KINGSTON UNIVERSITY LONDON

CI7320

DATABASES AND DATA MANAGEMENT

COURSEWORK 1

KU NUMBER	:	K2203709
NAME	:	Pratiksha Rajendran
MODULE LEADER	:	Beryl Jones

Credentials for Oracle Apex:

Workspace: K2203709

USER NAME : k2203709@kingston.ac.uk

Password : Admitkard@1234

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

ENTITY RELATIONSHIP DIAGRAM

The below diagram represents the Entity Relationship Diagram for the Everblue Ocean Express British Shipping Company.

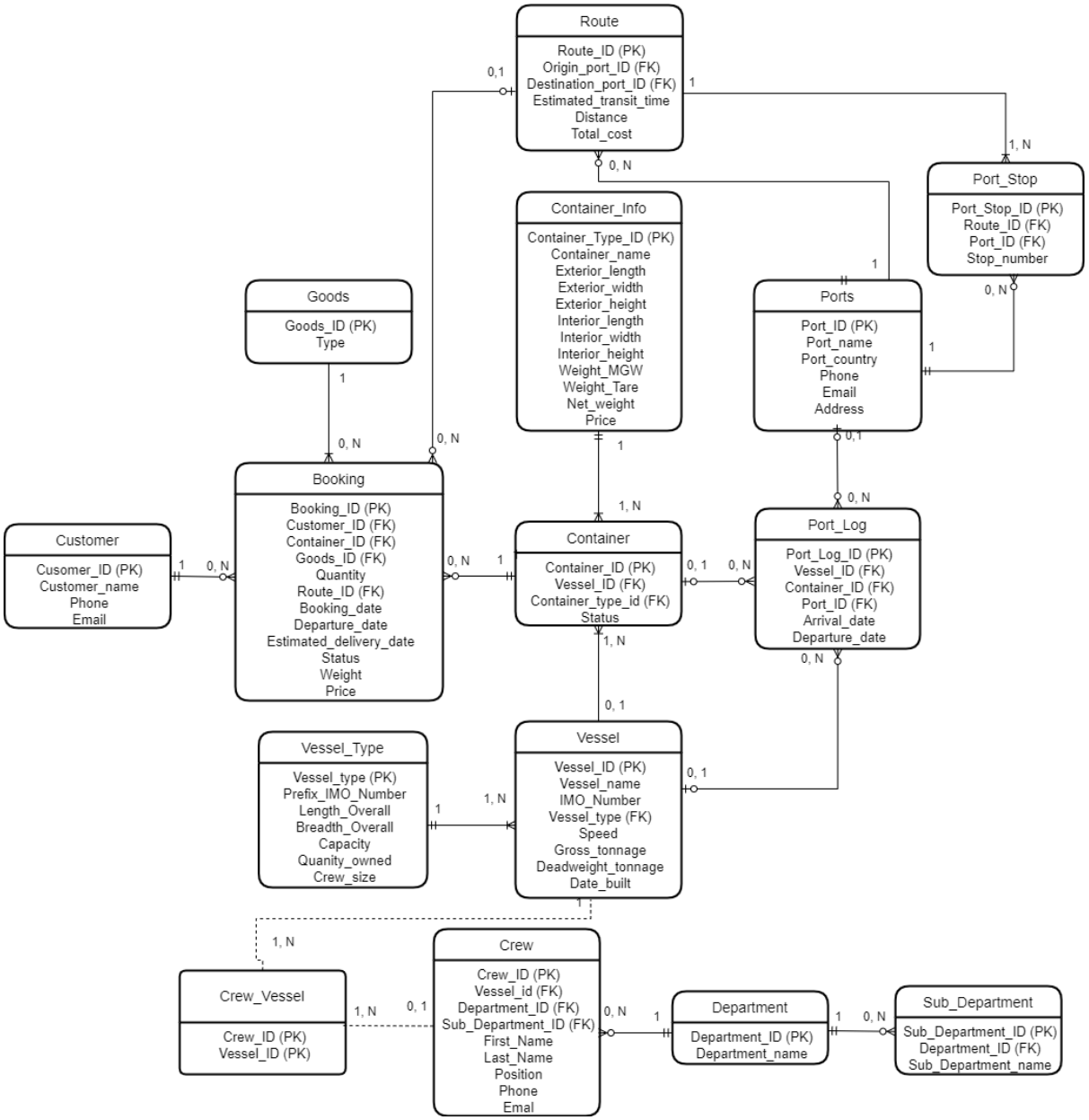


Fig 1. ER Diagram

ASSUMPTIONS:

1. Customer details can be stored in the system without having to book a container. Subsequently, the customer's information can be utilized when booking a container in the BOOKING table.
2. Each booking is associated with a single customer.
3. A booking can only be linked to one container and one type of goods at any given time.
4. A container and a type of goods can have multiple bookings.
5. Both containers and vessels can have several different types.
6. A vessel can carry one or more containers.
7. A vessel can be manned by multiple crew members who may be assigned to different vessels at various intervals.
8. A port may be associated with multiple routes.
9. A route can have multiple ports.
10. A route can comprise multiple port stops.
11. The Portlog stores all arrival and departure information of the vessels at the port office.
12. Each crew member must belong to a department, and a department can have multiple crew members.
13. A department can contain multiple sub-departments.

CHAPTER 2

IMPLEMENTATION

LIST OF SQL TABLE DEFINITIONS

Below are the list of tables and their respective SQL table definitions:

1. CUSTOMER:

```
CREATE TABLE CUSTOMER(
  CUSTOMER_ID NUMBER(6) PRIMARY KEY,
  CUSTOMER_NAME VARCHAR2(40),
  PHONE NUMBER(15),
  EMAIL VARCHAR2(80))
```

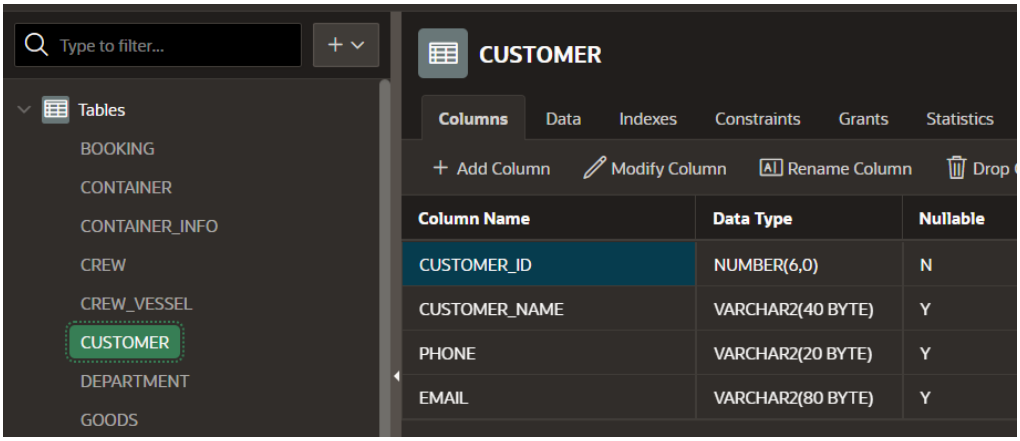


Fig 2. Customer Table

2. VESSEL_TYPE:

```
CREATE TABLE VESSEL_TYPE(
  VESSEL_TYPE VARCHAR2(1) PRIMARY KEY,
  PREFIX_IMO_NUMBER NUMBER(7),
  LENGTH_OVERALL_M NUMBER(5,2),
  BREADTH_OVERALL_M NUMBER(5,2),
  CAPACITY_TEU NUMBER(10),
  QUANTITY_OWNED NUMBER(3),
  CREW_SIZE NUMBER(4))
```

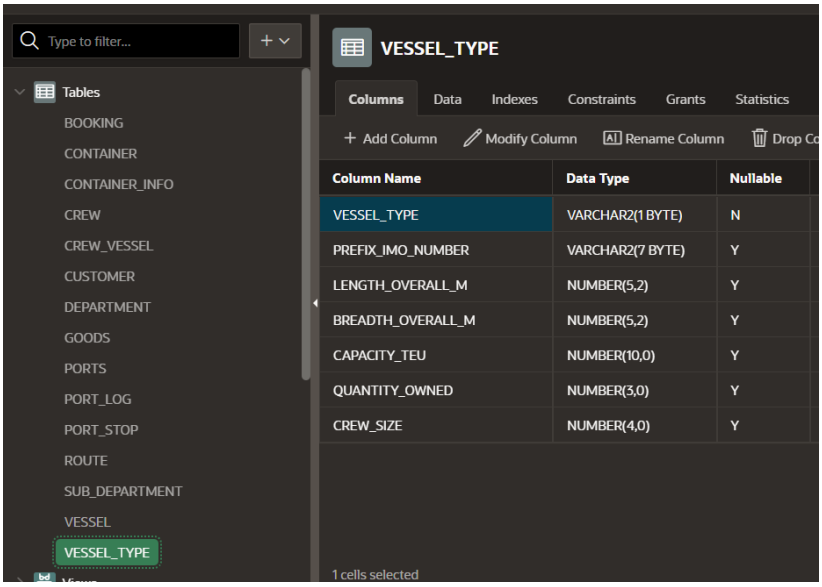


Fig 3. Vessel_Type Table

3. CONTAINER_INFO:

```
CREATE TABLE CONTAINER_INFO(
    CONTAINER_TYPE_ID VARCHAR2(3) PRIMARY KEY,
    CONTAINER_NAME VARCHAR2(100),
    EXTERIOR_LENGTH NUMBER(4,3),
    EXTERIOR_WIDTH NUMBER(4,3),
    EXTERIOR_HEIGHT NUMBER(4,3),
    INTERIOR_LENGTH NUMBER(4,3),
    INTERIOR_WIDTH NUMBER(4,3),
    INTERIOR_HEIGHT NUMBER(4,3),
    WEIGHT_MGW NUMBER(10),
    WEIGHT_TARE NUMBER(10),
    NET_WEIGHT NUMBER(10))
```

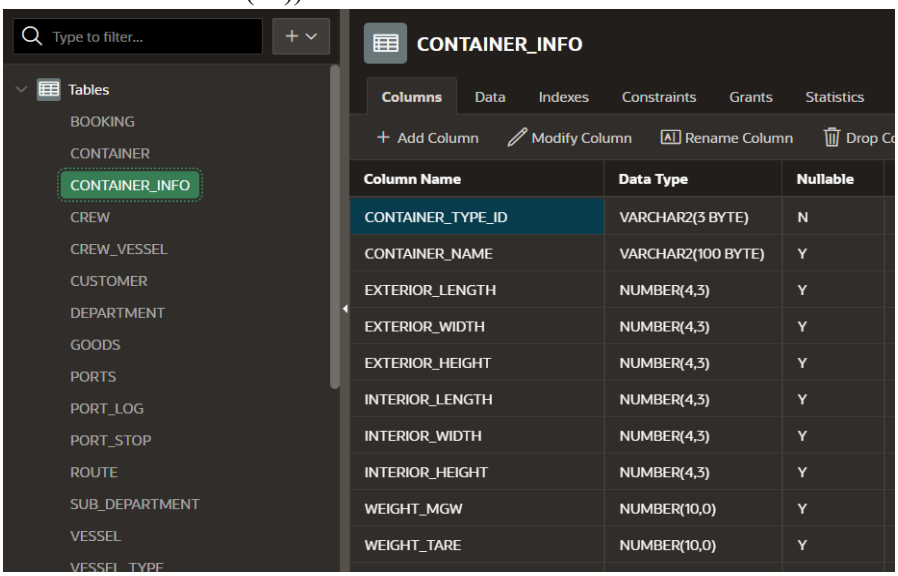


Fig 4. Container_info Table

4. CONTAINER:

```
CREATE TABLE CONTAINER(
    CONTAINER_ID VARCHAR2(10) PRIMARY KEY,
    VESSEL_ID REFERENCES VESSEL(VESSEL_ID),
    CONTAINER_TYPE_ID REFERENCES CONTAINER_INFO(CONTAINER_TYPE_ID),
    STATUS VARCHAR2(50))
```

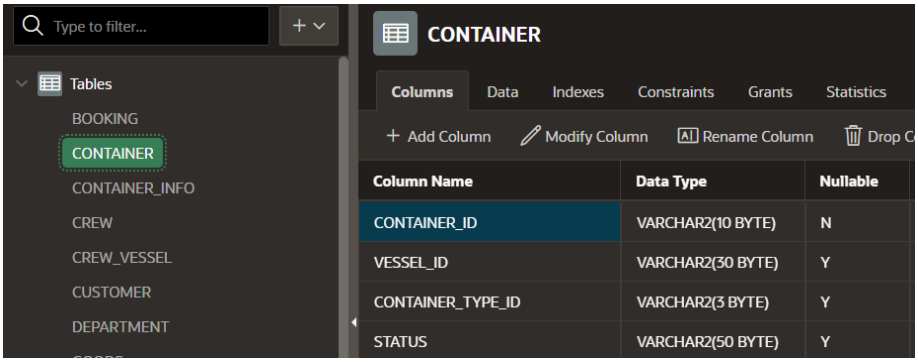


Fig 5. Container Table

5. GOODS:

```
CREATE TABLE GOODS(
    GOODS_ID VARCHAR2(10) PRIMARY KEY,
    TYPE VARCHAR2(500))
```

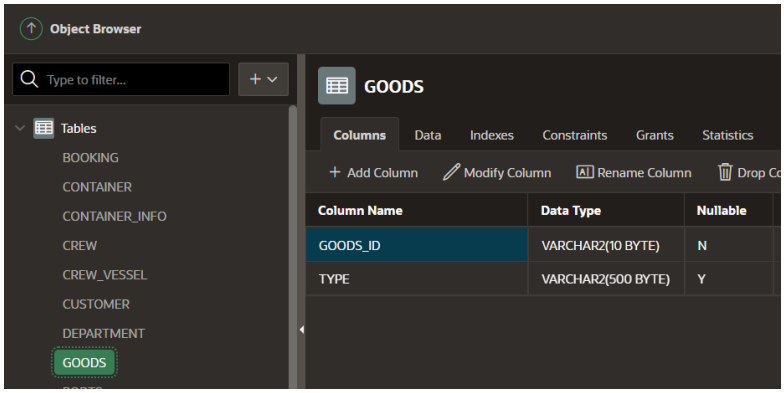


Fig 6. Goods Table

6. PORTS:

```
CREATE TABLE PORTS(
    PORT_ID VARCHAR2(40) PRIMARY KEY,
    PORT_NAME VARCHAR2(100),
    PORT_COUNTRY VARCHAR2(100),
    PHONE NUMBER(15),
    EMAIL VARCHAR2(100),
```

ADDRESS VARCHAR2(500))

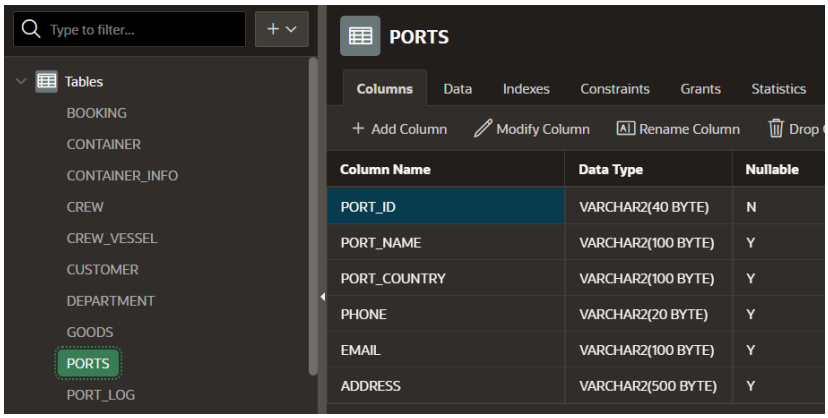


Fig 7. Ports Table

7. ROUTE:

```
CREATE TABLE ROUTE(
   ROUTE_ID VARCHAR2(40) PRIMARY KEY NOT NULL,
   ORIGIN_PORT_ID REFERENCES PORTS(PORT_ID),
   DESTINATION_PORT_ID REFERENCES PORTS(PORT_ID),
   ESTIMATED_TRANSIT_TIME TIMESTAMP,
   DISTANCE NUMBER(20,3),
   TOTAL_COST NUMBER(30,4)
)
```

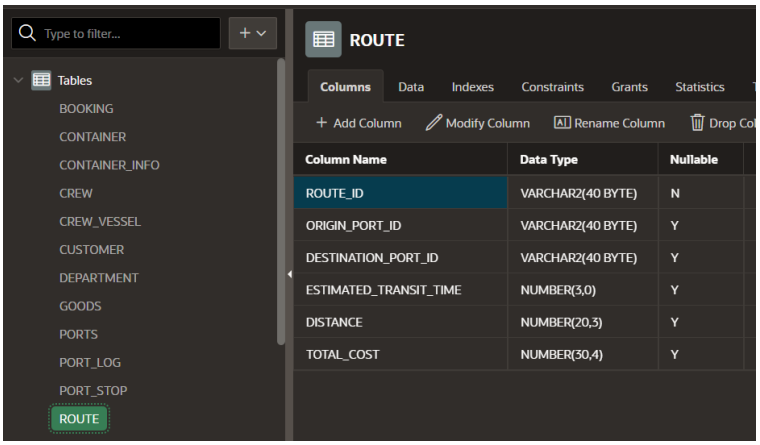


Fig 8. Route Table

8. PORT_STOP:

```
CREATE TABLE PORT_STOP(
   PORT_STOP_ID VARCHAR2(40) PRIMARY KEY,
   ROUTE_ID REFERENCES ROUTE(ROUTE_ID),
   PORT_ID REFERENCES PORTS(PORT_ID),
   STOP_NUMBER NUMBER(2)
)
```

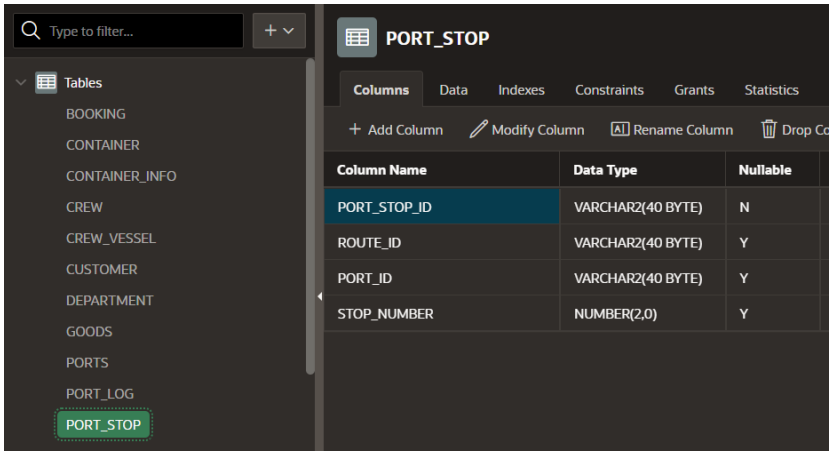


Fig **9**. Port_Stop Table

9. DEPARTMENT:

```
CREATE TABLE DEPARTMENT(
   DEPARTMENT_ID NUMBER(1) PRIMARY KEY,
   DEPARTMENT_NAME VARCHAR2(50) NOT NULL)
```

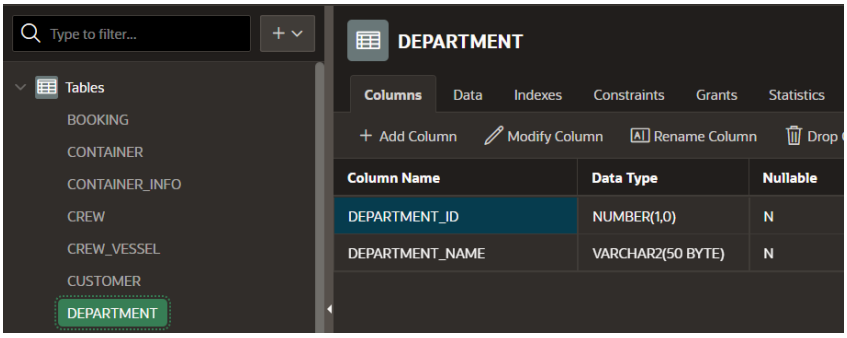


Fig 10. Department Table

10. SUB_DEPARTMENT:

```
CREATE TABLE SUB_DEPARTMENT(
   SUB_DEPARTMENT_ID NUMBER(1) PRIMARY KEY,
   DEPARTMENT_ID REFERENCES DEPARTMENT(DEPARTMENT_ID),
   SUB_DEPARTMENT_NAME VARCHAR2(50) NOT NULL)
```

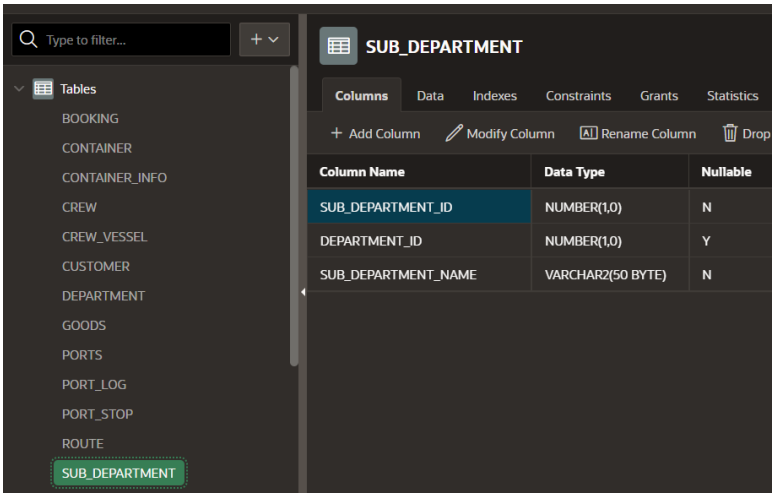


Fig 11. Sub_Department Table

11. CREW:

```
CREATE TABLE CREW(
   CREW_ID VARCHAR2(10) PRIMARY KEY,
   VESSEL_ID REFERENCES VESSEL(VESSEL_ID),
   DEPARTMENT_ID REFERENCES DEPARTMENT(DEPARTMENT_ID),
   SUB_DEPARTMENT_ID REFERENCES
SUB_DEPARTMENT(SUB_DEPARTMENT_ID),
   FIRST_NAME VARCHAR2(50) NOT NULL,
   LAST_NAME VARCHAR2(50),
   POSITION VARCHAR2(50),
   PHONE NUMBER(15),
   EMAIL VARCHAR2(80))
```

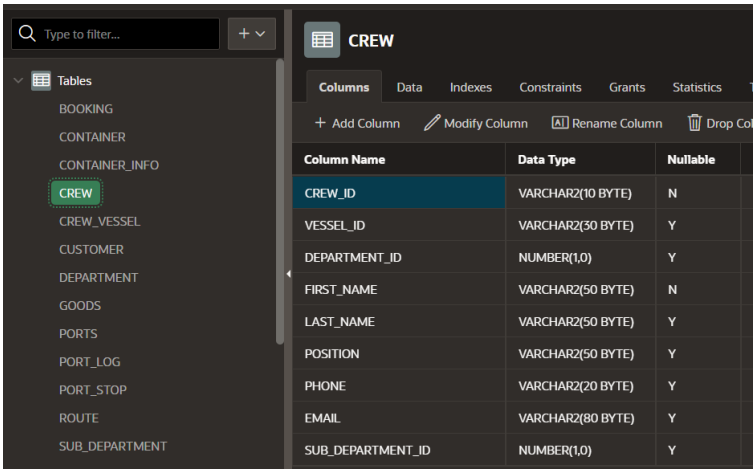


Fig 12. Crew Table

12. VESSEL:

```
CREATE TABLE VESSEL(
   VESSEL_ID VARCHAR(30) PRIMARY KEY,
   VESSEL_NAME VARCHAR2(100),
```

IMO_NUMBER NUMBER(7),
VESSEL_TYPE VARCHAR2(1) REFERENCES VESSEL_TYPE(VESSEL_TYPE),
SPEED_KNOTS NUMBER(2),
GROSS_TONNAGE NUMBER(20),
DEADWEIGHT_TONNAGE NUMBER(20),
DATE_BUILT DATE,
CREW_ID REFERENCES CREW(CREW_ID))

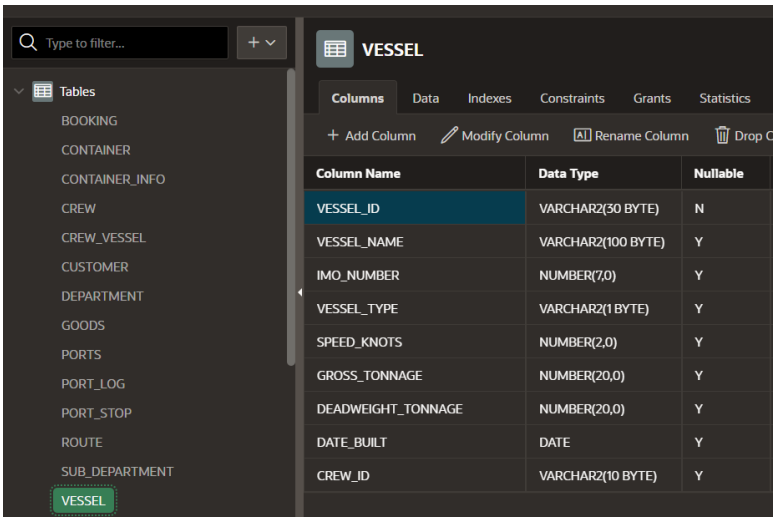


Fig 13. Vessel Table

13. PORT_LOG:

CREATE TABLE PORT_LOG(
PORT_LOG_ID NUMBER(35) PRIMARY KEY,
VESSEL_ID REFERENCES VESSEL(VESSEL_ID) NOT NULL,
CONTAINER_ID REFERENCES CONTAINER(CONTAINER_ID) NOT NULL,
PORT_ID REFERENCES PORTS(PORT_ID) NOT NULL,
ARRIVAL_DATE DATE,
DEPARTURE_DATE DATE)

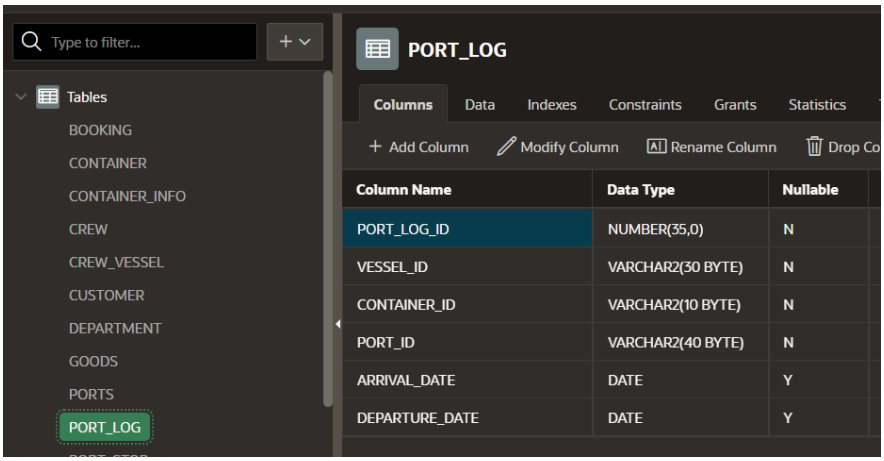


Fig 14. Port_Log Table

14. BOOKING:

```
CREATE TABLE BOOKING(
   BOOKING_ID NUMBER(35) PRIMARY KEY,
   CUSTOMER_ID REFERENCES CUSTOMER(CUSTOMER_ID),
   CONTAINER_ID REFERENCES CONTAINER(CONTAINER_ID),
    GOODS_ID REFERENCES GOODS(GOODS_ID),
   ROUTE_ID REFERENCES ROUTE(ROUTE_ID),
   BOOKING_DATE DATE,
   DEPARTURE_DATE DATE,
   ESTIMATED_DELIVERY_DATE DATE,
   STATUS VARCHAR2(20),
   WEIGHT NUMBER(35),
   PRICE NUMBER(35))
```

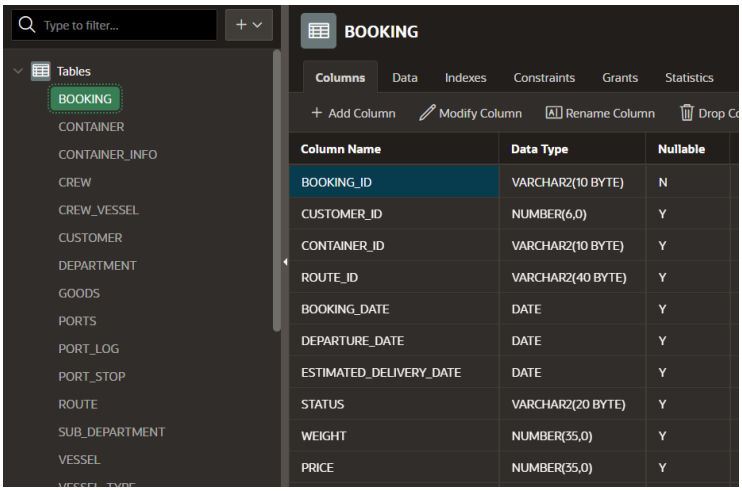


Fig 15. Booking Table

15. CREW_VESSEL:

```
CREATE TABLE CREW_VESSEL(
    CREW_ID REFERENCES CREW(CREW_ID),
    VESSEL_ID REFERENCES VESSEL(VESSEL_ID))
```

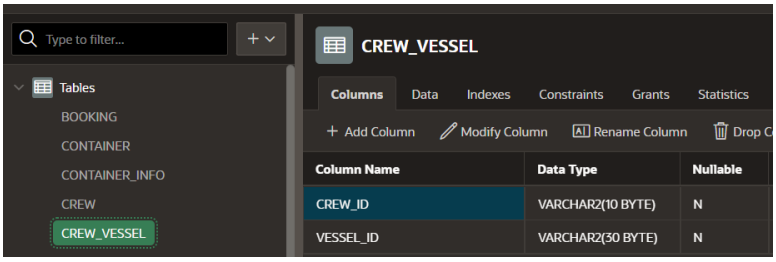


Fig 16. Crew_Vessel Table

CHAPTER 3

MULTIPLICITY REFLECTION ON DATA ENTERED

The relationships between the tables are briefed out below:

1. One to Many Relationships:

a. Customer and Booking:

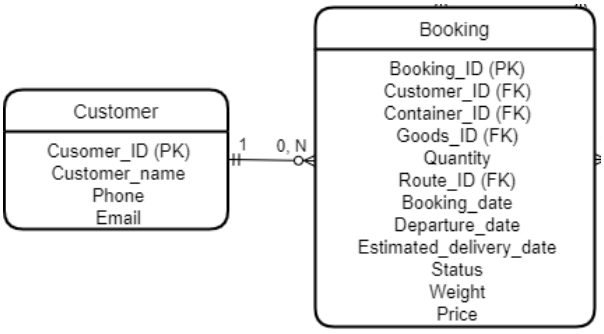


Fig 17. Customer to Booking Relationship

- A customer may place one or more booking.
- A booking must be placed by one and only one customer.

CUSTOMER_ID	CUSTOMER_NAME	PHONE
726	Tyler Larvent	915-586-3179
727	Ardyth Rope	672-758-3402
728	Latashia Goding	396-211-3174
729	Margarethe Kenna	435-828-4371
730	Brenn Master	941-407-8366

Fig 18. Customer Table highlighting Customer_id 727

BOOKING_ID	CUSTOMER_ID	CONTAINER_ID	ROUTE_ID
B6594	727	C0452	R02
B7684	727	C7120	R09

Fig 19. Booking table with Customer_id 727 showing multiplicity of relationship

- In the Fig 18 and Fig 19, one to many relationship is portrayed between the customer and booking tables.
- The customer with Customer_id has 2 bookings with different booking id.
- The figures Fig 18 and Fig 19 also represent the participation of Booking i.e., a booking must be associated with one and only one customer.

b. Container and Booking:

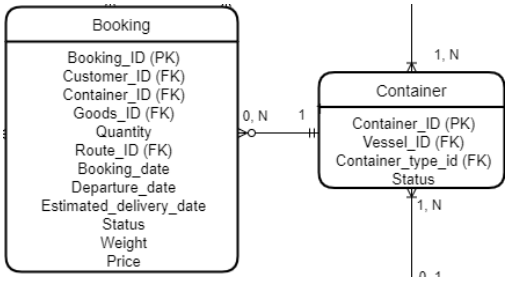


Fig 20. Container to Booking Relationship

- A booking must be associated with one and only one container.
- A container may have one or more booking.

CONTAINER_ID	VESSEL_ID	CONTAINER_TYPE_ID
C0601	V_O10	C06
C4616	V_G06	C07
C9367	V_F07	C02
C1877	V_S03	C06
C6398	V_S02	C08

Fig 21. Container Table highlighting Container_id C4616

BOOKING_ID	CUSTOMER_ID	CONTAINER_ID	ROUTE_ID	BOOKING_DATE
B4348	230	C4616	R23	02/20/2020
B3324	955	C4616	R43	06/12/2016
B9956	752	C4616	R10	07/10/2016

Fig 22. Booking table with Container_id C4616 showing multiplicity of relationship

- The figures Fig 21 and Fig 22 also represent the participation of Booking i.e., a booking must be associated with one and only one container.

c. Container and Port_Log:

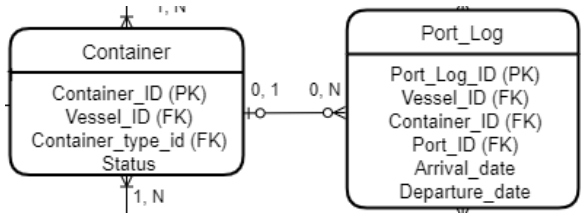


Fig 23. Container to Port_Log Relationship

- A booking may have one or more port log entries.
- A port log entry may have one and only one container.

CONTAINER_ID	VESSEL_ID	CONTAINER_TYPE_ID
C5936	V_F15	C07

Fig 24. Container Table highlighting Container_id C5936

PORT_LOG_ID	VESSEL_ID	CONTAINER_ID	PORT_ID	ARRIVAL_DATE
1	V_F15	C5936	P050	06/19/2022
2	V_F15	C5936	P051	06/20/2022
3	V_F15	C5936	P058	06/22/2022
4	V_F15	C5936	P060	06/24/2022
5	V_F15	C5936	P061	06/26/2022
6	V_F15	C5936	P059	06/28/2022
7	V_B15	C7267	P095	11/26/2022
8	V_B15	C7267	P096	11/27/2022

Fig 25. Port_Log table with Container_id C5936 showing multiplicity of relationship

- The figures Fig 24 and Fig 25 also represent the participation of container not mandatory in port_log table.

d. Vessel_Type and Vessel:

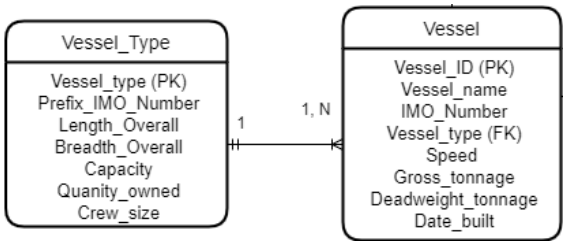


Fig 26. Vessel_Type to Vessel Relationship

- A vessel type must have one or more vessels.
- A vessel must be associated with one and only one vessel type.

VESSEL_TYPE	PREFIX_IMO_NUMBER	LENGTH_OVERALL_M	BREADTH_OVERALL_M
T	72XXXXX	368.47	51
O	78XXXXX	195	40.1
G	71XXXXX	399.98	58.8
B	77XXXXX	211.9	40.4
C	79XXXXX	172.07	38.6

Fig 27. Vessel_Type Table highlighting Vessel Type ‘T’

VESSEL_ID	VESSEL_NAME	IMO_NUMBER	VESSEL_TYPE	SPEED_KNOTS	GROSS_TONNAGE
V_T01	Ever Ark	7265846	T	23	136050
V_T02	Ever Stream	7259639	T	24	148099
V_T03	Ever Light	7260992	T	23	144825
V_T04	Ever Trader	7270619	T	24	141010
V_T05	Ever Unity	7211587	T	24	135615
V_T06	Ever Summit	7286228	T	24	143035
V_T07	Ever Fortune	7234316	T	24	144230
V_T08	Ever Dynamic	7263645	T	24	137030

Fig 28. Vesssel table with Vessel_Type ‘T’ showing multiplicity of relationship

- The figures Fig 27 and Fig 28 also represent the participation of vessel ie., a vessel must be associated with one and only one vessel type.

e. Department and Crew:

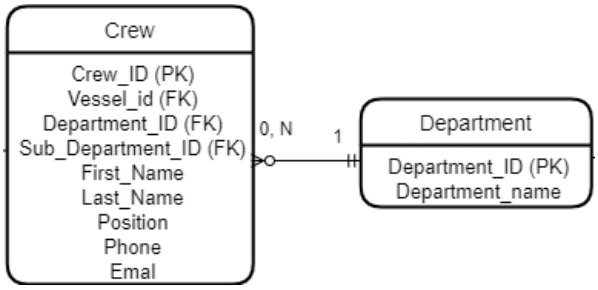


Fig 29. Department to Crew Relationship

- A department may have one or more crew.
- A crew must be associated with one and only one department.

DEPARTMENT_ID	DEPARTMENT_NAME
1	Captain
2	Deck
4	Steward's
3	Engineering

Fig 30. Department Table highlighting Department id 2

CREW_ID	VESSEL_ID	DEPARTMENT_ID	FIRST_NAME	LAST_NAME	POSITION	PHONE
M1340	V_C03	2	Eustacia	Bassom	Second Mate	860-192-0126
M3354	V_T13	2	Kurtis	Grayston	Second Mate	341-530-4779
M1542	V_C21	2	Bancroft	Guitonneau	Able Bodies Seaman	786-553-1982
M1222	V_O04	2	Alvin	Goulthorp	Third Mate	546-437-1175
M1000	V_F20	2	Lian	Rein	Chief Mate	839-764-9933
M2676	V_F18	2	Jannel	Vane	Able Bodies Seaman	511-795-5354
M4554	V_T06	2	Del	De Pero	Able Bodies Seaman	956-949-3142
M5702	V_T04	2	Jaimie	Purcell	Ordinary Seaman	328-691-4561

Fig 31. Crew table with Department id 2 showing multiplicity of relationship

- The figures Fig 30 and Fig 31 also represent the participation of crew ie., a crew must be associated with one and only one department.

f. Department and Sub_Department:

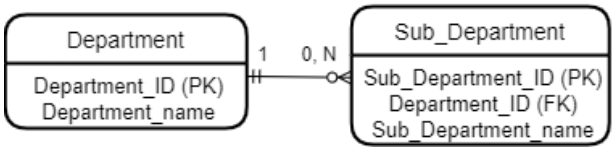


Fig 32. Department to Sub_Department Relationship

- A department may have one or more sub-departments.
- A sub-department must inclusive of one and only one department.

DEPARTMENT_ID	DEPARTMENT_NAME
1	Captain
2	Deck
4	Steward's
3	Engineering

Fig 33. Department Table highlighting Department id 2

SUB_DEPARTMENT_ID	DEPARTMENT_ID	SUB_DEPARTMENT_NAME
1	2	Officers of Watch
2	2	Ratings

Fig 34. Sub_Department table with Department id 2 showing multiplicity of relationship

- The figures Fig 33 and Fig 34 also represent the participation of sub_department ie., a sub-department must inclusive of one and only one department.

g. Vessel and Port_Log:

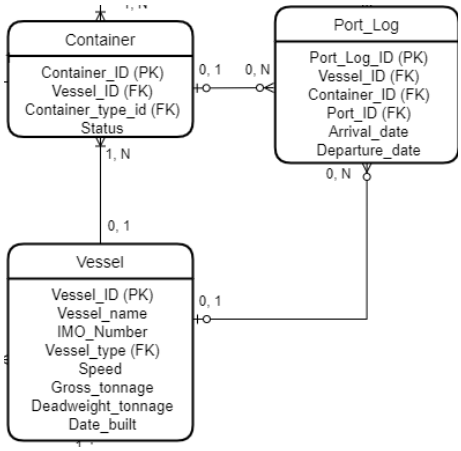


Fig 35. Vessel to Port_Log Relationship

- A vessel may have one or more port log entries.
- A port log entry may have one and only one vessel

VESSEL_ID	VESSEL_NAME	IMO_NUMBER	VESSEL_TYPE	SPEED_KNOTS
V_O11	Ever Duchess	7827022	O	23
V_C07	Ever Elixir	7975446	C	24
V_C12	Ever More	7978375	C	24
V_C17	Ever Garden	7941374	C	24
V_C22	Ever Dream	7921124	C	24
V_A01	Ever Gold	7074543	A	23
V_A02	Ever Neptune	7078296	A	23
V_A03	Ever Glow	7015254	A	23

Fig 36. Vessel Table highlighting Vessel id V_A02

PORT_LOG_ID	VESSEL_ID	CONTAINER_ID	PORT_ID
19	V_A02	C5421	P004
20	V_A02	C5421	P048
21	V_A02	C5421	P060
22	V_A02	C5421	P061
23	V_A02	C5421	P062
24	V_A02	C5421	P065

Fig 37. Port_Log table with Vessel id V_A02 showing multiplicity of relationship

- The figures Fig 36 and Fig 37 also represent the participation of vessel is not mandatory in port_log entry.

h. Port and Port_Log:

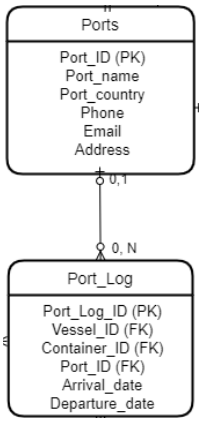


Fig 38. Port to Port_Log Relationship

- A port may have one or more entries in port log.
- A port log entry may be associated with one and only one port.

PORT_ID	PORT_NAME	PORT_COUNTRY	PHONE	EMAIL
P008	Mumbai	India	713-679-9998	ogalero0@livejournal.com
P097	Karachi	Pakistan	515-148-9921	mkoubu1@netlog.com
P036	King Fahd Industrial	Saudi Arabia	594-871-6486	alisciandri2@msu.edu
P021	Bremerhaven	Germany	924-985-2098	rpackman3@dagondesign.com
P068	Tianjin	China	898-383-4847	ckeeting4@irs.gov
P017	Piraeus	France	180-661-6283	rmcqueen5@wunderground.com

Fig 39. Port Table highlighting Port id P021

PORT_LOG_ID	VESSEL_ID	CONTAINER_ID	PORT_ID
49	V_S05	C0305	P021
76	V_C21	C9414	P021

Fig 40. Port_Log table with Port id P021 showing multiplicity of relationship

- The figures Fig 39 and Fig 40 also represent the participation of port is not mandatory in port_log entry.

i. Port and Port_Stop:

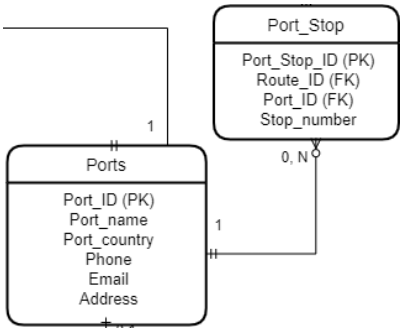


Fig 41. Port to Port_Stop Relationship

- A port may have one or more port stops.
- A port stop must be associated with one and only one port.

PORT_ID	PORT_NAME	PORT_COUNTRY	PHONE	EMAIL
P008	Mumbai	India	713-679-9998	ogalero0@livejournal.com
P097	Karachi	Pakistan	515-148-9921	mkoubu1@netlog.com
P036	King Fahd Industrial	Saudi Arabia	594-871-6486	alisciandri2@msu.edu
P021	Bremerhaven	Germany	924-985-2098	rpackman3@dagondesign.com
P068	Tianjin	China	898-383-4847	ckeeting4@irs.gov
P017	Piraeus	France	180-661-6283	rmcqueen5@wunderground.com

Fig 42. Port Table highlighting Port id P008

PORT_STOP_ID	ROUTE_ID	PORT_ID
PS004	R01	P008
PS008	R02	P008
PS034	R06	P008
PS123	R19	P008
PS210	R31	P008
PS238	R34	P008
PS250	R36	P008
PS295	R42	P008

Fig 43. Port_Stop table with Port id P008 showing multiplicity of relationship

- The figures Fig 42 and Fig 43 also represent the participation of port_stop i.e., a port stop must be associated with one and only one port.

j. Port and Route:

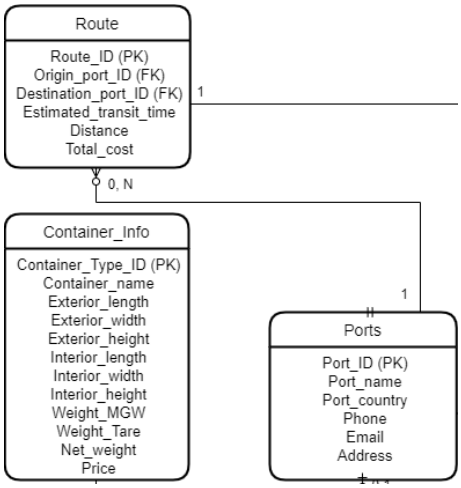


Fig 44. Port to Route Relationship

- A port may be associated with one or more routes.
- A route must have one and only one origin and destination port id.

PORT_ID	PORT_NAME	PORT_COUNTRY	PHONE	EMAIL
P008	Mumbai	India	713-679-9998	ogalero0@livejournal.com
P097	Karachi	Pakistan	515-148-9921	mkoubu1@netlog.com
P036	King Fahd Industrial	Saudi Arabia	594-871-6486	alisciandri2@msu.edu
P021	Bremerhaven	Germany	924-985-2098	rpackman3@dagondesign.com
P068	Tianjin	China	898-383-4847	ckeeting4@irs.gov
P017	Piraeus	France	180-661-6283	rmcqueen5@wunderground.com

Fig 45. Port Table highlighting Port id P008

ROUTE_ID	ORIGIN_PORT_ID	DESTINATION_PORT_ID
R61	P008	P097
R52	P008	P021
R70	P008	P068

Fig 46. Route table with Port id P008 showing multiplicity of relationship

- The figures Fig 45 and Fig 46 also represent the participation of route i.e., a route must have one and only one origin and destination port id**.**

k. Route and Booking:

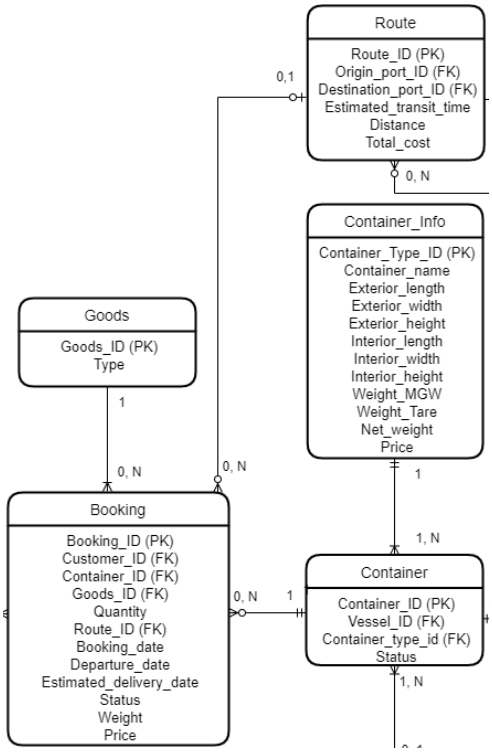


Fig 47. Route to Booking Relationship

- A route may be associated with one or more bookings.
- A booking may be associated with one and only one route

ROUTE_ID	ORIGIN_PORT_ID	DESTINATION_PORT_ID	ESTIMATED_TRANSIT_TIME
R61	P008	P097	110
R35	P058	P004	178
R60	P007	P021	130
R39	P076	P006	140
R51	P007	P097	100
R42	P066	P009	197
R11	P058	P065	35
R48	P005	P025	36

Fig 48. Route Table highlighting Route id R42

BOOKING_ID	CUSTOMER_ID	CONTAINER_ID	ROUTE_ID	BOOKING_DATE	DEPARTURE_DATE
B2811	409	C6850	R42	01/30/2013	02/23/2014
B6520	236	C6430	R42	11/05/2021	03/02/2014
B4223	393	C2715	R42	02/07/2016	09/13/2015
B5274	115	C9167	R42	09/12/2020	05/29/2022
B6288	640	C5801	R42	02/19/2021	09/03/2022
B0391	311	C1101	R42	01/29/2016	08/01/2016
B1065	999	C3808	R42	10/10/2015	10/10/2018
B6599	881	C2294	R42	06/09/2020	09/21/2020

Fig 49. Booking table with Route id R42 showing multiplicity of relationship

- The figures Fig 48 and Fig 49 also represent the participation of route i.e., a route may be associated with one or more bookings.

l. Container and Container_Info:

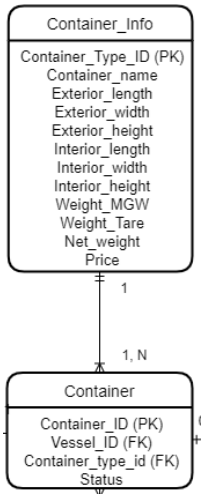


Fig 50. Container to Container_Info Relationship

- A container must be associated to one and only container type.
- A container type must be associated with one or more container.

CONTAINER_TYPE_ID	CONTAINER_NAME	EXTERIOR_LENGTH	EXTERIOR_WIDTH	EXTERIOR_HEIGHT
C01	20' Steel Dry Cargo Container	6.058	2.438	2.591
C02	40' Steel Dry Cargo Container	7.023	3.013	3.123
C05	45' Hi-Cube Steel Dry Cargo Container	6.205	2.295	3.332
C07	40' Full Height Open Top Container	6.922	2.093	3.713

Fig 51. Container_info Table highlighting Container_type_id C07

CONTAINER_ID	VESSEL_ID	CONTAINER_TYPE_ID
C4616	V_G06	C07
C2044	V_T20	C07
C7537	V_004	C07
C3391	V_G08	C07
C0571	V_C01	C07
C5942	V_C11	C07
C7229	V_B13	C07
C9258	V_T11	C07

Fig 52. Container table with Container_type_id C07 showing multiplicity of relationship

- The figures Fig 51 and Fig 52 also represent the participation of container i.e., a container must be associated to one and only container type.

m. Container and Vessel:

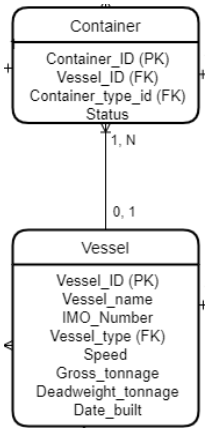


Fig 53. Container to Vessel Relationship

- A container may be associated to one and only vessel.
- A vessel must have one or more containers.

VESSEL_ID	VESSEL_NAME	IMO_NUMBER	VESSEL_TYPE	SPEED_KNOTS
V_O11	Ever Duchess	7827022	O	23
V_C07	Ever Elixir	7975446	C	24
V_C12	Ever More	7978375	C	24
V_C17	Ever Garden	7941374	C	24
V_C22	Ever Dream	7921124	C	24
V_A01	Ever Gold	7074543	A	23
V_A02	Ever Neptune	7078296	A	23
V_A03	Ever Glow	7015254	A	23

Fig 54. Vessel Table highlighting Vessel_id V_A01

CONTAINER_ID	VESSEL_ID	CONTAINER_TYPE_ID
C0174	V_A01	C11
C1021	V_A01	C04
C7926	V_A01	C08
C5387	V_A01	C01
C4496	V_A01	C02
C0366	V_A01	C05
C6108	V_A01	C03
C3594	V_A01	C04

Fig 55. Container table with Vessel_id V_A01 showing multiplicity of relationship

- The figures Fig 54 and Fig 55 also represent the participation of container and vessel i.e., a vessel must have one or more containers.

n. Booking and Goods:

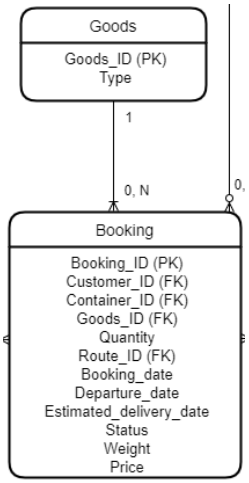


Fig 56. Booking to Goods Relationship

- A type of good may be associated with one or more bookings.
- A booking must have one and only one type of good.

GOODS_ID	TYPE
G326	Cars
G655	Trucks
G482	Motorcycles
G320	Bicycles
G303	Boats
G949	Aircraft
G749	Clothing

Fig 57. Goods Table highlighting Goods_id G326

BOOKING_ID	CUSTOMER_ID	CONTAINER_ID	ROUTE_ID	BOOKING_DATE	DEPARTURE_DATE	ESTIMATED_DELIVERY_DATE	STATUS	WEIGHT	PRICE	GOODS_ID
B6288	640	C5801	R42	02/19/2021	09/03/2022	01/01/2022	Delivered	9915	2599	G326
B2844	809	C8346	R45	07/27/2016	01/23/2022	06/06/2018	In transit	7801	4587	G326
B2755	716	C2572	R10	12/31/2019	01/24/2020	01/25/2013	Delivered	9795	9333	G326
B1304	766	C7969	R26	10/23/2014	03/28/2020	03/28/2014	In transit	8962	1958	G326
B9866	395	C4194	R23	11/13/2015	02/14/2017	11/22/2013	Delivered	631	362	G326
B3928	565	C2701	R48	03/22/2016	12/25/2017	08/15/2013	In transit	338	5241	G326
B2669	577	C7926	R07	08/08/2017	05/20/2013	04/12/2015	Delivered	3102	7555	G326

Fig 58. Booking table with Goods_id G326 showing multiplicity of relationship

- The figures Fig 57 and Fig 58 also represent the participation of booking and goods i.e., a booking must have one and only one type of good.

o. Port_Stop and Route:

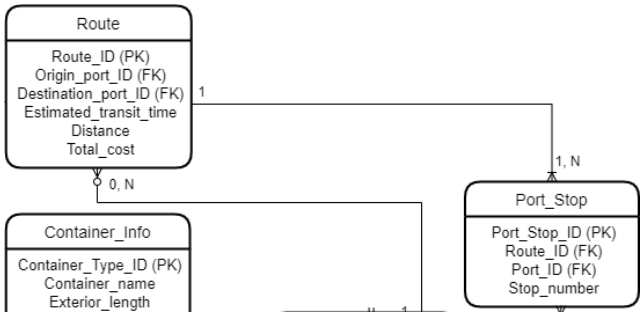


Fig 59. Port_Stop to Route Relationship

- A port stop must be associated to one and only one route.
- A route must have one or more port stops.

ROUTE_ID	ORIGIN_PORT_ID	DESTINATION_PORT_ID	ESTIMATED_TRANSIT_TIME
R61	P008	P097	110
R35	P058	P004	178
R60	P007	P021	130
R39	P076	P006	140
R51	P007	P097	100
R42	P066	P009	197
R11	P058	P065	35
R48	P005	P025	36

Fig 60. Route Table highlighting Route id R42

PORT_STOP_ID	ROUTE_ID	PORT_ID
PS291	R42	P066
PS292	R42	P068
PS293	R42	P069
PS294	R42	P097
PS295	R42	P008
PS296	R42	P012
PS297	R42	P009

Fig 61. Port_Stop table with Route id R42 showing multiplicity of relationship

- The figures Fig 60 and Fig 61 also represent the participation of route and port_stop i.e., a port stop must be associated to one and only one route.

2. Many to Many Relationship:

a. Crew and Vessel:

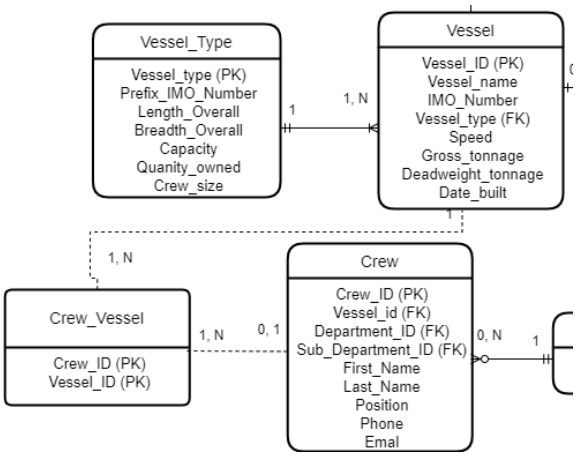


Fig 62. Crew to Vessel Relationship

- A crew may be associated with one or more vessels.

A vessel must have one or more crew member

CREW_ID	VESSEL_ID
M8545	V_B02
M8545	V_B03
M8545	V_B04
M8545	V_B05
M8545	V_B06
M8545	V_B07
M8545	V_B08
M8545	V_B09

Fig 63. Crew_Vessel Table highlighting Crew ids and Vessel id V_B02

M1542	V_B02
M1542	V_B03
M1542	V_B04
M1542	V_B05
M1542	V_B06
M1542	V_B07
M1542	V_B08
M1542	V_B09
M1542	V_B10
M1542	V_B11

Fig 64. Crew_Vessel table with another set of Crew ids and Vessel id V_B02 showing multiplicity of relationship

- In the above figures, Vessel ID is associated with many crew members and vice versa.
- The participation of crew is mandatory in a vessel.

CHAPTER 4

QUERIES THAT SHOW THE SYSTEM REQUIREMENTS

Few queries are listed below to show that the system requirements are met accordingly.

QUERY 1

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

SELECT R.ROUTE_ID, R.ORIGIN_PORT_ID, R.DESTINATION_PORT_ID, P.PORT_ID, P.PORT_COUNTRY, P.PORT_NAME, PS.PORT_STOP_ID,PS.STOP_NUMBER,R.ESTIMATED_TRANSIT_TIME

FROM ROUTE R

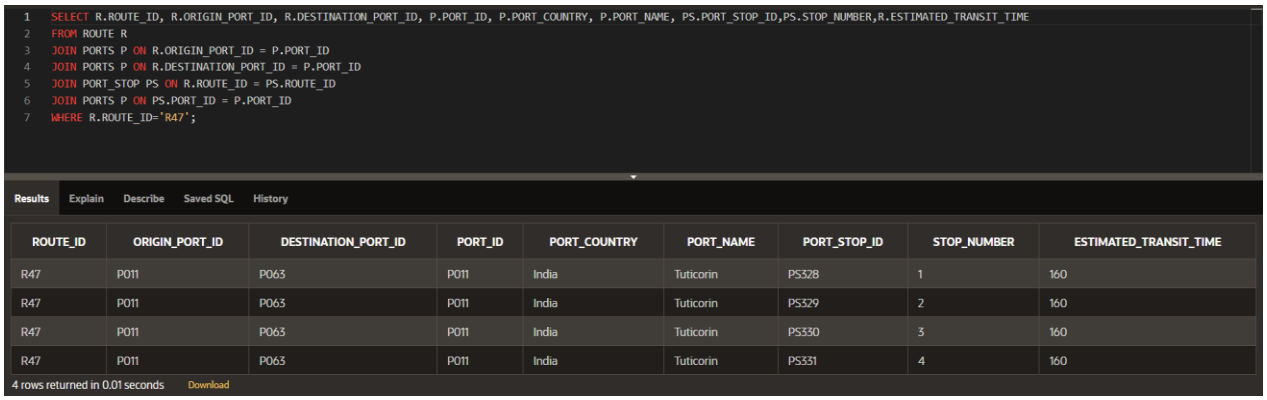
JOIN PORTS P ON R.ORIGIN_PORT_ID = P.PORT_ID

JOIN PORTS P ON R.DESTINATION_PORT_ID = P.PORT_ID

JOIN PORT_STOP PS ON R.ROUTE_ID = PS.ROUTE_ID

JOIN PORTS P ON PS.PORT_ID = P.PORT_ID

WHERE R.ROUTE_ID='R47';



```
SELECT R.ROUTE_ID, R.ORIGIN_PORT_ID, R.DESTINATION_PORT_ID, P.PORT_ID, P.PORT_COUNTRY, P.PORT_NAME, PS.PORT_STOP_ID,PS.STOP_NUMBER,R.ESTIMATED_TRANSIT_TIME
FROM ROUTE R
JOIN PORTS P ON R.ORIGIN_PORT_ID = P.PORT_ID
JOIN PORTS P ON R.DESTINATION_PORT_ID = P.PORT_ID
JOIN PORT_STOP PS ON R.ROUTE_ID = PS.ROUTE_ID
JOIN PORTS P ON PS.PORT_ID = P.PORT_ID
WHERE R.ROUTE_ID='R47';
```

Results | Explain | Describe | Saved SQL | History

ROUTE_ID	ORIGIN_PORT_ID	DESTINATION_PORT_ID	PORT_ID	PORT_COUNTRY	PORT_NAME	PORT_STOP_ID	STOP_NUMBER	ESTIMATED_TRANSIT_TIME
R47	P011	P063	P011	India	Tuticorin	PS328	1	160
R47	P011	P063	P011	India	Tuticorin	PS329	2	160
R47	P011	P063	P011	India	Tuticorin	PS330	3	160
R47	P011	P063	P011	India	Tuticorin	PS331	4	160

4 rows returned in 0.01 seconds Download

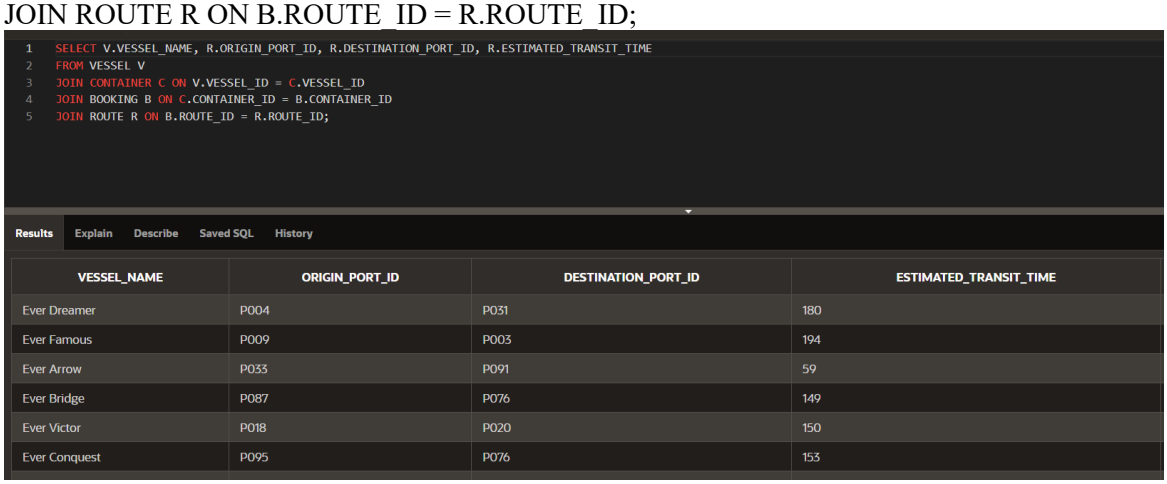
Fig 65. Result of Query 1

- The query mentioned above is designed to fetch all the information related to a particular route.
- A route consists of two essential components, namely an origin port ID and a destination port ID.
- Additionally, the Port_Stop table represents the possibility of multiple port stops being associated with a single route.
- By joining all the relevant tables, a comprehensive view of the service routing network can be obtained.

QUERY 2

Enable customers to search for sailing schedules.

```
SELECT V.VESSEL_NAME, R.ORIGIN_PORT_ID, R.DESTINATION_PORT_ID,
R.ESTIMATED_TRANSIT_TIME

FROM VESSEL V

JOIN CONTAINER C ON V.VESSEL_ID = C.VESSEL_ID

JOIN BOOKING B ON C.CONTAINER_ID = B.CONTAINER_ID

JOIN ROUTE R ON B.ROUTE_ID = R.ROUTE_ID;
```



Results | Explain | Describe | Saved SQL | History

VESSEL_NAME	ORIGIN_PORT_ID	DESTINATION_PORT_ID	ESTIMATED_TRANSIT_TIME
Ever Dreamer	P004	P031	180
Ever Famous	P009	P003	194
Ever Arrow	P033	P091	59
Ever Bridge	P087	P076	149
Ever Victor	P018	P020	150
Ever Conquest	P095	P076	153

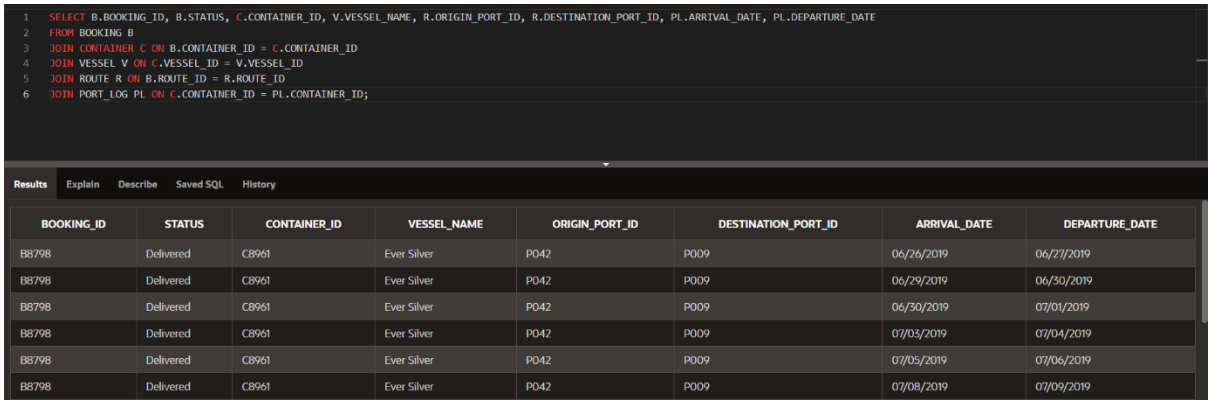
Fig 66. Result of Query 2

- The data resulted allows customers to search for sailing schedules on a specific route.
- The information displayed pertains only to containers that have been booked, ensuring that customers can view accurate sailing schedules.

QUERY 3

Enable customers to track cargo.

```
SELECT B.BOOKING_ID, B.STATUS, C.CONTAINER_ID, V.VESSEL_NAME,
R.ORIGIN_PORT_ID, R.DESTINATION_PORT_ID, PL.ARRIVAL_DATE,
PL.DEPARTURE_DATE

FROM BOOKING B

JOIN CONTAINER C ON B.CONTAINER_ID = C.CONTAINER_ID

JOIN VESSEL V ON C.VESSEL_ID = V.VESSEL_ID

JOIN ROUTE R ON B.ROUTE_ID = R.ROUTE_ID

JOIN PORT_LOG PL ON C.CONTAINER_ID = PL.CONTAINER_ID;
```



Results | Explain | Describe | Saved SQL | History

BOOKING_ID	STATUS	CONTAINER_ID	VESSEL_NAME	ORIGIN_PORT_ID	DESTINATION_PORT_ID	ARRIVAL_DATE	DEPARTURE_DATE
B8798	Delivered	C8961	Ever Silver	P042	P009	06/26/2019	06/27/2019
B8798	Delivered	C8961	Ever Silver	P042	P009	06/29/2019	06/30/2019
B8798	Delivered	C8961	Ever Silver	P042	P009	06/30/2019	07/01/2019
B8798	Delivered	C8961	Ever Silver	P042	P009	07/03/2019	07/04/2019
B8798	Delivered	C8961	Ever Silver	P042	P009	07/05/2019	07/06/2019
B8798	Delivered	C8961	Ever Silver	P042	P009	07/08/2019	07/09/2019

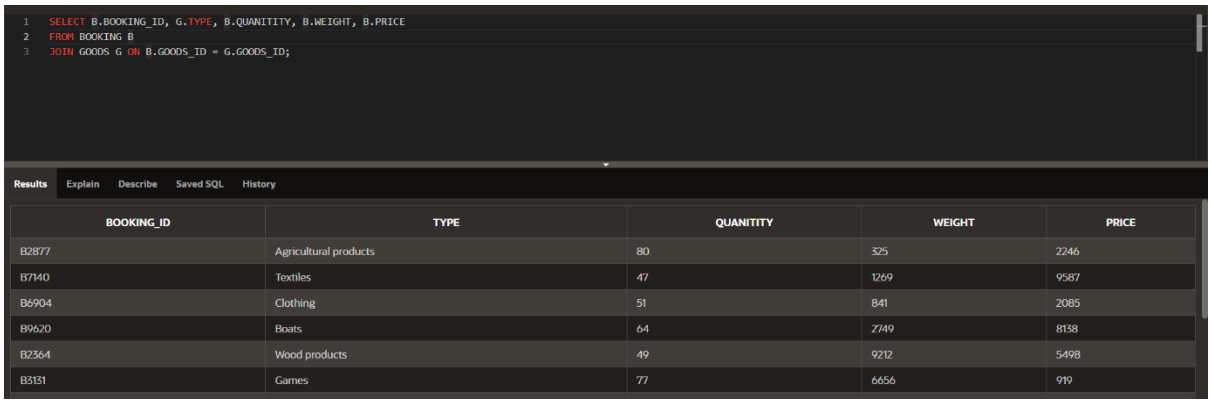
Fig 67. Result of Query 3

- The query facilitates customers in tracking their respective bookings based on their Booking_ID.
- Customers can access comprehensive information regarding their bookings, including the booking's status, container ID, and the vessel in which the container is placed.
- Moreover, customers can track their cargo's arrival and departure dates at each port, allowing for seamless cargo tracking.

QUERY 4

Record details of goods conveyed

```
SELECT B.BOOKING_ID, G.TYPE, B.QUANITITY, B.WEIGHT, B.PRICE
FROM BOOKING B
JOIN GOODS G ON B.GOODS_ID = G.GOODS_ID;
```



Results | Explain | Describe | Saved SQL | History

BOOKING_ID	TYPE	QUANITITY	WEIGHT	PRICE
B2877	Agricultural products	80	325	2246
B7140	Textiles	47	1269	9587
B6904	Clothing	51	841	2085
B9620	Boats	64	7749	8138
B2364	Wood products	49	9212	5498
B3131	Games	77	6656	919

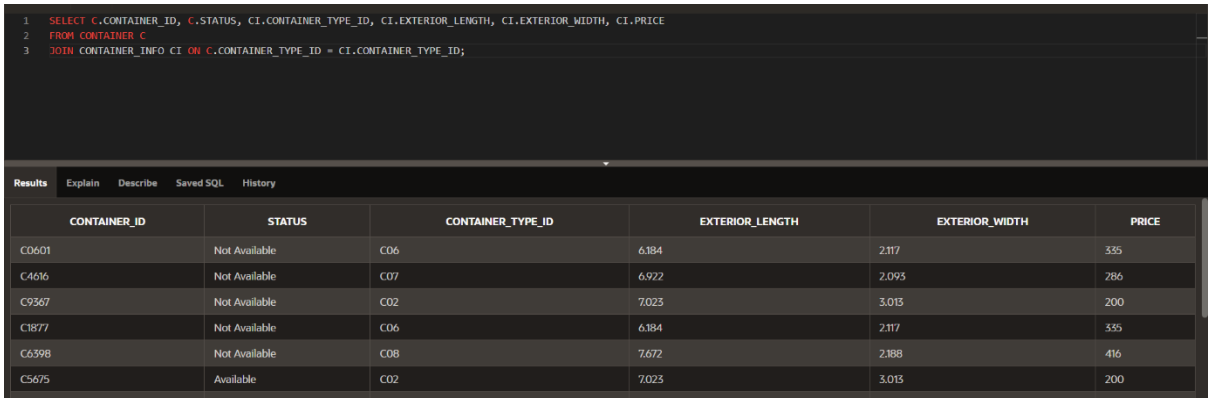
Fig 68. Result of Query 4

- The above query serves to maintain a comprehensive record of all goods booked by customers.
- This facilitates the identification of frequently requested goods and can provide valuable insights.
- Additionally, customers can verify that the goods shipped correspond to the type of goods booked.

QUERY 5

Enable customers to search for containers and book containers.

```
SELECT C.CONTAINER_ID, C.STATUS, CI.CONTAINER_TYPE_ID, CI.EXTERIOR_LENGTH,
CI.EXTERIOR_WIDTH, CI.PRICE

FROM CONTAINER C

JOIN CONTAINER_INFO CI ON C.CONTAINER_TYPE_ID = CI.CONTAINER_TYPE_ID;
```



Results | Explain | Describe | Saved SQL | History

CONTAINER_ID	STATUS	CONTAINER_TYPE_ID	EXTERIOR_LENGTH	EXTERIOR_WIDTH	PRICE
C0601	Not Available	C06	6.184	2.117	335
C4616	Not Available	C07	6.922	2.093	286
C9367	Not Available	C02	7.023	3.013	200
C1877	Not Available	C06	6.184	2.117	335
C6398	Not Available	C08	7.672	2.188	416
C5675	Available	C02	7.023	3.013	200

Fig 69. Result of Query 5

- The query above allows customers to check the availability of containers.
- Containers that have been booked but not yet delivered are considered unavailable.
- With this query, customers can monitor the status of container availability.

QUERY 6

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

SELECT V.VESSEL_ID, V.VESSEL_NAME, R.ORIGIN_PORT_ID, R.DESTINATION_PORT_ID, R.ESTIMATED_TRANSIT_TIME, SUM(B.WEIGHT) AS TOTAL_WEIGHT

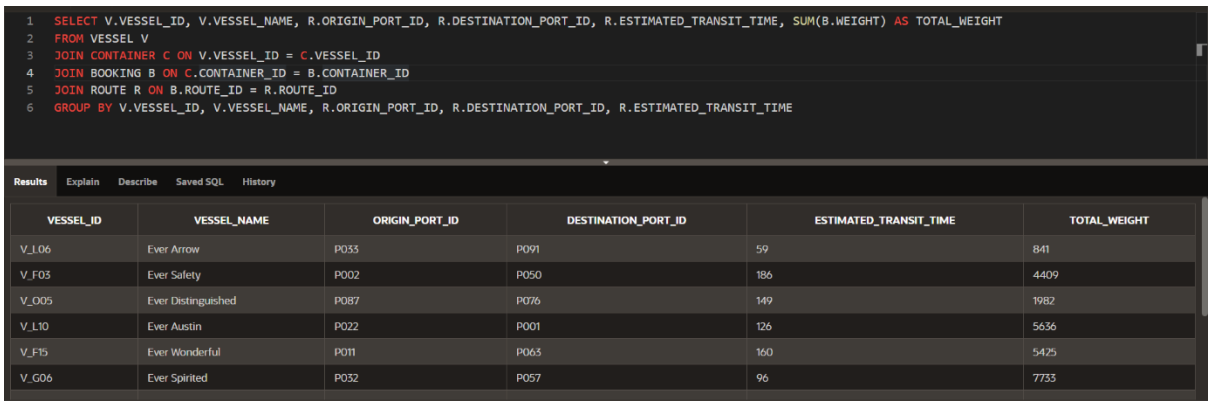
FROM VESSEL V

JOIN CONTAINER C ON V.VESSEL_ID = C.VESSEL_ID

JOIN BOOKING B ON C.CONTAINER_ID = B.CONTAINER_ID

JOIN ROUTE R ON B.ROUTE_ID = R.ROUTE_ID

GROUP BY V.VESSEL_ID, V.VESSEL_NAME, R.ORIGIN_PORT_ID, R.DESTINATION_PORT_ID, R.ESTIMATED_TRANSIT_TIME



```
1  SELECT V.VESSEL_ID, V.VESSEL_NAME, R.ORIGIN_PORT_ID, R.DESTINATION_PORT_ID, R.ESTIMATED_TRANSIT_TIME, SUM(B.WEIGHT) AS TOTAL_WEIGHT
2  FROM VESSEL V
3  JOIN CONTAINER C ON V.VESSEL_ID = C.VESSEL_ID
4  JOIN BOOKING B ON C.CONTAINER_ID = B.CONTAINER_ID
5  JOIN ROUTE R ON B.ROUTE_ID = R.ROUTE_ID
6  GROUP BY V.VESSEL_ID, V.VESSEL_NAME, R.ORIGIN_PORT_ID, R.DESTINATION_PORT_ID, R.ESTIMATED_TRANSIT_TIME
```

Results | Explain | Describe | Saved SQL | History

VESSEL_ID	VESSEL_NAME	ORIGIN_PORT_ID	DESTINATION_PORT_ID	ESTIMATED_TRANSIT_TIME	TOTAL_WEIGHT
V_L06	Ever Arrow	P033	P091	59	841
V_F03	Ever Safety	P002	P050	186	4409
V_O05	Ever Distinguished	P087	P076	149	1982
V_L10	Ever Austin	P022	P001	126	5636
V_F15	Ever Wonderful	P011	P063	160	5425
V_G06	Ever Spirited	P032	P057	96	7733

Fig 70. Result of Query 6

- The above query optimizes the allocation of cargo for the transportation of goods, ensuring that containers are assigned to vessels in a way that maximizes efficiency and minimizes the risk of overloading.
- This process considers the weight of the cargo and ensures that containers are allocated to vessels in a balanced manner.

ADDITIONAL QUERIES:

1. **--Query to track the container which results in the arrival and departure dates at each port**

--Date format is in mm-dd-yyyy format

```
SELECT DISTINCT PL.CONTAINER_ID, PL.PORT_ID, PL.ARRIVAL_DATE,
PL.DEPARTURE_DATE

FROM PORT_LOG PL, ROUTE R, PORT_STOP PS

WHERE R.ROUTE_ID=PS.ROUTE_ID AND CONTAINER_ID='C5936';
```

CONTAINER_ID	PORT_ID	ARRIVAL_DATE	DEPARTURE_DATE
C5936	P050	06/19/2022	06/19/2022
C5936	P051	06/20/2022	06/21/2022
C5936	P059	06/28/2022	-
C5936	P061	06/26/2022	06/27/2022
C5936	P060	06/24/2022	06/24/2022
C5936	P058	06/22/2022	06/22/2022

Fig 71. Result of additional query 1

2. **--Query to calculate the total weight on a particular vessel i.e., weight of the goods and weight of container**

```
SELECT V.VESSEL_ID, V.DEADWEIGHT_TONNAGE AS TOTAL_WEIGHT_OF_VESSEL,
SUM(CI.NET_WEIGHT + B.WEIGHT) AS TOTAL_WEIGHT

FROM CONTAINER C

JOIN VESSEL V ON C.VESSEL_ID = V.VESSEL_ID

JOIN CONTAINER_INFO CI ON C.CONTAINER_TYPE_ID = CI.CONTAINER_TYPE_ID

JOIN BOOKING B ON C.CONTAINER_ID = B.CONTAINER_ID

WHERE C.VESSEL_ID = 'V_F10'

GROUP BY V.VESSEL_ID, V.DEADWEIGHT_TONNAGE;
```

VESSEL_ID	TOTAL_WEIGHT_OF_VESSEL	TOTAL_WEIGHT
V_F10	524769	405149

Fig 72. Result of additional query 2

3. **--Query results in the most busiest port with the arrival of number of containers**

```
SELECT PL.PORT_ID, P.PORT_NAME, COUNT(PL.CONTAINER_ID) AS
NUM_CONTAINERS

FROM PORT_LOG PL

JOIN PORTS P ON PL.PORT_ID = P.PORT_ID

GROUP BY PL.PORT_ID, P.PORT_NAME

ORDER BY NUM_CONTAINERS DESC

FETCH FIRST 1 ROWS ONLY;
```

PORT_ID	PORT_NAME	NUM_CONTAINERS
P020	Salerno	5

Fig 73. Result of additional query 3

4. **--Query results the list of customers who have booked goods of more weight for transportation**

```
SELECT C.CUSTOMER_NAME, SUM(B.WEIGHT) AS TOTAL_WEIGHT

FROM BOOKING B

JOIN CUSTOMER C ON B.CUSTOMER_ID = C.CUSTOMER_ID

WHERE B.BOOKING_DATE BETWEEN TO_DATE('01-JAN-2013', 'DD-MON-YYYY') AND
TO_DATE('31-DEC-2022', 'DD-MON-YYYY')

GROUP BY C.CUSTOMER_NAME

ORDER BY TOTAL_WEIGHT DESC

FETCH FIRST 10 ROWS ONLY;
```

CUSTOMER_NAME	TOTAL_WEIGHT
Ammamaria Pessler	34963
Ricki Allibon	27009
Cindelyn Bossingham	25304
Mechelle Ennor	25022

Fig 74. Result of additional query 4

5. **--Query to retrieve the total number of containers on each vessel and sort the result in descending order of total number of containers.**

```
SELECT V.VESSEL_NAME, V.IMO_NUMBER, COUNT(C.CONTAINER_ID) AS
TOTAL_CONTAINERS

FROM VESSEL V

JOIN CONTAINER C ON C.VESSEL_ID = V.VESSEL_ID

GROUP BY V.VESSEL_NAME, V.IMO_NUMBER

ORDER BY TOTAL_CONTAINERS DESC;
```

VESSEL_NAME	IMO_NUMBER	TOTAL_CONTAINERS
Ever True	7354490	13
Ever Loyal	7947563	13
Ever Luminous	7266626	12
Ever Omega	7115404	12

Fig 75. Result of additional query 5

6. **--Query to show the list of all customers whose containers passed Shangai port**

```
SELECT DISTINCT C.CUSTOMER_NAME

FROM CUSTOMER C
```

```
JOIN BOOKING B ON B.CUSTOMER_ID = C.CUSTOMER_ID
JOIN ROUTE R ON R.ROUTE_ID = B.ROUTE_ID
JOIN PORT_STOP PS ON PS.ROUTE_ID = R.ROUTE_ID
JOIN PORTS P ON P.PORT_ID = PS.PORT_ID
WHERE P.PORT_NAME = 'Shanghai';
```

CUSTOMER_NAME
Allison McGloughlin
Dierdre Phillcock
Myrvyn Besemer
Steve Stolberg

Fig 76. Result of additional query 6

7. **--Query to retrieve all the vessels passing through Singapore port**

```
SELECT V.VESSEL_ID, V.VESSEL_NAME, P.PORT_NAME
FROM VESSEL V
JOIN CONTAINER C ON C.VESSEL_ID = V.VESSEL_ID
JOIN BOOKING B ON B.CONTAINER_ID = C.CONTAINER_ID
JOIN ROUTE R ON R.ROUTE_ID = B.ROUTE_ID
JOIN PORT_STOP PS ON PS.ROUTE_ID = R.ROUTE_ID
JOIN PORTS P ON P.PORT_ID = PS.PORT_ID
WHERE P.PORT_NAME = 'Singapore'
GROUP BY V.VESSEL_ID, V.VESSEL_NAME, P.PORT_NAME;
```

VESSEL_ID	VESSEL_NAME	PORT_NAME
V_T07	Ever Fortune	Singapore
V_F03	Ever Safety	Singapore
V_A07	Ever Royal	Singapore
V_T15	Ever Famous	Singapore

Fig 77. Result of additional query 7

8. **--Query to find the most common goods shipped by each customer**

```
SELECT C.CUSTOMER_ID, C.CUSTOMER_NAME, G.TYPE AS
MOST_COMMON_GOODS
FROM CUSTOMER C
LEFT JOIN BOOKING B ON C.CUSTOMER_ID = B.CUSTOMER_ID
LEFT JOIN GOODS G ON B.GOODS_ID = G.GOODS_ID
GROUP BY C.CUSTOMER_ID, C.CUSTOMER_NAME, G.TYPE
HAVING COUNT(*) = (
```

```
  SELECT MAX(SUB_COUNT)
  FROM (
    SELECT COUNT(*) AS SUB_COUNT
    FROM BOOKING B2
    INNER JOIN GOODS G2 ON B2.GOODS_ID = G2.GOODS_ID
    WHERE B2.CUSTOMER_ID = C.CUSTOMER_ID
    GROUP BY G2.TYPE
  )
)
```

CUSTOMER_ID	CUSTOMER_NAME	MOST_COMMON_GOODS
727	Ardyth Rope	Leather products
734	Kirstin Keeton	Heavy equipment
745	Allison McGloughlin	Home appliances
749	Taffy Pigeon	Heavy equipment

Fig 78. Result of additional query 8

9. **--Query to find the total number of containers of each type that passed through each port**

```
SELECT P.PORT_ID, CI.CONTAINER_TYPE_ID, COUNT(*) AS NUMBER_CONTAINERS
FROM PORT_LOG PL
INNER JOIN CONTAINER C ON PL.CONTAINER_ID = C.CONTAINER_ID
INNER JOIN CONTAINER_INFO CI ON C.CONTAINER_TYPE_ID =
CI.CONTAINER_TYPE_ID
INNER JOIN PORTS P ON PL.PORT_ID = P.PORT_ID
GROUP BY P.PORT_ID, CI.CONTAINER_TYPE_ID;
```

PORT_ID	CONTAINER_TYPE_ID	NUMBER_CONTAINERS
P008	C08	1
P097	C08	1
P096	C09	2
P016	C06	1

Fig 79. Result of additional query 9

CHAPTER 6
CONCLUSION

EVALUATION OF THE PROJECT

- The database project aimed to design a system for a British shipping company to manage their service routing network, cargo tracking, allocation, and customer bookings. The project required significant effort to design an efficient database schema and queries to retrieve the necessary information.
- One of the main challenges faced was managing the service routes, which required frequent updates. Additionally, data redundancy posed a significant challenge, particularly in relation to routes and ports. To overcome this, a Port_Stop table was implemented to manage ports effectively. The weight of containers was also a challenge, and ensuring that vessels were not overloaded was essential.
- Throughout the project, I learned the importance of minimizing data redundancy, as this allowed us to load data more effectively. I also used Mockaroo to generate regular expressions for all the identifiers (IDs) in the database, which was a useful and fun tool to work with.
- One limitation of the current design is scalability concerns, as the database may not be able to handle larger volumes of data in the future. Additionally, further improvements could be made to enhance the usability and user-friendliness of the database, such as implementing more intuitive interfaces for users. However, overall the project was a success, and has the potential to significantly impact the operations of the shipping company by providing more efficient management of their logistics.

SELF-EVALUATION

Criterion	***Pratiksha Rajendran*** ***(Marks 0-5)***
Multiplicity of data	4
Table creation	5
Entity Relationship Diagram	4
Queries to meet the requirements	4

WHAT DID I THINK OF THE ASSIGNMENT

- The database assignment proved to be a fascinating experience that expanded my knowledge of the shipping industry. Initially, designing the database schema seemed straightforward, but as I delved deeper into the concepts, I realized the challenges involved. The initial design lacked

normalization and needed frequent revisions, which ultimately resulted in an effective database design for a shipping company. Through this project, I learned the importance of normalization and reducing data redundancy to ensure a more efficient system.

- Furthermore, the assignment allowed me to gain a deeper understanding of the logistics involved in managing the service routing network, tracking cargo, and allocating it efficiently. I also gained insights into the difficulties of managing customer bookings and ensuring vessels were not overloaded, which required careful consideration of container weight. Overall, the project provided me with valuable skills and knowledge that will be useful in future database design and management projects.
- The assignment provided an opportunity to apply theoretical concepts learned in class to a practical problem. It was a great way to consolidate knowledge gained from lectures and coursework.
- As the assignment progressed, it became apparent that managing a shipping company involves a complex system that requires careful planning and organization. Designing a database for this purpose was not a trivial task, but it was rewarding to see how each change and modification improved the efficiency and effectiveness of the database.
- Lastly, the assignment was a great way to develop technical skills in database design and management, which are highly valued in the industry. It provided hands-on experience in designing and implementing a database, which can be a valuable asset in a future career.

WHAT WENT WELL AND WHAT DID NOT

- The database assignment had some notable successes and challenges.
- The ER diagram was effectively designed after multiple revisions, indicating a mastery of the concept.
- However, loading data into certain tables such as Port_Log proved to be a more challenging task due to the complexity of the data to be entered.
- During the process of loading data into the Booking table, which involved inputting the booking date, departure date, and estimated delivery date using Mockaroo, it became evident that the dates were not correctly related to each other in a way that was necessary for an optimal database schema. This was a huge task for me.
- Despite this difficulty, a comprehensive and efficient database design was achieved, meeting the requirements of the shipping company.

WHAT WOULD I DO DIFFERENTLY

- To design an effective database for a shipping company, it is important to have a clear understanding of the company's requirements from the beginning. This can help identify potential issues and prevent time-consuming changes in the future.
- Moreover, including containers of smaller dimensions for smaller enterprises can be beneficial as it allows for the transportation of smaller quantities of goods.