Relational Languages

CSci 150 – Fundamentals of Database Systems

Rodney M. Maniego Jr.
Instructor I / DCST / VSU
Version: 20231011

COURSE OUTCOME

Use relational languages in retrieving data in databases.

LEARNING OUTCOME

- Identify and explain the concepts and operations on Relational Languages.

TOPICS

1. Relational Languages
2. Operations on Relational Algebra
3. Operations on Relational Calculus

Relational Languages

Relational Languages

The formalism on querying and manipulating relations emerged in the subsequent development on the relational model following the initial work of E.F. Codd.

- Relational Algebra
- Relational Calculus
- Standard Query Language

1.5. SOME LINGUISTIC ASPECTS

The adoption of a relational model of data [...] permits the development of a universal data sub-language based on an applied predicate calculus. [...] Such a language would provide a yard-stick of linguistic power for all other proposed data languages.

2.1. OPERATIONS ON RELATIONS

Since relations are sets, all of the usual set operations are applicable to them. [...]

Codd, E. F. (1970). A relational model of data for large shared data banks. Communications of the ACM, 13(6), 377–387.

https://dl.acm.org/doi/10.1145/362384.362685

Relational Algebra

What is Algebra?

Algebraic Operations involve manipulating mathematical expressions and equations using symbols, variables, and operators.

Branches of Algebra:

- Elementary
- Linear
- Boolean
- Relational

Relational Algebra

It refers to **a set of operations** and rules **on relations** that allow various operations and transformations.

It is **a procedural query language** where the series of operations and transformations must be defined to query or manipulate relations.

Simply, it takes one or more relations as an input and produces a relation as an output.

Comparison Operators

- $=$ equal to
- $\neq$ not equal to
- $<$ less than
- $\leq$ less than or equal
- $>$ greater than
- $\geq$ greater than or equal to.

Logical Connectives

- $\wedge$ and (Conjunction)
- $\vee$ or (Disjunction)
- $\neg$ not (Negation)

Fundamental RA Operations

- σ Select
- Π Project
- ∪ Union
- ∩ Intersection
- - Set Difference
- × Cartesian Product
- ρ Rename

#1 **Select** (σ)

Select the tuples from a relation that satisfy the conditions.

Notation: $\sigma_p(R)$

where:

- σ is the selection predicate
- p is a propositional logic formula
- R is a relation

#1 **Select** (σ) - Example

AID	FirstName	LastName	Username	BirthDate	Role
1	Juan	Dela Cruz	OneDy	2000-09-11	0
2	Maria	Sta. Ana	lilLamb	1999-06-29	0
3	Pedro	Malaya	peter20	1998-01-03	1
4	Elias	Pinaglabanan	elyuzzz	2002-08-17	0
5	Mateo	Abante	10math01	2000-03-22	1

$\sigma_{\text{Role} = 1}(\text{Account})$

AID	FirstName	LastName	Username	BirthDate	Role
3	Pedro	Malaya	peter20	1998-01-03	1
5	Mateo	Abante	10math01	2000-03-22	1

#1 **Select** (σ) - Example

AID	FirstName	LastName	Username	BirthDate	Role
1	Juan	Dela Cruz	OneDy	2000-09-11	0
2	Maria	Sta. Ana	lilLamb	1999-06-29	0
3	Pedro	Malaya	peter20	1998-01-03	1
4	Elias	Pinaglabanan	elyuzzz	2002-08-17	0
5	Mateo	Abante	10math01	2000-03-22	1

σ ? (Account)

AID	FirstName	LastName	Username	BirthDate	Role
2	Maria	Sta. Ana	lilLamb	1999-06-29	0
3	Pedro	Malaya	peter20	1998-01-03	1

#1 **Select** (σ) - Example

AID	FirstName	LastName	Username	BirthDate	Role
1	Juan	Dela Cruz	OneDy	2000-09-11	0
2	Maria	Sta. Ana	lilLamb	1999-06-29	0
3	Pedro	Malaya	peter20	1998-01-03	1
4	Elias	Pinaglabanan	elyuzzz	2002-08-17	0
5	Mateo	Abante	10math01	2000-03-22	1

σ BirthDate < 2000-01-01 (Account)

AID	FirstName	LastName	Username	BirthDate	Role
2	Maria	Sta. Ana	lilLamb	1999-06-29	0
3	Pedro	Malaya	peter20	1998-01-03	1

#2 **Project** (Π)

Projects the columns from a given relation.

Notation: $\prod_{A_1,A_2,\ldots,A_n}(R)$

where:

- ∏ is the projection predicate
- A_n are attribute names of the relation R
- R is a relation

NOTE: Duplicate rows are eliminated.

#2 **Project** (Π) - Example

AID	FirstName	LastName	Username	BirthDate	Role
1	Juan	Dela Cruz	OneDy	2000-09-11	0
2	Maria	Sta. Ana	lilLamb	1999-06-29	0
3	Pedro	Malaya	peter20	1998-01-03	1
4	Elias	Pinaglabanan	elyuzzz	2002-08-17	0
5	Mateo	Abante	10math01	2000-03-22	1

Π FirstName, LastName (σ Username = “peter20” (Account))

FirstName	LastName
Pedro	Malaya

#2 **Project** (Π) - Example

AID	FirstName	LastName	Username	BirthDate	Role
1	Juan	Dela Cruz	OneDy	2000-09-11	0
2	Maria	Sta. Ana	lilLamb	1999-06-29	0
3	Pedro	Malaya	peter20	1998-01-03	1
4	Elias	Pinaglabanan	elyuzzz	2002-08-17	0
5	Mateo	Abante	10math01	2000-03-22	1

$\Pi_{?}(\sigma_{?}(\text{Account}))$

Username
OneDy
lilLamb
elyuzzz

#2 **Project** (Π) - Example

AID	FirstName	LastName	Username	BirthDate	Role
1	Juan	Dela Cruz	OneDy	2000-09-11	0
2	Maria	Sta. Ana	lilLamb	1999-06-29	0
3	Pedro	Malaya	peter20	1998-01-03	1
4	Elias	Pinaglabanan	elyuzzz	2002-08-17	0
5	Mateo	Abante	10math01	2000-03-22	1

$$\Pi_{\text{Username}}(\sigma_{\text{Role} = 0}(\text{Account}))$$

Username
OneDy
lilLamb
elyuzzz

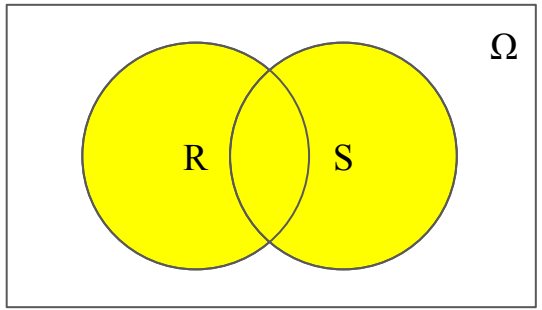
#3 Union (∪)

Performs a binary union between two relations.

Notation: R ∪ S

where:

- ∪ is the union operator
- R is the first relation
- S is the second relation



NOTE: Relations can be from the database or temporary relations from previous operation. For a union operation to be valid, R and S must have the same number of attributes, the attribute domains must be compatible, and the duplicate tuples will be eliminated.

#3 Union (∪) - Example

R ∪ S

SupplierID	Company	Email
1	Hello Paper, Inc.	hello@ppr.com
2	Loom GmbH	customer@loom.de
3	SoBasic LLC	so@basic.com
4	XieXie Corp.	thank@you.cn
5	Happy Dragon Inc.	hello@dragon.cn

ShipperID	Company	Email
1	Ride Fast LLC	fast@rider.com
2	Green Giant Inc.	customer@giant
3	Happy Dragon Inc.	hello@dragon.cn
4	Lite Shipper GmbH	lite@shipper.com
5	Zooom Corp.	hello@zooom.com

#3 Union (∪) - Example

$$(\Pi_{\text{Company, Email}}(\text{Supplier})) \cup (\Pi_{\text{Company, Email}}(\text{Shipper}))$$

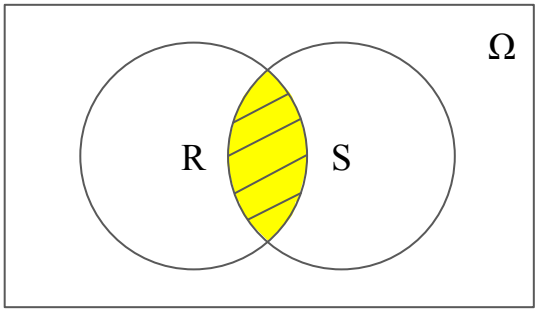
SupplierID	Company	Email
1	Hello Paper, Inc.	hello@ppr.com
2	Loom GmbH	customer@loom.de
3	SoBasic LLC	so@basic.com
4	XieXie Corp.	thank@you.cn
5	Happy Dragon Inc.	hello@dragon.cn

ShipperID	Company	Email
1	Ride Fast LLC	fast@rider.com
2	Green Giant Inc.	customer@giant
3	Happy Dragon Inc.	hello@dragon.cn
4	Lite Shipper GmbH	lite@shipper.com
5	Zooom Corp.	hello@zooom.com

Company	Email
Hello Paper, Inc.	hello@ppr.com
Loom GmbH	customer@loom.de
SoBasic LLC	so@basic.com
XieXie Corp.	thank@you.cn
Happy Dragon Inc.	hello@dragon.cn
Ride Fast LLC	fast@rider.com
Green Giant Inc.	customer@giant
Lite Shipper GmbH	lite@shipper.com
Zooom Corp.	hello@zooom.com

#4 Intersection (∩)

Defines a temporary relation consisting a set of tuples found in both R and S relations.



Notation: R∩S

where:

- ∩ is the intersection operator
- R is the first relation
- S is the second relation

NOTE: Relations can be from the database or temporary relations from previous operation. Both R and S must be union-compatible.

#4 **Intersection** (∩) - Example

R∩S

SupplierID	Company	Email
1	Hello Paper, Inc.	hello@ppr.com
2	Loom GmbH	customer@loom.de
3	SoBasic LLC	so@basic.com
4	XieXie Corp.	thank@you.cn
5	Happy Dragon Inc.	hello@dragon.cn

ShipperID	Company	Email
1	Ride Fast LLC	fast@rider.com
2	Green Giant Inc.	customer@giant
3	Happy Dragon Inc.	hello@dragon.cn
4	Lite Shipper GmbH	lite@shipper.com
5	Zooom Corp.	hello@zooom.com

#4 **Intersection** (∩) - Example

$$(\Pi_{\text{Company, Email}}(\text{Supplier})) \cap (\Pi_{\text{Company, Email}}(\text{Shipper}))$$

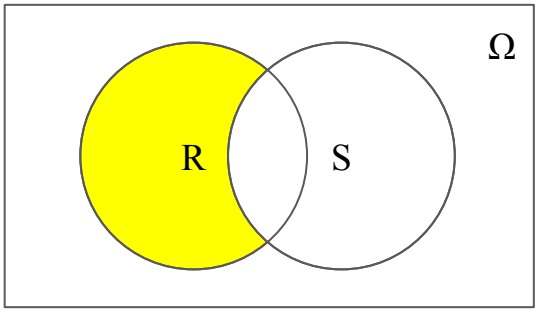
SupplierID	Company	Email
1	Hello Paper, Inc.	hello@ppr.com
2	Loom GmbH	customer@loom.de
3	SoBasic LLC	so@basic.com
4	XieXie Corp.	thank@you.cn
5	Happy Dragon Inc.	hello@dragon.cn

ShipperID	Company	Email
1	Ride Fast LLC	fast@rider.com
2	Green Giant Inc.	customer@giant
3	Happy Dragon Inc.	hello@dragon.cn
4	Lite Shipper GmbH	lite@shipper.com
5	Zooom Corp.	hello@zooom.com

Company	Email
Happy Dragon Inc.	hello@dragon.cn

#5 Set Difference (–)

Defines a temporary relation consisting a set of tuples found in the relation R, but not in S.



Notation: R—S

where:

- – is the set difference operator
- R is the first relation
- S is the second relation

NOTE: Relations can be from the database or temporary relations from previous operation. Both R and S must be compatible or union-compatible.

#5 **Set Difference** (–) - Example

R–S

SupplierID	Company	Email
1	Hello Paper, Inc.	hello@ppr.com
2	Loom GmbH	customer@loom.de
3	SoBasic LLC	so@basic.com
4	XieXie Corp.	thank@you.cn
5	Happy Dragon Inc.	hello@dragon.cn

ShipperID	Company	Email
1	Ride Fast LLC	fast@rider.com
2	Green Giant Inc.	customer@giant
3	Happy Dragon Inc.	hello@dragon.cn
4	Lite Shipper GmbH	lite@shipper.com
5	Zooom Corp.	hello@zooom.com

#5 Set Difference (–) - Example

$$(\Pi_{\text{Company, Email}}(\text{Supplier})) - (\Pi_{\text{Company, Email}}(\text{Shipper}))$$

SupplierID	Company	Email
1	Hello Paper, Inc.	hello@ppr.com
2	Loom GmbH	customer@loom.de
3	SoBasic LLC	so@basic.com
4	XieXie Corp.	thank@you.cn
5	Happy Dragon Inc.	hello@dragon.cn

ShipperID	Company	Email
1	Ride Fast LLC	fast@rider.com
2	Green Giant Inc.	customer@giant
3	Happy Dragon Inc.	hello@dragon.cn
4	Lite Shipper GmbH	lite@shipper.com
5	Zooom Corp.	hello@zooom.com

Company	Email
Hello Paper, Inc.	hello@ppr.com
Loom GmbH	customer@loom.de
SoBasic LLC	so@basic.com
XieXie Corp.	thank@you.cn

#6 Cartesian Product (×)

Defines a temporary relation combination of information from the relations R and S.

Notation: $R \times S$

where:

- × is the cartesian product operator
- R is the first relation
- S is the second relation

#6 **Cartesian Product** (×) - Example

R×S

OrderID	AccountID	GrossTotal
1	119	12999.99
2	21	23789.54
3	506	89281.01
4	98	599.99

DeliveryID	Name	Rate
1	Standard	0.0
2	Express	1.2
3	Next-day	2.0

#6 **Cartesian Product** (×) - Example

$\sigma_{\text{OrderID} = 3}(\text{Order}) \times \text{Shipper}$

OrderID	AccountID	GrossTotal
1	119	12999.99
2	21	23789.54
3	506	89281.01
4	98	599.99

DeliveryID	Name	Rate
1	Standard	0.0
2	Express	1.2
3	Next-day	2.0

OrderID	AccountID	GrossTotal	DeliveryID	Name	Rate
3	506	89281.01	1	Standard	0.0
3	506	89281.01	2	Express	1.2
3	506	89281.01	3	Next-day	2.0

#7 Rename (ρ)

Defines a temporary relation consisting a set of tuples found in the relation R, but not in S.

Notation: $\rho_{X(B_1,B_2,...,B_n)}(R)$

where:

- ρ is the rename predicate
- X is the new name of the relation R
- B_n are new attribute names of the relation X/R
- R is a relation

NOTE: Relations can be from the database or temporary relations from previous operation.

#7 **Rename (ρ)** - Example

$$\rho_{X(B_1,B_2,\ldots,B_n)}(R)$$

TransactionData

TID	AccountID	NetTotal
1	119	12999.99
2	21	23789.54
3	506	89281.01
4	98	599.99

Transaction

TID	AccountID	NetTotal
1	119	12999.99
2	21	23789.54
3	506	89281.01
4	98	599.99

#7 **Rename (ρ)** - Example

$$\rho_{\text{Transaction}}(\text{TransactionData})$$

Transaction

TID	AccountID	NetTotal
1	119	12999.99
2	21	23789.54
3	506	89281.01
4	98	599.99

#7 **Rename (ρ)** - Example

$$\rho_{(B_1,B_2,\ldots,B_n)}(\text{Transaction})$$

Transaction

TID	AccountID	NetTotal
1	119	12999.99
2	21	23789.54
3	506	89281.01
4	98	599.99

Transaction

TID	AccountID	GrandTotal
1	119	12999.99
2	21	23789.54
3	506	89281.01
4	98	599.99

#7 **Rename (ρ)** - Example

$$\rho_{(\text{NetTotal as GrandTotal})}(\text{Transaction})$$

Transaction

TID	AccountID	GrandTotal
1	119	12999.99
2	21	23789.54
3	506	89281.01
4	98	599.99

#7 **Rename (ρ)** - Example

$$\rho\,?\,(\text{Transaction})$$

Transaction

TID	AccountID	GrandTotal
1	119	12999.99
2	21	23789.54
3	506	89281.01
4	98	599.99

Sales

SID	AccountID	GrandTotal
1	119	12999.99
2	21	23789.54
3	506	89281.01
4	98	599.99

#7 **Rename (ρ)** - Example

$$\rho_{\text{Transaction (TID as SID)}}(\text{Transaction})$$

Transaction

TID	AccountID	GrandTotal
1	119	12999.99
2	21	23789.54
3	506	89281.01
4	98	599.99

Sales

SID	AccountID	GrandTotal
1	119	12999.99
2	21	23789.54
3	506	89281.01
4	98	599.99

Additional RA Operations

- Assignment (←)
- Division (÷)
- Joins (Inner and Outer Join)

#1 Assignment (←)

Select the tuples from a relation that satisfy the conditions.

Notation: R ⟵ (E)

where:

- ← is the assignment operator
- R is the new relation
- E is an expression

#2 Division (÷)

Select the tuples from a relation that satisfy the conditions.

Notation: R ÷ S

where:

- ÷ is the assignment operator
- R is the first relation
- S is the second relation

#3 Joins

It refers to a binary operation that combines two tables and selecting pairs that meet a specified condition.

Types:

- Inner Join
- Outer Join

#3.1 Inner Join

It is a type of JOIN operation that returns only the rows from both tables that have matching values in the specified join condition.

Types:

- Theta Join
- EQUI Join
- Natural Join

#3.1 **Inner Join:** Theta Join ($\bowtie_\theta$)

It is a generalization of the inner join, where you can use any relational operators to specify the join condition.

Notation: R $\bowtie_{\theta}$ S

where:

- $\bowtie$ is the JOIN operator
- θ is the condition using any relational operator ($=, \neq, <, \leq, >, \geq$)
- R is the first relation
- S is the second relation

#3.1 **Inner Join:** Theta Join ($\bowtie_\theta$)

Product $\bowtie_{\text{Product.PID} = \text{Rating.PID} \wedge \text{Product.Stars} \geq 3}$ Category

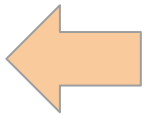
Product

PID	CID	ProductName
1	2	Apple Tea
2	2	Brewed Coffee
3	2	Coca Loca
4	1	Potato Crisps

Rating

RID	AID	PID	Stars	Feedback	TimeStamp
1	119	3	**3**	Fast delivery	2020-10-02 22:08:25
2	21	1	**5**	Must have!	2021-09-27 06:32:13
3	289	3	**4**	Great product	2022-01-15 04:21:41
4	506	2	1	Do not buy	2023-03-09 15:45:58

PID	CID	ProductName	RID	AID	Stars	Feedback	TimeStamp
1	2	Apple Tea	2	21	5	Must have!	2021-09-27 06:32:13
3	2	Coca Loca	1	119	3	Fast delivery	2020-10-02 22:08:25
3	2	Coca Loca	3	289	4	Great product	2022-01-15 04:21:41



#3.1 **Inner Join:** EQUI Join

It is a specific type of inner join that uses only equality comparisons to match rows from two tables based on specified columns.

Notation: $R \bowtie_C S$

where:

- $\bowtie$ is the JOIN operator
- C is the join condition
- R is the first relation
- S is the second relation

#3.1 **Inner Join:** EQUI Join

Product $\bowtie_{\text{Product.CID} = \text{Category.CID}}$ Category

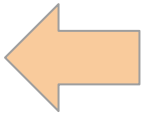
Product

PD	CID	ProductName
1	2	Apple Tea
2	2	Brewed Coffee
3	2	Coca Loca
4	1	Potato Crisps

Category

CID	CategoryName	Description
1	Chips	Crispy, flavored snacks.
2	Beverage	Drinks for human consumption.
3	Pastries	Baked goods, sweet or savory.

PID	CID	ProductName	CategoryName	Description
1	2	Apple Tea	Beverage	Drinks for human consumption.
2	2	Brewed Coffee	Beverage	Drinks for human consumption.
3	2	Coca Loca	Beverage	Drinks for human consumption.
4	1	Potato Crisps	Chips	Crispy, flavored snacks.



#3.1 **Inner Join:** Natural Join ($\bowtie_*$)

It is a special type of inner join that automatically matches rows based on columns with the same name in both tables.

Notation: $R \bowtie S$

where:

- $\bowtie$ is the JOIN operator
- R is the first relation
- S is the second relation

$$R \bowtie S = \prod_{A, B, C} (\sigma_{R.B=S.B} (R \times S))$$

Assuming that both relation R and S shares common B attributes.

#3.1 **Inner Join:** Natural Join ($\bowtie_*$)

Product ⋈ Category

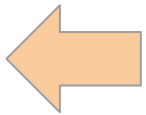
Product

PID	CID	ProductName
1	2	Apple Tea
2	2	Brewed Coffee
3	2	Coca Loca
4	1	Potato Crisps

Supplier

SID	SupplierName	Location
1	Hello Paper, Inc.	123 Stationery St., Paper Town, Inkwell County, USA, 12345
2	Loom GmbH	456 Textile Ave, Fabric City, Threadshire, Germany, 78901
3	SoBasic LLC	789 Simplicity Lane, Miniville, Simple County, USA, 56789
4	XieXie Corp.	1010 Thank You Bvd, Gratitude City, Appreciation County, China, 88888

PID	CID	ProductName	RID	AID	Stars	Feedback	TimeStamp
1	2	Apple Tea	2	21	5	Must have!	2021-09-27 06:32:13
3	2	Coca Loca	1	119	3	Fast delivery	2020-10-02 22:08:25
3	2	Coca Loca	3	289	4	Great product	2022-01-15 04:21:41



#3.2 Outer Join

It is a type of JOIN operation that returns all the rows from one table and the matching rows from the other table, leaving the unmatched cells with NULL values.

Types:

- Left Outer Join
- Right Outer Join
- Full Outer Join

#3.2 **Outer Join:** Left Outer Join (⟕)

It is a type of JOIN operation that returns all the rows from left table and the matching rows from the right table, leaving the unmatched columns with with NULL values.

Notation: R ⟕ S

where:

- ⟕ is the JOIN operator
- R is the first relation
- S is the second relation

#3.2 **Outer Join:** Right Outer Join (⟖)

It is a type of JOIN operation that returns all the rows from right table and the matching rows from the left table, leaving the unmatched columns with NULL values.

Notation: R ⟖ S

where:

- ⟖ is the JOIN operator
- R is the first relation
- S is the second relation

#3.2 **Outer Join:** Full Outer Join (⟗)

It is the union of the left and right outer join operations.

Notation: R ⟗ S

where:

- ⟗ is the JOIN operator
- R is the first relation
- S is the second relation

Relational Calculus

What is Calculus?

It is a branch of mathematics that deals with the study of rates of change and accumulation of quantities.

Relational Calculus

It refers to **a declarative query language** that expresses what data is needed without explicitly telling how to obtain it.

Types of Relational Calculus:

- Tuple Relational Calculus
- Domain Relational Calculus

NOTE: The “calculus” is Relational Calculus is derived from the similarity of its formalism, symbolic manipulation, and systematic approach to Calculus; specifically, it is based on Predicate Calculus, where a predicate is a truth-valued function with arguments.

Huh?

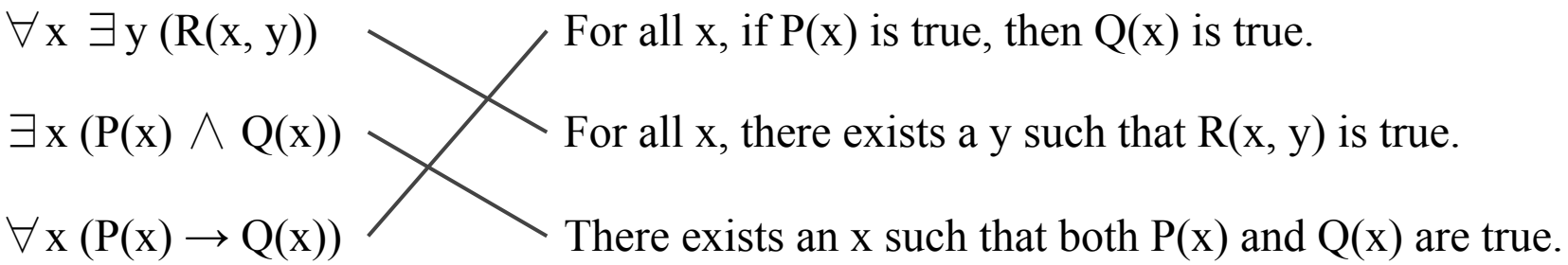
Relational Calculus

An approach to specify what data you want,
but not how to do it.

Tuple Relational Calculus

- Attributes and constants
- Comparison operators ($=$, $\neq$, $<$, $\leq$, $>$, $\geq$)
- Connectives ($\wedge$, $\vee$, $\neg$)
- Quantifiers ($\exists$, $\forall$)

Tuple Relational Calculus



Tuple Relational Calculus

It expresses the desired results as a set of tuples, where each tuple represents a row in a relation.

Expression: { t | P(t) }

where:

- {t} is the resulting set of tuples
- P(t) is the condition for selecting the set of tuples

The set of all tuple t, such that the condition for t is true.

Tuple Relational Calculus

Select all tuples from R where amount is greater than 100.

- Set-Builder Notation: $\{ t \mid t \in R \land t[amount] > 100 \}$
- Path Expression Notation: $\{ t \mid R(t) \land t.amount > 100 \}$

Tuple Relational Calculus

Q: Retrieve the tuples from the “Transaction” relation where the grand total of the transaction is greater than P1,000:

{ }

Transaction

TID	AccountID	GrandTotal
1	119	12999.99
2	21	23789.54
3	506	89281.01
4	98	599.99

Tuple Relational Calculus

Q: Retrieve the tuples from the “Transactions” relation where the grand total of the transaction is greater than P1,000:

$$\{ t \mid t \in \text{Transaction} \wedge t[\text{GrandTotal}] > 1000 \}$$

Transaction

TID	AccountID	GrandTotal
1	119	12999.99
2	21	23789.54
3	506	89281.01
4	98	599.99

Tuple Relational Calculus

Q: Retrieve the tuples from the “Transactions” relation where the grand total of the transaction is greater than P1,000:

$$\{ t \mid Transaction\ (t) \wedge t.GrandTotal > 1000) \}$$

Transaction

TID	AccountID	GrandTotal
1	119	12999.99
2	21	23789.54
3	506	89281.01
4	98	599.99

Tuple Relational Calculus

Q: Retrieve all tuples from the Account relation where the account is still active and with no associated loans:

{ }

Account

AID	FirstName	LastName	Username	Active	Role
1	Juan	Dela Cruz	OneDy	1	0
2	Maria	Sta. Ana	lilLamb	1	0
3	Pedro	Malaya	peter20	0	1
4	Elias	Pinaglabanan	elyuzzz	0	0
5	Mateo	Abante	10math01	1	1

Loan

LID	AID	Amount	Timestamp
1	4	24589.01	2020-10-27 22:08:25
2	5	50281.22	2021-09-27 06:32:13
3	3	10789.54	2022-01-09 15:45:58
4	1	129281.22	2022-12-23 00:19:30

Tuple Relational Calculus

Q: Retrieve all tuples from the Account relation where the account is still active and with no associated loans:

$$\{ t \mid \exists s \in \text{Account}\,(s[\text{Active}] = 1 \wedge \exists u \in \text{Loan}(s[\text{AID}] \neq u[\text{AID}])) \}$$

Account

AID	FirstName	LastName	Username	Active	Role
1	Juan	Dela Cruz	OneDy	1	0
2	Maria	Sta. Ana	lilLamb	1	0
3	Pedro	Malaya	peter20	0	1
4	Elias	Pinaglabanan	elyuzzz	0	0
5	Mateo	Abante	10math01	1	1

Loan

LID	AID	Amount	Timestamp
1	4	24589.01	2020-10-27 22:08:25
2	5	50281.22	2021-09-27 06:32:13
3	3	10789.54	2022-01-09 15:45:58
4	1	129281.22	2022-12-23 00:19:30

Tuple Relational Calculus

Q: Retrieve all tuples from the Product relation where there exists a discount matching the product falling within December 2023:

Product

PID	CID	Name	DateCreated
1	9	Power Coffee 30g	2019-10-27 22:08:25
2	15	Bubble Shorts Men S	2020-09-27 06:32:13
3	4	Biskeet Crackers 10s	2021-01-09 15:45:58
4	23	Washing Machine Pro	2021-12-23 00:19:30
5	2	Sangsek Shampoo 12s	2022-03-22 04:21:34

Discount

DID	PID	Percent	Start	End
1	3	5	2023-06-15 00:00:00	2020-06-30 23:59:59
2	3	20	2023-08-08 00:00:00	2021-08-08 23:59:59
3	1	5	2023-11-11 00:00:00	2022-11-11 23:59:59
4	4	10	2023-12-01 00:00:00	2022-12-31 23:59:59

Tuple Relational Calculus

Q: Retrieve all tuples from the Product relation where there exists a discount matching the product falling within December 2023:

$$\{ t \mid \exists s \in Product\ (\exists u \in Discount(s[PID] = u[PID] \wedge u[Start] \geq '2023\text{-}12\text{-}01\ 00{:}00{:}00' \wedge u[End] \leq '2023\text{-}12\text{-}31\ 00{:}00{:}00')) \}$$

Product

PID	CID	Name	DateCreated
1	9	Power Coffee 30g	2019-10-27 22:08:25
2	15	Bubble Shorts Men S	2020-09-27 06:32:13
3	4	Biskeet Crackers 10s	2021-01-09 15:45:58
4	23	Washing Machine Pro	2021-12-23 00:19:30
5	2	Sangsek Shampoo 12s	2022-03-22 04:21:34

Discount

DID	PID	Percent	Start	End
1	3	5	2023-06-15 00:00:00	2020-06-30 23:59:59
2	3	20	2023-08-08 00:00:00	2021-08-08 23:59:59
3	1	5	2023-11-11 00:00:00	2022-11-11 23:59:59
4	4	10	2023-12-01 00:00:00	2022-12-31 23:59:59

Domain Relational Calculus

It expresses the desired results as tuples
by focusing on attributes rather than specific rows.

Expression: $\{ \langle x_1, x_2, .., x_n \rangle \mid P(x_1, x_2, .., x_n) \}$

where:

- $\{\langle x_1, x_2, .., x_n \rangle\}$ is the set of resulting tuples
- $P(x_1, x_2, .., x_n)$ is the condition for selecting the set of tuples

The set of values for $\langle x_1, x_2, .., x_n \rangle$, such that the condition for $\langle x_1, x_2, .., x_n \rangle$ is true.

Huh?

Domain Relational Calculus

- In TRC, everything is about tuples.
- In DRC, you have more control using attributes.

Domain Relational Calculus

Q: Retrieve the account IDs from the “Logs” relation where the date falls within August 22, 2023:

$$\{ AID \mid \exists \langle AID, Timestamp \rangle \in Logs\ (Timestamp \geq '2023\text{-}08\text{-}22\ 00{:}00{:}00' \wedge Timestamp \leq '2023\text{-}08\text{-}22\ 00{:}00{:}00') \}$$

Logs

LID	AID	Action	Timestamp
1	238	View ...	2023-06-27 01:19:31
2	32	Upload ...	2023-08-19 23:52:44
3	617	Update ...	2023-08-22 09:23:12
4	98	Delete ...	2023-08-22 17:45:22
5	109	Download ...	2023-09-01 17:37:01

Domain Relational Calculus

Q: Retrieve the log ID and account IDs from the “Logs” relation where the date falls within August 22, 2023:

$$\{ \langle LID, AID \rangle \mid \exists \langle LID, AID, Timestamp \rangle \in Logs\ (Timestamp \geq '2023\text{-}08\text{-}22\ 00{:}00{:}00' \wedge Timestamp \leq '2023\text{-}08\text{-}22\ 00{:}00{:}00') \}$$

Logs

LID	AID	Action	Timestamp
1	238	View ...	2023-06-27 01:19:31
2	32	Upload ...	2023-08-19 23:52:44
3	617	Update ...	2023-08-22 09:23:12
4	98	Delete ...	2023-08-22 17:45:22
5	109	Download ...	2023-09-01 17:37:01

Domain Relational Calculus

Q: Retrieve the book title from the “Books” relation written by “Charles Dickens” and page count is greater or equal than 300:

$$\{ \text{Title} \mid \exists \langle \text{B.AID}, \text{Title}, \text{PageCount} \rangle \in \text{Books} (\text{PageCount} \geq 300 \wedge \exists \langle \text{A.AID}, \text{Name} \rangle \in \text{Author} (\text{Name} = \text{'Charles Dickens'} \wedge \text{A.AID} = \text{B.AID})) \}$$

Books

BID	AID	Title	PageCount
1	1	Pride and Prejudice	400
2	3	A Tale of Two Cities	350
3	4	Alice's Adventures in Wonderland	80
4	3	A Christmas Carol	100
5	2	1984	300

Authors

AID	Name	DateAdded
1	Jane Austen	2023-06-27 01:19:31
2	George Orwell	2023-08-19 23:52:44
3	Charles Dickens	2023-08-22 09:23:12
4	Lewis Carroll	2023-08-22 17:45:22

Summary

- Relational languages emerged after the publication of E.F. Codd's work to formally describe operations on relations.
- Relational Algebra is a formalism of the unary and binary operations that can be performed on relations.
- Relational Calculus is a type of declarative formalism to define what is needed without explicitly detailing how it will performed to get the results.
- The fundamentals of RL can be used to create multiple unique ways to solve one problem.

QUESTIONS?

THANK YOU!

References

- Codd, E. F. (1970). A relational model of data for large shared data banks. Communications of the ACM, 13(6), 377-387.
- Elmasri, R. et al. (2016). Fundamentals of Database Systems, 7th ed. Pearson Higher Education, 221 River Street, Hoboken, NJ 07030
- Kahl, W., Winter, M., & Oliveira, J. (2015). Relational and algebraic methods in computer science: 15th International Conference, RAMiCS 2015, Braga, Portugal, September 28 - October 1, 2015, Proceedings. Springer.