



# 计算思维方法4

Created	@2025年12月23日 12:51
Class	计算思维

## 栈的储存与处理

栈由下到上分为栈底和栈顶，栈顶允许插入和删除，栈底不允许插入和删除。

指针变量top被称为栈顶指针，指向待输入元素。

```
int a[7],top;
top=0; //设置数组a中元素指针
for(int i=0;i<7;i++)
{
    top++; //指针后移
    a[top]=m; //入栈赋值
    top+=1;
}
```

若有A,B,C三个元素入栈，顺序为A,B,C，则可能出栈的顺序为

A,B,C→a进a出，b进b出，c进c出✓

B,A,C→a进，b进b出，a出c进，c出→满足A,B,C✓

A,C,B→a进a出，b进c进c出，b出✓

B,C,A→a进b进b出，a出c进c出✓

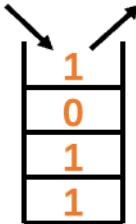
C,B,A→a进，b进，c进，c出，b出，a出✓

C,A,B→a进b进c进，c出，此时b在a上面，无法跳过b出栈✗

## 数制转换

$$N = (N \text{div} d) * d + N \text{mod} d$$

N	<u>N</u> div 2	N mod 2
11	5	1
5	2	1
2	1	0
1	0	1



$$(11)_{10} = (1011)_2$$

当输入一个值N，将N%r，及N对r取余的值压入栈中，位于top0，用N/r代替N进行如上操作后反复将结果压入栈中，得到进制转换后的数，再出栈倒序输出，得到转换值。

## 队列的存储与处理

队列与栈相同，操作位置都受到限制。队列只能在一端插入，在另一端删除。先进先出。

队列的入队顺序和出队顺序相同。

队可以看作一个一维数组。设一维数组a，共有9个存储单元。其中指针front和rear分别指向队列开头和结尾的元素位置，则有以下

```

int rear=0,front=0;
char a[9];
for(int i=0;i<9;i++)
{
    a[rear]=m; //rear指向队列结尾元素，负责元素入队
    rear+=1;
    m=a[front]; //front指向队列开头元素位置，负责元素出队
    front+=1;
}

```

队列中一定会出现假溢出现象，当数组a中9个存储单元均被元素填充时，front移动至a9，rear移动至a10，此时rear=10，认为溢出，实际并未溢出。

解决办法：存入a0，将队列视为首尾相接的圆环，这种队列为循环队列，rear和front在结尾向开始跳转。

## 数据管理基本概念

- 在20世纪50年代中之前，数据由人工管理；采用科学计算，无直接存储设备，无操作系统。数据无共享，应用程序对应单一数据集。
- 20世纪50年代后期～20世纪60年代中期，出现操作系统，采用磁盘存储数据，但修改不共享，依赖应用程序，独立性差。
- 20世纪60年代以来，同一数据库可以由多个操作端口控制，多用户，多应用共享。出现数据库管理系统。
- 数据库系统是由数据库、软件、硬件、用户构成，有时数据库系统也简称为数据库。

DBMS，数据库管理系统，一般的操作流程是，用户向dbms提出操作请求，dbms分析用户请求，控制转移至操作系统；操作系统分析指令，在数据库实现具体操作；dbms输出缓冲区数据。

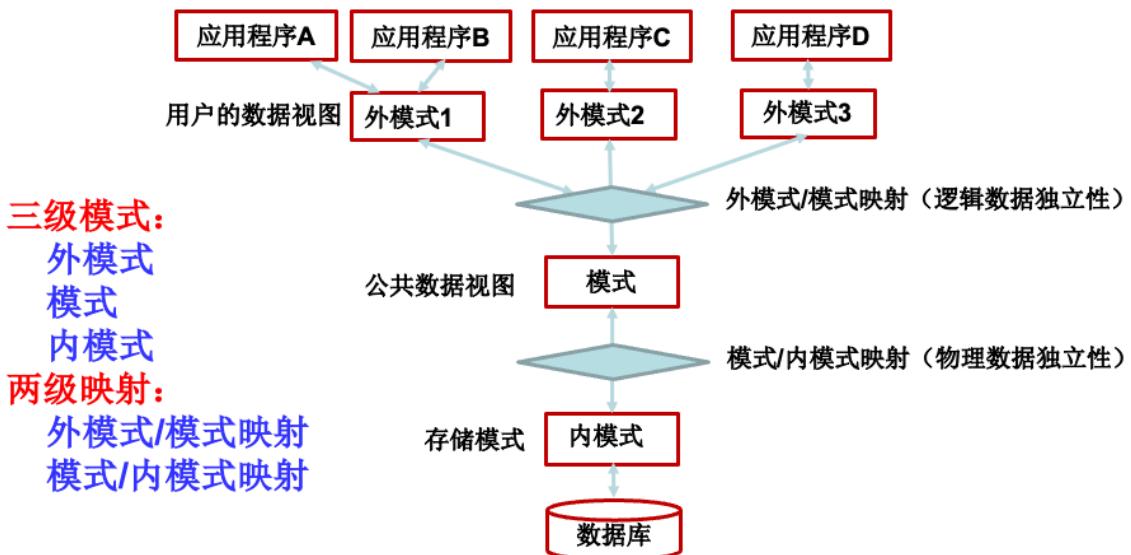
DBMS的基本功能：

1. 数据定义
2. 数据操纵
3. 数据库控制
4. 数据组织、存储和管理
5. 其他功能

## 数据库的基本特征

1. 整体数据结构化
2. 数据共享度高，冗余度低
3. 数据独立性高（物理独立性，逻辑独立性）

## 数据库存储模式——三级模式、二级映射



## 数据表示

数据库系统的设计：需求分析→概念设计→逻辑设计→物理设计

数据概念模型：

- 实体：客观存在可区分的事物
- 联系：实体与实体之间的联系
- 属性：实体或联系所具有的特征——例如学生和课程建立联系后具有成绩属性

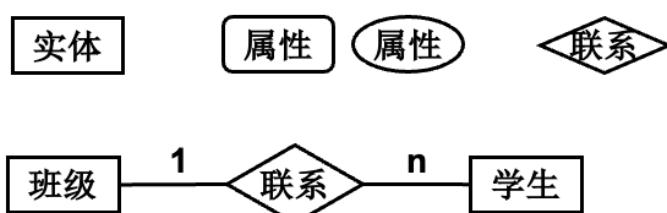
联系可以分为三种：

- 一对一联系：班级和班长——1:1
- 一对多联系：班级和学生——1:n
- 多对多联系：课程和学生——m:n

## E-R图

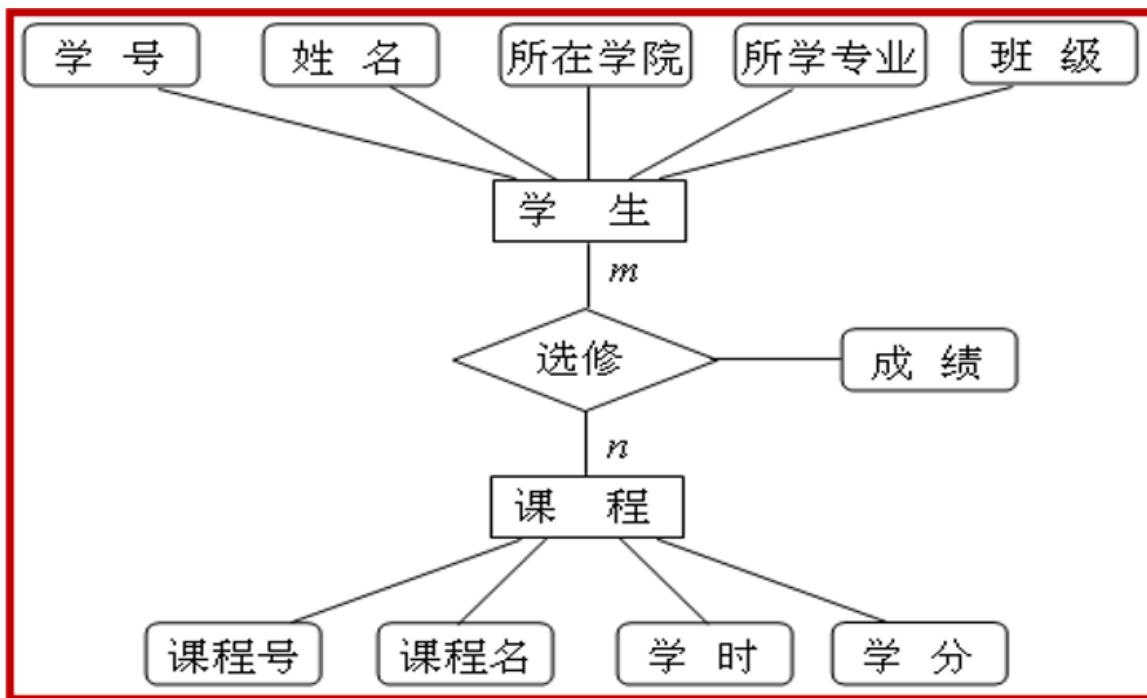
用实体-联系图 (E-R图) 描述现实世界的模型概念

实体和联系相连的弧线上注明联系的类型 (1:1, 1:n或m:n)



例：学生选课系统包含学生和课程两个实体，学生用学号、姓名、所在学院、所学专业、班级这几个属性描述，课程包含属性：课程号、课程名、学时、学分。

学生实体和课程实体间的联系为：一名学生可以选修若干门课程，一门课程可以有多名学生选修，学生选修了某门课程就应该有成绩



## 数据模型——关系数据库

### 数据模型的三个要素：

1. 数据结构
2. 数据操作
3. 数据完整性约束

### 数据模型的类型：

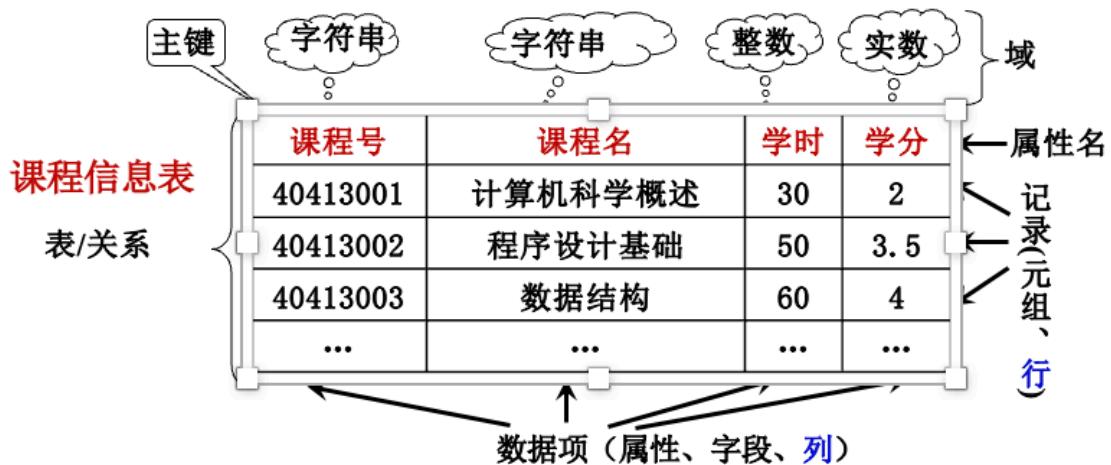
- 层次模型
- 网状模型
- 关系模型

## 关系模型

关系模型把实体及实体间的联系看成关系，以二维表的形式描述关系。

每个实体对应表中的行，称为记录。列对应实体的属性，称为数据项。

唯一标识一个记录的属性/属性集称为**主键**

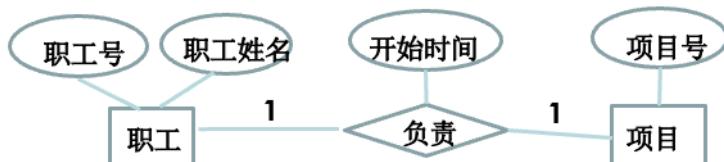


## 将E-R图转换为关系模型

实体间的联系分为以下三种：

- 1:1型

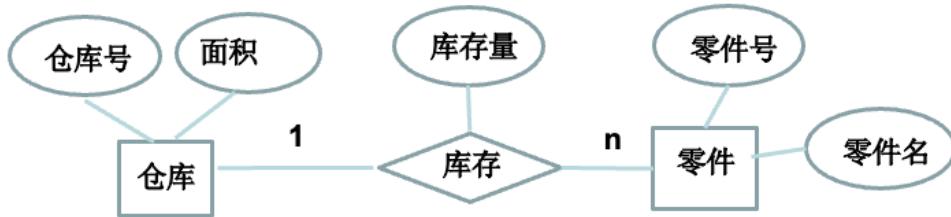
可以与任意一端对应的关系表合并：在一端属性集中加入另一端的主码和联系本身的属性。



职工表（职工号, 职工姓名, 项目号, 开始时间）  
项目表（项目号）

- 1:n型

在n端实体对应的表中，新增1端实体所对应表的主键属性和联系本身的属性。



仓库表（仓库号, 面积）

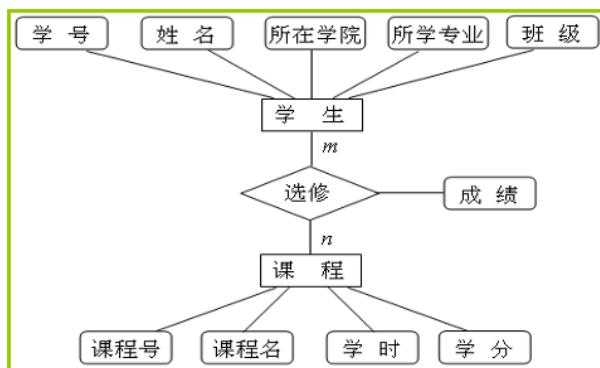
零件表（零件号, 零件名, 仓库号, 库存量）

- 为什么是1:n?

在现实中，一种零件存放在一个仓库，而一个仓库可以存放多种零件。

- m:n型

引入一个新表，两端实体表的主键和联系本身的属性组成新表的属性。新表的主键由联系两端实体表的主键组成。



学生信息表（学号, 姓名, 所在学院, 所学专业, 班级）

课程信息表（课程号, 课程名, 学时, 学分）

学生成绩表（学号, 课程号, 成绩）

## 模式

数据库中全体数据的逻辑结构和特征的描述，所有用户的公共视图。一个数据库只有一个模式

## 关系子模式

- 一个数据库可以有多个外模式
- 外模式就是用户视图
- 外模式保障了数据安全性

内模式：数据在数据库内部的表示方式。例如，储存方式有顺序存储，树结构存储等。一个数据库只有一个内模式

## 完整性约束

1. 实体完整性规则——元组的主键值不能为空或重复
2. 参照完整性规则——不允许引用不存在的实体

外键：一个表的外键用来指向另一个表的主键，包含外键的表为从表，外键用于将两个表连接在一起

3. 用户定义完整性规则