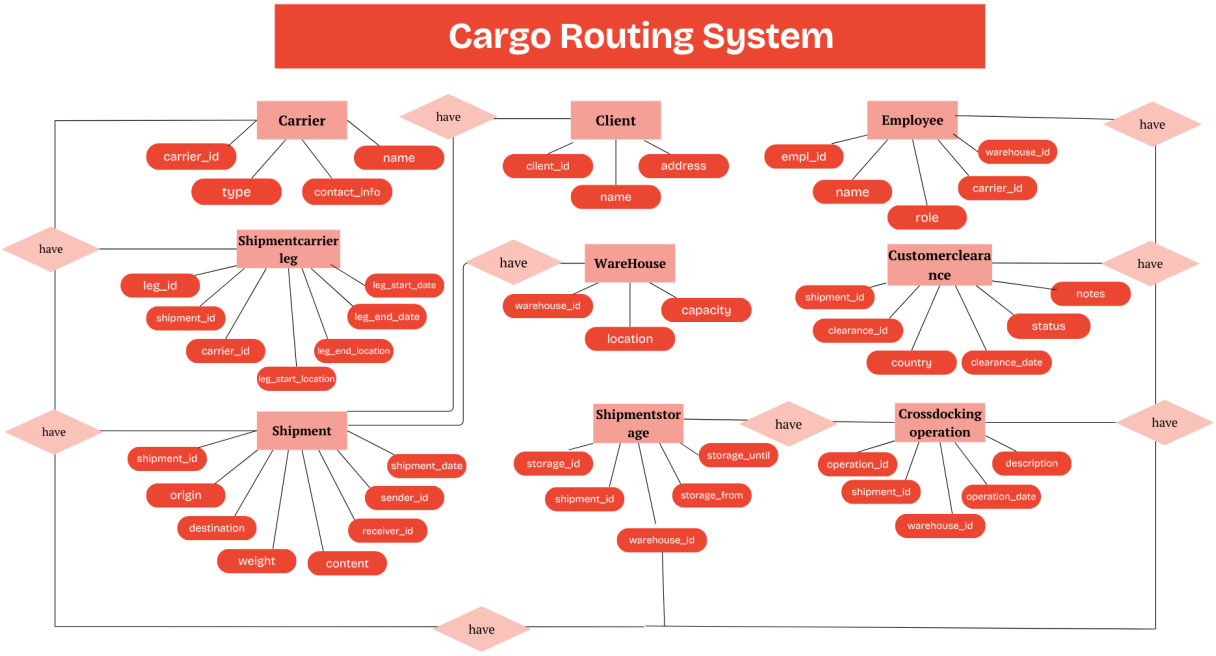
Project 5:Cargo Routing System

Design an Entity-Relationship schema for a global logistics and supply chain management system. The database should store information about shipments with shipment ID, origin, destination, weight, content description, shipment date, and carrier. Warehouses have warehouse ID, location, capacity, and the shipments they store. Carriers have carrier ID, name, type (air, land, sea), and contact information. Clients have client ID, name, address, shipments they send, and shipments they receive. Employees have employee ID, name, assigned warehouse or carrier, and role. Shipments can pass through multiple warehouses before reaching the destination, but can be stored in only one warehouse at any given time. Each shipment is handled by a carrier, but carriers can change across different legs of the route.
Clients can be both sender and receiver of shipments. Employees can either be warehouse staff or carrier crew but cannot be both. Warehouses can store multiple shipments and shipments may change storage warehouses multiple times before delivery. The system should handle complex scenarios like international shipments, customs clearance, and cross-docking operations.



🗒️ SQL Table Creation Statements:

```
CREATE TABLE Client (
    client_id INT PRIMARY KEY,
```

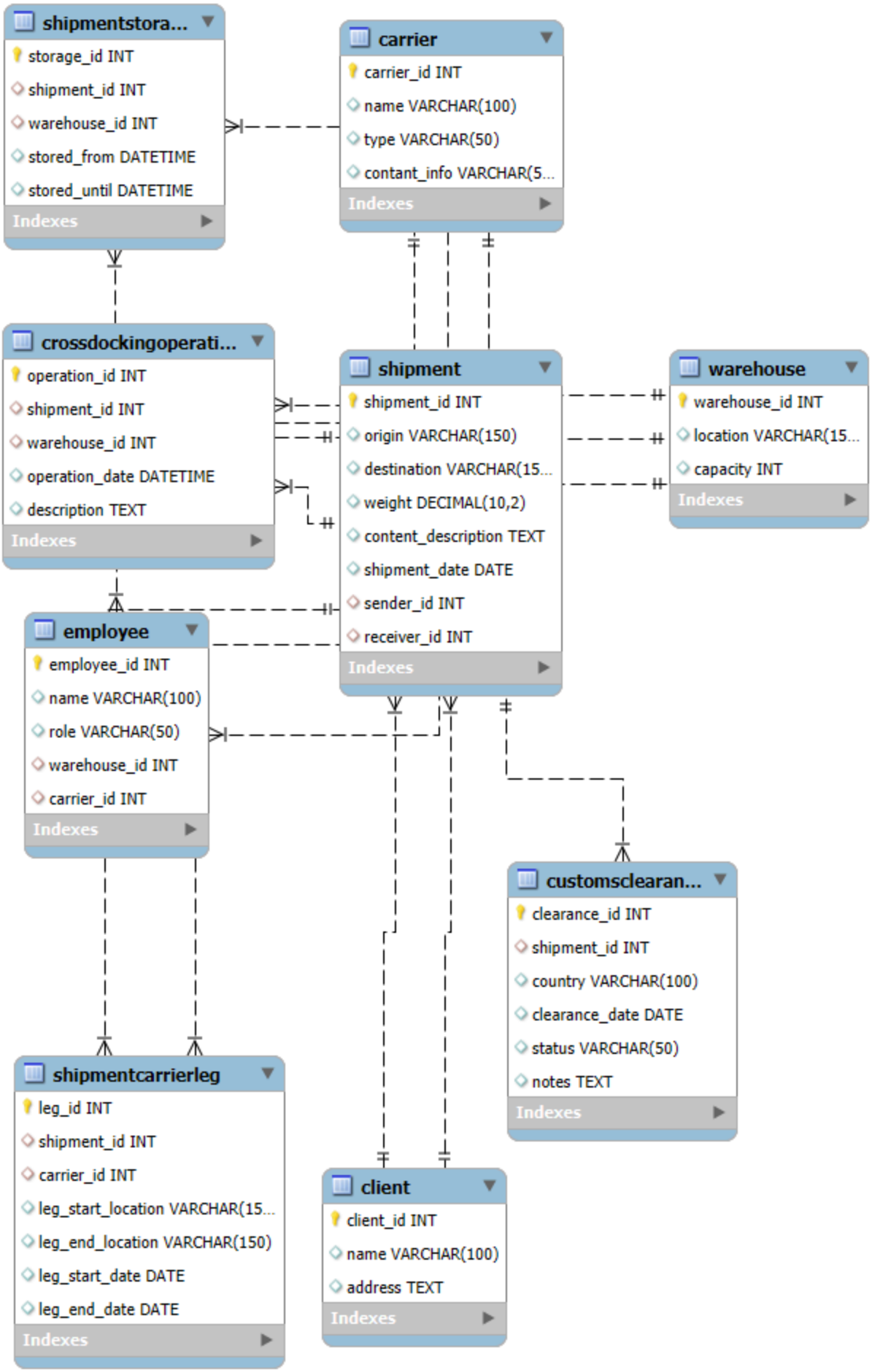
```
    name VARCHAR(100),
    address TEXT
);
CREATE TABLE Carrier (
    carrier_id INT PRIMARY KEY,
    name VARCHAR(100),
    type eg:('air', 'land', 'sea'),
   contant_info
  );
CREATE TABLE Warehouse (
    warehouse_id INT PRIMARY KEY,
    location VARCHAR(150),
    capacity INT
);
CREATE TABLE Employee (
    employee_id INT PRIMARY KEY,
    name VARCHAR(100),
    role VARCHAR(50),
    warehouse_id INT,
    carrier_id INT,
    CHECK ((warehouse_id IS NOT NULL AND carrier_id IS NULL) OR
          (warehouse_id IS NULL AND carrier_id IS NOT NULL)),
    FOREIGN KEY (warehouse_id) REFERENCES Warehouse(warehouse_id),
    FOREIGN KEY (carrier_id) REFERENCES Carrier(carrier_id)
);

CREATE TABLE Shipment (
    shipment_id INT PRIMARY KEY,
    origin VARCHAR(150),
    destination VARCHAR(150),
    weight DECIMAL(10,2),
    content_description TEXT,
    shipment_date DATE,
    sender_id INT,
    receiver_id INT,
    FOREIGN KEY (sender_id) REFERENCES Client(client_id),
    FOREIGN KEY (receiver_id) REFERENCES Client(client_id)
);

CREATE TABLE ShipmentCarrierLeg (
```

```
    leg_id INT PRIMARY KEY,
    shipment_id INT,
    carrier_id INT,
    leg_start_location VARCHAR(150),
    leg_end_location VARCHAR(150),
    leg_start_date DATE,
    leg_end_date DATE,
    FOREIGN KEY (shipment_id) REFERENCES Shipment(shipment_id),
    FOREIGN KEY (carrier_id) REFERENCES Carrier(carrier_id)
);

CREATE TABLE ShipmentStorage (
    storage_id INT PRIMARY KEY,
    shipment_id INT,
    warehouse_id INT,
    stored_from DATETIME,
    stored_until DATETIME,
    FOREIGN KEY (shipment_id) REFERENCES Shipment(shipment_id),
    FOREIGN KEY (warehouse_id) REFERENCES Warehouse(warehouse_id)
);

CREATE TABLE CustomsClearance (
    clearance_id INT PRIMARY KEY,
    shipment_id INT,
    country VARCHAR(100),
    clearance_date DATE,
    status VARCHAR(50),
    notes TEXT,
    FOREIGN KEY (shipment_id) REFERENCES Shipment(shipment_id)
);

CREATE TABLE CrossDockingOperation (
    operation_id INT PRIMARY KEY,
    shipment_id INT,
    warehouse_id INT,
    operation_date DATETIME,
    description TEXT,
    FOREIGN KEY (shipment_id) REFERENCES Shipment(shipment_id),
    FOREIGN KEY (warehouse_id) REFERENCES Warehouse(warehouse_id)
);
```

INSERT VALUES

```
-- Clients
INSERT INTO Client VALUES (1, 'Aishwarya Exports', 'Hyderabad, India');
INSERT INTO Client VALUES (2, 'Narashima Imports', 'Berlin, Germany');

-- Carriers
INSERT INTO Carrier VALUES (1, 'SkyJet Logistics', 'air', 'skyjet@example.com');
INSERT INTO Carrier VALUES (2, 'OceanTrans Co.', 'sea',
'oceantrans@example.com');

-- Warehouses
INSERT INTO Warehouse VALUES (1, 'Mumbai Port', 5000);
INSERT INTO Warehouse VALUES (2, 'Hamburg Dock', 3000);

-- Employees
INSERT INTO Employee VALUES (1, 'Rithika', 'Manager', 1, NULL);  -- warehouse staff
INSERT INTO Employee VALUES (2, 'Sahithya', 'Pilot', NULL, 1);   -- carrier crew

-- Shipments
INSERT INTO Shipment VALUES (
  1001, 'Hyderabad, India', 'Berlin, Germany', 200.50,
  'Electronic Components', '2025-06-10', 1, 2
);

-- Shipment Carrier Legs
INSERT INTO ShipmentCarrierLeg VALUES (
  1, 1001, 1, 'Hyderabad Airport', 'Dubai Airport', '2025-06-10', '2025-06-11'
);
INSERT INTO ShipmentCarrierLeg VALUES (
  2, 1001, 2, 'Dubai Port', 'Hamburg Port', '2025-06-12', '2025-06-15'
);

-- Shipment Storage
INSERT INTO ShipmentStorage VALUES (
  1, 1001, 1, '2025-06-09 08:00:00', '2025-06-10 05:00:00'
);
INSERT INTO ShipmentStorage VALUES (
  2, 1001, 2, '2025-06-16 08:00:00', '2025-06-18 12:00:00'
);
```

```
-- Customs Clearance
INSERT INTO CustomsClearance VALUES (
  1, 1001, 'Germany', '2025-06-16', 'Cleared', 'No issues found.'
);

-- Cross Docking Operation
INSERT INTO CrossDockingOperation VALUES (
  1, 1001, 2, '2025-06-16 10:00:00', 'Transferred to outbound truck for delivery.'
);
```


shipmentstora...
storage_id INT
shipment_id INT
warehouse_id INT
stored_from DATETIME
stored_until DATETIME
Indexes
carrier
carrier_id INT
name VARCHAR(100)
type VARCHAR(50)
contant_info VARCHAR(5...
Indexes
crossdockingoperati...
operation_id INT
shipment_id INT
warehouse_id INT
operation_date DATETIME
description TEXT
Indexes
shipment
shipment_id INT
origin VARCHAR(150)
destination VARCHAR(15...
weight DECIMAL(10,2)
content_description TEXT
shipment_date DATE
sender_id INT
receiver_id INT
Indexes
warehouse
warehouse_id INT
location VARCHAR(15...
capacity INT
Indexes
employee
employee_id INT
name VARCHAR(100)
role VARCHAR(50)
warehouse_id INT
carrier_id INT
Indexes
customsclearan...
clearance_id INT
shipment_id INT
country VARCHAR(100)
clearance_date DATE
status VARCHAR(50)
notes TEXT
Indexes
shipmentcarrierleg
leg_id INT
shipment_id INT
carrier_id INT
leg_start_location VARCHAR(15...
leg_end_location VARCHAR(150)
leg_start_date DATE
leg_end_date DATE
Indexes
client
client_id INT
name VARCHAR(100)
address TEXT
Indexes