

# Database Project: Star Wars

Zachary Meade

17340169



**Trinity College Dublin**

Coláiste na Tríonóide, Baile Átha Cliath

The University of Dublin

## Table of Contents

<b>Description of Database .....</b>	<b>2</b>
<b>Entity Relationship Diagram .....</b>	<b>3</b>
<b>Relational Schema .....</b>	<b>4</b>
<b>Functional Dependency Diagram .....</b>	<b>5</b>
<b>Constraints .....</b>	<b>6</b>
<b>Security .....</b>	<b>6</b>
<b>View Creation .....</b>	<b>7</b>
<b>Triggers .....</b>	<b>8</b>
<b>Appendix .....</b>	<b>9</b>

## Description of Database

I have chosen to implement a database based around the Star Wars cinematic universe. As the Star Wars universe is very complex and ever expanding I have focused my database on the Jedi Order in the third movie, Star Wars III: Attack of the clones. The tables I have used in my database consist of Jedi, Jedi Council, Padawan, Battalion, Clones and Ships.

The first table Jedi, tracks all of the force users who have completed their Jedi training. Each individual Jedi will have an ID which is used as their primary key. The Jedi are also given the option to have a Padawan (Jedi in training). Although a Jedi takes his/her orders from the Jedi council it is possible for a Jedi to exist without a Jedi council. Other personal information is stored about each Jedi such as their first and second, their preferred lightsaber colour and also what race they are.

The second table is Jedi Council. This table track each active member of the Jedi Council. A Jedi's rank in the council is used as the primary key. The table has one foreign key which is Jedi\_ID. The table also tracks how many years a member has been in the council for and their area of expertise, if they have one. The Jedi council cannot exist without a Jedi.

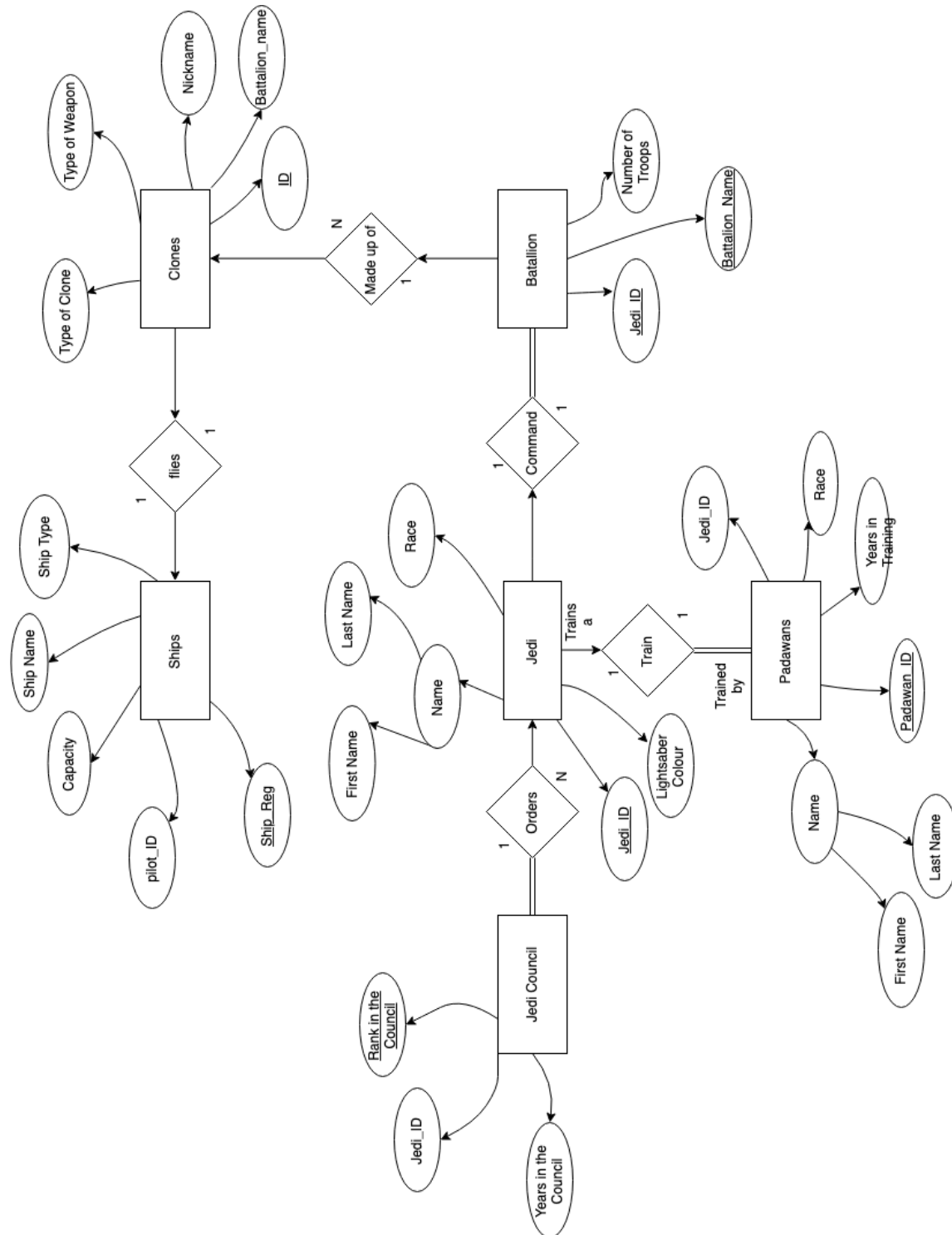
The third table is Padawan. This table keeps track of all the Jedi in training. Each Padawan has a primary key Padawan\_ID. Padawan also has one foreign key which is their master's id (Jedi\_ID). Each Padawan must have a master and cannot exist without one. The Padawan table also keeps track of each Padawan's first and second name, their years in training and their race.

The fourth table is the Battalion table. A battalion is a group of clones which are commanded by one Jedi. A battalion cannot exist without their Jedi commander. Each battalion has a unique name which is used as the primary key. A battalion also has foreign key that is the commanding Jedi's ID. The table also tracks the number of troops in each battalion.

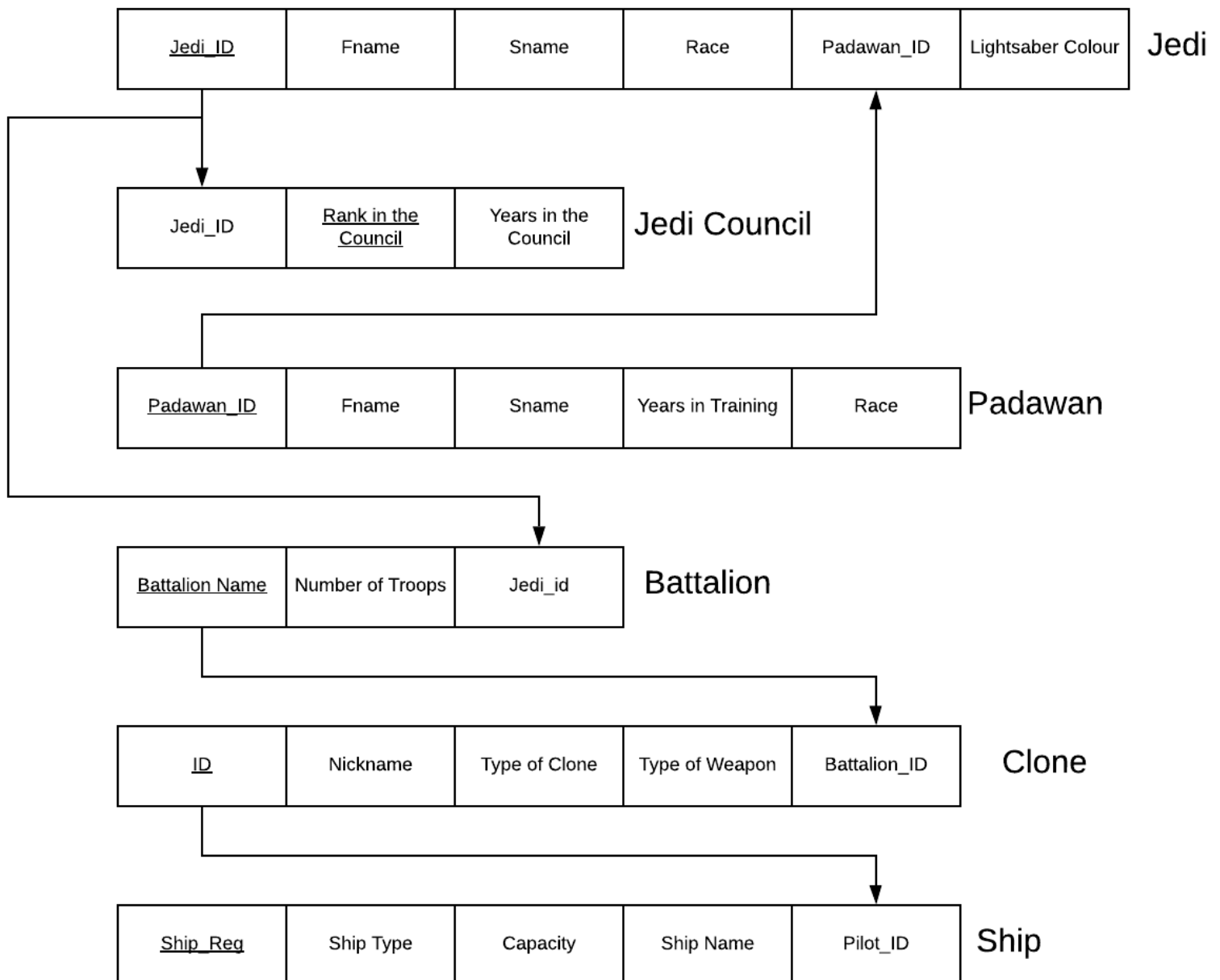
The fifth table is Clones. The clones table keeps track of each clone that makes up the battalions. Each clone has unique id (ID) this ID is used as the primary key in the table. The

table tracks what battalion each clone is in with the foreign key `Battalion_name`. The table also tracks their nickname, type of weapon and what type of clone they are.

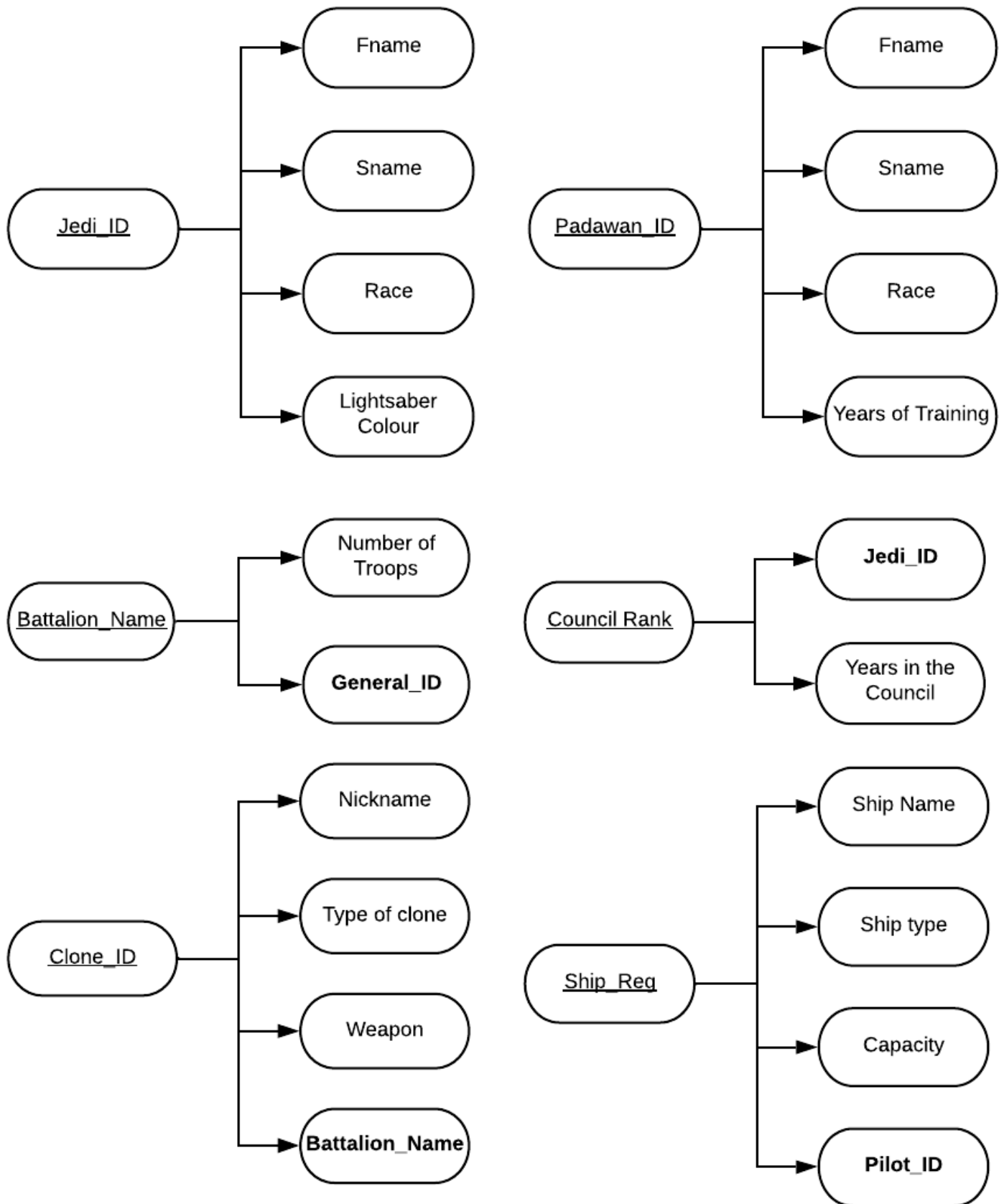
## Entity Relationship Diagram



## Relational Schema



## Functional Dependency Diagram



## Constraints

Every table in my Database had its Primary Key created as NOT NULL. Each Primary key must be made NOT NULL so each entry can be uniquely identified. Having the NOT NULL also helps to avoid integrity constraint violations which in turn creates entity Integrity Constraint. Similarly to the primary keys in my tables, I also created my foreign keys as NOT NULL. These keys formally reference a referential integrity

There are a number of semantic constraints in this database. For example the Jedi Council can only have 10 members. This means that Rank\_in\_the\_council can only be a number between 1 and 10. Similarly, in the Clones table there are only 5 types of clones so the attribute Type\_of\_clone is restricted to the 5 possible types, 'Heavy', 'Scout', 'Pilot', 'Medic' and 'Engineer'. Furthermore, I used constraints to limit the possible colours of a Jedi's lightsaber.

I implemented the following checks on my tables to maintain data integrity:

- CONSTRAINT check\_lightsaber\_colour CHECK( Lightsaber Colour IN ('green', 'blue', 'purple'));
- CONSTRAINT check\_Jedi\_ID CHECK(Jedi\_ID> 99999999);
- CONSTRAINT check\_council\_rank CHECK(rank\_in\_council>0 AND rank\_in\_council<11);
- CONSTRAINT check\_clone\_type CHECK(type\_of\_clone IN ('heavy', 'scout', 'medic', 'pilot', 'engineer'));

## Security

A good example of a security constraint that could be implemented in this application would be for use by a Jedi who is the commander of a Battalion of troops. A commander would need to have SELECT and UPDATE privileges on the battalion table they would also require

SELECT and DELETE privileges on the clones table. They may require this kind of access, if one of their clones die in battle they would have to removed from the clones table.

The following commands are my implementation of the above security constraint:

```
Create ROLE commander_role IDENTIFIED by iCommander;
Grant select on Table Battalion to commander_role;
Grant Update on Table Battalion to commander_role;
Grant select on Table Clones to commander_role;
Grant delete on Table Clones to commander_role;
Grant commander_role TO yoda;
```

## View Creation

For my database I have implemented 2 views. The first view is uses a trigger so that each time a new Battalion is added a new view of the of all the troops in the battalion.

```
CREATE TRIGGER commanderTable
BEFORE INSERT ON Battalion
FOR EACH ROW
BEGIN
CREATE VIEW commanderView AS
SELECT clone_ID, Nickname, Type_of_Weapon
FROM Clones
WHERE Clones.Battalion_name = Battalion.Battalion_name;
```

The second view I created was for the Jedi council. The view gives all names and races the members of the Jedi Council.

```
CREATE VIEW councilView AS
SELECT First_name, Second_name, Race
FROM Jedi
WHERE Jedi_Council.jedi_ID = Jedi.jedi_ID;
```

## Triggers

I defined one trigger in my database to decrement the number of troops in a battalion if a clone is removed from the table.

```
CREATE TRIGGER number_of_troops
BEFORE DELETE ON Clones
FOR EACH ROW
BEGIN
    UPDATE Battalion
    SET Number_Of_Troops = Number_Of_Troops - 1
    WHERE Clones.Battalion_name = Battalion.Battalion_name
END number_of_troops;
RUN;
```

```
CREATE TRIGGER not_the_younglings
BEFORE INSERT ON Jedi
FOR EACH ROW
BEGIN
    DROP TABLE Padawan
    WHERE Jedi.First_name = Anakin
END not_the_younglings;
RUN;
```



## Appendix

```
CREATE TABLE `Battalion` (  
  `Battalion_name` varchar(20) NOT NULL,  
  `Jedi_ID` int(11) NOT NULL,  
  `Number_Of_Troops` int(11) NOT NULL,  
  PRIMARY KEY (`Battalion_name`),  
  KEY `Jedi_ID` (`Jedi_ID`),  
  CONSTRAINT `battalion_ibfk_1` FOREIGN KEY (`Jedi_ID`) REFERENCES `jedi`  
  (`jedi_ID`)  
);
```

```
INSERT INTO `Battalion` VALUES ('104th Wold pack',100000003,16000),  
('13th Iron Battalion',100000016,650),  
('212th Attack',100000010,800),  
('21st nova Corps',100000002,40000),  
('327th star Corps',100000005,220000),  
('41st Elite Corps',100000012,32000),(  
'501st division',100000011,13000);
```

```
CREATE TABLE `Clones` (  
  `clone_ID` varchar(14) NOT NULL,  
  `Nickname` varchar(20) DEFAULT NULL,  
  `Type_of_Clone` varchar(10) NOT NULL,  
  `Battalion_name` varchar(20) NOT NULL,  
  `Type_of_Weapon` varchar(30) DEFAULT NULL,  
  PRIMARY KEY (`clone_ID`),  
  KEY `Battalion_name` (`Battalion_name`),  
  CONSTRAINT `clones_ibfk_1` FOREIGN KEY (`Battalion_name`) REFERENCES  
  `battalion` (`Battalion_name`),  
  CONSTRAINT `check_clone_type` CHECK ((`Type_of_clone` in  
  (_utf8mb4'heavy',_utf8mb4'scout',_utf8mb4'medic',_utf8mb4'pilot',_utf8mb4'engineer',_utf8  
  mb4'commander')))  
);
```

```
INSERT INTO `Clones` VALUES ('CC-1004','Gree','commander','41st Elite Corps','DT-15  
Blaster Pistol'),  
('CC-1138','Bacara','commander','21st nova Corps','Blaster Pistol'),  
('CC-2224','Cody','commander','212th Attack','E-11 blaster rifle'),  
('CC-2237','Odd Ball','pilot','212th Attack','Blaster Pistol'),  
('CC-3012','Lucky','pilot','327th star Corps','E-11 blaster rifle'),  
('CC-3014','Ace','pilot','13th Iron Battalion','E-11 blaster rifle'),  
('CC-3021','Jag','pilot','104th Wold pack','E-11 blaster rifle'),  
('CC-3636','Wolffe','commander','104th Wold pack','E-11 blaster rifle'),  
('CC-5052','Bly','commander','327th star Corps','Blaster Pistol'),  
('CT-4981','Contrail','pilot','501st division','Blaster Pistol');
```

```
CREATE TABLE `Jedi` (  
  `jedi_ID` int(11) NOT NULL,  
  `First_name` varchar(20) NOT NULL,
```

```
`Second_Name` varchar(20) DEFAULT NULL,  
`Lightsaber_colour` varchar(10) DEFAULT NULL,  
`Race` varchar(20) DEFAULT NULL,  
PRIMARY KEY (`jedi_ID`),  
CONSTRAINT `check_Jedi_ID` CHECK ((`Jedi_ID` > 99999999)),  
CONSTRAINT `check_lightsaber_colour` CHECK ((`Lightsaber_colour` in  
(_utf8mb4'green',_utf8mb4'blue',_utf8mb4'purple',_utf8mb4'yellow')))  
);
```

```
INSERT INTO `Jedi` VALUES (100000000,'Yoda','','green','Race unknown'),  
(100000001,'Mace','Windu','purple','Human'),  
(100000002,'Ki','Adi-mundi','blue','Cerean'),  
(100000003,'Plo','Koon','blue','Kel dor'),  
(100000004,'Yaddle','','green','Race Unknown'),  
(100000005,'Kit','Fisto','green','Nautolan'),  
(100000006,'Saesee','Tiin','green','Iktotchi'),  
(100000007,'Shaak','Ti','blue','Togruta'),  
(100000008,'Oppo','Rancisis','green','Thisspiasian'),  
(100000009,'Eeth','Koth','green','Zabrak'),  
(100000010,'Obi-wan','Kenobi','blue','Human'),  
(100000011,'Anakin','Skywalker','blue','Human'),  
(100000012,'Luminara','Unduli','blue','Mirialan'),  
(100000014,'Quinlan','Vos','green','Kiffar'),  
(100000015,'Qui-gon','Jinn','green','Human'),  
(100000016,'Jaro','Tapal','blue','Lasat');
```

```
CREATE TABLE `Jedi_Council` (  
  `Rank_in_Council` int(11) NOT NULL,  
  `Jedi_ID` int(11) NOT NULL,  
  `Years_in_Council` int(11) NOT NULL,  
  PRIMARY KEY (`Rank_in_Council`),  
  KEY `Jedi_ID` (`Jedi_ID`),  
  CONSTRAINT `council_key` FOREIGN KEY (`Jedi_ID`) REFERENCES `jedi`  
  (`jedi_ID`)  
)
```

```
INSERT INTO `Jedi_Council` VALUES (1,100000000,200),  
(2,100000001,10),  
(3,100000002,15),  
(4,100000003,4),  
(5,100000004,56),  
(6,100000010,6),  
(7,100000003,23),  
(8,100000006,2),  
(9,100000007,7),  
(10,100000008,12);
```

```
CREATE TABLE `Padawan` (  
  `Padawan_ID` int(11) NOT NULL,  
  `Years_in_Training` int(11) NOT NULL,
```

```
`Jedi_ID` int(11) NOT NULL,  
`First_Name` varchar(20) NOT NULL,  
`Second_Name` varchar(20) NOT NULL,  
`Race` varchar(30) NOT NULL,  
PRIMARY KEY (`Padawan_ID`),  
KEY `Jedi_ID` (`Jedi_ID`),  
CONSTRAINT `padawan_ibfk_1` FOREIGN KEY (`Jedi_ID`) REFERENCES `jedi`  
(`jedi_ID`)  
);
```

```
INSERT INTO `Padawan` VALUES (1,3,100000012,'Barriss','Offee','Mirialan'),  
(2,3,100000011,'Ahsoka','Tano','Togruta'),  
(3,6,100000014,'Aayla','Secura','Twi-lek'),  
(4,4,100000005,'Nahdar','Vebb','Moncalamari'),  
(5,2,100000016,'Cal','Kestis','Human');
```

```
CREATE TABLE `Ships` (  
`Ship_Reg` varchar(10) NOT NULL,  
`Ship_Name` varchar(20) DEFAULT NULL,  
`Ship_capacity` int(11) NOT NULL,  
`Ship_Type` varchar(30) NOT NULL,  
`Pilot_ID` varchar(14) NOT NULL,  
PRIMARY KEY (`Ship_Reg`),  
KEY `Pilot_ID` (`Pilot_ID`),  
CONSTRAINT `ships_ibfk_1` FOREIGN KEY (`Pilot_ID`) REFERENCES `clones`  
(`clone_ID`)  
);
```

```
INSERT INTO `Ships` VALUES ('01-D-23412','speedy',2,'V-19 Torrent starfighter','CC-  
2237'),  
('04-L-42193','BOOMER',6,'BTL-B-Y-wing fighter','CC-3014'),  
('12-K-13231','',2,'ARC-170 Starfighter','CC-3021'),  
('34-B-32414','Sturdy',14,'Low Altitude Assault Transport','CT-4981'),  
('67-A-98765','rebel',1,'X-wing starfighter','CC-3012');
```