Assignment #1, Problem Statement #2
CSCI 5408 - Data Management, Warehousing, Analytics

1.1 Problem Statement: Perform analysis on the dataset provided by Dalhousie Ocean Research.

1.2 Dataset Context: http://oceantrackingnetwork.org/about/#oceanmonitoring

2.1 Preliminary Analysis of the dataset (as received):

SI No	**CSV Name**	**Representation**
1.	otnunit_aat_datacenter_attributes_8a94_cefd_f8a3	Data Centers: OTN-NEP, OTN-GLOBAL, SAF, NEP.
2.	otnunit_aat_project_attributes_f29c_fb21_23a3	Projects across the world under OTN Umbrella.
3.	otnunit_aat_receivers_c595_05f4_68b2	Receivers placed at strategic locations and Deployment details.
4.	otnunit_aat_recover_offload_details_4b23_f002_f89a	Recoveries of receivers.
5.	otnunit_aat_manmade_platform_0735_7c9f_329c	Receiver Platforms: Animal, Glider, Underwater Mooring
6.	otnunit_aat_animals_8dc3_4d15_c278	Animals in Oceans around the world.
7.	otnunit_aat_tag_releases_b793_03e7_a230	Tagged (Outfitted with sophisticated sensors) sea creature releases and Transmitter details.
8.	otnunit_aat_detections_9062_5923_1394	Detection of transmitters by the receivers.

2.2 Report on the data-set and attributes (250+ words):

Color Coding for the summary:

🟦 Entities

🟩 Attributes

🟧 Values

The key models/entities derived from the dataset are Data Center, Project, Receiver, Transmitter/Tag, Animal (Sea Creatures), Deployment, and Recovery of Receiver/Transmitter. There are 4 data centers shared across projects advocated by various organizations across the world. Scientists Tag (transmitter) and release a wide range of animals in the sea. Different types of receivers such as gliders and underwater mooring are deployed in strategic locations in the ocean. Signals from transmitters (tagged animals) are captured by the receivers when the tagged animal crosses the defined proximity of the receiver. The recovery of the data from the receivers may be done remotely or requires manual intervention to collect the data for further analysis. The time and geolocation (latitude and longitude) of all receivers at the time of deployment & recovery, similarly for transmitters at the time of tagging and detection, are among the essential and common attributes across entities. Hence it is evident that every tagged animal can have many detections from various receivers, and receivers require frequent deployment and recovery for collecting the data. In addition, all the entities have many other important attributes, which play a crucial role in data analysis. For example,

Receiver: serial number, manufacturer, platform type, deployment & recovery (time & geolocation), and depth.

Animal: vernacular name, scientific name, length, weight, life stage (adult, sub-adult, juvenile, smolt).

Release Tag: location, time, manufacturer, model, serial number, end date, coding type, and transmitter details.

Recovery: deployment, location, time, outcome/status, offload time, and comments.

Detection: transmitter details, location, time, receiver & deployment details.

Furthermore, the project and data center details are always in context for all entities.

Note: The names of Entities and attributes used in the above summary are not the exact name mentioned in the dataset, rather a more generic terminology.

2.3 All attributes in the dataset (as received):

Note: Attributes with comments are in bold in the below table.

Entity	Attributes
Data Center Attributes:	1. **datacenter_reference**: Datacenter Unique ID. 2. **datacenter_name**: Data Center Name 3. **datacenter_abstract**: Remove extra white spaces. 4. datacenter_citation 5. datacenter_pi 6. datacenter_pi_organization 7. datacenter_pi_contact 8. datacenter_infourl 9. datacenter_keywords 10. datacenter_keywords_vocabulary 11. datacenter_doi 12. **datacenter_license**: Remove extra white spaces. 13. **datacenter_distribution_statement**: Redundant. 14. **datacenter_date_modified**: Redundant. 15. datacenter_geospatial_lon_min 16. datacenter_geospatial_lon_max 17. datacenter_geospatial_lat_min 18. datacenter_geospatial_lat_max 19. **time_coverage_start**: Redundant. 20. **time_coverage_end**: Redundant.
Project Attributes	1. **project_reference**: Project Unique ID, primary key 2. **datacenter_reference**: Foreign Key with Data Center 3. **project_name**: Project Candidate Key for Primary 4. **project_abstract**: Remove extra white spaces. 5. project_citation 6. project_pi 7. project_pi_organization 8. project_pi_contact

	9. project_infourl 10. **project_keywords**: Remove extra white spaces. 11. project_keywords_vocabulary 12. **project_references**: Redundant. 13. **project_doi**: Redundant. 14. project_license 15. **project_distribution_statement**: Redundant. 16. **project_date_modified**: Redundant. 17. project_datum 18. **project_geospatial_lon_min**: Clean and Convert to Decimal(11,8) 19. **project_geospatial_lon_max**: Clean and Convert to Decimal(11,8) 20. **project_geospatial_lat_min**: Clean and Convert to Decimal(10,8) 21. **project_geospatial_lat_max**: Clean and Convert to Decimal(10,8) 22. **project_linestring**: Redundant. 23. geospatial_vertical_min 24. geospatial_vertical_max 25. **geospatial_vertical_positive**: Redundant. 26. **time_coverage_start**: Redundant. 27. **time_coverage_end**: Redundant.
Receivers	1. **deployment_project_reference**: Foreign Key - Project 2. **datacenter_reference**: Foreign Key - Datacenter, redundant key as the project is already present. 3. **deployment_id**: Unique ID represents the deployment of receivers. 4. **deployment_guid**: Deployment ID concatenated with datacenter, project, and receiver details. 5. receiver_manufacturer 6. **receiver_model**: receiver model, receiver serial number,

	and receiver reference id is a composite key to uniquely identify a receiver. 7. **frequencies_monitored**: Redundant. 8. **receiver_coding_scheme**: Redundant. **9. receiver_serial_number** 10. **latitude**: Clean and Convert to Decimal(10,8) 11. **longitude**: Clean and Convert to Decimal(11,8) 12. **time**: Clean and Convert to MySQL Datetime. 13. **recovery_datetime**: Clean and Convert to MySQL Datetime. 14. **array_name**: duplicate column of project reference. 15. receiver_reference_type **16. receiver_reference_id:** Cleaned to remove project reference mentioned twice. Ex: “HFX_HFX001” to “HFX001”. 17. **bottom_depth**: Clean and convert to decimal. 18. **depth**: Clean and convert to decimal, represents the depth at which deployment is done. 19. **deployment_comments**: remove extra white spaces. 20. **deployed_by**: Redundant. 21. **expected_receiver_life**: Redundant.
Recover Offload Details	1. **recovery_project_reference**: FK Project Reference. 2. **datacenter_reference**: FK Datacenter reference, redundant as the project is already present. 3. **recovery_id**: PK to uniquely identify a recovery. 4. **deployment_id**: FK deployment. Important to co-relate the recovery and deployment. 5. **recovery_guid**: Combination of Datacenter, project, and recovery reference. 6. **recovery_latitude**: Clean and Convert to Decimal(10,8) 7. **recovery_longitude**: Clean and Convert to Decimal(11,8) 8. **recovery_datetime**: Clean and Convert to MySQL

	Datetime. 9. recovery_outcome 10. data_offloaded 11. **offload_datetime**: Clean and Convert to MySQL Datetime. 12. log_filenames 13. recovery_comments 14. **clock_synchronized**: Redundant. 15. **recovered_by**: Redundant.
Manmade Platform	1. **platform_project_reference**: FK Project Reference. 2. **datacenter_reference**: Foreign key with Data Center: Redundant column as project FK is already present. 3. **platform_reference_id**: Unique ID - Primary Key represents mandate platforms such as gliders (receivers). 4. **platform_guid**: Datacenter + Project + Platform Reference ID. 5. **platform_type**: Glider and Underwater Mooring. 6. **platform_depth**: Clean and convert to decimal. 7. platform_name 8. **latitude**: Clean and Convert to Decimal(10,8) 9. **longitude**: Clean and Convert to Decimal(11,8)
Animals	1. **animal_project_reference**: FK project reference. 2. **datacenter_reference**: FK Datacenter reference, redundant as project reference is present. 3. **animal_reference_id**: PK to uniquely identify an animal (sea-creatures) 4. **animal_guid**: Project Reference + Animal Reference. 5. vernacularname 6. **scientificname**: Fix blank values with correct implicit records. 7. **taxonrank**: Redundant. 8. aphiaid

	9. tsn 10. animal_origin 11. stock 12. **length**: Clean and convert to decimal. 13. length_type 14. **weight**: Clean and convert to decimal. 15. life_stage 16. **age**: Clean and convert to decimal. 17. sex
Tag Releases	1. **release_project_reference**: FK project reference. 2. **datacenter_reference**: FK datacenter reference. 3. **tag_device_id**: Unique identifier for a tag device: tag model + tag serail number + transmitter ID + Coding System. 4. **release_guid**: Datacenter + Project + Release reference. 5. **release_reference_id**: FK, Represents the animal or the station based on the release reference type. 6. **release_reference_type**: STATION or ANIMAL 7. **latitude**: Clean and Convert to Decimal(10,8) 8. **longitude**: Clean and Convert to Decimal(11,8) 9. **time**: Clean and Convert to MySQL Datetime. 10. **expected_enddate**: Clean and Convert to MySQL Datetime. 11. manufacturer **12. tag_model** 13. **tag_serial_number** 14. **tag_frequency**: Redundant. 15. tag_coding_system 16. **transmitted_id**: Presuming transmitter refers to the tag and uniquely identifies a transmitter. 17. transmittername 18. **transmitter_type**: Redundant. 19. **tag_programming_id**: Redundant.

Detections	1. **detection_project_reference**: FK project reference. 2. **datacenter_reference**: FK datacenter reference. 3. **detection_id**: Uniquely represents a detection by the receiver of an animal, hence has a lot of entries. 4. **detection_guid**: Datacenter + Project + Detection Reference. 5. **time**: Clean and Convert to MySQL Datetime. 6. **latitude**: Convert to Decimal(10,8) 7. **longitude**: Convert to Decimal(11,8) 8. tracker_reference 9. **detection_reference_id**: FK with Animals or Stations (Always animals as per the dataset). 10. **detection_reference_type**: ANIMAL or STATION. 11. transmitter_codespace 12. **transmitter_id**: uniquely identifies a transmitter. 13. detection_transmittername 14. detection_serial_number 15. sensor_data 16. sensor_data_units 17. receiver_log_id: Redundant. 18. **deployment_id**: FK Deployment Reference. 19. **detection_quality**: Redundant. 20. **depth**: Clean and convert to decimal. 21. position_data_source 22. uncertainty_in_latitude 23. uncertainty_in_longitude 24. **depth_data_source**: Redundant. 25. **uncertainty_in_depth**: Redundant. 26. other_position_data 27. dataset_quality

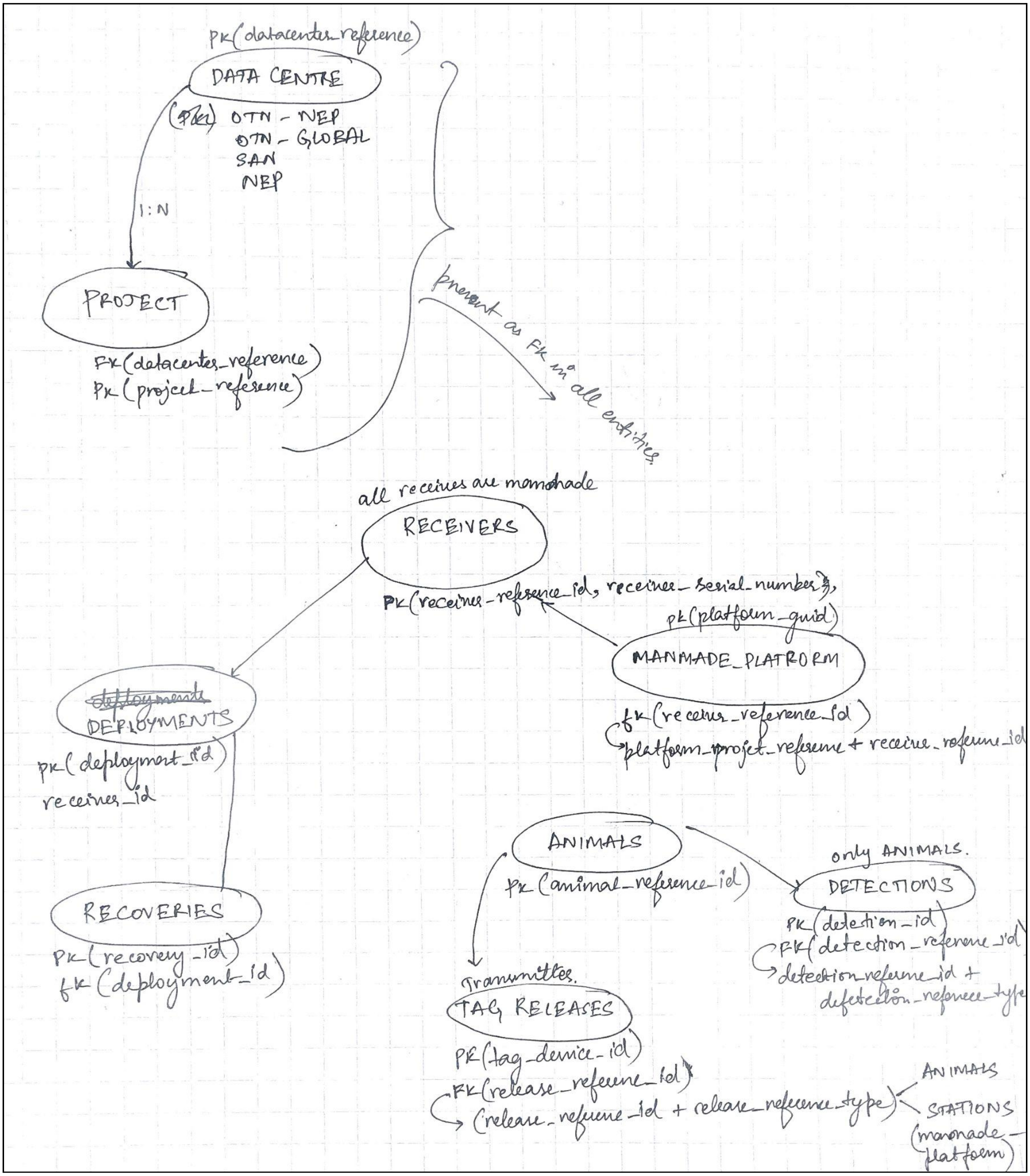
3.1 Data Cleaning and Transformation:

3.2 Common:

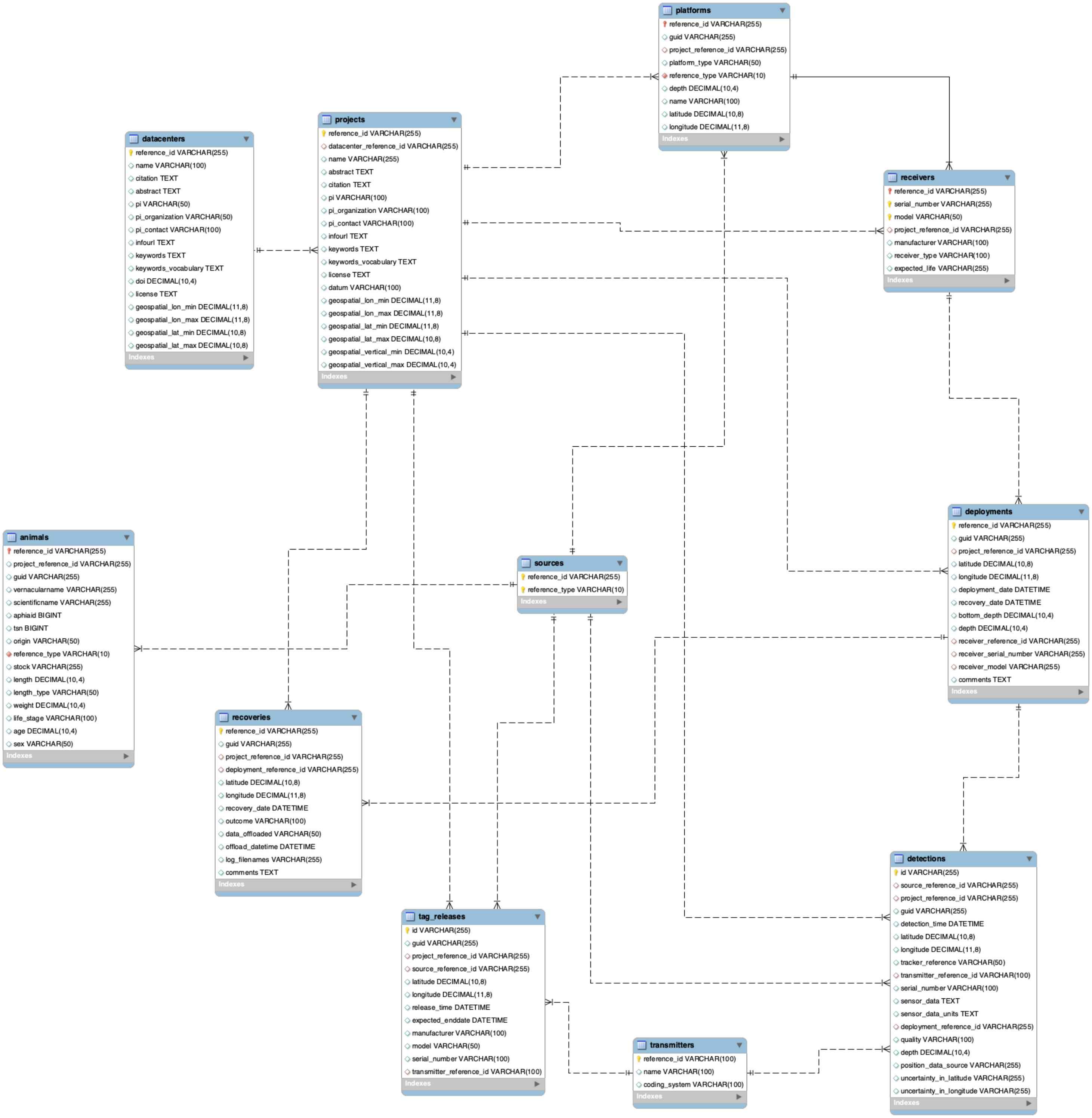
1. The first row in the CSV represents the attribute names, and the second row is the units of the attributes; the second row is deleted from all the CSVs, as it’s a convention to assume that the CSV contains data from the second row onwards.
   However, it is vital to know the units of attributes; although deleted, the attributes and units details are always known (stored) for analysis and transformations.
2. Columns/attributes which are empty/null for all rows are deleted.
3. Since the Data Center entity relation can be determined if the project_reference_id is known, the datacenter_reference_id column is removed across entities expect Project table.
4. Columns with values junk values are replaced by NULL, as it's performant to perform null checks rather than string comparison.
5. Columns (Depth, Height, Age, etc.) with Decimal/Integer values stored as a string are parsed to a Decimal. Example: "10" is number 10.
6. The prefix of the table/entity name is removed from the attributes across all entities. Example: "datacenter_abstract" is just "abstract" in the datacenter table, as it's implicit.
7. All the date/time attributes are converted to MySQL DateTime format to run queries without pre-formatting.
8. Attributes holding Latitude are cleaned, and the datatype is Decimal(10, 8), similarly, Decimal(11, 8) for Longitude.
9. Remove Duplicate prefixes for consistency. For example, animal reference ID is for the format <project reference id>-<Interger>, while the same is not followed while referencing in detections.
10. Fix platform reference ID based on data in GUID for values without the project reference.
11. To ensure consistency, all the unique identifiers, GUID and enums are capital-cased.
12. Columns marked as redundant are empty columns and deleted. However, columns with junk data are cleaned but not deleted, as it translates to improper data ingestion and could be fixed at the source at a later stage.

4.1 Relationational Database Schema - Rough Sketch:

Note: The figure below does not represent an ERD but rather a projection of different possible entities and relationships derived from the dataset for the initial design.



4.1 MySQL Reverse Engineer - ERD - Version 1 (has design issues)


platforms
reference_id VARCHAR(255)
guid VARCHAR(255)
project_reference_id VARCHAR(255)
platform_type VARCHAR(50)
reference_type VARCHAR(10)
depth DECIMAL(10,4)
name VARCHAR(100)
latitude DECIMAL(10,8)
longitude DECIMAL(11,8)
Indexes
datacenters
reference_id VARCHAR(255)
name VARCHAR(100)
citation TEXT
abstract TEXT
pi VARCHAR(50)
pi_organization VARCHAR(50)
pi_contact VARCHAR(100)
infourl TEXT
keywords TEXT
keywords_vocabulary TEXT
doi DECIMAL(10,4)
license TEXT
geospatial_lon_min DECIMAL(11,8)
geospatial_lon_max DECIMAL(11,8)
geospatial_lat_min DECIMAL(10,8)
geospatial_lat_max DECIMAL(10,8)
Indexes
projects
reference_id VARCHAR(255)
datacenter_reference_id VARCHAR(255)
name VARCHAR(255)
abstract TEXT
citation TEXT
pi VARCHAR(100)
pi_organization VARCHAR(100)
pi_contact VARCHAR(100)
infourl TEXT
keywords TEXT
keywords_vocabulary TEXT
license TEXT
datum VARCHAR(100)
geospatial_lon_min DECIMAL(11,8)
geospatial_lon_max DECIMAL(11,8)
geospatial_lat_min DECIMAL(11,8)
geospatial_lat_max DECIMAL(10,8)
geospatial_vertical_min DECIMAL(10,4)
geospatial_vertical_max DECIMAL(10,4)
Indexes
receivers
reference_id VARCHAR(255)
serial_number VARCHAR(255)
model VARCHAR(50)
project_reference_id VARCHAR(255)
manufacturer VARCHAR(100)
receiver_type VARCHAR(100)
expected_life VARCHAR(255)
Indexes
deployments
reference_id VARCHAR(255)
guid VARCHAR(255)
project_reference_id VARCHAR(255)
latitude DECIMAL(10,8)
longitude DECIMAL(11,8)
deployment_date DATETIME
recovery_date DATETIME
bottom_depth DECIMAL(10,4)
depth DECIMAL(10,4)
receiver_reference_id VARCHAR(255)
receiver_serial_number VARCHAR(255)
receiver_model VARCHAR(255)
comments TEXT
Indexes
animals
reference_id VARCHAR(255)
project_reference_id VARCHAR(255)
guid VARCHAR(255)
vernacularname VARCHAR(255)
scientificname VARCHAR(255)
aphiaid BIGINT
tsn BIGINT
origin VARCHAR(50)
reference_type VARCHAR(10)
stock VARCHAR(255)
length DECIMAL(10,4)
length_type VARCHAR(50)
weight DECIMAL(10,4)
life_stage VARCHAR(100)
age DECIMAL(10,4)
sex VARCHAR(50)
Indexes
sources
reference_id VARCHAR(255)
reference_type VARCHAR(10)
Indexes
recoveries
reference_id VARCHAR(255)
guid VARCHAR(255)
project_reference_id VARCHAR(255)
deployment_reference_id VARCHAR(255)
latitude DECIMAL(10,8)
longitude DECIMAL(11,8)
recovery_date DATETIME
outcome VARCHAR(100)
data_offloaded VARCHAR(50)
offload_datetime DATETIME
log_filenames VARCHAR(255)
comments TEXT
Indexes
detections
id VARCHAR(255)
source_reference_id VARCHAR(255)
project_reference_id VARCHAR(255)
guid VARCHAR(255)
detection_time DATETIME
latitude DECIMAL(10,8)
longitude DECIMAL(11,8)
tracker_reference VARCHAR(50)
transmitter_reference_id VARCHAR(100)
serial_number VARCHAR(100)
sensor_data TEXT
sensor_data_units TEXT
deployment_reference_id VARCHAR(255)
quality VARCHAR(100)
depth DECIMAL(10,4)
position_data_source VARCHAR(255)
uncertainty_in_latitude VARCHAR(255)
uncertainty_in_longitude VARCHAR(255)
Indexes
tag_releases
id VARCHAR(255)
guid VARCHAR(255)
project_reference_id VARCHAR(255)
source_reference_id VARCHAR(255)
latitude DECIMAL(10,8)
longitude DECIMAL(11,8)
release_time DATETIME
expected_enddate DATETIME
manufacturer VARCHAR(100)
model VARCHAR(50)
serial_number VARCHAR(100)
transmitter_reference_id VARCHAR(100)
Indexes
transmitters
reference_id VARCHAR(100)
name VARCHAR(100)
coding_system VARCHAR(100)
Indexes

4.1.1 Structure Dump - Initial Modeling

4.1.2 ERD - Correct Modeling - V1

4.1.3 Structure + Value Dump - Correct Modeling - V1

3.2 Normalization and Denormalization:

1. Receivers (otnunit_aat_receivers_c595_05f4_68b2) has been normalized to two entities, namely deployment and receiver. Deployment in this context refers to a deployment of a receiver in strategic locations. Hence the same receiver can have multiple deployments over time, resulting in a one-to-many relationship.

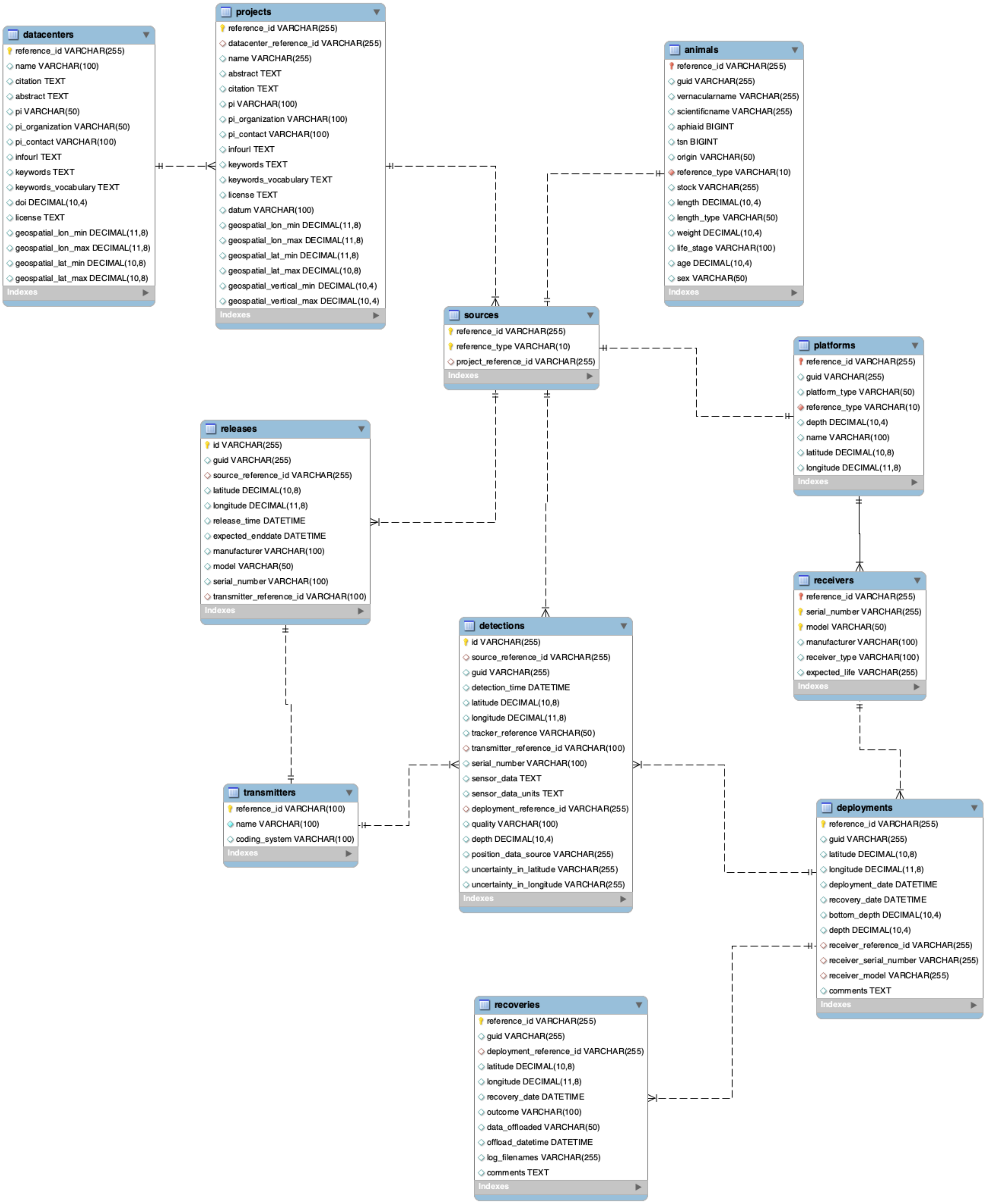
2. Tag Releases (otnunit_aat_tag_releases_b793_03e7_a230) have been normalized to Tag Releases and Transmitters; Transmitter in this context has transmitter coding details and a Release corresponds to the action (Source + Tag). Therefore, the transmitter table has the transmitter details and a one-to-one relationship with Releases and a one-to-many relationship with Detections.

3. Since Transmitter is a table, transmitter details are removed from the Detections table (otnunit_aat_detections_9062_5923_1394), with transmitter_reference_id as the foreign key.

4. Tag releases and Detections (otnunit_aat_tag_releases_b793_03e7_a230 & otnunit_aat_detections_9062_5923_1394) store the reference ID and the reference type of ANIMAL or STATION. For ANIMAL, the reference_id points to the Animal Entity and STATION points to Manmade Platform Entity. However, in MySQL, creating a foreign key constraint on two different tables for the same column is not feasible/possible. To facilitate this, a new table called "sources" had been introduced with a reference_id as the unique column; thereby, Releases and Detections don't have to hold reference type anymore and can have a foreign key on the Sources entity (refer to the ERD diagram for visualization).

5. The introduction of the table "sources" makes it easier to include a new reference type other than ANIMAL and STATION, making the system flexible for change and improving the query performance.

5.1 MySQL Reverse Engineer - ERD - Version 2 (Refer 6.1.2 Design Issues Mirage Justication)


datacenters
reference_id VARCHAR(255)
name VARCHAR(100)
citation TEXT
abstract TEXT
pi VARCHAR(50)
pi_organization VARCHAR(50)
pi_contact VARCHAR(100)
infourl TEXT
keywords TEXT
keywords_vocabulary TEXT
doi DECIMAL(10,4)
license TEXT
geospatial_lon_min DECIMAL(11,8)
geospatial_lon_max DECIMAL(11,8)
geospatial_lat_min DECIMAL(10,8)
geospatial_lat_max DECIMAL(10,8)
Indexes
projects
reference_id VARCHAR(255)
datacenter_reference_id VARCHAR(255)
name VARCHAR(255)
abstract TEXT
citation TEXT
pi VARCHAR(100)
pi_organization VARCHAR(100)
pi_contact VARCHAR(100)
infourl TEXT
keywords TEXT
keywords_vocabulary TEXT
license TEXT
datum VARCHAR(100)
geospatial_lon_min DECIMAL(11,8)
geospatial_lon_max DECIMAL(11,8)
geospatial_lat_min DECIMAL(11,8)
geospatial_lat_max DECIMAL(10,8)
geospatial_vertical_min DECIMAL(10,4)
geospatial_vertical_max DECIMAL(10,4)
Indexes
animals
reference_id VARCHAR(255)
guid VARCHAR(255)
vernacularname VARCHAR(255)
scientificname VARCHAR(255)
aphiaid BIGINT
tsn BIGINT
origin VARCHAR(50)
reference_type VARCHAR(10)
stock VARCHAR(255)
length DECIMAL(10,4)
length_type VARCHAR(50)
weight DECIMAL(10,4)
life_stage VARCHAR(100)
age DECIMAL(10,4)
sex VARCHAR(50)
Indexes
sources
reference_id VARCHAR(255)
reference_type VARCHAR(10)
project_reference_id VARCHAR(255)
Indexes
platforms
reference_id VARCHAR(255)
guid VARCHAR(255)
platform_type VARCHAR(50)
reference_type VARCHAR(10)
depth DECIMAL(10,4)
name VARCHAR(100)
latitude DECIMAL(10,8)
longitude DECIMAL(11,8)
Indexes
releases
id VARCHAR(255)
guid VARCHAR(255)
source_reference_id VARCHAR(255)
latitude DECIMAL(10,8)
longitude DECIMAL(11,8)
release_time DATETIME
expected_enddate DATETIME
manufacturer VARCHAR(100)
model VARCHAR(50)
serial_number VARCHAR(100)
transmitter_reference_id VARCHAR(100)
Indexes
receivers
reference_id VARCHAR(255)
serial_number VARCHAR(255)
model VARCHAR(50)
manufacturer VARCHAR(100)
receiver_type VARCHAR(100)
expected_life VARCHAR(255)
Indexes
detections
id VARCHAR(255)
source_reference_id VARCHAR(255)
guid VARCHAR(255)
detection_time DATETIME
latitude DECIMAL(10,8)
longitude DECIMAL(11,8)
tracker_reference VARCHAR(50)
transmitter_reference_id VARCHAR(100)
serial_number VARCHAR(100)
sensor_data TEXT
sensor_data_units TEXT
deployment_reference_id VARCHAR(255)
quality VARCHAR(100)
depth DECIMAL(10,4)
position_data_source VARCHAR(255)
uncertainty_in_latitude VARCHAR(255)
uncertainty_in_longitude VARCHAR(255)
Indexes
transmitters
reference_id VARCHAR(100)
name VARCHAR(100)
coding_system VARCHAR(100)
Indexes
deployments
reference_id VARCHAR(255)
guid VARCHAR(255)
latitude DECIMAL(10,8)
longitude DECIMAL(11,8)
deployment_date DATETIME
recovery_date DATETIME
bottom_depth DECIMAL(10,4)
depth DECIMAL(10,4)
receiver_reference_id VARCHAR(255)
receiver_serial_number VARCHAR(255)
receiver_model VARCHAR(255)
comments TEXT
Indexes
recoveries
reference_id VARCHAR(255)
guid VARCHAR(255)
deployment_reference_id VARCHAR(255)
latitude DECIMAL(10,8)
longitude DECIMAL(11,8)
recovery_date DATETIME
outcome VARCHAR(100)
data_offloaded VARCHAR(50)
offload_datetime DATETIME
log_filenames VARCHAR(255)
comments TEXT
Indexes

5.1.1 ERD - Correct Modeling V2

5.1.2 Structure + Value Dump - Correct Modeling - V2

6.1 Summary and Versions

6.1.1 Version Differences

1. The primary difference between Version 1 and 2 is that V1 has the foreign key project_reference_id in most tables, and V2 has an FK for the project in the "sources" table.
2. V1 Usecase example: To get all the detections, releases, or deployments for a given project, having an FK for the project would reduce the number of joins; furthermore, having the project context at all times ensures clean data segregation.
3. V1 has the wrong cardinality for the transmitters table, which is fixed in V2.
4. If performing several joins is not a concern or project context is not an essential attribute at all times, V2 would be ideal.

6.1.2 Design Issues Mirage Justification

1. The table "sources," have a one-to-one relationship with Platforms and Animals. At any given point, the sources table is meant to perform joins with Animals or Platforms, but never together (2 different entities). The table "sources" was introduced to ensure Tag Releases and Detections can have joins with both Platforms and Animals with a foreign key constraint, which otherwise wouldn't have been possible.
2. On similar lines, the sources have a one-to-many relationship with Detections and Releases. But this does not mean they are co-related using the Source table. Releases and Detections already have a one-to-many relationship.
3. To conclude, the sources table is not meant to co-relate entities. Instead, it allows joins with Animals/Platforms.
4. Detections and Recoveries are not directly related.

7.1 Observations and Acknowledgments :

1. Animals, stock attribute: UNK seems to the short for UNKNOWN. However, to ensure data is not misinterpreted, values are not updated.
2. **Food for thought:** Adding an attribute's unit of measurement as a suffix to a column name is a bad practice, limiting unit conversion for standardization at a later stage. That said, an alternative to consider is using **Entity-Attribute-Value Model**; the gist of the EAV model is to create another table for Unit of Measurement with an FK on unit_id in other entities.

8.1 References

1. Relational Data Store used: MySQL.
2. ERD generated using MySQL Workbench.
3. Terminology and context from Ocean Tracking Network.
4. Microsoft Excel and Google Sheets for data cleaning and transformation.
5. Wiki: Entity-Attribute-Value Model

9.1 Conclusion

The dataset in the context of OTN has been analyzed, cleaned, transformed, normalized, and stored in MySQL RDBMS to perform further analysis on the data with clear-cut defined entities. However, the solution is not a one-stop for all kinds of analysis. Depending on the scale of data and frequently used indexes, the modeling has to be re-looked, and an appropriate data store can be chosen.