Aiden Inglis

NMIT  322 Hardy St Nelson 7010

DAT601 – milstone 1

Contents

[Description of conceptual modelling: 2](#_Toc194869999)

[Common components of a entity Chen diagram/ERD: 2](#_Toc194870000)

[Entities: 2](#_Toc194870001)

[Attributes: 2](#_Toc194870002)

[Relationships: 2](#_Toc194870003)

[Relationships Degrees 2](#_Toc194870004)

[Chen conceptual Entity relationship model: 2](#_Toc194870005)

[Reasoning and purpose of all parts of the model: 2](#_Toc194870006)

[Data Dictionary: 4](#_Toc194870007)

[Entities: 4](#_Toc194870008)

[Relationships: 7](#_Toc194870009)

[Attributes: 8](#_Toc194870010)

[Director: 8](#_Toc194870011)

[Executive Administrator 9](#_Toc194870012)

[Employee: 11](#_Toc194870013)

[Salesman: 12](#_Toc194870014)

[Engineer: 12](#_Toc194870015)

[Subscription: 12](#_Toc194870016)

[Customer 12](#_Toc194870017)

[Contract 13](#_Toc194870018)

[Invoice: 13](#_Toc194870019)

[Region: 14](#_Toc194870020)

[Zone: 15](#_Toc194870021)

[Data: 15](#_Toc194870022)

[FRED: 16](#_Toc194870023)

[Service: 17](#_Toc194870024)

[Maintainer Company: 17](#_Toc194870025)

[Maintenance: 18](#_Toc194870026)

[Maintainer: 18](#_Toc194870027)

[Maintenance Schedule: 19](#_Toc194870028)

[Parts: 20](#_Toc194870029)

[Parts Supplier: 20](#_Toc194870030)

[Standard: 20](#_Toc194870031)

[Gold: 21](#_Toc194870032)

[Platinum: 21](#_Toc194870033)

[Super Platinum: 22](#_Toc194870034)

[Assumed Business rules with reasoning: 23](#_Toc194870035)

[REFERENCES 23](#_Toc194870036)

# Description of conceptual modelling:

Conceptual modelling is a vital part of information systems and analysis as it allows designers/project managers to create models of the requirements of each system by visually diagramming the relationships between entities and their attributes. The conceptual model helps developers, businesses analysts, directors gain a further understanding of their business workflow by seeing it in a visual representation and helps spotlight issues in the workflow.

Fundamental principles of effective data management:

**Naming and Conventions:**

Make sure all naming and data types are all standardized meaning its easy to know a name of something without looking it up in the system. (e.g. an ID of a customer would be Customer\_ID)

Sticking to strict data format rules such as datetime HH:MM:SS, and DD:MM:YYYY.(*7 Best Practices for Successful Data Management | Tableau*, n.d.)

**Data Storage Management:**

Keeping good storage of your data is very important in todays world as an error or loss of data could lead to incredible legal complexities and unexpected financial losses.

To ensure that you have good data, try the 3-2-1 data storage methodology, this is where you have three copies of your data, where 3 are stored anywhere locally, 2 are located on different devices and 1 copy is located on the cloud. This means that total data loss is close to impossible using this method as you can have so many backup methods to restore data.(*7 Best Practices for Successful Data Management | Tableau*, n.d.)

**Data Security:**

Good data security means you have secure applications that store your data, and this data cannot be seen or accessed by any unauthorized third party. Data breaches are one of the most common and dangerous scenarios to have on your plate, which is why knowing and protecting yourself from risks is important, so keeping a plan in case to prevent it from happening by auditing users and access or inactivity tokens could help reduce some concerns.

**Data Strategy Roadmap:**

This is where all data has a vision of where it begins to where it ends, this means your identifying all data and figuring out what to do with it all for storage, security, documentation which will increase your data quality.(Team, n.d.)

Relationship between Conceptual, Logical data modelling and Physical implementation.

Conceptual Model:

A conceptual model is a first iteration of design at the highest level, which defines what parameters and entities we are working with along with the relationships between the entities.

Logical Model:

A logical model is basically the conceptual model except we include more details in the diagram such as cardinality, fk/pk, datatypes and normalisation/participations.

Physical Model:

A physical model is a process of designing the actual database which can be done in programs such as visual paradigm. The physical model is a extension of the logical model which specifies the exact protocols, systems, entities, attributes, Primary/Foreign Keys and operations that will be used.

# Common components of a entity Chen diagram/ERD:

**An introduction to conceptual modelling using Chen ERDs with a depiction and description of all components of a Chen ERD, include extended components.**

**Describe how the fundamental principles of data management are applied through of the extended Chen ERD in enterprise modelling. (10 marks) (LO1, LO4)**

The conceptual model using a Chen Extended Entity Relationship Diagram is a amazing tool that is used by lots of people worldwide, it shows people the high-level operations and relationships of entities in a business which can be shown to stakeholders, directors, analysts and managers. This diagram shows the entities, their attributes and the relationships between them which also shows cardinality and normalisation.   
The EERD goes further into depth than a normal ERD as it reveals types of entities like weak and associative entities. The Attributes are also more details in this extended version of the diagram as they can now have derived and composite attributes.

In enterprise data modelling using a Chen EERD means that all persons involved with the company can understand the basic flows and relationships of the enterprise without needing any technical knowledge. It provides a more In-Depth version of the ERD doesn’t like catching errors in data types early before the logical diagram is made, this more in-depth version means that more can be reviewed earlier on, and all the entities and there relationships can be clarified better provided less room for error meaning better data consistency and design.

**How fundamental principles of data management are applied through the EERD in enterprise modelling:**

The principles like data security, integrity, conventions and design directly support enterprising companies through the EERD by the use of good first design iteration. The EERD principles are applied thoroughly though my EERD, an example of this is naming conventions where “supply company” would be “SupplyCompany” and also the relationships should have a one-two word description of what happens in the relationship between the two entities. Data Manage can be shown through the detail and errors found and the review process and update process the diagram has gone through, this can be known as there may not be too many error upon first looking but as you look closer you might notice one. This wouldn’t be the case if there was bad data management as if you have good processes you would review the flows of data and find any new problems that arise. Data roadmap is used in this diagram as its directly represented, it shows a in-depth look at where data is created and where the data is going to go and end up, and whos going to use and look at such data, the diagram below shows that the subscription can be made on the top right, but that its still represented through the contract and then through the assigned zones that the freds are assigned to and their limitations provided through that same subscription.

## Entities:

Strong Entity:

An entity is a thing that is represented in a system like “service”, “manager” or “maintenance”. Has primary key

Weak Entity:  
is an entity that relies on another entity to exist, so an invoice wouldn’t have existed without t a contract.

## Attributes:

Attributes:

Attributes are a property of an entity.

Key Attributes:

A key attribute is a key property that can be used as an identifier.

Derived Attributes:

Is an attribute that is made from another attribute in the same entity eg age from date of birth.

Multi-valued Attributes:

Is a attribute that can store multiple values, EG if someone has two email addresses or two phone numbers they wish to store, that is what this attribute type can do.(*Multivalued Attribute - an Overview | ScienceDirect Topics*, n.d.)

Composite Attributes:

Can be broken into smaller pieces eg, address can have multiple sub attributes that make it up like street, house number and postal code.

## Relationships:

Cardinality:

We have One to One, One to Many, Many to One, Many to Many.

Participation:  
Full(required) or Partial(optional)

Generalization/Specifications:

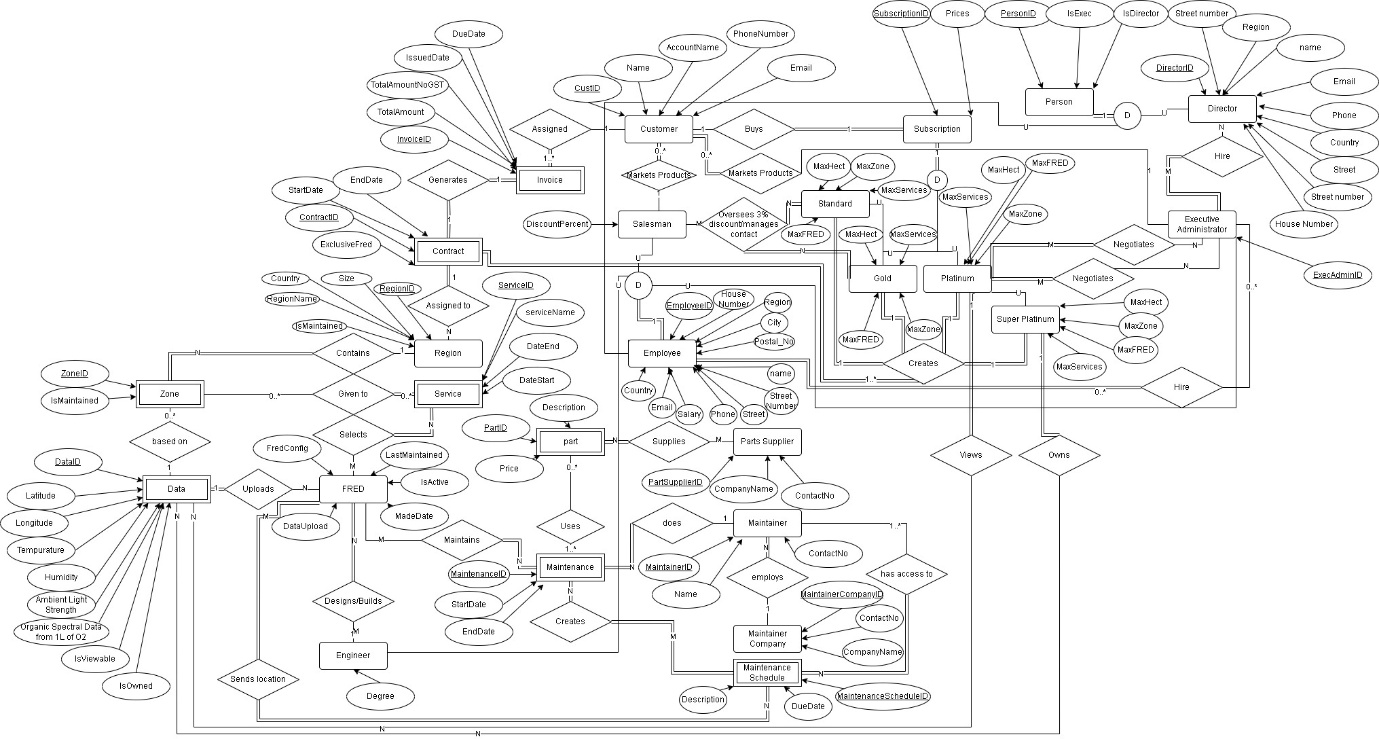
This can be represented by the person entity and how it is the superclass and then director and employee entities inherit attributes from It, and then from that salesman and engineer and executive administrator also inherit directly from employee.

Weak Entities and their identifying relationships are described:

A Weak Entity is an entity which cannot be identified by its attributes, this means that it will require a foreign key/s to identify what its related to.(‘Weak Entity’, 2024)  
  
To identify a weak entity, check if it has a primary key, if it doesn’t then the next fact to check is if it has foreign keys if it does that’s good.

A weak Entity is commonly used to join two different entities, like contract to a customer, there is a invoice, and a subscription. In our scenario, when a subscription is bought then a contract is made, and then an invoice is made off of the contract. But without a contract the invoice wont have any way to identify itself to its customer as it was connected through the customer foreign key in contract, so this makes it a weak entity.

# Chen conceptual Entity relationship model:



## Reasoning and purpose of all parts of the model:

* Many Maintainers is employed by 1 maintainer company. Relationship is forced as the maintainer needs have a company to hire them.
* Maintenance forced Many to 1 relationship with maintainer because a maintainer may not have done any maintenance yet but all maintenance needs a maintainer.
* 1 Maintenance forced weak relationship with 1 maintenance record, this is because a maintenance record relies on the maintenance to exist so its forced, and for each maintenance record needs a maintenance and vice-versa.
* Maintenance doesn’t need to use any parts but parts need to be used by maintenance and parts wouldn’t exist if there were no freds to fix so it is a weak entity.
* All parts need to be supplied by a supplier but not all suppliers need to supply parts. Many to many as many companies can supply many parts.
* Maintenance maintains a FRED is a forced relationship where the fred doesn’t always need maintenance, but the maintenance always needs to be assigned to a FRED. Many maintenances can be done on many FREDs as they can receive more than 1.
* FREDs are designed and built by the engineers, Many FREDs can be built by 1 Engineer (we will assume that 1 engineer can make a FRED by himself) A FRED must be made by a engineer but a engineer doesn’t have to make FREDs as they may have not made one yet.
* 1 FRED doesn’t need to upload 0..\* data to the database, as it may not have done any work yet. However, all data must come from a fred so that relationship is forced.
* 1 Parkwork database has data on 0 to many zones.
* 0 to Many FREDs can select 0 to Many services to do. The services must be done by a drone so it’s a forced relationship and it relies on the FRED for its existence so service is a weak entity as well.
* 0..\* services can be given to 0..\* zones, this is because not all zones need services. All services require a zone which is why its forced.
* Many zones can exists in 1 region. Zones need a region and regions don’t need to have zones so the zone side relationship is forced.
* A contract needs to be assigned to a region so its forced, 1 contract can have many regions assigned to it.
* 1 contract generates 1 invoice, an invoice can’t exist without a contract, so invoice is a weak entity. Both relationships are forced as an invoice needs to be produced by a contract and a contract needs to produce a invoice.
* 1 to Many invoices can be assigned to 1 customer making it a forced relationship as the minimum is one. Also an invoice cant exist without a customer so also confirming it’s a weak entity.
* 1 customer can buy 1 subscription, a subscription needs to be assigned to a customer so it’s a forced relationship.
* 0 to Many Directors can hire 0 to Many, meaning not all directors need to hire someone and a director that is hiring doesn’t have to hire 1. An Executive administrator needs to be hired by a director so it’s a forced relationship.
* An 0..\* executive Administrator hires 0..\* Employees. This means that not all admins need to hire an employee and an employee needs to be hired so that’s a forced relationship.
* An employee can either be a salesman or an engineer shown by the disjoint symbol “D” which makes an employee be one or the other and not both. An engineer and salesman is a employee which explains the ‘U’ aka subset symbol.
* A salesman can market subscriptions to 0..\* customers. One customer needs to have a salesman if they want a standard or gold subscription. Which makes it a forced relationship.
* A Executive administrator can market subscriptions to 0..\* customers. One customer needs to have a Executive Administrator if they want a Platinum or Super Platinum subscription. Which makes it a forced relationship.
* Many salesman can manage and oversee 3% discounts on many gold/standard subscriptions. Each gold/standard subscription needs to have a salesman but not vise-versa so it’s a forced relationship.
* Many Executive Administrators can Negotiate on many Platinum/Super Platinum subscriptions. Each Platinum/Super Platinum subscription needs to have a Executive Administrator but not vise-versa so it’s a forced relationship.
* An Subscription can either be a Standard, Gold, Platinum, or Super Platinum shown by the disjoint symbol “D” which makes an Subscription be one of them and not more than one type. A Standard, Gold, Platinum, or Super Platinum is a subscription which explains the ‘U’ aka subset symbol.
* Each subscription made no matter the type will create 1 to many contracts. The contract needs to have a subscription so it’s a forced relationship there.
* Many FRED upload their locations to the Maintenance Schedule it is forced from the FREDs as they need to do it.
* Many Maintenance Schedules can be accessed by 1 to many maintainers. All schedules must be assigned to a maintainer so that’s a forced relationship.
* Directors can only hire executives, and executives can only hire employees that aren’t executives, forced relationship.
* A person is everyone to a director and all employees forces as they must be.
* Maintenance provides a maintenance schedule based on the freds last maintenance or whether they are fixed. Full partitipation as both are required for each to happen, schedule cant be made without any maintenance, and no maintenance can’t be done without a schedule.

# Data Dictionary:

## Entities:

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity Name** | **Description** | **Aliases** | **Occurrence** |
| Director | The directors of the company | Big Boss, leader, superintendent | 1 record for how many directors there are. |
| Executive Administrator | The exec admin that does platinum + subscriptions and writes contracts and hires employees. | Boss, Chief, Supervisor, overseer | 1 record for how many exec admins there are. |
| Person | Everyone is a person is either a director or employee, which will be the exec. | human | This record exists for each person that is an employee exec or director. |
| Employee | Is a person that works for the company doing tasks | Worker | 1 record for each employee. |
| Salesman | Specifically sells subscriptions to customers. | Marketer | 1 record for each salesman employee. |
| Engineer | Designs and builds the FREDs. | Designer, builder, mechanical engineer, electrical engineer | 1 record for each engineer |
| Customer | Can buy subscriptions and have contracts for FREDs to do maintenance on. | Visitor, Supporter, Guest, Sponsor | 1 record for each customer who has a subscription or is looking to buy a subscription |
| Subscription | Is what the customer buys which has 1 to many contracts. Which is paid in monthly or annually. | Plan, membership | 1 record for each subscription |
| Gold | Gold level subscription, 3 FRED, 3 Zones 100Hect Max | Plan, membership | 1 record for each gold subscription |
| Standard | standard level subscription, 1 FRED, 1 Zones 10Hect Max | Plan, membership | 1 record for each standard subscription |
| Platinum | platinum level subscription, details negotiated by executive. | Plan, membership | 1 record for each platinum subscription |
| Super Platinum | Super platinum level subscription, details negotiated by executive. | Plan, membership | 1 record for each super platinum subscription |
| Invoice | A list of services billed to the customer as per the contracts pricing. | Statement, bill, charge | 1 record for every contract |
| Contract | A details list of what services, Zones, Timelines, Payments, signatures are listed to describe what is expected out of the agreement. | Agreement, arrangement, pact, bargain | 1 record for every contract made from each subscription |
| Region | A large area in which contains zones. | Section, area, location, site, place | 1 record for every region monitored by FREDs |
| Zone | A portion of an area located inside a region. | Area, location, site, place, section | 1 record for every zone monitored by FREDs |
| Service | A type of maintenance provided to the zone, like trimming trees, moving grass etc. | Go over, check, maintain, repair, mend, fix | 1 record for each service provided by a fred |
| Data | Is a entry of details of each service performed by a fred and the details about each zone. | Information, details, entries, figures | 1 record for each entry uploaded by a FRED. |
| FRED | A Flying robot or Drone, that maintains zones by using services. | Robot, drone, servicer, FREDDY | 1 record for each FRED drone. |
| Maintenance | To provide upkeep to something like a FRED, Service it like swapping parts and equipment. | Preservation, conservation, service, upkeep | 1 Record for each maintenance that’s been done on a drone |
| Maintainer | A person that fixes/performs routine maintenance on the FREDs. | Upholder, sustainer, worker, employee | 1 record for each maintainer hired by the maintainer company |
| Maintenance Schedule | A schedule that is given to a maintainer to let them know what FREDs need maintenance. | Schedule, agenda, calendar, appointment | 1 record for each FRED. |
| Maintainer Company | A third party company that hires maintainers to maintain the FREDs. | Company, third party, maintainers, entity | 1 record for each company hiring maintainers that maintain the FREDs and have access to the schedules. |
| Part | A component of the FREDs that can be interchanged when maintenance happens and is also supplied by parts suppliers. | Piece, segment, component | 1 record for each part supplied by the supplier/s. |
| Parts Supplier | Is a third-party company that provides parts for the FREDs. | Company, supplier, entity | 1 record for each company that helps supply parts. |

## Relationships:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Cardinality** | **Participation** | **Relationship** | **Participation** | **Cardinality** | **Entity Name** |
| Director | N | Partial | Hire | Full | M | Executive Administrator |
| Executive Administrator | 0..\* | Partial | Hire | Full | 0..\* | Employee |
| Person | 1 | Full | D | Partial | U | Director |
| Person | 1 | Full | D | Partial | U | Employee |
| Employee | 1 | Full | D | Partial | U | Executive Administrator |
| Data | N | Partial | Views | Full | 1 | Platinum |
| Data | N | Partial | Owns | Full | 1 | Super Platinum |
| Platinum | M | Full | Negotiates | Partial | N | Executive Administrator |
| Super Platinum | M | Full | Negotiates | Partial | N | Executive Administrator |
| Employee | U | Partial | D | Full | U | Salesman |
| Employee | U | Partial | D | Full | U | Engineer |
| Salesman | 1 | Partial | Markets Products | Full | 0..\* | Customer |
| Executive Administrator | 1 | Partial | Markets Products | Full | 0..\* | Customer |
| Customer | 1 | Partial | Buys | Full | 1 | Subscription |
| Subscription | U | Partial | D | Full | U | Standard |
| Subscription | U | Partial | D | Full | U | Gold |
| Subscription | U | Partial | D | Full | U | Platinum |
| Subscription | U | Partial | D | Full | U | Super Platinum |
| Standard | 1 | Full | Creates | Full | 1..\* | Contract |
| Gold | 1 | Full | Creates | Full | 1..\* | Contract |
| Platinum | 1 | Full | Creates | Full | 1..\* | Contract |
| Super Platinum | 1 | Full | Creates | Full | 1..\* | Contract |
| Contract | 1 | Full | Generates | Full | 1 | Invoice |
| Invoice | 1..\* | Full | Assigned | Partial | 1 | Customer |
| Contract | 1 | Full | Assigned to | Partial | N | Region |
| Region | 1 | Partial | Contains | Full |  | Zone |
| Zone | 0..\* | Partial | Based on | Partial | 1 | Data |
| FRED | N | Partial | Uploads | Full | 1 | Data |
| FRED | M | Partial | Selects | Full | N | Service |
| Service | 0..\* | Full | Given to | Partial | 0..\* | Zone |
| Fred | N | Full | Designs/Builds | Partial | M | Engineer |
| Fred | M | Partial | Maintains | Full | N | Maintenance |
| Fred | M | Full | Sends location | Full | N | Maintenance Schedule |
| Maintenance | N | Full | Creates | Full | M | Maintenance Schedule |
| Maintenance Schedule | N | Full | Has access to | Partial | 1..\* | Maintainer |
| Maintainer | N | Full | Employs | Partial | 1 | Maintainer Company |
| Maintenance | 1..\* | Partial | Uses | Partial | 0..\* | Part |
| Part | N | Full | Supplies | Partial | M | Part Supplier |

## Attributes:

### Director:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **Director** | DirectorID | Identifier for director | INT | ID | no | no | no | Primary |  |
|  | Name | Name of director | VARCHAR(255) | Username | no | no | no |  |  |
|  | Email | Email of director | VARCHAR(255) | Mail | no | no | no |  |  |
|  | Phone | Phone of director | VARCHAR(16) | PH, cell number | no | no | no |  |  |
|  | country | Country of director | VARCHAR(255) |  | no | no | no |  |  |
|  | Region | Region of director | VARCHAR(255) | area | no | no | no |  |  |
|  | City |  | VARCHAR(255) | Large town | no | no | no |  |  |
|  | Postal\_No |  | INT(4) | Post Numbe, Post Address | no | no | no |  |  |
|  | Street |  | VARCHAR(255) |  | no | no | no |  |  |
|  | Street Number |  | VARCHAR(10) |  | no | no | no |  |  |
|  | House Number |  | VARCHAR(10) | Apartment Number | no | no | yes |  |  |

### Executive Administrator

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **Executive Administrator** | ExecAdminID | Identifier for Executive Administrator | INT | ID | no | no | no | Primary |  |

### Person:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **Person** | PersonID | Identifier for Person | INT | ID | no | no | no | Primary |  |
|  | IsDirector | Determines who is able to hire exec | Bool |  | No | No | No |  |  |
|  | IsExec | Who is able to hire employees | Bool |  | No | No | no |  |  |

### Employee:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **Employee** | EmployeeID | Identifier for Employee | INT | ID | no | no | no | Primary |  |
|  | Name | Name of Employee | VARCHAR(255) | Username | no | no | no |  |  |
|  | Email | Email of Employee | VARCHAR(255) | Mail | no | no | no |  |  |
|  | Phone | Phone of Employee | VARCHAR(16) | PH, cell number | no | no | no |  |  |
|  | country | Country of Employee | VARCHAR(255) |  | no | no | no |  |  |
|  | Region | Region of Employee | VARCHAR(255) | area | no | no | no |  |  |
|  | City | City of Employee | VARCHAR(255) | Large town | no | no | no |  |  |
|  | Postal\_No | Post No. of Employee | INT(4) | Post Numbe, Post Address | no | no | no |  |  |
|  | Street | Street of Employee | VARCHAR(255) |  | no | no | no |  |  |
|  | Street Number | Street No. of Employee | VARCHAR(10) |  | no | no | no |  |  |
|  | House Number | House No. of Employee | VARCHAR(10) | Apartment Number | no | no | yes |  |  |
|  | Salary | Amount paid per year | INT |  | No | No | no |  |  |

### Salesman:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Salesman** | DiscountPercent | Name of Employee | INT |  | no | no | no |  |  |

### Engineer:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Engineer** | Degree | Proof of skill | INT | certificate | no | no | yes |  |  |

### Subscription:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Subscription** | SubscriptionID | Identifier for subscription | INT | id | no | no | no | Primary |  |
|  | Prices | Price of subscription | INT | Price, cost | No | No | No |  |  |

### Customer

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **Customer** | CustID | Identifier for customer | INT | ID | no | no | no | Primary |  |
|  | Name | Name of customer | VARCHAR(255) |  | No | No | No |  |  |
|  | Phone Number | Cell number | VARCHAR(16) | Cell, line | No | No | no |  |  |
|  | Account Name | Name of entity | VARCHAR(255) | Trading name | No | No | yes |  |  |
|  | Email | Email of entity | VARCHAR(255) | Mail | No | No | yes |  |  |

### Contract

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **Contract** | ContractID | Identifier for Contract | INT | ID | no | no | no | Primary |  |
|  | StartDate | date for start | DATETIME |  | No | No | no |  |  |
|  | EndDate | Date for end | DATETIME |  | No | No | no |  |  |
|  | ExclusiveFRED | Exclusive FREDs |  |  | No | no | yes |  |  |

### Invoice:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **Invoice** | InvoiceID | Identifier for invoice | INT | ID | no | no | no | Primary |  |
|  | TotalAmount | Total cost | INT | totalprice | Yes add +15% gst | no | no |  |  |
|  | TotalAmountNoGST | No gst included | INT | No gst cost | No | No | No |  |  |
|  | IssuedDate | Date issued | DATETIME | Date | No | No | No |  |  |
|  | DueDate | Date due of payment | DATETIME | date | No | No | no |  |  |

### Region:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **Region** | RegionID | Identifier for Region | INT | ID | no | no | no | Primary |  |
|  | Size | Size of region in Ha | INT | Ha | No | No | No |  |  |
|  | Country | Name of county | VARCHAR(255) |  | No | No | No |  |  |
|  | regionName | Region name | VARCHAR(255) |  | No | No | No |  |  |
|  | isMaintained | Is area activelt got drones | Boolean |  | No | No | No |  |  |

### Zone:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **Zone** | ZoneID | Identifier for Zone | INT | ID | no | no | no | Primary |  |
|  | IsMaintained | Is area drone active | Boolean |  | No | No | No |  |  |

### Data:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **Data** | DataID | Identifier for Data | INT | ID | no | no | no | Primary |  |
|  | Latitude | location | VARCHAR(255) |  | No | No | No |  |  |
|  | Longitude | location | VARCHAR(255) |  | No | No | no |  |  |
|  | Temperature | How hot it is | VARCHAR(255) |  | No | No | no |  |  |
|  | Humidity | How moist it is | VARCHAR(255) |  | No | No | no |  |  |
|  | Ambient light strength | How much light is there | VARCHAR(255) |  | No | No | no |  |  |
|  | Organic spectral data from one litre of air | Organic air health data | VARCHAR(255) |  | No | No | no |  |  |
|  | IsViewable | For prem subs | BOOL |  | No | No | No |  |  |
|  | IsOwned | For super prem sub | BOOL |  | No | No | no |  |  |

### FRED:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **FRED** | FREDID | Identifier for FRED | INT | ID | no | no | no | Primary |  |
|  | LastMaintained |  | DATETIME |  |  |  |  |  |  |
|  | IsActive | Is working | Bool |  | No | No | No |  |  |
|  | MadeDate | Date fred is made | DATETIME |  | No | No | no |  |  |
|  | DataUpload | Last data upload time | DATETIME |  | No | No | no |  |  |
|  | FREDConfig | What job its meant to do | VARCHAR(255) |  | No | No | No |  |  |

### Service:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **Service** | ServiceID | Identifier for Service | INT | ID | no | no | no | Primary |  |
|  | serviceName | Name of service | VARCHAR(255) |  | No | No | No |  |  |
|  | DateStart | Date started | DATETIME |  | No | No | No |  |  |
|  | DateEnd | Date ended | DATETIME |  | No | No | No |  |  |
|  | FREDConfig | Config meets service requirement | VARCHAR(255) |  | No | No | No | Foreign key fred table |  |

### Maintainer Company:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **Maintainer Company** | MaintainerCompanyID | Identifier for MaintainerCompany | INT | ID | no | no | no | Primary |  |
|  | CompanyName | Identifies company name | VARCHAR(255) | name | No | No | No |  |  |
|  | ContactNo | Phone number if needed | VARCHAR(16) | Phone, cell, call | No | No | No |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **Maintainer** | MaintainerID | Identifier for Maintainer | INT | ID | no | no | no | Primary |  |
|  | Name | Name of person | VARCHAR(255) | Username | No | No | No |  |  |
|  | PhoneNumber | Phone number of person | VARCHAR(255) | Number, cell, phone | No | No | No |  |  |

### Maintenance:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **Maintenance** | MaintenanceID | Identifier for Zone | INT | ID | no | no | no | Primary |  |
|  | StartDate | Date of start | DATETIME |  | No | No | no |  |  |
|  | EndDate | End of maintenance | DATETIME |  | No | No | no |  |  |

### Maintainer:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **Maintainer** | MaintainerID | Identifier for maintainer | INT | ID | no | no | no | Primary |  |
|  | Name | Name of maintainer | VARCHAR(255) |  | No | No | No |  |  |
|  | ContactNo | Phone number | VARCHAR(16) | Cell, phone | No | No | No |  |  |

### Maintenance Schedule:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **Maintainer** | MaintenanceSheduleID | Identifier for maintenance schedule | INT | ID | no | no | no | Primary |  |
|  | DueDate | Date for due maintenance | DATETIME |  | No | No | No |  |  |
|  | description | For marking as routine/specific problems | VARCHAR(1000) | desc | No | No | No |  | Routine maintenance |

### Parts:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **Parts** | PartID | Identifier for part | INT | ID | no | no | no | Primary |  |
|  | Price | The cost of part | INT |  | No | No | No |  |  |
|  | Description | Short desc of part | VARCHAR(500) | Desc | No | No | Yes |  |  |

### Parts Supplier:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **PartSupplier** | PartSupplierID | Identifier for partsupplier | INT | ID | no | no | no | Primary |  |
|  | CompanyName | Name of company | VARCHAR(255) | Name | No | No | No |  |  |
|  | ContactNo | Phone number in case | VARCHAR(16) | Phone, cell, line | No | No | No |  |  |

### Standard:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **Standard** | MaxHect | How many hectares the sub is allowed | INT | Area, location | No | No | no |  | 10 |
|  | MaxFRED | How many freds the sub is allowed | INT | Robot, drone | No | No | No |  | 1 |
|  | MaxZone | How many zones the sub is allowed | INT | Region area, location | No | No | No |  | 1 |

### Gold:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **Standard** | MaxHectares | How many hectares the sub is allowed | INT | Area, location | No | No | no |  | 100 |
|  | MaxFREDs | How many freds the sub is allowed | INT | Robot, drone | No | No | No |  | 3 |
|  | MaxZones | How many zones the sub is allowed | INT | Region area, location | No | No | No |  | 3 |
|  | MaxServices | How many allowed services | INT | Service number | No | No | No |  | 3 |

### Platinum:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **Standard** | MaxHectares | How many hectares the sub is allowed | INT | Area, location | No | No | yes |  |  |
|  | MaxFREDs | How many freds the sub is allowed | INT | Robot, drone | No | No | yes |  |  |
|  | MaxZones | How many zones the sub is allowed | INT | Region area, location | No | No | yes |  |  |
|  | MaxServices | How many allowed services | INT | Service number | No | No | yes |  |  |

### Super Platinum:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| **Standard** | MaxHectares | How many hectares the sub is allowed | INT | Area, location | No | No | yes |  |  |
|  | MaxFREDs | How many freds the sub is allowed | INT | Robot, drone | No | No | yes |  |  |
|  | MaxZones | How many zones the sub is allowed | INT | Region area, location | No | No | yes |  |  |
|  | MaxServices | How many allowed services | INT | Service number | No | No | yes |  |  |

## Rationale on data dictionary:

Primary Key Entities:

* Subscription
* Parts
* PartsSupplier
* Maintainer
* Maintenance
* MaintenanceSchedule
* MaintainerCompany
* Service
* FRED
* Zone
* Region
* Contract
* Person
* Employee
* Data
* Invoice
* Customer

Weak Entities:

* Invoice
* Contract
* Service
* Zone
* Data
* Part
* MaintenanceSchedule

Listed above are all my entities, They are all there as I have been making the conceptual model I have been filling in the gaps and rearranging things, like how Person wasn’t an entity till later on in the model creation process.  
  
The relationships are all mentioned in the analysis of the diagram itself so there Is no need for repetition.

Attributes:

Director:

For the director I gave him all basic attribute to identify them, Phone, address, name, ID all to keep track of each director and their details.  
  
Executive administrator:

The executive administrator has ID in it which ins’t much but that’s because the only difference is that they can hire employees, negotiate higher level contracts and change subscription prices.

Person:

In Person I have an Id to identify the person, and also a two type system to determine what kind of person it is.

IsDirector and IsExec are the options here because they determine how I know whether a person is an employee that isn’t a executive or a director or neither.

Employee:

Same as above with director exactly as I have an ID, name, email, phone, and an address, all are required as various ways are required to contact employees and personell.

Salesman:  
These employees get the discountpercent attribute determining the percentages that they are allowed to give, this attribute exists to show whether a salesman has the authorization to give a discount or whether they could only be allowed to give a lower percent than other salesman.

Engineer:

Engineer just has degree as it will show what area of expertise the engineer is at building freds, and in future we could create more entities that would help partner engineers to work together to create drones better and more efficiently.

Subscription:

This only has an ID and prices, as most of the fine details about the subscription are in what type you choose and even more refined details can be found after a contract has been made.

Customer:

This one has ID as we need to know what customer is linked to what contract/subscription/salesman, we also need to know the contact details like the company name(accountname), and the name of customer(main contact we deal with), phone number and their email.  
It is vital to have all of your customers contact info in cases of emergency and if contact is needed for queries.

Contract:

For this I have StartDate, EndDate and ID. ID is so that we can identify what ID will be linked to what customer and what subscription is hooked up to set the limitations for the freds. Start and end date are for the contract duration for the freds.

Invoice:

This has its own ID as many customer can have many contracts and Invoices so it may get difficult to keep track without an indentifier. In the brief it states that we should have total price with and excluding GST, we also have a Due Date for when the payment is required to be received by. We also keep track of issue date just in case.

Region:

This one has an ID, and the size, this is measured in hectares. The region name and isMaintained must also be listed to show the name of the place, what country its in, and whether we currently are performing services in the region.

Zone:

Zone has an ID, and also has a IsMaintained as some zones may not be active, zones may overlap each other from the same regions.

Data:

Data contains a lot of attributes, all required by the fred to upload in records. To start each piece of data uploaded must have its own identifier (id), and then we need to know the location of the datas origin so we need its latitude and longitude, after this we will then go into the data the fred has collected like Temperature, Humidity, Ambient Light Strength, organic spectral data from 1L of air and then finally we have the views and owns attribute which are limited to the platinum and super platinum subscribers.

FRED:

Each fred has its own id, Each FRED is required to get regular maintenance so I have given it an attribute LastMaintained which is the datetime of its last maintenance. I also have a made to represent when the fred was made for first maintenanceschedule dates. Each FRED will complete a service and then upload the data from the previous entity into it, it will also upload its current location to the scheduleMaintenance class so that maintainers can find it. The last attribute there is called FREDConfig, which basically is whats the FREDs specific service it can do is based on what equipment it has, or based on the size of the drone.

Service:

In service I have included an ID, Startdate, enddate and the FREDConfig of the drone so that we ensure that the drone trying to do the task can do it. The dates are the timeframe of the service to start and end.

MaintainerCompany:

In this entity I have assigned the attributes, id, Name, and Contact number, contact number in case we need to report or get a new maintainer, name for generic storage means of their data, and id to keep track of different maintainer companies.

Maintainer:

In this entity I have ID, name and contact number so that I can contact them if they need to do anything and name for matching a name to a phone number.

MaintenanceSchedule:

In this Entity I have given these attributes ID, DueDate and Description. Description is to change if its not a standard checkup and a fred needs more than the usual work, duedate is when the maintainer should get to the fred, and ID to keep track of maintenance ever done just in case we need to hold records.

Part:

For this I have an id for all parts supplied, a price for each part, and a quick description of what the part actually is and its purpose.

PartSupplier:

For this I have got an ID to track all companies supplying parts to us, and also keeping their names and contact numbers on record if we need to order more parts from them later on.

Standard:

The standard subscription has Max Hectares of 10 default, Max Zone of 1 default and a Max FRED of 1. This is the default provided to us by the company brief.

Gold/Platinum/Super Platinum:

All of these subscription have similar attributes like Max Hectares, Max FRED, Max Services, Max Zones for gold these are 3 fred, 3 services, 3 zones and 100Hectares, but for the other 2 subscription the numbers are all upto negotiation of the contract with the customers assigned executive administrator.  
  
The Data dictionary is a support line that reinforces the conceptual model by dictating all of the data into entities, their relationships and their attributes. This allows for less redundancy of work in future as less time will be spent working on documenting attributes and their types.

Another reason why data dictionaries help the model is because it can help other project managers and developers understand certain entities and attributes by reading the dictionary and seeing the reasons behind certain relationships and datatypes. In this data dictionary we were told that we didn’t have to add foreign keys, but we did add primary keys, nullable options, and added Boolean options for certain attributes.

Basically the Data Dictionary helps with data management, consistency and data integrity by plotting out the possible data to prevent errors and showing clarity to the setup of the system in its entirety.

# Assumed Business rules with reasoning:

That Directors can’t hire employees, because it only says that they can hire executive administrators.  
  
Executive administrators can only hire employees that aren’t executive administrators. This is because it wasn’t mentioned that they could in the brief

I assume that when a subscription is purchased that a contract is made reflecting that subscription, and more contracts of the same subscription can be added later on. This is however unless they want to upgrade their subscription to get better features like more drones/zones.

I assume that Not all maintenance requires the use of parts to complete the checkups/maintenance.

I assume that the maintainer is hired by a third party maintenance company.

I assume that a customer can also be classified as a company.  
  
I assume that one customer can have 1 subscription but multiple contracts.

I assume that all contracts have assigned zones in them, and all of the zones are in regions

I assume that an invoice is created timely depending on the agreement made on the contract which can either be monthly or annually.

I assume that Maintenance wont be done without a schedule and that a schedule can’t be done without a drone having its first problem.

I assume that there will be some part suppliers that wont have supplied parts as they may not have been ordered yet, or we have just added them to the system so far.

I assume that FREDs perform maintenance in a zone based of off the contract.

I assume that freds can work in multiple zones and aren’t limited to one contract, this is because they might have finished that drones task in a zone and can move onto another seamlessly.

I assume that all FREDs have one configuration setting at a time, meaning that they can either be trimming trees OR moving the grass etc.

I assume that everytime a fred completes a service it collects and uploads its location and data.

I assume that a salesman and executive Administrator can have many customers this is so they aren’t limited to only having one customer which would make no sense.

I assume that everytime maintenance is done, a schedule is made for a fred and the most recent location upload will help the maintainer find the drone to maintain it.

I assume that Maintainers can change the configurations of the FREDs, this is so if they have a shortage of some tasks in many regions that can switch some drones that are inactive and make them more useful.

# REFERENCES

*7 Best Practices for Successful Data Management | Tableau*. (n.d.). Retrieved 6 April 2025, from https://www.tableau.com/learn/articles/data-management-best-practices

*Multivalued Attribute—An overview | ScienceDirect Topics*. (n.d.). Retrieved 8 April 2025, from https://www.sciencedirect.com/topics/computer-science/multivalued-attribute

Team, J. E. (n.d.). *6 key data management principles | The Jotform Blog*. Jotform. Retrieved 6 April 2025, from https://www.jotform.com/blog/data-management-principles/

Weak entity. (2024). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Weak\_entity&oldid=1221372225