

COSMO-PLAN

SPACE -EXPLORATION

MANAGEMENT SYSTEM

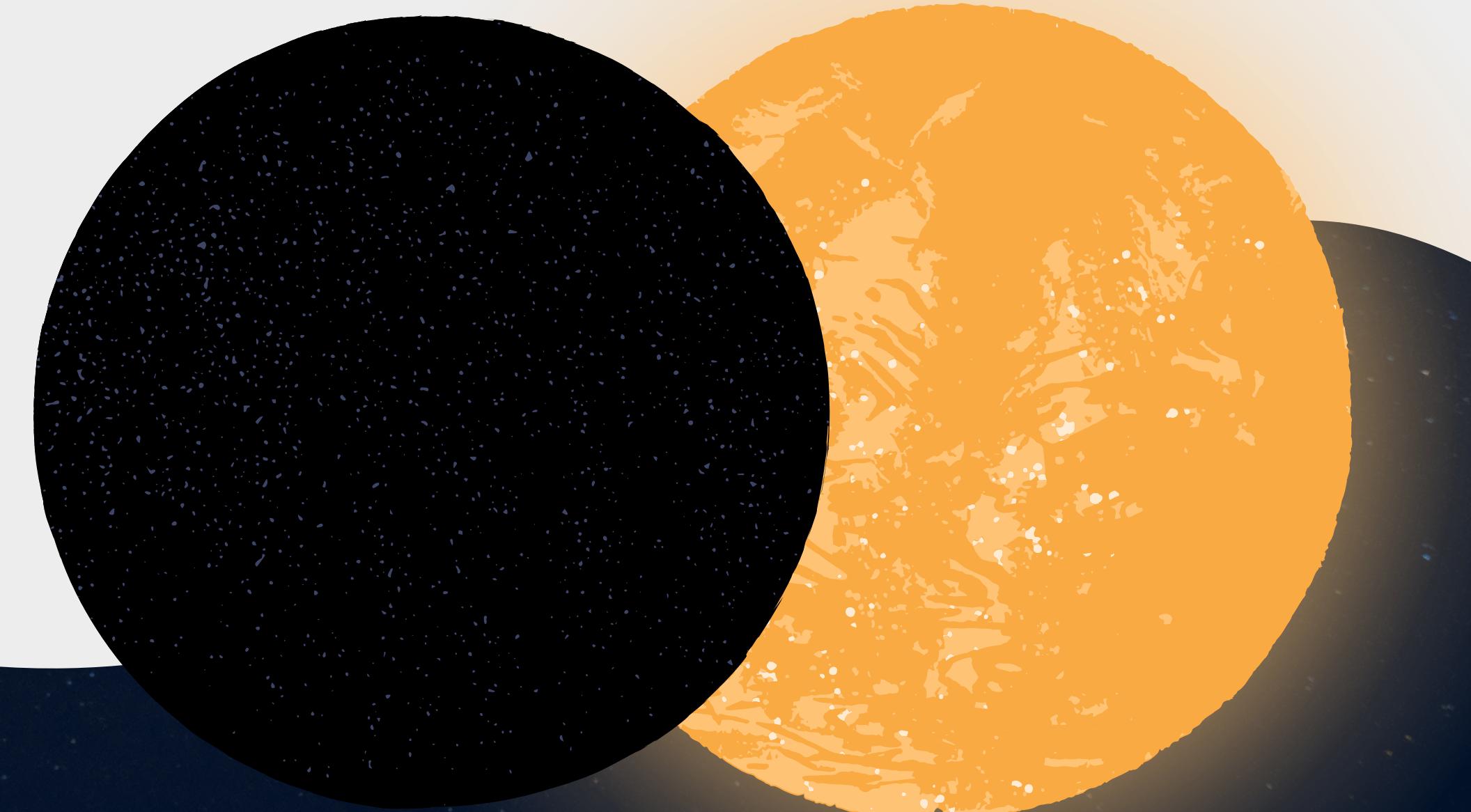
Dhwani Panchani

Shalini Dutta

Zeel Patel

Atharv Konge

Kiran Reddy



Mission Statement

Develop integrated database system for streamlined space exploration missions

Objectives:

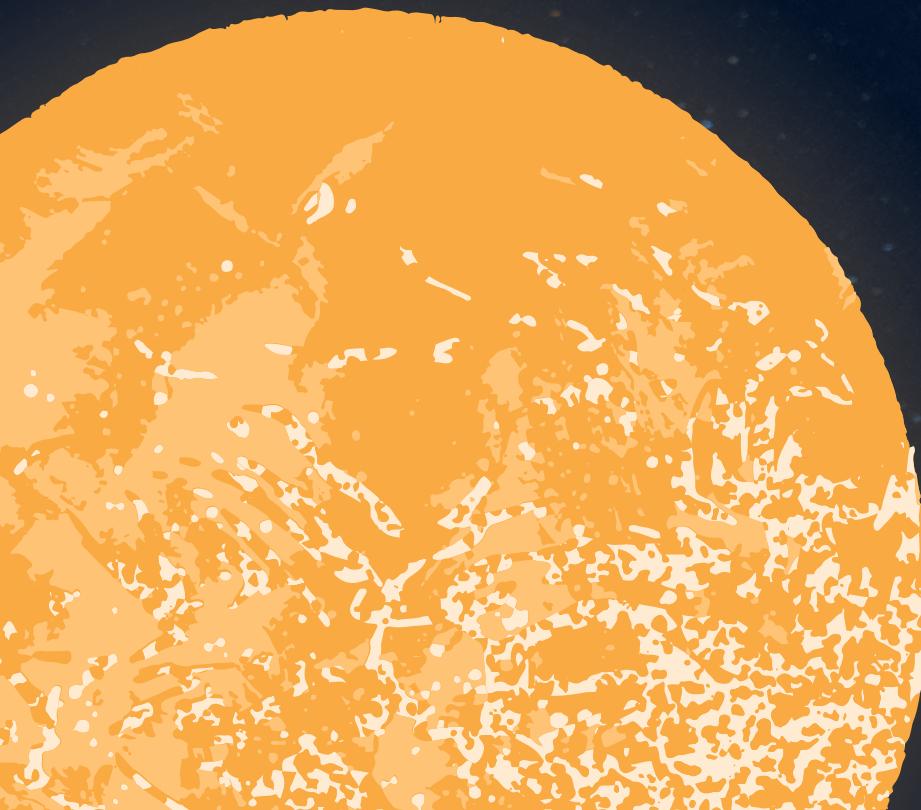
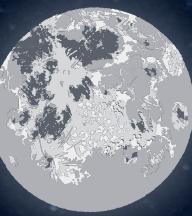
- Optimize planning processes and resource allocation
- Data Centralization and Accessibility
- Enhance communication channels
- Implement advanced risk management strategies
- Effective Mission Planning

Approach:

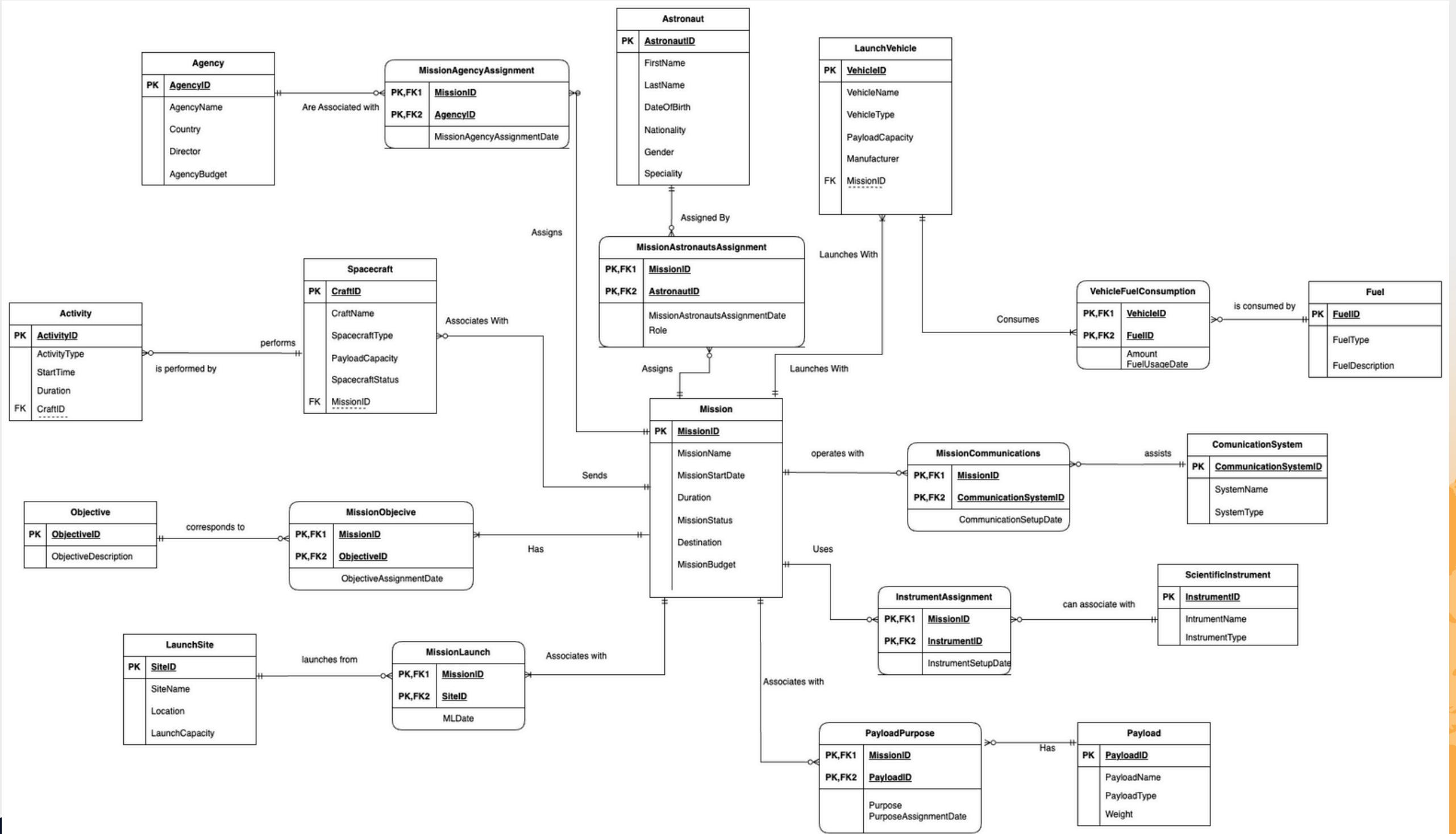
- Build robust centralized database
- Leverage SQL to add valuable functionalities
- Facilitate analytical visualizations

Database Design:

- Identify required entities
- Establish relationships
- Define essential attributes



ERD



Stored procedures (6)

```
1 ----- Stored Procedure 5 -----
2 -----Task - This procedure adjusts the budget of a given mission either by a fixed amount or a percentage.
3
4 CREATE PROCEDURE AdjustMissionBudget
5     @MissionID VARCHAR(255),
6     @AdjustmentAmount DECIMAL(19,4) = 0,
7     @IsPercentage BIT = 0,
8     @ResultMessage NVARCHAR(255) OUTPUT
9 AS
10 BEGIN
11     SET NOCOUNT ON;
12
13     DECLARE @CurrentBudget DECIMAL(19,4);
14
15     BEGIN TRY
16         SELECT @CurrentBudget = MissionBudget FROM Mission WHERE MissionID = @MissionID;
17
18         IF @IsPercentage = 1
19             BEGIN
20                 -- Adjust by a percentage
21                 UPDATE Mission
22                     SET MissionBudget = MissionBudget * (1 + @AdjustmentAmount / 100)
23                     WHERE MissionID = @MissionID;
24             END
25         ELSE
26             BEGIN
27                 -- Adjust by a fixed amount
28                 UPDATE Mission
29                     SET MissionBudget = MissionBudget + @AdjustmentAmount
30                     WHERE MissionID = @MissionID;
31             END
32
33         SET @ResultMessage = 'Mission budget adjusted successfully.';
34     END TRY
35     BEGIN CATCH
36         SET @ResultMessage = ERROR_MESSAGE();
37     END CATCH
38 END;
39
40 -----Testing-----
41 DECLARE @Msg NVARCHAR(255);
42 EXEC AdjustMissionBudget @MissionID = 'M101', @AdjustmentAmount = 5000000, @IsPercentage = 0, @ResultMessage = @Msg OUTPUT;
43 SELECT @Msg AS Message;
44
```

Results	Messages
	Message
1	Mission budget adjusted successfully.

```
1 -----Stored Procedure 6-----
2 -----Task - This procedure could check if the actual spend on a mission has exceeded the allocated budget.
3
4 CREATE PROCEDURE CheckBudgetAllocation
5     @MissionID VARCHAR(255),
6     @ActualSpend DECIMAL(19,4),
7     @ResultMessage NVARCHAR(255) OUTPUT
8 AS
9 BEGIN
10     SET NOCOUNT ON;
11
12     DECLARE @Budget DECIMAL(19,4);
13
14     BEGIN TRY
15         SELECT @Budget = MissionBudget FROM Mission WHERE MissionID = @MissionID;
16
17         IF @Budget < @ActualSpend
18             BEGIN
19                 SET @ResultMessage = 'Warning: Actual spend exceeds the budget.';
20             END
21         ELSE
22             BEGIN
```

Results	Messages
	Message
1	Budget allocation is within limits.

Views(3)

```

7 CREATE VIEW vw_MissionComprehensiveReport AS
8
9     SELECT
10        m.MissionID,
11        m.MissionName,
12        m.MissionStartDate,
13        DATEADD(DAY, m.Duration, m.MissionStartDate) AS 'EstimatedEndDate',
14        m.Duration,
15        m.MissionStatus,
16        m.Destination,
17        a.AgencyName AS 'ResponsibleAgency',
18        (SELECT COUNT(*) FROM MissionAstronautsAssignment WHERE MissionID = m.MissionID) AS 'NumberOfAstronauts',
19        (SELECT STRING_AGG(ObjectiveDescription, '; ') FROM Objective
20            INNER JOIN MissionObjective ON Objective.ObjectiveID = MissionObjective.ObjectiveID
21            WHERE MissionObjective.MissionID = m.MissionID) AS 'MissionObjectives',
22        ls.SiteName AS 'LaunchSite',
23        ls.Location AS 'LaunchSiteLocation',
24        (SELECT STRING_AGG(PayloadName, '; ') FROM Payload
25            INNER JOIN PayloadPurpose ON Payload.PayloadID = PayloadPurpose.PayloadID
26            WHERE PayloadPurpose.MissionID = m.MissionID) AS 'MissionPayloads',
27        s.CraftName AS 'AssignedSpacecraft',
28        lv.VehicleName AS 'AssignedLaunchVehicle'
29
30    FROM
31        Mission m
32    LEFT JOIN Agency a ON a.AgencyID = (SELECT TOP 1 AgencyID FROM MissionAgencyAssignment WHERE MissionID = m.MissionID ORDER BY MissionAgencyAssignmentDate)
33    LEFT JOIN LaunchSite ls ON ls.SiteID = (SELECT TOP 1 SiteID FROM MissionLaunch WHERE MissionID = m.MissionID ORDER BY LaunchDate DESC)
34    LEFT JOIN Spacecraft s ON m.MissionID = s.MissionID
35    LEFT JOIN LaunchVehicle lv ON m.MissionID = lv.MissionID
36
37 
```

Results

MissionID	MissionName	MissionStartDate	EstimatedEndDate	Duration	MissionStatus	Destination
M105	Venus Surface Study	2024-11-30	2025-03-30	120	Active	Venus
M108	Lunar South Pole Water Survey	2024-12-18	2025-02-16	60	Active	Moon
M110	International Space Station Resupply	2024-06-21	2024-07-21	30	Active	Low Earth Orbit
M114	Lunar Base Construction Phase 1	2025-02-20	2025-08-19	180	Active	Moon
M115	Global Satellite Internet Constellation Deployment	2024-04-01	2025-04-01	365	Active	Low Earth Orbit
M118	Space Junk Cleanup Initiative	2025-10-26	2026-10-26	365	Active	Low Earth Orbit

User Defined Functions(5)

```

-- UDF 5
-- UDF to Check Payload Capacity vs. Weight
-- This UDF will determine whether the total weight of payloads for a given mission is within the payload capacity of the assigned spacecraft.
GO

CREATE FUNCTION dbo.fn_CheckPayloadCapacity
(
    @MissionID VARCHAR(255)
)
RETURNS VARCHAR(100)
AS
BEGIN
    DECLARE @TotalWeight DECIMAL(18,2);
    DECLARE @PayloadCapacity DECIMAL(18,2);

    -- Calculate the total weight of payloads assigned to the mission
    SELECT @TotalWeight = ISNULL(SUM(Payload.Weight), 0)
    FROM Payload
    INNER JOIN PayloadPurpose ON Payload.PayloadID = PayloadPurpose.PayloadID
    WHERE PayloadPurpose.MissionID = @MissionID;

    -- Retrieve the payload capacity of the spacecraft assigned to the mission
    SELECT @PayloadCapacity = ISNULL(Spacecraft.PayloadCapacity, 0)
    FROM Spacecraft
    WHERE Spacecraft.MissionID = @MissionID;

    -- Determine if the total payload weight exceeds the spacecraft's capacity
    IF @TotalWeight > @PayloadCapacity
        RETURN 'Exceeds Capacity';
    RETURN 'Within Capacity';
END;
 
```

Results

MissionID	CapacityCheck
M102	Within Capacity
M103	Within Capacity
M104	Within Capacity
M105	Exceeds Capacity
M106	Exceeds Capacity
M107	Within Capacity
M108	Within Capacity

Triggers(4)

```
--This trigger ensures that the sum of the mission budgets doesn't exceed the
--agency budget when updating a mission
CREATE TRIGGER tr_CheckAgencyBudget1
ON Mission
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    -- Get the inserted or updated rows
    SELECT ma.AgencyID, SUM(i.MissionBudget) AS TotalMissionBudget
    INTO #TempTable
    FROM inserted i
    INNER JOIN MissionAgencyAssignment ma ON i.MissionID = ma.MissionID
    GROUP BY ma.AgencyID;

    -- Check if the total mission budget exceeds the agency budget
    IF EXISTS (
        SELECT 1
        FROM #TempTable t
        INNER JOIN Agency a ON t.AgencyID = a.AgencyID
        WHERE t.TotalMissionBudget > a.AgencyBudget
    )
    BEGIN
        RAISERROR('The total mission budget exceeds the agency budget.', 16, 1);
        ROLLBACK TRANSACTION;
    END
END
GO
```

Non-clustered Indexes(7)

```
CREATE INDEX idx_Astronaut_LastName ON Astronaut(LastName);
CREATE INDEX idx_Astronaut_Nationality ON Astronaut(Nationality);

CREATE INDEX idx_Mission_MissionName ON Mission(MissionName);
CREATE INDEX idx_Mission_MissionStatus ON Mission(MissionStatus);
CREATE INDEX idx_Mission_Destination ON Mission(Destination);

CREATE INDEX idx_Agency_AgencyName ON Agency(AgencyName);
CREATE INDEX idx_Agency_Country ON Agency(Country);
```

Data Encryption(2)

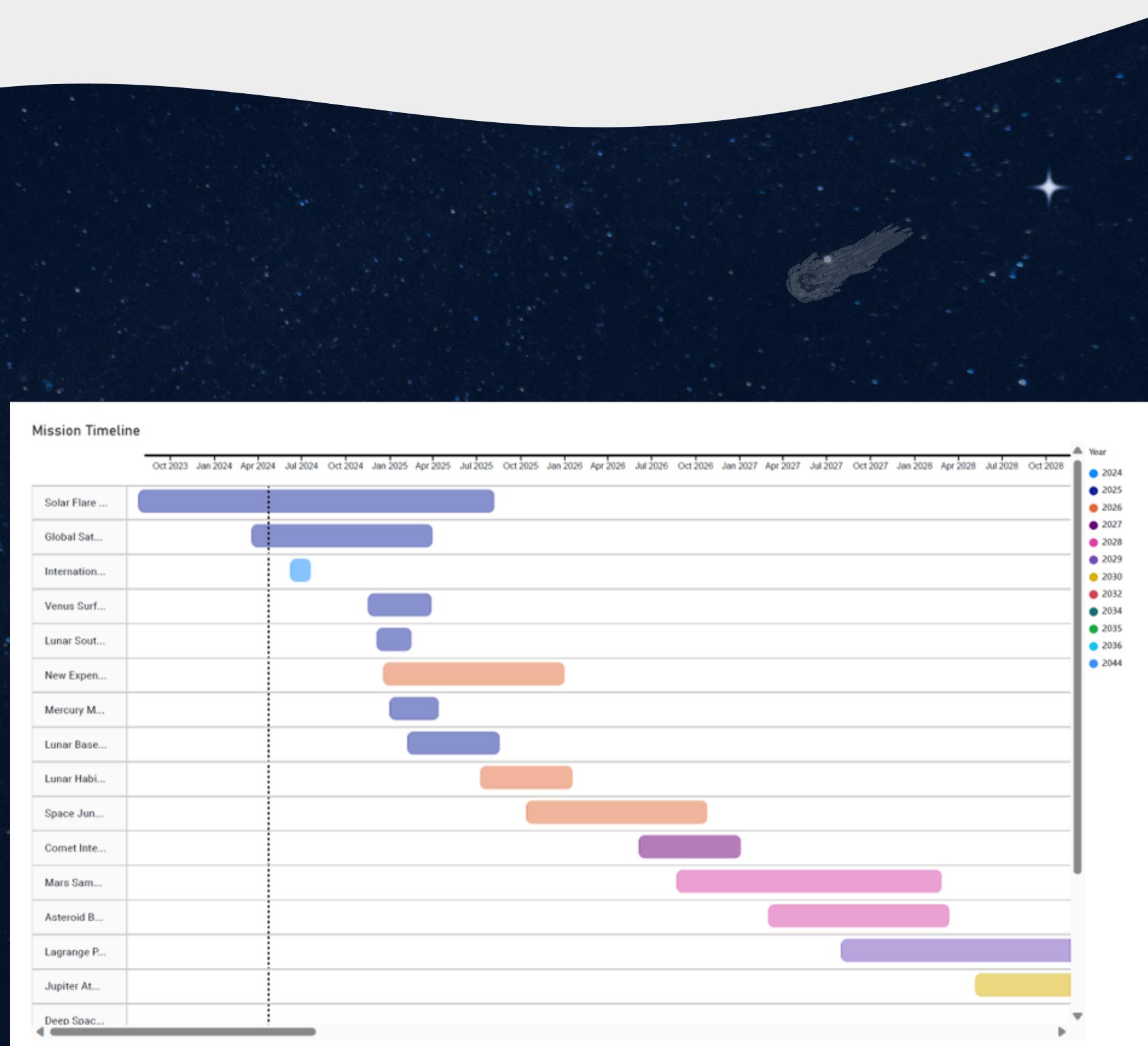
```
5  CREATE MASTER KEY
6  ENCRYPTION BY PASSWORD = 'CosmoPlan24';
7
8  -- Verify that master key exists
9  SELECT name KeyName,
10 symmetric_key_id KeyID,
11 key_length KeyLength,
12 algorithm_desc KeyAlgorithm
13 FROM sys.symmetric_keys;
14 go
15
16
17 -- Create a self-signed certificate for Manufacturer and Location
18 CREATE CERTIFICATE ManufacturerCert
19 WITH SUBJECT = 'Manufacturer Encryption Certificate';
20
21 CREATE CERTIFICATE LocationCert
22 WITH SUBJECT = 'Location Encryption Certificate';
23
24 -- Create symmetric keys for Manufacturer and Location encryption
25 CREATE SYMMETRIC KEY ManufacturerKey
26 WITH ALGORITHM = AES_256
27 ENCRYPTION BY CERTIFICATE ManufacturerCert;
28
29 CREATE SYMMETRIC KEY LocationKey
30 WITH ALGORITHM = AES_256
31 ENCRYPTION BY CERTIFICATE LocationCert;
32
33 -- Encrypt the Manufacturer column in LaunchVehicle table
34 OPEN SYMMETRIC KEY ManufacturerKey
35 DECRYPTION BY CERTIFICATE ManufacturerCert;
36 |
37 UPDATE LaunchVehicle
38 SET Manufacturer = ENCRYPTBYKEY(KEY_GUID('ManufacturerKey'), Manufacturer);
39
40 -- Encrypt the Location column in LaunchSite table
41 OPEN SYMMETRIC KEY LocationKey
42 DECRYPTION BY CERTIFICATE LocationCert;
43
44 UPDATE LaunchSite
45 SET Location = ENCRYPTBYKEY(KEY_GUID('LocationKey'), Location);
```

Results Messages

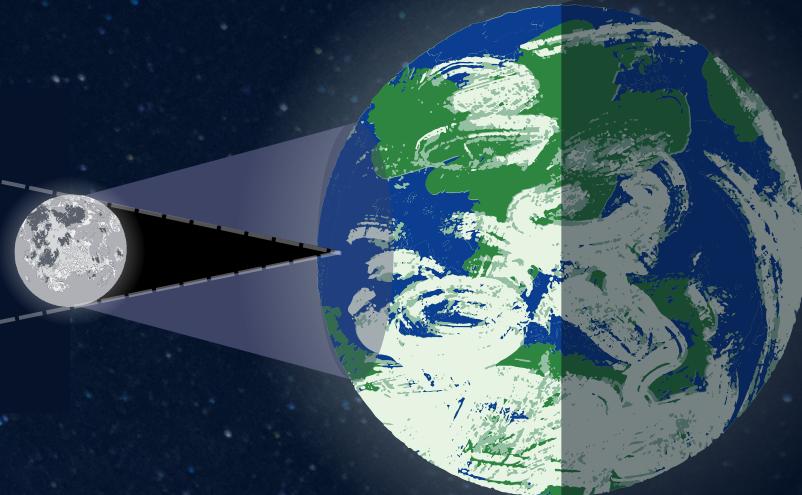
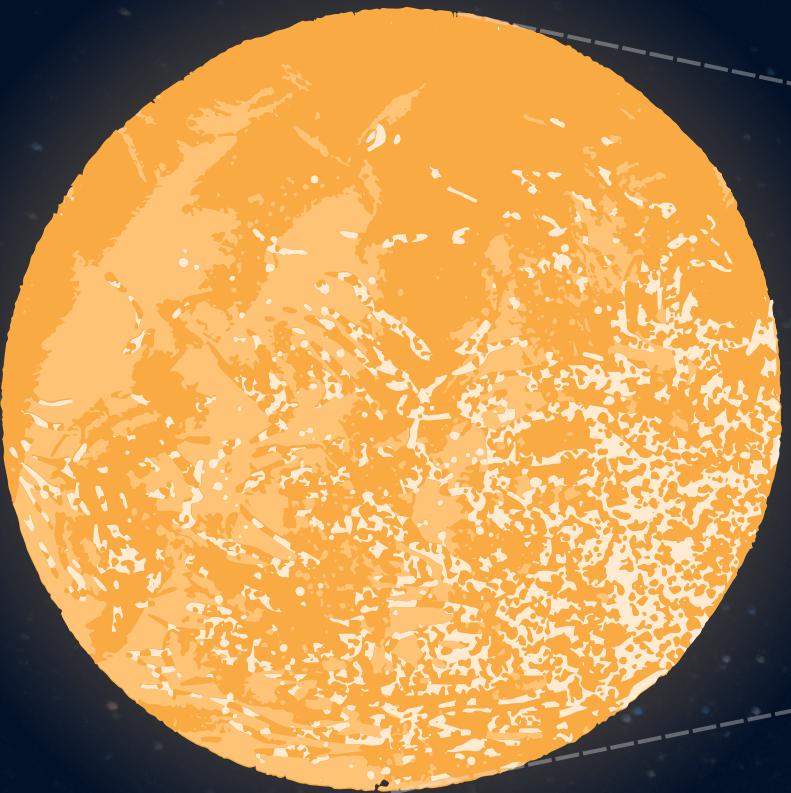
VehicleID	VehicleName	VehicleType	PayloadCapacity	Manufacturer	MissionID	DecryptedManufacturer
2	Atlas V	Medium Lift	20000.0000	_UD8...çM¢¹ ?ÊHÜ× C]•éc\$tr4•«Ñµ xYŽš~ ſw,òFæñdí Ó	M102	ULA
3	SLS Block 1	Super Heavy Lift	95000.0000	_UD8...çM¢¹ ?ÊHÜ× M~>„F'FýÜ ŠEZ ½gÈ!~AÍ³± , Ó,	M103	NASA
4	Starship Mars	Super Heavy Lift	100000.0000	_UD8...çM¢¹ ?ÊHÜ× 'ôfÑ>p+yþæ fV¾QRø1] "g'êé¾"žÅšê	M104	SpaceX

SiteID	SiteName	Location	LaunchCapacity	DecryptedLocation
1	Cape Canaveral	f&žDË‰B‰L*I‡=1 ī ZÅ# L‡ & ê°t}...ŽSP°Cà- µ+Lxf5 Z#þ...	40	Florida, USA
2	Vandenberg Air Force Base	f&žDË‰B‰L*I‡=1 & A RóÚeíé °G [= Y?JDø[i-NSiqi)ü ý}lZ...	20	California, USA
3	Baikonur Cosmodrome	f&žDË‰B‰L*I‡=1 %,3·â€«iQf "rI®?J"7a¾4,¤çí@*ÓQDÂTÉM...	30	Kazakhstan

PowerBi Dashboard



GUI & Demo



Astronaut Management

Astronaut ID	<input type="text"/>
Call Sign	<input type="text"/>
First Name	<input type="text"/>
Last Name	<input type="text"/>
Date of Birth (YYYY-MM-DD)	<input type="text"/>
Nationality	<input type="text"/>
Gender (M/F/O)	<input type="text"/>
Specialty	<input type="text"/>

Action Buttons:

- Create
- Read
- Update
- Delete

Mission Management

Mission ID	<input type="text" value="M122"/>
Mission Name	<input type="text" value="Indian Space Mission"/>
Start Date (YYYY-MM-DD)	<input type="text" value="2025-07-24"/>
Duration (days)	<input type="text" value="170"/>
Status	<input type="text" value="Planned"/>
Destination	<input type="text" value="Mars"/>
Budget	<input type="text" value="5000.00"/>

Action Buttons:

- Create
- Read
- Update
- Delete

Thank You

Any Questions?

