Bozhi Chen

DAT601 DATAbase

DAT601 – MileStone 1.2.3

Content

[MileStone 1 1](#_Toc170248973)

[A Description of Conceptual Modelling 1](#_Toc170248974)

[Common components of a Chen ntity-Relationship Diagram or ERD. 2](#_Toc170248975)

[Entities 3](#_Toc170248976)

[Attributes 4](#_Toc170248977)

[Keys 7](#_Toc170248978)

[Relationships 7](#_Toc170248979)

[Relationship Degrees 9](#_Toc170248980)

[Chen Conceptual Entity Relationship Model 9](#_Toc170248981)

[Reasoning and Purpose of all Parts of the Model 9](#_Toc170248982)

[Data Dictionary 11](#_Toc170248983)

[Entities 11](#_Toc170248984)

[Relationships 12](#_Toc170248985)

[Attributes 13](#_Toc170248986)

[Assumed Business Rules (with reasoning) 18](#_Toc170248987)

[MileStone 2 21](#_Toc170248988)

[Introduction 21](#_Toc170248989)

[Conceptual to Logical 21](#_Toc170248990)

[Chens ERD Notation 21](#_Toc170248991)

[Mapping Rules 23](#_Toc170248992)

[Normalisation 30](#_Toc170248993)

[First Normal Form 30](#_Toc170248994)

[Second Normal Form 30](#_Toc170248995)

[Third Normal Form 31](#_Toc170248996)

[Boyce-Codd Normal Form 31](#_Toc170248997)

[Fourth Normal Form 31](#_Toc170248998)

[Logical ERD 33](#_Toc170248999)

[Rational & Relations 35](#_Toc170249000)

[Data Dictionary 35](#_Toc170249001)

[Entities 35](#_Toc170249002)

[Attributes 36](#_Toc170249003)

[NaLER Analysis 43](#_Toc170249004)

[STEP 1. Identify and document the diagram conventions 43](#_Toc170249005)

[STEP 2. Perform a syntax check of the model in accordance with the step 1 assumptions and make and mark any adjustments 50](#_Toc170249006)

[1. Check the name of Entity and Attribute 50](#_Toc170249007)

[2. Check Primary Key 51](#_Toc170249008)

[3. Check Foreign Key 52](#_Toc170249009)

[4. Check Relationship 52](#_Toc170249010)

[5. Check composite Entity 53](#_Toc170249011)

[Step 3.1 & 3.2 - Attribute Sentences 53](#_Toc170249012)

[Step 3.3 - Relationship Sentences 56](#_Toc170249013)

[Step 4.1 – Examples of Sentences for Attributes 57](#_Toc170249014)

[Step 4.2 – Examples of Sentences for Relationships 60](#_Toc170249015)

[Conclusion 61](#_Toc170249016)

[MileStone 3 62](#_Toc170249017)

[Step 4 Translate global logical data model for target DBMS 62](#_Toc170249018)

[Step 4.1 Design base relations for target DBMS 62](#_Toc170249019)

[Step 4.2 Design enterprise constraints for target DBMS 69](#_Toc170249020)

[Step 5 Design physical representation 70](#_Toc170249021)

[Step 5.1 Analyze transactions 70](#_Toc170249022)

[Step 5.2 Choose file organizations 77](#_Toc170249023)

[Step 5.3 Choose secondary indexes 78](#_Toc170249024)

[Step 5.4 Consider the introduction of controlled redundancy 79](#_Toc170249025)

[Step 5.5 Estimate disk space requirements 79](#_Toc170249026)

[Step 6 Design security mechanisms 85](#_Toc170249027)

[Step 6.1 Design user views 85](#_Toc170249028)

[Step 6.2 Design access rules 85](#_Toc170249029)

[Issues & Resolutions 85](#_Toc170249030)

[Conclusions from the Physical Design 86](#_Toc170249031)

[Reference 88](#_Toc170249032)

# MileStone 1

## A Description of Conceptual Modelling

**The fundamental principles of data management**

The purpose of Fundamental principles of effective data manager is to ensure that data can be fully utilized in information systems and data security is guaranteed. The following is an overview of some basic principles.

**Data Modelling:**

Data modelling can be considered as the most important basic principle of data management in data systems. Data modelling is actually the process of creating a visual representation of the whole or part of an information system. Essentially, data modelling creates an abstract data representation that defines the structure of the data and the relationships between the data, which is important for eliminating hyper data and creating an efficient database. Data modelling also helps the system to create a structured framework that facilitates and helps the system to design, develop and optimize business processes, so it is said that data modelling has an extremely important role in information systems. There are three types of data modelling which are conceptual data model, logical data model and physical data model.

**“The first and foremost data management guiding principle is data modeling. Data modeling means designing and structuring your data assets, their properties, and their inter-relationships in a logical manner.”** **(ehsanelahi, 2022)**

**Data Integration:** The format and structure of data stored internally is consistent to ensure data reliability.

**Data security and privacy:** Data security is ensured through reliable and comprehensive data protection measures and programmes to ensure that data is protected from security incidents such as illegal access, modification or being leaked. This includes both physical and cyber security measures.

**Data Availability:** Data can be accessed and worked on in a long-term, stable manner when it needs to be used, and this data needs to be stored properly. Security programmes and backup plans can be put in place.

**Data legality:** The storage and management of all data must comply with local laws and regulations to ensure that all data is legal.

**Minimum redundancy:** Standardise the structure of stored data, optimise data management methods, reduce all useless and repetitive data as much as possible, maximise data management efficiency and improve database retrieval performance.

**Data Backup:** Data backup and data recovery are essential measures. The use of data backup and data recovery techniques in the event of data corruption or loss can ensure data integrity and business process continuity.

**Data lifecycle:** The data lifecycle usually includes collection, storage, use and deletion. Systematic management of the data lifecycle can optimise data.

## Common components of a Chen ntity-Relationship Diagram or ERD.

**Conceptual modelling using Chen ERD:**

Chen ERD is a tool used for conceptual data modelling. Conceptual modelling using Chen ERD can be divided into the following steps, which are identifying entities, defining relationships, adding attributes, defining primary and foreign keys, and checking the integrity of ERD. Firstly, identify the key entities in the system, entities can represent real world things or concepts and are represented by rectangle. Secondly, define inter-entity relationships for the entities by representing the relationships with diamond and connecting the related entities with lines. Then, attributes are added to the entities. Attributes provide additional descriptions of the entities and relationships (this includes defining the primary and foreign keys of the entities, which are represented by oval). Finally, the integrity of the created ERD is checked (e.g. entities are identified correctly, entity attributes are added correctly, relationships between entities are connected correctly, etc.) to make sure that all the steps have been completed and to ensure the successful completion of the conceptual data modelling.

**Specialization**

Specialization is the process of defining subclasses of entity types. Some specific entities do not have the usual properties, but rather have specific properties. Through Specialization, specific properties can be defined for a particular entity type that are more specific and appropriate, and each entity can have its own specific properties and relationships that can be used to make the representation more specific and vivid.

**Generalization**

Generalization

Generalization has some similarities with abstract methods. Generalization is a method of extracting common attributes and behaviors from several entities to create a new, universal parent class entity. For example, a new entity "Staff" is abstracted from "Sales person" and "Administrative Executive". "Staff" is the generalization of "Sales person" and "Administrative Executive", and "Sales person" and "Administrative Executive" are the specialization of the general category "Staff".

**Aggregation**

Aggregation is a way to represent the whole and its parts. An object is a combination of several objects, e.g. a course is a combination of several course modules, each of which is part of the course.

**"Data aggregation is the process of combining, compiling and organising large volumes of data from multiple sources into one unified body. This can be done manually or using tools and software designed to collect, store and analyse data." (Chia, 2023)**

**Categorisation**

Categorisation is the process of modelling subclasses with multiple superclasses. In databases, entities can be grouped into subcategories of objects based on different attributes and criteria, each subcategory being a particular instance of the entity.Categorisation allows us to view an entity from different

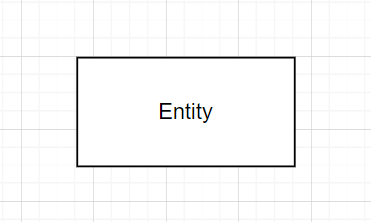
criteria to look at an entity, which can give us a more complete picture of a particular entity.

**"Categorisation is the process of dividing the world into groups of entities whose members are in some way similar to each other. " (FIRICAN, 2021)**

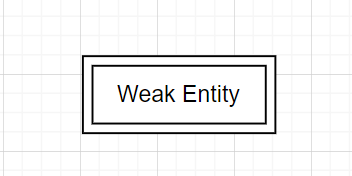
### Entities

**Entities:** Entities are the core component of conceptual modelling. Entities are used to represent objects or concepts in the real world, each entity is described by attributes to describe its information, and entities can be identified by unique attributes. In Extended Chen ERD, entities are represented by rectangles and the entity name is contained within the rectangle.

**Entity**



**Weak Entity:** Weak entities, unlike strong entities, cannot be uniquely identified by attributes. The existence of a weak entity depends on the entity and the dependent entity is called 'owner entity'. The identifier of a weak entity is a combination of the owner entity and the partial key of the weak entity.



### Attributes

**Attributes:** Attributes represent the characteristics and information of an entity, attributes include various types including Key attribute, Partial key attribute, Multivalued attribute, Derived attribute. In Extended Chen ERD, attributes are represented using oval. In Extended Chen ERD, attributes are represented using oval, and the attribute name is contained in the oval.

**Attribute**

图示, 维恩图

描述已自动生成

**Key attribute**

人的脸

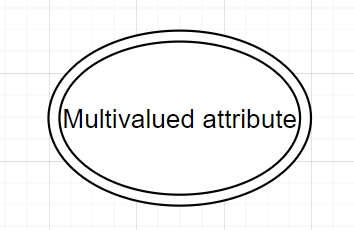
低可信度描述已自动生成

**Partial key attribute**

图示, 维恩图

描述已自动生成

**Multivalued attribute**



**Derived attribute**

图示

描述已自动生成

**Composite attribute**

图示

描述已自动生成

### Keys

**Candidate Keys:** Candidate keys are attributes/attribute combinations used to uniquely identify other records in a database management system.

### Relationships

**Relationship:** Indicates a relationship between different entities. Including "one to one", "one to many", "many to one", "many to many"

One to one

**图表, 折线图

描述已自动生成**

**One to many**

**门上的瓷砖

中度可信度描述已自动生成**

**Many to one**

**图表, 折线图

描述已自动生成**

**Many to many**

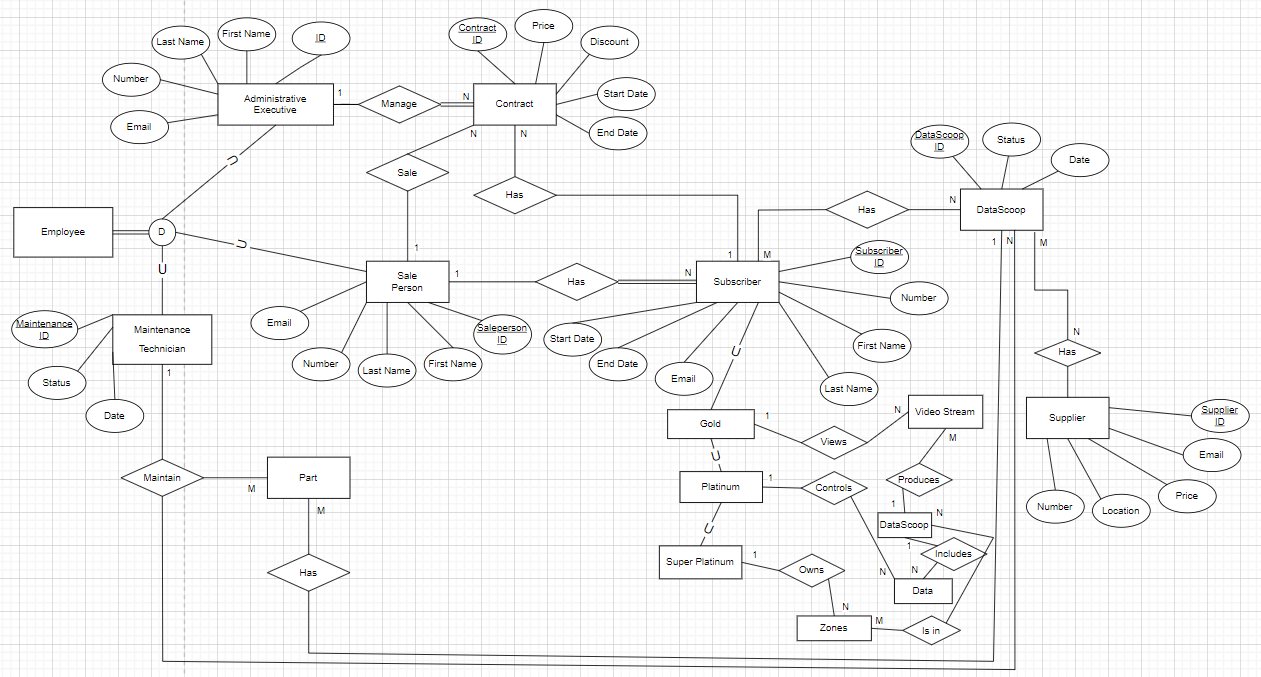
图表, 折线图

描述已自动生成

**Weak Entities and their Identifying Relationships:** weak entities depend on the existence of other entities for their existence. For aggregation relationships, if if the entity on which the weak entity depends disappears, then the weak entity will also cease to exist.

### Relationship Degrees

## Chen Conceptual Entity Relationship Model



## Reasoning and Purpose of all Parts of the Model

“There are three different types of data models: conceptual, logical and physical, and each has a specific purpose.

* Conceptual Data Models: High-level, static business structures and concepts
* Logical Data Models: Entity types, data attributes and relationships between entities
* Physical Data Models: The internal schema database design.” (Tharpe, 2020)

**The relationship between Conceptual, Logical data modelling and Physical implementation**

Conceptual data modelling, logical data modelling, and physical implementation are the three main phases of data modelling, which are carried out in sequence, indicating that database design progressively moves from the abstract to the concrete.

**Conceptual data modelling:** conceptual data modelling is the stage with the highest level of abstraction in data modelling, this stage of data modelling focuses on business requirements and high-level concepts. Conceptual modelling represents entities, attributes, inter-entity relationships at the abstraction level and defines business requirements. Conceptual modelling helps data analysts and system analysts to better understand the framework and business requirements of the system.

**Logical Data Modelling:** Logical data modelling is the progression of conceptual data modelling, this phase is tasked with further defining the data structure and logic. This phase focuses on defining entity attributes, primary keys, foreign keys, inter-entity relationships, and constraints, but this phase does not address specific data storage methods and details. The logical model usually provides a more concrete model/blueprint for developers and analysts, providing developers with more specific implementation guidance.

Physical implementation: Physical implementation is the actual implementation and operational phase of data modelling. This phase translates concrete data structures into actual, detailed technical details such as database operations.

In summary, all three phases, from conceptual data modelling to logical data modelling to physical implementation, build on the previous phase (except for conceptual data modelling). These three phases of progression are characterized by a gradual shift from abstraction to concreteness, with less and less abstraction. From high-level concepts and business requirements to concrete database structures and relationships, and finally from translating the design into actual database operations. This is a concrete manifestation of the gradual transformation of abstract models into realistic business requirements. The reliability and availability of a data system can be ensured through these three main phases of data modelling.

## Data Dictionary

### Entities

**Table 1: Document Entities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity Name** | **Description** | **Aliases** | **Occurrence** |
| Data Scoop | The product of Flight Stream |  | 46 |
| Contract | The contract of customer orders the paid service. |  | 24 |
| Subscriber | The customer who orders the premium service. |  | 21 |
| Maintenance Record | The record of maintenance Data Scoop |  | 3 |
| Supplier | The person who apply the part of Data Scoop |  | 4 |
| Zones | Areas where Data scoop can be deployed |  | 5 |
| Salesperson | The employee who sale the subscription |  | 5 |
| Administrative Executive | The employee who manage the company |  | 2 |
| Data | The data of Data Scoop |  | 80 |
| Gold | The |  | 3 |
| Platinum | The second highest level of service |  | 8 |
| Super Platinum | The highest level of service |  | 5 |
| Part | The component of Data Scoop |  | 10 |
|  |  |  |  |

### Relationships

**Table 2: Document Relationships**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Cardinality** | **Participation** | **Relationship** | **Participation** | **Cardinality** | **Entity Name** |
| Administrative  Executive | 1 | Partial | Manage | Partial | N | Contract |
| Contract | N | Total | Has | Total | a | Subscriber |
| Sale Person | 1 | Partial | Sale | Total | N | Contract |
| Sale Person | 1 | Partial | Has | Partial | N | Subscriber |
| Maintenance Technician | 1 | Partial | Maintain | Total | N | Data Scoop |
| Maintenance Technician | 1 | Partial | Has | Partial | M | Part |
| Subscriber | M | Partial | Has | Patial | N | Data Scoop |
| Supplier | N | Partial | supply | Partial | M | Data Scoop |
| Gold | 1 | Partial | Views | Partial | N | Video Stream |
| Platinum | 1 | Partial | Controls | Partial | N | Data |
| Super Platinum | 1 | Partial | Owns | Partial | N | Zones |

### Attributes

**Table 3: Document Attributes**

| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Administrative  Executive | ID | The unique identifier for an Administrative Executive. | Int |  |  |  | NOT NULL | PK |  |
|  | First Name | The first name of the Administrative Executive. | String |  |  |  | NOT NULL |  |  |
|  | Last Name, | The last name of the Administrative Executive. | String |  |  |  | NOT NULL |  |  |
|  | Number | The contact number of the Administrative Executive. | String |  |  |  | NOT NULL |  |  |
|  | Email | The email address of the Administrative Executive. | String |  |  |  | NOT NULL |  |  |
| **Sale Person** | ID | The unique identifier of the Sale Person | Int |  |  |  | NOT NULL | PK |  |
|  | First Name | The First Name of the Sale Person | String |  |  |  | NOT NULL |  |  |
|  | Last Name, | The Last Name of the Sale Person | String |  |  |  | NOT NULL |  |  |
|  | Number | The Contact Number of the Sale Person | String |  |  |  | NOT NULL |  |  |
|  | Email | The email address of the Sale Person | String |  |  |  | NOT NULL |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **Maintenance Technician** | ID | The unique identifier of Maintenance | Int |  |  |  | NOT NULL |  |  |
|  | Status | Current maintenance status | String |  |  |  | NOT NULL |  |  |
|  | Date | The date of last time maintenance |  |  |  |  | NOT NULL |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **Subscriber** | ID | The unique identifier of the Subscriber | Int |  |  |  | NOT NULL | Pk |  |
|  | First Name | The First Name of the Subscriber | String |  |  |  | NOT NULL |  |  |
|  | Last Name, | The Last Name of the Subscriber | String |  |  |  | NOT NULL |  |  |
|  | Number | The Contact Number of the Subscriber | String |  |  |  | NOT NULL |  |  |
|  | Email | The email address of the Subscriber | String |  |  |  | NOT NULL |  |  |
|  | Start Date | Subscription start date | Date |  |  |  | NOT NULL |  |  |
|  | End Date | Subscription end date | Date |  |  |  | NOT NULL |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **Data Scoop** | ID | The unique identifier of Data Scoop | Int |  |  |  | NOT NULL | PK |  |
|  | Status | Current Date Scoop working status | String |  |  |  | NOT NULL |  |  |
|  | Date | Current Date | Date |  |  |  | NOT NULL |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **Supplier** | ID | The unique identifier of Supplier | Int |  |  |  | NOT NULL | PK |  |
|  | Price | The price of the part | Int |  |  |  | NOT NULL |  |  |
|  | Location | The Supplier’s location | String |  |  |  | NOT NULL |  |  |
|  | Email | The Email of Supplier | String |  |  |  | NOT NULL |  |  |
|  | Number | The Contact Number of Supplier | Int |  |  |  | NOT NULL |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **Contract** | ID | The unique identifier of Contract | Int |  |  |  | NOT NULL | PK |  |
|  | Price | The price of Contract | Int |  |  |  | NOT NULL |  |  |
|  | Discount | The discount of Contract | Int |  |  |  | NOT NULL |  |  |
|  | Email | The email of Contract customer | String |  |  |  | NOT NULL |  |  |
|  | Number | The contact number of Contract customer | int |  |  |  | NOT NULL |  |  |
|  | Start Date | Subscription start date | Date |  |  |  | NOT NULL |  |  |
|  | End Date | Subscription end date | Date |  |  |  | NOT NULL |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

## Assumed Business Rules (with reasoning)

**INTRODUCTION**

In modern database design and development, how to manage data effectively is an important consideration that cannot be ignored, and business rules are an important part of managing data effectively in an information system. Business rules define how to manage, store and utilise data effectively, and designing suitable business rules plays an important role in the performance, security scalability and availability of the system.

**Assumptions about business rules**

**Role authority assumption:** All roles accessing database data need to go through a strict authentication process to ensure data security.

**Data Input Assumptions:** All data inputs must comply with the requirements of data types, data formats, etc.. All data input should go through a strict validation process to avoid incorrect input leading to data confusion or system errors.

**Data security assumption:** all data in the database and system should have multiple reliable protection measures to ensure data security.

**Reasons for making assumptions**

**Role Permission Assumption:** Each different role has different access rights to the database, and each role's permissions are monitored and restricted to prevent permissions from being overstepped and data from being illegally accessed and modified as a way to protect data security.

**Data Entry Assumptions:** Ensuring accurate data entry ensures the accuracy and reliability of business data, which is extremely important in certain important information areas, such as company customer information, national confidential document repositories, and so on.

**Data Security Assumptions:** The security of information systems and databases is a top priority, and data security must be ensured to reduce the likelihood of data leakage or unauthorised access, and a variety of effective protection measures need to be deployed and applied to the data.

**How Data Management Fundamentals Apply to Business Rules Assumptions**

In order to achieve the above assumptions about business rules, it is indispensable to apply the basic principles of data management.

Data integrity, accuracy and consistency: Data integrity and data accuracy are closely related to the "data entry assumptions" in the business rules. The data system through strict validation strategies to ensure that all data input is correct and accurate, if the input is wrong or irregular data, the system will appear warning and refuse to write the data into the database, which ensures data integrity, data accuracy and data consistency.

**Data security:** Data security is related to the "data security assumptions" and "role authority assumptions" in the business rules. Through the authentication process to confirm the identity and rights of visitors, through the rights restrictions and encryption technology to ensure data security, to prevent data leakage and invasion. In addition, setting up a strong security system, deploying security facilities and designing security decisions are also important measures to protect data security.

“Data security is a combination of processes and tools that aim to protect an organization’s sensitive assets. Valuable data must be protected both at rest and in transit. Security officers should also consider other vectors of data security such as safe data creation and use.” (Ekran, 2022)

# MileStone 2

## Introduction

The purpose of these tasks is to transform the Conceptual Moedl created in the previous assessment into a logical model. I will complete this assessment by following the mapping rules to create an ERD, creating a data dictionary, and performing a NaLER analysis on the ERD. A report will also be created for the above work.

## Conceptual to Logical

## Chens ERD Notation

Transforming a Conceptual model to logical model is an important part of database design. In database design, Conceptual model means high level concepts and relationships. For relational logical model, the purpose is describing data table, field, and relationship. These are some steps that transforming conceptual model to logical model.

1. **Review Conceptual Model:** Review Conceptual Model to understand and identify all entity and relationships.
2. **Identify Entity and Attributes:** Identify all Entities and Attributes. These entities and attributes will be included in Logical Model. In logical model, each entity will be transformed to table, and each attribute will be transformed to column of table.
3. **Define Primary key:** Assign primary key to each entity. Every table should have a primary key. The primary key is a uniquely identifier an entity instance.
4. **Define Relationship:** Define relationship between each entity. In Conceptual model, relationships are represented using diamonds. In Logical model, relationships are represented using foreign keys.
5. **Define data type:** Assign suitable data type to each attribute.
6. **Normalization:** Follow normalization principles to normalize Logical model to ensure data completeness and consistency. (At lease 3NF)
7. **Create Logical ERD:** Create Logical ERD, which should includetables, columns, primary keys, foreign keys, and data types.

## Mapping Rules

|  |  |  |
| --- | --- | --- |
| Conceptual Model | Logical Model | Rules |
| Entity | Table | Each entity is mapped to a table. |
| Attribute | Column | Each property is mapped to a column. |
| Primary Key | Primary Key | The primary key of the entity becomes the primary key of the table. |
| One to One | Foreign Key | Add the primary key of one entity as a foreign key to another entity. |
| One to Many | Foreign Key | Add the primary key of the "one" side entity as a foreign key to the "many" side entity. |
| Many to Many | Relationship Table | Create a new relationship table containing the primary keys of both entities. (Composite Primary Key) |

1. **Entity Mapping**

**Every Entity mapped to a table.**

Employee -> Table Employee

Maintenance Technician -> Table MaintenanceTechnician

Sale Person -> Table SalePerson

Administrative Executive -> Table AdministrativeExecutive

Maintenance Records -> Table MaintenanceRecords

Subscriber -> Table Subscriber

Contract -> Table Contract

Gold -> Table Gold

Platinum -> Table Platinum

Super Platinum -> Table Super Platinum

Video Stream -> Table VideoStream

DataScoop -> Table DataScoop

Zones -> Table Zones

Data -> Table Data

Part -> Table Part

Supplier -> Table Supplier

1. **Attribute Mapping**

**Maintenance Technician**

* MaintenanceID -> column MaintenanceID
* Email -> column Email
* Number -> column Number
* LastName -> column LastName
* Date -> column Date
* Status -> column Status

**Sale Person**

* SalespersonID -> column SalespersonID
* FirstName -> column FirstName
* LastName -> column LastName
* Email -> column Email
* Number -> column Number
* StartDate -> column StartDate
* EndDate -> column EndDate

**Administrative Executive**

* ID -> column ID
* Number -> column Number
* FirstName -> column FirstName
* LastName -> column LastName
* Email -> column Email

**Subscriber**

* SubscriberID -> column SubscriberID
* FirstName -> column FirstName
* LastName -> column LastName
* Email -> column Email
* Number -> column Number
* Type -> column Type

**Contract**

* ContractID -> column ContractID
* Price -> column Price
* StartDate -> column StartDate
* EndDate -> column EndDate
* Discount -> column Discount

**Video Stream**

* VideoStreamID -> column VideoStreamID
* Views -> column Views

**DataScoop**

* DataScoopID -> column DataScoopID
* Date -> column Date
* Status -> column Status

**Data**

* DataID -> column DataID
* Type -> column Type

**Zones**

* ZoneID -> column ZoneID
* Name -> column Name

**Maintenance Records**

* RecordID -> column RecordID
* MaintainDate -> column MaintainDate

**Part**

* PartID -> column PartID
* Location -> column Location
* Price -> column Price

**Supplier**

* SupplierID -> column SupplierID
* Email -> column Email
* Number -> column Number
* Status -> column Status

1. **Primary Key Mapping**

The primary key of each entity becomes the primary key of the corresponding relationship.

1. **Relationship Mapping**

**One-to-one relationship:** add the primary key of any entity as a foreign key to the other entity.

**One-to-many relationship:** add the primary key of the "one" end entity as a foreign key to the "many" end entity.

**Many-to-many relationship:** create a new relationship table to connect the two tables, and this table uses the primary keys of the two entities as a composite primary key.

## Normalisation

## First Normal Form

**For 1NF, the most important thing is to ensure that the value of each cell in each table is independent and atomic.**

A relation belongs to 1NF if it meets the following requirements.

1. Each row in the table is unique

2. The intersection of each row and column (cell) in the table can only have one independent value (i.e. no multiple values)

3. The values ​​of all fields are atomic (i.e. indivisible)

## Second Normal Form

**The most important thing for 2NF is that all non-primary key attributes must be completely dependent on the primary key, and all partial key dependencies need to be eliminated.**

If a relation is in 1NF and meets the above requirements, it is in 2NF.

## Third Normal Form

**For 3NF, the most important thing is that each relation follows Non-Transitive Dependencies, and all Non Transitive Dependencies need to be eliminated.**

### Boyce-Codd Normal Form

”Boyce–Codd Normal Form (BCNF) is based on [functional dependencies](https://www.geeksforgeeks.org/functional-dependency-and-attribute-closure/) that take into account all candidate keys in a relation; however, BCNF also has additional constraints compared with the general definition of 3NF.” (Upadhyay, 2019)

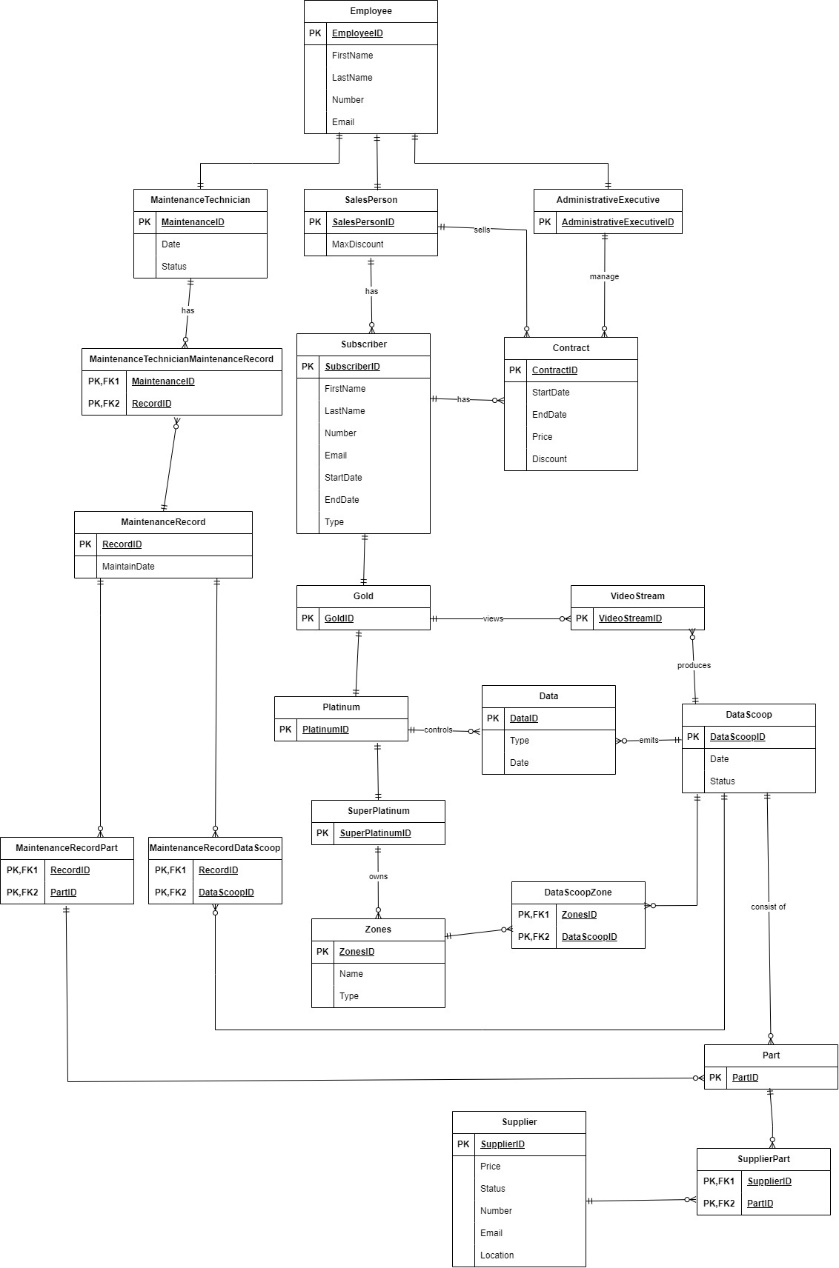
BCNF is essentially 3NF, but BCNF is stricter. If there are still multi-valued dependencies in the relationship, they must be eliminated to achieve true BCNF.

## Fourth Normal Form

“Fourth normal form (4NF) is a level of database normalization where there are no non-trivial multivalued dependencies other than a candidate key.” (Rouse, 2014)

4NF is a further development of BCNF. The most important thing is to eliminate multi-valued dependencies in relations. For example, in a relation, attribute A determines attributes B and C at the same time, and attributes B and C are independent of each other. This situation is called multi-valued dependency.

## Logical ERD



## Rational & Relations

First, I started to modify the conceptual model created in Milestone 1 using Extended Chen Notation according to the feedback from my mentor.

Next, I checked all entities, attributes and relationships.

Then, I drew all the identified entities, attributes and relationships into a logical ERD according to the mapping rules. For example, I used rectangular boxes to represent entities, used PK and FK in front of fields to represent primary keys and foreign keys, and created a data dictionary for the created logical model.

Then, I performed normalization and NaLER analysis (some sample values ​​were generated for the logical table in this phase) to make the logical design more reasonable and efficient.

## Data Dictionary

## Entities

**Document relations**

|  |  |  |  |
| --- | --- | --- | --- |
| **Relation Name** | **Start Volume** No. of rows loaded at the beginning | **Growth** e.g. no growth / 10% per year | **Comments** |
| Employee | 150 | 10% | All Employees |
| MaintenanceTechnician | 20 | 5% | Technicians of employee |
| SalesPerson | 100 | 5% | SalesPerson of employee |
| AdministrativeExecutive | 30 | 5% | Administrative staff of company Employee |
| MaintenanceTechnicianMaintenanceRecord | 200 | 10% | Relation between MaintenanceTechnician and MaintenanceRecord |
| Subscriber | 200 | 20% | Subscribed customers |
| Contract | 100 | 10% | Customer contracts |
| MaintenanceRecord | 100 | 10% | Maintenance records |
| Gold | 100 | 5% | Gold level Subscriber |
| Platinum | 100 | 5% | Platinum level Subscriber |
| SuperPlatinum | 50 | 5% | Super platinum level Subscriber |
| Data | 500 | 15% | The data that DataScoop collect. |
| DataScoop | 200 | 10% | Drone Data scoops |
| Zones | 20 | 5% | Deploy DataScoop Zones |
| DataScoopZone | 100 | 10% | Relation between Datascoop and Zones |
| MaintenanceRecordPart | 300 | 15% | Relation between Part and MaintenanceRecord |
| MaintenanceRecordDataScoop | 150 | 10% | Relation between Datascoop and MaintenanceRecord. |
| Part | 400 | 10% | Different parts used in DataScoop |
| Supplier | 50 | 5% | Suppliers providing parts |
| SupplierPart | 200 | 10% | Relation between suppliers and parts |

## Attributes

**Document Attributes**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Relation Name** | **Attribute** | **Description** | **Data type** | **Length** | **Value range** | **Validation Rules** | **Default Value** | **Nulls** | **Key?** | **References Entity** | **Integrity Constraints** |
| Employee | EmployeeID | The unique ID of  Employee | Int |  | 1 to 2147483647 | Unique |  | No | PK |  | PK |
|  | FirstName | The First Name of Employee. | String | 50 |  |  |  | No |  |  |  |
|  | LastName | The Last Name of Employee. | String | 50 |  |  |  | No |  |  |  |
|  | Number | The phone number of Employee. | Int |  |  |  |  | No |  |  |  |
|  | Email | The email of Employee. | String | 30 |  | Must be valid email format |  | No |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| MaintenanceTechnician | MaintenanceID | The unique ID of Maintenance Technician | Int |  | 1 to 2147483647 | Unique |  | No | PK |  | PK |
|  | Date | The date of maintenance | Date |  |  |  |  | No |  |  |  |
|  | Status | The status of the maintenance | String | 50 |  |  |  | No |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| SalesPerson | SalesPersonID | The unique ID of Sales Person. | Int |  | 1 to 2147483647 | Unique |  | No | PK |  | PK |
|  | MaxDiscount | Maximum discount that Sales Person allowed | Int |  | 0 to 5 |  |  | Yes |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| AdministrativeExecutive | AdministrativeExecutiveID | The unique ID of Administrative Executive | Int |  | 1 to 2147483647 | Unique |  | No | PK |  | PK |
|  | rrrr |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| MaintenanceTechnicianMaintenanceRecord | MaintenanceID | The unique ID of | Int |  | 1 to 2147483647 | Unique |  | No | PK | MaintenanceTechnician | PK, FK |
|  | RecordID |  | Int |  | 1 to 2147483647 | Unique |  | No | PK | MaintenanceRecord | PK, FK |
|  |  |  |  |  |  |  |  |  |  |  |  |
| Subscriber | SubscriberID | The unique ID of Subscriber | Int |  | 1 to 2147483647 | Unique |  | No | PK |  | PK |
|  | FirstName | Subscriber's first name | String | 50 |  |  |  | No |  |  |  |
|  | LastName | Subscriber's last name | String | 50 |  |  |  | No |  |  |  |
|  | Number | Subscriber's contact number | Int | 20 | 1 to 2147483647 |  |  | No |  |  |  |
|  | Email | Subscriber's email address | String | 50 |  |  |  | No |  |  |  |
|  | StartDate | Subscription start date | Date |  |  |  |  | No |  |  |  |
|  | EndDate | Subscription end date | Date |  |  |  |  | No |  |  |  |
|  | Type | Subscription type | String | 20 |  |  |  | No |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| Contract | ContractID | The unique ID of Contract | Int |  | 1 to 2147483647 | Unique |  | No | PK |  | PK |
|  | StartDate | Contract start date | Date |  |  |  |  | No |  |  |  |
|  | EndDate | Contract end date | Date |  |  |  |  | No |  |  |  |
|  | Price | Contract price | Float | 20 |  |  |  | No |  |  |  |
|  | Discount | Contract discount | Int | 20 |  |  |  | Yes |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| MaintenanceRecord | RecordID | The unique ID of Maintenance Record | Date | 20 | 1 to 2147483647 | Unique |  | No | PK |  | PK |
|  | MaintainDate | Maintenance date | Int |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| Gold | GoldID | The unique ID of Gold | Int |  | 1 to 2147483647 | Unique |  | No | PK |  | PK |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| Platinum | PlatinumID | The unique ID of Platinum | Int |  | 1 to 2147483647 | Unique |  | No | PK |  | PK |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| SuperPlatinum | SuperPlatinumID | The unique ID of Super Platinum | Int |  | 1 to 2147483647 | Unique |  | No | PK |  | PK |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| VideoStream | VideoStreamID | The unique ID of Video Stream | Int |  | 1 to 2147483647 | Unique |  | No | PK |  | PK |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| Data | DataID | The unique ID of Data | Int |  | 1 to 2147483647 | Unique |  | No | PK |  | PK |
|  | Type | Data type | String | 20 |  |  |  | No |  |  |  |
|  | Date | Data date | Date |  |  |  |  | No |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| DataScoop | DataScoopID | The unique ID of Data | Int |  | 1 to 2147483647 | Unique |  | No | PK |  | PK |
|  | Date | Data Scoop date | Date |  |  |  |  | No |  |  |  |
|  | Status | Data Scoop status | String | 50 |  |  |  | No |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| Zones | ZonesID | The unique ID of Zone | Int |  | 1 to 2147483647 | Unique |  | No | PK |  | PK |
|  | Name | Zone name | String | 50 |  |  |  | No |  |  |  |
|  | Type | Zone type | String | 20 |  |  |  | No |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| DataScoopZone | ZonesID | Foreign Key to Zone | Int |  | 1 to 2147483647 | Unique |  | No | PK | Zones | PK, FK |
|  | DataScoopID | Foreign Key to Data Scoop | Int |  | 1 to 2147483647 | Unique |  | No | PK | DataScoop | PK, FK |
|  |  |  |  |  |  |  |  |  |  |  |  |
| MaintenanceRecordPart | RecordID | Foreign key referencing MaintenanceRecord | Int |  | 1 to 2147483647 | Unique |  | No | PK | MaintenanceRecord | PK, FK |
|  | PartID | Foreign key referencing Part Entity | Int |  | 1 to 2147483647 | Unique |  | No | PK | Part | PK, FK |
|  |  |  |  |  |  |  |  |  |  |  | PK |
| MaintenanceRecordDataScoop | RecordID | Foreign key referencing MaintenanceRecord | Int |  | 1 to 2147483647 | Unique |  | No | PK | MaintenanceRecord | PK, FK |
|  | DataScoopID | Foreign key referencing DataScoop | Int |  | 1 to 2147483647 | Unique |  | No | PK | DataScoop | PK, FK |
|  |  |  |  |  |  |  |  |  |  |  |  |
| Part | PartID | The unique ID of Part | Int |  | 1 to 2147483647 | Unique |  | No | PK |  | PK |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| Supplier | SupplierID | The unique ID of Supplier | Int |  | 1 to 2147483647 | Unique |  | No | PK |  | PK |
|  | Status | Supplier status | String | 50 |  |  |  | No |  |  |  |
|  | Number | Supplier contact number | Int |  |  |  |  | No |  |  |  |
|  | Email | Supplier email | String | 30 |  |  |  | No |  |  |  |
|  | Location | Supplier location | String | 50 |  |  |  | No |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| SupplierPart | SupplierID | Foreign key referencing Supplier | Int |  | 1 to 2147483647 | Unique |  | No | PK | Supplier | PK, FK |
|  | PartID | Foreign key referencing Part | Int |  |  | Unique |  | No | PK | Part | PK, FK |

## NaLER Analysis

## STEP 1. Identify and document the diagram conventions

1. **Entity - Use a rectangular box to represent**

**Example:**

* **Employee**
* **MaintenanceTechnician**
* **SalesPerson**
* **AdministrativeExecutive**
* **Subscriber**
* **Contract**
* **MaintenanceRecord**
* **Gold**
* **Platinum**
* **Data**
* **DataScoop**
* **Zones**
* **Part**
* **Supplier**

表格

低可信度描述已自动生成

1. **Primary Key - Fields are marked with "PK"**

**Example:**

* **Employee: EmployeeID**
* **MaintenanceTechnician: MaintenanceID**
* **SalesPerson: SalesPersonID**
* **AdministrativeExecutive: AdministrativeExecutiveID**
* **Subscriber: SubscriberID**
* **Contract: ContractID**
* **MaintenanceRecord: RecordID**
* **Gold: GoldID**
* **Platinum: PlatinumID**
* **Data: DataID**
* **DataScoop: DataScoopID**
* **Zones: ZonesID**
* **Part: PartID**
* **Supplier: SupplierID**

表格

描述已自动生成

1. **Foreign Key - Fields are marked with "FK"**

**Example:**

* **MaintenanceTechnicianMaintenanceRecord: MaintenanceID, RecordID**
* **MaintenanceRecordPart: RecordID, PartID**
* **MaintenanceRecordDataScoop: RecordID, DataScoopID**
* **SupplierPart: SupplierID, PartID**
* **DataScoopZone: ZonesID, DataScoopID**

表格

描述已自动生成

1. **Relationship – Use connecting lines to connect entities to represent and make notes on the connecting lines**

**Example:**

* **The has relationship between Employee and MaintenanceTechnician**
* **The has relationship between Employee and SalesPerson**
* **The has relationship between Employee and AdministrativeExecutive**
* **The has relationship between MaintenanceTechnician and MaintenanceRecord**
* **The sells relationship between SalesPerson and Contract**
* **The has relationship between Subscriber and Contract**
* **The views relationship between Gold and Data**
* **The controls relationship between Platinum and Data**
* **The owns relationship between SuperPlatinum and Zones**
* **The emits relationship between DataScoop and Data**
* **The consists of relationship between DataScoopZone and DataScoop and Zones**
* **The has relationship between Supplier and Part**

图片包含 箱线图

描述已自动生成

**Entities:**

**Employee**

**Employee**

**Primary key: EmployeeID**

**Attributes: FirstName, LastName, Number, Email**

**Relationship: has relationship - MaintenanceTechnician, SalesPerson, AdministrativeExecutive**

**MaintenanceTechnician**

**Primary key: MaintenanceID (PK)**

**Attributes: Date, Status**

**Relationship: has relationship - MaintenanceRecord, MaintenanceTechnicianMaintenanceRecord**

**SalesPerson**

**Primary key: SalesPersonID (PK)**

**Attributes: MaxDiscount**

**Relationship: sells relationship - Contract**

**AdministrativeExecutive**

**Primary key: AdministrativeExecutiveID (PK)**

**Attributes: None**

**Relationship: manage relationship - Subscriber**

**Subscriber**

**Primary key: SubscriberID (PK)**

**Attributes: FirstName, LastName, Number, Email, StartDate, EndDate, Type**

**Relationship: has relationship - Contract**

**Contract**

**Primary key: ContractID (PK)**

**Attributes: StartDate, EndDate, Price, Discount**

**Relationship: None**

**MaintenanceRecord**

**Primary key: RecordID (PK)**

**Attribute: MaintainDate**

**Relationship: None**

**Gold**

**Primary key: GoldID (PK)**

**Attribute: None**

**Relationship: views relationship – Data**

**Platinum**

**Primary key: PlatinumID (PK)**

**Attribute: None**

**Relationship: controls relationship – Data**

**Super Platinum**

**Primary key: SuperPlatinumID (PK)**

**Attribute: None**

**Relationship: owns relationship – Zones**

**Data**

**Primary key: DataID (PK)**

**Attribute: Type, Date**

**Relationship: emits relationship – DataScoop**

**DataScoop**

**Primary key: DataScoopID (PK)**

**Attribute: Date, Status**

**Relationship: consists of relationship – DataScoopZone**

**Zones**

**Primary key: ZonesID (PK)**

**Attribute: Name, Type**

**Relationship: owns relationship – SuperPlatinum**

**Part**

**Primary key: PartID (PK)**

**Attributes: None**

**Relationship: None**

**Supplier**

**Primary key: SupplierID (PK)**

**Attributes: Price, Status, Number, Email, Location**

**Relationship: has relationship – Part**

## STEP 2. Perform a syntax check of the model in accordance with the step 1 assumptions and make and mark any adjustments

### Check the name of Entity and Attribute

**Employee: The name is correct and the attributes are correct.**

**MaintenanceTechnician: The name is correct and the attributes are correct.**

**SalesPerson: The name is correct and the attributes are correct.**

**AdministrativeExecutive: The name is correct and the attributes are correct.**

**Subscriber: The name is correct and the attributes are correct.**

**Contract: The name is correct and the attributes are correct.**

**MaintenanceRecord: The name is correct and the attributes are correct.**

**Gold: The name is correct and the attributes are correct.**

**Platinum: The name is correct and the attributes are correct.**

**SuperPlatinum: The name is correct and the attributes are correct.**

**Data: The name is correct and the attributes are correct.**

**DataScoop: The name is correct and the attributes are correct.**

**Zones: The name is correct and the attributes are correct.**

**Part: The name is correct and the attributes are correct.**

**Supplier: The name is correct and the attributes are correct.**

**MaintenanceTechnicianMaintenanceRecord: The name is correct and the attributes are correct.**

**MaintenanceRecordPart: The name is correct and the attributes are correct.**

**MaintenanceRecordDataScoop: The name is correct and the attributes are correct.**

**SupplierPart: The name is correct and the attributes are correct.**

**DataScoopZone: The name is correct and the attributes are correct.**

### Check Primary Key

**Employee: EmployeeID (PK)**

**MaintenanceTechnician: MaintenanceID (PK)**

**SalesPerson: SalesPersonID (PK)**

**AdministrativeExecutive: AdministrativeExecutiveID (PK)**

**Subscriber: SubscriberID (PK)**

**Contract: ContractID (PK)**

**MaintenanceRecord: RecordID (PK)**

**Gold: GoldID (PK)**

**Platinum: PlatinumID (PK)**

**SuperPlatinum: SuperPlatinumID (PK)**

**Data: DataID (PK)**

**DataScoop: DataScoopID (PK)**

**Zones: ZonesID (PK)**

**Part: PartID (PK)**

**Supplier: SupplierID (PK)**

**MaintenanceTechnicianMaintenanceRecord: MaintenanceID (PK, FK1), RecordID (PK, FK2)**

**MaintenanceRecordPart: RecordID (PK, FK1), PartID (PK, FK2)**

**MaintenanceRecordDataScoop: RecordID (PK, FK1), DataScoopID (PK, FK2)**

**SupplierPart: SupplierID (PK, FK1), PartID (PK, FK2)**

**DataScoopZone: ZonesID (PK, FK1), DataScoopID (PK, FK2)**

### Check Foreign Key

**MaintenanceTechnicianMaintenanceRecord:**

**MaintenanceID (FK1) references the primary key of MaintenanceTechnician.**

**RecordID (FK2) references the primary key of MaintenanceRecord.**

**MaintenanceRecordPart:**

**RecordID (FK1) references the primary key of MaintenanceRecord.**

**PartID (FK2) references the primary key of Part.**

**MaintenanceRecordDataScoop:**

**RecordID (FK1) references the primary key of MaintenanceRecord.**

**DataScoopID (FK2) references the primary key of DataScoop.**

**SupplierPart:**

**SupplierID (FK1) references the primary key of Supplier.**

**PartID (FK2) references the primary key of Part.**

**DataScoopZone:**

**ZonesID (FK1) references the primary key of Zones.**

**DataScoopID (FK2) references the primary key of DataScoop.**

### Check Relationship

**Employee has a has relationship with MaintenanceTechnician, SalesPerson, and AdministrativeExecutive: Correct.**

**SalesPerson has a sells relationship with Contract: Correct.**

**AdministrativeExecutive has a manage relationship with Contract: Correct.**

**Subscriber has a has relationship with Contract: Correct.**

**Gold has a views relationship with Data: Correct.**

**Platinum has a controls relationship with Data: Correct.**

**SuperPlatinum has an owns relationship with Zones: Correct.**

**Data has an emits relationship with DataScoop: Correct.**

**DataScoopZone has a consist of relationship with DataScoop and Zones: Correct.**

**Supplier has a has relationship with Part: Correct.**

### Check composite Entity

**MaintenanceTechnicianMaintenanceRecord: Contains MaintenanceID (FK1) and RecordID (FK2), which uniquely identify each instance.**

**MaintenanceRecordPart: Contains RecordID (FK1) and PartID (FK2), which uniquely identify each instance.**

**MaintenanceRecordDataScoop: Contains RecordID (FK1) and DataScoopID (FK2), which uniquely identify each instance.**

**SupplierPart: Contains SupplierID (FK1) and PartID (FK2), which uniquely identify each instance.**

**DataScoopZone: Contains ZonesID (FK1) and DataScoopID (FK2), which uniquely identify each instance.**

## Step 3.1 & 3.2 - Attribute Sentences

1. Each Employee is uniquely identified by EmployeeID.
2. One Employee identified by EmployeeID must have one FirstName.
3. One Employee identified by EmployeeID must have one LastName.
4. One Employee identified by EmployeeID must have one Number.
5. One Employee identified by EmployeeID must have one Email.
6. Each MaintenanceTechnician is uniquely identified by MaintenanceID.
7. One MaintenanceTechnician identified by MaintenanceID must have one Date.
8. One MaintenanceTechnician identified by MaintenanceID must have one Status.
9. Each SalesPerson is uniquely identified by SalesPersonID.
10. One SalesPerson identified by SalesPersonID must have one MaxDiscount.
11. Each AdministrativeExecutive is uniquely identified by AdministrativeExecutiveID.
12. Each Subscriber is uniquely identified by SubscriberID
13. One Subscriber identified by SubscriberID must have one FirstName.
14. One Subscriber identified by SubscriberID must have one LastName.
15. One Subscriber identified by SubscriberID must have one Number.
16. One Subscriber identified by SubscriberID must have one Email.
17. One Subscriber identified by SubscriberID must have one StartDate.
18. One Subscriber identified by SubscriberID must have one EndDate.
19. One Subscriber identified by SubscriberID must have one Type.
20. Each Contract is uniquely identified by ContractID.
21. One Contract identified by ContractID must have one StartDate.
22. One Contract identified by ContractID must have one EndDate.
23. One Contract identified by ContractID must have one Price.
24. One Contract identified by ContractID must have one Discount.
25. Each MaintenanceRecord is uniquely identified by RecordID.
26. One MaintenanceRecord identified by RecordID must have one MaintainDate.
27. Each Gold is uniquely identified by GoldID.
28. Each Platinum is uniquely identified by PlatinumID.
29. Each SuperPlatinum is uniquely identified by SuperPlatinumID.
30. Each Data is uniquely identified by DataID.
31. One Data identified by DataID must have one Type.
32. One Data identified by DataID must have one Date.
33. Each DataScoop is uniquely identified by DataScoopID.
34. One DataScoop identified by DataScoopID must have one Date.
35. One DataScoop identified by DataScoopID must have one Status.
36. Each Zones is uniquely identified by ZonesID.
37. One Zones identified by ZonesID must have one Name.
38. One Zones identified by ZonesID must have one Type.
39. Each Part is uniquely identified by PartID.
40. Each Supplier is uniquely identified by SupplierID.
41. One Supplier identified by SupplierID must have one Price.
42. One Supplier identified by SupplierID must have one Status.
43. One Supplier identified by SupplierID must have one Number.
44. One Supplier identified by SupplierID must have one Email.
45. One Supplier identified by SupplierID must have one Location.
46. Each MaintenanceTechnicianMaintenanceRecord is uniquely identified by MaintenanceID and RecordID.
47. Each MaintenanceRecordPart is uniquely identified by RecordID and PartID.
48. Each MaintenanceRecordDataScoop is uniquely identified by RecordID and DataScoopID.
49. Each SupplierPart is uniquely identified by SupplierID and PartID.
50. Each DataScoopZone is uniquely identified by ZonesID and DataScoopID.

## Step 3.3 - Relationship Sentences

1. One Employee identified by EmployeeID may have one or more MaintenanceTechnicians identified by MaintenanceID.
2. One Employee identified by EmployeeID may have one or more SalesPersons identified by SalesPersonID.
3. One Employee identified by EmployeeID may have one or more AdministrativeExecutives identified by AdministrativeExecutiveID.
4. One SalesPerson identified by SalesPersonID may sell one or more Contracts identified by ContractID.
5. One AdministrativeExecutive identified by AdministrativeExecutiveID may manage one or more Contracts identified by ContractID.
6. One Subscriber identified by SubscriberID may have one or more Contracts identified by ContractID.
7. One Platinum identified by PlatinumID may control one or more Data identified by DataID.
8. One SuperPlatinum identified by SuperPlatinumID may own one or more Zones identified by ZonesID.
9. One Data identified by DataID may emit one or more DataScoops identified by DataScoopID.
10. One DataScoopZone identified by ZonesID and DataScoopID may consist of one or more DataScoops identified by DataScoopID.
11. One Supplier identified by SupplierID may have one or more Parts identified by PartID.
12. One MaintenanceTechnician identified by MaintenanceID may have one or more MaintenanceRecords identified by RecordID.
13. One MaintenanceRecord identified by RecordID may have one or more Parts identified by PartID.
14. One MaintenanceRecord identified by RecordID may have one or more DataScoops identified by DataScoopID.
15. One Supplier identified by SupplierID may have one or more Parts identified by PartID.
16. One DataScoop identified by DataScoopID may consist of one or more Zones identified by ZonesID.

## Step 4.1 – Examples of Sentences for Attributes

1. Employee (0000-001) is uniquely identified by EmployeeID 0000-001.
2. Employee (0000-001) has one FirstName 'Bo'.
3. Employee (0000-001) has one LastName 'Jeon'.
4. Employee (0000-001) has one Number '1234567890'.
5. Employee (0000-001) has one Email 'john.doe@example.com'.
6. MaintenanceTechnician (1111-001) is uniquely identified by MaintenanceTechnicianID 1111-001
7. MaintenanceTechnician (1111-001) has one Date '2023-05-01'.
8. MaintenanceTechnician (1111-001) has one Status 'Completed'.
9. SalesPeron (2222-001) is uniquely identified by SalesPeronID 2222-001
10. SalesPerson (2222-001) has one MaxDiscount '5%'.
11. AdministrativeExecutive (3333-001) is uniquely identified by AdministrativeExecutiveID 333-001.
12. Subscriber (4444-001) is uniquely identified by SubscriberID 4444-001.
13. Subscriber (4444-001) has one FirstName 'Alex'.
14. Subscriber (4444-001) has one LastName 'Josef'.
15. Subscriber (4444-001) has one Number '0273024824'.
16. Subscriber (4444-001) has one Email 'alex.josef@example.com'.
17. Subscriber (4444-001) has one StartDate '2024-01-01'.
18. Subscriber (4444-001) has one EndDate '2024-12-31'.
19. Subscriber (4444-001) has one Type 'Gold'.
20. Contract (5555-001) is uniquely identified by SubscriberID 5555-001.
21. Contract (5555-001) has one StartDate '2023-06-01'.
22. Contract (5555-001) has one EndDate '2023-12-31'.
23. Contract (5555-001) has one Price '$5000'.
24. Contract (5555-001) has one Discount '$250'.
25. MaintenanceRecord (6666-001) is uniquely identical by MaintenanceRecordID 6666-001
26. MaintenanceRecord (6666-001) has one MaintainDate '2023-07-15'.
27. Gold (111-001) is uniquely identified by GoldID 111-001.
28. Platinum (222-001) is uniquely identified by PlatinumID 222-111.
29. SuperPlatinum (333-001) is uniquely identified by SuperPlatinumID 333-001.
30. Data (444-001) is uniquely identified by DataID 444-001.
31. Data (444-001) has one Type 'Altitude'.
32. Data (444-001) has one Date '2023-08-01'.
33. DataScoop (555-001) is uniquely identified by DataScoopID 555-001.
34. DataScoop (555-001) has one Date '2023-08-15'.
35. DataScoop (555-001) has one Status 'Active'.
36. Zones (666-001) is uniquely identified by ZonesID 666-001.
37. Zones (666-001) has one Name 'Zone A'.
38. Zones (666-001) has one Type 'Mountain'.
39. Part (888-001) is uniquely identified by PartID 888-001.
40. Supplier (777-001) is uniquely identified by SupplierID 777-001.
41. Supplier (777-001) has one Price '$100'.
42. Supplier (777-001) has one Status 'Available'.
43. Supplier (777-001) has one Number '012345679'.
44. Supplier (777-001) has one Email 'supplier@example.com'.
45. Supplier (777-001) has one Location '101 Main St'.
46. MaintenanceTechnicianMaintenanceRecord (1111-001, 6666-001) is uniquely identified by MaintenanceID 1111-001 and RecordID 6666-001.
47. MaintenanceRecordPart (6666-001, 888-001) is uniquely identified by RecordID 6666-001 and PartID 888-001.
48. MaintenanceRecordDataScoop (6666-001, 555-001) is uniquely identified by RecordID 6666-001 and DataScoopID 555-001.
49. SupplierPart (777-001, 888-001) is uniquely identified by SupplierID 777-001and PartID 888-001.
50. DataScoopZone (666-001, 555-001) is uniquely identified by ZonesID 666-001and DataScoopID 555-001.

## Step 4.2 – Examples of Sentences for Relationships

1. Employee (0000-001) has MaintenanceTechnician (1111-001).
2. Employee (0000-001) has SalesPerson (2222-001).
3. Employee (0000-001) has AdministrativeExecutive (3333-001).
4. SalesPerson (2222-001) sells Contract (5555-001).
5. AdministrativeExecutive (1111-001) manages Contract (5555-001).
6. Subscriber (4444-001) has Contract (5555-001).
7. Platinum (222-001) controls Data (444-001).
8. SuperPlatinum (333-001) owns Zones (666-001).
9. Data (444-001) emits DataScoop (555-001).
10. DataScoopZone (666-001, 555-001) consists of DataScoop (555-001) and Zones (666-001)
11. Supplier (777-001) has Part (888-001).
12. MaintenanceTechnician (1111-001) has MaintenanceRecord (6666-001).
13. MaintenanceRecord (6666-001) has Part (888-001).
14. MaintenanceRecord (6666-001) has DataScoop (555-001).
15. Supplier (777-001) has Part (888-001).
16. DataScoop (555-001) consists of Zones (666-001).

## Conclusion

In this assessment, I learned how to convert conceptual models into logical models and wrote corresponding reports for each part of the work. In this process, mapping rules, building ERDs, normalization, and creating data dictionaries are very important for data modeling. In each step of the work, the logical data model is continuously optimized and modified to make the data model more comprehensive and reasonable; at the same time, effective data management principles are also applied, which all help to improve database work and can improve work efficiency, data security, and operational stability in actual application.

# MileStone 3

## Step 4 Translate global logical data model for target DBMS

### Step 4.1 Design base relations for target DBMS

-- Create FlightStreamDB Database

USE master;

GO

DROP DATABASE IF EXISTS FlightStreamDB;

GO

CREATE DATABASE FlightStreamDB;

GO

USE FlightStreamDB;

-- Create Table

-- Create Employee Table

CREATE TABLE Employee (

    EmployeeID INT PRIMARY KEY,

    FirstName VARCHAR(50) NOT NULL,

    LastName VARCHAR(50) NOT NULL,

    Number VARCHAR(20) NOT NULL,

    Email VARCHAR(100) NOT NULL UNiQUE

);

-- --

-- Create MaintenanceTechnician table

CREATE TABLE MaintenanceTechnician (

    MaintenanceID INT PRIMARY KEY,

    EmployeeID INT NOT NULL,

    Date DATE  NOT NULL,

    Status VARCHAR(50) DEFAULT 'Active',

    FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)

);

-- Create SalesPerson table

CREATE TABLE SalesPerson (

    SalesPersonID INT PRIMARY KEY,

    MaxDiscount DECIMAL CHECK (MaxDiscount > 0)

);

-- Create AdministrativeExecutive table

CREATE TABLE AdministrativeExecutive (

    AdministrativeExecutiveID INT PRIMARY KEY

);

-- Create MaintenanceRecord table

CREATE TABLE MaintenanceRecord (

    RecordID INT PRIMARY KEY,

    MaintainDate DATE NOT NULL

);

-- Create Subscriber table

CREATE TABLE Subscriber (

    SubscriberID INT PRIMARY KEY,

    FirstName VARCHAR(50) NOT NULL,

    LastName VARCHAR(50) NOT NULL,

    Number VARCHAR(20) NOT NULL,

    Email VARCHAR(100) NOT NULL,

    StartDate DATE NOT NULL,

    EndDate DATE NOT NULL,

    Type VARCHAR(50) NOT NULL

);

-- Create Contract table

CREATE TABLE DataScoop (

    DataScoopID INT PRIMARY KEY,

    Date DATE NOT NULL,

    Status VARCHAR(50) NOT NULL

);

CREATE TABLE Contract (

    ContractID INT PRIMARY KEY,

    StartDate DATE NOT NULL,

    EndDate DATE NOT NULL,

    Price DECIMAL(10, 2) NOT NULL,

    Discount DECIMAL(5, 2) CHECK (Discount > 0),

    OrganisationName VARCHAR(100),

    DataScoopID INT NOT NULL,

    FOREIGN KEY (DataScoopID) REFERENCES DataScoop(DataScoopID)

);

CREATE TABLE Gold (

    GoldID INT PRIMARY KEY

);

CREATE TABLE Platinum (

    PlatinumID INT PRIMARY KEY

);

CREATE TABLE SuperPlatinum (

    SuperPlatinumID INT PRIMARY KEY

);

CREATE TABLE Data (

    DataID INT PRIMARY KEY,

    Date DATE NOT NULL,

    Latitude FLOAT NOT NULL,

    Longitude FLOAT NOT NULL,

    Temperature FLOAT NOT NULL,

    Humidity FLOAT NOT NULL,

    AmbientLightStrength FLOAT NOT NULL,

    OrganicSpectralDate DATE NOT NULL

);

CREATE TABLE Zones (

    ZonesID INT PRIMARY KEY,

    Name VARCHAR(50) NOT NULL,

    Type VARCHAR(20) NOT NULL

);

CREATE TABLE Supplier (

    SupplierID INT PRIMARY KEY,

    Name VARCHAR(100) NOT NULL,

    Price DECIMAL(10, 2) NOT NULL,

    Status VARCHAR(50) NOT NULL,

    Number VARCHAR(20) NOT NULL,

    Email VARCHAR(100) NOT NULL,

    Location VARCHAR(100) NOT NULL

);

CREATE TABLE Part (

    PartID INT PRIMARY KEY,

    Name VARCHAR(100) NOT NULL

);

CREATE TABLE VideoStream (

    VideoStreamID INT PRIMARY KEY

);

/\* Connect table \*/

CREATE TABLE MaintenanceTechnicianMaintenanceRecord (

    MaintenanceID INT,

    RecordID INT,

    PRIMARY KEY (MaintenanceID, RecordID),

    FOREIGN KEY (MaintenanceID) REFERENCES MaintenanceTechnician(MaintenanceID),

    FOREIGN KEY (RecordID) REFERENCES MaintenanceRecord(RecordID)

);

CREATE TABLE MaintenanceRecordPart (

    RecordID INT,

    PartID INT,

    PRIMARY KEY (RecordID, PartID),

    FOREIGN KEY (RecordID) REFERENCES MaintenanceRecord(RecordID),

    FOREIGN KEY (PartID) REFERENCES Part(PartID)

);

CREATE TABLE MaintenanceRecordDataScoop (

    RecordID INT,

    DataScoopID INT,

    PRIMARY KEY (RecordID, DataScoopID),

    FOREIGN KEY (RecordID) REFERENCES MaintenanceRecord(RecordID),

    FOREIGN KEY (DataScoopID) REFERENCES DataScoop(DataScoopID)

);

CREATE TABLE DataScoopZone (

    DataScoopID INT,

    ZonesID INT,

    PRIMARY KEY (ZonesID, DataScoopID),

    FOREIGN KEY (DataScoopID) REFERENCES DataScoop(DataScoopID),

    FOREIGN KEY (ZonesID) REFERENCES Zones(ZonesID)

);

CREATE TABLE SupplierPart (

    SupplierID INT,

    PartID INT,

    PRIMARY KEY (SupplierID, PartID),

    FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID),

    FOREIGN KEY (PartID) REFERENCES Part(PartID)

);

Data modeling can provide great help for database construction. Data modeling can be used to design the basic structure of the database, which is a necessary step in building a database. First, before building a database, using data modeling can clarify requirements, define the relationship between each table (such as one-to-one, one-to-many, many-to-many relationships) and the scope of the database. Secondly, it is about normalization. In this assessment work, we require the database to meet the normalization standard of the third paradigm, which is to eliminate redundant data and ensure the integrity and identity of the data. In addition, it can improve the performance of the database. Data modeling can help the database simplify and manage data, ensure the integrity and simplicity of data, optimize the database structure, and improve the performance of the database.

In summary, data modeling has played a huge role in promoting the work of building a database. Through data modeling, it is easier to build a database, ensure the rationality, simplicity and efficiency of database construction, and better apply it to the production environment.

“Data models are used to create databases, populate data warehouses, manage data for analytical processing, and implement applications that enable users to access information in meaningful ways. Data modeling is a process that you use to define the data structure of a database.” (Vivek, 2021)

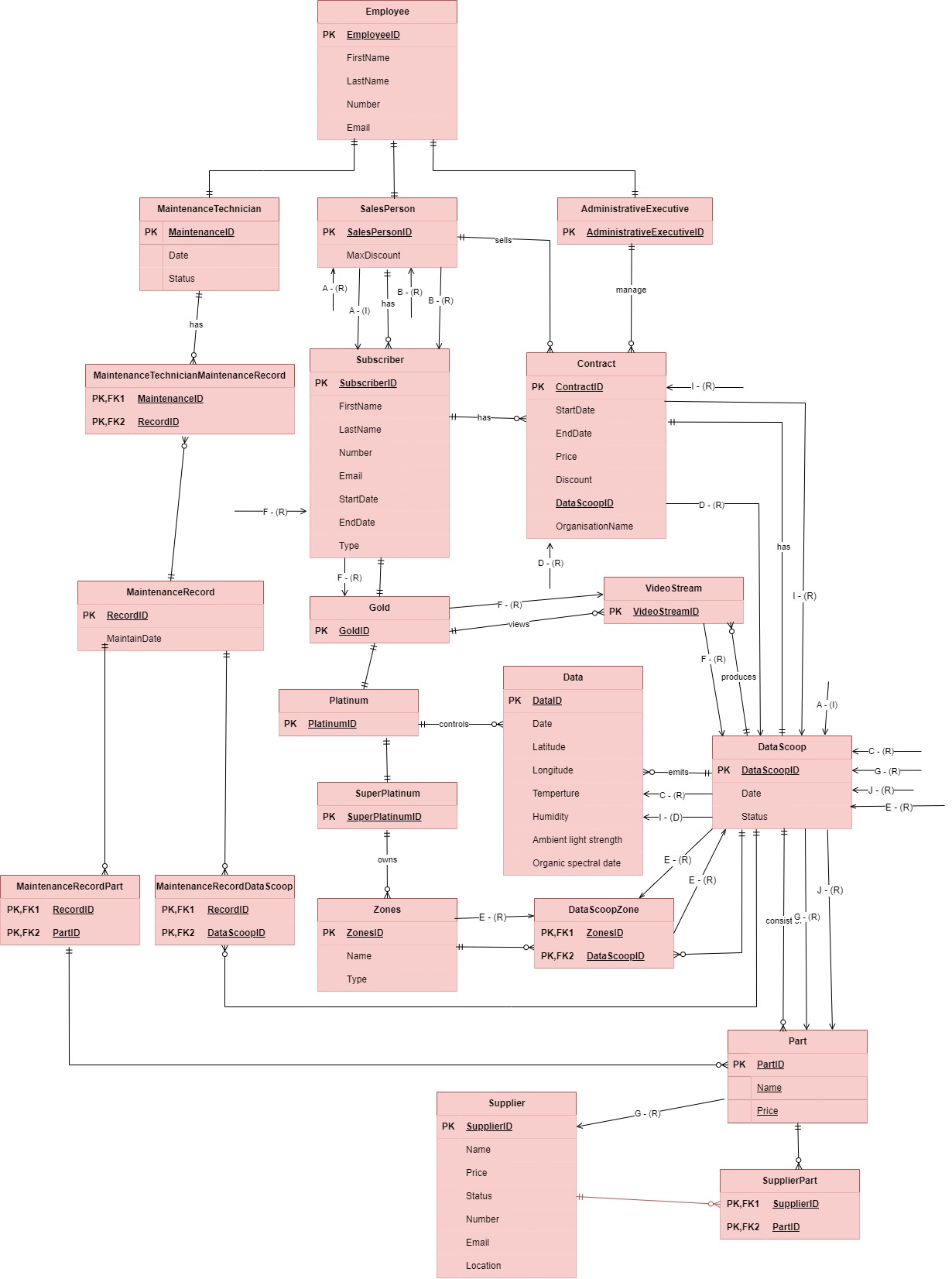
**Application of data modeling in test data:**

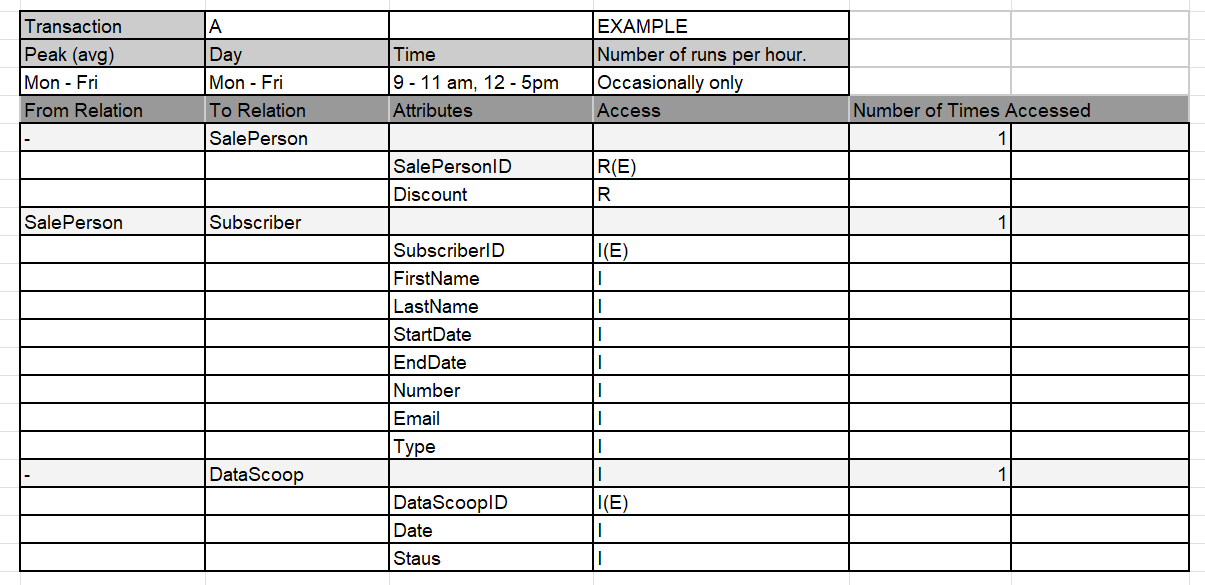
Data modeling plays an important role in test data. Creating test data is an essential stage in the database construction process. Test data can be used to test the function operation of the database and perform performance and security tests on the database to ensure that the database can run stably. Data modeling defines the various attributes in the table and the relationship between tables, and defines the data structure of the attributes, which can help us generate test data that meets the database requirements for testing based on the data structure.

### Step 4.2 Design enterprise constraints for target DBMS

## Step 5 Design physical representation

### Step 5.1 Analyze transactions

**Transaction A**



**Transaction B**

图形用户界面, 表格

描述已自动生成

**Transaction C**

表格

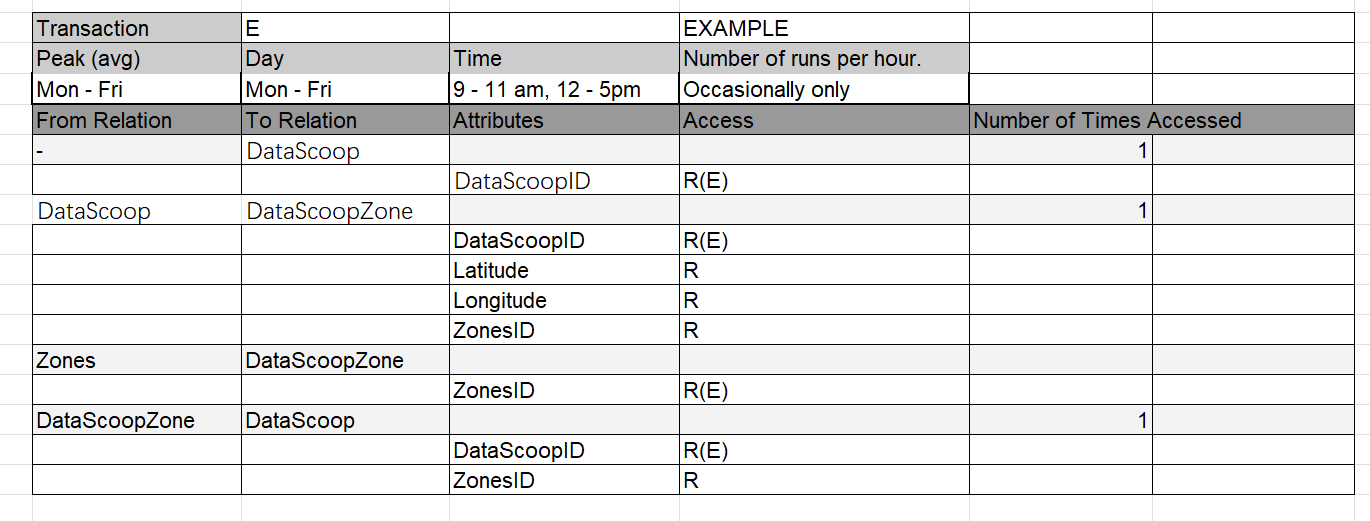
描述已自动生成

**Transaction D**

表格

描述已自动生成

**Transaction E**



**Transaction F**

表格

描述已自动生成

**Transaction G**

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

**Transaction H**

**Transaction I**

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

**Transaction J**

图形用户界面, 表格

描述已自动生成

Data modeling plays a very important role in transaction analysis. Data modeling can define the data structure of a table. By modeling the data of relationships and attributes, we can clearly see the relationship between each relationship and the attributes of the relationship, which helps us identify requirements and conduct demand analysis. Data modeling defines a clear, easy-to-understand graphical view to help us understand the interaction between relationships, output data, and input data to ensure that the operations performed by transactions are in line with the defined data structure and business logic. Through data modeling, we can define the primary key and foreign key of the relationship, add constraints to the relationship, and ensure the integrity and consistency of the data. Finally, through data modeling, the relationship between data can also be better identified and analyzed.

“Data modeling helps business analysts understand and document the data requirements of an organization, as well as gain insights into what kind of data is necessary to support various business processes.

Data modeling provides a clear and standardized way of illustrating data structures, allowing business analysts to effectively communicate data requirements to different stakeholders.

By using data modeling, business analysts can design integrated data structures that facilitate the data exchange between different systems and applications.” (Andrea, 2023)

### Step 5.2 Choose file organizations

For my FlightStream database, I plan to use B+ Tree File organization. Using B+ Tree file organization is suitable for searching and traversing the database, and it improves the speed of data traversal. For the database, you may need to consider the future growth of the database, and B+ Tree has no limit on the size of the database, which is very suitable for the currently designed database.

For example, I can create an index for abstract in the database, so that the index will use the B+Tree structure and can quickly find relevant data within the range when searching.

“**Advantages of B+ Tree File Organization**

[Tree traversal](https://www.geeksforgeeks.org/tree-traversals-inorder-preorder-and-postorder/) is easier and faster.

Searching becomes easy as all records are stored only in leaf nodes and are sorted in sequentially linked lists.

There is no restriction on B+ tree size. It may grow/shrink as the size of the data increases/decreases.“(*File Organization in DBMS | Set 3*, 2017)”

### Step 5.3 Choose secondary indexes

For the database I created, I think it is not worth adding secondary indexes. The reason is as follows: this database is just a very small database, and the total number of data contained in it may not even reach 500, which is an extremely small amount of data storage for a database. Adding secondary indexes will take up additional storage space, so adding secondary indexes is redundant for the current database.

“Queries with Complex Search Criteria: Secondary indexes can be used to support complex queries that search for data based on multiple conditions. By creating a secondary index based on the columns used in the search criteria, database management systems can access the data more efficiently.

Large Data Sets: Secondary indexing can be beneficial for large data sets where the time and resources required for data retrieval operations can be significant. By creating a secondary index, database management systems can access the data more quickly, reducing the time and resources required for data retrieval operations.” (*Secondary Indexing in Databases*, 2023)

### Step 5.4 Consider the introduction of controlled redundancy

I think the FlightStream database created in this assessment is a small database. The system can fully meet the performance requirements. There is little data in the database and the query is fast. Adding redundant data does not bring much benefit and will only increase costs.

### Step 5.5 Estimate disk space requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Relation Name** | **Attribute** | **Data Type** | **Length** | **Total Per**  **Row** |
|  | EmployeeID | Integer | 4 |  |
|  | FirstName | VarChar | 50 |  |
| Employee | LastName | VarChar | 50 |  |
|  | Number | VarChar | 20 |  |
|  | Email | VarChar | 100 |  |
|  |  |  |  | 224 |
|  |  | Integer | 4 |  |
| MaintenanceTechnician |  | Date | 4 |  |
|  |  | VarChar | 50 |  |
|  |  |  |  | 58 |
| SalePerson | SalesPersonID | Integer | 4 |  |
|  | MaxDiscount | Decimal | 5,2 |  |
|  |  |  |  | 9 |
| AdministrativeExecutive | AdministrativeExecutiveID | Integer | 4 |  |
|  |  |  |  | 4 |
| MaintenanceRecord | RecordID | Integer | 4 |  |
|  | MaintainDate | Date | 8 |  |
|  |  |  |  | 12 |
| Subscriber | SubscriberID | Integer | 4 |  |
|  | FirstName | VarChar | 50 |  |
|  | LastName | VarChar | 50 |  |
|  | Number | VarChar | 20 |  |
|  | Email | VarChar | 100 |  |
|  | StartDate | Date | 8 |  |
|  | EndDate | Date | 8 |  |
|  | Type | VarChar | 50 |  |
|  |  |  |  | 290 |
| DataScoop | DataScoopID | Integer | 4 |  |
|  | Date | Date | 8 |  |
|  | Status | VarChar | 50 |  |
|  |  |  |  | 62 |
| Contract | ContractID | Integer | 4 |  |
|  | StartDate | Date | 8 |  |
|  | EndDate | Date | 8 |  |
|  | Price | Decimal | 10,2 |  |
|  | Discount | Decimal | 5,2 |  |
|  | OrganisationName | VarChar | 100 |  |
|  | DataScoopID | Integer | 4 |  |
|  |  |  |  |  |
| Gold |  |  |  |  |
|  | GoldID | Integer | 4 |  |
|  |  |  |  | 4 |
| Platinum |  |  |  |  |
|  | PlatinumID | Integer | 4 |  |
|  |  |  |  | 4 |
| SuperPlatinum |  |  |  |  |
|  | SuperPlatinumID | Integer | 4 |  |
|  |  |  |  | 4 |
| Data |  |  |  |  |
|  | DataID | Integer | 4 |  |
|  | Date | Date | 8 |  |
|  | Latitude | Float | 8 |  |
|  | Longitude | Float | 8 |  |
|  | Temperature | Float | 8 |  |
|  | Humidity | Float | 8 |  |
|  | AmbientLightStrength | Float | 8 |  |
|  | OrganicSpectralDate | Date | 8 |  |
|  |  |  |  | 60 |
| Zones |  |  |  |  |
|  | ZonesID | Integer | 4 |  |
|  | Name | VarChar | 50 |  |
|  | Type | VarChar | 20 |  |
|  |  |  |  | 64 |
| Supplier |  |  |  |  |
|  | SupplierID | Integer | 4 |  |
|  | Name | VarChar | 100 |  |
|  | Price | Decimal | 10,2 |  |
|  | Status | VarChar | 50 |  |
|  | Number | VarChar | 20 |  |
|  | Email | VarChar | 100 |  |
|  | Location | VarChar | 100 |  |
|  |  |  |  | 384 |
| Part |  |  |  |  |
|  | PartID | Integer | 4 |  |
|  | Name | VarChar | 100 |  |
|  |  |  |  | 104 |
| VideoStream |  |  |  |  |
|  | VideoStreamID | Integer | 4 |  |
|  |  |  |  | 4 |
| MaintenanceTechnicianMaintenanceRecord |  |  |  |  |
|  | MaintenanceID | Integer | 4 |  |
|  | RecordID | Integer | 4 |  |
|  |  |  |  | 8 |
| MaintenanceRecordPart |  |  |  |  |
|  | RecordID | Integer | 4 |  |
|  | PartID | Integer | 4 |  |
|  |  |  |  | 8 |
| MaintenanceRecordDataScoop |  |  |  |  |
|  | RecordID | Integer | 4 |  |
|  | DataScoopID | Integer | 4 |  |
|  |  |  |  | 8 |
| DataScoopZone |  |  |  |  |
|  | DataScoopID | Integer | 4 |  |
|  | ZonesID | Integer | 4 |  |
|  |  |  |  | 8 |
| SupplierPart |  |  |  |  |
|  | SupplierID | Integer | 4 |  |
|  | PartID | Integer | 4 |  |
|  |  |  |  | 8 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Relation Name** | **Total per row** | **Max Size** | **Total Relation**  **Size** | **Pages Per**  **Relation** | **New Relation**  **Size** |
| Employee | 224 | 1120 | 5600 | 1.00 | 8192 |
| MaintenanceTechnician | 58 | 290 | 1459 | 1.00 | 8192 |
| SalePerson | 9 | 45 | 225 | 1.00 | 8192 |
| AdministrativeExecutive | 4 | 20 | 100 | 1.00 | 8192 |
| MaintenanceRecord | 12 | 60 | 300 | 1.00 | 8192 |
| Subscriber | 290 | 1450 | 7250 | 1.00 | 8192 |
| DataScoop | 62 | 310 | 1550 | 1.00 | 8192 |
| Contract | 4 | 20 | 100 | 1.00 | 8192 |
| Gold | 4 | 20 | 100 | 1.00 | 8192 |
| Platinum | 4 | 20 | 100 | 1.00 | 8192 |
| Data | 60 | 300 | 1500 | 1.00 | 8192 |
| Zones | 64 | 320 | 1600 | 1.00 | 8192 |
| Supplier | 384 | 1920 | 9600 | 2.00 | 16384 |
| Part | 104 | 520 | 2600 | 1.00 | 8192 |
| VideoStream | 4 | 20 | 100 | 1.00 | 8192 |
| MaintenanceTechnicianMaintenanceRecord | 8 | 40 | 200 | 1.00 | 8192 |
| MaintenanceRecordPart | 8 | 40 | 200 | 1.00 | 8192 |
| MaintenanceRecordDataScoop | 8 | 40 | 200 | 1.00 | 8192 |
| DataScoopZone | 8 | 40 | 200 | 1.00 | 8192 |
| SupplierPart | 8 | 40 | 200 | 1.00 | 8192 |

|  |  |
| --- | --- |
| **Total Database size (bytes):** | 180224 |
| **Total Database size (kilobytes)**: | 176 |
| **Total Database size (megabytes):** | 0.171875 |
| **Total Database size (gigabytes):** | 0.000168 |

## Step 6 Design security mechanisms

### Step 6.1 Design user views

**SalesPerson**

**MaintenanceTechnician**

**AdministrativeExecutive**

**Subscriber**

### Step 6.2 Design access rules

## Issues & Resolutions

I didn't encounter too many problems. When creating the database, I mistakenly placed the DataScoop table creation script after , which caused a foreign key constraint error. When I adjusted the order of DataScoop table creation in the script, this problem was also solved.

At the same time, an error occurred when creating and inserting test data. I used the semicolon at the end of each line of test data script code as a comma, which caused the test data insertion to fail. After modifying this problem, this problem was solved.

## Conclusions from the Physical Design

In the process of designing and implementing the physical database, I encountered many basic principles of effective data management. For different basic principles of data management, I also adopted different methods and working processes to correspond to different principles.

**1. Data integrity**

Ensuring data integrity is an important part of physical database design. When designing a database, the integrity and consistency of data can be ensured by defining primary key and foreign key constraints. When designing a physical database, I used foreign keys to confirm the relationship between tables when creating the Contract table and DataScoop table.

**2. Data security**

Data security is crucial at all stages of the database. By creating user views and user access rules, roles can be assigned to database personnel, permissions can be set, and the security of the database can be ensured. Certain specific and sensitive data can only be accessed and modified by authorized personnel.

**3. Data backup and recovery**

Data risk response plan is one of the issues that need to be considered in database design and development. By regularly backing up and restoring database data, losses can be stopped in time when data problems occur and losses can be minimized. At the same time, this is also one of the measures to ensure database security.

**4. Data scalability**

The database is not static, but will grow larger and larger over time, which requires considering the scalability of data. As business needs increase, the potential risks that come with it will also increase. Therefore, when designing a database, data scalability should be considered. This is a consideration for long-term development in the future, and a way to optimize database performance. In the future, as the amount of data increases and the burden on the database increases, it is very important to ensure that the database works long-term and stably.

**5. Data auditing and logging**

Usually, it is a good idea to create a run log for the database. The run log can record all database operations and data changes. Assuming that the database crashes and errors are encountered now, you can quickly find the location of the error and develop a solution for it by viewing the run log. So this is also an aspect that needs to be considered in database design.

## Reference

Andrea. (2023, December 5). *The Role of Data Modeling in Business Analysis | Cardanit*. Www.cardanit.com. <https://www.cardanit.com/blog/data-modeling-business-analysis/>

Chia, A. (2023, May 23). *Data Aggregation 101: Definition, Uses & Benefits*. Splunk-Blogs. <https://www.splunk.com/en_us/blog/learn/data-aggregation.html>

ehsanelahi. (2022, August 1). *8 Principles of Data Management*. Data Ladder. <https://dataladder.com/8-principles-of-data-management/>

Ekran. (2022, April 20). *10 Data Security Best Practices: Simple Steps to Protect Your Data*. Www.ekransystem.com. <https://www.ekransystem.com/en/blog/data-security-best-practices>

*File Organization in DBMS | Set 3*. (2017, December 26). GeeksforGeeks. <https://www.geeksforgeeks.org/file-organization-in-dbms-set-3/>

FIRICAN, G. (2021, August 3). *What is the difference between data classification and data categorization?* LightsOnData. <https://www.lightsondata.com/what-is-the-difference-between-data-classification-and-data-categorization/>

Upadhyay, M. (2019, July 31). *Boyce-Codd Normal Form (BCNF) - GeeksforGeeks*. GeeksforGeeks. <https://www.geeksforgeeks.org/boyce-codd-normal-form-bcnf/>

Rouse, M. (2014, November 24). *What is Fourth Normal Form (4NF)? - Definition from Techopedia*. Techopedia.com. <https://www.techopedia.com/definition/19453/fourth-normal-form-4nf>

*Secondary Indexing in Databases*. (2023, February 12). GeeksforGeeks. <https://www.geeksforgeeks.org/secondary-indexing-in-databases/>

Tharpe, B. (2020, January 24). *Types of Data Models: Conceptual, Logical & Physical*. Erwin Expert Blog. <https://blog.erwin.com/blog/types-of-data-models-conceptual-logical-physical/>

Vivek, J. (2021, November 30). *What is Data Modeling and Why it is Important?* Zuci Systems. <https://www.zucisystems.com/blog/what-is-data-modeling-and-why-is-it-important/>