# Assignment-Courier Management System

Database Name : CMS

Tables used in Database CMS :

**Table Name: User**

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| UserID | int | NO | PRI | NULL | |
| Name | varchar(255) | YES | | NULL | |
| Email | varchar(255) | YES | UNI | NULL | |
| Password | varchar(255) | YES | | NULL | |
| ContactNumber | varchar(20) | YES | | NULL | |
| Address | text | YES | | NULL | |

**Table Name: Courier**

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| CourierID | int | NO | PRI | NULL | |
| SenderName | varchar(255) | YES | | NULL | |
| SenderAddress | text | YES | | NULL | |
| ReceiverName | varchar(255) | YES | | NULL | |
| ReceiverAddress | text | YES | | NULL | |
| Weight | decimal(5,2) | YES | | NULL | |
| Status | varchar(50) | YES | | NULL | |
| TrackingNumber | varchar(20) | YES | UNI | NULL | |
| DeliveryDate | date | YES | | NULL | |

**Table Name: CourierServices**

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| ServiceID | int | NO | PRI | NULL | |
| ServiceName | varchar(100) | YES | | NULL | |
| Cost | decimal(8,2) | YES | | NULL | |

**Table Name: Employee**

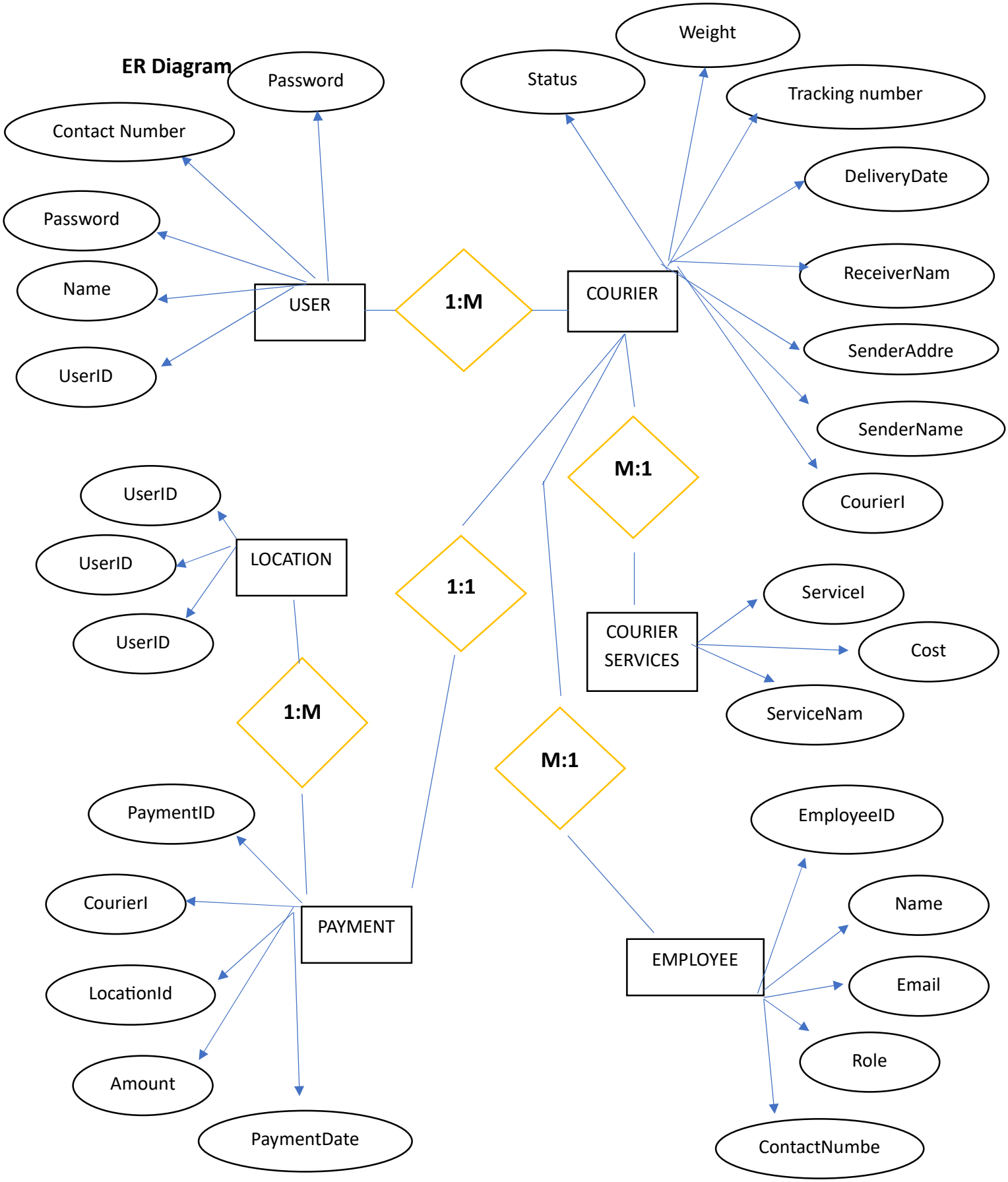
| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| EmployeeID | int | NO | PRI | NULL | |
| Name | varchar(255) | YES | | NULL | |
| Email | varchar(255) | YES | UNI | NULL | |
| ContactNumber | varchar(20) | YES | | NULL | |
| Role | varchar(50) | YES | | NULL | |
| Salary | decimal(10,2) | YES | | NULL | |

**Table Name: Location**

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| LocationID | int | NO | PRI | NULL | |
| LocationName | varchar(100) | YES | | NULL | |
| Address | text | YES | | NULL | |

**Table Name: Payment**

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| PaymentID | int | NO | PRI | NULL | |
| CourierID | int | YES | MUL | NULL | |
| LocationId | int | YES | MUL | NULL | |
| Amount | decimal(10,2) | YES | | NULL | |
| PaymentDate | date | YES | | NULL | |


ER Diagram
Password
Contact Number
Password
Name
UserID
USER
1:M
COURIER
Weight
Status
Tracking number
DeliveryDate
ReceiverNam
SenderAddre
SenderName
CourierI
M:1
UserID
UserID
UserID
LOCATION
1:1
1:M
COURIER SERVICES
ServiceI
Cost
ServiceNam
M:1
PaymentID
CourierI
LocationId
Amount
PaymentDate
PAYMENT
EMPLOYEE
EmployeeID
Name
Email
Role
ContactNumbe

## Task 2: Select, Where

## Solve the following queries in the Schema that you have created above

1. **List all customers**

   Select * from user;

```
mysql> select * from user;
+--------+--------------+---------------------------+------------+---------------+---------------------------+
| UserID | Name         | Email                     | Password   | ContactNumber | Address                   |
+--------+--------------+---------------------------+------------+---------------+---------------------------+
|      1 | Rohan Sharma | rohan.sharma@example.com  | Rohan@123  | 9876543210    | Andheri, Mumbai           |
|      2 | Priya Verma  | priya.verma@example.com   | Priya@456  | 8765432109    | Rajouri Garden, Delhi     |
|      3 | Amit Kumar   | amit.kumar@example.com    | Amit@789   | 7654321098    | Banjara Hills, Hyderabad  |
|      4 | Neha Singh   | neha.singh@example.com    | Neha@321   | 6543210987    | Sector 62, Noida          |
|      5 | Vikram Patil | vikram.patil@example.com  | Vikram@654 | 5432109876    | Shivajinagar, Pune        |
|      6 | Ananya Iyer  | ananya.iyer@example.com   | Ananya@987 | 4321098765    | Indira Nagar, Bangalore   |
|      7 | Kunal Joshi  | kunal.joshi@example.com   | Kunal@111  | 3210987654    | Alkapuri, Vadodara        |
|      8 | Meera Nair   | meera.nair@example.com    | Meera@222  | 2109876543    | Kakkanad, Kochi           |
+--------+--------------+---------------------------+------------+---------------+---------------------------+
```

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

   Select * from courier where sendername = "amit kumar";

```
mysql> select * from courier where sendername = "amit kumar";
+-----------+------------+-----------------------+--------------+-----------------+--------+-----------+----------------+--------------+
| CourierID | SenderName | SenderAddress         | ReceiverName | ReceiverAddress | Weight | Status    | TrackingNumber | DeliveryDate |
+-----------+------------+-----------------------+--------------+-----------------+--------+-----------+----------------+--------------+
|         8 | Amit Kumar | Banjara Hills, Hyderabad | Neha Singh   | Sector 62, Noida |   3.75 | Delivered | TRK008         | 2025-03-19   |
+-----------+------------+-----------------------+--------------+-----------------+--------+-----------+----------------+--------------+
```

3. **List all couriers**

   Select * from courier;

```
mysql> select * from courier;
+-----------+--------------+---------------------------+--------------+---------------------------+--------+------------+----------------+--------------+
| CourierID | SenderName   | SenderAddress             | ReceiverName | ReceiverAddress           | Weight | Status     | TrackingNumber | DeliveryDate |
+-----------+--------------+---------------------------+--------------+---------------------------+--------+------------+----------------+--------------+
|         1 | Rohan Sharma | Andheri, Mumbai           | Priya Verma  | Rajouri Garden, Delhi     |   2.50 | In Transit | TRK001         | 2025-03-25   |
|         2 | Neha Singh   | Sector 62, Noida          | Amit Kumar   | Banjara Hills, Hyderabad  |   5.00 | Delivered  | TRK002         | 2025-03-20   |
|         3 | Vikram Patil | Shivajinagar, Pune        | Ananya Iyer  | Indira Nagar, Bangalore   |   1.75 | Pending    | TRK003         | 2025-03-28   |
|         4 | Kunal Joshi  | Alkapuri, Vadodara        | Meera Nair   | Kakkanad, Kochi           |   3.25 | In Transit | TRK004         | 2025-03-22   |
|         5 | Meera Nair   | Kakkanad, Kochi           | Rohan Sharma | Andheri, Mumbai           |   4.50 | Delivered  | TRK005         | 2025-03-18   |
|         6 | Priya Verma  | Rajouri Garden, Delhi     | Vikram Patil | Shivajinagar, Pune        |   2.00 | Pending    | TRK006         | 2025-03-27   |
|         7 | Ananya Iyer  | Indira Nagar, Bangalore   | Kunal Joshi  | Alkapuri, Vadodara        |   6.00 | In Transit | TRK007         | 2025-03-23   |
|         8 | Amit Kumar   | Banjara Hills, Hyderabad  | Neha Singh   | Sector 62, Noida          |   3.75 | Delivered  | TRK008         | 2025-03-19   |
+-----------+--------------+---------------------------+--------------+---------------------------+--------+------------+----------------+--------------+
```

4. **List all packages for a specific order ***

   select * from courier where trackingnumber = "TRK003";

```
mysql> select * from courier where trackingnumber = "TRK003";
+-----------+--------------+--------------------+--------------+---------------------------+--------+---------+----------------+--------------+
| CourierID | SenderName   | SenderAddress      | ReceiverName | ReceiverAddress           | Weight | Status  | TrackingNumber | DeliveryDate |
+-----------+--------------+--------------------+--------------+---------------------------+--------+---------+----------------+--------------+
|         3 | Vikram Patil | Shivajinagar, Pune | Ananya Iyer  | Indira Nagar, Bangalore   |   1.75 | Pending | TRK003         | 2025-03-28   |
+-----------+--------------+--------------------+--------------+---------------------------+--------+---------+----------------+--------------+
```

5. **List all deliveries for a specific courier***

   select * from courier where courierid = 4;

```
mysql> select * from courier where courierid = 4;
+-----------+-------------+--------------------+--------------+-----------------+--------+------------+----------------+--------------+
| CourierID | SenderName  | SenderAddress      | ReceiverName | ReceiverAddress | Weight | Status     | TrackingNumber | DeliveryDate |
+-----------+-------------+--------------------+--------------+-----------------+--------+------------+----------------+--------------+
|         4 | Kunal Joshi | Alkapuri, Vadodara | Meera Nair   | Kakkanad, Kochi |   3.25 | In Transit | TRK004         | 2025-03-22   |
+-----------+-------------+--------------------+--------------+-----------------+--------+------------+----------------+--------------+
```

### 6. List all undelivered packages

select * from courier where status != "Delivered" ;

```
mysql> select * from courier where status != "Delivered" ;
+-----------+-------------+-----------------------+-------------+-----------------------+--------+-----------+----------------+--------------+
| CourierID | SenderName  | SenderAddress         | ReceiverName | ReceiverAddress       | Weight | Status    | TrackingNumber | DeliveryDate |
+-----------+-------------+-----------------------+-------------+-----------------------+--------+-----------+----------------+--------------+
|         1 | Rohan Sharma | Andheri, Mumbai       | Priya Verma  | Rajouri Garden, Delhi |   2.50 | In Transit | TRK001         | 2025-03-25   |
|         3 | Vikram Patil | Shivajinagar, Pune    | Ananya Iyer  | Indira Nagar, Bangalore |   1.75 | Pending    | TRK003         | 2025-03-28   |
|         4 | Kunal Joshi  | Alkapuri, Vadodara    | Meera Nair   | Kakkanad, Kochi       |   3.25 | In Transit | TRK004         | 2025-03-22   |
|         6 | Priya Verma  | Rajouri Garden, Delhi | Vikram Patil | Shivajinagar, Pune    |   2.00 | Pending    | TRK006         | 2025-03-27   |
|         7 | Ananya Iyer  | Indira Nagar, Bangalore | Kunal Joshi  | Alkapuri, Vadodara    |   6.00 | In Transit | TRK007         | 2025-03-23   |
+-----------+-------------+-----------------------+-------------+-----------------------+--------+-----------+----------------+--------------+
```

### 7. List all packages that are scheduled for delivery today

select * from courier where deliverydate = "2025-03-20";

```
mysql> select * from courier where deliverydate = "2025-03-20";
+-----------+-------------+----------------+--------------+-----------------------+--------+-----------+----------------+--------------+
| CourierID | SenderName  | SenderAddress  | ReceiverName | ReceiverAddress       | Weight | Status    | TrackingNumber | DeliveryDate |
+-----------+-------------+----------------+--------------+-----------------------+--------+-----------+----------------+--------------+
|         2 | Neha Singh  | Sector 62, Noida | Amit Kumar   | Banjara Hills, Hyderabad |   5.00 | Delivered | TRK002         | 2025-03-20   |
+-----------+-------------+----------------+--------------+-----------------------+--------+-----------+----------------+--------------+
```

### 8. List all packages with a specific status*

select * from courier where status="In Transit";

```
mysql> select * from courier where status="In Transit";
+-----------+-------------+-----------------------+-------------+-----------------------+--------+-----------+----------------+--------------+
| CourierID | SenderName  | SenderAddress         | ReceiverName | ReceiverAddress       | Weight | Status    | TrackingNumber | DeliveryDate |
+-----------+-------------+-----------------------+-------------+-----------------------+--------+-----------+----------------+--------------+
|         1 | Rohan Sharma | Andheri, Mumbai       | Priya Verma  | Rajouri Garden, Delhi |   2.50 | In Transit | TRK001         | 2025-03-25   |
|         4 | Kunal Joshi  | Alkapuri, Vadodara    | Meera Nair   | Kakkanad, Kochi       |   3.25 | In Transit | TRK004         | 2025-03-22   |
|         7 | Ananya Iyer  | Indira Nagar, Bangalore | Kunal Joshi  | Alkapuri, Vadodara    |   6.00 | In Transit | TRK007         | 2025-03-23   |
+-----------+-------------+-----------------------+-------------+-----------------------+--------+-----------+----------------+--------------+
```

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

select sendername, count(*) as totalpackage from courier group by sendername;

```
mysql> select sendername,count(*) as totalpackage from courier group by sendername;
+--------------+--------------+
| sendername   | totalpackage |
+--------------+--------------+
| Rohan Sharma |            1 |
| Neha Singh   |            1 |
| Vikram Patil |            1 |
| Kunal Joshi  |            1 |
| Meera Nair   |            1 |
| Priya Verma  |            1 |
| Ananya Iyer  |            1 |
| Amit Kumar   |            1 |
+--------------+--------------+
```

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

select sendername, avg(datediff(deliverydate,now())) from courier group by sendername;

```
mysql> select sendername, avg(datediff(deliverydate,now())) from courier group by sendername;
+--------------+-----------------------------------+
| sendername   | avg(datediff(deliverydate,now())) |
+--------------+-----------------------------------+
| Rohan Sharma |                            5.0000 |
| Neha Singh   |                            0.0000 |
| Vikram Patil |                            8.0000 |
| Kunal Joshi  |                            2.0000 |
| Meera Nair   |                           -2.0000 |
| Priya Verma  |                            7.0000 |
| Ananya Iyer  |                            3.0000 |
| Amit Kumar   |                           -1.0000 |
+--------------+-----------------------------------+
```

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

select * from courier where weight between 2 and 6;

```
mysql> select * from courier where weight between 2 and 6;
+-----------+-------------+--------------------------+--------------+--------------------------+--------+------------+----------------+--------------+
| CourierID | SenderName  | SenderAddress            | ReceiverName | ReceiverAddress          | Weight | Status     | TrackingNumber | DeliveryDate |
+-----------+-------------+--------------------------+--------------+--------------------------+--------+------------+----------------+--------------+
|         1 | Rohan Sharma | Andheri, Mumbai          | Priya Verma  | Rajouri Garden, Delhi    |   2.50 | In Transit | TRK001         | 2025-03-25   |
|         2 | Neha Singh   | Sector 62, Noida         | Amit Kumar   | Banjara Hills, Hyderabad |   5.00 | Delivered  | TRK002         | 2025-03-20   |
|         4 | Kunal Joshi  | Alkapuri, Vadodara       | Meera Nair   | Kakkanad, Kochi          |   3.25 | In Transit | TRK004         | 2025-03-22   |
|         5 | Meera Nair   | Kakkanad, Kochi          | Rohan Sharma | Andheri, Mumbai          |   4.50 | Delivered  | TRK005         | 2025-03-18   |
|         6 | Priya Verma  | Rajouri Garden, Delhi    | Vikram Patil | Shivajinagar, Pune       |   2.00 | Pending    | TRK006         | 2025-03-27   |
|         7 | Ananya Iyer  | Indira Nagar, Bangalore  | Kunal Joshi  | Alkapuri, Vadodara       |   6.00 | In Transit | TRK007         | 2025-03-23   |
|         8 | Amit Kumar   | Banjara Hills, Hyderabad | Neha Singh   | Sector 62, Noida         |   3.75 | Delivered  | TRK008         | 2025-03-19   |
+-----------+-------------+--------------------------+--------------+--------------------------+--------+------------+----------------+--------------+
```

**12. Retrieve employees whose names contain ‘Rajesh’**

select * from employee where name like '%Rajesh%';

```
mysql> select * from employee where name like '%Rajesh%';
+-------------+--------------+---------------------------+---------------+----------+-----------+
| EmployeeID | Name         | Email                     | ContactNumber | Role     | Salary    |
+-------------+--------------+---------------------------+---------------+----------+-----------+
|           1 | Rajesh Mehta | rajesh.mehta@example.com | 9876543211    | Manager | 90000.00 |
+-------------+--------------+---------------------------+---------------+----------+-----------+
```

**13. Retrieve all courier records with payments greater than $50.**

Select c.courierid,c.senderaddress,c.receivername,c.ReceiverAddress,c.weight,c.status,c.TrackingNumber,c. DeliveryDate from courier c join payment p on c.courierid = p.courierid where amount > 500;

```
mysql> select c.courierid,c.senderaddress,c.receivername,c.ReceiverAddress,c.weight,c.status,c.TrackingNumber,c. DeliveryDate from courie
+-----------+--------------------------+--------------+--------------------------+--------+------------+----------------+--------------+
| courierid | senderaddress            | receivername | ReceiverAddress          | weight | status     | TrackingNumber | DeliveryDate |
+-----------+--------------------------+--------------+--------------------------+--------+------------+----------------+--------------+
|         2 | Sector 62, Noida         | Amit Kumar   | Banjara Hills, Hyderabad |   5.00 | Delivered  | TRK002         | 2025-03-20   |
|         3 | Shivajinagar, Pune       | Ananya Iyer  | Indira Nagar, Bangalore  |   1.75 | Pending    | TRK003         | 2025-03-28   |
|         4 | Alkapuri, Vadodara       | Meera Nair   | Kakkanad, Kochi          |   3.25 | In Transit | TRK004         | 2025-03-22   |
|         5 | Kakkanad, Kochi          | Rohan Sharma | Andheri, Mumbai          |   4.50 | Delivered  | TRK005         | 2025-03-18   |
|         6 | Rajouri Garden, Delhi    | Vikram Patil | Shivajinagar, Pune       |   2.00 | Pending    | TRK006         | 2025-03-27   |
|         8 | Banjara Hills, Hyderabad | Neha Singh   | Sector 62, Noida         |   3.75 | Delivered  | TRK008         | 2025-03-19   |
+-----------+--------------------------+--------------+--------------------------+--------+------------+----------------+--------------+
```

# Task 2

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

select e.employeeid, e.name, count(c.courierid) as totalcourierhandel from employee e join courier c on e.employeeid = c.employeeid group by courierid**;**

| employeeid | name | totalcourierhandel |
|---|---|---|
| 1 | Rajesh Mehta | 1 |
| 2 | Sunita Rao | 1 |
| 3 | Arjun Kapoor | 1 |
| 4 | Kiran Das | 1 |
| 5 | Ritika Gupta | 1 |
| 6 | Manish Reddy | 1 |
| 7 | Sneha Choudhury | 1 |
| 8 | Aakash Sen | 1 |

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

select locationid, sum(amount) as totalrevenu from payment group by locationid;

| locationid | totalrevenu |
|---|---|
| 1 | 500.00 |
| 2 | 1200.00 |
| 3 | 2500.00 |
| 4 | 3500.00 |
| 5 | 1800.00 |
| 6 | 8000.00 |
| 7 | 200.00 |
| 8 | 12000.00 |

8 rows in set (0.00 sec)

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

select SenderAddress, count(*) as totalcourierdelivered from courier where status="Delivered" group by SenderAddress;

| SenderAddress | totalcourierdelivered |
|---|---|
| Sector 62, Noida | 1 |
| Kakkanad, Kochi | 1 |
| Banjara Hills, Hyderabad | 1 |

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

select sendername, avg(datediff(DeliveryDate,now())) as highestavgdeliverytime from courier group by SenderName order by highestavgdeliverytime desc limit 1;

| sendername | highestavgdeliverytime |
|---|---|
| Vikram Patil | 8.0000 |

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

select l.locationid,l.locationname, l.address, sum(p.amount) as pay from location l join payment p on l.locationid = p.locationid group by locationid having pay<2000;

| locationid | locationname | address | pay |
|---|---|---|---|
| 1 | Mumbai Hub | Andheri, Mumbai | 500.00 |
| 2 | Delhi Hub | Rajouri Garden, Delhi | 1200.00 |
| 5 | Pune Hub | Shivajinagar, Pune | 1800.00 |
| 7 | Vadodara Hub | Alkapuri, Vadodara | 200.00 |

**19. Calculate Total Payments per Location**

select l.locationid, l.locationname, l.address, sum(p.amount) as totalpayment from location l join payment p on l.locationid=p.locationid group by locationid order by totalpayment;

| locationid | locationname | address | totalpayment |
|---|---|---|---|
| 7 | Vadodara Hub | Alkapuri, Vadodara | 200.00 |
| 1 | Mumbai Hub | Andheri, Mumbai | 500.00 |
| 2 | Delhi Hub | Rajouri Garden, Delhi | 1200.00 |
| 5 | Pune Hub | Shivajinagar, Pune | 1800.00 |
| 3 | Hyderabad Hub | Banjara Hills, Hyderabad | 2500.00 |
| 4 | Noida Hub | Sector 62, Noida | 3500.00 |
| 6 | Bangalore Hub | Indira Nagar, Bangalore | 8000.00 |
| 8 | Kochi Hub | Kakkanad, Kochi | 12000.00 |

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

select c.courierid, sum(p.amount) as totalpay from courier c join payment p on c.courierid = p.courierid where p.locationid = 4 group by c.courierid having totalpay>1000;

| courierid | totalpay |
|---|---|
| 4 | 3500.00 |

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

select c.courierid,c.deliverydate, sum(p.amount) as totalpay from courier c join payment p on c.courierid = p.courierid where  c.DeliveryDate > "2025-03-20" group by c.courierid having totalpay > 1000;

| courierid | deliverydate | totalpay |
|---|---|---|
| 3 | 2025-03-28 | 2500.00 |
| 4 | 2025-03-22 | 3500.00 |
| 6 | 2025-03-27 | 8000.00 |

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

select l.locationid,p.paymentdate,l.locationname,sum(p.amount) as totalpay from location l join payment p on l.locationid=p.locationid where p.paymentdate > "2025-03-20" group by l.locationid,p.paymentdate having totalpay > 5000;

| locationid | paymentdate | locationname | totalpay |
|---|---|---|---|
| 6 | 2025-03-23 | Bangalore Hub | 8000.00 |
| 8 | 2025-03-24 | Kochi Hub | 12000.00 |

**23. Retrieve Payments with Courier Information**

select payment.*,courier.* from payment inner join courier on payment.CourierID = courier.CourierID;

```
mysql> select payment.*,courier.* from payment inner join courier on payment.CourierID = courier.CourierID;
```

| PaymentID | CourierID | LocationId | Amount | PaymentDate | CourierID | SenderName | SenderAddress | ReceiverName | ReceiverAddress | Weight | Status | TrackingNumber | DeliveryDate | EmployeeID |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 500.00 | 2025-03-20 | 1 | Rohan Sharma | Andheri, Mumbai | Priya Verma | Rajouri Garden, Delhi | 2.50 | In Transit | TRK001 | 2025-03-25 | 1 |
| 2 | 2 | 2 | 1200.00 | 2025-03-18 | 2 | Neha Singh | Sector 62, Noida | Amit Kumar | Banjara Hills, Hyderabad | 5.00 | Delivered | TRK002 | 2025-03-20 | 2 |
| 3 | 3 | 3 | 2500.00 | 2025-03-22 | 3 | Vikram Patil | Shivajinagar, Pune | Ananya Iyer | Indira Nagar, Bangalore | 1.75 | Pending | TRK003 | 2025-03-28 | 3 |
| 4 | 4 | 4 | 3500.00 | 2025-03-19 | 4 | Kunal Joshi | Alkapuri, Vadodara | Meera Nair | Kakkanad, Kochi | 3.25 | In Transit | TRK004 | 2025-03-22 | 4 |
| 5 | 5 | 5 | 1800.00 | 2025-03-25 | 5 | Meera Nair | Kakkanad, Kochi | Rohan Sharma | Andheri, Mumbai | 4.50 | Delivered | TRK005 | 2025-03-18 | 5 |
| 6 | 6 | 6 | 8000.00 | 2025-03-23 | 6 | Priya Verma | Rajouri Garden, Delhi | Vikram Patil | Shivajinagar, Pune | 2.00 | Pending | TRK006 | 2025-03-27 | 6 |
| 7 | 7 | 7 | 200.00 | 2025-03-21 | 7 | Ananya Iyer | Indira Nagar, Bangalore | Kunal Joshi | Alkapuri, Vadodara | 6.00 | In Transit | TRK007 | 2025-03-23 | 7 |
| 8 | 8 | 8 | 12000.00 | 2025-03-24 | 8 | Amit Kumar | Banjara Hills, Hyderabad | Neha Singh | Sector 62, Noida | 3.75 | Delivered | TRK008 | 2025-03-19 | 8 |

**24. Retrieve Payments with Location Information**

select payment.*,location.* from payment inner join location on location.locationid = payment.locationid;

```
mysql> select payment.*,location.* from payment inner join location on location.locationid = payment.locationid;
+-----------+-----------+------------+---------+-------------+------------+---------------+--------------------------+
| PaymentID | CourierID | LocationId | Amount  | PaymentDate | LocationID | LocationName  | Address                  |
+-----------+-----------+------------+---------+-------------+------------+---------------+--------------------------+
|         1 |         1 |          1 |  500.00 | 2025-03-20  |          1 | Mumbai Hub    | Andheri, Mumbai          |
|         2 |         2 |          2 | 1200.00 | 2025-03-18  |          2 | Delhi Hub     | Rajouri Garden, Delhi    |
|         3 |         3 |          3 | 2500.00 | 2025-03-22  |          3 | Hyderabad Hub | Banjara Hills, Hyderabad |
|         4 |         4 |          4 | 3500.00 | 2025-03-19  |          4 | Noida Hub     | Sector 62, Noida         |
|         5 |         5 |          5 | 1800.00 | 2025-03-25  |          5 | Pune Hub      | Shivajinagar, Pune       |
|         6 |         6 |          6 | 8000.00 | 2025-03-23  |          6 | Bangalore Hub | Indira Nagar, Bangalore  |
|         7 |         7 |          7 |  200.00 | 2025-03-21  |          7 | Vadodara Hub  | Alkapuri, Vadodara       |
|         8 |         8 |          8 | 12000.00| 2025-03-24  |          8 | Kochi Hub     | Kakkanad, Kochi          |
+-----------+-----------+------------+---------+-------------+------------+---------------+--------------------------+
```

**25. Retrieve Payments with Courier and Location Information**

select payment.*,location.*,courier.* from payment inner join courier on payment.courierid = courier.courierid inner join location on payment.locationid = location.locationid;

```
+-----------+-----------+------------+---------+-------------+------------+---------------+--------------------------+-----------+---------------+---------------------------+
| PaymentID | CourierID | LocationId | Amount  | PaymentDate | LocationID | LocationName  | Address                  | CourierID | SenderName    | SenderAddress             |
+-----------+-----------+------------+---------+-------------+------------+---------------+--------------------------+-----------+---------------+---------------------------+
|         1 |         1 |          1 |  500.00 | 2025-03-20  |          1 | Mumbai Hub    | Andheri, Mumbai          |         1 | Rohan Sharma  | Andheri, Mumbai           |
|         2 |         2 |          2 | 1200.00 | 2025-03-18  |          2 | Delhi Hub     | Rajouri Garden, Delhi    |         2 | Neha Singh    | Sector 62, Noida          |
|         3 |         3 |          3 | 2500.00 | 2025-03-22  |          3 | Hyderabad Hub | Banjara Hills, Hyderabad |         3 | Vikram Patil  | Shivajinagar, Pune        |
|         4 |         4 |          4 | 3500.00 | 2025-03-19  |          4 | Noida Hub     | Sector 62, Noida         |         4 | Kunal Joshi   | Alkapuri, Vadodara        |
|         5 |         5 |          5 | 1800.00 | 2025-03-25  |          5 | Pune Hub      | Shivajinagar, Pune       |         5 | Meera Nair    | Kakkanad, Kochi           |
|         6 |         6 |          6 | 8000.00 | 2025-03-23  |          6 | Bangalore Hub | Indira Nagar, Bangalore  |         6 | Priya Verma   | Rajouri Garden, Delhi     |
|         7 |         7 |          7 |  200.00 | 2025-03-21  |          7 | Vadodara Hub  | Alkapuri, Vadodara       |         7 | Ananya Iyer   | Indira Nagar, Bangalore   |
|         8 |         8 |          8 | 12000.00| 2025-03-24  |          8 | Kochi Hub     | Kakkanad, Kochi          |         8 | Amit Kumar    | Banjara Hills, Hyderabad  |
+-----------+-----------+------------+---------+-------------+------------+---------------+--------------------------+-----------+---------------+---------------------------+
8 rows in set (0.00 sec)
```

**26. List all payments with courier details**

select payment.*, courier.* from payment left outer join courier on payment.courierid = courier.courierid;

```
mysql> select payment.*, courier.* from payment left outer join courier on payment.courierid = courier.courierid;
+-----------+-----------+------------+---------+-------------+-----------+---------------+--------------------------+---------------+--------------------------+--------+-----------+----------------+------+
| PaymentID | CourierID | LocationId | Amount  | PaymentDate | CourierID | SenderName    | SenderAddress            | ReceiverName  | ReceiverAddress          | Weight | Status    | TrackingNumber | Deli |
+-----------+-----------+------------+---------+-------------+-----------+---------------+--------------------------+---------------+--------------------------+--------+-----------+----------------+------+
|         1 |         1 |          1 |  500.00 | 2025-03-20  |         1 | Rohan Sharma  | Andheri, Mumbai          | Priya Verma   | Rajouri Garden, Delhi    |   2.50 | In Transit | TRK001         | 2025
|         2 |         2 |          2 | 1200.00 | 2025-03-18  |         2 | Neha Singh    | Sector 62, Noida         | Amit Kumar    | Banjara Hills, Hyderabad |   5.00 | Delivered  | TRK002         | 2025
|         3 |         3 |          3 | 2500.00 | 2025-03-22  |         3 | Vikram Patil  | Shivajinagar, Pune       | Ananya Iyer   | Indira Nagar, Bangalore  |   1.75 | Pending    | TRK003         | 2025
|         4 |         4 |          4 | 3500.00 | 2025-03-19  |         4 | Kunal Joshi   | Alkapuri, Vadodara       | Meera Nair    | Kakkanad, Kochi          |   3.25 | In Transit | TRK004         |
|         5 |         5 |          5 | 1800.00 | 2025-03-25  |         5 | Meera Nair    | Kakkanad, Kochi          | Rohan Sharma  | Andheri, Mumbai          |   4.50 | Delivered  | TRK005         | 2025
|         6 |         6 |          6 | 8000.00 | 2025-03-23  |         6 | Priya Verma   | Rajouri Garden, Delhi    | Vikram Patil  | Shivajinagar, Pune       |   2.00 | Pending    | TRK006         | 2025
|         7 |         7 |          7 |  200.00 | 2025-03-21  |         7 | Ananya Iyer   | Indira Nagar, Bangalore  | Kunal Joshi   | Alkapuri, Vadodara       |   6.00 | In Transit | TRK007         | 2025
|         8 |         8 |          8 | 12000.00| 2025-03-24  |         8 | Amit Kumar    | Banjara Hills, Hyderabad | Neha Singh    | Sector 62, Noida         |   3.75 | Delivered  | TRK008         | 2025
+-----------+-----------+------------+---------+-------------+-----------+---------------+--------------------------+---------------+--------------------------+--------+-----------+----------------+------+
8 rows in set (0.00 sec)
```

**27. Total payments received for each courier**

select c.courierid, c.sendername,sum(p.amount) as totalpayment from courier c inner join payment p on p.courierid = c.courierid group by c.courierid,c.sendername;

```
+-----------+---------------+--------------+
| courierid | sendername    | totalpayment |
+-----------+---------------+--------------+
|         1 | Rohan Sharma  |       500.00 |
|         2 | Neha Singh    |      1200.00 |
|         3 | Vikram Patil  |      2500.00 |
|         4 | Kunal Joshi   |      3500.00 |
|         5 | Meera Nair    |      1800.00 |
|         6 | Priya Verma   |      8000.00 |
|         7 | Ananya Iyer   |       200.00 |
|         8 | Amit Kumar    |     12000.00 |
+-----------+---------------+--------------+
```

**28. List payments made on a specific date**

mysql> select * from payment where paymentdate  = "2025-03-18";

```
mysql> select * from payment where paymentdate  = "2025-03-18"
+-----------+-----------+------------+---------+-------------+
| PaymentID | CourierID | LocationId | Amount  | PaymentDate |
+-----------+-----------+------------+---------+-------------+
|         2 |         2 |          2 | 1200.00 | 2025-03-18  |
+-----------+-----------+------------+---------+-------------+
```

**29. Get Courier Information for Each Payment**

select payment.*,courier.* from payment left outer join courier on courier.CourierID = payment.CourierID;

| PaymentID | CourierID | LocationId | Amount | PaymentDate | CourierID | SenderName | SenderAddress | ReceiverName |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 500.00 | 2025-03-20 | 1 | Rohan Sharma | Andheri, Mumbai | Priya Verma |
| 2 | 2 | 2 | 1200.00 | 2025-03-18 | 2 | Neha Singh | Sector 62, Noida | Amit Kumar |
| 3 | 3 | 3 | 2500.00 | 2025-03-22 | 3 | Vikram Patil | Shivajinagar, Pune | Ananya Iyer |
| 4 | 4 | 4 | 3500.00 | 2025-03-19 | 4 | Kunal Joshi | Alkapuri, Vadodara | Meera Nair |
| 5 | 5 | 5 | 1800.00 | 2025-03-25 | 5 | Meera Nair | Kakkanad, Kochi | Rohan Sharma |
| 6 | 6 | 6 | 8000.00 | 2025-03-23 | 6 | Priya Verma | Rajouri Garden, Delhi | Vikram Patil |
| 7 | 7 | 7 | 200.00 | 2025-03-21 | 7 | Ananya Iyer | Indira Nagar, Bangalore | Kunal Joshi |
| 8 | 8 | 8 | 12000.00 | 2025-03-24 | 8 | Amit Kumar | Banjara Hills, Hyderabad | Neha Singh |

**30. Get Payment Details with Location**

select payment.* , location.* from payment left outer join location on payment.locationid = location.locationid;

| PaymentID | CourierID | LocationId | Amount | PaymentDate | LocationID | LocationName | Address |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 500.00 | 2025-03-20 | 1 | Mumbai Hub | Andheri, Mumbai |
| 2 | 2 | 2 | 1200.00 | 2025-03-18 | 2 | Delhi Hub | Rajouri Garden, Delhi |
| 3 | 3 | 3 | 2500.00 | 2025-03-22 | 3 | Hyderabad Hub | Banjara Hills, Hyderabad |
| 4 | 4 | 4 | 3500.00 | 2025-03-19 | 4 | Noida Hub | Sector 62, Noida |
| 5 | 5 | 5 | 1800.00 | 2025-03-25 | 5 | Pune Hub | Shivajinagar, Pune |
| 6 | 6 | 6 | 8000.00 | 2025-03-23 | 6 | Bangalore Hub | Indira Nagar, Bangalore |
| 7 | 7 | 7 | 200.00 | 2025-03-21 | 7 | Vadodara Hub | Alkapuri, Vadodara |
| 8 | 8 | 8 | 12000.00 | 2025-03-24 | 8 | Kochi Hub | Kakkanad, Kochi |

**31. Calculating Total Payments for Each Courier**

mysql> select c.courierid,c.sendername,sum(p.amount) as totalpayment from payment p inner join courier c on p.courierid=c.courierid group by c.courierid,c.sendername;

| courierid | sendername | totalpayment |
|---|---|---|
| 1 | Rohan Sharma | 500.00 |
| 2 | Neha Singh | 1200.00 |
| 3 | Vikram Patil | 2500.00 |
| 4 | Kunal Joshi | 3500.00 |
| 5 | Meera Nair | 1800.00 |
| 6 | Priya Verma | 8000.00 |
| 7 | Ananya Iyer | 200.00 |
| 8 | Amit Kumar | 12000.00 |

**32. List Payments Within a Date Range**

select * from payment where paymentdate between " 2025-03-18" and "2025-03-24";

| PaymentID | CourierID | LocationId | Amount | PaymentDate |
|---|---|---|---|---|
| 1 | 1 | 1 | 500.00 | 2025-03-20 |
| 2 | 2 | 2 | 1200.00 | 2025-03-18 |
| 3 | 3 | 3 | 2500.00 | 2025-03-22 |
| 4 | 4 | 4 | 3500.00 | 2025-03-19 |
| 6 | 6 | 6 | 8000.00 | 2025-03-23 |
| 7 | 7 | 7 | 200.00 | 2025-03-21 |
| 8 | 8 | 8 | 12000.00 | 2025-03-24 |

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

mysql> select user.*,courier.* from courier left outer join user on user.name = courier.sendername

union

select user.*,courier.* from courier right outer join user on user.name = courier.sendername;

| UserID | Name | Email | Password | ContactNumber | Address | CourierID | SenderName | SenderAddress | ReceiverName | ReceiverAddress | Weight | Status | TrackingNumber | DeliveryDate | EmployeeI |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Rohan Sharma | rohan.sharma@example.com | Rohan@123 | 9876543210 | Andheri, Mumbai | 1 | Rohan Sharma | Andheri, Mumbai | Priya Verma | Rajouri Garden, Delhi | 2.50 | In Transit | TRK001 | 2025-03-25 | |
| 4 | Neha Singh | neha.singh@example.com | Neha@321 | 6543210987 | Sector 62, Noida | 2 | Neha Singh | Sector 62, Noida | Amit Kumar | Banjara Hills, Hyderabad | 5.00 | Delivered | TRK002 | 2025-03-20 | |
| 5 | Vikram Patil | vikram.patil@example.com | Vikram@654 | 5432109876 | Shivajinagar, Pune | 3 | Vikram Patil | Shivajinagar, Pune | Ananya Iyer | Indira Nagar, Bangalore | 1.75 | Pending | TRK003 | 2025-03-28 | |
| 7 | Kunal Joshi | kunal.joshi@example.com | Kunal@111 | 3210987654 | Alkapuri, Vadodara | 4 | Kunal Joshi | Alkapuri, Vadodara | Meera Nair | Kakkanad, Kochi | 3.25 | In Transit | TRK004 | 2025-03-22 | |
| 8 | Meera Nair | meera.nair@example.com | Meera@222 | 2109876543 | Kakkanad, Kochi | 5 | Meera Nair | Kakkanad, Kochi | Rohan Sharma | Andheri, Mumbai | 4.50 | Delivered | TRK005 | 2025-03-18 | |
| 2 | Priya Verma | priya.verma@example.com | Priya@456 | 8765432109 | Rajouri Garden, Delhi | 6 | Priya Verma | Rajouri Garden, Delhi | Vikram Patil | Shivajinagar, Pune | 2.00 | Pending | TRK006 | 2025-03-27 | |
| 6 | Ananya Iyer | ananya.iyer@example.com | Ananya@987 | 4321098765 | Indira Nagar, Bangalore | 7 | Ananya Iyer | Indira Nagar, Bangalore | Kunal Joshi | Alkapuri, Vadodara | 6.00 | In Transit | TRK007 | 2025-03-23 | |
| 3 | Amit Kumar | amit.kumar@example.com | Amit@789 | 7654321098 | Banjara Hills, Hyderabad | 8 | Amit Kumar | Banjara Hills, Hyderabad | Neha Singh | Sector 62, Noida | 3.75 | Delivered | TRK008 | 2025-03-19 | |

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

mysql> select courierservices.*,courier.* from courier left outer join courierservices on courierservices.serviceid = courier.courierid

union

select courierservices.*,courier.* from courier right outer join courierservices on courierservices.serviceid = courier.courierid;

| ServiceID | ServiceName | Cost | CourierID | SenderName | SenderAddress | ReceiverName | ReceiverAddress | Weight | Status |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Standard Shipping | 500.00 | 1 | Rohan Sharma | Andheri, Mumbai | Priya Verma | Rajouri Garden, Delhi | 2.50 | In Transit |
| 2 | Express Shipping | 1200.00 | 2 | Neha Singh | Sector 62, Noida | Amit Kumar | Banjara Hills, Hyderabad | 5.00 | Delivered |
| 3 | Overnight Shipping | 2500.00 | 3 | Vikram Patil | Shivajinagar, Pune | Ananya Iyer | Indira Nagar, Bangalore | 1.75 | Pending |
| 4 | International Shipping | 8000.00 | 4 | Kunal Joshi | Alkapuri, Vadodara | Meera Nair | Kakkanad, Kochi | 3.25 | In Transit |
| 5 | Same-Day Delivery | 3500.00 | 5 | Meera Nair | Kakkanad, Kochi | Rohan Sharma | Andheri, Mumbai | 4.50 | Delivered |
| 6 | Two-Day Shipping | 1800.00 | 6 | Priya Verma | Rajouri Garden, Delhi | Vikram Patil | Shivajinagar, Pune | 2.00 | Pending |
| 7 | Local Pickup | 200.00 | 7 | Ananya Iyer | Indira Nagar, Bangalore | Kunal Joshi | Alkapuri, Vadodara | 6.00 | In Transit |
| 8 | Freight Shipping | 12000.00 | 8 | Amit Kumar | Banjara Hills, Hyderabad | Neha Singh | Sector 62, Noida | 3.75 | Delivered |

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

select employee.*,payment.* from employee left outer join payment on payment.courierid = employee.employeeid

union

select employee.*,payment.* from employee right outer join payment on payment.courierid = employee.employeeid;

| EmployeeID | Name | Email | ContactNumber | Role | Salary | PaymentID | CourierID | LocationId | Amount |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Rajesh Mehta | rajesh.mehta@example.com | 9876543211 | Manager | 90000.00 | 1 | 1 | 1 | 500.00 |
| 2 | Sunita Rao | sunita.rao@example.com | 9876543212 | Courier Agent | 45000.00 | 2 | 2 | 2 | 1200.00 |
| 3 | Arjun Kapoor | arjun.kapoor@example.com | 9876543213 | Driver | 35000.00 | 3 | 3 | 3 | 2500.00 |
| 4 | Kiran Das | kiran.das@example.com | 9876543214 | Dispatcher | 55000.00 | 4 | 4 | 4 | 3500.00 |
| 5 | Ritika Gupta | ritika.gupta@example.com | 9876543215 | Customer Support | 40000.00 | 5 | 5 | 5 | 1800.00 |
| 6 | Manish Reddy | manish.reddy@example.com | 9876543216 | Warehouse Manager | 70000.00 | 6 | 6 | 6 | 8000.00 |
| 7 | Sneha Choudhury | sneha.choudhury@example.com | 9876543217 | Driver | 38000.00 | 7 | 7 | 7 | 200.00 |
| 8 | Aakash Sen | aakash.sen@example.com | 9876543218 | Security | 50000.00 | 8 | 8 | 8 | 12000.00 |

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

select user.* , courierservices.* from user cross join courierservices;(large output)

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

Select employee.*,location.* from location cross join employee; (large output)

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

select c.courierid,c.sendername,c.senderaddress from courier c;

| courierid | sendername | senderaddress |
|---|---|---|
| 1 | Rohan Sharma | Andheri, Mumbai |
| 2 | Neha Singh | Sector 62, Noida |
| 3 | Vikram Patil | Shivajinagar, Pune |
| 4 | Kunal Joshi | Alkapuri, Vadodara |
| 5 | Meera Nair | Kakkanad, Kochi |
| 6 | Priya Verma | Rajouri Garden, Delhi |
| 7 | Ananya Iyer | Indira Nagar, Bangalore |
| 8 | Amit Kumar | Banjara Hills, Hyderabad |

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

select c.courierid,c.receivername,c.receiveraddress from courier c;

| courierid | receivername | receiveraddress |
|---|---|---|
| 1 | Priya Verma | Rajouri Garden, Delhi |
| 2 | Amit Kumar | Banjara Hills, Hyderabad |
| 3 | Ananya Iyer | Indira Nagar, Bangalore |
| 4 | Meera Nair | Kakkanad, Kochi |
| 5 | Rohan Sharma | Andheri, Mumbai |
| 6 | Vikram Patil | Shivajinagar, Pune |
| 7 | Kunal Joshi | Alkapuri, Vadodara |
| 8 | Neha Singh | Sector 62, Noida |

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

select c.courierid,c.sendername,c.senderaddress,cs.serviceid,cs.servicename,cs.cost from courier c left outer join courierservices cs on cs.serviceid = c.courierid;

| courierid | sendername | senderaddress | serviceid | servicename | cost |
|---|---|---|---|---|---|
| 1 | Rohan Sharma | Andheri, Mumbai | 1 | Standard Shipping | 500.00 |
| 2 | Neha Singh | Sector 62, Noida | 2 | Express Shipping | 1200.00 |
| 3 | Vikram Patil | Shivajinagar, Pune | 3 | Overnight Shipping | 2500.00 |
| 4 | Kunal Joshi | Alkapuri, Vadodara | 4 | International Shipping | 8000.00 |
| 5 | Meera Nair | Kakkanad, Kochi | 5 | Same-Day Delivery | 3500.00 |
| 6 | Priya Verma | Rajouri Garden, Delhi | 6 | Two-Day Shipping | 1800.00 |
| 7 | Ananya Iyer | Indira Nagar, Bangalore | 7 | Local Pickup | 200.00 |
| 8 | Amit Kumar | Banjara Hills, Hyderabad | 8 | Freight Shipping | 12000.00 |

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

select e.employeeid,e.name,count(c.courierid) as couriercount from employee e left outer join courier c on c.employeeid = e.employeeid group by e.employeeid,e.name;

| employeeid | name | couriercount |
|---|---|---|
| 1 | Rajesh Mehta | 1 |
| 2 | Sunita Rao | 1 |
| 3 | Arjun Kapoor | 1 |
| 4 | Kiran Das | 1 |
| 5 | Ritika Gupta | 1 |
| 6 | Manish Reddy | 1 |
| 7 | Sneha Choudhury | 1 |
| 8 | Aakash Sen | 1 |

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

select l.locationid,l.locationname,sum(p.amount) as totalamount from location l join payment p on l.locationid = p.locationid group by l.locationid,l.locationname order by totalamount;

| locationid | locationname | totalamount |
|---|---|---|
| 7 | Vadodara Hub | 200.00 |
| 1 | Mumbai Hub | 500.00 |
| 2 | Delhi Hub | 1200.00 |
| 5 | Pune Hub | 1800.00 |
| 3 | Hyderabad Hub | 2500.00 |
| 4 | Noida Hub | 3500.00 |
| 6 | Bangalore Hub | 8000.00 |
| 8 | Kochi Hub | 12000.00 |

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

| sendername | totalcourier |
|---|---|
| Rohan Sharma | 1 |
| Neha Singh | 1 |
| Vikram Patil | 1 |
| Kunal Joshi | 1 |
| Meera Nair | 1 |
| Priya Verma | 1 |
| Ananya Iyer | 1 |
| Amit Kumar | 1 |

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

| role | employee |
|---|---|
| Courier Agent | Sunita Rao |
| Customer Support | Ritika Gupta |
| Dispatcher | Kiran Das |
| Driver | Arjun Kapoor,Sneha Choudhury |
| Manager | Rajesh Mehta |
| Security | Aakash Sen |
| Warehouse Manager | Manish Reddy |

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

select c.senderaddress, sum(p.amount) as totalpayment from payment p inner join courier c on c.courierid = p.courierid group by c.senderaddress;

| senderaddress | totalpayment |
|---|---|
| Andheri, Mumbai | 500.00 |
| Sector 62, Noida | 1200.00 |
| Shivajinagar, Pune | 2500.00 |
| Alkapuri, Vadodara | 3500.00 |
| Kakkanad, Kochi | 1800.00 |
| Rajouri Garden, Delhi | 8000.00 |
| Indira Nagar, Bangalore | 200.00 |
| Banjara Hills, Hyderabad | 12000.00 |

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

select senderaddress, count(*) as totalcouriers from courier group by senderaddress;

| senderaddress | totalcouriers |
|---|---|
| Andheri, Mumbai | 1 |
| Sector 62, Noida | 1 |
| Shivajinagar, Pune | 1 |
| Alkapuri, Vadodara | 1 |
| Kakkanad, Kochi | 1 |
| Rajouri Garden, Delhi | 1 |
| Indira Nagar, Bangalore | 1 |
| Banjara Hills, Hyderabad | 1 |

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

select e.employeeid,e.name, count(c.courierid) as courierdelivered from employee e inner join courier c on e.employeeid = c.employeeid where c.status = "Delivered" group by e.employeeid,e.name;

| employeeid | name | courierdelivered |
|---|---|---|
| 2 | Sunita Rao | 1 |
| 5 | Ritika Gupta | 1 |
| 8 | Aakash Sen | 1 |

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

select payment.*, courierservices.* from payment inner join courierservices on payment.courierid = courierservices.serviceid where payment.amount > courierservices.cost;

| PaymentID | CourierID | LocationId | Amount | PaymentDate | ServiceID | ServiceName | Cost |
|---|---|---|---|---|---|---|---|
| 6 | 6 | 6 | 8000.00 | 2025-03-23 | 6 | Two-Day Shipping | 1800.00 |

**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 );

| CourierID | SenderName | SenderAddress | ReceiverName | ReceiverAddress | Weight | Status | TrackingNumber | DeliveryDate | EmployeeID |
|---|---|---|---|---|---|---|---|---|---|
| 2 | Neha Singh | Sector 62, Noida | Amit Kumar | Banjara Hills, Hyderabad | 5.00 | Delivered | TRK002 | 2025-03-20 | 2 |
| 5 | Meera Nair | Kakkanad, Kochi | Rohan Sharma | Andheri, Mumbai | 4.50 | Delivered | TRK005 | 2025-03-18 | 5 |
| 7 | Ananya Iyer | Indira Nagar, Bangalore | Kunal Joshi | Alkapuri, Vadodara | 6.00 | In Transit | TRK007 | 2025-03-23 | 7 |
| 8 | Amit Kumar | Banjara Hills, Hyderabad | Neha Singh | Sector 62, Noida | 3.75 | Delivered | TRK008 | 2025-03-19 | 8 |

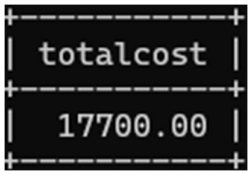
**50. Find the names of all employees who have a salary greater than the average salary:**

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

| name | salary |
|---|---|
| Rajesh Mehta | 90000.00 |
| Kiran Das | 55000.00 |
| Manish Reddy | 70000.00 |

**51. Find the total cost of all courier services where the cost is less than the maximum cost**

select sum(cost) as totalcost from courierservices where cost < (select max(cost) from courierservices);



| totalcost |
|---|
| 17700.00 |

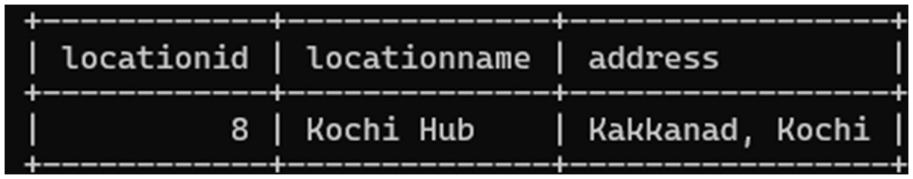
**52. Find all couriers that have been paid for **

select * from courier where courierid in (select distinct(courierid) from payment);

| CourierID | SenderName | SenderAddress | ReceiverName | ReceiverAddress | Weight | Status | TrackingNumber | DeliveryDate | EmployeeID |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Rohan Sharma | Andheri, Mumbai | Priya Verma | Rajouri Garden, Delhi | 2.50 | In Transit | TRK001 | 2025-03-25 | 1 |
| 2 | Neha Singh | Sector 62, Noida | Amit Kumar | Banjara Hills, Hyderabad | 5.00 | Delivered | TRK002 | 2025-03-20 | 2 |
| 3 | Vikram Patil | Shivajinagar, Pune | Ananya Iyer | Indira Nagar, Bangalore | 1.75 | Pending | TRK003 | 2025-03-28 | 3 |
| 4 | Kunal Joshi | Alkapuri, Vadodara | Meera Nair | Kakkanad, Kochi | 3.25 | In Transit | TRK004 | 2025-03-22 | 4 |
| 5 | Meera Nair | Kakkanad, Kochi | Rohan Sharma | Andheri, Mumbai | 4.50 | Delivered | TRK005 | 2025-03-18 | 5 |
| 6 | Priya Verma | Rajouri Garden, Delhi | Vikram Patil | Shivajinagar, Pune | 2.00 | Pending | TRK006 | 2025-03-27 | 6 |
| 7 | Ananya Iyer | Indira Nagar, Bangalore | Kunal Joshi | Alkapuri, Vadodara | 6.00 | In Transit | TRK007 | 2025-03-23 | 7 |
| 8 | Amit Kumar | Banjara Hills, Hyderabad | Neha Singh | Sector 62, Noida | 3.75 | Delivered | TRK008 | 2025-03-19 | 8 |

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

select locationid,locationname, address from location where locationid in (select locationid from payment where amount=(select max(amount) from payment));



| locationid | locationname | address |
|---|---|---|
| 8 | Kochi Hub | Kakkanad, Kochi |

**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 > all(select weight from courier where sendername = 'Meera Nair');

| CourierID | SenderName | SenderAddress | ReceiverName | ReceiverAddress | Weight | Status | TrackingNumber | DeliveryDate | EmployeeID |
|---|---|---|---|---|---|---|---|---|---|
| 2 | Neha Singh | Sector 62, Noida | Amit Kumar | Banjara Hills, Hyderabad | 5.00 | Delivered | TRK002 | 2025-03-20 | 2 |
| 7 | Ananya Iyer | Indira Nagar, Bangalore | Kunal Joshi | Alkapuri, Vadodara | 6.00 | In Transit | TRK007 | 2025-03-23 | 7 |

**Coding**

**Task 1: Control Flow Statements**

1. **Write a program that checks whether a given order is delivered or not based on its status (e.g., "Processing," "Delivered," "Cancelled"). Use if-else statements for this.**

```
package cmstask1;

import java.util.Scanner;

public class OrderStatusCheck {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);

      System.out.print("Enter order status (Processing, Delivered, Cancelled): ");
      String status = scanner.nextLine();

      if (status.equals("Delivered")) {
         System.out.println("Order is delivered.");
      } else if (status.equals("Processing")) {
         System.out.println("Order is still being processed.");
      } else if (status.equals("Cancelled")) {
         System.out.println("Order was cancelled.");
      } else {
         System.out.println("Invalid status entered.");
      }

      scanner.close();
   }
}
```

**Output :**

```
Enter order status (Processing, Delivered, Cancelled): Delivered
Order is delivered.
```

2. **Implement a switch-case statement to categorize parcels based on their weight into "Light," "Medium," or "Heavy."**

```
package cmstask1;

import java.util.Scanner;

public class ParcelWeightCategory {
   public static void main(String[] args) {
```

```
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter parcel weight category (1 for Light, 2 for Medium, 3
    for Heavy): ");
        int weightCode = scanner.nextInt();


        switch (weightCode) {
            case 1:
                System.out.println("Parcel is Light.");
                break;
            case 2:
                System.out.println("Parcel is Medium.");
                break;
            case 3:
                System.out.println("Parcel is Heavy.");
                break;
            default:
                System.out.println("Invalid input.");
        }

        scanner.close();
    }
}
```

```
Enter parcel weight category (1 for Light, 2 for Medium, 3 for Heavy):
2
Parcel is Medium.
```

**3. Implement User Authentication 1. Create a login system for employees and customers using Java control flow statements.**

```
package cmstask1;

import java.util.Scanner;

public class LoginSystem {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Login as: ");
        System.out.println("1. Employee");
        System.out.println("2. Customer");
        System.out.print("Enter your choice (1 or 2): ");
        int choice = scanner.nextInt();
```

```
        scanner.nextLine();

        System.out.print("Enter username: ");
        String username = scanner.nextLine();
        System.out.print("Enter password: ");
        String password = scanner.nextLine();


        if (choice == 1) {

            if (username.equals("employee1") && password.equals("emp123")) {
                System.out.println("Employee login successful!");
            } else {
                System.out.println("Invalid employee credentials.");
            }
        } else if (choice == 2) {

            if (username.equals("customer1") && password.equals("cust123")) {
                System.out.println("Customer login successful!");
            } else {
                System.out.println("Invalid customer credentials.");
            }
        } else {
            System.out.println("Invalid choice.");
        }

        scanner.close();
    }
}
```

```
Login as:
1. Employee
2. Customer
Enter your choice (1 or 2):
Enter username: employee1
Enter password: emp123
Employee login successful!
```

4. **Implement Courier Assignment Logic 1. Develop a mechanism to assign couriers to shipments based on predefined criteria (e.g., proximity, load capacity) using loops.**

```
package cmstask1;

public class CourierAssignment {
    private String name;
    private int loadCapacity;

    public CourierAssignment(String name, int loadCapacity) {
        this.name = name;
        this.loadCapacity = loadCapacity;
    }

    public void assignCourier(int parcelWeight) {
        if (parcelWeight <= loadCapacity) {
            System.out.println("Courier " + name + " is assigned to deliver the parcel.");
        } else {
            System.out.println("Courier " + name + " cannot take this load.");
        }
    }

    public static void main(String[] args) {
        CourierAssignment courier1 = new CourierAssignment("John", 15);
        CourierAssignment courier2 = new CourierAssignment("Emma", 10);

        int parcelWeight = 12;

        courier1.assignCourier(parcelWeight);
        courier2.assignCourier(parcelWeight);
    }
}
```

```
Courier John is assigned to deliver the parcel.
Courier Emma cannot take this load.
```

**Task 2: Loops and Iteration**

5. **Write a Java program that uses a for loop to display all the orders for a specific customer.**

```
package cmstask2;

public class CustomerOrders {
    public static void main(String[] args) {
        String customerName = "John";
        String[] orders = {
            "Order 101 - Mobile Phone",
            "Order 102 - Laptop",
            "Order 103 - Headphones"
        };

        System.out.println("Orders for customer: " + customerName);
        for (int i = 0; i < orders.length; i++) {
            System.out.println(orders[i]);
        }
    }
}
```

```
Orders for customer: John
Order 101 - Mobile Phone
Order 102 - Laptop
Order 103 - Headphones
```

6. **Implement a while loop to track the real-time location of a courier until it reaches its destination.**

```
package cmstask2;

public class CourierTracking {
    public static void main(String[] args) {
        int location = 0;
        int destination = 10;

        System.out.println("Tracking courier location:");

        while (location < destination) {
            System.out.println("Courier is at location: " + location);
            location++;
        }
```

```
            System.out.println("Courier has reached the destination.");
        }
    }
```

```
Tracking courier location:
Courier is at location: 0
Courier is at location: 1
Courier is at location: 2
Courier is at location: 3
Courier is at location: 4
Courier is at location: 5
Courier is at location: 6
Courier is at location: 7
Courier is at location: 8
Courier is at location: 9
Courier has reached the destination.
```

**Task 3: Arrays and Data Structures**

7. **Create an array to store the tracking history of a parcel, where each entry represents a location update.**

```
package cmstask3;

public class ParcelTrackingHistory {
    public static void main(String[] args) {
        String[] trackingHistory = new String[5];

        trackingHistory[0] = "Parcel picked up from warehouse";
        trackingHistory[1] = "Arrived at City Hub";
        trackingHistory[2] = "Out for delivery";
        trackingHistory[3] = "Near destination";
        trackingHistory[4] = "Delivered to customer";

        System.out.println("Tracking History:");
        for (int i = 0; i < trackingHistory.length; i++) {
            System.out.println((i + 1) + ". " + trackingHistory[i]);
        }
    }
}
```

```
Tracking History:
1. Parcel picked up from warehouse
2. Arrived at City Hub
3. Out for delivery
4. Near destination
5. Delivered to customer
```

8. **Implement a method to find the nearest available courier for a new order using an array of couriers.**

```
package cmstask3;

public class NearestCourierFinder {
    String name;
    int distance; // distance from the pickup location
```

```
    boolean available;

    public NearestCourierFinder(String name, int distance, boolean available) {
        this.name = name;
        this.distance = distance;
        this.available = available;
    }

    public static NearestCourierFinder findNearestCourier(NearestCourierFinder[]
couriers) {
        NearestCourierFinder nearest = null;

        for (int i = 0; i < couriers.length; i++) {
            if (couriers[i].available) {
                if (nearest == null || couriers[i].distance < nearest.distance) {
                    nearest = couriers[i];
                }
            }
        }

        return nearest;
    }

    public static void main(String[] args) {
        NearestCourierFinder[] couriers = {
            new NearestCourierFinder("John", 10, true),
            new NearestCourierFinder("Emma", 5, true),
            new NearestCourierFinder("Alex", 7, false)
        };

        NearestCourierFinder assigned = findNearestCourier(couriers);

        if (assigned != null) {
            System.out.println("Assigned courier: " + assigned.name + " (Distance: " +
assigned.distance + ")");
        } else {
            System.out.println("No available courier found.");
        }
    }
}
```

```
Assigned courier: Emma (Distance: 5)
```

**Task 4: Strings,2d Arrays, user defined functions,Hashmap**

9. **Parcel Tracking: Create a program that allows users to input a parcel tracking number.Store the tracking number and Status in 2d String Array. Initialize the array with values. Then, simulate the tracking process by displaying messages like "Parcel in transit," "Parcel out for delivery," or "Parcel delivered" based on the tracking number's status.**

```
package cmstask4;

import java.util.Scanner;

public class ParcelTrackingSystem {
   public static void main(String[] args) {
      String[][] parcels = {
         {"TRK101", "In Transit"},
         {"TRK102", "Out for Delivery"},
         {"TRK103", "Delivered"},
         {"TRK104", "In Transit"},
         {"TRK105", "Delivered"}
      };

      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter your tracking number: ");
      String input = scanner.nextLine();
      boolean found = false;

      for (int i = 0; i < parcels.length; i++) {
         if (parcels[i][0].equalsIgnoreCase(input)) {
            found = true;
            String status = parcels[i][1];

            if (status.equalsIgnoreCase("In Transit")) {
               System.out.println("Your parcel is currently in transit.");
            } else if (status.equalsIgnoreCase("Out for Delivery")) {
               System.out.println("Your parcel is out for delivery.");
            } else if (status.equalsIgnoreCase("Delivered")) {
               System.out.println("Your parcel has been delivered.");
            } else {
               System.out.println("Unknown status.");
            }
            break;
         }
      }

      if (!found) {
         System.out.println("Tracking number not found.");
```

```
        }

        scanner.close();
    }
}
```

```
Enter your tracking number: TRK101
Your parcel is currently in transit.
```

**10. Customer Data Validation: Write a function which takes 2 parameters, data-denotes the data and detail-denotes if it is name addtress or phone number.Validate customer information based on following critirea. Ensure that names contain only letters and are properly capitalized, addresses do not contain special characters, and phone numbers follow a specific format (e.g., ###-###-####).**

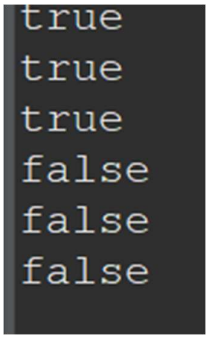
```
package cmstask4;

public class CustomerDataValidation {

    public static boolean validate(String data, String detail) {
        if (detail.equalsIgnoreCase("name")) {
            return data.matches("[A-Z][a-z]+( [A-Z][a-z]+)*");
        } else if (detail.equalsIgnoreCase("address")) {
            return data.matches("[A-Za-z0-9 ,.-]+");
        } else if (detail.equalsIgnoreCase("phone")) {
            return data.matches("\\d{3}-\\d{3}-\\d{4}");
        }
        return false;
    }

    public static void main(String[] args) {
        System.out.println(validate("John Doe", "name"));
        System.out.println(validate("123 Main Street", "address"));
        System.out.println(validate("123-456-7890", "phone"));

        System.out.println(validate("john123", "name"));
        System.out.println(validate("Main@Street", "address"));
        System.out.println(validate("1234567890", "phone"));
    }
}
```




**11. Address Formatting: Develop a function that takes an address as input (street, city, state, zip code) and formats it correctly, including capitalizing the first letter of each word and properly formatting the zip code.**

```
package cmstask4;

public class AddressFormatter {

    public static String formatAddress(String street, String city, String state, String
zip) {
        if (!zip.matches("\\d{5}")) {
            return "Invalid ZIP Code.";
        }

        street = capitalizeWords(street);
        city = capitalizeWords(city);
        state = capitalizeWords(state);

        return street + ", " + city + ", " + state + " - " + zip;
    }

    public static String capitalizeWords(String input) {
        String[] words = input.toLowerCase().split(" ");
        StringBuilder result = new StringBuilder();

        for (String word : words) {
            if (!word.isEmpty()) {
                result.append(Character.toUpperCase(word.charAt(0)))
                      .append(word.substring(1)).append(" ");
            }
        }
        return result.toString().trim();
    }

    public static void main(String[] args) {
        String formatted = formatAddress("123 main street", "new york", "new york",
"10001");
```

```
        System.out.println(formatted);
    }
}
```

```
123 Main Street, New York, New York - 10001
```

**12. Order Confirmation Email: Create a program that generates an order confirmation email. The email should include details such as the customer's name, order number, delivery address, and expected delivery date.**

```
package cmstask4;

public class OrderConfirmationEmail {

    public static String generateEmail(String name, String orderNumber, String
address, String deliveryDate) {
        return "Subject: Order Confirmation - " + orderNumber + "\n\n" +
                "Dear " + name + ",\n\n" +
                "Thank you for your order!\n\n" +
                "Order Number: " + orderNumber + "\n" +
                "Delivery Address: " + address + "\n" +
                "Expected Delivery Date: " + deliveryDate + "\n\n" +
                "We hope you enjoy your purchase.\n\n" +
                "Best regards,\n" +
                "Courier Service Team";
    }

    public static void main(String[] args) {
        String name = "Shantanu Agarkar";
        String orderNumber = "ORD123456";
        String address = "123 Main Street, Pune, Maharashtra - 411001";
        String deliveryDate = "April 4, 2025";

        String email = generateEmail(name, orderNumber, address, deliveryDate);
        System.out.println(email);
    }
}
```

```
Subject: Order Confirmation - ORD123456

Dear Shantanu Agarkar,

Thank you for your order!

Order Number: ORD123456
Delivery Address: 123 Main Street, Pune, Maharashtra - 411001
Expected Delivery Date: April 14, 2025

We hope you enjoy your purchase.

Best regards,
Courier Service Team
```

**13. Calculate Shipping Costs: Develop a function that calculates the shipping cost based on the distance between two locations and the weight of the parcel. You can use string inputs for the source and destination addresses.**

```
package cmstask4;

public class ShippingCostCalculator {

   public static double calculateShippingCost(String source, String destination,
double weight) {
      double baseCost = 50.0;
      double ratePerKm = 5.0;
      double ratePerKg = 10.0;

      int distance = Math.abs(source.length() - destination.length()) * 10;

      double cost = baseCost + (distance * ratePerKm) + (weight * ratePerKg);
      return cost;
   }

   public static void main(String[] args) {
      String source = "Pune";
      String destination = "Mumbai";
      double weight = 3.5;

      double cost = calculateShippingCost(source, destination, weight);
      System.out.println("Shipping Cost: ₹" + cost);
   }
}
```

```
Shipping Cost: ₹185.0
```

**14. Password Generator: Create a function that generates secure passwords for courier system accounts. Ensure the passwords contain a mix of uppercase letters, lowercase letters, numbers, and special characters.**

```
package cmstask4;

import java.util.Random;

public class PasswordGenerator {

    public static String generatePassword(int length) {
        String upper = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        String lower = "abcdefghijklmnopqrstuvwxyz";
        String numbers = "0123456789";
        String special = "!@#$%^&*()-_=+<>?";

        String allChars = upper + lower + numbers + special;

        Random rand = new Random();
        StringBuilder password = new StringBuilder();

        password.append(upper.charAt(rand.nextInt(upper.length())));
        password.append(lower.charAt(rand.nextInt(lower.length())));
        password.append(numbers.charAt(rand.nextInt(numbers.length())));
        password.append(special.charAt(rand.nextInt(special.length())));

        for (int i = 4; i < length; i++) {
            password.append(allChars.charAt(rand.nextInt(allChars.length())));
        }

        return password.toString();
    }

    public static void main(String[] args) {
        String password = generatePassword(12);
        System.out.println("Generated Password: " + password);
    }
}
```

```
Generated Password: Wq2*Y0I+k^+T
```

**15. Find Similar Addresses: Implement a function that finds similar addresses in the system. This can be useful for identifying duplicate customer entries or optimizing delivery routes.Use string functions to implement this.**

```
package cmstask4;

public class SimilarAddressFinder {

    public static void findSimilarAddresses(String[] addresses, String input) {
        String cleanedInput = input.toLowerCase().replaceAll("[^a-z0-9 ]", "");

        System.out.println("Similar addresses to: " + input);
        for (String addr : addresses) {
            String cleanedAddr = addr.toLowerCase().replaceAll("[^a-z0-9 ]", "");
            if (cleanedAddr.contains(cleanedInput) ||
cleanedInput.contains(cleanedAddr)) {
                System.out.println("- " + addr);
            }
        }
    }

    public static void main(String[] args) {
        String[] addressList = {
            "123 Main Street, Pune",
            "124 Main St., Pune",
            "45 MG Road, Mumbai",
            "123 main street pune",
            "123 Main St, PUNE",
            "88 Central Ave, Nagpur"
        };

        String searchAddress = "123 Main Street, Pune";

        findSimilarAddresses(addressList, searchAddress);
    }
}
```

```
Similar addresses to: 123 Main Street, Pune
- 123 Main Street, Pune
- 123 main street pune
```

**Task 5-9**

**Package Entity**

**Class : User**

```
package entity;

public class User {
   private int userID;
   private String userName;
   private String email;
   private String password;
   private String contactNumber;
   private String address;

   public User() {}

   public User(int userID, String userName, String email, String password, String
contactNumber, String address) {
      this.userID = userID;
      this.userName = userName;
      this.email = email;
      this.password = password;
      this.contactNumber = contactNumber;
      this.address = address;
   }

   public int getUserID() {
      return userID;
   }

   public void setUserID(int userID) {
      this.userID = userID;
   }

   public String getUserName() {
      return userName;
   }

   public void setUserName(String userName) {
      this.userName = userName;
   }

   public String getEmail() {
      return email;
   }
```

```
    public void setEmail(String email) {
        this.email = email;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getContactNumber() {
        return contactNumber;
    }

    public void setContactNumber(String contactNumber) {
        this.contactNumber = contactNumber;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    @Override
    public String toString() {
        return "User{" +
                "userID=" + userID +
                ", userName='" + userName + '\'' +
                ", email='" + email + '\'' +
                ", password='" + password + '\'' +
                ", contactNumber='" + contactNumber + '\'' +
                ", address='" + address + '\'' +
                '}';
    }
}
```

**Class : Courier**

```
package entity;

public class Courier {
   private static int trackingSeed = 10000;
   private int courierID;
   private String senderName;
   private String senderAddress;
   private String receiverName;
   private String receiverAddress;
   private double weight;
   private String status;
   private String trackingNumber;
   private String deliveryDate;
   private int userId;
   private int employeeID;

   public Courier() {
      this.trackingNumber = generateTrackingNumber();
   }

   public Courier(int courierID, String senderName, String senderAddress, String
receiverName,
            String receiverAddress, double weight, String status, String deliveryDate, int
userId) {
      this.courierID = courierID;
      this.senderName = senderName;
      this.senderAddress = senderAddress;
      this.receiverName = receiverName;
      this.receiverAddress = receiverAddress;
      this.weight = weight;
      this.status = status;
      this.deliveryDate = deliveryDate;
      this.userId = userId;
      this.trackingNumber = generateTrackingNumber();
   }

   private String generateTrackingNumber() {
      return "TRK" + (++trackingSeed);
   }

   public int getCourierID() {
      return courierID;
   }

   public void setCourierID(int courierID) {
```

```
    this.courierID = courierID;
}

public String getSenderName() {
    return senderName;
}

public void setSenderName(String senderName) {
    this.senderName = senderName;
}

public String getSenderAddress() {
    return senderAddress;
}

public void setSenderAddress(String senderAddress) {
    this.senderAddress = senderAddress;
}

public String getReceiverName() {
    return receiverName;
}

public void setReceiverName(String receiverName) {
    this.receiverName = receiverName;
}

public String getReceiverAddress() {
    return receiverAddress;
}

public void setReceiverAddress(String receiverAddress) {
    this.receiverAddress = receiverAddress;
}

public double getWeight() {
    return weight;
}

public void setWeight(double weight) {
    this.weight = weight;
}

public String getStatus() {
    return status;
}
```

```
public void setStatus(String status) {
   this.status = status;
}

public String getTrackingNumber() {
   return trackingNumber;
}

public void setTrackingNumber(String trackingNumber) {
   this.trackingNumber = trackingNumber;
}

public String getDeliveryDate() {
   return deliveryDate;
}

public void setDeliveryDate(String deliveryDate) {
   this.deliveryDate = deliveryDate;
}

public int getUserId() {
   return userId;
}

public void setUserId(int userId) {
   this.userId = userId;
}

@Override
public String toString() {
   return "Courier{" +
         "courierID=" + courierID +
         ", senderName='" + senderName + '\'' +
         ", senderAddress='" + senderAddress + '\'' +
         ", receiverName='" + receiverName + '\'' +
         ", receiverAddress='" + receiverAddress + '\'' +
         ", weight=" + weight +
         ", status='" + status + '\'' +
         ", trackingNumber='" + trackingNumber + '\'' +
         ", deliveryDate='" + deliveryDate + '\'' +
         ", userId=" + userId +
         '}';
}

      public int getEmployeeID() {
            return employeeID;
      }
```

```
        public void setEmployeeID(int employeeID) {
                this.employeeID = employeeID;
        }
}
```

**Class : Employee**

```
package entity;

public class Employee {
   private int employeeID;
   private String employeeName;
   private String email;
   private String contactNumber;
   private String role;
   private double salary;

   public Employee() {}

   public Employee(int employeeID, String employeeName, String email, String
contactNumber, String role, double salary) {
      this.employeeID = employeeID;
      this.employeeName = employeeName;
      this.email = email;
      this.contactNumber = contactNumber;
      this.role = role;
      this.salary = salary;
   }

   public int getEmployeeID() {
      return employeeID;
   }

   public void setEmployeeID(int employeeID) {
      this.employeeID = employeeID;
   }

   public String getEmployeeName() {
      return employeeName;
   }

   public void setEmployeeName(String employeeName) {
      this.employeeName = employeeName;
   }
```

```
    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getContactNumber() {
        return contactNumber;
    }

    public void setContactNumber(String contactNumber) {
        this.contactNumber = contactNumber;
    }

    public String getRole() {
        return role;
    }

    public void setRole(String role) {
        this.role = role;
    }

    public double getSalary() {
        return salary;
    }

    public void setSalary(double salary) {
        this.salary = salary;
    }

    @Override
    public String toString() {
        return "Employee{" +
                "employeeID=" + employeeID +
                ", employeeName='" + employeeName + '\'' +
                ", email='" + email + '\'' +
                ", contactNumber='" + contactNumber + '\'' +
                ", role='" + role + '\'' +
                ", salary=" + salary +
                '}';
    }
}
```

**Class : courier company**

```
package entity;

import java.util.List;

public class CourierCompany {
   private String companyName;
   private List<Courier> courierDetails;
   private List<Employee> employeeDetails;
   private List<Location> locationDetails;

   public CourierCompany() {}

   public CourierCompany(String companyName, List<Courier> courierDetails,
                  List<Employee> employeeDetails, List<Location> locationDetails) {
      this.companyName = companyName;
      this.courierDetails = courierDetails;
      this.employeeDetails = employeeDetails;
      this.locationDetails = locationDetails;
   }

   public String getCompanyName() {
      return companyName;
   }

   public void setCompanyName(String companyName) {
      this.companyName = companyName;
   }

   public List<Courier> getCourierDetails() {
      return courierDetails;
   }

   public void setCourierDetails(List<Courier> courierDetails) {
      this.courierDetails = courierDetails;
   }

   public List<Employee> getEmployeeDetails() {
      return employeeDetails;
   }

   public void setEmployeeDetails(List<Employee> employeeDetails) {
      this.employeeDetails = employeeDetails;
   }

   public List<Location> getLocationDetails() {
```

```
        return locationDetails;
    }

    public void setLocationDetails(List<Location> locationDetails) {
        this.locationDetails = locationDetails;
    }

    @Override
    public String toString() {
        return "CourierCompany{" +
                "companyName='" + companyName + '\'' +
                ", courierDetails=" + courierDetails +
                ", employeeDetails=" + employeeDetails +
                ", locationDetails=" + locationDetails +
                '}';
    }
}
```

**Class : Location**

```
package entity;

public class Location {
    private int locationID;
    private String locationName;
    private String address;

    public Location() {}

    public Location(int locationID, String locationName, String address) {
        this.locationID = locationID;
        this.locationName = locationName;
        this.address = address;
    }

    public int getLocationID() {
        return locationID;
    }

    public void setLocationID(int locationID) {
        this.locationID = locationID;
    }

    public String getLocationName() {
        return locationName;
    }
```

```
    public void setLocationName(String locationName) {
        this.locationName = locationName;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    @Override
    public String toString() {
        return "Location{" +
                "locationID=" + locationID +
                ", locationName='" + locationName + '\'' +
                ", address='" + address + '\'' +
                '}';
    }
}
```

**Class : Payment**

```
package entity;

import java.util.Date;

public class Payment {
    private long paymentID;
    private long courierID;
    private double amount;
    private Date paymentDate;

    public Payment() {}

    public Payment(long paymentID, long courierID, double amount, Date paymentDate)
{
        this.paymentID = paymentID;
        this.courierID = courierID;
        this.amount = amount;
        this.paymentDate = paymentDate;
    }

    public long getPaymentID() {
```

```
        return paymentID;
    }

    public void setPaymentID(long paymentID) {
        this.paymentID = paymentID;
    }

    public long getCourierID() {
        return courierID;
    }

    public void setCourierID(long courierID) {
        this.courierID = courierID;
    }

    public double getAmount() {
        return amount;
    }

    public void setAmount(double amount) {
        this.amount = amount;
    }

    public Date getPaymentDate() {
        return paymentDate;
    }

    public void setPaymentDate(Date paymentDate) {
        this.paymentDate = paymentDate;
    }

    @Override
    public String toString() {
        return "Payment{" +
                "paymentID=" + paymentID +
                ", courierID=" + courierID +
                ", amount=" + amount +
                ", paymentDate=" + paymentDate +
                '}';
    }
}
```

**Package : dao**

**Interface : ICourierUserService**

```
package dao;

import entity.Courier;
import exception.InvalidEmployeeIdException;
import exception.TrackingNumberNotFoundException;

import java.util.List;

public interface ICourierUserService {


   String placeOrder(Courier courierObj);


   String getOrderStatus(String trackingNumber) throws
TrackingNumberNotFoundException;


   boolean cancelOrder(String trackingNumber) throws
TrackingNumberNotFoundException;


   List<Courier> getAssignedOrder(int courierStaffId) throws
InvalidEmployeeIdException;
}
```

**Interface  : IcourierAdminService**

```
package dao;

import entity.Employee;

public interface ICourierAdminService {

   int addCourierStaff(Employee obj);
}
```

**Class : CourierUserServiceImpl**

```
package dao;

import entity.Courier;
import exception.InvalidEmployeeIdException;
import exception.TrackingNumberNotFoundException;
import util.DBConnUtil;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.util.*;

public class CourierUserServiceImpl implements ICourierUserService {

   private static Map<String, Courier> courierMap = new HashMap<>();
   private static Set<Integer> validEmployeeIds = new HashSet<>(Arrays.asList(101,
102, 103)); // Sample valid IDs

   @Override
   public String placeOrder(Courier courierObj) {
      courierMap.put(courierObj.getTrackingNumber(), courierObj);
      saveCourierToDB(courierObj); // Save to database
      return courierObj.getTrackingNumber();
   }

   @Override
   public String getOrderStatus(String trackingNumber) throws
TrackingNumberNotFoundException {
      Courier c = courierMap.get(trackingNumber);
      if (c == null) {
         throw new TrackingNumberNotFoundException("Tracking number not found: "
+ trackingNumber);
      }
      return c.getStatus();
   }

   @Override
   public boolean cancelOrder(String trackingNumber) throws
TrackingNumberNotFoundException {
      Courier c = courierMap.get(trackingNumber);
      if (c == null) {
         throw new TrackingNumberNotFoundException("Cannot cancel. Tracking
number not found: " + trackingNumber);
      }
      if (!"Delivered".equalsIgnoreCase(c.getStatus())) {
         c.setStatus("Cancelled");
```

```
            return true;
        }
        return false;
    }

    @Override
    public List<Courier> getAssignedOrder(int courierStaffId) throws
InvalidEmployeeIdException {
        if (!validEmployeeIds.contains(courierStaffId)) {
            throw new InvalidEmployeeIdException("Invalid employee ID: " +
courierStaffId);
        }

        List<Courier> assigned = new ArrayList<>();
        for (Courier c : courierMap.values()) {
            if (c.getEmployeeID() == courierStaffId) {
                assigned.add(c);
            }
        }
        return assigned;
    }

    // ✅ Save Courier to Database
    public boolean saveCourierToDB(Courier courier) {
        String query = "INSERT INTO Courier (senderName, senderAddress, receiverName,
receiverAddress, weight, status, trackingNumber, deliveryDate, userId, employeeID)
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

        try (Connection conn = DBConnUtil.getConnection("src/db.properties");
             PreparedStatement ps = conn.prepareStatement(query)) {

            ps.setString(1, courier.getSenderName());
            ps.setString(2, courier.getSenderAddress());
            ps.setString(3, courier.getReceiverName());
            ps.setString(4, courier.getReceiverAddress());
            ps.setDouble(5, courier.getWeight());
            ps.setString(6, courier.getStatus());
            ps.setString(7, courier.getTrackingNumber());
            ps.setString(8, courier.getDeliveryDate());
            ps.setInt(9, courier.getUserId());
            ps.setInt(10, courier.getEmployeeID());

            int rows = ps.executeUpdate();
            return rows > 0;
        } catch (Exception e) {
            e.printStackTrace();
        }
```

```
        return false;
    }
}
```

**Class : CourierAdminServiceImpl**

```
package dao;

import entity.Employee;
import util.DBConnUtil;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.util.HashMap;
import java.util.Map;

public class CourierAdminServiceImpl implements ICourierAdminService {

    private static Map<Integer, Employee> employeeMap = new HashMap<>();
    private static int employeeIdCounter = 2000;

    @Override
    public int addCourierStaff(Employee obj) {
        int id = ++employeeIdCounter;
        obj.setEmployeeID(id);
        employeeMap.put(id, obj);
        saveEmployeeToDB(obj); // Save employee to database
        return id;
    }

    // ✅ Save Employee to DB
    public boolean saveEmployeeToDB(Employee emp) {
        String query = "INSERT INTO Employee (employeeID, employeeName, email,
contactNumber, role, salary) VALUES (?, ?, ?, ?, ?, ?)";

        try (Connection conn = DBConnUtil.getConnection("src/db.properties");
             PreparedStatement ps = conn.prepareStatement(query)) {

            ps.setInt(1, emp.getEmployeeID());
            ps.setString(2, emp.getEmployeeName());
            ps.setString(3, emp.getEmail());
            ps.setString(4, emp.getContactNumber());
            ps.setString(5, emp.getRole());
            ps.setDouble(6, emp.getSalary());

            int rows = ps.executeUpdate();
```

```
            return rows > 0;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return false;
    }
}
```

**Package : Exceptions**

**Class : InvalidEmployeeIdException**

```
package exception;

public class InvalidEmployeeIdException extends Exception {
    public InvalidEmployeeIdException(String message) {
        super(message);
    }
}
```

**Class : TrackingNumberNotFoundException**

```
package exception;

public class TrackingNumberNotFoundException extends Exception {
    public TrackingNumberNotFoundException(String message) {
        super(message);
    }
}
```

**Package : main**

**Class :MainModule**

```
package main;

import dao.*;
import entity.Courier;
import entity.Employee;
import exception.InvalidEmployeeIdException;
import exception.TrackingNumberNotFoundException;

import java.util.List;
import java.util.Scanner;
```

```
public class MainModule {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      ICourierUserService userService = new CourierUserServiceImpl();
      ICourierAdminService adminService = new CourierAdminServiceImpl();

      int choice;
      do {
         System.out.println("\n====== Courier Management System ======");
         System.out.println("1. Add Courier Staff");
         System.out.println("2. Place Order");
         System.out.println("3. Get Order Status");
         System.out.println("4. Cancel Order");
         System.out.println("5. Get Assigned Orders");
         System.out.println("6. Exit");
         System.out.print("Enter your choice: ");
         choice = sc.nextInt();
         sc.nextLine(); // consume newline

         try {
            switch (choice) {
            case 1:
               System.out.print("Enter Employee Name: ");
               String name = sc.nextLine();

               System.out.print("Enter Email: ");
               String email = sc.nextLine();

               System.out.print("Enter Contact Number: ");
               String contactNumber = sc.nextLine();

               System.out.print("Enter Role: ");
               String role = sc.nextLine();

               System.out.print("Enter Salary: ");
               double salary = sc.nextDouble();
               sc.nextLine(); // consume leftover newline

               Employee emp = new Employee(0, name, email, contactNumber, role,
salary);
               int empId = adminService.addCourierStaff(emp);
               System.out.println("Courier Staff Added with ID: " + empId);
               break;


               case 2:
```

```
                System.out.print("Enter sender name: ");
                String sender = sc.nextLine();
                System.out.print("Enter sender address: ");
                String senderAddress = sc.nextLine();
                System.out.print("Enter receiver name: ");
                String receiver = sc.nextLine();
                System.out.print("Enter receiver address: ");
                String receiverAddress = sc.nextLine();
                System.out.print("Enter weight: ");
                double weight = sc.nextDouble();
                sc.nextLine();
                System.out.print("Enter delivery date (yyyy-mm-dd): ");
                String deliveryDate = sc.nextLine();
                System.out.print("Enter employee ID to assign: ");
                int staffId = sc.nextInt();
                sc.nextLine();

                Courier courier = new Courier(0, sender, senderAddress, receiver,
receiverAddress, weight, "yetToTransit", deliveryDate, staffId);
                String tracking = userService.placeOrder(courier);
                System.out.println("Order placed successfully. Tracking Number: " +
tracking);
                break;

            case 3:
                System.out.print("Enter tracking number: ");
                String trackStatus = sc.nextLine();
                String status = userService.getOrderStatus(trackStatus);
                System.out.println("Order Status: " + status);
                break;

            case 4:
                System.out.print("Enter tracking number to cancel: ");
                String trackCancel = sc.nextLine();
                boolean cancelled = userService.cancelOrder(trackCancel);
                if (cancelled) {
                    System.out.println("Order cancelled successfully.");
                } else {
                    System.out.println("Unable to cancel order (maybe already delivered
or invalid).");
                }
                break;

            case 5:
                System.out.print("Enter employee ID: ");
                int staffID = sc.nextInt();
                List<Courier> list = userService.getAssignedOrder(staffID);
```

```
                System.out.println("Assigned Orders:");
                for (Courier c : list) {
                    System.out.println(c);
                }
                break;

            case 6:
                System.out.println("Exiting... Thank you!");
                break;

            default:
                System.out.println("Invalid choice! Try again.");
            }
        } catch (TrackingNumberNotFoundException | InvalidEmployeeIdException e) {
            System.err.println("Error: " + e.getMessage());
        } catch (Exception e) {
            System.err.println("Unexpected Error: " + e.getMessage());
        } finally {
            System.out.println("---------------------------------------");
        }
    } while (choice != 6);

    sc.close();
  }
}
```

**Package : Util**

**Class : DBConnUtil**

```
package util;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

public class DBConnUtil {

  public static Connection getConnection(String propertyFileName) {
    Connection conn = null;
    try {
      Properties props = DBPropertyUtil.getConnectionProperties(propertyFileName);
      String url = props.getProperty("db.url");
      String username = props.getProperty("db.username");
      String password = props.getProperty("db.password");
```

```
            conn = DriverManager.getConnection(url, username, password);
        } catch (SQLException e) {
            System.err.println("Database connection failed: " + e.getMessage());
        }
        return conn;
    }
}
```

**Class : DBPropertyutil**

```
package util;

import java.io.FileInputStream;
import java.io.IOException;
import java.util.Properties;

public class DBPropertyUtil {
    public static Properties getConnectionProperties(String fileName) {
        Properties props = new Properties();
        try (FileInputStream fis = new FileInputStream(fileName)) {
            props.load(fis);
        } catch (IOException e) {
            e.printStackTrace();
        }
        return props;
    }
}
```

**Db.properties file**

```
db.url=jdbc:mysql://localhost:3306/CourierDB
db.username=root
db.password=root
```

**Output :**

```
====== Courier Management System ======
1. Add Courier Staff
2. Place Order
3. Get Order Status
4. Cancel Order
5. Get Assigned Orders
6. Exit
Enter your choice:
```

**1.**

```
====== Courier Management System ======
1. Add Courier Staff
2. Place Order
3. Get Order Status
4. Cancel Order
5. Get Assigned Orders
6. Exit
Enter your choice: 1
Enter Employee Name: mohan
Enter Email: mohan@gmail.com
Enter Contact Number: 987655
Enter Role: manager
Enter Salary: 90000
Courier Staff Added with ID: 2002
-----------------------------------
```

**2.**

```
====== Courier Management System ======
1. Add Courier Staff
2. Place Order
3. Get Order Status
4. Cancel Order
5. Get Assigned Orders
6. Exit
Enter your choice: 2
Enter sender name: raj
Enter sender address: pune
Enter receiver name: maya
Enter receiver address: mumbai
Enter weight: 12
Enter delivery date (yyyy-mm-dd): 2025-01-01
Enter employee ID to assign: 2
Order placed successfully. Tracking Number: TRK10002
-------------------------------------
```

**3.**

```
====== Courier Management System ======
1. Add Courier Staff
2. Place Order
3. Get Order Status
4. Cancel Order
5. Get Assigned Orders
6. Exit
Enter your choice: 3
Enter tracking number: TRK10002
Order Status: yetToTransit
-------------------------------------
```

**4.**

```
====== Courier Management System ======
1. Add Courier Staff
2. Place Order
3. Get Order Status
4. Cancel Order
5. Get Assigned Orders
6. Exit
Enter your choice: 4
Enter tracking number to cancel: TRK10002
Order cancelled successfully.
--------------------------------------
```