Information Management Database Report

Chloe Conneely 17323080

# *Description*

My database is designed on the Japanese animated science fiction series ‘Mobile Suit Gundam’. The series tells the story of a war between the Earth Federation and the space colony, the Principality of Zeon. In this series, the soldiers fight battles in spacecrafts called “Mobile Suits”. These are large robots into which a soldier (referred to as a pilot) sits and controls. When mobile suits are not in battle, they are based in a spaceship. My database describes the relations between the soldiers, mobile suits, ships and battles that are shown in the series.

The soldiers have an ID, a name, an age (not based off date of birth since the series takes place in an alternate timeline), a rank, the ship they are aboard, their allegiance, if they are a pilot and a status indicating whether they are dead or alive. A soldier can be assigned to be aboard only one ship.

The mobile suits have an ID, a name, a pilot ID and a base. A mobile suit can only have one pilot and one base. Multiple mobile suits can fight multiple battles.

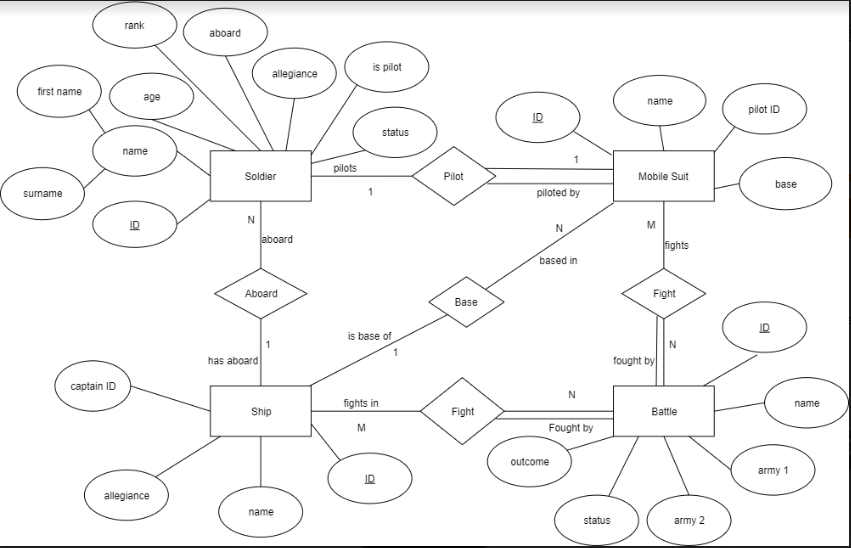
The spaceships (ships) have an ID, a name, an allegiance and a captain ID. A ship can only have one captain. Multiple ships can fight in multiple battles. A ship can be a base for multiple mobile suits and can have multiple soldiers aboard.

The battles have an ID, a name, a first army, second army, status and an outcome.

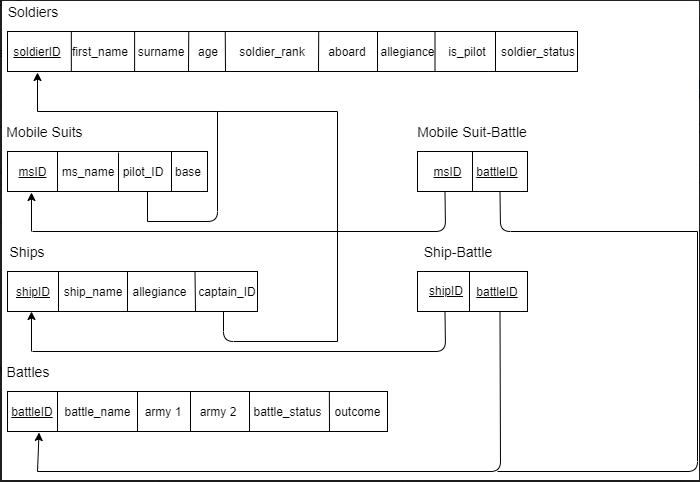
There is a table describing the relationship between the mobile suits and battles in which they fought.

Another table describes the relationship between the ships and the battles in which they fought.

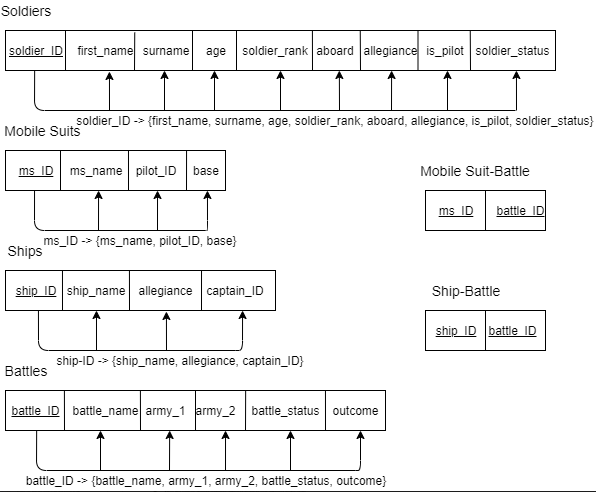
# *Entity Relationship Diagram*



# *Relational Schema*



# *Relational Dependency*



# *Normalisation*

1st Normal Form: The tables are in 1st normal form as each attribute is atomic and contains a single value.

2nd Normal Form: The tables are in 2nd normal form as every non key attribute is fully functionally dependent on the primary key for that table.

3rd Normal Form: The tables are in 3rd normal form as no non-key attributes are transitively dependent on the primary key.

Boyce-Codd Normal Form: The tables are in Boyce-Codd normal form as all attributes in the relation are dependent on the key, the whole key and nothing but the key.

# *Constraints*

The constraint “check\_status” constrains soldier status to be either “Alive” or “Dead”.

The constraint “check\_allegiance” constrains soldier allegiance to be either “Earth Federation Forces” or “Principality of Zeon”.

The constraint “check\_is\_pilot” constrains is\_pilot to be either “Yes” or “No”.

The constraint “check\_ship\_allegiance” constrains ship allegiance to be either “Earth Federation Forces” or “Principality of Zeon”.

The constraint “check\_battle\_status” constrains battle status to be either “Ongoing” or “Complete”.

Primary Keys:

* “soldierID” is the primary key for the “soldiers” table
* “msID” is the primary key for the “mobile\_suits” table
* “shipID” is the primary key for the “ships” table
* “battleID” is the primary key for the “battles” table
* “msID” and “battleID” are the primary key for the “mobile\_suit\_battle” table
* “shipID” and “battleID” are the primary keys for the “ship\_battle” table

Foreign Keys:

* “battleID” in “mobile\_suit\_battle” and “ship\_battle” are foreign keys of “battleID” in “battles”
* “msID” in “mobile\_suit\_battle” is a foreign key of “msID” in “mobile\_suits”
* “base” in “mobile\_suit\_battle” is a foreign key of “ship\_name” in “ships”
* “shipID” in “ship\_battle” is a foreign key of “shipID” in “ships”
* “pilotID” in “mobile\_suits” is a foreign key of “soldierID” in “soldiers”
* “captainID” in “ships” is a foreign key of “soldierID” in “soldiers”

Not Null:

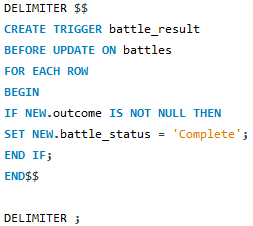
* All attributes except for “outcome” in “battles” are not null
* “outcome” CAN be null as the outcome of a battle cannot be determined until a battle has been complete

Unique:

* “ship\_name” is unique as all ships must have unique names to prevent confusion
* “captainID” is unique as a captain can only be assigned to one ship

Trigger:

The trigger “battle\_result” is triggered when “outcome” in “battles” is no longer NULL. The trigger will change “battle\_status” in “battles” to “complete” when “outcome” is no longer NULL.



# *Security*

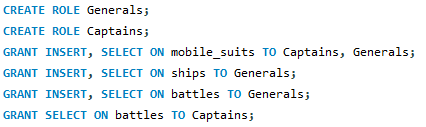
The security of a database is concerned with deliberate corruption. Security is maintained using access control and security policies. Access control prevents unauthorised persons from accessing the system. The database has a database administrator who controls user account creation, privilege granting and revocation, security level assignment and role-based access.

The database administrator grants privileges through the command GRANT <privilege> ON <relation> TO <user>.

The database administrator revokes privileges through the command REVOKE <privilege> ON <relation> FROM <user>.

Role-based access grants privileges to roles. Users are then assigned roles from which they receive privileges. The database administrator can create roles using CREATE ROLE <role>. The database administrator can also destroy roles using DESTROY ROLE <role>. Roles are assigned to users by the command GRANT <role> TO <user>. A role can be assigned to multiple users. A user can be assigned multiple roles. Any user assigned a role receives the privileges of that role.

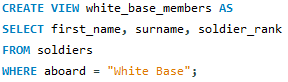
In this database the database administrator creates roles for ship captains (“Captains”) and generals (“Generals”). “Captains” can retrieve information on battles and update and retrieve information on mobile suits as they oversee the assignment of pilots to mobile suits based on their ship. “Generals” can update and retrieve information on mobile suits, ships and battles as they assign mobile suits to ships, assign captains to ships and record information relating to battles.



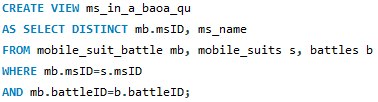
# *Views*

Views allow the owner of a relation to grant limited access to information in the relation. The view itself acts as a new relation. A view is created using CREATE <view> AS SELECT <attributes> FROM <relation> WHERE <condition>. This database contains two views.

The first view “white\_base\_members” shows which soldiers are aboard the ship “White Base” and what their rank is. “White Base” is the ship on which the protagonists of “Mobile Suit Gundam” are stationed.



The second view “ms\_in\_a\_baoa\_qu” shows which mobile suits are fighting in The Battle of A Baoa Qu. This view uses a join to search “mobile\_suits”, “battles” and “mobile\_suit\_battle” to display IDs and names. The Battle of A Baoa Qu is the final battle in “Mobile Suit Gundam”.



# *Code*

-- create soldier table

CREATE TABLE soldiers(

soldierID INT NOT NULL,

first\_name VARCHAR(20) NOT NULL,

surname VARCHAR(50) NOT NULL,

age INT,

soldier\_rank VARCHAR (45),

aboard VARCHAR(30) NOT NULL,

allegiance VARCHAR(100) NOT NULL,

is\_pilot VARCHAR(5) NOT NULL,

soldier\_status VARCHAR (10) NOT NULL,

PRIMARY KEY (soldierID)

);

-- create mobile suit table

CREATE TABLE mobile\_suits(

msID INT NOT NULL,

ms\_name VARCHAR(50) NOT NULL,

pilotID INT NOT NULL,

baseID INT NOT NULL,

PRIMARY KEY (msID)

);

-- create ship table

CREATE TABLE ships(

shipID INT NOT NULL,

ship\_name VARCHAR(30) NOT NULL,

allegiance VARCHAR (100) NOT NULL,

captainID INT,

PRIMARY KEY (shipID)

);

-- create battles table

CREATE TABLE battles(

battleID INT NOT NULL AUTO\_INCREMENT,

battle\_name VARCHAR(100) NOT NULL,

army\_1 VARCHAR(100) NOT NULL,

army\_2 VARCHAR(100) NOT NULL,

winner VARCHAR(100) NOT NULL,

outcome VARCHAR(2000) NULL,

PRIMARY KEY(battleID)

);

-- create mobile suit to battle table

CREATE TABLE mobile\_suit\_battle(

msID INT NOT NULL,

battleID INT NOT NULL,

PRIMARY KEY(msID, battleID)

);

-- create ship to battle table

CREATE TABLE ship\_battle(

shipID INT NOT NULL,

battleID INT NOT NULL,

PRIMARY KEY(shipID, battleID)

);

-- insert data

-- insert into soldiers

INSERT INTO soldiers (soldierID, first\_name, surname, age, soldier\_rank, aboard, allegiance, is\_pilot, soldier\_status)

VALUES (1, 'Amuro', 'Ray', 16, 'Ensign','White Base', 'Earth Federation Forces', 'Yes', 'Alive');

INSERT INTO soldiers (soldierID, first\_name, surname, age, soldier\_rank, aboard, allegiance, is\_pilot, soldier\_status)

VALUES (2, 'Char', 'Aznable', 20, 'Lieutenant Commander','Zanzibar Ingolstadt', 'Principality of Zeon', 'Yes', 'Alive');

INSERT INTO soldiers (soldierID, first\_name, surname, age, soldier\_rank, aboard, allegiance, is\_pilot, soldier\_status)

VALUES (3, 'Fraw', 'Bow', 15, 'Superior Private','White Base', 'Earth Federation Forces', 'No', 'Alive');

INSERT INTO soldiers (soldierID, first\_name, surname, age, soldier\_rank, aboard, allegiance, is\_pilot, soldier\_status)

VALUES (4, 'Sayla', 'Mass', 17, 'Petty Officer','White Base', 'Earth Federation Forces', 'Yes', 'Alive');

INSERT INTO soldiers (soldierID, first\_name, surname, age, soldier\_rank, aboard, allegiance, is\_pilot, soldier\_status)

VALUES (5, 'Garma', 'Zabi', 20, 'Lieutenant General','Gaw', 'Principality of Zeon', 'Yes', 'Dead');

INSERT INTO soldiers (soldierID, first\_name, surname, age, soldier\_rank, aboard, allegiance, is\_pilot, soldier\_status)

VALUES (6, 'Bright', 'Noa', 19, 'Lieutenant Junior Grade','White Base', 'Earth Federation Forces', 'Yes', 'Alive');

INSERT INTO soldiers (soldierID, first\_name, surname, age, soldier\_rank, aboard, allegiance, is\_pilot, soldier\_status)

VALUES (7, 'Kycilia', 'Zabi', 27, 'Rear Admiral','BB-38 Gwazine', 'Principality of Zeon', 'Yes', 'Dead');

INSERT INTO soldiers (soldierID, first\_name, surname, age, soldier\_rank, aboard, allegiance, is\_pilot, soldier\_status)

VALUES (8, 'Degwin', 'Zabi', 62, 'Sovereign Ruler','Great Degwin', 'Principality of Zeon', 'No', 'Dead');

-- insert into ships

INSERT INTO ships (shipID, ship\_name, allegiance, captainID)

VALUES (1, 'White Base', 'Earth Federation Forces', 6);

INSERT INTO ships (shipID, ship\_name, allegiance, captainID)

VALUES (2, 'Zanzibar Ingolstadt', 'Principality of Zeon', 2);

INSERT INTO ships (shipID, ship\_name, allegiance, captainID)

VALUES (3, 'Gaw', 'Principality of Zeon', 5);

INSERT INTO ships (shipID, ship\_name, allegiance, captainID)

VALUES (4, 'BB-38 Gwazine', 'Principality of Zeon', 7);

INSERT INTO ships (shipID, ship\_name, allegiance, captainID)

VALUES (5, 'Great Degwin', 'Principality of Zeon', 8);

-- insert into mobile\_suits

INSERT INTO mobile\_suits (msID, ms\_name, pilotID, base)

VALUES (1, 'RX-78-2 Gundam', 1, "White Base");

INSERT INTO mobile\_suits (msID, ms\_name, pilotID, base)

VALUES (2, 'MSN-02 Zeong', 2, "Zanzibar Ingolstadt");

INSERT INTO mobile\_suits (msID, ms\_name, pilotID, base)

VALUES (3, 'G-Fighter', 4, "White Base");

INSERT INTO mobile\_suits (msID, ms\_name, pilotID, base)

VALUES (4, 'MS-06FS Zaku II', 5, "Gaw");

INSERT INTO mobile\_suits (msID, ms\_name, pilotID, base)

VALUES (5, 'MAX-03 Adzam', 7, "BB-38 Gwazine");

-- insert into battles

INSERT INTO battles (battleID, battle\_name, army\_1, army\_2, battle\_status, outcome)

VALUES (1, 'Battle of Odessa', "Earth Federation Forces", "Principality of Zeon", "Complete", "Victory for the Earth Federation Forces. Zeon mining facilities captured. Zeon forces removed from Europe and Asia.");

INSERT INTO battles (battleID, battle\_name, army\_1, army\_2, battle\_status, outcome)

VALUES (2, 'Battle of Jaburo', "Earth Federation Forces", "Principality of Zeon", "Complete", "Victory for the Earth Federation Forces despite heavy losses. Zeon forces retreated to space.");

INSERT INTO battles (battleID, battle\_name, army\_1, army\_2, battle\_status, outcome)

VALUES (3, 'Operation Rubicon', "Earth Federation Forces", "Principality of Zeon", "Complete", "Cyclops Team failed to destroy RX-78NT-1 Gundam. RX-78NT-1 Gundam suffered damage and requires repairs. Zeon violated the Antarctic Treaty by attacking a colony.");

INSERT INTO battles (battleID, battle\_name, army\_1, army\_2, battle\_status, outcome)

VALUES (4, 'Operation Star One', "Earth Federation Forces", "Principality of Zeon", "Complete", "Victory for Earth Federation Forces. Death of Degwin Zabi, Gihren Zabi and Kycilia Zabi, ending the Zabi's rule of Zeon.");

INSERT INTO battles (battleID, battle\_name, army\_1, army\_2, battle\_status)

VALUES (5, 'Battle of A Baoa Qu', "Earth Federation Forces", "Principality of Zeon", "Ongoing");

INSERT INTO battles (battleID, battle\_name, army\_1, army\_2, battle\_status, outcome)

VALUES (6, 'One Week Battle', "Earth Federation Forces", "Principality of Zeon", "Complete", "Victory for Principality of Zeon. Side 1, Side 2 and Side 4 conquered by Zeon and civilians killed by poison gas. Side 2 colony dropped on Earth causing death and destruction.");

INSERT INTO battles (battleID, battle\_name, army\_1, army\_2, battle\_status, outcome)

VALUES (7, 'The Battle of Loum', "Earth Federation Forces", "Principality of Zeon", "Complete", "Victory for Principality of Zeon. Complete destruction of Side 5 by Zeon. Earth Federation Forces prevented colony drop of Side 5 and save some civilians.");

-- insert into mobile\_suit\_battle

INSERT INTO mobile\_suit\_battle (msID, battleID)

VALUES (1, 1);

INSERT INTO mobile\_suit\_battle (msID, battleID)

VALUES (1, 2);

INSERT INTO mobile\_suit\_battle (msID, battleID)

VALUES (2, 4);

INSERT INTO mobile\_suit\_battle (msID, battleID)

VALUES (1, 5);

INSERT INTO mobile\_suit\_battle (msID, battleID)

VALUES (2, 5);

INSERT INTO mobile\_suit\_battle (msID, battleID)

VALUES (3, 5);

-- insert into ship\_battle

INSERT INTO ship\_battle (shipID, battleID)

VALUES (1, 1);

INSERT INTO ship\_battle (shipID, battleID)

VALUES (1, 2);

INSERT INTO ship\_battle (shipID, battleID)

VALUES (1, 5);

INSERT INTO ship\_battle (shipID, battleID)

VALUES (2, 5);

INSERT INTO ship\_battle (shipID, battleID)

VALUES (4, 5);

INSERT INTO ship\_battle (shipID, battleID)

VALUES (4, 7);

-- create foreign keys

-- msID in suit-battle table is foreign key of msID in suit table

ALTER TABLE mobile\_suit\_battle

ADD FOREIGN KEY (msID) REFERENCES mobile\_suits(msID);

-- battleID in suit-battle table is foreign key of battleID in battles

ALTER TABLE mobile\_suit\_battle

ADD FOREIGN KEY (battleID) REFERENCES battles(battleID);

-- shipID in ship-battle table is foreign key of ship ID in ship table

ALTER TABLE ship\_battle

ADD FOREIGN KEY (shipID) REFERENCES ships(shipID);

-- battleID in ship-battle table is foreign key of battleID in battles

ALTER TABLE ship\_battle

ADD FOREIGN KEY (battleID) REFERENCES battles(battleID);

-- pilotID in suit table is foreign key of soldierID in soldier table

ALTER TABLE mobile\_suits

ADD FOREIGN KEY (pilotID) REFERENCES soldiers(soldierID);

-- base in suit table is foreign key of ship\_name in ship table

ALTER TABLE mobile\_suits

ADD FOREIGN KEY (base) REFERENCES ships(ship\_name);

-- aboard in soldier table is foreign key of ship\_name in ship table

ALTER TABLE soldiers

ADD FOREIGN KEY (aboard) REFERENCES ships(ship\_name);

-- captainID in ship table is foreign key of soldierID in soldier table

ALTER TABLE ships

ADD FOREIGN KEY (captainID) REFERENCES soldiers(soldierID);

--create views

CREATE VIEW white\_base\_members AS

SELECT first\_name, surname, soldier\_rank

FROM soldiers

WHERE aboard = "White Base";

CREATE VIEW ms\_in\_a\_baoa\_qu

AS SELECT DISTINCT mb.msID, ms\_name

FROM mobile\_suit\_battle mb, mobile\_suits s, battles b

WHERE mb.msID=s.msID

AND mb.battleID=b.battleID;

--add constraints

ALTER TABLE soldiers

ADD CONSTRAINT `check\_status` CHECK (soldier\_status = 'Alive' OR soldier\_status = 'Dead');

ALTER TABLE soldiers

ADD CONSTRAINT `check\_allegiance` CHECK (allegiance = 'Earth Federation Forces' OR allegiance = 'Principality of Zeon');

ALTER TABLE soldiers

ADD CONSTRAINT `check\_pilot` CHECK (is\_pilot = 'Yes' OR is\_pilot = 'No');

ALTER TABLE ships

ADD CONSTRAINT `check\_ship\_allegiance` CHECK (allegiance = 'Earth Federation Forces' OR allegiance = 'Principality of Zeon');

ALTER TABLE battles

ADD CONSTRAINT `check\_battle\_status` CHECK (battle\_status = 'Ongoing' OR battle\_status = 'Complete');

--add trigger

DELIMITER $$

CREATE TRIGGER battle\_result

BEFORE UPDATE ON battles

FOR EACH ROW

BEGIN

IF NEW.outcome IS NOT NULL THEN

SET NEW.battle\_status = 'Complete';

END IF;

END$$

DELIMITER ;

--create roles for security features

CREATE ROLE Generals;

CREATE ROLE Captains;

GRANT INSERT, SELECT ON mobile\_suits TO Captains, Generals;

GRANT INSERT, SELECT ON ships TO Generals;

GRANT INSERT, SELECT ON battles TO Generals;

GRANT SELECT ON battles TO Captains;