5/9/2019

By Muhammad Ashkar Yousuf Milton

NMIT

Being There. Conceptual Model, Logical Model, & Physical Model

Milestone 1, 2, and 3

Table of Contents

[1. Executive Summary of the Conceptual Model of Being There 5](#_Toc5563984)

[2. A DESCRIPTION OF CONCEPTUAL MODELING 6](#_Toc5563985)

[2.1 Conceptual Model 6](#_Toc5563986)

[2.2 Chen Diagram 7](#_Toc5563987)

[3. Common Components of a Chen Entity-Relationship Diagram or ERD 8](#_Toc5563988)

[3.1 Entity 8](#_Toc5563989)

[3.2 Entity Type 8](#_Toc5563990)

[3.3 Entity Instance 9](#_Toc5563991)

[3.4 Weak Entity 10](#_Toc5563992)

[3.5 Attributes 10](#_Toc5563993)

[3.6 Keys 12](#_Toc5563994)

[3.7 Relationships 13](#_Toc5563995)

[3.8 Optionality 15](#_Toc5563996)

[3.9 Cardinality 16](#_Toc5563997)

[3.10 Extended Model 18](#_Toc5563998)

[4. The Entity Relationship Model of ‘Being There’ 20](#_Toc5563999)

[4.1 Diagram Rationale of Being There 20](#_Toc5564000)

[4.2 Diagram and Links to Diagram 21](#_Toc5564001)

[4.2.1 Diagram: 21](#_Toc5564002)

[4.2.2 Links 21](#_Toc5564003)

[5. DATA DICTIONARY for Being There 22](#_Toc5564004)

[5.1 Table 1: Entities 22](#_Toc5564005)

[5.2 Table 2: Relationships 26](#_Toc5564006)

[5.3 Table 3: Attribute 33](#_Toc5564007)

[6. Reasoning and Purpose of all Parts of the Model 45](#_Toc5564008)

[7. Rationale 47](#_Toc5564009)

[7.1 BTDrone 47](#_Toc5564010)

[7.2 BTDroneData 47](#_Toc5564011)

[7.3 RoamingZone 47](#_Toc5564012)

[7.4 BTDronePart 47](#_Toc5564013)

[7.5 Terrain 47](#_Toc5564014)

[7.6 Region 48](#_Toc5564015)

[7.7 Supplier 48](#_Toc5564016)

[7.8 Maintenance 48](#_Toc5564017)

[7.9 MaintenancePart 48](#_Toc5564018)

[7.10 Technician 48](#_Toc5564019)

[7.11 StaffMember 48](#_Toc5564020)

[7.12 SalesPerson 49](#_Toc5564021)

[7.13 Account 49](#_Toc5564022)

[7.14 Contractor 49](#_Toc5564023)

[7.15 Subscriber 49](#_Toc5564024)

[7.16 Subscription 49](#_Toc5564025)

[7.17 AdministrativeExecutive 49](#_Toc5564026)

[7.18 Standard 49](#_Toc5564027)

[7.19 Gold 50](#_Toc5564028)

[7.20 Platinum 50](#_Toc5564029)

[7.21 SuperPlatinum 50](#_Toc5564030)

[7.22 VideoStream 50](#_Toc5564031)

[7.23 BTWebsite 50](#_Toc5564032)

[8. Assumed Business Rules (with reasoning) 51](#_Toc5564033)

[9. Bibliography 52](#_Toc5564034)

[Contents 54](#_Toc5564035)

[1. Executive Summary 57](#_Toc5564036)

[2. Conceptual to Logical Model 58](#_Toc5564037)

[3. Crow’s foot Notation 64](#_Toc5564038)

[4. Images and Links to The Logical Model of Being There 65](#_Toc5564039)

[5. The ERD model of Being There included with Crows Foot Notation 66](#_Toc5564040)

[6. Rationale for ERD 67](#_Toc5564041)

[7. Normalization 69](#_Toc5564042)

[8. Normalization of Being There’s Data 79](#_Toc5564043)

[9. Data Dictionary 0](#_Toc5564044)

[10. Introduction to NaLER Analysis 25](#_Toc5564045)

[11. NaLER Analysis of Logical Model of Being There 27](#_Toc5564046)

[12. Conclusion 37](#_Toc5564047)

[13. Bibliography 39](#_Toc5564048)

# Executive Summary of the Conceptual Model of Being There

The aim of this report is to represent the way the BTDrones are deployed by the organization, Being There, and to find out how those those BTDrones explore different remote regions to collect data for scientific and commercial use. With the help of the conceptual model of being there, we would be exploring the physical aspect of ‘Being There” organization’s BTDrones and how the conceptual model can be used to represent the relationships of entities within the database of ‘Being There”. Furthermore, with the conceptual model of this project we will be able to represent both the behavior and data at the same time. To represent the Conceptual model of the project, we will be undergoing several steps which include the description of the ‘Being There’ conceptual model, the entity-relationship description of Being There, and the explanation of reasoning and purpose of all parts of the model used in this report.

# A DESCRIPTION OF CONCEPTUAL MODELING

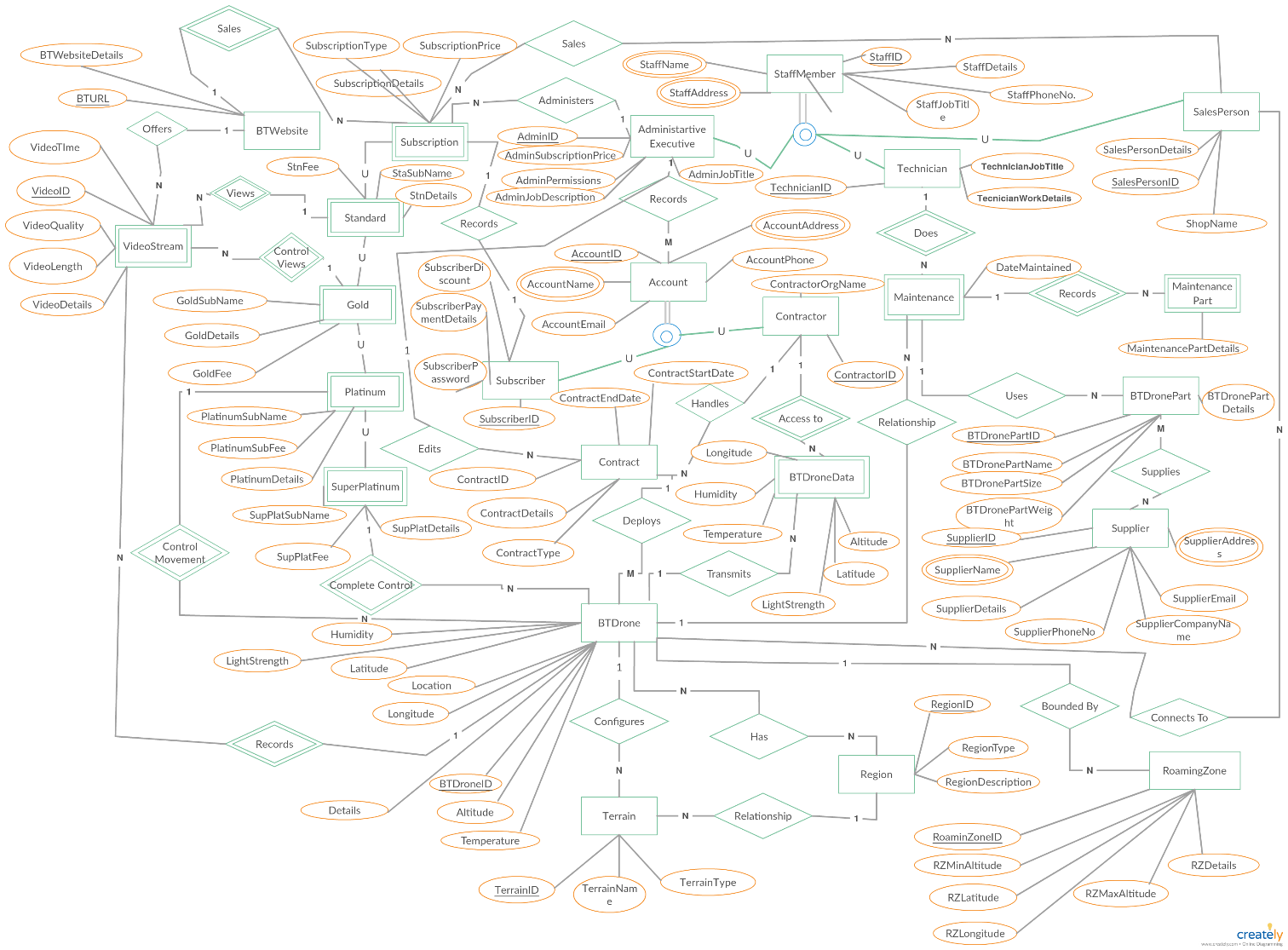
## Conceptual Model

A conceptual model is also known as the data model which can be used to describe the conceptual schema when a database system is implemented. It hides the internal details of physical storage and targets on describing entities, datatype, relationships, and constraints. It is also known as conceptual database design. It is also a process of constructing the model of information that can be implemented in an enterprise, thus can act independently upon implementation on the target Database Management System, program-based application, programming languages, or any physical subject. This model can be referred as conceptual data model which is independent of details that has been implemented.

Conceptual database design is used to construct the conceptual representation of the database which include entities, relationships, and attributes that are important to be identified. If we begin with the phase of conceptual database design, we need to be aware of the steps associated with the conceptual database design. Building the conceptual model is the primary stage of the conceptual database model design. It is assembled with several steps. We will be using these steps to design the conceptual data model as required by the project. As such we have designed a conceptual diagram of the ‘Being There’ project, that exhibits, the following steps while working on that.

* + Identifying the types of Entities that are involved in the conceptual diagram
  + Identifying different Relationships types
  + Identifying and associating the attributes with the entity or the relationship types.
  + Determining domains for the attributes that can be obtained from the conceptual data model.
  + Identifying the foreign key(s) that is associated with the entity type. If there is one more foreign key, we can choose one to be the primary key and the rest to be the alternative key.
  + Considering the use of specialization, aggregation, and compensation for the conceptual model.
  + Checking if there is any redundancy in the conceptual model.

## Chen Diagram



# Common Components of a Chen Entity-Relationship Diagram or ERD

A basic Chen Diagram is the first original entity-relationship diagram that is designed by Peter Chan. This diagram represents the entities and their relationships associated with each other. The Chen diagram is also known as ER (Entity Relationship) Diagram. It is considered as the most important model to develop the conceptual diagram of the database. Entity Relationship diagram helps us to understand the concepts behind entity and relationship, thus it can be used to understand the methodology to plot the concept related with entity relationship diagram.

## Entity

The best way to describe entity is that it is a concept or item that can exist on its own, having the same properties. They can be identified through a project as independent and unique properties. This entity could be an object that may exist physically which includes person, house, etc. or it can be an object that may exist conceptually which includes company, job, course, etc. An entity can be different depending on the project and the designer. An entity type is shown with the help of rectangle labeled.

**Entity**

## Entity Type

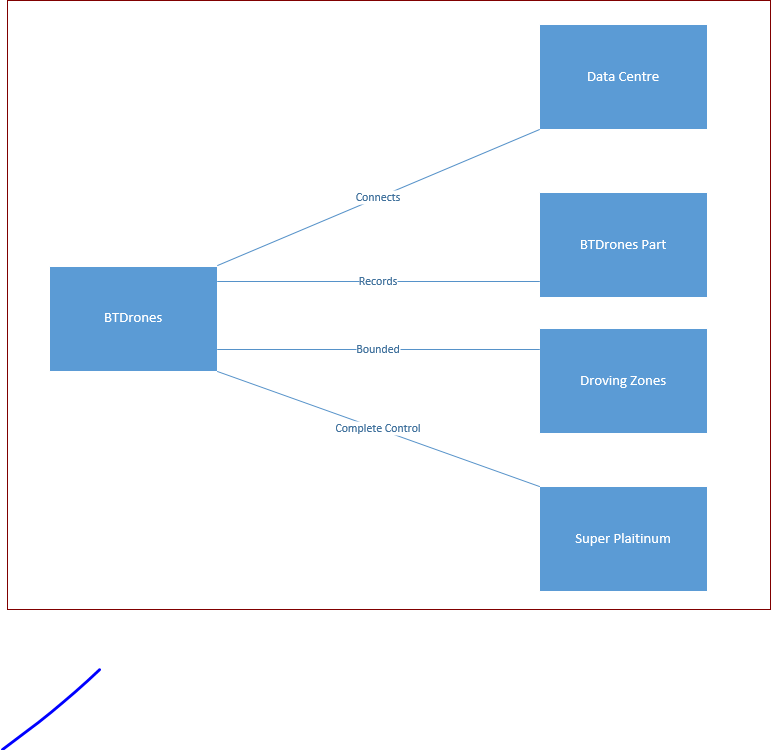
Entity type can be defined as collection of entities that have same attribute. In a database entity type in a database is defined by its name and attribute. To describe further, an entity type is the fundamental building block of a conceptual diagram, that describes the structure of entity-relationship model diagram. An entity type is represented with the help of rectangle that encloses the name which is singular noun. An entity type can be referred as a pattern for the entity type instances. Each entity type instance may include:

* A Unique identifier.
* An entity key which can be defined by one or more properties.
* It can be data that may appear as properties in the entity type instances.

Entity Type

Student

If we are talking about conceptual model that has 4 entity types, that we can use this example to define entity types:



## Entity Instance

It is a single occurrence of an entity type which is uniquely identifiable object of the entity type. In ER diagram instance means:

* The primary key is identified as single instance.

## Weak Entity

This is an entity that can’t exist on its own without having a relationship to its adjacent entity which it depends on. It also doesn’t exist without having a relationship with another existent entity. As a result, it is identified by this relationship. To explain further a weak entity is always included with a total participation constraint with respect to its identifying relationship because it cannot be identified without its owner identity. A weak entity may be included with a partial key provided with a list of attributes which identifies the weak entity is related to the owner’s entity. A weak entity is identified with a rectangle label with dotted line on top of it (Onsman, 2018).

Weak Entity

## Attributes

They are the aspects that describe the entity related with it. Attributes names are enclosed by ovals and connected to their entity by using single line. To explain further we can say that they are properties of an individual entity. It is represented with the help of ellipses and is connected directly with the entity (enclosed in a closed rectangle). We can distinguish attributes into 6 types. These include:

* + - 1. *Simple Key Attribute*: It is an attribute that is unique of a entity. It is a unique identifier that identifies each entity. As such, CustomerID could be a unique identifier to Customer entity, as each customer has unique identifier of its own. Another example of simple key attribute includes StudentID. To illustrate simple key attribute:

**Student**

* + - 1. *Composite Attribute*: This is attribute that can be divided into smaller subparts which thus represents the more basic attributes providing independent meaning (Cochrane, 2019). To explain further it is an attribute that is associated with many other attributes. An example includes: Address. This address can be divided into smaller subparts having the Street, City, ZIP, Country, State, and Town. To illustrate, a diagram of the Composite Key Attribute is provided.

**Address**

* + - 1. *Single Valued Attribute*: This attribute that has only a single value for entity. An example includes: StudentAge. An illustration of single valued

**Student**

* + - 1. *Derived Attribute*: It is an attribute which can be derived from other attributes. The derived attribute doesn’t necessarily have to be physically stored within the database; instead, it can be derived by using an algorithm (Cochrane, 2019). In a ER diagram a Derived Attribute can be represented by dashed oval. Example of derived attribute incudes Age. An illustration of derived attribute includes:

**Student**

* + - 1. *Multi Valued Attribute*: This attribute consists of more than one value for an entity (Cochrane, 2019). In an ER diagram it is denoted by double circled oval. An example of multi valued attribute includes, PhoneNumber, CollegeDegree, etc. An illustration of multi valued attribute includes:

**Student**

* + - 1. Stored Attribute: They are attributes that are already stored in the database and from which the value of another attribute can be derived. As such, we can calculate the age of a Student from the Student’s DOB and the current date.

**Student**

## Keys

Keys can be into several parts. These include:

1. *Key*: It is the data item that will allow us to uniquely identify individual occurrences of an entity type. Example includes StudentID from the Student Entity.
2. *Super Key*: It is an attribute or set of attributes that uniquely identifies a tuple. An example includes:

Student

Primary Key

**StudentID**

StudentName

StudentDOB

StudentAddress

StudentDepartment

PassportNo

IRDNumber

Super Keys

Candidate Key

1. *Candidate Key*: It is the minimal super key with the property of irreducibility and uniqueness.
2. *Primary Key*: An entity type may have one or more possible candidate keys, the one which is selected as the primary key.
3. *Composite Key*: Candidate key is a key that consists of two or more attributes.
4. *Foreign Key*: An attribute or set of attributes that matches the candidate key or other or some relation. An example includes: DepartmentID is a foreign key in Student.

Department

Student

**DepartmentID**

DepartmentName

**StudentID**

StudentName

StudentDOB

StudentAddress

DepartmentID

PassportNo

IRDNumber

## Relationships

A relationship represents how two or more entities are inter-related with each other. To put into an example:

Enrolled In

Course

Student

In the above example, a Student enrolled in a Course, and both of these are entities. Here, ‘Enrolled In’ defines the relationship between Student and Course.

A diamond shape is used to show the relationship between the entities. We can represent a strong entity with the help of a single line, whereas we use double line to represent a weak entity.

Weak Entity

Normal Entity

Relationship can be classified into a degree of relationship. It is the number of entity types which participate in a relationship. To explain further:

1. Unary Relationship: It is a relationship where only entity participates in a relation. Here, One person is married to one person only, is an example of Unary Relationship

Married To

Person

1. Binary Relationship: It is a relationship where two entities participate in a relation. Here, Student is Enrolled in a Course, is an example of a binary relationship.

Enrolled In

Course

Student

1. Ternary Relationship: It is a relationship where three entities are involved. Here, Physician, Patient, and Treatment must have connection with the relationship, log. This shows the example of Ternary Relationship.

Physician

Log

Treatment

Patient

1. N-Array Relationship: It is a relationship where n number of entities are involved. It is also known as Quaternary relationship. Here, the entities, Group, Class, Classroom, and Schedule have relationship with ‘Takes’, which is an example of quaternary relationship.

Group

takes

Class

Classroom

Schedule

## Optionality

It is also known as the Participation Constraints. This is a representation of an entity involved in the relation. It determines whether all the entity or some of the entity occurrences participate in a relationship. It means, if all the values of the entity are participating in a relation, then it is called total participation. If only few values of an entity are a part of relation, then it is called as partial participation (Concepts of ER Data Model, n.d.). An example for total participation includes that every student must attend the class then participation of the student in attend is Total.

Class

Attend

Student

1

N

In this aspect, total participation is also called as Existence dependencies. Every entity in total set of students must be related to a class through Attend. However, we can’t say that every student must STUDY in the Class. Hence, we can say that the relationship is Partial.

The Total Participation is indicated with the help of double line and the Partial Participation is indicated by a single line (Connolly, 2005). It means, the number of times an entity participates in a relationship of another entity. If we are provided with two entities like A and B, then the cardinality constraint specifies the number of instances of Entity A must or may be associated with the instances of Entity B (Cochrane, 2019).

## Cardinality

Cardinality describes the maximum number of possible relationship occurrences for an entity participating in a given relationship type.

Cardinality can be of different types. These include:

1. **One to One Relationship (1:1)** – In this relationship each entity in each entity set can take part only once in the relationship, thus saying the cardinality is one to one. To put an example, a Student attends one Class (Database Management System | ER Model, 2018).

1

1

Class

Attend

Student

1. **One to Many Relationship (1:M)** – In this relationship each entity in many entities of a set can take part many times in the relationship, thus concluding the cardinality is one to many. To put an example, a Student Enrolled in Many Course (Database Management System | ER Model, 2018).

Enrolled

1

M

Course

Student

1. **Many to One Relationship (M:1)** – In this relationship when entities in one entity set can take part only once in the relationship set and entities in other entity set can take part more than once in the relationship set, thus concluding the cardinality is Many to One. To put an example, Many Students works in One Project (Database Management System | ER Model, 2018).

Works

Student

M

1

Project

1. **Many to Many Relationship (M:M)** – In this relationship, when in all entity sets can take part more than once in the relationship, thus concluding that the cardinality is Many to Many. To put an example, Many Students Enrolled in Many Course (Database Management System | ER Model, 2018).

M

Enrolled In

Course

M

Student

## Extended Model

Sometimes when we are working on a project, it is hard to determine if a particular concept can be considered as entity type or attribute type or relationship type. To clear that problems, there are certain essential concepts that we should remember. These include:

1. First modeling the concept as an attribute and then refine that into a relationship because the concept can be determined as an attribute is reference to another attribute. A pair of such relationship that are inverses of one another are refined to as binary relationship (Cochrane, 2019).

1

N

Student

Enrolled In

Course

They can be refined to binary relation

Course

Student

1. An attribute that exists in several entities can be upgraded to an independent entity. To explain further, for NMIT there are many entity types which include Student, Course Instructor, and Course. Each of this entity has an entity named DepartmentName. We are going to represent with the help of ER diagrams.

BelongsTo

WorksFor

fed In

Register

Course

Student

Course Instructor

Department

Can be refined as

Course Instructor

Student

Course

# The Entity Relationship Model of ‘Being There’

## Diagram Rationale of Being There

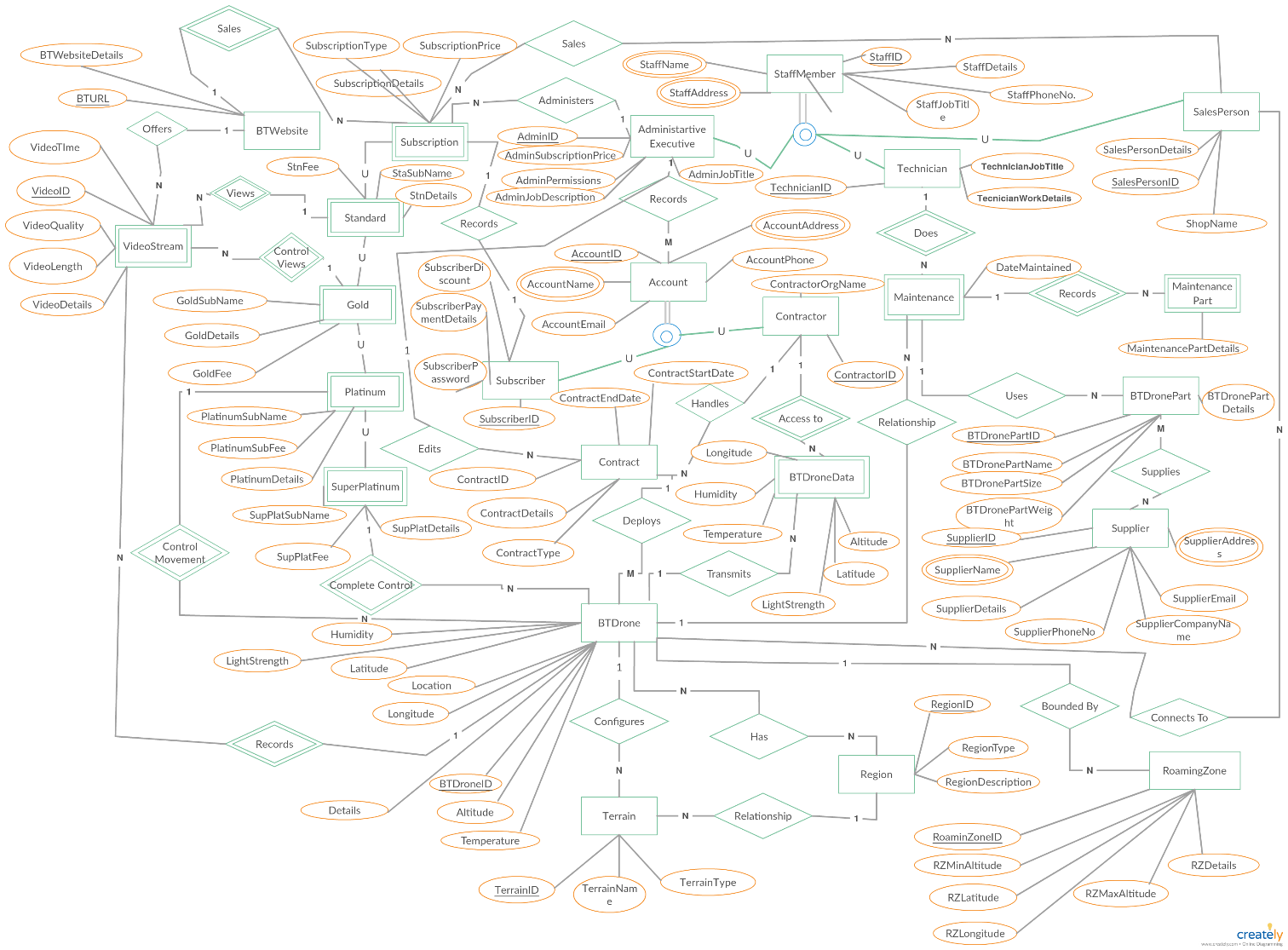
The aim of this diagram to show how each BTDrone can be used to collect data by travelling different regions. The diagram shows a graphical representation of relationship between entities and attributes related with the drones.

Being There is a non-profit organization that provides remote data sensing and to provide service through drones, Being There is offering two types of services which include the Contract and Subscription. The subscribers can be of different types which allow excess to different usability to the users depending on the subscription they choose, and each contract are provided with contract details. The information that is gathered by each drone, can be accessed by the contractor having a contract.

Being There also hires different types of people assigning them different roles. They hire administrative executive whose job is to change the price of the subscriptions and to edit the contract details. Again, SalesPerson are there to sale the subscriptions to different users. Furthermore, Technician are there to fix the drone if it needs fixing or needs maintenance.

## Diagram and Links to Diagram

## 4.2.1 Diagram:



## 4.2.2 Links

https://drive.google.com/file/d/1VEhgQNdlThUBu0X9rU5NTJZMPSP3nqh-/view?usp=sharing

# DATA DICTIONARY for Being There

## 5.1 Table 1: Entities

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity Name** | **Description** | **Aliases** | **Occurrence** |
| ***BTDrone*** | Data is stored on each BTDrone. BTDrone is deployed by Being There organization |  | Each BTDrone must produce one BTDroneData  One BTDrone must be bounded by one or many RoamingZone(s)  One BTDrone may configure one or many Terrain(s)  Each BTDrone is connected to one or more Supplier(s)  Each BTDrone has one Region  Each BTDrone may records one or more VideoStream(s)  One BTDrone must be maintained by one or more Maintenance  Each BTDrone must be deployed by one Contract.  Each BTDrone must have control movement by one Platinum.  Each BTDrone must have complete control by one SuperPlatinum.  One BTDrone must edit one or more contract details. |
| ***BTDroneData*** | Data is stored in each BTDroneData. Data is transmitted to BTDroneData from each drone. It stores information regarding humidity, temperature, and light sensitivity send by each BTDrone | Data Storage Centre | Each BTDroneData must be transmitted by one BTDrone  Each BTDroneData must have access to one or more Contractor(s). |
| ***RoamingZone*** | Data is stored in each RoamingZone. RoamingZone is the destination zones each BTDrone covered. | Droving Zone | One RoamingZone must be bounded by one or many BTDrone(s). |
| ***BTDronePart*** | Data is stored on each BTDronePart. BTDrone is the part that is required for each BTDrone and is identified by its unique code. | Tool | Each BTDronePart must be supplied by one or more Supplier(s).  Each BTDronePart must be used by one Maintenance. |
| ***Terrain*** | Data is stored on each Terrain. Terrain is where the drone drift around and is configured by the BTDrone every time it hovers around particular terrain. |  | Each Terrain must be configured by one BTDrone  Each Terrain must have one Region |
| ***Region*** | Data is stored on each Region. Region is where each BTDrone has allocated area, specified by a certain boundary. |  | One Region may have one or more BTDrone(s).  One Region may have one or more Terrain(s). |
| ***Supplier*** | Data is stored by each Supplier about BTDrone parts. Supplier is required to supply the BTDrone part wen a part is required for a BTDrone |  | Each Supplier must supply one or more BTDronePart. |
| ***Maintenance*** | Data is stored on each Maintenance. Maintenance is done by a Technician when any BTDrone breaks or needs to be repaired | Repair | Each Maintenance must be done by one Technician  Each Maintenance must record one or more MaintenancePart |
| ***MaintenancePart*** | Data is stored in each MaintenancePart. When a Maintenance is done MaintenancePart keeps a record of the drone that has undergone maintenance. |  | Each MaintenancePart must be recorded by one or more MaintenancePart |
| ***Technician*** | Data is stored on each Technician. Technician is hired by the Staff of Being There, to fix any issue associated with BTDrone. |  | Each Technician must be a StaffMember.  One Technician may do one or more Maintenance. |
| ***StaffMember*** | Data is stored on the Staff. StaffMember represents the general member of the BTDrone. It is specialized as SalesPerson, AdministrativeExecutive, and | SalesPerson, AdministrativeExecutive, Technician. CEO | One StaffMember may specialize as a Technician.  One StaffMember may specialize as an AdministrativeExecutive.  One StaffMember may specialize as a SalesPerson. |
| ***SalesPerson*** | Data is stored on each SalesPerson. SalesPerson are hired to sale Subscription for the BTDrone. They each have a SalesPersonID |  | Each SalesPerson must be a StaffMember.  Each SalesPerson may sale one or more Subscription. |
| ***Account*** | Data is stored on each Account. Account is created to keep record of the BTDrone users which include the Subscriber and the Contractor |  | Each Account must be recorded by one AdministrativeExecutive.  One Account may keep record of one or more Subscriber  One Account may keep record of one or more Contractee |
| ***Contractor*** | Data is stored in each Contractor. Contractor provides Contract for the BTDrone. |  | Each Contractor must be recorded by one Account  Each Contractor must handle one or more Contract.  Each Contractor may have access to one or more BTDroneData. |
| ***Contract*** | Data is stored in each Contract. Contract provides access to the Super Platinum Subscriber and they are informed with the type of Contract the subscriber is having. They also the one who allows the BTDrone to be deployed |  | One Contract may deploy one or more BTDrone.  One Contract must be handled by one Contractor.  Each Contract must be edited by one AdministrativeExecutive |
| ***Subscriber*** | Data is stored in each Subscriber. Subscriber handles the type of subscription, users can have access to. They each have a SubscriberID |  | Each Subscriber must be recorded by one Account.  One Subscriber must record one Subscription. |
| ***Subscription*** | Data is stored in each Subscription. Subscription records the type of subscriptions provided by the BTDrone. |  | Each Subscription may be sold by one or more SalesPerson.  Each Subscription must be administered by one Administrative Executive.  One Subscription records one Subscriber |
| ***AdministrativeExecutive*** | Data is stored in each AdministrativeExecutive. AdministrativeExecutive is hired by the Staff of Being There to administer over BTDrone |  | Each AdministrativeExecutive must be a StaffMember  One AdministrativeExecutive must administer one or more Subscription(s).  One AdministrativeExecutive must record one or more Account(s). |
| ***Standard*** | Data is stored in each Standard. Standard is a type of subscription, created by the Subscriber. | User, Subscriber | One Standard may view one or more VideoStream(s) |
| ***Gold*** | Data is stored in each Gold. Gold is a type of subscription created by the Subscriber. | User, Subscriber | One Gold may control view one or more VideoStream(s) |
| ***Platinum*** | Data is stored in each Platinum. Platinum is a type of subscription created by the Subscriber. | User, Subscriber | One Platinum may control movement one or more BTDrone(s). |
| ***SuperPlatinum*** | Data is stored in each SuperPlatinum. SuperPlatinum is a type of subscription created by the Subscriber. | User, Subscriber | One SuperPlatinum may have complete control on one or more BTDrone(s). |
| ***VideoStream*** | Data is stored in each VideoStream. VideoStream is the video shot by each BTDrone, then making it accessible for the Subscribers to watch, associated with different functions to view the video. Each video has its own VideoStreamID | Video | Each VideoStream may be viewed by one Standard.  Each VideoStream may have control view by one Gold.  Each VideoStream must be offered in one BTWebsite |
| ***BTWebsite*** | Data is stored in each BTWebsite. BTWebsite has access to all the video streams, shot by each BTDrone. The BTWebsite has its own unique URL | Website | One BTWebsite may sale one or more Subscription(s)  One BTWebsite must offer one or more VideoStream(s). |

## 5.2 Table 2: Relationships

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Participation** | **Cardinality** | **Relationship** | **Cardinality** | **Participation** | **Entity Name** |
| ***BTDrone*** | Mandatory | 1 | Configures | M | Optional | Terrain |
| Optional | 1 | Has | 1 | Mandatory | Region |
| Mandatory | 1 | Bounded By | M | Optional | RoamingZone |
| Optional | 1 | Transmits | M | Mandatory | BTDroneData |
| Mandatory | M | Connects to | M | Optional | Salesperson |
| Optional | 1 | Maintains | M | Mandatory | Maintenance |
| Optional | M | Deploys | 1 | Mandatory | Contract |
| Mandatory | M | Complete Control | 1 | Mandatory | SuperPlatinum |
| Mandatory | M | Control Movement | 1 | Mandatory | Platinum |
| Optional | M | Records | M | Optional | VideoStream |
| Mandatory | 1 | Edits | M | Mandatory | Contract |
| ***RoamingZone*** | Optional | M | Has | 1 | Mandatory | BTDrone |
| ***Region*** | Mandatory | 1 | Has | M | Optional | BTDrone |
| Mandatory | 1 | Has | M | Optional | Region |
| ***Terrain*** | Optional | M | Configures | 1 | Mandatory | BTDrone |
| Optional | M | Has | 1 | Mandatory | Region |
| ***BTDroneData*** | Mandatory | M | Transmits | 1 | Optional | BTDrone |
| Optional | M | Access to | 1 | Mandatory | Contractor |
| ***BTDronePart*** | Mandatory | M | Supplied by | M | Optional | Supplier |
| Optional | M | Used By | M | Mandatory | Maintenance |
| ***Supplier*** | Optional | M | Supplies | M | Optional | BTDronePart |
| ***Maintenance*** | Mandatory | 1 | Uses | M | Optional | BTDronePart |
|  | Optional | 1 | Records | M | Optional | MaintenancePart |
| Mandatory | M | Done By | 1 | Optional | Technician |
| Mandatory | M | Maintains | 1 | Optional | BTDrone |
| ***MaintenancePart*** | Optional | M | Recorded by | 1 | Optional | Maintenance |
| ***Technician*** | Optional | 1 | Does | M | Mandatory | Maintenance |
| Mandatory | 1 | Is | 1 | Mandatory | Staff |
| ***Staff*** | Mandatory | 1 | Is | 1 | Mandatory | Technician |
| Mandatory | 1 | Is | 1 | Mandatory | SalesPerson |
| Mandatory | 1 | Is | 1 | Mandatory | AdministrativeExecutive |
| ***SalesPerson*** | Optional | M | Sales | M | Mandatory | Subscription |
| Optional | M | Connects to | M | Mandatory | BTDrone |
| ***AdministrativeExecutive*** | Mandatory | 1 | Administers | M | Mandatory | Subscription |
| Mandatory | 1 | Records | M | Mandatory | Account |
| ***Subscription*** | Mandatory | M | Administered By | 1 | Mandatory | AdministrativeExecutive |
| Mandatory | 1 | Recorded by | 1 | Mandatory | Subscriber |
| Optional | 1 | Includes | M | Mandatory | Standard |
| Optional | 1 | Includes | M | Mandatory | Gold |
| Optional | 1 | Includes | M | Mandatory | Platinum |
| Optional | 1 | Includes | M | Mandatory | SuperPlatinum |
| Optional | M | Sold By | 1 | Mandatory | BTWebsite |
| ***Subscriber*** | Mandatory | 1 | Records | 1 | Mandatory | Subscription |
| Mandatory | 1 | Is | 1 | Mandatory | Account |
| ***Account*** | Mandatory | 1 | Is | 1 | Mandatory | Subscriber |
| Mandatory | 1 | Is | 1 | Mandatory | Contractor |
| ***BTWebsite*** | Mandatory | 1 | Offers | M | Optional | VideoStream |
| Mandatory | 1 | Sales | M | Optional | Subscription |
| ***VideoStream*** | Optional | M | Offered By | 1 | Mandatory | BTWebsite |
| Optional | M | Viewed By | 1 | Mandatory | Standard |
| Optional | M | Controlled View | 1 | Mandatory | Gold |
| Mandatory | M | Recorded By | M | Optional | BTDrone |
| ***Standard*** | Mandatory | 1 | Views | M | Optional | VideoStream |
| ***Gold*** | Mandatory | 1 | Control Views | M | Optional | VideoStream |
| ***Platinum*** | Mandatory | 1 | Control Movement | M | Optional | BTDrone |
| ***SuperPlatinum*** | Mandatory | 1 | Complete Control | M | Optional | BTDrone |
| ***Contract*** | Mandatory | 1 | Deploys | M | Optional | BTDrone |
| Mandatory | M | Edited by | 1 | Mandatory | Contract |
| Mandatory | 1 | Handled by | 1 | Mandatory | Contractor |
| ***Contractor*** | Mandatory | 1 | Is | 1 | Mandatory | Account |
|  | Mandatory | 1 | Access to | M | Optional | BTDroneData |
|  | Mandatory | 1 | Handles | 1 | Mandatory | Contract |

## 5.3 Table 3: Attribute

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Null** | **Key** | **Default Value** |
| ***BTDrone*** | *BTDroneID* | It is a unique number to identify each BTDrone. | AutoNumber | - | No | No | No | CK | ---- |
| Location | Location of each drone which is identified by its altitude and latitude | Character (30) | - | No | NO | No | No | -- |
| Humidity | Atmospheric Moisture the BTDrone is going to be in. | Character (30) | - | No | No | Yes | No | - |
| LightStrength | Intensity of the light the BTDrone is going to face | Character (30) | - | No | No | Yes | No | - |
| Altitude | Altitude is the vertical direction of the BTDrone from ground. | Decimal | - | No | No | No | No | - |
| Latitude | It is an angle which range from 0 degree at the Equator to 90 degree at the poles, for the BTDrone | Character (20) | Angle | No | No | No | No | - |
| Longitude | It is used to specify the precise location on the surface of the BTDrone | Character (20) |  | No | No | No | No | - |
| Temperature | The degree of intensity of heat the BTDrone is going to face in Celsius | Decimal | Celsius | No | No | Yes | No | - |
| Details |  | Character (150) | - | No | No | Yes | No | - |
|  |  | The additional details telling us more about the drone. |  |  |  |  |  |  |  |
| ***RoamingZone*** | *RoamingZoneID* | Unique number to identify the Roaming Zone | AutoNumber | DrovingZone | NO | NO | NO | CK | ---- |
| RZMaxAltitude | Maximum Altitude the roaming zone is for the BTDrone | Decimal | Altitude | No | No | No | NO | -- |
| RZMinAltitude | Minimum Altitude the roaming zone is for the BTDrone | Decimal | Altitude | No | No | No | No | -- |
| RZLatitude | Latitude of the BTDrone bounded by specific roaming zone | Character (30) | Latitude | No | No | No | NO | -- |
| RZLongitude | Longitude of the BTDrone bounded by specific roaming zone | Character (30) | Longitude | No | No | No | NO | -- |
| RZDetails | Information about the RoamingZone | Character (150) | - | NO | No | Yes | No | -- |
|  |  |  |  |  |  |  |  |  |  |
| ***Region*** | RegionID | Unique Number to identify each region | AutoNumber | - | No | No | No | CK | -- |
| RegionType | The type of earth surface that has unifying characteristics like climate and weather. | Character (50) | - | No | No | No | No | -- |
| RegionDescription | The details associated with the region | Character (150) | - | No | No | No | NO | -- |
|  |  |  |  |  |  |  |  |  |  |
| ***Terrain*** | *TerrainID* | Unique number to identify the terrain | AutoNumber | No | - | No | No | CK | -- |
| TerrainName | The name of the terrain given to each terrain | Character (30) | No | - | Yes | No | No | -- |
| TerrainType | The types of terrain | Character (30) | No | No | NO | NO | No | - |
|  |  |  |  |  |  |  |  |  |  |
| ***BTDroneData*** | Humidity | Atmospheric Moisture the BTDrone is going to be in. | Character (30) | - | No | No | Yes | No | - |
| Temperature | The degree of intensity of heat the BTDrone is going to face in Celsius | Decimal | Celsius | No | No | Yes | No | - |
| Altitude | Altitude is the vertical direction of the BTDrone from ground. | Decimal | - | No | No | No | No | - |
| Latitude | It is an angle which range from 0 degree at the Equator to 90 degree at the poles, for the BTDrone | Character (20) | Angle | No | No | No | No | - |
| Longitude | It is used to specify the precise location on the surface of the BTDrone | Character (20) | - | No | No | No | No | - |
| LightStrength | Intensity of the light the BTDrone is going to face | Character (30) | - | No | No | Yes | No | - |
|  |  |  |  |  |  |  |  |  |  |
| ***BTDronePart*** | *BTDronePartID* | Unique number of each part | AutoNumber | Part No. | No | No | No | CK | - |
| BTDronePartName | Name of each drone part | Character (150) | - | No | No | Yes | No | -- |
| BTDronePartSize | Size of the part which includes height, length, and width | Decimal | - | NO | No | Yes | NO | - |
| BTDroneWeight | Weight of the part which can be measured in grams | Decimal | - | No | No | Yes | No | -- |
| BTDronePartDetail | Details associated with each part of the drone | Character (150) | - | No | No | Yes | No | -- |
|  |  |  |  |  |  |  |  |  |  |
| ***Contract*** | *ContractID* | Unique number to identify the contract | AutoNumber | - | No | No | No | NO | - |
| ContractType | The type of the contract | Character (30) | - | No | NO | No | NO | - |
| ContractDetails | Details and information associated with the contract | Character (150) | - | NO | NO | Yes | NO | - |
| ContractStartDate | The day the contract is going to start | Date/Time | - | NO | NO | NO | NO | - |
| ContractEndDate | The day the contract is going to end. | Date/Time | - | NO | NO | No | NO | - |
|  |  |  |  |  |  |  |  |  |  |
| ***Contractor*** | *ContractorID* | Unique number to identify the contractor | AutoNumber | - | No | No | No | No | - |
| ContractorOrgName | The name of the contractor’s organization Name | Character (150) | - | Yes | No | Yes | No | - |
|  |  |  |  |  |  |  |  |  |  |
| ***Maintenance*** | DateMaintained | The date when the maintenance of the drone part is done | Date/Time | - | No | No | No | No | - |
|  |  |  |  |  |  |  |  |  |  |
| ***Supplier*** | *SupplierID* | Unique identifying number to identify the supplier of the BTDrone part. | AutoNumber | - | No | No | No | CK | - |
| *SupplierName*  (Composite Attribute) | The name of the Supplier who is going to supply the BTDrone Part | Character (200) | Name | Yes | No | No | No | - |
| SupplierPhoneNo | The phone number of the supplier tom contact them | Character (30) | - | No | No | No | No |  |
| SupplierCompanyName | The name of the supplier’s company | Character (100) | CompanyName | No | No | No | No |  |
| SupplierEmail | The email address associated with the supplier | Character (100) | - | No | No | No | No |  |
| SupplierAddress  (Composite Attribute) | The address of the supplier which include city, suburb, zip code, street number, and street name | Character (150) | - | Yes | No | No | No |  |
| SupplierDetails | The details and information of the seller | Character (150) | - | No | No | No | No | - |
|  |  |  |  |  |  |  |  |  |  |
| ***MaintenancePart*** | MaintenancePartDetails | The details of the part details that has been maintained | Character (150) | - | No | No | No | No | - |
|  |  |  |  |  |  |  |  |  |  |
| ***Salesperson*** | *SalesPersonID* | Unique identifying number to identify the salesperson | AutoNumber | - | No | No | No | CK | - |
| SalesPersonDetails | Details of the salesperson | Character (350) | - | No | No | Yes | No | - |
| ShopName | The shop name of the salesperson | Character (30) | - | Yes | No | Yes | No | - |
|  |  |  |  |  |  |  |  |  |  |
| ***Technician*** | *TechnicianID* | Unique identifying number to Technician | AutoNumber | - | No | No | No | CK | - |
| TechnicianJobTitle | The title of the job the technician is holding | Character (30) | - | No | No | Yes | No | - |
| TechnicanWorkDetails | Work Details of the Technician | Character (450) | - | No | No | Yes | No | - |
|  |  |  |  |  |  |  |  |  |  |
| ***AdministrativeExecutive*** | *AdminID* | Unique number to identify the Administrator | AutoNumber | - | No | No | No | CK | - |
| AdminDetails | Details associated with the admin | Character (150) | - | No | No | Yes | No | - |
| AdminPermissions | The permission to let the subscriber or contractor to have access to the BTDrone | Character (10) | - | No | No | No | No | - |
| AdminJobTitle | The title of the job the admin is holding. | Character (50) | - | No | No | No | No | - |
| AdminSubscriptionPrice | Price of the subscription that admin can change | Character (30) | - | No | No | No | No | - |
|  |  |  |  |  |  |  |  |  |  |
| ***StaffMember*** | *StaffID* | Unique number to identify the staff of Being There | AutoNumber | - | No | No | No | CK | - |
| StaffAddress  (Composite Attribute) | The address of the Staff | Character (250) | - | Yes | No | Yes | No | - |
| StaffJobTitle | The title of the job of the staff in Being There | Character (30) | - | No | No | NO | No | - |
| StaffName  (Composite Attribute) | The name of the staff which includes the first name and the last name. | Character (200) | - | Yes | No | Yes | No | - |
| StaffPhoneNo | The Phone number of the staff | Character (20) | - | No | No | No | No | - |
| StaffDetails | Details of the staff associated with the Being There | Character (150) | - | No | No | Yes | No | - |
|  |  |  |  |  |  |  |  |  |  |
| ***Account*** | *AccountID* | Unique Identifying number to identify the account | AutoNumber | - | No | No | No | CK | - |
| AccountAddress  (Composite Attribute) | It is the address of the account holder | Character (250) | - | Yes | No | NO | No | - |
| AccountName  (Composite Attribute) | The name of the account holder | Character (200) | - | No | No | Yes | No | - |
| AccountPhoneNo | The contact number of the account holder | Character (50) | - | No | No | No | No | - |
| AccountDetails | The details associated with the account | Character (250) | - | No | No | Yes | No | - |
| AccountEmail | The email associated with the account | Character (50) | - | No | No | Yes | No | - |
|  |  |  |  |  |  |  |  |  |  |
| ***Subscriber*** | *SubscriberID* | Unique identifying number of the subscriber | AutoNumber | UserID | No | No | No | CK | - |
| SubscriberPaymentDetails | The payment details which include card number and other information to pay for the subscription | Character (150) | - | No | No | No | No | - |
| SubscriberPassword | The password which is unique and the helps the subscriber to login and know about the subscription or to change the subscription. | Character (50) | UserPassword | No | No | No | No | - |
| SubscriberDiscount | The discount offered to the subscriber which is measured in percentage | Decimal | - | No | No | Yes | No | - |
|  |  |  |  |  |  |  |  |  |  |
| ***Subscription*** | SubscriptionType | The type of subscription the users or subscribers are going to choose | Character (30) | - | No | No | No | NO | - |
| SubscriptionDetails | The details and information associated with the subscriber. | Character (150) | - | No | No | Yes | NO | - |
| SubscriptionPrice | The Price of the subscription that the AdministrativeExecutive has assigned for different subscription(s). | Character (30) | - | No | No | No | No | - |
|  |  |  |  |  |  |  |  |  |  |
| ***Standard*** | StnSubName | Name of the standard subscriber | Character (50) | - | No | No | No | No | - |
|  | StndFee | The fee associated with the standard subscriber | Character (20) | - | No | No | No | No | - |
|  | StnDetails | The details of standard subscription | Character (150) | - | No | No | No | Yes | - |
|  |  |  |  |  |  |  |  |  |  |
| ***Gold*** | GoldSubName | Name of the gold subscriber | Character (50) | - | No | No | No | No | - |
|  | GoldFee | The fee associated with the gold subscriber | Character (20) | - | No | No | No | No | - |
|  | GoldDetails | The details of gold subscription | Character (150) | - | No | No | No | Yes | - |
|  |  |  |  |  |  |  |  |  |  |
| ***Platinum*** | PlatinumSubName | Name of the platinum subscriber | Character (50) | - | No | No | No | No | - |
|  | PlatinumFee | The fee associated with the platinum subscriber | Character (20) | - | No | No | No | No | - |
|  | PlatinumDetails | The details of platinum subscription | Character (150) | - | No | No | No | Yes | - |
|  |  |  |  |  |  |  |  |  |  |
| ***SuperPlatinum*** | SupPlatSubName | Name of the super platinum subscriber | Character (50) | - | No | No | No | No | - |
|  | SupPlatFee | The fee associated with the super platinum subscriber | Character (20) | - | No | No | No | No | - |
|  | SupPlatDetails | The details of the super platinum subscription | Character (150) | - | No | No | No | Yes | - |
|  |  |  |  |  |  |  |  |  |  |
| ***BTWebsite*** | BTURL | The address of the ‘Being There’ website | Character- (150) | - | No | No | No | CK | - |
| BTWebsiteDetails | The details about the website | Character (1000) | - | No | No | Yes | No | - |
|  |  |  |  |  |  |  |  |  |  |
| ***VideoStream*** | VideoID | The unique identifying number to identify the video | AutoNumber | - | No | No | No | CK | - |
| VideoLength | The time of the video that will take to play till end | Character (30) | - | No | No | No | No | - |
| VideoQuality | The quality of the video | Character (15) | - | No | No | No | No | - |
| VideoDetails | The additional information associated with the video | Character (150) | - | No | No | Yes | No | - |
| VideoTime | The time when the video was shot | Date/Time | - | No | No | No | No | - |

# Reasoning and Purpose of all Parts of the Model

The following project is about deploying BTDrones and collecting information for Government purpose. Information are collected by deploying the drones in different regions. While, these drones travel across different regions, they are controlled by different people behind the scene. For that, we had started working on our Conceptual Diagram, that depicts the way the drones are operated and how people are linked with the drones.

BTDrone is created as an entity where:

* BTDrone is maintained by the Technician, if associated with any technical difficulties. Here, Technician is a strong entity, who is a part of the staff hired by Being There organization. The Technician does the Maintenance, where the information regarding the date of maintenance and the type of maintenance is required. The job of technician is also to order certain parts from different suppliers for the BTDrones, if any part is required for a drone.
* The MaintenancePart is created as an entity which has been formed due to the Maintenance, as MaintenancePart will keep record of all the details done during the maintenance of the BTDrone(s).
* Different Suppliers will send BTDrone parts when asked by the Technician.
* The SalesPerson is created as an entity who will help to subscribe the audience to different subscription. The SalesPerson will sale the subscription either through BTWebsite or through retailing shop. The SalesPerson thus will have different IDs to sale the subscriptions to different users.
* The users are the subscribers, who could be either standard, gold, platinum, or super platinum.
* The standard subscribers will have the access to watch the video stream shot by the BTDrone(s), whereas the Gold Subscribers will have the access to watch the video stream through different controls views.
* The Platinum Subscribers can have control movement over BTDrones. Again, the SuperPlatinum Subscribers BTDrones.
* The contracts are created by the contractor, having unique ContractorID. Having contract, allows the SuperPlatinum Subscribers to deploy BTDrones over different specified bounded region that is required by the contracting organization.
* Each BTDrone is bounded by different RoamingZone. The RoamingZone has its own unique RoamingZoneID. Each RoamingZone has its own Maximum Altitude, Minimum Altitude, Longitude, and Latitude, keeping that in mind that each RoamingZone is no smaller than 100 cubic meters.
* Each BTDrone has access to different region and each region is connected to different terrains. The BTDrones can be configured to go to any terrain which include Jungle, Forest, Savannah, Extreme Cold – Ice and Snow, Mountain, Desert, and Urban deployment.
* An administrative executive is a staff who is hired by the “Being There” organization, so that they can have the ability to change the price of the subscription if required. Additionally, the Administrative Executive has the ability to enter the contract details and record the subscribers’ details.

# Rationale

## BTDrone

BTDrone is going to store the data of the BTDrone. Every BTDrone is going to be deployed by the ‘Being There’ organization. Each BTDrone needs to produce one BTDrone Data. The BTDrone is bounded by the Roaming Zone, as each BTDrone is restricted by at least one Roaming Zone. One BTDrone is configured by one or more Terrain(s). Apart from that, each BTDrone must relate to the supplier, as they need to what parts the BTDrone needs if it needs maintenance. That’s why, BTDrone needs to be maintained by the Technician. Every Contract needs to deploy at least one BTDrone. Furthermore, each Platinum subscriber can control the movement of one BTDrone whereas each Super Platinum subscriber can have complete control over one BTDrone.

## BTDroneData

Data is transmitted from each BTDrone to BTDroneData. It stores the information of each BTDrone which includes the temperature, humidity, and light strength the drone is facing. The information accumulated by the BTDroneData must be sent to one or more contractor(s).

## RoamingZone

Roaming Zone is the destination each BTDrone covered while hovering. That’s why one Roaming Zone must be bounded by at least one BTDrone.

## BTDronePart

It is the part that the BTDrone requires when it is associated with technical difficulties. That means, the part of a specific BTDrone needs to have a unique ID so that supplier knows what part needed to be sent if needs fixing by the Technician.

## Terrain

Terrain is lay of the land that is expressed in terms of elevation, slope, and orientation, where the drone drifts around and is configured by it. Terrain can be of many types that’s why we are identifying each terrain with ID of its own. Each terrain has also one region.

## Region

Region is where each BTDrone has its allocated area, that is specified by a specified boundary. Each region is identified with a unique identifier number. Apart from that, each region may have one or more BTDrone(s) and one region may have one or more terrain(s).

## Supplier

Data is also stored in Supplier, like all other entities. Supplier is required to supply parts to the Technician for each BTDrone, if the drone needs maintenance. The supplier needs to have idea about each of the BTDrone parts. That’s why each drone part needs to have its unique so that each supplier can evidently provide the technician with the part if required. The supplier also needs to have his own unique id of its own.

## Maintenance

Maintenance is undertaken by a technician when a BTDrone needs fixing. Maintenance keep records of the date regarding the maintenance.

## MaintenancePart

When maintenance is done by a technician for a specified BTDrone, the MaintenancePart keeps a record of the part that has undergone maintenance.

## Technician

A technician is a kind of staff who has been hired by the organization, ‘Being There’. Their purpose is to fix any issue related with the BTDrone, if needs any repair or fixing.

## StaffMember

Staff Member represents the general member of the organization ‘Being There’. The staff member can be either a technician, a salesperson, or an administrative executive. They are uniquely identified by a StaffID.

## SalesPerson

SalesPerson are hired to sale the subscription to different subscribers. SalesPerson is a type of StaffMember. Each salesperson can sale the subscription with the help of BTWebsite or they can sale the subscription from stores.

## Account

An account is created to keep the record of BTDrone users which include the contractors and the subscribers. A user needs to have an account id and account name to have access to the account.

## Contractor

Contractor provides the contract for each BTDrone. Each contractor needs to be recorded by the account, that means each contractor needs to have an account to be the user of BTDrone.

## Subscriber

A subscriber is a person who has arranged to receive or access to a service. In this regard, a subscriber gets excess to use the BTDrone service as required. Each subscriber is uniquely identified by a subscriber ID.

## Subscription

Subscription is where the information of each subscriber is recorded. Subscription can be of many types and each subscription type has its own price and own function.

## AdministrativeExecutive

Being There organization employs staffs who can take the role of Administrative Executive. They have the role to change the subscriptions and they are the only staff member of Being There who can change the price of BTDrone. They are also the one who can edit contract details if required.

## Standard

A standard is type of subscription for BTDrone who can only have excess to the video stream of a BTDrone. These subscribers can view one or more of the video streams.

## Gold

A gold is another type of subscriber who not only can view the subscriber, but they also can view the video stream by a BTDrone in different angles.

## Platinum

Platinum is a type of subscriber who can control movement of one or more BTDrones. They can control the movement of BTDrones in any angle they want.

## SuperPlatinum

Platinum subscribers can have the complete control of the BTDrones. They can control any number of BTDrones the way they want it.

## VideoStream

Video Stream is the video recorded by each BTDrone. The subscribers can have excess to the video recorded by the BTDrone. Each video is uploaded in the BTWebsite.

## BTWebsite

This the website where subscription is sold to the subscribers and it is also the place where the video recorded by each BTDrone is uploaded and can be excessed by the subscribers.

# Assumed Business Rules (with reasoning)

While I was working on the BTDrone Conceptual diagram there are few things that I had assumed and continued working on the project. The certain aspects that I assumed, or I had taken into consideration includes:

* The Droving Zone to be the same as Roaming Zone, that’s why an entity has been created, naming it RoamingZone, where its alias is DrovingZone.
* It has been assumed that a Staff could play a role as either as Administrative Executive, Technician, or Salesperson.
* The technician is someone who oversees the BTDrone and serve the role of fixing it and order BTDrone part if needed.
* The Salesperson is going to sale the subscription to the subscribers and they will do so either through online or through any retail shop.
* The administrative executive is the staff who might be able to change the contact details.
* BTWebsite is an entity which is identified by the unique URL link of the website. In this project the website is going to provide access to video streaming to the subscribers. Furthermore, salesperson can sale subscriptions to the audience through this website.
* Administrative Executive has ability to record both the subscribers and the contractor. For that, both the subscribers and the contractors must connect to an account where they have their unique AccountID and AccountName. This information can be excessed by the Administrative Executive.
* The Technician does the Maintenance; thus, Maintenance is created as an entity which depends on the Technician. It creates an unique identifying ID for each maintenance and stores the information about the date and time the maintenance of the BTDrone(s) has been performed.
* The Maintenance keeps record of the BTDrone part that needed fixing or done fixing; thus, created another separate entity naming it as MaintenancePart.

# Bibliography

Cochrane, T. (2019, March). ER Diagram. Nelson.

*Concepts of ER Data Model*. (n.d.). Retrieved from Tutorial Cup: https://www.tutorialcup.com/dbms/basic-concept-er-data-model.htm

Connolly, T. (2005). Entity Relationship Modeling . In T. Connolly, *Database Systems* (pp. 362-363). London: Pearson.

*Database Management System | ER Model*. (2018). Retrieved from Geeks for Geeks: https://www.geeksforgeeks.org/database-management-system-er-model/

Onsman, A. ( 2018, July 24). *Strong and Weak Entity*. Retrieved from Tutorial Point: https://www.tutorialspoint.com/Difference-between-Strong-Entity-and-Weak-Entity

5/9/2019

By Muhammad Ashkar Yousuf Milton

Being There. Conceptual Model, Logical Model, & Physical Model

Milestone 1, 2, and 3

# Contents

[Contents 1](#_Toc5543027)

[1. Executive Summary 2](#_Toc5543028)

[2. Conceptual to Logical Model 3](#_Toc5543029)

[3. Crow’s foot Notation 9](#_Toc5543030)

[4. Images and Links to The Logical Model of Being There 10](#_Toc5543031)

[5. The ERD model of Being There included with Crows Foot Notation 11](#_Toc5543032)

[6. Rationale for ERD 12](#_Toc5543033)

[7. Normalization 13](#_Toc5543034)

[8. Normalization of Being There’s Data 23](#_Toc5543035)

[9. Data Dictionary 0](#_Toc5543036)

[10. Introduction to NaLER Analysis 4](#_Toc5543037)

[11. NaLER Analysis of Logical Model of Being There 6](#_Toc5543038)

[12. Conclusion 18](#_Toc5543039)

[13. Bibliography 19](#_Toc5543040)

# Executive Summary

Milestone 2 is included with all the data, information, and the diagrams associated with the Logical Model, which has been transformed from the Conceptual Model of Being There Database. This milestone provides the audience with explanation of each ER model and the way they make this Logical Model possible. I have provided this milestone with appropriate logical diagram and rationale of the ERD model. Furthermore, I also have provided this milestone with the reasons for every part of this model that helped me to accomplish this milestone.

This report is provided with description normalization which is included with description of First Normal Form, Second Normal Form, Third Normal Form, Boyce-Codd Normal Form, Fourth Normal Form, and Fifth Normal Form.

I also included this report with data dictionary that shows which describes the number of rows an entity will start with in the database and how much we will be expected to grow over the year. Additionally, a data dictionary of the attributes is documented that shows the Relation Name, Attribute, Description, Data Type, Length, Value Range, Validation Rules, Default Value, Nulls, Key, Reference Entity, and Integrity Constraints.

Furthermore, a description of NaLER analysis had been included in this report, which shows the meaning and information related with NaLER analysis and NaLER analysis of my ERD model.

# Conceptual to Logical Model

#### Mapping Conceptual to Logical

Conceptual model is quite different from logical model. In the conceptual model we implemented the database by hiding the physical storage and targeting the entities, relationships, and constraints briefly and from those entities, relationships, and attributes the important ones were identified. However, in the logical model we will be concentrating on all the information associated with Being There database, while we introduce to some features. These include:

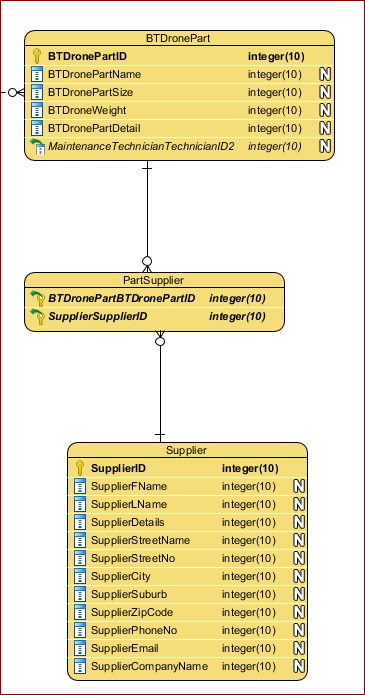
* Identifying all the entities and relationships within themselves
* Specifying all the attributes for each entity of Being There.
* Specifying the Primary Key for each entity
* Identifying the Foreign Key, that recognizes the relationship between different types of entity.
* Normalizing as best as possible.

While we will be working on the mapping of conceptual to logical model, we need to be aware of certain rules and regulations between these two, models that would make the logical model worthwhile.

#### Relations

During the mapping of the mapping of conceptual to logical model, each entity will form a relationship within the logical model. During the conceptual modeling we can have many-to-many relationships, however, in the logical model to prevent many to many relationships, we are going to create join table to eliminate the, many to many relationships. Thus, the join table will be included with one to many relationships.

All the relationships we have identified in the conceptual model will be implemented in the Being There database either with the help of a new table or by forming new column.



#### Attributes

When we mapped from conceptual to logical model, all the entities that we included in the conceptual model, will be added to each relation. The attributes that are simple are going to be added to its designated entity and will remain the way it is (Cochrane, 2019), whereas the composite that are composite entity are going to form two or more other attributes in the logical model.

Furthermore, the multi-valued attributes will be decomposed into separate columns so that they can perform certain relation within a specified attribute. And when we are mapping to logical model, we need to be aware of these certain relationship types. These include:

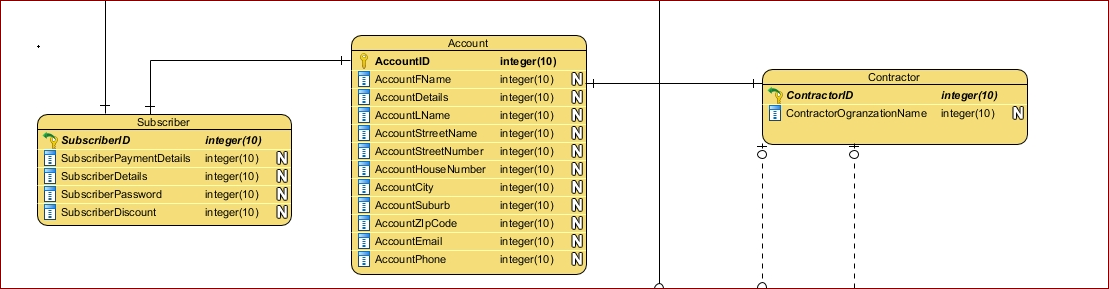
* Approaching the Foreign Key by choosing the primary key from an entity - A and then making a foreign key in another entity - B referencing the primary key of that entity - A.
* Merging the relation option by merging both of those entities that will corresponds to the row in the other entity.
* Cross referencing the relation by creating a thirds table, that is included with foreign keys from the primary keys of both entities (Elgabry, 2016).

#### Keys

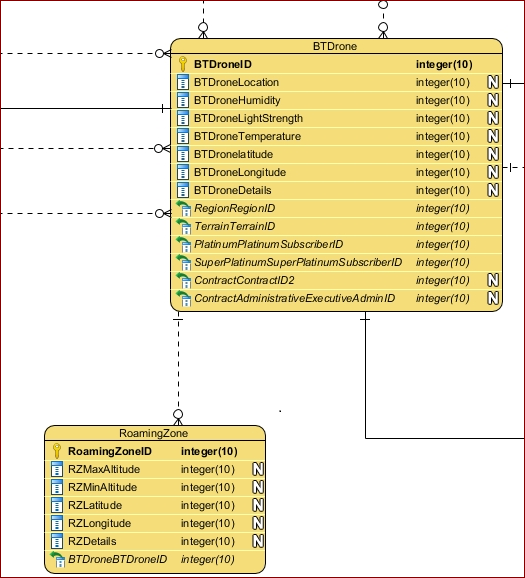
In the Logical Model, the unique identifier might or might not have a possibility to become the Primary Key. This is dependent based on test included by the criteria which is used to determine whether that primary key is suitable or not. To be precise these factors include: whether it is minimal, stable, and meaningful. Again, we are going to approach the Foreign Key by choosing the primary key from an entity - A and then make a foreign key in another entity - B referencing the primary key of that entity – A. This will create a relationship between two of these entities and thus will allow a flow of data within themselves, in Being There database.

Additionally, in the relationship model we will be implementing several rules and regulations using the foreign keys. These include:

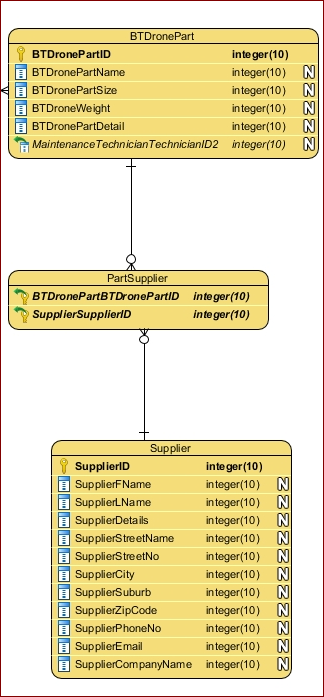
* In a one-to-one relationship, we can place the foreign key in any relation. That is, the primary key of relation A may be placed in relation B as a foreign key, or the primary key of B may be placed in relation A as a foreign key (Cochrane, 2019).
  + Example of One-To-One relationship is:



* In a one-to-many relationship, the primary key from the “one” side of the relationship is placed in the “many” relations as a foreign key (Cochrane, 2019).
  + Example of One-To-Many relationship is:



* A many-to-many relationship must be resolved using an associative entity/relation. This contains the primary keys of the participating relations as the foreign keys. Together, the foreign keys make up the primary key. This relation exists to resolve the issue of representing Multi-dimensional data in a two-dimensional format and is not necessarily required in other database models (Cochrane, 2019).
  + An example of Many-To-Many Relationship is:



* Relationship with attributes- as there is no such thing as relationship that exists in the relational model (only the PK-FK link), relationships with attributes can’t exist. To resolve this issue, a new relation must be created. The PK-FK links will be the same as for an associative relation, but this will also include the attributes that were previously linked to the relationship. We need to create an association type. The common attributes will be in the generalisation (Cochrane, 2019).

#### Relationship Containing Attributes

While we worked on the conceptual model, we found out that there are relationships that are many to many relationships. However, when we will be modeling the logical model, we can’t have many-to-many relationships. That’s why we are going to eliminate the many-to-many relationships by creating a join table between them. This time the new relationship will one-to-many relation, where the new table will have the Foreign Key, which is the primary key of the original entity table. This key in the new table will be the composite primary key, where all the key attributes of the two primary entities will be associated.

#### Super Entities / Classes

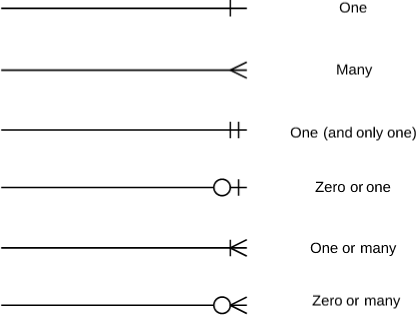
There are 3 different types of options that we can adopt for converting the super type and subtype into relations. These include:

* Creating one relation for the super type that holds all the common attributes and one relation for each subtype that holds the specific attributes for that subtype.
* Creating one relation for each subtype that holds all the common attributes and the specific attributes for that subtype. There will be no relation for the super type.
* Creating one relationship for the super type that holds all the common attributes and all the specific attributes of all the subtypes. There will be no relations for the subtypes (Cochrane, 2019).

# Crow’s foot Notation

Crow’s foot notation is quite like the cardinality that we did in conceptual model. However, in Crow’s foot we will be using the Chen Notation which will show the relationship between different entities within the database. In Crow’s foot diagram, each entity is considered as a box and the relationships are considered as lines between these boxes. So, different shapes at the end of these lines in going to determine the cardinality of these relationship.

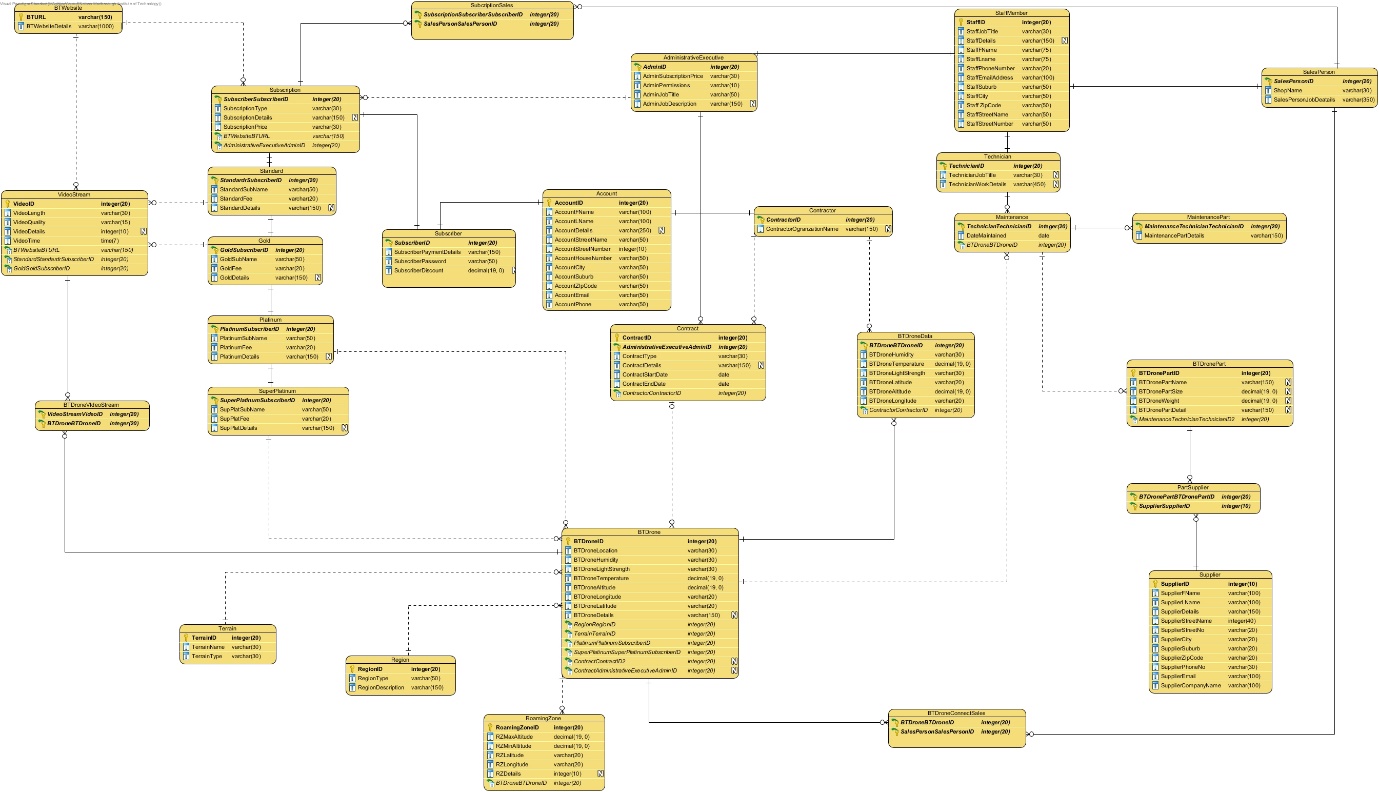
The types of Crow’s Foot include:



<https://user-images.githubusercontent.co>

# Images and Links to The Logical Model of Being There

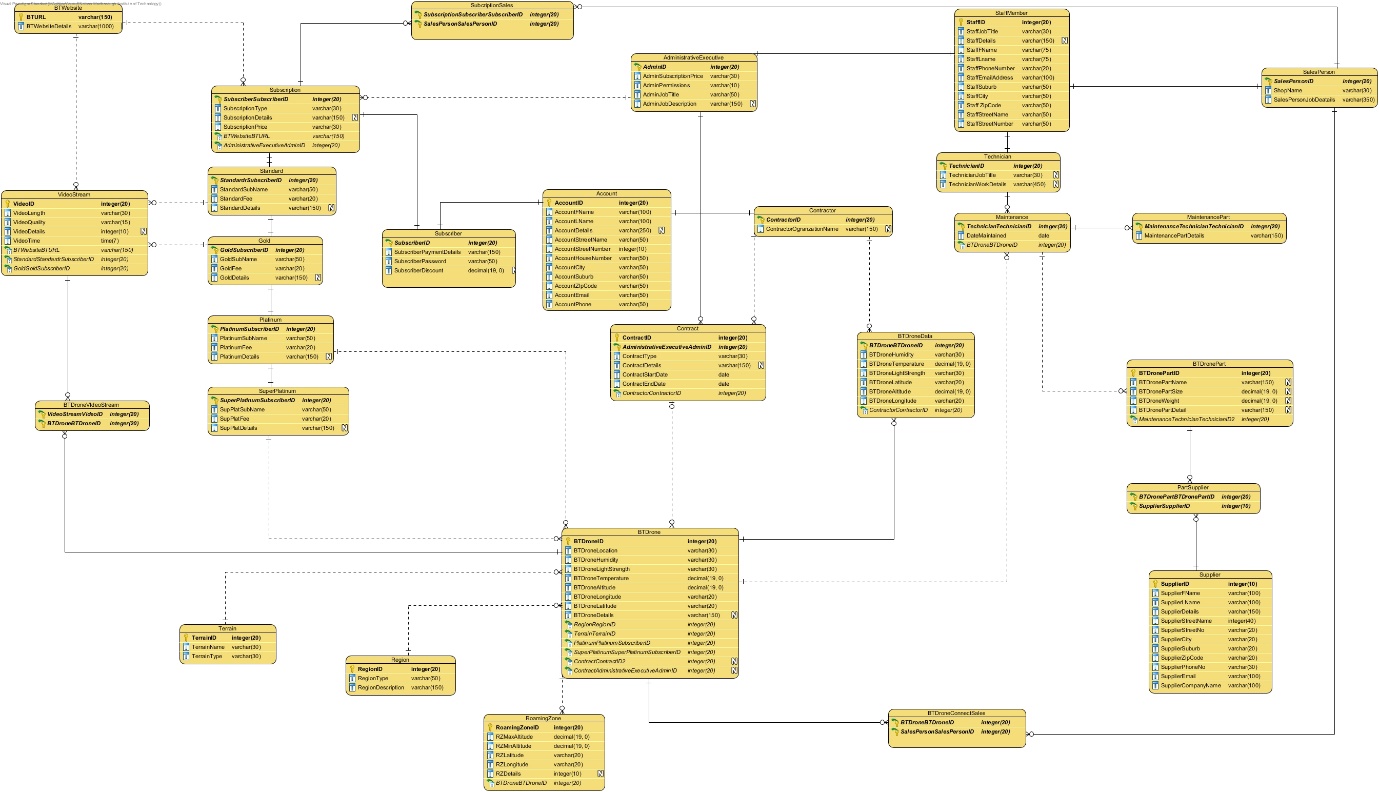
#### Image of the Logical Model



#### Links of the Logical Model

<https://drive.google.com/file/d/1wWm66OaMvPuCRlzOpno6q5VuEOkhgf0J/view?usp=sharing>

# The ERD model of Being There included with Crows Foot Notation

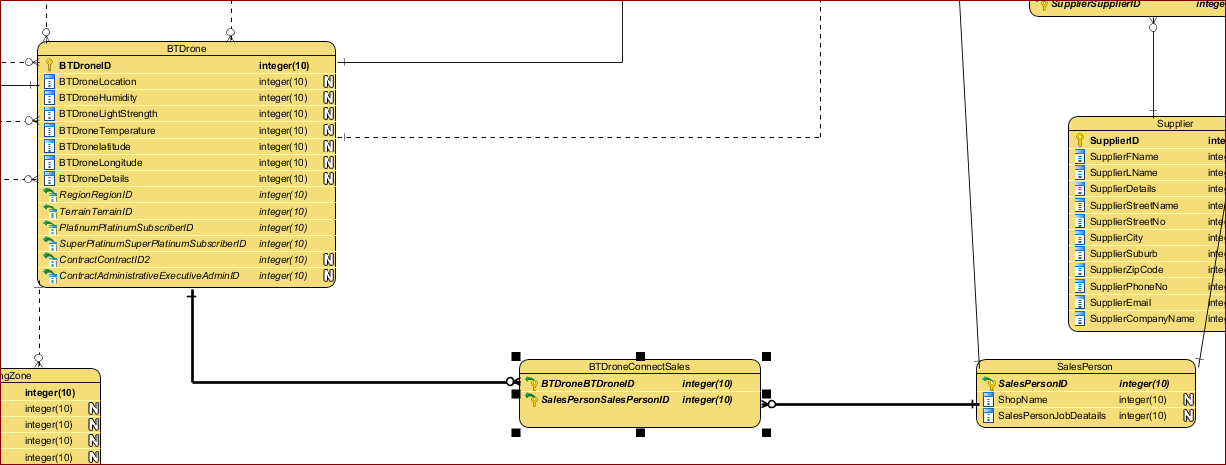


# Rationale for ERD

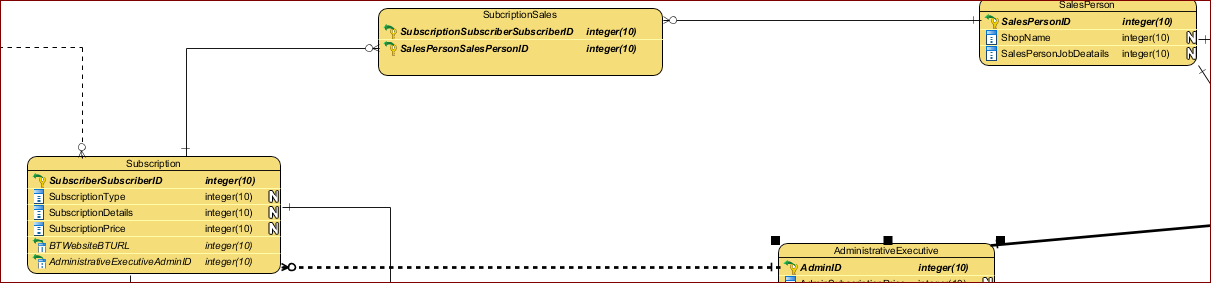
In the logical model I used all the information from my conceptual model and converted them to be used in the logical model. For that I used Visual Paradigm to build entity tables and attributes included in each table. I also used Crow’s Foot Notation to explain the relationship between each entity. Some entities are included with Primary Keys and again some entities are included with the Foreign Keys. These foreign keys’ values must always exist in the primary keys. That’s why, in the logical model there shouldn’t be any foreign hey that doesn’t include in the primary key. These foreign keys couldn’t be unique as they are included with different values. However, the data types of these Foreign Keys must match with the data types of the Primary Keys.

While working on the logical model I found there are few Many-to-Many relationships, which include:

* Many BTDrones connects to Many SalesPerson



* Many SalesPerson sales Many Subscription



That’s why to map these relationships, I created a new table so that I can connect the BTDrones with the SalesPerson and SalesPerson with Subscription. To do that, I started by adding two foreign keys, in each new table and each of these foreign keys will point towards the Primary Keys of those two entity tables.

Furthermore, I tried my best to normalize the database as best as possible. As such, the entity tables Account and Contract are normalized so that they don’t have any redundancy or repeated information.

Again, there are few composite attribute that has been broken into different attributes and put in their designated entity tables.

# Normalization

It is a process of organising relations and attributes of a relational database to improve data integrity and to eliminate data redundancy and inappropriate functional dependencies. Before we normalise we need to focus on certain criteria. These include:

* Keys
* Functional dependency
* Anomalies

Edgar F. Codd was the person who developed the relational model, thus proposed the theory of normalization. He also proposed the 3 initial normal forms and the Boyce-Codd normal form.

Accroding to the theory of normalization, it is a database design technique which organizes the table in such a way that reduces the redundancy and dependency of data as less as possible. It also divides smaller tables into smaller table, then later connect those using the relationships (Guru99, n.d.).

In order to normalize a database, we need to test a relational schema to certify if it belongs to a certain normal form. This involves arranging attributes based on dependencies between the attributes ensuring that it is properly ensuring that it is properly enforced by database integrity constraints. This is accomplished through synthesis and decomposition.

#### Unnormalized Form

It is also known as non-first normal form. It is also denoted by UNF. It is a simple database model that doesn’t provide us with the efficiency of database normalization. It has redundant data where we will be able to find multiple data or values, storing complex data structures within a single field or attribute (Codd).

#### First Normal Form

It is the property of a relation in a relational model database that follows certain schema, of the atomic data. The rules include:

* It shouldn’t contain columns with repeated or similar data.
* Columns have no repeating or similar data.
* Duplication of rows isn’t allowed
* Each field should have a unique name among other fields

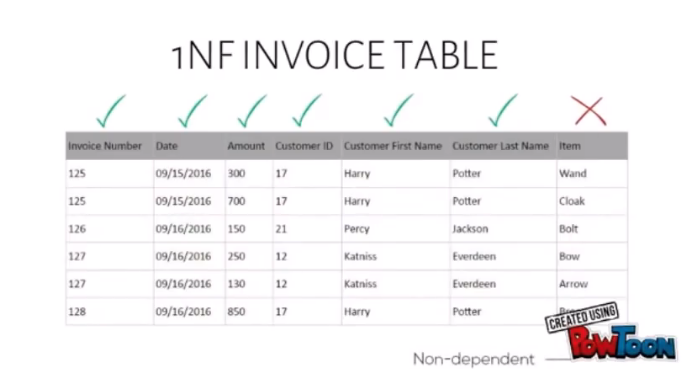
#### Second Normal Form

It is a normal form that is used as a relation in the relational database model and follows certain schema, which includes:

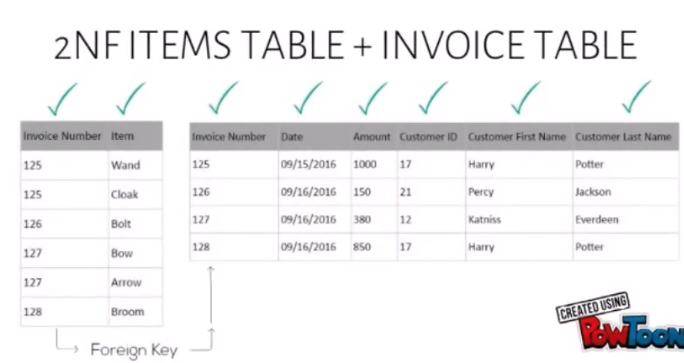
* Should be in the 1NF
* Attributes other than the primary key should be dependent on it.
* To this end, any table that is in 1NF that has a simple primary key is also considered in 2NF.

According to the information, I provided in my blog:

An invoice table is already in the first normal form, but the item column is not dependent on our primary key: Invoice Number. We can split that into another table.



We can simply link the invoice table and the items table using a foreign key to associate an item with the invoice.



#### Third Normal Form

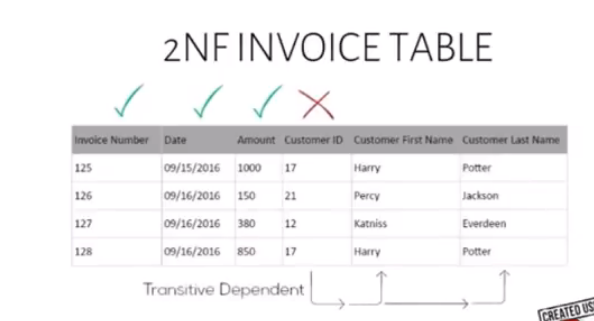
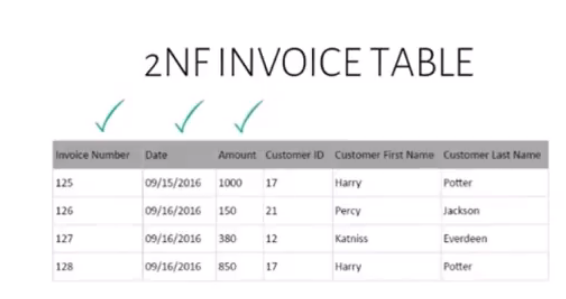
It is a normal form that is used normalizing a relational database model, reducing the number of redundancy and repetition. A relation or table is in third normal form when the following conditions are met. These include:

* Should be in second normal form
* No transitive dependencies exist.

Transitive dependency is a functional dependency between two or more non-key attributes in a relation

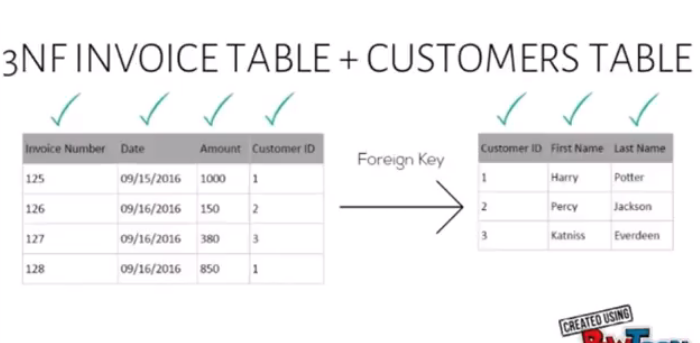
According to the information I gathered from the video (Link to video: <https://www.youtube.com/watch?v=v3N5PlbUHTs>) and the notes I wrote on my blog (Link to blog: <https://miltonsdat601journal.wordpress.com/2019/03/07/7-week-2-session-7/>):

In the invoice table, the invoice number determines the date, amount, customer ID, customer’s first name and last name. The customer ID column can also determine the customer’s last name and first name.



This is a transitive dependency and we need to separate these columns like we did in the second normal form (2NF).

We now have a customers’ table that links to the invoice table via the Customer ID. The schema is now in the third normal form (3NF).



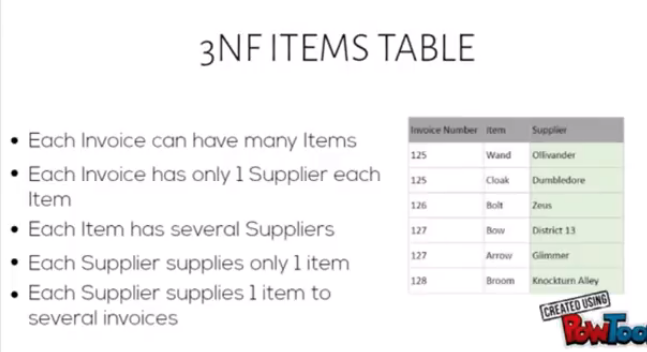
#### Boyce-Codd Normal Form

BCNF is a stronger version of 3NF and its objective is to address the anomalies found in 3NF. For a table to be in Boyce-Codd Normal Form, the rules for BCNF includes:

* The table should be in the third normal form.
* The table shouldn’t contain any trivial dependencies between keys except the primary key.
* According to the information I gathered from the video (Link to video: <https://www.youtube.com/watch?v=v3N5PlbUHTs>) and the notes I wrote on my blog (Link to blog: <https://miltonsdat601journal.wordpress.com/2019/03/07/7-week-2-session-7/>):

The invoice table used earlier has a new attribute called Supplier. This table confirms to these rules:

* Each invoice can have many items
* Each invoice has one supplier for each item
* Each item has several suppliers.
* Each supplier supplies only one item
* But, each supplier can supply the item to several stores (invoices).



Since an invoice can have many items and several suppliers for each item, having the invoice number as the only primary key is not enough.

To convert the table above to BCNF, we separated the item column from the original table and created a new table with Supplier and Item as the fields and link the two tables together using the Supplier attribute. The schema is now in Boyce-Codd Normal Form.



#### Fourth Normal Form

It is a database that is used in the normalization of relational database model and included with certain rules. They are:

* The database should be in the Boyce-Codd Normal form.
* It shouldn’t contain multivalued dependencies already because 4NF will add those.

According to the information I gathered from the video (Link to video: <https://www.youtube.com/watch?v=v3N5PlbUHTs>) and the notes I wrote on my blog (Link to blog: <https://miltonsdat601journal.wordpress.com/2019/03/07/7-week-2-session-7/>):

We can look at the Suppliers’ table. We have a new column called the Colour. For each item, there is a supplier for it and there’s a colour for that item.



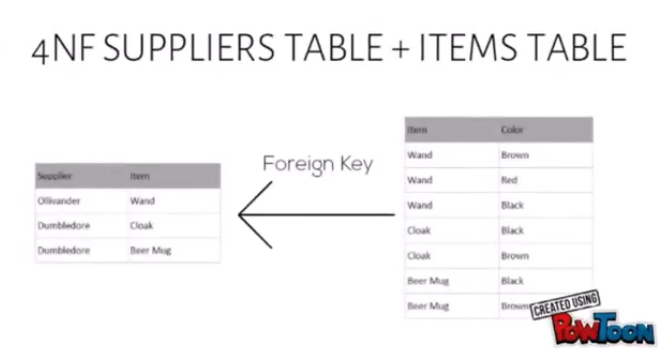
However, Supplier and Colour is independent from each other.



That’s why we split the table into two:

* The Suppliers table
* The Items table.

The Supplier table has the Supplier name and the item it carries, whereas the Item table contains the Item and the colour of that item. That this now in the 4th Normal Form (4NF).



#### Fifth Normal Form

This is the considered to be the last normal form of the normalization of the relational database model. The fifth normal form follows certain rules which includes:

* That a schema is in 4th Normal form.
* The attributes of the multivalued dependencies are related.

According to the information I gathered from the video (Link to video: <https://www.youtube.com/watch?v=v3N5PlbUHTs>) and the notes I wrote on my blog (Link to blog: <https://miltonsdat601journal.wordpress.com/2019/03/07/7-week-2-session-7/>):

This is our Supplier table, with a new column called the Shop Name. This is in 4NF, but not in 5NF because there is a lot of data duplication that’s happening. There is only solution now, and that is to break this table into three tables. 





We now have the Suppliers table, the Shop-Items table and the Shop-Suppliers table. Here, the Supplier table tells us what items a Supplier brings. The Shop-Items table tells us what Shop carries what item. And, the Shop-Suppliers table tells us what Shop is being supplied by what Supplier. The two new tables will be associated using Supplier and Item respectively. This is the 5th Normal Form (5NF).

# Normalization of Being There’s Data

Explain where you have done normalization and why did you do that?

EXPLAIN HOW YOU CAME TO THIS ...  
MAddress (mAddressID, mStreet, mPOBox, mMailCentre, mSuburb, mCity, mPostCode)

PAddress (pAddressID, pStreet, pSuburb, pCity, pPostCode)

Branch (BranchOfficeNo, bEmail, bPhone, bPhysAddress\*, bMailAddress\*)

Licensee (LicenseeID, LicName, LicPosition, LicMobPhone, LicHomePhone, LicAddress\*,     BranchOfficeNo\*, ManagerID\*)

SyndLicensee (SynID\*, LicenseeID\*)

Syndicate (SynID, SynLegalName, SynAuthPerson, SynDateAgree)

SynFund (SFAdvPerCent, SFGuaranteed, SFPercentageFee, SynID\*)

SFFunding (SFID\*, FundingID\*)

Funding (FundingID, FPerCentFund, FAmtToBeFund, FAdjustment, FDateFund, FDatePaid, DateRebatePd, FTDaysOut\*, PriceStruct\*)

Lead (LeadID, LeadAdvPerCent, LicenseeID\*)

FeeTable (FTDaysOut, PriceStruct\*, FTPercent)

Invoice (InvNumber, Client. BusinessID\*, InvAmt, InvDate, InvDueDate, CustPONumber, AMLSuspicious, FundingID\*, CustID\*)

Client (BusinessID, cBank, cBanker, cBankAcc, cOwner1, cOwner2, cOwner3, cOwner4, BusinessID\*, LicenseeID\*, PriceStruct\*)

Customer (BusinessID, AccPayContact, AccPayEmail, AccPayPhone)

Security (SecurityID, PPSRNum, PPSRPin, SecDateReg, SecDateRelease, SecUsed, CustID\*, BusinessID\*)

Business (BusinessID, bName, bStructure, bContact, bEmail, bPhone, bMobile, bCreditStat, bMailAdd\*, bPhysAdd\*)

StoredDoc (SDName, SDFileType, SDLocation)

WebDetails (Username, Password, LicenseeID\*, ClientID\*)

PriceStructure (PriceStruct, Description)

# 

# Data Dictionary

#### Document Relation

This table documents the Entities in the database, with their size and expected growth over time.

|  |  |  |  |
| --- | --- | --- | --- |
| Entity Name | No. of rows loaded at the beginning | Growth Per Year | Comments |
| BTDrone | 30,000 | 10% | The number of BTDrone would be increased due to the raise in subscribers and contractors |
| RoamingZone | 150 | 2% | It is highly unlikely to change due to the allocation of prices. |
| Region | 100 | 10% | As the BTDrone is going to influence more, the ‘Being There’ organization will allow more region to be covered |
| Terrain | 10,000 | 20% | More terrain will be covered according to the demand |
| BTDroneData | 3,000,000 | 20% | Will grow due to the transfer of more data by each BTDrone every second |
| BTDronePart | 2,000 | 10% | It depends on the maintenance performed on each drone |
| Contract | 1,000 | 15% | More contract may be formed due to increase in more Contractor. |
| Contractor | 1,000 | 15% | More contractor may sign up for having more contract over BTDrones |
| Maintenance | 15,000 | 20% | The more the BTDrones are deployed and taking working the more maintenance is required. |
| Supplier | 50 | 8% | The number of suppliers may increase as the influence of BTDrone will increase |
| MaintenancePart | 15,000 | 20% | The more the BTDrones are deployed and taking working the more maintenance is required. Thus, more information will be recorded. |
| SalesPerson | 20 | 0 | The supplier will remain the same |
| Technician | 50 | 5% | Depends on the number of maintenances the number of technicians may increase. |
| AdministrativeExecutive | 6 | 0 | The number will remain the same as the table size is highly unlikely to change |
| Subscriber | 50,000 | 20% | It depends on ‘Being There’ and they have influence on subscribers. |
| StaffMember | 200 | 10% | There are other staffs apart from administrative executive, salesperson, and technician who would be involved with ‘Being There’ organization. |
| Account | 5,000,000 | 25% | Will increase due to the number of subscribers and contractors. |
| Subscription | 4,000,000 | 20% | Will grow due to the increase in more subscribers. |
| Standard | 2000 | 10% | Will increase due to more popularity of BTDrone. |
| Gold | 200 | 0 | Will remain unchanged due to a limited number of gold subscriber |
| Platinum | 100 | 0 | Will remain unchanged due to a limited number of Platinum subscriber |
| SuperPlatinum | 50 | 0 | Will remain unchanged due to a limited number of Super Platinum subscriber |
| BTWebsite | 1 | 0 | Will be the only website to reach the customers. |
| VideoStream | 1,000,000 | 30% | Will increase to more video streaming by more BTDrones. |

#### 

#### Document Attributes

This table shows the attributes that make up each entity with a description of each.  The data types show how the data will be stored in a database and the length is the size of the data stored.  The value range is for pre-set fields and indicates the options available. Validation rules show what criteria data must fulfil to be entered.  Default value is the starting point for any data in the table. The null column is to show whether the data must be present when entering a database.  The Key identifies any primary keys, while the Integrity constraints identify those attributes that are either Primary or Foreign keys. The references entity column shows which table a foreign key will reference.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Entity Name | Attributes | Description | Data types | Length | Value range | Validation Rules | Nulls | Default values | Key | Reference Entity | Integrity Constraints |
| BTDrone | BTDroneID | Unique number of the drone | Integer | 20 | - | Unique | NO |  | PK |  |  |
| BTDroneLocation | The location of the BTDrone | Varchar | 30 | - |  | NO |  |  |  |  |
| BTDroneHumidity | Atmospheric Moisture the BTDrone is going to be in. | Varchar | 30 | - |  | NO |  |  |  |  |
| BTDroneLightStrength | Intensity of the light the BTDrone is going to face | Varchar | 30 | - |  | NO |  |  |  |  |
| BTDroneTemperature | The degree of intensity of heat the BTDrone is going to face in Celsius | Decimal | 19,0 | - |  | NO |  |  |  |  |
| BTDroneAltitude | Altitude is the vertical direction of the BTDrone from ground. | Varchar | 19,0 | - |  | NO |  |  |  |  |
| BTDroneLogitude | It is used to specify the precise location on the surface of the BTDrone | Varchar | 20 | - |  | NO |  |  |  |  |
| BTDroneLatitude | It is an angle which range from 0 degree at the Equator to 90 degree at the poles, for the BTDrone | Varchar | 20 | - |  | NO |  |  |  |  |
| BTDroneDetails | The additional details telling us more about the drone. | Varchar | 150 | - |  | Yes |  |  |  |  |
| RegionID | The region a drone is connected to | Integer | 20 | - |  | NO |  | FK | Region | FK |
| TerrainID | The terrain the drone is connected to | Integer | 20 | - |  | NO |  | FK | Terrain | FK |
| SubscriberID | The subscriber the drone is connected to | Integer | 20 | - |  | NO |  | FK | Subscriber | FK |
| ContractID | The contract the BTDrone is connected to | Integer | 20 | - |  | NO |  | FK | Contract | FK |
| AdminID | The admin the drone is connected to | Integer | 20 | - |  | NO |  | FK | AdministrativeExecutive | FK |
| RoamingZone | RoamingZoneID | Unique number to identify the Roaming Zone | Integer | 20 | - | Unique | NO |  | PK |  |  |
| RZMaxAltitude | Maximum Altitude the roaming zone is for the BTDrone | Decimal | 19,0 | - |  | NO |  |  |  |  |
| RZMinAltitude | Minimum Altitude the roaming zone is for the BTDrone | Decimal | 19,0 | - |  | NO |  |  |  |  |
| RZLatitude | Latitude of the BTDrone bounded by specific roaming zone | Varchar | 20 | - |  | NO |  |  |  |  |
| RZLongitude | Longitude of the BTDrone bounded by specific roaming zone | Varchar | 20 | - |  | NO |  |  |  |  |
| RZDetails | Information about the RoamingZone | Varchar | 150 | - |  | Yes |  |  |  |  |
| BTDroneID | The Drone that is connected to | Integer | 20 | - |  | NO |  | FK | BTDrone | FK |
| Region | RegionID | Unique Number to identify each region | Integer | 20 | - | Unique | NO |  | PK |  |  |
| RegionType | The type of earth surface that has unifying characteristics like climate and weather. | Varchar | 50 | - |  | NO |  |  |  |  |
| RegionDescription | The details associated with the region | Varchar | 150 | - |  | Yes |  |  |  |  |
| Terrain | TerrainID | Unique number to identify the terrain | Integer | 20 | - |  | NO |  | PK |  |  |
| TerrainName | The name of the terrain given to each terrain | Varchar | 30 | - |  | NO |  |  |  |  |
| TerrainType | The types of terrain | Varchar | 30 | - |  | NO |  |  |  |  |
| BTDroneData | BTDroneID | The drone that sends data to the data storage | Integer | 20 | - |  | NO | PK |  | BTDrone | FK |
| BTDroneDataHumidity | Atmospheric Moisture the BTDrone is going to be in. | Varchar | 30 | - |  | NO |  |  |  |  |
| BTDroneDataTemperature | The degree of intensity of heat the BTDrone is going to face in Celsius | Decimal | 19,0 | - |  | NO |  |  |  |  |
| BTDroneDataLightStregth | Intensity of the light the BTDrone is going to face | Varchar | 30 | - |  | NO |  |  |  |  |
| BTDroneDataLatitude | It is an angle which range from 0 degree at the Equator to 90 degree at the poles, for the BTDrone | Varchar | 20 | - |  | NO |  |  |  |  |
| BTDroneDataLogitude | It is used to specify the precise location on the surface of the BTDrone | Varchar | 20 | - |  | NO |  |  |  |  |
| BTDroneDataAltitude | Altitude is the vertical direction of the BTDrone from ground. | Decimal | 19,0 | - |  | NO |  |  |  |  |
| ContractorID | The contractor who has access to the drone | Integer | 20 | - |  | NO | FK |  | Contractor | FK |
| BTDronePart | BtDronePartID | Unique number of each part | Integer | 20 | - | Unique | NO | PK |  |  |  |
| BtDronePartName | Name of each drone part | Varchar | 150 | - |  | NO |  |  |  |  |
| BtDronePartSIze | Size of the part which includes height, length, and width | Decimal | 19,0 | - |  | NO |  |  |  |  |
| BtDronePartWeight | Weight of the part which can be measured in grams | Decimal | 19,0 | - |  | NO |  |  |  |  |
| BtDronePartDetail | Details associated with each part of the drone | Varchar | 150 | - |  | Yes |  |  |  |  |
| TechnicianID | The technician does maintenance of a drone and records the data | Integer | 20 | - |  | NO | PK |  | Technician | FK |
| Contract | ContractID | Unique number to identify the contract | Integer | 20 | - | Unique | NO | PK |  |  |  |
| AdminID | The admin who records the information of each contract | Integer | 20 | - |  | NO | PK |  | AdministrativeExecutive | FK |
| ContractType | The type of the contract | Varchar | 30 | - |  | NO |  |  |  |  |
| ContractDetails | Details and information associated with the contract | Varchar | 150 | - |  | Yes |  |  |  |  |
| ContractStartDate | The day the contract is going to start | Date/Time | - | - |  | NO |  |  |  |  |
| ContractEndDate | The day the contract is going to end. | Date/Time | - | - |  | NO |  |  |  |  |
| ContractorID | The Contractor creates a contract | Integer | 20 | - |  | NO | FK |  | Contractor | FK |
| Contractor | ContractorID | Unique number to identify the contractor | Integer | 20 | - | Unique | NO | PK |  |  |  |
| ContractorOrganizationName | The name of the contractor’s organization Name | Varchar | 150 | - |  | NO |  |  |  |  |
| Maintenance | TechnicianID | The Technician who does maintenance | Integer | 20 | - |  | NO | PK |  |  |  |
| DateMaintained | The date when the maintenance of the drone part is done | Date/Time | - | - |  | NO |  |  |  |  |
| BTDroneID | The drone whose maintenance has been perfomed | Integer | 20 | - |  | NO | FK |  | BTDrone | FK |
| Supplier | SupplierID | Unique identifying number to identify the supplier of the BTDrone part. | Integer | 20 | - | Unique | NO | PK |  |  |  |
| SupplierFName | Supplier first name | Varchar | 100 | - |  | NO |  |  |  |  |
| SupplierLName | Supplier last name | Varchar | 100 | - |  | NO |  |  |  |  |
| SupplierDetails | The details and information of the seller | Varchar | 150 | - |  | Yes |  |  |  |  |
| SupplierStreetName | The street name of the supplier | Varchar | 40 | - |  | NO |  |  |  |  |
| SupplierStreetNo | The street number of the supplier | Varchar | 20 | - |  | NO |  |  |  |  |
| SupplierCity | The city of the supplier | Varchar | 20 | - |  | NO |  |  |  |  |
| SupplierSuburb | The suburb of the supplier | Varchar | 20 | - |  | NO |  |  |  |  |
| SupplierZipCode | The zip code of the supplier’s location. | Varchar | 20 | - |  | NO |  |  |  |  |
| SupplierPhoneNo | The phone number of the supplier | Varchar | 20 | - |  | NO |  |  |  |  |
| SupplierEmail | The email address of the supplier | Varchar | 20 | - |  | NO |  |  |  |  |
| SupplierCompanyName | The name of the supplier’s company | Varchar | 100 | - |  | NO |  |  |  |  |
| MaintenancePart | TechnicianID | The Technician who does the maintenance | Integer | 20 | - |  | NO |  | PK | Technician | FK |
| MaintenancePartDetails | The details of the part details that has been maintained | Varchar | 150 | - |  | NO |  |  |  |  |
| SalesPerson | SalesPersonID | Unique identifying number to identify the salesperson | Integer | 20 | - | Unique | NO |  | PK |  |  |
| ShopName | Details of the salesperson | Varchar | 30 | - |  | NO |  |  |  |  |
| SalesPersonJobDetails | The shop name of the salesperson | Varchar | 150 | - |  | Yes |  |  |  |  |
| Technician | TechnicianID | Unique identifying number to Technician | Integer | 20 | - | Unique | NO |  | PK |  |  |
| TechnicianJobTitle | The title of the job the technician is holding | Varchar | 30 | - |  | Yes |  |  |  |  |
| TechnicianWorkDetails | Work Details of the Technician | Varchar | 450 | - |  | Yes |  |  |  |  |
| AdministrativeExecutive | AdminID | Unique number to identify the Administrator | Integer | 20 | - | Unique | NO |  | PK |  |  |
| AdminSubscriptionPrice | Details associated with the admin | Varchar | 30 | - |  | NO |  |  |  |  |
| AdminPermissions | The permission to let the subscriber or contractor to have access to the BTDrone | Varchar | 10 | - |  | NO |  |  |  |  |
| AdminJobTitle | The title of the job the admin is holding. | Varchar | 50 | - |  | Yes |  |  |  |  |
| AdminJobDescription | Price of the subscription that admin can change | Varchar | 150 | - |  | Yes |  |  |  |  |
| Subscriber | SubscriberID | Unique identifying number of the subscriber | Integer | 20 | - | Unique | NO |  | PK |  |  |
| SubscriberPayementDetails | The payment details which include card number and other information to pay for the subscription | Varchar | 150 | - |  | NO |  |  |  |  |
| SubscriberPassword | The password which is unique and the helps the subscriber to login and know about the subscription or to change the subscription. | Varchar | 50 | - |  | NO |  |  |  |  |
| SubscriberDiscount | The discount offered to the subscriber which is measured in percentage | Decimal | 19,0 | - |  | NO |  |  |  |  |
| StaffMember | StaffID | Unique number to identify the staff of Being There | Integer | 20 | - | Unique | NO |  | PK |  |  |
| StaffJobTitle | The title of the job of the staff in Being There | Varchar | 30 | - |  | NO |  |  |  |  |
| StaffDetails | Details of the staff associated with the Being There | Varchar | 150 | - |  | Yes |  |  |  |  |
| StaffFName | Staff’s First Name | Varchar | 75 | - |  | NO |  |  |  |  |
| StaffLName | Staff’s last name | Varchar | 75 | - |  | NO |  |  |  |  |
| StaffPhoneNumber | Staff’s Phone number to contact | Varchar | 20 | - |  | NO |  |  |  |  |
| StaffEmailAddress | Staff’s email address | Varchar | 100 | - |  | NO |  |  |  |  |
| StaffSuburb | Staff’s suburb of the address | Varchar | 50 | - |  | NO |  |  |  |  |
| StaffCity | The city the staff is residing | Varchar | 50 | - |  | NO |  |  |  |  |
| StaffZipCode | The Zip Code of the address | Varchar | 50 | - |  | NO |  |  |  |  |
| StaffStreetName | The street name the staff is. | Varchar | 50 | - |  | NO |  |  |  |  |
| StaffStreetNumber | The street number the staff is | Varchar | 50 | - |  | NO |  |  |  |  |
| Account | AccountID | Unique Identifying number to identify the account | Integer | 20 | - | Unique | NO |  | PK |  |  |
| AccountFName | It is the First Name of the account holder | Varchar | 100 | - |  | NO |  |  |  |  |
| AccountLName | It is the account holder’s last name | Varchar | 100 | - |  | NO |  |  |  |  |
| AccountDetails | It is the detailed information of the account | Varchar | 250 | - |  | Yes |  |  |  |  |
| AccountStreetName | It is the account holder’s street name | Varchar | 50 | - |  | NO |  |  |  |  |
| AccountStreetNumber | It is the account holder’s street number | Varchar | 50 | - |  | NO |  |  |  |  |
| AccountHouseNumber | It is the account holder’s house number | Varchar | 50 | - |  | NO |  |  |  |  |
| AccountCity | It is the account holder’s city | Varchar | 50 | - |  | NO |  |  |  |  |
| AccountSuburb | It is the account holder’s suburb | Varchar | 50 | - |  | NO |  |  |  |  |
| AccountZIpCode | It is the account holder’s zip code | Varchar | 50 | - |  | NO |  |  |  |  |
| AccountEmail | It is the account holder’s email address | Varchar | 50 | - |  | NO |  |  |  |  |
| AccountPhone | It is the account holder’s phone number | Varchar | 50 | - |  | NO |  |  |  |  |
| Subscription | SubscriberID | The subscriber who relates to the subscription | Integer | 20 | - |  | NO |  | PK | Subscriber | FK |
| SubscriptionType | The type of subscription the users or subscribers are going to choose | Varchar | 30 | - |  | NO |  |  |  |  |
| SubscriptionDetails | The details and information associated with the subscriber. | Varchar | 150 | - |  | Yes |  |  |  |  |
| SubscriptionPrice | The Price of the subscription that the AdministrativeExecutive has assigned for different subscription(s). | Varchar | 150 | - |  | NO |  |  |  |  |
| BTURL | The website where the subscription is sold, and the subscriber purchase it from | Varchar | 150 | - |  | NO |  |  | BTWebsite |  |
| AdminID | The admin who is the controlling the subscription and changing the price if required | Integer | 20 | - |  | NO |  | FK | AdministrativeExecutive | FK |
| Standard | SubscriberID | The subscriber who is having standard subscription | Integer | 20 | - |  | NO |  | PK | Subscriber | FK |
| StandardSubName | Name of the standard subscriber | Varchar | 50 | - |  | NO |  |  |  |  |
| StandardFee | The fee associated with the standard subscriber | Varchar | 20 | - |  | NO |  |  |  |  |
| StandardDetails | The details of standard subscription | Varchar | 150 | - |  | Yes |  |  |  |  |
| Gold | SubscriberID | The subscriber who is having gold subscription | Integer | 20 | - |  | NO |  | PK | Subscriber |  |
| GoldSubName | Name of the gold subscriber | Varchar | 50 | - |  | NO |  |  |  |  |
| GoldFee | The fee associated with the gold subscriber | Varchar | 20 | - |  | NO |  |  |  |  |
| GoldDetails | The details of gold subscription | Varchar | 150 | - |  | Yes |  |  |  |  |
| Platinum | SubscriberID | The subscriber who is having platinum subscription | Integer | 20 | - |  | NO |  | PK | Subscriber | FK |
| PlatinumSubName | Name of the platinum subscriber | Varchar | 50 | - |  | NO |  |  |  |  |
| PlatinumFee | The fee associated with the platinum subscriber | Varchar | 20 | - |  | NO |  |  |  |  |
| PlatinumDetails | The details of platinum subscription | Varchar | 150 | - |  | Yes |  |  |  |  |
| SuperPlatinum | SubscriberID | The subscriber who is having super platinum subscription | Integer | 20 | - |  | NO |  | PK | Subscriber | FK |
| SupPlatSubName | Name of the super platinum subscriber | Varchar | 50 | - |  | NO |  |  |  |  |
| SupPlatFee | The fee associated with the super platinum subscriber | Varchar | 20 | - |  | NO |  |  |  |  |
| SupPlatDetails | The details of the super platinum subscription | Varchar | 150 | - |  | Yes |  |  |  |  |
| BTWebsite | BTURL | The address of the ‘Being There’ website | Integer | 150 | - | Unique | NO |  | PK |  |  |
| BTWebsiteDetails | The details about the website | Varchar | 1000 | - |  | Yes |  |  |  |  |
| VideoStream | VideoID | The unique identifying number to identify the video | Integer | 20 | - | Unique | NO |  | PK |  |  |
| VideoLength | The time of the video that will take to play till end | Varchar | 30 | - |  | NO |  |  |  |  |
| VideoQuality | The quality of the video | Varchar | 15 | - |  | NO |  |  |  |  |
| VideoDetails | The additional information associated with the video | Varchar | 150 | - |  | Yes |  |  |  |  |
| VideoTime | The time when the video was shot | Date/Time | - | - |  | NO |  |  |  |  |
| BTURL | The website where the videos are uploaded and made accessible | Varchar | 150 | - |  | NO |  | FK | BTWebsite |  |
| SubscriberID | The subscriber who can have access to the video stream | Integer | 20 | - |  | NO |  | FK | Subscriber | FK |
| BTDroneVideoStream | VideoID | The video stream that is taken by the drone | Integer | 20 | - |  | NO |  | PK | VideoStream | PK, FK |
| BTDroneID | The drone that does video stream | Integer | 20 | - |  | NO |  | PK | BTDrone | FK, PK |
| BTDroneConnectSales | BTDroneID | The drone that needs parts and supplier sends parts according to its requirements | Integer | 20 | - |  | NO |  | PK | BTDrone | PK, FK |
| SalesPersonID | The salesperson who have access to the drone so that parts can be send when required. | Integer | 20 | - |  | NO |  | PK | SalesPerson | FK, PK |
| SubscriptionSales | SubscriberID | The subscriber who purchase the subscription from the salesperson | Integer | 20 | - |  | NO |  | PK | Subscriber | PK, FK |
| SalesPersonID | The sales person who sales subscription to the subscriber | Integer | 20 | - |  | NO |  | PK | SalesPerson | FK, PK |

#### 

#### Derived attributes

This table describes all the derived attributes, the entities they came from and how they can be calculated.

There is no table for derived attributed for this database

# 

# Introduction to NaLER Analysis

This is a Natural Language method for ER modelling.  It allows the users to create expressions about each instance of data which more easily allows comprehension about the way data will be used and stored in a database.  The NaLER analysis was intended to capture the meaning represented by the database constructs in a formal way to reduce individual interpretations and assumptions that can be projected on them.  NaLER requires primary key information in the relationship sentences and requires sentences about all the attributes of each entity (Atkins & Patrick, 2000).

There are five steps system that can be used to check whether the entities and attributes are in the correct order. This information is collected from the notes I made on my blog, (Link of Blog: <https://miltonsdat601journal.wordpress.com/2019/03/10/1-week-3-session-9/>) . This includes:

1. **Step-1:**

It is to identify and document the diagram conventions. To do this, we must:

* + identify the notation that has been used to construct the diagram and the data dictionary
  + and make a legend, which will help us to define each part of the model, thus aiding us to make an interpretation of the model.

1. **Step-2:**

It is to perform a syntax that checks the model in accordance with STEP-1 assumptions and making and marking any adjustments.

1. **Step-3:**
   * It is to construct a sentence for the primary key attribute(s) in the following format:

Each <Entity Name> is uniquely identified by the <Primary KEY>

* It is to construct a sentence for each attribute which doesn’t include a Primary or Foreign KEY in the entity format:

*One****<E-Name>****identified by***<PK>** must have one **<Attribute Name>**

* + It is to construct a sentence for each binary relationship, which includes recursive ones, that the entity participates in:

*One***<E-Name>***identified by***<PK>** **<Optionality> <Relationship-Name> <Cardinality> <E-Name>***identified by* **<Foreign KEY>.**

1. **Step-4:**

We need to populate the sentences from Step-3.2 and Step-3.3

1. **Step-5A:**

If an entity represents a ternary or higher degree Many-to-Many relationship and has a composite name, we need to construct a sentence with the format.

*One***<Entity Name 1>** *identified by****<Primary Key 1> <Optionality> <Relationship Name 1> <Cardinality> <Entity Name 2>****identified by***<Primary Key 2> <Relationship Name 2> <Cardinality> <Entity Name 2>***identified by***<Primary Key 3>**

1. **Step-5B:**

If an entity represents a ternary or higher degree Many-to-Many relationship and has clearly identified name and a single attribute key, we can construct a sentence within the format below.

*One* **<Entity Name 1>** *identified by* **<Primary Key 1>** *must***<Relationship Name 1>** *one* **<Entity Name 2>** *identified by***<Foreign Key 1>** **<Relationship Name 2>** *one* **<Entity Name 3>***identified by* **<Foreign Key 2>** *and* **<Relationship Name 3>** *one***<Entity Name 4>** *identified by* **<Foreign Key 3>.**

# NaLER Analysis of Logical Model of Being There

#### Attribute Sentences

1. Each Account is uniquely identified by AccountID
2. One Account (AccountID) must have one AccountFName
3. One Account (AccountID) must have one AccountLName
4. One Account (AccountID) may have one AccountDetails
5. One Account (AccountID) must have one AccountStreetName
6. One Account (AccountID) must have one AccountStreetNumber
7. One Account (AccountID) must have one AccountCity
8. One Account (AccountID) must have one AccountSuburb
9. One Account (AccountID) must have one AccountZipCode
10. One Account (AccountID) must have one AccountHouseNumber
11. One Account (AccountID) must have one AccountEmail
12. One Account (AccountID) must have one AccountPhone
13. Each Contractor is uniquely identified by one ContractorID
14. One Contractor (ContractorID) must have one ContractorOrganizationName
15. Each Contract is uniquely identified by ContractID, StaffID
16. One Contract (ContractID, StaffID) may have one ContractType
17. One Contract (ContractID, StaffID) may have one ContractDetails
18. One Contract (ContractID, StaffID) may have one ContractStartDate
19. One Contract (ContractID, StaffID) must have one ContractStartDate
20. One Contract (ContractID, StaffID) must have one ContractEndDate
21. One Contract (ContractID, StaffID) must have one ContractFName
22. One Contract (ContractID, StaffID) must have one ContractLName
23. One Contract (ContractID, StaffID) must have one ContractStreetName
24. One Contract (ContractID, StaffID) must have one ContractSteetNo
25. One Contract (ContractID, StaffID) must have one ContractHouseNumber
26. One Contract (ContractID, StaffID) must have one ContractEmail
27. One Contract (ContractID, StaffID) must have one ContractPhone
28. One Contract (ContractID, StaffID) must have one ContractSuburb
29. One Contract (ContractID, StaffID) must have one ContractCity
30. One Contract (ContractID, StaffID) must have one ContractZipCode
31. Each BTDroneData must be uniquely identified by BTDroneID
32. One BTDroneData (BTDroneID) must have one BTDroneHumidity
33. One BTDroneData (BTDroneID) must have one BTDroneTemperature
34. One BTDroneData (BTDroneID) must have one BTDroneLightStrength
35. One BTDroneData (BTDroneID) must have one BTDroneLatitude
36. One BTDroneData (BTDroneID) must have one BTDroneAltitude
37. One BTDroneData (BTDroneID) must have one BTDroneAltitude
38. One BTDroneData (BTDroneID) must have one BTDroneLongitude
39. Each Maintenance is uniquely identified by TechnicianID
40. One Maintenance (TechnicianID) must have one DateMaintained
41. Each MaintenacePart is uniquely identified by one TechnicianID
42. One MaintenancePart (TechnicianID) must have one MaintenancePartDetails
43. Each Technician must be uniquely identified by TechnicianID
44. One Technician (TechnicianID) must have one TechnicianJobTitle
45. One Technician (TechnicianID) may have one TechnicianWorkDetails
46. Each StaffMember is uniquely identified by StaffID
47. One Staff (StaffID) may have one StaffDetails
48. One Staff (StaffID) must have one StaffFName
49. One Staff (StaffID) must have one StaffLName
50. One Staff (StaffID) must have one StaffPhoneNumber
51. One Staff (StaffID) must have one StaffEmailAddress
52. One Staff (StaffID) must have one StaffSuburb
53. One Staff (StaffID) must have one StaffCity
54. One Staff (StaffID) must have one StaffZipCode
55. One Staff (StaffID) must have one StaffStreetNumber
56. One Staff (StaffID) must have one StaffStreetName
57. One SalesPerson is uniquely identified by SalesPersonID
58. One SalesPerson (SalesPersonID) may have one ShopName
59. One SalesPerson (SalesPersonID) may have one SalesPersonDetails
60. Each BTDronePart is uniquely identified by BTDronePartID
61. One BTDronePart (BTDronePartID) must have one BTDronePartName
62. One BTDronePart (BTDronePartID) must have one BTDronePartSize
63. One BTDronePart (BTDronePartID) must have one BTDronePartWeight
64. One BTDronePart (BTDronePartID) may have one BTDronePartDetail
65. Each PartSupplier is uniquely identified by BTDronePartID, SupplierID
66. Each Supplier is uniquely identified by SupplierID
67. One Supplier (Supplier ID) must have one SupplierFName
68. One Supplier (Supplier ID) must have one SupplierLName
69. One Supplier (Supplier ID) must have one SupplierDetails
70. One Supplier (Supplier ID) must have one SupplierStreetName
71. One Supplier (Supplier ID) must have one SupplierStreetNo
72. One Supplier (Supplier ID) must have one SupplierCity
73. One Supplier (Supplier ID) must have one SupplierSuburb
74. One Supplier (Supplier ID) must have one SupplierZipCode
75. One Supplier (Supplier ID) must have one SupplierPhoneNo
76. One Supplier (Supplier ID) must have one SupplierEmail
77. One Supplier (Supplier ID) must have one SupplierCompanyName
78. One Supplier (Supplier ID) must have one SupplierFName
79. Each BTDroneConnectSales is uniquely identified by BTDroneID, SalesPersonID
80. Each RoamingZone is uniquely identified by RoamingZoneID
81. One RoamingZone (RoamingZoneID) must have RZMaxAltitude
82. One RoamingZone (RoamingZoneID) must have RZMinAltitude
83. One RoamingZone (RoamingZoneID) must have RZLatitude
84. One RoamingZone (RoamingZoneID) may have RZDetails
85. Each BTDrone is uniquely identified by BTDroneID
86. One BTDrone (BTDroneID) must have one BTDroneLocation
87. One BTDrone (BTDroneID) must have one BTDroneHumidity
88. One BTDrone (BTDroneID) must have one BTDroneLightStrength
89. One BTDrone (BTDroneID) must have one BTDroneTemperature
90. One BTDrone (BTDroneID) must have one BTDroneLongitude
91. One BTDrone (BTDroneID) must have one BTDroneLatitude
92. One BTDrone (BTDroneID) may have one BTDroneDetails
93. One BTDrone (BTDroneID) must have one BTDroneLocation
94. Each Region is uniquely identified by Region (RegionID)
95. One Region (RegionID) must have a RegionType
96. One Region (RegionID) may have a RegionDescription
97. Each Terrain is uniquely identified by TerrainID
98. One Terrain (TerrainID) must have one TerrainName
99. One Terrain (TerrainID) may have one TerrainType
100. Each AdministrativeExecutive is uniquely identified by AdminID
101. One AdministrativeExecutive (AdminID) must have one AdminSubscriptionPrice
102. One AdministrativeExecutive (AdminID) must have one AdminPermissions
103. One AdministrativeExecutive (AdminID) must have one AdminJobTitle
104. One AdministrativeExecutive (AdminID) may have one AdminJobDescription
105. Each SubscriptionSales is uniquely identified by SubscriberID, SalesPersonID
106. Each Subscription is uniquely identified by SubscriberID
107. One Subscription (SubscriberID) must have one SubscriptionType
108. One Subscription (SubscriberID) must have one SubscriptionDetails
109. One Subscription (SubscriberID) must have one SubscriptionPrice
110. Each Standard is uniquely identified by SubcriberID
111. One Standard (SubscriberID) must have one StandardSubName
112. One Standard (SubscriberID) must have one StandardFee
113. One Standard (SubscriberID) must have one StandardDetails
114. Each Gold is uniquely identified by SubcriberID
115. One Gold (SubscriberID) must have one GoldSubName
116. One Gold (SubscriberID) must have one GoldFee
117. One Gold (SubscriberID) must have one GoldDetails
118. Each Platinum is uniquely identified by SubcriberID
119. One Platinum (SubscriberID) must have one PlatinumSubName
120. One Platinum (SubscriberID) must have one PlatinumFee
121. One Platinum (SubscriberID) must have one PlatinumDetails
122. Each SuperPlatinum is uniquely identified by SubcriberID
123. One SuperPlatinum (SubscriberID) must have one SupPlatSubName
124. One SuperPlatinum (SubscriberID) must have one SupPlatFee
125. One SuperPlatinum (SubscriberID) may have one SupPlatDetails
126. One BTWebsite is uniquely identified by BTURL
127. One BTWebsite (BTURL) may have BTWebsiteDetails
128. Each VideoStream is uniquely identified by VideoID
129. One VideoStream (VideoID) must have VideoLength
130. One VideoStream (VideoID) must have VideoQuality
131. One VideoStream (VideoID) must have VideoDetails
132. One VideoStream (VideoID) must have VideoTime
133. Each BTDroneVideoStream is uniquely identified by VideoStreamID, BTDroneID

#### Relationship Sentences

1. One Account (AccountID) must have one Subscriber (SubscriberID)
2. One Subscriber (SubscriberID) must relate to one Account (AccountID)
3. One Account (AccountID) may have one or more Contractor (ContractorID)
4. One Contractor (ContractorID) must relate to or more Game Account (AccountID)
5. One Contractor (ContractorID) may have one or more BTDroneData (BTDroneID)
6. Each BTDroneData (BTDroneID) must relate to one Contractor (ContractorID)
7. One Contractor (ContractorID) must have one or more Contract (ContractID, AdminID)
8. Each Contract (ContractID, AdminID) must have one Contractor (ContractorID)
9. One AdministrativeExecutive (AdminID) may have one Contract (ContractID, AdminID)
10. Each Contract (ContractID, AdminID) must be controlled by one AdministrativeExecutive (AdminID)
11. One AdministrativeExecutive (AdminID) must have one or more Subscription (SubscriberID)
12. Each Subscription (SubscriberID) must have one AdministrativeExecutive (AdminID)
13. One AdministrativeExecutive (AdminID) must have one StaffMember (StaffID)
14. One StaffMember (StaffID) must have one AdministrativeExecutive (AdminID)
15. One StaffMember (StaffID) must have one SalesPerson (SalesPersonID)
16. One SalesPerson (SalesPersonID) must have one StaffMember (StaffID)
17. One StaffMember (StaffID) must have one Technician (TechnicianID)
18. One Technician (TechnicianID) must have one StaffMember (StaffID)
19. One Technician (TechnicianID) may have one or more Maintenance (TechnicianID)
20. Each Maintenance (TechnicianID) must have one Technician (TechnicianID)
21. One Maintenance (TechnicianID) must relate to one or more MaintenacePart (TechnicianID)
22. Each MaintenacePart (TechnicianID) may have one Maintenance (TechnicianID)
23. One Maintenance (TechnicianID) may relate to one or more BTDronePart (BTDronePartID)
24. Each BTDronePart (BTDronePartID) must have one Maintenance (TechnicianID)
25. One BTDronePart (BTDronePartID) may have one or more PartSupplier (BTDronePartID, SupplierID)
26. Each PartSupplier (BTDronePartID, SupplierID) must have one BTDronePart (BTDronePartID).
27. One Supplier (SupplierID) may have one or more PartSupplier (BTDronePartID, SupplierID)
28. Each PartSupplier (BTDronePartID, SupplierID) must have one Supplier (SupplierID)
29. One Contract (ContractID, AdminID) must have one or more BTDrone (BTDroneID)
30. Each BTDrone (BTDroneID) may have one Contract (ContractID, AdminID).
31. One BTDrone (BTDroneID) must have one or more BTDroneData (BTDroneID)
32. Each BTDroneData (BTDroneID) must have one BTDrone (BTDroneID).
33. One BTDrone (BTDroneID) must have one or more Maintenance (TechnicianID)
34. Each Maintenance (TechnicianID) must have one BTDrone (BTDroneID).
35. One BTDrone (BTDroneID) may have one or more BTDroneConnectSales (BTDroneID, SalesPersonID)
36. Each BTDroneConnectSales (BTDroneID, SalesPersonID) must have one BTDrone (BTDroneID).
37. Each BTDroneConnectSales (BTDroneID, SalesPersonID) may have one SalesPerson (SalesPersonID)
38. One SalesPerson (SalesPersonID) must relate to one or more BTDroneConnectSales (BTDroneID, SalesPersonID)
39. One BTDrone (BTDroneID) may have one or more RoamingZone (RoamingZoneID).
40. Each RoamingZone (RoamingZoneID) must relate to one BTDrone (BTDroneID)
41. Each BTDrone (BTDroneID) must have one Region (RegionID).
42. One Region (RegionID) may relate to one or more BTDrone (BTDroneID)
43. Each BTDrone (BTDroneID) must have one Terrain (TerrainID).
44. One Terrain (TerrainID) may relate to one or more BTDrone (BTDroneID)
45. One BTDrone (BTDroneID) must have one BTDroneVideoStream (VideoID, BTDroneID)
46. Each BTDroneVideoStream (VideoID, BTDroneID) may relate to one or more BTDrone (BTDroneID)
47. Each BTDrone (BTDroneID) must have one Platinum (SubscriberID)
48. One Platinum (SubscriberID) may relate to one or more BTDrone (BTDroneID)
49. Each BTDrone (BTDroneID) must have one SuperPlatinum (SubscriberID)
50. One SuperPlatinum (SubscriberID) may relate to one or more BTDrone (BTDroneID)
51. One SalesPerson (SalesPersonID) may have one or more SubscriptionSales (SubscriberID, SalesPersonID)
52. Each SubscriptionSales (SubscriberID, SalesPersonID) may have one SalesPerson (SalesPersonID)
53. Each SubscriptionSales (SubscriberID, SalesPersonID) must have one Subscription (SubscriberID)
54. One Subscription (SubscriberID) may have one or more SubscriptionSales (SubscriberID, SalesPersonID)
55. One Subscription (SubscriberID) may relate to one Standard (SubscriberID)
56. One Standard (SubscriberID) must relate to Subscription (SubscriberID)
57. One Subscription (SubscriberID) may relate to one Gold (SubscriberID)
58. One Gold (SubscriberID) must relate to Subscription (SubscriberID)
59. One Subscription (SubscriberID) may relate to one Platinum (SubscriberID)
60. One Platinum (SubscriberID) must relate to Subscription (SubscriberID)
61. One Subscription (SubscriberID) may relate to one SuperPlatinum (SubscriberID)
62. One SuperPlatinum (SubscriberID) must relate to Subscription (SubscriberID)
63. One Standard (SubscriberID) may connect to one or more VideoStream (VideoID)
64. Each VideoStream (VideoID) may relate to one Standard (SubscriberID)
65. One Gold (SubscriberID) may connect to one or more VideoStream (VideoID)
66. Each VideoStream (VideoID) may relate to one Gold (SubscriberID)
67. Each VideoStream (VideoID must be connected to one BTWebsite (BTURL)
68. One BTWebsite (BTURL) may relate to one or more VideoStream (VideoID)
69. One VideoStream (VideoID) must have one or more BTDroneVideoStream (VideoID, BTDroneID)
70. Each BTDroneVideoStream (VideoID, BTDroneID) must relate to one VideoStream (VideoID)
71. One BTWebsite (BTURL) must relate to one more Subscription (SubscriberID)
72. Each Subscription (SubscriberID) may relate to one BTWebsite (BTURL)

# Conclusion

This assignment has been very helpful as I progress my conceptual model of the project Being There database through to the logical design stage.  This process of transforming the conceptual model to logical model allowed me to understand every aspect of a project and all the new possibilities that can be presented as effectively as possible.

Now, I can say that I have a better understanding of normalization and the techniques associated with this project. After completing this assignment, I was able to understand how we can normalize a database to reduce redundancy and repetition. This will thus ensure me a better database model of a certain project with effectiveness and adequacy, related with the project’s database.

Furthermore, I had experienced working with NaLER analysis and as it helped me to identify and interpret the entity relationship models. It also helped me to be more organized with my database models and allowing me to assess the information associated with it.

Finally, through a careful analysis, I can say I have a better understanding of my both logical and conceptual model, so I believe I would be able to build an effective database model from it.

# Bibliography

Atkins, C. F., & Patrick, J. (2000). Clare Frances Atkins. *Campus-Wide Information Systems* , 85-93.

Cochrane, T. (2019). *Conceptual to Logical Model.* Nelson: NMIT.

Codd, E. (n.d.). A Relational Model of Data for Large Shared Data Banks. *Communication* , 377-387.

Elgabry, O. (2016, September 15). *Database Design*. Retrieved from Omar's ElGabry's Blog: https://medium.com/omarelgabrys-blog/database-modeling-logical-design-part-6-af029e93cc1f

Guru99. (n.d.). *What is Normalization?* Retrieved from Guru99: https://www.guru99.com/database-normalization.html

5/9/2019

By Muhammad Ashkar Yousuf Milton

NMIT

Being There. Conceptual Model, Logical Model, & Physical Model

Milestone 1, 2, and 3

Table of Contents

[1. Executive Summary of the Physical Design of Being There 4](#_Toc8324970)

[2. Description of the Physical Model 5](#_Toc8324971)

[3. Translate global logical data model for target Being There database 7](#_Toc8324972)

[3.1 Design Base Relations for DBMS 7](#_Toc8324973)

[3.2 Design Data entry in the table of Being There Database 16](#_Toc8324974)

[3.3 Design Enterprise constraints for DBMS 40](#_Toc8324975)

[4. Design Physical Representation 47](#_Toc8324976)

[4.1 Transaction Analysis with the Help of the Diagram 47](#_Toc8324977)

[4.2 Transaction A 48](#_Toc8324978)

[4.2.1 Tabular Form 48](#_Toc8324979)

[4.2.2 Description of Procedure A 49](#_Toc8324980)

[4.2.3 Procedure A 49](#_Toc8324981)

[4.3 Transaction B 51](#_Toc8324982)

[4.3.1 Tabular Form 51](#_Toc8324983)

[4.3.1 Description of Procedure B 52](#_Toc8324984)

[4.3.2 Procedure B 52](#_Toc8324985)

[4.4 Transaction C 54](#_Toc8324986)

[4.4.1 Tabular Form 54](#_Toc8324987)

[4.4.2 Description of Procedure C 54](#_Toc8324988)

[4.4.3 Procedure C 55](#_Toc8324989)

[4.5 Transaction D 56](#_Toc8324990)

[4.5.1 Tabular Form 56](#_Toc8324991)

[4.5.2 Description of Procedure D 56](#_Toc8324992)

[4.5.3 Procedure D 57](#_Toc8324993)

[4.6 Transaction E 57](#_Toc8324994)

[4.6.1 Tabular Form 57](#_Toc8324995)

[4.6.2 Description of Procedure E 58](#_Toc8324996)

[4.6.3 Procedure E 58](#_Toc8324997)

[4.7 Transaction F 59](#_Toc8324998)

[4.7.1 Tabular Form 59](#_Toc8324999)

[4.7.2 Description of Procedure F 59](#_Toc8325000)

[4.7.3 Procedure F 60](#_Toc8325001)

[4.8 Transaction G 61](#_Toc8325002)

[4.8.1 Tabular Form 61](#_Toc8325003)

[4.8.2 Description of Procedure G 61](#_Toc8325004)

[4.8.3 Procedure G 61](#_Toc8325005)

[4.9 Transaction H 63](#_Toc8325006)

[4.9.1 Tabular Form 63](#_Toc8325007)

[4.9.2 Description of Procedure H 63](#_Toc8325008)

[4.9.3 Procedure H 63](#_Toc8325009)

[4.10 Transaction I 65](#_Toc8325010)

[4.10.1 Tabular Form 65](#_Toc8325011)

[4.10.2 Description of Procedure I 65](#_Toc8325012)

[4.10.3 Procedure I 66](#_Toc8325013)

[4.11 Choose file organizations 66](#_Toc8325014)

[4.12 Choose Secondary Indexes 69](#_Toc8325015)

[4.13 Consider the Introduction of Controlled Redundancy 70](#_Toc8325016)

[4.14 Estimated Disk Space requirements 0](#_Toc8325017)

[4.14.1 Each Tables of Being There Database 0](#_Toc8325018)

[4.14.2 Total Being There Database Size 0](#_Toc8325019)

[4.15 Design Security Mechanisms 0](#_Toc8325020)

[4.16 Monitor and Tune the Operational System 0](#_Toc8325021)

[5. Physical Design for Being There database 0](#_Toc8325022)

[6. Issues and Resolutions 0](#_Toc8325023)

[7. Conclusion from Physical Design 6](#_Toc8325024)

[8. Bibliography 7](#_Toc8325025)

# Executive Summary of the Physical Design of Being There

In the last phase we will be working on the Physical design of Being There. To have a complete database, we need to make sure that we have a complete Physical Design of Being there. For the last phase

The last phase was to complete the physical design for the database before building it. The process used has been laid out in the last section of this report which includes the calculations to determine the maximum size of the completed database. A transaction analysis for the queries involved in completing a sales agreement between Invoice Funders and a client has also been included. The last element of this report is an accounting of all the issues that I discovered during the undertaking of this project and the steps I have taken to overcome them.

In the phase of Physical model, we will be learning how to create tables inn SQL server and then how to put insert data in the tables, using various number of queries. Through this model, we would be able to learn how to create transactional analysis for different procedures. We would be able to learn how to work on transactional analysis in both design and tabular form.

Later, in the phase of Physical model we would learn about choosing file organizations , choosing Secondary Indexes from our Being There database, considering the Introduction of Controlled Redundancy to avoid repeated data, estimating the disk space for Being There Database, learning about design security mechanisms, and finally we will learn about monitoring and tuning the operational system.

# Description of the Physical Model

Physical model of a database is quite different from the Logical model of the database, as the purpose of the physical model is to represent the blueprint of the relational database. The physical model represents how a data can be structured and they can be related to specific Database Management System, so we need to consider the convention and the restriction associated with Database Management System when we are implementing our data on the physical ERD design (Paradigm, 2018).

The physical design is the place where we translate the actual schema into the actual database structure. For that we must use the concept of the logical design that we have used in Milestone 2. Thus, we must map certain aspects, which include:

* 1. Entities to table
  2. Relationships to foreign keys.
  3. Different attributes to different columns
  4. Primary Unique Identifiers to the Primary Key.
  5. Unique identifier to Unique Keys (Oracle, 1999).

In physical model if we have many too many to relationships, we are going to use the join table to connect two tables. For us to do a complete physical database design, we must understand how frequently and how efficiently we are going to use the data. To begin with, if we want a high-performance physical database system, we must gather and analysis the data in such a way that the user can access the data. Furthermore, we also need to use certain tools if we want full access of the data. For Being There database, we are going to use SQL Server (Tupper, 2011).

The physical design of the database is also going to identify the physical configuration of the database used and the storage it requires to run the database. These includes certain specification which include: data elements, data character, data types, data indexing, and other specification in the Database Management System’s data dictionary. Additionally, in the physical model of the database we are also going to include all the elements that are required to create relationships between the tables. We are also going to include indexes, joining tables, constraints with definitions, or clusters. We can also use the physical model to calculate the storage capacity of the complete Being There database model.

# Translate global logical data model for target Being There database

## Design Base Relations for DBMS

Use BeingThere

GO

---1) Table for Staff Member---

CREATE TABLE StaffMember (

StaffID INT UNIQUE NOT NULL,

StaffJobTitle VARCHAR (50) NOT NULL,

StaffDetails VARCHAR (150) NULL,

StaffFName VARCHAR (100) NOT NULL,

StaffLName VARCHAR (100) NOT NULL,

StaffPhoneNumber VARCHAR (50) NOT NULL,

StaffEmailAddress VARCHAR (100) NOT NULL,

StaffSuburb VARCHAR (50) NOT NULL,

StaffCity VARCHAR (50) NOT NULL,

StaffZipCode VARCHAR (20) NOT NULL,

StaffStreetName VARCHAR (50) NOT NULL,

StaffStreetAddress VARCHAR (100) NOT NULL,

PRIMARY KEY (StaffID)

);

---2) Table for Account---

CREATE TABLE Account (

AccountID INT UNIQUE NOT NULL,

AccountFName VARCHAR (100) NOT NULL,

AccountLName VARCHAR (100) NOT NULL,

AccountDetails VARCHAR (250) NULL,

AccountStreetName VARCHAR (50) NOT NULL,

AccountStreetAddress VARCHAR (100) NOT NULL,

AccountCity VARCHAR (50) NOT NULL,

AccountSuburb VARCHAR (50) NOT NULL,

AccountZipCode VARCHAR (20) NOT NULL,

AccountEmail VARCHAR (100) NOT NULL,

AccountPhone VARCHAR (50) NOT NULL,

PRIMARY KEY (AccountID)

);

---3) Table for Subscriber---

CREATE TABLE Subscriber (

SubscriberID INT UNIQUE NOT NULL,

SubscriberPaymentDetails VARCHAR (150) NULL,

SubscriberPassword VARCHAR (50) NOT NULL,

SubscriberDiscount DECIMAl (8,3) NOT NULL,

PRIMARY KEY (SubscriberID),

FOREIGN KEY (SubscriberID) REFERENCES Account (AccountID)

);

---4) Table for Contractor---

CREATE TABLE Contractor (

ContractorID INT UNIQUE NOT NULL,

ContractorOrganizationName VARCHAR (250) NOT NULL,

PRIMARY KEY (ContractorID),

FOREIGN KEY (ContractorID) REFERENCES Account (AccountID)

);

---5) Table for Administrative Executive---

CREATE TABLE AdministrativeExecutive (

AdminID INT UNIQUE NOT NULL,

AdminSubscriptionPrice VARCHAR (50) NOT NULL,

AdminPermissions VARCHAR (20) NOT NULL,

AdminJobTitle VARCHAR (50) NOT NULL,

AdminJobDescription VARCHAR (250) NULL,

PRIMARY KEY (AdminID),

FOREIGN KEY (AdminID) REFERENCES StaffMember (StaffID) ON DELETE CASCADE

);

---6) Table for SalesPerson---

CREATE TABLE SalesPerson (

SalesPersonID INT UNIQUE NOT NULL,

ShopName VARCHAR (50) NOT NULL,

SalesPersonShopDetails VARCHAR (250) NOT NULL,

PRIMARY KEY (SalesPersonID),

FOREIGN KEY (SalesPersonID) REFERENCES StaffMember (StaffID) ON DELETE CASCADE

);

---7) Table for Technician---

CREATE TABLE Technician (

TechnicianID INT UNIQUE NOT NULL,

TechnicianJobTitle VARCHAR (50) NOT NULL,

TechnicianWorkDetails VARCHAR (250) NULL,

PRIMARY KEY (TechnicianID),

FOREIGN KEY (TechnicianID) REFERENCES StaffMember (StaffID) ON DELETE CASCADE

);

---8) Table for Contract---

CREATE TABLE Contract (

ContractID INT UNIQUE NOT NULL,

ContractType VARCHAR (50) NOT NULL,

ContractDetails VARCHAR (250) NOT NULL,

ContractStartDate DATE NOT NULL,

ContractEndDate DATE NOT NULL,

AdminID INT NOT NULL,

ContractorID INT NOT NULL,

PRIMARY KEY (ContractID),

FOREIGN KEY (AdminID) REFERENCES AdministrativeExecutive (AdminID) ON DELETE NO ACTION,

FOREIGN KEY (ContractorID) REFERENCES Contractor (ContractorID) ON DELETE NO ACTION

);

---9) Table for Region---

CREATE TABLE Region (

RegionID INT UNIQUE NOT NULL,

RegionType VARCHAR (50) NOT NULL,

RegionDescription VARCHAR (250) NOT NULL,

PRIMARY KEY (RegionID),

);

---10) Table for Terrain--

CREATE TABLE Terrain (

TerrainID INT UNIQUE NOT NULL,

TerrainName VARCHAR (50) NOT NULL,

TerrainType VARCHAR (50) NOT NULL,

PRIMARY KEY (TerrainID)

);

---11) Table for BTDrone----

CREATE TABLE BTDrone (

BTDroneID INT UNIQUE NOT NULL,

BTDroneLocation VARCHAR(50) NOT NULL,

BTDroneHumidity Decimal(8,3) NOT NULL,

BTDroneTemperature Decimal (8,3) NOT NULL,

BTDroneLightStrength Decimal (8,3) NOT NULL,

BTDroneLatitude Decimal (8,3) NOT NULL,

BTDroneLongitude Decimal (8,3) NOT NULL,

BTDroneAltitude Decimal (8,3) NOT NULL,

BTDroneDetails Varchar (250) NULL,

RegionID INT NULL,

PRIMARY KEY (BTDroneID),

FOREIGN KEY (RegionID) REFERENCES Region (RegionID)

);

---12) Table for RoamingZone---

CREATE TABLE RoamingZone (

RoamingZoneID INT UNIQUE NOT NULL,

RZDetails VARCHAR (250) NULL,

BTDroneID INT NOT NULL,

PRIMARY KEY (RoamingZoneID),

FOREIGN KEY (BTDroneID) REFERENCES BTDrone (BTDroneID)

);

---13) Table for Altitude---

CREATE TABLE Altitude (

AltitudeID Integer UNIQUE NOT NULL,

MaxAltitude DECIMAL (8,3) NOT NULL,

MinAltitude DECIMAL (8,3) NOT NULL,

RoamingZoneID INT NOT NULL,

PRIMARY KEY (AltitudeID),

FOREIGN KEY (RoamingZoneID) REFERENCES RoamingZone (RoamingZoneID) ON DELETE NO ACTION

);

---14) Table for ZoneBoundary---

CREATE TABLE ZoneBoundary (

ZoneBoundaryID INT UNIQUE NOT NULL,

LatitudeCalculated VARCHAR (50) NOT NULL,

LongitudeCalculated VARCHAR (50) NOT NULL,

RoamingZoneID INT NOT NULL,

PRIMARY KEY (ZoneBoundaryID),

FOREIGN KEY (RoamingZoneID) REFERENCES RoamingZone (RoamingZoneID) ON DELETE NO ACTION

);

---15) Table for Supplier---

CREATE TABLE Supplier (

SupplierID INT UNIQUE NOT NULL,

SupplierFName VARCHAR(100) NOT NULL,

SupplierLName VARCHAR(100) NOT NULL,

SupplierDetails VARCHAR(150) NOT NULL,

SupplierStreetName VARCHAR(50) NOT NULL,

SupplierSteetAddress VARCHAR(150) NOT NULL,

SupplierCity VARCHAR(50) NOT NULL,

SupplierSuburb VARCHAR(50) NOT NULL,

SupplierZipCode VARCHAR(30) NOT NULL,

SupplierPhoneNo VARCHAR(30) NOT NULL,

SupplierEmail VARCHAR(100) NOT NULL,

SupplierCompanyName VARCHAR(100) NOT NULL,

PRIMARY KEY (SupplierID)

);

---16) Table for Maintenace---

CREATE TABLE Maintenance (

DateMaintained DATE NOT NULL,

TechnicianID INT NOT NULL,

BTDroneID INT NOT NULL,

PRIMARY KEY (DateMaintained),

FOREIGN KEY (TechnicianID) REFERENCES Technician (TechnicianID) ON DELETE NO ACTION,

FOREIGN KEY (BTDroneID) REFERENCES BTDrone (BTDroneID) ON DELETE NO ACTION

);

---17) Table for Maintenance Part---

CREATE TABLE MaintenancePart (

DateMaintained DATE NOT NULL,

MaintenancePartDetails VARCHAR (250) NOT NULL,

PRIMARY KEY (DateMaintained),

FOREIGN KEY (DateMaintained) REFERENCES Maintenance (DateMaintained)

);

---18) Table for BTDronePart---

CREATE TABLE BTDronePart (

BTDronePartID INT UNIQUE NOT NULL,

BTDronePartName VARCHAR (150) NOT NULL,

BTDronePartPartSIze DECIMAL (8,3) NOT NULL,

BTDronePartWeight DECIMAL (8,3) NOT NULL,

BTDronePartDetail VARCHAR (250) NULL,

DateMaintained DATE NULL,

PRIMARY KEY (BTDronePartID),

FOREIGN KEY (DateMaintained) REFERENCES Maintenance (DateMaintained)

);

---19) Table for BTWebsite--

CREATE TABLE BTWebsite (

BTSalesID INT UNIQUE NOT NULL,

BTSalesDetails VARCHAR (250) NOT NULL,

PRIMARY KEY (BTSalesID)

);

---20) Table for Subscription---

CREATE TABLE Subscription (

SubscriberID INT UNIQUE NOT NULL,

SubscriptionType VARCHAR (50) NOT NULL,

SubscriptionDetails VARCHAR (250) NULL,

SubscriptionPrice VARCHAR (50) NOT NULL,

SubscriptionStartDate DATE NOT NULL,

SubscriptionEndDate DATE NOT NULL,

BTSalesID INT NULL,

PRIMARY KEY (SubscriberID),

FOREIGN KEY (SubscriberID) REFERENCES Subscriber (SubscriberID) ON DELETE NO ACTION,

FOREIGN KEY (BTSalesID) REFERENCES BTWebsite (BTSalesID)

);

---21) Table for Standard---

CREATE TABLE Standard (

StandardSubscriberID INT NOT NULL,

PRIMARY KEY (StandardSubscriberID),

FOREIGN KEY (StandardSubscriberID) REFERENCES Subscriber (SubscriberID)

);

---22) Table for Gold---

CREATE TABLE Gold (

GoldSubscriberID INT NOT NULL,

PRIMARY KEY (GoldSubscriberID),

FOREIGN KEY (GoldSubscriberID) REFERENCES Subscriber (SubscriberID)

);

---23) Table for Platinum---

CREATE TABLE Platinum (

PlatinumSubscriberID INT NOT NULL,

BTDroneID INT NOT NULL,

PRIMARY KEY (PlatinumSubscriberID),

FOREIGN KEY (PlatinumSubscriberID) REFERENCES Subscriber(SubscriberID),

FOREIGN KEY (BTDroneID) REFERENCES BTDrone(BTDroneID),

);

---24) Table for Super Platinum---

CREATE TABLE SuperPlatinum (

SuperPlatinumSubscriberID INT NOT NULL,

ContractID INT NOT NULL,

BTDroneID INT NOT NULL,

PRIMARY KEY (SuperPlatinumSubscriberID),

FOREIGN KEY (SuperPlatinumSubscriberID) REFERENCES Subscriber (SubscriberID),

FOREIGN KEY (ContractID) REFERENCES Contract (ContractID),

FOREIGN KEY (BTDroneID) REFERENCES BTDrone(BTDroneID),

);

---25) Table for Video Stream---

CREATE TABLE VideoStream (

VideoID INT NOT NULL,

VideoLength VARCHAR (30) NOT NULL,

VideoQuality VARCHAR (30) NOT NULL,

VideoDetails VARCHAR (250) NULL,

VideoTime DATETIME NOT NULL,

BTSalesID INT NOT NULL,

StandardSubscriberID INT NOT NULL,

GoldSubscriberID INT NOT NULL,

PRIMARY KEY (VideoID),

FOREIGN KEY (BTSalesID) REFERENCES BTWebsite (BTSalesID),

FOREIGN KEY (StandardSubscriberID) REFERENCES Subscriber (SubscriberID),

FOREIGN KEY (GoldSubscriberID) REFERENCES Subscriber (SubscriberID)

);

---26) Table for BTDrone Data---

CREATE TABLE BTDroneData (

BTDataID INT UNIQUE NOT NULL,

BTDroneLocation VARCHAR(50) NOT NULL,

BTDroneHumidity Decimal(8,3) NOT NULL,

BTDroneTemperature Decimal (8,3) NOT NULL,

BTDroneLightStrength Decimal (8,3) NOT NULL,

BTDroneLatitude Decimal (8,3) NOT NULL,

BTDroneLongitude Decimal (8,3) NOT NULL,

BTDroneAltitude Decimal (8,3) NOT NULL,

ContractorID INT NULL,

BTDroneID INT NULL,

PRIMARY KEY (BTDataID),

FOREIGN KEY (BTDroneID) REFERENCES BTDrone (BTDroneID),

FOREIGN KEY (ContractorID) REFERENCES Contractor (ContractorID)

);

----27) Tables for BTDroneTerrain ----

CREATE TABLE BTDroneTerrain (

TerrainID INT NOT NULL,

BTDroneID INT NOT NULL,

PRIMARY KEY (TerrainID, BTDroneID),

FOREIGN KEY (TerrainID) REFERENCES Terrain (TerrainID) ON DELETE CASCADE,

FOREIGN KEY (BTDroneID) REFERENCES BTDrone (BTDroneID) ON DELETE CASCADE

);

---28) Table for BTDroneSubscription----

CREATE TABLE BTDroneSubscription (

BTDroneID INT NOT NULL,

SubscriberID INT NOT NULL,

PRIMARY KEY (BTDroneID, SubscriberID),

FOREIGN KEY (BTDroneID) REFERENCES BTDrone (BTDroneID) ON DELETE CASCADE,

FOREIGN KEY (SubscriberID) REFERENCES Subscriber (SubscriberID) ON DELETE CASCADE

);

---29) Table for Subscription Sales---

CREATE TABLE SubscriptionSales (

SubscriberID INT NOT NULL,

SalesPersonID INT NOT NULL,

PRIMARY KEY (SubscriberID, SalesPersonID),

FOREIGN KEY (SubscriberID) REFERENCES Subscriber (SubscriberID) ON DELETE CASCADE,

FOREIGN KEY (SalesPersonID) REFERENCES SalesPerson (SalesPersonID) ON DELETE CASCADE

);

---30) Table for BT Drone Video Stream---

CREATE TABLE BTDroneVideoStream (

VideoID INT NOT NULL,

BTDroneID INT NOT NULL,

PRIMARY KEY ( VideoID, BTDroneID),

FOREIGN KEY (VideoID) REFERENCES VideoStream (VideoID) ON DELETE CASCADE,

FOREIGN KEY (BTDroneID) REFERENCES BTDrone (BTDroneID) ON DELETE CASCADE

);

---31) Table for Part Supplier---

CREATE TABLE PartSupplier (

BTDronePartID INT NOT NULL,

SupplierID INT NOT NULL,

PRIMARY KEY (BTDronePartID, SupplierID),

FOREIGN KEY (BTDronePartID) REFERENCES BTDronePart (BTDronePartID) ON DELETE CASCADE,

FOREIGN KEY (SupplierID) REFERENCES Supplier (SupplierID) ON DELETE CASCADE

);

---32) Table for BT Drone Contract---

CREATE TABLE BTDroneContract (

ContractID INT NOT NULL,

BTDroneID INT NOT NULL,

PRIMARY KEY (ContractID, BTDroneID),

FOREIGN KEY (ContractID) REFERENCES Contract (ContractID) ON DELETE CASCADE,

FOREIGN KEY (BTDroneID) REFERENCES BTDrone (BTDroneID) ON DELETE CASCADE

);

---33) Table for Contract BT Drone Data---

CREATE TABLE ContractData (

ContractID INT NOT NULL,

BTDataID INT NOT NULL,

PRIMARY KEY (ContractID, BTDataID),

FOREIGN KEY (ContractID) REFERENCES Contract (ContractID) ON DELETE CASCADE,

FOREIGN KEY (BTDataID) REFERENCES BTDroneData (BTDataID) ON DELETE CASCADE

);

---34) Table for Contract Roaming Zone----

CREATE TABLE ContractRoamingZone (

ContractID INT NOT NULL,

RoamingZoneID INT NOT NULL,

PRIMARY KEY (ContractID, RoamingZoneID),

FOREIGN KEY (ContractID) REFERENCES Contract (ContractID) ON DELETE CASCADE,

FOREIGN KEY (RoamingZoneID) REFERENCES RoamingZone (RoamingZoneID) ON DELETE CASCADE

);

---35) Table for Subscription Admin---

CREATE TABLE SubscriptionAdmin (

SubscriberID INT NOT NULL,

AdminID INT NOT NULL,

PRIMARY KEY (SubscriberID,AdminID),

FOREIGN KEY (SubscriberID) REFERENCES Subscriber (SubscriberID) ON DELETE CASCADE,

FOREIGN KEY (AdminID) REFERENCES AdministrativeExecutive (AdminID) ON DELETE CASCADE

);

---36) Table for VideoSubscription---

CREATE TABLE VideoSubscription (

VideoID INT NOT NULL,

SubscriberID INT NOT NULL,

PRIMARY KEY (VideoID,SubscriberID),

FOREIGN KEY (VideoID) REFERENCES VideoStream (VideoID) ON DELETE CASCADE,

FOREIGN KEY (SubscriberID) REFERENCES Subscriber (SubscriberID) ON DELETE CASCADE

);

## Design Data entry in the table of Being There Database

Use BeingThere

GO

----1) Insert Into Staff Member (15 Staffs)---

INSERT INTO StaffMember([StaffID],[StaffJobTitle],[StaffDetails],[StaffFName],[StaffLName],[StaffPhoneNumber],[StaffEmailAddress],[StaffSuburb],[StaffCity],[StaffZipCode],[StaffStreetAddress],[StaffStreetName]) VALUES

(101,'Administrative Executive','Administrative executives provide high-level clerical support to an entire organization','Claudia','Duffy','(08) 5901 0171','non.enim@Aliquamnisl.co.uk','Cartago','Carmen','45162','344-9727 Et Street','22 Edien'),

(102,'Salesperson,','Sales representatives sell retail products, goods and services to customers.','Farrah','Gutierrez','(03) 2497 6669','libero@feugiattellus.com','RS','Rio Grande','84-708','744 In Rd.','7 Koina'),

(103,'Technician,','A person employed to look after technical equipment or do practical work in a laboratory.','Rhonda','Dale','(02) 9820 3142','nec.enim@diam.co.uk','Henegouwen','Labuissire','41679','493-8619 Est, Road','16 Norse'),

(104,'Technician','A person employed to look after technical equipment or do practical work in a laboratory.','Imani','Koch','(04) 0534 6981','nec.imperdiet.nec@Pellentesquehabitantmorbi.ca','Maryland','Columbia','609600','Ap #311-4039 Odio Road','36 Odio'),

(105,'Administrative Executive','Administrative executives provide high-level clerical support to an entire organization','Francis','Haynes','(08) 7261 9999','sapien.molestie.orci@molestie.edu','Maule','Maule','40306','767-1911 Tincidunt St.','54 Tincidunt'),

(106,'Salesperson','Sales representatives sell retail products, goods and services to customers.','Buffy','Vazquez','(04) 1162 0882','Curabitur.massa@nunc.ca','AQ','Anglet','9790','Ap #196-4596 Ut Av.','49 Utamaew'),

(107,'Technician','A person employed to look after technical equipment or do practical work in a laboratory.','Patrick','Luna','(02) 6005 7064','dolor.elit@ultricessitamet.edu','NE','Omaha','9478 IM','8805 Sem Ave','35 Semwiner'),

(108,'Salesperson','Sales representatives sell retail products, goods and services to customers.','Sylvester','Garza','(03) 3717 3314','ipsum.dolor.sit@massalobortisultrices.ca','Podlaskie','Suwałki','60993','Ap #813-2610 Hendrerit St.','7 Hendrerit'),

(109,'Technician','A person employed to look after technical equipment or do practical work in a laboratory.','Louis','Simmons','(05) 9149 7533','nunc.In.at@IncondimentumDonec.com','NI','Whangarei','88251','926 Dolor. Ave','60 Dolor'),

(110,'Salesperson','Sales representatives sell retail products, goods and services to customers.','Darryl','Casey','(03) 8456 9825','dolor@scelerisque.edu','Oost-Vlaanderen','Lede','00789-014','4804 Pellentesque. Av.','26 Pellentesque'),

(111,'Administrative Executive','Administrative executives provide high-level clerical support to an entire organization','Moudia','Pluffy','(08) 5901 0171','non.enim@Aliquamnisl.co.uk','Cartago','Carmen','45162','344-9727 Et Street','22 Edien'),

(112,'Administrative Executive','Administrative executives provide high-level clerical support to an entire organization','Cdiduu','Morse','(08) 5901 0171','non.enim@Aliquamnisl.co.uk','Cartago','Carmen','45162','344-9727 Et Street','22 Edien'),

(113,'Administrative Executive','Administrative executives provide high-level clerical support to an entire organization','Claudia','Sgwan','(08) 5901 0171','non.enim@Aliquamnisl.co.uk','Cartago','Carmen','45162','344-9727 Et Street','22 Edien'),

(114,'Technician','A person employed to look after technical equipment or do practical work in a laboratory.','Pouis','Plousa','(05) 9149 7533','nunc.In.at@IncondimentumDonec.com','NI','Whangarei','88251','926 Dolor. Ave','60 Dolor'),

(115,'Salesperson','Sales representatives sell retail products, goods and services to customers.','Maryl','Jesssey','(03) 8456 9825','dolor@scelerisque.edu','Oost-Vlaanderen','Lede','00789-014','4804 Pellentesque. Av.','26 Pellentesque');

---2) Insert table Acccount (40 Accounts)---

INSERT INTO Account([AccountID],[AccountFName],[AccountLName],[AccountDetails],[AccountStreetAddress],[AccountStreetName],[AccountZipCode],[AccountSuburb],[AccountCity],[AccountEmail],[AccountPhone]) VALUES

(10901,'Jack','Knapp','non enim commodo','9619 Erat Road','T7H',397,'Levin','NI','Vestibulum.ut.eros@quisturpis.co.uk','(01) 707 341 8774'),

(10902,'Moses','Henderson','non enim','3027 Nec, Street','Y5H',357,'Morrinsville','NI','mattis@ante.org','(08) 716 548 3265'),

(10903,'Kennedy','Nguyen','luctus, ipsum leo','4576 Erat. Av.','P4H',139,'Tokoroa','North Island','eget.ipsum.Suspendisse@Quisqueimperdiet.com','(08) 436 574 5769'),

(10904,'Idona','Dixon','eleifend non,','P.O. Box 957, 6897 Magna. Rd.','A0O',123,'Rotorua','NI','ultrices.a.auctor@nibh.com','(03) 306 569 4307'),

(10905,'Ashton','Mays','amet ornare lectus','9601 Risus. Ave','N0F',304,'Lower Hutt','NI','risus@nonmagnaNam.ca','(09) 631 404 0958'),

(10906,'Blaze','Morse','Donec','760-3483 Accumsan St.','C5X',315,'Waitara','NI','sem@metusAenean.ca','(02) 781 916 3283'),

(10907,'Benedict','Trevino','dui nec','5833 Nam Road','O2J',31,'Matamata','NI','et.netus@tellusimperdietnon.org','(04) 418 828 8889'),

(10908,'Stephanie','Burris','mi tempor','9744 A Rd.','C7O',15,'Wanganui','NI','sagittis@pulvinar.com','(02) 320 319 1953'),

(10909,'Valentine','Bryant','fringilla purus mauris','256-7351 Vehicula St.','D7T',257,'Gisborne','NI','Morbi.sit.amet@egestasDuis.org','(03) 208 492 0521'),

(10910,'Calvin','Mcknight','auctor','154-1533 Lorem St.','G7J',18,'Kawerau','North Island','enim.commodo@risus.co.uk','(03) 922 858 5779'),

(10911,'Kyra','Bradley','neque. Nullam','807-8187 Posuere Rd.','N1E',149,'Wanganui','NI','sollicitudin@ornare.ca','(07) 760 996 5497'),

(10912,'Nero','Stanley','Nunc','P.O. Box 991, 7425 Arcu Street','D9C',23,'Gore','SI','et@montesnascetur.edu','(06) 024 158 3436'),

(10913,'Chantale','Rivera','sem,','Ap #874-8144 Accumsan Av.','C1J',291,'Waitara','NI','pharetra.Quisque.ac@nec.ca','(06) 545 218 3543'),

(10914,'Todd','Mcleod','per conubia','P.O. Box 786, 4104 Imperdiet Av.','J0A',128,'Invercargill','SI','a@montes.net','(01) 532 360 6344'),

(10915,'Lee','Cotton','feugiat','P.O. Box 637, 2625 Sed Rd.','H8F',407,'Balclutha','SI','vehicula.risus.Nulla@Donec.com','(02) 663 483 7839'),

(10916,'Ignatius','Sharpe','Cum sociis natoque','P.O. Box 528, 8642 Primis St.','X5K',358,'Hamilton','NI','Cras.eget@malesuadaaugueut.net','(08) 439 485 4563'),

(10917,'Alec','Wells','sit','3157 Nec, Ave','B2W',25,'Kawerau','NI','orci.sem.eget@nonlorem.net','(06) 353 114 4411'),

(10918,'Paul','Lopez','eu neque pellentesque','469-7499 Aliquam Ave','R9C',151,'Manukau','NI','In@aliquetsem.co.uk','(05) 446 388 7615'),

(10919,'Austin','Pickett','habitant morbi','P.O. Box 123, 5262 Vivamus Rd.','E5D',300,'Matamata','NI','lobortis@luctus.edu','(08) 666 139 7503'),

(10920,'Nyssa','Knowles','magna. Sed eu','269-2555 Nullam St.','Z4Z',331,'Tauranga','North Island','Etiam.bibendum.fermentum@scelerisquemollis.org','(04) 439 763 9048'),

(10921,'Ryan','Nnapp','non enim commodo','9619 Erat Road','T7H',397,'Levin','NI','Vestibulum.ut.eros@quisturpis.co.uk','(01) 707 341 8774'),

(10922,'Loses','Ienderson','Pellentesque habitant morbi','3027 Nec, Street','Y5H',357,'Morrinsville','NI','mattis@ante.org','(08) 716 548 3265'),

(10923,'Mennedy','Pguyen','luctus, ipsum leo','4576 Erat. Av.','P4H',139,'Tokoroa','North Island','eget.ipsum.Suspendisse@Quisqueimperdiet.com','(08) 436 574 5769'),

(10924,'Udona','Lixon','eleifend non,','P.O. Box 957, 6897 Magna. Rd.','A0O',123,'Rotorua','NI','ultrices.a.auctor@nibh.com','(03) 306 569 4307'),

(10925,'Jashton','Lays','amet ornare lectus','9601 Risus. Ave','N0F',304,'Lower Hutt','NI','risus@nonmagnaNam.ca','(09) 631 404 0958'),

(10926,'Drlaze','Lorse','Donec','760-3483 Accumsan St.','C5X',315,'Waitara','NI','sem@metusAenean.ca','(02) 781 916 3283'),

(10927,'Cenedict','Urevino','dui nec','5833 Nam Road','O2J',31,'Matamata','NI','et.netus@tellusimperdietnon.org','(04) 418 828 8889'),

(10928,'Pstephanie','Ourris','mi tempor','9744 A Rd.','C7O',15,'Wanganui','NI','sagittis@pulvinar.com','(02) 320 319 1953'),

(10929,'Yalentine','Rryant','fringilla purus mauris','256-7351 Vehicula St.','D7T',257,'Gisborne','NI','Morbi.sit.amet@egestasDuis.org','(03) 208 492 0521'),

(10930,'Kalvin','Pcknight','auctor','154-1533 Lorem St.','G7J',18,'Kawerau','North Island','enim.commodo@risus.co.uk','(03) 922 858 5779'),

(10931,'Jyra','Oradley','neque. Nullam','807-8187 Posuere Rd.','N1E',149,'Wanganui','NI','sollicitudin@ornare.ca','(07) 760 996 5497'),

(10932,'Tero','Ptanley','Nunc','P.O. Box 991, 7425 Arcu Street','D9C',23,'Gore','SI','et@montesnascetur.edu','(02) 024 158 3436'),

(10933,'Qhantale','Livera','sem,','Ap #874-8144 Accumsan Av.','C1J',291,'Waitara','NI','pharetra.Quisque.ac@nec.ca','(02) 545 218 3543'),

(10934,'Lodd','Tcleod','per conubia','P.O. Box 786, 4104 Imperdiet Av.','J0A',128,'Invercargill','SI','a@montes.net','(02) 532 360 6344'),

(10935,'Bsee','Lotton','feugiat','P.O. Box 637, 2625 Sed Rd.','H8F',407,'Balclutha','SI','vehicula.risus.Nulla@Donec.com','(02) 663 483 7839'),

(10936,'OLignatius','Uiharpe','Cum sociis natoque','P.O. Box 528, 8642 Primis St.','X5K',358,'Hamilton','NI','Cras.eget@malesuadaaugueut.net','(02) 439 485 4563'),

(10937,'Mlec','Pells','sit','3157 Nec, Ave','B2W',25,'Kawerau','NI','orci.sem.eget@nonlorem.net','(06) 353 114 4411'),

(10938,'Loaul','Mopez','eu neque pellentesque','469-7499 Aliquam Ave','R9C',151,'Manukau','NI','In@aliquetsem.co.uk','(02) 446 388 7615'),

(10939,'Ollsustin','Uuickett','habitant morbi','P.O. Box 123, 5262 Vivamus Rd.','E5D',300,'Matamata','NI','lobortis@luctus.edu','(02) 666 139 7503'),

(10940,'Myssa','Kpowles','magna. Sed eu','269-2555 Nullam St.','Z4Z',331,'Tauranga','North Island','Etiam.bibendum.fermentum@scelerisquemollis.org','(02) 539 763 9048');

---3) Insert Into the Subsriber (20 Subscriber listed)---

INSERT INTO Subscriber([SubscriberID],[SubscriberPaymentDetails],[SubscriberPassword],[SubscriberDiscount]) VALUES

(10921,'Credit Card','POZ35HMX7KK','1.376'),

(10922,'Debit Card ','KLC93JUD2PI','2.226'),

(10923,'Paypal','LQU18MNO3YO','1.098'),

(10924,'Credit Card','PRC43MLZ2UA','2.341'),

(10925,'PayPal','SHI85RHC7YK','2.767'),

(10926,'BitCoins','ITR74HAF4OP','1.465'),

(10927,'EFTPOS','JLC13SED3BR','1.154'),

(10928,'Debit Card','BBE06OQP5YE','2.418'),

(10929,'PayPal','YKI28GIH5BH','3.000'),

(10930,'EFTPOS','JMC93UMD7LW','1.239'),

(10931,'Credit Card','POZ35HMX7KK','0.476'),

(10932,'Debit Card ','KLC93JUD2PI','0.726'),

(10933,'Paypal','LQU18MNO3YO','0.598'),

(10934,'Credit Card','PRC43MLZ2UA','1.651'),

(10935,'PayPal','SHI85RHC7YK','2.527'),

(10936,'BitCoins','ITR74HAF4OP','1.865'),

(10937,'EFTPOS','JLC13SED3BR','3.000'),

(10938,'Debit Card','BBE06OQP5YE','2.818'),

(10939,'PayPal','YKI28GIH5BH','1.913'),

(10940,'EFTPOS','JMC93UMD7LW','2.839');

----4) Insert Into the Contractor (20 Contractors)---

INSERT INTO Contractor([ContractorID],[ContractorOrganizationName]) VALUES

(10901,'Tempor Company'),

(10902,'Montes Nascetur LLC'),

(10903,'Class Industries'),

(10904,'Duis Incorporated'),

(10905,'Consectetuer Associates'),

(10906,'Eu Industries'),

(10907,'Dis PC'),

(10908,'Eleifend Vitae Ltd'),

(10909,'Cursus Foundation'),

(10910,'Tellus Foundation'),

(10911,'Rempor Lopas Company'),

(10912,'Yontes LLC Limited'),

(10913,'Pobortis Class Industries'),

(10914,'Kioduis Incorporated'),

(10915,'Ior Associates'),

(10916,'Milsallamcorper Eu Industries'),

(10917,'OP Sis PC'),

(10918,'M Vitae Ltd'),

(10919,'Lemper Oursus Foundation'),

(10920,'Pellus Opu Foundation');

---5) Insert Into Administrative Executive (5 Admins)----

INSERT INTO AdministrativeExecutive([AdminID],[AdminSubscriptionPrice],[AdminPermissions],[AdminJobTitle],[AdminJobDescription]) VALUES

(101,'Price of the subscription','No','Administrative Services Manager','Plan, direct, or coordinate an organization'),

(105,'Price of the subscription','Yes','Administrative Manager','Developing, reviewing, and improving administrative systems, policies, and procedures.'),

(111,'Price of the subscription','No','Administrative Services Manager','Plan, direct, or coordinate an organization'),

(112,'Price of the subscription','Yes','Administrative Executive Manager','Developing, reviewing, and improving administrative systems, policies, and procedures.'),

(113,'Price of the subscription','Yes','Administrative Senior Manager','Developing, reviewing, and improving administrative systems, policies, and procedures.');

----6) Insert Into SalesPerson (5 SalesPerson)----

INSERT INTO SalesPerson([SalesPersonID],[ShopName],[SalesPersonShopDetails]) VALUES

(102,'Erat Volutpat Nulla Associates','Donec dignissim'),

(106,'Orci Company','ipsum nunc'),

(108,'Velit Justo Limited','diam dictum'),

(110,'In PC','nulla. Donec'),

(115,'ILoen PC','nulla. Donec Lipessim');

---7) Insert Into Technician (5 Technician)---

INSERT INTO Technician([TechnicianID],[TechnicianJobTitle],[TechnicianWorkDetails]) VALUES

(103,'Specialty Systems','velit. Aliquam'),

(104,'Drone Operations','Aliquam rutrum'),

(107,'Drone Repair','sed dui.'),

(109,'Drone Test','quis, pede.'),

(114,'Drone Running','quis, pede.');

----8) Insert Into BTWebsite (20 Entries)----

INSERT INTO BTWebsite([BTSalesID],[BTSalesDetails]) VALUES

(192301,'leo.'),

(192302,'non'),

(192303,'placerat.'),

(192304,'a'),

(192305,'est.'),

(192306,'nec'),

(192307,'justo'),

(192308,'Nunc'),

(192309,'odio'),

(192310,'consectetuer'),

(192311,'ligula'),

(192312,'lacus.'),

(192313,'orci'),

(192314,'adipiscing'),

(192315,'vulputate,'),

(192316,'netus'),

(192317,'dolor.'),

(192318,'felis,'),

(192319,'felis'),

(192320,'nec');

-----9) Insert into the table of Subscription (20 Entries)---

INSERT INTO Subscription([SubscriberID],[SubscriptionType],[SubscriptionDetails],[SubscriptionPrice],[SubscriptionStartDate],[SubscriptionEndDate], [BTSalesID]) VALUES

(10921,'Standard,','Phasellus','NZ$ 1000','2018-11-14 18:54:54','2021-12-01 19:27:28', 192301),

(10922,'Gold','magna. Nam','NZ$ 2,500','2018-06-24 12:10:41','2022-08-26 13:53:23',192302),

(10923,'Platinum','gravida','NZ$ 4,700','2019-07-28 14:18:02','2021-04-14 12:11:17', 192303),

(10924,'Super Platinum','Integer','NZ$ 9,000','2018-07-24 00:58:18','2021-09-15 03:17:13', 192304),

(10925,'Standard','nec quam.','NZ$ 1000','2019-01-01 22:42:29','2021-05-10 02:40:41',192305),

(10926,'Gold','ultrices. Duis','NZ$ 2,500','2019-10-25 20:04:26','2020-12-06 13:36:40',192306),

(10927,'Platinum','ipsum','NZ$ 4,700','2018-08-07 12:54:54','2020-06-29 15:05:28',192307),

(10928,'Super Platinum','elit,','NZ$ 9,000','2018-08-06 07:33:09','2023-10-24 13:34:07', 192308),

(10929,'Standard','non, sollicitudin','NZ$ 1000','2020-03-22 10:53:24','2020-07-11 04:51:28', 192309),

(10930,'Gold','pellentesque massa','NZ$ 2,500','2019-10-07 21:32:33','2022-05-14 23:58:09', 192310),

(10931,'Platinum','in','NZ$ 4,700','2019-12-31 03:46:59','2018-12-10 08:30:55',192311),

(10932,'Super Platinum','ac tellus.','NZ$ 9,000','2018-08-23 21:24:18','2018-08-06 18:36:42',192312),

(10933,'Standard','tristique','NZ$ 1000','2019-09-25 17:32:10','2018-09-22 11:25:11',192313),

(10934,'Gold','dolor','NZ$ 2,500','2019-03-01 02:06:32','2018-12-14 03:05:57',192314),

(10935,'Platinum.','eleifend non,','NZ$ 4,700','2019-12-24 22:30:30','2018-06-02 12:34:40',192315),

(10936,'Super Platinum','nibh. Quisque','NZ$ 9,000','2018-08-04 14:22:27','2018-09-22 19:25:40',192316),

(10937,'Standard','nisi','NZ$ 1000','2019-10-29 18:05:48','2019-10-17 02:09:23',192317),

(10938,'Gold','bibendum','NZ$ 2,500','2020-01-01 04:13:29','2020-04-19 21:39:39',192318),

(10939,'Platinum','diam at','NZ$ 4,700','2020-02-15 12:32:00','2018-10-23 05:06:00',192319),

(10940,'Super Platinum','Nulla','NZ$ 9,000','2019-09-24 04:00:13','2018-08-23 07:24:54',192320);

----10) Insert into the Standard Table (5 Entries)---

INSERT INTO Standard([StandardSubscriberID]) VALUES

(10921),

(10925),

(10929),

(10933),

(10937);

----11) Insert into the Gold Table (5 Entries)---

INSERT INTO Gold([GoldSubscriberID]) VALUES

(10922),

(10926),

(10930),

(10934),

(10938);

---12) Table for Contract (30 Contracts)

INSERT INTO Contract([ContractID],[ContractType],[ContractDetails],[ContractStartDate],[ContractEndDate], [AdminID], [ContractorID]) VALUES

(2061,'sUAS Services Contract','nec, leo. Morbi','2018-12-24 13:25:34','2019-10-04 12:17:17', 101, 10901),

(2062,'Notice of UAV Operations','eget, volutpat ornare,','2018-08-24 21:14:53','2019-07-22 19:12:49', 105 , 10902),

(2063,'Pilot Contractor Agreement','lorem, luctus ut,','2018-08-20 08:43:43','2019-08-14 12:57:34', 111, 10903),

(2064,'Airspace Authorization','erat volutpat. Nulla','2019-02-05 20:10:46','2020-04-14 17:23:06', 112, 10904),

(2065,'Part 107 Operation Manual','urna. Nunc quis','2018-12-17 19:53:38','2019-10-18 10:53:02', 113, 10905),

(2066,'Copyright Assignment Forms','neque non quam.','2018-09-14 14:03:34','2019-12-15 05:39:01', 101, 10906),

(2067,'Single Member LLC','adipiscing elit. Etiam','2018-08-02 23:40:21','2019-10-04 16:09:13', 105, 10907),

(2068,'Copyright Assignment Forms','magna nec quam.','2018-11-08 07:56:03','2020-01-14 01:49:15', 111, 10908),

(2069,'Web Site Language','dictum ultricies ligula.','2019-01-26 18:49:08','2019-05-16 03:02:19', 112, 10909),

(2070,'Non-Compete Agreements','sem magna nec','2018-05-05 23:36:48','2019-10-29 22:06:02', 113, 10910),

(2071,'Trademark Registration','Pellentesque ultricies dignissim','2019-03-12 12:39:53','2019-11-04 21:14:20', 113, 10911),

(2072,'Copyright Registration','ac metus vitae','2018-11-20 07:08:08','2019-12-15 20:01:17', 113, 10912),

(2073,'Custom Contract Drafting','gravida non, sollicitudin','2018-07-09 11:53:22','2019-10-19 16:12:56', 113, 10913),

(2074,'Operations Procedure','habitant morbi tristique','2018-06-23 21:47:35','2019-10-17 22:05:54', 112, 10914),

(2075,'Custom flight simulation ','urna convallis erat,','2019-02-24 03:43:44','2020-01-07 14:36:55', 112, 10915),

(2076,'Patent Protection','mauris. Suspendisse aliquet','2019-02-19 12:41:47','2019-05-27 19:43:31', 112, 10916),

(2077,'Software Licensing','Duis elementum, dui','2018-08-31 12:21:23','2019-11-25 02:30:02', 112, 10917),

(2078,'FAA compliance','eleifend nec, malesuada','2018-12-09 02:45:39','2020-02-05 04:34:24', 111, 10918),

(2079,'UAS educational flight','pede, malesuada vel,','2018-06-07 07:12:30','2019-06-30 21:36:13', 111, 10919),

(2080,'UAS training','quis urna. Nunc','2018-11-01 05:32:00','2019-11-20 14:24:54', 111, 10920),

(2081,'FMRA','velit justo nec','2018-05-27 00:34:16','2019-06-16 01:13:06', 111, 10903),

(2082,'OPs','at arcu. Vestibulum','2019-01-05 10:39:07','2019-12-21 09:02:57', 105 , 10902),

(2083,'sUAS Service Agreement','Cras dictum ultricies','2018-09-07 21:26:49','2019-12-15 11:40:28', 105 , 10901),

(2084,'Notice of UAV Operations','non dui nec','2018-12-30 16:23:09','2019-07-13 05:57:34', 105 , 10909),

(2085,'Employee Non-Compete Forms','mollis non, cursus','2018-07-20 17:07:47','2020-03-13 22:35:57', 105 , 10911),

(2086,'Due Diligence Forms','sed leo. Cras','2018-05-28 23:29:23','2020-02-23 02:19:37', 101, 10912),

(2087,'Work For Hire Agreements','mauris elit, dictum','2018-05-08 02:17:34','2019-07-22 21:23:30', 101, 10915),

(2088,'Traverse Legal','ac turpis egestas.','2018-07-22 04:50:15','2019-06-01 14:18:40', 101, 10917),

(2089,'Safety Manual Drafting','lectus justo eu','2019-02-21 12:52:19','2019-07-12 07:45:26', 101, 10918),

(2090,'Prospection','Cum sociis natoque','2018-04-22 13:31:18','2019-04-30 11:41:36', 101, 10919);

---13) Insert into the Terrain table (14 Terrains)---

INSERT INTO Terrain([TerrainID],[TerrainName],[TerrainType]) VALUES

(9201801,'Rainforest','Jungle'),

(9201802,'Dense Woodlands','Jungle'),

(9201803,'Tropical forests','Forest'),

(9201804,'Boreal forests','Forest'),

(9201805,' Tropical','Savannahs'),

(9201806,'Subtropical','Savannahs'),

(9201807,'Coastal','Extreme Cold - Ice and Snow'),

(9201808,'Cold','Extreme Cold - Ice and Snow'),

(9201809,'Upwarped','Mountain'),

(9201810,'Volcanic','Mountain'),

(9201811,'Flat Plains','Desert'),

(9201812,'Sand Dunes','Desert'),

(9201813,'Towns','Urban'),

(9201814,'Cities','Urban');

----14) Insert into the table of Region ( 10 Regions)----

INSERT INTO Region([RegionID],[RegionDescription],[RegionType]) VALUES

(212101,'Małopolskie','City'),

(212102,'MB','Metropolitan Area'),

(212103,'Ontario','States'),

(212104,'Gl','Mountain range'),

(212105,'Istanbul','Functional region'),

(212106,'Andalucía','Himalayas'),

(212107,'Meghalaya','City'),

(212108,'Cape Hunululu','Towns'),

(212109,'Alabaros','Village'),

(212110,'Upper Penninsula','City');

----15) Insert into the BTDrone (40 Drones)---

INSERT INTO BTDrone([BTDroneID],[BTDroneLocation],[BTDroneHumidity],[BTDroneTemperature],[BTDroneLightStrength],[BTDroneLatitude],[BTDroneLongitude],[BTDroneAltitude],[BTDroneDetails], [RegionID]) VALUES

(2120301,'Borno','5.043','8.391','6.229','52.783','-64.99','9.399','aliquam', 212101),

(2120302,'WV','5.24','7.931','5.568','-27.70356','68.88357','8.759','metus', 212102),

(2120303,'SJ','5.08','8.084','5.858','-26.63402','-135.75772','8.873','facilisis.', 212103),

(2120304,'Wie','4.91','8.091','6.053','23.18896','-38.15491','9.621','in', 212104),

(2120305,'PR','4.783','7.697','6.075','-82.21218','17.26948','9.099','ac', 212105),

(2120306,'Gelderland','4.986','7.988','6.247','31.25187','10.42069','8.353','malesuada', 212106),

(2120307,'N.','4.51','7.875','5.611','6.24396','-83.65979','9.279','Aliquam', 212107),

(2120308,'Ohio','5.26','7.918','5.866','-69.39124','-15.61761','8.106','parturient', 212108),

(2120309,'L','4.785','7.648','5.359','72.81369','-12.95467','9.153','ad', 212109),

(2120310,'Västra Götalands län','4.989','8.246','5.67','-49.00388','53.76667','8.526','Cras', 212110),

(2120311,'Zeeland','4.945','7.624','6.034','19.29364','-13.11239','8.601','quam', 212101),

(2120312,'Connacht','5.077','7.427','5.937','75.40489','50.06528','9.021','ac', 212102),

(2120313,'Bavaria','4.566','7.844','5.784','-21.18254','2.53521','8.518','egestas', 212103),

(2120314,'AB','5.008','8.198','5.99','-86.63993','60.8566','9.187','amet', 212104),

(2120315,'Ontario','5.513','8.28','6.241','58.6717','-17.08248','8.691','tincidunt', 212105),

(2120316,'Connacht','5.276','8.086','6.101','-34.32445','-83.98094','7.837','quis', 212106),

(2120317,'Rajasthan','4.701','7.369','6.165','-87.32989','-100.263','9.081','aliquet.', 212108),

(2120318,'Istanbul','5.236','8.109','6.535','41.02128','-172.84123','9.375','Duis', 212109),

(2120319,'Antwerpen','5.069','7.931','6.202','-34.20439','160.11502','8.972','lectus.', 212110),

(2120320,'UT','5.2','8.443','5.87','23.68401','71.61684','9.092','sem', 212110),

(2120321,'Bavaria','5.249','7.561','6.109','-21.91991','-132.67947','8.947','In', 212110),

(2120322,'CAM','4.542','7.911','5.625','49.88942','119.73925','9.533','sed,', 212103),

(2120323,'HB','4.88','7.882','5.856','-61.60773','-19.50604','8.348','faucibus.', 212103),

(2120324,'Minnesota','5.036','7.869','6.489','-3.83381','51.88035','10.47','senectus',212104),

(2120325,'MB','5.058','7.682','6.38','-2.75607','-163.63666','9.49','risus',212104),

(2120326,'MB','4.813','8.138','5.623','15.20219','-13.59494','8.58','Pellentesque', 212107),

(2120327,'Mississippi','5.436','7.579','5.66','-3.62847','-95.39645','9.211','massa', 212107),

(2120328,'Vienna','4.99','7.7','5.963','22.23559','29.33488','9.594','aliquet.',212109),

(2120329,'WV','5.381','8.229','6.171','6.085','75.356','8.912','ultricies',212109),

(2120330,'OR','5.136','8.184','6.194','-89.06521','17.91291','8.539','velit', 212102),

(2120331,'Sardegna','4.694','7.862','5.544','10.72106','-78.7039','8.723','nibh.', 212102),

(2120332,'NSW','5.323','7.535','6.188','34.77119','-88.405','8.53','vel', 212101),

(2120333,'Leinster','4.883','8.073','5.78','-69.42886','12.295','8.982','sodales.', 212101),

(2120334,'AQ','4.897','8.098','5.452','-79.652','-13.56738','9.608','dui', 212108),

(2120335,'L','4.709','8.436','6.396','-46.333','-17.29055','8.405','justo.', 212108),

(2120336,'Minas Gerais','5.22','8.109','6.137','-80.02746','139.23919','8.297','nunc', 212106),

(2120337,'Z.','5.14','8.073','6.476','25.119','71.206','9.387','Lorem', 212105),

(2120338,'Osun','4.962','7.855','6.258','-0.13687','12.82','8.042','metus', 212105),

(2120339,'LOM','4.918','8.41','5.924','-5.942','-18.86','9','sed', 212106),

(2120340,'Provence-Alpes-Côte d''Azur','4.646','8.151','6.372','71.236','-93.639','9.83','odio', 212106);

----16) Insert into the Platinum Table (5 Entries)---

INSERT INTO Platinum([PlatinumSubscriberID],[BTDroneID]) VALUES

(10923,2120312),

(10927,2120321),

(10931,2120327),

(10935,2120335),

(10939,2120338);

----17) Insert into the Super Platinum Table (5 Entries)---

INSERT INTO SuperPlatinum([SuperPlatinumSubscriberID],[ContractID], [BTDroneID]) VALUES

(10924,2061,2120305),

(10928,2062,2120306),

(10932,2063,2120307),

(10936,2064,2120308),

(10940,2065,2120309);

---18) Insert into RoamingZone (15 Romaing Zones)----

INSERT INTO RoamingZone([RoamingZoneID],[RZDetails], [BTDroneID]) VALUES

(578201,'nisi', 2120301),

(578202,'Cras',2120302),

(578203,'vehicula',2120303),

(578204,'euismod',2120304),

(578205,'placerat, orci lacus',2120305),

(578206,'libero.',2120306),

(578207,'enim,',2120307),

(578208,'mauris',2120308),

(578209,'neque',2120309),

(578210,'arcu iaculis',2120310),

(578211,'interdum. Sed',2120311),

(578212,'Aliquam ultrices',2120312),

(578213,'dui nec',2120313),

(578214,'quis, pede. Praesent',2120314),

(578215,'magna',2120315);

----19) Insert into Altitude (6 Altitude Entities)---

INSERT INTO Altitude([AltitudeID],[MaxAltitude],[MinAltitude], [RoamingZoneID]) VALUES

(70101,'195.918','5.934', 578201),

(70102,'1289.671','9.136', 578205),

(70103,'659.659','10.954', 578207),

(70104,'189.683','13.951', 578209),

(70105,'89.944','14.177', 578211),

(70106,'490.172','9.914', 578215);

---20) Insert Into the table of ZoneBoundary (20 Latitude entries)----

INSERT INTO ZoneBoundary([ZoneBoundaryID],[LatitudeCalculated],[LongitudeCalculated],[RoamingZoneID]) VALUES

(201,'151.856','-30.05911', 578201),

(202,'258.749', '44.08713',578201),

(203,'-71.064','71.01679', 578201),

(204,'-86.157','-105.989', 578201),

(205,'-79.054','-138.815', 578202),

(206,'3.087','156.901', 578202),

(207,'110.865', '37.871', 578202),

(208,'-13.368','-80.621', 578202),

(209,'-19.764','-28.056', 578203),

(210,'68.753', '157.637',578203),

(211,'151.856', '-30.05911',578203),

(212,'258.749','44.08713', 578203),

(213,'-71.064', '71.01679',578204),

(214,'-86.157','-105.989', 578204),

(215,'-79.054','-138.815', 578204),

(216,'3.087','156.901', 578204),

(217,'110.865','-80.621', 578205),

(218,'-13.368','56.901', 578205),

(219,'-19.764','-28.056', 578205),

(220,'68.753','157.637', 578205);

---21) Insert Into the Maintenance Table (15 Maintenance Listed)---

INSERT INTO Maintenance([TechnicianID],[DateMaintained], [BTDroneID]) VALUES

(103,'2019-08-03 17:13:47', 2120301),

(104,'2019-06-23 05:23:17', 2120311),

(107,'2019-09-25 19:43:23', 2120313),

(109,'2019-08-27 20:38:02', 2120302),

(114,'2019-08-05 17:13:47', 2120315),

(103,'2019-04-29 05:23:17', 2120326),

(107,'2019-11-30 19:43:23', 2120328),

(109,'2019-10-27 20:38:02', 2120329),

(103,'2019-09-23 17:13:47', 2120333),

(104,'2019-01-13 05:23:17', 2120335),

(107,'2019-12-15 19:43:23', 2120337),

(109,'2019-07-17 20:38:02', 2120305),

(114,'2019-11-23 08:28:18', 2120307),

(107,'2019-01-05 09:41:23', 2120340),

(104,'2019-07-18 22:19:02', 2120327);

---22) Insert Into the MaintenanceParts (15 MainteanancePart)---

INSERT INTO MaintenancePart([DateMaintained],[MaintenancePartDetails]) VALUES

('2019-08-03 17:13:47','Flight Controller'),

('2019-06-23 05:23:17','Transmitter Reciever'),

('2019-09-25 19:43:23','Flight Controller'),

('2019-08-27 20:38:02','Power Distributor Board'),

('2019-08-05 17:13:47','Electronic Speed Controller'),

('2019-04-29 05:23:17','Multiple Motors'),

('2019-11-30 19:43:23','Multiple Props'),

('2019-10-27 20:38:02','Battery'),

('2019-09-23 17:13:47','Power Supply'),

('2019-01-13 05:23:17','Electronic Speed Controls'),

('2019-12-15 19:43:23','Propellers'),

('2019-07-17 20:38:02','Battery Charger'),

('2019-11-23 08:28:18','IR'),

('2019-01-05 09:41:23','Drone pad'),

('2019-07-18 22:19:02','GPS');

----23) Insert into the table of BTDronePart (15 parts)----

INSERT INTO BTDronePart([BTDronePartID],[BTDronePartName],[BTDronePartPartSIze],[BTDronePartWeight],[BTDronePartDetail], [DateMaintained]) VALUES

(546801,'Frame','14.844','72.303','also known as the chassis, breaks down, so does every other part of the drone','2019-08-03 17:13:47'),

(546802,'Transmitter Reciever','16.215','75.492','This is the electronic combo that allows you to directly control the aircraft','2019-06-23 05:23:17'),

(546803,'Flight Controller','14.232','76.822','the part of the machine that carries the software used to fly the drone','2019-09-25 19:43:23'),

(546804,'Power Distributor Board','15.762','74.928','electricity is distributed efficiently among these parts','2019-08-27 20:38:02'),

(546805,'Electronic Speed Controller','14.316','74.5','control the speed of the drone','2019-08-05 17:13:47'),

(546806,'Multiple Motors','14.848','71.523','lifting the machine off the ground','2019-04-29 05:23:17'),

(546807,'Multiple Props','13.629','75.959','props attached to the motors','2019-11-30 19:43:23'),

(546808,'Battery','15.323','76.411','providing electricity for the drone body','2019-10-27 20:38:02'),

(546809,'Power Supply','15.283','74.97','Run the drone','2019-09-23 17:13:47'),

(546810,'Electronic Speed Controls','14.855','73.704','how fast to spin at any given time','2019-01-13 05:23:17'),

(546811,'Propellers','14.899','75.526','To fly the drones','2019-07-17 20:38:02'),

(546812,'Battery Charger','14.047','74.748','Charger to chrage the Battery','2019-07-17 20:38:02'),

(546813,'IR','14.277','75.05','Infra red signals','2019-11-23 08:28:18'),

(546814,'Drone pad','14.677','78.326','Drone landing gear','2019-01-05 09:41:23'),

(546815,'GPS','15.356','73.36','location Detection','2019-07-18 22:19:02');

----24) Insert into the Supplier Table (20 Supplier supplying)---

INSERT INTO Supplier([SupplierID],[SupplierFName],[SupplierLName],[SupplierDetails],[SupplierStreetName],[SupplierSteetAddress],[SupplierCity],[SupplierSuburb],[SupplierZipCode],[SupplierPhoneNo],[SupplierEmail],[SupplierCompanyName]) VALUES

(1193101,'Merrill','Young','ullamcorper magna.','Eagan','Ap #441-5831 Pellentesque St.','Alajuela','A','36346','(02) 183 544 6415','Cum@nec.org','Macromedia'),

(1193102,'Tyler','Garrison','et arcu','Carlos','881 Sem Av.','Betim','Minas Gerais','50614','(02) 370 556 8475','cursus@cursusdiamat.com','Macromedia'),

(1193103,'Anthony','Livingston','tempor arcu.','Martena','P.O. Box 320, 2107 Sem Ave','Hamilton','ON','879857','(02) 472 987 2308','dui.Fusce.aliquam@nullamagnamalesuada.edu','Chami'),

(1193104,'Addison','Underwood','feugiat tellus','Scarlett','P.O. Box 494, 5979 Semper Avenue','Istanbul','Ist','2247','(02) 982 129 9904','lorem@Nuncuterat.ca','Cakewalk'),

(1193105,'Sonya','Hebert','senectus et','Daria','718-5689 Nunc St.','Gorinchem','Z.','757104','(02) 400 130 6825','ligula@aliquet.edu','Sibelius'),

(1193106,'August','Graham','tortor at','Lucian','P.O. Box 487, 7949 Aliquet Rd.','Palmilla','VI','6975 VH','(02) 120 955 1367','adipiscing.elit@Donecdignissimmagna.net','Finale'),

(1193107,'Mia','Carlson','Nunc lectus','Nicholas','6956 Lacinia St.','Hartford','CT','8631','(02) 762 043 1961','tincidunt.adipiscing@vel.org','Macromedia'),

(1193108,'Lars','Livingston','massa. Vestibulum','Tasha','127-8502 Quam Street','Saint Louis','Missouri','5796','(02) 107 451 3216','Phasellus@at.net','Sibelius'),

(1193109,'Macaulay','Carr','facilisis non,','Ella','P.O. Box 930, 1433 Aliquam Avenue','Wollongong','New South Wales','43961','(02) 081 908 2622','aptent.taciti@Cumsociisnatoque.co.uk','Microsoft'),

(1193110,'Scott','Clay','et pede.','Eagan','9327 Ac Street','Port Augusta','South Australia','75868','(02) 706 490 2164','Duis@veliteusem.net','Apple Systems'),

(1193111,'Adam','Contreras','non, egestas','Roanna','941 Sodales. Street','La Rochelle','Poitou-Charentes','8831 QP','(02) 436 890 1480','pede@Maecenas.ca','Altavista'),

(1193112,'Gareth','Petty','pharetra. Quisque','Murphy','475-1410 Pede. St.','Camaçari','Bahia','7802','(02) 716 161 1534','quis.urna@cursusdiamat.ca','Cakewalk'),

(1193113,'Lavinia','Rowe','nec tempus','Perry','805-7604 Orci. St.','Pangnirtung','NU','E5 3JI','(02) 054 041 1391','pede.nec@orci.net','Yahoo'),

(1193114,'Brenda','Morton','est. Mauris','Roth','169-1399 Mi, Ave','Belfast','U','644772','(02) 305 497 0111','morbi.tristique@sitametmassa.edu','Finale'),

(1193115,'Griffin','Webster','interdum enim','Kylie','P.O. Box 238, 2497 Facilisis. St.','Vienna','Wie','31817','(02) 135 566 8900','Class@mienim.org','Adobe'),

(1193116,'Hadassah','Oneal','molestie. Sed','Eliana','Ap #121-6377 Turpis. Road','Białystok','PD','28270','(02) 160 530 2214','et@massalobortisultrices.net','Yahoo'),

(1193117,'Leila','Heath','tortor nibh','Leah','679-8625 Duis Ave','Treguaco','VII','1516 ZC','(02) 620 872 0300','lectus@quamquisdiam.edu','Cakewalk'),

(1193118,'Imogene','Bright','ac mattis','Cooper','716 Lacus. Av.','Hamburg','HH','17972-131','(02) 723 978 8827','adipiscing.ligula@ultriciesornare.net','Sibelius'),

(1193119,'Elijah','Fry','eget, volutpat','Madison','514-4131 Tellus Rd.','Bath','ON','04106','(02) 810 465 6436','massa.lobortis.ultrices@velitPellentesqueultricies.org','Microsoft'),

(1193120,'India','Frank','tellus id','Hashim','P.O. Box 706, 2494 Proin Rd.','Rothes','MO','49790','(02) 090 810 5045','aliquet.lobortis.nisi@sitamet.com','Borland');

---25) Insert into the Video Stream (30 Entries)----

INSERT INTO VideoStream([VideoId],[VideoLength],[VideoQuality],[VideoDetails],[VideoTime], [BTSalesID] ,[StandardSubscriberID], [GoldSubscriberID]) VALUES

(216701,'02 Hour(s) 98 Minute(s)','aliquam','egestas','2019-08-06 21:50:40',192301 ,10921, 10922),

(216702,'00 Hour(s) 62 Minute(s)','imperdiet','eu, eleifend','2018-09-19 23:59:54',192302 , 10925, 10926),

(216703,'01 Hour(s) 91 Minute(s)','mauris,','ipsum porta','2019-12-30 04:06:03',192303 , 10929, 10930),

(216704,'03 Hour(s) 49 Minute(s)','eu','arcu.','2019-02-06 14:22:39',192304 , 10933, 10934),

(216705,'00 Hour(s) 37 Minute(s)','ut','risus. In','2018-08-19 22:57:15',192305 , 10937, 10938),

(216706,'00 Hour(s) 38 Minute(s)','est','Proin eget','2019-07-23 19:10:25', 192306 ,10921, 10922),

(216707,'03 Hour(s) 25 Minute(s)','Nullam','nec','2019-07-13 08:47:24',192307 , 10925, 10926),

(216708,'02 Hour(s) 47 Minute(s)','eget','lobortis','2019-11-06 11:09:38',192308 , 10929, 10930),

(216709,'01 Hour(s) 21 Minute(s)','mi','at, egestas','2019-03-30 13:26:23',192309 ,10921, 10922),

(216710,'05 Hour(s) 01 Minute(s)','ac','ipsum primis','2018-06-03 09:26:02',192310 ,10921, 10922),

(216711,'00 Hour(s) 33 Minute(s)','lorem','auctor','2019-12-20 17:00:53',192311 , 10921, 10922),

(216712,'00 Hour(s) 47 Minute(s)','nibh.','odio','2019-02-27 09:02:38',192312 , 10925, 10926),

(216713,'20 Hour(s) 44 Minute(s)','interdum.','Donec sollicitudin','2018-09-11 17:06:12',192313 , 10929, 10930),

(216714,'00 Hour(s) 99 Minute(s)','Sed','luctus','2019-12-14 22:14:21',192314 , 10933, 10934),

(216715,'00 Hour(s) 59 Minute(s)','Aenean','ullamcorper. Duis','2019-09-12 22:40:21',192315 ,10921, 10922),

(216716,'01 Hour(s) 58 Minute(s)','euismod','Donec','2018-07-03 13:51:54',192316 , 10921, 10922),

(216717,'04 Hour(s) 22 Minute(s)','eu,','Pellentesque ut','2019-02-28 20:59:26', 192317 ,10925, 10926),

(216718,'02 Hour(s) 88 Minute(s)','Fusce','molestie tellus.','2019-08-18 22:23:37', 192318 ,10929, 10930),

(216719,'00 Hour(s) 33 Minute(s)','non','arcu. Curabitur','2019-10-08 09:38:20', 192319 ,10933, 10934),

(216720,'01 Hour(s) 71 Minute(s)','neque.','sodales','2019-03-18 01:39:38',192320 ,10921, 10922),

(216721,'01 Hour(s) 45 Minute(s)','augue','Duis risus','2019-05-15 09:37:09', 192301 ,10921, 10922),

(216722,'05 Hour(s) 14 Minute(s)','sollicitudin','tristique senectus','2018-11-07 06:58:54',192302 , 10925, 10926),

(216723,'02 Hour(s) 65 Minute(s)','aliquam','facilisis, magna','2018-05-06 07:49:58', 192303 ,10929, 10926),

(216724,'05 Hour(s) 71 Minute(s)','rutrum,','sem egestas','2019-12-27 13:04:32', 192305 ,10933, 10934),

(216725,'03 Hour(s) 45 Minute(s)','tincidunt','ac','2018-12-07 11:28:37',192307 ,10921, 10922),

(216726,'03 Hour(s) 11 Minute(s)','aliquet','Praesent','2020-03-16 06:00:05',192308 , 10921, 10938),

(216727,'04 Hour(s) 02 Minute(s)','et','ut','2019-03-19 11:22:45',192309, 10925, 10934),

(216728,'00 Hour(s) 49 Minute(s)','Quisque','Nulla facilisis.','2019-01-27 05:06:41',192311 , 10929, 10930),

(216729,'02 Hour(s) 05 Minute(s)','hendrerit','non enim.','2018-10-15 15:16:50',192312 , 10933, 10926),

(216730,'01 Hour(s) 17 Minute(s)','ornare','est tempor','2019-08-01 00:18:01',192315 , 10937, 10922);

----26) Insert into the BTDrone Data (40 entries)---

INSERT INTO BTDroneData([BTDataID], [BTDroneLocation],[BTDroneHumidity],[BTDroneTemperature],[BTDroneLightStrength],[BTDroneLatitude],[BTDroneLongitude],[BTDroneAltitude], [ContractorID], [BTDroneID]) VALUES

(101,'Borno','5.043','8.391','6.229','52.783','-64.99','9.399', 10901, 2120301),

(102,'WV','5.24','7.931','5.568','-27.70356','68.88357','8.759', 10902, 2120302),

(103,'SJ','5.08','8.084','5.858','-26.63402','-135.75772','8.873', 10903, 2120303),

(104,'Wie','4.91','8.091','6.053','23.18896','-38.15491','9.621', 10904, 2120304),

(105,'PR','4.783','7.697','6.075','-82.21218','17.26948','9.099', 10905, 2120305),

(106,'Gelderland','4.986','7.988','6.247','31.25187','10.42069','8.353', 10906, 2120306),

(107,'N.','4.51','7.875','5.611','6.24396','-83.65979','9.279', 10907, 2120307),

(108,'Ohio','5.26','7.918','5.866','-69.39124','-15.61761','8.106', 10908, 2120308),

(109,'L','4.785','7.648','5.359','72.81369','-12.95467','9.153', 10909, 2120309),

(110,'Västra Götalands län','4.989','8.246','5.67','-49.00388','53.76667','8.526', 10910, 2120310),

(111,'Zeeland','4.945','7.624','6.034','19.29364','-13.11239','8.601', 10911, 2120311),

(112,'Connacht','5.077','7.427','5.937','75.40489','50.06528','9.021', 10912, 2120312),

(113,'Bavaria','4.566','7.844','5.784','-21.18254','2.53521','8.518', 10913, 2120313),

(114,'AB','5.008','8.198','5.99','-86.63993','60.8566','9.187', 10914, 2120314),

(115,'Ontario','5.513','8.28','6.241','58.6717','-17.08248','8.691', 10915, 2120315),

(116,'Connacht','5.276','8.086','6.101','-34.32445','-83.98094','7.837', 10916, 2120316),

(117,'Rajasthan','4.701','7.369','6.165','-87.32989','-100.263','9.081', 10917, 2120317),

(118,'Istanbul','5.236','8.109','6.535','41.02128','-172.84123','9.375', 10918, 2120318),

(119,'Antwerpen','5.069','7.931','6.202','-34.20439','160.11502','8.972', 10919, 2120319),

(120,'UT','5.2','8.443','5.87','23.68401','71.61684','9.092', 10920, 2120320),

(121,'Bavaria','5.249','7.561','6.109','-21.91991','-132.67947','8.947', 10901, 2120321),

(122,'CAM','4.542','7.911','5.625','49.88942','119.73925','9.533', 10902, 2120322),

(123,'HB','4.88','7.882','5.856','-61.60773','-19.50604','8.348', 10903,2120323),

(124,'Minnesota','5.036','7.869','6.489','-3.83381','51.88035','10.47', 10904, 2120324),

(125,'MB','5.058','7.682','6.38','-2.75607','-163.63666','9.49', 10905, 2120325),

(126,'MB','4.813','8.138','5.623','15.20219','-13.59494','8.58', 10906,2120326),

(127,'Mississippi','5.436','7.579','5.66','-3.62847','-95.39645','9.211', 10907, 2120327),

(128,'Vienna','4.99','7.7','5.963','22.23559','29.33488','9.594', 10908, 2120328),

(129,'WV','5.381','8.229','6.171','6.085','75.356','8.912', 10909, 2120329),

(130,'OR','5.136','8.184','6.194','-89.06521','17.91291','8.539', 10910, 2120330),

(131,'Sardegna','4.694','7.862','5.544','10.72106','-78.7039','8.723', 10911, 2120331),

(132,'NSW','5.323','7.535','6.188','34.77119','-88.405','8.53', 10912, 2120332),

(133,'Leinster','4.883','8.073','5.78','-69.42886','12.295','8.982', 10913, 2120333),

(134,'AQ','4.897','8.098','5.452','-79.652','-13.56738','9.608', 10914, 2120334),

(135,'L','4.709','8.436','6.396','-46.333','-17.29055','8.405', 10915, 2120335),

(136,'Minas Gerais','5.22','8.109','6.137','-80.02746','139.23919','8.297', 10916, 2120337),

(137,'Z.','5.14','8.073','6.476','25.119','71.206','9.387', 10917, 2120336),

(138,'Osun','4.962','7.855','6.258','-0.13687','12.82','8.042', 10918, 2120338),

(139,'LOM','4.918','8.41','5.924','-5.942','-18.86','9', 10919, 2120339),

(140,'Provence-Alpes-Côte d''Azur','4.646','8.151','6.372','71.236','-93.639','9.83', 10920, 2120340);

---27) Insert Into BTDroneTerrain (10 Entries)----

INSERT INTO BTDroneTerrain ([TerrainID], [BTDroneID]) VALUES

(9201801, 2120301),

(9201814, 2120302),

(9201803, 2120303),

(9201805, 2120304),

(9201806, 2120305),

(9201813, 2120306),

(9201808, 2120307),

(9201807, 2120308),

(9201806, 2120309),

(9201803, 2120310);

---28) Insert Into BTDroneSubscription (10 Entries)---

INSERT INTO BTDroneSubscription ([BTDroneID], [SubscriberID]) VALUES

(2120301, 10921),

(2120302, 10922),

(2120303, 10923),

(2120304, 10924),

(2120305, 10925),

(2120306, 10926),

(2120307, 10927),

(2120308, 10928),

(2120309, 10929),

(2120310, 10930);

----29) Insert into SubscriptionSales (10 Entries)---

INSERT INTO SubscriptionSales ([SalesPersonID], [SubscriberID]) VALUES

(106, 10922),

(108, 10923),

(110, 10925),

(115, 10926),

(102, 10927),

(108, 10928),

(110, 10929),

(115, 10930),

(102, 10933);

----30) Insert into BTDroneVideoStream (10 Entries)---

INSERT INTO BTDroneVideoStream ([VideoID], [BTDroneID]) VALUES

(216701, 2120301),

(216711, 2120305),

(216715, 2120309),

(216716, 2120310),

(216721, 2120326),

(216723, 2120318),

(216726, 2120333),

(216727, 2120336),

(216730, 2120339),

(216705, 2120340);

----31) Insert into PartSupplier (10 Entries)---

INSERT INTO PartSupplier ([BTDronePartID], [SupplierID]) VALUES

(546801, 1193101),

(546802, 1193103),

(546805, 1193105),

(546806, 1193106),

(546807, 1193109),

(546809, 1193111),

(546810, 1193112),

(546811, 1193113),

(546812, 1193114),

(546815, 1193115);

----32) Insert into BTDroneContract (10 Entries)---

INSERT INTO BTDroneContract([ContractID],[BTDroneID]) VALUES

(2061, 2120301),

(2062, 2120311),

(2063, 2120312),

(2065, 2120315),

(2067, 2120317),

(2068, 2120318),

(2069, 2120321),

(2071, 2120323),

(2073, 2120316),

(2080, 2120340);

---33) Insert into ContracteData (10 Entries)---

INSERT INTO ContractData ([ContractID], [BTDataID])VALUES

(2061, 101),

(2062, 102),

(2063, 103),

(2065, 110),

(2067, 111),

(2068, 115),

(2069, 116),

(2071, 117),

(2073, 118),

(2080, 124);

---34) Insert into ContractRoamingZone (10 Entries)---

INSERT INTO ContractRoamingZone ([ContractID], [RoamingZoneID])VALUES

(2061, 578201),

(2062, 578202),

(2063, 578203),

(2065, 578205),

(2067, 578206),

(2068, 578209),

(2069, 578210),

(2071, 578211),

(2073, 578213),

(2080, 578215);

---35) Insert into SubscriptionAdmin (20 Entries)---

INSERT INTO SubscriptionAdmin ([SubscriberID], [AdminID])VALUES

(10921, 101),

(10922, 105),

(10923, 111),

(10924, 112),

(10925, 113),

(10926, 101),

(10927, 105),

(10928, 111),

(10929, 112),

(10930, 113),

(10931, 101),

(10932, 105),

(10933, 111),

(10934, 112),

(10935, 113),

(10936, 101),

(10937, 105),

(10938, 111),

(10939, 112),

(10940, 113);

---36) Insert into VideoSubscription (10 Entries)---

INSERT INTO VideoSubscription ([VideoID],[SubscriberID])VALUES

(216705, 10921),

(216709, 10925),

(216711, 10929),

(216712, 10933),

(216713, 10937),

(216714, 10922),

(216716, 10926),

(216719, 10930),

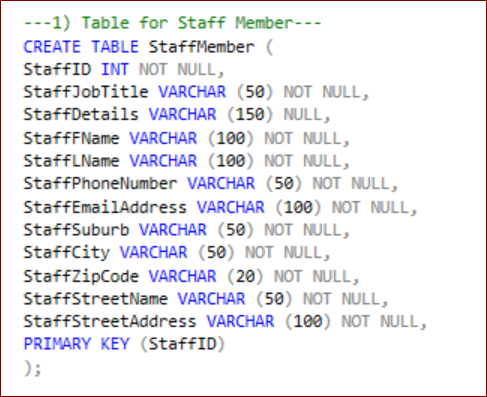
(216721, 10934),

(216728, 10938);

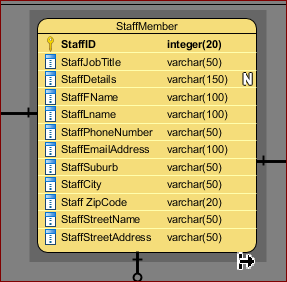
## Design Enterprise constraints for DBMS

For our, physical model of Being There Database, we are going to work with several types of constraints. These include:

**NOT NULL Constraint** – We are going to use a NOT NULL constraints in each table of our physical database model, so that we can prevent null values from being entered in one or more columns within a table. As such, we are going to use NOT NULL constraints in one of the tables like StaffMember.

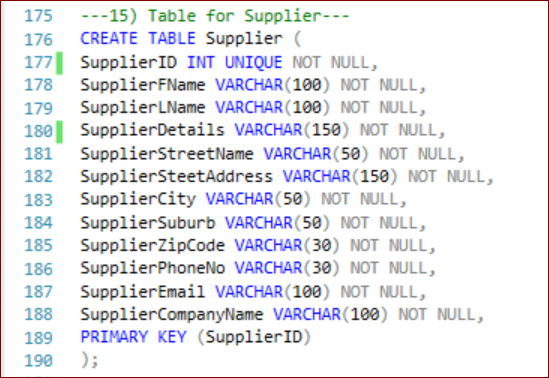


According to the physical model of Being There for StaffMember table, it is marked with ‘N’.

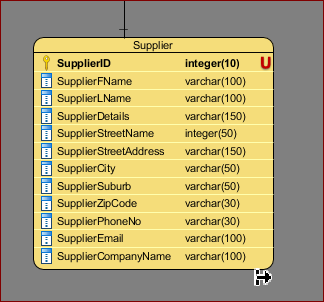


According to the above code on SQL Server, we can see that we are using **NOT NULL** on columns like StaffID, StaffJobTitle, StaffFName, StaffLName, StaffPhoneNumber, StaffEmailAddress, StaffSuburb, StaffCity, StaffZipCode, StaffStreetName, and StaffStreetAddress. This will prevent from entering null values to any of this column. However, we are using **NULL** constraints on StaffDetails. Here, we can enter null values. So, we aren’t restricting it.

**Unique Constraint** – For this database I am going to use unique constraints as identifier in certain number of tables. First, I will provide a brief description of the unique constraints. A unique constraint and the primary key of a certain table are quite related to each other. A unique constraint provides a rule that forbids any duplication in one or more columns within a table. As such we can use unique constraints for a supplier table, which will ensure us that the supplier id is unique for and only supplier, not to two suppliers. As such we are going to use the Unique Constraint in the Supplier table:



According to the physical model of Being There for Supplier table, it is marked with ‘U’, colored in Red.

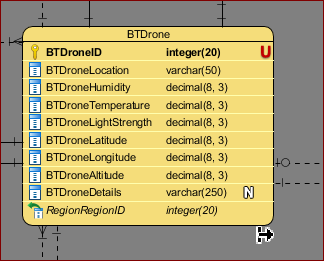


I used unique constraints for different tables of Being There database. As I have assigned unique number for each table of the database rather an auto generated number, I believe having a unique constraint with the primary key will prevent any duplication associated with more than one unique identifier.

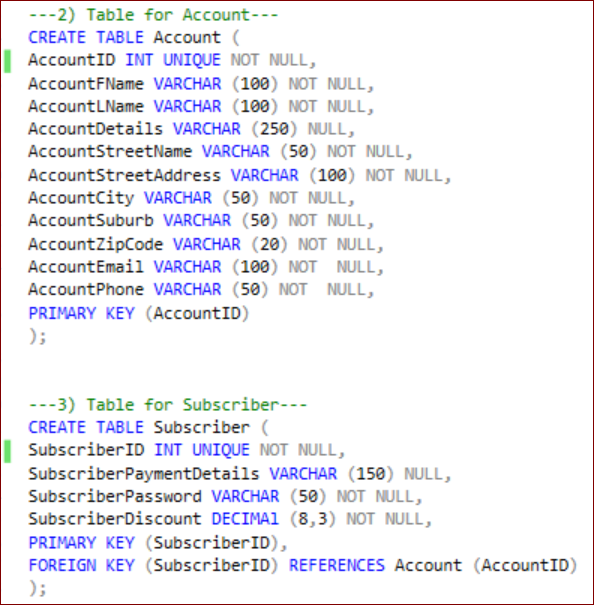
**Primary Key Constraint** – It is the combination of a NOT NULL and Unique constraints. They uniquely identify each row of a table. As such each drone used in the organization Being There, has its own unique ID of its own and when I will be creating the table of BTDrone, I will assign unique ID for each drone which will be different from each other and each ID of the drone can be used in further development of data development. As such in the SQL server, the following code is used to identify the Primary Key:



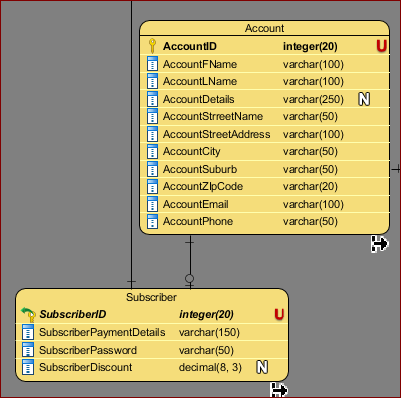
According to the physical model of Being There for BTDrone table, the Primary Key is marked with key and bold characters.



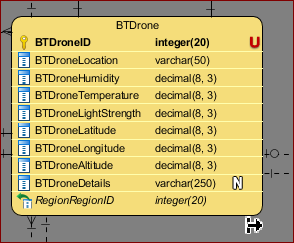
**Foreign KEY Constraint** – It is also known as referential constraints or referential integrity constraints, which enable us to define the required relationship within the tables (IBM, n.d.). To express further, a foreign key is a field or a collection of fields in one table that can be referred to another primary key in another table. As such, in the database Being There, we have account table and each account holder in the account table has their own primary key. That primary is taken to the subscriber table, who have the same ID in the subscriber table, as they have in the account table. This Foreign Key is linking between the Subscriber table and the Account Table. However, this Foreign Key, uniquely identifies the Subscriber table. As such, in the SQL Server, we used the following code:



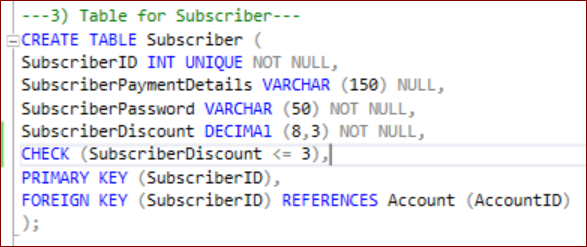
According to the physical model of Being There, in the Subscriber table, we can see that the Foreign Key is assigned with a Green Arrow. If the Foreign Key is the Primary Key of that table, it is assigned with bold character.



If the Foreign Key isn’t the primary key of a particular table like the Subscriber Table, but is there in certain table like BTDrone Table, we can see that it is connecting the Region table by providing the BTDrone table with a column of RegionID.

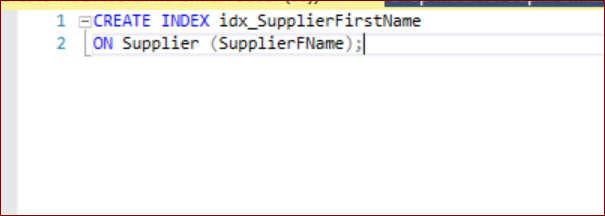


**Check Constraint** – It is another constraint that is used in to limit the value range that can be placed in a column. If I define a CHECK constraint in a single column of a certain database, it will allow certain values for that particular column. Again, if I define a CHECK constraint on a table, it can help me to limit the values in certain columns based on values in other columns of the row (W3Schools, 2019). I have used check constraints in my Subscriber table for SubsciberDiscount, as we are offering discount no more than 3%. The SQL code includes:



**DEFAULT Constraints** – It is a constraint that is used to provide a default value for a certain column. This default value will then be added to all the new record, if we don’t specify any other value. I haven’t used the Default Constraint in my Being There database.

**INDEX Constraints** – Indexes are type of constraints that are used to retrieve data from a database fast. The users of the database won’t be able to see the index; however, they can be used to speed up our searches or queries. In order to create an index, I will be using a single column or group of columns in a table. When I create the index, it will be assigned to a RowID for each row before a data is sorted out. As such, I created an index for the SupplierFName using the Supplier table. According to the SQL Server:



For Being There database, we need to provide the subscriber with 100 videos and we can have only 100 videos. So, if new video would be added beyond 100, we can create constraints to restrict it. For that, I created a function to restrict the VideoStream to 100. The SQL Server code includes:

--Create Function to check VideoStream constraints for 100 Videos--

DROP FUNCTION IF EXISTS dbo.VideoStreamCheck

GO

CREATE FUNCTION dbo.VideoStreamCheck (@VideoID AS INTEGER)

RETURNS INTEGER

BEGIN

DECLARE @ReturnValue int

SELECT @ReturnValue = COUNT (\*)

FROM VideoStream

WHERE @VideoID = VideoID

RETURN @ReturnValue

END

GO

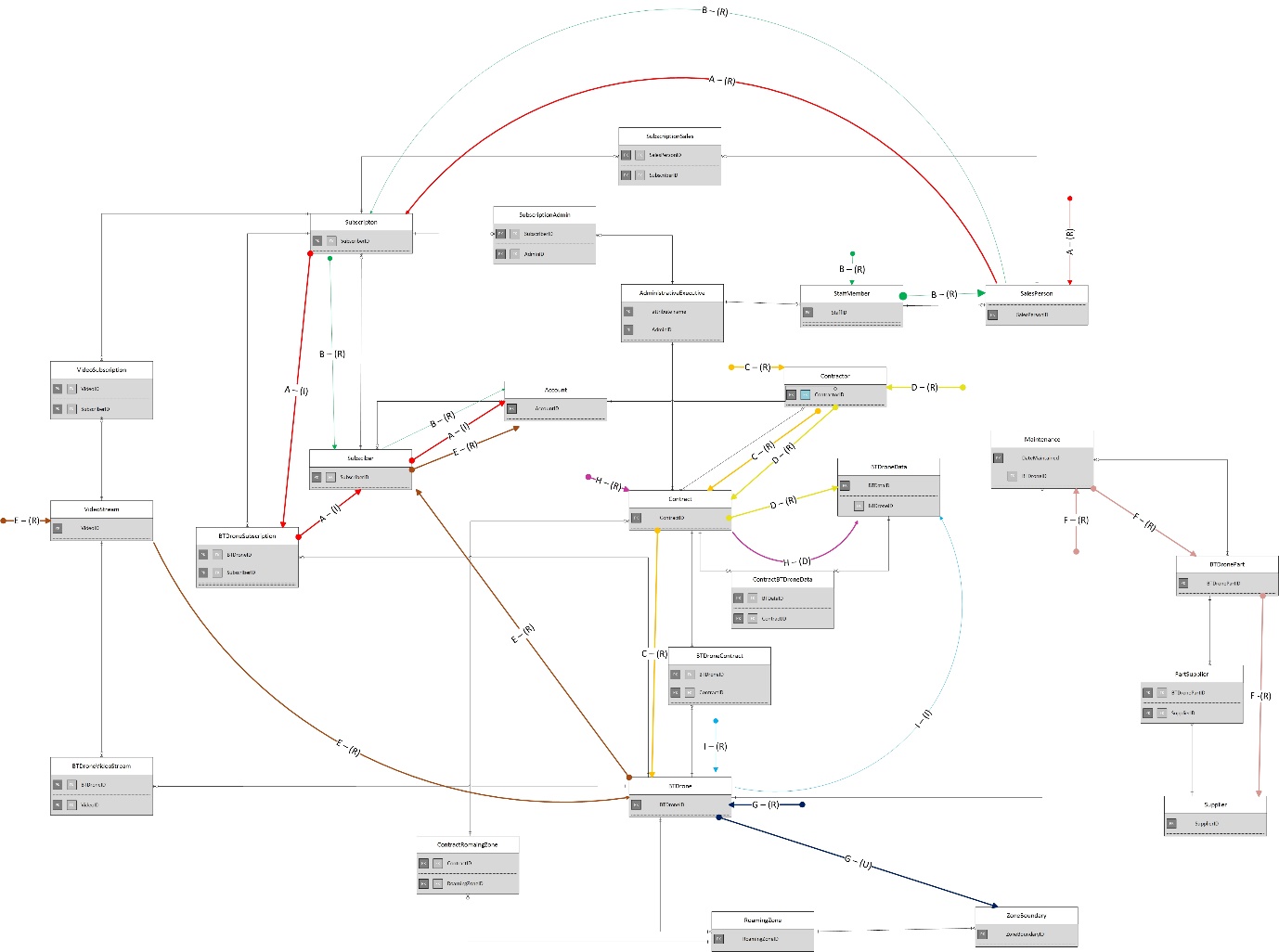
ALTER TABLE VideoStream

ADD CONSTRAINT CheckingVideoCount CHECK ((dbo.VideoStreamCheck(VideoID) >=216701) AND (dbo.VideoStreamCheck(VideoID) <=216799))

# Design Physical Representation

## Transaction Analysis with the Help of the Diagram

According to the procedures generated from the transactions, I have designed a transaction analysis of Being There Database. In the diagram below, I have shown all the procedures that I have worked on for this project. The diagram is included with all the procedures and each different color combination has been used to identify each procedure. I also have added a link to the diagram.



The link to the Transactional Analysis is:

<https://livenmitac-my.sharepoint.com/:b:/g/personal/ashkar-milton_live_nmit_ac_nz/ESF73gpTK3lLuWLdetA4CtgB7PfnoTAGY6-A6mq6phmhiA?e=EqxCqe>

## 

## Transaction A

A sales person subscribes a new standard subscription to a BTDrone. The transaction receives the sales person Id, a discount %, all subscriber details, and a BTDrone ID.

### Tabular Form

|  |  |  |  |
| --- | --- | --- | --- |
| Transaction A | | | |
|  | Day | Time | No. Runs Per Hour |
| Peak | -- | -- | -- |
| Average | -- | -- | Occasionally only |
| From relation | To relation | Attributes | Access | No. Times Accessed |
| -- | SalesPerson |  |  | 1 |
|  |  | SalesPersonID | R(E) |  |
| SalesPerson | Subscription |  |  | 1 |
|  |  | SubscriberID | I (E) |  |
|  |  | SubscriptionType | I |  |
|  |  | SubscriptionDetails | I |  |
|  |  | SubscriptionPrice | I |  |
|  |  | SubscriptionStartDate | I |  |
|  |  | SubscriptionEndDate | I |  |
| Subscription | BTDroneSubscription |  | I | 1 |
|  |  | BTDroneID | I |  |
|  |  | SubscriberID | I (E) |  |
| BTDroneSubscription | Subscriber |  | I | 1 |
|  |  | SubscriberID | I (E) |  |
|  |  | SubscriberPaymentDetails | I |  |
|  |  | SubscriberPassword | I |  |
|  |  | SubscriberDiscount | I |  |
| Subscriber | Account |  | I | 1 |
|  |  | AccountID | I (E) |  |
|  |  | AccountFName | I |  |
|  |  | AccountLName | I |  |
|  |  | AccountDetails | I |  |
|  |  | AccountStreetName | I |  |
|  |  | AccountStreetAddress | I |  |
|  |  | AccountCity | I |  |
|  |  | AccountSuburb | I |  |
|  |  | AccountZipCode | I |  |
|  |  | AccountEmail | I |  |
|  |  | AccountPhone | I |  |

### Description of Procedure A

The purpose of using this query is to add new subscriber to the subscription table of my database. Additionally, using this query I would like to find out who was the salesperson that sold the subscription and what drone the new subscriber is subscribed to. In this query I will use a SalesPerson who has his unique ID, SalesPersonID who would be adding a new standard subscriber. For that the SalesPerson will collect the SubscriberID and the BTDroneID, the drone that will be assigned to the new subscriber. The new subscriber will also receive a discount. Furthermore, in the account the new subscriber’s information will be added which include the subscriber First Name, Last Name, City, Suburb, Street Name, Street Address, Email, and Phone Number. The Subscriber ID will be same as the Account ID of the new subscriber. Finally, we will have a transaction of the new subscriber which includes the salesperson sold the subscription, the percentage discount, the subscriber’s details and the drone the subscriber is subscribed to.

### Procedure A

--Transaction A--

DROP PROCEDURE IF EXISTS TransactionA

GO

CREATE PROCEDURE TransactionA (

@AccountID AS INT,

@AccountFName AS VARCHAR (100),

@AccountLName AS VARCHAR (100),

@AccountDetails AS VARCHAR (250) = NULL,

@AccountStreetName AS VARCHAR (50),

@AccountStreetAddress AS VARCHAR (100),

@AccountCity AS VARCHAR (50),

@AccountSuburb AS VARCHAR (50),

@AccountZipCode AS VARCHAR (20),

@AccountEmail AS VARCHAR (100),

@AccountPhone AS VARCHAR (50),

@SubscriberID INT,

@SubscriberPaymentDetails AS VARCHAR (150),

@SubscriberPassword AS VARCHAR (50),

@SubscriberDiscount AS DECIMAL (8,3),

@SubscriptionType AS VARCHAR (50),

@SubscriptionDetails AS VARCHAR (250) = NULL,

@SubscriptionPrice AS VARCHAR (50),

@SubscriptionStartDate AS DATE,

@SubscriptionEndDate AS DATE,

@SalesPersonID AS INT,

@BTDroneID AS INT

)

AS

BEGIN

INSERT Account (AccountID, AccountFName, AccountLName, AccountDetails, AccountStreetName, AccountStreetAddress, AccountCity, AccountSuburb, AccountZipCode, AccountEmail, AccountPhone)

VALUES (@AccountID, @AccountFName, @AccountLName,@AccountDetails, @AccountStreetName, @AccountStreetAddress, @AccountCity, @AccountSuburb, @AccountZipCode, @AccountEmail, @AccountPhone)

INSERT Subscriber (SubscriberID, SubscriberPaymentDetails, SubscriberPassword, SubscriberDiscount)

VALUES (@SubscriberID, @SubscriberPaymentDetails, @SubscriberPassword, @SubscriberDiscount)

INSERT SubscriptionSales (SalesPersonID, SubscriberID)

VALUES (@SalesPersonID, @SubscriberID)

INSERT Subscription (SubscriberID, SubscriptionType, SubscriptionDetails, SubscriptionPrice, SubscriptionStartDate, SubscriptionEndDate)

VALUES (@SubscriberID, @SubscriptionType, @SubscriptionDetails, @SubscriptionPrice, @SubscriptionStartDate, @SubscriptionEndDate)

INSERT BTDroneSubscription (BTDroneID, SubscriberID)

VALUES (@BTDroneID, @SubscriberID)

SELECT SalesPerson.SalesPersonID, Subscriber.SubscriberDiscount, Account.AccountID, Account.AccountFName, Account.AccountLName, Account.AccountStreetName, Account.AccountStreetAddress, Account.AccountStreetAddress, Account.AccountCity, Account.AccountSuburb, Account.AccountZipCode, Account.AccountEmail, Account.AccountPhone, BTDroneSubscription.BTDroneID

FROM SalesPerson

JOIN SubscriptionSales ON SubscriptionSales.SalesPersonID = SalesPerson.SalesPersonID

JOIN Subscription ON Subscription.SubscriberID = SubscriptionSales.SubscriberID

JOIN BTDroneSubscription ON BTDroneSubscription.SubscriberID = Subscription.SubscriberID

JOIN BTDrone ON BTDroneSubscription.BTDroneID = BTDrone.BTDroneID

JOIN Subscriber ON Subscriber.SubscriberID = Subscription.SubscriberID

JOIN Account ON Account.AccountID = Subscriber.SubscriberID

WHERE dbo.SalesPerson.SalesPersonID = @SalesPersonID

END

EXECUTE TransactionA '10942', 'Logan', 'Lolferin','None', 'BNC', '55 Norkshire St', 'New York Central', 'Manhattan', '2011', 'Mogan.podolf619@gmail.com', '(02) 420 548 4555', '10942','Debit Card', 'x0PLmssI99kaLL','6.222', 'Standard', 'None', 'NZ$ 1,000', '2018-12-01 15:27:28', '2021-12-01 19:37:48','102', '2120320';

## Transaction B

For each sales person list the subscribers they have sold a subscription to. The transaction receives the sales person's name as input, and presents each subscriber's name, address, and the % they were discounted.

### Tabular Form

|  |  |  |  |
| --- | --- | --- | --- |
| Transaction A | | | |
|  | Day | Time | No. Runs Per Hour |
| Peak | -- | -- | -- |
| Average | -- | -- | Occasionally only |
| From relation | To relation | Attributes | Access | No. Times Accessed |
| -- | StaffMember |  |  | 1 |
|  |  | StaffID | R(E) |  |
|  |  | StaffFName | R |  |
| StaffMemeber | SalesPerson |  |  | 1 |
|  |  | SalesPersonID | R(E) |  |
|  |  |  |  |  |
| SalesPerson | Subscription |  |  | 1 |
|  |  | SubscriberID | R (E) |  |
| Subscription | Subscriber |  |  | 1 |
|  |  | SubscriberID | R (E) |  |
|  |  | SubscriberDiscount | R |  |
| Subscriber | Account |  |  | 1 |
|  |  | AccountID | R (E) |  |
|  |  | AccountFName | R |  |
|  |  | AccountLName | R |  |
|  |  | AccountStreetName | R |  |
|  |  | AccountStreetAddress | R |  |
|  |  | AccountCity | R |  |
|  |  | AccountSuburb | R |  |
|  |  | AccountZipCode | R |  |
|  |  | AccountEmail | R |  |
|  |  | AccountPhone | R |  |

### Description of Procedure B

The purpose of this query is to find the list of subscribers by entering a salesperson name who have sold the subscription to those subscribers. This query will display the name and information of each subscribers when the Salesperson name has been used as an input. Additionally, this query will also display those subscribers discount percentage in subscription. For this query we will use the salesperson ID to connect with the subscription to find the subscriber ID. From the subscription table we will move towards the subscriber table, where we will get the subscriber discount. Finally, we will go the account table to get the subscriber first name, last name, and the required addresses. We will use this query to obtain all the subscriber’s details.

### Procedure B

--Transaction B--

DROP PROCEDURE IF EXISTS TransactionB

GO

CREATE PROCEDURE TransactionB (@StaffFName AS VARCHAR (100))

AS

BEGIN

SELECT Account.AccountFName, Account.AccountLName, Account.AccountStreetName, Account.AccountStreetAddress, Account.AccountCity, Account.AccountSuburb, Account.AccountZipCode, SubscriberDiscount

FROM Subscriber

JOIN Account ON Subscriber.SubscriberID = Account.AccountID

JOIN Subscription ON Subscription.SubscriberID = Subscriber.SubscriberID

JOIN SubscriptionSales ON Subscription.SubscriberID = SubscriptionSales.SubscriberID

JOIN StaffMember ON SubscriptionSales.SalesPersonID = StaffMember.StaffID

WHERE StaffMember.StaffFName = @StaffFName

END;

EXECUTE TransactionB 'Farrah';

EXECUTE TransactionB 'Buffy';

## Transaction C

List the location in latitude, longitude coordinates, of each BTDrone that is currently

in a contract. The transaction presents the Contracting organization's name, a BTDrone ID, a Latitude, and a Longitude.

### Tabular Form

|  |  |  |  |
| --- | --- | --- | --- |
| Transaction A | | | |
|  | Day | Time | No. Runs Per Hour |
| Peak | -- | -- | -- |
| Average | -- | -- | Occasionally only |
| From relation | To relation | Attributes | Access | No. Times Accessed |
| -- | Contractor |  |  | 1 |
|  |  | ContractorID | R (E) |  |
|  |  | ContractorOrganizationName | R |  |
| Contractor | Contract |  |  | 1 |
|  |  | ContractID | R (E) |  |
| Contract | BTDrone |  |  | 1 |
|  |  | BTDroneID | R (E) |  |
|  |  | BTDroneID | R |  |
|  |  | BTDroneLongitude | R |  |
|  |  | BTDroneLongitude | R |  |

### Description of Procedure C

The purpose of this query is to find a list of location composed of latitude and longitude for a certain BTDrone that has been assigned to a contract. This query will display the organization name of the contract along with the unique ID of the drone, latitude, and longitude. For this query, first I will go to the Contractor table to get the Organization Name, then I will move to the Contract Table to get the list of ContractID. Finally, I will move to the BTDrone table to get the drone to get the BTDroneID related with the contract and each BTDrones existing latitude and longitude.

### Procedure C

--Transaction C--

DROP PROCEDURE IF EXISTS TransactionC

GO

CREATE PROCEDURE TransactionC

AS

BEGIN

SELECT ContractorOrganizationName,BTDrone.BTDroneID, BTDrone.BTDroneLongitude, BTDrone.BTDroneLatitude

FROM Contract

JOIN Contractor ON Contractor.ContractorID = Contract.ContractorID

JOIN BTDroneContract ON BTDroneContract. ContractID = Contract.ContractID

JOIN BTDrone ON BTDrone.BTDroneID = BTDroneContract.BTDroneID

END;

EXECUTE TransactionC;

## Transaction D

For a contract list all the data collected. The transaction receives the contracting organisation's name and presents for each collected data record, the contracting organisation's name, a BTDrone ID, Temperature, Humidity and Ambient light strength.

### Tabular Form

|  |  |  |  |
| --- | --- | --- | --- |
| Transaction A | | | |
|  | Day | Time | No. Runs Per Hour |
| Peak | -- | -- | -- |
| Average | -- | -- | Occasionally only |
| From relation | To relation | Attributes | Access | No. Times Accessed |
| -- | Contractor |  |  | 1 |
|  |  | ContractorID | R(E) |  |
|  |  | ContractorOrganizationName | R |  |
| Contractor | Contract |  |  | 1 |
|  |  | ContractID |  |  |
| Contract | BTDroneContract |  |  | 1 |
|  |  | ContractID | R (E) |  |
|  |  | BTDroneID | R |  |
| BTDroneContract | BTDroneData |  |  | 1 |
|  |  | BTDroneID | R (E) |  |
|  |  | BTDroneTemperature | R |  |
|  |  | BTDroneHumidity | R |  |
|  |  | BTDroneLightStrength | R |  |

### Description of Procedure D

The purpose of this procedure is to collect all the data collected for each contract. This query will be be taking the organization name as an input and we will be able to list the organization name, the drone ID, temperature, humidity, and light strength from the drone data. For this query, we will first go to the contractor table to get the organization name and then we will be going to the contract tablet to get the contract ID. Finally, we will go to the table of Drone data to get the Drone ID, Temperature, Humidity, and Light Strength recorded by each drone.

### Procedure D

--Transaction D--

DROP PROCEDURE IF EXISTS TransactionD

GO

CREATE PROCEDURE TransactionD (@ContractOrgnanizationName AS VARCHAR (250))

AS

BEGIN

---USE BEING THERE DATABASE

SELECT Contractor.ContractorOrganizationName, BTDroneData.BTDroneID, BTDroneData.BTDroneTemperature , BTDroneData.BTDroneHumidity, BTDroneData.BTDroneLightStrength

FROM Contractor

JOIN Contract ON Contract.ContractorID = Contractor.ContractorID

JOIN ContractData ON ContractData. ContractID = Contract.ContractID

JOIN BTDroneData ON BTDroneData.BTDataID = ContractData.BTDataID

WHERE dbo.Contractor.ContractorOrganizationName = @ContractOrgnanizationName

END;

EXECUTE TransactionD 'Tempor Company';

## Transaction E

For each BTDrone present the list of subscribers who are viewing a livestream. The transaction lists BTDroneID, Subscriber Name, Stream ID.

### Tabular Form

|  |  |  |  |
| --- | --- | --- | --- |
| Transaction A | | | |
|  | Day | Time | No. Runs Per Hour |
| Peak | -- | -- | -- |
| Average | -- | -- | Occasionally only |
| From relation | To relation | Attributes | Access | No. Times Accessed |
| -- | VideoStream |  |  | 1 |
|  |  | VideoID | R (E) |  |
| VideoStream | BTDrone |  |  | 1 |
|  |  | BTDroneID | R |  |
| BTDrone | Subscription |  |  | 1 |
|  |  | SubscriberID | R (E) |  |
| Subscription | Account |  |  |  |
|  |  | AccountID | R (E) |  |
|  |  | AccountFName | R |  |
|  |  | AccountLName | R |  |

### Description of Procedure E

The purpose of this query is to provide us the list of subscribers who are viewing the live 3D video stream obtained by each drone. In order to run the query, I will be typing the drone ID as an input which will result in connecting different tables by displaying the Drone ID, the name of the subscriber who is viewing the live 3D Video stream, and the Video stream ID. For this query, first I will go to the VideoStream table to get the VideoID. Then I will go to the BTDrone table to obtain the drone ID and from there I will move to the subscriber table to get the subscriber ID. Finally, I will go to the Account table to obtain the subscriber’s first name and last name.

### Procedure E

--Transaction E--

DROP PROCEDURE IF EXISTS TransactionE

GO

CREATE PROCEDURE TransactionE

AS

BEGIN

SELECT BTDrone.BTDroneID, VideoStream.VideoID, Account.AccountFName, Account.AccountLName

FROM BTDrone

JOIN BTDroneVideoStream ON BTDrone.BTDroneID = BTDroneVideoStream.BTDroneID

JOIN VideoStream ON VideoStream.VideoID = BTDroneVideoStream.VideoID

JOIN VideoSubscription ON VideoSubscription.VideoID = VideoStream.VideoID

JOIN Subscription ON VideoSubscription. SubscriberID = Subscription.SubscriberID

JOIN Subscriber ON Subscription.SubscriberID = Subscriber.SubscriberID

JOIN Account ON Account.AccountID = Subscription.SubscriberID

END;

EXECUTE TransactionE1;

## Transaction F

For a given BTDrone list all the suppliers of parts. The transaction receives the BTDrone ID, and presents the Supplier Name and, Part Name.

### Tabular Form

|  |  |  |  |
| --- | --- | --- | --- |
| Transaction A | | | |
|  | Day | Time | No. Runs Per Hour |
| Peak | -- | -- | -- |
| Average | -- | -- | Occasionally only |
| From relation | To relation | Attributes | Access | No. Times Accessed |
| -- | Maintenance |  |  | 1 |
|  |  | BTDroneID | R(E) |  |
| Maintenance | BTDronePart |  |  | 1 |
|  |  | BTDronePartID | R (E) |  |
|  |  | BTDronePartName | R |  |
| BTDronePart | Supplier |  |  | 1 |
|  |  | SupplierID | R (E) |  |
|  |  | SupplierFName | R |  |

### Description of Procedure F

The purpose of this query is to list all the suppliers name who are associate with the part of each drone. To run this query, I will be taking the drone Id as an input which will thus help me to connect with different tables to obtain the Supplier Name and the Part name of the drones. For this query, first I will go to the Maintenance table to get the drone Id that is related with the part. From there I will move to the BTDrone part to match the part id with the drone ID and get the name of the part. Finally, I will go to the supplier’s table to get the list of names of supplier who are associated with certain parts of a certain drone.

### Procedure F

--Transaction F--

DROP PROCEDURE IF EXISTS TransactionF

GO

CREATE PROCEDURE TransactionF (@BTDroneID AS INT)

AS

BEGIN

SELECT Supplier.SupplierFName, Supplier.SupplierLName, BTDronePart.BTDronePartName

FROM BTDrone

JOIN Maintenance ON BTDrone.BTDroneID = Maintenance.BTDroneID

JOIN BTDronePart ON BTDronePart.DateMaintained = Maintenance.DateMaintained

JOIN PartSupplier ON PartSupplier.BTDronePartID = BTDronePart.BTDronePartID

JOIN Supplier ON PartSupplier.SupplierID = Supplier.SupplierID

WHERE dbo.BTDrone.BTDroneID = @BTDroneID

END;

EXECUTE TransactionF'2120305';

EXECUTE TransactionF'2120301';

EXECUTE TransactionF'2120315';

## Transaction G

Update the location and region of a BTDrone. The transaction receives the BTDroneID, a location and a region expressed as a list of coordinates in latitude, longitude pairs. It updates the location of the BTDrone and its corresponding region.

### Tabular Form

|  |  |  |  |
| --- | --- | --- | --- |
| Transaction A | | | |
|  | Day | Time | No. Runs Per Hour |
| Peak | -- | -- | -- |
| Average | -- | -- | Occasionally only |
| From relation | To relation | Attributes | Access | No. Times Accessed |
| -- | BTDrone |  |  | 1 |
|  |  | BTDroneID | R (E) |  |
|  |  | BTDroneLatitude | R |  |
|  |  | BTDroneLongitude | R |  |
| BTDrone | ZoneBoundary |  |  |  |
|  |  | RomainZoneID | U (E) |  |
|  |  | LatitudeCalculated | U |  |
|  |  | LongitudeCalculated | U |  |

### Description of Procedure G

The purpose of this query is to update the location and region of a BTDrone. This query will allow us to update the latitude and longitude of the drone. To do that, we will input the drone ID and the longitude and latitude of each drone in different zone. This will then be updated to provide us with the location and region of the drone. For this query to work, first we will go to the BTDrone to obtain the drone ID, and the current latitude and longitude of that drone. Finally, we will move to the ZoneBoundary table to update the RoamingZoneID, the latitude , and the longitude.

### Procedure G

--Transaction F--

DROP PROCEDURE IF EXISTS TransactionG

GO

CREATE PROCEDURE TransactionG2 (@BTDroneID AS INT, @ZoneBoundaryID1 AS INT, @ZoneBoundaryID2 AS INT,@ZoneBoundaryID3 AS INT,@ZoneBoundaryID4 AS INT)

AS

BEGIN

UPDATE BTDrone

SET BTDroneLatitude = 52.156, BTDroneLongitude = -25.189

WHERE (BTDroneID = @BTDroneID);

UPDATE ZoneBoundary

SET LatitudeCalculated = -45.126, LongitudeCalculated = -155.38

WHERE (ZoneBoundaryID = @ZoneBoundaryID1);

UPDATE ZoneBoundary

SET LatitudeCalculated = 55.126, LongitudeCalculated = -895.248

WHERE (ZoneBoundaryID = @ZoneBoundaryID2);

UPDATE ZoneBoundary

SET LatitudeCalculated = 35.466, LongitudeCalculated = -205.788

WHERE (ZoneBoundaryID = @ZoneBoundaryID3);

UPDATE ZoneBoundary

SET LatitudeCalculated = -125.126, LongitudeCalculated = -23.788

WHERE (ZoneBoundaryID = @ZoneBoundaryID4);

SELECT BTDrone.BTDroneID, BTDrone.BTDroneLatitude, BTDrone.BTDroneLongitude, ZoneBoundary.LatitudeCalculated, ZoneBoundary.LongitudeCalculated

FROM BTDrone

JOIN RoamingZone ON RoamingZone.BTDroneID = BTDrone.BTDroneID

JOIN ZoneBoundary ON ZoneBoundary.RoamingZoneID = RoamingZone.RoamingZoneID

WHERE dbo.BTDrone.BTDroneID = @BTDroneID

END;

EXECUTE TransactionG 2120301, 201, 202, 203, 204;

## Transaction H

Delete the data collected for a given Contract. The transaction receives a ContractID, the data collected for a Contract is deleted

### Tabular Form

|  |  |  |  |
| --- | --- | --- | --- |
| Transaction A | | | |
|  | Day | Time | No. Runs Per Hour |
| Peak | -- | -- | -- |
| Average | -- | -- | Occasionally only |
| From relation | To relation | Attributes | Access | No. Times Accessed |
| -- | Contract |  |  | 1 |
|  |  | ContractID | R(E) |  |
| Contract | BTDroneData |  |  | 1 |
|  |  | BTDataID | D (E) |  |
|  |  | BTDroneLocation | D |  |
|  |  | BTDroneHumidity | D |  |
|  |  | BTDroneTemperature | D |  |
|  |  | BTDroneLightStrength | D |  |
|  |  | BTDroneLatitude | D |  |
|  |  | BTDroneLongitude | D |  |
|  |  | BTDroneAltitude | D |  |

### Description of Procedure H

The purpose of this query is deleting the data that has been collected by each individual contract. In order to work this query, we be using the contract ID as an input which will provide me the idea on which Contract should be deleted. For this query, first I will go to the Contract table to search the contract that has been entered by me, and from there I will go the BTDroneData to match the information associated with that contract and delete the information associated with the contract.

### Procedure H

--Transaction H--

DROP PROCEDURE IF EXISTS TransactionH

GO

CREATE PROCEDURE TransactionH1 (@ContractID AS INT)

AS

BEGIN

DELETE

FROM ContractData

WHERE BTDataID IN

(SELECT BTDataID

FROM ContractData

WHERE ContractID = @ContractID);

SELECT Contract.ContractID, BTDronedata.BTDroneLocation, BTDroneData.BTDroneHumidity, BTDroneData.BTDroneTemperature, BTDroneData.BTDroneLightStrength, BTDroneData.BTDroneLatitude, BTDroneData.BTDroneLongitude, BTDroneData.BTDroneAltitude

FROM Contract

LEFT JOIN ContractData ON ContractData.ContractID = Contract.ContractID

LEFT JOIN BTDroneData ON BTDroneData.BTDroneID = ContractData.BTDataID

WHERE Contract.ContractID = @ContractID

END;

EXECUTE TransactionH 2061;

## Transaction I

Insert data from a BTDrone to its stored data. The transaction receives the BTDroneID.

### Tabular Form

|  |  |  |  |
| --- | --- | --- | --- |
| Transaction A | | | |
|  | Day | Time | No. Runs Per Hour |
| Peak | -- | -- | -- |
| Average | -- | -- | Occasionally only |
| From relation | To relation | Attributes | Access | No. Times Accessed |
| -- | BTDrone |  |  | 1 |
|  |  | BTDroneID | R(E) |  |
|  |  | BTDroneLocation | R |  |
|  |  | BTDroneHumidity | R |  |
|  |  | BTDroneTemperature | R |  |
|  |  | BTDroneLightStrength | R |  |
|  |  | BTDroneLatitude | R |  |
|  |  | BTDroneLongitude | R |  |
|  |  | BTDroneAltitude | R |  |
| BTDrone | BTDroneData |  |  |  |
|  |  | BTDataID | I (E) |  |
|  |  | BTDroneLocation | I |  |
|  |  | BTDroneHumidity | I |  |
|  |  | BTDroneTemperature | I |  |
|  |  | BTDroneLightStrength | I |  |
|  |  | BTDroneLatitude | I |  |
|  |  | BTDroneLongitude | I |  |
|  |  | BTDroneAltitude | I |  |

### Description of Procedure I

The purpose of this query is to insert the data from a drone to its stores data. This query is used to transfer information from one table to another. In order to run this query, I will be using the drone ID as an input as this will tell me which drone’s information need to be transferred to the stored data. For this query, first I will go the BTDrone table to find the drone according to the drone ID input, and then I will take all the information like location, humidity, latitude, longitude, temperature, light strength, and altitude associated with that drone. Then I will move to the BTDroneData table to transfer that information for that drone. This procedure allows me to insert data from one table to another.

### Procedure I

--Transaction I----

--DROP PROCEDURE IF EXISTS TransactionI;--

CREATE PROCEDURE TransactionI2 (@UpdateBTDroneID AS INT)

AS

BEGIN

INSERT

dbo.BTDroneData (BTDataID, BTDroneLocation, BTDroneHumidity,BTDroneTemperature, BTDroneLightStrength, BTDroneLatitude, BTDroneLongitude, BTDroneAltitude)

SELECT

BTDrone.BTDroneID, BTDrone.BTDroneLocation, BTDrone.BTDroneHumidity,BTDrone.BTDroneTemperature, BTDrone.BTDroneLightStrength, BTDrone.BTDroneLatitude, BTDrone.BTDroneLongitude, BTDrone.BTDroneAltitude

FROM

dbo.BTDrone

WHERE dbo.BTDrone.BTdroneID = @UpdateBTDroneID

SELECT BTDataID, BTDroneLocation, BTDroneHumidity,BTDroneTemperature, BTDroneLightStrength, BTDroneLatitude, BTDroneLongitude, BTDroneAltitude

FROM BTDroneData

END

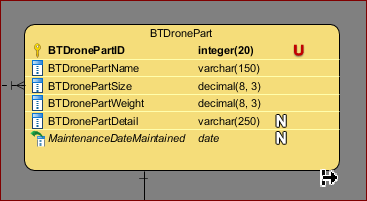
EXECUTE TransactionI 2120303;

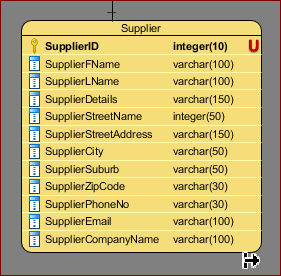
EXECUTE TransactionI 2120302;

## Choose file organizations

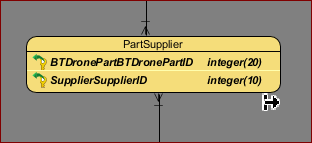
There are several types of file organizations method can be used for a database system, but for for Being There database system, I will be using Cluster File Organization. In a cluster file organization, we can store two or more related tables within the same clusters. These files will be consisting of two or more tables in the same data table and the Primary keys of those tables are used to map those together at once. As such, we have two tables:

BTDronePart and Supplier



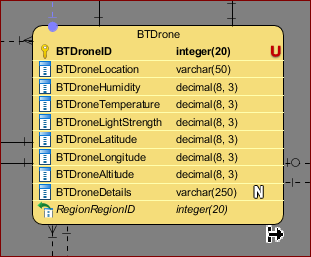
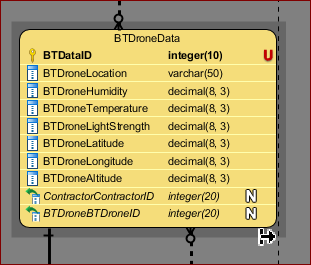


These two tables are related to each other, according to the Being There database. Therefore, we these table are combined together with the help of a join table.



So, now if we want to update, insert, or delete any record that is associated with these tables, then we can directly do that with the help of join table. The data is sorted accordingly for these tables with the help of Primary key. And this primary key can be used to do the searching. I also used cluster key in the Procedure, assigned to this milestone.

However, we can say there are two types of cluster key that we can use while organizing the file. These include:

* **Indexed Clusters:** Indexed clustering means the information of Being There are organized based on the cluster key and then they are stored together. As such the table mentioned below, BTDrone and BTDroneData is an example of index clustering where the records are based on the relationships between these two tables. The record of these two tables are based on BTDroneID.
  1.  
* **Hash Clusters:**  Hash Clustering is quite similar to index clustering, however instead of developing the clustering key, we are creating a hash key value and then store the records using the same hash key value (Semwal, n.d.). I will not be working on Hash Clustering in my Being There database, as I am not familiar with this.

## Choose Secondary Indexes

A secondary index is sometimes referred as a special table that can be used to store a subset of specific field from a certain table. The purpose of Secondary index is to provide efficient access to a wide range of queries in a database, so that we can obtain different query data through different fields rather than depending on the Primary key of a certain table. As such for my Being There database, I have a supplier table where all the supplier details are stored. If a user wants to find a Supplier, it would be difficult to find the supplier with the primary key. However, if I index the Supplier table by supplier first name or last name, it would be very easy for any user to find the information about the supplier. That’s why we use secondary indexes to increase the response time of the queries.

Again, if we don’t use indexes, the main problem we encounter is that we limit the interaction of the database. However, if we use indexes, we reduce the limitation of interaction and this in turn will reduce the number of documents read by the client no matter how big the database is. Thus, we can consider having higher productivity and higher performance (MapR Data Technologies, Inc., 201).

I used indexes in Supplier to use the Supplier first name as Index. If a user wants to find a supplier, then using indexing would be difficult but if we use the supplier name to find the supplier it will be easier. That’s why we are using index on Supplier First Name.

The index code for the Supplier name is:

--Indexing the Supplier First Name

CREATE INDEX idx\_SupplierFirstName

ON Supplier (SupplierFName);

Next, I used Indexing on SalesPeron. If a user wants to find information associated with the shop then they can use ShopName by indexing it.

The index code for ShopName is:

--Indexing the Shop Name

CREATE INDEX idx\_ShopNameSalesPerson

ON SalesPerson (ShopName);

Next, I used indexing in Account to use the account first name as Index. If a user again wants to find an account, he or she can limit the search by using the Account First Name. That’s why I will be performing secondary index on Account First Name

--Indexing the Account First Name

CREATE INDEX idx\_AccountFirstName

ON Acocunt (AccountFName);

## Consider the Introduction of Controlled Redundancy

Data is redundancy is a type of condition that is created in a database where the same data is held in two places. As such we have the information about Latitude, Longitude, Altitude, Temperature, Humidity, Location, and LightStrength in both the BTDrone and BTDroneData. We don’t want to conflict between these two tables. However, I have designed the Being There database is such a way that the data from the BTDrone doesn’t effect the data on the BTDroneData. That’s why I have used Unique identifying number to separate both tables.

## Estimated Disk Space requirements

### Each Tables of Being There Database

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Relation Name** | **Attribute** | **Data type** | **Length** | **Total Per Row** |
| **StaffMember** | StaffID | Integer | 20 |  |
| StaffJobTitle | Varchar | 50 |  |
| StaffDetails | Varchar | 150 |  |
| StaffFName | Varchar | 100 |  |
| StaffLName | Varchar | 100 |  |
| StaffPhoneNumber | Varchar | 50 |  |
| StaffEmailAddress | Varchar | 100 |  |
| StaffSuburb | Varchar | 50 |  |
| StaffCity | Varchar | 50 |  |
| StaffZipCode | Varchar | 20 |  |
| StaffStreetName | Varchar | 50 |  |
| StaffStreetAddress | Varchar | 50 |  |
|  |  |  |  | **790** |
| **Account** | AccountID | Integer | 20 |  |
| AccountFName | Varchar | 100 |  |
| AccountLName | Varchar | 100 |  |
| AccountDetails | Varchar | 250 |  |
| AccountStreetName | Varchar | 50 |  |
| AccountStreetAddress | Varchar | 100 |  |
| AccountCity | Varchar | 50 |  |
| AccountSuburb | Varchar | 50 |  |
| AccountZipCode | Varchar | 20 |  |
| AccountEmail | Varchar | 100 |  |
| AccountPhone | Varchar | 50 |  |
|  |  |  |  | **890** |
| **Subscriber** | SubscriberID | Integer | 20 |  |
| SubscriberPaymentDetails | Varchar | 150 |  |
| SubscriberPassword | Varchar | 50 |  |
| SubscriberDiscount | Decimal | 8 |  |
|  |  |  |  | **228** |
| **Contractor** | ContractorID | Integer | 20 |  |
| ContractorOrganizationName | Varchar | 250 |  |
|  |  |  |  | **270** |
| **AdministrativeExecutive** | AdminID | Integer | 20 |  |
| AdminSubscriptionPrice | Varchar | 50 |  |
| AdminPermissions | Varchar | 20 |  |
| AdminJobTitle | Varchar | 50 |  |
| AdminJobdDescription | Varchar | 250 |  |
|  |  |  |  | **390** |
| **SalesPerson** | SalesPersonID | Integer | 20 |  |
| ShopName | Varchar | 50 |  |
| SalesPersonDetails | Varchar | 250 |  |
|  |  |  |  | **320** |
| **Technician** | TechnicianID | Integer | 20 |  |
| TechnicianJobTitle | Varchar | 50 |  |
| TechnicianWorkDetails | Varchar | 250 |  |
|  |  |  |  | **320** |
| **BTWebsite** | BTSalesID | Integer | 20 |  |
| BTSalesDetails | Varchar | 100 |  |
|  |  |  |  | **120** |
| **Subscription** | SubscriberID | Integer | 20 |  |
| SubscriptionType | Varchar | 50 |  |
| SubscriptionDetails | Varchar | 250 |  |
| SubscriptionPrice | Varchar | 50 |  |
| SubscriptionStartDate | Date | 8 |  |
| SubscriptionEndDate | Date | 8 |  |
| BTSalesID | Integer | 20 |  |
|  |  |  |  | **406** |
| **Standard** | StandardSubscriberID | Integer | 20 |  |
|  |  |  |  | **20** |
| **Gold** | GoldSubscriberID | Integer | 20 |  |
|  |  |  |  | **20** |
| **Contract** | ContractID | Integer | 20 |  |
| ContractType | Varchar | 50 |  |
| ContractDetails | Varchar | 250 |  |
| ContractStartDate | Date | 8 |  |
| ContractEndDate | Date | 8 |  |
| AdminID | Integer | 20 |  |
| ContractorID | Integer | 20 |  |
|  |  |  |  | **376** |
| **Terrain** | TerrainID | Integer | 20 |  |
| TerrainName | Varchar | 50 |  |
| TerrainType | Varchar | 50 |  |
|  |  |  |  | **120** |
| **Region** | RegionID | Integer | 20 |  |
| RegionType | Varchar | 50 |  |
| RegionDescription | Varchar | 250 |  |
|  |  |  |  | **320** |
| **BTDrone** | BTDroneID | Integer | 20 |  |
| BTDroneLocation | Varchar | 50 |  |
| BTDroneHumidity | Decimal | 8 |  |
| BTDroneTemperature | Decimal | 8 |  |
| BTDroneLightStrength | Decimal | 8 |  |
| BTDroneLatitude | Decimal | 8 |  |
| BTDroneLongitude | Decimal | 8 |  |
| BTDroneAltitude | Decimal | 8 |  |
| BTDroneDetails | Varchar | 250 |  |
| RegionID | Integer | 20 |  |
|  |  |  |  | **388** |
| **Platinum** | PlatinumSubscriberID | Integer | 20 |  |
| BTDroneID | Integer | 20 |  |
|  |  |  |  | **40** |
| **SuperPlatinum** | SuperPlatinumSubscriberID | Integer | 20 |  |
| ContractID | Integer | 20 |  |
| BTDroneID | Integer | 20 |  |
|  |  |  |  | **60** |
| **RoamingZone** | RoamingZoneID | Integer | 20 |  |
| RZDetails | Varchar | 250 |  |
| BTDroneID | Integer | 20 |  |
|  |  |  |  | **290** |
| **Altitude** | AltitudeID | Integer | 20 |  |
| MaxAltitude | Decimal | 8 |  |
| MinAltitude | Decimal | 8 |  |
| RoamingZoneID | Integer | 20 |  |
|  |  |  |  | **56** |
| **ZoneBoundary** | ZoneBoundaryID | Integer | 20 |  |
| LatitudeCalculated | Decimal | 8 |  |
| LongitudeCalculated | Decimal | 8 |  |
| RoamingZoneID | Integer | 20 |  |
|  |  |  |  | **56** |
| **Maintenance** | DateMaintained | Date | 8 |  |
| TechnicianID | Integer | 20 |  |
| BTDroneID | Integer | 20 |  |
|  |  |  |  | **48** |
| **MaintenancePart** | DateMaintained | Date | 8 |  |
| MaintenacePartDetail | Varchar | 250 |  |
|  |  |  |  | **258** |
| **BTDronePart** | BTDronePartID | Integer | 20 |  |
| BTDronePartName | Varchar | 150 |  |
| BTDronePartWeight | Decimal | 8 |  |
| BTDronePartSize | Decimal | 8 |  |
| BTDronePartDetail | Varchar | 250 |  |
| DateMaintained | Date | 8 |  |
|  |  |  |  | **444** |
| **Supplier** | SupplierID | Integer | 20 |  |
| SupplierFName | Varchar | 100 |  |
| SupplierLName | Varchar | 100 |  |
| SupplierDetails | Varchar | 150 |  |
| SupplierStreetName | Varchar | 50 |  |
| SupplierStreetAddress | Varchar | 150 |  |
| SupplierCity | Varchar | 50 |  |
| SupplierSuburb | Varchar | 50 |  |
| SupplierZipCode | Varchar | 30 |  |
| SupplierPhoneNo | Varchar | 30 |  |
| SupplierEmail | Varchar | 100 |  |
| SupplierCompanyName | Varchar | 100 |  |
|  |  |  |  | **930** |
| **VideoStream** | VideoID | Integer | 20 |  |
| VideoLength | Varchar | 30 |  |
| VideoDetails | Varchar | 250 |  |
| VideoQuality | Varchar | 30 |  |
| VideoTime | Time | 8 |  |
| BTSalesID | Integer | 20 |  |
| StandardSubscriberID | Integer | 20 |  |
| GoldSubscriberID | Integer | 20 |  |
|  |  |  |  | **398** |
| **BTDroneData** | BTDataID | Integer | 20 |  |
| BTDroneLocation | Varchar | 50 |  |
| BTDroneHumidity | Decimal | 8 |  |
| BTDroneTemperature | Decimal | 8 |  |
| BTDroneLightStrength | Decimal | 8 |  |
| BTDroneLatitude | Decimal | 8 |  |
| BTDroneLongitude | Decimal | 8 |  |
| BTDroneAltitude | Decimal | 8 |  |
| ContractorID | Integer | 20 |  |
| BTDroneID | Integer | 20 |  |
|  |  |  |  | **158** |
| **SubscriptionSales** | SalesPersonID | Integer | 20 |  |
| SubscriberID | Integer | 20 |  |
|  |  |  |  | **40** |
| **BTDroneTerrain** | TerrainID | Integer | 20 |  |
| BTDroneID | Integer | 20 |  |
|  |  |  |  | **40** |
| **BTDroneSubscription** | BTDroneID |  |  |  |
| SubscriberID |  |  |  |
|  |  |  |  | **40** |
| **BTDroneVideoStream** | VideoID | Integer | 20 |  |
| BTDroneID | Integer | 20 |  |
|  |  |  |  | **40** |
| **PartSupplier** | BTDronePartID | Integer | 20 |  |
| SupplierID | Integer | 20 |  |
|  |  |  |  | **40** |
| **BTDroneContract** | ContractID | Integer | 20 |  |
| BTDroneID | Integer | 20 |  |
|  |  |  |  | **40** |
| **ContractBTDrone** | BTDataID | Integer | 20 |  |
| ContractID | Integer | 20 |  |
|  |  |  |  | **40** |
| **ContractRoamingZone** | ContractID | Integer | 20 |  |
| RoamingZoneID | Integer | 20 |  |
|  |  |  |  | **40** |
| **SubscriptionAdmin** | SubscriberID | Integer | 20 |  |
| AdminID | Integer | 20 |  |
|  |  |  |  | **40** |
| **VideoSubscription** | VideoID | Integer | 20 |  |
| SubscriberID | Integer | 20 |  |
|  |  |  |  | **40** |

### Total Being There Database Size

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Relation Name** | Total Per Row | Max Size | Total Relation Size | Pages Per Relation  (Total Size/8192) | New Relation Size  (Pages \* 8192) |
| **StaffMember** | 790 | 200 | 158,000 | 20.00 | **163,840** |
| **Account** | 890 | 500,100 | 445,089,000 | 54,333.00 | **445,095,936** |
| **Subscriber** | 228 | 500,000 | 114,000,000 | 13,916.00 | **113,999,872** |
| **Contractor** | 270 | 100 | 27,000 | 4.00 | **32,768** |
| **AdministrativeExecutive** | 390 | 20 | 7,800 | 1.00 | **8,192** |
| **SalesPerson** | 320 | 100 | 32,000 | 4.00 | **32,768** |
| **Technician** | 320 | 80 | 25,600 | 4.00 | **32,768** |
| **BTWebsite** | 120 | 1 | 120 | 1 | **8,192** |
| **Subscription** | 406 | 500,000 | 203,000,000 | 24,781.00 | **203,005,952** |
| **Standard** | 20 | 492,400 | 9,848,000 | 1,203.00 | **9,854,976** |
| **Gold** | 20 | 5,000 | 100,000 | 13.00 | **106,496** |
| **Contract** | 376 | 2,000 | 752,000 | 92.00 | **786,432** |
| **Terrain** | 120 | 200 | 24,000 | 3.00 | **24,576** |
| **Region** | 320 | 100 | 32,000 | 4.00 | **32,768** |
| **BTDrone** | 388 | 50,000 | 19,400,000 | 2,369.00 | **19,406,848** |
| **Platinum** | 40 | 2,500 | 100,000 | 13.00 | **106,496** |
| **SuperPlatinum** | 60 | 100 | 6,000 | 1.00 | **8,192** |
| **RoamingZone** | 290 | 50,000 | 14,500,000 | 1771.00 | **14,508,032** |
| **Altitude** | 56 | 50,000 | 280,000 | 35.00 | **286,720** |
| **ZoneBoundary** | 56 | 1,000,000 | 56,000,000 | 6,836.00 | **56,000,512** |
| **Maintenance** | 48 | 10,000 | 480,000 | 59.00 | **483,328** |
| **MaintenancePart** | 258 | 10,000 | 2,580,000 | 315.00 | **2,580,480** |
| **BTDronePart** | 444 | 10,000 | 4,440,000 | 542.00 | **4,440,064** |
| **Supplier** | 930 | 100 | 93,000 | 12.00 | **98,304** |
| **VideoStream** | 398 | 100 | 39,800 | 5.00 | **40,960** |
| **BTDroneData** | 158 | 50,000 | 7,900,000 | 965.00 | **7,905,280** |
| **BTDroneTerrain** | 40 | 50,000 | 2,000,000 | 245.00 | **2,007,040** |
| **BTDroneSubscription** | 40 | 50,000 | 2,000,000 | 245.00 | **2,007,040** |
| **SubscriptionSales** | 40 | 50,000 | 2,000,000 | 245.00 | **2,007,040** |
| **BTDroneVideoStream** | 40 | 50,000 | 2,000,000 | 245.00 | **2,007,040** |
| **PartSupplier** | 40 | 50,000 | 2,000,000 | 245.00 | **2,007,040** |
| **BTDroneContract** | 40 | 500,000 | 20,000,000 | 2,442.00 | **20,004,864** |
| **ContractData** | 40 | 1,000,000 | 40,000,000 | 4883.00 | **40,001,536** |
| **ContractRoamingZone** | 40 | 500,000 | 20,000,000 | 2442.00 | **20,040,864** |
| **SubscriptionAdmin** | 40 | 500,000 | 20,000,000 | 2442.00 | **20,040,864** |
| **VideoSubscription** | 40 | 200,000 | 8,000,000 | 977.00 | **8,003,584** |
| **Total Database Size (Bytes)** | | | | | **997,177,664.00** |
| **Total Database Size (Kilobytes)** | | | | | **973,817.00** |
| **Total Database Size (Megabytes)** | | | | | **951.00** |
| **Total Database Size (Gigabytes)** | | | | | **1.00** |

## Design Security Mechanisms

When we are working with our database we need to make sure the data we created are safe and sound. The security of the database is very important as the database that is built for a certain organization like Being There, is very important as an organization needs to have access to the data almost every day. Having security also helps an organization to counter the security threats. The SQL server provides us with security mechanism so that the database administrators and the developers can create and oversees the database system in a secure working environment. Every SQL server provides an organization like Being There, with permissions that can be assigned to a principal, that is assigned to an individual, group, or a process. The SQL server provides an organization with security with the help of authentication and authorization.

* **Authentication** – It is the process we can log in to our SQL Server with the help of principal requests. In the authentication process, the SQL server identify the user so that they can authenticated.
* **Authorization** – It is the process where the SQL server determines if the securable resources a principal can access, and what operation can be run for those resources (Microsoft, Overview of SQL Server Security, n.d.).

## Monitor and Tune the Operational System

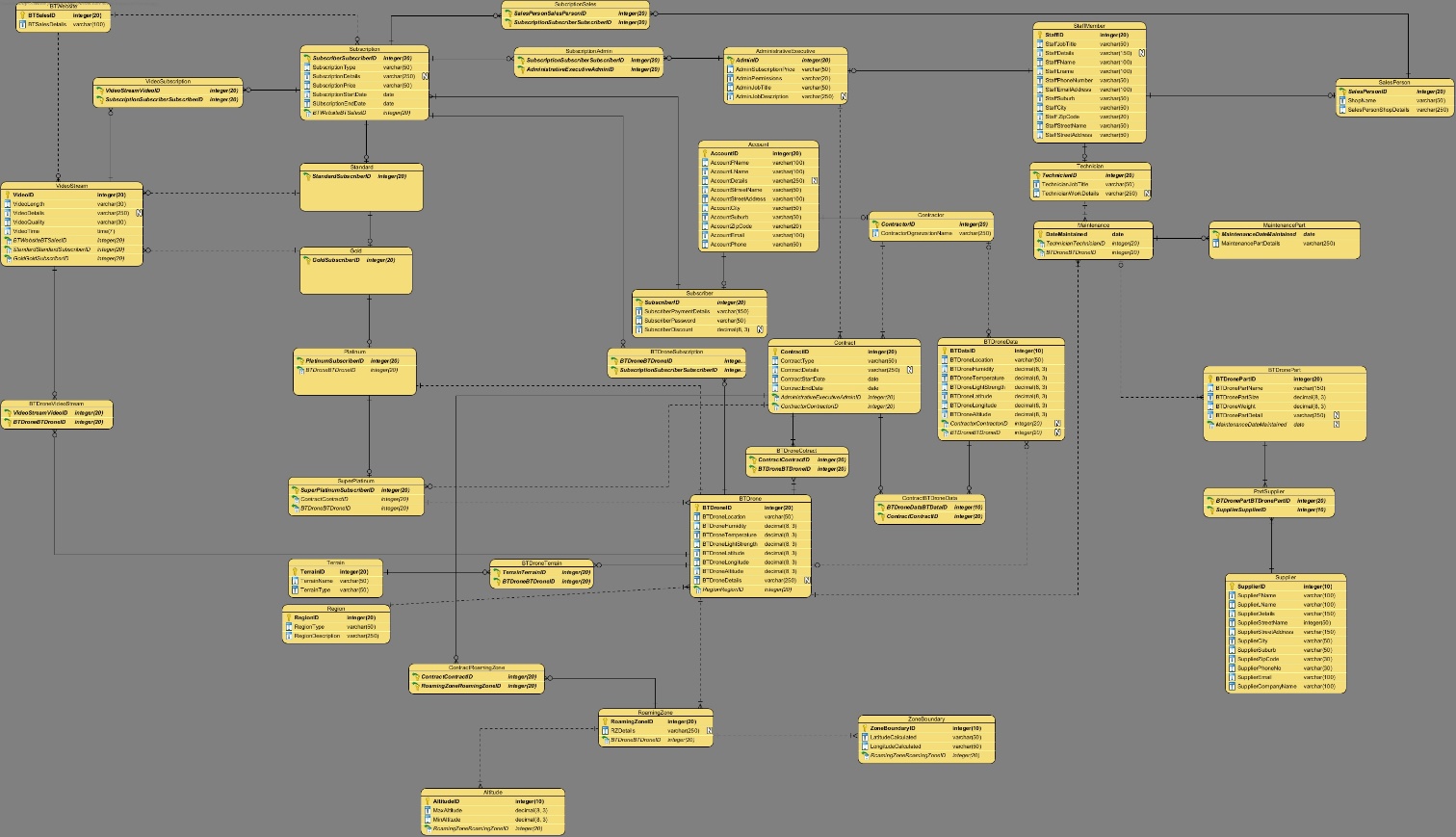
The purpose of monitoring and tuning the database is to improve the performance of a certain system. Monitoring the SQL server can hep us t o determine if we can improve the performance of a system. As such, if we have queries that are frequently used, then we need to make sure if we really need to change the indexes or query of a certain table.

We can also use monitoring and tuning to evaluate the user activity. This will help us to find out whether the security of a certain database is setup adequately or not. We can also troubleshoot and debug problems that may be associated with the components of the application. We can also use several types of monitoring and tuning tools in SQL server like, Live Query Statistics and Engine Tuning Advisor for our Being There database (Microsoft, Monitor and Tune for Performance, n.d.).

# Physical Design for Being There database

The link to database design:

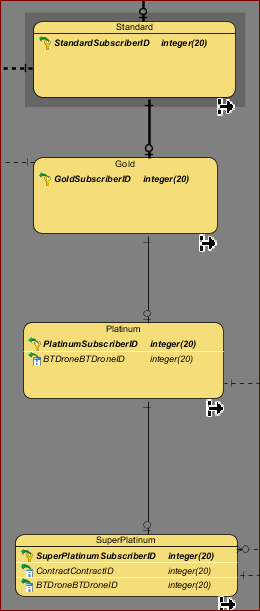
<https://livenmitac-my.sharepoint.com/:i:/g/personal/ashkar-milton_live_nmit_ac_nz/EQRnGfKyx6BKiH9bUQq2rO8BpyE8Zd60aZezA7KqCq2bYg?e=4PWvgZ>



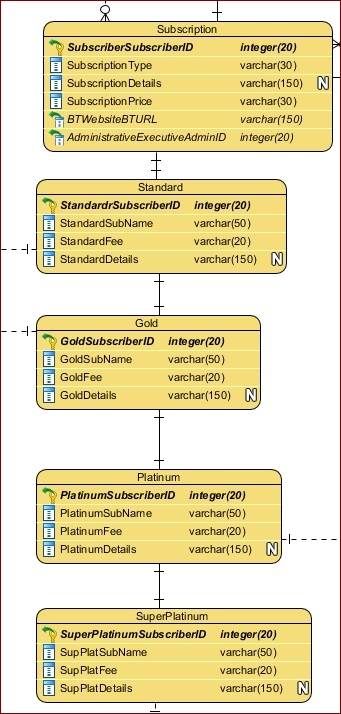
# Issues and Resolutions

While working on the Being There, I had worked on the conceptual and logical model of the diagram. But, after having feedback from my milestone 1 and 2, I have changed certain stuffs for the Physical Design of Being There database. The issues and resolutions include:

* I have changed the Physical model of Being There database. According to the design of my logical model. According to my logical model, I had a join table between the SalesPerson and the BTDrone. However, in my updated logical model, I didn’t include that. The reason is the SalesPerson is connected to the BTDrone through subscription. They don’t have relationship with the BTDrone directly.
* In the maintenance table, I have used DateMaintained as the primary key rather than the TechnicianID for the Logical model. The reason I choose to change this is because, if we want to find out about Maintenance, we need to go by the date to find out of the maintenance. These maintenances can be done by several Technician. However, the most important thing to know about the date a BTDrone has been maintained.
* I have created join table BTDroneContract in the physical model, because many BTDrones are connected to many Contracts
* I have created join table ContractBTDroneData in the physical model, because many Contracts gets many BTDroneData
* I have created join table ContractRoamingZone in the physical model, because many Contract have many relationships with Roaming Zone.
* I have created join table BTDroneSubscription in the physical model, because many Subscription are associated with many BTDrones.
* I have created join table SubscriptionAdmin in the physical model, because many Administrators have controlled over many Subscriptions.
* I have created join table VideoSubscription in the physical model, because many Subscriptions have access to many VideoStream.
* I have created a join table called BTDroneTerrain as many BTDrone have many Terrains.
* I have removed the Attributes related with Standard, Gold, Platinum, and SuperPlatinum tables in the Physical model. The reason for doing this is because I am already getting the information about the subscriber from the Account table, so I don’t need to put repeated information in the Standard, Gold, Platinum, and SuperPlatinum tables.



Whereas in the logical model I have included with several attributes:



* I have added ZoneBoundary as a new table, as I need to find the latitude and longitude for each ZoneBoundary. I have used ZoneBoundary table in Procedure G.
* I have assigned a new Primary key on BTDrone Data which is BTDataID, by removing the BTDroneID as Primary key. This is to prevent database redundancy of Being There. Having to do this allows new information of BTDrone to be updated on the BTDroneData table.
* I have removed the Primary key of AdminID from the Contract table as we need only ContractID as a unique identifier to identify each contract. Thus, the AdminID becomes the foreign key of Contract.
* I added BTDroneDetails as an attribute in the BTDrone table, as if I choose to put some additional information regarding BTDrone, I can use this row. It is Null.
* I have connected the Platinum table with the BTDrone as each PlatinumSubscriber is connected to one BTDrone, thus I have a foreign key of BTDroneID in the Platinum table.
* I have connected the SuperPlatinum table with the Contract as each Contract have control over one or more SuperPlatinumSubscriber. The updated SuperPlatinum as foreign key constraints which include the ContractID and BTDroneID.
* I have added new row SubscriptionStartDate and SubscriptionEndDate on the Subscription table, as I need to know when a subscriber is going to start his or her subscription and when the subscription is going to end.
* I have changed all the Primary Key Id length to integer – 20, except DateMaintained which is in Date.
* I have changed the length and data types for certain attributes in my physical model, and these includes:
  + ShopName - Varchar (50) instead of varchar (30)
  + DateMaintained – Date instead of BTDroneID (Integer0
  + StaffZipCode - varchar (20) instead of varchar (30)
  + TechnicianJobTitle – Varchar (50) instead of varchar (30)
  + SalesPersonJobTitle – varchar (250) instead of varchar (150)
  + MaintenancePartDetail – Varchar (250) instead of varchar (30)
  + BTDronePartSize – Decimal (8,3) instead of Decimal (19,0)
  + BTDronePartWeight – Decimal (8,3) instead of Decimal (19,0)
  + BTDronePartDetail – varchar (250) instead of varchar (150)
  + SupplierStreetAddress – Varchar (250) instead of SupplierStreetNo
  + SupplierSuburb – Varchar (50) instead of varchar (30)
  + SupplierZipCode – Varchar (50) instead of varchar (20)
  + SupplierCity – Varchar (50) instead of varchar (20)
  + TechnicianJobTitle – Varchar (50) instead of varchar (30)
  + TechnicianWorkDetails – Varchar (250) instead of varchar (150)
  + StaffFName – varchar (100) instead of varchar (30)
  + StaffLName – varchar (100) instead of varchar (30)
  + StaffZipCode – Varchar (20) instead of varchar (30)
  + ContractType – Varchar (50) instead of varchar (30)
  + BTDroneLocation – Varchar (50) instead of varchar (30)
  + BTDroneHumidity – Decimal (8,3) instead of Decimal (19,0))
  + BTDroneLightStrength – Decimal (8,3) instead of Decimal (19,0)
  + BTDroneLatitude – Decimal (8,3) instead of Decimal (19,0)
  + BTDroneLongitude – Decimal (8,3) instead of Decimal (19,0)
  + BTDroneAltitude – Decimal (8,3) instead of Decimal (19,0)
  + SubscriberDiscount – Decimal (8,3) instead of Decimal (19,0)
  + VideoDetails – Varchar (250) instead of integer (10)
  + VideoQuality – varchar (30) instead of varchar (15)
  + TerrainName – varchar (50) instead of varchar (30)
  + TerrainType – Varchar (50) instead of varchar (30)
  + RegionDescription – Varchar (250) instead of varchar (150)
  + ContractorOrganizationName – Varchar (250) instead of varchar (150)
* I have added the connection between drones and subscription, as only the platinum and super platinum subscribers has connection with the particular drones. So, we can find which drones has connection with the subscription according to their subscriber id added. These subscribers add their information on the account.
* While I was doing procedure H and I wasn’t familiar with cascade until I understood what it means. I was trying to delete the data for a given contract, but I was prevented to do. As the ContractData is connected. So, I must put cascade on the table to make it work.
* While I was working on the Physical Model of Being There, there are many times I have to update the table. I have to drop the database and create the database again.
* When I was working on the procedures, I have to create certain join tables like BTDroneContract and ContractBTDroneData, to make sure the information I needed to create the Procedures is accurate.
* At the last moment I realized that the Subscriber can get discount no more than 3%, so I have to go back to data table to fix it.
* At the end I heard that we have to create a function table to limit the number of VideoStream to 100. So, I have to spend few more minutes to build the function table on Check Constraints for VideoStream table.
* One more issue I faced was calculating the data storage. I used 8Kb = (8\*1024) = 8192 bucket per page. So, instead of rounding off to 8,000 I used 8192 bucket per page.
* The biggest issue right now I could think of is to plot the Transactional Analysis diagram on VISIO.

# Conclusion from Physical Design

I can say at the end that I have had a quite a great journey working in DAT-601. I really enjoyed working with SQL server and I got to do it in the Milestone 3. I was really intrigued by the way I have to build different tables for Being There database, and then I have to insert data in each table. It may be a difficult job, but at-least I can say I really learned how to insert data in the table.

From Physical Design, I learned how to create procedures, as this taught me how to Insert, Read, Update, or Delete data from the table. I was able to go through different transactional analysis because of these procedures, as plotting transactional analysis diagram taught me to how Insert, Read, Update, or Delete to a table or from a table.

Additionally, I also learned about different constraints that I have used in my database. I also used secondary indexes in my database so that any user can retrieve the data if needed other than using the primary key. I also learned about different types of file organizations, and later on decided to use clustered index.

Furthermore, I learned about security mechanism and how it is important for an organization creating different databases. It gave me a brief idea how problematic it would for a database organization, if not have proper security mechanisms. At last I would say, I learned how to work disk management for database.

In conclusion, I will say that it has been quite a journey and I am optimistic that what I learned from Physical model and Logical and Conceptual model, can be used in my later courses like DAT-601 and DAT-701. So, I look forward to work more on database.

# Bibliography

IBM. (n.d.). *Types of constraints*. Retrieved from DB2 Version 10.1 for Linux, UNIX, and Windows: https://www.ibm.com/support/knowledgecenter/en/SSEPGG\_10.1.0/com.ibm.db2.luw.admin.dbobj.doc/doc/c0020149.html

MapR Data Technologies, Inc. (201). *Secondary Indexes*. Retrieved from MapR Data Technologies: https://mapr.com/docs/61/MapR-DB/Indexes/Indexes.html

Microsoft. (n.d.). *Monitor and Tune for Performance*. Retrieved from SQL: https://docs.microsoft.com/en-us/sql/relational-databases/performance/monitor-and-tune-for-performance?view=sql-server-2017

Microsoft. (n.d.). *Overview of SQL Server Security*. Retrieved from SQL Server and ADO.NET: https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/sql/overview-of-sql-server-security

Oracle. (1999). Overview of Physical Design. *Oracle8i Data Warehousing Guide*, A76994-01.

Paradigm, V. (2018). *Conceptual, Logical and Physical Data Model*. Retrieved from Visual Paradigm User's Guide: https://www.visual-paradigm.com/support/documents/vpuserguide/3563/3564/85378\_conceptual,l.html

Semwal, S. D. (n.d.). *DBMS | File Organization – Set 3*. Retrieved from Geeks for Geeks: https://www.geeksforgeeks.org/dbms-file-organization-set-3/

Tupper, C. D. (2011). Physical Database Considerations. *Data Architecture,*.

W3Schools. (2019). *SQL CHECK Constraint*. Retrieved from W3 Schools: https://www.w3schools.com/Sql/sql\_check.asp