RELATIONAL ALGREBRA SOLUTIONS

--Prepared by: Pradip Paudel

1. **Consider the following relational database, where primary keys are underlined.**

   employee(person_name,street,city)

   works(person_name,company_name,salary)

   company(company_name,city)

   manages(person_name,manager_name)

   Given an expression in the relational algebra to express each of the following queries.

   i)Find the name of all employees who work for first bank corporation.

   $\Pi_{person\_name}(\sigma_{company\_name="first\ bank\ corporation"}(works))$

   ii)Find the name and cities of residence of all employees who work for first bank corporation.

   $\Pi_{person\_name,city}(\sigma_{company\_name="first\ bank\ corporation"}(employee \bowtie works))$

   iii)Find the name, street address and city of residence of all employees who work for first bank corporation and earn more than $10000 per annum.

   $\Pi_{person\_name,street,city}(\sigma_{company\_name="first\ bank\ corporation" \wedge salary>10000}(employee \bowtie works))$

   (Here we assume attribute named salary represent Annual Salary)

   iv)Find the name of all employees in this database who lives in same city as the company for which they work.

   $\Pi_{person\_name}(employee \bowtie works \bowtie company)$

   v)Find the name of all employees who live in the same city and on the same street as do their managers.

   $\Pi_{person\_name}((employee \bowtie manages) \bowtie_{(manages.manager\_name=employee2.person\_name \wedge employee.street=employee2.street \wedge employee.city=employee2.city)} (\rho_{employee2}(employee)))$

   vi)Find the name of all employees in this database who do not work for first bank corporation

   $\Pi_{person\_name}(\sigma_{company\_name \neq "first\ bank\ corporation"}(works))$

   vii) Assume that companies may located in several cities. Find all companies located in every city in which small bank corporation is located.

   $\Pi_{company\_name,city}(company) \div \Pi_{city}(\sigma_{company\_name ="small\ bank\ corporation"}(company))$

2. Consider the following relational database.

Students(RollNo, StudentName, Address, Semester)

Teachers(TeacherID, TeacherName,CourseID,Salary,Department)

Courses(CourseID,RollNo,CourseTitle,Semester)

Write relational Algebra Expressions for the following requests.

i) Find the name of students of 4th semester and studying "Operating System"

$\prod_{StudentName}(\sigma_{Semester= "4th" \wedge CourseTitle="Operating System"}(Students \bowtie Courses))$

ii) Find the name of teacher who teaches subject "DBMS" to "Arten Khadka"

$\prod_{TeacherName}(\sigma_{CourseTitle="DBMS" \wedge StudentName="Arten Khadka"}((Students \bowtie Courses) \bowtie Teachers))$

iii) Delete Record of 2nd semester students of Account Department

$Students \leftarrow \prod_{RollNo, StudentName, Address, Semester}(Students)$

$- \prod_{RollNo, StudentName, Address, Semester}(\sigma_{Department="account" \wedge Semester=2nd}((Students \bowtie Courses) \bowtie Teachers))$

iv) Increase salary of "Bhaskar Bhatta" by 6%

$Teachers \leftarrow \prod_{TeacherID, TeacherName,CourseID,Salary*1.06,Department}(\sigma_{TeacherName= "Bhaskar Bhatta"}(Teachers)) \cup (Teachers - \sigma_{TeacherName= "Bhaskar Bhatta"}(Teachers))$

3. Consider the following schema

SUPPLIER(Sid,S_name,S_addr)

PARTS(Pid,p_name,color)

CATALOG(sid,pid,cost)

[PU:2014 fall]

Now answer the following queries in Relational Algebra.

i) Find the name of all supplier who supply yellow parts

$\prod_{S\_name}(\sigma_{color='yellow'}(SUPPLIER \bowtie CATALOG \bowtie PARTS))$

ii) Find the name of suppliers who supply Both blue and black parts.

$\prod_{S\_name,color}(SUPPLIER \bowtie (CATALOG \bowtie PARTS)) \div \prod_{color}(\sigma_{color='blue' \wedge color='black'}(PARTS))$

iii) Find the name of suppliers who supply all parts.

$\prod_{S\_name,Pid}(SUPPLIER \bowtie CATALOG) \div \prod_{Pid}(PARTS)$

4. Consider the following schema

customer(cus_id,cus_name,cus_phno)
employee(cus_id,emp_id,emp_name,emp_add)
works(branch_id,salary,cus_id)
branch(branch_id,branch_name)

Write relational algebra notations for the following queries for the given schema. [PU:2011 spring]

i) select name of all employees.

Π_{emp_name} (employee)

ii) Give salary rise to 5% to all the employee

works$\leftarrow \Pi_{branch_id,salary*1.05,cust_id}$(works)

iii) List all branch names

Π_{branch_name}(branch)

iv) select name of all employees working for "manang" branch

Π_{emp_name} ($\sigma_{branch\_name= "manang"}$(branch$\bowtie$ (works$\bowtie$employee)))

v) Delete any record from work table

works$\leftarrow$works-$\sigma_{branch\_id=101}$(works)

This operation can be customized according to requirement.

vi) List the name and phno of all customers

Π_{cust_name,cus_phno}(customer)

vii) select name of all employees deal with customer having id "201"

Π_{emp_name} ($\sigma_{cus\_id=201}$ (employee))

viii) Delete all records from works table whose salary is less than 10000

works$\leftarrow$works-$\sigma_{salary<10000}$(works)

ix) Delete all records from works table

works$\leftarrow$works-$\Pi_{branch\_id,salary,cus\_id}$(works)

5. **By using the following schemas write relational algebraic expression and SQL statements. (underlined attributes represent primary key attributes)**

EMPLOYEE(EMPNO,NAME,ADDRESS)
PROJECT(PNO,PNAME)
WORKON(EMPNO,PNO)
PART(PARTNO,PARTNAME,QTY_ON_HAND)
USE(EMP_NO,PNO,PARTNO,NUMBER)

i)Listing all employees details who are not working yet

$\prod_{EMPNO,NAME,ADDRESS}$ (Employee) - $\prod_{EMPNO,NAME,ADDRESS}$ (Employee⋈WORKON)

ii) Listing Part Name and Quantity on hand those were used in DBMS project

$\prod_{PARTNAME,QTY\_ON\_HAND}$ (PART⋈ (USE⋈$\sigma_{PNAME="DBMS"}$(PROJECT)))

OR

$\prod_{PARTNAME,QTY\_ON\_HAND}$($\sigma_{PNAME="DBMS"}$ (PART⋈ (USE⋈ PROJECT)))

iii) List the name of projects that are used by employee from Kathmandu

$\prod_{PNAME}$($\sigma_{ADDRESS="Kathmandu"}$((EMPLOYEE⋈WORKON)⋈ PROJECT))

6. **Consider the following relations for order processing database application in a company.**

CUSTOMER(Cust#,Cname,City)

ORDER(Order#,Odate,Cust#,ord_Amt)

ORDER_ITEM(Order#,Item#,Qty)

ITEM(Item#,Unit_price)

SHIPMENT(Order#,Warehouse#,Ship_date)

WAREHOUSE(Warehouse#,City)

Answer the following queries in relational algebra.

i) List the order# and ship_date for all orders shipped from Warehouse number "W2".

$\prod_{Order\#,Ship\_date}$($\sigma_{warehouse\#="w2"}$ (SHIPMENT))

ii) List the warehouse information for which the customer named 'JOSE Copez' was supplied his orders.

$\prod_{warehouse\#,city}(\sigma_{Cname="JOSE\ Copez"}(CUSTOMER \bowtie (ORDER \bowtie (SHIPMENT \bowtie WAREHOUSE))))$

iii) List the orders that were not shipped within 30 days of ordering.

$TIMELY\_SHIPPED \leftarrow \sigma_{Ship\_date \leq Odate+30}(ORDER \bowtie SHIPMENT)$

$RESULT \leftarrow \prod_{Order\#}(ORDER) - \prod_{Order\#}(TIMELY\_SHIPPED)$

iv) List the order# for orders that were shipped from all warehouses in the network.

$\prod_{Order\#,\ Warehouse\#}(Shipment) \div \prod_{Warehouse\#}(\sigma_{City = "New\ York"}(Warehouse))$

7. consider the following database: [PU:2012 fall]

Student(sid,name,age)
Has(sid,cid)
College(cid,cname)

Write relational algebra expression to perform the following.

i) find the average age of student.

$\mathcal{G}_{avg(age)}(Student)$

ii) Display the name of student who studies in "QWERT" college.

$\prod_{name}(\sigma_{cname="QWERT"}(Student \bowtie (Has \bowtie College)))$

iii) Insert a new student.

$Student \leftarrow Student \cup \{104,"Roshan",29\}$

iv) Delete record of "ASDFG" college from college relation

$College \leftarrow College - (\sigma_{cname="ASDFG"}(College))$

v) Display name of students whose name begin from 'S.'

$\prod_{name}(\sigma_{name\ LIKE\ 'S\%'}(Student))$

8. Consider the following schema: [PU:2017 fall]

employee(person_name,street,city)

works(person_name,company_name,salary)

company(company_name,city)

manages(person_name,manager_name)

Given an expression in relational algebra to express each of the following queries.

i) Find the name of all employees who earn more than their managers.

$\Pi_{\text{person\_name}}$ ((works ⋈ manages) $\bowtie_{\text{manages.manager\_name=e2.person\_name ^ works.salary>e2.salary}}$ (ρ_{e2} (works)))

In SQL this will works

```
SELECT works.person_name
FROM works NATURAL JOIN manages
JOIN works AS e2 ON manages.manager_name = e2.person_name
AND works.salary > e2.salary;
```

ii) Find the name of all employees who live in the same city and on the same street as their managers.

$\Pi_{\text{person\_name}}$ ((employee ⋈ manages) $\bowtie_{\text{(manages.manager\_name=employee2.person\_name ^ employee.street=employee2.street ^ employee.city=employee2.city)}}$ ($\rho_{\text{employee2}}$ (employee)))

iii) Find the name of all employees with database that do not work for "NBL company".

$\Pi_{\text{person\_name}}$($\sigma_{\text{company\_name}\neq\text{"NBL company"}}$ (works))

iv) Find the name of all employees in the database who earn more than top earner at "NBL company in the database".

topearner← $\mathcal{G}_{\text{max(salary)}}$ ($\sigma_{\text{company\_name=" NBL company"}}$(works))
result← $\Pi_{\text{personname}}$($\sigma_{\text{salary>topearner}}$ (works))

9. Consider the relational database of figure below, where primary keys are underlined. Given an expression in the relational algebra to express each of the following queries. [PU:2014 spring]

Employee(person_name,street,city)
Works(person_name,bank_name,salary)
Bank(bank_name,city)
Manages(person_name,manager_name)

i) Find the total salary sum of all the banks.

$\mathcal{G}_{sum(salary)}$ (works)

ii) Modify the database so that Ram now lives in Kathmandu.

Employee$\leftarrow \prod_{person_name,street,"Kathmandu"} (\sigma_{person_name="Ram"}(Employee))$
U (Employee-$\sigma_{person\_name="Ram"}$(Employee))

iii) Find the name, street address and cities of residence of all employees who work for Nepal world Bank corporation and earn more than $10,000 per annum.

$\prod_{person\_name,street,city} (\sigma_{bank\_name="Nepal\ World\ Bank\ Corporation"\wedge salary>10000}(Employee \bowtie Works))$

Here, we assume attribute named salary represent annual salary.

iv) Delete all tuples in work relation for employee of small bank corporation.

Works$\leftarrow$Works - $\sigma_{bank\_name="Small\ Bank\ Corporation"}$(Works))

10. Consider the following relational database of figure below, Where primary keys are underlined. Given an expression in the relational algebra to express each of the following queries.

employee(person_name,street,city)
works(person_name,bank_name,salary)
bank(bank_name,city)
manages(person_name,manager_name)

i) Find the name of all employees who work for Nepal Rastra Bank and Salary greater than $10000.

$\Pi_{person\_name}(\sigma_{bank\_name="Nepal\ Rastra\ Bank" \wedge salary > 10000}(employee \bowtie works))$

ii) Find the name and cities of residence of all employees who work for Nepal Rastra Bank

$\Pi_{person\_name,city}(\sigma_{bank\_name = "Nepal\ Rastra\ Bank"}(employee \bowtie works))$

iii) Find name, street address, and cities of residence of all employees who work for Nepal Rastra Bank Corporation and earn more than $10000 per annum.

$\Pi_{person\_name,street,city}(\sigma_{bank\_name = "Nepal\ Rastra\ Bank\ corporation" \wedge salary > 10000*12}(employee \bowtie works))$

(Here we assume that attributed named salary represents monthly salary)

iv) Delete all tuples in work relation for employee of Nepal Rastra Bank

$works \leftarrow works - \sigma_{bank\_name = "Nepal\ Rastra\ Bank"}(works))$

11. Consider the student registration database comprising of below schema.

Student(CRN,Name,Gender,Address,Telephone)
Course(CourseID,CourseName,Hour,TeacherID)
Teacher(TeacherID,TeacherName,Office)
Registration(CRN,CourseID,Date)

i) Count the number of student registered subject in year 2015 gender wise.

$_{Gender}\mathcal{G}_{count(CRN)}(\sigma_{EXTRACT(YEAR\ FROM\ Date)=2015}(Student))$

ii) Show student details taught by teacher Ronit Shreshta.

$\Pi_{CRN,Name,Gender,Address,Telephone}(\sigma_{TeacherName="Ronit\ Shreshta)}(((Student \bowtie Registration) \bowtie course) \bowtie Teacher))$

iii) Delete student information taught by teacher N.Mathema

$Student \leftarrow Student - \Pi_{CRN,Name,Gender,Address,Telephone}(\sigma_{TeacherName="N.Mathema"}(((Student \bowtie Registration) \bowtie course) \bowtie Teacher))$

12. Consider the following relational schema **[PU:2018 Fall]**

Department(**DepartmentID**,DepartmentName)
Designation(**DesignationID,**DesignationName,Salary)
Employee(**EmpID**,EmpName,Gender,DesignationID,DepartmentID)
Allowance(**AllowanceID**,AllowanceName)
Allowance_Details(DetailID,EmpID,AllowanceID,Amount)

Write the relational algebraic expression for the following task:

i) Find the number of employees department-wise.

$_{\text{DepartmentName}}\mathcal{G}_{\text{count(eid)}}$ (Employee⋈ Department)

ii) List the employee details whose salary is above 50000.

$\Pi_{\text{EmpID,EmpName,Gender,DesignationID,DepartmentID}}(\sigma_{\text{(Salary+amount)>50000}}((\text{Employee}\bowtie \text{Designation}) \bowtie \text{Allowance}))$

iii) List the employee those who are getting house allowance.

$\Pi_{\text{EmpName}}(\sigma_{\text{AllowanceName="houseallowance"}}((\text{Employee}\bowtie \text{Allowance\_Details}) \bowtie \text{Allowance}))$

13. Consider the relational database model

Users(uid,cname,city)
Items(itemid,itemname,city,quantity,price)
Manager(mid,aname,city)
Query(queryno,uid,mid,itemid,query_details,hitratio)
[PU:2018 spring]

Write the relational algebraic expressions for the following tasks.

i) Find all (queryno,uid) pairs for query with a hit ratio value greater than 500.

$\Pi_{\text{queryno,uid}}(\sigma_{\text{hitratio>500}}(\text{Query}))$

ii) Find all item names of items in Pokhara ordered with query details as Pokhara details.

$\Pi_{\text{itemname}}(\sigma_{\text{querydetails="Pokhara details"^city="Pokhara"}}(\text{Items}\bowtie \text{Query}))$

iii) Find item ids of items ordered through manager 35 but not through manager 27.

$\Pi_{\text{itemid}}(\sigma_{\text{mid=35}}(\text{Query})) - \Pi_{\text{itemid}}(\sigma_{\text{mid=27}}(\text{Query}))$

14. Using the following schema represent the following queries using Relational algebra.

PROJECT(Projectnum,ProjectName,ProjectType,ProjectManager)
EMPLOYEE(Empnum,Empname)
ASSIGNED_TO(Projectnum,Empnum)

[PU:2019 spring]

i) Find employee details working on project name starts with 'L'.

$\prod_{Empnum,Empname}(\sigma_{ProjectName\ Like\ 'L\%'}$ ((EMPLOYEE⋈ ASSIGNED_TO) ⋈PROJECT))

ii) List all the employee details who are working under project manager "Roshan".

$\prod_{Empnum,Empname}(\sigma_{ProjectManager="Rohan"}$((EMPLOYEE⋈ ASSIGNED_TO) ⋈PROJECT))

iii) List the employes who are still not assigned with any project.

$\prod_{Empnum,Empname}$(EMPLOYEE) - $\prod_{Empnum,Empname}$(EMPLOYEE⋈ ASSIGNED_TO)

iv) List the employees who are working in more than one project.

temp←$_{Empname}\mathcal{G}_{count\ (Projectnum)}$ (Employee ⋈ ASSIGNED_TO)

result← $\prod_{Empname}((\sigma_{Projectnum>1}(temp))$

15. Write relational algebra for the following schemas. (Underlined indicates Primary key)[PU:2020 spring]

Employee(**Emp_No**,Name,Address)
Project(**PNO**,Pname)
Workon(**Emp_No**,PNo)
Part(**Partno**,Part_name,Qty_on_hand)
Use(**Emp_No**,**PNO**,**Partno**,Number)

i)Listing all employees details who are not working yet.

$$\prod_{Emp\_No,Name,Address} (Employee) - \prod_{Emp\_No,Name,Address} (Employee \bowtie Workon)$$

ii) Listing Part Name and Quantity on hand those were used in DBMS project

$$\prod_{Part\_name,Qty\_On\_hand} (Part \bowtie (Use \bowtie \sigma_{Pname="DBMS"}(Project)))$$

iii) List the name of projects that are used by employee from London

$$\prod_{Pname}(\sigma_{Address="London"}((Employee \bowtie Workon) \bowtie Project))$$

iv) Modify the database so that Jones now live in USA.

$$Employee \leftarrow \prod_{Emp\_No,Name,"USA"}(\sigma_{Name="Jones"}(Employee)) \cup (Employee - \sigma_{Name="Jones"}(Employee))$$

v) Update address of an employee 'Japan' to 'USA'

$$Employee \leftarrow \prod_{Emp\_No,Name,"USA"}(\sigma_{Address="Japan"}(Employee)) \cup (Employee - \sigma_{Address="Japan"}(Employee))$$