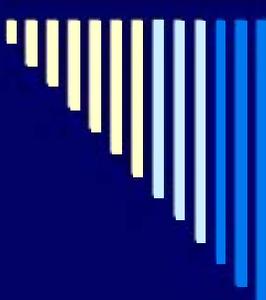


系统设计

- 系统设计系统总体结构设计
- 代码设计
- 数据结构与数据库设计
- 输入输出设计
- 模块功能与处理过程
- 系统设计报告

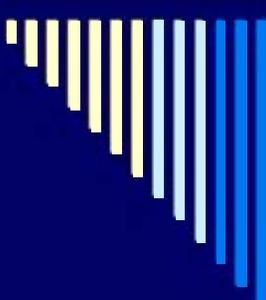




系统设计

- 在系统分析报告的基础上，进行新系统的设计，“怎么做”
- 内容
 - 总体结构设计：系统功能的实现
 - 物理模型设计：系统的实施方案
- 目标：设计出一个易于理解、维护和扩充的系统
- 方法：
 - 自顶向下的结构化设计
 - 原型法

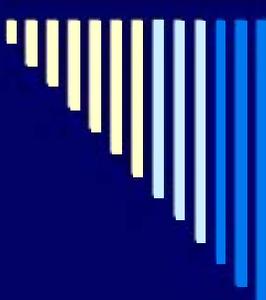




1、系统总体结构设计

- 将系统划分成模块；
- 决定每个模块的功能
- 决定模块的调用关系
- 决定模块的界面，即模块间信息的传递





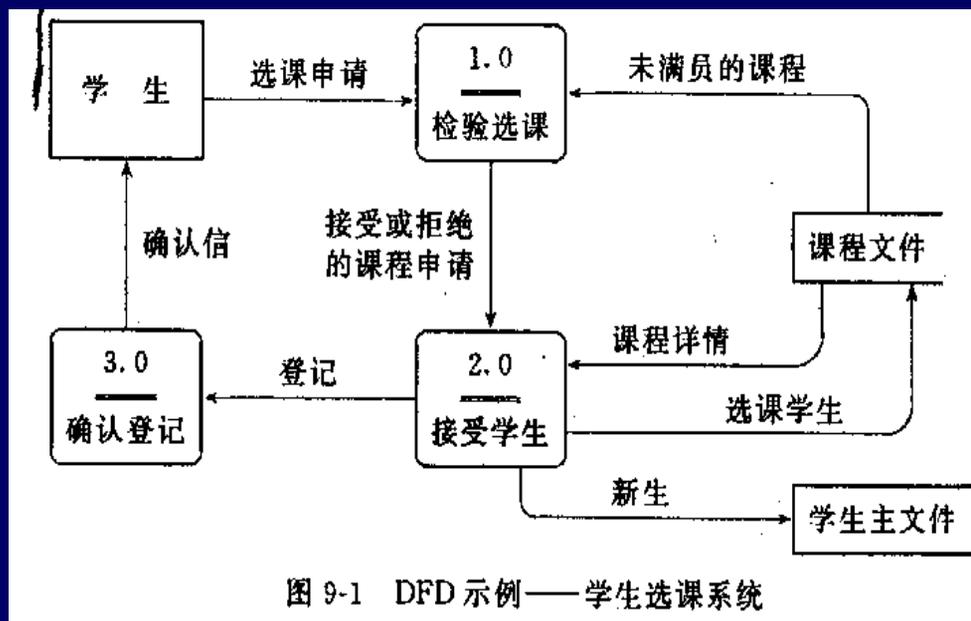
系统划分方法

- 按业务处理的功能划分
- 按业务处理的顺序划分
- 根据系统分析的结果划分

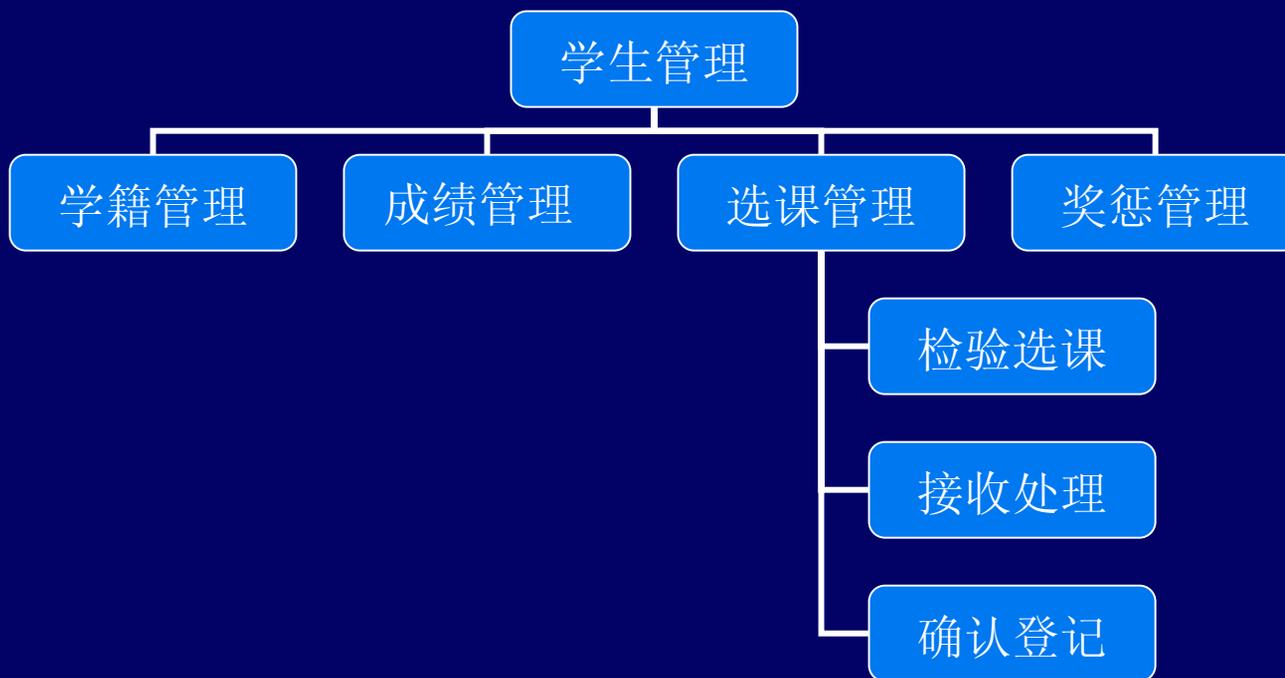


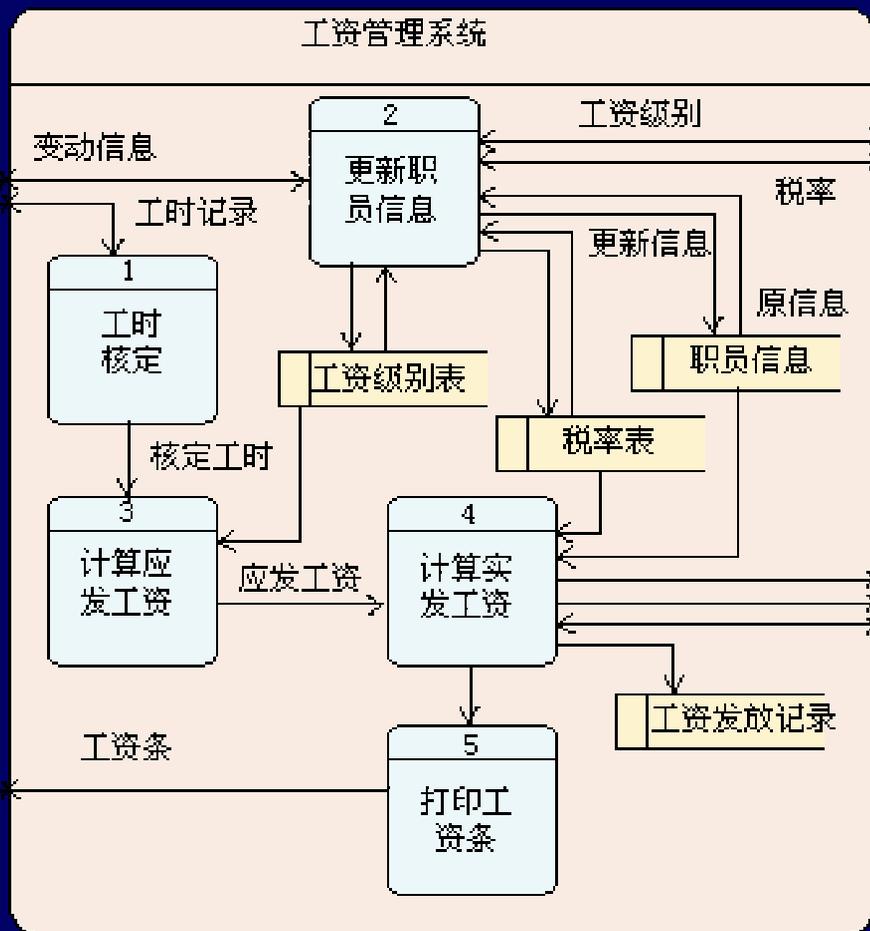
模块划分的例子

- 比如学生管理系统的分析中对选课处理的数据流程图DFD如下

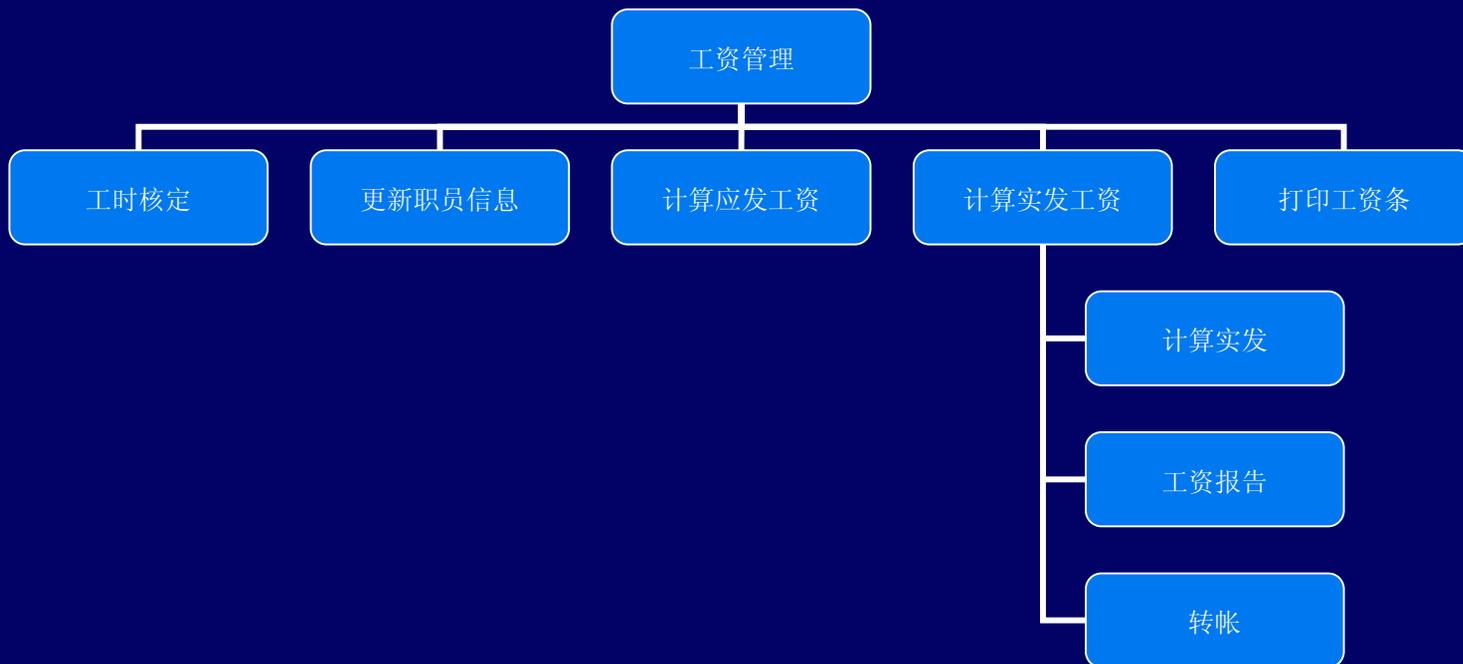


模块划分的例子

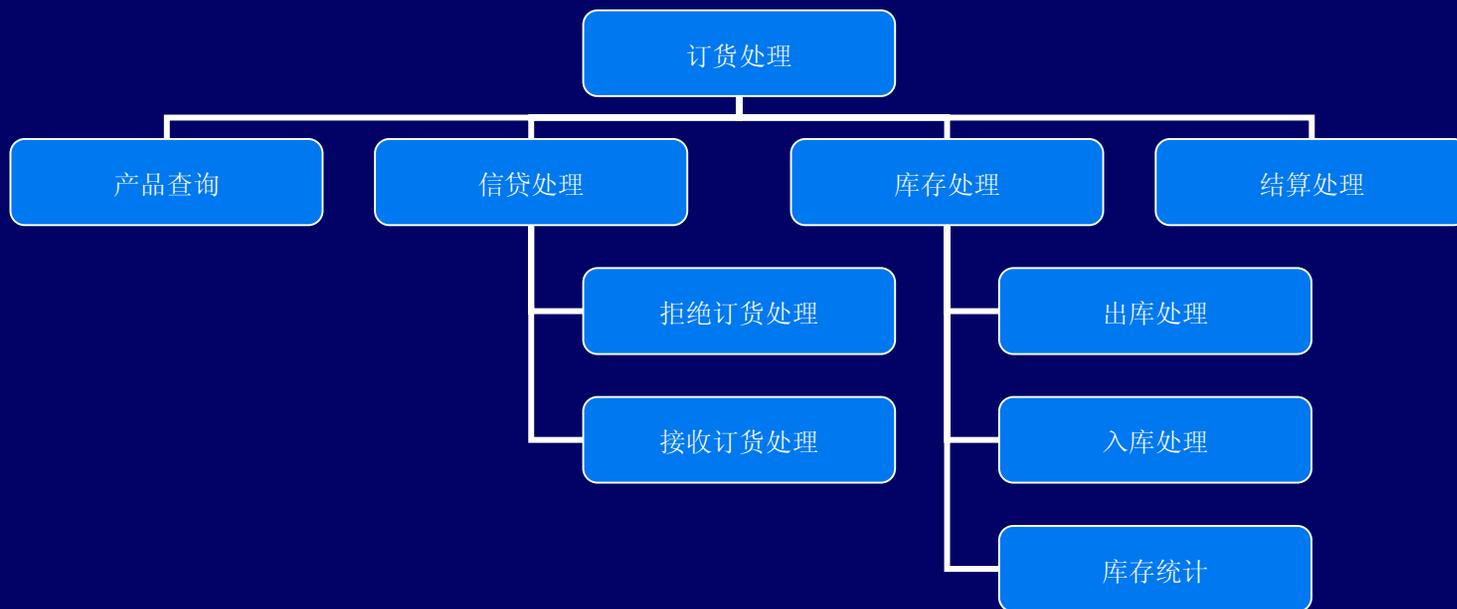


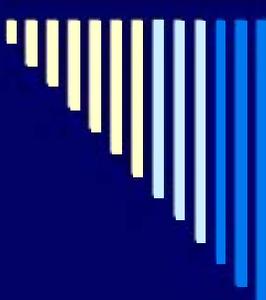


工资管理



订货业务处理

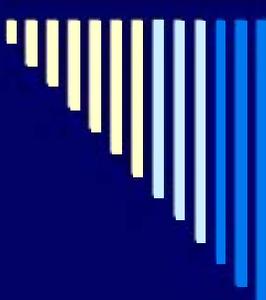




硬件与网络环境设计

- 网络设计
 - 网络结构
 - 网络协议
 - 网络软件
- 设备和网络配置
- 计算机处理流程设计

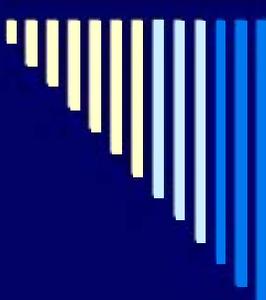




2 代码设计

- 易于计算机和人识别和处理
- 代码的主要功能
 - 鉴别
 - 分类
 - 排序
- 代码的类型
 - 顺序
 - 数字
 - 字符
 - 混合

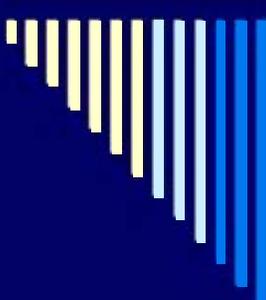




代码设计的原则

- 唯一
- 可扩充
- 简单
- 适用
- 规范
- 示例：学号 yyymmddnnn
 - yyyy：入学年； mm：学院编号
 - dd：专业； nnn：顺序号

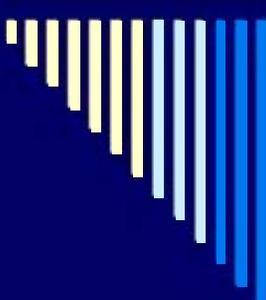




3 数据结构与数据库设计

- 根据数据的用途、使用、统计、保密等要求，决定数据的结构
- 以E-R图为基础
- 由数据库管理系统提供建立DB的方法
 - 数据定义语言
 - 交互式

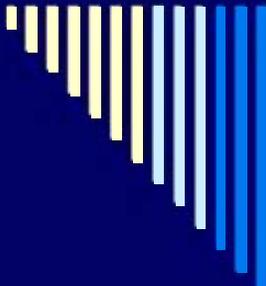




计算机中数据的组织形式

- 位
- 字节（字符）
- 字段（字段名，类型，宽度，小数位数）
- 记录
- 表
- 数据库



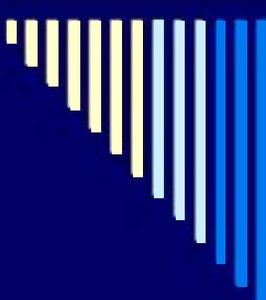


E-R (Entity-Relationship)

实体-联系方法

- 表示信息世界中的实体, 属性和关系的模型
- 实体: 代表现实世界的事物. 比如学生, 零件, 合同等. 用方框代表
- 属性: 反映事物的特征. 比如学生的属性包括学号, 姓名, 性别, 所属院系, 专业, 班级, 入学年等. 用椭圆框表示
- 关系: 反映实体之间的联系, 比如学生与课程, 学生与教师, 教师与课程的关系. 用菱形框表示



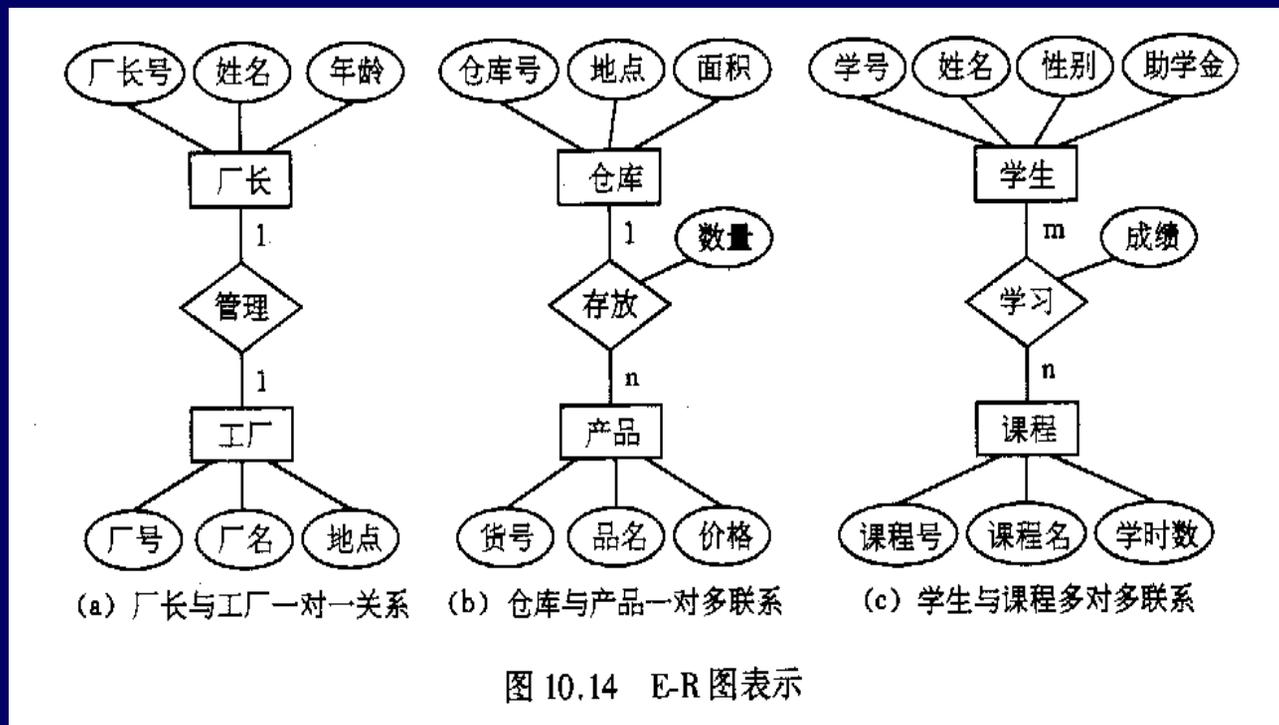


实体之间的联系方式

- 一对一 (1:1)
- 一对多 (1:n) 或多对一 (n:1)
- 多对多 (N:m)



实体关系的例子

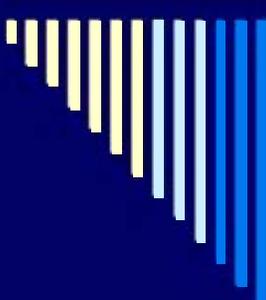


客观世界, 信息世界和数据库

表 10-7 三个不同世界术语对照表

| 客观世界 | 信息世界 | 数据世界 |
|------------|--------|-----------|
| 组织(事物及其联系) | 实体及其联系 | 数据库(概念模型) |
| 事物类(总体) | 实体集 | 文件 |
| 事物(对象,个体) | 实体 | 记录 |
| 特征(性质) | 属性 | 数据项 |

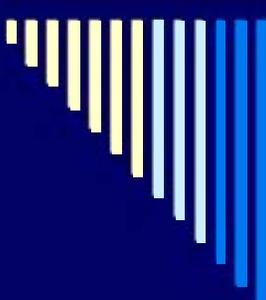




从E-R图导出关系数据模型

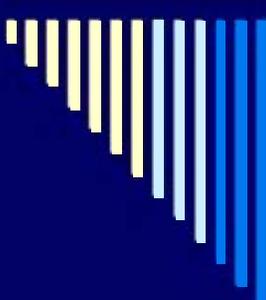
- 将实体转换成一个关系，为每个关系设定关键字，关键字具备唯一性，比如工厂实体的关键字是“厂号”，产品实体的关键字是“货号”
- 根据联系方式的不同，建立不同的联系方式





1: 1关系

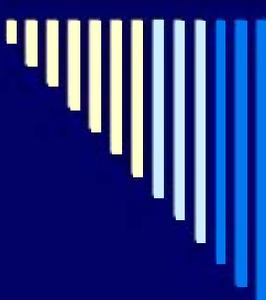
- 在一个实体中添加另一个实体的关键字，比如在厂长实体中添加“厂号”属性，或者在工厂添加“厂长号”属性。
- 厂长（厂长号，厂号，姓名，年龄）
- 工厂（厂号，厂名，地点）



1: n关系

- 将“1”方的关键字加入对应到“n”方实体的属性中，作为外部关键字
- 仓库（仓库号，地点，面积）
- 产品（货号，品名，价格，数量，仓库号）

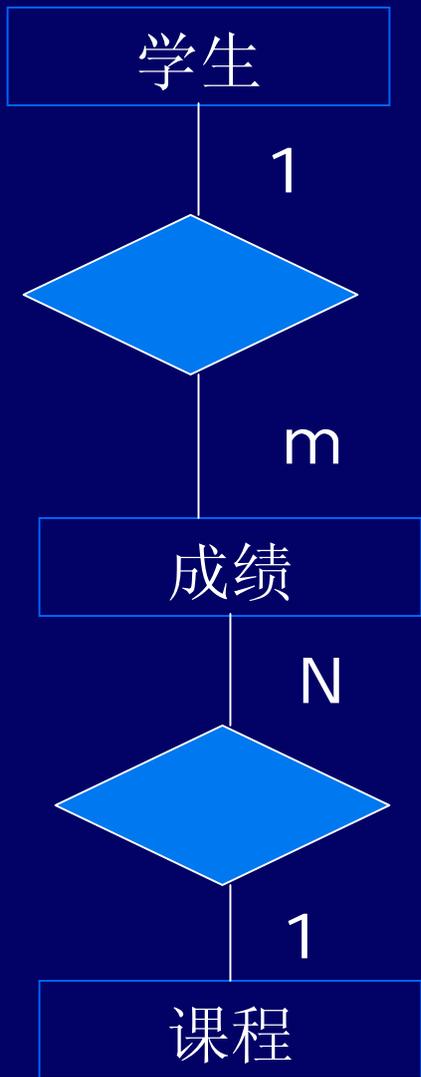


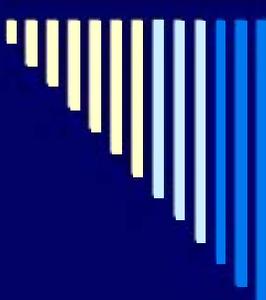


M: n的关系

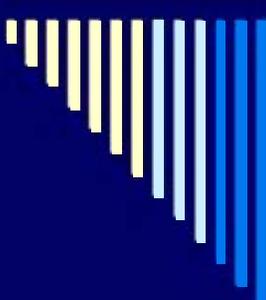
- 增加一个实体，将两个实体中的关键字作为新的实体的关键字，以及相关属性
- 学生（学号，姓名，性别，助学金）
- 课程（课程号，课程名，学时数）
- 成绩（学号，课程号，成绩）





- 
- 数据的规范化处理
 - 三个范式
 - 数据资源的分布
 - 数据的安全和保密

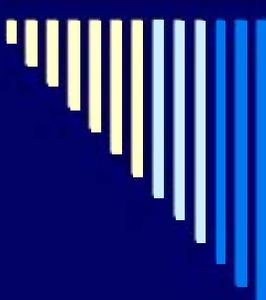




4、输入输出设计

- 输入方式
 - 键盘
 - 数模转换
 - 数据的导入
 - 输入格式
 - 校对的方法

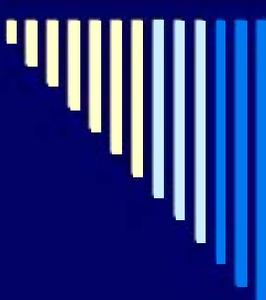




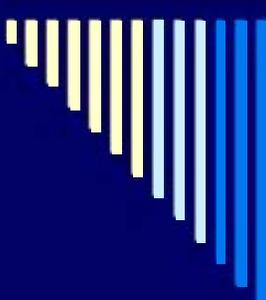
□ 用户界面

- 菜单
- 对话
- 提示
- 权限



- 
- 输出方式
 - 报表的设计
 - 图形的输出

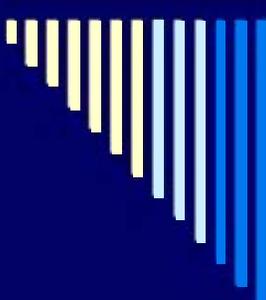




5、模块功能与处理过程

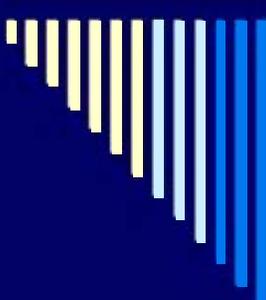
- 详细设计
- 设计模块的功能
 - 处理过程
 - 输入的数据
 - 输出的数据
- 模块之间的联结方式





□ 主要方法

- HIP0图 (Hierarchy plus Input-Process-Output)
 - 层次模块结构图描述模块之间的联接方式
- IPO图 (Input-Process-Output)
 - 描述每个模块内部的输入部分，处理描述，输出部分



IOP图

| 编号 | 标题：拒绝订货 | 制作 |
|----------------|---|-----------------|
| 输入 | 处理 | 输出 |
| 客户付款信息 订货数据 | 根据订货数据中的客户号查找客户的付款记录 如果存在不良记录，发拒绝订货信息给客户 | 客户付款信息 拒绝订货单 |



HIPO图例子

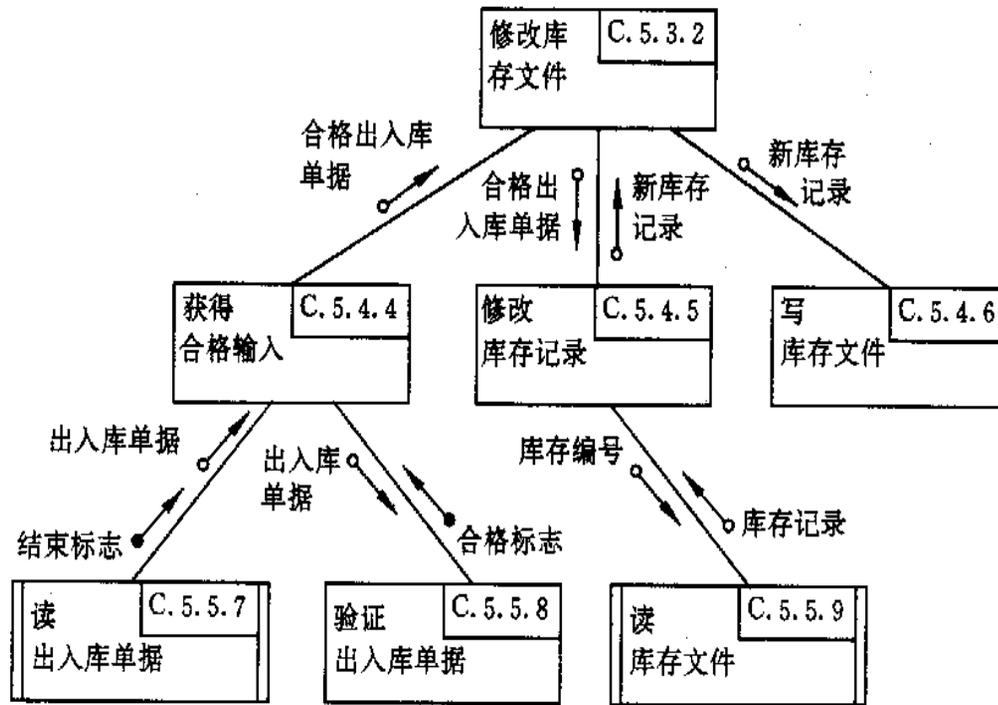


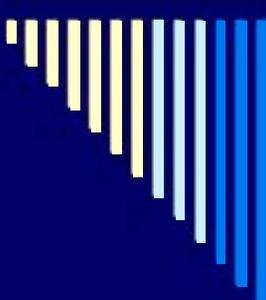
图 19.13 层次化模块结构图

IPO图例子

| IPO 图编号(即模块号):C.5.5.8 | | HIPO 图编号:C.5.0.0 | |
|---|---|---|---------------|
| 数据库设计文件编号:C.3.2.2,C.3.2.3 | | 编码文件号:C.2.3 | 编程要求文件号:C.1.1 |
| 模块名称:××××× | 设计者:××× | 使用单位:××× | 编程要求:COBOL,C |
| 输入部分(I) | 处理描述(P) | 输出部分(O) | |
| <ul style="list-style-type: none"> · 上组模块送入单据数据 · 读单据存根文件 · 读价格文件 · 读用户记录文件 · · | <ul style="list-style-type: none"> ① 核对单据与单据存根记录 ② 计算并核实价格 ③ 检查用户记录和信贷情况 · · <div style="margin-left: 40px;"> <p>处理过程—①—OK—</p> <ul style="list-style-type: none"> ②—OK— <ul style="list-style-type: none"> 出错信息(记录不合格) 价格不对处理 ③—OK— <ul style="list-style-type: none"> 用户信贷记录不好处理 记录合格 </div> | <ul style="list-style-type: none"> · 将合理标志送回上一级调用模块 · 将检查的记录记入×××文件 · 修改用户记录文件 · · | |

图 19.14 IPO 图





处理流程描述

- 处理流程描述方法
 - 结构化英语
 - 流程图
 - NS图
- 处理流程类型
 - 顺序
 - 选择
 - 循环



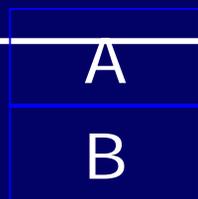
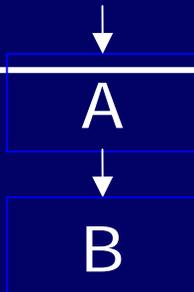
结构语言

流程图

NS图

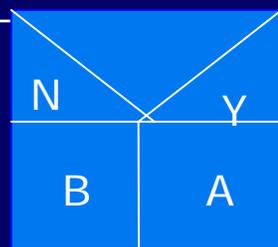
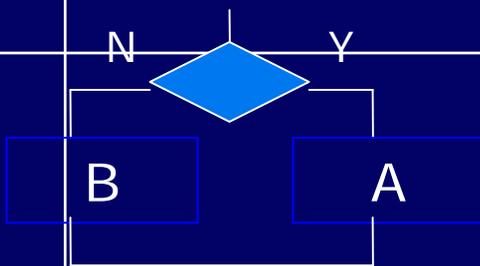
顺序

处理A
处理B



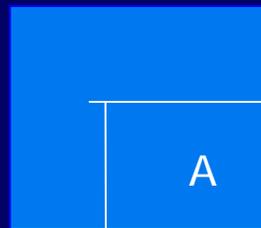
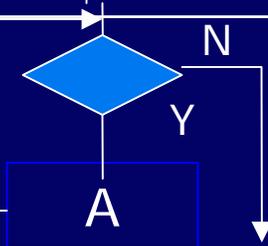
选择

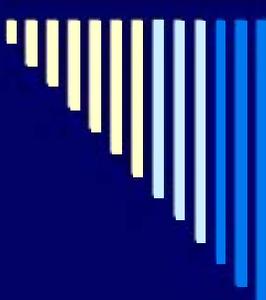
IF 条件成立
处理A
ELSE
处理B



循环

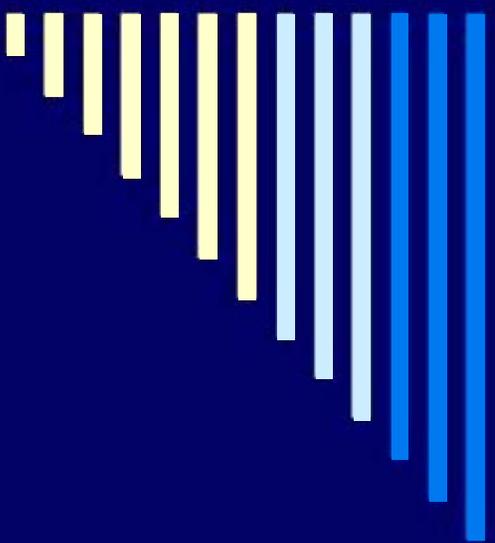
WHILE 条件成立
处理A





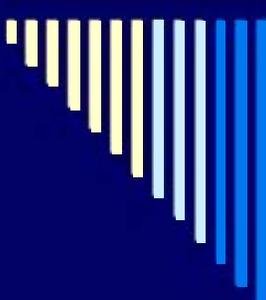
6、系统设计报告





系统实施、评价和运行

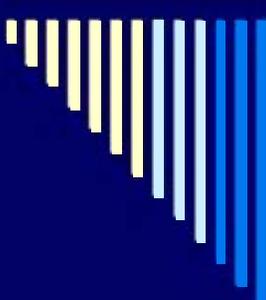
将设计模型转换成可实际运行的系统



系统实施的主要内容

- 硬件准备（主机、I/O、存储、辅助设备）
- 网络环境
- 软件准备
 - 系统软件
 - 数据库
- 编写程序
- 人员培训
- 数据准备

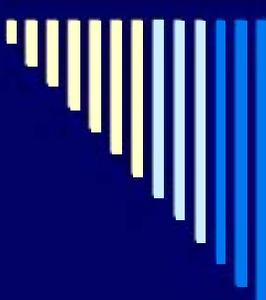




系统测试

- 模块
- 联合
- 验收
- 系统
- 目的是发现错误





作业一

■ 画出出版社的相关数据实体的E-R图

- 图书
- 编辑
- 作者
- 客户

要求:

标注出每个实体的属性, 关键字

画出实体之间的关系

如果两个实体是 $m:n$ 的关系, 增加一个实体, 变成 $1:m$ 和 $n:1$ 的关系。

