2.9 考虑图 2-15 所示银行数据库。

a. 适当的主码是什么？

b. 给出你选择的主码，确定适当的外码。

branch(branch_name, branch_city, assets)
customer (customer_name, customer_street, customer_city)
loan (loan_number, branch_name, amount)
borrower (customer_name, loan_number)
account (account_number, branch_name, balance)
depositor (customer_name, account_number)

图 2-15 习题 2.8、习题 2.9 和习题 2.13 的银行数据库

a.

假设customer_name都是唯一的，customer可以有多个account和loan

branch：{branch_name}

customer：{customer_name}

loan：{loan_number}

borrower：{account_number, loan_number}

account：{account_number}

depositor：{customer_name, account_number}

b.

loan：{branch_name}参照branch

borrower：{account_number}参照account，{loan_number}参照loan

account：{branch_name}参照branch

depositor：{customer_name}参照customer，{account_number}参照account

2.10 考虑图 2-8 所示 *advisor* 关系，*advisor* 的主码是 *s_id*。假设一个学生可以有多位指导老师。那么，*s_id* 还是 *advisor* 关系的主码吗？如果不是，*advisor* 的主码会是什么呢？

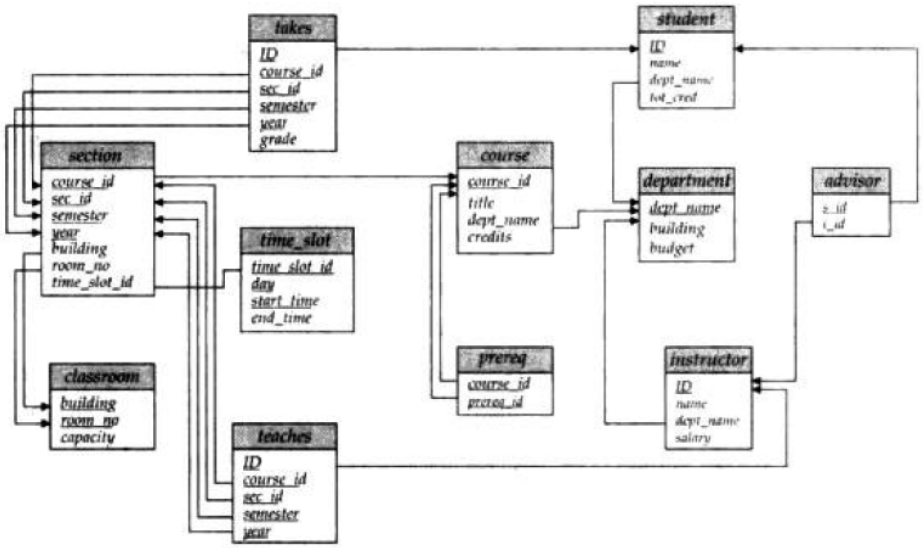

图 2-8 大学数据库的模式图

否。一个学生可以有多位指导老师，一位指导老师可以有多个学生。
{s_id, t_id}

2.11 解释术语关系和关系模式在意义上的区别。

关系模式是类型定义，关系是关系模式的实例。

2.12 考虑图 2-14 所示关系数据库。给出关系代数表达式来表示下列每一个查询：

a. 找出为“First Bank Corporation”工作的所有员工姓名。

b. 找出为“First Bank Corporation”工作的所有员工的姓名和居住城市。

c. 找出为“First Bank Corporation”工作且挣钱超过 10 000 美元的所有员工的姓名、街道地址和居住城市。

employee(*person-name*, *street*, *city*)
works(person-name, *company-name*, *salary*)
company(*company-name*, *city*)

图 2-14 习题 2.1、习题 2.7 和习题 2.12 的关系数据库

a.

$$\Pi_{\text{person-name}}(\sigma_{\text{company-name=”First Bank Corporation”}}(\text{works}))$$

b.

$$\Pi_{\text{person-name,city}}(\text{employee} \bowtie (\sigma_{\text{company-name=”First Bank Corporation”}}(\text{works})))$$

c.

$$\Pi_{\text{person-name,street,city}}(\text{employee} \bowtie (\sigma_{\text{company-name=”First Bank Corporation”} \wedge \text{salary}>10000}(\text{works})))$$

2.13 考虑图 2-15 所示银行数据库。对于下列每个查询，给出一个关系代数表达式：

a. 找出贷款额度超过 10 000 美元的所有贷款号。

b. 找出所有这样的存款人姓名，他拥有一个存款额大于 6000 美元的账户。

c. 找出所有这样的存款人姓名，他在“Uptown”支行拥有一个存款额大于 6000 美元的账户。

branch(branch_name, branch_city, assets)
customer (customer_name, customer_street, customer_city)
loan (loan_number, branch_name, amount)
borrower (customer_name, loan_number)
account (account_number, branch_name, balance)
depositor (customer_name, account_number)

图 2-15 习题 2.8、习题 2.9 和习题 2.13 的银行数据库

a.

$$\Pi_{\text{loan\_number}}(\sigma_{\text{amount}>10000}(\text{loan}))$$

b.

$$\Pi_{\text{customer\_name}}(\text{depositor} \bowtie (\sigma_{\text{balance}>6000}(\text{loan})))$$

c.

$$\Pi_{\text{customer\_name}}(\text{depositor} \bowtie (\sigma_{\text{branch\_name}=\text{"Uptown"} \wedge \text{balance}>6000}(\text{account})))$$

2.14 列出在数据库中引入空值的两个原因。

表示未知或不存在

2.15 讨论过程化和非过程化语言的相对优点。

非过程语言极大地简化了查询的规范（至少，它们旨在处理的查询类型）。用户无需担心如何评估查询；这不仅减少了编程工作量，而且实际上，在大多数情况下，查询优化器比通过反复试验的程序员能够更好地选择最佳的评估查询方法。另一方面，就程序语言可执行的计算而言，它们的功能要强大得多。某些任务要么无法使用非过程语言来完成，要么很难使用非过程语言来表达，或者如果以非过程方式指定，则执行效率很低。