2022

DAT601 REPORT

NICOLE BARNETT-FORSTER

Milestone1

Conceptual Modelling Description

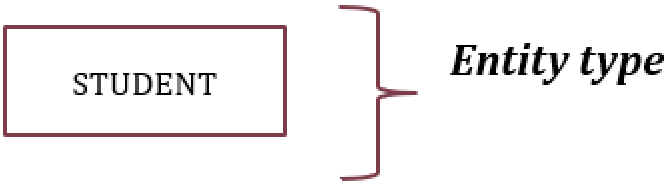
Conceptual modelling is a way of visualising database design, using diagrams and tables. We use it to show the relationships between entities, with their attributes, and to show constraints.
It's helpful to map the structure of the database before starting the physical design. The diagram should be able to be understood by both technical and non-technical people.

This phase should produce the following:

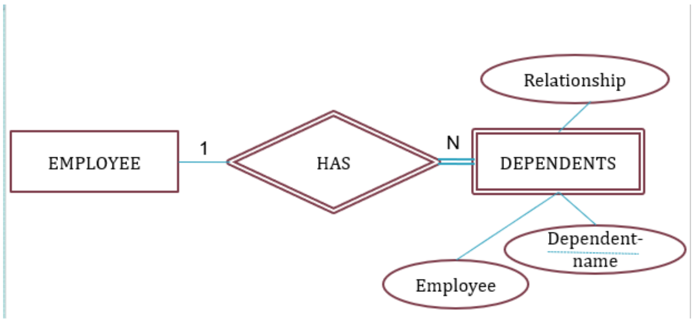
- Conceptual data model
- All entities identified
- Relationships identified
- Attributes identified and assigned to entities
- Identify attribute domains
- Identify primary and candidate keys
- Entity types
- Produce ER diagram
- Review

Diagram components

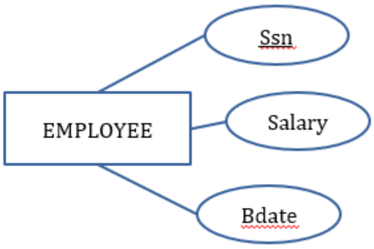
An entity is anything worth storing or modelling data about. It can be a person, object, or something intangible. An entity is represented by a square.



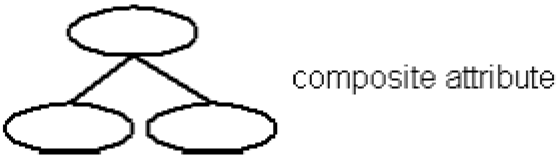
A weak entity is an entity that is dependent on another entity and cannot exist without it.



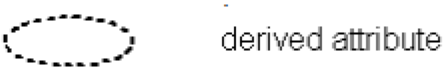
An attribute is a characteristic or a property that belong to an entity. These are useful for giving better understanding of an entity. They are represented by circles or ovals.



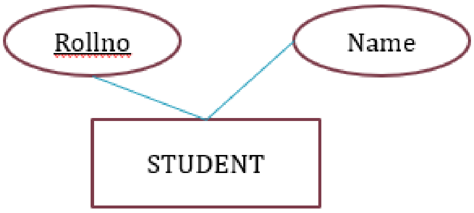
A composite attribute is an attribute that could be split into multiple values. These are represented by multiple attributes stemming from a single entity.



A derived attribute is an attribute that has/can be derived or calculated from another attribute. This is represented by a circle or oval drawn with a dotted line.



A primary key Is a unique identifier. Represented by the dotted underline.



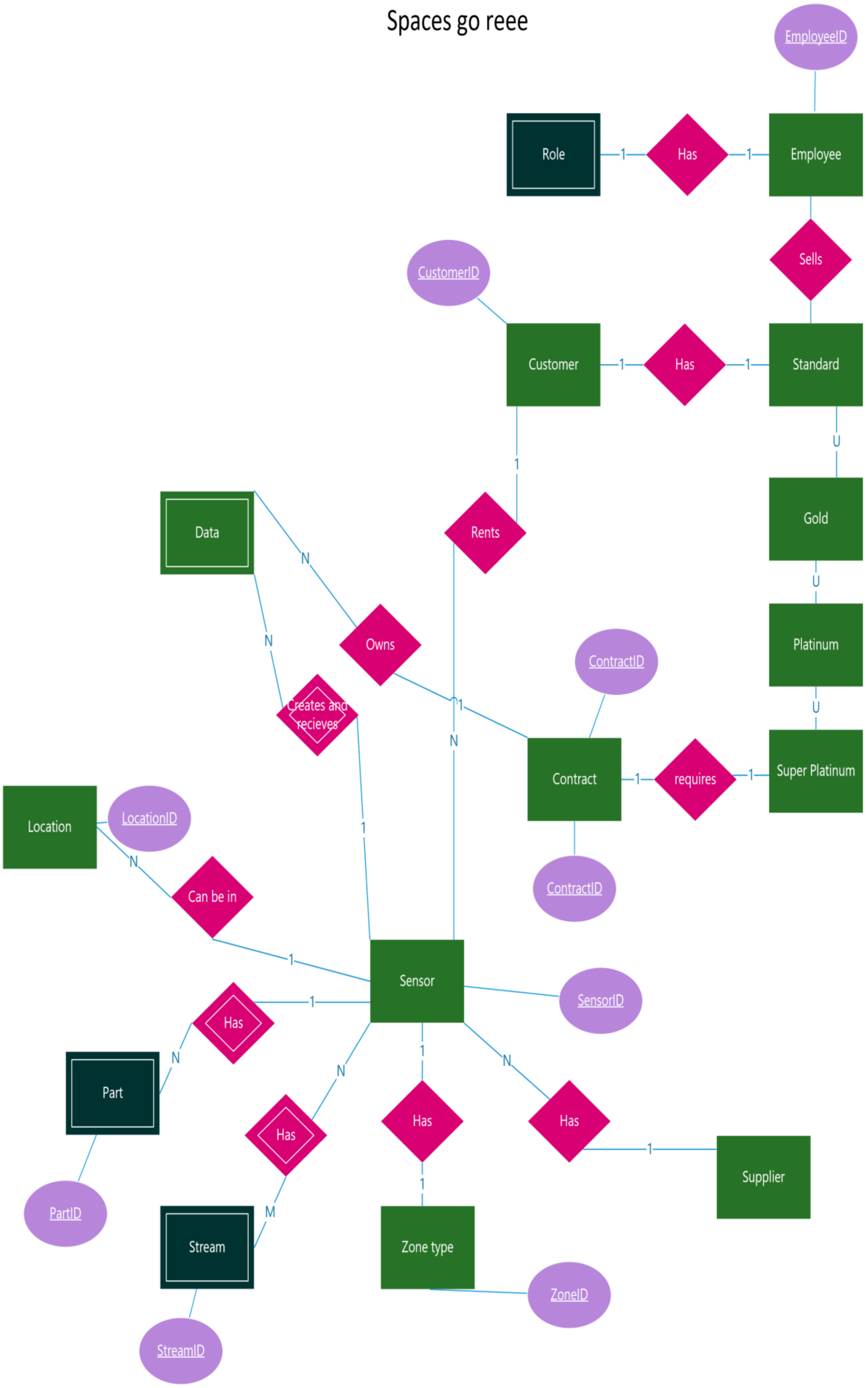
A candidate key is also an identifier, but you can have many, as opposed to a primary key.
A composite key is a key made up of multiple values.
A foreign key is a unique identifier from another table.

A relationship between two entities represents the flow of data through the system.

The optionality of relationships determines if they are Mandatory or Optional.

The cardinality defines how many instances of one entity are related to instances of another entity

Conceptual Model with Chens Notation



SpacesconceptV2final.pdf

Data Dictionary

Entity Name	Description	Aliases	Occurrence
Contract	Legally binding agreement between customer and Spaces	Subscription	Customer purchases Super Platinum
Standard	Base level subscription	-	Customer purchases Standard
Gold	Second level subscription	-	Customer purchases Gold
Platinum	Third level subscription	-	Customer purchases Platinum
Super Platinum	Top tier subscription	-	Customer purchases Super Platinum
User	Anyone using the system	-	Person accesses system
Employee	Person employed by Spaces	Staff	Person has staff contract with Spaces
Role	Dictates staff permissions	-	Staff allocated permissions
Customer	A person who pays for Spaces goods and services	Subscriber	Person pays for and receives subscription
Data	Info/input from 3d sensors	-	Sensor takes input from surroundings
DataSet	Stored data from sensor	-	Data is stored
Stream	Live data transfer from sensor to users	-	Sensor sends data
Sensor	Hardware providing data to system	-	Is hardware
Zone	General area sensor belongs to	-	Has sensor in it
Location	Area that sensor is located	-	Has coordinates
Parts list	List of parts belonging to one sensor	-	Sensor has parts
Supplier	Organisation that provides specialty hardware	-	Has agreement with Spaces
Zone type	Different zones types	-	Has environment

Entity Name	Cardinality	Participation	Relationship	Participation	Cardinality	Entity Name
Employee	1	Partial	Sells	Partial	N	Standard
Employee	1	Partial	Has a	Partial	1	Role
Super Platinum	1	Partial	Requires	Partial	1	Contract
Customer	1	Total	Rents	Total	N	Sensor
Standard	1	Partial	Can be upgraded to	Partial	1	Gold
Gold	1	Partial	Can be upgraded to	Partial	1	Platinum
Platinum	1	Partial	Can be upgraded to	Partial		Super Platinum
		Partial		Partial		
Contract	1	Total	Owns	Total	N	Data
Sensor	N	Total	Has a	Total	1	Supplier
Sensor	1	Partial	Has a	Partial	1	Zone type
Sensor	N	Partial	Has	Partial	M	Stream
Sensor	1	Partial	Creates and receives	Partial	N	Data
Sensor	1	Total	Has	Total	M	Parts
Sensor	1	Partial	Has	Partial	1	Zone

Relationships

Entity Name	Cardinality	Participation	Relationship	Participation	Cardinality	Entity Name
Employee	1	Partial	Sells	Partial	N	Standard
Employee	1	Partial	Has a	Partial	1	Role
Super Platinum	1	Total	Requires		1	Contract
Customer	1	Total	Rents		N	Sensor
Standard	1	Partial	Can be upgraded to	Partial	1	Gold
Gold	1	Partial	Can be upgraded	Partial	1	Platinum

			to			
Platinum	1	Partial	Can be upgraded to	Partial		Super Platinum
Contract	1	Partial	Owns	Partial	N	Data
Sensor	N	Total	Has a	Total	1	Supplier
Sensor	1	Partial	Has a	Partial	1	Zone type
Sensor	N	Partial	Has	Partial	M	Stream
Sensor	1	Partial	Creates and receives	Partial	N	Data
Sensor	1	Total	Has	Total	M	Parts
Sensor	1	Partial	Has	Partial	1	Zone

Attributes

Entity name	Attributes	Description	Domain	Aliases	Composite	Derived	Nulls	Key?	Default Value
Employee	EmployeeID							Yes	
Customer	CustomerID							Yes	
Contract	ContractID							Yes	
Sensor	SensorID							Yes	
Location	LocationID							Yes	
ZoneType	ZoneID							Yes	
Part	PartID							Yes	
Stream	StreamID							Yes	

Rationale

Role

This entity stores the various positions that can be held by employees. One employee can only hold one title at a time. Role has a relationship with the Employee entity.

Employee

This entity stores all current employees. Employee has a relationship with the Standard entity. The ID of the employee that sold a Standard subscription will be stored for record keeping.

Standard

This entity stores the functionality and constraints for a Standard subscription. Standard has a relationship with Gold, Gold will inherit the properties of Standard and its own additional properties.

Gold

This entity stores the functionality and constraints of a Gold subscription. Gold has a relationship with Platinum, Platinum will inherit the properties of Gold and its own additional properties.

Platinum

This entity stores the functionality and constraints for a Platinum subscription. Platinum has a relationship with Super Platinum, Super Platinum will inherit the properties of Gold and its own additional properties.

Super Platinum

This entity stores the functionality and constraints for a Super Platinum subscription. Super Platinum has a relationship with Contract, as all customers who hold a Super Platinum must have a contract. This is the only tier that requires a contract.

Customer

This entity stores all data about current customers, regardless of subscription tier. Customer has a relationship with the Sensor entity, as a customer will be renting/interacting with a sensor.

Contract

This entity will store Data from the Data entity that is specific to a customer. Contract has a relationship with the Data entity.

Data

This entity will store all data collected from sensors; thus it will have a relationship with the Sensor entity.

Sensor

This entity stores info about its parts, supplier, location, zone type, and as a result will have relationships with these entities. A sensor must know about its parts so records of its maintenance are kept. Supplier so new parts can be ordered. Location to keep track of all sensors. Zone type so viewers and staff know the zone type of the sensor. A sensor will also need to know to send relevant data to customer, so sensor will have a relationship with the Data entity.

Supplier

This is an external system, but Sensor needs to know about it to receive parts, so Sensor will have a relationship with the Supplier entity. This entity will store sensor parts and maintenance records.

Zone Type

This entity will store the differing Zones sensors can be placed in, so it will need to have a relationship with Sensor.

Stream

This entity will store data that goes between the customer and the sensor, but only customers with a contract, so it will need a relationship with the Contract entity.

Part

All sensors will need maintenance, which can involve parts needing replaced. The Part entity will store data about the parts sensors have. (I realise now I need a relationship between supplier and part but is too late now)

Location

This entity will store the coordinates that sensors have been placed in. It will need a relationship with the Sensor entity.

Milestone 2

Logical Modelling Description

This is a more detailed entity relationship diagram than the conceptual model.
There are a few rules to be observed in this style of modelling, which are:

- Primary and Foreign keys are required for the Logical model

- Composite attributes become entities

- Simple attributes will remain as such

MANY TO MANY RELATIONSHIPS

Many to many relationships do not exist in a database, so there must be a way to model that sort of relationship. Many to many can exist in a conceptual model but you must remove direct many to many relationships in a logical model. An example of how this might work is a many to many relationships between student and course.
Many students can be part of many courses and many courses can have many students.
What you would do is add a table in between the two tables called StudentCourse. It will contain the courseID and Student ID and additional info as required. The 2 tables will each have a one to many relationships to the StudentCourse table.

ONE TO ONE

When transforming to logical, in a one to one relationship, the ID from the either side of the relationship gets added as a new column to the other side of the relationship as a foreign key.

ONE TO MANY

When transforming to logical, in a one to many relationship, the ID from the One side of the relationship gets added as a new column to the Many side of the relationship as a foreign key.

ONE TO ONE

When transforming to logical, in a one to one relationship, the ID from the either side of the relationship gets added as a new column to the other side of the relationship as a foreign key.

ONE TO MANY.

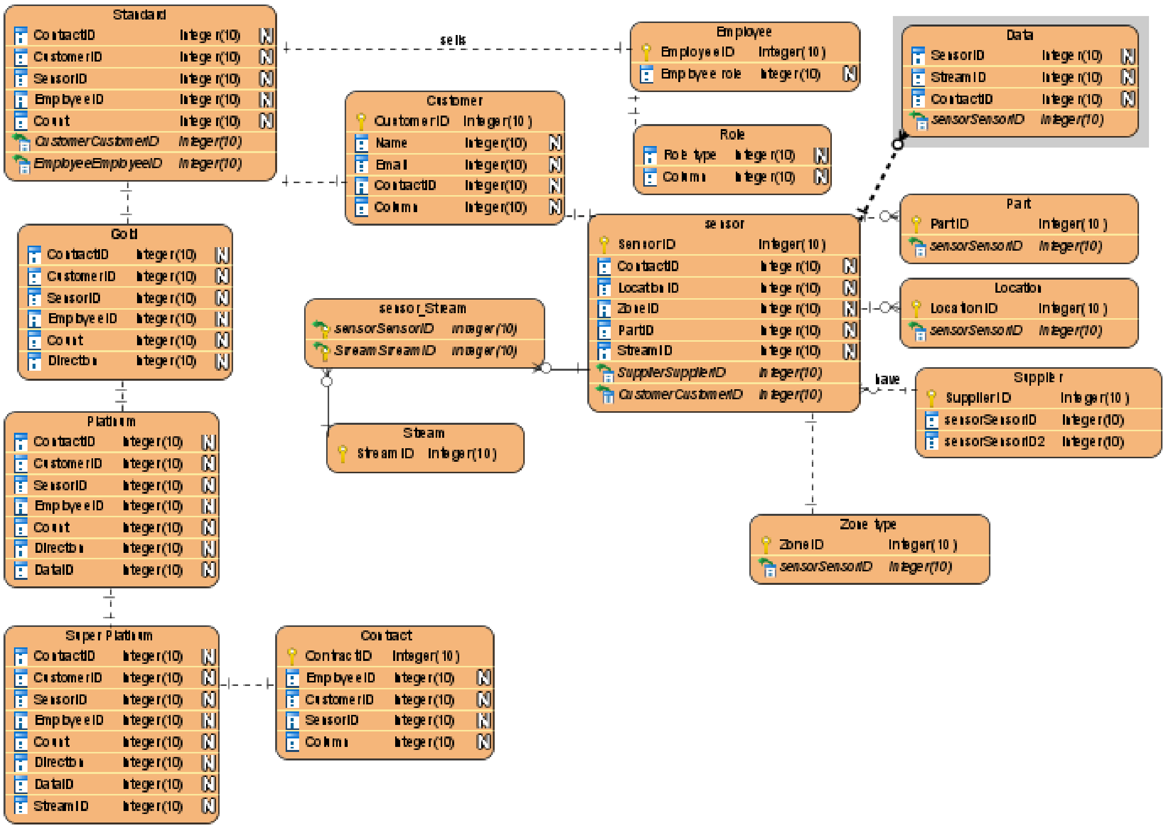
When transforming to logical, in a one to many relationship, the ID from the One side of the relationship gets added as a new column to the Many side of the relationship as a foreign key.

CROWS FOOT NOTATION

Crows foot diagrams represent entities as boxes and relationships as lines between these boxes. The different shaped at each end of these lines represents the cardinality between entities. The crow's foot notation does not allow relationships to have attributes, where relationships would usually have attributes, these entities are promoted to existing as entities of their own.

Conceptual to Logical

Crow's Foot Notation ERD



spaceslog.vpp

ERD Rationale

Normalisation

Unnormalized Form

First Normal

Second Normal Form

Third Normal Form

Boyce-Codd Normal Form

Fourth Normal Form

Data ditionary

Describe how to transform a conceptual model based on CHEN ERDs to a logical model.
Create a list of “mapping rules”.
○ A logical diagram, with a clear explanation of the purpose of the relations.
Describe
how you undertook normalisation of your database design.
○ Documentation of your relations. Include a data dictionary -this must be in tabular
form as given in class.
○ A copy of the NaLER analysis carried out by you.

MS3

Transaction	Marks allocated	Mark given
a. A salesperson subscribes to a new standard subscription to a 3D sensor. The transaction receives the salesperson Id, a discount %, all subscriber details, and a 3D sensor ID.	4	

Transaction diagram, and form, query		
b. For each salesperson list the subscribers they have sold a subscription to. The transaction receives the salesperson's name as input, and presents each subscriber's name, address, and the % they were discounted	2	
c. Write SQL to be used to insert data from a 3D sensor to its stored data on the Spaces database. The transaction receives the 3D sensor ID and all the data from a data stream. That is made up of one or more records of 3D human imagery – texture, body shape and estimated skeletal points Human voice, Shared audio from other sources, time, longitude, latitude, and altitude	2	
d. List the location in latitude, longitude coordinates, of each 3D sensor that is currently in a contract (subscribed to). The transaction presents the Contracting organisation or person's name, a 3D sensor ID, a Latitude, and a Longitude.	3	
e. For a contract list all the data collected. The transaction receives the contracting organisation's name and presents for each collected data record, the contracting organisation's name, a 3D sensor ID, 3D human imagery – texture, body shape and estimated skeletal points, Human voice, Shared audio from other sources, time, longitude, latitude, and altitude	3	
f. For each 3D sensor present the list of subscribers who are viewing a live 3D video stream. The transaction lists 3D sensor ID, Subscriber Name, Stream ID.	3	
g. For a given 3D sensor list all the suppliers of parts. The transaction receives the 3D sensor ID, and presents the Supplier Name and, Part Name	3	
h. Update the location and Zone of a 3D sensor. The transaction receives the 3D sensor ID, a location and a Zone expressed as a list of coordinates in latitude, longitude pairs. It updates the location of the 3D sensor and its corresponding Zone. (This transaction may require more than one update query.)	4	
i. Delete the data collected for a given Contract. The transaction receives a Contract ID, the data collected for a Contract is deleted.	2	
j. Write a query that displays the total cost of all parts replaced in maintenance of a 3D sensor. The transaction displays the 3D sensor ID, Total Cost of replaced parts, for every 3D sensor.	4	
Total	30	/30

Conclusion