-------第七章作业-- -------------

7.2，7.22，7.23，7.25

--------第八章作业--- -----------

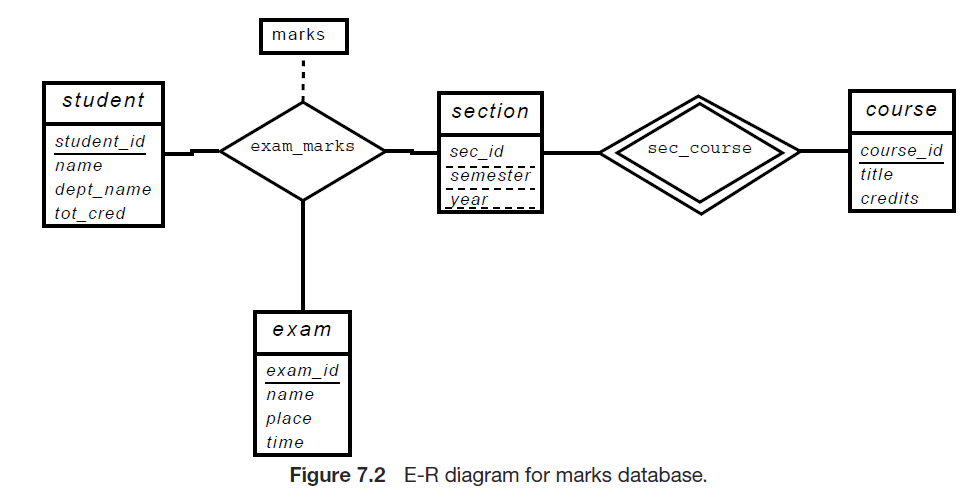
8.1，8.6，8.7，8.19，8.20，8.29

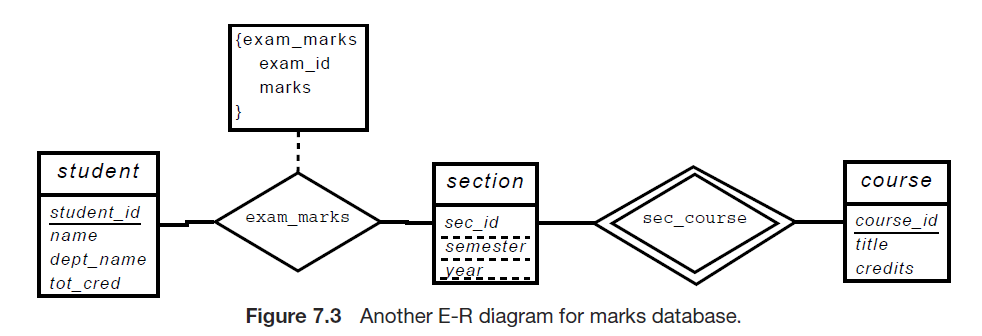
第七章 E-R图设计 作业参考答案

7.2

a. E-R 图显示如Figure 7.2。注：另一个替代方案是将exam考试建模为一个弱实体集关联到section，而不是强实体集。如此marks联系就成为*student*和*exam*间的二元联系, 而不直接包含*section*。

b. E-R 图显示如Figure 7.3. 注意这里我们没有建模name, place和time作为联系的属性。 如果这样做会导致对一个学生考试这些信息的重复，且如果没有一个相关的学生，我们也无法记录这个信息。如果我们希望表达这个信息, 将需要保留一个单独的实体对应于每一个考试。





7.22





Figure 7.6显示了两种可行的E-R图。

第二种E-R图的关系模式(包括主码和外码约束), 显示如下：.

*customer(customer\_id, name, address)*

*packet(packet\_id, weight)*

*place(place\_id, city, country, address)*

*sends(sender\_id, receiver\_id, packet\_id, time\_received, time\_sent,*

**foreign key** *sender\_id* **references** *customer,*

**foreign key** *receiver\_id* **references** *customer,*

**foreign key** *packet\_id* **references** *packet*

*)*

*has\_gone\_through(packet\_id, place\_id,*

**foreign key** *packet\_id* **references** *packet,*

**foreign key** *place\_id* **references** *place*

*)*

7.23



E-R图显示如Figure 7.7。我们假设在一段时间区间航班的时刻表schedule是固定的, 尽管我们允许指定航班在哪一天飞。对一个具体的航班飞行实例flight instance，我们要记录实际的起飞和到达时间。在现实情况下, 时刻表是能改变的, 这样具体航班的时刻表和飞行路线是针对具体日期, 或某一指定时间段的; 我们忽略这种复杂性。

*flight\_instance(flight\_inst\_id, capacity, date)*

*customer(customer\_id, name, address)*

*flight(flight\_id, runs\_on\_days)*

*station(station\_id, name, country)*

*has\_traveller(flight\_ inst\_id, customer\_id, seat\_number,*

**foreign key** *flight\_inst\_id* **references** *flight\_instance,*

**foreign key** *customer\_id* **references** *customer*

*)*

*instance\_of(flight\_inst\_id, flight\_id,*

**foreign key** *flight\_inst\_id* **references** *flight\_instance,*

**foreign key** *flight\_id* **references** *flight*

*)*

*has\_schedule(flight\_id, station\_id, order, arrival\_dayno, arrival\_time, departure\_dayno, departure\_time,*

**foreign key** *flight\_id* **references** *flight,*

**foreign key** *station\_id* **references** *station*

*)*

*has\_status(flight\_inst\_id, station\_id, arrival\_date, arrival\_time, departure\_date, departure\_time,*

**foreign key** *flight\_inst\_id* **references** *flight\_instance,*

**foreign key** *station\_id* **references** *station*

*)*

7.25 对课本7.6部分, 从图7.15 E-R图转换出的每一个模式, 给出它应有的外码约束（如果应该有，就创建）。

答案:

teaches(

foreign key ID references instructor,

foreign key (course\_id, sec\_id, semester, year) references sec course

)

takes(

foreign key ID references student,

foreign key (course\_id, sec\_id, semester, year) references sec course

)

prereq(

foreign key course\_id references course,

foreign key prereq\_id references course

)

advisor(

foreign key s\_ID references student,

foreign key i\_id references instructor

)

sec\_course(

foreign key course\_id references course,

foreign key (sec\_id, semester, year) references section

)

se\_time\_slot(

foreign key (course\_id, sec\_id, semester, year) references sec course

foreign key time\_slot\_id references time slot

)

sec\_class(

foreign key (course\_id, sec\_id, semester, year) references sec course

foreign key (building, room\_number) references classroom

)

inst\_dept(

foreign key ID references instructor

foreign key dept\_name references department

)

stud\_dept(

foreign key ID references student

foreign key dept\_name references department

)

course dept(

foreign key course\_id references course

foreign key dept\_name references department

)

第八章 规范化设计 作业参考答案

8.1 证明：(A, B, C) ∩ (A, D, E) = A

由已知A 🡪 BC 得 A 🡪 ABC

所以是无损分解。

**扩充：如何证明R1 ∩R2 🡪 R1 是无损连接分解的充分条件？**

答案: 关系模式 {R 1 , R 2 } 的无损连接分解{R1}、{R2}，只要满足R 1 ∩ R 2 → R 1 or

R 1 ∩ R 2 → R 2 。

设 R 1 = (A, B, C), R 2 =(A, D, E), 且 R 1 ∩ R 2 = A. 因为A 是R1的候选码 (见8.6), 因此 R 1 ∩ R 2 → R 1 .

8.6 解：

从*A → BC* 开始, 我们能得出: *A → B* and *A → C*.

由 *A → B* 且 *B → D*, 得*A → D* (分解, 传递律)

由 *A → CD* and *CD → E*, 得*A → E* (合并, 分解, 传递律)

由*A → A*, 我们有 (自反律)

*A → ABCDE* 从以上步骤(合并律)

由 *E → A*, *E → ABCDE* (transitive)

由 *CD → E*, *CD → ABCDE* (transitive)

由 *B → D* and *BC → CD*, *BC → ABCDE* (增补律, transitive).

因此, 任何左边是*A*, *E*, *BC*, or *CD* 的函数依赖是在函数依赖集闭包*F*+中, 无论还有 其它的属性出现在函数依赖中。

用 \* 来表示R中任意的属性集，那么 *F*+ 是 *BD → B*, *BD → D*,

*C → C*, *D → D*, *BD → BD*, *B → D*, *B → B*, *B → BD*, 和所有的行使如下的函数依赖

*A∗ → α*, *BC∗ → α*, *CD∗ → α*, *E∗ → α* ，这里*α*是的*{A, B, C, D, E}*任意子集。

候选码是 *A, BC, CD,* and *E*.

8.7

FC= F

答案: 给出的函数依赖集F是：

A → BC

CD → E

B → D

E → A

F中的每一个函数依赖的左边都是唯一的，且函数依赖左边或右边的属性没有一个是无关属性，因此此依赖集F是一个函数依赖正则覆盖Fc。

8.19 解：(B, D) (A, B, C, E)

求解思路：

对R = (A, B, C, D, E) 逐个检查F中的依赖，发现非平凡的*B → D* ，B不是超码，分解得 (B, D) (A, B, C, E)

(B, D)是BCNF。

对(A,B, C, E)逐个检查F在(A, B, C, E)上的限定，没有发现违背BCNF的非平凡依赖，(A,B, C, E)是BCNF。

（难点：确定F在…上的限定；判断α是否超码。）

8.20 解：(A, B, C) (C, D, E) (B, D) (A, E)

求解思路：

先求Fc，得Fc = F

依据Fc进行构造，得：

(A, B, C) (C, D, E) (B, D) (A, E)

8.29

1. {BDACE}

c. {A🡪BC, B🡪DE, D🡪A}

d. (A,B,C) (B, D, E) ( A, D) (A, F)

e. (A, B, C, D) (A, E, F)

(A, B, C, D) (A, E) (A, F)

f. NO (对于其他关系模式，答案可能是 YES)