CSCI 6315 Applied Database Systems
ASSIGNMENT 2: Formal Relational Query Languages
02/18/2020 20:03

Ulvi Bajarani
Student ID 20539914
E-mail: ulvi.bajarani01@utrgv.edu

Initial Database:

employee(employee-id, employee-name, street, city)

works(employee-id, company-id, salary)

company(company-id, company-name, city)

manages(employee-id, manager-id)

Answers:

a) *Find the names of all employees who work for First Bank Corporation.*

the Relational algebra expression:

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

the Tuple relational calculus expression:

{ **t** | ∃ **s** ∈ *employee*(t[*employee-name*] = s[*employee-name*]) ∧
∃ **u** ∈ *works*(u[*employee-id*] = s[*employee-id*]) ∧
∃ **v** ∈ *company*(v[*company-id*] = u[*company-id*] ∧ v[*company-name*] = 'First Bank Corporation') }

the Domain relational calculus expression:

{ < **n** > | ∃ e-i, c-i, c-n (< n, e-i, s, c > ∈ *employee* ∧
< e-i, c-i, sa > ∈ *works* ∧
< c-i, c-n, c > ∈ *company* ∧ c-n = 'First Bank Corporation') }

b) *Find the names and cities of residence of all employees who work for First Bank Corporation.*

the Relational algebra expression:

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

the Tuple relational calculus expression:

{ **t** | ∃ **s** ∈ *employee*(t[*employee-name*] = s[*employee-name*] ∧ t[*city*] = s[*city*]) ∧
∃ **u** ∈ *works*(u[*employee-id*] = s[*employee-id*]) ∧
∃ **v** ∈ *company*(v[*company-id*] = u[*company-id*] ∧ v[*company-name*] = 'First Bank Corporation') }

the Domain relational calculus expression:

{ < **n, c** > | ∃ e-i, c-i, c-n (< n, e-i, s, c > ∈ *employee* ∧
< e-i, c-i, sa > ∈ *works* ∧
< c-i, c-n, c > ∈ *company* ∧ c-n ≠'First Bank Corporation') }

c) *Find the names, street addresses, and cities of residence of all employees who work for First Bank Corporation and earn more than $10,000.*

the Relational algebra expression:

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

the Tuple relational calculus expression:

{ **t** | ∃ **s** ∈ *employee*(t[*employee-name*] = s[*employee-name*] ∧ t[*street*] = s[*street*] ∧ t[*city*] = s[*city*]) ∧
∃ **u** ∈ *works*(u[*employee-id*] = s[*employee-id*] ∧ u[*salary*] > 10000) ∧
∃ **v** ∈ *company*(v[*company-id*] = u[*company-id*] ∧ v[*company-name*] = 'First Bank Corporation') }

the Domain relational calculus expression:

{ < **n, s, c** > | ∃ e-i, c-i, sa, c-n (< n, e-i, s, c > ∈ *employee* ∧
< e-i, c-i, sa > ∈ *works* ∧ sa > 10000 ∧
< c-i, c-n, c > ∈ *company* ∧ c-n = 'First Bank Corporation') }

d) *Find all employees in the database who live in the same cities as the companies for which they work.*

the Relational algebra expression:

$\Pi_{employee-name}(\sigma_{employee.city=company.city}(employee \bowtie works \bowtie company))$

the Tuple relational calculus expression:

$\{ \mathbf{t} \mid \exists\, \mathbf{s} \in employee(\ \mathrm{t}[employee\text{-}name] = \mathrm{s}[employee\text{-}name]\ ) \wedge$
$\exists\, \mathbf{u} \in works(\ \mathrm{u}[employee\text{-}id] = \mathrm{s}[employee\text{-}id]\ ) \wedge$
$\exists\, \mathbf{v} \in company(\ \mathrm{v}[company\text{-}id] = \mathrm{u}[company\text{-}id] \wedge \mathrm{v}[city] = \mathrm{s}[city]\ ) \}$

the Domain relational calculus expression:

$\{ < \mathbf{n} > \mid \exists$ e-i, c, c-i $(<$ n, e-i, s, c $> \in employee \wedge$
$<$ e-i, c-i, sa $> \in works \wedge$
$<$ c-i, c-n, c $> \in company\ ) \}$

e) *Find all employees in the database who live in the same cities and on the same streets as do their managers.*

the Relational algebra expression:

$$\Pi_{employee-name}(\sigma_{manages.manager-id\ =\ employee2.employee-id\ \wedge\ employee.street=employee2.street\ \wedge\ employee.street=employee2.street}(employee \bowtie manages \bowtie \rho_{employee2}(employee)))$$

the Tuple relational calculus expression:

$\{\ \mathbf{t} \mid \exists\ \mathbf{s} \in$ *employee*(t[*employee-name*] = s[*employee-name*]) $\wedge$
$\exists\ \mathbf{v} \in$ *manages*(v[*employee-id*] = s[*employee-id*] $\wedge$ v[*manager-id*] = s[*employee-id*] $\wedge$ v[*city*]
= s[*city*] $\wedge$ v[*street*] = s[*street*]) $\}$

the Domain relational calculus expression:

$\{ <$ **n** $> \mid \exists$ e-i, c, s, m-i($<$ n, e-i, s, c $> \in$ *employee* $\wedge$
$\exists$ <m-i, e-i> $\in$ *manages* $\wedge$
$<$ n, m-i, s, c $> \in$ *employee*)) $\}$

f) *Find all employees in the database who do not work for the First Bank Corporation.*

the Relational algebra expression:

$\Pi_{employee-name}(employee \bowtie (\sigma_{company-name \neq \text{“}First\ Bank\ Corporation\text{”}} (works \bowtie company)))$

the Tuple relational calculus expression:

{ **t** | ∃ **s** ∈ *employee*(t[*employee-name*] = s[*employee-name*]) ∧
∃ **u** ∈ *works*(u[*employee-id*] = s[*employee-id*]) ∧
∃ **v** ∈ *company*(v[*company-id*] = u[*company-id*] ∧ v[*company-name*] ≠ 'First Bank Corporation') }

the Domain relational calculus expression:

{ < **n** > | ∃ e-i, c-i, c-n (< n, e-i, s, c > ∈ *employee* ∧
< e-i, c-i, sa > ∈ *works* ∧
< c-i, c-n, c > ∈ *company* ∧ c-n ≠ 'First Bank Corporation') }

g) *Find all employees in the database who earn more than each employee of Small Bank Corporation.*

the Relational algebra expression:

$\Pi_{employee-name}(employee) -$
$\Pi_{employee-name}(employee \bowtie (\sigma_{company-name = \text{“}Small\ Bank\ Corporation\text{”}} \wedge$
$_{works.salary \leq works2.salary}(company \bowtie works \bowtie \rho_{works2}(works))))$

the Tuple relational calculus expression:

$\{$ **t** | $(\forall$ **s** $\in$ *employee*(t[*employee-name*] = s[*employee-name*]) $\wedge$
$\forall$ **u** $\in$ *works*(u[*employee-id*] = s[*employee-id*]) $\wedge$ *works*(u[*salary*] = s[*salary*]) $\wedge$
$\forall$ **v** $\in$ *company*(v[*company-id*] = u[*company-id*] $\wedge$ v[*company-name*] = 'Small Bank Corporation') $\Longrightarrow$
$\exists$ **z** $\in$ *company*(z[*company-name*] $\neq$ 'Small Bank Corporation' $\wedge$ *works*(t[*salary*] $\geq$ u[*salary*]) $\}$

the Domain relational calculus expression:

$\{ <$ **n** $> \mid \exists$ e-i, c-i, c-n ($<$ n, e-i, s, c $> \in$ *employee* $\wedge$
$<$ e-i, c-i, sa $> \in$ *works* $\wedge$
$\forall$ e-i2, c-i2, sa2, c
($<$ e-i2, c-i2, sa2 $> \in$ *works* $\wedge <$ c-i, c-n, c $> \in$ *company* $\wedge$ c-n = 'Small Bank Corporation'
$\Longrightarrow$ sa $\geq$ sa2)) $\}$

h) *Assume that the companies may be located in several cities. Find all companies located in every city in which Small Bank Corporation is located.*

the Relational algebra expression:

$\Pi_{company-name}(company \bowtie \Pi_{company.city}(\sigma_{company-name\ =\ \text{“}Small\ Bank\ Corporation\text{”}}(company)))$

the Tuple relational calculus expression:

$\{\ \mathbf{t} \mid (\ \forall\ \mathbf{z} \in company(\ \mathrm{t}[city] = \mathrm{z}[city] \wedge \mathrm{z}[company] = \text{‘Small Bank Corporation’}\ ) \implies \exists\ \mathbf{s} \in company(\ \mathrm{t}[company] = \mathrm{s}[company]\ ) \wedge company(\ \mathrm{s}[city] = \mathrm{z}[city]\ )\ )\ \}$

the Domain relational calculus expression:

$\{ < \mathbf{c\text{-}n2} > \mid \exists\ \mathrm{c2}\ (< \text{c-i2, c-n2, c2} > \in company \wedge (\ \forall\ \text{c-i, c-n, c}\ (< \text{c-i, c-n, c} > \in company \wedge \text{c-n} = \text{‘Small Bank Corporation’} \implies \wedge\ \mathrm{c} = \mathrm{c2}\ )\ )\ )\ \}$