DAT601 Assignment 1

Table of Contents

[Part 1. Conceptual database design – for the project 2](#_Toc166120756)

[0. An introduction to data modeling in information systems that includes an outline of the fundamental principles of effective data management. Include an overview of Conceptual, Logical data modelling and Physical implementation. (5 marks) (LO1, LO4) 2](#_Toc166120757)

[1. A description of conceptual modelling using Extended (aka Enhanced) Chen Entity Relationship Diagramming (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) 3](#_Toc166120758)

[2. A conceptual ER model using Chen Extended ERD notation of the database as derived from the project brief /case study. Accompany this with your rationale; describe and explain the reasoning and purpose of all parts of the model. Write and reflect on how te fundamental principles of data management are applied through conceptual ER **modelling** using the Chen Extended ERD. (20 Marks) (LO1, LO4) 10](#_Toc166120759)

[3. A data dictionary using the template provided on the course website, that covers the following: ○ Entities (e.g. name description, aliases, occurrences) ○ Relationships (e.g. name, multiplicity) ○ Attributes (e.g. name, description, domain, aliases, composite, derived, nulls, key, default value). 11](#_Toc166120760)

[4. Discuss the assumptions made about the business rules and the reasons for the choices you made. Include a reflection in outline of how the fundamental principles of data management as they are applied to assumptions about the business rules. (LO1, LO4)(5marks) 13](#_Toc166120761)

Assignment 1:

# Part 1. Conceptual database design – for the project

## An introduction to data modeling in information systems that includes an outline of the fundamental principles of effective data management. Include an overview of Conceptual, Logical data modelling and Physical implementation. (5 marks) (LO1, LO4)

* Data modeling in information systems is the process of creating a visual representation that defines the structure and relationship between data objects within the information systems of organizations and businesses *(IBM, n.d.).* To visualize data modelling, ‘Entity-Relationship Diagrams’ or ERD are commonly used which makes it easier to organize the collected information by which then can be used to build effective information systems that meets the organizations and business’s needs.
* There are 8 key principles in data management - Accountability, Transparency, Integrity, Protection, Compliance, Availability, Retention, and Disposition. These principles are known as Generally Acceptable Recordkeeping Principles (GARP) and it describes how data should be maintained. Incorporating these principles and implementing them in the development of information systems will allows for a better and effective data management *(Salahuddin, 2021).*
* There are three stages in database design methodology – Conceptual, Logical and Physical design. Each stage refers to a different level or phases of a database design process.
* Conceptual database design is the process of understanding the organizations requirement. This stage defines the overall structure of the database and focus on identifying the relationship of one data to another *(Visual Paradigm, 2023)* .
* Logical database design takes the previous phase, conceptual database design into more detail. This stage defines the relationship and the constraints between the data which helps ensure data integrity within the database. Logical database design process involves defining tables, entities, attributes, keys and the data types of the database.  
  This stage is independent on specific Database Management System (DBMS) and should align with the requirements and needs of the organization. To visualize the logical database design process, a data model such as an Entity-Relationship Diagram (ERD) is commonly used, which helps in understanding the relationship between the entities or data much clearer.
* Physical database design is the last stage of the database design process. This stage builds on the previous phases conceptual and logical design and begin the implementation on the chosen database management system (DBMS) and create a database.  
  The Physical design process involves considering the database storage, performance optimization, security, and access controls.

## A description of conceptual modelling using Extended (aka Enhanced) Chen Entity Relationship Diagramming (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)

* Enhanced Entity-Relationship diagram or EER is used when the complexity of data modelling rises, and the Entity-Relationship diagram (ERD) could no longer be used to visualize in conceptual modelling due to the complicated structure of organization *(javatpoint, n.d.)*.
* Conceptual modelling is used to represent the organization’s structure without independent of all physical considerations. It’s the stage of understanding the organization’s objects or entities, relationships, attributes and visualizing it using diagrams such as Chen Entity-Relationship (ERD) and Enhanced Entity-Relationship diagram (EER).
* Chen Entity-Relationship diagram (ERD) as well as the Chen Extended Entity-Relationship diagram (EER) is a widely used diagrams in representing an organization’s structure visually. There are various components, and each are represented with different shapes and symbols *(Chen Notation & Crow’s Foot | Gleek Diagram Maker, n.d.)*.
* Components of Chen ERDs includes:
* **Entities:** entities are real-world objects or concepts. They can either be physical as well as non-physical.
  + In Chen ERD, entities are distinguished into three types – Strong entity, Weak entity, and Associative entity.
    - Strong entity are entities that does not dependent on any other entity and always have a primary key. It is represented by a single ‘rectangle’ shape *(GreeksforGreeks, 2023).*
    - Weak entity are entities that depend on a strong entity to ensure its existence. They don’t have a primary key and is represented by a double ‘rectangle’ shape *(GreeksforGreeks, 2023).*
    - Associative entity are entities that is used to define the relationship between entities, often in many-to-many relationships. It is represented as a diamond with a surrounding rectangle *(fuchs, 2021).*

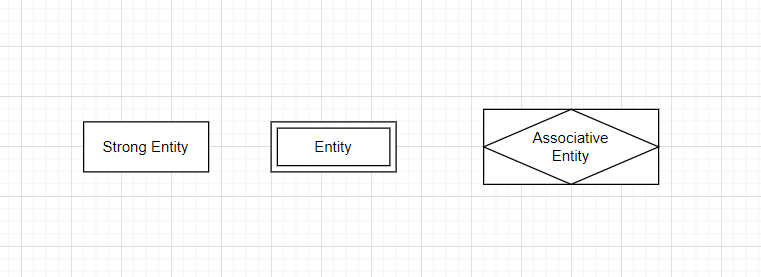


Figure 1Diagram representations of Strong Entity, Weak Entity and Associative Entity

**Attributes**:

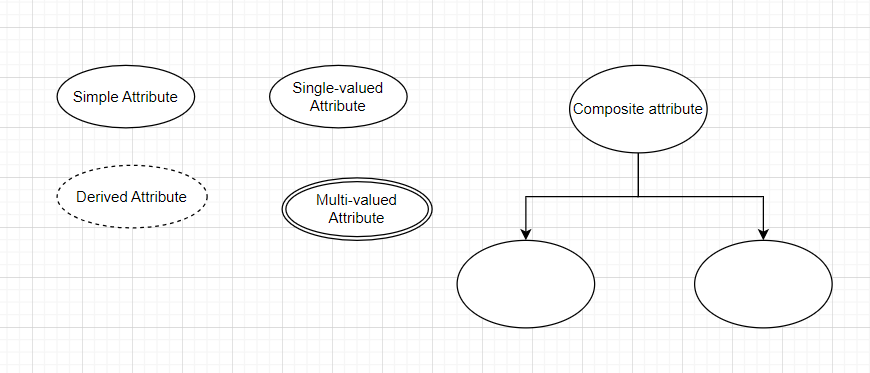
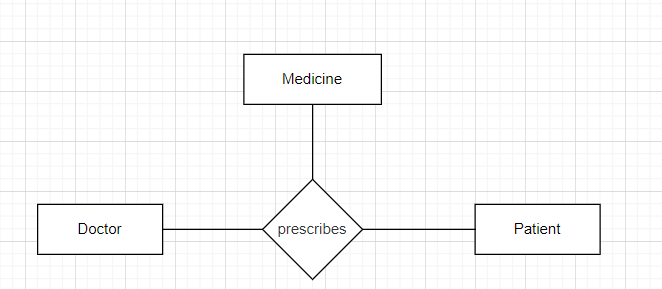
* Attributes are properties that describe the entity. They are represented with an ‘oval’ shape and are connected to its entity with a single line.
  + There are many types of attributes and each type have a different meaning in the context of a database. The attributes are categorized into 6 different attributes – Simple attribute, Composite attribute, Single valued attribute, Multi valued attribute, Derived attribute, and Stored attribute.
    - Simple attribute are atomic values that cannot be broken down further.
    - Composite attribute are attributes that can be further divided into smaller subparts.
    - Single valued attribute are attributes that hold just one value for particular entity.
    - Multi valued attribute are attributes that can store more than one value for an entity.
    - Derived attribute have its values derived from another attribute and is dependent on other attributes for its values *(Jain, 2022).*
    - Stored attribute are attributes that other attributes values are derived from.

Figure 2 Diagram representations of the different attributes

* Simple attribute and Single-valued attribute are both represented using the same shape ‘Oval/Ellipse’ shape.
* **Stored attribute** are attributes from which the values of other attributes are derived – something like ‘Date of Birth’ and the derived attribute would be ‘age’

**Relationships:**

* In Chen ERD, we can define the degree of a relationship into four different types - Binary, Ternary, n-array, quaternary and unary – also known recursive relationships.
  + Binary relationship refers to when two entities are involved and have a relationship.
  + In ternary relationship, there are three entities involved.
  + There can also be quaternary relationships when four entities are involved together in a relationship.
  + There is also an N-ary relationship when the number of entities involved in a relationship is many and is not define.
  + Unary relationships also known as ‘recursive relationship’ refers to when one entity plays more than one role in a relationship.

  
- This way of clarifying the relationships between entities helps with effective database design and management.

A diagram of a graph

Description automatically generated

Figure 3 Shows Ternary-Relationship

Figure 4 Shows Binary-Relationship

A diagram of a course

Description automatically generated

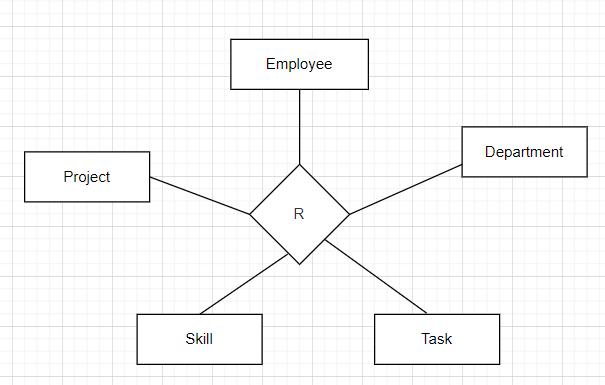


Figure 5 Shows Quaternary-Relationship

Figure 6 Shows N-ary Relationship

A diagram of a company

Description automatically generated

Figure 7 Shows Unary-Relationship

* To define the types of relationship between entities, there are four possible ways to identify in Chen ERD’s notation: one-to-one, one-to-many, many-to-one, and many-to-many relationships. It is also known

as ‘Cardinality’.

* In Chen ERD’s notation, the ‘one’ side of relationship is represented as ‘1’ while the ‘many’ said of the relationship is represented as ‘N’.
* There is also another widely used notation called ‘Crow’s foot notation’ and it uses two graphic symbols on relationship lines to represent cardinality.

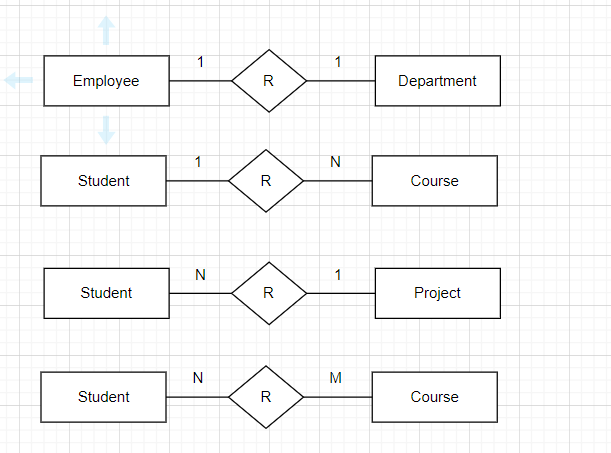


Figure 8 Diagrams: Types of Relationship

**Identifier/Key Attributes**

* Identifier attributes also known as key attributes are used to uniquely identify each entity and therefore is mandatory *(Relationaldbdesign, n.d.)*. Key attributes in CHEN’s ERD are represented by underlining the attributes. In database design, the key attributes are known as the primary key – which can be either combination of attributes set as unique identifier, or a specific attribute chosen.
* There can be more than one identifier in each entity -such as candidate key, super key, composite key, and foreign key.
  + Super key refers to an attribute or set attributes that uniquely identify an entity within a table.
  + Candidate key is a set of attributes that uniquely identified a table. A table can have multiple candidate keys.
  + Composite key is a candidate key that contains two or more attributes and uniquely identifies any record.
  + A Foreign key refers to an attribute or set of attributes that references the primary key in another table.

Extended Components (EER)

* Enhanced entity-relationship diagram (EER) is an extended version of entity-relationship diagram (ERD). Enhanced entity-relationship diagram is a high-level model and is used to represent complexities in complex databases *(Lucidchart, n.d.).*
* **Subclass and super class**
  + Super class is an entity that can be divided into further smaller subgroup while subclass are entity or instances of the super class, that inherited the properties and attribute of the super class *(geeksforgeeks, 2024).*

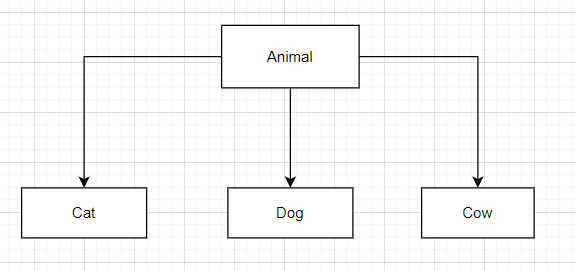


Figure 9 Superclass and Subclasses

* **Specialization and Generalization**
  + Specialization is the process of breaking down an entity into a smaller subset that share characteristics of the main entity.
  + Generalization refers to the process of identifying common features of several entities and grouping them into a single entity – a super class. This process of generalization is identifying the single parent entity of several entities or children *(geeksforgeeks, 2024).*

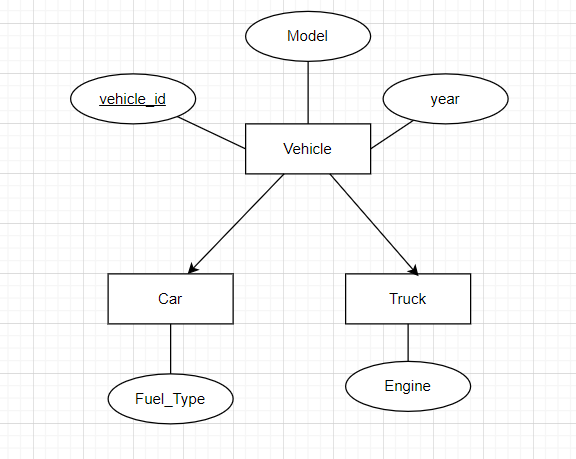
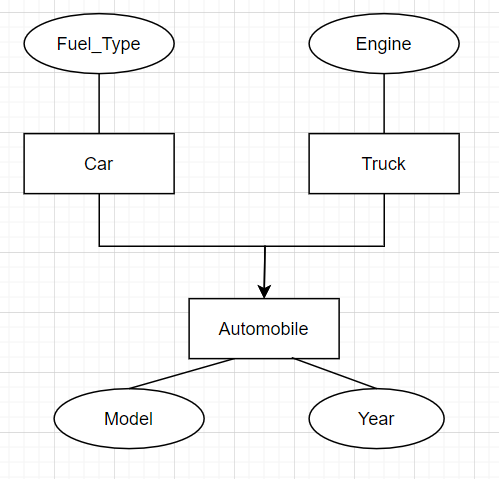


Figure 11 Specialization

* **Categorization**

Figure 10 Generalization

* + Categorization also known as Union is when a subset has relationship with more than one super class *(Barnes, 2023).*

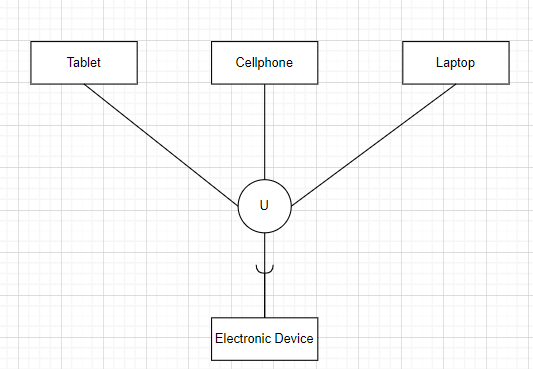
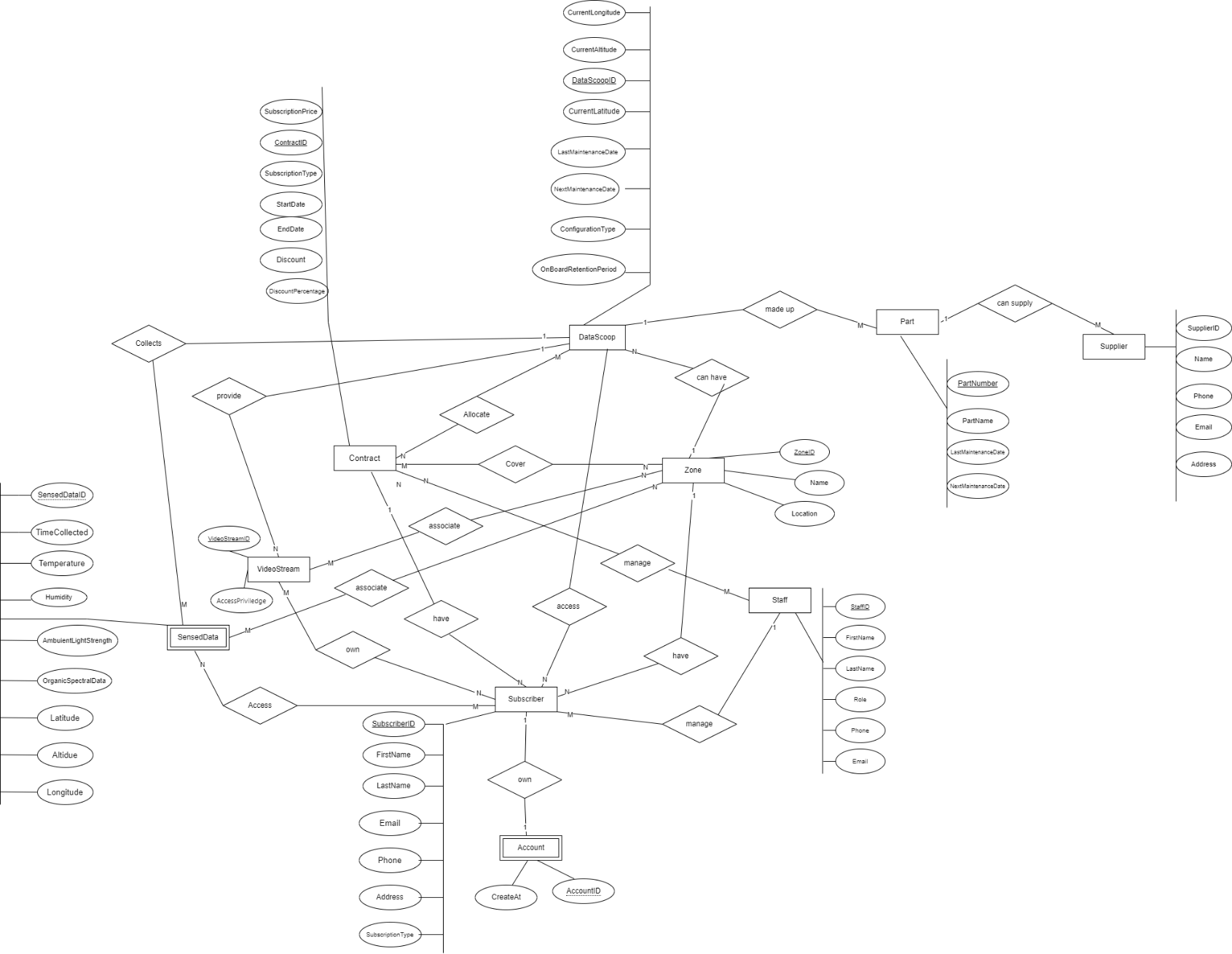


Figure 12 Categorization (UNION TYPES)

* **Attribute and Relationship Inheritance**
  + Enhanced entity-relationship diagram (EER) allows sub-classes to inherited attributes and its values from the superclass that it related to. *(Wikipedia, 2024).*
  + Relationship inheritance refers to a subclass entity inheriting the relationships that superclass participates in *(Wikipedia, 2024).*

## A conceptual ER model using Chen Extended ERD notation of the database as derived from the project brief /case study. Accompany this with your rationale; describe and explain the reasoning and purpose of all parts of the model. Write and reflect on how the fundamental principles of data management are applied through conceptual ER **modelling** using the Chen Extended ERD. (20 Marks) (LO1, LO4)

* Conceptual ER Model



* The purpose of this conceptual model is to represent the identified entities, attributes and the relationships involved from the data dictionary into a clearer and a better formatted version. This conceptual model uses Chen entity-relationship diagram components to shows the entities, attributes, and the relationship.
* This conceptual model used a rectangle-shape to represent what each entity are in terms of whether the entity is a weak or strong entity.   
  The different types of attributes are also shown in this model such as the primary-key attribute of the entity by underlying the attributes.
* The relationship or cardinality between the entities in this conceptual model are represent using the diamond-shape and straight line which are connected to the specific entities.
* This steps of creating a conceptual model allows for further discovering of any missed errors and anomalies data for the database.  
  It also helps deepen the understanding of what each data are and other important information.
* The conceptual model further ensures that the data are refined and are high quality as well as being consistent across for the whole database. These apply the principle of Integrity, Transparency and Protection of the fundamental principles of Data Management.

## A data dictionary using the template provided on the course website, that covers the following: ○ Entities (e.g. name description, aliases, occurrences) ○ Relationships (e.g. name, multiplicity) ○ Attributes (e.g. name, description, domain, aliases, composite, derived, nulls, key, default value).

Accompany this with your rationale; describe and explain the reasoning and purpose of all parts of the model. Write and reflect on how the fundamental principles of data management are applied by using data dictionaries. (LO1, LO4)(10 marks)

Purpose of all parts of the model

* In this Conceptual ER model, there are three main components -Entity, Attribute and Cardinality (or relationship).
* This conceptual model has 10 total identified entities – Datascoop, Contract, Part, Zone, Subscriber, Staff, Supplier, SensedData and VideoStream and Account.
* **DataScoop entity** represent each drone that is used for data collecting and to provide live 3D video stream. It has attributes like ‘DataScoopID’, ‘CurrentAltitude’, ‘CurrentLongitude’, ‘CurrentLatitude’, ‘LastMaintenanceDate’ , ‘NextMaintenanceDate’ and ‘OnBoardRetentionPeriod’. These attributes provide important information that is need for managing this entity.

DataScoop entity has relationship with ‘SensedData’, ‘Contract’, ‘Zone’, ‘Part’, ‘Subscriber’ and ‘VideoStream’ entities.

* **Contract entity** represent a subscription contract between the subscriber and the FlightStream organization. It has attributes like ‘ ContractID’, ‘SubscriptionType’, ‘StartData’, ‘EndDate’, ‘SubscriptionPrice’, ‘Discount’ and ‘DiscountPrecentage’. These attributes are important and crucial to for Contract entity to have.  
  This entity has relationship with ‘Zone’, ‘Subscriber’, ‘Staff’ and ‘DataScoop’ entities.
* **Part entity** represent each physical component that made up the drone/DataScoop. It has several attributes – ‘PartNumber’, ‘PartName’, ‘LastMaintenanceDate’ , and ‘NextMaintenanceDate’.  
  These attributes contain values that contains important information relating to this ‘Part’ entity.  
  Part entity has relationship with ‘DataScoop’ and ‘Supplier’ entities.
* **Zone entity** represents a geographic region where the FlightStream company released DataScoop to collect data.  
  It has attributes like – ‘ZoneID’, ‘Name’ and ‘Location’ that provides information regarding to this entity.  
  Zone entity has relationship with ‘DataScoop’, ‘Contract’, ‘Subscriber’, ‘VideoStream’ and ‘SensedData’.
* **Subscriber entity** represent subscriber who brought subscription/contract from FlightStream company. It has many attributes - ‘SubscriberID’, ‘FirstName’, ‘LastName’, ‘Email’, ‘Phone’, ‘Address’ and SubscriptionType’.  
  These attributes provide crucial information about Subscriber entity.  
  Subscriber entity has relationship with ‘Contract’, ‘Zone’, ‘Staff’, ‘DataScoop’, ‘SensedData’, ‘Account’ and ‘VideoStream’ entities.
* **Staff entity** represent employee members of the FlightStream organization. It has attributes like – ‘StaffID’, ‘FirstName’, ‘LastName’, ‘Email’, ‘Phone’ and ‘Role’.  
  These attributes provide information about the staff entity.

Staff entity has relationships with ‘Contract’ and ‘Subscriber’ entities.

* **Supplier entity** represents supplier where the FlightStream organization get their drone part supply from. This entity has attributes including – ‘SupplierID’, ‘SupplierName’, ‘Phone’, ‘Email’ and ‘Address’. These attributes provide information regarding the Supplier entity.

Supplier entity has relationship with ‘Part’ entity.

* **SensedData entity** represents data that are sensed and collected by the DataScoop. It has several attributes – ‘SensedDataID’, ‘DataScoopID’, ‘TimeCollected’, ‘Temperature’, ‘Humidity’, ‘AmbuientLightStrength’, ‘OrganicSpectralData’, ‘Latitude’, ‘Longitude’ and ‘Altitude’.

These attributes include the types of data the DataScoop can sensed and collected as well as other crucial information.

SensedData entity has relationships with ‘DataScoop’, ‘Subscriber’ and ‘Zone’ entities.

* **VideoStream entity** represents the data of viewing of live 3D video from DataScoop.  
  It has attributes like –‘VideoStreamID’ and ‘AccessPriviledge’. These attributes contain important information regarding the VideStream entity.  
  VideoStream entity has relationships with ‘Subscriber’, ‘Zone’ and ‘DataScoop’ entities.
* **Account entity** represents an account owns by subscriber. It has ‘AccountID’ and ‘CreatedAt’ attributes. These two attributes are crucial to Account entity.

Account entity has relationships with ‘Subscriber’ entity.

Data dictionaries provide a collection of data entities, attributes and relationships which are analysed and structured in a way that can be effectively used in a database.  
It helps to understand the overall design and structure of a database before attempting to create an actual database.  
This allows errors and anomalies in data to be discovered earlier which makes the whole process of creating a database much faster and efficient. It also assists with data consistency and security as well as managing quality of data *(Atlan, 2023).*

## Discuss the assumptions made about the business rules and the reasons for the choices you made. Include a reflection in outline of how the fundamental principles of data management as they are applied to assumptions about the business rules. (LO1, LO4)(5marks)

**Assumptions:** Each DataScoop can be associate with one or more Contract – this would result in a one-to-many relationship.  
However, the project brief states that *“For contracts that have overlapping regions a DataScoop may be allocated to more than one contract.”*  
This implies that the relationship between DataScoop and Contract is a many-to-many relationship instead.  
Identifying the correct relationship between the entities ensure data integrity.

**Assumptions:** A Staff member can manage many Contracts. However, as the project brief states “*FlightStream employs staff who can take an Administrative Executive role. Only staff who have this role may change the price of subscriptions. Only an Administrative Executive may enter contract details.”*   
While the assumption is correct, this only apply to a staff member whose role is Administrative Executive.  
This adds constraints of who has access to what specific information in the database, ensuring the Principle of Accountability, Protection and Compliance.

**Assumptions:** Each DataScoop having to keep record of maintenance. The project brief states *“FlightStream is required to keep maintenance records for each DataScoop. Over time there is a record of the maintenance of each DataScoop.”*The DataScoop entity has attributes ‘LastMaintenanceDate’ and ‘NextMaintenanceDate’ which stored the date the maintenance checking was done and need to be done.   
This information allows for easier to track and manage the Datascoop and applying the Principle of Retention, Disposition and Availability of data management.

References:

Atlan. (2023, March 17). *Top 6 Benefits of a Data Dictionary (2024 Guide)*. https://atlan.com/benefits-of-a-data-dictionary/

Barnes, R. (2023, September). *Extended Entity-Relationship (EE-R) Model*. https://www.tutorialspoint.com/Extended-Entity-Relationship-EE-R-Model

*Chen notation & Crow’s Foot | Gleek diagram maker*. (n.d.). Retrieved 9 March 2024, from https://www.gleek.io/er-diagrams

fuchs, michael. (2021, 03). *Entity-Relationship-Diagram (ERD)—Michael Fuchs SQL*. https://michael-fuchs-sql.netlify.app/2021/03/03/entity-relationship-diagram-erd/#associative-entity

geeksforgeeks. (2024, March 22). *Generalization, Specialization and Aggregation in ER Model*. GeeksforGeeks. https://www.geeksforgeeks.org/generalization-specialization-and-aggregation-in-er-model/

GreeksforGreeks. (2023, February 22). *Difference between Strong and Weak Entity*. GeeksforGeeks. https://www.geeksforgeeks.org/difference-between-strong-and-weak-entity/

IBM. (n.d.). *What is Data Modeling? | IBM*. Retrieved 27 February 2024, from https://www.ibm.com/topics/data-modeling

Jain, M. (2022, May 25). *Types of Attributes in DBMS*. Scaler Topics. https://www.scaler.com/topics/types-of-attributes-in-dbms/

javatpoint. (n.d.). *Enhanced ER Model—Javatpoint*. Www.Javatpoint.Com. Retrieved 9 March 2024, from https://www.javatpoint.com/enhanced-er-model

Lucidchart. (n.d.). *Enhanced ER Diagram Tutorial*. Lucidchart. Retrieved 1 May 2024, from https://www.lucidchart.com/pages/enhanced-entity-relationship-diagram

Relationaldbdesign. (n.d.). *Purpose of Entity Identifiers (Attributes)*. Retrieved 22 March 2024, from https://www.relationaldbdesign.com/database-design/module5/store-entity-data-values.php

Salahuddin, M. (2021, December 19). *8 Key Principles of Data Management | LinkedIn*. https://www.linkedin.com/pulse/8-key-principles-data-management-muhammad-salahuddin/

Visual Paradigm. (2023, September 18). Navigating the Three Levels of Database Design: Conceptual, Logical, and Physical. *Visual Paradigm Guides*. https://guides.visual-paradigm.com/navigating-the-three-levels-of-database-design-conceptual-logical-and-physical/

Wikipedia. (2024). Enhanced entity–relationship model. In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Enhanced\_entity%E2%80%93relationship\_model&oldid=1212862365

**DAT601 - DATABASE DESIGN METHODOLOGY**

**Conceptual Model Documentation**

**Table 1: Document Entities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity Name** | **Description** | **Aliases** | **Occurrence** |
| DataScoop | Drones deploys by Flight Stream |  | Is deploy as needed by the geographic coverage requirements of the contracts. |
| Contract | Subscription in the form of contract | Subscription | Is created when a subscriber purchases a subscription for services. |
| Part | Parts of the drones |  | Data scoop’s physical parts |
| Zone (region) | Region where the datascoop can be found |  | Is define when a new geographic region is added to the lists for datascoop to collect data. |
| Subscriber | Who subscribe to DS subscription |  | Is created when a person or an organisation brought a subscription from FlightStream company. |
| Staff | The employee of the Fight Stream |  | Is created for existing and new member of Flightstream organisation’s employee/staff. |
| Supplier | From which the drone’s parts is supply |  | Is created for existing and new supplier that supply parts for Datascoop. |
| SensedData | Data senses by each DataScoop |  | Is created for every data that is collected by each Datascoop.   * Is a weak entity dependent on Datascoop. |
| VideoStream | Viewing of live 3D video from datascoop. data from the video stream. |  | created when datascoop enable live 3d video. |
| Account | Account owns by subscriber. |  | Created every time there is a new subscriber. |

**Table 2: Document Relationships**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Cardinality** | **Participation** | **Relationship** | **Participation** | **Cardinality** | **Entity Name** |
| DataScoop | 1 | Partial | collects | Full | n | SensedData |
| DataScoop | m | Partial | allocate | Partial | n | Contract |
| DataScoop | 1 | Partial | cover | Partial | n | Zone |
| DataScoop | 1 | Full | Made up | Full | n | Part |
| Contract | 1 | Full | have | Full | n | Subscriber |
| Contract | 1 | Full | Cover zone | Full | n | Zone |
| Staff | 1 | Partial | Sell many | Partial | n | Contract |
| Zone | 1 | Full | Can have | Full | n | Subscriber |
| Supplier | N | Full | Supply many | Full | 1 | Part |
| Staff | 1 | Partial | manages | Partial | n | Subscriber |
| DataScoop | 1 | Partial | access | Partial | n | Subscriber |
| SensedData | N | Full | own | Full | M | Subscriber |
| Subscriber | 1 | Full | have | Full | 1 | Account |
| Subscriber | N | Full | Can own | partial | M | VideoStream |
| Zone | N | Full | Associate with | Partial | M | VideoStream |
| Datascoop | 1 | Full | provide | Partial | N | VideoStream |
| SensedData | M | Partial | Be collect in | Full | N | Zone |

**Table 3: Document Attributes**

| **Entity / Relationship Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DataScoop | DataScoopID | A unique id for every datascoop | Integer generated for every datascoop |  |  |  | NOT NULL | YES | Auto-generated number - 1 |
|  | CurrentAltitude | Current location altitude of the datascoop | Decimal: to represent the current altitude. |  |  |  | NOT NULL |  | 0 |
|  | CurrentLongitude | Current location longitude of the datascoop | Decimal: to represent the current latitude |  |  |  | NOT NULL |  | 0 |
|  | CurrentLatitude | Current location latitude of the datascoop | Decimal : to represent the current latitude |  |  |  | NOT NULL |  | 0 |
|  | LastMaintenanceDate | The date the datascoop was last maintained | DATE |  |  |  | NOT NULL |  | --/--/---- |
|  | NextMaintenanceDate | The next date for the datascoop to be check in | DATE |  |  |  | NOT NULL |  | --/--/---- |
|  | ConfigurationType | The type of configuration that the datascoop can be configured. | One of Enum type of configuration. |  |  |  | NOT NULL |  | First Enum value |
|  | OnBoardRetentionPeriod | Number of days data is stored on board in Datascoop | Integer: days |  |  |  | NOT NULL |  | 0 |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SensedData | SensedDataID | A unique id to refer to the data sensed by datascoop. | Auto generated. |  |  |  | NOT NULL | YES 1 | |
|  | DataScoopID | Refer to the Datascoop entity. | Auto generated. |  |  |  | NOT NULL | 1 | |
|  | TimeCollected | The exact time of when the data was collected. | Timestamp |  |  |  | NOT NULL | YYYY-MM-DD HH:MI:SS | |
|  | Temperature | The temperature data that datascoop collects. | Decimal Data type |  |  |  | NOT NULL | 0 | |
|  | Humidity | The humidity data that datascoop collect. | Float datatype |  |  |  | NOT NULL | 0 | |
|  | AmbuientLightStrength | The data sensed by datascoop. | Decimal datatype |  |  |  | NOT NULL | 0 | |
|  | OrganicSpectralData | The data sensed by datascoop. | Decimal datatype |  |  |  | NOT NULL | 0 | |
|  | Latitude | Refers to the latitude of when the data was collected | Decimal |  |  |  | NOT NULL | 0 | |
|  | Longitude | Refers to the longitude of when the data was collected. | Decimal |  |  |  | NOT NULL | 0 | |
|  | Altitude | Refers to the longitude of when the data was collected. | Decimal |  |  |  | NOT NULL | 0 | |
| Contract | ContractID | A unique id for every contract. | Integer |  |  |  | NOT NULL | YES 1 | |
|  | SubscriptionType | Types of subscription available | Enum String |  |  |  | NOT NULL |  | |
|  | StartDate | The date the contract begins. | Date |  |  |  | NOT NULL |  | |
|  | EndDate | The date the contract end. | Date |  |  |  | NOT NULL |  | |
|  | SubscriptionPrice | The price value of the specific subscription . | Decimal |  |  |  | NOT NULL |  | |
|  | Discount | Whether a discount is given or not | Boolean |  |  |  | NOT NULL |  | |
|  | DiscountPercentage | Percentage of discount given by saleperson. up to 3% | Integer |  |  |  |  |  | |
|  |  |  |  |  |  |  |  |  | |
|  |  |  |  |  |  |  |  |  | |
| ContractZone | ContractID | Foreign key in ContractZone entity | Integer |  | yes |  | NOT NULL | Auto-generated | |
|  | ZoneID | Foreign key in ContractZone entity | Integer |  | yes |  | NOT NULL | Auto-generated | |
| ContractSubscriber | ContractID | Foreign key reference | Integer |  | yes |  | NOT NULL | Auto-generated | |
|  | SubscriberID | Foreign key reference | Integer |  | yes |  | NOT NULL | Auto-generated | |
|  |  |  |  |  |  |  |  |  | |
| StaffContract | ContractID | Foreign key reference | Integer |  | yes |  | NOT NULL | Auto-generated | |
|  | StaffID | Foreign key reference | Integer |  | yes |  | NOT NULL | Auto-generated | |
|  |  |  |  |  |  |  |  |  | |
| Subscriber | SubscriberID | Unique id for every subscriber | Integer |  |  |  | NOT NULL | YES Auto-generate | |
|  | FirstName | First name of subscriber | String |  |  |  | NOT NULL |  | |
|  | LastName | Last name of subscriber | String |  |  |  | NOT NULL |  | |
|  | Email | Email address of subscriber | String |  |  |  | NOT NULL |  | |
|  | Phone | Phone number of subscriber. | Integer |  |  |  |  |  | |
|  | Address | Current residency address of subscriber | String |  |  |  | NOT NULL |  | |
|  | SubscriptionType | The type of subscription that the subscriber subscribed to. | String |  |  |  | NOT NULL |  | |
|  |  |  |  |  |  |  |  |  | |
| Staff | StaffID | Unique id for every staff | integer |  |  |  | NOT NULL | YES auto-generate | |
|  | FirstName | First name of staff | String |  |  |  | NOT NULL |  | |
|  | LastName | Last name of staff | String |  |  |  | NOT NULL |  | |
|  | Email | Email address of the staff | String character |  |  |  | NOT NULL |  | |
|  | Phone | Phone number of the staff | integer |  |  |  |  |  | |
|  | Role | Type of role | String |  |  |  | NOT NULL |  | |
|  |  |  |  |  |  |  |  |  | |
|  |  |  |  |  |  |  |  |  | |
| Zone | ZoneID | Unique id for each zone | Integer |  |  |  | NOT NULL | YES auto-generate | |
|  | Name | Name of the zone | String |  |  |  | NOT NULL |  | |
|  | Location | The location or coordinates of the zone | String |  |  |  | NOT NULL |  | |
|  |  |  |  |  |  |  |  |  | |
| Supplier | SupplierID | Unique id for each supplier | Integer |  |  |  | NOT NULL | YES auto generate | |
|  | SupplierName | The name of the supplier | String |  |  |  | NOT NULL |  | |
|  | Phone | The phone number of the supplier | Integer |  |  |  | NOT NULL |  | |
|  | Email | The email address of the supplier | String |  |  |  | NOT NULL |  | |
|  | Address | The location of the supplier |  |  |  |  | NOT NULL |  | |
|  |  |  |  |  |  |  |  |  | |
| Part | PartNumber | Unique number for each part by manufacturer. | String |  |  |  | Not Null |  | |
|  | PartName | The name of the part. | String |  |  |  | Not null |  | |
|  | LastMaintenanceDate | The last date time it was maintenance | Date |  |  |  | Not null |  | |
|  | NextMaintenanceDate | The next date due for maintenance | Date |  |  |  | Not null |  | |
|  |  |  |  |  |  |  |  |  | |
| PartSupplier | PartID | Foreign key to part table | Integer |  | yes |  | Not null | Auto generate | |
|  | SupplierID | Foreign key to supplier table | Integer |  | yes |  | Not null | Auto generate | |
|  |  |  |  |  |  |  |  |  | |
| ContractDataScoop | ContractID | Foreign key reference | Integer |  | Yes |  | Not null |  | |
|  | DataScoopID | Foreign key reference | Integer |  | Yes |  | Not null |  | |
| Account | AccountID | Unique number for each account number. is weak-entity, dependent on ‘Subscriber’ | Integer |  |  |  | Not null | Yes auto-generate | |
|  | CreatedAt | The date the account was created | Date |  |  |  | Not null |  | |
| VideoStream | VideoStreamID | Unique identifier for every video stream. |  |  |  |  | Not null | Yes auto-generate | |
|  | AccessPriviledge | Define the level of access each level of contract allow. | Enum |  |  |  | NOT NULL | One of the first value of enum. | |
|  | ZoneID | Foreign key reference |  |  |  |  | NOT NULL |  | |
|  | DatascoopID | Foreign key reference |  |  |  |  | NOT NULL |  | |
| VideostreamSubscriber | videostreamID | Reference to the foreign table ‘VideoStream’  Junction table | Integer |  |  |  | NOT NULL |  | |
|  | SubscriberID | Reference to the foreign table ‘Subscriber’  Define junction table | Integer |  |  |  | NOT NULL |  | |
| ZoneSensedData | ZoneID | Junction table  Foreign key reference | Integer |  |  |  | Not null | yes Auto-generate | |
|  | SensedData | Junction table  Foreign key reference | Integer |  |  |  | Not null | Yes auto-generate | |