Project Deliverable 3

Name: Trevor Hofmann

Chosen Area: Space Industry

**Section 1 - Introduction and Current Problems**

Chosen Area

I am choosing the Space Industry. I do not work in this industry, but it is an area I have always been interested in. Specifically, I would like to build a database that assists with object and entity tacking of the objects in space. There is a real danger as we launch more into space that we may cause collisions with other objects and cause a chain reaction that causes us to lose access to space all together.

Data Management Problems

There are a few potential data management problems that affect this area. One big problem is having multiple data sources, especially from different companies with different regulating bodies over their space industries. This means verifying the quality and integrity of the data is very important. Another problem I see is keeping as close to live data as possible. This problem can be increased because of the first problem. If the data source is not meeting the standards and speed is desired, then bad data can happen.

Motivation

Having many sources of data can cause many redundant and incomplete databases. It can be difficult to gain a complete understanding of data when it is limited to begin with. Having one central database that has already vetted its sources and verified the data integrity are some of the leading motivations.

Potential Benefits

We would need to have a standard for verifying the data integrity and not having repeated redundant data in our central database. If we could do this, live and even predict the path of objects as they travel would be a main benefit in this. People would be able to trust our database. Over all the others.

Potential Users

The benefits lead right to the possible users. Other space agencies would be some of the main users. They could use it to make sure a predicted trajectory of a launch is safe and free of debris. Also, satellite operators could use it to assist in collision avoidance systems. As long as our data is free of redundancies, is accurate, and is trusted we could assist is almost every area of the space industry.

**Section 2 - Business Rules**

1. Objects
   1. Each Object needs to track details about it such as the name, type, mass, width, height, depth.
   2. An object has one and only one type, Original Launch, Orbit, Projected Reentry, and Launch Sites.
   3. An Object can be docked to none or many other objects.
2. Object Type
   1. This is meant to track many types of objects like rocket body, payload, launch vehicle, station part, or fragment.
   2. Each Object Type can belong to one or many objects.
3. Original Launch
   1. This is the original launch information, location, date, time, failure status.
   2. Each Launch can have one or many objects.
   3. Each Launch has one and only one launch site.
4. Orbit
   1. This is the current orbit of the object. This should track the Inclination, Periapsis, Apoapsis
   2. Each orbit belongs to only one object.
5. Launch Sites
   1. These are the ground-based locations for launches. It should track the name, latitude, longitude, altitude.
   2. Each launch site can have many Launches.
6. Projected Reentry
   1. This is the projected re-entry of an object. All objects obit decay and this is tracking that. This should track the Inclination, Periapsis, Apoapsis, date, time, projected latitude, projected longitude.
   2. Each entry has only one object
7. Docked Objects
   1. This is to track things like stations that are made of many objects with unique launch information. It needs to track all objects docked together
   2. A Docked object can have by definition 2 or many objects.

**Section 3 - Entity Relation Diagram**

A screenshot of a computer screen

Description automatically generated

A black screen with white text

Description automatically generated**Section 4 – Relational Diagram**

**Section 5 - SQL Code**

**SQL Create Tables**

--Create the database itself

CREATE SCHEMA Project;

-- Object Type Table

CREATE TABLE Project.Object\_Type (

Type\_ID INT PRIMARY KEY,

Type VARCHAR(255)

);

-- Launch Sites Table

CREATE TABLE Project.Launch\_Sites (

Site\_ID INT PRIMARY KEY,

Name VARCHAR(255),

Latitude FLOAT,

Longitude FLOAT,

Altitude FLOAT

);

-- Original Launch Table

CREATE TABLE Project.Original\_Launch (

Launch\_ID INT PRIMARY KEY,

Site\_ID INT,

Launch\_Date DATE,

Launch\_Time TIME,

Success\_Failure VARCHAR(50),

FOREIGN KEY (Site\_ID) REFERENCES Project.Launch\_Sites(Site\_ID)

);

-- Orbit Table

CREATE TABLE Project.Orbit (

Orbit\_ID INT PRIMARY KEY,

Inclination FLOAT,

Periapsis FLOAT,

Apoapsis FLOAT

);

-- Projected Reentry Table

CREATE TABLE Project.Projected\_Reentry (

Projected\_ID INT PRIMARY KEY,

Inclination FLOAT,

Periapsis FLOAT,

Apoapsis FLOAT,

Projected\_Date DATE,

Projected\_Time TIME,

Projected\_Latitude FLOAT,

Projected\_Longitude FLOAT

);

-- Docked Objects Table

CREATE TABLE Project.Docked\_Objects (

Dock\_ID INT PRIMARY KEY,

Name VARCHAR(255),

Shipment\_Date DATE

);

-- Objects Table

CREATE TABLE Project.Objects (

Object\_ID INT PRIMARY KEY,

Mass FLOAT,

Width FLOAT,

Height FLOAT,

Name VARCHAR(255),

Type\_ID INT,

Launch\_ID INT,

Orbit\_ID INT,

Projected\_ID INT,

Sites\_ID INT,

Dock\_ID INT,

FOREIGN KEY (Type\_ID) REFERENCES Project.Object\_Type(Type\_ID),

FOREIGN KEY (Launch\_ID) REFERENCES Project.Original\_Launch(Launch\_ID),

FOREIGN KEY (Orbit\_ID) REFERENCES Project.Orbit(Orbit\_ID),

FOREIGN KEY (Projected\_ID) REFERENCES Project.Projected\_Reentry(Projected\_ID),

FOREIGN KEY (Sites\_ID) REFERENCES Project.Launch\_Sites(Site\_ID),

FOREIGN KEY (Dock\_ID) REFERENCES Project.Docked\_Objects(Dock\_ID)

);

**SQL Insert Data into Tables**

-- Insert data into Object\_Type

INSERT INTO Project.Object\_Type (Type\_ID, Type)

VALUES

(1, 'Rocket Body'),

(2, 'Payload'),

(3, 'Launch Vehicle'),

(4, 'Station Part'),

(5, 'Fragment'),

(6, 'Observation Satellite'),

(7, 'Communication Satellite'),

(8, 'Space Station Module'),

(9, 'Weather Satellite'),

(10, 'Navigation Satellite');

-- Insert data into Launch\_Sites

INSERT INTO Project.Launch\_Sites (Site\_ID, Name, Latitude, Longitude, Altitude)

VALUES

(1, 'Kennedy Space Center', 28.3922, -80.6077, 3),

(2, 'Baikonur Cosmodrome', 45.965, 63.305, 90),

(3, 'Cape Canaveral Space Force Station', 28.5623, -80.5774, 3),

(4, 'Vostochny Cosmodrome', 51.8507, 128.3317, 300),

(5, 'Guiana Space Centre', 5.2361, -52.7681, 50),

(6, 'Satish Dhawan Space Centre', 13.7337, 80.2344, 15),

(7, 'Jiuquan Satellite Launch Center', 40.9601, 100.2986, 100);

-- Insert data into Original\_Launch

INSERT INTO Project.Original\_Launch(Launch\_ID, Site\_ID, Launch\_Date, Launch\_Time, Success\_Failure)

VALUES

(1, 1, '2023-11-12', '18:30:00', 'Success'),

(2, 2, '2023-11-15', '15:45:00', 'Failure'),

(3, 3, '2023-11-20', '12:00:00', 'Success'),

(4, 4, '2023-11-25', '08:00:00', 'Failure'),

(5, 5, '2023-12-01', '14:30:00', 'Success'),

(6, 6, '2023-12-05', '10:15:00', 'Failure'),

(7, 7, '2023-12-10', '20:00:00', 'Success');

-- Insert data into Orbit

INSERT INTO Project.Orbit (Orbit\_ID, Inclination, Periapsis, Apoapsis)

VALUES

(1, 28.5, 200, 400),

(2, 51.6, 300, 600),

(3, 45.0, 250, 500),

(4, 60.0, 400, 700),

(5, 30.0, 180, 350),

(6, 75.0, 500, 800),

(7, 35.0, 220, 450);

-- Insert data into Projected\_Reentry

INSERT INTO Project.Projected\_Reentry (Projected\_ID, Inclination, Periapsis, Apoapsis, Projected\_Date, Projected\_Time, Projected\_Latitude, Projected\_Longitude)

VALUES

(1, 30, 250, 450, '2023-12-01', '12:00:00', 35.0, -90.0),

(2, 45, 350, 650, '2023-12-10', '08:30:00', 40.0, 120.0),

(3, 25, 200, 400, '2023-12-15', '14:45:00', 30.0, -100.0),

(4, 40, 300, 600, '2023-12-20', '10:00:00', 45.0, 130.0),

(5, 20, 150, 300, '2023-12-25', '18:30:00', 25.0, -95.0),

(6, 55, 450, 750, '2023-12-30', '16:15:00', 50.0, 110.0),

(7, 33, 210, 420, '2024-01-05', '22:00:00', 37.0, -85.0);

-- Insert data into Docked\_Objects

INSERT INTO Project.Docked\_Objects (Dock\_ID, Name, Shipment\_Date)

VALUES

(1, 'Space Station Alpha', '2023-10-01'),

(2, 'Satellite Cluster Beta', '2023-09-15'),

(3, 'Research Module Gamma', '2024-01-15'),

(4, 'Communication Hub Delta', '2023-11-30'),

(5, 'Weather Satellite Epsilon', '2023-08-20'),

(6, 'Navigation Satellite Zeta', '2023-12-05'),

(7, 'Observation Satellite Eta', '2023-09-10');

-- Insert data into Objects

INSERT INTO Project.Objects (Object\_ID, Mass, Width, Height, Name, Type\_ID, Launch\_ID, Orbit\_ID, Projected\_ID, Sites\_ID, Dock\_ID)

VALUES

(1, 5000, 3, 10, 'Satellite A', 2, 1, 1, 2, 1, NULL),

(2, 8000, 5, 15, 'Rocket Body X', 1, 2, 2, NULL, 2, 1),

(3, 3500, 2, 8, 'Satellite B', 2, 3, 3, 3, 4, NULL),

(4, 6000, 4, 12, 'Rocket Body Y', 1, 4, 4, NULL, 5, NULL),

(5, 4200, 3, 11, 'Satellite C', 2, 5, 5, 5, 6, NULL),

(6, 7500, 6, 18, 'Rocket Body Z', 1, 6, 6, NULL, 7, 2),

(7, 2800, 2, 7, 'Observation Satellite X', 6, 7, 7, 1, 3, NULL);Write the SQL code (script file) which includes the commands to create tables and to insert records into the tables for your databases. There should tables must participate in a meaningful interaction (relationships) according to your business rule.

a) The tables participating in meaningful interaction should also be available as a result of step 2.

**Section 6 - SQL Tables**

-- Select all data from Object\_Type

SELECT \* FROM Project.Object\_Type ORDER BY Type\_ID DESC;

A screenshot of a computer

Description automatically generated

-- Select all data from Launch\_Sites

SELECT \* FROM Project.Launch\_Sites ORDER BY Site\_ID DESC;

A screenshot of a computer

Description automatically generated

-- Select all data from Original\_Launch

SELECT \* FROM Project.Original\_Launch ORDER BY Launch\_ID DESC;

A screenshot of a computer

Description automatically generated

-- Select all data from Orbit

SELECT \* FROM Project.Orbit ORDER BY Orbit\_ID DESC;

A screenshot of a computer

Description automatically generated

-- Select all data from Projected\_Reentry

SELECT \* FROM Project.Projected\_Reentry ORDER BY Projected\_ID DESC;

A screenshot of a computer

Description automatically generated

-- Select all data from Docked\_Objects

SELECT \* FROM Project.Docked\_Objects ORDER BY Dock\_ID DESC;

A screenshot of a computer

Description automatically generated

-- Select all data from Objects

SELECT \* FROM Project.Objects ORDER BY Object\_ID DESC;

**A screenshot of a computer

Description automatically generated**

**Section 7 - SQL Queries**

-- List Launch Sites with the Total Number of Launches

SELECT ls.Name AS LaunchSite, COUNT(ol.Launch\_ID) AS TotalLaunches

FROM Project.Launch\_Sites ls

LEFT JOIN Project.Original\_Launch ol ON ls.Site\_ID = ol.Site\_ID

GROUP BY ls.Name;

A screenshot of a computer

Description automatically generated

-- Retrieve Objects and Their Types

SELECT o.Name AS ObjectName, ot.Type AS ObjectType

FROM Project.Objects o

JOIN Project.Object\_Type ot ON o.Type\_ID = ot.Type\_ID;

A screenshot of a computer

Description automatically generated

-- Average Mass of Objects by Object Type

SELECT ot.Type AS ObjectType, AVG(o.Mass) AS AvgMass

FROM Project.Objects o

JOIN Project.Object\_Type ot ON o.Type\_ID = ot.Type\_ID

GROUP BY ot.Type;

A screenshot of a computer

Description automatically generated

-- List Docked Objects and Their Shipment Dates

SELECT do.Name AS DockedObjectName, do.Shipment\_Date

FROM Project.Docked\_Objects do;

A screenshot of a computer

Description automatically generated

-- Total Mass Launched from Each Launch Site

SELECT ls.Name AS LaunchSite, SUM(o.Mass) AS TotalMassLaunched

FROM Project.Launch\_Sites ls

LEFT JOIN Project.Original\_Launch ol ON ls.Site\_ID = ol.Site\_ID

LEFT JOIN Project.Objects o ON ol.Launch\_ID = o.Launch\_ID

GROUP BY ls.Name;

A screenshot of a computer

Description automatically generated