

Section 1:Final Version Of SRS

Software Requirements Specification

for

SPACE AGENCY DATABASE

Version 3.0 Final

Prepared by
Divyesh (201801086)
Janvi (201801178)
Skand(201801210)
Tulsi(201801180)

**Dhirubhai Ambani Institute of Information and
Communication Technology**

***Created on: 26th
September 2020 Last
Modified: 3rd
October 2020***

Introduction

1.1 Purpose

1. Our System will keep track of all old missions, current missions and future missions launched by our space agency ISRO.
2. Our system will keep track of satellites, its objectives, overall functionality and its real time position.
3. Also, we will keep track of important exoplanets which have the potential of earth-like-characteristics and potential life; which will aid in future exoplanetary missions.
4. Also all the important exoplanetary objects such as black holes, stars, big asteroids which are under observations because of their unique behaviour and their potential to impact the nearby galaxies in any way will also be documented in the system along with their detailed description.
5. We will also keep a functionality which enables scientists to automatically get notified about some new event in the exoplanetary objects specified above. This will include, asteroids coming very near to earth, supernova explosions, change in strength of blackholes and more.
6. Also, various experiments being conducted by rovers, satellites will also be mentioned in the system.
7. The system will be part wise accessible to ISRO personnel according to their clearance.

1.2 Intended Audience:

This SRS document is intended for database and website developers, the project manager, section managers and system testing team. This will also be available to read for ISRO scientists and researchers; as they can understand how the system works and contribute to this project as well.

What does this SRS have?

1. This SRS document contains the purpose, Overview of the working of the system, its functionality, operating environment, design and idea how to implement it.
2. It also contains detailed information about the system design comparing our system with some other already existing systems.
3. This system doesn't already exist but exists as different piecewise systems; Our goal will be to integrate different functionalities into one single system.

1.3 Reading Suggestions:

This document is divided into various sections:

1. Purpose and Overall Description: This is for all the people working on making this system and anyone new who wants to know about this project/system.
2. Overall Description: Will contain how the system functions, who will use this system, in which front end and back end will the system work and assumptions System Features.

1.4 Product Scope:

- Keep track of satellite health, positions and services.
- It should also keep track of stars, planets, black holes, moving stars, big asteroids and notify the scientists when some event occurs, or alert about the future collisions with earth or other entities.
- This system should be able keep track of their experiment and outcomes.
- When needs arise different agencies should be able to work-together to complete the mission by sharing information from this system.
- We are going to make a product which will carry data about ISRO missions name, launching data, launching vehicle and mission outcomes.
- This system can be used to share data to various private sector space agencies on a controlled basis by making them pay a fee for the same.
- Also, other space agencies can collaborate on similar objectives of some missions and save money and manpower.

2. Overall Description

2.1 Product Description

- The name of the project is “Space Agency Database”.
- The name of the product is “xyz”.
- This system will be a new product which is not a replacement of any other existing system instead will work alongside them. This won’t be an independent system.
- This system will depend on eportal of ISRO i.e, ISTRAC, etc.
- For data will be updated on the system only if the data on above portals is updated. For independent updation of data, this system will require access to ISRO servers; which right now is not an product objective.

- The tables distribution as follows:
 - A table for a list of satellites with a separate table for each satellite containing relevant information all in the schema “SAT”.
 - A table for current missions with each separate table having information about mission progress recorded at a regular time interval; all in schema “MISSIONS”.
 - Separate tables for black holes, stars, exoplanets and big asteroids with relevant information; all in schema “EXO”.
 - A table for keeping track of Near Earth Objects in schema “NEO”.

2.2 Existing Systems

1. https://solarsystem.nasa.gov/missions/?order=launch_date+desc&per_page=50&page=0&search=&fs=&fc=33&ft=&dp=&category=33
 • This system/website focuses on NASA's missions to study planets, moons, asteroids, comets, etc
 • It also has a filtering mechanism which has a type of satellite/ rover, location based filtering like; Kuiper Belt, Sun, Oort cloud, etc.
 • It also shows past, active and future missions.
2. <https://www.nasa.gov/open/data.html>
 • Just typing a keyword in the search box provides us with each of the links on the web available on that particular keyword.
 • It gives us plenty of news articles, nasa links related to it, links to videos and much more. It is very different from the system which we are making, but it gives us an idea about the front end of our database from a civilian point of view.
3. <https://directory.eoportal.org/web/eoportal/satellite-missions/space-agencies>
 This is a list of space agencies that own, fund or are involved in the missions. To see which missions each agency is associated with, click on the link. This includes information about the Space agency, their respective country, agency website and information about their missions.
4. <https://www.isro.gov.in/list-of-spacecrafts>
 From this system we got the idea what type of columns will be there in our database and how the data is organised including launch vehicle, launch date and objective/application.

2.3 Real World Flow and Working

- The system is intended to give immediate information about the data asked and inputs from the user will be like start date-time, end date-time and selected attributes. The dates are required because each tuple will be associated with a time stamp; as the data will be updated live.
- The system should be able to send its data to other already existing ISRO systems for computations and calculations. Our system won't do that on its own.
- The only calculations it will do is that if a celestial object comes very close to earth then it will give an alert message.
- The issues can be data asked and haven't been updated yet from the live data stream due to server issues and it is being queried by the user. This will show an error message, data will be shown soon. Server under maintenance.
- Another issue is that users can ask for data from outside its list of privileges; then as all the people are to be respected and someone's point of view shouldn't hurt; some good appropriate output should be there. The user may be in need of that data in urgency; so there should be some mechanism to handle such requests optimally. We haven't thought this out yet.

2.4 User Classes and Characteristics

The users will be differentiated on the level of responsibility they have in the organization (ISRO).

The user classes can be:

Mission Control:

Mission Manager: This user will have access to the mission data from the particular satellite and can reposition or reconfigure satellite based on current requirements.

Mission Member: Will have access to mission data only.

Satellite Control:

Satellite Manager: All exclusive access to particular satellite data and over its control.

Satellite Member: Will have access to data from satellite for analysis.

Black holes/asteroids/Exoplanet Researcher:

Will have access to data from satellites from this particular domain only.

NEO Department:

The users (researchers, scientists, managers) will have access to the NEO database of objects which are undergoing study.

NEO Threat Assessment Department:

The users here will have privilege to *raise alerts* and ask for additional data from other satellite databases along with the access as specified above.

Satellite Repair and Troubleshooting Department:

The users here will have access to satellite health data and have privilege to raise alerts accordingly.

3. Background Readings

3.1 Description of some readings

We read everything online, from various ISRO websites which had information about

1. Missions: Like launch vehicle, launch date, orbit type, payload and application.
2. Satellites: Launch date, launch mass, launch vehicle, orbit type. Additional information like current position, health and other data not available.
3. Real time tracking available but it is how much reliable is not known. The website (mentioned below) gives NORAD ID, Latitude, Longitude, Altitude and current speed.
4. We also read about some automated networks of telescopes like HATNet which tracks extrasolar planets.
5. The Jet propulsion Laboratory of California Institute of Technology has CNEOS: Centre for Near Earth Objects Studies. It has impact risk data which includes year range, impact probability, estimated diameter of the object, estimated velocity, etc.

The links for the data read above respectively:

1. <https://www.isro.gov.in/all-missions-0>
2. <https://www.isro.gov.in/spacecraft/list-of-earth-observation-satellites>
3. [LIVE REAL TIME SATELLITE TRACKING AND PREDICTIONS: BEIDOU 3 G2](#)
4. [HATNet Project - Wikipedia](#)
5. [Sentry: Earth Impact Monitoring](#)

ISRO

Mock Interview 1

Summary

System: Space Agency Database

Project Reference: Number 10

Interviewee: Hetvi Patel(**Role Play**)

Designation: Scientist/Engineer-H (in) AstroStat Mission

Interviewer:

1) Divyesh Rohit(**Role Play**)

Designation: Business Development Executive DSQL Solutions

2) Tulsi Shah(**Role Play**)

Designation: Developer - DSQL Solutions

Date: 29/09/2020 **Time:** 14:30 Hrs

Duration: 45 minutes **Place:** Ms. Patel's Office, ISRO

Purpose of Interview:

Preliminary meeting to identify problems and requirements regarding the new system being created for ease in access to mission data to ISRO's scientists.

Observations and recommendations by Ms. Patel:

1. Only current mission details being showbiz not of much use to us. Actual details like objective completion, new findings, budget used, failures, successes, mission system health details, instrument health, satellite/payload health, etc should be added.
2. Make a separate schema for civilians to access data which has no sensitive information about ongoing missions.
3. Proper privileges should be used for the users; which will be decided by the mission leader/department head/ the secretary/ the director only.
4. Data of required space elements like satellites, rovers should be accessible by mission control without any authorization from satellite division.

Further discussions needed after some work on the system is done.

ISRO

Mock Interview 2

Summary

System: Space Agency Database

Project Reference: Number 10

Interviewee: Nazeem Siddique (**Role Play**)

Designation: Distinguished Scientist (in) EXO Division

Interviewer:

1) Skand Vala(**Role Play**)

Designation: CEO Business Development Executive DSQ Solutions

2) Divyesh Rohit(**Role Play**)

Designation: Business Development Executive DSQ Solutions

Date: 30/09/2020 **Time:** 16:00 Hrs

Duration: 60 minutes **Place:** Mr. Siddique's Office, ISRO

Purpose of Interview:

Preliminary meeting to identify problems and requirements regarding the new system being created for ease in access to EXOP Division of ISRO.

Recommendations from Mr. Siddique :

1. There should be an open source table accessible by anyone who wants to report a new planet discovery. The person has to give all the information like Name, Address, AU distance, location in the celestial coordinate system.
2. Then it should be verified by the current database of exoplanets which is separate and verified by ISRO. If the planet is not there, then if it has some special features which will be notified to the system by an EXO executive, the planet will be added to the main list.
3. (He was quite busy at the moment and asked for a follow up interview next week if he gets any new ideas)

ISRO

Mock Interview 3

Summary

System: Space Agency Database

Project Reference: Number 10

Interviewee: Chintan Senapati (**Role Play**)

Designation: Deputy Associate Administrator

Interviewer:

1) Skand Vala(Role Play**)**

Designation: CEO Business Development Executive DSQL Solutions

2) Janvi Patel(Role Play**)**

Designation: Database System Expert DSQL Solutions

Date: 01/10/2020 **Time:** 10:00 Hrs

Duration: 30 minutes **Place:** Admin Building, ISRO

Purpose of Interview:

To assess how the user classes are divided in the ISRO and how the privileges are distributed and how to share data with other agencies.

Points to be considered very carefully told by Mr. Senapati:

1. The users are divided into three groups: Admin department, directorates and members of the directorates.
2. The data from hier classes cannot be accessed by below ones, and if needed that is only if there is a separate table for such users with extra privileges that also for the mentioned time period.
3. The sharing of information among various departments has to be documented in a separate table to keep track of information being shared.

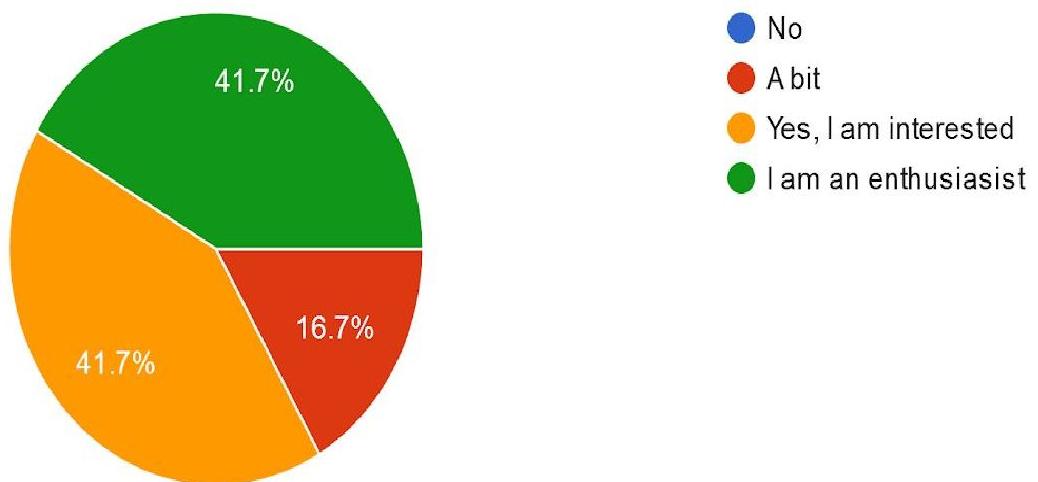
Survey Analysis

Q1) Are you interested about what's happening in outer space?

- A. No
- B. A bit
- C. Yes, I am interested
- D. I am an enthusiast
- E. Other:

Are you interested about what's happening in outer space?

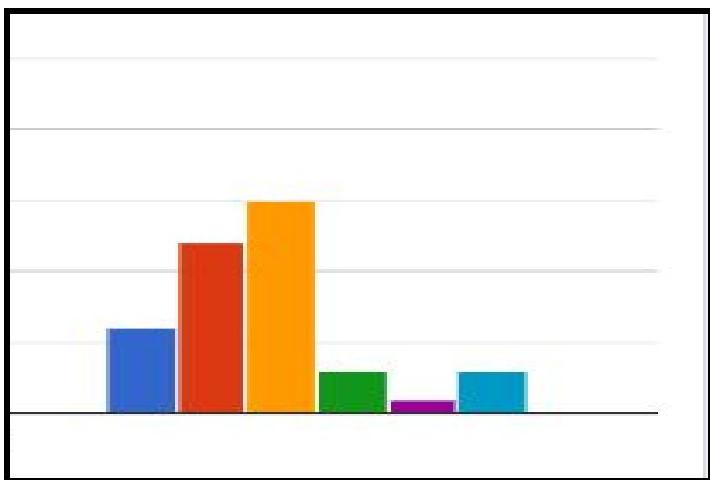
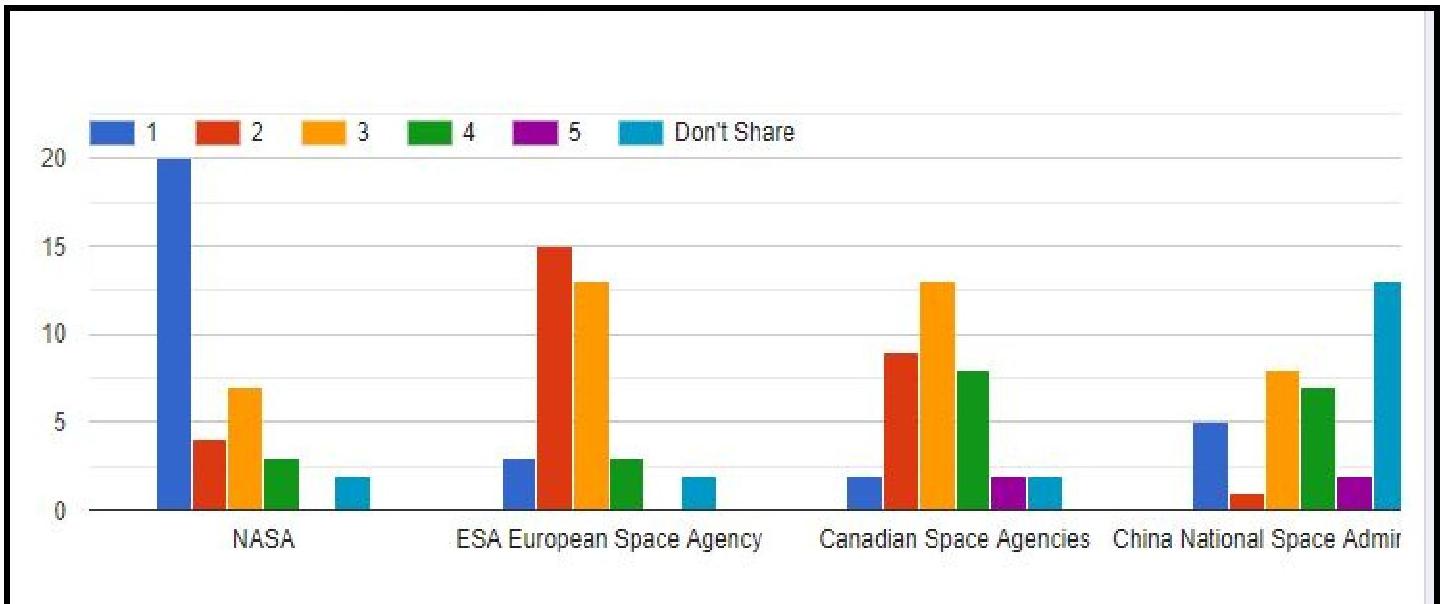
36 responses



Q_2) Should ISRO share their mission findings and experimental outcomes with what other space agencies? Choose ranking for each Organization.. Number of responses:

- NASA
- ESA European Space Agency
- Canadian Space Agencies
- China National Space Administration
- Japan Aerospace Exploration Agency

Answer in 1 to 5 and Don't Share

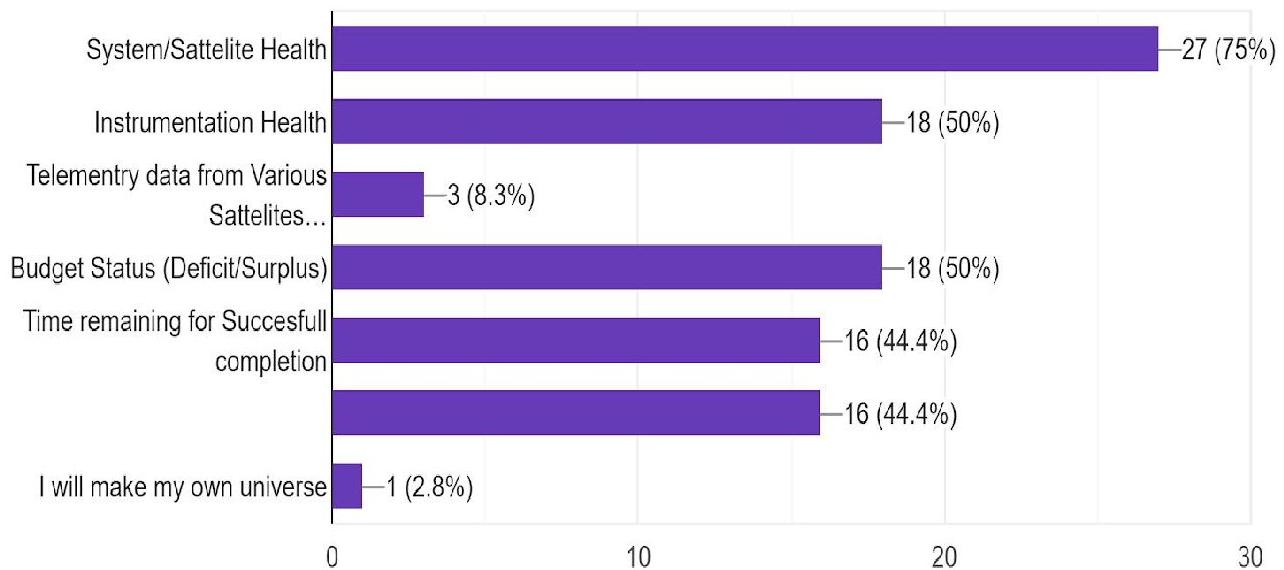


Japan Aerospace Exploration Agency

Q_3)ISRO needs to keep track of their ongoing missions with all the important data presented concurrently for proper analysis and successful completion of objectives. Which ones you think are important ? (Multiple choice)

- A. System/Satellite Health
- B. Instrumentation Health
- C. Telemetry data from Various Satellites like Velocity, disturbance, etc
- D. Budget Status (Deficit/Surplus)
- E. Time remaining for Successful completion

ISRO needs to keep track of their ongoing missions with all the important data presented concurrently for proper analysis and successful co...ink are important ? (Choose only important ones)
36 responses



For the above question if you have any other thoughts.

Some Thoughts

1. Tracking of celestial entities will be helpful for scientists.
2. We should make ties with SpaceX.

Q_4)Choose which user classes are required for the database to function properly. Choose only the important ones. (Multiple choice)

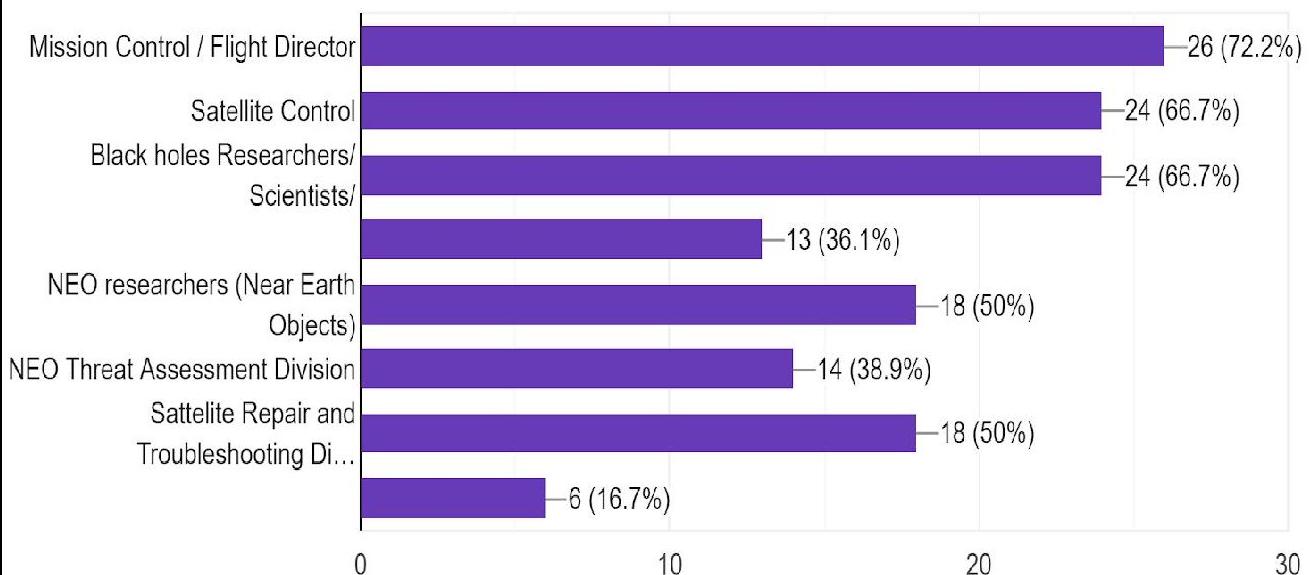
- A. Mission Control / Flight Director
- B. Satellite Control
- C. Black holes Researchers/Scientists/
- D. Exoplanets Researchers/Scientists/
- E. NEO researchers (Near Earth Objects)
- F. NEO Threat Assessment Division

G. Satellite Repair and Troubleshooting Division

H. Other:

Choose which user classes are required for the database to function properly. Choose only the important ones.

36 responses



For the above question if you have any other thoughts.

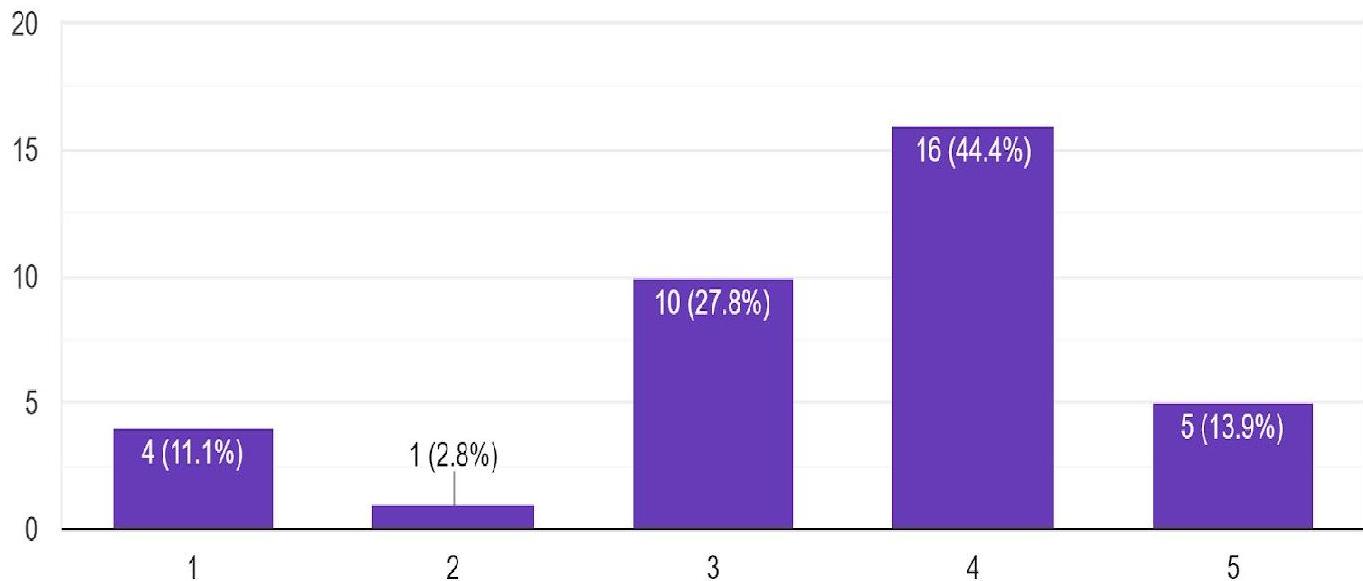
No response came here.

Q_5)Should NASA build a base on the Moon before going to Mars?
(give rating between 1 to 5)

1-strongly disagree....5-strongly agree

Should NASA build a base on the Moon before going to Mars?

36 responses



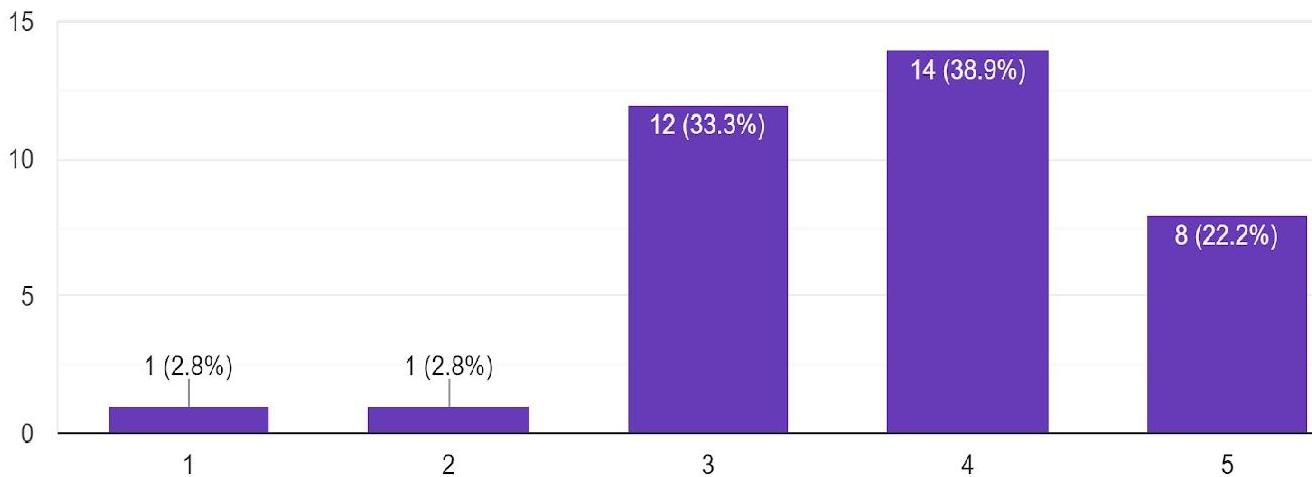
**Q_6)How much are you satisfied about the data which ISRO provides to the Indian citizens?
(Answer as a citizen)**

(give rating between 1 to 5)

1- Very Dissatisfied....5-Satisfied

How much are you satisfied about the data which ISRO provides to the Indian citizens? (Answer as a citizen)

36 responses



Q_7) If you have any thoughts about the data which should be shared by ISRO to the civilians please let us know below, as it is part of project objective.

Some thoughts came here. We have covered it into Observation.

Observation

From survey we observe below things

- Maximum people(80%) are interested in what's happening in outer space.
- Maximum number of people said that ISRO should share its data with NASA and not with any other(Canadian Space Agencies,China National Space Administration Japan Aerospace Exploration Agency) agencies.
- Maximum number(75%) of people think that ISRO needs to keep track of System/Satellite health more than any other operation.
- Above 70% people required mission control/flight control for the database to function properly and 67% people needed blackholes researchers and satellite control.
- Maximum people agree (45%) that NASA should build a base on the Moon before going to Mars.
- Maximum people(39%) are satisfied with ISRO data.some people have their own thoughts like
 - 1) Apart from missions ISRO should share information regarding satellites, space entities and future threats too.
 - 2) Data should not be shared more than which they were/are sharing for any project.
 - 3) Data sharing is good as civilians can know about progress report of ISRO and feels proud
 - 4) Isro provides information to us which we need to know and the person gets all needed information.

6. List of Requirements:

6.1 Product Requirements

A table for a list of satellites with a separate table for each satellite containing relevant information all in the schema “SAT”.

A table for current missions with each separate table having information about mission progress recorded at a regular time interval; all in schema “MISSIONS”.

Separate tables for black holes, stars, exoplanets and big asteroids with relevant information; all in schema “EXO”.

A table for keeping track of Near Earth Objects in schema “NEO”.

Sensor requirements for notify the scientists when some event occurs, Alert about the future collisions with earth or other entities.

6.2 Safety requirements:

If there's extensive damage to a large portion of the database because of catastrophic failure, like a disk crash, the recovery method restores a past copy of the database that was saved to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the saved log, up to the time of failure.

6.3 Security requirements:

Security systems need database storage similar to many other applications. However, the special requirements of the security market mean that clients must choose their database partner carefully.

Also to ease the user interaction with the database, the developers hide internal irrelevant details from users. There are three level of abstraction:

Let's say we are storing some information in a client table. At physical level these records can be described as blocks of storage in memory. These details are often hidden from the programmers.

We will make different relations for the Mission control department,satellite department, NEO department ,Black holes/asteroids/Exoplanet Researcher, NEO Threat Assessment.

Department,Satellite Repair and Troubleshooting Department so that Mission department can't see the information about any other department and same for all so privacy will maintain.

6.4 Interview Requirements

- Requirements for actual details like objective completion, new findings, budget used, failures, successes, mission system health details, instrument health, satellite/payload health.
- Separate schema for civilians to access data which has no sensitive information about ongoing missions.
- Proper privileges should be used for the users; which will be decided by the mission leader/department head/ the secretary/ the director only.
- Data of required space elements like satellites, rovers should be accessible by mission control without any authorization from satellite division.
- An open source table accessible by anyone who wants to report a new planet discovery. The person has to give all the information like Name, Address, AU distance, location in the celestial coordinate system.
- Then it should be verified by the current database of exoplanets which is separate and verified by ISRO. If the planet is not there, then if it has some

special features which will be notified to the system by an EXOP executive, the planet will be added to the main list.

6.5 Observations Requirements

- Maximum number(75%) of people think that ISRO needs to keep track of System/Satellite health more than any other operation.
- Above 70% people required mission control/flight control for the database to function properly and 67% people needed blackholes researchers and satellite control.
- Maximum people agree (45%) that there should be a base on the Moon before going to Mars.
- Maximum people(39%) are satisfied with ISRO data.some people have their own thoughts like
 - a. Data should not be shared more than which they were/are sharing for any project.
 - b. Data sharing is good as civilians can know about the progress report of ISRO and feel proud.
 - c. Isro provides information to us which we need to know and the person gets all needed information.
 - d. Apart from missions ISRO should share information regarding satellites, space entities and future threats too.

6.6 Software Quality Attributes

- **Availability:** The missions should be available on the specified date and specified time as many clients are doing research.
- **Correctness:** the clients should get the correct information of missions and its launch date & time.
- **Maintainability:** The administrators and mission in chargers should maintain correct schedules of missions.
- **Usability:** The mission schedules and related information should satisfy a maximum number of customer needs.

6.7 Removing duplicates and prioritizing according to Frequency

Note: *The requirements are listed according to the priority we thought u=is more important rather than frequency of occurrence as this system of prioritizing was not much effective in our case/*

Priority	Requirements	Frequency
1	Requirements for actual details like objective completion, new findings, budget used, failures, successes, mission system health details, instrument health, satellite/payload health.	3
2	Sensor requirements for notify the scientists when some event occurs, Alert about the future collisions with earth or other entities.	1
3	A separate table for each mission and satellite.	1
4	Separate tables for black holes, stars, exoplanets and big asteroids with relevant information; all in schema “EXO”.	2
5	Separate schema for civilians to access data which has no sensitive information about ongoing missions.	3
6	Sharing information with private companies like SpaceX may also be done. So we will add an option which enables us to share data with private entities as well.	2
7	Proper privileges should be used for the users; which will be decided by the mission leader/department head/ the secretary/ the director only.	1
8	If there's extensive damage to a large portion of the database because of catastrophic failure, like a disk crash, the recovery method restores a past copy of the database that was saved to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the saved log, up to the time of failure.	1
9	Maximum number of people responded that data shouldn't be shared with the chinese space agency. Our team agreed with the user's views. But the main decision lies with the ISRO administration and we will do as they say.	1
10	An open source table accessible by anyone who wants to report a new planet discovery. The person has to give all the	1

	information like Name, Address, AU distance, location in the celestial coordinate system.	
--	---	--

E) User Classes and Requirements:

I. Below Users will be under **DBA class** i.e, **Database Administrator**.

- A. Administrator/Director
- B. Deputy Administrator/Director
- C. Associate Administrator
- D. Deputy Associate Administrator
- E. Chief of Staff
- F. Associate Administrator for Strategy and Plans

The above Users roles will have access to all the schemas and can revoke/add privileges to all the user roles.

II. Below Users will under **RESOURCE class** i.e, They

can Define data,

Create database procedures,

Grant privileges for their database objects to other database users.

- A. Mission Directorate
- B. Mission Support Directorate
- C. Satellite Directorate
- D. EXO Division
- E. NEO Threat Assessment Division
- F. Satellite Repair and Troubleshooting Division

III. Below Users will be under **STANDARD class** i.e, they can

Access data and database procedures with necessary granted privileges

Define view tables, synonyms, and temporary tables

- A. Ground Mission Crew (Mission Directorate)
- B. Associate of Human Capital Management (MS Directorate)
- C. Strategic Infrastructure (MS Directorate)
- D. Satellite Control Crew
- E. EXO Engineers, scientists and researchers.
- F. NEO Engineers, scientists and researchers.
- G. Engineers from the Repair and troubleshooting Division.

Operating Requirements

Hardware Requirements:

- A 1 GHz processor
- 2 GB of RAM.
- 512 MB of HDD
- Intel Pentium Processor
- Latest Version of any browser.

Software Requirements:

- PGSQl

External Interface Requirements:

- MS Access
- Java IDE
- PHPMyadmin

(We will be using any of the above)

Product Functions:

Our mission will facilitate users to get information about any missions of ISRO. If a user wants any information about any mission he can get it by our product only by mission name. Our product will provide information about mission launching date, Orbit type, Application, mission was successful or not.

1. Will let users add new missions, access current live mission data, experiments and outcomes.
2. Will let users access satellite data along with its health, position and services offered.
3. Will have a dedicated login feature to access each tier of information in the database. Every extra information access will require a different username and password.
4. Will alert the users with DBA class if there is some possibility of future collisions with earth.
5. Will have a feature that lets some data to be shared with other space organizations which will be decided exclusively by Director or Deputy Director.

6. There will be a search bar for searching particular data with filters.

Privileges:

1. Live data will be accessible to only the users pertaining to a particular directorate.
2. Old data of missions, experiments and its outcomes will be accessible by anyone as and when the director wants it to go public.
3. Only the directorate head will have access to edit and add new tables.
4. The data received LIVE from sensors cannot be edited unless permission is granted from the administrator.

Assumptions:

1. We assumed that we would have separate directorate/departments for every sub-objective were given. For example, mission control directorate for mission finding and so on.
2. We did this because it will make organising data for separate sections in separate schemas very easy.
3. We also assumed that we would get live data from ISRO servers for our database. In the tables the data will be filled out automatically. For the sake of showing the working, we will input our own random data.
4. In user classes, we made up the user roles in accordance with the organizational hierarchy used in NASA. We searched the same for ISRO but didn't get a proper structure. So, we improvised a bit.
5. We based our questionnaire very specifically on areas which were crucial for the system like which features we need in this one and so on. Also in these areas we were unsure too.

Business Constraints:

1. The most important and serious one will be the dependence on ISRO servers for live data. It's uptime cost will be a bit heavy, because if implemented it will be heavily used by civilians, ISRO and other space organizations too.
2. The other one which we figured out is "Security". As a lot of sensitive information will be in it and also it will be live 24x7, it will be constantly attacked by hackers for that worthy data. So, a big costly 24x7 security team will be a must.
3. We were able to figure out only these two.

Section 2: Final Noun Analysis

Space Agency Database

The original problem description was:

Design a system that helps space agencies like ISRO, NASA to track their missions, findings, and space element movements. Keep track of satellite health, positions and services. It should also keep track of stars, planets, blackholes, moving stars, big asteroids and notify the scientists when some event occurs, or alert about the future collisions with earth or other entities. This system should be able to keep track of their experiment and outcomes. When need arises different agencies should be able to work-together to complete the mission.

Proceeding in an orderly fashion **Design a system which should have at least the below-specified information in your database, which can be in turn helpful to the space organizations like ISRO and NASA; as follows:**

1. Mission details like
 - a. Launch Vehicle
 - b. Spacecraft
 - c. Ground Segment
 - d. Mission profile
 - e. Payload details
 - f. Duration
2. For each mission, there should be
 - a. Regular updates about the orbiter/rover/satellite being used.
 - b. Update duration can be once a day.
 - c. The data can be generated on your own or can be sourced from an external system/database as per your convenience.
 - d. The updates should include update time-date, objectives completed for the day, and anything else you want to add.
 - e. Also, there should be a brief description of the space elements involved. Suppose the mission is “Mangalyaan” so the relevant space object is Mars. So, the current atmospheric conditions of Mars can be tracked side by side with the probe being sent.

3. Satellites

a. Celestial coordinates/position

- b. Satellite orbit is geostationary, low earth orbit (LEO), medium earth orbit (MEO), High earth orbit (HEO).
- c. Orbit Distance and altitude
- d. Orbit velocity
- e. Orbital Period
- f. Navigation system being used
- g. Number of malfunctioning parts (if any)
- h. Names of the malfunctioning parts
- i. Overall health percentage
- j. Services provided.

Below is some basic information about stars, planets, black holes, and Asteroids for the student's reference.

Stars

A star is an astronomical object consisting of a luminous spheroid of plasma held together by its own gravity. The nearest star to Earth is the Sun. Many other stars are visible to the naked eye from Earth during the night, appearing as a multitude of fixed luminous points in the sky due to their immense distance from Earth. The stars appear to be always in the same place relative to each other. It is as if they are fixed points in a crystal sphere that rotates around the earth. Once you have a map, you need only an ephemeris to keep track of the pesky ephemeral objects like planets and comets that move against it.

Star's life begins with the gravitational collapse of a gaseous nebula of material composed primarily of hydrogen, along with helium and trace amounts of heavier elements. When the stellar core is sufficiently dense, hydrogen becomes steadily converted into helium through nuclear fusion, releasing energy in the process

Planets

A planet is an astronomical body orbiting a star or stellar remnant that is massive enough to be rounded by its own gravity, is not massive enough to cause thermonuclear fusion, and has cleared its neighbouring regions of planetesimals.

The Milky Way is littered with a vast diversity of planets: giants that blur the line between planet and failed-star brown dwarf; tiny worlds similar in size to Earth's moon; planets that take 100,000 years to orbit their suns or whip around in hours; lava worlds; ice worlds; and planets that circle multiple suns or whirling pulsars.

Scientists find them by watching stars that wobble, change gravity, vary in color, or dip slightly in brightness.

Blackholes

Very massive stars(more than 15 to 20 times the sun's mass)

Black holes cannot suck matter into them except under certain conditions. If the sun instantly turned into a black hole, Earth and even Mercury would continue to orbit the new object and not fall in. There are two common cases in the universe in which matter can be dragged into a black hole.

Case 1: If a body orbits close to the event horizon in an elliptical orbit, it emits gravitational radiation, and its orbit will eventually decay in millions of years.

Case 2: A disk of gas can form around a black hole, and through friction, the matter will slowly slide into the black hole over time.

Asteroids

Asteroids are rocky worlds revolving around the sun that are too small to be called planets. They are also known as planetoids or minor planets. There are millions of asteroids, ranging in size from hundreds of miles to several feet across. In total, the mass of all the asteroids is less than that of Earth's moon.

Asteroids are left over from the formation of our solar system. Our solar system began about 4.6 billion years ago when a big cloud of gas and dust collapsed. When this happened, most of the material fell to the center of the cloud and formed the sun.

Some of the condensing dust in the cloud became planets. The objects in the asteroid belt never had the chance to be incorporated into planets. They are leftovers from that time long ago when planets formed.

4. The following data about the stars is required.
 - a. Tracking ID/Name
 - b. Classification i.e, yellow/orange/red dwarf, supergiant, neutron star, failed star, etc.
 - c. Its size
 - d. It's shape
 - e. Its angular distance from say, earth
 - f. Its Declination
 - g. Its composition. Like how much percentage nitrogen, argon, oxygen, etc the star is made up of.
5. About the Planets, the following information should be there in the database:
 - a. Its mass, diameter, density
 - b. Its gravity
 - c. Its length of the day
 - d. Its distance from the sun
 - e. Its mean temperature
 - f. Its number of moons
 - g. Is it a ring system or not?
6. The database should contain this information about BlackHoles:
 - a. Black-Hole name
 - b. Mass
 - c. Distance Mass (Light years)
 - d. Threat Level to Us
7. Database should contains below information about Asteroids
 - a. Asteroid Number and Name
 - b. Diameter
 - c. Mass
 - d. Rotation Period
 - e. Orbital Period
 - f. Spectral class
 - g. Semi Major Axis

- h. Orbital Eccentricity
 - i. Orbital Inclination
 - j. Current, past and future missions about Asteroids
8. There should be a mechanism to detect collisions and alert the scientists if there is a possibility between some important (for simplicity) space elements.
- a. Suppose a user enters a new asteroid in the database and if it is on the way to collide with earth, or flow past earth at a dangerous distance which will be calculated according to some data, which is left to the student's imagination; then a warning message should be displayed.
 - b. There should be alert/notification if there is a supernova explosion, gamma ray burst, etc.
9. If need arises then the agency currently using the database should be able to share the data with other space agencies.
- a. Mention how the data will be shared.
 - b. Mention how the authorization will work to share the data.
10. There should be a list with scientists/researchers accessing the database with their clearance level and their department/directorate.

Table.1.
All Extracted Nouns & Verbs from Problem Description

Nouns	Verbs
Ongoing Missions	<i>Mission Updates</i> Expedition <i>Ongoing Missions</i>
Past Missions	<i>Scientists</i> Assigned <i>Ongoing Missions</i>
Future Missions	<i>Satellites</i> Satellites Allocated <i>Ongoing Missions</i>
Satellites	<i>Past Missions</i> Mission Archives <i>Scientists</i>
Stars	<i>Stars, Black Holes, Planets</i> Being Studied <i>Researchers</i>
Black holes	<i>Satellites</i> Orbiting <i>Planets</i>
Asteroids	<i>Asteroids</i> NEO <i>Collision Threat</i>
Collision Threat	<i>Asteroids</i>

Planets	<i>Satellites</i> Sat Update <i>Satellites Updates</i>
Scientists	<i>Missions Updates Shared</i> Missions Exchanged <i>Space Agencies</i>
Mission Updates (for each mission)	<i>Satellite Updates Shared</i> Satellites Exchanged <i>Space Agencies</i>
Satellite Updates (for each mission)	
Researchers	
Space Agencies	
Missions Updates Shared	
Satellites Updates Shared	

Table.2. Accepted Noun & Verbs list

Candidate entity set	Candidate attribute set	Candidate relationship set
Missions	1. Mission Name 2. Mission ID 3. Launch Vehicle 4. Ground Segment 5. Space Segment 6. Mission profile 7. Payload details 8. Start Date 9. End Date 10. Status	Expedition Assigned
Satellites	1. Satellite Name 2. Satellite ID 3. Orbit Details 4. Orbit Distance and altitude 5. Orbit velocity 6. Orbital Period 7. Navigation system 8. Number of malfunctioning parts 9. Malfunctioning parts name 10. Health percentage 11. Services Provided	Orbiting Satellite Allocated Sat_Update Sharing_Staelite_Info
Stars	1. Star ID 2. Classification 3. Size 4. Shape 5. AU Distance	Changing_Stars Star_Studied

Blackholes	<ol style="list-style-type: none"> 1. Blackhole ID 2. Name 3. Mass 4. AU Distance 	Blackholes_Studied Changing_Blachole
Asteroids	<ol style="list-style-type: none"> 1. Asteroid Name 2. Asteroid Id 3. Diameter 4. Mass 5. Spectral class 	Asteroid_Studied NEO
Planets	<ol style="list-style-type: none"> 1. Planet ID 2. Name 3. Mass 4. Diameter 5. Density 6. Length of the day 7. Number of moons 8. AU Distance 9. Mean temperature 	Orbiting Planet_Studied
Collision Threat	<ol style="list-style-type: none"> 1. Tracking ID 2. Threat Status 3. Threat Details 4. Date-Time 	NEO
Mission Updates (for each mission)	<ol style="list-style-type: none"> 1.Date-Time 2.Experiments 3.Outcomes 	Expedition

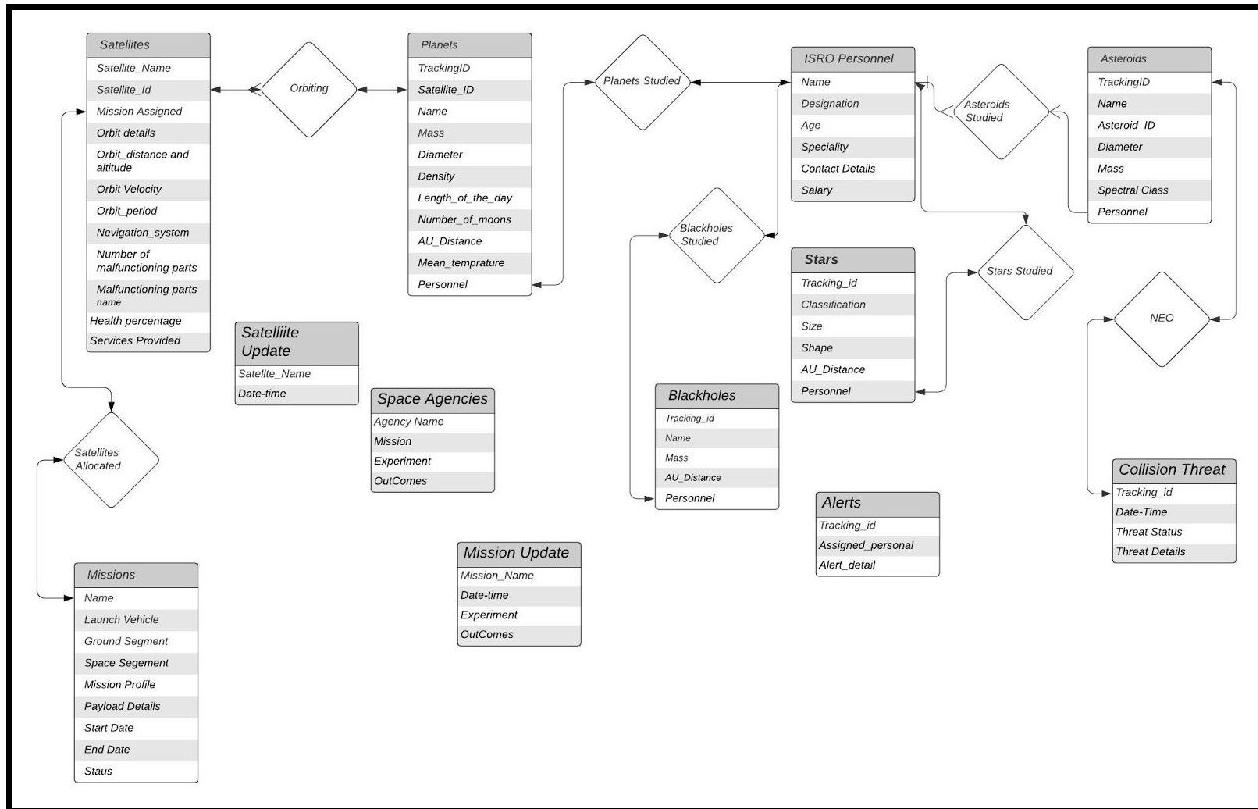
Satellite Updates (for each mission)	1.Update 2.Date-Time	Sat Update
ISRO_Personnel	1. Name 2. ID 3. Designation 4. Age 5. Speciality 6. Salary 7. Start-Date 8. {Contact Details}	Planets Studied Blackholes Studied Asteroids Studied Stars Studied Assigned
Space Agencies	1. Agencies_Name 2.Tracking_ID	Misson_Exchanged Sharing_satelite_info

Table.3. Rejected Noun & Verbs list

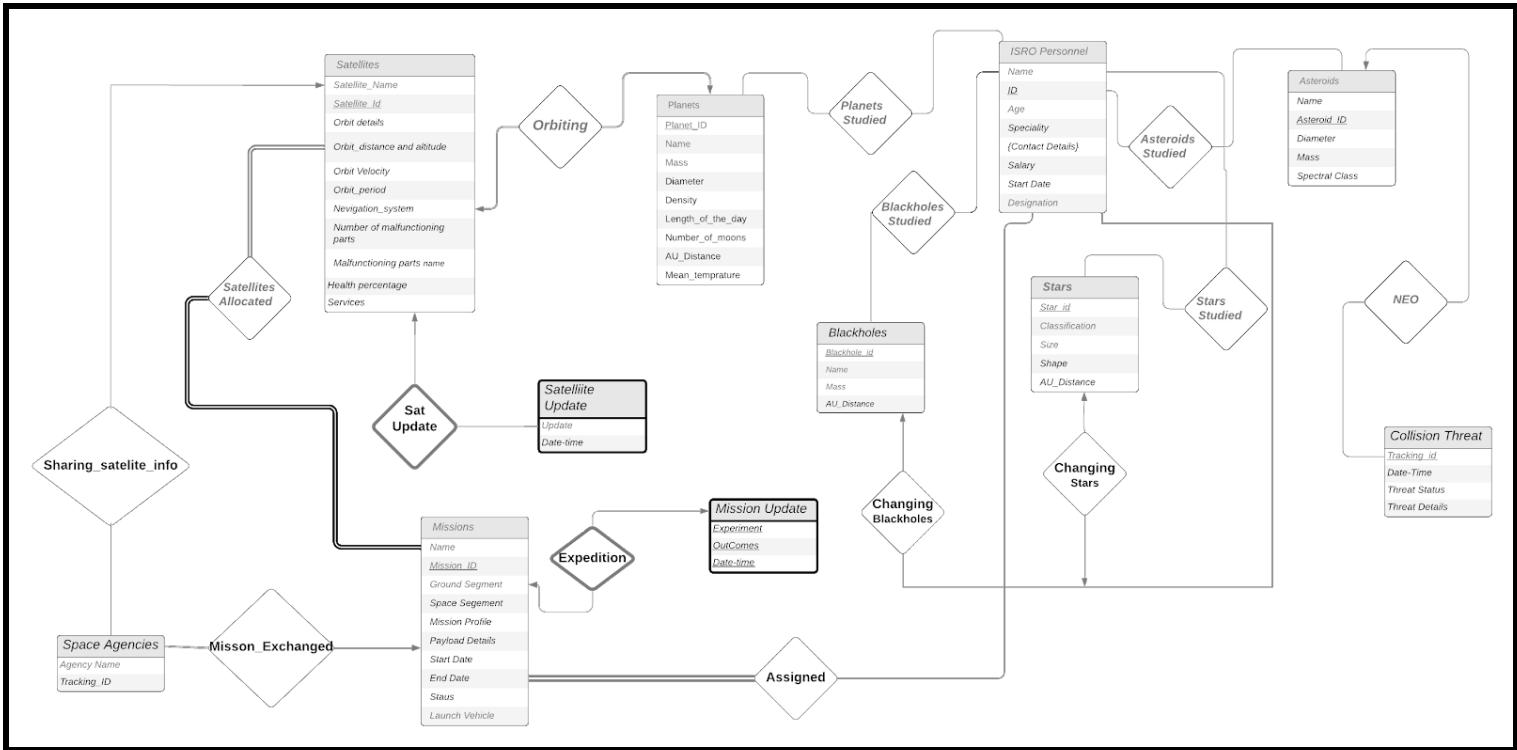
Nouns	Verbs
Past Missions	General
Future Missions	General
Researcher	General

Section 3: Final ER Diagrams of all versions

ER Diagram Version 1



ER Diagram Final Version



Section4: Conversion of Final ER-Diagram to Relational Model

Mapping E-R Model to Relational Model

Satellites(Satellite_ID,Satellite_Name,Orbit Details,Orbit_Distance and Altitude,Orbit Velocity,Orbit_period,Navigation System,Number of malfunctioning parts, malfunctioning parts name,Health Percentage,Services,Planet_ID)

ISRO Personnel(ID,Name,Designation,Age,Speciality,Salary,Start-Date)

Planet(Planet_id,name,mass,Diameter,Density,length_of_the_day,number_of_moons,AU-distance,Mean_temprature)

Missions(Mission ID,Launch Vehicle,Ground Segment, Mission profile,Payload details,Start Date,End Date,Status)

Stars(Star_ID,Classification,Size,Shape,AU_Distance)

Blackholes(Blackhole ID,Planets_ID,Name,Mass,AU_Distance)

Asteroids(Asteroid Id , Asteroid Name,Diameter,Mass,Spectral class)

Collision Threat(Tracking ID,Threat Status,Threat Details,Date-Time,Asteroid_id)

Space Agencies(Tracking_ID ,Agencies_Name,Satellite_id,Mission_id)

Satellite_allocated(Satllite_id,Mission_id)
Planet_studied(Planet_id,Personnel_id)
Blackhole_studied(Blackhole_id,Personnel_id)
Asteroids_Studied(Asteroid_ID,Personnel_ID)
Star_studied(Star_id,Personnel_id)
Assigned(Mission_id,Personnel_id)
Personnel_Details(Personnel_Id,Phone_Number)
Satellite_update(satellite_id,update,date_time)
Mission_update(Mission_id,Experiment,outcomes,date_time)

Section5: Normalization and Schema Refinement

PK dependency

1. Satellite_Id - Satellites
2. Personnel_ID - ISRO Personnel
3. Planet_id - Planet
4. Mission ID-Missions
5. star_id-Stars
6. Blackhole ID - Blackholes
7. Asteroid Id-Asteroids
8. Personnel_Id,Phone_Number -Personnel_Details
9. Tracking ID-Collision Threat
10. (Alert_id,Personnel_id) - Alert_assigned
11. Tracking_ID-Space Agencies
12. (Mission_id,Date_Time,Experiments,Outcomes)-Mission Updates
13. (Satellite_id,Update,Date-Time)-Satellite Updates
14. (Blackhole_id,Personnel_id)-Blackhole_studied
15. (Asteroid_ID,Personnel_ID)-Asteroids_Studied
16. (Star_id,Personnel_id)-Star_studied
17. (Mission_id,Personnel_id)-Assigned
18. (Planet_id,Personnel_id)-Planet_studied
19. (Satllite_id,Mission_id)-Satellite_allocated

FK dependencies:

1. **Planet_ID** references to **Satellite, Planet_Studied**
2. **Satellite_ID** references to **Satellite_Updates, Satellite_Allocate, Space_Agencies**
3. **Mission_ID** references to **Satellite_Allocate, Space_Agencies, Mission_updates, Assigned**
4. **Personnel_ID** references to **Planet_Studied, Blackhole_Studied, Asteroids_Studied, Star_Studied, Assigned, Personnel_Studied**
5. **Blackhole_ID** references to **Blackhole_Studied**
6. **Star_ID** references to **Star_Studied**
7. **Asteroid_ID** references to **Asteroids_Studied, Collision_Threat**

Functional dependency

1. Designation -> Salary
2. mass,Diameter->Density
3. Asteroid Name->Diameter
4. Asteroid Name->Mass
5. Asteroid Name->Spectral class

Anomalies

Insert,delete,Update Anomalies

1. **Planet_ID**
2. **Satellite_ID**
3. **Mission_ID**
4. **Personnel_ID**
5. **Blackhole_ID**

6. Star_ID
7. Asteroid_ID

1 NF

No composite and multivalued attributes in our database.

2 NF

We don't have Composite primary keys that are partial. So our schema is in 2NF already.

3NF

Our database is only in 3 NF.

SECTION 6

1. Final DDL scripts:

We've created 5 Schemas as mentioned below:

1. Agency and personnel
2. Threats
3. PABS
4. Satellites
5. Missions

1. Isro_Personnel

```
create table "Agency and personnel".ISRO_Personnel
(
    ID int,
    Name char(50),
    Designation char(50),
    Age int,
    Salary Bigint,
    Start_Date date,
    primary key(id)
);
```

2. Personnel_Details

```
create table "Agency and personnel".Personnel_Details
(
    Personnel_Id int,
    Phone_Number char(10) check (length(Phone_Number) = 10),
    Primary key(personnel_id,Phone_Number),
    Foreign key (Personnel_id) REFERENCES "Agency and
personnel".Isro_personnel(ID) on delete cascade
);
```

3.Stars

```
create table "PABS".Stars
(
    Star_ID int,
    Classification char(50),
    Size Bigint,
    Shape varchar,
    AU_Distance int,
    primary key(Star_id)
);
```

4.blackhole

```
create table "PABS".blackholes
(
    Blackhole_ID int,
    Name char(50),
    Mass bigint,
    AU_Distance int,
    primary key(blackhole_id)
);
```

5.Star Studied

```
create table "PABS".Star_studied
(
    Star_id int,
    Personnel_id int,
    primary key(star_id,Personnel_id),
    FOREIGN KEY (Personnel_id) REFERENCES "Agency and
personnel".isro_personnel(id) ON
DELETE cascade,
    FOREIGN KEY (star_id) REFERENCES "PABS".stars(star_id) ON
DELETE cascade
);
```

6.blackhole_studied

```
create table "PABS".blackhole_studied
(
    Blackhole_id int,
    Personnel_id int,
    primary key(Blackhole_id,Personnel_id),
    FOREIGN KEY (Personnel_id) REFERENCES "Agency and
personnel".isro_personnel(id) ON
DELETE cascade,
    FOREIGN KEY (blackhole_id) REFERENCES
"PABS".blackholes(blackhole_id) ON
DELETE cascade
);
```

7.Asteriods

```
create table "PABS".Asteroids
(
    Asteroid_Id int,
    Asteroid_Name varchar,
    Diameter bigint,
    Mass bigint,
    Spectral_class char(50),
    primary key(Asteroid_Id)
);
```

8.Asteriod_studied

```
create table "PABS".asteroid_studied
(
    asteroid_id int,
    Personnel_id int,
    primary key(asteroid_id,Personnel_id),
    FOREIGN KEY (Personnel_id) REFERENCES "Agency and
personnel".isro_personnel(id) ON
DELETE cascade,
    FOREIGN KEY (asteroid_id) REFERENCES
"PABS".asteroids(asteroid_id) ON
```

```
DELETE cascade
```

```
);
```

9. Planets

```
create table "PABS".Planets  
(
```

```
    Planet_id int,  
    name varchar,  
    mass real,  
    Diameter bigint,  
    Density bigint,  
    length_of_the_day bigint,  
    number_of_moons bigint,  
    AU_distance real,  
    Mean_temprature int,  
    primary key(planet_id)
```

```
);
```

10.Missions

```
create table "Missions".Missions
```

```
(
```

```
    Mission_ID int,  
    Launch_Vehicle varchar,  
    Ground_Segment varchar,  
    Mission_profile varchar,  
    Payload_details varchar,  
    Start_Date date,  
    End_Date date,  
    Status varchar,  
    primary key(Mission_ID)
```

```
);
```

11.Mission_Updates

```
create table "Missions".Mission_Updates
```

```
(
```

```
    Mission_id Int,
```

```
Date_Time date,  
Experiments Varchar,  
Outcomes varchar,  
Primary key(Mission_id,Date_Time,Experiments,Outcomes),
```

```
FOREIGN KEY (Mission_id) REFERENCES  
"Missions".missions(Mission_ID) ON  
DELETE cascade  
);
```

12. Satellites

```
CREATE TABLE "Satellites".Satellites  
(  
    Satelite_ID INTEGER,  
    Satellite_Name char varying ,  
    Orbit_Details char varying ,  
    Orbit_Distance Bigint,  
    Orbit_Velocity bigint,  
    Orbit_period real,  
    Navigation_System char varying,  
    Number_of_malfunctioning_parts bigint,  
    malfunctioning_parts_name char varying,  
    Health_Percentage int,  
    Services char varying,  
    Planet_ID int,  
    primary key (Satelite_ID),  
    FOREIGN KEY (Planet_ID) REFERENCES "PABS".Planets(Planet_ID)  
ON  
DELETE cascade  
);
```

13.Space_Agencies

```
create table "Agency and personnel".Space_Agencies  
(  
    Tracking_ID int,  
    Agencies_Name varchar,
```

```

Satellite_id int,
Mission_id int,
primary key(Tracking_ID),
FOREIGN KEY (Satellite_id) REFERENCES
"Satellites".satellites(Satelite_id) ON
DELETE cascade,
FOREIGN KEY (Mission_Id) REFERENCES
"Missions".Missions(Mission_id) ON
DELETE cascade
);

```

14. Mission Assigned

```

create table "Missions".assigned
(
mission_id int,
Personnel_id int,
primary key(mission_id,Personnel_id),
FOREIGN KEY (Personnel_id) REFERENCES "Agency and
personnel".isro_personnel(id) ON
DELETE cascade,
FOREIGN KEY (mission_id) REFERENCES
"Missions".missions(mission_id) ON
DELETE cascade
);

```

15.planet_studied

```

create table "PABS".planet_studied
(
planet_id int,
Personnel_id int,
primary key(planet_id,Personnel_id),
FOREIGN KEY (Personnel_id) REFERENCES "Agency and
personnel".isro_personnel(id) ON
DELETE cascade,

```

```
    FOREIGN KEY (planet_id) REFERENCES "PABS".planets(planet_id)
ON
DELETE cascade
);
```

16.Collision_Threat

```
create table "Threats".Collision_Threat
(
    Tracking_ID Int,
    Threat_Status char(50),
    Threat_Details char(50),
    Date_Time date,
    Asteroid_id int,
    primary key(Tracking_id),
    FOREIGN KEY (Asteroid_id) REFERENCES
    "PABS".Asteroids(Asteroid_id) ON
    DELETE cascade
);
```

17.Satellite_allocated

```
create table "Satellites".Satellite_allocated
(
    Satellite_id int,
    Mission_id int,
    primary key(Satellite_id,Mission_id),
    FOREIGN KEY (Satellite_id) REFERENCES
    "Satellites".satellites(satelite_id) ON
    DELETE cascade,
    FOREIGN KEY (Mission_id) REFERENCES
    "Missions".missions(mission_id) ON
    DELETE cascade
);
```

18. Satellite_Updates

```
create table "Satellites".Satellite_Updates
(
    Satelite_id int,
    Update varchar,
    Date_Time date,
    Primary key(Satelite_id,Update,Date_Time),
    Foreign key (Satelite_id) REFERENCES "Satellites".Satellites(Satelite_id)
on delete cascade
);
```

LAB_9 -- Snapshots of select * queries of all the tables (Data snapshots)

1. Isro_personnel:

The screenshot shows the pgAdmin 4 interface with the following details:

- Browser Tree:** Shows the database structure under "Agency and personnel".
- Query Editor:** Contains the SQL query: `select * from "Agency and personnel".isro_personnel`.
- Data Output:** Displays the results of the query, showing 10 rows of personnel data.

	id	name	designation	age	salary	start_date
95	195	B Jayakumar	Technical Officer-E...	46	50000	2012-07-05
96	196	MS Pannirselvam	Library Officer-E ...	34	250000	2012-07-06
97	197	Nandini Harinath	Technical Officer-E...	56	200000	2012-07-07
98	198	MYS Prasad	Controller	31	300000	2012-07-08
99	199	SK Shivakumar	Library Officer-E ...	45	400000	2012-07-09
100	200	Mylswamy Annadu	Library Officer-E ...	42	200000	2012-07-10

2. Personnel_details:

The screenshot shows the pgAdmin 4 interface with the following details:

- Browser Tree:** Shows the database structure under "Agency and personnel".
- Query Editor:** Contains the SQL query: `select * from "Agency and personnel".personnel_details`.
- Data Output:** Displays the results of the query, showing 6 rows of personnel details data.

	personnel_id	phone_number
95	195	9135052006
96	196	9995692545
97	197	9765251026
98	198	9470002467
99	199	9356326660
100	200	9319823293

3. Space_agencies

pgAdmin File Object Tools Help

Browser Catalogs Event Triggers Extensions Foreign Data Wrappers Languages Schemas (6)

Agency and personnel Collations Domains FTS Configurations FTS Dictionaries Aa FTS Parsers FTS Templates Foreign Tables Functions Materialized Views 1.. Sequences Tables (3) Iso_personnel personnel_details space_agencies Trigger Functions Types Views Missions PABS

Query Editor Query History

```
1 select * from "Agency and personnel".space_agencies
```

Notifications

Recorded time	Event	Process ID	Payload
No data found			

Data Output Explain Messages

tracking_id [PK] integer	agencies_name character varying	satellite_id integer	mission_id integer
70	1069 South American Space...	569	10093
71	1070 Sri Lanka Aeronautics ...	514	10002
72	1071 Armenian Space Agency	503	10096
73	1072 Russian Federal Space ...	559	10021
74	1073 Turkish Space Agency	595	10048
75	1074 Philippine Space Agency	581	10049

4. Assigned

pgAdmin File Object Tools Help

Browser Trigger Functions Types Views Missions Collations Domains FTS Configurations FTS Dictionaries Aa FTS Parsers FTS Templates Foreign Tables Functions Materialized Views 1.. Sequences Tables (3) assigned mission_updates missions Trigger Functions Types Views PABS Satellites Threats public

Query Editor Query History

```
1 select * from "Missions".assigned
```

Notifications

Recorded time	Event	Process ID	Payload
No data found			

Data Output Explain Messages

mission_id [PK] integer	personnel_id [PK] integer
95	10000
96	10097
97	10095
98	10055
99	10087
100	10035

5. Mission_updates

Screenshot of pgAdmin 4 showing the "Missions" table in the "mission_updates" schema.

Query Editor:

```
1 select * from "Missions".mission_updates
```

Notifications:

Recorded time	Event	Process ID	Payload
No data found			

Data Output:

mission_id	date_time	experiments	outcomes
95	10006	Sols 2945-2946: Should We...	Mission Pending
96	10097	Sols 2938-2939: Eyes on the...	Mission Failed due to techni...
97	10095	Sols 2926-2927: Honing in o...	Mission Cancelled by agency
98	10055	Sols 2926-2927: Honing in o...	Mission Pending
99	10087	Sols 2914-2916: Curiosity Br...	Mission Cancelled by agency
100	10035	Sols 2940-2941: Curiosity Ey...	Mission Pending

6. Missions

Screenshot of pgAdmin 4 showing the "missions" table in the "Missions" schema.

Query Editor:

```
1 select * from "Missions".missions
```

Notifications:

Recorded time	Event	Process ID	Payload
No data found			

Data Output:

mission_id	launch_vehicle	ground_segment	mission_profile	payload_details	start_date	end_date	status	
96	10097	Atlas LV-3B	Surrey Ground segment	Study working of Laser...	Secondary Ion Mass Spec...	2009-09-12	2027-04-10	ACTIV
97	10095	GSLV Mk III	ATLAS Global Network	Study atmosphere of J...	Cross-track Infrared S...	2009-09-13	2027-04-11	ACTIV
98	10055	Ares IV	Surrey Ground segment	ISS assembly flight UL...	Neuron Spectrometer, ...	2009-09-14	2027-04-12	COMF
99	10087	Ares I	ATLAS Global Network	Study planet formatio...	Cross-track Infrared S...	2009-09-15	2027-04-13	SCHEI
100	10035	PSLV-3S	KSAT Lite	Dragonfly mission to st...	Dual Frequency Synth...	2009-09-16	2027-04-14	INACT

7. Asteroid_studied

The screenshot shows the pgAdmin 4 interface with the following details:

- Browser:** Shows the database structure under the schema "PABS". The "Tables (8)" section is expanded, showing tables like asteroid_studied, asteroids, blackhole_studied, blackholes, planet_studied, planets, star_studied, and stars.
- Query Editor:** Displays the SQL query: `select * from "PABS".asteroid_studied`.
- Data Output:** Shows the results of the query as a table:

	asteroid_id [PK] integer	personnel_id [PK] integer
96	1095	196
97	1096	197
98	1097	198
99	1098	199
100	1099	200
101	1100	101

8. Asteroids

The screenshot shows the pgAdmin 4 interface with the following details:

- Browser:** Shows the database structure under the schema "PABS". The "Tables (8)" section is expanded, showing tables like asteroid_studied, asteroids, blackhole_studied, blackholes, planet_studied, planets, star_studied, and stars.
- Query Editor:** Displays the SQL query: `select * from "PABS".asteroids`.
- Data Output:** Shows the results of the query as a table:

	asteroid_id [PK] integer	asteroid_name character varying	diameter bigint	mass bigint	spectral_class character (50)
96	1095	Minamino	437	27938	X
97	1096	Dar al Gani 593	133	24757	C
98	1097	Dhofar 1226	906	17785	C
99	1098	Jiddat al Harasis 730	967	23207	P
100	1099	Kharga	290	34112	C
101	1100	Tarfa	689	32411	S

9. Blackhole_studied

The screenshot shows the pgAdmin 4 interface with the following details:

- Browser:** Shows the database structure under the schema "PABS". The "Tables (8)" section is expanded, listing: asteroid_studied, asteroids, blackhole_studied, blackholes, planet_studied, planets, star_studied, and stars.
- Query Editor:** Contains the SQL query: `select * from "PABS".blackhole_studied`.
- Data Output:** Displays the results of the query as a table. The columns are blackhole_id [PK] integer, personnel_id [PK] integer, and payload. The data is as follows:

blackhole_id	personnel_id	payload
95	354	122
96	321	122
97	325	119
98	344	130
99	392	112
100	382	153

10. Blackholes

The screenshot shows the pgAdmin 4 interface with the following details:

- Browser:** Shows the database structure under the schema "PABS". The "Tables (8)" section is expanded, listing: asteroid_studied, asteroids, blackhole_studied, blackholes, planet_studied, planets, star_studied, and stars.
- Query Editor:** Contains the SQL query: `select * from "PABS".blackholes`.
- Data Output:** Displays the results of the query as a table. The columns are blackhole_id [PK] integer, name character (50), mass bigint, and au_distance integer. The data is as follows:

blackhole_id	name	mass	au_distance
95	NGC 4791	4866886	1319986
96	NGC 4564	2866984	3919055
97	NGC 4486B	3986155	4735685
98	NGC 4473	1556186	2408578
99	NGC 4459	1891194	3808694
100	NGC 4438	1613941	4177450

11. Planet_studied

Screenshot of pgAdmin 4 showing the "PABS" schema in the Browser pane. The "Tables" section is selected, displaying eight tables: asteroid_studied, asteroids, blackhole_studied, blackholes, planet_studied, planets, star_studied, and stars. The "Query Editor" pane contains the SQL query: `select * from "PABS".planet_studied`. The results show a list of planet IDs and personnel IDs.

planet_id	personnel_id
95	405
96	406
97	407
98	408
99	409
100	401

12. Planets

Screenshot of pgAdmin 4 showing the "PABS" schema in the Browser pane. The "Tables" section is selected, displaying eight tables: asteroid_studied, asteroids, blackhole_studied, blackholes, planet_studied, planets, star_studied, and stars. The "Query Editor" pane contains the SQL query: `select * from "PABS".planets`. The results show detailed information about the planets.

planet_id	name	mass	diameter	density	length_of_the_day	number_of_moons	au_distance	mean_temperature
3	Moon	0.0073	3475	3340	708	0	0.0025	
4	Mars	0.642	6792	3933	25	2	1.524	
5	Jupiter	1898	142984	1326	10	79	5.203	
6	Saturn	568	120536	687	11	82	9.539	
7	Uranus	86.8	51118	1271	17	27	19.18	
8	Neptune	102	49528	1638	16	14	30.06	
9	Pluto	0.0146	2370	2095	153	5	247.7	

13. Stars

Screenshot of pgAdmin 4 showing the "PABS" schema in the Browser pane. The "stars" table is selected in the Query Editor pane, displaying the following SQL query:

```
1 select * from "PABS".stars
```

The Data Output pane shows the results of the query:

star_id	classification	size	shape	au_distance	
93	O	...	3397542	Heptagram	1504000
94	M	...	2901346	Handecagram	3566250
95	M	...	4242176	Pentagram	2910656
96	O	...	2096013	Heptagram	3776780
97	F	...	2780739	Polygon	1566018
98	A	...	4022630	Pentagram	1955153
99	G	...	3390140	Pentagram	1899181
100	K	...	1378946	Enneagram	3305089

14. Star_studied

Screenshot of pgAdmin 4 showing the "PABS" schema in the Browser pane. The "star_studied" table is selected in the Query Editor pane, displaying the following SQL query:

```
1 select * from "PABS".star_studied
```

The Data Output pane shows the results of the query:

star_id	personnel_id
93	37
94	18
95	49
96	41
97	47
98	43
99	90
100	35

15. Satellites

pgAdmin File Object Tools Help

Browser

- Languages
- Schemas (6)
 - Agency and personnel
 - Missions
 - PABS
 - Satellites
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Sequences
 - Tables (3)
 - satellite_allocated
 - satellite_updates
 - satellites
 - Trigger Functions
 - Types
 - Views
 - Threats
 - public
- postres

Query Editor Query History

```
1 select * from "Satellites".satellites
```

Notifications

Recorded time	Event	Process ID	Payload
No data found			

Data Output Explain Messages

satellite_id	[PK] integer	satellite_name	character varying	orbit_details	character varying	orbit_distance	bigrnt	orbit_velocity	bigrnt	orbit_period	real	navigation_system	character varying
93	534	Dove 2k-41 (0 Flock 2K...		Non-Polar Inclined		872		6374		99.87		Galileo	
94	590	Globalstar M074 (Glob...		Polar		1022		6320		1435.92		Galileo	
95	585	Zhuhai-1 (OHS 5)		Sun-Synchronous		1308		7364		95.1		NAVIC	
96	552	Sentinel 3A		Polar		1274		6899		99.6		GPS	
97	546	SPIRALE-A (Système P...		Polar		1456		7938		1461.81		NAVIC	
98	561	HEAD-5		Non-Polar Inclined		508		7030		1436.06		GPS	
99	549	Starlink-1081		Sun-Synchronous		825		6742		95.4		BeiDou	

16. Satellite_updates

pgAdmin File Object Tools Help

Browser

- Languages
- Schemas (6)
 - Agency and personnel
 - Missions
 - PABS
 - Satellites
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Sequences
 - Tables (3)
 - satellite_allocated
 - satellite_updates
 - satellites
 - Trigger Functions
 - Types
 - Views
 - Threats
 - public
- postres

Query Editor Query History

```
1 select * from "Satellites".satellite_updates
```

Notifications

Recorded time	Event	Process ID	Payload
No data found			

Data Output Explain Messages

satellite_id	[PK] integer	update	[PK] character varying	date_time	[PK] date
92	531	Satellite 6.1 Beta Release		2015-06-19	
93	534	Satellite 6.0.8 GA Update		2015-02-12	
94	590	Satellite 6.0.7 GA Update		2015-01-02	
95	585	Satellite 6.0.6 GA Update		2014-11-08	
96	552	Satellite 6.0.5 GA Update		2014-09-09	
97	546	Satellite 6.0.4 GA Release		2014-09-24	
98	561	Satellite 5.8 GA Beta Release		2017-06-30	
99	549	Satellite 5.7 GA Beta Release		2015-01-18	

17. Satellite_allocated

Query Editor:

```
1 select * from "Satellites".satellite_allocated
```

Data Output:

satellite_id	mission_id
93	534
94	590
95	585
96	552
97	546
98	561
99	549
	10039
	10073
	10006
	10097
	10095
	10055
	10087

18. Collision_threat

Query Editor:

```
1 select * from "Threats".collision_threat
```

Data Output:

tracking_id	threat_status	threat_details	date_time	asteroid_id
93	Tackled/Over	High Neutron Bursts	2019-01-04	1085
94	Low Tier	Debris Alert for Sat...	2019-01-05	1001
95	Medium Tier	bad upper layer sp...	2019-01-06	1076
96	High Tier	Low Neutrino Burst...	2019-01-07	1039
97	Imminent	Space GPS jamming...	2019-01-08	1002
98	High Tier	Asteroid Collision ...	2019-01-09	1062
99	Tackled/Over	bad upper layer sp...	2019-01-10	1083
100	Imminent	Supernova	2019-01-11	1097

LAB_10 -- SQL Queries

1. Show all the data from Mission table
2. Show mission ground segments whose names start from O
3. Show all the data from Satellites where orbit velocity is greater

Than 7000.

4. Show satellite details where planet id is 405.
5. Show all Satellite names where orbit distance greater than average of orbit distance
6. Show name and designation from isro_personnel where start_date is less than the first january 2010.
7. List personnel ID and phone numbers who have salary greater than 50000 and are currently studying blackholes using joins OR by using a function. This can be a requirement when high level personnel are required during an emergency.
8. List count of tuples from collision_threats where status is Tracked/over.
9. Show average of all designations from isro personnel.
10. Show top 10 Asteroids names which have the highest mass.
11. Create a function to show all the imminent threats to earth in the next 6 months caused by asteroids and show the personnel assigned to those extraterrestrial entities if assigned.
12. Select star details which have the highest AU distance.
13. List all the satellite orbiting planet mars and have manufacturing parts more than 20
14. What is the satellite name for tracking ID 1020.
15. List out personnel name,mission profile,launch vehicle details Where mission status is active.
16. Name of mission profile and payload details for given agency name “Australian Space Agency”
17. List the number of satellites for each planet with a planet name.
18. List all the experiments having start date between “6/9/2009” and “6/20/2009”
19. List mission profile,satellite name connected to space agency with agency name.
20. What is the satellite update version of Moon in “9/2/2020”
21. What is threat status for asteroid “4 vesta”
22. Get the mission details and corresponding satellite id for all the future launch missions (hint mission_status “SCHEDULED” as we are considering future missions)
23. List all the phone numbers with name of personnel whose age is between 25 to 35.
24. List all asteroid name related threat status.

25. Display a warning message if a new threat is added which is going to happen in the next one month. Preferably create an INSERT trigger.
26. Create a function which displays all Mars missions updates for experiments whose outcomes are still “Pending” along with associated satellites.
27. Get all the personnel names and designations who is working on the satellite “Gaofen-1-04”
28. Create a trigger that checks whether the primary key exists or not in that table if not then prints it.
29. Create a trigger that checks foreign key exists or not in the referenced table if not then prints it.
30. List star id from star where classification is K and shape is Decagram.
31. List all the health percentages of satellites orbiting Mars.
32. List the mission and its experiments details where the mission was completed and successful.
33. List Scientist name and assigned blackhole name.
34. Sort the personnel details by its personnel salary.
35. Which personnel are working on a ‘M’ type star?
36. Get all the satellite ids whose data is being shared to Australian Space Agency.
37. Write a query when all the satellite data being shared with the UK space agency is immediately halted due to security concerns.
38. Delete missions updates which missions are canceled by agency.
39. Update a salary by 10000 of a personnel whose mission is completed.
40. List all planets name which have negative temperature in decreasing order by mass

1)Select * from “Missions”.missions

The screenshot shows the pgAdmin 4 interface. In the left sidebar, under the 'Schemas' section, the 'ISRO_Agency' schema is selected. The 'missions' table is listed under it. A query window titled 'Query - ISRO_Agency on postgres@PostgreSQL 10' contains the SQL command:

```
1 select * from "Missions".missions
```

The results are displayed in a table format:

	mission_id	launch_vehicle	ground_segment	mission_profile	payload_details	start_date	end_date	status
	integer	character varying	character varying	character varying	character varying	date	date	character varying
1	10078	GSLV Mk II	ATLAS Global Netwo...	ISS assembly flight ...	Neuron Spectromete...	2016-06-17	2023-03-14	INACTIVE
2	10076	Atlas SLV-3	ATLAS Global Netwo...	Explore ice caps of ...	Hyperspectral Image...	2009-10-06	1996-08-28	COMPLETED
3	10062	GSLV Mk I	ISIS Small satellite G...	ISS assembly flight ...	Dual Frequency Synt...	2014-12-24	2003-09-28	INACTIVE
4	10001	PSLV-G	OSAGS	Collect sample from ...	Lunar Gravity Mapper	2014-09-18	1996-05-20	SCHEDULED
5	10070	GSLV Mk II	GAMALINK	ISS assembly flight ...	Lunar Gravity Mapper	1999-09-10	2004-05-27	COMPLETED
6	10077	Atlas LV-3B	KSAT Lite	Study venus atmosph...	Dual Frequency Synt...	2019-12-21	2013-08-29	SCHEDULED

2) e ground_segment like ‘O%’

SPACE_DATABASE/postgres@PostgreSQL 10

Query Editor Query History

```
1 select * from "Missions".missions where ground_segment like '%%'
```

Data Output Explain Messages Notifications

	mission_id	launch_vehicle	ground_segment	mission_profile	payload_details	start_date	end_date	status
1	10001	PSLV-G	OSAGS	Collect sample from A...	Lunar Gravity Mapper	2014-09-18	1996-05-20	SCHEDULED
2	10024	PSLV-XL	OSAGS	mapping composition...	Lunar Gravity Digicon...	1996-03-11	2008-04-12	COMPLETED
3	10038	PSLV-XL	OSAGS	Observe radio signals f...	Lunar Gravity Digicon I...	2000-06-23	2020-01-02	INACTIVE
4	10000	PSLV-CA	OSAGS	Tracking and data relay...	Cross-track Infrared S...	2018-12-07	2023-02-28	COMPLETED
5	10000	Ares IV	OSAGS	Explore moons of satur...	Methane Sensor,Mars ...	1998-09-24	2002-10-25	INACTIVE
6	10045	Ares II	OSAGS	ISS assembly flight UL...	Lunar Gravity Digicon I...	2020-09-26	2013-11-15	COMPLETED
7	10079	Ares IV	OSAGS	Study venus atmospher...	Neuron Spectrometer,...	2000-08-09	2019-03-08	ACTIVE
8	10050	M-3SI	OSAGS	CHEOPS: Measure dia...	Methane Sensor,Mars ...	2024-01-31	2011-07-11	ACTIVE
9	10042	Atlas LV-3B	OSAGS	Study GH123 and SI14...	Methane Sensor,Mars ...	2011-05-30	2015-08-10	COMPLETED
10	10079	PSLV-CA	OSAGS	Study atmosphere of E...	Solar X-ray monitor,Or...	2000-06-09	2006-01-16	INACTIVE

3)select * from "Satellites".Satellites where orbit_velocity<7000

TimeTable x | (10) WhatsApp x | Lab_10_DB_Project_Assignments pgAdmin 4 x | Lab10 - Google Docs x + - o

127.0.0.1:52295/browser/

pgAdmin File Object Tools Help

Browser Servers (1) PostgreSQL 10 Databases (3) Space Agency (2) 201801180.db Schemas (6) Agency and personnel Missions PABS Satellites Threats public postres Login/Group Roles Tablespace

Space Agency /postgres@PostgreSQL 10

Query Editor Query History

```
1 select * from "Satellites".Satellites where orbit_velocity<7000
```

Data Output Explain Messages Notifications

	satellite_id	satellite_name	orbit_details	orbit_distance	orbit_velocity	orbit_period	navigation_system	number_of_malfunctioning_parts
1	541	CSG-1 (COSMO-SkyMe... Non-Polar Inclined		959	6493	94.6	QZSS	
2	588	Gaofen-1-04 Sun-Synchronous		580	6222	100.4	GPS	
3	556	Grace Follow-on-1 (Gra... Non-Polar Inclined		552	6444	94.85	Galileo	
4	515	O3b FM06 Sun-Synchronous		1346	6249	90.3	GPS	
5	513	Starlink-1166 Sun-Synchronous		818	6505	90.3	GLONASS	
6	519	Lemur-2-SarahBettyBo... Polar		1121	6911	94.5	BeiDou	
7	571	Starlink-1172 Sun-Synchronous		1231	6374	95.13	GPS	
8	539	Galaxy-19 Non-Polar Inclined		969	6108	100.4	BeiDou	Activate Windows
9	509	Starlink-1193 Sun-Synchronous		767	6762	93.8	BeiDou	Go to Settings to activate Windows.

4)

select * from "Satellites".satellites
where "Satellites".satellites.planet_id = 405

```

1 select * from "Satellites".satellites
2 where "Satellites".satellites.planet_id = 405

```

	satellite_id	satellite_name	orbit_details	orbit_distance	orbit_velocity	orbit_period	navigation_system	num
1	571	Starlink-1172	Sun-Synchronous	1231	6374	95.13	GPS	
2	563	Starlink-1009	Sun-Synchronous	949	6832	116.04	QZSS	
3	564	Jilin-1 (High Resoluti...	Non-Polar Inclined	1436	7501	1436.03	NAVIC	
4	514	Lemur-2-Pappy (Lem...	Non-Polar Inclined	1001	7884	96.47	NAVIC	
5	555	Jilin-1-5 (Shipin 5)	Deep Highly Eccentric	1342	7149	1436.1	Galileo	
6	558	Starlink-74	Non-Polar Inclined	1166	6799	100.53	QZSS	
7	544	TianYi-15 (Xiaoxiang...	Sun-Synchronous	896	7780	115.8	GLONASS	
8	500	HYLAS 4 (Highly Adap...	Polar	1077	7162	100	GLONASS	
9	575	Worldview 2	Polar	817				

Successfully run. Total query runtime: 212 msec. 11 rows affected.

5)

select satellite_name from "Satellites".Satellites
where orbit_distance >(select avg(orbit_distance) from "Satellites".Satellites)

```

1 select satellite_name from "Satellites".Satellites
2 where orbit_distance >(select avg(orbit_distance) from "Satellites".Satellites)

```

	satellite_name
1	COSMIC 2-3 (COSMI...
2	Starlink-1043
3	Starlink-1064
4	O3b FM06
5	Lemur-2-SarahBetty...
6	Globalstar M070 (Gl...
7	Starlink-1172
8	Starlink-1073
9	Meridian-3
10	Navstar PIR-M-2 ...

Successfully run. Total query runtime: 230 msec. 48 rows affected.

6)

select name,designation from "Agency and personnel".isro_personnel where start_date<'2010-01-01'

The screenshot shows the pgAdmin 4 interface. On the left, the 'Browser' pane displays a database structure with 'Tables (3)' selected. In the center, the 'Query Editor' pane contains the following SQL query:

```
1 select name,designation from "Agency and personnel".isro_personnel where start_date<'2010-01-01'
```

The 'Data Output' pane below shows the results of the query:

	name	designation
1	Ram Narain Agrawal	Technical Officer-E...
2	Thekkethil Kochand...	Secretary
3	Mylswamy Annadur...	Member (Finance) ...
4	M. Annamalai	Technical Officer-E ...
5	Anuradha TK	Library Officer-E ...
6	R. Aravananudan	Controller
7	Subbiah Arunan	Junior Engineer, ...
8	John Bamabas	Controller
9	Anil Bhardwaj	Sr. Head
10	Preful Bhavasar	Member (Finance) ...

A green message bar at the bottom right indicates: 'Successfully run. Total query runtime: 773 msec. 69 rows affected.'

7)

select isro_personnel.id, personnel_details.phone_number from "Agency and personnel".isro_personnel
join "PABS".blackhole_studied on
isro_personnel.id=blackhole_studied.personnel_id
join "Agency and personnel".personnel_details on
isro_personnel.id=personnel_details.personnel_id
Where "Agency and personnel".isro_personnel.salary>50000

```

select isro_personnel.id, personnel_details.phone_number from "Agency and personnel".isro_personnel
join "PABS".blackhole_studied on isro_personnel.id=blackhole_studied.personnel_id
join "Agency and personnel".personnel_details on isro_personnel.id=personnel_details.personnel_id
where "Agency and personnel".isro_personnel.salary>50000

```

Data Output	Explain	Messages	Notifications																										
<table border="1"> <thead> <tr> <th>id</th> <th>phone_number</th> </tr> <tr> <th>integer</th> <th>character (10)</th> </tr> </thead> <tbody> <tr><td>1</td><td>101</td></tr> <tr><td>2</td><td>101</td></tr> <tr><td>3</td><td>103</td></tr> <tr><td>4</td><td>103</td></tr> <tr><td>5</td><td>103</td></tr> <tr><td>6</td><td>106</td></tr> <tr><td>7</td><td>106</td></tr> <tr><td>8</td><td>109</td></tr> <tr><td>9</td><td>109</td></tr> <tr><td>10</td><td>110</td></tr> <tr><td>11</td><td>110</td></tr> </tbody> </table>	id	phone_number	integer	character (10)	1	101	2	101	3	103	4	103	5	103	6	106	7	106	8	109	9	109	10	110	11	110			
id	phone_number																												
integer	character (10)																												
1	101																												
2	101																												
3	103																												
4	103																												
5	103																												
6	106																												
7	106																												
8	109																												
9	109																												
10	110																												
11	110																												

8.select count(*) from "Threats".collision_threat where threat_status = 'Tackled/Over'

```

select count(*) from "Threats".collision_threat where threat_status = 'Tackled/Over'

```

Data Output	Explain	Messages	Notifications				
<table border="1"> <thead> <tr> <th>count</th> </tr> <tr> <th>bigint</th> </tr> </thead> <tbody> <tr><td>1</td></tr> <tr><td>23</td></tr> </tbody> </table>	count	bigint	1	23			
count							
bigint							
1							
23							

9.select designation, avg(salary) from "Agency and personnel".isro_personnel
group by designation

pgAdmin 4 - ISRO_Agency on postgres@PostgreSQL 10

```
1 select designation, avg(salary) from "Agency and personnel".isro_personnel
2 group by designation
3
```

Data Output Explain Messages Notifications

designation	avg
character (50)	numeric
Joint Director (OL)	244444.4444444444
Junior Engineer,	250000.00000000000000
Technical Officer-E	125000.00000000000000
Secretary	193750.00000000000000
Director-DOS	255555.55555555556
Technical Officer-E	137500.00000000000000
Registrar	150000.00000000000000
Controller	200000.00000000000000
Library Officer-E	209090.909090909091
Assistant Engineer	266666.66666666667
Dy. Secretary	200000.00000000000000

Successfully run. Total query runtime: 178 msec. 13 rows affected.

10)select * from "PABS".asteroids
order by mass desc
limit 10

pgAdmin 4 - ISRO_Agency on postgres@PostgreSQL 10

```
1 select * from "PABS".asteroids
2 order by mass desc
3 limit 10
```

Data Output Explain Messages Notifications

asteroid_id	asteroid_name	diameter	mass	spectral_class
bigint	character varying	bigint	bigint	character (50)
1070	Miller Range 0515	819	49973	G
1007	Rio Rancho	856	49321	P
1080	Tanezrouft 079	403	48773	B
1009	Northwest Africa 14...	918	48507	S
1049	Hammadah al Hamr...	610	48060	P
1016	Salar de Imlil	578	47953	C
1039	Northwest Africa 42...	147	47608	P
1086	Sayh al Uhaymir 269	677	47143	S
1085	Dhofar 396	456	47108	D
1011	Valdininza	874	45905	B

Successfully run. Total query runtime: 201 msec. 10 rows affected.

11)
create or replace function imminentf()
returns table

```

(
    tracking_id integer, threat_status varchar, date_time date,
personnel_name char(50),
    personnel_id integer, asteroid_id integer, mass bigint
)
LANGUAGE 'plpgsql'
AS $BODY$
BEGIN
return query

select "Threats".collision_threat.tracking_id,
"Threats".collision_threat.threat_status,
"Threats".collision_threat.date_time, "Agency and
personnel".isro_personnel.name,
"Agency and personnel".isro_personnel.id, "PABS".asteroids.asteroid_id,
"PABS".asteroids.mass
from "Threats".collision_threat

join "PABS".asteroids on
"Threats".collision_threat.asteroid_id="PABS".asteroids.asteroid_id
join "PABS".asteroid_studied on
collision_threat.asteroid_id="PABS".asteroid_studied.asteroid_id
join "Agency and personnel".isro_personnel on
"Agency and
personnel".isro_personnel.id="PABS".asteroid_studied.personnel_id

where (collision_threat.date_time between' 28-11-2020' and '28-05-2021');

END
$BODY$;

```

SPACE_DATABASE/postgres@PostgreSQL 10

Query Editor Query History

```

1 create or replace function imminentf()
2 returns table
3 (
4     tracking_id integer, threat_status varchar, date_time date, personnel_name char(50),
5     personnel_id integer, asteroid_id integer, mass bigint
6 )
7
8 LANGUAGE 'plpgsql'
9 AS $BODYS|
10 BEGIN
11 return query
12
13 select "Threats".collision_threat.tracking_id, "Threats".collision_threat.threat_status,
14 "Threats".collision_threat.date_time, "Agency and personnel".isro_personnel.name,
15 "Agency and personnel".isro_personnel.id, "PABS".asteroids.asteroid_id, "PABS".asteroids.mass
16 from "Threats".collision_threat
17
18 join "PABS".asteroids on "Threats".collision_threat.asteroid_id="PABS".asteroids.asteroid_id
19 join "PABS".asteroid_studied on collision_threat.asteroid_id="PABS".asteroid_studied.asteroid_id
20 join "Agency and personnel".isro_personnel on
21 "Agency and personnel".isro_personnel.id="PABS".asteroid_studied.personnel_id
22

```

Data Output Explain Messages Notifications

tracking_id	threat_status	date_time	personnel_name	personnel_id	asteroid_id	mass	
1	Tackled/Over	2021-05-18	Satish Dhawan	...	119	1018	24462
2	Imminent	2020-12-11	MS Pannirselvam	...	159	1058	43503
3	Simple Warning	2021-04-26	V. Adimurthy	...	164	1063	37487

12)

`select * from "PABS".stars`

`where au_distance = (select max(au_distance) from "PABS".stars)`

127.0.0.1:53674/browser/

pgAdmin 4

File ▾ Object ▾ Tools ▾ Help ▾

ISRO_Agency on postgres@PostgreSQL 10

Query Editor Query History

```

1 select * from "PABS".stars
2 where au_distance = (select max(au_distance) from "PABS".stars)

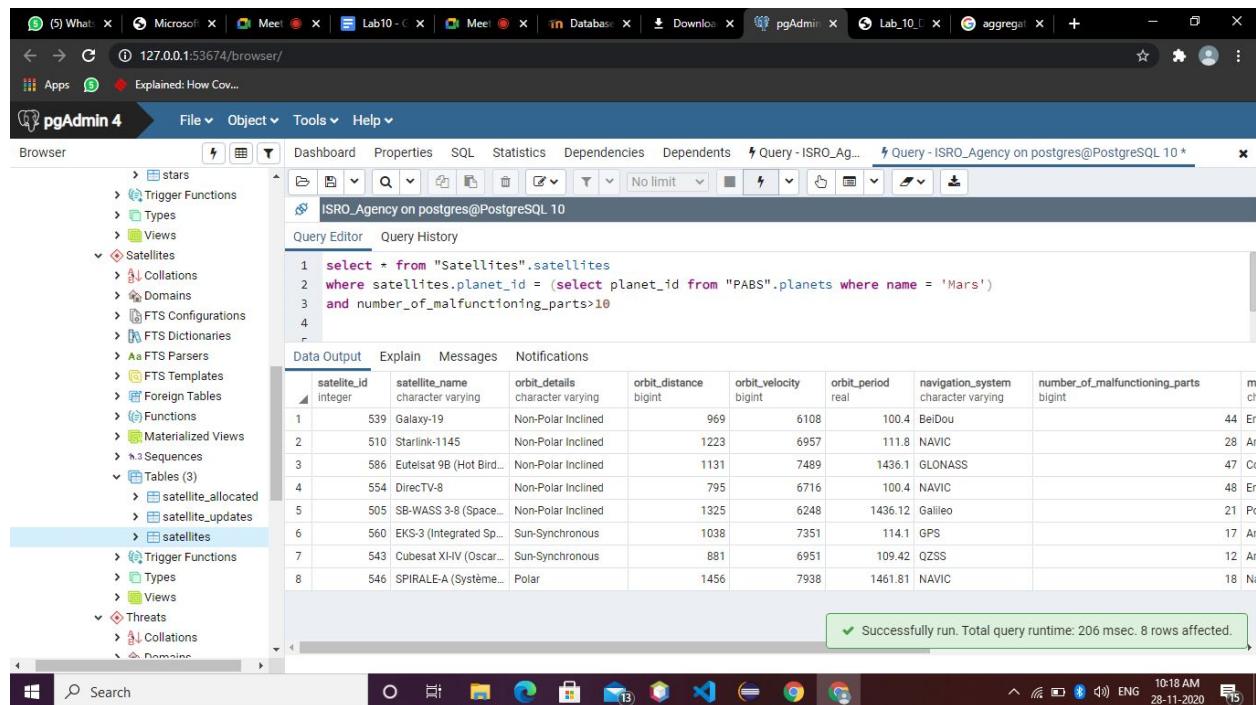
```

Data Output Explain Messages Notifications

star_id	classification	size	shape	au_distance	
1	A	...	4314433	Heptagram	4963905

Successfully run. Total query runtime: 182 msec. 1 rows affected.

13)select * from "Satellites".satellites
 where satellites.planet_id = (select planet_id from "PABS".planets where name = 'Mars')
 and number_of_malfunctioning_parts>10



The screenshot shows the pgAdmin 4 interface with a query editor window. The query is:

```

1 select * from "Satellites".satellites
2 where satellites.planet_id = (select planet_id from "PABS".planets where name = 'Mars')
3 and number_of_malfunctioning_parts>10
4
  
```

The results table has the following columns and data:

	satellite_id	satellite_name	orbit_details	orbit_distance	orbit_velocity	orbit_period	navigation_system	number_of_malfunctioning_parts	m ch
1	539	Galaxy-19	Non-Polar Inclined	969	6108	100.4	BeiDou		44 En
2	510	Starlink-1145	Non-Polar Inclined	1223	6957	111.8	NAVIC		28 Ar
3	586	Eutelsat 9B (Hot Bird)	Non-Polar Inclined	1131	7489	1436.1	GLONASS		47 Cc
4	554	DirectTV-8	Non-Polar Inclined	795	6716	100.4	NAVIC		48 En
5	505	SB-WASS 3-8 (Space...	Non-Polar Inclined	1325	6248	1436.12	Galileo		21 Pd
6	560	EKS-3 (Integrated Sp...	Sun-Synchronous	1038	7351	114.1	GPS		17 Ar
7	543	Cubesat XI-IV (Oscar...	Sun-Synchronous	881	6951	109.42	QZSS		12 Ar
8	546	SPIRALE-A (Système...	Polar	1456	7938	1461.81	NAVIC		18 Nt

Message bar: Successfully run. Total query runtime: 206 msec. 8 rows affected.

14)
 select satellite_name from "Satellites".satellites
 join "Agency and personnel".space_agencies on
 "Agency and personnel".space_agencies.satellite_id =
 "Satellites".satellites.satellite_id
 where "Agency and personnel".space_agencies.tracking_id = '1020'

SPACE/postgres@PostgreSQL 10

Query Editor Query History

```

1 select satellite_name from "Satellites".satellites
2 join "Agency and personnel".space_agencies on
3 "Agency and personnel".space_agencies.satellite_id = "Satellites".satellites.satelite_id
4 where "Agency and personnel".space_agencies.tracking_id = '1020'

```

Notifications

Recorded time	Event	Process ID	Payload
No data found			

Data Output Explain Messages

satellite_name	
character varying	
1	OneWeb-0013

15)

```

select name,mission_profile,launch_vehicle from
"Missions".missions
join "Missions".assigned on "Missions".missions.mission_id =
"Missions".assigned.mission_id
join "Agency and personnel".isro_personnel
on
"Agency and personnel".isro_personnel.id =
"Missions".assigned.personnel_id

```

where status = 'ACTIVE'

Dashboard Properties SQL Statistics Dependencies Dependents Query - ISRO_Ag... Query - ISRO_Agency on postgres@PostgreSQL 10 *

ISRO_Agency on postgres@PostgreSQL 10

Query Editor Query History

```

1 select name,mission_profile,launch_vehicle from
2 "Missions".missions
3 join "Missions".assigned on "Missions".missions.mission_id = "Missions".assigned.mission_id
4 join "Agency and personnel".isro_personnel
5 on
6 "Agency and personnel".isro_personnel.id = "Missions".assigned.personnel_id
7

```

Data Output Explain Messages Notifications

name	mission_profile	launch_vehicle
character (50)	character varying	character varying
1 R. Aravamudan	Study GH123 and SL...	Ares I
2 Subbiah Aruman	Study working of Las...	GSLV MK II
3 Vasant Gowarikar...	mapping composition...	Ares II
4 T. S. Prahlad	Experiments aboard ...	GSLV Mk III
5 M. V. S. Prasad	Study working of Las...	Ares I
6 Janvi Patel	Study atmosphere of ...	PSLV-XL
7 Divyesh Rohit	Study the rings of Ne...	PSLV-3S
8 S Ramakrishnan	Study venus atmosph...	Ares IV
9 Ritu Karidhal	CHEOPS: Measure di...	M-3SII

Successfully run. Total query runtime: 163 msec. 26 rows affected.

16)

```
select mission_profile,payload_details from "Missions".missions
join "Agency and personnel".space_agencies on
"Agency and personnel".space_agencies.mission_id =
"Missions".missions.mission_id
where "Agency and personnel".space_agencies.Agencies_Name =
'Australian Space Agency'
```

The screenshot shows a PostgreSQL query editor interface. At the top, it displays the connection information: SPACE/postgres@PostgreSQL 10. Below this, there are tabs for 'Query Editor' (which is selected) and 'Query History'. The main area contains the SQL query:

```
1 select mission_profile,payload_details from "Missions".missions
2 join "Agency and personnel".space_agencies on
3 "Agency and personnel".space_agencies.mission_id = "Missions".missions.mission_id
4 where "Agency and personnel".space_agencies.Agencies_Name = 'Australian Space Agency'|
```

Below the query, there are sections for 'Notifications' (empty), 'Recorded time', 'Event', 'Process ID', and 'Payload' (also empty). A message 'No data found' is displayed. At the bottom, there are tabs for 'Data Output' (selected), 'Explain', and 'Messages'. The 'Data Output' tab shows a table with two columns: 'mission_profile' and 'payload_details'. The first row contains the value 'ISS assembly flight UL...' under 'mission_profile' and 'Neuron Spectrometer, ...' under 'payload_details'.

17. select "PABS".planets.name,count(*) from "Satellites".satellites join
"PABS".planets on "PABS".planets.planet_id="Satellites".satellites.planet_id
group by "PABS".planets.planet_id

The screenshot shows the pgAdmin 4 interface. On the left, the Browser pane displays a tree view of database objects under the schema 'Space Agency'. In the center, the Query Editor pane contains the following SQL query:

```

1 select "PABS".planets.name,count(*) from "Satellites".satellites join
2 "PABS".planets on "PABS".planets.planet_id="Satellites".satellites.planet_id group by "PABS".plan
3

```

The Data Output tab shows the results of the query:

	name	character varying	count	digit
1	Uranus		11	
2	Venus		11	
3	Saturn		11	
4	Jupiter		11	
5	Neptune		11	
6	Mercury		12	
7	Pluto		10	
8	Moon		11	
9	Mars		11	

18)

select experiments from "Missions".mission_updates
join "Missions".missions on
"Missions".missions.mission_id = "Missions".mission_updates.mission_id
where "Missions".missions.start_date between '6/9/2009' and '6/20/2009'

The screenshot shows the pgAdmin 4 interface. On the left, the Browser pane displays a tree view of database objects under the schema 'SPACE'. In the center, the Query Editor pane contains the following SQL query:

```

1 select experiments from "Missions".mission_updates
2 join "Missions".missions on
3 "Missions".missions.mission_id = "Missions".mission_updates.mission_id
4 where "Missions".missions.start_date between '6/9/2009' and '6/20/2009'

```

The Data Output tab shows the results of the query:

	experiments	character varying
5	Sols 2919-2920: Penultimate Plan	
6	Sols 2940-2941: Curiosity Eyes a Comfortable 'Bench' to Park on for the Weekend	
7	Sol 2925: 'Maybole' Up Close and Personal	
8	Sols 2914-2915: Curiosity Breaking Open the Mystery of the 'Groken' Nodules	
9	Sols 2914-2916: Curiosity Breaking Open the Mystery of the 'Groken' Nodules	
10	Sols 2951-2953: Pre-Holiday Scramble	
11	Sol 2918: A Short but Sweet Day of Planning	
12	Sols 2938-2939: Eyes on the Prize	

19)

```

select agencies_name,satellite_name,mission_profile from "Agency and personnel".space_agencies
join "Missions".missions
on "Missions".missions.mission_id="Agency and personnel".space_agencies.mission_id
join "Satellites".satellites on
"Satellites".satellites.satellite_id= "Agency and personnel".space_agencies.satellite_id

```

The screenshot shows a PostgreSQL query editor window titled 'ISRO_Agency on postgres@PostgreSQL 10'. The query is:

```

1 select agencies_name,satellite_name,mission_profile from "Agency and personnel".space_agencies
2 join "Missions".missions
3 on "Missions".missions.mission_id="Agency and personnel".space_agencies.mission_id
4 join "Satellites".satellites on
5 "Satellites".satellites.satellite_id= "Agency and personnel".space_agencies.satellite_id
6
7

```

The results table has three columns: 'agencies_name' (character varying), 'satellite_name' (character varying), and 'mission_profile' (character varying). The data is as follows:

agencies_name	satellite_name	mission_profile
Australian Space Ag...	CSG-1 (COSMO-Sky...	ISS assembly flight ...
Mohammed bin Ras...	Gaofen-1-04	Explore ice caps of ...
Belarus Space Agency	Astrocast 0.1	ISS assembly flight ...
United Arab Emirate...	COSMIC 2-3 (COSMI...	Collect sample from ...
Central American As...	Starlink-1043	ISS assembly flight ...
United Nations Com...	Swift	Study venus atmosph...
Française Space Agen...	Orion Emissary 1/2	Studying CMEs and CI

A green message bar at the bottom right indicates: 'Successfully run. Total query runtime: 265 msec. 75 rows affected.'

20)

```

select update from "Satellites".satellite_updates
join "Satellites".satellites on
"Satellites".satellites.satellite_id = "Satellites".satellite_updates.satellite_id
join "PABS".planets on
"PABS".planets.planet_id = "Satellites".satellites.planet_id
where "PABS".planets.name = 'Moon' and
"Satellites".satellite_updates.date_time = '9/2/2020'

```

SPACE/postgres@PostgreSQL 10

Query Editor Query History

```

1 select update from "Satellites".satellite_updates
2 join "Satellites".satellites on
3 "Satellites".satellites.satellite_id = "Satellites".satellite_updates.satelite_id
4 join "PABS".planets on
5 "PABS".planets.planet_id = "Satellites".satellites.planet_id
6 where "PABS".planets.name = 'Moon' and "Satellites".satellite_updates.date_time = '9/2/2020'

```

Notifications

Recorded time	Event	Process ID	Payload
No data found			

Data Output Explain Messages

	update	character varying	lock
1	Satellite	6.7.3 Update	

21)

```

select threat_status from "Threats".collision_threat
join "PABS".asteroids on
"PABS".asteroids.asteroid_id = "Threats".collision_threat.asteroid_id
where "PABS".asteroids.asteroid_name = '4 vesta'

```

SPACE/postgres@PostgreSQL 10

Query Editor Query History

```

1 select threat_status from "Threats".collision_threat
2 join "PABS".asteroids on
3 "PABS".asteroids.asteroid_id = "Threats".collision_threat.asteroid_id
4 where "PABS".asteroids.asteroid_name = '4 vesta'

```

Notifications

Recorded time	Event	Process ID	Payload
No data found			

Data Output Explain Messages

	threat_status	character (50)	lock
1	Tackled/Over	...	

22)

```

create or replace function futuremissions()
returns table

```

```
(  
    satellite_id integer, mission_id integer, mission_profile varchar,  
    payload_details varchar, start_date date,  
    status varchar  
)  
  
LANGUAGE 'plpgsql'  
AS $BODY$  
BEGIN  
return query  
  
select "satellite".satellites.satellite_id, "Missions".missions.mission_id,  
"Missions".missions.mission_profile,  
"Missions".missions.payload_details, "Missions".missions.start_date,  
"Missions".missions.status  
from "Missions".missions  
  
join "satellite".satellite_allocated on  
"satellite".satellite_allocated.mission_id="Missions".missions.mission_id  
join "satellite".satellites on  
"satellite".satellites.satellite_id = "satellite".satellite_allocated.satellite_id  
  
where (missions.start_date>now() and missions.status='SCHEDULED');  
  
END  
$BODY$;
```

SPACE_DATABASE/postgres@PostgreSQL 10

Query Editor Query History

```

1  create or replace function futuremissions()
2  returns table
3  (
4      satellite_id integer, mission_id integer, mission_profile varchar, payload_details varchar, start_date date,
5      status varchar
6  )
7
8  LANGUAGE 'plpgsql'
9  AS $BODY$
10 BEGIN
11     return query
12
13     select "satellite".satellites.satellite_id, "Missions".missions.mission_id, "Missions".missions.mission_profile,
14     "Missions".missions.payload_details, "Missions".missions.start_date, "Missions".missions.status
15     from "Missions".missions
16
17     join "satellite".satellite_allocated on
18     "satellite".satellite_allocated.mission_id="Missions".missions.mission_id
19     join "satellite".satellites on
20     "satellite".satellites.satellite_id = "satellite".satellite_allocated.satellite_id
21
22

```

Data Output Explain Messages Notifications

satellite_id	mission_id	mission_profile	payload_details	start_date	status	
1	583	10082	Collect sample from A...	Neuron Spectrometer, ...	2024-09-21	SCHEDULED
2	542	10084	ISS assembly flight UL...	Cadmium Zinc Tellurid...	2021-09-04	SCHEDULED

23)

```

select phone_number, "Agency and personnel".isro_personnel.name from
"Agency and personnel".personnel_details
join "Agency and personnel".isro_personnel on
"Agency and personnel".isro_personnel.id = "Agency and
personnel".personnel_details.personnel_id
where "Agency and personnel".isro_personnel.age between '25' and '35'

```

The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of database objects under the 'Browser' tab, including Schemas, Tables, and Views. The 'Tables' section is currently selected. The main area contains a 'Query Editor' tab with the following SQL code:

```

1 select phone_number, "Agency and personnel".isro_personnel.name from "Agency and personnel".personnel_det
2 join "Agency and personnel".isro_personnel on
3 "Agency and personnel".isro_personnel.id = "Agency and personnel".personnel_details.personnel_id
4 where "Agency and personnel".isro_personnel.age between '25' and '35'

```

Below the query editor is a 'Notifications' section with a table showing recorded time, event, process ID, and payload. The table is empty with the message 'No data found'. The 'Data Output' tab is selected, displaying a table with columns 'phone_number' and 'name'. The data rows are:

	phone_number	name
22	9910855275	Thekkethil Kochan...
23	9793444266	Satish Dhawan ...
24	9470201212	Ramanujam Varath...
25	9720528672	M. Pitchaimani ...
26	9409855263	B Jayakumar ...
27	9995692545	MS Pannirselvam ...
28	9470002467	MYS Prasad ...

24)select asteroid_name,threat_status from "Threats".collision_threat join "PABS".asteroids
on "Threats".collision_threat.asteroid_id= "PABS".asteroids.asteroid_id

The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of database objects under the 'Browser' tab, including Schemas, Tables, and Views. The 'Tables' section is currently selected. The main area contains a 'Query Editor' tab with the following SQL code:

```

1 select asteroid_name,threat_status from "Threats".collision_threat join "PABS".asteroids
2 on "Threats".collision_threat.asteroid_id= "PABS".asteroids.asteroid_id
3

```

Below the query editor is a 'Data Output' tab with a table showing columns 'asteroid_name' and 'threat_status'. The data rows are:

	asteroid_name	threat_status
1	Paposo 008	Tackled/Over ...
2	Northwest Africa 2226	Tackled/Over ...
3	Tanezrouft 079	Low Tier ...
4	4 vesta	Tackled/Over ...
5	Asuka 881850	Medium Tier ...
6	Northwest Africa 7479	Simple Warning ...
7	Rooikop 001	Medium Tier ...
8	Dar al Gani 148	Tackled/Over ...
9	Northwest Africa 764	Medium Tier ...
10	Queen Alexandra Ranq...	Imminent ...

A green notification bar at the bottom right indicates: 'Successfully run. Total query runtime: 263 msec. 100 rows affected.'

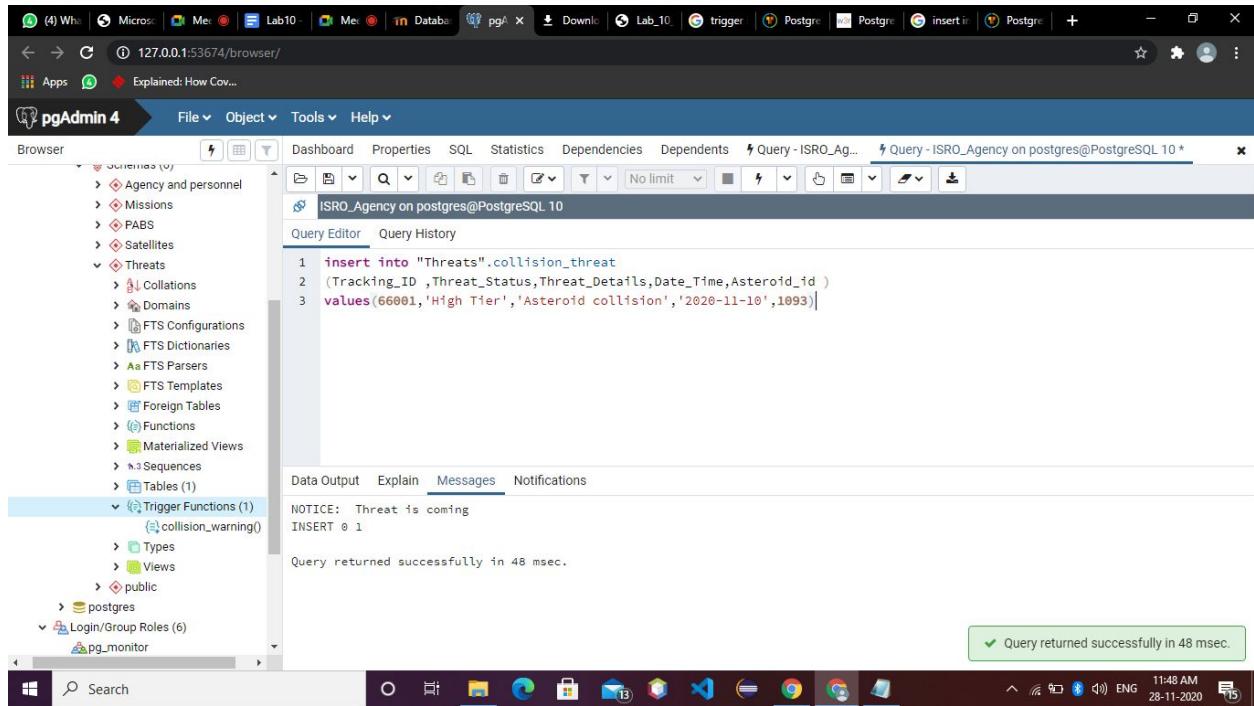
25)

CREATE OR REPLACE FUNCTION "Threats".collision_warning()
RETURNS trigger AS

```
$BODY$  
BEGIN  
  IF (new.date_time-current_date<=30) THEN  
    raise notice 'Threat is coming';  
  
  end if;  
  return new;  
  
END  
$BODY$  
LANGUAGE plpgsql;
```

```
create trigger collision_warn  
after insert  
on "Threats".collision_threat  
for each row  
execute procedure "Threats".collision_warning()
```

```
insert into "Threats".collision_threat  
(Tracking_ID ,Threat_Status,Threat_Details,Date_Time,Asteroid_id )  
values(66001,'High Tier','Asteroid collision','2020-11-10',1093)
```



26)

create or replace function mission_updates()

returns table

(

mission_id integer, date_time date, experiments varchar, planet varchar,
outcomes varchar, sat_id integer,

sat_name varchar

)

LANGUAGE 'plpgsql'

AS \$BODY\$

BEGIN

return query

```

select "Missions".mission_updates.mission_id,
"Missions".mission_updates.date_time,
"Missions".mission_updates.experiments, "PABS".planets.name,
"Missions".mission_updates.outcomes, "satellite".satellites.satellite_id,
"satellite".satellites.satellite_name
from "Missions".mission_updates

```

```

join
  "satellite".satellite_allocated
on
  "satellite".satellite_allocated.mission_id="Missions".mission_updates.mission
_id

join "satellite".satellites
on "satellite".satellites.satellite_id="satellite".satellite_allocated.satellite_id

join "PABS".planets
on "PABS".planets.planet_id="satellite".satellites.planet_id

where "PABS".planets.name='Mars' and
  "Missions".mission_updates.outcomes='PENDING';

END
$BODY$;

```

The screenshot shows a PostgreSQL query editor window titled 'SPACE_DATABASE/postgres@PostgreSQL 10'. The code area contains the following PL/pgSQL function definition:

```

1 create or replace function mission_updates()
2 returns table
3 (
4   mission_id integer, date_time date, experiments varchar, planet varchar, outcomes varchar, sat_id integer,
5   sat_name varchar
6 )
7
8 LANGUAGE 'plpgsql'
9 AS $BODY$
10 BEGIN
11   return query
12
13   select "Missions".mission_updates.mission_id, "Missions".mission_updates.date_time,
14   "Missions".mission_updates.experiments, "PABS".planets.name,
15   "Missions".mission_updates.outcomes, "satellite".satellites.satellite_id, "satellite".satellites.satellite_name
16   from "Missions".mission_updates
17
18   join
19     "satellite".satellite_allocated
20   on "satellite".satellite_allocated.mission_id="Missions".mission_updates.mission_id
21
22 $BODY$;

```

The 'Data Output' tab is selected, showing the schema of the function's return table:

mission_id	date_time	experiments	planet	outcomes	sat_id	sat_name
integer	date	character varying	character varying	character varying	integer	character varying

27)

select "Agency and personnel".isro_personnel.name,designation

```

from "Agency and personnel".isro_personnel
join "PABS".planet_studied on
"PABS".planet_studied.personnel_id = "Agency and
personnel".isro_personnel.id
join "PABS".planets on
"PABS".planets.planet_id = "PABS".planet_studied.planet_id
join "Satellites".satellites on
"Satellites".satellites.planet_id = "PABS".planets.planet_id
where "Satellites".satellites.satellite_name = 'Gaofen-1-04'

```

name	designation	
Ritu Kirdhal	Secretary	
S Ramakrishnan	Member (Finance)	
Chandradathan	Assistant Engineer	
MYS Prasad	Joint Director (OL)	
R. Aravamudan	Dy. Secretary	
SK Shivakumar	Secretary	
Jitendra Nath Gos...	Director-DOS	
Nandini Harinath	Technical Officer-E...	

28)

```

CREATE OR REPLACE FUNCTION "Satellites".primary_key()
RETURNS trigger AS
$BODY$
BEGIN
  IF (SELECT COUNT(*) FROM "Satellites".satellites WHERE
  satellites.satelite_id = NEW.satelite_id) > 0 THEN
    raise notice 'Not able to insert';
  else
    return new;
  end if;

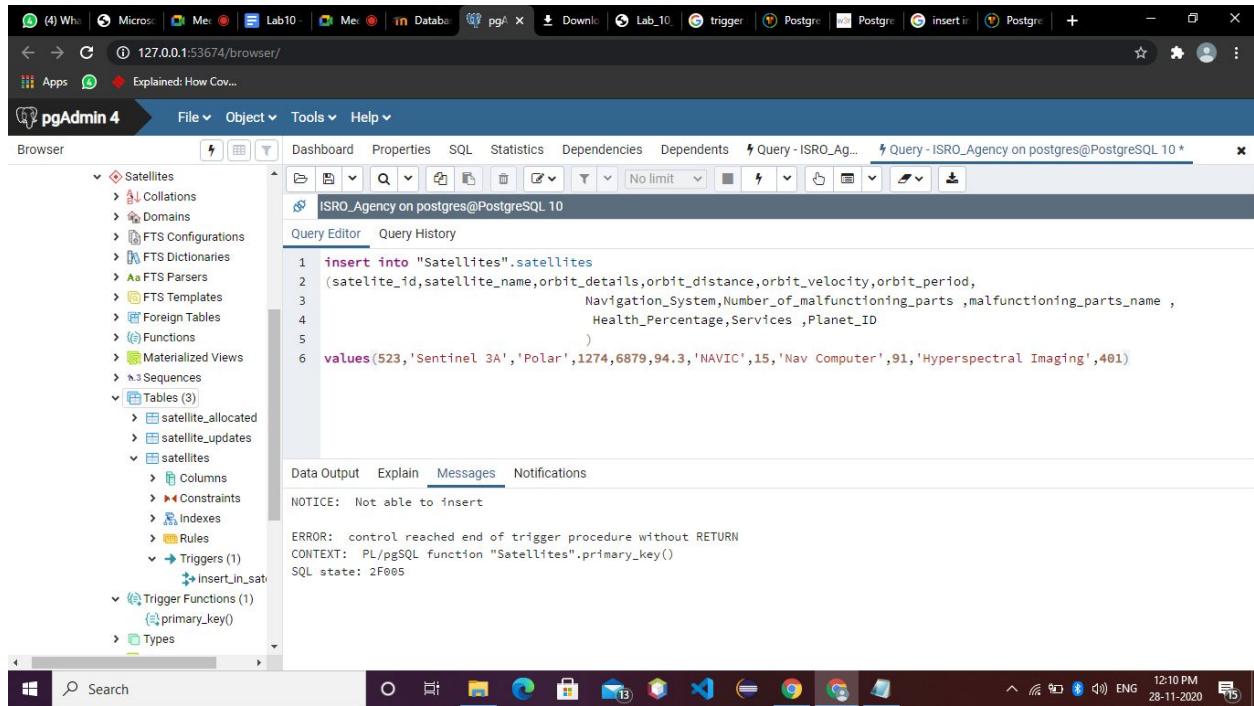
```

```
END;
$BODY$
LANGUAGE plpgsql;
```

```
create trigger insert_in_satellite
before insert on "Satellites".satellites
for each row
execute procedure "Satellites".primary_key();
```

```
insert into "Satellites".satellites
(satelite_id,satellite_name,orbit_details,orbit_distance,orbit_velocity,orbit_per
iod,
```

```
Navigation_System,Number_of_malfunctioning_parts
,malfunctioning_parts_name ,
Health_Percentage,Services
,Planet_ID
)
values(523,'Sentinel 3A','Polar',1274,6879,94.3,'NAVIC',15,'Nav
Computer',91,'Hyperspectral Imaging',401)
```



29)

CREATE OR REPLACE FUNCTION "Satellites".foriegn_key()

RETURNS trigger AS

\$BODY\$

BEGIN

IF (SELECT COUNT(*) FROM "Satellites".satellites WHERE "Satellites".satellites.planet_id= NEW.planet_id)= 0 THEN

 raise notice 'Foriegn key planet_id value not exist in planets';

else

 return new;

end if;

END;

\$BODY\$

LANGUAGE plpgsql;

create trigger foriegn_id_checking
before insert on "Satellites".satellites
for each row

```
execute procedure "Satellites".foriegn_key()
```

```
insert into "Satellites".satellites  
(satelite_id,satellite_name,orbit_details,orbit_distance,orbit_velocity,orbit_per  
iod,
```

```
Navigation_System,Number_of_malfunctioning_parts  
,malfunctioning_parts_name ,  
Health_Percentage,Services  
,Planet_ID  
)  
values(600,'Sentinel 3A','Polar',1274,6879,94.3,'NAVIC',15,'Nav  
Computer',91,'Hyperspectral Imaging',410)
```

```
Space Agency /postgres@PostgreSQL 10
Query Editor Query History
Scratch Pad x

20 execute procedure "Satellites".foriegn_key()
21
22 |
23 select * from "Satellites".satellites
24
25 insert into "Satellites".satellites
26 (satelite_id,satellite_name,orbit_details,orbit_distance,orbit_velocity,orbit_period,
27 Navigation_System,Number_of_malfunctioning_parts ,malfunctionin
28 Health_Percentage,Services ,Planet_ID
29 )
30 values(600,'Sentinel 3A','Polar',1274,6879,94.3,'NAVIC',15,'Nav Computer',91,'Hyperspectral Imag
31

Data Output Explain Messages Notifications
NOTICE: Foreign key planet_id value not exist in planets
ERROR: control reached end of trigger procedure without RETURN
CONTEXT: PL/pgSQL function "Satellites".foriegn_key()
SQL state: 2F005

Activate Windows
Go to Settings to activate Windows.

Windows taskbar icons
```

30)select star_id from "PABS".stars where classification='K' and shape='Decagram'

```

1 select star_id from "PABS".stars where classification='K' and shape='Decagram'

```

star_id	[PK] integer
1	

Activate Windows

Successfully run. Total query runtime: 356 msec. 1 rows affected.

31)

```

select satellite_name,health_percentage from "Satellites".satellites
join "PABS".planets on
"PABS".planets.planet_id = "Satellites".satellites.planet_id
where "PABS".planets.name = 'Mars'

```

```

1 select satellite_name,health_percentage from "Satellites".satellites
2 join "PABS".planets on
3 "PABS".planets.planet_id = "Satellites".satellites.planet_id
4 where "PABS".planets.name = 'Mars'

```

satellite_name	health_percentage
Azerospace Zintelsat-38	81
Lucky-7	69
DirectTV-8	87
SB-WASS 3-8 (Space B...	74
EKS-3 (Integrated Spac...	75
Cubesat XIHV (Oscar 5...	78
Yinhe 1 (Galaxy 1, GS-S...	95
SPIRALE-A (Système P...	87

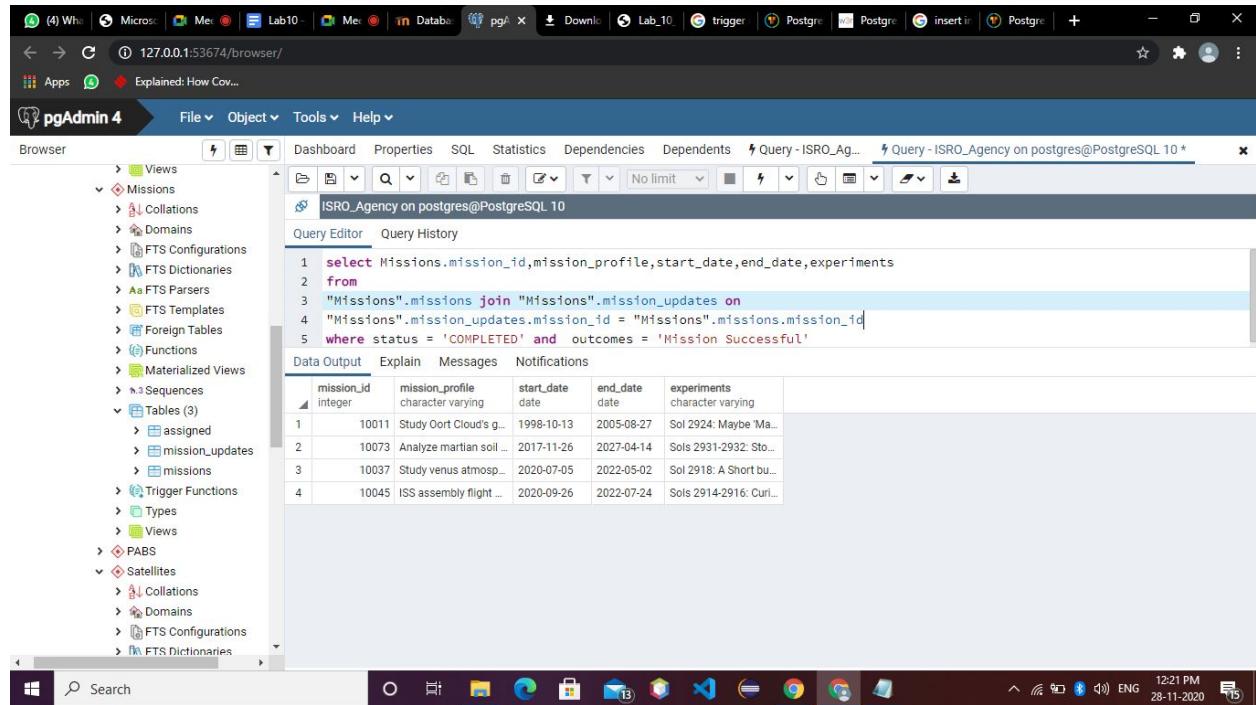
32)

```

select Missions.mission_id,mission_profile,start_date,end_date,experiments
from
"Missions".missions join "Missions".mission_updates on

```

```
"Missions".mission_updates.mission_id = "Missions".missions.mission_id  
where status = 'COMPLETED' and outcomes = 'Mission Successful'
```



The screenshot shows the pgAdmin 4 interface. On the left is a tree view of database objects under the schema 'ISRO_Agency'. The central area is a 'Query Editor' window with the following SQL query:

```
1 select Missions.mission_id,mission_profile,start_date,end_date,experiments
2 from
3 "Missions".missions join "Missions".mission_updates on
4 "Missions".mission_updates.mission_id = "Missions".missions.mission_id
5 where status = 'COMPLETED' and outcomes = 'Mission Successful'
```

Below the query editor is a 'Data Output' table showing the results of the query:

mission_id	mission_profile	start_date	end_date	experiments
1	Study Oort Cloud's g...	1998-10-13	2005-08-27	Sol 2924: Maybe 'Ma...
2	Analyze martian soil ...	2017-11-26	2027-04-14	Sols 2931-2932: Sto...
3	Study venus atmosph...	2020-07-05	2022-05-02	Sol 2918: A Short bu...
4	ISS assembly flight ...	2020-09-26	2022-07-24	Sols 2914-2916: Curi...

33)

```
select "PABS".blackholes.name,"Agency and  
personnel".isro_personnel.name from "PABS".blackholes join  
"PABS".blackhole_studied on  
"PABS".blackholes.blackhole_id="PABS".blackhole_studied.blackhole_id  
join "Agency and personnel".isro_personnel  
on "Agency and  
personnel".isro_personnel.id="PABS".blackhole_studied.personnel_id
```

The screenshot shows the pgAdmin 4 interface. On the left, the Browser pane displays a tree view of database objects under the schema "Space Agency". The "Tables" node is expanded, showing tables like "asteroid_studied", "blackhole_studied", "blackholes", "planet_studied", and "planets". The "Columns" node under "planets" is also visible. The main area is the Query Editor, which contains the following SQL query:

```

1 select "PABS".blackholes.name,"Agency and personnel".isro_personnel.name from "PABS".blackholes
2 "PABS".blackholes.blackhole_id="PABS".blackhole_studied.blackhole_id
3 join "Agency and personnel".isro_personnel
4 on "Agency and personnel".isro_personnel.id="PABS".blackhole_studied.personnel_id

```

Below the query, the Data Output tab shows a table with 10 rows of data:

	name	name
1	Messier 81	V Koteswara
2	1E1740-72942	Prem Shanker Goel
3	NGC 3377	Anuradha TK
4	NGC-3608	MS Pannirselvam
5	Messier 110	Ritu Kardhal
6	NGC-1023	Anuradha TK
7	RX J1131	BS Kiran
8	Mrk 501	K Radhakrishnan
9	NGC 3245	SK Shivakumar
10	3C 75	Tulsi Shah

A green message bar at the bottom right indicates: "Successfully run. Total query runtime: 242 msec. 100 rows affected."

34)

select * from "Agency and personnel".isro_personnel
order by salary

The screenshot shows the pgAdmin 4 interface. The Query Editor contains the following SQL query:

```

1 select * from "Agency and personnel".isro_personnel
2 order by salary desc
3

```

Below the query, the Data Output tab shows a table with 11 rows of data:

	id	name	designation	age	salary	start_date
1	129	T. S. Prahlad	Director-DOS	67	400000	2003-12-03
2	166	Thekkethil Kocha...	Assistant Engine...	27	400000	2001-09-29
3	134	Ritu Kirdhal	Secretary	23	400000	2001-08-01
4	162	SK Shivakumar	Library Officer-E...	26	400000	2000-12-28
5	188	Jitendra Nath Go...	Director-DOS	41	400000	2010-07-13
6	155	SK Shivakumar	Member (Financ...	34	400000	2004-01-15
7	187	Prem Shanker Go...	Joint Director (O...	40	400000	2012-10-14
8	115	B. Codanayaguy	Joint Director (O...	55	400000	2013-04-08
9	199	SK Shivakumar	Library Officer-E...	45	400000	2011-12-30
10	151	V Koteswara	Director-DOS	30	400000	2009-08-09
11	108	Subbiah Arunan	Junior Engineer, ...	21	400000	2003-12-18

35)

select name from "Agency and personnel".isro_personnel
join "PABS".star_studied on

```

"PBS".star_studied.personnel_id = "Agency and
personnel".isro_personnel.id
join "PBS".stars on
"PBS".stars.star_id = "PBS".star_studied.star_id
where "PBS".stars.classification = 'M'

```

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database schema with tables like asteroid_studied, asteroids, blackhole_studied, blackholes, planet_studied, planets, star_studied, and stars selected. The main area contains a query editor with the provided SQL code and a data output table showing names.

```

2 join "PBS".star_studied on
3 "PBS".star_studied.personnel_id = "Agency and personnel".isro_personnel.id
4 join "PBS".stars on
5 "PBS".stars.star_id = "PBS".star_studied.star_id
6 where "PBS".stars.classification = 'M'

```

Recorded time	Event	Process ID	Payload
No data found			

	name	
10	M. Annamalai (sci...	
17	BS Kiran	...
18	V Koteswara	...
19	SK Shivakumar	...
20	B Jayakumar	...
21	B Jayakumar	...
22	Moumita Dutta	...
23	Mylswamy Annadu...	

36)

```

select satellite_id from "Agency and personnel".space_agencies where
agencies_name='Australian Space Agency'

```

The screenshot shows the pgAdmin 4 interface with a query editor containing the provided SQL code and a data output table showing satellite_id.

```

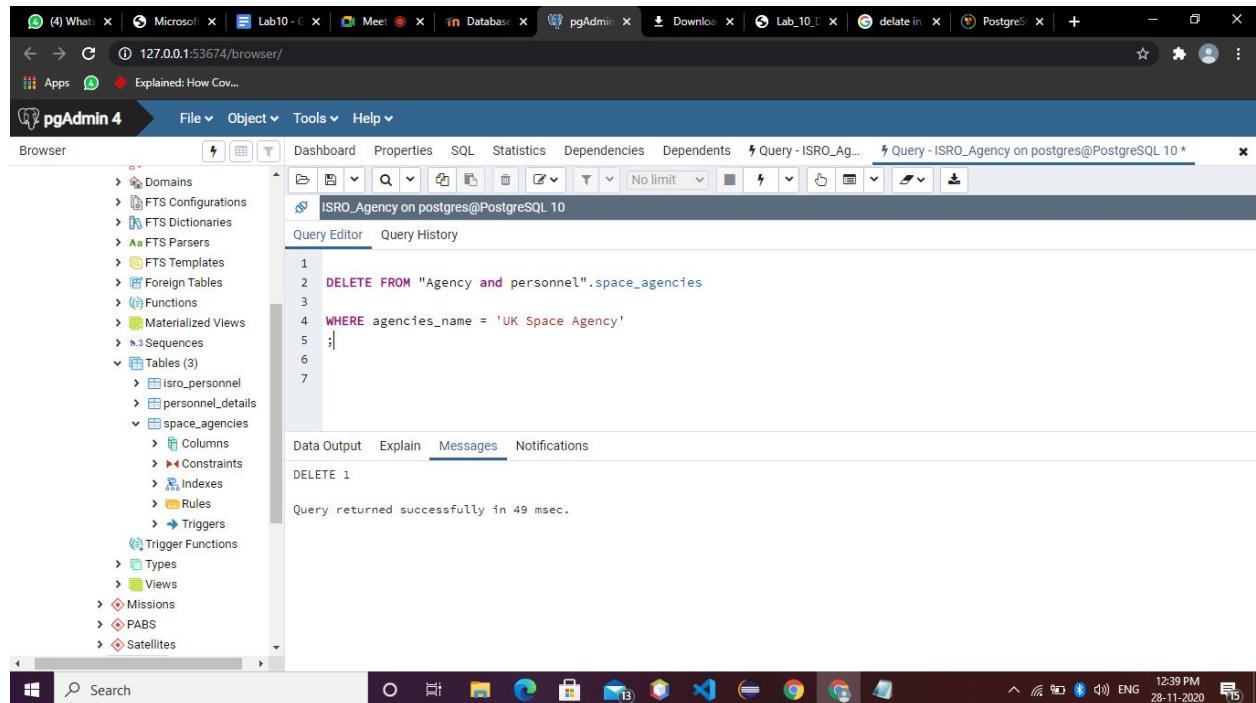
1 select satellite_id from "Agency and personnel".space_agencies where agencies_name='Australian Space Agency'

```

	satellite_id	
1	541	

37)

```
DELETE FROM "Agency and personnel".space_agencies  
WHERE agencies_name = 'UK Space Agency';
```



The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of database objects under the 'ISRO_Agency' schema, including Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Sequences, and three Tables: 'isro_personnel', 'personnel_details', and 'space_agencies'. The 'space_agencies' table is currently selected. The main window contains a 'Query Editor' tab with the following SQL code:

```
1  
2 DELETE FROM "Agency and personnel".space_agencies  
3  
4 WHERE agencies_name = 'UK Space Agency'  
5 ;  
6  
7
```

Below the query editor, the 'Messages' tab is active, showing the message: 'DELETE 1'. At the bottom of the pgAdmin window, the status bar indicates 'Query returned successfully in 49 msec.' The system tray at the bottom right shows the date and time as '28-11-2020 12:39 PM'.

38)

```
delete  
from  
"Missions".mission_updates  
where  
outcomes = 'Mission Cancelled by agency'
```

The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of database objects under the 'ISRO_Agency' schema, including FTS Configurations, Functions, Tables (with 'mission_updates' selected), and Views. The main window has a toolbar at the top with various icons. Below the toolbar is a search bar and a breadcrumb trail showing 'ISRO_Agency on postgres@PostgreSQL 10'. The central area is divided into 'Query Editor' and 'Query History'. The 'Query Editor' contains the following SQL code:

```
1 delete
2 from
3 "Missions".mission_updates
4 where
5 outcomes = 'Mission Cancelled by agency'
```

Below the code, the 'Messages' tab is active, showing the output: 'DELETE 26'. A green message box at the bottom right indicates 'Query returned successfully in 53 msec.' The system tray at the bottom right shows the date and time as 28-11-2020 12:41 PM.

39)

update

"Agency and personnel".isro_personnel

set salary = salary+10000

where

id in (select mission_id from "Missions".missions where status = 'COMPLETED')

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database schema with several schemas and tables. The main area shows a query editor with the following SQL code:

```
1 update
2 "Agency and personnel".isro_personnel
3 set salary = salary+10000
4 where
5 id in (select mission_id from "Missions".missions where status = 'COMPLETED')
```

The query was executed successfully, returning 0 rows affected.

40)

select name from "PABS".planets where mean_tempreature<0 order by mass desc

The screenshot shows the pgAdmin 4 interface. The query editor contains the following SQL code:

```
1 select name from "PABS".planets where mean_tempreature<0 order by mass desc
```

The results are displayed in a table:

name
1 Jupiter
2 Saturn
3 Neptune
4 Uranus
5 Mars
6 Pluto
7 Moon

A message at the bottom indicates the query was successfully run with a runtime of 185 msec and 7 rows affected.

