SPACE DEBRIS MANAGEMENT

PRESENTATION BY,

SNEHA MANJUNATH CHAKRABHAVI
JANANI KARTHIKEYAN
GROUP 18

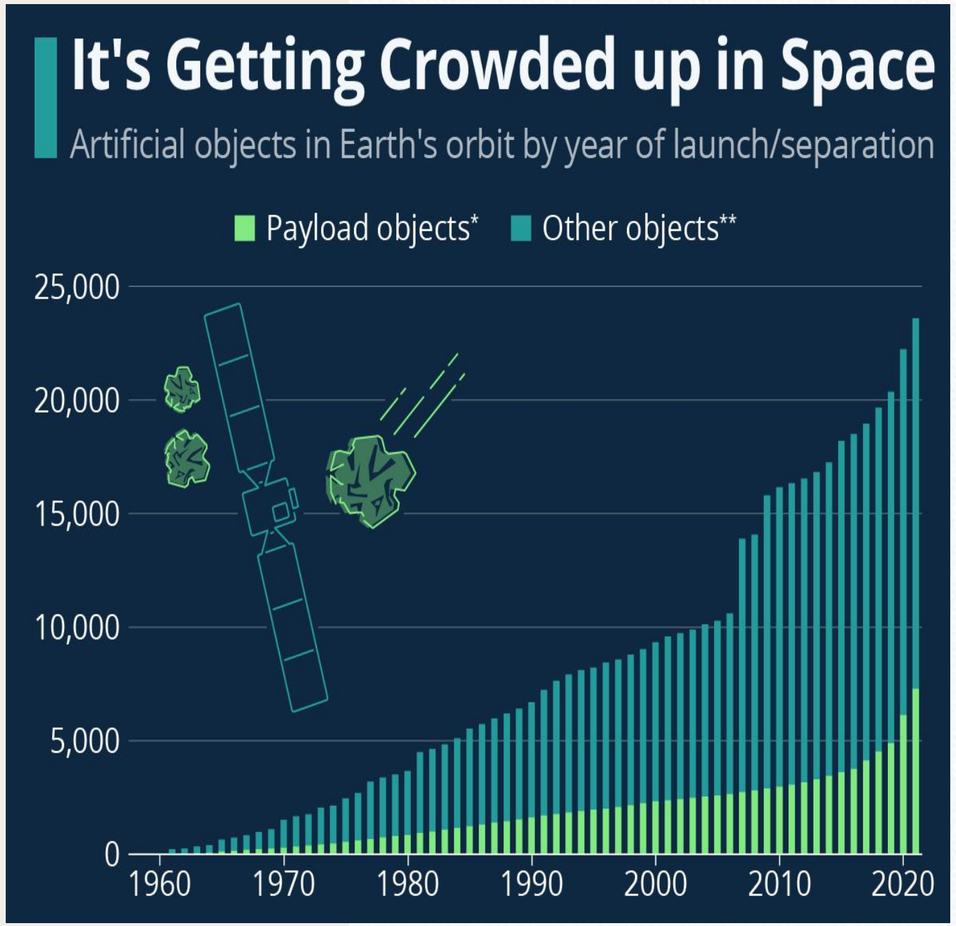
TABLE OF CONTENTS

INTRODUCTION

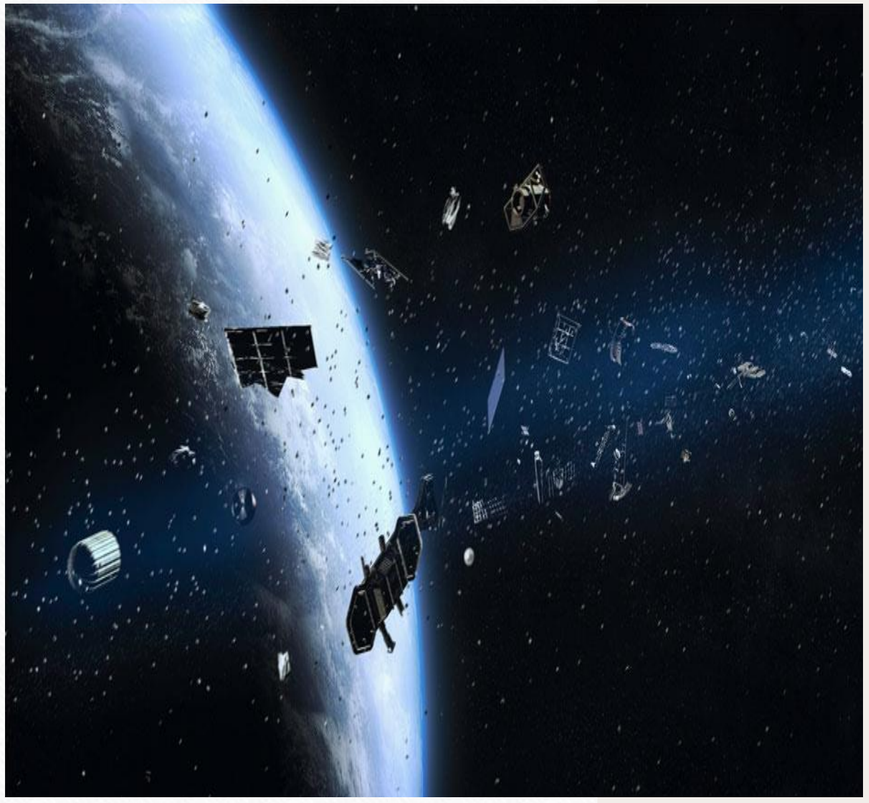
- Space debris management is a critical but often under-addressed issue that requires immediate action.
- Collisions in space can lead to abundant debris, a scenario known as the Kessler Syndrome, which can exponentially increase the likelihood of further collisions, potentially making certain orbits unusable for generations.
- Space Debris Management focuses on the development of information-centric strategies that leverage data-driven methodologies to mitigate the risks associated with space debris.
- By combining advanced database systems, we aim to enhance the analysis, tracking, and prediction of space debris trajectories and their origin.
- Establish a robust framework for the collection, processing, and management of space debris data, with real-time tracking and prediction capabilities.
- There is a pressing need for comprehensive strategies to mitigate space debris and manage the orbital environment.

BACKGROUND INFORMATION

- **Definition:** Space debris, also called space junk or orbital debris, consists of old satellites, discarded and unnecessary parts of rockets, and non-functional objects orbiting Earth.
- **Composition:** Includes spent rocket stages, dysfunctional satellites, fragments resulting from disintegration, and debris generated from diverse space missions and activities.
- **Speed:** Space debris travels at high speeds, making even small pieces hazardous due to the potential for collisions.
- **Risk Factors:** The accumulation of space debris presents risks to ongoing and future space missions, as collisions can cause damage to operational satellites and spacecraft.
- **Global Impact:** Space debris is a global concern, requiring international collaboration and responsible space practices to address the challenges associated with its presence in Earth's orbit.
- **Mitigation Strategies:** Efforts include tracking and monitoring debris, developing strategies for debris removal, and promoting responsible space practices to minimize the creation of additional debris.



BUSINESS PROBLEM



Critical challenge: Increased risk of collisions and orbital mishaps due to growing space debris.

Business Dilemma:

- Absence of an efficient real-time system for space debris data.
- Challenges in collecting, processing, managing, and analyzing space debris data.

Data-Driven Approaches:

- Advocates the use of data analysis, orbital surveillance, and predictive modeling.
- Highlights the need for methodologies driven by accurate and timely data.

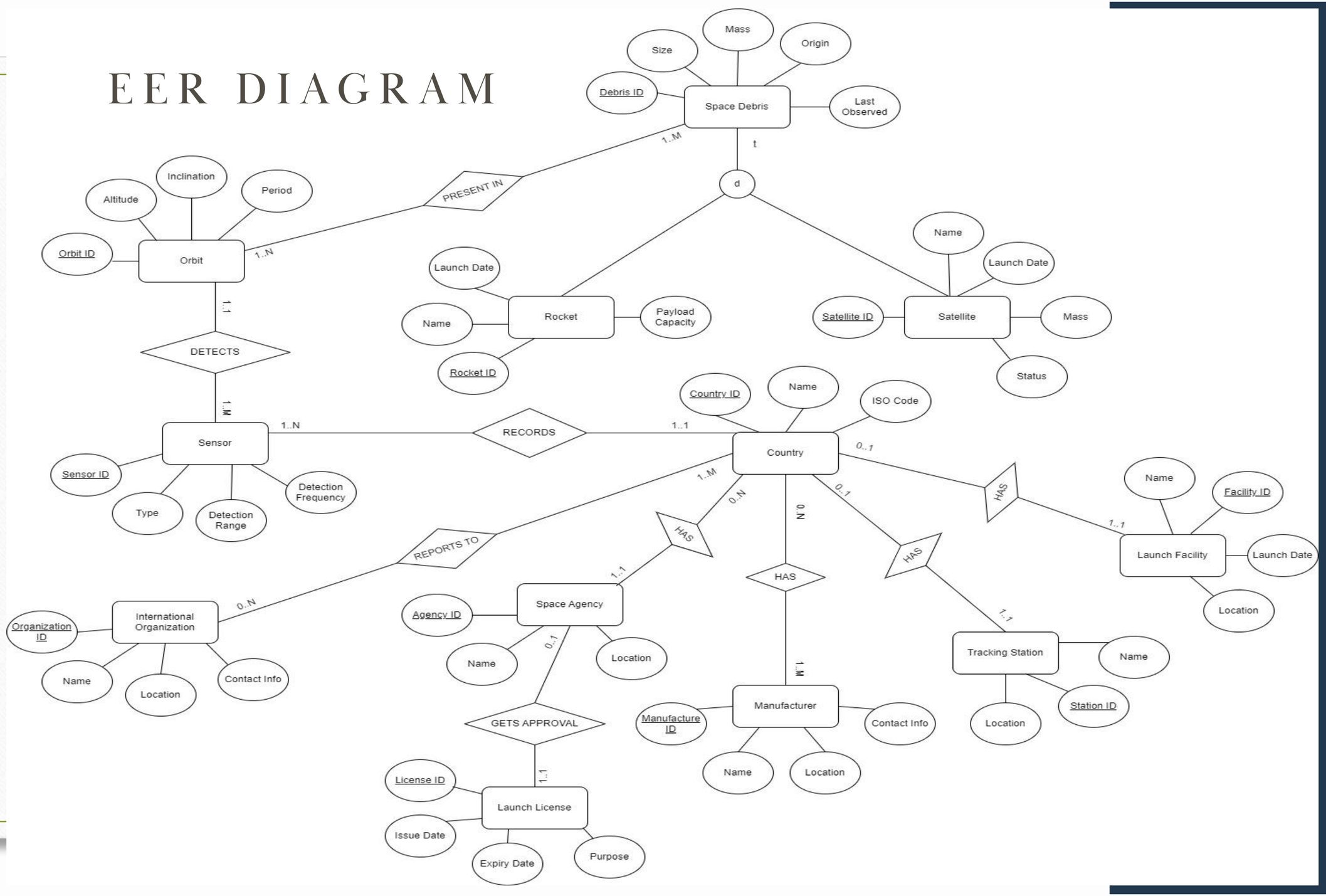
Integrated Approach Needed:

- Calls for an integrated approach to address the challenge.
- Advocates the fusion of advanced database systems and deep learning techniques like MySQL, Python, etc.

Enhancing Surveillance and Analysis:

- Aims to significantly enhance the efficiency and accuracy of space debris surveillance and analysis.
- Emphasizes the role of advanced technologies in improving space debris management.

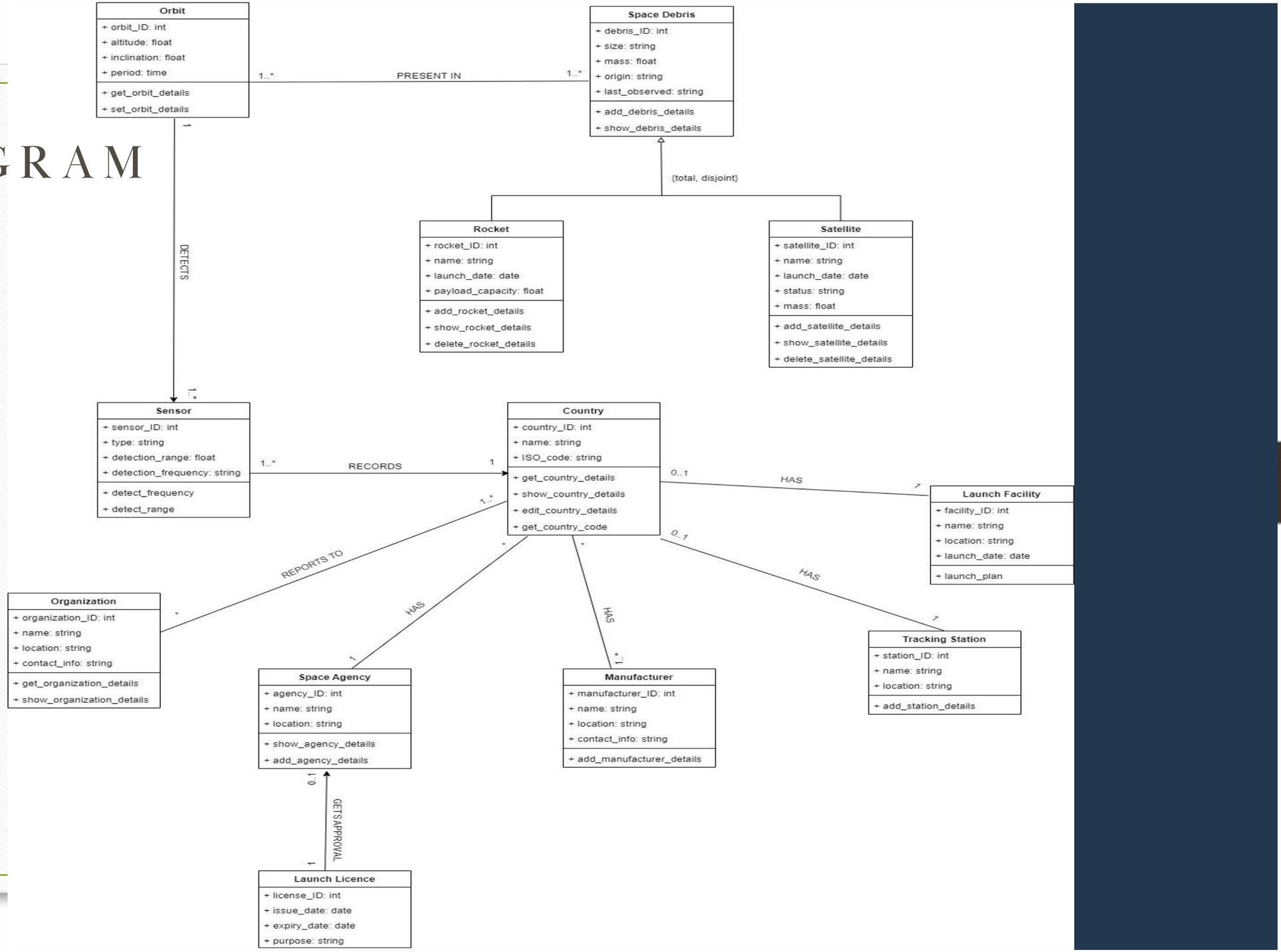
EER DIAGRAM


Mass
Size
Origin
Debris ID
Space Debris
Last Observed
1..M
t
PRESENT IN
d
Inclination
Altitude
Period
Orbit ID
Orbit
1..N
Launch Date
Name
Rocket
Payload Capacity
Rocket ID
Name
Launch Date
Satellite ID
Satellite
Mass
Status
1..1
DETECTS
1..M
Country ID
Name
ISO Code
Sensor
1..N
RECORDS
1..1
Country
0..1
Sensor ID
Type
Detection Range
Detection Frequency
1..M
0..N
0..N
0..1
HAS
HAS
HAS
HAS
Name
Facility ID
1..1
Launch Facility
Launch Date
Location
REPORTS TO
0..N
1..1
Organization ID
International Organization
Name
Location
Contact Info
Agency ID
Space Agency
Name
0..1
Location
1..1
Tracking Station
Name
Station ID
Location
Manufacture ID
Manufacturer
Name
Location
Contact Info
1..M
GETS APPROVAL
1..1
License ID
Launch License
Issue Date
Expiry Date
Purpose

EER DIAGRAM ENTITIES AND ATTRIBUTES

1. ORBIT Identified by an Orbit ID, it has attributes such as Altitude, Inclination, and Period.
2. SPACE_DEBRIS Identified by a Debris ID, it has attributes like Size, Origin, and the Last Observed time
3. ROCKET Has a Rocket ID, Name, Launch Date, and Payload Capacity.
4. SATELLITE Identified by Satellite ID, it holds details such as Name, Mass, and Status
5. SENSOR Each sensor, with a unique Sensor ID, can detect objects and is characterized by its Type, Detection Range, and Detection Frequency
6. COUNTRY Characterized by Country ID, Name, and ISO.
7. ORGANIZATION Identified by Organization ID, includes Name, Location, and Contact Info
8. SPACE_AGENCY *: Identified by Agency ID, including Name and Location.
9. MANUFACTURER Identified by Manufacturer ID, including Name, Location, and Contact Info.
10. LAUNCH_LICENSE Identified by a License ID, it has an Issue Date, Expiry Date, and Purpose
11. TRACKING_STATION Identified by Station ID, including Name and Location.
12. LAUNCH_FACILITY Identified by a Facility ID, including Name and Location.

UML DIAGRAM


Orbit
+ orbit_ID: int
+ altitude: float
+ inclination: float
+ period: time
+ get_orbit_details
+ set_orbit_details
1..*
PRESENT IN
1..*
Space Debris
+ debris_ID: int
+ size: string
+ mass: float
+ origin: string
+ last_observed: string
+ add_debris_details
+ show_debris_details
{total, disjoint}
Rocket
+ rocket_ID: int
+ name: string
+ launch_date: date
+ payload_capacity: float
+ add_rocket_details
+ show_rocket_details
+ delete_rocket_details
Satellite
+ satellite_ID: int
+ name: string
+ launch_date: date
+ status: string
+ mass: float
+ add_satellite_details
+ show_satellite_details
+ delete_satellite_details
1
DETECTS
1..*
Sensor
+ sensor_ID: int
+ type: string
+ detection_range: float
+ detection_frequency: string
+ detect_frequency
+ detect_range
1..*
RECORDS
1
Country
+ country_ID: int
+ name: string
+ ISO_code: string
+ get_country_details
+ show_country_details
+ edit_country_details
+ get_country_code
0..1
HAS
1
Launch Facility
+ facility_ID: int
+ name: string
+ location: string
+ launch_date: date
+ launch_plan
1..*
REPORTS TO
*
Organization
+ organization_ID: int
+ name: string
+ location: string
+ contact_info: string
+ get_organization_details
+ show_organization_details
*
HAS
1
Space Agency
+ agency_ID: int
+ name: string
+ location: string
+ show_agency_details
+ add_agency_details
*
HAS
1..*
Manufacturer
+ manufacturer_ID: int
+ name: string
+ location: string
+ contact_info: string
+ add_manufacturer_details
0..1
HAS
1
Tracking Station
+ station_ID: int
+ name: string
+ location: string
+ add_station_details
0..1
GETS APPROVAL
1
Launch Licence
+ license_ID: int
+ issue_date: date
+ expiry_date: date
+ purpose: string

RELATIONAL MODEL

1. ORBIT(ORBIT_ID,ALTITUDE,INCLINATION,PERIOD)

Primary Key: orbitID

Foreign Key: NA

2. SPACE_DEBRIS(DEBRIS_ID,SIZE,SD_MASS,ORIGIN,LAST_OBSERVED)

Primary Key: debrisID

Foreign Key: NA

3. PRESENT_IN(*ORBIT_ID*,*DEBRIS_ID*)

Primary Key: orbitID refers to Orbit (Not Null) and debrisID refers to space_debris (Not Null)

Foreign Key: NA

4. ROCKET(*ROCKET_DEBRIS_ID*,R_NAME,R_LAUNCHDATE,PAYLOAD_CAPACITY)

Primary Key: rocket_debris refers to space_debris

Foreign Key: NA

5. SATELLITE(*SATELLITE_DEBRIS_ID*,S_NAME,STATUS,S_LAUNCHDATE,S_MASS)

Primary Key: satellite_debris refers to space_debris

Foreign Key: NA

6. SENSOR(SENSOR_ID,TYPE,DETECTION_RANGE,DETECTION_FREQUENCY,*ORBIT_ID*,*COUNTRY_ID*)

Primary Key: sensorID refers to sensor

Foreign Key: orbitID refers to orbit (Not Null) and countryID refers to country (Not Null)

7. COUNTRY(COUNTRY_ID,C_NAME,ISO,*AGENCY_ID*,*STATION_ID*,*FACILITY_ID*)

Primary Key: countryID refers to country

Foreign Key: agencyID refers to space agency (Not Null), stationID refers to tracking station (Not Null) and facilityID refers to launch facility (Not Null)

8. ORGANIZATION(ORGANIZATION_ID,O_NAME,O_LOCATION,O_CONTACT)

Primary Key: organizationID refers to organization

Foreign Key: NA

9. REPORTS_TO(*ORGANIZATION_ID*,*COUNTRY_ID*)

Primary Key: organizationID refers to organization and countryID refers to country

Foreign Key: NA

10. SPACE_AGENCY(AGENCY_ID,A_NAME,A_LOCATION,*LICENSE_ID*)

Primary Key: agencyID

Foreign Key: licenseID refers to launch license (Not Null)

11. MANUFACTURER(MANUFACTURER_ID,M_NAME,M_LOCATION,M_CONTACT)

Primary Key: manufacturerID

Foreign Key: NA

12. COUNTRY_MANUFACTURER(*COUNTRY_ID*,*MANUFACTURER_ID*)

Primary Key: countryID refers to country and manufacturerID refers to manufacturer

Foreign Key: NA

13. LAUNCH_LICENSE(LICENSE_ID,ISSUE_DATE,EXPIRY_DATE,PURPOSE)

Primary Key: licenseID

Foreign Key: NA

14. TRACKING_STATION(STATION_ID,T_NAME,T_LOCATION)

Primary Key: stationID

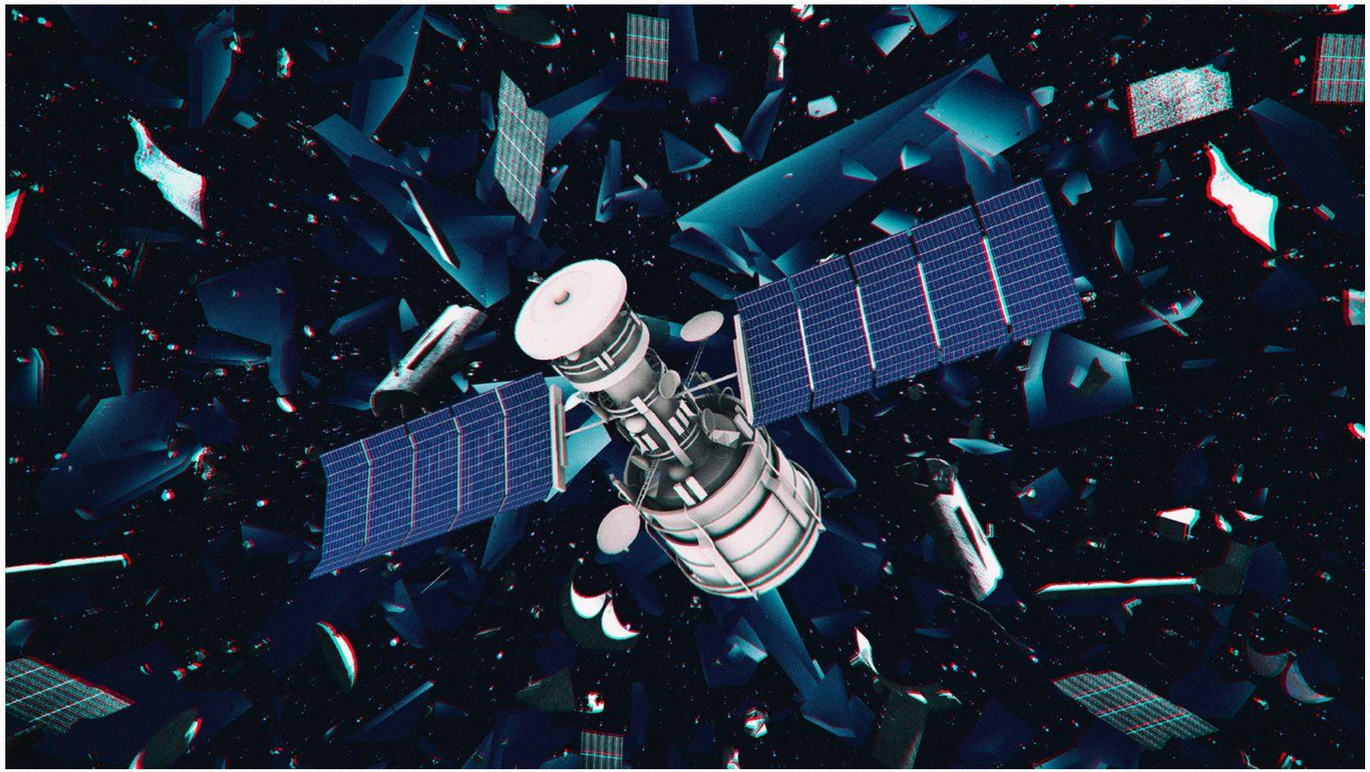
Foreign Key: NA

15. LAUNCH_FACILITY(FACILITY_ID,F_NAME,F_LAUNCHDATE,F_LOCATION)

Primary Key: facilityID

Foreign Key: NA

THANK YOU



Group 18

Janani Karthikeyan
+1-413-557-9761
karthikeyan.j@northeastern.edu

Sneha Manjunath
+1-857-891-3226
chakrabhavi.s@northeastern.edu