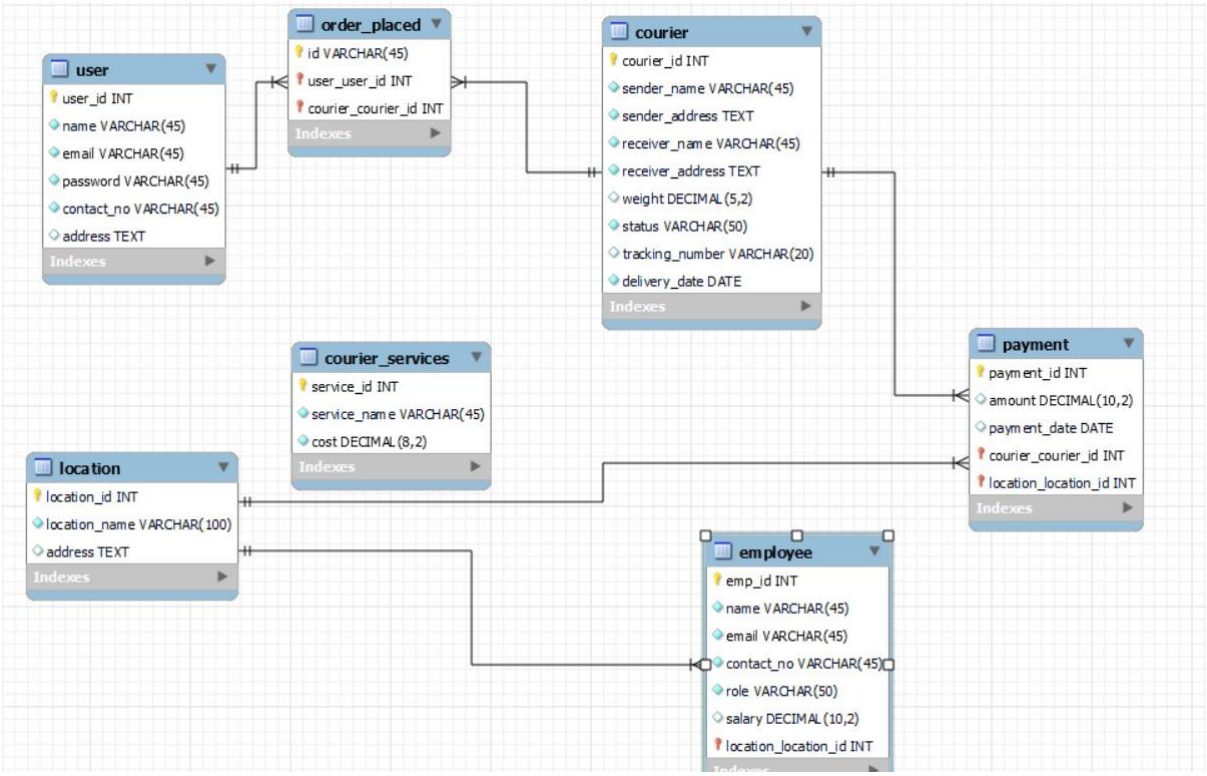
COURIER

ER DIAGRAM:



TASK 1

```
CREATE SCHEMA IF NOT EXISTS `courier` DEFAULT CHARACTER SET utf8 ;
USE `courier` ;
-- Table `courier`.`user`
CREATE TABLE IF NOT EXISTS `courier`.`user` (
 `user_id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NOT NULL,
  `email` VARCHAR(45) NOT NULL,
  `password` VARCHAR(45) NOT NULL,
  `contact_no` VARCHAR(45) NOT NULL,
```

```
`address` TEXT NULL,
PRIMARY KEY (`user_id`))
ENGINE = InnoDB;
-- Table `courier`.`courier`
CREATE TABLE IF NOT EXISTS `courier`.`courier` (
`courier_id` INT NOT NULL AUTO_INCREMENT,
`sender_name` VARCHAR(45) NOT NULL,
`sender_address` TEXT NOT NULL,
`receiver_name` VARCHAR(45) NOT NULL,
`receiver_address` TEXT NOT NULL,
`weight` DECIMAL(5,2) NULL,
`status` VARCHAR(50) NOT NULL,
`tracking_number` VARCHAR(20) NULL,
`delivery_date` DATE NOT NULL,
PRIMARY KEY (`courier_id`),
UNIQUE INDEX `tracking_number_UNIQUE` (`tracking_number` ASC) VISIBLE)
ENGINE = InnoDB;
-- Table `courier`.`location`
CREATE TABLE IF NOT EXISTS `courier`.`location` (
`location_id` INT NOT NULL AUTO_INCREMENT,
`location_name` VARCHAR(100) NOT NULL,
`address` TEXT NULL,
 PRIMARY KEY (`location_id`))
 ENGINE = InnoDB;
 -- Table `courier`.`payment`
 CREATE TABLE IF NOT EXISTS `courier`.`payment` (
`payment_id` INT NOT NULL AUTO_INCREMENT,
`amount` DECIMAL(10,2) NULL,
`payment_date` DATE NULL,
```

```
`courier_courier_id` INT NOT NULL,
`location_location_id` INT NOT NULL,
PRIMARY KEY (`payment_id`, `courier_courier_id`, `location_location_id`),
INDEX `fk_payment_courier1_idx` (`courier_courier_id` ASC) VISIBLE,
INDEX `fk_payment_location1_idx` (`location_location_id` ASC) VISIBLE,
CONSTRAINT `fk_payment_courier1`
FOREIGN KEY (`courier_courier_id`)
REFERENCES `courier`.`courier` (`courier_id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_payment_location1`
FOREIGN KEY (`location_location_id`)
REFERENCES `courier`.`location` (`location_id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
-- Table `courier`.`employee`
CREATE TABLE IF NOT EXISTS `courier`.`employee` (
`emp_id` INT NOT NULL AUTO_INCREMENT,
`name` VARCHAR(45) NOT NULL,
`email` VARCHAR(45) NOT NULL,
`contact_no` VARCHAR(45) NOT NULL,
`role` VARCHAR(50) NOT NULL,
`salary` DECIMAL(10,2) NULL,
`location_location_id` INT NOT NULL,
PRIMARY KEY (`emp_id`, `location_location_id`),
UNIQUE INDEX `email_UNIQUE` (`email` ASC) ,
INDEX `fk_employee_location_idx` (`location_location_id` ASC) ,
CONSTRAINT `fk_employee_location`
```

```
  FOREIGN KEY (`location_location_id`)
  REFERENCES `courier`.`location` (`location_id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
 ENGINE = InnoDB;
-- Table `courier`.`courier_services`
CREATE TABLE IF NOT EXISTS `courier`.`courier_services` (
`service_id` INT NOT NULL AUTO_INCREMENT,
`service_name` VARCHAR(45) NOT NULL,
`cost` DECIMAL(8,2) NOT NULL,
 PRIMARY KEY (`service_id`))
 ENGINE = InnoDB;
 -- Table `courier`.`order_placed`
CREATE TABLE IF NOT EXISTS `courier`.`order_placed` (
`id` VARCHAR(45) NOT NULL,
`user_user_id` INT NOT NULL,
 `courier_courier_id` INT NOT NULL,
 PRIMARY KEY (`id`, `user_user_id`, `courier_courier_id`),
 INDEX `fk_user_has_courier_courier1_idx` (`courier_courier_id` ASC),
 INDEX `fk_user_has_courier_user1_idx` (`user_user_id` ASC),
 CONSTRAINT `fk_user_has_courier_user1`
 FOREIGN KEY (`user_user_id`)
 REFERENCES `courier`.`user` (`user_id`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION,
 CONSTRAINT `fk_user_has_courier_courier1`
  FOREIGN KEY (`courier_courier_id`)
  REFERENCES `courier`.`courier` (`courier_id`)
  ON DELETE NO ACTION
```

```
ON UPDATE NO ACTION)
ENGINE = InnoDB;

use courier;
show tables;
desc user;
insert into user(name,email,password,contact_no,address)
values('harry','harry@gmail.com','h@123','8754012458','mumbai'),
('potter','potter@gmail.com','p#123','7512648923','chennai'),
('weasley','wea@gmail.com','w$345','9874521458','pune'),
('hermoine','hermo@gmail.com','h@#$','7845968125','mumbai');
desc location;
insert into location(location_name,address) values('navi','mumbai'),
('vadachennai','chennai'),
('siruseri','chennai'),
('belandur','bangalore'),
('showdown','pune');
select * from location;
desc courier_services;
insert into courier_services(service_name,cost) values ('Express Service',1800),
('Overnight Services',2300),('Standard Service',1000),('Parcel Services','1500');
select * from courier_services;
desc employee;
insert into employee(name,email,contact_no,role,salary,location_location_id)
values('ram','ram@gmail.com','7584692145','manager',45000,2),
('ramu','ramu@gmail.com','8457961256','postman',25000,1),
('raju','rj@gmail.com','8754123698','deputy manager',50000,3),
('vani','vani@gmail.com','7548123695','cover man',23000,1),
('bani','bani@gmail.com','7538123695','cover man',23000,4);
select * from employee;
```

```
create table courier(courier_id int primary key not null auto_increment,sender_name varchar(255) not null,

sender_address text,receiver_name varchar(255),receiver_address text,weight decimal(5,2),

status varchar(50),tracking_number varchar(20),delivery_date date);

insert into courier(sender_name,sender_address,receiver_name,receiver_address,weight,status,tracking_number,delivery_date) values

('harry','chennai','janu','bangalore',450,'on board','145','2024-03-01'),

('hermoine','bangalore','banu','pune',350,'on door','421','2024-02-22'),

('harry','mumbai','jayesh','pune',150,'on the way','545','2024-04-01'),

('weasley','pune','janu','chennai',500,'on board','135','2024-03-11');

insert into courier(sender_name,sender_address,receiver_name,receiver_address,weight,status,tracking_number,delivery_date) values

('potter','chennai','janu','bangalore',450,'delivered','195','2024-04-30');

delete from courier where courier_id =5;

desc order_placed;

insert into order_placed values(1,1,1),(2,2,2),(3,4,3),(4,3,4);

select * from order_placed;

desc payment;

drop database courier;

insert into payment(amount,payment_date,courier_courier_id,location_location_id) values(1500,'2024-02-22',2,1),

(2300,'2024-04-05',3,2),

(1800,'2024-03-03',4,4),

(1500,'2024-03-07',1,3),

(1800,'2024-05-03',4,1);

select * from payment;
```

TASK - 2

#1.list all customers

```
select * from user;
```

#2.list all orders for a specific customers

```
select * from courier where sender_name='harry';
```

#3.list all couriers

```
select * from courier;
```

#4.list all packages for a specific order

```
select * from courier_services order by cost desc;
```

#5.list all deliveries for a specific courier

```
select * from courier where courier_id=2;
```

#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=curdate();
```

#8.list all packages with specific status

```
select courier_id,status from courier;
```

#9.calculate the total number of packages for each courier

```
select courier_id,status,count(courier_id) from courier
group by status order by courier_id ;
```

#10.find the average delivery time for each courier

```
select courier_id,delivery_date as avg_delivery_date from courier;
```

#11.list all packages with a specific weight range

```
select * from courier where weight between 100 and 400;
```

#12.retrieve employees whose names contain 'john'

```
select * from employee where name like '%john%';
```

#13.retrieve all courier records with payments greater than 2000

```
select * from courier c,payment p
where c.courier_id=p.courier_courier_id and amount>2000;
```

TASK - 3

#14.find the total number of couriers handles by each employee

```
select e.name,count(c.courier_id)
from courier c,payment p,location l,employee e
where c.courier_id=p.courier_courier_id and p.location_location_id=l.location_id and
l.location_id=e.location_location_id
group by e.name;
```

#15.calculat the total revenue generated by each location

```
select l.location_name,sum(p.amount),p.location_location_id
from location l join payment p
on l.location_id=p.location_location_id
group by p.location_location_id;
```

#16.find the total number of couriers deliveres to each loctaion

```
select receiver_address,count(receiver_address)
from courier
group by receiver_address;
```

#17.find the courier with the highest average delivery time

```
select courier_id,sender_name,delivery_date
from courier
order by delivery_date desc;
```

#18.find locations with total payments less than a certain amount

```
select l.address,sum(p.amount) as tot_amt
from location l join payment p
on p.location_location_id=l.location_id
group by p.location_location_id
having tot_amt<2000;
```

#19.calculate total payments per location

```
select l.address,sum(p.amount) as tot_amt
from location l join payment p
on p.location_location_id=l.location_id
group by p.location_location_id;
```

#20.retrieve couriers who have received payments totaling more than $1000 in a specific location(LocationId=X)

```
select c.courier_id,sum(p.amount) as total,l.address
from courier c join payment p
on c.courier_id=p.courier_courier_id join location l on p.location_location_id=l.location_id
group by l.address
having total>1000 and l.address='chennai' ;
```

#21.retrieve couriers who have received payments totaling more than $1000 after a certain date

```
select c.courier_id,sum(p.amount) as total,l.address
from courier c join payment p
on c.courier_id=p.courier_courier_id join location l on p.location_location_id=l.location_id
where payment_date>'2024-04-01'
group by l.address
having total>1000 ;
```

#22.retrieve location where the total amount received is more than $5000 before a certain date

```
select c.courier_id,sum(p.amount) as total,l.address
from courier c join payment p
on c.courier_id=p.courier_courier_id join location l on p.location_location_id=l.location_id
where payment_date<'2024-04-01'
group by l.address
having total>1000 ;
```

TASK - 4

#1.weight greater than the average weight of all couriers

```
select courier_id from courier where weight>(select avg(weight) from courier);
select avg(weight) from courier;
```

#2.names of employee who have salary greater than avg salary

```
select name,salary from employee where salary>(select avg(salary) from employee);
```

#3.weight greater than the average weight of all couriers sent by specific sender

```
select courier_id from courier where weight>(select avg(weight) from courier where
sender_name='harry');
```

#4.location where the highest payment made

```
select location_location_id from payment where amount=
(select max(amount) from payment);
```