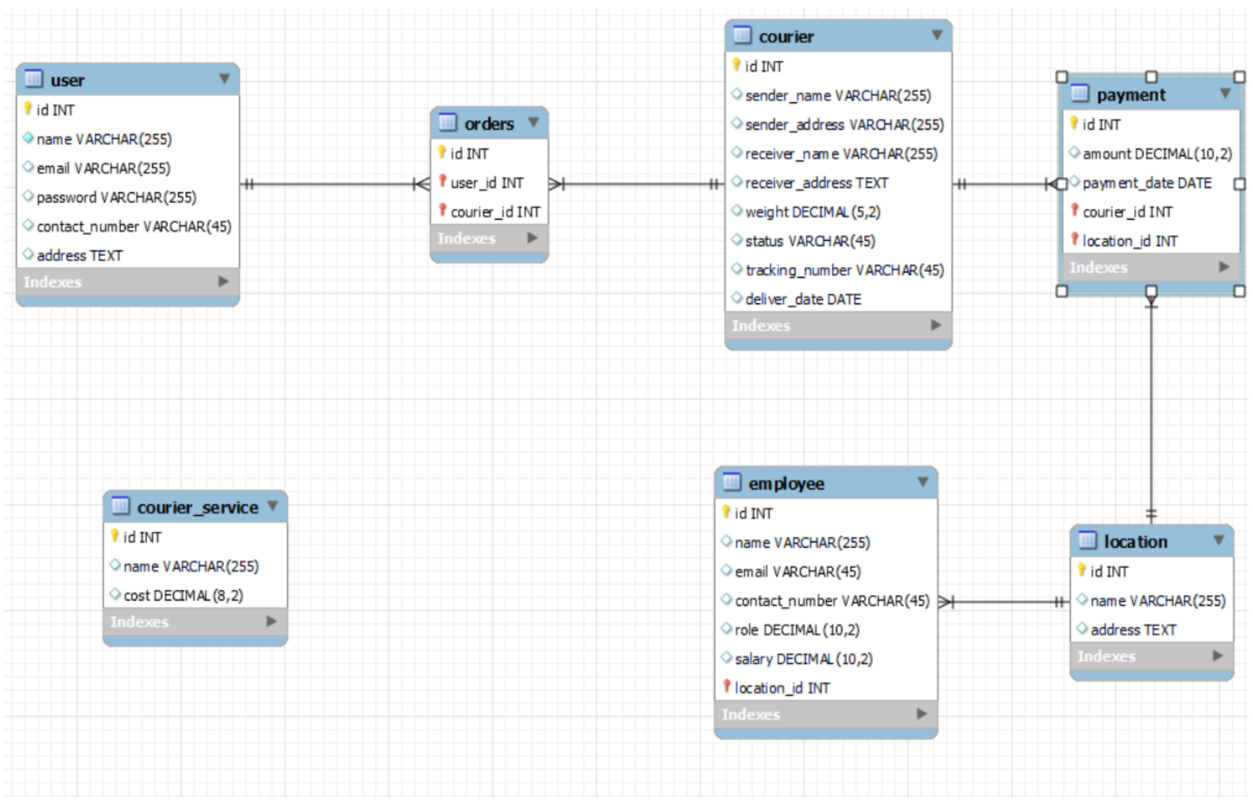
ASSIGNMENT NO : 3

COURIER MANAGEMENT SYSTEM

ER DIAGRAM:



Task:1. Database Design:

```
-- MySQL Workbench Forward Engineering


-- -----------------------------------------------------
-- Schema couriermg
-- -----------------------------------------------------

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

-- -----------------------------------------------------
-- Table `couriermg`.`user`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `couriermg`.`user` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(255) NOT NULL,
  `email` VARCHAR(255) NULL,
  `password` VARCHAR(255) NULL,
  `contact_number` VARCHAR(45) NULL,
  `address` TEXT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `contact_number_UNIQUE` (`contact_number` ASC) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `couriermg`.`courier`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `couriermg`.`courier` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `sender_name` VARCHAR(255) NULL,
  `sender_address` VARCHAR(255) NULL,
  `receiver_name` VARCHAR(255) NULL,
  `receiver_address` TEXT NULL,
  `weight` DECIMAL(5,2) NULL,
  `status` VARCHAR(45) NULL,
  `tracking_number` VARCHAR(45) NULL,
  `deliver_date` DATE NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `couriermg`.`orders`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `couriermg`.`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_orders_user_idx` (`user_id` ASC) ,
  INDEX `fk_orders_courier1_idx` (`courier_id` ASC) ,
  CONSTRAINT `fk_orders_user`
    FOREIGN KEY (`user_id`)
    REFERENCES `couriermg`.`user` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_orders_courier1`
    FOREIGN KEY (`courier_id`)
    REFERENCES `couriermg`.`courier` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
```

```
ENGINE = InnoDB;


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


-- -----------------------------------------------------
-- Table `couriermg`.`payment`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `couriermg`.`payment` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `amount` DECIMAL(10,2) NULL,
  `payment_date` DATE 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 `couriermg`.`courier` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_payment_location1`
    FOREIGN KEY (`location_id`)
    REFERENCES `couriermg`.`location` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


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

```
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `couriermg`.`employee`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `couriermg`.`employee` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(255) NULL,
  `email` VARCHAR(45) NULL,
  `contact_number` VARCHAR(45) NULL,
  `role` DECIMAL(10,2) NULL,
  `salary` DECIMAL(10,2) NULL,
  `location_id` INT NOT NULL,
  PRIMARY KEY (`id`, `location_id`),
  UNIQUE INDEX `contact_number_UNIQUE` (`contact_number` ASC) ,
  INDEX `fk_employee_location1_idx` (`location_id` ASC) ,
  CONSTRAINT `fk_employee_location1`
    FOREIGN KEY (`location_id`)
    REFERENCES `couriermg`.`location` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

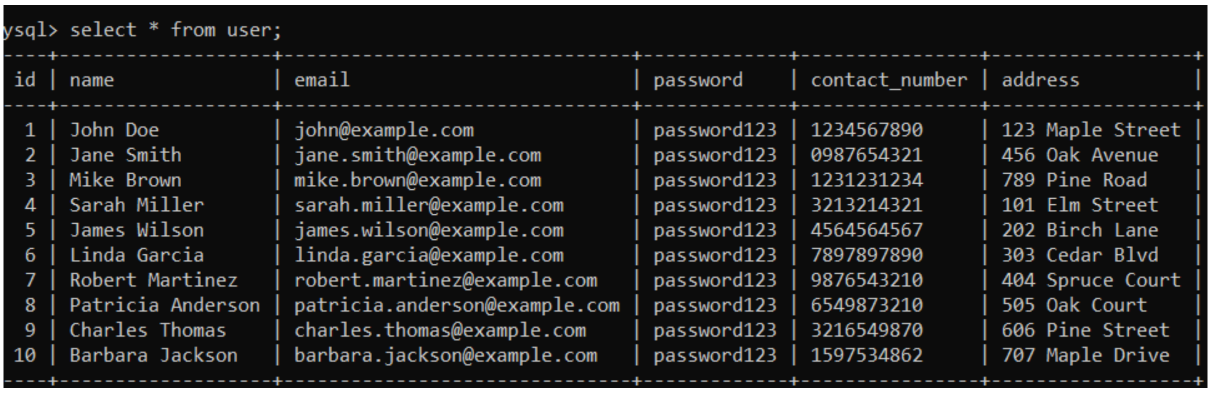
INSERTION:

```
-- user insertion


INSERT INTO `couriermg`.`user` (`name`, `email`, `password`, `contact_number`, `address`)
VALUES

('John Doe', 'john@example.com', 'password123', '1234567890', '123 Maple Street'),

('Jane Smith', 'jane.smith@example.com', 'password123', '0987654321', '456 Oak Avenue'),

('Mike Brown', 'mike.brown@example.com', 'password123', '1231231234', '789 Pine Road'),

('Sarah Miller', 'sarah.miller@example.com', 'password123', '3213214321', '101 Elm Street'),

('James Wilson', 'james.wilson@example.com', 'password123', '4564564567', '202 Birch
Lane'),

('Linda Garcia', 'linda.garcia@example.com', 'password123', '7897897890', '303 Cedar Blvd'),

('Robert Martinez', 'robert.martinez@example.com', 'password123', '9876543210', '404
Spruce Court'),
```

('Patricia Anderson', 'patricia.anderson@example.com', 'password123', '6549873210', '505 Oak Court'),

('Charles Thomas', 'charles.thomas@example.com', 'password123', '3216549870', '606 Pine Street'),

('Barbara Jackson', 'barbara.jackson@example.com', 'password123', '1597534862', '707 Maple Drive');



```
ysql> select * from user;
+----+------------------+-------------------------------+-------------+----------------+-----------------+
| id | name             | email                         | password    | contact_number | address         |
+----+------------------+-------------------------------+-------------+----------------+-----------------+
|  1 | John Doe         | john@example.com              | password123 | 1234567890     | 123 Maple Street |
|  2 | Jane Smith       | jane.smith@example.com        | password123 | 0987654321     | 456 Oak Avenue  |
|  3 | Mike Brown       | mike.brown@example.com        | password123 | 1231231234     | 789 Pine Road   |
|  4 | Sarah Miller     | sarah.miller@example.com      | password123 | 3213214321     | 101 Elm Street  |
|  5 | James Wilson     | james.wilson@example.com      | password123 | 4564564567     | 202 Birch Lane  |
|  6 | Linda Garcia     | linda.garcia@example.com      | password123 | 7897897890     | 303 Cedar Blvd  |
|  7 | Robert Martinez  | robert.martinez@example.com   | password123 | 9876543210     | 404 Spruce Court |
|  8 | Patricia Anderson | patricia.anderson@example.com | password123 | 6549873210     | 505 Oak Court   |
|  9 | Charles Thomas   | charles.thomas@example.com    | password123 | 3216549870     | 606 Pine Street |
| 10 | Barbara Jackson  | barbara.jackson@example.com   | password123 | 1597534862     | 707 Maple Drive |
+----+------------------+-------------------------------+-------------+----------------+-----------------+
```

-- courier insertion

INSERT INTO courier(`sender_name`, `sender_address`, `receiver_name`, `receiver_address`, `weight`, `status`, `tracking_number`, `deliver_date`) VALUES

('Karthik Srinivasan', '33 Ram Nagar, Coimbatore', 'Surya Raghavan', '77 Gandhi Road, Madurai', 2.1, 'Dispatched', 'TRK1002003', '2023-09-10'),

('Anitha Gopal', '15 Nehru Street, Tiruchirappalli', 'Balaji Murugan', '89 Chetty Street, Chennai', 1.5, 'In Transit', 'TRK1002004', '2023-09-12'),

('Meena Kandasamy', '42 Bose Lane, Salem', 'Rajesh Kumar', '112 Patel Road, Erode', 3.3, 'Delivered', 'TRK1002005', '2023-09-14'),

('Priya Dharshini', '25 Vivekanandar Street, Thanjavur', 'Vijay Anand', '63 Main Road, Tirunelveli', 2.4, 'In Transit', 'TRK1002006', '2023-09-16'),

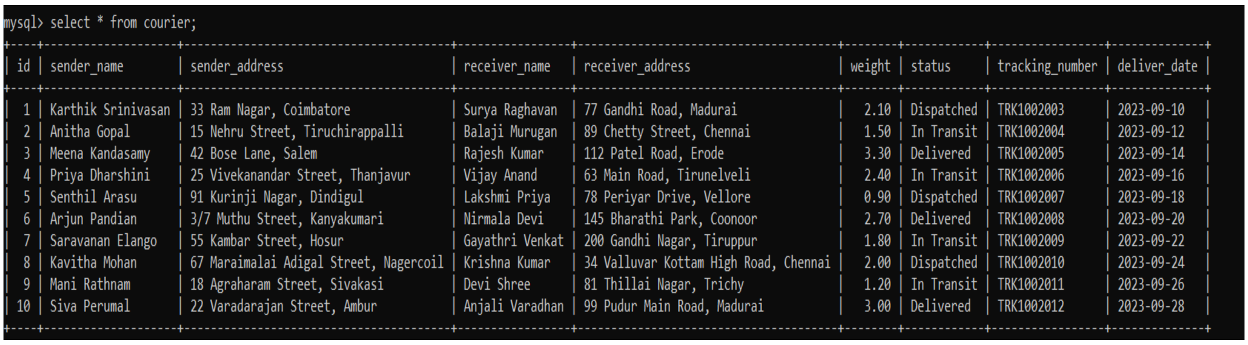
('Senthil Arasu', '91 Kurinji Nagar, Dindigul', 'Lakshmi Priya', '78 Periyar Drive, Vellore', 0.9, 'Dispatched', 'TRK1002007', '2023-09-18'),

('Arjun Pandian', '3/7 Muthu Street, Kanyakumari', 'Nirmala Devi', '145 Bharathi Park, Coonoor', 2.7, 'Delivered', 'TRK1002008', '2023-09-20'),

('Saravanan Elango', '55 Kambar Street, Hosur', 'Gayathri Venkat', '200 Gandhi Nagar, Tiruppur', 1.8, 'In Transit', 'TRK1002009', '2023-09-22'),

('Kavitha Mohan', '67 Maraimalai Adigal Street, Nagercoil', 'Krishna Kumar', '34 Valluvar Kottam High Road, Chennai', 2.0, 'Dispatched', 'TRK1002010', '2023-09-24'), ('Mani Rathnam', '18 Agraharam Street, Sivakasi', 'Devi Shree', '81 Thillai Nagar, Trichy', 1.2, 'In Transit', 'TRK1002011', '2023-09-26'),

('Siva Perumal', '22 Varadarajan Street, Ambur', 'Anjali Varadhan', '99 Pudur Main Road, Madurai', 3.0, 'Delivered', 'TRK1002012', '2023-09-28');



```
mysql> select * from courier;
+----+--------------------+----------------------------------------+-----------------+---------------------------------------+--------+------------+-----------------+--------------+
| id | sender_name        | sender_address                         | receiver_name   | receiver_address                      | weight | status     | tracking_number | deliver_date |
+----+--------------------+----------------------------------------+-----------------+---------------------------------------+--------+------------+-----------------+--------------+
|  1 | Karthik Srinivasan | 33 Ram Nagar, Coimbatore               | Surya Raghavan  | 77 Gandhi Road, Madurai               |   2.10 | Dispatched | TRK1002003      | 2023-09-10   |
|  2 | Anitha Gopal       | 15 Nehru Street, Tiruchirappalli       | Balaji Murugan  | 89 Chetty Street, Chennai             |   1.50 | In Transit | TRK1002004      | 2023-09-12   |
|  3 | Meena Kandasamy    | 42 Bose Lane, Salem                    | Rajesh Kumar    | 112 Patel Road, Erode                 |   3.30 | Delivered  | TRK1002005      | 2023-09-14   |
|  4 | Priya Dharshini    | 25 Vivekanandar Street, Thanjavur      | Vijay Anand     | 63 Main Road, Tirunelveli             |   2.40 | In Transit | TRK1002006      | 2023-09-16   |
|  5 | Senthil Arasu      | 91 Kurinji Nagar, Dindigul             | Lakshmi Priya   | 78 Periyar Drive, Vellore             |   0.90 | Dispatched | TRK1002007      | 2023-09-18   |
|  6 | Arjun Pandian      | 3/7 Muthu Street, Kanyakumari          | Nirmala Devi    | 145 Bharathi Park, Coonoor            |   2.70 | Delivered  | TRK1002008      | 2023-09-20   |
|  7 | Saravanan Elango   | 55 Kambar Street, Hosur                | Gayathri Venkat | 200 Gandhi Nagar, Tiruppur            |   1.80 | In Transit | TRK1002009      | 2023-09-22   |
|  8 | Kavitha Mohan      | 67 Maraimalai Adigal Street, Nagercoil | Krishna Kumar   | 34 Valluvar Kottam High Road, Chennai |   2.00 | Dispatched | TRK1002010      | 2023-09-24   |
|  9 | Mani Rathnam       | 18 Agraharam Street, Sivakasi          | Devi Shree      | 81 Thillai Nagar, Trichy              |   1.20 | In Transit | TRK1002011      | 2023-09-26   |
| 10 | Siva Perumal       | 22 Varadarajan Street, Ambur           | Anjali Varadhan | 99 Pudur Main Road, Madurai           |   3.00 | Delivered  | TRK1002012      | 2023-09-28   |
+----+--------------------+----------------------------------------+-----------------+---------------------------------------+--------+------------+-----------------+--------------+
```

-- courier_service insertion

INSERT INTO courier_service (`name`, `cost`) VALUES

('Anbu Parcel Service', 150.00),

('Vetri Fast Delivery', 200.00),

('Malar Door-to-Door Service', 175.00),

('Kodi Logistics Solutions', 225.00),

('Thamarai Express Delivery', 250.00),

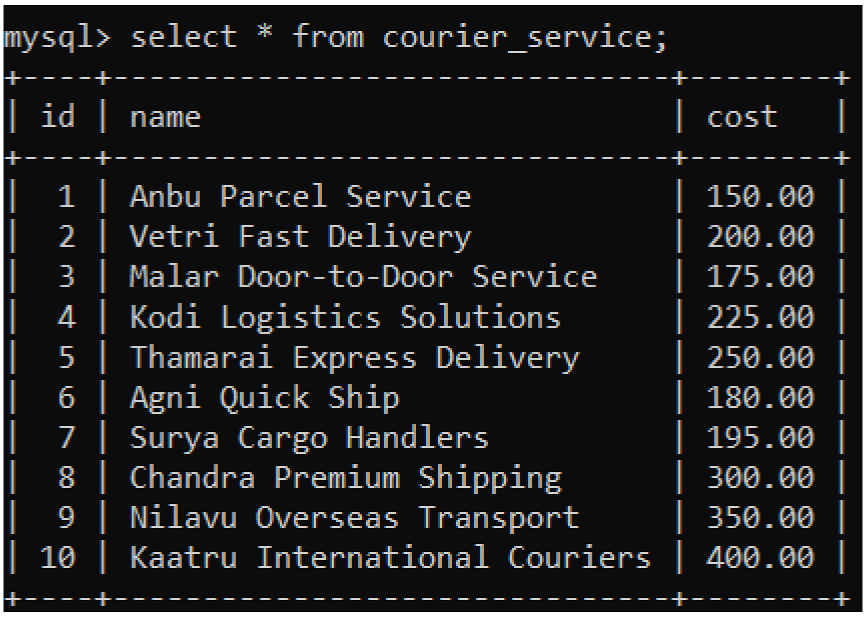
('Agni Quick Ship', 180.00),

('Surya Cargo Handlers', 195.00),

('Chandra Premium Shipping', 300.00),

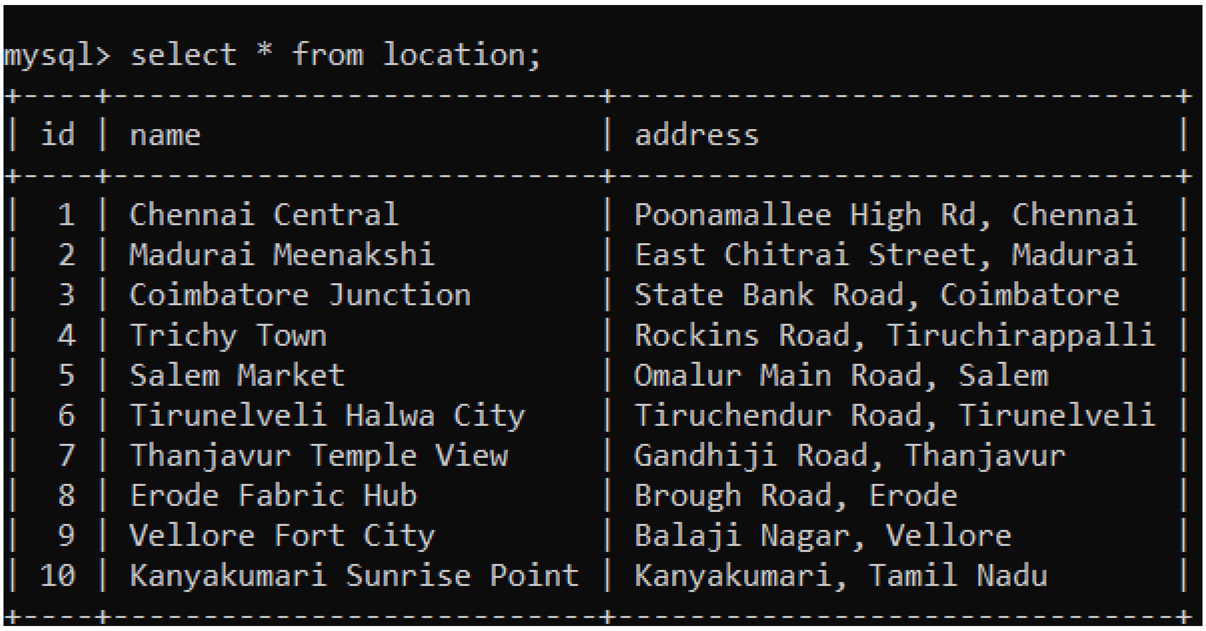
('Nilavu Overseas Transport', 350.00),

('Kaatru International Couriers', 400.00);



```
mysql> select * from courier_service;
+----+--------------------------------+--------+
| id | name                           | cost   |
+----+--------------------------------+--------+
|  1 | Anbu Parcel Service            | 150.00 |
|  2 | Vetri Fast Delivery            | 200.00 |
|  3 | Malar Door-to-Door Service     | 175.00 |
|  4 | Kodi Logistics Solutions       | 225.00 |
|  5 | Thamarai Express Delivery      | 250.00 |
|  6 | Agni Quick Ship                | 180.00 |
|  7 | Surya Cargo Handlers           | 195.00 |
|  8 | Chandra Premium Shipping       | 300.00 |
|  9 | Nilavu Overseas Transport      | 350.00 |
| 10 | Kaatru International Couriers  | 400.00 |
+----+--------------------------------+--------+
```

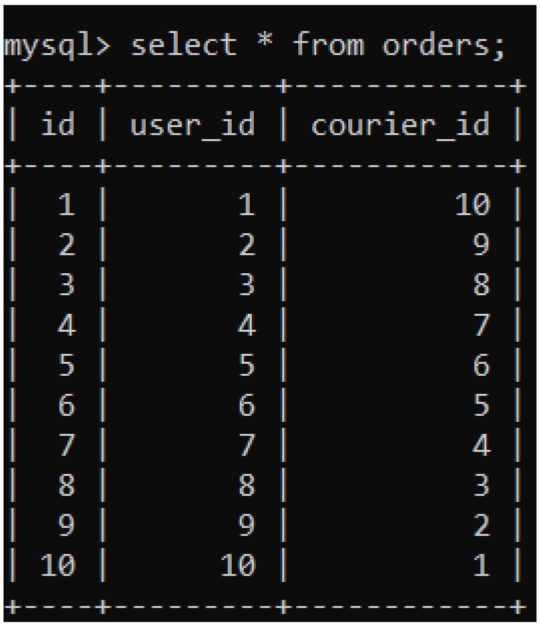
```
-- location insertion
INSERT INTO location(`name`, `address`) VALUES
('Chennai Central', 'Poonamallee High Rd, Chennai'),
('Madurai Meenakshi', 'East Chitrai Street, Madurai'),
('Coimbatore Junction', 'State Bank Road, Coimbatore'),
('Trichy Town', 'Rockins Road, Tiruchirappalli'),
('Salem Market', 'Omalur Main Road, Salem'),
('Tirunelveli Halwa City', 'Tiruchendur Road, Tirunelveli'),
('Thanjavur Temple View', 'Gandhiji Road, Thanjavur'),
('Erode Fabric Hub', 'Brough Road, Erode'),
('Vellore Fort City', 'Balaji Nagar, Vellore'),
('Kanyakumari Sunrise Point', 'Kanyakumari, Tamil Nadu');
```



```
mysql> select * from location;
+----+---------------------------+-------------------------------+
| id | name                      | address                       |
+----+---------------------------+-------------------------------+
|  1 | Chennai Central           | Poonamallee High Rd, Chennai  |
|  2 | Madurai Meenakshi         | East Chitrai Street, Madurai  |
|  3 | Coimbatore Junction       | State Bank Road, Coimbatore   |
|  4 | Trichy Town               | Rockins Road, Tiruchirappalli |
|  5 | Salem Market              | Omalur Main Road, Salem       |
|  6 | Tirunelveli Halwa City    | Tiruchendur Road, Tirunelveli |
|  7 | Thanjavur Temple View     | Gandhiji Road, Thanjavur      |
|  8 | Erode Fabric Hub          | Brough Road, Erode            |
|  9 | Vellore Fort City         | Balaji Nagar, Vellore         |
| 10 | Kanyakumari Sunrise Point | Kanyakumari, Tamil Nadu       |
+----+---------------------------+-------------------------------+
```

-- orders insertion

```
INSERT INTO orders (`user_id`, `courier_id`) VALUES
(1, 10),(2, 9),(3, 8),(4, 7),(5, 6),(6, 5),(7, 4),(8, 3),(9, 2),(10, 1);
```



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

-- payment insertion

```
INSERT INTO payment (`amount`, `payment_date`, `courier_id`, `location_id`) VALUES
(500.00, '2023-09-01', 1, 1),
(750.00, '2023-09-02', 2, 2),
```

(600.00, '2023-09-03', 3, 3),

(450.00, '2023-09-04', 4, 4),

(800.00, '2023-09-05', 5, 5),

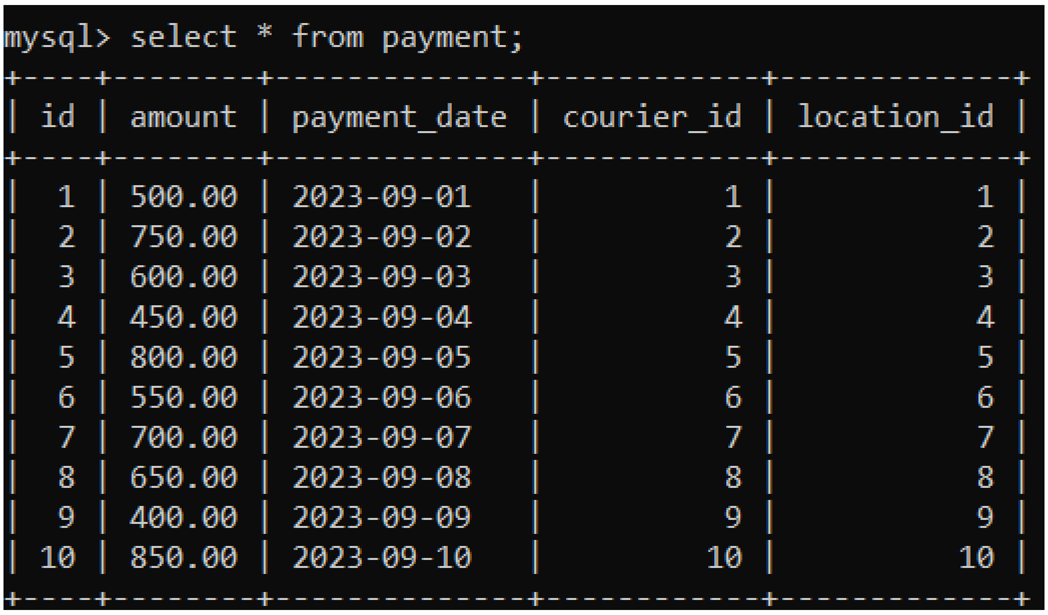
(550.00, '2023-09-06', 6, 6),

(700.00, '2023-09-07', 7, 7),

(650.00, '2023-09-08', 8, 8),

(400.00, '2023-09-09', 9, 9),

(850.00, '2023-09-10', 10, 10);



```
mysql> select * from payment;
+----+--------+--------------+------------+-------------+
| id | amount | payment_date | courier_id | location_id |
+----+--------+--------------+------------+-------------+
|  1 | 500.00 | 2023-09-01   |          1 |           1 |
|  2 | 750.00 | 2023-09-02   |          2 |           2 |
|  3 | 600.00 | 2023-09-03   |          3 |           3 |
|  4 | 450.00 | 2023-09-04   |          4 |           4 |
|  5 | 800.00 | 2023-09-05   |          5 |           5 |
|  6 | 550.00 | 2023-09-06   |          6 |           6 |
|  7 | 700.00 | 2023-09-07   |          7 |           7 |
|  8 | 650.00 | 2023-09-08   |          8 |           8 |
|  9 | 400.00 | 2023-09-09   |          9 |           9 |
| 10 | 850.00 | 2023-09-10   |         10 |          10 |
+----+--------+--------------+------------+-------------+
```

-- employee insertion

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

('Saravanan Muthu', 'saravanan.muthu@gmail.com', '9887654321', 1, 27000.00, 1),

('Manoj kumar', 'manoj.kumar@gmail.com', '9887654322', 2, 29000.00, 2),

('Kumar Ganesan', 'kumar.ganesan@gmail.com', '9887654323', 3, 31000.00, 3),

('vinay kumar', 'vinay.kumar@gmail.com', '9887654324', 4, 24000.00, 4),

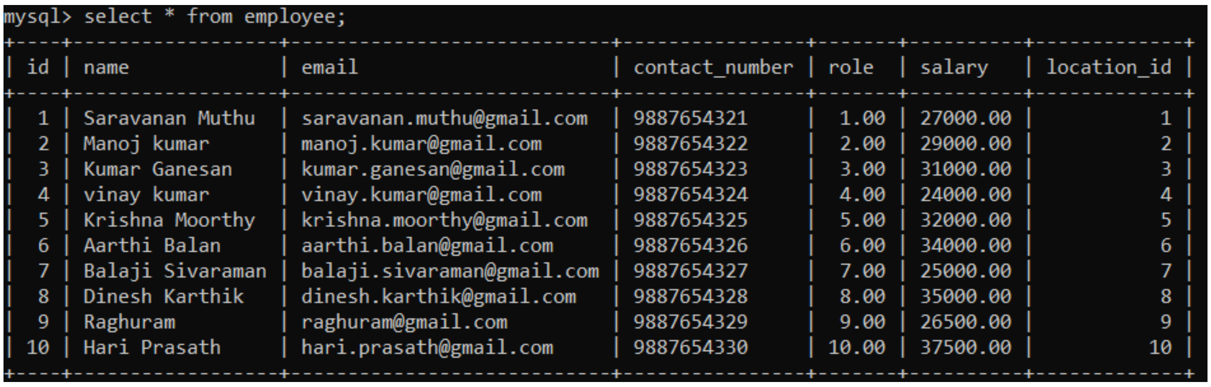
('Krishna Moorthy', 'krishna.moorthy@gmail.com', '9887654325', 5, 32000.00, 5),

('Aarthi Balan', 'aarthi.balan@gmail.com', '9887654326', 6, 34000.00, 6),

('Balaji Sivaraman', 'balaji.sivaraman@gmail.com', '9887654327', 7, 25000.00, 7), ('Dinesh Karthik', 'dinesh.karthik@gmail.com', '9887654328', 8, 35000.00, 8),

('Raghuram', 'raghuram@gmail.com', '9887654329', 9, 26500.00, 9),

('Hari Prasath', 'hari.prasath@gmail.com', '9887654330', 10, 37500.00, 10);



```
mysql> select * from employee;
+----+------------------+--------------------------+----------------+-------+----------+-------------+
| id | name             | email                    | contact_number | role  | salary   | location_id |
+----+------------------+--------------------------+----------------+-------+----------+-------------+
|  1 | Saravanan Muthu  | saravanan.muthu@gmail.com | 9887654321    |  1.00 | 27000.00 |           1 |
|  2 | Manoj kumar      | manoj.kumar@gmail.com    | 9887654322     |  2.00 | 29000.00 |           2 |
|  3 | Kumar Ganesan    | kumar.ganesan@gmail.com  | 9887654323     |  3.00 | 31000.00 |           3 |
|  4 | vinay kumar      | vinay.kumar@gmail.com    | 9887654324     |  4.00 | 24000.00 |           4 |
|  5 | Krishna Moorthy  | krishna.moorthy@gmail.com | 9887654325    |  5.00 | 32000.00 |           5 |
|  6 | Aarthi Balan     | aarthi.balan@gmail.com   | 9887654326     |  6.00 | 34000.00 |           6 |
|  7 | Balaji Sivaraman | balaji.sivaraman@gmail.com | 9887654327   |  7.00 | 25000.00 |           7 |
|  8 | Dinesh Karthik   | dinesh.karthik@gmail.com | 9887654328     |  8.00 | 35000.00 |           8 |
|  9 | Raghuram         | raghuram@gmail.com       | 9887654329     |  9.00 | 26500.00 |           9 |
| 10 | Hari Prasath     | hari.prasath@gmail.com   | 9887654330     | 10.00 | 37500.00 |          10 |
+----+------------------+--------------------------+----------------+-------+----------+-------------+
```

Task 2: Select,Where

1. List all customers:

    select * from user;

2. List all orders for a specific customer:

    select o.id AS OrderID, o.user_id, o.courier_id from orders o join user u on o.user_id = u.id where u.email = 'john@example.com';

3. List all couriers:

    Select * from courier;

4. List all packages for a specific order:

    select c.* from courier c JOIN orders on c.id = o.courier_id where o.id = 1;

5. List all deliveries for a specific courier:

    select o.* from orders o where o.courier_id = 1;

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 deliver_date = CURDATE();
```

8. List all packages with a specific status:

```
Select  * from courier where status = 'In Transit';
```

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

```
Select courier_id, count(*) as total_packages from orders

Group by courier_id;
```

10. Find the average delivery time for each courier

```
Select  courier_id, AVG(DATEDIFF(deliver_date, payment_date)) as
average_delivery_time_days from courier,payment join orders ON courier.id =
orders.courier_id group by courier_id;
```

11. List all packages with a specific weight range:

```
Select * from courier where weight between 1.0 and 2.0;
```

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.* from courier c join payment p on c.id = p.courier_id where p.amount >
50.00;
```

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

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

15. Calculate the total revenue generated by each location

```
Select l.id as LocationID, l.name as LocationName, SUM(p.amount) as TotalRevenue

From location l join payment p ON l.id = p.location_id group by l.id, l.name;
```

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

```
Select l.id as LocationID, l.name as LocationName, COUNT(c.id)
```

as TotalDeliveredCouriers from location l JOIN courier c ON l.id = c.location_id

WHERE c.status = 'Delivered' GROUP BY l.id, l.name;

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

Select courier_id, AVG(DATEDIFF(deliver_date, payment_date)) as AverageDeliveryTime from courier GROUP BY courier_id ORDER BY AverageDeliveryTime DESC LIMIT 1;

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

Select l.id as LocationID, l.name as LocationName, SUM(p.amount) as TotalPayments

from location l JOIN payment p ON l.id = p.location_id GROUP BY l.id, l.name

HAVING SUM(p.amount) < 1000;

19. Calculate Total Payments per Location

Select l.id as LocationID, l.name as LocationName, SUM(p.amount) as TotalPayments

FROM location l JOIN payment p ON l.id = p.location_id GROUP BY l.id, l.name;

20. Retrieve couriers who have received payments totaling more than $1000 in a specific location (LocationID = X):

Select c.id as CourierID, c.sender_name, c.receiver_name, SUM(p.amount) as TotalPayments FROM courier c JOIN payment p ON c.id = p.courier_id

WHERE p.location_id = 1 GROUP BY c.id, c.sender_name, c.receiver_name 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.id as CourierID, c.sender_name, c.receiver_name, SUM(p.amount) as TotalPayments FROM courier c JOIN payment p ON c.id = p.courier_id

WHERE p.payment_date > '2023-09-01' GROUP BY c.id, c.sender_name, c.receiver_name 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 l.id as LocationID, l.name as LocationName, SUM(p.amount) as TotalPayments

FROM location l JOIN payment p ON l.id = p.location_id WHERE p.payment_date < '2023-09-01' GROUP BY l.id, l.name HAVING SUM(p.amount) > 5000;

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

23. Retrieve Payments with Courier Information

Select p.id as PaymentID, p.amount, p.payment_date, c.id as CourierID, c.sender_name, c.receiver_name FROM payment p JOIN courier c ON p.courier_id = c.id;

24. Retrieve Payments with Location Information

Select p.id as PaymentID, p.amount, p.payment_date, l.id as LocationID, l.name as LocationName FROM payment p JOIN location l ON p.location_id = l.id;

25. Retrieve Payments with Courier and Location Information

SELECT p.id as PaymentID, p.amount, p.payment_date, c.id as CourierID, c.sender_name, c.receiver_name, l.id as LocationID, l.name as LocationName

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.id as PaymentID, p.amount, p.payment_date, p.courier_id,

c.sender_name, c.receiver_name, c.weight, c.status, c.tracking_number, c.deliver_date FROM payment p JOIN courier c ON p.courier_id = c.id;

27. Total payments received for each courier

SELECT p.courier_id, c.sender_name, c.receiver_name, SUM(p.amount) as TotalPayments FROM payment p JOIN courier c ON p.courier_id = c.id

GROUP BY p.courier_id, c.sender_name, c.receiver_name;

28. List payments made on a specific date

```
SELECT * FROM payment WHERE payment_date = '2023-05-02';
```

29. Get Courier Information for Each Payment

```
SELECT p.id as PaymentID, p.amount, p.payment_date, p.courier_id,

c.sender_name, c.receiver_name, c.weight, c.status, c.tracking_number,
c.deliver_date FROM payment p JOIN courier c ON p.courier_id = c.id;
```

30. Get Payment Details with Location

```
SELECT p.id as PaymentID, p.amount, p.payment_date, p.location_id,

 l.name as LocationName, l.address as LocationAddress FROM payment p

JOIN location l ON p.location_id = l.id;
```

31. Calculating Total Payments for Each Courier

```
SELECT c.id as CourierID, c.sender_name, c.receiver_name, SUM(p.amount) as
TotalPayments FROM courier c JOIN payment p ON c.id = p.courier_id

GROUP BY c.id, c.sender_name, c.receiver_name;
```

32. List Payments Within a Date Range

```
SELECT * FROM payment WHERE payment_date BETWEEN ‘2023-04-01' AND '2024-05-19';
```

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

```
SELECT u.id as UserID, u.name as UserName, c.id as CourierID, c.sender_name,
c.receiver_name FROM user u LEFT JOIN orders o ON u.id = o.user_id

LEFT JOIN courier c ON o.courier_id = c.id UNION

SELECT u.id as UserID, u.name as UserName, c.id as CourierID, c.sender_name,
c.receiver_name FROM user u RIGHT JOIN orders o ON u.id = o.user_id

RIGHT JOIN courier c ON o.courier_id = c.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 CourierID, c.sender_name, c.receiver_name, cs.id as ServiceID,
cs.name as ServiceName FROM courier c LEFT JOIN courier_service cs ON c.id = cs.id
UNION SELECT c.id as CourierID, c.sender_name, c.receiver_name, cs.id as ServiceID,
cs.name as ServiceName FROM courier c RIGHT 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.id as EmployeeID, e.name as EmployeeName, p.id as PaymentID, p.amount

FROM employee e LEFT JOIN courier c ON e.id = c.employee_responsible_id

LEFT JOIN payment p ON c.id = p.courier_id UNION SELECT e.id as EmployeeID,
e.name as EmployeeName, p.id as PaymentID, p.amount FROM employee e

RIGHT JOIN courier c ON e.id = c.employee_responsible_id  RIGHT JOIN payment p
ON c.id = p.courier_id;
```

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

```
SELECT u.id as UserID, u.name as UserName, cs.id as ServiceID, cs.name as
ServiceName FROM user u CROSS JOIN courier_service cs;
```

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

```
SELECT e.id as EmployeeID, e.name as EmployeeName, l.id as LocationID, l.name as
LocationName FROM employee e CROSS JOIN location l;
```

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

```
SELECT id as CourierID, sender_name, sender_address, receiver_name,
receiver_address, weight, status, tracking_number, deliver_date FROM courier;
```

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

```
SELECT id as CourierID, receiver_name, receiver_address, sender_name,
sender_address, weight, status, tracking_number, deliver_date FROM courier;
```

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

```
SELECT c.id AS CourierID, c.sender_name, c.receiver_name, cs.id AS ServiceID,
cs.name AS ServiceName, cs.cost FROM courier c LEFT JOIN courier_service cs ON
c.service_id = cs.id;
```

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

```
SELECT e.id AS EmployeeID, e.name AS EmployeeName, COUNT(c.id) AS
NumberOfCouriersAssigned FROM employee e LEFT JOIN courier c ON e.id =
c.employee_id GROUP BY e.id, e.name;
```

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

```
Select l.id AS LocationID, l.name AS LocationName, SUM(p.amount) AS
TotalPaymentAmount FROM location l join payment p ON l.id = p.location_id

GROUP BY l.id, l.name;
```

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

```
Select sender_name, GROUP_CONCAT(id) AS CourierIDs, COUNT(id) AS TotalCouriers
from courier GROUP BY sender_name;
```

44. List all employees who share the same role.

```
SELECT role, GROUP_CONCAT(id) AS EmployeeIDs, GROUP_CONCAT(name) AS
EmployeeNames, COUNT(id) AS TotalEmployees FROM employee GROUP BY role;
```

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

```
Select loc.id AS LocationID, loc.name AS LocationName, SUM(p.amount) AS
TotalPayments FROM payment p JOIN courier c ON p.courier_id = c.id

join location loc ON c.location_id = loc.id  GROUP BY loc.id, loc.name;
```

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

```
Select sender_address, GROUP_CONCAT(id) AS CourierIDs, COUNT(id) as
TotalCouriers from courier GROUP BY sender_address;
```

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

```
Select e.id as EmployeeID, e.name as EmployeeName, COUNT(c.id) as
CouriersDelivered from employee e LEFT JOIN courier c on e.id =
c.delivered_by_employee_id  group by e.id, e.name;
```

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

```
Select c.id AS CourierID, cs.name as ServiceName, cs.cost as ServiceCost, p.amount
as PaymentAmount from courier c JOIN courier_service cs on c.service_id = cs.id

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