DNA

HW-2-ER DIAGRAM

Evaluating Team 24 Requirements Doc

Team

Ketaki Shetye - 2022114013

Maitreya Chitale - 2022114011

Prabhav Shetty - 2022111015

Sujal Deoda - 2022115001

Vinit Mehta - 2022111001

Evaluating Team 24's Requirements Analysis Document

Below mentioned are some of the inconsistencies that we noticed in the requirements analysis document of team 24:

Inconsistency in Entity Types and its Attributes:

(They did not mention any of the foreign keys so we have mentioned them entity type wise)

Marines:

- Instead of storing age we can store DOB as a stored attribute and make age as a derived attribute. Since age keeps changing each year, it would be difficult to manipulate the table every time to avoid inconsistency in data.
- Marine Base Number Attribute (Foreign Key) should be added to link each marine to its appointed marine base.

Marine Base:

- Location is specified to be a composite attribute having the same structure as address but the constituent attributes of neither of them is mentioned anywhere in the document. It can be Plot No., Street, Area, Village and Island.
- Number of officers is mentioned to be a stored attribute but it can be made into a derived attribute since each marine works on exactly one marine base so we can easily count the number of marines working on a particular base at any point of time. This would make it easier to maintain data consistency as we need not update the stored value of the number of marines everytime.

Pirates:

- Pirate Id (Primary Key) is mentioned to have NOT NULL constraint but PRIMARY KEY constraint in itself includes the NOT NULL constraint so we need not mention it explicitly. Therefore here there is redundancy of constraints.
- Crew Name Attribute (Foreign Key) should be added to link each pirate with its crew.
- Village Name Attribute (Foreign Key) should be added to link each pirate to its origin village.

Pirate Crew:

- We need not store the size of the crew, it can be made a derived attribute. Since each pirate belongs to exactly one pirate crew we can easily calculate the number of pirates in a pirate crew at any point in time.
- The current state attribute is mentioned to be varchar(50) but since it can take only 3 particular values - active, dormant and dissolved - it should be made an enum.

- Crew Name cannot be made a primary key as two pirates can have the same name (No assumption is made that they must have different names). So we would add an assumption that all pirates have distinct names.
- Crew Name (Primary Key) is mentioned to have NOT NULL constraint but PRIMARY KEY constraint in itself includes the NOT NULL constraint so we need not mention it explicitly. Therefore here there is redundancy of constraints.
- Village Name Attribute (Foreign Key) should be added which would store the name of the village where the particular pirate crew was spotted, to create a link between village last seen and pirate crew.

Village:

- Name of the village cannot be made a primary key as two villages can have the same name (No assumption on villages having distinct names is made). So we need to add an assumption that villages will have distinct names.
- Marine Base Number Attribute (Foreign Key) should be added to link each village with the corresponding marine base supervising that village.

Inconsistency in Relationship Types:

- **Relationship : Village (under supervision of) Marine Base** is mentioned under both binary and ternary relationship but it is a binary relationship as there are just 2 participating entities so it should not be mentioned under ternary relationship.
- All ternary relationships are also part of degree > 2 relationships (Redundancy) so either we should change the last heading to degree > 3 or include all ternary relationships under the degree > 2 heading at last.
- **Relationship : Pirate belongs to Pirate Crew** should have cardinality ratio N:1 since multiple pirates belong to the same pirate crew.
- **Relationship : Marine position in Marine Base** should have cardinality ratio N:1 since multiple marines are positioned on the same base.
- A pirate crew can participate in only one raid at a time in 1 village (By the assumption that if multiple pirate crews raid the same village at the same time they would be considered separate entities) then accordingly the cardinality ratio of the **relationship : Pirate Crew participated in Raid in Village** should be 1:1:1
- **Relationship : Marine Base (participated in) Marine Operation (on) Pirate Crew (headed by) Marine** should have the cardinality ratio N:1:M:1 since in one marine operation multiple marine bases can participate and multiple pirate crews can be targeted and the operation would be led by one head marine.
- They have also not mentioned the Participation constraints in minmax format, so considering the above rectifications the final set of relationships in minmax format is:
    - Pirate (1,N) (belongs to) Pirate Crew (1,1) - N:1
    - Pirate Crew (0,N) (last spotted at) Village (0,1) - N:1
    - Pirate (0,N) (originate from) Village (1,1) - N:1
    - Marine (0,N) (position in) Marine Base (1,1) - N:1

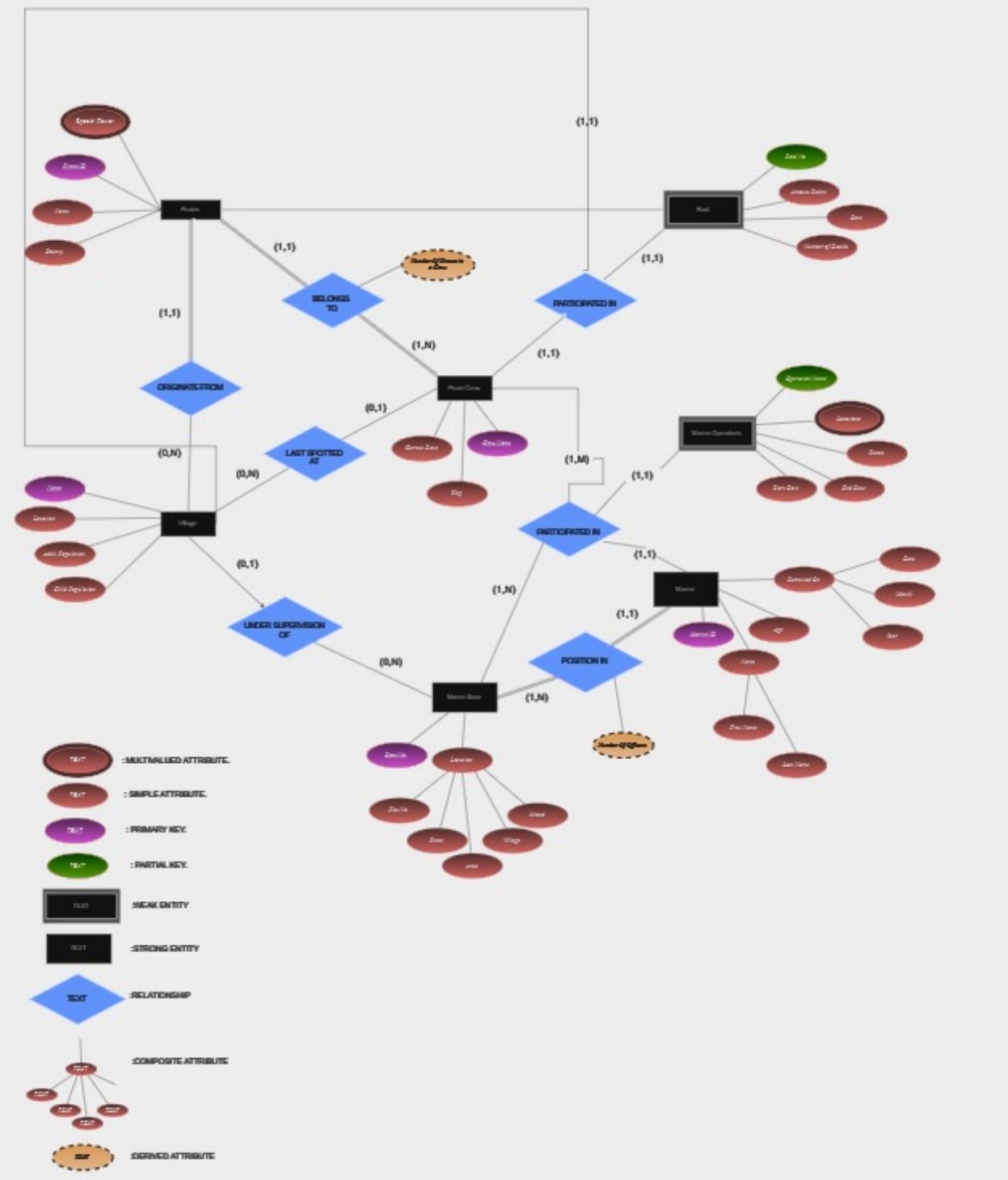
- Village (0,N) (under supervision of) Marine Base (0,1) - N:1
- Pirate Crew (1,1) (participated in) Raid (1,1) (on) Village (1,1) - 1:1:1
- Marine Base (1,N) (participated in) Marine Operation (1,1) (on) Pirate Crew (1,M) (headed by) Marine (1,1) - N:1:M:1

Inconsistency in Functional Requirements:

Delete:

- It is mentioned in the current state attribute of the pirate crew whether it is dissolved or not but we are also deleting the entry when the pirate crew is disbanded, so we need to keep one and discard the other. We choose to not delete the pirate crew entry when it is dissolved/disbanded and just update the current state of the crew.
- In Fact we should not delete any of the entries from pirates/pirate crew/village/marine/marine base etc. because all of these entities might have some entries corresponding to them for raids and marine operations so if we delete these entries then we will also have to delete all the dependent entries of raids and marine operations. But we don't want to delete all this data about all the raids and operations that have been done so we prefer not to delete all this data but rather keep a status attribute for each of the entity types that would keep track of whether the entity is currently active or not.

Final ER-Diagram based on above modifications



https://app.diagrams.net/#G114B0VmQf9AS6S8YHhDu43whgJynMN2oX