

**Assignment 1 :Royal Anchor Shipping Database Report**

NAME : Madhavi Sanjay Kunte

STUDENT ID : K2461488

MODULE : Databases and Data Management

MODULE ID : CI7320

PROFESSOR : Dr Beryl Jones

DATE : 15 April 2025

## Table of Contents

<b>1.Chapter 1: The Entity Relationship Diagram with constraints and assumption.....</b>	<b>4</b>
1.1 Assumptions.....	5
1.2 Constraints.....	5
<b>2.Chapter 2: SQL Table definitions.....</b>	<b>6</b>
2.1 Table 1-Harbors .....	6
2.2 Table 2-Ship Categories.....	6
2.3 Table 3-Cargo(containers) .....	7
2.4 Table 4-Clients.....	8
2.5 Table 5-Staff(crew) Positions .....	8
2.6 Table 6-Ships (vessels).....	9
2.7 Table 7-Journeys(route).....	10
2.8 Table 8-Personnel.....	10
2.9 Table 9-Voyages.....	11
2.10Table 10-Cargo Boxes table.....	12
2.11 Table 11-JourneyLegs.....	12
2.12 Table 12-Shipment Orders.....	13
2.13 Table 13-Ship personnel(crew) Assignments.....	14
2.14 Table 14-Voyage Itinerary.....	14
2.15 Table 15-Shipping Rates.....	15
2.16 Table 16-Shipment.....	16
2.17 Definition of Sequences for Auto-Incrementing Primary Keys.....	17
<b>Chapter 3: Multiplicity of Relationships.....</b>	<b>26</b>
3.1 Insert Queries for Harbors Table.....	26
3.2 Insert Queries for Ship Categories Table.....	26
3.3 Insert Queries for Cargo Box Types Table.....	28
3.4 Insert Queries for Ship Categories Table.....	30
3.5 Insert Queries for Staff Positions Table.....	31
3.6 Insert Queries for Ship Table.....	33
3.7 Insert Queries for Journeys Table.....	34
3.8 Insert Queries for Personnel Table.....	34
3.9 Insert Queries for Voyages Table.....	36
3.10 Insert Queries for Cargo Boxes Table.....	37
3.11-Insert Queries for Journey Legs Table.....	39
3.12 Insert Queries for Shipment Orders Table.....	40
3.13 Insert Queries for Ship Personnel Assignments Table.....	41
3.14 Insert Queries for Voyage Stops Table.....	43
3.15 Insert Queries for Shipping Rates Table.....	45
3.16 Insert Queries for Shipments Table.....	46
3.17 Multiplicity of Relationships(Queries).....	48

<b>4 Chapter 4: Queries to demonstrate that your database meets the requirements of the System.....</b>	<b>50</b>
4.1 Query 1: Displaying Journey Paths with Stop Order.....	50
4.2 Query 2: Ship Voyage Summary with Container Count.....	51
4.3 Query 3: Shipment Details with Customer Information.....	52
4.4 Query 4: List of Ready Cargo Boxes with Location and Weight Capacity.....	54
4.5 Query 5: Shipment Order Tracking with Vessel and Harbor Details.....	54
4.6 Query 6: Shipment Order Tracking with Vessel and Harbor Details.....	55
<b>5 Chapter 5: Conclusion.....</b>	<b>56</b>

# 1. Chapter 1: The Entity Relationship Diagram with constraints and assumptions



## **1.1 Assumptions:**

- 1.Every single ship will have a unique registry code and it will be assigned to single ship category
- 2.Personnel (crew members) will be assigned to ships through the ship personnel table and assignments will include starting and ending date.
3. Cargo boxes will be assigned to ships and it will be tracked by their current condition and location.
- 4.Every cargo box is defined by a cargo box type, which demonstrates dimensions and capacity of the cargo box.
- 5.Every Client can place shipment order and one order can have multiple shipments.
- 6.Shipment orders have a defined pickup and delivery harbor, which are used to calculate shipping rates.
- 7.Voyages will show journeys taken by ships, and each voyage carries shipments from one harbor to another harbor.
- 8.Journey logs track stops at harbors, including docking and departure times.
- 9.Shipping cost will depend on the journey and cargo box type, with valid time periods for each rate.
- 10.Every journey can consist of multiple voyage logs, indicating a multi-stop trip.
- 11.Each personnel has a position from the staff positions table, which may include whether the position is a command role.
- 12.Ship categories define structural limits, such as max weight and number of containers.
- 13.Harbors are uniquely identified and may act as both origin and destination points for orders and voyages.

## **1.2 Constraints:**

### **-Primary & Foreign Keys:**

-All tables include primary keys for unique identification and foreign keys to maintain valid links between related records.

### **-Relationships:** One-to-many relationships exist between

-Ship Categories and Ships

-Journeys and Journey Legs

### **-Referential integrity:**

Orders are referenced with a valid client and harbors (pickup and delivery).

### **-Unique Fields:**

Fields are marked as **UK (unique keys)** to avoid duplicates.

### **Categorical Integrity:**

-Ships is assigned to a valid category defined in ship categories

-Personnel is assigned to valid staff positions.

## **2. Chapter 2: SQL Table definitions**

### **2.1 Table 1-Harbors**

```
CREATE TABLE harbors (
    harbor_id NUMBER PRIMARY KEY,
    harbor_name VARCHAR2(100) NOT NULL,
    nation VARCHAR2(100) NOT NULL,
    location_address VARCHAR2(200),
    contact_phone VARCHAR2(20),
    contact_email VARCHAR2(100),
    CONSTRAINT uk_harbor_name UNIQUE (harbor_name)
)
```

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', 'Gallery', 'Search', and user information 'Madhavi Kunte k2461488'. The main area is titled 'HARBORS' and displays the table structure. On the left, there's a tree view of tables: CARGO\_BOXES, CARGO\_BOX\_TYPES, CLIENTS, HARBORS (selected), HTMLDB\_PLAN\_TABLE, JOURNEYS, JOURNEY\_LEGS, PERSONNEL, SHIPMENTS, SHIPMENT\_ORDERS, SHIPPING\_RATES, SHIPS, SHIP\_CATEGORIES, SHIP\_PERSONNEL, and STAFF\_POSITIONS. The right panel shows the 'Columns' tab for the HARBORS table, listing columns: HARBOR\_ID (NUMBER, Primary Key, Not Null), HARBOR\_NAME (VARCHAR2(100 BYTE), Not Null), NATION (VARCHAR2(100 BYTE), Not Null), LOCATION\_ADDRESS (VARCHAR2(200 BYTE), Null), CONTACT\_PHONE (VARCHAR2(20 BYTE), Null), and CONTACT\_EMAIL (VARCHAR2(100 BYTE), Null). The bottom status bar shows '1 - 6 of 6' and 'Oracle APEX 24.2.2'.

### **2.2 Table 2-Ship Categories**

```
CREATE TABLE ship_categories (
    category_id CHAR(1) PRIMARY KEY,
    category_name VARCHAR2(10) NOT NULL,
    max_length_meters NUMBER(7,2) NOT NULL,
    max_containers NUMBER(6) NOT NULL,
    fleet_count NUMBER(3) NOT NULL,
    CONSTRAINT uk_ship_category_name UNIQUE (category_name)
)
```

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, user profile for 'Madhavi Kunte' (k2461488), and a schema dropdown set to 'WKSP\_K2461488'. The main area is titled 'SHIP\_CATEGORIES' and displays its column structure. The columns are:

Column Name	Data Type	Nullable	Default	Primary Key	Comment	Identity
CATEGORY_ID	CHAR(1 BYTE)	N		1		
CATEGORY_NAME	VARCHAR2(10 BYTE)	N				
MAX_LENGTH_METERS	NUMBER(7,2)	N				
MAX_CONTAINERS	NUMBER(6,0)	N				
FLEET_COUNT	NUMBER(3,0)	N				

At the bottom, it says '1 cells selected' and '1 - 5 of 5'. The footer includes copyright information for Oracle and the APEX version 'Oracle APEX 24.2.2'.

### **2.3 Table 3-Cargo(containers) Box Types**

```
CREATE TABLE cargo_box_types (
    box_type_id NUMBER PRIMARY KEY,
    box_model VARCHAR2(100) NOT NULL,
    classification VARCHAR2(50) NOT NULL,
    outer_length_m NUMBER(5,3) NOT NULL,
    outer_width_m NUMBER(5,3) NOT NULL,
    outer_height_m NUMBER(5,3) NOT NULL,
    inner_length_m NUMBER(5,3) NOT NULL,
    inner_width_m NUMBER(5,3) NOT NULL,
    inner_height_m NUMBER(5,3) NOT NULL,
    total_weight_capacity_kg NUMBER(8),
    empty_weight_kg NUMBER(8),
    payload_capacity_kg NUMBER(8),
    CONSTRAINT uk_box_model UNIQUE (box_model)
)
```

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, user profile for 'Madhavi Kunte' (k2461488), and a schema dropdown set to 'WKSP\_K2461488'. The main area is titled 'CARGO\_BOX\_TYPES' and displays its column structure. The columns are:

Column Name	Data Type	Nullable	Default	Primary Key	Comment	Identity
BOX_TYPE_ID	NUMBER	N		1		
BOX_MODEL	VARCHAR2(100 BYTE)	N				
CLASSIFICATION	VARCHAR2(50 BYTE)	N				
OUTER_LENGTH_M	NUMBER(5,3)	N				
OUTER_WIDTH_M	NUMBER(5,3)	N				
OUTER_HEIGHT_M	NUMBER(5,3)	N				
INNER_LENGTH_M	NUMBER(5,3)	N				
INNER_WIDTH_M	NUMBER(5,3)	N				
INNER_HEIGHT_M	NUMBER(5,3)	N				
TOTAL_WEIGHT_CAPACITY_KG	NUMBER(8,0)	Y				
EMPTY_WEIGHT_KG	NUMBER(8,0)	Y				
PAYOUT_CAPACITY_KG	NUMBER(8,0)	Y				

At the bottom, it says '1 cells selected' and '1 - 12 of 12'. The footer includes copyright information for Oracle and the APEX version 'Oracle APEX 24.2.2'.

## **2.4 Table 4-Clients**

```
CREATE TABLE clients (
    client_id NUMBER PRIMARY KEY,
    company_name VARCHAR2(100) NOT NULL,
    representative VARCHAR2(100),
    phone_number VARCHAR2(20),
    company_email VARCHAR2(100) NOT NULL,
    business_address VARCHAR2(200),
    CONSTRAINT uk_company_email UNIQUE (company_email)
)
```

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. On the right, there's a user profile for 'Madhavi Kunte' (k2461488). The main area is titled 'CLIENTS' and displays the table structure. The 'Columns' tab is selected, showing the following columns:

Column Name	Data Type	Nullable	Default	Primary Key..	Comment	Identity
CLIENT_ID	NUMBER	N		1		
COMPANY_NAME	VARCHAR2(100 BY...	N				
REPRESENTATIVE	VARCHAR2(100 BY...	Y				
PHONE_NUMBER	VARCHAR2(20 BYTE)	Y				
COMPANY_EMAIL	VARCHAR2(100 BY...	N				
BUSINESS_ADDRESS	VARCHAR2(200 B...	Y				

Below the table definition, the status bar shows '1 cells selected' and '1 - 6 of 6'. The bottom right corner indicates 'Oracle APEX 24.2.2'.

## **2.5 Table 5-Staff(crew) Positions**

```
CREATE TABLE staff_positions (
    position_id NUMBER PRIMARY KEY,
    position_title VARCHAR2(100) NOT NULL,
    division VARCHAR2(50) NOT NULL,
    is_command CHAR(1) DEFAULT 'N' NOT NULL,
    CONSTRAINT uk_position_title UNIQUE (position_title),
    CONSTRAINT chk_is_command CHECK (is_command IN ('Y', 'N'))
)
```

The screenshot shows the Oracle APEX SQL Workshop interface. The left sidebar lists various database objects: HARBORS, HTMLDB\_PLAN\_TABLE, JOURNEYS, JOURNEY\_LEGS, PERSONNEL, SHIPMENTS, SHIPMENT\_ORDERS, SHIPPING\_RATES, SHIPPING\_RATES, SHIPS, SHIP\_CATEGORIES, SHIP\_PERSONNEL, STAFF\_POSITIONS (which is selected), VOYAGES, and VOYAGE\_STOPS. Below this is a navigation bar with Views, Indexes, and Sequences. The main panel displays the 'STAFF\_POSITIONS' table creation screen. The table has four columns: POSITION\_ID (NUMBER, Nullable N, Default null, Primary Key), POSITION\_TITLE (VARCHAR2(100 BYTE), Nullable N), DIVISION (VARCHAR2(50 BYTE), Nullable N), and IS\_COMMAND (CHAR(1 BYTE), Nullable N, Default 'N'). Buttons at the top include '+ Add Column', 'Modify Column', 'Rename Column', 'Drop Column', 'UI Defaults', 'Refresh', and 'More'. The status bar at the bottom indicates '1 - 4 of 4' and 'Oracle APEX 24.2.2'.

## 2.6 Table 6-Ships (vessels)

```
CREATE TABLE ships (
    registry_code VARCHAR2(7) PRIMARY KEY,
    ship_name VARCHAR2(100) NOT NULL,
    category_id CHAR(1) NOT NULL,
    max_velocity_knots NUMBER(3,1),
    total_tonnage NUMBER(10,2),
    cargo_tonnage NUMBER(10,2),
    hull_length_m NUMBER(6,2),
    hull_width_m NUMBER(6,2),
    construction_date DATE,
    CONSTRAINT fk_ship_category FOREIGN KEY (category_id) REFERENCES
ship_categories(category_id),
    CONSTRAINT uk_ship_name UNIQUE (ship_name)
)
```

The screenshot shows the Oracle APEX SQL Workshop interface. The left sidebar lists various database objects: PERSONNEL, SHIPMENTS, SHIPMENT\_ORDERS, SHIPPING\_RATES, SHIPS (which is selected), SHIP\_CATEGORIES, SHIP\_PERSONNEL, STAFF\_POSITIONS, VOYAGES, and VOYAGE\_STOPS. Below this is a navigation bar with Views, Indexes, and Sequences. The main panel displays the 'SHIPS' table creation screen. The table has nine columns: REGISTRY\_CODE (VARCHAR2(7 BYTE), Nullable N, Primary Key), SHIP\_NAME (VARCHAR2(100 BYTE), Nullable N), CATEGORY\_ID (CHAR(1 BYTE), Nullable N), MAX\_VELOCITY\_KNOTS (NUMBER(3,1), Nullable Y), TOTAL\_TONNAGE (NUMBER(10,2), Nullable Y), CARGO\_TONNAGE (NUMBER(10,2), Nullable Y), HULL\_LENGTH\_M (NUMBER(6,2), Nullable Y), HULL\_WIDTH\_M (NUMBER(6,2), Nullable Y), and CONSTRUCTION\_DATE (DATE, Nullable Y). Buttons at the top include '+ Add Column', 'Modify Column', 'Rename Column', 'Drop Column', 'UI Defaults', 'Refresh', and 'More'. The status bar at the bottom indicates '1 - 9 of 9' and 'Oracle APEX 24.2.2'.

## 2.7 Table 7-Journeys(route)

```
CREATE TABLE journeys (
    journey_id NUMBER PRIMARY KEY,
    journey_code VARCHAR2(100) NOT NULL,
    journey_details VARCHAR2(1000),
    CONSTRAINT uk_journey_code UNIQUE (journey_code)
)
```

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX' (with a red icon), 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows a user profile 'MK Madhavi Kunte' and the schema 'WKSP\_I2461488'. The main area is titled 'JOURNEYS' and displays the table structure. On the left, a sidebar lists various tables: CARGO\_BOXES, CARGO\_BOX\_TYPES, CLIENTS, HARBORS, HTMLDB\_PLAN\_TABLE, JOURNEYS (selected and highlighted in green), JOURNEY\_LEGS, PERSONNEL, SHIPMENTS, SHIPMENT\_ORDERS, SHIPPING\_RATES, SHIPS, SHIP\_CATEGORIES, SHIP\_PERSONNEL, and STAFF\_POSITIONS. The 'Tables' section is expanded. The 'Columns' tab of the JOURNEYS table is selected, showing the following columns:

Column Name	Data Type	Nullable	Default	Primary Key	Comment	Identity
JOURNEY_ID	NUMBER	N		1		
JOURNEY_CODE	VARCHAR2(100 BYTE)	N				
JOURNEY_DETAILS	VARCHAR2(1000 BY...	Y				

At the bottom of the page, there are footer links for 'k2461488@kingston.ac.uk', 'en', 'Copyright © 1999, 2024, Oracle and/or its affiliates.', and 'Oracle APEX 24.2.2'.

## 2.8 Table 8-Personnel (crew members)

```
CREATE TABLE personnel (
    personnel_id NUMBER PRIMARY KEY,
    given_name VARCHAR2(50) NOT NULL,
    family_name VARCHAR2(50) NOT NULL,
    position_id NUMBER NOT NULL,
    birthdate DATE,
    citizenship VARCHAR2(50),
    license_number VARCHAR2(50),
    license_expiry DATE,
    mobile_number VARCHAR2(20),
    personal_email VARCHAR2(100),
    CONSTRAINT fk_person_position FOREIGN KEY (position_id) REFERENCES
staff_positions(position_id)
)
```

APEX App Builder SQL Workshop Team Development Gallery

Object Browser

**PERSONNEL**

Column Name	Data Type	Nullable	Default	Primary Key	Comment	Identity
PERSONNEL_ID	NUMBER	N		1		
GIVEN_NAME	VARCHAR2(50 BYTE)	N				
FAMILY_NAME	VARCHAR2(50 BYTE)	N				
POSITION_ID	NUMBER	N				
BIRTHDATE	DATE	Y				
CITIZENSHIP	VARCHAR2(50 BYTE)	Y				
LICENSE_NUMBER	VARCHAR2(50 BYTE)	Y				
LICENSE_EXPIRY	DATE	Y				
MOBILE_NUMBER	VARCHAR2(20 BYTE)	Y				
PERSONAL_EMAIL	VARCHAR2(100 BYTE)	Y				

1 - 10 of 10

Copyright © 1999, 2024, Oracle and/or its affiliates.

Oracle APEX 24.2.2

## 2.9 Table 9-Voyages

```
CREATE TABLE voyages (
    voyage_id NUMBER PRIMARY KEY,
    ship_registry VARCHAR2(7) NOT NULL,
    journey_id NUMBER NOT NULL,
    departure_date DATE NOT NULL,
    arrival_date DATE,
    voyage_status VARCHAR2(20) DEFAULT 'Scheduled' NOT NULL,
    CONSTRAINT fk_voyage_ship FOREIGN KEY (ship_registry) REFERENCES ships(registry_code),
    CONSTRAINT fk_voyage_journey FOREIGN KEY (journey_id) REFERENCES journeys(journey_id),
    CONSTRAINT chk_voyage_status CHECK (voyage_status IN ('Scheduled', 'Active',
    'Finished'))
)
```

APEX App Builder SQL Workshop Team Development Gallery

Object Browser

**VOYAGES**

Column Name	Data Type	Nullable	Default	Primary Key	Comment	Identity
VOYAGE_ID	NUMBER	N		1		
SHIP_REGISTRY	VARCHAR2(7 BYTE)	N				
JOURNEY_ID	NUMBER	N				
DEPARTURE_DATE	DATE	N				
ARRIVAL_DATE	DATE	Y				
VOYAGE_STATUS	VARCHAR2(20 BYTE)	N	'Scheduled'			

1 - 6 of 6

Copyright © 1999, 2024, Oracle and/or its affiliates.

Oracle APEX 24.2.2

## 2.10 Table 10-Cargo Boxes table

```
CREATE TABLE cargo_boxes (
    box_id VARCHAR2(20) PRIMARY KEY,
    box_type_id NUMBER NOT NULL,
    current_condition VARCHAR2(20) NOT NULL,
    harbor_location NUMBER,
    assigned_ship VARCHAR2(7),
    CONSTRAINT fk_box_type FOREIGN KEY (box_type_id) REFERENCES
cargo_box_types(box_type_id),
    CONSTRAINT fk_box_harbor FOREIGN KEY (harbor_location) REFERENCES harbors(harbor_id),
    CONSTRAINT fk_box_ship FOREIGN KEY (assigned_ship) REFERENCES ships(registry_code),
    CONSTRAINT chk_box_location CHECK (
        (harbor_location IS NOT NULL AND assigned_ship IS NULL) OR
        (harbor_location IS NULL AND assigned_ship IS NOT NULL) OR
        (harbor_location IS NULL AND assigned_ship IS NULL)
    )
)
```

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a user profile for 'Madhavi Kunte' (k2461488). The main area is titled 'CARGO\_BOXES'. On the left, a sidebar lists various tables: CARGO\_BOXES (selected), CARGO\_BOX\_TYPES, CLIENTS, HARBORS, HTMLDB\_PLAN\_TABLE, JOURNEYS, JOURNEY\_LEGS, PERSONNEL, SHIPMENTS, SHIPMENT\_ORDERS, SHIPPING\_RATES, SHIPS, SHIP\_CATEGORIES, SHIP\_PERSONNEL, and STAFF\_POSITIONS. The central panel displays the columns of the CARGO\_BOXES table:

Column Name	Data Type	Nullable	Default	Primary Key	Comment	Identity
BOX_ID	VARCHAR2(20 BYTE)	N		1		
BOX_TYPE_ID	NUMBER	N				
CURRENT_CONDITION	VARCHAR2(20 BYTE)	N				
HARBOR_LOCATION	NUMBER	Y				
ASSIGNED_SHIP	VARCHAR2(7 BYTE)	Y				

At the bottom of the central panel, it says '1 cells selected'. The footer includes links for k2461488@kingston.ac.uk, k2461488, en, Copyright © 1999, 2024, Oracle and/or its affiliates, and Oracle APEX 24.2.2.

## 2.11 Table 11-Journey Legs

```
CREATE TABLE journey_legs (
    leg_id NUMBER PRIMARY KEY,
    journey_id NUMBER NOT NULL,
    harbor_id NUMBER NOT NULL,
    stop_order NUMBER(3) NOT NULL,
    CONSTRAINT fk_leg_journey FOREIGN KEY (journey_id) REFERENCES journeys(journey_id),
    CONSTRAINT fk_leg_harbor FOREIGN KEY (harbor_id) REFERENCES harbors(harbor_id),
    CONSTRAINT uk_journey_harbor_order UNIQUE (journey_id, harbor_id, stop_order)
)
```

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. On the right, there's a search bar, a user icon, and a profile for 'Medhavi Kunte k2461488'. The main area is titled 'JOURNEY\_LEGS' and displays its column structure. The columns are:

Column Name	Data Type	Nullable	Default	Primary Key	Comment	Identity
LEG_ID	NUMBER	N		1		
JOURNEY_ID	NUMBER	N				
HARBOR_ID	NUMBER	N				
STOP_ORDER	NUMBER(3,0)	N				

Below the table structure, it says '1 cells selected' and '1 - 4 of 4'. The bottom of the screen shows the copyright notice 'Copyright © 1999, 2024, Oracle and/or its affiliates.' and 'Oracle APEX 24.2.2'.

## 2.12 Table 12-Shipment Orders

```
CREATE TABLE shipment_orders (
    order_id NUMBER PRIMARY KEY,
    client_id NUMBER NOT NULL,
    pickup_harbor NUMBER NOT NULL,
    delivery_harbor NUMBER NOT NULL,
    order_date DATE DEFAULT SYSDATE NOT NULL,
    order_status VARCHAR2(20) DEFAULT 'Confirmed' NOT NULL,
    CONSTRAINT fk_order_client FOREIGN KEY (client_id) REFERENCES clients(client_id),
    CONSTRAINT fk_order_pickup FOREIGN KEY (pickup_harbor) REFERENCES harbors(harbor_id),
    CONSTRAINT fk_order_delivery FOREIGN KEY (delivery_harbor) REFERENCES
    harbors(harbor_id),
    CONSTRAINT chk_different_harbors CHECK (pickup_harbor != delivery_harbor),
    CONSTRAINT chk_order_status CHECK (order_status IN ('Confirmed', 'Shipping',
    'Completed', 'Voided'))
)
```

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. On the right, there's a search bar, a user icon, and a profile for 'Medhavi Kunte k2461488'. The main area is titled 'SHIPMENT\_ORDERS' and displays its column structure. The columns are:

Column Name	Data Type	Nullable	Default	Primary Key	Comment	Identity
ORDER_ID	NUMBER	N		1		
CLIENT_ID	NUMBER	N				
PICKUP_HARBOR	NUMBER	N				
DELIVERY_HARBOR	NUMBER	N				
ORDER_DATE	DATE	N	SYSDATE			
ORDER_STATUS	VARCHAR2(20 BYTE)	N	'Confirmed'			

Below the table structure, it says '1 cells selected' and '1 - 6 of 6'. The bottom of the screen shows the copyright notice 'Copyright © 1999, 2024, Oracle and/or its affiliates.' and 'Oracle APEX 24.2.2'.

## **2.13 Table 13-Ship personnel(crew) Assignments**

```
CREATE TABLE ship_personnel (
    assignment_id NUMBER PRIMARY KEY,
    ship_registry VARCHAR2(7) NOT NULL,
    personnel_id NUMBER NOT NULL,
    duty_start DATE NOT NULL,
    duty_end DATE,
    CONSTRAINT fk_sp_ship FOREIGN KEY (ship_registry) REFERENCES ships(registry_code),
    CONSTRAINT fk_sp_personnel FOREIGN KEY (personnel_id) REFERENCES
personnel(personnel_id),
    CONSTRAINT uk_ship_personnel_duty UNIQUE (ship_registry, personnel_id, duty_start)
)
```

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for App Builder, SQL Workshop, Team Development, and Gallery. The user is logged in as 'Madhavi Kunte' (k2461488). The main area displays the 'SHIP\_PERSONNEL' table structure. On the left, there is a sidebar with a tree view of other tables: HARBORS, JOURNEYS, PERSONNEL, SHIPMENTS, SHIPMENT\_ORDERS, SHIPPING\_RATES, SHIPS, and SHIP\_CATEGORIES. The 'SHIP\_PERSONNEL' table is selected and highlighted in green. The table structure is shown in a grid format with columns: Column Name, Data Type, Nullable, Default, Primary Key, Comment, and Identity. The primary key is 'ASSIGNMENT\_ID'. Other columns include 'SHIP\_REGISTRY', 'PERSONNEL\_ID', 'DUTY\_START', and 'DUTY\_END'. The bottom of the screen shows copyright information: Copyright © 1999, 2024, Oracle and/or its affiliates. and Oracle APEX 24.2.2.

## **2.14 Table 14-Voyage Itinerary**

```
CREATE TABLE voyage_stops (
    stop_id NUMBER PRIMARY KEY,
    voyage_id NUMBER NOT NULL,
    harbor_id NUMBER NOT NULL,
    docking_time DATE,
    departure_time DATE,
    stop_sequence NUMBER(3) NOT NULL,
    CONSTRAINT fk_stop_voyage FOREIGN KEY (voyage_id) REFERENCES voyages(voyage_id),
    CONSTRAINT fk_stop_harbor FOREIGN KEY (harbor_id) REFERENCES harbors(harbor_id),
    CONSTRAINT uk_voyage_harbor_sequence UNIQUE (voyage_id, harbor_id, stop_sequence),
    CONSTRAINT chk_docking_departure CHECK (departure_time >= docking_time)
)
```

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The user is logged in as 'Madhavi Kunte' (K2461488). The main area is titled 'VOYAGE\_STOPS' and displays the table structure with the following columns:

Column Name	Data Type	Nullable	Default	Primary Key	Comment	Identity
STOP_ID	NUMBER	N		1		
VOYAGE_ID	NUMBER	N				
HARBOR_ID	NUMBER	N				
DOCKING_TIME	DATE	Y				
DEPARTURE_TIME	DATE	Y				
STOP_SEQUENCE	NUMBER(5,0)	N				

On the left sidebar, under 'Object Browser', 'VOYAGE\_STOPS' is selected. Other objects listed include HARBORS, HTMldb\_plan\_table, JOURNEYS, JOURNEY\_LEGS, PERSONNEL, SHIPMENTS, SHIPMENT\_ORDERS, SHIPPING\_RATES, SHIPS, SHIP\_CATEGORIES, SHIP\_PERSONNEL, STAFF\_POSITIONS, VOYAGES, and VOYAGE\_STOPS. The bottom status bar shows 'Copyright © 1999, 2024, Oracle and/or its affiliates.' and 'Oracle APEX 24.2.2'.

## 2.15 Table 15-Shipping Rates

```

CREATE TABLE shipping_rates (
    rate_id NUMBER PRIMARY KEY,
    departure_harbor NUMBER NOT NULL,
    arrival_harbor NUMBER NOT NULL,
    box_type_id NUMBER NOT NULL,
    standard_fee NUMBER(10,2) NOT NULL,
    valid_from DATE NOT NULL,
    valid_until DATE,
    CONSTRAINT fk_rate_departure FOREIGN KEY (departure_harbor) REFERENCES
    harbors(harbor_id),
    CONSTRAINT fk_rate_arrival FOREIGN KEY (arrival_harbor) REFERENCES harbors(harbor_id),
    CONSTRAINT fk_rate_box_type FOREIGN KEY (box_type_id) REFERENCES
    cargo_box_types(box_type_id),
    CONSTRAINT uk_rate_route_box UNIQUE (departure_harbor, arrival_harbor, box_type_id,
    valid_from),
    CONSTRAINT chk_rate_harbors CHECK (departure_harbor != arrival_harbor),
    CONSTRAINT chk_rate_dates CHECK (valid_until >= valid_from)
)

```

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, a user icon for 'Madhavi Kunte k2461488', and a help icon. The main area is titled 'Object Browser' and shows a tree view of database objects. Under the 'SHIPMENT\_ORDERS' category, the 'SHIPPING\_RATES' table is selected. The table structure is displayed in a grid with columns: Column Name, Data Type, Nullable, Default, Primary Key, Comment, and Identity. The 'RATE\_ID' column is defined as NUMBER, nullable 'N', and is marked as the Primary Key ('1'). Other columns include DEPARTURE\_HARBOR, ARRIVAL\_HARBOR, BOX\_TYPE\_ID, STANDARD\_FEE, VALID\_FROM, and VALID\_UNTIL.

## 2.16 Table 16-Shippments

```
CREATE TABLE shipments (
    shipment_id NUMBER PRIMARY KEY,
    order_id NUMBER NOT NULL,
    box_id VARCHAR2(20) NOT NULL,
    commodity_type VARCHAR2(100) NOT NULL,
    commodity_details VARCHAR2(1000),
    mass_tonnes NUMBER(10,2),
    voyage_id NUMBER,
    pickup_date DATE,
    delivery_date DATE,
    CONSTRAINT fk_shipment_order FOREIGN KEY (order_id) REFERENCES
shipment_orders(order_id),
    CONSTRAINT fk_shipment_box FOREIGN KEY (box_id) REFERENCES cargo_boxes(box_id),
    CONSTRAINT fk_shipment_voyage FOREIGN KEY (voyage_id) REFERENCES voyages(voyage_id),
    CONSTRAINT chk_shipment_dates CHECK (delivery_date >= pickup_date)
)
```

The screenshot shows the Oracle APEX SQL Workshop interface, similar to the previous one but with a different table selected. The main area is titled 'Object Browser' and shows a tree view of database objects. Under the 'SHIPMENT\_ORDERS' category, the 'SHIPPMENTS' table is selected. The table structure is displayed in a grid with columns: Column Name, Data Type, Nullable, Default, Primary Key, Comment, and Identity. The 'SHIPMENT\_ID' column is defined as NUMBER, nullable 'N', and is marked as the Primary Key ('1'). Other columns include ORDER\_ID, BOX\_ID, COMMODITY\_TYPE, COMMODITY\_DETAILS, MASS\_TONNES, VOYAGE\_ID, PICKUP\_DATE, and DELIVERY\_DATE.

Below is the screenshot of tables created in Oracle:

Number ↑	Elapsed	Statement	Feedback	Rows
1	0.08	CREATE TABLE harbors ( harbor_id NUMBER PRIMARY KEY,	Table created.	0
2	0.04	CREATE TABLE ship_categories ( category_id CHAR(1) PRIM	Table created.	0
3	0.05	CREATE TABLE cargo_box_types ( box_type_id NUMBER PRIMA	Table created.	0
4	0.04	CREATE TABLE clients ( client_id NUMBER PRIMARY KEY,	Table created.	0
5	0.04	CREATE TABLE staff_positions ( position_id NUMBER PRIMA	Table created.	0
6	0.05	CREATE TABLE ships ( registry_code VARCHAR2(7) PRIMARY	Table created.	0
7	0.04	CREATE TABLE journeys ( journey_id NUMBER PRIMARY KEY,	Table created.	0
8	0.03	CREATE TABLE personnel ( personnel_id NUMBER PRIMARY KE	Table created.	0
9	0.04	CREATE TABLE voyages ( voyage_id NUMBER PRIMARY KEY,	Table created.	0
10	0.04	CREATE TABLE cargo_boxes ( box_id VARCHAR2(20) PRIMARY	Table created.	0
11	0.05	CREATE TABLE journey_legs ( leg_id NUMBER PRIMARY KEY,	Table created.	0
12	0.05	CREATE TABLE shipment_orders ( order_id NUMBER PRIMARY	Table created.	0
13	0.05	CREATE TABLE ship_personnel ( assignment_id NUMBER PRIM	Table created.	0
14	0.05	CREATE TABLE voyage_stops ( stop_id NUMBER PRIMARY KEY,	Table created.	0
15	0.06	CREATE TABLE shipping_rates ( rate_id NUMBER PRIMARY KE	Table created.	0

Img: Tables created in Oracle

## 2.17 Definition of Sequences for Auto-Incrementing Primary Keys

### Explanation:

In Oracle databases, SEQUENCE objects are used to generate unique numeric values—typically for auto-incrementing primary keys. These sequences ensure that each new record inserted into a table gets a unique identifier, which is essential for maintaining data integrity and enabling proper entity relationships.

The following sequences are created for various entities in the database to support this functionality:

Sequence Name	Purpose
harbor_seq	Generates unique IDs for harbors
box_type_seq	Generates unique IDs for cargo box types
client_seq	Generates unique IDs for clients
journey_seq	Generates unique IDs for journeys
leg_seq	Generates unique IDs for journey legs or stops
position_seq	Generates unique IDs for staff positions
personnel_seq	Generates unique IDs for personnel (crew)
assignment_seq	Generates unique IDs for ship personnel assignments
voyage_seq	Generates unique IDs for voyages
stop_seq	Generates unique IDs for voyage stops
order_seq	Generates unique IDs for shipment orders
shipment_seq	Generates unique IDs for individual shipments
rate_seq	Generates unique IDs for shipping rates

## **2.18 Queries to Generate sequences:**

1.CREATE SEQUENCE harbor\_seq START WITH 1 INCREMENT BY 1

The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'APEX' is selected. Below it, 'Object Browser' is open. The left pane shows a tree structure of objects under 'VOYAGE\_STOPS': 'Views', 'Indexes', 'Sequences'. The 'SEQUENCES' node is expanded, and 'HARBOR\_SEQ' is selected, highlighted with a green border. The right pane displays the 'Object Details' for 'HARBOR\_SEQ'. The details are as follows:

Attribute	Value
Min Value	1
Max Value	99999999999999999999999999999999
Increment By	1
Cycle Flag	N
Order Flag	N
Cache Size	20
Last Number	21

2.CREATE SEQUENCE box\_type\_seq START WITH 1 INCREMENT BY 1

↑ Object Browser

Type to filter... +

VOYAGE\_STOPS

> Views

> Indexes

Sequences

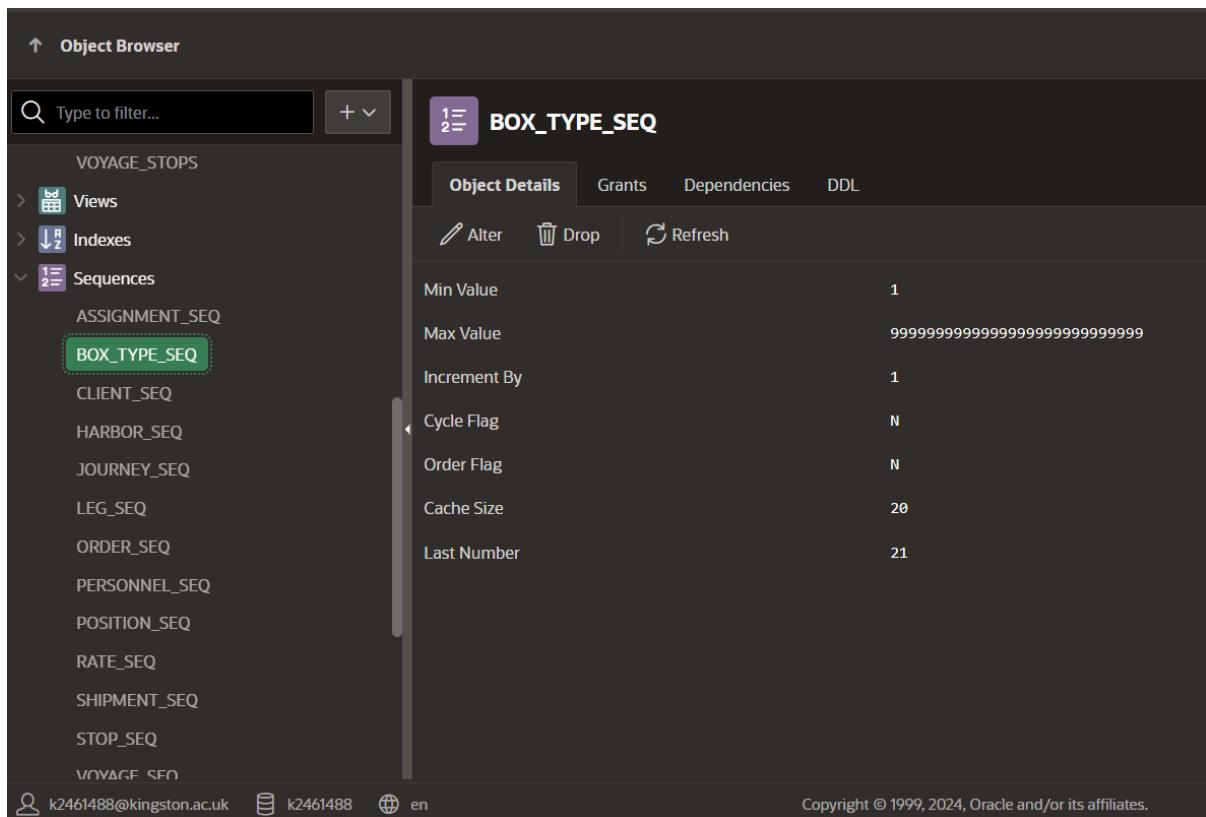
- ASSIGNMENT\_SEQ
- BOX\_TYPE\_SEQ**
- CLIENT\_SEQ
- HARBOR\_SEQ
- JOURNEY\_SEQ
- LEG\_SEQ
- ORDER\_SEQ
- PERSONNEL\_SEQ
- POSITION\_SEQ
- RATE\_SEQ
- SHIPMENT\_SEQ
- STOP\_SEQ
- VOYAGE\_SEQ

Object Details Grants Dependencies DDL

Alter Drop Refresh

Min Value	1
Max Value	99999999999999999999999999999999
Increment By	1
Cycle Flag	N
Order Flag	N
Cache Size	20
Last Number	21

k2461488@kingston.ac.uk k2461488 en Copyright © 1999, 2024, Oracle and/or its affiliates.



3.CREATE SEQUENCE client\_seq START WITH 1 INCREMENT BY 1

APEX App Builder SQL Workshop Team Development Gallery

↑ Object Browser

Type to filter... +

VOYAGE\_STOPS

> Views

> Indexes

Sequences

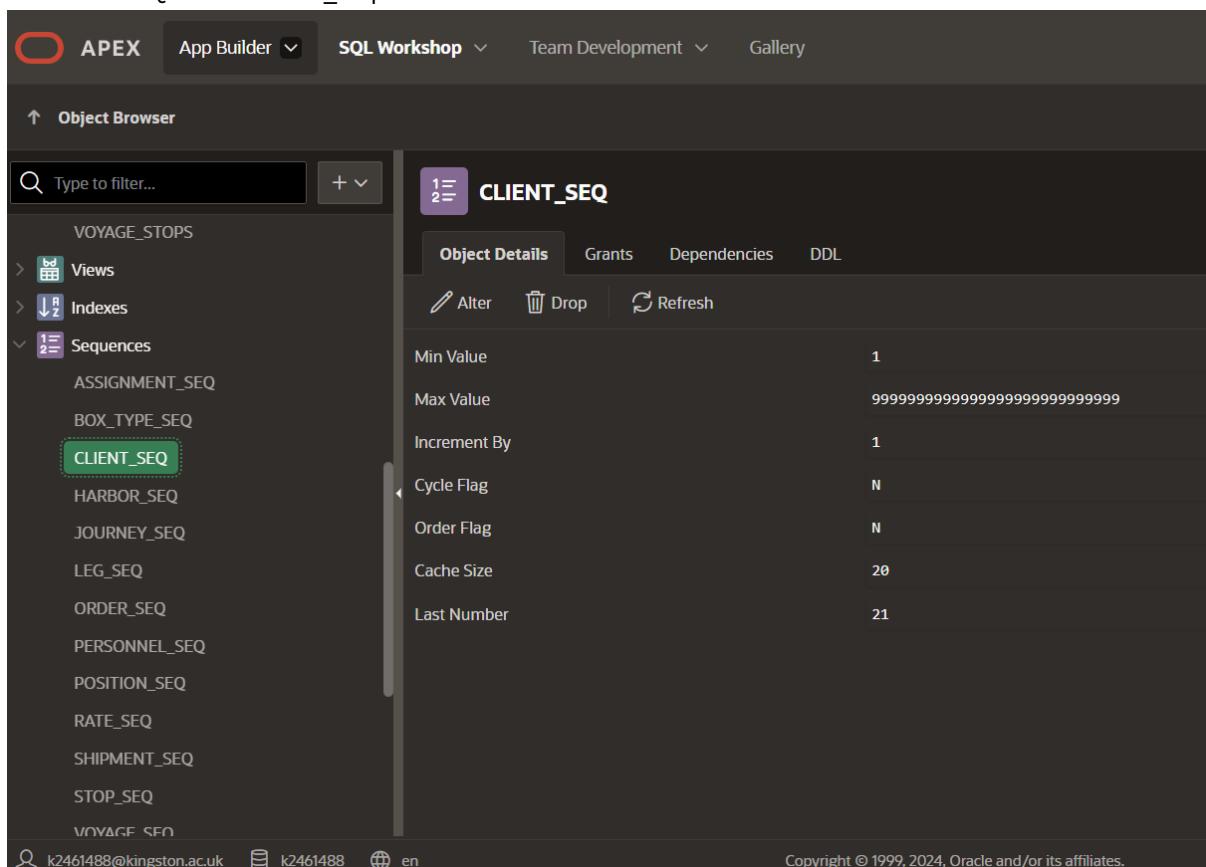
- ASSIGNMENT\_SEQ
- BOX\_TYPE\_SEQ
- CLIENT\_SEQ**
- HARBOR\_SEQ
- JOURNEY\_SEQ
- LEG\_SEQ
- ORDER\_SEQ
- PERSONNEL\_SEQ
- POSITION\_SEQ
- RATE\_SEQ
- SHIPMENT\_SEQ
- STOP\_SEQ
- VOYAGE\_SEQ

Object Details Grants Dependencies DDL

Alter Drop Refresh

Min Value	1
Max Value	99999999999999999999999999999999
Increment By	1
Cycle Flag	N
Order Flag	N
Cache Size	20
Last Number	21

k2461488@kingston.ac.uk k2461488 en Copyright © 1999, 2024, Oracle and/or its affiliates.



4. CREATE SEQUENCE journey\_seq START WITH 1 INCREMENT BY 1

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, the 'SQL Workshop' tab is selected. The left sidebar is titled 'Object Browser' and lists various database objects under 'VOYAGE\_STOPS', including 'Views', 'Indexes', and 'Sequences'. The 'Sequences' section contains items like 'ASSIGNMENT\_SEQ', 'BOX\_TYPE\_SEQ', 'CLIENT\_SEQ', 'HARBOR\_SEQ', 'JOURNEY\_SEQ' (which is highlighted with a green border), 'LEG\_SEQ', 'ORDER\_SEQ', 'PERSONNEL\_SEQ', 'POSITION\_SEQ', 'RATE\_SEQ', 'SHIPMENT\_SEQ', 'STOP\_SEQ', and 'VOYAGE\_SEQ'. The main panel displays the 'Object Details' for the 'JOURNEY\_SEQ' sequence. The details are as follows:

Setting	Value
Min Value	1
Max Value	99999999999999999999999999999999999999
Increment By	1
Cycle Flag	N
Order Flag	N
Cache Size	20
Last Number	21

At the bottom of the main panel, there are buttons for 'Alter', 'Drop', and 'Refresh'. The footer of the interface includes the user information 'k2461488@kingston.ac.uk' and 'k2461488', the language 'en', and the copyright notice 'Copyright © 1999, 2024, Oracle and/or its affiliates.'

5. CREATE SEQUENCE leg\_seq START WITH 1 INCREMENT BY 1

This screenshot is nearly identical to the previous one, showing the 'Object Details' for the 'LEG\_SEQ' sequence. The sequence details are as follows:

Setting	Value
Min Value	1
Max Value	99999999999999999999999999999999999999
Increment By	1
Cycle Flag	N
Order Flag	N
Cache Size	20
Last Number	21

The 'Object Browser' sidebar shows the same list of sequences, with 'LEG\_SEQ' also highlighted. The footer information remains the same.

6.CREATE SEQUENCE position\_seq START WITH 1 INCREMENT BY 1

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (which is currently selected), 'Team Development', and 'Gallery'. The left sidebar is titled 'Object Browser' and lists 'SEQUENCES' under 'VOYAGE\_STOPS', which contains various sequences like 'ASSIGNMENT\_SEQ', 'BOX\_TYPE\_SEQ', etc., with 'POSITION\_SEQ' highlighted in green. The main right panel shows the 'POSITION\_SEQ' object details. The 'Object Details' tab is active, displaying the following properties:

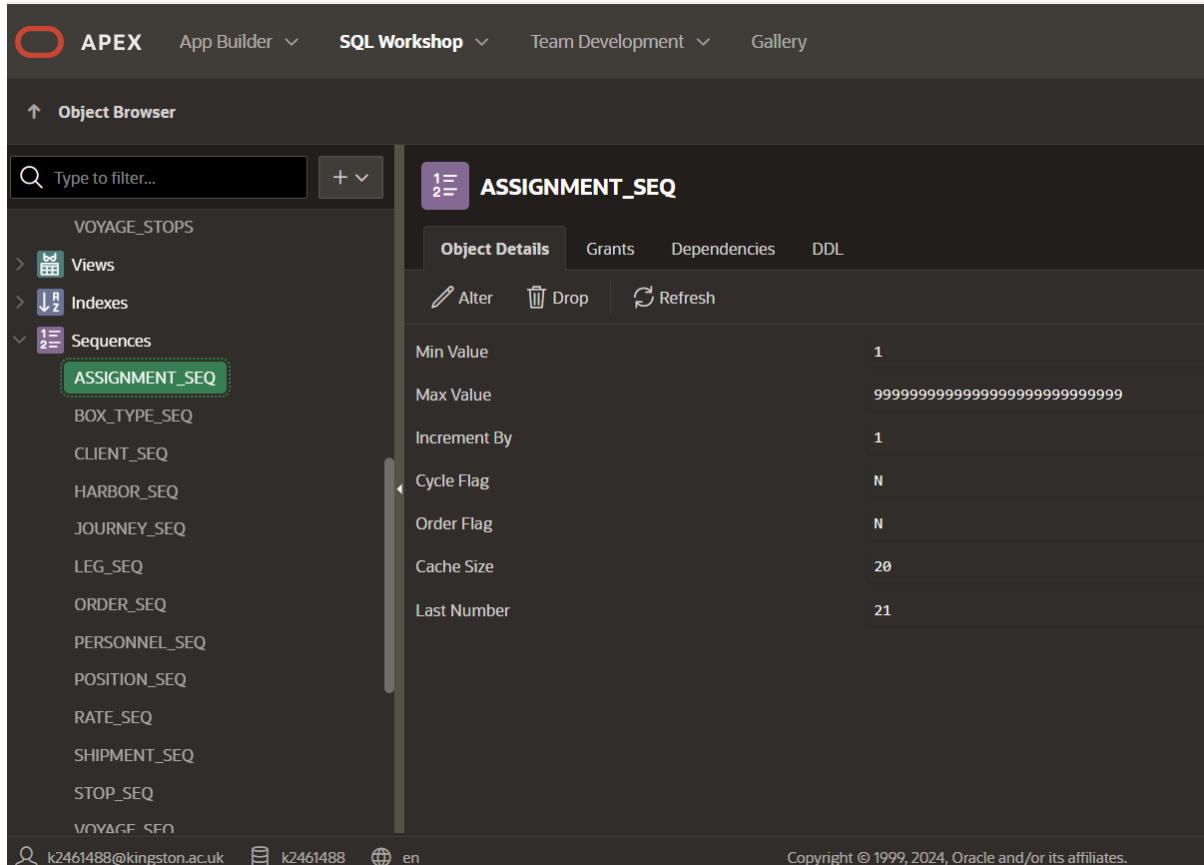
Property	Value
Min Value	1
Max Value	999999999999999999999999999999999
Increment By	1
Cycle Flag	N
Order Flag	N
Cache Size	20
Last Number	21

Below the table are buttons for 'Alter', 'Drop', and 'Refresh'. The bottom right corner of the main panel displays the copyright notice 'Copyright © 1999, 2024, Oracle and/or its affiliates.'

7.CREATE SEQUENCE personnel\_seq START WITH 1 INCREMENT BY 1

This screenshot is nearly identical to the previous one, showing the creation of a sequence named 'PERSONNEL\_SEQ'. The 'SEQUENCES' section in the Object Browser now lists 'PERSONNEL\_SEQ' as the selected item. The main right panel displays the 'PERSONNEL\_SEQ' object details with the same properties and values as the previous sequence. The copyright notice at the bottom right is also present.

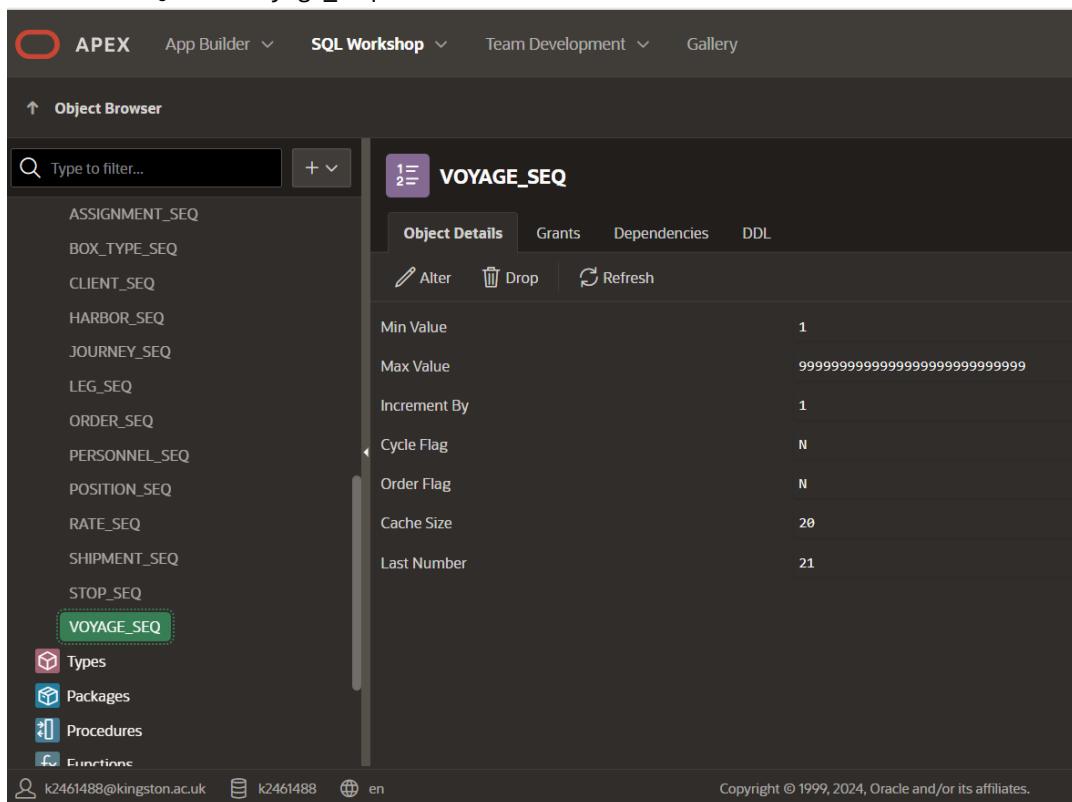
```
8.CREATE SEQUENCE assignment_seq START WITH 1 INCREMENT BY 1
```



The screenshot shows the Oracle Database SQL Workshop interface. In the Object Browser on the left, under the Sequences section, 'ASSIGNMENT\_SEQ' is selected. On the right, the 'Object Details' tab is active, displaying the following configuration for the sequence:

Setting	Value
Min Value	1
Max Value	99
Increment By	1
Cycle Flag	N
Order Flag	N
Cache Size	20
Last Number	21

```
9.CREATE SEQUENCE voyage_seq START WITH 1 INCREMENT BY 1
```



The screenshot shows the Oracle Database SQL Workshop interface. In the Object Browser on the left, under the Sequences section, 'VOYAGE\_SEQ' is selected. On the right, the 'Object Details' tab is active, displaying the following configuration for the sequence:

Setting	Value
Min Value	1
Max Value	99999999999999999999999999999999
Increment By	1
Cycle Flag	N
Order Flag	N
Cache Size	20
Last Number	21

```
10.CREATE SEQUENCE stop_seq START WITH 1 INCREMENT BY 1
```

The screenshot shows the Oracle APEX interface with the SQL Workshop tab selected. In the Object Browser on the left, a sequence named 'STOP\_SEQ' is highlighted. The main panel displays the 'Object Details' for 'STOP\_SEQ' with the following configuration:

Setting	Value
Min Value	1
Max Value	99999999999999999999999999999999
Increment By	1
Cycle Flag	N
Order Flag	N
Cache Size	20
Last Number	21

Below the configuration table, there are buttons for Alter, Drop, and Refresh. The status bar at the bottom shows the user's email (k2461488@kingston.ac.uk), session ID (k2461488), and language (en).

```
11.CREATE SEQUENCE order_seq START WITH 1 INCREMENT BY 1
```

The screenshot shows the Oracle APEX interface with the SQL Workshop module selected. In the Object Browser on the left, several sequence names are listed, including ORDER\_SEQ, which is highlighted with a green dashed border. The main panel displays the details for ORDER\_SEQ, including its current value (1), maximum value (99999999999999999999999999999999), increment by (1), and other properties like Cache Size (20) and Last Number (21). The interface includes standard navigation buttons like Alter, Drop, and Refresh.

12. CREATE SEQUENCE shipment\_seq START WITH 1 INCREMENT BY 1

This screenshot is similar to the previous one but shows the creation of a new sequence named SHIPMENT\_SEQ. The sequence details are identical to ORDER\_SEQ, with a Min Value of 1, a Max Value of 99999999999999999999999999999999, an Increment By of 1, and a Last Number of 21. The sequence name SHIPMENT\_SEQ is highlighted in the Object Browser.

### 13.CREATE SEQUENCE rate\_seq START WITH 1 INCREMENT BY 1

The screenshot shows the Oracle Object Browser interface. On the left, a sidebar lists various sequences: ASSIGNMENT\_SEQ, BOX\_TYPE\_SEQ, CLIENT\_SEQ, HARBOR\_SEQ, JOURNEY\_SEQ, LEG\_SEQ, ORDER\_SEQ, PERSONNEL\_SEQ, POSITION\_SEQ, RATE\_SEQ (which is highlighted in green), SHIPMENT\_SEQ, STOP\_SEQ, and VOYAGE\_SEQ. Below this sidebar are icons for Types, Packages, Procedures, and Functions. The main panel displays the details for the selected sequence RATE\_SEQ. The 'Object Details' tab is active, showing the following properties:

Min Value	1
Max Value	99999999999999999999999999999999999999
Increment By	1
Cycle Flag	N
Order Flag	N
Cache Size	20
Last Number	21

At the bottom of the main panel, there are buttons for Alter, Drop, and Refresh. The status bar at the bottom of the browser window shows the user k2461488@kingston.ac.uk, session k2461488, and language en.

Below are the sequences created in Oracle

The screenshot shows the Oracle SQL Workshop interface with the 'Results' tab selected. The results table displays the execution of 29 CREATE SEQUENCE statements. The columns are: Number, Elapsed, Statement, Feedback, and Rows.

Number	Elapsed	Statement	Feedback	Rows
16	0.05	CREATE TABLE shipments ( shipment_id NUMBER PRIMARY KEY)	Table created.	0
17	0.02	CREATE SEQUENCE harbor_seq START WITH 1 INCREMENT BY 1	Sequence created.	0
18	0.01	CREATE SEQUENCE box_type_seq START WITH 1 INCREMENT BY 1	Sequence created.	0
19	0.01	CREATE SEQUENCE client_seq START WITH 1 INCREMENT BY 1	Sequence created.	0
20	0.01	CREATE SEQUENCE journey_seq START WITH 1 INCREMENT BY 1	Sequence created.	0
21	0.01	CREATE SEQUENCE leg_seq START WITH 1 INCREMENT BY 1	Sequence created.	0
22	0.01	CREATE SEQUENCE position_seq START WITH 1 INCREMENT BY 1	Sequence created.	0
23	0.01	CREATE SEQUENCE personnel_seq START WITH 1 INCREMENT BY 1	Sequence created.	0
24	0.00	CREATE SEQUENCE assignment_seq START WITH 1 INCREMENT BY 1	Sequence created.	0
25	0.00	CREATE SEQUENCE voyage_seq START WITH 1 INCREMENT BY 1	Sequence created.	0
26	0.00	CREATE SEQUENCE stop_seq START WITH 1 INCREMENT BY 1	Sequence created.	0
27	0.05	CREATE SEQUENCE order_seq START WITH 1 INCREMENT BY 1	Sequence created.	0
28	0.01	CREATE SEQUENCE shipment_seq START WITH 1 INCREMENT BY 1	Sequence created.	0
29	0.01	CREATE SEQUENCE rate_seq START WITH 1 INCREMENT BY 1	Sequence created.	0

Img: Sequences created in Oracle

### **3 Chapter 3: Multiplicity of the Relationships**

#### **3.1 Insert Queries for Harbors Table:**

##### **Query 1:**

```
INSERT INTO harbors (harbor_id, harbor_name, nation, location_address, contact_phone, contact_email)
VALUES (harbor_seq.NEXTVAL, 'Liverpool', 'United Kingdom', '21 Maritime Way, Liverpool', '+44-151-987-6543', 'liverpool@marinelog.com')
```

##### **Query 2:**

```
INSERT INTO harbors (harbor_id, harbor_name, nation, location_address, contact_phone, contact_email)
VALUES (harbor_seq.NEXTVAL, 'Bremen', 'Germany', '84 Hafenstrasse, Bremen', '+49-421-765-4321', 'bremen@marinelog.com')
```

##### **Query 3:**

```
INSERT INTO harbors (harbor_id, harbor_name, nation, location_address, contact_phone, contact_email)
VALUES (harbor_seq.NEXTVAL, 'Tampa', 'USA', '143 Gulf Coast Boulevard, Tampa, FL', '+1-813-555-9876', 'tampa@marinelog.com')
```

##### **Query 4:**

```
INSERT INTO harbors (harbor_id, harbor_name, nation, location_address, contact_phone, contact_email)
VALUES (harbor_seq.NEXTVAL, 'Qingdao', 'China', '57 Seaport Avenue, Qingdao', '+86-532-8765-4321', 'qingdao@marinelog.com')
```

##### **Query 5:**

```
INSERT INTO harbors (harbor_id, harbor_name, nation, location_address, contact_phone, contact_email)
VALUES (harbor_seq.NEXTVAL, 'Mombasa', 'Kenya', '92 Harbor Street, Mombasa', '+254-41-234-5678', 'mombasa@marinelog.com')
```

##### **Query 6:**

```
INSERT INTO harbors (harbor_id, harbor_name, nation, location_address, contact_phone, contact_email)
VALUES (harbor_seq.NEXTVAL, 'Mumbai', 'India', '165 Dockyard Road, Mumbai', '+91-22-876-5432', 'mumbai@marinelog.com')
```

##### **Query 7:**

```
INSERT INTO harbors (harbor_id, harbor_name, nation, location_address, contact_phone, contact_email)
VALUES (harbor_seq.NEXTVAL, 'Sydney', 'Australia', '38 Pacific Drive, Sydney', '+61-2-9876-5432', 'sydney@marinelog.com')
```

##### **Query 8:**

```
INSERT INTO harbors (harbor_id, harbor_name, nation, location_address, contact_phone, contact_email)
VALUES (harbor_seq.NEXTVAL, 'Savannah', 'USA', '275 Atlantic Avenue, Savannah, GA', '+1-912-555-7890', 'savannah@marinelog.com')
```

##### **Query 9:**

```
INSERT INTO harbors (harbor_id, harbor_name, nation, location_address, contact_phone, contact_email)
VALUES (harbor_seq.NEXTVAL, 'Boston', 'USA', '493 Harbor Point, Boston, MA', '+1-617-555-3456', 'boston@marinelog.com')
```

##### **Query 10:**

```
INSERT INTO harbors (harbor_id, harbor_name, nation, location_address, contact_phone, contact_email)
```

VALUES (harbor\_seq.NEXTVAL, 'Rotterdam', 'Netherlands', '37 Europoort Lane, Rotterdam', '+31-10-987-6543', 'rotterdam@marinelog.com')

## Insert Queries for Harbors Table in Apex:

```

APEX SQL Workshop Team Development Gallery
SQL Scripts \ Script Editor
Script Name add
Cancel Download Delete Save Create App Run

-- 1. Insert Harbors
1 INSERT INTO harbors (harbor_id, harbor_name, nation, location_address, contact_phone, contact_email)
2 VALUES (harbor_seq.NEXTVAL, 'Liverpool', 'United Kingdom', '21 Maritime Way, Liverpool', '+44-151-987-6543', 'liverpool@marinelog.com')
3 /
4
5
6 INSERT INTO harbors (harbor_id, harbor_name, nation, location_address, contact_phone, contact_email)
7 VALUES (harbor_seq.NEXTVAL, 'Bremen', 'Germany', '84 Hafenstrasse, Bremen', '+49-421-765-4321', 'bremen@marinelog.com')
8 /
9
10 INSERT INTO harbors (harbor_id, harbor_name, nation, location_address, contact_phone, contact_email)
11 VALUES (harbor_seq.NEXTVAL, 'Tampa', 'USA', '143 Gulf Coast Boulevard, Tampa, FL', '+1-813-555-9876', 'tampa@marinelog.com')
12 /
13
14 INSERT INTO harbors (harbor_id, harbor_name, nation, location_address, contact_phone, contact_email)
15 VALUES (harbor_seq.NEXTVAL, 'Qingdao', 'China', '57 Seaport Avenue, Qingdao', '+86-532-8765-4321', 'qingdao@marinelog.com')
16 /
17
18 INSERT INTO harbors (harbor_id, harbor_name, nation, location_address, contact_phone, contact_email)
19 VALUES (harbor_seq.NEXTVAL, 'Mombasa', 'Kenya', '92 Harbor Street, Mombasa', '+254-41-234-5678', 'mombasa@marinelog.com')
20 /
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35

```

```

APEX SQL Workshop Team Development Gallery
SQL Scripts \ Script Editor
Script Name add
Cancel Download Delete Save Create App Run

16 /
17
18 INSERT INTO harbors (harbor_id, harbor_name, nation, location_address, contact_phone, contact_email)
19 VALUES (harbor_seq.NEXTVAL, 'Mombasa', 'Kenya', '92 Harbor Street, Mombasa', '+254-41-234-5678', 'mombasa@marinelog.com')
20 /
21
22 INSERT INTO harbors (harbor_id, harbor_name, nation, location_address, contact_phone, contact_email)
23 VALUES (harbor_seq.NEXTVAL, 'Mumbai', 'India', '165 Dockyard Road, Mumbai', '+91-22-876-5432', 'mumbai@marinelog.com')
24 /
25
26 INSERT INTO harbors (harbor_id, harbor_name, nation, location_address, contact_phone, contact_email)
27 VALUES (harbor_seq.NEXTVAL, 'Sydney', 'Australia', '38 Pacific Drive, Sydney', '+61-2-9876-5432', 'sydney@marinelog.com')
28 /
29
30 INSERT INTO harbors (harbor_id, harbor_name, nation, location_address, contact_phone, contact_email)
31 VALUES (harbor_seq.NEXTVAL, 'Savannah', 'USA', '275 Atlantic Avenue, Savannah, GA', '+1-912-555-7890', 'savannah@marinelog.com')
32 /
33
34 INSERT INTO harbors (harbor_id, harbor_name, nation, location_address, contact_phone, contact_email)
35 VALUES (harbor_seq.NEXTVAL, 'Boston', 'USA', '493 Harbor Point, Boston, MA', '+1-617-555-3456', 'boston@marinelog.com')
36 /

```

## Inserted Data Reflected in Harbors Table:

HARBOR_ID	HARBOR_NAME	NATION	LOCATION_ADDRESS	CONTACT_PHONE	CONTACT_EMAIL
1	Liverpool	United Kingdom	21 Maritime Way, Liverpool	+44-151-987-6543	liverpool@marinelog.com
2	Bremen	Germany	84 Hafenstrasse, Bremen	+49-421-765-4321	bremen@marinelog.com
3	Tampa	USA	143 Gulf Coast Boulevard, Tampa, FL	+1-813-555-9876	tampa@marinelog.com
4	Qingdao	China	57 Seaport Avenue, Qingdao	+86-532-8765-4321	qingdao@marinelog.com
5	Mombasa	Kenya	92 Harbor Street, Mombasa	+254-41-234-5678	mombasa@marinelog.com
6	Mumbai	India	165 Dockyard Road, Mumbai	+91-22-876-5432	mumbai@marinelog.com
7	Sydney	Australia	38 Pacific Drive, Sydney	+61-2-9876-5432	sydney@marinelog.com
8	Savannah	USA	275 Atlantic Avenue, Savannah, GA	+1-912-555-7890	savannah@marinelog.com
9	Boston	USA	493 Harbor Point, Boston, MA	+1-617-555-3456	boston@marinelog.com
10	Rotterdam	Netherlands	37 Europoort Lane, Rotterdam	+31-10-987-6543	rotterdam@marinelog.com

### **3.2 Insert Queries for Ship Categories Table:**

#### **Query 1:**

```
INSERT INTO ship_categories (category_id, category_name, max_length_meters, max_containers, fleet_count)
VALUES ('P', '80xxxxx', 385.45, 22500, 9)
```

#### **Query 2:**

```
INSERT INTO ship_categories (category_id, category_name, max_length_meters, max_containers, fleet_count)
VALUES ('M', '81xxxxx', 382.60, 18750, 12)
```

#### **Query 3:**

```
INSERT INTO ship_categories (category_id, category_name, max_length_meters, max_containers, fleet_count)
VALUES ('N', '82xxxxx', 366.80, 14200, 15)
```

#### **Query 4:**

```
INSERT INTO ship_categories (category_id, category_name, max_length_meters, max_containers, fleet_count)
VALUES ('R', '83xxxxx', 329.50, 11600, 18)
```

#### **Query 5:**

```
INSERT INTO ship_categories (category_id, category_name, max_length_meters, max_containers, fleet_count)
VALUES ('V', '84xxxxx', 328.70, 8900, 25)
```

### **Insert Queries for Ship Categories Table in Apex:**

```
41
42 -- 2. Insert Ship Categories
43 INSERT INTO ship_categories (category_id, category_name, max_length_meters, max_containers, fleet_count)
44 VALUES ('P', '80xxxxx', 385.45, 22500, 9)
45 /
46
47 INSERT INTO ship_categories (category_id, category_name, max_length_meters, max_containers, fleet_count)
48 VALUES ('M', '81xxxxx', 382.60, 18750, 12)
49 /
50
51 INSERT INTO ship_categories (category_id, category_name, max_length_meters, max_containers, fleet_count)
52 VALUES ('N', '82xxxxx', 366.80, 14200, 15)
53 /
54
55 INSERT INTO ship_categories (category_id, category_name, max_length_meters, max_containers, fleet_count)
56 VALUES ('R', '83xxxxx', 329.50, 11600, 18)
57 /
58
59 INSERT INTO ship_categories (category_id, category_name, max_length_meters, max_containers, fleet_count)
60 VALUES ('V', '84xxxxx', 328.70, 8900, 25)
61 /
62
```

### **Inserted Data Reflected in Ship Categories Table:**

CATEGORY_ID	CATEGORY_NAME	MAX_LENGTH_METERS	MAX_CONTAINERS	FLEET_COUNT
P	80xxxxx	385.45	22500	9
M	81xxxxx	382.6	18750	12
N	82xxxxx	366.8	14200	15
R	83xxxxx	329.5	11600	18
V	84xxxxx	328.7	8900	25

### **3.3 Insert Queries for Cargo Box Types Table:**

#### **Query 1:**

```
INSERT INTO cargo_box_types (box_type_id, box_model, classification, outer_length_m, outer_width_m,
outer_height_m, inner_length_m, inner_width_m, inner_height_m, total_weight_capacity_kg,
empty_weight_kg, payload_capacity_kg)
VALUES (box_type_seq.NEXTVAL, '20ft Standard Box', 'General Cargo', 6.096, 2.438, 2.591, 5.900, 2.350,
2.390, 30500, 2250, 28250)
```

#### **Query 2:**

```
INSERT INTO cargo_box_types (box_type_id, box_model, classification, outer_length_m, outer_width_m,
outer_height_m, inner_length_m, inner_width_m, inner_height_m, total_weight_capacity_kg,
empty_weight_kg, payload_capacity_kg)
VALUES (box_type_seq.NEXTVAL, '40ft Standard Box', 'General Cargo', 12.192, 2.438, 2.591, 12.030, 2.350,
2.390, 32750, 3700, 29050)
```

#### **Query 3:**

```
INSERT INTO cargo_box_types (box_type_id, box_model, classification, outer_length_m, outer_width_m,
outer_height_m, inner_length_m, inner_width_m, inner_height_m, total_weight_capacity_kg,
empty_weight_kg, payload_capacity_kg)
VALUES (box_type_seq.NEXTVAL, '40ft Refrigerated Box', 'Temperature Controlled', 12.192, 2.438, 2.591,
11.580, 2.300, 2.280, 34500, 4750, 29750)
```

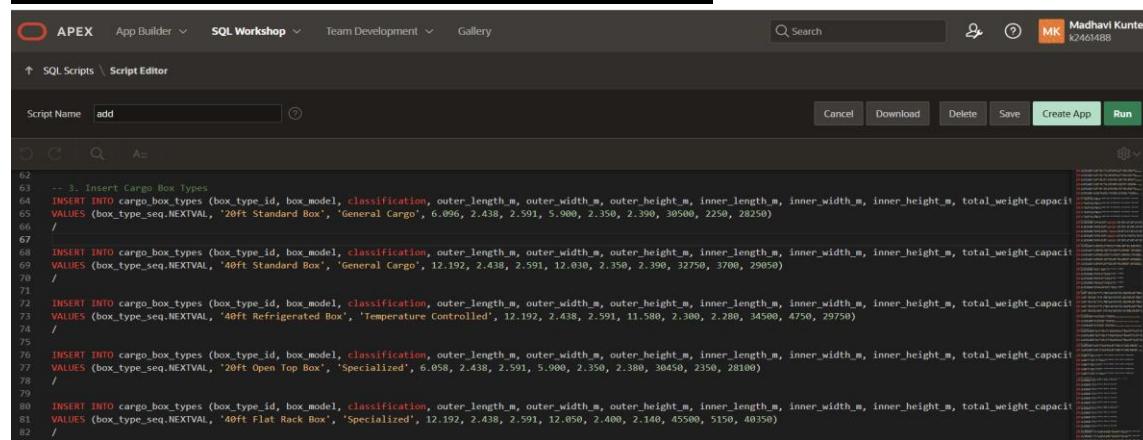
#### **Query 4:**

```
INSERT INTO cargo_box_types (box_type_id, box_model, classification, outer_length_m, outer_width_m,
outer_height_m, inner_length_m, inner_width_m, inner_height_m, total_weight_capacity_kg,
empty_weight_kg, payload_capacity_kg)
VALUES (box_type_seq.NEXTVAL, '20ft Open Top Box', 'Specialized', 6.058, 2.438, 2.591, 5.900, 2.350,
2.380, 30450, 2350, 28100)
```

#### **Query 5:**

```
INSERT INTO cargo_box_types (box_type_id, box_model, classification, outer_length_m, outer_width_m,
outer_height_m, inner_length_m, inner_width_m, inner_height_m, total_weight_capacity_kg,
empty_weight_kg, payload_capacity_kg)
VALUES (box_type_seq.NEXTVAL, '40ft Flat Rack Box', 'Specialized', 12.192, 2.438, 2.591, 12.050, 2.400,
2.140, 45500, 5150, 40350)
```

### **Insert Queries for Cargo Box Types Table in Apex:**



The screenshot shows the Oracle SQL Workshop interface with the following details:

- Header:** APEX, App Builder, SQL Workshop, Team Development, Gallery.
- User Profile:** Madhavi Kunte, ID: k2461488.
- Script Editor:** Script Name: add.
- Code Content:** Five INSERT statements labeled 3, 4, 5, 6, and 7, corresponding to the queries listed above. The code includes column names like box\_type\_id, box\_model, classification, outer\_length\_m, etc., and values from the previous queries.
- Buttons:** Cancel, Download, Delete, Save, Create App, Run.

## **Inserted Data Reflected in Cargo Box Types Table:**

CARGO_BOX_TYPES											
Columns	Data	Indexes	Constraints	Grants	Statistics	Triggers	Dependencies	DDL	Sample Queries		
+ Insert Row	Columns...	Filter...	Count Rows	Load Data	Download	Refresh					
BOX_TYPE_ID	BOX_MODEL	CLASSIFICATION	OUTER_LENGTH_M.	OUTER_WIDTH_M	OUTER_HEIGHT_M...	INNER_LENGTH_M...	INNER_WIDTH_M	INNER_HEIGHT_M	TOTAL_WEIGHT_CAF	EMPTY_WEIGHT_KG	PAYOUT_CAPACITY
1	20ft Standard Box	General Cargo	6.096	2.438	2.591	5.9	2.35	2.39	30500	2250	28250
2	40ft Standard Box	General Cargo	12.192	2.438	2.591	12.03	2.35	2.39	32750	3700	29050
3	40ft Refrigerated...	Temperature Con...	12.192	2.438	2.591	11.58	2.3	2.28	34500	4750	29750
4	20ft Open Top Box	Specialized	6.058	2.438	2.591	5.9	2.35	2.38	30450	2350	28100
5	40ft Flat Rack Box	Specialized	12.192	2.438	2.591	12.05	2.4	2.14	45500	5150	40350

## **3.4. Insert Queries for Ship Categories Table:**

### **Query 1:**

```
INSERT INTO clients (client_id, company_name, representative, phone_number, company_email,
business_address)
VALUES (client_seq.NEXTVAL, 'WorldTech Solutions', 'David Chen', '+1-555-987-6543',
'david.chen@worldtech.com', '458 Innovation Drive, Seattle, WA, USA')
```

### **Query 2:**

```
INSERT INTO clients (client_id, company_name, representative, phone_number, company_email,
business_address)
VALUES (client_seq.NEXTVAL, 'Continental Motors GmbH', 'Anna Schmidt', '+49-555-123-4567',
'anna.schmidt@continentalmotors.de', '72 Industrieweg, Stuttgart, Germany')
```

### **Query 3:**

```
INSERT INTO clients (client_id, company_name, representative, phone_number, company_email,
business_address)
VALUES (client_seq.NEXTVAL, 'Pacific Manufacturing Ltd.', 'Hiro Tanaka', '+81-555-876-5432',
'hiro.tanaka@pacificmfg.com', '319 Industry Street, Osaka, Japan')
```

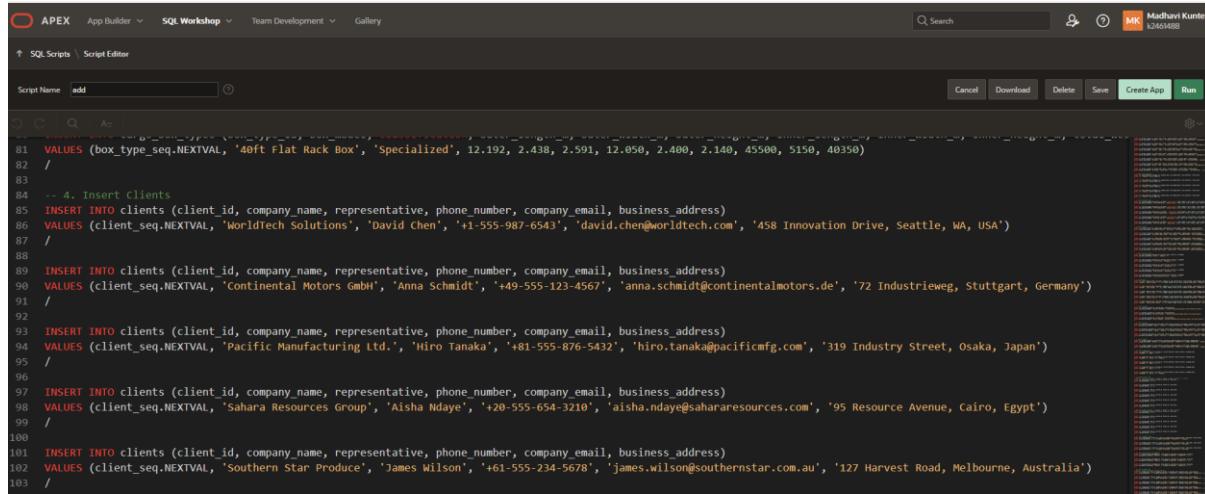
### **Query 4:**

```
INSERT INTO clients (client_id, company_name, representative, phone_number, company_email,
business_address)
VALUES (client_seq.NEXTVAL, 'Sahara Resources Group', 'Aisha Ndaye', '+20-555-654-3210',
'aisha.ndaye@sahararesources.com', '95 Resource Avenue, Cairo, Egypt')
```

### **Query 5:**

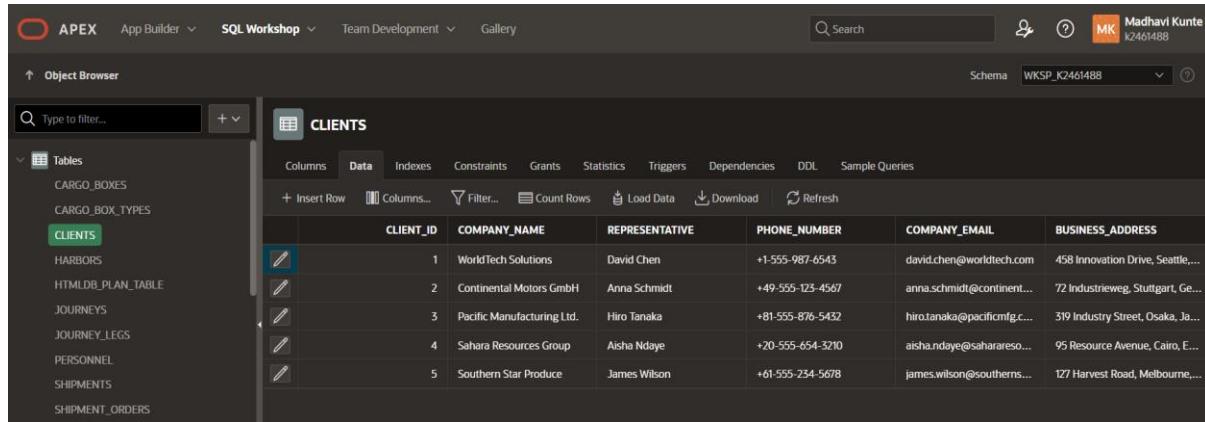
```
INSERT INTO clients (client_id, company_name, representative, phone_number, company_email,
business_address)
VALUES (client_seq.NEXTVAL, 'Southern Star Produce', 'James Wilson', '+61-555-234-5678',
'james.wilson@southernstar.com.au', '127 Harvest Road, Melbourne, Australia')
```

## Insert Queries for Clients Table in Apex:



```
81 VALUES (box_type_seq.NEXTVAL, '40ft Flat Rack Box', 'Specialized', 12.192, 2.438, 2.591, 12.050, 2.400, 2.140, 45500, 5150, 40350)
82 /
83
84 -- 4. Insert Clients
85 INSERT INTO clients (client_id, company_name, representative, phone_number, company_email, business_address)
86 VALUES (client_seq.NEXTVAL, 'WorldTech Solutions', 'David Chen', '+1-555-987-6543', 'david.chen@worldtech.com', '458 Innovation Drive, Seattle, WA, USA')
87 /
88
89 INSERT INTO clients (client_id, company_name, representative, phone_number, company_email, business_address)
90 VALUES (client_seq.NEXTVAL, 'Continental Motors GmbH', 'Anna Schmidt', '+49-555-123-4567', 'anna.schmidt@continentalmotors.de', '72 Industrieweg, Stuttgart, Germany')
91 /
92
93 INSERT INTO clients (client_id, company_name, representative, phone_number, company_email, business_address)
94 VALUES (client_seq.NEXTVAL, 'Pacific Manufacturing Ltd.', 'Hiro Tanaka', '+81-555-876-5432', 'hiro.tanaka@pacificmfg.com', '319 Industry Street, Osaka, Japan')
95 /
96
97 INSERT INTO clients (client_id, company_name, representative, phone_number, company_email, business_address)
98 VALUES (client_seq.NEXTVAL, 'Sahara Resources Group', 'Aisha Ndaye', '+20-555-654-3210', 'aisha.ndaye@sahararesources.com', '95 Resource Avenue, Cairo, Egypt')
99 /
100
101 INSERT INTO clients (client_id, company_name, representative, phone_number, company_email, business_address)
102 VALUES (client_seq.NEXTVAL, 'Southern Star Produce', 'James Wilson', '+61-555-234-5678', 'james.wilson@southernstar.com.au', '127 Harvest Road, Melbourne, Australia')
103 /
```

## Inserted Data Reflected in Clients Table:



CLIENT_ID	COMPANY_NAME	REPRESENTATIVE	PHONE_NUMBER	COMPANY_EMAIL	BUSINESS_ADDRESS
1	WorldTech Solutions	David Chen	+1-555-987-6543	david.chen@worldtech.com	458 Innovation Drive, Seattle, WA, USA
2	Continental Motors GmbH	Anna Schmidt	+49-555-123-4567	anna.schmidt@continentalmotors.de	72 Industrieweg, Stuttgart, Germany
3	Pacific Manufacturing Ltd.	Hiro Tanaka	+81-555-876-5432	hiro.tanaka@pacificmfg.com	319 Industry Street, Osaka, Japan
4	Sahara Resources Group	Aisha Ndaye	+20-555-654-3210	aisha.ndaye@sahararesources.com	95 Resource Avenue, Cairo, Egypt
5	Southern Star Produce	James Wilson	+61-555-234-5678	james.wilson@southernstar.com.au	127 Harvest Road, Melbourne, Australia

## 3.5 Insert Queries for Staff Positions Table:

### **Query 1:**

```
INSERT INTO staff_positions (position_id, position_title, division, is_command)
VALUES (position_seq.NEXTVAL, 'Master', 'Command', 'Y')
```

### **Query 2:**

```
INSERT INTO staff_positions (position_id, position_title, division, is_command)
VALUES (position_seq.NEXTVAL, 'First Officer', 'Navigation', 'Y')
```

### **Query 3:**

```
INSERT INTO staff_positions (position_id, position_title, division, is_command)
VALUES (position_seq.NEXTVAL, 'Second Officer', 'Navigation', 'Y')
```

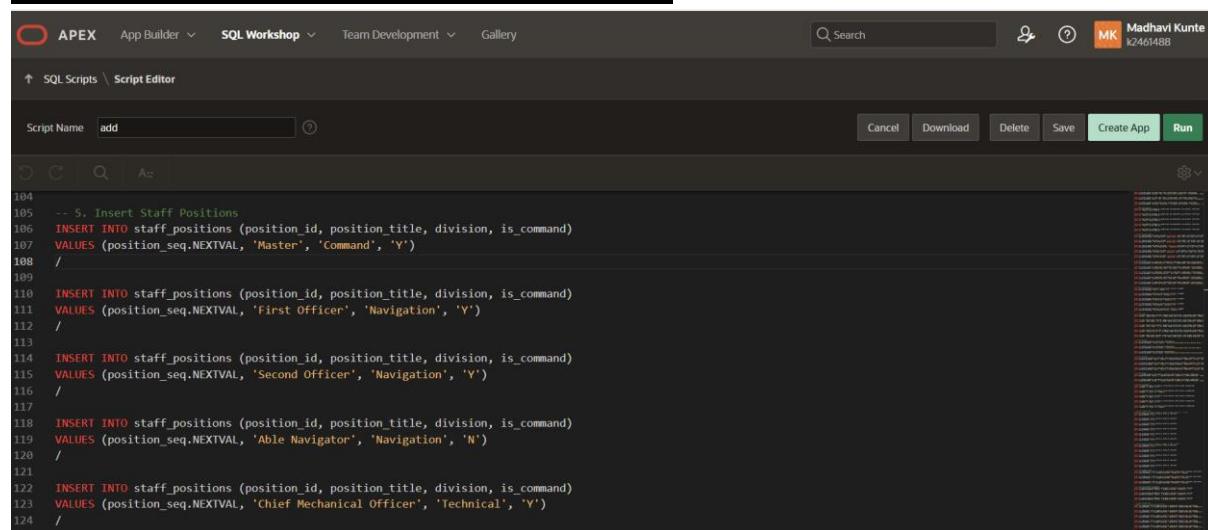
### **Query 4:**

```
INSERT INTO staff_positions (position_id, position_title, division, is_command)
VALUES (position_seq.NEXTVAL, 'Able Navigator', 'Navigation', 'N')
```

### **Query 5:**

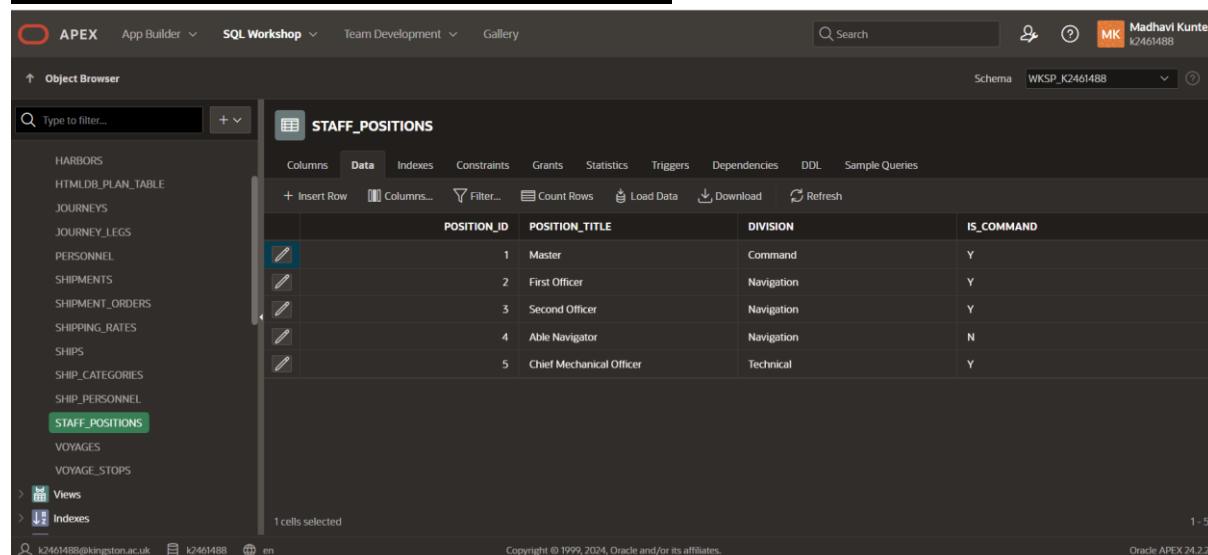
```
INSERT INTO staff_positions (position_id, position_title, division, is_command)
VALUES (position_seq.NEXTVAL, 'Chief Mechanical Officer', 'Technical', 'Y')
```

## Insert Queries for Staff Positions Table in Apex:



```
104 -- S. Insert Staff Positions
105 INSERT INTO staff_positions (position_id, position_title, division, is_command)
106 VALUES (position_seq.NEXTVAL, 'Master', 'Command', 'Y')
107 /
108
109 INSERT INTO staff_positions (position_id, position_title, division, is_command)
110 VALUES (position_seq.NEXTVAL, 'First Officer', 'Navigation', 'Y')
111 /
112
113 INSERT INTO staff_positions (position_id, position_title, division, is_command)
114 VALUES (position_seq.NEXTVAL, 'Second Officer', 'Navigation', 'Y')
115 /
116
117 INSERT INTO staff_positions (position_id, position_title, division, is_command)
118 VALUES (position_seq.NEXTVAL, 'Able Navigator', 'Navigation', 'N')
119 /
120
121 INSERT INTO staff_positions (position_id, position_title, division, is_command)
122 VALUES (position_seq.NEXTVAL, 'Chief Mechanical Officer', 'Technical', 'Y')
123 /
124
```

## Inserted Data Reflected in Staff Positions Table:



	POSITION_ID	POSITION_TITLE	DIVISION	IS_COMMAND
	1	Master	Command	Y
	2	First Officer	Navigation	Y
	3	Second Officer	Navigation	Y
	4	Able Navigator	Navigation	N
	5	Chief Mechanical Officer	Technical	Y

### **3.6 Insert Queries for Ship Table:**

#### **Query 1:**

```
INSERT INTO ships (registry_code, ship_name, category_id, max_velocity_knots, total_tonnage,
cargo_tonnage, hull_length_m, hull_width_m, construction_date)
VALUES ('8042761', 'Neptune Pioneer', 'P', 24, 232500, 258700, 385.45, 60.20, TO_DATE('15-MAY-2022',
'DD-MON-YYYY'))
```

#### **Query 2:**

```
INSERT INTO ships (registry_code, ship_name, category_id, max_velocity_knots, total_tonnage,
cargo_tonnage, hull_length_m, hull_width_m, construction_date)
VALUES ('8157482', 'Ocean Voyager', 'M', 24, 217800, 196400, 382.60, 58.50, TO_DATE('23-SEP-2019',
'DD-MON-YYYY'))
```

#### **Query 3:**

```
INSERT INTO ships (registry_code, ship_name, category_id, max_velocity_knots, total_tonnage,
cargo_tonnage, hull_length_m, hull_width_m, construction_date)
VALUES ('8236954', 'Pacific Trader', 'M', 24, 218500, 196800, 382.60, 58.50, TO_DATE('07-FEB-2020', 'DD-
MON-YYYY'))
```

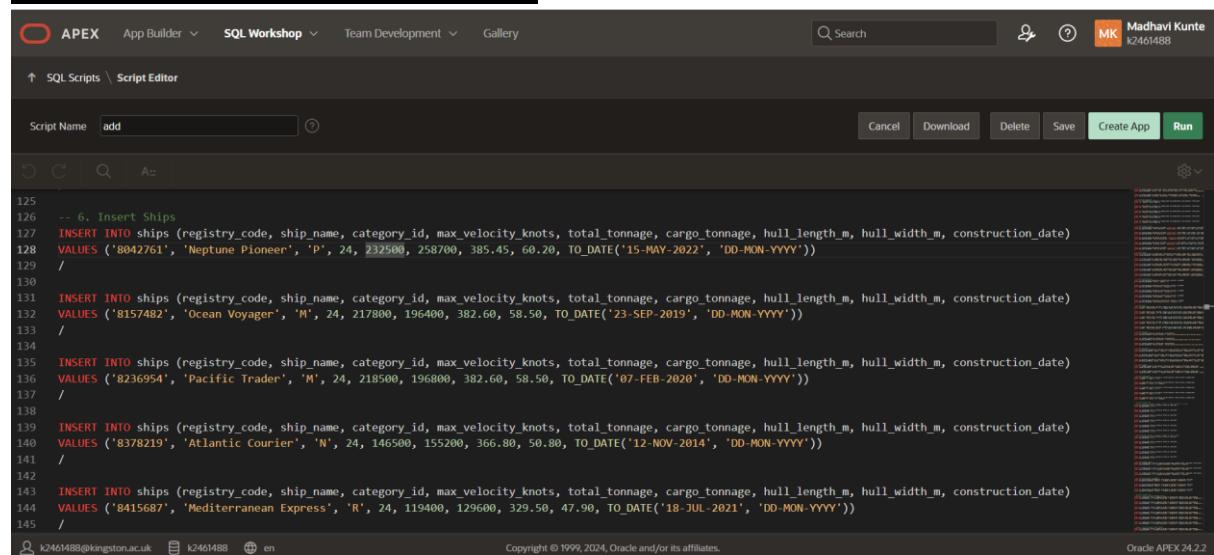
#### **Query 4:**

```
INSERT INTO ships (registry_code, ship_name, category_id, max_velocity_knots, total_tonnage,
cargo_tonnage, hull_length_m, hull_width_m, construction_date)
VALUES ('8378219', 'Atlantic Courier', 'N', 24, 146500, 155200, 366.80, 50.80, TO_DATE('12-NOV-2014',
'DD-MON-YYYY'))
```

#### **Query 5:**

```
INSERT INTO ships (registry_code, ship_name, category_id, max_velocity_knots, total_tonnage,
cargo_tonnage, hull_length_m, hull_width_m, construction_date)
VALUES ('8415687', 'Mediterranean Express', 'R', 24, 119400, 129600, 329.50, 47.90, TO_DATE('18-JUL-
2021', 'DD-MON-YYYY'))
```

### **Insert Queries for Ship Table in Apex:**



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The user's name, MK, and ID, k2461488, are displayed on the right. The main area is titled "Script Editor". The script content is as follows:

```
125
126 -- 6. Insert Ships
127 INSERT INTO ships (registry_code, ship_name, category_id, max_velocity_knots, total_tonnage, cargo_tonnage, hull_length_m, hull_width_m, construction_date)
128 VALUES ('8042761', 'Neptune Pioneer', 'P', 24, 232500, 258700, 385.45, 60.20, TO_DATE('15-MAY-2022', 'DD-MON-YYYY'))
129 /
130
131 INSERT INTO ships (registry_code, ship_name, category_id, max_velocity_knots, total_tonnage, cargo_tonnage, hull_length_m, hull_width_m, construction_date)
132 VALUES ('8157482', 'Ocean Voyager', 'M', 24, 217800, 196400, 382.60, 58.50, TO_DATE('23-SEP-2019', 'DD-MON-YYYY'))
133 /
134
135 INSERT INTO ships (registry_code, ship_name, category_id, max_velocity_knots, total_tonnage, cargo_tonnage, hull_length_m, hull_width_m, construction_date)
136 VALUES ('8236954', 'Pacific Trader', 'M', 24, 218500, 196800, 382.60, 58.50, TO_DATE('07-FEB-2020', 'DD-MON-YYYY'))
137 /
138
139 INSERT INTO ships (registry_code, ship_name, category_id, max_velocity_knots, total_tonnage, cargo_tonnage, hull_length_m, hull_width_m, construction_date)
140 VALUES ('8378219', 'Atlantic Courier', 'N', 24, 146500, 155200, 366.80, 50.80, TO_DATE('12-NOV-2014', 'DD-MON-YYYY'))
141 /
142
143 INSERT INTO ships (registry_code, ship_name, category_id, max_velocity_knots, total_tonnage, cargo_tonnage, hull_length_m, hull_width_m, construction_date)
144 VALUES ('8415687', 'Mediterranean Express', 'R', 24, 119400, 129600, 329.50, 47.90, TO_DATE('18-JUL-2021', 'DD-MON-YYYY'))
145 /
```

The interface includes a search bar, a toolbar with buttons for Cancel, Download, Delete, Save, Create App, and Run, and a sidebar on the right showing a list of other database objects.

## Inserted Data Reflected in Ship Table:

The screenshot shows the Oracle SQL Workshop interface with the 'SHIPS' table selected in the object browser. The table has columns: REGISTRY\_CODE, SHIP\_NAME, CATEGORY\_ID, MAX\_VELOCITY\_K, TOTAL\_TONNAGE, CARGO\_TONNAGE, HULL\_LENGTH\_M, HULL\_WIDTH\_M, and CONSTRUCTION\_DATE. There are five rows of data displayed.

REGISTRY_CODE	SHIP_NAME	CATEGORY_ID	MAX_VELOCITY_K	TOTAL_TONNAGE	CARGO_TONNAGE	HULL_LENGTH_M	HULL_WIDTH_M	CONSTRUCTION_DATE
8042761	Neptune Pioneer	P	24	252500	258700	385.45	60.2	5/15/2022
8157482	Ocean Voyager	M	24	217800	196400	382.6	58.5	9/23/2019
8236954	Pacific Trader	M	24	218500	196800	382.6	58.5	2/7/2020
8378219	Atlantic Courier	N	24	146500	155200	366.8	50.8	11/12/2014
8415687	Mediterranean ...	R	24	119400	129600	329.5	47.9	7/18/2021

## 3.7 Insert Queries for Journeys Table:

### **Query 1:**

```
INSERT INTO journeys (journey_id, journey_code, journey_details)
VALUES (journey_seq.NEXTVAL, 'Atlantic Connector', 'Journey connecting European and North American
harbors')
```

### **Query 2:**

```
INSERT INTO journeys (journey_id, journey_code, journey_details)
VALUES (journey_seq.NEXTVAL, 'Euro-Asia Express', 'Journey connecting European harbors with Asian
harbors')
```

### **Query 3:**

```
INSERT INTO journeys (journey_id, journey_code, journey_details)
VALUES (journey_seq.NEXTVAL, 'Pacific Passage', 'Journey connecting North American harbors with Asian
harbors')
```

## Insert Queries for Journeys Table in Apex:

The screenshot shows the Oracle SQL Workshop script editor with three insert statements for the 'journeys' table:

```
145 /
146
147 -- 7. Insert Journeys
148 INSERT INTO journeys (journey_id, journey_code, journey_details)
149 VALUES (journey_seq.NEXTVAL, 'Atlantic Connector', 'Journey connecting European and North American harbors')
150 /
151
152 INSERT INTO journeys (journey_id, journey_code, journey_details)
153 VALUES (journey_seq.NEXTVAL, 'Euro-Asia Express', 'Journey connecting European harbors with Asian harbors')
154 /
155
156 INSERT INTO journeys (journey_id, journey_code, journey_details)
157 VALUES (journey_seq.NEXTVAL, 'Pacific Passage', 'Journey connecting North American harbors with Asian harbors')
158 /
159
```

## Inserted Data Reflected in Journeys Table:

The screenshot shows the Oracle Database SQL Workshop interface. The left sidebar lists various tables: HARBORS, HTMDB\_PLAN\_TABLE, JOURNEYS (which is selected and highlighted in green), JOURNEY\_LEGS, PERSONNEL, SHIPMENTS, SHIPMENT\_ORDERS, and SHIPPING\_RATES. The main panel displays the JOURNEYS table with the following data:

	JOURNEY_ID	JOURNEY_CODE	JOURNEY_DETAILS
	1	Atlantic Connector	Journey connecting European and North American harbors
	2	Euro-Asia Express	Journey connecting European harbors with Asian harbors
	3	Pacific Passage	Journey connecting North American harbors with Asian h...

## 3.8 Insert Queries for Personnel Table:

### **Query 1:**

```
INSERT INTO personnel (personnel_id, given_name, family_name, position_id, birthdate, citizenship, license_number, license_expiry, mobile_number, personal_email)
VALUES (personnel_seq.NEXTVAL, 'Robert', 'Anderson', 1, TO_DATE('28-APR-1976', 'DD-MON-YYYY'), 'British', 'MST98765', TO_DATE('30-NOV-2026', 'DD-MON-YYYY'), '+44-7765-432198', 'robert.anderson@marinelog.com')
```

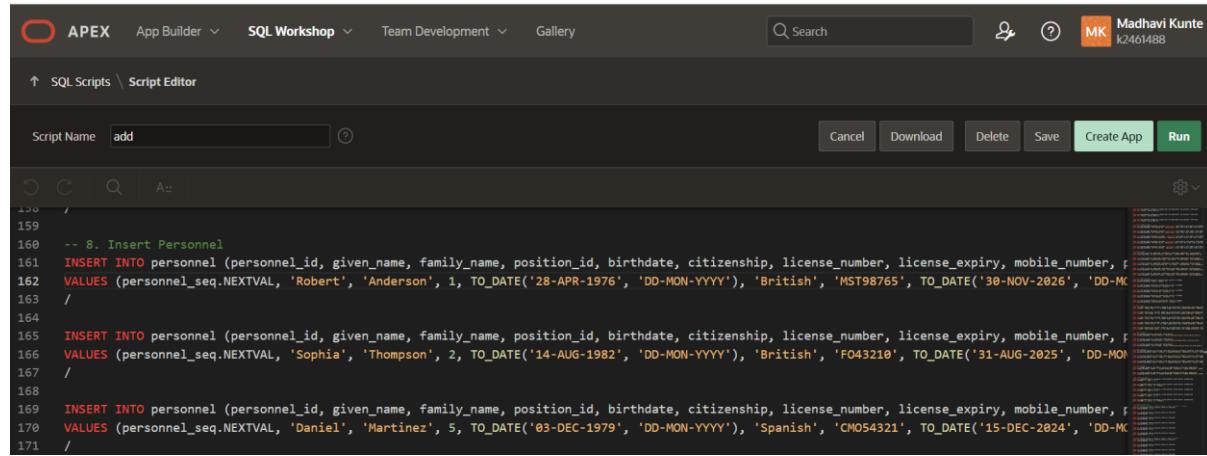
### **Query 2:**

```
INSERT INTO personnel (personnel_id, given_name, family_name, position_id, birthdate, citizenship, license_number, license_expiry, mobile_number, personal_email)
VALUES (personnel_seq.NEXTVAL, 'Sophia', 'Thompson', 2, TO_DATE('14-AUG-1982', 'DD-MON-YYYY'), 'British', 'FO43210', TO_DATE('31-AUG-2025', 'DD-MON-YYYY'), '+44-7912-345678', 'sophia.thompson@marinelog.com')
```

### **Query 3:**

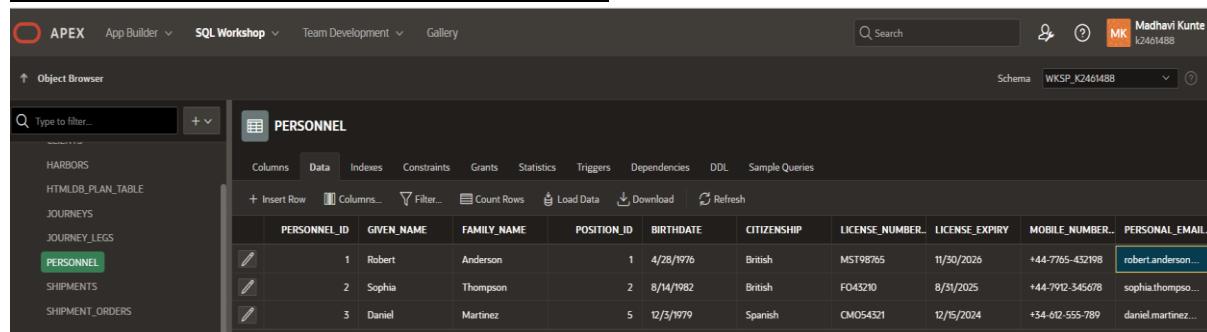
```
INSERT INTO personnel (personnel_id, given_name, family_name, position_id, birthdate, citizenship, license_number, license_expiry, mobile_number, personal_email)
VALUES (personnel_seq.NEXTVAL, 'Daniel', 'Martinez', 5, TO_DATE('03-DEC-1979', 'DD-MON-YYYY'), 'Spanish', 'CMO54321', TO_DATE('15-DEC-2024', 'DD-MON-YYYY'), '+34-612-555-789', 'daniel.martinez@marinelog.com')
```

## Insert Queries for Personnel Table in Apex:



```
159
160 -- 8. Insert Personnel
161 INSERT INTO personnel (personnel_id, given_name, family_name, position_id, birthdate, citizenship, license_number, license_expiry, mobile_number, personal_email)
162 VALUES (personnel_seq.NEXTVAL, 'Robert', 'Anderson', 1, TO_DATE('28-APR-1976', 'DD-MON-YYYY'), 'British', 'MST98765', TO_DATE('30-NOV-2026', 'DD-MON-YYYY'), '+44-7765-452198', 'robert.anderson@company.com')
163 /
164
165 INSERT INTO personnel (personnel_id, given_name, family_name, position_id, birthdate, citizenship, license_number, license_expiry, mobile_number, personal_email)
166 VALUES (personnel_seq.NEXTVAL, 'Sophia', 'Thompson', 2, TO_DATE('14-AUG-1982', 'DD-MON-YYYY'), 'British', 'FO43210', TO_DATE('31-AUG-2025', 'DD-MON-YYYY'), '+44-7912-545678', 'sophia.thompson@company.com')
167 /
168
169 INSERT INTO personnel (personnel_id, given_name, family_name, position_id, birthdate, citizenship, license_number, license_expiry, mobile_number, personal_email)
170 VALUES (personnel_seq.NEXTVAL, 'Daniel', 'Martinez', 5, TO_DATE('03-DEC-1979', 'DD-MON-YYYY'), 'Spanish', 'CM054321', TO_DATE('15-DEC-2024', 'DD-MON-YYYY'), '+34-012-555-789', 'daniel.martinez@company.com')
171 /
```

## Inserted Data Reflected in Personnel Table:



PERSONNEL_ID	GIVEN_NAME	FAMILY_NAME	POSITION_ID	BIRTHDATE	CITIZENSHIP	LICENSE_NUMBER	LICENSE_EXPIRY	MOBILE_NUMBER	PERSONAL_EMAIL
1	Robert	Anderson	1	4/28/1976	British	MST98765	11/30/2026	+44-7765-452198	robert.anderson@company.com
2	Sophia	Thompson	2	8/14/1982	British	FO43210	8/31/2025	+44-7912-545678	sophia.thompson@company.com
5	Daniel	Martinez	5	12/3/1979	Spanish	CM054321	12/15/2024	+34-012-555-789	daniel.martinez@company.com

## 3.9 Insert Queries for Voyages Table:

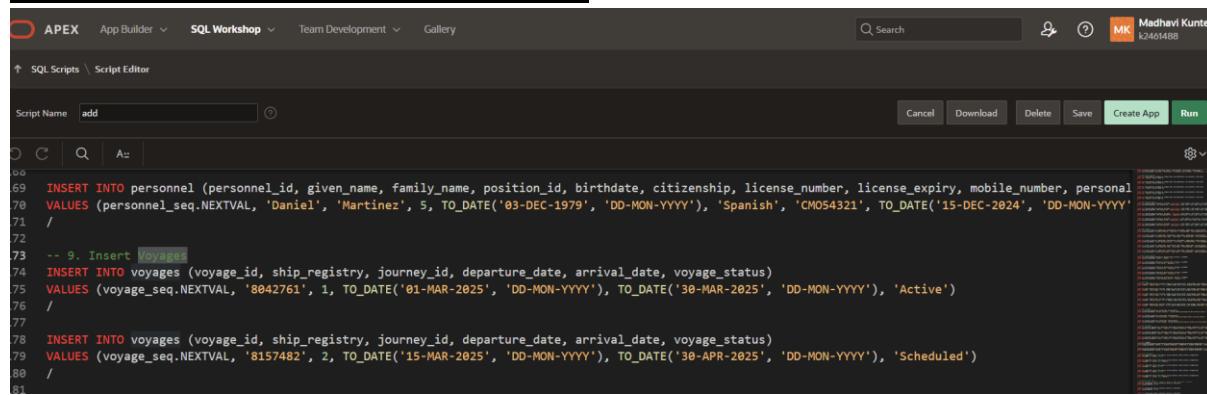
### **Query 1:**

```
INSERT INTO voyages (voyage_id, ship_registry, journey_id, departure_date, arrival_date, voyage_status)
VALUES (voyage_seq.NEXTVAL, '8042761', 1, TO_DATE('01-MAR-2025', 'DD-MON-YYYY'),
TO_DATE('30-MAR-2025', 'DD-MON-YYYY'), 'Active')
```

### **Query 2:**

```
INSERT INTO voyages (voyage_id, ship_registry, journey_id, departure_date, arrival_date, voyage_status)
VALUES (voyage_seq.NEXTVAL, '8157482', 2, TO_DATE('15-MAR-2025', 'DD-MON-YYYY'),
TO_DATE('30-APR-2025', 'DD-MON-YYYY'), 'Scheduled')
```

## Insert Queries for Voyages Table in Apex:



```
66
67 INSERT INTO personnel (personnel_id, given_name, family_name, position_id, birthdate, citizenship, license_number, license_expiry, mobile_number, personal_email)
68 VALUES (personnel_seq.NEXTVAL, 'Daniel', 'Martinez', 5, TO_DATE('03-DEC-1979', 'DD-MON-YYYY'), 'Spanish', 'CM054321', TO_DATE('15-DEC-2024', 'DD-MON-YYYY'), '+34-012-555-789', 'daniel.martinez@company.com')
69 /
70
71 -- 9. Insert Voyages
72 INSERT INTO voyages (voyage_id, ship_registry, journey_id, departure_date, arrival_date, voyage_status)
73 VALUES (voyage_seq.NEXTVAL, '8042761', 1, TO_DATE('01-MAR-2025', 'DD-MON-YYYY'), TO_DATE('30-MAR-2025', 'DD-MON-YYYY'), 'Active')
74 /
75
76 INSERT INTO voyages (voyage_id, ship_registry, journey_id, departure_date, arrival_date, voyage_status)
77 VALUES (voyage_seq.NEXTVAL, '8157482', 2, TO_DATE('15-MAR-2025', 'DD-MON-YYYY'), TO_DATE('30-APR-2025', 'DD-MON-YYYY'), 'Scheduled')
78 /
79
80
```

## **Inserted Data Reflected in Voyages Table:**

The screenshot shows the Oracle APEX Object Browser interface. On the left, there is a sidebar with a search bar and a list of tables under the 'Schemas' section, including HARBORS, HTMLDB\_PLAN\_TABLE, JOURNEYS, JOURNEY\_LEGS, PERSONNEL, SHIPMENTS, SHIPMENT\_ORDERS, SHIPPING\_RATES, SHIPS, SHIP\_CATEGORIES, SHIP\_PERSONNEL, and STAFF\_POSITIONS. The 'VOYAGES' table is selected. The main area displays the 'VOYAGES' table with the following data:

VOYAGE_ID	SHIP_REGISTRY	JOURNEY_ID	DEPARTURE_DATE	ARRIVAL_DATE	VOYAGE_STATUS
1	8042761	1	3/1/2025	3/30/2025	Active
2	8157482	2	3/15/2025	4/30/2025	Scheduled

Below the table, it says '1 cells selected'. The bottom right corner of the interface shows 'Oracle APEX 24.2.2'.

## **3.10 Insert Queries for Cargo Boxes Table:**

### **Query 1:**

```
INSERT INTO cargo_boxes (box_id, box_type_id, current_condition, harbor_location, assigned_ship)
VALUES ('ML250001', 1, 'Ready', 1, NULL)
```

### **Query 2:**

```
INSERT INTO cargo_boxes (box_id, box_type_id, current_condition, harbor_location, assigned_ship)
VALUES ('ML250002', 1, 'Active', NULL, '8042761')
```

### **Query 3:**

```
INSERT INTO cargo_boxes (box_id, box_type_id, current_condition, harbor_location, assigned_ship)
VALUES ('ML250003', 2, 'Ready', 3, NULL)
```

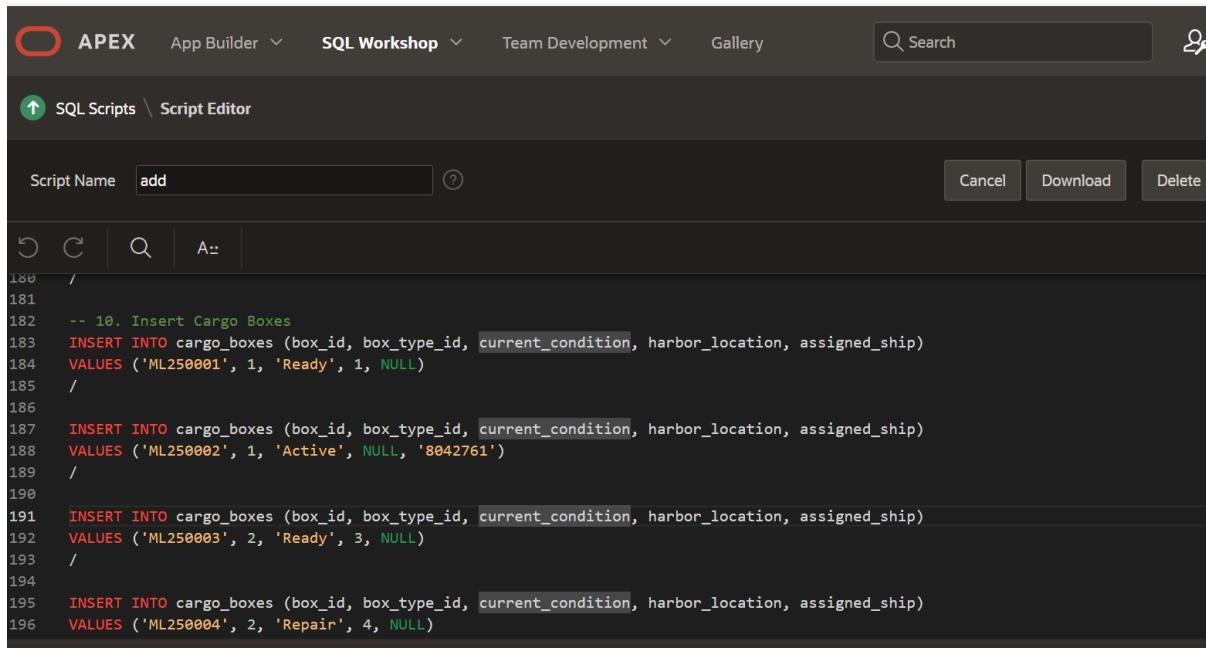
### **Query 4:**

```
INSERT INTO cargo_boxes (box_id, box_type_id, current_condition, harbor_location, assigned_ship)
VALUES ('ML250004', 2, 'Repair', 4, NULL)
```

### **Query 5:**

```
INSERT INTO cargo_boxes (box_id, box_type_id, current_condition, harbor_location, assigned_ship)
VALUES ('ML250005', 3, 'Active', NULL, '8157482')
```

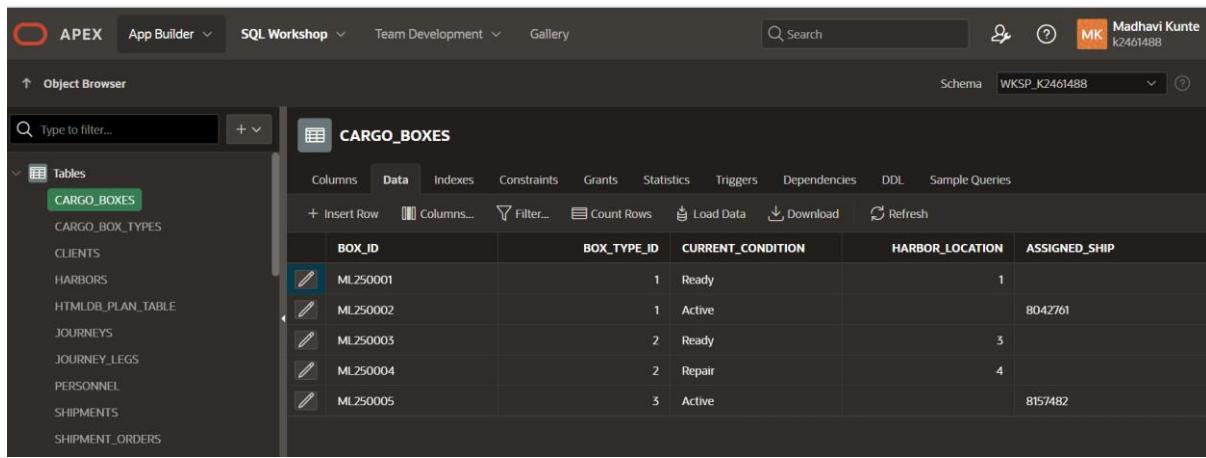
## Insert Queries for Cargo Boxes Table in Apex:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. A search bar and user profile are also present. The main area is titled "SQL Scripts \ Script Editor". A script named "add" is displayed, containing the following SQL code:

```
180 /
181
182 -- 10. Insert Cargo Boxes
183 INSERT INTO cargo_boxes (box_id, box_type_id, current_condition, harbor_location, assigned_ship)
184 VALUES ('ML250001', 1, 'Ready', 1, NULL)
185 /
186
187 INSERT INTO cargo_boxes (box_id, box_type_id, current_condition, harbor_location, assigned_ship)
188 VALUES ('ML250002', 1, 'Active', NULL, '8042761')
189 /
190
191 INSERT INTO cargo_boxes (box_id, box_type_id, current_condition, harbor_location, assigned_ship)
192 VALUES ('ML250003', 2, 'Ready', 3, NULL)
193 /
194
195 INSERT INTO cargo_boxes (box_id, box_type_id, current_condition, harbor_location, assigned_ship)
196 VALUES ('ML250004', 2, 'Repair', 4, NULL)
```

## Inserted Data Reflected in Cargo Boxes Table:



The screenshot shows the Oracle SQL Workshop Object Browser. The left sidebar lists tables: CARGO\_BOXES, CARGO\_BOX\_TYPES, CLIENTS, HARBORS, HTMLDB\_PLAN\_TABLE, JOURNEYS, JOURNEY\_LEGS, PERSONNEL, SHIPMENTS, and SHIPMENT\_ORDERS. The CARGO\_BOXES table is selected and shown in the main pane. The table has columns: BOX\_ID, BOX\_TYPE\_ID, CURRENT\_CONDITION, HARBOR\_LOCATION, and ASSIGNED\_SHIP. The data is as follows:

	BOX_ID	BOX_TYPE_ID	CURRENT_CONDITION	HARBOR_LOCATION	ASSIGNED_SHIP
1	ML250001	1	Ready		
2	ML250002	1	Active		8042761
3	ML250003	2	Ready		
4	ML250004	2	Repair		
5	ML250005	3	Active		8157482

### **3.11-Insert Queries for Journey Legs Table:**

**Atlantic Connector: Liverpool -> Bremen -> Boston -> Tampa -> Savannah -> Liverpool**

#### **Query 1:**

```
INSERT INTO journey_legs (leg_id, journey_id, harbor_id, stop_order)
VALUES (leg_seq.NEXTVAL, 1, 1, 1)
```

#### **Query 2:**

```
INSERT INTO journey_legs (leg_id, journey_id, harbor_id, stop_order)
VALUES (leg_seq.NEXTVAL, 1, 2, 2)
```

#### **Query 3:**

```
INSERT INTO journey_legs (leg_id, journey_id, harbor_id, stop_order)
VALUES (leg_seq.NEXTVAL, 1, 9, 3)
```

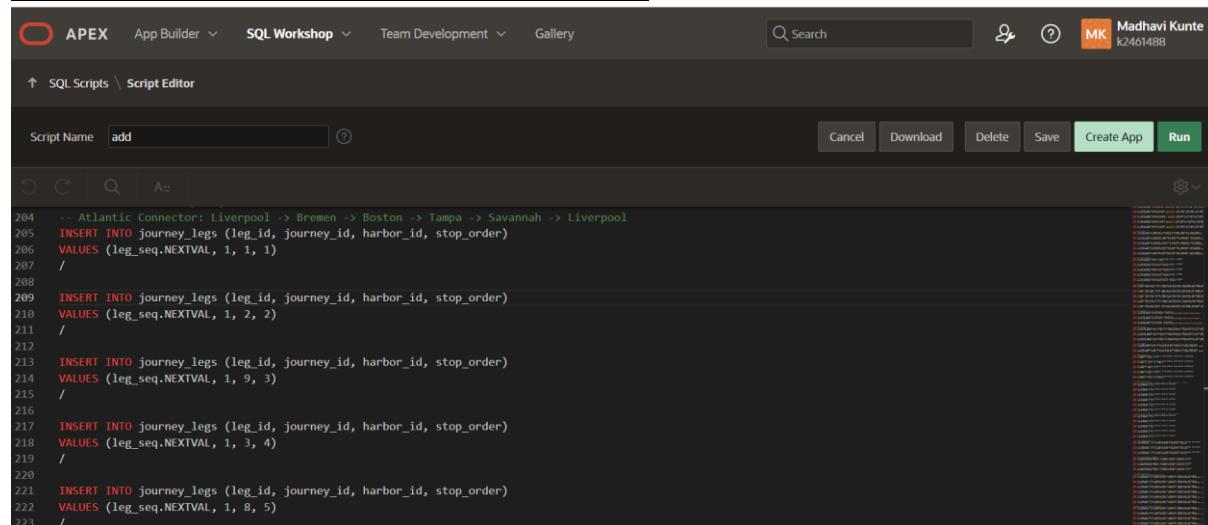
#### **Query 4:**

```
INSERT INTO journey_legs (leg_id, journey_id, harbor_id, stop_order) VALUES (leg_seq.NEXTVAL, 1, 3, 4)
INSERT INTO journey_legs (leg_id, journey_id, harbor_id, stop_order)
VALUES (leg_seq.NEXTVAL, 1, 8, 5)
```

#### **Query 5:**

```
INSERT INTO journey_legs (leg_id, journey_id, harbor_id, stop_order)
VALUES (leg_seq.NEXTVAL, 1, 1, 6)
```

### **Insert Queries for Journey Legs Table in Apex:**



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there are icons for Search, Help, and a user profile (MK k2461488). The main area is titled "SQL Scripts \ Script Editor". A script named "add" is open, containing the following code:

```
204 -- Atlantic Connector: Liverpool -> Bremen -> Boston -> Tampa -> Savannah -> Liverpool
205 INSERT INTO journey_legs (leg_id, journey_id, harbor_id, stop_order)
206 VALUES (leg_seq.NEXTVAL, 1, 1, 1)
207 /
208
209 INSERT INTO journey_legs (leg_id, journey_id, harbor_id, stop_order)
210 VALUES (leg_seq.NEXTVAL, 1, 2, 2)
211 /
212
213 INSERT INTO journey_legs (leg_id, journey_id, harbor_id, stop_order)
214 VALUES (leg_seq.NEXTVAL, 1, 9, 3)
215 /
216
217 INSERT INTO journey_legs (leg_id, journey_id, harbor_id, stop_order)
218 VALUES (leg_seq.NEXTVAL, 1, 3, 4)
219 /
220
221 INSERT INTO journey_legs (leg_id, journey_id, harbor_id, stop_order)
222 VALUES (leg_seq.NEXTVAL, 1, 8, 5)
223 /
```

## Euro-Asia Express: Bremen -> Liverpool -> Qingdao -> Sydney -> Bremen

```
-- Euro-Asia Express: Bremen -> Liverpool -> Qingdao -> Sydney -> Bremen
INSERT INTO journey_legs (leg_id, journey_id, harbor_id, stop_order)
VALUES (leg_seq.NEXTVAL, 1, 1, 1)
/
INSERT INTO journey_legs (leg_id, journey_id, harbor_id, stop_order)
VALUES (leg_seq.NEXTVAL, 1, 2, 2)
/
INSERT INTO journey_legs (leg_id, journey_id, harbor_id, stop_order)
VALUES (leg_seq.NEXTVAL, 2, 1, 2)
/
INSERT INTO journey_legs (leg_id, journey_id, harbor_id, stop_order)
VALUES (leg_seq.NEXTVAL, 2, 4, 3)
/
INSERT INTO journey_legs (leg_id, journey_id, harbor_id, stop_order)
VALUES (leg_seq.NEXTVAL, 2, 7, 4)
/
INSERT INTO journey_legs (leg_id, journey_id, harbor_id, stop_order)
VALUES (leg_seq.NEXTVAL, 2, 2, 5)
```

### Inserted Data Reflected in Journey Legs Table:

LEG_ID	JOURNEY_ID	HARBOR_ID	STOP_ORDER
1	1	1	1
2	1	2	2
3	1	9	3
4	1	3	4
5	1	8	5
6	1	1	6
7	2	2	1
8	2	1	2

### 3.12 Insert Queries for Shipment Orders Table:

#### Query 1:

```
INSERT INTO shipment_orders (order_id, client_id, pickup_harbor, delivery_harbor, order_date, order_status)
VALUES (order_seq.NEXTVAL, 1, 1, 3, TO_DATE('20-FEB-2025', 'DD-MON-YYYY'), 'Shipping')
```

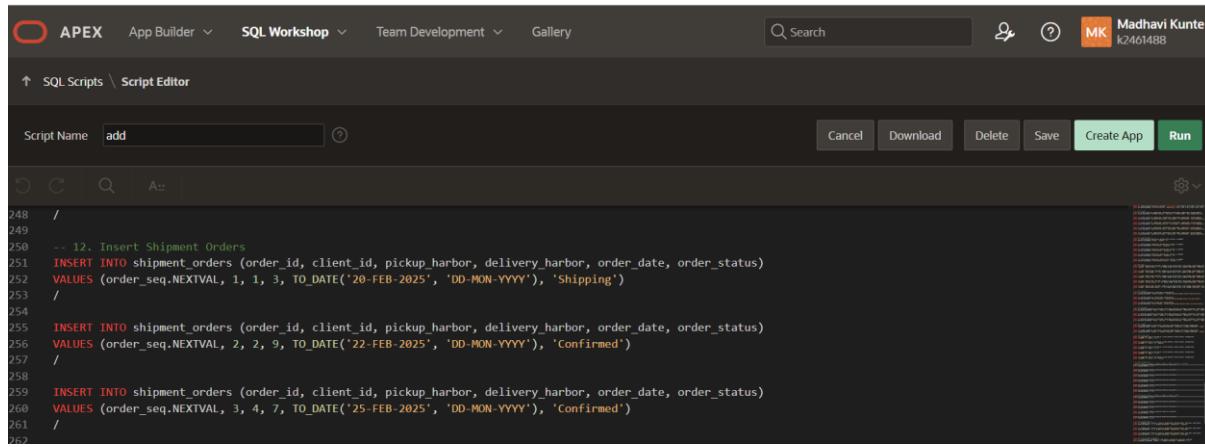
#### Query 2:

```
INSERT INTO shipment_orders (order_id, client_id, pickup_harbor, delivery_harbor, order_date, order_status)
VALUES (order_seq.NEXTVAL, 2, 2, 9, TO_DATE('22-FEB-2025', 'DD-MON-YYYY'), 'Confirmed')
```

#### Query 3:

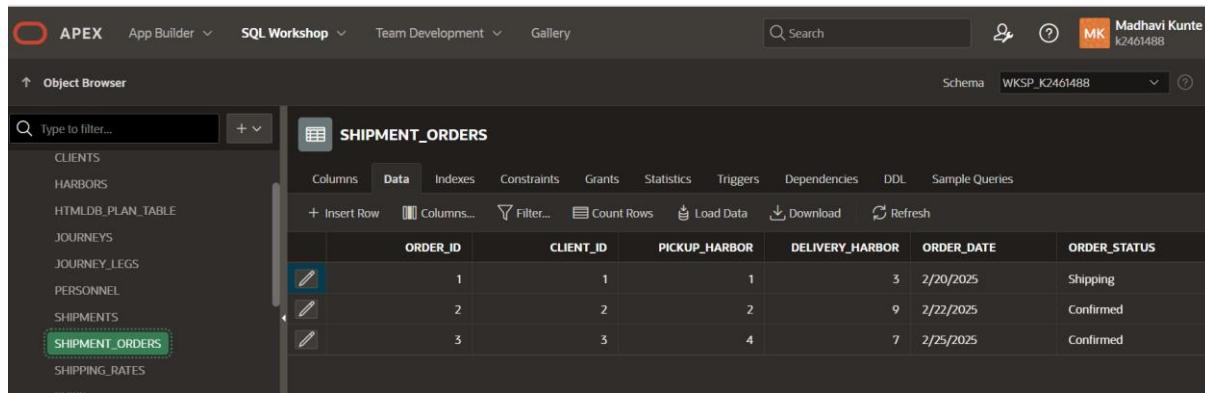
```
INSERT INTO shipment_orders (order_id, client_id, pickup_harbor, delivery_harbor, order_date, order_status)
VALUES (order_seq.NEXTVAL, 3, 4, 7, TO_DATE('25-FEB-2025', 'DD-MON-YYYY'), 'Confirmed')
```

## Insert Queries for Shipment Orders Table in Apex



```
248 /
249
250 -- 12. Insert Shipment Orders
251 INSERT INTO shipment_orders (order_id, client_id, pickup_harbor, delivery_harbor, order_date, order_status)
252 VALUES (order_seq.NEXTVAL, 1, 1, 3, TO_DATE('20-FEB-2025', 'DD-MON-YYYY'), 'Shipping')
253 /
254
255 INSERT INTO shipment_orders (order_id, client_id, pickup_harbor, delivery_harbor, order_date, order_status)
256 VALUES (order_seq.NEXTVAL, 2, 2, 9, TO_DATE('22-FEB-2025', 'DD-MON-YYYY'), 'Confirmed')
257 /
258
259 INSERT INTO shipment_orders (order_id, client_id, pickup_harbor, delivery_harbor, order_date, order_status)
260 VALUES (order_seq.NEXTVAL, 3, 4, 7, TO_DATE('25-FEB-2025', 'DD-MON-YYYY'), 'Confirmed')
261 /
262
```

## Inserted Data Reflected in Shipment Orders Table:



	ORDER_ID	CLIENT_ID	PICKUP_HARBOR	DELIVERY_HARBOR	ORDER_DATE	ORDER_STATUS
	1	1	1		3	2/20/2025
	2	2	2		9	2/22/2025
	3	3	4		7	2/25/2025

## 3.13 Insert Queries for Ship Personnel Assignments Table:

### Query 1:

```
INSERT INTO ship_personnel (assignment_id, ship_registry, personnel_id, duty_start, duty_end)
VALUES (assignment_seq.NEXTVAL, '8042761', 1, TO_DATE('15-JAN-2025', 'DD-MON-YYYY'), NULL)
```

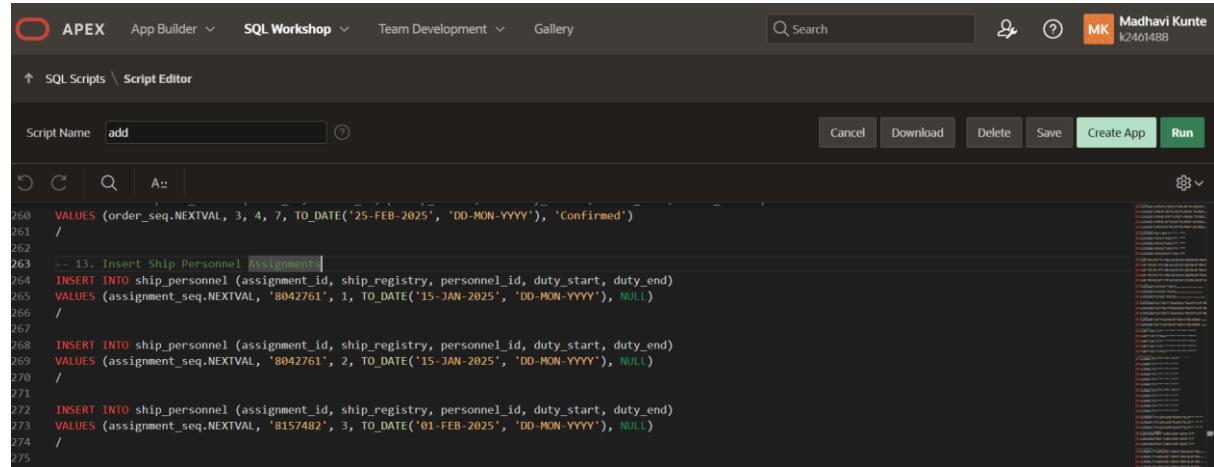
### Query 2:

```
INSERT INTO ship_personnel (assignment_id, ship_registry, personnel_id, duty_start, duty_end)
VALUES (assignment_seq.NEXTVAL, '8042761', 2, TO_DATE('15-JAN-2025', 'DD-MON-YYYY'), NULL)
```

### Query 3:

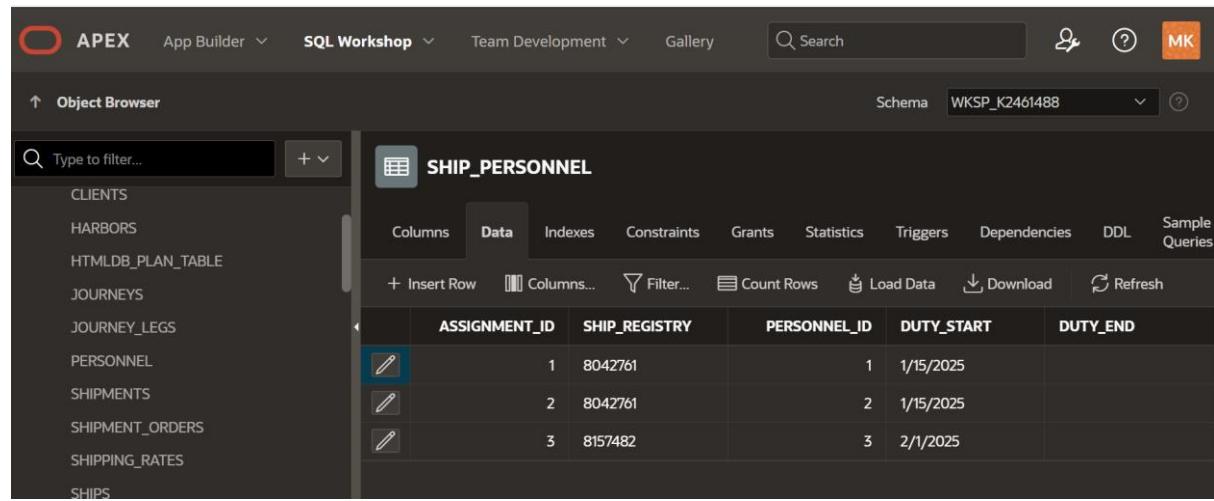
```
INSERT INTO ship_personnel (assignment_id, ship_registry, personnel_id, duty_start, duty_end)
VALUES (assignment_seq.NEXTVAL, '8157482', 3, TO_DATE('01-FEB-2025', 'DD-MON-YYYY'), NULL)
```

## Insert Queries for Ship Personnel Table in Apex:



```
260 VALUES (order_seq.NEXTVAL, 3, 4, 7, TO_DATE('25-FEB-2025', 'DD-MON-YYYY'), 'Confirmed')
261 /
262
263 -- 13. Insert Ship Personnel [assignments]
264 INSERT INTO ship_personnel (assignment_id, ship_registry, personnel_id, duty_start, duty_end)
265 VALUES (assignment_seq.NEXTVAL, '8042761', 1, TO_DATE('15-JAN-2025', 'DD-MON-YYYY'), NULL)
266 /
267
268 INSERT INTO ship_personnel (assignment_id, ship_registry, personnel_id, duty_start, duty_end)
269 VALUES (assignment_seq.NEXTVAL, '8042761', 2, TO_DATE('15-JAN-2025', 'DD-MON-YYYY'), NULL)
270 /
271
272 INSERT INTO ship_personnel (assignment_id, ship_registry, personnel_id, duty_start, duty_end)
273 VALUES (assignment_seq.NEXTVAL, '8157482', 3, TO_DATE('01-FEB-2025', 'DD-MON-YYYY'), NULL)
274 /
275
```

## Inserted Data Reflected in Ship Personnel Table:



	ASSIGNMENT_ID	SHIP_REGISTRY	PERSONNEL_ID	DUTY_START	DUTY_END
1	8042761		1	1/15/2025	
2	8042761		2	1/15/2025	
3	8157482		3	2/1/2025	

### **3.14 Insert Queries for Voyage Stops Table:**

#### **Query 1:**

```
INSERT INTO voyage_stops (stop_id, voyage_id, harbor_id, docking_time, departure_time, stop_sequence)
VALUES (stop_seq.NEXTVAL, 1, 1, TO_DATE('01-MAR-2025', 'DD-MON-YYYY'), TO_DATE('02-MAR-2025', 'DD-MON-YYYY'), 1)
```

#### **Query 2:**

```
INSERT INTO voyage_stops (stop_id, voyage_id, harbor_id, docking_time, departure_time, stop_sequence)
VALUES (stop_seq.NEXTVAL, 1, 2, TO_DATE('04-MAR-2025', 'DD-MON-YYYY'), TO_DATE('05-MAR-2025', 'DD-MON-YYYY'), 2)
```

#### **Query 3:**

```
INSERT INTO voyage_stops (stop_id, voyage_id, harbor_id, docking_time, departure_time, stop_sequence)
VALUES (stop_seq.NEXTVAL, 1, 9, TO_DATE('12-MAR-2025', 'DD-MON-YYYY'), TO_DATE('13-MAR-2025', 'DD-MON-YYYY'), 3)
```

#### **Query 4:**

```
INSERT INTO voyage_stops (stop_id, voyage_id, harbor_id, docking_time, departure_time, stop_sequence)
VALUES (stop_seq.NEXTVAL, 1, 3, TO_DATE('15-MAR-2025', 'DD-MON-YYYY'), TO_DATE('16-MAR-2025', 'DD-MON-YYYY'), 4)
```

#### **Query 5:**

Voyage 2: Euro-Asia Express (Ocean Voyager)

```
INSERT INTO voyage_stops (stop_id, voyage_id, harbor_id, docking_time, departure_time, stop_sequence)
VALUES (stop_seq.NEXTVAL, 2, 2, TO_DATE('15-MAR-2025', 'DD-MON-YYYY'), TO_DATE('16-MAR-2025', 'DD-MON-YYYY'), 1)
```

#### **Query 6:**

```
INSERT INTO voyage_stops (stop_id, voyage_id, harbor_id, docking_time, departure_time, stop_sequence)
VALUES (stop_seq.NEXTVAL, 2, 1, TO_DATE('18-MAR-2025', 'DD-MON-YYYY'), TO_DATE('19-MAR-2025', 'DD-MON-YYYY'), 2)
```

#### **Query 7:**

```
INSERT INTO voyage_stops (stop_id, voyage_id, harbor_id, docking_time, departure_time, stop_sequence)
VALUES (stop_seq.NEXTVAL, 2, 4, TO_DATE('10-APR-2025', 'DD-MON-YYYY'), TO_DATE('12-APR-2025', 'DD-MON-YYYY'), 3)
```

#### **Query 8:**

```
INSERT INTO voyage_stops (stop_id, voyage_id, harbor_id, docking_time, departure_time, stop_sequence)
VALUES (stop_seq.NEXTVAL, 2, 7, TO_DATE('20-APR-2025', 'DD-MON-YYYY'), TO_DATE('22-APR-2025', 'DD-MON-YYYY'), 4)
```

#### **Query 9:**

```
INSERT INTO voyage_stops (stop_id, voyage_id, harbor_id, docking_time, departure_time, stop_sequence)
VALUES (stop_seq.NEXTVAL, 2, 2, TO_DATE('29-APR-2025', 'DD-MON-YYYY'), TO_DATE('30-APR-2025', 'DD-MON-YYYY'), 5)
```

## Insert Queries for Voyage Stops Table in Apex:

### Voyage 1:

```

14. Insert Voyage Stops
-- Voyage 1: Atlantic Connector (Neptune Pioneer)
INSERT INTO voyage_stops (stop_id, voyage_id, harbor_id, docking_time, departure_time, stop_sequence)
VALUES (stop_seq.NEXTVAL, 1, 1, TO_DATE('01-MAR-2025', 'DD-MON-YYYY'), TO_DATE('02-MAR-2025', 'DD-MON-YYYY'), 1)
/
INSERT INTO voyage_stops (stop_id, voyage_id, harbor_id, docking_time, departure_time, stop_sequence)
VALUES (stop_seq.NEXTVAL, 1, 2, TO_DATE('04-MAR-2025', 'DD-MON-YYYY'), TO_DATE('05-MAR-2025', 'DD-MON-YYYY'), 2)
/
INSERT INTO voyage_stops (stop_id, voyage_id, harbor_id, docking_time, departure_time, stop_sequence)
VALUES (stop_seq.NEXTVAL, 1, 9, TO_DATE('12-MAR-2025', 'DD-MON-YYYY'), TO_DATE('13-MAR-2025', 'DD-MON-YYYY'), 3)
/
INSERT INTO voyage_stops (stop_id, voyage_id, harbor_id, docking_time, departure_time, stop_sequence)
VALUES (stop_seq.NEXTVAL, 1, 3, TO_DATE('15-MAR-2025', 'DD-MON-YYYY'), TO_DATE('16-MAR-2025', 'DD-MON-YYYY'), 4)
/

```

### Voyage 2:

```

-- Voyage 2: Euro-Asia Express (Ocean Voyager)
INSERT INTO voyage_stops (stop_id, voyage_id, harbor_id, docking_time, departure_time, stop_sequence)
VALUES (stop_seq.NEXTVAL, 2, 2, TO_DATE('15-MAR-2025', 'DD-MON-YYYY'), TO_DATE('16-MAR-2025', 'DD-MON-YYYY'), 1)
/
INSERT INTO voyage_stops (stop_id, voyage_id, harbor_id, docking_time, departure_time, stop_sequence)
VALUES (stop_seq.NEXTVAL, 2, 1, TO_DATE('18-MAR-2025', 'DD-MON-YYYY'), TO_DATE('19-MAR-2025', 'DD-MON-YYYY'), 2)
/
INSERT INTO voyage_stops (stop_id, voyage_id, harbor_id, docking_time, departure_time, stop_sequence)
VALUES (stop_seq.NEXTVAL, 2, 4, TO_DATE('10-APR-2025', 'DD-MON-YYYY'), TO_DATE('12-APR-2025', 'DD-MON-YYYY'), 3)
/
INSERT INTO voyage_stops (stop_id, voyage_id, harbor_id, docking_time, departure_time, stop_sequence)
VALUES (stop_seq.NEXTVAL, 2, 7, TO_DATE('20-APR-2025', 'DD-MON-YYYY'), TO_DATE('22-APR-2025', 'DD-MON-YYYY'), 4)
/
INSERT INTO voyage_stops (stop_id, voyage_id, harbor_id, docking_time, departure_time, stop_sequence)
VALUES (stop_seq.NEXTVAL, 2, 2, TO_DATE('29-APR-2025', 'DD-MON-YYYY'), TO_DATE('30-APR-2025', 'DD-MON-YYYY'), 5)

```

## Inserted Data Reflected in Voyage Stops Table:

VOYAGE\_STOPS

	STOP_ID	VOYAGE_ID	HARBOR_ID	DOCKING_TIME	DEPARTURE_TIME	STOP_SEQUENCE
	1	1	1	3/1/2025	3/2/2025	1
	2	1	2	3/4/2025	3/5/2025	2
	3	1	9	3/12/2025	3/13/2025	3
	4	1	3	3/15/2025	3/16/2025	4
	5	1	8	3/18/2025	3/19/2025	5
	6	1	1	3/28/2025	3/30/2025	6
	7	2	2	3/15/2025	3/16/2025	1
	8	2	1	3/18/2025	3/19/2025	2
	9	2	4	4/10/2025	4/12/2025	3

### **3.15 Insert Queries for Shipping Rates Table:**

#### **Query 1:**

```
INSERT INTO shipping_rates (rate_id, departure_harbor, arrival_harbor, box_type_id, standard_fee, valid_from, valid_until)
VALUES (rate_seq.NEXTVAL, 1, 3, 1, 2650.00, TO_DATE('01-JAN-2025', 'DD-MON-YYYY'), TO_DATE('31-DEC-2025', 'DD-MON-YYYY'))
```

#### **Query 2:**

```
INSERT INTO shipping_rates (rate_id, departure_harbor, arrival_harbor, box_type_id, standard_fee, valid_from, valid_until)
VALUES (rate_seq.NEXTVAL, 1, 3, 2, 4250.00, TO_DATE('01-JAN-2025', 'DD-MON-YYYY'), TO_DATE('31-DEC-2025', 'DD-MON-YYYY'))
```

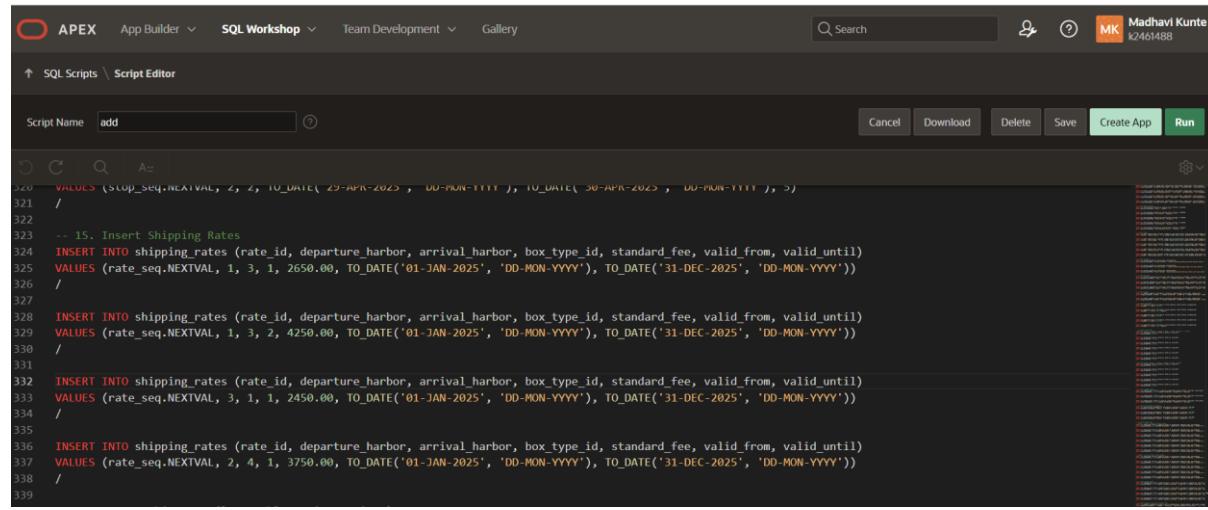
#### **Query 3:**

```
INSERT INTO shipping_rates (rate_id, departure_harbor, arrival_harbor, box_type_id, standard_fee, valid_from, valid_until)
VALUES (rate_seq.NEXTVAL, 3, 1, 1, 2450.00, TO_DATE('01-JAN-2025', 'DD-MON-YYYY'), TO_DATE('31-DEC-2025', 'DD-MON-YYYY'))
```

#### **Query 4:**

```
INSERT INTO shipping_rates (rate_id, departure_harbor, arrival_harbor, box_type_id, standard_fee, valid_from, valid_until)
VALUES (rate_seq.NEXTVAL, 2, 4, 1, 3750.00, TO_DATE('01-JAN-2025', 'DD-MON-YYYY'), TO_DATE('31-DEC-2025', 'DD-MON-YYYY'))
```

### **Insert Queries for Shipping Rates Table in Apex:**



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there are links for Search, Help, and a user profile (MK k2461488). Below the navigation, the main area has tabs for SQL Scripts and Script Editor, with 'SQL Scripts' selected. A script editor window displays four numbered insert statements:

```
320  VALUES ($stop_seq.NEXTVAL, 2, 2, TO_DATE('29-APR-2025', 'DD-MON-YYYY'), TO_DATE('30-APR-2025', 'DD-MON-YYYY'), 5)
321  /
322
323  -- 15. Insert Shipping Rates
324  INSERT INTO shipping_rates (rate_id, departure_harbor, arrival_harbor, box_type_id, standard_fee, valid_from, valid_until)
325  VALUES (rate_seq.NEXTVAL, 1, 3, 1, 2650.00, TO_DATE('01-JAN-2025', 'DD-MON-YYYY'), TO_DATE('31-DEC-2025', 'DD-MON-YYYY'))
326  /
327
328  INSERT INTO shipping_rates (rate_id, departure_harbor, arrival_harbor, box_type_id, standard_fee, valid_from, valid_until)
329  VALUES (rate_seq.NEXTVAL, 1, 3, 2, 4250.00, TO_DATE('01-JAN-2025', 'DD-MON-YYYY'), TO_DATE('31-DEC-2025', 'DD-MON-YYYY'))
330  /
331
332  INSERT INTO shipping_rates (rate_id, departure_harbor, arrival_harbor, box_type_id, standard_fee, valid_from, valid_until)
333  VALUES (rate_seq.NEXTVAL, 3, 1, 1, 2450.00, TO_DATE('01-JAN-2025', 'DD-MON-YYYY'), TO_DATE('31-DEC-2025', 'DD-MON-YYYY'))
334  /
335
336  INSERT INTO shipping_rates (rate_id, departure_harbor, arrival_harbor, box_type_id, standard_fee, valid_from, valid_until)
337  VALUES (rate_seq.NEXTVAL, 2, 4, 1, 3750.00, TO_DATE('01-JAN-2025', 'DD-MON-YYYY'), TO_DATE('31-DEC-2025', 'DD-MON-YYYY'))
338  /
339
```

## Inserted Data Reflected in Shipping Rates Table:

The screenshot shows the Oracle SQL Workshop interface with the 'Object Browser' on the left and the 'SHIPPING\_RATES' table in the center. The table has columns: RATE\_ID, DEPARTURE\_HARBO, ARRIVAL\_HARBOR, BOX\_TYPE\_ID, STANDARD\_FEE, VALID\_FROM, and VALID\_UNTIL. There are four rows inserted with values: (1, 1, 3, 1, 2650, '1/1/2025', '12/31/2025'), (2, 1, 3, 2, 4250, '1/1/2025', '12/31/2025'), (3, 3, 1, 1, 2450, '1/1/2025', '12/31/2025'), and (4, 2, 4, 1, 3750, '1/1/2025', '12/31/2025').

RATE_ID	DEPARTURE_HARBO	ARRIVAL_HARBOR	BOX_TYPE_ID	STANDARD_FEE	VALID_FROM	VALID_UNTIL
1	1	3	1	2650	1/1/2025	12/31/2025
2	1	3	2	4250	1/1/2025	12/31/2025
3	3	1	1	2450	1/1/2025	12/31/2025
4	2	4	1	3750	1/1/2025	12/31/2025

## 3.16 Insert Queries for Shipments Table:

### **Query 1:**

```
INSERT INTO shipments (shipment_id, order_id, box_id, commodity_type, commodity_details, mass_tonnes, voyage_id, pickup_date, delivery_date)
VALUES (shipment_seq.NEXTVAL, 1, 'ML250002', 'Technology', 'Computing equipment and peripherals', 17.8, 1, TO_DATE('02-MAR-2025', 'DD-MON-YYYY'), NULL)
```

### **Query 2:**

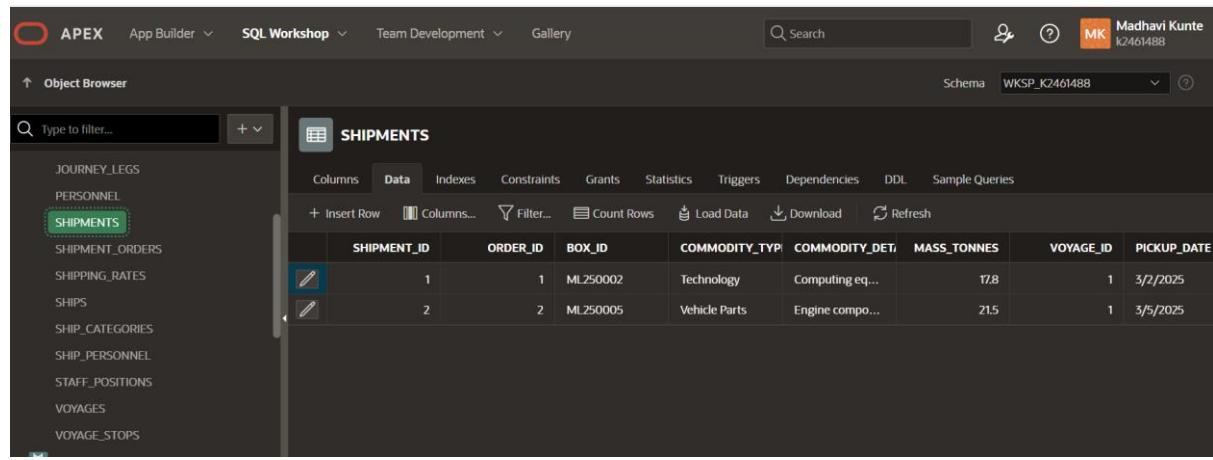
```
INSERT INTO shipments (shipment_id, order_id, box_id, commodity_type, commodity_details, mass_tonnes, voyage_id, pickup_date, delivery_date)
VALUES (shipment_seq.NEXTVAL, 2, 'ML250005', 'Vehicle Parts', 'Engine components and chassis parts', 21.5, 1, TO_DATE('05-MAR-2025', 'DD-MON-YYYY'), NULL)
```

## Insert Queries for Shipments Table in Apex:

The screenshot shows the Oracle Apex 'Script Editor' with the following SQL code:

```
339 -- 16. Insert Shipments (last table to insert into)
340 INSERT INTO shipments (shipment_id, order_id, box_id, commodity_type, commodity_details, mass_tonnes, voyage_id, pickup_date, delivery_date)
341 VALUES (shipment_seq.NEXTVAL, 1, 'ML250002', 'Technology', 'Computing equipment and peripherals', 17.8, 1, TO_DATE('02-MAR-2025', 'DD-MON-YYYY'),
342 /
343
344 INSERT INTO shipments (shipment_id, order_id, box_id, commodity_type, commodity_details, mass_tonnes, voyage_id, pickup_date, delivery_date)
345 VALUES (shipment_seq.NEXTVAL, 2, 'ML250005', 'Vehicle Parts', 'Engine components and chassis parts', 21.5, 1, TO_DATE('05-MAR-2025',
346 /
347
348
```

## **Inserted Data Reflected in Shipments Table:**



The screenshot shows the Oracle SQL Workshop interface with the 'SHIPMENTS' table selected in the object browser. The table has the following structure and data:

SHIPMENT_ID	ORDER_ID	BOX_ID	COMMODITY_TYPE	COMMODITY_DETAIL	MASS_TONNES	VOYAGE_ID	PICKUP_DATE
1	1	ML250002	Technology	Computing eq...	17.8	1	3/2/2025
2	2	ML250005	Vehicle Parts	Engine compo...	21.5	1	3/5/2025

## **3.17 Multiplicity of Relationships**

### **Query 1: Ship Categories and Associated Ships:**

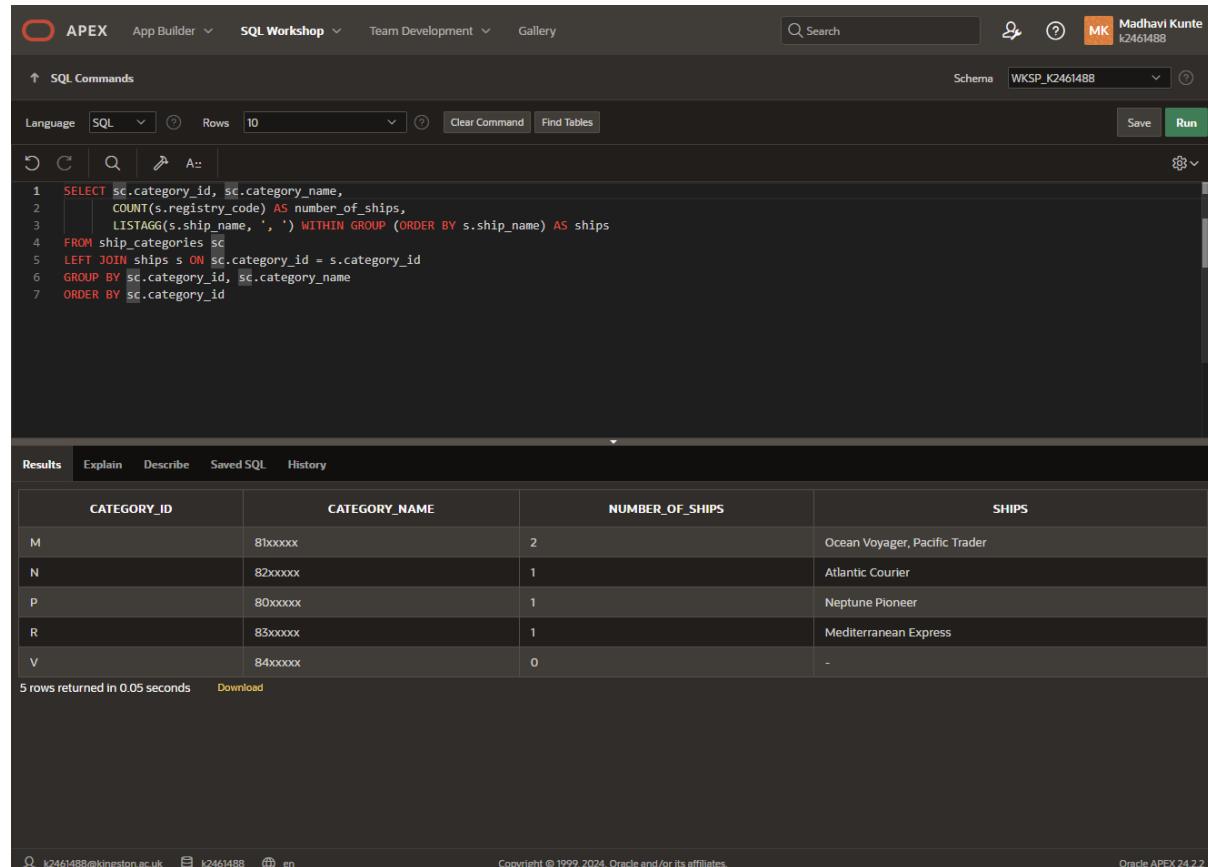
This SQL query retrieves information about each ship category, including the total number of ships associated with it and the names of those ships. It shows one to many relationships between Ship Categories and Ships

**Query:**

```
SELECT sc.category_id, sc.category_name,
       COUNT(s.registry_code) AS number_of_ships,
       LISTAGG(s.ship_name, ', ') WITHIN GROUP (ORDER BY s.ship_name) AS ships
  FROM ship_categories sc
 LEFT JOIN ships s ON sc.category_id = s.category_id
 GROUP BY sc.category_id, sc.category_name
 ORDER BY sc.category_id
```

**Purpose of the Query:** This query provides a summary of how many ships belong to each category and lists their names. It helps users understand ship distribution across various categories, including identifying categories that currently have no ships.

### **Query Result:**



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The top right corner shows the user's name 'Madhavi Kunte' and profile icon. The main workspace is titled 'SQL Commands'. The code editor contains the SQL query provided above. The results pane shows a table with five rows of data:

CATEGORY_ID	CATEGORY_NAME	NUMBER_OF_SHIPS	SHIPS
M	81xxxxx	2	Ocean Voyager, Pacific Trader
N	82xxxxx	1	Atlantic Courier
P	80xxxxx	1	Neptune Pioneer
R	83xxxxx	1	Mediterranean Express
V	84xxxxx	0	-

Below the table, a message indicates '5 rows returned in 0.05 seconds'. The bottom of the page includes standard footer links for Oracle APEX and copyright information.

## Query 2: Journey Summary with Leg Count and Path

This SQL query shows a detailed summary of each journey, including the number of legs (stops) and the full journey path showing the harbors visited in order. It shows one to many relationships between Journeys and Journey Legs

### Query:

```
SELECT j.journey_id, j.journey_code,
       COUNT(jl.leg_id) AS number_of_legs,
       LISTAGG(h.harbor_name || '#' || jl.stop_order || ')', ' → ')
          WITHIN GROUP (ORDER BY jl.stop_order) AS journey_path
FROM journeys j
LEFT JOIN journey_legs jl ON j.journey_id = jl.journey_id
LEFT JOIN harbors h ON jl.harbor_id = h.harbor_id
GROUP BY j.journey_id, j.journey_code
ORDER BY j.journey_id
```

**Purpose of the Query:** This query helps visualize each journey's structure:

- **How many legs** it includes.
- **The exact path** (i.e., the sequence of harbors visited).

It's especially useful for logistics, tracking, or reporting purposes where understanding route structure is essential.

## Query Result:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there is a user profile for 'Madhavi Kunte' (k2461488). The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab displays the query code. The Results tab shows the output:

JOURNEY_ID	JOURNEY_CODE	NUMBER_OF_LEGS	JOURNEY_PATH
1	Atlantic Connector	6	Liverpool (#1) → Bremen (#2) → Boston (#3) → Tampa (#4) → Savannah (#5) → Liverpool (#6)
2	Euro-Asia Express	5	Bremen (#1) → Liverpool (#2) → Qingdao (#3) → Sydney (#4) → Bremen (#5)
3	Pacific Passage	0	(#)

At the bottom, it says '3 rows returned in 0.05 seconds' and provides a 'Download' link. The footer includes the user email 'k2461488@kingston.ac.uk', session ID 'k2461488', language 'en', copyright information 'Copyright © 1999, 2024, Oracle and/or its affiliates.', and the version 'Oracle APEX 24.2.2'.

## 4 Chapter 4: Queries to demonstrate that your database meets the requirements of the system

### 4.1 Query 1: Displaying Journey Paths with Stop Order

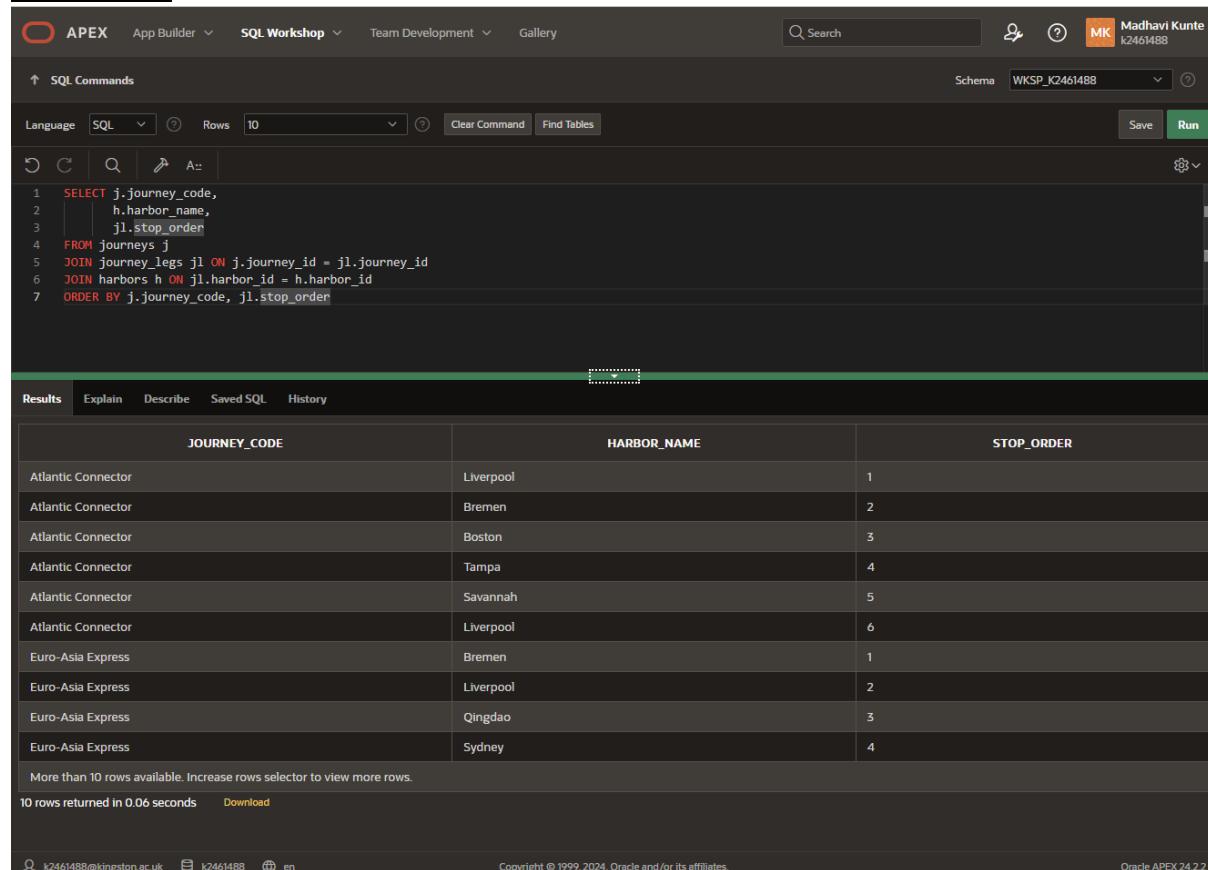
This query retrieves and displays the step-by-step route for each journey, showing the harbor names visited and their order of stops.

```
SELECT j.journey_code,
       h.harbor_name,
       jl.stop_order
  FROM journeys j
 JOIN journey_legs jl ON j.journey_id = jl.journey_id
 JOIN harbors h ON jl.harbor_id = h.harbor_id
 ORDER BY j.journey_code, jl.stop_order
```

**Purpose of the Query:** This query is useful for:

- Tracking the journey route of each shipment.
- Visualizing port sequences for logistics and planning.
- Auditing or validating the planned journey structure.

#### Query Result:



The screenshot shows the Oracle APEX SQL Workshop interface. The query is displayed in the SQL Commands pane:

```
1 SELECT j.journey_code,
2        h.harbor_name,
3        jl.stop_order
4   FROM journeys j
5  JOIN journey_legs jl ON j.journey_id = jl.journey_id
6  JOIN harbors h ON jl.harbor_id = h.harbor_id
7 ORDER BY j.journey_code, jl.stop_order
```

The Results pane displays the output:

JOURNEY_CODE	HARBOR_NAME	STOP_ORDER
Atlantic Connector	Liverpool	1
Atlantic Connector	Bremen	2
Atlantic Connector	Boston	3
Atlantic Connector	Tampa	4
Atlantic Connector	Savannah	5
Atlantic Connector	Liverpool	6
Euro-Asia Express	Bremen	1
Euro-Asia Express	Liverpool	2
Euro-Asia Express	Qingdao	3
Euro-Asia Express	Sydney	4

Below the results, a message indicates "More than 10 rows available. Increase rows selector to view more rows." and "10 rows returned in 0.06 seconds". The bottom of the screen shows navigation links and copyright information.

## **4.2 Query 2: Ship Voyage Summary with Container Count**

This query provides a **summary of each ship's voyage**, along with details such as the **journey code, departure and arrival dates**, and the **total number of cargo containers** loaded onto each ship.

```
SELECT s.ship_name,
       v.voyage_id,
       j.journey_code,
       v.departure_date,
       v.arrival_date,
       COUNT(cb.box_id) AS cargo_boxes_loaded
  FROM ships s
 LEFT JOIN voyages v ON s.registry_code = v.ship_registry
 LEFT JOIN journeys j ON v.journey_id = j.journey_id
 LEFT JOIN cargo_boxes cb ON cb.assigned_ship = s.registry_code
 GROUP BY s.ship_name, v.voyage_id, j.journey_code, v.departure_date, v.arrival_date
 ORDER BY v.departure_date, s.ship_name
```

### **Purpose of the Query:**

- To monitor shipping activity and voyage history.
- To track cargo loading on different voyages.

### **Query Result:**

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command window displays the query from the previous section. The results window shows the output of the query, which includes five rows of data for ships Neptune Pioneer, Ocean Voyager, Atlantic Courier, Mediterranean Express, and Pacific Trader. The columns are SHIP\_NAME, VOYAGE\_ID, JOURNEY\_CODE, DEPARTURE\_DATE, ARRIVAL\_DATE, and CARGO\_BOXES\_LOADED.

SHIP_NAME	VOYAGE_ID	JOURNEY_CODE	DEPARTURE_DATE	ARRIVAL_DATE	CARGO_BOXES_LOADED
Neptune Pioneer	1	Atlantic Connector	3/1/2025	3/30/2025	1
Ocean Voyager	2	Euro-Asia Express	3/15/2025	4/30/2025	1
Atlantic Courier	-	-	-	-	0
Mediterranean Express	-	-	-	-	0
Pacific Trader	-	-	-	-	0

### **4.3 Query 3: Shipment Details with Customer Information**

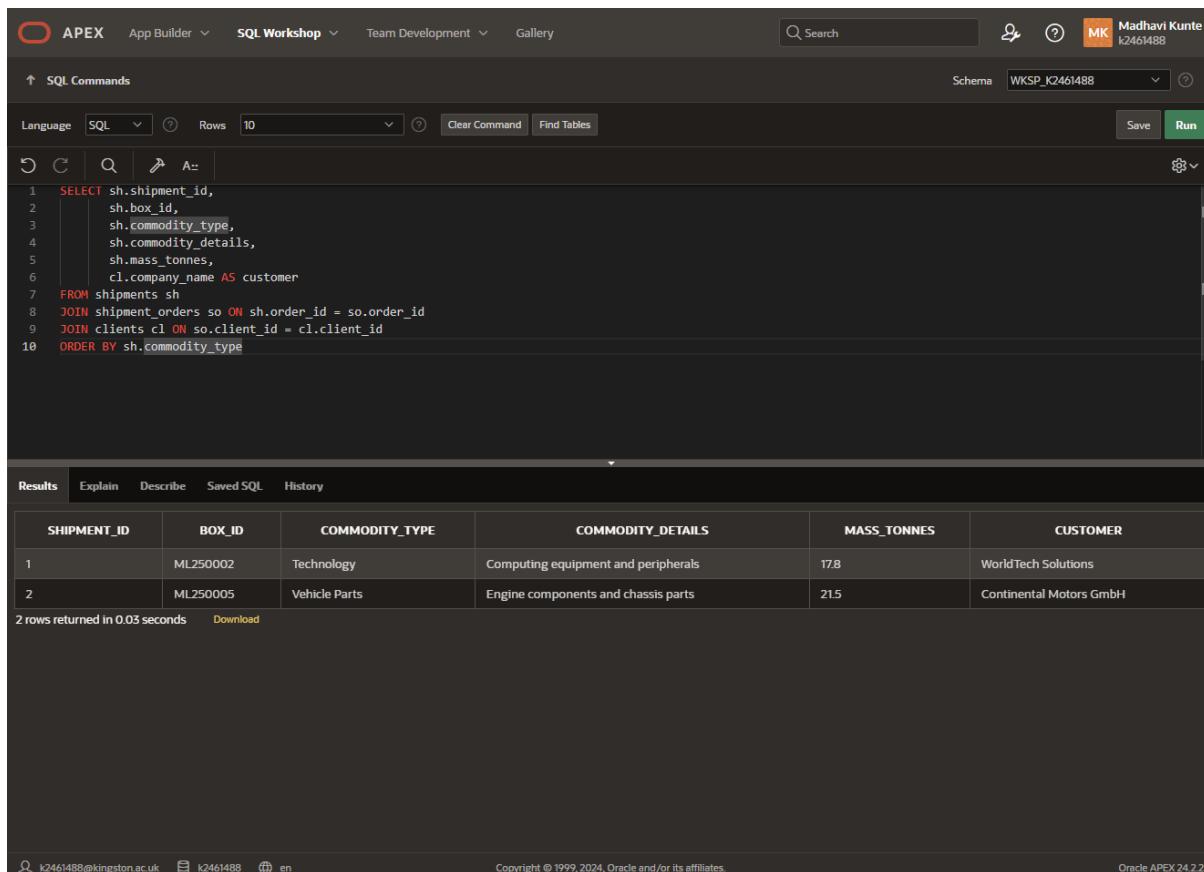
This query retrieves a detailed list of **individual shipments**, including the type and description of the commodity, its weight, and the name of the **customer (client)** who placed the shipment order.

```
SELECT sh.shipment_id,
       sh.box_id,
       sh.commodity_type,
       sh.commodity_details,
       sh.mass_tonnes,
       cl.company_name AS customer
  FROM shipments sh
 JOIN shipment_orders so ON sh.order_id = so.order_id
 JOIN clients cl ON so.client_id = cl.client_id
 ORDER BY sh.commodity_type
```

#### **Purpose of the Query:**

- To track and analyze cargo shipments by type and customer.
- Useful in generating customer shipment summaries.

#### **Query Result:**



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The top right corner shows the user's profile (MK) and schema (WKSP\_K2461488). The main workspace has tabs for SQL Commands, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the query code shown above. Below the code, the Results tab displays a table with two rows of data. The table has columns: SHIPMENT\_ID, BOX\_ID, COMMODITY\_TYPE, COMMODITY\_DETAILS, MASS\_TONNES, and CUSTOMER. The first row corresponds to the query results shown in the code block.

SHIPMENT_ID	BOX_ID	COMMODITY_TYPE	COMMODITY_DETAILS	MASS_TONNES	CUSTOMER
1	ML250002	Technology	Computing equipment and peripherals	17.8	WorldTech Solutions
2	ML250005	Vehicle Parts	Engine components and chassis parts	21.5	Continental Motors GmbH

2 rows returned in 0.03 seconds [Download](#)

#### **4.4 Query 4: List of Ready Cargo Boxes with Location and Weight Capacity**

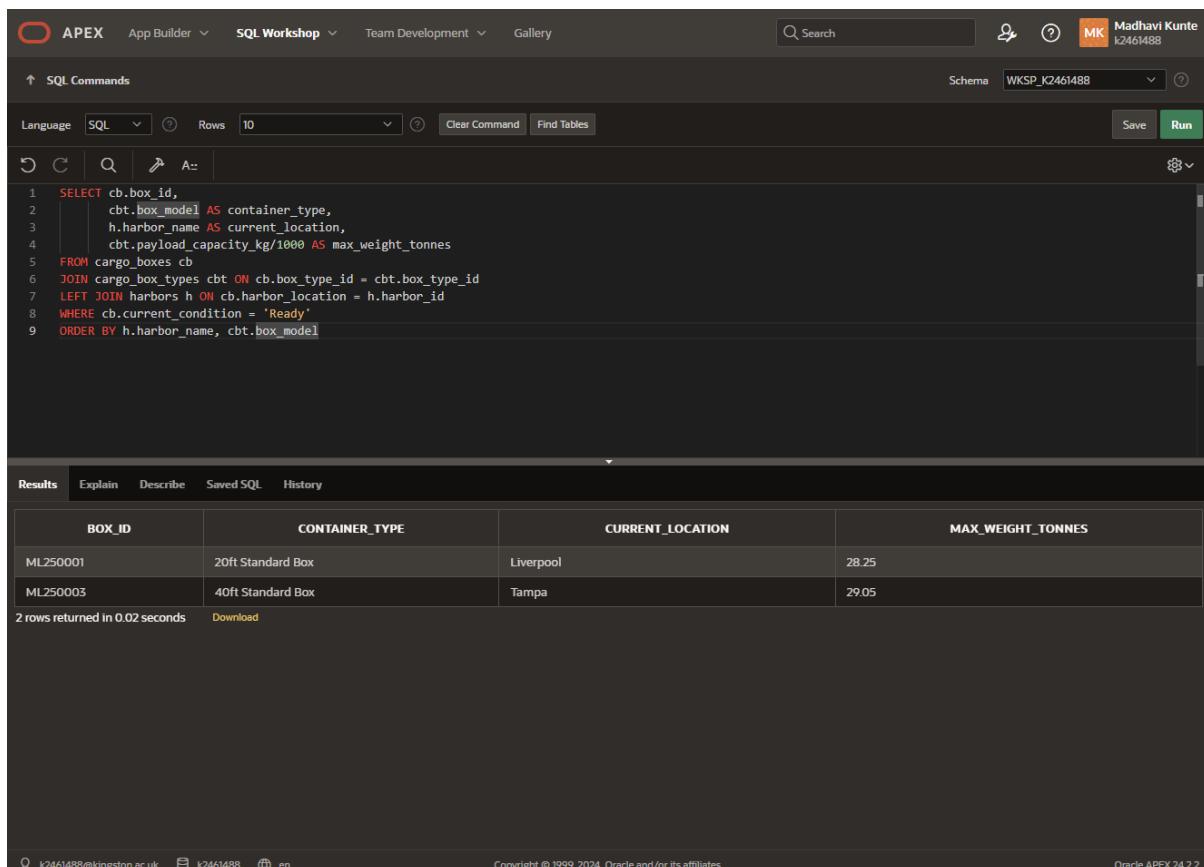
This query shows a list of all **cargo boxes that are currently marked as 'Ready'** for shipment, including the type of container, its current location, and its maximum weight capacity in tonnes.

```
SELECT cb.box_id,
       cbt.box_model AS container_type,
       h.harbor_name AS current_location,
       cbt.payload_capacity_kg/1000 AS max_weight_tonnes
  FROM cargo_boxes cb
 JOIN cargo_box_types cbt ON cb.box_type_id = cbt.box_type_id
 LEFT JOIN harbors h ON cb.harbor_location = h.harbor_id
 WHERE cb.current_condition = 'Ready'
 ORDER BY h.harbor_name, cbt.box_model
```

#### **Purpose of the Query:**

- To monitor available cargo boxes across different harbors.

#### **Query Result:**



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows the user 'Madhavi Kunte k2461488'. The main area has tabs for 'SQL Commands' (selected) and 'Results'. In the 'SQL Commands' tab, the query is pasted. In the 'Results' tab, a table is displayed with the following data:

BOX_ID	CONTAINER_TYPE	CURRENT_LOCATION	MAX_WEIGHT_TONNES
ML250001	20ft Standard Box	Liverpool	28.25
ML250003	40ft Standard Box	Tampa	29.05

Below the table, it says '2 rows returned in 0.02 seconds' and there is a 'Download' link. The bottom of the page shows the footer with user information and copyright details.

## **4.5 Query 5: Shipment Order Tracking with Vessel and Harbor Details**

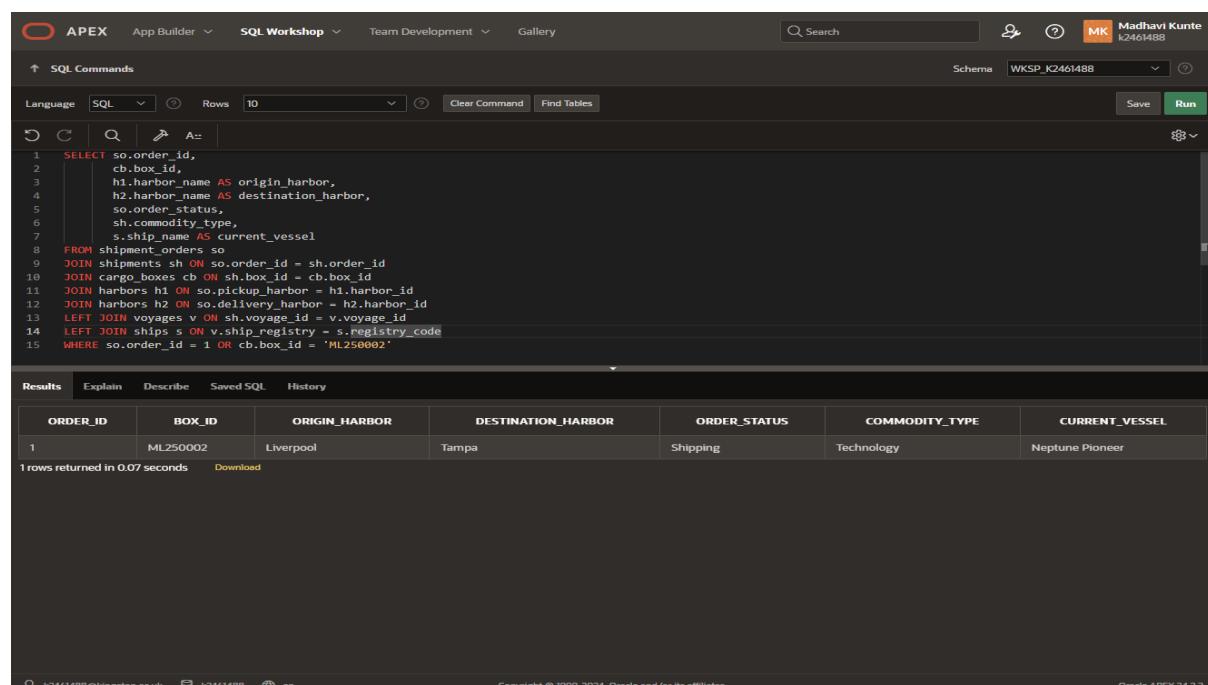
This query demonstrates a view of a specific shipment order or cargo box, including details like the **origin and destination harbors**, **order status**, **commodity type**, and the **vessel currently assigned** to the shipment (if any).

```
SELECT so.order_id,
       cb.box_id,
       h1.harbor_name AS origin_harbor,
       h2.harbor_name AS destination_harbor,
       so.order_status,
       sh.commodity_type,
       s.ship_name AS current_vessel
  FROM shipment_orders so
 JOIN shipments sh ON so.order_id = sh.order_id
 JOIN cargo_boxes cb ON sh.box_id = cb.box_id
 JOIN harbors h1 ON so.pickup_harbor = h1.harbor_id
 JOIN harbors h2 ON so.delivery_harbor = h2.harbor_id
 LEFT JOIN voyages v ON sh.voyage_id = v.voyage_id
 LEFT JOIN ships s ON v.ship_registry = s.registry_code
 WHERE so.order_id = 1 OR cb.box_id = 'ML250002'
```

### **Purpose of the Query:**

- To track shipment status and current vessel assignment.
- Helps view the movement plan for specific orders or containers.

### **Query Result:**



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The right side shows a user profile for 'Madhavi Kunte' (k2461488). The main area has tabs for 'SQL Commands' (selected), 'Explain', 'Describe', 'Saved SQL', and 'History'. The SQL editor contains the query from the previous code block. The results tab shows a single row of data:

ORDER_ID	BOX_ID	ORIGIN_HARBOR	DESTINATION_HARBOR	ORDER_STATUS	COMMODITY_TYPE	CURRENT_VESSEL
1	ML250002	Liverpool	Tampa	Shipping	Technology	Neptune Pioneer

Below the table, it says '1 rows returned in 0.07 seconds' and there is a 'Download' link. The bottom footer includes the URL 'k2461488@kingston.ac.uk', session ID 'k2461488', language 'en', copyright notice 'Copyright © 1999, 2024, Oracle and/or its affiliates.', and 'Oracle APEX 24.2.2'.

## **4.6 Query 6: Shipment Order Tracking with Vessel and Harbor Details**

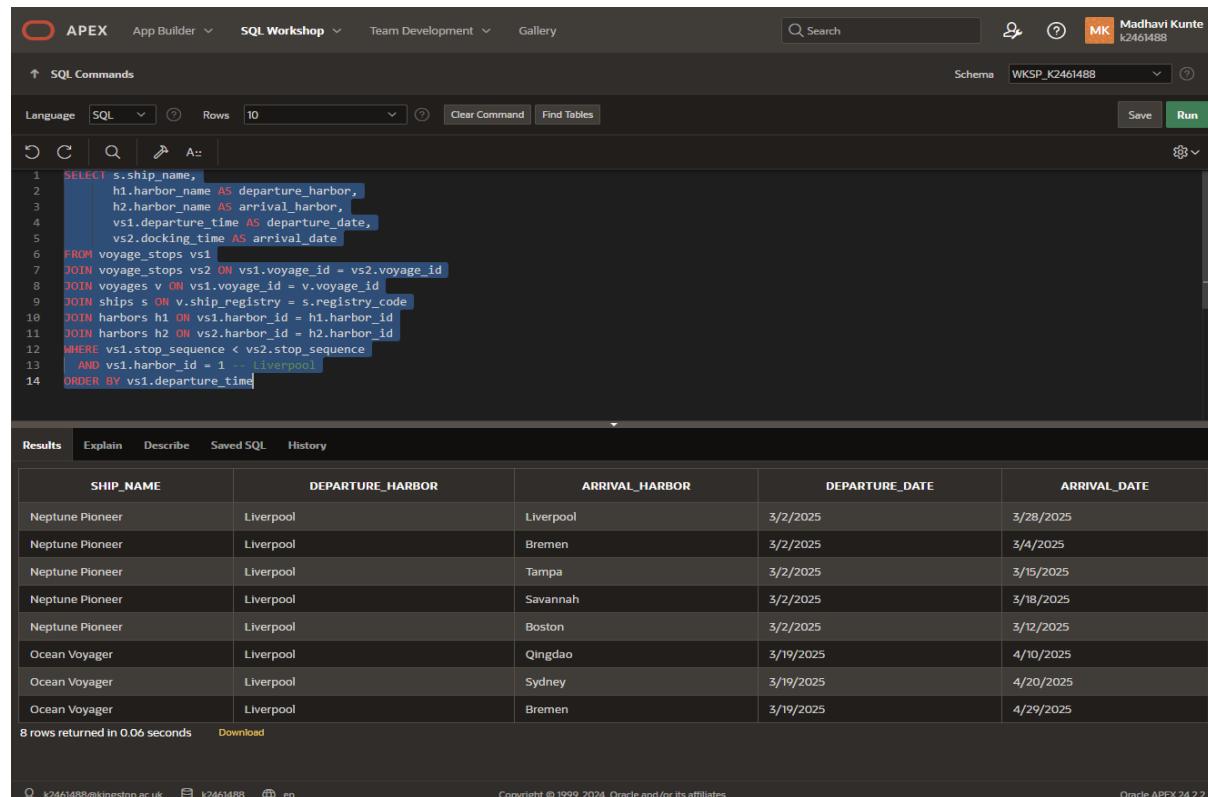
This query will list all **ship voyages that depart from Liverpool**, including the **arrival harbor** and the **estimated departure and arrival times**.

```
SELECT s.ship_name,
       h1.harbor_name AS departure_harbor,
       h2.harbor_name AS arrival_harbor,
       vs1.departure_time AS departure_date,
       vs2.docking_time AS arrival_date
  FROM voyage_stops vs1
 JOIN voyage_stops vs2 ON vs1.voyage_id = vs2.voyage_id
 JOIN voyages v ON vs1.voyage_id = v.voyage_id
 JOIN ships s ON v.ship_registry = s.registry_code
 JOIN harbors h1 ON vs1.harbor_id = h1.harbor_id
 JOIN harbors h2 ON vs2.harbor_id = h2.harbor_id
 WHERE vs1.stop_sequence < vs2.stop_sequence
   AND vs1.harbor_id = 1 -- Liverpool
 ORDER BY vs1.departure_time
```

### **Purpose of the Query:**

- To track ship departures from Liverpool and to their next destination.

### **Query Result:**



The screenshot shows the Oracle APEX SQL Workshop interface. The query has been run, and the results are displayed in a table. The table has columns: SHIP\_NAME, DEPARTURE\_HARBOR, ARRIVAL\_HARBOR, DEPARTURE\_DATE, and ARRIVAL\_DATE. The results show multiple rows for different ships, with Liverpool as the departure harbor and various destinations like Bremen, Tampa, and Sydney. The departure dates range from March 2, 2025, to April 19, 2025, with corresponding arrival dates.

SHIP_NAME	DEPARTURE_HARBOR	ARRIVAL_HARBOR	DEPARTURE_DATE	ARRIVAL_DATE
Neptune Pioneer	Liverpool	Liverpool	3/2/2025	3/28/2025
Neptune Pioneer	Liverpool	Bremen	3/2/2025	3/4/2025
Neptune Pioneer	Liverpool	Tampa	3/2/2025	3/15/2025
Neptune Pioneer	Liverpool	Savannah	3/2/2025	3/18/2025
Neptune Pioneer	Liverpool	Boston	3/2/2025	3/12/2025
Ocean Voyager	Liverpool	Qingdao	3/19/2025	4/10/2025
Ocean Voyager	Liverpool	Sydney	3/19/2025	4/20/2025
Ocean Voyager	Liverpool	Bremen	3/19/2025	4/29/2025

## **5. Chapter 5: Conclusion**

### **Evaluation of the Project:**

This project helped developing a relational database model for a shipping company called “Royal Anchor Shipping”. In this project the database was created in Oracle Apex to test the system requirements. The system is designed to manage operations such as Ship and crew management, tracking of cargo, routing, shipment orders and customer handling, port management. This project really helped in gaining more depth technical knowledge about how the database works and different concepts of database.

The project covered important things like ER Diagram creation with constraints and assumptions. Creating tables for the database keeping in mind all the constraints and assumption for Primary key, foreign key and maintaining relationships between tables. Also, it covered creating queries to test the database. The Oracle APEX Platform was really robust and the interface made testing easy and convenient.

In addition, the project illustrates the ability to practically simulate logistics operations from a real-world work environment to a digital-based database. The queries to assist in knowledge extraction across different database tables provided a rationale for the database operating correctly.

### **Self-Evaluation:**

This coursework greatly improved my knowledge about database concepts from basics like DDL, DML statements, ER diagram creation, use of different types of keys. I also got clear understanding about how “Joins” works between two tables. The implementation in Oracle APEX was challenging but rewarding, particularly when seeing how the relationships worked together to support the business logic. The most successful aspect was creating a normalized database schema that prevents data redundancy while maintaining referential integrity across all tables. Moreover, the experience of creating tables and inserting data by forming correct data and inserting correct data really helped me in revising my concepts of basic database. Also, I learned about sequencing part in Oracle database which are normally used to generate unique numeric values for primary keys. These sequences ensure that each new record inserted into a table gets a unique identifier, which is essential for maintaining data integrity and enabling proper entity relationships. This project also helped me in understanding new concepts of database.

### **What went well? and what didn't?**

Firstly, Introduction to Oracle Apex was really helpful for me. I found creation of tables and implementation of insert queries very easy as compared to other databases. I studied in depth about ER diagram like what are attributes? entities? different arrow symbols used in ER diagram which really cleared my concepts.

Secondly, Creation of tables and insertion data helped in clearing basic concepts of creation and insertion query syntax. Working on Joins also went well as I got deep understanding on how data is fetched from different tables

Thirdly, I specifically felt I have enhanced my expertise and understanding of Oracle APEX in developing correct complex queries with several joins and constraints. Moreover, the experience of constructing an ER diagram gave me increased precision around logical reasoning in thinking about multiplicity, referential integrity, and normalization concepts.

The most challenging aspects were troubleshooting SQL script execution issues in Oracle APEX and ensuring proper ordering of data insertion to prevent foreign key constraint violations. Additionally, some complex queries required several iterations to perfect. Queries required several iterations to perfect.

### **What should you do differently in future?**

If starting over, I would create a more detailed initial data model with sample data before implementation to identify potential issues earlier. I would also develop a more thorough testing strategy for each table and relationship. Using SQL Developer Data Modeler from the beginning would have simplified the creation of the ER diagram and subsequent conversion to SQL. Moreover, I could have tried out more of Oracle APEX features like forms and reports to create a more friendly and interactive system. I would also try to implement some more concepts in the project like stored procedure and triggers.

Overall, this project provided valuable experience in database design and implementation, showcasing how a properly structured database can address complex real-world business requirements.