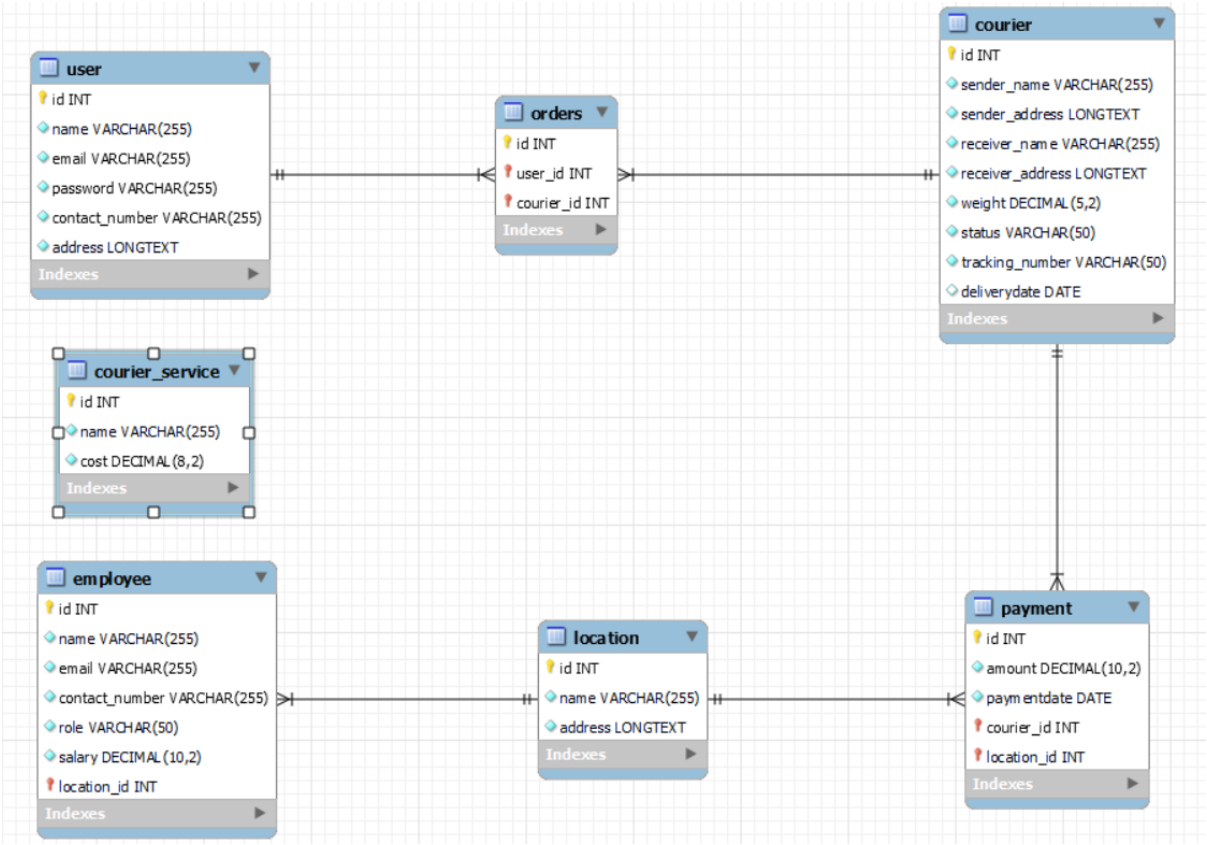
ASSIGNMENT NO 4

COURIER MANAGEMENT

ER DIAGRAM:



Task 1: Database Design

-- MySQL Workbench Forward Engineering

-- ---

-- Schema assignment_courier_management

-- ---

-- ---

-- Schema assignment_courier_management

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


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



-- -----------------------------------------------------
-- Table `assignment_courier_management`.`courier`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `assignment_courier_management`.`courier` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `sender_name` VARCHAR(255) NOT NULL,
  `sender_address` LONGTEXT NOT NULL,
  `receiver_name` VARCHAR(255) NOT NULL,
  `receiver_address` LONGTEXT NOT NULL,
  `weight` DECIMAL(5,2) NOT NULL,
  `status` VARCHAR(50) NOT NULL,
```

```
  `tracking_number` VARCHAR(50) NOT NULL,
  `deliverydate` DATE NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `tracking_number_UNIQUE` (`tracking_number` ASC) )
ENGINE = InnoDB
COMMENT = '
                                                                    ';



-- -----------------------------------------------------
-- Table `assignment_courier_management`.`location`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `assignment_courier_management`.`location` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(255) NOT NULL,
  `address` LONGTEXT NOT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;



-- -----------------------------------------------------
-- Table `assignment_courier_management`.`payment`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `assignment_courier_management`.`payment` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `amount` DECIMAL(10,2) NOT NULL,
  `paymentdate` DATE NOT NULL,
  `courier_id` INT NOT NULL,
  `location_id` INT NOT NULL,
  PRIMARY KEY (`id`, `courier_id`, `location_id`),
  INDEX `fk_payment_courier1_idx` (`courier_id` ASC),
```

```
  INDEX `fk_payment_location1_idx` (`location_id` ASC),
  CONSTRAINT `fk_payment_courier1`
    FOREIGN KEY (`courier_id`)
    REFERENCES `assignment_courier_management`.`courier` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_payment_location1`
    FOREIGN KEY (`location_id`)
    REFERENCES `assignment_courier_management`.`location` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `assignment_courier_management`.`employee`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `assignment_courier_management`.`employee` (
  `id` INT NOT NULL,
  `name` VARCHAR(255) NOT NULL,
  `email` VARCHAR(255) NOT NULL,
  `contact_number` VARCHAR(255) NOT NULL,
  `role` VARCHAR(50) NOT NULL,
  `salary` DECIMAL(10,2) NOT 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 `assignment_courier_management`.`location` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `assignment_courier_management`.`courier_service`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `assignment_courier_management`.`courier_service` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(255) NOT NULL,
  `cost` DECIMAL(8,2) NOT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `assignment_courier_management`.`orders`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `assignment_courier_management`.`orders` (
  `id` INT NOT NULL,
  `user_id` INT NOT NULL,
  `courier_id` INT NOT NULL,
  PRIMARY KEY (`user_id`, `courier_id`, `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 `assignment_courier_management`.`user` (`id`)
```

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

INSERTION:

```
insert into user(name,email,password,contact_number,address) values
('dhoni','dhoni@gmail.com','sakshi','95567','jharkhand farmer area'),
('virat','virat@gmail.com','as','88999','delhi'),
('jadeja','jaddu@gmail.com','riva','45555','gujarat'),
('ashwin','ash@gmail.com','ash','98899','chennai'),
('rohit','rk@gmail.com','rs','90009','mumbai'),
('shami','sh@gmail.com','md','45009','delhi'),
('shubman','gl@gmail.com','gl','67080','delhi'),
('siraj','sr@gmail.com','srj','33450','hyderabad'),
('bumrah','boom@gmail.com','boom','24566','mumbai'),
('rahul','rahul@gmail.com','kl','46890','karnataka');
```

```
mysql> select * from user;
+----+---------+-----------------+----------+----------------+----------------------+
| id | name    | email           | password | contact_number | address              |
+----+---------+-----------------+----------+----------------+----------------------+
|  1 | dhoni   | dhoni@gmail.com | sakshi   | 95567          | jharkhand farmer area |
|  2 | virat   | virat@gmail.com | as       | 88999          | delhi                |
|  3 | jadeja  | jaddu@gmail.com | riva     | 45555          | gujarat              |
|  4 | ashwin  | ash@gmail.com   | ash      | 98899          | chennai              |
|  5 | rohit   | rk@gmail.com    | rs       | 90009          | mumbai               |
|  6 | shami   | sh@gmail.com    | md       | 45009          | delhi                |
|  7 | shubman | gl@gmail.com    | gl       | 67080          | delhi                |
|  8 | siraj   | sr@gmail.com    | srj      | 33450          | hyderabad            |
|  9 | bumrah  | boom@gmail.com  | boom     | 24566          | mumbai               |
| 10 | rahul   | rahul@gmail.com | kl       | 46890          | karnataka            |
+----+---------+-----------------+----------+----------------+----------------------+
10 rows in set (0.02 sec)
```

INSERT INTO employee (id, name, email, contact_number, role, salary,location_id) VALUES

(1, 'axel', 'axel@gmail.com', '11111', 'Manager', 50000.00,1),

(2, 'bob', 'bob@gmail.com', '22222', 'Driver', 30000.00,2),

(3,'steyn','steyn@gmail.com','33333','Clerk',35000.00,3),

(4,'malinga','m@gmail.com','44444','Clerk',40000.00,4),

(5,'james','j@gmail.com','55555','packer',20000.00,5),

(6,'warner','w@gmail.com','66666','packer',20000.00,6),

(7,'russel','r@gmail.com','77777','Clerk',40000.00,7),

(8,'narine','n@gmail.com','88888','packer',20000.00,8),

(9,'broad','b@gmail.com','99999','packer',20000.00,9),

(10,'cummins','c@gmail.com','10101','Driver',30000.00,10);

```
mysql> select * from employee;
+----+---------+-----------------+----------------+---------+----------+-------------+
| id | name    | email           | contact_number | role    | salary   | location_id |
+----+---------+-----------------+----------------+---------+----------+-------------+
|  1 | axel    | axel@gmail.com  | 11111          | Manager | 50000.00 |           1 |
|  2 | bob     | bob@gmail.com   | 22222          | Driver  | 30000.00 |           2 |
|  3 | steyn   | steyn@gmail.com | 33333          | Clerk   | 35000.00 |           3 |
|  4 | malinga | m@gmail.com     | 44444          | Clerk   | 40000.00 |           4 |
|  5 | james   | j@gmail.com     | 55555          | packer  | 20000.00 |           5 |
|  6 | warner  | w@gmail.com     | 66666          | packer  | 20000.00 |           6 |
|  7 | russel  | r@gmail.com     | 77777          | Clerk   | 40000.00 |           7 |
|  8 | narine  | n@gmail.com     | 88888          | packer  | 20000.00 |           8 |
|  9 | broad   | b@gmail.com     | 99999          | packer  | 20000.00 |           9 |
| 10 | cummins | c@gmail.com     | 10101          | Driver  | 30000.00 |          10 |
+----+---------+-----------------+----------------+---------+----------+-------------+
10 rows in set (0.00 sec)
```

INSERT INTO courier (id, sender_name, sender_address, receiver_name, receiver_address, Weight, Status, tracking_number, deliverydate) VALUES

(1, 'dhoni', 'jharkhand farmer area', 'jadeja', 'gujarat', 2.5, 'Delivered', 'TN123456', '2024-03-02'),

(2, 'virat', 'delhi', 'dhoni', 'jharkhand farmer area', 5.3, 'Delivered', 'TN739012', '2024-03-05'),

(3, 'rohit', 'mumbai', 'dhoni', 'jharkhand farmer area', 8.3, 'Delivered', 'TN789012', '2024-02-25'),

(4, 'ashwin', 'chennai', 'rohit', 'mumbai', 7.3, 'Delivered', 'TN759012', '2024-03-15'),

(5, 'virat', 'delhi', 'rohit', 'mumbai', 5.3, 'Delivered', 'TN786012', '2024-03-08'),

(6, 'shami','delhi','ashwin','chennai',4.8,'shipping started','TN845900','2024-03-27'),

(7,'gill','delhi','rohit','mumbai',9.3,'shipping started','TN456789','2024-03-28'),

(8,'jadeja','gujarat','bumrah','mumbai',4.9,'ordered','TN342109','2024-02-19'),

(9,'rahul','karnataka','gill','delhi',6.2,'ordered','TN383109','2024-03-06'),

(10,'siraj','gujarat','bumrah','mumbai',5.8,'ordered','TN377109','2024-02-17');

```
mysql> select * from courier;
+----+-------------+----------------------+---------------+----------------------+--------+-----------------+-----------------+-------------+
| id | sender_name | sender_address       | receiver_name | receiver_address     | weight | status          | tracking_number | deliverydate |
+----+-------------+----------------------+---------------+----------------------+--------+-----------------+-----------------+-------------+
|  1 | dhoni       | jharkhand farmer area | jadeja        | gujarat              |   2.50 | Delivered       | TN123456        | 2024-03-02   |
|  2 | virat       | delhi                | dhoni         | jharkhand farmer area |   5.30 | Delivered       | TN739012        | 2024-03-05   |
|  3 | rohit       | mumbai               | dhoni         | jharkhand farmer area |   8.30 | Delivered       | TN789012        | 2024-02-25   |
|  4 | ashwin      | chennai              | rohit         | mumbai               |   7.30 | Delivered       | TN759012        | 2024-03-15   |
|  5 | virat       | delhi                | rohit         | mumbai               |   5.30 | Delivered       | TN786012        | 2024-03-08   |
|  6 | shami       | delhi                | ashwin        | chennai              |   4.80 | shipping started | TN845900        | 2024-03-27   |
|  7 | gill        | delhi                | rohit         | mumbai               |   9.30 | shipping started | TN456789        | 2024-03-28   |
|  8 | jadeja      | gujarat              | bumrah        | mumbai               |   4.90 | ordered         | TN342109        | 2024-02-19   |
|  9 | rahul       | karnataka            | gill          | delhi                |   6.20 | ordered         | TN383109        | 2024-03-06   |
| 10 | siraj       | gujarat              | bumrah        | mumbai               |   5.80 | ordered         | TN377109        | 2024-02-17   |
+----+-------------+----------------------+---------------+----------------------+--------+-----------------+-----------------+-------------+
10 rows in set (0.01 sec)
```

insert into orders(id,user_id,courier_id) values

(1,1,1),

(2,2,2),

(3,3,3),

(4,4,4),

(5,5,5),

(6,6,6),

(7,7,7),

(8,8,8),

(9,9,9),

(10,10,10);

```
mysql> select * from orders;
+----+---------+------------+
| id | user_id | courier_id |
+----+---------+------------+
|  1 |       1 |          1 |
|  2 |       2 |          2 |
|  3 |       3 |          3 |
|  4 |       4 |          4 |
|  5 |       5 |          5 |
|  6 |       6 |          6 |
|  7 |       7 |          7 |
|  8 |       8 |          8 |
|  9 |       9 |          9 |
| 10 |      10 |         10 |
+----+---------+------------+
10 rows in set (0.00 sec)
```

INSERT INTO Location (id, name, address) VALUES

(1, 'Warehouse A', '789 pqr St'),

(2, 'Warehouse B', '987 abc St'),

(3, 'Warehouse C', '089 Elm St'),

(4, 'Warehouse D', '787 kkr St'),

(5, 'Warehouse E', '787 rbc St'),

(6, 'Warehouse F', '587 who St'),

(7, 'Warehouse G', '795 gt St'),

(8, 'Warehouse H', '777 mi St'),

(9, 'Warehouse I', '457 csk St'),

(10, 'Warehouse J', '987 rcb St');

```
mysql> select * from location;
+----+-------------+------------+
| id | name        | address    |
+----+-------------+------------+
|  1 | Warehouse A | 789 pqr St |
|  2 | Warehouse B | 987 abc St |
|  3 | Warehouse C | 089 Elm St |
|  4 | Warehouse D | 787 kkr St |
|  5 | Warehouse E | 787 rbc St |
|  6 | Warehouse F | 587 who St |
|  7 | Warehouse G | 795 gt St  |
|  8 | Warehouse H | 777 mi St  |
|  9 | Warehouse I | 457 csk St |
| 10 | Warehouse J | 987 rcb St |
+----+-------------+------------+
10 rows in set (0.00 sec)
```

INSERT INTO courier_service (id, name, cost) VALUES

(1, 'Standard', 1000),

(2, 'Express', 2000),

(3, 'Valuables', 5000),

(4,'fast',3000),

(5,'speed',2000);

```
mysql> select * from courier_service;
+----+-----------+---------+
| id | name      | cost    |
+----+-----------+---------+
|  1 | Standard  | 1000.00 |
|  2 | Express   | 2000.00 |
|  3 | Valuables | 5000.00 |
|  4 | fast      | 3000.00 |
|  5 | speed     | 2000.00 |
+----+-----------+---------+
5 rows in set (0.00 sec)
```

INSERT INTO payment (id, amount, paymentdate, location_id, courier_id) VALUES

(1, 2500, '2024-03-01',1,1),

(2, 5500, '2024-03-02',2,2),

(3, 3850, '2024-02-23',3,3),

(4, 1350, '2024-03-11',4,4),

(5, 6500, '2024-03-02',5,5),

(6,2200,'2024-02-24',6,6),

(7,2500,'2024-03-03',7,7),

(8,2750,'2024-03-18',8,8),

(9,3000,'2024-03-22',9,9),

(10,4000,'2024-01-16',10,10);

```
mysql> select * from payment;
+----+---------+-------------+------------+-------------+
| id | amount  | paymentdate | courier_id | location_id |
+----+---------+-------------+------------+-------------+
|  1 | 2500.00 | 2024-03-01  |          1 |           1 |
|  2 | 5500.00 | 2024-03-02  |          2 |           2 |
|  3 | 3850.00 | 2024-02-23  |          3 |           3 |
|  4 | 1350.00 | 2024-03-11  |          4 |           4 |
|  5 | 6500.00 | 2024-03-02  |          5 |           5 |
|  6 | 2200.00 | 2024-02-24  |          6 |           6 |
|  7 | 2500.00 | 2024-03-03  |          7 |           7 |
|  8 | 2750.00 | 2024-03-18  |          8 |           8 |
|  9 | 3000.00 | 2024-03-22  |          9 |           9 |
| 10 | 4000.00 | 2024-01-16  |         10 |          10 |
+----+---------+-------------+------------+-------------+
10 rows in set (0.00 sec)
```

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='ashwin';
```

3. List all couriers:

```
select * from courier;
```

4. List all packages for a specific order:

```
select o.id,c.* from orders o,courier c
where o.courier_id=c.id
and o.id=6;
```

5. List all deliveries for a specific courier:

```
select id from orders
where courier_id=7;
```

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 deliverydate=curdate();
```

8. List all packages with a specific status:

```
select * from courier
where status='shipping started';
```

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

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

10. Find the average delivery time for each courier

```
select sender_name, avg(datediff(deliverydate,curdate())) as avg_del_date
from courier
group by id;
```

11. List all packages with a specific weight range:

```
select * from courier
where weight between 4 and 8;
```

12. Retrieve employees whose names contain 'ha'

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

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

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

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

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

```
select e.name, count(e.id) as total_couriers_handled
from employee e , courier c
where e.id=c.id
group by e.name;
```

15. Calculate the total revenue generated by each location

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

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

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

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

```
select sender_name, avg(datediff(deliverydate,curdate())) as avg_del_date
from courier
order by avg_del_date limit 0,1;
```

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

```
select l.*, p.amount
from location l , payment p
where l.id=p.location_id
and p.amount<5000;
```

19. Calculate Total Payments per Location

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

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 p.courier_id=c.id
and p.amount>=1000 and paymentdate<'2024-03-02';
```

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_amount
 from location l ,payment p
 where l.id=p.location_id and p.paymentdate>'2024-03-01'
 group by l.id
 having total_amount>5000;
```

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

23. Retrieve Payments with Courier Information

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

24. Retrieve Payments with Location Information

```
select p.*,l.name as city_name,l.address
from payment p join location l
on p.location_id=l.id;
```

25. Retrieve Payments with Courier and Location Information

```
select c.*,p.amount,p.paymentdate,l.name,l.address
from courier c join payment p on c.id=p.courier_id
join location l on l.id=p.location_id;
```

26. List all payments with courier details

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

27. Total payments received for each courier

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

28. List payments made on a specific date

```
select c.*,p.amount,p.paymentdate
from courier c join payment p on c.id=p.courier_id
where p.paymentdate='2024-03-12';
```

29. Get Courier Information for Each Payment

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

30. Get Payment Details with Location

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

31. Calculating Total Payments for Each Courier

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

32. List Payments Within a Date Range

```
select * from payment
 where paymentdate between '2024-03-01' and '2024-03-22';
```

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.sender_name,c.sender_address,c.receiver_name,c.receiver_address
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.name,sum(p.amount)
 from employee e join payment p on p.location_id=e.location_id
 group by e.location_id;
```

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

```
select * from user,courier_service;
```

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 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 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;
```

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

```
select e.name,count(c.id) as courier_assigned
 from employee e join courier c on c.id=e.id
 group by e.id;
```

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

```
select l.name,sum(p.amount) as total_paymnet
 from location l join payment p
 on p.location_id=l.id
 group by l.id;
```

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

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

44. List all employees who share the same role.

```
select role,group_concat(name) from employee group by role;
```

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

```
select l.name as city,l.address, sum(p.amount) as total_payment
 from location l join payment p on p.location_id=l.id
 group by l.id;
```

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

```
select * from courier
 where sender_address in (select sender_address from courier
 group by sender_address
 having count(*)>1);
```

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

```
 select e.name,count(c.id) as courier_delivered
 from employee e join courier c on c.id=e.id
 where 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.* from
 courier c join payment p on c.id=p.courier_id
 join courier_service cs on c.id=cs.id
 where cs.cost<p.amount;
```

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 * 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 p.courier_id=c.id
 where p.paymentdate< curdate();
```

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

```
select l.name as city,l.address , sum(p.amount) as total_payment
 from location l join payment p on p.location_id=l.id
 group by l.id
 order by total_payment desc;
```

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 sum(weight) from courier where sender_name='shami');
```