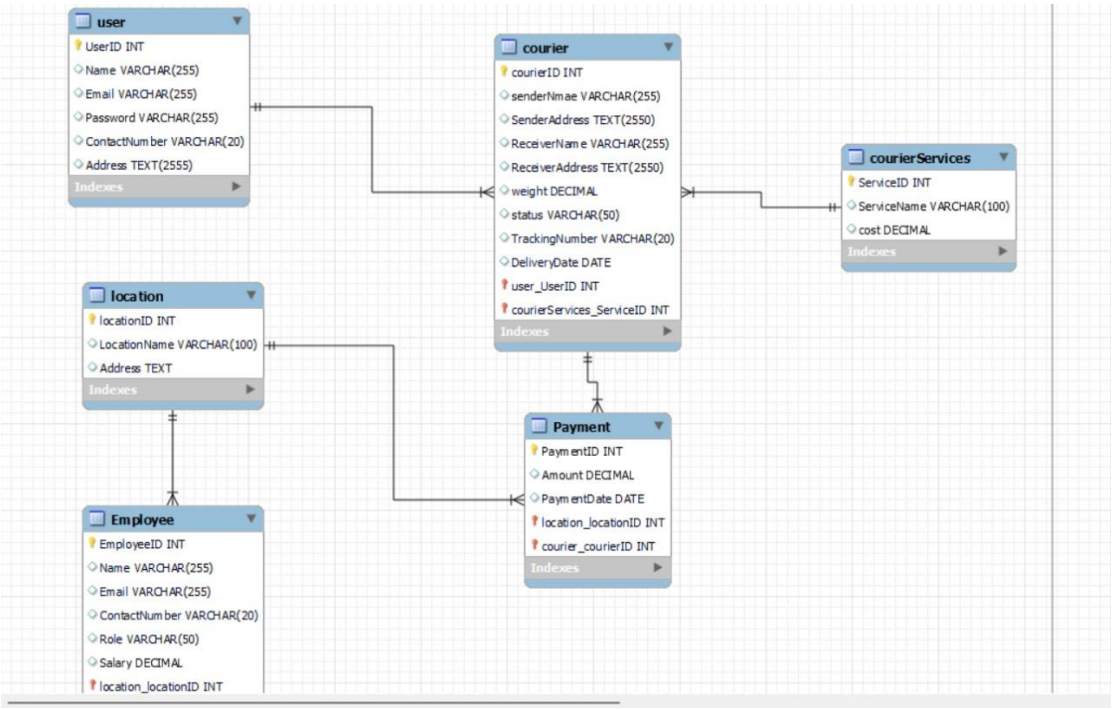
ASSIGNMENT 4

COURIER MANAGEMENT SYSTEM

ER DIAGRAM:



QUERIES:

#Courier system

create database Couriermanagement_db;

use Couriermanagement_db;

show tables;

show databases;

#insertions

drop table location;

create table user(user_id int primary key auto_increment,Name varchar(50),Email varchar(50),Password varchar(50),

ContactNumber varchar(50),Address varchar(50));

insert into user(Name,Email,Password,ContactNumber,Address)

values

('Lakshana', 'laksh@gmail.com', '12a8123' ,'5464546','delhi'),

('Kavitha','kavi@gmail.com' , 'tra3734' , '5464546' , 'chennai'),

('Bala','bala@gmail.com' , 'aio1474' , '5464546' , 'coimbatore'),

('Neha', 'neha@gmail.com', 'ioer243' ,'5464546','pune'),

('Aadhya' , 'aadhya@gmail.com','aer234','5464546','ooty');

create table courier(courier_id int primary key auto_increment,senderName varchar(50),

senderAddress varchar(50), ReceiverName varchar(50) ,

Receiveraddress varchar(50) ,weight double , status varchar(50) ,

```
trackingnumber varchar(50), DeliveryDate date);
insert into courier(senderName,  senderAddress, ReceiverName , Receiveraddress ,
weight , status , trackingnumber , DeliveryDate)
values
('zoya','kulu','rachel','chennai','10','transit', '124zx34', '2023-12-01'),
('noyal', 'tuticod','rihana','delhi','35','Delivered' ,'234256ty','2024-10-19'),
('niva','pune','reya','kovai','31','transit', '36io897', '2024-03-01'),
('vikranth','ooty','akim','mumbai','19','delivered', '123io897', '2024-02-29');
create table location( Location_id int primary key auto_increment,
LocationName varchar(50), address varchar(50));
insert into location ( LocationName , address)
values
('chennai','teenagar'),
('ooty','upper bazar'),
('delhi','muza seit');
create table payment(payment_id int primary key auto_increment,Amount double,
PaymentDate date,Location_id int,courier_id int);
insert into payment(Amount,PaymentDate,location_id,courier_id) values
(300000,'2024-01-13',1,2),
(1734000, '2023-12-13',2,4),
(50000,'2023-11-24',2,1),
(17000,'2023-03-12',3,4);
create table Employee(employee_id int primary key auto_increment,name varchar(50),
Email varchar(50), ContactNumber varchar(10),Role varchar(50),Salary double,
location_id int);
insert into Employee(name, Email, ContactNumber,Role,Salary,location_id)
values
('rayan','rayan@gmail.com','2342342', 'postman','60000',3),
('laksh','laksh@gmail.com','2342342', 'courierboy','50000',1),
('aadhira','aadhira@gmail.com','2342342', 'dataentry','450000',2),
('hazel','hazel@gmail.com','2342342', 'operator','80000',2);
create table courierServices(service_id int primary key auto_increment,
ServiceName varchar(50), cost double);
insert into courierServices( ServiceName,cost)
values
('medium',230),
('slow',220),
('fast',250) ,
('slow',220);
```

```
select * from courierServices;
select * from employee;
select * from payment;
select * from location;
#TASK 1:
-- 1. List all customers:
select* from user;
-- 2. List all orders for a specific customer:
-- 3. List all couriers:
select * from courier;
-- 4. List all packages for a specific order:
select * from courier
where courier_id=2;
-- 5. List all deliveries for a specific courier:
select * from courier
where status='Delivered';
-- 6.. List all undelivered packages:alter
select * from courier
where status!='Delivered';
-- 7. List all packages that are scheduled for delivery today:
select * from courier
where DeliveryDate=CURDATE();
-- 8. List all packages with a specific status:
select * from courier where status = 'delivered';
-- 11. List all packages with a specific weight range:
select * from courier
where weight between 10 and 20;
-- 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.senderName,  c.senderAddress, c.ReceiverName , c.Receiveraddress ,c.weight , c.status ,
c.trackingnumber , c.DeliveryDate
from courier c , payment p
where c.courier_id= p.courier_courier_id AND amount >50;

#Task 3: GroupBy, Aggregate Functions, Having, Order By, where
-- 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_id AND
l.location_id=p.location_id AND
l.location_id=e.location_id
group by e.name;
-- 15. Calculate the total revenue generated by each location
select LocationName ,sum(Amount) as total_revenue
from Location l, payment p
where l.location_id=p.location_location_id
group by LocationName;
-- 16. Find the total number of couriers delivered to each location.
select l.LocationName ,count(c.courier_id) as No_of_couriers
from location l , courier c , payment p
where c.courier_id =p.courier_courier_id AND
l.location_id =p.location_id
group by l.LocationName;
-- 17. Find the courier with the highest average delivery time:
select c.courier_id , avg(c.DeliveryDate-p.paymentDate) as average_time
from courier c , payment p
where c.courier_id=p.courier_id
group by c.courier_id
order by courier_id DESC
limit 0, 1;
-- 18. Find Locations with Total Payments Less Than a Certain Amount
select l.locationName , sum(p.amount) as total_payments
from location l , payment p
where l.locationid = p.location_id
group by LocationName
HAving total_payments>2000;
-- 19. Calculate Total Payments per Location
select l.locationName , sum(p.amount) as total_payments
from location l , payment p
where l.location_id = p.location_id
group by LocationName;
-- 20. Retrieve couriers who have received payments totaling more than $1000 in a specific
location (LocationID = X):
select c.courier_id, c.sendername, sum(p.amount) as total_pay
from  courier c , location l , payment p
where l.location_id = p.location_id AND l.location_id =1
AND  c.courier_id = p.courier_id
```

```
group by courier_id
HAVING sum(p.amount)>1000;
-- 21. Retrieve couriers who have received payments totaling more than $1000 after a certain
date (PaymentDate > 'YYYY-MM-DD'):
select c.courier_id, c.sendername, sum(p.amount) as total_pay
from  courier c , location l , payment p
where l.location_id = p.location_id AND p.paymentdate > '2023-01-01'
AND  c.courier_id = p.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.courier_id, c.sendername, sum(p.amount) as total_pay
from  courier c , location l , payment p
where l.location_id = p.location_id AND p.paymentdate > '2023-01-01'
AND  c.courier_id = p.courier_id
group by courier_id
HAVING sum(p.amount)>5000;

#TASK 4:
-- 23. Retrieve Payments with Courier Information
 select * from
 payment p left join courier c on p.courier_courier_id = c.courier_id;

-- 24. Retrieve Payments with Location Information
select * from
payment p join location l on p.location_location_id= l.location_id;

-- 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_Payment
from payment p left join courier c on  p.courier_courier_id = c.courier_id
group by c.courierid;
-- 28. List payments made on a specific date
```

```
select * from payment
where paymentdate ='2023-03-12';
-- 29. Get Courier Information for Each Payment
select p.payment_id , c.courier_id , c.senderAddress , c.Receivername , c.weight , c.status ,
c.trackingnumber , c.deliverydate
from courier c join payment p on  p.courier_courier_id = c.courier_id
group by paymentid;
-- 30. Get Payment Details with Location
select p.paymentid, p.amount , p.paymentdate ,l.locationname 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 Total_Payment
from payment p left join courier c on  p.courier_courier_id = c.courier_id
group by c.courierid;
-- 32. List Payments Within a Date Range
select paymentid , amount ,paymentdate from payment
where paymentdate between '2023-03-12' AND '2024-01-03';
-- 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.userid = 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 courierservices cs on cs.serviceid = c.courierServices_serviceid;
-- 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.employee_id =p.paymentid;
-- 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 , sendernmae, senderAddress
```

```
from courier;
-- 39. Retrieve a list of couriers and their corresponding receiver information (if available):
select courier_id , Receivername ,receiverAddress
from courier;
-- 40. Retrieve a list of couriers along with the courier service details (if available):
select c.courier_id ,cs.service_id, cs.servicename, cs.cost
from courier c left join courierservices cs on cs.service_id=c.courierservices_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.contactNUmber, e.salary,
c.courier_id , c.senderAddress , c.Receivername , c.weight , c.status , c.trackingnumber ,
c.deliverydate
from employee e left join location l on l.locationid = e. location_location_id
join payment p on l.location_id = p.location_location_id join courier c on c.courierid =
p.courier_courier_id;
-- 42. Retrieve a list of locations and the total payment amount received at each location:
select l.location_id , l.locationname , sum(p.amount) as total_payment
from location l join payment p on l.locationid = p.location_location_id
group by l.location_id;
-- 43. Retrieve all couriers sent by the same sender (based on SenderName).
select courier_id ,sendernmae , senderaddress , receivername , receiveraddress ,weight , status ,
trackingnumber
from courier
where sendernmae ='arun';
-- 44. List all employees who share the same role.-- subquery
select employeeid , name ,role , email , contactnumber , salary
from employee where role  in(
select role
from employee
group by role
having count(employeeid) > 1);

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

select p.payment_id , p.amount,p.paymentdate , l.locationname
from payment p join location l on p.location_location_id = l.location_id where location_id IN (
select l.location_id
from payment p join location l on p.location_location_id = l.location_id
group by l.location_id
having count(l.location_id)>1);
-- 46. Retrieve all couriers sent from the same location (based on SenderAddress).
```

```
select courier_id , senderAddress , Receivername ,Receiveraddress, weight , status ,
trackingnumber , deliverydate
from courier where senderaddress IN(
select senderaddress
from courier
group by senderaddress
having count(senderaddress)>1);
-- 47. List employees and the number of couriers they have delivered:
select e.employee_id , e.name , e.email ,count(c.status) as Number_of_couriers_Delivered
from employee e left join location l on l.locationid = 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 ;
-- 48. Find couriers that were paid an amount greater than the cost of their respective courier
services
select c.courier_id , c.senderAddress , c.Receivername ,c.Receiveraddress, c.weight , c.status ,
c.trackingnumber , c.deliverydate , p.amount as payment , cs.cost as service_cost
from courierservices cs join courier c  on cs.serviceid = c.courierservices_serviceid
join  payment p ON p.courier_courierid = c.courier_id
 where p.amount >cs.cost;

#TASK  4: Scope: Inner Queries, Non Equi Joins, Equi joins,Exist,Any,All
/*49. Find couriers that have a weight greater than the average weight of all couriers*/
select courier_id,weight
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 Salary
 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)
from courierServices
where cost<(select max(cost) from courierServices);
-- 52. Find all couriers that have been paid for
-- 53. Find the locations where the maximum payment amount was made
select Location_id
from  payment
where Amount in(select max(Amount) from payment);
```