

Project Title:

Common Platform For Ports, Terminals And Other Custodians.

ABSTRACT

Ports and terminals are critical components of the global supply chain, connecting producers, suppliers, and consumers across the world. However, managing ports and terminals is a complex and challenging task, involving multiple stakeholders with their own processes and systems. To address these challenges, a common platform for ports, terminals, and other custodians is proposed. This platform aims to streamline the flow of cargo and vessel movements through the port ecosystem, providing real-time visibility and tracking, improving collaboration and communication among stakeholders, and enhancing data accuracy, integrity, and security. The platform will be developed using the latest technologies, including cloud computing, big data analytics, artificial intelligence, and blockchain, and will be designed to be scalable, flexible, and adaptable. The project has the potential to transform the port ecosystem, improving efficiency, productivity, and security, and delivering significant benefits to all stakeholders.

DESIGN REQUIREMENTS:

- 1) Vessel Table
- 2) Cargo Table
- 3) Booking Table
- 4) User Table
- 5) Event Table

ATTRIBUTES AND DOMAIN TYPES:

Vessel table:

VesselID: integer (primary key)

VesselName: varchar(50)

RegistrationNumber: varchar(20)

VesselSize: integer

VesselType: varchar(20)

Cargo table:

CargoID: integer (primary key)

CargoType: varchar(50)

CargoWeight: decimal(10,2)

CargoVolume: decimal(10,2)

Destination: varchar(50)

Booking table:

BookingID: integer (primary key)

CargoID: integer (foreign key to Cargo table)

VesselID: integer (foreign key to Vessel table)

BookingTime: datetime

ExpectedArrivalTime: datetime

ExpectedDepartureTime: datetime

User table:

UserID: integer (primary key)

Username: varchar(50)

Password: varchar(50)

Role: varchar(20)

Event table:

EventID: integer (primary key)

EventType: varchar(50)

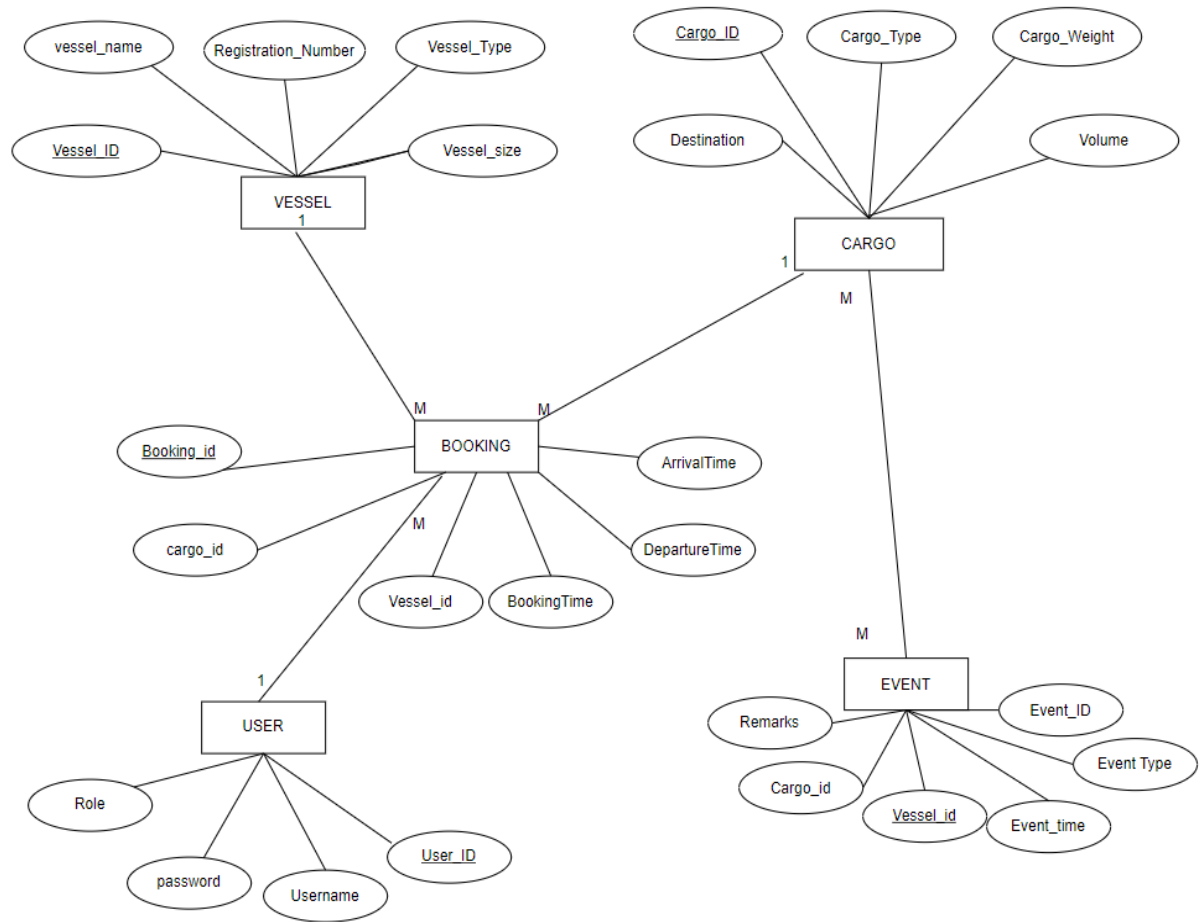
EventTime: datetime

VesselID: integer (foreign key to Vessel table)

CargoID: integer (foreign key to Cargo table)

Remarks: varchar(200)

ER DIAGRAM



RELATIONAL MODEL and DML Operations

Vessel Table:

```
SQL> select * from Vessel;
```

VESSELID	VESSELNAME	REGISTRATI	VESSELSIZE	VESSELTYPE
1	Ship A	ABC123	100	Cargo
2	Ship B	DEF456	150	Passenger
3	Ship C	GHI789	200	Cargo
4	Ship D	JKL012	250	Passenger
5	Ship E	MNO345	300	Cargo

Query:

Create:

```
SQL> spool C:\dbms_files\spool1.txt
SQL> CREATE TABLE Vessel (
  2     VesselID INTEGER PRIMARY KEY,
  3     VesselName VARCHAR(50),
  4     RegistrationNumber VARCHAR(20),
  5     VesselSize INTEGER,
  6     VesselType VARCHAR(20)
  7 );
```

Table created.

Insert:

```
SQL> INSERT INTO Vessel (VesselID, VesselName, RegistrationNumber, VesselSize, VesselType)
  2 values (&VesselID, '&VesselName', '&RegistrationNumber', &VesselSize, '&VesselType');
Enter value for vesselid: 3
Enter value for vesselname: Ship C
Enter value for registrationnumber: GHI789
Enter value for vesselsize: 200
Enter value for vesseltype: Cargo
old 2: values (&VesselID, '&VesselName', '&RegistrationNumber', &VesselSize, '&VesselType')
new 2: values (3, 'Ship C', 'GHI789', 200, 'Cargo')

1 row created.
```

Cargo Table:

```
SQL> select * from cargo;
```

CARGOID	CARGOTYPE	CARGOWEIGHT	CARGOVOLUME	DESTINATION
1	Furniture	500	100	Hyderabad
2	Electronics	250	50	Tirupati
3	Clothing	100	25	Mumbai
4	Food	150	50	Vijayawada
5	Books	50	10	Warangal

Query

Create

```
SQL> CREATE TABLE Cargo (  
2     CargoID INTEGER PRIMARY KEY,  
3     CargoType VARCHAR(50),  
4     CargoWeight DECIMAL(10,2),  
5     CargoVolume DECIMAL(10,2),  
6     Destination VARCHAR(50)  
7 );
```

Table created.

Insert:

```
SQL> INSERT INTO Cargo (CargoID, CargoType, CargoWeight, CargoVolume, Destination)  
2 values(&CargoID, '&CargoType', &CargoWeight, &CargoVolume, '&Destination');  
Enter value for cargoid: 1  
Enter value for cargotype: Furniture  
Enter value for cargoweight: 500.00  
Enter value for cargovolume: 100.00  
Enter value for destination: Hyderabad  
old 2: values(&CargoID, '&CargoType', &CargoWeight, &CargoVolume, '&Destination')  
new 2: values(1, 'Furniture', 500.00, 100.00, 'Hyderabad')  
  
1 row created.
```

Booking Table:

```
SQL> select * from Booking;
```

BOOKINGID	CARGOID	VESSELID
-----------	---------	----------

BOOKINGTIME

EXPECTEDARRIVALTIME

EXPECTEDDEPARTURETIME

	1	1	1
01-MAY-23	09.00.00.000000	AM	
10-MAY-23	12.00.00.000000	PM	
01-MAY-23	10.00.00.000000	AM	

BOOKINGID	CARGOID	VESSELID
-----------	---------	----------

BOOKINGTIME

EXPECTEDARRIVALTIME

EXPECTEDDEPARTURETIME

	2	2	1
02-MAY-23	12.00.00.000000	PM	
12-MAY-23	03.00.00.000000	PM	
02-MAY-23	01.00.00.000000	PM	

BOOKINGID	CARGOID	VESSELID
-----------	---------	----------

BOOKINGTIME

EXPECTEDARRIVALTIME

EXPECTEDDEPARTURETIME

	3	3	2
03-MAY-23	03.00.00.000000	PM	
13-MAY-23	06.00.00.000000	PM	
03-MAY-23	04.00.00.000000	PM	

Query:

```
SQL> CREATE TABLE Booking (  
2     BookingID INTEGER PRIMARY KEY,  
3     CargoID INTEGER,  
4     VesselID INTEGER,  
5     BookingTime TIMESTAMP DEFAULT TO_TIMESTAMP('2022-05-01 10:30:00', 'YYYY-MM-DD HH24:MI:SS'),  
6     ExpectedArrivalTime TIMESTAMP,  
7     ExpectedDepartureTime TIMESTAMP,  
8     FOREIGN KEY (CargoID) REFERENCES Cargo(CargoID),  
9     FOREIGN KEY (VesselID) REFERENCES Vessel(VesselID)  
10 );
```

Table created.

```
SQL> INSERT INTO Booking (BookingID, CargoID, VesselID, BookingTime, ExpectedArrivalTime, ExpectedDepartureTime)  
2 VALUES  
3 (1, 1, 1, TO_TIMESTAMP('2023-05-01 09:00:00', 'YYYY-MM-DD HH24:MI:SS'), TO_TIMESTAMP('2023-05-10 12:00:00', 'YYYY-MM-DD HH24:MI:SS'), TO_TIMESTAMP('2023-05-01 10:00:00', 'YYYY-MM-DD HH24:MI:SS'));  
1 row created.
```

USERS TABLE:

```
SQL> select * from users;
```

USERID	USERNAME	PASSWORD	ROLE
1	user1	\$Y.WYD4FvJpZS5qlgX9o5q	customer
2	user2	\$lmdfngdsbZS5qlgX9o5q	customer
3	admin	\$lmdfngdsbZS5qlgX9o5q	admin

Query:

```
Table created.  
  
SQL> CREATE TABLE User (  
2     UserID INTEGER PRIMARY KEY,  
3     Username VARCHAR(50),  
4     Password VARCHAR(50),  
5     Role VARCHAR(20)  
6 );  
CREATE TABLE User (
```


EVENT TABLE:

```
EVENTID EVENTTYPE
-----
EVENTTIME
-----
VESSELID  CARGOID REMARKS
-----
      1 Departure
01-MAY-23 09.00.00.000000 AM
      1          1 Cargo loaded onto ship

      3 Departure
15-MAY-23 10.00.00.000000 AM
      2          2 Cargo loaded onto ship

EVENTID EVENTTYPE
-----
EVENTTIME
-----
VESSELID  CARGOID REMARKS
-----
      4 Arrival
20-MAY-23 04.00.00.000000 PM
      2          2 Ship arrived at destination port

      5 Departure
25-MAY-23 08.00.00.000000 AM

EVENTID EVENTTYPE
-----
EVENTTIME
-----
VESSELID  CARGOID REMARKS
-----
      3          3 Cargo loaded onto ship
```

```
SQL> CREATE TABLE Event (
2   EventID INTEGER PRIMARY KEY,
3   EventType VARCHAR(50),
4   EventTime TIMESTAMP DEFAULT TO_TIMESTAMP('2022-05-01 10:30:00', 'YYYY-MM-DD HH24:MI:SS'),
5   VesselID INTEGER,
6   CargoID INTEGER,
7   Remarks VARCHAR(200),
8   FOREIGN KEY (VesselID) REFERENCES Vessel(VesselID),
9   FOREIGN KEY (CargoID) REFERENCES Cargo(CargoID)
10 );
```

Table created.

```
SQL> INSERT INTO Event (EventID, EventType, EventTime, VesselID, CargoID, Remarks)
2  VALUES (4, 'Arrival', TO_TIMESTAMP('2023-05-20 16:00:00', 'YYYY-MM-DD HH24:MI:SS'), 2, 2, 'Ship arrived at destination port');
```

1 row created.

```
SQL> INSERT INTO Event (EventID, EventType, EventTime, VesselID, CargoID, Remarks)
2  VALUES (5, 'Departure', TO_TIMESTAMP('2023-05-25 08:00:00', 'YYYY-MM-DD HH24:MI:SS'), 3, 3, 'Cargo loaded onto ship');
```

1 row created.

DELETION

```
SQL> select * from users;
```

USERID	USERNAME	PASSWORD	ROLE
1	user1	\$Y.WYD4FvJpZS5qlgX9o5q	customer
2	user2	\$lmdfngdsbZS5qlgX9o5q	customer
3	admin	\$lmdfngdsbZS5qlgX9o5q	admin

```
SQL> DELETE FROM Users WHERE UserID = 2;
```

```
1 row deleted.
```

```
SQL> select * from users;
```

USERID	USERNAME	PASSWORD	ROLE
1	user1	\$Y.WYD4FvJpZS5qlgX9o5q	customer
3	admin	\$lmdfngdsbZS5qlgX9o5q	admin