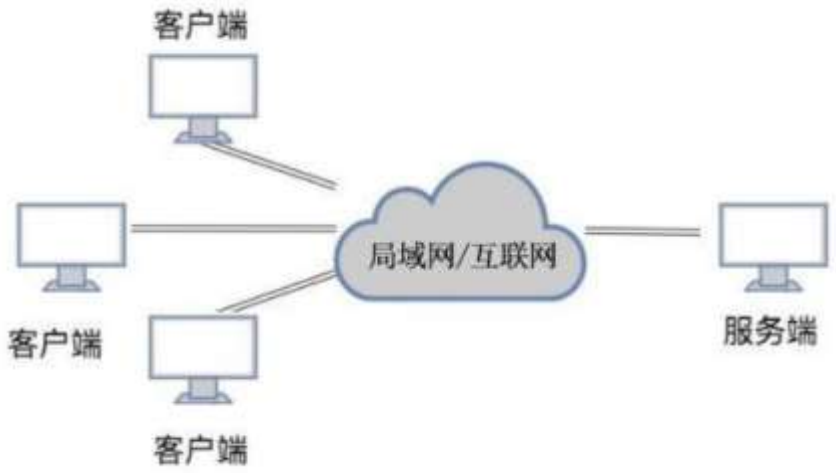
- 如何实施数据库应用系统的安全问题？
  - ✓ 视图机制
  - ✓ 自主存取控制措施
  - ✓ 应用程序段的角色
  - ✓ 审计
  - ✓ 数据加密
  - ✓ 安全性保护

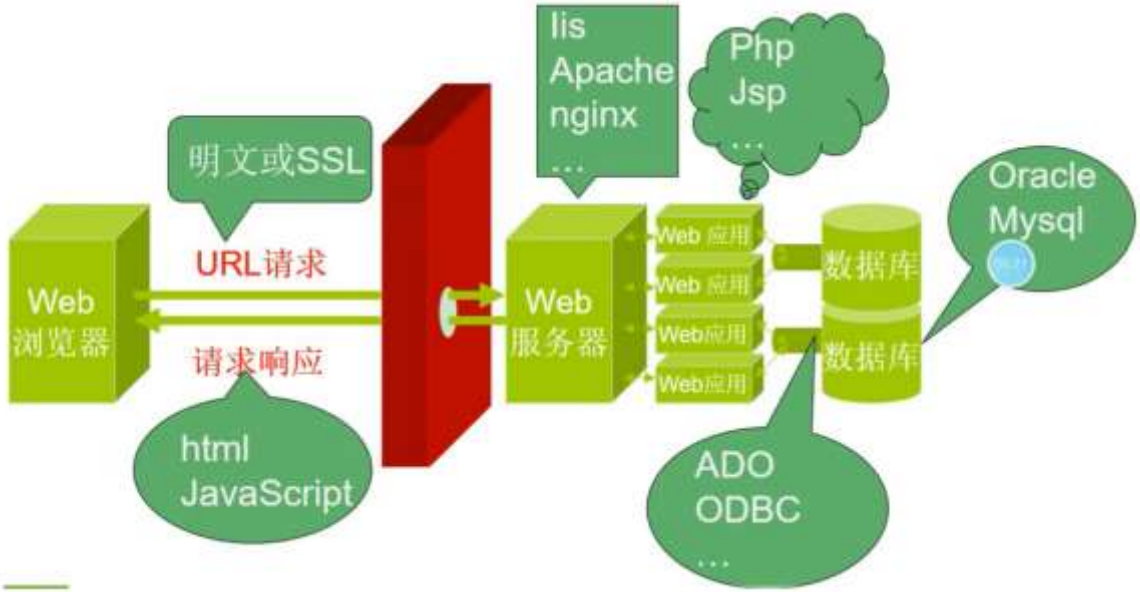

DBMS Server
TCP/IP
MySQL 默认端口：3306
行命令模式
可视化界面模式
DB接口
应用程序
DB


全民
阅读

数据库系统的构成




客户端
局域网/互联网
客户端
服务端
客户端


明文或SSL
URL请求
请求响应
Web
浏览器
html
JavaScript
Iis
Apache
nginx
…
Web
服务器
Php
Jsp
…
Web 应用
Web 应用
Web应用
Web应用
数据库
数据库
Oracle
Mysql
ADO
ODBC
…

主要功能：

- **屏蔽路径**：防火墙介于数据库服务器和应用服务器之间，屏蔽直接访问的通道
- **访问控制**：对数据库的访问行为进行细粒度访问控制，对高危的SQL访问语句与行为进行实时阻断
- **漏洞防护**：及时检测阻断SQL注入或缓冲区溢出等攻击
- **黑白名单**：通过黑白名单限制数据库的访问
- **安全审计**：支持事中、事后数据库审计，有效支持事件溯源，满足合规要求。