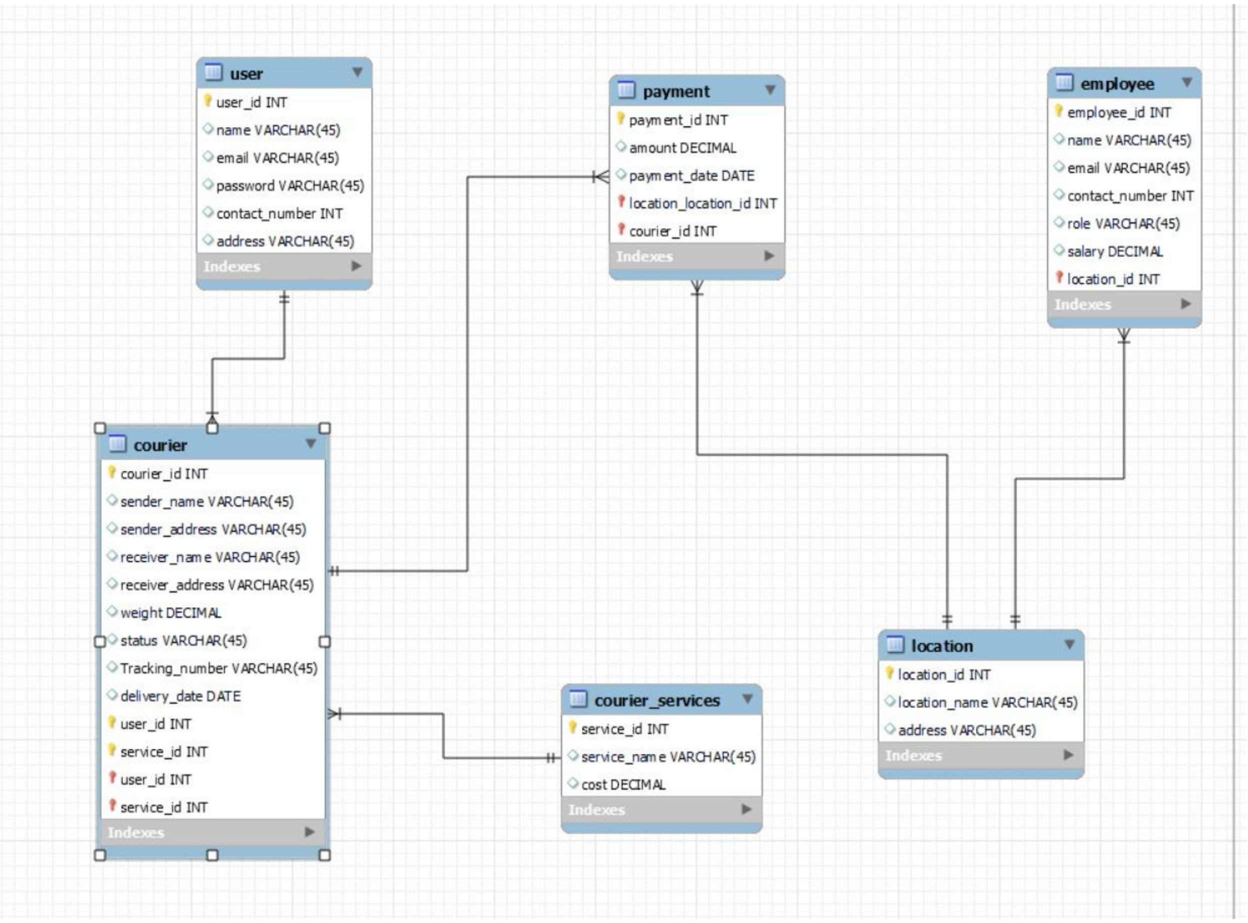
COURIER MANAGEMENT SYSTEM

ER DIAGRAM



```
use CourierManagementSystem;

show tables;

insert into user(name,email,password,contact_number,address)
values("Arun","arun@gmail.com","arun123","1111111111","Bangalore"),

("Abinaya","abii@gmail.com","abi123","2222222222","Chennai"),

("Nams","nams@gmail.com","anash23","3333333333","Chennai"),

("Ajith","ajii@gmail.com","arji123","4444444444","Coimbatore"),

("Amarnath","amargmail.com","amar123","5555555555","Coimbatore");
```

```
select * from user;

insert into courier_services(service_name,cost)
values("AAA",200.20),("BAC",156.56),("ABC",340.56),("CBA",120.45),("BCA",150.54);

select * from courier_services;

insert into
courier(sender_name,sender_address,receiver_name,receiver_address,weight,status,Tracking_number,delivery_date,
user_user_id,courier_services_service_id)

 values("Vasanth","Coimbatore","Viki","Chennai",5.23,"Onprogress","11345","2024-04-06",1,2),

("Varun","Bangalore","Dharun","Chennai",7.25,"Delivered","56536","2023-06-05",1,3),

("Giri","Pune","Joy","Bangalore",8.23,"Delivered","35475","2023-08-09",1,3),

("Haresh","Coimbatore","Ram","Pune",9.23,"Onprogress","65865","2024-09-08",2,4),

("Jack","Coimbatore","Rose","Delhi",7.73,"Delivered","87845","2023-02-02",2,3);

select * from courier;

insert into location(location_name,address) values("Pune","Pune"),

("Bangalore","Bangalore"),

("Pune","Pune"),

("Coimbatore","Coimbatore"),

("Coimbatore","Coimbatore");

select * from location;

insert into employee(name,email,contact_number,role,salary,location_location_id)
values("Manikam","maik@gmail.com","1212121212","Analyst",40000.00,1),

("Manoj","manoj@gmail.com","1313131313","Developer",39000.00,2),

("Varun","varun@gmail.com","1414141414","Treasurer",38000.00,3),

("Krithik","krithik@gmail.com","1515151515","Manager",50000.00,1),

("Nandha","nandhh@gmail.com","1616161616","Deliveryman",25000.00,2);

select * from employee;

insert into payment (amount,payment_date,location_location_id,courier_courier_id)values

(150.00,"2024-09-06",1,11),

(250.00,"2023-07-05",2,12),

(500.00,"2023-09-09",3,13),

(350.00,"2024-08-08",4,14),

(250.00,"2023-04-02",5,15);

select * from payment;
```

```sql
-- Task 2

-- 1. List all customers:

select * from user;

-- 2. List all orders for a specific customer:

select * from courier where sender_name="Jack";

-- 3. List all couriers:

select * from courier;

-- 4. List all packages for a specific order:

select * from courier where courier_id = 1;

-- 5. List all deliveries for a specific courier:

select * from courier where status="Delivered";

-- 6. List all undelivered packages:

select * from courier where Status = "Onprogress";

-- 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 = "Delivered";

-- 9. Calculate the total number of packages for each courier.

select courier_id,count(courier_id) as total_num_of_packages from courier group by courier_id;

-- 10. Find the average delivery time for each courier

select c.courier_id , avg(abs(c.delivery_date-p.payment_date)) as avg_delivery_time from courier c,payment p
where c.courier_id=p.courier_courier_id group by c.courier_id;

-- 11. List all packages with a specific weight range:

select * from courier where weight between 2 and 3;

-- 12. Retrieve employees whose names contain 'John'

select * from employee where name  like "%John%";

-- 13. Retrieve all courier records with payments greater than $50.

select
c.sender_name,c.sender_address,c.receiver_name,c.receiver_address,c.weight,c.status,c.Tracking_number,c.delivery_date

from courier c, payment p where c.courier_id=p.courier_courier_id and amount>50;
```

```sql
-- Task 3

-- 14. Find the total number of couriers handled by each employee.

select e.name , e.employee_id ,count(c.courier_id) from courier c , employee e, payment p , location l where c.courier_id=p.courier_courier_id and

l.location_id=p.location_location_id and l.location_id=e.location_location_id group by e.name;

-- 15. Calculate the total revenue generated by each location

select location_name ,sum(amount) as total_revenue from location l, payment p

where l.location_id=p.location_location_id group by location_name;

-- 16. Find the total number of couriers delivered to each location.

select l.location_name ,count(c.courier_id) as total_no_of_couriers_delivered from location l , courier c , payment p where

c.courier_id =p.courier_courier_id AND l.location_id =p.location_location_id group by l.location_name;

-- 17. Find the courier with the highest average delivery time.

select c.courier_id , avg(abs(c.delivery_date-p.payment_date) )as average_time from courier c , payment p where c.courier_id=p.courier_courier_id

group by c.courier_id order by courier_id asc limit 0, 1;

-- 18. Find Locations with Total Payments Less Than a Certain Amount

select l.location_name , sum(p.amount) as total_payments from location l , payment p where l.location_id = p.location_location_id

group by location_name having total_payments<5000000;

-- 19. Calculate Total Payments per Location

select l.location_name , sum(p.amount) as total_payments

from location l , payment p where l.location_id = p.location_location_id group by location_name;

-- 21. Retrieve couriers who have received payments totaling more than $1000 after a certain date (PaymentDate > 'YYYY-MM-DD').

select c.courier_id, c.sender_name, sum(p.amount) as payments_totalling

from courier c , location l , payment p where l.location_id = p.location_location_id AND p.payment_date > '2023-09-09'

AND c.courier_id = p.courier_courier_id group by courier_id HAVING sum(p.amount)>1000;

-- 22. Retrieve locations where the total amount received is more than $5000 before a certain date (PaymentDate > 'YYYY-MM-DD')

select c.courierID, c.sendernmae, sum(p.amount) as total_pay

from courier c , location l , payment p

where l.location_id = p.location_location_id AND p.payment_date > '2023-09-09'
```

```sql
AND c.courier_id = p.courier_courier_id group by courier_id HAVING sum(p.amount)>5000;

-- 24. Retrieve Payments with Location Information

select * from payment p join location l on p.location_location_id= l.locationID;

-- 25. Retrieve Payments with Courier and Location Information

select *  from payment p join courier c on p.courier_courier_id = c.courier_id join location l on p.location_location_id= l.location_id;

-- 26. List all payments with courier details

select * from payment p left join courier c on p.courier_courier_id = c.courier_id;

-- 27. Total payments received for each courier

select c.courier_id, sum(p.amount) as Total_Payments from payment p left join courier c on p.courier_courier_id = c.courier_id group by c.courier_id;

-- 28. List payments made on a specific date

select * from payment  where payment_date ='2023-09-09';

-- 29. Get Courier Information for Each Payment

select p.payment_id , c.courier_id , c.sender_address , c.receiver_name , c.weight , c.status , c.Tracking_number , c.delivery_date

from courier c join payment p on p.courier_courier_id = c.courier_id group by payment_id;

-- 30. Get Payment Details with Location

select p.payment_id, p.amount , p.payment_date ,l.location_name from

payment p left join location l on p.location_location_id= l.location_id;

-- 31. Calculating Total Payments for Each Courier

select c.courier_id, sum(p.amount) as Tot_Pay

from payment p left join courier c on p.courier_courier_id = c.courier_id group by c.courier_id;

-- 32. List Payments Within a Date Range

select payment_id , amount ,payment_date from payment where payment_date between '2023-02-12' AND '2024-09-09';

-- 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.user_id = c.user_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.service_id = c.courier_services_service_id;

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

```sql
select * from employee e left join payment p on e.employee_id =p.payment_id;

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

select * from user , courier;

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

select * from employee , location;

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

select courier_id , sender_name, sender_address from courier;

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

select courier_id , receiver_name ,receiver_address from courier;

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

select c.courier_id ,cs.service_id, cs.service_name from courier c left join courier_services cs on
cs.service_id=c.courier_services_service_id;

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

select e.employee_id , e.name , e.email ,e.contact_number, e.salary , c.courier_id , c.sender_address ,
c.receiver_name , c.weight , c.status , c.Tracking_number , c.delivery_date

from employee e left join location l on l.location_id = e. location_location_id  join payment p on l.location_id =
p.location_location_id join courier c on c.courier_id = p.courier_courier_id;

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

select l.location_id , l.location_name , sum(p.amount) as total_pay

from location l join payment p on l.location_id = p.location_location_id

group by l.location_id;

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

select courier_id ,sender_name , sender_address , receiver_name , receiver_address ,weight , status ,
Tracking_number

from courier where sender_name ='Jack';

-- 44. List all employees who share the same role.-- subquery

SELECT *  FROM employee  WHERE role IN (SELECT role FROM employee GROUP BY role HAVING
COUNT(*) > 1);

-- 45. Retrieve all payments made for couriers sent from the same location.

-- 47. List employees and the number of couriers they have delivered.

select e.employee_id,e.name ,count(c.status) as Num_of_couriers_Delivered

from employee e left join location l on l.location_id = e. location_location_id

join payment p on l.location_id = p.location_location_id join courier c on c.courier_id = p.courier_courier_id group
by c.status ;
```

```sql
-- 48. Find couriers that were paid an amount greater than the cost of their respective courier services

select c.courier_id , c.sender_address , c.receiver_name ,c.receiver_address, c.weight , c.status , c.Tracking_number , c.delivery_date , p.amount as payment , cs.cost as service_cost

from courier_services cs join courier c on cs.service_id = c.courier_services_service_id

join payment p ON p.courier_courier_id = c.courier_id where p.amount >cs.cost;

-- 49. Find couriers that have a weight greater than the average weight of all couriers

SELECT *FROM courier WHERE weight > (SELECT AVG(weight)FROM courier);

-- 50. 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);

-- 51. Find the total cost of all courier services where the cost is less than the maximum cost

SELECT SUM(cost) AS total_cost FROM courier_services WHERE cost < (SELECT MAX(cost)FROM courier_services);

-- 53. Find the locations where the maximum payment amount was made

SELECT location_location_id

FROM payment

WHERE amount = (SELECT MAX(amount) FROM payment);

  -- 54. 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 MAX(weight)FROM courier WHERE sender_name = 'Giri');
```