SPACE DEBRIS MANAGEMENT - STUDY REPORT

Group No.: Group 18

Student Names: Janani Karthikeyan & Sneha Manjunath Chakrabhavi

Executive Summary:

The project tackles the pressing problem of space debris management, focusing on the more than 20,000 objects in orbit around Earth, that pose significant collision risks to satellites and spacecraft. The group proposes an advanced data management system capable of tracking and predictive analysis of debris movement to preemptively avoid collisions. A key component of the system is a sophisticated database designed for efficient information storage and retrieval, merging traditional database technologies with cutting-edge innovations for real-time data updates and rapid access to relevant data.

The project incorporates a comprehensive space debris database for improved debris trajectory prediction. It also features a user-friendly interface to aid mission planners and space agencies in making informed decisions. This management project was implemented by constructing suitable EER and UML models and deriving a normalized relational model. The relational model was further implemented in MySQL Workbench by creating the necessary relations and mapping them. The Application model and visualizations were implemented in Python in Jupyter Notebooks IDE. The analysis and visualizations were obtained in the form of Bar Graphs, Histograms, Line Graphs, Scatter Plots, and Pie Charts. Based on the implemented MySQL queries, the NOSQL queries were created and deployed on MongoDB Compass by importing the relations as collections and performing complex queries to ensure the integrity of the application.

The main aim of the space debris management project is to address the challenges associated with space debris tracking, mitigation, and removal to ensure the safety and sustainability of activities in Earth's orbit. The specific requirements for such a project can vary depending on its scope and objectives.

1) Orbital Data Sharing: We need to establish protocols for sharing orbital data with other space agencies, organizations, and satellite operators by creating a data-sharing agreement with international space agencies to exchange tracking information.

2) Collision Avoidance: To develop algorithms and systems to predict potential collisions between operational satellites and space debris by implementing automated collision avoidance maneuvers when necessary.

3) Debris Mitigation Guidelines: Define and promote best practices for spacecraft design to minimize the creation of space debris during launch and operation by instigating satellite operators to deorbit satellites at the end of their missions or move them to higher altitudes to reduce the risk of collision.

4) Policy and Regulatory Framework: Stress the contribution to the development of international regulations and guidelines for space debris management by advocating for the adoption of space debris mitigation and remediation measures in international space treaties.

5) Public Awareness and Education: Raise awareness about the issue of space debris and its potential impact on space activities using educational materials, workshops, and public outreach programs.

6) Debris Catalog Maintenance: Maintain an up-to-date catalog of space debris objects, including their orbital parameters and potential collision risks, by regularly updating the catalogs.

7) Emergency Response Plan: Develop procedures and plans for responding to critical space debris incidents or uncontrolled re-entries by establishing communication protocols and response strategies for satellite operators in case of emergencies.

I. Introduction

This initiative presents a comprehensive approach to address the challenges posed by space debris, emphasizing the need for an effective management system to predict and analyze collision risks in the vast expanse of space. The primary objective centers around the creation of a sophisticated database, elevating the management of space debris data to new heights.

The core functionality of this system revolves around the real-time tracking and predictive analysis of space debris movement, a critical aspect in preventing potential collisions with operational satellites and spacecraft. At the heart of the proposal lies the design of an advanced database system, meticulously crafted to efficiently store and retrieve crucial information concerning space debris. What sets this system apart is its ability to seamlessly integrate conventional database methodologies with cutting-edge technologies, presenting an optimized solution capable of accommodating real-time data updates and ensuring prompt information retrieval.

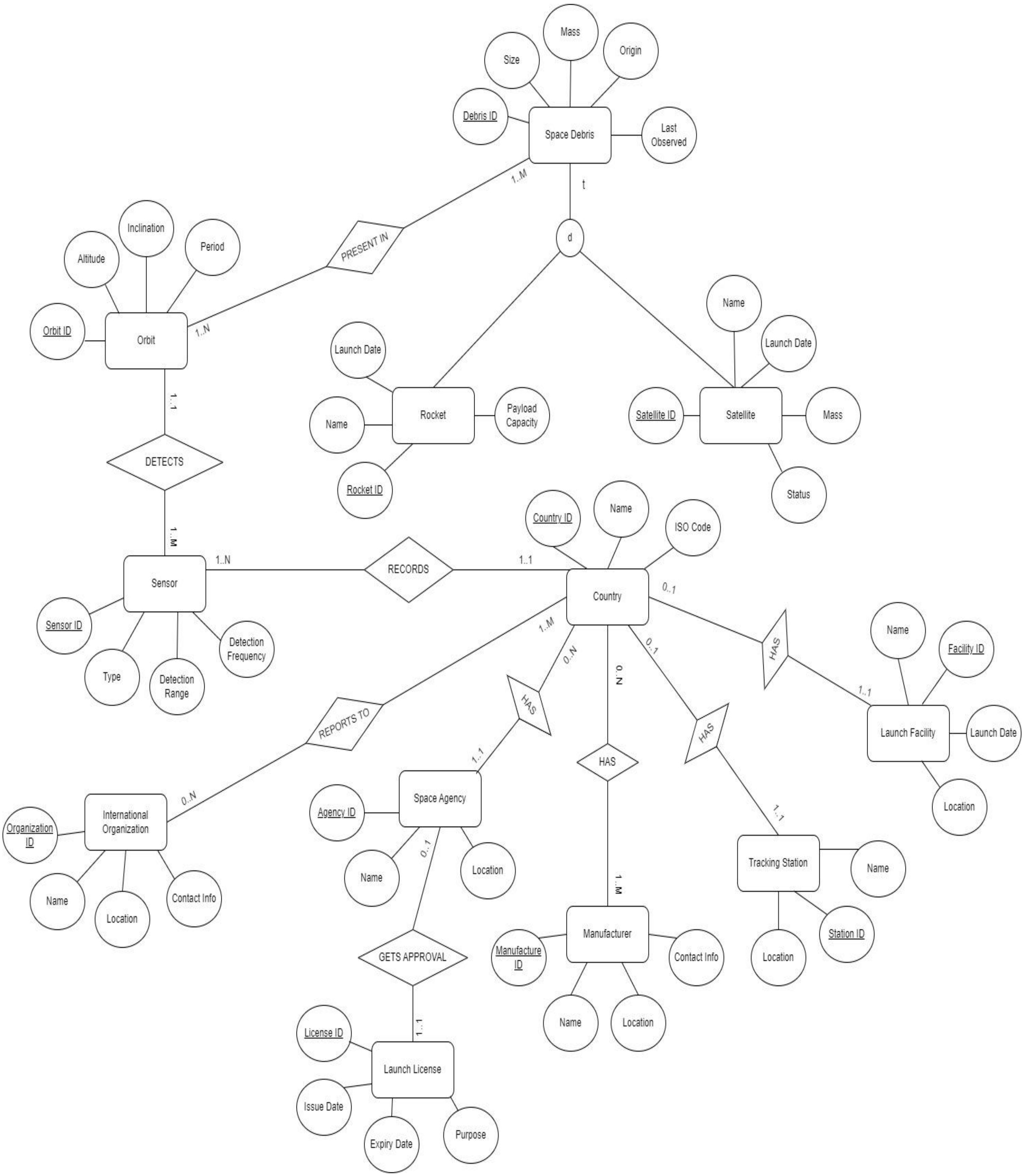
The technological backbone of this project is tailored for predicting the trajectory and behavior of space debris. By combining a comprehensive space debris database with state-of-the-art predictive analytics, this initiative strives to enhance the precision and reliability of collision risk assessments.

To ensure practical usability and accessibility, the project places a strong emphasis on user-friendliness, providing a well-designed interface tailored for mission planners and space agencies. This interface aims to empower entities with efficient tools, fostering better-informed choices in the intricate realm of space exploration and satellite operations.

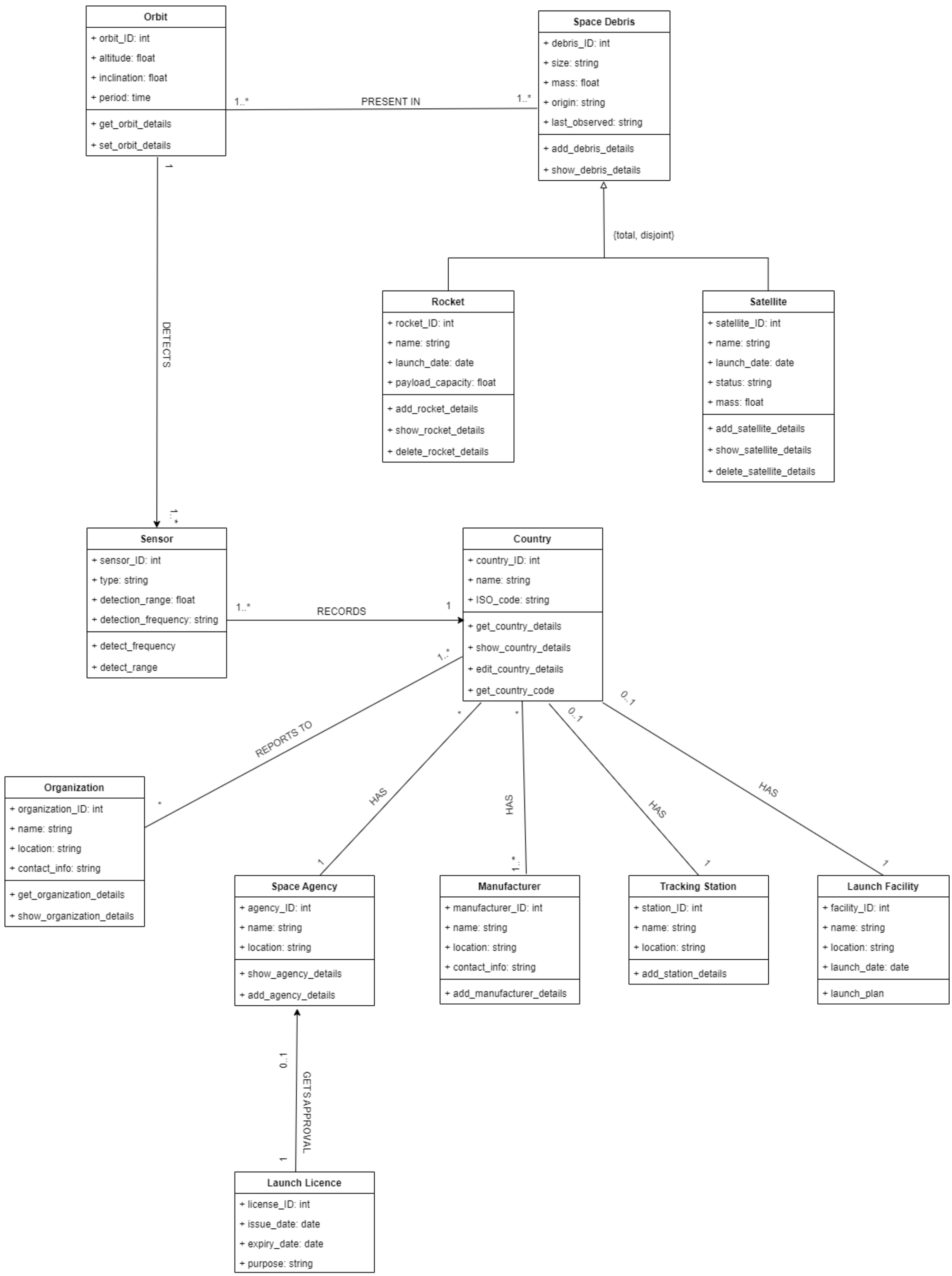
In the broader context, this undertaking emerges as a significant contribution to the sustainability of space activities. By mitigating collision risks and fortifying data management capabilities, this project signifies a pivotal step toward creating a safer and more sustainable orbital environment. Through its innovative solutions, the initiative addresses the pressing challenges of space debris, positioning itself as a possibility of advancement in the pursuit of a secure and resilient future for space exploration.

II. Conceptual Data Modeling

1. EER Diagram



2. UML Diagram



III. Mapping Conceptual Model to Relational Model

Primary Key- Underlined
Foreign Key- *Italicized*

orbit(orbit_ID,altitude,inclination,period)
PK- orbit ID, not null

space_debris(debris_ID,size,SD_mass,origin,last_observed)
PK- debris ID, not null

present_in(*orbit_ID*,*debris_ID*)
PK & FK - orbit ID refers to Orbit, not null
PK & FK - Debris ID refers to space_debris, not null

rocket(*rocket_debris_ID*,R_name,R_launchdate,payload_capacity)
PK - Rocket_debris refers to space_debris

satellite(*satellite_debris_ID*,S_name,status,S_launchdate,S_mass)
PK - Satellite_debris refers to space_debris

sensor(sensor_ID,type,detection_range,detection_frequency,*orbit_ID*,*country_ID*)
PK - sensor_ID refers to sensor, not null
FK - orbit ID refers to orbit, not null
FK - country ID refers to country, not null

country(country_ID,C_name,ISO,*agency_ID*,*station_ID*,*facility_ID*)
PK - country_ID refers to country, not null
FK - Agency ID refers to space agency and it's not null
FK - station ID refers to tracking station and it's not null
FK - facility ID refers to launch facility and it's not null

organization(organization_ID,O_name,O_location,O_contact)
PK - organization ID refers to organization

reports_to(*organization_ID*,*country_ID*)
PK & FK - Organization ID refers to organization and not null
PK & FK - country ID refers to country and not null

space_agency(agency_ID,A_name,A_location,*license_ID*)
PK - agency ID, not null
FK - license ID refers to launch license and it's not null

manufacturer(manufacturer_ID,M_name,M_location,M_contact)
PK - manufacturer ID, not null

country_manufacturer(*country_ID*,*manufacturer_ID*)
PK & FK - country ID refers to country, not null
PK & FK - manufacturer ID refers to manufacturer, not null

launch_license(license_ID,issue_date,expiry_date,purpose)
PK - license ID, not null

tracking_station(<u>station_ID</u>,T_name,T_location)
PK - station ID, not null

launch_facility(<u>facility_ID</u>,F_name,F_launchdate,F_location)
PK - facility ID, not null

IV. Implementation of Relation Model via MySQL and NoSQL

MySQL Implementation:

The database was created in MySQL and the following queries were performed:

Query 1: Simple Query: Retrieve the Rocket IDs, names, launch dates, and the corresponding launch facility details for rockets launched after January 1, 2021.

SELECT R.ROCKET_ID, R.R_NAME,
R.R_LAUNCH_DATE, LF.FACILITY_ID, LF.F_NAME
FROM ROCKET R
JOIN LAUNCH_FACILITY LF
ON R.R_LAUNCH_DATE = LF.F_LAUNCHED_DATE
WHERE R.R_LAUNCH_DATE > '2021-01-01';

ROCKET_ID	R_NAME	R_LAUNCH_DATE	FACILITY_ID	F_NAME
18	Angara	2021-02-19	LF0067	Orion Spaceport

Query 2: Aggregate Query: Retrieve the total mass of space debris launched by each rocket.

SELECT R.ROCKET_ID, R.R_NAME, SD.DEBRIS_ID,
SUM(SD.D_MASS) AS TOTAL_D_MASS
FROM ROCKET R, SPACE_DEBRIS SD, PRESENT_IN P
WHERE R.ROCKET_ID = P.ROCKET_ID
AND P.DEBRIS_ID = SD.DEBRIS_ID
GROUP BY R.ROCKET_ID, R.R_NAME,
SD.DEBRIS_ID;

ROCKET_ID	R_NAME	DEBRIS_ID	TOTAL_D_MASS
1	Falcon 9	100	50627782.50
2	Atlas V	101	76311808.23
3	Delta IV	102	32977925.00
4	Soyuz	103	21346950.03
5	Long March 5	104	79321433.33
6	Proton-M	105	80919564.80
7	Ariane 5	106	17986562.98
8	H-IIA	107	68251580.56
9	GSLV Mk III	108	93130006.20
10	Vega	109	80973187.00

Query 3: Inner and Outer Join Query

INNER JOIN: Retrieve the details of satellites, including their names, launch dates, and the corresponding rocket names used for their launches.

SELECT S.SATELLITE_ID, S.S_NAME,
S.S_STATUS, S.S_LAUNCHED_DATE, R.R_NAME,
R.R_PAYLOAD_CAPACITY
FROM SATELLITE S
INNER JOIN ROCKET R ON
S.S_LAUNCHED_DATE = R.R_LAUNCH_DATE;

SATELLITE_ID	S_NAME	S_STATUS	S_LAUNCHED_DATE	R_NAME	R_PAYLOAD_CAPACITY
SAT009	Satellite 9	operational	2022-03-08	Falcon 9	96888794.44
SAT005	Satellite 5	battery low	2022-03-08	Falcon 9	96888794.44
SAT004	Satellite 4	active	2022-03-08	Falcon 9	96888794.44
SAT002	Satellite 2	active	2022-03-08	Falcon 9	96888794.44
SAT025	Satellite 25	system reboot	2020-08-21	Proton-M	23186115.99
SAT017	Satellite 17	testing	2020-08-21	Proton-M	23186115.99

OUTER JOIN: Retrieve the average size of space debris and the launch facility names where each debris was last observed.

SELECT D.DEBRIS_ID, AVG(D.D_SIZE)
AS AVERAGE_D_SIZE, LF.F_NAME
FROM SPACE_DEBRIS D
LEFT JOIN LAUNCH_FACILITY LF ON
D.D_LAST_OBSERVED_DATE =
LF.F_LAUNCHED_DATE
GROUP BY LF.F_NAME, D.DEBRIS_ID;

DEBRIS_ID	AVERAGE_D_SIZE	F_NAME
100	59676192.110000	NULL
101	24162452.760000	NULL
102	60907255.530000	NULL
103	85070511.790000	Zodiac Spaceport
103	85070511.790000	Cosmic Launch Center
103	85070511.790000	Infinity Launch Center
103	85070511.790000	Saturn Spaceport
103	85070511.790000	Celestial Launch Center
103	85070511.790000	Starlight Launch Site
103	85070511.790000	Alpha Spaceport
104	95575034.800000	NULL
105	4121951.560000	NULL
106	32408910.150000	NULL
107	19898032.550000	NULL
108	1714102.090000	Eclipse Spaceport
109	53400283.140000	NULL
110	22338795.070000	NULL

Query 4: Nested Query: Find the launch facilities that have been used to launch rockets with a payload capacity greater than the average payload capacity of all rockets.

SELECT LF.FACILITY_ID, LF.F_NAME,
LF.F_LAUNCHED_DATE, LF.F_LOCATION, R.R_NAME
FROM LAUNCH_FACILITY LF, ROCKET R
WHERE EXISTS (SELECT *
FROM ROCKET R
WHERE R.R_PAYLOAD_CAPACITY > (
SELECT AVG(R_PAYLOAD_CAPACITY)
FROM ROCKET)
AND R.R_LAUNCH_DATE = LF.F_LAUNCHED_DATE);

FACILITY_ID	F_NAME	F_LAUNCHED_DATE	F_LOCATION	R_NAME
LF0067	Orion Spaceport	2021-02-19	United States	Falcon 9
LF0067	Orion Spaceport	2021-02-19	United States	Atlas V
LF0067	Orion Spaceport	2021-02-19	United States	Delta IV
LF0067	Orion Spaceport	2021-02-19	United States	Soyuz
LF0067	Orion Spaceport	2021-02-19	United States	Long March 5
LF0067	Orion Spaceport	2021-02-19	United States	Proton-M
LF0067	Orion Spaceport	2021-02-19	United States	Ariane 5
LF0067	Orion Spaceport	2021-02-19	United States	H-IIA
LF0067	Orion Spaceport	2021-02-19	United States	GSLV Mk III
LF0067	Orion Spaceport	2021-02-19	United States	Vega

Query 5: Correlated Query: Retrieve the debris with a size larger than the average size of debris originating from the same country as either the rocket named 'Falcon 9' or 'Starship'.

SELECT D.DEBRIS_ID, R.ROCKET_ID, D.D_SIZE,
D.D_MASS, R.R_NAME
FROM SPACE_DEBRIS D, ROCKET R
WHERE D.D_SIZE > (SELECT AVG(D2.D_SIZE)
FROM SPACE_DEBRIS D2
INNER JOIN PRESENT_IN P
ON D2.DEBRIS_ID = P.DEBRIS_ID
INNER JOIN ROCKET R
ON P.ROCKET_ID = R.ROCKET_ID
WHERE (R.R_NAME = 'Falcon 9'
OR R.R_NAME = 'Starship')
AND D2.D_ORIGIN = D.D_ORIGIN);

DEBRIS_ID	ROCKET_ID	D_SIZE	D_MASS	R_NAME
100	30	59676192.11	50627782.50	CZ-4B
100	29	59676192.11	50627782.50	CZ-6
100	28	59676192.11	50627782.50	CZ-2F
100	27	59676192.11	50627782.50	CZ-11
100	26	59676192.11	50627782.50	CZ-7
100	25	59676192.11	50627782.50	CZ-5
100	24	59676192.11	50627782.50	CZ-3B
100	23	59676192.11	50627782.50	Zenit
100	22	59676192.11	50627782.50	GSLV Mk II
100	21	59676192.11	50627782.50	SLS

Query 6: >=ALL/>ANY/: Retrieve the names of rockets with a payload capacity greater than or equal to the maximum payload capacity of any rocket launched in 2021.

SELECT ROCKET_ID, R_NAME
FROM ROCKET
WHERE R_PAYLOAD_CAPACITY >= ALL (
SELECT MAX(R_PAYLOAD_CAPACITY)
FROM ROCKET
WHERE YEAR(R_LAUNCH_DATE) = 2020);

ROCKET_ID	R_NAME
1	Falcon 9
3	Delta IV
4	Soyuz
7	Ariane 5
9	GSLV Mk III
11	Antares
17	LauncherOne
18	Angara
23	Zenit
24	CZ-3B
27	CZ-11
28	CZ-2F
NULL	NULL

Query 7: Exists and Not Exists Query

EXISTS: Rockets observed by sensors.

SELECT R.ROCKET_ID, R.R_NAME
FROM ROCKET R
WHERE EXISTS (
SELECT 1
FROM PRESENT_IN P
JOIN SENSOR S ON P.ROCKET_ID = R.ROCKET_ID
WHERE S.SEN_DETECTION_FREQUENCY IS NOT NULL);

ROCKET_ID	R_NAME
1	Falcon 9
2	Atlas V
3	Delta IV
4	Soyuz
5	Long March 5
6	Proton-M
7	Ariane 5
8	H-IIA
9	GSLV Mk III
10	Vega

NOT EXISTS: Rockets without Launch Licenses

SELECT D.DEBRIS_ID, D.D_SIZE, D.D_MASS,
D.D_LAST_OBSERVED_DATE, D.D_ORIGIN
FROM SPACE_DEBRIS D
WHERE NOT EXISTS (SELECT 1FROM PRESENT_IN PI
WHERE PI.DEBRIS_ID = D.DEBRIS_ID
) OR D.D_LAST_OBSERVED_DATE < '2021-01-01';

DEBRIS_ID	D_SIZE	D_MASS	D_LAST_OBSERVED_DATE	D_ORIGIN
104	95575034.80	79321433.33	2020-12-25	satellite breakups
111	8732048.57	91455960.28	2020-12-06	lost equipments
124	87111456.09	44822320.03	2020-11-13	fragmentation
130	45446815.47	6301933.82	2021-04-01	launch vehicle debris
NULL	NULL	NULL	NULL	NULL

Query 8: Set Operations (Union) Query: Retrieve rockets launched by Long March 5 and retrieve satellites launched in France.

```
SELECT R_NAME AS NEW_NAME, R_LAUNCH_DATE
AS NEW_LAUNCH_DATE FROM ROCKET WHERE R_NAME
IN('Long March 5', 'Proton M','Soyuz','Electron','Minotaur','Pegasus')
UNION SELECT S_NAME, S_LAUNCHED_DATE
FROM SATELLITE WHERE SATELLITE_ID IN (
SELECT DISTINCT S.SATELLITE_ID FROM SATELLITE S
INNER JOIN LAUNCH_FACILITY F
ON S.S_LAUNCHED_DATE = F.F_LAUNCHED_DATE
WHERE F.F_LOCATION IN ('Turkey', 'France', 'Russia', 'China',' Germany', 'Brazil'))
ORDER BY NEW_LAUNCH_DATE;
```

Result Grid | Filter Rows: | Export:

NEW_NAME	NEW_LAUNCH_DATE
Minotaur	2020-11-20
Satellite 10	2021-02-24
Pegasus	2021-04-17
Long March 5	2021-08-14
Electron	2022-03-16
Soyuz	2022-04-21

Query 9: Retrieve the satellites launched by a specific space agency

```
SELECT S.*
FROM SATELLITE S
WHERE S.SATELLITE_ID IN (
 SELECT SATELLITE_ID
 FROM LAUNCH_LICENSE LL
 WHERE LL.LICENSE_ID IN (
 SELECT LICENSE_ID
 FROM SPACE_AGENCY SA
 WHERE SA.AGENCY_ID = 'AG001' ) );
```

SATELLITE_ID	S_NAME	S_STATUS	S_LAUNCHED_DATE	S_MASS
SAT001	Satellite 1	inactive	2021-02-13	29617702.56
SAT002	Satellite 2	active	2022-03-08	62282918.05
SAT003	Satellite 3	communication failure	2021-03-13	98297509.44
SAT004	Satellite 4	active	2022-03-08	69730844.27
SAT005	Satellite 5	battery low	2022-03-08	37703441.25
SAT006	Satellite 6	under maintenance	2020-07-25	98905535.28
SAT007	Satellite 7	ready for launch	2020-05-20	91044577.52
SAT008	Satellite 8	operational	2020-08-31	42320812.25
SAT009	Satellite 9	operational	2022-03-08	60321428.50
SAT010	Satellite 10	malfunctioning	2021-02-24	99481309.74

NoSQL Implementation:

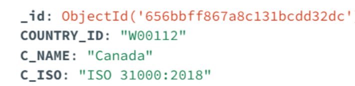
All the tables (country, country_manufacturer, launch_facility, luanch_license, manufacturer, orbit, organization, present_in, reports_to, rocket, satellite, sensor, space_agency, space_debris, tracking_station) created in MySQL has been imported into MongoDB Compass. The following NoSQL queries were done:

Simple Queries:

Query 1: Retrieve the document where the field's value "C_NAME" is "Canada" in the country collection.

db.countries.findOne({ "C_NAME": "Canada" })

In JSON format: { "C_NAME": "Canada" }



Query 2: Retrieve all documents from the "country" collection.

In JSON format: [{ "$match": {} }]

More Complex Queries:

Query 3: Retrieve the documents from the "country" collection where the country name (C_NAME) starts with "Uni" in a case-insensitive manner.

[{ "$match": { "C_NAME": { "$regex": "^Uni", "$options": "i" } } }]

Query 4: Retrieve the 'C_NAME' and 'C_ISO' fields for documents where the 'C_NAME' is either 'Canada' or 'Mexico'.

[{ "$match": { "C_NAME": { "$in": ["Canada", "Mexico"] } } }, { "$project": { "C_NAME": 1, "C_ISO": 1 } }]

Aggregate Queries:

Query 5: How many documents (records) are there in the country collection for each unique value of 'C_ISO'?

[{ "$group": { "_id": "$C_ISO", "count": { "$sum": 1 } } }]

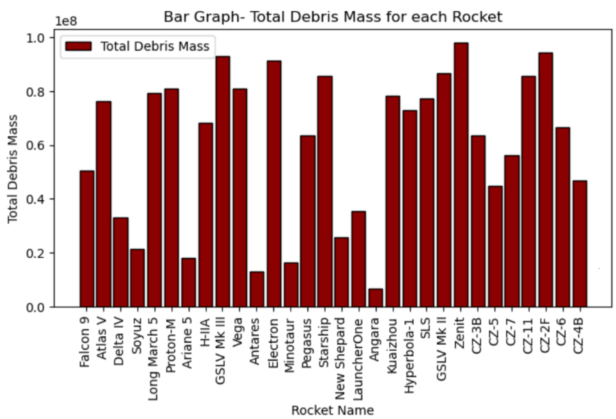
Query 6: Retrieve a summary of countries grouped by their ISO standard code, including the list of countries and the total count for each ISO code from the 'country' collection.

[{ "$group": { "_id": "$C_ISO", "countries": { "$push": "$C_NAME" } } }, { "$project": { "ISO_Standard": "$_id", "Countries": "$countries", "Total": { "$size": "$countries" } } }]

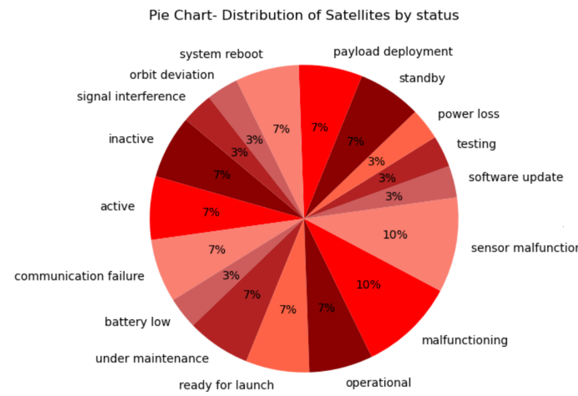
V. Database Access via Python

Python is used to access the database, and the data that was analyzed visualization is displayed below. Mysql.connector is used to connect MySQL to Python, and ran the query with the command "fetchall," then used the panda's package to turn the list into a data frame, and finally use matplotlib to plot the graphs for the analytics.

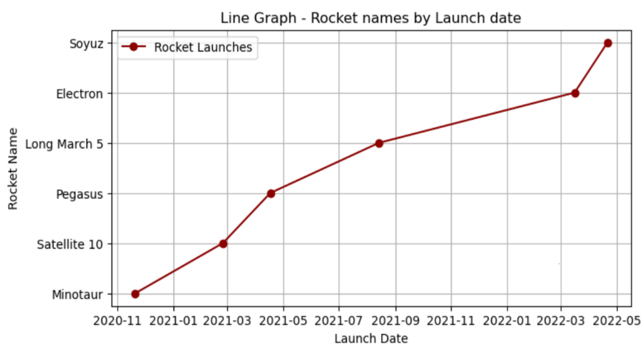
Graph1: Bar Graph - Total Debris Mass for each Rocket



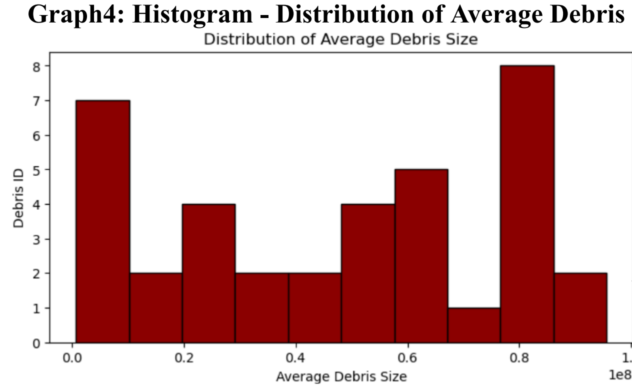
Graph2: Pie Chart - Distribution of Satellites by status



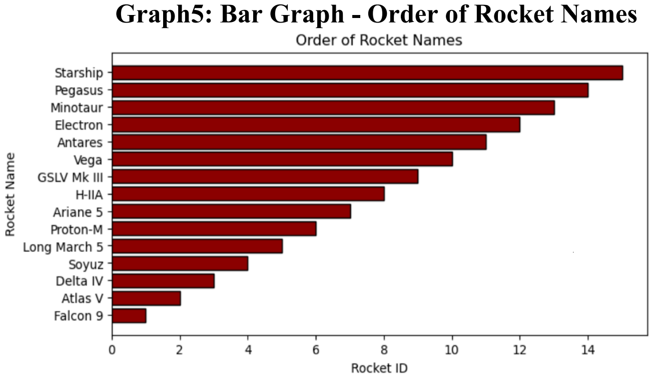
Graph3: Line Graph - Rocket names by Launch Date



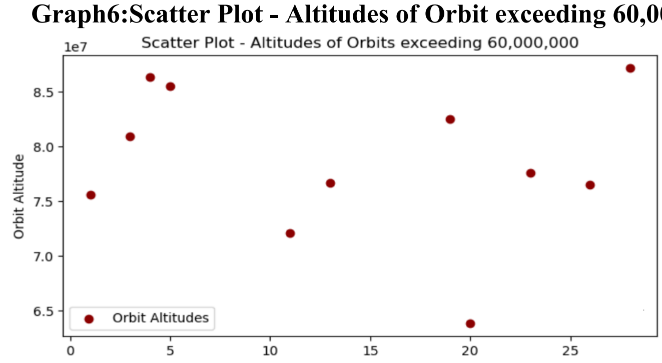
Graph4: Histogram - Distribution of Average Debris Size



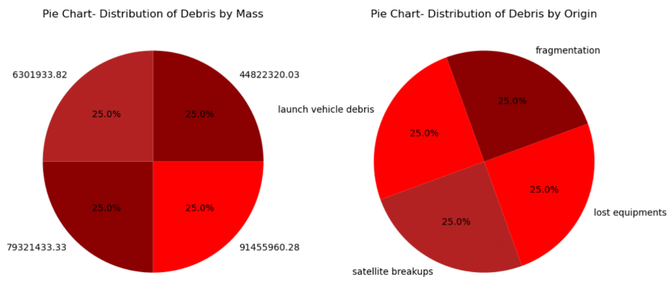
Graph5: Bar Graph - Order of Rocket Names



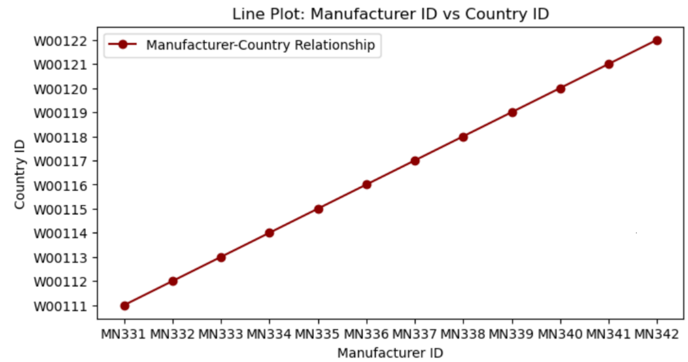
Graph6:Scatter Plot - Altitudes of Orbit exceeding 60,000,000

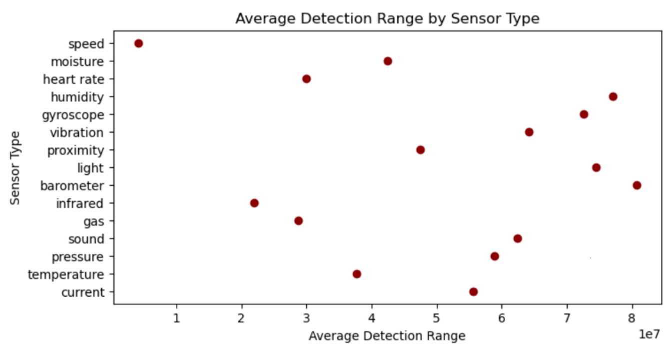


Graph7: Pie Chart - Distribution of Debris by Mass & Distribution of Debris by Origin



Graph8: Line Plot - Manufacturer ID vs Country ID





Graph9: Scatter Plot - Average Detection Range by Sensor Type

VI. Summary and Recommendation

The project investigates the management of space debris and aims to mitigate the risks associated with orbital clutter. It introduces an advanced data-driven framework combining database systems and deep learning to enhance the surveillance and analysis of space debris. This includes the development of a comprehensive database for efficient data storage and retrieval, using machine learning algorithms for trajectory prediction, and integrating sensor networks for real-time tracking.
The below recommendations aim to bolster the project's impact on safeguarding space infrastructure and contribute to the sustainable exploration and use of outer space.

1. Expand Data Collection: Broaden the scope of data collection to include more diverse sources of space debris information for a more comprehensive database.

2. Improve Predictive Models: Continuously refine the predictive models used for debris trajectory forecasting to increase their accuracy and reliability.

3. Enhance Real-time Capabilities: Develop more sophisticated real-time data processing capabilities to enable quicker response times to potential collisions.

4. Strengthen International Collaboration: Foster stronger collaboration with international space agencies to share data and insights, which is crucial for global space safety.

5. Invest in Technology Upgrades: Allocate resources for upgrading technological infrastructure to support the increasing data demands of the project.

6. Conduct Regular System Audits: Regularly audit the system to ensure data integrity and the effective functioning of all components.

7. Policy Advocacy: Advocate for policies that support the sustainable use of space and the development of regulations for space debris management.