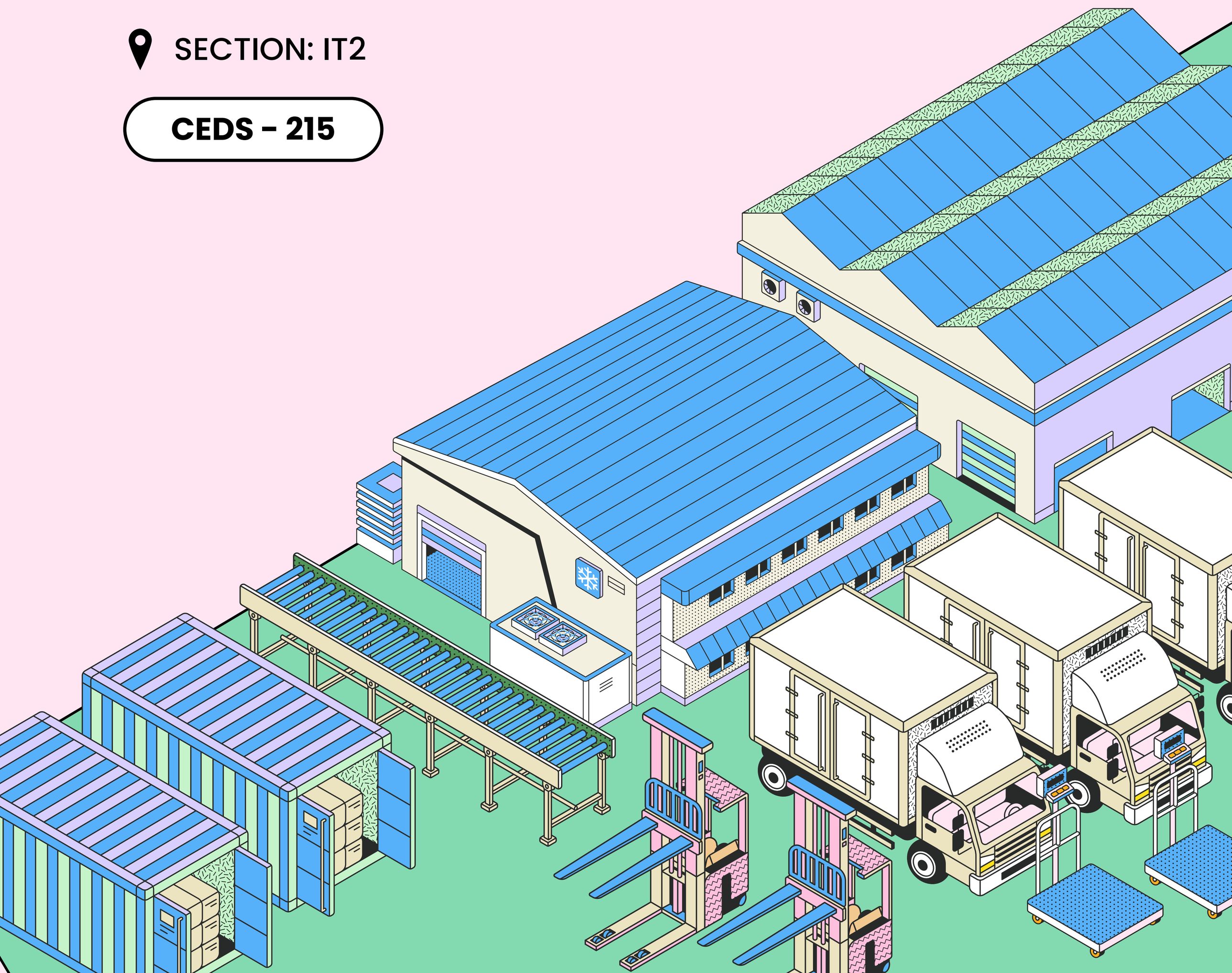
جامعة جدة
University of Jeddah

TRANSPORTATION COMPANY

PROJECT OF DATABASE MANEGMENT SYSTEM

SECTION: IT2

CEDS - 215



OUR TEAM MEMBERS

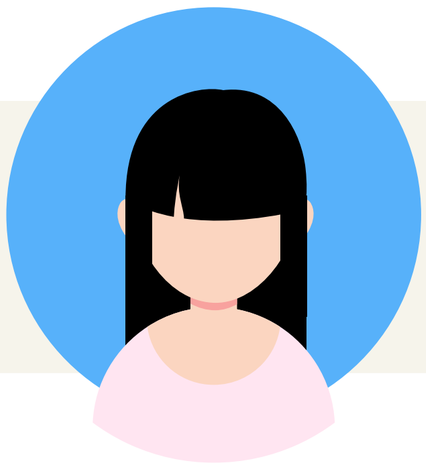
LAMA ALSHEHRI

2410836

NUHA ALBERAIKI

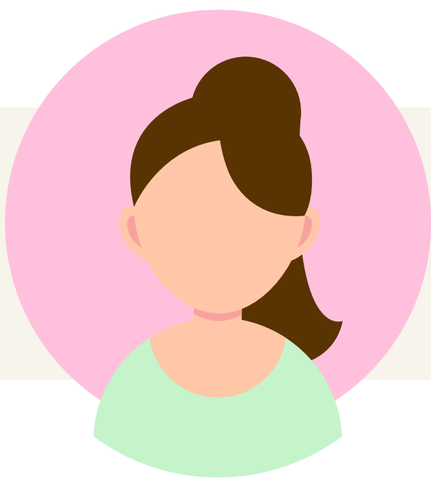
2411956

RAMA IBRAHIM



2411245

RUAA ALBERAIKI



2410862

INSTRUCTOR: MASHAEL KHAYYAT

Table of CONTENTS

Transportation Company Database

System analysis

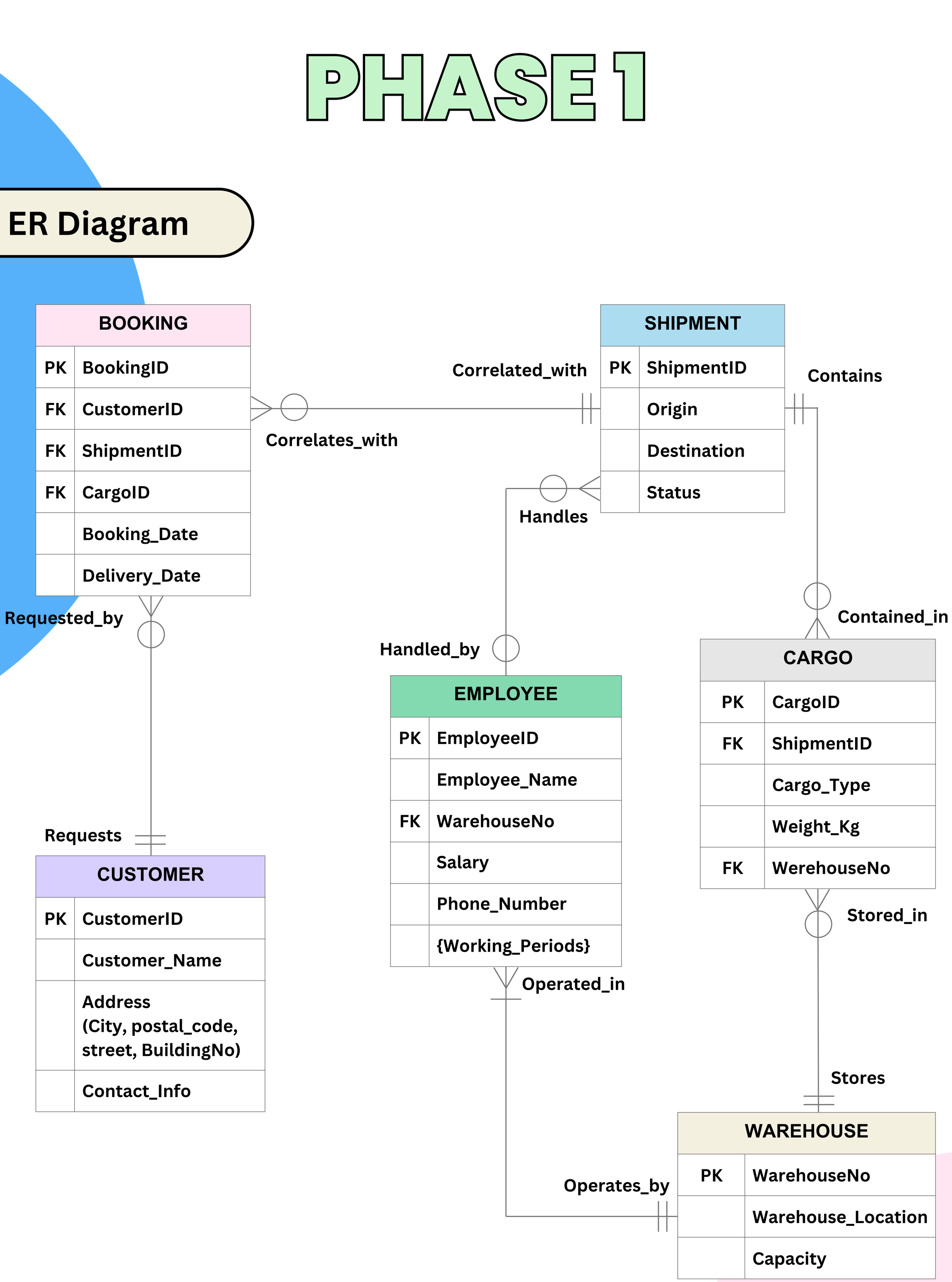
Our transportation Company is a company specialized in transporting products through water crossings and is distinguished by being a reliable and punctual company, but the marine logistics company encountered challenges resulting from outdated systems and disjointed processes, leading to inefficiencies and customer dissatisfaction. However, following the implementation of a new system, the company has undergone a remarkable transformation. The relationships between various aspects of the company's operations, such as bookings, tracking, and warehouse management have been improved to ensure an ideal booking experience and real-time tracking. Moreover, the company's integrated warehouse inventory management ensures efficient order fulfillment, leading to smoother operations and fewer errors. Consequently, the company's reputation as a reliable and efficient marine logistics provider has been strengthened. Whether the client requires dependable shipping or customs clearance, the Marine Logistics Company is well-positioned to streamline the supply chain and ensure the excellent movement of your cargo!

General description

The company operates by receiving one or many bookings from each customer, organizing each shipment that may contain many cargo items, utilizing warehouses for storage and distribution, and employing personnel to manage various logistics operations. The company's objective is to provide efficient, reliable, and secure transportation services to meet the needs of its customers and ensure the smooth flow of goods across the town.

PHASE 1

ER Diagram



PHASE 2

Relational Schema

CUSTOMER

CustomerID	Customer_Name	City	Postal_code	Streeet	BuildingNo	Contact_Info

BOOKING

Booking ID	CustomerID	Shipment ID	CargoID	Booking_Date	Delivery_Date

SHIPMENT

Shipment ID	Origin	Destination	Status

CARGO

Cargo ID	ShipmentID	Cargo_Type	Weight_Kg	WerehouseNo

WAREHOUSE

WarehouseNo	Warehouse_Location	Capacity

EMPLOYEE

Employee ID	Employee _Name	Working_Periods	WarehouseNo	Salary	Phone_Number

Functional Dependencies(FD)

- **FD <u>CustomerID</u>** → Customer_Name, Address, Contact_Info.
- **FD <u>BookingID</u>** → Booking_Date, Delivery_Date.
- **FD <u>ShipmentID</u>** → Origin, Destination, Status.
- **FD <u>CargoID</u>** → Cargo_Type, Weight_Kg.
- **FD <u>WarehouseNo</u>** → Warehouse_Location, Capacity.
- **FD <u>EmployeeID</u>** → Employee_Name, Working_Periods, Salary, Phone_No.

Normalization

- **1NF**

  A table is in the first normal form:
  - Does not contain any composite or multi-valued attributes.
  - Each relation has a unique key.
  - No repeating groups.

  Multi-valued attribute (Working_Periods) in the EMPLOYEE table.
  All other tables are already in 1NF.

- **2NF**

  A table is in the second normal form:
  - Does not have a partial functional dependency.

  All tables are already in 2NF.

- **3NF**

  A table is in the third normal form:
  - Does not have a transitive dependency.

  All tables are already in 3NF.

Normalization

CUSTOMER

CustomerID	Customer_Name	City	Postal_code	Streeet	BuildingNo	Contact_Info

BOOKING

Booking ID	CustomerID	Shipment ID	CargoID	Booking_Date	Delivery_Date

SHIPMENT

Shipment ID	Origin	Destination	Status

CARGO

Cargo ID	ShipmentID	Cargo_Type	Weight_Kg	WerehouseNo

WAREHOUSE

WarehouseNo	Warehouse_Location	Capacity

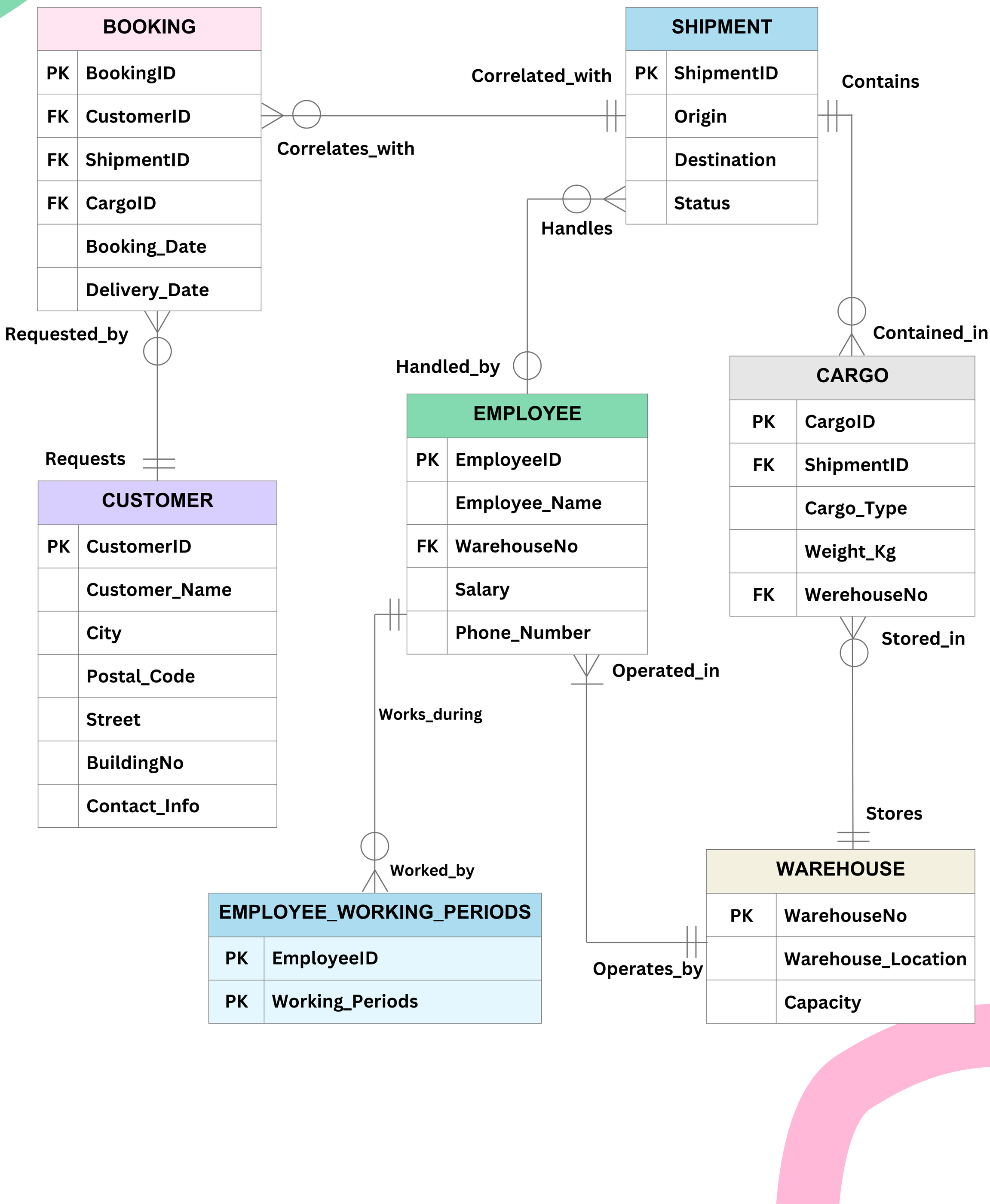
EMPLOYEE

Employee ID	Employee _Name	WarehouseNo	Salary	Phone_Number

EMPLOYEE_WORKING_PERIODS

Employee ID	Working_periods

Logical modeling



PHASE 3

Table name	Attributes	Data type	Cardinalities
CUSTOMER	CustomerID Customer_Name City Postal_Code Street Building_No Contact_Info	NUMBER(6) VARCHAR2(20) VARCHAR2(15) NUMBER(5) VARCHAR2(15) NUMBER(5) VARCHAR2(30)	Primary key
BOOKING	BookingID CustomerID ShipmentID CargoID Booking_Date Delivery_Date	NUMBER(6) NUMBER(6) NUMBER(6) NUMBER(6) DATE DATE	Primary key Foreign key Foreign key Foreign key
SHIPMENT	ShipmentID Origin Destination Status	NUMBER(6) VARCHAR2(30) VARCHAR2(30) VARCHAR2(25)	Primary key
CARGO	CargoID ShipmentID Cargo_Type Weight WerehouseNo	NUMBER(6) NUMBER(6) VARCHAR2(20) NUMBER(5.2) NUMBER(3)	Primary key Foreign key
WAREHOUSE	WarehouseNo Warehouse_Location Capacity	NUMBER(3) VARCHAR2(25) NUMBER(4)	Primary key
EMPLOYEE	EmployeeID WarehouseNo Employee_Name Salary Phone_Number	NUMBER(6) NUMBER(3) VARCHAR2(30) NUMBER(5) NUMBER(15)	Primary key Foreign key
EMPLOYEE_WORKING_PERIODS	EmployeeID Working_Periods	NUMBER(6) VARCHAR2(45)	Primary key

CODING

CUSTOMER Table

- Table Creation

SQL Worksheet Clear Find Actions Save Run

```
1  CREATE TABLE CUSTOMER (
2      CustomerID NUMBER(6),
3      Customer_Name VARCHAR2(20),
4      City VARCHAR2(15),
5      Postal_Code NUMBER(5),
6      Street VARCHAR2(15),
7      Building_No NUMBER(5),
8      Contact_Info VARCHAR2(30),
9      CONSTRAINT CustomerID_pk PRIMARY KEY(CustomerID)
10 );
11 DESC CUSTOMER;
```

```
Table created.
```

TABLE CUSTOMER

Column	Null?	Type
CUSTOMERID	NOT NULL	NUMBER(6,0)
CUSTOMER_NAME	-	VARCHAR2(20)
CITY	-	VARCHAR2(15)
POSTAL_CODE	-	NUMBER(5,0)
STREET	-	VARCHAR2(15)
BUILDING_NO	-	NUMBER(5,0)
CONTACT_INFO	-	VARCHAR2(30)

Download CSV

```
7 rows selected.
```

CODING

CUSTOMER Table

- Insert Valus

SQL Worksheet

Clear Find Actions

```sql
-- Inserting values into CUSTOMER table
INSERT INTO CUSTOMER (CustomerID, Customer_Name, City, Postal_Code, Street, Building_No, Contact_Info)
VALUES
    (1, 'John Doe', 'New York', '10001', '123 Main St','5678', '123-456-7890');
INSERT INTO CUSTOMER (CustomerID, Customer_Name, City, Postal_Code, Street, Building_No, Contact_Info)
VALUES (2, 'Jane Smith', 'Los Angeles', '90001', '456 Elm St', '6377','987-654-3210');
INSERT INTO CUSTOMER (CustomerID, Customer_Name, City, Postal_Code, Street, Building_No, Contact_Info)
VALUES (3, 'Alice Johnson', 'Chicago', '60601', '789 Oak St', '4435','555-123-4567');
INSERT INTO CUSTOMER (CustomerID, Customer_Name, City, Postal_Code, Street, Building_No, Contact_Info)
VALUES (4, 'Bob Brown', 'Houston', '77001', '321 Pine St', '9446','456-789-0123');
INSERT INTO CUSTOMER (CustomerID, Customer_Name, City, Postal_Code, Street, Building_No, Contact_Info)
VALUES(5, 'Emily Davis', 'Miami', '33101', '654 Maple St', '6556','321-654-0987');
INSERT INTO CUSTOMER (CustomerID, Customer_Name, City, Postal_Code, Street, Building_No, Contact_Info)
VALUES (6, 'Michael Wilson', 'San Francisco', '94101', '987 Cedar St','3867', '789-012-3456');
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

SQL Worksheet

Clear Find Actions

```sql
SELECT * FROM CUSTOMER;
```

CUSTOMERID	CUSTOMER_NAME	CITY	POSTAL_CODE	STREET	BUILDING_NO	CONTACT_INFO
1	John Doe	New York	10001	123 Main St	5678	123-456-7890
2	Jane Smith	Los Angeles	90001	456 Elm St	6377	987-654-3210
3	Alice Johnson	Chicago	60601	789 Oak St	4435	555-123-4567
4	Bob Brown	Houston	77001	321 Pine St	9446	456-789-0123
5	Emily Davis	Miami	33101	654 Maple St	6556	321-654-0987
6	Michael Wilson	San Francisco	94101	987 Cedar St	3867	789-012-3456

Download CSV

6 rows selected.

CODING

WAREHOUSE Table

- Table Creation

SQL Worksheet

Clear | Find | Actions | Save | Run

```
1  CREATE TABLE WAREHOUSE (
2      WarehouseNo NUMBER(3),
3      Warehouse_Location VARCHAR2(25),
4      Capacity NUMBER(4),
5      CONSTRAINT WarehouseNo_pk PRIMARY KEY(WarehouseNo)
6  );
7  DESC WAREHOUSE;
```

Table created.

TABLE WAREHOUSE

Column	Null?	Type
WAREHOUSENO	NOT NULL	NUMBER(3,0)
WAREHOUSE_LOCATION	-	VARCHAR2(25)
CAPACITY	-	NUMBER(4,0)

Download CSV

3 rows selected.

CODING

WAREHOUSE Table

- Table Creation

SQL Worksheet

Clear | Find | Actions | Save | Run

```
1  CREATE TABLE WAREHOUSE (
2      WarehouseNo NUMBER(3),
3      Warehouse_Location VARCHAR2(25),
4      Capacity NUMBER(4),
5      CONSTRAINT WarehouseNo_pk PRIMARY KEY(WarehouseNo)
6  );
7  DESC WAREHOUSE;
```

Table created.

TABLE WAREHOUSE

Column	Null?	Type
WAREHOUSENO	NOT NULL	NUMBER(3,0)
WAREHOUSE_LOCATION	-	VARCHAR2(25)
CAPACITY	-	NUMBER(4,0)

Download CSV

3 rows selected.

CODING

WAREHOUSE Table

- Insert Valus

SQL Worksheet Clear Find Actions Save Run

```
1  INSERT INTO WAREHOUSE ( WarehouseNo, Warehouse_Location, Capacity)
2  VALUES
3       (1, 'New York Warehouse', 1000);
4  INSERT INTO WAREHOUSE ( WarehouseNo, Warehouse_Location, Capacity)
5  VALUES (2, 'Los Angeles Warehouse', 800);
6  INSERT INTO WAREHOUSE ( WarehouseNo, Warehouse_Location, Capacity)
7  VALUES (3, 'Chicago Warehouse', 1200);
8  INSERT INTO WAREHOUSE ( WarehouseNo, Warehouse_Location, Capacity)
9  VALUES (4, 'Houston Warehouse', 900);
10 INSERT INTO WAREHOUSE ( WarehouseNo, Warehouse_Location, Capacity)
11 VALUES (5, 'Miami Warehouse', 700);
12 INSERT INTO WAREHOUSE ( WarehouseNo, Warehouse_Location, Capacity)
13 VALUES (6, 'San Francisco Warehouse', 1100);
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

SQL Worksheet Clear Find Actions Save Run

```
1  SELECT * FROM WAREHOUSE;
```

WAREHOUSENO	WAREHOUSE_LOCATION	CAPACITY
1	New York Warehouse	1000
2	Los Angeles Warehouse	800
3	Chicago Warehouse	1200
4	Houston Warehouse	900
5	Miami Warehouse	700
6	San Francisco Warehouse	1100

Download CSV

6 rows selected.

CODING

SHIPMENT Table

- Table Creation

SQL Worksheet

Clear | Find | Actions | Save | Run

```
1  CREATE TABLE SHIPMENT (
2      ShipmentID NUMBER(6),
3      Origin VARCHAR2(30),
4      Destination VARCHAR2(30),
5      Status VARCHAR2(25),
6      CONSTRAINT ShipmentID_pk PRIMARY KEY(ShipmentID)
7  );
8  DESC SHIPMENT;
```

Table created.

TABLE SHIPMENT

Column	Null?	Type
SHIPMENTID	NOT NULL	NUMBER(6,0)
ORIGIN	-	VARCHAR2(30)
DESTINATION	-	VARCHAR2(30)
STATUS	-	VARCHAR2(25)

Download CSV

4 rows selected.

CODING

SHIPMENT Table

- Insert Valus

SQL Worksheet Clear Find Actions Save Run

```
INSERT INTO SHIPMENT (ShipmentID, Origin, Destination, Status)
VALUES
    (101, 'New York', 'Los Angeles', 'In Transit');
INSERT INTO SHIPMENT (ShipmentID, Origin, Destination, Status)
VALUES (102, 'Chicago', 'Houston', 'Delayed');
INSERT INTO SHIPMENT (ShipmentID, Origin, Destination, Status)
VALUES (103, 'Miami', 'San Francisco', 'Scheduled');
INSERT INTO SHIPMENT  (ShipmentID, Origin, Destination, Status)
VALUES (104, 'Houston', 'New York', 'In Transit');
INSERT INTO SHIPMENT (ShipmentID, Origin, Destination, Status)
VALUES(105, 'Los Angeles', 'Miami', 'Scheduled');
INSERT INTO SHIPMENT(ShipmentID, Origin, Destination, Status)
VALUES (106, 'San Francisco', 'Chicago', 'In Transit');
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

SQL Worksheet Clear Find Actions Save Run

```
SELECT *FROM SHIPMENT;
```

SHIPMENTID	ORIGIN	DESTINATION	STATUS
101	New York	Los Angeles	In Transit
102	Chicago	Houston	Delayed
103	Miami	San Francisco	Scheduled
104	Houston	New York	In Transit
105	Los Angeles	Miami	Scheduled
106	San Francisco	Chicago	In Transit

Download CSV

6 rows selected.

CODING

CARGO Table

- Table Creation

SQL Worksheet Clear Find Actions Save Run

```
1  CREATE TABLE CARGO (
2      CargoID NUMBER(6),
3      ShipmentID NUMBER(6),
4      Cargo_Type VARCHAR2(20),
5      Weight_Kg NUMBER(5, 2),
6      WarehouseNo NUMBER(3),
7      CONSTRAINT CargoID_pk PRIMARY KEY(CargoID),
8      CONSTRAINT ShipmentID_fk FOREIGN KEY (ShipmentID) REFERENCES SHIPMENT(ShipmentID),
9      CONSTRAINT WarehouseNo_Cargo_fk FOREIGN KEY (WarehouseNo) REFERENCES WAREHOUSE(WarehouseNo)
10 );
11 DESC CARGO;
```

Table created.

TABLE CARGO

Column	Null?	Type
CARGOID	NOT NULL	NUMBER(6,0)
SHIPMENTID	-	NUMBER(6,0)
CARGO_TYPE	-	VARCHAR2(20)
WEIGHT_KG	-	NUMBER(5,2)
WAREHOUSENO	-	NUMBER(3,0)

CODING

CARGO Table

- Insert Valus

SQL Worksheet Clear Find Actions Save Run

```
1  INSERT INTO CARGO (CargoID, ShipmentID,  Cargo_Type,  Weight_Kg,  WarehouseNo)
2  VALUES(201, 101, 'Electronics', 500, 1);
3  INSERT INTO CARGO (CargoID, ShipmentID,  Cargo_Type,  Weight_Kg,  WarehouseNo)
4  VALUES (202, 102, 'Furniture', 800, 3);
5  INSERT INTO CARGO (CargoID, ShipmentID,  Cargo_Type,  Weight_Kg,  WarehouseNo)
6  VALUES(203, 103, 'Clothing', 300, 5);
7  INSERT INTO CARGO (CargoID, ShipmentID,  Cargo_Type,  Weight_Kg,  WarehouseNo)
8  VALUES (204, 104, 'Books', 400, 4);
9  INSERT INTO CARGO (CargoID, ShipmentID,  Cargo_Type,  Weight_Kg,  WarehouseNo)
10 VALUES(205, 105, 'Food', 600, 2);
11 INSERT INTO CARGO (CargoID, ShipmentID,  Cargo_Type,  Weight_Kg,  WarehouseNo)
12 VALUES (206, 106, 'Toys', 200, 6);
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

SQL Worksheet Clear Find Actions Save Run

```
1  SELECT *FROM CARGO;
```

CARGOID	SHIPMENTID	CARGO_TYPE	WEIGHT_KG	WAREHOUSENO
201	101	Electronics	500	1
202	102	Furniture	800	3
203	103	Clothing	300	5
204	104	Books	400	4
205	105	Food	600	2
206	106	Toys	200	6

Download CSV

6 rows selected.

CODING

BOOKING Table

- Table Creation

SQL Worksheet Clear Find Actions Save Run

```
1  CREATE TABLE BOOKING (
2      BookingID NUMBER(6),
3      CustomerID NUMBER(6),
4      ShipmentID NUMBER(6),
5      CargoID NUMBER(6),
6      Booking_Date DATE DEFAULT SYSDATE,
7      Delivery_Date DATE,
8      CONSTRAINT BookingID_pk PRIMARY KEY(BookingID),
9      CONSTRAINT CustomerID_fk FOREIGN KEY (CustomerID) REFERENCES CUSTOMER(CustomerID),
10     CONSTRAINT ShipmentID_Booking_fk FOREIGN KEY (ShipmentID) REFERENCES SHIPMENT(ShipmentID),
11     CONSTRAINT CargoID_fk FOREIGN KEY (CargoID) REFERENCES CARGO(CargoID)
12 );
13 DESC BOOKING;
```

```
Table created.

TABLE BOOKING
```

Column	Null?	Type
BOOKINGID	NOT NULL	NUMBER(6,0)
CUSTOMERID	-	NUMBER(6,0)
SHIPMENTID	-	NUMBER(6,0)
CARGOID	-	NUMBER(6,0)
BOOKING_DATE	-	DATE
DELIVERY_DATE	-	DATE

Download CSV

6 rows selected.

CODING

BOOKING Table

- Insert Valus

SQL Worksheet Clear Find Actions Save Run

```
1  INSERT INTO BOOKING (BookingID, CustomerID, ShipmentID, CargoID,  Delivery_Date)
2  VALUES
3       (301, 1, 101, 201, TO_DATE('2024-05-05', 'YYYY-MM-DD'));
4  INSERT INTO BOOKING (BookingID, CustomerID, ShipmentID, CargoID,  Delivery_Date)
5  VALUES (302, 2, 102, 202, TO_DATE('2024-05-06', 'YYYY-MM-DD'));
6  INSERT INTO BOOKING (BookingID, CustomerID, ShipmentID, CargoID,  Delivery_Date)
7  VALUES (303, 3, 103, 203, TO_DATE('2024-05-07', 'YYYY-MM-DD'));
8  INSERT INTO BOOKING (BookingID, CustomerID, ShipmentID, CargoID,  Delivery_Date)
9  VALUES (304, 4, 104, 204, TO_DATE('2024-05-08', 'YYYY-MM-DD'));
10 INSERT INTO BOOKING (BookingID, CustomerID, ShipmentID, CargoID,  Delivery_Date)
11 VALUES(305, 5, 105, 205, TO_DATE('2024-05-09', 'YYYY-MM-DD'));
12 INSERT INTO BOOKING (BookingID, CustomerID, ShipmentID, CargoID,  Delivery_Date)
13 VALUES (306, 6, 106, 206, TO_DATE('2024-05-10', 'YYYY-MM-DD'));
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

SQL Worksheet Clear Find Actions Save Run

```
1  SELECT *FROM BOOKING;
```

BOOKINGID	CUSTOMERID	SHIPMENTID	CARGOID	BOOKING_DATE	DELIVERY_DATE
301	1	101	201	08-MAY-24	05-MAY-24
302	2	102	202	08-MAY-24	06-MAY-24
303	3	103	203	08-MAY-24	07-MAY-24
304	4	104	204	08-MAY-24	08-MAY-24
305	5	105	205	08-MAY-24	09-MAY-24
306	6	106	206	08-MAY-24	10-MAY-24

Download CSV

6 rows selected.

CODING

EMPLOYEE Table

- Table Creation

SQL Worksheet Clear Find Actions Save Run

```
1  CREATE TABLE EMPLOYEE (
2      EmployeeID NUMBER(6),
3      WarehouseNo NUMBER(3),
4      Employee_Name VARCHAR2(30),
5      Salary NUMBER(5),
6      Phone_Number VARCHAR2(15),
7      CONSTRAINT EmployeeID_pk PRIMARY KEY(EmployeeID),
8      CONSTRAINT WarehouseNo_Employee_fk FOREIGN KEY (WarehouseNo) REFERENCES WAREHOUSE(WarehouseNo)
9  );
10 DESC EMPLOYEE;
```

Table created.

TABLE EMPLOYEE

Column	Null?	Type
EMPLOYEEID	NOT NULL	NUMBER(6,0)
WAREHOUSENO	-	NUMBER(3,0)
EMPLOYEE_NAME	-	VARCHAR2(30)
SALARY	-	NUMBER(5,0)
PHONE_NUMBER	-	VARCHAR2(15)

CODING

EMPLOYEE Table

- Insert Valus

SQL Worksheet Clear Find Actions Save Run

```
1  INSERT INTO EMPLOYEE (EmployeeID,  WarehouseNo, Employee_Name, Salary, Phone_Number)
2  VALUES
3      (401, 1, 'John Smith', 50000, '1234567890');
4  INSERT INTO EMPLOYEE (EmployeeID,  WarehouseNo, Employee_Name, Salary, Phone_Number)
5  VALUES (402, 2, 'Jane Doe', 55000, '9876543210');
6  INSERT INTO EMPLOYEE (EmployeeID,  WarehouseNo, Employee_Name, Salary, Phone_Number)
7  VALUES (403, 3, 'Alice Johnson', 48000, '5551234567');
8  INSERT INTO EMPLOYEE (EmployeeID,  WarehouseNo, Employee_Name, Salary, Phone_Number)
9  VALUES (404, 4, 'Bob Brown', 52000, '4567890123');
10 INSERT INTO EMPLOYEE (EmployeeID,  WarehouseNo, Employee_Name, Salary, Phone_Number)
11 VALUES (405, 5, 'Emily Davis', 49000, '3216540987');
12 INSERT INTO EMPLOYEE (EmployeeID,  WarehouseNo, Employee_Name, Salary, Phone_Number)
13 VALUES (406, 6, 'Michael Wilson', 53000, '7890123456');
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

SQL Worksheet Clear Find Actions Save Run

```
1  SELECT *FROM EMPLOYEE;
```

EMPLOYEEID	WAREHOUSENO	EMPLOYEE_NAME	SALARY	PHONE_NUMBER
401	1	John Smith	50000	1234567890
402	2	Jane Doe	55000	9876543210
403	3	Alice Johnson	48000	5551234567
404	4	Bob Brown	52000	4567890123
405	5	Emily Davis	49000	3216540987
406	6	Michael Wilson	53000	7890123456

Download CSV

6 rows selected.

CODING

EMPLOYEE_WORKING_PERIODS Table

- Table Creation

SQL Worksheet

Clear | Find | Actions | Save | Run

```
1  CREATE TABLE EMPLOYEE_WORKING_PERIODS (
2      EmployeeID NUMBER(6),
3      Working_Period VARCHAR2(45),
4      CONSTRAINT Employee_WorkingPeriod_pk PRIMARY KEY(EmployeeID, Working_Period),
5      CONSTRAINT EmployeeID_fk FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE(EmployeeID)
6  );
7  DESC EMPLOYEE_WORKING_PERIODS;
```

```
Table created.
```

TABLE EMPLOYEE_WORKING_PERIODS

Column	Null?	Type
EMPLOYEEID	NOT NULL	NUMBER(6,0)
WORKING_PERIOD	NOT NULL	VARCHAR2(45)

Download CSV

2 rows selected.

CODING

EMPLOYEE_WORKING_PERIODS Table

- Insert Valus

SQL Worksheet Clear Find Actions Save Run

```
INSERT INTO EMPLOYEE_WORKING_PERIODS (EmployeeID, Working_Period)
VALUES
    (401, 'Monday-Morning Shift');
INSERT INTO EMPLOYEE_WORKING_PERIODS (EmployeeID, Working_Period)
VALUES (402, 'Wednesday-Evening Shift');
INSERT INTO EMPLOYEE_WORKING_PERIODS (EmployeeID, Working_Period)
VALUES(406, 'Thursday-Night Shift');
INSERT INTO EMPLOYEE_WORKING_PERIODS (EmployeeID, Working_Period)
VALUES (405, 'Sunday-Evening Shift');
INSERT INTO EMPLOYEE_WORKING_PERIODS (EmployeeID, Working_Period)
VALUES(406, 'Tuseday-Night Shift');
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

SQL Worksheet Clear Find Actions Save Run

```
SELECT *FROM EMPLOYEE_WORKING_PERIODS;
```

EMPLOYEEID	WORKING_PERIOD
401	Monday-Morning Shift
402	Wednesday-Evening Shift
405	Sunday-Evening Shift
406	Thursday-Night Shift
406	Tuseday-Night Shift

Download CSV

5 rows selected.

CODING

The Queries

1

SQL Worksheet

Clear | Find | Actions | Save | Run

```
1 SELECT ShipmentID, Origin, Destination
2 FROM SHIPMENT
3 WHERE Status = 'In Transit'
4 ORDER BY shipmentID ASC;
```

SHIPMENTID	ORIGIN	DESTINATION
101	New York	Los Angeles
104	Houston	New York
106	San Francisco	Chicago

Download CSV

3 rows selected.

2

SQL Worksheet

Clear | Find | Actions | Save | Run

```
1 SELECT City, COUNT(CustomerID) AS Customer
2 FROM CUSTOMER
3 GROUP BY City
4 ORDER BY Customer ASC;
```

CITY	CUSTOMER
Miami	1
Los Angeles	1
Houston	1
New York	1
San Francisco	1
Chicago	1

Download CSV

6 rows selected.

CODING

The Queries

3

Live SQL — Feedback — Help — 2410862@uj.edu.sa

SQL Worksheet — Clear — Find — Actions — Save — Run

```
1  SELECT E.Employee_Name, W.Warehouse_Location
2  FROM EMPLOYEE E
3  JOIN WAREHOUSE W ON E.WarehouseNo = W.WarehouseNo
4  WHERE E.Salary > 50000;
```

EMPLOYEE_NAME	WAREHOUSE_LOCATION
Jane Doe	Los Angeles Warehouse
Bob Brown	Houston Warehouse
Michael Wilson	San Francisco Warehouse

Download CSV

3 rows selected.

4

SQL Worksheet — Clear — Find — Actions — Save — Run

```
1  SELECT ShipmentID, Destination
2  FROM SHIPMENT
3  WHERE ShipmentID IN (
4      SELECT ShipmentID
5      FROM CARGO
6      WHERE Weight_Kg > (
7      SELECT AVG (Weight_Kg)
8      FROM CARGO
9      )
10 );
```

SHIPMENTID	DESTINATION
105	Miami
101	Los Angeles
102	Houston

Download CSV

3 rows selected.

CODING

Stored Procedures

- **Parameter**

SQL Worksheet — Clear | Find | Actions | Save | Run

```
1  CREATE OR REPLACE PROCEDURE EMPLOYEES_BY_WAREHOUSE(
2  p_WarehouseNo IN VARCHAR2
3
4  )AS
5     V_EmployeeID EMPLOYEE.EmployeeID%TYPE;
6     V_Employee_Name EMPLOYEE.Employee_Name%TYPE;
7     V_Phone_Number EMPLOYEE.Phone_Number%TYPE;
8
9  BEGIN
10   SELECT EmployeeID, Employee_Name, Phone_Number
11   INTO V_EmployeeID, V_Employee_Name, V_Phone_Number
12   FROM EMPLOYEE
13   WHERE WarehouseNo = p_WarehouseNo;
14
15    DBMS_OUTPUT.PUT_LINE('Employee ID: ' || V_EmployeeID);
16    DBMS_OUTPUT.PUT_LINE('Employee Name: ' || V_Employee_Name);
17    DBMS_OUTPUT.PUT_LINE('Phone Number: ' || V_Phone_Number);
18
19 END EMPLOYEES_BY_WAREHOUSE;
```

```
Procedure created.
```

- **Procedure Call**

SQL Worksheet — Clear | Find | Actions | Save | Run

```
1  EXECUTE EMPLOYEES_BY_WAREHOUSE(3);
```

```
Statement processed.
Employee ID: 403
Employee Name: Alice Johnson
Phone Number: 5551234567
```

CODING

Stored Procedures

- **Update Procedure**

SQL Worksheet — Clear | Find | Actions | Save | Run

```
1  CREATE OR REPLACE PROCEDURE UPDATE_STATUS (
2      p_ShipmentID IN NUMBER,
3      p_Status IN VARCHAR2
4  )
5  AS
6  BEGIN
7      UPDATE SHIPMENT
8      SET Status = p_Status
9      WHERE ShipmentID = p_ShipmentID;
10
11     COMMIT;
12
13     DBMS_OUTPUT.PUT_LINE('Shipment status updated successfully.');
14 END UPDATE_STATUS;
```

```
Procedure created.
```

- **Procedure Call**

SQL Worksheet — Clear | Find | Actions | Save | Run

```
1  SELECT* FROM SHIPMENT;
2  EXECUTE UPDATE_STATUS (105,'In Transit')
3  SELECT* FROM SHIPMENT;
```

SHIPMENTID	ORIGIN	DESTINATION	STATUS
101	New York	Los Angeles	In Transit
102	Chicago	Houston	Delayed
103	Miami	San Francisco	Scheduled
104	Houston	New York	In Transit
105	Los Angeles	Miami	Scheduled
106	San Francisco	Chicago	In Transit

Download CSV

6 rows selected.

Statement processed.
Shipment status updated successfully.

SHIPMENTID	ORIGIN	DESTINATION	STATUS
101	New York	Los Angeles	In Transit
102	Chicago	Houston	Delayed
103	Miami	San Francisco	Scheduled
104	Houston	New York	In Transit
105	Los Angeles	Miami	In Transit
106	San Francisco	Chicago	In Transit

Project Team
TASKS

Name	Tasks
Lama Alshehri	• Relational schema • Functional Dependencies • Normalization • Table
Nuha Alberaiki	• Normalization • The qerise • Stored procedures
Rama Ibrahim	• Logical modeling • Create tables • Insert values
Ruaa Alberaiki	• Relational schema • Insert values • Final design
All of Us	• ER_Digram