# 数据库系统简介

## 数据库系统的目标

为了解决文件系统中存储组织信息的弊端：

1. 数据的冗余和不一致：相同的信息可能在不同的几个地方（文件）重复存储，导致数据冗余；相同信息存储在不同文件时，当修改其中一个文件的该信息，可能导致其他文件的该信息未被修改，导致数据不一致。
2. 数据访问困难：每次检索特定需求的某些数据时，需要每次重新编写程序对不同文件进行检索。因此我们需要开发出通用的、能对变化需要作出快速反应的检索系统。
3. 数据孤立：由于数据分散在不同文件，而文件又存在不同的格式，因此需要编写新程序来检索文件中的适当的数据非常麻烦。
4. 完整性问题：增加、修改、删除数据的约束相当复杂麻烦，尤其是约束涉及多个文件时。
5. 原子性问题：传统的文件系统很难做到。原子操作例如：A系统转账500元到B系统时，A余额减去500元时B系统还没来得急增加500元。 A减少500元与B增加500元为一个原子操作，要么都发生，要么都还原。
6. 并发访问异常：A系统余额10000元， 两个人同时取出，一个人取出500，一个人取出400，

他们同时读到数据是10000元，取出钱后，导致一个为9500元。，另一个为9600元元，写入系统则出现余额9500或者9600元（正确为9100元）。

1. 安全性：如限制应用程序对某些文件的数据访问比较困难。

# 数据模型

## 关系模型

关系数据库的构成：由表的集合组成，每个表都具有唯一的名字。

为什么叫关系数据库：表中的一行代表了一组值的关系，表的概念与数学的关系密切相关，因此叫关系数据模型。

关系用来指代表，元组用来指代行，属性用来指代列。属性的域表示属性的取值范围。

### 超键(surper key)

在关系中能唯一标识元组的属性集称为关系模式的超键（只要有一个键唯一，再随便组其他的键，合起来叫主键）。如（学号）、（学号，姓名）、（身份证号，性别）都属于超键。

### 候选键(candidate key)

不含有多余属性的超键称为候选键，候选键属于超键，它是最小的超键，就是说如果再去掉候选键中的任何一个属性它就不再是超键了。 如（学号，姓名）由于去掉（姓名），（学号）还能表示超键， 因此（姓名）属于多余属性，则（学号，姓名）不属于候选键。

### 主键(primary key)

关系型数据库中的一条记录中有若干个属性，若其中某一个属性集(注意是集)能唯一标识一条记录，该属性组就可以成为一个主键 （在超键选取一个作为主键，如果有多个字段的叫为联合主键）

主键的作用：1. 惟一地标识一行。2. 作为一个可以被外键有效引用的对象。

### 外键(foreign key)

如果关系模式R1中的某属性集不是R1的主键，而是另一个关系R2的主键则该属性集是关系模式R1的外键， R1为外键的主表，R2为外键的子表。如学生表中的外键是(教师编号)。

外键的作用：保持数据的一致性、完整性。 如何保证：例如禁止在子表中插入一条在主表中不存在的数据；或者禁止在先删除主表的数据，导致子表该数据在主表中找不到。

外键的使用规范：1.外键必须是主表的唯一键。2.主表子表字段类型一致。

学生（学号，姓名，性别，身份证号，教师编号）

教师（教师编号，姓名，工资）

# SQL语句

SQL：结构化查询语言(Structured Query Language)，是一种关系型数据库查询语言，不仅具有查询，还支持定义数据结构、修改数据库中数据及安全约束性等。

SQL语句对大小写字母不敏感。

## 数据定义语言(DDL)

### 数据类型

### 创建表

主键约束建立：

create table testtable

(name varchar(20), primary key (name));

外键约束建立：

1. 先建立主表

create table table1 (id numeric(12,2), name1 varchar(20), age numeric(12,2), PRIMARY key(id), UNIQUE(name1))

1. 在建立从表

create table table2 (id numeric(12,2), name2 varchar(20), age numeric(12,2), PRIMARY key(name2), foreign key(name2) references table1(name1))

### 修改表

添加一列：alter table testtable add name varchar(10)

### 删除表

drop table table\_name

## 数据操纵语言(DML)

# 各类数据库管理系统