REQUIREMENTS DOCUMENT

TEAM NUMBER: 12

TEAM MEMBERS:

Sriteja Reddy Pashya (2021111019) , Keval Jain (2021111030) , Romica Raisinghani (2021101053)

MOVIE: Star Wars

SECTION 1 : Introduction to the mini-world

After the fall of the Jedi order, the Empire has taken control of the Galactic Republic. The rebel alliance is an interstellar coalition of anti-imperial cell systems as a resistance movement against the Empire.The mini world is constructed from a rebellions' point of view.It stores the data known by the rebels about the status of the war, and relevant information about their assets and operations.

SECTION 2 : Purpose of the database

This database keeps track of the information such as quantity, usage and whereabouts of vehicles, weapons and other equipment, information about the soldiers and strategic locations like battlefields. This data is essential in helping the rebels take tactical decisions for defense and plan offensives.

SECTION 3 : Users of the database

The users of the database are officials of the rank commander or higher from the rebel stronghold.

SECTION 4 : Applications of the database

1. To keep track of various equipment and weapons and their quantities
2. To track of the usage and whereabouts of vehicles
3. To keep track of information about the soldiers and their roles
4. To use the information to help them plan out wartime policy

SECTION 5 : Database Requirements

a) Assumptions

1. "Location" is a weak entity which is dependent on a Planet and has "Name" as the partial key
2. A person inhabits only one planet at a time
3. A person can use multiple vehicles at different instances of time
4. People use Vehicles to transport Equipment
5. A single person commands multiple person

b) Strong Entity types

Planets: Name (Primary Key), Inhabitants, Political Status (Rebel Ally/Empire), [Position(Compound) : Grid coordinates, Region (Derived Attribute)]

People: [Name (Primary Key Attribute & Compound Attribute) :First Name, Last Name], DOB (DD/MM/YYYY), Age (Derived Attribute: Can be derived from DoB), Gender, Home Planet, Living Status (Alive/Dead), Occupation/Role (Multi-Valued)

Vehicle: Name, Vehicle Model, Vehicle-ID (Primary Key), Purpose (Multi-valued), [Vehicle Info(Compound) : Dimensions (Derived attribute from Model), Weight (Derived attribute from Model), Date of Manufacture]

Equipment: Name, Type (Weapon, Droid, etc), [Equipment Info(Compound) : Dimensions, Date of manufacture], EquipmentID (Primary Key)

c) Weak entity types

Location: Name (Partial Key),Purpose

d) Relationship types

1. **Inhabit:** People inhabit planets
    1. Degree: 2
    2. Participating Entity Types: People - Planet
    3. Cardinality Ratio, People : Planet = N:1
2. **Use:** People use Vehicles
    1. Degree: 2
    2. Participating Entity Types: People - Vehicle
    3. Cardinality Ratio, People : Vehicle = N:M
3. **Transported_by(Ternary Relationship Type):** People use Vehicles to transport Equipment
    1. Degree: 3
    2. Participating Entity Types: People - Vehicle - Equipment
    3. People - Vehicle: many to many
    4. Vehicle - Equipment: many to many
    5. People - Equipment: many to many
    6. Cardinality Ratio, People : Vehicle : Equipment = N:M:K
4. **Situated_in(Identifying Relationship Type):** Locations are situated in Planet
    1. Degree: 2
    2. Participating Entity Types: Location - Planet
    3. Cardinality Ratio, Location : Planet = N:1
5. **Command(Recursive Relationship Type):** People can command other People
    1. Degree: 2
    2. Participating Entity Types: People - People
    3. Cardinality Ratio, People : People = N : 1

SECTION 6 : Functional Requirements

Modifications

1. **Insert:**

insert_planet: inserts a new planet with corresponding attributes, **insert_people**: inserts a person and relates it with its home planet and other attributes, **insert_vehicle**: inserts a new vehicle with corresponding attributes, **insert_equipment**: inserts a new equipment which can be a weapon or a droid, **insert_location**: inserts a new location that is dependent on the planet where it is inserted

2. Delete:

delete_planet: delete the entire data corresponding to a particular planet, **delete_people**: delete the entire data corresponding to a particular person, **delete_vehicle**: delete the entire data corresponding to a particular vehicle, **delete_equipment**: delete the entire data corresponding to a particular equipment, **delete_location**: delete the entire data corresponding to a particular location

3. Update:

remove_vehicle: removes a link between the vehicle and a particular person without actually deleting the data of the vehicle from the database, **replace_vehicle**: updates the link between the vehicle and a particular person by replacing the old vehicle with a new one, **remove_equipment**: removes a link between the equipment and a particular person without actually deleting the data of the equipment from the database, **replace_equipment**: updates the link between the equipment and a particular person by replacing the old equipment with a new one

Retrievals

1. Selection:

- **Origin:** Give details of all people who are from a certain planet
- **Vehicle_Purpose:** Give details of all vehicles with a certain purpose
- **Equipment_Type:** Give details of all equipment of a certain type
- **People_Vehicle_Purpose:** Give details of people who use a certain vehicle purpose
- **Planet_Region:** Gives details of all planets in a certain region

2. Projection:

- **Vehicle_Model:** Gives the id, name and model of all vehicles
- **Equipment_Type:** Gives the id, name and equipment type for all equipment
- **People_Occupation:** Gives the name, occupation, age of all people
- **Vehicle_Planet:** Gives the vehicle id and name of all vehicles on a certain planet
- **Equipment_Planet:** Gives the equipment id and name of all equipment on a certain planet

3. Aggregate:

- **No_People_Planet:** Gives the number of people in a planet
- **No_People_Occupation:** Gives the number of people with a certain occupation
- **No_Vehicle_Purpose:** Gives the number of vehicle with a certain purpose
- **No_Equipment_Type:** Gives the number of Equipment of a certain type

4. Search:

- **Search_planet:** Searches a planet that is uniquely identified by its name
- **Search_people:** Searches the person that is uniquely identified by their name
- **Search_vehicle:** Searches the vehicle that is uniquely identified by its Vehicle-ID
- **Search_equipment:** Searches the equipment that is uniquely identified by its Equipment-ID
- **Search_location:** Searches for the location by its name in the respective planet

5. Analysis:

- **No_People_vehicle_purpose_planet:** Number of people who are using vehicles of a certain purpose on a given planet
- **Equipment_Type_Planet:** Gives the equipment id and name of equipment of a certain type on a given planet
- **People_Occupation_Region:** Gives the name and age of all people of a certain occupation in a certain region

SECTION 7 : Summary

This database aims to serve as the first stop information source for the rebels. It stores all the necessary data, and provides a wide range of functionalities to retrieve information that could be needed by the rebels.

A surprise attack has been launched, we need to quickly get the current manpower there to decide whether we should fight back or retreat —> This database has you covered.

This is a comprehensive mini-world supporting various kinds of operations for the rebel camp.