Title: Mars Exploration
Name: Gurpreet Natt
Student ID: 922883894
GitHub Username: gpreet2

Checkpoint #	Date Submitted
Checkpoint 1	09/18/2023
Checkpoint 2	10/04/2023
Checkpoint 3	10/25/2023

Table of Contents

Project Description

Motivations:

The recent surge in interest and investments in Mars exploration has led to an exponential increase in data generated from various Mars missions. However, there's a conspicuous gap in the centralization and optimal utilization of this vast amount of data. Researchers, scientists, and space enthusiasts often find themselves navigating through fragmented databases, incompatible systems, or outdated data formats, leading to inefficiencies and potential loss of insights. The Mars Exploration Database System (MEDS) is designed to address these issues by creating a cohesive, advanced, and accessible repository for all Mars-related data.

High Level Non-Technical Descriptions:

At its core, MEDS is a comprehensive database system that will serve as a single source of truth for all Mars exploration-related data. It will integrate information from various Mars missions, studies, and research projects, enabling users to easily search, retrieve, and analyze pertinent information. The system will be user-friendly, designed with a modern interface, and will cater to both technical and non-technical users, ensuring ease of access and comprehension.

Unique Features:

Integrated Data Visualizations: MEDS is designed with an embedded visualization toolkit, allowing users to generate a wide variety of visual representations, from basic graphs to intricate 3D depictions of Martian topography. This feature is not just about aesthetics or user interface. It’s about enabling deeper insights into Mars-related data without the need to switch between platforms. By integrating visualization tools directly into the database system, we ensure that users can move seamlessly from data collection to analysis, enhancing efficiency and promoting a holistic understanding of the data at hand.

Real-Time Data Updates:

The dynamism of MEDS is exemplified by its real-time data update feature. A cutting-edge API

gateway is woven into the system, connecting it directly to satellite communication channels. This ensures that the data relayed from Mars missions, whether it's a rover's latest finding or a satellite's recent imagery, gets reflected in the system instantaneously. With this feature, researchers and scientists can be confident that they are always working with the freshest data, a necessity in the rapidly evolving field of Mars exploration.

Advanced Search Capabilities:

In MEDS, we're reimagining search functionalities. Instead of relying on keyword-based searches alone, our system is augmented with natural language processing capabilities. This means users can pose complex questions or commands in everyday language, and the system will interpret and execute them. Imagine asking MEDS, "Show me temperature data from the Gale Crater in the last Martian year," and receiving precise, relevant results. This level of intuitiveness can radically simplify data retrieval processes.

Adaptive Learning:

MEDS is not just a passive repository of information; it’s a learning system. Leveraging machine learning algorithms, MEDS monitors and learns from user interactions. If a researcher repeatedly explores data from a particular Martian region or concerning a specific element, MEDS might preemptively present related new data or suggest pertinent studies. Such adaptability ensures a personalized user experience, making data navigation more intuitive and responsive.

Use cases:

Research Analysis:

Consider a team of scientists at a renowned space research institution. Their primary focus is uncovering the mysteries of Mars' southern pole, specifically regarding water presence. With MEDS, they can effortlessly extract historical sensor data, charting soil moisture changes across Martian years. Further, they can juxtapose this data against other variables such as temperature or seismic readings, fostering a more holistic understanding and potentially groundbreaking discoveries.

Mission Planning:

As space agencies worldwide contemplate future rover missions, the planning and execution of these missions become paramount. Imagine a team gearing up for a new Mars venture. With MEDS, they can scout prospective landing zones, filtering regions based on specific criteria like terrain flatness, historical weather patterns, or proximity to previously explored points of interest. This informed approach, powered by MEDS, can significantly enhance mission success probabilities.

Educational Purposes:

MEDS isn't exclusively for the scientific community. Picture a university setting where a professor, entrusted with enlightening young minds about planetary science, turns to MEDS. Through interactive assignments, students might be asked to discern regions with notable temperature fluctuations or trace the journey of a specific rover. Such hands-on engagements can make learning immersive, bridging the gap between theoretical knowledge and practical application.

Real-Time Monitoring:

The true test of any Mars mission is its live execution. Envision a critical phase where a rover is scheduled to traverse a particularly challenging Martian terrain. The team at mission control, relying on MEDS, can access real-time data feeds, watching every move of the rover. In the face of unforeseen events, like a sudden Martian sandstorm, the team can instantly reference historical data to predict storm behavior, ensuring timely interventions and safeguarding mission objectives.

Public Engagement:

MEDS also serves as a bridge to the curious public. Space enthusiasts, equipped with limited access credentials, can dive into the system. They can browse user-friendly dashboards, partake in community-led data challenges, or even attend virtual talks by leading scientists unveiling the latest Martian discoveries. This democratization of space data can foster a more informed and engaged global community.

Existing Software Tools

NASA’s Eyes:

As a leading visualization software, NASA's Eyes brings the cosmos to our screens. But imagine the transformation when it's integrated with MEDS. The real-time and expansive Mars data from MEDS can overlay live simulations on NASA's Eyes. Users, whether casual space enthusiasts or seasoned researchers, can witness live Martian events, tracking rover pathways or observing real-time weather shifts. The blend of MEDS' rich data repository with NASA's Eyes' visual capabilities can redefine space exploration storytelling.

StarWalk:

StarWalk, a cherished tool for stargazers, provides insights into celestial entities. Now, imagine the leap in its offerings when it taps into MEDS. As users zoom into Mars, they aren’t just greeted with static images but a dynamic, data-rich, and interactive Martian experience. From current atmospheric readings to detailed topographical outlines, StarWalk can transform from a stargazing app to an insightful Mars exploration companion, all thanks to its potential integration with MEDS.

Functional Database Requirements:

1. User
   1.1. A user shall create only one MEDS account.
   1.2. A user shall be able to save multiple data searches for future reference.
   1.3. A user shall have at least one role (e.g., Scientist, Mission Planner, Educator, etc.).
   1.4. A user can belong to multiple research teams.
   1.5. A user shall be able to set and reset their password.

2. Account
   2.1. An account shall be created by only one user.
   2.2. An account can have multiple saved visualizations.
   2.3. An account shall have a unique email identifier.
   2.4. An account can follow multiple missions for real-time updates.

3. Role
   3.1. A role shall be linked to many users.
   3.2. Each role shall have specific permissions defined.
   3.3. A role can be assigned specific datasets it can access.

4. Mission
   4.1. A mission can be monitored by multiple users.
   4.2. Each mission shall have a designated mission head.
   4.3. A mission can produce multiple sets of data (e.g., imagery, sensor data).
   4.4. A mission shall have a unique mission identifier.

5. Research Team
   5.1. A research team shall consist of multiple users.
   5.2. A research team can work on multiple projects.
   5.3. A research team can have multiple collaborative workspaces within MEDS.

6. Dataset
   6.1. A dataset can be accessed by multiple roles.
   6.2. Each dataset shall have one source (e.g., specific rover, satellite).
   6.3. A dataset can be part of an aggregation with other datasets.
   6.4. A dataset may reference another dataset (recursive relationship).

7. Visualization
   7.1. A visualization can integrate data from multiple datasets.
   7.2. Each visualization shall have one creator.
   7.3. A visualization can be shared with multiple users or teams.

8. Sensor
   8.1. A sensor shall be linked to only one rover or satellite.
   8.2. A sensor can produce multiple types of data (e.g., temperature, imagery).
   8.3. A sensor shall have a unique identifier.

9. Rover
   9.1. A rover can be equipped with multiple sensors.
   9.2. Each rover shall belong to one mission.
   9.3. A rover can send data to multiple datasets.

10. Satellite
   10.1. A satellite can monitor multiple zones on Mars.

10.2. A satellite can be equipped with multiple sensors.
10.3. Each satellite shall belong to one mission.

11. Discussion Thread
11.1. A discussion thread can have multiple participants (users).
11.2. Each discussion thread shall belong to one workspace.
11.3. A discussion thread may reference other threads (recursive relationship).

12. Notification System
12.1. A user shall be able to set preferences for receiving notifications.
12.2. A notification can inform multiple users based on their roles or interests.
12.3. Each notification shall have a unique identifier.
12.4. A user shall be able to clear or archive notifications.

13. Martian Terrain
13.1. A terrain shall have specific data points like mineral compositions.
13.2. Each terrain type can be associated with multiple mission landing sites.
13.3. A terrain can be explored by multiple rovers over time.

14. Research Document
14.1. A research document can cite multiple datasets.
14.2. Each document shall be linked to a user or a research team as its author.
14.3. A document can be peer-reviewed by multiple users.
14.4. Each document shall have a unique digital object identifier (DOI).

15. Training Module
15.1. A module can be accessed by multiple roles for educational purposes.
15.2. Each training module shall have a feedback section.
15.3. A training module can have prerequisites, with other modules needing to be completed first.

16. API Access
16.1. MEDS shall provide API keys for third-party tool integration.
16.2. Each API key can be linked to specific datasets or functionalities.
16.3. An API key can have varied levels of access permissions.

17. Saved Searches
17.1. Users must be able to save a search query for future reference.
17.2. Users shall be able to name a saved search for easier identification.
17.3. The system shall allow users to delete or modify saved searches.

18. Mission Sensors
18.1. Mission planners must be able to associate sensors with a specific mission.
18.2 The system shall list all available sensors during mission planning.
18.3. Users should be able to view the status (active/inactive) of sensors associated with a mission.
19. Discussion Replies
19.1 Users must be able to reply to an existing discussion thread.
19.2 Each reply should reference the user who made the reply and the discussion thread it belongs to.
19.3.The system shall allow users to edit or delete their replies

20. Document Citations
20.1. Users must be able to cite documents within their research papers or reports.
20.2. Each citation should reference the original document and the user or research team citing it.
20.3 The system shall provide a standardized citation format for users to follow.

21. Module Feedback
21.1. Users must be able to leave feedback on completed training modules.
21.2 Each feedback entry should reference the user giving the feedback and the module it pertains to.
21.3. Module creators or administrators must receive notifications when new feedback is submitted.

22. Machine
22.1 Each machine shall have a unique identifier.
22.2. Machines can be a part of one or more spacecraft.
22.3. Maintenance and status information of machines shall be tracked.
22.4. Machines can be associated with specific components.

23. Payment
23.1. Each payment shall have a unique transaction identifier.
23.2. Payments are associated with a user's account.
23.3. Each payment shall include details such as the amount, date, and payment method.
23.4. Payments shall be securely processed and stored.

24. Mailing List:
24.1. Each mailing list subscription shall have a unique identifier.
24.2. Users can subscribe to and unsubscribe from the mailing list.

24.3. Mailing lists shall be used to send updates and information to subscribers.
24.4. Mailing list subscriptions shall include the user's email address and subscription preferences.

25. Weather
25.1. Each weather record shall have a unique identifier.
25.2. Weather records are associated with Mars, providing atmospheric and surface conditions.
25.3. Weather data shall be time stamped to indicate the exact time of the recorded conditions.
25.4. Weather records shall include various meteorological parameters such as temperature, wind speed, atmospheric pressure, and humidity.

Non-functional Requirements:

1. Performance
1.1. MEDS shall support concurrent access for up to 5,000 users.
1.2. Response time for common queries (e.g., mission data retrieval) shall not exceed 2 seconds under normal load.
1.3. The database system shall guarantee 99.9% uptime.

2. Storage
2.1. Each dataset related to Mars terrain should not exceed 50 MB in size.
2.2. MEDS shall automatically compress datasets older than one year to optimize storage.
2.3. The system shall support incremental storage upgrades without significant downtime.

3. Security
3.1. All data transmissions to and from MEDS shall be encrypted using modern encryption standards.
3.2. MEDS shall implement role-based access control to ensure appropriate data access levels.
3.3. Failed login attempts shall be logged and users shall be temporarily locked out after five failed attempts.

4. Scalability
4.1. MEDS shall be capable of scaling horizontally to accommodate future Mars missions and increased datasets.
4.2. The system shall support seamless integration with future data analytics tools.

5. Reliability

5.1. MEDS shall undergo regular maintenance checks every month without disrupting the main services.

5.2. The system shall automatically reroute users to backup servers in the event of a main server failure.

6. Usability

6.1. MEDS shall provide an intuitive user interface suitable for both technical and non-technical users.

6.2. The system shall offer multilingual support to cater to an international user base.

6.3. On-demand training modules shall be available for new users.

7. Compatibility

7.1. MEDS shall support API integrations to allow third-party tools to fetch and push data.

7.2. The system shall be accessible through major web browsers, including Chrome, Firefox, and Safari.

8. Maintainability

8.1. MEDS shall be modular in design to allow for easy updates and patches.

8.2. Regular updates shall be deployed at least quarterly, addressing both functionality and security enhancements.

9. Backup and Recovery

9.1. MEDS shall maintain a mirrored backup in a geographically separate location to ensure data durability.

9.2. In the event of a system failure, MEDS shall guarantee data recovery within 24 hours.

10. Data Integrity

10.1. MEDS shall enforce referential integrity to avoid orphan records.

10.2. Data validation checks shall be in place to ensure the accuracy and consistency of data entered.

Entity Descriptions

1. User (Strong)
   - user_id: key, numeric
   - username: composite, alphanumeric
   - email: composite, alphanumeric
   - date_joined: single-value, timestamp
   - last_login: single-value, timestamp

2. Mission (Strong)
   - mission_id: key, numeric
   - start_date: single-value, timestamp
   - end_date: single-value, timestamp
   - objective: composite, alphanumeric
   - status: composite, alphanumeric

3. Rover (Strong)
   - rover_id: key, numeric
   - name: composite, alphanumeric
   - launch_date: single-value, timestamp
   - arrival_date: single-value, timestamp
   - status: composite, alphanumeric

4. Satellite (Strong)
   - satellite_id: key, numeric
   - name: composite, alphanumeric
   - launch_date: single-value, timestamp
   - operational_date: single-value, timestamp

5. Dataset (Strong)
   - dataset_id: key, numeric
   - source_id: composite, numeric (can relate to Rover or Satellite)
   - date_collected: single-value, timestamp

6. Visualization (Strong)
   - visualization_id: key, numeric
   - created_by: composite, numeric (relating to User)
   - date_created: single-value, timestamp

7. Component (Strong)
   - component_id: key, numeric
   - type: alphanumeric
   - status: alphanumeric

- data_output: alphanumeric

8. Research Team (Strong)
   - team_id: key, numeric
   - team_name: composite, alphanumeric
   - lead_researcher: composite, numeric (relating to User)

9. Discussion Thread (Strong)
   - thread_id: key, numeric
   - title: composite, alphanumeric
   - created_by: composite, numeric (relating to User)
   - created_date: single-value, timestamp

10. Mars (Strong)
   - mars_id: key, numeric
   - type: alphanumeric
   - status: alphanumeric
   - data_output: alphanumeric

11. Research Document (Strong)
   - document_id: key, numeric
   - title: composite, alphanumeric
   - author_id: composite, numeric (relating to User)
   - publish_date: single-value, timestamp

12. Training Module (Strong)
   - module_id: key, numeric
   - title: composite, alphanumeric
   - length: composite, numeric
   - difficulty_level: composite, alphanumeric

13. Notification System (Strong)
   - notification_id: key, numeric
   - type: composite, alphanumeric
   - message: composite, alphanumeric
   - date_sent: single-value, timestamp

14. API Access (Strong)
   - apikey_id: key, numeric
   - access_level: composite, alphanumeric
   - issue_date: single-value, timestamp
   - expiry_date: single-value, timestamp

15. Role (Strong)

- role_id: key, numeric
- role_name: composite, alphanumeric
- permissions: multivalue, alphanumeric

16. Account (Strong)
- account_id: key, numeric
- user_id: composite, numeric (relating to User)
- created_date: single-value, timestamp
- last_activity_date: single-value, timestamp

17. Mailing List (strong)
    - list_id: key, numeric
    - name: alphanumeric
    - description: alphanumeric

18. Payment (strong)
    - payment_id: key, numeric
    - amount: numeric
    - date: timestamp
    - method: alphanumeric

19. Weather (strong)
    - weather_id: key, numeric
    - temperature: numeric
    - pressure: numeric
    - wind_speed: numeric

20. Machine (strong)
    - machine_id: key, numeric
    - type: alphanumeric
    - status: alphanumeric
    - location: alphanumeric

21. Saved Searches (Weak)
- search_id: key, numeric
- account_id: composite, numeric (relating to Account)
- search_query: composite, alphanumeric

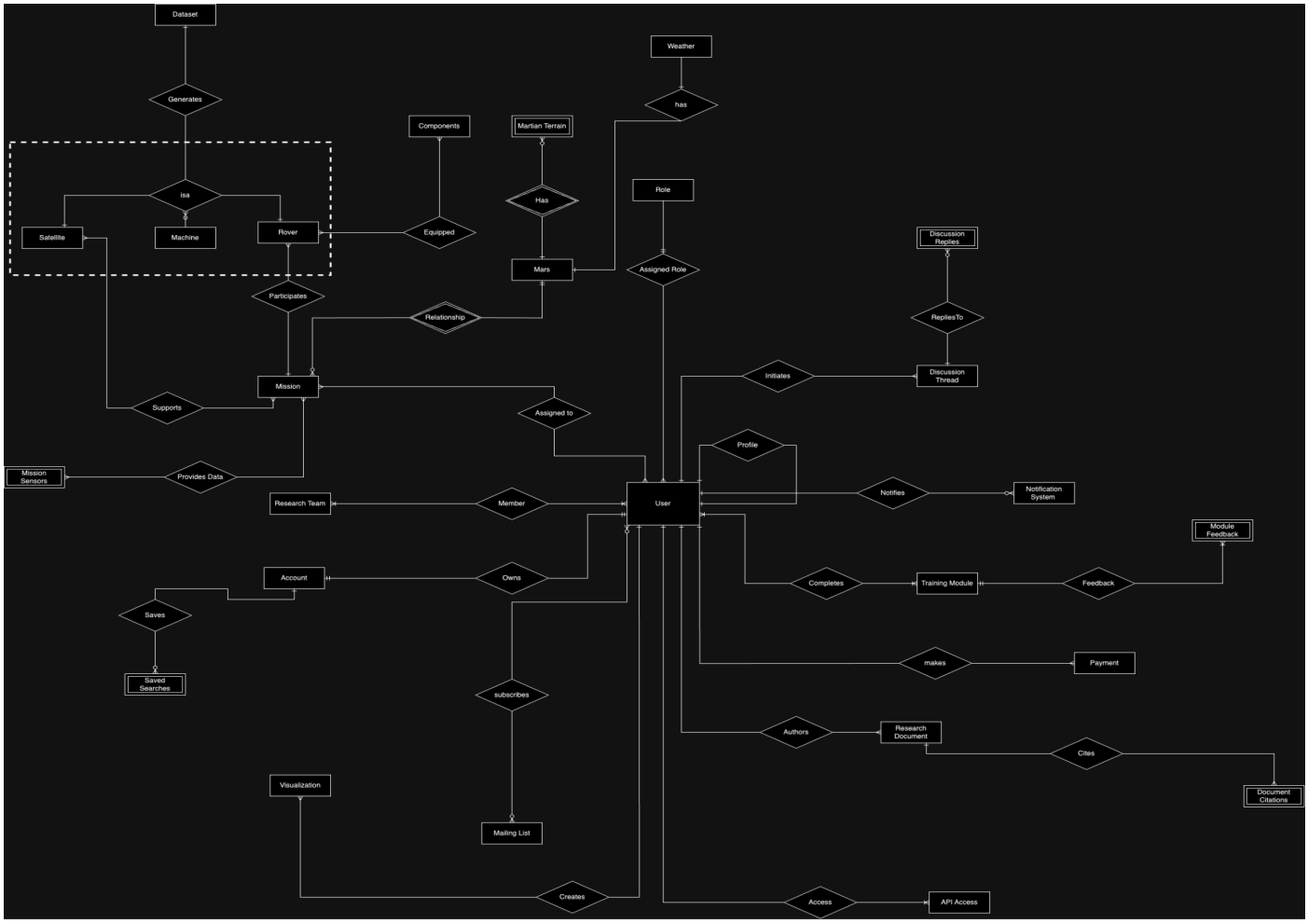
22. Mission Sensors (Weak)
- mission_sensor_id: key, numeric
- sensor_id: composite, numeric (relating to Sensor)
- rover_id/satellite_id: composite, numeric

23. Discussion Replies (Weak)

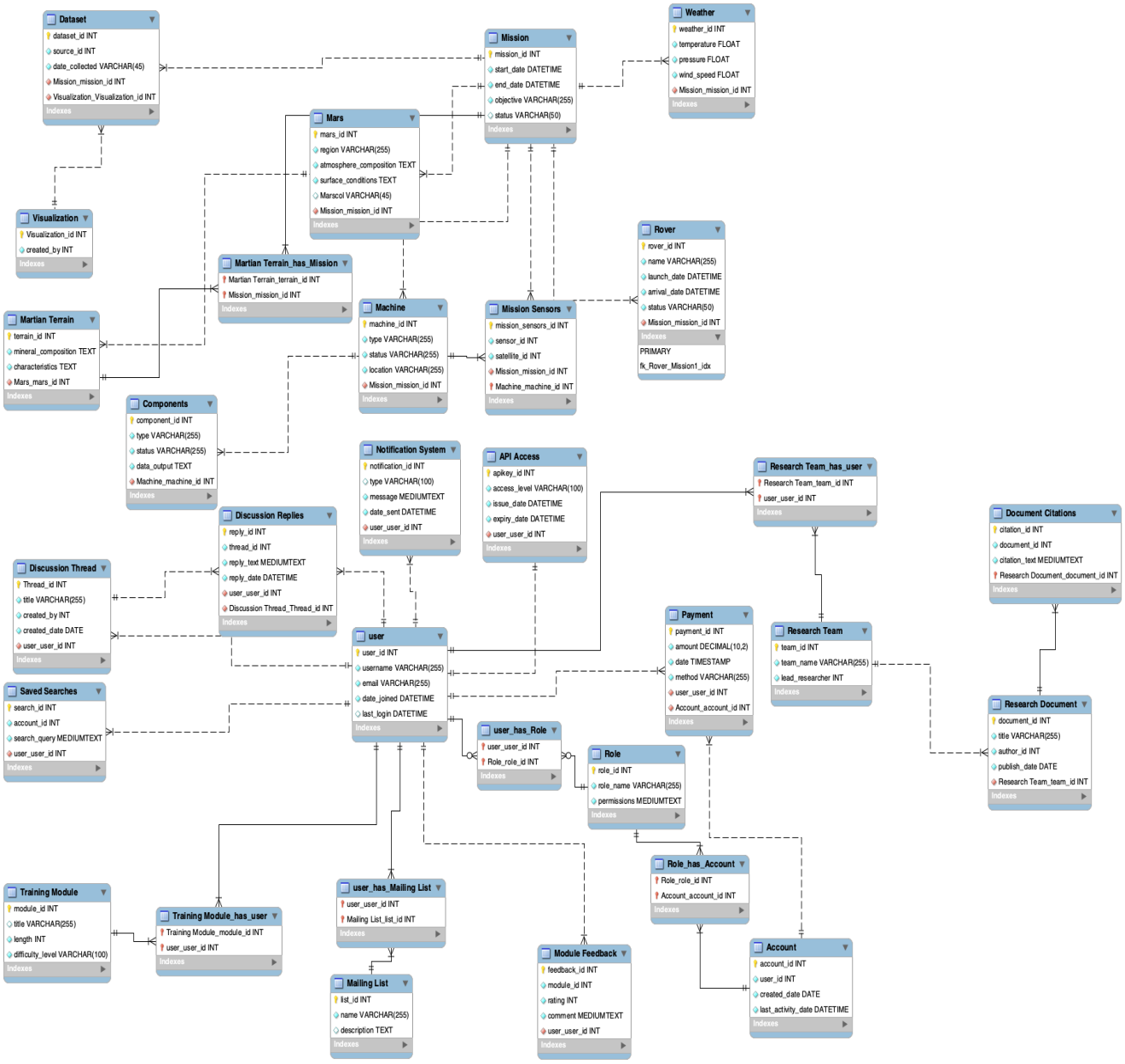
- reply_id: key, numeric
- thread_id: composite, numeric (relating to Discussion Thread)
- reply_text: composite, alphanumeric
- reply_date: single-value, timestamp

24. Document Citations (Weak)
- citation_id: key, numeric
- document_id: composite, numeric (relating to Research Document)
- citation_text: composite, alphanumeric

25. Module Feedback (Weak)
- feedback_id: key, numeric
- module_id: composite, numeric (relating to Training Module)
- rating: composite, numeric
- comment: composite, alphanumeric


Dataset
Generates
Weather
has
Components
Martian Terrain
Isa
Has
Role
Satellite
Machine
Rover
Equipped
Mars
Assigned Role
Discussion Replies
Participates
Relationship
RepliesTo
Mission
Initiates
Discussion Thread
Supports
Assigned to
Profile
Mission Sensors
Provides Data
Research Team
Member
User
Notifies
Notification System
Module Feedback
Account
Owns
Completes
Training Module
Feedback
Saves
makes
Payment
Saved Searches
subscribes
Authors
Research Document
Cites
Visualization
Document Citations
Mailing List
Creates
Access
API Access

Entity Establishment Relationship Diagram (EER)



Constraints Table

Table	FK	ON DELETE	ON UPDATE	COMMENT
Account	user_id	ON CASCADE	ON CASCADE	If a user is deleted, then the account associated with that user must be deleted as well.
Discussion Thread	created_by	SET NULL	ON CASCADE	If a user is deleted, the threads they created remain but the creator is set to NULL.
Research Team	lead_researcher	SET NULL	ON CASCADE	If a user is deleted, the lead researcher of the team is set to NULL.
Research Document	author_id	SET NULL	ON CASCADE	If a user is deleted, the author of the document is set to NULL.
Visualization	created_by	SET NULL	ON CASCADE	If a user is deleted, the visualizations they created remain but the creator is set to NULL.
API Access	user_id	ON CASCADE	ON CASCADE	If a user is deleted, their API access is also deleted.
Payment	user_id	ON CASCADE	ON CASCADE	If a user is deleted, their payment information is also deleted.

Saved Searches	account_id	SET NULL	ON CASCADE	If an account is deleted, the saved searches remain but are disassociated from any account.
Mission Sensors	sensor_id	ON CASCADE	ON CASCADE	If a sensor is deleted, the mission sensor information related to it is also deleted.
Discussion Replies	thread_id	ON CASCADE	ON CASCADE	If a thread is deleted, the replies to that thread are also deleted.
Document Citations	document_id	ON CASCADE	ON CASCADE	If a document is deleted, the citations of that document are also deleted.
Module Feedback	module_id	ON CASCADE	ON CASCADE	If a training module is deleted, the feedback associated with it is also deleted.
Martian Terrain	mars_id	ON CASCADE	ON CASCADE	If Mars information is deleted, the specific terrains of Mars associated with it are also deleted.
Mission Sensors (Rover)	rover_id	ON CASCADE	ON CASCADE	If a rover is deleted, the mission sensors associated with it are also deleted.

Mission Sensors (Satellite)	satellite_id	ON CASCADE	ON CASCADE	If a satellite is deleted, the mission sensors associated with it are also deleted.
Dataset (Rover)	rover_id	ON CASCADE	ON CASCADE	If a rover is deleted, the datasets associated with it are also deleted.
Dataset (Satellite)	satellite_id	ON CASCADE	ON CASCADE	If a satellite is deleted, the datasets associated with it are also deleted.
Machine	mission_id	ON CASCADE	ON CASCADE	If a mission is deleted, the machines associated with it are also deleted.
Component	mission_id	ON CASCADE	ON CASCADE	If a mission is deleted, the components associated with it are also deleted.