Relational Algebra

Relational Algebra is a formal query language used in databases to retrieve, manipulate, and modify data stored in relational databases. It provides a set of operations that take one or more relations (tables) as input and produce a new relation as output. These operations help in defining complex queries in a structured way.

Basic Operations in Relational Algebra

1. Selection (σ) → row filtering using conditions

-> Selection is an operation that retrieves a subset of rows from a relation (table) that satisfy a specified predicate (condition).
-> It filters tuples (rows) but does not alter the schema (columns).

→ $\sigma_{predicate}(Relation)$

→ Example : $\sigma_{dname = \text{"IT"}}(dept)$

-> Fetches all rows from the dept table where the department name is "IT
-> SELECT * FROM dept WHERE dname = 'IT';

→ operators include $=, \neq, <, \leq, >, \geq$

- Example : salary > 50000

→ AND (∧) , or (∨) , not (~)

both true | one true | negation

example :

$$\sigma_{dept\ no = 1\ \wedge\ salary > 50{,}000}(emp)$$

-> Fetches all rows from employees in department 1 with a salary above 50,000.

-> SELECT * FROM emp WHERE deptno = 1 AND salary > 50000;

$$\sigma_{dept\ no = 1\ \vee\ salary > 500}(emp)$$

-> Fetches employees in department 1 or with a salary above 50,000.

-> SELECT * FROM emp WHERE deptno = 1 OR salary > 50000;

2. Project Operation

$\longrightarrow$ handle column selection

-> Project Operation is denoted by the uppercase Greek letter Π (Pi).

-> It selects specific columns from a relation (table) and discards others.

-> Equivalent to the SELECT clause in SQL (but only for column selection, not filtering rows).

$$\Pi_{column1,\ column2 \ldots}(Relation)$$

-> Retrieve id, name, and salary from the emp table.

$$\Pi_{emp\_id,\ ename,\ salary}(emp)$$

-> SELECT emp_id, ename, salary FROM emp;

-> Retrieve name and salary of employees in department 1.

$$\Pi_{ename,\ salary}(\sigma_{deptno=1}(emp))$$

->SELECT ename, salary FROM emp WHERE deptno = 1;

π	Projection	SELECT (columns)
σ	Selection	WHERE (condition)

3. Rename Operation (ρ)

AS in SQL

-> Used to rename relations (tables) or attributes (columns) in relational algebra.

$$\rho_{NewName}(Expression)$$

-> Rename the result to $\sigma_{age>30}(emp)$ to "senior staff"

-> $\rho_{senior\_staff}(\sigma_{age>30}(emp))$

-> rename the "emp-id" column to "id" in the "employee" table

-> $\rho_{employee(id,\, name,\, dept)}(employee)$

(renaming attributes)

4. Union Operation (U)

-> Combines two relations into one

-> Conditions for validity

- -> Both relations must have the same number of attributes.
- -> Corresponding attributes must have compatible domains (same data type).

-> Duplicate tuples are automatically eliminated.

Syntax : $R \cup S$

-> Find courses taught in Fall 2009 or Spring 2010.

$$\Pi_{course\_id}\left(\sigma_{semester = \text{"Fall"} \wedge year = 2009}(section)\right) \cup$$

$$\Pi_{course\_id}\left(\sigma_{semester = \text{"Spring"} \wedge year = 2010}(section)\right)$$

-> SELECT course_id
FROM section
WHERE semester = 'Fall' AND year = 2009
UNION
SELECT course_id
FROM section
WHERE semester = 'Spring' AND year = 2010;

5. Set Difference Operation (−)

-> Returns tuples present in the first relation but not in the second.

-> Conditions for validity:

-> Relations must be union-compatible

-> Schema of the result matches the schema of the input relations.

-> Syntax: R−S

-> Authors in Books but not in Articles

$$\Pi_{author}(Books) - \Pi_{author}(articles)$$

```
SELECT author FROM Books
MINUS
SELECT author FROM Articles;
```

-> Courses in Fall 2009 but not in Spring 2010

↓

remove ∪ with − in above

6. Cartesian Product Operation (×)

-> Combines every tuple of one relation with every tuple of another relation.

-> Results in a large relation with all possible tuple combinations.

-> Often paired with selection (σ) to form joins (e.g., natural join).

-> R×S

-> Combine Dept and Emp Tables

-> dept×emp

-> SELECT * FROM dept CROSS JOIN emp;

no ON condition like normal JOIN

-> Employee Names in IT Department

$$\Pi_{ename}\left(\sigma_{dname = \text{“IT”}}(dept \times emp)\right)$$

-> SELECT ename FROM dept CROSS JOIN emp
WHERE dname = 'IT';

Set Intersection (∩)

-> Returns tuples present in both relations

-> r and s must be union-compatible (same schema).

-> r ∩ s

-> Task: Find courses taught in both Fall 2009 and Spring 2010.

→ osi ko — ma ∩ same #

```
-> SELECT course_id FROM section
WHERE semester = 'Fall' AND year = 2009
INTERSECT
SELECT course_id FROM section
WHERE semester = 'Spring' AND year = 2010;
```

Natural Join (⋈)

-> Combines two relations by equating common attributes and removing duplicates.

-> Steps:

1. Compute Cartesian product.
2. Select tuples with matching values in common attributes.
3. Remove duplicate attributes

$$\boxed{r \bowtie s}$$

-> Task: Find instructor names and their taught course IDs.

-> SELECT name, course_id
FROM instructor
NATURAL JOIN teaches;

$$\Pi_{name,\ course\_id}\ (instructor \bowtie teaches)$$

Theta Join (θ-Join)

-> Generalizes natural join with any condition θ (not just equality).

-> Combines Cartesian product and selection into one operation.

$$\text{Syntax: } r \bowtie_{\theta} s = \sigma_{\theta}(r \times s)$$

-> Task: Find departments named "IT" with employees earning > 50,000.

-> SELECT * FROM dept CROSS JOIN emp
WHERE dept.dname = 'IT' AND emp.salary > 50000;

$$\sigma_{dept.name = \text{"IT"} \wedge emp.salary > 50000}\ (dept \times emp)$$

$dept \bowtie_{dept\_name = \text{"IT"} \wedge emp\_salary > 50{,}000} emp$

Outer JOINS

Operation	Symbol (Relational Algebra)	Description
Left Outer	$r ⟕ s$	All tuples from r, NULLs for non-matching s.
Right Outer	$r ⟖ s$	All tuples from s, NULLs for non-matching r.
Full Outer	$r ⟗ s$	All tuples from both r and s, NULLs for non-matches.

Generalized Projection

-> Extends projection to include arithmetic/string operations in the projection list.

- **Syntax**:

$$\Pi_{F1,F2,...,Fn}(E)$$

  - E: Input relation.
  - $F1, F2, \ldots, Fn$: Expressions involving attributes/constants (e.g., $\text{salary} \div 12$).

example:

$$\Pi_{emp\_id,\ emp\_name,\ salary \div 12}(emp)$$

→ gives emp-id, name and monthy salary of each employee.

Aggregate Functions

-> Compute a single value from a collection (e.g., sum, average).

- **Syntax**:

$$\mathcal{G}_{\text{function(attribute)}}(E)$$

Function	Description	Example (Relational Algebra)	SQL Equivalent
SUM	Sum of values	$\mathcal{G}_{\text{SUM(salary)}}(\text{emp})$	`SELECT SUM(salary) FROM emp;`
AVG	Average of values	$\mathcal{G}_{\text{AVG(salary)}}(\text{emp})$	`SELECT AVG(salary) FROM emp;`
COUNT	Total number of elements	$\mathcal{G}_{\text{COUNT(emp\_id)}}(\text{emp})$	`SELECT COUNT(emp_id) FROM emp;`
COUNT-DISTINCT	Distinct elements	$\mathcal{G}_{\text{COUNT-DISTINCT(emp\_id)}}(\text{emp})$	`SELECT COUNT(DISTINCT emp_id) FROM emp;`
MIN/MAX	Minimum/Maximum value	$\mathcal{G}_{\text{MIN(salary)}}(\text{emp})$	`SELECT MIN(salary) FROM emp;`

Group By Operation

> Apply aggregate functions to groups of tuples.

- **Syntax**:

$$\text{grouping\_attributes}\ \mathcal{G}_{\text{aggregate\_function(attribute)}}(E)$$

-> Find average salary per department.

$$\text{dept\_name}\ \mathcal{G}_{\text{avg(salary)}}(\text{emp})$$

-> SELECT dept_name, AVG(salary)
FROM emp
GROUP BY dept_name;

3. a) Consider the following relational schema. [4×2]

Employee(empid,ename,address, title, headid)
Project(pid, pname, budget, location)
Work(empid, pid, responsibility,duration)
Payment(title, salary)

i) Write relational algebra to find the name and salary of employees working in 'kathmandu'.

ii) Write the relational algebra to show the employee name along with their head name. (Note: the data in headid is empid of their corresponding head)

iii) Write relational algebra to display the details of those employee who live in the same location of their project.

iv) Display employee's title, name along with their project name, salary if the project duration is more than 5 years.

i) Name and salary of employees working in 'Kathmandu':

join

$$\pi_{ename, salary}\left(\sigma_{location = \text{"kathmandu"}}\left(Project \bowtie Work \bowtie Employee \bowtie Payment\right)\right)$$

(imagine like in SQL Joins)

ii) Employee name along with their head name:

$$\pi_{ename, HeadName}\left(\sigma_{E1.headid = E2.empid}\left(\rho_{E1(empid, ename, address, title, headid)}(Employee) \times \rho_{E2(empid, HeadName, address, title, headid)}(Employee)\right)\right)$$

iii) Employees living in the same location as their project:

$$\pi_{empid, ename, address, title, head\text{-}id}\left(\sigma_{address = location}\left(Employee \bowtie Work \bowtie Project\right)\right)$$

iv) Title, name, project name, salary for duration >5 years:

$\Pi_{title, ename, pname, salary}(\sigma_{duration > 5}(Employee \bowtie$

$Work \bowtie Project \bowtie Payment))$

b) Consider the following relational database.

Client (Cid, Lid, Cname, birthYear, caseField)
Lawyer (Lid, Fname, Lname, Speciality, Salary, startingYear)
Firm (Fname, City, managerName)

i) Write relational algebra expression to find names of clients that were born after 1980 and their case field was "Traffic".
ii) Write relational algebra expression to find names of clients who were presented only by lawyers whose salary is at 8000
iii) Write relational algebra expression to find all pairs of client id and firm name such that no lawyer of that law firm presented this client and this client had some case in a field other than "Divorce".

i) Names of clients born after 1980 with case field "Traffic"

$\Pi_{cname}(\sigma_{birthYear > 1980 \wedge caseField = \text{"Traffic"}}(Client))$

ii) Names of clients represented only by lawyers with salary 8000:

$\Pi_{cname}(\Pi_{cname}(Client) - \Pi_{cname}(\sigma_{salary \neq 8000}$

$(Client \bowtie_{(Client.Lid = Lawyer.Lid)} Lawyer)$

(all client − lawyer client whose salary is not 8000

= lawyer client whose salary is 8000

iii) Pairs of client ID and firm name where no lawyer from the firm represents the client, and the client has a non-"Divorce" case:

$$\Pi_{cid,\ Fname}\left(\sigma_{case\ Field \neq \text{"Divorce"}}(Client) \times Firm\right)$$

no direct join from client to firm so

$$- \ \Pi_{cid,\ Fname}\left(Client \bowtie_{(client.lid = Lawyer.lid)} Lawyer\right)$$

all the lawyers that represent the client

3. Consider the following relational database model:

Passenger (pid, pname, pgender, pbirthplace)

Agency (aid, aname, acity)

Flight (fid, fdate, time, source, destination)

Booking (pid, aid, fid, bookdate, amount)

... of bookings for current date.

For the relational database model given in the Question No. 3. Write relational algebraic expression for the following: [2×4]

a) Find the details of all flights to "Kathmandu".
b) Find name of all passengers who have booked at least one flight.
c) List the name of all agencies who have made highest booking till date.
d) Find all the passenger who have booked from agency "Esewa".

a) Find the details of all flights to "Kathmandu".

$$\Pi_{fid,\ fdate,\ time,\ source,\ destination}\left(\sigma_{destination = \text{"Kathmandu"}}(Flight)\right)$$

b) Find name of all passengers who have booked at least one flight.

$$\Pi_{pname}\left(Passenger \bowtie_{(passenger.pid = booking.pid)} Booking\right)$$

c) List the name of all agencies who have made the highest booking till date.

→ book ko solution and gpt ko ni bujena

d) Find all passengers who have booked from agency "Esewa".

$\Pi_{pname} (\sigma_{aname = 'Esewa'} (Agency \bowtie Booking \bowtie Passenger))$

Consider the relational model.
Employee (empid, empname, address, title)
Project (pid, pname, budget, location)
Assignment(empid, pid, responsibility, duration)
Payment (title, salary)

CAD/CAM project.

f) Write Relational Algebra to find the name and salary of employees working in Kathmandu.

$\Pi_{empname, salary} (\sigma_{location = \text{"kathmandu"}} (Project \bowtie_{(pid)} Assignment \bowtie_{(empid)} Employee \bowtie_{(title)} Payment))$

4. Write relational algebra. [Consider following relations.] [8]
Emp (<u>Eid</u>, Ename, Address, Salary, Dptid)
Depart(<u>Dptid</u>, Dname)

a) Insert a single record in Emp table.(100,'Ram','Balaju',10000,5)
b) Retrieve the record of employee who earns more than 10000 in computer department.
c) Increase the salary of all employee by 10 percent.
d) Delete all the record of employee who are from ELX department. (Dptid='ELX')

a) Insert a single record into Emp:

$$Emp \leftarrow Emp \cup \{(100, 'Ram', 'Balaju', 100000, 5)\}$$

b) Retrieve employees earning >10000 in the "Computer" department:

$$\Pi_{Eid, Ename, Address, Salary, Dptid}(\sigma_{salary>10000 \wedge Dname = 'computer'}(Emp \bowtie Depart))$$

c) Increase all salaries by 10%:

$$Emp \leftarrow \Pi_{Eid, Ename, Address, salary \times 1.1 \text{ AS } salary, Dptid}(Emp)$$

d) Delete employees from department with Dptid = 'ELX':

$$Emp \leftarrow Emp - \sigma_{Dptid = 'ELX'}(Emp)$$

→ all rows at Dptid = 'ELX'

Consider the following relational data model

Employee (empid, empname, address, title)
Project (pid, proj_name, budget, location)
Assignment (empid, pid, responsibility, duration)
Payment (title, salary)

(i) Find the name of employees working for more than 2 years in "Software" project and earning more than 1000 K.
(ii) Find the names of employees working in "PCB Fabrication" project other than John.
(iii) Find the salaries of Engineers working in "Fabrication" projects.

Find the name of employees working for more than 2 years in "Software" project and earning more than 1000 K:

$$\Pi_{empname}\left(\sigma_{proj = \text{"software"} \wedge duration > 2 \wedge salary > 1000k}\left(Project \bowtie Assignment \bowtie Employee \bowtie payment\right)\right)$$

(ii) Find the names of employees working in "PCB Fabrication" project other than John:

$$\Pi_{empname}\left(\sigma_{proj = \text{"PCB Fabrication"} \wedge empname \neq \text{"John"}}\left(Project \bowtie Assignment \bowtie Employee\right)\right)$$

(iii) Find the salaries of Engineers working in "Fabrication" projects:

$$\Pi_{salary}\left(\sigma_{proj = \text{"Fabrication"} \wedge title = \text{"Enginery"}}\left(Project \bowtie Assignment \bowtie Employee \bowtie payment\right)\right)$$

Consider the following relational database [2×4]

sailor (sailorid, sname, rating, age)

boat (boatid, boatname, color)

reserves (sailorid, boatid, date)

'rite relational algebra expressions for the following:

i) Find the names of sailor who has reserved boat number 105.
ii) Find the names of sailors who have reserved a red boat.
iii) Find the names of all sailor who have reserved either a red boat or a green boat.

i) Names of sailors who reserved boat 105:

$$\Pi_{sname} \left(\sigma_{boatid = 105} \left(sailor \bowtie reserves \right) \right)$$

ii) Names of sailors who reserved a red boat:

$$\Pi_{sname} \left(\sigma_{color = \text{"red"}} \left(boat \bowtie reserves \bowtie sailor \right) \right)$$

iii) Names of sailors who reserved a red or green boat:

$$\Pi_{sname} \left(\sigma_{color = \text{"red"}} \vee color = \text{"green"} \left(boat \bowtie reserves \bowtie sailor \right) \right)$$

4. Consider the following relational database model

Author(<u>a name</u>, citizenship, birthYear)
Topic(<u>isbn</u>, subject)
Instock(<u>isbn, libname</u>, quantity)
Book(<u>isbn</u>, title, a name)
Branch(<u>libname</u>, city)

Write relational algebra expressions for the following: [2×4

a) Give the cities where each book is held.
b) Give the title and author of each book of whichat least two copies are held in a branch located in Kathmandu.
c) Delete those books that are from author 'xyz'
d) List total no. of available books of each subject.

a) Cities where each book is held:

$$\Pi_{isbn, city} \left(Branch \bowtie Instock \right)$$

b) Title and author of books with ≥2 copies in Kathmandu branches:

$$\Pi_{title, a\_name} \left(\sigma_{quantity \geq 2 \wedge city = \text{"kathmandu"}} \left(Instock \bowtie Branch \bowtie Book \right) \right)$$

c) Delete books by author 'xyz':

$$Book \leftarrow Book - \sigma_{a\text{-}name = \text{"xyz"}}(Book)$$

d) Total available books per subject:

$$\leftarrow {}_{Subject}\mathcal{G}_{sum(quantity) \rightarrow total}(Topic \bowtie Instock)$$

group by subject

4. Consider the following relational database [2×4]

Account (<u>account-number</u>, branch-name, balance)
Branch (<u>branch-name</u>, branch-city, assets)
Customer (<u>cust-name</u>, cust-street, cust-city)
Loan (<u>loan-number</u>, branch-name, amount)
Depositor (<u>cust-name</u>, <u>account-number</u>)
Borrower (<u>cust-name</u>, <u>loan-number</u>)

Write the relational algebra expressions for the following:

(i) Find the names of customers who has loan at "Koteshwor" branch.
(ii) Find the largest account balance.
(iii)Find the names of all depositors along with their account number, street and city address.

(i) Names of customers with loans at "Koteshwor" branch:

$$\Pi_{cust\text{-}name}(\sigma_{branch\text{-}name = \text{"Koteshwor"}}(Loan \bowtie Borrower))$$

(ii) Largest account balance:

$$\mathcal{G}_{max(balance) \rightarrow max\text{-}balance}(Account)$$

(iii) Depositors with account number, street, and city:

$$\Pi_{cust\text{-}name,\ account\text{-}number,\ cust\text{-}street,\ cust\text{-}city}(Depositor \bowtie Customer)$$

3. a) Consider the following relational data model.

Employee(empid, name, address, manager_id)

Department(deptid, dname)

Project(pid, title, budget, deptid)

Works_on(empid, pid, hours)

Write down the relational algebraic expression for the following:

i. Find the names of all employee from computer department [redacted]

ii. Find the names of all the employees who works on project with budget more than 50000.

iii. Find the total number of projects from each department along with the department name.

1. Names of all employees from the "Computer" department:

$$\Pi_{name}(\sigma_{dname = \text{"Computer"}}(Department \bowtie Project \bowtie Work\text{-}on \bowtie Employee))$$

2. Names of employees working on projects with budget > 50000:

$$\Pi_{name}(\sigma_{budget > 50000}(Project \bowtie Works\text{-}on \bowtie Employee))$$

3. Total number of projects per department (with department name):

$$_{dname}\mathcal{G}_{COUNT(pid) \rightarrow total\_projects}(Project \bowtie Department)$$

group by dname

Consider the following relational schema

Employee (Ename, street, city)

Works (Ename, company_name, salary)

Company (company_name, city)

Manages (Ename, manager_name)

a) Write the queries in Relational Algebra. [2×3]

i. Find all the employees name who work in 'NMB bank'.

ii. Find all the employee names who live in the same city as their company is located.

iii. Find the name and city of those employees whose salary is greater than 30000 and lives in 'ktm' city.

i. Employees who work at 'NMB bank':

$$\Pi_{Ename} \left(\sigma_{company\_name = \text{'NMB bank'}} (works) \right)$$

ii. Employees living in the same city as their company:

$$\Pi_{Ename} \left(\sigma_{Employee.city = company.city} (Employee \bowtie works \bowtie company) \right)$$

iii. Employees earning >30000 and living in 'ktm' city:

$$\Pi_{Ename,\ city} \left(\sigma_{salary > 30000 \wedge city = \text{'ktm'}} (Employee \bowtie works) \right)$$

3. Consider the following relational scheme:

Account (account number, branch_name, balance)

Branch (branch name, branch_city, assets)

Customer (Customer name, customer_street, customers_city)

Loan (loan number, branch_name, amount)

Depositor (customer name, account number)

Borrower (customer name, loan number)

are more than $10,00,000 in the city.

e) Write relational algebra expression to count the number of accounts in each branch.

f) Write relational algebra expression to delete all loans less than $1,000 in amount

e) Count the number of accounts in each branch:

$_{branch\_name}\gamma_{\text{COUNT}(account\_number) \rightarrow total\_accounts}(Account)$

f) Delete all loans with amount < $1,000:

$Loans \leftarrow Loan - \sigma_{amount < 1000}(Loan)$

3. Consider the following relational database model

Employee(eid, name, address, supervisor_eid)
Department(dept_id, name)
Project(pid, title, dept_id)
Works_on(eid, pid, hours)

Write relational algebra expressions for the following: [2 X 4 = 8]

a) List the titles of all projects along with the department names.
b) Find the names of all employees who live in "Kathmandu" and are supervised by employee who also lives in "Kathmandu".
c) Increase the working hours of all employees who work in the "Voter registration" project by 5 hrs.
d) Find the total number of employees involved in each project along with the project title.

a) List the titles of all projects along with the department names:

$\Pi_{title,\, name}(Project \bowtie Department)$

b) Names of employees in Kathmandu supervised by someone in Kathmandu:

$\Pi_{name}(\sigma_{address = \text{'Kathmandu'}}(Employee) \bowtie_{(supervisor\_eid = eid)}$

$\sigma_{address\, \text{'Kathmandu'}}(Employee))$

c) Increase the working hours of all employees who work in the "Voter registration" project by 5 hrs.

$$Works\_on \leftarrow (works\_on - (\sigma_{title='voter\ registration'}(works\_on \bowtie project)) \cup \Pi_{eid,\ pid,\ hours+5}(\sigma_{title='voter\ registration'}(work\_on \bowtie project))$$

d) Find the total number of employees involved in each project along with the project title.

$$\Pi_{title,\ total\_employees}(\ {}_{pid}\mathcal{G}_{COUNT(eid) \rightarrow total\_employees}(works\_on \bowtie Project))$$

3. Consider the following relational database model

Employee(eid, name, address, supervisor_eid)
Department(dept_id, name)
Project(pid, title, dept_id)
Works_on(eid, pid, hours)

Write relational algebra expressions for the following: [2 X 4 = 8]

a) List the name of all employees from Computer department along with the name of their supervisor.
b) Find the name of all employees who work on the "Network monitoring" project for more than 15 hours.
c) Delete all projects which belong to the "Electrical" department.
d) Find the total number of projects from each department, along with the department name.

yesto wala eq do like this

a) List the name of all employees from Computer department along with the name of their supervisor.

$$\Pi_{e.name,\ s.name}(\sigma_{d.name='computer'}(Department \bowtie Project) \bowtie works\_on \bowtie \rho_e(Employee) \bowtie_{(e.supervisor\_eid = s.eid)} \rho_s(Employee))$$

b) Find the name of all employees who work on the "Network monitoring" project for more than 15 hours.

$$\Pi_{employee.name} (\sigma_{project.title = 'Network\ monitoring' \wedge work\text{-}on.hours \geq 15} (Project \bowtie Works\text{-}on \bowtie Employee))$$

c) Delete all projects which belong to the "Electrical" department.

$$Project \leftarrow project - (\sigma_{Department.name = 'Electrical'} (Project \bowtie Department))$$

d) Find the total number of projects from each department, along with the department name.

$$\Pi_{Department.name,\ total\_projects} ({}_{dept\text{-}id} \mathcal{G}_{COUNT(pid) \rightarrow total\_projects} (Projects \bowtie Department))$$

---

3. Consider the following relational database model

Product (pid, name, price, category, maker-cid)
Purchase (buyer-ssn, seller-ssn, quantity, pid)
Company (cid, name, stock price, country)
Person(ssn, name, phone number, city)

Write relational algebra expressions for the following: [2 X 4]

a) Find the *ssn* and *name* of all people who have purchased products of category "telephone"
b) List the *pid* and *name* of all products which is more expensive than $500 and made in China.
c) Increase the price of all products of "television" category by 10%.
d) List the *ssn* and *name* of each seller along with the total quantity of products sold.

a) Find the ssn and name of all people who have purchased products of category "telephone":

$$\Pi_{person.ssn,\ person.name} (\sigma_{product.category = "telephone"} (Product \bowtie Purchase \bowtie Person))$$

$\bowtie_{buyer\text{-}ssn = ssn}$

b) List the pid and name of all products more expensive than $500 and made in China:

$$\pi_{pid,\, name}\left(\sigma_{price > 500 \wedge country = 'china'}\left(Product \bowtie_{maker\text{-}cid = cid} company\right)\right)$$

c) Increase the price of all products of "television" category by 10%:

$$Product \leftarrow \overbrace{\left(product - \sigma_{category = 'television'}(Product)\right)}^{optional} \cup$$

$$\pi_{pid,\, name,\, price \times 1.10,\, category,\, maker\text{-}cid}\left(\sigma_{category = 'television'}(Product)\right)$$

d) List the ssn and name of each seller along with the total quantity of products sold:

$$\pi_{ssn,\, name,\, total}\left({}_{seller\text{-}ssn}\gamma_{sum(quantity) \rightarrow total}\left(Purchase \bowtie_{ssn = seller\text{-}ssn} Person\right)\right)$$

---

3. Consider the following relational database model

employee (*person-name*, *street*, *city*)
works (*person-name*, *company-name*, *salary*)
company (*company-name*, *city*)
manages (*person-name*, *manager-name*)

Write relational algebra expressions for the following: [2 X 4 = 8]

a) Find the names and cities of residence of all employees who work for First Bank Corp.
b) Find the names of all employees who live in the same city as the company for which they work.
c) Modify the database so that the employee Jones now lives in Newtown.
d) Find the average salary offered by each company.

a) Find the names and cities of residence of all employees who work for First Bank Corp.

$$\Pi_{person\text{-}name,\, city} \left(\sigma_{company\text{-}name = \text{'First Bank Corp'}} \left(employee \bowtie works \right) \right)$$

b) Find the names of all employees who live in the same city as the company for which they work.

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

c) Modify the database so that the employee Jones now lives in Newtown.

$$employee \leftarrow \left(employee - \sigma_{person\text{-}name = \text{'Jones'}} (employee) \right) \cup \left(\Pi_{person\text{-}name,\, street,\, city = \text{'Newtown'}} \left(\sigma_{person\text{-}name = \text{'Jones'}} (employee) \right) \right)$$

d) Find the average salary offered by each company.

$$_{company\text{-}name}\, \mathcal{G}_{\,AVG(salary) \rightarrow average\_salary} (works)$$

3. Consider the following relational database model

employee (person-name, street, city)
works (person-name, company-name, salary)
company (company-name, city)
manages (person-name, manager-name)

Write relational algebra expressions for the following: [2 X 4 = 8]

a) Find the names and street address of all employees who work for First Bank Corp. and earn more than $10,000 per annum.
b) Find the names of all employees who do not work for First Bank Corp.
c) Give all employees working at First Bank Corp. a 10 % salary raise.
d) Count the number of employees in each company.

a) Find the names and street address of all employees who work for First Bank Corp. and earn more than \$10,000 per annum.

$$\Pi_{\text{person-name, street}}\left(\sigma_{\text{company-name} = \text{'First Bank Corp'} \wedge \text{salary} > 1000}(\text{employee} \bowtie \text{works})\right)$$

b) Find the names of all employees who do not work for First Bank Corp.

$$\Pi_{\text{person-name}}(\text{works}) - \Pi_{\text{person-name}}\left(\sigma_{\text{company-name} = \text{'First Bank Corp'}}(\text{works})\right)$$

c) Give all employees working at First Bank Corp. a 10% salary raise.

$$\text{works} \leftarrow \left(\text{works} - \sigma_{\text{company-name} = \text{"First Bank Corp"}}(\text{works})\right) \cup \Pi_{\text{person-name, company-name, salary} \times 1.10}\left(\sigma_{\text{company-name} = \text{"First Bank Corp"}}(\text{works})\right)$$

better way

$$\text{works} \leftarrow \Pi_{\text{person-name, company-name, } \dots}(\text{works})$$

(directly)

d) Count the number of employees in each company.

$$_{\text{company-name}}\mathcal{G}_{\text{COUNT(person-name)} \rightarrow \text{employee\_count}}(\text{works})$$

7. Consider the following relational database model:
Employee (eid, ename, email, phone, address)
Department (dept_id, dname, block)
Designation (did, title,salary)
Works (eid, dept_id, did, effective_date) [2078 Bhadra]

Write relational algebra expression for the following:

i. **List all the designation title with starting letter 'M'.**

Ans: $\Pi_{title}(\sigma_{title\ like\ \text{"M\%"}}(Designation))$