CI7320 – Databases and Data Management
Coursework 1

Kingston University

Kantheti Dhana Shravya - K2247431

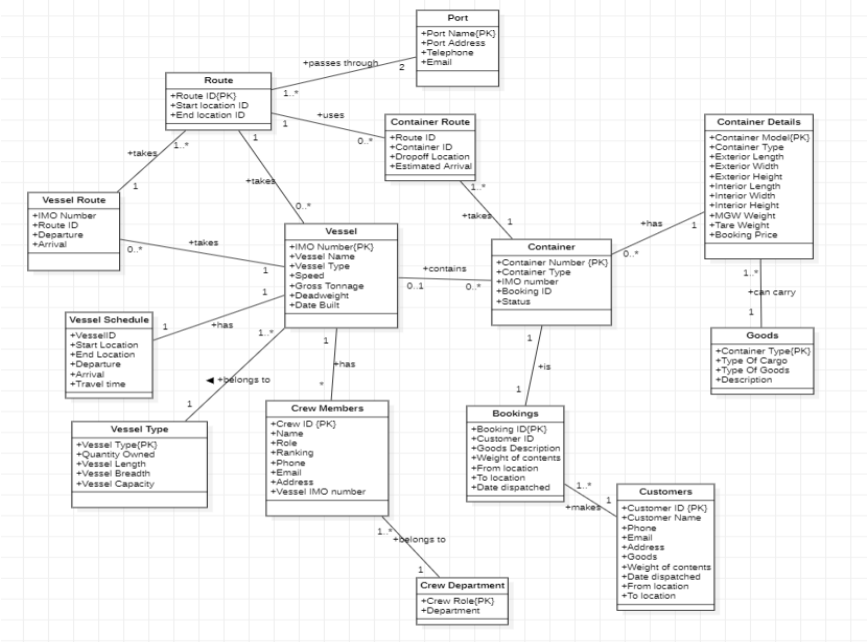
Table of Contents

I. INTRODUCTION

The aim of this coursework is to design and implement a database system that will help Everblue Ocean Express to efficiently track and maintain vessel schedules to allocate the containers in their vessel containers efficiently. Different entities are identified and relationships between the identified entities are made. The design of the database is made in such a way that the company can keep track of the vessels, customers, and vessel schedules.

II. ER DIAGRAM

Figure 1 shows the designed ER Diagram for the database. Different entities are identified and are represented using classes. Relationships between entities are shown using associations and multiplicity and cardinality for each of the relationships are shown.



A. *Constraints*

The table 1 shows primary keys and foreign keys in the designed schema.

Table	Primary Key	Foreign Key	Foreign Key Reference Table
Vessel	IMO Number	Vessel Type	Vessel Type
Vessel Type	Vessel Type		
Vessel Container	Container ID	Container Type IMO number Booking ID	Container Details Vessel Bookings
Container Details	Container Model	Container Type	Goods
Goods	Container Type		
Customers	Customer ID		
Bookings	Booking ID	Customer ID	Customers
Route	Route ID	Start Location ID End Location ID	Ports
Ports	Port Name		
Crew Members	Crew ID	Vessel IMO Number Crew Role	Vessel Crew Department
Crew Department	Crew Role		

B. Assumptions:

Assumptions made to design the database for the given scenario are listed below:

Vessel and Vessel Route:

- Each vessel has a route that it takes, which is stored in the Vessel Route table along with the arrival and departure time at the start and end locations of the routes. The vessels are predefined with certain schedules, stored in vessel schedule and VesselRoute maps the intermediate stops it takes to deliver the cargo.
- A vessel carries multiple containers and can be allocated based on the capacity of the vessel and the weight of the container. A vessel is assigned one particular start and end location which consists of multiple break-out routes(intermediate routes) and each route can be taken by one or more vessels, therefore there are many to many relationships between vessel and route. To resolve this, another table is created that maps vessel routes with the IMO numbers of the vessels.
- Each vessel has different crew members working in 4 departments, therefore each vessel has at least 4 crew members in each department.
- All the vessels of the same vessel type have the same measurements like capacity, length etc.

Type of goods and Cargo:

- Type of goods and type of cargo that a container can carry is mapped in the 'Goods' table, from which different Container Details can be displayed to the customer. When the customer specifies the type of cargo that needs to be shipped, the system maps the type of goods to the type of container and displays different options or types of containers available to the customer.

Ports and routes:

- The company has only one port office in the given port locations, so Port Location is unique.
- The route table has the mapping of the port locations, representing routes between the ports.

Container and Routes:

- Container information is stored only when a booking is made.
- One or more containers are loaded into the vessel and they are dropped off at intermediate locations. To keep a track of it, another table is created that maps the container route with the existing routes.
- Net Weight can be calculated from MGW and tare weights using the formula MGW-Tare weight = Net Weight, therefore only two of them are stored in the database to reduce the memory usage.
- Container Tariff id is calculated and stored in the Customer table by calculating the number of days taken by the container to reach the destination and multiplied by the container tariff booked by the customer.

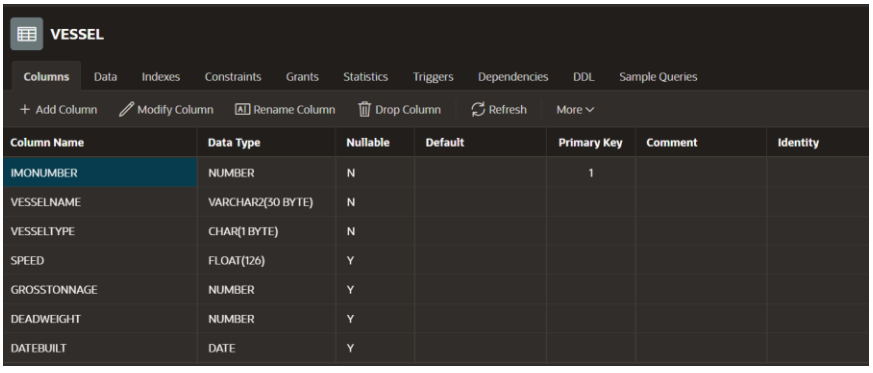
III. IMPLEMENTATION

SQL Queries to create the required tables:

- **Vessel**:

The vessel table has general information about the vessel like the date it was built, its speed, what type of vessel does it belong to.

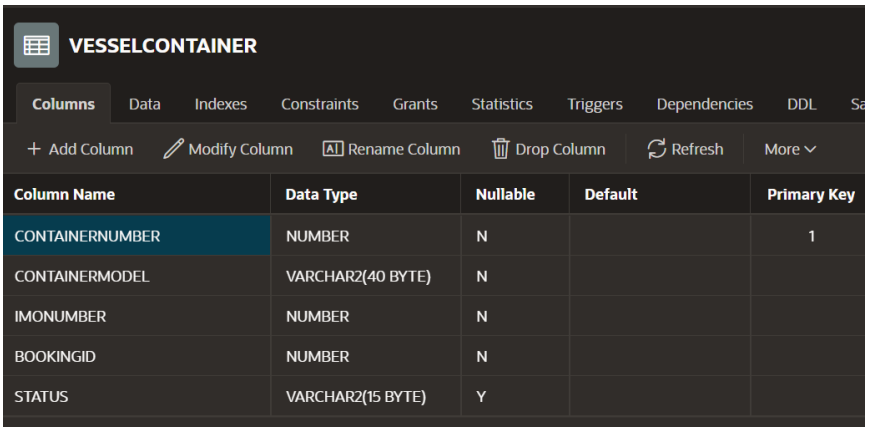
```
CREATE TABLE Vessel (
IMONumber number,
Vesselname varchar2(30) NOT NULL,
Vesseltype char NOT NULL,
Speed float,
Grosstonnage number,
Deadweight number,
DateBuilt date,
PRIMARY KEY (imonumber),
FOREIGN KEY(vesseltype) REFERENCES VesselType(vesseltype)
);
```



- **VesselContainer:**

Container table has the information about container capacity, what vessel is it in and under what booking ID the container is booked under.

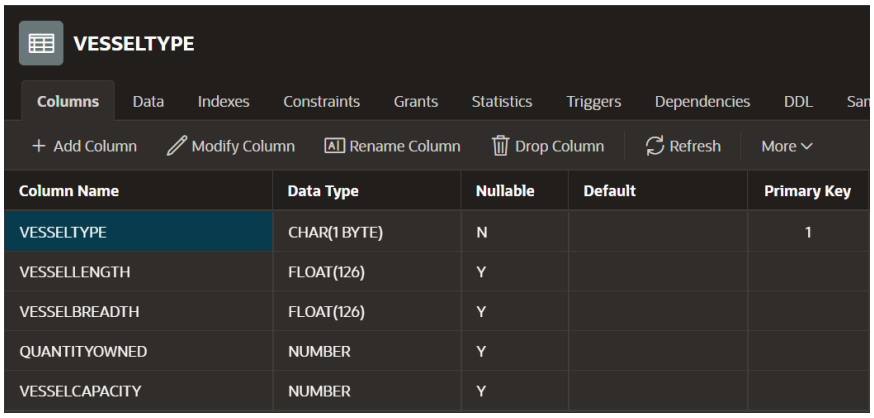
```
CREATE TABLE VesselContainer(
ContainerNumber number,
ContainerModel varchar(40) NOT NULL,
IMONumber number NOT NULL,
BookingID number NOT NULL,
Status varchar(15),
PRIMARY KEY(ContainerNumber),
FOREIGN KEY(ContainerModel) REFERENCES ContainerDetails(ContainerModel),
FOREIGN KEY(IMONumber) REFERENCES Vessel(IMONumber),
FOREIGN KEY(BookingID) REFERENCES Bookings(BookingID));
```



- **VesselType:**

VesselType stores the length,breadth and the number of vessels the company owns of each type and their capacity.

```
CREATE TABLE VesselType(
VesselType char,
VesselLength float,
VesselBreadth float,
QuantityOwned int,
VesselCapacity number,
PRIMARY KEY(VesselType)
);
```

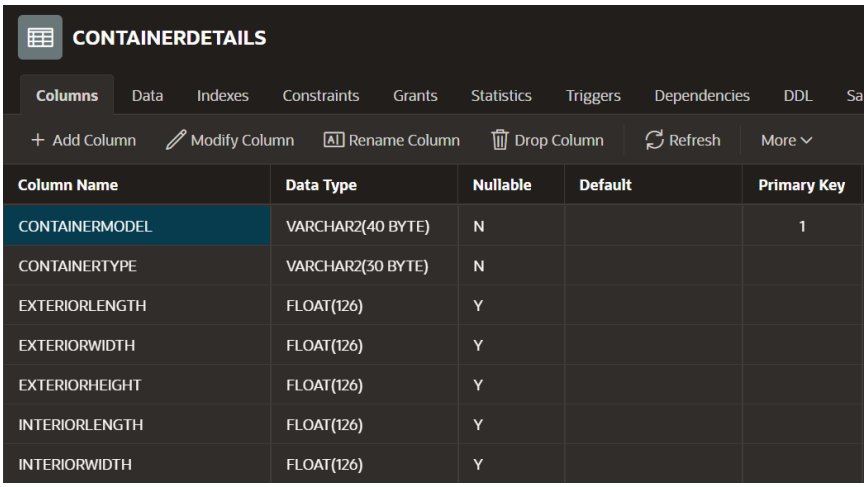


- **ContainerDetails:**

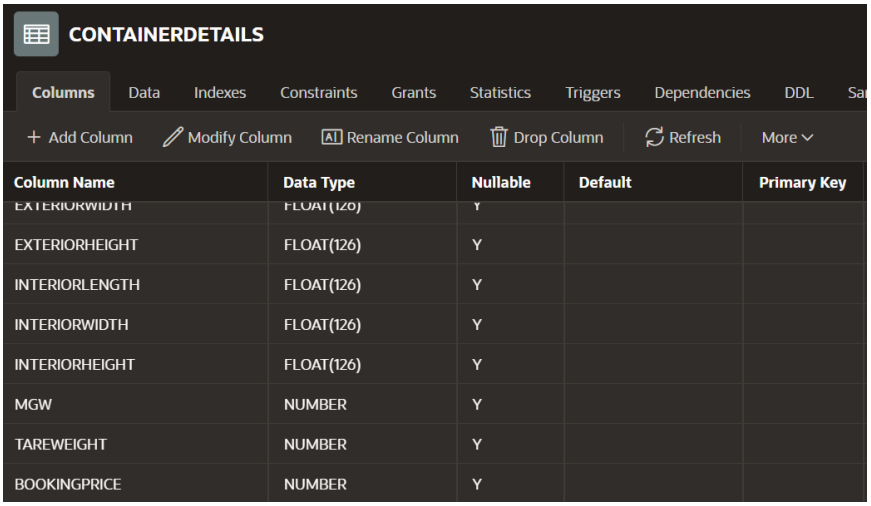
ContainerDetails table has information about the measurements of each type of container model that the company provides as a service and what type it belongs to.

```
CREATE TABLE ContainerDetails(
ContainerModel varchar(40),
ContainerType varchar(30) NOT NULL,
ExteriorLength float, ExteriorWidth float,
```

```
ExteriorHeight float, InteriorLength float,
InteriorWidth float, InteriorHeight float,
MGW number, TareWeight number,
BookingPrice number,
PRIMARY KEY(ContainerModel),
FOREIGN KEY(ContainerType) REFERENCES Goods(ContainerType));
```



Column Name	Data Type	Nullable	Default	Primary Key
CONTAINERMODEL	VARCHAR2(40 BYTE)	N		1
CONTAINERTYPE	VARCHAR2(30 BYTE)	N		
EXTERIORLENGTH	FLOAT(126)	Y		
EXTERIORWIDTH	FLOAT(126)	Y		
EXTERIORHEIGHT	FLOAT(126)	Y		
INTERIORLENGTH	FLOAT(126)	Y		
INTERIORWIDTH	FLOAT(126)	Y		

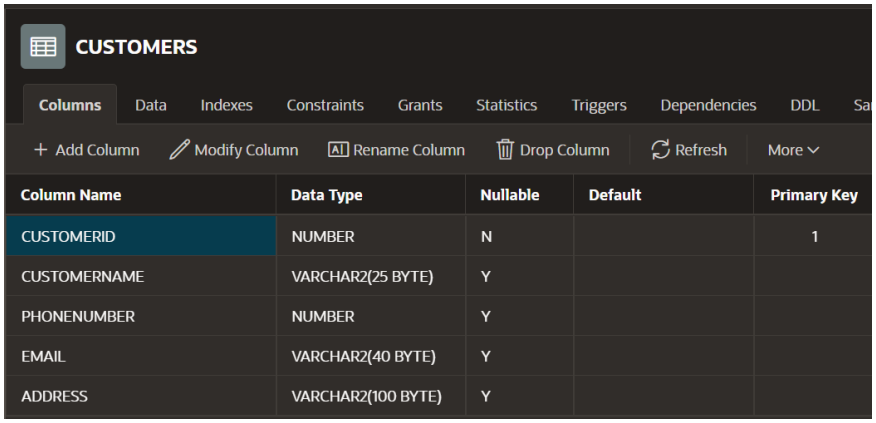


Column Name	Data Type	Nullable	Default	Primary Key
EXTERIORWIDTH	FLOAT(126)	Y		
EXTERIORHEIGHT	FLOAT(126)	Y		
INTERIORLENGTH	FLOAT(126)	Y		
INTERIORWIDTH	FLOAT(126)	Y		
INTERIORHEIGHT	FLOAT(126)	Y		
MGW	NUMBER	Y		
TAREWEIGHT	NUMBER	Y		
BOOKINGPRICE	NUMBER	Y		

- **Customers:**

Customers table has the mapping of customers and the details of the goods that need to be shipped with the Booking ID.

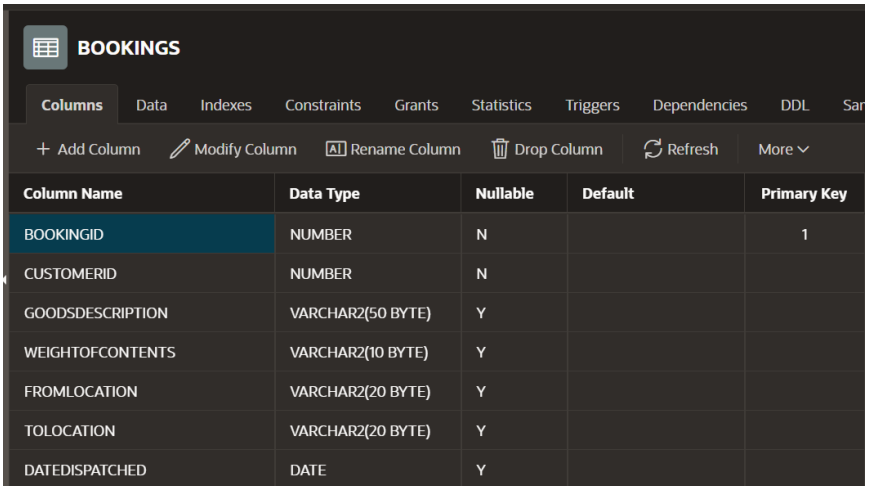
```
CREATE TABLE Customers(
CustomerID number,
CustomerName varchar(25),
PhoneNumber number,
Email varchar(40),
Address varchar(100),
PRIMARY KEY(CustomerID));
```



- **Bookings:**

Bookings table maps the customer IDs with the details of the orders. Using BookingIDs, containers are allocated.

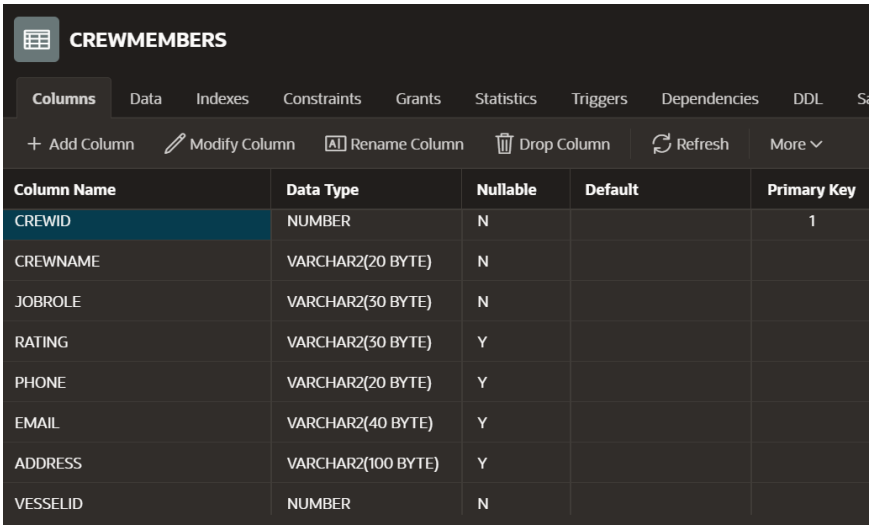
```
create table Bookings(
BookingID number,
CustomerID number NOT NULL,
GoodsDescription varchar(50),
WeightOfContents varchar(10),
FromLocation varchar(20),
ToLocation varchar(20),
DateDispatched date,
PRIMARY KEY(BookingID),
FOREIGN KEY(CustomerID) REFERENCES Customers(CustomerID));
```



- **CrewMembers:**

Crew members table has information about crew working in each vessel.

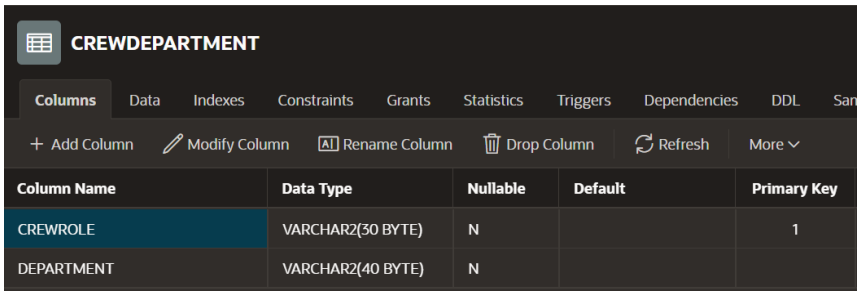
```
CREATE TABLE CrewMembers(
CrewID number,
CrewName varchar(20) NOT NULL,
JobRole varchar(30) NOT NULL,
Rating varchar(30),
Phone number,
Email varchar(40),
Address varchar(100),
VesselID number NOT NULL,
PRIMARY KEY(CrewID),
FOREIGN KEY(JobRole) REFERENCES CrewDepartment(CrewRole),
FOREIGN KEY(VesselID) REFERENCES Vessel(IMONumber));
```



- **CrewDepartment**:

Crew Department is used to map the role of the crew to which department the role belongs to, this table is added to reduce the redundancy of adding department column to the Crew members table.

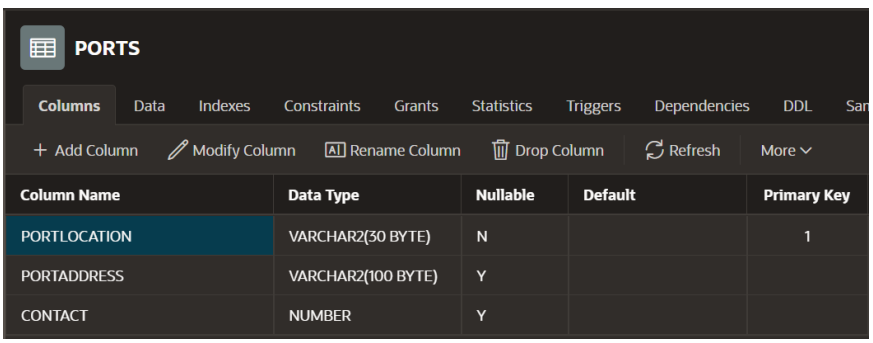
```
CREATE TABLE CrewDepartment(
CrewRole varchar(30),
Department varchar(40) NOT NULL,
PRIMARY KEY(CrewRole)
);
```



- **Ports:**

The company has offices at different ports and the details of the offices are stored in the Ports table.

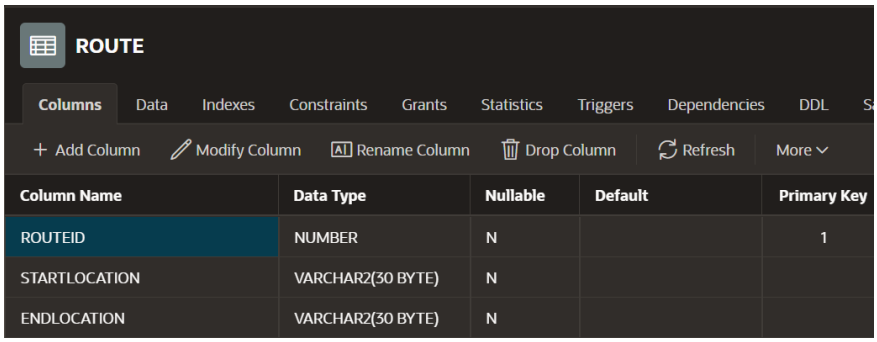
```
CREATE TABLE Ports(
PortLocation varchar(30),
PortAddress varchar(100),
Contact number,
PRIMARY KEY(PortLocation)
);
```



- **Routes:**

Routes table has the mapping of different routes showing that a route exists between the 2 locations.

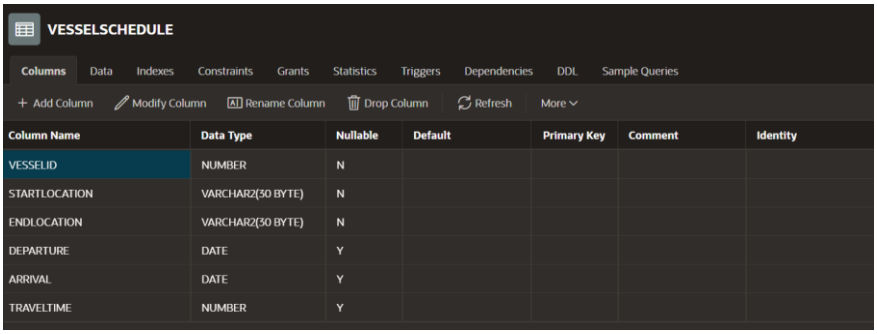
```
CREATE TABLE Route(
RouteID number,
StartLocation varchar(30) NOT NULL,
EndLocation varchar(30) NOT NULL,
PRIMARY KEY(RouteID)
);
```



- **VesselSchedule:**

Each vessel is assigned a schedule which defines the from and to destinations and time taken to travel. This table keeps track of the schedule each vessel is assigned to.

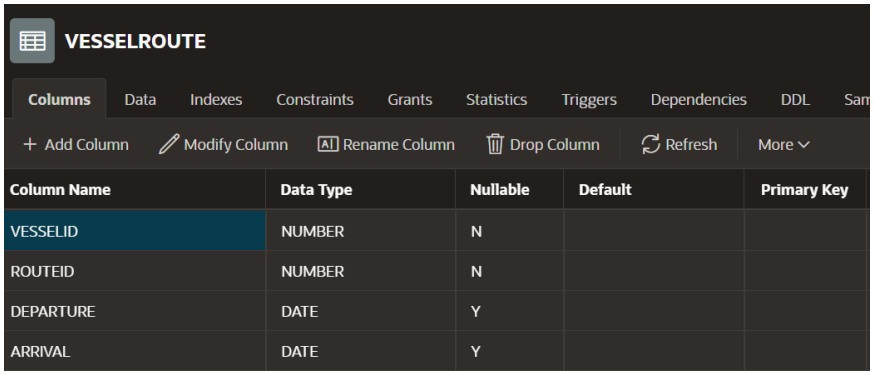
```
CREATE TABLE VesselSchedule(
VesselID number NOT NULL UNIQUE ,
StartLocation varchar(30) NOT NULL,
EndLocation varchar(30) NOT NULL,
Departure date,
Arrival date,
TravelTime number,
FOREIGN KEY(VesselID) REFERENCES Vessel(IMONumber));
```



- **VesselRoute**:

Vessel Route is an added table to resolve the many to many relationship between vessels and routes. This table stores the mapping of each vessel with the route it takes.

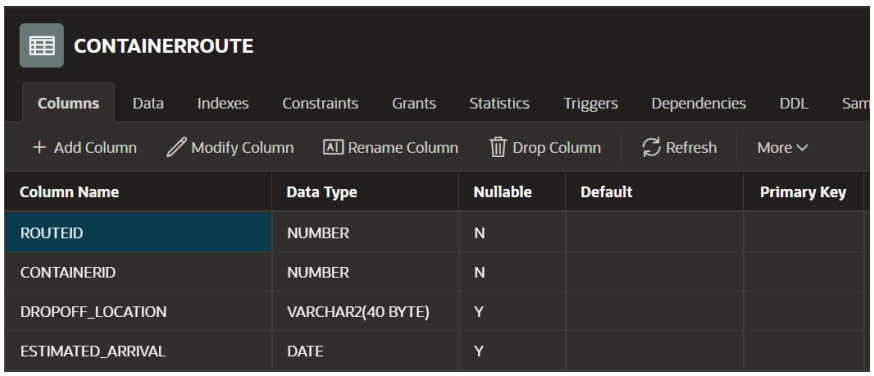
```
CREATE TABLE VesselRoute(
VesselID number NOT NULL,
RouteID number NOT NULL,
Departure date,
Arrival date,
FOREIGN KEY(VesselID) REFERENCES Vessel(IMONumber),
FOREIGN KEY(RouteID) REFERENCES Route(RouteID)
);
```



- **ContainerRoute:**

ContainerRoute table is similar to VesselRoute that is it resolves the many to many relationship between containers and routes. To resolve this, container route keeps track of the routes containers take. Containers can be dropped off at intermediate locations.

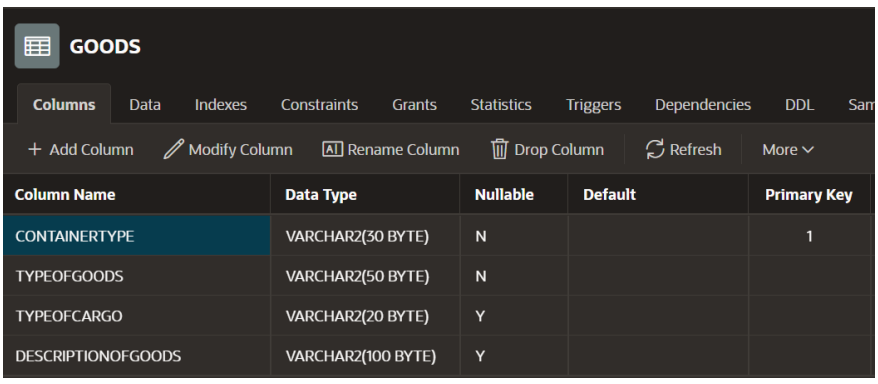
```
CREATE TABLE ContainerRoute(
RouteID number NOT NULL,
ContainerID number NOT NULL,
Dropoff_location varchar(40),
Estimated_Arrival date,
FOREIGN KEY(RouteID) REFERENCES Route(RouteID),
FOREIGN KEY(ContainerID) REFERENCES VesselContainer(ContainerNumber)
);
```



- **Goods:**

Goods table stores the information about the type of goods that can go inside different types of containers.

```
CREATE TABLE Goods(
ContainerType varchar(30),
TypeOfGoods varchar(50) NOT NULL,
TypeOfCargo varchar(20),
DescriptionOfGoods varchar(100),
PRIMARY KEY(ContainerType)
);
```



IV. MULTIPLICITY OF RELATIONSHIPS

The multiplicity of a relationship specifies the number of instances of an Entity type that can be associated with instances of other entity type. It shows the minimum and maximum allowed members of one entity type in another. Different types of multiplicity are

- One to many: every time one entity occurs, there are multiple occurrences of another entity.
- One to one: every time one entity occurs, there is exactly one occurrence of another entity.
- Many to many: for each occurrence of an entity, there can be one or many occurrences of another and vice versa.

To represent the multiplicity of relationships in our database design, the "Look Across Cardinality Constraint" method is used. So, the multiplicity of one entity can be understood by looking across the diagram.
Below is the discussion and evidence of how data is stored based on the multiplicity of the entities.

- **Vessel and containers:**

One vessel can carry one or more containers (one to many), but one container can be carried by only one vessel (one to one).

VESSELCONTAINER

EDIT	CONTAINERNUMBER	CONTAINERMODEL	IMONUMBER	BOOKINGID	STATUS
	1001	20' Special	7034567	101	Shipped
	1002	40' Tank	7145678	102	Shipped
	1003	40' Open Top	7478901	103	In transit
	1004	40' Special	7567890	104	Shipped
	1005	40' Hi-Cube	7480134	105	In transit
	1006	20' Open Top	7145678	106	Shipped
	1007	20' Standard	7601356	107	Shipped
	1008	20' Standard	7267912	108	Shipped
	1009	40' Open Top	7701234	109	Shipped
	1010	20' Reefer	7712467	110	In transit

The containers "1002, 1006" are carried by one vessel that is "7145678". This shows that one vessel can carry multiple containers.

- **Vessel and Crew members:**

Each vessel has at least one crew member working in each department (one to many), and each crew member works in one vessel (one to one). This shows that multiple records of crew mwmbers are mapped to one vessel.

CREWMEMBERS

EDIT	CREWID	CREWNAME	JOBROLE	RATING	PHONE	EMAIL	ADDRESS	VESSELID
	22	Lily Lee	Chief Officer	AbleBodies Seaman	+1 555-456-7890	lily.lee@example.com	345 Sixth St, Boston, MA 02101	7256789
	23	Ava Wong	Second Officer	AbleBodies Seaman	+1 555-789-0123	ava.wong@example.com	789 Broad St, San Francisco, CA 94109	7256789
	24	Caleb Park	Third Officer	AbleBodies Seaman	+1 555-234-5678	caleb.park@example.com	456 Park Ave, New York, NY 10001	7256789
	25	William Chen	Chief Engineer	Engine Room rating	+1 555-678-9012	william.chen@example.com	123 Main St, Los Angeles, CA 90001	7256789
	26	Ethan Kim	Second Engineer	Electrician	+1 555-789-0123	ethan.kim@example.com	789 Broad St, San Francisco, CA 94109	7256789
	27	Logan Lee	Third Engineer	Electrician	+1 555-234-5678	logan.lee@example.com	456 Park Ave, New York, NY 10001	7256789

CREWMEMBERS

	27	Logan Lee	Third Engineer	Electrician	+1 555-234-5678	logan.lee@example.com	456 Park Ave, New York, NY 10001	7256789
	28	Harper Park	Fourth Engineer	Head Tunnelman	+1 555-678-9012	harper.park@example.com	123 Main St, Los Angeles, CA 90001	7256789
	29	Michael Johnson	Chief Steward	-	+1 555-678-9012	michael.johnson@example.com	123 Main St, Los Angeles, CA 90001	7256789
	1	John Smith	Captain	Master	-	john.smith@example.com	123 Main St, Anytown, US	7034567
	2	Mark Johnson	Chief Officer	AbleBodies Seaman	+1 (555) 555-0202	mark.johnson@example.com	456 Elm St, Anytown, US	7034567
	3	Sarah Lee	Second Officer	Ordinary Seaman	+1 (555) 555-0303	sarah.lee@example.com	789 Maple St, Anytown, US	7034567
	4	James Chen	Third Officer	Ordinary Seaman	+1 (555) 555-0404	james.chen@example.com	234 Oak St, Anytown, US	7034567

- **Vessel and Vessel Type:**

One vessel type can belong to one or more vessels (one to many) whereas one vessel can be of one vessel type(one to one).

VESSEL

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST | Sample Queries

Query | Count Rows | Insert Row | Load Data

EDIT	IMONUMBER	VESSELNAME	VESSELTYPE	SPEED	GROSSTONNAGE	DEADWEIGHT	DATEBUILT
	7034567	Ever Oceanic Dream	A	24.5	18500	28000	01/07/2014
	7145678	Ever Sailor Delight	G	22.3	11500	17500	09/13/2010
	7256789	Ever Marine Adventure	T	19.8	6200	9000	04/21/2008
	7367890	Ever Sea Queen	F	25.1	19500	29500	06/05/2013
	7478901	Ever Starlight Express	L	23.7	12000	18500	11/29/2011
	7589012	Ever Coral Princess	S	18.600000000000001	5400	8200	02/16/2009
	7690123	Ever Sunset Voyager	E	26.3	20000	30500	03/22/2015

VESSEL

	7478901	Ever Starlight Express	L	23.7	12000	18500	11/29/2011
	7589012	Ever Coral Princess	S	18.600000000000001	5400	8200	02/16/2009
	7690123	Ever Sunset Voyager	E	26.3	20000	30500	03/22/2015
	7701234	Ever Sea Breeze	B	24.2	13000	19500	08/08/2012
	7812345	Ever Majestic Explorer	O	20.100000000000001	6000	9200	01/01/2010
	7934578	Ever Island Hopper	C	23.5	17500	26500	11/14/2014
	7156801	Ever Sea Serpent	G	20	5700	8400	12/09/2008
	7045689	Ever Pacific Star	A	21.9	9500	15000	05/28/2011
	7267912	Ever Ocean Voyager	T	25.5	19000	29000	09/01/2013
	7379023	Ever Starfish	F	23.3	11800	18000	02/23/2011
	7480134	Ever Coral Reef	L	18.2	5000	7700	07/15/2009

- **Container and Container Type:**

Each container is of one container type (one-to-one), a container type can belong to zero or more containers.

CONTAINERDETAILS

Query | Count Rows | Insert Row | Load Data

EDIT	CONTAINERMODEL	CONTAINERTYPE	EXTERIORLENGTH	EXTERIORWIDTH	EXTERIORHEIGHT	INTERIORLENGTH	INTERIORWIDT
	20' Standard	Dry Cargo Container	6.05799999999999998	2.4380000000000002	2.5910000000000002	5.9020000000000001	2.351999999999
	40' Standard	Dry Cargo Container	12.192	2.4380000000000002	2.5910000000000002	12.036	2.351999999999
	40' Hi-Cube	Dry Cargo Container	12.192	2.4380000000000002	2.8959999999999999	12.036	2.351999999999
	45' Hi-Cube	Dry Cargo Container	13.715999999999999	2.4380000000000002	2.8959999999999999	13.555999999999999	2.351999999999
	20' Tank	Tank Cotainer	6.05799999999999998	2.4380000000000002	2.5910000000000002	5.9020000000000001	2.351999999999
	40' Tank	Tank Cotainer	12.192	2.4380000000000002	2.5910000000000002	12.036	2.351999999999

- **Customer and Containers:**

Each customer can book one or more containers (one to many) under different booking IDs and one container can be booked by only one customer (one to one). The bookings table shows that one customer can make multiple bookings.

CUSTOMERS

EDIT	CUSTOMERID	CUSTOMERNAME	PHONENUMBER	EMAIL	ADDRESS
	1	Mike Brown	6591234567	mikebrown@email.com	123 Main St, London
	2	Sophie Wu	86217654321	sophiewu@email.com	456 Nanjing Rd, Shanghai
	3	Tom Wilson	91229876543	tomwilson@email.com	789 Marine Dr, Mumbai
	4	Lisa Kim	442345678901	lisakim@email.com	234 Ocean Ave, Southampton
	5	Eric Chen	3105551212	ericchen@email.com	901 Galle Face Rd, Colombo
	6	Anna Chen	86212345678	annachen@email.com	345 Champs-Élysées, Paris
	7	John Smith	44201234567	johnsmith@email.com	123 Nanjing Rd, Shanghai
	8	Mary Johnson	442098765432	maryjohnson@email.com	678 Rue de Rivoli, Paris
	9	David Lee	941176543211	davidlee@email.com	567 Main St, Los Angeles
	10	Anna Chen	86212345678	annachen@email.com	345 Champs-Élysées, Paris

BOOKINGS

EDIT	BOOKINGID	CUSTOMERID	GOODSDESCRIPTION	WEIGHTOFCONTENTS	FROMLOCATION	TOLOCATION	DATEDISPATCHED
	1001	1	Fragile Artwork	50kg	London	Hamburg	04/10/2023
	1002	2	Liquids	300kg	Shanghai	Tokyo	04/11/2023
	1003	3	Raw Materials	1000kg	Mumbai	Shanghai	04/12/2023
	1004	4	Dangerous Chemicals	500kg	Sothhampton	Rotterdam	04/10/2023
	1005	5	Consumer Electronics	150kg	Colombo	Shanghai	04/14/2023
	1006	6	Household Appliances	250kg	Paris	Tokyo	04/09/2023
	1007	7	Electronics	100kg	Shanghai	New York	04/08/2023
	1008	8	Clothing	500kg	Paris	Tokyo	04/09/2023
	1009	9	Industrial Machinery	2000kg	Los Angeles	New York	04/11/2023
	1010	6	Perishable Goods	100kg	Paris	Singapore	04/14/2023

- **Crew Department and Crew members:**

The crew department has one or more crew members (one to many) and each crew member belongs to one crew department (one to one). As different one department has multiple roles and each crew member has a role associated, this represents one to many relationships.

CREWMEMBERS

EDIT	CREWID	CREWNAME	JOBROLE	RATING	PHONE	EMAIL	ADDRESS	VESSELID
	22	Lily Lee	Chief Officer	AbleBodies Seaman	+1 555-456-7890	lily.lee@example.com	345 Sixth St, Boston, MA 02101	7256789
	23	Ava Wong	Second Officer	AbleBodies Seaman	+1 555-789-0123	ava.wong@example.com	789 Broad St, San Francisco, CA 94109	7256789
	24	Caleb Park	Third Officer	AbleBodies Seaman	+1 555-234-5678	caleb.park@example.com	456 Park Ave, New York, NY 10001	7256789
	25	William Chen	Chief Engineer	Engine Room rating	+1 555-678-9012	william.chen@example.com	123 Main St, Los Angeles, CA 90001	7256789
	26	Ethan Kim	Second Engineer	Electrician	+1 555-789-0123	ethan.kim@example.com	789 Broad St, San Francisco, CA 94109	7256789
	27	Logan Lee	Third Engineer	Electrician	+1 555-234-5678	logan.lee@example.com	456 Park Ave, New York, NY 10001	7256789

CREWDEPARTMENT

EDIT	CREWROLE	DEPARTMENT
	Captain	Captain
	Chief Officer	Deck Department
	Second Officer	Deck Department
	Third Officer	Deck Department
	Chief Engineer	Engineering Department
	Second Engineer	Engineering Department
	Third Engineer	Engineering Department
	Fourth Engineer	Engineering Department
	Chief Steward	Steward's Department
	Assistant Steward	Steward's Department

- **Ports and Routes:**

The company has offices at different port locations and routes are a possible combination of 2 port location defining the route between the locations. The startlocation and endlocation are the port locations from port table.

ROUTE

EDIT	ROUTEID	STARTLOCATION	ENDLOCATION
	1	Paris	Tokyo
	2	Paris	Singapore
	3	Singapore	Tokyo
	4	Paris	Shanghai
	5	Shanghai	Tokyo
	6	Paris	Busan
	7	Busan	Tokyo
	8	Shanghai	Busan
	9	Busan	Los Angeles
	10	Los Angeles	New York

- **Routes and Vessels:**

- Each vessel has a predefined start and end location, but the start and end locations can have multiple sub-routes between them that a vessel can take to reach the end location and one route can be taken by zero or more vessels. This has many to

many relationships. Many to many relationships add confusion and complexity to the database design. To resolve this, we create another entity that has a one-to-one or one-to-many relationship with the two entities.

- Vessel Route and Container route are added to serve this purpose.

VESSELROUTE

7034567	15	04/11/2023	04/12/2023	1
7034567	16	04/12/2023	04/12/2023	0
7034567	17	04/13/2023	04/15/2023	2
7034567	18	04/15/2023	04/16/2023	1
7145678	4	04/09/2023	04/10/2023	1
7145678	5	04/11/2023	04/13/2023	2
7478901	19	04/12/2023	04/14/2023	2
7478901	20	04/14/2023	04/16/2023	2
7478901	21	04/16/2023	04/17/2023	1
7367890	14	04/10/2023	04/11/2023	1
7367890	15	04/11/2023	04/13/2023	2

- **Goods and Container Details:**

One container model or type can carry one type of goods based on the category of the goods the customer wants to be shipped. One Container type has multiple container models and each model belongs to one type.

CONTAINERDETAILS

CONTAINERMODEL	CONTAINERTYPE	EXTERIORLENGTH	EXTERIORWIDTH	EXTERIORHEIGHT	INTERIORLENGTH	INTERIORWIDTH
20' Standard	Dry Cargo Container	6.0579999999999998	2.4380000000000002	2.5910000000000002	5.9020000000000001	2.3519999999999999
40' Standard	Dry Cargo Container	12.192	2.4380000000000002	2.5910000000000002	12.036	2.3519999999999999
40' Hi-Cube	Dry Cargo Container	12.192	2.4380000000000002	2.8959999999999999	12.036	2.3519999999999999
45' Hi-Cube	Dry Cargo Container	13.715999999999999	2.4380000000000002	2.8959999999999999	13.555999999999999	2.3519999999999999
20' Tank	Tank Cotainer	6.0579999999999998	2.4380000000000002	2.5910000000000002	5.9020000000000001	2.3519999999999999
40' Tank	Tank Cotainer	12.192	2.4380000000000002	2.5910000000000002	12.036	2.3519999999999999
20' Open Top	Open top Container	6.0579999999999998	2.4380000000000002	2.5910000000000002	5.8179999999999996	2.347

V. SYSTEM REQUIREMENTS

SQL queries that are implemented to justify that the database design meets the system requirements are listed below:

1) Maintaining details of the service routing network in order to work out the routing of the vessels.

Each vessel has a predefined schedule that it has to follow, but the start and end locations can have multiple routes to reach the destination. By running the below query, the system displays the route a vessel takes to reach the end destination.

```
SELECT vr.VesselID,r.StartLocation,
r.EndLocation,
vr.TravelTime AS TravelTime_in_days
FROM VesselRoute vr
INNER JOIN Route r
ON vr.RouteID = r.RouteID
WHERE vr.VesselID = 7601356;
```

Results | Explain | Describe | Saved SQL | History

VESSELID	STARTLOCATION	ENDLOCATION	TRAVELTIME_IN_DAYS
7601356	Shanghai	Hong Kong	2
7601356	Hong Kong	Vancouver	1
7601356	Vancouver	New York	1

3 rows returned in 0.01 seconds Download

2) Enable customers to search for sailing schedules

The following query helps the customers to search for the vessel sailing schedules. Using this query, customers can find out what vessels are taking a route which passes by one specific location, departing on a particular date.

```
SELECT vs.VesselID, vs.StartLocation,
vs.EndLocation,vs.Departure,
```

```
vs.Arrival
FROM VesselSchedule vs
WHERE vs.StartLocation = 'London'
AND vs.EndLocation = 'Dubai'
AND vs.Departure = '04/10/2023';
```

Results | Explain | Describe | Saved SQL | History

VESSELID	STARTLOCATION	ENDLOCATION	DEPARTURE	ARRIVAL
7034567	London	Dubai	04/10/2023	04/16/2023
7367890	London	Dubai	04/10/2023	04/17/2023

Based on the given query, the vessels with IDs '7034567' and '7367890' start from London to Dubai on the given date. The customers can use the query to know which vessels are starting from a certain location on a particular date and search for different sailing schedules.

3) Enable customers to track cargo.

```
SELECT cu.CustomerID,cu.CustomerName,
r.StartLocation,r.EndLocation,
vc.Status,cr.Estimated_Arrival
FROM Customers cu
INNER JOIN Bookings b
ON cu.CustomerID = b.CustomerID
INNER JOIN VesselContainer vc
ON vc.BookingID = b.BookingID
INNER JOIN ContainerRoute cr
ON vc.ContainerNumber = cr.ContainerID
INNER JOIN Route r
ON cr.RouteID = r.RouteID
WHERE cu.CustomerID = 1;
```

Output:
The query displays the tracking of the shipment when the customer gives their Customer ID in the query. So, for customer with Customer ID 1, the cargo tracking is as shown below along with the status of the shipment.

Results | Explain | Describe | Saved SQL | History

CUSTOMERID	CUSTOMERNAME	STARTLOCATION	ENDLOCATION	STATUS	ESTIMATED_ARRIVAL
1	Mike Brown	London	Felixstowe	Shipped	04/11/2023
1	Mike Brown	Felixstowe	Southampton	Shipped	04/12/2023
1	Mike Brown	Southampton	Rotterdam	Shipped	04/12/2023
1	Mike Brown	Rotterdam	Hamburg	Shipped	04/15/2023

4 rows returned in 0.02 seconds Download

4) Record details of goods conveyed

To record the details of the goods conveyed, we use the below query to set the 'Status' of the customer shipment in the customer table.

```
UPDATE VesselContainer vc
SET vc.status =
      CASE WHEN vc.ContainerNumber IN (SELECT cr.ContainerID FROM ContainerRoute
cr WHERE cr.Estimated_Arrival > SYSDATE) THEN 'In transit'
    ELSE 'Shipped'
END
```

Once this query is executed, the status field in the customer table is updated with the status of the shipment based on the estimated arrival time and the current time. Estimated Arrival time is entered based on the vessel schedule in which the container is present. Once this is done, the details of the shipment along with the status can be tracked using the query below.

```
SELECT cu.CustomerID,cu.CustomerName,
b.GoodsDescription,b.WeightOfContents,
b.FromLocation,b.ToLocation,
b.DateDispatched,vc.Status
FROM Customers cu
```

```
INNER JOIN Bookings b
ON cu.CustomerID = b.CustomerID
INNER JOIN VesselContainer vc
ON b.BookingID = vc.BookingID
WHERE cu.CustomerID = 3;
```

Results | Explain | Describe | Saved SQL | History

CUSTOMERID	CUSTOMERNAME	GOODSDESCRIPTION	WEIGHTOFCONTENTS	FROMLOCATION	TOLOCATION	DATEDISPATCHED	STATUS
3	Tom Wilson	Raw Materials	1000kg	Mumbai	Shanghai	04/12/2023	Shipped

1 rows returned in 0.01 seconds Download

5) Enable customers to search for containers and book containers.

This command will display the type of container models available based on the type of goods that the customer wants to ship. The query gives the capacity of the container and booking price per day so that the customer can book them based on their requirement.

```
SELECT cd.ContainerModel,
cd.ContainerType,cd.MGW as MaximumGrossWeight,
cd.TareWeight,
cd.BookingPrice as BookingPrice_per_day
FROM ContainerDetails cd
WHERE ContainerModel
IN(
SELECT cd.ContainerModel
FROM ContainerDetails cd
INNER JOIN Goods g ON g.ContainerType=cd.ContainerType
WHERE g.TypeOfGoods= 'Liquid goods'
);
```

Results | Explain | Describe | Saved SQL | History

CONTAINERMODEL	CONTAINERTYPE	MAXIMUMGROSSWEIGHT	TAREWEIGHT	BOOKINGPRICE_PER_DAY
20' Tank	Tank Cotainer	32470	2400	200
40' Tank	Tank Cotainer	30480	2600	280

2 rows returned in 0.02 seconds Download

6) Production of vessel schedules which will utilise the allocation of cargo efficiently for the transportation of goods.

The following query gives the information about vessel schedules that includes to, from locations of a vessel journey,the number of containers the vessel can carry (Vessel Capacity, in terms of TEU) using which cargo can be efficiently allocated to the vessels based on their schedule.

```
SELECT vs.VesselID, vs.StartLocation,
vs.EndLocation,
vs.Departure,vs.Arrival,vs.TravelTime,
vt.VesselCapacity
FROM VesselSchedule vs
INNER JOIN Vessel v
ON vs.VesselID = v.IMONumber
INNER JOIN VesselType vt
ON vt.VesselType = v.VesselType;
```

Results | Explain | Describe | Saved SQL | History

VESSELID	STARTLOCATION	ENDLOCATION	DEPARTURE	ARRIVAL	TRAVELTIME	VESSELCAPACITY
7034567	London	Dubai	04/10/2023	04/16/2023	6	23992
7145678	Paris	Tokyo	04/09/2023	04/13/2023	4	20124
7367890	London	Dubai	04/10/2023	04/17/2023	7	12118
7478901	Mumbai	Shanghai	04/12/2023	04/17/2023	5	9466
7480134	Mumbai	Shanghai	04/12/2023	04/16/2023	4	9466
7601356	Shanghai	NewYork	04/08/2023	04/12/2023	4	6332

VI. CONCLUSION

The ECE company must maintain large amounts of data in order to efficiently manage and allocate cargo to help customers deliver their shipments safely and fast. Although, the currently designed database contains data that represents a very small portion of it the company might have large amounts of data. In order to deal with this, an efficient database system needs to be designed and implemented. The database is designed in such a way that the system requirements of the company are met and customers can efficiently track their cargo and see its status. A few assumptions are made that suit the database design. Below are the takeaways and learnings from the assignment.

- Identifying entities from the problem scenario.
- Understanding multiplicity and cardinality between the entities.
- Resolving many to many relationships

About the assignment:

- The assignment helps us understand the business logic.
- Designing the database is a step-by-step process where the requirements of the business should be taken into consideration, finding the important entities and relationships between the entities.
- The database design should be modelled in a way that the logic is sensible and well suited to the basic requirements of the company along with making assumptions so that the product is achievable.

Things that went well:

- Understand the objective of the assignment and come up with a solution that works for the design.
- Identify key entities and tables that can help meet the system requirements.
- Identify the constraints in each table to be able to perform joins between two or more tables.
- Write queries that will fetch desired output to meet the system requirement.
- Able to normalize the tables created to avoid redundancy and complexity in the designed database.
- Load the database in such a way that the data is mapped well to satisfy the cardinality and multiplicity of the relationships.

Things that I would have done differently:

- While implementing the assignment, I created the tables first before finalizing the data, which made me realize a few relationships and I changed the tables more than once because of this reason.
- Though ER diagram gives a clear picture of how the design of the database should look, entering data in the tables will give a better idea of how the tables or entities should be linked to each other.
- Filling the data and mapping it manually was a tedious task and took up a lot of time and is redundant. If I would have realized this at an early stage of designing the database, I would have planned the assignment accordingly. This has been learning to get the required data at the early stages of designing the database so that enough time is allotted to all the tasks.

To conclude, the assignment is a practical approach to understanding how the database can be designed with relationships so that any kind of required information can be fetched from the database. With the use of triggers, the database design can be improved and updated automatically. Fields like the number of days to travel and the status of the shipment based on the estimated time of arrival can be set timely basis and can be made more dynamic. Using advanced SQL methods like procedures and triggers would have helped improve the design overall.

ORACLE APEX CREDENTIALS:

Username : k2247431@kingston.ac.uk
Password: MyApex*1
WorkSpace: shravya