

**KINGSTON UNIVERSITY LONDON**

**DATABASE DESIGN (CI7320)**

**COURSEWORK - I**

NAME	SUJAY GRAMA SURESH KUMAR
KU ID	K2201621
EMAIL ID	K2201621@KINGSTON.AC.UK
PROFESSOR NAME	Dr. BERYL JONES

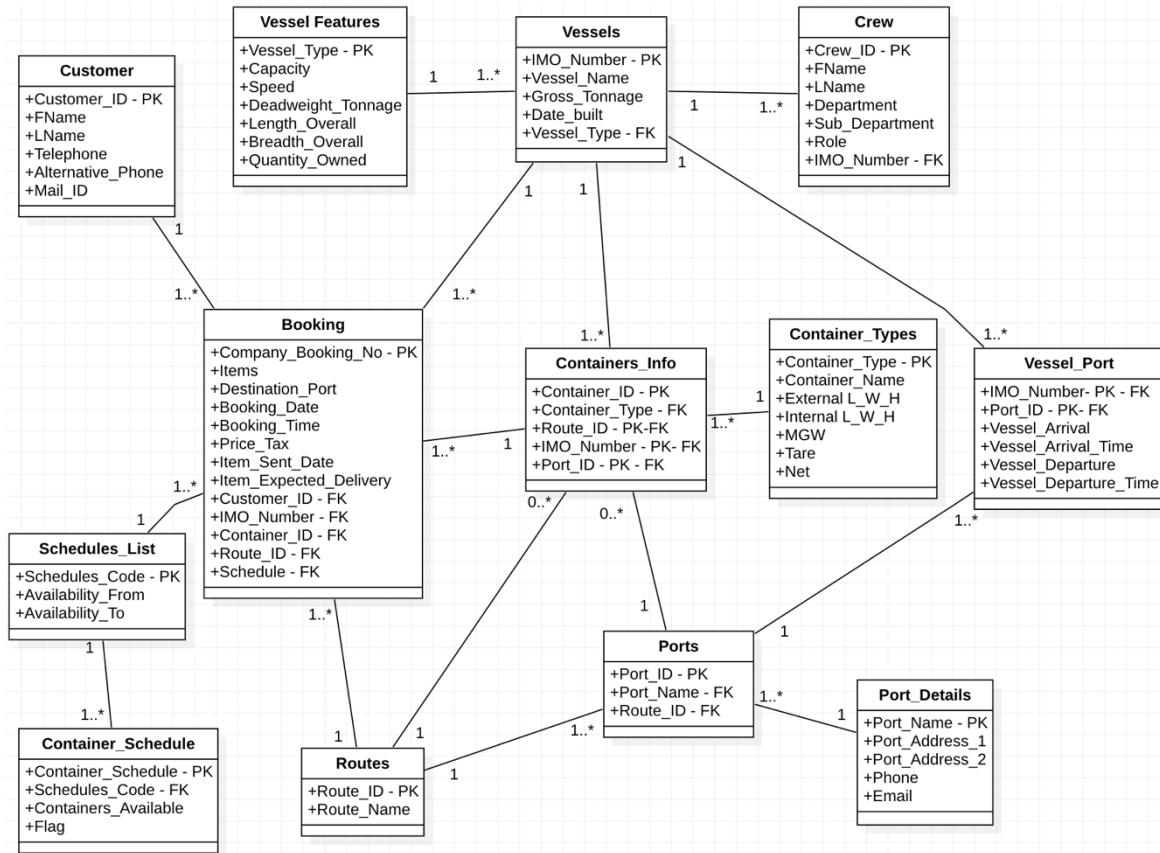
## **TABLE OF CONTENTS**

<b>Chapter No.</b>	<b>Title</b>	<b>Page No.</b>
1	<b>ER DIAGRAM WITH CONSTRAINTS AND ASSUMPTIONS</b>	
1.2	Constraints and Assumptions	1-2
2	<b>IMPLEMENTATION</b>	
2.1	Table definitions with object browser tables in Oracle Apex	3-11
3	<b>DISCUSSION ON THE DATASET</b>	
3.1	Database showing the multiplicity of the relationship	12-17
4	<b>SQL QUERIES</b>	<b>18-26</b>
5	<b>CONCLUSION</b>	<b>27-28</b>

## CHAPTER 1

### ER DIAGRAM WITH CONSTRAINTS AND ASSUMPTIONS

#### 1.1 ENTITY RELATIONSHIP DIAGRAM



KU ID - K2201621  
Sujay Grama Suresh Kumar

Fig. 1 Entity-Relationship Diagram

#### 1.2 CONSTRAINTS AND ASSUMPTIONS

- Each Customer\_ID can have one or many Booking\_No., but each Booking\_No. can only belong to one Customer\_ID for one Item.
- Each Vessel\_Type can have one or many IMO\_Number(Vessels), but each IMO\_Number (Vessel) can only belong to one Vessel\_Type.
- Each IMO\_Number (Vessel) can have one or many Crew\_ID(Crew Members), but each Crew\_ID(Crew Member) can belong to only one IMO\_Number(Vessel).
- Each IMO\_Number(Vessel) can have one or many Booking\_No., but each Booking\_No. can only belong to one IMO\_Number (Vessel).
- Each Container\_Type can belong to one or many Containers, but each Container can only belong to one Container\_Type.

- Each IMO\_Number (Vessel) can be associated with zero or many Ports, and each Port can be associated with zero or many IMO\_Number (Vessels). This allows for many-to-many relationships between Vessels and Ports.
- Each Port\_Name can be associated with one or many Port\_IDs, but each Port\_ID can only be associated with one Port\_Name.
- Each Route\_ID can have one or many Port\_ID, but each Port\_ID can have only one Route\_ID.
- Each Schedule\_Code in Schedule List table can be associated with one or many Container\_Schedules, but each Container\_Schedule can only be associated with one Schedule\_Code.
- Each Schedule\_Code in Schedule List table can be associated with one or many Booking\_No., but each Booking\_No. can only be associated with one Schedule\_Code.
- Each Vessel can have one or many Routes, and each Route can have one or more Vessels. This allows many-to-many relationship between Vessels and Routes.
- Each Vessel can have one or many Containers, but each container can only belong to one Vessel.
- Each Route can have Zero or many Containers, but each Container can be travelled only in one Route.
- Each Port can have one or many Containers, but one Container can be delivered to only one Port.

## CHAPTER 2

### IMPLEMENTATION

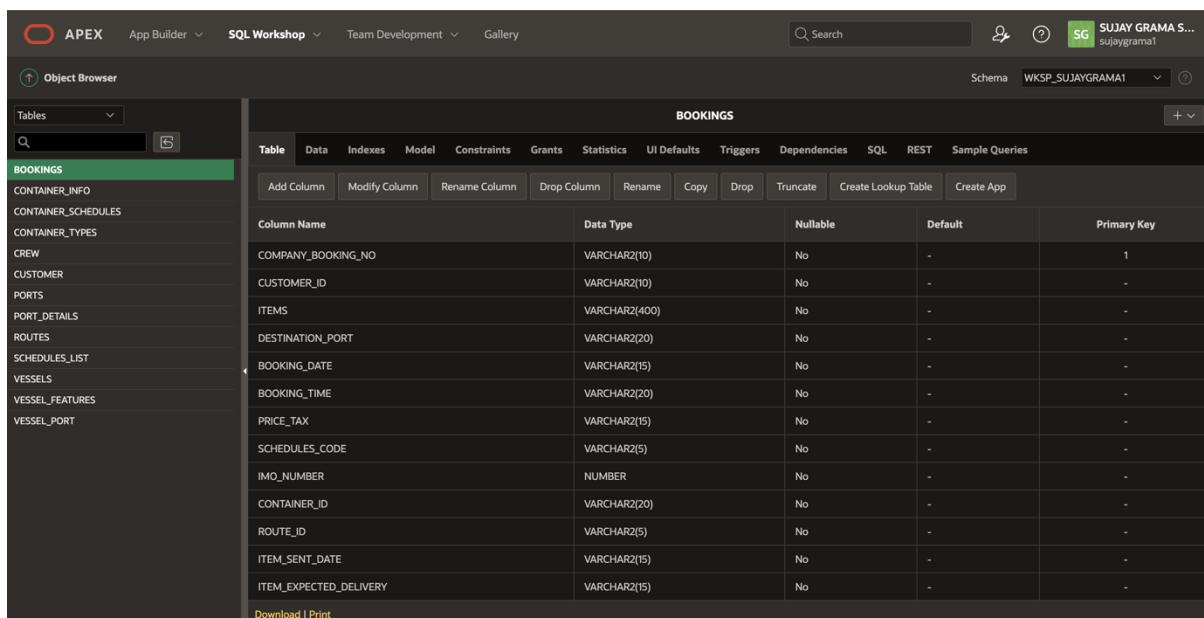
#### 2.1 TABLE DEFINITIONS WITH OBJECT BROWSER TABLES IN ORACLE APEX

##### 2.1.1 BOOKINGS

```
CREATE TABLE BOOKINGS (
    COMPANY_BOOKING_NO VARCHAR(10) PRIMARY KEY,
    CUSTOMER_ID VARCHAR(10) NOT NULL,
    ITEMS VARCHAR(400) NOT NULL,
    DESTINATION_PORT VARCHAR(20) NOT NULL,
    BOOKING_DATE VARCHAR(15) NOT NULL,
    BOOKING_TIME VARCHAR(20) NOT NULL,
    PRICE_TAX VARCHAR(15) NOT NULL,
    SCHEDULES_CODE VARCHAR(5) NOT NULL,
    IMO_NUMBER NUMBER NOT NULL,
    CONTAINER_ID VARCHAR(20) NOT NULL,
    ROUTE_ID VARCHAR(5) NOT NULL,
    ITEM_SENT_DATE VARCHAR(15) NOT NULL,
    ITEM_EXPECTED_DELIVERY VARCHAR(15) NOT NULL,
    FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID),
    FOREIGN KEY (SCHEDULES_CODE) REFERENCES SCHEDULES_LIST(SCHEDULES_CODE),
    FOREIGN KEY (IMO_NUMBER) REFERENCES VESSELS(IMO_NUMBER),
    FOREIGN KEY (CONTAINER_ID) REFERENCES CONTAINER_INFO(CONTAINER_ID),
    FOREIGN KEY (ROUTE_ID) REFERENCES ROUTES(ROUTE_ID)
);
```

**SQL Query-1**

- SQL Query 1 represents the SQL table definitions for Bookings.



The screenshot shows the Oracle APEX Object Browser interface. On the left, there is a sidebar with a tree view of database objects: CONTAINER\_INFO, CONTAINER\_SCHEDULES, CONTAINER\_TYPES, CREW, CUSTOMER, PORTS, PORT\_DETAILS, ROUTES, SCHEDULES\_LIST, VESSELS, VESSEL\_FEATURES, and VESSEL\_PORT. The 'BOOKINGS' table is selected and highlighted in green. The main panel displays the 'BOOKINGS' table definition with the following columns:

Column Name	Data Type	Nullable	Default	Primary Key
COMPANY_BOOKING_NO	VARCHAR2(10)	No	-	1
CUSTOMER_ID	VARCHAR2(10)	No	-	-
ITEMS	VARCHAR2(400)	No	-	-
DESTINATION_PORT	VARCHAR2(20)	No	-	-
BOOKING_DATE	VARCHAR2(15)	No	-	-
BOOKING_TIME	VARCHAR2(20)	No	-	-
PRICE_TAX	VARCHAR2(15)	No	-	-
SCHEDULES_CODE	VARCHAR2(5)	No	-	-
IMO_NUMBER	NUMBER	No	-	-
CONTAINER_ID	VARCHAR2(20)	No	-	-
ROUTE_ID	VARCHAR2(5)	No	-	-
ITEM_SENT_DATE	VARCHAR2(15)	No	-	-
ITEM_EXPECTED_DELIVERY	VARCHAR2(15)	No	-	-

**Table-1**

- Table 1 represents all the attributes present in the Bookings table.

## 2.1.2 CONTAINER\_INFO

```
CREATE TABLE CONTAINER_INFO (
    CONTAINER_ID VARCHAR(20) PRIMARY KEY,
    CONTAINER_TYPE VARCHAR(10) NOT NULL,
    IMO_NUMBER NUMBER NOT NULL,
    ROUTE_ID VARCHAR(5) NOT NULL,
    PORT_ID VARCHAR(10) NOT NULL,
    CHOSEN_CONTAINER VARCHAR(4) NOT NULL,
    FOREIGN KEY (CONTAINER_TYPE) REFERENCES CONTAINER_TYPES(CONTAINER_TYPE),
    FOREIGN KEY (IMO_NUMBER) REFERENCES VESSELS(IMO_NUMBER),
    FOREIGN KEY (ROUTE_ID) REFERENCES ROUTES(ROUTE_ID),
    FOREIGN KEY (PORT_ID) REFERENCES PORTS(PORT_ID)
);
```

**SQL Query-2**

- SQL Query 1 represents the SQL table definitions for Container\_Info.

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. A search bar and user information ('SUJAY GRAMA S... sujaygrama1') are also present. The main area is titled 'CONTAINER\_INFO' and displays its schema. The table has columns: CONTAINER\_ID (VARCHAR2(20)), CONTAINER\_TYPE (VARCHAR2(10)), IMO\_NUMBER (NUMBER), ROUTE\_ID (VARCHAR2(5)), PORT\_ID (VARCHAR2(10)), and CHOSEN\_CONTAINER (VARCHAR2(4)). Primary key constraints are defined for CONTAINER\_TYPE, IMO\_NUMBER, and PORT\_ID. Buttons for 'Add Column', 'Modify Column', 'Rename Column', etc., are visible at the top of the table definition.

**Table-2**

- Table 2 represents all the attributes present in the Container Info.

## 2.1.3 CONTAINER\_SCHEDULES

```
CREATE TABLE CONTAINER_SCHEDULES (
    CONTAINER_SCHEDULE VARCHAR(10) PRIMARY KEY,
    SCHEDULES_CODE VARCHAR(5) NOT NULL,
    CONTAINERS_AVAILABLE VARCHAR(10) NULL,
    FLAG NUMBER NULL,
    FOREIGN KEY (SCHEDULES_CODE) REFERENCES SCHEDULES_LIST(SCHEDULES_CODE)
);
```

**SQL Query-3**

- SQL Query 1 represents the SQL table definitions for Container\_Schedules.

The screenshot shows the Oracle SQL Developer Object Browser. On the left, under the 'Tables' section, 'CONTAINER\_SCHEDULES' is selected. The main panel displays the 'CONTAINER\_SCHEDULES' table structure with the following columns:

Column Name	Data Type	Nullable	Default	Primary Key
CONTAINER_SCHEDULE	VARCHAR2(10)	No	-	1
SCHEDULES_CODE	VARCHAR2(5)	No	-	-
CONTAINERS_AVAILABLE	VARCHAR2(10)	Yes	-	-
FLAG	NUMBER	Yes	-	-

**Table-3**

- Table 3 represents all the attributes present in the Container\_Schedules.

## 2.1.4 CONTAINER\_TYPES

```
CREATE TABLE CONTAINER_TYPES (
    CONTAINER_TYPE CHAR(3) PRIMARY KEY,
    CONTAINER_NAME VARCHAR(50) NOT NULL,
    EXT_LENGTH VARCHAR(5) NOT NULL,
    EXT_WIDTH VARCHAR(5) NOT NULL,
    EXT_HEIGHT VARCHAR(5) NOT NULL,
    INT_LENGTH VARCHAR(5) NOT NULL,
    INT_WIDTH VARCHAR(5) NOT NULL,
    INT_HEIGHT VARCHAR(5) NOT NULL,
    MGW_WEIGHT VARCHAR(5) NOT NULL,
    TARE_WEIGHT VARCHAR(4) NOT NULL,
    NET_WEIGHT VARCHAR(5) NOT NULL
);
```

**SQL Query-4**

- SQL Query 4 represents the SQL table definitions for Container\_Types.

The screenshot shows the Oracle SQL Developer Object Browser. On the left, under the 'Tables' section, 'CONTAINER\_TYPES' is selected. The main panel displays the 'CONTAINER\_TYPES' table structure with the following columns:

Column Name	Data Type	Nullable	Default	Primary Key
CONTAINER_TYPE	VARCHAR2(10)	No	-	1
CONTAINER_NAME	VARCHAR2(50)	No	-	-
EXT_LENGTH	VARCHAR2(10)	No	-	-
EXT_WIDTH	VARCHAR2(10)	No	-	-
EXT_HEIGHT	VARCHAR2(10)	No	-	-
INT_LENGTH	VARCHAR2(10)	No	-	-
INT_WIDTH	VARCHAR2(10)	No	-	-
INT_HEIGHT	VARCHAR2(10)	No	-	-
MGW_WEIGHT	VARCHAR2(10)	No	-	-
TARE_WEIGHT	VARCHAR2(10)	No	-	-
NET_WEIGHT	VARCHAR2(10)	No	-	-

**Table-4**

- Table 4 represents all the attributes present in the Container\_Types.

## 2.1.5 CREW

```
CREATE TABLE CREW (
    CREW_ID VARCHAR(10) PRIMARY KEY,
    FNAME VARCHAR(20) NOT NULL,
    LNAME VARCHAR(20) NOT NULL,
    DEPT VARCHAR(40) NOT NULL,
    SUB_DEPT VARCHAR(40) NOT NULL,
    ROLE_DEPT VARCHAR(40) NOT NULL,
    IMO_NUMBER NUMBER NULL,
    FOREIGN KEY (IMO_NUMBER) REFERENCES VESSELS(IMO_NUMBER)
);
```

SQL Query-5

- SQL Query 5 represents the SQL table definitions for Crew.

The screenshot shows the Oracle Database Object Browser interface. On the left, there is a tree view of tables: BOOKINGS, CONTAINER\_INFO, CONTAINER\_SCHEDULES, CONTAINER\_TYPES, CREW (which is selected and highlighted in green), CUSTOMER, PORTS, PORT\_DETAILS, ROUTES, SCHEDULES\_LIST, VESSELS, VESSEL\_FEATURES, and VESSEL\_PORT. On the right, the 'CREW' table is displayed in a grid format with the following columns:

Column Name	Data Type	Nullable	Default	Primary Key
CREW_ID	VARCHAR2(10)	No	-	1
FNAME	VARCHAR2(20)	No	-	-
LNAME	VARCHAR2(20)	No	-	-
DEPT	VARCHAR2(40)	No	-	-
SUB_DEPT	VARCHAR2(40)	No	-	-
ROLE_DEPT	VARCHAR2(40)	No	-	-
IMO_NUMBER	NUMBER	Yes	-	-

Table-5

- Table 5 represents all the attributes present in the Crew.

## 2.1.6 CUSTOMER

```
CREATE TABLE CUSTOMER(
    CUSTOMER_ID VARCHAR(10) PRIMARY KEY,
    FNAME VARCHAR(20) NOT NULL,
    LNAME VARCHAR(20) NOT NULL,
    TELEPHONE VARCHAR(20) NOT NULL,
    ALTERNATIVE_PHONE VARCHAR(25) NOT NULL,
    MAIL_ID VARCHAR(50) NOT NULL
);
```

SQL Query-6

- SQL Query 6 represents the SQL table definitions for Customer.

Column Name	Data Type	Nullable	Default	Primary Key
CUSTOMER_ID	VARCHAR2(10)	No	-	1
FNAME	VARCHAR2(20)	No	-	-
LNAME	VARCHAR2(20)	No	-	-
TELEPHONE	VARCHAR2(20)	No	-	-
ALTERNATIVE_PHONE	VARCHAR2(25)	No	-	-
MAIL_ID	VARCHAR2(50)	No	-	-

**Table-6**

- Table 6 represents all the attributes present in the Customer.

### 2.1.7 PORTS

```
CREATE TABLE PORTS (
    PORT_ID VARCHAR(10) PRIMARY KEY,
    PORT_NAME VARCHAR(20) NOT NULL,
    ROUTE_ID VARCHAR(5) NOT NULL,
    FOREIGN KEY (PORT_NAME) REFERENCES PORT_DETAILS(PORT_NAME),
    FOREIGN KEY (ROUTE_ID) REFERENCES ROUTES(ROUTE_ID)
);
```

**SQL Query-7**

- SQL Query 7 represents the SQL table definitions for Ports.

Column Name	Data Type	Nullable	Default	Primary Key
PORT_ID	VARCHAR2(10)	No	-	1
PORT_NAME	VARCHAR2(20)	No	-	-
ROUTE_ID	VARCHAR2(5)	No	-	-

**Table-7**

- Table 7 represents all the attributes present in the Ports.

## 2.1.8 PORT DETAILS

```
CREATE TABLE PORT_DETAILS (
    PORT_NAME VARCHAR(20) PRIMARY KEY,
    PORT_ADDRESS_1 VARCHAR(50) NOT NULL,
    PORT_ADDRESS_2 VARCHAR(50) NOT NULL,
    PHONE VARCHAR(20) NOT NULL,
    EMAIL VARCHAR(50) NOT NULL
);
```

**SQL Query-8**

- SQL Query 8 represents the SQL table definitions for Ports Details.

The screenshot shows the Oracle Database Object Browser interface. On the left, there is a tree view of tables: BOOKINGS, CONTAINER\_INFO, CONTAINER\_SCHEDULES, CONTAINER\_TYPES, CREW, CUSTOMER, PORTS, PORT\_DETAILS (which is selected and highlighted in green), ROUTES, SCHEDULES\_LIST, VESSELS, VESSEL\_FEATURES, and VESSEL\_PORT. On the right, the details for the selected 'PORT\_DETAILS' table are displayed in a grid format. The columns are: Column Name, Data Type, Nullable, Default, and Primary Key. The data is as follows:

Column Name	Data Type	Nullable	Default	Primary Key
PORT_NAME	VARCHAR2(20)	No	-	1
PORT_ADDRESS_1	VARCHAR2(50)	No	-	-
PORT_ADDRESS_2	VARCHAR2(50)	No	-	-
PHONE	VARCHAR2(20)	No	-	-
EMAIL	VARCHAR2(50)	No	-	-

**Table-8**

- Table 8 represents all the attributes present in the Port Details.

## 2.1.9 ROUTES

```
CREATE TABLE ROUTES (
    ROUTE_ID VARCHAR(5) PRIMARY KEY,
    ROUTE_NAME VARCHAR(20) NOT NULL
);
```

**SQL Query-9**

- SQL Query 9 represents the SQL table definitions for Routes.

The screenshot shows the Oracle Database Object Browser interface. The left sidebar lists various database objects: BOOKINGS, CONTAINER\_INFO, CONTAINER\_SCHEDULES, CONTAINER\_TYPES, CREW, CUSTOMER, PORTS, PORT\_DETAILS, ROUTES (which is selected and highlighted in green), SCHEDULES\_LIST, VESSELS, VESSEL\_FEATURES, and VESSEL\_PORT. The main panel displays the structure of the 'ROUTES' table. The table has two columns: 'ROUTE\_ID' (VARCHAR2(5)) and 'ROUTE\_NAME' (VARCHAR2(20)). Both columns are nullable and have no default values assigned. The primary key is indicated by a '1' in the 'Primary Key' column.

ROUTES				
Column Name	Data Type	Nullable	Default	Primary Key
ROUTE_ID	VARCHAR2(5)	No	-	1
ROUTE_NAME	VARCHAR2(20)	No	-	-

**Table-9**

- Table 9 represents all the attributes present in the Routes.

### 2.1.10 SCHEDULES LIST

```
CREATE TABLE SCHEDULES_LIST (
    SCHEDULES_CODE VARCHAR(5) PRIMARY KEY,
    AVAILABLE_FROM DATE NOT NULL,
    AVAILABLE_TO DATE NOT NULL
);
```

**SQL Query-10**

- SQL Query 10 represents the SQL table definitions for Schedules List.

The screenshot shows the Oracle Database Object Browser interface. The left sidebar lists various database objects: BOOKINGS, CONTAINER\_INFO, CONTAINER\_SCHEDULES, CONTAINER\_TYPES, CREW, CUSTOMER, PORTS, PORT\_DETAILS, ROUTES, SCHEDULES\_LIST (selected and highlighted in green), VESSELS, VESSEL\_FEATURES, and VESSEL\_PORT. The main panel displays the structure of the 'SCHEDULES\_LIST' table. The table has three columns: 'SCHEDULES\_CODE' (VARCHAR2(5)), 'AVAILABLE\_FROM' (VARCHAR2(12)), and 'AVAILABLE\_TO' (VARCHAR2(12)). All columns are nullable and have no default values assigned. The primary key is indicated by a '1' in the 'Primary Key' column.

SCHEDULES_LIST				
Column Name	Data Type	Nullable	Default	Primary Key
SCHEDULES_CODE	VARCHAR2(5)	No	-	1
AVAILABLE_FROM	VARCHAR2(12)	No	-	-
AVAILABLE_TO	VARCHAR2(12)	No	-	-

**Table-10**

- Table 10 represents all the attributes present in the Schedules List.

### 2.1.11 VESSELS

```
CREATE TABLE VESSELS (
    IMO_NUMBER NUMBER PRIMARY KEY,
    VESSEL_NAME VARCHAR(20) NOT NULL,
    VESSEL_TYPE CHAR(1) NOT NULL,
    GROSS_TONNAGE VARCHAR(10) NOT NULL,
    DATE_BUILT NUMBER NOT NULL,
    FOREIGN KEY (VESSEL_TYPE) REFERENCES VESSEL_FEATURES(VESSEL_TYPE)
);
```

**SQL Query-11**

- SQL Query 11 represents the SQL table definitions for Vessels.

The screenshot shows the Oracle Database Object Browser. On the left, there is a tree view of tables: BOOKINGS, CONTAINER\_INFO, CONTAINER\_SCHEDULES, CONTAINER\_TYPES, CREW, CUSTOMER, PORTS, PORT\_DETAILS, ROUTES, SCHEDULES\_LIST, VESSELS, VESSEL\_FEATURES, and VESSEL\_PORT. The 'VESSELS' table is selected and highlighted with a green background. On the right, the 'VESSELS' table definition is displayed in a grid format. The columns are: Column Name, Data Type, Nullable, Default, and Primary Key. The data is as follows:

Column Name	Data Type	Nullable	Default	Primary Key
IMO_NUMBER	NUMBER	No	-	1
VESSEL_NAME	VARCHAR2(20)	No	-	-
VESSEL_TYPE	CHAR(1)	No	-	-
GROSS_TONNAGE	VARCHAR2(10)	No	-	-
DATE_BUILT	DATE	No	-	-

**Table-11**

- Table 11 represents all the attributes present in the Vessels.

### 2.1.12 VESSEL FEATURES

```
CREATE TABLE VESSEL_FEATURES (
    VESSEL_TYPE CHAR(1) PRIMARY KEY,
    VESSEL_CAPACITY VARCHAR(10) NOT NULL,
    SPEED NUMBER NOT NULL,
    DEADWEIGHT_TONNAGE VARCHAR(10) NOT NULL,
    LENGTH_OVERALL VARCHAR(10) NOT NULL,
    BREADTH_OVERALL VARCHAR(10) NOT NULL,
    QUANTITY_OWNED NUMBER NOT NULL
);
```

**SQL Query-12**

- SQL Query 12 represents the SQL table definitions for Vessel Features.

The screenshot shows the Oracle SQL Developer Object Browser. On the left, there is a tree view of tables: BOOKINGS, CONTAINER\_INFO, CONTAINER\_SCHEDULES, CONTAINER\_TYPES, CREW, CUSTOMER, PORTS, PORT\_DETAILS, ROUTES, SCHEDULES\_LIST, VESSELS, VESSEL\_FEATURES (which is selected and highlighted in green), and VESSEL\_PORT. On the right, the details for the VESSEL\_FEATURES table are displayed. The table has 7 columns: VESSEL\_TYPE, VESSEL\_CAPACITY, SPEED, DEADWEIGHT\_TONNAGE, LENGTH\_OVERALL, BREADTH\_OVERALL, and QUANTITY\_OWNED. The primary key is VESSEL\_TYPE.

VESSEL_FEATURES				
Column Name	Data Type	Nullable	Default	Primary Key
VESSEL_TYPE	CHAR(1)	No	-	1
VESSEL_CAPACITY	VARCHAR2(10)	No	-	
SPEED	NUMBER	No	-	
DEADWEIGHT_TONNAGE	VARCHAR2(10)	No	-	
LENGTH_OVERALL	VARCHAR2(10)	No	-	
BREADTH_OVERALL	VARCHAR2(10)	No	-	
QUANTITY_OWNED	NUMBER	No	-	

Table-12

- Table 12 represents all the attributes present in the Vessel Features.

### 2.1.13 VESSEL PORT

```
CREATE TABLE VESSEL_PORT (
    IMO_NUMBER NUMBER NOT NULL,
    PORT_ID VARCHAR(10) NOT NULL,
    VESSEL_ARRIVAL VARCHAR(15) NOT NULL,
    VESSEL_ARRIVAL_TIME VARCHAR(20) NOT NULL,
    VESSEL_DEPARTURE VARCHAR(15) NOT NULL,
    VESSEL_DEPARTURE_TIME VARCHAR(20) NOT NULL,
    PRIMARY KEY(IMO_NUMBER,PORT_ID),
    FOREIGN KEY (IMO_NUMBER) REFERENCES VESSELS(IMO_NUMBER),
    FOREIGN KEY (PORT_ID) REFERENCES PORTS(PORT_ID)
);
```

SQL Query-13

- SQL Query 13 represents the SQL table definitions for Vessel Port.

The screenshot shows the Oracle SQL Developer Object Browser. On the left, there is a tree view of tables: BOOKINGS, CONTAINER\_INFO, CONTAINER\_SCHEDULES, CONTAINER\_TYPES, CREW, CUSTOMER, PORTS, PORT\_DETAILS, ROUTES, SCHEDULES\_LIST, VESSELS, VESSEL\_FEATURES, and VESSEL\_PORT (which is selected and highlighted in green). On the right, the details for the VESSEL\_PORT table are displayed. The table has 7 columns: IMO\_NUMBER, PORT\_ID, VESSEL\_ARRIVAL, VESSEL\_ARRIVAL\_TIME, VESSEL\_DEPARTURE, and VESSEL\_DEPARTURE\_TIME. The primary key is IMO\_NUMBER and PORT\_ID.

VESSEL_PORT				
Column Name	Data Type	Nullable	Default	Primary Key
IMO_NUMBER	NUMBER	No	-	1
PORT_ID	VARCHAR2(10)	No	-	2
VESSEL_ARRIVAL	VARCHAR2(15)	No	-	
VESSEL_ARRIVAL_TIME	VARCHAR2(20)	No	-	
VESSEL_DEPARTURE	VARCHAR2(15)	No	-	
VESSEL_DEPARTURE_TIME	VARCHAR2(20)	No	-	

Table-13

- Table 13 represents all the attributes present in the Vessel Port.

## CHAPTER 3

### DISCUSSION ON THE DATASET

#### 3.1 DATABASE SHOWING THE MULTIPLICITY OF THE RELATIONSHIPS

CUSTOMER							
Table	Data	Indexes	Model	Constraints	Grants	Statistics	
EDIT		CUSTOMER_ID	FNAME	LNAME	TELEPHONE	ALTERNATIVE_PHONE	MAIL_ID
		CID0001	Karlik	Filinkov	(704) 8869180	+86 (626) 199-8777	kfilinkov@sciencedirect.com
		CID0002	Heinrick	Ovenden	(793) 6950320	+86 (746) 726-0986	hovenden1@accuweather.com
		CID0003	Kirstyn	Harkins	(212) 7807625	+352 (701) 719-7318	kharkins2@etsy.com
		CID0004	Eddy	Vidineev	(566) 8354884	+351 (965) 356-5601	evidineev3@w3.org
		CID0005	Zaneta	Alsobrook	(651) 6494253	+86 (605) 594-2254	zalsobrook4@facebook.com
		CID0006	Odie	Rollings	(668) 5301305	+62 (745) 713-2873	orollings5@google.com
		CID0007	Jecho	Hurnell	(902) 3640215	+7 (759) 276-2404	jhurnell6@4shared.com
		CID0008	Binni	Baskeyfield	(658) 3755289	+1 (213) 550-0716	basakiyfield7@usgs.gov
		CID0009	Abelard	Soligne	(641) 6314186	+371 (137) 526-1755	asoligne8@yahoo.co.jp
		CID0010	Xavier	Jacketts	(212) 1729148	+62 (698) 933-3989	xjacketts9@storify.com
		CID0011	Orly	Klishin	(598) 7008047	+60 (646) 255-2709	oklishina@clickbank.net
		CID0012	Michelle	Derges	(304) 8454863	+86 (725) 538-9773	mdergesb@ylasite.com
		CID0013	Mannie	Wicklin	(195) 6239121	+34 (702) 861-4586	mwicklin@lycos.com

Table-1

- Table-1 represents the Customer details in the Database.

BOOKINGS													
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries	
Query	Count Rows	Insert Row	Load Data										
EDIT	COMPANY_BOOKING_NO	CUSTOMER_ID	ITEMS	DESTINATION_PORT	BOOKING_DATE	BOOKING_TIME	PRICE_TAX	SCHEDULES_CODE	IMO_NUMBER	CONTAINER_ID	ROUTE_ID	ITEM_SENT_DATE	
	ECEBN100	CID0001	1000 Chairs	Kabul	06/01/2022	4:37	\$7695.29	SCH2	7084582	CONUM9511	RID1	06/01/2022	
	ECEBN101	CID0001	500 Aiml Books	Oran	07/01/2022	05:37	\$7257.58	SCH2	7084582	CONUM9512	RID1	07/01/2022	
	ECEBN102	CID0002	Chemicals	Oran	01/01/2022	21:45	\$3679.58	SCH1	7084582	CONUM9513	RID1	01/01/2022	
	ECEBN103	CID0003	Fuels	Soyo	09/01/2022	14:50	\$5069.91	SCH3	7084582	CONUM9514	RID1	09/01/2022	
	ECEBN104	CID0006	280 units Dining Table	Soyo	11/01/2022	22:26	\$6390.67	SCH5	7084582	CONUM9515	RID1	11/01/2022	
	ECEBN105	CID0006	600 Bottles Soda (Each 4L)	Madras	13/01/2022	04:59	\$4018.65	SCH4	7185229	CONUM9516	RID2	13/01/2022	
	ECEBN106	CID0001	1000 Kg Raw Meat	Jambi	07/01/2022	15:03	\$5427.87	SCH2	7185229	CONUM9517	RID2	07/01/2022	
	ECEBN107	CID0004	1500 Ton Cement	Madras	14/01/2022	3:36	\$6521.27	SCH4	7185229	CONUM9518	RID2	14/01/2022	

Table-2

- Table-2 displays the details of bookings made by a specific Customer (for instance CID0001), includes multiple bookings with unique booking numbers. These bookings are further described in the database above.

PORT_DETAILS					
Table	Data	Indexes	Model	Constraints	Grants
<a href="#">Statistics</a> <a href="#">UI Defaults</a> <a href="#">Triggers</a> <a href="#">Dependencies</a> <a href="#">SQL</a> <a href="#">REST</a> <a href="#">Sample Queries</a>					
<a href="#">Query</a>	<a href="#">Count Rows</a>	<a href="#">Insert Row</a>	<a href="#">Load Data</a>		
EDIT	PORT_NAME	PORT_ADDRESS_1	PORT_ADDRESS_2	PHONE	EMAIL
<a href="#">Edit</a>	Kabul	99582 Hollow Ridge Court	PO Box 14548	(718) 9655624	edletsche0@samsung.com
<a href="#">Edit</a>	Jambi	69790 Dryden Place	PO Box 91407	(613) 7546264	nhaw1@yahoo.co.jp
<a href="#">Edit</a>	Pune	67 Dawn Center	Apt 1653	(199) 7524768	nredd2@columbia.edu
<a href="#">Edit</a>	Madras	579 Rusk Point	Suite 51	(788) 3429024	oelsmer4@taobao.com
<a href="#">Edit</a>	Cork	31 Waubesa Alley	Apt 318	(407) 4304778	splayford7@thetimes.co.uk
<a href="#">Edit</a>	Zhuhai	35308 Iowa Plaza	Suite 81	(547) 5030038	bworstell9@nhs.uk
<a href="#">Edit</a>	Alexandria	783 Daystar Court	Suite 96	(997) 6533585	edeemana@intel.com
<a href="#">Edit</a>	Bristol	956 American Ash Court	14th Floor	(765) 6081655	rselewayb@weather.com
<a href="#">Edit</a>	Kotka	56579 Lotheville Point	Room 622	(367) 3307743	cdigmarc@joomla.org

**Table-3**

- Table-3 represents the Port Details in the Database.

ROUTES		
Table	Data	Indexes
<a href="#">Model</a> <a href="#">Constraints</a> <a href="#">Grants</a> <a href="#">Statistics</a> <a href="#">UI Defaults</a> <a href="#">Triggers</a> <a href="#">Dependencies</a> <a href="#">SQL</a> <a href="#">REST</a> <a href="#">Sample Queries</a>		
<a href="#">Query</a>	<a href="#">Count Rows</a>	<a href="#">Insert Row</a>
	<a href="#">Load Data</a>	
EDIT	ROUTE_ID	ROUTE_NAME
<a href="#">Edit</a>	RID1	ROUTE AFG-ANG
<a href="#">Edit</a>	RID2	ROUTE IND-IRE
<a href="#">Edit</a>	RID3	ROUTE CHI-EGY
<a href="#">Edit</a>	RID4	ROUTE ENG-FRA
<a href="#">Edit</a>	RID5	ROUTE GAB-GER
<a href="#">Edit</a>	RID6	ROUTE IRA-ITA
<a href="#">Edit</a>	RID7	ROUTE BAN-BRZ
<a href="#">Edit</a>	RID8	ROUTE CAN-CHL
<a href="#">Edit</a>	RID9	ROUTE PER-RUS

**Table-4**

- Table 4 represents Routes defined in Database.

VESSEL_PORT						
Table	Data	Indexes	Model	Constraints	Grants	Statistics
<a href="#">UI Defaults</a> <a href="#">Triggers</a> <a href="#">Dependencies</a> <a href="#">SQL</a> <a href="#">REST</a> <a href="#">Sample Queries</a>						
<a href="#">Query</a>	<a href="#">Count Rows</a>	<a href="#">Insert Row</a>	<a href="#">Load Data</a>			
EDIT	IMO_NUMBER	PORT_ID	VESSEL_ARRIVAL	VESSEL_ARRIVAL_TIME	VESSEL_DEPARTURE	VESSEL_DEPARTURE_TIME
<a href="#">Edit</a>	7084582	PID101	09/01/2022	17:57	10/01/2022	22:34
<a href="#">Edit</a>	7084582	PID102	08/01/2022	9:46	09/01/2022	08:22
<a href="#">Edit</a>	7084582	PID103	12/01/2022	4:29	13/01/2022	02:47
<a href="#">Edit</a>	7084582	PID104	12/01/2022	21:58	13/01/2022	03:45
<a href="#">Edit</a>	7185229	PID201	18/01/2022	0:01	19/01/2022	05:19
<a href="#">Edit</a>	7185229	PID202	14/01/2022	1:32	15/01/2022	15:42
<a href="#">Edit</a>	7185229	PID203	16/01/2022	18:35	17/01/2022	21:58
<a href="#">Edit</a>	7185229	PID204	25/2/2022	22:42	25/2/2022	11:10
<a href="#">Edit</a>	7228001	PID301	24/2/2022	16:29	25/2/2022	06:00
<a href="#">Edit</a>	7228001	PID302	21/2/2022	23:33	22/2/2022	23:17
<a href="#">Edit</a>	7305003	PID401	20/2/2022	1:16	21/2/2022	11:48
<a href="#">Edit</a>	7305003	PID402	05/03/2022	6:41	06/03/2022	15:37
<a href="#">Edit</a>	7305003	PID404	16/2/2022	9:58	17/2/2022	09:00

**Table-5**

- Table-5 represents a Vessel (IMO Number) having stops at multiple Ports.

PORTS			
Table	Data	Indexes	Model
Constraints Grants Statistics UI Defaults Triggers Dependencies SQL REST Sample Queries			
	Query Count Rows Insert Row Load Data		
EDIT	PORT_ID	PORT_NAME	ROUTE_ID
	PID101	Kabul	RID1
	PID102	Jambi	RID1
	PID103	Pune	RID1
	PID104	Madras	RID1
	PID201	Madras	RID2
	PID202	Jambi	RID2
	PID203	Cork	RID2
	PID204	Pune	RID2
	PID301	Zhuhai	RID3
	PID302	Alexandria	RID3
	PID401	Bristol	RID4
	PID402	Kotka	RID4
	PID403	Pune	RID4

**Table-6**

- Table 6 represents, the different Vessels arriving from different Routes can have the same Port for an item delivery i.e one Port is having multiple Vessels from multiple Routes.
- For instance, Table-6 shows that a vessel has a stop in the Port named Madras from Route 1, and also another vessel from Route 2 passes through the Port Madras.

CONTAINER_TYPES												
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries
	Query Count Rows Insert Row Load Data											
EDIT	CONTAINER_TYPE	CONTAINER_NAME	EXT_LENGTH	EXT_WIDTH	EXT_HEIGHT	INT_LENGTH	INT_WIDTH	INT_HEIGHT	MGW_WEIGHT	TARE_WEIGHT	NET_WEIGHT	
	CTA	20' Steel Dry Cargo Container	6.193m	4.813m	3.304m	5.633m	2.599m	3.786m	33438 Kg	3700 Kg	28687 Kg	
	CTB	40' Steel Dry Cargo Container	6.844m	4.48m	3.718m	5.654m	3.042m	3.822m	31056 Kg	2218 Kg	29619 Kg	
	CTC	40' Steel Refrigerated Cargo Container	6.189m	2.659m	2.302m	5.522m	2.793m	4.488m	35215 Kg	3659 Kg	29975 Kg	
	CTD	40' Hi-Cube Steel Dry Cargo Container	6.99m	2.897m	3.123m	5.318m	2.953m	4.398m	34144 Kg	2199 Kg	28473 Kg	
	CTE	45' Hi-Cube Steel Dry Cargo Container	6.77m	2.091m	2.145m	5.725m	2.485m	3.738m	30332 Kg	3175 Kg	29659 Kg	
	CTF	20' Full Height Open Top Container	6.475m	2.052m	2.225m	5.892m	3.171m	3.853m	31560 Kg	2680 Kg	28499 Kg	
	CTG	40' Full Height Open Top Container	6.974m	2.981m	3.439m	5.629m	2.242m	4.386m	31101 Kg	3649 Kg	28781 Kg	
	CTH	40' Hi-Cube Open Top Container	6.085m	2.875m	2.077m	5.47m	3.182m	2.857m	30450 Kg	2390 Kg	28908 Kg	
	CTI	20' Flat Rack Container with Collapsible End	6.096m	3.231m	3.934m	5.555m	3.547m	3.022m	30082 Kg	3957 Kg	28915 Kg	
	CTJ	40' Flat Rack Hi-Cube Container	6.266m	4.993m	3.985m	5.746m	2.342m	2.012m	31461 Kg	3556 Kg	28085 Kg	

**Table-7**

- Table-7 represents different container types defined in the Database.

VESSEL_FEATURES							
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults
Query	Count Rows	Insert Row	Load Data				
EDIT	VESSEL_TYPE	VESSEL_CAPACITY	SPEED	DEADWEIGHT_TONNAGE	LENGTH_OVERALL	BREADTH_OVERALL	QUANTITY_OWNED
✓	A	23992	23	224999	399.98m	61.5m	5
✓	G	20124	23	199089	399.98m	58.8m	5
✓	T	15808	23	152344	368.47m	51.0m	5
✓	F	12118	23	127076	333.96m	48.4m	5
✓	L	9466	28	267025	334.98m	45.31m	5
✓	S	6944	17	199999	299.99m	52.44m	5
✓	E	6332	37	382928	299.99m	61.05m	5
✓	B	2882	25	144222	211.9m	49.43m	5
✓	O	2634	23	187392	195.00m	72.33m	5
✓	C	1778	15	280002	172.07m	68.92m	5

Table-8

- Table-8 represents the Vessel Features defined in the Database.

VESSELS					
Table	Data	Indexes	Model	Constraints	Grants
Query	Count Rows	Insert Row	Load Data		
EDIT	IMO_NUMBER	VESSEL_NAME	VESSEL_TYPE	GROSS_TONNAGE	DATE_BUILT
✓	7084582	Ever Gold	A	190267	01/26/2010
✓	7036819	Ever Silver	A	249424	05/20/2002
✓	7061515	Ever Platinum	A	195511	09/02/2010
✓	7082311	Ever Diamond	A	202147	05/18/1999
✓	7054434	Ever Green	A	319195	09/05/2015
✓	7185229	Ever Rcb	G	339409	07/05/2011
✓	7147770	Ever Army	G	163759	06/13/2007
✓	7107965	Ever Ark	G	331467	05/13/2003
✓	7126961	Ever Sophia	G	155579	07/05/1993
✓	7119004	Ever Louie	G	336291	03/22/2018
✓	7228001	Ever Faith	T	208980	07/11/1998
✓	7225278	Ever Dino	T	265194	03/01/2016
✓	7227839	Ever Mishell	T	160551	05/01/1994

Table-9

- Table-9 represents the details of Vessels defined in the Database.

CONTAINER_INFO						
Table	Data	Indexes	Model	Constraints	Grants	Statistics
Query	Count Rows	Insert Row	Load Data			
EDIT	CONTAINER_ID	CONTAINER_TYPE	IMO_NUMBER	ROUTE_ID	PORT_ID	CHOSEN_CONTAINER
✓	CONUM95174	CTJ	7702579	RID9	PID901	YES
✓	CONUM95172	CTA	7327734	RID5	PID501	YES
✓	CONUM95111	CTA	7084582	RID1	PID101	YES
✓	CONUM95112	CTB	7084582	RID1	PID102	YES
✓	CONUM95113	CTC	7084582	RID1	PID103	YES
✓	CONUM95114	CTD	7084582	RID1	PID103	YES
✓	CONUM95115	CTE	7084582	RID1	PID104	YES
✓	CONUM95116	CTF	7185229	RID2	PID201	YES
✓	CONUM95117	CTG	7185229	RID2	PID202	YES
✓	CONUM95118	CTH	7185229	RID2	PID203	YES
✓	CONUM95119	CTI	7185229	RID2	PID204	YES
✓	CONUM95120	CTJ	7228001	RID3	PID301	YES
✓	CONUM95121	CTA	7228001	RID3	PID302	YES

Table-10

	CONUM95142	CTB	7889692	RID9	PID901	YES
	CONUM95143	CTC	7889692	RID9	PID902	YES
	CONUM95144	CTD	7889692	RID9	PID903	YES
	CONUM95145	CTE	7889692	RID9	PID904	YES
	CONUM95146	CTF	7920790	RID1	PID101	YES
	CONUM95147	CTG	7920790	RID1	PID103	YES
	CONUM95148	CTH	7036819	RID2	PID201	YES
	CONUM95149	CTI	7036819	RID2	PID202	YES
	CONUM95150	CTJ	7036819	RID2	PID202	YES

Table-11

- Tables 10 and 11 demonstrate that a single vessel may carry multiple containers and visit multiple ports. Additionally, it is evident that different vessels carrying distinct containers can pass through the same ports for unloading.
- Further analysis of Tables 10 and 11 indicates that a single route, such as RID1, may have multiple vessels that traverse the same course. For example, RID1 includes 7084582, 7920790 and possibly more.

CREW								
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	
Edit		CREW_ID	FNAME	LNAME	DEPT	SUB_DEPT	ROLE_DEPT	IMO_NUMBER
		CRID111	Bryn	Oldey	Captain	Captain	Captain	7084582
		CRID112	Raddie	Trinke	Deck Dept	Officers of the Watch	Chief Mate	7084582
		CRID113	Antonino	MacMillan	Engineering Dept	Officers	Chief Engineer	7084582
		CRID114	Christa	Banstead	Steward's Dept	Unlicensed Crew	Chief Cook	7084582
		CRID115	Alameda	Gerault	Deck Dept	Ratings	Able Bodies Seaman	7084582
		CRID116	Frederigo	Matcham	Captain	Captain	Captain	7185229
		CRID117	Jackie	Ciraldo	Engineering Dept	Ratings	Engine Room Rating	7185229
		CRID118	Elvira	O'Nolan	Steward's Dept	Unlicensed Crew	Second Cook	7185229
		CRID119	Concordia	Barthrup	Engineering Dept	Officers	Second Engineer	7185229
		CRID120	Delmar	Shall	Deck Dept	Officers of the Watch	Second Mate	7185229
		CRID121	Florrie	Gateman	Captain	Captain	Captain	7228001
		CRID122	Lucky	McGuffog	Steward's Dept	Unlicensed Crew	Chief Cook	7228001
		CRID123	Angelita	Bowdery	Deck Dept	Ratings	Ordinary Seaman	7228001

Table-12

- Table -12 represents a Vessel (IMO Number) having multiple Crew members from different Department with different Crew IDs.

SCHEDULES_LIST				
Table	Data	Indexes	Model	
Edit		SCHEDULES_CODE	AVAILABLE_FROM	AVAILABLE_TO
		SCH1	01/01/2022	05/01/2022
		SCH2	06/01/2022	08/01/2022
		SCH3	09/01/2022	12/01/2022
		SCH4	13/01/2022	15/01/2022
		SCH5	19/01/2022	30/01/2022
		SCH6	10/02/2022	14/02/2022
		SCH7	15/02/2022	17/02/2022
		SCH8	18/02/2022	20/02/2022
		SCH9	21/02/2022	27/02/2022

Table-13

- Table-13 represents the available dates for booking the containers.

CONTAINER_SCHEDULES				
Table	Data	Indexes	Model	Constraints
	Query	Count Rows	Insert Row	Load Data
EDIT	CONTAINER_SCHEDULE	SCHEDULES_CODE	CONTAINERS_AVAILABLE	FLAG
	COSCH001	SCH1	CONUM95113	1
	COSCH002	SCH1	CONUM95125	1
	COSCH003	SCH1	CONUM95135	1
	COSCH004	SCH1	CONUM95145	1
	COSCH005	SCH1	CONUM95153	1
	COSCH006	SCH2	CONUM95111	1
	COSCH007	SCH2	CONUM95112	1
	COSCH008	SCH2	CONUM95117	1
	COSCH009	SCH2	CONUM95136	1
	COSCH010	SCH2	CONUM95146	1
	COSCH011	SCH2	CONUM95154	1
	COSCH012	SCH2	CONUM95162	1
	COSCH013	SCH2	CONUM95163	1

**Table-14**

- Table-14 represents the bookings of different Containers in a Schedule.

## CHAPTER 4

### SQL QUERIES

#### 4.1 SQL QUERY-1 AND RESULTS

##### 4.1.1 Tracking the Customer Order through Booking Number

```

SELECT
CUSTOMER.FNAME,CUSTOMER.LNAME,CUSTOMER.TELEPHONE,BOOKINGS.ITEMS,
BOOKINGS.DESTINATION_PORT,BOOKINGS.CONTAINER_ID,PORTS.PORT_NAME,
BOOKINGS.BOOKING_DATE,BOOKINGS.BOOKING_TIME,BOOKINGS.PRICE_TAX,
PORT_DETAILS.PORT_ADDRESS_1,PORT_DETAILS.PHONE,
ROUTES.ROUTE_NAME,BOOKINGS.IMO_NUMBER,VESSEL_PORT.VESSEL_ARRIVAL,
VESSEL_PORT.VESSEL_DEPARTURE
FROM VESSEL_PORT
JOIN BOOKINGS ON VESSEL_PORT.IMO_NUMBER = BOOKINGS.IMO_NUMBER
JOIN CUSTOMER ON BOOKINGS.CUSTOMER_ID = CUSTOMER.CUSTOMER_ID
JOIN PORTS ON VESSEL_PORT.PORT_ID = PORTS.PORT_ID
JOIN PORT_DETAILS ON PORTS.PORT_NAME = PORT_DETAILS.PORT_NAME
JOIN ROUTES ON PORTS.ROUTE_ID = ROUTES.ROUTE_ID
WHERE VESSEL_PORT.PORT_ID IN (
    SELECT PORT_ID
    FROM PORTS
    WHERE PORT_ID IN (
        SELECT PORT_ID
        FROM CONTAINER_INFO
        WHERE CONTAINER_ID IN (
            SELECT CONTAINER_ID
            FROM BOOKINGS
            WHERE COMPANY_BOOKING_NO = 'ECEBN102'
        )
    )
)
AND VESSEL_PORT.IMO_NUMBER IN (
    SELECT IMO_NUMBER
    FROM CONTAINER_INFO
    WHERE CONTAINER_ID IN (
        SELECT CONTAINER_ID
        FROM BOOKINGS
        WHERE COMPANY_BOOKING_NO = 'ECEBN102'
    )
)
AND BOOKINGS.COMPANY_BOOKING_NO = 'ECEBN102'

```

From the above SQL query, a customer can track the order details of an item through a Booking Number.

#### 4.1.2 Result Obtained

Results														
FNAME	LNAME	TELEPHONE	ITEMS	DESTINATION_PORT	CONTAINER_ID	PORT_NAME	BOOKING_DATE	BOOKING_TIME	PRICE_TAX	PORT_ADDRESS_1	PHONE	ROUTE_NAME		
Heinrick	Ovenden	(793) 6950320	Chemicals	Oran	CONUM95113	Pune	01/01/2022	21:45	\$3679.58	67 Dawn Center	(199) 7524768	ROUTE AFG-ANG		
												IMO_NUMBER	VESSEL_ARRIVAL	VESSEL_DEPARTURE
												7084582	12/01/2022	13/01/2022

#### Result-1

- + The output shows the tracking of a customer's shipment through a unique booking ID. Upon entering the booking number, the customer can access various details including their personal information such as first name, last name, and telephone number, as well as information regarding the shipment itself, including the items being shipped, destination port, container number, current port name of the vessel, booking date and time, amount paid, and the current address and phone number of the vessel's port.
- + Additionally, the tracking system also provides the name of the route the vessel is travelling, vessel number, and the expected arrival and departure dates of the vessel at the customer's destination port.

## 4.2 SQL QUERY-2 AND RESULTS

#### 4.2.1 To obtain the count of total bookings in each port

```

SELECT PORTS.PORT_NAME, COUNT(*) AS TOTAL_BOOKINGS
FROM PORT_DETAILS
JOIN PORTS ON PORTS.PORT_NAME = PORT_DETAILS.PORT_NAME
JOIN CONTAINER_INFO ON CONTAINER_INFO.PORT_ID = PORTS.PORT_ID
JOIN BOOKINGS ON BOOKINGS.CONTAINER_ID = CONTAINER_INFO.CONTAINER_ID
GROUP BY PORTS.PORT_NAME
HAVING COUNT(CASE WHEN PORT_DETAILS.PORT_NAME IN (
    SELECT PORT_NAME
    FROM PORTS
    WHERE PORT_ID IN (
        SELECT PORT_ID
        FROM CONTAINER_INFO
        WHERE CONTAINER_ID IN (
            SELECT CONTAINER_ID
            FROM BOOKINGS
        )
    )
)
)
THEN 1

```

```

    ELSE NULL
END) > 0;

```

From the above query, the ECE company can get to know the total count of bookings in each port.

#### 4.2.2 Result Obtained

Results	Explain	Describe	Saved SQL	History
PORT_NAME	TOTAL_BOOKINGS			
Pune	4			
Zhuhai	3			
Kabul	2			
Bristol	2			
Argentina	3			
Arica	2			
Madras	3			
Bangalore	3			
Banjul	2			
Turbo	4			
CapeTown	2			
Primorsk	2			
Cork	3			
Naples	2			
Dhaka	2			
Vitoria	2			
Praia	2			
Pisco	2			
Kotka	1			
Bremen	3			
Jambi	4			
Alexandria	4			
Neka	2			
Haifa	1			
Soyo	1			

25 rows returned in 0.12 seconds [Download](#)

**Result -2**

+ The provided output showcases the total number of bookings recorded for each port, thereby providing valuable information to the company, ECE, regarding the volume of container traffic being unloaded at each port.

#### 4.3 SQL QUERY-3 AND RESULTS

##### 4.3.1 To obtain the details and total count of containers in every vessel and also to display the containers present in each vessel.

```

SELECT
VESSELS.VESSEL_NAME,
VESSELS.IMO_NUMBER,
COUNT(CONTAINER_INFO.CONTAINER_ID) AS TOTAL_CONTAINERS,
LISTAGG(CONTAINER_INFO.CONTAINER_ID, ',') WITHIN GROUP (ORDER BY
CONTAINER_INFO.CONTAINER_ID) AS CONTAINERS,
VESSELS.GROSS_TONNAGE,
VESSEL_FEATURES.VESSEL_CAPACITY,
VESSEL_FEATURES.SPEED,

```

```

VESSEL_FEATURES.DEADWEIGHT_TONNAGE,
VESSEL_FEATURES.LENGTH_OVERALL,
VESSEL_FEATURES.BREADTH_OVERALL
FROM VESSELS
JOIN CONTAINER_INFO ON VESSELS.IMO_NUMBER = CONTAINER_INFO.IMO_NUMBER
INNER JOIN VESSEL_FEATURES ON VESSELS.VESSEL_TYPE =
VESSEL_FEATURES.VESSEL_TYPE
GROUP BY
VESSELS.IMO_NUMBER,
VESSELS.VESSEL_NAME,
VESSELS.GROSS_TONNAGE,
VESSEL_FEATURES.VESSEL_CAPACITY,
VESSEL_FEATURES.SPEED,
VESSEL_FEATURES.DEADWEIGHT_TONNAGE,
VESSEL_FEATURES.LENGTH_OVERALL,
VESSEL_FEATURES.BREADTH_OVERALL;

```

From the above query, we can obtain the total count of containers in every vessel and display the details and also list of containers present in each vessel.

#### 4.3.2 Result Obtained

Results	Explain	Describe	Saved SQL	History					
VESSEL_NAME	IMO_NUMBER	TOTAL_CONTAINERS	CONTAINERS	GROSS_TONNAGE	VESSEL_CAPACITY	SPEED	DEADWEIGHT_TONNAGE		
Ever Sos	7510300	3	CONUM95133,CONUM95134,CONUM95135	257631	6944	17	199999		
Ever Silver	7036819	4	CONUM95148,CONUM95149,CONUM95150,CONUM95151	249424	23992	23	224999		
Ever Gold	7084582	5	CONUM95111,CONUM95112,CONUM95113,CONUM95114,CONUM95115	190267	23992	23	224999		
Ever Army	7147770	3	CONUM95152,CONUM95153,CONUM95154	163759	20124	23	199089		
Ever Rcb	7185229	4	CONUM95116,CONUM95117,CONUM95118,CONUM95119	339409	20124	23	199089		
Ever Mishell	7227839	3	CONUM95155,CONUM95156,CONUM95157	160551	13808	23	152344		
Ever Faith	7228001	2	CONUM95120,CONUM95121	208980	13808	23	152344		
Ever Karnataka	7305003	4	CONUM95122,CONUM95123,CONUM95124,CONUM95125	241438	12118	23	127076		
Ever God	7327754	5	CONUM95158,CONUM95159,CONUM95160,CONUM95172,CONUM95173	275062	12118	23	127076		
Ever Contessa	7474318	3	CONUM95161,CONUM95162,CONUM95163	182209	9466	28	267025		
Ever Krishna	7495118	4	CONUM95126,CONUM95127,CONUM95128,CONUM95129	156707	9466	28	267025		
Ever Titanic	7519843	2	CONUM95164,CONUM95165	205199	6944	17	199999		
Ever Vadiraj	7678487	2	CONUM95136,CONUM95137	236379	6332	37	382928		
Ever Senora	7697439	2	CONUM95166,CONUM95167	273977	6332	37	382928		
Ever Verbana	7702579	4	CONUM95168,CONUM95169,CONUM95170,CONUM95174	173599	2882	25	144222		
Ever Lucia	7783049	4	CONUM95138,CONUM95139,CONUM95140,CONUM95141	339778	2882	25	144222		
Ever Dutchman	7889692	4	CONUM95142,CONUM95143,CONUM95144,CONUM95145	225220	2634	23	187392		
Ever Kraken	7893345	1	CONUM95171	288021	2634	23	187392		
Ever Dreamboat	7920790	2	CONUM95146,CONUM95147	267497	1778	15	280002		
Ever Goldfield	7996867	3	CONUM95130,CONUM95131,CONUM95132	299790	1778	15	280002		

Result -3

- + The output obtained provides a comprehensive summary of the number of containers contained within each vessel, as well as a detailed account of the specifications and a comprehensive list of the containers present in each individual vessel.

#### 4.4 SQL QUERY- 4a & 4b AND RESULTS

#### 4.4.1 The objective is to retrieve details on the containers available for the customers across various schedules.

```

SELECT SCHEDULES_LIST.AVAILABLE_FROM, SCHEDULES_LIST.AVAILABLE_TO,
LISTAGG(CONTAINER_SCHEDULES.CONTAINERS_AVAILABLE, ',') WITHIN GROUP
(ORDER BY CONTAINER_SCHEDULES.CONTAINERS_AVAILABLE) AS ALL_CONTAINERS
FROM SCHEDULES_LIST
JOIN CONTAINER_SCHEDULES ON SCHEDULES_LIST.SCHEDULES_CODE =
CONTAINER_SCHEDULES.SCHEDULES_CODE
WHERE SCHEDULES_LIST.SCHEDULES_CODE IN
('SCH1','SCH2','SCH3','SCH4','SCH5','SCH6','SCH7','SCH8','SCH9')
GROUP BY SCHEDULES_LIST.AVAILABLE_FROM, SCHEDULES_LIST.AVAILABLE_TO;

```

The above query provide the containers available for all the schedules available in the database.

#### 4.4.2 Result Obtained

Results Explain Describe Saved SQL History		
AVAILABLE_FROM	AVAILABLE_TO	ALL_CONTAINERS
01/01/2022	05/01/2022	CONUM95113, CONUM95125, CONUM95135, CONUM95145, CONUM95153, CONUM95172, CONUM95173
06/01/2022	08/01/2022	CONUM95111, CONUM95112, CONUM95117, CONUM95136, CONUM95146, CONUM95154, CONUM95162, CONUM95163, CONUM95170, CONUM95171, CONUM95174
09/01/2022	12/01/2022	CONUM95114, CONUM95115, CONUM95119, CONUM95126, CONUM95127, CONUM95128, CONUM95137, CONUM95139, CONUM95147, CONUM95148, CONUM95156, CONUM95157, CONUM95164, CONUM95165
10/02/2022	14/02/2022	CONUM95122, CONUM95129, CONUM95133, CONUM95140, CONUM95143, CONUM95149
13/01/2022	15/01/2022	CONUM95116, CONUM95118, CONUM95130, CONUM95155
15/02/2022	17/02/2022	CONUM95120, CONUM95131, CONUM95141, CONUM95150, CONUM95158, CONUM95159, CONUM95166, CONUM95167
18/02/2022	20/02/2022	CONUM95121, CONUM95132, CONUM95134, CONUM95142, CONUM95144, CONUM95151, CONUM95152, CONUM95160, CONUM95168, CONUM95169
19/01/2022	30/01/2022	CONUM95124, CONUM95138
21/02/2022	27/02/2022	CONUM95123, CONUM95161

#### Result – 4a

- + The resulting output displays the complete list of available containers for the specified schedule

#### 4.4.3 To obtain the number of available containers in a particular schedule, both before and after booking.

```

SELECT DISTINCT
SCHEDULES_LIST.SCHEDULES_CODE,
SCHEDULES_LIST.AVAILABLE_FROM,
SCHEDULES_LIST.AVAILABLE_TO,
LISTAGG(
CASE
WHEN CONTAINER_SCHEDULES.CONTAINERS_AVAILABLE IN ('CONUM95113',
'CONUM95125') THEN CONTAINER_SCHEDULES.CONTAINERS_AVAILABLE
ELSE NULL
END,''
) WITHIN GROUP (ORDER BY CONTAINER_SCHEDULES.CONTAINERS_AVAILABLE) AS
BOOKED_CONTAINERS,
LISTAGG(
CASE

```

```

WHEN CONTAINER_SCHEDULES.CONTAINERS_AVAILABLE NOT IN
('CONUM95113', 'CONUM95125') THEN
CONTAINER_SCHEDULES.CONTAINERS_AVAILABLE
ELSE NULL
END,''
) WITHIN GROUP (ORDER BY CONTAINER_SCHEDULES.CONTAINERS_AVAILABLE) AS
AVAILABLE_CONTAINERS,
COUNT(CASE WHEN CONTAINER_SCHEDULES.CONTAINERS_AVAILABLE IN
('CONUM95113', 'CONUM95125') THEN 1 ELSE NULL END) AS TOTAL_BOOKED,
COUNT(CASE WHEN CONTAINER_SCHEDULES.CONTAINERS_AVAILABLE NOT IN
('CONUM95113', 'CONUM95125') THEN 1 ELSE NULL END) AS TOTAL_AVAILABLE
FROM SCHEDULES_LIST
JOIN CONTAINER_SCHEDULES ON SCHEDULES_LIST.SCHEDULES_CODE =
CONTAINER_SCHEDULES.SCHEDULES_CODE
WHERE SCHEDULES_LIST.SCHEDULES_CODE = 'SCH1'
GROUP BY
SCHEDULES_LIST.SCHEDULES_CODE,
SCHEDULES_LIST.AVAILABLE_FROM,
SCHEDULES_LIST.AVAILABLE_TO;

```

The above SQL query provides the information about the number of available containers in a particular schedule, both before and after booking.

#### 4.4.4 Result Obtained

Results	Explain	Describe	Saved SQL	History			
SCHEDULES_CODE	AVAILABLE_FROM	AVAILABLE_TO	BOOKED_CONTAINERS	AVAILABLE_CONTAINERS	TOTAL_BOOKED	TOTAL_AVAILABLE	
SCH1	01/01/2022	05/01/2022	CONUM95113,CONUM95125	CONUM95135,CONUM95145,CONUM95153,CONUM95172,CONUM95173	2	5	
1 rows returned in 0.00 seconds <a href="#">Download</a>							

Result – 4b

+ The result displays the quantity of containers that have been reserved and the available quantity of containers that can still be booked.

### 4.5 SQL QUERY- 5 AND RESULTS

#### 4.5.1 To obtain the information about a client's booking for a container on a particular day.

```

SELECT
SCHEDULES_LIST.SCHEDULES_CODE,
SCHEDULES_LIST.AVAILABLE_FROM,
SCHEDULES_LIST.AVAILABLE_TO,
LISTAGG(
CASE
WHEN CONTAINER_SCHEDULES.CONTAINERS_AVAILABLE IN ('CONUM95113',
'CONUM95125') THEN CONTAINER_SCHEDULES.CONTAINERS_AVAILABLE
ELSE NULL
END,''

```

```

) WITHIN GROUP (ORDER BY CONTAINER_SCHEDULES.CONTAINERS_AVAILABLE)
AS BOOKED_CONTAINERS,
LISTAGG(
CASE
    WHEN CONTAINER_SCHEDULES.CONTAINERS_AVAILABLE NOT IN
('CONUM95113', 'CONUM95125') THEN
CONTAINER_SCHEDULES.CONTAINERS_AVAILABLE
    ELSE NULL
END,''
) WITHIN GROUP (ORDER BY CONTAINER_SCHEDULES.CONTAINERS_AVAILABLE)
AS AVAILABLE_CONTAINERS,
COUNT(CASE WHEN CONTAINER_SCHEDULES.CONTAINERS_AVAILABLE IN
('CONUM95113', 'CONUM95125') THEN 1 ELSE NULL END) AS TOTAL_BOOKED,
COUNT(CASE WHEN CONTAINER_SCHEDULES.CONTAINERS_AVAILABLE NOT IN
('CONUM95113', 'CONUM95125') THEN 1 ELSE NULL END) AS TOTAL_AVAILABLE,
MAX(CASE WHEN CONTAINER_SCHEDULES.CONTAINERS_AVAILABLE IN
('CONUM95113', 'CONUM95125') THEN BOOKINGS.CUSTOMER_ID ELSE NULL END) AS
CUSTOMER_ID,
BOOKINGS.BOOKING_DATE
FROM SCHEDULES_LIST
JOIN CONTAINER_SCHEDULES ON SCHEDULES_LIST.SCHEDULES_CODE =
CONTAINER_SCHEDULES.SCHEDULES_CODE
INNER JOIN BOOKINGS ON BOOKINGS.CONTAINER_ID =
CONTAINER_SCHEDULES.CONTAINERS_AVAILABLE
WHERE SCHEDULES_LIST.SCHEDULES_CODE = 'SCH1'
GROUP BY
SCHEDULES_LIST.SCHEDULES_CODE,
SCHEDULES_LIST.AVAILABLE_FROM,
SCHEDULES_LIST.AVAILABLE_TO,
BOOKINGS.BOOKING_DATE;

```

The above query retrieves information about customer who have made bookings for containers on a specific schedule. It includes details such as the customer's ID, the containers they have booked, and the booking date.

#### 4.5.2 Result Obtained

Results									
SCHEDULES_CODE	AVAILABLE_FROM	AVAILABLE_TO	BOOKED_CONTAINERS	AVAILABLE_CONTAINERS	TOTAL_BOOKED	TOTAL_AVAILABLE	CUSTOMER_ID	BOOKING_DATE	
SCH1	01/01/2022	05/01/2022	CONUM95113	-	1	0	CID0002	01/01/2022	
SCH1	01/01/2022	05/01/2022	CONUM95125	-	1	0	CID0010	03/01/2022	
SCH1	01/01/2022	05/01/2022	-	CONUM95135,CONUM95145,CONUM95153	0	3	-	05/01/2022	

3 rows returned in 0.06 seconds [Download](#)

**Result – 5**

+ The above query result displays the containers that have been booked by a particular customer along with their respective customer IDs and Booking date. It can be observed that once a customer has booked a container, it becomes unavailable for the other customers to book.

## 4.6 SQL QUERY- 6 AND RESULTS

### 4.6.1 To display the details of crew belonging to a vessel with containers associated with the vessel.

```

SELECT IMO_NUMBER, VESSEL_NAME, DATE_BUILT, VESSEL_CAPACITY, SPEED,
DEADWEIGHT_TONNAGE,
CASE WHEN rn = 1 THEN CREW_ID ELSE NULL END AS CREW_ID,
FNAME, LNAME, DEPT, SUB_DEPT, ROLE_DEPT,
CONTAINERS, CONTAINER_NAME
FROM (
  SELECT VESSELS.IMO_NUMBER, VESSELS.VESSEL_NAME, VESSELS.DATE_BUILT,
  VESSEL_FEATURES.VESSEL_CAPACITY, VESSEL_FEATURES.SPEED,
  VESSEL_FEATURES.DEADWEIGHT_TONNAGE,
  CREW.CREW_ID, CREW.FNAME, CREW.LNAME, CREW.DEPT, CREW.SUB_DEPT,
  CREW.ROLE_DEPT,
  ROW_NUMBER OVER (PARTITION BY CREW.CREW_ID ORDER BY
  CONTAINER_INFO.CONTAINER_ID) AS rn,
  LISTAGG (CONTAINER_INFO.CONTAINER_ID, ',') WITHIN GROUP (ORDER BY
  CONTAINER_INFO.CONTAINER_ID)
  OVER (PARTITION BY CREW.CREW_ID) AS CONTAINERS,
  CONTAINER_TYPES.CONTAINER_NAME
  FROM VESSELS
  JOIN VESSEL_FEATURES ON VESSELS.VESSEL_TYPE =
  VESSEL_FEATURES.VESSEL_TYPE
  JOIN CREW ON VESSELS.IMO_NUMBER = CREW.IMO_NUMBER
  JOIN CONTAINER_INFO ON VESSELS.IMO_NUMBER =
  CONTAINER_INFO.IMO_NUMBER
  JOIN CONTAINER_TYPES ON CONTAINER_INFO.CONTAINER_TYPE =
  CONTAINER_TYPES.CONTAINER_TYPE
  WHERE VESSELS.IMO_NUMBER LIKE '7084582'
)
WHERE rn = 1
ORDER BY CREW_ID, CONTAINER_NAME;

```

The above query displays the details of crew members with their unique IDs assigned to a specific vessel along with the container associated with that vessel.

### 4.6.2 Results Obtained

Results Explain Describe Saved SQL History											
IMO_NUMBER	VESSEL_NAME	DATE_BUILT	VESSEL_CAPACITY	SPEED	DEADWEIGHT_TONNAGE	CREW_ID	FNAME	LNAME	DEPT	SUB_DEPT	ROLE_DEPT
7084582	Ever Gold	01/26/2010	23992	23	224999	CRID111	Bryn	Oldey	Captain	Captain	Captain
7084582	Ever Gold	01/26/2010	23992	23	224999	CRID112	Raddie	Trinke	Deck Dept	Officers of the Watch	Chief Mate
7084582	Ever Gold	01/26/2010	23992	23	224999	CRID113	Antonino	MacMillan	Engineering Dept	Officers	Chief Engineer
7084582	Ever Gold	01/26/2010	23992	23	224999	CRID114	Christa	Banstead	Steward's Dept	Unlicensed Crew	Chief Cook
7084582	Ever Gold	01/26/2010	23992	23	224999	CRID115	Alameda	Gerault	Deck Dept	Ratings	Able Bodies Seaman

Result - 6

CONTAINERS	CONTAINER_NAME
CONUM95111,CONUM95112,CONUM95113,CONUM95114,CONUM95115	20' Steel Dry Cargo Container
CONUM95111,CONUM95112,CONUM95113,CONUM95114,CONUM95115	20' Steel Dry Cargo Container
CONUM95111,CONUM95112,CONUM95113,CONUM95114,CONUM95115	20' Steel Dry Cargo Container
CONUM95111,CONUM95112,CONUM95113,CONUM95114,CONUM95115	20' Steel Dry Cargo Container
CONUM95111,CONUM95112,CONUM95113,CONUM95114,CONUM95115	20' Steel Dry Cargo Container

### Result - 6 contd.

- + The above query result displays the crew members present in a vessel with their unique IDs, along with their departments, sub-departments, and roles. Additionally, the output also displays the containers associated with the vessel and their respective types.

## CHAPTER 5

### CONCLUSION

- After a critical evaluation, the project was able to track customer orders with booking numbers by fetching details from multiple tables.
- Customers can also view the schedules available for the range of dates and book containers through container numbers.
- The query was written in such a way that once a container is booked, it is not displayed to other customers who are willing to book. Customers can also get the total count of containers booked and available containers.
- The project also reviews crew details, who have been assigned to a particular vessel and display containers associated with the vessel. This information is crucial in managing and scheduling the crew for each vessel.
- The project helps the ECE company to know the customer details for those who booked the containers along with customer id and container id respectively. This information is crucial in managing customer relationship and ensuring that customers are satisfied with the services provided.
- The company head can also get the total count of bookings in each and every port in the routes. This information is essential in understanding the demand for shipping services in each port and planning accordingly.
- Through this assignment, I was able to learn how to draw ER diagrams for different case studies and understand the importance of data normalization. Normalizing the data was an essential step in ensuring that data redundancy was minimized, and data consistency was maintained throughout the database.
- This coursework also helped to gain the insights about resolving one-to-one and many-to-many relationships practically. Through this process, I was able to understand the rules associated with resolving these relationships and apply them to different case studies.
- This also provided me with practical understanding of resolving one-to-one and many-to-many relationships, as well as a deep understanding of SQL queries and their associated functions.
- Regarding the data preparation and system requirements, it can be reported that the process of creating the data from scratch and addressing the complexity of the relationship between the data was successful. Furthermore, the majority of the system requirements were adequately addressed. However, the task of data filtering posed a significant challenge that required resolution. It is worth noting that considerable progress has been made in this area, and the desired outcome is nearly achieved.
- Through this, I was able to develop my skills in drawing ER diagrams for different case studies and loading data into databases. This helped to improve the understanding of SQL queries, including the various terms associated with it and their functions.

- Overall, this allowed me to enhance my knowledge and skills in database management, providing me with the necessary tools to handle complex data management tasks with ease.
- Upon reflecting on the completed project, I have identified an area that could have been improved upon. Although it was able to successfully manage the relationships within the data. In future projects, I aim to explore more complex relationships and strive to extract even more information through highly complex queries. This would enable to provide more comprehensive insights and solutions for ultimately enhancing the project's outcome.

## **ORACLE APEX**

**Workspace Name :** sujaygrama1  
**Email-id :** [K2201621@KINGSTON.AC.UK](mailto:K2201621@KINGSTON.AC.UK)  
**Password :** kingston@1234