FACTFILE: SOFTWARE SYSTEMS DEVELOPMENT

(vi) ENTITY RELATIONSHIP (ER) MODELS

Entity Relationship (ER) Models

Learning Outcomes

Students should be able to:

- explain ER models:
  - entities;
  - relationships;
- identify and explain different types of relationships:
  - *one-to-one*;
  - *one-to-many*;
  - *many-to-many*;

- construct ER models.

Content

Entity Relationship (ER) modelling is regarded as a high level, top-down data modelling process which is useful for complex databases. It begins by looking at the strategy of an organisation and identifying the entities and relationships of interest to the organisation. It can be used for any type of relational database and even for a hierarchical or network database. An ER model is said to be independent of the type of database, programming languages or hardware used.

Later an analyst will modify/adapt this ER model to enable a fit to the particular data model being used: relational, network or hierarchical. Using the relational model, a schema (tables) for the database would be derived using the primary-foreign key mechanism for relationships, ensuring that all many-to-many relationships are decomposed to one-to-many relationships. **Normalisation** (discussed in next fact file) may also be used to validate the model. After mapping to the relational model the specific DBMS should be selected, for example MS Access, SQL Server or My SQL. It is at this stage that consideration should be given to physical data storage and performance issues.

Terminology

Entity (Type):[1] A set of objects or concepts (persons, places, things, or events) about which data is to be recorded and maintained, relevant to the organisation. Each entity can be uniquely identified. An example of an entity is Customer about which a range of data is stored, for example a name and an address. The entity is represented by a rectangle labelled with a singular noun.

Relationship (Type):[2] A meaningful association between two entities. Relationships are labelled with verbs and the relationship between the entities read as a simple sentence. The **cardinality** of a relationship may be *one-to-one* (1:1), *one-to-many* (1:M) or *many-to-many* (M:N). Cardinality shows the *maximum number* of entities that can be involved in a relationship.

[1] The set of objects/concepts is formally called an entity *type*; an instance of the entity type is called an entity *instance/occurrence* but usually the term entities is used and the precise meaning implied.

[2] The set of relationships is formally called a relationship *type*; an instance of the entity type is called a *relationship instance/* occurrence but usually the term relationships is used and the precise meaning implied.

An example of each type is shown in the ER model[3] below that models a business scenario in which customers order products, possibly on a website, paying with a credit or debit card.

- A Customer **places** an Order **[one-to-many]**:
  A customer can place many orders (possibly only 1 order, 2 orders, 10 orders or even 1000; i.e. more than one order is allowed); an order, however, in this business scenario can be placed by one customer only.

- A Customer ***pays_with*** a Card [***one-to-one***]:
  In this business scenario the (credit/debit) card details that the customer uses are stored. One credit card is associated with one customer only and the details can be edited if required.

- An Order ***lists*** Products [***many-to-many***]:
  An order can list many products and a product can be listed in many orders.

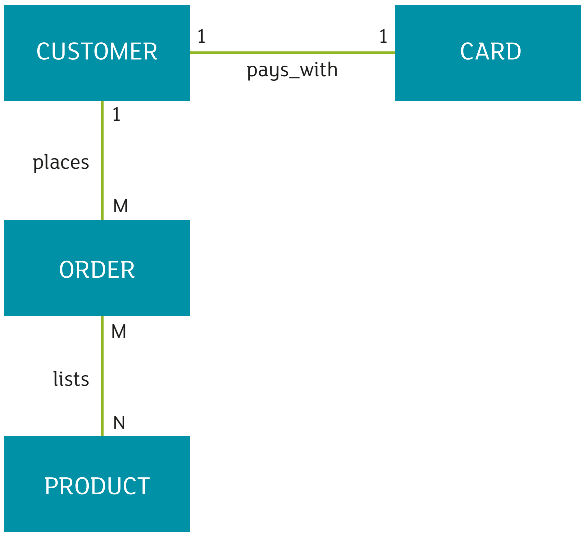


Figure 6.1 – ER Model of a typical customer order

[3] Other alternative formats may use diamonds for relationships or crow's feet to represent the many side of a relationship.

Identifying Entities and Relationships

Case Study - Caroline's Cosmetics

Caroline has been making homemade cosmetics for over 5 years and she sells them on the Internet. She makes many types of cosmetic product including bath bombs, melts and soap. She makes each cosmetic product in large batches at once in her kitchen. Her most popular product is called 'Night Night', a bath bomb, which she labels with the following description: 'Uses Lavender and Chamomile essential oils to calm and relax the mind and body'. Caroline also adds a sticker which indicates the directions of use. Each different type of product (bath bombs, melts and soap) needs a different sticker.

If an Environmental Health Officer visits, Caroline must supply a PIP (Product Improvement Programme) report. Each PIP report shows a batch number, date of manufacture, name and description of each product made. It must also show for each batch the list of all materials used including the material ID, supplier, batch number, common name (e.g. Shea butter) and weight. Caroline also likes to include the number of items produced in each batch and uses this to calculate the average cost of each product. Caroline will also need to tell the officer the suppliers of each material used, if questioned.

On Caroline's website, customers can log in and place an order for a range of products. One debit or credit card is stored for future transactions.

When modelling this scenario:

- Identify nouns and verbs: underline the <u>nouns</u> and show the verbs in italics as these will be evaluated as **potential** <u>entities</u> and *relationships* respectively. Nouns may also refer to information that is stored about an entity (attributes) or could refer to the actual data itself. Note that attributes are not identified here simply entities and relationships.

- An entity should have more than one occurrence: that is there should be more than one of them e.g. more than one customer, more than one product.

- It should be possible to uniquely identify entities. Look out for an attribute that could be used such as Customer Number.

- The entity should have several items of information stored about it; it should have several attributes. For example a customer would have a name and an address.

- The entity should be of interest to the organisation/business. There is no point in storing information which is not going to be used. Reports and other summarised information, which can be derived from information held in other entities, will not need to be stored in separate entities.

Caroline has been *making* homemade cosmetics for over 5 years and she *sells* them on the Internet.

She *makes* many types of cosmetic product including bath bombs, melts and soaps. She *makes* each cosmetic product in large batches at once in her kitchen.

Her most popular product is *called* 'Night Night', a bath bomb, which she *labels* with the following description: '*Uses* Lavender and Chamomile essential oils to *calm* and *relax* the mind and body'. Caroline also *adds* a sticker which *indicates* the directions of use. Each different type of product (bath bombs, melts and soap) *needs* a different sticker.

- Information will not be stored about Caroline as she is the only person making the cosmetics. Entities store information about a set of objects - if there was more than one person making these cosmetics there might have to be an entity called staff and a relationship 'Making'.
- **Product** is a very likely candidate for an entity. Several pieces of information about a product such as name and description should be stored.
- **Type** is a possible entity about which information might be stored or it could be something stored in a product entity. For each type there are different directions of use; an entity for type will be created to store the information for each type. Note that nouns such as bath bombs, melts, and soaps refer to examples of the data that could be stored and are not entities themselves.
- **Batch** is a noun and will be a possible entity but further information is required for design.
- A relationship is suggested in that **products** are ***made*** in **batches**.
- Kitchen is a noun – suggesting a location of where the cosmetics are made. As there is only one location, Barbara's kitchen, an entity will not be needed for this (if Barbara made them in different locations than it might be necessary to have an entity called location and relationship called ***made_in***).
- The sentence describing the name of the most popular product, gives an example of the name of a product and its description. This is data that will be stored probably in the product entity.
- Information for the sticker is derived from other information and this will not be an entity.

If an Environmental Health Officer *visits*, Caroline must supply a PIP (Product Improvement Programme) report. Each PIP report *shows* a batch number, date of manufacture, name and description of each product *made*.

It must also *show* for each batch the list of all materials *used* including the material ID, supplier ID, common name (e.g. Shea butter) and weight. Caroline also likes to *include* the number of items *produced* in each batch and *uses* this to *calculate* the average weight of each product. Caroline will also need to *tell* the *officer* the suppliers of each material used, if *questioned*.

- Information about the environmental health officer calling or regulations will not be stored. This is relevant only to the officer and not to Caroline's business.
- The PIP report is produced from the data in the database and so is not actually an entity.
- Many nouns are included for the PIP report which are simple things that are stored in other entities as data such as Batch (batch number, a unique identifier, and date of manufacture) and product (name and description).
- To produce the PIP report an extra entity for **material** is required as well as an extra relationship: each **batch *uses* material**. The material entity will store information such as material ID, a unique identifier, common name. The batch entity may also store the number produced. The average weight is probably calculated from information already in the database and it doesn't need to be stored.
- A **supplier** entity is also required (contact details of each supplier would be stored) and linked with a relationship: **supplier *supplies* material**.

On Caroline's website, customers can *log* in and *place* an order for a range of products. One debit or credit card is stored for future transactions.

- Information will be probably be stored in a **customer**, **order**, **product** and **card** entity; although no details have been given about the exact information to be stored. Relationships will be needed: customer ***places*** order; customer ***stores/uses*** card or *customer* ***pays with*** card; order ***is_for/lists*** product.

Drawing the ER Diagram

The initial ER diagram can be drawn showing the entities, relationships and cardinality. Cardinality of the relationships (one-to-one, one-to-many, many-to-many) could be determined with the help of occurrence diagrams (Further reading). ER diagrams drawn by different analysts may be different but equally valid.

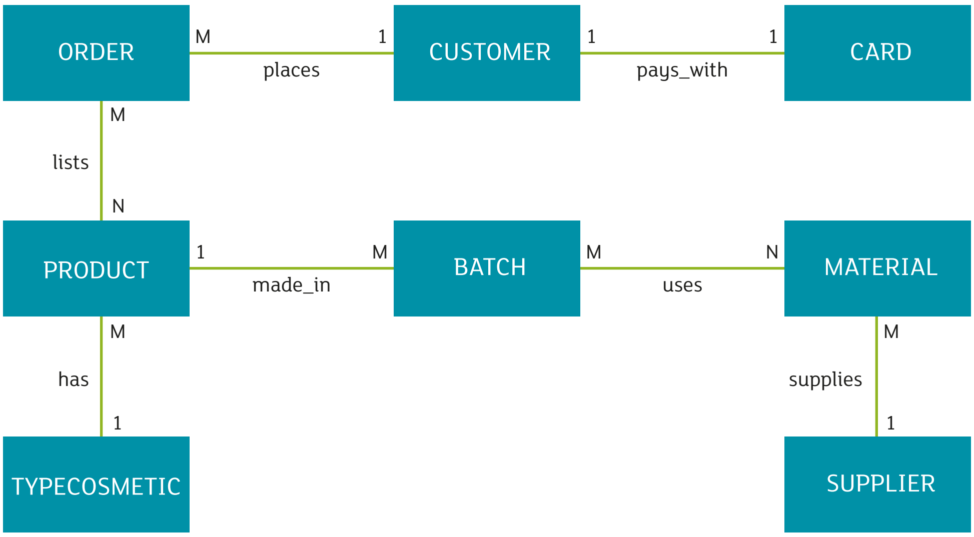


Figure 6.2 – Initial ER Model of Caroline's Cosmetics

Evaluating the Design and Adapting/Mapping to the Relational Model

The database designer has evaluated the original database design and adapted/mapped it to enable implementation in a relational database.

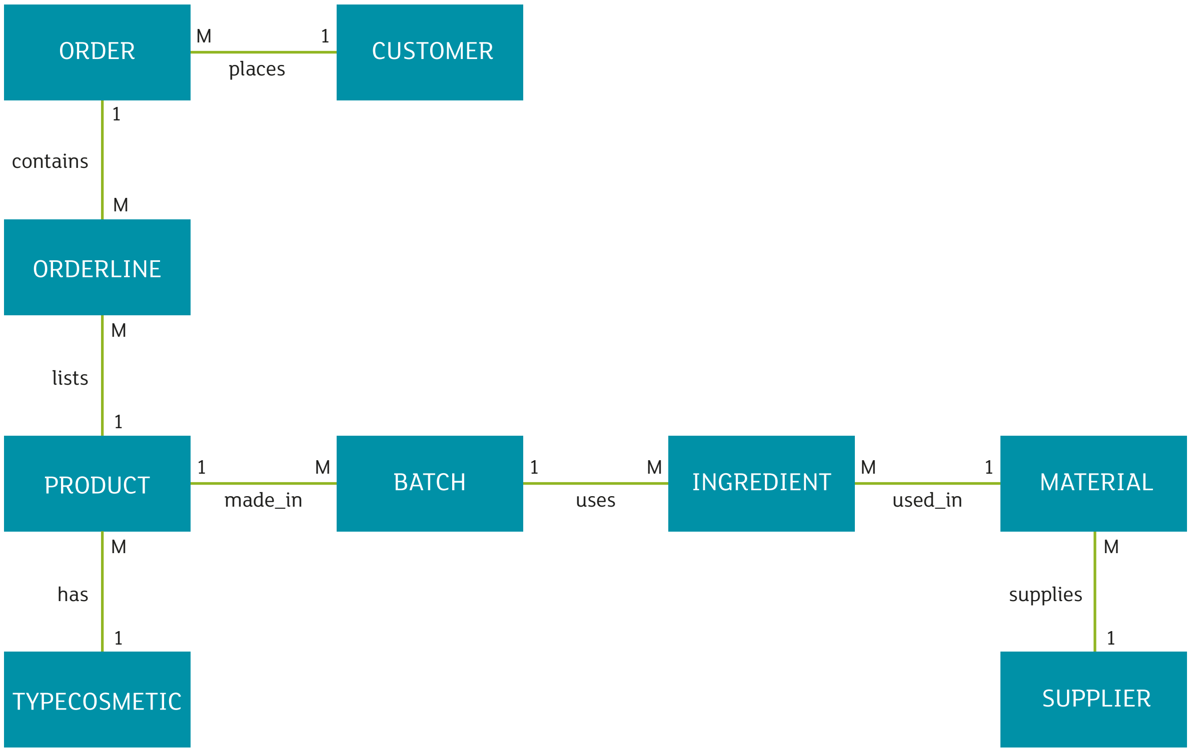


Figure 6.3 – Modified ER Model of Caroline's Cosmetics

Two changes have been performed:

- The Card and Customer entities have been combined: sometimes two entities with a one-to-one relationship are simply combined - requires less joins in queries.

- Each many-to-many relationship has been decomposed into two one-to-many relationshipswith a new associative entity in between. This is required as many-to-many relationships cannot be directly formed using the primary-foreign key mechanism in the relational model, only one-to-many and one-to-one are possible.
  - Orderline is the specific order product
  - Ingredient is the specific Batch Material
  -

The analyst may also check that there are just enough relationships allowing queries to be created according to Caroline's needs. They might also check that all entities are required (an entity which only stores one piece of data may not be needed).

Specify how Relationships and Tables are formed in a Relational Database

The analyst will then specify the database schema for the relational model.

- Each entity becomes a table/relation in the database.

- A primary key is chosen to uniquely identify each record.

- Relationships are formed by posting a copy of this primary key from the table on the one side of the relationship to the table on the many side of the relationship.

- Other fields/attributes are added based on the case study and assisted by normalisation.

The diagram below shows how a copy of the unique identifier of the entity, which will be the primary key of the table, is posted from the one to the many side of the relationship.

The skeleton tables obtained are:

TYPECOSMETIC (TypeName)
PRODUCT(ProductName, TypeName*)
BATCH(Batch No, ProductName*)

Extra fields may then be added to these tables

(normalisation will assist this).

TYPECOSMETIC (TypeName, DirectionsOfUse)
PRODUCT(ProductName, ProdDescription, TypeName*)
BATCH(Batch No, Date of Manufacture, No Produced, ProductName*)

* Indicates the Foreign Key

Note there are 3 tables and 2 foreign keys linking these tables together.

The derived data in the tables has not been shown here as they can be calculated from data already present in the tables but for various reasons derived data may be stored, for example to improve performance.

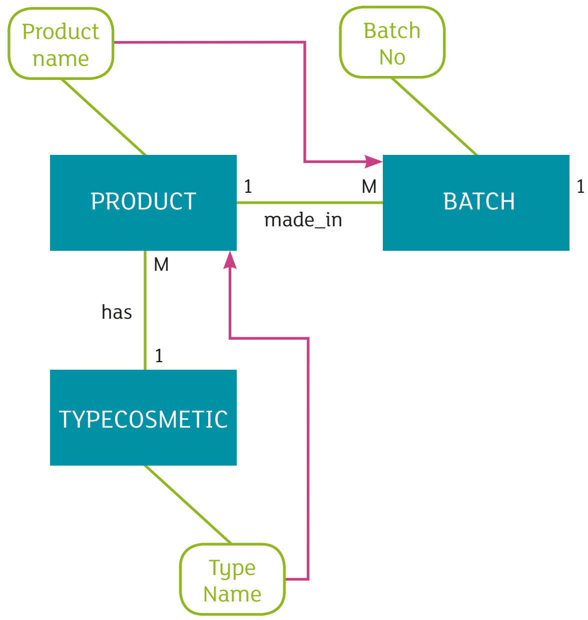
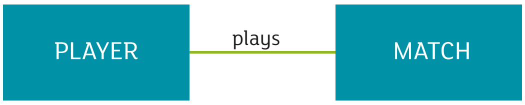
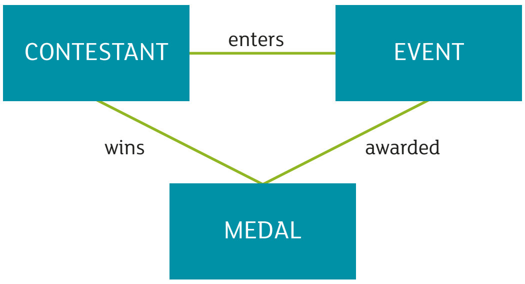


Figure 6.4 – Showing how relationships are made

Questions

1 Add cardinality to the following relationships. Note any assumptions made.

ER Diagram	Note
	A zoo has many animals. Each animal is looked after by one specific keeper. The keeper called Dave looks after both the elephants and the lions.
	A detective arrests the suspect.
	A recipe contains ingredients.
	The database holds details on matches played by football players over the past two seasons.
	On admission to hospital a patient is sent to a ward. The database only holds details of current patients. Each ward can contain up to 25 patients.
	A student can register on one course each year. The database will store the registrations of students over the past 10 years. A course will only run when there are more than 10 students registered on it.
	A contestant can enter many events. At each event a gold, silver and bronze medal may be awarded to some of the winning contestants.

Questions

2 The analyst draws an initial ER diagram for Customers hiring equipment from a shop. The shop keeps a record of all customers who hire equipment over the past 5 years.

Before the analyst derives tables for the MS SQL Server Database the ER model is adapted.

Why did the analyst adapt the ER model?

....

....

....

....

....

....

....

....

....

....

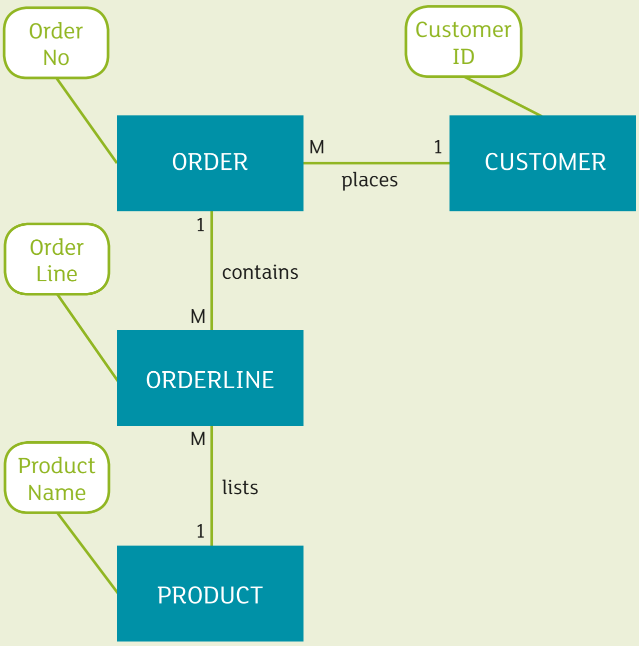
....

....

....

Questions

3 The analyst wants to use the ER model to derive tables for the relational database, each entity will become a table. The unique identifier for each entity on the ER model will be drawn. These will be part of the primary key of each table. Using arrows show how the primary keys would be copied and posted as foreign keys to create relationships.



Some skeleton tables are provided below with primary keys. Insert the foreign keys.

CUSTOMER (CustomerID, ...)

ORDER (OrderNo, ...)

ORDERLINE (OrderNo, OrderLine, ...)

PRODUCT (ProductName, ...)

Questions

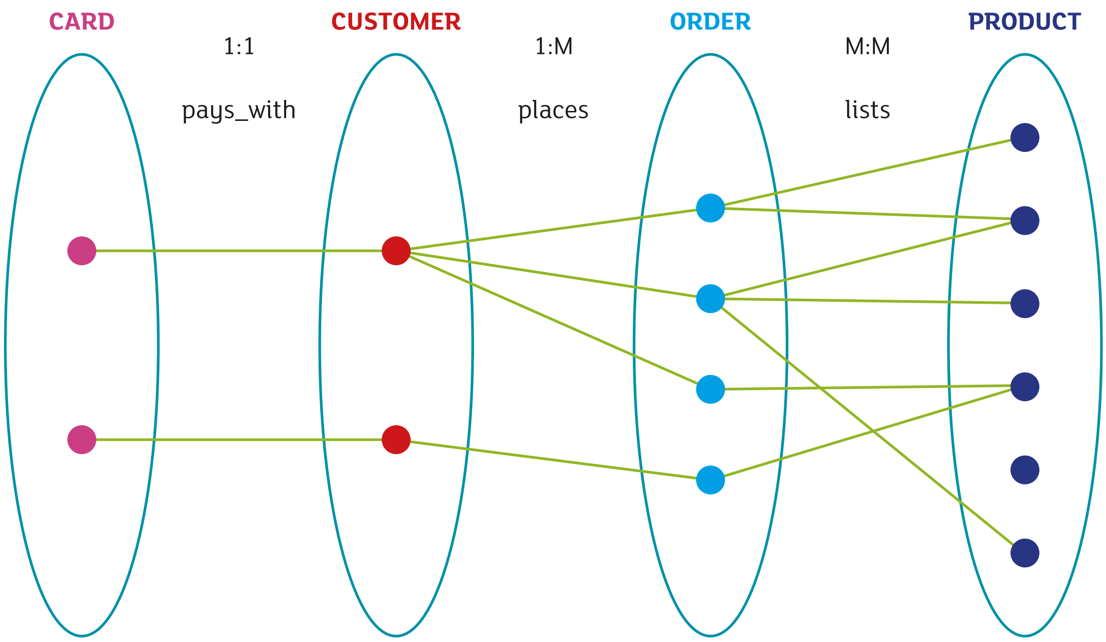
4 Examine each statement below about ER modelling and state whether it is true or false.

A	ER diagramming is a top-down data modelling process.	
B	Different analysts may produce different ER models which are equally valid.	
C	A good strategy for identifying entities is to identify verbs in a textual description of the organisation.	
D	Entities usually only store one piece of data about an object.	
E	Entities are normally created to hold information about an object even if there is only ever one instance of that object in the business concerned.	
F	Foreign keys are shown on ER diagrams.	
G	Entities associated with one another in a one-to-one relationship are sometimes combined together into one entity.	
H	Many-to-many relationships cannot be implemented in a relational database unless they are decomposed.	
I	A one-to-many relationship is implemented in a relational database by posting a copy of the primary key from the table on the one side of the relationship to the table on the many side of the relationship.	
J	It would be wise to link every entity to all other entities using relationships.	
K	It is important to have the correct relationships which facilitate the creation of queries relevant to the business.	

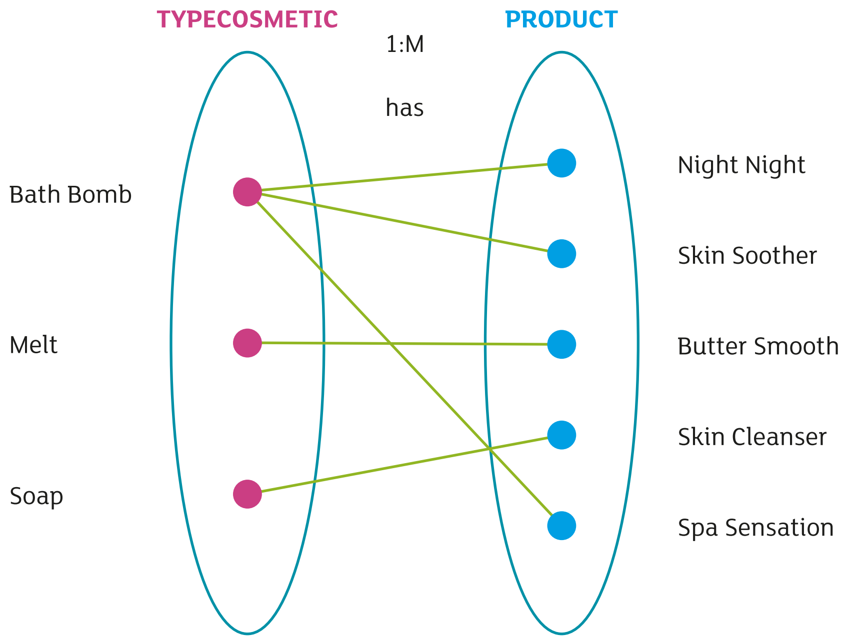
Further Reading

Occurrence Diagrams

The use of occurrence diagrams is useful for the explanation of cardinality. The set of occurrences in each entity type is represented by small circles (add in examples of data if preferred).



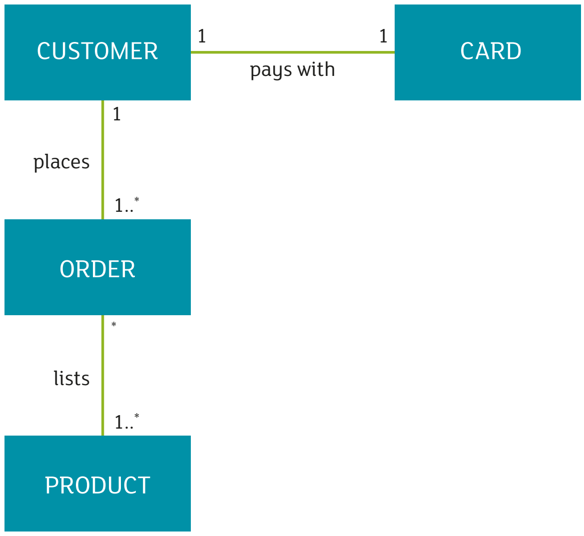
In the case study/scenario called Caroline's Cosmetics occurrence diagrams may also be used.



Draw occurrence diagrams for the other relationships (using imaginary data if necessary): between Product and Batch, Batch and Material and Material and Supplier. Occurrence diagrams are also useful to check if queries can be designed using these relationships, for example, 'Which products are bath bombs?'

Link to UML Class Diagrams

Database designers may also be presented with static UML class diagrams for database design. There are many similarities but some differences. Examine the simple UML Class diagram opposite. [4] No attributes or methods/operations are shown in the diagram (and neither are any software classes) as we are in an early stage of analysis.



Implementation of many-to-many relationships.

It may be useful to know how the many-to-many relationship between BATCH and MATERIAL is actually implemented in a database by inserting an associative table called INGREDIENT between them and linking with one-to-many relationships.

INGREDIENT

BatchID	MaterialID
1	1
1	2
1	3
1	4
1	6
2	1
2	2
2	3
2	8

The associative table allows the BatchID, e.g, 1, of the BATCH table to be matched with all the MaterialIDs of the MATERIAL Table (e.g. 1, 2, 3, 4, 6). The primary key of this table is a composite of the primary keys found in the BATCH and the MATERIAL table. Another field in this table could be the weight required for this recipe.

[4] Note that ER models can also explicitly show the minimum and maximum values for participation