Create an introduction to conceptual modelling and a conceptual ERD for the digital company Spaces©. Demonstrate the construction of Transact-SQL queries.

DAT601 Assessment One

Conceptual Modelling and T-SQL Practice

3/5/23

Jayden Houghton – 13524050

Contents

[Introduction 1](#_Toc134025166)

[Part One – Conceptual Database Design 2](#_Toc134025167)

[Introduction to Data Modelling 2](#_Toc134025168)

[Conceptual Model 2](#_Toc134025169)

[Logical Model 2](#_Toc134025170)

[Physical Model 3](#_Toc134025171)

[Diagram Example 3](#_Toc134025172)

[Conceptual Modelling 3](#_Toc134025173)

[Description 3](#_Toc134025174)

[Components of Chen’s Notation 4](#_Toc134025175)

[Extended Chen’s Notation 7](#_Toc134025176)

[Conceptual ERD 8](#_Toc134025177)

[Data Dictionary 9](#_Toc134025178)

[Entities 9](#_Toc134025179)

[Relationships 11](#_Toc134025180)

[Attributes 12](#_Toc134025181)

[Business Rules 13](#_Toc134025182)

[Part Two – TSQL Queries 16](#_Toc134025183)

[Basic Queries 16](#_Toc134025184)

[Sub Queries 19](#_Toc134025185)

[Combined Query 20](#_Toc134025186)

[Views 21](#_Toc134025187)

[References 22](#_Toc134025188)

# Introduction

This assignment will cover an introduction and overview of database modelling. It will explore conceptual modelling and the components of Chen’s Notation, including the extended notation. A conceptual model and data dictionary tables designed for Spaces© is included. This assignment will also demonstrate basic transact-SQL queries on a small business database.

# Part One – Conceptual Database Design

## Introduction to Data Modelling

*An introduction to data modelling in information systems. Include an overview of Conceptual, Logical and Physical data modelling.*

Data modelling is the process of creating a visual representation of an information system’s database (What is data modeling?, Retireved 2023). The goal is to highlight the data points and structures, the data used and stored within the system, and the relationships between these, so data can be grouped and organised in an efficient and effective manner. This results in a more cohesive database structure plan that can be applied to various database management systems (DBMS). Business needs can also be easily considered and checked that they can be applied within the system being designed.

In data modelling, there are three main models that are used: conceptual, logical and physical. Each cover different levels of detail in relation how the database will work and the physical implementation.

### Conceptual Model

Conceptual models offer an overview of what the database will contain, how it will be organised, and which business rules are involved, independent of all physical considerations. This is to show the overall structure of how the database will be designed (Sherman, 2015). Chen’s notation is one of the most commonly used notation system for creating conceptual models.

### Logical Model

Logical models cover the same database information as the conceptual model, however they represent it in the specific data model that has been selected for the database. These models are used to establish the structure of data elements and the relationships between them to show how the system will be implemented (What is a Logical Data Model?, Retrieved 2023). The main focus of a logical model is how the system has to be implemented.

### Physical Model

A physical diagram is the third step in the process, with the diagram describing how the system will be implemented using the specific DBMS selected. This will include the final selected keys and the exact data types (with lengths assigned, default values given and allow null chosen) selected in the DBMS. This will be used to help with the actual implementation of the database.

### Diagram Example

Diagram

Description automatically generated

(Data Modeling, Retrieved 2023)

## Conceptual Modelling

*A description of conceptual modelling. An introduction and description of the components of a Chen Entity-Relationship Diagram ERD, include extended components.*

### Description

Conceptual modelling is the process of creating an abstract and high-level (but not necessarily) summary of the database’s content, independent of all physical considerations. This creates a clear and concise overall model to assist with understanding how the database is designed. This is essential for ensuring that designers, developers and stakeholders are on the same page and can communicate effectively about the design.

There are many different ways to conceptually model a database. Chen’s notation is one of the most used notation method, that provides a more in-depth look at the database then many other forms.

Chen’s notation was developed by Peter Chen in 1976, who was one of the pioneers of using entity relationship concepts in information system modelling (Chen Notation, 2014). Chen’s notation falls under the category of an Entity Relationship Diagram (ERD), as it uses entities and their relationships to represent the database. Due to this nature, it is primarily used for relational databases but can be adapted for non-relational databases (Shin, Hwang, & Jung, 2017).

### Components of Chen’s Notation

Component summary information (Chen Notation, 2014).

**Entities**

* These are represented by rectangles.
* Entities represent real-world things or objects that can be distinguished from each other.
* These will become the tables in relational logical or physical models.
* Example: a book or an author.
* There are different types of entities:
  + Strong – can be uniquely identified by its attributes (e.g. a book)
  + Weak – depends on an owner entity (e.g. a book chapter)
  + Associative – represents a many-to-many relationship table(e.g. books to authors relationship, as authors can write multiple books and books can be written by multiple authors)

Graphical user interface, application

Description automatically generated

**Attributes**

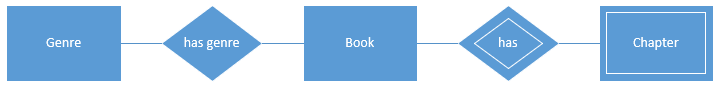
* These are represented by ovals.
* Attributes are linked to its parent entity with a line and represent a property of the entity.
* There are different types of attributes:
  + Simple – a basic property of the entity (e.g. book title)
  + Key – uniquely identifies an entity (e.g. International Standard Book Number (ISBN) code)
  + Partial key (discriminator) – combined with the owner entity key to make a key for the weak entity (e.g. book ID + chapter ID, if you don’t know the book, you can’t find the chapter)
  + Multivalued – can contain multiple values (e.g. book subject, can fit into multiple categories)
  + Derived – value calculated from other attributes (e.g. age calculated from date of birth)
  + Composite – contains smaller attributes (e.g. author name could be split into first name and last name)

Diagram

Description automatically generated

**Relationships**

* Represented by diamonds.
* Relationships connect two entities together to show how they relate and interact.
* There are different types of attributes:
  + Strong – entity is independent of other entities (e.g. book)
  + Weak – child entity depends on parent entity (e.g. a book’s chapter)



**Cardinality**

* Represented by the characters “1”, “N” and “M”.
* Cardinality shows how many records are related between entities in a relationship.
* There are four types of cardinality:
  + One-to-one (1:1) – One record is related to one record (e.g. a book only has one ISBN number and each number can only relate to one book)
  + One-to-many (1:N) – One record can relate to many records (e.g. one author can have many books but each is only written by one author)
  + Many-to-one (N:1) – Many records can relate to one record (e.g. many books belong to one library but that book cannot be in other libraries at the same time)
  + Many-to-many (N:M) – Many records relate to many records (e.g. many books can have many genres and each genre can relate to many books)

A picture containing chart

Description automatically generated

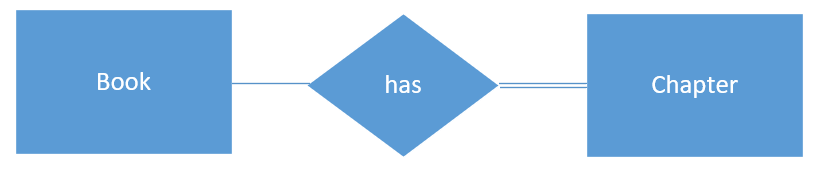
**Optionality**

* Represented by a solid or dashed line connecting entities to the relationship.
* Optionality refers to whether an entity can exist independantely of the relationship.
* There are two types:
  + Mandatory – Shown by a solid line (e.g. a book must have an author)
  + Optional – Shown by a dashed line (e.g. an author doesn’t have to have a book)



**Participation**

* Represented by a single or double line connecting entities to the relationship.
* Participation refers to whether all records must be involved in the relationship or only some.
* There are two types:
  + Total – Shown by two lines (e.g. all chapters must be related to a book)
  + Partial – Shown by a single line (e.g. each book doesn’t need to be related to a chapter)

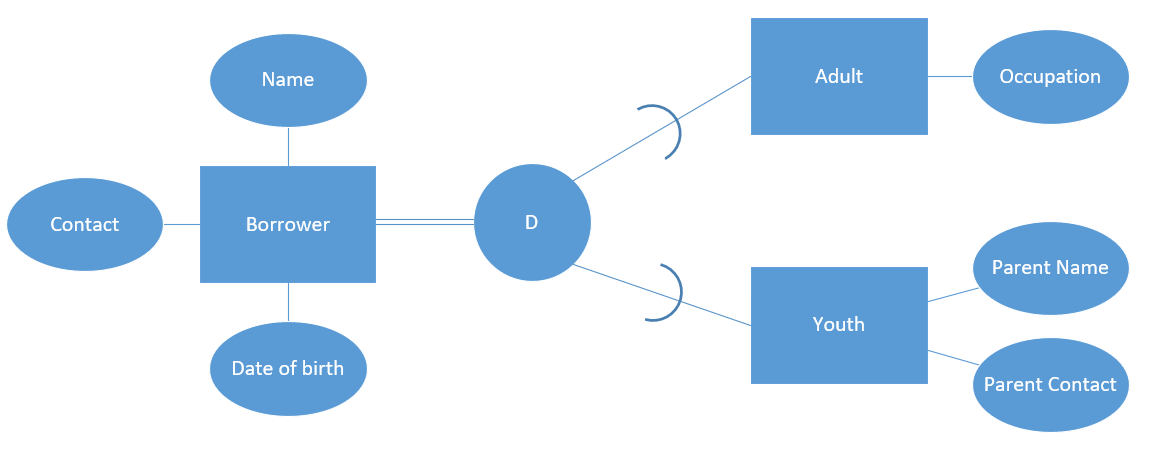


### Extended Chen’s Notation

Basic Chen’s Notation can be too simple to model more complex databases. In these cases, it can be extended to include extra notation concepts. The most common is generalisation and specialisation, which is based off the concept of Object Orientation (Yue, 2018). This extension uses the idea of a general superclass which has multiple specialised subclasses. This eliminates repeating similar (but slightly different) entites by combining them into a class system.

Components:

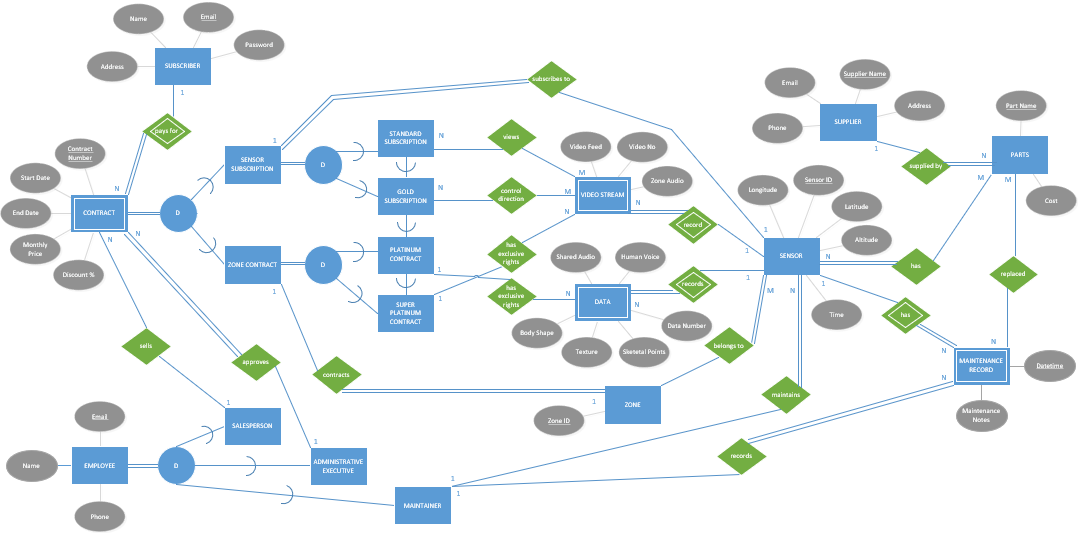
* Superclass – The generalisaed entity that contains the shared attributes (e.g. borrower)
* Subclass – The specialised entity that contains the attributes only for that specific type (e.g. youth borrowser)
* These are connected via a circle. This circle will contain either the letter “D” or “O”.
  + D – Means that each subclass is disjoint. Each instance can only have one subclass.
  + O – Means that the subclasses are overlapping. Each instance can have multiple subclasses.
* Each line connecting the subclass to the circle has a “C” shape on it to show that the entity includes all of the attributes from the superclass.



## Conceptual ERD

*A conceptual ER model using Chen Extended ERD notation of the database as derived from the case study. Accompany this with your rationale; describe and explain the reasoning and purpose of all parts of the model.*

Please refer to the conceptual model PDF file.



I have decided to have subscription as a separate entity to subscriber, so a subscriber can have multiple subscriptions, allowing access to multiple sensors and/or hosting zones.

I decided to have all contracts and subscriptions under a superclass called “Contract”. This is because they have the same attributes, so this avoids repetition. This class is broken into relevant subclasses to categorise and hold the different subscription levels.

I decided to have another subclass for my employees, again because of duplicated attributes.

The sensor subscription entity has a relationship to a sensor directly to avoid any traps going through the subscription level and video stream. This is the same for zone contract to zone.

The majority of my rationale are contained in the business rules section.

## Data Dictionary

*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.*

### Entities

|  |  |  |  |
| --- | --- | --- | --- |
| Entity Name | Description | Aliases | Occurrence |
| Subscriber | Someone who has, is about to, or is currently in a subscription. | User | Created by website or salesperson. Similar amount as sensors. |
| Contract | An agreement between a subscriber (individual or organisation) and Spaces that the subscriber will gain access to one of more of the following depending on the subscription fee:   * View a sensor’s video streams * Control the direction of the sensor * Gain exclusive rights to the video stream * Gain Exclusive rights to the data stream * Set up a zone with sensors | Contract | Created when a subscriber signs up or when a current subscriber wants another contract. Estimated at least as many contracts as subscribers. |
| Sensor Subscription | A standard or gold tier subscription from an individual to a 3d sensor. | Video Contract | Created when a subscriber signs a contract for a standard or gold level subscription. This will be the most popular type of contract. Estimated 98% of contracts |
| Zone Contract | A platinum or super platinum tier contract (generally with an organisation) to a zone. This involves setting up a zone. | Data Contract | Zone contracts are created when a business signs a contract to set up a zone with a platinum or super platinum contract. Estimated 2% of contracts. |
| Standard Subscription | An agreement between a subscriber and Spaces where the subscriber can view the video streams of a specified 3d sensor. | Standard, Standard Contract | Created when a subscriber signs a contract for a standard level subscription. This will be the most popular type of subscription. Estimated 90% of contracts. |
| Gold Subscription | An agreement between a subscriber and Spaces where the subscriber can view the video streams of a specified 3d sensor and change its direction. | Gold, Gold Contract | Created when a subscriber signs a contract for a gold level subscription. Estimated 8% of contracts. |
| Platinum Contract | An agreement between a subscriber and Spaces where the subscriber can view the video streams of a specified 3d sensor, change its direction and have exclusive rights to the data stream. A new zone and sensors can be set up. | Platinum, Platinum Subscription | Created when a business signs a contract to set up a zone with a platinum level contract. Estimated 1.9% of contracts. |
| Super Platinum Contract | An agreement between a subscriber and Spaces where the subscriber has exclusive rights to view the video streams of a specified 3d sensor, change its direction and have exclusive rights to the data stream. A new zone and sensors can be set up. | Super Platinum, Super Platinum Subscription | Created when a business signs a contract to set up a zone with a super platinum level contract. Estimated 0.1% of contracts. |
| Video Stream | The constant video feed coming from a sensor. | Video | Created by a sensor when it is live. Up to 100 video feeds per sensor. |
| Data Stream | The constant data being transmitted from a sensor. Includes 3d human imagery, shared audio, human voice | Data | Created by a sensor when it is live. One stream of data per data focus (imagery, audio, voice). |
| 3d Sensor | A 3d sensor device that can capture video, audio and estimate human positioning data. | Sensor | Created when a sensor is added to a zone. Estimated 10 sensors per zone. |
| Zone | A zone is a network of one or more sensors put in place for the same purpose and from the same contracting organisation. |  | A zone is created with each zone contract. There should be the same number of zones as zone contracts. |
| Parts | Mechanical and electronic components of the sensors that must be replaced over time. |  | Parts are created when a new part is required by a sensor. Estimated few hundred parts. |
| Supplier | A business that supplies parts for the sensors. |  | A supplier is created when a part is sourced from a different company or a new part from a new company is added. Estimated less than the number of parts. |
| Maintenance Record | Details of any maintenance that has occurred on a sensor. |  | Created when a maintainer performs a check-up or maintenance on a sensor. |
| Employee | An employee of Spaces©. | Staff | Created when a new employee joins the Spaces©. |
| Maintainer | An employee of Spaces who maintains and checks the sensors. |  | A maintainer is created when a new employee is assigned that role. Estimated 60% of employees. |
| Administrative Executive | An employee of Spaces who approves contracts, gives discounts and performs other administrative work. | Admin | An administrative executive is created when a new employee is assigned that role. Estimated 2% of staff. |
| Salesperson | An employee of Spaces who represents the company and sells subscriptions to potential customers. |  | A salesperson is created when a new employee is assigned that role. Estimate 28% of employees. |

### Relationships

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity Name | Cardinality | Participation | Relationship | Participation | Cardinality | Entity Name |
| Subscriber | One | Partial | Pays for | Total | Many | Contract |
| Sensor Subscription | One | Total | Subscribes to | Partial | One | Sensor |
| Zone Contract | One | Partial | Contracts | Total | One | Zone |
| Standard | Many | Partial | Views | Partial | Many | Video Stream |
| Gold | Many | Partial | Controls direction | Partial | Many | Video Stream |
| Platinum | One | Partial | Has exclusive rights | Partial | Many | Data |
| Super Platinum | One | Partial | Has exclusive rights | Partial | Many | Video stream |
| Sensor | One | Partial | Records | Total | Many | Video Stream |
| Sensor | One | Partial | Records | Total | Many | Data |
| Sensor | Many | Total | Belongs to | Partial | One | Zone |
| Sensor | Many | Total | Has | Partial | Many | Parts |
| Sensor | One | Partial | Has | Total | Many | Maintenance Record |
| Sensor | Many | Total | Maintains | Partial | One | Maintainer |
| Parts | Many | Total | Supplied by | Partial | Many | Supplier |
| Parts | Many | Partial | Replaced | Partial | Many | Maintenance Record |
| Maintainer | One | Partial | Records | Total | Many | Maintenance Record |
| Admin Executive | One | Partial | Approves | Total | Many | Contract |
| Salesperson | One | Partial | Sells | Partial | Many | Contract |

### Attributes

Note: the data types for the data being recorded by sensors (audio, imagery) is unknown in what formats will be used and how the data will be packaged so I have put those as “data file”.

\* Spaces© will likely have standard base prices for standard and gold subscriptions which would be the default.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Entity Name | Attributes | Description | Domain | Aliases | Composite | Derived | Nulls | Key? | Default Value |
| Subscriber | Name | Subscriber name | Char(50) |  | First name + last name |  |  |  |  |
| Email | Contact email | Char(50) |  |  |  |  | y |  |
| Password | Account access | Char(10) |  |  |  |  |  |  |
| Address | Subscriber location | Char(100) |  |  |  | n |  |  |
| Contract | Contract No | ID | Int |  |  |  |  | Y |  |
| Start Date | Contract start | Date |  |  |  |  |  |  |
| End Date | Contract end | Date |  |  |  | n |  |  |
| Discount % | Any discount given | Int |  |  |  |  |  | 0 |
| Monthly Price | Cost each month | Int | Cost |  |  |  |  | \* |
| Sensor | Sensor ID | ID | Int |  |  |  |  | Y |  |
| Longitude | Location | Char(10) |  |  |  |  |  |  |
| Latitude | Location | Char(10) |  |  |  |  |  |  |
| Time | Current time | Time |  |  |  |  |  |  |
| Zone | Zone ID | ID | Int |  |  |  |  | y |  |
| Maintenance Record | Datetime | Datetime for history | Datetime |  |  |  |  | Y |  |
| Maintenance Notes | Log of maintenance changes/info | Char(254) |  |  |  |  |  | “Check-up, no issues.” |
| Parts | Part Name | Unique name | Char(254) |  |  |  |  | Y |  |
| Cost | Price, NZD | Decimal | Price |  |  |  |  |  |
| Supplier | Supplier Name | Company name | Char(254) |  |  |  |  | Y |  |
| Address | Location | Char(50) |  | Street address + suburb + city + country |  |  |  |  |
| Email | Contact email | Char(50) |  |  |  |  |  |  |
| Phone | Contact number | Int |  |  |  | n |  |  |
| Employee | Name | Employee name | Char(50) |  | First name + last name |  |  |  |  |
| Email | Contact email | Char(50) |  |  |  |  | Y |  |
| Phone | Contact phone | Int |  |  |  |  |  |  |
| Video Stream | Video Feed | Video data | Data file | Video, Video data |  |  |  |  |  |
| Video No | ID | Int |  |  |  |  | Y |  |
| Zone Audio | Audio in the zone picked up by sensor | Data file |  |  |  |  |  |  |
| Data Stream | Data No | ID | Int |  |  |  |  | Y |  |
| Human Voice | Isolated human voice audio | Data file |  |  |  |  |  |  |
| Shared Audio | Shared audio from other sources | Data file |  |  |  |  |  |  |
| Skeletal Points | Estimated human skeletal point data | Data file |  |  |  |  |  |  |
| Texture | Human texture data | Data file |  |  |  |  |  |  |
| Body Shape | Human body shape data | Data file |  |  |  |  |  |  |

## Business Rules

*Discuss the assumptions made about the business rules and the reasons for the choices you made.*

**Subscriptions vs Contracts**

A contract is an agreement between Spaces© and a subscriber (individual or company) to provide specified access to their services, both subscribing to a sensor or hosting a zone. Contracts are the top-level agreement. Essential, contracts are subscriptions are subcategories of the main contract entity.

Within contracts there are two subcategories:

* Contracts refer to large agreements, often made with a company. These are Zone contracts (platinum or super platinum contracts) which involve purchasing sensors and setting up a zone. These are more likely to have discounts.
* Subscriptions usually refer to smaller contracts with individuals to gain access to video feed. These fall into the category of sensor subscription (standard and gold subscriptions).

While these do have different conations, they can be used interchangeably to some degree. It is best not to do so, however to reduce confusion.

“A subscription is a contract.” – Project Brief, page 2, Subscriptions and contracts

**Subscribers and Subscriptions**

I have decided to have subscribers and contracts as separate entities. This is because a subscriber could have previously had a contract but does not currently (but their account still exists), or the subscriber could have subscriptions to multiple sensors (see next assumption). A business may also have a zone contract to host a zone and sensor subscriptions simultaneously.

**Multiple Subscriptions**

I have assumed that since each subscription is only to a single sensor, then, subscribers must be able to have multiple subscriptions in order to access other sensors, both in and outside the zone. This allows for easier management of contracts to different sensors and zones, and allows for individual prices and payment dates.

**Subscription Generalisation and Specialisation**

Each contract is represented by a base contract entity in the database (generalisation), which will then be connected to the relevant contract type entity (sensor or zone). I have chosen to do this as while they are both contracts, sensor subscriptions and zone contracts function differently so it is clearer to separate them into categories. Under the contract type entity is the subscription/contract level. These contract levels entities are connected as a subscription will include its relevant level access plus any lower-level subscription access. E.g., a gold level subscription will also have access to the same access that a standard subscription has.

**Direction Control**

A gold level subscription allows subscribers to change the direction of the video stream. Spaces© has not specified how this would function in terms of having multiple gold watching a single video stream. I have decided to use a “first-in first served” based. What this means is that the first gold subscriber to start watching a video stream is grated the ability to control the direction. Any other golds subscribers watching at the same time will not be able to control the direction, as this could result in “fights” constantly changing the direction and disrupting an enjoyable video stream. When the gold subscriber in charge leaves the video stream, the gold subscriber who joined next will be granted the permission. If other gold subscribers are watching when one joins, they will gain access straight away, and if there are none at all, the direction cannot be changed. This could be thought about like a queue of gold subscribers, and whoever is at the top has control. This would eliminate any issues of conflict between gold subscribers.

**Sensor Video Streams**

Sensors can have any number of video streams, however this number is limited to 100 maximum. Having an unlimited amount could result in overloading the capacity of the sensor and the network connection to the servers and database.

**Considerations**

There were a couple of changes I considered but decided against.

* Subscriber and employee could be combined as a “person” superclass with employee and subscriber as subclasses, as there is some duplication between attributes. However, I think it is best to keep separate as they are quite different in how they interact with the rest of the database and from a business perspective (customer vs worker).
* Subscriber could include date of birth which could have the derived attribute age. This would allow Spaces© to have age restricted zones (perhaps for some concerts or various events). Spaces© has not mentioned any interest in this and I do not see any other reason to store subscribers’ age, so I have not included it.
* Subscriber could include payment details, but I think it is best for payments to go through an external payment system for better security.

**Audio**

Originally all audio was categorised as sensor data by Spaces©, which required a platinum subscription or higher. This implies that the video streams would have no audio. I think that this is a bad and unusual business decision, as subscribers will want to hear audio from the zone. I have decided to add “zone audio” as an attribute to the video stream entity. This will include all zone audio picked up by the sensor. Any other audio, such as shared audio or isolated human voice data will still be part of the data entity.

**IDs**

I decided to create a couple of ID attributes in the case of no appropriate candidate keys.

For example, a contract could have the same start and end dates as other contracts. Monthly price will be similar between lots of the contracts, especially for standard and gold subscriptions. Discount will also have many duplicate values between contracts. None of these are good candidate keys, so I have decided to create Contract No. This is the same for all other IDs created.

**Weak/Strong Entities**

I have included four weak entities:

1. Contract – By definition a contract is an agreement between the subscriber and Spaces©, so a contract cannot exist without a subscriber.
2. Video Stream – A video stream cannot exist without being recorded by a sensor.
3. Data - A data stream cannot exist without being recorded by a sensor.
4. Maintenance Record – A maintenance record cannot exist without a sensor, as the record is for maintenance on a sensor.

I considered having parts as a weak entity owned by a supplier, but since a part can be supplied by multiple suppliers, it is independent of supplier and can exist on its own, so it must be a strong entity.

The same situation happened with sensors and zones. Since sensors can be in multiple zones, they are independent and can’t be weak entities.

# Part Two – TSQL Queries

Note 1: all SQL queries are in a separate SQL file from GitHub: <https://github.com/Jayden-Htn/DAT601_TSQL_Practice.git>

Note 2: the queries script is set to use small business database. Please make sure that this name and the populated database have the same name.

## Basic Queries

**Query 1**

|  |  |
| --- | --- |
|  | Graphical user interface, text, application, table  Description automatically generated |

**Query 2**

|  |  |
| --- | --- |
|  |  |

**Query 3**

|  |  |
| --- | --- |
| Graphical user interface, text, website  Description automatically generated |  |

**Query 4**

|  |  |
| --- | --- |
|  |  |

**Query 5**

|  |  |
| --- | --- |
|  |  |

**Query 6**

|  |  |
| --- | --- |
|  |  |

**Query 7**

|  |  |
| --- | --- |
|  |  |

**Query 8**

|  |  |
| --- | --- |
|  |  |

**Query 9**

|  |  |
| --- | --- |
|  |  |

Note: this answer could be rounded to a specified number of decimal points using the ROUND() function.

**Query 10**

|  |  |
| --- | --- |
|  |  |

**Query 11**

|  |  |
| --- | --- |
|  |  |

**Query 12**

|  |  |
| --- | --- |
|  |  |

****

**Query 13**

|  |  |
| --- | --- |
|  |  |

**Query 14**

|  |  |
| --- | --- |
|  |  |

Note: change population script to have correct value (5-> 50 for Raggedy Ann).

## Sub Queries

**Query 15**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

**Query 16**

|  |  |
| --- | --- |
|  |  |

## Combined Query

**Query 17**

|  |  |
| --- | --- |
|  |  |

Note: this query could be done with a union, but a subquery approach feels much more logical to me.

## Views

**Query 18**

|  |  |
| --- | --- |
|  |  |

**Query 19**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# References

*Chen Notation*. (2014, August 2). Retrieved from Vertabelo: https://vertabelo.com/blog/chen-erd-notation/

*Data Modeling*. (Retrieved 2023). Retrieved from 1 Key Data: https://www.1keydata.com/datawarehousing/data-modeling-levels.html

Dybka, P. (2014, August 2). *Chen Notation*. Retrieved from Vertabelo: https://vertabelo.com/blog/chen-erd-notation/

Peinl, P. (2021, January). Teaching Database Design and the Importance of Integrity. .

Sherman, R. (2015). *Business Intelligence Guidebook: From Data Integration to Analytics.* Retrieved from Science Direct: https://www.sciencedirect.com/topics/computer-science/conceptual-data-model#:~:text=The%20conceptual%20data%20model%20represents,data%20in%20a%20business%20context.

Shin, K., Hwang, C., & Jung, H. (2017). *NoSQL Database Design Using UML Conceptual Data Model Based on Peter Chen’s Framework.* Retrieved from DocPlayer: https://docplayer.net/49346596-Nosql-database-design-using-uml-conceptual-data-model-based-on-peter-chen-s-framework.html

*What is a Logical Data Model?* (Retrieved 2023, 9 March). Retrieved from TIBCO: https://www.tibco.com/reference-center/what-is-a-logical-data-model

*What is data modeling?* (Retireved 2023, March 3). Retrieved from IBM: https://www.ibm.com/topics/data-modeling

Yue, K. (2018). *The Extended ER Diagram.* Retrieved from University of Houston-Clear Lake: http://dcm.uhcl.edu/yue/courses/itec3335/Fall2018/notes/model/Extended\_ERDiagram.html