

考情分析



数据库设计的任务是针对一个给定的应用环境，在给定的（或选择的）硬件环境和操作系统及数据库管理系统等软件环境下，创建一个性能良好的数据库模式，建立数据库及其应用系统，使之能有效地存储和管理数据，满足各类用户的需求。

数据库设计工作量大而且过程比较复杂，是一项数据库工程也是一项庞大的软件工程。数据库设计包括结构特性的设计和行为特性的设计两方面的内容。结构特性的设计是指确定数据库的数据模型。数据模型反映了现实世界的数据及数据间的联系，要求在满足应用需求的前提下，尽可能减少冗余，实现数据共享。行为特征的设计是指确定数据库应用的行为和动作，应用的行为体现在应用程序中，所以行为特性的设计主要是应用程序的设计。

3.1 考情分析

随着数据库系统的发展，文件系统已经慢慢被淘汰，因此，从历年试题来看，在软件设计师的下午考试的试题中，一定会出现数据库设计的试题。数据库设计的考点和上午试题的考点基本重叠，只是考查的方式不同。主要考点包括数据的规范化、E-R模型、数据库的逻辑设计与物理设计、SQL语句等内容。

版权方授权希赛网发布，侵权必究

上一节 本书简介 下一节

考试大纲要求分析

根据考试大纲，数据库设计要求考生掌握以下知识点：

（1）理解和掌握数据库管理系统的功能和特征。

关系模式。

第一范式、第二范式、第三范式、BC范式。

（4）掌握数据库设计过程，能进行数据库逻辑设计，重点掌握E-R模型向关系模型转换的方法。

（5）掌握给定一个实际的应用问题如何设计E-R模型，如何将E-R模型转换成关系模式，确定联系类型、主键、候选键、外键，判断关系模式规范化的程度。

（6）掌握数据操作，包括集合运算和关系运算。

（7）掌握数据库语言，即SQL语言。

（8）掌握给定一个实际的应用问题如何用SQL进行数据定义（创建表、视图、索引、完整性定义及权限定义。

考情分析



## 命题特点与趋势分析

关于数据库设计的考题通常出现在软件设计师考试下午科目的第2题，所占分值为15分，约占试卷总分值75分中的20%。

数据库设计本身的特点使得考查的题型和考查的内容都比较集中。其中，数据库设计高频考点为：

- (1) E-R图的画法。
- (2) 关系模式的设计。
- (3) 数据库概念结构设计和逻辑结构设计。
- (4) SQL语言及索引相关知识。

考查形式一般是根据有关的实体联系图和需求分析的结果，按照要求回答问题或填空。试题内容涉及到E-R图的画法、关系模式设计、数据库概念结构设计及逻辑结构设计、函数依赖与范式和实体间联系等多个方面，要求考生熟练地掌握数据库设计的常用方法和技术。

大多数试题侧重于实践应用，检验考生是否理解相关的知识点。对于基本概念的考生一般结合在题目中，有时也会针对这些基本概念出题，比如有的题目要求说明弱实体、索引等基本概念。

鉴于软件设计师考试采用模块化的命题风格，因此在今后考试中有关数据库设计的试题的题型、所占分值都不会有太大的变化，考查的内容将仍然会集中在以上高频考点。建议考生一定要在这几方面的知识点多下功夫，尽可能做到熟能生巧，以便考试时能灵活应变，节约在这些知识点上的解题思考时间，争取多拿分，拿满分。

## 考点精讲

数据库设计涉及的范围很广，要设计一个性能良好的数据库并非易事。从本质上讲，数据库设计的过程是将数据库系统与现实世界密切、有机、协调一致地结合起来的过程。数据库的设计质量与设计者的知识、经验和水平密切相关。作为数据库应用系统的重要组成部分，数据库设计的成败往往直接关系到整个应用系统的成败。

数据库设计属于系统设计的范畴。通常把使用数据库的系统统称为数据库应用系统，把对数据库应用系统的设计简称为数据库设计。数据库设计的任务是针对一个给定的应用环境，在给定的（或选择的）硬件环境和操作系统及数据库管理系统等软件环境下，创建一个性能良好的数据库模式，建立数据库及其应用系统，使之能有效地存储和管理数据，满足各类用户的需求。

合理的数据库结构是数据库应用系统性能良好的基础和保证，但数据库的设计和开发却是一项庞大而复杂的工程。从事数据库设计的人员，不仅要具备数据库知识和数据库设计技术，还要有程序开发的实际经验，掌握软件工程的原理和方法。数据库设计人员必须深入应用环境，了解用户具体的专业业务。在数据库设计的前期和后期，与应用单位人员密切联系，共同开发，可大大提高数据库设计的成功率。

以数据库为基础的数据库应用系统与其他计算机应用系统相比往往具有数据量庞大、数据保存时间长、数据关联复杂、用户要求多样化等特点。

在数据库设计中面临的主要困难和问题有：

(1) 同时具备数据库知识与应用业务知识的人很少。懂计算机与数据库的人一般都缺乏应用业务知识和实际经验，而熟悉应用业务的人往往不懂计算机和数据库。

(2) 项目初期往往不能确定应用业务的数据库系统的目标。

(3) 缺乏完善的设计工具和设计方法。

(4) 需求的不确定性。用户总是在系统的开发过程中不断提出新的要求，甚至在数据库建立之后还会要求修改数据库结构或增加新的应用。

(5) 应用业务系统千差万别，很难找到一种适合所有业务的工具和方法，这就增加了研究数据库自动生成工具的难度。因此，研制适合一切应用业务的全自动数据库生成工具是不可能的。

数据库设计的很多阶段都可以和软件工程的各阶段对应起来，软件工程的某些方法和工具同样也适合于数据库工程，但数据库设计还有很多自己的特点。

(1) 从数据结构即数据模型开始，并以数据模型为核心展开。这是数据库设计的一个主要特点。

(2) 静态结构设计与动态行为设计分离。静态结构设计是指数据库的模式结构设计，包括概念结构、逻辑结构和物理结构的设计；动态行为设计是指应用程序设计，包括功能组织、流程控制等方面的设计。传统的软件工程往往注重处理过程的设计，不太注重数据结构的设计，在结构程序设计中只要可能就尽量推迟数据结构的设计，而数据库设计正好与之相反，需要把主要精力放在数据结构的设计上，如数据库的表结构、视图等。

(3) 试探性 (Tentative)：数据库设计比较复杂，又缺少完善的设计模型和统一的过程，设计的过程往往是试探性的过程，因此设计的结果往往不是惟一的。有时多种方案并存，供设计者选择，而且这种选择并不是完全客观的，多少取决于用户的偏爱和观点。

(4) 反复性 (Iterative)：由于数据库设计是一种试探性的过程，这就决定了数据库的设计是一个反复推敲和修改的过程，而不可能“一气呵成”。

(5) 多步性 (Multi-stage)：传统的数据库设计采用直观设计法或单步设计法，它由设计者通过对用户的调查访问，确认需求，熟悉用户应用问题的语义，结合结构限制与DBMS功能，凭设计人员的经验进行分析、选择、综合与抽象之后，建立数据模型，并用DDL写出模式。由于这种设计方法要求设计人员有比较丰富的经验和熟练的技巧，不易被一般人掌握，并且因其缺乏工程规范支持和科学根据，现已经抛弃不用。新的数据库设计是分步（阶段）进行的，前一阶段的设计结果可作为后一阶段设计的依据，后一阶段也可向前一阶段反馈其要求，反复修改，逐步完善。

## 概念设计

---

概念结构设计阶段所涉及的信息不依赖于任何实际实现时的环境即计算机的硬件和软件系统。概念结构设计的目标是产生一个用户易于理解的，反映系统信息需求的整体数据库概念结构。概念结构设计任务是，在需求分析中产生的需求说明书的基础上按照一定的方法抽象成满足应用需求的用户的信息结构，即通常所称的概念模型。概念结构的设计过程就是正确选择设计策略、设计方法和概念数据模型并加以实施的过程。

### 1. 概念结构

概念模型是现实世界到机器世界的一个过渡的中间层次，它有以下特点：

- (1) 概念模型有丰富的语义表达能力，能表达用户的各种需求，是对现实世界的抽象和概括，它真实、充分地反映了现实世界中事务与事务之间的联系，能满足用户对数据的处理要求。
- (2) 易于交流和理解。由于概念模型简洁、明晰、独立于机器，是数据库设计人员和用户之间的主要交流工具，因此可以用概念模型与不熟悉计算机的用户交换意见，使用户能积极参与数据库的设计工作，保证设计工作进行顺利。
- (3) 概念模型易于更改。当应用环境和应用要求发生变化时，能方便地对概念模型修改和扩充，以反映这些变化。
- (4) 概念模型很容易向关系、网状、层次等各种数据模型转换，易于导出与DBMS有关的逻辑模型。

概念数据模型的作用是：

- (1) 提供能够识别和理解系统要求的框架。
- (2) 为数据库提供一个说明性结构，作为设计数据库逻辑结构即逻辑模型的基础。

概念模型描述工具应该能够体现概念模型的特点，如E-R模型。近年来，由于面向对象数据模型具有更丰富的语义、更强的描述能力而越来越受到人们的重视，不但出现了商品化的面向对象DBMS，而且开始实际应用于概念模型的设计中，作为数据库概念设计的工具。Teory等人提出的扩展的E-R模型（称为E-E-R模型）增加了类似面向对象数据模型中的普遍化和聚合等语义描述机制，为这种最为人们熟悉的数据模型注入了新的生机，为概念模型描述增加了一种理想的选择。

概念结构的设计策略主要有自底向上、自顶向下、由里向外和混合策略。在具体实现设计目标时有两种极端的策略或方案，一是建立一个覆盖整个单位所有功能域的全局数据库，称之为全局方案或全局策略（Global Approach）；另一种则是对每一个应用都建立一个单独的数据库，称之为应用方案（Application Approach）或应用策略。

全局策略要求设计一个能够支持单位所有应用需求的单一数据库，此法在数据库应用早期曾受到广泛的推崇。但实践证明，对于一个较大的单位，采用此种策略设计数据库，会使设计任务十分复杂，开发周期过长，因而很难取得成功。应用策略则对每个应用单独设计一个数据库，优点是简化了分析工作且单库单用，运行效率较高，但也存在明显的缺点，会造成大量数据冗余、数据的不一致性、破坏整个单位中数据的完整性及难以实现数据共享。因此，这一策略与传统的文件系统并无本质上的差别，从而与全局策略一样可能导致失败。

介于全局和应用两种策略之间的一个折中的策略是根据对事务活动的分析，先设计出一个单位

的初步信息结构，该结构表示单位活动中涉及的主要实体、实体间的联系，然后根据对信息流的分析，按自然合理的分组原则将这些实体及实体间的联系分别组成若干便于管理的较小数据库，再进行每个数据库的详细设计。

折中策略在大多数情况下提供了一种最好的选择。需要注意的是，对初步信息结构中实体及实体联系的划分不应依赖于某个应用及职能部门的界限，可以跨越多个应用域或职能部门，否则就变成了应用策略了。此外，对于信息结构不太复杂的较小单位，也可以考虑选择全局策略。

## 2. 概念结构设计的方法和步骤

概念结构设计常用的方法是实体分析法（Entity Analysis）和属性综合法（Attribute Synthesis）。

实体分析法又称自顶向下（Top-Down）方法。它从总体概念入手，从分析一个单位的事务活动开始，首先识别用户所关心的实体及实体间的联系，建立一个初步的数据模型框架，然后再用逐步求精的方法加上必需的描述属性，形成一个个完整的局部数据模型，称为用户视图，最后再将这些视图集成为一个统一的数据模式，称为用户视图的集成，这种统一的数据模式（即全局信息结构）通常用E-R图表示。可见，实体分析法是通过3个不同的步骤完成概念模型设计的：

- （1）建立用户视图（局部信息结构）。
- （2）视图集成为统一的数据模型（全局信息结构）。
- （3）用E-R图描述全局信息结构。

这种方法的优点是：减少了分析中所涉及的对象数，简化了分析过程，且采用图形表示法，因而更为直观，易于理解，有利于用户在设计过程中的介入，所以被广泛采用。

属性综合法又称自底向上（Bottom-Up）方法。其基本点是将需求分析中收集的数据元素作为分析对象，高层实体及联系通过低层属性综合而成的设计技术。实际上，这是一种基于统计分析推导的方法，即通过对数据元素与应用任务联系的定性定量统计分析技术来推导出相应的信息结构。其处理过程可分为属性分类、实体构成、联系的确定等相对独立的步骤。这种方法适用于较为简单的设计对象而不适于复杂的应用环境，因为要对几百甚至数千个数据元素进行综合处理就不容易，何况在数据分析的前期阶段，数据库管理员通常并不完全清楚在数据库模式中究竟包含哪些数据元素。

## 3. 数据抽象和局部视图设计

数据模型是数据库系统的核心和基础，各种机器上实现的DBMS软件都是基于某种数据模型且都有一个共同的特点，因为它们是在具体的机器上实现的，所以在许多方面给出了细致严格的限制。而在现实世界中应用环境是复杂多变的，各种事务的表现形式也与机器世界中的相距甚远。在进行数据库设计时，如果将现实世界中的客观对象直接转换为机器世界中的对象，就会感到非常不方便，注意力往往被牵扯到更多的细节限制方面，而不能集中在最重要的信息的组织结构和处理模式上。因此往往是将现实世界中的客观对象首先抽象为不依赖任何具体机器的信息结构，这种信息结构不是DBMS支持的数据模型，而是概念级模型，然后再把概念级模型转换为具体机器上DBMS支持的数据模型。

由于各个部门对于数据的需求和处理方法各不相同，对同一类数据的观点也可能不一样，它们有自己的视图，所以可以首先根据需求分析阶段产生的各个部门的数据流图和数据字典中的相关数据设计出各自的局部视图，然后进行视图集成。

在实体分析法中，局部视图设计的第一步是确定其所属的范围，即它所对应的用户组，然后对

每个用户组建立一个仅由实体、联系及它们的标识码组成的局部信息结构（局部数据模式）框架，最后再加入有关的描述信息，形成完整的局部视图（局部数据模式）。这样做的目的是为了集中精力处理好用户数据需求的主要方面，避免因无关紧要的描述细节而影响局部信息结构的正确性。整个过程可分为确定局部视图的范围，识别实体及其标识，确定实体间的联系，分配实体及联系的属性等几个步骤。

#### （1）确定局部视图的范围

需求说明书中标明的用户视图范围可以作为确定局部视图范围的基本依据，但它通常与子模式范围相对应，有时因为过大而不利于局部信息结构的构造，故可根据情况修改，但也不宜分得过小，过小会造成局部视图的数量太大及大量的数据冗余和不一致性，给以后的视图集成带来很大的困难。

局部视图范围确定的基本原则是：

①各个局部视图支持的功能域之间的联系应最少。

②实体个数适量。一个局部视图所包含的实体数量反映了该局部视图的复杂性，按照信息论中“ $7 \pm 2$ ”的观点，人们在同一时刻可同时顾及的事情一般在5-9之间，以6或7最为适当。因此，一个局部视图内的实体数不宜超过9个，否则就会过于复杂，不便于人们理解和管理。

#### （2）识别实体及其标识

在需求分析中，人们已经初步地识别了各类实体、实体间的联系及描述其性质的数据元素，这些统称为数据对象，它们是进一步设计的基本素材。这一步的任务就是在确定的局部视图范围内，识别哪些数据对象作为局部视图的基本实体及其标识，并定义有关数据对象在E-R模型中的地位。

##### ①数据对象的分类

为了确定数据对象在局部E-R数据模型中的地位，首先必须对所有已识别的数据对象加以适当的分类，以便于根据它们所属的对象类来确定它们在相应局部E-R模型中的身份。数据对象分类的原则是：同一类中的对象在概念上应该具有共性。例如高校中的教师这个概念是指在职的教学人员，他们的性质由姓名、性别、出生年月、工作单位、职称、专业特长等数据项加以描述。因此，教授、副教授、讲师和助教均可归入教师这一类，但他们在概念上并不完全相同，除了共性之外，还各有其特殊之处，如教授、副教授有研究方向、指导研究生等描述项，他们的职称也不一样。因此数据对象存在分类层次问题，为此可运用面向对象数据模型中子类与超类的概念。

##### ②识别实体与属性

建立局部E-R数据模型，还须识别每个对象类在局部E-R模型中的地位：实体、属性或联系。实体和属性之间在形式上的界限并不明显，常常是现实世界对它们已有的大体的自然区分，随应用环境的不同而不同。在给定的应用环境中区分实体和属性的总的原则是：要在此应用环境中显得合理，且同一个对象类在同一局部视图内只能做一种成分，不能既是实体又是属性。此外，为了对已给定的需求目标，更合理地确认局部E-R模型中的实体和属性，以便在逻辑设计阶段从E-R模型得到更接近于规范化的关系模式，可按下述一般规则来区分实体与属性。

##### · 描述信息原则

在E-R模型中的实体均有描述信息，而属性则没有。据此，可将一个需要描述信息的对象类作为实体，而将只需有一个标识的对象类归为属性。例如，仓库这个对象类在某些事务处理中需要用到仓库的面积、地点、管理员姓名等描述信息，则宜将其归入实体。但如人的年龄、物品的重量等对象类，在一般应用中都不需要描述信息，所以它们在通常情况下都作为属性。

#### · 多值性原则

所谓多值性是指若一个描述项存在多个值描述其描述对象，则即使该描述项本身没有描述信息也宜划为实体。例如加工种类与其描述的工件之间就符合多值性原则，因为每个工件实体可能需要多个工种的加工，尽管工种除了工种名外不需要其他的描述，但还是将工种作为实体为好。需要注意的是，这一原则最好与存在性原则结合起来使用，如果将被描述对象删除后，描述项没有再存在的意义，则即使此描述项是多值的也不宜作为实体。例如零件的颜色可以是多值的，但颜色离开了其描述的对象就没有单独存在的意义，因此还是作为属性为宜。

#### · 存在性原则

设对象类R的描述项A和B的值集分别为  $\{a_1, a_2, \dots, a_n\}$  和  $\{b_1, b_2, \dots, b_n\}$ 。若从R中去掉B的某个值  $b_i$ ，从而去掉与它的所有联系，如果  $b_i$  的消失对应用不产生任何影响，则B作为属性，否则B应作为实体。

#### · 多对一联系性

属性不再与其描述对象以外的其他对象类发生联系。相反，如果一个对象类的某个描述项与另一个对象类存在着多对一联系，则此描述项即使本身没有描述信息，也宜将其作为实体。例如在前面所讲的工件与加工种类的例子中，如果还有一个车间实体，且加工种类与车间之间存在多对一联系，则将加工种类划为实体更合适。

#### · 组合标识判别原则

若一个对象类的标识是由其他对象类的标识组成的，该对象一般应定义为联系。相反，如果组成某对象类标识的各成分不是其他实体的标识，且作为实体在应用的上下关系中很自然，则可以定义为实体。例如，一个工厂生产的零件须由名称及规格组成组合标识，在一般应用中应将零件作为实体较为适宜。

实体与属性的识别过程是个相互作用的反复过程，随着实体的确认，属性也将趋于明朗，反之亦然。在此过程中根据应用需求对已经识别的实体和属性做适当指派，发现问题再来调整，在识别过程中必须遵循前面所讲的总原则。

### ③对象的命名

在需求分析中得到的数据对象通常都是有名称的，但由于这些名称未经规范化，常常存在诸如同名异义、异名同义等许多命名上的冲突、不一致性及语义不清等问题，是造成数据不一致性及数据冗余的一个重要原因。此外，数据对象原有的名称有的过于冗长，给使用带来了很大不便。为此，需要按一定的规范对每一类数据对象重新命名，给它指定一个简洁明了且惟一的名字，避免异义同名和异名同义存在。命名规范一般包含以下原则和规定：

- 数据对象名应清晰明了便于记忆，并尽可能采用用户熟悉的名字。
- 名字要反映数据对象的主要特点并力求简洁，以利于减少冲突和方便使用。
- 遵守缩写规则。缩写规则包括一般缩写规则和系统中专用名称的缩写规定，对于较大的复杂系统应编制缩写字典，便于参照。凡有缩写规定的不得使用全称，以免混淆。
- 规定统一的命名约定并加以遵守。例如对属性的命名可以采用如下形式的约定：

实体名·分类词——修饰词

其中，分类词指单位内通用的数据项名，如名称、号码、小计、总计、合计、单位、摘要、日期、时间等。每个单位应有一个标准的分类词表，加于分类前面的实体名和后面的修饰词可用以说明该分类词在特定的上下文中的特殊含义。例如，合同上的数据项日期可命名为：

制定命名约定的基本原则是简明一致、避免混淆，具体可根据单位内数据对象的复杂程度自行制定。

在完成了实体与属性的识别后，必须按照命名规范仔细地审核每类数据对象的名字，纠正不符合规范的命名，务必使每个对象名符合规范要求。

#### ④确定实体的标识（Identifier）

实体的标识能够惟一地标识一个实体的属性或属性组，也就是该实体的关键字。确定实体标识在实体识别与规范化命名之后进行，首先确定每个实体的候选关键字。一个实体可能有若干候选关键字，选择其中对有关用户最熟悉的一个候选关键字作为主关键字（或主码）。并将每个实体的候选关键字、主关键字记入数据字典。

### （3）确定实体间的联系

实际上，识别联系的主要任务是在需求分析阶段完成的。这里的工作一是从局部视图的角度进行一次审核，检查有无遗漏之处；二是确切地定义每一种联系。

在现实世界中，诸多形式的联系大致可以分为三类：存在性联系、功能性联系和事件联系。

存在性联系如学校有教师、教师有学生、工厂有产品、产品有顾客等等；功能性联系如教师讲授课程、教师参与科研、仓库管理员管理仓库等等；事件联系如学生借书、产品发运等等。

根据上述分类仔细检查在给定的局部视图范围内是否有未识别的联系，在确认所有的联系都已识别并无遗漏之后，还须对联系进行正确的定义。定义联系就是对联系语义的仔细分析，识别联系的类型，确定实体在联系中的参与度。

#### ①二元联系的类型与定义

二元联系是指两个实体类之间的联系。根据参与联系的两个实体类值之间的对应关系分为一对一、一对多及多对多3种类型。对每一种联系类型，要确定实体在联系中的参与度，并以 $m:n$ 的形式标在E-R图上要说明的实体旁。若 $m>0$ ，表明该实体参与联系是强制性的，若 $m=0$ 则是非强制性的。下面分别讨论上述三类联系的定义，以及另一种二元联系——实体类内部的联系。

##### · 一对一联系

这是一种最简单的联系类型。若对于实体集A中的每一个实体，实体集B中至多有一个实体与之联系，反之亦然，则称实体集A与实体集B具有一对一联系，记为1:1。例如在一个施工单位中，如果规定每项工程最多只能由一名工程师负责管理，而一名工程师最多也只能负责一项工程。则工程师与工程间的这种管理联系便是一对一联系。按照实体参与联系的强制性情况，又可分为以下3种情况：

##### a. 两类实体都是强制性的。

假如规定每个工程师一定要负责一项工程，每项工程也一定要有一位工程师负责，便属于此种情况。如果工程师的标识为职工号，工程的标识为工程号，对于工程师与工程间的1:1联系，可用职工号或工程号作为标识。

##### b. 其中仅有一类实体是强制性的。

若规定每项工程必须由一名工程师负责，但并不是所有工程师都必须负责一项工程（因为有可能没有那么多工程），此时，每一项工程一定对应着惟一的负责联系，所以工程号可用做联系的标识。

##### c. 两类实体均为非强制性的。



如工程师不一定安排负责管理工程，有的工程项目暂时可以不安排工程师负责管理，这种情况表示凡分到工程项目的工程师与工程项目之间总存在一一对应的联系，因此职工号或工程号均可选为负责联系的标识，确定了其中一个为标识，另一个就可以作为候选关键字。

#### · 一对多联系

若对于实体集A中的每一个实体。实体集B中有 $n$ 个实体（ $n \geq 0$ ）与之联系，反之，对于实体集B中的每一个实体，实体集A中至多有一个实体与之联系，则称实体集A与实体集B有一对多的联系，记为 $1:n$ 。以专业与学生间的关系为例，如规定一个专业可以管理许多学生，每个学生只能属于一个专业，这种联系就是一对多联系。对这种联系我们只需关心“多端”实体的强制性，分两种情况进行讨论。

##### a. “多”端的实体是强制性的。

此时，每个学生必须归属某个专业，即每个学生都有一个确定的专业，但每个专业都不惟一地对应一个学生，故可以选择学号作为联系的标识。

b. “多”端的实体是非强制性的。对本例而言是指有些学生（如新生）不属于任何专业的情况。此时实际上仅表示已经分配专业的学生与专业之间的联系，对这些学生中的每一个都有一个确定的专业，因此，应以学号为联系的标识，而专业代号只作为联系的一般属性。

#### · 多对多联系

若对于实体集A中的每一个实体，实体集B中有 $n$ 个实体（ $n \geq 0$ ）与之联系，反过来，对实体集B中每一个实体，实体集A中也有 $m$ 个实体（ $m \geq 0$ ）与之联系，则称实体集A与实体集B具有多对多联系，记为 $m:n$ 。教师与学生这两个实体类间的教与学的联系就是多对多的联系。这时，只有<教师、学生>对才能确定一个特定的教学联系。因此，一般情况下可以用两个关联实体的标识拼凑（Concatenate）作为联系的标识。但这种方法对某些情况就不能构成有效的联系标识。当一个实体值在同一个联系上可能存在多个不同的联系值时，就会出现这种情况。如教师与其讲授的课程之间的联系，同一个教师可讲授几门不同的课程，也可以多次讲授同一门课程，这是一种特殊的多对多联系，这种情况可用如图3-1（d）所示的值图（Occurrence Diagram）表示。值图是表示具体的实体及其关联的一种图示法，其中圆点表示具体的实体，连线表示实体间的联系。而通常的E-R图只能表示型而不表示值，所以称为型图。显然，对于教师与讲授课程间的联系，如在教师档案中要求记录担任教学工作的情况，就需要在联系标识中增加表示授课日期的属性，即其合适的联系标识可能为（教师号，课程号，授课日期）。

#### · 实体类内部的联系

这种联系发生在同一类实体的不同实体之间，因此称为内部联系或自联系，它也是一种二元联系，其表示方式与前面的二元联系并无不同，要注意仔细区别同一实体类中的不同实体在联系中扮演的不同角色及联系标识的选择。例如在职工类实体中间就存在着管理者与被管理者的联系。一个职工可以管理别的职工，称为管理者或领导者。一个管理者可以管理多个职工，而一个职工最多只从属于一位管理者，从而构成了一对多联系。若规定所有职工都要受管理（最高管理者考虑自己管理自己），但不是所有职工都是管理者，则此联系在“多”端呈现强制性。其中每个联系实体包含两个职工号值：职工号和管理员职工号，以区别不同实体在联系中的角色。

若省略去实体与其属性图，以上实体间的二元联系可用图3-1表示。

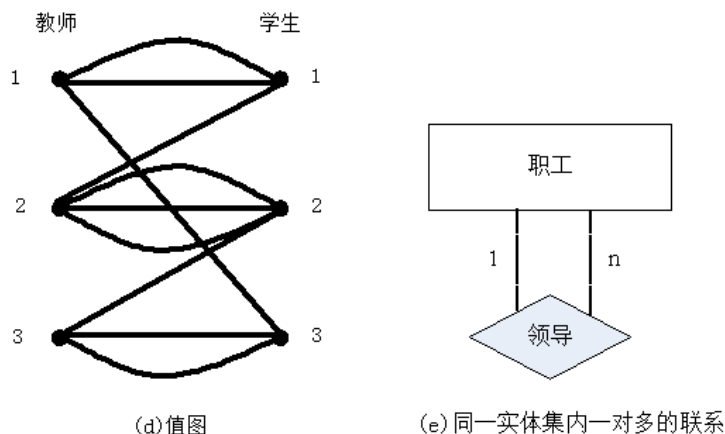
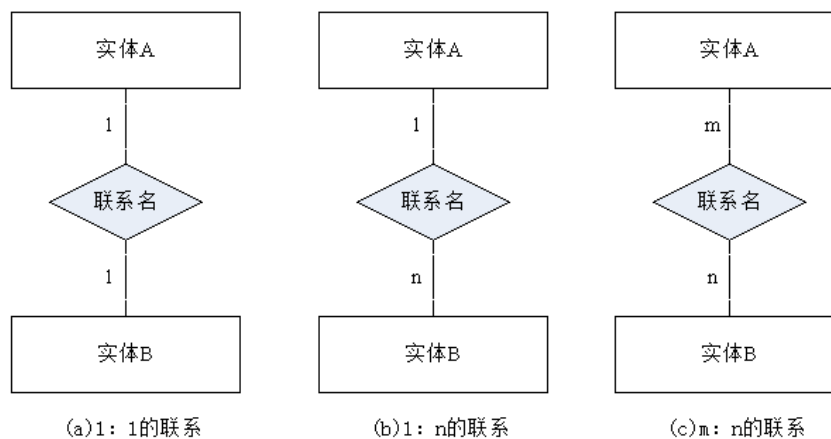


图3-1 实体间的联系

## ②多元关系的类型与定义

两个以上的实体类之间的联系称为多元联系。例如在供应商向工程供应零件这类事件中，如果任一供应商可向任一工程供应任一种零件，则为了确定哪个供应商向哪个工程供应了何种零件，就必须定义一个三元联系，因为只有供应商、工程及零件三者一起才能惟一地确定一个联系值。其联系的标识由参与联系的实体类的标识拼接而成，在此例中由供应商、工程、零件3个实体类的标识拼接而成。

需要注意的是，涉及到多个实体的事件是否属于多元联系完全取决于问题的语义，不可一概而论。例如，如果上例中的问题说明变成每个工程需要订购一定的零件，而任一供应商可向任一工程供应零件。这里有两层意思，一是只有工程确定了才能确定订购的零件，二是只有供应商及工程确定了，才能确定一个供应关系。根据这一情况，应定义两个二元联系，如图3-2 (a) 所示。

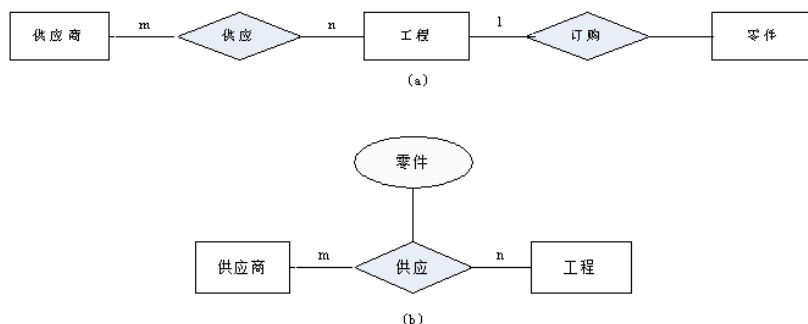


图3-2 多元联系

假如问题的说明是任一供应商向任一工程供应零件，但某个供应商向某项工程供应的零件是一定的，则在供应商与工程之间的关系确定后，供应的零件也就确定了。由此可知，只需定义一个二元联系就可以，如图3-2 (b) 所示，其中供应的零件作为供应联系的一个属性。

总之，具体问题应该具体分析，以便使定义的模式确切地表达问题的语义。

### ③消除冗余联系

若出现两个或两个以上的联系表示的是同一概念，则存在着冗余的联系，具有冗余联系的E-R模型转换为关系模型可能会得到非规范化的联系，因此必须予以消除。

出现冗余联系的一个重要原因是存在传递联系。例如，图3-3表示了产品与零件之间的“组成”联系，零件与材料之间的“消耗”联系及产品与材料间的“使用”联系。其中，材料与零件间的联系是1:n，零件与产品之间的组成联系是n:m，其实由这两个联系必然得出产品与材料之间的使用联系m:n。因此，图中产品与材料间的联系是冗余的，应将其去掉。

### ④警惕连接陷阱（Connection Traps）

连接陷阱是一种存在语义缺陷的联系结构，它是由于在定义联系过程中对语义的理解出现偏差而造成的，因而无法由它得到需要的信息。连接陷阱可分为扇形陷阱、断层陷阱和深层的扇形陷阱3种情况。

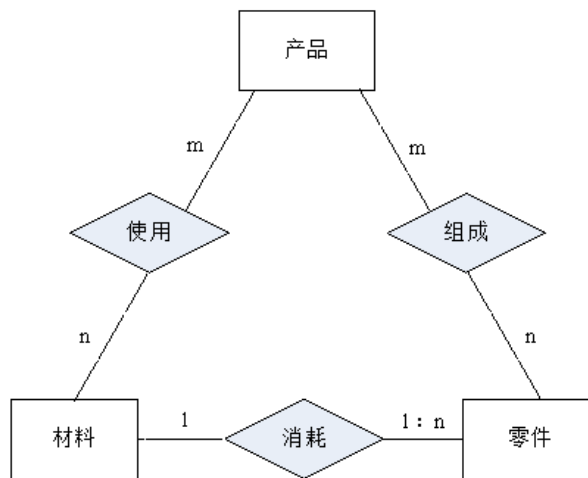


图3-3 存在冗余联系

### · 扇形陷阱

两个实体类间的一对多联系，由一个实体值引出多个同一类型的联系值，其值图是一种扇形结构，故称之为扇形联系。扇形陷阱则是指由一个实体引出两种不同类型的扇形联系，形成双扇形结构。图3-4（a）是这种结构的一个例子，图3-4（b）是一个值图。从图3-4（b）值图可以看出，从这种联系结构无法获得哪个职工属于哪个专科的信息，其原因是将专科与职工之间的联系通过医院来连接了。如采用图3-5（a）所示的联系结构，图3-5（b）是它的值图。新的结构能较自然地表示了医院、专科及职工之间的层次关联。假定任一医院的职工无一例外地分属于医院的各个专科的话，该结构可以确定一个职工所属的专科或医院，但如果允许某些职工直属医院而不属于任何专科，那么这种结构还是不适用的。

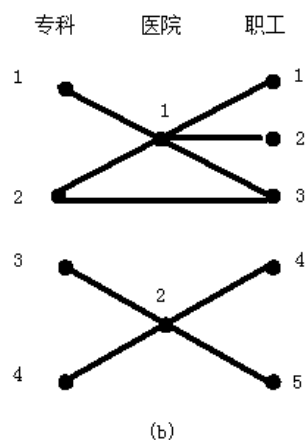
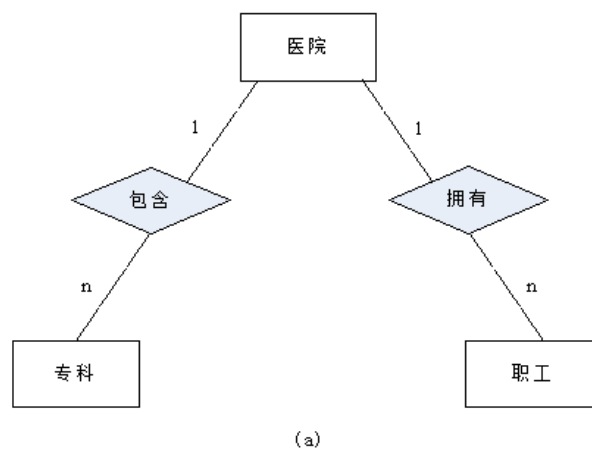


图3-4 扇形陷阱及其值图1

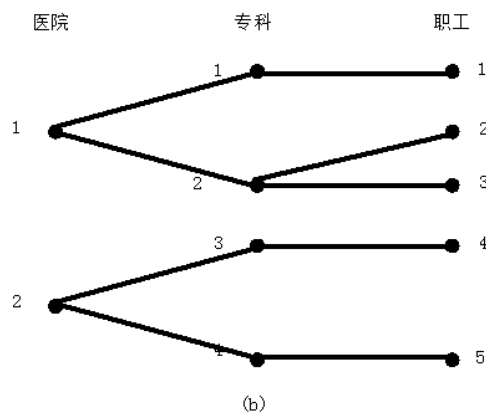
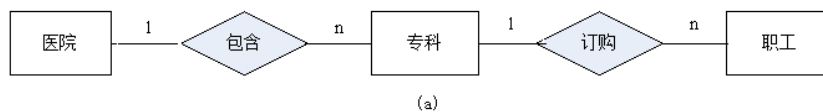


图3-5 扇形陷阱及其值图2

#### · 断层陷阱

断层陷阱是指因为型图所含的传递联系掩盖了某些特定的直接联系的现象。例如图3-5 (a) 的联系结构虽然隐含了医院与其职工的联系（传递联系），却没有提供部分职工直属医院的联系路径，因而出现了断层陷阱。解决办法是设置一个虚构的专科或增加一个联系，如在本例中增加一个医院-职工联系。增加虚构实体和现实世界不符，因此可考虑增加一个联系，但增加新的联系在某些情况也可能带来新的麻烦，见下面的内容。

#### · 深层的扇形陷阱

举一个“教师指导学生参加课题”的例子，若每个学生可在多位教师指导下参加多个课题研究，每位教师可指导多名学生，但现在只允许一名教师指导一名学生参加一个课题，不允许多名教

师指导同一名学生参加某个课题。对此可先建立一个由两个多对多的二元联系组成的模式，如图3-6 (a) 所示。利用联系的分解法则将其分解为如图3-6 (b) 所示的结构，它是以学生为中心的双扇形结构，图3-6 (c) 是它的值图，从该联系结构无法得到哪位教师指导哪个学生参加何课题的信息，这表明存在扇形陷阱。图3-7 (a) 是对它的一种改进，在其中增加了一个教师与课题的联系。该联系结构能确切地提供“教师1指导学生1参与课题1”及“教师2指导学生2参与课题1”的信息，但从此图无法确定教师1指导学生2参与了哪一个课题。对教师2和学生1也是如此，因为参加课题1或2均是正确的语义。之所以如此，原因在于新增加的教师与课题间的多对多联系带来了两个新的双扇形结构，即以教师为中心的及以课题为中心的双扇形结构。增加的新联系虽然消除了原来的陷阱，却产生了新的陷阱。对这类问题的有效解决办法是将三个实体间的联系定义成一个单一的三元联系，它的E-R模型如图3-7 (b) 所示。现在，每一个联系值惟一地确定了另一个联系值，从中可获得哪位教师指导某学生参加哪一课题的确定信息。可见问题的实质是对于应该定义成三元联系的问题千万不能用二元联系代替。

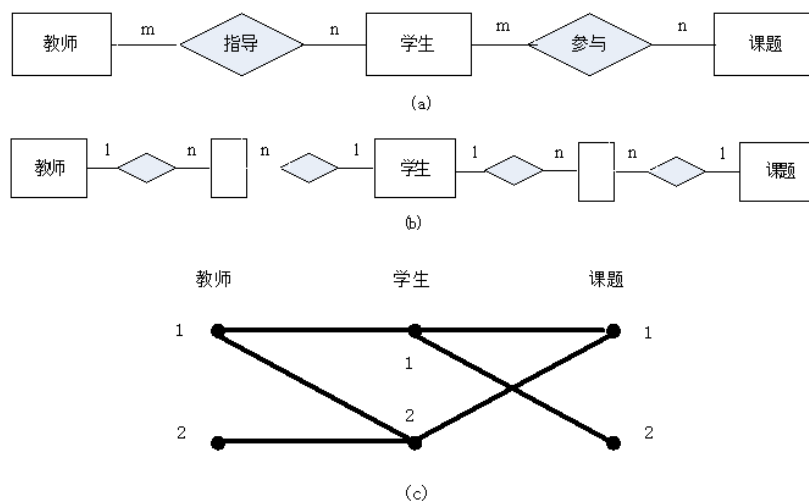


图3-6 深层的扇形陷阱

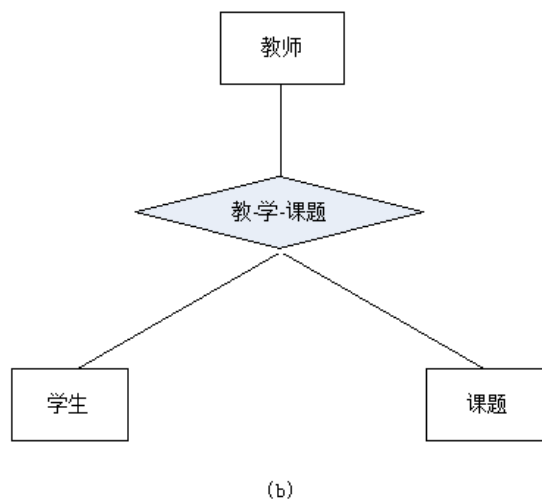
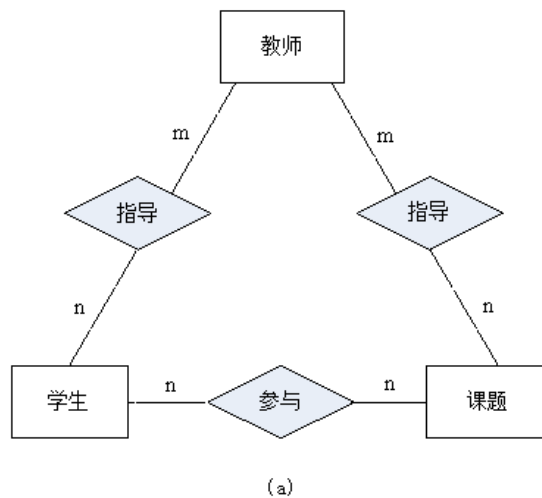


图3-7 改进的扇形结构

#### (4) 分配实体及联系的属性

在需求分析中收集的数据对象集内，除去已识别的实体、联系及标识外，剩下的主要是非标识属性。问题的实质就是将这些非标识属性恰当地分配给有关实体或联系。不过分配这些属性时，应避免使用用户不易理解的属性间的函数依赖关系及其有关准则，而应该从用户需求的概念上去识别框架中实体或联系必须有的描述属性，并按下述两条原则分配属性。

##### ①非空值原则

所谓非空值原则是指当一个属性的分配在几个实体或联系中可以选择时，应避免使属性值出现空值的分配方案。例如在前面曾经讲过的工程师负责工程的模式中，其联系是一对一且两个实体类均属非强制性的情况，现有一个属性项“施工期限”，按依赖关系考虑可加给工程师、工程或负责联系三者中的任何一个，因为工程师与工程在这里是一对一的联系，工程师定了，工程的施工期限也就定了。但有的工程师可能没有分配到负责工程的任务，若把施工期限作为工程师的属性，在此情况下便会出现空值；若作为工程的属性，则工程可能还未纳入计划，也会出现空值，可见将“施工期限”分配给负责联系最为适宜。

##### ②增加一个新的实体或联系

在分配属性过程中，有时会出现有的数据项在框架模式中似乎找不到适合依附的实体或联系。对于这种情况，常常可通过在原模式中增加一个新的实体或联系加以解决。例如，图3-8表示了一个病区/病人模式，这是一个一对多，且“多”端为强制性的联系，在所有明显的属性均已分配完后，尚有属性项手术名及手术号待分配。这里手术名依赖于手术号，一个病人可能接受多种手术，一个

病区可能接纳不同种手术的病人。在此情况下，手术号与手术名两个属性项不论作为哪个实体或联系的属性均不合适，为此可以在原来的模式中增加一个新的实体——手术，再加一个病人与手术的联系就可以了，如图3-9所示。

按上述属性分配原则建立的模式将有利于转换成规范化的关系模式。

在属性综合法中，其高层的实体、实体间的联系是通过对底层数据元素的分析、归类和聚集而逐步形成的。从具体的设计过程而言，属性综合法是一种统计分析的方法，它利用统计数据元素与事务处理的关系导出结果。此法的优点是只要有一本好的数据字典和一张能反映用户需求的数据流程图，设计者即使不太熟悉业务，也能应用这种方法开展工作。其缺点是当应用环境比较复杂、数据元素较多时，很难用人工进行统计分析。

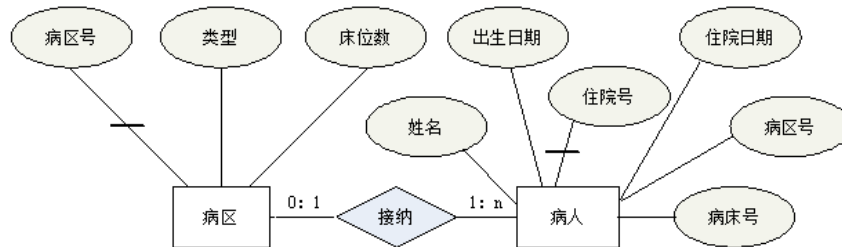


图3-8 病区/病人模式

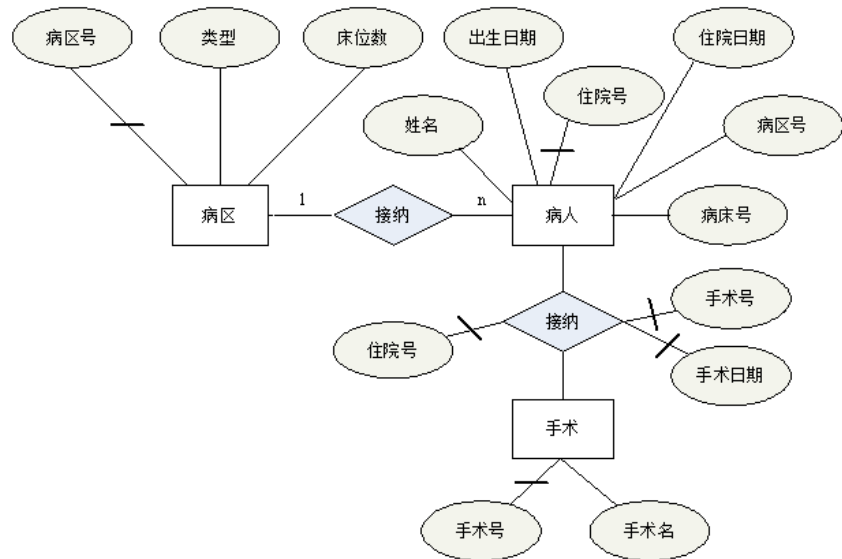


图3-9 改进后的模式

版权方授权希赛网发布，侵权必究

[上一节](#)    [本书简介](#)    [下一节](#)

## 逻辑设计

数据库逻辑结构设计任务就是把概念结构设计阶段设计好的基本E-R图（或E-E-R图）转换为与具体机器上的DBMS产品所支持的数据模型相符合的逻辑结构。这一阶段是数据库设计的重要阶段。

数据库逻辑设计的基础是概念设计的结果，而其成果应包括为某DBMS支持的外模式、概念模式及其说明，以及建立外模式和概念模式的DDL程序。因此，进行逻辑设计前，必须了解数据库设

计的需求说明和概念设计的成果（包括E-E-R图和其他文档），并仔细阅读有关DBMS的文件。数据库的外模式和概念模式是用户所看到的数据库，是应用程序访问数据库的接口。因此在数据库逻辑设计阶段，还必须提供应用程序编制的有关说明，最好试编一些典型的访问数据库的应用程序，以检验所设计的概念模式是否满足使用要求。

概念模式是数据库的基础，它的设计质量对数据库的使用和性能及数据库在今后发展过程中的稳定性均有直接的影响。为了设计出能够正确反映一个项目的数据库的内在联系的概念模式，设计时必须正确处理各种应用程序之间、数据库性能与数据模式的合理性和稳定性之间的各种矛盾，对设计中出现的各种矛盾要求要权衡利弊，分清主次，按统筹兼顾的原则加以正确处理。

逻辑结构设计一般分为以下几个步骤：

- （1）将概念结构向一般关系模型转化。
- （2）将第一步得到的结构向特定的DBMS支持下的数据模型转换。
- （3）依据应用的需求和具体的DBMS的特征进行调整与完善。

下面以常用的E-R模型和扩充E-R模型为主，针对关系数据库的逻辑设计介绍基本原则和方法。

### 1. E-R图向关系模型的转换

这里讨论基本E-R数据模型向关系模型的转换及扩充E-R数据模型的转换。

#### （1）基本E-R模型向关系模型的转换

基本E-R模型主要包含实体和联系两个抽象概念，实体和联系本身还可能附有若干属性。其转换的基本原则是，实体和联系分别转换成关系，属性则转换成相应关系的属性。因此，E-R模型向关系模型的转换比较直观，但不同元数的联系具体转换方法稍有不同，下面根据不同的情况分别讨论。

##### ①一对一联系（1:1）

设有两个实体E1和E2之间为一对一联系。此情况存在3种可能的转换方案。

· 方案1：将实体E1、E2和联系名R分别转换成为关系E1、E2和R，它们的属性分别转为相应关系的属性，即得到：

$E1(\underline{k1}, a)$

$E2(\underline{k2}, b)$

$R(\underline{k1}, k2, r)$ （k2为候选关键字）

其中属性下面带横线者为关系的关键字。

· 方案2：将实体E1转换为关系E1，将实体E2与联系名R一起转换成关系E2'，E2'的属性由E2和R的属性加上E1的关键字组成，其关键字为k2，k1为其候选关键字。转换后的关系为：

$E1(\underline{k1}, a)$

$E2'(\underline{k2}, b, k1, r)$ （k1为候选关键字）

· 方案3：与方案2类似，不过是把实体E1与联系R一起转换成关系E1'，其结果为：

$E1'(\underline{k1}, a, k2, r)$ （k2为候选关键字）

$E2(\underline{k2}, b)$

上述3个方案实际上可归结为转换成3个关系和转换成两个关系两种。如果每个实体的属性数较少，而联系的属性与两个实体之一关系又密切，则可采用方案2或方案3，其优点是可减少关系数，有利于减少联接运算提高查询效率，但如果每个实体的属性较多，且合并后，会造成较大数据冗余和操作异常，则以采用方案1为宜。

##### ②一对多联系（1:n）



这种情况存在两种转换方案：

第一个方案是把两个实体类和一个联系类分别转换成对应的关系，实体的属性转换为对应关系的属性，其标识属性即为对应关系的关键字，而联系转换得到的关系其属性由两个实体的标识属性和联系类本身的属性组成，并以多端实体类的标识属性为其关键字。其转换结果为3个关系。

第二个方案是转换成两个关系，设少端和多端的两个实体类分别为E1、E2，联系名R。转换时，将E1转换为一个关系E1，E2和R合起来转换成一个关系E2'，E2'的属性由E2和R的属性加上E1的标识属性组成，并以E2的标识属性为其关键字。

### ③多对多联系（m：n）

两个实体类之间多对多联系组成的E-R模型向关系模型转换时，将两个实体和一个联系分别转换成关系，实体类的属性分别转换成对应关系的属性，其标识属性为其关键字，由联系转换得到的关系的属性由两个实体类的标识属性和联系本身的属性组成，其关键字是由两个联系的实体类的标识属性组成的。

### ④多元联系

实体类分别转换为相应的关系。3个实体类间的多元联系转换为以该联系名为关系名的关系，关系的属性由各实体的标识属性及其联系的属性组成，并以各实体的标识属性为其关键字。

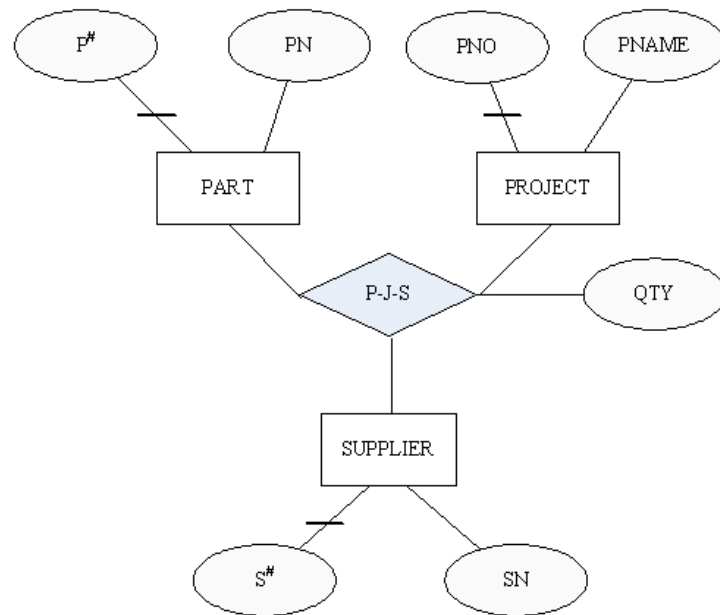


图3-10 多元联系

如图3-10所示的部件（PART）、工程（PROJECT）和供应者（SUPPLIER）三者之间的联系为P-J-S，其属性为QTY。转换时，把PART、PROJECT、SUPPLIER和联系PJS分别转换为相应的关系，其结果如下：

PART ( P# , PN )

PROJECT ( PNO , PNAME )

SUPPLIER ( S# , SN )

PJS ( P# , PNO , S# , QTY )

### ⑤自联系

自联系是同一实体集的实体间的联系。例如对于职工实体类内部有领导与被领导的联系，在部件这个实体集的实体之间有组成成分与组成者之间的联系等，均属于实体类的自联系。在这种联系中，参与联系的实体虽然来自同一实体类，但所起的作用不一样。

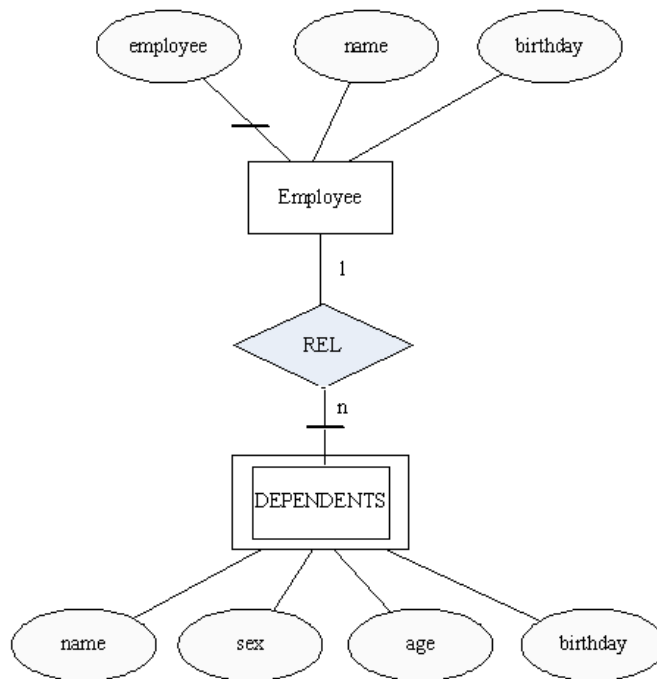


图3-11 弱实体类的转换

#### ⑥弱实体类的转换

一个实体类，如果它的存在依赖于另一实体类，则称为弱实体类。例如职工的亲属（DEPENDENTS）是依赖于职工（EMPLOYEE）实体类而存在，因此实体集亲属（DEPENDENTS）是弱实体类，它们之间的关系如图3-11所示。由于弱实体类不能独立地存在，而是由其他实体标识而存在。所以不能单独地转换成一个关系，因此上图可转换成如下两个关系：

EMPL ( empno , name , birthday )

DEPENDENTS ( empno , name , sex , age , kinship )

其中kinship表示职工亲属与职工的关系，可以取值为“配偶”、“儿子”、“女儿”，等等。

#### ⑦带有非原子属性的实体的转换

属性的原子性是关系模型对每个关系的基本要求。但E-R数据模型允许用集合或聚合（Aggregate）作为属性，这类实体的转换与一般实体的转换有所不同。假如k是标识属性，a是普通原子属性，r是集合属性， $r = \{ r_i | i = 1, 2, \dots, n \}$ ，g是聚合属性，由原子属性g1、g2聚合而成，则此实体E可转换成下列两个关系：

$E(k, a, g1, g2)$

$E'(k, r_i), i = 1, 2, \dots, n$

其中k表示关系的主关键字。

#### （2）扩充E-R模型向关系模型的转换

扩充E-R模型是基本E-R模型的扩充。它主要扩充了两点，一是一个实体集可能是另一个实体集中的某个属性，即一个实体集可以附属于另一个实体集，二是增加了一种叫ISA的特殊联系，这种联系建立了两个实体集间的继承（Inheritance）关系，通过这种联系可以构成实体集之间普遍化/特殊化层次结构。下面讨论这些扩充部分的转换，这些转换方法也可推广到其他具有这些概念的数据模型。

##### ①一个实体集同时是另一实体集的属性

设实体集E1有属性k1、a1、k2，其中k1为E1的标识属性，而属性k2是另一实体E2的标识属性，E2有属性k2、e1、e2，则可转换成两个关系：

$E_1(k_1, a_1, k_2)$

$E_2(k_2, e_1, e_2)$

图3-12 (a) 是这种情况的一个实例，它可转换成工厂与厂长两个关系：

工厂 (厂名, 性质, 地址, 厂长工作证号)

厂长 (身份证号, 姓名, 性别, 年龄)

## ②两个实体集间ISA联系的转换

这种情况中的实体 $E_0$ 是实体集 $E_1$ 的超集，而实体 $E_1$ 是实体集 $E_0$ 的子集。 $E_1$ 继承 $E_0$ 的全部属性，同时， $E_1$ 可以有其自己的属性。图3-12 (b) 是这种联系的一个实例。

对这种E-E-R图，将其超类和子类分别转换成两个关系，并且子实体集以其超类实体集的关键字为关键字。即：

$E_0(k, a)$

$E_1(k, b, c)$

据此，图3-12 (b) 所示的实例可转换成下面两个关系：

学生 (学号, 姓名, 年龄, 系别)

研究生 (学号, 导师姓名, 研究方向, 攻读学位)

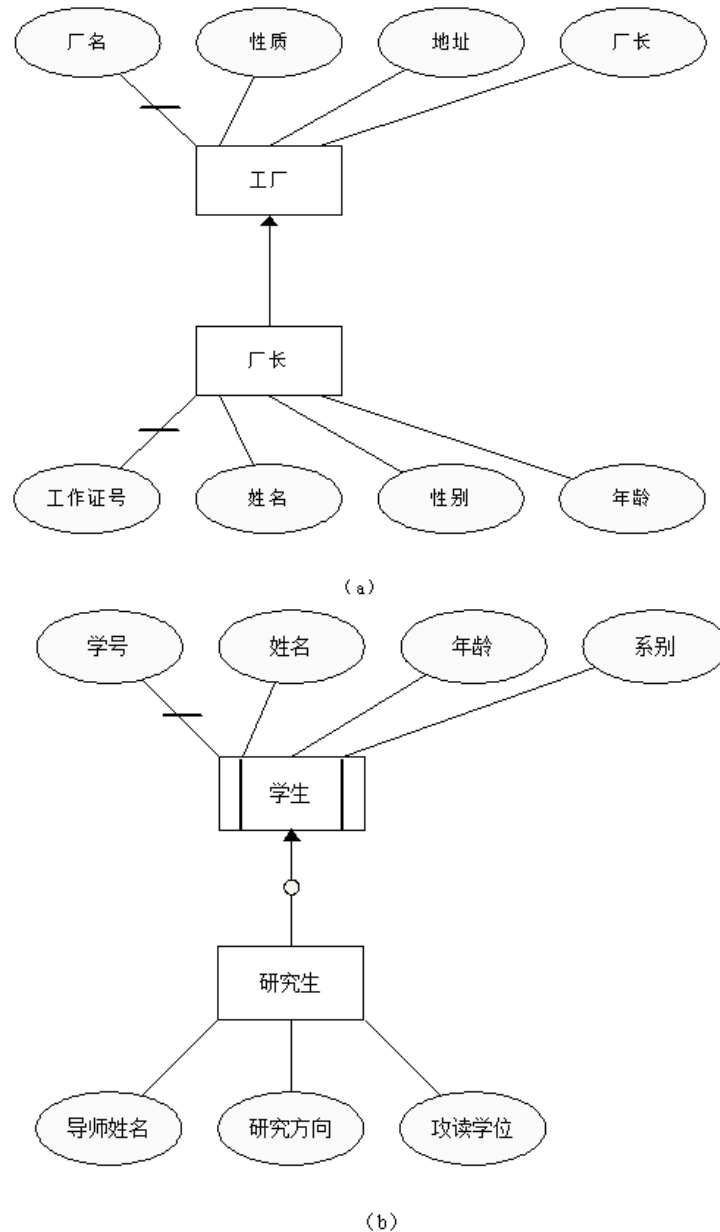


图3-12 扩充E-R模型向关系模型的转换

### ③一个超集具有多个子集时的转换

设超实体集的属性集为  $\{k, a_1, \dots, a_n\}$ ，它有  $m$  个子实体集  $E_1, \dots, E_m$ ，子实体集的属性集为  $\text{Attr}(E_i)$  ( $1 \leq i \leq m$ )，子实体可能不相交，也可能重叠。这种情况可把  $E_0, E_1, \dots, E_m$  分别转换成如下关系：

$E_0(k, a_1, \dots, a_n)$

$E_i(k, \text{Attr}(E_i))$  ( $1 \leq i \leq m$ )

### (3) 一般关系模型向特定关系模型的转换

在逻辑结构方面，一般关系模型结构与目前较常用的DBMS支持的关系模型结构并无明显冲突。设计时需要注意下述问题：

#### ①DBMS支持的数据类型

一般而言，DBMS只支持有限的几种数据类型，而E-R数据模型及E-E-R模型则无此限制，据此转换而得到的关系模型，必然保留了E-R或E-E-R模型中的数据类型。因此，在向具体的DBMS所支持的关系模型转换时，对DBMS不支持的数据类型必须做相应修改。如果用户坚持要使用原来的数据类型，就可能导致数据库的数据类型与应用程序中的数据类型不一致，应用程序必须负责这两者之间数据类型的转换。

②DBMS对关系模式的数量、一个关系中属性的个数、关系名与属性名的长度等的限制。如果一般关系模型与DBMS的这些限制存在冲突，则按特定DBMS的要求进行修正。

必须注意的是，在向特定DBMS转换之前，须先对转换所得到的关系模式进行规范化处理。

## 2. 设计用户子模式

在ANSI/SPARC建议的数据库三级结构中，用户子模式（也称外模式）是用户所看到的数据库的数据逻辑结构。各个用户（或用户组）可以有各自的外模式。外模式是概念模式的子集，但在结构和形式上可以不同于概念模式，甚至可采用不同的数据模型，不过一般都是同一数据模型。

关系数据库的外模式由与用户有关的基表及按需要定义的视图构成。设计外模式时，可参照概念设计中的局部E-R（或E-E-R）图。在关系模型中，设计外模式是比较简单的。

## 3. 数据模型优化

由E-R（或E-E-R）图表示的概念模型转换得到的关系模式经过规范化以后，基本上可以反映一个企业数据的内在联系，但不一定能满足应用的全部需要和系统要求，因此，还必须根据需求分析对模式做进一步的改善和调整，其内容主要是改善数据库的性能和节省存储空间两个方面。

### (1) 改善数据库性能的考虑

查询速度是关系数据库应用中影响性能的关键问题，必须在数据库的逻辑设计和物理设计中认真加以考虑，特别是那些对响应时间要求较苛刻的应用，应予以特别注意。

就数据库的逻辑设计而论，可从下列几个方面提高查询的速度：

#### ①减少联接运算

联接（Joins）运算对关系数据库的查询速度有着重要的影响，联接的关系越多，参与联接的关系越大，开销也愈大，因而查询速度也愈慢。对于一些常用的、性能要求较高的数据库查询，最好是一元查询，这与规范化的要求相矛盾。有时为了保证性能，往往不得不牺牲规范化要求，把规范化的关系再合并起来，称为逆规范化（Renormalization）。当然，这样做会引起更新异常。总之，逆规范化有得有失，设计者可根据实际情况进行权衡。

#### ②减小关系大小及数据量

被查询的关系的大小对查询速度影响较大。为了提高查询速度，可以采用水平分割或垂直分割等方法把一个关系分成几个关系，使每个关系的数据量减少。例如，对于大学中有关学生的数据，既可以把全校学生的数据集中在一个关系中，也可以用水平分割的方法，分系建立关系，从而减少了每个关系的元组数。前者对全校范围内的查询较方便，后者则可以显著提高对指定系的查询速度。也可采用垂直分割的方法，把常用数据与非常用数据分开，以提高常用数据的查询速度。例如，高校中教职工档案的属性很多，有些需经常查询，有些则很少查询，如果放在一起，则关系的数据量就很大，影响查询速度，把常用属性和非常用属性分开，就可提高对常用属性的查询速度。

### ③尽量使用快照 ( Snapshot )

快照是某个用户所关心的那部分数据，与视图一样是一种导出关系 ( Derived relation )，但它与视图有两点不同。一是视图是虚关系，数据库中并不存储作为视图的导出关系，仅仅保留它的定义。快照则是一个由系统事先生成后保留在数据库中的实关系。二是视图随数据当前值的变化而变化，快照则不随原来关系中数据的改变而及时改变，它只反映数据库中某一时刻的状态，不反映数据库的当前状态，犹如照片只反映某一时刻的情景，不能反映情景变化一样，之所以称它为快照，原因就在于此。

但它与照片又有不同，快照不是一成不变的，它可以由系统周期性地刷新 ( Refresh )，或由用户用命令刷新。刷新时用当前值更新旧值。在实际应用中，快照可满足相当一部分应用的需要，甚至有些应用就是需要快照，而不是当前值。例如注明列出“某年某月某日截止”的统计或报表就是快照。由于快照是事先生成并存储在数据库中的，因而可大大缩短响应时间。目前不少DBMS，如Oracle、MS-SQL Server等支持快照。对不支持快照的DBMS，用户也可以把需要作为实关系使用的导出关系作为一个独立关系存于数据库中，但这种做法只能供查询使用，对它们的刷新及管理由用户负责。

### ( 2 ) 节省存储空间的一些考虑

尽管随着硬件技术的发展，提供用户使用的存储空间越来越大，但毕竟仍是有限度的，而数据库，尤其是复杂应用的大型数据库，需要占用较大的外存空间。因此，节省存储空间仍是数据库设计中应该考虑的问题，不但要在数据库的物理设计中考虑，而且还应在逻辑设计中加以考虑。

数据库逻辑设计中可采取以下措施：

#### ①缩小每个属性占用的空间

减少每个属性占用的空间，是节省存储空间的一个有效的措施。通常可以有两种方法。

- 用编码表示属性。例如用编码代替校名、系名、专业名称等比直接用文字表示要短得多，因而有效地节省存储空间。

- 用省略符表示。例如，用DS表示数据结构，用OS表示操作系统等。凡是能用省略符表示的尽量用省略符表示，同样能节省空间。

这两种方法的缺点是失去了属性值含义的直观性。

#### ②采用假属性 ( Dummy Attribute )

采用假属性可以减少重复数据占用的存储空间。设某关系模式R的属性A和B之间存在函数依赖 $A \rightarrow B$ ，B的每一个值需要占用较大的空间，但B的域中不同的值却比较少，A的域具有较多的不同值，则B的同一值可能在多个元组中重复出现，从而需要占用较多的空间。为了节省空间，可利用属性B的域中不同值少的特点，对B的值进行分类，用B'表示B的类型，则 $A \rightarrow B$ 可分解成两个函数依赖，即

$A \rightarrow B'$ ,  $B' \rightarrow B$

这样,就可用 $B'$ 代替原来元组较多的关系 $R$ 中的属性 $B$ ,而另外建立一个较小的关系 $R'$ 来描述 $B'$ 与 $B$ 的对应关系。这里 $B'$ 在原关系 $R$ 中起了属性 $B$ 的替身的作用,所以称 $B'$ 为假属性。例如,在职工关系中,职工的经济状况这一属性,通常由职工号决定。一个大型企业的职工人数虽多,工资级别、其他经济来源等,如每一职工逐一填写,就要占用较多的空间。为了节省空间可把经济状况分为几种类型,在元组较多的职工关系中用经济状况的类型代替原来的经济状况,这里经济状况的类型就是假属性。另外建立一个较小的关系来描述每种经济状况类型的具体内容。

数据库设计与数学问题求解不同,它是一项综合性工作,受到各种各样要求和因素的制约,有些要求往往又是彼此矛盾的,因此,设计结果很难说是最佳的,常常是有得有失。设计者必须根据实际情况,综合运用上述原则和有关理论,在基本合理的总体设计的基础上,做一些仔细的调整,力求最大限度地满足用户各种各样的要求。

版权方授权希赛网发布,侵权必究

[上一节](#)      [本书简介](#)      [下一节](#)

第3章：数据库设计

作者：希赛教育软考学院    来源：希赛网    2014年05月05日

## 主键及相关概念

关系数据库系统是支持关系数据模型的数据库系统。关系数据模型 ( Relation Model ) 由关系数据结构、关系操作集合和关系完整性约束3大要素组成。

关系模型把数据库表示为关系的复合,在关系模型中,现实世界的实体以及实体间的各种联系均用关系来表示。

在关系数据模型中,数据结构用单一的二维表结构来表示实体及实体间的联系,如图3-13所示。

### (1) 关系 ( Relation )

一个关系对应一个二维表,二维表的名称就是关系的名称。

例如,图3-13包含两个表,也即两个关系:学生登记表关系和系信息表关系。

### (2) 属性 ( Attribute ) 和值域 ( Domain )

在二维表中的列 ( 字段 ) 称为属性。属性的个数称为关系的元数。列的值称为属性值;属性值的取值范围称为值域。

例如,图3-13中学生登记表关系的属性有学号、姓名、性别、年龄、系号、原单位共6个属性,所以元数是6。年龄属性的值域是大于等于15岁,小于等于40岁。系信息表关系的属性有系号、系名、办公室、主任、电话共5个属性,所以元数是5。

### (3) 关系模式 ( Relation Schema )

在二维表中的行定义 ( 记录的型 ), 即对关系的描述称为关系模式。

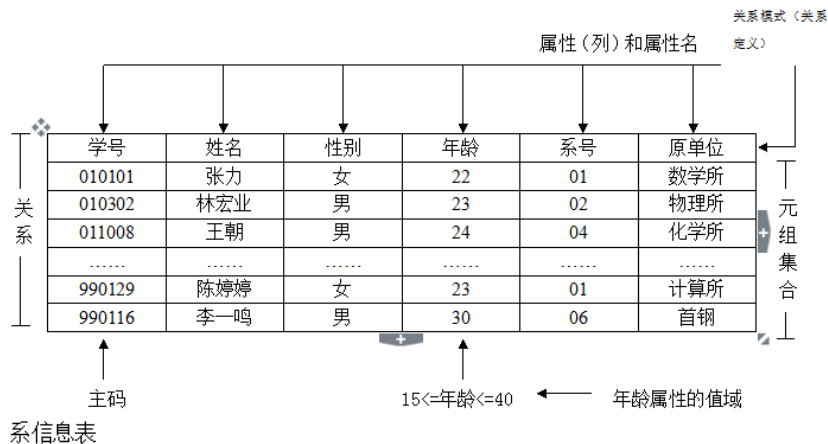
一般表示为:

关系名 ( 属性1, 属性2, ....., 属性n )

例如,图3-13中有两个关系模式,分别表示为:

学生登记表 ( 学号, 姓名, 性别, 年龄, 系号, 原单位 )

系信息表（系号，系名，办公室，主任，电话）



系信息表

系号	系名	办公室	主任	原单位
01	计算机	教 209	张立	301
02	物理	教 501	李记新	276
03	数学	教 410	王鸣利	346
04	化学	教 306	高明	417
.....	.....	.....	.....	.....
16	外语	教 701	陈钢	628

图3-13 关系数据模型的数据结构示例

#### （4）元组（Tuple）

在二维表中的一行（记录的值），称为一个元组。关系模式和元组的集合通称为关系。

例如，在学生登记表关系中的元组有（010101，张力，女，22，01，数学所），（010302，林宏业，男，23，02，物理所），（011008，王朝，男，24，04，化学所）等。

#### （5）分量（Component）

元组中的一个属性值。

例如，在学生登记表关系中元组（010101，张力，女，22，01，数学所）的每一个属性值010101，张力，女，22，01，数学所，都是它的分量。

#### （6）候选码（Candidate Key）或候选键

如果在一个关系中，存在多个属性（或属性组合）都能用来惟一标识该关系的元组，这些属性（或属性组合）都称为该关系的候选码或候选键。

例如，在学生登记表关系中，如果姓名不允许重名时，学号和姓名都是候选码。

#### （7）主码（Primary Key）或主键

在一个关系的若干个候选码中指定一个用来惟一标识该关系的候选码称为该关系的主码或主键。

例如，在学生登记表关系中，学号一般都是惟一的，如果姓名不允许重名时，存在两个候选码：学号和姓名，若选中学号作为惟一标识，那么，学号就是学生登记表关系的主码或主键。

#### （8）主属性（Primary Attribute）和非主属性（Nonprimary Attribute）

关系中包含在任何一个候选码中的属性称为主属性或码属性，不包含在任何一个候选码中的属性称为非主属性或非码属性。

例如，在学生登记表关系中，如果姓名不允许重名时，学号和姓名是主属性，其他属性是非主属性。

#### （9）外码（Foreign Key）或外键

当关系中的某个属性（或属性组）虽然不是该关系的主码或只是主码的一部分，但却是另一个

关系的主码时，称该属性（或属性组）为这个关系的外码。

例如，如果图3-13中的系信息表关系的主码是系号，那么，在学生登记表关系中的系号就是外码，因为它是另一个关系系信息表的主码。

（10）参照关系（Referencing Relation）与被参照关系（Referenced Relation）

参照关系也称为从关系，被参照关系也称为主关系，它们是以外码相关联的两个关系。以外码作为主码的关系称为被参照关系；外码所在的关系称为参照关系。由此可见，被参照关系与参照关系是通过外码相联系的，这种联系通常是1：n的联系。

例如，图3-13中的系信息表关系是被参照关系，而学生登记表关系是参照关系。它们通过外码“系号”相联系。

对于数据库要分清型（Type）和值（Value）的概念。数据库的型是指对数据库的结构和属性的说明，关系数据库的型即对关系数据库结构的描述，它包括若干关系模式。数据库的值是型的一个具体赋值，关系数据库的值是这些关系模式在某一时刻对应的关系的集合。所以，数据库的型亦称为数据库的内涵（Intension），数据库的值亦称为数据库的外延（Extensive）。数据库的型是稳定的，而数据库的值是随时间不断变化的，因为数据库中的数据在不断变更。

关系模型的数据结构表示为二维表，但不是任意的一个二维表都能表示一个关系。从上面的关系的形式定义可见关系数据库对关系是有限定的。

（1）每一个属性是不可分解的。

这是关系数据库对关系的最基本的一条限定，要求关系的每一个分量必须是一个不可分的数据项，也就是说，不允许表中还有表。

例如，图3-14中的两个关系就不符合要求。因为，在学生基本情况表关系中，属性成绩被分为英语、数学、数据库等多项，这相当于大表中还有一张小表（关于成绩的表）。在职工工资表关系中，属性工资和扣除都是可以再分解的。

学生基本情况表

序号	姓名	性别	年龄	系别	籍贯	年级	成绩		
							英语	数学	数据库
97001	刘红	女	18	管理	江苏	97	83	78	88
.....	.....	.....	.....	.....	.....	.....	.....	.....	.....
97500	陈列	男	19	计算机	北京	97	80	80	86

职工工资表

职工号	姓名	职称	工资			扣除		实发
			基本	补助	职务	房租	水电	
86051	李明	讲师	805	120	50	60	12	903
86052	王立刚	副教授	1000	150	90	100	125	1015
.....	.....	.....	.....	.....	.....	.....	.....	.....

图3-14 不满足关系数据库限定的关系

（2）每一个关系模式中属性的数据类型以及属性的个数是固定的，并且每个属性必须命名，在同一个关系模式中，属性名必须是不同的。

（3）每一个关系仅仅有一种记录类型，即一种关系模式。

（4）在关系中元组的顺序（即行序）是无关紧要的。

（5）在关系中属性的顺序可任意交换，交换时应连同属性名一起交换才行，否则顺序是错误



的。

(6) 同一个关系中不允许出现完全相同的元组。

版权方授权希赛网发布，侵权必究

上一节      本书简介      下一节

SQL语言

SQL ( Structured Query Language ) 称为结构化查询语言。SQL语言集数据查询 ( Data Query )、数据操纵 ( Data Manipulation )、数据定义 ( Data Definition ) 和数据控制 ( Data Control ) 功能于一体，充分体现了关系数据语言的特点和优点。其主要特点如下：

(1) 综合统一

SQL语言集数据定义语言 ( DDL )、数据操纵语言 ( DML )、数据控制语言 ( DCL ) 的功能于一体，语言风格统一，可以独立完成数据库生命周期中的全部活动，包括定义关系模式、录入数据以建立数据库、查询、更新、维护、数据库重构、数据库安全性控制等一系列操作的要求，这就为数据库应用系统开发提供了良好的环境。例如，用户在数据库投入运行后，还可以根据需要随时地逐步修改模式，而并不影响数据库的运行，从而使系统具有良好的可扩充性。

(2) 高度非过程化

非关系模型的数据库数据操纵语言是面向过程的语言，使用这样的语言进行数据操作，必须指定存取路径。而用SQL语言进行数据操作，用户只需提出“做什么”，而不必指明“怎么做”，因此用户无需了解存取路径，存取路径的选择以及SQL语句的操作过程由系统自动完成。这不但大大减轻了用户负担，而且有利于提高数据独立性。

(3) 面向集合的操作方式

非关系数据模型采用的是面向记录的操作方式，操作的对象都是一条记录（一次一个记录），需要指明如何用循环结构按照某条路径一条一条地把满足条件的记录读出来。而SQL语言采用集合操作方式，不仅查找结果可以是元组的集合，而且一次插入、删除、更新操作的对象也可以是元组的集合（一次一个集合）。

(4) 以同一种语法结构提供两种使用方式

SQL语言既是自含式语言，又是嵌入式语言。作为自含式语言，它能够独立地用于联机交互的使用方式，用户可以直接键入SQL命令对数据库进行操作。作为嵌入式语言，SQL语句能够嵌入到高级语言（例如C，COBOL，Fortran、PL/I）程序中，供程序员编写程序时使用。而在两种不同的使用方式下，SQL语言的语法结构基本上是一致的。这种以统一的语法结构提供两种不同的使用方式的作法，为用户使用提供了极大的灵活性与方便性。

表3-2 SQL语言的动词

SQL 功能	动词
数据查询	SELECT
数据定义	CREATE，DROP，ALTER
数据操纵	INSERT，UPDATE，DELETE
数据控制	GRANT，REVOKE

(5) 语言简洁，易学易用

SQL语言功能极强，但由于设计巧妙，语言十分简洁，完成数据定义、数据操纵、数据控制的核心功能只用了CREATE，DROP，ALTER，SELECT，INSERT，UPDATE，DELETE，GRANT，REVOKE等9个动词，如表3-2所示。

### 1 . SQL数据库的体系结构

SQL语言支持数据库3级模式结构，如图3-15所示。有些术语与传统的关系数据库术语不同。在SQL中，模式对应于“基本表（Base Table）”，内模式对应于“存储文件”，外模式对应于“视图（View）”和部分基本表。元组对应于表中的“行（Row）”，属性对应于表中的“列（Column）”。

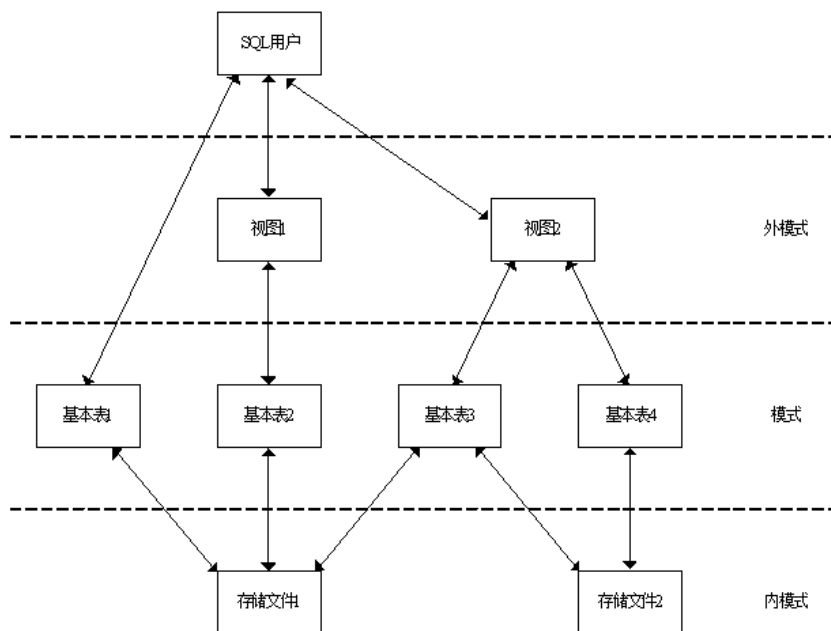


图3-15 SQL数据库的体系结构

SQL数据库体系结构特点如下：

- (1) 一个SQL数据库是表（Table）的汇集。
- (2) 一个SQL表由行集构成，一行是列的序列，每列对应一个数据项。
- (3) 一个表可以带若干索引，索引也存放在存储文件中。存储文件的逻辑结构组成了关系数据库的内模式。存储文件的物理结构是任意的，对用户是透明的。
- (4) 一个表或者是一个基本表，或者是一个视图。基本表是实际存储在数据库中的表；视图是从一个或几个基本表或其他视图导出的表。它本身不独立存储在数据库中，即数据库中只存放视图的定义而不存放视图对应的数据，这些数据仍存放在导出视图的基本表中，因此视图是一个虚表。视图在概念上与基本表是等同的，都是关系，用户可以在视图上再定义视图。
- (5) 一个基本表可以跨一个或多个存储文件存放，一个存储文件可存放一个或多个基本表。每个存储文件与外部存储器上一个物理文件对应。

(6) SQL用户可以是应用程序，也可以是终端用户。SQL的宿主语言有Fortran，COBOL，Pascal，PL/I、C和Ada等。SQL也能够作为独立的用户接口，供交互环境下的终端用户使用。

### 2 . SQL的数据定义

关系数据库由模式、外模式和内模式组成，即关系数据库的基本对象是表、视图和索引。因此SQL的数据定义功能包括定义表、定义视图和定义索引，如表3-3所示。由于视图是基于基本表的虚表，索引是依附于基本表的，因此SQL通常不提供修改视图定义和修改索引定义的操作。用户如果想

修改视图定义或索引定义，只能先将它们删除掉，然后再重建。不过有些关系数据库产品如Oracle允许直接修改视图定义。

表3-3 SQL的数据定义语句

操作对象	操作方式		
	创建	删除	修改
表	CREATE TABLE	DROP TABLE	ALTER TABLE
视图	CREATE VIEW	DROP VIEW	
索引	CREATE INDEX	DROP INDEX	

(1) 基本表

SQL语言使用CREATE TABLE语句创建基本表，其一般格式如下：

```
CREATE TABLE <表名> ( <列名> <数据类型> [列级完整性约束条件]
[ , <列名> <数据类型> [列级完整性约束条件]... ]
[ , <表级完整性约束条件> ] )
[其他参数];
```

其中，任选项“其他参数”是与物理存储有关的参数，随具体系统的不同而不同；<表名>是要创建的基本表的名字，它可以由一个或多个属性（列）组成。

定义表的各个属性时需要指明其<数据类型>。不同的数据库系统支持的数据类型不完全相同，实际使用时应根据具体数据库系统支持的数据类型声明。

建表的同时通常还可以定义与该表有关的完整性约束条件，这些完整性约束条件被存入系统的数据字典中，当用户对表进行操作时，由DBMS自动检查该操作是否违背所定义的完整性约束条件。声明完整性约束条件有两个层次（或称两个级别）：如果完整性约束条件涉及到该表的多个属性时，则必须在表级上定义，称为表级完整性约束条件；否则既可以在列级定义，也可以在表级定义，若在列级定义，称为列级完整性约束条件。

SQL支持空值的概念，空值是未知值，任何非主属性列可以有空值，除非在CREATE TABLE语句列的定义中指定了NOT NULL。

定义表的各个属性时必须指明其数据类型和长度。不同的数据库管理系统支持的数据类型不完全相同。具体的数据类型定义参照相应的数据库管理系统说明。

随着应用环境和应用需求的变化，有时需要修改已建立好的基本表，包括增加新列、增加新的完整性约束条件、修改原有的列定义或删除已有的完整性约束条件等。SQL语言用ALTER TABLE语句修改基本表，其一般格式为：

```
ALTER TABLE <表名>
[ADD <新列名> <数据类型> [完整性约束]]
[DROP <完整性约束名>]
[MODIFY <列名> <数据类型>];
```

其中<表名>为指定需要修改的基本表名，ADD子句用于增加新列和新的完整性约束条件，DROP子句用于删除指定的完整性约束条件，MODIFY子句用于修改原有的列定义。

不论基本表中原来是否已有数据，新增加的列一律为允许空值。

SQL没有提供删除属性列的语句，用户只能间接实现这一功能。即先将原表中要保留的列及其值复制到一个新表中，然后删除原表，再将新表重新命名为原表名。

当某个基本表不再需要时，可以使用SQL语句DROP TABLE对其进行删除。其一般格式为：

DROP TABLE <表名>;

基本表定义一旦被删除,表中的数据 and 在此表上建立的索引都将自动被删除掉,而建立在此表上的视图虽然仍然保留,但已经无法引用。因此执行删除基本表的操作一定要格外小心。

## (2) 索引

创建索引是加快表的查询速度的有效手段。SQL语言支持用户根据应用的需要,在基本表上建立一个或多个索引,以提供多种存取路径,加快查找速度。一般说来,创建与删除索引由数据库管理员(DBA)或表的属主(即建立表的人)负责完成。系统在存取数据时会自动选择合适的索引作为存取路径,用户不必也不能选择索引。

创建索引的一般格式为:

CREATE [UNIQUE][CLUSTER] INDEX <索引名>

ON <表名> (<列名>[<顺序>] [, <列名>[<顺序>]...)

其中,<表名>指定要建索引的基本表的名字。索引可以建在该表的一列或多列上,多列时各列名之间用逗号分隔。每个<列名>后面还可以用<顺序>指定索引值的排列顺序,包括ASC(升序)和DESC(降序)两种,缺省值为ASC。UNIQUE表示此索引的每一个索引值只对应惟一的数据记录。CLUSTER表示要建立的索引是聚簇索引。所谓聚簇索引是指索引项的顺序与表中记录的物理顺序一致的索引组织。

例如,执行下面的CREATE INDEX语句:

CREATE CLUSTER INDEX name-index ON emp ( name );

将会在emp表的name列上建立一个聚簇索引,而且emp表中的记录将按照name值的升序存放。

用户可以在最频繁查询的列上建立聚簇索引以提高查询效率。显然在一个基本表上最多只能建立一个聚簇索引。建立聚簇索引后,更新索引列数据时,往往导致表中记录的物理顺序的变更,代价较大,因此,对于经常更新的列不宜建立聚簇索引。

索引一经建立,就由系统使用和维护它,不需用户干预。创建索引是为了减少查询操作的时间,但如果数据增删改频繁,系统会花费许多时间来维护索引。这时,可以删除一些不必要的索引。

删除索引的一般格式为:

DROP INDEX <索引名>;

删除索引时,系统会同时从数据字典中删去有关该索引的描述。

## 3. SQL的数据操纵

SQL语言的数据操纵功能包括SELECT、INSERT、DELETE和UPDATE四个语句,即检索查询和修改(包括插入、删除、更新)两部分功能。

### (1) SQL的查询语句

数据库查询是数据库操作的核心。SQL语言提供了SELECT语句对数据库进行查询,该语句的一般格式是:

SELECT [ALL|DISTINCT] <目标列表达式>[, <目标列表达式>]...

FROM <基本表(或视图)>[, <基本表(或视图)>]...

[WHERE <条件表达式>]

[GROUP BY <列名>][HAVING <内部函数表达式>]]

[ORDER BY <列名2>[ASC|DESC]]；

整个语句的含义是：

根据WHERE子句的条件表达式，从基本表（或视图）中找出满足条件的元组。按SELECT子句中的目标列表表达式，选出元组中的属性值形成结果表。如果有ORDER子句，则结果表要根据指定的列名2按升序或降序排序。GROUP子句将结果按列名1分组，每个组产生结果表中的一个元组。通常在每组中作用集合函数，分组的附加条件用HAVING短语给出，只有满足内部函数表达式的组才能输出。

SQL语言对数据库的操作十分灵活方便，原因在于SELECT语句中的成分丰富多样，有许多可选形式，尤其是目标列和条件表达式。如包括简单查询、连接查询、嵌套查询、集合查询等。

## （2）SQL的修改语句

SQL的修改语句包括更新、删除和插入3类语句。

更新操作语句的一般格式为：

UPDATE <表名>

SET <列名>=<表达式>[，<列名>=<表达式>]...

[WHERE 谓词]

语句的含义是：

更新指定表中满足谓词的元组，把这些元组按SET子句中的表达式修改相应字段上的值。

删除语句的一般格式为：

DELETE FROM表名

[WHERE 谓词]

语句的含义是：

从指定表中删除满足谓词的那些记录，没有WHERE子句时表示删去此表中的全部记录，但此表的定义仍在数据字典中。DELETE语句删除的是表中的数据，而不是关于表的定义。

插入语句的一般格式有两种：

· 插入一个元组

INSERT INTO 表名[( 字段名[， 字段名]... )]

VALUES ( 常量[， 常量]... )；

· 插入子查询结果

INSERT INTO 表名[( 字段名[， 字段名]... )]

子查询；

第一种格式把一个新记录插入指定的表中；第二种格式把子查询的结果插入指定的表中。若表中有些字段在插入语句中没有出现，则这些字段上的值取空值NULL。当然在表定义中说明了NOT NULL的字段在插入时不能取NULL。若插入语句中没有指出字段名，则新记录必须在每个字段上均有值。

插入和删除与更新操作一样，都会引起完整性被破坏的问题。支持关系模型的系统应该自动地检查，对破坏完整性的插入操作拒绝执行。

## 4．视图

视图是关系数据库系统提供给用户以多种角度观察数据库中数据的重要机制。

视图是从一个或几个基本表（或其他视图）导出的表，它与基本表不同，是一个虚表。数据库

中只存放视图的定义，而不存放视图对应的数据，这些数据仍存放在原来的基本表中。基本表中的数据发生变化，从视图中查询出的数据也就随之改变了。

视图一经定义，就可以和基本表一样被查询、被删除，也可以在一个视图之上再定义新的视图，但对视图的修改（插入、删除、更新）操作则有一定的限制。

### （1）创建视图

创建视图语句的一般格式为：

```
CREATE VIEW <视图名>[( <列名>[, <列名>]...)]
```

```
AS <子查询>
```

```
[WITH CHECK OPTION];
```

其中，子查询可以是不含有ORDER BY子句和DISTINCT短语的任意的SELECT语句。

可选择项WITH CHECK OPTION表示当对视图进行UPDATE，INSERT和DELETE操作时，保证更新、插入或删除的行满足视图定义中的谓词条件（即子查询中的条件表达式）。

如果CREATE VIEW语句仅指定了视图名，省略了组成视图的各个属性列名，则隐含该视图由子查询中SELECT子句目标列中的诸字段组成。但在下列3种情况下必须明确指定组成视图的所有列名：

- 其中某个目标列不是单纯的属性名，而是集合函数或列表表达式。
- 多表连接时选出了几个同名列作为视图的字段。
- 需要在视图是为某个列启用新的更合适的名字（重新命名列名）。

需要说明的是，组成视图的属性列名必须依照上面的原则，或者全部省略或者全部指定。

### （2）删除视图

视图创建好后，若导出此视图的基本表被删除了，该视图将失效，但一般不会被自动删除。删除视图通常需要显式地使用DROP VIEW语句进行。

该语句的格式为：

```
DROP VIEW <视图名>;
```

一个视图被删除后，由该视图导出的其他视图也将失效，用户应该使用DROP VIEW语句将它们一一删除。

### （3）查询视图

通过视图进行查询时，首先要进行有效性检查，检查查询所涉及的表、视图等是否在数据库中存在。如果存在，则从数据字典中取出查询涉及的视图的定义，把定义中的子查询和用户对视图的查询结合起来，转换成对基本表的查询，然后再执行这个经过修正的查询。

把对视图的查询转换为对基本表的查询的过程称为视图的消解。视图可以和其他基本表一起使用。实现连接查询或嵌套查询。这也就是说，在关系数据库的3级模式结构中，外模式不仅包括视图，而且还可以包括一些基本表。

假定我们已经使用了下列语句建立了信息系学生的视图IS\_S：

```
CREATE VIEW IS_S AS SELECT s#, sname, age FROM S WHILE dept = "IS";
```

查询信息系选修1号课的学生的语句为：

```
SELECT s#, sname FROM IS_S WHILE S.s# = SC.s# AND c# = "1";
```

执行这个查询时，将转化为对基本表的查询：

```
SELECT s#, sname FROM S, SC WHILE S.s# = SC.s# AND c# = "1" AND dept = "IS";
```

#### （4）修改视图

修改视图包括插入（INSERT）、删除（DELETE）和更新（UPDATE）3类操作。

由于视图是虚表，因此对视图的更新最终要转换为对基本表的更新。

为防止用户通过视图对数据进行插入、删除和更新时，无意或故意操作不属于视图范围内的基本表数据，可在定义视图时加上WITH CHECK OPTION子句，这样在视图上进行修改数据时，DBMS会进一步检查视图定义中的条件，若不满足条件，则拒绝执行该操作。

与查询视图类似，DBMS执行此语句时，首先进行有效性检查。检查所涉及的表、视图等是否存在于数据库中。如果存在，则从数据字典中取出该语句涉及的视图的定义，把定义中的子查询和用户视图的更新操作结合起来，转换成对基本表的更新（插入和删除亦同），然后再执行这个经过修正的更新操作。

注意：在关系数据库中，并不是所有的视图都是可更新的，因为有些视图的更新不能惟一地有意义地转换成对相应基本表的更新。

一般对所有行列子集视图都可以执行修改和删除元组的操作，如果基本表中所有不允许空值的列都出现在视图中，则也可以对其执行插入操作。除行列子集视图外，还有些视图理论上是可以更新的，但它们的确切特征还是尚待研究的课题。另外还有些视图从理论上是不可更新的。

目前各个关系数据库系统一般都只允许对行列子集视图的更新，而且各个系统对视图的更新还有更进一步的规定，由于各系统在视图更新的实现方法上的差异，这些规定也不尽相同。

应该指出的是，不可更新的视图与不允许更新的视图是两个不同的概念。前者指理论上已证明其是不可更新的视图。后者指实际系统中不支持其更新，但它本身有可能是可更新的视图。

#### （5）视图的作用

视图是关系数据库系统提供给用户以多种角度观察数据库中数据的重要机制。视图最终是定义在基本表之上的，对视图的一切操作最终也要转换为对基本表的操作。合适地定义和合理地使用视图的优点有：

- 视图能够简化用户的操作。

视图机制使用户可以将注意力集中在他所关心的数据上。如果这些数据不是直接来自基本表，则可以通过定义视图，使用户眼中的数据库结构简单、清晰，并且可以简化用户的数据查询操作。例如，对于那些经常需要通过计算或要从若干张表连接来获得数据的查询，可将这类查询定义为一个视图，然后就可以容易地对该视图进行操作。

- 视图使用户能以多种角度观察同一数据。

视图机制能使不同的用户以不同的方式观察同一数据，当许多不同种类的用户使用同一个数据库时，这种灵活性是非常重要的。

- 视图对重构数据库提供了一定程度的逻辑独立性。

数据的逻辑独立性是指当数据库重新构造时，如增加新的关系或对原有关系增加新的字段等，用户和用户程序不会受影响。在关系数据库中，数据库的重构造往往是不可避免的。当然，视图只能在一定程度上提供数据的逻辑独立性，比如由于对视图的更新是有条件的，因此应用程序中修改数据的语句可能仍会因基本表结构的改变而改变。

- 视图能够对机密数据提供安全保护。

有了视图机制，就可以在设计数据库应用系统时，对不同的用户定义不同的视图，使机密数据不出现在不应看到这些数据的用户视图上，这样就由视图的机制自动提供了对机密数据的安全保护

功能。

版权方授权希赛网发布，侵权必究

[上一节](#)

### 第 3 章：数据库设计

### 典型真题解析

本节从历年考试真题中，精选出5道典型的试题进行分析，这5道试题所考查的知识点基本上覆盖了本章的所有内容，非常具有代表性。

版权方授权希赛网发布，侵权必究

## 十一节

第 3 章：数据库设计

### 例题1

某公司拟开发一多用户电子邮件客户端系统，部分功能的初步需求分析结果如下。

(1) 邮件客户端系统支持多个用户，用户信息主要包括用户名和用户密码，且系统中的用户名不可重复。

(2) 邮件账号信息包括邮件地址及其相应的密码, 一个用户可以拥有多个邮件地址 (如 user1@123.com)。

(3) 一个用户可拥有一个地址簿, 地址簿信息包括联系人编号、姓名、电话、单位地址、邮件地址1、邮件地址2, 邮件地址3等信息。地址簿中一个联系人只能属于一个用户, 且联系人编号唯一标识一个联系人。

(4) 一个邮件账号可以含有多封邮件，一封邮件可以含有多个附件。邮件主要包括邮件号、发件人地址、收件人地址、邮件状态、邮件主题、邮件内容、发送时间、接收时间。其中，邮件号在整个系统内唯一标识一封邮件，邮件状态有已接收、待发送、已发送和已删除4种，分别表示邮件是属于收件箱、发件箱、已发送箱和废件箱。一封邮件可以发送给多个用户。附件信息主要包括附件号、附件文件名、附件大小。一个附件只属于一封邮件，附件号仅在一封邮件内唯一。

【问题1】

根据以上说明设计的E-R图如图3-16所示，请指出地址簿与用户、电子邮件账号与邮件、邮件与附件之间的联系类型。

【问题2】

该邮件客户端系统的主要关系模式如下，请填补 (a) - (c) 的空缺部分。

用户（用户名，用户密码）

地址簿 ( a ), 联系人编号, 姓名, 电话, 单位地址, 邮件地址 1, 邮件地址 2, 邮件地址 3 )

邮件账号 ( 邮件地址, 邮件密码, 用户名 )



邮件（(b)），收件人地址，邮件状态，邮件主题，邮件内容，发送时间，接收时间）

附件（(c)），附件号，附件文件名，附件大小）

### 【问题3】

（1）请指出【问题2】中给出的地址簿、邮件和附件关系模式的主键，如果关系模式存在外键请指出。

（2）附件属于弱实体吗？请用50字以内的文字说明原因。

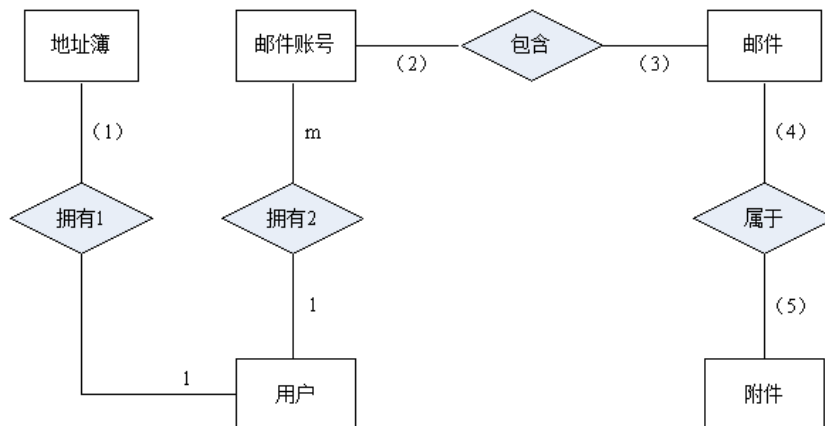


图3-16 电子邮件客户端系统的E-R图

### 例题1分析

该题是一个数据库设计题，题目以多用户电子邮件客户端系统为背景。考查E-R模型、E-R模型转换关系模式，求解主键和外键等知识点。值得注意的是，本题中出现了一个不常考的概念“弱实体”。

### 【问题1】

由说明的第（3）条可知，一个用户可拥有一个地址簿，一个地址簿只属于一个用户，所以地址簿和用户之间是一对一的关系；由说明中的第（4）条可知，一个邮件账号可以含有多封邮件，一封邮件可以含有多个附件，显然，电子邮件账号与邮件、邮件与附件之间的联系类型都是一对多。

### 【问题2】

考查关系模式的补充，这种题和问题1中的题目都属于送分题，只要仔细阅读题目说明，就非常容易找到答案。

由说明的第（3）条“地址簿信息包括联系人编号、姓名、电话、单位地址、邮件地址1、邮件地址2，邮件地址3等信息”可知地址簿所包含的属性数据。但是不要忽略了一个地址簿是属于某个用户的，所以这个关系模式中缺少的是用户信息，即用户名。

由说明的第（4）条“邮件主要包括邮件号、发件人地址、收件人地址、邮件状态、邮件主题、邮件内容、发送时间、接收时间”可知，在邮件这个关系模式中缺少的是邮件号。

由说明的第（4）条“附件信息主要包括附件号、附件文件名、附件大小”可知附件信息中所包含的属性数据。但是一个附件只属于一封邮件，所以附件关系模式中缺少的是标识邮件的邮件号。

### 【问题3】

（1）本题考查主键和外键的概念。主键也称为主码，是关系中的一个或一组属性，其值能够惟一标识一个元组。如果公共关键字在一个关系中是主关键字，那么这个公共关键字被称为另一个关系的外键。由此可见，外键表示了两个关系之间的联系。以另一个关系的外键作主关键字的表被称为主表，具有此外键的表被称为主表的从表。外键又称作外关键字。

由说明的第（3）条“地址簿中一个联系人只能属于一个用户，且联系人编号惟一标识一个联系

人”可知，联系人编号必定是地址簿的主键，但是不同用户的地址簿中可能有相同的联系人编号，所以地址簿的主键还应该加上用户名。

由说明的第（4）条“邮件号在整个系统内惟一标识一封邮件，邮件状态有已接收、待发送、已发送和已删除4种，分别表示邮件是属于收件箱、发件箱、已发送箱和废件箱”可知，邮件关系模式的主键是邮件号。

由说明的第（4）条“一个附件只属于一封邮件，附件号仅在一封邮件内惟一”，再由主键和外键的概念可知，附件关系模式的主键是附件号，外键是邮件号。

（2）本题考查弱实体的概念，知道弱实体的概念就能解答出该问题。弱实体是一种依赖联系：在现实世界中，有些实体对另一些实体有很强的依赖关系，即一个实体的存在必须以另一实体的存在为前提。前者就称为“弱实体”，如在人事管理系统中，职工子女的信息就是以职工的存在为前提的，子女实体是弱实体，子女与职工的联系是一种依赖联系。

在本题中，一个附件是属于一封邮件的，所以它是弱实体，依赖于邮件。

#### 例题1参考答案

##### 【问题1】

一对一，一对多，一对多

##### 【问题2】

（a）用户名

（b）邮件号

（c）邮件号

##### 【问题3】

（1）地址簿主键：用户名和联系人编号；邮件主键：邮件号；附件主键：附件号；外键：邮件号。

（2）附件属于弱实体。一个实体的键是由另一个实体的部分或全部属性构成，这样的实体叫做弱实体。附件的外键——邮件号是属于邮件这个实体的，所以它属于弱实体，依赖于邮件这个实体。

版权方授权希赛网发布，侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

## 例题2

某集团公司拥有多个大型连锁商场，公司需要构建一个数据库系统以方便管理其业务运作活动。

##### 【需求分析结果】

（1）商场需要记录的信息包括商场编号（编号惟一）、商场名称、地址和联系电话。某商场信息如表3-4所示。

表3-4 商场信息表

商场编号	商场名称	地址	联系电话
PS2101	淮海商场	淮海中路 918 号	021-64158818
PS2902	西大街商场	西大街时代盛典大厦	029-87283229
PS2903	东大街商场	碑林区东大街 239 号	029-87450287
PS2901	长安商场	雁塔区长安中路 38 号	029-85264950

(2) 每个商场包含不同的部门，部门需要记录的信息包括部门编号（集团公司分配）、部门名称、位置分布和联系电话。某商场的部门信息如表3-5所示。

表3-5 部门信息表

部门编号	部门名称	位置分布	联系电话
DT002	财务部	商场大楼 6 层	82504342
DT007	后勤部	商场地下副一层	82504347
DT021	安保部	商场地下副一层	82504358
DT005	人事部	商场大楼 6 层	82504446
DT004	管理部	商场裙楼 3 层	82504668

(3) 每个部门雇佣多名员工处理日常事务，每名员工只能隶属一个部门（新进员工在培训期不隶属于任何部门）。员工需要记录的信息包括员工编号（集团公司分配）、姓名、岗位、电话号码和工资。员工信息如表3-6所示。

表3-6 员工信息表

员工编号	姓名	岗位	电话号码	工资
XA3310	周超	理货员	13609257638	1500.00
SH1075	刘飞	防损员	13477293487	1500.00
XA0048	江雪花	广播员	15234567893	1428.00
BJ3123	张正华	部门主管	13345698432	1876.00

(4) 每个部门的员工中有一名是经理，每个经理只能管理一个部门，系统需要记录每个经理的任职时间。

【概念模型设计】

根据需求阶段搜集的信息，设计实体联系图（如图3-17）和关系模式（不完整）。

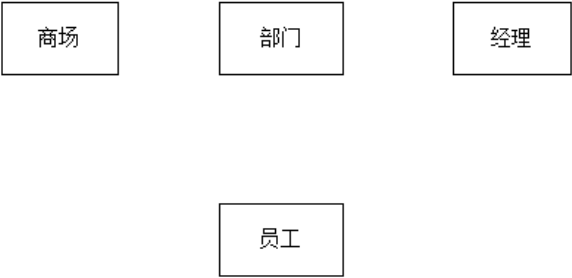


图3-17 实体联系图

【关系模式设计】

商场（商场编号，商场名称，地址，联系电话）

部门（部门编号，部门名称，位置分布，联系电话，(a)）

员工（员工编号，员工姓名，岗位，电话号码，工资，(b)）

经理（(c)，任职时间）

【问题1】

根据问题描述，补充4个联系，完善图3-17所示的实体联系图。联系名可用联系1、联系2、联系3和联系4代替，联系的类型分为1：1、1：n和m：n。

【问题2】

根据实体联系图，将关系模式中的空（a）-（c）补充完整，并分别给出部门、员工和经理 关系模式的主键和外键。

【问题3】

为了使商场有紧急事务时能联系到轮休的员工，要求每位员工必须且只能登记一位紧急联系人的姓名和联系电话，不同的员工可以登记相同的紧急联系人。则在图3-17中还需添加的实体是（1），该实体和图3-17中的员工存在（2）联系（填写联系类型）。给出该实体的关系模式。

例题2分析

本题考查数据库的基本知识，如E-R图的画法等。

【问题1】

本题要求考生根据问题的描述，补充4个联系，完善图3-17的实体联系图。根据题目的描述“每个商场包含不同的部门”、“每个部门雇佣了多名员工处理日常事务，每名员工只能属于一个部门”、“每个部门的员工中有一个是经理，每个经理只能管理一个部门”可知，需要补充的4个联系是商场和部门之间、部门和员工之间、部门和经理之间及员工和经理之间的关系。显然，一个商场对应多个部门（一对多），一个部门有多个员工（一对多），一个部门对应一个经理（一对一），每个员工只有一个经理（一对一）。

【问题2】

本题考查关系模式设计的相关知识，需仔细分析需求分析结果来解题。

由需求分析结果第（2）条或者部门信息表可知，部门需要记录的信息包括部门编号（集团公司分配）、部门名称、位置分布和联系电话。但是在本题中设置部门是为了服务商场的，所以必须记录其对应的商场。所以部门关系缺少的属性是“商场编号”。主键为部门编号，外键是商场编号。

由需求分析结果第（3）条或者员工信息表可知，每名员工只能隶属于一个部门（新进员工在培训期不隶属于任何部门）。员工需要记录的信息包括员工编号（集团公司分配）、姓名、岗位、电话号码和工资。培训期外的员工必然是归属于某个部门的，所以在此缺少的是员工所在的部门编号。主键是员工编号，外键是其所在部门的编号。

经理的情况比较特殊，首先他是员工，所以必须记录其员工的编号；其次是他对应一个部门，所以记录中必须有一个部门编号，但是每个员工都对应了一个部门，即知道经理的员工编号，就知道了经理的部门。

【问题3】

题目已经说得很明白了，为了使商场有紧急事务时能联系到轮休的员工，要求每位员工必须且只能登记一位紧急联系人的姓名和联系电话，不同的员工可以登记相同的紧急联系人。所以需要添加的实体必定是紧急联系人，并且不同的员工可以对应同一个紧急联系人，所以紧急联系人和员工的关系是一对多，其关系模式中必然有姓名和联系电话，但是为了避免重名造成的麻烦还要有联系人的编号来作为主键。

例题2参考答案

【问题1】完整的实体联系图如图3-18所示。

联系人	关系实体	联系类型
联系 1	商场与部门	1: n
联系 2	部门与员工	1: n
联系 3	部门与经理	1: 1
联系 4	员工与经理	1: 1

【问题2】

（a）所在商场编号

（b）所在部门编号

（c）员工编号

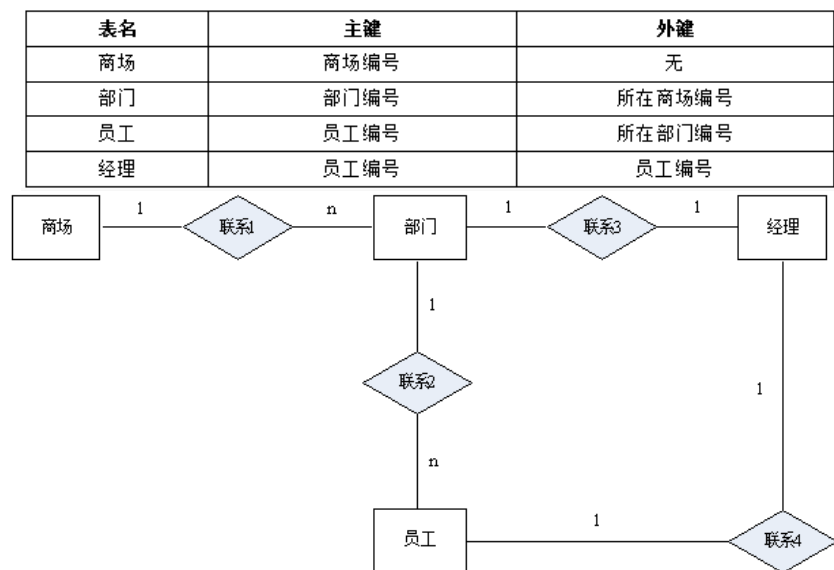


图3-18 完整的实体联系图

### 【问题3】

(1) 紧急联系人

(2) 1:n

关系模式为：紧急联系人（紧急联系人编号，姓名，联系电话）。

版权方授权希赛网发布，侵权必究

上一节

本书简介

下一节

## 例题3

某宾馆拟开发一个宾馆客房预订子系统，主要是针对客房的预订和入住等情况进行管理。

### 【需求分析结果】

(1) 员工信息主要包括员工号、姓名、出生年月、性别、部门、岗位、住址、联系电话和密码等信息。岗位有管理和服务两种。岗位为“管理”的员工可以更改（添加、删除和修改员工表中的本部门员工的岗位和密码，要求将每一次更改前的信息保留；岗位为“服务”的员工只能修改员工表中本人的密码，且负责多个客房的清理等工作。

(2) 部门信息主要包括部门号、部门名称、部门负责人、电话等信息；一个员工只能属于一个部门，一个部门只有一位负责人。

(3) 客房信息包括客房号、类型、价格、状态等信息。其中类型是指单人间、三人间、普通标准间、豪华标准间等；状态是指空闲、入住和维修。

(4) 客户信息包括身份证号、姓名、性别、单位和联系电话。

(5) 客房预订情况包括客房号、预订日期、预订入住日期、预订入住天数、身份证号等信息。一条预订信息必须且仅对应一位客户，但一位客户可以有多条预订信息。

### 【概念模型设计】

根据需求阶段收集的信息，设计的实体联系图（不完整）如图3-19所示。

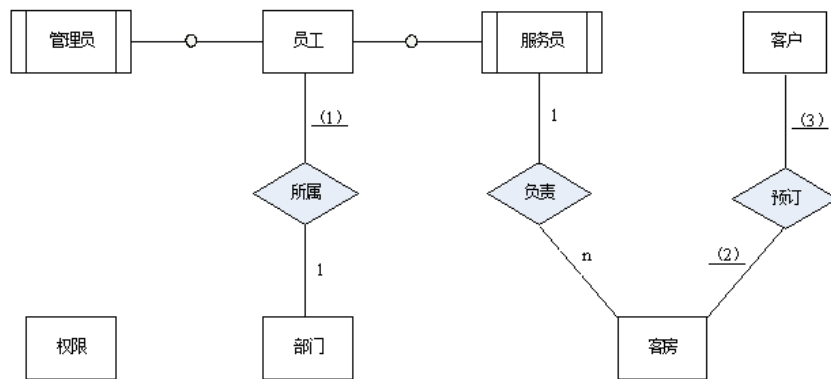


图3-19 实体联系图

#### 【逻辑结构设计】

逻辑结构设计阶段设计的部分关系模式（不完整）如下。

员工（(4)，姓名，出生年月，性别，岗位，住址，联系电话，密码）

权限（岗位，操作权限）

部门（部门号，部门名称，部门负责人，电话）

客房（(5)，类型，价格，状态，入住日期，入住时间，员工号）

客户（(6)，姓名，性别，单位，联系电话）

更改权限（员工号，(7)，密码，更改日期，更改时间，管理员号）

预订情况（(8)，预订日期，预订入住日期，预订入住天数）

#### 【问题1】

根据问题描述，填写图3-19中的（1）-（3）处联系的类型。

#### 【问题2】

补充图3-19中的联系并指明其联系类型。

#### 【问题3】

根据需求分析结果和图3-19所示，将逻辑结构设计阶段生成的关系模式中的空（4）-（8）补充完整。（注：一个空可能需要填多个属性）

#### 【问题4】

若去掉权限表，并将权限表中的操作权限属性放在员工表中（仍保持管理和服务岗位的操作权限规定），则与原有设计相比有什么优缺点（请从数据库设计的角度进行说明）？

#### 例题3分析

本题考查数据库设计。涉及的考点有概念模型设计（E-R图的补充）和逻辑模型设计。下面具体分析试题。

#### 【问题1】

（1）按常规来说，一个员工只能属于一个部门，一个部门只有一个负责人，所以部门与员工之间的关系是一对多的关系，所以（1）应该填写n。

（2）由于一条预订信息必须仅对应一个客户，但一个客户可以有多个预订信息，所以客户与预订信息之间是一对多的关系。需要注意：题目要求的是客户与客房之间的预订信息，一位客户可以预订多个客房，而一个客房在不同的时间也可以被多个客户预订，所以客户与客房的预订关系是多对多的。故（2）和（3）都应填写n。

#### 【问题2】

由图可知需要增加的是员工与权限的关系，因为“管理员”和“服务员”都属于“员工”。一

类员工（比如服务员A、服务员B、……、服务员N）使用同一权限，所以员工与权限之间是多对一的关系。

**【问题3】**

（1）由需求分析结果第（1）条可知，员工信息主要包括员工号、姓名、出生年月、性别、部门、岗位、住址、联系电话和密码等信息。即员工信息包括员工本身的信息和他所在的部门信息，员工本身最具代表性的信息就是员工号，而部门在该系统中是一个关系，所以在此处要记录部门的相关信息，只需记录部门号即可，其余的相关信息可以通过部门号查询来获得。

（2）由需求分析结果第（3）条可知，客房信息包括客房号、类型、价格、状态等信息。显然（5）空应填写：客房号。

（3）由需求分析结果第（4）条可知，客户信息包括身份证号、姓名、性别、单位和联系电话。显然（6）空应填写：身份证号。

（4）岗位有管理和服务两种，岗位为“管理”的员工可以更改（添加、删除和修改）员工表中的本部门员工的岗位和密码，要求将每一次更改前的信息保留。所以“更改权限”这个关系模式是指岗位为“管理”的员工可以更改员工表中本部门员工的岗位和密码。“更改前的信息”包括该员工所涉及的全部信息。该关系中已经记录了“员工号”，从员工号可查询获得该员工的所有个人信息和部门信息，同时记录了员工的密码及本次修改的时间、操作和管理员。仔细观察不难发现，该关系中惟一缺少的是岗位的信息，而本系统的设计是由岗位确定该员工的权限的，因此（7）空应填写：岗位。

（5）由需求分析结果第（5）条可知，客房预订情况包括客房号、预订日期、预订入住日期、预订入住天数、身份证号等信息。显然（8）空应填写：客房号、身份证号。

**【问题4】**

本题考查考生对数据库规范化的理解。

去掉权限表后的缺点：去掉权限表后，权限字段就得添加到员工表中，员工表中有很多员工记录，而同一类员工的权限都相同，权限数据却要多次重复存储，显然有大量的数据冗余。同时，此时若要对权限字段进行更新，很有可能产生更新异常，若某一岗位的员工全部离职，将导致权限数据的丢失（删除异常）。

去掉权限表的优点：获取某一员工权限数据时，不必再将员工表与权限表进行连接查询，可以提高存储速度。

**例题3参考答案**

**【问题1】**

（1）n或m或\*

（2）n或m或\*

（3）n或m或\*

**【问题2】**

员工到权限的联系，联系类型m：1。

**【问题3】**

（4）员工号，部门号

（5）客房号

（6）身份证号

- (7) 岗位
- (8) 客房号, 身份证号

【问题4】

优点：如果合为一个表，可以只查一次表就能得出岗位和操作权限信息，加快了查找速度。

缺点：如果合为一个表，则岗位、操作权限将多次重复出现，会产生冗余数据和增加数据库存储量。

版权方授权希赛网发布，侵权必究

上一节      本书简介      下一节

例题4

某地区举行篮球比赛，需要开发一个比赛信息管理系统来记录比赛的相关信息。

【需求分析结果】

- (1) 登记参赛球队的信息。记录球队的名称、代表地区、成立时间等信息。系统记录球队每个队员的姓名、年龄、身高、体重等信息。每个球队有一个教练负责管理球队，一个教练仅负责一个球队。系统记录教练的姓名、年龄等信息。
- (2) 安排球队的训练信息。比赛组织者的球队提供了若干个场地，供球队进行适应性训练。系统记录现有的场地信息，包括场地名称、场地规模、位置等信息。系统可为每个球队安排不同的训练场地，如表3-7所示。系统记录训练场地安排的信息。
- (3) 安排比赛。该赛事聘请专职裁判，每场比赛只安排一个裁判。系统记录裁判的姓名、年龄、级别等信息。系统按照一定的规则，首先分组，然后根据球队、场地和裁判情况安排比赛（每场比赛的对阵双方分别称为甲队和乙队）。记录参赛球队名称、比赛时间、比分、比赛场地等信息，如表3-8所示。
- (4) 所有球员、教练和裁判可能在表中出现重名情况。

表3-7 训练安排表

球队名称	场地名称	训练时间
解放军	一号球场	2008-06-09 14: 00-18: 00
解放军	一号球场	2008-06-12 09: 00-12: 00
解放军	二号球场	2008-06-11 14: 00-18: 00
山西	一号球场	2008-06-10 09: 00-12: 00

表3-8 比赛安排表

A 组				
甲队-乙队	场地名称	比赛时间	裁判	比分
解放军-北京	一号球场	2008-06-17 15: 00	李大明	
天津-山西	一号球场	2008-06-17 17: 00	胡学海	
B 组				
甲队-乙队	场地名称	比赛时间	裁判	比分
上海-安徽	二号球场	2008-06-17 15: 00	丁鸿平	
山东-辽宁	二号球场	2008-06-17 19: 00	郭爱琪	

【概念模型设计】

根据需求阶段收集的信息，设计的实体联系图和关系模式（不完整）如图3-20所示。



(1) 实体联系图

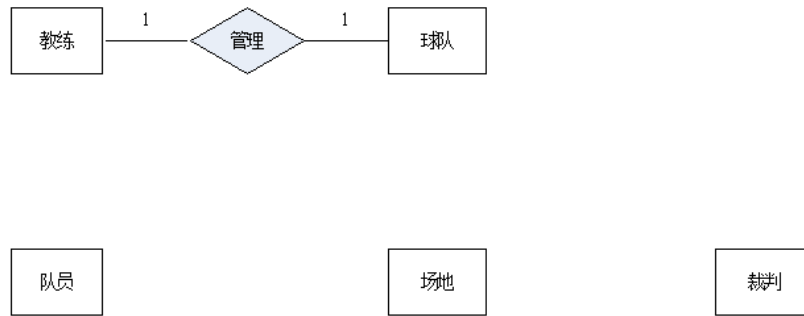


图3-20 实体联系图

(2) 关系模式

教练 (教练编号, 姓名, 年龄)

队员 (队员编号, 姓名, 年龄, 身高, 体重, (a))

球队 (球队名称, 代表地区, 成立时间, (b))

场地 (场地名称, 场地规模, 位置)

训练记录 ((c))

裁判 (裁判编号, 姓名, 年龄, 级别)

比赛记录 ((d))

【问题1】

根据问题描述, 补充联系及其类型, 完善图3-20所示的实体联系图。

【问题2】

根据实体联系图3-20, 填充关系模式中的 (a)、(b)、(c) 和 (d), 并给出训练记录和比赛记录关系模式的主键和外键。

【问题3】

如果考虑记录一些特别资深的热心球迷的情况, 每个热心球迷可能支持多个球队。热心球迷包括姓名、住址和喜欢的俱乐部等基本信息。根据这一要求修改图3-20所示的实体联系图, 给出修改后的关系模式。(仅给出增加的关系模式描述)

例题4分析

本题考查数据库设计, 设计考点有数据库的概念结构设计和逻辑结构设计。

【问题1】

问题1考查的是数据库的概念结构设计, 题目要求补充完整实体联系图中的联系和联系的类型。

根据题目的需求描述可知:

一个球队必然有多名队员, 所以球队和队员之间的关系是一对多。

一个球队可到多个场地进行比赛, 同时一个场地也可以给多个球队来比赛, 所以球队和场地之间的关系是多对多; 由于球队和场地之间的关系有“比赛”和“训练”, 所以本图中在它们之间设置了两个关系。

一个球队在参加比赛时可以任意聘请裁判, 每场比赛只有一个裁判; 然后同时一个裁判可以给多个球队的比赛做裁定, 因此球队和裁判之间的关系是多对多。

【问题2】

本题要求将各关系模式补充完整, 并给出各关系模式的主键, 考查的是数据库的逻辑结构设计。考生仔细分析题目中的需求分析结果, 不难得出此题答案。

根据实体联系图和表3-7、表3-8中的数据可知：

（1）记录中已有队员的基本信息。另外，一个队员必然是属于某个球队的，所以（a）空填写：球队名称。

（2）需求分析结果已经说明，每个球队有一个教练负责管理球队，所以（b）空中填写：教练编号。

（3）由需求分析结果可知，比赛组织者为球队提供若干个场地，供球队进行适应性训练。系统可为每个球队安排不同的训练场地。由训练安排表可以很容易地判断出训练记录应有的属性为：球队名称、场地名称、开始时间、结束时间。

主键表示该记录的惟一性。不同时间段有不同的球队在不同的场地训练，所以训练时间的不同能够区分不同的记录，因此“训练记录”关系模式的主键是：球队，开始时间；或球队，结束时间；或场地名称，开始时间；或场地名称，结束时间。它的外键是：场地名称，球队名称。

（4）同样，由比赛安排表可知，比赛记录的属性有甲队、乙队、比赛时间、球场名称、比分、裁判和分组。

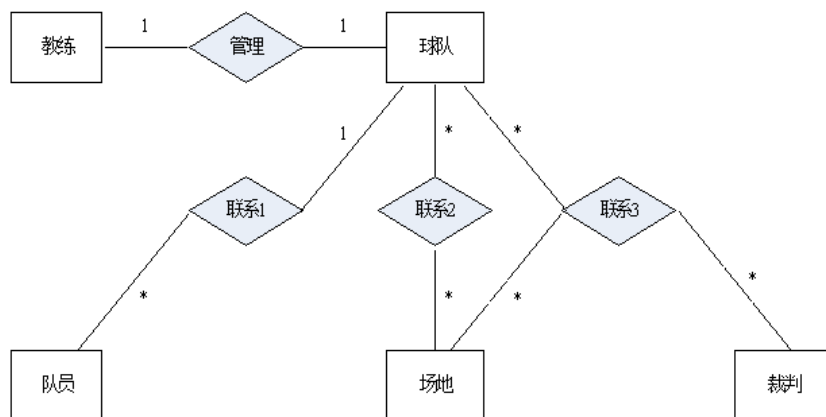
一支球队在某个时间进行一场比赛，可以确定一条比赛记录，所以可以用（甲队，比赛时间）或（乙队，比赛时间）来做比赛记录关系模式的主键；而一个场地在某个时间进行一场比赛，也可以确定一条比赛记录，所以还可以用（场地名称，比赛时间）来做比赛记录关系模式的主键；某个时间的一场比赛由一个裁判来执行，也可以确定一条比赛记录，所以也可以用（裁判，比赛时间）来做比赛记录关系模式的主键。它的外键是：甲队，乙队，场地名称，裁判。

#### 【问题3】

此题考查数据库概念结构设计的实体联系图的修改，新增加一个实体——热心球迷。由于每个热心球迷可能支持多个球队，当然一个球队会有许多个热心球迷在支持，所以热心球迷和球队之间的关系是多对多。题目中已经指出热心球迷包括姓名、住址和喜欢的俱乐部等基本信息，但是由于球迷之间可能有重名现象，因此还需要增加一个“球迷编号”属性，可得到关系模式：热心球迷（球迷编号，姓名，住址，俱乐部）；支持球队（球迷编号，球队）。

#### 例题4参考答案

##### 【问题1】



完整的实体联系图如上所示。

##### 【问题2】

（a）球队名称

（b）教练编号

（c）球队名称，场地名称，开始时间，结束时间

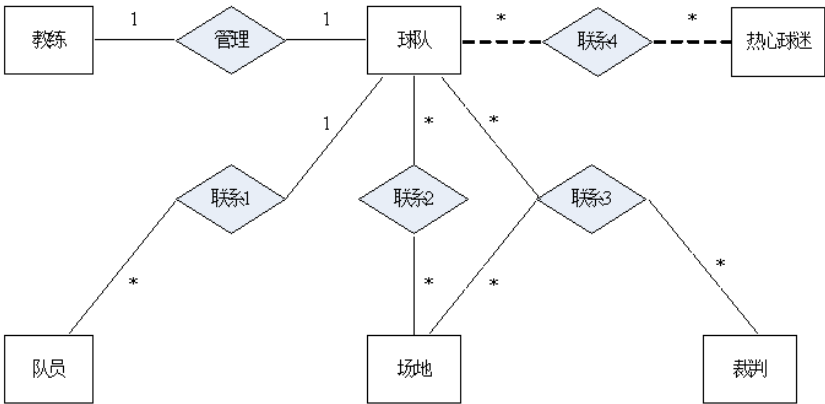
(d) 甲队，乙队，比赛时间，球场名称，比分，裁判，分组

训练记录和比赛记录关系模式的主键和外键如下表所示：

训练记录	主键	球队，开始时间；或球队，结束时间；或场地名称，开始时间；或场地名称，结束时间
	外键	球队名称，场地名称
比赛记录	主键	甲队，比赛时间；或场地名称，比赛时间；或裁判，比赛时间；或乙队，比赛时间
	外键	甲队，乙队，场地名称，裁判

【问题3】

修改后的实体联系图如下图所示。



关系模式：

热心球迷（球迷编号，姓名，住址，俱乐部）

支持球队（球迷编号，球队）

版权方授权希赛网发布，侵权必究

上一节      本书简介      下一节

例题5

某汽车维修站拟开发一套小型汽车维修管理系统，对车辆的维修情况进行管理。

【需求分析】

(1) 对于新客户及车辆，汽车维修管理系统首先登记客户信息，包括客户编号、客户名称、客户性质（个人、单位）、折扣率、联系人、联系电话等信息；还要记录客户的车辆信息，包括车牌号、车型、颜色等信息。一个客户至少有一台车。客户及车辆信息如表3-9所示。

表3-9 客户及车辆信息

客户编号	GS0051	客户名称	**公司	客户性质	单位
折扣率	95%	联系人	杨浩东	联系电话	82638779
车牌号		颜色	车型	车辆类别	
**0765		白色	帕萨特	微型车	

(2) 记录维修车辆的故障信息，包括维修类型（普通、加急）、作业分类（大、中、小修）、结算方式（自付、三包、索赔）等信息。维修厂的员工分为维修员和业务员。车辆维修首先委托给

业务员。业务员对车辆进行检查和故障分析后，与客户磋商，确定故障现象，生成维修委托书，如表3-10所示。

（3）维修车间根据维修委托书和车辆故障现象，在已有维修项目中选择并确定一个或多个具体维修项目，安排相关的维修工及工时，生成维修派工单。维修派工单如表3-11所示。

表3-10 维修委托书

NO.20070702003			登记日期：2007-07-02		
车牌号	**0765	客户编号	GS0051	维修类型	普通
作业分类	中修	结算方式	自付	进场时间	20070702 11: 09
业务员	张小江	业务员编号	012	预计完工时间	
故障描述					
车头损坏，水箱漏水					

表3-11 维修派工单

NO.20070702003				
维修项目编号	维修项目	工时	维修员编号	维修员工种
012	维修车头	5.00	012	机修
012	维修车头	2.00	023	漆工
015	水箱焊接补漏	1.00	006	焊工
017	更换车灯	1.00	012	机修

（4）客户车辆在车间修理完毕后，根据维修项目单价和维修派工单中的工时计算车辆此次维修的总费用，并记录在委托书中。

【概念模型设计】

根据需求阶段收集的信息，设计的实体联系图和关系模式（不完整）如下所示。图3-21所示的业务员和维修工是员工的子实体。

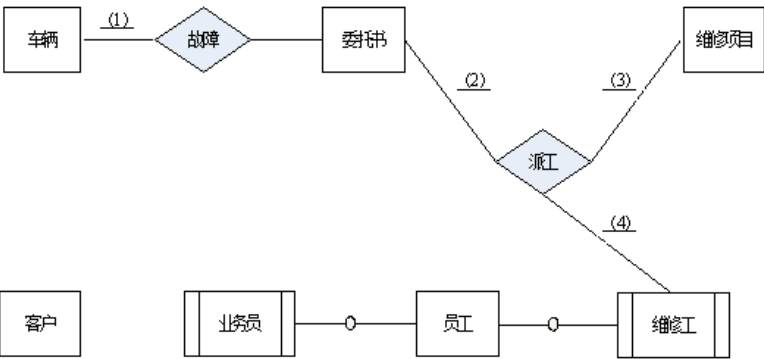


图3-21 实体联系图

【逻辑结构设计】

- 客户（5），折扣率，联系人，联系电话）
  - 车辆（车牌号，客户编号，车型，颜色，车辆类别）
  - 委托书（6），维修类型，作业分类，结算方式，进厂时间，预计完工时间，登记日期，故障描述，总费用）
  - 维修项目（维修项目编号，维修项目，单价）
  - 派工单（7），工时）
  - 员工（8），工种，员工类型，级别）
- 【问题1】
- 根据问题描述，填写图3-21中（1）-（4）处联系的类型。联系类型分为一对一、一对多和多对多3种，分别使用1：1、1：n或1：\*、m：n或\*：\*表示。
- 【问题2】
- 补充图3-21所示的联系并指明其联系类型。联系名可为联系1，联系2，.....。

**【问题3】**

根据图3-21和说明，将逻辑结构设计阶段生成的关系模式中的空（5）-（8）补充完整。

**【问题4】**

根据问题描述，写出客户、委托书和派工单这3个关系的主键。

**例题5分析**

本题考查数据库设计，设计考点有数据库的概念结构设计和逻辑结构设计。

**【问题1】**

由维修委托书的故障描述、维修类型、作业分类可知，一台车可能有多个故障，对应多个维修委托书，所以（1）空填写：\*；根据题目中“维修车间根据维修委托书和车辆的故障现象，在已有的维修项目中选择并确定一个或多个具体维修项目，安排相关的维修工及工时，生成维修派工单”的说明，很明显，一份委托书包含了一个或多个维修项目，而每个维修项目可以由多个维修工来完成，但一个维修工只可以完成一个维修项目，所以（2）空填写：1，（3）、（4）填写：\*。

**【问题2】**

需要补充车辆和客户之间以及委托书和业务员之间的关系。

由题目“一个客户至少拥有一台车”可知，客户和车辆之间是“拥有”关系，且是一对多的关系；在由题目中“业务员对车辆进行检查和故障分析后，与客户磋商，确定故障现象，生成维修委托书”可知，业务员与委托书之间是“委托”关系，且一名业务员可以受理多份委托书，而一份委托书由一名业务员来生成。

**【问题3】**

本题又是补充逻辑结构设计题，几乎每年都考，这类题目只要仔细看需求分析结果或者仔细观察题目中已知的表，很容易就能做出，关键是需要细心，不要漏掉什么属性。

根据客户和车辆信息表可知，客户关系应包括客户编号、客户名称、客户性质、折扣率、联系人等属性，主键显然为客户编号；而车辆关系应包括车牌号、客户编号、车型、颜色、车辆类别等属性，主键为车牌号。

根据维修委托书表可知委托书应包括委托书编号、车牌号、客户编号、业务员编号、维修类型等属性，其主键为委托书编号。

根据维修派工单可知，派工单应包括委托书编号、维修项目编号、维修工编号、工时等属性，主键是：委托书编号、维修项目编号和维修员编号。

根据实体联系图可知，员工包括业务员和维修工，他们共有的属性是员工编号、员工姓名、工种、员工类型、级别等属性，主键为员工编号。

**【问题4】**

本题的分析参考问题3的分析。

**例题5参考答案**

**【问题1】**

（1）n或m或\*

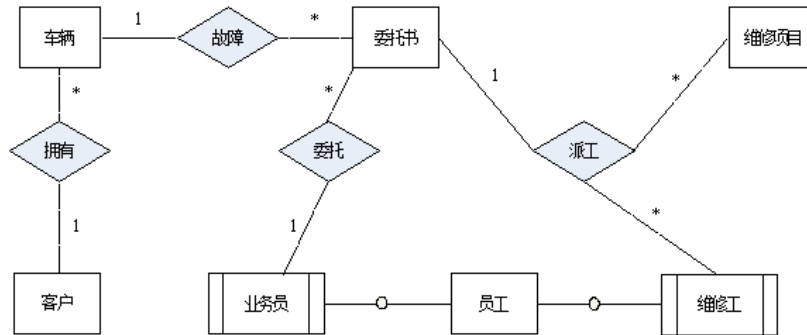
（2）1

（3）n或m或\*

（4）n或m或\*

**【问题2】**

完整的实体联系图如下图所示：



### 【问题3】

(5) 客户编号, 客户名称, 客户性质

(6) 委托书编号, 客户编号, 车牌号, 业务员编号 或者

委托书编号, 车牌号, 业务员编号

(7) 委托书编号, 维修工编号, 维修项目编号

(8) 员工编号, 员工姓名

#### 【问题4】

客户：客户编号

委托书：委托书编号

派工单：委托书编号，维修项目编号，维修工编号

版权方授权希赛网发布，侵权必究

上一节

## 本书简介

下一节

## 第 3 章：数据库设计

作者：希赛教育软考学院 来源：希赛网 2014年05月05日

## 考前必练

为了帮助读者进行考前训练，本节给出5道典型的试题，以及这些试题的分析与解答。请读者独立完成这些练习题，然后再去阅读试题分析与解答。根据自己所做试题的情况，查漏补缺。

版权方授权希赛网发布，侵权必究

上一节

## 本书简介

下一节

### 第 3 章：数据库设计

作者：希赛教育软考学院 来源：希赛网 2014年05月05日

## 试题1

## 考前必做的练习题

某医院的门诊管理系统实现了为患者提供挂号、处方药品收费的功能。具体的需求及设计如下。

### 【需求分析结果】

(1) 医院医师具有编号、姓名、科室、职称、出诊类型和出诊费用，其中出诊类型分为专家门诊和普通门诊，与医师职称无关；各个医师可以具有不同的出诊费用，与职称和出诊类型无关。

(2) 患者首先在门诊挂号处挂号，选择科室和医师，根据所选择的医师缴纳挂号费（医师出诊费）。收银员为患者生成挂号单，如表3-12所示，其中，就诊类型为医师的出诊类型。

(3) 患者在医师处就诊后，凭借挂号单和医师手写处方到门诊药房交费买药。收银员根据就诊号和医师处方中开列的药品信息，查询药品库（如表3-13所示），生成门诊处方单（如表3-14所示）。

(4) 由于药品价格会发生变化，因此门诊管理系统必须记录处方单上药品的单价。

**表3-12 \*\*医院门诊挂号单**

收银员：13011			时间：2007年2月1日 08:58		
就诊号	姓名	科室	医师	就诊类型	挂号费
20070205015	叶萌	内科	杨玉明	专家门诊	5元

**表3-13 药品库**

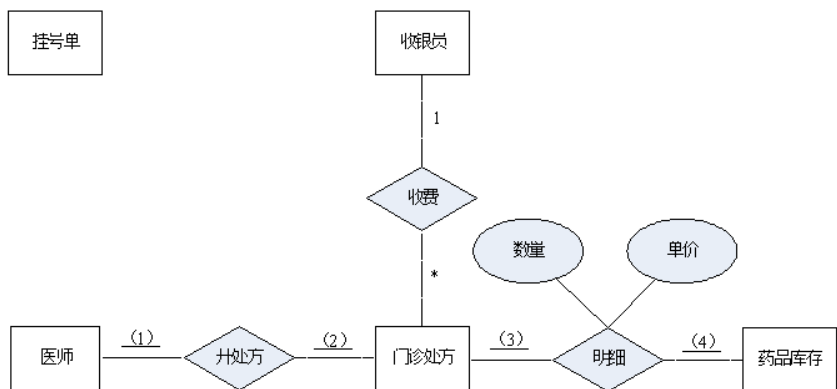
药品编码	药品名称	类型	库存	货架编号	单位	规格	单价
12007	牛蒡子	中药	51590	B1410	G	炒	0.0340
11090	百部	中药	36950	B1523	G	片	0.0313

**表3-14 \*\*医院门诊处方单**

时间：2007年2月1日 10:31					
就诊号	20070205015	病人名称	叶萌	医师姓名	杨玉明
金额总计	0.65	项目总计	2	收银员	21081
药品编码	药品名称	数量	单位	单价	金额（元）
12007	牛蒡子	10	G	0.0340	0.34
11090	百部	10	G	0.0313	0.31

#### 【概念模型设计】

根据需求阶段收集的信息，设计的实体联系图和关系模式（不完整）如图3-22所示。



**图3-22 实体联系图**

#### 【逻辑结构设计】

根据概念模型设计的结果，设计关系模式如下：

挂号单（就诊号，病患姓名，医师编号，时间，(5)）

收银员（编号，姓名，级别）

医师（编号，姓名，科室，职称，出诊类型，出诊费用）

门诊处方（(6)，收银员，时间）

处方明细（就诊号，(7)）

药品库（药品编码，药品名称，（8））

【问题1】

根据问题描述，填写图3-22所示的实体联系图中（1）-（4）处联系的类型。

【问题2】

图3-22中还缺少几个联系？请指出每个联系两端的实体名，格式如下：

实体1：实体2

例如，若收银员与门诊处方之间存在联系，则表示如下：

收银员：门诊处方 或 门诊处方：收银员

【问题3】

根据实体联系图3-22填写挂号单、门诊处方、处方明细和药品库关系模式中的空（5）-（8）处，并指出挂号单、门诊处方和处方明细关系模式的主键。

版权方授权希赛网发布，侵权必究

[上一节](#)      [本书简介](#)      [下一节](#)

## 试题2

某宾馆需要建立一个住房管理系统，部分的需求分析结果如下。

（1）一个房间有多个床位，同一房间内的床位具有相同的收费标准。不同房间的床位收费标准可能不同。

（2）每个房间有房间号（如201、202等）、收费标准、床位数目等信息。

（3）每位客人有身份证号码、姓名、性别、出生日期和地址等信息。

（4）对每位客人的每次住宿，应该记录其入住日期、退房日期和预付款额等信息。

（5）管理系统可查询出客人所住房间号。

根据以上的需求分析结果，设计一种关系模型如图3-23所示。



图3-23 住房管理系统的实体联系图

【问题1】

根据上述说明和实体联系图，得到该住房管理系统的关系模式如下所示，请补充住宿 关系。

房间（房间号，收费标准，床位数目）

客人（身份证号，姓名，性别，出生日期，地址）

住宿（（1），入住日期，退房日期，预付款额）

【问题2】

请给出问题1中住宿关系的主键和外键。

【问题3】

若将上述各关系直接实现为对应的物理表，现需查询在2009年1月1日——2009年12月31日期



间，在该宾馆住宿次数大于5次的客人身份证号，并且按照入住次数进行降序排列。

下面是实现该功能的SQL语句，请填补语句中的空缺。

```
SELECT住宿.身份证号, count ( 入住日期 ) FROM 住宿, 客人
WHERE入住日期>="20090101" AND 入住日期<="20091231"
AND住宿.身份证号=客人.身份证号
GROUP BY ( 2 )
( 3 ) count ( 入住日期 ) >5
( 4 )
```

【问题4】

为加快SQL语句的执行效率，可在相应的表上创建索引。根据问题3中的SQL语句，除主键和外键外，还需要在哪个表的哪些属性上创建索引？应该创建什么类型的索引？请说明原因。

版权方授权希赛网发布，侵权必究

上一节      本书简介      下一节

试题3

某单位资料室需要建立一个图书管理系统，初步的需求分析结果如下。

- （1）资料室有图书管理员若干名，他们负责已购入图书的编目和借还工作，每名图书管理员的信息包括工号和姓名。
- （2）读者可在阅览室读书，也可通过图书流通室借还图书，读者信息包括读者ID、姓名、电话和E-mail，系统为不同读者生成不同的读者ID。
- （3）每部书在系统中对应惟一的一条图书在版编目数据（CIP，以下简称书目），书目的基本信息包括ISBN号、书名、作者、出版商、出版年月，以及本资料室拥有该书的册数（以下简称册数），不同书目的ISBN号不同。
- （4）资料室对于同一书目的图书可拥有多册（本），图书信息包括图书ID、ISBN号、存放位置、当前状态，每一本书在系统中被赋予惟一的图书ID。
- （5）一名读者最多只能借阅10本图书，且每本图书最多只能借两个月，读者借书时需要由图书管理员登记读者ID、所借图书ID、借阅时间和应还时间，读者还书时图书管理员在对应的借书信息中记录归还时间。
- （6）当某书目的可借出图书的数量为零时，读者可以对其进行预约登记，即记录读者ID、需要借阅图书的ISBN号、预约时间。

某书目的信息如表3-15所示，与该书目对应的图书信息如表3-16所示。

表3-15 数目信息

书名	作者	出版商	ISBN 号	出版年月	册数	经办人
数据结构	严蔚敏 吴伟民	清华大学出版社	ISBN7-302-02368-9	1997.4	4	01

表3-16 图书信息



预约登记 ( (b) , 预约时间, 预约期限, 图书ID )

注：时间格式为 “年.月.日 时:分:秒”

【问题3】

请指出问题2中给出的读者、书目关系模式的主键，以及图书、借还记录和预约登记关系模式的主键和外键。

版权方授权希赛网发布，侵权必究

上一节      本书简介      下一节

试题4

某市人才交流中心为促进当地人力资源的合理配置，加强当地企业与人才的沟通，拟建立人才信息交流网。

【需求分析结果】

（1）每个前来登记的个人需填写《人才入库登记表》（如表3-17所示），并出示相关证件，经工作人员审核后录入个人信息。

（2）每个前来登记的企业需填写《企业信息登记表》（如表3-18所示），并出示相关证明及复印件，经工作人员核实后录入企业信息。

表3-17 人才入库登记表

个人编号：\_\_\_\_\_ 登记日期：\_\_\_\_年\_\_\_\_月\_\_\_\_日

姓名		性别		出生日期		照片
身份证号						
毕业院校		专业		学历		
证书名称	1.		编号			
	2.					
联系电话				电子邮件		
求职意向及薪水	职位名称				最低薪水	
	1.					
	2.					
	3.					
个人简历及特长						

表3-18 企业信息登记表

企业名称		地址		企业网址	
联系人		联系电话		电子邮件	
岗位需求	职位名称	专业	学历	薪水	备注
企业简介					

（3）个人和企业的基本信息只需在第一次登记时填写，个人编号和企业编号由系统自动生成。个人和企业的基本信息由计算机长期存储，以后个人只需提供个人编号和求职意向信息，企业只需提供企业编号和岗位需求信息。

（4）个人的求职意向信息和企业的岗位需求信息在两个工作日内由工作人员录入数据库并发布。

【概念模型设计】

根据需求阶段收集的信息，设计人才、岗位和企业的实体联系图（不完整）如图3-25所示：

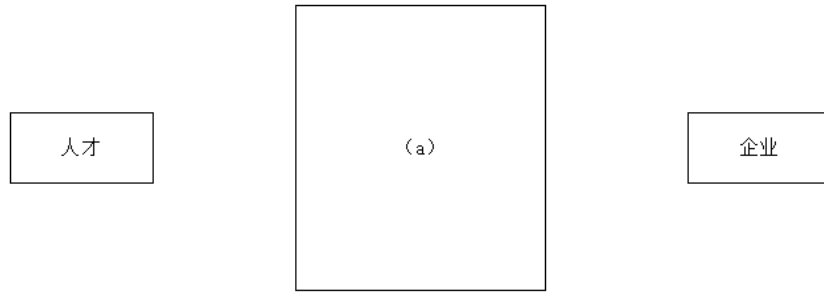


图3-25 人才、岗位和企业的实体联系图

【逻辑结构设计】

（1）将概念模型设计的实体联系图转换为以下关系模式：

人才（个人编号，姓名，性别，出生日期，身份证号，毕业院校，专业，学历，证书名称，证书编号，联系电话，电子邮件，个人简历及特长）

企业（企业编号，企业名称，联系人，联系电话，地址，企业网址，电子邮件，企业简介）

求职意向（\_\_\_\_\_（b）\_\_\_\_\_）

岗位需求（\_\_\_\_\_（c）\_\_\_\_\_）

（2）由于一个人可能持有多个证书，对“人才”关系模式进行优化，得到如下两个新的关系模式：

人才（\_\_\_\_\_（d）\_\_\_\_\_）

证书（\_\_\_\_\_（e）\_\_\_\_\_）

根据上述的设计过程，回答以下问题：

【问题1】

在（a）处填入所需的实体、联系及其属性，完成概念模型设计。

【问题2】

在（b）、（c）、（d）、（e）处填入对应关系的属性，完成逻辑结构设计。

【问题3】

对最终的各关系模式，以下画线指出其主键和外键。

【问题4】

张工设计的实体联系图如图3-26所示，请用200字以内的文字分析这样设计存在的问题。

【问题5】

如果允许企业通过互联网修改本企业的基本信息，应对数据库的设计做何种修改？请用200字以内的文字叙述实现方案。

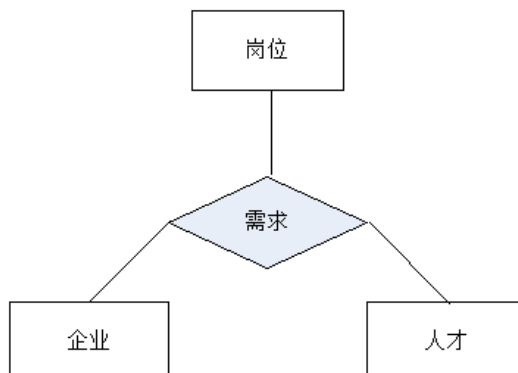


图3-26 张工设计的实体联系图

版权方授权希赛网发布，侵权必究

[上一节](#)
[本书简介](#)
[下一节](#)

试题5

M公司为某旅游公司设计机票销售专用数据库，其关系模式如图3-27所示。

航班（航班名，飞行日期，航空公司名称，出发地点，出发时间，目的地，到达时间）
旅游申请（团队编号，旅客编号，申请日期，出发日期，返回日期，担保人）
旅客（旅客编号，姓名，性别，出生日期，身份证号，联系方法）
搭乘航班（旅程编号，旅客编号，搭乘日期，航班名）

图3-27 机票销售专用数据库

关系模式的主要属性、含义及约束如表3-19所示，属性间函数依赖的标记方法如图3-28所示，属性间的函数依赖关系如图3-29所示。

表3-19 主要属性、含义及约束

属性	含义和约束条件
旅程编号	惟一标识每个能按期出发的旅行团队的编号。相同旅程编号的旅客，在同一日程中搭乘相同航班
旅客编号	惟一标识一个旅行团队中的每一位旅客的编号
团队编号	惟一标识每个旅行团队的编号，如“2004-8-4 云南双飞”
身份证号	惟一识别身份的编号

	表达式标记	图标记
函数依赖	$A \rightarrow B$	
	$\{A, B\} \rightarrow C$	
	$C \rightarrow \{A, B\}$ $C \rightarrow A$ $C \rightarrow B$	
多值依赖	$A \twoheadrightarrow \{B, C\}$ $A \twoheadrightarrow D$	

图3-28 函数依赖属性的标记方法

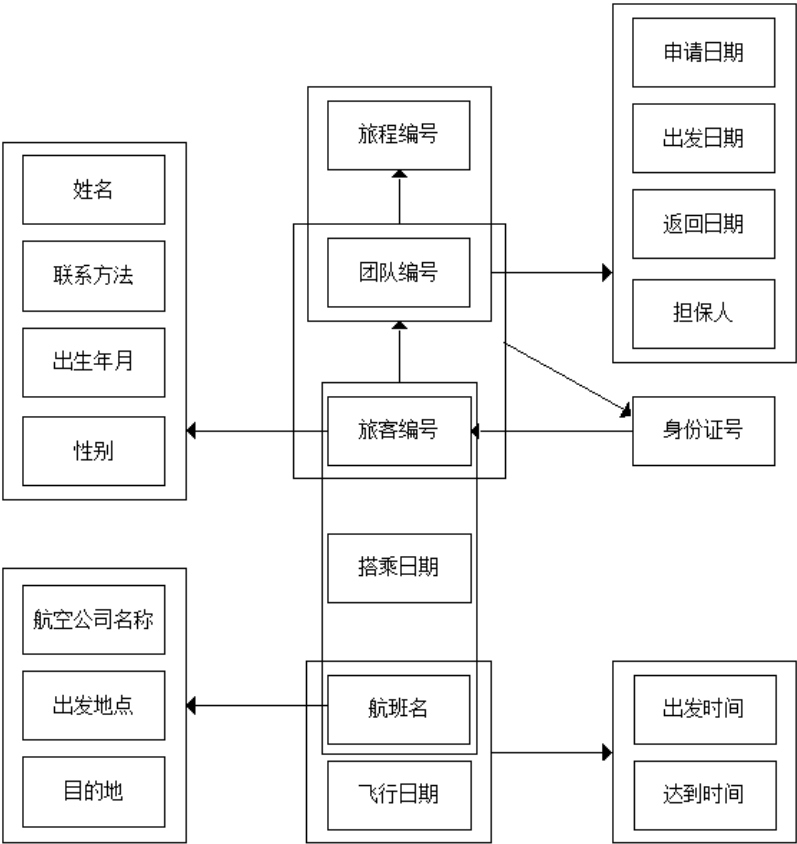


图3-29 航空公司销售的函数依赖图

旅客旅行前需要向旅行社提出申请，说明要参加的旅行团队。旅行社建立的旅行申请包括，旅行出发日期和到达日期的机票预订、购票等信息。旅行社还需要为每个团队制定“旅程”和“搭乘航班”表。有关“旅程”和“搭乘航班”的示例如表3-20、表3-21所示。

表3-20 “旅程” 示例

旅客编号	A01		旅程编号		P1
搭乘日期	出发地	目的地	出发时间	到达时间	航班名
2004.5.1	西安	桂林	10：00	13：00	JJ100
2004.5.1	桂林	昆明	17：00	19：00	CC4100
2004.5.5	昆明	西安	9：00	12：00	JJ600

表3-21 “搭乘航班” 示例

旅程编号	旅客编号	搭乘日期	航班名
P1	A01	2004.5.1	JJ100
P1	A01	2004.5.1	CC4100
P1	A01	2004.5.5	JJ600
P1	B02	2004.5.1	JJ100
P1	B02	2004.5.1	CC4100
P1	B02	2004.5.5	JJ600
P2	C03	2004.5.1	JJ200
P2	C03	2004.5.5	JJ700

【问题1】

对关系“航班” 请回答以下问题：

- （1）列举出所有不属于任何候选键的属性（非键属性）。
- （2）关系“航班” 可达到第几范式，用不超过60个字的内容叙述理由。

【问题2】

对关系“旅客” ，请回答以下的问题：

- （1）针对“旅客” 关系，用100字以内的文字简要说明会产生什么问题，并加以修正。
- （2）列出修正后的关系模式的所有候选键。

(3) 把“旅客”分解为第三范式，并用图3-27所示的关系模式的形式表示，分解后的关系名依次取旅客1、旅客2、.....

#### 【问题3】

对关系“搭乘航班”，请回答以下的问题：

(1) 把非平凡的多值依赖属性（图3-29中没有表示）的例子用满足图3-28的方式表示出来。

(2) 关系“搭乘航班”是BC范式而不是第四范式，请用200字以内文字阐述理由。

(3) 把“搭乘航班”关系分解成第四范式，并采用附录中规定的形式表示，分解后的关系名依次取搭乘航班1、搭乘航班2、.....

版权方授权希赛网发布，侵权必究

[上一节](#)      [本书简介](#)      [下一节](#)

第3章：数据库设计

作者：希赛教育软考学院    来源：希赛网    2014年05月06日

## 练习题解析

### 试题1分析

本题是一道数据库设计题，该类型的提问形式比较固定，在数据库系统工程师下午试题中是比较好得分的。

#### 【问题1】

该问题是求实体间的联系，这类问题主要通过“生活常识”+“系统描述”解题。

根据题目需求描述和表3-14中的数据可知：

由于一名医生在不同时间段可以给多个病人看病，也就可以开多张门诊处方，而一张门诊处方由一名医生开出。所以对于医生实体与门诊处方实体之间的联系“开处方”，其联系的类型为一对多（1:n）。所以（1）空的答案为1，（2）空的答案为n。

由于一张门诊处方包含多种库存中的药品（如“\*\*医院门诊处方单”表所示），一种库存中的药品也可以在多张门诊处方中。所以该联系的类型为多对多（m:n）。因此（3）空和（4）空均应填写n。

#### 【问题2】

根据“\*\*医院门诊挂号单”可以看出，挂号单由收银员进行收费，同时收银员的编号记录到了该挂号单中，因此挂号单实体与收银员实体之间存在联系——挂号单：收银员。

病人挂某个医师的号，将挂号信息记录在挂号单实体中，因此挂号单实体与医师实体之间存在联系——挂号单：医师。

收银员根据挂号单和医师的手写处方生成门诊处方，所以挂号单实体与门诊处方实体之间存在联系——挂号单：门诊处方。

#### 【问题3】

本题考查将E-R模型转换为关系模式。在此转化过程中，每一个实体转换成一个关系模式，对于联系的转换，相对比较复杂。可单独转换为关系模式，也可以将其并入实体关系模式中（注意：多对多的联系只能单独转换成一个关系模式，且该关系模式的主键为各个与之关联实体主键的组合）。所以一个关系模式的属性有两类，一类是实体本身具备的属性，另一类是为了保存实体与实

体之间的联系而记录的属性。下面将根据实体及与之相关的联系类型，并结合系统说明来分析。

(1) 对于“挂号单”关系模式，由于挂号单与收银员实体有联系，且它们之间的联系没有单独转换成关系模式，所以需要在“挂号单”关系模式中记录对应的收银员，因此“挂号单”关系模式需补充属性为“收银员”。

(2) 从“\*\*医院门诊处方单”可以得知“门诊处方”关系模式应具有的信息。但在此需要注意的是，哪些信息是“门诊处方”关系模式应直接存储的，哪些信息是可以通过查询从其他关系模式获取的。结合题目可知该关系就缺“就诊号”，若补充“就诊号”，则其他信息可通过“明细”、“收费”、“挂号门诊联系”、“开处方”等联系查询出来。

(3) 由于多张门诊处方中包含多项药品信息，而一种药品也可以属于多张门诊处方，所以通过“处方明细”关系模式来表示这种多对多的联系。并且由于每种药品的具体信息已经在“药品库存”关系模式中记录，所以“处方明细”关系模式主要记录的是门诊处方与药品的对应关系和处方所需药品的具体数量。并且，根据题目描述，由于药品价格会发生变化，门诊管理系统必须记录处方单上药品的当前单价。因此，“药品库存”关系模式补充属性：药品编号，数量，单价。其中就诊号和药品编号一起作为主键。

(4) “药品库存”关系模式主要记录药品的详细信息和库存信息，“药品库”表中已经说明需要记录的信息，所以应补充属性：类型、库存、货架编号、单位、规格、单价。

(5) 综上所述，挂号单与门诊处方主键均为“就诊号”。而处方明细是一个多对多的联系，它的主键应为与之关联的实体主键的组合，即：就诊号，药品编码。

#### 试题1参考答案

##### 【问题1】

(1) 1

(2) n

(3) n

(4) n

##### 【问题2】

缺少的联系数：3

挂号单：收银员

挂号单：医师

挂号单：门诊处方

##### 【问题3】

(5) 收银员

(6) 就诊号

(7) 药品编码，数量，单价

(8) 类型，库存，货架编号，单位，规格，单价

挂号单主键：就诊号

门诊处方主键：就诊号

处方明细主键：就诊号、药品编码

#### 试题2分析

本题考查数据库设计，涉及的知识点包括主键与外键的概念、SQL语言及索引相关知识。难点是



问题4，对索引相关知识的考查。

#### 【问题1】

此题要求补充住宿关系。从“住房管理系统的实体联系图”可以明显看出，住宿关系是从实体联系图中的联系“住宿”转换而来的，此联系是一个多对多的联系。若将多对多的联系转换为关系，则关系中应有联系的所有属性，以及与联系相关的所有实体的主键。

现在的住宿关系中已包含联系的所有属性，只缺客人关系的主键：身份证号，以及房间关系的主键：房间号。所以问题1的答案为：房间号，身份证号。

#### 【问题2】

本小题考查模式关系的主键，题目中有两个地方给出了提示。

(1) 实体联系图中房间与客人之间的关系是多对多，也就是说在住宿关系中，一个身份证号可以对应多个房间号。一个人显然不会同时住多个房间，所以多个房间情况的产生是因为多次入住，这就有可能产生多条记录的(房间号，身份证号)值相同，所以它不能成为主键，只有加上入住日期才能成为主键。此外在住宿关系中，房间号和身份证号都不是住宿关系的主键。但它们分别是房间关系和客人关系的主键，它们对于住宿关系来说是外键。

所以主键为：房间号，身份证号，入住日期；外键为：房间号，身份证号。

(2) 在问题3中提及“住宿次数大于5次的客人”，这也表示住宿关系中(房间号，身份证号)值相同是有可能的。

#### 【问题3】

此题考查简单的SQL语句。

位置(2)在填入SQL进行查询时，以什么关键字来进行分组，由于SQL语句前段部分有“Select 住宿.身份证号，count(入住日期)”，同时知道在分组SQL中，结果集的字段只能有两种情况：一是分组关键字；二是聚合函数。count(入住日期)属于聚合函数，剩下的住宿.身份证号只能是分组关键字，否则SQL非法。所以(2)应填入：住宿身份证号。

位置(3)是填入一个条件关键字，因为题目要求SQL找出“住宿次数大于5次的客人”，那么是不是填入“Where”子句呢？不是，应是“Having”。这两者的区别在于，“Having”后面的条件是当分组结束以后再进行判别，而“Where”是在分组之前进行判别，在分组之前，对每一条记录的count(入住日期)值必定是1，这样的条件是毫无意义的。所以此处应填：Having。

位置(4)需要完成的是“按照入住次数进行降序排列”功能。此功能可以用Order by count(入住日期) Desc子句来完成，也可以用Order by 2 Desc来完成，其中的Desc表示按降序排列，若要按升序排列，可以将其替换为Asc或直接取出Desc关键字，因为Order子句默认按升序排列。

#### 【问题4】

索引是加快检索表中数据的方法。在数据库中，索引也允许数据库程序迅速找到表中数据，而不必扫描整个数据库。其实，索引就相当于表中数据和相应存储位置的列表。使用索引可以大大减少数据库管理系统查找数据的时间。

索引的优点如下：

- (1) 通过创建惟一性索引，可以保证数据库表中每一行数据的惟一性。
- (2) 可以大大加快数据的检索速度，这也是创建索引的最主要原因。
- (3) 可以加速表和表之间的连接，特别是在实现数据的参照完整性方面很有意义。

(4) 在使用分组和排序子句进行数据检索时,同样可以显著减少查询中分组和排序的时间。

(5) 通过使用索引,可以在查询的过程中使用优化隐藏器,以提高系统性能。

索引的缺点如下:

(1) 创建索引和维护索引要耗费时间,这种时间随着数据量的增加而增加。

(2) 索引需要占物理空间,除了数据表占数据空间之外,每一个索引还要占一定的物理空间,如果要建立聚簇索引,那么需要的空间会更大。

(3) 当对表中的数据进行增加、删除和修改时,索引也要动态地维护,这样就降低了数据的维护速度。

索引的类型:根据索引的顺序与数据表的物理顺序是否相同,可以把索引分为两种类型:一种是数据表的物理顺序与索引顺序相同的聚簇索引(一个表只能建一个聚簇索引);另一种是数据表的物理顺序与索引顺序不同的非聚簇索引(一个表最多能建249个非聚簇索引)。

下面对这个题目进行具体分析。问题3的SQL涉及查询的字段有身份证号和入住日期。根据索引的相关性质可知,只有身份证号和入住日期有建立索引的必要,其余的字段不适合建立索引。又因为题目要求“除主键和外键外”,所以“身份证号”字段应排除,现只有“入住日期”需要建立索引。在索引中,聚簇索引适宜建单索引,而非聚簇索引适宜建多索引,所以此处建聚簇索引比较合适。当对“入住日期”建立聚簇索引后,可大大提高语句的查询速度。

#### 试题2参考答案

【问题1】

(1) 房间号,身份证号

【问题2】

住宿主键:房间号,身份证号,入住日期

住宿外键:房间号,身份证号

【问题3】

(2) 住宿.身份证号

(3) Having

(4) Order by count(入住日期) Desc 或者 Order by 2 Desc

【问题4】

表:住宿

属性:入住日期

类型:聚簇索引或聚集索引或cluster

原因:表中记录的物理顺序与索引项的顺序一致,根据索引访问数据时,一次读取操作可以获取多条记录数据,因而可以减少查询时间。

#### 试题3分析

本题考查数据库系统中实体联系模型(E-R模型)的设计和关系模式的设计。

【问题1】

两个实体型之间的联系可以分为以下3类。

(1) 一对一联系(1:1)

如果对于实体集A中的每一个实体,实体集B中至多有一个(也可以没有)实体与之联系,反之亦然,则称实体集A与实体集B具有一对一联系,记为1:1。

(2) 一对多联系 (1:n)

如果对于实体集A中的每一个实体，实体集B中有n个实体 ( $n \geq 0$ ) 与之联系，反之，对于实体集B中的每一个实体，实体集A中至多只有一个实体与之联系，则称实体集A与实体集B具有一对多联系，记为1:n。

(3) 多对多联系 (m:n)

如果对于实体集A中的每一个实体，实体集B中有n个实体 ( $n \geq 0$ ) 与之联系，反之，对于实体集B中的每一个实体，实体集A中也有m个实体 ( $m \geq 0$ ) 与之联系，则称实体集A与实体集B具有多对多联系，记为m:n。

该图书管理系统中的读者与图书之间形成了借还关系，题中说明“一名读者最多只能借阅10本图书”，显然一种图书可以被多名读者借阅，而每名读者应该能够借阅多本图书，因此读者与图书之间的借还关系为多对多联系 (n:m)。另外，资料室对于同一书目的图书可拥有多本，每一本书在系统中被赋予惟一的图书ID，所以书目与图书之间的联系类型为一对多联系 (1:m)。当某种书目的可借出图书的数量为零时，读者可以对其进行预约登记，由于一名读者可以借阅多种图书，因此书目与读者之间的预约联系类型为多对多联系 (n:m)。

【问题2】

由于读者借书时需由图书管理员登记读者ID、所借图书ID、借出时间和应还时间，还书时图书管理员在对应的借书信息中记录归还时间，因此借还记录关系中的空缺处应填入“读者ID，图书ID”。

读者对某书目进行预约登记时，需记录读者ID、需要借阅的图书的ISBN号、预约时间等，目前的预约登记关系中已经有预约时间、预约期限、图书ID信息，显然还需要记录是哪位读者预约了该书，以及书的ISBN号，因此预约登记关系模式中的空缺处应填入“读者ID，ISBN号”。

【问题3】

主键也称为码，是关系中的一个或一组属性，其值能惟一标识一个元组。根据题干部分的描述“系统为不同读者生成不同的读者ID”，因此读者关系的主键显然是“读者ID”。不同书目的ISBN号不相同，书目关系的主键为书的ISBN号。同一书目的多本图书具有相同的ISBN号，因此所有的图书依据“图书ID”相互区分，图书关系的主键是“图书ID”，外键是书目关系的“ISBN号”。借还记录关系用于记录读者的借书和还书信息，为了区分读者在同一日期对同一本书多次借还，借还记录的主键为“读者ID，图书ID，借出时间”。由于借还联系类型是多对多，故必须转换为一个独立的关系模式。这样，通过借还关系中的“借还记录”建立了图书和读者的关系。因此，借还记录具有外键“读者ID”和“图书ID”，分别与读者和图书相关联。同理，预约登记关系的主键为“读者ID，ISBN号，预约时间”，外键为读者关系的“读者ID”、书目关系的“ISBN号”和图书关系的“图书ID”。

试题3参考答案

【问题1】

- |       |         |
|-------|---------|
| (1) n | (2) m   |
| (3) 1 | (4) n或m |
| (5) m | (6) n   |

注：(1)、(2) 答案可互换，(5)、(6) 答案可互换。

【问题2】

(a) 读者ID, 图书ID

(b) 读者ID, ISBN号

【问题3】见下表

关系模式	主键	外键
读者	读者 ID	
书目	ISBN 号	
图书	图书 ID	ISBN 号
借还记录	读者 ID, 图书 ID, 借出时间	读者 ID, 图书 ID
预约登记	读者 ID, ISBN 号, 预约时间	读者 ID, ISBN 号, 图书 ID

#### 试题4分析

【问题1】

根据试题描述及表3-17和表3-18, 可以得出:

(1) 一个“人才”可以有多个“求职意向”。同时, 根据现实世界的实际情况, 一个“岗位”可以有多个“人才”去应聘, 因此, 实体“人才”和“岗位”之间的联系(“求职意向”)为多对多联系。而根据表3-17, 联系“求职意向”的属性有“最低薪水”。另外, “人才”的“求职意向”不是在初始时就已经确定的, 而是在以后可以由管理员录入的, 因此, 联系“求职意向”还应该有属性“登记日期”。

(2) 一个“岗位”可以有多个“企业”有需求, 例如, 很多软件企业都有数据库工程师这个职位。同时, 很显然, 一个“企业”也会有很多“岗位”。因此, 实体“岗位”和“企业”之间的联系(“岗位需求”)也为多对多的联系。而根据表3-18, 联系“岗位需求”的属性有“专业”、“学历”、“薪水”、“备注”。同时, 因为“企业”的“岗位需求”信息不是一次性的, 而是可以随时发布的, 因此, 联系“岗位需求”还应该有属性“登记日期”。根据上面的分析, 该系统的E-R图如图3-30所示。

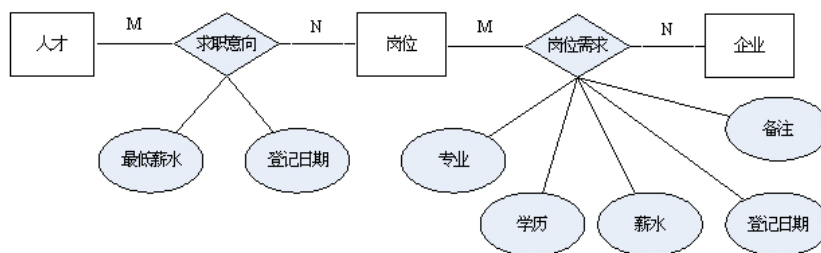


图3-30 完整的人才、岗位和企业的实体联系图

【问题2】

根据E-R图转换为关系模式的规则(一个m:n联系转换为一个关系模式, 与该联系相连的各实体的键及联系本身的属性均转换为关系的属性, 而关系的主键为各实体键的组合), 以及我们对问题1的分析, 可以很容易地写出关系“求职意向”和“岗位需求”的属性。

求职意向(个人编号, 岗位, 最低薪水, 登记日期)

岗位需求(企业编号, 岗位, 专业, 学历, 薪水, 备注, 登记日期)

在原来的“人才”关系中, 直接保存了“证书名称”, 因此, 一个人只能有一个证书, 但事实上, 一个人往往持有多个证书, 所以需要“人才”关系进行修改。具体修改方法是, 删除“人才”关系中属性“证书名称”和“证书编号”, 同时增加一个名为“证书”的关系, 其属性有“个人编号”、“证书名称”和“证书编号”。

【问题3】

问题3要求考生对最终的各关系模式, 以下画线指出其主键和外键。这就相当容易了。

关系“企业”的主键为“企业编号”，无外键。

关系“求职意向”的主键为“个人编号”和“岗位”，其中“个人编号”还为外键。

关系“岗位需求”的主键为“企业编号”和“岗位”，其中“企业编号”还为外键。

关系“人才”的主键为“个人编号”，无外键。

关系“证书”的主键为“个人编号”和“证书编号”，其中“个人编号”还为外键。

#### 【问题4】

图3-18中的“需求”是“岗位”、“企业”和“人才”三个实体之间的联系，但事实上，只有在“人才”被“企业”聘用之后，三者之间才产生关系。本系统解决的是人才的求职和企业的岗位需求，人才与企业之间没有直接的联系。

#### 【问题5】

在本系统中，所有信息都是由“人才”和“企业”提交给管理员，由管理员录入和分布的。这与现实是相违背的，事实上，现在的基于互联网的人才交流系统都允许企业或个人通过互联网修改自己的基本信息。

但是，在前面设计的关系模式中，并没有识别“企业”自身信息的属性。所以，如果要实现“允许企业通过互联网修改本企业的基本信息”的功能，需要建立企业的登录信息表，包含用户名和密码，记录企业的用户名和密码，将对本企业的基本信息的修改权限赋予企业的用户名，企业工作人员通过输入用户名和密码，经过服务器将其与登录信息表中记录的该企业的用户名和密码进行验证后，合法用户才有权限修改企业的信息。

值得注意的是，这里是指允许企业用户直接登录数据库管理系统进行系统维护。但通常的做法并不是这样，而是使用一个统一的数据库用户名（注意，这里不是通常网站上的注册用户名），通过一个网页，企业管理人员通过输入自己的注册用户名和密码登录，然后系统判断其是否有相应的操作权限，如果有，则允许修改。因此，一般的做法是在关系“企业”中增加“用户名”和“密码”的属性。

#### 试题4参考答案

##### 【问题1】

（a）如图3-30所示。

##### 【问题2】

（b）个人编号，岗位，最低薪水，登记日期

（c）企业编号，岗位，专业，学历，薪水，备注，登记日期

（d）个人编号，姓名，性别，出生日期，身份证号，毕业院校，专业，学历，联系电话，电子邮件，个人简历及特长

（e）个人编号，证书编号，证书名称

##### 【问题3】

企业（企业编号，企业名称，联系人，联系电话，地址，企业网址，电子邮件，企业简介）

求职意向（个人编号，岗位，最低薪水，登记日期）

岗位需求（企业编号，岗位，专业，学历，薪水，备注，登记日期）

人才（个人编号，姓名，性别，出生日期，身份证号，毕业院校，专业，学历，联系电话，电子邮件，个人简历及特长）

证书（个人编号，证书编号，证书名称）

#### 【问题4】

此处的“需求”是“岗位”、“企业”和“人才”三个实体之间的联系，而事实上只有人才被聘用之后三者之间才产生关系。本系统解决的是人才的求职和企业的岗位需求，人才与企业之间没有直接的联系。

#### 【问题5】

建立企业的登录信息表，包含用户名和密码，记录企业的用户名和密码，将对本企业的基本信息的修改权限赋予企业的用户，企业工作人员通过输入用户名和密码，经过服务器将其与登录信息表中记录的该企业的用户名和密码进行验证后，合法用户才有权修改企业的信息。

#### 试题5分析

##### 【问题1】

由图3-27可知，航班名 $\rightarrow$ （航空公司名称，出发地点，目的地），（航班名，飞行日期） $\rightarrow$ （出发时间，到达时间）。因此（航班名，飞行日期）是该关系“航班”的候选键，（航空公司名称，出发地点，目的地，出发时间，到达时间）是非主属性。显然，（航空公司名称，出发地点，目的地）这三个属性部分依赖于候选键，即只依赖于航班名。因此，“航班”关系不属于2NF，但每个属性都是不可再分割的，所以关系“航班”可达到1NF。

##### 【问题2】

由图3-27可知，（团队编号，旅客编号） $\rightarrow$ 身份证号，身份证号 $\rightarrow$ 旅客编号，旅客编号 $\rightarrow$ （姓名，联系方法，出生年月，性别）。因此，会出现当同一位旅客在不同时间参加了不同团队编号的团队时，旅客信息重复存储的情形，造成数据冗余。

采用分解方法把“旅客”同时还加上团队编号，结果为：

旅客1（团队编号，旅客编号，身份证号），其中候选键是（团队编号，旅客编号）和（团队编号，身份证号）；

旅客2（身份证号，姓名，联系方法，出生年月，性别），其中候选键是身份证号。

##### 【问题3】

根据不平凡的多值依赖属性的定义，易得出关系如图3-31所示。

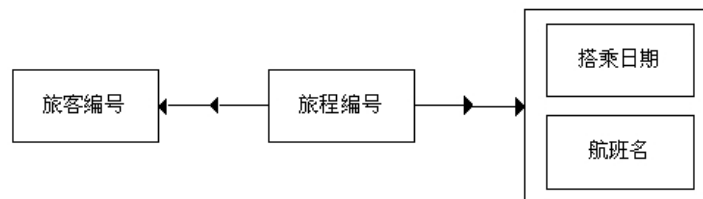


图3-31 多值依赖

4NF就是限制关系模式的属性之间不允许有非平凡且非函数依赖的多值依赖。因为根据定义，对于每一个不平凡的多值依赖 $X \twoheadrightarrow Y$ ， $X$ 都含有候选键，于是就有 $X \rightarrow Y$ ，所以4NF所允许的不平凡的多值依赖实际上是函数依赖。

因为旅客编号多值依赖于旅程编号，（航班名，搭乘日期）多值也依赖于旅程编号。同时“搭乘航班”关系是全码关系（即所有的属性都是主属性），因此它不属于4NF。

可以把“搭乘航班”关系进行分解成4NF，分解如下：

搭乘航班1（旅程编号，旅客编号），其中旅程编号是主键。

搭乘航班2（旅程编号，搭乘日期，航班名），其中旅程编号是主键。

#### 试题5参考答案

【问题1】

(1) “航班”关系模式的候选键为(航班名, 飞行日期), 非键属性为: 航空公司名称, 出发地点, 目的地, 出发时间, 到达时间。

(2) 关系“航班”可达到1NF。因为存在(航空公司名称, 出发地点, 目的地)部分依赖于候选键(航班名, 飞行日期)。该关系模式存在如下函数依赖: 航班名→航空公司名称, 出发地点, 目的地; (航班名, 飞行日期)→出发时间, 到达时间。

【问题2】

参考答案1:

(1) 在题中给出的“旅客”关系中, 不同的团队会有相同的旅客编号, 所以, 旅客编号不能作为候选键, 如果同一旅客不同时间参加不同的团队将导致“身份证号”无法确定关系中的每一个元组, 所以“身份证号”也不能作为候选键。为此, 需要增加一个“团队编号”的属性。又由于{(身份证号, 团队编号)→旅客编号; (旅客编号, 团队编号)→身份证号; 身份证号→(姓名, 联系方法, 出生日期, 性别)}, 所以该关系模式存在部分函数依赖, 导致旅客参加多少次团队, 其姓名等信息将重复多次。

(2) 候选键: (身份证号, 团队编号)和(旅客编号, 团队编号)

(3) “旅客”分解为第三范式如下所示:

旅客1(身份证号, 姓名, 联系方法, 出生日期, 性别)

旅客2(旅客编号, 身份证号, 团队编号)

参考答案2:

(1) 在题中给出的“旅客”关系中, 不同的团队会有相同的旅客编号, 所以, 旅客编号不能作为候选键, 如果同一旅客不同时间参加不同团队将导致“身份证号”无法确定关系中的每一个元组, 所以“身份证号”也不能作为候选键。为此, 可修改“旅客编号”属性定义加以解决, 旅客编号由“团队编号+队内编号”来解决。这时关系的候选键为“旅客编号”, 该关系模式存在传递依赖, 导致旅客参加多少次团队, 其姓名等信息就将重复多少次。

(2) 候选键: 旅客编号

(3) “旅客”分解为第三范式如下所示:

旅客1(身份证号, 姓名, 联系方法, 出生日期, 性别)

旅客2(旅客编号, 身份证号)

【问题3】

(1) 如图3-31所示。

(2) 在“搭乘航班”关系中, 存在着非平凡的多值依赖, 旅程编号→→旅客编号, 旅程编号→→(搭乘日期, 航班名), 而该关系模式的候选键为(旅程编号, 旅客编号, 搭乘日期, 航班名), 所以, 根据第四范式的定义, 该关系模式不属于第四范式。

(3) 分解为第四范式的结果如下:

搭乘航班1(旅程编号, 旅客编号)

搭乘航班2(旅程编号, 航班名, 搭乘日期)

## 试题解答方法

从历年试题的问答形式上来看，数据库设计部分考核的重点包括补充完整的实体联系图、补充关系模式、确定关系模式的主键和外键、数据库概念结构设计和逻辑结构设计、确定实体间联系、数据库的规范化等。下面我将介绍一些解决这些问题的方法。

(1) 关于关系模式的设计以及补充完整的实体联系图，需要仔细阅读需求分析的结果来解题，由需求分析的结果可知，关系实体有哪些、所包含的属性有哪些、实体之间的联系属于哪种类型，从而绘制出完整的实体联系图，补充完整的关系模式，同时确定关系模式的主键、外键。只要仔细地阅读题目中的有关说明，就非常容易找到答案，因此属于送分题。

(2) 关于数据库概念结构设计和逻辑结构设计，需要仔细阅读需求分析的结果或者仔细观察题目中已知的表，就很容易能够做出。关键是需要细心，不要漏掉什么属性。

所谓概念结构设计，就是将需求分析得到的用户需求抽象为信息结构，它是整个数据库设计的关键。在这个过程中有个很重要的环节，就是设计E-R图。

在概念设计阶段中，数据抽象是对实际的人、物、事和概念进行人为处理，抽取所关心的共同特性。有三种抽象形式，分别是分类、聚集和概括。其中概括是定义类型之间的一种子集联系，其重要性质是继承性。也就是说子类继承了超类上定义的所有抽象。

概念结构设计是独立于任何一种数据模型的信息结构。而逻辑结构设计任务是把概念结构设计阶段设计好的基本E-R图转换为与选用DBMS产品所支持的数据模型相符合的逻辑结构。

(3) 关于实体间的联系，这类问题主要通过“生活常识”+“系统描述”解题，就可以十分容易的确定实体之间的联系属于哪种类型。

两个实体型之间的联系可以分为一对一联系(1:1)，一对多联系(1:n)和多对多联系(m:n)3类。

(4) 关于函数依赖和数据库的规范化问题，需要结合函数依赖的定义以及题目中有关的说明进行分析，找出存在的函数依赖，同时确定关系模式属于第几范式，并对该模式进行分解，以达到更高的规范化要求。

函数依赖是通过一个关系中属性间值的相同与否体现出来的数据间的相互关系。函数依赖是关系模式内属性间最常见的数据依赖。

若对于一个关系模式中所有具体关系的属性之间都满足如下约束：对于X的每一个具体值，Y有惟一的具体值与之对应，则称Y函数依赖于X，或X函数决定Y，记作 $X \rightarrow Y$ ，X称作决定因素。如果 $X \rightarrow Y$ ，并且Y不是X的子集，则称 $X \rightarrow Y$ 是非平凡的函数依赖。若Y是X的子集，则称 $X \rightarrow Y$ 是平凡的函数依赖。

版权方授权希赛网发布，侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

## 考情分析



---

数据结构设计在软件设计师考试中的一个重点，它主要考查考生对数据结构部分的理解程度。对数据结构部分概念及应用掌握的好坏能直接反映出考生对于软件设计的基础掌握是否充分，以及对于算法的设计是否具备扎实的功底。因此，熟练掌握数据结构中的基本概念以及应用不仅有助于提高考生在考试中的得分能力，同时也能帮助考试为算法设计及编程打下一个良好的基础。

根据考试大纲，数据结构设计是每年必考的知识点，一般以C语言的形式进行描述。本章从基本的数据结构概念入手，着重描述了有关数据结构中栈，链表，二叉树和图等考试的主要知识点。

版权方授权希赛网发布，侵权必究

[上一节](#)    [本书简介](#)    [下一节](#)

第 4 章：数据结构设计

作者：希赛教育软考学院    来源：希赛网    2014年05月06日

## 考试大纲要求分析

根据考试大纲，数据结构设计要求考试掌握如下知识点：

（1）掌握C语言标准函数库。从以往考试来看，C语言的标准函数库是需要考生们注意掌握的，特别是一些常用的标准库函数，比如字符串中函数和一些常用的数学函数。

（2）能熟练掌握数据结构中的一些重要概念及使用C语言实现。数据结构中的算法通常是由C语言代码来实现的，而常用的数据结构有栈，树，图等概念。从考试角度出发，考生一定要理解这些概念并熟悉其性质。

（3）掌握数据结构设计和实现：线性表、查找表、树、图的顺序存储结构和链表存储结构的设计和实现。

在数据结构中，栈，队列，链表，二叉树和图的存储方式基本上有两种，一种是顺序存储，一种是链式存储。而在近5年的考试中，考试的重点放在的链式存储结构的考查上，这要求考生加深对链式存储结构的了解。

版权方授权希赛网发布，侵权必究

[上一节](#)    [本书简介](#)    [下一节](#)

第 4 章：数据结构设计

作者：希赛教育软考学院    来源：希赛网    2014年05月06日

## 命题特点与趋势分析

近5年来考查了数据结构设计部分的知识点是栈，链表，二叉树，拓扑排序，在近5年的8次考试中，有关树的知识点占了2次，图形处理考了1次，栈考查了2次，链表，二叉树，拓扑各考查了1次。从近几次命题的方向来看，图和栈相继被考查到了，估计在以后的考试中，会加强对二叉树和链式存储的有关操作的考查。望广大考生在备战的时候注意。

版权方授权希赛网发布，侵权必究