

# **DBMS MINI-PROJECT REPORT**

**PES University**

## **Database Management Systems**

**UE17CS252**

**TOPIC: COURIER MANAGEMENT SYSTEM**

**SUBMITTED BY: TEAM NUMBER-9**

- Lakshmi Harakuni SRN: PES1UG19EC145
- P.M. Shalini SRN: PES1UG19EE075
- Navya N SRN: PES1UG19EE069

### **Introduction(Problem Statement):**

✚ This project aims to develop a database for managing the data and operations of a parcel delivery service. Delivery services are a necessary component of modern society and should be made as seamless and efficient as possible. A lot of data is generated in this service, making it impossible to maintain manually.

✚ A computerized database is more professional and economical to handle. This database will hold records of all clients associated with the service, all parcels being handled, consignments ferrying these parcels between hubs and information about the network of messengers who complete the process. It keeps track of the payment status and delivery status and delays.

✚ In addition, it can be used to find patterns in consumer behaviour, employee performance, and keep up with the demands of customers. The database can be used to extract critical information, such as the demand in each city, performance of the delivery agents and allocation of potential bonuses, and obtain relevant statistics to improve the consumer experience. It can be used to identify trends in the market, and company growth in terms of keeping up with the changes.

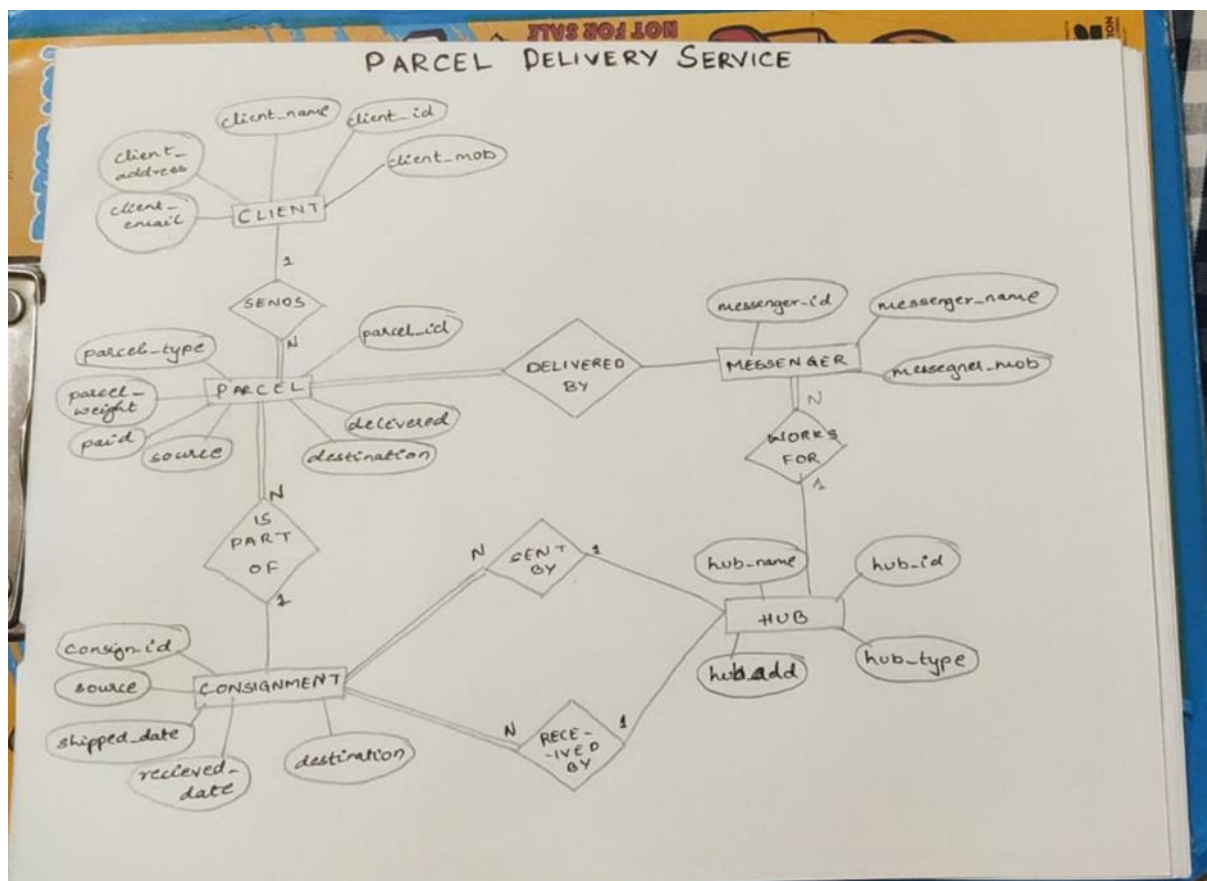
✚ In conclusion, this is a simple, yet robust database that can effectively keep track of transactions in a courier delivery system and can be used to extract required information as and when required, as well as ensure smooth functioning of the service.

## INTRODUCTION:

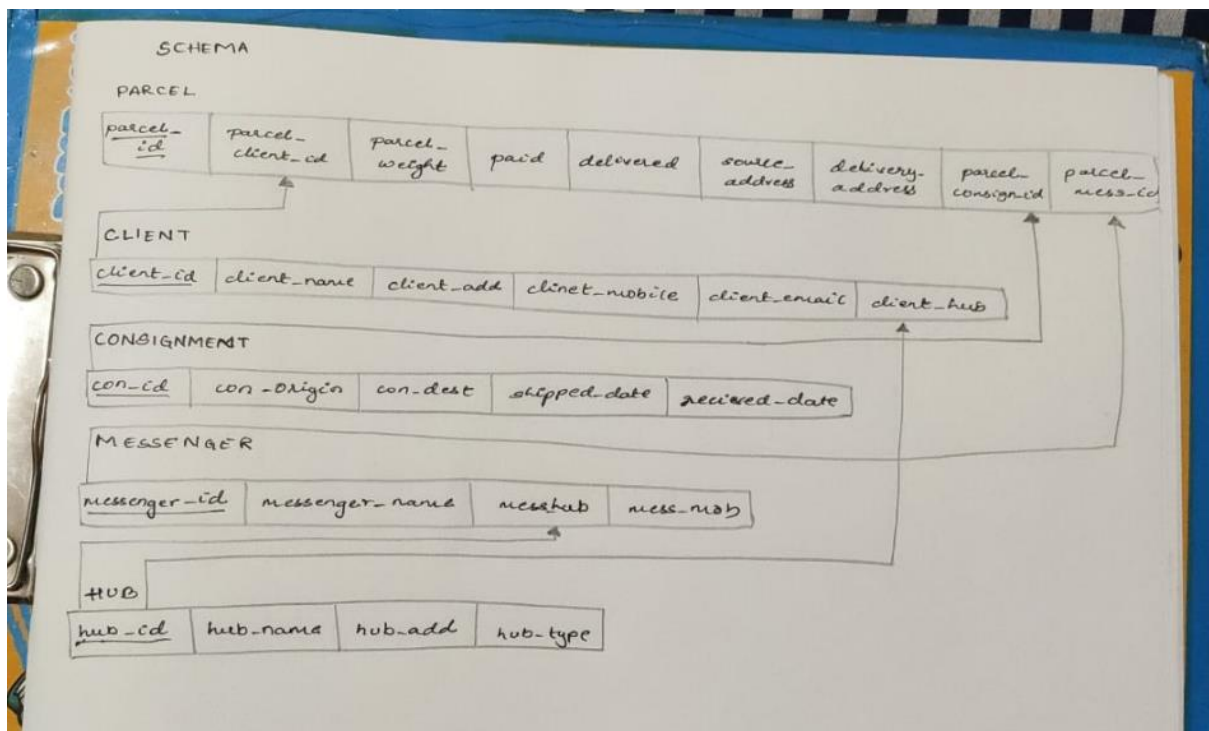
The Parcel Service has clients(CLIENT), and each client possesses a unique client\_ID, name, mobile number, email\_ID and address. The clients send parcels(PARCEL), and these parcels have a unique parcelID, type, weight, source address, destination address, payment status, and delivery status.

These packages are part of consignments(CONSIGNMENT), which have a unique consignmentID, source address, destination address, shipping date, and received date. The consignments are transported between HUBS(HUB), which have a unique HUBID, address, type and name. The parcels are delivered by messengers(MESSENGER), who have a unique ID and mobile number, and are associated with a HUB.

## ER Diagram:



## Relational schema of parcel Delivery Service:



### Data Types Used:

**INT:** parcel\_id, client\_id, consignment\_id, messenger\_id, hub\_id (since they are all unique identifiers)

**VARCHAR:** parcel\_type, paid, delivered, source\_address, dest\_address, client\_name, client\_address, client\_email, ship\_src\_addr, ship\_dst\_addr, messenger\_name, hub\_name, hub\_address, hub\_type (since they are all strings of varying length)

**DECIMAL:** parcel\_weight (since weight can be a decimal value, expressed in KGs).

**CHAR:** client\_mobile, messenger\_mobile (since mobile numbers have a fixed length of 10 digits).

**DATE:** shipped\_date, received\_date (since they are dates)

**parcel\_type** takes two possible values: regular and speed

**payment\_status** takes two possible values: No and Yes

**delivery\_status** takes two possible values: No and Yes

**hub\_type** takes two possible values: sales and warehouse

### Creating The Tables:

### Constraints included:

- 1) Enforcing the Primary Key Constraint.
- 2) Check constraint to ensure mobile number is valid.
- 3) Check constraint to ensure email id is valid.
- 4) Check constraint to ensure received\_date is greater than or equal to shipped\_date.
- 5) Enforcing Foreign Key Constraints for all foreign keys.

Creating the database <courier\_shalini>

```
postgres=#  
postgres=#  
postgres=# CREATE DATABASE COURIER_SHALINI;  
CREATE DATABASE  
postgres=#
```

## CREATING TABLES AND INSERTING THE VALUES INTO THE TABLES:

TABLE HUB:

```
postgres=# \c courier_shalini;  
You are now connected to database "courier_shalini" as user "postgres".  
courier_shalini=#  
courier_shalini=#  
courier_shalini=#  
courier_shalini=# CREATE TABLE HUB  
courier_shalini=# (  
courier_shalini(#  hub_id          INT PRIMARY KEY,  
courier_shalini(#  hub_name        VARCHAR(50) NOT NULL,  
courier_shalini(#  hub_add         VARCHAR(100) NOT NULL,  
courier_shalini(#  hub_type        VARCHAR(10) DEFAULT 'warehouse'  
courier_shalini(# );  
CREATE TABLE  
courier_shalini=#
```

```

courier_shalini=# INSERT INTO HUB VALUES (1,'PUN1','Pune','sales');
INSERT 0 1
courier_shalini=# INSERT INTO HUB VALUES (2,'PUN2','Pune','warehouse');
INSERT 0 1
courier_shalini=# INSERT INTO HUB VALUES (3,'DEL1','Delhi','sales');
INSERT 0 1
courier_shalini=# INSERT INTO HUB VALUES (4,'DEL2','Delhi','warehouse');
INSERT 0 1
courier_shalini=# INSERT INTO HUB VALUES (5,'BLR1','Bengaluru','sales');
INSERT 0 1
courier_shalini=# INSERT INTO HUB VALUES (6,'BLR2','Bengaluru','warehouse');
INSERT 0 1
courier_shalini=# INSERT INTO HUB VALUES (7,'CHN1','Chennai','warehouse');
INSERT 0 1
courier_shalini=# INSERT INTO HUB VALUES (8,'LCK2','Lucknow','warehouse');
INSERT 0 1
courier_shalini=#
courier_shalini=#
courier_shalini=# select * from hub;
 hub_id | hub_name | hub_add | hub_type
-----+-----+-----+-----
      1 | PUN1     | Pune    | sales
      2 | PUN2     | Pune    | warehouse
      3 | DEL1     | Delhi   | sales
      4 | DEL2     | Delhi   | warehouse
      5 | BLR1     | Bengaluru | sales
      6 | BLR2     | Bengaluru | warehouse
      7 | CHN1     | Chennai | warehouse
      8 | LCK2     | Lucknow  | warehouse
(8 rows)

```

## TABLE MESSENGER:

```

courier_shalini=# CREATE TABLE MESSENGER
courier_shalini=# (
courier_shalini(# messenger_id INT PRIMARY KEY,
courier_shalini(# messenger_name VARCHAR(50) NOT NULL,
courier_shalini(# mess_hub INT NOT NULL,
courier_shalini(# mess_mob CHAR(10) NOT NULL CHECK (mess_mob SIMILAR TO '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
courier_shalini(# FOREIGN KEY (mess_hub) REFERENCES HUB(hub_id)
courier_shalini(# );
CREATE TABLE
courier_shalini=#

```

```

courier_shalini=#
courier_shalini=# INSERT INTO MESSENGER VALUES (1,'Ram',2,'8792431902');
INSERT 0 1
courier_shalini=# INSERT INTO MESSENGER VALUES (2,'Sham',4,'1729333609');
INSERT 0 1
courier_shalini=# INSERT INTO MESSENGER VALUES (3,'Sita',6,'9027395815');
INSERT 0 1
courier_shalini=# INSERT INTO MESSENGER VALUES (4,'Gita',7,'7403814927');
INSERT 0 1
courier_shalini=# INSERT INTO MESSENGER VALUES (5,'Rukmini',8,'9204719192');
INSERT 0 1
courier_shalini=# INSERT INTO MESSENGER VALUES (6,'Ganesh',2,'8373102748');
INSERT 0 1
courier_shalini=# INSERT INTO MESSENGER VALUES (7,'Radha',4,'8102746192');
INSERT 0 1
courier_shalini=# INSERT INTO MESSENGER VALUES (8,'Krishna',6,'7102906565');
INSERT 0 1
courier_shalini=# INSERT INTO MESSENGER VALUES (9,'Shiva',7,'1728192731');
INSERT 0 1
courier_shalini=# INSERT INTO MESSENGER VALUES (10,'Vishnu',8,'9274482190');
INSERT 0 1
courier_shalini=#
courier_shalini=# select * from messenger;
 messenger_id | messenger_name | mess_hub | mess_mob
-----+-----+-----+-----
          1 | Ram            |         2 | 8792431902
          2 | Sham           |         4 | 1729333609
          3 | Sita           |         6 | 9027395815
          4 | Gita           |         7 | 7403814927
          5 | Rukmini        |         8 | 9204719192
          6 | Ganesh         |         2 | 8373102748
          7 | Radha          |         4 | 8102746192
          8 | Krishna        |         6 | 7102906565
          9 | Shiva          |         7 | 1728192731
         10 | Vishnu         |         8 | 9274482190
(10 rows)

```

## TABLE CONSIGNMENT:

```

courier_shalini=# CREATE TABLE CONSIGNMENT
courier_shalini=# (
courier_shalini(# con_id INT PRIMARY KEY,
courier_shalini(# con_origin VARCHAR(100) NOT NULL,
courier_shalini(# con_dest VARCHAR(100) NOT NULL,
courier_shalini(# shipped_date DATE NOT NULL,
courier_shalini(# received_date DATE NOT NULL CHECK (received_date >= shipped_date AND shipped_date >= '1970-01-01') );
CREATE TABLE
courier_shalini=#

```

```

courier_shalini=# INSERT INTO CONSIGNMENT VALUES (1,'Pune','Delhi','2020-05-02','2020-05-10');
INSERT 0 1
courier_shalini=# INSERT INTO CONSIGNMENT VALUES (2,'Delhi','Bengaluru','2020-03-25','2020-04-01');
INSERT 0 1
courier_shalini=# INSERT INTO CONSIGNMENT VALUES (3,'Pune','Lucknow','2020-02-21','2020-02-26');
INSERT 0 1
courier_shalini=# INSERT INTO CONSIGNMENT VALUES (4,'Delhi','Chennai','2020-03-06','2020-03-10');
INSERT 0 1
courier_shalini=# INSERT INTO CONSIGNMENT VALUES (5,'Bengaluru','Pune','2020-04-07','2020-04-15');
INSERT 0 1
courier_shalini=# INSERT INTO CONSIGNMENT VALUES (6,'Pune','Delhi','2020-03-25','2020-04-01');
INSERT 0 1
courier_shalini=# INSERT INTO CONSIGNMENT VALUES (7,'Delhi','Lucknow','2020-02-21','2020-02-26');
INSERT 0 1
courier_shalini=# INSERT INTO CONSIGNMENT VALUES (8,'Bengaluru','Chennai','2020-03-25','2020-04-01');
INSERT 0 1
courier_shalini=# INSERT INTO CONSIGNMENT VALUES (9,'Pune','Delhi','2020-03-25','2020-03-27');
INSERT 0 1
courier_shalini=# INSERT INTO CONSIGNMENT VALUES (10,'Delhi','Pune','2020-02-23','2020-02-26');
INSERT 0 1
courier_shalini=#
courier_shalini=# select * from consignment;
 con_id | con_origin | con_dest | shipped_date | received_date
-----+-----+-----+-----+-----
      1 | Pune       | Delhi    | 2020-05-02  | 2020-05-10
      2 | Delhi      | Bengaluru| 2020-03-25  | 2020-04-01
      3 | Pune       | Lucknow  | 2020-02-21  | 2020-02-26
      4 | Delhi      | Chennai  | 2020-03-06  | 2020-03-10
      5 | Bengaluru  | Pune     | 2020-04-07  | 2020-04-15
      6 | Pune       | Delhi    | 2020-03-25  | 2020-04-01
      7 | Delhi      | Lucknow  | 2020-02-21  | 2020-02-26
      8 | Bengaluru  | Chennai  | 2020-03-25  | 2020-04-01
      9 | Pune       | Delhi    | 2020-03-25  | 2020-03-27
     10 | Delhi      | Pune     | 2020-02-23  | 2020-02-26
(10 rows)

courier_shalini=#

```

## TABLE CLIENT:

```

courier_shalini=# CREATE TABLE CLIENT
courier_shalini=# (
courier_shalini(# client_id INT PRIMARY KEY,
courier_shalini(# client_name VARCHAR(50) NOT NULL,
courier_shalini(# client_add VARCHAR(100) NOT NULL,
courier_shalini(# client_mobile CHAR(10) NOT NULL CHECK (client_mobile SIMILAR TO '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
courier_shalini(# client_email VARCHAR(50) NOT NULL CHECK (client_email SIMILAR TO '%@%.%'),
courier_shalini(# client_hub INT NOT NULL,
courier_shalini(# FOREIGN KEY (client_hub) REFERENCES HUB(hub_id) );
CREATE TABLE
courier_shalini=#

```



```

courier_shalini=# INSERT INTO CLIENT VALUES (1,'Hari','Pune','3791204739','hari@gmail.com',1);
INSERT 0 1
courier_shalini=# INSERT INTO CLIENT VALUES (2,'Kishan','Delhi','2738218102','kishan@gmail.com',3);
INSERT 0 1
courier_shalini=# INSERT INTO CLIENT VALUES (3,'Prabhu','Bengaluru','2839109244','prabhu@gmail.com',6);
INSERT 0 1
courier_shalini=# INSERT INTO CLIENT VALUES (4,'Kartik','Chennai','6201458139','kartik@gmail.com',7);
INSERT 0 1
courier_shalini=# INSERT INTO CLIENT VALUES (5,'Manjeet','Lucknow','8791237398','manjeet@gmail.com',8);
INSERT 0 1
courier_shalini=# INSERT INTO CLIENT VALUES (6,'Kavita','Pune','8536343142','kavita@gmail.com',1);
INSERT 0 1
courier_shalini=# INSERT INTO CLIENT VALUES (7,'Geeta','Delhi','7102749271','geeta@gmail.com',3);
INSERT 0 1
courier_shalini=# INSERT INTO CLIENT VALUES (8,'Raj','Bengaluru','8712093476','raj@gmail.com',6);
INSERT 0 1
courier_shalini=# INSERT INTO CLIENT VALUES (9,'Ravi','Chennai','8712038192','ravi@gmail.com',7);
INSERT 0 1
courier_shalini=# INSERT INTO CLIENT VALUES (10,'Surya','Lucknow','8291027394','surya@gmail.com',8);
INSERT 0 1
courier_shalini=# select * from client;

```

client_id	client_name	client_add	client_mobile	client_email	client_hub
1	Hari	Pune	3791204739	hari@gmail.com	1
2	Kishan	Delhi	2738218102	kishan@gmail.com	3
3	Prabhu	Bengaluru	2839109244	prabhu@gmail.com	6
4	Kartik	Chennai	6201458139	kartik@gmail.com	7
5	Manjeet	Lucknow	8791237398	manjeet@gmail.com	8
6	Kavita	Pune	8536343142	kavita@gmail.com	1
7	Geeta	Delhi	7102749271	geeta@gmail.com	3
8	Raj	Bengaluru	8712093476	raj@gmail.com	6
9	Ravi	Chennai	8712038192	ravi@gmail.com	7
10	Surya	Lucknow	8291027394	surya@gmail.com	8

```

(10 rows)

```

## TABLE PARCEL:

```

courier_shalini=# CREATE TABLE PARCEL
courier_shalini=# (
courier_shalini(#  parcel_id          INT          PRIMARY KEY,
courier_shalini(#  parcel_sender_id      INT          NOT NULL,
courier_shalini(#  parcel_type VARCHAR(50)    DEFAULT 'regular',
courier_shalini(#  parcel_weight      DECIMAL    NOT NULL,
courier_shalini(#  paid varchar(5)    DEFAULT 'YES',
courier_shalini(#  delivered varchar(5) DEFAULT 'NO',
courier_shalini(#  src_city      VARCHAR(100) NOT NULL,
courier_shalini(#  dst_city      VARCHAR(100) NOT NULL,
courier_shalini(#  parcel_consign_id INT    NOT NULL,
courier_shalini(#  par_mess_id   INT    NOT NULL,
courier_shalini(#  FOREIGN KEY (parcel_sender_id) REFERENCES CLIENT(client_id),
courier_shalini(#  FOREIGN KEY (parcel_consign_id) REFERENCES CONSIGNMENT(con_id),
courier_shalini(#  FOREIGN KEY (par_mess_id) REFERENCES MESSENGER(messenger_id) );
CREATE TABLE

```



```
courier_shalini=# INSERT INTO PARCEL VALUES (1,1,'regular',12.34,'YES','NO','Pune','Delhi',9,2);
INSERT 0 1
courier_shalini=# INSERT INTO PARCEL VALUES (2,2,'regular',8.09,'YES','NO','Delhi','Bengaluru',2,3);
INSERT 0 1
courier_shalini=# INSERT INTO PARCEL VALUES (3,3,'speed',7.23,'NO','NO','Bengaluru','Chennai',8,4);
INSERT 0 1
courier_shalini=# INSERT INTO PARCEL VALUES (4,6,'regular',6.54,'NO','NO','Pune','Lucknow',3,10);
INSERT 0 1
courier_shalini=# INSERT INTO PARCEL VALUES (5,7,'speed',19.2,'YES','YES','Delhi','Lucknow',7,5);
INSERT 0 1
courier_shalini=# INSERT INTO PARCEL VALUES (6,3,'regular',6.89,'YES','YES','Bengaluru','Pune',5,6);
INSERT 0 1
courier_shalini=# INSERT INTO PARCEL VALUES (7,6,'regular',5.32,'NO','NO','Pune','Delhi',1,2);
INSERT 0 1
courier_shalini=#
courier_shalini=#
courier_shalini=#
courier_shalini=# select * from parcel;
 parcel_id | parcel_sender_id | parcel_type | parcel_weight | paid | delivered | src_city | dst_city | parcel_consign_id | par_mess_id
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
          1 |                1 | regular    |         12.34 | YES  | NO        | Pune     | Delhi    |                9 |            2
          2 |                2 | regular    |          8.09 | YES  | NO        | Delhi    | Bengaluru |                2 |            3
          3 |                3 | speed      |          7.23 | NO   | NO        | Bengaluru | Chennai  |                8 |            4
          4 |                6 | regular    |          6.54 | NO   | NO        | Pune     | Lucknow  |                3 |           10
          5 |                7 | speed      |         19.2 | YES  | YES       | Delhi    | Lucknow  |                7 |            5
          6 |                3 | regular    |          6.89 | YES  | YES       | Bengaluru | Pune     |                5 |            6
          7 |                6 | regular    |          5.32 | NO   | NO        | Pune     | Delhi    |                1 |            2
(7 rows)
```

## SIMPLE QUERIES:

1) Print all the cities where the service delivers

```
courier_shalini=#
courier_shalini=#
courier_shalini=# SELECT HUB_add FROM HUB WHERE HUB_type='warehouse';
 hub_add
-----
Pune
Delhi
Bengaluru
Chennai
Lucknow
(5 rows)
```

2) Print the city in which each delivery messenger delivers

```
courier_shalini=# SELECT DISTINCT MESSENGER.messenger_name, HUB.HUB_add
courier_shalini=# FROM MESSENGER,HUB
courier_shalini=# WHERE MESSENGER.mess_hub=HUB.hub_id;
 messenger_name | hub_add
-----+-----
 Shiva          | Chennai
 Sita           | Bengaluru
 Rukmini        | Lucknow
 Krishna        | Bengaluru
 Radha          | Delhi
 Ram            | Pune
 Gita           | Chennai
 Sham           | Delhi
 Vishnu         | Lucknow
 Ganesh         | Pune
(10 rows)
```

3) Print details of all consignments arriving into Pune

```
courier_shalini=# SELECT CONSIGNMENT.con_origin,CONSIGNMENT.con_dest,CONSIGNMENT.shipped_date,CONSIGNMENT.received_date
courier_shalini=# FROM CONSIGNMENT
courier_shalini=# WHERE CONSIGNMENT.con_dest='Pune';
 con_origin | con_dest | shipped_date | received_date
-----+-----+-----+-----
 Bengaluru  | Pune    | 2020-04-07  | 2020-04-15
 Delhi      | Pune    | 2020-02-23  | 2020-02-26
(2 rows)
```

4) Print the client name and delivery messenger for each parcel

```
courier_shalini=# SELECT PARCEL.parcel_id, CLIENT.client_name, MESSENGER.messenger_name
courier_shalini=# FROM PARCEL,CLIENT,MESSENGER
courier_shalini=# WHERE CLIENT.client_id=PARCEL.parcel_sender_id
courier_shalini=# AND MESSENGER.messenger_id=PARCEL.par_mess_id;
 parcel_id | client_name | messenger_name
-----+-----+-----
          7 | Kavita      | Sham
          1 | Hari        | Sham
          2 | Kishan      | Sita
          3 | Prabhu      | Gita
          5 | Geeta       | Rukmini
          6 | Prabhu      | Ganesh
          4 | Kavita      | Vishnu
(7 rows)
```

## **COMPLEX QUERIES:**

1) Find the number of deliveries undertaken by each messenger

(Aggregate function)

```
courier_shalini=# SELECT MESSENGER.messenger_name, COUNT(*)
courier_shalini=# FROM MESSENGER,PARCEL
courier_shalini=# WHERE MESSENGER.messenger_id=PARCEL.par_mess_id
courier_shalini=# GROUP BY MESSENGER.messenger_name;
 messenger_name | count
-----+-----
 Ganesh         |      1
 Vishnu         |      1
 Rukmini        |      1
 Sita           |      1
 Sham           |      2
 Gita           |      1
(6 rows)

courier_shalini=#
```

2) Check the usage of delivery services by clients by ranking them according to number of parcels sent. This information can be used to identify clients for gold memberships.

(aggregate function)

```
courier_shalini=#
courier_shalini=# SELECT CLIENT.client_name, COUNT(*)
courier_shalini=# FROM CLIENT,PARCEL
courier_shalini=# WHERE CLIENT.client_id=PARCEL.parcel_sender_id
courier_shalini=# GROUP BY CLIENT.client_name;
 client_name | count
-----+-----
 Hari        |      1
 Kishan      |      1
 Kavita      |      2
 Geeta       |      1
 Prabhu      |      2
(5 rows)
```

3) Print the parcelr\_id, mobile number of associated client and source address for each parcel

(Full Outer Join)

```
courier_shalini=#
courier_shalini=# SELECT parcel_id, client_name, client_add
courier_shalini-# FROM PARCEL
courier_shalini-# FULL OUTER JOIN CLIENT
courier_shalini-# ON PARCEL.parcel_sender_id=CLIENT.client_id;
 parcel_id | client_name | client_add
-----+-----+-----
          1 | Hari        | Pune
          2 | Kishan      | Delhi
          3 | Prabhu      | Bengaluru
          4 | Kavita      | Pune
          5 | Geeta       | Delhi
          6 | Prabhu      | Bengaluru
          7 | Kavita      | Pune
           | Surya       | Lucknow
           | Manjeet     | Lucknow
           | Raj         | Bengaluru
           | Kartik      | Chennai
           | Ravi        | Chennai
(12 rows)
```

4) Identify the number of outgoing parcels from each city

(Nested Query)

```
courier_shalini=# SELECT PARCEL.src_city,COUNT(*)
courier_shalini-# FROM PARCEL
courier_shalini-# WHERE (PARCEL.src_city IN
courier_shalini(# (SELECT CONSIGNMENT.con_origin
courier_shalini(# FROM CONSIGNMENT,PARCEL
courier_shalini(# WHERE PARCEL.parcel_consign_id=CONSIGNMENT.con_id))
courier_shalini-# GROUP BY PARCEL.src_city ;
 src_city | count
-----+-----
 Delhi    |      2
 Bengaluru |      2
 Pune     |      3
(3 rows)
```

## **Conclusion**

In conclusion, this is a robust system for maintaining and recording the day-to-day activities of a delivery service, including but not limited to sending parcels, monitoring consignments, keeping track of delays, recording client activity, client details and performance of messengers. It is also resourceful in compiling statistics for different cities and messengers, and tracking the company performance to provide better service.

### Limitations and Future enhancements:

This system does not keep any track of the amount of money to be paid for each parcel. In the future, I would like to categorize parcels by weight with different prices for each category,

and charge extra in case of excess weight. This system does not allow the client to track a consignment, or determine the means of transport being used for the shipment. There is also no provision for the system to assign the most optimal consignment to each parcel automatically. I would also like to differentiate between standard and express consignments and the constraints associated with these. These are some things which I would like to include going forward, along with developing a user friendly interface.