COMP1140: Database and Information Management

Lecture Note – Week 4

Dr Suhuai Luo

School of EEC

University of Newcastle

Notice

- Assignment 1 is due this Friday (Aug 24) at 12pm
  - Submit both soft & hard copy
    - Soft copy –

    Blackboard -> Assessment -> AssignmentsSubmission-> Assignment1

    - Hardcopy with signed cover page, submit to School Office
  - In your submission, make sure to put your lab session/time on the cover sheet.
  - Make sure your EER has big enough fonts & black color and doesnot spread too many pages in print.

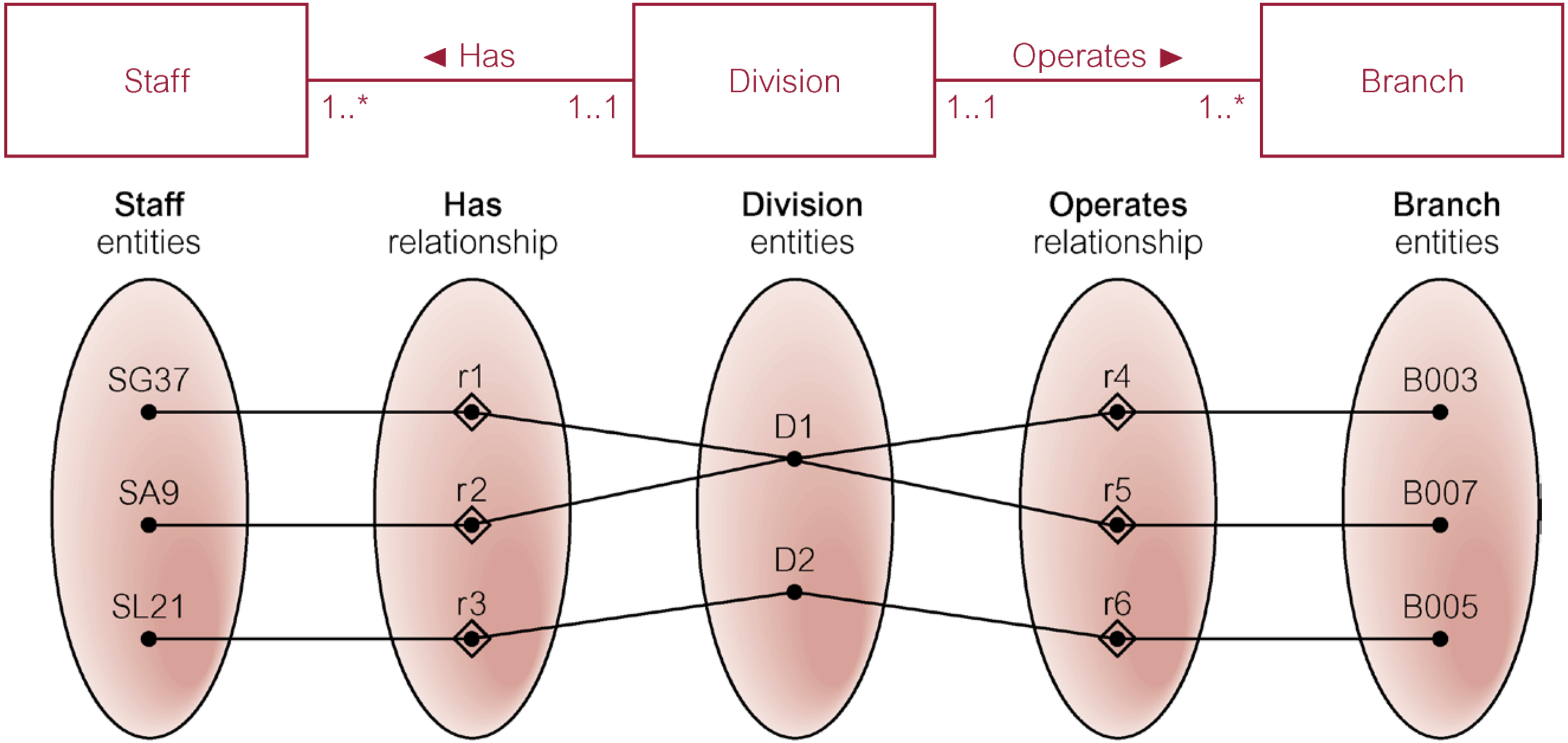
Last lecture

- Conceptual DB Design with EER

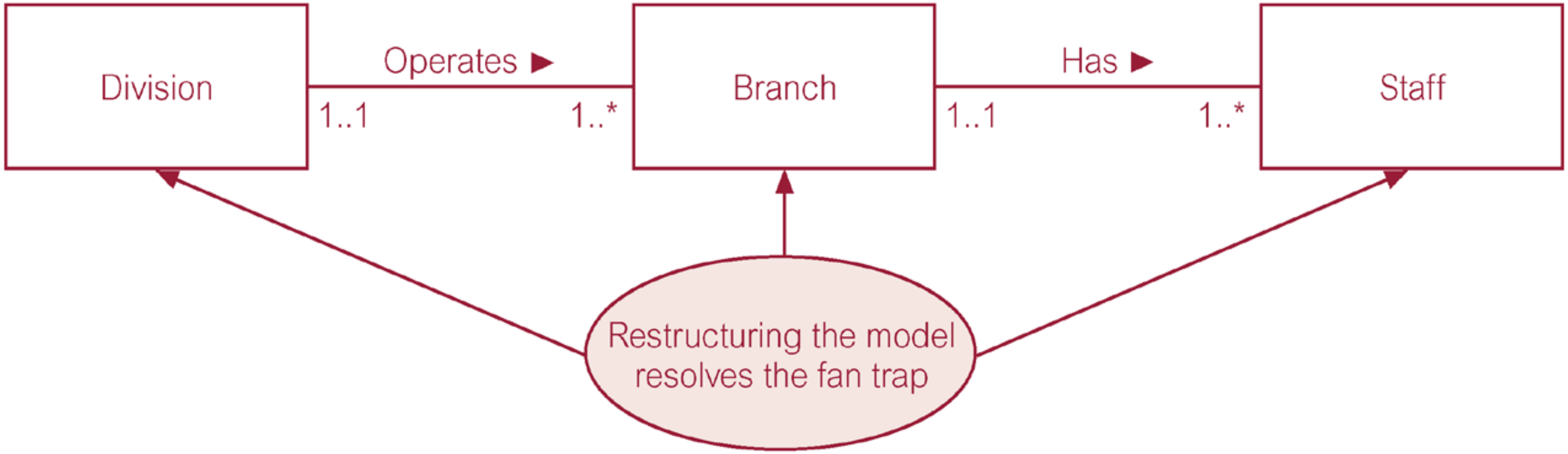
- Any questions?

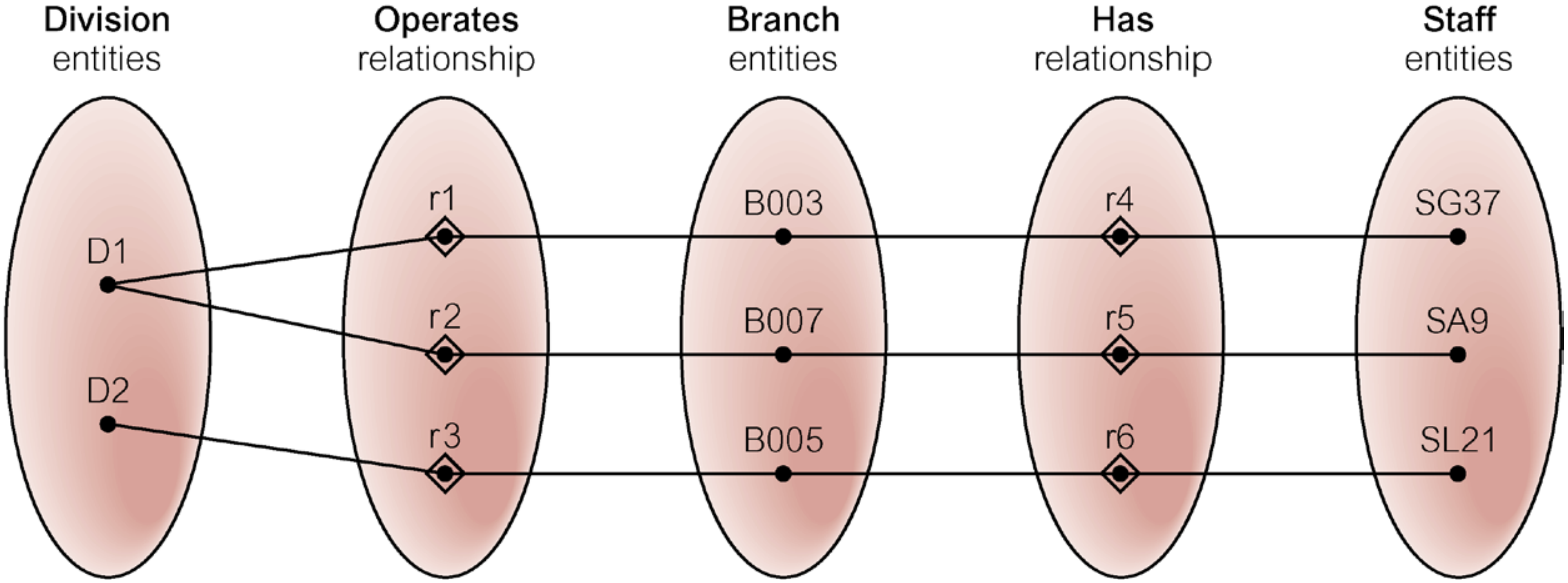
Sometimes, it’s necessary to re-arrange entities

- Where a model represents a relationship between entity types, but pathway between certain entity occurrences is ambiguous - fan trap .



Q: At which branch office does staff number SG37 work?

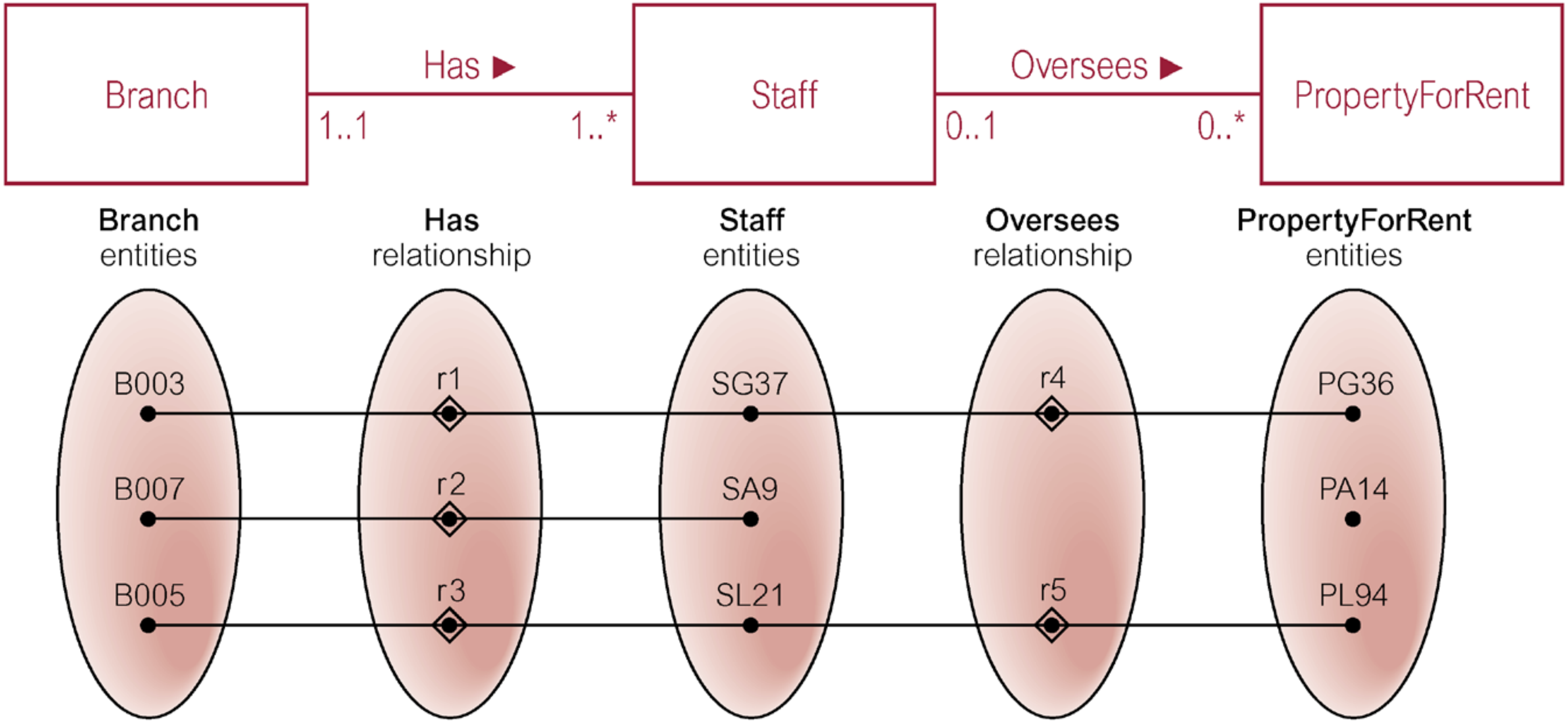




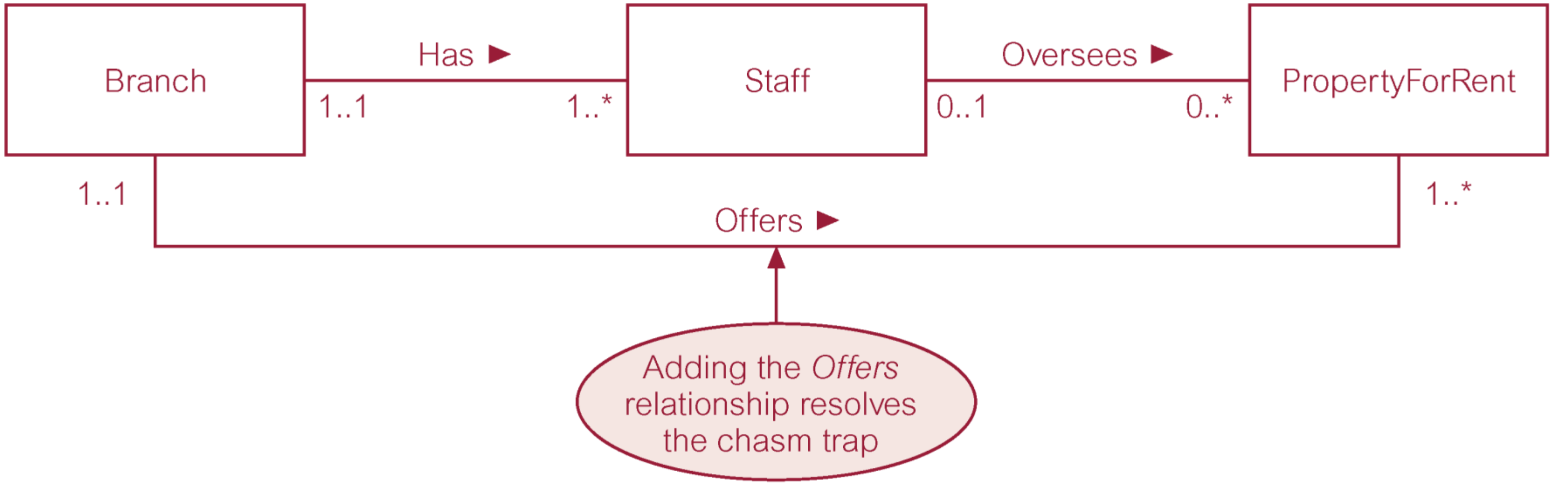
A: SG37 works at branch B003

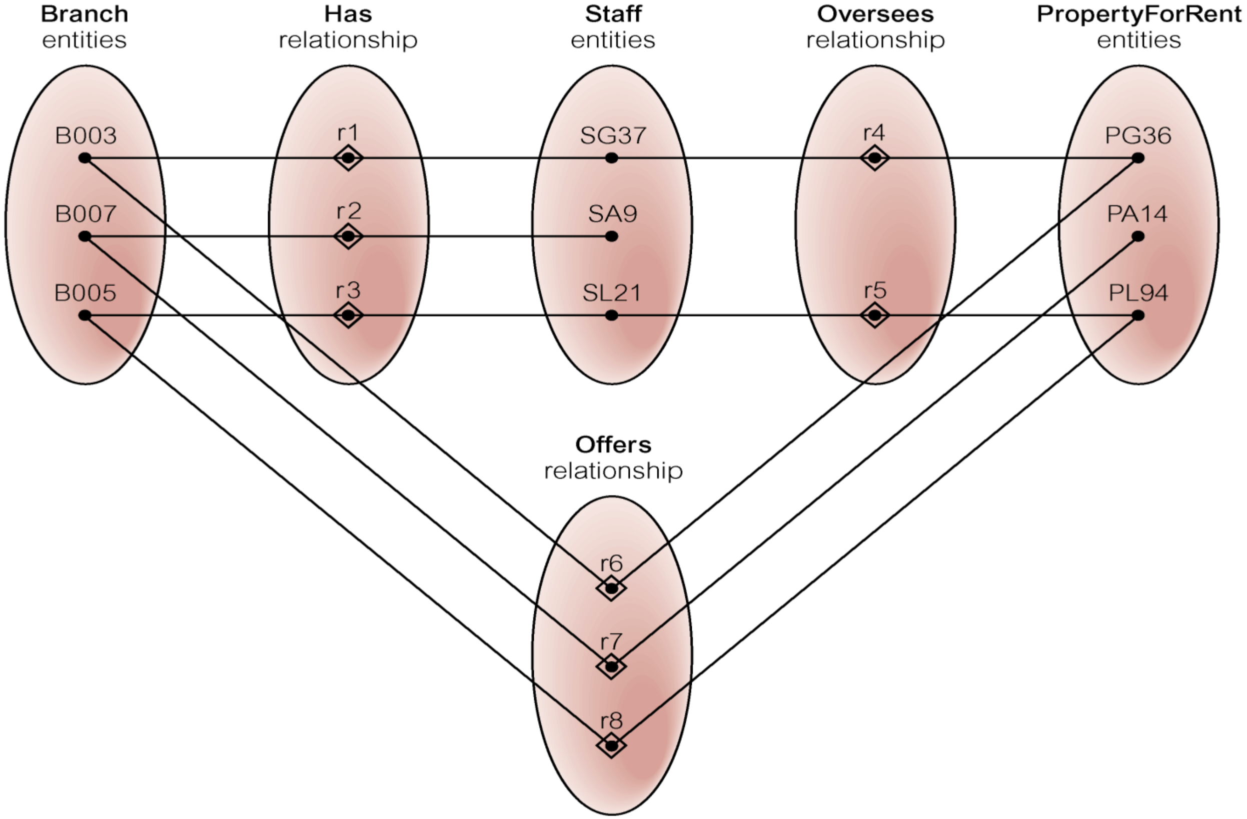
Sometimes, it's necessary to add another link

- Where a model suggests the existence of a relationship between entity types, but pathway does not exist between certain entity occurrences - chasm trap.



Q: At which branch office is property PA14 available?





A: B007 offers PA14

This lecture

- Relational Model
- Mapping from EER model to relational model (i.e. first step of Logical Database Design)
- Discussion on A1
- References: chapters 4 & 17

Logical Database Design

- Process of constructing a model of the data used in an enterprise based on a specific data model, but independent of a particular DBMS and other physical considerations.

Logical Database Design – 7 processes

- *Get relations from EER*
- Relation normalisation
- Validate relations against user transaction
- *Check integrity constrains*
- Review logical data model with user
- Merge logical models into global model
- Check for future growth

Relational Model

- Initially proposed by E.F. Codd in 1970 in a paper titled “A relational model of data for large shared data banks”

- Major advantages over previous approaches
  - **Data independence** (Application programs not affected by internal representation of data)
  - **Sound theoretical foundation** for semantics, redundancy and consistency problems (based on set theory)
  - Set-oriented **data manipulation languages**

Relational Model (contd.)

- In the 70s many research prototypes of DBMSs were implemented to prove its feasibility and power

- A notable one is IBM’s System R project which led to the development of “Structured English Query Language (SEQUEL)”

- SEQUEL was later standardized by ISO/ANSI as Structured Query Language (SQL)

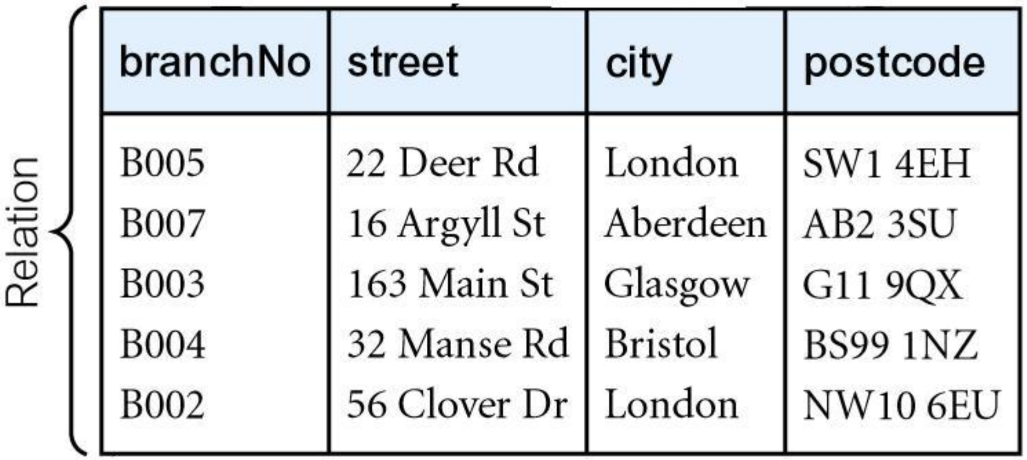
Relational Model (contd.)

- Today, a majority of databases used are relational or extensions of it!
- Over US$ 25 billion/year in sales in software licenses for DBMSs and tools
- Codd won the Turing Award for his work

Relational Model Terminology

- A relation is a table with columns and rows.
  - Only applies to logical structure of the database, not the physical structure.

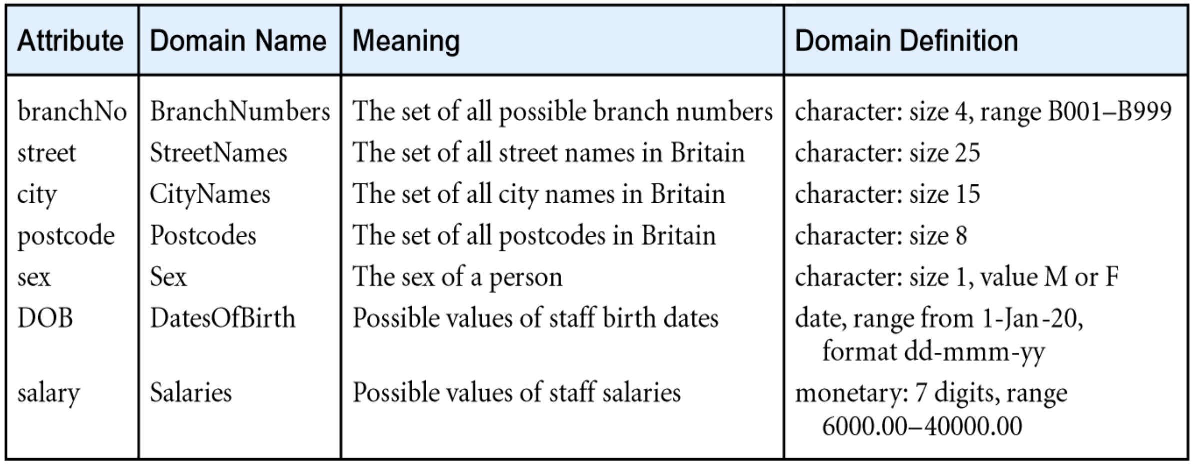
- Example: Branch



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

Relational Model Terminology (contd.)

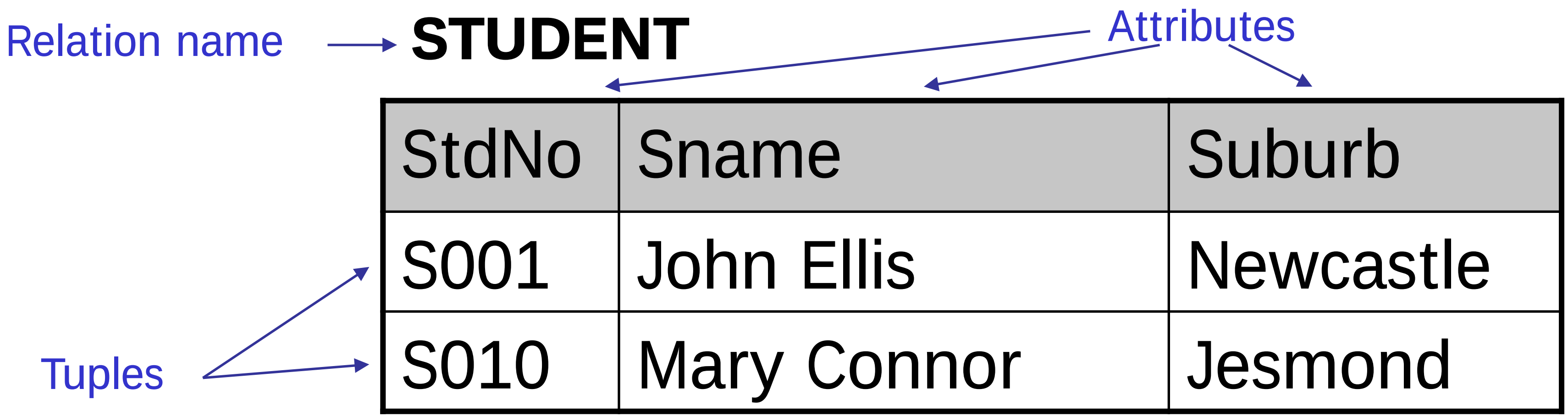
- **Attribute** is a named column of a relation.
- **Domain** is the set of allowable values for one or more attributes.
- Example:



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 (contd.)

- **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.
- E.g.



Relation name → **STUDENT**

Attributes

Tuples

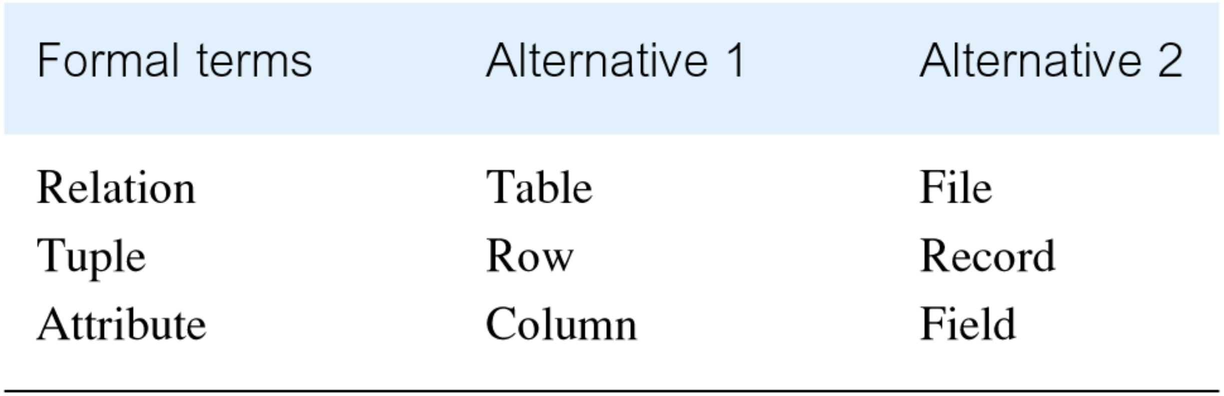
StdNo	Sname	Suburb
S001	John Ellis	Newcastle
S010	Mary Connor	Jesmond

Degree = ?, Cardinality = ?

Relational Model Terminology (contd.)

- **Relational database** is a collection of relations with distinct names

- Alternative terminology:



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

© Pearson Education Limited 1995, 2005

Database Relations

- Formally, *a relation R with attributes* $A_1, \ldots,$ A_n is a **set** of tuples such that each tuple is a collection of values

  $\langle d_1, d_2, \ldots, d_n \rangle$

  where $d_i \in \{D_i \text{ or NULL}\}$,

  $i = 1, \ldots, n$ and

  D_i is the domain of attribute $A_{i.}$

  *NULL is a special value (meaning unknown, unspecified, undefined, etc.)

Database Relations (contd.)

- Relation schema:
  - **relation name**
  - {**name** of each field & **domain** of each field}

Example

STUDENT

(*StdNo*: {characters of S001-S999},
sName: {set of all student names},
suburb: {set of all suburbs of students})

Database Relations (contd.)

- Relation instance:
  - Set of tuples that belong to the relation at a particular point in time
- Example

STUDENT

StdNo	Sname	Suburb
S001	John Ellis	Newcastle
S010	Mary Connor	Jesmond

Database Relations (contd.)

- Relational database schema:
  - A set of relation schemas with distinct names

Keys

- **Superkey**: A set of one or more attributes that uniquely identify a tuple

- **Candidate key**: A superkey such that no proper subset is a superkey

- E.g. Student(StdNo, name, suburb)

  Superkeys:
      {StdNo,
      (StdNo, name),
      (StdNo, suburb),
      (StdNo, name, suburb)}
  Candidate keys:
      {StdNo}

Keys (contd.)

- **Primary key**: A candidate key selected by the database designer to uniquely identify a tuple

- Only one primary key exists for a relation

- Representing database schema:

Student(StdNo, name, suburb)
Course(courseNo, name, credits)

*relations with primary keys underlined

Integrity Constraints

- **Integrity constraint** is a rule (or check) that is enforced to ensure that correct data exists in the database

- There are different kinds of integrity constraints:
  - **Domain constraints**
  - **Entity Integrity Constraints**
  - **Referential Integrity Constraints**
  - **General Constraints**

Domain constraints

- Domain constraints specify that each value for an attribute (say A) must be an *atomic* value from the ***domain*** of A or the special ***null*** value

- Atomic value: That is, the value is non divisible into components
  - Therefore, relational model doesn’t allow multi-valued attributes or composite attributes!

- Relational database provides data types to specify valid domains (e.g. characters, integers, etc.)

Entity Integrity Constraint

- Entity integrity constraint specifies that no attribute in a primary key can be null

STUDENT

StdNo	Sname	Suburb
S001	John Ellis	Newcastle
NULL	Mark Anthony	Belmont
S010	Mary Connor	Jesmond

NULLs are not allowed as primary key values

Foreign keys and referential integrity

- Foreign Key: an attribute in a relation, serving as PK or matching a candidate key of another relation in the same DB.
- Foreign Key Constraints
  - We can designate an attribute(s) as a **foreign key**(**s**)
  - Foreign key(s) attribute(s) always refer to PK attribute(s) of another entity
  - Foreign keys enforce referential integrity constraints
- Referential Integrity Constraints: a rule stating that either each FK value must match a PK value in another relation, or the FK value be NULL

Foreign keys … (contd.)

Foreign key (FK) attributes in R_1 referring to R_2 have the following rules:

- The FK attributes in R_1 have the same domain(s) as the candidate key attributes of R_2
- The value of FK in tuple t_1 in R_1 must reference an existing candidate key value in tuple t_2 of R_2

Foreign keys … (contd.)

STUDENT

Primary Key

StdNo	Sname	Suburb
S001	John Ellis	Newcastle
S020	Mark Anthony	Belmont

REGISTER

Foreign Key

StdNo	Course	Grade
S001	INFT2040	A
S012	INFT2009	B
S001	INFT2031	A-

NOT ALLOWED: Foreign key value must refer to an existing primary key value

General Constraints

- Additional constraints that exists in the enterprise.

  E.g. “A student can register to at most 7 courses per semester”

Summary – Relational Model

- Relation – schema, instance, cardinality, degree
- Attribute
- Domain
- Tuple
- Superkey, candidate key and primary key
- Relational database, relational database schema
- Integrity constraints
  - Domain constraints
  - Entity integrity constrains
  - Referential Integrity constraints (foreign keys)
  - General constraints

EER – Relational Mapping

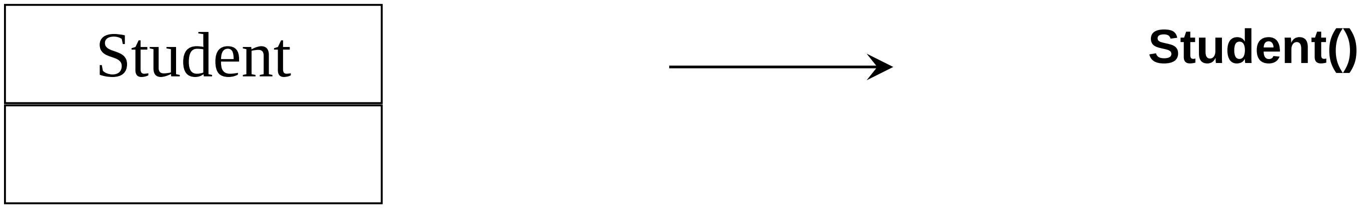
- Mapping guidelines overview
  - Note: the relational schema is presented using Database Definition Language (DBDL) notation

EER to Relational Mapping

EER Model **Relational Model**

Strong Entity → relation

For example,



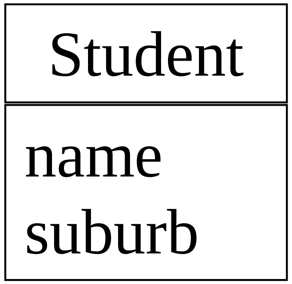
Student

Student()

EER to Relational Mapping (contd.)

EER Model		**Relational Model**
Simple Attributes	→	attributes

For example,



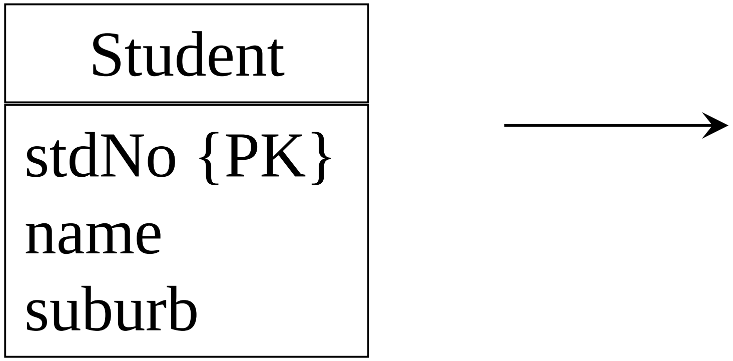
→

Student(name, suburb**)**

EER to Relational Mapping (contd.)

EER Model		**Relational Model**
Primary Key	→	Primary Key

For example,



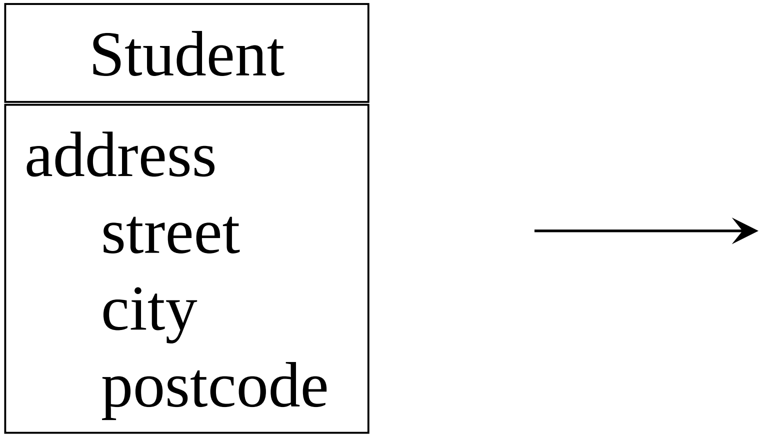
Student(stdNo, name, suburb**)**
Primary Key stdNo

EER to Relational Mapping (contd.)

EER Model **Relational Model**

Composite attributes → Set of simple attributes

For example,



Student(street, city, postcode**)**

EER to Relational Mapping (contd.)

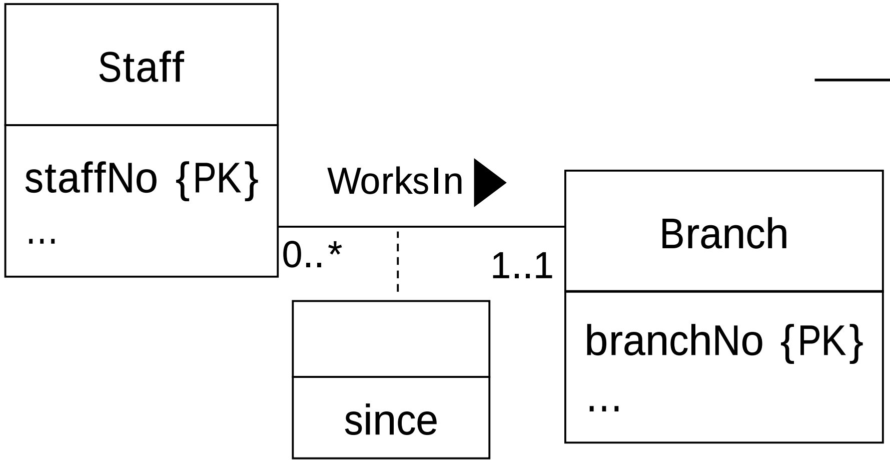
EER Model

1:*

For example,

Relational Model

Foreign key relationship



Staff(staffNo,..., worksIn, since, branchNo **)**
Primary Key staffNo
Foreign Key branchNo **references** Branch(branchNo)

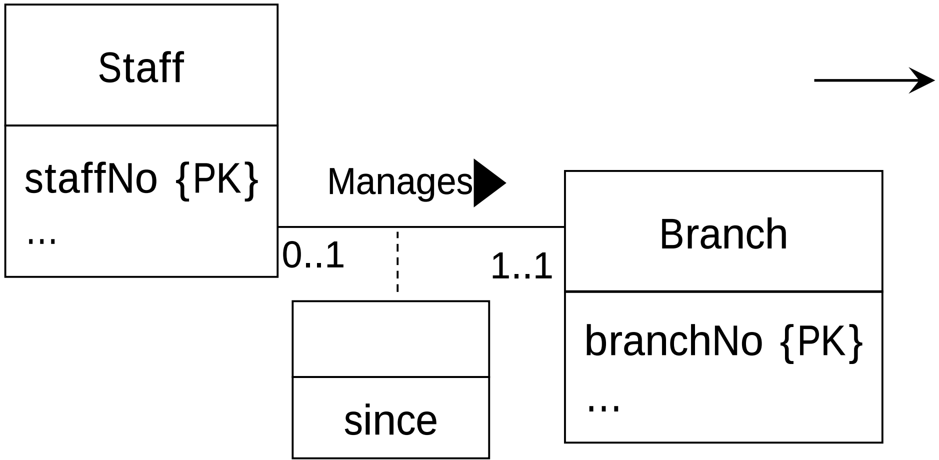
Branch(branchNo, ...**)**
Primary Key branchNo

EER to Relational Mapping (contd.)

EER Model **Relational Model**

1:1 relationship → Foreign key relationship

For example,



Staff(staffNo,...**)**
Primary Key staffNo

Branch(branchNo,..., manager, since**)**
Primary Key branchNo
Foreign Key staffNo **references** Staff(staffNo)

EER to Relational Mapping (contd.)

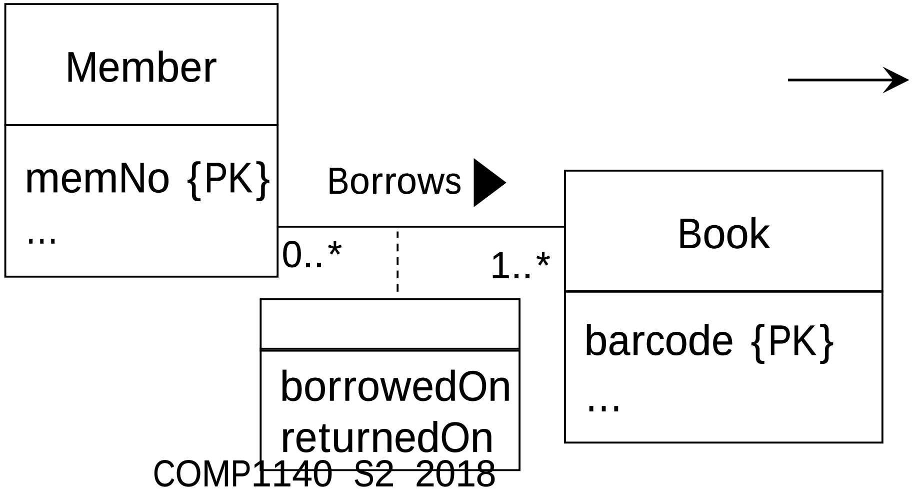
- In a 1:1 relationship R from entity A to B and
  - Mandatory participation on one side:
    - If only B is in mandatory participation with A on R (i.e., 1..1 to 0..1), then the foreign key is placed in B
  - Mandatory participation on both sides:
    - If both A and B are in mandatory participation with R (i.e., 1..1 to 1..1), then A & B are represented as a single relation (containing attributes of both A and B)
  - Optional participation on both sides:
    - Otherwise (like 0..1 to 0..1), foreign key can be placed on either relation A or B

EER to Relational Mapping (contd.)

EER Model **Relational Model**

: → "Relationship" relation and two foreign keys

For example,



→

Member(memNo,...**)**
Primary Key memNo

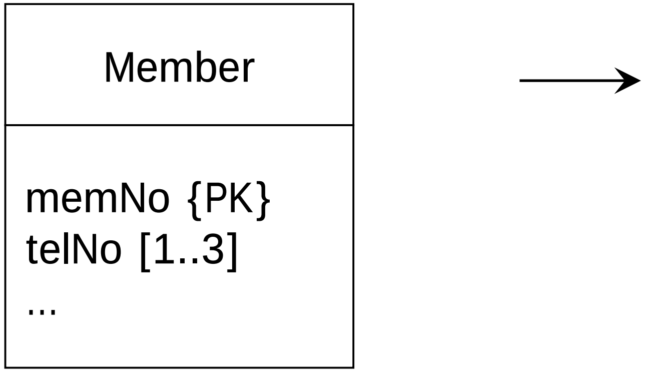
Book(barCode, ...**)**
Primary Key barCode

Borrows(memNo, barCode, borrowedOn, returnedOn**)**
Primary Key memNo, barCode, borrowedOn
Foreign Key memNo **references** Member(memNo)
Foreign Key barCode **references** Book(barCode)

EER to Relational Mapping (contd.)

EER Model	**Relational Model**
Multivalued attribute →	Relation & Foreign Key

For example,



Member

memNo {PK}
telNo [1..3]
...

Member(memNo,...**)**
Primary Key memNo

Telephone (memNo, telNo**)**
Primary Key memNo, telNo
Foreign Key memNo **references** Member(memNo)

EER to Relational Mapping (contd.)

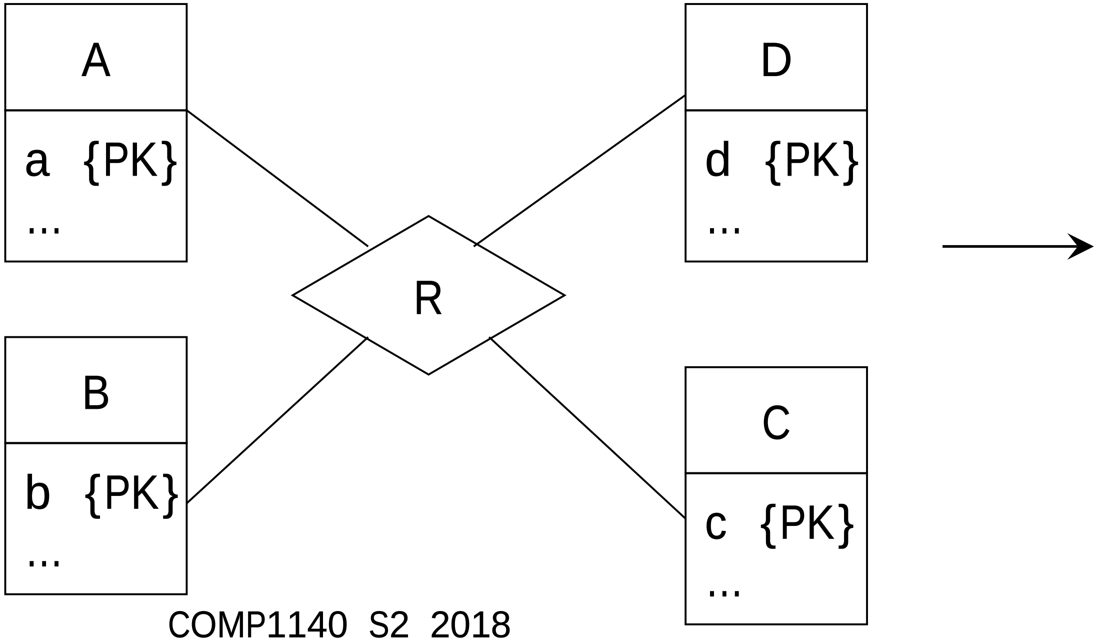
EER Model

N-ary relationship

For example,

Relational Model

“Relationship” relation and n foreign keys



A(a,...**)**
Primary Key a

B(b,...**)**
Primary Key b

C(c,...**)**
Primary Key c

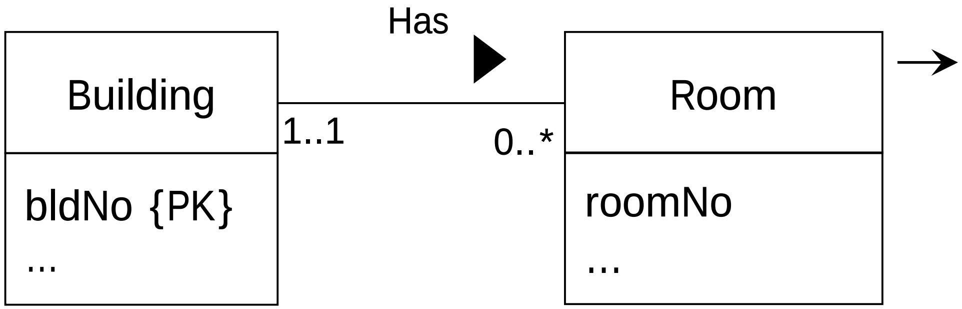
D(d,...**)**
Primary Key d

R(a, b, c, d**)**
Primary Key a, b, c, d
Foreign Key a **references** A(a)
Foreign Key b **references** B(b)
Foreign Key c **references** C(c)
Foreign Key d **references** D(d)

EER to Relational Mapping (contd.)

EER Model	**Relational Model**
Weak entity	Relation with foreign key.

For example

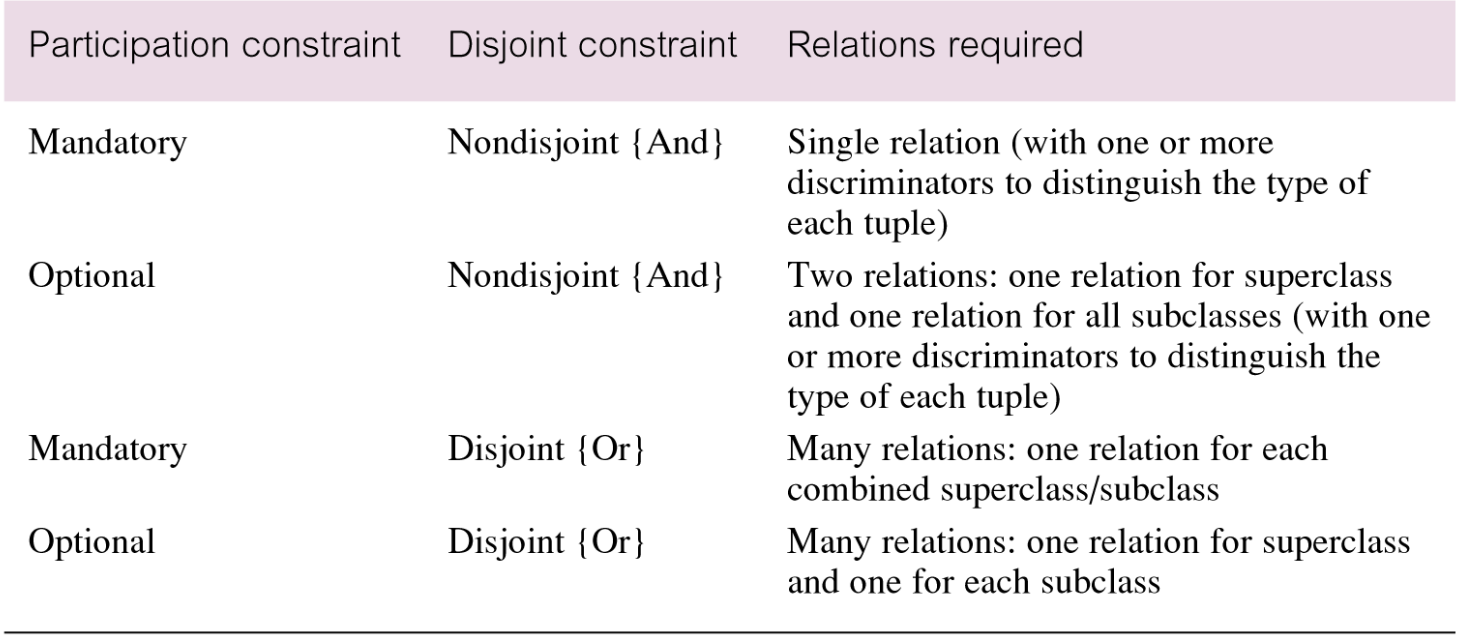


Building(bldNo, ...**)**
Primary Key bldNo

Room(bldNo, roomNo, ...**)**
Primary Key bldNo, roomNo
Foreign Key bldNo **references** Building(bldNo)

EER to Relational Mapping (contd.)

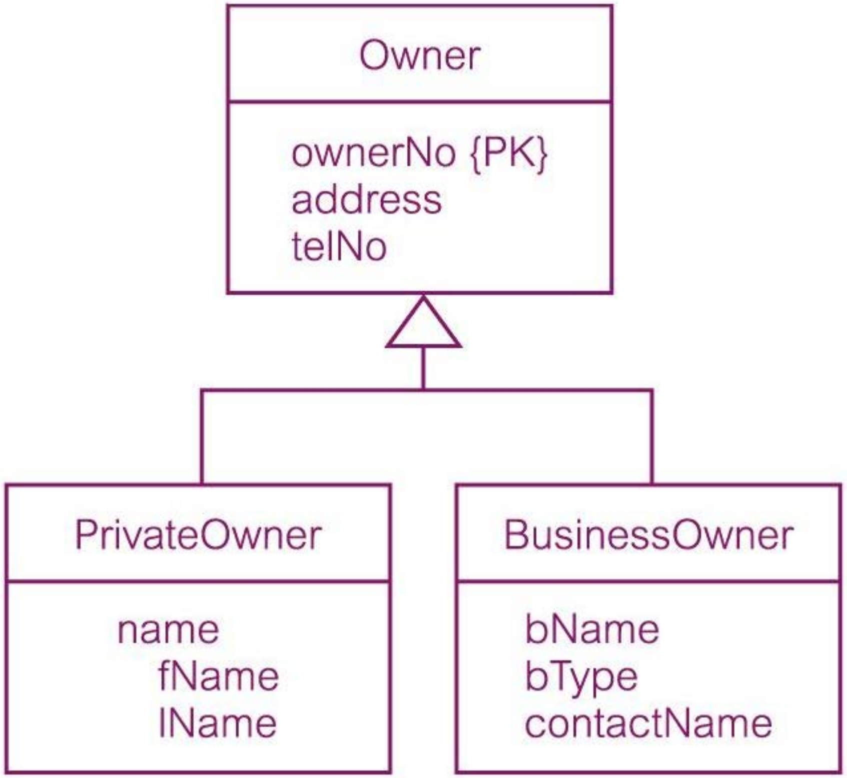
- Mapping superclass/subclass relationships have many options



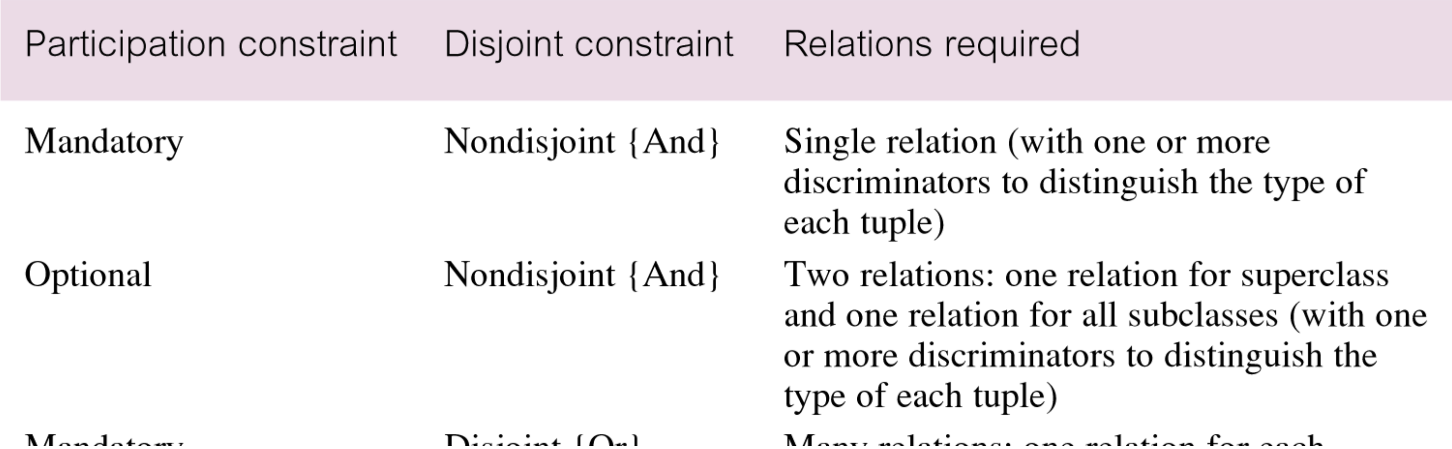
Participation constraint	Disjoint constraint	Relations required
Mandatory	Nondisjoint {And}	Single relation (with one or more discriminators to distinguish the type of each tuple)
Optional	Nondisjoint {And}	Two relations: one relation for superclass and one relation for all subclasses (with one or more discriminators to distinguish the type of each tuple)
Mandatory	Disjoint {Or}	Many relations: one relation for each combined superclass/subclass
Optional	Disjoint {Or}	Many relations: one relation for superclass and one for each subclass

EER to Relational Mapping (contd.)

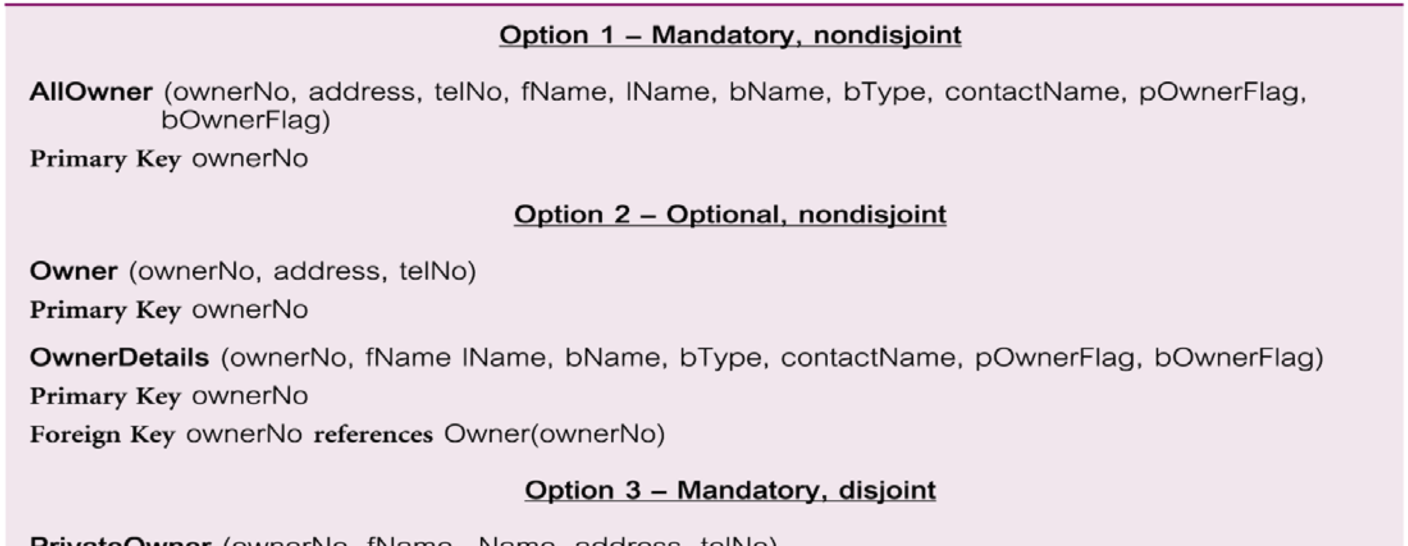
- Example



EER to Relational Mapping (contd.)



Participation constraint	Disjoint constraint	Relations required
Mandatory	Nondisjoint {And}	Single relation (with one or more discriminators to distinguish the type of each tuple)
Optional	Nondisjoint {And}	Two relations: one relation for superclass and one relation for all subclasses (with one or more discriminators to distinguish the type of each tuple)
Mandatory	Disjoint {Or}	Many relations: one relation for each



Option 1 – Mandatory, nondisjoint

AllOwner (ownerNo, address, telNo, fName, lName, bName, bType, contactName, pOwnerFlag, bOwnerFlag)
Primary Key ownerNo

Option 2 – Optional, nondisjoint

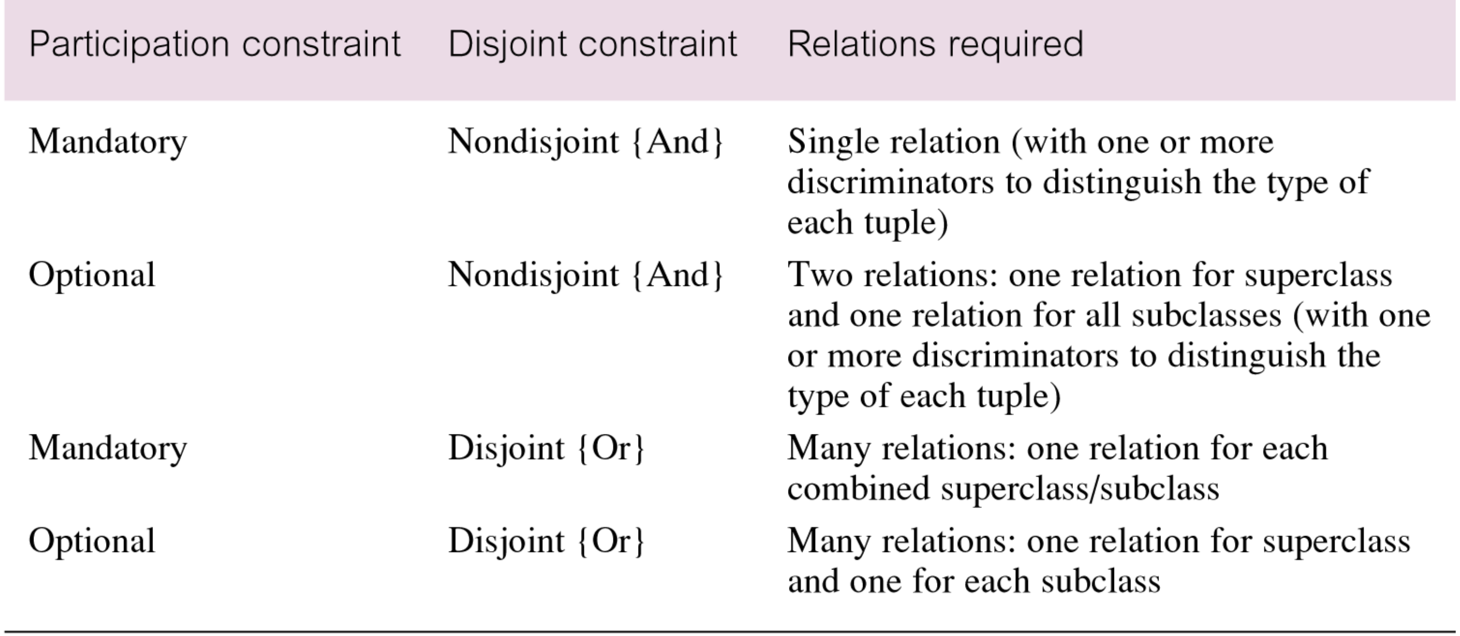
Owner (ownerNo, address, telNo)
Primary Key ownerNo

OwnerDetails (ownerNo, fName lName, bName, bType, contactName, pOwnerFlag, bOwnerFlag)
Primary Key ownerNo
Foreign Key ownerNo **references** Owner(ownerNo)

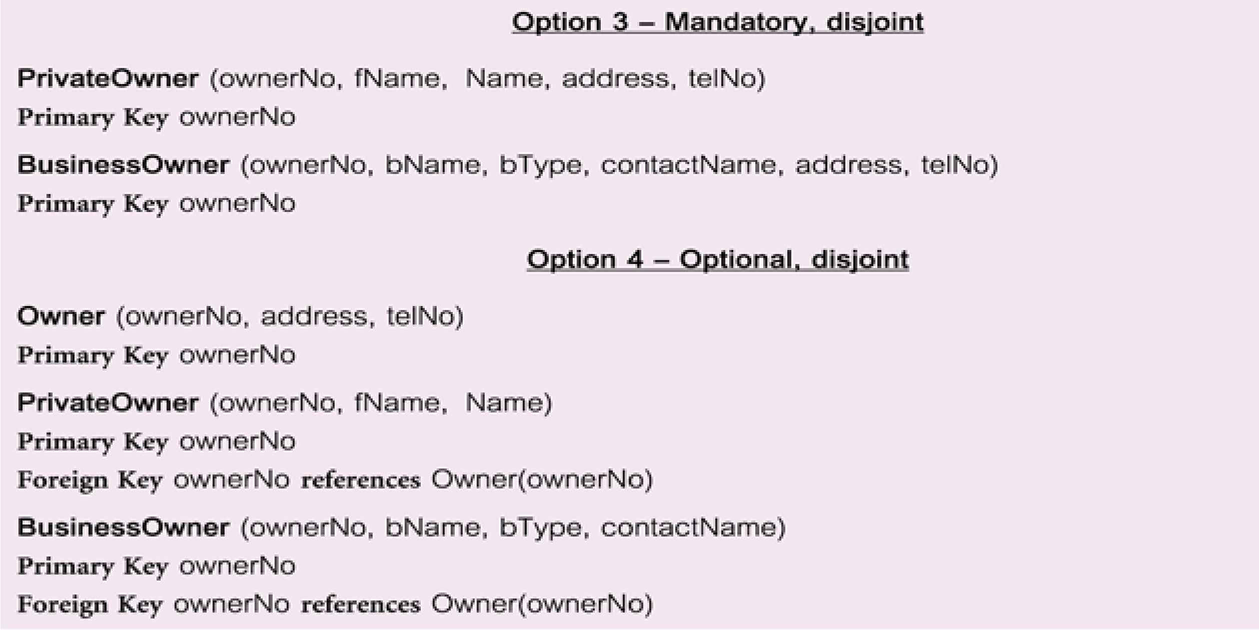
Option 3 – Mandatory, disjoint

PrivateOwner (ownerNo, fName, lName, address, telNo)

E
(



Participation constraint	Disjoint constraint	Relations required
Mandatory	Nondisjoint {And}	Single relation (with one or more discriminators to distinguish the type of each tuple)
Optional	Nondisjoint {And}	Two relations: one relation for superclass and one relation for all subclasses (with one or more discriminators to distinguish the type of each tuple)
Mandatory	Disjoint {Or}	Many relations: one relation for each combined superclass/subclass
Optional	Disjoint {Or}	Many relations: one relation for superclass and one for each subclass



Option 3 – Mandatory, disjoint

PrivateOwner (ownerNo, fName, Name, address, telNo)
Primary Key ownerNo

BusinessOwner (ownerNo, bName, bType, contactName, address, telNo)
Primary Key ownerNo

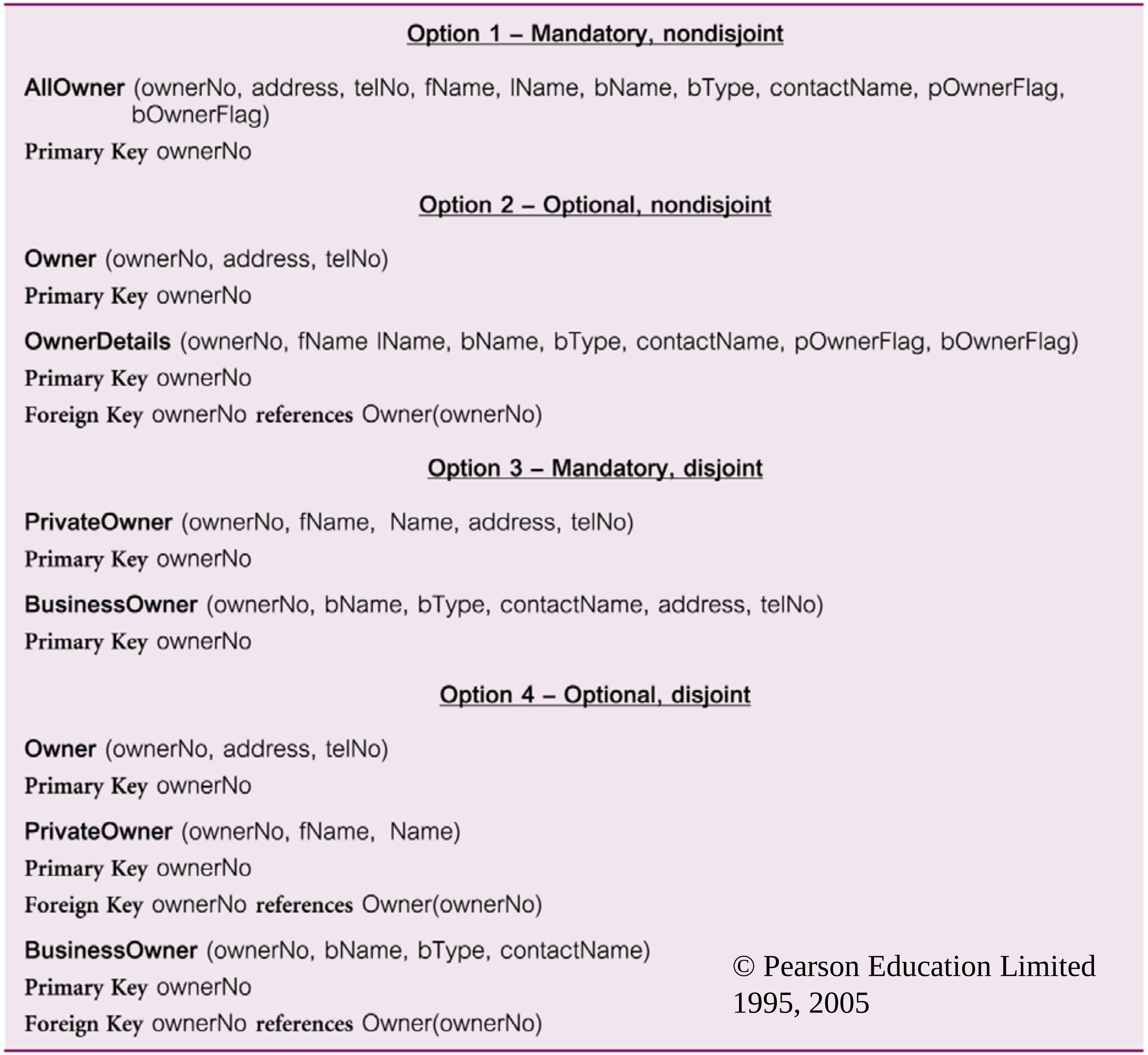
Option 4 – Optional, disjoint

Owner (ownerNo, address, telNo)
Primary Key ownerNo

PrivateOwner (ownerNo, fName, Name)
Primary Key ownerNo
Foreign Key ownerNo **references** Owner(ownerNo)

BusinessOwner (ownerNo, bName, bType, contactName)
Primary Key ownerNo
Foreign Key ownerNo **references** Owner(ownerNo)

EER to Relational Mapping (contd.)



Option 1 – Mandatory, nondisjoint

AllOwner (ownerNo, address, telNo, fName, lName, bName, bType, contactName, pOwnerFlag, bOwnerFlag)
Primary Key ownerNo

Option 2 – Optional, nondisjoint

Owner (ownerNo, address, telNo)
Primary Key ownerNo

OwnerDetails (ownerNo, fName lName, bName, bType, contactName, pOwnerFlag, bOwnerFlag)
Primary Key ownerNo
Foreign Key ownerNo **references** Owner(ownerNo)

Option 3 – Mandatory, disjoint

PrivateOwner (ownerNo, fName, Name, address, telNo)
Primary Key ownerNo

BusinessOwner (ownerNo, bName, bType, contactName, address, telNo)
Primary Key ownerNo

Option 4 – Optional, disjoint

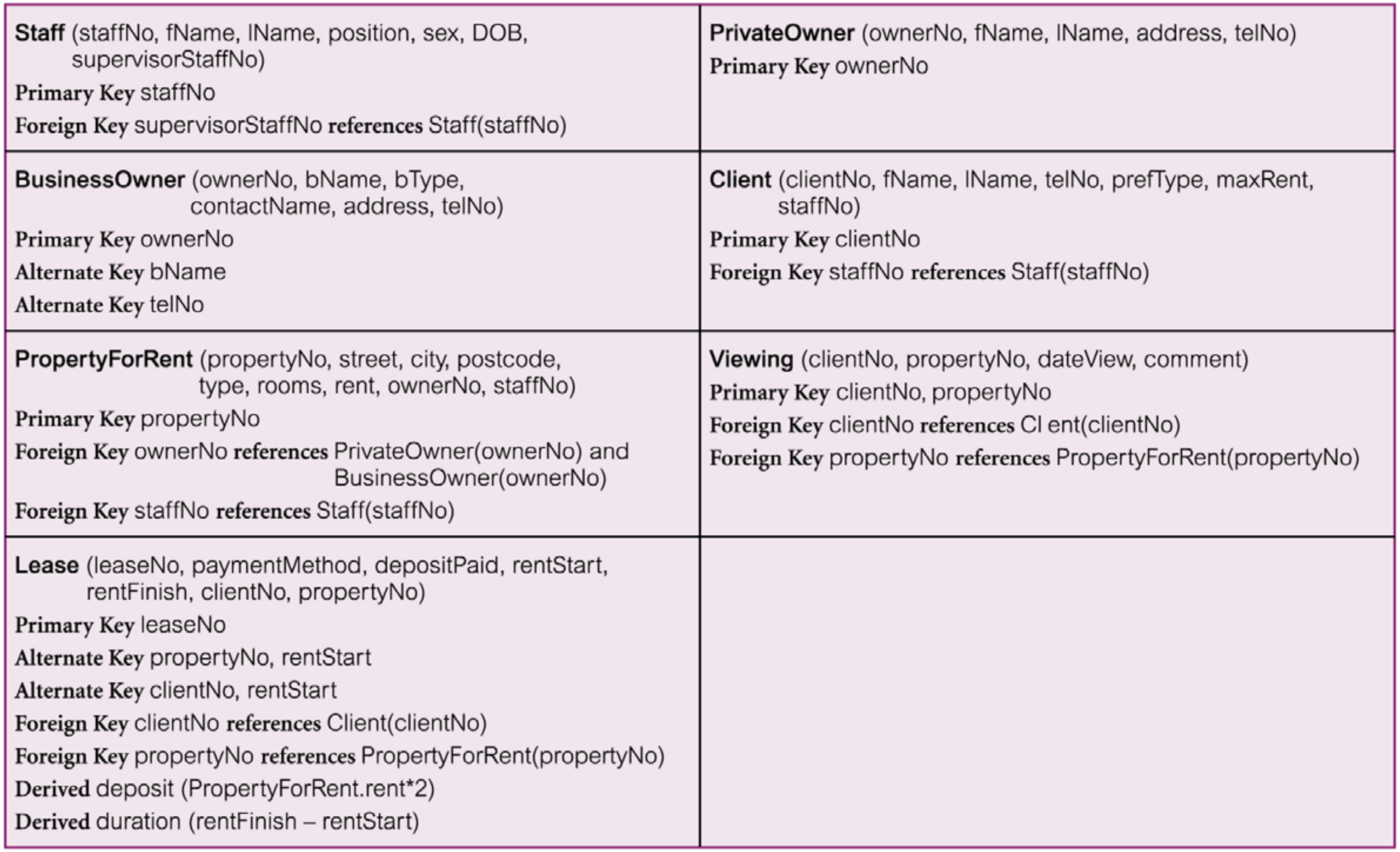
Owner (ownerNo, address, telNo)
Primary Key ownerNo

PrivateOwner (ownerNo, fName, Name)
Primary Key ownerNo
Foreign Key ownerNo **references** Owner(ownerNo)

BusinessOwner (ownerNo, bName, bType, contactName)
Primary Key ownerNo
Foreign Key ownerNo **references** Owner(ownerNo)

EER – Relational Mapping (contd.)

Example of a relational schema in DBDL



Staff (staffNo, fName, lName, position, sex, DOB, supervisorStaffNo) **Primary Key** staffNo **Foreign Key** supervisorStaffNo **references** Staff(staffNo)	**PrivateOwner** (ownerNo, fName, lName, address, telNo) **Primary Key** ownerNo
BusinessOwner (ownerNo, bName, bType, contactName, address, telNo) **Primary Key** ownerNo **Alternate Key** bName **Alternate Key** telNo	**Client** (clientNo, fName, lName, telNo, prefType, maxRent, staffNo) **Primary Key** clientNo **Foreign Key** staffNo **references** Staff(staffNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo, staffNo) **Primary Key** propertyNo **Foreign Key** ownerNo **references** PrivateOwner(ownerNo) and BusinessOwner(ownerNo) **Foreign Key** staffNo **references** Staff(staffNo)	**Viewing** (clientNo, propertyNo, dateView, comment) **Primary Key** clientNo, propertyNo **Foreign Key** clientNo **references** Cl ent(clientNo) **Foreign Key** propertyNo **references** PropertyForRent(propertyNo)
Lease (leaseNo, paymentMethod, depositPaid, rentStart, rentFinish, clientNo, propertyNo) **Primary Key** leaseNo **Alternate Key** propertyNo, rentStart **Alternate Key** clientNo, rentStart **Foreign Key** clientNo **references** Client(clientNo) **Foreign Key** propertyNo **references** PropertyForRent(propertyNo) **Derived** deposit (PropertyForRent.rent*2) **Derived** duration (rentFinish – rentStart)	

Summary – EER to Relational Mapping

- Strong entity types
- Attributes – Simple, composite, multi-valued
- Primary keys
- Relationships (1:1, 1:*, *:*, n-ary)
- Weak entity types
- Superclass/subclass relationships

Assignment 2 – part 1

- Map EER to relational model

Assignment 1 Discussion

- Note: way of submission:
- **zip** all required files into one zip file (basically 2 files - one is your word-format report, the other is Visio or other-format EER). The file name **MUST** be identified by 4 sections: A1, your first name, your surname, and your student number, e.g., ***A1SimonLee1234567.zip***
- The specification
- Sample requirements
- Sample EER
- Marking Scheme
- Q?
  - (SQL Server installation, Visio usage...)

Lab This Week

- Map EER models to relational models
- SQL exercises on creating relational database schema, considering domain, default value, referential integrity constrains

Summary Qs

- What are the processes of logical DB design?
- Understand relation model – terminology & mapping from EER