Review :

RELATIONAL ALGEBRA

Relational database systems are expected to be equipped with a query language that can assist its users to query the database instances. There are two kinds of query languages – relational algebra and relational calculus.

Relational Algebra

Relational algebra is a procedural query language, which takes instances of relations as input and yields instances of relations as output. It uses operators to perform queries. An algebra whose operands are relations or variables that represent relations. Operators are designed to do the most common things that we need to do with relations in a database. The result is an algebra that can be used as a query language for relations. We need to know about relational algebra to understand query execution and optimization in a relational DBMS.

Relations are seen as *sets of tuples*, which means that no duplicates are allowed. SQL behaves differently in some cases. Remember the SQL keyword distinct. SQL is *declarative*, which means that you tell the DBMS *what* you want, but not *how* it is to be calculated. A C++ or Java program is *procedural*, which means that you have to state, step by step, exactly how the result should be calculated. Relational algebra is (more) procedural than SQL (Actually, relational algebra is mathematical expressions).

The fundamental operations of relational algebra are as follows :

1. Select
2. Project
3. Union
4. Set different
5. Cartesian product
6. Rename

Select Operation (σ)

It selects tuples that satisfy the given predicate from a relation (choose rows).

Notation: $\sigma_c(R)$

where
R is a table (relation)
Condition c is a boolean expression. It can use comparison operators (=, ≠, ≥, < , >, ≤) and boolean operators (**and, or,** and **not).** The result is a relation with the same schema as the operand but with only the tuples that satisfy the condition

For example :

student

name	gpa	Country
Bob	3.00	Canada
John	3.00	Britain
Tom	3.50	Canada
Maria	4.00	Mexico

Query :
list students with gpa > 3

Relational Algebra :
*s = **σ** gpa>3 (student)*

Result :
s

name	Gpa	Country
Tom	3.50	Canada
Maria	4.00	Mexico

Query :
list students with gpa > 3 and country= "Canada"

Relational Algebra :
*s = **σ** gpa>3 and country="Canada" (student)*

Result :
s

name	Gpa	Country
Tom	3.50	Canada

Project Operation (π)

It projects column*s* that satisfy a given predicate (choose columns).

Notation: πL(R)

where
R is a table (relation)
L is a list of attributes from the schema of R. The result is a relation with all the tuples from R but with only the attributes in L, and in that order.

For example :

student

name	gpa	Country
Bob	3.00	Canada
John	3.00	Britain
Tom	3.50	Canada
Maria	4.00	Mexico

Query :
list attribute name

Relational Algebra :

s = π*name (student)*

s

name
Bob
John
Tom
Maria

Query :
list attribute name, country

Relational Algebra :

s = π *name,country (student)*

s

name	Country
Bob	Canada
John	Britain
Tom	Canada
Maria	Mexico

Union Operation ($\cup$)

It performs binary union between two given relations and is defined as $r \cup s = \{ t \mid t \in r \text{ or } t \in s\}$

Notation: **r ∪ s**

Where **r** and **s** are either database relations or relation result set *temporaryrelation*.
For a union operation to be valid, the following conditions must hold :

- **r**, and **s** must have the same number of attributes.
- Attribute domains must be compatible.
- Duplicate tuples are automatically eliminated.

For example :

student

name	Gpa	Country
Bob	3.00	Canada
John	3.00	Britain
Tom	3.50	Canada
Maria	4.00	Mexico

professor

name	rank
Dr. Monk	Professor
Dr. Pooh	Associate Professor
Dr. Patel	Assistant Professor
Dr. Smith	Professor

Query :
list names of all people in the department

Can we do ? → T = student **∪** professor

Relational Algebra :

T = π name (student) **∪** π name **(**professor)

T

name
Bob
John
Tom
Maria
Dr. Monk
Dr. Pooh
Dr. Patel
Dr. Smith

Set Difference (−)

The result of set difference operation is tuples, which are present in one relation but are not in the second relation.

Notation: **r − s**

Where **r** and **s** are either database relations or relation result set *temporaryrelation*. Finds all the tuples that are present in **r** but not in **s**.

For example :

RegisteredFor

Name	Topic
Bob	Algorithms
John	Algorithms
Tom	Algorithms
Bob	Python
Tom	Python
Bob	Databases
John	Databases
Maria	Databases
John	GUI
Maria	GUI

Query :
Who is registered in the Database course but not in the Algorithms?

Relational Algebra :

T = ***σ*** *topic = "Databases" (RegisteredFor)* **-** ***σ*** *topic = "Algorithms" (RegisteredFor)*

Or

T = ***(σ*** *topic = "Databases" (RegisteredFor))* **–** ***(σ*** *topic = "Algorithms" (RegisteredFor))*

T

Name	Topic
Maria	Databases

Intersection (∩)

The result of intersection operation is tuples, which are present the set of tuples that are in both relation.

Notation: **r ∩ s**

The intersection of r and s, is the set of tuples that are in both r and s .

For example :

RegisteredFor

Name	Topic
Bob	Algorithms
John	Algorithms
Tom	Algorithms
Bob	Python
Tom	Python
Bob	Databases
John	Databases
Maria	Databases
John	GUI
Maria	GUI

Query :
Who is registered in the Database and the Algorithms course ?

Relational Algebra :

T = **π** name ***(σ*** *topic = "Databases" (RegisteredFor))* **∩ π** name ***(σ*** *topic = "Algorithms" (RegisteredFor))*

Or

T = (**π** name ***(σ*** *topic = "Databases" (RegisteredFor)))* **∩ (π** name ***(σ*** *topic = "Algorithms" (RegisteredFor)))*

T

Name
Bob
John

Cartesian Product (X)

The *cartesian product* of two tables combines each row in one table with each row in the other table.

Notation: **R X S**

The result is a relation with every combination of a tuple from R concatenated to a tuple from S.
Its schema is every attribute from R followed by every attribute of S.

For example :

Course

Topic	Year
Algorithms	2
Python	2
Databases	3
GUI	3

Professor

Name	Rank
Dr. Monk	Professor
Dr. Pooh	Associate Professor
Dr. Patel	Assistant Professor
Dr. Smith	Professor

T = Course X Professor

T

Topic	Year	Name	Rank
Algorithms	2	Dr. Monk	Professor
Algorithms	2	Dr. Pooh	Associate Professor
Algorithms	2	Dr. Patel	Assistant Professor
Algorithms	2	Dr. Smith	Professor
Python	2	Dr. Monk	Professor
Python	2	Dr. Pooh	Associate Professor
Python	2	Dr. Patel	Assistant Professor
Python	2	Dr. Smith	Professor
Databases	3	Dr. Monk	Professor
Databases	3	Dr. Pooh	Associate Professor
Databases	3	Dr. Patel	Assistant Professor
Databases	3	Dr. Smith	Professor
GUI	3	Dr. Monk	Professor
GUI	3	Dr. Pooh	Associate Professor
GUI	3	Dr. Patel	Assistant Professor
GUI	3	Dr. Smith	Professor

Combining Cross-Product with Select and Project Operation

Student

Name	Country	GPA
Bob	Canada	3.0
John	Britain	3.0
Tom	Canada	3.5
Maria	Mexico	4.0

Teaches

Name	Topic
Dr. Monk	Algorithms
Dr. Pooh	Python
Dr. Patel	Databases
Dr. Smith	GUI

RegisteredFor

Name	Topic
Bob	Algorithms
John	Algorithms
Tom	Algorithms
Bob	Python
Tom	Python
Bob	Databases
John	Databases
Maria	Databases
John	GUI
Maria	GUI

Query :
List of the students (name, country and GPA) taught by Dr. Monk?

Relational Algebra :

Teach_Register = π Teaches.Name, RegisteredFor (σ *Teaches.Name = "Dr. Monk" and Teaches.Topic = RegisteredFor.Topic (Teaches X RegisteredFor))*

Teach_Register

Teaches.Name	RegisteredFor.Name
Dr. Monk	Bob
Dr. Pooh	John
Dr. Patel	Tom

List_students = π Student.Name, Country, GPA (σ *Student.Name = Teach_Register.Name (Student X Teach_Register))*

List_students

Name	Country	GPA
Bob	Canada	3.0
John	Britain	3.0
Tom	Canada	3.5

Or

List_students = π Student.Name, Country, GPA (*σ Teaches.Name = "Dr. Monk" and Teaches.Topic = RegisteredFor.Topic and Student.Name = Teach_Register.Name (Student X Teaches X RegisteredFor))*

List_students

Name	Country	GPA
Bob	Canada	3.0
John	Britain	3.0
Tom	Canada	3.5

Rename Operation (ρ)

The results of relational algebra are also relations but without any name. The rename operation allows us to rename the output relation. 'rename' operation is denoted with small Greek letter **rho (ρ).**

Notation: **ρ** s(A1,A2,..,An)(R)

1. Resulting relation has exactly the same tuples as R, but the name of the relation is s.
2. The attributes of the result relation s can be renamed A1, A2, ..., An in order from the left.
3. If not all attributes are renamed, can specify rename attributes :
   ρ s, a→a1, b→b1 (R)

For example :

E

nr	name	dept
1	Bill	A
2	Sarah	B

1. rename both the table and the columns
   ρ R(enr, ename, dept)(E)
2. rename the columns nr into nomor, and the other fixed.
   ρ nr→nomor(E)

Exercise :

The following table data describe a relational model for air travel bookings.

Airport

airportId	name	city
LHR	Heathrow	London
LGW	Gatwick	London
CDG	Charles de Gaulle	Paris
ORY	Orly	Paris

Flight

flightNo	flightCompany	depAirport	arrAirport
AF1231	Air France	LHR	CDG
AF1232	Air France	CDG	LHR
AF1234	Air France	LGW	CDG
AF1235	Air France	CDG	LGW
BA2943	British Airways	LGW	ORY
BA2944	British Airways	ORY	LGW
BA4059	British Airways	LHR	CDG
BA4060	British Airways	CDG	LHR

Booking

ticketNo	name	nationality	flightNo	seatNo
EAG129489	John Jones	British	AF1232	12D
EAF123456	Fraser McEwan	British	AF1232	30E
ABS958332	Mathilde Duval	French	BA2944	10A
ORE394895	Fiona Stewart	British	BA4060	5D
EYR149583	Karen Woods	British	BA4059	14B
EAG348595	Pierre Fontaine	French	BA2944	30D

Seat

seatNo	flightNo	class
12D	AF1232	Business
30E	AF1232	Economy
10A	BA2944	Business
5D	BA4060	Business
14B	BA4059	Economy
30D	BA2944	Economy

Question 1: Operations in Relational Algebra

For each of the following queries in relational algebra, calculate the output table and give a brief statement of what query it answers.

(a) σ class='Business' (Seat)

(b) π nationality (Booking)

(c) σ nationality='French' (Booking) X σ class='Business' (Seat)

(d) π name(σ class='Business' (Booking X Seat))

(f) Airport $\cup$ Seat

Question 2: Constructing Queries

For each of the following questions, formulate the speci_ed queries in tuple-relational calculus and as a computation in relational algebra.

(a) Retrieve all information about airports in London. The schema of the output table should be same as that of the Airport table.

(b) Retrieve details of all bookings by British and French passengers. The schema of the output table should be same as that of the Booking table.

(c) Retrieve the names of all passengers.

(d) Retrieve the ight number, Departure and Arrival airports of all British Airways ights.

(e) Retrieve the name of every passenger together with their ight number and the associated ight company.

(f) Retrieve details of all ights from all airports in London. The output schema should be same as that of Flight table.

(g) Find out the ticket numbers and names of all passengers departing from London.

(h) Retrieve the ight number and company of all ights from London to Paris.