

**AMERICAN INTERNATIONAL UNIVERSITY  
BANGLADESH (AIUB)**

***FACULTY OF SCIENCE & TECHNOLOGY***



Course Title  
**INTRODUCTION TO DATABASE (CSC2108)**

**Semester: FALL 2024-2025**

**Section: [D]**

**TITLE**

**Freight Forwarder Management System (FFMS)**

**Supervised By**

**Syed Nafiul Shefat**

**Submitted By Group no:**

Name	ID
MOHAMMAD ALI MASUD	23-51676-2
MD. MAHABUBUR RAHAMAN SIAM	23-51706-2
SHAKILA SULTANA ESHITA	23-51719-2
TAWAFIQUR RAHMAN MAHIR	23-51687-2

## TABLE OF CONTENTS

<b>TOPICS</b>	<b>Page no.</b>
<b>Title Page</b>	<b>1</b>
<b>Table of Content</b>	<b>2</b>
<b>1. Introduction</b>	<b>3</b>
<b>2. Case Study</b>	<b>4</b>
<b>3. ER Diagram</b>	<b>5</b>
<b>4. Normalization</b>	<b>6-11</b>
<b>5. Finalization</b>	<b>12</b>
<b>6. Table Creation</b>	<b>13-32</b>
<b>7. Data Insertion</b>	<b>23-29</b>
<b>8. Query Test</b>	<b>30-34</b>
<b>9. DB connection</b>	<b>35</b>

## Introduction

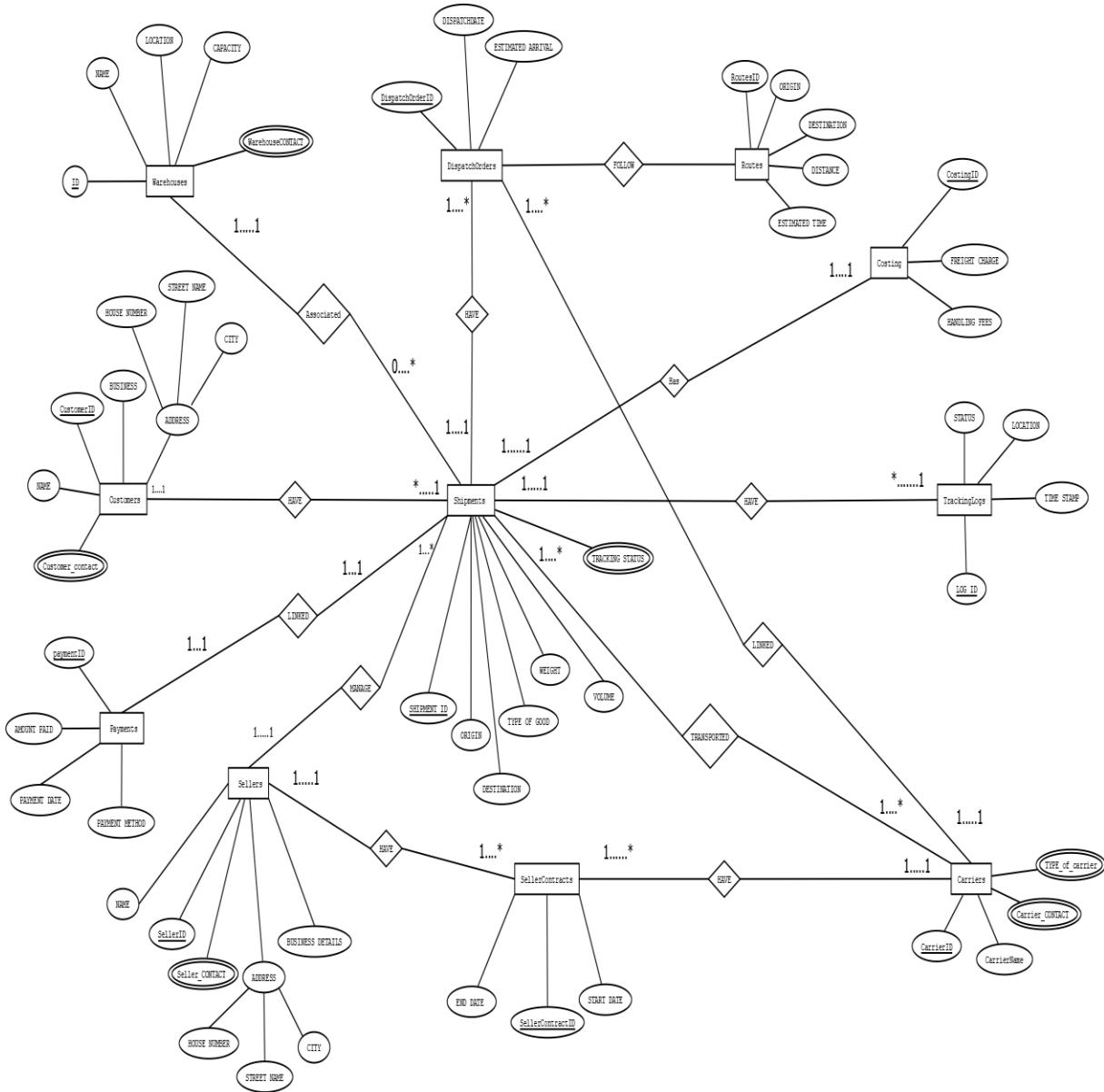
The Freight Forwarder Management System (FFMS) is a comprehensive software solution designed to streamline and automate logistics operations for freight forwarding companies. Built using **Oracle 10g** for robust database management and **Java** for backend development, the system integrates features like customer and shipment management, real-time tracking, carrier and seller contracts, costing, payments, and warehouse management. The goal is to enhance efficiency, reduce manual errors, and provide real-time visibility into shipment statuses and financial transactions. The project aims to revolutionize the logistics industry by offering a centralized, scalable, and user-friendly platform that optimizes costs, improves transparency, and supports sustainable practices. Targeted users include freight forwarders, customers, sellers, carriers, and logistics managers, all of whom benefit from streamlined operations, real-time updates, and secure transactions. This project contributes to the tech sector by fostering innovation in logistics software and encouraging digital transformation in the industry.

## Case Study / Scenario

### **The case study of Freight Forwarder**

**Case study:**In a Freight Forwarder Management System, a customer may initiate many shipments. A shipment may be associated with exactly one customer. A customer is identified by a customer ID. The system also stores the customer name, contact information, and address. A customer's address is composed of house number, street name, and city. A shipment is identified by shipment ID. The shipment details include origin, destination, type of goods, weight, volume, and tracking status. There may be multiple updates to the tracking status as the shipment progresses. While shipping, tracking status are recorded.A shipment is transported by at least one carrier. A carrier may handle many shipments but the system stores information about carriers who have managed at least one shipment. To identify a carrier, the system stores carrier ID along with carrier name ,Contact and type.There will be multiple contact number. A shipment belongs to exactly one seller, and for a seller, there must be at least one shipment being handled. Each seller has a unique seller ID and the system stores seller name and business details,contact number,address.The seller address is composed of house number, street name, and city.A shipment has associated costing that includes details such as ID, freight charges and handling fees. Seller contracts are created between sellers and carriers to manage the shipment of goods. Each contract has a unique contract ID, and the system stores contract details such as the start date, end date, and the specific carrier and seller involved.Each shipment may be stored in a warehouse before it is dispatched. A warehouse is identified by warehouse ID, and the system stores the warehouse name, location, capacity, and multiple contact information.Each shipment can also have multiple tracking logs that record the location, status, and time stamp. These logs provide updates on the shipment's progress. The shipment may also have dispatch orders, where a dispatch order ID is assigned to each order, and also stored date,estimated arrival linking the shipment to the carrier and route used for the transport.Finally, the system also manages payments for each shipment. A payment is linked to a shipment and includes payment ID, amount paid, payment method, and payment date.

# ER Diagram



# Normalization

## **1...1 shipment have 1...\* dispatch order**

**Unf:** ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume, TrackingStatus, DispatchOrderID(PK), Dispatch\_Date, Estimated\_Arrival.

**1NF:** TrackingStatus is a multivalued attribute.

### **2NF:**

1. ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume.
2. DispatchOrderID (PK), DispatchDate, EstimatedArrival, ShipmentID(FK)
3. TrackingStatus, TrackingID(PK) , ShipmentID(FK)

### **3NF:**

1. ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume.
2. DispatchOrderID(PK), DispatchDate, EstimatedArrival, ShipmentID(FK)
3. TrackingStatus, TrackingID(PK) , ShipmentID(FK).

## **1....1 shipment have 1....\* trackinglogs**

**Unf:** ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume, TrackingStatus, LogID (PK), Timestamp, Location, Status.

**1NF:** TrackingStatus is a multivalued attribute.

### **2NF:**

1. ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume,
2. LogID (PK), Timestamp, Location, Status, ShipmentID(FK)
3. TrackingStatus, TrackingID(PK), ShipmentID(FK).

### **3NF:**

1. ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume.
2. LogID (PK), Timestamp, Location, Status, ShipmentID(FK).
3. TrackingStatus, TrackingID(PK) , ShipmentID(FK).

## **1....1 shipment linked 1....1 payments**

**Unf:** ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume, TrackingStatus, PaymentID (PK), AmountPaid, PaymentDate, PaymentMethod.

**1NF:** TrackingStatus is a multivalued attribute.

**2NF:**

1. ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume.

2. PaymentID (PK), AmountPaid, PaymentDate, PaymentMethod, ShipmentID(FK).

3. TrackingStatus, TrackingID(PK), ShipmentID(FK).

**3NF:**

1. ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume.

2. PaymentID (PK), AmountPaid, PaymentDate, PaymentMethod, ShipmentID(FK).

3. TrackingStatus, TrackingID(PK), ShipmentID(FK).

### **1....\* shipment transported 1....\* carriers**

**Unf:** ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume, TrackingStatus, CarrierID(PK), Name, carrier\_Contact, TypeOfCarriers

**1NF:** TrackingStatus is a multivalued attribute, carrier\_Contact is a multivalued attribute, Type is a multivalued attribute.

**2NF:**

1. ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume.

2. CarrierID (PK), Name, ShipmentID(FK).

3. TrackingStatus, TrackingID(PK), ShipmentID(FK).

4. carrier\_Contact, Carrier\_ContactID(PK), CarrierID (FK).

5. TypeOfCarriers, TypeID(PK), CarrierID (FK).

**3NF:**

1. ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume.

2. CarrierID (PK), Name, ShipmentID(FK).

3. TrackingStatus, TrackingID(PK), ShipmentID(FK).

4. carrier\_Contact, carrier\_ContactID(PK), CarrierID (FK),

5. TypeOfCarriers, TypeOfCarriersID(PK), CarrierID (FK),

**1....\* shipment manage 1....1 sellers**

**UNF:** ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume, TrackingStatus, SellerID(PK), Name, Seller\_Contact, HouseNumber, streetName, city, BusinessDetails.

**1NF:** TrackingStatus is a multivalued attribute, Seller\_Contact is a multivalued attribute.

**2NF:**

1. ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume, SellerID (FK)

2. SellerID (PK), Name, House\_Number, StreetName, City, BusinessDetails.

3. TrackingStatus, TrackingID(PK), ShipmentID(FK).

4. Seller\_Contact, Seller\_ContactID(PK), SellerID (FK)

**3NF:**

1. ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume, SellerID (FK)

2. SellerID (PK), Name, BusinessDetails, Seller\_CityID (FK)

3. TrackingStatus, TrackingID(PK), ShipmentID(FK).

4. Seller\_Contact, Seller\_ContactID(PK), SellerID (FK)

5. House\_Number, street\_name, city, Seller\_CityID(PK)

**1....\*shipment**

**associated**

**1....1**

**warehouses**

**UNF:** ShipmentID(PK), Weight, Volume, Type of Goods, Origin, Destination, TrackingStatus, ID(PK), Name, Location, Capacity, warehouse\_Contact.

**1NF:** TrackingStatus is a multivalued attribute, warehouse\_Contact is a multivalued attribute.

**2NF:**

1: ShipmentID(PK), Weight, volume, TypeOfGoods,

Origin, Destination, WarehouseID(FK),

2: WarehouseID(PK), Name, Location, Capacity,

3: TrackingStatus, TrackingID(PK), ShipmentID(FK).

4: warehouse\_Contact, warehouse\_Contact ID(PK), WarehouseID(FK),

**3NF:**

1: ShipmentID(PK), Weight, volume, TypeOfGoods,

Origin, Destination, WarehouseID(FK),

2: WarehouseID(PK), Name, Location, Capacity,

3: TrackingStatus, TrackingID(PK), ShipmentID(FK).

4: warehouse\_Contact, warehouse\_ContactID(PK), WarehouseID(FK).

**1....1 Shipments linked 1....1 payments**

**UNF:** ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume, TrackingStatus, PaymentID(PK), AmountPaid, PaymentDate, PaymentMethod

**1NF:** Trackingstatus is multivalued attribute.

**2NF:**

1: ShipmentID(PK), Origin, Destination, Typeofgoods, Weight, Volume

2: PaymentID(PK), AmountPaid, PaymentDate, PaymentMethod, Shipment ID (FK)

3: TrackingStatus, TrackingID(PK), ShipmentID(FK).

**3NF:**

1: ShipmentID(PK), Origin, Destination, Typeofgoods, Weight, Volume

2: PaymentID(PK), AmountPaid, PaymentDate, PaymentMethod, Shipment ID (FK)

3: TrackingStatus, TrackingID(PK), ShipmentID(FK).

**1....\***              **shipment**              **have**              **1....1**              **customer**

**UNF:** ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume, TrackingStatus, CustomerID(PK), Name, Customer\_Contact, House\_Number, Street\_Name, City, Business

**1NF:** TrackingStatus is a multivalued attribute,

Customer\_Contact is a multivalued attribute.

**2NF:**

- 1 :ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume, CustomerID (FK)
- 2:CustomerID (PK), Name, house\_Number, Street\_Name, City, Business
- 3:TrackingStatus, TrackingID(PK), ShipmentID(FK).
- 4: Customer\_Contact, Customer\_ContactID(PK), CustomerID (FK)

**3NF:**

- 1 :ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume, CustomerID (FK)
- 2:CustomerID (PK), Name, Business, Customer\_CityID(FK).
- 3:TrackingStatus, TrackingID(PK), ShipmentID(FK).
- 4: Customer\_Contact, Customer\_ContactID(PK), CustomerID (FK)
- 5:, House\_Number, Street\_Name, City, Customer\_CityID(PK).

**1.....1 Shipments has 1.....1costing**

UNF: ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume, TrackingStatus, CostingID (PK), FreightCharges , Handling Fees

**1NF:** TrackingStatus is a multivalued attribute

- 2NF:
1. ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume
  2. CostingID (PK) , FreightCharges , Handling Fees, ShipmentID(FK)
  3. TrackingStatus, TrackingID(PK), ShipmentID(FK)

- 3NF:
1. ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume
  2. CostingID (PK) , FreightCharges , Handling Fees, ShipmentID(FK)
  3. TrackingStatus, TrackingID(PK), ShipmentID(FK)

**1.....\*Dispatches Follow 1.....1Routes**

**UNF:** DispatchOrderID(PK), DispatchDate, EstimatedArrival, Routes\_ID(PK), Origin, Destination, Distance, EstimatedTime

**1NF:**

- 2 NF:
1. DispatchOrderID(PK), DispatchDate, EstimatedArrival, Routes\_ID(FK)

2. Routes\_ID(PK),Origin, Destination, Distance,EstimatedTime

3 NF: 1. DispatchOrderID(PK),DispatchDate,EstimatedArrival, Routes\_ID(FK)

2. Routes\_ID(PK),Origin, Destination, Distance,EstimatedTime

#### **1....\*DispatchOrders Linked 1...1Carriers**

UNF : DispatchOrderID(PK),DispatchDate,EstimatedArrival, CarrierID(PK), CarrierName, Carrier\_Contact, TypeOfCarriers

1NF: CarrierContact is a multivalued attribute, TypeOfCarriers is a multivalued attribute.

2NF :1. DispatchOrderID(PK),DispatchDate,EstimatedArrival, CarrierID(FK)

2. CarrierID(PK),CarrierName

3. Carrier\_Contact, Carrier\_ContactID(PK), CarrierID(FK)

4. TypeOfCarriers, TypeOfCarrierID(PK), CarrierID(FK)

3NF: 1. DispatchOrderID(PK),DispatchDate,EstimatedArrival, CarrierID(FK)

2. CarrierID(PK),CarrierName

3. Carrier\_Contact, Carrier\_ContactID(PK), CarrierID(FK)

4. TypeOfCarriers, TypeOfCarrierID(PK), CarrierID(FK)

#### **1...1Sellers Have 1...\*SellerContracts**

UNF: SellerID(PK), Name, Seller\_Contact, House\_Number,street\_Name, city, BusinessDetails, SellerContractsId(pk),EndDate,Startdate

1NF: Seller\_Contact is a multivalued attribute.

2NF:1. SellerID (PK),Name,House\_Number, StreetName, City, BusinessDetails.

2. SellerContractsId(pk),EndDate,Startdate, SellerID (FK)

3. Seller\_Contact, Seller\_Contact(PK), SellerID (FK)

3NF:1. SellerID (PK),Name, BusinessDetails, Seller\_CityID(FK)

2. SellerContractsId(pk),EndDate,Startdate, SellerID (FK), Seller\_CityID(PK).

3. Seller\_Contact, Seller\_ContactID(PK), SellerID(FK)

4. House\_Number, StreetName, City,Seller\_CityID(PK)

## Finalization

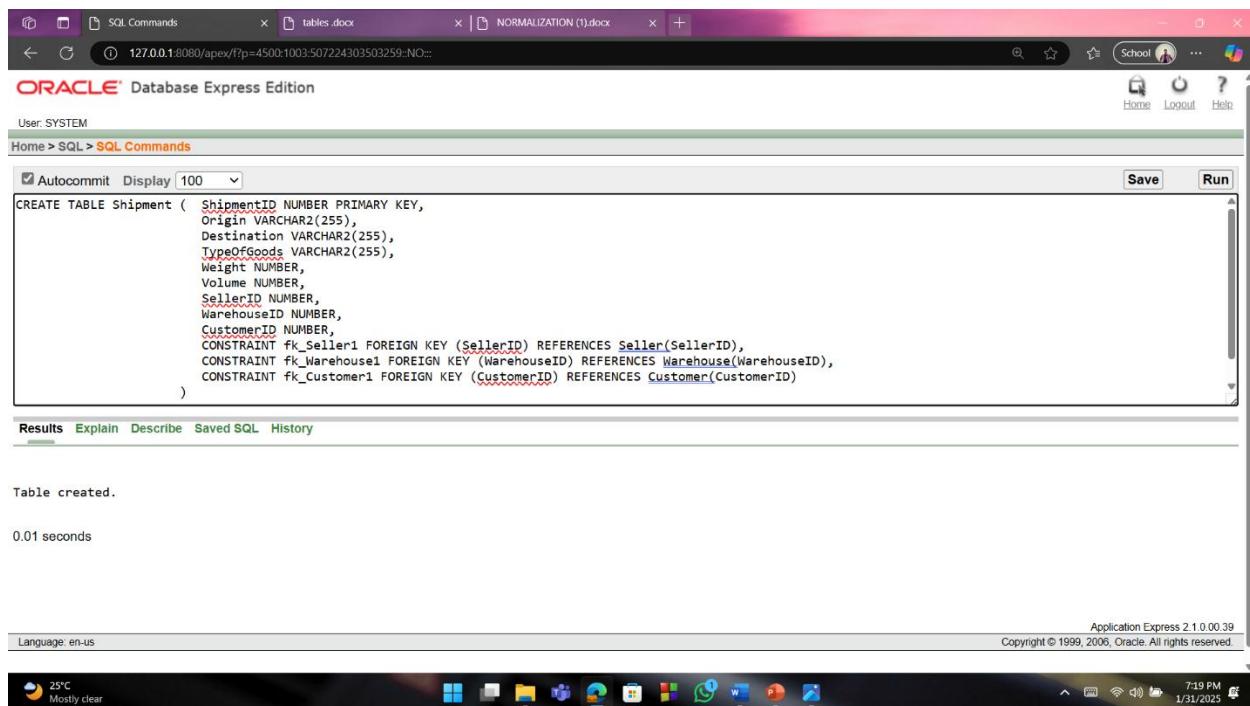
1. ShipmentID(PK), Origin, Destination, TypeOfGoods, Weight, Volume, SellerID (FK), WarehouseID(FK), CustomerID (FK).
2. DispatchOrderID(PK), DispatchDate, EstimatedArrival, ShipmentID(FK), Routes\_ID(FK), CarrierID(FK)
3. TrackingStatus, TrackingID(PK) ,ShipmentID(FK).
4. LogID (PK), Timestamp, Location, Status, ShipmentID(FK).
5. PaymentID(PK), AmountPaid, PaymentDate, PaymentMethod, ShipmentID (FK).
6. CarrierID(PK), Name, ShipmentID(FK).
7. Carrier\_Contact, Carrier\_ContactID(PK), CarrierID (FK),
8. TypeOfCarriers, TypeOfCarriersID(PK), CarrierID (FK)
9. SellerID (PK), Name, BusinessDetails, Seller\_CityID(FK),
10. Seller\_Contact, Seller\_ContactID(PK), SellerID (FK)
11. House\_Number, street\_name, city, Seller\_CityID(PK)
12. House\_Number, Street\_Name, City, Customer\_CityID(PK).
13. WarehouseID(PK), Name, Location, Capacity,
14. Warehouse\_Contact, warehouse\_ContactID(PK), WarehouseID(FK),
15. CustomerID (PK), Name, Business, Customer\_CityID(FK).
16. Customer\_Contact, Customer\_ContactID(PK), CustomerID (FK)
17. CostingID (PK) , FreightCharges , Handling Fees, ShipmentID(FK)
18. Routes\_ID(PK), Origin, Destination, Distance, EstimatedTime
19. SellerContractsId(pk), EndDate, Startdate, SellerID (FK), Seller\_CityID(FK).

## Table Creation (DDL Operations)

StudentID1:23-51676-2  Name: MOHAMMAD ALI MASUD	StudentID3: 23-51687-2  Name: TAWAFIQUR RAHMAN MAHIR
StudentID2: 23-51719-2  Name: SHAKILA SULTANA ESHITA	StudentID4: 23-51706-2  Name: MD. MAHABUBUR RAHAMAN SIAM
<b>CO4:</b> Creating DML, DDL using Oracle and connection with ODBC/JDBC for existing JAVA application	
<b>PO-e-2:</b> Use modern engineering and IT tools for prediction and modeling of complex computer science and engineering problem	Marks

## Table Creation :

### SHIPMENT TABLE



The screenshot shows the Oracle Database Express Edition interface. The SQL Commands tab is active, displaying the following DDL code:

```

CREATE TABLE Shipment (
    ShipmentID NUMBER PRIMARY KEY,
    Origin VARCHAR2(255),
    Destination VARCHAR2(255),
    TypeOfGoods VARCHAR2(255),
    Weight NUMBER,
    Volume NUMBER,
    SellerID NUMBER,
    WarehouseID NUMBER,
    CustomerID NUMBER,
    CONSTRAINT fk_Seller1 FOREIGN KEY (SellerID) REFERENCES Seller(SellerID),
    CONSTRAINT fk_Warehouse1 FOREIGN KEY (WarehouseID) REFERENCES Warehouse(WarehouseID),
    CONSTRAINT fk_Customer1 FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
)

```

Below the code, the results show "Table created." and "0.01 seconds". At the bottom, the status bar indicates "Language: en-us" and "Application Express 2.1.0.00.39". The system tray shows a weather icon for 25°C and mostly clear skies.

SQL Commands    tables.docx    NORMALIZATION (1).docx

Autocommit Display 100 Save Run

```
desc Shipment
```

Results Explain Describe Saved SQL History

Object Type TABLE Object SHIPMENT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SHIPMENT	SHIPMENTID	Number	-	-	-	1	-	-	-
	ORIGIN	VARCHAR2	255	-	-	-	✓	-	-
	DESTINATION	VARCHAR2	255	-	-	-	✓	-	-
	TYPEOFGOODS	VARCHAR2	255	-	-	-	✓	-	-
	WEIGHT	NUMBER	-	-	-	-	✓	-	-
	VOLUME	NUMBER	-	-	-	-	✓	-	-
	SELLERID	NUMBER	-	-	-	-	✓	-	-
	WAREHOUSEID	NUMBER	-	-	-	-	✓	-	-
	CUSTOMERID	NUMBER	-	-	-	-	✓	-	-

1 - 9

Language: en-us Application Express 2.1.0.00.39  
25°C Mostly clear Copyright © 1999, 2006, Oracle. All rights reserved. 7:20 PM 1/31/2025

## DISPATCH ORDER TABLE

127.0.0.1:8080/apex/f?p=4500:100:507224303503259::NO::

ORACLE Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 100 Save Run

```
CREATE TABLE DispatchOrder ( DispatchOrderID NUMBER PRIMARY KEY,
                             DispatchDate DATE,
                             EstimatedArrival DATE,
                             ShipmentID NUMBER,
                             Routes_ID NUMBER,
                             CarrierID NUMBER,
                             CONSTRAINT fk_Dispatch FOREIGN KEY (ShipmentID) REFERENCES Shipment(ShipmentID),
                             CONSTRAINT fk_Routes FOREIGN KEY (Routes_ID) REFERENCES Routes(Routes_ID),
                             CONSTRAINT fk_Carrier FOREIGN KEY (CarrierID) REFERENCES Carrier(CarrierID)
                           )
```

Results Explain Describe Saved SQL History

Table created.  
0.04 seconds

Language: en-us Application Express 2.1.0.00.39  
25°C Mostly clear Copyright © 1999, 2006, Oracle. All rights reserved. 7:28 PM 1/31/2025

127.0.0.1:8080/apex/f?p=4500:1003:507224303503259::NO::

**ORACLE® Database Express Edition**

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
desc DispatchOrder |
```

Results Explain Describe Saved SQL History

Object Type TABLE Object DISPATCHORDER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DISPATCHORDER	DISPATCHORDERID	Number	-	-	-	1	-	-	-
	DISPATCHDATE	Date	7	-	-	-	✓	-	-
	ESTIMATEDARRIVAL	Date	7	-	-	-	✓	-	-
	SHIPMENTID	Number	-	-	-	-	✓	-	-
	ROUTES_ID	Number	-	-	-	-	✓	-	-
	CARRIERID	Number	-	-	-	-	✓	-	-

1 - 6

25°C Mostly clear Application Express 2.1.0.0.39 7:29 PM 1/31/2025

## TRACKING TABLE

127.0.0.1:8080/apex/f?p=4500:1003:507224303503259::NO::

**ORACLE® Database Express Edition**

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
CREATE TABLE Tracking (
    TrackingID NUMBER PRIMARY KEY,
    TrackingStatus VARCHAR2(255),
    ShipmentID NUMBER,
    CONSTRAINT fk_Tracking_Shipment FOREIGN KEY (ShipmentID) REFERENCES Shipment(ShipmentID)
)
```

Results Explain Describe Saved SQL History

Table created.

0.01 seconds

Language: en-us Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

25°C Mostly clear 7:33 PM 1/31/2025

127.0.0.1:8080/apex/f?p=4500:1003:507224303503259::NO::

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit Display 100 Save Run

DESC Tracking

Results Explain Describe Saved SQL History

Object Type TABLE Object TRACKING

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TRACKING	TRACKINGID	Number	-	-	-	1	-	-	-
	TRACKINGSTATUS	VARCHAR2	255	-	-	-	✓	-	-
	SHIPMENTID	Number	-	-	-	-	✓	-	-

1 - 3

Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

25°C Mostly clear 7:33 PM 1/31/2025

## LOG TABLE

127.0.0.1:8080/apex/f?p=4500:1003:507224303503259::NO::

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit Display 100 Save Run

```
CREATE TABLE Log ( LogID NUMBER PRIMARY KEY,
Timestamp TIMESTAMP,
Location VARCHAR2(255),
Status VARCHAR2(255),
ShipmentID NUMBER,
CONSTRAINT fk_Log_Shipment FOREIGN KEY (ShipmentID) REFERENCES Shipment(ShipmentID)
)
```

Results Explain Describe Saved SQL History

Table created.

0.03 seconds

Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

25°C Mostly clear 7:36 PM 1/31/2025

← ⏪ 127.0.0.1:8080/apex/?p=4500:1003:507224303503259::NO::

**ORACLE® Database Express Edition**

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 100

DESC Log

Results Explain Describe Saved SQL History

Object Type TABLE Object LOG

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
LOG	LOGID	Number	-	-	-	1	-	-	-
	TIMESTAMP	Timestamp(6)	11	-	6	-	✓	-	-
	LOCATION	Varchar2	255	-	-	-	✓	-	-
	STATUS	Varchar2	255	-	-	-	✓	-	-
	SHIPMENTID	Number	-	-	-	-	✓	-	-

1 - 5

Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us 25°C Mostly clear 7:36 PM 1/31/2025

## PAYMENT TABLE

← ⏪ 127.0.0.1:8080/apex/?p=4500:1003:507224303503259::NO::

**ORACLE® Database Express Edition**

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 100

```
CREATE TABLE Payment (
    PaymentID NUMBER PRIMARY KEY,
    AmountPaid NUMBER,
    PaymentDate DATE,
    PaymentMethod VARCHAR2(255),
    ShipmentID NUMBER,
    CONSTRAINT fk_Payment_Shipment FOREIGN KEY (ShipmentID) REFERENCES Shipment(ShipmentID)
)
```

Results Explain Describe Saved SQL History

Table created.

0.02 seconds

Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us 25°C Mostly clear 7:40 PM 1/31/2025

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 100 Save Run

DESC Payment

Results Explain Describe Saved SQL History

Object Type TABLE Object PAYMENT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PAYMENT	PAYMENTID	Number	-	-	-	1	-	-	-
	AMOUNTPAID	Number	-	-	-	-	✓	-	-
	PAYMENTDATE	Date	7	-	-	-	✓	-	-
	PAYMENTMETHOD	Varchar2	255	-	-	-	✓	-	-
	SHIPMENTID	Number	-	-	-	-	✓	-	-

1 - 5

Language: en-us

25°C Mostly clear

Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.  
1/31/2025 7:40 PM

**Introduction to Database (2108): Semester**

## CARRIER TABLE

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is entered:

```
CREATE TABLE Carrier (
    CarrierID NUMBER PRIMARY KEY,
    Name VARCHAR2(255) NOT NULL,
    ShipmentID NUMBER NOT NULL,
    CONSTRAINT fk_carrier_shipment FOREIGN KEY (ShipmentID) REFERENCES Shipment(ShipmentID)
)
```

The code is highlighted with red underlines indicating syntax errors. The 'Run' button is visible at the top right of the editor.

Results Explain Describe Saved SQL History

Table created.

0.02 seconds

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is entered:

```
desc Carrier
```

The results show the table structure:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CARRIER	CARRIERID	Number	-	-	-	1	-	-	-
	NAME	Varchar2	255	-	-	-	-	-	-
	SHIPMENTID	Number	-	-	-	-	-	-	-

Below the table, a note indicates "1 - 3".

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

## CARRIER CONTACT TABLE

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is entered:

```
CREATE TABLE CarrierContact (
    Carrier_ContactID NUMBER PRIMARY KEY,
    Carrier_Contact VARCHAR2(255),
    CarrierID NUMBER,
    CONSTRAINT fk_Carrier_Contact FOREIGN KEY (CarrierID) REFERENCES Carrier(CarrierID)
)
```

Below the code, the results show:

Table created.

0.02 seconds

Language: en-us Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is entered:

```
DESC CarrierContact
```

Below the code, the results show:

Object Type TABLE Object CARRIERCONTACT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CARRIERCONTACT	CARRIER_CONTACTID	Number	-	-	-	1	-	-	-
	CARRIER_CONTACT	Varchar2	255	-	-	-	✓	-	-
	CARRIERID	Number	-	-	-	-	✓	-	-

1 - 3

Language: en-us Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

## TYPE OF CARRIERS

SQL Commands    tables.docx    NORMALIZATION (1).docx    (1763) Jontrona | Tanveer E...    +

User: SYSTEM

ORACLE Database Express Edition

Home > SQL > SQL Commands

Autocommit  Display 100  Save  Run

```
CREATE TABLE TypeOfCarriers (
    TypeOfCarriersID NUMBER PRIMARY KEY,
    TypeOfCarriers VARCHAR2 (255),
    CarrierID NUMBER,
    CONSTRAINT fk_TypeOfCarriers FOREIGN KEY (CarrierID) REFERENCES Carrier(CarrierID)
)
```

Results Explain Describe Saved SQL History

Table created.

0.02 seconds



SQL Commands    tables.docx    NORMALIZATION (1).docx    (1763) Jontrona | Tanveer E...    +

User: SYSTEM

ORACLE Database Express Edition

Home > SQL > SQL Commands

Autocommit  Display 100  Save  Run

```
desc TypeOfCarriers
```

Results Explain Describe Saved SQL History

Object Type TABLE Object TYPEOFCARRIERS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TYPEOFCARRIERS	TYPEOFCARRIERSID	Number	-	-	-	1	-	-	-
TYPEOFCARRIERS	TYPEOFCARRIERS	Varchar2	255	-	-	-	✓	-	-
CARRIERID	CARRIERID	Number	-	-	-	-	✓	-	-

1 - 3



## SELLER TABLE

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
CREATE TABLE Seller (
    SellerID NUMBER PRIMARY KEY,
    Name VARCHAR2(255),
    BusinessDetails VARCHAR2(255),
    Seller_CityID NUMBER,
    CONSTRAINT fk_Seller_City_Address FOREIGN KEY (Seller_CityID) REFERENCES Seller_Address(Seller_CityID)
)
```

Results Explain Describe Saved SQL History

Table created.

0.06 seconds

Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
desc Seller
```

Results Explain Describe Saved SQL History

Object Type: TABLE Object: SELLER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SELLER	SELLERID	Number	-	-	-	1	-	-	
	NAME	VARCHAR2	255	-	-	-	✓	-	-
	BUSINESSDETAILS	VARCHAR2	255	-	-	-	✓	-	-
	SELLER_CITYID	Number	-	-	-	-	✓	-	-

1 - 4

Language: en-us Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

## SELLER CONTACT

SQL Commands    tables.docx    NORMALIZATION (1).docx

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
CREATE TABLE SellerContact (
    Seller_ContactID NUMBER PRIMARY KEY,
    Seller_Contact VARCHAR2(255),
    SellerID NUMBER,
    CONSTRAINT fk_Seller_Contact FOREIGN KEY (SellerID) REFERENCES Seller(SellerID)
)
```

Results Explain Describe Saved SQL History

Table created.

0.02 seconds

Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

25°C Mostly clear

SQL Commands    NORMALIZATION (1).docx    tables.docx    (1763) Coke Studio Season

ORACLE Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit

```
desc SellerContact
```

Results Explain Describe Saved SQL History

Object Type: TABLE Object: SELLERCONTACT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SELLERCONTACT	Seller_ContactID	Number	-	-	-	1	-	-	-
	SELLER_CONTACT	Varchar2	255	-	-	-	✓	-	-
	SELLERID	Number	-	-	-	-	✓	-	-

1 - 3

Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.



## SELLER ADDRESS

SQL Commands    NORMALIZATION (1).docx

ORACLE Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

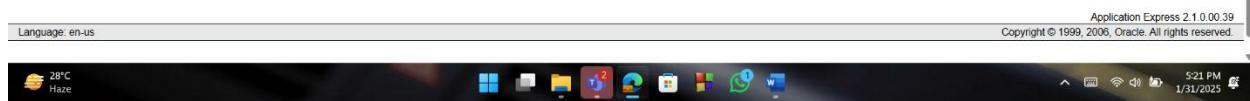
Autocommit

```
CREATE TABLE Seller_Address (Seller_CityID NUMBER PRIMARY KEY,
House_Number VARCHAR2(10) NOT NULL,
Street_Name VARCHAR2(100) NOT NULL,
City VARCHAR2(50) NOT NULL
)
```

Results Explain Describe Saved SQL History

Table created.

0.18 seconds



SQL Commands    NORMALIZATION (1).docx    tables.docx

ORACLE Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

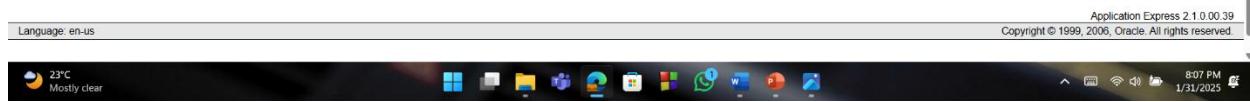
Autocommit

```
desc Seller_Address
```

Results Explain Describe Saved SQL History

Object Type: TABLE Object: SELLER\_ADDRESS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SELLER_ADDRESS	SELLER_CITYID	Number	-	-	-	1	-	-	-
	HOUSE_NUMBER	Varchar2	10	-	-	-	-	-	-
	STREET_NAME	Varchar2	100	-	-	-	-	-	-
	CITY	Varchar2	50	-	-	-	-	-	-
1 - 4									



## CUSTOMER ADDRESS

SQL Commands tables.docx

ORACLE Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit

```
CREATE TABLE CustomerAddress ( Customer_CityID NUMBER PRIMARY KEY,
House_Number VARCHAR2(255),
Street_Name VARCHAR2(255),
City VARCHAR2(255)
)
```

Results Explain Describe Saved SQL History

Table created.

0.03 seconds



SQL Commands NORMALIZATION (1).docx tables.docx

ORACLE Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit

```
desc CustomerAddress
```

Results Explain Describe Saved SQL History

Object Type: TABLE Object: CUSTOMERADDRESS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMERADDRESS	CUSTOMER_CITYID	Number	-	-	-	1	-	-	-
	HOUSE_NUMBER	Varchar2	255	-	-	-	✓	-	-
	STREET_NAME	Varchar2	255	-	-	-	✓	-	-
	CITY	Varchar2	255	-	-	-	✓	-	-
1 - 4									

Language: en-us Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.



## WAREHOUSE TABLE

The screenshot shows the Oracle Database Express Edition interface. The title bar says "SQL Commands". The URL is "127.0.0.1:8080/apex/f?p=4500:1003:507224303503259::NO::". The user is "SYSTEM". The page shows the SQL command to create the WAREHOUSE table:

```
CREATE TABLE Warehouse ( WarehouseID NUMBER PRIMARY KEY,
    Name      VARCHAR2(255) NOT NULL,
    Location  VARCHAR2(255) NOT NULL,
    Capacity   NUMBER CHECK (Capacity > 0)
)
```

Buttons at the top right include "Save" and "Run". Below the code, the status message "Table created." is displayed.

Table created.

0.06 seconds

The screenshot shows the Oracle Database Express Edition interface. The title bar says "SQL Commands". The URL is "127.0.0.1:8080/apex/f?p=4500:1003:507224303503259::NO::". The user is "SYSTEM". The page shows the SQL command to describe the WAREHOUSE table:

```
desc Warehouse
```

Buttons at the top right include "Save" and "Run". Below the command, the status message "Object Type TABLE Object WAREHOUSE" is displayed. A table below shows the columns and their properties:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
WAREHOUSE	WAREHOUSEID	Number	-	-	-	1	-	-	-
	NAME	Varchar2	255	-	-	-	-	-	-
	LOCATION	Varchar2	255	-	-	-	-	-	-
	CAPACITY	Number	-	-	-	-	✓	-	-

At the bottom, it says "1 - 4". The status bar at the bottom right shows "Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved."

## WAREHOUSE CONTACT

127.0.0.1:8080/apex/f?p=4500:1003:507224303503259::NO::

**ORACLE Database Express Edition**

User SYSTEM

Home > SQL > SQL Commands

Autocommit Display 100

```
CREATE TABLE WarehouseContact (
    warehouse_ContactID NUMBER PRIMARY KEY,
    warehouse_Contact VARCHAR2(255),
    WarehouseID NUMBER,
    CONSTRAINT fk_Warehouse FOREIGN KEY (WarehouseID) REFERENCES Warehouse(WarehouseID)
)
```

Results Explain Describe Saved SQL History

Table created.

0.01 seconds

Application Express 2.1.0.00.39  
Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved.

25°C Haze 6:37 PM 1/31/2025

127.0.0.1:8080/apex/f?p=4500:1003:507224303503259::NO::

**ORACLE Database Express Edition**

User SYSTEM

Home > SQL > SQL Commands

Autocommit Display 100

```
DESC WarehouseContact
```

Results Explain Describe Saved SQL History

Object Type: TABLE Object: WAREHOUSECONTACT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
WAREHOUSECONTACT	WAREHOUSE_CONTACTID	Number	-	-	-	1	-	-	
	WAREHOUSE_CONTACT	Varchar2	255	-	-	-	✓	-	-
	WAREHOUSEID	Number	-	-	-	-	✓	-	-

1 - 3

Application Express 2.1.0.00.39  
Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved.

25°C Haze 6:38 PM 1/31/2025

## CUSTOMER TABLE

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands panel, the following SQL code is entered:

```
CREATE TABLE Customer (
    CustomerID NUMBER PRIMARY KEY,
    Name VARCHAR2(255) NOT NULL,
    Business VARCHAR2(255),
    Customer_CityID NUMBER NOT NULL,
    CONSTRAINT fk_customer_city FOREIGN KEY (Customer_CityID) REFERENCES CustomerAddress(Customer_CityID)
)
```

Below the code, the results show "Table created." and a execution time of "0.04 seconds".

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands panel, the following SQL code is entered:

```
DESC Customer
```

Below the code, the results show the description of the CUSTOMER table, including columns: CUSTOMERID, NAME, BUSINESS, and CUSTOMER\_CITYID.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER	CUSTOMERID	Number	-	-	-	1	-	-	-
	NAME	Varchar2	255	-	-	-	-	-	-
	BUSINESS	Varchar2	255	-	-	-	✓	-	-
	CUSTOMER_CITYID	Number	-	-	-	-	-	-	-

At the bottom, the results show "Object Type TABLE Object CUSTOMER" and a page number "1 - 4".

## CUSTOMER CONTACT TABLE

SQL Commands NORMALIZATION (1).docx tables.docx (1763) LYRICAL: Kaise Mujhe

ORACLE Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit

```
CREATE TABLE CustomerContact (
    Customer_ContactID NUMBER PRIMARY KEY,
    Customer_Contact VARCHAR2(255),
    CustomerID NUMBER,
    CONSTRAINT fk_Customer_Contact FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
)
```

Results Explain Describe Saved SQL History

Table created.

0.03 seconds

Language: en-us Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

SQL Commands NORMALIZATION (1).docx tables.docx (1763) LYRICAL: Kaise Mujhe

ORACLE Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit

```
DESC CustomerContact
```

Results Explain Describe Saved SQL History

Object Type: TABLE Object: CUSTOMERCONTACT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMERCONTACT	CUSTOMER_CONTACTID	Number	-	-	-	1	-	-	-
	CUSTOMER_CONTACT	Varchar2	255	-	-	-	✓	-	-
	CUSTOMERID	Number	-	-	-	-	✓	-	-

1 - 3

Language: en-us Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

23°C Mostly clear 8:02 PM 1/31/2025

## COSTING TABLE

User: SYSTEM

Home > SQL > SQL Commands

Autocommit

```
CREATE TABLE Costing (
    CostingID NUMBER PRIMARY KEY,
    FreightCharges NUMBER,
    HandlingFees NUMBER,
    ShipmentID NUMBER,
    CONSTRAINT fk_Costing_Shipment FOREIGN KEY (ShipmentID) REFERENCES Shipment(ShipmentID)
)
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

Table created.

0.01 seconds

Language: en-us Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

23°C Mostly clear

Home > SQL > SQL Commands

Autocommit

desc Costing

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

Object Type: TABLE Object: COSTING

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
COSTING	COSTINGID	Number	-	-	-	1	-	-	-
	FREIGHTCHARGES	Number	-	-	-	-	✓	-	-
	HANDLINGFEES	Number	-	-	-	-	✓	-	-
	SHIPMENTID	Number	-	-	-	-	✓	-	-
						1 - 4			

Language: en-us Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

23°C Mostly clear

## ROUTES TABLE

SQL Commands    tables.docx    NORMALIZATION (1).docx

ORACLE Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit

```
CREATE TABLE Routes (
    Routes_ID NUMBER PRIMARY KEY,
    Origin VARCHAR2(255),
    Destination VARCHAR2(255),
    Distance NUMBER,
    EstimatedTime VARCHAR2(255)
)
```

Results Explain Describe Saved SQL History

Table created.

0.03 seconds



SQL Commands    NORMALIZATION (1).docx    tables.docx    (1763) Coke Studio Season

ORACLE Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit

```
desc Routes
```

Results Explain Describe Saved SQL History

Object Type TABLE Object ROUTES

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ROUTES	ROUTES_ID	Number	-	-	-	1	-	-	-
	ORIGIN	Varchar2	255	-	-	-	✓	-	-
	DESTINATION	Varchar2	255	-	-	-	✓	-	-
	DISTANCE	Number	-	-	-	-	✓	-	-
	ESTIMATEDTIME	Varchar2	255	-	-	-	✓	-	-
1 - 5									



## SELLERCONTRACTS

The screenshot shows the Oracle Database Express Edition interface. The title bar says "SELLERCONTRACTS". The main area contains the SQL command to create the SELLERCONTRACTS table:

```
CREATE TABLE SellerContracts ( SellerContractSID NUMBER PRIMARY KEY,
                               StartDate DATE,
                               EndDate DATE,
                               SellerID NUMBER,
                               Seller_CityID NUMBER,
                               CONSTRAINT fk_Seller_Contracts_Seller FOREIGN KEY (SellerID) REFERENCES Seller(SellerID),
                               CONSTRAINT fk_Seller_Contracts_SellerCity FOREIGN KEY (Seller_CityID) REFERENCES Seller_Address (Seller_CityID)
                             )
```

Below the code, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is selected.

Table created.

0.02 seconds

The screenshot shows the Oracle Database Express Edition interface. The title bar says "SELLERCONTRACTS". The main area contains the SQL command to describe the SELLERCONTRACTS table:

```
desc SellerContracts
```

Below the command, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Describe tab is selected. A table titled "Object Type TABLE Object SELLERCONTRACTS" is displayed, showing the columns and their properties:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SELLERCONTRACTS	SELLERCONTRACTSID	Number	-	-	-	1	-	-	-
	STARTDATE	Date	7	-	-	-	✓	-	-
	ENDDATE	Date	7	-	-	-	✓	-	-
	SELLERID	Number	-	-	-	-	✓	-	-
	SELLER_CITYID	Number	-	-	-	-	✓	-	-

At the bottom of the table, it says "1 - 5". Below the table, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Describe tab is selected.

# Data Insertion

## **SHIPMENT TABLE**

The screenshot shows a browser window for Oracle Database Express Edition. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:8944280080423593:NO:::1. The page title is "ORACLE Database Express Edition". The main content area contains the following SQL code:

```
INSERT INTO Shipment VALUES (1, 'Dhaka', 'Chattogram', 'Electronics', 1000, 500, 1, 1, 1);
INSERT INTO Shipment VALUES (2, 'Sylhet', 'Rajshahi', 'Furniture', 2000, 1000, 2, 2, 2);
INSERT INTO Shipment VALUES (3, 'Khulna', 'Rangpur', 'Gadgets', 1500, 800, 3, 3, 3);
INSERT INTO Shipment VALUES (4, 'Dhaka', 'Sylhet', 'Kitchenware', 1800, 900, 4, 4, 4);
INSERT INTO Shipment VALUES (5, 'Chattogram', 'Khulna', 'Clothing', 2200, 1200, 5, 5, 5);
INSERT INTO Shipment VALUES (6, 'Rajshahi', 'Dhaka', 'Groceries', 1100, 600, 6, 6, 6);
select * from Shipment
```

Below the code, there is a results table:

SHIPMENTID	ORIGIN	DESTINATION	TYPEOFGOODS	WEIGHT	VOLUME	SELLERID	WAREHOUSEID	CUSTOMERID
1	Dhaka	Chattogram	Electronics	1000	500	1	1	1
2	Sylhet	Rajshahi	Furniture	2000	1000	2	2	2
3	Khulna	Rangpur	Gadgets	1500	800	3	3	3
4	Dhaka	Sylhet	Kitchenware	1800	900	4	4	4
5	Chattogram	Khulna	Clothing	2200	1200	5	5	5
6	Rajshahi	Dhaka	Groceries	1100	600	6	6	6

At the bottom, it says "6 rows returned in 0.02 seconds" and "CSV Export". The status bar at the bottom right shows "Application Express 2.1.0.0.39", "Copyright © 1999, 2006, Oracle. All rights reserved.", "Language: en-us", "2/1/2025", and "7:54 AM".

## DISPATCH ORDER TABLE

User SYSTEM

ORACLE Database Express Edition

Home > SQL > SQL Commands

Autocommit

```
INSERT INTO DispatchOrder VALUES (1, TO_DATE('2025-01-01', 'YYYY-MM-DD'), TO_DATE('2025-01-02', 'YYYY-MM-DD'), 1, 1, 1);
INSERT INTO DispatchOrder VALUES (2, TO_DATE('2025-01-03', 'YYYY-MM-DD'), TO_DATE('2025-01-04', 'YYYY-MM-DD'), 2, 2, 2);
INSERT INTO DispatchOrder VALUES (3, TO_DATE('2025-01-05', 'YYYY-MM-DD'), TO_DATE('2025-01-06', 'YYYY-MM-DD'), 3, 3, 3);
INSERT INTO DispatchOrder VALUES (4, TO_DATE('2025-01-07', 'YYYY-MM-DD'), TO_DATE('2025-01-08', 'YYYY-MM-DD'), 4, 4, 4);
INSERT INTO DispatchOrder VALUES (5, TO_DATE('2025-01-09', 'YYYY-MM-DD'), TO_DATE('2025-01-10', 'YYYY-MM-DD'), 5, 5, 5);
INSERT INTO DispatchOrder VALUES (6, TO_DATE('2025-01-11', 'YYYY-MM-DD'), TO_DATE('2025-01-12', 'YYYY-MM-DD'), 6, 6, 6);
select * from DispatchOrder;
```

Results Explain Describe Saved SQL History

DISPATCHORDERID	DISPATCHDATE	ESTIMATEDARRIVAL	SHIPMENTID	ROUTES_ID	CARRIERID
1	01-JAN-25	02-JAN-25	1	1	1
2	03-JAN-25	04-JAN-25	2	2	2
3	05-JAN-25	06-JAN-25	3	3	3
4	07-JAN-25	08-JAN-25	4	4	4
5	09-JAN-25	10-JAN-25	5	5	5
6	11-JAN-25	12-JAN-25	6	6	6

6 rows returned in 0.01 seconds [CSV Export](#)

Application Express 2.1 0 0 39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

23°C Clear 8:04 AM 2/1/2025

## TRACKING TABLE

User SYSTEM

ORACLE Database Express Edition

Home > SQL > SQL Commands

Autocommit

```
INSERT INTO Tracking VALUES (1, 'In Transit', 1);
INSERT INTO Tracking VALUES (2, 'Delivered', 2);
INSERT INTO Tracking VALUES (3, 'Pending', 3);
INSERT INTO Tracking VALUES (4, 'In Transit', 4);
INSERT INTO Tracking VALUES (5, 'Shipped', 5);
INSERT INTO Tracking VALUES (6, 'Delivered', 6);
select * from Tracking;
```

Results Explain Describe Saved SQL History

TRACKINGID	TRACKINGSTATUS	SHIPMENTID
1	In Transit	1
2	Delivered	2
3	Pending	3
4	In Transit	4
5	Shipped	5
6	Delivered	6

6 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1 0 0 39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

23°C Clear 8:04 AM 2/1/2025

## LOG TABLE

127.0.0.1:8080/apex/f?p=4500:1003:8944280080423593::NO::

**ORACLE Database Express Edition**

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
INSERT INTO Log VALUES (1, SYSTIMESTAMP, 'New York', 'Dispatched', 1);
INSERT INTO Log VALUES (2, SYSTIMESTAMP, 'Chicago', 'In Transit', 2);
INSERT INTO Log VALUES (3, SYSTIMESTAMP, 'Houston', 'Pending', 3);
INSERT INTO Log VALUES (4, SYSTIMESTAMP, 'Dallas', 'In Transit', 4);
INSERT INTO Log VALUES (5, SYSTIMESTAMP, 'San Francisco', 'Shipped', 5);
INSERT INTO Log VALUES (6, SYSTIMESTAMP, 'Miami', 'Delivered', 6);
select * from Log
```

**Results Explain Describe Saved SQL History**

LOGID	TIMESTAMP	LOCATION	STATUS	SHIPMENTID
1	01-FEB-25 08.05.57 480000 PM	New York	Dispatched	1
2	01-FEB-25 08.06.02 394000 PM	Chicago	In Transit	2
3	01-FEB-25 08.06.07 319000 PM	Houston	Pending	3
4	01-FEB-25 08.06.12 288000 PM	Dallas	In Transit	4
5	01-FEB-25 08.06.17 077000 PM	San Francisco	Shipped	5
6	01-FEB-25 08.06.23 696000 PM	Miami	Delivered	6

6 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

23°C Clear 8:08 AM 2/1/2025 G

## PAYMENT TABLE

127.0.0.1:8080/apex/f?p=4500:1003:8944280080423593::NO::

**ORACLE Database Express Edition**

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
INSERT INTO Payment VALUES (1, 5000, TO_DATE('2025-01-07', 'YYYY-MM-DD'), 'bKash', 1);
INSERT INTO Payment VALUES (2, 10000, TO_DATE('2025-01-08', 'YYYY-MM-DD'), 'Nagad', 2);
INSERT INTO Payment VALUES (3, 7500, TO_DATE('2025-01-09', 'YYYY-MM-DD'), 'Rocket', 3);
INSERT INTO Payment VALUES (4, 9000, TO_DATE('2025-01-10', 'YYYY-MM-DD'), 'Bank Transfer', 4);
INSERT INTO Payment VALUES (5, 12000, TO_DATE('2025-01-11', 'YYYY-MM-DD'), 'Cash', 5);
INSERT INTO Payment VALUES (6, 15000, TO_DATE('2025-01-12', 'YYYY-MM-DD'), 'bKash', 6);
select * from Payment
```

**Results Explain Describe Saved SQL History**

PAYOUTMENTID	AMOUNTPAID	PAYOUTMENTDATE	PAYOUTMENTMETHOD	SHIPMENTID
1	5000	07-JAN-25	bKash	1
2	10000	08-JAN-25	Nagad	2
3	7500	09-JAN-25	Rocket	3
4	9000	10-JAN-25	Bank Transfer	4
5	12000	11-JAN-25	Cash	5
6	15000	12-JAN-25	bKash	6

6 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

23°C Clear 8:02 PM 2/1/2025 G

## CARRIER CONTACT TABLE

User: SYSTEM

Home > SQL > SQL Commands

Autocommit

```
INSERT INTO CarrierContact VALUES (1, '01711-223344', 1);
INSERT INTO CarrierContact VALUES (2, '01822-334455', 2);
INSERT INTO CarrierContact VALUES (3, '01933-445566', 3);
INSERT INTO CarrierContact VALUES (4, '01644-556677', 4);
INSERT INTO CarrierContact VALUES (5, '01555-667788', 5);
INSERT INTO CarrierContact VALUES (6, '01366-778899', 6);
select * from CarrierContact |
```

**Results** [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

CARRIER_CONTACTID	CARRIER_CONTACT	CARRIERID
1	01711-223344	1
2	01822-334455	2
3	01933-445566	3
4	01644-556677	4
5	01555-667788	5
6	01366-778899	6

6 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

## TYPE OF CARRIERS TABLE

User: SYSTEM

Home > SQL > SQL Commands

Autocommit

```
INSERT INTO TypeOfCarriers VALUES (1, 'Truck', 1);
INSERT INTO TypeOfCarriers VALUES (2, 'Cargo Van', 2);
INSERT INTO TypeOfCarriers VALUES (3, 'Refrigerated Truck', 3);
INSERT INTO TypeOfCarriers VALUES (4, 'Container Ship', 4);
INSERT INTO TypeOfCarriers VALUES (5, 'Air Freight', 5);
INSERT INTO TypeOfCarriers VALUES (6, 'Rail Transport', 6);
select * from TypeOfCarriers |
```

**Results** [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

TYPEOFCARRIERSID	TYPEOFCARRIERS	CARRIERID
1	Truck	1
2	Cargo Van	2
3	Refrigerated Truck	3
4	Container Ship	4
5	Air Freight	5
6	Rail Transport	6

6 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

## SELLER TABLE

127.0.0.1:8080/apex/f?p=4500:1003:8944280080423593::NO::

ORACLE Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
INSERT INTO Seller VALUES (1, 'Rahman Electronics', 'Electronics', 1);
INSERT INTO Seller VALUES (2, 'Khan Furnitures', 'Furniture', 2);
INSERT INTO Seller VALUES (3, 'IT Solutions BD', 'Gadgets', 3);
INSERT INTO Seller VALUES (4, 'Sadia Kitchenware', 'Kitchenware', 4);
INSERT INTO Seller VALUES (5, 'Trendy Fashion', 'Clothing', 5);
INSERT INTO Seller VALUES (6, 'Deshi Foods', 'Groceries', 6);
select * from Seller;
```

Results Explain Describe Saved SQL History

SELLERID	NAME	BUSINESSDETAILS	SELLER_CITYID
1	Rahman Electronics	Electronics	1
2	Khan Furnitures	Furniture	2
3	IT Solutions BD	Gadgets	3
4	Sadia Kitchenware	Kitchenware	4
5	Trendy Fashion	Clothing	5
6	Deshi Foods	Groceries	6

6 rows returned in 0.02 seconds CSV Export

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

24°C Clear 7:42 AM 2/1/2025

## SELLER CONTACT TABLE

127.0.0.1:8080/apex/f?p=4500:1003:8944280080423593::NO::

ORACLE Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
INSERT INTO SellerContact VALUES (1, '01712-111111', 1);
INSERT INTO SellerContact VALUES (2, '01813-222222', 2);
INSERT INTO SellerContact VALUES (3, '01914-333333', 3);
INSERT INTO SellerContact VALUES (4, '01615-444444', 4);
INSERT INTO SellerContact VALUES (5, '01516-555555', 5);
INSERT INTO SellerContact VALUES (6, '01317-666666', 6);
select * from SellerContact;
```

Results Explain Describe Saved SQL History

SELLER_CONTACTID	SELLER_CONTACT	SELLERID
1	01712-111111	1
2	01813-222222	2
3	01914-333333	3
4	01615-444444	4
5	01516-555555	5
6	01317-666666	6

6 rows returned in 0.00 seconds CSV Export

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

24°C Clear 7:52 AM 2/1/2025

## SELLER ADDRESS TABLE

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
INSERT INTO Seller_Address VALUES (1, '123A', 'Dhanmondi 32', 'Dhaka'); I
INSERT INTO Seller_Address VALUES (2, '456B', 'Baizid Bostami', 'Chattogram');
INSERT INTO Seller_Address VALUES (3, '789C', 'Zindabazar', 'Sylhet');
INSERT INTO Seller_Address VALUES (4, '101D', 'Khulna Sadar', 'Khulna');
INSERT INTO Seller_Address VALUES (5, '202E', 'Rajshahi Court', 'Rajshahi');
INSERT INTO Seller_Address VALUES (6, '303F', 'Rangpur Station Rd', 'Rangpur');
select * from Seller_Address
```

**Results** **Explain** **Describe** **Saved SQL** **History**

SELLER_CITYID	HOUSE_NUMBER	STREET_NAME	CITY
1	123A	Dhanmondi 32	Dhaka
2	456B	Baizid Bostami	Chattogram
3	789C	Zindabazar	Sylhet
4	101D	Khulna Sadar	Khulna
5	202E	Rajshahi Court	Rajshahi
6	303F	Rangpur Station Rd	Rangpur

6 rows returned in 0.02 seconds [CSV Export](#)

Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

22°C Clear 8:31 PM 2/1/2025

## CUSTOMER ADDRESS TABLE

User SCOTT

Home > SQL > SQL Commands

Autocommit

```
INSERT INTO CustomerAddress VALUES (1, '789C', 'Banani', 'Dhaka');
INSERT INTO CustomerAddress VALUES (2, '101D', 'GEC Circle', 'Chattogram');
INSERT INTO CustomerAddress VALUES (3, '202E', 'Amborkhana', 'Sylhet');
INSERT INTO CustomerAddress VALUES (4, '303F', 'Bohra', 'Khulna');
INSERT INTO CustomerAddress VALUES (5, '404G', 'Laxmipur', 'Rajshahi');
INSERT INTO CustomerAddress VALUES (6, '505H', 'Cantonment', 'Rangpur');
select * from CustomerAddress
```

**Results** **Explain** **Describe** **Saved SQL** **History**

CUSTOMER_CITYID	HOUSE_NUMBER	STREET_NAME	CITY
6	505H	Cantonment	Rangpur
1	789C	Banani	Dhaka
2	101D	GEC Circle	Chattogram
3	202E	Amborkhana	Sylhet
4	303F	Bohra	Khulna
5	404G	Laxmipur	Rajshahi

6 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

82°F Haze 1:57 PM 2/1/2025

## WAREHOUSE TABLE

User SYSTEM

Home > SQL > **SQL Commands**

Autocommit

```
INSERT INTO Warehouse VALUES (1, 'Warehouse Gulshan', 'Dhaka', 5000);
INSERT INTO Warehouse VALUES (2, 'Warehouse Agrabad', 'Chattogram', 8000);
INSERT INTO Warehouse VALUES (3, 'Warehouse Uposhohor', 'Sylhet', 6000);
INSERT INTO Warehouse VALUES (4, 'Warehouse Khalishpur', 'Khulna', 7000);
INSERT INTO Warehouse VALUES (5, 'Warehouse New Market', 'Rajshahi', 9000);
INSERT INTO Warehouse VALUES (6, 'Warehouse Station Road', 'Rangpur', 10000);
select * from Warehouse
```

**Results** [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

WAREHOUSEID	NAME	LOCATION	CAPACITY
1	Warehouse Gulshan	Dhaka	5000
2	Warehouse Agrabad	Chattogram	8000
3	Warehouse Uposhohor	Sylhet	6000
4	Warehouse Khalishpur	Khulna	7000
5	Warehouse New Market	Rajshahi	9000
6	Warehouse Station Road	Rangpur	10000

6 rows returned in 0.02 seconds [CSV Export](#)

Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

22°C Clear 8:28 AM 2/1/2025

## WAREHOUSE CONTACT TABLE

User SYSTEM

Home > SQL > **SQL Commands**

Autocommit

```
INSERT INTO WarehouseContact VALUES (1, '01711-123456', 1);
INSERT INTO WarehouseContact VALUES (2, '01822-234567', 2);
INSERT INTO WarehouseContact VALUES (3, '01933-345678', 3);
INSERT INTO WarehouseContact VALUES (4, '01644-456789', 4);
INSERT INTO WarehouseContact VALUES (5, '01555-567890', 5);
INSERT INTO WarehouseContact VALUES (6, '01366-678901', 6);
select * from emp
```

**Results** [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

WAREHOUSE_CONTACTID	WAREHOUSE_CONTACT	WAREHOUSEID
1	01711-123456	1
2	01822-234567	2
3	01933-345678	3
4	01644-456789	4
5	01555-567890	5
6	01366-678901	6

6 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

24°C Clear 7:34 AM 2/1/2025

## CUSTOMER TABLE

127.0.0.1:8080/apex/f?p=4500:1003:8944280080423593::NO:-

**ORACLE Database Express Edition**

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
INSERT INTO Customer VALUES (1, 'Md. Rafiq Islam', 'Retail', 1);
INSERT INTO Customer VALUES (2, 'Nasima Akter', 'Wholesale', 2);
INSERT INTO Customer VALUES (3, 'Shahidul Haque', 'E-commerce', 3);
INSERT INTO Customer VALUES (4, 'Farhana Rahman', 'Manufacturing', 4);
INSERT INTO Customer VALUES (5, 'Kamal Uddin', 'Construction', 5);
INSERT INTO Customer VALUES (6, 'Jannatul Ferdous', 'Fashion', 6);
select * from Customer
```

Results Explain Describe Saved SQL History

CUSTOMERID	NAME	BUSINESS	CUSTOMER_CITYID
1	Md. Rafiq Islam	Retail	1
2	Nasima Akter	Wholesale	2
3	Shahidul Haque	E-commerce	3
4	Farhana Rahman	Manufacturing	4
5	Kamal Uddin	Construction	5
6	Jannatul Ferdous	Fashion	6

6 rows returned in 0.00 seconds CSV Export

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

24°C Clear 7:38 PM 2/1/2025

## ROUTES TABLE

127.0.0.1:8080/apex/f?p=4500:1003:8944280080423593::NO:-

**ORACLE Database Express Edition**

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
INSERT INTO Routes VALUES (1, 'Dhaka', 'Chattogram', 280, '8 Hours');
INSERT INTO Routes VALUES (2, 'Sylhet', 'Rajshahi', 350, '10 Hours');
INSERT INTO Routes VALUES (3, 'Khulna', 'Rangpur', 450, '12 Hours');
INSERT INTO Routes VALUES (4, 'Dhaka', 'Sylhet', 280, '7 Hours');
INSERT INTO Routes VALUES (5, 'Chattogram', 'Khulna', 400, '11 Hours');
INSERT INTO Routes VALUES (6, 'Rajshahi', 'Dhaka', 250, '6 Hours');
select * from Routes |
```

Results Explain Describe Saved SQL History

ROUTES_ID	ORIGIN	DESTINATION	DISTANCE	ESTIMATEDTIME
1	Dhaka	Chattogram	280	8 Hours
2	Sylhet	Rajshahi	350	10 Hours
3	Khulna	Rangpur	450	12 Hours
4	Dhaka	Sylhet	280	7 Hours
5	Chattogram	Khulna	400	11 Hours
6	Rajshahi	Dhaka	250	6 Hours

6 rows returned in 0.02 seconds CSV Export

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

24°C Clear 7:40 PM 2/1/2025

## CUSTOMER CONTACT TABLE

127.0.0.1:8080/apex/f?p=4500:1003:8944280080423593::NO::

**ORACLE Database Express Edition**

User: SYSTEM

Home > SQL > SQL Commands

Autocommit

```
INSERT INTO CustomerContact VALUES (1, '01711-987654', 1);
INSERT INTO CustomerContact VALUES (2, '01822-876543', 2);
INSERT INTO CustomerContact VALUES (3, '01933-765432', 3);
INSERT INTO CustomerContact VALUES (4, '01644-654321', 4);
INSERT INTO CustomerContact VALUES (5, '01555-543210', 5);
INSERT INTO CustomerContact VALUES (6, '01366-432109', 6);
select * from CustomerContact
```

**Results Explain Describe Saved SQL History**

CUSTOMER_CONTACTID	CUSTOMER_CONTACT	CUSTOMERID
1	01711-987654	1
2	01822-876543	2
3	01933-765432	3
4	01644-654321	4
5	01555-543210	5
6	01366-432109	6

6 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved.

23°C Clear 8:18 AM 2/1/2025

## COSTING TABLE

127.0.0.1:8080/apex/f?p=4500:1003:8944280080423593::NO::

**ORACLE Database Express Edition**

User: SYSTEM

Home > SQL > SQL Commands

Autocommit

```
INSERT INTO Costing VALUES (1, 5000, 200, 1);
INSERT INTO Costing VALUES (2, 10000, 300, 2);
INSERT INTO Costing VALUES (3, 7500, 250, 3);
INSERT INTO Costing VALUES (4, 9000, 400, 4);
INSERT INTO Costing VALUES (5, 12000, 350, 5);
INSERT INTO Costing VALUES (6, 15000, 500, 6);
select * from Costing
```

**Results Explain Describe Saved SQL History**

COSTINGID	FREIGHTCHARGES	HANDLINGFEES	SHIPMENTID
1	5000	200	1
2	10000	300	2
3	7500	250	3
4	9000	400	4
5	12000	350	5
6	15000	500	6

6 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved.

23°C Clear 8:21 AM 2/1/2025

## CARRIER TABLE

127.0.0.1:8080/apex/?p=4500:1003:8944280080423593:No:

**ORACLE Database Express Edition**

User: SYSTEM

Home > SQL > SQL Commands

Autocommit

```
INSERT INTO Carrier VALUES (1, 'Bangla Logistics', 1);
INSERT INTO Carrier VALUES (2, 'Deshi Transport', 2);
INSERT INTO Carrier VALUES (3, 'Fast Movers BD', 3);
INSERT INTO Carrier VALUES (4, 'Reliable Cargo', 4);
INSERT INTO Carrier VALUES (5, 'Quick Express', 5);
INSERT INTO Carrier VALUES (6, 'Trust Move BD', 6);
select * from Carrier;
```

**Results Explain Describe Saved SQL History**

CARRIERID	NAME	SHIPMENTID
1	Bangla Logistics	1
2	Deshi Transport	2
3	Fast Movers BD	3
4	Reliable Cargo	4
5	Quick Express	5
6	Trust Move BD	6

6 rows returned in 0.01 seconds

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

23°C Clear 7:55 AM 2/1/2025

## SELLER CONTRACTS TABLE

127.0.0.1:8080/apex/?p=4500:1003:8944280080423593:No:

**ORACLE Database Express Edition**

User: SYSTEM

Home > SQL > SQL Commands

Autocommit

```
INSERT INTO SellerContracts VALUES (1, TO_DATE('2024-01-01', 'YYYY-MM-DD'), TO_DATE('2025-01-01', 'YYYY-MM-DD'), 1, 1);
INSERT INTO SellerContracts VALUES (2, TO_DATE('2024-02-15', 'YYYY-MM-DD'), TO_DATE('2025-02-15', 'YYYY-MM-DD'), 2, 2);
INSERT INTO SellerContracts VALUES (3, TO_DATE('2024-03-10', 'YYYY-MM-DD'), TO_DATE('2025-03-10', 'YYYY-MM-DD'), 3, 3);
INSERT INTO SellerContracts VALUES (4, TO_DATE('2024-04-05', 'YYYY-MM-DD'), TO_DATE('2025-04-05', 'YYYY-MM-DD'), 4, 4);
INSERT INTO SellerContracts VALUES (5, TO_DATE('2024-05-20', 'YYYY-MM-DD'), TO_DATE('2025-05-20', 'YYYY-MM-DD'), 5, 5);
INSERT INTO SellerContracts VALUES (6, TO_DATE('2024-06-30', 'YYYY-MM-DD'), TO_DATE('2025-06-30', 'YYYY-MM-DD'), 6, 6);
select * from SellerContracts;
```

**Results Explain Describe Saved SQL History**

SELLERCONTRACTSID	STARTDATE	ENDDATE	SELLERID	SELLER_CITYID
1	01-JAN-24	01-JAN-25	1	1
2	15-FEB-24	15-FEB-25	2	2
3	10-MAR-24	10-MAR-25	3	3
4	05-APR-24	05-APR-25	4	4
5	20-MAY-24	20-MAY-25	5	5
6	30-JUN-24	30-JUN-25	6	6

6 rows returned in 0.00 seconds

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

23°C Clear 8:20 AM 2/1/2025

## Query Test

### a. SIMPLE QUERY

Q1. Write a query to display the seller id ,type of goods , origin and destination from shipment where weight is greater than 1500.

The screenshot shows the Oracle Database Express Edition interface. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:1974211415186660::NO--. The user is SYSTEM. The SQL Commands window contains the following query:

```
select sellerid, typeofgoods, origin, destination
from shipment
where weight>1500
```

The results window displays the following table:

SELLERID	TYPEOFGOODS	ORIGIN	DESTINATION
2	Furniture	Sylhet	Rajshahi
4	Kitchenware	Dhaka	Sylhet
5	Clothing	Chattogram	Khulna

3 rows returned in 0.02 seconds CSV Export

Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

### b. Query with a single row function

Q2. Display the payment ID and payment method in upper case from payment table where paid amount is between 7500 to 12000. Level the payment id as ID and payment method as Payment.

127.0.0.1:8080/appex/?p=4500:1003:1974711415186660..NO::

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
select paymentid as ID,upper(paymentmethod) as Payment from payment where amountpaid between 7500 and 12000
```

Results Explain Describe Saved SQL History

ID	PAYMENT
2	NACAD
3	ROCKET
4	BANK TRANSFER
5	CASH

4 rows returned in 0.00 seconds

Language: en-us Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

20°C Haze

### c.Query with a Multiple row function/ aggregate function

Q3.write a query to show the total freight charge and average of handing fees from costing table.  
label the total freight charge as total\_charges and avarage of handlingfees as Average\_fees

#### d.2 Single row subquery and 2 multiple row subquery

Q4.Find the payment ID and payment date from payment that have the same amount paid as the shipment with Shipment ID 4.

The screenshot shows the Oracle Database Express Edition interface. The SQL command entered is:

```
select paymentid, paymentdate from payment where amountpaid=(select amountpaid from payment where shipmentid=4)
```

The results show one row:

PAYMENTID	PAYMENTDATE
4	10-JAN-25

1 rows returned in 0.00 seconds

Q5.write the subquery to print the names of all warehouses that have the same capacity as the warehouse with WAREHOUSEID = 2.

The screenshot shows the Oracle Database Express Edition interface. The SQL command entered is:

```
SELECT name FROM warehouse WHERE capacity = (SELECT capacity FROM warehouse where warehouseid=2)
```

The results show one row:

NAME
Warehouse Agrabad

1 rows returned in 0.00 seconds

Q6.Display all from payment table where the payment method was used exactly twice in the PAYMENT table."

127.0.0.1:8080/apex/f?p=4500:1003:1974211415186660:NO:-

**ORACLE Database Express Edition**

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT * FROM PAYMENT
WHERE PAYMENTMETHOD IN (SELECT PAYMENTMETHOD FROM PAYMENT GROUP BY PAYMENTMETHOD HAVING COUNT(*) =2);
```

**Results Explain Describe Saved SQL History**

PAYMENTID	AMOUNTPAID	PAYMENTDATE	PAYMENTMETHOD	SHIPMENTID
1	5000	07-JAN-25	bKash	1
6	15000	12-JAN-25	bKash	6

2 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.00.39  
Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved.

Q7. Display all payments where the amount paid is greater than at least one payment made between '09-JAN-25' and '11-JAN-25'.

127.0.0.1:8080/apex/f?p=4500:1003:1974211415186660:NO:-

**ORACLE Database Express Edition**

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT * FROM PAYMENT WHERE AMOUNTPAID > ANY (SELECT AMOUNTPAID FROM PAYMENT WHERE paymentdate between '09-JAN-25' and '11-JAN-25');
```

**Results Explain Describe Saved SQL History**

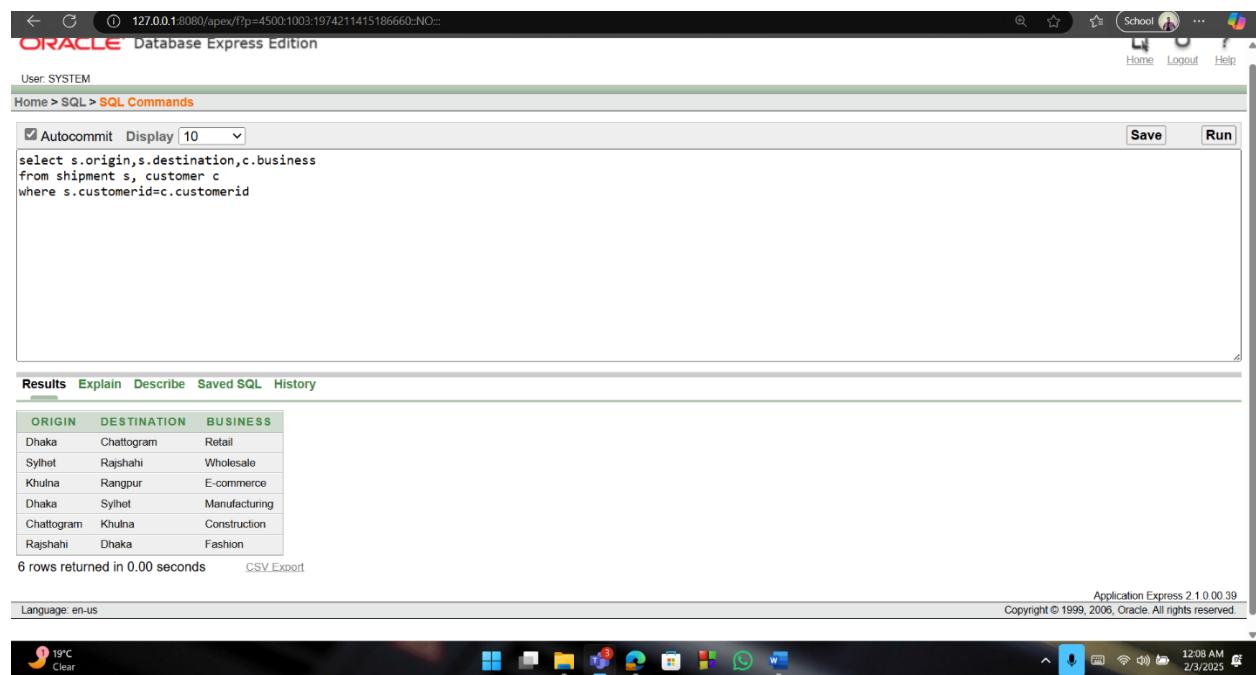
PAYMENTID	AMOUNTPAID	PAYMENTDATE	PAYMENTMETHOD	SHIPMENTID
2	10000	08-JAN-25	Nagad	2
4	9000	10-JAN-25	Bank Transfer	4
5	12000	11-JAN-25	Cash	5
6	15000	12-JAN-25	bKash	6

4 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.00.39  
Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved.

## e.4 kinds of joining

### Equijoin:



The screenshot shows a web-based Oracle Database Express Edition interface. The URL in the address bar is 127.0.0.1:8080/apev/?p=4500:1003:1974211415186660:NO:. The page title is "ORACLE Database Express Edition". The user is logged in as "SYSTEM". The navigation bar includes links for Home, Logout, and Help. The main content area shows an SQL command being run:

```
select s.origin,s.destination,c.business  
from shipment s, customer c  
where s.customerid=c.customerid
```

The "Run" button is highlighted. Below the command, the results are displayed in a table:

ORIGIN	DESTINATION	BUSINESS
Dhaka	Chattogram	Retail
Sylhet	Rajshahi	Wholesale
Khulna	Rangpur	E-commerce
Dhaka	Sylhet	Manufacturing
Chattogram	Khulna	Construction
Rajshahi	Dhaka	Fashion

Below the table, it says "6 rows returned in 0.00 seconds" and there is a "CSV Export" link. At the bottom, it shows the system status with a weather icon (19°C Clear), a taskbar with various application icons, and a system tray with network, battery, and time indicators (12:08 AM, 2/3/2025).

### Self join:

The screenshot shows a SQL query being run in Oracle SQL Developer. The query is:

```
select s1.shipmentid, s1.weight, s2.typeofgoods, s2.weight  
from shipment s1,shipment s2  
where s1.weight between s2.weight-500 and s2.weight+500 and s1.shipmentid!=s2.shipmentid
```

The results are displayed in a table:

SHIPMENTID	WEIGHT	TYPEOFGOODS	WEIGHT
1	1000	Gadgets	1500
1	1000	Groceries	1100
2	2000	Gadgets	1500
2	2000	Kitchenware	1800
2	2000	Clothing	2200
3	1500	Electronics	1000
3	1500	Furniture	2000
3	1500	Kitchenware	1800
3	1500	Groceries	1100
4	1800	Furniture	2000
4	1800	Gadgets	1500
4	1800	Clothing	2200
5	2200	Furniture	2000
5	2200	Kitchenware	1800
6	1100	Electronics	1000
6	1100	Gadgets	1500

### Outer join:

The screenshot shows a SQL query being run in Oracle SQL Developer. The query is:

```
select s.origin,s.volume,se.name,se.businessdetails  
from shipment s, seller se  
where s.sellerid=se.sellerid(+)
```

The results are displayed in a table:

ORIGIN	VOLUME	NAME	BUSINESSDETAILS
Dhaka	500	Rahman Electronics	Electronics
Sylhet	1000	Khan Furnitures	Furniture
Khulna	800	IT Solutions BD	Gadgets
Dhaka	900	Sadia Kitchenware	Kitchenware
Chittogram	1200	Trendy Fashion	Clothing
Rajshahi	600	Deshi Foods	Groceries

6 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

### Non equi join:

Report structure OBE 2 (1).docx    SQL Commands    tables.docx

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit   Display 100  

```
select l.location,l.status,c.name
from customer c, log l
where l.logid between c.customerid and c.customer_cityid
```

**Results**   **Explain**   **Describe**   **Saved SQL**   **History**

LOCATION	STATUS	NAME
New York	Dispatched	Md. Rafiq Islam
Chicago	In Transit	Nasima Akter
Houston	Pending	Shahidul Haque
Dallas	In Transit	Farhana Rahman
San Francisco	Shipped	Kamal Uddin
Miami	Delivered	Jannatul Ferdous

6 rows returned in 0.02 seconds   [CSV Export](#)

Language: en-us   Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

## SIMPLE VIEW:

127.0.0.1:8080/apex/f?p=4500:1003:60541297797738::NO::

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit   Display 100  

```
CREATE VIEW TEST AS
SELECT CUSTOMER_CITYID, HOUSE_NUMBER, STREET_NAME,CITY
FROM CustomerAddress
WHERE city = 'Dhaka';
```

```
desc test
```

**Results**   **Explain**   **Describe**   **Saved SQL**   **History**

Object Type: **VIEW** Object: **TEST**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TEST	CUSTOMER_CITYID	Number	-	-	-	-	✓	-	-
	HOUSE_NUMBER	VARCHAR2	255	-	-	-	✓	-	-
	STREET_NAME	VARCHAR2	255	-	-	-	✓	-	-
	CITY	VARCHAR2	255	-	-	-	✓	-	-

1 - 4

Language: en-us   Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

127.0.0.1:8080/apex/f?p=4500:1003:60641297797738::NO::

**ORACLE Database Express Edition**

User SYSTEM

Home > SQL > SQL Commands

Autocommit Display 100 Use arrow keys or mouse to move toolbar

select \* from test

Save Run

Results Explain Describe Saved SQL History

CUSTOMER_CITYID	HOUSE_NUMBER	STREET_NAME	CITY
1	789C	Banani	Dhaka

1 rows returned in 0.00 seconds CSV Export

Application Express 2.1 00 39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

20°C Clear 107 2/3/2025 G

## COMPLEX VIEW

Report structure OBE 2 (1).docx SQL Commands tables.docx

127.0.0.1:8080/apex/f?p=4500:1003:60641297797738::NO::

**ORACLE Database Express Edition**

User SYSTEM

Home > SQL > SQL Commands

Autocommit Display 100

```
CREATE VIEW shipment_view(TypeOfGoods, MinWeight, MaxWeight, AvgWeight) AS
SELECT
    s.TYPEOFGOODS,
    MIN(s.WEIGHT),
    MAX(s.WEIGHT),
    AVG(s.WEIGHT)
FROM shipment s
GROUP BY s.TYPEOFGOODS;
select * from shipment view
```

Save Run

Results Explain Describe Saved SQL History

TYPEOFGOODS	MINWEIGHT	MAXWEIGHT	AVGWEIGHT
Clothing	2200	2200	2200
Electronics	1000	1000	1000
Furniture	2000	2000	2000
Kitchenware	1800	1800	1800
Groceries	1100	1100	1100
Gadgets	1500	1500	1500

6 rows returned in 0.02 seconds CSV Export

Application Express 2.1 00 39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

20°C Clear 111 2/3/2025 G

Report structure OBE 2 (1).docx SQL Commands tables.docx

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
CREATE VIEW shipment_view(TypeOfGoods, MinWeight, MaxWeight, AvgWeight) AS
SELECT
    s.TYPEOFGOODS,
    MIN(s.WEIGHT),
    MAX(s.WEIGHT),
    AVG(s.WEIGHT)
FROM shipment s
GROUP BY s.TYPEOFGOODS;
desc shipment_view
```

Results Explain Describe Saved SQL History

Object Type: VIEW Object: SHIPMENT\_VIEW

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SHIPMENT_VIEW	TYPEOFGOODS	Varchar2	255	-	-	-	✓	-	
	MINWEIGHT	Number	-	-	-	-	✓	-	
	MAXWEIGHT	Number	-	-	-	-	✓	-	
	AVGWEIGHT	Number	-	-	-	-	✓	-	

1 - 4

Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

20°C Clear 111 km/h 2/3/2025 G

## DB connection

Name: Shakila Sultana Eshita

Id:23-51719-2

oracle java - Search localhost / 127.0.0.1 / war New tab (2) Chat | Database project tables.docx Report structure OBE 2 (1)

localhost/phpmyadmin/index.php?route=sqldb=warehouse&table=warehouse&sql\_query=SELECT+\*+FROM+warehouse%60+\*%60ORDER+BY+warehouse%60+%

Showing rows 0 - 5 (6 total, Query took 0.0005 seconds) [Warehouse id: 6... - 1...]

SELECT \* FROM `warehouse` ORDER BY `warehouse`.`Warehouse\_id` DESC

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all Number of rows 25 Filter rows Search this table Sort by key None

	Warehouse Id	Name	Location	Capacity
<input type="checkbox"/>	6	Warehouse Station Road	Rangpur	10000
<input type="checkbox"/>	5	Warehouse New Market	Rajshahi	9000
<input type="checkbox"/>	4	Warehouse Khalishpur	Khulna	7000
<input type="checkbox"/>	3	Warehouse Uposhohor	Sylhet	6000
<input type="checkbox"/>	2	Warehouse Agrabad	Chittagong	8000
<input type="checkbox"/>	1	Warehouse Gulshan	Dhaka	5000

Check all With selected:  Edit  Copy  Delete  Export

Show all Number of rows 25 Filter rows Search this table Sort by key None

Query results operations Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

Console Let everyone access this bookmark

82°F Mostly cloudy 1:51 PM 2/2/2025

The screenshot shows the Eclipse IDE interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Toolbar:** Standard Eclipse toolbar icons.
- Explorer View:** Shows a project named "WAREHOUSE" with files ".vscode", "bin", "lib", "src" (containing "App.class" and "App.java"), and "README.md".
- Code Editor:** The "App.java" file is open, containing Java code to connect to a MySQL database and print warehouse information. The code uses JDBC to execute a query: "select \* from warehouse".
- Terminal View:** The terminal window shows the command-line output of running the Java application. It includes the command to set the classpath and run the application, followed by the printed results of the database query.
- Bottom Status Bar:** Shows the current weather (Mostly cloudy), Java status (Java Ready), system icons, and the date/time (2/2/2025, 1:51 PM).

```
src > J App.java > App > main(String[])
1  java.sql.*;
2
3  class App {
4
5      Run | Debug
6      lic static void main(String[] args) {
7          try {
8              Class.forName("com.mysql.cj.jdbc.Driver"); // register jdbc driver of mysql
9              Connection conn = DriverManager.getConnection(url:"jdbc:mysql://localhost:3306/warehouse", user:"root", password:"");
10             System.out.println("connected");
11             Statement st = conn.createStatement();
12             ResultSet rs = st.executeQuery(sql:"select * from warehouse");
13
14
15             while(rs.next()) {
16                 System.out.println("Warehouse id " + rs.getInt(columnIndex:1) + " Name=" + rs.getString(columnIndex:2)+ " Location " + rs.getString(columnIndex:3));
17             }
18             // Connection.close();
19         } catch (Exception s) {
20             System.out.println(s);
21         }
22     }
23 }
```

```
PS Z:\dbbb\warehouse> ^c
PS Z:\dbbb\warehouse>
PS Z:\dbbb\warehouse> zt; cd 'z:\dbbb\warehouse'; & 'C:\Program Files\Eclipse Adoptium\jdk-17.0.14.7-hotspot\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:55946' '@C:\Users\User\AppData\Local\Temp\cp_3navt732ncv5oook6qesvg.argfile' 'App'
connected
Warehouse id 1 Name=Warehouse Gulshan Location Dhaka Capacity 5000
Warehouse id 2 Name=Warehouse Agrabad Location Chattogram Capacity 8000
Warehouse id 3 Name=Warehouse Upozohor Location Sylhet Capacity 6000
Warehouse id 4 Name=Warehouse Khalishpur Location khulna Capacity 7000
Warehouse id 5 Name=Warehouse New Market Location Rajshahi Capacity 9000
Warehouse id 6 Name=Warehouse Station Road Location Rangpur Capacity 10000
PS Z:\dbbb\warehouse>
```

Name: TAWAFIQUR RAHMAN MAHIR

ID: 23-51687-2

Screenshot showing the MySQL Workbench interface and an integrated terminal window.

**MySQL Workbench:**

- Top navigation bar: New tab, Chat | Meeting in Database, tables.docx, Report structure OBE 2 (1).docx, localhost / 127.0.0.1 / customeraddress
- Left sidebar: Server 127.0.0.1 > Database: customeraddress > Table: customeraddress
- Table data view:
 

	Customer_CityID	House_Number	Street_Name	City
1	789C	Banani	Dhaka	
2	101D	GEC Circle	Chattogram	
3	202E	Amborkhana	Sylhet	
4	303F	Boyra	Khulna	
5	404G	Lakmpur	Rajshahi	
6	505H	Cantonment	Rangpur	
- Bottom toolbar: Show all, Number of rows: 25, Filter rows: Search this table, Sort by key: None

**Eclipse IDE Terminal:**

- File, Edit, Selection, View, Go, Run, Terminal, Help
- Terminal tab: warehouse
- Code editor: App.java
 

```
src > J App.java > 4s App > main(String[])
1 import java.sql.*;
2
3 public class App {
4
5     public static void main(String[] args) {
6         try {
7             Class.forName("com.mysql.cj.jdbc.Driver"); // register jdbc driver of mysql
8             Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/customeraddress", "root", "");
9             System.out.println("connected");
10            Statement st = conn.createStatement();
11            ResultSet rs = st.executeQuery("select * from customeraddress");
12
13
14            while (rs.next()) {
15                System.out.println(" Customer_CityID " + rs.getInt(columnIndex:1) + " House_Number " + rs.getString(columnIndex:2)+ " Street_Name " + rs.getString(columnIndex:3) + " City " + rs.getString(columnIndex:4));
16            }
17            // Connection.close();
18        } catch (Exception s) {
19            System.out.println(s);
20        }
21    }
22}
23}
```
- Terminal output:
 

```
PS Z:\dbbbb\warehouse> ^C
PS Z:\dbbbb\warehouse>
PS Z:\dbbbb\warehouse> z; cd 'z:\dbbbb\warehouse'; 8;c:\Program Files\Eclipse Adoptium\jdk-17.0.14.7-hotspot\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:56259' '@c:\Users\>User\AppData\Local\Temp\cp_34nav1732ncv5oxoekqesvgzg.orgfile' 'App'
Connected
Customer_CityID 1 House_Number 789C Street_Name Banani City Dhaka
Customer_CityID 2 House_Number 101D Street_Name GEC Circle City Chattogram
Customer_CityID 3 House_Number 202E Street_Name Amborkhana City Sylhet
Customer_CityID 4 House_Number 303F Street_Name Boyra City Khulna
Customer_CityID 5 House_Number 404G Street_Name Lakmpur City Rajshahi
Customer_CityID 6 House_Number 505H Street_Name Cantonment City Rangpur
```
- Bottom status bar: Ln 18, Col 34, Spaces: 4, UTF-8, CRLF, Java, 2:11 PM, 2/2/2025

Name: Mohammad Ali Masud

ID:23-51676-2

The screenshot displays two windows side-by-side. The left window is MySQL Workbench, showing a database named 'roots' with a table 'routes'. The table has columns: Routes\_ID, Origin, Destination, Distance, and EstimatedTime. The data shows six routes: Dhaka to Chatogram, Sylhet to Rajshahi, Khulna to Rangpur, Dhaka to Sylhet, Chattogram to Khulna, and Rajshahi to Dhaka. The right window is Eclipse IDE, showing a Java project named 'WAREHOUSE'. The 'src' folder contains 'App.java'. The code prints the routes from the database. The terminal shows the execution of the Java code and its output, which matches the data in the MySQL table.

Routes_ID	Origin	Destination	Distance	EstimatedTime
1	Dhaka	Chatogram	280	8 Hours
2	Sylhet	Rajshahi	350	10 Hours
3	Khulna	Rangpur	450	12 Hours
4	Dhaka	Sylhet	280	7 Hours
5	Chattogram	Khulna	400	11 Hours
6	Rajshahi	Dhaka	250	6 Hours

```

src > J App.java > ↗ App > main(String[])
1 import java.sql.*;
2
3 public class App {
4
5     Run|Debug
6     public static void main(String[] args) {
7         try {
8             Class.forName("com.mysql.cj.jdbc.Driver"); // register jdbc driver of mysql
9             Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/roots", "root", "");
10            System.out.println("connected");
11            Statement st = conn.createStatement();
12            ResultSet rs = st.executeQuery("select * from routes");
13
14
15            while (rs.next()) {
16                System.out.println(" Routes_ID " + rs.getInt(columnIndex:1) + " Origin " + rs.getString(columnIndex:2)+ " Destination " +
17            }
18            // Connection.close();
19        } catch (Exception s) {
20            System.out.println(s);
21        }
22    }
23}

```

PS Z:\dbbbb\warehouse> ^C  
PS Z:\dbbbb\warehouse>  
PS Z:\dbbbb\warehouse> z; cd 'z:\dbbbb\warehouse'; & 'c:\Program Files\Eclipse Adoptium\jdk-17.0.14.7-hotspot\bin\java.exe' '-agentlib:jdwp=transport=dt\_socket,server=n,suspend=y,address=localhost:56549' '@C:\Users\User\AppData\Local\Temp\cp\_34nav1732ncv5oxoekqesvg.orgfile' 'App'  
Connected  
Routes\_ID 1 Origin Dhaka Destination Chatogram Distance 280 EstimatedTime 8 Hours  
Routes\_ID 2 Origin Sylhet Destination Rajshahi Distance 350 EstimatedTime 10 Hours  
Routes\_ID 3 Origin Khulna Destination Rangpur Distance 450 EstimatedTime 12 Hours  
Routes\_ID 4 Origin Dhaka Destination Sylhet Distance 280 EstimatedTime 7 Hours  
Routes\_ID 5 Origin Chattogram Destination Khulna Distance 400 EstimatedTime 11 Hours  
Routes\_ID 6 Origin Rajshahi Destination Dhaka Distance 250 EstimatedTime 6 Hours

Name: MD. MAHABUBUR RAHAMAN SIAM

ID: 23-51706-2

The screenshot displays two main windows side-by-side. The top window is a web browser showing the MySQL Workbench interface. It has a header bar with tabs for 'Report structure OBE 2 (1).docx' and 'localhost / 127.0.0.1 / seller\_address'. Below the header are tabs for 'Server 127.0.0.1', 'Database: seller\_address', and 'Table: seller\_address'. The main content area shows a table with 6 rows of data:

Seller_CityID	House_Number	Street_Name	City
1	123A	Dhamondi 32	Dhaka
2	456B	Baizid Bostami	Chittogram
3	789C	Zindabazar	Sylhet
4	101D	Khulna Sadar	Khulna
5	202E	Rajshahi Court	Rajshahi
6	303F	Rangpur Station Rd	Rangpur

The bottom window is an Eclipse IDE interface. It shows a Java project named 'WAREHOUSE' with a 'src' folder containing 'App.class' and 'App.java'. The 'App.java' code is as follows:

```

src > J App.java > 4s App > mainString()
1 import java.sql.*;
2
3 public class App {
4
5     Run|Debug
6     public static void main(String[] args) {
7         try {
8             Class.forName("com.mysql.cj.jdbc.Driver"); // register jdbc driver of mysql
9             Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/seller_address", "root", "");
10            System.out.println("connected");
11            Statement st = conn.createStatement();
12            ResultSet rs = st.executeQuery("select * from seller_address");
13
14
15            while (rs.next()) {
16                System.out.println(" Seller_CityID " + rs.getInt(columnIndex:1) + " House_Number " + rs.getString(columnIndex:2)+ " Street_Name " + rs.getString(columnIndex:3));
17            }
18            // Connection.close();
19        } catch (Exception s) {
20            System.out.println(s);
21        }
22    }
23}

```

The terminal window at the bottom of the Eclipse interface shows the command-line output of the executed SQL query:

```

PS Z:\dbbbb\warehouse> ^C
PS Z:\dbbbb\warehouse>
PS Z:\dbbbb\warehouse> z; cd 'z:\dbbbb\warehouse'; & c:\Program Files\Eclipse Adoptium\jdk-17.0.14.7-hotspot\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:56768' '@C:\Users\User\AppData\Local\Temp\cp_34nav1732ncv5oxoekqesvg.orgfile' 'App'
connected
Seller_CityID 1 House_Number 123A Street_Name Dhamondi 32 City Dhaka
Seller_CityID 2 House_Number 456B Street_Name Baizid Bostami City Chittogram
Seller_CityID 3 House_Number 789C Street_Name Zindabazar City Sylhet
Seller_CityID 4 House_Number 101D Street_Name Khulna Sadar City Khulna
Seller_CityID 5 House_Number 202E Street_Name Rajshahi Court City Rajshahi
Seller_CityID 6 House_Number 303F Street_Name Rangpur Station Rd City Rangpur
PS Z:\dbbbb\warehouse>

```

## Conclusion

The Freight Forwarder Management System is a game-changer for the logistics industry, leveraging modern technologies to address pain points in freight forwarding operations. By automating processes, providing real-time visibility, and ensuring secure transactions, the system will benefit freight forwarders,

customers, sellers, carriers, and warehouse managers alike. This project not only contributes to the tech sector but also paves the way for a more efficient and sustainable logistics ecosystem.