

# 7BUIS030W

## Data System Concepts and Fundamentals

Lecture -3




## Lecture-3 Outline

Conceptual database modelling,  
conceptual database model  
design methodology, conceptual  
ERDs , Cardinality, Participation,  
Multiplicity, Attributes , Keys

# Conceptual Database Model

- Conceptual ERD models the business objects that should exist in a system and the relationships between them.
- A conceptual model is developed to present an overall picture of the system by recognizing the business objects involved.
- Conceptual model defines what entities exist, not which tables.



# Conceptual database model design methodology

## Design methodology:

A structured approach that uses procedures, techniques, tools and documentation that aids to support and facilitate the process of design

## Conceptual database design

The process of constructing a model of the data used in an enterprise independent of all physical constraints.

## A conceptual data model comprises:

**Entity types**

**Relationship types**

**Attributes and  
attribute domains**

**Primary keys and  
alternate keys**

**Integrity  
constraints**

# Conceptual database design-

## Identify the entity types

- First step in building the database model
- Determines and defines the main objects that the users are interested in.
- One way of identifying is through user's specifications
- Look for major objects and nouns and noun phrases that are mentioned which qualifies the objects
- Eg:
  - nouns and noun phrases like staff number, staff name can all be grouped under an entity *Staff*
  - Authors, book name all represents the entity *book*



## Conceptual database design-

# Identify the relationship types

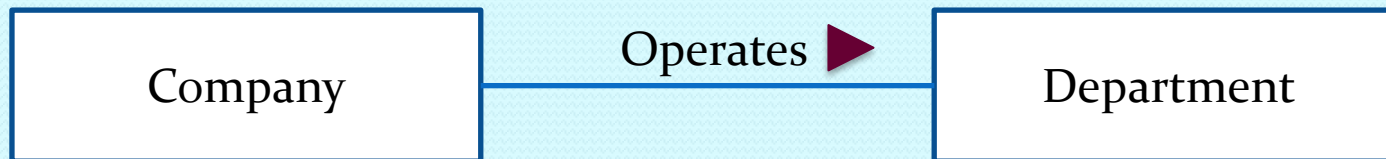
- ❑ To identify the important relationships that exist between the entity types
- ❑ Typically relationships are indicated by verbs or verbal expressions
- ❑ In most instances, the relationships are **binary**, i.e the relationship exist between two entities
- ❑ Ensure that all the explicit and implicit relationships in the user requirements are detected
- ❑ Entity Relationship Diagrams (**ERD**) are the best way to represent entities and how they relate to one another

# Examples of Entity and relationships

- staff working in a branch



- A company runs departments



- Department has employees



# Multiplicity constraints of relationship types

Once the relationships are identified the next crucial step is to determine the **multiplicity** of each relationship

Multiplicity is the number of possible occurrences of an entity type that may relate to single occurrence of an associated entity type through a particular relationship.

Eg: Each company operates four departments and each department belongs to one company

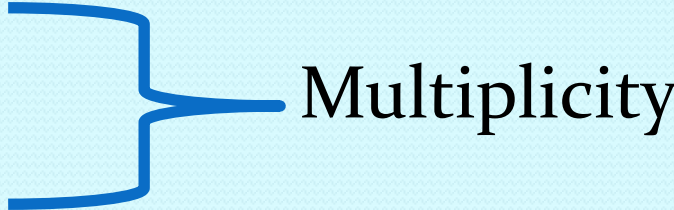
1 company operates 4 departments

1 department belongs to 1 company



# Multiplicity constraints of relationship types

Multiplicity consists of two separate constraints :

- Cardinality
  - Participation
- 
- Multiplicity

Cardinality: Describes the **maximum** number of possible occurrences for an entity participating in a given relationship

Participation: Determines whether all or some entity occurrences participate in a relationship i.e. the **minimum** number of occurrences



# Multiplicity constraints of relationship types

Eg: Staff manages branch

To establish the participation ask the following questions:

What is the minimum number of branches that one staff can manage?

Answer: 0 The staff is new and he is not assigned any branch

What is the minimum number of staff who can manage one branch?

Answer : 1 One branch should be managed by at least one staff



# Multiplicity constraints of relationship types

Eg: Staff manages branch

To establish the cardinality ask the following questions:

What is the maximum number of branches that one staff can manage?

Answer: 1. Each staff is assigned a maximum of one branch not more than that

What is the maximum number of staff who can manage one branch?

Answer : 1 One branch should be managed by no more than one staff

# Multiplicity constraints of relationship types

Eg: Staff manages branch

One **staff** manages a maximum of one branch

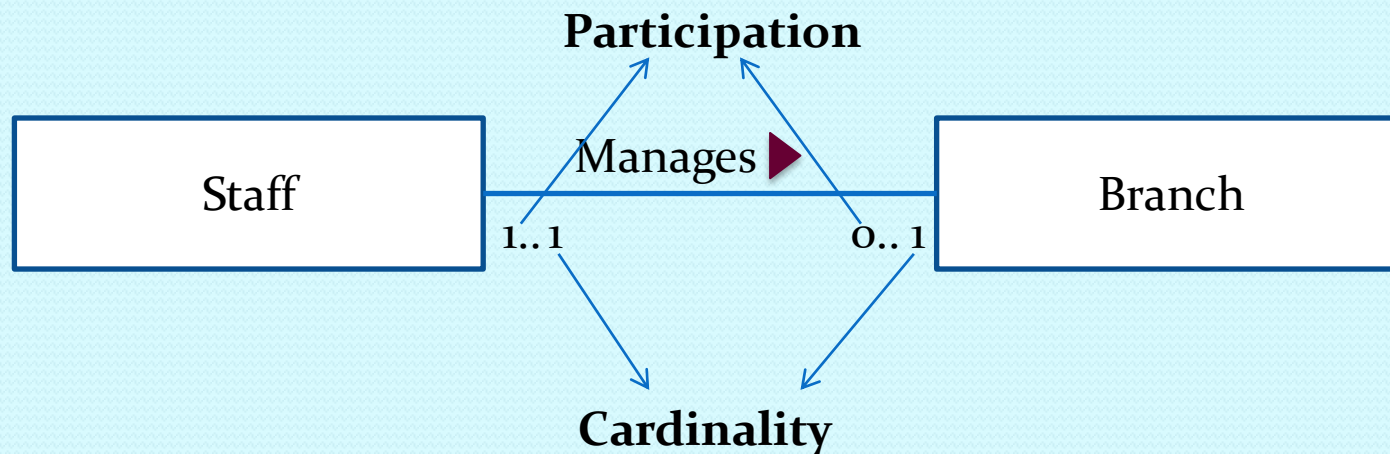
One **branch** is managed by maximum one staff

One **staff** manages no branches at all

One **branch** is managed by at least one staff

Cardinality

Participation



# Multiplicity constraints of relationship types

Eg:

Manager of a company manages staff



Newspaper advertise properties for rent



# Multiplicity constraints of relationship types

## Representation of multiplicity constraints

0..1	Zero or one occurrence
1..1 or just 1	Exactly one entity occurrence
0..* or just *	Zero or many occurrences
1..*	One or many occurrences

# Multiplicity constraints of relationship types

## Exercise-1

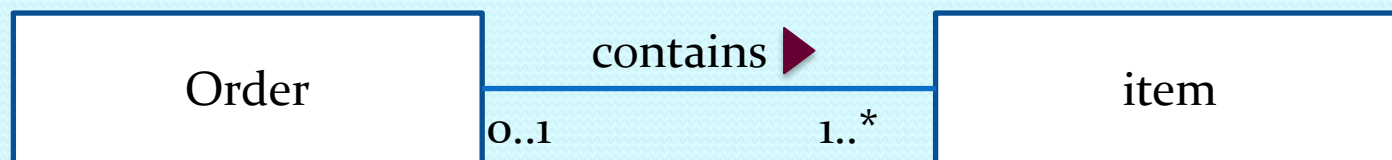
1. Supermarkets sell products



2. Customer places order



3. Customer order contains items



# Identifying Attributes on Relationships

- Attributes are properties of an entity or a relationship type.
- Attributes hold values that describe each entity occurrence and represent the main part of the data stored in the database.
- **Attribute domain:** Set of allowable values for one or more attributes. The domain defines the potential values that an attribute may hold . Eg: character , character string
- Set of attributes may share a domain:
- Eg: address attributes of the branch entity shares the same domain as privateOwner, businessOwner entity types



# Attributes on Relationships

## Simple attribute:

- An attribute composed of a single component with an independent existence
- Simple attributes cannot be further subdivided into smaller component

Eg:

- Staff entity has simple attributes such as *StaffPosition*, *salary* .

# Attributes on Relationships

## Composite attribute:

- An attribute composed of multiple components, each with an independent existence
- Some attributes can be further subdivided into smaller component

Eg:

- Branch entity has *address* attribute that can be subdivided into *street name* , *postcode* attributes

# Attributes on relationships

Identify the simple and composite attributes of the following entities

Staff

*staffNo, name, position*

Simple attribute

Composite- fName, sName

Branch

*branchNo, address*

Simple attribute

Composite: street, city, postcode

# Determine the keys

## Candidate key

The minimal set of attributes that uniquely identifies each occurrence of entity type

Eg:

The branchNo attribute is the candidate key for the entity Branch

StudentId is the candidate key for the entity student



# Determine the keys

## Primary Key:

The candidate key that is selected to uniquely identify each occurrence of an entity type.

A candidate key is a minimal set of attributes of an entity that uniquely identifies each occurrence of that entity.

In the case of more than one candidate key, one of them should be chosen as the primary key

Eg: Entity staff has candidate keys staffId and staffName. staffName can be shared by multiple records. Hence the unique identifier can be the staffId which is the primary key for the entity staff.



# Determine the keys

## **Criteria to choose Primary key:**

- The candidate key with minimal set of attributes
- The candidate key that is least likely to have its value changed
- The candidate key that is easiest to use from the user's point of view

# Determine the keys

## **Alternate key:**

The candidate key other than the primary key

## **Composite key:**

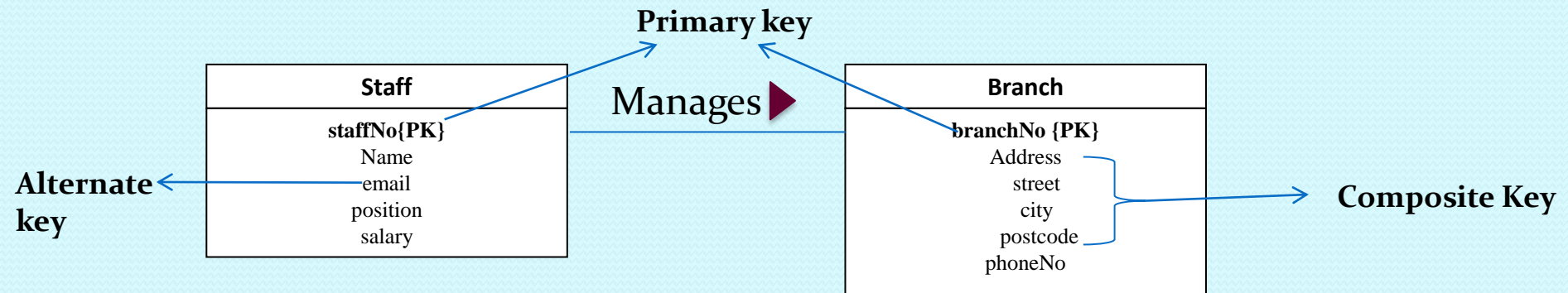
The candidate key that consists of two or more attributes

Eg: property advert in multiple newspapers.

The entity advert has a composite primary key made up of propertyNo, newspaperName, dateAdvert to identify each occurrence of the advert

# Diagrammatic representation of attributes and keys

## Examples of the keys and attributes





# Diagrammatic representation of attributes and keys

**Exercise:** Explain the multiplicity and Identify the primary key, composite key and alternate keys in the following ERD

