CSCC43 UTSC

Tutorial Week 2

Relational Algebra Exercises for Tutorial

Schema 1

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

Consider the bank database above. Assume that branch names and customer IDs uniquely identify branches and customers, but loans and accounts can be associated with more than one customer.

1. What are the appropriate primary keys?
   Answer:
   The primary keys of the various schemas are underlined.
   We allow customers to have more than one account, and more than one loan. That's why both ID and loan_number are keys for the borrower to uniquely identify the borrower tuple. Similarly both ID and account_number are keys for the depositor to uniquely identify depositor tuple.

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

2. Given your choice of primary keys, identify appropriate foreign keys.
   i. For loan: branch name referencing branch.
   ii. For borrower: Attribute ID referencing customer and loan number referencing loan
   iii. For account: branch name referencing branch.
   iv. For depositor: Attribute ID referencing customer and account number referencing account

3. Give an expression in relational algebra to find each loan number with a loan amount greater than $10000.
   Π_{loan_number} ($\sigma_{amount > 10000}$ (loan))

Schema 2

Consider the employee database of with the appropriate primary keys underlined

employee (<u>emp_ID</u>, person_name)
company (<u>company_name</u>, rank)
roster(<u>emp_ID</u>, <u>company_name</u>, salary, city)

roster [emp_ID] ⊆ employee [emp_ID]
roster [company_name] ⊆ company [company_name]

1. Find all the IDs of all employees with the name “Rahul” or with the name “Emma”
   Answer:
   Π_{emp_ID} ($\sigma_{person\_name = \text{“Rahul”} \vee person\_name = \text{“Emma”}}$ employee)

2. Find the name of each employee who lives in city Miami
   Answer:
   Π_{person_name} (($\sigma_{city = \text{“Miami”}}$ roster) $\bowtie$ employee)

3. Find the name of each employee whose salary is greater than $100000.
   Answer:
   Π_{person_name} (($\sigma_{salary > 100000}$ roster) $\bowtie$ employee)

4. Find the ID and names of each employee who lives in Miami and whose salary is greater than $100000.
   Answer:
   $\Pi_{emp\_ID,\ person\_name}$ ($\sigma_{city = \text{“Miami”} \wedge salary > 100000}$ (roster $\bowtie$ employee))

5. Find the names of companies that have a rank of at least 5 and are in Miami.
   Answer:
   $\Pi_{company\_name}$ ($\sigma_{city = \text{“Miami”} \wedge rank \geq 5}$ (company $\bowtie$ roster))

Schema 3

Suppliers(<u>sID</u>, sName, address)
Parts(<u>pID</u>, pName, colour)
Catalog(<u>sID</u>, <u>pID</u>, price)

Catalog[sID] $\subseteq$ Suppliers[sID]

Catalog[pID] $\subseteq$ Parts[pID]

Solve all queries below using only select, project, Cartesian product, and natural join.

1. If sID is a key for the Suppliers relation, could it be a key for the Catalog relation?
   Answer:
   No.
   Key is relative to a particular relation. Just because it is a key in one relation doesn't mean it is in another.
   It is not a key for Catalog mainly because we want to be able to list multiple parts by one supplier in our catalog.

2. Find the names of all red parts.
   Answer:
   $\Pi_{pName}(\sigma_{colour= \text{“red”}}Parts)$

3. Find the sIDs of all suppliers who supply a part that is red or green.
   Answer:
   $\Pi_{sID}((\sigma_{colour= \text{“red”} \vee colour= \text{“green”}}Parts) \bowtie Catalog)$

4. Find all prices for parts that are red or green.
   Answer:
   $\Pi_{price}((\sigma_{colour= \text{“red”} \vee colour= \text{“green”}}Parts) \bowtie Catalog)$

5. Find the names of all suppliers who supply a part that is red or green.
   Answer:
   $\Pi_{sName}((\Pi_{sID}((\sigma_{colour= \text{“red”} \vee colour= \text{“green”}}Parts) \bowtie Catalog)) \bowtie Suppliers)$

6. Find the sIDs of all suppliers who supply a part that is red *and* green.
   Answer:
   It is not possible for a part to be red *and* green.
   Each tuple has only one colour, and each part has only one tuple (since pID is a key), so no part can be recorded as both red and green.