

个人项目报告：概念设计

● 引言

■ E-R 法概述

E-R 法（Entity-Relation Approach，实体联系方法）也称为 E-R 模型；

E-R 法的思想是用 E-R 图描述现实世界的信息，这种信息结构称为概念结构，然后根据具体系统的要求将概念结构转换成特定系统所能接受的逻辑结构。

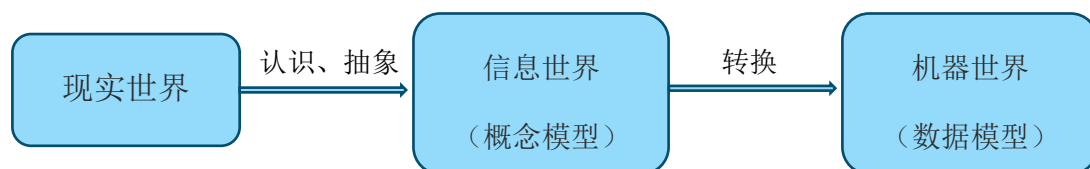
E-R 法由两部分组成：

1. 用 E-R 图描述现实世界
2. 将 E-R 图转换成相应的数据模型

E-R 模型中，其“联系”能够对不同实体关系进行描述。实践中，首先需要通过局部结构随机确定 2 个实体类型；其次是按照需求分析结果，判断实体类型间有无关系。如果有关系，那么必须明确是何种关系，比如 1:N，M:N，1:1 等。与此同时，还需判断各实体类型内部有无关系，实体类型间有无关系，众多实体间有无关系等。当关系类型明确时，不仅要避免发生冗余关系，而且要进行命名、设置主键。

■ 概念模型的重要性

概念模型用于信息世界建模，是现实世界到信息世界的抽象，是用户和数据库设计人员进行交流的语言。



在完成概念设计工作的时候，核心是对本项目涉及的各种对象进行抽象处理，得到整个系统的概念模型。在具体的项目中，应当把真实世界客观对象抽象处理，从而准确地描述企业组织的信息。建立对应的概念结构后，接下来的工作就是将其转化为对应的物理结构。对于本项目所使用的数据库系统来说，因为逻辑结构、概念模型之间互不影响，两者之间彼此独立、彼此支持，概念模型能够支持数据库 DBMS。概念模型既能够真实充分地反映现实世界中事物与事物之间的联系，又能够清晰地表达概念设计结果。概念模型，简洁明确独立于机器，易于用户接受理解，便于应用人员和数据库人员进行相互交流。同时概念模型容易向层次模型、关系模型以及网状模型数据转变，概念模型有效实现了现实世界到机器世界的过渡转变。

● 实体

■ 定义

实体是现实世界中可区别于所有其他对象的一个“事物”。每个实体有一组性质，其中一些性质的值可以唯一地标识一个实体。实体可以是实实在在的，如人或书；也可以是抽象的，如课程或机票预订等。

实体通过一组属性来表示。属性是实体集中每个成员所拥有的描述性性质。为某实体集指定一个属性表明数据库为该实体集中每个实体存储相似的信息；但每个实体在每个属性上都有各自的值。每个实体的每个属性都有一个值。

实体集是相同类型即具有相同性质（或属性）的一个实体集合。例如，实体集`student`可以表示大学中所有学生的集合。

■ 学生宿舍信息管理系统实体集设计

当我们使用 E-R 模型设计数据库时，通常从确定那些应当包含的实体集开始。当决定好实体集后，我们必须挑选恰当的属性，这些属性要表示我们在数据库中所捕获的不同的值。一旦选择好实体和它们相应的属性，不同实体间的联系集就建立起来了。

一个好的实体-联系设计不包含冗余的属性。对于我们学生宿舍信息管理系统的例子，在下面列出实体集以及他们的属性，主码以下划线表明。

1. student: 包含属性（Sno、Sname、Sgender、Sdept）
2. dorm: 包含属性（Dno、Dphone）
3. fee: 包含属性（Fcost、Fleftfee、Fctfee）
4. visitor: 包含属性（Vname、Vgender、Vin、Vout）
5. manager: 包含属性（Mno、Mpassword）
6. possession: 包含属性（Pno、Pname）

● 实体之间的联系

■ 定义

联系是指多个实体间的相互关联。同一个实体集内部各实体之间也可以存在一对一、一对多、多对多的联系（如图 1）。三个或多个实体型间可能具有联系（如图 2）、两个实体型之间可具有多种联系（如图 3）。

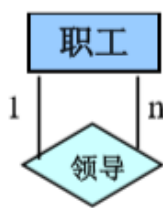


图 1

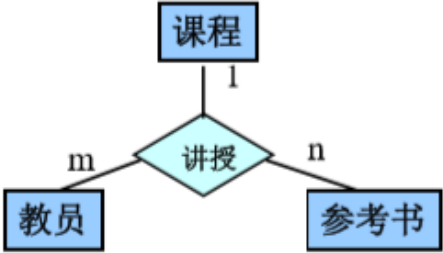


图 2

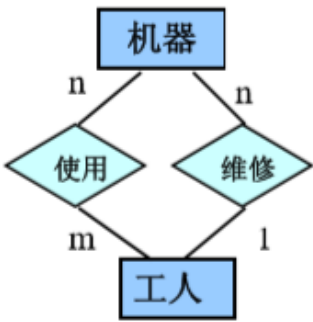


图 3

联系集是相同类型联系的集合。正规地说，联系集是 $n \geq 2$ 个（可能相同的）实体集上的数学关系。如果 E_1, E_2, \dots, E_n 为实体集，那么联系集 R 是

$$\{(e_1, e_2, \dots, e_n) | e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}$$

的一个子集，而 (e_1, e_2, \dots, e_n) 是一个联系。

■ 学生宿舍信息管理系统联系集设计

我设计的联系集如下。checkin、manage、visit、repair、quit、leave、check。

- 1. checkin: 关联学生和宿舍
- 2. manage: 关联宿舍管理员和宿舍
- 3. visit: 关联学生和来访人员
- 4. repair: 关联学生和宿舍财产
- 5. quit: 关联学生和宿舍
- 6. leave: 关联学生和宿舍
- 7. check: 关联宿舍管理员和宿舍

● 概念结构设计方法

设计概念结构通常有四类方法：

- 1. 自顶向下：首先定义全局概念结构的框架，然后逐步细化
- 2. 自底向上：首先定义各局部应用的概念结构，然后将其集成起来
- 3. 逐步扩张：首先定义核心概念结构，然后向外扩充
- 4. 混合策略：自顶向下设计全局概念框架，自底向上设计各局部概念结构

其中最常用的方法是自底向上方法。

■ 基于 E-R 法的概念模型设计

基于 E-R 法的概念模式自底向上的设计方法分为两个阶段：

1. 数据抽象与局部 E-R 图设计
2. 综合局部 E-R 图形成总 E-R 图

■ 数据抽象

局部 E-R 图设计的基本思想是利用数据抽象机制对需求分析阶段收集的数据进行分类、组织，形成实体模型并确定实体之间的联系类型，设计局部 E-R 图。

数据抽象机制包括：

1. 分类：定义某一概念作为现实世界中一组对象的类型。这些对象具有某些共同的特性和行为，如实体型
2. 聚集：定义某一类型的组成成分，如属性的聚集组成了实体型
3. 概括：定义类型之间的一种子集联系

■ 局部 E-R 图设计

局部 E-R 图具体设计方法：

1. 选择局部应用
2. 以需求分析中得到的数据元素表为基础，建立实体模型
3. 确定实体之间的联系类型。用 E-R 图表示这些实体与实体之间的联系，形成局部 E-R 图

■ 综合局部 E-R 图形成总 E-R 图

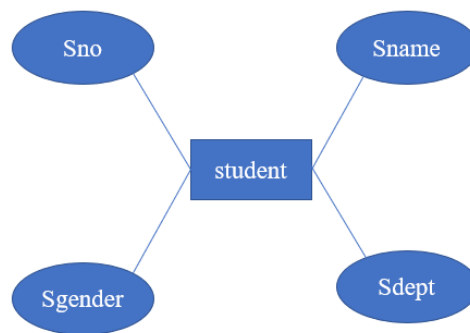
设计总 E-R 图（全局概念模式）可以有两种方法：

1. 多个局部 E-R 图一次集成
2. 逐步集成，用累加的方式一次集成两个局部 E-R 图

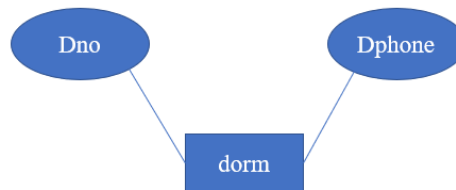
● 实体属性局部 E-R 图

根据需求分析，总结出该学生宿舍信息管理系统中的实体包括：student、dorm、fee、visitor、manager、possession。

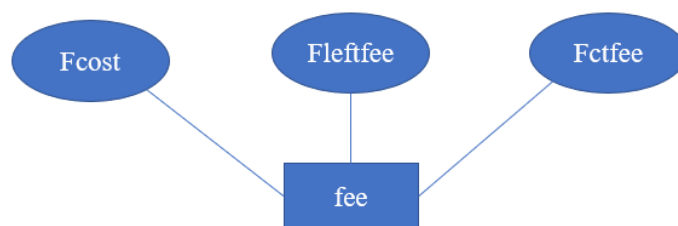
■ student 实体局部 E-R 图



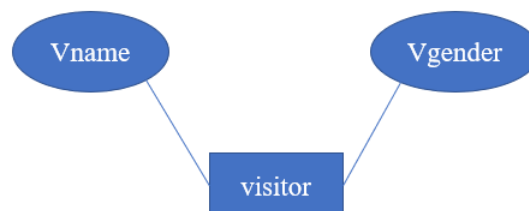
■ dorm 实体局部 E-R 图



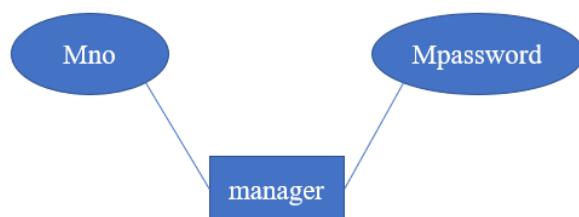
■ fee 实体局部 E-R 图



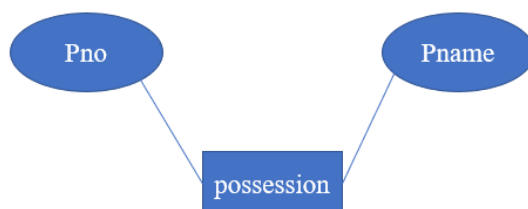
■ visitor 实体局部 E-R 图



■ manager 实体局部 E-R 图



■ possession 实体局部 E-R 图

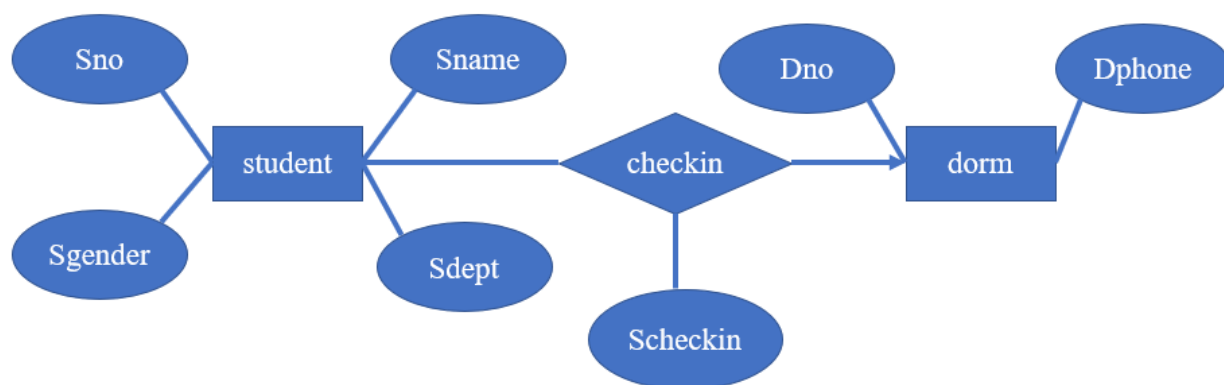


● 实体联系局部 E-R 图

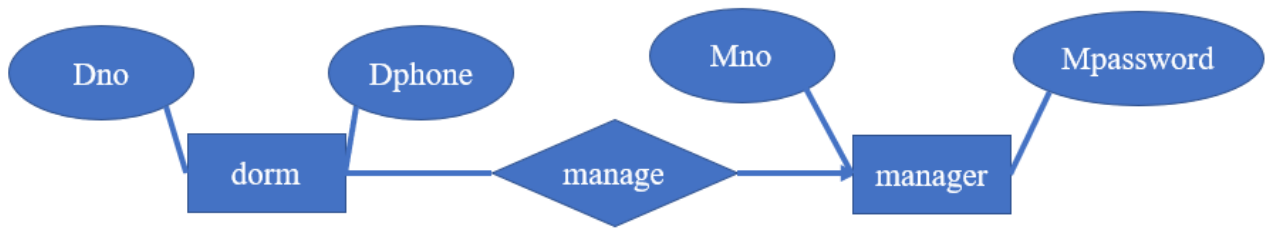
根据需求分析，总结出该学生宿舍信息管理系统中的联系包括：checkin、manage、visit、repair、quit、leave、check。

在学生宿舍信息管理系统中，数据库设计必须重点考虑如下实体：student、manager、dorm 等，通过上述实体间的关系设计不难发现，关系比较复杂，包括一对一、一对多、多对多等关系。例如一个学生一次只能住进一间宿舍，而一间宿舍可以住进多个学生，所以，这两者之间就是一种一对多的关系。在数据库设计过程中，这些一般通过 E-R 图进行描述，详情可参照下图继续了解：

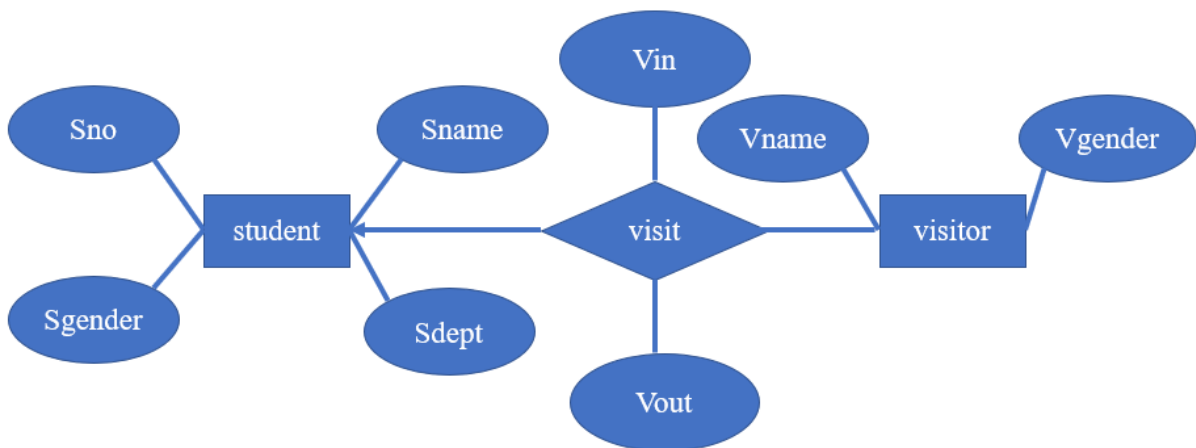
■ checkin 联系局部 E-R 图



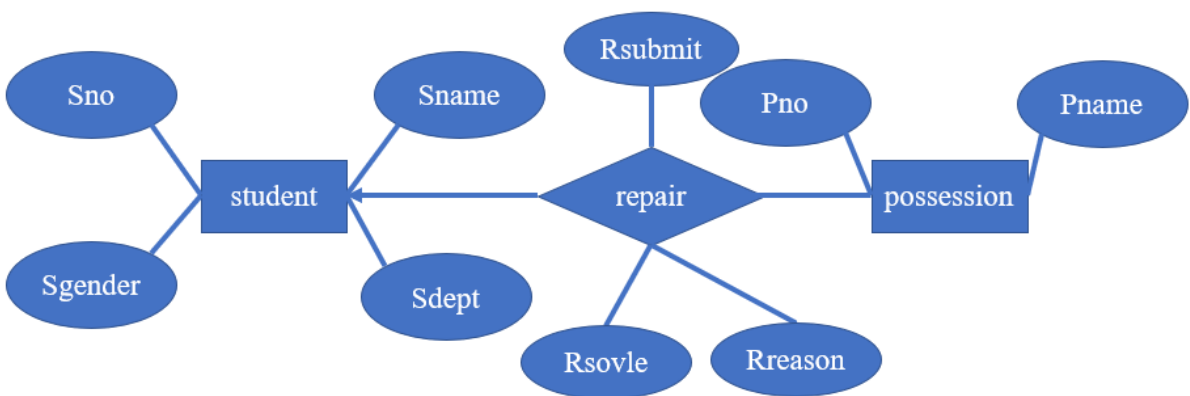
■ manage 联系局部 E-R 图



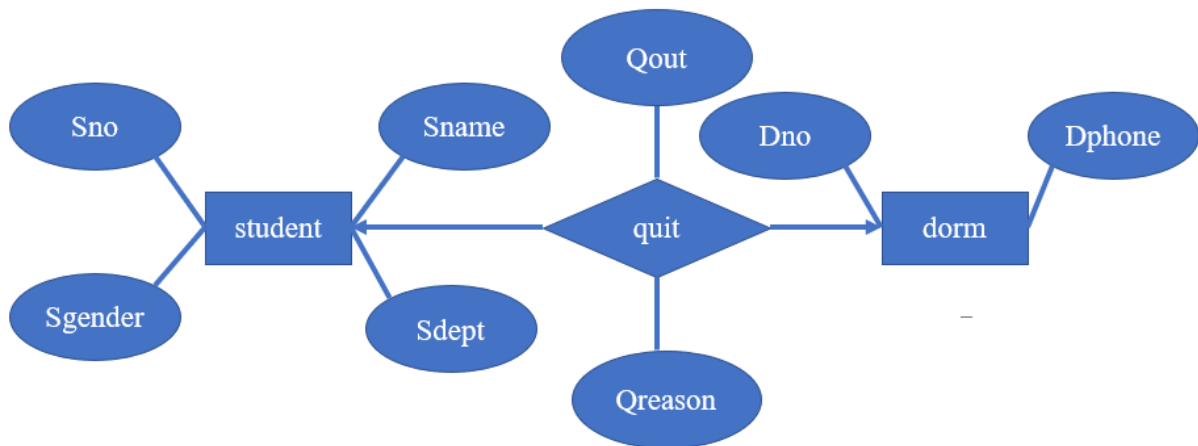
■ visit 联系局部 E-R 图



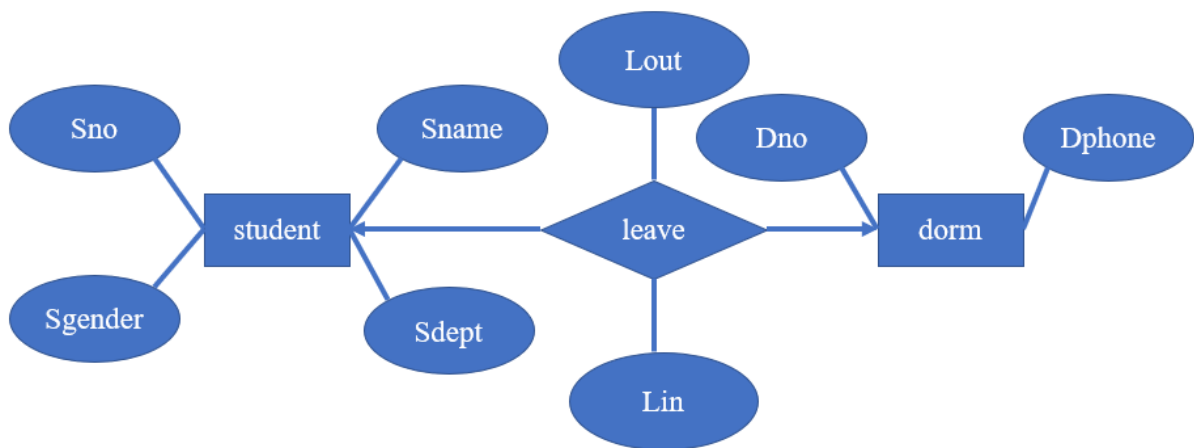
■ repair 联系局部 E-R 图



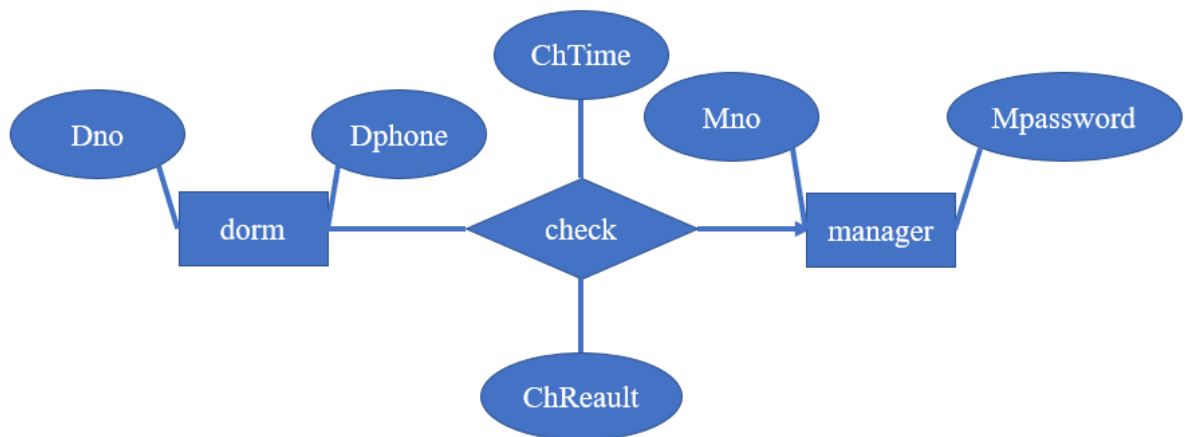
■ quit 联系局部 E-R 图



■ leave 联系局部 E-R 图



■ check 联系局部 E-R 图



● 全局 E-R 图

