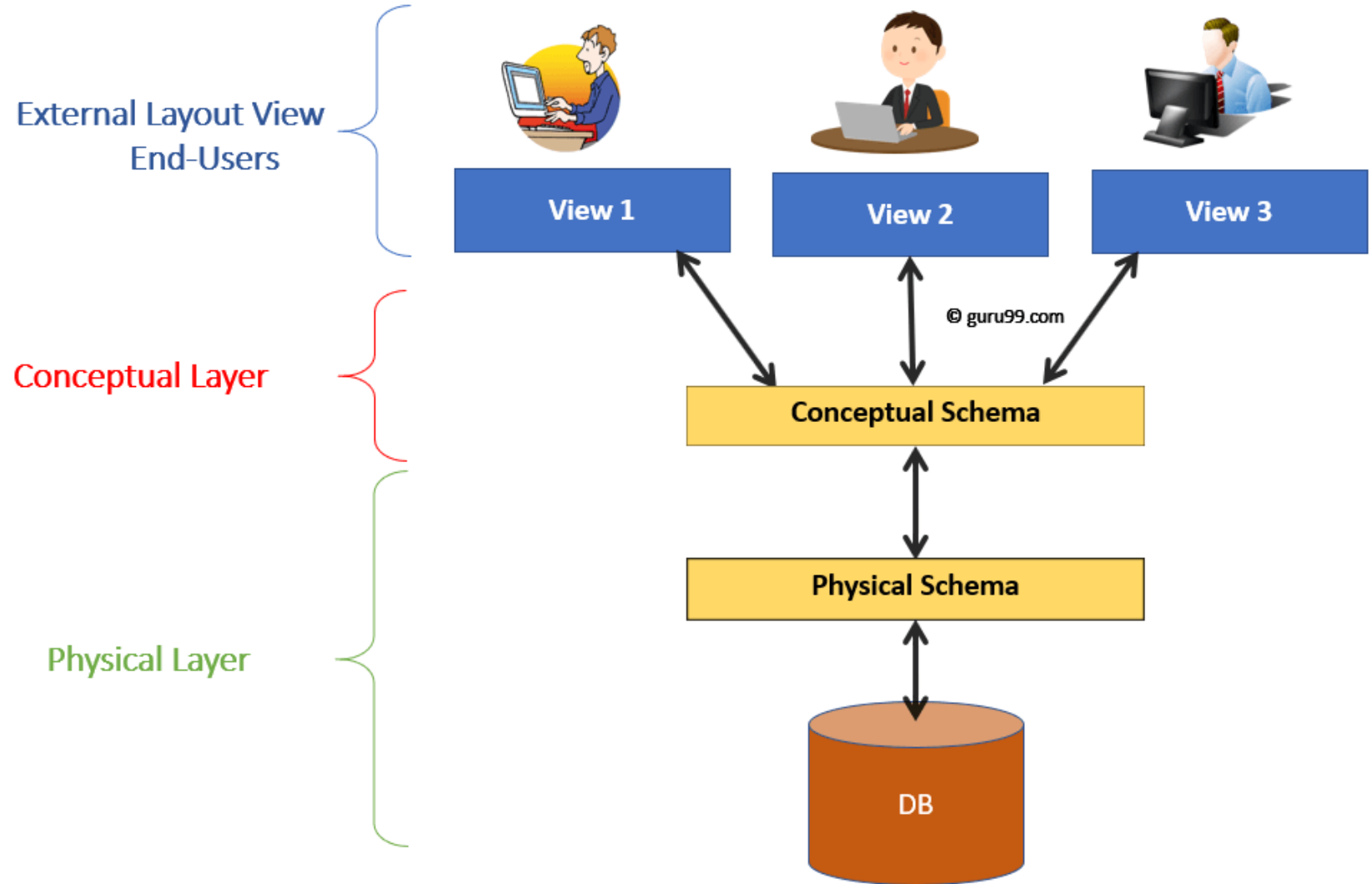


Quick overview of Keys & Entity Relationship Diagram

ERD to Relational Table – Mapping Algorithm

Recap:

Three Level ANSI-SPARC Architecture



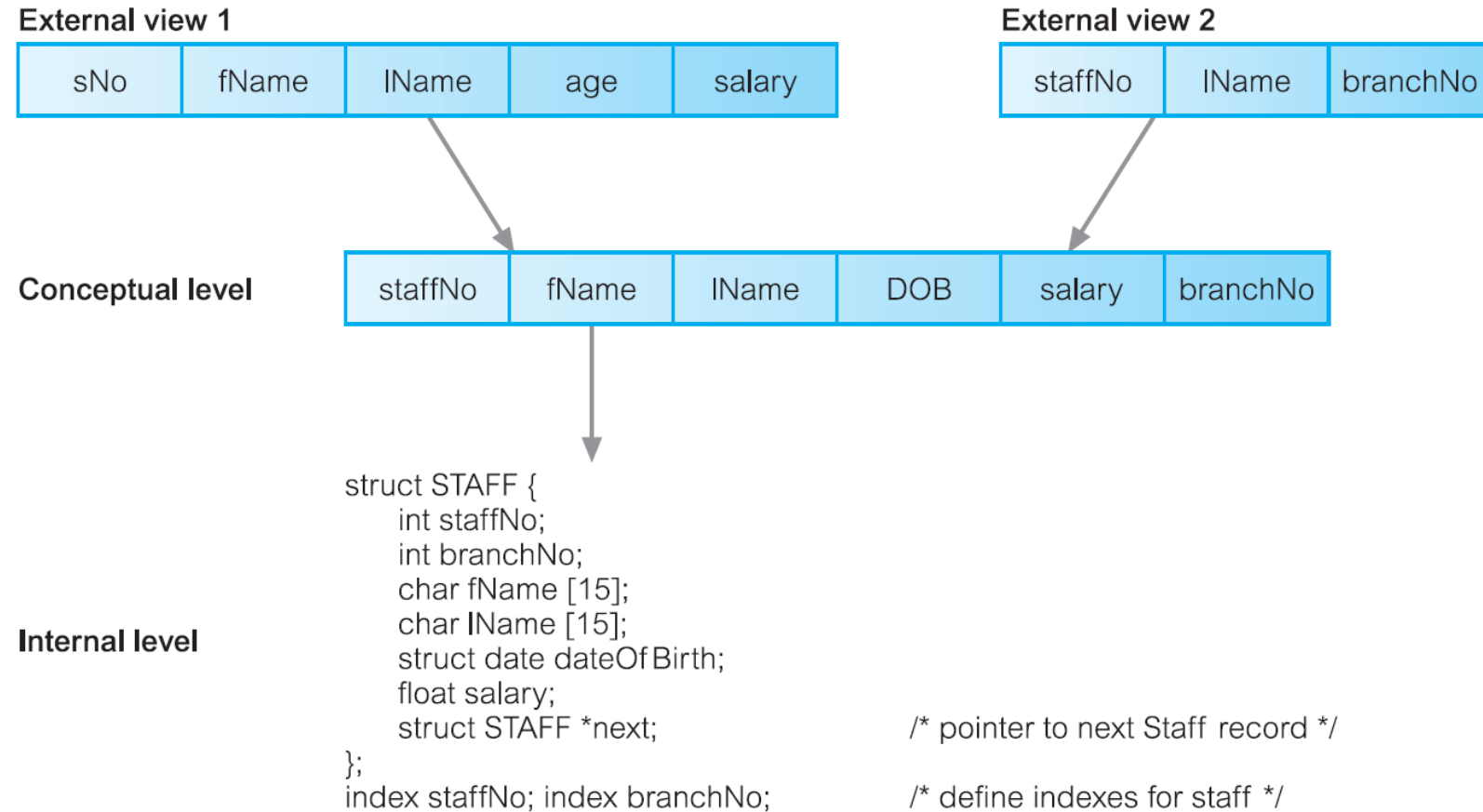


Figure 2.2 Differences between the three levels.

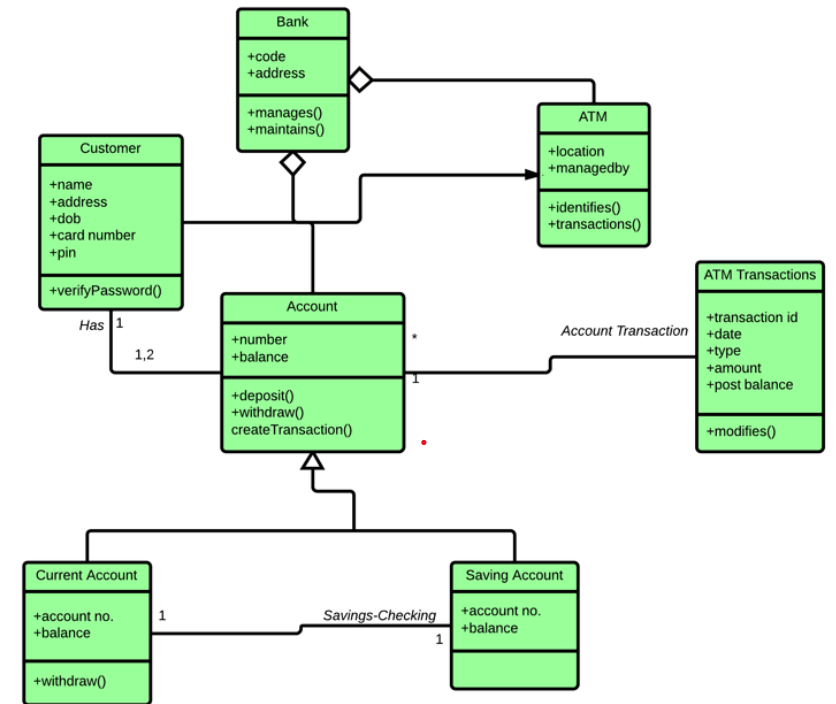
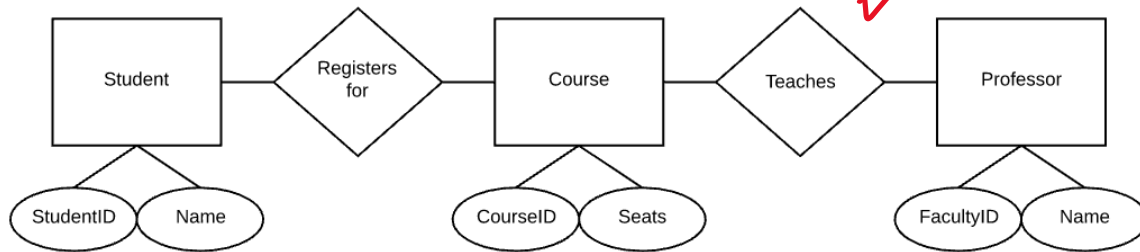
Entity-Relationship Model (ER Model):

- The ER model is a conceptual and abstract representation of the data structure within a database.
- The ER model is usually documented using natural language, text, or a combination of text and diagrams
- Example:
 - Customer entity has attributes: CustomerID (Primary Key), FirstName, LastName, Email.
 - Product entity has attributes: ProductID (Primary Key), Name, Price, and QuantityInStock.
 - The Customer entity has a FirstName attribute of data type Text, and it is required (NOT NULL).
 - A one-to-many relationship exists between Customer and Order, where each customer can place multiple orders, but each order is placed by one customer

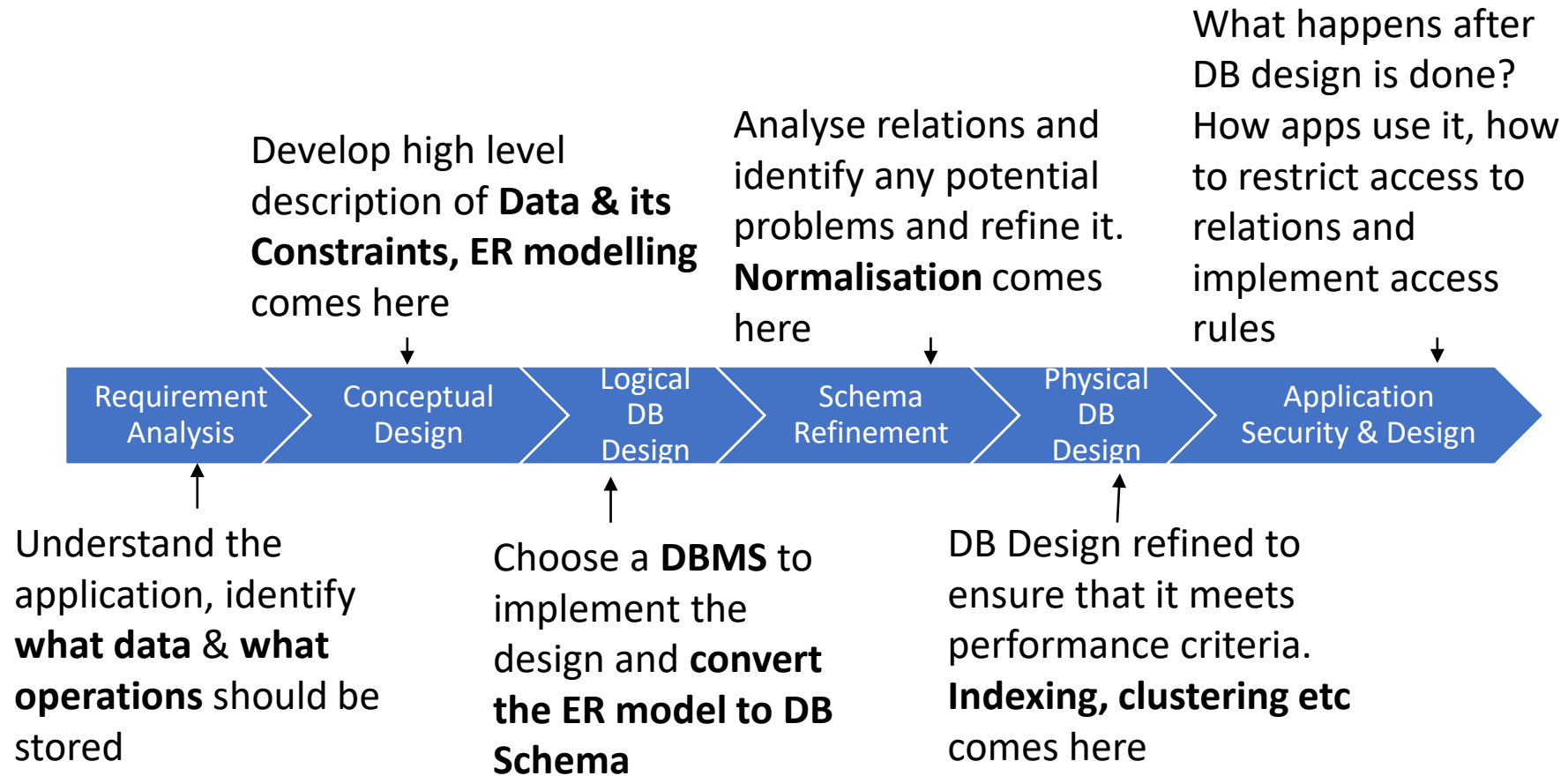


Entity-Relationship Model (ER Model):

- The ER model is usually documented using natural language, text, or a combination of text and diagrams
- Example:



Database Design Process





Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

200
1822 - 2022



<https://forms.office.com/e/H86RAAJVM6>

Man Geni Addysg Uwch yng Nghymru
Where Higher Education Began in Wales

www.uwtsd.ac.uk

Keys

Identify the following keys from this Employee table

- Candidate Key?
- Primary Key?

Employee

EMP_NI	Emp_id	Emp_name	Emp_email
41201	1	Carl	carl@email.com
98653	2	Riddle	riddle@email.com
47652	3	Mary	mark@email.com
46378	4	John	john@email.com

Keys

- Candidate Key
 - Minimal set of attributes that uniquely identifies each occurrence of an entity type.
- Primary Key
 - Candidate key selected to uniquely identify each occurrence of an entity type.

Employee

EMP_NI	Emp_id	Emp_name	<u>Emp_email</u>
41201	1	Carl	carl@email.com
98653	2	Riddle	riddle@email.com
47652	3	Mary	mark@email.com
46378	4	John	john@email.com

Keys

Identify the composite key from this Employee table

Sales

SalesPersonID	SaleAmount	SalesDate
101	£500	2023-10-23
102	£750	2023-10-23
101	£600	2023-10-24

Keys

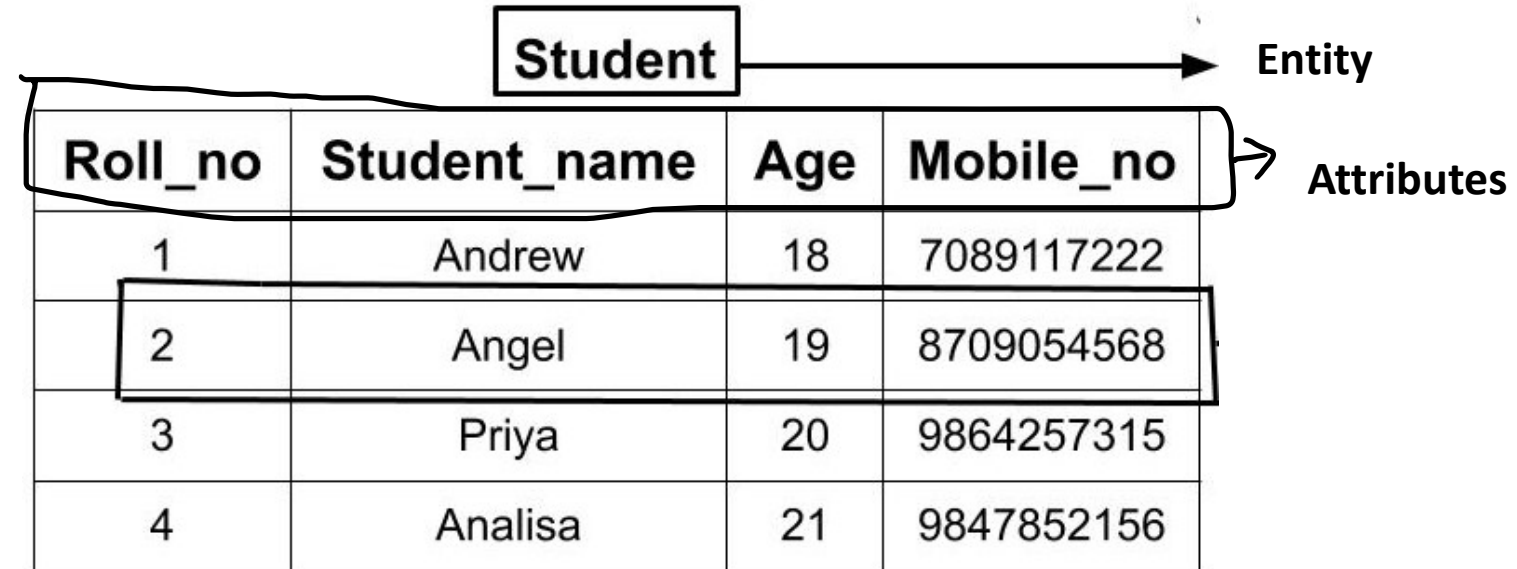
Identify the composite key from this Employee table

- Composite Key
 - A candidate key that consists of two or more attributes.

Sales

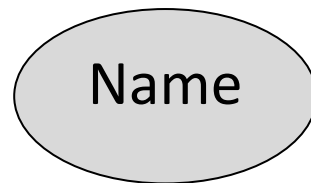
SalesPersonID	SaleAmount	SalesDate
101	£500	2023-10-23
102	£750	2023-10-23
101	£600	2023-10-24

Attributes

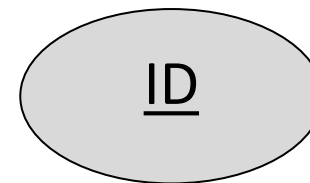


- **Attribute**
 - Property of an entity or a relationship type.
- **Attribute Domain**
 - Set of allowable values for one or more attributes.

Notation for an Attribute



Notation for an **Key** Attribute



Attributes-Type 1

- **Simple Attribute**

- Attribute composed of a single component with an independent existence.

- **Composite Attribute**

- Attribute composed of multiple components, each with an independent existence.

Entity:

Person(Firstname,Lastname,DateOfBirth,Gender,Address)

Simple Attributes:

- 1.First Name
- 2.Last Name
- 3.DateOfBirth
- 4.Gender

Composite Attribute:

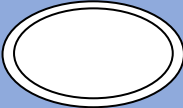
5. Address

- Street address
- State
- Postal Code

- Street address
 - House Number
 - Street Name

sub attributes

Attribute Types - 2

Type	Description	Example
Single Valued Attribute 	An attribute that holds a single value for each occurrence of an entity type.	Employee entity: Salary → can be only one value. Cannot draw more than one salary.
Multi Valued Attribute 	An attribute that holds multiple values for each occurrence of an entity type.	Employee entity: PhoneNo → A person can have one or more mobile phone number, landline etc.

Attribute Types - 3

Type	Description	Example
Stored Attribute	An attribute that holds a value provided as is.	Employee entity: Name → can be set by the user as used as is.
Derived Attribute 	An attribute that represents a value that is derivable from the value of a related attribute or set of attributes, not necessarily in the same entity type.	Employee entity: Age → can be derived from dateOfBirth and currentDate.

Exercise- ERD

Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents.

Exercise- Possible Solution

Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents.

Entities:

1.Customer:

Attributes: CustomerID (Primary Key), First Name, Last Name, Address, Phone Number, Email, etc.

2.Car:

Attributes: CarID (Primary Key), Make, Model, Year, License Plate, VIN (Vehicle Identification Number), etc.

3.Accident:

Attributes: ReportId (Primary Key), Date, Location, Description, Damage Amount, etc.

Exercise- Possible Solution

Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents.

Relationships:

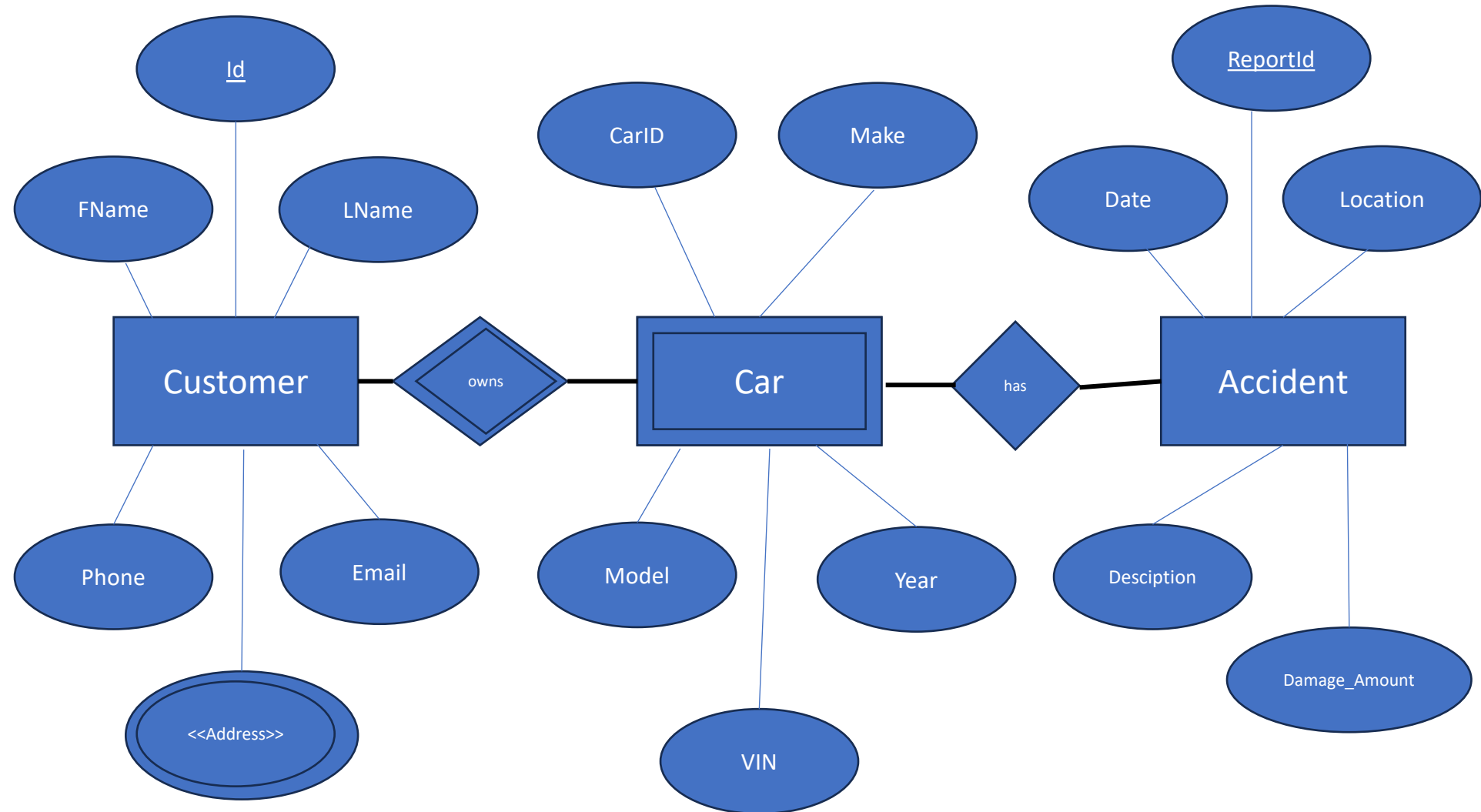
1.Owns:

1. Relationship between Customer and Car.
2. One customer can own one or more cars.
3. One car is owned by one and only one customer.
4. Cardinality: One (Customer) to Many (Car).

2. Has:

1. Relationship between Car and Accident.
2. One car can be associated with zero to many accidents.
3. One accident is associated with one and only one car.
4. Cardinality: Zero or Many (Car) to One (Accident).

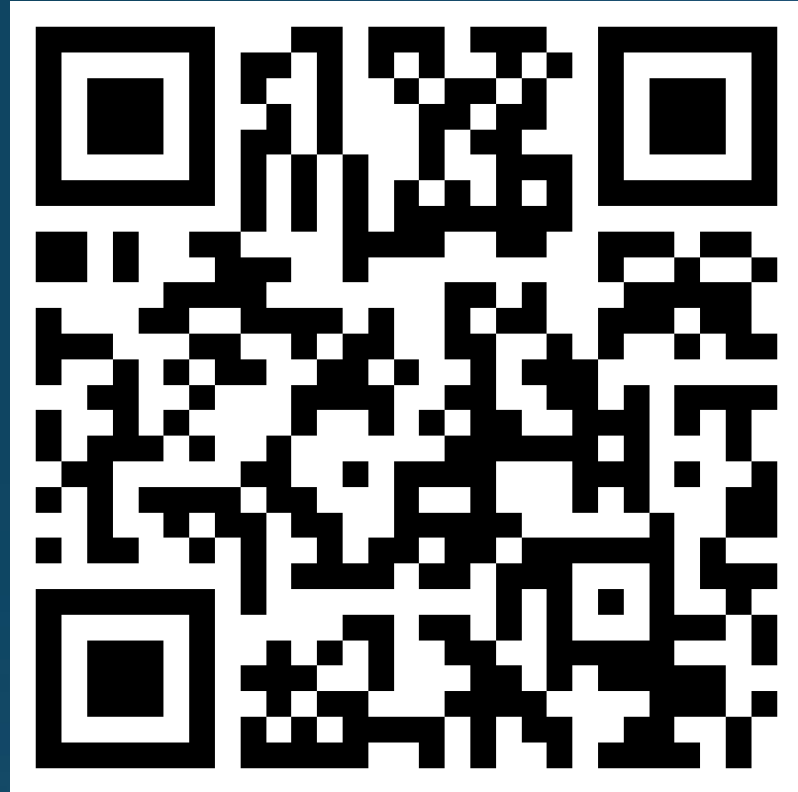
Exercise- Possible Solution





Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

200
1822 - 2022



Recap Quiz for ERD

Man Geni Addysg Uwch yng Nghymru
Where Higher Education Began in Wales

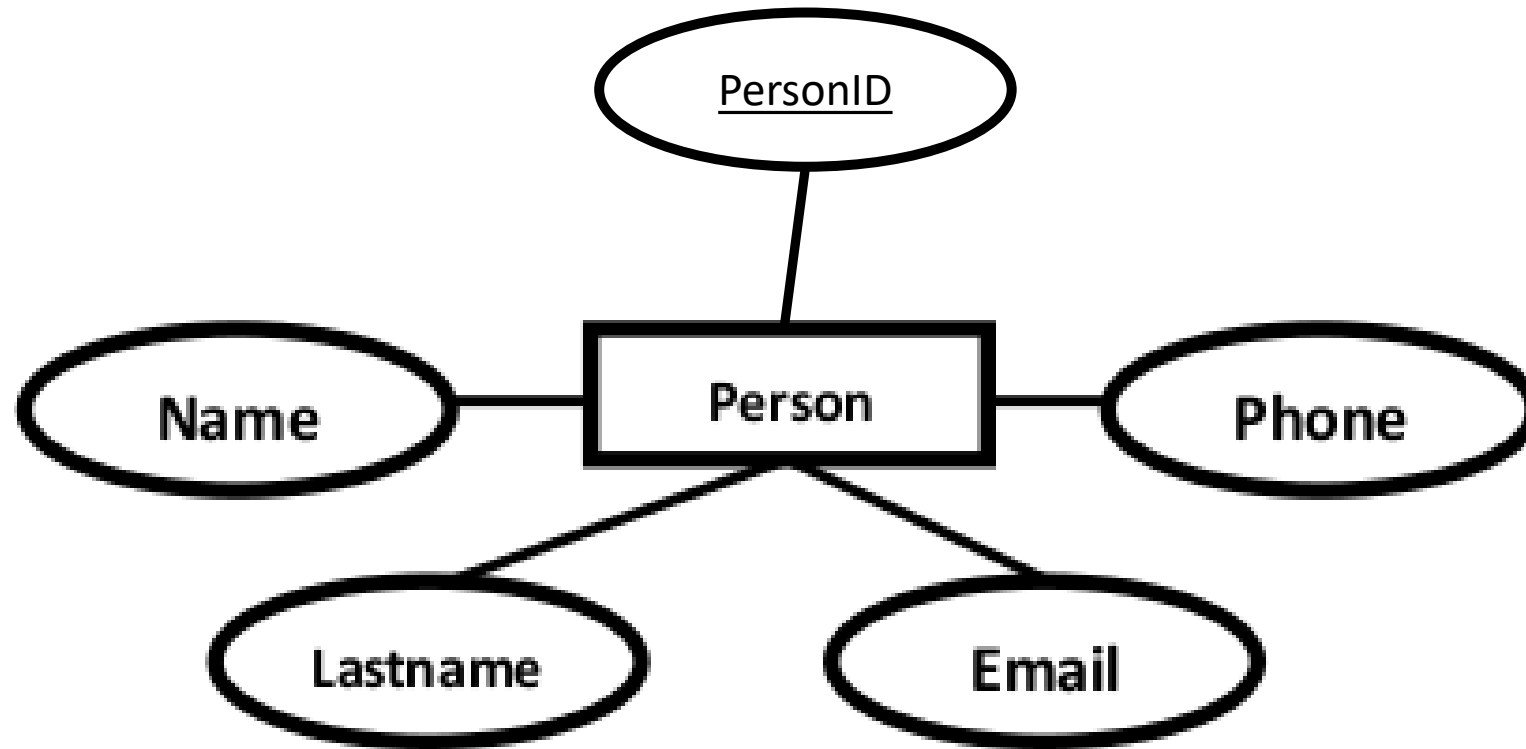
www.uwtsd.ac.uk

ERD to Relational Table – Mapping Algorithm

Mapping Algorithm - Entities and Simple Attributes

Person(personid , name, lastname, email)

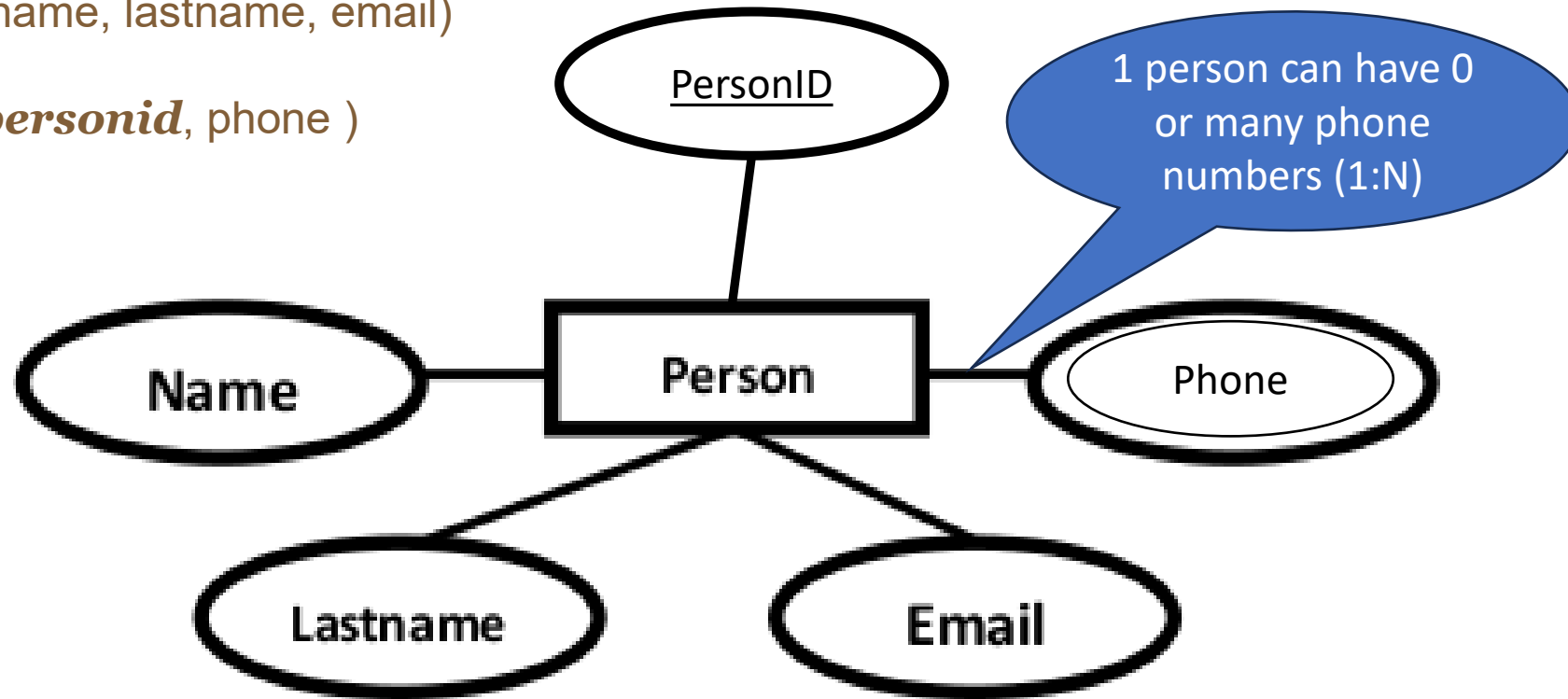
What happens to Phone-multivalued attribute?



Mapping Algorithm – Multivalued attribute

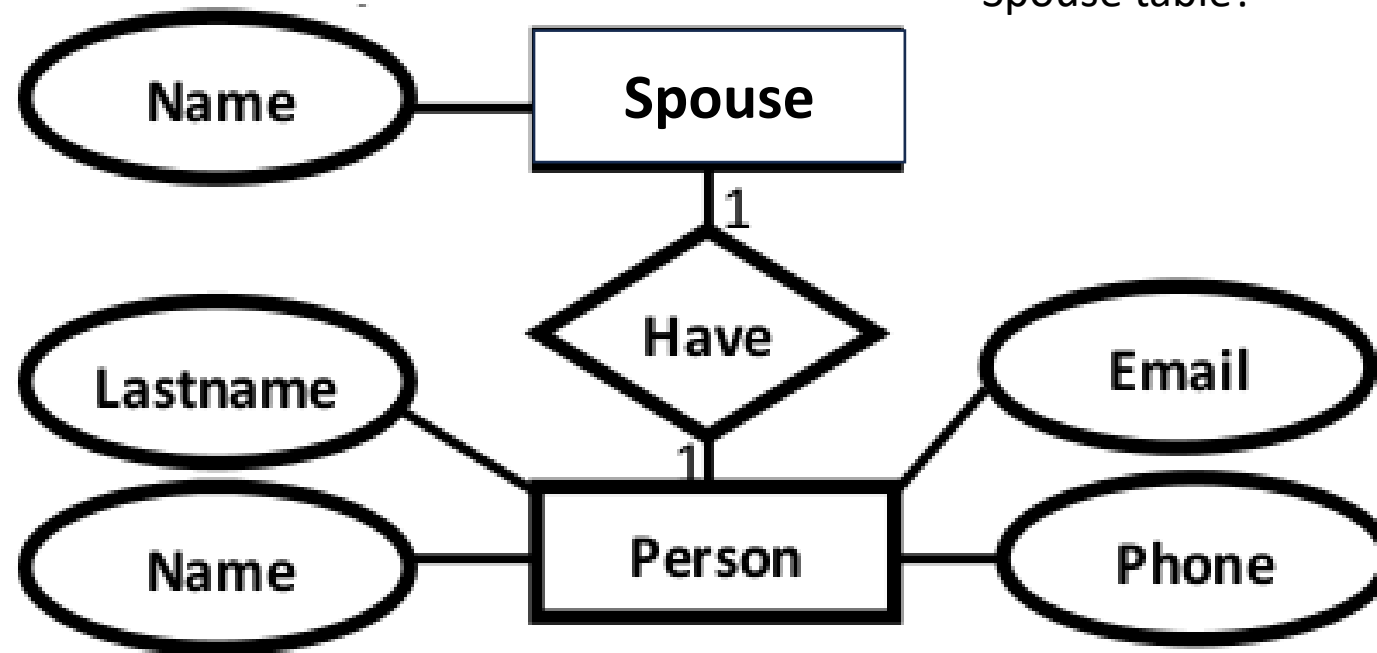
Person(personid , name, lastname, email)

Phone (phoneid , **personid**, phone)

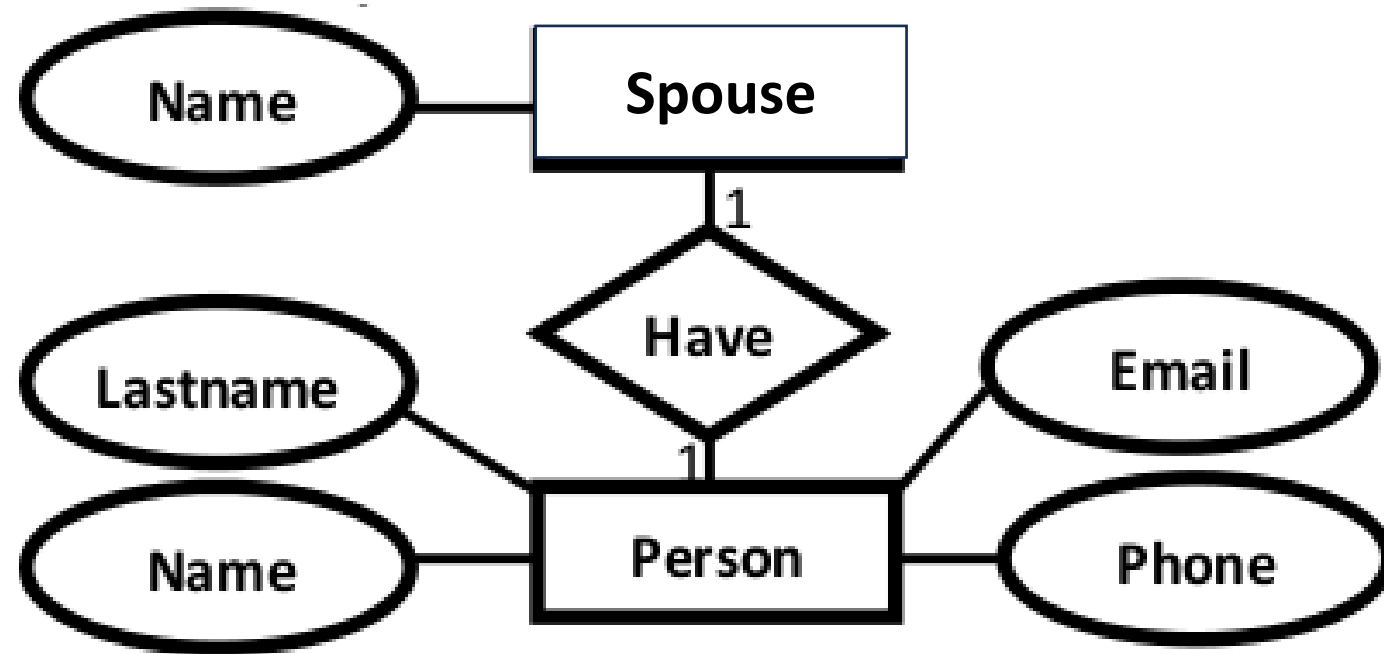


Mapping Algorithm – 1:1 Relationships

which will be the foreign key creating a reference between Persons and Spouse table?



Mapping Algorithm – 1:1 Relationships

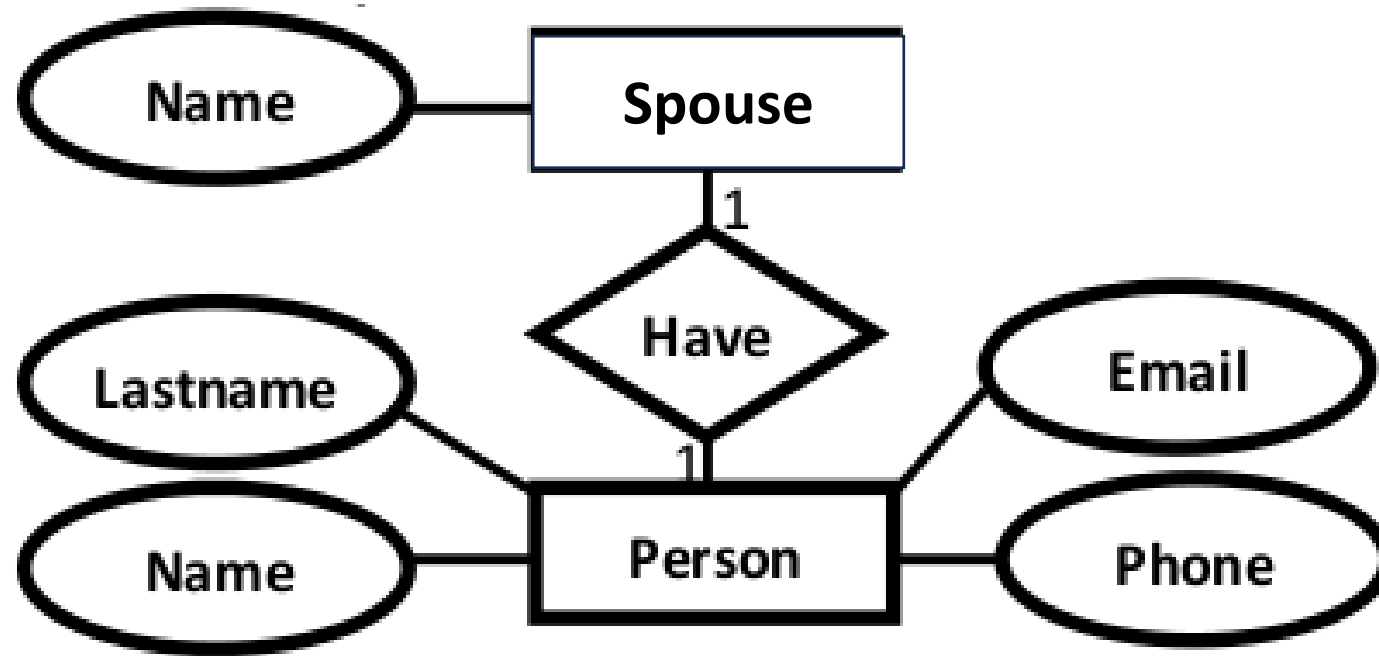


Person(personid{PK}, name, lastname, email, spouseid{FK})

Spouse (spouseid{PK}, **name**)

Can you think of an alternative approach?

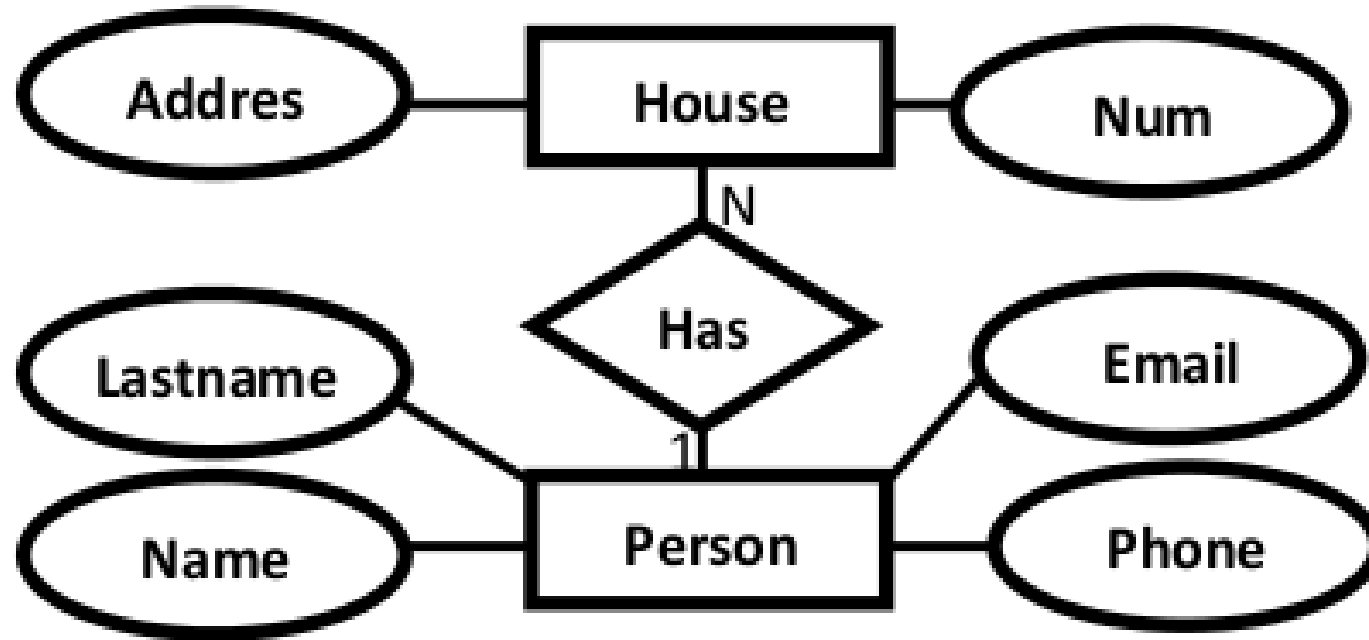
Mapping Algorithm – 1:1 Relationships



Persons(personid{PK}, name, lastname, email)

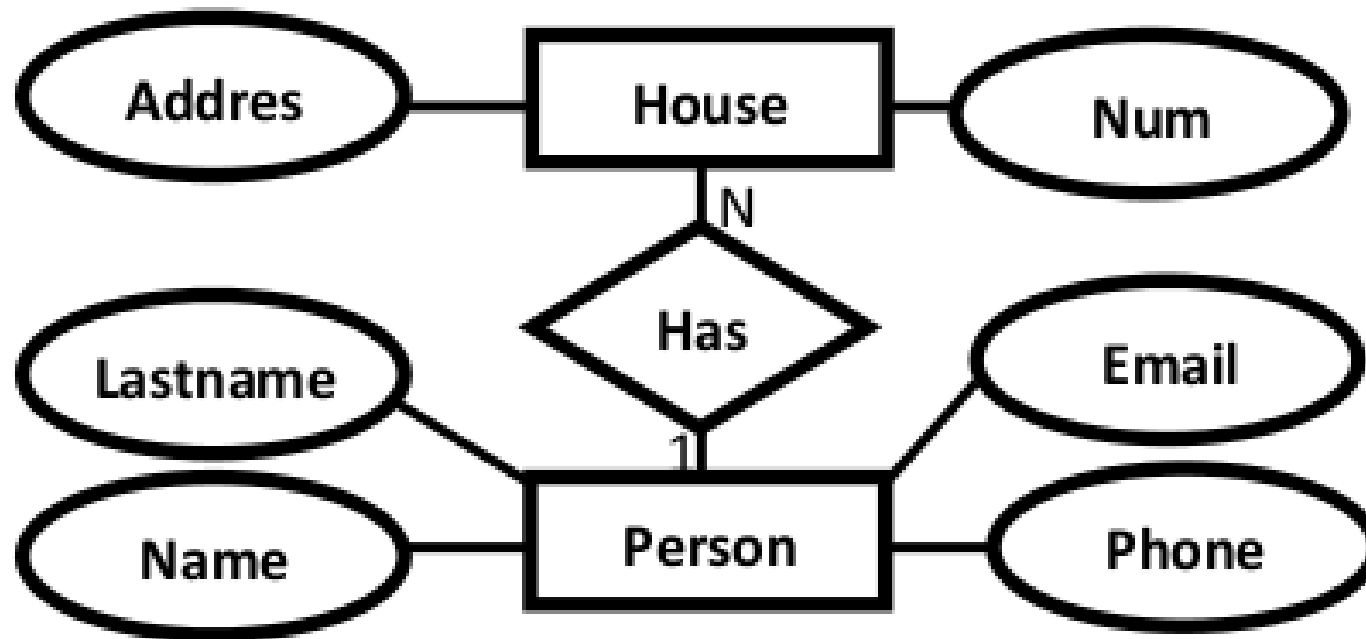
Spouse (spouseid{PK}, **name**, **personid**{FK})

Mapping Algorithm – 1:N Relationships



How is this converted to?
Give a try...

Mapping Algorithm – 1:N Relationships



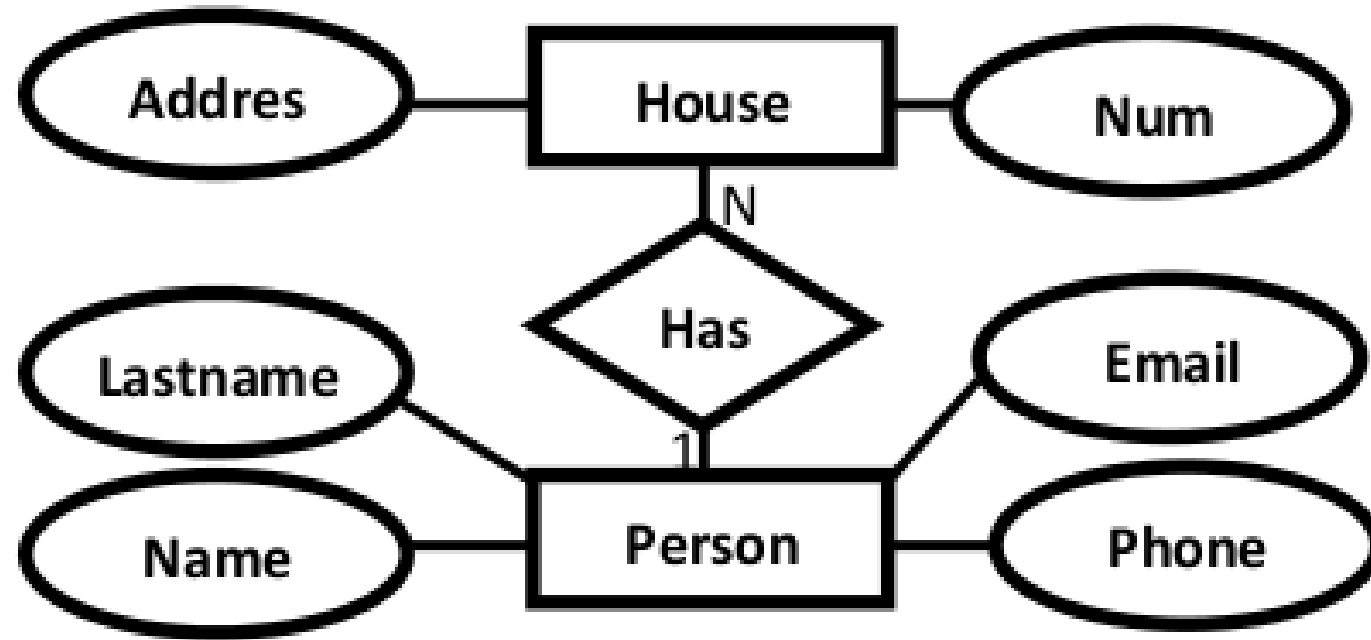
House

houseid	personid
H1	1
H2	2
H3	1

person

personid	houseid
1	H1,H3
2	H2
3	<null>

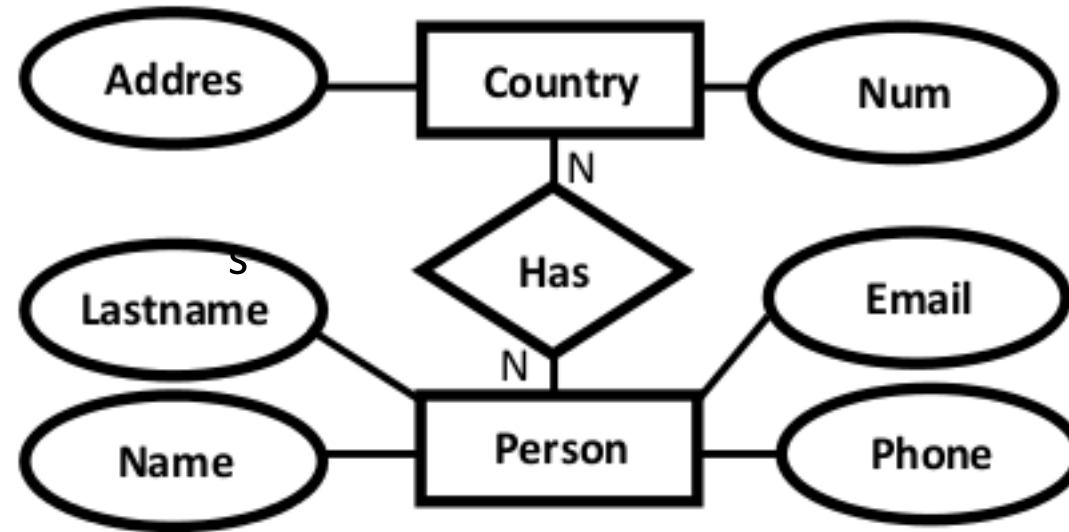
Mapping Algorithm – 1:N Relationships



Person(personid , name, lastname, email)

House (houseid , num , address, *personid*)

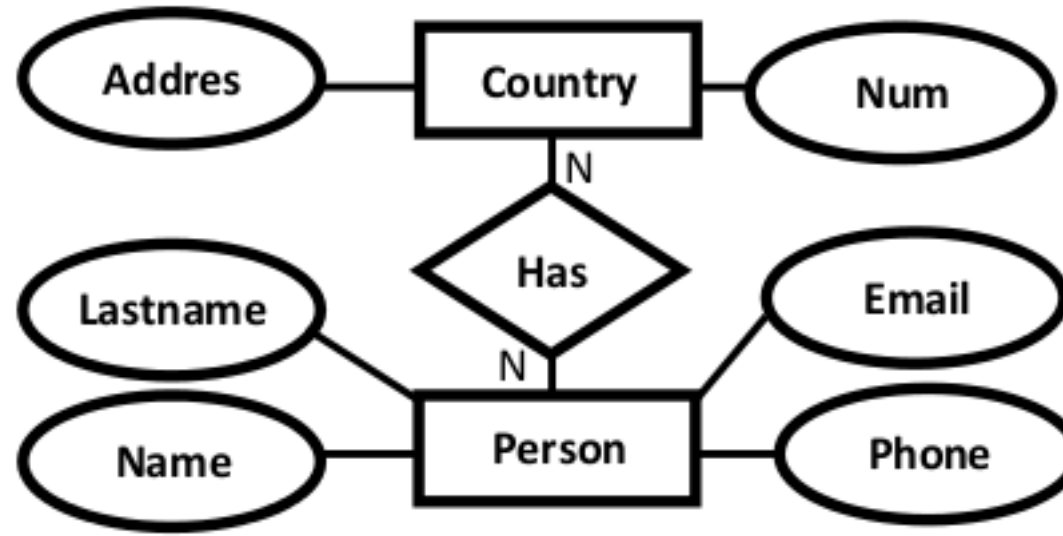
Mapping Algorithm – N:N Relationships



How is this converted?

Give a try...

Mapping Algorithm – N:N Relationships



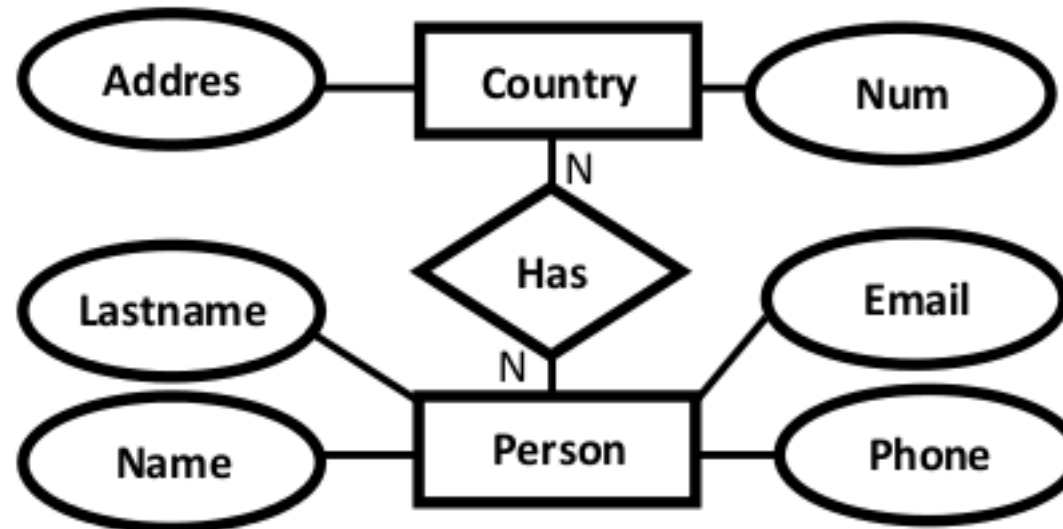
Person

Country

personid	countryid
p1	c1
p2	c1
p3	c1,c2

countryid	personid
c1	p1,p2,p3
c2	p3
c3	-

Mapping Algorithm – N:N Relationships



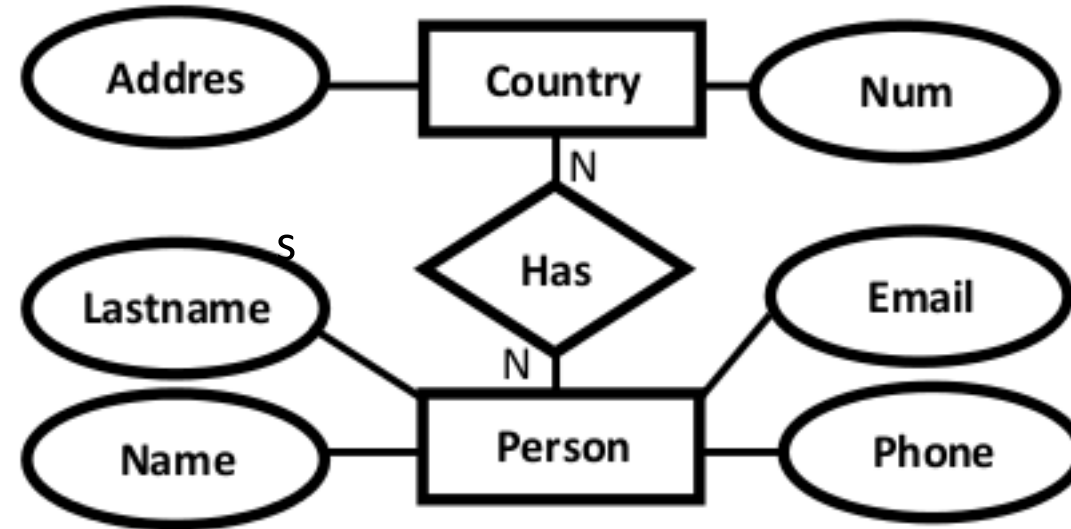
countryid	personid
c1	p1,p2,p3
c2	p3
c3	-

personid	countryid
p1	c1
p2	c1
p3	c1,c2

hasrelationid	personid	countryid
has1	P1	C1
has2	P2	C1
has3	p3	C1
has4	p3	c2

Composite
key

Mapping Algorithm – N:N Relationships

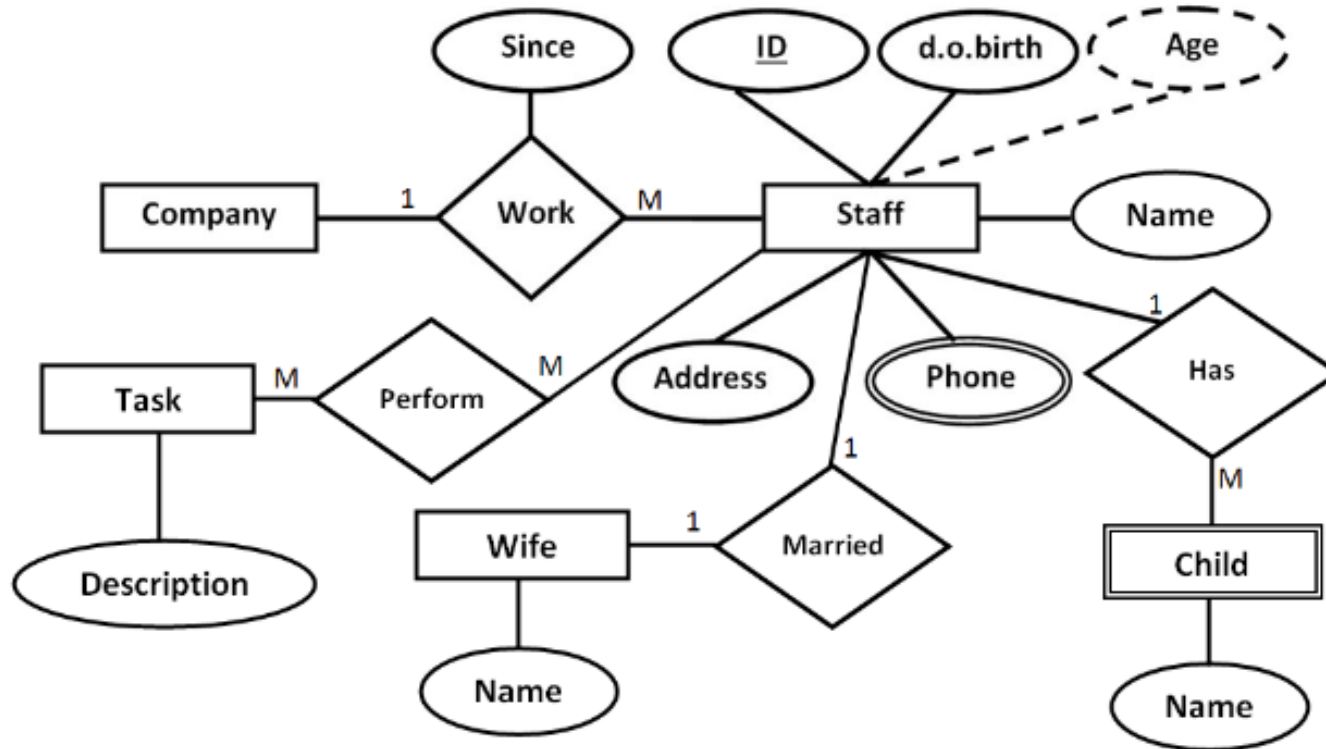


Persons(personid , name, lastname, email)

Countries (countryid , name, code)

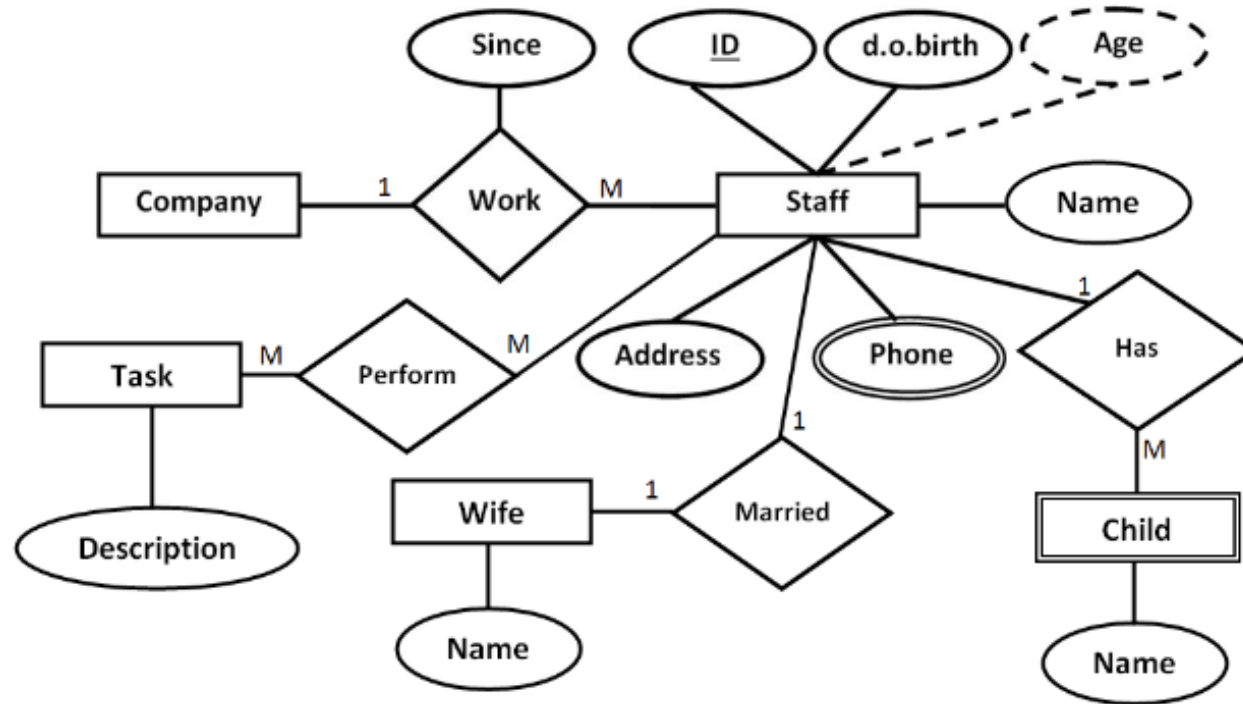
HasRelation (hasrelationid , personid , countryid)

Mapping Algorithm – Activity 1



<https://forms.office.com/e/3Dzw97CRvA>

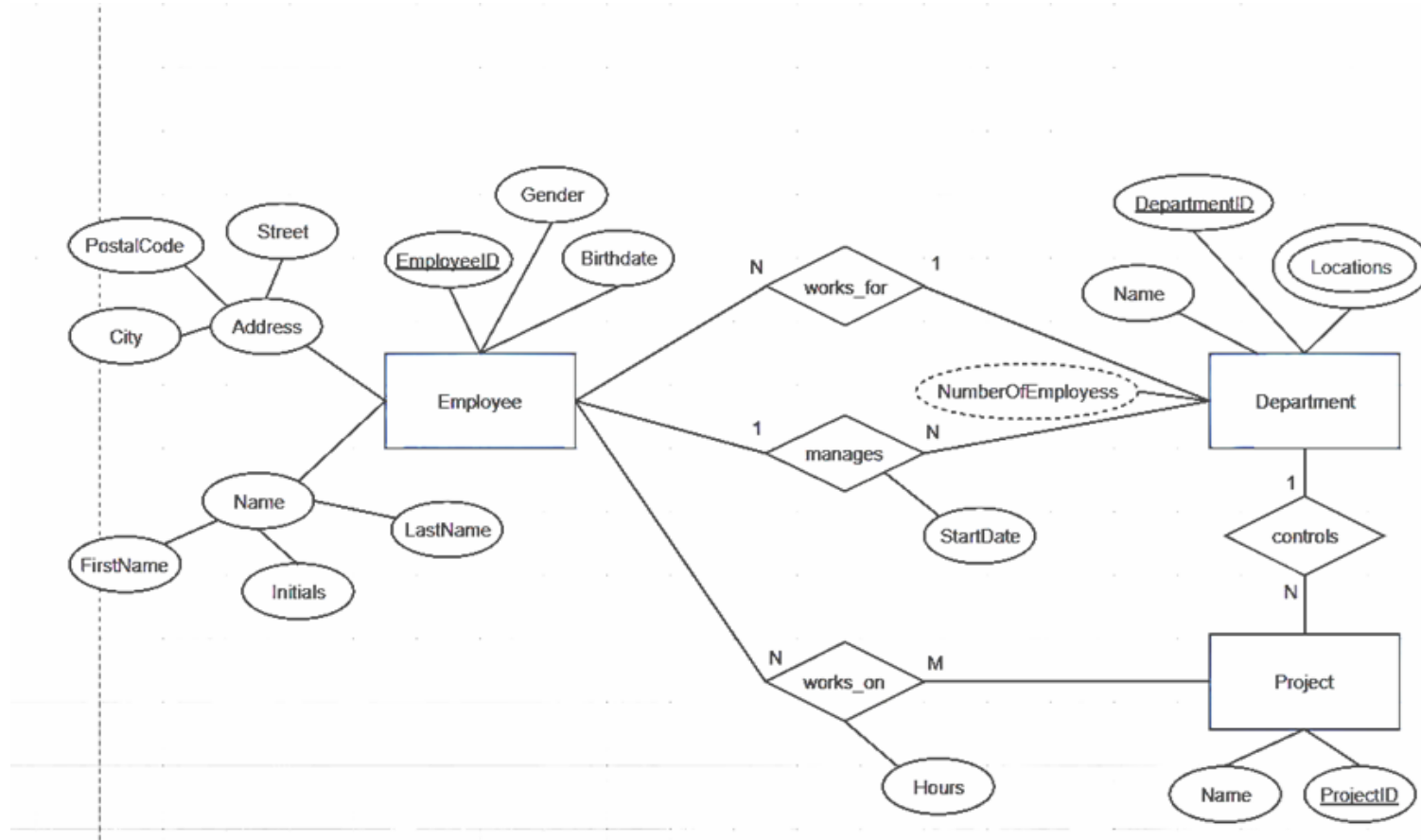
Mapping Algorithm – Activity 1



Company(CompanyID , name , address)
Staff(StaffID , dob , address , **WifeID**)
Child(ChildID , name , **StaffID**)
Wife (WifeID , name)
Phone(PhoneID , phoneNumber , **StaffID**)
Task (TaskID , description)
Work(WorkID , **CompanyID** , **StaffID** , since)
Perform(PerformID , **StaffID** , **TaskID**)

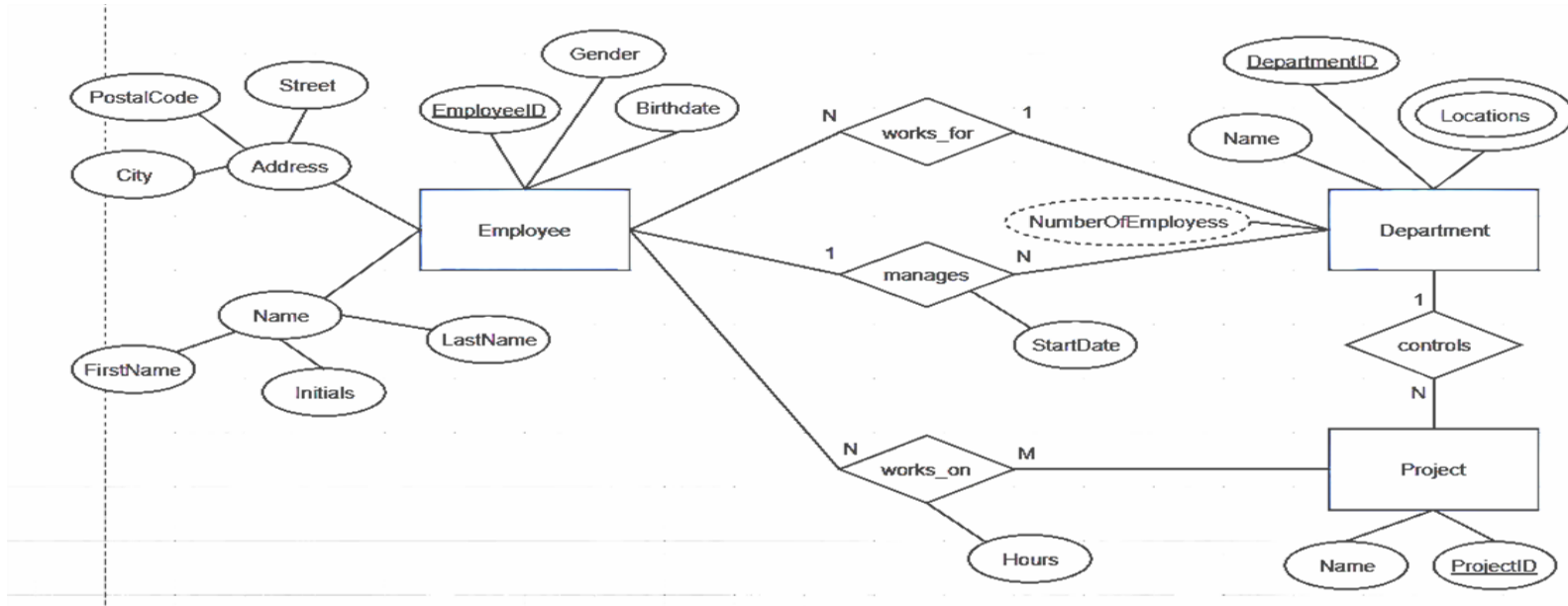
<https://forms.office.com/e/3Dzw97CRvA>

Mapping Algorithm – Activity 2



<https://forms.office.com/e/nZnGcYpCEn>

Mapping Algorithm – Activity 2



Employee(Employeeid,gender,birthdate,street,postalcode,city,Firstname,Initials,Lastname,*Departmentid*)

Department(departmentID,name, *LocationID*,*employeeid*)

Location(LocationID,locationname)

Project(Projectid,name,*departmentid*)

Works_on(hours, *employeeid*, *projectid*)

ERD to Relational Tables: Mapping Algorithm

Correspondence between ER and Relational Models

ER MODEL	RELATIONAL MODEL
Entity type	<i>Entity</i> relation
1:1 or 1:N relationship type	Foreign key (or <i>relationship</i> relation)
M:N relationship type	<i>Relationship</i> relation and <i>two</i> foreign keys
<i>n</i> -ary relationship type	<i>Relationship</i> relation and <i>n</i> foreign keys
Simple attribute	Attribute
Composite attribute	Set of simple component attributes
Multivalued attribute	Relation and foreign key
Value set	Domain
Key attribute	Primary (or secondary) key

Properties of Database Relations

- Each relation name is distinct from all other relation names in the relational schema.
- Each attribute has a distinct name.
- Each cell of a relation contains exactly one atomic (single) value.
 - Values of an attribute all come 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.

Next Session.....

Normalization