INTRODUCTION TO RELATIONAL ALGEBRA

Query Language
Define data retrieval operations for relational model
Categories of languages
Procedural: What you want and how to get it
Non-procedural, or declarative: What you want (without how)
SQL: High-level language for relational algebra.

Relational Algebra

- The **relational algebra** is a **procedural query language**
- It consists of a set of operations that take one or two relations as input and produce a new relation as their result.
- These operations enable a user to specify basic **retrieval requests (or queries)**

Cont.

- The fundamental operations in the relational algebra are *select, project, union, set difference, Cartesian product, and rename*

- The **select, project, and rename** operations are called *unary operations*, *because* they operate on one relation

- The other three operations operate on pairs of relations and are, therefore, called *binary operations*

Unary and Binary Operations

- **Unary**
  - **selection**
  - **projection**
  - **Rename**
- **Binary**
  - **Union,**
  - **difference,**
  - **Cartesian product**
  - **Join operations**

selection

- The selection operation works on a single relation R and defines a relation that contains only those tuples of R that satisfy the specified condition

$$\sigma_{Selection\_Criteria}(Input)$$

A relation instance

Manipulates data in a single relation

The selection operator specifies the tuples to retain through selection criteria.

A boolean combination (i.e. using V and Λ) of terms

Attribute op constant or attribute1 op attribute2

< , <=, =, ≠, >=, or >

Example

Consider the relation STUDENT shown later:

STUDENT

Student Roll. No	Name	GPA
001	Aravind	7.2
002	Anand	7.5
003	Balu	8.2
004	Chitra	8.0
005	Deepa	8.5
006	Govind	7.2
007	Hari	6.5

Query 1: List the Roll. No, Name, and GPA of those students who are having GPA of above 8.0

Query expressed in relational algebra as $\sigma_{GPA} > 8$ (Student).

The result of the earlier query is:

Student Roll. No	Name	GPA
003	Balu	8.2
005	Deepa	8.5

Query 2: Give the details of first four students in the cl

Relational algebra expression is $\sigma_{\text{Roll. No}} \leq (\text{student})$.

Table as a result of query 2 is

Student Roll. No	Name	GPA
001	Aravind	7.2
002	Anand	7.5
003	Balu	8.2
004	Chitra	8.0

Select operator: picks certain rows

Students with GPA>3.7

$\sigma_{GPA > 3.7}$ Student

σ_{cond} Rel.

Students with GPA>3.7 and HS<1000

$\sigma_{GPA > 3.7 \wedge HS < 1000}$ Student

Applications to Stanford CS major

$\sigma_{cName = 'Stanford' \wedge major = 'cs'}$ Apply

College

cName	state	enr
William	CA	500
Stanford	CA	400
MIT	MA	300
UoI	IL	400

Student

sID	sName	GPA	HS
1	Victor	8.6	700
~~2~~	~~Mike~~	~~2.2~~	~~200~~
~~3~~	~~William~~	~~7.3~~	~~1500~~
4	Jack	5.0	700

Apply

sID	cName	Major	dec
3	Stanford	CS	Y
~~4~~	~~MIT~~	~~EE~~	~~N~~
~~1~~	~~William~~	~~CS~~	~~N~~
~~2~~	~~William~~	~~CS~~	~~Y~~

Projection Operation

- The projection operation works on a single relation R and defines a relation that contains a vertical subject of R, extracting the values of specified attributes and elimination duplicates

$$\pi_{fields}(Input)$$

Allows us to extract columns from a relation

Example:

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

$\pi_{age}(S2)$

age
35.0
55.5

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

$$\pi_{sname,rating}(\sigma_{rating>8}(S2))$$

sname	rating
yuppy	9
rusty	10

To pick both rows and columns…

ID and name of students with GPA>3.7

sID	sName	GPA	HS
1	Victor	8.6	700
3	William	7.3	1500
4	Jack	5.0	700

$$\Pi_{sID,sName}(\sigma_{GPA>3.7}\ Student)$$

$$\sigma_{cond}(Expr)$$

$$\Pi_{A_1,..,A_n}(Expr)$$

College

cName	state	enr
William	CA	500
Stanford	CA	400
MIT	MA	300
UoI	IL	400

Student

sID	sName	GPA	HS
1	Victor	8.6	700
2	Mike	2.2	200
3	William	7.3	1500
4	Jack	5.0	700

Apply

sID	cName	Major	dec
3	Stanford	CS	Y
4	MIT	EE	N
1	William	CS	N
2	William	CS	Y

Rename operation (ρ)

- The rename operator returns an existing relation under a new name. ρA(B) is the relation B with its name changed to A.

Binary Operation

- Four standard operations
- – Union
- – Intersection
- – Set-difference
- – Cross-product

Union

- The union of two relations R and S defines a relation that contains all the tuples of R or S or both R and S, duplicate tuples being eliminated.

Customer 1

Name	City
Anand	Coimbatore
Aravind	Chennai
Gopu	Tirunelveli
Helan	Palayankottai

Customer 2

Name	City
Gopu	Tirunelveli
Balu	Kumbakonam
Rahu	Chidambaram
Helan	Palayamkottai

Example

Query Determine Customer 1 ∪ Customer 2
Result of Customer 1 ∪ Customer 2

Customer 1 ∪ Customer 2

Name	City
Anand	Coimbatore
Aravind	Chennai
Balu	Kumbakonam
Gopu	Tirunelveli
Rahu	Chidambaram
Helan	Palayamkottai

Set Operation: Union

S1

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

S2

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0
44	guppy	5	35.0
28	yuppy	9	35.0

S1 U S2

Intersection Operation

- The intersection operation defines a relation consisting of the set of all tuples that are in both R and S.

Set Operation: Intersection

S1

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

S2

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

sid	sname	rating	age
31	lubber	8	55.5
58	rusty	10	35.0

S1 ∩ S2

Difference

- **R – S: returns a relation instance**
- containing all tuples that occur in R but not in S.
- R and S must be union-compatible.
- Scheme of the result is the schema of R.

- Union, intersection, and difference require the two input set to be union compatible
- – They have the same number of fields
- – corresponding fields, taken in order from left
- to right, have the same domains

Set Operation: Set-Difference

S1

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

S2

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

S1 – S2

sid	sname	rating	age
22	dustin	7	45.0

- Consider that you sell cars and a customer comes to you with the following request. “I need a number of cars. The models I want are Toyota Corolla, Honda CRV, and Maruti Swift. And, I need my cars to be white, grey, and red and I want them to have 1.4, 1.6, and 1.8 engine capacities. I want a combination of cars that satisfy the specifications I’ve stated.”

- M = {Toyota Corolla, Honda CRV, Maruti Swift}
- C = {White, Grey, Red}
- E = {1.4L, 1.6L, 1.8L}

Toyota Corolla
Honda CRV
Maruti Swift

White
Grey
Red

1.4
1.6
1.8

Toyota	**White**	**1.4**
Toyota	White	1.6
Toyota	White	1.8
Toyota	Grey	1.4
Toyota	Grey	1.6
Toyota	Grey	1.8
Toyota	Red	1.4
Toyota	Red	1.6
Toyota	Red	1.8
Honda CRV	White	1.4
Honda CRV	White	1.6
Honda CRV	White	1.8

Cross-Product

- R x S: Returns a relation instance whose
- scheme contains:
- – All the fields of R (in the same order as they appear in R)
- – All the fields os S (in the same order as they appear in S)
- The result contains one tuple <r,s> for each
- pair with r ∈ R and s ∈ S
- Basically, it is the Cartesian product.

Set Operation: Cross-Product

S1

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

R1

sid	bid	day
22	101	10/10/96
58	103	11/12/96

S1 x R1

(sid)	sname	rating	age	(sid)	bid	day
22	dustin	7	45.0	22	101	10/10/96
22	dustin	7	45.0	58	103	11/12/96
31	lubber	8	55.5	22	101	10/10/96
31	lubber	8	55.5	58	103	11/12/96
58	rusty	10	35.0	22	101	10/10/96
58	rusty	10	35.0	58	103	11/12/96

Cross-product: combine two relations (a.k.a. **Cartesian product**)

Names and GPAs of students with HS>1000 who applied to CS and were rejected

$$\Pi_{sName,\ GPA}\left(\sigma_{Student.sID=Apply.sID\ \wedge\ HS>1000\ \wedge\ major='CS'\ \wedge\ dec='N'}(Student \times Apply)\right)$$

College

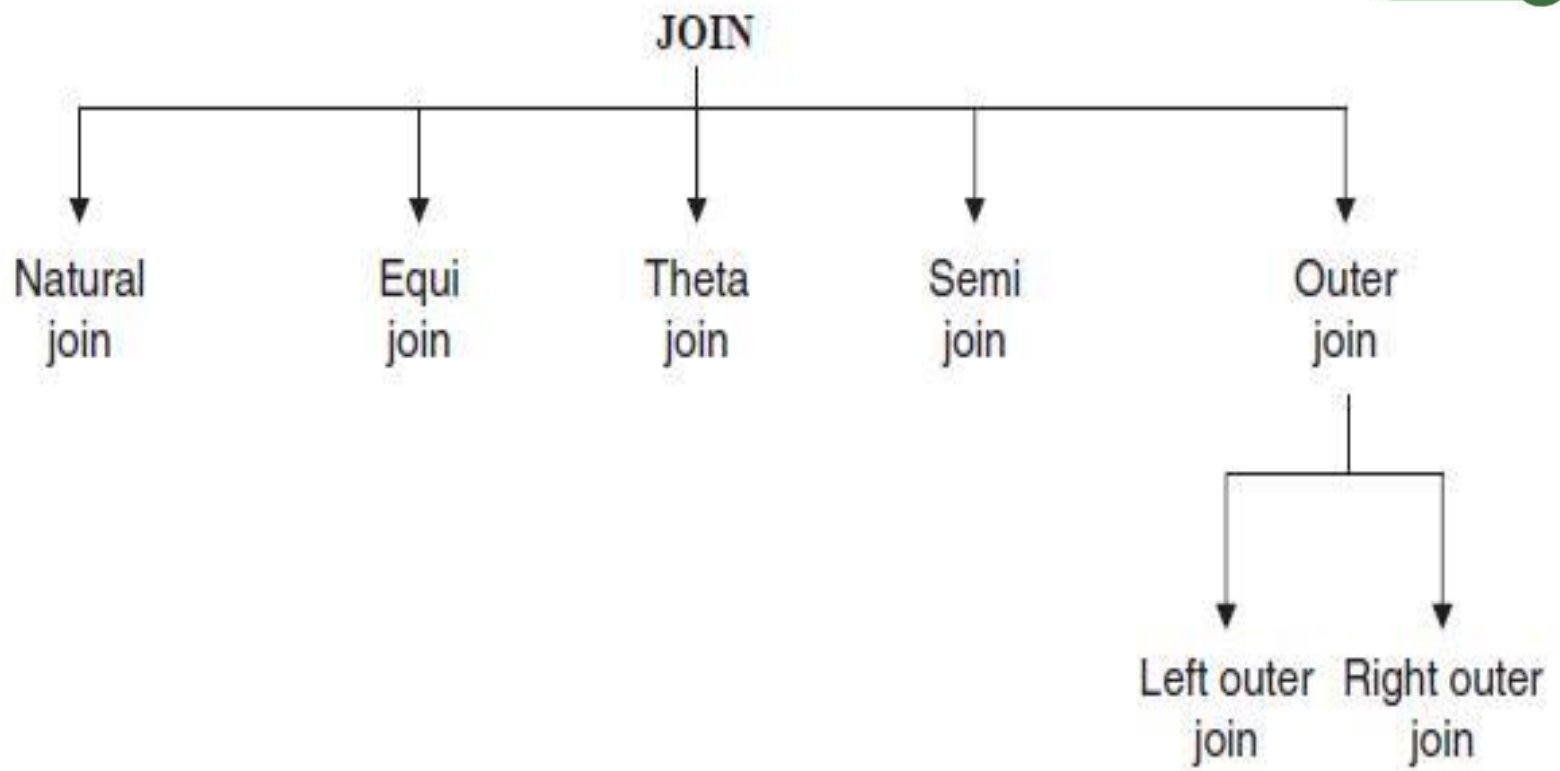
cName	state	enr
William	CA	500
Stanford	CA	400
MIT	MA	300
UoI	IL	400

Student

sID	sName	GPA	HS
1	Victor	8.6	700
2	Mike	2.2	200
3	William	7.3	1500
4	Jack	5.0	700

Apply

sID	cName	Major	dec
3	Stanford	CS	Y
4	MIT	EE	N
1	William	CS	N



Natural Join

$$R \bowtie (S \bowtie T) = (R \bowtie S) \bowtie T$$

Employee

Employee ID	Designation	Dept Number
C100	Lecturer	E1
C101	Assistant Professor	E2
C102	Professor	C1

$\bowtie$

Department

Dept name	Dept Number
Electrical	E1
Computer	C1

Employee $\bowtie$ Department

Employee ID	Designation	Dept Number	Dept name
C100	Lecturer	E1	Electrical
C102	Professor	C1	Computer

Outer Join

1. Full outer join
2. Left outer join
3. Right outer join

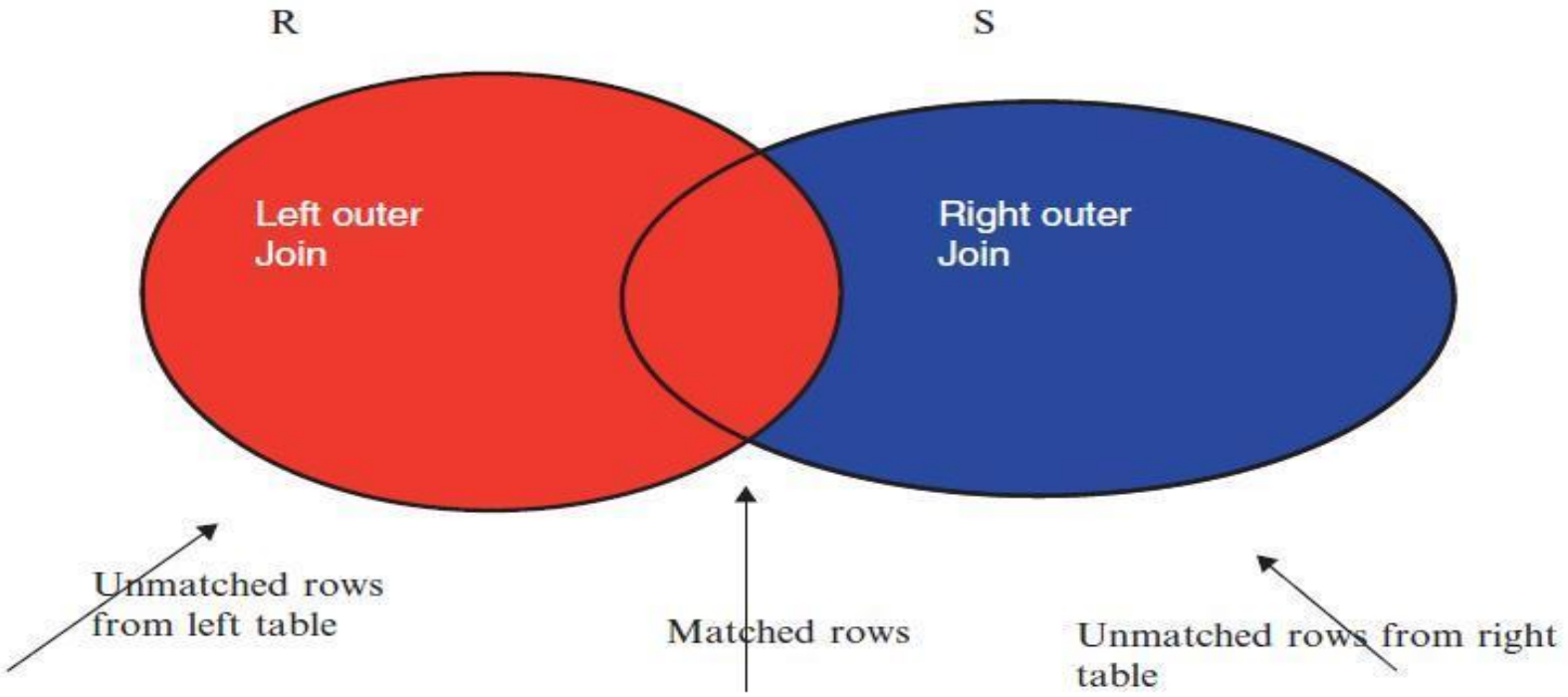


Fig. Representation of left and right outer join

- 1. ***Left Outer Join***. Left outer joins is a join in which tuples from R that do not have matching values in the common column of S are also included in the result relation.

- 2. ***Right Outer Join***. Right outer join is a join in which tuples from S that do not have matching values in the common column of R are also included in the result relation.

- 3. ***Full Outer Join.*** Full outer join is a join in which tuples from R that do not have matching values in the common columns of S still appear and tuples in S that do not have matching values in the common columns of R still appear in the resulting relation

Left Outer Join:

List all students and mention College if they apply in a college

$$Student \bowtie_{student.sID=Apply.sID} Apply$$

Student.sID	sName	GPA	HS	Apply.sID	cName	Major	Dec
1	Victor	8.6	700	1	William	CS	N
2	Mike	2.2	200	NULL	NULL	NULL	NULL
3	William	7.3	1500	3	Stanford	CS	Y
4	Jack	5.0	700	4	MIT	EE	N
4	Jack	5.0	700	4	Stanford	CS	Y

College

cName	state	enr
William	CA	500
Stanford	CA	400
MIT	MA	300
UoI	IL	400

Student

sID	sName	GPA	HS
1	Victor	8.6	700
2	Mike	2.2	200
3	William	7.3	1500
4	Jack	5.0	700

Apply

sID	cName	Major	dec
3	Stanford	CS	Y
4	MIT	EE	N
4	Stanford	CS	Y
1	William	CS	N

Right Outer Join:

List all College and student details who have applied there

$$Student \bowtie_{student.sID=Apply.sID} Apply$$

Student.sID	sName	GPA	HS	Apply.sID	cName	Major	Dec
3	William	7.3	1500	3	Stanford	CS	Y
4	Jack	5.0	700	4	MIT	EE	N
1	Victor	8.6	700	1	William	CS	N
NULL	NULL	NULL	NULL	5	UoI	MBA	Y
3	William	7.3	1500	3	MIT	CS	Y

College

cName	state	enr
William	CA	500
Stanford	CA	400
MIT	MA	300
UoI	IL	400

Student

sID	sName	GPA	HS
1	Victor	8.6	700
2	Mike	2.2	200
3	William	7.3	1500
4	Jack	5.0	700

Apply

sID	cName	Major	dec
3	Stanford	CS	Y
4	MIT	EE	N
1	William	CS	N
5	UoI	MBA	Y
3	MIT	CS	Y

Full Outer Join:

List all students and mention College if they apply in a college

$$Student \bowtie_{student.sID=Apply.sID} Apply$$

Student.sID	sName	GPA	HS	Apply.sID	cName	Major	Dec
1	Victor	8.6	700	1	William	CS	N
2	Mike	2.2	200	NULL	NULL	NULL	NULL
3	William	7.3	1500	3	Stanford	CS	Y
4	Jack	5.0	700	4	MIT	EE	N
NULL	NULL	NULL	NULL	5	UoI	MBA	Y

College

cName	state	enr
William	CA	500
Stanford	CA	400
MIT	MA	300
UoI	IL	400

Student

sID	sName	GPA	HS
1	Victor	8.6	700
2	Mike	2.2	200
3	William	7.3	1500
4	Jack	5.0	700

Apply

sID	cName	Major	dec
3	Stanford	CS	Y
4	MIT	EE	N
1	William	CS	N
5	UoI	MBA	Y

Name	Age	Food
Raja	21	Idly
Ravi	22	Dosa
Rani	20	Pizza
Devi	21	Pongal

Food	Day
Pongal	Monday
Idly	Tuesday
Dosa	Wednesday
Fried Rice	Thursday
Parotta	Friday

Table Left outer join of PEOPLE and MENU relation

PEOPLE ⟕ **PEOPLE. Food = MENU. Food MENU**

Name	Age	People.Food	Menu.Food	Day
Raja	21	Idly	Idly	Tuesday
Ravi	22	Dosa	Dosa	Wednesday
Rani	20	Pizza	NULL	NULL
Devi	21	Pongal	Pongal	Monday

Table Right outer join of PEOPLE and MENU relation

PEOPLE ⟖ **PEOPLE.Food = Menu.Food MENU**

Name	Age	People.Food	Menu.Food	Day
Devi	21	Pongal	Pongal	Monday
Raja	21	Idly	Idly	Tuesday
Ravi	22	Dosa	Dosa	Wednesday
NULL	NULL	NULL	Fried rice	Thursday
NULL	NULL	NULL	Parotta	Friday

PEOPLE

Name	Age	Food
Raja	21	Idly
Ravi	22	Dosa
Rani	20	Pizza
Devi	21	Pongal

MENU

Food	Day
Pongal	Monday
Idly	Tuesday
Dosa	Wednesday
Fried rice	Thursday
Parotta	Friday

Table Full outer join of PEOPLE and MENU relation

Name	Age	People.Food	Menu.Food	Day
Raja	21	Idly	Idly	Tuesday
Ravi	22	Dosa	Dosa	Wednesday
Rani	20	Pizza	NULL	NULL
Devi	21	Pongal	Pongal	Monday
NULL	NULL	NULL	Fried rice	Thursday
NULL	NULL	NULL	Parotta	Friday

From this table, it is clear that all tuples from the right-hand side relation (in our case the right hand relation is MENU) appears in the result.

3. The full outer join of PEOPLE and MENU on Food is represented in the relational algebra as PEOPLE $\bowtie_{\text{PEOPLE.Food} = \text{MENU.Food}}$ MENU. The result of the full outer join is shown in Table
From this table, it is clear that tuples from both the PEOPLE and the MENU relation appears in the result.

SEMI JOIN

- The semi-join of a relation R, defined over the set of attributes A, by relation S, defined over the set of attributes B, is the subset of the tuples of R that participate in the join of R with S.

EMPLOYEE

Employee Number	Employee Name	Designation
E1	Rajan	Programmer
E2	Krishnan	System Analyst
E3	Devi	Database Administrator
E4	Vidhya	Consultant

PAY

Designation	Salary
Programmer	25,000
Consultant	70,000

The semi-join of EMPLOYEE with the PAY is denoted by: EMPLOYEE $\ltimes_{\text{EMPLOYE.DESIGNATION=PAY.DESIGNATION}}$ PAY. The result of this semi-join is given later:

Employee Number	Employee Name	Designation
E1	Rajan	Programmer
E4	Vidhya	Consultant

From the result of the semi-join it is clear that a semi-join is half of a join: the rows of one table that match with at least one row of another table. Only the rows of the first table appear in the result.

Theta (θ) Join

- In theta join we apply the condition on input relation(s) and then only those selected rows are used in the cross product to be merged and included in the output.
- **Notation**

  $R1 \bowtie_{\theta} R2$

Theta Join

Names and GPAs of students with HS>1000 who applied to CS and were rejected

$$\pi_{sName,GPA}(\text{Student} \bowtie_{HS>1000 \wedge Major='CS' \wedge dec='N' \wedge Student.sID=Apply.sID} Apply)$$

Student.sID	sName	GPA	HS	Apply.sID	cName	Major	Dec
2	Mike	2.2	1200	2	William	CS	Y
2	Mike	2.2	1200	2	William	CS	Y
~~3~~	~~William~~	~~7.3~~	~~1500~~	~~3~~	~~Stanford~~	~~CS~~	~~Y~~
~~4~~	~~Jack~~	~~5.0~~	~~700~~	~~4~~	~~MIT~~	~~EE~~	~~N~~

College

cName	state	enr
William	CA	500
Stanford	CA	400
MIT	MA	300
UoI	IL	400

Student

sID	sName	GPA	HS
1	Victor	8.6	700
2	Mike	2.2	1200
3	William	7.3	1500
4	Jack	5.0	700

Apply

sID	cName	Major	dec
3	Stanford	CS	Y
4	MIT	EE	N
1	William	CS	N
2	William	CS	Y

Division

sno	pno
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

A

pno
p2

B1

sno
s1
s2
s3
s4

A/B1

pno
p2
p4

B2

sno
s1
s4

A/B2

pno
p1
p2
p4

B3

sno
s1

A/B3

- Find name of the customer having account in all branches

r1

Cust_ Name	Branch
Raja	Krishna Nagar
Ravi	Krishna Nagar
Rani	Radha Puram
Devi	Holi Gate
Raja	Radha Puram
Devi	Holi Gate
Raja	Holi Gate

r2

Branch
Krishna Nagar
Radha Puram
Holi Gate

r1÷ r2