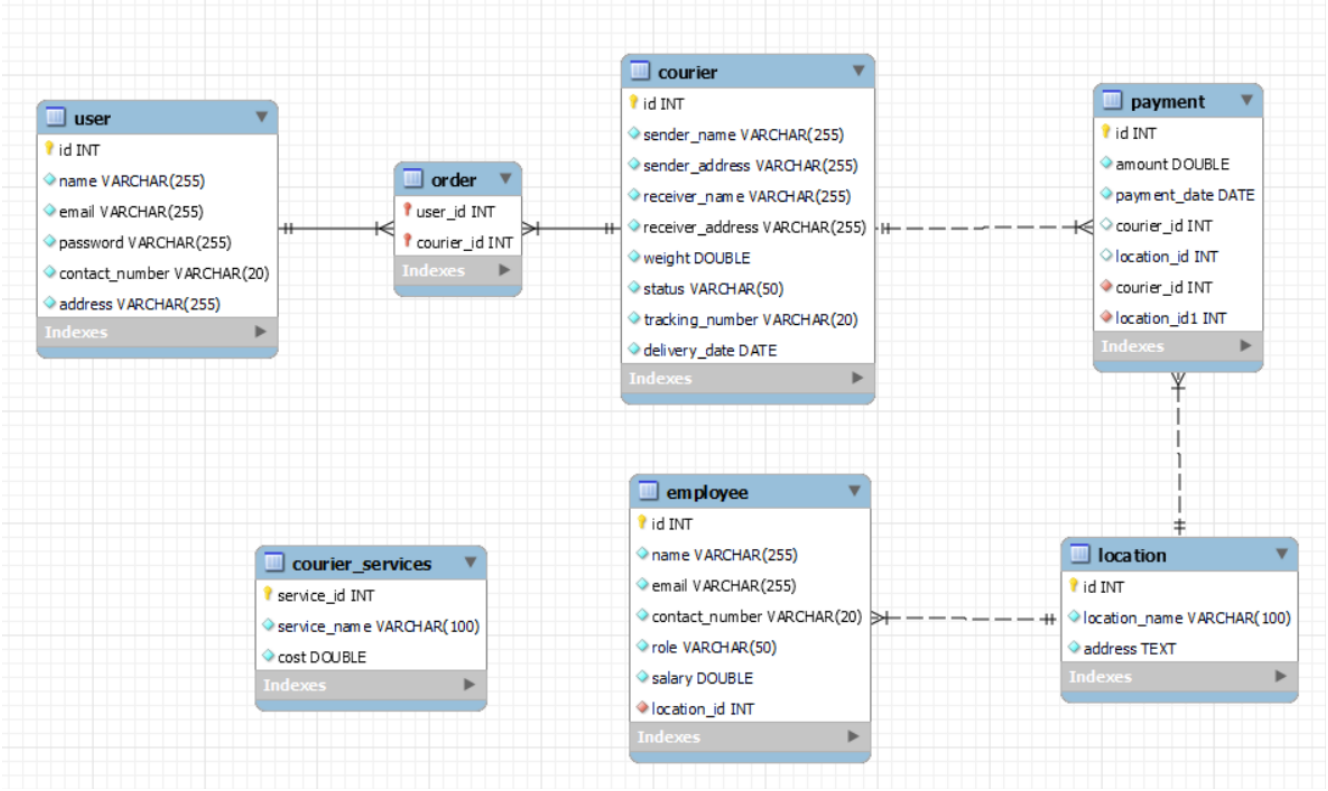
ASSIGNMENT 2 – COURIER MANAGEMENT SYSTEM

ER DIAGRAM:



Queries:

```
create database courierdb;
use courierdb;

-- Task 1 : Creation of database and insertion of values
-- creation of tables

create table user(id INT PRIMARY KEY NOT NULL,
name VARCHAR(255) NOT NULL,
email VARCHAR(255) UNIQUE NOT NULL,
password VARCHAR(255) NOT NULL,
contact_number VARCHAR(20) NOT NULL,
address TEXT NOT NULL);

create table courier(id INT PRIMARY KEY NOT NULL,
sender_name VARCHAR(255) NOT NULL,
sender_address TEXT NOT NULL,
receiver_name VARCHAR(255) NOT NULL,
receiver_address TEXT NOT NULL,
weight DECIMAL(5, 2) NOT NULL,
status VARCHAR(50) NOT NULL,
tracking_number VARCHAR(20) UNIQUE NOT NULL,
delivery_date DATE);
```

```
create table courier_services(service_id INT PRIMARY KEY NOT NULL,
service_name VARCHAR(100) NOT NULL,
cost DECIMAL(8, 2)NOT NULL);

create table employee(id INT PRIMARY KEY NOT NULL,
name VARCHAR(255) NOT NULL,
email VARCHAR(255) UNIQUE NOT NULL,
contact_number VARCHAR(20) NOT NULL,
role VARCHAR(50) NOT NULL,
salary DECIMAL(10, 2) NOT NULL);

CREATE TABLE location (
   id INT PRIMARY KEY NOT NULL,
   location_name VARCHAR(100) NOT NULL,
   address TEXT NOT NULL
);

create table payment(id INT PRIMARY KEY NOT NULL,
amount DECIMAL(10, 2) NOT NULL,
payment_date DATE NOT NULL,
courier_id INT NOT NULL,
location_id INT NOT NULL,
FOREIGN KEY (courier_id) REFERENCES courier(id),
FOREIGN KEY (location_id) REFERENCES location(id));
```

-- insertion of values

```
insert into user(id, name, email, password, contact_number, address) values
(1, 'John Doe', 'john.doe@example.com', 'password123', '1234567890', '123 Main St, City'),
(2, 'Jane Smith', 'jane.smith@example.com', 'pass456', '9876543210', '456 Oak St, Town'),
(3, 'Alice Johnson', 'alice.j@example.com', 'securepass', '5551234567', '789 Pine St, Village'),
(4, 'Bob Wilson', 'bob.wilson@example.com', 'bobpass', '1112223333', '101 Maple St, City'),
(5, 'Eva Martinez', 'eva.m@example.com', 'evapass', '9998887777', '456 Elm St, Town'),
(6, 'Charlie Brown', 'charlie.b@example.com', 'charliepass', '3334445555', '789 Cedar St, Village'),
(7, 'Sophie White', 'sophie.w@example.com', 'sophiepass', '6667778888', '123 Pine St, City'),
(8, 'David Lee', 'david.lee@example.com', 'davidpass', '4445556666', '456 Oak St, Town'),
(9, 'Grace Taylor', 'grace.t@example.com', 'gracepass', '7778889999', '789 Elm St, Village'),
(10, 'Michael Turner', 'michael.t@example.com', 'michaelpass', '5556667777', '101 Cedar St, City');

insert into courier(id, sender_name, sender_address, receiver_name, receiver_address, weight, status,
tracking_number, delivery_date) values
(111, 'Sender A', '123 Sender St, City', 'Receiver X', '456 Receiver St, Town', 2.5, 'In Transit', 'TN123457',
'2024-03-01'),
(102, 'Sender B', '789 Sender St, Village', 'Receiver Y', '101 Receiver St, City', 3.2, 'Delivered',
'TN789012', '2024-03-02'),
```

(103, 'Sender C', '456 Sender St, Town', 'Receiver Z', '789 Receiver St, Village', 1.8, 'Pending', 'TN345678', NULL),
(104, 'Sender D', '101 Sender St, City', 'Receiver W', '789 Receiver St, Village', 2.0, 'In Transit', 'TN567890', '2024-03-03'),
(105, 'Sender E', '456 Sender St, Town', 'Receiver V', '101 Receiver St, City', 3.5, 'Delivered', 'TN234567', '2024-03-04'),
(106, 'Sender F', '789 Sender St, Village', 'Receiver U', '123 Receiver St, City', 1.2, 'Delivered', 'TN890123', '2024-03-05'),
(107, 'Sender G', '101 Sender St, City', 'Receiver T', '456 Receiver St, Town', 2.8, 'Pending', 'TN456789', NULL),
(108, 'Sender H', '456 Sender St, Town', 'Receiver S', '789 Receiver St, Village', 1.5, 'Delivered', 'TN012345', '2024-03-06'),
(109, 'Sender I', '789 Sender St, Village', 'Receiver R', '101 Receiver St, City', 2.3, 'In Transit', 'TN678901', '2024-03-07'),
(110, 'Sender J', '101 Sender St, City', 'Receiver Q', '123 Receiver St, City', 4.0, 'Pending', 'TN345679', NULL);

insert into courier_services(service_id, service_name, cost) values
(1, 'Standard', 10.00),
(2, 'Express', 20.00),
(3, 'Same Day', 30.00),
(4, 'Overnight', 25.00),
(5, 'Two-Day', 15.00),
(6, 'Economy', 8.00),
(7, 'Priority', 22.00),
(8, 'Next Day', 18.00),
(9, 'Ground', 12.00),
(10, 'International', 40.00);

insert into employee(id, name, email, contact_number, role, salary) values
(201, 'Mukesh', 'manager@example.com', '1112223344', 'Manager', 50000.00),
(202, 'Durai', 'driver@example.com', '5556667788', 'Driver', 30000.00),
(203, 'Canesan', 'clerk@example.com', '9990001122', 'Clerk', 25000.00),
(204, 'Senthil', 'supervisor@example.com', '1234567890', 'Supervisor', 35000.00),
(205, 'Anand', 'analyst@example.com', '9876543210', 'Analyst', 40000.00),
(206, 'Thomas', 'technician@example.com', '5551234567', 'Technician', 32000.00),
(207, 'Arun', 'assistant@example.com', '2223334444', 'Assistant', 28000.00),
(208, 'Carmel', 'coordinator@example.com', '8889990000', 'Coordinator', 31000.00),
(209, 'Srinivas', 'specialist@example.com', '4445556666', 'Specialist', 38000.00),
(210, 'Coumarane', 'consultant@example.com', '6667778888', 'Consultant', 42000.00);

insert into location(id, location_name, address) values
(301, 'Warehouse', '789 Warehouse St, City'),
(302, 'Office', '101 Office St, Town'),
(303, 'Hub', '123 Hub St, Village'),

```
(304, 'Distribution Center', '456 Distribution St, City'),
(305, 'Facility', '789 Facility St, Village'),
(306, 'Station', '101 Station St, Town'),
(307, 'Center', '456 Center St, City'),
(308, 'Depot', '789 Depot St, Village'),
(309, 'Hub1', '101 Hub St, Town'),
(310, 'Warehouse1', '123 Warehouse St, City');


insert into payment(id, amount, payment_date, courier_id, location_id) values
(401, 15.00, '2024-03-03', 101, 301),
(402, 25.00, '2024-03-04', 102, 302),
(403, 20.00, '2024-03-05', 103, 303),
(404, 18.50, '2024-03-06', 104, 304),
(405, 30.00, '2024-03-07', 105, 305),
(406, 22.50, '2024-03-08', 106, 306),
(407, 15.00, '2024-03-09', 107, 307),
(408, 28.00, '2024-03-10', 108, 308),
(409, 24.50, '2024-03-11', 109, 309),
(410, 35.00, '2024-03-12', 110, 310);

-- Task 2 : select, where

-- Q1. List all customers
select name as customers from user;
update courier set sender_name='Sender A' where id=107;

-- Q2. List all orders for a specific customer
select * from courier where sender_name='Sender A';

-- Q3. List all couriers
select * from courier;

-- Q4. List all packages for a specific order
select * from courier where id=111 or id=102;

-- Q5. List all deliveries for a specific courier
select * from courier where status='delivered';

-- Q6. List all undelivered packages
select * from courier where status!='delivered';
update courier set delivery_date = '2024-02-28' where id=111;

-- Q7. List all packages that are scheduled for delivery today
select * from courier where date(delivery_date)= curdate();
```

```
-- Q8. List all packages with a specific status:
select * from courier where status='pending';

-- Q9. Calculate the total number of packages for each courier.
select id, count(id) as total_packages
    from courier
    group by id;

-- Q10. Find the average delivery time for each courier
-- not provided with any specific column with order_Date

-- Q11. List all packages with a specific weight range:
select id,weight
    from courier
    where weight between 1 and 2;

-- Q12. Retrieve employees whose names contain 'John'
select id, name
    from employee
    where name='%John%';

-- Q13. Retrieve all courier records with payments greater than $50.
select c.*
    from courier c join payment p
    on c.id=p.courier_id
    where p.amount > 50.00;

-- Task 3:.....

-- Q14. Find the total number of couriers handled by each employee.
-- cannot be found since there is no foreign key that connects employee table with courier table

-- Q15. Calculate the total revenue generated by each location
select l.location_name, sum(p.amount)
    from location l join payment p
    on l.id=p.location_id
    group by p.location_id;

-- Q16. Find the total number of couriers delivered to each location.
select l.location_name, count(p.courier_id) as no_of_couriers
    from location l join payment p
    on l.id=p.location_id
    group by p.courier_id;

-- Q17. Find the courier with the highest average delivery time:
-- couldnt find since there is no info about order_date
```

```sql
-- Q18. Find Locations with Total Payments Less Than a Certain Amount
select l.location_name, sum(p.amount)
    from location l join payment p
    on l.id=p.location_id
    group by p.location_id
    having sum(p.amount) > 30.00;

-- Q19. Calculate Total Payments per Location
select location_id, sum(amount)
    from payment
    group by location_id;

-- Q20. Retrieve couriers who have received payments totaling more than $1000 in a specific location (LocationID = X):
select c.*
    from courier c join payment p
    on c.id=p.courier_id
    where p.amount>1000 and p.location_id-301;

-- Q21. Retrieve couriers who have received payments totaling more than $1000 after a certain date (PaymentDate > 'YYYY-MM-DD'):
select c.*
    from courier c join payment p
    on c.id=p.courier_id
    where p.amount>1000
    having p.payment_date > '2023-11-01';

-- Q22. Retrieve locations where the total amount received is more than $5000 before a certain date (PaymentDate > 'YYYY-MM-DD')
select c.*
    from courier c join payment p
    on c.id=p.courier_id
    where p.amount>5000
    having p.payment_date < '2023-11-01';

-- Task 4: .....

-- Q23. Retrieve Payments with Courier Information
select c.*, p.amount
    from courier c join payment p
    on c.id=p.courier_id;

-- Q24. Retrieve Payments with Location Information
select p.amount, l.*
    from payment p join location l
    on l.id=p.location_id;
```

```
 -- Q25. Retrieve Payments with Courier and Location Information
select c.*,l.*
    from courier c join payment p
    on c.id=p.courier_id join location l
    on l.id=p.location_id;

-- Q26. List all payments with courier details
select p.*,c.*
    from payment p join courier c
    on c.id=p.courier_id;

-- Q27. Total payments received for each courier
select courier_id, sum(amount)
    from payment
    group by courier_id;

-- Q28. List payments made on a specific date
select * from payment
    where payment_date='2023-03-03';

-- Q29. Get Courier Information for Each Payment
select p.id as payment_id ,c.*
    from courier c join payment p
    on c.id=p.courier_id;

-- Q30. Get Payment Details with Location
select p.*, l.location_name
    from payment p join location l
    on l.id=p.location_id;

-- Q31. Calculating Total Payments for Each Courier
select courier_id, sum(amount) as total_payment
    from payment
    group by courier_id;

-- Q32. List Payments Within a Date Range
select * from payment
    where payment_date between '2024-01-01' and '2024-03-01';

-- Q33. Retrieve a list of all users and their corresponding courier records, including cases where there are no matches on either side
-- couldnt retreive since there is no foreign key reference with user and courier table

-- Q34. Retrieve a list of all couriers and their corresponding services, including cases where there are no matches on either side
-- couldnt retreive since there is no info about courier service in the courier table
```

```
-- Q35. Retrieve a list of all employees and their corresponding payments, including cases where there are no matches on either side
-- couldnt retreive since there is no linkage between employee and payment

-- Q36. List all users and all courier services, showing all possible combinations.
select * from user cross join courier_services;

-- Q37. List all employees and all locations, showing all possible combinations:
select e.name, l.location_name
    from employee e cross join location l;

-- Q38. Retrieve a list of couriers and their corresponding sender information (if available)
select id, sender_name,sender_address from courier;

-- Q39. Retrieve a list of couriers and their corresponding receiver information (if available):
select id, reciever_name,receiver_address from courier;

-- Q40. Retrieve a list of couriers along with the courier service details (if available):
select id, weight, status, tracking_number, delivery_date from courier;

-- Q41. Retrieve a list of employees and the number of couriers assigned to each employee
-- couldnt retrieve since there is no info about employee related to courier

-- Q42. Retrieve a list of locations and the total payment amount received at each location
select l.location_name, sum(p.amount) as total_amount
    from location l left join payment p
    on l.id=p.location_id
    group by p.location_id;

-- Q43. Retrieve all couriers sent by the same sender (based on SenderName).
select * from courier
    where sender_name like '%A%';

-- Q44. List all employees who share the same role.
select name from employee
    where role='manager';

-- Q45. Retrieve all payments made for couriers sent from the same location.
select p.*
    from payment p join location l
    on l.id=p.location_id
    group by p.location_id
    having count(p.location_id) >1;
```

```sql
-- Q46. Retrieve all couriers sent from the same location (based on SenderAddres).
select * from courier
    where sender_address like '%Sender St%';

-- Q47. List employees and the number of couriers they have delivered:
-- no info about relation between courier and employees

-- Q48. Find couriers that were paid an amount greater than the cost of their respective courier services
-- info not provided about each courier service in couriers table

-- Scope.....

-- Q49. Find couriers that have a weight greater than the average weight of all couriers
select * from courier
    where weight > (select avg(weight) from courier);

-- Q50. Find the names of all employees who have a salary greater than the average salary
select name from employee
    where salary > (select avg(salary) from employee);

-- Q51. Find the total cost of all courier services where the cost is less than the maximum cost
select sum(cost) as tot_cost
    from courier_services
    where cost < (select max(cost) from courier_services);

-- Q52. Find all couriers that have been paid for
select c.*
    from courier c
    join payment p
    on c.id = p.courier_id
    where p.courier_id is not null;

-- Q53. Find the locations where the maximum payment amount was made
select l.location_name
    from location l join payment p
    on l.id=p.location_id
    where p.amount = (select max(amount) from payment);

-- Q54. Find all couriers whose weight is greater than the weight of all couriers sent by a specific sender (e.g., 'SenderName'):
select * from courier
    where weight > (select sum(weight) from courier) and
    sender_name='Sender A';
```