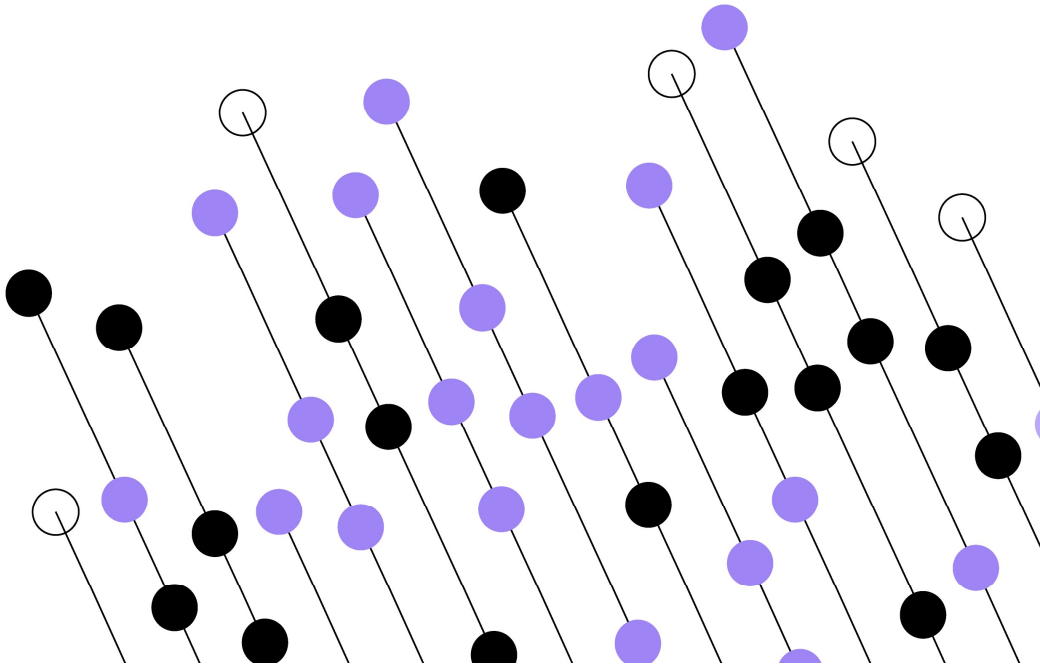
Logical Data Models

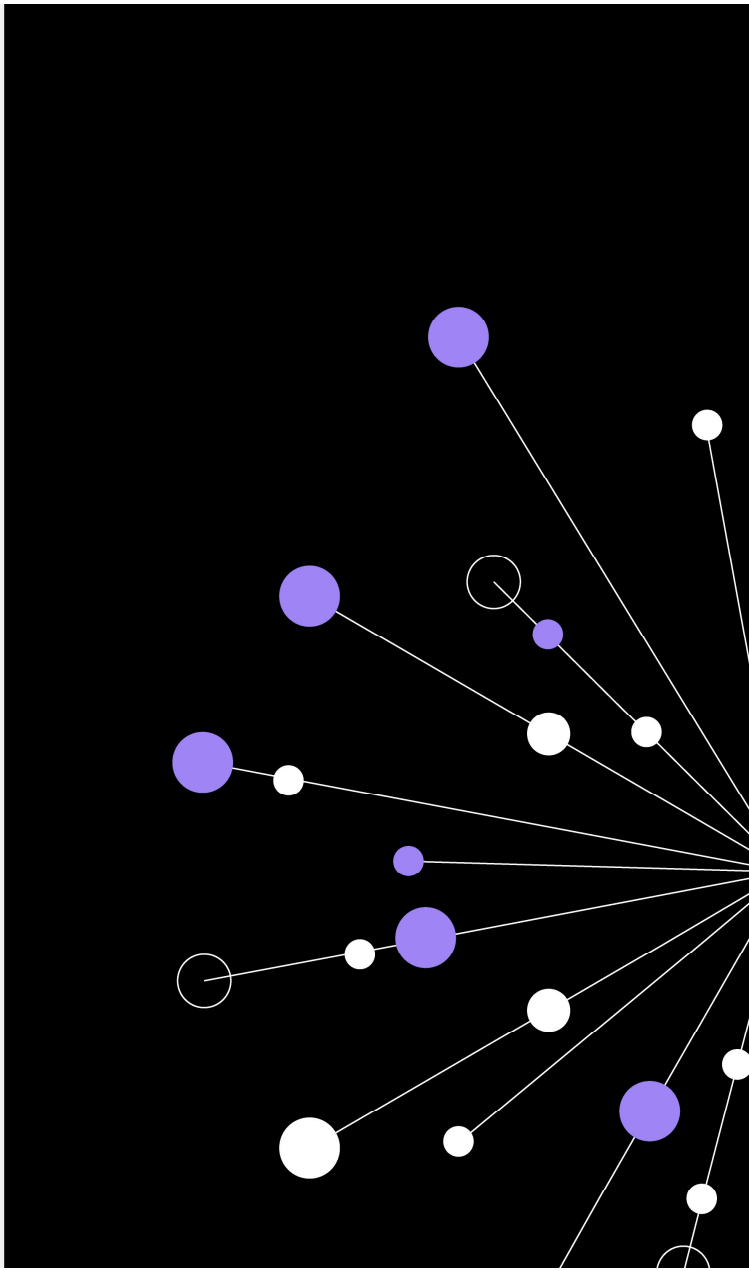


Data Modelling

Contents

Logical Data Model:

- All entities and relationships
- All attributes for each entity
- The primary key for each entity
- Foreign keys



QA

Logical data model

An entity in a logical data model has more detail than a conceptual data model. At this level, you'd identify:

- the entity's attributes, i.e. all the data about the entity that need storing.
- a primary key: one or more attributes that uniquely identifies each entity.

The relationships between the entities are represented in terms of the primary keys. To connect two entities in a logical data model, one of the entities needs an attribute that references the other entity's primary key. This is known as a **foreign key**.

Single-valued attributes

An attribute is data about the entity. There are two main types of attribute:

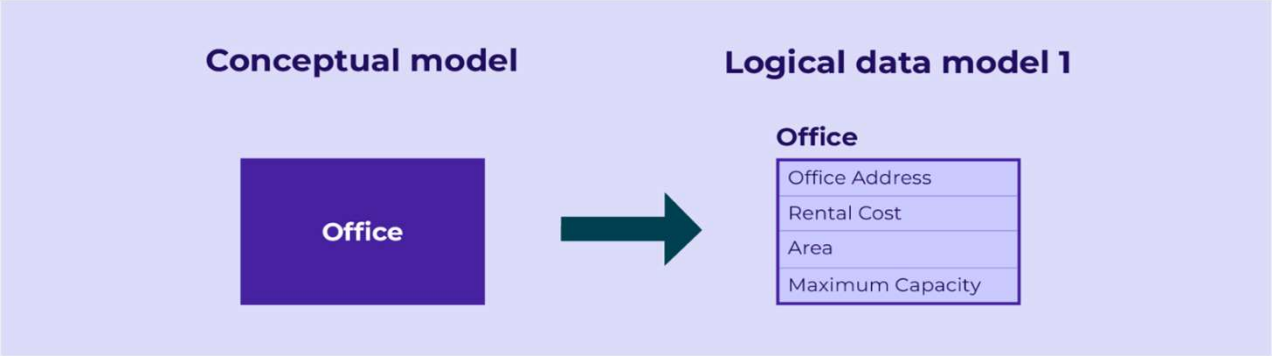
- Single-valued
- Multi-valued

A single-valued attribute is an attribute that you only need to store a single value for. If the value should be changed, the previous value will be replaced.

Example:

An office has an **address**, a monthly **rental cost**, an **area** in square meters, and a **maximum capacity**.

Address, rental cost, area, and maximum capacity are single-valued attributes.



Multi-valued attributes

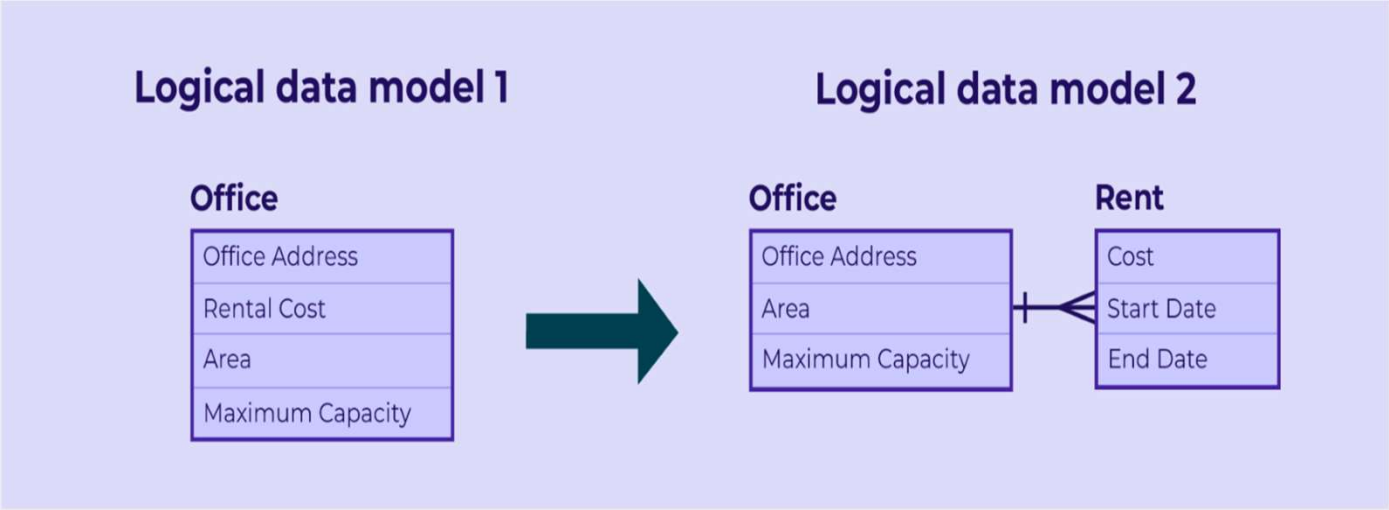
A multi-valued attribute is an attribute that may require multiple values to be stored. Reasons for needing multiple values:

- the attribute could change over time and history is required to be kept;
- there can be multiple values for the attribute simultaneously.

Multi-valued attributes are added to a new entity, and a relationship is added between the original entity and the new entity.

Multi-valued attributes: example

In the example of office rental data, we may want to keep track of office rental costs over time. A change in the value of the attribute ‘rental cost’ would cause the previous value to disappear. This can be handled by adding a new entity – ‘rent’ – with attributes cost, start date, and end date.



Attributes and primary keys

Entities may have one or more attributes that uniquely identify their instances.

- If it is a single attribute, then that attribute will likely be a good candidate to be the **primary key** for that entity.
- If multiple attributes need to be combined to ensure uniqueness, then this is called a **composite primary key**.

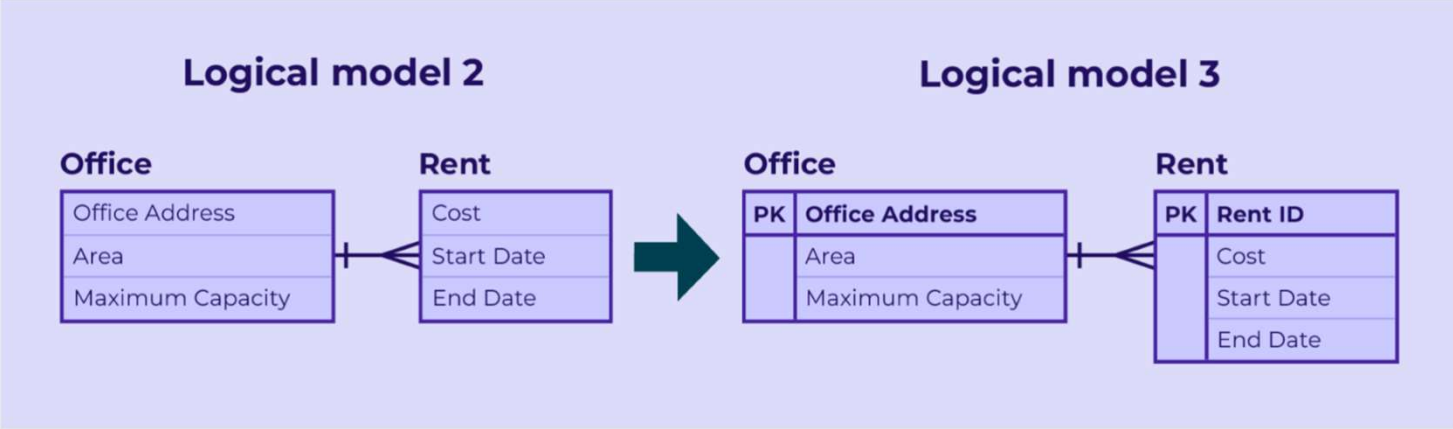
If no attributes can be used to ensure uniqueness, or the use of a composite primary key is inappropriate, it's normal to introduce a new attribute that is guaranteed to be unique. This is known as a **surrogate primary key**.

It is most often implemented as an incrementing integer, e.g. the first instance's primary key value will be 1, the second instance's primary key value will be 2, etc.

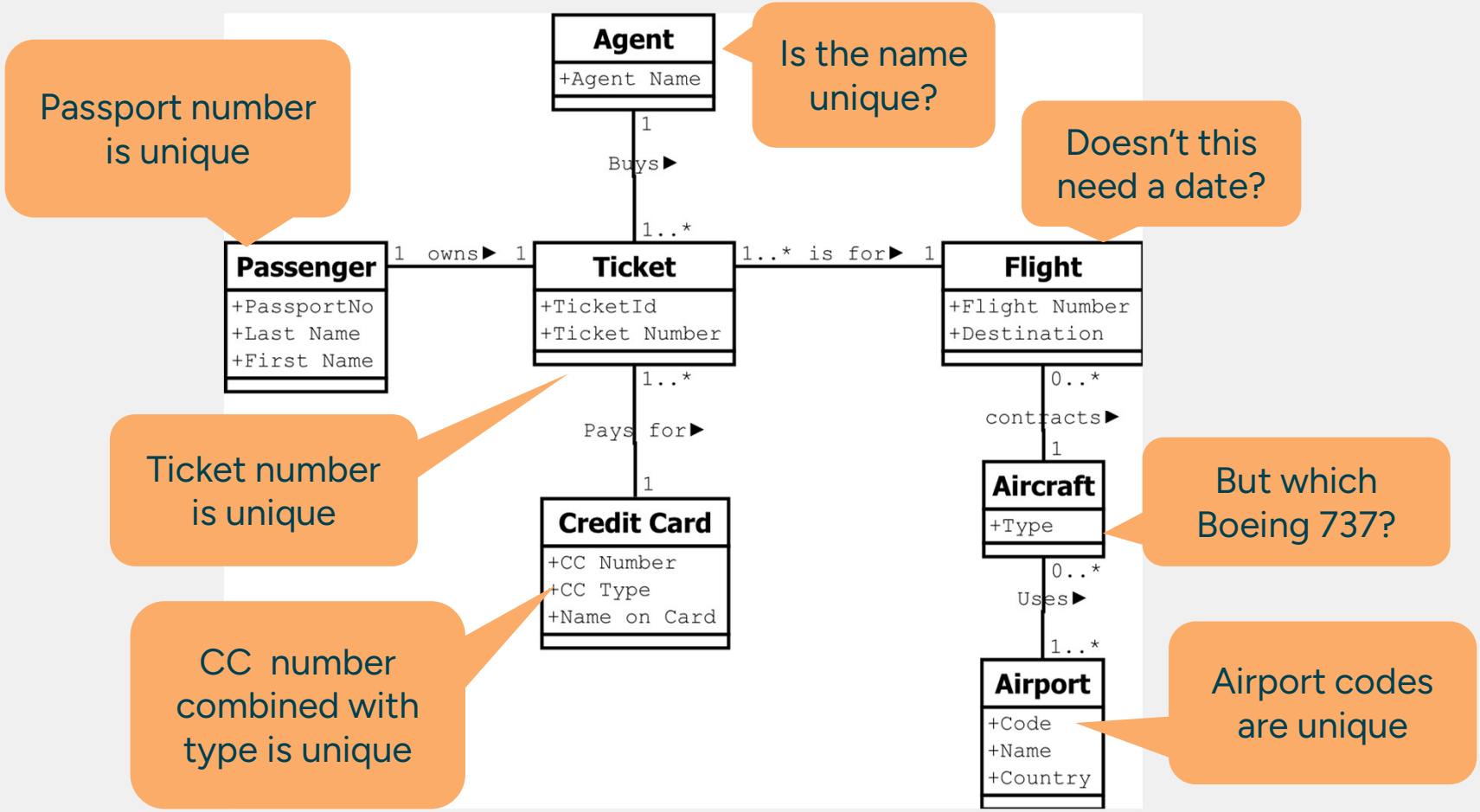
Attributes and primary keys

Consider the office rental data example.

For the office entity, the address attribute seems to be sufficient to ensure uniqueness. For the rent entity, none of the attributes can guarantee uniqueness - different rental periods might overlap with the same cost. In this case, we must introduce a surrogate primary key.



Airline system – conceptual model



Airline system – conceptual model