



Projektarbeit

Flottenverwaltung EVE Online

Inhaltsverzeichnis

1. Problemstellung	4
1.1 Formulierung der Aufgabenstellung	4
1.2 Ziele.....	4
1.3 Randbedingung.....	4
2. Use Cases	5
2.1 Übersicht.....	5
2.2 Admiral, Captain - Schiff ausrüsten:.....	5
2.3 Admiral, Commander, Captain - Schiff verwalten:.....	5
2.4 Admiral - Flotten formatieren:	5
2.5 Admiral - Destinationen verwalten:.....	5
2.6 Commander - Division formatieren	5
2.7 Admiral - Flotten verwalten:	5
2.8 Admiral, Commander - Division verwalten:.....	5
2.9 Admiral, Captain, Crew - Fracht verwalten:.....	6
3. Grobes ERD Design.....	7
3.1 Übersicht.....	7
3.2 SpaceshipRole:	7
3.3 Shipequipment:.....	7
3.4 CrewRole:.....	7
3.5 Cargohold:.....	7
3.6 Spaceship:.....	7
3.7 Crew:.....	8
3.8 Ammunitionhold:.....	8
3.9 Captain:	8
3.10 Division:	8
3.11 Commander:.....	8
3.12 Destination:.....	8
3.13 Fleet:.....	8
3.14 Admiral:	8
4. Detailliertes Design.....	9
4.1 tspaceship:	9
4.2 tshipequipment:	9
4.3 tspaceship_shipequipment:.....	9
4.4 tcrew_spaceship:.....	9
4.5 tcrew:	9
4.6 tcrewrole:.....	9
4.7 tdivison:	9

4.8	tfleet:	10
4.9	tdest_moon:	10
4.10	tdestination:	10
4.11	tmoon:	10
4.12	tspaceship_ammunition:	10
4.13	tammunitionhold:	10
4.14	tspaceship_cargo:	10
4.15	tcargohold:	10
4.16	tspaceshiprole:	10
5.	SQL-Einstellungs-Script	11
6.	Tests	19

1. Problemstellung

1.1 Formulierung der Aufgabenstellung

Zur Aufgabenstellung gehören Flottenzusammenstellung, Personal, Schiffsausrüstung, Bewaffnung, Aufstellung, Fracht etc. Die jeweiligen Vorgesetzten werden Zugriff auf die Daten ihrer Zuständigkeit haben. Der Admiral muss die Flottenzusammensetzung anpassen können sowie jegliche Informationen über diese erhalten zu können. Es sollt möglich sein, verschiedene Divisionen zu bilden und den Commander dazu zu bestimmen sowie die Schiffe dieser Division zuzuordnen. Dinge wie Fracht und Ausrüstung sollten anpassbar sein.

- Jedes Schiff kann separat ausgerüstet und bemannt werden.
- Alle Schiffe werden einer Division zugeordnet.
- Schiff werden nach Klasse eingeordnet.
- Der Admiral hat kompletten Zugriff auf alle Daten und die Verwaltung aller Schiffe und Divisionen.
- Der Commander hat Zugriff auf alle Daten von der Division sowie deren Verwaltung
- Der Captain hat Zugriff auf alle Daten des Schiffes sowie dessen Verwaltung

1.2 Ziele

Das Ziel ist es eine Datenbank mit welcher man alle Daten über ein Flotte abspeichern kann.

1.3 Randbedingung

Abgabe: 14.12.2021

Sprache: Englisch

Flottengrösse: ca. 5 Schiffe

2. Use Cases

2.1 Übersicht



2.2 Admiral, Captain - Schiff ausrüsten:

Der Captain kann eine Umrüstung der Schiffs Ausrüstung anfordern. Diese wird dann von der Crew ausgeführt. Die Ersatz Ausrüstung wird im Frachtraum gelagert

2.3 Admiral, Commander, Captain - Schiff verwalten:

Der Captain verwaltet das Schiff bedeutet er erfasst den Stand des Schiffes so wie deren Lage. Der Commander hat zugriff auf sämtliche Schiffs daten und kann gegebenen falls Änderungen vornehmen

2.4 Admiral - Flotten formatieren:

Der Admiral hat die Möglichkeit die Flotte neu einzuteilen und Schiffe zwischen Division zu verschieben. Er kann ebenfalls die Besatzung Zuweisung ändern.

2.5 Admiral - Destinationen verwalten:

Der Admiral hat die Möglichkeit die Destination zu bestimmen sowie den Aufenthaltsort aller Schiffe und Divisionen.

2.6 Commander - Division formatieren

Der Commander hat die Möglichkeit die Division zu formatieren heisst er kann die Schiffe innerhalb einer Division anders zuordnen und der Formation ändern.

2.7 Admiral - Flotten verwalten:

Der Admiral hat die volle Kontrolle über die Flotte er kann Divisionen ändern. Die Schiffe bearbeiten und neu zuordnen sowie generelle Änderungen vornehmen. Ebenfalls hat er vollen Einblick auf alle Daten jeglicher Schiffe.

2.8 Admiral, Commander - Division verwalten:

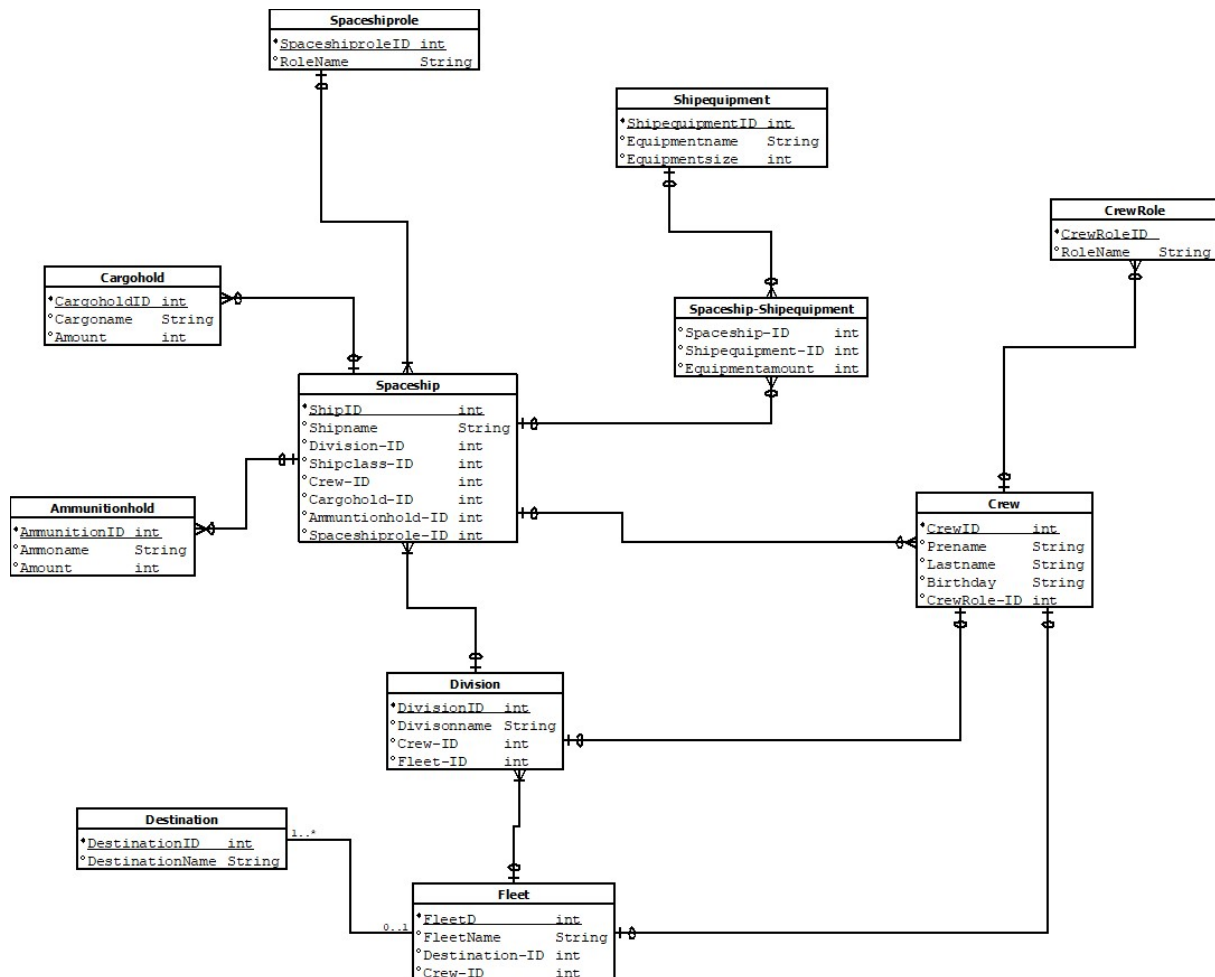
Der Commander hat die volle Kontrolle über die Division, er besitzt die Befugnisse Änderungen an der Division vorzunehmen und auch einzelne Schiffe dieser Division zu kommandieren.

2.9 Admiral, Captain, Crew - Fracht verwalten:

Der Captain besitzt vollen Zugriff auf die Fracht auf dem Schiff, er kann bestimmen was gelagert wird und wo und hat alleinigen Zugriff auf das Wertlager. Die Crew hat zugriff auf die Normalen Lager und kann da Objekte beladen und entladen sowie jederzeit auf dieses Zugreifen.

3. Grobes ERD Design

3.1 Übersicht



3.2 SpaceshipRole:

Die Rolle des Raumschiffes. Jedes Raumschiff ist durch seine Bauweise und Ausrüstung in einer bestimmten Rolle eingeteilt dies ermöglicht es diese effektiv zu benutzen. (Kampfschiff, Transporter etc.)

3.3 Shipequipment:

Die Schiffsausrüstung ist ausschlaggebend für die Möglichkeiten eines Raumschiffes durch eine angepasste Ausrüstung kann ein Raumschiff optimal seine Arbeit ausführen. Es ist möglich die Ausrüstung eines Schiffes anzupassen um es auf verschiedene Gegebenheiten anzupassen. (Bewaffnung, Schilde etc.)

3.4 CrewRole:

Durch die Komplexität eines Raumschiffes und deren Operationen braucht es viel spezialisiertes Personal diese werden in verschiedenen Rollen eingeteilt was unter anderem auch bestimmt zu welchen Teilen des Schiffes sie Zugang haben. (Mechaniker, Soldat etc.)

3.5 Cargohold:

Hier wird so ziemlich alles aus militärischer Ausrüstung gelagert. (Ausrüstung, Materialien, Nahrung etc.)

3.6 Spaceship:

Das Raumschiff ist der Kern der Datenbank die meisten Tabellen stehen damit in Verbindung.

3.7 Crew:

Eine Liste des ganzen Personals welche sich in der Flotte befinden mit all ihren Daten und der Zuordnung eines Schiffes und deren Rolle.

3.8 Ammunitionhold:

In diesem Frachtraum befinden sich Militärische Ausrüstung. Nur bestimmte Personal wie z.B. Soldaten oder der Captain haben Zutritt zu diesem Frachtraum. (Ballistische Munition, Raketen etc.)

3.9 Captain:

Der Kapitän eines Schiffes er hat die Vollmacht über dieses Schiff.

3.10 Division:

Die Divisionen der Flotte. Verschiedenen Divisionen erfüllen unterschiedliche Aufgaben wie z.B. das Ausspähen von Zielorten oder die Versorgung der Flotte. (Versorgungs-Division, Späh-Division etc.)

3.11 Commander:

Der Kommandeur hat das Befehlsrecht über seine Division.

3.12 Destination:

Die Datenbank der Koordinaten für die verschiedenen Sternensysteme etc.

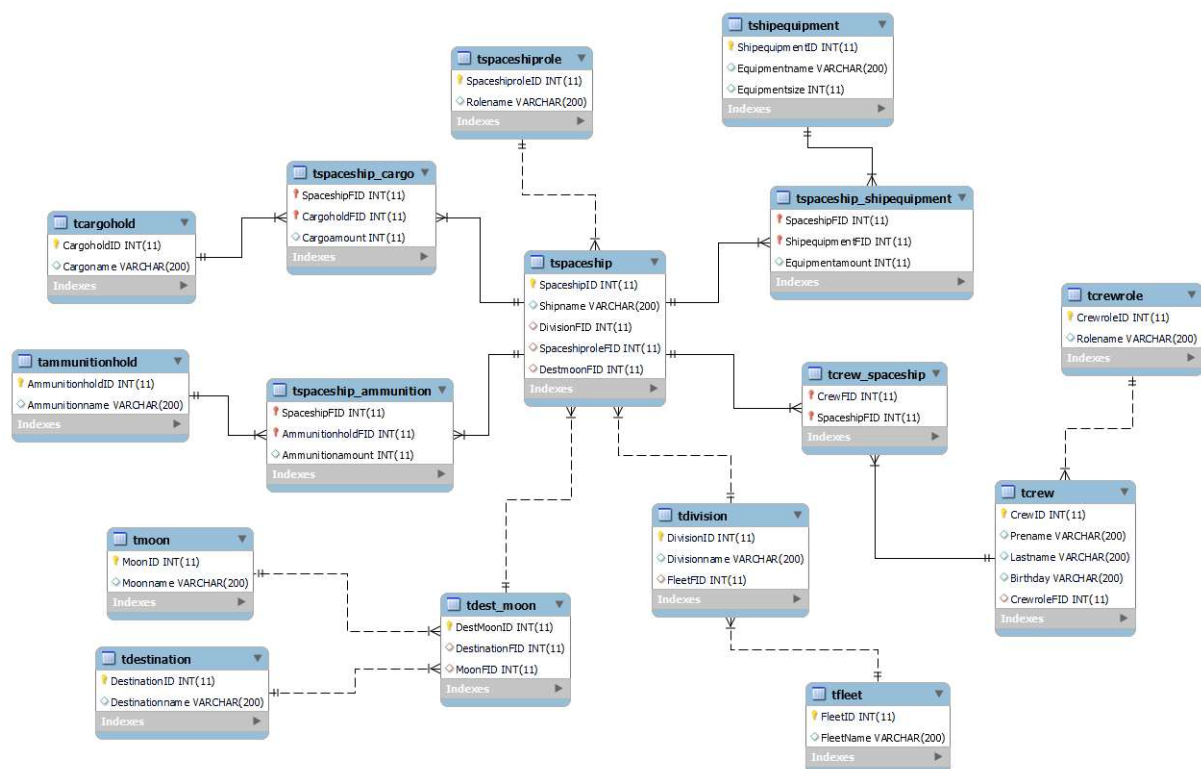
3.13 Fleet:

Die Flotte und deren Identifikation.

3.14 Admiral:

Der Admiral der Flotte er besitzt die komplette Kontrolle über die Flotte.

4. Detailliertes Design



4.1 tspaceship:

Das Raumschiff bildet quasi den Kern der Datenbank. Jedes Raumschiff ist einer Division und dadurch auch einer Flotte zugeordnet. Ebenfalls besitzt es eine Ausrüstung und Bewaffnung, ein Fracht- und Munitionslager sowie eine Crew. Dazu ist es noch in eine Schifffrolle eingeteilt

4.2 tshipeqipment:

Die Schiffsausrüstung sind die Ausgerüsteten Komponenten. Darunter zählen Generatoren, Schilde, Bewaffnung etc. Jede Ausrüstung besitzt eine Bezeichnung sowie eine Grösse.

4.3 tspaceship_shipequipment:

Die Zwischentabelle von `tshipecquipment` und `tspaceship`. Sie enthält die Anzahl der ausgerüsteten Komponenten.

4.4 tcrow_spaceship:

Die Zwischentabelle von tcrew und tspaceship.

4.5 tcrew:

Enthält das Personal der Schiffe mit ihren Persönlichen Daten sowie den Fremdschlüssel zu tcrewrole.

4.6 tcrewrole:

Dies ist die Einteilung des Personals in bestimmte Rollen sei dies der Captain, Admiral oder ein Mechaniker. Enthält den Rollennamen.

4.7 tdivison:

Jede Flotte ist unterteilt in verschiedene Divisionen was sozusagen Gruppierungen von Raumschiffen einer Flotte sind. Jede Division besitzt einen Namen und ist einer Flotte zugeteilt. Ebenfalls ist jeder Division ein Kommandeur zugeteilt.

4.8 tfleet:

Eine Flotte besteht aus mehreren Divisionen und besitzt einen Admiral welche das Oberste Entscheidungsrecht hat.

4.9 tdest_moon:

Diese Tabelle beinhaltet die Fremdschlüssel aus tdestination und tmoon woraus zusammen sich die Position des Raumschiffes zusammensetzt. Z.b. tdestination = Erde + tmoon = Mond. Als befindet sich das Raumschiff im Orbit um die Sonne mit der Erde ist jedoch beim Mond stationiert.

4.10 tdestination:

Hier sind die Planeten des Sonnensystems eingetragen.

4.11 tmoon:

In dieser Tabelle sind alle Monde der Planeten aus tdestination eingetragen.

4.12 tspaceship_ammunition:

Die Zwischentabelle von tammunitionhold und tspaceship. Sie enthält die Menge der jeweiligen Munition.

4.13 tammunitionhold:

Das Munitionslager enthält die Munition für Ballistische Schiffswaffen sowie die Munition für die Handfeuerwaffen

4.14 tspaceship_cargo:

Die Zwischentabelle von tcargohold und tspaceship. Sie enthält die Menge der jeweiligen Fracht.

4.15 tcargohold:

Der Frachtraum lagert alle Möglich Dinge von Erzen bis zu Wasser und Nahrung für die Crew.

4.16 tspaceshiprole:

Jedes Raumschiff ist einer Rolle zugeteilt. Faktoren wie die Grösse sowie die Ausrüstung spielen dabei eine grosse Rolle.

5. SQL-Einstellungs-Script

```
DROP DATABASE IF EXISTS Fleetmanagement;

CREATE DATABASE Fleetmanagement;

USE Fleetmanagement;

/* Create's */

CREATE TABLE tSpaceshiprole (
    SpaceshiproleID INT NOT NULL auto_increment,
    Rolename VARCHAR(200),
    PRIMARY KEY (SpaceshiproleID)
);

CREATE INDEX iRolename_spaceship
ON tSpaceshiprole (Rolename);

CREATE TABLE tShipequipment (
    ShipequipmentID INT NOT NULL auto_increment,
    Equipmentname VARCHAR(200),
    Equipmentsize INT,
    PRIMARY KEY (ShipequipmentID)
);

CREATE INDEX iShipequipment_name_size
ON tShipequipment (Equipmentname, Equipmentsize);

CREATE TABLE tCrewrole (
    CrewroleID INT NOT NULL auto_increment,
    Rolename VARCHAR(200),
    PRIMARY KEY (CrewroleID)
);

CREATE INDEX iCrewrole_name
ON tCrewrole (Rolename);

CREATE TABLE tCrew (
    CrewID INT NOT NULL auto_increment,
    Prenom VARCHAR(200),
    Lastname VARCHAR(200),
    Birthday VARCHAR(200),
    CrewroleFID INT,
    PRIMARY KEY (CrewID),
    FOREIGN KEY (CrewroleFID) REFERENCES tCrewrole(CrewroleID)
);

CREATE INDEX iCrew_name
ON tCrew (Prenom, Lastname);

CREATE TABLE tCargohold (
    CargoholdID INT NOT NULL auto_increment,
    Cargoname VARCHAR(200),
    PRIMARY KEY (CargoholdID)
);

CREATE INDEX iCargohold_name
```

```
ON tCargohold (Cargoname);
```

```
CREATE TABLE tAmmunitionhold (  
    AmmunitionholdID INT NOT NULL auto_increment,  
    Ammunitionname VARCHAR(200),  
    PRIMARY KEY (AmmunitionholdID)  
);
```

```
CREATE INDEX iAmmunitionhold_name  
ON tAmmunitionhold (Ammunitionname);
```

```
CREATE TABLE tMoon (  
    MoonID INT NOT NULL auto_increment,  
    Moonname VARCHAR(200),  
    PRIMARY KEY (MoonID)  
);
```

```
CREATE INDEX iMoon_name  
ON tMoon (Moonname);
```

```
CREATE TABLE tDestination (  
    DestinationID INT NOT NULL auto_increment,  
    Destinationname VARCHAR(200),  
    PRIMARY KEY (DestinationID)  
);
```

```
CREATE INDEX iDestination_name  
ON tDestination (Destinationname);
```

```
CREATE TABLE tFleet (  
    FleetID INT NOT NULL auto_increment,  
    FleetName VARCHAR(200),  
    PRIMARY KEY (FleetID)  
);
```

```
CREATE INDEX iFleet_name  
ON tFleet (FleetName);
```

```
CREATE TABLE tDivision (  
    DivisionID INT NOT NULL auto_increment,  
    Divisionname VARCHAR(200),  
    FleetFID INT,  
    PRIMARY KEY (DivisionID),  
    FOREIGN KEY (FleetFID) REFERENCES tFleet(FleetID)  
);
```

```
CREATE INDEX iDivision_name  
ON tDivision (Divisionname);
```

```
CREATE TABLE tDest_Moon (  
    DestMoonID INT NOT NULL auto_increment,  
    DestinationFID INT,  
    MoonFID INT,  
    PRIMARY KEY (DestMoonID),  
    FOREIGN KEY (DestinationFID) REFERENCES tDestination(DestinationID),  
    FOREIGN KEY (MoonFID) REFERENCES tMoon(MoonID)  
);
```

```
CREATE INDEX iDest_Moon_Foreignkey
```

```
ON tDest_Moon (DestinationFID, MoonFID);
```

```
CREATE TABLE tSpaceship (  
    SpaceshipID INT NOT NULL auto_increment,  
    Shipname VARCHAR(200),  
    DivisionFID INT,  
    SpaceshiproleFID INT,  
    DestmoonFID INT,  
    PRIMARY KEY (SpaceshipID),  
    FOREIGN KEY (DivisionFID) REFERENCES tDivision(DivisionID),  
    FOREIGN KEY (SpaceshiproleFID) REFERENCES tSpaceshiprole(SpaceshiproleID),  
    FOREIGN KEY (DestmoonFID) REFERENCES tDest_Moon(DestmoonID)  
);
```

```
CREATE INDEX iSpaceship_shipname_shiprole  
ON tSpaceship (Shipname, SpaceshiproleFID);
```

```
CREATE INDEX iSpaceship_shipname_destmoon  
ON tSpaceship (Shipname, DestmoonFID);
```

```
CREATE TABLE tSpaceship_Shipequipment (  
    SpaceshipFID INT,  
    ShipequipmentFID INT,  
    Equipmentamount INT,  
    PRIMARY KEY (SpaceshipFID, ShipequipmentFID),  
    FOREIGN KEY (SpaceshipFID) REFERENCES tSpaceship(SpaceshipID),  
    FOREIGN KEY (ShipequipmentFID) REFERENCES tShipequipment(ShipequipmentID)  
);
```

```
CREATE INDEX iSpaceship_Shipequipment_amount  
ON tSpaceship_Shipequipment (Equipmentamount);
```

```
CREATE TABLE tCrew_Spaceship (  
    CrewFID INT,  
    SpaceshipFID INT,  
    PRIMARY KEY (CrewFID, SpaceshipFID),  
    FOREIGN KEY (CrewFID) REFERENCES tCrew(CrewID),  
    FOREIGN KEY (SpaceshipFID) REFERENCES tSpaceship(SpaceshipID)  
);
```

```
CREATE INDEX iCrew_Spaceship_FID  
ON tCrew_Spaceship (SpaceshipFID);
```

```
CREATE TABLE tSpaceship_Cargo (  
    SpaceshipFID INT,  
    CargoholdFID INT,  
    Cargoamount INT,  
    PRIMARY KEY (SpaceshipFID, CargoholdFID),  
    FOREIGN KEY (SpaceshipFID) REFERENCES tSpaceship(SpaceshipID),  
    FOREIGN KEY (CargoholdFID) REFERENCES tCargohold(CargoholdID)  
);
```

```
CREATE INDEX iSpaceship_Cargo_amount  
ON tSpaceship_Cargo (Cargoamount);
```

```
CREATE TABLE tSpaceship_Ammunition (  
    SpaceshipFID INT,  
    AmmunitionholdFID INT,  
    Ammunitionamount INT,
```

```
PRIMARY KEY (SpaceshipFID, AmmunitionholdFID),  
FOREIGN KEY (SpaceshipFID) REFERENCES tSpaceship(SpaceshipID),  
FOREIGN KEY (AmmunitionholdFID) REFERENCES  
tAmmunitionhold(AmmunitionholdID)  
);
```

```
CREATE INDEX iSpaceship_Ammunition_amount  
ON tSpaceship_Ammunition (Ammunitionamount);
```

```
/* Inserts */
```

```
INSERT INTO tSpaceshiprole  
(SpaceshiproleID, Rolename)  
VALUES  
(1, "Gunship"),  
(2, "Scout"),  
(3, "Frigate"),  
(4, "Freighter"),  
(5, "Cruiser"),  
(6, "Destroyer"),  
(7, "Capitalship");
```

```
INSERT INTO tShipequipment  
(ShipequipmentID, Equipmentname, Equipmentsize)  
VALUES  
(1, "Shieldgenerator", 1),  
(2, "Shieldgenerator", 2),  
(3, "Shieldgenerator", 3),  
(4, "Shieldgenerator", 4),  
(5, "Cooler", 1),  
(6, "Cooler", 2),  
(7, "Cooler", 3),  
(8, "Cooler", 4),  
(9, "Power-Plant", 1),  
(10, "Power-Plant", 2),  
(11, "Power-Plant", 3),  
(12, "Power-Plant", 4),  
(13, "Laser-Repeater", 1),  
(14, "Laser-Repeater", 2),  
(15, "Laser-Repeater", 3),  
(17, "Laser-Repeater", 4),  
(18, "Gatling-Gun", 1),  
(19, "Gatling-Gun", 2),  
(20, "Gatling-Gun", 3),  
(21, "Gatling-Gun", 4),  
(22, "Heat-Missile", 1),  
(23, "Heat-Missile", 2),  
(24, "Heat-Missile", 3),  
(25, "Heat-Missile", 4),  
(26, "Flair-Launcher", 1),  
(27, "Flair-Launcher", 2);
```

```
INSERT INTO tCargohold  
(CargoholdID, Cargoname)  
VALUES  
(1, "Iron-ore [kg]"),  
(2, "Copper-ore [kg]"),  
(3, "Lithium-ore [kg]"),  
(4, "MRE-Pack"),
```

```
(5, "Textile [kg]"),  
(6, "Scrap [kg]"),  
(7, "Spacesuit"),  
(8, "Water [l]");
```

```
INSERT INTO tAmmunitionhold  
  (AmmunitionholdID, Ammunitionname)  
VALUES  
  (1, "Ballistic-Ammunition (Size 1)"),  
  (2, "Ballistic-Ammunition (Size 2)"),  
  (3, "Ballistic-Ammunition (Size 3)"),  
  (4, "Ballistic-Ammunition (Size 4)"),  
  (5, "9mm-Magazin"),  
  (6, "5,56x45-Magazin"),  
  (7, "7,62x39-Magazin"),  
  (8, "Flair (Size 1)"),  
  (9, "Flair (Size 2)");
```

```
INSERT INTO tCrewrole  
  (CrewroleID, Rolename)  
VALUES  
  (1, "Admiral"),  
  (2, "Commander"),  
  (3, "Captain"),  
  (4, "Soldier"),  
  (5, "Mechanic"),  
  (6, "Worker");
```

```
INSERT INTO tCrew  
  (CrewID, Prenome, Lastname, Birthday, CrewroleFID)  
VALUES  
  (1, "Luca", "Luu", "23.11.75", 1),  
  (2, "Michael", "Frantzes", "9.1.77", 2),  
  (3, "Donald", "Tate", "30.3.81", 2),  
  (4, "Ryan", "King", "10.12.74", 3),  
  (5, "Roland", "Sherman", "23.11.72", 3),  
  (6, "Nick", "Lauda", "11.5.83", 3),  
  (7, "Nedo", "Lui", "3.7.69", 3);
```

```
INSERT INTO tDestination  
  (DestinationID, Destinationname)  
VALUES  
  (1, "Mercury"),  
  (2, "Venus"),  
  (3, "Earth"),  
  (4, "Mars"),  
  (5, "Jupiter"),  
  (6, "Saturn"),  
  (7, "Uranus"),  
  (8, "Neptun"),  
  (9, "Pluto");
```

```
INSERT INTO tMoon  
  (MoonID, Moonname)  
VALUES  
  (1, "Planetary"),  
  (2, "Moon"),  
  (3, "Phobos"),  
  (4, "Deimos"),  
  (5, "Ganymed"),  
  (6, "Kallisto"),
```



```
(7, "Io"),  
(8, "Europa"),  
(9, "Titan"),  
(10, "Rhea"),  
(11, "Iapetus"),  
(12, "Dione"),  
(13, "Titania"),  
(14, "Oberon"),  
(15, "Umbriel"),  
(16, "Ariel"),  
(17, "Triton"),  
(18, "Proteus"),  
(19, "Nereid"),  
(20, "Charon");
```

```
INSERT INTO tDest_Moon  
(DestmoonID, DestinationFID, MoonFID)  
VALUES  
(1, 1, 1),  
(2, 2, 1),  
(3, 3, 1),  
(4, 3, 2),  
(5, 4, 1),  
(6, 4, 3),  
(7, 4, 4),  
(8, 5, 1),  
(9, 5, 5),  
(10, 5, 6),  
(11, 5, 7),  
(12, 5, 8),  
(13, 6, 1),  
(14, 6, 9),  
(15, 6, 10),  
(16, 6, 11),  
(17, 6, 12),  
(18, 7, 1),  
(19, 7, 13),  
(20, 7, 14),  
(21, 7, 15),  
(22, 7, 16),  
(23, 8, 1),  
(24, 8, 17),  
(25, 8, 18),  
(26, 8, 19),  
(27, 8, 20),  
(28, 9, 1);
```

```
INSERT INTO tFleet  
(FleetID, Fleetname)  
VALUES  
(1, "GuiltySpark");  
-- (x, "x");
```

```
INSERT INTO tDivision  
(DivisionID, Divisionname, FleetFID)  
VALUES  
(1, "Maindivison", 1),  
(2, "Frontdivison", 1);
```

```
INSERT INTO tSpaceship
```

```
(SpaceshipID, Shipname, DivisionFID, SpaceshiproleFID,  
DestmoonFID)
```

```
VALUES
```

```
(1, "Pillar of Autumn", 1, 7, 3),  
(2, "Nexus", 1, 6, 4),  
(3, "Bolt", 2, 5, 4 ),  
(4, "Passasge", 2, 3, 5);
```

```
INSERT INTO tCrew_Spaceship  
(CrewFID, SpaceshipFID)
```

```
VALUES
```

```
(1, 1),  
(2, 1),  
(3, 2),  
(4, 1),  
(5, 2),  
(6, 3),  
(7, 4);
```

```
INSERT INTO tSpaceship_Ammunition  
(SpaceshipFID, AmmunitionholdFID, Ammunitionamount)
```

```
VALUES
```

```
(1, 4, 20000),  
(1, 5, 250),  
(1, 6, 150),  
(1, 7, 150),  
(1, 9, 50),  
(2, 2, 15000),  
(2, 5, 150),  
(2, 6, 100),  
(2, 7, 120),  
(2, 8, 60),  
(3, 2, 12000),  
(3, 5, 200),  
(3, 6, 150),  
(3, 7, 100),  
(3, 8, 40),  
(4, 9, 50);
```

```
INSERT INTO tSpaceship_Cargo  
(SpaceshipFID, CargoholdFID, Cargoamount)
```

```
VALUES
```

```
(1, 1, 15000),  
(1, 2, 6000),  
(1, 3, 3000),  
(1, 4, 300),  
(1, 7, 20000),  
(1, 8, 5000),  
(2, 1, 10000),  
(2, 2, 3000),  
(2, 3, 6000),  
(2, 4, 500),  
(2, 6, 10000),  
(2, 8, 10000),  
(3, 1, 8000),  
(3, 2, 2000),  
(3, 3, 3000),  
(3, 4, 250),  
(3, 7, 8000),  
(3, 8, 9000),  
(4, 1, 35000),  
(4, 2, 9000),
```

```
(4, 3, 5000),  
(4, 4, 200),  
(4, 5, 6000),  
(4, 6, 600),  
(4, 7, 16000),  
(4, 8, 12000);
```

```
INSERT INTO tSpaceship_Shipequipment  
(SpaceshipFID, ShipecquipmentFID, Equipmentamount)  
VALUES  
(1, 4, 2),  
(1, 5, 2),  
(1, 9, 2),  
(1, 13, 4),  
(1, 21, 2),  
(1, 23, 4),  
(1, 27, 1),  
(2, 2, 4),  
(2, 5, 2),  
(2, 9, 2),  
(2, 13, 4),  
(2, 19, 2),  
(2, 22, 4),  
(2, 26, 1),  
(3, 2, 4),  
(3, 6, 3),  
(3, 9, 2),  
(3, 13, 2),  
(3, 19, 4),  
(3, 22, 2),  
(3, 25, 1),  
(4, 2, 2),  
(4, 6, 2),  
(4, 10, 2),  
(4, 14, 2),  
(4, 23, 4),  
(4, 27, 2);
```

6. Tests

```
/* Selects */
```

```
-- Gib den Admiral der Flotte aus -
```

```
SELECT * FROM tCrew JOIN tCrewrole ON tCrewrole.CrewroleID=CrewroleFID WHERE  
Rolename = "Admiral";
```

```
-- Gibt alle Captain der Flotte aus -
```

```
SELECT Prename, Lastname, Shipname FROM tFleet  
JOIN tDivision ON tFleet.FleetID=tDivision.FleetFID  
JOIN tSpaceship ON tDivision.DivisionID=tSpaceship.DivisionFID  
JOIN tCrew_Spaceship ON tSpaceship.SpaceShipID=tCrew_Spaceship.SpaceShipFID  
JOIN tCrew ON tCrew_Spaceship.CrewFID=tCrew.CrewID  
JOIN tCrewrole ON tCrew.CrewroleFID=tCrewrole.CrewroleID  
WHERE Rolename = "Captain";
```

```
-- Gibt die Anzahl an Crewmitglieder pro Schiff aus -
```

```
SELECT COUNT(Prename) FROM tSpaceship  
JOIN tCrew_Spaceship ON tSpaceship.SpaceShipID=tCrew_Spaceship.SpaceShipFID  
JOIN tCrew ON tCrew_Spaceship.CrewFID=tCrew.CrewID  
WHERE SpaceShipID LIKE 1;
```

```
-- Gibt die Anzahl an Schiffe in der Flotte aus welche "Scrap [kg]"[Cargoname]  
geladen haben -
```

```
SELECT COUNT(Cargoname) FROM tSpaceship  
JOIN tSpaceship_Cargo ON tSpaceship.SpaceShipID=tSpaceship_Cargo.SpaceShipFID  
JOIN tCargohold ON tSpaceship_Cargo.CargoholdFID=tCargohold.CargoholdID  
WHERE Cargoname = "Scrap [kg]";
```

```
-- Gibt die Ganze Fracht des Schiffs aus -
```

```
SELECT Cargoname, Cargoamount FROM tSpaceship  
JOIN tSpaceship_Cargo ON tSpaceship.SpaceShipID=tSpaceship_Cargo.SpaceShipFID  
JOIN tCargohold ON tSpaceship_Cargo.CargoholdFID=tCargohold.CargoholdID  
WHERE SpaceShipID LIKE 1;
```

```
-- Gibt die Durchschnittliche Menge "Water [l]"[Cargoname] aus welche pro  
Schiff gelden ist"
```

```
SELECT AVG(Cargoamount) FROM tSpaceship  
JOIN tSpaceship_Cargo ON tSpaceship.SpaceShipID=tSpaceship_Cargo.SpaceShipFID  
JOIN tCargohold ON tSpaceship_Cargo.CargoholdFID=tCargohold.CargoholdID  
WHERE Cargoname = "Water [l]";
```

```
-- Zählt das Gesamtmenge der Fracht des Schiffs -
```

```
SELECT SUM(Cargoamount) FROM tSpaceship  
JOIN tSpaceship_Cargo ON tSpaceship.SpaceShipID=tSpaceship_Cargo.SpaceShipFID  
JOIN tCargohold ON tSpaceship_Cargo.CargoholdFID=tCargohold.CargoholdID  
WHERE SpaceShipID LIKE 1;
```

```
-- Gibt aus welche Munition unter dem gewünschten Wert ist -
```

```
SELECT Ammunitionname, Ammunitionamount FROM tSpaceship
JOIN tSpaceship_Ammunition ON
tSpaceship.SpaceShipID=tSpaceship_Ammunition.SpaceShipFID
JOIN tAmmunitionhold ON
tSpaceship_Ammunition.AmmunitionholdFID=tAmmunitionhold.AmmunitionholdID
WHERE SpaceShipID LIKE 1 AND Equipmentname LIKE "Ballistic-Ammunition%" AND
Ammunitionamount < 20000;
```

```
-- Gibt alle Schiffe der Flotte aus welche sich an dieser Destination befinden
--
```

```
SELECT Shipname FROM tFleet
JOIN tDivision ON tFleet.FleetID=tDivision.FleetFID
JOIN tSpaceship ON tDivision.DivisionID=tSpaceship.DivisionFID
JOIN tDest_Moon ON tSpaceship.DestmoonFID=tDest_Moon.DestMoonID
JOIN tDestination ON tDest_Moon.DestinationFID=tDestination.DestinationID
JOIN tMoon ON tDest_Moon.MoonFID=tMoon.MoonID
WHERE Destinationname = "Earth" AND Moonname = "Moon";
```

```
-- Gibt die Ballistischen Waffen und deren Munitionsmenge für alle Schiffe der
Flotte aus -
```

```
SELECT Shipname, Equipmentname, Equipmentsize, Ammunitionname,
Ammunitionamount FROM tShipequipment
JOIN tSpaceship_Shipequipment ON
tShipequipment.ShipequipmentID=tSpaceship_Shipequipment.ShipequipmentFID
JOIN tSpaceship ON
tSpaceship_Shipequipment.SpaceShipFID=tSpaceship.SpaceShipID
JOIN tSpaceship_Ammunition ON
tSpaceship.SpaceShipID=tSpaceship_Ammunition.SpaceShipFID
JOIN tAmmunitionhold ON
tSpaceship_Ammunition.AmmunitionholdFID=tAmmunitionhold.AmmunitionholdID
WHERE Equipmentname = "Gatling-Gun" AND Ammunitionname LIKE "Ballistic-
Ammunition%" OR Equipmentname = "Flair-Launcher" AND Ammunitionname LIKE
"Flair%"
ORDER BY Shipname DESC, Equipmentname DESC, Equipmentsize DESC;
```

```
-- Gibt die Schiffe aus welche weniger "Iron-Ore"[Cargoname] als das Schiff
"Pillar of Autumn"[Shipname] haben -
```

```
SELECT Shipname, Cargoamount FROM tSpaceship
JOIN tSpaceship_Cargo ON tSpaceship.SpaceShipID=tSpaceship_Cargo.SpaceShipFID
JOIN tCargohold ON tSpaceship_Cargo.CargoholdFID=tCargohold.CargoholdID
WHERE Cargoname = "Iron-ore [kg]" AND Cargoamount <
  (SELECT Cargoamount FROM tSpaceship
  JOIN tSpaceship_Cargo ON
  tSpaceship.SpaceShipID=tSpaceship_Cargo.SpaceShipFID
  JOIN tCargohold ON tSpaceship_Cargo.CargoholdFID=tCargohold.CargoholdID
  WHERE Shipname = "Pillar of Autumn" AND Cargoname = "Iron-ore [kg]" )
```

```
-- Gibt alle Raumschiffe aus welche sich am gleichen Ort wie die
"Nexus"[Shipname] befinden --
```

```
SELECT Shipname FROM tSpaceship
JOIN tDest_Moon ON tSpaceship.DestmoonFID=tDest_Moon.DestMoonID
WHERE DestMoonID =
  (SELECT DestMoonID FROM tSpaceship
  JOIN tDest_Moon ON tSpaceship.DestmoonFID=tDest_Moon.DestMoonID
  WHERE Shipname = "Nexus")
```

-- Gibt alle Raumschiffe aus welchen sich beim gleichen Planeten und den zugehörigen Monde wie die "Pillar of Autumn"[Shipname] befinden --

```
SELECT Shipname, Destinationname, Moonname FROM tSpaceship
JOIN tDest_Moon ON tSpaceship.DestmoonFID=tDest_Moon.DestMoonID
JOIN tDestination ON tDest_Moon.DestinationFID=tDestination.DestinationID
JOIN tMoon ON tDest_Moon.MoonFID=tMoon.MoonID
WHERE DestinationID =
    (SELECT DestinationID FROM tSpaceship
     JOIN tDest_Moon ON tSpaceship.DestmoonFID=tDest_Moon.DestMoonID
     JOIN tDestination ON tDest_Moon.DestinationFID=tDestination.DestinationID
     JOIN tMoon ON tDest_Moon.MoonFID=tMoon.MoonID
     WHERE Shipname = "Pillar of Autumn")
```

-- Gibt alle Raumschiffe aus welche weniger "Iron-ore [kg]" als der Durchschnitt aller Raumschiffe geladen haben --

```
SELECT Shipname, Cargoamount FROM tSpaceship
JOIN tSpaceship_Cargo ON tSpaceship.SpaceShipID=tSpaceship_Cargo.SpaceShipFID
JOIN tCargohold ON tSpaceship_Cargo.CargoholdFID=tCargohold.CargoholdID
WHERE Cargoname = "Iron-ore [kg]" AND Cargoamount <
    (SELECT AVG(Cargoamount) FROM tSpaceship
     JOIN tSpaceship_Cargo ON
tSpaceship.SpaceShipID=tSpaceship_Cargo.SpaceShipFID
     JOIN tCargohold ON tSpaceship_Cargo.CargoholdFID=tCargohold.CargoholdID
     WHERE Shipname LIKE "%" AND Cargoname = "Iron-ore [kg]")
```

-- Gibt alle Crewmitglieder mit all ihren Persönlichen Daten eines jeweiligen Raumschiffs auf --

```
SELECT Prenom, Lastname, Birthday, tCrewrole.rolename FROM tSpaceship
JOIN tCrew_Spaceship ON tSpaceship.SpaceShipID=tCrew_Spaceship.SpaceShipFID
JOIN tCrew ON tCrew_Spaceship.CrewFID=tCrew.CrewID
JOIN tCrewrole ON tCrew.CrewroleFID=tCrewrole.CrewroleID
WHERE Shipname = "Pillar of Autumn" ORDER BY Lastname ASC;
```