

CPSC 304 Project Cover Page

Milestone #: 2

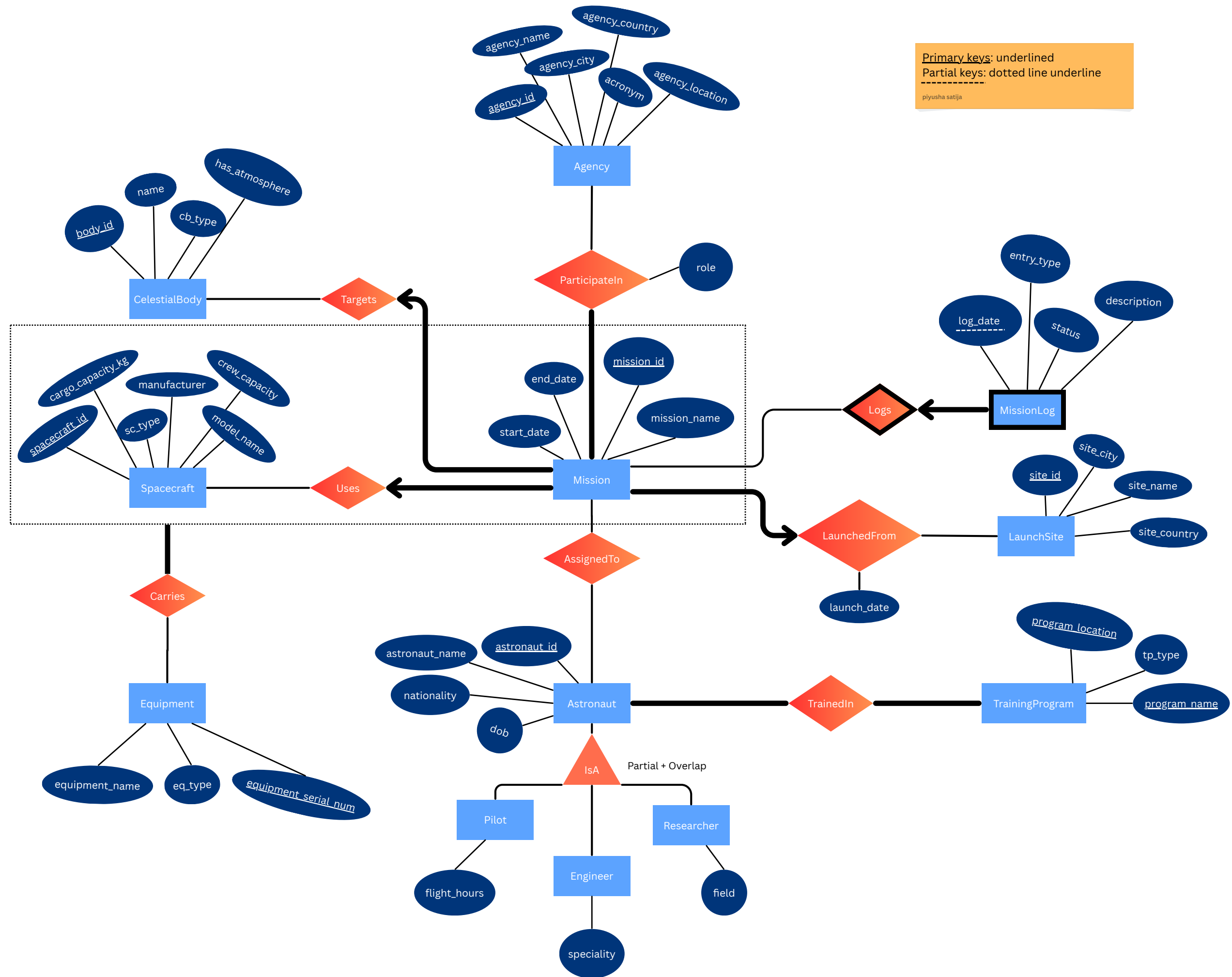
Date: Jul 23, 2025

Group Number: 1

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Piyusha Satija	39217971	psatij01	piyushasatija9@gmail.com
Eason Feng	87333597	fgy1223	fengguoyi1223@gmail.com
Avireet Uppal	45588489	auppal08	auppal73@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia



Milestone 2

1. Project Description

Our project is a Space Mission Tracker that helps users manage and track various aspects of space missions. It models entities such as missions, astronauts, agencies, spacecrafts, equipment, launch sites, training programs, mission logs and celestial bodies, and supports operations like assigning astronauts, logging mission details, tracking spacecraft used and overall tracking mission progress.

2. ER Diagram (attached on a separate page) Modifications:

(green are attributes, red are entities and purple are relations)

- Added attributes *model_name*, *crew_capacity* and *cargo_capacity_kg* to the *Spacecraft* entity to introduce additional non-PK/CK FDs.
- Removed *spacecraft_name* attribute from *Spacecraft* and moved it to *Mission* instead (spacecrafts can be reused - a mission can have a different name for each spacecraft).
- Added *agency_city* attribute to *Agency* and *site_city* to *LaunchSite* entities to introduce additional non-PK/CK FD.
- Corrected primary key/partial key notation per TA's feedback.
- Added *equipment_serial_num* attribute to *Equipment* and made it the primary key instead of *equipment_name* per TA's feedback.
- Modified relation *Uses* between *Mission* and *Spacecraft* from many-to-many to many-to-one and aggregated the relation to simplify design and make it more cohesive (since relation *LaunchedFrom* between *Mission* and *LaunchSite* is many-to-one). The reason for aggregation was to ensure that only a spacecraft in use by a mission needs to carry at least one equipment; spacecrafts not in use don't need to have this constraint. And, it makes more sense for spacecrafts to carry equipment if they are part of a mission.
- Modified names of the *type* attribute of all entities by appending entity name to it - making it more specific.
- Added descriptive attribute *launch_date* to the relation *LaunchedFrom* between *Mission* and *LaunchSite*.

- Added attribute *acronym* to *Agency* entity to allow for full names as well as acronyms of space agencies.
- Modified primary key of *TrainingProgram* from *program_name* to *program_name* and *location*.

3. Relational Schema, FDs and Normalization:

PK

FK

- **Equipment** (equipment_serial_num: VARCHAR(30), equipment_name: VARCHAR(100))

Candidate Keys: {equipment_serial_num}

FDs:

equipment_serial_num → equipment_name, eq_type (superkey in LHS, in BCNF, already normalized)

- **Mission** (mission_id: VARCHAR(30), **site_id**: VARCHAR(30), **body_id**: VARCHAR(30), **spacecraft_id**: VARCHAR(30), spacecraft_name: VARCHAR(30), mission_name: VARCHAR(100), start_date: DATE, end_date: DATE, launch_date: DATE)

site_id: NOT NULL

body_id: NOT NULL

spacecraft_id: NOT NULL

Candidate Keys: {mission_id}, {site_id, body_id, spacecraft_id}

FDs:

mission_id → mission_name, spacecraft_name, start_date, end_date, launch_date, site_id, body_id, spacecraft_id (superkey in LHS, in BCNF, already normalized)

- **MissionLog** (log_date: DATE, **mission_id**: VARCHAR(30), entry_type: VARCHAR(100), status: VARCHAR(100), description: VARCHAR(100))

Candidate Keys: {Log_date, mission_id}

FDs:

mission_id, log_date → entry_type, status, description (superkey in LHS, in BCNF, already normalized)

- **LaunchSite** (site_id: VARCHAR(30), site_city: VARCHAR(100), site_name: VARCHAR(30), site_country: VARCHAR(30))

Candidate Keys: {site_id}, {site_name}

FDs:

site_id \rightarrow site_name, site_city, site_country (superkey in LHS, in BCNF, already normalized)

site_name \rightarrow site_city, site_country

Normalization:

site_name+ : {site_name, site_city, site_country}

Non-trivial FDs:

site_name \rightarrow site_city

site_name \rightarrow site_country

Since site_name is not a superkey of LaunchSite, we decompose on site_name \rightarrow site_city:
R1(site_name, site_city), R2(site_name, site_id, site_country)

R1: in BCNF, site_name is a superkey.

Now, in R2: site_name is not a superkey, decompose again on second FD: R3(site_name, site_country), R4(site_name, site_id)

R3: in BCNF, site_name is a superkey.

R4: in BCNF, site_id is a superkey.

Decomposition:

R1(site_name, site_city), R3(site_name, site_country) and R4(site_name, site_id)

While BCNF (according to lectures) requires a single attribute on the right, following this however gave us site_country and site_city in separate relations. It makes more sense to keep them together with site_name.

Final decomposition:

R5(site_name, site_city, site_country) and R4(**site_name**, site_id)

Renaming tables:

LaunchSiteLocation(site_name, site_city, site_country) and LaunchSite(**site_name**, site_id)

Modified Schema:

LaunchSiteLocation(site_name:VARCHAR(30), site_city:VARCHAR(100),
site_country:VARCHAR(30))

Candidate Keys: {site_name}

LaunchSite(site_id:VARCHAR(30), **site_name**:VARCHAR(30))

Candidate Keys: {site_id}

- **Spacecraft** (spacecraft_id: VARCHAR(30), model_name: VARCHAR(50), manufacturer: VARCHAR(30), sc_type: VARCHAR(10), crew_capacity: INTEGER, cargo_capacity: DECIMAL(8,2))

Candidate Keys: {spacecraft_id}

FDs:

spacecraft_id → model_name, manufacturer, sc_type, crew_capacity, cargo_capacity_kg
(superkey in LHS, in BCNF, already normalized)

manufacturer, model_name → sc_type, crew_capacity, cargo_capacity_kg

Normalization:

manufacturer, model_name+: {manufacturer, model_name, sc_type, crew_capacity, cargo_capacity_kg}

Non-trivial FDs:

manufacturer, model_name → sc_type

manufacturer, model_name → crew_capacity

manufacturer, model_name → cargo_capacity_kg

Since (manufacturer, model_name) is not a superkey of Spacecraft, we decompose on
manufacturer, model_name → sc_type: R1(manufacturer, model_name, sc_type),
R2(manufacturer, model_name, crew_capacity, cargo_capacity_kg, spacecraft_id)

R1: in BCNF, (manufacturer, model_name) is a superkey.

Now, in R2: (manufacturer, model_name) is not a superkey, decompose again on second FD:
R3(manufacturer, model_name, crew_capacity), R4(manufacturer, model_name,
cargo_capacity_kg, spacecraft_id)

R3: in BCNF, (manufacturer, model_name) is a superkey.

R4: (manufacturer, model_name) is not a superkey, decompose on third FD: R5(manufacturer,
model_name, cargo_capacity_kg), R6(manufacturer, model_name, spacecraft_id)

BCNF leads to too many relations, instead normalizing in 3NF to preserve dependencies:

Since (manufacturer, model_name) violates 3NF, find minimal cover:

manufacturer, model_name → sc_type

manufacturer, model_name → crew_capacity

manufacturer, model_name → cargo_capacity_kg

Minimize LHS of any FDs: N/A since manufacturer+ : {manufacturer} and model_name+ : {model_name}

Remove unnecessary FDs: N/A

Therefore, all original FDs are in minimal cover.

Decompose: R1(manufacturer, model_name, sc_type, crew_capacity, cargo_capacity_kg)

Add relation with key: R2(spacecraft_id, **manufacturer**, **model_name**)

Final decomposition:

R1(manufacturer, model_name, sc_type, crew_capacity, cargo_capacity_kg), R2(spacecraft_id, **manufacturer**, **model_name**)

Renaming tables:

SpacecraftModel(manufacturer, model_name, sc_type, crew_capacity, cargo_capacity_kg),
Spacecraft(spacecraft_id, **manufacturer**, **model_name**)

Schema:

SpacecraftModel(manufacturer:VARCHAR(30), model_name:VARCHAR(50),
sc_type:VARCHAR(10), crew_capacity:INTEGER, cargo_capacity_kg:DECIMAL(8,2))

Candidate Keys: {manufacturer, model_name}

Spacecraft(spacecraft_id:VARCHAR(30), **manufacturer**:VARCHAR(30),
model_name:VARCHAR(50))

Candidate Keys: {spacecraft_id}

- **Carries** (spacecraft_id:VARCHAR(30), mission_id:VARCHAR(30),
equipment_serial_number:VARCHAR(30))

Candidate Key: {spacecraft_id, mission_id, equipment_serial_number}

FDs:

none

- **Agency** (agency_id: INTEGER, agency_name: VARCHAR(100), acronym: VARCHAR(10), agency_location: VARCHAR(100), agency_city: VARCHAR(30), agency_country: VARCHAR(30))

agency_name: UNIQUE, NOT NULL

acronym: UNIQUE

Candidate Keys: {agency_id}, {agency_name, agency_location}

FDs:

agency_id → agency_name, acronym, agency_location, agency_city, agency_country (superkey in LHS, in BCNF, already normalized)

agency_name → acronym

agency_location, agency_city → agency_country

Normalization:

agency_name+ : {agency_name, acronym}

agency_location, agency_city+: {agency_location, agency_city, agency_country}

Non-trivial FDs:

agency_name → acronym

agency_location, agency_city → agency_country

Since agency_name is not a superkey of Agency, we decompose: R1(agency_name, acronym), R2(agency_name, agency_id, agency_location, agency_city, agency_country)

R1: in BCNF, agency_name is a superkey.

Now, in R2: (agency_location, agency_city) is not a superkey, decompose again on second FD:

R3(agency_location, agency_city, agency_country), R4(agency_location, agency_city, agency_name, agency_id)

R3: in BCNF, (agency_location, agency_city) is a superkey.

Final decomposition:

R1(agency_name, acronym), R3(agency_location, agency_city, agency_country) and R4(agency_location, agency_city, agency_name, agency_id)

Renaming tables:

AgencyName(agency_name, acronym), AgencyLocation(agency_location, agency_city, agency_country) and Agency(agency_location, agency_city, agency_name, agency_id)

Modified Schema:

AgencyName(agency_name:VARCHAR(100), acronym:VARCHAR(10))

Candidate Keys: {agency_name}

AgencyLocation(agency_location:VARCHAR(100), agency_city:VARCHAR(30),
agency_country:VARCHAR(30))

Candidate Keys: {agency_location, agency_city}

Agency(**agency_location**:VARCHAR(100), **agency_name**:VARCHAR(100),
agency_id:INTEGER)

Candidate Keys: {agency_id}, {agency_location, agency name}

- **ParticipateIn** (agency_id: INTEGER, mission_id: VARCHAR(30), role: VARCHAR(50))

Candidate Keys: {agency_id, mission_id}

FDs:

none

- **Astronaut** (astronaut_id: INTEGER, astronaut_name: VARCHAR(50),
nationality: VARCHAR(5), dob: DATE)

Candidate Keys: {astronaut_id}

FDs:

astronaut_id → astronaut_name, nationality, dob (trivial, in BCNF, already normalized)

- **Pilot** (astronaut_id: INTEGER, flight_hours DECIMAL(6,2))

Candidate Keys: {astronaut_id}

FDs:

astronaut_id → flight_hours (trivial, in BCNF, already normalized)

- **Engineer** (astronaut_id: INTEGER, flight_hours DECIMAL(6,2))

Candidate Keys: {astronaut_id}

FDs:

astronaut_id → specialty (trivial, in BCNF, already normalized)

- **Researcher** (astronaut_id: INTEGER, field: VARCHAR(30))

Candidate Keys: {astronaut_id}

FDs:

astronaut_id → field (trivial, in BCNF, already normalized)

- **AssignedTo** (astronaut_id: INTEGER, mission_id: VARCHAR(30))

Candidate Keys: {astronaut_id, mission_id}

FDs:

none

- **TrainingProgram** (program_name: VARCHAR(50), program_location: VARCHAR(100), tp_type: VARCHAR(50))

Candidate Keys: program_name, program_location}

tp_type: NOT NULL

FDs:

program_name, program_location → tp_type (trivial, in BCNF, already normalized)

- **TrainedIn** (astronaut_id:INTEGER, program_name:VARCHAR(50), program_location:VARCHAR(100))

Candidate Keys: {astronaut_id, program_name, program_location}

FDs:

none

4. SQL DDL CREATE:

- **CREATE TABLE Equipment**
(equipment_serial_num VARCHAR(30) PRIMARY KEY,
equipment_name VARCHAR(100),
eq_type VARCHAR(30));
- **CREATE TABLE Mission**
(mission_id VARCHAR(30) PRIMARY KEY,

```
site_id VARCHAR(30) NOT NULL,  
body_id VARCHAR(30) NOT NULL,  
spacecraft_id VARCHAR(30) NOT NULL,  
spacecraft_name VARCHAR(30),  
mission_name VARCHAR(100),  
start_date DATE,  
end_date DATE,  
launch_date DATE,  
FOREIGN KEY (site_id) REFERENCES LaunchSite,  
FOREIGN KEY (body_id) REFERENCES CelestialBody,  
FOREIGN KEY (spacecraft_id) REFERENCES Spacecraft);
```

- **CREATE TABLE MissionLog**
(log_date DATE,
mission_id VARCHAR(30),
entry_type VARCHAR(100),
status VARCHAR(100),
description VARCHAR(100)
PRIMARY KEY (log_date, mission_id),
FOREIGN KEY (mission_id) REFERENCES Mission);
- **CREATE TABLE LaunchSite**
(site_id VARCHAR(30) PRIMARY KEY,
site_name VARCHAR(50),
FOREIGN KEY (site_name) REFERENCES LaunchSiteLocation);
- **CREATE TABLE LaunchSiteLocation**
(site_name VARCHAR(50) PRIMARY KEY,
site_city VARCHAR(50),
site_country VARCHAR(30));
- **CREATE TABLE CelestialBody**
(body_id VARCHAR(30) PRIMARY KEY,
name VARCHAR(100),
cb_type VARCHAR(30),
has_atmosphere BOOLEAN);
- **CREATE TABLE Spacecraft**
(spacecraft_id VARCHAR(30) PRIMARY KEY,
model_name VARCHAR(50),
manufacturer VARCHAR(30),
FOREIGN KEY (model_name, manufacturer) REFERENCES SpacecraftModel);

- **CREATE TABLE SpacecraftModel**
 (model_name VARCHAR(50),
 manufacturer VARCHAR(30),
 sc_type VARCHAR(10),
 crew_capacity INTEGER,
 cargo_capacity_kg DECIMAL(8,2),
 PRIMARY KEY (manufacturer, model_name));
 *sc_types: cargo, crew, hybrid
- **CREATE TABLE Carries**
 (spacecraft_id VARCHAR(30),
 mission_id VARCHAR(30),
 equipment_serial_num VARCHAR(30),
 PRIMARY KEY (spacecraft_id, mission_id, equipment_serial_num),
 FOREIGN KEY (spacecraft_id) REFERENCES Spacecraft,
 FOREIGN KEY (mission_id) REFERENCES Mission,
 FOREIGN KEY (equipment_serial_num) REFERENCES Equipment);
 *this table requires assertions since each spacecraft used by a mission must carry at least one piece of equipment
- **CREATE TABLE Agency**
 (agency_id INTEGER PRIMARY KEY,
 agency_name VARCHAR(100) UNIQUE NOT NULL,
 agency_location VARCHAR(100),
 agency_city VARCHAR(30),
 FOREIGN KEY (agency_location, agency_city) REFERENCES AgencyLocation,
 FOREIGN KEY (agency_name) REFERENCES AgencyName);
 *all agencies must have names but may not have acronyms, agency_id self-assigned to avoid duplication (mismatch names)
- **CREATE TABLE AgencyName**
 (agency_name VARCHAR(100) PRIMARY KEY,
 acronym VARCHAR(10) UNIQUE);
- **CREATE TABLE AgencyLocation**
 (agency_location VARCHAR(100),
 agency_city VARCHAR(30),
 agency_country VARCHAR(30)
 PRIMARY KEY (agency_location, agency_city));
- **CREATE TABLE ParticipateIn**
 (agency_id INTEGER,
 mission_id VARCHAR(30),

role VARCHAR(50),
PRIMARY KEY (agency_id, mission_id),
FOREIGN KEY (agency_id) REFERENCES Agency,
FOREIGN KEY (mission_id) REFERENCES Mission);
*this table requires assertions since each mission must have at least one agency

- **CREATE TABLE Astronaut**
(astronaut_id INTEGER PRIMARY KEY,
astronaut_name VARCHAR(50),
nationality VARCHAR(5),
dob DATE);
*nationality = ISO country codes - usually 2-3 characters but max of 5 to be safe
- **CREATE TABLE Pilot**
(astronaut_id INTEGER PRIMARY KEY,
flight_hours DECIMAL(6,2),
FOREIGN KEY (astronaut_id) REFERENCES Astronaut);
- **CREATE TABLE Engineer**
(astronaut_id INTEGER PRIMARY KEY,
speciality VARCHAR(30),
FOREIGN KEY (astronaut_id) REFERENCES Astronaut);
- **CREATE TABLE Researcher**
(astronaut_id INTEGER PRIMARY KEY,
field VARCHAR(30),
FOREIGN KEY (astronaut_id) REFERENCES Astronaut);
- **CREATE TABLE AssignedTo**
(astronaut_id INTEGER,
mission_id VARCHAR(30),
PRIMARY KEY (astronaut_id, mission_id),
FOREIGN KEY (astronaut_id) REFERENCES Astronaut,
FOREIGN KEY (mission_id) REFERENCES Mission);
- **CREATE TABLE TrainingProgram**
(program_name VARCHAR(50),
program_location VARCHAR(100),
tp_type VARCHAR(50) NOT NULL,
PRIMARY KEY (program_name, program_location));
- **CREATE TABLE TrainedIn**
(astronaut_id INTEGER,

```
program_name VARCHAR(50),
program_location VARCHAR(100),
PRIMARY KEY (astronaut_id, program_name, program_location),
FOREIGN KEY (astronaut_id) REFERENCES Astronaut,
FOREIGN KEY (program_name, program_location) REFERENCES TrainingProgram);
*this table requires assertions since each Astronaut must have trained in a training program
and each training program must have at least one Astronaut enrolment
```

5. INSERT Statements:

- **Equipment:**

```
INSERT INTO Equipment (equipment_serial_num, equipment_name, eq_type) VALUES
('EQ-001', 'Canadarm2', 'Robotic Arm'),
('EQ-002', 'Zvezda Module Solar Array', 'Power System'),
('EQ-003', 'Advanced Resistive Exercise Device (ARED)', 'Exercise Equipment'),
('EQ-004', 'APXS', 'Scientific Instrument'),
('EQ-005', 'VEGA Soil Sampler', 'Sampling Tool'),
('EQ-006', 'Cameras - NavCam', 'Navigation Instrument'),
('EQ-007', 'Ku-Band Antenna System', 'Communication');
```

- **Mission:**

```
INSERT INTO Mission (mission_id, site_id, body_id, spacecraft_id, spacecraft_name,
mission_name, start_date, end_date, launch_date) VALUES
('M001', 'LS001', 'CB002', 'SC001', 'Orion CM-002', 'Artemis I', '2022-11-16', '2022-12-11',
'2022-11-16'),
('M002', 'LS001', 'CB006', 'SC002', 'Lucy', 'Lucy Mission', '2021-10-16', NULL, '2021-10-16'),
('M003', 'LS001', 'CB007', 'SC003', 'OSIRIS-REx', 'OSIRIS-REx Sample Return', '2016-09-08',
'2023-09-24', '2016-09-08'),
('M004', 'LS006', 'CB004', 'SC004', 'VSS Unity', 'Galactic 06', '2024-01-26', '2024-01-26',
'2024-01-26'),
('M005', 'LS001', 'CB008', 'SC005', 'Psyche', 'Psyche Asteroid Orbiter', '2023-10-13', NULL,
'2023-10-13');
```

- **MissionLog:**

```
INSERT INTO MissionLog (log_date, mission_id, entry_type, status, description) VALUES
('2022-11-16', 'M101', 'Launch', 'Success', 'Artemis I launched successfully from Kennedy
Space Center.'),
('2021-10-17', 'M102', 'Status Update', 'Nominal', 'Lucy spacecraft systems check complete.
All instruments operational.'),
```

```
('2018-12-03', 'M103', 'Arrival', 'Success', 'OSIRIS-REx arrived at asteroid Bennu.'),
('2024-01-26', 'M104', 'Arrival', 'Success', 'Galactic 06 landed safely after suborbital flight.'),
('2023-10-13', 'M105', 'Launch', 'Success', 'Psyche mission launched toward asteroid 16 Psyche.');
```

- **LaunchSite:**

```
INSERT INTO LaunchSite (site_id, site_name) VALUES
('LS001', 'Kennedy Space Center'),
('LS002', 'Cape Canaveral Space Force Station'),
('LS003', 'Baikonur Cosmodrome'),
('LS004', 'Vandenberg Space Force Base'),
('LS005', 'Tanegashima Space Center'),
('LS006', 'Spaceport America');
```

- **LaunchSiteLocation:**

```
INSERT INTO LaunchSiteLocation (site_name, site_city, site_country) VALUES
('Kennedy Space Center', 'Merritt Island', 'USA'),
('Cape Canaveral Space Force Station', 'Cape Canaveral', 'USA'),
('Baikonur Cosmodrome', 'Baikonur', 'Kazakhstan'),
('Vandenberg Space Force Base', 'Lompoc', 'USA'),
('Tanegashima Space Center', 'Kagoshima', 'Japan'),
('Spaceport America', 'Truth or Consequences', 'USA');
```

- **CelestialBody:**

```
INSERT INTO CelestialBody (body_id, name, cb_type, has_atmosphere) VALUES
('CB001', 'Moon', 'Natural Satellite', FALSE),
('CB002', 'Mars', 'Planet', TRUE),
('CB003', 'ISS', 'Artificial Satellite', TRUE),
('CB004', 'Suborbital Space', 'Space Environment', FALSE),
('CB005', 'Venus', 'Planet', TRUE),
('CB006', 'Jupiter Trojan Asteroids', 'Asteroid Cluster', FALSE),
('CB007', 'Bennu', 'Asteroid', FALSE),
('CB008', 'Psyche', 'Asteroid', FALSE);
```

- **Spacecraft:**

```
INSERT INTO Spacecraft (spacecraft_id, model_name, manufacturer) VALUES
('SC001', 'Orion', 'NASA'),
('SC002', 'Lucy', 'Lockheed Martin'),
('SC003', 'OSIRIS-REx', 'Lockheed Martin'),
```

```
('SC004', 'VSS Unity', 'Virgin Galactic'),  
( 'SC005', 'Psyche', 'Maxar Technologies');
```

- **SpacecraftModel:**

```
INSERT INTO SpacecraftModel (model_name, manufacturer, sc_type, crew_capacity,  
cargo_capacity_kg) VALUES  
( 'Orion', 'NASA', 'crew', 4, 1000.00),  
( 'Lucy', 'Lockheed Martin', 'cargo', 0, 500.00),  
( 'OSIRIS-REx', 'Lockheed Martin', 'cargo', 0, 150.00),  
( 'VSS Unity', 'Virgin Galactic', 'crew', 6, 0.00),  
( 'Psyche', 'Maxar Technologies', 'cargo', 0, 2800.00);
```

- **Carries:**

```
INSERT INTO Carries (spacecraft_id, mission_id, equipment_serial_num) VALUES  
( 'SC001', 'M001', 'EQ-001'),  
( 'SC002', 'M002', 'EQ-005'),  
( 'SC003', 'M003', 'EQ-004'),  
( 'SC004', 'M004', 'EQ-007'),  
( 'SC005', 'M005', 'EQ-006');
```

- **Agency:**

```
INSERT INTO Agency (agency_id, agency_name, agency_location, agency_city) VALUES  
(1, 'National Aeronautics and Space Administration', 'Lyndon B. Johnson Space Center',  
'Houston'),  
(2, 'European Space Agency', 'European Space Research and Technology Centre', 'Noordwijk'),  
(3, 'Japan Aerospace Exploration Agency', 'Tsukuba Space Center', 'Tsukuba'),  
(4, 'Indian Space Research Organisation', 'U R Rao Satellite Centre', 'Bangalore'),  
(5, 'Canadian Space Agency', 'John H. Chapman Space Centre', 'Saint-Hubert');
```

- **AgencyName:**

```
INSERT INTO AgencyName ( agency_name, acronym) VALUES  
( 'National Aeronautics and Space Administration', 'NASA'),  
( 'European Space Agency', 'ESA'),  
( 'Japan Aerospace Exploration Agency', 'JAXA'),  
( 'Indian Space Research Organisation', 'ISRO'),  
( 'Canadian Space Agency', 'CSA');
```


- **AgencyLocation:**

```
INSERT INTO AgencyLocation (agency_location, agency_city, agency_country) VALUES  
( 'Lyndon B. Johnson Space Center', 'Houston', 'USA'),  
( 'European Space Research and Technology Centre', 'Noordwijk', 'Netherlands'),  
( 'Tsukuba Space Center', 'Tsukuba', 'Japan'),  
( 'U R Rao Satellite Centre', 'Bangalore', 'India'),  
( 'John H. Chapman Space Centre', 'Saint-Hubert', 'Canada');
```

- **ParticipateIn:**

```
INSERT INTO ParticipateIn (agency_id, mission_id, role) VALUES  
(1, 'M001', 'Lead Agency'),  
(2, 'M002', 'Collaborator'),  
(1, 'M002', 'Lead Agency'),  
(1, 'M003', 'Lead Agency'),  
(3, 'M003', 'Scientific Partner'),  
(5, 'M004', 'Technology Partner');
```

- **Astronaut:**

```
INSERT INTO Astronaut (astronaut_id, astronaut_name, nationality, dob) VALUES  
(1, 'Jessica Watkins', 'USA', '1988-05-14'),  
(2, 'Koichi Wakata', 'JPN', '1963-08-01'),  
(3, 'David Saint-Jacques', 'CAN', '1970-01-06'),  
(4, 'Samantha Cristoforetti', 'ITA', '1977-04-26'),  
(5, 'Rakesh Sharma', 'IND', '1949-01-13');
```

Inserting few into the children of Astronaut to display Partial + Overlap ISA:

- **Pilot:**

```
INSERT INTO Pilot (astronaut_id, flight_hours) VALUES  
(1, 1200.50),  
(2, 2000.75);
```

- **Engineer:**

```
INSERT INTO Engineer (astronaut_id, speciality) VALUES  
(3, 'Electrical Systems'),  
(4, 'Aerospace'),  
(2, 'Mechanical');
```

- **Researcher:**

```
INSERT INTO Researcher (astronaut_id, field) VALUES  
(3, 'Astrophysics'),  
(4, 'Space Medicine'),  
(5, 'Human Factors');
```

- **AssignedTo:**

```
INSERT INTO AssignedTo (astronaut_id, mission_id) VALUES  
(1, 'M001'),  
(2, 'M002'),  
(3, 'M003'),  
(4, 'M005'),  
(5, 'M004');
```

- **TrainingProgram:**

```
INSERT INTO TrainingProgram (program_name, program_location, tp_type) VALUES  
( 'Astronaut Candidate Training', 'Johnson Space Center, Houston', 'Basic'),  
( 'International Spaceflight Training', 'Tsukuba Space Center, Japan', 'Advanced'),  
( 'CSA Robotics Training', 'Saint-Hubert, Quebec', 'Robotics'),  
( 'ESA Science Training', 'European Astronaut Centre, Cologne', 'Science'),  
( 'Gagarin Cosmonaut Training', 'Star City, Russia', 'Survival');
```

- **TrainedIn:**

```
INSERT INTO TrainedIn (astronaut_id, program_name, program_location) VALUES  
(1, 'Astronaut Candidate Training', 'Johnson Space Center, Houston'),  
(2, 'International Spaceflight Training', 'Tsukuba Space Center, Japan'),  
(3, 'CSA Robotics Training', 'Saint-Hubert, Quebec'),  
(4, 'ESA Science Training', 'European Astronaut Centre, Cologne'),  
(5, 'Gagarin Cosmonaut Training', 'Star City, Russia');
```