HARPRIT SINGH

Conceptual Modelling

Assignment 1

Milestone 1

Contents

PAGES

1. CONCEPTUAL MODELING 3
2. CHEN ERD 5
3. PROJECT BRIEF ERD 14
4. DATA DICTONARY 16
5. ASSSUMED BUSINESS RULE 23

Conceptual Modeling

HISTORY OF CONCEPTUAL modeling

Three decades are represented in the conceptual modelling: the 1970s, 1980s, and 1990s.

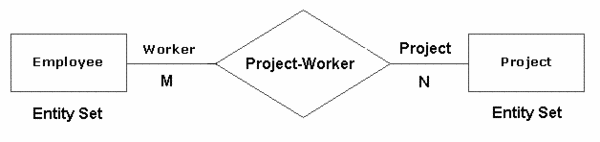


In the 1970s, database architecture was critical.

In the field of data modelling and database architecture, Peter Chen's paper "The Entity-Relationship Model: Toward a Unified View of Data" is a landmark.

Smith and Smith incorporate abstraction and generalization of database architecture.  
There have also been efforts to create high-level data definition languages for identifying and describing mathematical schemas.

The Conceptual Schema Language is one such language (CSL). Information structures and their semantics are becoming more fascinating.



1980

Several approaches to extending Chen's Entity Relationship Model were established in the 1980s.We also look at the 1980 Pingree Park Workshop. Roland Colette's REMORA is another significant work from this decade. Really, information systems and their architecture are fascinating subjects.

1990

In the early 1990s, there were many issues in the field of database architecture, such as schema convergence, schema adaptation, and consistency controls for logical schemas. Object-oriented simulation approaches and languages in software engineering, however, have an impact on this period. Object-oriented programming languages became more popular in the 1980s.

In software engineering, data modelling is becoming increasingly relevant.

At the start of the 1990s, there were more than fifty different modelling languages available.

Booch, Jacobson, and Rumbaugh worked together to create the Unified Modeling Language (UML), which became a staple in object-oriented modelling in 1997.

According, designer wants their user to understand their models to make it perfectly they uses conceptual modelling. By using the software and documentation the users draw a rough diagram and gave it to the designer.

However, Conceptual models define structure models in terms of entities, relationships, and constraints, and behaviors or functional models in terms of states, transitions between states, and actions performed in states and transitions. Conceptual modelling is used to define the semantics of software applications at a high level of abstraction.

Structure models are defined in terms of entities, relationships, and constraints by conceptual models, while behaviors or functional models are defined in terms of states, transitions between states, and acts performed in states and transitions by behavioral models.

The conceptual data model is a business rules view of the data needed to support business processes, monitor relevant performance metrics, and document business events.

The emphasis of this model is on defining the data used in the sector, rather than the processing flow or physical characteristics. The viewpoint of this model is unaffected by any underlying business applications. It allows businesspeople to see sales data, cost data, clients, and goods, for example—all of which are business subjects in the integrated model but not in the applications themselves.

When creating a design model for a future software programme, one aim is to keep it as straightforward as possible. The fewer definitions that users must learn, the easier, if the requisite functionality is included.

Mind the following while developing computer applications, as well as many other things:

Mies van der Rohe once said, "Less is better."

Chen ERD Diagram:

The entity–relationship model (ER model) is a type of entity–relationship model.

This model is made up of object categories that all classify the so-called "things of interest." It represents the interrelated things of interest in a narrow domain of information.

Many of the relationships that may occur between instances of the entity types are defined in the model.

As I told earlier Peter Chen created entity–relationship modelling in 1976 with the aim of using it in database design.

Other ER-modelers insist on displaying subtype entities bound by generalization and specialization relationships, so the ER-model can be used to specify domain-specific concepts as well. This model does not describe certain business processes, but it does provide a graphical representation of a business data schema.

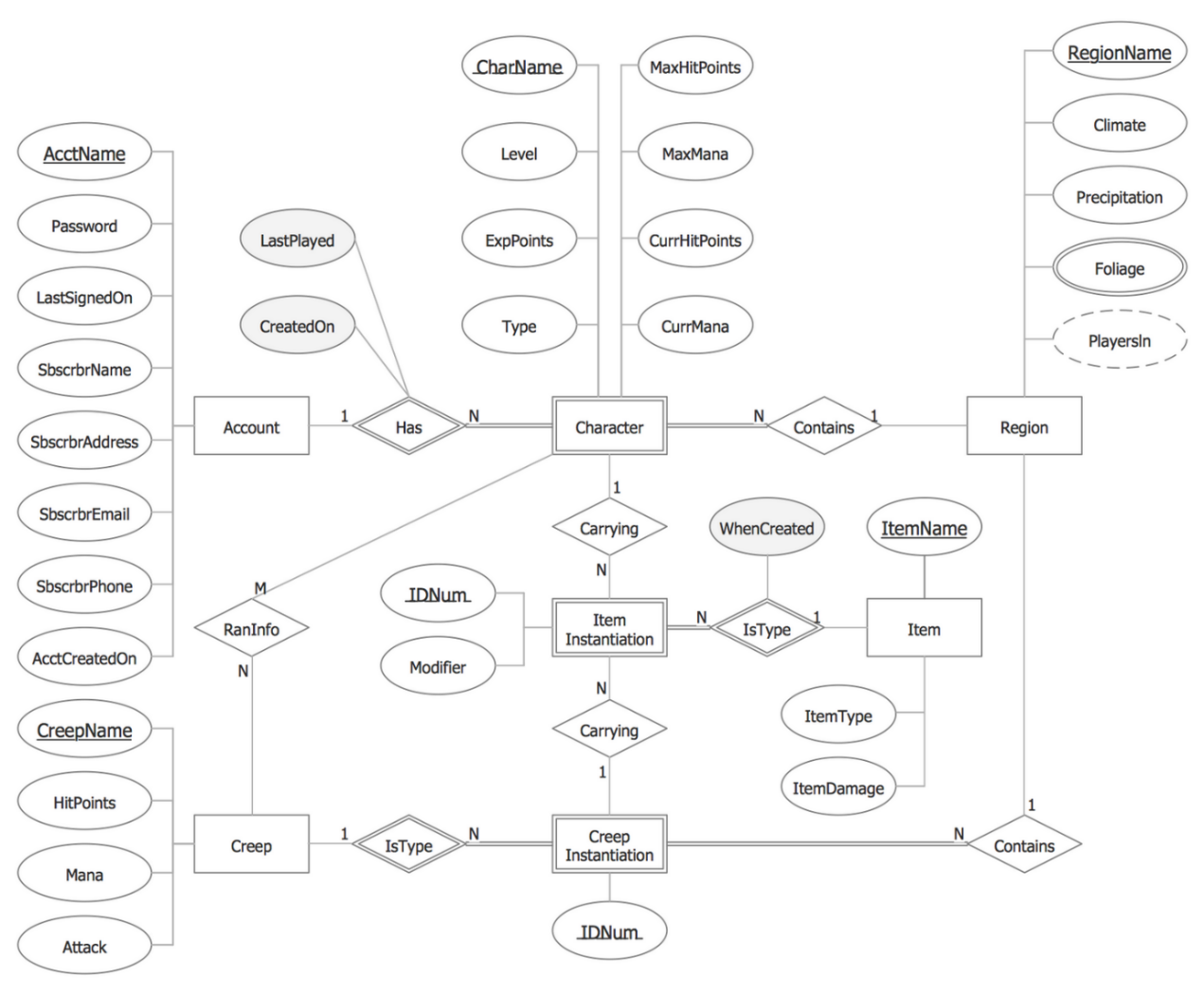
The ER model can be represented graphically in the form of boxes that represent entities and lines that represent relationships, representing interactions and connections between those entities.

"Entity-attribute-relationship diagrams" or "entity-relationship templates" are diagrams generated to display the necessary attributes and entities, as well as any relationships (which is not as common).

There are several different types of data models, the most common of which are "conceptual," "logical," and "real." There is the least grained complexity within the “conceptual” model (which is the highest degree possible of any ER model), which determines the general definition of what must be contained within the model set itself. This type of "conceptual ER model" is intended to describe the master reference data entities that are widely used by any company participating in the process.

This model can also be used as the basis for one or more logical data models, with the aim of establishing "structural metadata" between the various logical ER models. An ER model is implemented by a database. Remember that this ER model can be used to create commonality relationships between related models and serve as a foundation for further data model integration.

EXAMPLE:



Here I will explain about explain more accurate components of ERD

1. Entity:



It is represented by rectangle shape and contains entities name in it.

1. Weak entity

An entity that cannot be defined solely by its attributes. The presence of a weak entity is contingent on the existence of another entity known as the owner entity. The identifier of the weak entity is a mixture of the owner entity's identifier and the weak entity's partial key.



1. Associative entity:

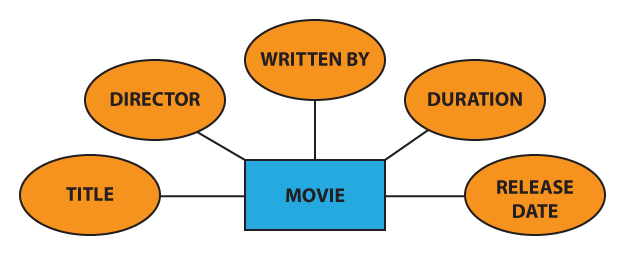
In a many-to-many relationship, an entity will be included (represents an extra table).

Many of the associative entity's relationships should be many.



1. Attributes:

Each attribute is represented by an oval containing atributte’s name: and its connected with straight lines



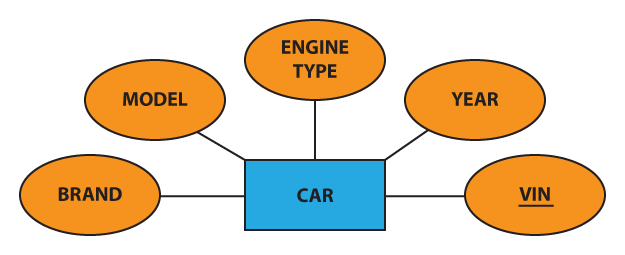
1. Key attribute:

A characteristic that distinguishes a certain person in a special way.

A main attribute's name is underlined and below is the example:



example 2: Since the Vehicle Identification Number (VIN) is a one-of-a-kind code used to classify particular vehicles (no two have the same VIN), “VIN” can be considered a key attribute for the “CAR” entity:



1. PARTIAL KEY attribute:

an attribute that, when paired with the owner entity's main attribute, gives the weak entity a unique identifier. With a dotted thread, we emphasize the discriminator:

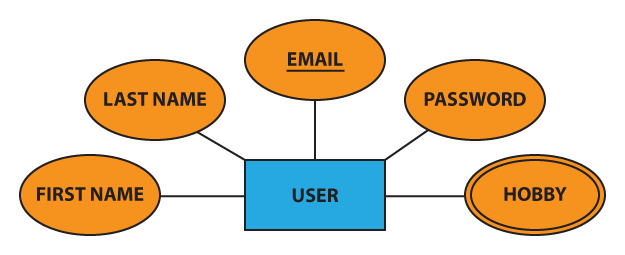


1. Multivalued Attribute:

an attribute with a wide range of possibilities (there are many distinct values entered for it in the same column of the table).A dual oval represents a multivalued attribute:



Example: each person can have several hobbies, the “hobby” attribute for the “user” object may be thought of as a multivalued attribute:



1. Derived Attribute:

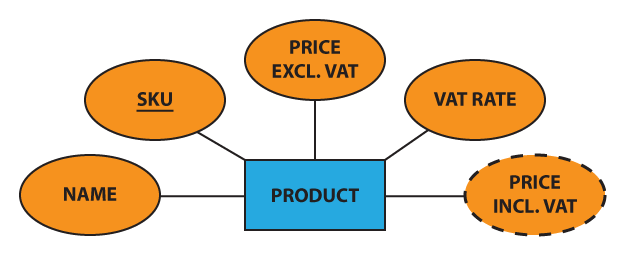
an attribute whose value is obtained (calculated) from the values of other attributes.

The database may or may not physically store the derived attribute.



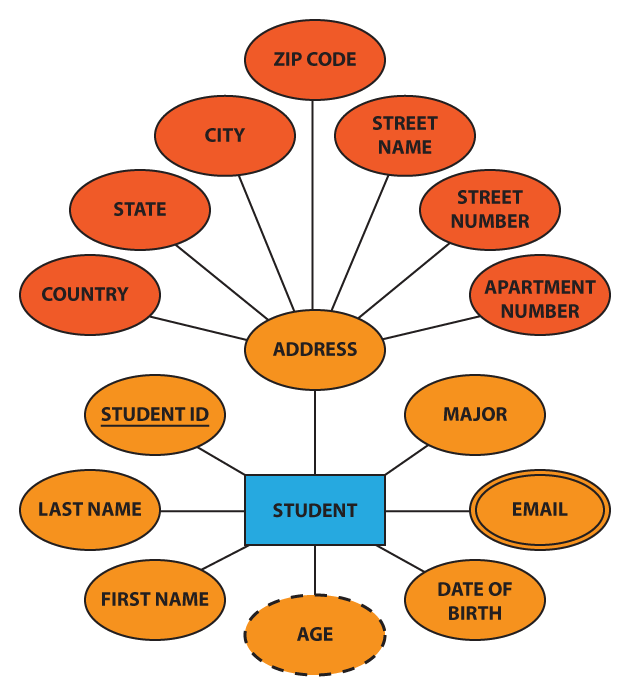
This attribute is represented by a dotted oval in Chen notation.

Example: For example, we can determine the price including VAT if we know the price excluding VAT and the VAT rate:



1. Composite attribute: Some attributes can be further divided into more smaller parts. These smaller parts are known as composite attribute.

Example: Address can be broken down into street name, street number, apartment number, area, state, zip code, and country.



Relationship: A relationship is defined in Chen notation by a diamond (rhombus) bearing the relationship's name and it is divide into two parts which are explain below:

1. Strong relationship: a relationship in which an individual's presence is unaffected by the existence of other entities, and the PK of the Child does not include the PK attribute of the Parent Entity. A single rhombus represents a close relationship:



1. WEAK Relationship: a relationship in which the presence of the Child Entity is contingent on the existence of the Parent Entity, and the PK of the Child Entity comprises the PK attribute of the Parent Entity. A double rhombus represents this relationship:



Optionality of Relationship:

1. Using a solid line a mandatory relationship is represented: Chen ERD notation - mandatory relationship line
2. Using dashed line optional relationship is represented:

Chen ERD notation - optional relationship line

Cardinality: Characters "1," "N," or "M" are normally inserted at the ends of relationships to denote the degree of relationship (cardinality):

1. ONE TO ONE ((1:1)):

Each employee can only oversee one department, and each department can only have one manager.



1. One to many (1:N):

Although a customer can place several orders, each order may only be placed by one customer:



1. Many to one (N:1)

Many workers can be assigned to one department, but each employee may only be assigned to one department.



1. Many to many:

A student may be a member of multiple student organizations, and one group can accept multiple students



Total participation:

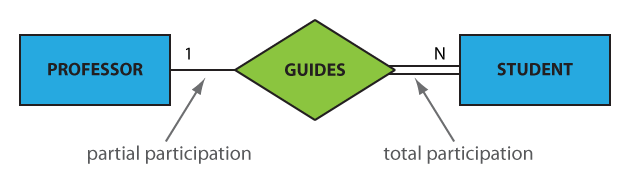
means that each person in the set is interested in the relationship; for example, each pupil must be supervised by a professor (there are no students who are not guided by any professor).

This type of relationship is represented by a double line in Chen notation.

Partial Participation:

means that not every entity in the set is interested in the relationship; for example, not every professor directs a student (some professors do not).

A single line in the Chen notation represents a partial presence.



According to the above relationship, each student must be led by a single professor, and one – but not any – professor will direct several students.

So, there is no student who is not directed by a professor, and there are professors who do not instruct any students on the other side.

ERD:

https://github.com/Harprit-singh/DAT601-conceputal-diagram.git

Pls open this link it contains proper image.

Explaining the diagram:

Flight stream is an organization which provides video streaming and data sensing to government services, commercial organizations, and presence services. To provide their service they use, drones so that the drone can collect, sense data and transmitting too. For doing this all stuff we need more people which we work in a team but in different area. So, I divide the company into four parts which are

1) Staff member

2) Salesperson

3) Admin executive

4) Maintenance

Flight stream provides their contract in form of subscription service and the person who buy the subscription is known as subscriber. So here role of staff person is to handle the contracts which has many subscription services for subscribes and to provide the benefits to different types of subscribers they have.

I further divided it into super platinum, platinum, gold and standard. All of these are related to each with their attributes.

Salesperson: The name itself says the role he does. The role of salesperson is to sale subscriptions to the customers. The role of Admin executive is to maintain records between the data scoop and can change the price of subscription and can enter contract details. Maintenance person plays an important role of maintaining records for each drone data, supplier details, maintenance report. Flight stream drone has website for video steam for their subscriber. Video stream is providing views in 3d video form and for super platinum subscriber they gets full control to it.

Datascoop sense different stuff like: Temperature, Humidity, Ambient light strength and it fly’s around the world and can be configured different regions like Mountain, Snow, Desert and urban deployment and data is stored. The drone shows the live location and the datascoop data stores the data which the drone is sending. Flight stream deploys many drones to many zones

Data Dictionary:

**Conceptual Model Documentation**

**Table 1: Document Entities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity Name** | **Description** | **Aliases** | **Occurrence** |
| Flight stream | Flight stream is the company name |  | Flightstream |
| Employees | Employees of the flight stream with unique id with different working sector | Flight stream |  |
|  |  |  |  |
| Administrative executive | Administrative executive will keep the records of the data and make changes to contract and subscription too | Flight stream |  |
| Data scoop | Data scoop will the shows details of data he senses | Data scoop |  |
| Video stream | Video stream will work with drone to provide video on their website |  |  |
| gold | Just a give weak entity which shows that type subscriber | subscriber |  |
| platinum | Just a give weak entity which shows that type subscriber | subscriber |  |
| Super platinum | Just a give weak entity which shows that type subscriber | subscriber |  |
| standard | Just a give weak entity which shows that type subscriber | subscriber |  |
| Datascoop data | Collect the data from the drone | Datascoop |  |
| Maintenance Person | Maintain the records, drone data, supplier parts . |  |  |
| Staff member | Shows details and handles the contracts of the company. | Staff member |  |
|  |  |  |  |
| Sale person | The person who sells the subscription as well as drones too |  |  |
|  |  |  |  |
| Data scoop parts | Details of data scoop parts | Data scope drone |  |
| supplier | Suppliers’ details who supply them drone parts |  |  |
| Website | A website for everyone | Flight stream |  |
| Subscription | Handling between subscribers and contract |  |  |
| zone | Zone are bounded by geographic regions | Data scoop |  |

**Table 2: Document Relationships**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name** | **Cardinality** | **Relationship** | **Cardinality** | **Entity Name** |
| website | 1 | provides | N | Video stream |
| Video stream | M | Records | 1 | Data scoop |
| Gold | N | Control view | 1 | Video stream |
| Platinum | 1 | Less control | N | Data scoop |
| Super platinum | 1 | Exclusive control | 1 | Data scoop |
| Standard | 1 | View | N | Video stream |
| Subscription | 1 | has | M | Subscriber |
| Subscription | N | Provides | M | Contracts |
| Contracts | N | Handles | M | Staff member |
| Subscription | n | Sells | M | Salesperson |
| Maintenance person | 1 | Maintain records | M | Datascoop drone parts |
| Maintenance person | 1 | Maintain records | N | Datascoop data |
| Flight stream | 1 | has | m | Employees |
| Datascoop | N | Transmit | 1 | Datascoop data |
| Data scoop drone part | M | Supplies | N | Supplier |
| Data scope | n | Bounded by | m | zone |
| Subscription | N | Sales | N | Salespersons |
| contract | N | Enter | 1 | Admin executive |

**Table 3: Document Attributes**

| **Entity Name** | **Attributes** | **Description** | **Domain** | **Aliases** | **Composite** | **Derived** | **Nulls** | **Key?** | **Default Value** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Video time | Duration of time | number |  |  | no | no |  | - |
|  | Video id | Unique number | - | Video stream | no | no | no | CK/PK | - |
|  | Video quality | Quality of video | number |  | no | no | no | - | - |
| **Video stream** | Video details | details | Char string |  |  | no | No |  | - |
|  | GOLD sub Id | Unique id | - |  |  |  |  |  | - |
|  | Gold sub name | Name of gold sub | Char string | Gold subscriber | no | no | No | - | -- |
| **Gold sub** | Gold sub details | Details of gold sub | Char string |  |  | no | no | - | - |
|  | Gold sub fees | Fees of gold sub | number |  |  | no | no | - | - |
|  | Platinum sub id | Unique id | - |  |  |  |  |  | -- |
|  | Platinum sub name | Name of platinum sub | Char string |  |  | no | No | - | - |
| **Platinum sub name** | Platinum sub details | Details of platinum sub | Char string | Platinum sub |  | no | No | - | - |
|  | Platinum sub fees | Fees of Platinum sub | number |  |  | no | No | - | - |
|  | Super platinum id |  |  |  |  |  |  |  | - |
|  | Super platinum name | Super platinum name | Char string |  |  | No | No | - | - |
| **Super platinum** | Super platinum fees | Payment of super platinum | Char string | Super platinum |  | No | no | - | - |
|  | Super platinum  details | Details of super platinum | number |  |  | No | no | - | - |
|  | Standard id | Unique id | - |  |  |  |  |  | - |
| **Standard** | Standard name | Name of standard user | Char string | Standard |  | no | no | - | - |
|  | Standard fees | Standard fees | number |  |  | No | No | - | - |
|  | Standard details | Standard details | Char string |  |  | No | No | - | - |
|  |  |  |  |  |  |  |  |  | - |
|  | Contract id | Unique number | - | Contract |  | No | No | CK/PK | - |
| **Contract** | Contract details | Details of contract | Char string |  |  | No | No | - | - |
|  | Contract type | Type of contract | Char string |  |  | no | No | - | - |
|  |  |  |  |  |  |  |  |  | - |
|  | Datascoop id | Unique number |  |  |  | No | No | - | - |
|  | Altitude | Details of altitude | number |  |  | NO | No | - | -- |
| **Datascoop drone data** | Light strength | Details of light strength | number | Datascoop  drone data |  | No | No | - | - |
|  | temperature | Details of temperature | number |  |  | No | No | - | - |
|  | Humidity | Details of humidity | number |  |  | no | No | - | - |
|  |  |  |  |  |  |  |  |  | - |
|  |  |  |  |  |  |  |  |  | - |
| **Flight stream** | Flight stream id | Unique id | - |  |  | No | no | Pk | - |
|  | Flight stream address | Address of the company | Char string | Flight stream | Street, postal code. | No | no | - | -- |
|  | Flight stream phoneno | Phone no the company | number |  |  | No | No | - | - |
|  |  |  |  |  |  |  |  |  |  |
|  | Admin id | Unique id | - |  |  | no | No | PK | -- |
| **Admin executive** | Admin details | Details of his information | Varchar string | Admin executive | First name, last name, email address | no | no | - | - |
|  | Admin records | Reports of everything | Char |  |  | no | No | - | - |
|  | Subscriber id | Unique id | - |  |  | no | No | Pk | - |
| **subscriber** | Subscriber password | Password of subscriber | Char string | subscriber |  | no | no | - | - |
|  | Subscriber details | Details of subscriber | Char string |  |  | no | No | - | - |
|  |  |  |  |  |  |  |  |  |  |
|  | Datascoop id | Unique no | - |  |  | no | no | PK | - |
|  | Altitude | Details of altitude | number |  |  | no | no | - | - |
|  | humidity | Details of humidity | number |  |  | no | no | - | -- |
|  | latitude | Details of latitude | number |  |  | no | No | - | - |
|  | Light strength | Details of light strength | number |  |  | No | no | - | -- |
| **Datascoop drone** | location | Details of location | Number | Datascoop drone |  | No | no | - | - |
|  | longitude | Details of longitude | number |  |  | No | no | - | -- |
|  | temperature | Details of temperature | number |  |  | No | no | - | - |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | - |  |
|  | Subscription id | Unique id | - |  |  |  |  | PK |  |
|  | Subscription type | Types of subscription | Char string |  |  | no | No | - | - |
| **Subscription** | Subscription details | Details of subscription | Char string | subscription |  | no | No | - | - |
|  | Subscription fees | Price of subscription | number |  |  | No | no | - | - |
|  | Staff id | Unique number | - |  |  | No | No | PK | -- |
|  | Staff first name | Tells you the first name of staff | Char string |  |  | No | no | - | - |
| **Office Staff member** | Staff address | Let you know the staff address | Char string | Staff address | Street name, Postal code | no | no | - | - |
|  | Staff details | Provide more details of the staff | Char string |  |  | No | No | - | - |
|  | Staff email address | Staff email address | Char string |  |  | No | no | - | - |
|  | Staff phone no | Phone number of staff to contact him | number |  |  | No | No | - | - |
|  | Staff last name | Last name of the staff | Char string |  |  | no | No |  |  |
| **Maintenance person** | Maintenance person Id | Unique number | Char string | Technician |  | No | no | - | - |
|  | Maintenance person last name | Last name | Char string |  |  | No | No | - | - |
|  | Maintenance person first name | First name | - |  |  | No | No | - | - |
|  |  |  |  |  |  |  |  | - |  |
| **Datascoop drone parts** | Datascoop drone part details | Details of drone parts | Char string |  |  | No | No | - | - |
|  | Datascoop drone ID | Unique no | - |  |  | no | No | PK | - |
|  |  |  |  |  |  |  |  |  |  |
| **supplier** | Supplier ID | Unique no | - |  |  | No | no | Pk | - |
|  | Supplier name | Name of the supplier | Char string |  |  | No | no | - | - |
|  | Supplier phone no | Phone no of supplier | Number |  |  | No | No | - | - |
|  | Supplier email | Email address of supplier | Email @ |  |  | No | No | - | - |
|  |  |  |  |  |  |  |  |  |  |
|  | Salesperson id | Unique number | - |  |  | No | no | Pk | - |
| **salesperson** | Salesperson first name | First name of sale person | Char string | salesperson |  | No | no | - | -- |
|  | Salesperson last name | Last name of salesperson | Char string |  |  | No | No | - | - |
|  | Salesperson phone no | Sale person phone no | Number |  |  | no | no |  |  |
|  | Sales person email address | Email address of salesperson to send him sale to him | email |  |  | no | no |  |  |
| **Website** | WEBSITE ID | Unique number | - | Website | no | no | no | pk |  |
|  | Website details | Details covering everything of the website |  |  |  | no | no |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **Zone** | Zone id | Unique id | - | Zone |  | no | no | PK |  |
|  | Zone area | Areas which cover the zone | Varchar |  |  | no | no |  |  |
| **Employees** | Employees id | Unique id | - | Employees |  | no | no | Pk |  |
|  | Employees details | Details of employee |  |  |  | no | no |  |  |
|  |  |  |  |  |  |  |  |  |  |

Assumption business rule: Business rule is quite good but it more costly. Depends on their sale.

How to transform a conceptual model to a logical model:

Codd was the created of Relational (Logical) Model in 1970. The primary distinction between the Conceptual Model and the Relational Model is that the Conceptual Model focuses on entities, whereas the Relational Model focuses on tables. The Relational Model's tables have a fixed number of columns but an infinite number of rows, or Tuples. A Tuples set is a record. The attributes are represented by the columns, and each attribute must have a domain.

Let’s compare it with conceptual and then I will explain it rules to keep in mind when you are moving to logical model:

1. Entities: An entity type is converted into a table and normally retains the same name.

Entities have an Attribute or Attributes that make up the Primary Key (PK); if more than one attribute makes up the PK, this is referred to as a Composite Key (CK). Weak entities contain their parent entity's as Foreign Key (FK).

1. Attributes : Attributes are converted into columns in the table. The entity's key characteristic in the Conceptual Model is likely to become the Relational Model's Primary Key (PK).Multi-value Attributes are transformed into tables, and the parent entity is transformed into a Foreign Key (FK) in the table.
2. Relationships: Because there are no relationships in the relational model, relationships with attributes will become tables. In one-to-one relationships, the PK of one entity becomes the FK of the other, with the designer having to decide which one.

In a one-to-many relationship, the ‘many entity' accepts the PK from the 'one entity' as an FK.

1. Participation: In most cases, the PK on the mandatory participation side becomes the foreign key on the optional participation side.
2. Superclass/Subclass: When dealing with Superclass and Subclass, there is no one-size-fits-all answer. The following are three possible solutions to the problem:
3. In the relational diagram, the superclass becomes a table with the attributes shared by all subclasses, and each subclass becomes a table with its unique attributes.
4. Remove the superclass entirely from the relational design; instead, each subclass becomes a table with all common and unique attributes.
5. The superclass is transformed into a table that contains all common and unique properties of each subclass.

Clearing the rules to in more simple way:

Each entity on the conceptual ERD is transformed into a relation.

In the corresponding relation, each attribute becomes an attribute.

Multiple attributes are created when composite attributes are combined.

The displayed candidate key may or may not become the primary key.

The relation should be named the same as the entity, the key(s) should be displayed, and all other properties should be listed.

The foreign key is inserted in either relation in a one-to-one relationship.

The primary key of the one relation is placed in the many relation in a one-to-many relationship.

An associated entity is generated in a many to many connections to connect the two relationships.

Because attributes cannot exist in relationships, a new relationship must be developed.

A new table is built to avoid null values in a partial participation to complete participation connection where the relationship is one to many with the many connection having partial participation.

History of CROW’s Foot notation:

The origins of crow's foot notation may be traced back to an article by Gordon Everest (1976, Fifth Computing Conference, IEEE). The naming convention for notations was changing; in fact, it had been evolving for some years.

When I questioned Mr. Everest about the situation, he stated:

To distinguish it from Bachman's notation, I named it the "inverted arrow." at the time. It was more appealing to me than the arrow since it did not suggest directivity or a physical access channel, and it was aesthetically intuitive, indicating many. Others began referring to it as chicken feet (e.g., Carlis textbook 1) after that.

I now call it a FORK, which is short and to the point and avoids the possessive crow's or the long chicken. The focus of my original work was on “Basic data structures described using a popular example” 2 (the title, which later became chapter 4 in my McGraw Hill text, Database Management, 1986). The usage of the notation was purely coincidental, but well designed.

[X]------< [Y]

demonstrating that a single X can be related to many Ys (and each Y relates to at most one X).

An entity is a representation of a class of object. might be a person, a location, an object, or anything else. Entities are often described through characteristics. An entity is represented by a rectangle with its name on top in crow's foot notation. The name is solitary (entity), not plural (entities).

Attributes:

The identifier is the attribute(s) that uniquely differentiates an instance of the entity. This sort of characteristic is usually denoted by an asterisk.

Relationship:

Relationships depict the connection between two entities.

They are shown as a straight line.

On the relationship line, each connection is usually given a name that is articulated as a verb.

This specifies the type of relationship that exists between the things.

Cardinality:

There are two signs in relationships.

These are shown on both sides of the line.

The first (commonly referred to as multiplicity) refers to the maximum number of times an instance of one entity can relate to instances of another entity.

It might be one or many.

The second specifies the smallest number of times one instance can be linked to another.

It might be zero or one, and it indicates whether the connection is optional or necessary.

These two signs are always combined in a precise order.

The emblem of multiplicity appears initially on the relationship's outskirts.

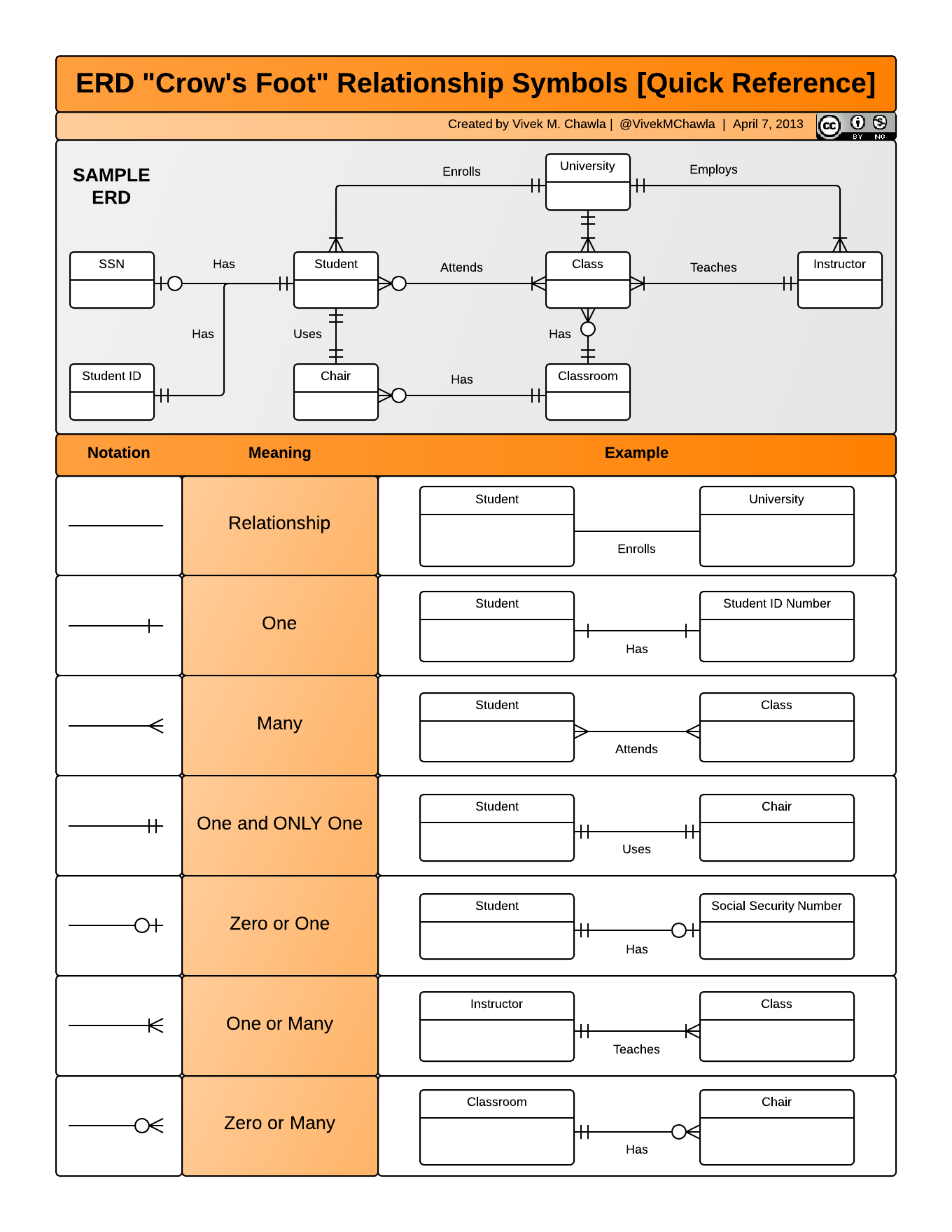
Following the multiplicity sign comes the sign indicating whether the connection is necessary or optional.

The following is written in crow's foot notation:

A straight line perpendicular to the relationship line represents a multiplicity of one and a mandatory relationship.

The three-pronged ‘crow-foot' emblem represents a multiplicity of many.

An empty circle represents an optional connection.



Why we use cross foot notation:

The logical modelling diagram is created to improve both the system and its visual appearance. The relational model describes the events that occur and identifies the data that is required. Another advantage of creating the relational diagram is that it lays the groundwork for the building of the Physical diagram.

Logical diagrams are frequently used to help non-technical business teams grasp the database. This is incredibly valuable to the development team because it allows them to engage the business team in the process. This can be incredibly valuable since it helps the business team to root out assumptions mistakes and discover any flaws in the database architecture.

HOW:

In some aspects, the transition from Conceptual to Relational can be relatively simple.

A significant amount of system visualization has been accomplished, and as a result, it can offer a significantly clearer grasp of what the business outcome is, which is an asset shared by the teams involved.

The goal is to follow the guidelines given above and understanding where judgments about which option to go must be made. This is especially true for more abstract features like multi-valued characteristics, relationships with their own characteristics, and the transition from a relationship structure to Primary Keys and Foreign Keys.

Normalization:

Database normalisation is a method for structuring data in a database. Normalization is a method of dissecting tables in order to minimise data redundancy (repetition) and undesired features such as Insertion, Update, and Deletion Anomalies. It is a multi-step procedure that converts data into tabular format while also deleting duplicate data from relation tables.

Normalization serves primarily two functions.

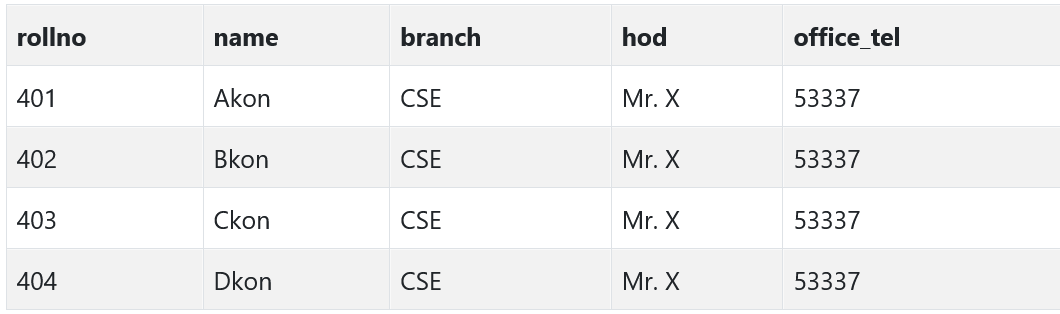
1)Data that is redundant (useless) is removed.

2) Assuring that data dependencies make sense, i.e. that data is stored properly.

Example:

Problems Due to a Lack of Normalization:

If a table is not correctly normalised and has data redundancy, it will not only use extra memory space, but it will also make it difficult to manage and update the database without experiencing data loss.

Let's look at a Student table to see how these anomalies work: 

We have data from four Computer Science students in the table above. As we can see, data for the fields branch, hod (Head of Department), and office tel are duplicated for students in the same branch at the institution; this is known as data redundancy.

**Insertion Anomaly**

I Assume that for a new admission, until and until a student chooses a branch, the student's data cannot be added; otherwise, the branch information must be set to NULL.

Furthermore, if we must input data for 100 students from the same branch, the branch information will be duplicated for all 100 students.

These possibilities are only Insertion anomalies.

Updating anomaly:

What if Mr. X drops out of college? Or is no longer the department head of computer science? In such instance, all student records must be updated, and if any record is missed by accident, data inconsistency would result. This is an update anomaly.

Delete Anomaly:

In our Student table, we keep two types of information together: Student information and Branch information.

As a result, if student records are removed at the conclusion of the academic year, we will also lose branch information.

This is known as the Deletion Anomaly.

**Normalization Rule**

Normalization rules are divided into the following normal forms:

1. First Normal Form: It should only contain attributes/columns with single (atomic) values.

The values put in a column should all belong to the same domain. Each, column in a table should have a distinct name. Furthermore, the sequence in which data is saved is unimportant.

The first typical form expects you to create your database in accordance with a few simple guidelines, which are as follows:

Rule 1: Attributes with a Single Value

Your table's columns should be single valued, which means they should not have multiple values.

Rule 2: Attribute Domain should not be modified. This is a rule of "Common Sense."

The values entered in each column must be of the same sort or type. It should just save the 'date of birth' for all records/rows.

For example, if you have a column dob to store the dates of birth of a group of people, you cannot or must not preserve the 'names' of some of them in that column with the 'dates of birth' of others.

Rule 3: Each Attribute/Column must have a unique name. This rule requires that each column in a table have a distinct name.

This is selected to reduce misunderstanding while retrieving data or doing other operations on stored data. If one or more columns have the identical name, the DBMS system will be perplexed.

Rule 4: It does not matter what order you do things in.

This rule states that the order in which the data in your table is stored is irrelevant.

1. Second Normal Form:

It should be written in First Normal Form.

Single column Primary Key

It should also not be partially dependent.

What are Transitive Functional dependencies:

1. Third Normal Form

1. It is written in the Second Normal Form.

2. It also lacks Transitive Dependency.

1. BCNF:

is a more advanced variation of Third Normal Form. This form deals with anomalies that are not handled by 3NF. BCNF refers to a 3NF table that does not contain several overlapping candidate keys. The following requirements must be met for a table to be in BCNF:

1. R must be in the third Normal Form.
2. Secondly, X should be a super Key for each functional dependence ( X Y ).
3. Fourth Normal Form:
4. It is written in Boyce-Codd Normal Form.
5. It also lacks Multi-Valued Dependency.

6) 5th normal form: A table is in 5th Normal Form if it is in 4NF and cannot be divided into any number of smaller tables without data loss.

7) 6th Normal form: Although the 6th Normal Form is not standardised, database professionals have been discussing it for some time.

Hopefully, soon, we will have a clear and consistent definition for 6th Normal Form.

Database Normal Forms

**Document relations**

|  |  |  |  |
| --- | --- | --- | --- |
| **Relation Name** | **Start Volume** No. of rows loaded at the beginning | **Growth** e.g. no growth / 10% per year | **Comments** |
| Data scoop drone | 50000 | 30 |  |
| Data scoop drone data | 50000 | 5 |  |
| Flight stream | 260 | 20 |  |
| Employees | 250 | 20 |  |
| Office staff member | 150 | 10 |  |
| Contract | 20000 | 25 |  |
| Subscription | 500000 | 20 |  |
| Subscriber | 500000 | 20 |  |
| Video stream | 25000 | 15 |  |
| Website | 2000 | 10 |  |
| Zone | 100 | 5 |  |
| Supplier | 50 | 5 |  |
| Data scoop parts | 150 | 10 |  |
| Salesperson | 50 | 20 |  |
| Admin executive | 10 | 1 |  |
| Maintenance person | 40 | 5 |  |

**Document Attributes**

| **Relation name** | **Attributes** | **Description** | **Data type** | **Length** | **Value range** | **Validation Rules** | **Default**  **value** | **Key?** | **Nulls** | **Reference’s entity** | **Integrity constraints** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Video time | Duration of time | time | 00:00:00.0000000 to 23:59:59.9999999 |  |  | no |  | - |  |  |
|  | Video id | Unique number | integer |  |  | Unique, Auto increment | no | PK | - | Data scoop drone |  |
|  | Video quality | Quality of video | Small integer | 4 |  | no | no | - | - |  |  |
| **Video stream** | Video details | details | varchar | 50 |  | no | No |  | - |  |  |
|  | GOLD sub-Id | Unique id | integer |  |  | Unique, Auto increment |  | PK | - | subscriber |  |
|  | Gold sub name | Name of gold sub | varchar | 20 |  | no | No | - | -- |  |  |
| **Gold sub** | Gold sub details | Details of gold sub | varchar |  |  | no | no | - | - |  |  |
|  | Gold sub fees | Fees of gold sub | decimal | 50 | A-E | no | no | - | - |  |  |
|  | Platinum sub id | Unique id | integer |  |  | Unique, Auto increment |  | PK | -- | subscriber |  |
|  | Platinum sub name | Name of platinum sub | varchar | 20 |  | no | No | - | - |  |  |
| **Platinum sub name** | Platinum sub details | Details of platinum sub | varchar | 50 |  | no | No | - | - |  |  |
|  | Platinum sub fees | Fees of Platinum sub | decimal | 50 | A-E | no | No | - | - |  |  |
|  | Super platinum id | Unique id | integer |  |  | Unique, Auto increment |  | PK | - | subscriber |  |
|  | Super platinum name | Super platinum name | varchar | 20 |  | No | No | - | - |  |  |
| **Super platinum** | Super platinum fees | Payment of super platinum | decimal | 50 |  | No | no | - | - |  |  |
|  | Super platinum  details | Details of super platinum | char | 50 |  | No | no | - | - |  |  |
|  | Standard id | Unique id | integer |  |  | Unique, Auto increment |  | PK | - | subscriber |  |
| **Standard** | Standard name | Name of standard user | varchar | 20 |  | no | no | - | - |  |  |
|  | Standard fees | Standard fees | decimal | 50 | A-E | No | No | - | - |  |  |
|  | Standard details | Standard details | varchar | 50 |  | No | No | - | - |  |  |
|  |  |  |  |  |  |  |  |  | - |  |  |
|  | Contract id | Unique number | integer |  |  | Unique, Auto increment | No | PK | - |  |  |
| **Contract** | Contract details | Details of contract | varchar | 50 |  | No | No | - | - |  |  |
|  | Contract type | Type of contract | varchar | 10 |  | no | No | - | - |  |  |
|  |  |  |  |  |  |  |  |  | - |  |  |
|  | Data scoop id | Unique number | integer |  |  | Unique, Auto increment | No | PK | - |  |  |
|  | Altitude | Details of altitude | decimal | 50 |  | NO | No | - | -- |  |  |
| **Data scoop drone data** | Light strength | Details of light strength | decimal | 50 |  | No | No | - | - | DATASCOOP drone |  |
|  | temperature | Details of temperature | integer | 50 |  | No | No | - | - |  |  |
|  | Humidity | Details of humidity | integer | 50 |  | no | No | - | - |  |  |
|  |  |  |  |  |  |  |  |  | - |  |  |
|  |  |  |  |  |  |  |  |  | - |  |  |
| **Flight stream** | Flight stream id | Unique id | integer |  |  | Unique, Auto increment | no | Pk | - |  |  |
|  | Flight stream address | Address of the company | Char string | 30 | . | No | no | - | -- |  |  |
|  | Flight stream phone no | Phone no the company | varchar | 10 |  | No | No | - | - |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | Admin id | Unique id | - integer |  |  | Unique, Auto increment | No | PK | -- | employee |  |
| **Admin executive** | Admin details | Details of his information | Varchar | 50 |  | no | no | - | - |  |  |
|  | Admin records | Reports of everything | varchar | 30 |  | no | No | - | - |  |  |
|  | Subscriber id | Unique id | integer |  |  | Unique, Auto increment | No | Pk | - |  |  |
| **subscriber** | Subscriber password | Password of subscriber | Char string | 10 |  | no | no | - | - | Subscription |  |
|  | Subscriber details | Details of subscriber | Char string | 50 |  | no | No | - | - |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | Data scoop id | Unique no | integer |  |  | Unique, Auto increment | no | PK | - | Flight stream |  |
|  | Altitude | Details of altitude | decimal | 50 |  | no | no | - | - |  |  |
|  | humidity | Details of humidity | decimal | 50 |  | no | no | - | -- |  |  |
|  | latitude | Details of latitude | decimal | 50 |  | no | No | - | - |  |  |
|  | Light strength | Details of light strength | Decimal | 50 |  | No | no | - | -- |  |  |
| **Data scoop drone** | location | Details of location | varchar | 50 |  | No | no | - | - |  |  |
|  | longitude | Details of longitude | Decimal | 50 |  | No | no | - | -- |  |  |
|  | temperature | Details of temperature | varchar | 50 |  | No | no | - | - |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | - |  |  |  |
|  | Subscription id | Unique id | integer |  |  | Unique, Auto increment |  | PK |  | contract |  |
|  | Subscription type | Types of subscription | varchar | 30 |  | no | No | - | - |  |  |
| **Subscription** | Subscription details | Details of subscription | varchar | 50 |  | no | No | - | - |  |  |
|  | Subscription fees | Price of subscription | decimal | 50 | A-E | No | no | - | - |  |  |
|  | Staff id | Unique number | integer |  |  | Unique, Auto increment | No | PK | -- |  |  |
|  | Staff first name | Tells you the first name of staff | varchar | 30 |  | No | no | - | - |  |  |
| **Office Staff member** | Staff address | Let you know the staff address | varchar | 40 |  | no | no | - | - | employee |  |
|  | Staff details | Provide more details of the staff | varchar | 50 |  | No | No | - | - |  |  |
|  | Staff email address | Staff email address | Varchar | 10 |  | Include @ | Include @ | - | - |  |  |
|  | Staff phone no | Phone number of staff to contact him | varchar | 10 |  | No | No | - | - |  |  |
|  | Staff last name | Last name of the staff | Varchar string | 50 |  | no | No |  |  |  |  |
| **Maintenance person** | Maintenance person Id | Unique number | integer |  |  | Unique, Auto increment | no | PK | - | employee |  |
|  | Maintenance person last name | Last name of the person | varchar | 30 |  | No | No | - | - |  |  |
|  | Maintenance person first name | First name of the person | Varchar | 30 |  | No | No | - | - |  |  |
|  | Data scoop drone part size | Size of the part | int | 10 |  | NO | No | - |  |  |  |
| **Data scoop drone parts** | Data scoop drone part name | Details of drone parts | Varchar | 50 |  | No | No | - | - |  |  |
|  | Data scoop drone part ID | Unique no | integer |  |  | Unique, Auto increment | No | PK | - | Maintenance person |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **supplier** | Supplier ID | Unique no | integer |  |  | Unique, Auto increment | no | Pk | - |  |  |
|  | Supplier name | Name of the supplier | varchar | 50 |  | No | no | - | - |  |  |
|  | Supplier phone no | Phone no of supplier | varchar | 10 |  | No | No | - | - |  |  |
|  | Supplier email | Email address of supplier | Email address of the supplier | 10 |  | Include @ | Include @ | - | - |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | Salesperson id | Unique number | integer |  |  | Unique, Auto increment | no | Pk | - | employee |  |
| **salesperson** | Salesperson first name | First name of sale person | varchar | 50 |  | No | no | - | -- |  |  |
|  | Salesperson last name | Last name of salesperson | varchar | 50 |  | No | No | - | - |  |  |
|  | Salesperson phone no | Sale person phone no | varchar | 10 |  | no | no |  |  |  |  |
|  | Salesperson email address | Email address of salesperson to send him sale to him | Email id of the salesperson | 10 |  | Include @ | no |  |  |  |  |
| **Website** | WEBSITE ID | Unique number | integer |  | no | Unique, Auto increment | no | pk |  | video stream |  |
|  | Website details | Details covering everything of the website | varchar | 50 |  | no | no |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Zone** | Zone id | Unique id | integer |  |  | Unique, Auto increment | no | PK |  | Data scoop |  |
|  | Zone area | Areas which cover the zone | Varchar | 10 |  | no | no |  |  |  |  |
| **Employees** | Employees id | Unique id | integers |  |  | Unique, Auto increment | no | Pk |  | Flight stream |  |
|  | Employees details | Details of employee | varchar | 50 |  | no | no |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

NALER analysis:

The NaLer analysis is natural method which helps us to interpret entity relationship models by using its five steps. The main reason behind using this system is because of know that attributes and entities are at right place.

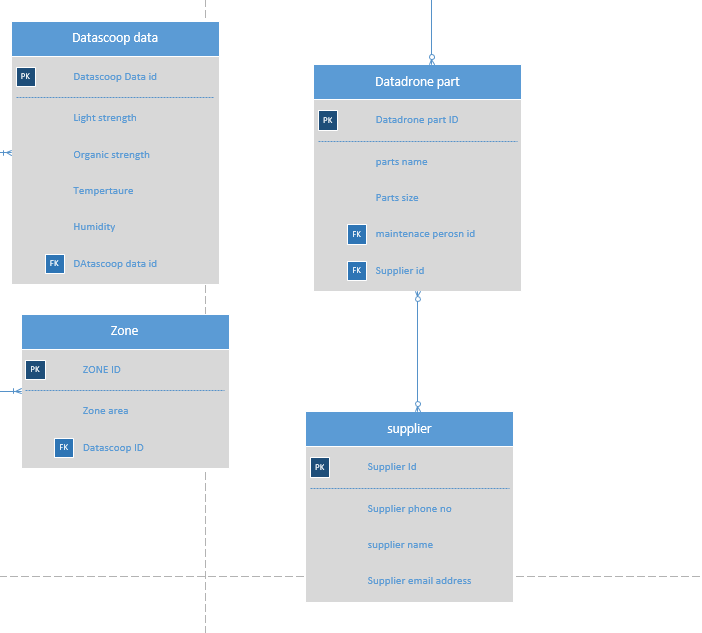
NALER analysis for my ERD:

Step 1:

Diagram convention:

I checked my diagram convention which is perfect according to me. In my diagram the entities name is display on the top. Primary key can be visible with a PK notation and for foreign key is FK which is at the bottom. The boxes with details are the attributes.

The minimum cardinality is indicated in the center of the connection, while the maximum cardinality is shown at the ends.



Step 2: Syntax check:

I have checked my syntax they are all good.

STEP 3:

* 1. Entity: Formula

*Each* entity name *is uniquely identified by* Primary key.

* 1. Attributes:

*One* E-name *identified by* PK *must have one* attribute name.

## Attribute Sentences:

S1: *Each* Data Scoop Drone is *uniquely identified by* DatascoopdroneID

S2: One Datascoopdrone (Datascoop drone id) must have one Organic spectral data

S3: One Datascoopdrone (Datascoop drone id) must have one location

S4: One Datascoopdrone (Datascoop drone id) must have one Light strength

S5: One Datascoopdrone (Datascoop drone id) must have one Temperature

S6: One Datascoopdrone (Datascoop drone id) must have one Humidity

S7: One Datascoopdrone (Datascoop drone id) must have one Latitude

S8: One Datascoopdrone (Datascoop drone id) must have one Longitude

S9: *Each* Flight stream is *uniquely identified by* FlightstreamID

S10: One Flightstream (Flightstream ID) must have one Address

S11: One Flightstream (Flightstream ID) must have one Phone

S12: *Each* ZONE is *uniquely identified by* ZONEID

S13: One ZONE (ZONE ID) must have zone area

S14: *Each* Supplier is *uniquely identified by* SupplierID

S15: One Supplier (SupplierID) must have supplier phone no

S16: One Supplier (SupplierID) must have supplier name

S17: One Supplier (SupplierID) must have Supplier email address

S18: *Each* Datadrone parts is *uniquely identified by* Datadrone partsID

S19: One Datadrone part (Datadrone partsID) must have parts name

S20: One Datadrone part (Datadrone partsID) must have part size

S21: *Each* Video stream is *uniquely identified by* Video stream ID

S22: One Video stream (Video stream ID) must have video stream details

S23: One Video stream (Video stream ID) must have video stream quality

S24: One Video stream (Video stream ID) must have video stream time

S25: *Each* Website is *uniquely identified by* WEBSITE ID

S26: One Website (WEBSITE ID) must have details

S27: *Each* Employee is *uniquely identified by Employee* ID

S28: One *Employee* (*Employee* ID) must have Employees Details

S29: *Each* Admin executive is *uniquely identified by Admin executive* ID

S30: One *Admin executive* (*Admin executive* ID) must have admin executive Details

S31: One *Admin executive* (*Admin executive* ID) must have admin executive reports

S32: *Each* *Maintenance* person is *uniquely identified by Maintenance* ID

S33: One *Maintenance person* (*Maintenance* ID) must have maintenance First name

S34: One *Maintenance* person (*Maintenance* ID) must have maintenance last name

S35: *Each* office staff member is *uniquely identified by office staff member* ID

S36: One *office staff member* ID(*office staff member* ID) must have staff member first name

S37: One *office staff member* ID (*office staff member* ID must have staff member last name

S38: One *office staff member* ID (*office staff member* ID must have staff member email address

S39: One *office staff member* ID (*office staff member* ID must have staff member Job title

S40: One *office staff member* ID (*office staff member* ID must have staff member address

S41: *Each* *Salesperson* is *uniquely identified by SALESPerson ID*

S42: One *Salesperson* (*SALESPerson ID*) must have sales person first name

S43: One *Salesperson* (*SALES Person ID*) must have sales person last name

S44: One *Salesperson* (*SALES Person ID*) must have sales person phone no

S45: One *Salesperson* (*SALESPerson ID*) must have sales person report

S46: *Each* contract is *uniquely identified by contract* ID

S47: One *contract* (*contract* ID) must have Contract details

S48: One *contract* (*contract* ID) must have Contract type

S49: *Each* Subscription is *uniquely identified by* Subscription ID

S50: One Subscription (Subscription ID) must have subscription type

S51: One Subscription (Subscription ID) must have subscription details

S52: One Subscription (Subscription ID) must have subscription fees

S53: *Each* subscriber is *uniquely identified by subscriber I*D

S54: One Subscriber (*Subscriber* ID) must have username

S55: One *Subscriber* (*Subscriber* ID) must have password

S56: One *Subscriber* (*Subscriber* ID) must have email address

S57: *Each* Standard is *uniquely identified by standard* ID

S58: One *standard* (*standard* ID) must have Standard details

S59: One *standard* (*standard* ID) must have Standard fees

S60: *Each* *super platinum sub uniquely identified by super platinum sub-ID*

S61: One *super platinum sub* (*super platinum sub-ID*) must have super platinum sub name

S62: One *super platinum sub* (*super platinum sub-ID*) must have super platinum sub details

S63: One *super platinum sub* (*super platinum sub-ID*) must have super platinum sub fees

S64: *Each* *gold subscriber uniquely identified by gold subscriber ID*

S65: One *gold subscriber* (*gold subscriber ID*) must have name

S66: One *gold subscriber sub* (*gold subscriber ID*) must have details

S67: One *gold subscriber sub* (*gold subscriber ID*) must have fees

S68: *Each* *Platinum uniquely identified by Platinum ID*

S69: One *Platinum* (*Platinum ID*) must have Platinum name

S70: One *Platinum (Platinum ID*) must have platinum details

S71: One *Platinum (Platinum ID*) must have platinum fees

Relationship sentence:

Formula:

*One* <E-name1> *identified by* <PK> <optionality><relationship-name><cardinality><E-name2>*identified by <foreign key>* or

*One e-name (PK) <optionality><relationship -name>< e-name 2(foreign key)*

One Datascoop drone (Datascoop drone id) may

MILESTONE 3

PHYSICAL Database design