COMP 430: Advanced Database Topics

Lecture 1: Review of Relational Model and Relational Algebra

Objectives

- Review the relational model
  - Define the terminology of relational model.
  - Identify Candidate Keys, Primary Keys and Foreign Keys.
  - Explore entity integrity and referential integrity.
- Review of relational algebra
  - Explain the basics of relational algebra.
  - Notion of relational algebra operations operating on relations and producing relations.
  - Relational algebra operations.

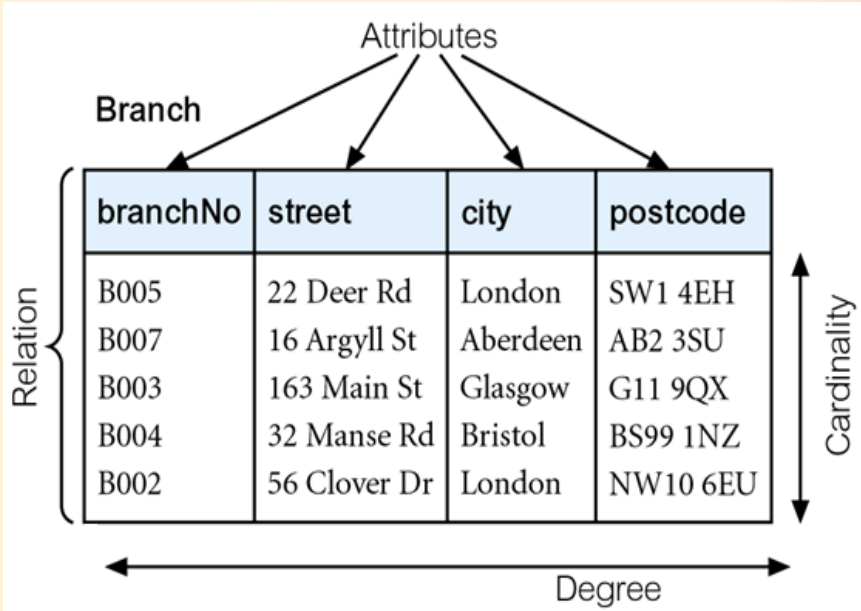
Outline

- Relational Model
  - Relational Model Terminology
  - Relational Keys
  - Domain and Integrity Constraints
  - Properties of Relations
  - Example Relational Database Schema
- Relational Algebra
  - Introduction
  - Unary Operations
  - Set Operations
  - Join Operations
  - Division Operation
  - Aggregation & Grouping Operations

Relational Model

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.



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

Relation

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

Relational Model Terminology

- Attributes and Domains

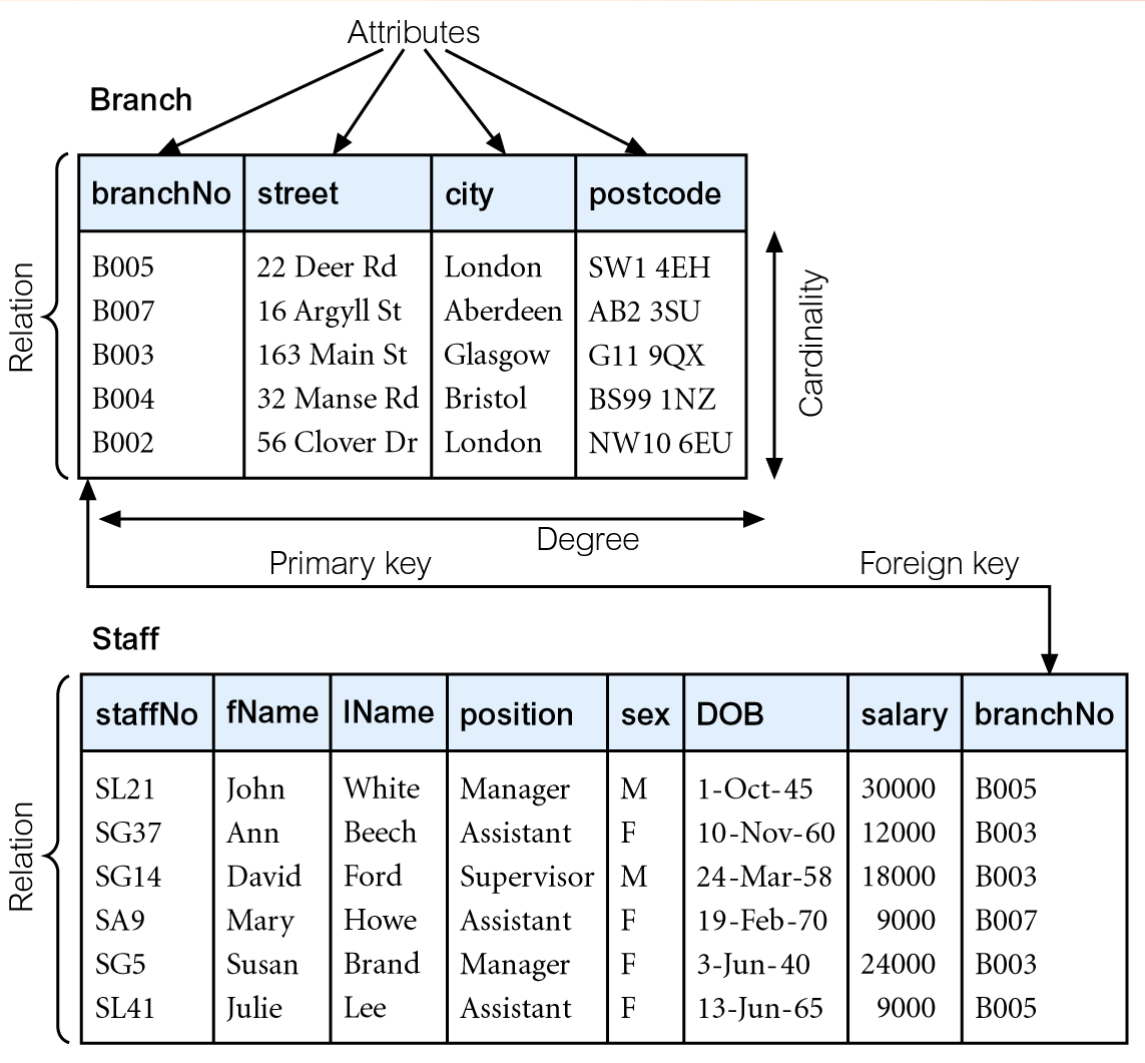
Attribute	Domain Name	Meaning	Domain Definition
branchNo	BranchNumbers	The set of all possible branch numbers	character: size 4, range B001–B999
street	StreetNames	The set of all street names in Britain	character: size 25
city	CityNames	The set of all city names in Britain	character: size 15
postcode	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

Relational Model Terminology

- **Relation schema**: Named relation defined by a set of attribute and domain name pairs.
  - Let $A_1, A_2, \ldots, A_n$ be attributes with domains $D_1, D_2, \ldots, D_n$. Then the set $\{A_1{:}D_1, A_2{:}D_2, \ldots, A_n{:}D_n\}$ is a relation schema.
  - A relation R defined by a relation schema S is a set of mappings from the attribute names to their corresponding domains.
  - Thus, relation R is a set of n-tuples: $\{A_1{:}d_1, A_2{:}d_2, \ldots, A_n{:}d_n\}$ such that $d_1 \in D_1, d_2 \in D_2, \ldots, d_n \in D_n$
- **Relational database schema:** Set of relation schemas, each with a distinct name.

Relational Keys

- **Superkey:** An attribute, or set of attributes, that uniquely identifies a tuple within a relation.
- **Candidate Key:** A superkey with minimal set of attributes
  - Uniqueness & irreducibility
- **Primary Key:** Candidate key selected to identify tuples uniquely within relation.
- **Alternate Keys:** Candidate keys that are not selected to be primary key.
- **Foreign Key:** Attribute, or set of attributes, within one relation that matches candidate key of some (possibly same) relation.

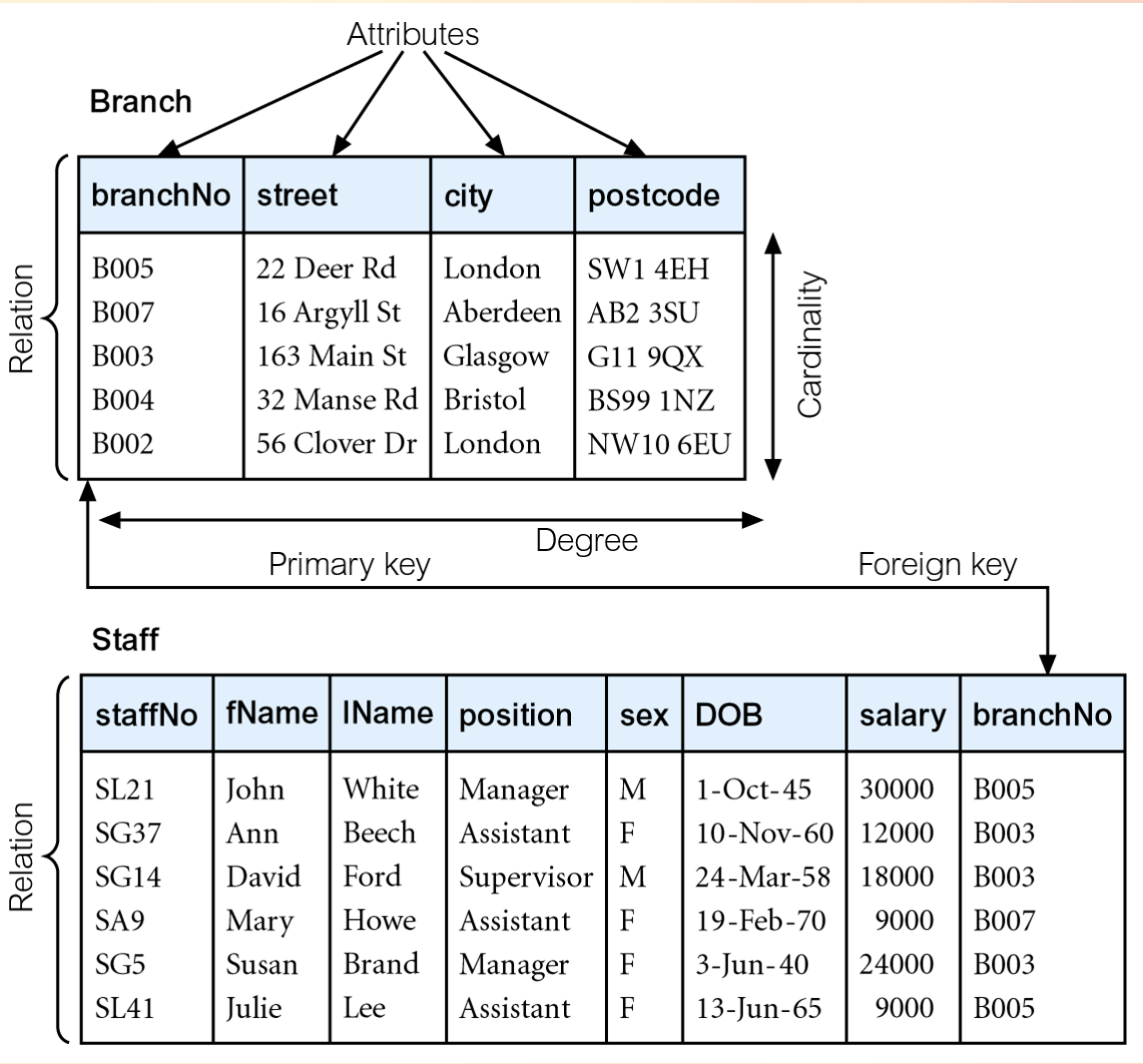


branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

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

Domain and Integrity Constraints

- Domain Constraints
  - Restriction on set of values allowed for an attribute
- Integrity Constraints
  - **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.
  - **General Constraints:** Additional rules specified by users or database administrators that define or constrain some aspect of the enterprise.



Properties of Relations

- Relation name is distinct from all other relation names in relational 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.

Example Relational Database Schema

- The relational schema for part of the *DreamHome* case study

Branch	(branchNo, street, city, postcode)
Staff	(staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent	(propertyNo, street, city, postcode, type, rooms, rent, ownerNo, staffNo, branchNo)
Client	(clientNo, fName, lName, telNo, prefType, maxRent, eMail)
PrivateOwner	(ownerNo, fName, lName, address, telNo, eMail, password)
Viewing	(clientNo, propertyNo, viewDate, comment)
Registration	(clientNo, branchNo, staffNo, dateJoined)

Relational Algebra

Introduction

- Relational algebra and relational calculus are formal languages associated with the relational model.
  - **They operate on relations and produce relations as results – Important Concept**
- Informally, relational algebra is a (high-level) procedural language and relational calculus a non-procedural language.
- However, formally both are equivalent to one another.
- A language that has the power to produce any relation that can be derived using relational calculus is **relationally complete**.
- These languages produce the basis of modern DMLs (SQL).
  - Modern DMLs have more expressive power

INTRODUCTION

- **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.
- **The relational algebra is a relation-at-a-time (or set) language in which all tuples, possibly from several relations, are manipulated in one statement without looping.**

Introduction

- Five basic operations:
  - Unary Operations: Selection, Projection
  - Set Operations: Cartesian product, Union, and Set Difference.
  - These perform most of the data retrieval operations needed.
- Also have Join, Intersection, and Division operations
  - These can be expressed in terms of 5 basic operations.

SELECTION

(a) Selection

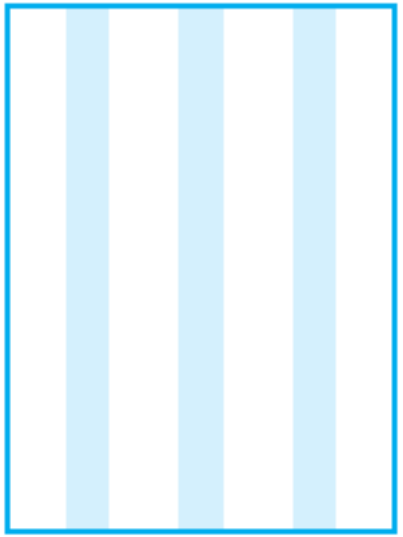
- Selection: Select a subset of tuples from a relation to form a new relation
- Selection (or Restriction) : $\sigma_{predicate}$ **(R)**
  - **Works on a single relation R** and **defines a relation** that contains **only those tuples** (**rows**) of R that **satisfy the specified condition (*predicate*)**.
- Selection Example
  - Select 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

PROJECTION


(b) Projection

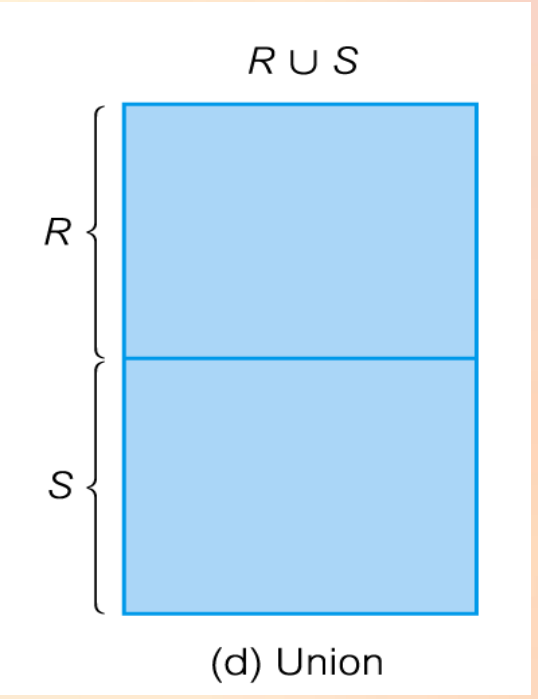
- Projection: Select a subset of attributes from a relation to form a new relation
- Projection : $\Pi_{col1, \ldots, coln}(R)$
  - Works on a single relation R and **defines a relation that contains a vertical subset of R, extracting the values of specified attributes** and **eliminating duplicates**.
- Projection Example:
  - **Produce a list of salaries for all staff, showing only staffNo, fName, lName, and salary details :** $\Pi_{staffNo, fName, lName, salary}(Staff)$

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

UNION



(d) Union

- **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.
  - 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.
- R and S must be **union-compatible**
  - **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**.
- **Union** Example:
  - List all cities where there is either a branch office or a property for rent.
  - Π_{city}**(Branch)** $\cup$ Π_{city}**(PropertyForRent)**

Branch

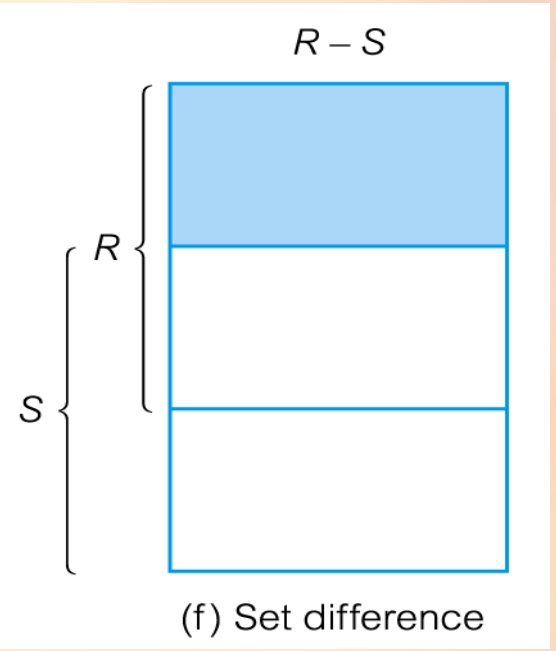
branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

PropertyForRent

propertyNo	street	city	postcode	ty
PA14	16 Holhead	Aberdeen	AB7 5SU	Ho
PL94	6 Argyll St	London	NW2	Fla
PG4	6 Lawrence St	Glasgow	G11 9QX	Fla
PG36	2 Manor Rd	Glasgow	G32 4QX	Fla
PG21	18 Dale Rd	Glasgow	G12	Ho
PG16	5 Novar Dr	Glasgow	G12 9AX	Fla

city
London
Aberdeen
Glasgow
Bristol

Set Difference



(f) Set difference

- **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**
- **Set Difference** Example:
  - List all cities where there is a branch office but no properties for rent.
  - Π_{city}**(Branch) -** Π_{city}**(PropertyForRent)**

Branch

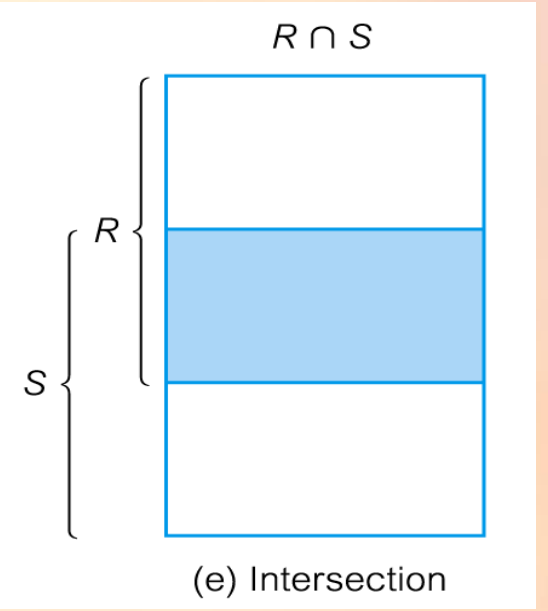
branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

PropertyForRent

propertyNo	street	city	postcode	ty
PA14	16 Holhead	Aberdeen	AB7 5SU	Ho
PL94	6 Argyll St	London	NW2	Fla
PG4	6 Lawrence St	Glasgow	G11 9QX	Fla
PG36	2 Manor Rd	Glasgow	G32 4QX	Fla
PG21	18 Dale Rd	Glasgow	G12	Ho
PG16	5 Novar Dr	Glasgow	G12 9AX	Fla

city
Bristol

INTERSECTION



- **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 basic operations: R ∩ S = R – (R – S)
- **Intersection** Example:
  - List all cities where there is both a branch office and at least one property for rent.
  - **Π_{city}(Branch) ∩ Π_{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

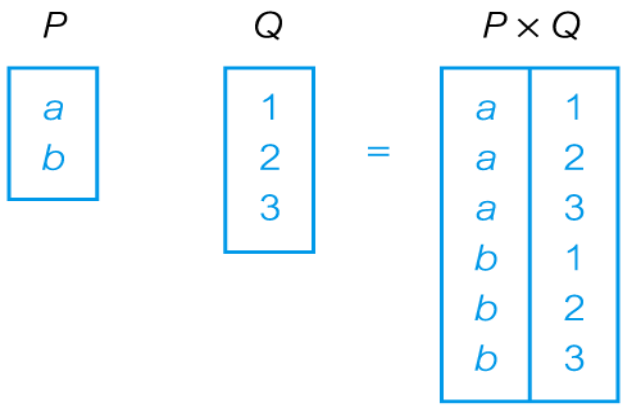
PropertyForRent

propertyNo	street	city	postcode	ty
PA14	16 Holhead	Aberdeen	AB7 5SU	Ho
PL94	6 Argyll St	London	NW2	Fla
PG4	6 Lawrence St	Glasgow	G11 9QX	Fla
PG36	2 Manor Rd	Glasgow	G32 4QX	Fla
PG21	18 Dale Rd	Glasgow	G12	Ho
PG16	5 Novar Dr	Glasgow	G12 9AX	Fla

city
Aberdeen
London
Glasgow

CARTESIAN PRODUCT

- **Cartesian Product**: R X S
- Defines a relation that is the concatenation of every tuple of relation R with every tuple of relation S.
- **Cartesian Product** Example:
  - List the names and comments of all clients who have viewed a property for rent.
  - **$(\Pi_{clientNo, fName, lName}(Client))$ X $(\Pi_{clientNo, propertyNo, comment}(Viewing))$**



(c) Cartesian product

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	

Figure 5.7 Cartesian product of reduced Client and Viewing relations.

Set Operations

- Example with Cartesian Product and Selection:
  - Use selection operation to extract those tuples where Client.clientNo = Viewing.clientNo.
  - $\sigma_{\text{Client.clientNo} = \text{Viewing.clientNo}}((\Pi_{\text{clientNo, fName, lName}}(\text{Client})) \times (\Pi_{\text{clientNo, propertyNo, comment}}(\text{Viewing})))$

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

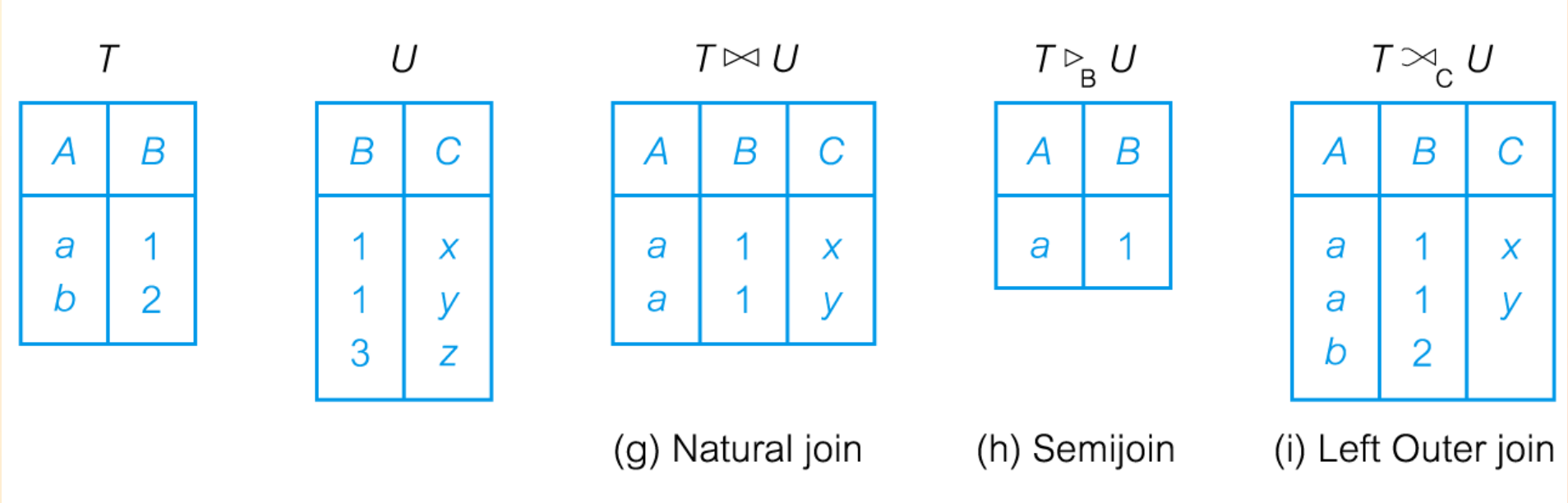
Figure 5.8 Restricted Cartesian product of reduced Client and Viewing relations.

Relational Algebra: Join Operations

- Cartesian product and Selection can be reduced to a single operation called a **Join**.
- Join is a derivative of Cartesian product.
- Equivalent to **performing a Selection**, **using join predicate as selection formula**, **over Cartesian product** of the two operand relations.
- One of the most difficult operations to implement efficiently in an RDBMS and one reason why RDBMSs have intrinsic performance problems.

Relational Algebra: Join Operations

- Various forms of join operation
  - Theta join, Equijoin (Theta join with condition for predicate is =), Natural join (Equijoin with column names being same and duplicate columns removed), Outer join, Semijoin (project attributes of first operand)



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

$T \bowtie_C U$

A	B	C
a	1	x
a	1	y
b	2	

(i) Left Outer join

Relational Algebra: Join Operations

- **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.
  - The **predicate F** is of the form $R.a_i \ \theta \ S.b_i$ where θ may be **one of the comparison operators** (<, ≤, >, ≥, =, ≠).
- 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.
- If **predicate F** contains only equality (=), the term **Equijoin** is used.

Relational Algebra: Join Operations

- Example – **Equijoin**
  - List the names and comments of all clients who have viewed a property for rent.
  - $(\Pi_{\text{clientNo, fName, lName}}(\text{Client})) \bowtie \text{Client.clientNo} = \text{Viewing.clientNo} (\Pi_{\text{clientNo, propertyNo, comment}}(\text{Viewing}))$

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

Restricted Cartesian product of reduced Client and Viewing relations.

Relational Algebra: Join Operations

- **Natural Join: R ⋈ S**
- An **Equijoin of the two relations R** and **S over all common attributes x**.
  - 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.
  - $(\Pi_{clientNo, fName, lName}(Client)) \bowtie (\Pi_{clientNo, propertyNo, comment}(Viewing))$

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

Natural join of restricted Client and Viewing relations.

Relational Algebra: Join Operations

- **Outer Join:**
  - To **display rows in the result that do not have matching values in the join column**, use Outer join.
- **Left Outer Join: R ⟕ S**
  - (Left) outer join is join in which **tuples from R that do not have matching values in common columns of S are also included in result** relation.
- **Right Outer Join is similar - ⟖**

Relational Algebra: Join Operations

- Example – **Left Outer Join**
  - Produce a status report on property viewings.
  - $\Pi_{\text{propertyNo, street , city}}$ (PropertyForRent) ⟕ Viewing

propertyNo	street	city	clientNo	viewDate	comment
PA14	16 Holhead	Aberdeen	CR56	24-May-13	too small
PA14	16 Holhead	Aberdeen	CR62	14-May-13	no dining room
PL94	6 Argyll St	London	null	null	null
PG4	6 Lawrence St	Glasgow	CR76	20-Apr- 13	too remote
PG4	6 Lawrence St	Glasgow	CR56	26-May-13	
PG36	2 Manor Rd	Glasgow	CR56	28-Apr- 13	
PG21	18 Dale Rd	Glasgow	null	null	null
PG16	5 Novar Dr	Glasgow	null	null	null

) Left (natural) Outer join of PropertyForRent and Viewing relations.

Viewing

clientNo	propertyNo	viewDate	comment

PropertyForRent

propertyNo	street	city	postcode	type	rooms	rent	ownerNo	staffNo	branchNo
PA14	16 Holhead	Aberdeen	AB7 5SU	House	6	650	CO46	SA9	B007

Relational Algebra: Join Operations

- **Semi Join: $R \rhd_F S$**
- Defines a relation that contains the tuples of R that participate in the join of R with S.
- Can rewrite Semijoin using Projection and Join:

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

A s the set of all the attributes of relation R.

Relational Algebra: Join Operations

- Example – **Semi Join**
  - List complete details of all staff who work at the branch in Glasgow.
  - **Staff** $\triangleright_{\text{Staff.branchNo=Branch.branchNo}}(\sigma_{\text{city='Glasgow'}}(\text{Branch}))$

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

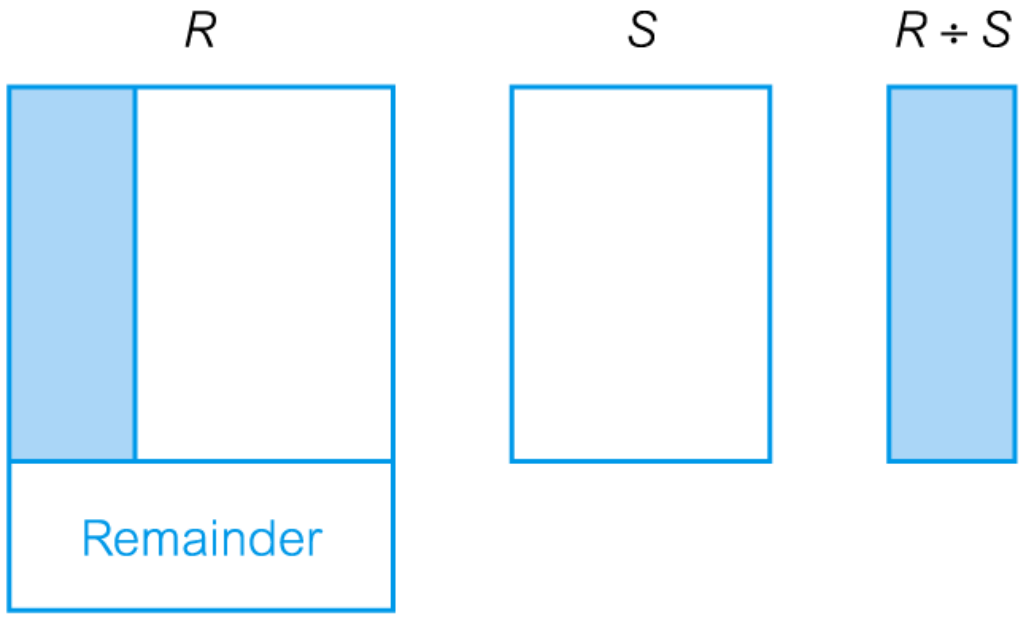
Semijoin of Staff and Branch relations.

Relational Algebra: Division Operation

- **Division** Operation: **R ÷ S**
- Assume that relation R is defined over the attribute set A and relation S is defined over the attribute set B such that B ⊆ A (B is a subset of A).
- Let C = A - B, that is, C is the set of attributes of R that are not attributes of S.
- The Division operation defines a relation over the attributes C that consists of the set of tuples from R that match the combination of every tuple in S.

Relational Algebra: Division Operation

- **Division** Operation: **R ÷ S**



(j) Divis on (shaded area)

V

A	B
a	1
a	2
b	1
b	2
c	1

W

B
1
2

V ÷ W

A
a
b

Example of division

Relational Algebra: Division Operation

- Example – **Division**
- Identify all clients who have viewed all properties with three rooms.
- **($\Pi_{clientNo, propertyNo}$(Viewing)) ÷ ($\Pi_{propertyNo}$($\sigma_{rooms = 3}$ (PropertyForRent)))**

$\Pi_{clientNo,propertyNo}$(Viewing)

clientNo	propertyNo
CR56	PA14
CR76	PG4
CR56	PG4
CR62	PA14
CR56	PG36

$\Pi_{propertyNo}(\sigma_{rooms=3}$(PropertyForRent))

propertyNo
PG4
PG36

RESULT

clientNo
CR56

12 Result of the Division operation on the Viewing and PropertyForRent relations.

Relational Algebra: Aggregate Operations

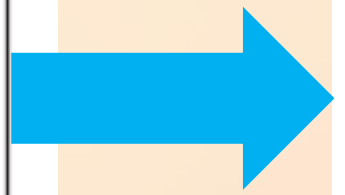
- **Aggregate: $\mathfrak{I}_{AL}(R)$**
  - Applies aggregate function list, AL, to R to define a relation over the aggregate list.
  - **AL contains one or more (<aggregate_function>, <attribute>) pairs .**
- Main aggregate functions are:
  - COUNT – returns the number of values in the associated attribute.
  - SUM – returns the sum of the values in the associated attribute.
  - AVG – returns the average of the values in the associated attribute.
  - MIN – returns the smallest value in the associated attribute.
  - MAX – returns the largest value in the associated attribute.

Relational Algebra: Aggregate Operations

- Example – **Aggregate: $\Im_{AL}(R)$**
- How many properties cost more than £350 per month to rent?
- ρ_R(**myCount**) $\Im_{\text{COUNT propertyNo}}$ ($\sigma_{\text{rent} > 350}$ (**PropertyForRent**))

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



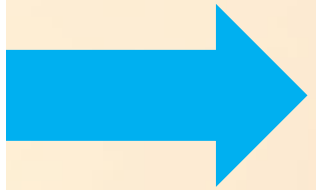
myCount
5

Relational Algebra: Aggregate Operations

- Example – **Aggregate: $\Im_{AL}(R)$**
- Find the minimum, maximum, and average staff salary.
- ρ_R(**myMin, myMax, myAverage**) $\Im_{\text{MIN salary MAX salary AVERAGE salary}}$ (**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



myMin	myMax	myAverage
9000	30000	17000

Relational Algebra: Grouping Operation

- **Grouping Operation:** $_{GA}\mathfrak{I}_{AL}(R)$
  - Groups tuples of R by **grouping attributes, GA**, and **then applies aggregate function list, AL, to define a new relation.**
  - **AL contains one or more (<aggregate_function>, <attribute>) pairs.**
  - Resulting relation contains the **grouping attributes, GA, along with results of each of the aggregate functions for each group defined by the combination of grouping attributes.**

Relational Algebra: Aggregate Operations

- Example – **Grouping:** $_{GA}\Im_{AL}(R)$
- Find the number of staff working in each branch and the sum of their salaries.
- $\rho_R(\text{branchNo}, \text{myCount}, \text{mySum})$ $_{\text{branchNo}}\Im$ COUNT staffNo, SUM salary (Staff)
- We first need to group tuples according to the branch number, and then use the aggregate functions COUNT and SUM to produce the required relation. The relational algebra expression is as follows:

Relational Algebra: Aggregate Operations

- Find the number of staff working in each branch and the sum of their salaries.
- ρ_R(**branchNo**, **myCount**, **mySum**) **branchNo** $\Im$ **COUNT staffNo**, **SUM salary** (**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

branchNo	myCount	mySum
B003	3	54000
B005	2	39000
B007	1	9000

Conclusion

- Relational model
  - Relation, Attribute, Domain, Tuple, Degree of a relation, and Cardinality of a relation
  - Relational schema and properties of relations
  - Keys: superkey, candidate key, primary key and foreign keys
  - Entity integrity and referential integrity constraints
- Relational algebra
  - Idea of relational algebra and relational completeness
  - Unary Operations: Selection and Projection
  - Set operations: Union, Set Difference, Intersection, and Cartesian Product
  - Join Operations: Theta join, Equijoin, Natural join, Outer join, and Semijoin
  - Division Operation
  - Aggregate and Grouping Operations

EXERCISES – DREAMHOME RENTAL DATABASE

- Exercise 1: **Select** list of **properties for rent**, which has **more than four rooms**.
- **Answer:** $\sigma_{rooms > 4}$ **(PropertyForRent)**

PropertyForRent

propertyNo	street	city	postcode	type	rooms	rent	ownerNo	staffNo	branchNo
PA14	16 Holhead	Aberdeen	AB7 5SU	House	6	650	CO46	SA9	B007
PG21	18 Dale Rd	Glasgow	G12	House	5	600	CO87	SG37	B003

EXERCISES – DREAMHOME RENTAL DATABASE

- Exercise 2: **Produce a list of email addresses** for **Clients**, showing only **fName, lName, and eMail**
- **Answer:** $\Pi_{\text{fName, lName, eMail}}(\text{Client})$

Client

clientNo	fName	lName	telephone	prefType	maxRent	eMail
C[redacted]6	John	Kay	02[redacted]774-5632	F[redacted]	4[redacted]	john.kay@gmail.com
C[redacted]6	Aline	Stewart	01[redacted]848-1825	F[redacted]	3[redacted]	astewart@hotmail.com
C[redacted]4	Mike	Ritchie	01[redacted]-392178	H[redacted]se	7[redacted]	mritchie01@yahoo.co.uk
C[redacted]2	Mary	Tregear	01[redacted]-196720	F[redacted]	6[redacted]	maryt@hotmail.co.uk

EXERCISES – DREAMHOME RENTAL DATABASE

- Exercise 3: Produce a list of **email addresses** for all outside users (**Clients and Prvate Owners**) showing only **fName, lName, and eMail**
- **Answer: (Π fName, lName, eMail(Client)) ∪ (Π fName, lName, eMail(PrivateOwner))**

Client

c[redacted]tNo	fName	lName	tel[redacted]	pr[redacted]ype	m[redacted]Rent	eMail
C[redacted]	John	Kay	020[redacted]74-5632	Fl[redacted]	4[redacted]	john.kay@gmail.com
						astewart@hotmail.com
						mritchie01@yahoo.co.uk
						maryt@hotmail.co.uk

PrivateOwner

ow[redacted]No	fName	lName	ad[redacted]ss	t[redacted]o	eMail	p[redacted]word
CO[redacted]	Joe	Keogh	2 H[redacted]us Dr, Aberdeen AB2 7SX	0[redacted]4-861212	jkeogh@lhh.com	** *****
CO[redacted]	Carol	Farrel	6 A[redacted]ay St, Glasgow G32 9DX	0[redacted]-357-7419	cfarrel@gmail.com	** *****
CO[redacted]	Tina	Murphy	63[redacted]l St, Glasgow G42	0[redacted]-943-1728	tinam@hotmail.com	** *****
CO[redacted]	Tony	Shaw	12[redacted]k Pl, Glasgow G4 0QR	0[redacted]-225-7025	tony.shaw@ark.com	** *****

EXERCISES – DREAMHOME RENTAL DATABASE

- Exercise 4: Produce a **list of clients (clientNos)** who viewed at least one property.
- **Answer:** $(\Pi_{\text{clientNo}}(\text{Client})) \cap (\Pi_{\text{clientNo}}(\text{Viewing}))$

Viewing

clientNo	prop yNo	vie ate	com ent
CR56	PA14	24- -13	too ll
CR76	PG4	20- -13	too ote
CR56	PG4	26- -13	
CR62	PA14	14- -13	no ng room
CR56	PG36	28- -13	

Client

clientNo	f me	IN e	telN	pre pe	m Rent	e l
CR76	J	Ka	0207 -5632	Flat	42	jo kay@gmail.com
CR56	A e	St rt	0141 -1825	Flat	35	as art@hotmail.com
~~CR74~~	M e	Ri e	0147 92178	Hou	75	m hie01@yahoo.co.uk
CR62	M y	Tr r	0122 96720	Flat	60	m @hotmail.co.uk

EXERCISES – DREAMHOME RENTAL DATABASE

- Exercise 5: Produce a **list of clients (clientNos)** who has not viewed any property.
- **Answer: ($\Pi_{clientNo}$(Client)) - ($\Pi_{clientNo}$(Viewing))**
  - Note: the order matters

Viewing

clientNo	prop[redacted]yNo	vie[redacted]ate	com[redacted]ent
CR56	PA14	24-[redacted]-13	too [redacted]ll
CR76	PG4	20-[redacted]-13	too [redacted]ote
CR56	PG4	26-[redacted]-13	
CR62	PA14	14-[redacted]-13	no [redacted]ng room
CR56	PG36	28-[redacted]-13	

Client

clientNo	f[redacted]me	IN[redacted]e	telN[redacted]	pre[redacted]pe	m[redacted]Rent	e[redacted]l
~~CR76~~	J	Ka	0207 [redacted]4-5632	Flat	42	jo[redacted]kay@gmail.com
~~CR56~~	A[redacted]e	St[redacted]rt	0141 [redacted]3-1825	Flat	35	as[redacted]art@hotmail.com
CR74	M[redacted]e	Ri[redacted]e	0147 [redacted]92178	Hou	75	m[redacted]hie01@yahoo.co.uk
~~CR62~~	M[redacted]y	Tr[redacted]r	0122 [redacted]96720	Flat	60	m[redacted]@hotmail.co.uk

EXERCISES – DREAMHOME RENTAL DATABASE

- Exercise 6: For each client list **first name, last name, branch (branch number) client is registered at and the date joined**.
- **Answer: Π fName, lName, branchNo, dateJoined (σ Client.clientNo = Registrtion.clientNo (Client X Registration))**

Client

clientNo	fName	lName	telNo	prefType	maxRent	eMail
C[illegible]6	John	Kay	02[illegible]774-5632	F[illegible]	4[illegible]	jo[illegible]kay@gmail.com
C[illegible]6	Aline	Stewart	01[illegible]848-1825	F[illegible]	3[illegible]	a[illegible]art@hotmail.com
C[illegible]4	Mike	Ritchie	01[illegible]-392178	H[illegible]se	7[illegible]	m[illegible]hie01@yahoo.co.uk
C[illegible]2	Mary	Tregear	01[illegible]-196720	F[illegible]	6[illegible]	m[illegible]t@hotmail.co.uk

Registration

clientNo	branchNo	staffNo	dateJoined
CR[illegible]	B005	S[illegible]	2-Jan-13
CR[illegible]	B003	S[illegible]7	11-Apr-12
CR[illegible]	B003	S[illegible]7	16-Nov-11
CR[illegible]	B007	S[illegible]	7-Mar-12

Take Cartesian Product and Select tuples with Client.clientNo = Registrtion.clientNo

EXERCISES – DREAMHOME RENTAL DATABASE

- Exercise 7: For each client list **first name, last name, branch (branch number) client is registered at and the date joined**.
- **Answer:** $\Pi_{\text{fName, lName, branchNo, dateJoined}}$ ((**Client** $\bowtie_{\text{Client.clientNo = Registrtion.clientNo}}$ **Registration**))

Client

clientNo	fName	lName	telNo	prefType	maxRent	email
CR76	John	Kay	0[redacted]774-5632	[redacted]	4[redacted]	j[redacted]kay@gmail.com
CR56	Aline	Stewart	0[redacted]848-1825	[redacted]	3[redacted]	a[redacted]art@hotmail.com
CR74	Mike	Ritchie	0[redacted]-392178	House	7[redacted]	m[redacted]hie01@yahoo.co.uk
CR62	Mary	Tregear	0[redacted]-196720	[redacted]	6[redacted]	m[redacted]t@hotmail.co.uk

Registration

clientNo	branchNo	staffNo	dateJoined
CR[redacted]	B005	S[redacted]	2-Jan-13
CR[redacted]	B003	S[redacted]7	11-Apr-12
CR[redacted]	B003	S[redacted]7	16-Nov-11
CR[redacted]	B007	S[redacted]	7-Mar-12

Take **Theta Join** with **Predicate Client.clientNo = Registrtion.clientNo**

This is an Equijoin

EXERCISES – DREAMHOME RENTAL DATABASE

- Exercise 8: For each client list **first name, last name, branch (branch number) client is registered at and the date joined** – use **Natural Join**
- **Answer:** $\Pi_{\text{fName, lName, branchNo, dateJoined}}$ ((**Client** $\bowtie$ **Registration**))

Client

clientNo	fName	lName	telNo	pType	rent	email
C[redacted]6	John	Kay	02[redacted]774-5632	F[redacted]	4[redacted]	jo[redacted]kay@gmail.com
C[redacted]6	Aline	Stewart	01[redacted]848-1825	F[redacted]	3[redacted]	a[redacted]art@hotmail.com
C[redacted]4	Mike	Ritchie	01[redacted]-392178	H[redacted]se	7[redacted]	m[redacted]hie01@yahoo.co.uk
C[redacted]2	Mary	Tregear	01[redacted]-196720	F[redacted]	6[redacted]	m[redacted]t@hotmail.co.uk

Registration

clientNo	branchNo	staffNo	dateJoined
CR[redacted]	B005	S[redacted]	2-Jan-13
CR[redacted]	B003	S[redacted]7	11-Apr-12
CR[redacted]	B003	S[redacted]7	16-Nov-11
CR[redacted]	B007	S[redacted]	7-Mar-12

Take Natural Join of Client and Registration

EXERCISES – DREAMHOME RENTAL DATABASE

- Exercise 9: **For each property**, list **property number, city, type, and name (first and last names) of the employee handing the property.**
- **Answer:** $\Pi_{\text{propertyNo, city, type, fName, lName,}}$ (**PropertyForRent** $\bowtie_{\text{PropertyForRent.staffNo = Staff.staffNo}}$ **Staff**)

PropertyForRent

propertyNo	stre	city	p	code	type	r	s	ov	No	st	lo	br	n
PA14	16 H	ad	Aberdeen	A	SU	House	6		C		S		B
PL94	6 Ar	t	London	N		Flat	4		C		S		B
PG4	6 La	ce St	Glasgow	G	QX	Flat	3		C				B
PG36	2 M	Rd	Glasgow	G	QX	Flat	3		C		S		B
PG21	18 D	Rd	Glasgow	G		House	5		C		S		B
PG16	5 Nc	Dr	Glasgow	G	AX	Flat	4		C		S		B

Staff

s	No	fName	lName	p	on		DO		y	bra	No
S		Mary	Howe	A	nt		19-	70		B0	
S		Julie	Lee	A	nt		13-	65		B0	
S		Ann	Beech	A	nt		10-	-60	0	B0	
S		Ann	Beech	A	nt		10-	-60	0	B0	
S		David	Ford	S	visor		24-	-58	0	B0	

Take Left Outer Join of PropertyForRent and Staff on staffNo

EXERCISES – DREAMHOME RENTAL DATABASE

- Exercise 10: For each city (that has a property for rent), list the number of properties available and their average rent.
- **Answer: ρ_R(city, count, avg rent) $_{city}\Im_{COUNT\ propertyNo,\ AVG\ rent}$ (PropertyForRent)**

PropertyForRent

propertyNo	street	city	postcode	type	rooms	rent	ownerN		
PA14	16 Holhead	Aberdeen	AB7 5SU	House	6	650	CO46		
PL94	6 Argyll St	London	NW2	Flat	4	400	CO87		
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	count	avg rent
Aberdeen	1	650
London	1	400
Glasgow	4	443.75

Practice Exercises Set 1 – No Submission Required

- Express relational algebra expressions to get following information from the DreamHome rental database.
  - Telephone number (first name, last name and telephone number) of all external users (clients and private owners)
  - First and last names of all the users (staff, clients and private owners).
  - List all the properties (propertyNo) that have been viewed at least once.
  - List all the properties (propertyNo) that have never been viewed by a client yet.
  - List the staff members (staffNo) who have registered at least one client.
  - List the staff members (staffNo) who have not registered any clients.
  - For each staff member, list the first name, last name and branch city.
  - For each property, list the Property number, address (street, city and postcode), and name (first and last names) of the owner.

Practice Exercises Set 2 – No Submission Required

- Express relational algebra expressions to get following information from the DreamHome rental database.
  - For each staff member, list the first name, last name and branch city.
  - For each property, list the Property number, address (street, city and postcode), and name (first and last names) of the owner.
  - For each client, list the first name, last name, telephone number and the date joined.
  - For each property, list the property number, city, type, and owners name (first and last).
  - List the property details (all) of the properties owned by Tony Shaw.
  - List branch number, street, city and first and last names of employees who work in branch.
    - Branch information may repeat.
    - **All the branches, even the ones with no employees must be listed.**

Practice Exercises Set 3 – No Submission Required

- Express relational algebra expressions to get following information from the DreamHome rental database.
  - What is the name of the employee who get highest salary?
  - What is the average rent for a house?
  - What is the average rent for a 3-bedroom flat?
  - For each property type, list the average, minimum, and maximum rent.
  - For each client, list the name and number of properties the client has viewed.
  - For each city where there is a branch, list the number of employees working in that city and the average salary of employees working in the city.