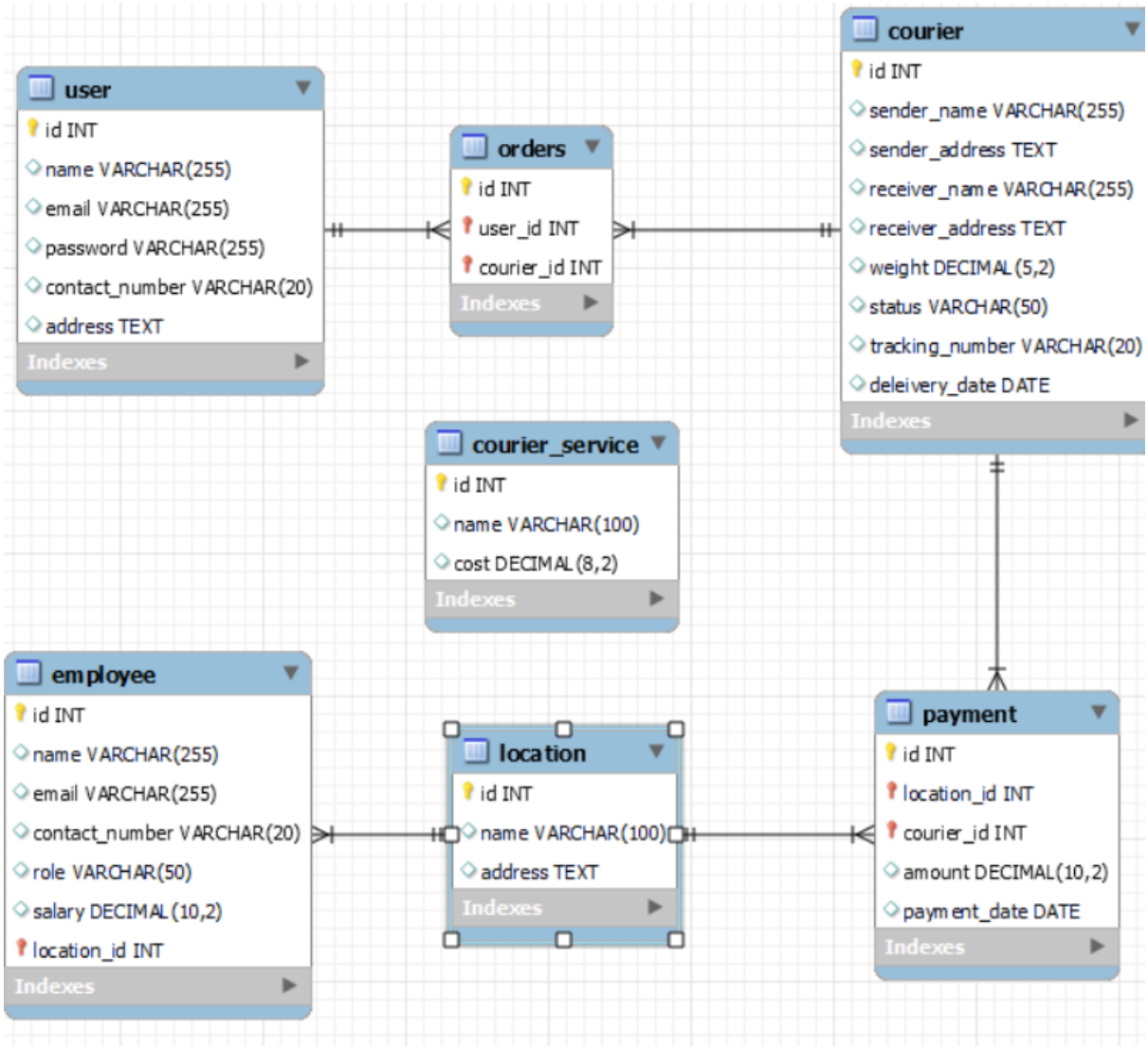
ASSIGNMENT NO : 4

Courier Management System

ER Diagram:



Task:1. Database Design:

-- MySQL Workbench Forward Engineering

-- ---

-- Schema couriersystem

-- ---

```
-- -----------------------------------------------------
-- Schema couriersystem
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `couriersystem` DEFAULT CHARACTER SET utf8 ;
USE `couriersystem` ;

-- -----------------------------------------------------
-- Table `couriersystem`.`user`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `couriersystem`.`user` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(255) NULL,
  `email` VARCHAR(255) NULL,
  `password` VARCHAR(255) NULL,
  `contact_number` VARCHAR(20) NULL,
  `address` TEXT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `email_UNIQUE` (`email` ASC) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `couriersystem`.`courier`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `couriersystem`.`courier` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `sender_name` VARCHAR(255) NULL,
  `sender_address` TEXT NULL,
  `receiver_name` VARCHAR(255) NULL,
  `receiver_address` TEXT NULL,
```

```
  `weight` DECIMAL(5,2) NULL,
  `status` VARCHAR(50) NULL,
  `tracking_number` VARCHAR(20) NULL,
  `deleivery_date` DATE NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `tracking_number_UNIQUE` (`tracking_number` ASC) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `couriersystem`.`courier_service`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `couriersystem`.`courier_service` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(100) NULL,
  `cost` DECIMAL(8,2) NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `couriersystem`.`location`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `couriersystem`.`location` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(100) NULL,
  `address` TEXT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;
-- -----------------------------------------------------
```

```
-- Table `couriersystem`.`employee`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `couriersystem`.`employee` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(255) NULL,
  `email` VARCHAR(255) NULL,
  `contact_number` VARCHAR(20) NULL,
  `role` VARCHAR(50) NULL,
  `salary` DECIMAL(10,2) NULL,
  `location_id` INT NOT NULL,
  PRIMARY KEY (`id`, `location_id`),
  UNIQUE INDEX `email_UNIQUE` (`email` ASC) ,
  INDEX `fk_employee_location1_idx` (`location_id` ASC) ,
  CONSTRAINT `fk_employee_location1`
    FOREIGN KEY (`location_id`)
    REFERENCES `couriersystem`.`location` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `couriersystem`.`orders`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `couriersystem`.`orders` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `user_id` INT NOT NULL,
  `courier_id` INT NOT NULL,
  PRIMARY KEY (`id`, `user_id`, `courier_id`),
  INDEX `fk_user_has_courier_courier1_idx` (`courier_id` ASC) ,
  INDEX `fk_user_has_courier_user_idx` (`user_id` ASC) ,
  CONSTRAINT `fk_user_has_courier_user`
```

```
    FOREIGN KEY (`user_id`)
    REFERENCES `couriersystem`.`user` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_user_has_courier_courier1`
    FOREIGN KEY (`courier_id`)
    REFERENCES `couriersystem`.`courier` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `couriersystem`.`payment`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `couriersystem`.`payment` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `location_id` INT NOT NULL,
  `courier_id` INT NOT NULL,
  `amount` DECIMAL(10,2) NULL,
  `payment_date` DATE NULL,
  PRIMARY KEY (`id`, `location_id`, `courier_id`),
  INDEX `fk_location_has_courier_courier1_idx` (`courier_id` ASC) ,
  INDEX `fk_location_has_courier_location1_idx` (`location_id` ASC) ,
  CONSTRAINT `fk_location_has_courier_location1`
    FOREIGN KEY (`location_id`)
    REFERENCES `couriersystem`.`location` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_location_has_courier_courier1`
    FOREIGN KEY (`courier_id`)
```

```
    REFERENCES `couriersystem`.`courier` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
use couriersystem;
```

INSERTION:

-- user insertion

```
insert into user(name,email,password,contact_number,address)values
('Ram Prasad','ram@gmail.com','ram#123','9024554745','123 main street'),
('Sandiya Vishwanath','sandiya@gmail.com','sandiya@456','9174543526','543 anna nagar'),
('Jayanthi Selvam','selvam@gmail.com','jay@123','9082707895','321 Ranipet'),
('Swetha Seetharaman','swetha@gmail.com','swetha_256','7098645321','456 White Town'),
('Divya Dharshini','divya@gmail.com','divya@345','9123765480','890 Madagadipet'),
('Nisha Vaithiyanathan','nisha@gmail.com','nisha#789','9865432178','698 Sowkarpet'),
('Darshini Balamurali','darshnini@gmail.com','darshini@123','709834521','987 nehru nagar'),
('Agalya Shanmugam','agalya@gmail.com','agalya#678','8143256790','678 Gandhi park'),
('Harini Murugavel','harini@gmail.com','harini@123','9024554745','234 Semmandalam'),
('Selva Ramaiah','selva@gmail.com','selva#908','9156473420','567 Manjakupam');
```

```
mysql> select * from user;
+----+---------------------+--------------------+-------------+----------------+-----------------+
| id | name                | email              | password    | contact_number | address         |
+----+---------------------+--------------------+-------------+----------------+-----------------+
|  1 | Ram Prasad          | ram@gmail.com      | ram#123     | 9024554745     | 123 main street |
|  2 | Sandiya Vishwanath  | sandiya@gmail.com  | sandiya@456 | 9174543526     | 543 anna nagar  |
|  3 | Jayanthi Selvam     | selvam@gmail.com   | jay@123     | 9082707895     | 321 Ranipet     |
|  4 | Swetha Seetharaman  | swetha@gmail.com   | swetha_256  | 7098645321     | 456 White Town  |
|  5 | Divya Dharshini     | divya@gmail.com    | divya@345   | 9123765480     | 890 Madagadipet |
|  6 | Nisha Vaithiyanathan| nisha@gmail.com    | nisha#789   | 9865432178     | 698 Sowkarpet   |
|  7 | Darshini Balamurali | darshnini@gmail.com| darshini@123| 709834521      | 987 nehru nagar |
|  8 | Agalya Shanmugam    | agalya@gmail.com   | agalya#678  | 8143256790     | 678 Gandhi park |
|  9 | Harini Murugavel    | harini@gmail.com   | harini@123  | 9024554745     | 234 Semmandalam |
| 10 | Selva Ramaiah       | selva@gmail.com    | selva#908   | 9156473420     | 567 Manjakupam  |
+----+---------------------+--------------------+-------------+----------------+-----------------+
```

-- courier insertion

insert into courier(sender_name, sender_address, receiver_name,

receiver_address, weight, status, tracking_number, deleivery_date)values

('Ram Prasad','123 main street','Swetha Seetharaman','456 White Town',2.5,'in transit','ABC123','2024-04-05'),

('Sandiya Vishwanath','543 anna nagar','Divya Dharshini','890 Madagadipet',3.0,'delivered','DEF456','2024-02-08'),

('Jayanthi Selvam','321 Ranipet','Darshini Balamurali','987 nehru nagar',1.8,'in transit','GHI789','2024-03-10'),

('Ram Prasad','123 main street','Agalya Shanmugam','678 Gandhi park',2.2,'delivered','JKL012','2024-03-15'),

('Divya Dharshini','890 Madagadipet','Nisha Vaithiyanathan','698 Sowkarpet',4.5, 'in transit', 'MNO345', '2024-04-20'),

('Nisha Vaithiyanathan','698 Sowkarpet','Harini Murugavel','234 Semmandalam',3.8, 'in transit', 'PQR678', '2024-05-25'),

('Darshini Balamurali','987 nehru nagar','Selva Ramaiah','567 Manjakupam',2.0, 'delivered', 'STU901', '2024-02-02'),

('Agalya Shanmugam','678 Gandhi park','Ram Prasad','123 main street',1.5, 'delivered', 'VWX234', '2024-02-25'),

('Harini Murugavel','234 Semmandalam','Jayanthi Selvam','321 Ranipet', 3.2, 'in transit', 'YZA567', '2024-03-08'),

('Selva Ramaiah','567 Manjakupam','Swetha Seetharaman','456 White Town',2.8, 'delivered', 'BCD890', '2024-01-18');

```
mysql> select * from courier;
+----+---------------------+----------------+---------------------+------------------+--------+------------+-----------------+----------------+
| id | sender_name         | sender_address | receiver_name       | receiver_address | weight | status     | tracking_number | deleivery_date |
+----+---------------------+----------------+---------------------+------------------+--------+------------+-----------------+----------------+
|  1 | Ram Prasad          | 123 main street | Swetha Seetharaman  | 456 White Town   |   2.50 | in transit | ABC123          | 2024-04-05     |
|  2 | Sandiya Vishwanath  | 543 anna nagar  | Divya Dharshini     | 890 Madagadipet  |   3.00 | delivered  | DEF456          | 2024-02-08     |
|  3 | Jayanthi Selvam     | 321 Ranipet     | Darshini Balamurali | 987 nehru nagar  |   1.80 | in transit | GHI789          | 2024-03-10     |
|  4 | Ram Prasad          | 123 main street | Agalya Shanmugam    | 678 Gandhi park  |   2.20 | delivered  | JKL012          | 2024-03-15     |
|  5 | Divya Dharshini     | 890 Madagadipet | Nisha Vaithiyanathan | 698 Sowkarpet   |   4.50 | in transit | MNO345          | 2024-04-20     |
|  6 | Nisha Vaithiyanathan | 698 Sowkarpet  | Harini Murugavel    | 234 Semmandalam  |   3.80 | in transit | PQR678          | 2024-05-25     |
|  7 | Darshini Balamurali | 987 nehru nagar | Selva Ramaiah       | 567 Manjakupam   |   2.00 | delivered  | STU901          | 2024-02-02     |
|  8 | Agalya Shanmugam    | 678 Gandhi park | Ram Prasad          | 123 main street  |   1.50 | delivered  | VWX234          | 2024-02-25     |
|  9 | Harini Murugavel    | 234 Semmandalam | Jayanthi Selvam     | 321 Ranipet      |   3.20 | in transit | YZA567          | 2024-03-08     |
| 10 | Selva Ramaiah       | 567 Manjakupam  | Swetha Seetharaman  | 456 White Town   |   2.80 | delivered  | BCD890          | 2024-01-18     |
+----+---------------------+----------------+---------------------+------------------+--------+------------+-----------------+----------------+
```

-- courier_services

insert into courier_service(name,cost)values

('kpn',1750),

('rnp',1000),

('vijay',2000),

('rrr',1500),

('krk',900);

```
mysql> select*from courier_service;
+----+-------+---------+
| id | name  | cost    |
+----+-------+---------+
|  1 | kpn   | 1750.00 |
|  2 | rnp   | 1000.00 |
|  3 | vijay | 2000.00 |
|  4 | rrr   | 1500.00 |
|  5 | krk   |  900.00 |
+----+-------+---------+
```

-- orders

insert into orders(user_id,courier_id)values

(1,1),

(2,2),

(3,3),

(1,4),

(5,5),

(6,6),

(7,7),

(8,8),

(9,9),

(10,10);

insert into orders(user_id,courier_id)values

(4,1),

(10,5);

```
mysql> select*from orders;
+----+---------+------------+
| id | user_id | courier_id |
+----+---------+------------+
|  1 |       1 |          1 |
|  2 |       2 |          2 |
|  3 |       3 |          3 |
|  4 |       1 |          4 |
|  5 |       5 |          5 |
|  6 |       6 |          6 |
|  7 |       7 |          7 |
|  8 |       8 |          8 |
|  9 |       9 |          9 |
| 10 |      10 |         10 |
+----+---------+------------+
```

-- location table

insert into location(name,address)values

('chennai', ' main street'),

('bangalore', 'anna nagar'),

('chennai', 'Ranipet'),

('pondicherry', ' White Town'),

('pondicherry', ' Madagadipet'),

('chennai', 'Sowkarpet'),

('mumbai', ' nehru nagar'),

('coimbatore', ' Gandhi park'),

('mumbai', 'Semmandalam'),

('villupuram', ' Manjakupam');

```
mysql> select*from location;
+----+-------------+--------------+
| id | name        | address      |
+----+-------------+--------------+
|  1 | chennai     |  main street |
|  2 | bangalore   | anna nagar   |
|  3 | chennai     | Ranipet      |
|  4 | pondicherry |  White Town  |
|  5 | pondicherry |  Madagadipet |
|  6 | chennai     | Sowkarpet    |
|  7 | mumbai      |  nehru nagar |
|  8 | coimbatore  |  Gandhi park |
|  9 | mumbai      | Semmandalam  |
| 10 | villupuram  |  Manjakupam  |
+----+-------------+--------------+
```

-- employee table

insert into employee(name,email,contact_number,role,salary,location_id)values

('vedha','vedha@gmail.com','7145454545','manager',60000,1),

('suruthy','suruthy@gmail.com','8045454545','dispatcher',40000,2),

('vaithi','vaithi@gmail.com','9145454545','courier support',10000,3),

('nikitha','nikitha@gmail.com','9045454545','packing',5000,2),

('yogesh','yogesh@gmail.com','8045454545','manager',65000,5),

('kiruba','kiruba@gmail.com','7945454001','packing',10000,6),

('devi','devi@gmail.com','9145454002','dispatcher',45000,8),

('pandi','pandi@gmail.com','7345454004','courier support',15000,7),

('aravindh','aravindh@gmail.com','7145454000','manager',70000,6),

('john','john@gmail.com','7145454003','dispatcher',45000,10);

```
mysql> select*from employee;
+----+----------+--------------------+----------------+-----------------+----------+-------------+
| id | name     | email              | contact_number | role            | salary   | location_id |
+----+----------+--------------------+----------------+-----------------+----------+-------------+
|  1 | vedha    | vedha@gmail.com    | 7145454545     | manager         | 60000.00 |           1 |
|  2 | suruthy  | suruthy@gmail.com  | 8045454545     | dispatcher      | 40000.00 |           2 |
|  3 | vaithi   | vaithi@gmail.com   | 9145454545     | courier support | 10000.00 |           3 |
|  4 | nikitha  | nikitha@gmail.com  | 9045454545     | packing         |  5000.00 |           2 |
|  5 | yogesh   | yogesh@gmail.com   | 8045454545     | manager         | 65000.00 |           5 |
|  6 | kiruba   | kiruba@gmail.com   | 7945454001     | packing         | 10000.00 |           6 |
|  7 | devi     | devi@gmail.com     | 9145454002     | dispatcher      | 45000.00 |           8 |
|  8 | pandi    | pandi@gmail.com    | 7345454004     | courier support | 15000.00 |           7 |
|  9 | aravindh | aravindh@gmail.com | 7145454000     | manager         | 70000.00 |           6 |
| 10 | john     | john@gmail.com     | 7145454003     | dispatcher      | 45000.00 |          10 |
+----+----------+--------------------+----------------+-----------------+----------+-------------+
```

-- payment table

insert into payment(courier_id,location_id,amount,payment_date)values

(1,4,400,'2024-04-03'),

(2,5,500,'2024-02-05'),

(3,7,600,'2024-03-06'),

(4,8,750,'2024-03-14'),

(5,6,400,'2024-04-18'),

(6,9,300,'2024-05-22'),

(7,10,500,'2024-01-31'),

(8,1,600,'2024-02-21'),

(9,3,550,'2024-03-06'),

(10,4,400,'2024-01-17');

```
mysql> select*from payment;
+----+-------------+------------+--------+--------------+
| id | location_id | courier_id | amount | payment_date |
+----+-------------+------------+--------+--------------+
|  1 |           4 |          1 | 400.00 | 2024-04-03   |
|  2 |           5 |          2 | 500.00 | 2024-02-05   |
|  3 |           7 |          3 | 600.00 | 2024-03-06   |
|  4 |           8 |          4 | 750.00 | 2024-03-14   |
|  5 |           6 |          5 | 400.00 | 2024-04-18   |
|  6 |           9 |          6 | 300.00 | 2024-05-22   |
|  7 |          10 |          7 | 500.00 | 2024-01-31   |
|  8 |           1 |          8 | 600.00 | 2024-02-21   |
|  9 |           3 |          9 | 550.00 | 2024-03-06   |
| 10 |           4 |         10 | 400.00 | 2024-01-17   |
+----+-------------+------------+--------+--------------+
```

-- Task 2: Select,Where

-- 1. List all customers:

```
select* from user;
```

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

```
select *
from courier
where sender_name = 'ram prasad';
```

-- 3. List all couriers:

```
select * from courier;
```

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

```
select c.*
from courier c,orders o
where c.id=o.courier_id and o.id=5;
```

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

```
select id,user_id
from orders
where courier_id=1;
```

-- 6. List all undelivered packages:

```
select sender_name,receiver_name,status
from courier
where status !='delivered';
```

-- 7. List all packages that are scheduled for delivery today:

```
select *
from courier
where deleivery_date=curdate();
```

-- 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 c.id,count(c.id)as total_packages
from courier c,orders o
where c.id=o.courier_id
group by courier_id;
```

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

```
select sender_name,avg(datediff(deleivery_date,current_date()))as avg_time
from courier
group by sender_name;
```

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

```
select *
from courier
where weight between 1.80 and 4.50;
```

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

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

-- 13. Retrieve all courier records with payments greater than $500.

```
select c.*
from courier c,payment p
where c.id=p.courier_id and amount>500;
```

-- Task 3: GroupBy, Aggregate Functions, Having, Order By, where

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

```
ALTER TABLE courier
ADD COLUMN employee_id INT,
ADD CONSTRAINT fk_employee_courier
FOREIGN KEY (employee_id) REFERENCES employee(id);

update courier
set employee_id=case
when id=1 then 1
when id=2 then 1
when id=3 then 2
when id=4 then 3
when id=5 then 4
when id=6 then 6
when id=7 then 5
when id=8 then 7
when id=9 then 3
when id=10 then 7
end;

select e.name,count(c.employee_id) as employee
```

```
from employee e,courier c
where e.id=c.employee_id
group by e.name;
```

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

```
select l.*,sum(p.amount)as revenue
from location l,payment p
where l.id=p.location_id
group by l.id;
```

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

```
select receiver_address,count(*) as delivered
from courier
where status='delivered'
group by receiver_address;
```

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

```
select courier.*,avg(datediff(deleivery_date,current_date()))as avg_time
from courier
group by sender_name
order by avg_time desc
limit 1;
```

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

```
select l.*
from payment p,location l
where l.id=p.location_id and amount<600
group by l.id;
```

-- 19. Calculate Total Payments per Location

```
select l.*,count(p.amount)as payment
from location l,payment p
where 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.*
from courier c,location l,payment p
where c.id=p.courier_id and l.id=p.location_id and p.amount>1000 and l.name='mumbai';
```

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

```
select c.*
from courier c,payment p
where c.id=p.courier_id and amount>1000 and p.payment_date>'2024-04-03';
```

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

```
select l.*, SUM(p.amount) AS total_payment
from location l,payment p
where l.id=p.location_id and  p.payment_date < '2024-04-03'
group by l.name
having total_payment > 5000;
```

-- Task 4: Inner Join,Full Outer Join, Cross Join, Left Outer Join,Right Outer Join

-- 23. Retrieve Payments with Courier Information

```
select p.amount, c.*
from payment p
join courier c on p.courier_id = c.id;
```

-- 24. Retrieve Payments with Location Information

```
select p.amount, l.*
from payment p
join location l on p.location_id = l.id;
```

-- 25. Retrieve Payments with Courier and Location Information

```
select p.amount, c.*, l.*
```

```
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 p.amount, c.*
from payment p
left join courier c on p.courier_id = c.id;
```

-- 27. Total payments received for each courier

```
select c.*, SUM(p.amount) AS total_payment
from courier c
join payment p ON c.id = p.courier_id
group by c.id;
```

-- 28. List payments made on a specific date

```
select *
from payment
where payment_date = '2024-03-06';
```

-- 29. Get Courier Information for Each Payment

```
select p.amount,c.*
from payment p
left join courier c ON p.courier_id = c.id;
```

-- 30. Get Payment Details with Location

```
select p.*, l.*
from payment p
left join location l on p.location_id = l.id;
```

-- 31. Calculating Total Payments for Each Courier

```
select c.*, SUM(p.amount) AS total_payment
from courier c
left join payment p ON c.id = p.courier_id
group by c.id;
```

-- 32. List Payments Within a Date Range

```
select p.id,p.amount
from payment p
where payment_date between '2024-03-05' and '2024-03-14';
```

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

```
select u.name,u.email,c.id,c.*
from user u left join orders o on o.user_id=u.id left join courier c on c.id=o.courier_id;
```

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

```
select c.id as courier_id,c.sender_name,c.sender_address,c.receiver_name,c.receiver_address,
cs.id as service_id,cs.name as service_name, cs.cost
from courier c left join courier_service cs on c.id=cs.id;
```

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

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

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

```
select u.*, cs.*
from user u
cross join courier_service cs;
```

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

```
select e.*,l.*
from employee e
cross join location l ;
```

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

```
select c.sender_name,c.sender_address,c.receiver_name,c.receiver_address,c.status,u.*
from user u left join orders o on u.id=o.user_id left join courier c on c.id=o.courier_id;
```

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

```
select c.receiver_name,receiver_address
from courier c;
```

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

```
select c.*,cs.*
from courier c join courier_service cs on c.id=cs.id;
```

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

```
select e.*, COUNT(c.id) AS num_couriers
from employee e left join courier c ON e.id = c.employee_id
group by e.id;
```

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

```
select l.*, SUM(p.amount) AS total_payment
from location l
left 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 c1.*
from courier c1 join courier c2 on c1.sender_address = c2.sender_address and c1.id <> c2.id;
```

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

```
select e1.*
from employee e1 join employee e2 on e1.role=e2.role and e1.id<>e2.id;
```

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

```
select p1.*
from payment p1 join payment p2 ON p1.location_id = p2.location_id AND p1.courier_id <>
p2.courier_id;
```

-- 46. Retrieve all couriers sent from the same location (based on SenderAddress).

```
select c1.*
from courier c1 join courier c2 on c1.sender_address = c2.sender_address AND c1.id <>
c2.id;
```

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

```
select e.id, e.name, COUNT(c.id) AS num_couriers_delivered
from employee e left join courier c on e.id = c.employee_id and c.status = 'delivered'
group by e.id;
```

/* 48. Find couriers that were paid an amount greater than the cost of their respective courier services */

```
select c.*, p.amount, cs.cost
from courier c join payment p on c.id = p.courier_id
join courier_service cs on c.id = cs.id
where p.amount>cs.cost;
```

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 *
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_service
where cost<(select max(cost) from courier_service);
```

-- 52. Find all couriers that have been paid for

```
select c.*
from courier c join  payment p on c.id = p.courier_id;
```

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

```
select l.*
from location l join payment p on l.id = p.location_id
where p.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 c.*
from courier c
where c.weight > all (
                        select weight from courier
                        where sender_name = 'ram prasad');
```