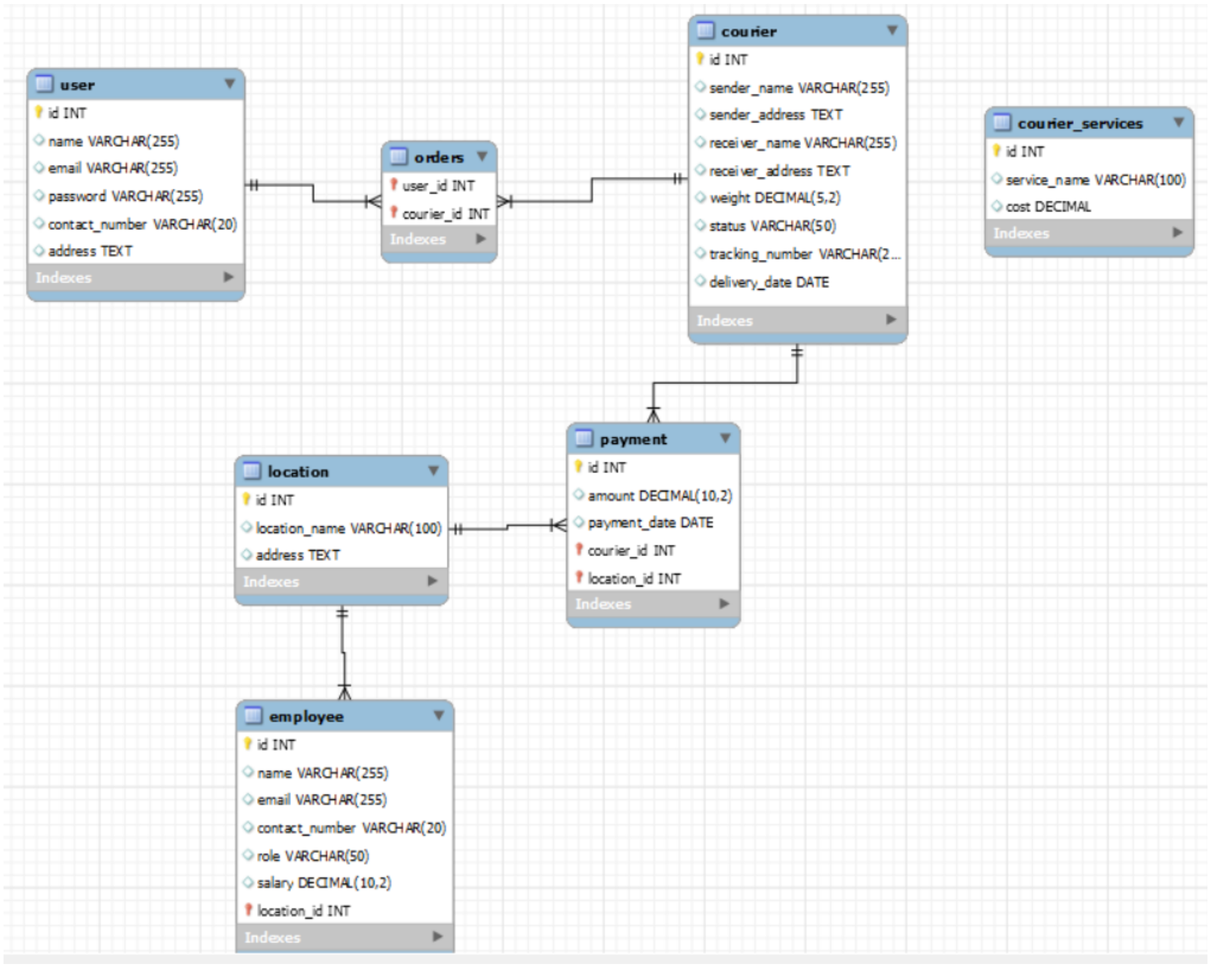
COURIER MANAGEMENT SYSTEM



<u>CODE:</u>

```
use courier_hex_feb_24;

# insertion

insert into user (name, email, contact_number) values
('diya', 'diya@gmail.com', 4545545455),
('dev', 'dev@gmail.com', 3434434344),
('tara', 'tara@gmail.com', 2323323233),
('atul', 'atul@gmail.com', 1212212122),
('ajay', 'ajay@gmail.com', 4545455454);
```

```
insert into courier_services (service_name, cost) values
('dtdc' , 2000),
('dhl', 1800),
('bluedart', 2500);

insert into courier (sender_name, sender_address, receiver_name, receiver_address, weight,
tracking_number) values
('john', 'chennai', 'jack', 'delhi', 500.2, '1234'),
('gorya', 'delhi', 'thyme', 'pondy', 100.0, '2345'),
('sona', 'pondy', 'soha', 'mumbai', 632.3, '3456'),
('abir', 'mumbai', 'kunal', 'chennai', 765.06, '4567'),
('arjun', 'chennai', 'krish', 'pune', 864.4, '5678');

insert into location (location_name, address) values
('t nagar', 'chennai'),
('white town', 'pondy'),
('bandra', 'mumbai');

insert into employee (name, email, role, salary,location_id) values
('misti', 'misti@gmail.com', 'admin', 20000,1),
('kuku', 'kuhu@gmail.com', 'delivery', 15000,2),
('benba', 'benba@gmail.com', 'delivery', 15000,3),
('rani', 'rani@gmail.com', 'delivery', 15000,2);

insert into payment (amount, payment_date, courier_id, location_id) values
(2400.65, '2024-03-02', 1,2),
(3426.23, '2024-02-28',1,3),
(7457.76, '2024-03-06',2,3),
(2165.43, '2024-02-21',3,1);
```

```
# Task 2

-- 1. List all customers:
select * from user;

-- 2. List all orders for a specific customer:
select * from courier
where sender_name = 'gorya';

-- 3. List all couriers:
 select * from courier;

-- 4. List all packages for a specific order:
select * from courier
where id = 1;

-- 5. List all deliveries for a specific courier:
select * from courier
where sender_name = 'sona';

-- 6. List all undelivered packages:
select * from courier
where status != 'delivered';

-- 7. List all packages that are scheduled for delivery today:
select * from courier
where delivery_date = current_date;

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

```
-- 9. Calculate the total number of packages for each courier.
select id, COUNT(*) AS packages
from courier
group by id;

-- 11. List all packages with a specific weight range:
select * from courier
where weight between 100 and 700;

-- 12. Retrieve employees whose names contain 'John'
select name
from employee
where name like '%john%';

-- 13. Retrieve all courier records with payments greater than $50.
select c.id, c.sender_name, c.receiver_name, p.amount
from courier c
join payment p on c.id = p.courier_id
where p.amount>50;


# Task 3

-- 14. Find the total number of couriers handled by each employee.
select e.name, count(c.id)
from employee e
JOIN courier c ON e.id = c.employee_id
group by e.name; -- if courier had employee id as foreign key
```

```sql
-- 15. Calculate the total revenue generated by each location
select l.id , l.location_name, SUM(p.amount)
from location l
JOIN payment p ON l.id = p.location_id
group by l.id, l.location_name;

-- 16. Find the total number of couriers delivered to each location.
   # update courier set status='delivered' where id=2;
select COUNT(c.id)
from courier c
JOIN payment p ON p.courier_id=c.id
JOIN location l ON p.location_id=l.id
where c.status='delivered';

-- 18. Find Locations with Total Payments Less Than a Certain Amount
select l.id, SUM(p.amount) as payments
from location l
INNER JOIN payment p ON l.id = p.location_id
group by l.id
having SUM(p.amount) < 2000;

-- 19. Calculate Total Payments per Location
select distinct l.id, SUM(p.amount) as payments
from location l
INNER JOIN payment p ON l.id = p.location_id
group by l.id;

-- 20. Retrieve couriers who have received payments totaling more than $1000 in a specific location
(LocationID = X):
select c.id, SUM(p.amount) as payments
from courier c
```

```sql
JOIN payment p ON c.id = p.courier_id
where p.location_id = 1
group by c.id
having payments > 1000;


-- 21. Retrieve couriers who have received payments totaling more than $1000 after a certain date
(PaymentDate > 'YYYY-MM-DD'):
select c.id, SUM(p.amount) as payments
from courier c
JOIN payment p ON c.id = p.courier_id
where p.payment_date>'2024-01-01'
group by c.id
having payments > 1000;


-- 22. Retrieve locations where the total amount received is more than $5000 before a certain date
(PaymentDate > 'YYYY-MM-DD')
select l.id , l.location_name, SUM(p.amount) as payments
from location l
JOIN payment p ON l.id = p.location_id
where p.payment_date > '2024-01-01'
group by l.id
having payments > 5000;



# Task 4
-- 23. Retrieve Payments with Courier Information
select *
from payment p
LEFT JOIN courier c ON p.courier_id = c.id;



-- 24. Retrieve Payments with Location Information
```

```sql
select *
from payment p
JOIN location l ON p.location_id= l.id;

-- 25. Retrieve Payments with Courier and Location Information
select *
from payment p
JOIN  courier c ON p.courier_id = c.id
JOIN  location l ON p.location_id= l.id;

-- 26. List all payments with courier details
select *
from payment p
LEFT JOIN courier c ON  p.courier_id = c.id;

-- 27. Total payments received for each courier
select c.id, sum(p.amount) as payments
from payment p
LEFT JOIN courier c ON  p.courier_id = c.id
group by c.id;

-- 28. List payments made on a specific date
select *
from payment
where payment_date ='2023-03-02';

-- 29. Get Courier Information for Each Payment
select p.id , c.ID , c.sender_name , c.receiver_name , c.weight
from courier c
JOIN payment p ON  p.courier_id = c.id
group by p.id;
```

```
-- 30. Get Payment Details with Location

select p.id, p.amount , p.payment_date ,l.location_name

from payment p

LEFT JOIN location l ON p.location_id= l.id;


-- 31. Calculating Total Payments for Each Courier

select c.id, sum(p.amount)

from payment p

LEFT JOIN courier c ON  p.courier_id = c.id

group by c.id;


-- 32. List Payments Within a Date Range

select id , amount ,payment_date

from payment

where payment_date between '2023-03-06' AND '2024-01-30';


-- 33. Retrieve a list of all users and their corresponding courier records, including cases where there are no matches on either side

select *

from user u

LEFT JOIN courier c ON u.id = c.user_id;


-- 34. Retrieve a list of all couriers and their corresponding services, including cases where there are no matches on either side

select *

from courier c

LEFT JOIN courier_services cs ON cs.id=c.courierservices_id; -- if courier and courier_service table had relation


-- 35. Retrieve a list of all employees and their corresponding payments, including cases where there are no matches on either side
```

```
select *

from employee e

LEFT JOIN payment p ON e.location_id =p.location_id;


-- 36. List all users and all courier services, showing all possible combinations.

select *

from user

CROSS JOIN courier;


-- 37. List all employees and all locations, showing all possible combinations:

select *

from employee

CROSS JOIN location;


-- 38. Retrieve a list of couriers and their corresponding sender information (if available)

select id , sender_name, sender_address

from courier;


-- 39. Retrieve a list of couriers and their corresponding receiver information (if available):

select id , receiver_name ,receiver_address

from courier;


-- 40. Retrieve a list of couriers along with the courier service details (if available):

select c.id ,cs.id, cs.id, cs.cost

from courier c

LEFT JOIN courier_services cs ON cs.id=c.courierservices_id; -- if courier and courier_service table
had relation


-- 41. Retrieve a list of employees and the number of couriers assigned to each employee:

select e.id, e.name, COUNT(c.id)

from employee e
```

```
LEFT JOIN location l ON l.id = e. location_id

JOIN payment p ON l.id = p.location_id

JOIN courier c on c.id = p.courier_id;


-- 42. Retrieve a list of locations and the total payment amount received at each location:

select l.id , l.location_name , sum(p.amount)

from location l

JOIN payment p ON l.id = p.location_id

group by l.id;


-- 43. Retrieve all couriers sent by the same sender (based on SenderName).

select *

from courier

where sender_nmae ='gorya';
```