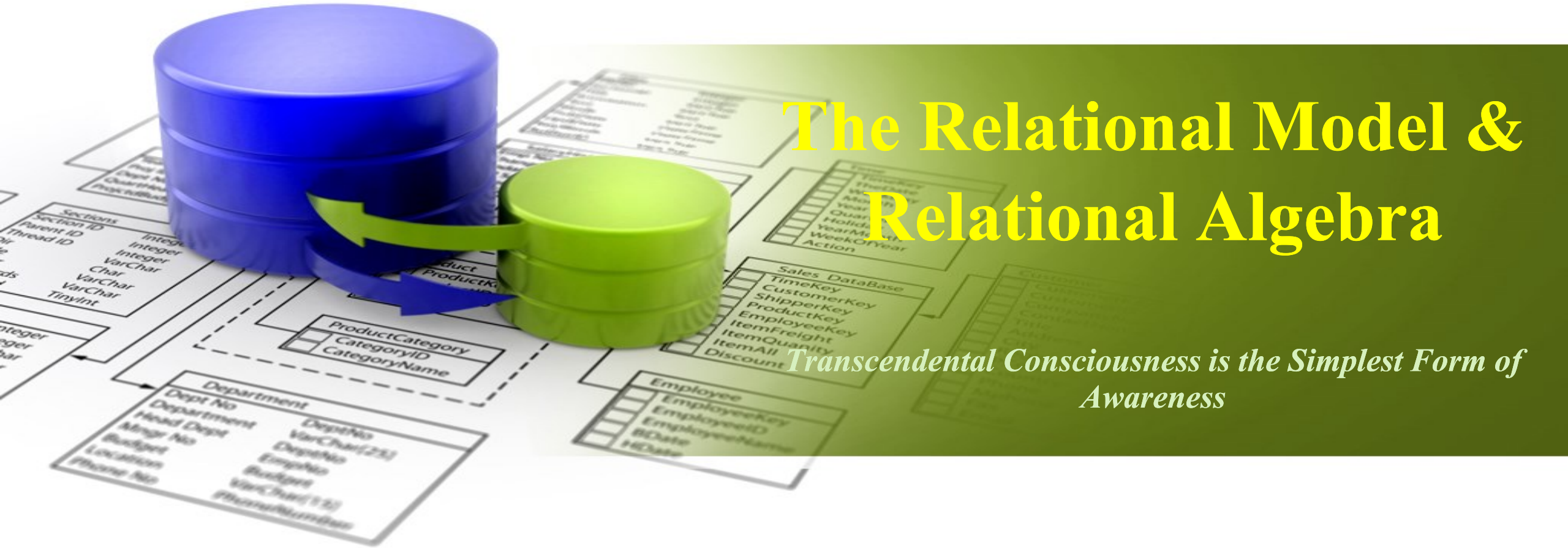
CS 422 - DBMS

Lesson 2 - Chapters 4 & 5



The Relational Model & Relational Algebra

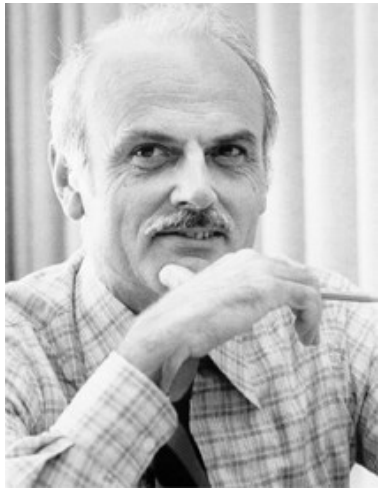
Transcendental Consciousness is the Simplest Form of Awareness

WHOLENESS OF THE LESSON

The relational model is an application of elementary relation theory to systems that provide shared access to large banks of formatted data. Science & Technology of Consciousness: Consciousness-based education is an application of Maharishi's Vedic Science in the field of education.

Relational Model Concepts

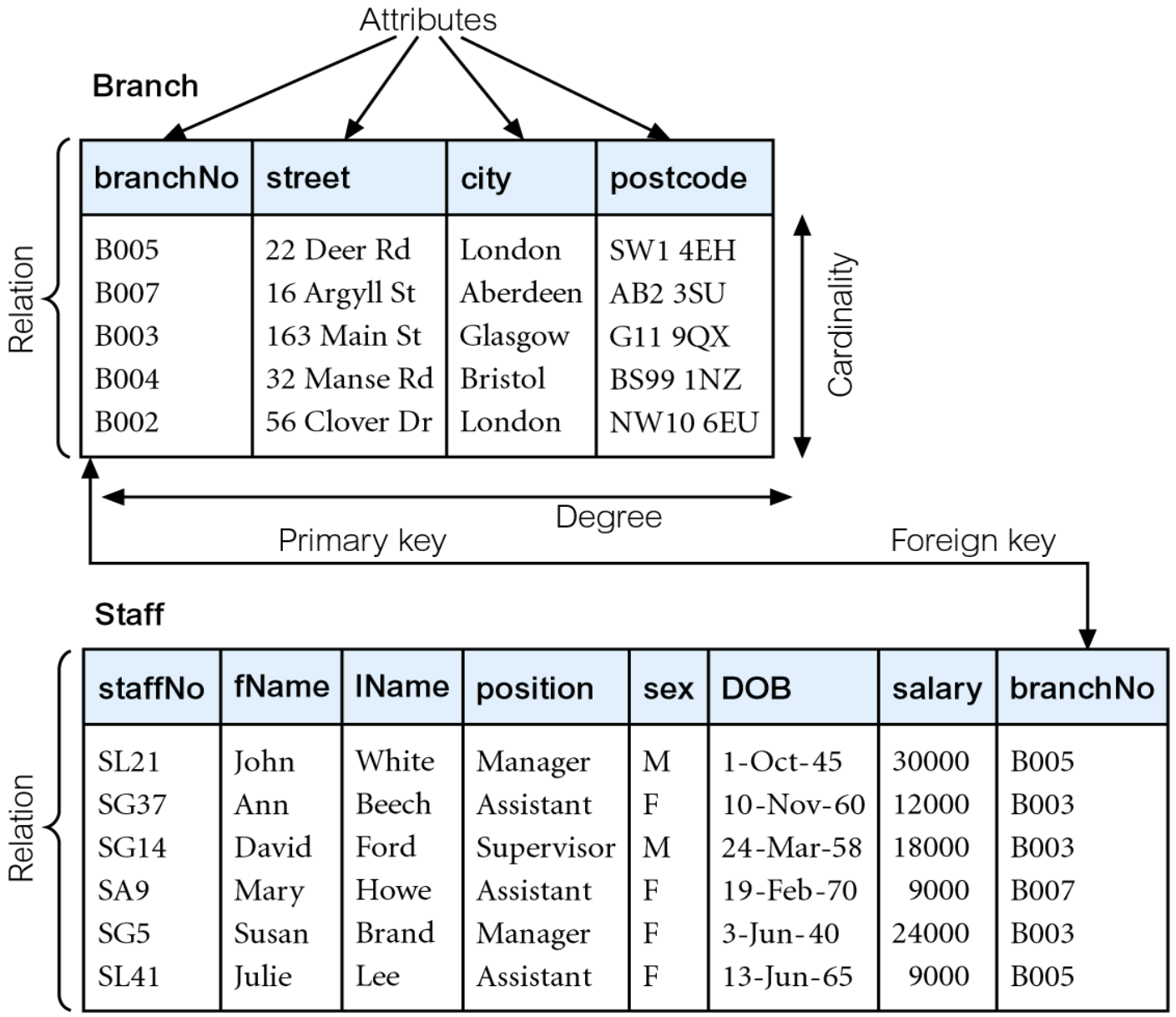
- A Relation is a mathematical concept based on the ideas of sets.
- In the relational model, all data is logically structured within relations.
- This model was first proposed by Dr. E.F. Codd of IBM Research in 1970 in the following paper:
  - "A Relational Model for Large Shared Data Banks," Communications of the ACM, June 1970
- The above paper caused a major revolution in the field of database management and earned Dr. Codd the coveted ACM Turing Award.



Relational Model Terminology

- A **Relation** is a table with columns and rows.
  - Only applies to logical structure of the database, not the physical structure.
- **Attribute** is a named column of a relation.
- **Domain** is the set of allowable values for one or more attributes.
- **Tuple** is a row of a relation.
- **Degree** is the number of attributes in a relation.
- **Cardinality** is the number of tuples in a relation.
- **Relational Database** is a collection of normalized relations with distinct relation names.
  - For now, consider normalized means "appropriately structured".

Instances of Branch and Staff Relations



Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

Alternative Terminology for Relational Model

Formal terms	Alternative 1	Alternative 2
Relation	Table	File
Tuple	Row	Record
Attribute	Column	Field

Examples of Attribute Domains

Attribute	Domain Name	Meaning	Domain Definition
branchNo	BranchNumbers	The set of all possible branch numbers	character: size 4, range B001–B999
street	StreetNames	The set of all street names in Britain	character: size 25
city	CityNames	The set of all city names in Britain	character: size 15
postcode	Postcodes	The set of all postcodes in Britain	character: size 8
sex	Sex	The sex of a person	character: size 1, value M or F
DOB	DatesOfBirth	Possible values of staff birth dates	date, range from 1-Jan-20, format dd-mmm-yy
salary	Salaries	Possible values of staff salaries	monetary: 7 digits, range 6000.00–40000.00

Schema

- **Relation schema**
  - Named relation defined by a set of attribute and domain name pairs.
- **Relational database schema**
  - Set of relation schemas, each with a distinct name.

Properties of Relations

- **Relation name** is **distinct** from all other relation names in relational DB schema.
- Each **cell** of relation contains exactly **one atomic** (single) value.
- Each **attribute** has a **distinct** name.
- **Values** of an **attribute** are all from the **same domain**.
- Each tuple is distinct; there are **no duplicate tuples**.
- **Order** of **attributes** has no significance.
- **Order** of **tuples** has no significance, theoretically.

Relational Keys

- Super Key
- Candidate Key
- Primary Key
- Foreign Key

- Composite Key
- Alternate Key
- Natural Key
- Surrogate Key

Relational Keys

- **Super Key**
  - An attribute, or set of attributes, that uniquely identifies a tuple within a relation at all time.

EmpSSN	EmpNum	Empname
9812345098	AB05	Shown
9876512345	AB06	Roslyn
199937890	AB07	James

Super keys

1. EmpSSN
2. EmpNum
3. EmpSSN, EmpNum
4. EmpSSN, EmpNum, Empname
5. EmpSSN, Empname
6. EmpNum, Empname

Relational Keys contd..

- **Candidate Key**
  - Superkey (K) such that no proper subset is a superkey within the relation.
    - In each tuple of R, values of K uniquely identify that tuple (uniqueness).
    - No proper subset of K has the uniqueness property (irreducibility).
  - Candidate Key is a minimal superkey that uniquely identifies a row

StudID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com
3	13	Dana	Natan	mno@yahoo.com

candidate key

candidate key

candidate key

Relational Keys contd..

- **Primary Key**
  - Candidate key selected to identify tuples uniquely within relation.
  - Characteristics of a primary key
    - Is unique
    - Cannot be NULL
    - Value never changes
    - Only one PK for one table.

StudID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com
3	13	Dana	Natan	mno@yahoo.com

primary key

Concept of NULL

- **Null**
  - Represents that a value for an attribute is currently unknown or not applicable for the tuple.
  - Deals with incomplete or exceptional data.
  - Represents the **absence of a value** and is not the same as zero or spaces, which are values.

Relational Keys contd..

- **Composite Key**
  - When a key consists of more than one attributes it's called as a **Composite Key**.

OrderNo	PorductID	Product Name	Quantity
B005	JAP102459	Mouse	5
B005	DKT321573	USB	10
B005	OMG446789	LCD Monitor	20
B004	DKT321573	USB	15
B002	OMG446789	Laser Printer	3

composite key

Relational Keys contd..

- **Alternate Keys**
  - Candidate keys that are not selected to be a primary key.
  - Can contain one NULL value.

StudID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com
3	13	Dana	Natan	mno@yahoo.com

primary key

alternate key

alternate key

Relational Keys contd..

- **Foreign Key**
  - Attribute, or set of attributes, within one relation that matches candidate key of some other (possibly same) relation.
  - A table can have 0 to many FKs.

primary key

DeptCode	DeptName
001	Science
002	English
005	Computer

Teacher ID	DeptCode	Fname	Lname
B002	002	David	Warner
B017	002	Sara	Joseph
B009	001	Mike	Brunton

primary key

foreign key

Relational Keys contd..

- **Natural Key**
  - Has a meaning in the business domain
    - SSN, ISBN
  - Often a composite key
    - (University, Department)

- **Surrogate Key**
  - Has no meaning in the business domain
    - AddressID
    - StudentID

Always a better choice

Integrity Constraints

- "Database integrity" = the correctness and consistency of stored data: it can be considered as another type of database protection.
- Integrity is concerned with the quality of data itself.
- Integrity is usually expressed in terms of constraints, which are consistency rules that the database is not permitted to violate.
  - For example, we may want to specify a constraint that no member of staff can manage more than 100 properties at any one time.

Integrity Constraints contd..

- Integrity constraints ensure that the data is accurate.
- **Entity Integrity**
  - In a base relation, no attribute of a primary key can be null.
- **Referential Integrity**
  - If foreign key exists in a relation, either foreign key value must match a candidate key value of some tuple in its home relation or foreign key value must be wholly null.

Integrity Constraints

- **General Constraints**
  - Additional rules specified by users or database administrators that define or constrain some aspect of the enterprise.
    - E.g. A branch may have only 20 staff members

- **Domain Constraints**
  - These are the domains of their associated attributes.

Main Point

A database can be modeled as a set of relations. It is always advantageous to find a simple basis for a complex field because it provides a way to manage the complexity of the field. Science & Technology of Consciousness: Vedic Science has discovered that the simplest form of awareness is the basis for all of the creation.

Relational Algebra

- Relational algebra operations work on one or more relations to define another relation without changing the original relations.
- Both operands and results are relations, so output from one operation can become input to another operation.
- Allows expressions to be nested, just as in arithmetic. This property is called closure.

Relational Algebra

- Five basic operations in relational algebra: **Selection**, **Projection**, **Cartesian product**, **Union**, and **Set Difference**.
- These perform most of the data retrieval operations needed.
- Also have **Join**, **Intersection**, and **Division** operations, which can be expressed in terms of 5 basic operations.
- Selection and Projection are ***unary*** operations, as they operate on one relation. The other operations are ***binary*** operations.

Selection (or Restriction)

$\sigma_{predicate}(R)$

(a) Selection

- Works on a single relation R and defines a relation that contains only those tuples (rows) of R that satisfy the specified condition (*predicate*).

Example - Selection (or Restriction)

- **List all staff with a salary greater than $10,000.**

$$\sigma_{salary>10000}(Staff)$$

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24- Mar-58	18000	B003
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003

Example - Selection (or Restriction)

Member

Member ID	Name	Date of Birth
1	Alice	03/03/1995
2	Bob	11/07/1993
3	Charlie	21/10/1997
4	Mike	16/09/1992
5	Katie	21/10/1997

Book

Book ID	Name	Author
1	Inferno	Dan Brown
2	Ash	Malinda Lo
3	Fences	August Wilson
4	Origin	Dan Brown
5	Inheritance	Malinda Lo

Borrow

Member ID	Book ID	Burrow Date	Return Date
1	1	02/03/2020	12/03/2020
3	5	05/03/2020	15/03/2020
3	3	10/03/2020	20/03/2020
4	2	13/03/2020	23/03/2020
5	4	13/03/2020	13/03/2020

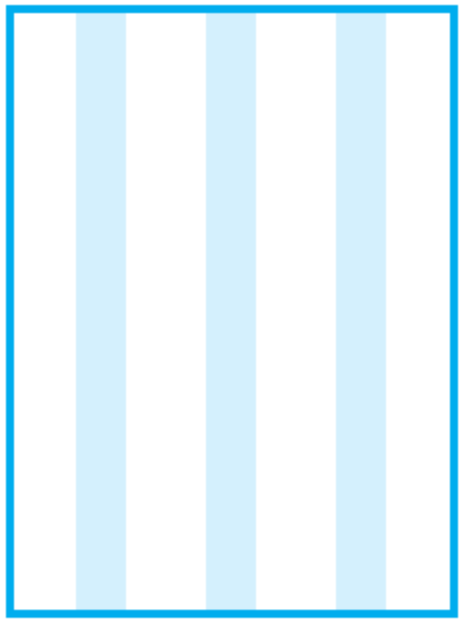
Query: Details of the members who were born on 21/10/1997.

$$\sigma_{Date\ of\ Birth=21/10/1997}(Member)$$

Member ID	Name	Date of Birth
3	Charlie	21/10/1997
5	Katie	21/10/1997

Projection

- $\Pi_{col1, \ldots, coln}(R)$



(b) Projection

- Works on a single relation R and defines a relation that contains a vertical subset of R, extracting the values of specified attributes and eliminating duplicates.

Example - Projection

- **Produce a list of salaries for all staff, showing only staffNo, fName, lName, and salary details.**

$$\Pi_{\text{staffNo, fName, lName, salary}}(\text{Staff})$$

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

staffNo	fName	lName	salary
SL21	John	White	30000
SG37	Ann	Beech	12000
SG14	David	Ford	18000
SA9	Mary	Howe	9000
SG5	Susan	Brand	24000
SL41	Julie	Lee	9000

Example - Projection

Member

Member ID	Name	Date of Birth
1	Alice	03/03/1995
2	Bob	11/07/1993
3	Charlie	21/10/1997
4	Mike	16/09/1992
5	Katie	21/10/1997

Book

Book ID	Name	Author
1	Inferno	Dan Brown
2	Ash	Malinda Lo
3	Fences	August Wilson
4	Origin	Dan Brown
5	Inheritance	Malinda Lo

Borrow

Member ID	Book ID	Burrow Date	Return Date
1	1	02/03/2020	12/03/2020
3	5	05/03/2020	15/03/2020
3	3	10/03/2020	20/03/2020
4	2	13/03/2020	23/03/2020
5	4	13/03/2020	13/03/2020

Member IDs of members who have borrowed books.

$\pi_{Member\ ID}(Borrow)$

Member ID
1
3
4
5

Query: *Member IDs of members and the Book IDs of the books they have borrowed books.*

$\pi_{Member\ ID,Book\ ID}(Borrow)$

Member ID	Book ID
1	1
3	5
3	3
4	2
5	4

Union

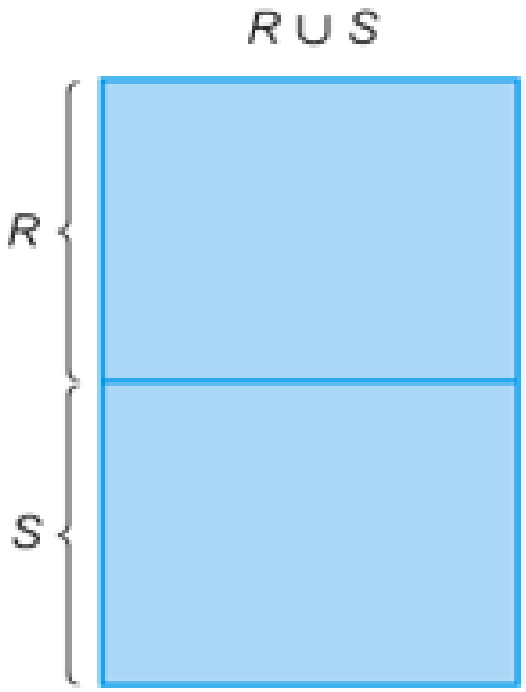
- **R ∪ S**
  - Union of two relations R and S defines a relation that contains all the tuples of R, or S, or both R and S, duplicate tuples being eliminated.
  - R and S must be **union-compatible**.

- If R and S have I and J tuples, respectively, union is obtained by concatenating them into one relation with a maximum of $(I + J)$ tuples.

Union-compatible

- Union is possible only if the schemas of the two relations match, that is, if they have the **same number of attributes** with each pair of corresponding attributes having the **same domain**.
- This is called as union compatible relations.
- *Note that attribute names are not used in defining union-compatibility*.
- In some cases, the projection operation is used to make 2 relations union-compatible.

Union



R

A	1
B	2
D	3
F	4
E	4

S

A	1
C	2
D	3
E	5

R UNION S

A	1
B	2
C	2
D	3
E	4
F	4
E	5

Example - Union

- **List all cities where there is either a branch office or a property for rent.**

$$\Pi_{city}(Branch) \cup \Pi_{city}(PropertyForRent)$$

Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

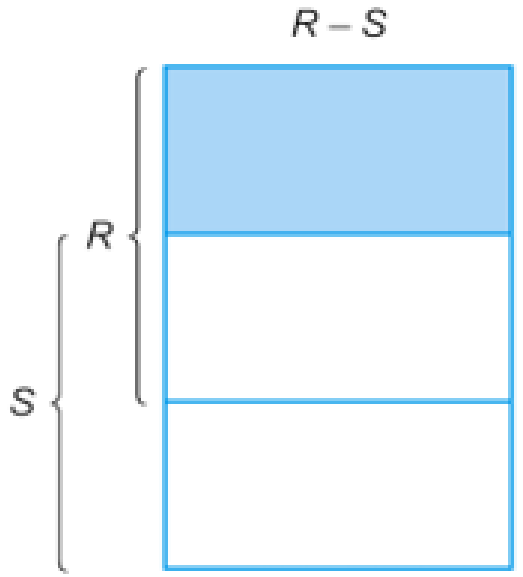
PropertyForRent

propertyNo	street	city	postcode	type	rooms	rent	ownerNo	staffNo	branchNo
PA14	16 Holhead	Aberdeen	AB7 5SU	House	6	650	CO46	SA9	B007
PL94	6 Argyll St	London	NW2	Flat	4	400	CO87	SL41	B005
PG4	6 Lawrence St	Glasgow	G11 9QX	Flat	3	350	CO40		B003
PG36	2 Manor Rd	Glasgow	G32 4QX	Flat	3	375	CO93	SG37	B003
PG21	18 Dale Rd	Glasgow	G12	House	5	600	CO87	SG37	B003
PG16	5 Novar Dr	Glasgow	G12 9AX	Flat	4	450	CO93	SG14	B003

city
London
Aberdeen
Glasgow
Bristol

Set Difference

- **R – S**
  - Defines a relation consisting of the tuples that are in relation R, but not in S.
  - R and S must be union-compatible.



Example - Set Difference

- **List all cities where there is a branch office but no properties for rent.**

$$\Pi_{city}(Branch) - \Pi_{city}(PropertyForRent)$$

Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

PropertyForRent

propertyNo	street	city	postcode	type	rooms	rent	ownerNo	staffNo	branchNo
PA14	16 Holhead	Aberdeen	AB7 5SU	House	6	650	CO46	SA9	B007
PL94	6 Argyll St	London	NW2	Flat	4	400	CO87	SL41	B005
PG4	6 Lawrence St	Glasgow	G11 9QX	Flat	3	350	CO40		B003
PG36	2 Manor Rd	Glasgow	G32 4QX	Flat	3	375	CO93	SG37	B003
PG21	18 Dale Rd	Glasgow	G12	House	5	600	CO87	SG37	B003
PG16	5 Novar Dr	Glasgow	G12 9AX	Flat	4	450	CO93	SG14	B003

city
Bristol

Example - Set Difference

ID	Name	Subject
100	Ankit	English
200	Pooja	Maths
300	Komal	Science

Relation R

ID	Name	Subject
100	Ankit	English
400	Kajol	French

Relation S

ID	Name	Subject
200	Pooja	Maths
300	Komal	Science

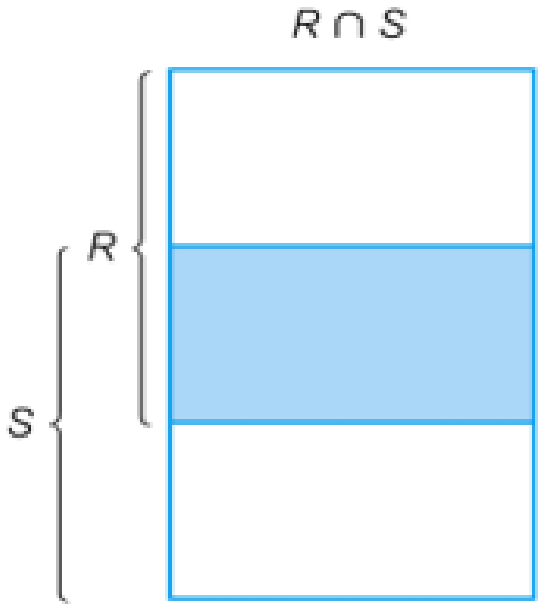
Relation R – S

Intersection

- **R ∩ S**
  - Defines a relation consisting of the set of all tuples that are in both R and S.
  - R and S must be union-compatible.

- Expressed using set difference operation as well :

  R ∩ S = R – (R – S)



Example - Intersection

- **List all cities where there is both a branch office and at least one property for rent.**

$$\Pi_{city}(Branch) \cap \Pi_{city}(PropertyForRent)$$

Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

PropertyForRent

propertyNo	street	city	postcode	type	rooms	rent	ownerNo	staffNo	branchNo
PA14	16 Holhead	Aberdeen	AB7 5SU	House	6	650	CO46	SA9	B007
PL94	6 Argyll St	London	NW2	Flat	4	400	CO87	SL41	B005
PG4	6 Lawrence St	Glasgow	G11 9QX	Flat	3	350	CO40		B003
PG36	2 Manor Rd	Glasgow	G32 4QX	Flat	3	375	CO93	SG37	B003
PG21	18 Dale Rd	Glasgow	G12	House	5	600	CO87	SG37	B003
PG16	5 Novar Dr	Glasgow	G12 9AX	Flat	4	450	CO93	SG14	B003

city
Aberdeen
London
Glasgow

Main Point

The relational algebra is a procedural query language which consists of a set of operations that take one or two relations as input and produce a new relation as their result. This closure allows for nested operations.

Science & Technology of Consciousness: Nature is orderly; from a mango seed grows a mango tree.

Cartesian Product

- **R X S**
  - Defines a relation that is the concatenation of every tuple of relation R with every tuple of relation S.
- Multiplies 2 relations to define another relation consisting of all possible pairs of tuples from the 2 relations.
- Therefore, if one relation has I tuples and N attributes and the other has J tuples and M attributes, the Cartesian product relation will contain (I * J) tuples with (N + M) attributes.
- It is possible that the two relations may have attributes with the same name. In this case, the attribute names are prefixed with the relation name to maintain the uniqueness of attribute names within a relation.

Example - Cartesian product

- **List the names and comments of all clients who have viewed a property for rent.**

($\Pi_{clientNo, fName, lName}$ (Client)) X ($\Pi_{clientNo, propertyNo, comment}$ (Viewing))

Client

clientNo	fName	lName	telNo	prefType	maxRent	eMail
CR76	John	Kay	0207-774-5632	Flat	425	john.kay@gmail.com
CR56	Aline	Stewart	0141-848-1825	Flat	350	astewart@hotmail.com
CR74	Mike	Ritchie	01475-392178	House	750	mritchie01@yahoo.co.uk
CR62	Mary	Tregear	01224-196720	Flat	600	maryt@hotmail.co.uk

Viewing

clientNo	propertyNo	viewDate	comment
CR56	PA14	24-May-13	too small
CR76	PG4	20-Apr-13	too remote
CR56	PG4	26-May-13	
CR62	PA14	14-May-13	no dining room
CR56	PG36	28-Apr-13	

client.clientNo	fName	lName	Viewing.clientNo	propertyNo	comment
CR76	John	Kay	CR56	PA14	too small
CR76	John	Kay	CR76	PG4	too remote
CR76	John	Kay	CR56	PG4	
CR76	John	Kay	CR62	PA14	no dining room
CR76	John	Kay	CR56	PG36	
CR56	Aline	Stewart	CR56	PA14	too small
CR56	Aline	Stewart	CR76	PG4	too remote
CR56	Aline	Stewart	CR56	PG4	
CR56	Aline	Stewart	CR62	PA14	no dining room
CR56	Aline	Stewart	CR56	PG36	
CR74	Mike	Ritchie	CR56	PA14	too small
CR74	Mike	Ritchie	CR76	PG4	too remote
CR74	Mike	Ritchie	CR56	PG4	
CR74	Mike	Ritchie	CR62	PA14	no dining room
CR74	Mike	Ritchie	CR56	PG36	
CR62	Mary	Tregear	CR56	PA14	too small
CR62	Mary	Tregear	CR76	PG4	too remote
CR62	Mary	Tregear	CR56	PG4	
CR62	Mary	Tregear	CR62	PA14	no dining room
CR62	Mary	Tregear	CR56	PG36	

Example - Cartesian Product & Selection

Use selection operation to extract those tuples where Client.clientNo = Viewing.clientNo

$$\sigma_{(Client.clientNo=Viewing.clientNo)}((\Pi_{clientNo,fName,lName}(Client))X(\Pi_{clientNo,propertyNo,comment}(Viewing)))$$

client.clientNo	fName	lName	Viewing.clientNo	propertyNo	comment
CR76	John	Kay	CR56	PA14	too small
CR76	John	Kay	CR76	PG4	too remote
CR76	John	Kay	CR56	PG4	
CR76	John	Kay	CR62	PA14	no dining room
CR76	John	Kay	CR56	PG36	
CR56	Aline	Stewart	CR56	PA14	too small
CR56	Aline	Stewart	CR76	PG4	too remote
CR56	Aline	Stewart	CR56	PG4	
CR56	Aline	Stewart	CR62	PA14	no dining room
CR56	Aline	Stewart	CR56	PG36	
CR74	Mike	Ritchie	CR56	PA14	too small
CR74	Mike	Ritchie	CR76	PG4	too remote
CR74	Mike	Ritchie	CR56	PG4	
CR74	Mike	Ritchie	CR62	PA14	no dining room
CR74	Mike	Ritchie	CR56	PG36	
CR62	Mary	Tregear	CR56	PA14	too small
CR62	Mary	Tregear	CR76	PG4	too remote
CR62	Mary	Tregear	CR56	PG4	
CR62	Mary	Tregear	CR62	PA14	no dining room
CR62	Mary	Tregear	CR56	PG36	

client.clientNo	fName	lName	Viewing.clientNo	propertyNo	comment
CR76	John	Kay	CR76	PG4	too remote
CR56	Aline	Stewart	CR56	PA14	too small
CR56	Aline	Stewart	CR56	PG4	
CR56	Aline	Stewart	CR56	PG36	
CR62	Mary	Tregear	CR62	PA14	no dining room

Cartesian product and Selection can be reduced to a single operation called as ***Join***.

Join Operations

- Join is a special form of cross product of two tables.
- Various forms of join operations:
  - Theta join
  - Equijoin (a particular type of Theta join)
  - Natural join
  - Outer join
  - Semijoin

Theta join (θ - join)

- $R \bowtie_F S$
  - Defines a relation that contains tuples satisfying the predicate F from the Cartesian product of R and S.

- Can rewrite Theta join using basic Selection and Cartesian product operations.

$$R \bowtie_F S = \sigma_F (R \times S)$$

- Degree of a Theta join is sum of degrees of the operand relations R and S.

Theta join (θ - join) Example

$$\textbf{Car} \bowtie_{\text{CarPrice} > \text{BoatPrice}} \textbf{Boat}$$

Car

CarModel	CarPrice
CarA	20'000
CarB	30'000
CarC	50'000

Boat

BoatModel	BoatPrice
Boat1	10'000
Boat2	40'000
Boat3	60'000

CarModel	CarPrice	BoatModel	BoatPrice
CarA	20'000	Boat1	10'000
CarB	30'000	Boat1	10'000
CarC	50'000	Boat1	10'000
CarC	50'000	Boat2	40'000

Example - Equijoin

- **List the names and comments of all clients who have viewed a property for rent.**

$$(\Pi_{\text{clientNo, fName, lName}}(\text{Client})) \bowtie_{\text{Client.clientNo = Viewing.clientNo}} (\Pi_{\text{clientNo, propertyNo,comment}}(\text{Viewing}))$$

Client

clientNo	fName	lName	telNo	prefType	maxRent	eMail
CR76	John	Kay	0207-774-5632	Flat	425	john.kay@gmail.com
CR56	Aline	Stewart	0141-848-1825	Flat	350	astewart@hotmail.com
CR74	Mike	Ritchie	01475-392178	House	750	mritchie01@yahoo.co.uk
CR62	Mary	Tregear	01224-196720	Flat	600	maryt@hotmail.co.uk

Viewing

clientNo	propertyNo	viewDate	comment
CR56	PA14	24-May-13	too small
CR76	PG4	20-Apr-13	too remote
CR56	PG4	26-May-13	
CR62	PA14	14-May-13	no dining room
CR56	PG36	28-Apr-13	

client.clientNo	fName	lName	Viewing.clientNo	propertyNo	comment
CR76	John	Kay	CR76	PG4	too remote
CR56	Aline	Stewart	CR56	PA14	too small
CR56	Aline	Stewart	CR56	PG4	
CR56	Aline	Stewart	CR56	PG36	
CR62	Mary	Tregear	CR62	PA14	no dining room

Natural join

- $R \bowtie S$
  - An Equijoin of the two relations R and S over all common attributes *x*.
  - The natural join operation performs an equijoin over all the attributes in the two relations that have the same name.
  - **One occurrence of each common attribute is eliminated from the result.**

Example - Natural join

- **List the names and comments of all clients who have viewed a property for rent.**

$$(\Pi_{clientNo,\ fName,\ lName}(Client)) \bowtie (\Pi_{clientNo,\ propertyNo,\ comment}(Viewing))$$

Client

clientNo	fName	lName	telNo	prefType	maxRent	eMail
CR76	John	Kay	0207-774-5632	Flat	425	john.kay@gmail.com
CR56	Aline	Stewart	0141-848-1825	Flat	350	astewart@hotmail.com
CR74	Mike	Ritchie	01475-392178	House	750	mritchie01@yahoo.co.uk
CR62	Mary	Tregear	01224-196720	Flat	600	maryt@hotmail.co.uk

Viewing

clientNo	propertyNo	viewDate	comment
CR56	PA14	24-May-13	too small
CR76	PG4	20-Apr-13	too remote
CR56	PG4	26-May-13	
CR62	PA14	14-May-13	no dining room
CR56	PG36	28-Apr-13	

clientNo	fName	lName	propertyNo	comment
CR76	John	Kay	PG4	too remote
CR56	Aline	Stewart	PA14	too small
CR56	Aline	Stewart	PG4	
CR56	Aline	Stewart	PG36	
CR62	Mary	Tregear	PA14	no dining room

Example - Natural join

Member

Member ID	Name	Date of Birth
1	Alice	03/03/1995
2	Bob	11/07/1993
3	Charlie	21/10/1997
4	Mike	16/09/1992
5	Katie	21/10/1997

Borrow

Member ID	Book ID	Burrow Date	Return Date
1	1	02/03/2020	12/03/2020
3	5	05/03/2020	15/03/2020
3	3	10/03/2020	20/03/2020
4	2	13/03/2020	23/03/2020
5	4	13/03/2020	13/03/2020

$$Member \bowtie Borrow$$

Member ID	Name	Date of Birth	Book ID	Borrow Date	Return Date
1	Alice	03/03/1995	1	02/03/2020	12/03/2020
3	Charlie	21/10/1997	5	05/03/2020	15/03/2020
3	Charlie	21/10/1997	3	10/03/2020	20/03/2020
4	Mike	16/09/1992	2	13/03/2020	23/03/2020
5	Katie	21/10/1997	4	13/03/2020	13/03/2020

Outer join

- Typically, you create joins that return rows only if they satisfy join conditions; these are called **Inner Joins** and are the default join used when querying.
- However, sometimes you may want to preserve all the rows in one table.
- To display rows in the result that do not have matching values in the join column, use **Outer Join.**
- Advantage of outer join is that the information is preserved; that is, the Outer join preserves tuples that would have been lost by other types of join.

Left Outer join

- $R \mathbin{⟕} S$
  - (Left) outer join is a join in which tuples from R that do not have matching values in common columns of S are also included in result relation.
  - Missing values in the 2nd relation are set to null.
  - One occurrence of each common attribute is eliminated from the result.

Example - Left Outer join

- **Produce a status report on property viewings.**

$$\Pi_{\text{propertyNo, street, city}}(\text{PropertyForRent}) \mathbin{⟕} \text{Viewing}$$

PropertyForRent

propertyNo	street	city	postcode	type	rooms	rent	ownerNo	staffNo	branchNo
PA14	16 Holhead	Aberdeen	AB7 5SU	House	6	650	CO46	SA9	B007
PL94	6 Argyll St	London	NW2	Flat	4	400	CO87	SL41	B005
PG4	6 Lawrence St	Glasgow	G11 9QX	Flat	3	350	CO40		B003
PG36	2 Manor Rd	Glasgow	G32 4QX	Flat	3	375	CO93	SG37	B003
PG21	18 Dale Rd	Glasgow	G12	House	5	600	CO87	SG37	B003
PG16	5 Novar Dr	Glasgow	G12 9AX	Flat	4	450	CO93	SG14	B003

Viewing

clientNo	propertyNo	viewDate	comment
CR56	PA14	24-May-13	too small
CR76	PG4	20-Apr-13	too remote
CR56	PG4	26-May-13	
CR62	PA14	14-May-13	no dining room
CR56	PG36	28-Apr-13	

propertyNo	street	city	clientNo	viewDate	comment
PA14	16 Holhead	Aberdeen	CR56	24-May-01	too small
PA14	16 Holhead	Aberdeen	CR62	14-May-01	no dining room
PL94	6 Argyll St	London	**null**	**null**	**null**
PG4	6 Lawrence St	Glasgow	CR76	20-Apr-01	too remote
PG4	6 Lawrence St	Glasgow	CR56	26-May-01	
PG36	2 Manor Rd	Glasgow	CR56	28-Apr-01	
PG21	18 Dale Rd	Glasgow	**null**	**null**	**null**
PG16	5 Novar Dr	Glasgow	**null**	**null**	**null**

Right Outer Join

- $R \overline{\ltimes} S$
- Keeping all the tuples from the right relation.
- However, if no matching tuple is found in the left relation, then the attributes of the left relation in the join result are filled with null values.

Semijoin

- $R \triangleright_F S$
  - Defines a relation that contains the tuples of R that participate in the join of R with S satisfying the predicate *F*.

- Can rewrite Semijoin using Projection and Join:

$$R \triangleright_F S = \Pi_A (R \bowtie_F S)$$

Where A is the set of all attributes of R.

Example - Semijoin

- **List complete details of all staff who work at the branch in Glasgow.** (only want to see the attributes of the Staff relation)

Staff ▷ $_{\text{Staff.branchNo=Branch.branchNo}}$(σ$_{\text{city='Glasgow'}}$(Branch))

Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24- Mar-58	18000	B003
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003

Example - Semijoin

Employee

Name	EmpId	DeptName
Harry	3415	Finance
Sally	2241	Sales
George	3401	Finance
Harriet	2202	Production

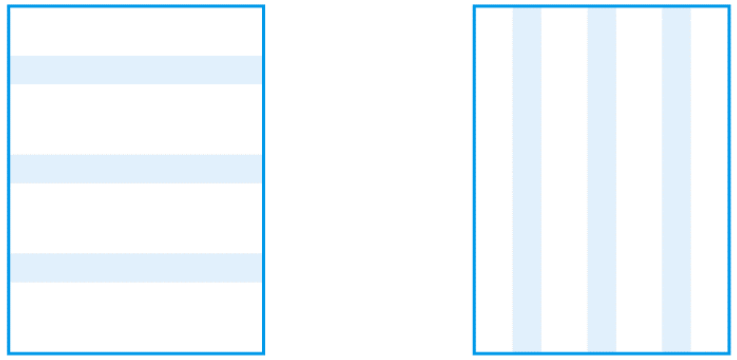
Dept

DeptName	Manager
Sales	Bob
Sales	Thomas
Production	Katie
Production	Mark

Employee ▷ Dept

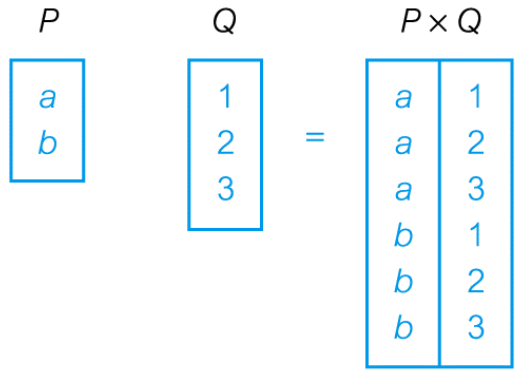
Name	EmpId	DeptName
Sally	2241	Sales
Harriet	2202	Production

Relational Algebra Operations

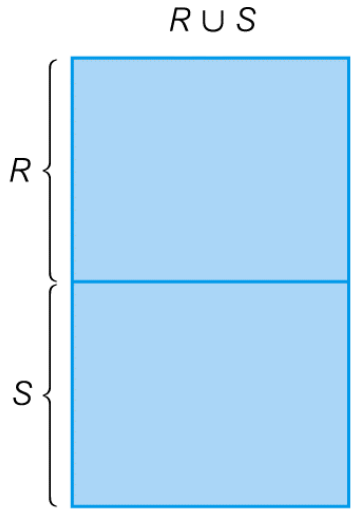


(a) Selection

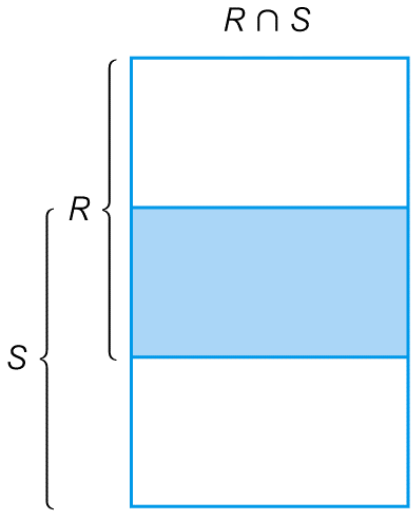
(b) Projection



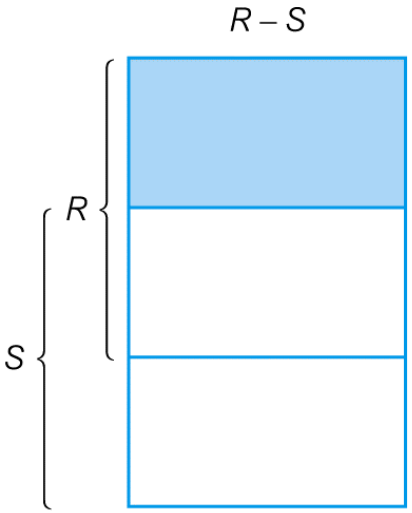
(c) Cartesian product



(d) Union



(e) Intersection



(f) Set difference

Relational Algebra Operations contd..

T

A	B
a	1
b	2

U

B	C
1	x
1	y
3	z

$T \bowtie U$

A	B	C
a	1	x
a	1	y

(g) Natural join

$T \triangleright_B U$

A	B
a	1

(h) Semijoin

$T ⟕_C U$

A	B	C
a	1	x
a	1	y
b	2	

(i) Left Outer join

UNITY CHART

CONNECTING THE PARTS OF KNOWLEDGE WITH THE WHOLENESS OF KNOWLEDGE:

Relational Model Basics & Relational Algebra Operations

1. Science repeatedly calls upon mathematics to help it model the real world.
2. The operators of the relational algebra have been found to be ideal for modeling the operations that are needed to query and update a database.

---

3. **Transcendental consciousness** is the experience of the simplest and most abstract state of awareness which underlies all states of greater excitation.
4. **Impulses within the Transcendental Field**: Nature accomplishes what it needs by having its impulses in the transcendental field be as efficient as possible.
5. **Wholeness moving within itself:** In unity consciousness one experiences everything as excitations of pure consciousness that underlies and connects all diversity.