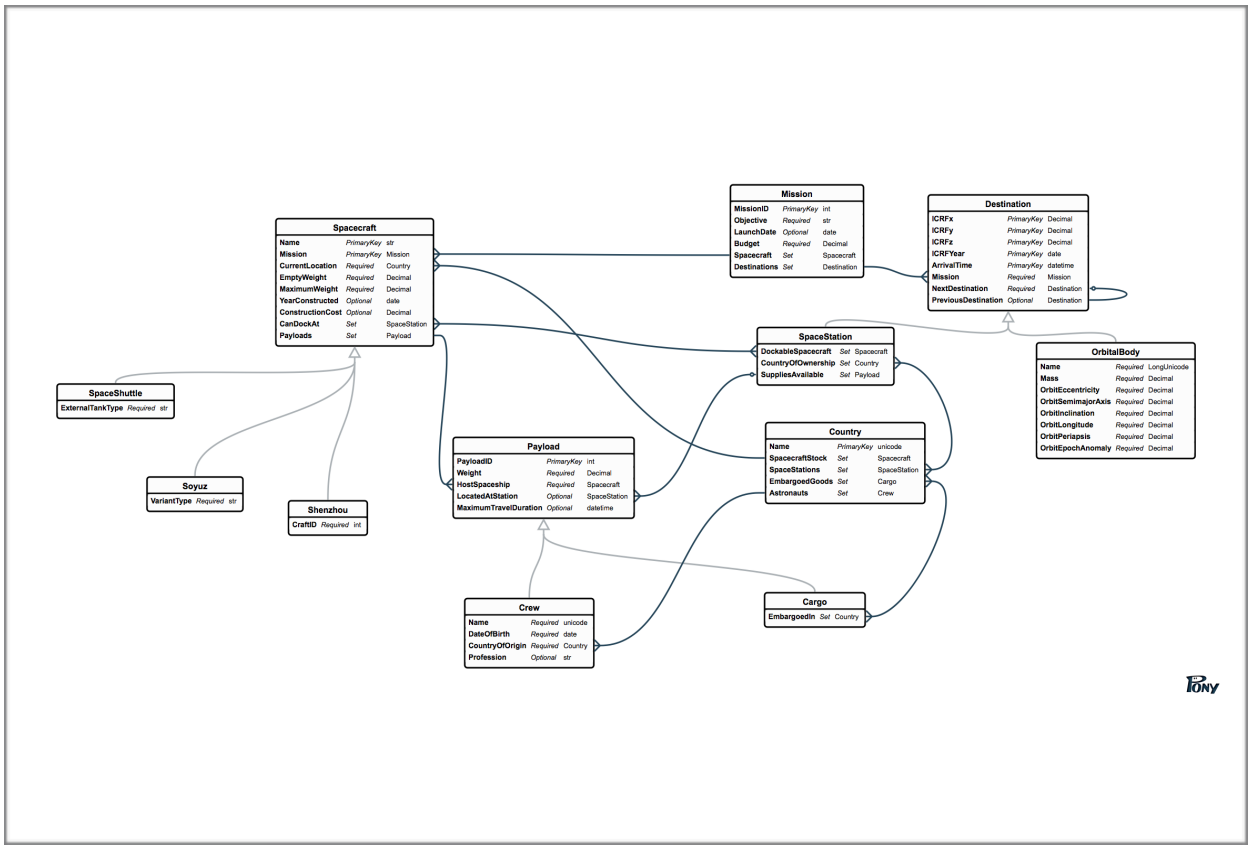
Space Mission ER Model

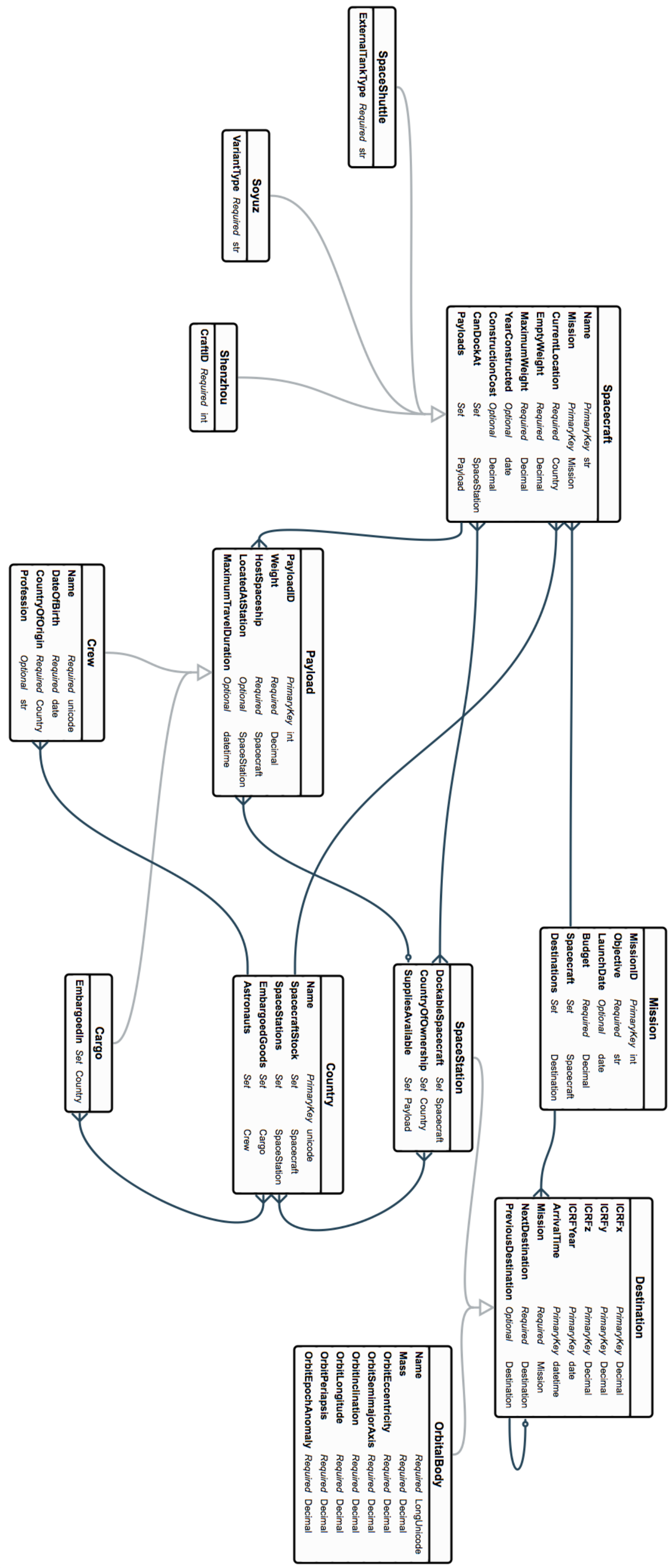
CS 5800 - Database Systems



Jonathan Petersen

Individual Project

Fall 2015


SpaceShuttle
ExternalTankType Required str
Soyuz
VariantType Required str
Shenzhou
CraftID Required int
Spacecraft
Name PrimaryKey str
Mission PrimaryKey Mission
CurrentLocation Required Country
EmptyWeight Required Decimal
MaximumWeight Required Decimal
YearConstructed Optional date
ConstructionCost Optional Decimal
CanDockAt Set SpaceStation
Payloads Set Payload
Crew
Name Required unicode
DateOfBirth Required date
CountryOfOrigin Required Country
Profession Optional str
Payload
PayloadID PrimaryKey int
Weight Required Decimal
HostSpaceship Required Spacecraft
LocatedAtStation Optional SpaceStation
MaximumTravelDuration Optional datetime
Mission
MissionID PrimaryKey int
Objective Required str
LaunchDate Optional date
Budget Required Decimal
Spacecraft Set Spacecraft
Destinations Set Destination
Cargo
EmbargoedIn Set Country
Country
Name PrimaryKey unicode
SpacecraftStock Set Spacecraft
SpaceStations Set SpaceStation
EmbargoedGoods Set Cargo
Astronauts Set Crew
SpaceStation
DockableSpacecraft Set Spacecraft
CountryOfOwnership Set Country
SuppliesAvailable Set Payload
Destination
ICRFx PrimaryKey Decimal
ICRFy PrimaryKey Decimal
ICRFz PrimaryKey Decimal
ICRFYear PrimaryKey date
ArrivalTime PrimaryKey datetime
Mission Required Mission
NextDestination Required Destination
PreviousDestination Optional Destination
OrbitalBody
Name Required LongUnicode
Mass Required Decimal
OrbitEccentricity Required Decimal
OrbitSemimajorAxis Required Decimal
OrbitInclination Required Decimal
OrbitLongitude Required Decimal
OrbitPeriapsis Required Decimal
OrbitEpochAnomaly Required Decimal

Model Overview

I've decided to model a Space Mission from the perspective of an organization like NASA that plans, prepares, and executes science missions in outer space. My model, however, does not assume a particular country of origin, so I've tried to keep international solutions to space travel in mind, instead of just relying on the US Space Shuttle program.

I began my model with the concept of a Mission. The Mission is an overall objective or goal that our space travel will aim to accomplish, and as such, it is the core concept of my model. A mission has two major components: an objective, and resources used to accomplish that objective. Most objectives have a destination in space, generally at the location of some celestial body. Resources come in a wide variety of types, but it is generally assumed that a mission will require some type of transportation from the Earth to a destination, and some type of payload to deliver to or gather information from that destination. As such, a Mission can be regarded as an entity that contains some number of resources and one or more destinations to bring those resources.

In addition to the central Mission concept, there are several other collections of information that, while tangential to the Mission, are important to the company attempting to launch the Mission. These include information about practical aspects such as budgets and locations of various spacecraft and payloads, alongside enumerations of what types of spacecraft we can use and what types of scientific instruments we can deploy as payloads. Some amount of planetary or other destination-local information is also maintained.

Entity Types

Mission

The central entity representing a Space Mission.

- MissionID: A unique, auto-incrementing, non-negative, non-null integer representing the number of missions our company has attempted. This also servers as a primary key for Missions.
- Objective: A string describing the mission's goals. Must be non-null.
- LaunchDate: The date the first spacecraft for the mission is scheduled to leave Earth.
- Budget: A Decimal holding the amount of money available to fund the mission, in USD. Because this value represents money, it must be non-null.
- Spacecraft: A set of space-faring vehicles available to transport payloads for the mission. There must be at least one spacecraft for each Mission.
- Destinations: A set of locations to which mission payloads must be delivered. There must be at least one destination for each Mission.

MissionID was created as a primary key because it's reasonable to assume that there might be multiple missions with the same combination of objective and launch date. It also makes sense for our company to maintain an archival record of which missions happened in which order.

Spacecraft

An abstract representation of a vehicle used to transport a payload from the Earth to a destination in outer space.

- Name: The name of the spacecraft. Must be a string, unique, and non-null. Part of the Primary Key.
- Mission: The Mission to which this spacecraft is assigned. The other part of the Primary Key.
- CurrentLocation: The country in which the spacecraft is currently located. If this is not the same as the cargo location we'll need to pay shipping costs, and if it's not the same as the crew's country we'll need to pay for travel. Must be non-null.
- EmptyWeight: How much the spacecraft weighs while empty. Must be non-null, and non-negative. This

weight plus the weight of the payloads, must not be larger than the maximum weight of the spacecraft.
- MaximumWeight: The maximum amount of weight the spacecraft can hold and still escape Earth's atmosphere. This value must be non-null, non-negative, and larger than the empty weight of the spacecraft plus the weight of all the payloads.
- YearConstructed: A date representing when construction was completed on the spacecraft. This value is optional, but if specified it must be a valid date, and the date must be before the mission launch date for this spacecraft.
- ConstructionCost: This is the amount of money in USD that must be funded to construct the spacecraft. If YearConstructed is a date in the future, this value must be non-null and non-negative, but if YearConstructed is in the past this value must be null.
- Payloads: A set of all the payloads to be carried by this spacecraft. The combined weight of all payloads must be less than the maximum weight of the spacecraft.

The primary key for spacecraft was chosen as Name and Mission since it is assumed we will never have the same spacecraft make two trips on one mission, and spacecraft names are generally unique.

Space Shuttle

An instance of Spacecraft, this was the primary vehicle used to transport US payloads into space.
- ExternalTankType: The type of external tank used to launch the space shuttle. Must be one of "ET", "LWT", or "SWLT".

This inherits a primary key from Spacecraft.

Soyuz

An instance of Spacecraft, this was a popular design for Russian spacecraft.
- VariantType: A string describing which variant of Soyuz spacecraft this is. Must be one of the valid Soyuz types. (See https://goo.gl/Iofyau for reference.)

Inherits a primary key from Spacecraft.

Shenzhou

An instance of Spacecraft, this vehicle is the result of a trade agreement between Russia and China. It is similar in design to the Soyuz craft, but manufactured by China for their space program.

- CraftID: All Shenzhou craft are numbered according to their production date.

This entity inherits a primary key from Spacecraft.

Payload

An abstract representation of the people and objects that ride in a Spacecraft to the Mission destination.

- PayloadID: A unique, non-null, non-negative integer representing each payload entity. All cargo and crew will be given this ID number as they are assigned to spacecraft. This forms the primary key for Payloads.
- Weight: The total weight in kilograms of all items constituting the payload. Must be non-null and non-negative.
- HostSpaceship: The spacecraft to which this payload is assigned.
- LocatedAtStation: If applicable, this is the location in space at which the payload is stored.
- MaximumTravelDuration: The total amount of time this payload may travel or remain in space. Used to track things like decomposition time of biological matter and health standards for human crew. If this value is null, it is assumed that the payload may remain in outer space indefinitely.

Payloads primary keys are specialized in their children, this entity is only an abstract classification of those entities.

Cargo

An instance of Payload, this data represents all of the materials and instruments required to complete the mission.

- EmbargoedIn: A set of countries where this cargo is embargoed. Cargo which is embargoed in a specific country may not be shipped through that country to load onto a spacecraft, and it may not be stored on a space station owned by that country.

The primary key for cargo is inherited from Payload.

Crew

An instance of Payload, this data represents the personal information about any humans who will be traveling to the Mission Destination.

- Name: The full name of each crew member. Must be non-null.
- DateOfBirth: A date representing when the crew member was born.

This must be specified in the Western Calendar.
- CountryOfOrigin: A country entity showing where this astronaut is from.
- Profession: A string describing what profession or service the astronaut is filling on this Mission. This attribute may be multivalued.

The primary key for crew is inherited from Payload.

Destination

An abstraction representing the location in space where the payload is to be delivered by the spacecraft.
- ICRFx: See Below
- ICRFy: See Below
- ICRFz: See Below
- IRCFYear: See Below

  These four entries hold the International Celestial Reference Frame coordinates of the destination in space at the estimated time of arrival. (See https://goo.gl/mceJ4T for reference.) The x, y, and z values must be valid Decimals, and the year value must be a valid date in a given year. These values must be non-null, and are all parts of the primary key of a Destination. These also form a composite Coordinates attribute.
- ArrivalTime: The planned arrival time of the spacecraft to the destination. Must be a valid date in the future. Combined with the ICRF coordinates, this forms the primary key for destinations.
- Mission: The mission to which this destination belongs. Must be non-null.
- NextDestination: If a number of destinations must be visited after this destination, the next destination immediately after this one is stored here.
- PreviousDestination: If a number of destinations must be visited in order, this is the destination required immediately before this one.

The ICRF values and arrival time were chosen as primary keys for Destination because they uniquely specify a time and place in space.

OrbitalBody

A type of destination representing a orbiting body.
- Name: The IAU name of the body. Must be non-null.
- Mass: The mass of the body. Must be non-null.
- OrbitEccentricity:
- OrbitSemimajorAxis:
- OrbitInclination:
- OrbitLongitude:
- OrbitPeriapsis:

- OrbitEpochAnomaly:
  The six Keplerian elements used to describe orbits of bodies in space. These six elements, when combined, form a description of the complete orbit of a celestial body when viewed from a given reference frame. These must all be valid decimals. (See https://goo.gl/tw63Wz for reference.) These combine to form a composite Orbit attribute.

OrbitalBody inherits a primary key from Destination.

Space Station

A type of destination representing a semipermanent human habitation in outer space.

- DockableSpacecraft: A set describing which types of spacecraft may dock at this space station. If we need to dock at this space station, then the spacecraft assigned to this destination must be included in this set.
- Country: The country which owns this spacecraft. In some cases, such as the ISS, there may be multiple countries which own the station, or parts thereof. Cargo embargoed in a specific country may not be stored on a space station owned by that country.
- SuppliesAvailable: The various types of payload available at this space station for pickup by spacecraft assigned to the mission.

Space Stations inherit a primary key from Destination.

Country

An entity representing a country on Earth. This entity may represent any country, not just countries that are interested parties in the mission.

- Name: The full name of the country, kept as up-to-date as possible. This must be a unique, non-null string, and forms the primary key for Countries.
- SpacecraftStock: The set of spacecraft harbored in this country available for Mission use.
- SpaceStations: The set of space stations this country owns, or partially owns.
- EmbargoedGoods: The set of payloads that may not be shipped through this country, or stored on their space stations.
- Astronauts: The set of crew for this mission which originate from this country.

The primary key for countries is chosen to be names as they are assumed to be unique.

Relationship Types

Mission : Spacecraft

Exactly one Mission is related to many spacecraft. This relationship is one-to-many since it may be possible to have multiple spacecraft required for any one Mission, but since spacecraft do not have ownership over Missions a separate Mission entity must be created if a spacecraft is to serve two separate missions. Participation must be total for each spacecraft, since we do not care to store information about spacecraft we are not considering for a Mission.

Mission : Destination

Similar to the Mission : Spacecraft relationship above, a single Mission is related to many Destinations. A Mission may require multiple destinations, but all destinations required for a particular mission must be owned by the singular Mission entity representing that mission. Participation must be total on the side of Destination since we should not keep track of destinations we are not considering for a Mission.

Spacecraft : Country

This relationship represents which country a spacecraft is currently stored in. Many spacecraft may be stored in a particular country, but a spacecraft can only be in one location at a time, so the relationship is many-to-one. Also, participation is total on the side of Spacecraft, since every craft must have a location.

Spacecraft : SpaceStation

This relationship represents which spacecraft can dock at which space stations. Since it is feasible to have spacecraft that can dock at several stations, and stations that can support docking of several spacecraft, this relationship is many-to-many. Neither side has total participation since it is conceivable to have a craft that cannot dock at any space stations or a station that cannot dock any of our current craft.

Spacecraft : Payload

A spacecraft contains many payload instances. A spacecraft may hold many

payloads, but a payload can only be physically located on one spacecraft at a time, so the relationship is many-to-one. There is total participation on the side of Payloads, since each payload must be assigned to a spacecraft in order to reach its destination.

Payload : SpaceStation

Many payloads could be stored on a space station, but there is no requirement for such. This relationship is therefore many to zero-or-one.

SpaceStation : Country

Each space station may be owned by many countries, and a country may own multiple space stations. Therefore, there exists a many-to-many relationship describing the ownership of countries and their space stations.

Country : Cargo

Each country may have a set of goods embargoed in that country, and a particular good may be embargoed in many countries. This is a many-to-many relationship describing that relationship.

Country : Crew

Each country has a number of astronauts, but those astronauts may only represent a single country at a time. Therefore this is a many-to-one relationship between countries and astronauts. This relationship is total on the part of astronauts, since each astronaut must have a country of origin.

Destination : Destination

All destinations must be specified in a particular order. A spacecraft will always have a terminating and originating destination (generally the Earth), so this is a one-to-one reflexive relationship with total participation.

Spacecraft : Payload => Cargo : Country : SpaceStations

There's a ternary relationship here describing which combinations of Spacecraft and Cargo may store or transport supplies to which SpaceStations, dependent on their owning Country attribute.

Spacecraft => SpaceShuttle

SpaceShuttle inherits from Spacecraft. The inheritance is disjoint, since if our particular spacecraft is a space shuttle it may not be any other derived type of spacecraft. SpaceShuttle is a weak entity.

Spacecraft => Soyuz

Soyuz inherits from Spacecraft. The inheritance is disjoint, since if our particular spacecraft is a Soyuz craft it may not be any other derived type of spacecraft. Soyuz is a weak entity.

Spacecraft => Shenzhou

Shenzhou inherits from Spacecraft. The inheritance is disjoint, since if our particular spacecraft is a Shenzhou it may not be any other derived type of spacecraft. Shenzhou is a weak entity.

Destination => OrbitalBody

OrbitalBody inherits from Destination. This makes OrbitalBody a weak entity that relies on Destination for a primary key. This inheritance is overlapping, as it's possible for an entity to be both a SpaceStation and an OrbitalBody.

Destination => SpaceStation

Similar to OrbitalBody, SpaceStation inherits from Destination and is a weak entity. This inheritance is overlapping, as it's possible for an entity to be both a SpaceStation and an OrbitalBody.

Payload => Crew

Crew inherits from Payload, and is a weak entity.

Payload => Cargo

Cargo inherits from Payload, and is a weak entity.

Alpha's Contribution

1. A many-one relationship type with total participation on the one side

   This is modeled in many places, but a particular example includes the relationship between Payload and Spacecraft. Many payloads may be contained on a single spacecraft, but a single payload may not be stored on several spacecraft, and every payload must be assigned to a spacecraft in order to reach the provided destinations.

2. A many-many relationship type

   An example of this is modeled in the relationship between Country and SpaceStation. A country can have ownership of many space stations, and each space station may be jointly owned by several countries.

3. A ternary relationship *

   This is modeled in the relationship between Spacecraft, Cargo, and SpaceStation via the intermediary entity Country. This relationship describes which Spacecraft and Cargo pairs can dock at a particular SpaceStation. (* Note that Pony had no way to show a ternary relationship, so I've included the intermediary entity to represent this relationship.)

4. A disjoint inheritance hierarchy

   This is modeled in by the Spacecraft inheritance hierarchy. A spacecraft can be either a SpaceShuttle, a Soyuz, or a Shenzhou, but no spacecraft could be any combination of the three.

5. A composite attribute *

   There are two examples of this in my model, one of which is the International Celestial Reference Frame Coordinates attribute of a Destination. The coordinates attribute is specified with four sub-attributes, namely the x, y, z, and year attributes. (* Note that Pony had no way to model compound attributes, so the diagram only shows the sub-attributes.)

6. A domain constraint

   All of the shown attributes should have domain constraints, a particular example is the use of a Unicode type on Crew name, allowing us to more accurately represent the wide variety of names in astronauts.

7. A multivalued attribute *

   A crew member may have multiple values for the Profession attribute, since they may be filling multiple roles at one time. (* Note that Pony had no way to represent multivalued attributes, so this does not show up in the diagram provided.)

Beta's Contribution

1. A one-one relationship type with total participation on at least one side

   This is modeled in the reflexive relationship on Destination. Each destination may have only one previous destination and one next destination. Every destination will have a previous destination (starting from Earth), meaning that this participation is total on at least one side.

2. A one-many relationship type with partial participation on both sides

   This is modeled by the relationship between Payload and SpaceStation. Each payload may be stored at a space station, but many payloads could be stored on a single space station at one time. However, there is no requirement for a payload to be on a space station or for a space station to have any payloads.

3. A relationship with an attribute(s) *

   The relationship between a Spacecraft and its associated Payloads contains an attribute, namely the combined weight of all payloads. (* Note that Pony had no way to assign an attribute to this relation, so it isn't shown in the diagram.)

4. An overlapping inheritance hierarchy

   This is modeled in the Destination inheritance hierarchy, since it is feasible for an object to be both a space station and an orbital body. (For example, the ISS is both a space station and in a stable orbit around the Earth.)

5. A weak-entity type *

   Crew is an example of a weak-entity, as they could be identified by a primary key composing of Name, DOB, and Country, completed by the PayloadID assigned to them. (* Note that Pony requires all primary keys to be in the parent Entity Type, so I've represented crew with just the PayloadID primary key in the diagram.)

6. A reflexive relationship type

   Destination has a reflexive relationship in order to specify the order in which each destination must be travelled to.

7. A check (integrity constraint on an attribute). *

   The MaximumWeight contains a check that the EmptyWeight and combined Payload weights do not exceed the value specified. (* Note, Pony does not allow me to specify the checks in the diagram.)

Physical Schema

As generated by PonyORM

```
CREATE TABLE `country` (
  `name` VARCHAR(255) PRIMARY KEY
);

CREATE TABLE `mission` (
  `missionid` INTEGER PRIMARY KEY AUTO_INCREMENT,
  `objective` VARCHAR(255) NOT NULL,
  `launchdate` DATE,
  `budget` DECIMAL(12, 2) NOT NULL
);

CREATE TABLE `destination` (
  `icrfx` DECIMAL(12, 2) NOT NULL,
  `icrfy` DECIMAL(12, 2) NOT NULL,
  `icrfz` DECIMAL(12, 2) NOT NULL,
  `icrfyear` DATE NOT NULL,
  `arrivaltime` DATETIME NOT NULL,
  `mission` INTEGER NOT NULL,
  `nextdestination_icrfx` DECIMAL(12, 2) NOT NULL,
  `nextdestination_icrfy` DECIMAL(12, 2) NOT NULL,
  `nextdestination_icrfz` DECIMAL(12, 2) NOT NULL,
  `nextdestination_icrfyear` DATE NOT NULL,
  `nextdestination_arrivaltime` DATETIME NOT NULL,
  `classtype` VARCHAR(255) NOT NULL,
  `name` LONGTEXT,
  `mass` DECIMAL(12, 2),
  `orbiteccentricity` DECIMAL(12, 2),
  `orbitsemimajoraxis` DECIMAL(12, 2),
  `orbitinclination` DECIMAL(12, 2),
  `orbitlongitude` DECIMAL(12, 2),
  `orbitperiapsis` DECIMAL(12, 2),
  `orbitepochanomaly` DECIMAL(12, 2),
```

```
  PRIMARY KEY (`icrfx`, `icrfy`, `icrfz`, `icrfyear`, `arrivaltime`)
);

CREATE INDEX `idx_destination__mission` ON `destination` (`mission`);

CREATE INDEX
`idx_destination__nextdestination_icrfx_nextdestination_icrfy_nex` ON
`destination` (`nextdestination_icrfx`, `nextdestination_icrfy`,
`nextdestination_icrfz`, `nextdestination_icrfyear`,
`nextdestination_arrivaltime`);

ALTER TABLE `destination` ADD CONSTRAINT `fk_destination__mission` FOREIGN
KEY (`mission`) REFERENCES `mission` (`missionid`);

CREATE TABLE `country_spacestation` (
  `country` VARCHAR(255) NOT NULL,
  `spacestation_icrfx` DECIMAL(12, 2) NOT NULL,
  `spacestation_icrfy` DECIMAL(12, 2) NOT NULL,
  `spacestation_icrfz` DECIMAL(12, 2) NOT NULL,
  `spacestation_icrfyear` DATE NOT NULL,
  `spacestation_arrivaltime` DATETIME NOT NULL,
  PRIMARY KEY (`country`, `spacestation_icrfx`, `spacestation_icrfy`,
`spacestation_icrfz`, `spacestation_icrfyear`, `spacestation_arrivaltime`)
);

CREATE INDEX `idx_country_spacestation` ON `country_spacestation`
(`spacestation_icrfx`, `spacestation_icrfy`, `spacestation_icrfz`,
`spacestation_icrfyear`, `spacestation_arrivaltime`);

ALTER TABLE `country_spacestation` ADD CONSTRAINT
`fk_country_spacestation__country` FOREIGN KEY (`country`) REFERENCES
`country` (`name`);

ALTER TABLE `country_spacestation` ADD CONSTRAINT
`fk_country_spacestation__spacestation_icrfx__spacestation_icrfy_` FOREIGN
KEY (`spacestation_icrfx`, `spacestation_icrfy`, `spacestation_icrfz`,
`spacestation_icrfyear`, `spacestation_arrivaltime`) REFERENCES `destination`
(`icrfx`, `icrfy`, `icrfz`, `icrfyear`, `arrivaltime`);

CREATE TABLE `spacecraft` (
  `name` VARCHAR(255) NOT NULL,
```

```
  `mission` INTEGER NOT NULL,
  `currentlocation` VARCHAR(255) NOT NULL,
  `emptyweight` DECIMAL(12, 2) NOT NULL,
  `maximumweight` DECIMAL(12, 2) NOT NULL,
  `yearconstructed` DATE,
  `constructioncost` DECIMAL(12, 2),
  `classtype` VARCHAR(255) NOT NULL,
  `craftid` INTEGER,
  `varianttype` VARCHAR(255),
  `externaltanktype` VARCHAR(255),
  PRIMARY KEY (`name`, `mission`)
);

CREATE INDEX `idx_spacecraft__currentlocation` ON `spacecraft`
(`currentlocation`);

CREATE INDEX `idx_spacecraft__mission` ON `spacecraft` (`mission`);

ALTER TABLE `spacecraft` ADD CONSTRAINT `fk_spacecraft__currentlocation`
FOREIGN KEY (`currentlocation`) REFERENCES `country` (`name`);

ALTER TABLE `spacecraft` ADD CONSTRAINT `fk_spacecraft__mission` FOREIGN KEY
(`mission`) REFERENCES `mission` (`missionid`);

CREATE TABLE `payload` (
  `payloadid` INTEGER PRIMARY KEY AUTO_INCREMENT,
  `weight` DECIMAL(12, 2) NOT NULL,
  `hostspaceship_name` VARCHAR(255) NOT NULL,
  `hostspaceship_mission` INTEGER NOT NULL,
  `locatedatstation_icrfx` DECIMAL(12, 2),
  `locatedatstation_icrfy` DECIMAL(12, 2),
  `locatedatstation_icrfz` DECIMAL(12, 2),
  `locatedatstation_icrfyear` DATE,
  `locatedatstation_arrivaltime` DATETIME,
  `maximumtravelduration` DATETIME,
  `classtype` VARCHAR(255) NOT NULL,
  `name` VARCHAR(255),
  `dateofbirth` DATE,
```

```
  `countryoforigin` VARCHAR(255),
  `profession` VARCHAR(255)
);

CREATE INDEX `idx_payload__countryoforigin` ON `payload` (`countryoforigin`);

CREATE INDEX `idx_payload__hostspaceship_name_hostspaceship_mission` ON
`payload` (`hostspaceship_name`, `hostspaceship_mission`);

CREATE INDEX
`idx_payload__locatedatstation_icrfx_locatedatstation_icrfy_locat` ON
`payload` (`locatedatstation_icrfx`, `locatedatstation_icrfy`,
`locatedatstation_icrfz`, `locatedatstation_icrfyear`,
`locatedatstation_arrivaltime`);

ALTER TABLE `payload` ADD CONSTRAINT `fk_payload__countryoforigin` FOREIGN
KEY (`countryoforigin`) REFERENCES `country` (`name`);

ALTER TABLE `payload` ADD CONSTRAINT
`fk_payload__hostspaceship_name__hostspaceship_mission` FOREIGN KEY
(`hostspaceship_name`, `hostspaceship_mission`) REFERENCES `spacecraft`
(`name`, `mission`);

ALTER TABLE `payload` ADD CONSTRAINT
`fk_payload__locatedatstation_icrfx__locatedatstation_icrfy__loca` FOREIGN
KEY (`locatedatstation_icrfx`, `locatedatstation_icrfy`,
`locatedatstation_icrfz`, `locatedatstation_icrfyear`,
`locatedatstation_arrivaltime`) REFERENCES `destination` (`icrfx`, `icrfy`,
`icrfz`, `icrfyear`, `arrivaltime`);

CREATE TABLE `cargo_country` (
  `cargo` INTEGER NOT NULL,
  `country` VARCHAR(255) NOT NULL,
  PRIMARY KEY (`cargo`, `country`)
);

CREATE INDEX `idx_cargo_country` ON `cargo_country` (`country`);

ALTER TABLE `cargo_country` ADD CONSTRAINT `fk_cargo_country__cargo` FOREIGN
KEY (`cargo`) REFERENCES `payload` (`payloadid`);
```

```
ALTER TABLE `cargo_country` ADD CONSTRAINT `fk_cargo_country__country`
FOREIGN KEY (`country`) REFERENCES `country` (`name`);


CREATE TABLE `spacestation_spacecraft` (
  `spacestation_icrfx` DECIMAL(12, 2) NOT NULL,
  `spacestation_icrfy` DECIMAL(12, 2) NOT NULL,
  `spacestation_icrfz` DECIMAL(12, 2) NOT NULL,
  `spacestation_icrfyear` DATE NOT NULL,
  `spacestation_arrivaltime` DATETIME NOT NULL,
  `spacecraft_name` VARCHAR(255) NOT NULL,
  `spacecraft_mission` INTEGER NOT NULL,
  PRIMARY KEY (`spacestation_icrfx`, `spacestation_icrfy`,
`spacestation_icrfz`, `spacestation_icrfyear`, `spacestation_arrivaltime`,
`spacecraft_name`, `spacecraft_mission`)
);


CREATE INDEX `idx_spacestation_spacecraft` ON `spacestation_spacecraft`
(`spacecraft_name`, `spacecraft_mission`);


ALTER TABLE `spacestation_spacecraft` ADD CONSTRAINT
`fk_spacestation_spacecraft__spacecraft_name__spacecraft_mission` FOREIGN KEY
(`spacecraft_name`, `spacecraft_mission`) REFERENCES `spacecraft` (`name`,
`mission`);


ALTER TABLE `spacestation_spacecraft` ADD CONSTRAINT
`fk_spacestation_spacecraft__spacestation_icrfx__spacestation_icr` FOREIGN
KEY (`spacestation_icrfx`, `spacestation_icrfy`, `spacestation_icrfz`,
`spacestation_icrfyear`, `spacestation_arrivaltime`) REFERENCES `destination`
(`icrfx`, `icrfy`, `icrfz`, `icrfyear`, `arrivaltime`)
```