

Chapter 2: Relational Model

关系的概念

- 集合 D_1, D_2, \dots, D_n , 关系则是 $D_1 \times D_2 \times \dots \times D_n$ 的一个子集。
- 结果为多个 n 元组 (a_1, a_2, \dots, a_n) 的集合, $a_i \in D_i$ 。

关于属性

- 一定要有名字
- 一定要有值域
- 通常要有原子性 —— 电话号码, 取其中几位没有意义
- 如果一个属性所有取值都是原子的, 那么称其值域也是原子的
- 值域中都包含null —— 空值允许参与运算, 运算后还是null

规范

- 大写字母R: 表示关系模式
- 小写字母r: 表示具体关系, r(R)
- 表中, 属性也称为列, 一行数据也称为元组或者行

银行的例子

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

问题一: Find all loans of over \$1200
 $\sigma_{amount > 1200}(loan)$

问题二: Find the loan number for each loan of an amount greater than \$1200
 $\Pi_{loan_name}(\sigma_{amount > 1200}(loan))$

问题三: Find the names of all customers who have a loan, an account, or both, from the bank
 $\Pi_{customer_name}(borrower) \cup \Pi_{customer_name}(depositor)$

问题四: Find the names of all customers who have a loan at the Perryridge branch.
 $\Pi_{customer_name}(\sigma_{branch_name = 'Perryridge'}(\sigma_{borrow.loan_number = loan.loan_number}(borrow \times loan)))$

问题四的另一种解法 $\Pi_{customer_name}(\sigma_{borrow.loan_number = loan.loan_number}(borrow \times \sigma_{branch_name = 'Perryridge'}(loan)))$

问题五: Find the names of all customers who have a loan at the Perryridge branch but do not have an account at any branch of the bank.
 $\Pi_{customer_name}(\sigma_{branch_name = 'Perryridge'}(\sigma_{borrow.loan_number = loan.loan_number}(borrow \times loan))) - \sigma_{customer_name}(depositor)$

问题六: Find the largest account balance
 $\Pi_{balance}(account) - \Pi_{account.balance}(\sigma_{account.balance < d.balance}(account \times \rho_d(account)))$

举例 $\sigma_{branch_name = 'Perryridge'}(account)$ —— 在关系r中, 且满足条件p —— 选择

举例 $\Pi_{account_number, balance}(account)$ —— 只罗列出部分属性, 会去重 —— 投影

举例 $\Pi_{customer_name}(despositor) \cup \Pi_{customer_name}(borrower)$ —— 要求r和s的属性要相容 (属性个数相同, 对于的值域也相同) —— 并

$r - s = \{t | t \in r \text{ or } t \notin s\}$

要求r和s的属性要相容 (属性个数相同, 对于的值域也相同) —— 差

举例 $\sigma_A = C(r \times s)$ —— 假设r和s没有共同属性, 往往要和其他运算配合 —— 乘

$\rho_X(A_1, A_2, \dots, A_n)(E)$
将关系模式E的名字换成X, 属性名也换成A1, A2, ..., An —— 将关系模式E的名字换成X —— 换名

keys (码)

- 如果每一行都不同, 则称为superkey (超码)
- 超码中属性最少的称为candidate (候选码)
- 从候选键中挑选一个始终不会变的作为Primary key (主码)
- 约束数据 —— 一个关系中的某个属性, 是另一个关系中的主码, 称为Foreign key (外键)

数据库

由很多关系构成

附加的关系代数运算

$r \cap s = r - (r - s)$
 $r \cap s = \{t | t \in r \text{ and } t \in s\}$ —— 交

R=(A, B, C), S=(E, B, D)
 $r \bowtie s = \Pi_{r.A, r.B, r.C, r.D, S.E}(\sigma_{r.B=s.B, r.D=s.D}(r \times s))$ —— 自然连接

R=(A1,...,Am,...,B1,...,Bn)
S=(B1,...,Bn)
R-S=(A1,...,Am)
 $r \div s = \{t | t \in \Pi_{R-S}(r) \wedge \forall u \in S(t, u \in r)\}$ —— 除
 $r \div s = \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times s) - \Pi_{R-S}(r))$

$tmp1 \leftarrow \Pi_{R-S}(r)$
 $tmp2 \leftarrow \Pi_{R-S}((\Pi_{R-S}(r) \times s) - \Pi_{R-S}(r))$
 $res = tmp1 - tmp2$ —— 赋值

问题一: Find the names of all customers who have a loan and an account at bank.
 $\Pi_{customer_name}(borrow) \cap \Pi_{customer_name}(depositor)$

问题二: Find the name of all customers who have a loan at the bank and the loan amount.
 $\Pi_{customer_name, loan_number, amount}(borrower \bowtie loan)$

例题

问题三: Find all customers who have an account from at least the "Downtown" and the Uptown" branches.
 $\Pi_{customer_name}(\sigma_{branch_name = 'Downtown'}(depositor \bowtie account)) \cap \Pi_{customer_name}(\sigma_{branch_name = 'Uptown'}(depositor \bowtie account))$

问题三"除"的解法: $\Pi_{customer_name, branch_name}(depositor \bowtie account) \div \rho_{tmp}(branch_name)((\text{" Downtown "}, (\text{" Uptown "}))$

问题五: Find all customers who have an account at all branches located in Brooklyn city.
 $\Pi_{customer_name, branch_name}((depositor \bowtie account)) \div \Pi_{branch_name}(\sigma_{branch_city = 'Brooklyn'}(branch))$

扩展关系代数运算

$\Pi_{F1, F2, \dots, Fn}(E)$ (F_i 代表算数表达式) —— 广义投影

函数: avg, min, max, sum, count —— 分组: 举例: $branch_name \text{ } g \text{ } sum(balance) \text{ as } sum_balance(account)$

左外连接 \ltimes 、右外连接 \ltimes 、全外连接 \ltimes (填充 null) —— 外连接

不存在

不知道是多少

OR:
(unknown or true) = true
(unknown or false) = unknown
(unknown or unknown) = unknown

AND:
(true and unknown) = unknown
(false and unknown) = false
(unknown and unknown) = unknown

举例1: Delete all account records in the Perryridge branch.
 $account \leftarrow account - \sigma_{branch_name = 'Perryridge'}(account)$

举例2: Delete all loan records with amount in the range of 0 to 50.
 $loan \leftarrow loan - \sigma_{amount \geq 0 \text{ and } amount \leq 50}(loan)$ —— 删除

举例3: Delete all accounts at branches located in Needham.
 $r_1 \leftarrow \sigma_{branch_city = 'Needham'}(account \bowtie branch)$
 $r_2 \leftarrow \Pi_{account_number, branch_name, balance}(r_1)$
 $r_3 \leftarrow \Pi_{customer_name, account_number}(r_2 \bowtie depositor)$
 $account \leftarrow account - r_2$
 $depositor \leftarrow depositor - r_3$

举例1: $account \leftarrow account \cup \{(\text{" A - 937 "}, \text{" Perryridge "}, 1200)\}$

举例2: Provide as a gift for all loan customers in the Perryridge branch, a \$ 200 savings account. Let the loan number serve as the account number for the new savings account.
 $r_1 \leftarrow \sigma_{branch_name = 'perryridge'}(borrower \bowtie loan)$
 $account \leftarrow account \cup \Pi_{loan_number, branch_name, 200}(r_1)$
 $depositor \leftarrow depositor \cup \Pi_{customer_name, loan_number}(r_1)$ —— 插入

举例1: Make interest payments by increasing all balances by 5 percent.
 $account \leftarrow \Pi_{account_number, branch_name, balance * 1.05}(account)$

举例2: Pay all accounts with balances over \$10,000 6 percent interest and pay all others 5 percent. (注意顺序)
 $account \leftarrow \Pi_{account_number, branch_name, balance * 1.06}(\sigma_{BAL > 1000}(account)) \cup \Pi_{account_number, branch_name, balance * 1.05}(\sigma_{BAL \leq 1000}(account))$ —— 更新