Information Storage and Management I

Relational Algebra (Zero Marks)

Lab Weeks 3 & 4

Part I

Translate the following join expression in SQL to relational Algebra (show parse tree of the expression)

SELECT TableA.A1, TableA.A2, TableA.A3, TableB.B2 FROM TableA, TableB WHERE TableA.A2 > TableB.B1 AND TableB.B2 = TableA.A2

Solution1:

$$\pi_{A1,A2,A3}(TableA \bowtie_{A2>B1 \wedge B2=A2} TableB)$$

Construct the Natural Join of the following two tables $(Table1 \bowtie Table2)$

Table1

A1	A2
1	Red
2	Blue
3	Black

Table2

A1	B2	B3
1	BMW	1972
2	Honda	1968
4	Ferrari	1978

Solution:

A1	A2	B2	B3
1	Red	BMW	1972
2	Blue	Honda	1968

Part 2

Consider the following relational DB, where the PKs are underlined. Give an expression in Relational algebra to express each of the following queries:

employee(person_name, street, city)

works(person_name, company_name, salary)

company(company_name, city)

manages(person_name, manager_name)

A. Find the names of all employees in this database who work for "First Bank Corporation"
$$\pi_{person\_name}(\sigma_{company\_name=" FirstBankCorporation"}(works))$$

B. Find the names and cities of residence of all employees who work for First Bank Corporation
$$\pi_{person\_name,city}(employee \bowtie \sigma_{company\_name=" FirstBankCorporation"}(works))$$

C. Find the names of all employees who live in the same city and on the same street as do their managers

$$\rho(emp2, employee)$$
$$\rho(temp1, (manager \times employee \times emp2))$$
$$\rho(temp2, \sigma_{manager.person\_name=employee.person\_name \wedge manager.manager\_name=emp2.person\_name}(temp1))$$
$$\rho(temp3, \sigma_{employee.stree=emp2.stree \wedge employee.city=emp2.city}(temp2))$$

Part 3

Consider relation schemas and instances given below. For each question below, write a relational algebra expression that computes the required answer. Furthermore, show both the parse tree and the actual result of the query.

Countries(country, continent)

country	continent
USA	North America
Canada	North America
UK	Europe
Germany	Europe
France	Europe
Andorra	Europe
Mali	Africa

Cities (city, country, is_capital, population)

city	country	is_capital	population
New York, NY	USA	no	8,000,000
Washington, DC	USA	yes	600,000
Philadelphia, PA	USA	no	1,500,000
Paris, TX	USA	no	25,000
Ottawa, ON	Canada	yes	800,000
Toronto	Canada	no	2,500,000
Berlin	Germany	yes	3,500,000
Hamburg	Germany	no	2,000,000
Bonn	Germany	no	300,000
Paris	France	yes	2,000,000
Lyon	France	no	700,000
Bamako	Mali	yes	2,000,000
Timbuktu	Mali	no	50,000
Mopti	Mali	no	100,000

1. List the names of North American countries whose capital cities have population of more than 1,000,000. Result should have the schema (country)

$$\pi_{country} \quad (\sigma_{continent=\text{“}NorthAmericia''}(Countries) \bowtie_{country} \sigma_{is\_capital=\text{“}yes'' \wedge population > 1000000}(cities))$$

2. List pairs of countries (country1, country2), such that the capital country1 is more populous than the capital of country2

$$\pi_{c1.country, c2.country} \quad (\rho(c1, \sigma_{is\_capita=\text{“}yes''}(Cities) \bowtie_{c1.population > c2.population} \rho(c2, \sigma_{is\_capital='yes'}(Cities))$$

3. List names of European countries for which no cities have been entered into the Cities table in our database. Result should have the schema: (country)

$$\pi_{country}(\sigma_{continent=\text{“}Europe''}(Countries)) - \pi_{country}(Cities)$$

Part 4

Consider the following schema:

Author(<u>id,</u> FirstName, LastName, YearOfBirth, Gender, LivingCityID)

Book(<u>id</u>, Name, Type, YearPublished, PubliserID)

Writes(<u>bookId, authorId</u>)

City(<u>id</u>, CityName, Country)

Publisher(<u>id</u>, PublisherName, LocationCityId)

Underlines attributes in bold are the primary keys. Assume that one book can be published by only one publisher; one author can write several books; one book may have several authors

1. Write SQL statements to create the above tables.
2. Specify a relational algebra expression and a SQL query to find all the book names in the database

$$\pi_{name}(Book)$$

3. Specify a relational algebra expression and a SQL query to find the IDs of authors born in 1964

$\pi_{id}(\sigma_{YearOfBirth>1967}(Author)$

4. Specify a relational algebra expression and a SQL query to find the IDs of authors that lives in Pittsburgh

$\pi_{author.id}(\sigma_{city.name=\text{“}Pitsburth''}(city \bowtie_{city.id=Author.LivingCityID} Author))$