**IMPROVEMENT OF PORTAL FOR ASSET AND NAVIGATIONAL INFORMATION(PANI) PORTAL**

**SUBJECT : DATABASE MANAGEMENT SYSTEM**

**ABSTRACT**

PANI is an integrated solution brining river navigation and infrastructure information on a single platform. It provides detailed information of various features of National Waterways and the assets such as fairway, infrastructure facilities, cross-river structures, connectivity at jetties, emergency services for facilitating transportation of cargo. The GIS-based Bharat Map portal also helps external stakeholders in voyage planning, leading to ease of business. A mobile application has also been prepared which is available on Google Play Store for easy dissemination of information and tracking of vessels This abstract suggests enhancements for the PANI Portal, focusing on improving the user interface, data visualization, real-time data integration, personalization, collaboration features, integration with external systems, and continuous updates and maintenance. These enhancements will make the portal more user-friendly and efficient, enabling better navigation planning and cargo transportation on river waterways.

.

**REQUIREMENTS ANALYSIS**

**List Of Tables:**

Overall 5 tables are required for the ER model representation,they are:

1.users

2.vessels

3.ports

4.navigation

5.weather

**List Of Attributes:**

**1.Users Table:**

Attributes:

-user\_id

-user\_name

-email

-password

-access\_level

Domain Types:

* User\_id: NUMBER
* User\_name:VARCHAR
* email :VARCHAR
* password:VARCHAR
* access\_level:VARCHAR

Constraints:

* Primary key:User\_id

**2.Vessels Table:**

Attributes:

-vessel\_id

-name

-Imo\_number

-length

-width

-draft

Domain Types:

* vessel\_id: NUMBER
* name: VARCHAR
* Imo\_number:VARCHAR
* length:float
* width:float
* draft:float

Constraints:

* Primary\_key: vessel\_id

**3.Ports table:**

Attributes:

-port\_id

-name

Domain Types:

* port\_id: NUMBER
* name: VARCHAR

Constraints:

* Primary key:port\_id

**4.Navigation Table:**

Attributes:

-navigation\_id

-vessel\_id

-position\_x

-position\_y

-heading

-speed

-ETA

Domain Types:

* Navigation\_id : NUMBER
* Vessel\_id:VARCHAR
* Position\_x:float
* Position\_y:float
* Heading:float
* Speed:float
* ETA:TIMESTAMP

Constraints:

* Primary key:navigation\_id,vessel\_id

**5.Weather Table:**

Attributes:

-weather\_id

-position\_x

-position\_y

-temperature

-wind\_speed

-precipitation

-location

-capacity

Domain Types:

* weather\_id : NUMBER
* Position\_x : float
* Position\_y : float
* Temperature : float
* Wind\_speed : float
* Precipitation : float
* location:VARCHAR
* capacity:VARCHAR

Constraints:

* Primary Key : weather\_id

**MAPPING CARDINALITIES:**

One-to-One (1:1) Mapping:

* Each vessel in the system is assigned a unique IMO number.
* Each user account is associated with a single email address.

One-to-Many (1:N) Mapping:

* Each vessel may have multiple navigation records associated with it.
* Each port may have multiple vessels docked at it at the same time.

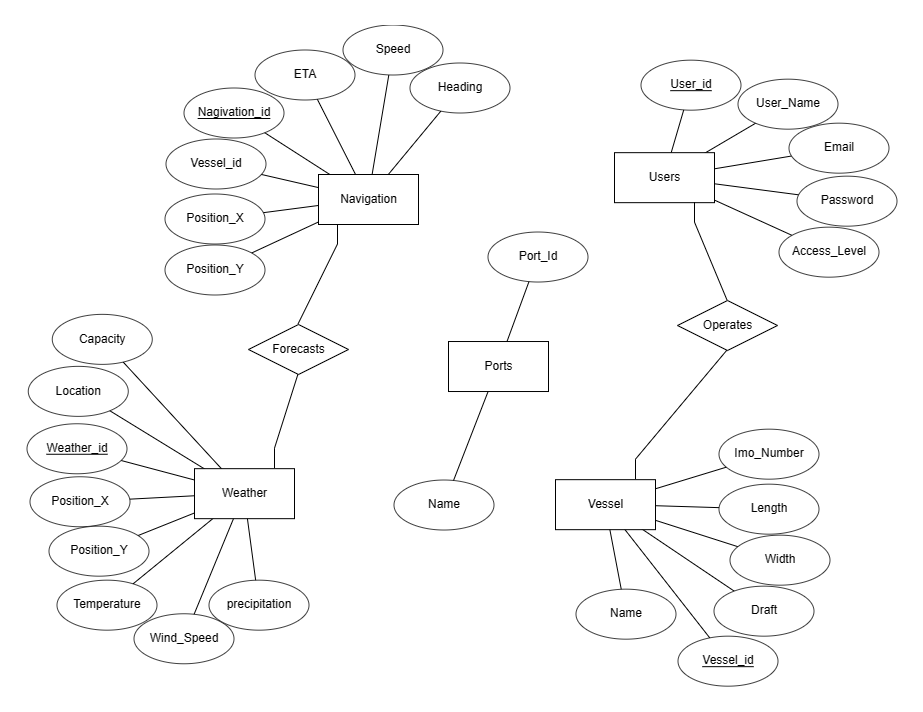
Many-to-One (N:1) Mapping:

* Multiple navigation records may be associated with a single vessel.
* Multiple weather records may be associated with a single geographic location.

Many-to-Many (N:N) Mapping:

* A vessel may be associated with multiple ports over time, and a port may have multiple vessels visiting it over time.

**ER DIAGRAM :**



**DDL COMMANDS:**

**1.USERS TABLE:**

CREATE TABLE users (

user\_id INT PRIMARY KEY ,

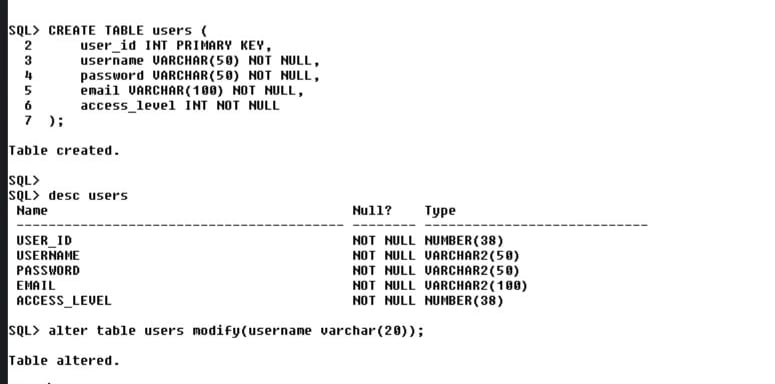
username VARCHAR(50) NOT NULL,

password VARCHAR(50) NOT NULL,

email VARCHAR(100) NOT NULL,

access\_level INT NOT NULL

);



**2.VESSELS TABLE:**

CREATE TABLE vessels (

vessel\_id INT PRIMARY KEY ,

name VARCHAR(100) NOT NULL,

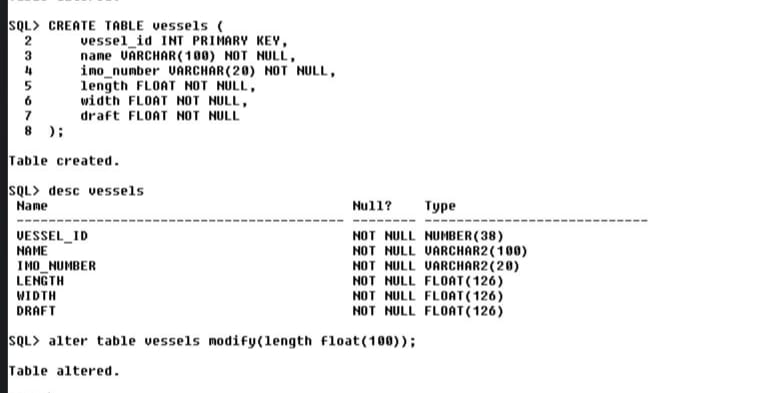
imo\_number VARCHAR(20) NOT NULL,

length FLOAT NOT NULL,

width FLOAT NOT NULL,

draft FLOAT NOT NULL

);



**3.PORTS TABLE:**

CREATE TABLE ports (

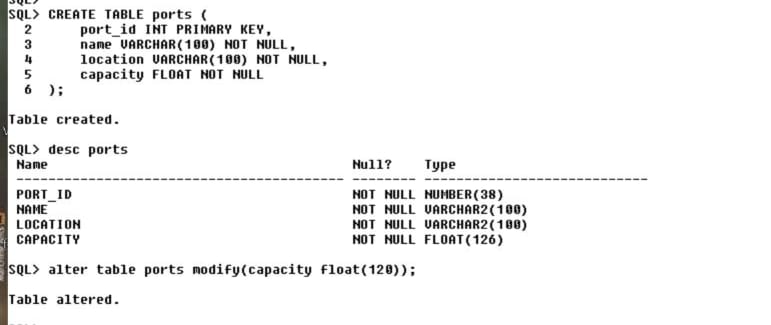
port\_id INT PRIMARY KEY ,

name VARCHAR(100) NOT NULL,

location VARCHAR(100) NOT NULL,

capacity FLOAT NOT NULL

);



**4.NAVIGATION TABLE:**

CREATE TABLE navigation (

navigation\_id INT PRIMARY KEY ,

vessel\_id INT NOT NULL,

position\_x FLOAT NOT NULL,

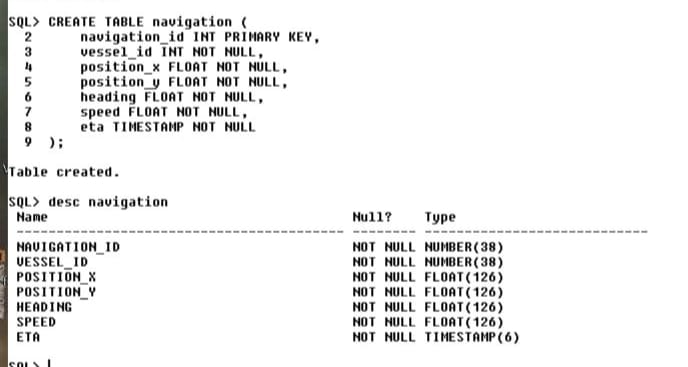
position\_y FLOAT NOT NULL,

heading FLOAT NOT NULL,

speed FLOAT NOT NULL,

eta TIMESTAMP NOT NULL

);



**5.WEATHER TABLE**:

CREATE TABLE weather (

weather\_id INT PRIMARY KEY AUTO\_INCREMENT,

position\_x FLOAT NOT NULL,

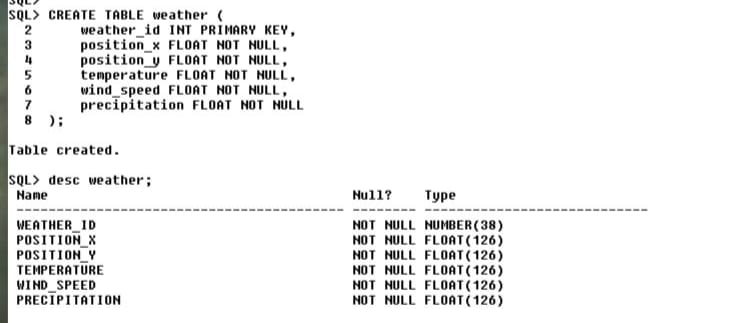
position\_y FLOAT NOT NULL,

temperature FLOAT NOT NULL,

wind\_speed FLOAT NOT NULL,

precipitation FLOAT NOT NULL

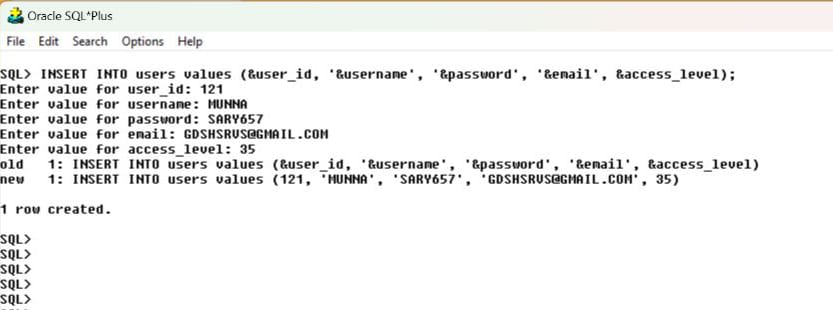
);



**DML COMMANDS:**

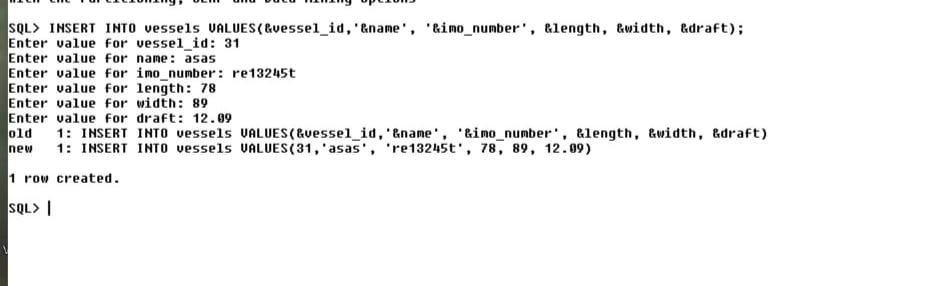
**1.Users table:**

INSERT INTO users values (&user\_id, '&username', '&password', '&email', &access\_level);



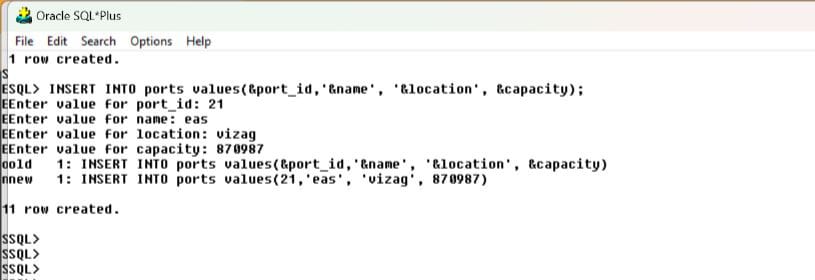
**2.VESSELS TABLE:**

INSERT INTO vessels VALUES(&vessel\_id,'&name', '&imo\_number', &length, &width, &draft);



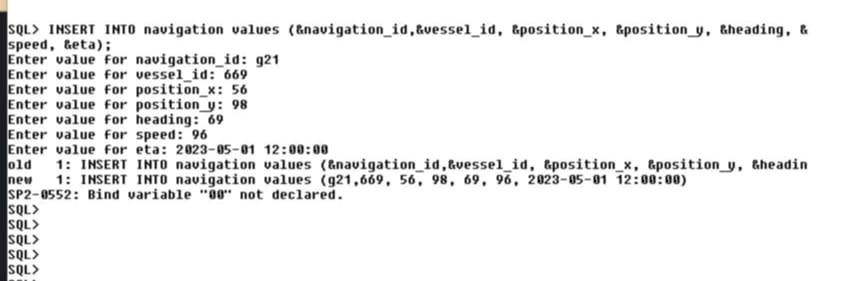
**3.PORTS TABLE:**

INSERT INTO ports values(&port\_id,'&name', '&location', &capacity);



**4.NAVIGATION TABLE:**

INSERT INTO navigation values (&navigation\_id,&vessel\_id, &position\_x, &position\_y, &heading, &speed);



**5.WEATHER TABLE:**

INSERT INTO weather values (&weather\_id,&position\_x, &position\_y, &temperature, &wind\_speed, & precipitation);

