Convert a conceptual model into a logical model. Apply normalisation and NaLER concepts.

DAT601 Assessment Two Part 1

Logical Modelling

26/5/23

Jayden Houghton – 13524050

Contents

[Model Conversion 2](#_Toc136025297)

[Logical ERD 3](#_Toc136025298)

[Normalisation 4](#_Toc136025299)

[Forms and Steps 4](#_Toc136025300)

[Unnormalised (0NF) 5](#_Toc136025301)

[First Normal Form (1NF) 5](#_Toc136025302)

[Second Normal Form (2NF) 6](#_Toc136025303)

[Third Normal Form (3NF) 6](#_Toc136025304)

[Boyce-Codd Normal Form (BCNF) 7](#_Toc136025305)

[Fourth Normal Form (4NF) 8](#_Toc136025306)

[Normalised ERD 8](#_Toc136025307)

[Data Dictionary 9](#_Toc136025308)

[Document relations 9](#_Toc136025309)

[Document Attributes 11](#_Toc136025310)

[NaLER Analysis 15](#_Toc136025311)

[Process 15](#_Toc136025312)

[Analysis 15](#_Toc136025313)

[Entities 15](#_Toc136025314)

[Relationships 19](#_Toc136025315)

[References 23](#_Toc136025316)

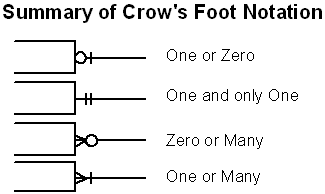
# Model Conversion

*Describe how to transform a conceptual model based on CHEN ERDs to a logical model. Create a list of “mapping rules”.*

Once a conceptual model has been created, it is a fairly simple process to convert this into a logical diagram. This guide will be focussed on converting a Chen’s Notation conceptual model to a logical ERD.

The 6 steps will help guide this process:

1. Create Tables
   1. Entities will become tables.
   2. Create new tables for any relationship attributes and connect with an appropriate key.
2. Update Attributes
   1. Attributes will be added as columns in their respective tables.
   2. Convert composite attributes to individual attributes.
   3. Multi-valued attributes become another weak table with a composite key of parent entity ID and item ID.
   4. Add data types and lengths if necessary.
3. Select Primary Keys
   1. Convert appropriate candidate key into primary keys.
   2. Create ID columns if no appropriate candidate keys are present for a table.
4. Create Relationships
   1. 1:1 - Add the foreign key to either table.
   2. 1:M – Add the foreign key to the many table.
   3. Create join tables for many to many relationships.
   4. Add the foreign key to the join table from both tables in the relationship.
5. Super/Subclass
   1. There are multiple options for converting a conceptual Chen Notation generalisation/specialisation structure. Select the method that is most appropriate for the database design and most logical to the designer /developer.
   2. Option 1 – Make one big table that includes all super and subclass attributes. Irrelevant fields will be null.
   3. Option 2 – Create a superclass table with the shared attributes and subclass tables. The subclass tables will have foreign keys to the superclass.
   4. Option 3 – Create a superclass table with the shared attributes and subclass table. The super and subclasses will share primary keys.
6. Check Crow’s Foot Notation
   1. Make sure all key connections have the correct Crow’s Foot notation for the relationship.
      1. Single goes to just a line.
      2. Many goes to crows foot.
      3. Optional adds a circle.
      4. Mandatory/one adds a perpendicular line.



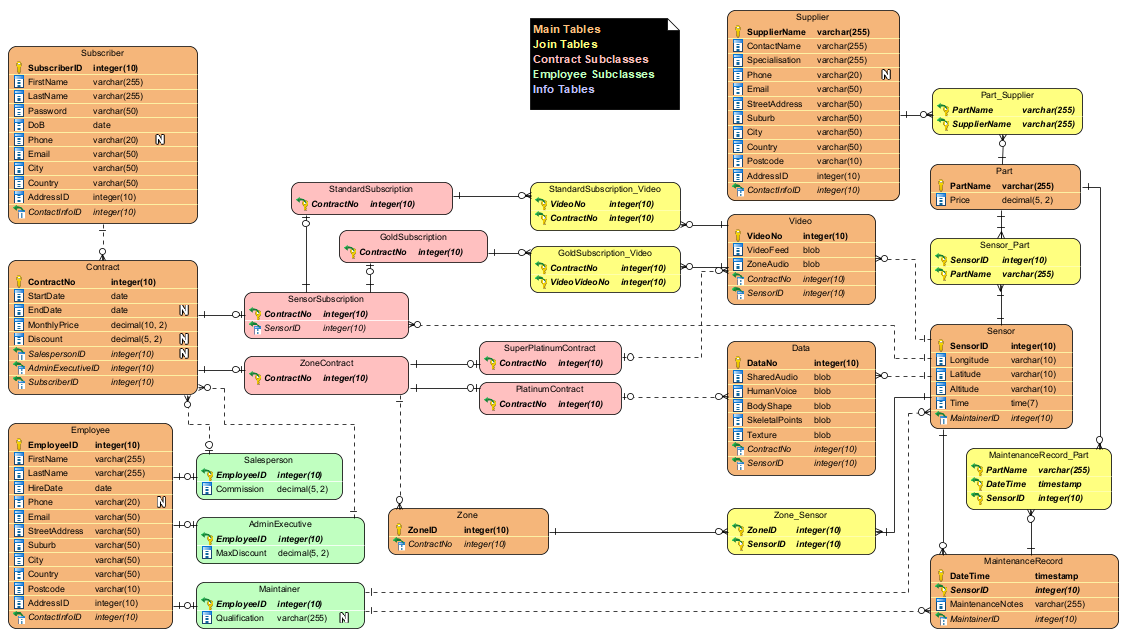
(Stewart, 2008)

# Logical ERD

*Create an ERD using an appropriate notation. Accompany this with your rationale; describe and explain the reasoning and purpose of all parts of the model. This is to demonstrate how you applied the “mapping rules” from the Conceptual to the Logical Model.*

Both diagrams can be viewed as separate attached images.

**Unnormalised ERD:**



The logical ERD was fairly simple to build due to my high-quality conceptual model. The mapping rules (view previous section) created a straightforward and simple guide to follow. I followed the process in the same step-by-step order as in the guide, starting with the tables and attributes, then adding keys and relationship details.

Most data types were very self-explanatory as to what to use, however I have selected the blob type for audio, video and 3d imagery data. I selected this as Spaces© has not specified what data formats they will be using to store data. Blob gives them the most flexibility in what is stored, and this can always be changed in the physical implementation if they require it.

IT may be beneficial for Spaces© to add in a manager attribute and relationship in the employee table, so they can store and see the company hierarchy. However, they have not stated that they wish to do this, so it has been excluded.

In the conceptual model, contract was a weak entity of subscriber, as a contract cannot exist without a subscriber. As contracts can identify themselves just with the contract number, it is not necessary to have the subscriber ID as well. I have decided to convert contract into a strong entity.

Both video and data entities are also now strong entities as with their own primary key identifiers, the parent sensor ID is not required to identify them. I think this makes more logical sense in the current ERD.

One major business change that has been made is allowing zone contracts to have multiple zones. This is a large business change but has a very minor effect on the ERD (1:1 -> 1:M).

I have also added a couple of minor attributes to the ERD as well, such as a hire date to employee and more contact information to subscribers, as I believe that this is useful information for Spaces©.

I considered storing both the date of birth and current age, where current age is calculated from the DoB, however I decided that it was simpler to just calculate age when needed from DOB instead of having to store and update it daily.

# Normalisation

## Forms and Steps

*Include a description of normalisation to 4th Normal form.*

Normalisation is the process of reworking a database to remove errors, anomalies and duplications within the data, based upon specified rules (Muñoz, 2022). Removing duplicate data will reduce the risk of data conflicts and anomalies from occurring, reducing future errors and other issues that may occur. Normalisation can also help improve efficiency and performance in some cases (Romani, 2023). There are many different normal forms, however this assignment will only discuss up to Boyce-Codd and 4th normal form.

As described by Romani (Romani, 2023), the following describes normalisation up to the fourth form.

### Unnormalised (0NF)

* A database with no normalisation.

*Book Table*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Book | Title | Author 1 | Author 2 | Author Name | Genre | Genre Description |
| 1 | Todd’s Terrific Databases | 1 | 2 | Todd Cochrane, Jayden Houghton | Educational | Educational IT non-fiction. |

### First Normal Form (1NF)

* Remove duplicate columns (repeated data).
* No multi-value attributes (atomic data).
* Select primary keys.

*Book Table*

|  |  |  |  |
| --- | --- | --- | --- |
| Book ID | Title | Genre | Genre Description |
| 1 | Todd’s Terrific Databases | Educational | Educational IT non-fiction. |

*Book-Author Table*

|  |  |  |  |
| --- | --- | --- | --- |
| Book ID | Author ID | First Name | Last Name |
| 1 | 1 | Todd | Cochrane |
| 1 | 2 | Jayden | Houghton |

### Second Normal Form (2NF)

* Requires 1NF.
* Full functional dependency on keys.

*Book Table*

|  |  |  |  |
| --- | --- | --- | --- |
| Book ID | Title | Genre | Genre Description |
| 1 | Todd’s Terrific Databases | Educational | Educational IT non-fiction. |

*Book- Author Table*

|  |  |
| --- | --- |
| Book | Author ID |
| 1 | 1 |
| 1 | 2 |

*Book-Author Table*

|  |  |  |
| --- | --- | --- |
| Author ID | First Name | Last Name |
| 1 | Todd | Cochrane |
| 1 | Jayden | Houghton |

### Third Normal Form (3NF)

* Requires 2NF.
* Remove transitive dependencies (non-key attributes dependant on other non-key attributes).

Often the first 3 forms are done together as they work hand in hand.

*Book Table*

|  |  |  |  |
| --- | --- | --- | --- |
| Book ID | Title | Genre | Genre Description |
| 1 | Todd’s Terrific Databases | Educational | Educational IT non-fiction. |

*Book-Genre Table*

|  |  |
| --- | --- |
| Book ID | Genre ID |
| 1 | 1 |

*Book Table*

|  |  |  |
| --- | --- | --- |
| Genre ID | Genre Name | Genre Description |
| 1 | Educational | Educational IT non-fiction. |

*Book- Author Table*

|  |  |
| --- | --- |
| Book | Author ID |
| 1 | 1 |
| 1 | 2 |

*Book-Author Table*

|  |  |  |
| --- | --- | --- |
| Author ID | First Name | Last Name |
| 1 | Todd | Cochrane |
| 1 | Jayden | Houghton |

### Boyce-Codd Normal Form (BCNF)

* Requires 3NF.
* 3NF focusses on removing functional dependencies on non-key attributes, where as BCNF does this for potential key attributes.

Sometimes BCNF is referred to as 3.5 normal form, as it is an extension on the third form. It is less commonly used as it doesn’t always apply to a database and can cause over-normalisation, potentially affecting database performance.

### Fourth Normal Form (4NF)

* Requires BCNF.
* Remove any multi-valued dependency for tables with at least 3 columns.

4NF is very rarely used and is mostly a theoretical practice to further reduce database anomalies.

## Normalised ERD

*Make sure your relations are fully normalised (to 3NF, consider Boyce Codd and 4th Normal forms). Document your relations. Describe how you undertook normalisation of your database design.*

Both ERDs can be viewed as separate attached images.

**Normalised ERD:**A picture containing text, screenshot, font, number

Description automatically generated

All tables are normalised, view the data dictionary or NaLER analysis for a list of relations.

Because of my high-quality conceptual model, after following the mapping rules as stated previously, the ERD fulfilled most normalisation requirements already. The only changes I implemented were separating out the address fields (street address, suburb, city, country, postcode) and contact details (phone number, email address, backup email address) into their own tables. I did this to remove the duplication of those fields between the employee, supplier and subscriber tables. After this change, the ERD meets 3NF.

I also considered moving *FirstName* and *LastName* into the contact table as the fields are duplicated, however this feels like over-normalisation and bad practice.

After normalising to 3NF, I considered any possible BCNF or 4NF issues that would require these normalisation forms to be implemented. I did not identify any violations of these forms and I believe that the database ERD already fulfils these requirements.

# Data Dictionary

*Documentation of your relations. Include a data dictionary – this must be in tabular form as given in class. Document your logical model in a data dictionary.*

* *Entities (e.g. name, start volume, growth)*
* *Attributes (e.g. name, description, data type, length, value range, validation rules, default value, nulls, keys, entity references, integrity constraints)*

## Document relations

|  |  |  |  |
| --- | --- | --- | --- |
| Relation Name | Start Volume No. of rows loaded at the beginning | Growth e.g. no growth / 10% per year | Comments |
| Subscriber | 1,000,000 | 10% |  |
| Employee | 6,000 | 5% | Staff required to grow to match demand |
| Supplier | 100 | 2 | Suppliers should be quite consistent as the required parts won’t change much |
| ContactInfo | 1,006,100 | 15% | All staff, suppliers and employees will have a contact |
| Address | 7,000 | 8,200 | Supplier, staff and zone contracts |
| Salesperson | 900 | 30 |  |
| Admin Executive | 100 | 5 |  |
| Maintainer | 5,000 | 100 | Maintain ~100 sensors each, lots are part-time contractors |
| Contract | 2,000,000 | 10% | Subscribers can have multiple contracts |
| Sensor Subscription | 1,997,900 | 10% | Majority of contracts |
| Standard Subscription | 1,700,000 | 10% | Majority of sensor subscriptions |
| Gold Subscription | 247,900 | 5% |  |
| Standard Subscription \_Video | 1,500,000,000 | 20% | Assume max 100 video streams |
| Gold Subscription \_Video | 447,900,000 | 20% | Assume max 100 video streams |
| Zone Contract | 2,100 | 110 |  |
| Platinum Contract | 2,000 | 100 |  |
| Super Platinum Contract | 100 | 10 |  |
| Zone | 50,000 | 5% | Similar rate as sensors |
| Zone\_Sensor | 600,000 | 10% | Sensors can be in multiple zones, so the amount will be greater than the sensor number |
| Data | 3,000,000,000 | 10% | Sensor\*100 video streams\*60fps month (note) |
| Video | 3,000,000,000 | 10% | Sensor\*100 video streams\*60fps |
| Sensor | 500,000 | 5% |  |
| Maintenance Record | 1,000,000,000 | 5% | Assume 2 per sensor (unknow how long the business has been operating, under 10y operations) |
| Maintenance Record\_Part | 1,000,000,000 | 5% | Some wont need parts, some will need multiple |
| Part | 400 | 5 | Parts likely won’t change much unless a new sensor is implemented |
| Sensor\_Part | 200,000,000 | 8% | parts\*sensors |
| Part\_Supplier | 500 | 4 | Some suppliers produce multiple |

Note: 1 month of data is stored as a backup on the sensor, but there is no backup in the database system itself.

## Document Attributes

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Relation Name | Attribute | Description | Data type | Length | Value range | Validation Rules | Default Value | Null | Key | References Entity | Integrity Constraints |
| Contract | ContractNo | Identifier | int | 10 | 0< |  | Auto |  | PK |  |  |
| StartDate | When the contract started | date |  |  |  |  |  |  |  |  |
| EndDate | Agreed end date if applicable | date |  |  |  |  | N |  |  |  |
| MonthlyPrice | Cost of subscription/contract | decimal | 10,2 | 0< |  |  |  |  |  |  |
| Discount | The discount % applied | decimal | 5,2 | 0<d<100 |  |  | N |  |  |  |
| SalespersonID | Salesperson who sold the contract | int | 10 | 0< |  |  | N |  | Salesperson | FK |
| AdminExecutiveID | Admin Exec who approved the contract | int | 10 | 0< |  |  |  |  | Admin  Executive | FK |
| SubscriberID | Subscriber who made the contract | int | 10 | 0< |  |  |  |  | Maintainer | FK |
| Subscriber | SubscriberID | Identifier | int | 10 | 0< |  | Auto |  | PK |  |  |
| FirstName | Subscriber first name | varchar | 255 |  |  |  |  |  |  |  |
| LastName | Subscriber last name | varchar | 255 |  |  |  |  |  |  |  |
| Password | Account password | varchar | 50 |  | Inc. num and symbol |  |  |  |  |  |
| DoB | Date of birth | date |  |  | 1900-(today-min age) |  |  |  |  |  |
| AddressID | Address link | int | 10 | 0< |  |  |  |  | Address | FK |
| ContactInfoID | Contact link | int | 10 | 0< |  |  |  |  | ContactInfo | FK |
| Employee | EmployeeID | Identifier | int | 10 | 0< |  | Auto |  | PK |  |  |
| FirstName | Employee first name | varchar | 255 |  |  |  |  |  |  |  |
| LastName | Employee last name | varchar | 255 |  |  |  |  |  |  |  |
| HireDate | Date hired | date |  |  |  | Today’s date |  |  |  |  |
| AddressID | Address link | int | 10 | 0< |  |  |  |  | Address | FK |
| ContactInfoID | Contact link | int | 10 | 0< |  |  |  |  | ContactInfo | FK |
| Supplier | SupplierName | Identifier | varchar | 255 |  |  |  |  | PK |  |  |
| ContactName | First name | varchar | 255 |  |  |  |  |  |  |  |
| Specialisation | Area of manufacturing | varchar | 255 |  |  |  |  |  |  |  |
| AddressID | Address link | int | 10 | 0< |  |  |  |  | Address | FK |
| ContactInfoID | Contact link | int | 10 | 0< |  |  |  |  | ContactInfo | FK |
| ContactInfo | ContactID | Identifier | int | 10 | 0< |  | Auto |  | PK |  |  |
| Phone | Phone number (mobile or landline) | varchar | 20 |  |  |  | N |  |  |  |
| Email | Email address | varchar | 50 |  |  |  |  |  |  |  |
| SecondaryEmail | Back up email | varchar | 50 |  |  |  | N |  |  |  |
| Address | AddressID | Identifier | int | 10 | 0< |  | Auto | N | PK |  |  |
| StreetAddress | Street and number | varchar | 50 |  |  |  | N |  |  |  |
| Suburb | Suburb name | varchar | 50 |  |  |  | N |  |  |  |
| City | City name | varchar | 50 |  |  |  |  |  |  |  |
| Country | Country | varchar | 50 |  |  |  |  |  |  |  |
| Postcode | Postcode | varchar | 10 |  |  |  | N |  |  |  |
| Sensor\_  Subscription | ContractNo | Superclass key | int | 10 |  | =superclass |  |  | PK | Contract | FK |
| SensorID | Identifier | int | 10 |  |  |  |  |  | Sensor | FK |
| Zone\_ Contract | ContractNo | Superclass key | int | 10 |  | =superclass |  |  | PK | Contract | FK |
| Standard  Subscription | ContractNo | Superclass key | int | 10 |  | =superclass |  |  | PK | Sensor Subscription | FK |
| Gold Subscription | ContractNo | Superclass key | int | 10 |  | =superclass |  |  | PK | Sensor Subscription | FK |
| Super Platinum  Contract | ContractNo | Superclass key | int | 10 |  | =superclass |  |  | PK | Zone Contract | FK |
| Platinum Contract | ContractNo | Superclass key | int | 10 |  | =superclass |  |  | PK | Zone Contract | FK |
| Standard  Subscription  \_Video | VideoNo | Identifier | int | 10 |  |  |  |  | PK | Video | FK |
| ContractNo | Superclass key | int | 10 |  | <100 per contract |  |  | PK | Standard  Subscription | FK |
| Gold Subscription  \_Video | VideoNo | Identifier | int | 10 |  |  |  |  | PK | Video | FK |
| ContractNo | Superclass key | int | 10 |  |  |  |  | PK | Gold  Subscription | FK |
| Salesperson | EmployeeID | Identifier | int | 10 |  | =superclass |  |  | PK | Employee | FK |
| Commission | Commission per sale | decimal | 5,2 |  |  | 0 |  |  |  |  |
| Admin Executive | EmployeeID | Identifier | int | 10 |  | =superclass |  |  | PK | Employee | FK |
| MaxDiscount | Max discount allowed | decimal | 5,2 |  |  | 0 |  |  |  |  |
| Maintainer | EmployeeID | Identifier | int | 10 |  | =superclass |  |  | PK | Employee | FK |
| Qualification | Qualification/certifications |  |  |  |  |  | N |  |  |  |
| Zone | ZoneID | Identifier | int | 10 | 0< |  | Auto |  | PK |  |  |
| ContractNo | Superclass key |  |  |  | =superclass |  |  |  | Zone Contract | FK |
| Zone\_ Sensor | ZoneID | Identifier | int | 10 |  |  |  |  | PK | Zone | FK |
| SensorID | Identifier | int | 10 |  |  |  |  | PK | Sensor | FK |
| Data | DataNo | Identifier | int | 10 | 0< |  | Auto |  | PK |  |  |
| SharedAudio | Shared audio from other sources | blob |  |  |  |  |  |  |  |  |
| HumanVoice | Recorded human voice | blob |  |  |  |  |  |  |  |  |
| BodyShape | 3d human body shape | blob |  |  |  |  |  |  |  |  |
| SkeletalPoints | Estimated 3d human skeletal points | blob |  |  |  |  |  |  |  |  |
| Texture | Human texture | blob |  |  |  |  |  |  |  |  |
| ContractNo | Superclass key | int | 10 |  |  |  |  |  | Platinum Contract | FK |
| SensorID | Identifier | int | 10 |  |  |  |  |  | Sensor | FK |
| Video | VideoNo | Identifier | int | 10 | 0< |  | Auto |  |  |  |  |
| VideoFeed | Recorded video | blob |  |  |  |  |  |  |  |  |
| ZoneAudio | Zone audio | blob |  |  |  |  |  |  |  |  |
| ContractNo | Superclass key | int | 10 |  |  |  |  |  | Standard Subscription  \_Video | FK |
| SensorID | Identifier | int | 10 |  |  |  |  |  | Sensor | FK |
| Sensor | SensorID | Identifier | int | 10 | 0< |  | Auto |  | PK |  |  |
| Longitude | Longitude position | varchar | 10 |  |  |  |  |  |  |  |
| Latitude | Latitude position | varchar | 10 |  |  |  |  |  |  |  |
| Altitude | Altitude | varchar | 10 |  |  |  |  |  |  |  |
| Time | Current time | time | 7 |  |  |  |  |  |  |  |
| MaintainerID | Identifier | int | 10 |  |  |  |  |  | Maintainer | FK |
| Maint-enance  Record | DateTime | Date and time | timestamp |  |  |  |  |  | PK |  |  |
| SensorID | Identifier | int | 10 |  |  |  |  | PK | Sensor | FK |
| Maintenance  Notes | Observations, part changes, issues, etc. | varchar | 255 |  |  | Routine check. |  |  |  |  |
| MaintainerID | Identifier | int | 10 |  |  |  |  |  | Maintainer | FK |
| Maintenance  Record\_Part | PartName | Part name | varchar | 255 |  | <today |  |  | PK | Part | FK |
| DateTime | Maintenance date+time | timestamp |  |  |  |  |  | PK | Maintenance Record | FK |
| SensorID | Identifier | int | 10 | 0< |  |  |  | PK | Maintenance Record | FK |
| Part | PartName | Identifier | varchar | 255 |  |  |  |  | PK |  |  |
| Price | Item price | decimal | 5,2 | 0< |  |  |  |  |  |  |
| Sensor\_Part | SensorID | Identifier | int | 10 |  |  |  |  | PK | Sensor | FK |
| PartName | Identifier | varchar | 255 |  |  |  |  | PK | Part | FK |
| Part\_Supplier | PartName | Identifier | varchar | 255 |  |  |  |  | PK | Part | FK |
| SupplierName | Identifier | varchar | 255 |  |  |  |  | PK | Supplier | FK |

# NaLER Analysis

*Carry out a NaLER analysis on the ERD. Make any changes to your logical model.*

## Process

NaLER is a natural language method for interpreting ERDs. It uses present sentence formats to identify any errors or holes in an ERD.

A NaLER analysis can be done in 6 steps:

1. Identify and document diagram conventions.
2. Perform a syntax check of the model.
3. Construct sentences attribute and binary relationships
   1. Construct a sentence for the primary key attributes.
   2. Construct a sentence for each non-key attribute.
   3. Construct a sentence for each binary relationship.
4. Populate the sentences from 3.2 ad 3.3. with valid examples.
5. Construct m:m relationship sentences
   1. Construct a sentence for any ternary or higher m:m relationships with a composite name.
   2. Construct a sentence for any ternary or higher m:m relationships with clear identifying name.
6. Produce a NaLER description by listing all constructed sentences with examples.

## Analysis

**Conventions:**

* Pascal case is used for all naming (e.g. FirstName).
* Table names are singular (e.g. Customer not Customers).
* Join tables join the table names with an underscore (e.g. Zone\_Sensor).
* Crow’s foot notation is used to show relationship nature.

### Entities

Each Subscriber is uniquely identified by a SubscriberID.

*One* Subscriber *identified by* SubscriberID *must have one* FirstName.

*One* Subscriber *identified by* SubscriberID *must have one* LastName.

*One* Subscriber *identified by* SubscriberID *must have one* Password.

*One* Subscriber *identified by* SubscriberID *must have one* DoB.

*One* Subscriber *identified by* SubscriberID *must have one* AddressID.

*One* Subscriber *identified by* SubscriberID *must have one* ContactInfoID.

Each Employee is uniquely identified by a EmployeeID.

*One* Employee *identified by* EmployeeID *must have one* FirstName.

*One* Employee *identified by* EmployeeID *must have one* LastName.

*One* Employee *identified by* EmployeeID *must have one* HireDate.

*One* Employee *identified by* EmployeeID *must have one* AddressID.

*One* Employee *identified by* EmployeeID *must have one* ContactInfoID.

Each Supplier is uniquely identified by a SupplierName.

*One* Supplier *identified by* SupplierName *must have one* ContactName.

*One* Supplier *identified by* SupplierName *must have one* Specialisation.

*One* Supplier *identified by* SupplierName *must have one* AddressID.

*One* Supplier *identified by* SupplierName *must have one* ContactInfoID.

Each ContactInfo is uniquely identified by a ContactInfoID.

*One* ContactInfo *identified by* ContactInfoID *may have one* Phone.

*One* ContactInfo *identified by* ContactInfoID *must have one* Email.

*One* ContactInfo *identified by* ContactInfoID *may have one* SecondaryEmail.

Each Address is uniquely identified by a AddressID.

*One* Address *identified by* AddressID *may have one* StreetAddress.

*One* Address *identified by* AddressID *may have one* Suburb.

*One* Address *identified by* AddressID *must have one* City.

*One* Address *identified by* AddressID *must have one* Country.

*One* Address *identified by* AddressID *may have one* Postcode.

Each Salesperson is uniquely identified by a EmployeeID.

*One* Salesperson *identified by* EmployeeID *must have one* Commission.

Each AdminExecutive is uniquely identified by a EmployeeID.

*One* AdminExecutive *identified by* EmployeeID *must have one* MaxDiscount.

Each Maintainer is uniquely identified by a EmployeeID.

*One* Maintainer *identified by* EmployeeID *may have one* Certification.

Each Contract is uniquely identified by a ContractNo.

*One* Contract *identified by* ContractNo *must have one* StartDate.

*One* Contract *identified by* ContractNo *may have one* EndDate.

*One* Contract *identified by* ContractNo *must have one* MonthlyPrice.

*One* Contract *identified by* ContractNo *may have one* Discount.

*One* Contract *identified by* ContractNo *must have one* Discount.

*One* Contract *identified by* ContractNo *may have one* SalespersonID.

*One* Contract *identified by* ContractNo *must have one* AdminExecutiveID.

*One* Contract *identified by* ContractNo *must have one* SubscriberID.

Each SensorSubscription is uniquely identified by a ContractNo.

*One* SensorSubscription *identified by* ContractNo *must have one* SensorID.

Each StandardSubscription is uniquely identified by a ContractNo.

Each GoldSubscription is uniquely identified by a ContractNo.

Each StandardSubscription\_Video is uniquely identified by a ContractNo and a VideoNo.

Each GoldSubscription\_Video is uniquely identified by a ContractNo and a VideoNo.

Each ZoneContract is uniquely identified by a ContractNo.

Each PlatinumContract is uniquely identified by a ContractNo.

Each SuperPlatinumContract is uniquely identified by a ContractNo.

Each Zone is uniquely identified by a ZoneID.

*One* Zone *identified by* ZoneID *must have one* ContractNo.

Each Zone\_Sensor is uniquely identified by a ZoneID and a SensorID.

Each Data is uniquely identified by a DataNo.

*One* Data *identified by* DataNo *must have one* SharedAudio.

*One* Data *identified by* DataNo *must have one* HumanVoice.

*One* Data *identified by* DataNo *must have one* BodyShape.

*One* Data *identified by* DataNo *must have one* SkeletalPoints.

*One* Data *identified by* DataNo *must have one* Texture.

*One* Data *identified by* DataNo *must have one* ContractNo.

*One* Data *identified by* DataNo *must have one* SensorID.

Each Video is uniquely identified by a VideoNo.

*One* Video *identified by* VideoNo *must have one* VideoFeed.

*One* Video *identified by* VideoNo *must have one* ZoneAudio.

*One* Video *identified by* VideoNo *must have one* ContractNo.

*One* Video *identified by* VideoNo *must have one* SensorID.

Each Sensor is uniquely identified by a SensorID.

*One* Sensor *identified by* SensorID *must have one* Longitude.

*One* Sensor *identified by* SensorID *must have one* Latitude.

*One* Sensor *identified by* SensorID *must have one* Altitude.

*One* Sensor *identified by* SensorID *must have one* Time.

*One* Sensor *identified by* SensorID *must have one* MaintainerID.

Each MaintenanceRecord is uniquely identified by a DateTime and a SensorID

*One* MaintenanceRecord *identified by* DateTime and SensorID *must have one* MaintenanceNotes.

*One* MaintenanceRecord *identified by* DateTime and SensorID *must have one* MaintainerID.

Each MaintenanceRecord\_Part is uniquely identified by a PartName, DateTime and a SensorID.

Each Part is uniquely identified by a PartName.

*One* Part *identified by* PartName *must have one* Price.

Each Sensor\_Part is uniquely identified by a SensorID and a PartName.

Each Part\_Supplier is uniquely identified by a PartName and a SupplierName.

### Relationships

The relationships are grouped by table. Each relationship has a constructed sentence and an example sentence.

*One* Subscriber *identified by* SubscriberID *must have one* ContactInfo *identified by* ContactInfoID.

*One* Subscriber *identified by* SubscriberID *must have one* ContactInfo *identified by* ContactInfoID.

*One* Subscriber *identified by* SubscriberID *must have one* Address *identified by* AddressID.

*One* Subscriber *identified by* SubscriberID *may pay for one or more* Contracts *identified by* ContractNo.

*One* Employee *identified by* EmployeeID *must have one* ContactInfo *identified by* ContactInfoID.

*One* Employee *identified by* EmployeeID *must have one* Address *identified by* AddressID.

*One* Employee *identified by* EmployeeID *may also be one* Salesperson *identified by* EmployeeID.

*One* Employee *identified by* EmployeeID *may also be one* AdminExecutive *identified by* EmployeeID.

*One* Employee *identified by* EmployeeID *may also be one* Maintainer *identified by* EmployeeID.

*One* Salesperson *identified by* EmployeeID may have sold one or more Contracts *identified by* ContractNo.

*One* Salesperson *identified by* EmployeeID belongs to one Employee *identified by* EmployeeID.

*One* AdminExecutive *identified by* EmployeeID may have approved (and discounted) one or more Contracts *identified by* ContractNo.

*One* AdminExecutive *identified by* EmployeeID belongs to one Employee *identified by* EmployeeID.

*One* Maintainer *identified by* EmployeeID may be maintaining one or more Sensors *identified by* SensorID.

*One* Maintainer *identified by* EmployeeID may have written one or more maintenanceRecords *identified by* DateTime and SensorID.

*One* Maintainer *identified by* EmployeeID belongs to one Employee *identified by* EmployeeID.

*One* Supplier *identified by* SupplierID *must have one* ContactInfo *identified by* ContactInfoID.

*One* Supplier *identified by* SupplierID *must have one* Address *identified by* AddressID.

*One* Supplier *identified by* SupplierID *may be the Part\_Supplier for one or more* Parts *identified by* PartName.

*One* ContactInfo *identified by* ContactInfoID *may have one* Employee *identified by* EmployeeID.

*One* ContactInfo *identified by* ContactInfoID *may have one* Subscriber *identified by* SubscriberID.

*One* ContactInfo *identified by* ContactInfoID *may have one* Supplier *identified by* SupplierName.

*One* Address *identified by* AddressID *may have one* Employee *identified by* EmployeeID.

*One* Address *identified by* AddressID *may have one* Subscriber *identified by* SubscriberID.

*One* Address *identified by* AddressID *may have one* Supplier *identified by* SupplierName.

*One* Contract *identified by* ContractNo may be sold by one or more Salespeople *identified by* EmployeeID.

*One* Contract *identified by* ContractNo must be approved (and discounted) by one AdminExecutive *identified by* EmployeeID.

*One* Contract *identified by* ContractNo must be paid for/owned by one Subscriber *identified by* SubscriberID.

*One* Contract *identified by* ContractNo may include one SensorSubscription *identified by* ContractNo.

*One* Contract *identified by* ContractNo may include one ZoneContract *identified by* ContractNo.

*One* SensorSubscription *identified by* ContractNo may include one StandardSubscription *identified by* ContractNo.

*One* SensorSubscription *identified by* ContractNo may include one GoldSubscription *identified by* ContractNo.

*One* SensorSubscription *identified by* ContractNo must have one Sensor *identified by* SensorID.

*One* SensorSubscription *identified by* ContractNo must belong to one Contract *identified by* ContractNo.

*One* StandardSubscription *identified by* ContractNo may be the StandardSubscription\_Video (viewed) for one or more Videos *identified by* VideoNo.

*One* StandardSubscription *identified by* ContractNo must belong one Contract *identified by* ContractNo.

*One* GoldSubscription *identified by* ContractNo may be the GoldSubscription\_Video (controlled) for one or more Videos *identified by* VideoNo.

*One* GoldSubscription *identified by* ContractNo must belong one Contract *identified by* ContractNo.

*One* ZoneContract *identified by* ContractNo may include one PlatinumContract *identified by* ContractNo.

*One* ZoneContract *identified by* ContractNo may include one SuperPlatinumContract *identified by* ContractNo.

*One* ZoneContract *identified by* ContractNo must have one Zone *identified by* ZoneID.

*One* ZoneContract *identified by* ContractNo must belong to one Contract *identified by* ContractNo.

*One* PlatinumContract *identified by* ContractNo may link to one or more pieces of Data *identified by* DataNo.

*One* PlatinumContract *identified by* ContractNo must belong one ZoneContract *identified by* ContractNo.

*One* SuperPlatinumContract *identified by* ContractNo may have exclusive access to one or more Videos *identified by* VideoNo.

*One* SuperPlatinumContract *identified by* ContractNo must belong one ZoneContract *identified by* ContractNo.

*One* Zone *identified by* ZoneID may be the zone (Zone\_Sensor) for one or more Sensors *identified by* SensorID.

*One* Zone *identified by* ZoneID must belong one ZoneContract *identified by* ContractNo.

*One* Video *identified by* VideoNo may have a StandardSubscription\_Video (viewed by) for one or more StandardSubscriptions *identified by* ContractNo.

*One* Video *identified by* VideoNo may have a GoldSubscription\_Video (controlled by) for one or more GoldSubscriptions *identified by* ContractNo.

*One* Video *identified by* VideoNo may have exclusive access by one SuperPlatinumContract *identified by* ContractNo.

*One* Video *identified by* VideoNo must be recorded by one Sensor *identified by* SensorNo.

*One* piece of Data *identified by* DataNo may be accessed by one PlatinumContract *identified by* ContractNo.

*One* piece of Data *identified by* DataNo must be recorded by one Sensor *identified by* SensorNo.

*One* Sensor *identified by* SensorID may record one or more Videos *identified by* VideoNo.

*One* Sensor *identified by* SensorID may record one or more pieces of Data *identified by* DataNo.

*One* Sensor *identified by* SensorID may be linked to one or more SensorSubscriptions *identified by* VideoNo.

*One* Sensor *identified by* SensorID must be maintained by one Maintainer *identified by* EmployeeID.

*One* Sensor *identified by* SensorID may record one or more Videos *identified by* VideoNo.

*One* Sensor *identified by* SensorID must be a sensor in a Zone\_Sensor for one or more Zones *identified by* ZoneID.

*One* Sensor *identified by* SensorID must have a sensor in a Sensor\_Part for one or more Parts *identified by* PartName.

*One* Part *identified by* PartName may have Part\_Suppliers for one or more Suppliers *identified by* SupplierName.

*One* Part *identified by* PartName must have Sensor\_Parts for one or more Sensors *identified by* SensorID.

*One* Part *identified by* PartName may have SensorMaintenanceRecord\_Parts for one or more MaintenanceRecords *identified by* DateTime and SensorID.

*One* MaintenanceRecord *identified by* DateTime and SensorID must belong to Sensor *identified by* SensorID.

*One* MaintenanceRecord *identified by* DateTime and SensorID may have MaintenanceRecord\_Parts for one or more Parts *identified by* PartName.

*One* MaintenanceRecord *identified by* DateTime and SensorID must be written by one Maintainer *identified by* EmployeeID.

# References

Muñoz, A. (2022, August 23). *Why is Database Normalization so Important?* Retrieved from Sales Layer: https://blog.saleslayer.com/why-is-database-normalization-so-important#:~:text=Normalization%20is%20an%20essential%20part,and%20making%20it%20more%20concise.

Romani, E. (2023, March 2). *What Is Database Normalization?* Retrieved from Built In: https://builtin.com/data-science/database-normalization

Stewart, J. (2008, June 1). *Crow’s Feet Are Best*. Retrieved from The Data Administration Newsletter: https://tdan.com/crows-feet-are-best/7474#