

# INFO20003 Tutorial – Week 4 Solutions

## Objectives:

This tutorial will cover:

- I. Additional concepts in ER modelling, plus a simple case study – 20 mins
- II. Bus company case study – conceptual and logical modelling – 30 mins

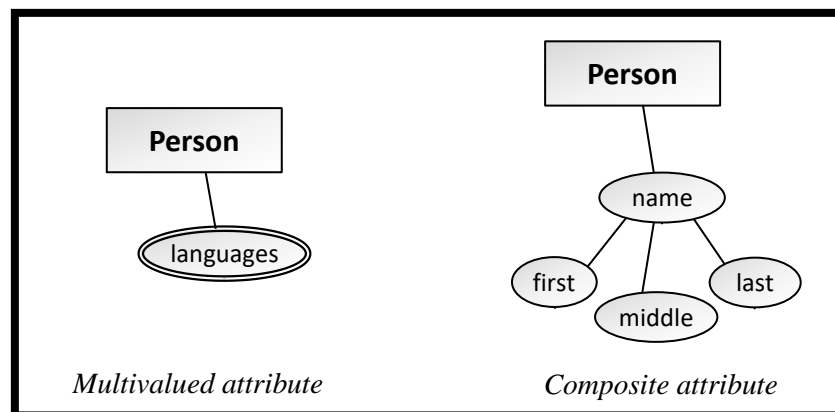
## Exercises:

### 1. ER modelling – additional concepts

- Multivalued and composite attributes

Multivalued attributes can have more than one value at the same time, such as phone numbers, skills, languages spoken, etc. These attributes are drawn using a double outline.

Composite attributes have multiple components and can be broken down into multiple attributes, such as a name that can be stored as first name and last name. These attributes are drawn by branching the sub-attributes off the composite attribute.



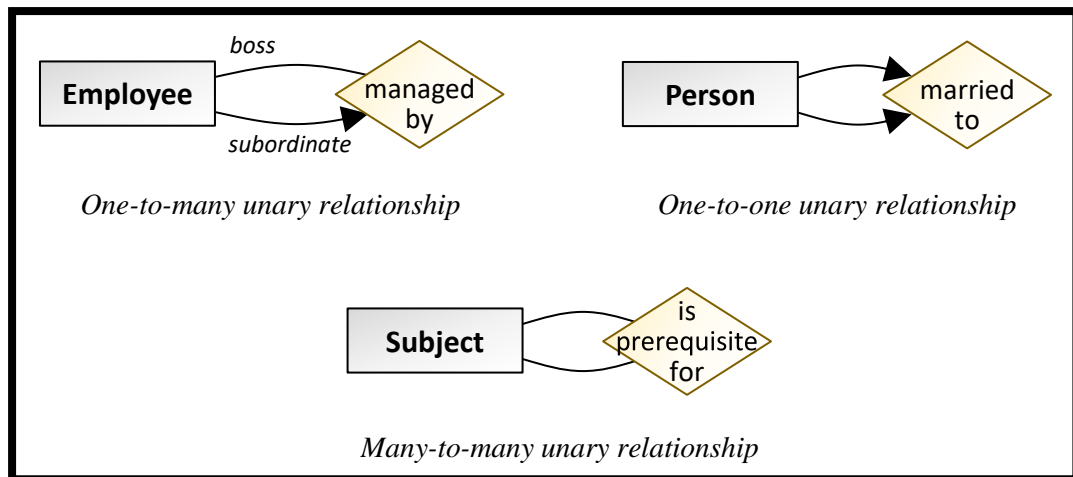
- Unary relationships

A unary relationship is a relationship between an entity and itself, that is, it allows relationships to be defined between multiple instances of the same entity. Examples of unary relationships include:

- the “managed by” relationship between an employee and their boss (who is another employee of the company);
- the “married to” relationship between a person and their spouse;
- the “prerequisite” relationship between university subjects.

Just like ordinary binary relationships, unary relationships have different cardinalities and constraints. “Managed by” would be a one-to-many relationship, “married to” is one-to-one, and “prerequisite” is many-to-many.

When the two sides of the relationship have different constraints, it is important to label the two ends of the unary relationship to make it clear what the constraints apply to.



## 2. Practising these concepts:

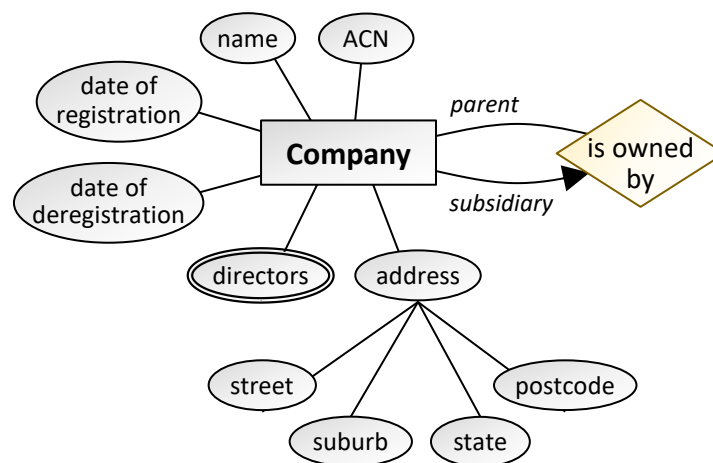
Australia's corporate regulator, ASIC, stores a range of information about every Australian company, including the name, the nine-digit ACN (Australian Company Number), the date of registration and deregistration, and the names of the company's directors. Every company has a registered address, made up of the street address, suburb, state and postcode. A company may be owned by another company; in this situation ASIC keeps track of the company's parent company.

Use this information to model a "company" entity using Chen's notation.

There are some obvious attributes – name, ACN, date of registration, date of deregistration. The company could have one or more directors, so the directors' names could be stored as a multivalued attribute. The address is composed of several clearly defined parts (street address, suburb, state, postcode) so we could store it as a composite attribute.

The relationship between a company and its parent company is a unary relationship. Each company may have a parent company (partial participation, "one"), and every company may own zero or more companies (partial participation, "many").

The final entity should look something like this:



### 3. Consider the following case study:

A bus company owns a number of buses. Each bus is allocated to a particular route, although some routes may have several buses. Each route passes through a number of towns. One or more drivers are allocated to each stage of a route, which corresponds to a journey through some or all of the towns on a route. Some of the towns have a depot where buses are kept – each bus always returns to its allocated depot at the end of the day.

Each of the buses is identified by its registration number and can carry different numbers of passengers, since the vehicles vary in size and can be single or double-decked. Each route is identified by a route number and information is available on the average number of passengers carried per day for each route. Drivers have an employee number, name, address, and sometimes a telephone number, and the names of the training courses they have completed need to be stored.

#### a. Identify the entities.

The following entities are identified from this case study:

- Bus ← Each bus is allocated to a particular route
- Route ← Each route passes through a number of towns
- Stage ← One or more drivers are allocated to each stage of a route
- Town ← Some of the towns have a depot where buses are kept
- Depot ← Some of the towns have a depot where buses are kept
- Driver ← One or more drivers are allocated to each stage of a route

#### b. Identify the relationships (use business rules to identify relationships). State all the key constraints and participation constraints.

##### Key constraints:

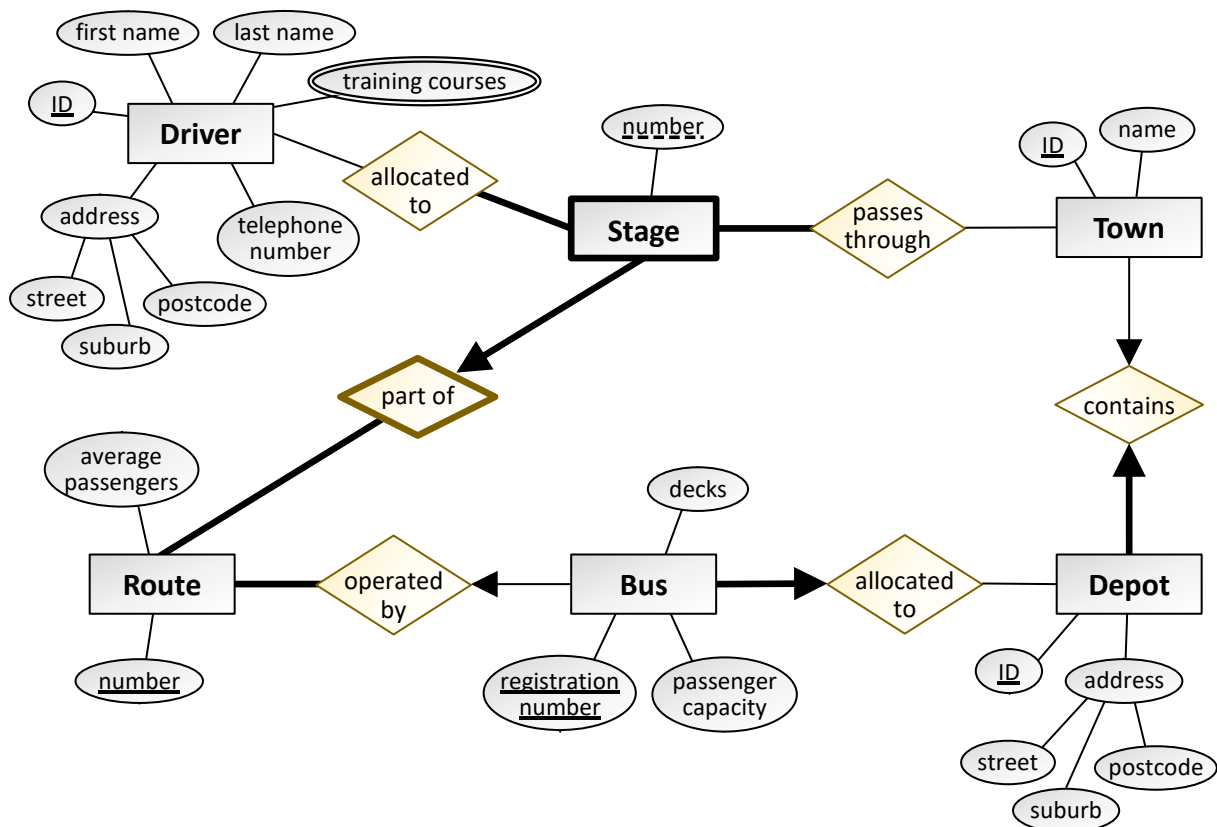
- One route can have several buses but each bus is attached to one route: Key constraint (one-to-many) and relationship 'operated by' between Bus and Route.
- Each route has more than one stages: Key constraint (one-to-many) and a relationship 'part of' between Stage and Route.
- Each stage can have more than one driver: Key constraint (many-to-many) as one driver can drive through more than one stage and a relationship 'allocated to' exists between Stage and Driver.
- Some of the towns have a depot: Key constraint (one-to-one) and a relationship 'contains' between Town and Depot
- Each depot can hold many buses and each bus is associated with one depot: Key constraint (one-to-many) and a relationship 'allocated to' between Bus and Depot.
- Each stage of a route corresponds to a journey through some or all of the towns on a route: Key constraint (many-to-many) and a relationship 'passes through' between Town and Stage.

### Participation constraints:

- The participation of Bus in the relationship ‘operated by’ is **partial** as at some point the bus could be under repair instead of being assigned to a route. However, the participation of Route is **total** as every route should have a bus assigned.
- The participation of Route and Stage is **total** in the relationship between the two entities.
- The participation of Stage is **total** in the relationship ‘allocated to’ as each stage must have a driver allocated, but the participation of driver is **partial** to accommodate newly appointed or on-leave drivers.
- The participation of Depot is **total** and Town is **partial** in ‘contains’ as not every town has a depot.
- The participation of Depot in relationship ‘allocated to’ is **partial**. However, the participation of Bus is **total**, as each bus will be associated with a depot where it is stored when not in use.
- Stage is **totally** participating in relationship ‘passes through’ as each stage has to pass through at least one town. However, each Town does not necessarily have a stage (perhaps it only has a depot), hence **partial** participation.

- c. Draw a conceptual model and populate entities with appropriate attributes (use Chen’s notation).

Bringing together all the information from part a and b, and using the concepts from lectures and revision, the following is one possible conceptual model:



- d. Discuss the logical modelling of the Driver entity.

The first step to create a logical model is to resolve multivalued and composite attributes.

**Composite** attributes are resolved by adding the component parts of the attribute directly to the entity.

The composite “address” attribute on Driver is resolved as follows (Depot address is resolved in the same way):

Driver (DriverID, FirstName, LastName, AddressStreet, AddressSuburb, AddressPostcode, PhoneNumber)

**Multivalued** attributes can be resolved in two ways. If there are a small number of distinct values for the multivalued attribute (such as work phone, home phone and mobile phone), it may be resolved in the same way as a composite attribute.

Otherwise, if the number of values is unlimited, the attribute is resolved by creating a new table with a primary foreign key (PFK) referring to the table which has the multivalued attribute, as well as a primary key column containing the attribute values themselves.

There is no limit to the number of training courses a driver can take, so the “training courses” multivalued attribute is resolved by adding a new DriverTrainingCourses table:

DriverTrainingCourses (DriverID, TrainingCourseName)  
FK

The final logical design of Driver is as follows:

Driver (DriverID, FirstName, LastName, AddressStreet, AddressSuburb, AddressPostcode, PhoneNumber)

DriverTrainingCourses (DriverID, TrainingCourseName)  
FK

**(Not covered in the workshop due to limited time)** The logical design for the remaining tables, after resolving all relationships, is as follows:

DriverAllocatedToStage (DriverID, RouteNumber, StageNumber)  
FK FK FK

Stage (RouteNumber, StageNumber)  
FK

Town (TownID, TownName)

StagePassesThroughTown (TownID, RouteNumber, StageNumber)  
FK FK FK

Route (RouteNumber, AveragePassengers)

Bus (RegistrationNumber, PassengerCapacity, Make, Deck, RouteNumber, DepotID)  
FK FK

Depot (DepotID, AddressStreet, AddressSuburb, AddressPostcode, TownID)  
FK

The physical model (Task 1.1 in Lab 4) would look like this:

