DAT601 ASSESSMENT 1

Conceptual modelling and TSQL prototyping

Caleb Eason

Contents

[1. Introduction to data modelling in Information Systems 2](#_Toc166235301)

[1.1. Overview 2](#_Toc166235302)

[1.2. Principles of data management 2](#_Toc166235303)

[1.2.1. Data quality management 2](#_Toc166235304)

[1.2.2. Data Governance 2](#_Toc166235305)

[1.2.3. Privacy 3](#_Toc166235306)

[1.2.4. Security 3](#_Toc166235307)

[1.2.5. Lifecycle management 3](#_Toc166235308)

[1.2.6. Metadata management 3](#_Toc166235309)

[2. Conceptual Modelling with Enhanced Chen ERD 4](#_Toc166235310)

[2.1. Elements of the conceptual data model. 5](#_Toc166235311)

[2.1.1. Entity 5](#_Toc166235312)

[2.1.2. Attribute 5](#_Toc166235313)

[2.1.3. Relationship 6](#_Toc166235314)

[2.1.4. Weak entities 7](#_Toc166235315)

[2.1.5. Specialisations 8](#_Toc166235316)

[3. Conceptual ER Model 9](#_Toc166235317)

[3.1. Rationale 10](#_Toc166235318)

[3.2. Application of Data Management Principles 11](#_Toc166235319)

[4. Data Dictionary 12](#_Toc166235320)

[4.1. Entities 12](#_Toc166235321)

[4.2. Relationships 15](#_Toc166235322)

[4.3. Attributes 16](#_Toc166235323)

[4.4. Rationale 20](#_Toc166235324)

[4.5. Review of Data Management Procedures 20](#_Toc166235325)

[5. Assumptions on Business Rules 21](#_Toc166235326)

[6. References 22](#_Toc166235327)

# Introduction to data modelling in Information Systems

## Overview

In information systems, data modelling is the process of creating a visual representation of data flow and storage within a system. Data models show how data points and structures are connected to each other through entity relationships, how data is stored and formatted, and how data is organised within a system (IBM, retrieved 2024).

There are three distinct phases of designing a database, conceptual modelling, logical modelling, and physical implementation.

The conceptual model is used for discovering the data requirements of system and defining how data entities relate to one another. In a conceptual model, data entities have attached attributes that define their properties and are linked together with relationship objects. For each entity, at least one candidate key is chosen as a possible identifier.

The logical model restructures the conceptual model to align with a relational database structure. In the logical model, entities are converted to relations which contain their attributes within themselves. Datatypes and sizes are defined for each attribute and a primary key is chosen from the available candidate keys. Relationship objects are removed and replaced with foreign keys in linked relations.

For the physical implementation, adjustments are made to the logical model to ensure it matches the requirements of the chosen database system. The physical implementation will vary depending on what database system is used. For example, one database system might use strings to store text, while another may use varchars. Physical implementation is sometimes accompanies by a physical data model, a redrawing of the logical model that matches the characteristics of the database system.

## Principles of data management

There are several fundamental principles that should be followed to ensure effective data management within an organisation.

### Data quality management

Data quality refers to the accuracy, completeness, consistency, validity, and timeliness of data. High quality data should correctly serve the outcome it is intended for (Sagacity Solutions, 2024). I.e. User account data should reflect a user’s details, ensuring they are accurately represented, and can properly utilise a system. An absence of data quality management will result in erroneous data entering a system. This can cause issues for users and businesses, who rely on data to make decisions and perform tasks. Low quality data can impair the functionality of a system which may require data in specific formats. To ensure data entering a system is of high quality, data management practices must be implemented. All incoming data should be profiled and controlled to prevent intake of erroneous data (Jotform, 2020).

### Data Governance

Data governance is the process of managing data to comply with internal and external policies and regulations, and to ensure effective use of data. Effective data governance aims to maintain data availability, integrity, security, and usability, to ensure data is accurate, consistent, and legal. In large organizations this is generally handled by a data governance team. This team will define and enforce polices and strategies for data governance within an organization (Stedman, 2022).

### Privacy

Data privacy refers to how an organisation handles sensitive personal and/or business information. To ensure compliance with legal requirements and to maintain a positive reputation, companies must utilise proper strategies for collection, transfer, storage, sharing, and use of sensitive data. Sensitive data includes any personal identifiable information from customers or employees such as contact details, health records or financial data, and any business information such as research or financial information (Gattis, 2024). Failure to implement proper data privacy policies could cause harm to a company and its customers and employees, which may result in legal repercussions. Companies must familiarize themselves with the legal requirements of the countries they are operating in and providing services to, to ensure they are in compliance.

### Security

Data security is the process of controlling and restricting access to information to protect it from unauthorised access, disclosure, tampering, or destruction. Data security can be enforced by implementing authentication and authorisation procedures to limit who has access to certain pieces of data. Authentication prevents any internet user who managers to find a database from accessing its contents. Authorisation ensures any individual who is allowed to access a database can only view the data they have been specifically authorised to access. For example, an employee may be able to view a customer’s name and email but may be restricted from viewing their payment information. Further security can be provided with encryption, masking, and anonymisation of data. Masking and anonymization scrubs data of any personal identifiable information, enabling data analysts to view data without compromising the privacy of the individuals it pertains to. Encryption of data ensures that even if a data breach occurs, the data cannot be interpreted without a key to decrypt the data (Fortinet, 2024).

### Lifecycle management

Data lifecycle management refers to how an organisation manages data in an information system throughout its lifecycle. The data lifecycle encompasses, creation, initial storage, use, long term storage, and eventual deletion of data. Companies will need to enact data lifecycle management policies to ensure data storage is inline with the key principles of data management throughout the data lifecycle (Sheldon, 2022).

### Metadata management

Metadata is data about data. Data Metadata provides information about data schemas, structures, content, context, access, ownership, data types, dependencies, models, and relationships. Effective management of metadata can significantly improve the useability of a data for an organisation. Good metadata enables better understanding of data, improving its consistency and availability, and supporting data governance and analysis. To ensure metadata is effective, business need to implement policies for capturing and storing metadata. Meta data must be actively maintained to ensure it is up to date and must be disseminated in a way that makes it easily accessible to those with access (Ranjan, 2024).

# Conceptual Modelling with Enhanced Chen ERD

A conceptual data model is a structured diagram that depicts the required data and data interactions to support business processes. The model is not focused on physical characteristics of data or processing flow, rather it focuses solely on identifying data requirements, and defining the overall structure of data. (Sherman, 2015). Conceptual models are built using specific components that represent different elements of a data structure. These elements are linked with relationships which define how they interact. The extended Chen notation expands on the conceptual model by enabling data entries (see below) to have specialisations (Ranjan, 2024).

The structured nature of the conceptual model helps enforce data quality. Defining exactly what attributes an entity has ensures data is complete and consistent. Carefully defining exactly what an entity is, how it relates with other entities, and what data it needs, helps to ensure data can be used to correctly serve its intended purpose. Creating a conceptual model requires a designer to consider strategies for data governance. The database architect will need to consider exactly how the entities and their interactions will align with internal and external polices when designing the structure of a conceptual model. The conceptual model will also highlight what types of information a system needs to store. This will bring attention to any sensitive information that will be stored in the database, requiring the designer to consider privacy policies for the database. While data security is not directly addressed in the conceptual model, the model provides a helpful view for visualising and determining what data access procedures need to be implemented. Relationships between entities can reveal some information on required access roles by showing which entities need to communicate with each other, and which don’t need anything to do with one another. Specialization of entities from the enhanced Chen ERD notation can also show what different access roles can be implemented into a database. Specialization of an entity that have different relationships will show the differing access roles that can be applied to an entity. The conceptual model promotes lifecycle management by directly addressing how data is stored, and how data can be used. The model also shows how the deletion of data from one entity will affect data in other entities though relationships. The conceptual model also reveals what metadata will be required for system structures and relationships. When designing a conceptual model, the database architect will need to consider candidate keys for identifying entities. Cardinality, optionality, and participation in relationships. And the content and context of data within entities. Specialization and generalisations from the enhanced Chen ERD notation can also provide additional information on data structure metadata.

## 2.1. Elements of the conceptual data model.

### Entity

A blue rectangle with white text

Description automatically generated

An entity represents a specific object within a system. An entity can be anything within a system that has data of some type. For FlightStream, some example entities include customers, DataScoops, and contracts. An entity is depicted as a rectangle with its name in the centre. An entity can have several attributes that describe its properties and can be connected to other entities through relationships.

### Attribute

A blue and white diagram

Description automatically generated

Attributes are attached to entities. They are the properties that describe an entity and can represent any piece of data that is attached to an entity. For example, a customer entity might have the attributes name, email, and phone number. An attribute is depicted as an oval attached to an entity with a straight line.

An entity can have a number of special types of attributes.

A blue oval sign with white text

Description automatically generated

A primary key attribute represents a unique identifier for an entity. Each instance of an entity must have a unique primary key attribute. A primary key attribute is indicated by underlining its name.

A multivalued attribute comprises multiple pieces of data as a single attribute. For example, latitude, longitude, and altitude could be stored as a single multivalued attribute called location. A multivalued attribute is represented with an inset outline.

A derived attribute is an attribute that is calculated from existing data. For example, a customer might have an attribute, age, which is calculated from another attribute, date of birth. There is no special notation for a derived attribute.

### Relationship

A blue diamond with white text

Description automatically generated

A relationship represents an connection between two entities, describing how they interact. For example, a customer entity might *view* a contract entity.

A blue diamond with white text

Description automatically generated

A relationship is described with a diamond and is attached to entities with a line.

Relationships between entities have different parameters that describe the relationship.

Participation dictates whether an entity is required to partake in a relationship. Participation can be either mandatory, or optional. Mandatory participation means an entity instance must be part of the relationship. For example, an employee of a company must be associated with a specific company branch. Optional participation means an entity is not required to partake in a relationship. For example, a company branch can exist without any employees. Mandatory participation is denoted with a double line, optional participation is denoted with a single line.

A blue diamond with white text

Description automatically generated

Cardinality dictates how many instances of an entity can partake in a relationship. Cardinality can be either one or many. For example, a branch of a company can have many employees, but an employee can only work for one branch. A cardinality of one is represented by a number 1 placed above the connecting line for the appropriate entity. A cardinality of many is represented by a letter M placed above the connecting line for the appropriate entity. In a relationship where both entities have a cardinality of many, one of the M symbols is replaced by an N to distinguish each side of the relationship.

A blue diamond with white text

Description automatically generated

A blue diamond with white text

Description automatically generated

### Weak entities

A blue sign with white text

Description automatically generated

A weak entity is an entity is an entity that is identified through a relationship with another entity. Weak entities can not exist on their own and do not have a primary key. For example, an employee may have dependents that can not exist without the employee. A weak entity is depicted with a inset outline. The relationship that defines the weak entity is called an identifying relationship and is depicted with an inset outline.

A blue diamond with white text

Description automatically generated

### Specialisations

A diagram of a company

Description automatically generated

Enhanced Chen notation introduces the idea of specialisations. A specialisation is a type of entity. That builds of a base entity called a generalisation. Specialisation entities inherit all the attributes and relationships of the generalisation and can augments it with additional attributes and relationships. Specialisations allow an entity to have different types without needing to define an entirely new entity. For example, a company may have several different types of employees, such as accountants, customer service representatives, and engineers. Each of these employee types will share some common attributes such as name and email, while also having some unique attributes. For this case, an employee entity can be created with the shared attributes, then specialisations can be created to contain the unique attributes for each type of employee.

If an instance of an entity can have multiple specialisations, the specialisations are described as overlapping. If an entity instance can only have a single specialisation, the specialisations are described as disjoint. Overlapping specialisations are depicted with a letter O placed in the connecting circle. Disjoint specialisations are depicted with a letter D placed in the connecting circle.

A blue circle with white text

Description automatically generated A blue circle with white letters

Description automatically generated

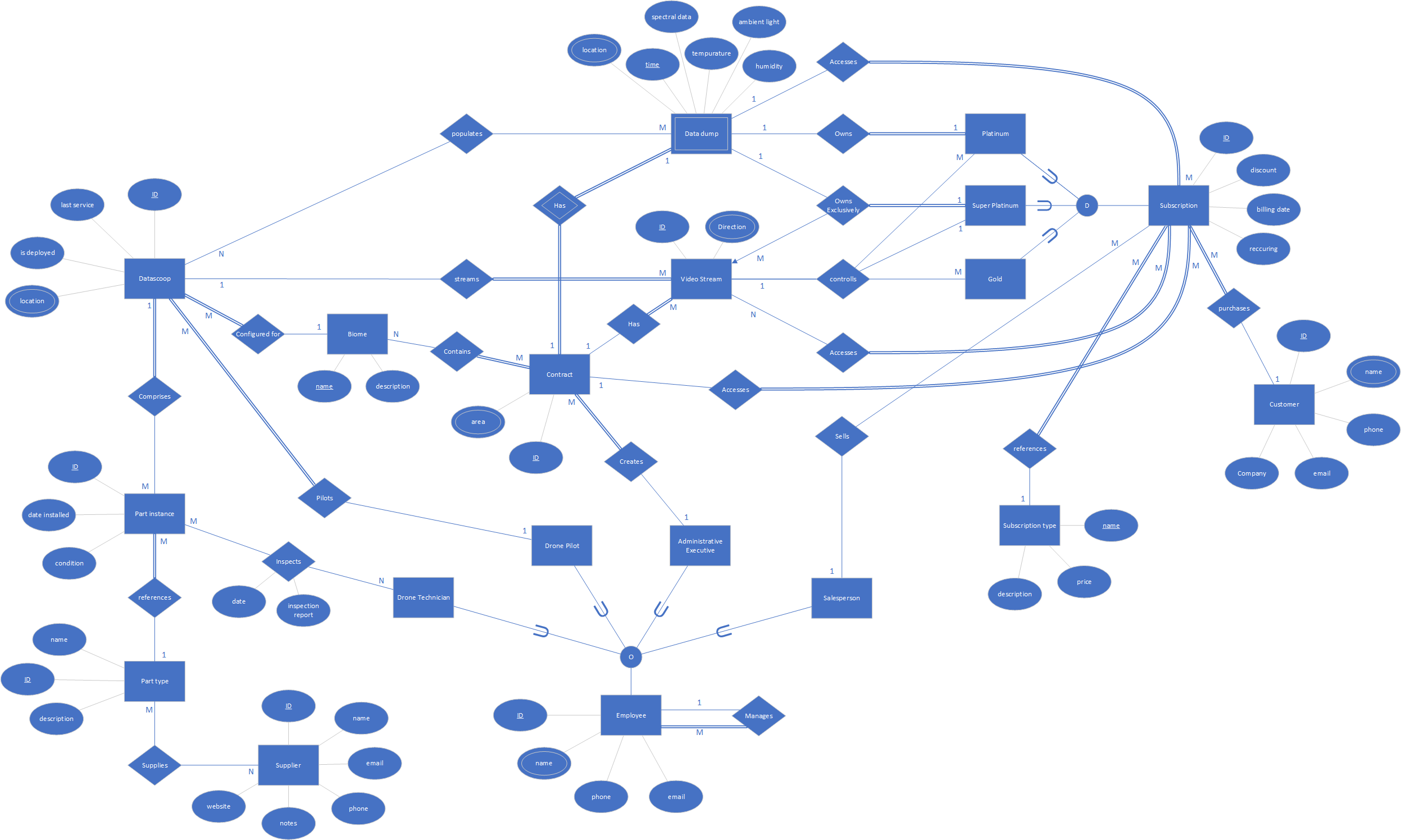
Specialisations can be mandatory or optional. This is depicted with the same notation as relationship participation. If an entity must have a specialsation, a double line is used for the connection between the generalisation and the specialsation circle. If an entity is not required to have a specialisation, a single line is used.

A blue sign with white text

Description automatically generated A blue sign with white text

Description automatically generated

# Conceptual ER Model



## Rationale

A diagram of a diagram

Description automatically generatedThe section of the diagram that displays the relationships between subscription specialisation, and the data dumps and video streams has some relationships that include more than two entities. It is important to note that this does NOT represent a ternary or quaternary relationship. This is a simplification of the actual relationship structure intended to improve the readability of the diagram. The actual relationships are as follows:

* Platinum Owns Data Dump
* Platinum Controls Video Stream
* Super Platinum Owns Exclusively Data Dump
* Super Platinum Owns Exclusively Video Stream
* Super Platinum Controls Video Stream
* Gold Controls Video Stream

This simplification of the diagram was made to greatly improve the readability of the diagram. Individually depicting each relationship in the diagram creates structure that is difficult to follow visually, as a result this compromise was made.

The customer entity is not directly connected to the contract, data damp, and video stream entities. This may appear as a chasm trap, but this is not the case. At first glance, it may seem that the customer should have a relationship with these entities. However further examination will reveal this is not the case. This is because accessing this information will be handled by a system external to the database, such as an online website for watching video streams. This system will determine access, by seeing if a customer has a relationship with a subscription that allows them access to the information from the contract, video stream and data dump entities. Within the database, there is no information in these entities that is dependent on a customer entity, so the relationship is not required.

The subscription type table is a reference table that holds information about a type of subscription. This table is referenced by the subscription table which represents a specific instantiation of a subscription. This means an administrative executive can easily update information about a subscription type without needing to change attributes in every single instance of a subscription.

The data dump is a weak entity identified by its relationship to the contract entity. This is because each contract entity has unique data dump linked to it. A data dump can not exist without being attached to a contract, so it is a weak entity. In contrast, the video stream is not a weak entity. A video stream is produced by a DataScoop, which can have up to 100 individual video streams. Gold subscribers have the ability to control one video stream, so it is necessary to have a method of identifying individual video streams. As a result, they can not be a weak entity.

The employee entity has a unary relationship with itself. This is to enable employees to be managed by other employees. The employee also has several specialisations which have different relationships with other entities. These specialisations are overlapping and are not mandatory. This makes the database flexible by allowing an employee to exist without having a specialised role. It also allows employees to take on multiple roles, so a drone pilot could also act as a drone technician as an example.

The biome entity acts as a lookup table. This ensures biomes remain consistent within DataScoops and contracts.

The part type table is a reference table that holds information about a type of drone part. This table is referenced by the part instance table which represents a specific instantiation of a part. This means information on a part can be stored in a single place and does not need to be duplicated for every instance of a part, ensuring data integrity. Part instances can partake in a relationship with the DataScoops entity. This provides an easy way to show what parts are within a DataScoop instance. This relationship is not mandatory for the part instance entity, allow parts to exist without being part of a DataScoop.

## Application of Data Management Principles

This model defines a specific structure for a data model. Every entity has all the required attributes defined; each required relationship is shown. This will support data quality, ensuing data is complete and consistent and can be used to fulfil its purpose. This model is designed with a data governance approach in mind to facilitate the company policies of FlightStream as defined in the project brief. This model implements the aspects of FlightStream’s muti-tiered subscription model into the database design through relationships and specialisations of with/of the subscription entity. This model also shows the sensitive information stored in the database, revealing data privacy needs. The database will store personal identifiable information of both FlightStream customers and employees, along with vast amounts of data collected from DataScoops that can belong to a customer in some cases (Super Platinum contracts own the rights to data collected for a contract). This conceptual model also shows which entities will need to communicate with each other -administrative executives need to access contract entities- and which entities shouldn’t have anything to do with one another -salespersons don’t need to know anything about drone parts-. This conceptual model provides a basis to develop lifecycle management strategies by showing how data will be stored, and how data from different entities can interact, informing how it can be used. The relationships in this model show how deleting data from entities will affect other entities, informing on requirements of lifecycle management. This conceptual model also reveals the metadata for system structures and relationships, and defines candidate keys. This will provide the information needed to develop comprehensive metadata management strategies in the future.

# Data Dictionary

## Entities

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity Name** | **Description** | **Aliases** | **Occurrence** |
| DataScoop | A DataScoop is a drone that is used to collect data from geographic areas and stream video. DataScoops are controlled remotely by a FlightStream employee. DataScoops can configured different types of geographical regions (biomes). | Drone | A DataScoop is purchased by FlightStream or assembled from parts provided by external suppliers. |
| Part instance | A mechanical component of a DataScoop. DataScoop drones are comprised of mechanical parts that can be replaced or modified. A part instance represents a physical part that is either installed in a DataScoop or possessed by FlightStream. | Drone part, part, drone component. | Drone parts are purchased from external suppliers. |
| Part type | A lookup table for all the different types are parts used by DataScoops. Does not represent a specific part instance. | Part list, parts lookup, component list. | A part type is created for each different type of drone component used in FlightStream’s DataScoops |
| Supplier | A third party that manufactures or supplies drone parts used by FlightStream. | Part supplier | Any suppliers that currently or have previously supplied drone parts are recorded by FlightStream. |
| Employee | A person employed by FlightStream. Employees can have specialised roles. |  | FlightStream employs individuals to work for the company. |
| Drone Technician | A specialisation of Employee. Drone technicians are responsible for inspecting and replacing drone parts, as well as changing drone biome configurations. | DataScoop Technician | FlightStream employs individuals to maintain and configure DataScoops. |
| Drone Pilot | A specialisation of employee. Drone pilots control DataScoops, piloting them through a mobile application. One pilot can control multiple drones. | Pilot, DataScoop Pilot. | FlightStream employs individuals to fly their DataScoops |
| Salesperson | A specialisation of employee. Salespersons attempt to sell contracts to customers or attract new customers. Salespersons can apply a 3% discount to subscriptions. |  | FlightStream employs individuals to sell subscriptions. |
| Administrative Executive | A specialisation of employee. Administrative executives can change the prices of subscriptions and enter contract details. |  | FlightStream employs individuals to manage subscriptions and contracts. |
| Customer | An individual registered with FlightStream as a consumer or potential consumer of FlightStream’s services. | Subscriber | Individuals who wish to use FlightStream’s services can register for a customer account online or through a FlightStream employee. |
| Subscription | Customers can subscribe to FlightStream contracts. These agreements are itemised as subscriptions. Subscriptions allows employees to access data and video streams from contracts. There are several types of subscriptions that provide additional benefits, modelled as specialisations. |  | Subscriptions are purchased by customers. |
| Gold | A specialisation of subscription. Gold subscriptions allow customers to control one of the streams from the DataScoop. | Gold Subscriptions | Gold subscriptions can be purchased by a customer. |
| Platinum | A specialisation of subscription. Platinum subscriptions allow customers to control a video stream from the DataScoop and grant exclusive ownership of data collected by the DataScoop. | Platinum Subscriptions | Platinum subscriptions can be purchased by a customer. |
| Super Platinum | A specialisation of subscription. Super Platinum subscriptions allow customers to control a video stream and grant exclusive rights and access to video streams and data. | Super Platinum Subscriptions | Super Platinum subscriptions are negotiated with FlightStream employees. |
| Subscription type | A lookup table that contains some information on subscriptions such as price. | Subscription lookup, Subscriptions list | Subscriptions models are designed by FlightStream. |
| Contract | A geographical area designating for DataScoop surveying. Contracts can be surveyed by one or many DataScoops. | Survey area, Contract area. | Contracts are designed by FlightStream alone or in conjunction with Customers. |
| Biome | A list of biomes DataScoops can be configured for. A biome is a type of geographical region that has specific climate conditions that effect the use of DataScoops. DataScoops need to be configured in different ways for different biomes. | Regional zone | Fight Stream has a static list of biome types that require specific DataScoop configuration. |
| Video Stream | A 3D stereographic video streamed live from a DataScoop in operation. Each data scoop has 100 individual video streams | Live video, video. | When in operation a DataScoop will create live video streams which can be viewed on the FlightStream website. |
| Data dump | A weak entity attached to a contract. The data dump contains all the data (not including video) transmitted from DataScoops in a contract area. |  | When a contract is made, a data dump is created to store data from DataScoops |

## Relationships

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Cardinality** | **Participation** | **Relationship** | **Participation** | **Cardinality** | **Entity Name** |
| Supplier | N | Optional | Supplies | Optional | M | Part type |
| Part instance | N | Mandatory | References | Optional | 1 | Part type |
| Datascoop | 1 | Mandatory | Comprises | Optional | M | Part instance |
| Drone Technician | M | Optional | Inspects (part instance)  Relationship attributes:   * “date”, the date of the inspection, date, not null, * “Inspection report”, the results of the inspection, string, not null. | Optional | N | Part instance |
| DataScoop | M | Mandatory | Is configured for | Optional | 1 | Biome |
| DataScoop | N | Optional | Populates | Optional | M | Data dump |
| DataScoop | 1 | Optional | Streams | Mandatory | M | Video stream |
| Done Pilot | 1 | Optional | Pilots | Mandatory | M | DataScoop |
| Administrative Executive | 1 | Optional | Creates | Mandatory | M | Contract |
| Salesperson | 1 | Optional | Sell | Optional | M | Subscription |
| Employee | 1 | Optional | Manages | Mandatory | M | Employee |
| Contract | M | Mandatory | Contains | Optional | N | Biomes |
| Contract | 1 | Mandatory | Has (identifying relationship) | Mandatory | 1 | Data dump |
| Contract | 1 | Optional | Has | Mandatory | M | Video stream |
| Subscription | M | Mandatory | References | Optional | 1 | Subscription type |
| Subscription | M | Mandatory | Accesses | Optional | 1 | Contract |
| Subscription | M | Mandatory | Accesses | Optional | N | Video stream |
| Subscription | M | Mandatory | Accesses | Optional | 1 | Data dump |
| Platinum | 1 | Mandatory | Owns | Optional | 1 | Data dump |
| Platinum | M | Optional | Controls | Optional | 1 | Video stream |
| Super platinum | 1 | Mandatory | Owns exclusively | Optional | 1 | Data dump |
| Super platinum | 1 | Mandatory | Owns exclusively | Optional | M | Video stream |
| Super Platinum | 1 | Optional | Controls | Optional | 1 | Video stream |
| Gold | M | Optional | Controls | Optional | 1 | Video stream |
| Customer | 1 | Optional | Purchases | Mandatory | M | subscription |

## Attributes

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attribute Name** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nullable?** | **Key?** | **Default Value** |
| Datascoop | ID | A unique identifier | integer |  |  |  | NO | YES | Autoincrementing |
| Last service | When the DataScoop was last serviced | string | Last inspected, serviced |  |  | YES | NO | null |
| Is deployed | States if the DataScoop is currently in use | boolean | In use, active |  |  | NO | NO | False |
| Location | The DataScoops current GPS coordinates and altitude. If the DataScoop is not in use, the value will be null. | [decimal, decimal, decimal] | Position | Composed of latitude, longitude, and altitude. |  | YES | NO | null |
| Part instance | ID | A unique identifier | integer |  |  |  | NO | YES | Autoincrementing |
| Date installed | The date when the part was installed in the DataScoop | date |  |  |  | NO | NO |  |
| Condition | The physical condition of the part (worn out, working, etc..) | string | State |  |  | NO | NO |  |
| Part type | ID | A unique identifier | string (e.g. CY255K) |  |  |  | NO | YES |  |
| Name | The name of the part | string |  |  |  | NO | NO |  |
| Description | A description of the part. | string |  |  |  | NO | NO |  |
| Supplier | ID | A unique identifier | integer |  |  |  | NO | YES | Autoincrementing |
| Name | The name of the company/individual | string | Manufacturer, company |  |  | NO | NO |  |
| Email | The supplier’s contact email address | string with regex mask | Contact email |  |  | YES | NO | null |
| Phone | The suppliers contact phone number | string with regex mask | Contact phone |  |  | YES | NO | null |
| Website | The supplier’s website | string | Webstore |  |  | YES | NO | null |
| Notes | Any notes on the suppler from FlightStream | string |  |  |  | YES | NO | null |
| Employee | ID | A unique identifier | integer |  |  |  | NO | YES | Autoincrementing |
| Name | The employee’s full name | string, string |  | Comprised of fist name and last name. |  | NO | NO |  |
| Phone | The employee’s work phone number | string with regex mask | Work phone |  |  | NO | NO |  |
| Email | The employee’s work email address | string with regex mask | Work email |  |  | NO | NO |  |
| Biome | Name | The name of the biome | string | Type |  |  | NO | YES |  |
| Description | Description of the requirements for operating in the biome. | String | Operational requirements |  |  | NO | NO |  |
| Contract | ID | A unique identifier | integer |  |  |  | NO | YES | Autoincrementing |
| Area | A list of GPS coordinates that defines the geographical area encompassed by the contract | [  [decimal, decimal],  [decimal, decimal],  . . .  ] |  | Comprised of a list of coordinate pairs of indeterminate length. |  | NO | NO |  |
| Data dump | Time | The exact time the data was recorded by the DataScoop | datetime |  |  |  | NO | NO | Current time |
| Location | The GPS location and altitude of the DataScoop when the data was recorded | [decimal, decimal, decimal] | Position | Comprised of latitude, longitude, and altitude |  | NO | NO |  |
| Spectral data | Organic spectral data from one litre of air | Exact datatype requirements are unclear, can store as a blob until datatype is determined | Organic spectral data | Unknown |  | YES | NO | null |
| Temperature | The ambient air temperature | decimal |  |  |  | YES | NO | null |
| Ambient light | The ambient light detected by the DataScoop | decimal | Light strength |  |  | YES | NO | null |
| Humidity | The ambient humidity | decimal |  |  |  | YES | NO | null |
| Video stream | ID | A unique identifier | integer |  |  |  | NO | YES | Autoincrementing |
| Direction | The pitch, roll, and yaw of the video feed | [decimal,  decimal,  decimal] | Angle | Comprised of pitch, roll and yaw. |  | NO | NO |  |
| Subscription | ID | A unique identifier | integer |  |  |  | NO | YES | Autoincrementing |
| Discount | A percentage off discount on the subscription price | decimal |  |  |  | NO | NO | 0.0 |
| Billing date | The date when the subscription is billed | date |  |  |  | NO | NO |  |
| Recurring | Whether the subscription will automatically renew. | boolean |  |  |  | NO | NO | False |
| Subscription type | Name | The name of the subscription model | string | Type |  |  | NO | YES |  |
| Price | The monthly cost of the subscription | decimal | Cost |  |  | NO | NO |  |
| Description | A description of the subscription’s benefits | string | Benefits |  |  | NO | NO |  |
| Customer | ID | A unique identifier | integer |  |  |  | NO | YES | Autoincrementing |
| Name | The customer’s full name | string, string |  | Comprised of fist name and last name. |  | NO | NO |  |
| Phone | The customer’s contact phone number | string with regex mask | Contact phone |  |  | YES | NO |  |
| Email | The customer’s contact email address | string with regex mask | Contact email |  |  | NO | NO |  |
| Company | The customer’s company, if applicable. | string |  |  |  | YES | NO | null |

## Rationale

For rational, refer to section 3.1. The same rationale applies for this section.

## Review of Data Management Procedures

Data dictionaries apply a high degree of data quality management by defining the datatypes used for each attribute. The rigid definition of attributes from data dictionaries eliminates the possibility of entering erroneous data into a database, enforcing data quality. Data dictionaries also provide data governance. A description of the purpose of an attribute and details on the attribute’s composition or derivation helps to ensure data complies with company policy. Descriptions of relationships and entities ensure data is easily useable and governable. Data dictionaries do not directly contribute to privacy and security. Like conceptual models though, they can be used to analyse what data is being stored in a system in order to highlight any privacy concerns and security needs. Data dictionaries help with data lifecycle management by defining exactly how data is stored and how it interacts with other data. The data dictionary also shows how data is created by defining occurrences for entities. A description of attributes also reveals how data can be used, assisting with lifecycle management. Data dictionaries define the metadata for database attributes, relationships and entities. Providing exact definitions for metadata makes data dictionaries an invaluable tool for metadata management.

# Assumptions on Business Rules

For this system, it is assumed that providing data from contracts to customer is handled by an external system, this results in not having a relationship between customer and contract directly. This is a reasonable assumption to make as there is no direct requirements to the customer entity to modify data in the contract, video stream, or data dump entities.

It is assumed that a contract relates to a specific geographic location. It is assumed that multiple subscriptions can be related to a contract unless the contract is negotiated as a super platinum contract. This is implied by the ability of multiple standard user to access a contract.

It is assumed that a super platinum contract can only be accessed by a single subscription. This is based off the fact that a super platinum subscriber owns the exclusive access rights to the data and video streams of a given contract.

It is assumed that FlightStream will have employee roles for piloting and maintaining DataScoop drones. This enables the creation of employee specialisations allowing control over which employees can participate in certain relationships.

It is assumed that “extreme cold – Ice and Snow” is meant to refer to a single biome, and not two distinct biomes due to use of a hyphen instead of a comma.

It is assumed that organic spectral data from one litre of air can be stored in a single attribute. What exactly this data entails is not defined, so this is assumed to be a single blob of data.

Platinum subscribers have exclusive rights to the video streams from a contract, however, it is later stated that only super platinum subscribers have exclusive *access* to video streams. This implies that other subscribers can view the video streams from a contract negotiated by a platinum subscriber, but do not own the video.

It is assumed that video streams are not stored, and it is up to the subscriber to record video streams themselves if they want.

Milestone 2

# Conceptual to Relational Logical Model

## Methodology

## Logical ERD

### Rationale

## Description of Normalisation Forms

### First Normal Form

### Second Normal Form

### Third Normal Form

### Boyce Codd Normal Form

### Fourth Normal Form

## Data Dictionary

# NaLER Analysis

# References

Craig Stedman. (2022, May 31). What is data governance and why does it matter? Data Management. <https://www.techtarget.com/searchdatamanagement/definition/data-governance>

Fortinet. (2024). Data security: Definition, importance, and types. <https://www.fortinet.com/resources/cyberglossary/data-security>

Gattis, T. (2024, February 14). Security, data privacy & data management | Best practices guide. Annex Cloud. <https://www.annexcloud.com/resources/guides/data-security-and-privacy-best-practices-guide/>

IBM. (n.d.). What is data modeling? Retrieved February 25, 2024, from <https://www.ibm.com/topics/data-modeling>

Jotform. (2020, August 4). 6 key data management principles. <https://www.jotform.com/blog/data-management-principles/>

Ranjan, S. (2024, January 1). Metadata management in data-driven organisations. Meer. <https://www.meer.com/en/75824-metadata-management-in-data-driven-organisations>

Sagacity solutions. (2024). What is data quality? Why it's important. <https://www.sagacitysolutions.co.uk/about/news-and-blog/what-is-data-quality/>

Sheldon, R. (2022, June 28). What is data lifecycle management (DLM)? TechTarget. <https://www.techtarget.com/searchstorage/definition/data-life-cycle-management>

Sherman, R. (2015). Foundational Data Modelling. ScienceDirect. <https://www.sciencedirect.com/topics/computer-science/conceptual-data-model>