RELATIONAL DATA MODEL

EGCO321 DATABASE SYSTEMS

KANAT POOLSAWASD
DEPARTMENT OF COMPUTER ENGINEERING
MAHIDOL UNIVERSITY

DATABASE MODEL

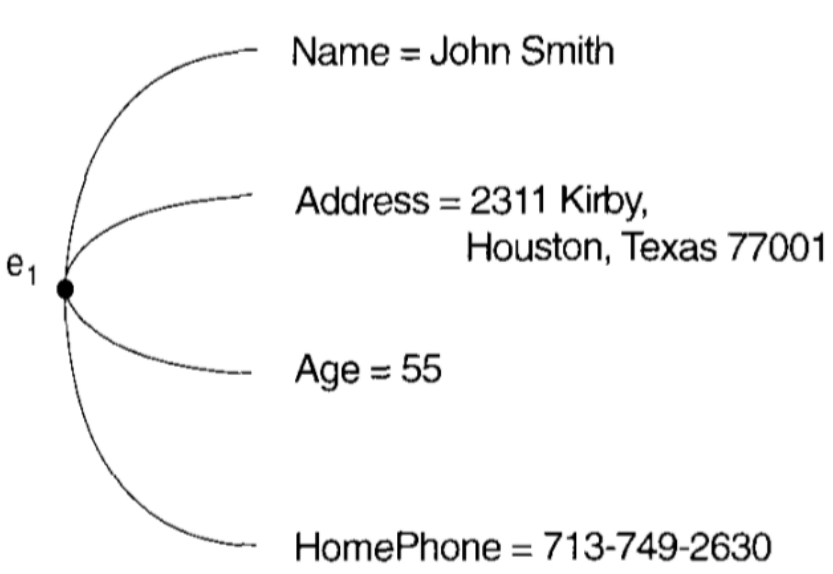
- Collection of logical constructs used to represent data structure and relationships within the database
- Conceptual models: logical nature of data representation
- Implementation models: emphasis on how the data are represented in the database

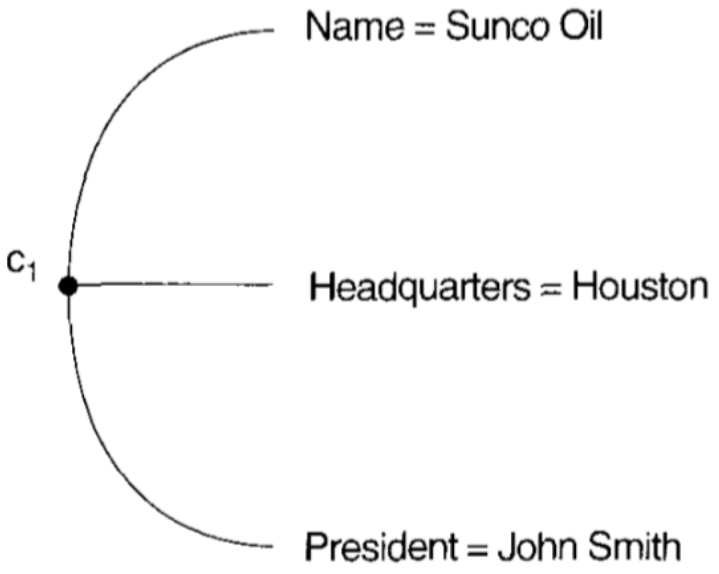
ENTITIES AND ATTRIBUTES (1)

- The basic object that the ER Model represents is an **entity**, which is a **thing** in the real world with an independent existence.
- Each entity has **attributes** — the particular properties that describe it.

ENTITIES AND ATTRIBUTES (2)

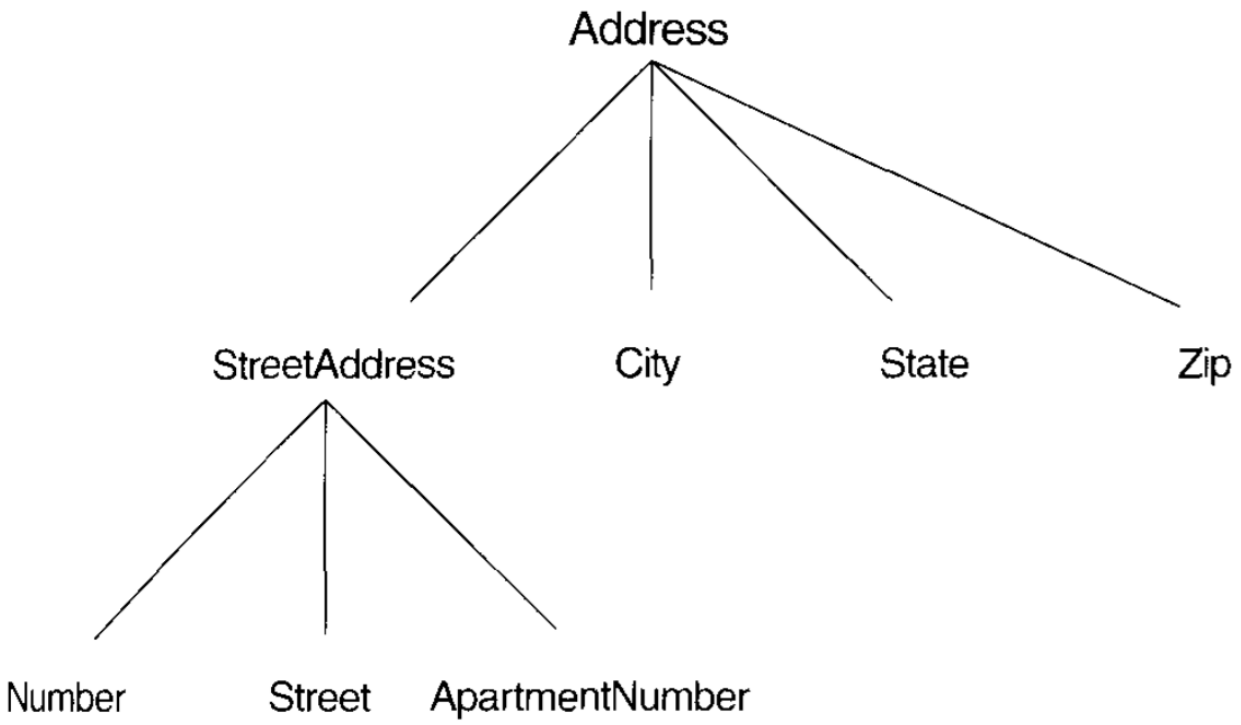
- Example: Two entities, employee e_1 and company c_1, and their attributes.





ENTITIES AND ATTRIBUTES (3)

- Composite and Simple (Atomic) Attributes.

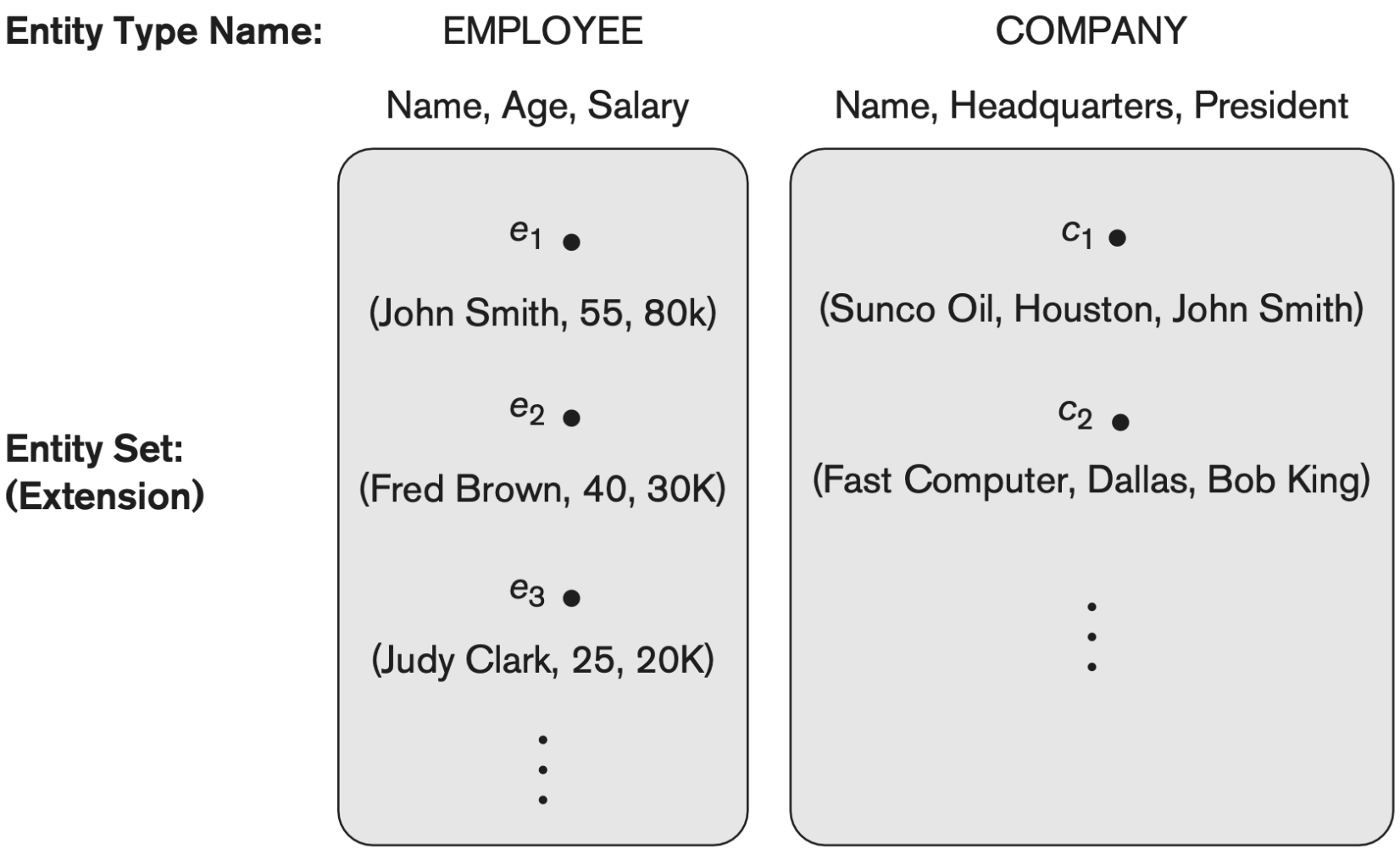


ENTITY TYPES AND ENTITY SETS (1)

- A database usually contains groups of entities that are similar.
  - For example, a company employing hundreds of employees may want to store similar information concerning each of the employees. These employee entities share the same attributes, but each entity has its own value(s) for each attribute.
- An entity type defines a collection (or set) of entities that have the same attributes.
- The entity set is usually referred to using the same name as the entity type.

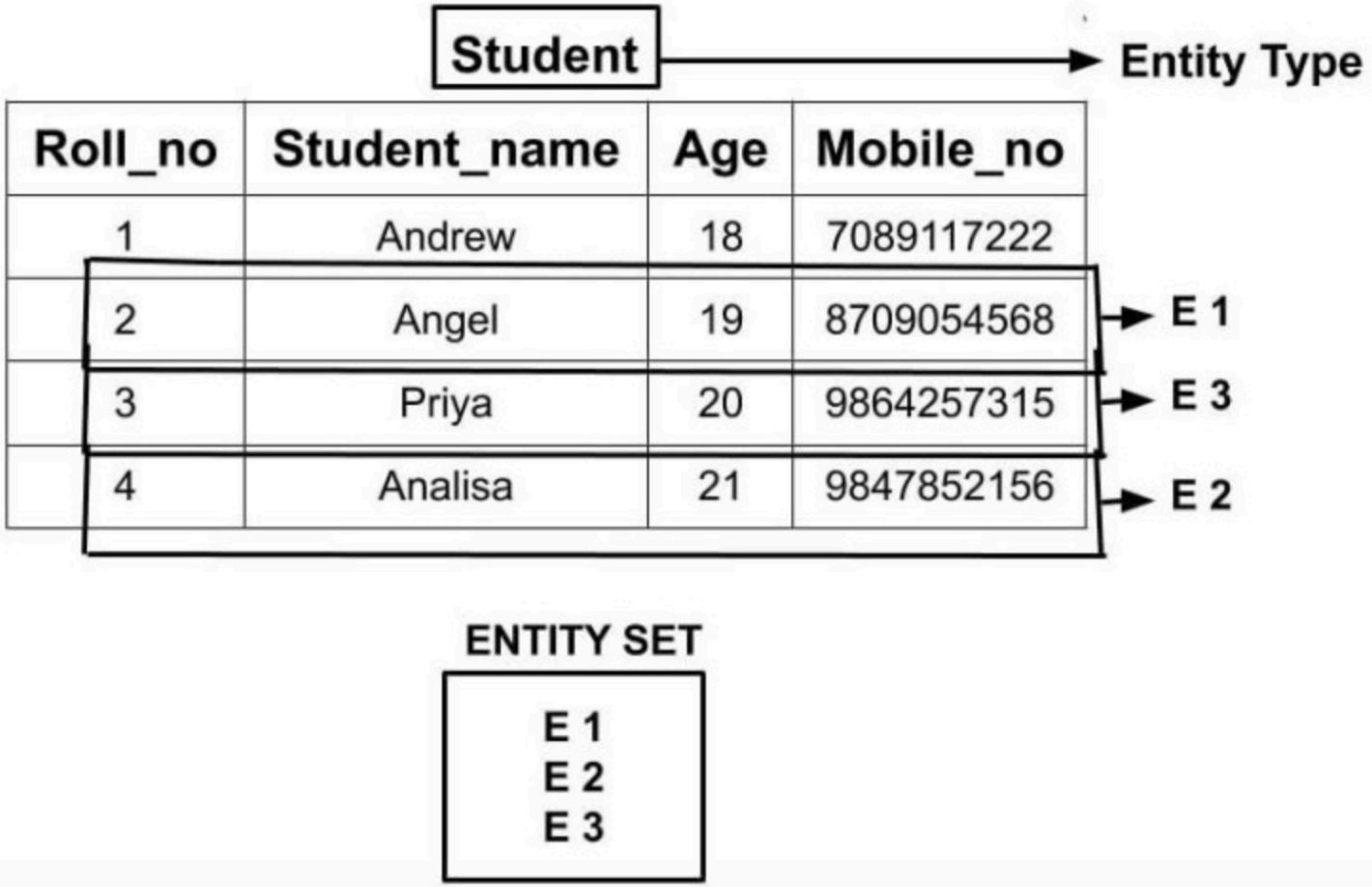
ENTITY TYPES AND ENTITY SETS (2)

- Two entity types, employee and company.



ENTITY TYPES AND ENTITY SETS (3)

- Example of entity type and entity set



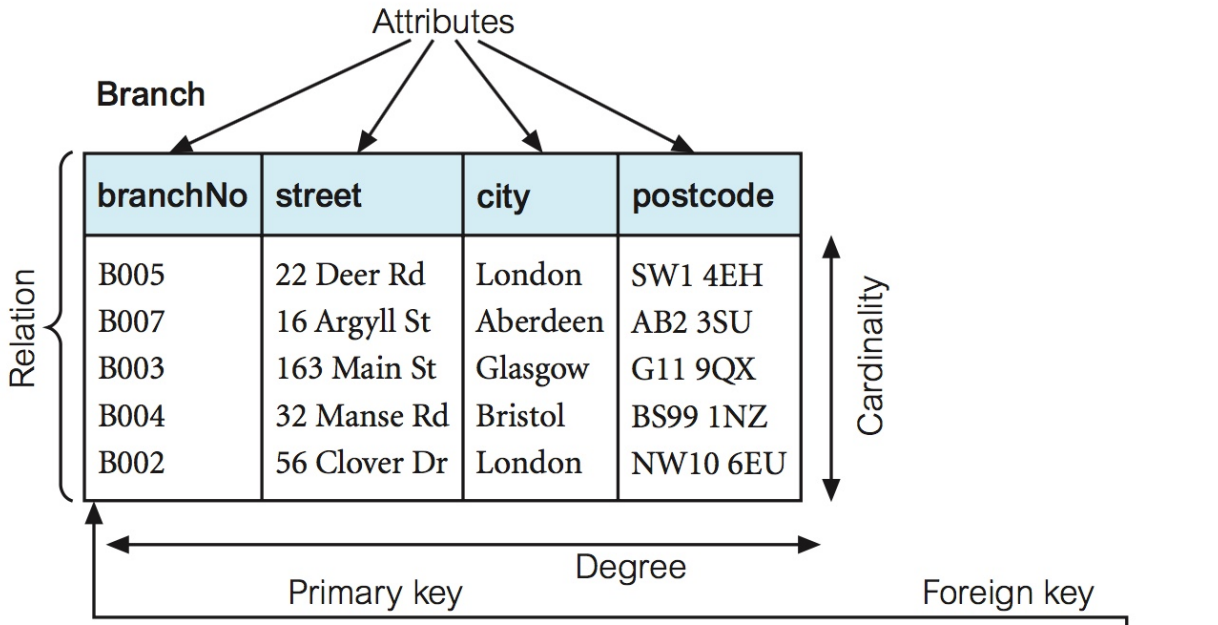
ENTITY TYPES AND ENTITY SETS (4)

- Value Sets (Domains) of Attributes.
  - Each simple attribute of an entity type is associated with a **value set** (or **domain** of values)
  - In previous figure (previous page), if the range of ages allowed for employees is between 16 to 70, we can specify the value set of the "Age" attribute of EMPLOYEE to be the set of integer number between 16 to 70.
  - Value sets are typically specified using the basic **data types** available in most programming languages, such as integer, string, or boolean.

RELATIONAL DATA STRUCTURE (1)

- **Relation:** A relation is a table with columns and rows.
- **Attribute:** An attribute is a named column of a relation.
- **Domain:** A domain is the set of allowable values for one or more attributes.
- **Tuple:** A tuple is a row of a relation.
- **Cardinality:** The cardinality of a relation is the number of tuples it contains.
- **Relational Database:** A collection of normalized relations with distinct relation names.

RELATIONAL DATA STRUCTURE (2)



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

RELATIONAL DATA STRUCTURE (3)

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

ALTERNATIVE TERMINOLOGY

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

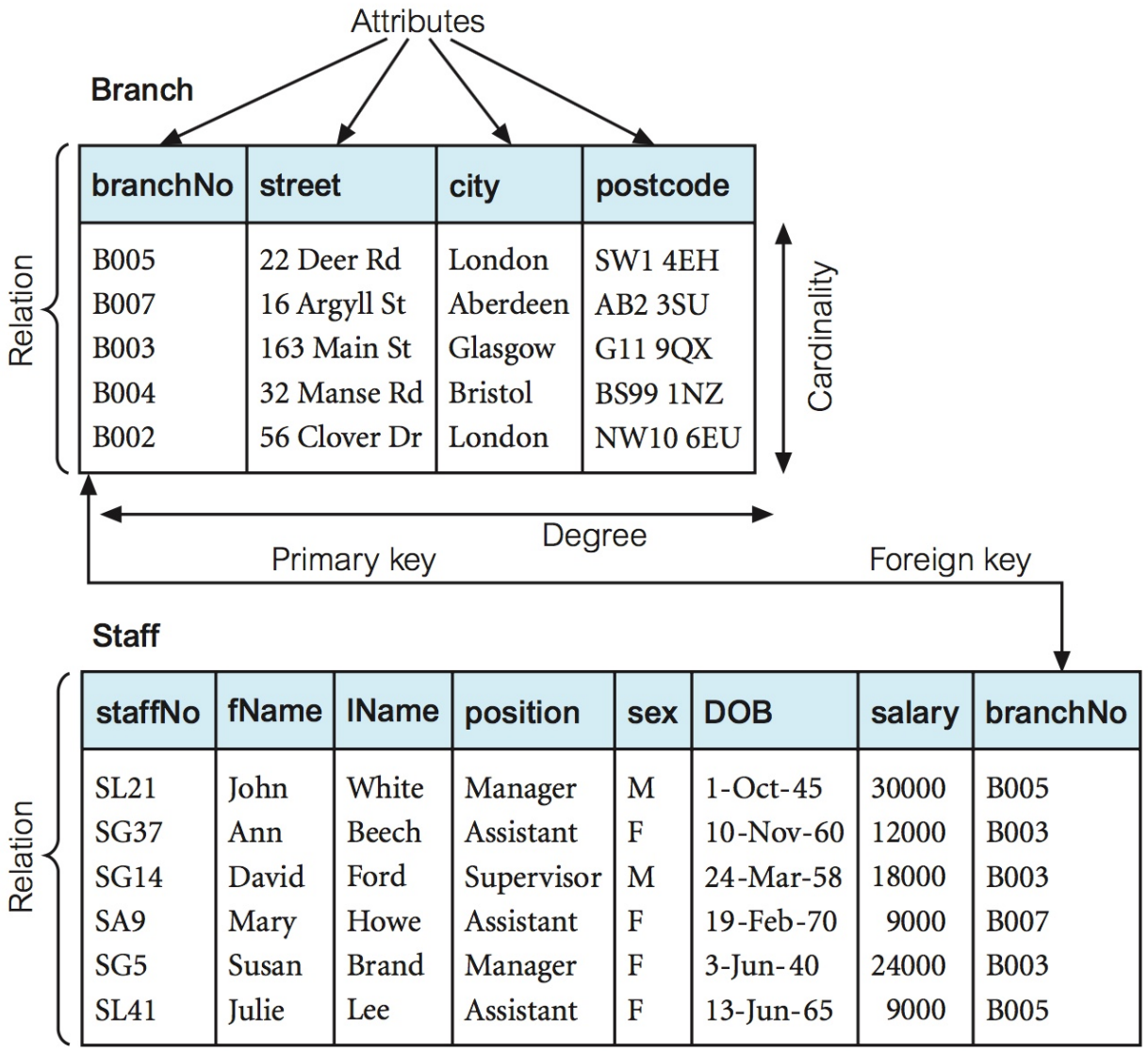
DATABASE RELATIONS

- Relational Schema: A named relation defined by a set of attribute and domain name pairs schema
- Relational Database Schema: A set of relational schemas, each with a distinct name.

PROPERTIES OF RELATIONS (1)

- The relation has a name that is distinct from all other relation names in the relational schema.
- Each cell of the relation contains exactly one atomic (single) value; n each attribute has a distinct name.
- The values of an attribute are all from the same domain.
- Each tuple is distinct; there are no duplicate tuples.
- The order of attributes has no significance.
- The order of tuples has no significance, theoretically. (However, in practice, the order may affect the efficiency of accessing tuples.)

PROPERTIES OF RELATIONS (2)



RELATIONAL KEYS (1)

- **Candidate Key:** a minimal super key. A super key is minimal if removing any column makes it no longer unique.
- **Super Key:** a column or combination of columns containing unique value for each row.
- **Primary Key:** a specially designated candidate key. The primary key for a table cannot contain null value.
- **Alternate Key:** An alternate key is the secondary candidate key that contains all the property of a candidate key but is an alternate option (did not become a primary key)

RELATIONAL KEYS (2)

- **Composite Key:** A set of multiple attributes that help us uniquely identify every tuple present in a table.
- **Unique Key:** A column/a set of columns that identify every record uniquely in a table.
  - Note: A unique key is different from a primary key. It is because it is only capable of having one null value. A primary key, on the other hand, cannot have a null value.
- **Foreign Key:** a foreign key to establish relationships between two available tables. A foreign key must have the same data type as its associated candidate key

INTEGRITY RULES (1)

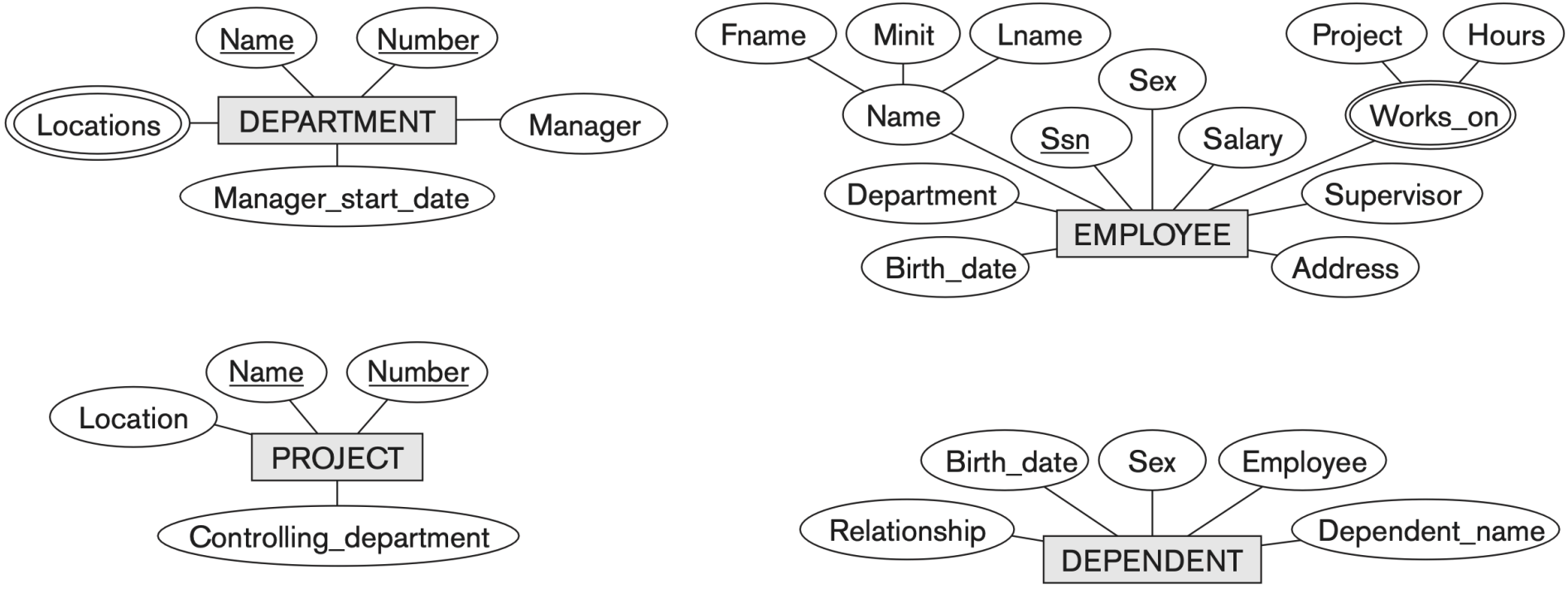
- **Entity Integrity** means that each table must have a column or combination of columns with unique values. Unique means that no two rows of a table have the same value.
- **Referential Integrity** means that the column values in one table must match column values in a related table.

INTEGRITY RULES (2)

- **Entity Integrity Rule:** No two rows of a table can contain the same value for the primary key. In addition, no row can contain a null value for any column of a primary key.
- **Referential Integrity Rule:** Only two kinds of values can be stored in a foreign key:
  - A value matching a candidate key value in some row of the table containing the associated candidate key.
  - A null value.

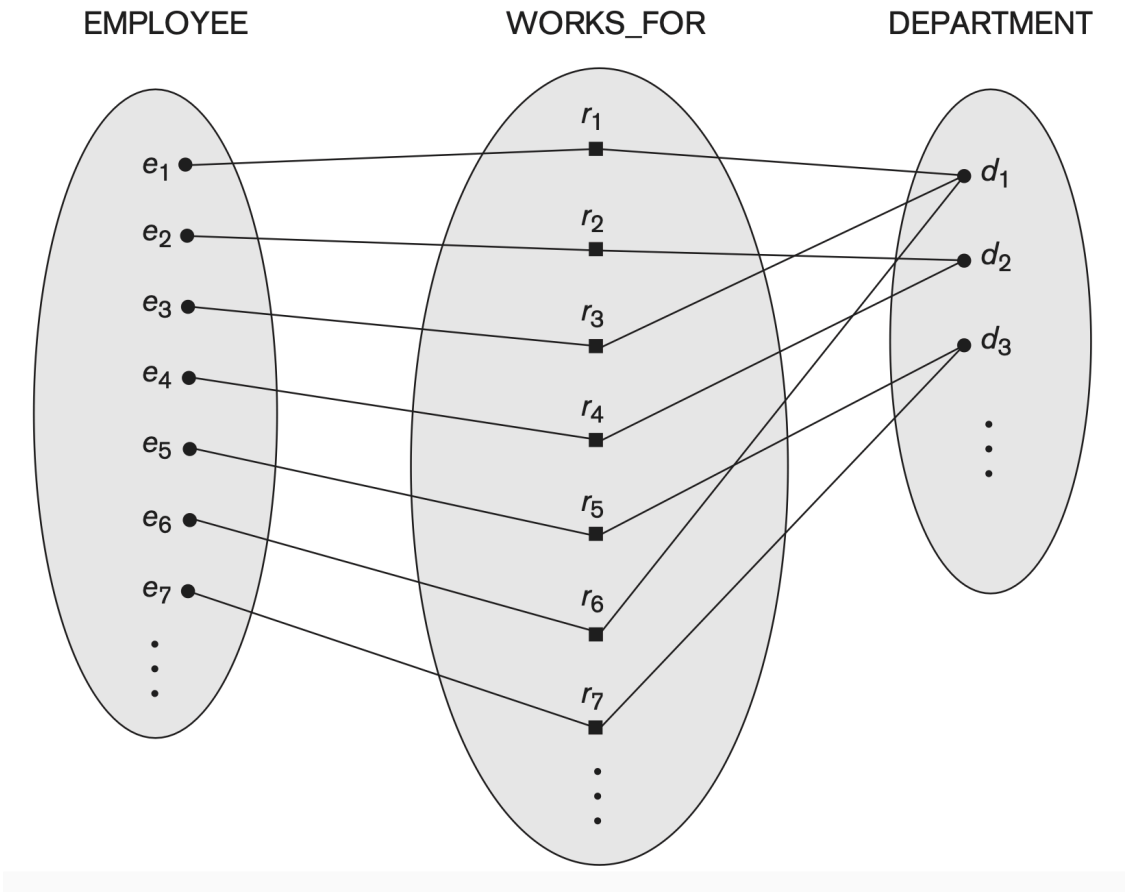
RELATIONSHIP (1)

- Design of the COMPANY database



RELATIONSHIP (2)

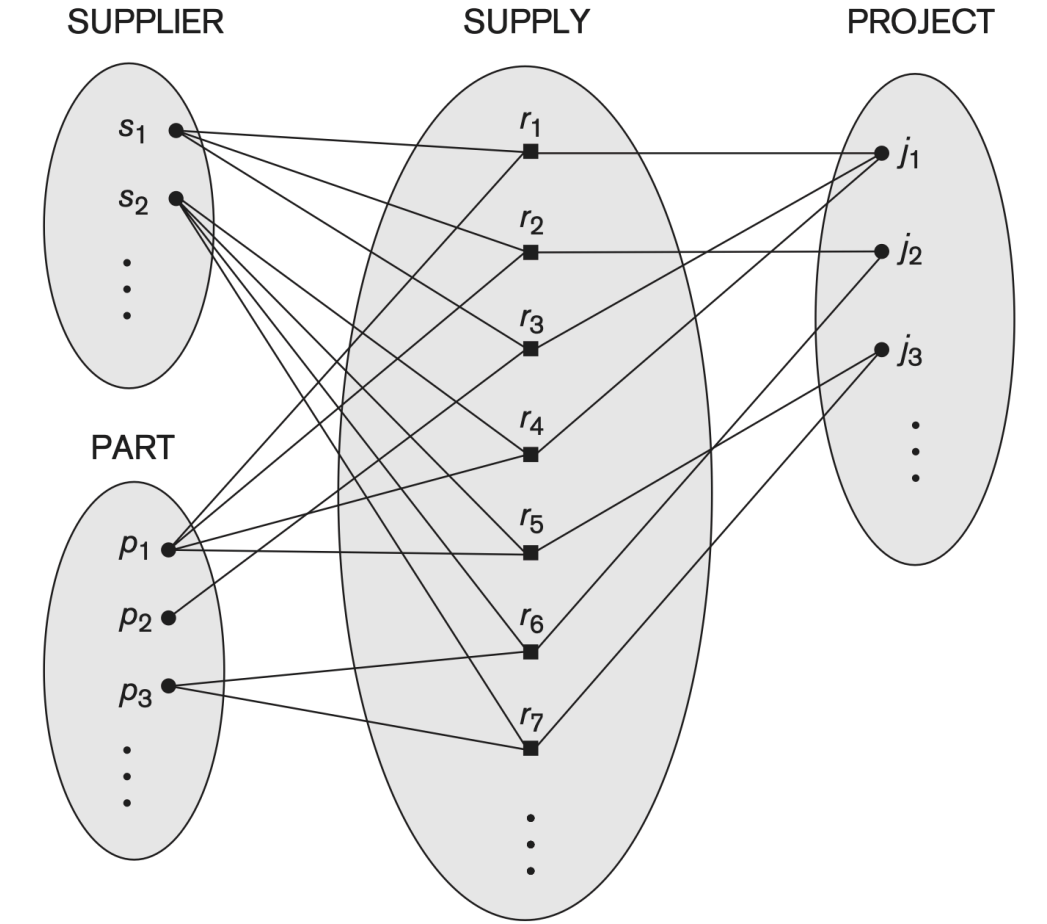
- Relationship Types and Relationship Sets.



Some instances in the WORKS_FOR relationship sets, which represents a relationship type WORKS_FOR between EMPLOYEE and DEPARTMENT

RELATIONSHIP (3)

- Relationship Degree, Role Names, and Recursive Relationships



Some relationship instances in the SUPPLY ternary relationship set.

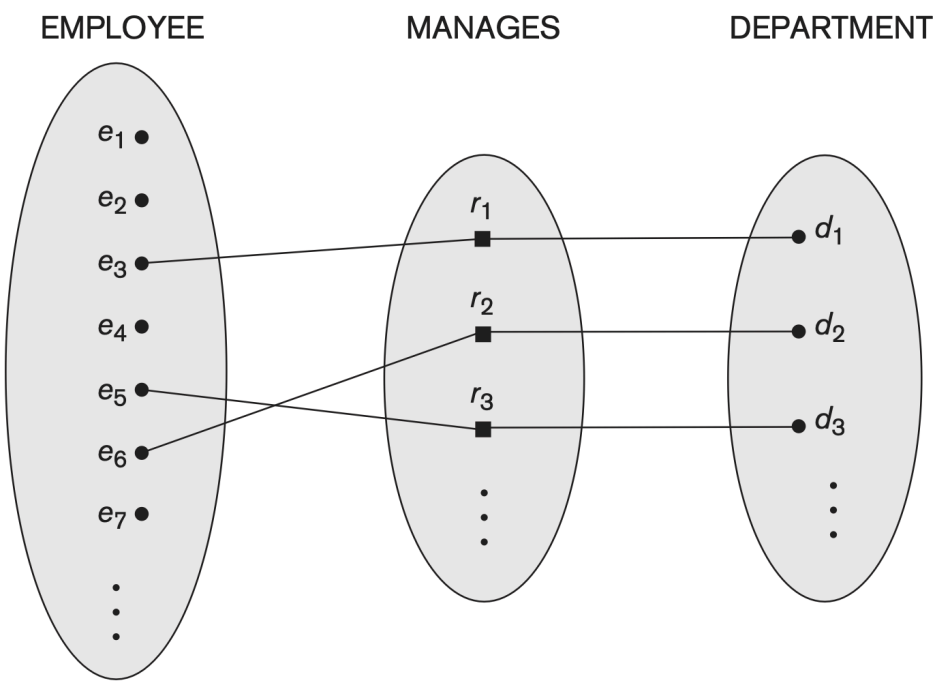
RELATIONSHIP (4)

- Constrains on Relationship Types
  - Relationship types usually have certain constraints that limit the possible combinations of entities that may participate in the corresponding relationship set.
  - We can distinguish two main types of relationship constraints:
    - Cardinality Ratio
    - Participation

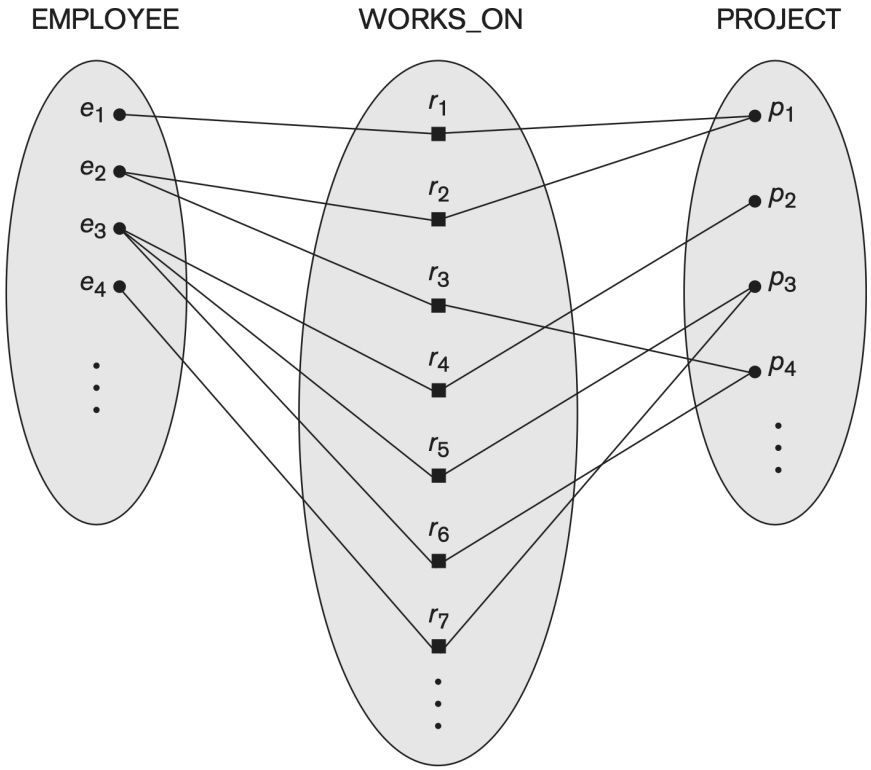
RELATIONSHIP (5)

- Cardinality Ratio for Binary Relationships
  - One-to-one (1:1)
  - One-to-many (1:M)
  - Many-to-many (M:N)

RELATIONSHIP (6)



A 1:1 relationship, MANAGES



A M:N relationship, WORKS_ON

RELATIONSHIP (7)

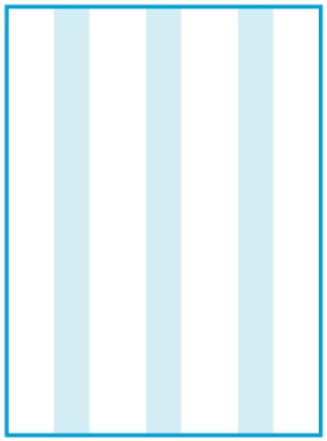
- Participation Constraints (Connectivity) specifies whether the existence of an entity depends on its being related to another entity via the relationship type.
- This constraint specifies the minimum number of relationship instances that each entity can participate in, sometime called the **minimum** cardinality constraint.

RELATIONSHIP (8)

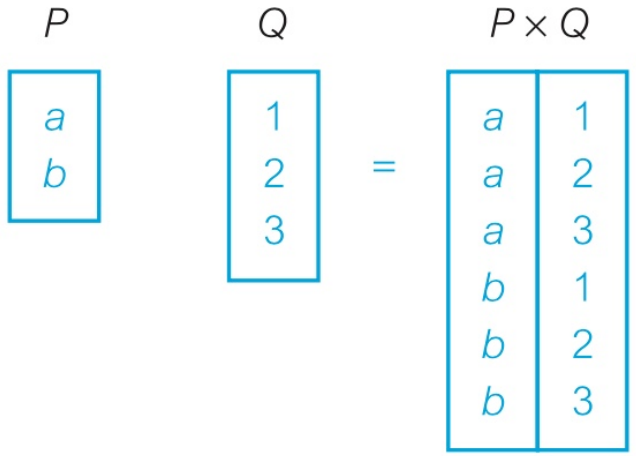
- Attributes of Relationship Types
  - Relationship types can also have attributes, similar to those of entity types.
  - For 1:1 or 1:N relationship types can be migrated to one of the participating entity types.
  - For M:N relationship types, some attributes may be determined by the combination of participating entities in a relationship instance, not by any single entity. Such attributes must be specified as relationship attributes.

RELATIONAL ALGEBRA (1)

(a) Selection

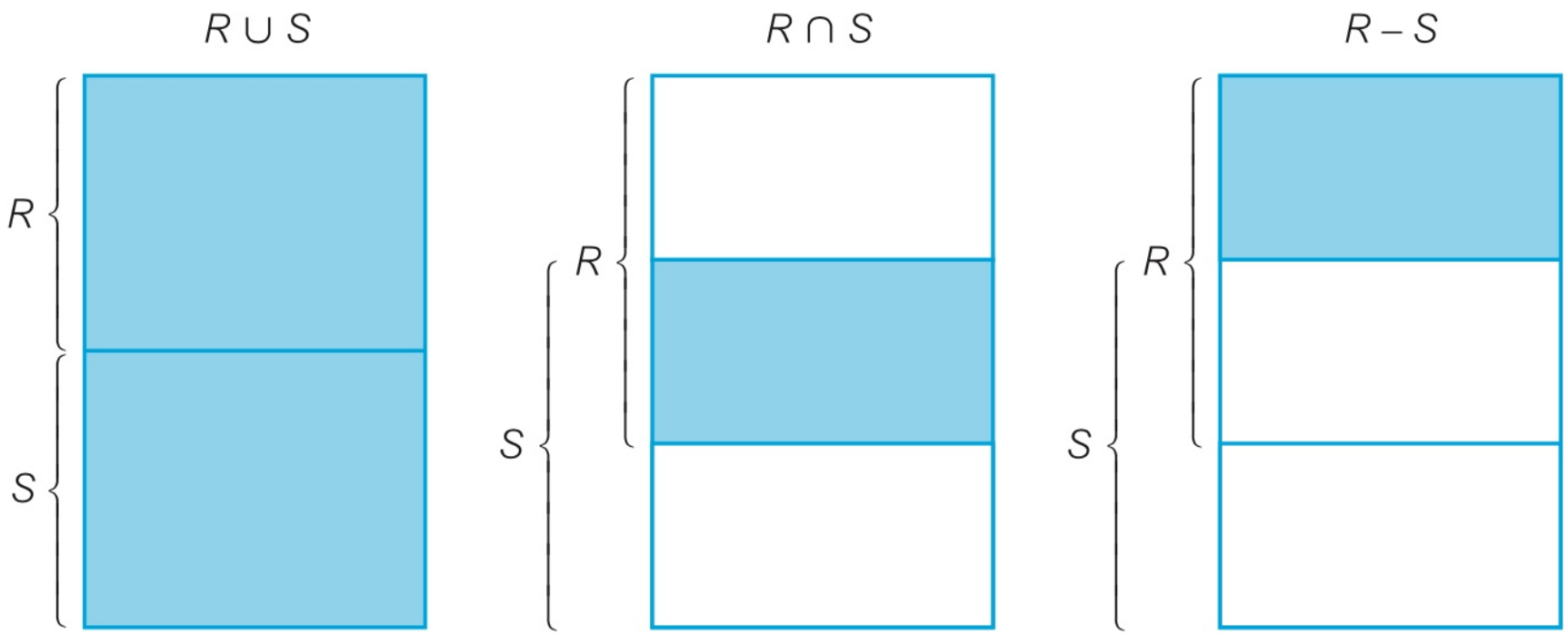


(b) Projection



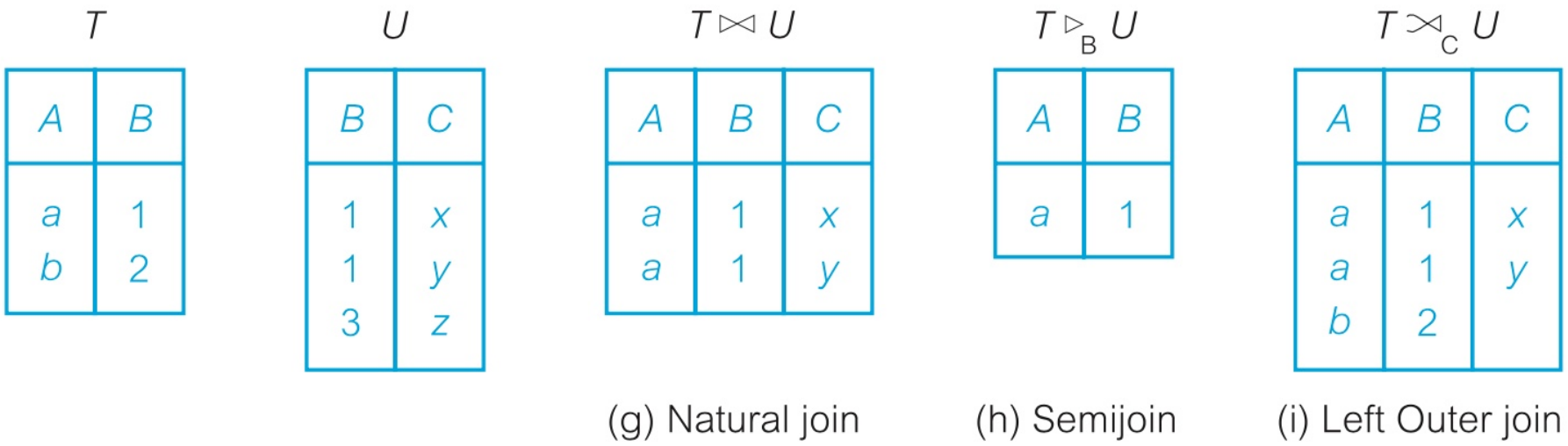
(c) Cartesian product

RELATIONAL ALGEBRA (2)



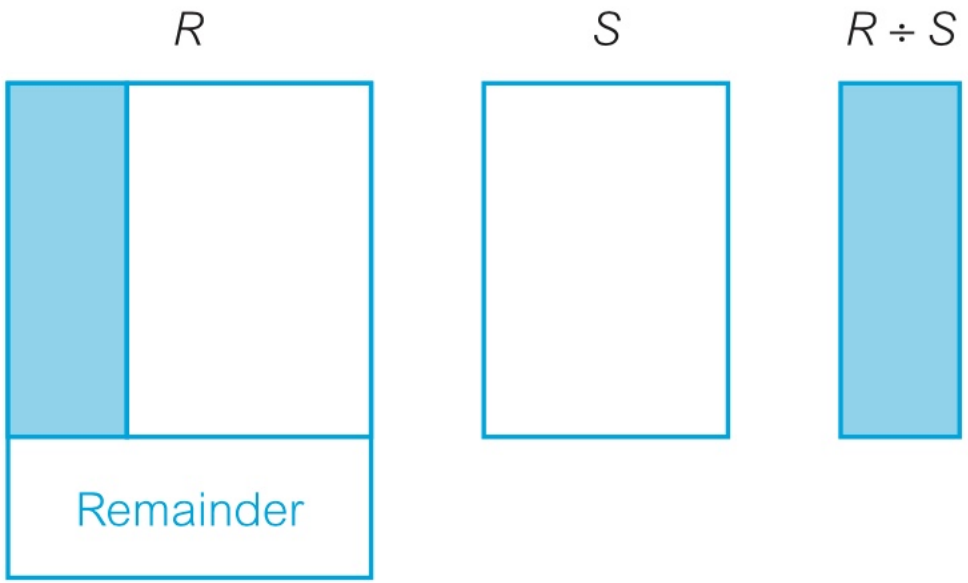
(d) Union (e) Intersection (f) Set difference

RELATIONAL ALGEBRA (3)



(g) Natural join (h) Semijoin (i) Left Outer join

RELATIONAL ALGEBRA (4)



(j) Division (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

SAMPLE DATA (1)

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

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

SAMPLE DATA (2)

Client

clientNo	fName	lName	telNo	prefType	maxRent
CR76	John	Kay	0207-774-5632	Flat	425
CR56	Aline	Stewart	0141-848-1825	Flat	350
CR74	Mike	Ritchie	01475-392178	House	750
CR62	Mary	Tregear	01224-196720	Flat	600

PrivateOwner

ownerNo	fName	lName	address	telNo
CO46	Joe	Keogh	2 Fergus Dr, Aberdeen AB2 7SX	01224-861212
CO87	Carol	Farrel	6 Achray St, Glasgow G32 9DX	0141-357-7419
CO40	Tina	Murphy	63 Well St, Glasgow G42	0141-943-1728
CO93	Tony	Shaw	12 Park Pl, Glasgow G4 0QR	0141-225-7025

SAMPLE DATA (3)

Viewing

clientNo	propertyNo	viewDate	comment
CR56	PA14	24-May-04	too small
CR76	PG4	20-Apr-04	too remote
CR56	PG4	26-May-04	
CR62	PA14	14-May-04	no dining room
CR56	PG36	28-Apr-04	

Registration

clientNo	branchNo	staffNo	dateJoined
CR76	B005	SL41	2-Jan-04
CR56	B003	SG37	11-Apr-03
CR74	B003	SG37	16-Nov-02
CR62	B007	SA9	7-Mar-03

EXAMPLE 1 (SELECTION)

- List all staff with a salary grater than 10,000

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 2 (PROJECTION)

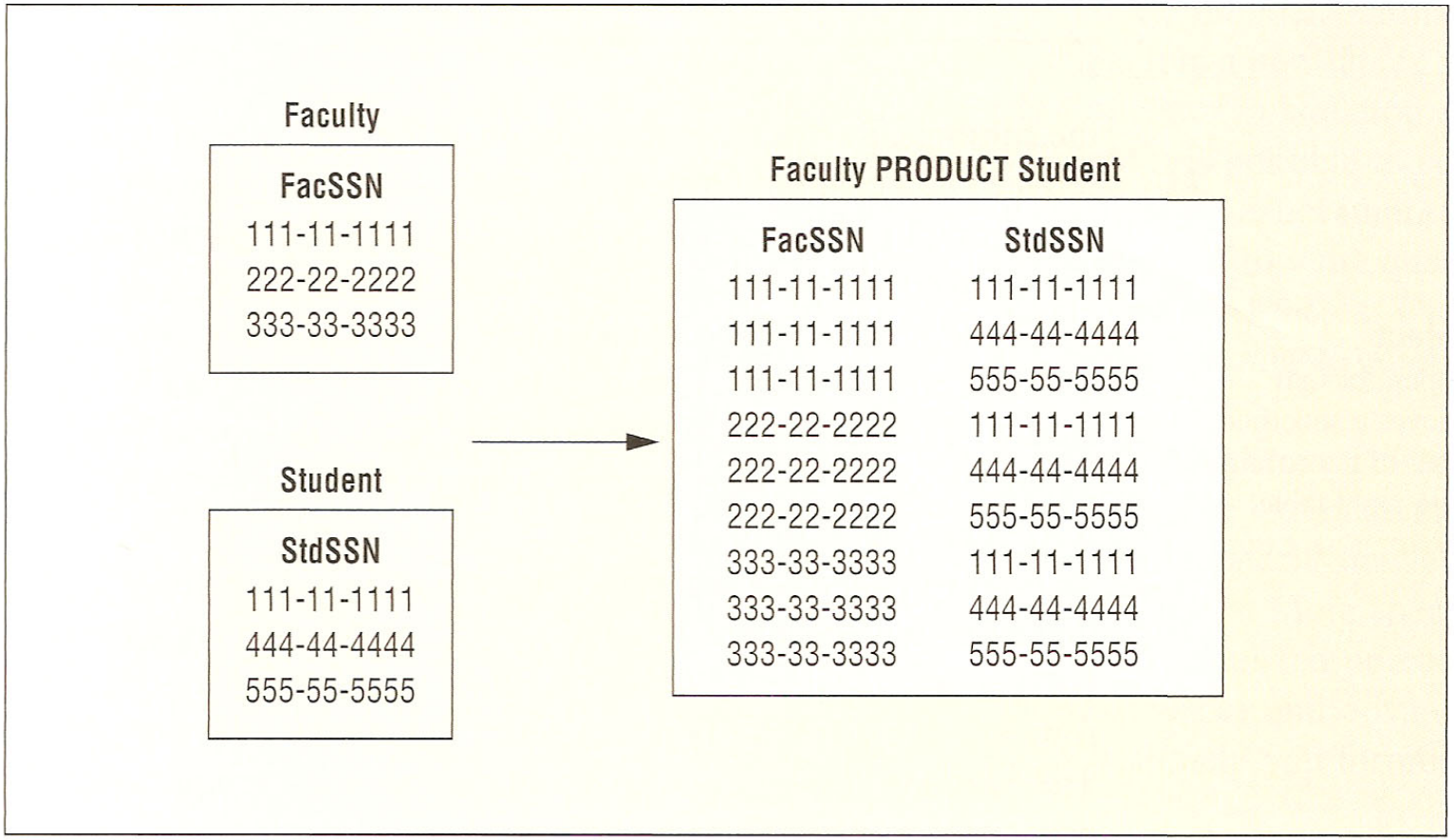
- Produce a list of salary for all staff, showing only the staffNo, fName, lName, and salary detail.

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 3: PRODUCT (1)



EXAMPLE 3: PRODUCT (2)

StdSSN	StdLastName	StdMajor	StdClass
123-45-6789	WELLS	IS	FR
124-56-7890	NORBERT	FIN	JR
234-56-7890	KENDALL	ACCT	JR

OfferNo	StdSSN	EnrGrade
1234	123-45-6789	3.3
1234	234-56-7890	3.5
4321	124-56-7890	3.2

Student **PRODUCT** *Enrollment*

Student.StdSSN	StdLastName	StdMajor	StdClass	OfferNo	Enrollment.StdSSN	EnrGrade
123-45-6789	WELLS	IS	FR	1234	123-45-6789	3.3
123-45-6789	WELLS	IS	FR	1234	234-56-7890	3.5
123-45-6789	WELLS	IS	FR	4321	124-56-7890	3.2
124-56-7890	NORBERT	FIN	JR	1234	123-45-6789	3.3
124-56-7890	NORBERT	FIN	JR	1234	234-56-7890	3.5
124-56-7890	NORBERT	FIN	JR	4321	124-56-7890	3.2
234-56-7890	KENDALL	ACCT	JR	1234	123-45-6789	3.3
234-56-7890	KENDALL	ACCT	JR	1234	234-56-7890	3.5
234-56-7890	KENDALL	ACCT	JR	4321	124-56-7890	3.2

JOIN OPERATIONS

- Typically, we want only combinations of the Cartesian product that satisfy certain conditions and so we would normally use a Join operation instead of the Cartesian product operation.
- There are mainly two types of joins
  - Inner join (Theta, Natural, EQUI)
  - Outer join (Left, Right, Full)

INNER JOIN

- Inner Join is used to return **rows from both tables** which satisfy the given condition.
- It is the most widely used join operation and can be considered as a default join-type
- An Inner join or equijoin is a comparator-based join which uses equality comparisons in the join-predicate. However, if you use other comparison operators like “>” it can’t be called equijoin.
- Inner Join further divided into three subtypes:
  - Theta join
  - Natural join
  - EQUI join

THETA JOIN (1)

- Theta Join allows you to merge two tables based on the condition represented by theta.
- Theta joins work for all comparison operators. It is denoted by symbol θ. The general case of JOIN operation is called a Theta join.
- Syntax:

$$A \bowtie_{\theta} B$$

THETA JOIN (2)

- Theta join can use any conditions in the selection criteria.
- Consider the following tables.

สินค้า

รหัสสินค้า	ชื่อ	ราคา
001	ปากกา	100
003	สมุด	50
005	ดินสอ	10

การสั่งซื้อ

เลขที่ใบสั่งซื้อ	รหัสลูกค้า	รหัสสินค้า
1	A001	003
2	A015	001

- สินค้า $\bowtie$ สินค้า.ชื่อ = "ปากกา" การสั่งซื้อ

สินค้า การสั่งซื้อ

เลขที่ใบสั่งซื้อ	รหัสลูกค้า	รหัสสินค้า	รหัสสินค้า	ชื่อสินค้า	ราคาสินค้า
1	A001	003	001	ปากกา	100
2	A015	001	001	ปากกา	100

EQUI JOIN

- EQUI Join is done when a Theta join uses only the **equivalence** condition.

สินค้า

รหัสสินค้า	ชื่อ	ราคา
001	ปากกา	100
003	สมุด	50
005	ดินสอ	10

การสั่งซื้อ

เลขที่ใบสั่งซื้อ	รหัสลูกค้า	รหัสสินค้า
1	A001	003
2	A015	001

- สินค้า ⋈ สินค้า.รหัสสินค้า = การสั่งซื้อ.รหัสสินค้า การสั่งซื้อ

สินค้า การสั่งซื้อ

เลขที่ใบสั่งซื้อ	รหัสลูกค้า	รหัสสินค้า	รหัสสินค้า	ชื่อสินค้า	ราคาสินค้า
1	A001	003	003	สมุด	50
2	A015	001	001	ปากกา	100

NATURAL JOIN (1)

- Natural Join does not utilize any of the comparison operators. In this type of join, the attributes should have the same name and domain. In Natural Join, there should be at least one common attribute between two relations.

NATURAL JOIN (2)

- Example

สินค้า

รหัสสินค้า	ชื่อ	ราคา
001	ปากกา	100
003	สมุด	50
005	ดินสอ	10

การสั่งซื้อ

เลขที่ใบสั่งซื้อ	รหัสลูกค้า	รหัสสินค้า
1	A001	003
2	A015	001

- Cartesian Product

สินค้า การสั่งซื้อ

เลขที่ใบสั่งซื้อ	รหัสลูกค้า	รหัสสินค้า	รหัสสินค้า	ชื่อสินค้า	ราคาสินค้า
1	A001	003	001	ปากกา	100
1	A001	003	003	สมุด	50
1	A001	003	005	ดินสอ	10
2	A015	001	001	ปากกา	100
2	A015	001	003	สมุด	50
2	A015	001	005	ดินสอ	10

NATURAL JOIN (3)

- Cartesian Product

สินค้า | การสั่งซื้อ

เลขที่ใบสั่งซื้อ	รหัสลูกค้า	รหัสสินค้า	รหัสสินค้า	ชื่อสินค้า	ราคาสินค้า
1	A001	003	001	ปากกา	100
1	A001	003	003	สมุด	50
1	A001	003	005	ดินสอ	10
2	A015	001	001	ปากกา	100
2	A015	001	003	สมุด	50
2	A015	001	005	ดินสอ	10

- EQUI Join: สินค้า $\bowtie$ สินค้า.รหัสสินค้า = การสั่งซื้อ.รหัสสินค้า การสั่งซื้อ

เลขที่ใบสั่งซื้อ	รหัสลูกค้า	รหัสสินค้า	รหัสสินค้า	ชื่อสินค้า	ราคาสินค้า
1	A001	003	003	สมุด	50
2	A015	001	001	ปากกา	100

NATURAL JOIN (4)

- EQUI Join: สินค้า ⋈ $_{\text{สินค้า.รหัสสินค้า = การสั่งซื้อ.รหัสสินค้า}}$ การสั่งซื้อ

เลขที่ใบสั่งซื้อ	รหัสลูกค้า	รหัสสินค้า	รหัสสินค้า	ชื่อสินค้า	ราคาสินค้า
1	A001	003	003	สมุด	50
2	A015	001	001	ปากกา	100

- Natural Join:

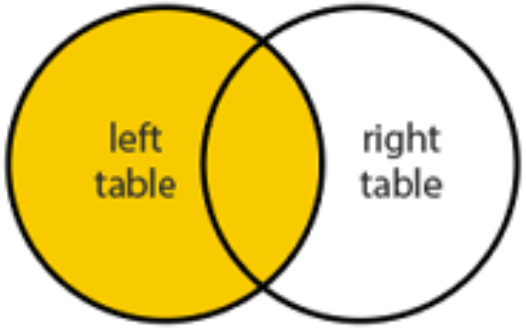
เลขที่ใบสั่งซื้อ	รหัสลูกค้า	รหัสสินค้า	ชื่อสินค้า	ราคาสินค้า
1	A001	003	สมุด	50
2	A015	001	ปากกา	100

OUTER JOIN (1)

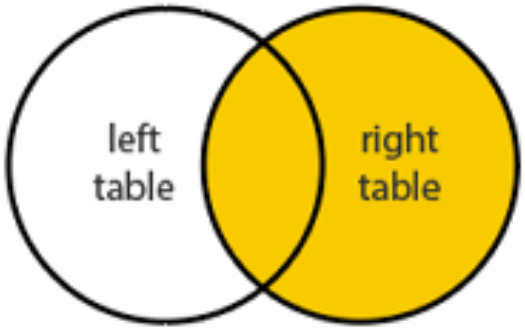
- An Outer Join doesn't require each record in the two join tables to have a matching record. In this type of join, the table retains each record even if no other matching record exists.
- Three types of Outer Joins are:
  - Left Outer Join
  - Right Outer Join
  - Full Outer Join

OUTER JOIN (2)

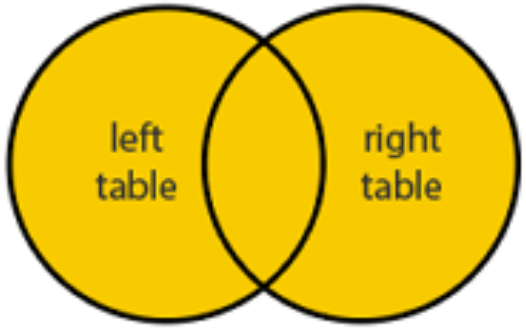
LEFT OUTER JOIN



RIGHT OUTER JOIN



FULL OUTER JOIN



EXAMPLE 4: LEFT OUTER JOIN

สินค้า

รหัสสินค้า	ชื่อ	ราคา
001	ปากกา	100
003	สมุด	50
005	ดินสอ	10

การสั่งซื้อ

เลขที่ใบสั่งซื้อ	รหัสลูกค้า	รหัสสินค้า
1	A001	003
2	A015	001
3	A016	004

ผลลัพธ์ของ Left Outer Join

เลขที่ใบสั่งซื้อ	รหัสลูกค้า	รหัสสินค้า	ชื่อสินค้า	ราคาสินค้า
1	A001	003	สมุด	50
2	A015	001	ปากกา	100
		005	ดินสอ	10

EXAMPLE 4: RIGHT OUTER JOIN

สินค้า

รหัสสินค้า	ชื่อ	ราคา
001	ปากกา	100
003	สมุด	50
005	ดินสอ	10

การสั่งซื้อ

เลขที่ใบสั่งซื้อ	รหัสลูกค้า	รหัสสินค้า
1	A001	003
2	A015	001
3	A016	004

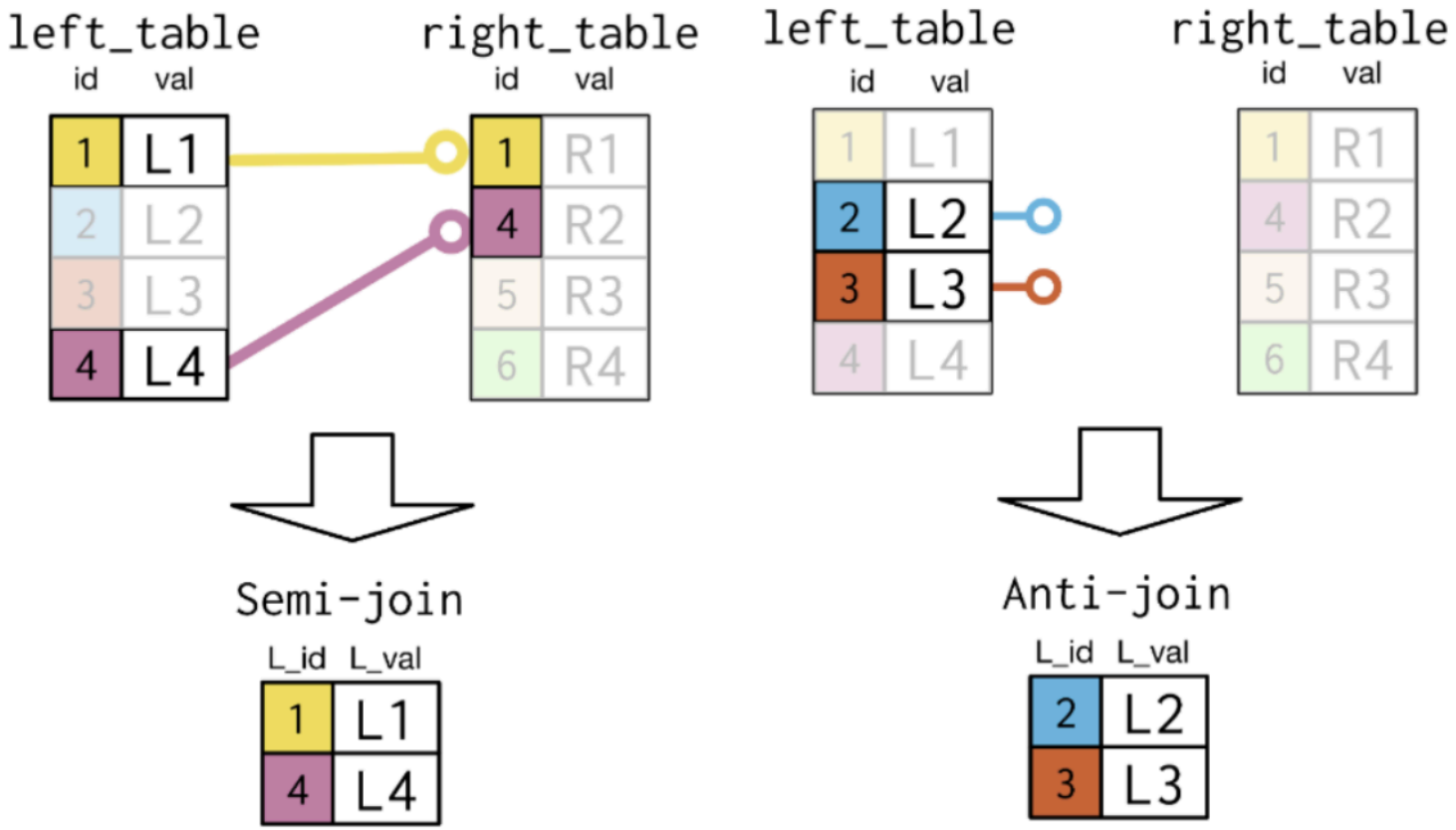
ผลลัพธ์ของ Right Outer Join

เลขที่ใบสั่งซื้อ	รหัสลูกค้า	รหัสสินค้า	ชื่อสินค้า	ราคาสินค้า
1	A001	003	สมุด	50
2	A015	001	ปากกา	100
3	A016	004		

SEMI-JOIN & ANTI-JOIN (1)

- Semi-join matches the rows of two relations and then show the matching rows of the relation whose name is mentioned to the left side of ⋉ Semi Join operator.
- While a semi-join returns one copy of each row in the first table for which at least one match is found, an anti-join returns one copy of each row in the first table for which no match is found.

SEMI-JOIN & ANTI-JOIN (2)



AGGREGATION AND GROUPING OPERATIONS

- As well as simply retrieving certain tuples and attributes of one or more relations, we often want to perform some form of **summation** or **aggregation** of data, similar to the totals at the bottom of a report, or some form of **grouping** of data, similar to subtotals in a report.
- The main aggregation function are:
  - COUNT
  - SUM
  - AVG
  - MIN
  - MAX

EXAMPLE 5: AGGREGATION

- How many properties cost more than £350 per month to rent?
- Find the minimum, maximum, and average staff salary.

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

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

(a)

myCount
5

(b)

myMin	myMax	myAverage
9000	30000	17000

EXAMPLE 6: GROUPING

- Find the number of staff working in each branch and the sum of their salaries.

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

TUPLE RELATIONAL CALCULUS

- In the tuple relational calculus we are interested in finding tuples for which a predicate is true.
- The calculus is based on the use of tuple variables.
- A tuple variable is a variable that ‘ranges over’ a named relation: that is, a variable whose only permitted values are tuples of the relation.
- For example:
  - List the names of all managers who earn more than £25,000.
  - List the staff who manage properties for rent in Glasgow.

EXAMPLE 7 (1)

- Use the database show in this figure to answer problem.

Table name: EMPLOYEE **Database name: CH2_STORE_CO**

EMP_CODE	EMP_TITLE	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_DOB	STORE_CODE
1	Mr.	Williamson	John	W	21-May-64	3
2	Ms.	Ratula	Nancy		09-Feb-69	2
3	Ms.	Greenboro	Lottie	R	02-Oct-61	4
4	Mrs.	Rumpersfro	Jennie	S	01-Jun-71	5
5	Mr.	Smith	Robert	L	23-Nov-59	3
6	Mr.	Renselaer	Cary	A	25-Dec-65	1
7	Mr.	Ogallo	Roberto	S	31-Jul-62	3
8	Ms.	Johnsson	Elizabeth	I	10-Sep-68	1
9	Mr.	Eindsmar	Jack	W	19-Apr-55	2
10	Mrs.	Jones	Rose	R	06-Mar-66	4
11	Mr.	Broderick	Tom		21-Oct-72	3
12	Mr.	Washington	Alan	Y	08-Sep-74	2
13	Mr.	Smith	Peter	N	25-Aug-64	3
14	Ms.	Smith	Sherry	H	25-May-66	4
15	Mr.	Olenko	Howard	U	24-May-64	5
16	Mr.	Archialo	Barry	V	03-Sep-60	5
17	Ms.	Grimaldo	Jeanine	K	12-Nov-70	4
18	Mr.	Rosenberg	Andrew	D	24-Jan-71	4
19	Mr.	Rosten	Peter	F	03-Oct-68	4
20	Mr.	Mckee	Robert	S	06-Mar-70	1
21	Ms.	Baumann	Jennifer	A	11-Dec-74	3

Table name: STORE

STORE_CODE	STORE_NAME	STORE_YTD_SALES	REGION_CODE	EMP_CODE
1	Access Junction	$1,003,455.76	2	8
2	Database Corner	$1,421,987.39	2	12
3	Tuple Charge	$986,783.22	1	7
4	Attribute Alley	$944,568.56	2	3
5	Primary Key Point	$2,930,098.45	1	15

Table name: REGION

REGION_CODE	REGION_DESCRIPT
1	East
2	West

EXAMPLE 7 (2)

- For each table, identify the primary key and the foreign key(s). If a table does not have a foreign key, write “None” in the assigned space.

TABLE	PRIMARY KEY	FOREIGN KEY(S)
EMPLOYEE		
STORE		
REGION		

EXAMPLE 7 (3)

- Do the tables exhibit entity integrity? Answer “Yes” or “No”, the explain your answer.

TABLE	ENTITY INTEGRITY?	EXPLANATION
EMPLOYEE		
STORE		
REGION		

EXAMPLE 7 (4)

- Do the tables exhibit referential integrity? Answer “Yes” or “No”, the explain your answer. Write “N/A” (Not Applicable) if the table does not have a foreign key.

TABLE	REFERENTIAL INTEGRITY?	EXPLANATION
EMPLOYEE		
STORE		
REGION		

CASE STUDY

- Design a table to collect data on students in the Faculty of Engineering.
- The table must indicate the **department** and **year of study**.
- The table may include other attributes as appropriate.
- The designed table must comply with database integrity rules.
- Write the table's metadata and at least four rows of sample data into an Excel file.