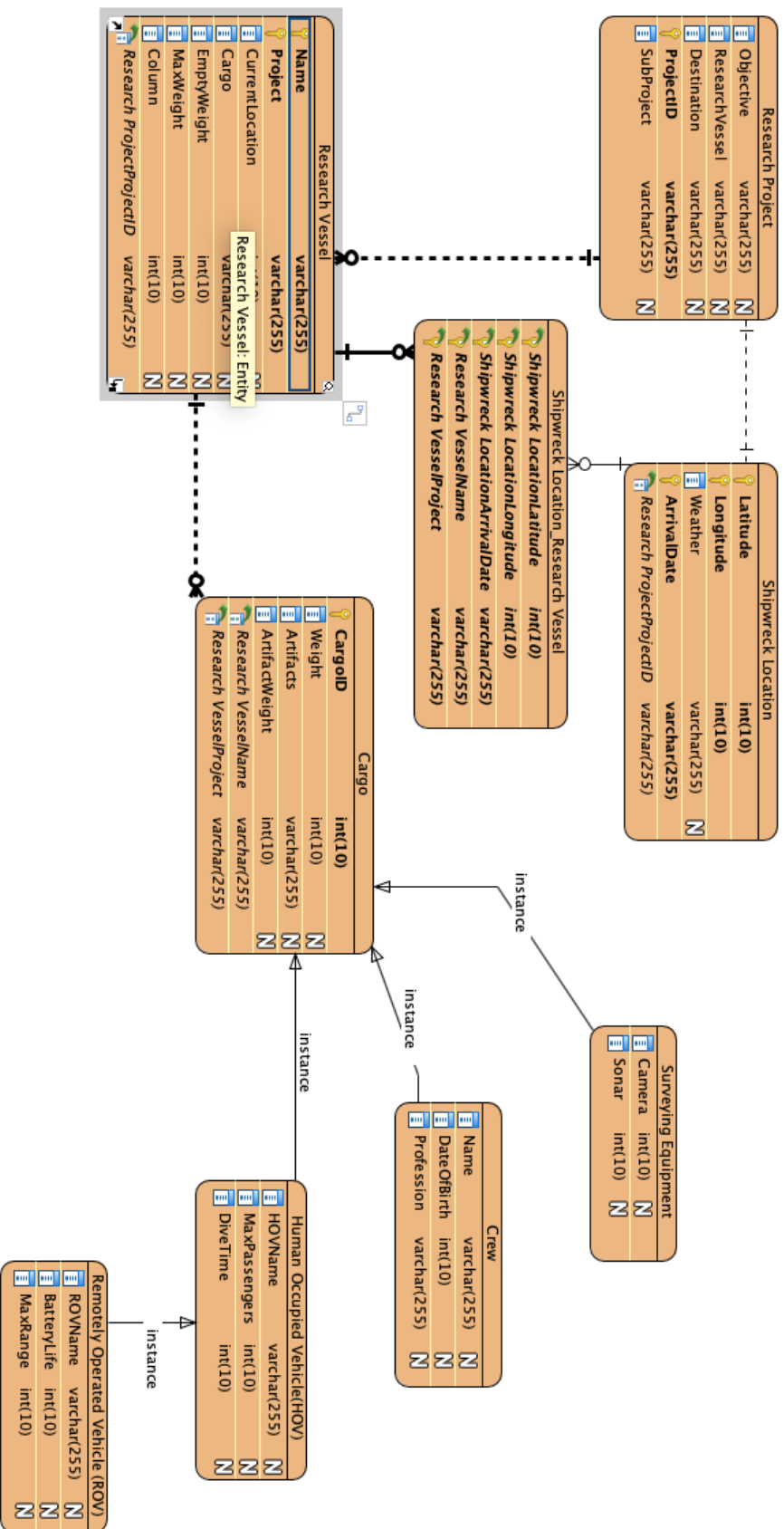


Ship Wreck ER Model

CS5800

DJ Wiley

A01099945



## **Model Overview**

I decided to do my ER model on researching and exploring a shipwreck similar to how the Titanic was researched in the 1980's. I used the model with the idea of research project in mind. The Project being the overall objective that the exploration will aim to accomplish. The Project has two major components: the data gathered, and the resources used to accomplish this objective. The resources used on this project will be various ocean transportation. The data gathered will also be measured in artifacts retrieved from the shipwreck.

## **Entity Types**

### **Research Project**

The central objective of the voyage.

- Objective: A string describing the project's goals in detail. Must be non-null.
- ResearchVessel: A set of watercraft that is used to transport the various materials required for the project.
- Destination: A set of locations to which the vessels and overall project are directed to in order to gather data.
- ProjectID: A unique name given to the project so that all of the data and artifacts collected may be organized into one main area. The name will be a string with the name of the ship being researched and also the year in which the research took place in case the same ship is explored multiple times.
- SubProject: A string describing various subprojects that could be going on at the same time. Projects like amount of pollutants being leaked into the ocean from the shipwreck, artifacts, remains gathered, etc.

The ProjectID was created as the primary key because it is reasonable to assume there could be multiple projects using the same vessels and general objectives. It also makes sense to organize all of the research materials to a single overall project.

### **Research Vessel**

An abstract representation of the main vessel used in the project to transport materials to the shipwreck location.

- Name: The name of the vessel. Must be a unique string and non-null. Part of the primary key.
- Project: The project to which the vessel is assigned. Inherits this from the Research Project entity
- CurrentLocation: Where the vessel is currently located in the ocean. This is a set of integers meaning latitude and longitude.
- Cargo: A set of all of the important cargo aboard the vessel.
- EmptyWeight: The weight of the vessel while holding no Cargo. Must be non-null, non-negative. This weight plus the weight of the cargo must not be larger than the max weight of the vessel.

- MaxWeight: The maximum amount of weight the vessel can hold and still float and travel at optimal speed. This value must be non-null, non-negative, and larger than the empty weight of the vessel.

The primary key of the vessel was chosen to be the Name and the Project because it is assumed that the vessel will be used for many different research projects over time. This helps identify which specific research project the data belongs to.

### **Shipwreck Location**

An abstraction representing the location in the ocean where the research vessel is to arrive at.

- Latitude: a non-null, integer representing the latitude of the shipwreck location.
- Longitude: a non-null, integer representing the longitude of the shipwreck location.
- Weather: a non-null, string representing the current projected weather at the location to help determine if the vessel can stay there.
- ArrivalDate: a non-null, string representing the date that the research vessel arrives at the location.

The Latitude, Longitude, and arrival date are the primary keys to help identify which shipwreck on which date.

### **Cargo**

An abstract representation of the people and objects that ride on the Research Vessel to the shipwreck location.

- CargoID: A unique non-null, non-negative integer representing each cargo entity. All cargo and personnel will be given this ID number as they are assigned to the vessel. This will be the primary key for the Cargo.
- Weight: The total weight in kilograms of all the initial items in the Cargo.
- Artifacts: A unique non-null, non-negative integer cataloging each artifact attribute.
- ArtifactWeight: The weight of all the additional Cargo taken aboard in kilograms.

Cargo primary keys are specialized in the children associated with this entity.

### **Surveying Equipment**

An instance of Cargo. This data represents the different surveying equipment used to analyze a shipwreck.

- Camera: a unique, non-null, non-negative integer given to each camera to help keep catalog of what equipment is used and what equipment will contain which data.
- Sonar: a unique, non-null, non-negative integer given to each sonar to help keep catalog of what equipment is used and what equipment will contain the data.

The primary key is inherited from the Cargo entity.

### **Crew**

An instance of Cargo. This data represents the personal information about any of the crew that will be on board the research vessel.

- Name: The full name of each crew member. Must be non-null

- DateOfBith: A date representing when the crew member was born.
- Profession: A string describing the crew member's purpose onboard. This attribute may have multiple values.

The primary key is inherited from the Cargo entity.

### **Human Occupied Vehicle (HOV)**

An instance of Cargo. This vehicle will be used to journey down to the shipwreck like a mini submarine, in order to gather information

- HOVName: a unique string that gives the HOV a specific name to help catalog the piece of cargo.
- MaxPassengers: a non-null, non-negative, integer representing how many crew members can go down in the HOV.
- DiveTime: a non-null, non-negative, integer representing how much time in minutes the vessel can be submerged

This inherits a primary key from Cargo.

### **Remotely Operated Vehicle (ROV)**

An instance of HOV. This is used in conjuncture with the HOV to help explore smaller areas of the shipwreck via remote control from someone in the HOV.

- ROVName: a unique string that gives the ROV a specific name to help catalog data gathered.
- BatteryLife: a non-null, non-negative, integer representing how much time in minutes the vessel can last without being recharged.
- MaxRange: a non-null, non-negative, integer representing how far in meters, the ROV may journey from the HOV before the cable length tethering it to the HOV runs out.

This inherits a primary key from the HOV

## **Relationship Types**

### **Project : Research Vessel**

Exactly one Project is related to at least one Research Vessel. There may be more than one vessel at a particular research site. This relationship is one-to-many since it is possible to have more than one vessel on a particular project. Since the research vessel does not have ownership over the project, a separate project entity must be created if a research vessel is to serve two separate projects. Participation must be total for each research vessel because we won't store data about vessels that aren't used in the project.

### **Project : Shipwreck Location**

Exactly one Project is related to a single shipwreck location. This relationship is one-to-one since each shipwreck will be unique in what we want to gather from the site. Anytime a new shipwreck location entity is created, a new project entity must also be created to serve that specific purpose. Participation must be total on the side of the shipwreck site since we should not keep a list of destinations not used in the project.

### **Research Vessel : Cargo**

A Research Vessel contains many different Cargo instances. But Cargo can only be physically located on one Research Vessel. So the relationship is many-to-one. There is a total participation on the side of Cargo since each cargo must be assigned to a vessel in order to complete a project.

### **Research Vessel : Shipwreck Location**

Many Research Vessels could be at a shipwreck location. It is feasible to have a vessel that can journey to several shipwreck locations, and the locations can have more than one vessel. This relationship is many-to-many

### **Shipwreck Location : Shipwreck Location**

All locations must be specified in a particular order. A research vessel will always have a port to return to and various locations in which research projects will be conducted on. So this is a one-to-one reflexive relationship with total participation.

### **Research Vessel : Cargo => Shipwreck Location**

There's a ternary relationship here describing which combinations of research vessels and cargo may be at a location

### **Cargo => Crew**

Crew inherits from Cargo and is a weak entity.

### **Cargo => Surveying Equipment**

Surveying Equipment inherits from Cargo and is a weak entity.

### **Cargo => HOV**

HOV inherits from Cargo and is a weak entity.

### **HOV => ROV**

ROV inherits from HOV and is a weak entity.

## **List of Deliverables**

### A many-one relationship type with total participation on the one side

This is modeled by the relationship between a Research Vessel and Cargo. A Research Vessel contains many different Cargo instances. But Cargo can only be physically located on one Research

Vessel. So the relationship is many-to-one. There is a total participation on the side of Cargo since each cargo must be assigned to a vessel in order to complete a project.

#### A many-many relationship type

This is modeled by Research Vessels at a shipwreck location. Many Research Vessels could be at a shipwreck location. It is feasible to have a vessel that can journey to several shipwreck locations, and the locations can have more than one vessel. This relationship is many-to-many

#### A ternary relationship

This is modeled by the relationship between the Research Vessel, Cargo, and the Shipwreck Location. This relationship describes which vessel and cargo pair are at a particular location.

#### A disjoint inheritance hierarchy

This is modeled by The ROV inheritance. The ROV is part of the HOV, but if an object is an HOV it does not mean it is an ROV or vice versa.

#### An overlapping inheritance hierarchy

This is modeled in several places by the inheritance from Cargo. If an entity is a member of the Crew then they are also Cargo. If an entity is Survey Equipment, then it is also Cargo.

#### A composite attribute

The CurrentLocation attribute of a research vessel is composite. It takes both latitude and longitude coordinates.

#### A multivalued attribute

A member of the crew may have multiple values for the Profession attribute.

#### A one-one relationship type with total participation on at least one side

This is modeled by the relationship between Project and Shipwreck. Exactly one Project is related to a single shipwreck location. This relationship is one-to-one since each shipwreck will be unique in what we want to gather from the site. Anytime a new shipwreck location entity is created, a new project entity must also be created to serve that specific purpose. Participation must be total on the side of the shipwreck site since we should not keep a list of destinations not used in the project.

#### A one-many relationship type with partial participation on both sides

This is modeled by there being a Project on a Research Vessel. Only one project can be on a single vessel, but there could many projects on many vessels. There is no requirement for a project to be on a vessel or for a vessel to have a project.

### A relationship with an attribute(s)

There is a relationship between a Research Vessel and its associated Cargo contains an attribute, the weight of all the Cargo.

### A weak-entity type

The Crew is an example of a weak-entity. This is because they can be identified by a primary key composing of Name, DOB, completed by the CargoID they are assigned.

### A reflexive relationship type

Shipwreck location has a reflexive relationship so that the order of each shipwreck location must be specified in order to know which one to travel to first.

## **Physical Schema**

```
CREATE TABLE Cargo (CargoID int(10) NOT NULL AUTO_INCREMENT, Weight int(10), Artifacts varchar(255), ArtifactWeight int(10), `Research VesselName` varchar(255) NOT NULL, `Research VesselProject` varchar(255) NOT NULL, PRIMARY KEY (CargoID));
```

```
CREATE TABLE Crew (Name varchar(255), DateOfBirth int(10), Profession varchar(255));
```

```
CREATE TABLE `Human Occupied Vehicle(HOV)` (HOVName varchar(255), MaxPassengers int(10), DiveTime int(10));
```

```
CREATE TABLE `Remotely Operated Vehicle (ROV)` (ROVName varchar(255), BatteryLife int(10), MaxRange int(10));
```

```
CREATE TABLE `Research Project` (Objective varchar(255), ResearchVessel varchar(255), Destination varchar(255), ProjectID varchar(255) NOT NULL, SubProject varchar(255), PRIMARY KEY (ProjectID));
```

```
CREATE TABLE `Research Vessel` (Name varchar(255) NOT NULL, Project varchar(255) NOT NULL, CurrentLocation int(10), Cargo varchar(255), EmptyWeight int(10), MaxWeight int(10), `Column` int(10), `Research ProjectProjectID` varchar(255) NOT NULL, PRIMARY KEY (Name, Project));
```

```
CREATE TABLE `Shipwreck Location` (Latitude int(10) NOT NULL, Longitude int(10) NOT NULL, Weather varchar(255), ArrivalDate varchar(255) NOT NULL, `Research ProjectProjectID` varchar(255) NOT NULL, PRIMARY KEY (Latitude, Longitude, ArrivalDate));
```

```
CREATE TABLE `Shipwreck Location_Research Vessel` (`Shipwreck LocationLatitude` int(10) NOT NULL, `Shipwreck LocationLongitude` int(10) NOT NULL, `Shipwreck LocationArrivalDate` varchar(255) NOT NULL, `Research VesselName` varchar(255) NOT NULL, `Research VesselProject` varchar(255) NOT NULL, PRIMARY KEY (`Shipwreck LocationLatitude`, `Shipwreck LocationLongitude`, `Shipwreck LocationArrivalDate`, `Research VesselName`, `Research VesselProject`));
```

```
CREATE TABLE `Surveying Equipment` (Camera int(10), Sonar int(10));
```



```
ALTER TABLE `Shipwreck Location` ADD CONSTRAINT `FKShipwreck 163683` FOREIGN KEY (`Research ProjectProjectID`) REFERENCES `Research Project` (ProjectID);
```

```
ALTER TABLE `Research Vessel` ADD CONSTRAINT `FKResearch V24879` FOREIGN KEY (`Research ProjectProjectID`) REFERENCES `Research Project` (ProjectID);
```

```
ALTER TABLE Cargo ADD CONSTRAINT FKCargo788468 FOREIGN KEY (`Research VesselName`, `Research VesselProject`) REFERENCES `Research Vessel` (Name, Project);
```

```
ALTER TABLE `Shipwreck Location_Research Vessel` ADD CONSTRAINT `FKShipwreck 465959` FOREIGN KEY (`Shipwreck LocationLatitude`, `Shipwreck LocationLongitude`, `Shipwreck LocationArrivalDate`) REFERENCES `Shipwreck Location` (Latitude, Longitude, ArrivalDate);
```

```
ALTER TABLE `Shipwreck Location_Research Vessel` ADD CONSTRAINT `FKShipwreck 288654` FOREIGN KEY (`Research VesselName`, `Research VesselProject`) REFERENCES `Research Vessel` (Name, Project);
```