

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。

答：a. branch 的主码是 branch_name, customer 的主码是 {customer_name, customer_street, customer_city}, loan 的主码是 loan_number, borrower 的主码是 loan_number, account 的主码是 account_name, depositor 的主码是 account_number。

- b. branch 没有外码, customer 没有外码, loan 的外码是 branch_name 和 loan_number, borrower 的外码是 loan_number, account 的外码是 account_number 和 branch_name, depositor 的外码是 account_number。

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?

答: s_id 不是 advisor 关系的主码, advisor 的主码可以是 {s_id, i_id}。

2.11 解释术语关系和关系模式在意义上的区别。

答: 关系的概念对应于程序设计语言中变量的概念, 而关系模式的概念对应于程序设计语言中类型定义的概念。

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 "Find Bank Corporation" 工作的所有员工姓名。

$\Pi_{person-name}(\sigma_{company-name="Find Bank Corporation"}(employee \bowtie works))$

- b. 找出为 "Find Bank Corporation" 工作的所有员工的姓名和居住城市。

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

- c. 找出为 "Find Bank Corporation" 工作且挣钱超过 10000 美元的所有员工的姓名、街道地址和居住城市。

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

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 10000 美元的所有贷款号。

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

- 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

号。

$\Pi_{customer_name}(\sigma_{balance>6000}(account \bowtie depositor))$

- 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。

$\Pi_{customer_name}(\sigma_{balance>6000 \wedge branch_name="Uptown"}(account \bowtie depositor)))$

- 2.14 列出在数据库引出空值的两个原因。

答： 1) 当数据库中的某个属性不存在的时候；
2) 当数据库中某个属性未知的时候；

- 2.15 讨论过程化和非过程化语言的相对优点。

答： 非过程化语言对于用户比较方便、友好，但过程化语言运行起来比非过程化语言快。