DAT601 assessment

Ollie Moss

Contents

[Chapter 1 Part 1: Conceptual database design 2](#_Toc168386054)

[Overview 2](#_Toc168386055)

[Scenario 3](#_Toc168386056)

[Introduction to data modelling in information systems. 3](#_Toc168386057)

[Conceptual model 3](#_Toc168386058)

[Logical Model 3](#_Toc168386059)

[Physical implementation 3](#_Toc168386060)

[Database lifecycle 4](#_Toc168386061)

[Extended (aka Enhanced) Chen Entity Relationship Diagramming (ERD) 4](#_Toc168386062)

[Entities 4](#_Toc168386063)

[Attributes 4](#_Toc168386064)

[Attribute types: 5](#_Toc168386065)

[Keys 5](#_Toc168386066)

[Types of Keys: 5](#_Toc168386067)

[Relationships 5](#_Toc168386068)

[Relationship Attributes: 5](#_Toc168386069)

[Cardinalities: 6](#_Toc168386070)

[Participation Constraints: 6](#_Toc168386071)

[Specialization and Generalization: 6](#_Toc168386072)

[Relationships Degrees 6](#_Toc168386073)

[Conceptual ER model 6](#_Toc168386074)

[Data dictionary 7](#_Toc168386075)

[Business Rules Assumptions 7](#_Toc168386076)

[Data Integrity 7](#_Toc168386077)

[Data Security 7](#_Toc168386078)

[Data Governance 7](#_Toc168386079)

[Chapter 1 Part 2: TSQL 7](#_Toc168386080)

[Overview 8](#_Toc168386081)

[Adding Foreign Keys 8](#_Toc168386082)

[SQL Queries 8](#_Toc168386083)

[Sub Queries 10](#_Toc168386084)

[Combined Query 11](#_Toc168386085)

[Views 11](#_Toc168386086)

[Chapter 2: Logical Model 13](#_Toc168386087)

[Conceptual to Logical 13](#_Toc168386088)

[Updated Conceptual Model 14](#_Toc168386089)

[Mapping Rules 14](#_Toc168386090)

[Entities 14](#_Toc168386091)

[Attributes 14](#_Toc168386092)

[Relationships 14](#_Toc168386093)

[Superclass / subclass relationships 15](#_Toc168386094)

[Normalisation 15](#_Toc168386095)

[First Normal Form 15](#_Toc168386096)

[Second Normal Form 15](#_Toc168386097)

[Third Normal Form 15](#_Toc168386098)

[Boyce-Codd Normal Form 16](#_Toc168386099)

[Fourth Normal Form 16](#_Toc168386100)

[Logical ERD 16](#_Toc168386101)

[Rational & Relations 16](#_Toc168386102)

[Data Dictionary 17](#_Toc168386103)

[Entities 17](#_Toc168386104)

[Attributes 0](#_Toc168386105)

[NaLER Analysis 0](#_Toc168386106)

[Diagram Notation 0](#_Toc168386107)

[Entities 0](#_Toc168386108)

[Attributes 1](#_Toc168386109)

[Relationships 2](#_Toc168386110)

[Attribute Examples 4](#_Toc168386111)

[Relationship Examples 5](#_Toc168386112)

[Conclusion 6](#_Toc168386113)

[Chapter 3: Physical Implementation 7](#_Toc168386114)

[**References** 7](#_Toc168386115)

# Chapter 1 Part 1: Conceptual database design

# Overview

This part of the assignment is focused on the conceptual model to be used in the context of the scenario described below. This is the first stage of the project that will be expanded upon using the logical and physical model in assignment 2.

# Scenario

FlightStream is a “not for profit” organisation that provides remote video streaming and data sensing to government scientific services and commercial organisations and presence services to subscribers. For example, data required by government departments such as MFAT and the Department of Conservation. FlightStream deploys data sensing, collecting, and transmitting drones that can be managed by specific mobile devices. FlightStream brands their drones as “DataScoop” with an accompanying application for mobile devices. The “DataScoop”, facilitates data sensing and video streaming to be used for scientific and commercial purposes. The “DataScoop”, maintains its communication with FlightStream by satellite and/or ground based mobile cellular communication networks.

Introduction to data modelling in information systems.

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)

## Conceptual model

A model of the information used in an enterprise independent of all physical considerations.

The conceptual model involves forming a basis and a solid structure to work with in the later stages. Although using a higher-level model it is not necessarily less detailed. The stages to this include defining the scope, boundaries of the database, its users, and areas of use. This helps to further understand the databases use cases and purpose.

Furthering upon this gathering requirement and analysing use cases, scenarios, and roles to further understand how the database will function.

This information can be used to identify entities, in the system will help to provide a basis for developing a conceptual ER model. There are various ER models in use today, however in the context of this assignment the extended ER model will be used for this.

This is the outline for the conceptual stage of data modelling in information systems and how it develops a structured framework to work with in the later stages of development.

## Logical Model

Model of the information used in an enterprise based on specific data model independent of DBMS and physical systems.

Physical implementation:

Producing a description of the implementation of the database on secondary storage; it describes the storage structures and the access methods to achieve efficiency.

Involves translating the logical data model into a physical database schema that can be implemented on a specific database management system (DBMS).

Defines data types, indexes, constraints, and storage structures to optimize performance and meet the requirements of the target DBMS.

Considers factors such as scalability, availability, and security when designing the physical database schema.

## Database lifecycle

Database planning: how to realise the stages efficiently and effectively.

System definition: Scope, boundaries of app database, app users, and app areas

Requirements collection and analysis: use case, scenarios and roles

Conceptual design

Logical design

Physical design

Application design

Implementation, data conversion and loading, and testing and maintenance.

Prototyping, Agile sprints, Testing

Chens notation – used in conceptual modelling.

# 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)

CHENs Notation is a high-level data model used for the conceptual design of a database. This is still not less detailed than further stages of the data modelling stage. Making use of the ER diagram helps to specify the desired components of a database and the relationships within it. It provides flexibility so that it can be used in any environments that require information to be modelled.

## Entities

Entities represent real-world elements that will be involved in the system.

Describing entities in the extended CHENs Notation are as follows:

* Entities are drawn in a box.
* Entity type defines a collection of entities that have same attributes.
* Entity type is defined by its name and attribute.
* Entity type is named as a singular noun. E.g. person instead of people.
* Entity instance is a single occurrence of an entity type.
* Collection of entities is called an entity set. Contains all records of entities.

Weak entities

## Attributes

Attributes are properties that describe entities.

Describing attributes in the extended CHENs Notation are as follows:

* Enclosed by ovals connected to their entities with a single line.
* Value set or domain – represents all the different values an attribute can have (Data type)
* An attribute that’s underlined represents a primary key or candidate key.

## Attribute types:

* Simple attribute – represents a single data type (can’t be split into separate attributes) e.g. IRD number.
* Composite attribute – can be divided into smaller subparts. E.g. address can be split into street number, street name, etc.
* Single valued attribute - single value for a particular entity (single oval)
* Multi valued attribute – Set of values (doubled circled oval)
* Derived attribute – Attribute that can be derived from another attribute e.g. age can be derived from birthdate. (dotted oval)
* Stored attribute - Attribute from which other attributes are derived from e.g. birthdate is used to derive age.

Determining whether an attribute should be split into a composite attribute can be done by looking at use cases and functions for attributes and deciding whether they should be composite or simple attributes e.g. Should address be split into its components, if there is no need to refer to the individual components then there is no need to split it into simple attributes.

## Keys

Keys play a crucial role in defining the uniqueness of data within a database model. Keys help identify individual instances of entities and establish relationships between entities.

## Types of Keys:

* Primary Key (PK) - The primary key identifies each entity within a set of entities. Represented by underlining the attribute that is defined as the primary key.
* Composite Key - A composite key is made up of multiple primary keys. Represented by underlining multiple attributes within an entity. Composite keys are used when a single attribute cannot uniquely identify each entity instance, but a combination of attributes can.
* Foreign Key (FK) - A foreign key defines a relationship between two entities by referencing the primary key of another entity set. Represented by connecting an attribute in one entity to the primary key of another entity using a dashed line.
* Partial Key – A partial key serves as a unique identifier for a weak entity as it only uniquely identifies an entity in the context of a relationship and not the entity set as a whole. Represented using a dotted underline on the chosen attribute.

## Relationships

Relationships describe the connection between two entities. They are Represented with a diamond and lines connecting to the related entities.

## Relationship Attributes:

Describe properties or characteristics of relationships, represented by ovals connected to the relationship lines. Relationships have attributes that are related to the relationship itself, rather than to the entities connected in the relationship.

## Cardinalities:

Indicate the minimum and maximum number of occurrences of one entity that can be associated with another entity through a relationship. Represented using numbers: "1" for one occurrence, "N" for many occurrences.

## Participation Constraints:

Specifies whether the participation of an entity in a relationship is mandatory or optional.

Represented with a double line on desired side of the relationship for mandatory participation.

## Specialization and Generalization:

Used to represent inheritance between entity types.

Specialization represents the process of defining subtypes based on common attributes.

Generalization represents the process of defining a supertype that subtypes can be derived from.

Aggregation represents a relationship where one entity is made up of multiple entities.

## Relationships Degrees

Relationship degrees describe how many entities are participating in a given relationship.

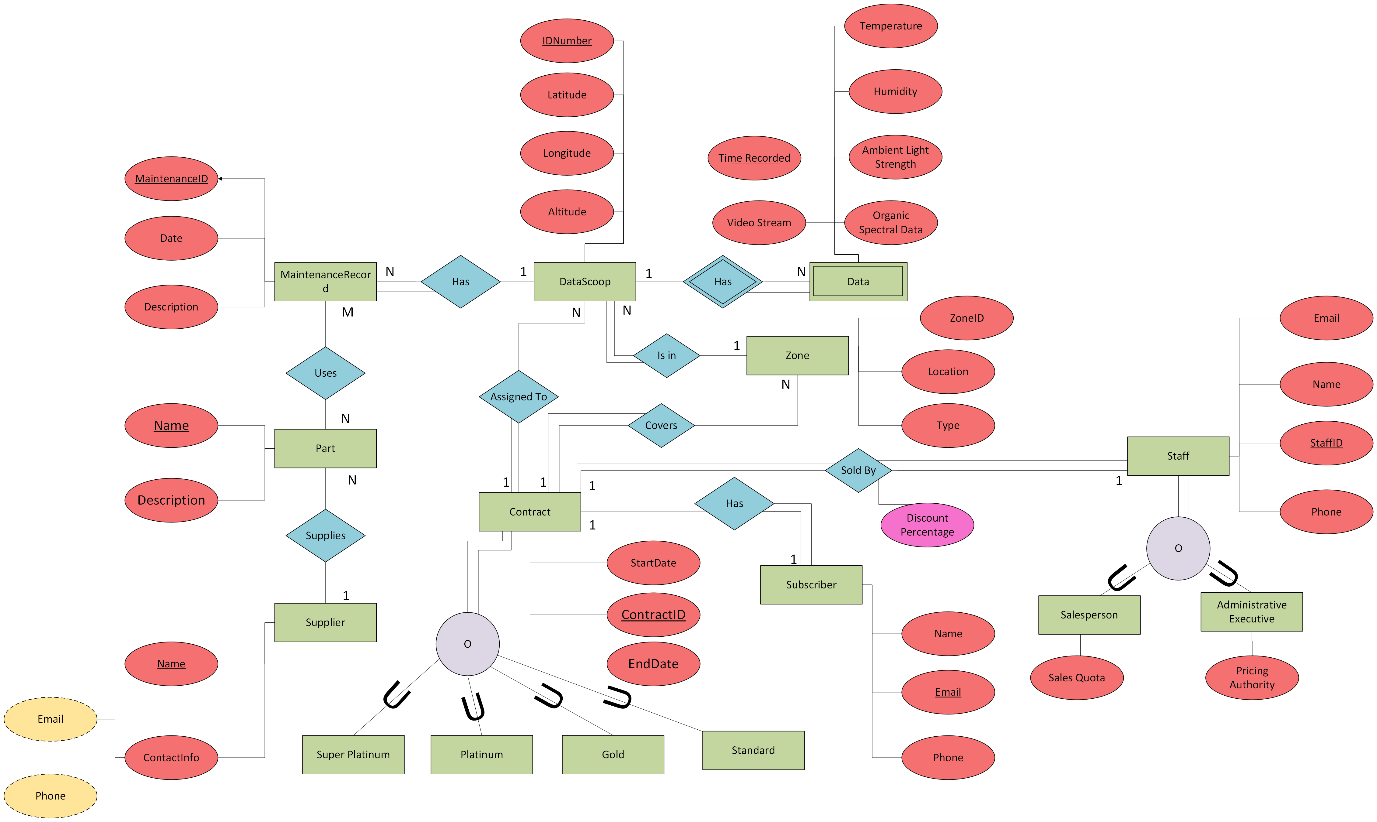
Unary Relationship: Relates an entity to itself.

Binary Relationship: Relates two different entities.

Ternary Relationship: Relates three entities.

Conceptual ER model

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)



Data dictionary

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)

Business Rules Assumptions

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)(5 marks)

## Data Integrity

Assumes that data recorded by DataScoops is accurate and reliable.

## Data Security

Assumes that effective security measures are put in place ensuring that privacy is upheld, and data breaches are minimised.

## Data Governance

Assumes that FlightStream adheres to regulations and standards in the context of data collection, and transmission.

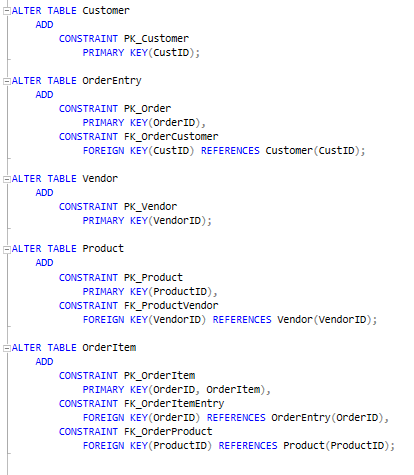
# Chapter 1 Part 2: TSQL

# Overview

Part 2 of this assignment focuses on SQL knowledge and practice. Using the provided database and data, using SQL Server tasks surrounding SQL queries and practice will be completed to demonstrate these skills. Including adding relationships, data and creating queries and views to retrieve data.

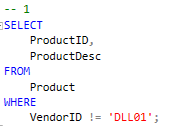
# Adding Foreign Keys

The database does not contain any Foreign Key constraints – add all the required Foreign Keys to your copy of the database using SQLServer SQL ALTER TABLE commands. Write an ALTER TABLE statement for each Foreign Key.



# SQL Queries

1. List all products not made by vendor DLL01.



1. List all the products with a price between $5.00 and $10.00.

A screen shot of a computer

Description automatically generated

1. List any products made by either vendor DLL01 or vendor BRS01 costing $10.00 or greater.

A computer screen shot of a computer

Description automatically generated with medium confidence

1. Return the average price of all the products in the Products table.

A close-up of a text

Description automatically generated

1. Return the total number of customers in the Customers table.

A close up of a number

Description automatically generated

1. Return the number of customers in the Customers table with an e-mail address.

A close up of a number

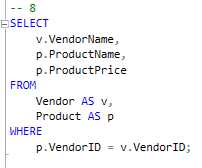
Description automatically generated

1. Return the number of product types, minimum, maximum and average product price from the products table.

A screenshot of a computer code

Description automatically generated

1. Return the vendor name, product price and product name from the vendors and products tables.



1. Return the product name, vendor name, product price and quantity for each item in order number 20007.

A screenshot of a computer

Description automatically generated

## Sub Queries

1. Create a list of all the customers (customer name and customer contact) who ordered item RGAN01.

A screenshot of a computer

Description automatically generated

1. Display the total number of orders placed by every customer in the Customers table, as well as the city the customer is in.

A screenshot of a computer program

Description automatically generated

## Combined Query

1. Create a report on all the customers in Nelson and Wellington. You also should include all Fun4All locations, regardless of city. The resulting customers should be in alphabetical order of customer name then customer contact.

A screenshot of a computer program

Description automatically generated

# Views

1. Create a view called vProductCustomer which joins the Customer, Order and OrderItem tables to return a list of all customers who have ordered any product. Now retrieve from that view a list of customers who ordered product RGAN01.

A screenshot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

1. Add a customer to the database: CustID = 1000000006 CustName = The Toy Emporium CustPhone = 09-546-8552

Using a view to format mailing list data: - First create a query that will display the customer name and then the address in the following format: CustName Customer address City/town, Phone number

Next turn this query into a view called vCustomerMailingLabel

Display all the “entries” in vCustomerMailingLabel

Try defining the customer mailing label view so that it filters out any incomplete addresses as these cannot be used for mailing labels.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

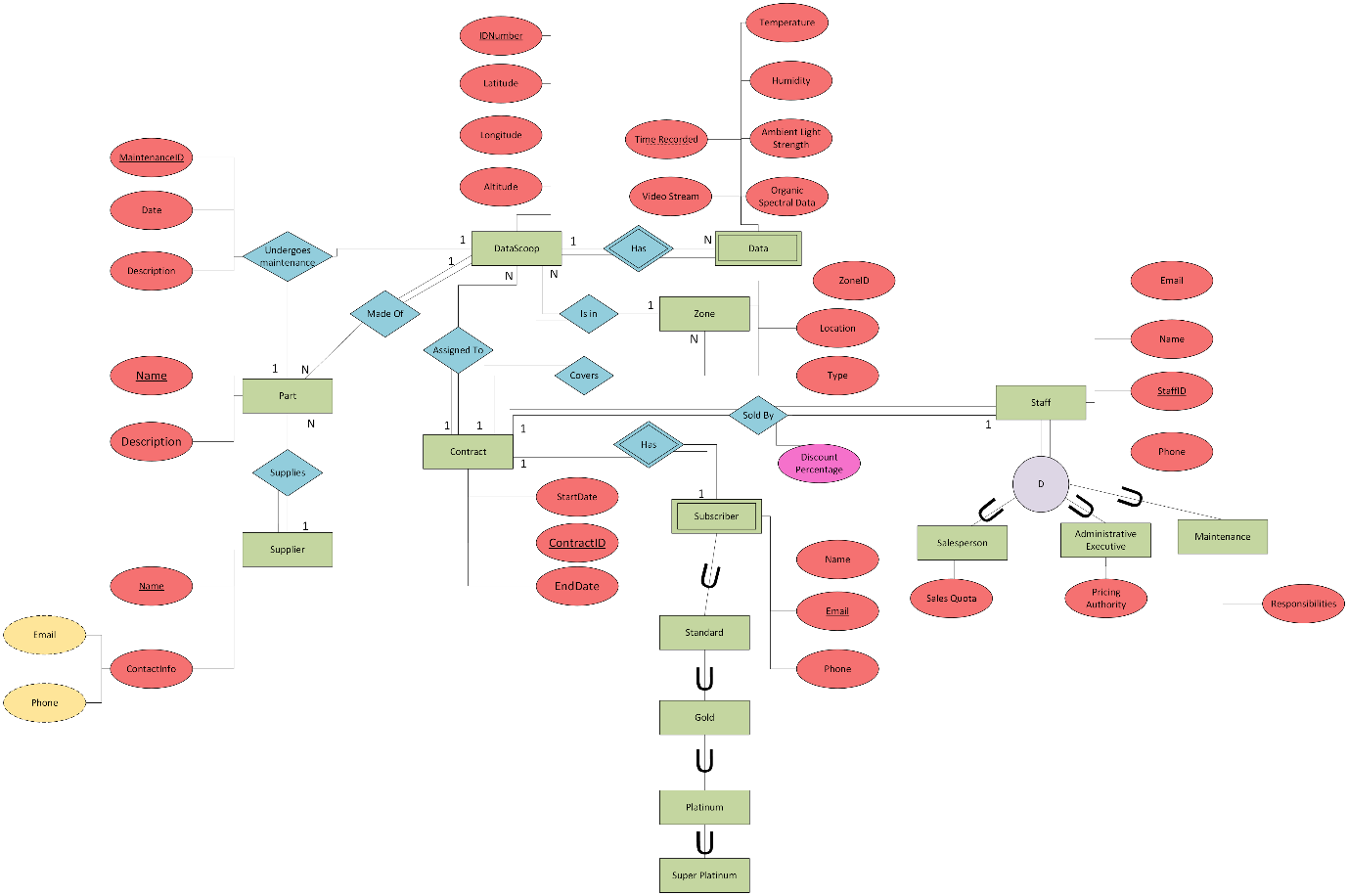
# Chapter 2: Logical Model

Model of the information used in an enterprise based on specific data model independent of DBMS and physical systems.

# Conceptual to Logical

This chapter contains the transformation between the conceptual design to the logical ERD model. This involves a series of steps: mapping rules, normalisation, data dictionary, and NaLER analysis to produce a design with no errors, or data anomalies.

## Updated Conceptual Model



## Mapping Rules

Mapping rules are a method to transform the conceptual ER model to the logical ERD model. This involves a series of rules to transform entities, attributes and relationships into relations, columns, and keys.

### Entities

Each entity becomes a relation.

### Attributes

* Composite attributes become separate attributes.
* Simple attributes remain the same.
* The unique identifier may become the primary key, dependent on whether it is suitable in practical use.

### Relationships

* One-to-one – The primary key of one relation is placed as in the other relation as a foreign key. The relationship does not require the foreign key to be placed on a specific side, focusing on the practical application of for the placement of the foreign key is suitable in this case.
* One to many – The primary key of the “one” side of the relationship is placed as the foreign key in the “many” side of the relationship.
* Many-to-many – Many to many relationships must be described using a relation table, this contains the primary keys for all related entities which make up the primary key of the relation.
* Relationships with attributes do not exist in the logical mode, to represent them they must be added as an attribute in a relational table used in a many-to-many relationship.

### Superclass / subclass relationships

There are 3 main solutions to generalization and specialization in the logical model, each with their own advantages and disadvantages.

* Solution A: One relation for the superclass with all common attributes, and a relation for each subclass with specialized attributes to the subclass
* Solution B: One relation for each subclass with both common and specialized attributes related to both the subclass and the superclass.
* Solution C: One relation with all common attributes and all specialized attributes, requiring null values for specialized attributes.

In most cases the solution A will be most suitable due to its more organized structure, however this may end up requiring more physical storage due to storing data in at least three table.

# Normalisation

Normalisation in database management is the process of organizing data into tables and defining relationships between them to minimize data redundancy (Microsoft, 2023). This in turn reduces chances of error during deletion, insertion and the updating of data (GeeksForGeeks, 2024). Normalisation follows guidelines set in the form of normal forms. These range from 1NF-5NF with the inclusion of BCNF (Boyce-Codd) in between 3NF and 4NF. The normalisation process for each form is as follows:

NOTE: A determinant is an attribute that determines other attributes within a relation. Otherwise known as a functional dependency.

## First Normal Form

First Normal Form is the first stage of database normalisation. In the first normal form each intersection between a column and a row must have only one value (McAleer, n.d.). This helps to build a basis for late normalizations to work with.

## Second Normal Form

Second Normal Form requires the database to already be in 1NF. 2NF involves removing all partial dependencies and separating them out into a different relation (McAleer, n.d.). Partial dependencies are where an attribute depends on part of the primary key. A new table must be made using the determinants of the dependent attribute as the primary key, allowing it to be removed from the original table.

## Third Normal Form

Third Normal Form requires the database to already be in 3NF. 3NF involves removing all transitive dependencies and creating a separate relation for them (McAleer, n.d.). Transitive dependencies are where one (non-prime) attribute depends on another (non-prime) attribute. This involves creating a table where the determinant attribute/s is the primary key.

In most cases 3NF is where the normalization process is sufficient and the database’s functionality will not change with further normalization, but it may not be as efficient or effective as possible.

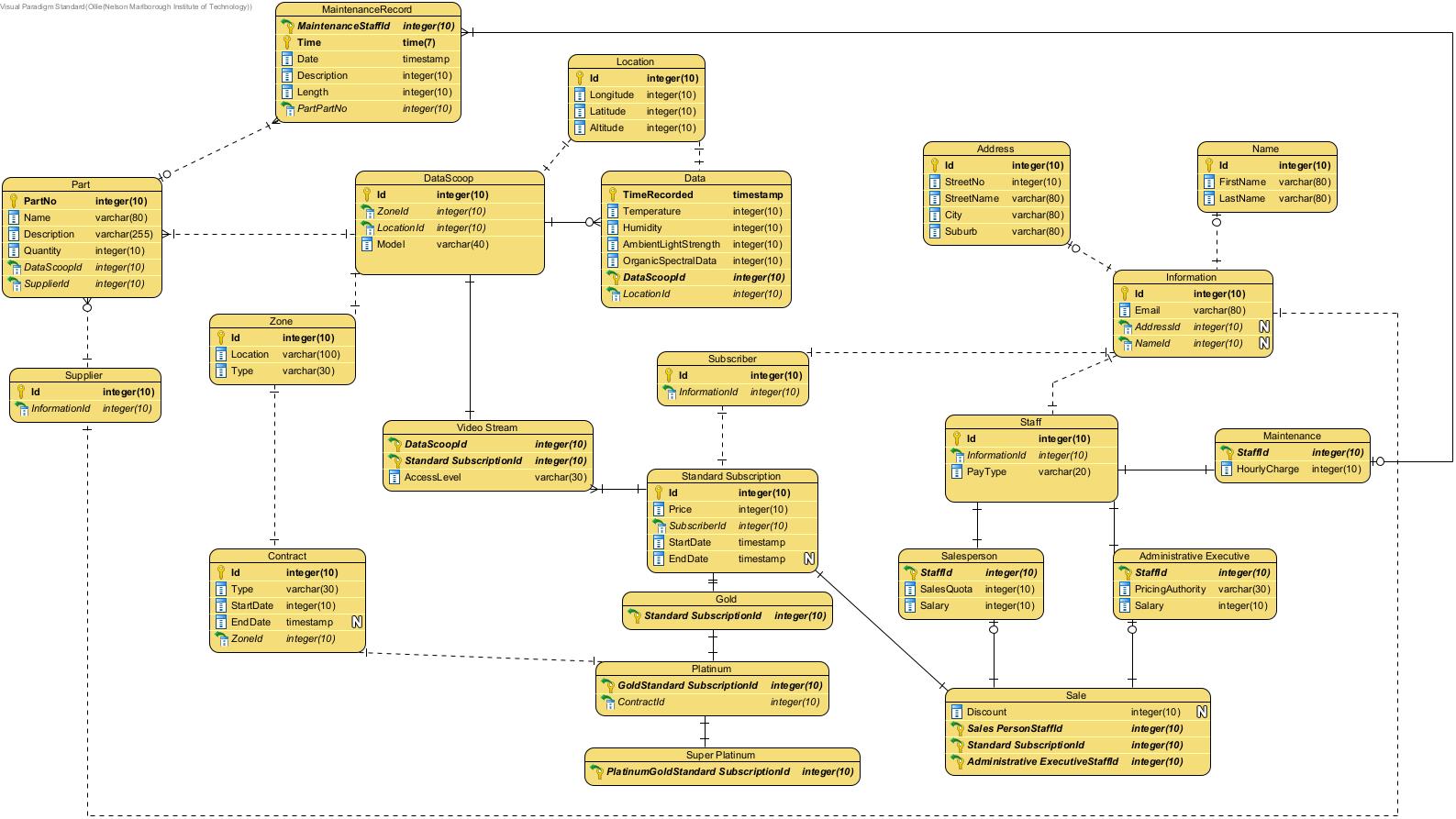
### Boyce-Codd Normal Form

Boyce-Codd Normal Form requires the database to be normalized to 3NF. BCNF involves ensuring that all determinants are candidate keys. Normally, a database normalized to 3NF will already be in BCNF, but this is something to consider in normalization but is not required to meet a functional database.

## Fourth Normal Form

Fourth Normal Form requires the database already be in BCNF. 4NF involves ensuring that there are no multi-valued dependencies in the database. A multi-valued dependency is where in a ternary or higher relationship one attribute is related to every other attribute and contains independent sets for each attribute. This involves creating new tables for each dependent attribute in a new table along with the determinant as the primary key.

# Logical ERD



# Rational & Relations

I began the process of the conceptual to logical model by recreating my conceptual design in Crows foot notation following the mapping rules discussed above.

Through this I found some larger issues within the conceptual design surrounding the understanding of the organization and the relationship between customers, their subscriptions, contracts and their access to data, video streams and control of the DataScoops. To remedy these concerns, I made the following changes:

A big change made to the logical model is the difference between a subscriber, subscription and a contract. A subscriber is the customer who is subscribed to FlightStream’s services, the subscription is the subscription level the subscriber is currently subscribed to, and a contract is reserved for platinum and super platinum subscribers and determine what zone they manage and their contract type either “data”, or “data and videostream”.

A new entity was created in the form of a video stream in which subscribers can now view/control based on the subscription level.

I feel these changes help the database more closely reflect the structure of FlightStream and how their services function.

I then selected the correct data types for each attribute as I missed the timestamp datatype for date attributes and chose appropriate sizes for varchar attributes (80 for names, 255 for descriptions)

Normalizing the database involved following the standard normalization process up to 3NF. Some further consideration to consider are BCNF and 4NF. For example, the Sale relation is a ternary relationship, and multi-valued dependency which could be normalized further.

# Data Dictionary

## Entities

|  |  |  |  |
| --- | --- | --- | --- |
| **Relation Name** | **Start Volume** No. of rows loaded at the beginning | **Growth** e.g. no growth / 10% per year | **Comments** |
| DataScoop | 50,000 | 10% |  |
| Data | 2,628,000 | 10% |  |
| Information | 505,000 | 9.1% |  |
| Name | 505,000 | 9.1% |  |
| Address | 505,000 | 9.1% |  |
| Staff | 5,000 | 5% |  |
| Maintenance | 1,000 | 10% |  |
| Administrative Executive | 1,000 | 10% |  |
| Salesperson | 3,000 | 10% |  |
| Sale | 500,000 | 10% |  |
| Standard Subscription | 500,000 | 10% |  |
| Gold | 5,000 | 5% |  |
| Platinum | 2,500 | 4% |  |
| Super Platinum | 100 | 5% |  |
| Contract | 2,000 | 5.25% |  |
| Video Stream | 50,000 | 10% |  |
| Zone | 100 | 0% |  |
| Part | 1,000,000 | 0%+/-0.1% |  |
| Supplier | 20 | 0% |  |
| Location | 2,678,000 | 0% |  |
| MaintenanceRecord | 1,000,000 | 0% |  |
| Subscriber | 500,000 | 10% |  |

## Attributes

| **Relation Name** | **Attribute** | **Description** | **Data type** | **Length** | **Value range** | **Validation Rules** | **Default Value** | **Nulls** | **Key?** | **References Entity** | **Integrity Constraints** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DataScoop | Id | Unique Id | Integer | 10 |  |  |  | No | PK |  |  |
| ZoneId | References the related record in the zone table | Integer | 10 |  |  |  | No | FK |  |  |
| LocationId | References the relatedrecord in the location table | Integer | 10 |  |  |  | No | FK |  |  |
| Model | Refers to specific model of the datascoop (drone type) | varchar | 40 |  |  |  |  |  |  |  |
| Data | TimeRecorded | The UTC time that the data was recorded | timestamp | - |  |  |  | No | PK |  |  |
| Temperature | The temperature recorded by a Datascoop | Integer | 10 |  |  |  | No |  |  |  |
| Humidity | Humidity information recorded by a DataScoop | Integer | 10 |  |  |  | No |  |  |  |
| AmbientLightStrength | Ambient light strength recorded by a DataScoop | Integer | 10 |  |  |  | No |  |  |  |
| OrganicSpectralData | Organic spectral data from one litre of air information recorded by a DataScoop | Integer | 10 |  |  |  | No |  |  |  |
| DataScoopId | The Id of the DataScoop that captured the data | Integer | 10 |  |  |  | No | PK, FK |  |  |
| LocationId | Id of the location at which the data was recorded | Integer | 10 |  |  |  | No | FK |  |  |
| Information | Id | Unique Id | Integer | 10 |  |  |  | No | PK |  |  |
| Email | The email address | varchar | 80 |  |  |  | No |  |  |  |
| AddressId | Id of the related address record | Integer | 10 |  |  |  | Yes | FK |  |  |
| NameId | Id of the related Name record | Integer | 10 |  |  |  | Yes | FK |  |  |
| Name | Id | Unique Id | Integer | 10 |  |  |  | No | PK |  |  |
| FirstName | The first name | varchar | 80 |  |  |  | No |  |  |  |
| LastName | The last name | varchar | 80 |  |  |  | No |  |  |  |
| Address | Id | Unique Id | Integer | 10 |  |  |  | No | PK |  |  |
| StreetNo | A street number of the address | Integer | 10 |  |  |  | No |  |  |  |
| StreetName | A street name of the address | varchar | 80 |  |  |  | No |  |  |  |
| City | The city in which the address is located | varchar | 80 |  |  |  | No |  |  |  |
| Suburb | The suburb in which the address is located | varchar | 80 |  |  |  | No |  |  |  |
| Staff | Id | Unique Id | Integer | 10 |  |  |  | No | PK |  |  |
| InformationId | Id of the related information record | Integer | 10 |  |  |  | No | FK |  |  |
| Maintenance | StaffId | Unique Id | Integer | 10 |  |  |  | No | PK, FK |  |  |
| HourlyCharge | The hourly charge of the maintenance person | Integer | 10 |  |  |  | No |  |  |  |
| Administrative Executive | StaffId | Unique Id | Integer | 10 |  |  |  | No | PK, FK |  |  |
| PricingAuthority | The pricing authority of the staff member | varchar | 30 |  |  |  | No |  |  |  |
| Salary | The salary of the staff member | Integer | 10 |  |  |  | No |  |  |  |
| Salesperson | StaffId | Unique Id | Integer | 10 |  |  |  | No | PK, FK |  |  |
| SalesQuota | The quota of the salesperson | Integer | 10 |  |  |  | No |  |  |  |
| Salary | The salary of the salesperson | Integer | 10 |  |  |  | No |  |  |  |
| Sale | Discount | Discount percentage on the sale | Integer | 10 |  |  |  | Yes |  |  |  |
| SalespersonStaffId | The id of the salesperson who sold the subscription | Integer | 10 |  |  |  | No | PK, FK |  |  |
| StandardSubscriptionId | The id of the subscription that was sold | Integer | 10 |  |  |  | No | PK, FK |  |  |
| AdministrativeExecutiveId | The id of the administrative executive who approved and may have applied a discount to the sale | Integer | 10 |  |  |  | No | PK, FK |  |  |
| Standard Subscription | Id | Unique Id | Integer | 10 |  |  |  | No | PK |  |  |
| Price | The price of the subscription | Integer | 10 |  |  |  | No |  |  |  |
| StartDate | The date a which the subscription started | timestamp | - |  |  |  | No |  |  |  |
| EndDate | The expected date at which the subscription will expire | timestamp | - |  |  |  | Yes |  |  |  |
| SubscriberId | The id of the subscriber who owns the subscription | Integer | 10 |  |  |  | No | FK |  |  |
| Gold | StandardSubscriptionId | The id of standard subscription that the gold subscription inherits from | Integer | 10 |  |  |  | No | PK, FK |  |  |
| Platinum | GoldStandardSubscriptionId | The id of gold subscription that the platinum subscription inherits from | Integer | 10 |  |  |  | No | PK, FK |  |  |
| ContractId | The contract that this subscription is assigned to | Integer | 10 |  |  |  | No |  |  |  |
| Super Platinum | PlatinumGoldStandSubscriptionId | The platinum subscription that the super platinum subscription inherits from | Integer | 10 |  |  |  | No | PK, FK |  |  |
| Contract | Id | Unique Id | Integer | 10 |  |  |  | No | PK |  |  |
| Type | The type of contract this is, either “data” or “data and video” | varchar | 30 |  |  |  | No |  |  |  |
| StartDate | Start date of the contract | timestamp | - |  |  |  | No |  |  |  |
| EndDate | The expected end date of the contract | timestamp | - |  |  |  | Yes |  |  |  |
| ZoneId | The id of the zone that this contract covers | Integer | 10 |  |  |  | No | FK |  |  |
| Video Stream | DataScoopId | Id of the datascoop that the video stream is related to | Integer | 10 |  |  |  | No | PK, FK |  |  |
| StandardSubscriptionId | The id of the subscription that can view/control the video stream | Integer | 10 |  |  |  | No | PK, FK |  |  |
| AccessLevel | The access level of the video access (control or view) | varchar | 30 |  |  |  | No |  |  |  |
| Zone | Id | Unique Id | Integer | 10 |  |  |  | No | PK |  |  |
| Location | The location that the zone covers | varchar | 100 |  |  |  | No |  |  |  |
| Type | The type of geographic region that the zone covers | varchar | 30 |  |  |  | No |  |  |  |
| Part | PartNo | Unique Number | Integer | 10 |  |  |  | No | PK |  |  |
| Name | The name of the part | varchar | 80 |  |  |  | No |  |  |  |
| Description | A brief description of the part | varchar | 255 |  |  |  | No |  |  |  |
| Quantity | The number of these parts used in the related datascoop | Integer | 10 |  |  |  | No |  |  |  |
| DataScoopId | The id of which these part/s are used in | Integer | 10 |  |  |  | No | FK |  |  |
| SupplierId | The id of the supplier that supplied the part/s | Integer | 10 |  |  |  | No | FK |  |  |
| Supplier | Id | Unique Id | Integer | 10 |  |  |  | No | PK |  |  |
| InformationId | Id of the information record related to the supplier | Integer | 10 |  |  |  | No | FK |  |  |
| Location | Id | Unique Id | Integer | 10 |  |  |  | No | PK |  |  |
| Longitude | The longitude of the location | Integer | 10 |  |  |  | No |  |  |  |
| Latitude | The latitude of the location | Integer | 10 |  |  |  | No |  |  |  |
| Altitude | The altitude of the location | Integer | 10 |  |  |  | No |  |  |  |
| MaintenanceRecord | MaintenanceStaffId | The id of the maintenance person who performed the maintenance | Integer | 10 |  |  |  | No | PK, FK |  |  |
| Time | The time at which the maintenance was performed | timestamp | - |  |  |  | No | PK |  |  |
| Date | The date at which the maintenance was performed | timestamp | - |  |  |  | No |  |  |  |
| Description | A brief description of the maintenance performed | varchar | 255 |  |  |  | No |  |  |  |
| Length | Amount of time the maintenance took to complete in hours | Integer | 10 |  |  |  | No |  |  |  |
| PartPartNo | The part number corresponding to the part the underwent maintenance. | Integer | 10 |  |  |  | No | FK |  |  |
| Subscriber | Id | Unique Id | Integer | 10 |  |  |  | No | PK |  |  |
| InformationId | The id of the information record related to the subscriber | Integer | 10 |  |  |  | No | FK |  |  |

# NaLER Analysis

NaLER is a natural language method for entity relations. It helps to validate the logic behind the logical ERD. This involves creating expressions of how the data will be stored helping to easily understand the how it will be used and stored. NaLER involves creating sentences for each attribute, entity, and relationship using primary and foreign keys.

## Diagram Notation

I have used Crows Foot notation to describe the logical ERD using visual paradigm, notation is automatically done, relations are shown in boxes with sections for the name, and attributes, using appropriate notation for primary and foreign keys.

Cardinality is shown through the same crow’s foot notation as depicted in the diagram below:

A screenshot of a computer

Description automatically generated

After checking my diagram there are some areas of improvement for my syntax and will be changed at a later iteration.

## Entities

S1: Each DataScoop is uniquely identified by Id

S2: Each Address is uniquely identified by Id

S3: Each Name is uniquely identified by Id

S4: Each Information is uniquely identified by Id

S5: Each Staff is uniquely identified by Id

S6: Each Salesperson is uniquely identified by StaffId

S7: Each Administrative Executive is uniquely identified by StaffId

S8: Each Maintenance is uniquely identified by StaffId

S9: Each Sale is uniquely identified by SalesPersonStaffId, StandardSubscriptionId, AdministrativeExecutiveId

S10: Each StandardSubscription is uniquely identified by Id

S11: Each Subscriber is uniquely identified by Id

S12: Each Gold is uniquely identified by StandardSubscriptionId

S13: Each Platinum is uniquely identified by GoldStandardSubscriptionId

S14: Each SuperPlatinum is uniquely identified by PlatinumGoldStandardSubscriptionId

S15: Each Contract is uniquely identified by Id

S16: Each Zone is uniquely identified by Id

S17: Each VideoStream is uniquely identified by DataScoopId, StandardSubscriptionId

S18: Each Data is uniquely identified by TimeRecorded, DataScoopId

S19: Each Location is uniquely identified by Id

S20: Each MaintenanceRecord is uniquely identified by MaintenanceStaffId, Time

S21: Each Part is uniquely identified by PartNo

S22: Each Supplier is uniquely identified by Id

## Attributes

S23: One DataScoop identified by Id must have one Model

S24: One Address identified by Id must have one StreetNo

S25: One Address identified by Id must have one StreetName

S26: One Address identified by Id must have one City

S27: One Address identified by Id must have one Suburb

S28: One Name identified by Id must have one FirstName

S29: One Name identified by Id must have one LastName

S30: One Information identified by Id must have one Email

S31: One Staff identified by Id must have one PayType

S32: One Salesperson identified by StaffId must have one SalesQuota

S33: One Salesperson identified by StaffId must have one Salary

S34: One Administrative Executive identified by StaffId must have one PricingAuthority

S35: One Administrative Executive identified by StaffId must have one Salary

S36: One Maintenance identified by StaffId must have one HourlyCharge

S37: One Sale identified by SalesPersonStaffId, StandardSubscriptionId, AdministrativeExecutiveId must have one Discount

S38: One StandardSubscription identified by Id must have one Price

S39: One StandardSubscription identified by Id must have one StartDate

S40: One StandardSubscription identified by Id must have one EndDate

S41: One Subscriber identified by Id must have one

S42: One Gold identified by StandardSubscriptionId must have one

S43: One Platinum identified by GoldStandardSubscriptionId must have one ContractId

S44: One SuperPlatinum identified by PlatinumGoldStandardSubscriptionId must have one

S45: One Contract identified by Id must have one Type

S46: One Contract identified by Id must have one StartDate

S47: One Contract identified by Id must have one EndDate

S48: One Zone identified by Id must have one Location

S49: One Zone identified by Id must have one Type

S50: One VideoStream identified by DataScoopId, StandardSubscriptionId must have one AccessLevel

S51: One Data identified by TimeRecorded, DataScoopId must have one Temperature

S52: One Data identified by TimeRecorded, DataScoopId must have one Humidity

S53: One Data identified by TimeRecorded, DataScoopId must have one AmbientLightStrength

S54: One Data identified by TimeRecorded, DataScoopId must have one OrganicSpecrtralData

S55: One Location identified by Id must have one Longitude

S56: One Location identified by Id must have one Latitude

S57: One Location identified by Id must have one Altitude

S58: One MaintenanceRecord identified by MaintenanceStaffId, Time must have one Date

S59: One MaintenanceRecord identified by MaintenanceStaffId, Time must have one Description

S60: One MaintenanceRecord identified by MaintenanceStaffId, Time must have one Length

S61: One Part identified by PartNo must have one Name

S62: One Part identified by PartNo must have one Description

S63: One Part identified by PartNo must have one Qunatity

S64: One Supplier identified by Id must have one

## Relationships

S65: DataScoop identified by Id is made up of many Part indentified by DataScoopId

S66: Maintenance identified by StaffId may carry out many MaintenanceRecord indentified by MaintenanceStaffId

S67: Address identified by Id may contain one Information indentified by AddressId

S68: Name identified by Id may contain one Information indentified by NameId

S69: Location identified by Id will locate one DataScoop indentified by LocationId

S70: Zone identified by Id contains many DataScoop indentified by ZoneId

S71: Contract identified by Id is assigned to one Platinum indentified by ContractId

S72: Zone identified by Id is assigned to one Contract indentified by ZoneId

S73: DataScoop identified by Id is streaming one VideoStream indentified by DataScoopId

S74: StandardSubscription identified by Id is assigned to many VideoStream indentified by StandardSubscriptionId

S75: DataScoop identified by Id records many Data indentified by DataScoopId

S76: Location identified by Id is stored in one Data indentified by Data

S77: Supplier identified by Id supplies many Part indentified by SupplierId

S78: Part identified by PartNo is assigned to many MaintenanceRecord indentified by PartPartNo

S79: Subscriber identified by Id is assigned to one StandardSubscription indentified by SubscriberId

S80: StandardSubscription identified by Id is assigned to one Gold indentified by StandardSubscriberId

S81: Gold identified by StandardSubscriptionId is assigned to one Platinum indentified by GoldStandardSubscriptionId

S82: Platinum identified by GoldStandardSubscriptionId is assigned to one SuperPlatinum indentified by PlatinumGoldStandardSubscriptionId

S83: Information identified by Id is related to one Staff indentified by InformationId

S84: Information identified by Id is related to one Supplier indentified by InformationId

S85: Information identified by Id is related to one Subscriber indentified by InformationId

S86: Staff identified by Id derives one Salesperson indentified by StaffId

S87: Staff identified by Id derives one Maintenance indentified by StaffId

S88: Staff identified by Id derives one Administrative Executive indentified by StaffId

S89: Salesperson identified by StaffId may sell one Sale indentified by SalespersonStaffId

S90: Administrative Executive identified by StaffId may discount one Sale indentified by AdministrativeExecutiveStaffId

S91: StandardSubscription identified by Id is sold through one Sale indentified by StandardSubscriptionId

## Attribute Examples

S92: One DataScoop identified by Id '1' must have one Model 'X4-9800'

S93: One Address identified by Id '1' must have one StreetNo '43'

S94: One Address identified by Id '1' must have one StreetName 'Scotia Street'

S95: One Address identified by Id '1' must have one City 'Nelson'

S96: One Address identified by Id '1' must have one Suburb 'Whakatu'

S97: One Name identified by Id '1' must have one FirstName 'Ollie'

S98: One Name identified by Id '1' must have one LastName 'Moss'

S99: One Information identified by Id '1' must have one Email 'olliemoss321@gmail.com'

S100: One Staff identified by Id '1' must have one PayType 'Hourly'

S101: One Salesperson identified by StaffId 'Id '1' must have one SalesQuota '4000'

S102: One Salesperson identified by StaffId 'Id '1' must have one Salary '72000'

S103: One Administrative Executive identified by StaffId '2' must have one PricingAuthority 'Discount'

S104: One Administrative Executive identified by StaffId '2' must have one Salary '112000'

S105: One Maintenance identified by StaffId '3' must have one HourlyCharge '31.50'

S106: One Sale identified by SalesPersonStaffId '1', StandardSubscriptionId '1', AdministrativeExecutiveId '2' must have one Discount '3'

S107: One StandardSubscription identified by Id '1' must have one Price '85.99'

S108: One StandardSubscription identified by Id '1' must have one StartDate '19/01/2021'

S109: One StandardSubscription identified by Id '1' must have one EndDate 'null'

S110: One Subscriber identified by Id '1' must have one

S111: One Gold identified by StandardSubscriptionId '1' must have one

S112: One Platinum identified by GoldStandardSubscriptionId '1' must have one ContractId '1'

S113: One SuperPlatinum identified by PlatinumGoldStandardSubscriptionId '1' must have one

S114: One Contract identified by Id '1' must have one Type 'Data'

S115: One Contract identified by Id '1' must have one StartDate '12/03/2019'

S116: One Contract identified by Id '1' must have one EndDate null

S117: One Zone identified by Id '1' must have one Location 'Tasman'

S118: One Zone identified by Id '1' must have one Type 'Plains'

S119: One VideoStream identified by 1, 1 must have one AccessLevel 'Control'

S120: One Data identified by TimeRecorded '11:03:49', DataScoopId '1' must have one Temperature '32.21'

S121: One Data identified by TimeRecorded '11:03:49', DataScoopId '1' must have one Humidity 53

S122: One Data identified by TimeRecorded '11:03:49', DataScoopId '1' must have one AmbientLightStrength 73260

S123: One Data identified by TimeRecorded '11:03:49', DataScoopId '1' must have one OrganicSpecrtralData unsure

S124: One Location identified by Id '1' must have one Longitude '41°12'11.8'

S125: One Location identified by Id '1' must have one Latitude '173°03'10.6'

S126: One Location identified by Id '1' must have one Altitude '92'

S127: One MaintenanceRecord identified by MaintenanceStaffId '3', Time '15:32:15' must have one Date '28/08/2022'

S128: One MaintenanceRecord identified by MaintenanceStaffId '3', Time '15:32:15' must have one Description 'Part broken needed replacement.'

S129: One MaintenanceRecord identified by MaintenanceStaffId '3', Time '15:32:15' must have one Length '2:34:48'

S130: One Part identified by PartNo '1' must have one Name '3D Camera'

S131: One Part identified by PartNo '1' must have one Description 'Provides 3D sterographic video'

S132: One Part identified by PartNo '1' must have one Quantity '1'

S133: One Supplier identified by Id '1' must have one

## Relationship Examples

S134: DataScoop identified by Id '1' is made up of many Part indentified by DataScoopId '1'

S135: Maintenance identified by StaffId '3' may carry out many MaintenanceRecord indentified by MaintenanceStaffId '3'

S136: Address identified by Id '1' may contain one Information indentified by AddressId '1'

S137: Name identified by Id '1' may contain one Information indentified by NameId '1'

S138: Location identified by Id '1' will locate one DataScoop indentified by LocationId '1'

S139: Zone identified by Id '1' contains many DataScoop indentified by ZoneId '1'

S140: Contract identified by Id '1' is assigned to one Platinum indentified by ContractId '1'

S141: Zone identified by Id '1' is assigned to one Contract indentified by ZoneId '1'

S142: DataScoop identified by Id '1' is streaming one VideoStream indentified by DataScoopId '1'

S143: StandardSubscription identified by Id '1' is assigned to many VideoStream indentified by StandardSubscriptionId '1'

S144: DataScoop identified by Id '1' records many Data indentified by DataScoopId '1'

S145: Location identified by Id '1' is stored in one Data indentified by LocationId '1'

S146: Supplier identified by Id '1' supplies many Part indentified by SupplierId '1'

S147: Part identified by PartNo '1' is assigned to many MaintenanceRecord indentified by PartPartNo '1'

S148: Subscriber identified by Id '1' is assigned to one StandardSubscription indentified by SubscriberId '1'

S149: StandardSubscription identified by Id '1' is assigned to one Gold indentified by StandardSubscriberId '1'

S150: Gold identified by StandardSubscriptionId '1' is assigned to one Platinum indentified by GoldStandardSubscriptionId '1'

S151: Platinum identified by GoldStandardSubscriptionId '1' is assigned to one SuperPlatinum indentified by PlatinumGoldStandardSubscriptionId '1'

S152: Information identified by Id '1' is related to one Staff indentified by InformationId '1'

S153: Information identified by Id '1' is related to one Supplier indentified by InformationId '2'

S154: Information identified by Id '1' is related to one Subscriber indentified by InformationId '3'

S155: Staff identified by Id '1' derives one Salesperson indentified by StaffId '1'

S156: Staff identified by Id '1' derives one Maintenance indentified by StaffId '2'

S157: Staff identified by Id '1' derives one Administrative Executive indentified by StaffId '3'

S158: Salesperson identified by StaffId 'Id '1' may sell one Sale indentified by SalespersonStaffId '1'

S159: Administrative Executive identified by StaffId '2' may discount one Sale indentified by AdministrativeExecutiveStaffId '2'

S160: StandardSubscription identified by Id '1' is sold through one Sale indentified by StandardSubscriptionId '3'

Gold -> GoldSubscription

Platinum -> PlatinumSubscription

SuperPlatinum -> SuperPlatinumSubscription

Maintenance -> MaintenancePerson

# Conclusion

The transformation between the conceptual and logical model has proved to be very insightful into the scenario surrounding FlightStream and how their organization functions and how this is modelled in a database.

Through performing normalisation, I was able to ensure that most of my tables are normalised to 3NF, reducing chances of anomalies occurring during physical implementation.

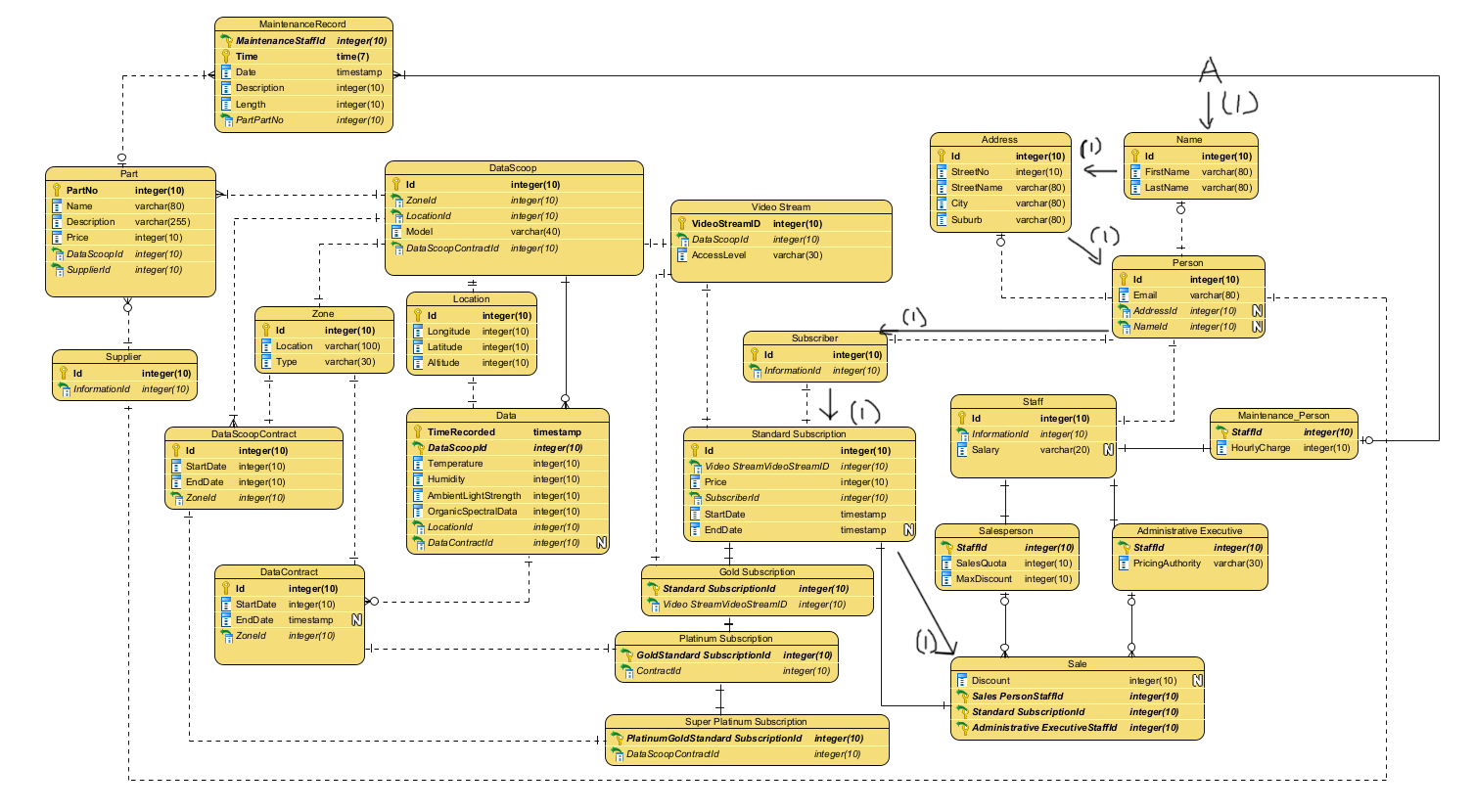
Performing the NaLER analysis proved to provide some insightful information surrounding the notation, and naming conventions chosen, which require some tweaking.

Through this process I believe that my understanding of the database and future iterations of how it will be developed has been greatly improved.

# Chapter 3: Physical Implementation

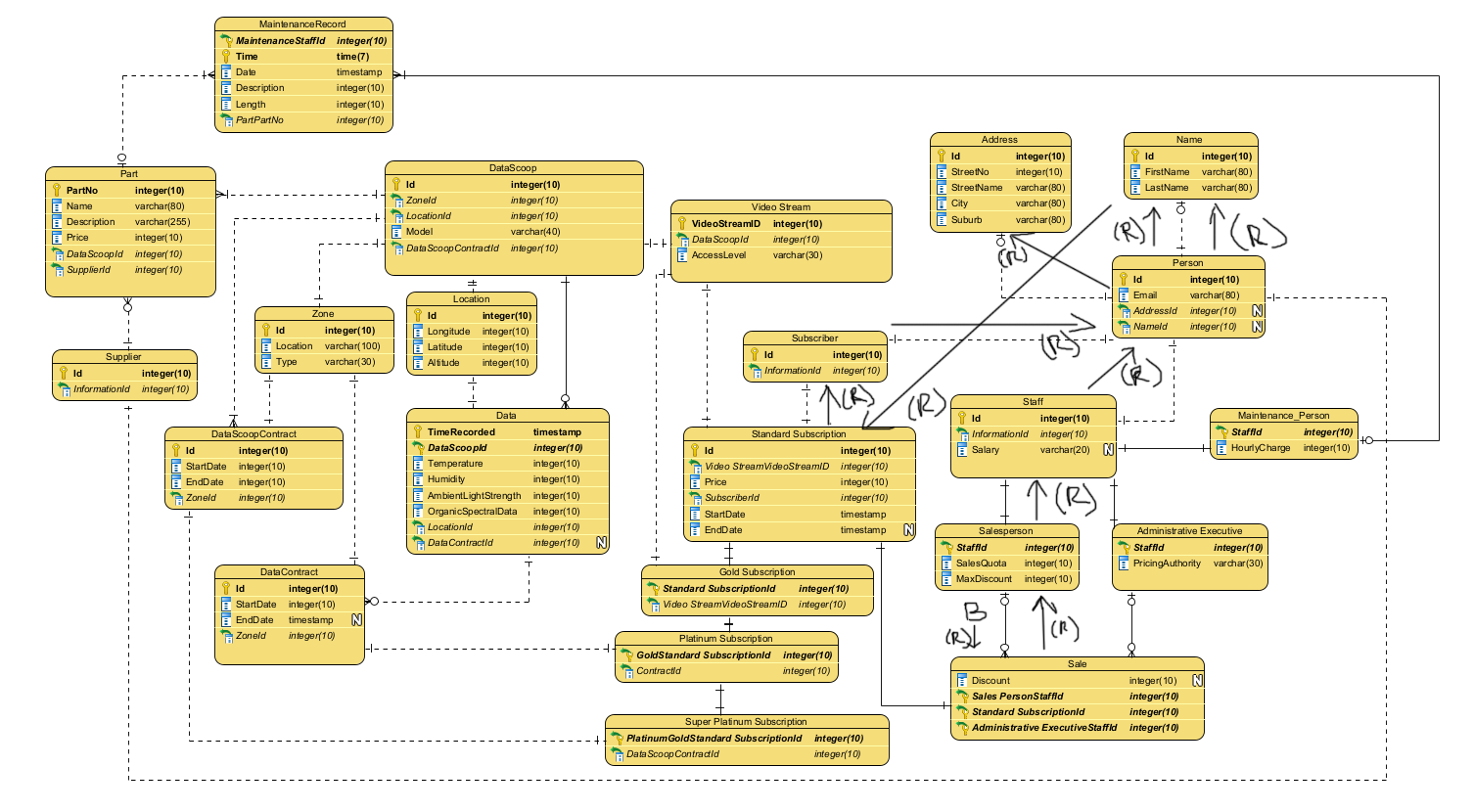
# Transaction Analysis

## Transaction A



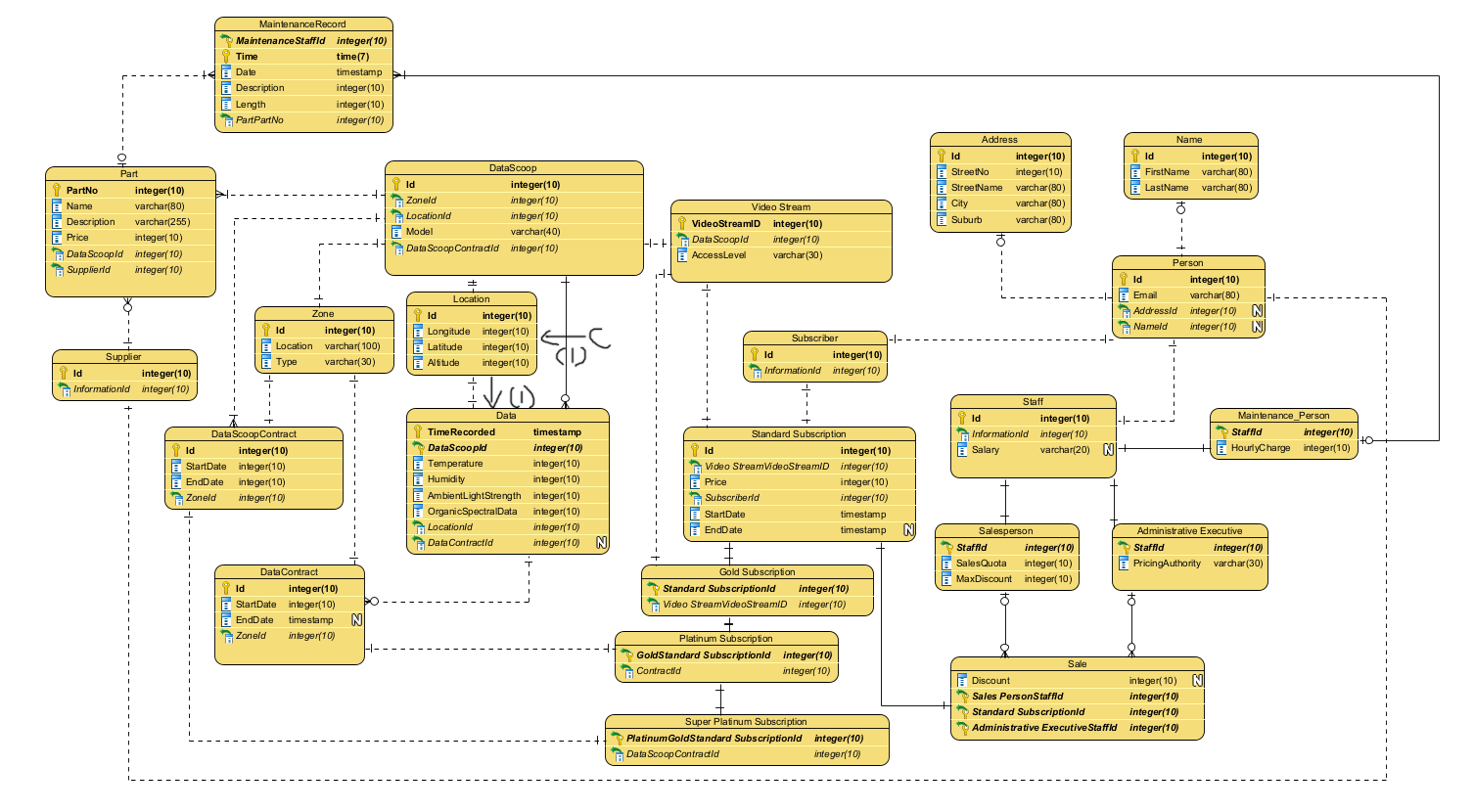
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Transaction | A |  |  |  |  |
| Peak (avg) | Day | Time | Number of runs per hour. |  |  |
| Daily | Mon-Fri | 9am-5pm | 10 |  |  |
| From Relation | To Relation | Attributes | Access | Number of Times Accessed |  |
| - | Name |  | I(E) | 1 |  |
|  |  | (all) |  |  |  |
| - | Address |  | I | 1 |  |
|  |  | (all) |  |  |  |
| Name, Address | Person |  | I | 1 |  |
|  |  | (all) |  |  |  |
| Person | Subscriber |  | I | 1 |  |
|  |  | (all) |  |  |  |
| Subscriber | Standard Subsription |  | I | 1 |  |
|  |  | (all) |  |  |  |
| Standard Subscription | Sale |  | I | 1 |  |
|  |  | (all) |  |  |  |

## Transaction B



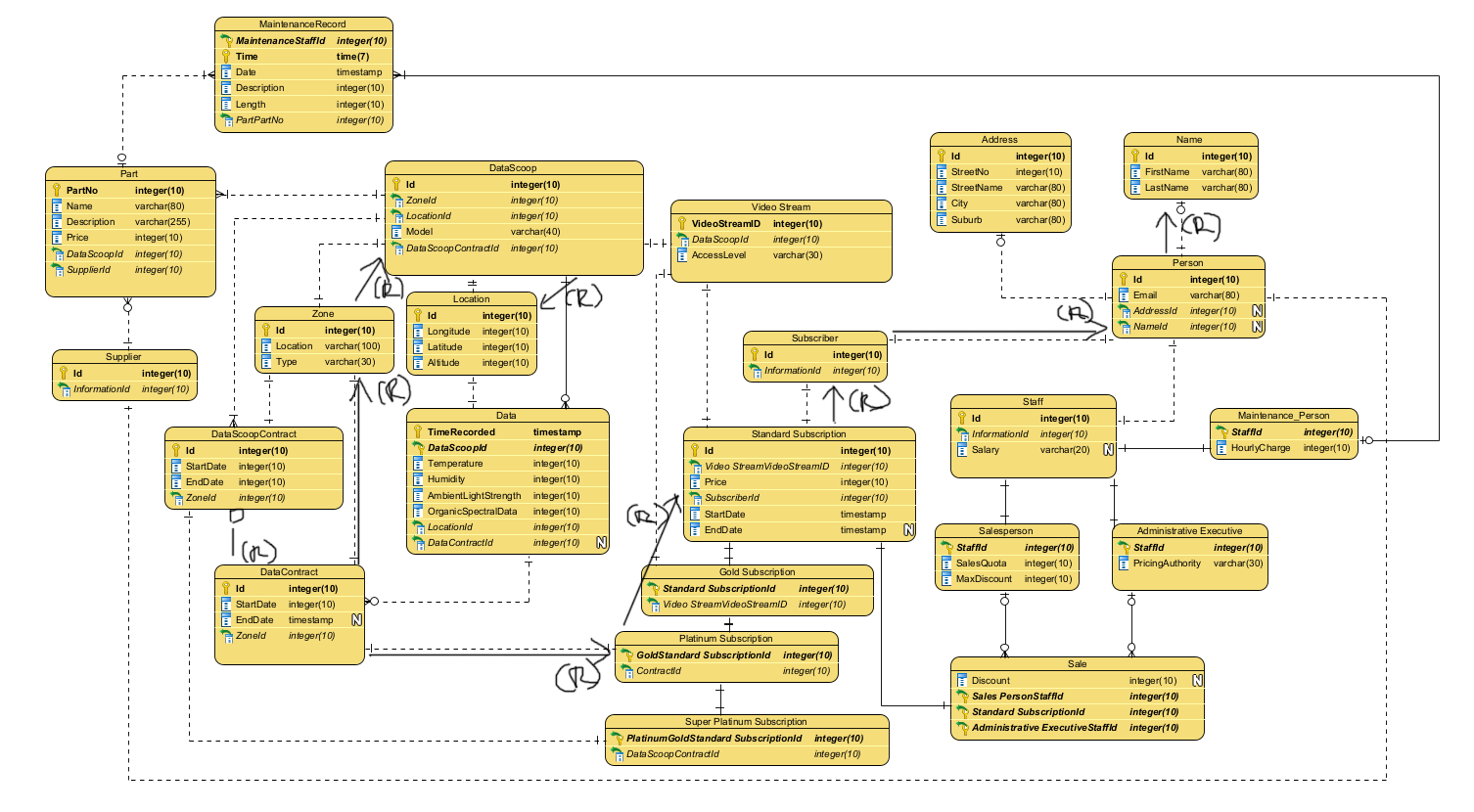
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Transaction | B |  |  |  |  |
| Peak (avg) | Day | Time | Number of runs per hour. |  |  |
| Daily | Mon-Sun | 9am-5pm | 10 |  |  |
| From Relation | To Relation | Attributes | Access | Number of Times Accessed |  |
| - | Salesperson |  |  | 3,000 |  |
|  |  | StaffID | R(E) |  |  |
|  |  |  |  |  |  |
| Salesperson | Staff |  |  | 3,000 |  |
|  |  | PersonID | R |  |  |
|  |  | StaffID | R |  |  |
| Staff | Person |  |  | 3,000 |  |
|  |  | PersonID | R |  |  |
|  |  | NameID | R |  |  |
| Person | Name |  |  | 3,000 |  |
|  |  | (all) |  |  |  |
| - | Sale |  |  | 166 |  |
|  |  | SalespersonID | R |  |  |
|  |  | SubscriptionID | R |  |  |
|  |  | Discount | R |  |  |
| Sale | Standard Subscription |  |  | 166 |  |
|  |  | SubscriptionID | R |  |  |
|  |  | SubscriberID | R |  |  |
| Standard Subscription | Subscriber |  |  | 166 |  |
|  |  | SubscriberID | R |  |  |
|  |  | PersonID | R |  |  |
| Subscriber | Person |  | R | 166 |  |
|  |  | (all) |  |  |  |
| Person | Name |  | R | 166 |  |
|  |  | (all) |  |  |  |
| Person | Address |  | R | 166 |  |
|  |  | (all) |  |  |  |

Transaction C



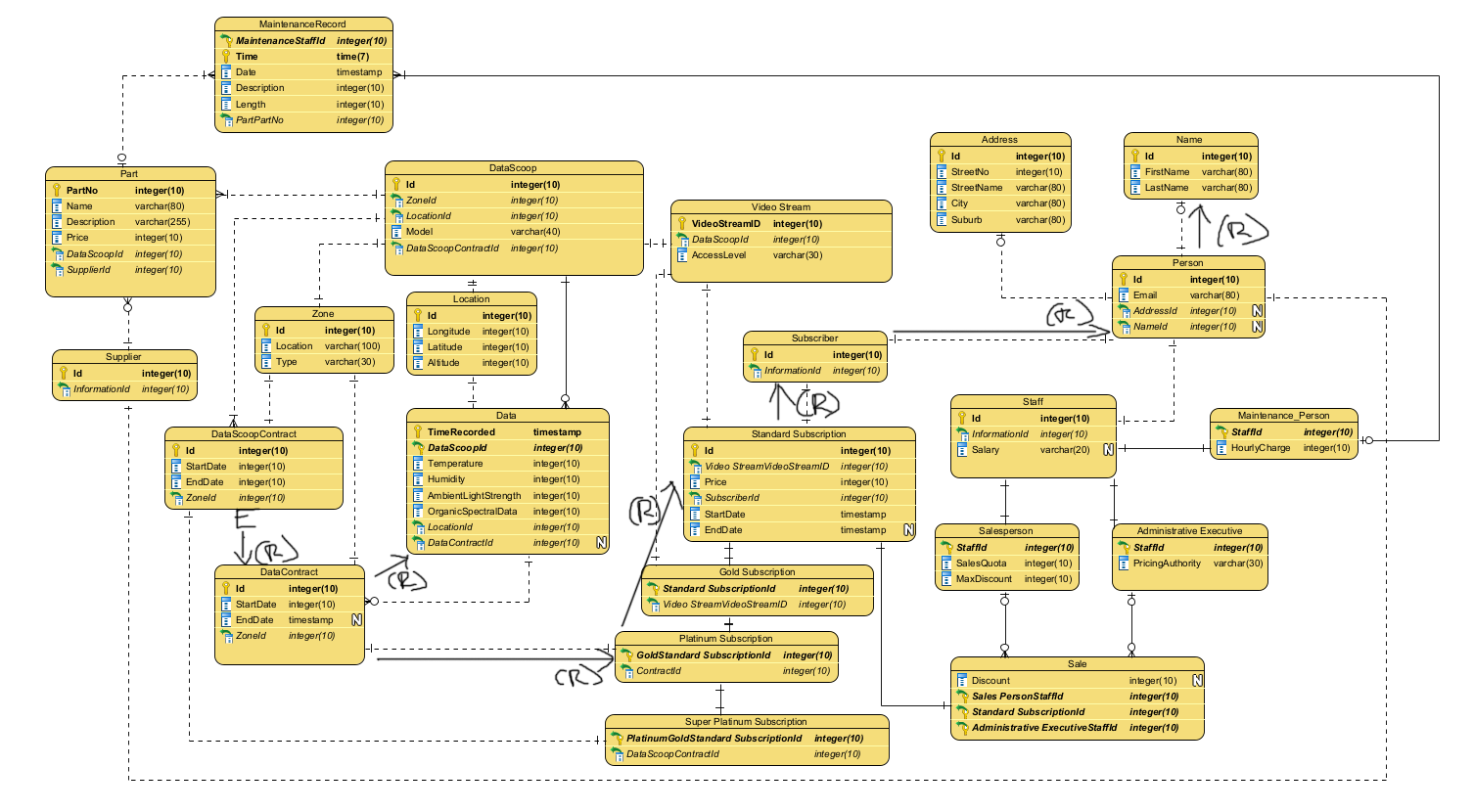
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Transaction | C |  |  |  |  |
| Peak (avg) | Day | Time | Number of runs per hour. |  |  |
| Daily | Mon-Sun | 12:00AM-11:59PM | 1000 |  |  |
| From Relation | To Relation | Attributes | Access | Number of Times Accessed |  |
| - | Location |  | I(E) | 1 |  |
|  |  | (all) |  |  |  |
| Location | Data |  | I | 1 |  |
|  |  | (all) |  | 1 |  |

## Transaction D



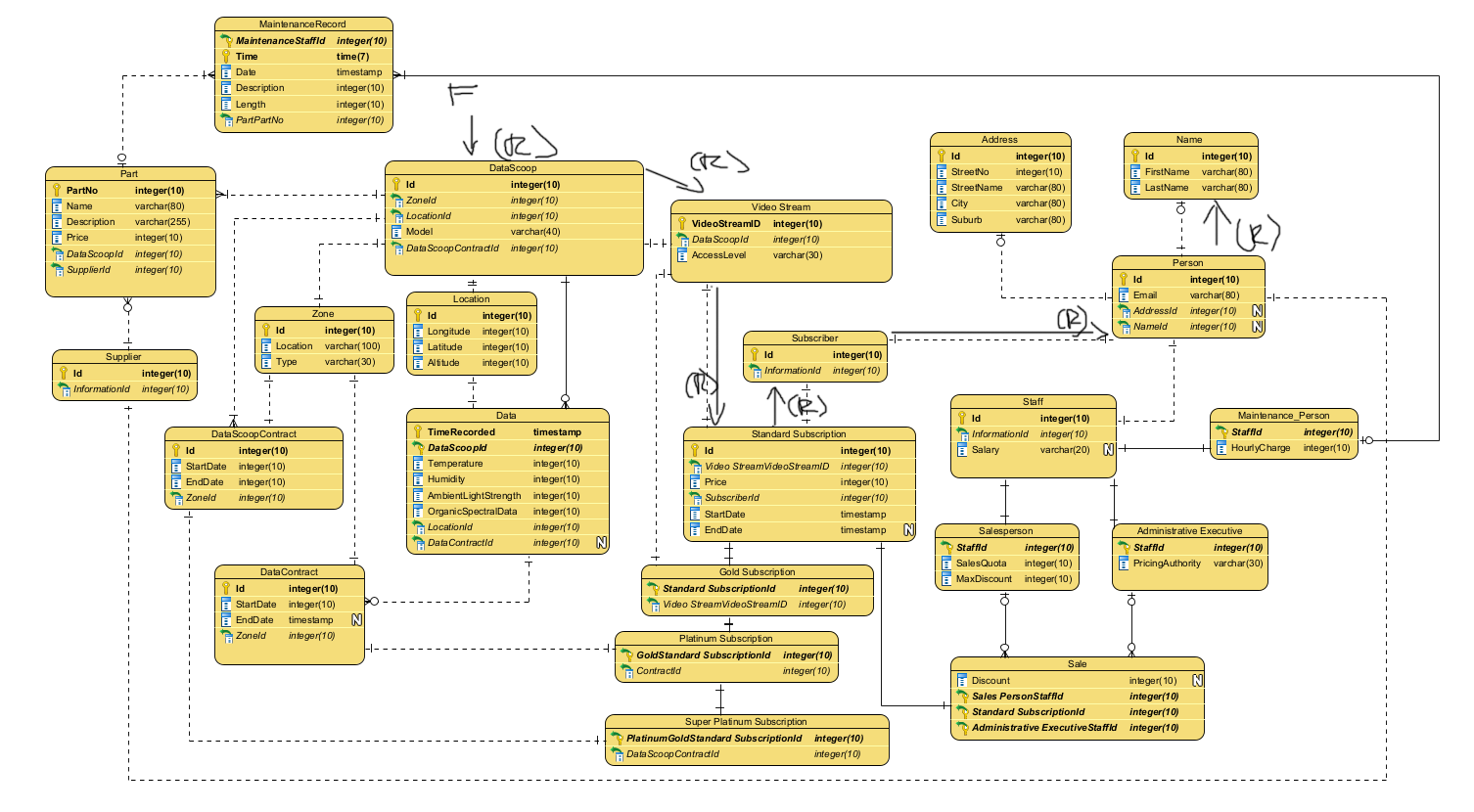
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Transaction | D |  |  |  |  |
| Peak (avg) | Day | Time | Number of runs per hour. |  |  |
| Daily | Mon-Sun | 9am-5pm | 100 |  |  |
| From Relation | To Relation | Attributes | Access | Number of Times Accessed |  |
| - | DataContract |  |  | 100 |  |
|  |  | ZoneID | R(E) |  |  |
|  |  | DataContractID | R |  |  |
| DataContract | Zone |  |  | 100 |  |
|  |  | ZoneID | R |  |  |
| Zone | DataScoop |  |  | 250,000 |  |
|  |  | ZoneID | R |  |  |
|  |  | DataScoopID | R |  |  |
|  |  | LocationID | R |  |  |
| DataScoop | Location |  |  | 250,000 |  |
|  |  | LocationID | R |  |  |
|  |  | Latitude | R |  |  |
|  |  | Longitude | R |  |  |
| DataContract | Platinum Subscription |  |  | 100 |  |
|  |  | SubscriptionID | R |  |  |
|  |  | DataContractID | R |  |  |
| Platinum Subscription | Standard Subscription |  |  |  |  |
|  |  | SubscriptionID | R | 100 |  |
|  |  | SubscriberID | R |  |  |
| Standard Subscription | Subscriber |  |  | 100 |  |
|  |  | SubscriberID | R |  |  |
|  |  | PersonID | R |  |  |
| Subscriber | Person |  | R | 100 |  |
|  |  | (all) |  |  |  |
| Person | Name |  | R | 100 |  |
|  |  | (all) |  |  |  |

## Transaction E



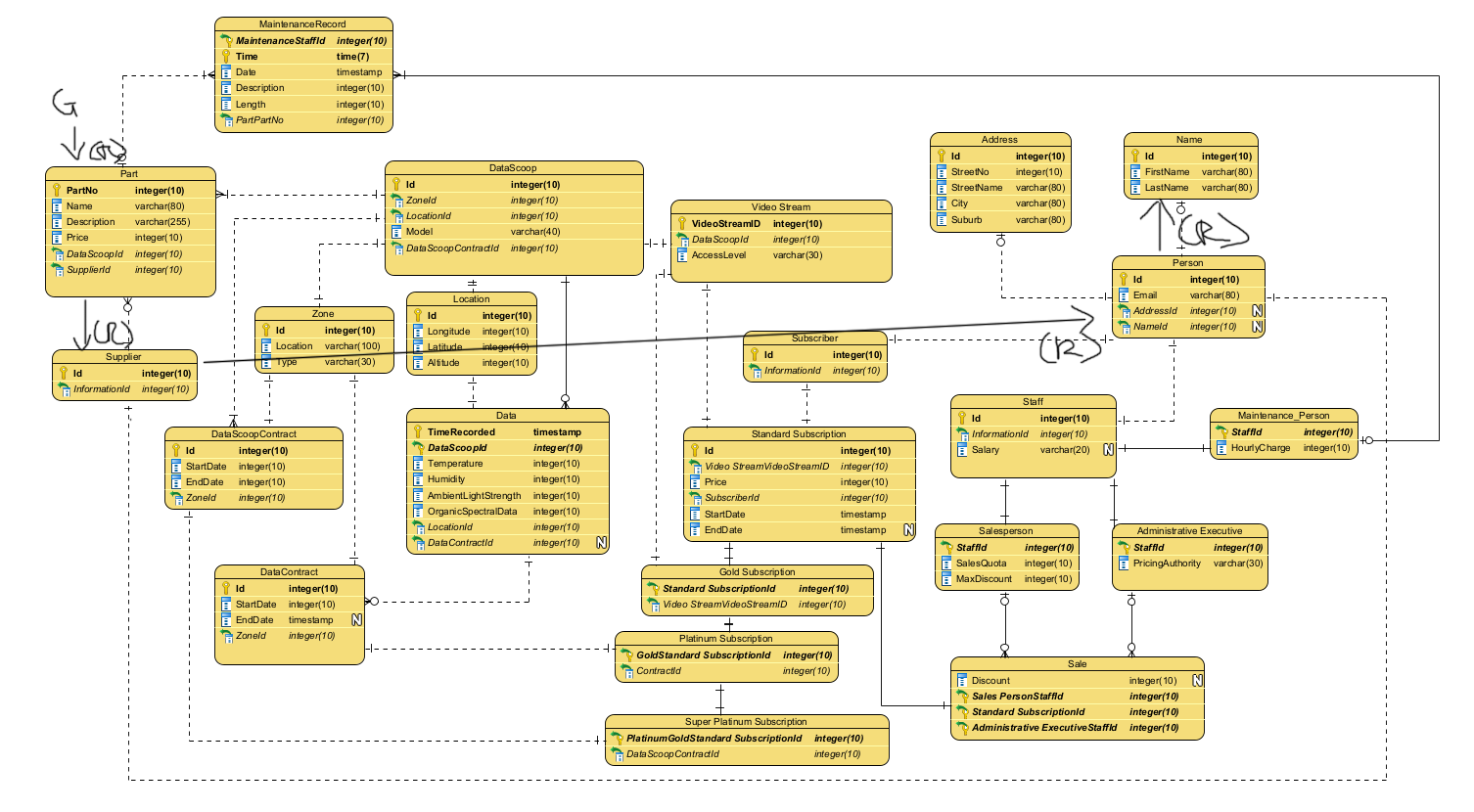
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Transaction | E |  |  |  |  |
| Peak (avg) | Day | Time | Number of runs per hour. |  |  |
| Daily | Mon-Sun | 9am-5pm | 100 |  |  |
| From Relation | To Relation | Attributes | Access | Number of Times Accessed |  |
| - | DataContract |  |  | 1 |  |
|  |  | DataContractID | R(E) |  |  |
| DataContract | Data |  | R | 100,000 |  |
|  |  | (all) |  |  |  |
| DataContract | Platinum Subscription |  |  | 1 |  |
|  |  | SubscriptionID | R |  |  |
|  |  | DataContractID | R |  |  |
| Platinum Subscription | Standard Subscription |  |  | 1 |  |
|  |  | SubscriptionID | R |  |  |
|  |  | SubscriberID | R |  |  |
| Standard Subscription | Subscriber |  |  | 1 |  |
|  |  | SubscriberID | R |  |  |
|  |  | PersonID | R |  |  |
| Subscriber | Person |  | R | 1 |  |
|  |  | (all) |  |  |  |
| Person | Name |  | R | 1 |  |
|  |  | (all) |  |  |  |

## Transaction F



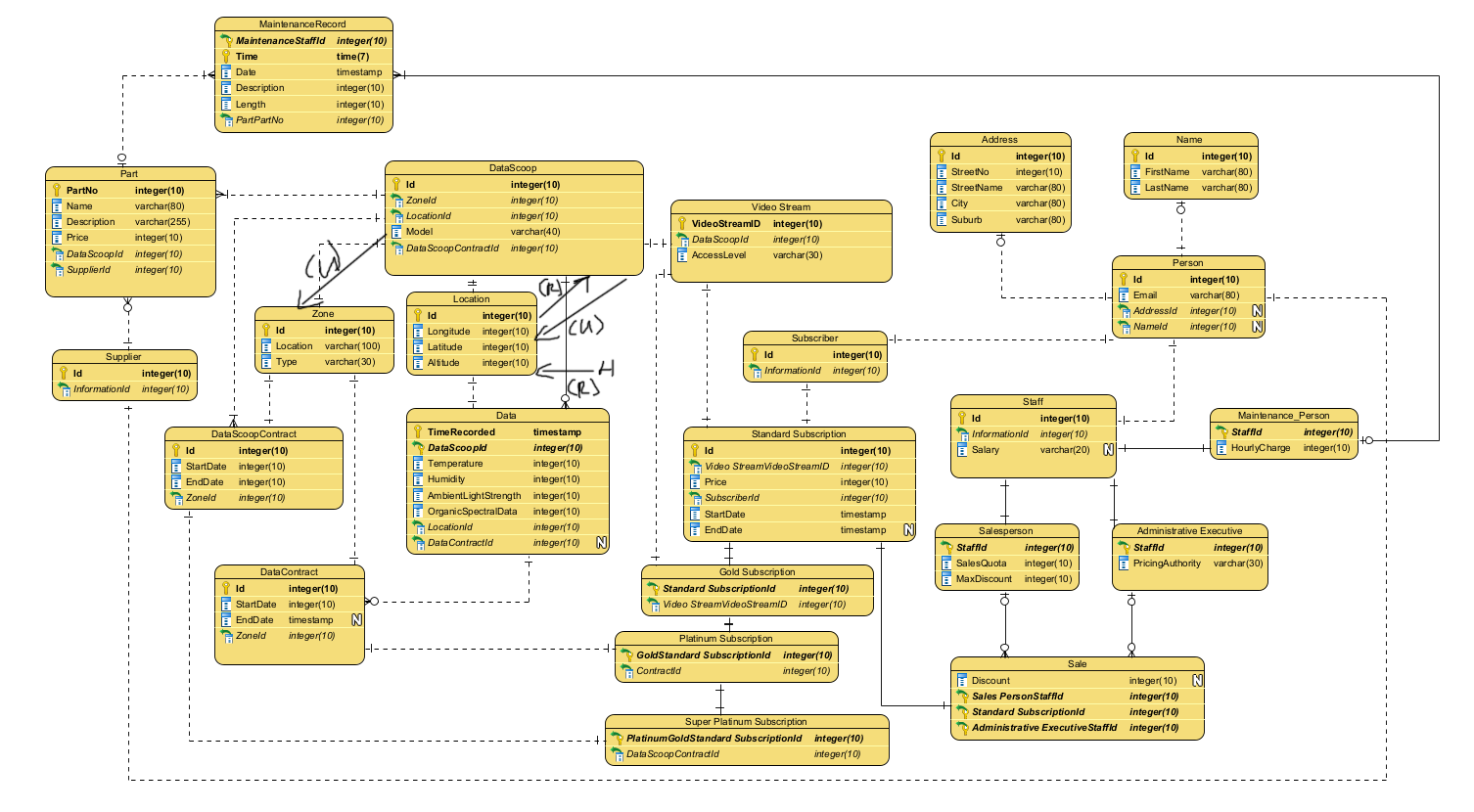
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Transaction | F |  |  |  |  |
| Peak (avg) | Day | Time | Number of runs per hour. |  |  |
| Daily | Mon-Sun | 9am-5pm | 60 |  |  |
| From Relation | To Relation | Attributes | Access | Number of Times Accessed |  |
| - | DataScoop |  |  | 50,000 |  |
|  |  | DataScoopID | R(E) |  |  |
|  |  | VideoStreamID | R |  |  |
| DataScoop | VideoStream |  |  | 50,000 |  |
|  |  | VideoStreamID |  |  |  |
|  |  | SubscriberID |  |  |  |
| VideoStream | Standard Subscription |  |  | 500,000 |  |
|  |  | SubscriptionID | R |  |  |
|  |  | SubscriberID | R |  |  |
| Standard Subscription | Subscriber |  |  | 500,000 |  |
|  |  | SubscriberID | R |  |  |
|  |  | PersonID | R |  |  |
| Subscriber | Person |  | R | 500,000 |  |
|  |  | (all) |  |  |  |
| Person | Name |  | R | 500,000 |  |
|  |  | (all) |  |  |  |

## Transaction G



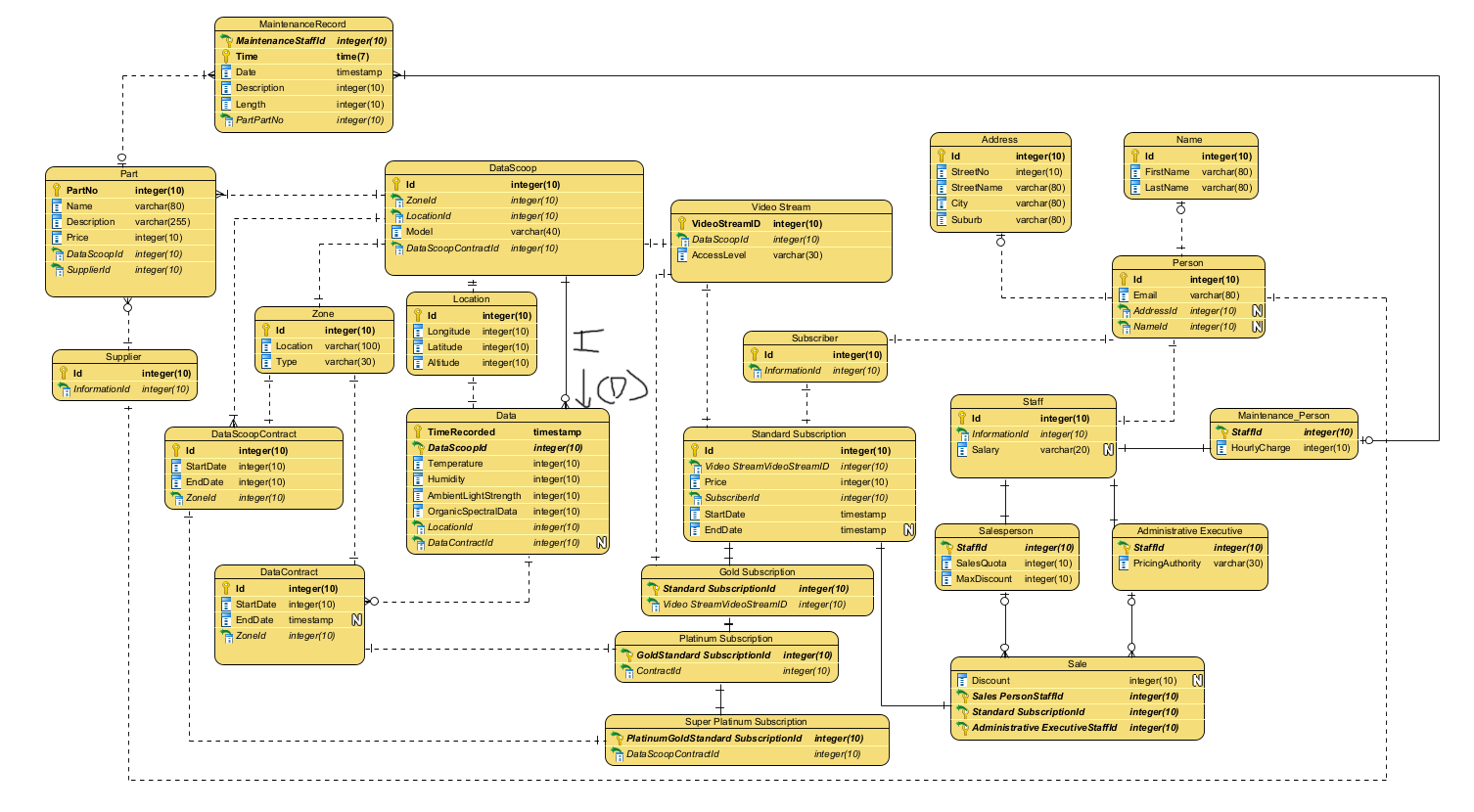
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Transaction | G |  |  |  |  |
| Peak (avg) | Day | Time | Number of runs per hour. |  |  |
| Daily | Mon-Sun | 9am-5pm | 10 |  |  |
| From Relation | To Relation | Attributes | Access | Number of Times Accessed |  |
| - | DataScoop |  |  | 1 |  |
|  |  | DataScoopID | R(E) |  |  |
|  |  |  |  |  |  |
| DataScoop | Part |  |  | 18-22 |  |
|  |  | Name | R |  |  |
|  |  | SupplierID | R |  |  |
| Part | Supplier |  |  | 10-20 |  |
|  |  | SupplierID | R |  |  |
|  |  | PersonID | R |  |  |
| Supplier | Person |  | R | 10-20 |  |
|  |  | (all) |  |  |  |
| Person | Name |  | R | 10-20 |  |
|  |  | (all) |  |  |  |

## Transaction H



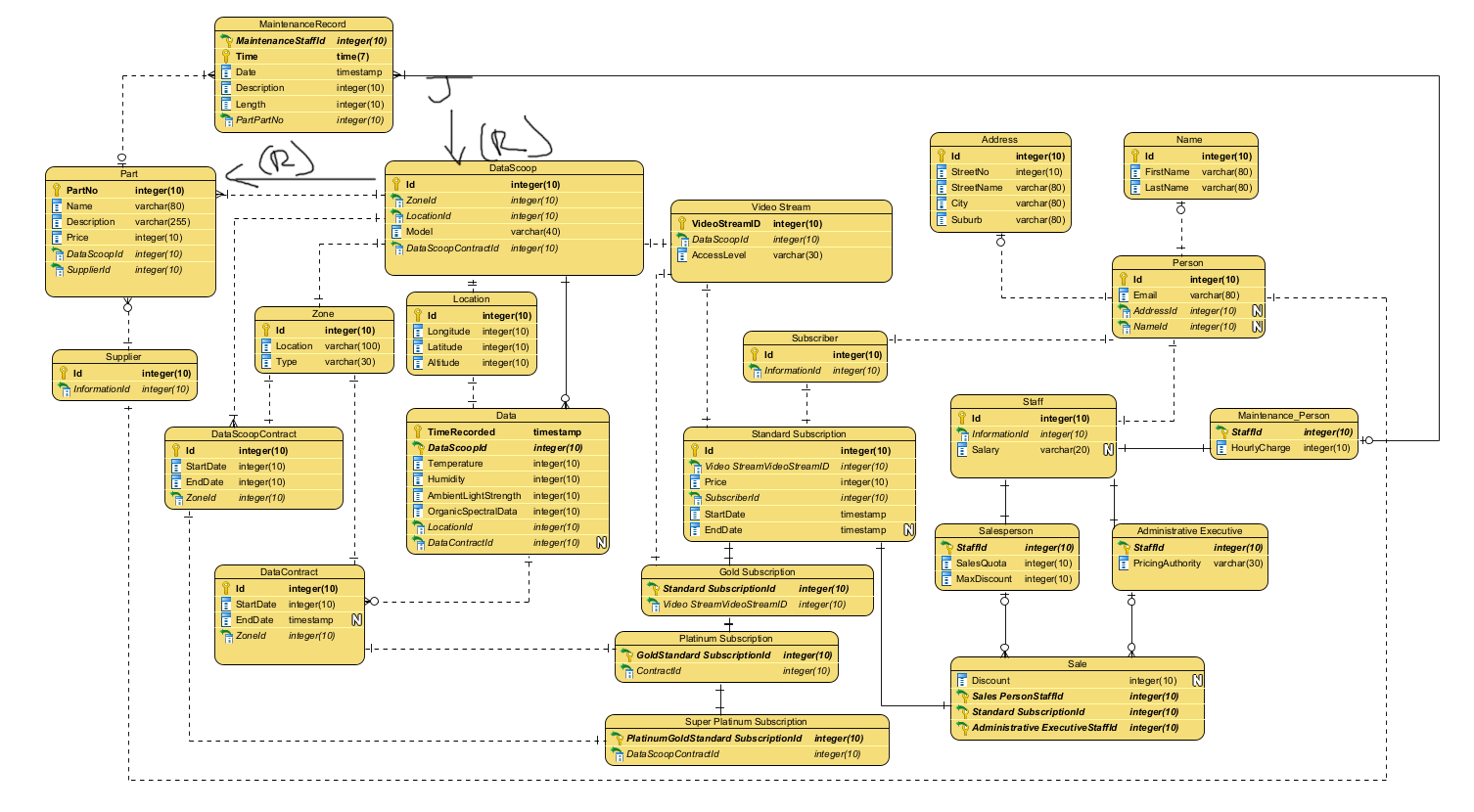
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Transaction | H |  |  |  |  |
| Peak (avg) | Day | Time | Number of runs per hour. |  |  |
| Daily | Mon-Sun | 12:00AM-11:59PM | 60 |  |  |
| From Relation | To Relation | Attributes | Access | Number of Times Accessed |  |
| - | Location |  |  | 1 |  |
|  |  | LocationID | R(E) |  |  |
| Location | DataScoop |  |  | 1 |  |
|  |  | LocationID | R |  |  |
|  |  | DataScoopID | R |  |  |
| DataScoop | Location |  | U | 1 |  |
|  |  | (all) |  |  |  |
| DataScoop | Zone |  | U | 1 |  |
|  |  | (all) |  |  |  |

## Transaction I



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Transaction | I |  |  |  |  |
| Peak (avg) | Day | Time | Number of runs per hour. |  |  |
| Daily | Mon-Sun | 9am-5pm | 60 |  |  |
| From Relation | To Relation | Attributes | Access | Number of Times Accessed |  |
|  | Data |  | D(E) | 1 |  |
|  |  | (all) |  |  |  |

## Transaction J



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Transaction | J |  |  |  |  |
| Peak (avg) | Day | Time | Number of runs per hour. |  |  |
| Daily | Mon-Sun | 9am-5pm | 1 |  |  |
| From Relation | To Relation | Attributes | Access | Number of Times Accessed |  |
| - | DataScoop |  |  | 50,000 |  |
|  |  | DataScoopID | R(E) |  |  |
| DataScoop | Part |  |  | 1,000,000 |  |
|  |  | Price | R |  |  |

Through the process of transaction analysis, there are various tables that frequently access columns should most likely have indexes on them. There needs to be a balance as indexing speeds up the access of data however slows down updating, inserting, or deleting to the table.

Indexes will most likely be added to foreign keys, especially on the Person, Name and Address tables as these are accessed very frequently. Columns such as FirstName and LastName are also accessed very frequently so adding indexes here are effective.

Adding indexes to the columns in the data table would seem to be effective, however Transaction C shows that this table is updated very frequently, so this would likely slow down Transaction C greatly.

Using indexes and transaction analysis a clearer idea of how the database will be constructed can be formed. It also helps to improve the efficiency of the database, and provides the ground work for creating the transactions.

# Database Implementation

Ensuring that the DDL accurately describes the logical model developed previously is crucial to ensuring an effective database implementation.

First tables are created with their respective attributes, ensuring that all datatypes are accurate with valid naming conventions. Ensuring that datatype sizes are reduced as far as possible without truncating data, to ensure that disk space is optimized is the next key step. Adding the required constraints, including FK, and PKs using ‘ALTER TABLE’ and ‘ADD CONSTRAINT’ key words. This ensures that data validation is being correctly implemented, reducing data malformities. Furthermore, to improve performance of transactions indexes, clustered and non-clustered can be added here, based on the transaction analysis results.

Once the database is correctly defined, it is ready to be loaded with test data.

Furthermore users, roles and views are defined to ensure that data is secure, with permissions aligned with the business rules and requirements.

Subscriber – This role is assigned to a subscriber. Permissions include the ability to update their VideoStream, Person details.

Salesperson – This role is assigned to a salesperson. Permissions include the ability to update, insert and delete on tables related to subscribers, subscriptions, and sales.

Admin-Executive – This role is assigned to all administrative executive employees. Permissions include the all access on tables related to subscribers, subscriptions, sales, contracts, and salespersons.

Pre-defined role “db\_owner” – Grants all access to a specific database, this is assigned to the db\_admin.

NOTE: In the case of the development environment all database, table and procedure definitions are include a ‘DROP “TYPE” “NAME” IF EXISTS’ to ensure that the script can be run multiple times without errors.

## Create Database

A white background with black text

Description automatically generated

## Create Table

A computer screen shot of a computer code

Description automatically generated

## Add Constraints

A close-up of a computer code

Description automatically generated

## Create Roles

A screenshot of a computer program

Description automatically generated

## Create Test User

A computer code with text

Description automatically generated with medium confidence

## Grant Permissions

A screenshot of a computer screen

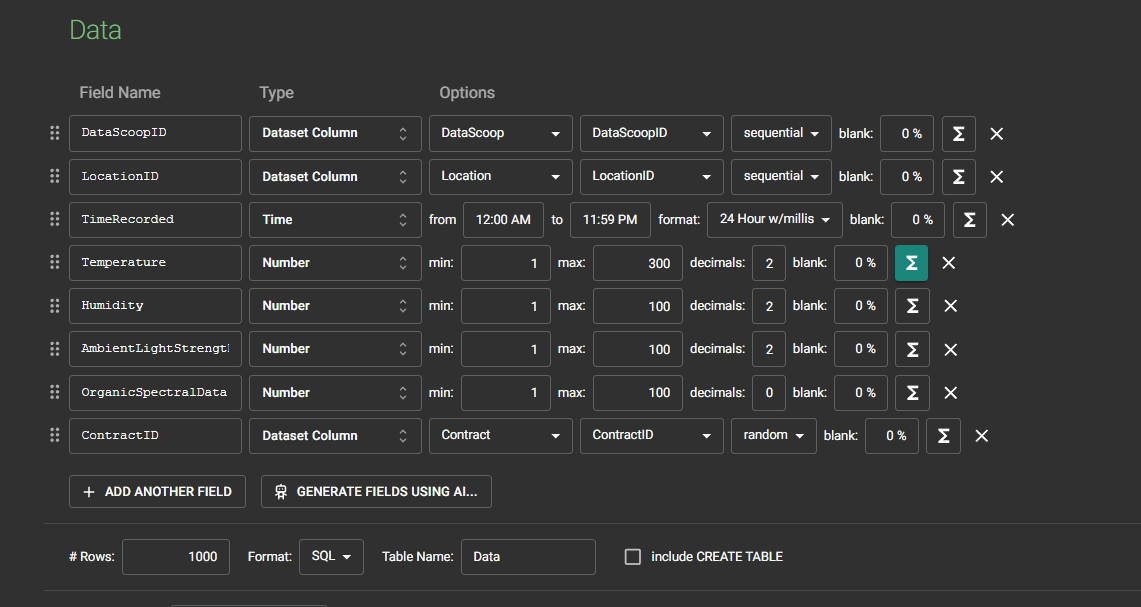
Description automatically generated

# Test Data

When generating test data, it is important to ensure that it is realistic in the sense that all FK and PK constraints are met with correct data types and somewhat realistic dummy data. This includes ensuring there are a reasonable number of records for each table. This allows for transactions, testing, etc to be done in a realistic environment where edge cases and other various errors may arise, allowing for effective debugging and refactoring. Mockaroo offers a wide range of different data generation tools in the context of a database.

Using schemas for each table in the database and assigning appropriate data types, names and meeting all constraints data was able to be generated and entered the database.

## Data Schema:



# Summary

Effective data management is crucial in any organization for ensuring that data is handled correctly, effectively, efficiently and aligns with the business objectives. This is particularly important for a company as data centric as FlightStream. Following some principles of data management will help to improve these aspects of data management in an organisation.

### The Principle of Integrity

“Data governance program shall be constructed so the records and information generated or managed by or for the organization have a reasonable and suitable guarantee of authenticity and reliability.” (ARMA, n.d.)

Through the use various data modelling techniques, including constraints, primary keys, foreign keys, and data types, ensures that data is reliable, accurate and maintainable. Normalization is a major stage in ensuring that data redundancy is minimized (Microsoft, 2023), in turn reducing chances of error during deletion, insertion and the updating of data (GeeksForGeeks, 2024).

Primary and foreign keys ensure that each record is unique and maintain referential integrity, e.g. The SubscriberID column in the Standard\_Subscription table refers to an entry in the Subscriber table.

Using appropriate data type ensures that data is aligned with the expected formats, such as dates, numbers, strings, etc. This ensures that data errors are reduced and maintains consistency across a table.

Normalisation follows guidelines set in the form of normal forms. These range from 1NF-5NF with the inclusion of BCNF (Boyce-Codd) in between 3NF and 4NF. The process essentially involves splitting tables until they can no longer be split and maintain uniqueness in records. Once again ensuring that data malformities, insertion, deletion and update errors are minimized.

### The Principle Of Protection

“The data governance program shall be constructed to ensure a reasonable level of protection to data that is personal or that otherwise requires protection.” (ARMA, n.d.)

This was done using roles, permissions, and user authentication. This helps to ensure that data can only be accessed by intended persons, and only required data is exposed.

By defining roles and granting only appropriate permissions ensures that users can only access data necessary for their role, e.g. the db\_admin will have full control over the database and its tables while a salesperson will only have access to insert data into specific tables and view certain data from pre-defined views. This limits data access to only what is required for a role.

By creating user accounts, with authentication required, ensures that no data is openly exposed to any unauthorized person. This is crucial to maintaining a secure environment.

### The Principle of Compliance

“An information governance program shall be constructed to comply with applicable laws and other binding authorities, as well as the organization’s policies.” (ARMA, n.d.)

Flight Stream should adhere to all applicable laws and regulations. To ensure that Flight Stream is effective following these rules, effective policies and procedures should be implemented surround data management, collections, and accessibility.

This can mostly be done through the design, and implementation of the database ensuring it follows other key principles of data management, along with ensuring that staff are correctly trained and will understand, and follow required procedures, and policies. These policies should be clear, consistent, and reviewed consistently to ensure they are effective for FlightStream, and to align them with the business objectives.

### The Principle of Availability

“An organization shall maintain its information or data assets in a manner that ensures timely, efficient, and accurate retrieval of its information. It includes the development of robust backup/recovery plans, efficient disaster recovery and Service Level Agreements as well.” (ARMA, n.d.)

The database should be designed to data is available, and easily accessible. This is done using indexes, derived from transactions analysis. This makes use of indexes on frequently accessed attributes, enhancing query performance. Role-based access, ensuring users, are assigned correct roles with sufficient permissions, making use of pre-defined views and stored procedures to ensure that data access is streamlined and efficient.

### The Principle of Retention

“An organization shall retain its information or data assets for an appropriate time, taking into account all operational, legal, regulatory and fiscal requirements, and those of all relevant binding authorities. “ (ARMA, n.d.)

This principle ensures that data is being retained for an appropriate period of time that aligns with business requirements and objectives and is justified for these reasons. DataScoop devices send data upstream to the database, along with local copies on the drone itself. By retaining this data for one month allows for data be analysed and is crucial for contracted subscribers.

Furthermore, customer support is backed by retained data surrounding customer information, this includes having access relevant data in case of an issue, allowing for trouble shooting and efficiency in resolving issues. These reasons justify the retention of data in the database, and algin with the business objectives.

Hand in the URL to a GitHUB repository created and owned by you that contains the work from parts one and two above.

<https://github.com/Ollie-Moss/DAT601>

# **References**

Microsoft. (2023, July 14). *Description of the database normalization basics*. https://learn.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description

McAleer, K. (n.d.). *FIRST NORMAL FORM (1NF)*. Kevs Robots. https://www.kevsrobots.com/learn/sqlite3/13\_1nf.html

McAleer, K. (n.d.). *SECOND NORMAL FORM (2NF)*. Kevs Robots. www.kevsrobots.com/learn/sqlite3/14\_2nf.html

McAleer, K. (n.d.). *THIRD NORMAL FORM (3NF)*. Kevs Robots. https://www.kevsrobots.com/learn/sqlite3/15\_3nf.html

McAleer, K. (n.d.). *BOYCE-CODD NORMAL FORM (BCNF)*. Kevs Robots. https://www.kevsrobots.com/learn/sqlite3/15\_3nf.html

Geeks for Geeks. (2024, May 23). *Normal Forms in DBMS*. https://www.geeksforgeeks.org/normal-forms-in-dbms/

ARMA. (n.d.). *The Principles*. https://www.arma.org/page/principles