# Design Phase Report

## TITLE: MAIL SERVICE DATABASE

**Team Leader**:

**Team No**: 4

**Batch**: E1&E2

**List of team members:**

| Name | Usn | Roll No |
|---|---|---|
| Nirmala R Kanti | 01FE21BCS180 | 534 |
| Prajwal Mutnal | 01FE21BCS159 | 529 |
| Rohan B Jadhav | 01FE21BCS272 | 557 |
| Kavya Handigund | 01FE21BCS131 | 526 |

**Responsibilities:** What is the role of each project member in this phase of project?

| Name | Responsibilities |
|---|---|
| Nirmala R Kanti | Report ,PLSQL ,frontend |
| Prajwal Mutnal | Report ,PLSQL ,frontend |
| Rohan Jadhav | Report ,PLSQL ,frontend |
| Kavya Handigund | Report ,PLSQL ,frontend |

## Problem Description:

The mail service database is a database that is used to store and manage information related to a mail delivery service. It could include information such as the sender and recipient of the mail, and delivery details. The database could be used to perform various functions such as tracking the delivery status of a particular piece of a mail , generating invoices and receipt for customers and analyzing sdata to improve the efficiency of the mail delivery service.Overall , the mail service database is an essential tool for any organization involved in mail delivery and can help streamline processes, reduce errors and improve customers satisfaction.

## Requirements:

- **Customer Information Management**: The database should allow for the management of customer information, including Name, address, contact information, and delivery preferences.

- **Payment And Billing Management:** The database should support the management of payment and billing for mail delivery services, including the generation of invoices and receipts,as well as the management of payment methods.

- **Employee Management:** The database should support the management of employee information, including name, contact information, and employee history. it should also enable the scheduling of employees for delivery services.

## Design Questions to be answered:

**Question 1:**From the problem description, identify the entities that need to be represented in the database, the attributes of each entity, the relationships between the entities, and the cardinality ratios of each relationship.

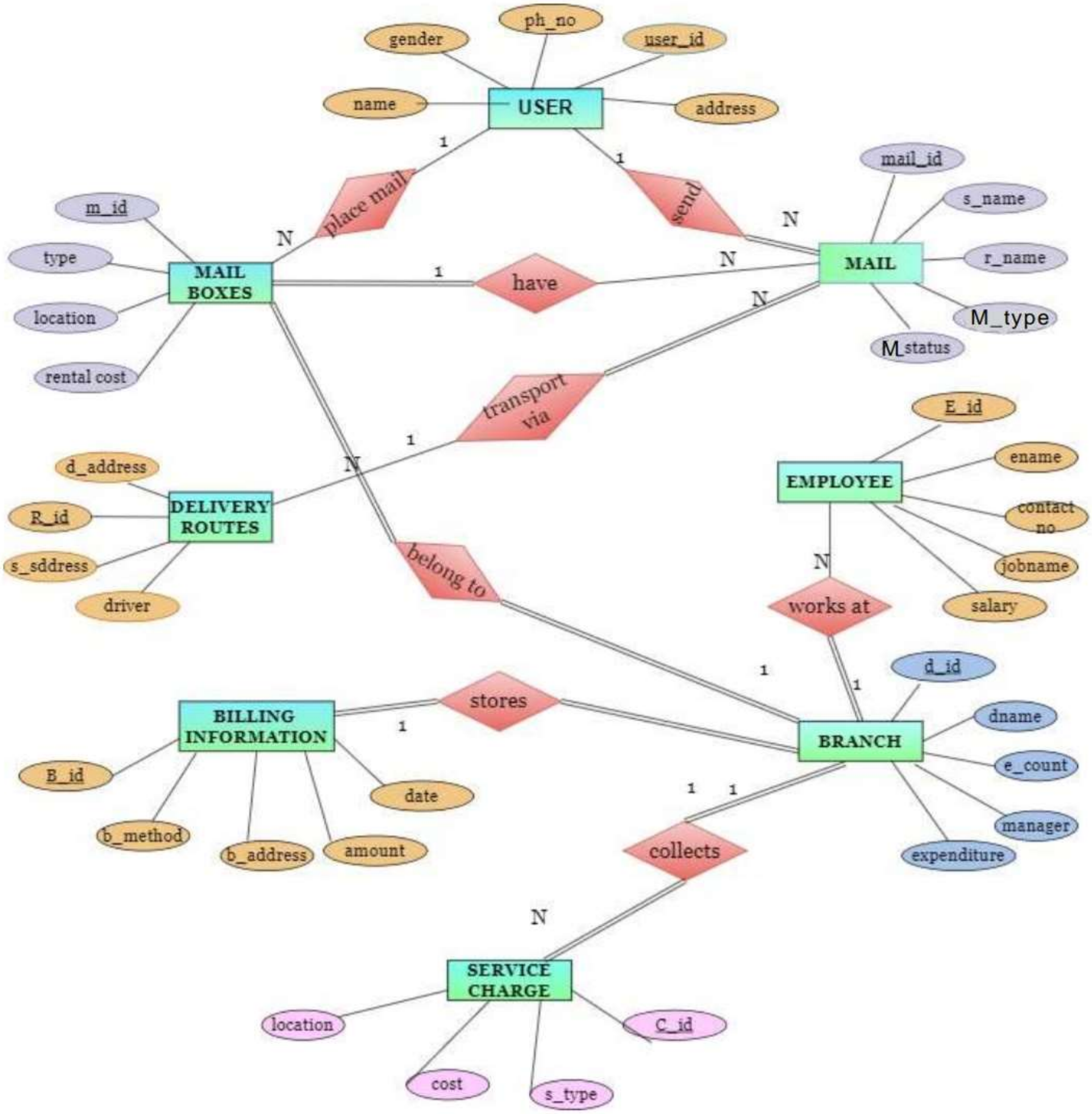
### Entities and Attributes:

| | | | | | | |
|---|---|---|---|---|---|---|
| **User** | U_Name | U_id | Gender | Address | Ph_no | |
| **Mail boxes** | M_id | Rentalcost | MB_type | Size | loc | |
| **Mail** | Mail_id | S_name | R_name | M_type | D_status | Date |
| **Employee** | E_id | E_name | Cont_no | J_name | | |
| **Branch** | De_id | De_name | M_name | E_cnt | Budget | |

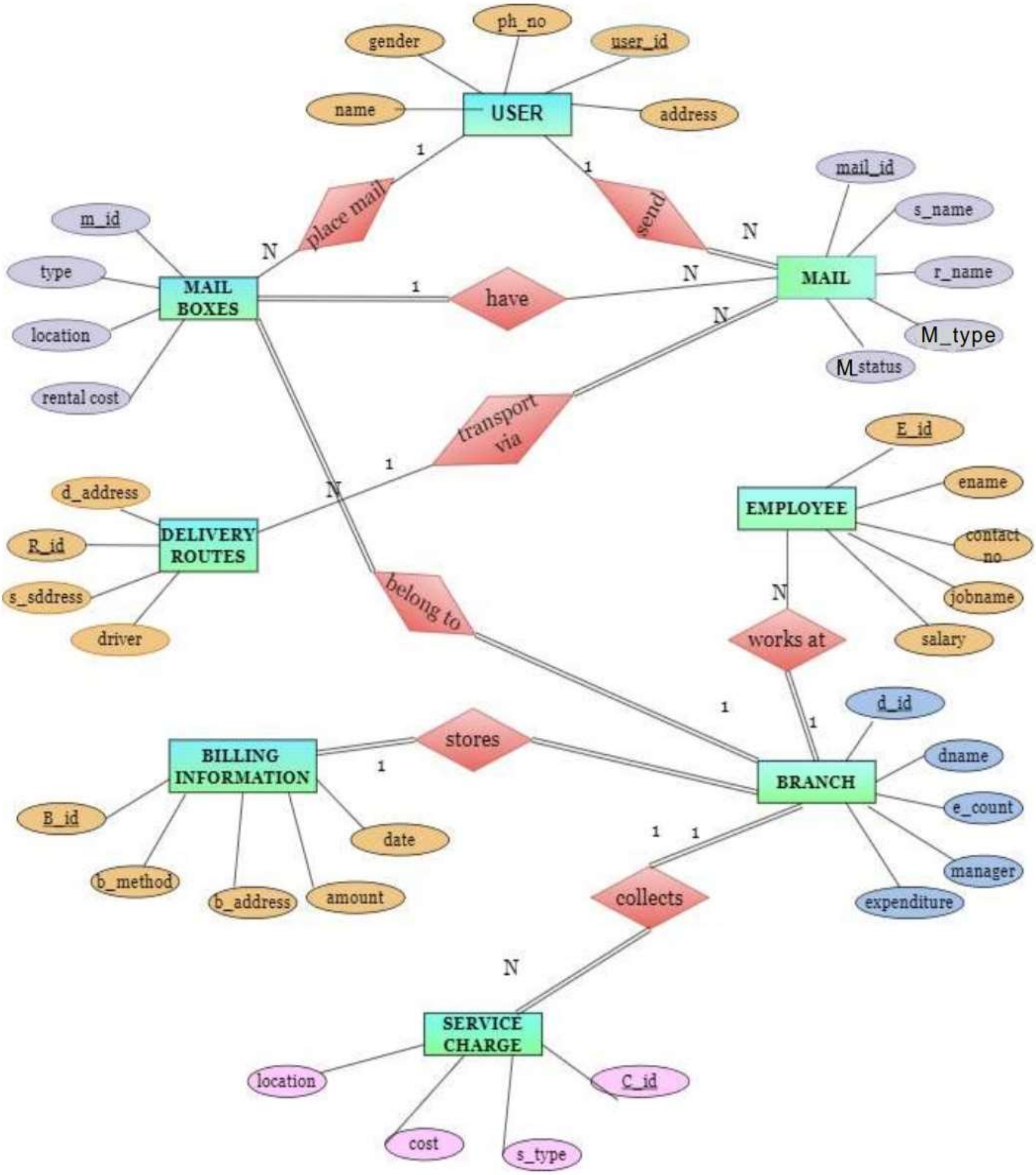
### Relationship between Entities and Cardinality Ratio:

| **Entity 1** | **Relation name** | **Entity 2** | **Cardinality ratio** |
|---|---|---|---|
| User | Send | Mail | 1:N |
| User | Place mail at | Mail boxes | 1:N |
| Mail boxes | Collect | Mail | 1:N |
| Mail boxes | Collect mail from | Employee | N:1 |
| Mail boxes | Belongs to | Branch | N:1 |
| Employee | Works at | Branch | 1:N |

**Question 2:**Draw an Entity-Relationship Diagram illustrating the information you have identified in Question 1.



**Question 3:**Draw alternate Entity-Relationship Diagram illustrating the information you have identified in Question 1 that you think are most likely to occur.



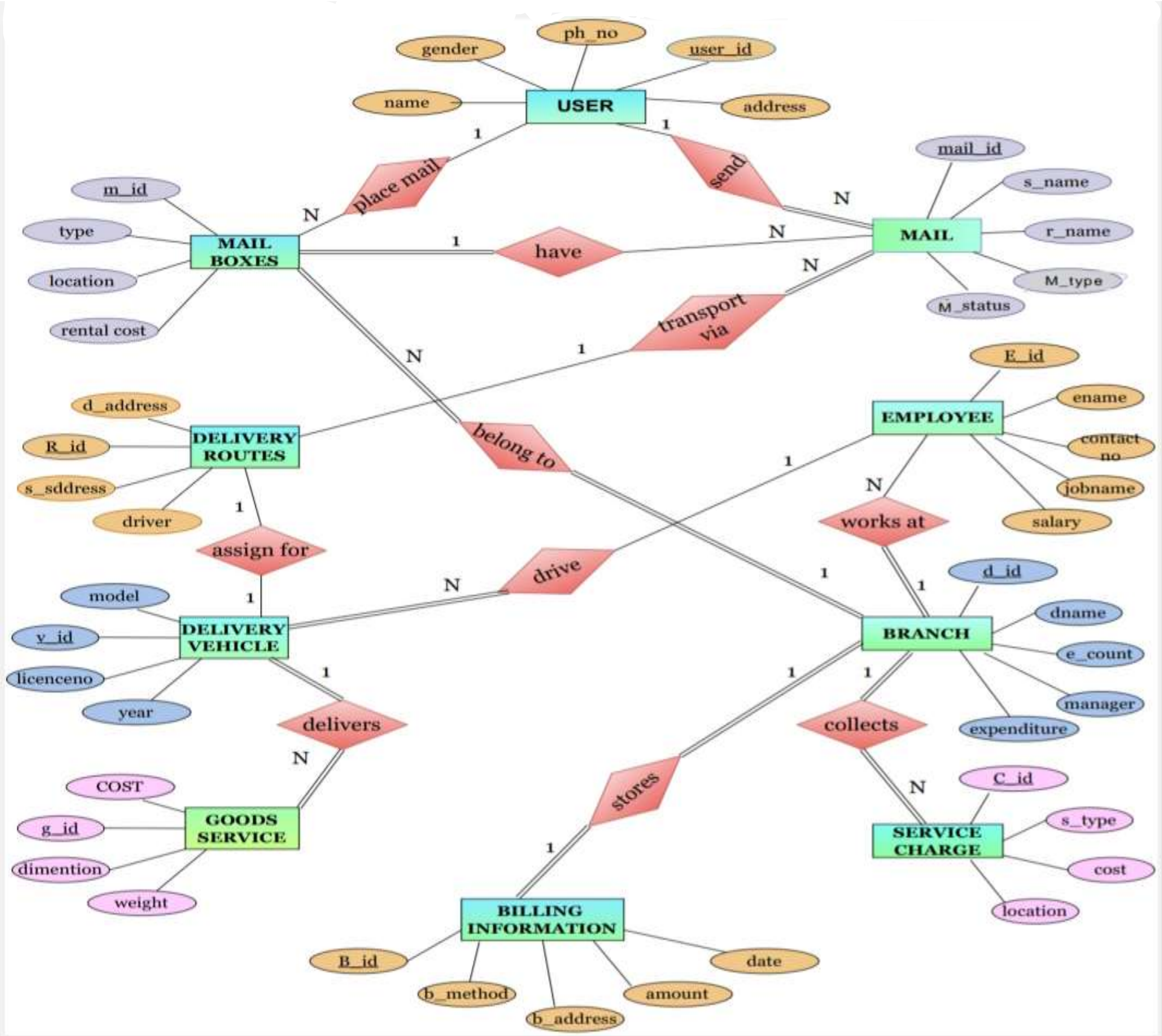
**Question 4:**Choose the optimal Entity-Relationship Diagram from the designs provided above and justify why you think this is an optimal solution for your identified problem specification.

a) We have identified some more entities which are important related to the system.

b) We have added some more attributes to the previous entities.

c) This is the optimal one because it has almost all the information which the system needs for any information

d) We have mostly covered all the information which will be needed.



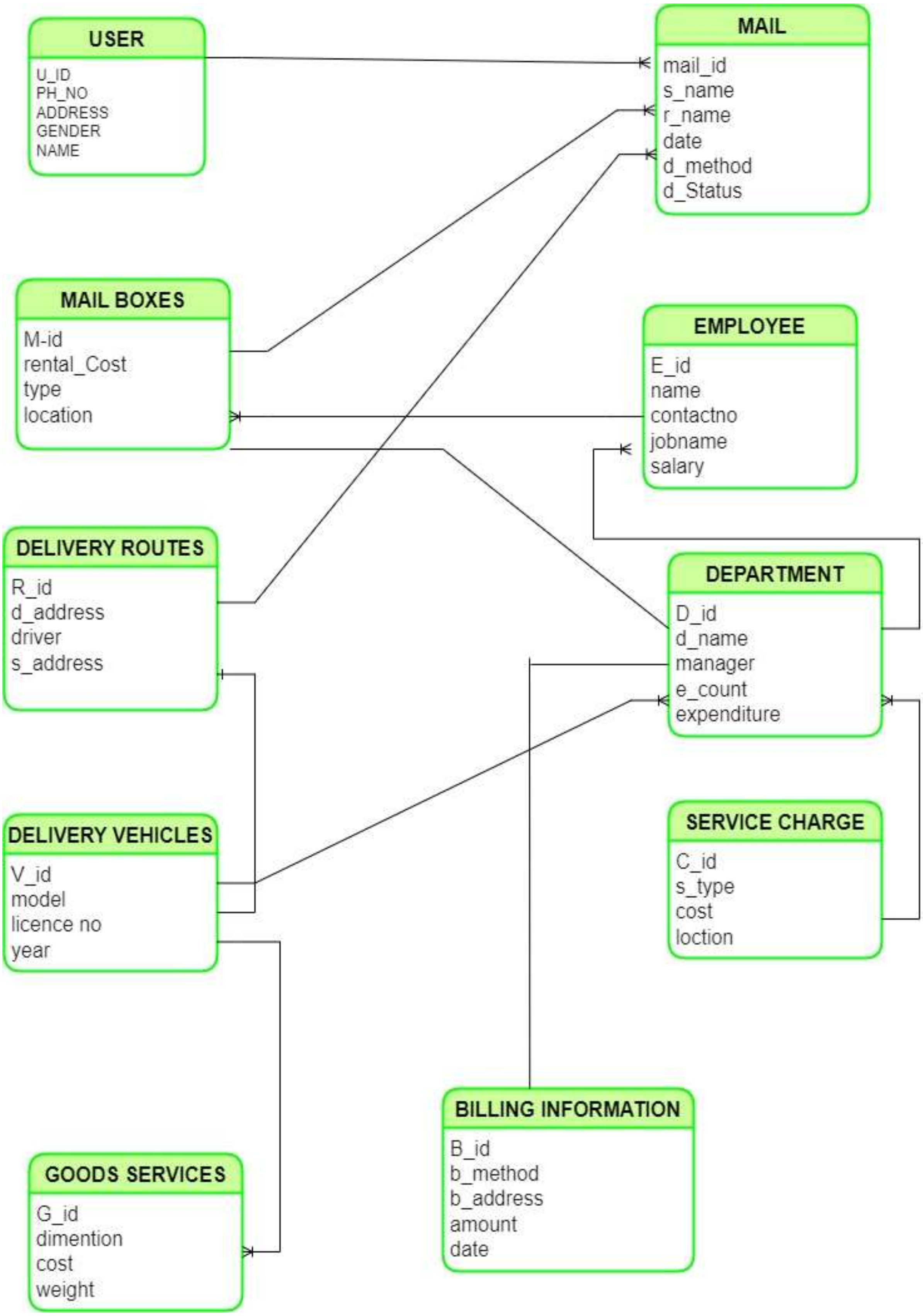
## Entities and Attributes:

| | | | | | | |
|---|---|---|---|---|---|---|
| **User** | U_Name | U_id | Gender | Address | Ph_no | |
| **Mail boxes** | M_id | Rentalcost | MB_type | Size | loc | |
| **Delivery routes** | D_id | S_address | E_address | Driver | | |
| **Delivary vehicles** | V_id | Model | Licence_no | year | | |
| **Goods services** | G_id | Dimension | Cost | Weight | | |
| **Mail** | Mail_id | S_name | R_name | M_type | D_status | Date |
| **Employee** | E_id | E_name | Cont_no | J_name | | |
| **Branch** | De_id | De_name | M_name | E_cnt | Budget | |
| **Service charge** | Sc_id | Sc_type | Sc_cost | Sc_loc | | |
| **Billing information** | B_date | B_id | B_address | method | Amount | |

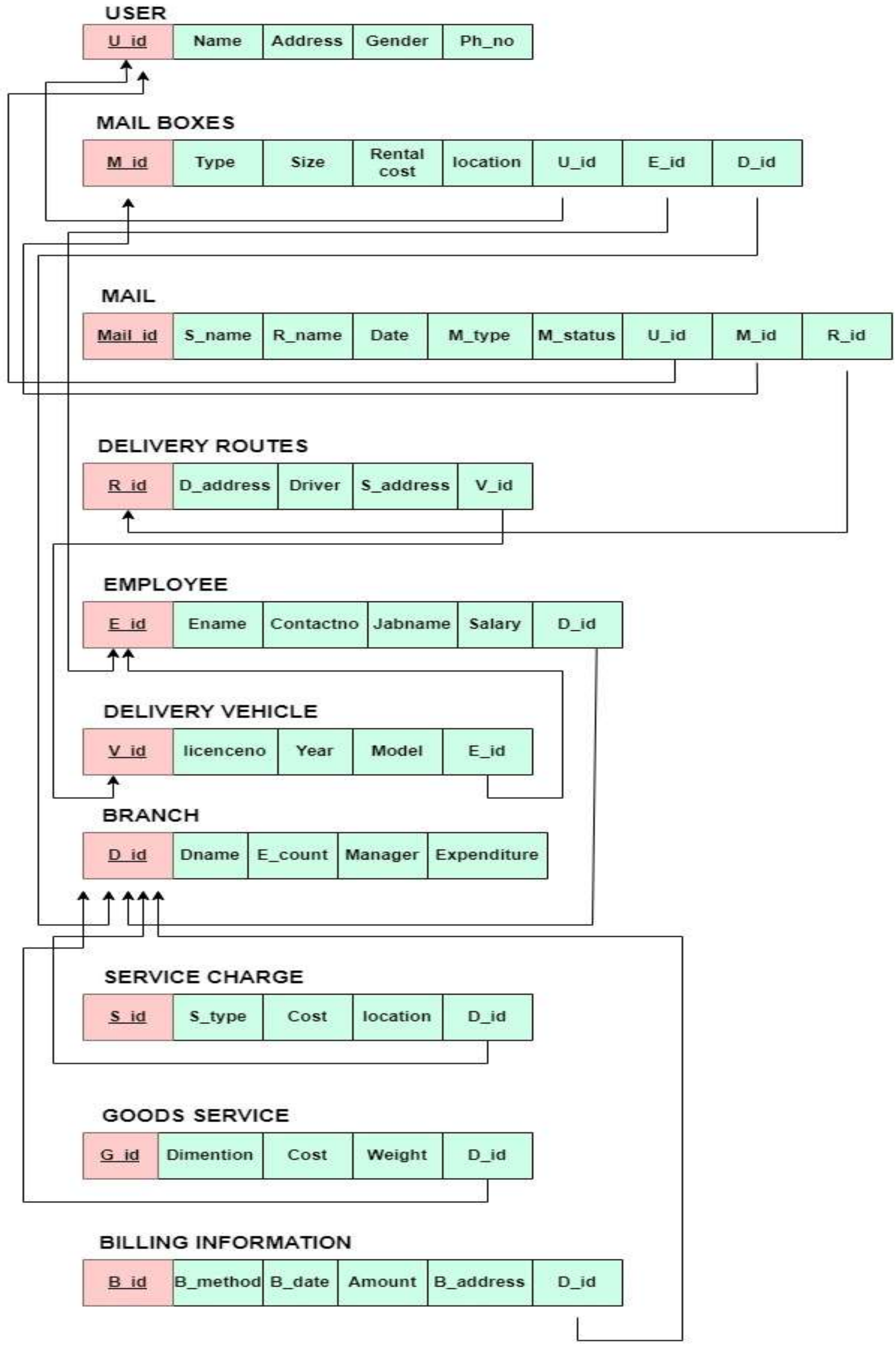
# Relationship between Entities and Cardinality Ratio:

| **Entity 1** | **Relation name** | **Entity 2** | **Cardinality ratio** |
|---|---|---|---|
| User | Send | Mail | 1:N |
| User | Place mail at | Mail boxes | 1:N |
| Mail boxes | Collect | Mail | 1:N |
| Mail boxes | Collect mail from | Employee | N:1 |
| Mail boxes | Belongs to | Branch | N:1 |
| Employee | Works at | Branch | 1:N |
| Mail | Transport through | Delivery routes | N:1 |
| Employee | Drives | Delivery vehicle | 1:1 |
| Delivery routes | Assigned by | Delivery vehicle | 1:1 |
| Delivery vehicle | Delivers | Goods services | 1:N |
| Branch | Stores | Billing information | 1:1 |
| Branch | Collect | Service charge | 1:N |

**Question 5:**Draw an Object Model illustrating the information you have identified in Question 4.



**Question 6:**Draw an ER to Relation Mapping illustrating the information you have identified in Question 4



USER

| U_id | Name | Address | Gender | Ph_no |
|---|---|---|---|---|

MAIL BOXES

| M_id | Type | Size | Rental cost | location | U_id | E_id | D_id |
|---|---|---|---|---|---|---|---|

MAIL

| Mail_id | S_name | R_name | Date | M_type | M_status | U_id | M_id | R_id |
|---|---|---|---|---|---|---|---|---|

DELIVERY ROUTES

| R_id | D_address | Driver | S_address | V_id |
|---|---|---|---|---|

EMPLOYEE

| E_id | Ename | Contactno | Jabname | Salary | D_id |
|---|---|---|---|---|---|

DELIVERY VEHICLE

| V_id | licenceno | Year | Model | E_id |
|---|---|---|---|---|

BRANCH

| D_id | Dname | E_count | Manager | Expenditure |
|---|---|---|---|---|

SERVICE CHARGE

| S_id | S_type | Cost | location | D_id |
|---|---|---|---|---|

GOODS SERVICE

| G_id | Dimention | Cost | Weight | D_id |
|---|---|---|---|---|

BILLING INFORMATION

| B_id | B_method | B_date | Amount | B_address | D_id |
|---|---|---|---|---|---|

**Question 7:**Draw a Data Dictionary illustrating the information you have identified in Question 6.

| **Objective (Entity)** | **Name (Attribute)** | **Type (Datatype)** | **Description** | **Primary key** | **Foreign key** |
|---|---|---|---|---|---|
| **User** | U_id | integer | Unique identification number for the user | Yes | No |
| | U_Name | Character | Name of the user | No | No |
| | Gender | Character | Gender of the user | No | No |
| | Address | Varchar | Address of the user | No | No |
| | Ph_no | integer | Phone number of the user | No | No |
| | | | | | |
| **Mailboxes** | M_id | Varchar | Unique identification id for the mail boxes | Yes | No |
| | Type | Character | Type of the mail boxes | No | No |
| | Size | Character | Type of the mail | No | No |
| | Rentalcost | integer | | No | No |
| | Location | Character | Location of mail boxes | No | No |
| | U_id | integer | Unique identification id for the user | No | Yes |
| | E_id | integer | Unique identification id for the employee | No | Yes |
| | D_id | Varchar | Unique identification of branch | No | Yes |

| | | | | | |
|---|---|---|---|---|---|
| **Mial** | Mail_id | Varchar | Unique identification id of mail | Yes | No |
| | S_name | Character | Name of the sender | No | No |
| | R_name | Character | Name of the Recipient | No | No |
| | Date | Date | Delivary time of mail | No | No |
| | M_type | Character | Type of the mail | No | No |
| | M_status | Character | Status of the mail | No | No |
| | U_id | integer | Unique identification id for the user | No | Yes |
| | M_id | Varchar | Unique identification id for the mail boxes | No | Yes |
| | R_id | Varchar | Unique identification id for the Delivery routes | No | Yes |
| | | | | | |
| **Delivery routes** | R_id | Varchar | Unique identification id for the Delivery routes | Yes | No |
| | D_address | Character | End address of the delivery | No | No |
| | Driver | Character | Name of the driver | No | No |
| | V_id | Varchar | Unique identification id for the Delivery vehicle | No | Yes |
| | S_address | Character | Starting address of the delivery | No | No |
| | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| **Employee** | E_id | integer | Unique identification id for the employee | Yes | No |
| | Ename | Character | Name of the employee | No | No |
| | Contactno | integer | Phone number of the employee | No | No |
| | Job name | Character | Job of the employee | No | No |
| | salary | integer | Salary of the employee | No | No |
| | D_id | Varchar | Unique identification of branch | No | Yes |
| | | | | | |
| **Delivery vehicle** | V_id | Varchar | Unique identification id for the Delivery vehicle | Yes | No |
| | Licence no | Varchar | Licence number of the vehicle | No | No |
| | Year | integer | Purchased year of the vehicle | No | No |
| | E_id | integer | Unique identification of employee | No | Yes |
| | Model | Varchar | Model of the vehicle | No | No |
| | | | | | |
| **Branch** | D_id | Integer | Unique identification of branch | Yes | No |
| | Dname | Character | Name of the branch | No | No |
| | E_count | Integer | Number of the employees | No | No |
| | Manager | Character | Name of the branch manager | No | No |
| | **Expenditure** | Integer | Total expenditure of the branch | No | No |

| | | | | | |
|---|---|---|---|---|---|
| **Service charge** | S_type | Character | Type of the service charge | No | No |
| | S_id | Varchar | Unique identification of service charge | Yes | No |
| | Cost | Integer | Total cost of service charge | No | No |
| | Location | Character | Locaton of  service charge | No | No |
| | D_id | Integer | Unique identification of branch | No | Yes |
| | | | | | |
| **Goods Services** | Cost | Integer | Cost of the Goods and Services | No | No |
| | Dimension | Integer | Dimensions of  Goods Services | No | No |
| | S_id | Varchar | Unique identification of Goods Services | Yes | No |
| | Weight | Integer | Total weight of Goods Services | No | No |
| | D_id | Integer | Unique identification of branch | No | Yes |
| | | | | | |
| **Billing information** | B_mathod | Character | Method of billing | No | No |
| | B_id | Varchar | Unique identification of Billing information | Yes | No |
| | B_date | Date | Date of billing | No | No |
| | Amount | Integer | Total billing amount | No | No |
| | B_address | Character | Billing address | No | No |
| | D_id | Integer | Unique identification of branch | No | Yes |

## Question 8:

**Normalization:**Are all the relations in your chosen schema in 3NF? Are they in BCNF? Explain your answers. If any of your relations are not in BCNF, normalize them to BCNF. If you choose to normalize your relations only till 2NF or 3NF, explain your reasons (e.g., the amount of redundancy introduced is limited or some other valid reason).

### USER

(U_id, Name , Address , Gender , Ph_no)

- The Relation is in 1NF as it has no attributes that can hold only multiple values. The attributes in the relation can hold only atomic values.
- The Relation is in 2NF as the primary key contains only one attribute. Hence, every attribute is fully functionally dependent on the key.
- We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.
- Similarly, it is in BCNF as there exists no non-key attribute that determines another nonkey attribute.

### MAIL BOXES

**(**M_id ,Type ,Size ,Rental_cost , loction ,U_id , E_id , D_id)

- The Relation is in 1NF as it has no attributes that can hold only multiple values. The attributes in the relation can hold only atomic values.

- The Relation is in 2NF as the primary key contains only one attribute. Hence, every attribute is fully functionally dependent on the key.
- We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.
- Similarly, it is in BCNF as there exists no non-key attribute that determines another nonkey attribute.

### MAIL

(Mail_id ,S_name , R_name ,Date ,M_type ,M_status ,U_id , M_id ,R_id )

- The Relation is in 1NF as it has no attributes that can hold only multiple values. The attributes in the relation can hold only atomic values.
- The Relation is in 2NF as the primary key contains only one attribute. Hence, every attribute is fully functionally dependent on the key.
- We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.
- Similarly, it is in BCNF as there exists no non-key attribute that determines another non key attribute.

### DELIVERY ROUTE

**(**R_id ,D_address ,Driver ,S_address ,V_id )

- The Relation is in 1NF as it has no attributes that can hold only multiple values. The attributes in the relation can hold only atomic values.

- The Relation is in 2NF as the primary key contains only one attribute. Hence, every attribute is fully functionally dependent on the key.
- We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.
- Similarly, it is in BCNF as there exists no non-key attribute that determines another nonkey attribute.

### EMPLOYEE

(E_id ,Ename,Contactno ,jobname,Salary ,D_id)

- The Relation is in 1NF as it has no attributes that can hold only multiple values. The attributes in the relation can hold only atomic values.
- The Relation is in 2NF as the primary key contains only one attribute. Hence, every attribute is fully functionally dependent on the key.
- We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.
- Similarly, it is in BCNF as there exists no non-key attribute that determines another nonkey attribute.

### DELIVERY VEHICLE

**(**V_id ,licenceno, year ,Model ,E_id)

- The Relation is in 1NF as it has no attributes that can hold only multiple values. The attributes in the relation can hold only atomic values.

- The Relation is in 2NF as the primary key contains only one attribute. Hence, every attribute is fully functionally dependent on the key.
- The relation is in 3NF to ensure that the table is in 3NF we need to eliminate any transitive dependencies .in this table the Model attribute is dependent only on V_id attribute , which is part of primary key hence there are no transitive dependencies.
- This relation is already in BCNF we need to ensure that all functional dependencies are determined by candidate key, in this table we have only one candidate key(V_id) and al attributes are fully dependent .
- since the table is already in 3NF and BCNF we need not split the table.

## BRANCH

**(**D_id ,Dname,E_count ,Manager ,Expenditure)

- The Relation is in 1NF as it has no attributes that can hold only multiple values. The attributes in the relation can hold only atomic values.
- The Relation is in 2NF as the primary key (D_id) contains only one attribute. Hence, every attribute is fully functionally dependent on the key.
- We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.
- Similarly, it is in BCNF as there exists no non-key attribute that determines another nonkey attribute.

## SERVICE CHARGE

(S_id ,S_type,Cost,D_id,location)

- The Relation is in 1NF as it has no attributes that can hold only multiple values. The attributes in the relation can hold only atomic values.
- The Relation is in 2NF as the primary key contains only one attribute. Hence, every attribute is fully functionally dependent on the key.
- The relation is in 3NF to ensure that the table is in 3NF we need to eliminate any transitive dependencies .in this table the all attributes are dependent only on S_id attribute, which is primary key hence there are no transitive dependencies.
- This relation is already in BCNF we need to ensure that all functional dependencies are determined by candidate key, in this table we have only one candidate key(S_id) and all attributes are fully dependent.
- since the table is already in 3NF and BCNF we need not split the table.

## GOODS SERVICE

(G_id , Dimention ,Cost,Weight,D_id)

- The Relation is in 1NF as it has no attributes that can hold only multiple values. The attributes in the relation can hold only atomic values.

- The Relation is in 2NF as the primary key (G_id) contains only one attribute. Hence, every attribute is fully functionally dependent on the key.
- We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

Similarly, it is in BCNF as there exists no non-key attribute that determines another nonkey attribute

## BILLING INFORMATION

**(**B_id,B_method,B_date,Amount,B_address,D_id)

- The Relation is in 1NF as it has no attributes that can hold only multiple values. The attributes in the relation can hold only atomic values.
- The Relation is in 2NF as the primary key contains only one attribute. Hence, every attribute is fully functionally dependent on the key.
- The relation is in 3NF to ensure that the table is in 3NF we need to eliminate any transitive dependencies .in this table the all attributes are dependent only on B_id attribute, which is primary key hence there are no transitive dependencies.
- This relation is already in BCNF we need to ensure that all functional dependencies are determined by candidate key, in this table we have only one candidate key(B_id) and all attributes are fully dependent .
- since the table is already in 3NF and BCNF we need not split the table.

**Question 10:** Choose the optimal normalized schema from Question 9 and justify why you think this is an optimal solution.

| | | | | | | |
|---|---|---|---|---|---|---|
| **User** | U_Name | U_id | Gender | Address | Ph_no | |
| **Mail boxes** | M_id | Rentalcost | MB_type | Size | loc | |
| **Delivery routes** | D_id | S_address | E_address | Driver | | |
| **Delivary vehicles** | V_id | Model | Licence_no | year | | |
| **Goods services** | G_id | Dimension | Cost | Weight | | |
| **Mail** | Mail_id | S_name | R_name | M_type | D_status | Date |
| **Employee** | E_id | E_name | Cont_no | J_name | | |
| **Branch** | De_id | De_name | M_name | E_cnt | Budget | |
| **Service charge** | Sc_id | Sc_type | Sc_cost | Sc_loc | | |
| **Billing information** | B_date | B_id | B_address | method | Amount | |

# Implementation Phase Report

## Creation of Tables:

### 1)Table Name:Userr

```
create table userr
(
u_id varchar(10),
u_name varchar(20),
gender varchar(10),
address varchar(20),
ph_no varchar(10),
primary key (u_id)
);
```

### 2)Table Name:branch

```
create table branch
(
d_id int,
dname varchar(20),
e_count int,
b_manager varchar(20),
expenditure int,
primary key(d_id)
);
```

### 3)Table Name:service_charge

```
create table service_charge
(
s_type char(10),
s_id varchar(20),
s_cost int,
s_location varchar(20),
d_id int,
primary key(s_id),
foreign key(d_id) references branch
);
```

## 4)Table Name:goods_service

```
create table goods_service
(
g_cost int,
dimensions int,
g_id varchar(20),
weight int,
d_id int,
primary key(g_id),
foreign key(d_id) references branch
);
```

## 5)Table Name:billing_info

```
create table billing_info
(
b_method varchar(20),
b_id varchar(20),
b_date date,
amount int,
b_address varchar(50),
d_id int,
primary key(b_id),
foreign key(d_id) references branch
);
```

## 6)Table Name:employee

```
create table employee
(
e_id int,
ename varchar(20),
contactno int,
job_name varchar(20),
salary int,
d_id int,
primary key(e_id),
foreign key(d_id) references branch
);
```

## 7)Table Name:d_vehicle

```
create table d_vehicle
(
v_id varchar(20),
licence_no varchar(20),
p_year int,
e_id int,
v_model varchar(20),
primary key(v_id),
foreign key(e_id) references employee
);
```

## 8)Table Name:d_routes

```
create table d_routes
(
r_id varchar(15),
d_address varchar(10),
driver char(5),
v_id varchar(20),
s_address varchar(20),
primary key(r_id),
foreign key(v_id) references d_vehicle
);
```

## 9)Table Name:mailbox

```
create table mailbox
(
m_id varchar(10),
u_id varchar(10),
e_id int,
d_id int,
mail_type varchar(15),
mail_size varchar(15),
rentalcost int,
m_location varchar(20),
primary key(m_id),
foreign key (u_id) references userr,
foreign key (e_id) references employee,
foreign key (d_id) references branch
);
```

## 10)Table Name:mail

```
create table mail
(
mail_id varchar(10),
s_name varchar(15),
r_name varchar(15),
d_date date,
m_type char(5),
m_status char(5),
u_id varchar(10),
m_id varchar(10),
r_id varchar(15),
primary key(mail_id),
foreign key (u_id) references userr,
foreign key (m_id) references mailbox,
foreign key (r_id) references d_routes
);
```

# Inserting Values in the Tables:

## 1)Table Name:Userr

```
insert into userr values ('U1','rajesh','male','BANGALORE',9875643092);
insert into userr values ('U2','priya','female','dharwad',9970646092);
insert into userr values ('U3','suresh','male','davangere',9545663071);
insert into userr values ('U4','abhay','male','tumkur',8875443583);
insert into userr values ('U5','pooja','female','bagalkot',8075663457);
```

## 2)Table Name:branch

```
insert into branch values ('1','branch A','50','ankit',100000);
insert into branch values ('2','branch B','75','vishal',150000);
insert into branch values ('3','branch C','60','prajwal',125000);
insert into branch values ('4','branch D','40','prasanna rao',200000);
insert into branch values ('5','branch E','70','preetham',175000);
```

## 3)Table Name:service_charge

```
insert into service_charge values ('post','KA01','50','karnataka',1);
insert into service_charge values ('mail','KA04','40','karnataka',2);
insert into service_charge values ('pop3','KA17','60','karnataka',3);
insert into service_charge values ('speedpost','KA25','100','karnataka',4);
insert into service_charge values ('post','KA54','50','karnataka',5);
```

## 4)Table Name:goods_service

```
insert into goods_service values ('100','10','G001','50',1);
insert into goods_service values ('200','20','G002','100',2);
insert into goods_service values ('150','15','G003','75',1);
insert into goods_service values ('300','25','G004','150',3);
insert into goods_service values ('175','18','G005','90',2);
insert into goods_service values ('250','22','G006','120',3);
```

## 5)Table Name:billing_info

```
insert into billing_info values ('creditcard','CC789012','18-01-2023',25,'hubli',5);
insert into billing_info values ('debitcard','DB2346212','15-01-2023',30,'bangalore',5);
insert into billing_info values ('googlepay','GP345678','02-01-2023',20,'davanagere',3);
insert into billing_info values ('cash','CS901234','27-12-2022',10,'tumkur',1);
insert into billing_info values ('check','Ck569090','31-12-2022',5,'bellary',2);
```

## 6)Table Name:employee

```
insert into employee values(101,'jagdish',9084655830,'Mail Sorting',10000,1);
insert into employee values(102,'suresh',9663357289, 'Package Delivery', 12500,5);
insert into employee values(103,'abhishek',9556427728,'Express Mail',15000,5);
insert into employee values(104,'ramesh',8908775464,'Bulk Mailing',16000,3);
insert into employee values(105,'mangesh',8876578900,'Certified Mail',8900,2);
insert into employee values(106,'dhanush',8764357884,'Postage Stamps',11000,1);
```

## 7)Table Name:d_vehicle

```
INSERT INTO d_vehicle VALUES ('V001', 'ABC123', 2020, 101, 'Sedan');
INSERT INTO d_vehicle VALUES ('V002', 'DEF456', 2018, 102, 'SUV');
INSERT INTO d_vehicle VALUES ('V003', 'GHI789', 2021, 103, 'Truck');
INSERT INTO d_vehicle VALUES ('V004', 'JKL012', 2019, 104, 'Coupe');
INSERT INTO d_vehicle VALUES ('V005', 'MNO345', 2022, 105, 'Hatchback');
INSERT INTO d_vehicle VALUES ('V006', 'PQR678', 2017, 102, 'Van');
INSERT INTO d_vehicle VALUES ('V007', 'YHO455', 2012, 105, 'mahindra');
```

## 8)Table Name:d_routes

```
INSERT INTO d_routes VALUES ('R01', 'HOSPETE', 'D1234', 'V001', 'TUMKURU');
INSERT INTO d_routes VALUES ('R02', 'BELLARY', 'D1234', 'V001', 'ARSIKERE');
INSERT INTO d_routes VALUES ('R03', 'GADAG', 'D6734', 'V003', 'DAVANGERE');
INSERT INTO d_routes VALUES ('R04', 'HUBBALLI', 'D6734', 'V003', 'SHIVAMOGGA');
INSERT INTO d_routes VALUES ('R05', 'BAGALKOT', 'D3077', 'V004', 'BIDAR');
INSERT INTO d_routes VALUES ('R06', 'RAICHUR', 'D6654', 'V002', 'KOLAR');
INSERT INTO d_routes VALUES ('R07', 'KARWAR', 'D5454', 'V006', 'MANGLORE');
INSERT INTO d_routes VALUES ('R08', 'BANGALORE', 'D5467', 'V007', 'HUBBALLI');
```

## 9)Table Name:MAILBOX.

```
INSERT INTO MAILBOX VALUES('M001', 'U1', 101, 1, 'Package', 'Large', 20, 'VIDYANAGAR,HUBLI');
INSERT INTO MAILBOX VALUES('M002', 'U2', 102, 2, 'Letter', 'Small', 50, 'KAIPETE,DAVANAGERE');
INSERT INTO MAILBOX VALUES('M003', 'U3', 103, 2, 'Package', 'Medium', 15, 'VIJAYNAGAR,BELLARY');
INSERT INTO MAILBOX VALUES('M004', 'U4', 104, 3, 'Letter', 'Large', 100, 'KESHAPUR,BANGALORE');
INSERT INTO MAILBOX VALUES('M005', 'U5', 105, 5, 'Package', 'Small',30, 'MG ROAD,BANGALORE');
INSERT INTO MAILBOX VALUES('M006', 'U5', 101, 4, 'Package', 'LARGE',150,'VIVEKNAGAR,BAGALKOT');
```

## 10)Table Name: MAIL

```
INSERT INTO MAIL VALUES('M101','MAHESH','PRANAV','12-01-2023','SPEED POST','R','U1','M001','R01');
INSERT INTO MAIL VALUES('M102','MANJU','CHETAN','01-01-2023','POSTCARD','NR','U2','M002','R02');
INSERT INTO MAIL VALUES('M103','NIDHI','ANJANA','25-01-2023','INLAND LETTER','R','U2','M002','R03');
INSERT INTO MAIL VALUES('M104','RAKSHITHA','ADITYA','14-02-2023','SPEED POST','R','U5','M0014','R03');
INSERT INTO MAIL VALUES('M105','RUTUJ','NISHCHAY','13-02-2023','ORDINARY POST','NR','U3','M002','R04');
INSERT INTO MAIL VALUES('M106','POOJA','POORNIMA','18-02-2023','SPEED POST','R','U4','M003','R01');
INSERT INTO MAIL VALUES('M107','AKASH','YASHANK','03-03-2023','PACKETS','R','U2','M005','R02');
```

# PL_SQL QUERIES

## 1)Retrieve the list of branches along with the count of employees working in each branch.

### Solution:

```
SELECT b.d_id, b.dname,
(SELECT COUNT(*)
FROM employee e
WHERE e.d_id = b.d_id)
AS employee_count
FROM branch b;
```

## 2)Retrieve the list of users who have rented mailboxes and the rental cost of their mailboxes.

### Solution:

```
SELECT u.u_id, u.u_name,
(SELECT SUM(m.rentalcost)
FROM mailbox m
WHERE m.u_id = u.u_id)
AS rental_cost
FROM userr u
INNER JOIN mailbox m ON m.u_id = u.u_id;
```

## 3)Retrieve the list of branches along with the total cost of goods services offered by each branch.

### Solution:

```
SELECT b.d_id, b.dname,
(SELECT SUM(g.g_cost)
FROM goods_service g
WHERE g.d_id = b.d_id)
AS total_cost
FROM branch b;
```

## 4)Retrieve the list of mailboxes and the total number of mails received by each mailbox.

### Solution:

```
SELECT m.m_id, m.u_id,
(SELECT COUNT(*)
FROM mail ml
WHERE ml.m_id = m.m_id)
AS mail_count
FROM mailbox m;
```

## 5)Retrieve the list of branches and the total expenditure of each branch as a percentage of the total expenditure of all branches.

### Solution:

```
SELECT b.d_id, b.dname,
(SELECT SUM(b1.expenditure) FROM branch b1) AS total_expenditure,
((b.expenditure / (SELECT SUM(b2.expenditure) FROM branch b2)) * 100) AS percentage
FROM branch b;
```

## Procedures

## 1)Create a procedure to insert a new record into the userr table.

## Solution:

```
CREATE OR REPLACE PROCEDURE add_user (
  p_id IN VARCHAR2,
  p_name IN VARCHAR2,
  p_gender IN VARCHAR2,
  p_address IN VARCHAR2,
  p_ph_no IN VARCHAR2
)
IS
BEGIN
  INSERT INTO userr (u_id, u_name, gender, address, ph_no)
  VALUES (p_id, p_name, p_gender, p_address, p_ph_no);
END;
```

## 2)Create a procedure to update the b_manager column for a specific branch in the branch table.

## Solution:

```
CREATE OR REPLACE PROCEDURE update_branch_manager (
  p_id IN INTEGER,
  p_manager IN VARCHAR2
)
IS
BEGIN
  UPDATE branch
  SET b_manager = p_manager
  WHERE d_id = p_id;
END;
```

## 3)Create a procedure to calculate the total cost of all services for a specific branch in the service_charge table.

### Solution:

```
CREATE OR REPLACE PROCEDURE total_service_cost (
  p_id IN INTEGER,
  p_total OUT INTEGER
)
IS
BEGIN
  SELECT SUM(s_cost)
  INTO p_total
  FROM service_charge
  WHERE d_id = p_id;
END;
```

## 4)Procedure to calculate the total expenditure for a particular branch in the branch table.

### Solution:

```
CREATE OR REPLACE PROCEDURE calculate_expenditure (
    in_did int,
    out_expenditure out int
)
IS
BEGIN
    SELECT expenditure INTO out_expenditure FROM branch WHERE d_id = in_did;
    dbms_output.put_line('Expenditure for branch ' || in_did || ' is ' || out_expenditure);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        dbms_output.put_line('Branch ID ' || in_did || ' not found.');
```

```
END;
```

## 5)Create a procedure to delete a record from the d_vehicle table and update the e_count column for the corresponding branch in the branch table.

### Solution:

```
CREATE OR REPLACE PROCEDURE delete_vehicle (
   vehicle_id IN d_vehicle.vehicle_id%TYPE
)
IS
   branch_id branch.branch_id%TYPE;
BEGIN
   -- Get the branch ID for the vehicle
   SELECT branch_id INTO branch_id FROM d_vehicle WHERE vehicle_id =
delete_vehicle.vehicle_id;

   -- Delete the vehicle record
   DELETE FROM d_vehicle WHERE vehicle_id = delete_vehicle.vehicle_id;

   -- Update the e_count column for the corresponding branch
   UPDATE branch SET e_count = e_count - 1 WHERE branch_id = branch_id;

   COMMIT;
   dbms_output.put_line('Vehicle deleted successfully');
EXCEPTION
   WHEN NO_DATA_FOUND THEN
      dbms_output.put_line('Vehicle not found');
   WHEN OTHERS THEN
      dbms_output.put_line('Error: ' || SQLERRM);
      ROLLBACK;
END;
/
BEGIN
   delete_vehicle(1234);
END;
```

## Cursors

### 1)Fetch all the rentals with their corresponding customer names and rental durations in days.

**Solution:**

```
DECLARE
  CURSOR rental_cur IS SELECT r.rental_id, c.cust_name, r.rental_duration
    FROM d_rental r JOIN d_customer c ON r.cust_id = c.cust_id;
  rental_rec rental_cur%ROWTYPE;
BEGIN
  OPEN rental_cur;
  LOOP
    FETCH rental_cur INTO rental_rec;
    EXIT WHEN rental_cur%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE(rental_rec.cust_name || ' rented a vehicle for ' || rental_rec.rental_duration || ' days (rental ID ' || rental_rec.rental_id || ')');
  END LOOP;
  CLOSE rental_cur;
END;
```

### 2)Fetch all the rentals with their corresponding customer names and vehicle makes.

**Solution:**

```
DECLARE
  CURSOR rental_cur IS SELECT r.rental_id, c.cust_name, v.vehicle_make
    FROM d_rental r JOIN d_customer c ON r.cust_id = c.cust_id
      JOIN d_vehicle v ON r.vehicle_id = v.vehicle_id;
  rental_rec rental_cur%ROWTYPE;
BEGIN
  OPEN rental_cur;
  LOOP
    FETCH rental_cur INTO rental_rec;
    EXIT WHEN rental_cur%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE(rental_rec.cust_name || ' rented a ' ||
rental_rec.vehicle_make || ' (rental ID ' || rental_rec.rental_id || ')');
  END LOOP;
  CLOSE rental_cur;
END;
```

## 3)Fetch all the vehicles with their corresponding branch names and addresses.

**Solution:**

```
DECLARE
  CURSOR vehicle_cur IS SELECT v.vehicle_id, v.vehicle_make, v.vehicle_model,
b.branch_name, b.branch_address
    FROM d_vehicle v JOIN branch b ON v.branch_id = b.branch_id;
  vehicle_rec vehicle_cur%ROWTYPE;
BEGIN
  OPEN vehicle_cur;
  LOOP
    FETCH vehicle_cur INTO vehicle_rec;
    EXIT WHEN vehicle_cur%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE(vehicle_rec.vehicle_make || ' ' ||
vehicle_rec.vehicle_model || ' is located at ' || vehicle_rec.branch_name || ' ('
|| vehicle_rec.branch_address || ')');
  END LOOP;
  CLOSE vehicle_cur;
END;
```

## 4)Fetch all the employee names and salaries from the e_employee table.

**Solution:**

```
DECLARE
  CURSOR emp_cur IS SELECT emp_name, emp_salary FROM e_employee;
  emp_rec emp_cur%ROWTYPE;
BEGIN
```

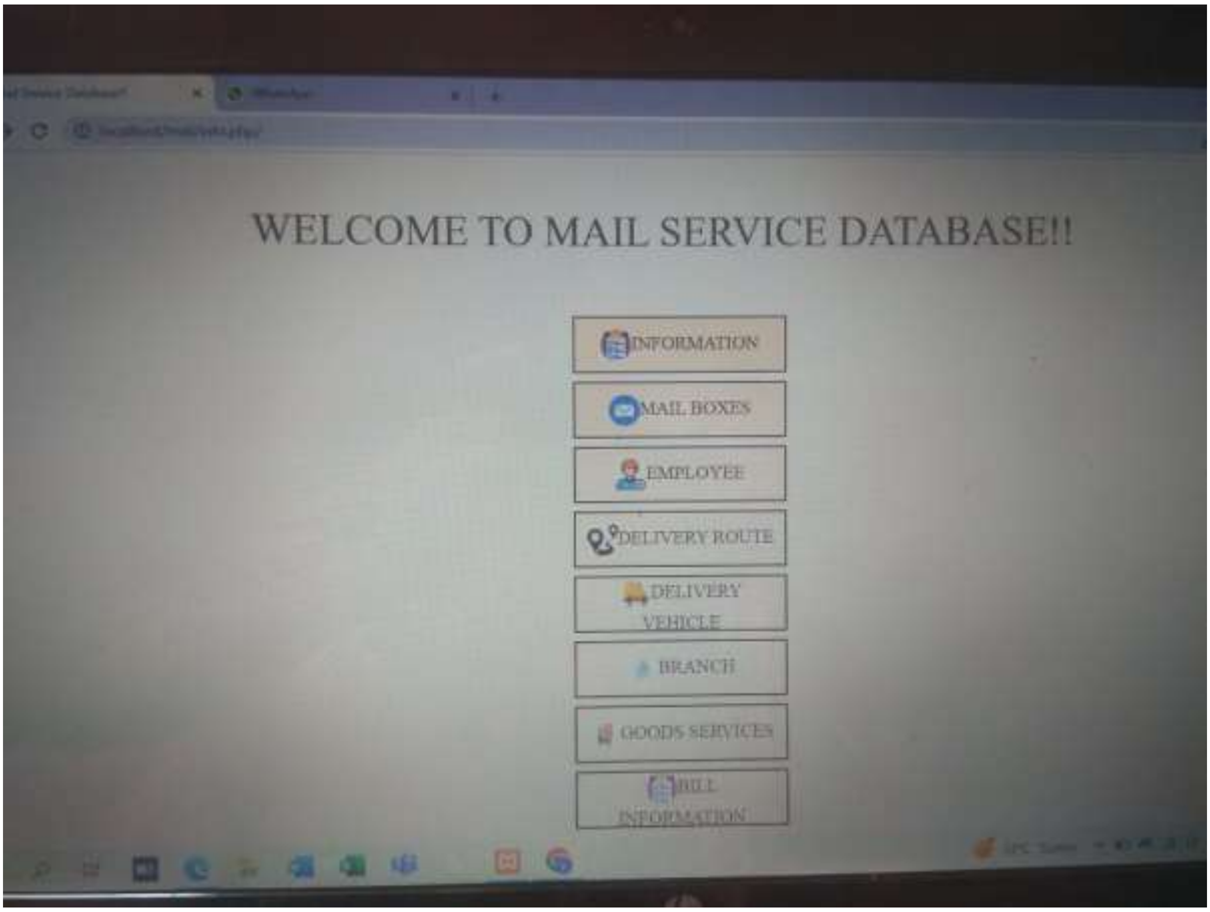
```
  OPEN emp_cur;
  LOOP
    FETCH emp_cur INTO emp_rec;
    EXIT WHEN emp_cur%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE(emp_rec.emp_name || ' makes ' ||
emp_rec.emp_salary || ' per year');
  END LOOP;
  CLOSE emp_cur;
END;
```

## 5)Fetch all the branch names and their addresses from the branch table.

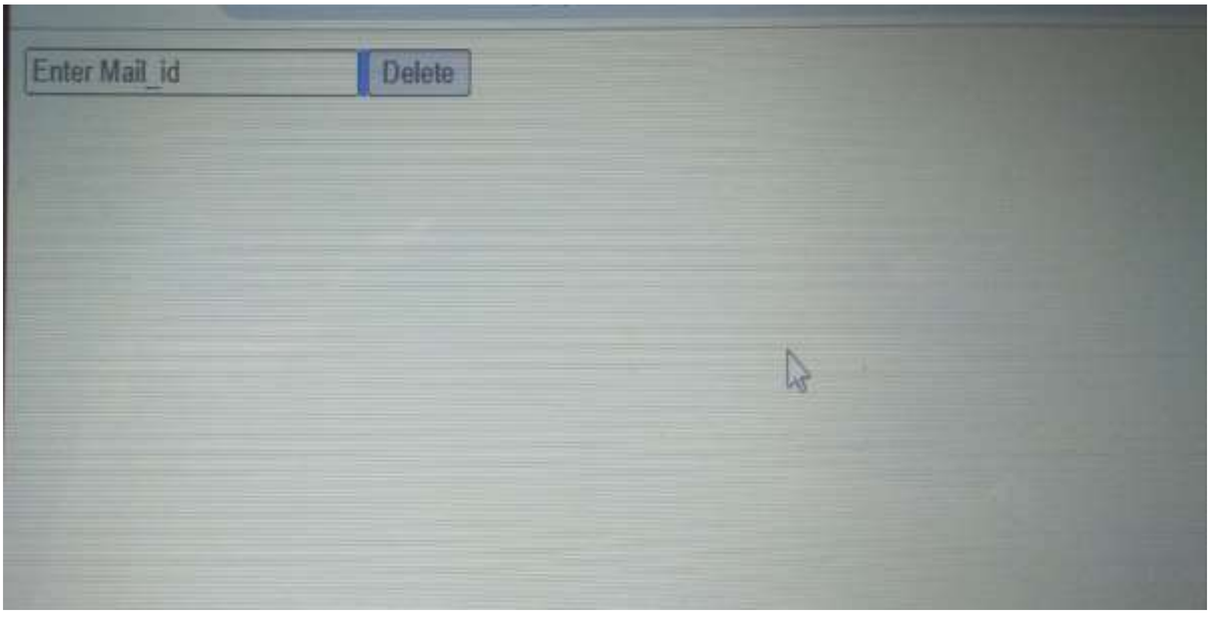
**Solution:**

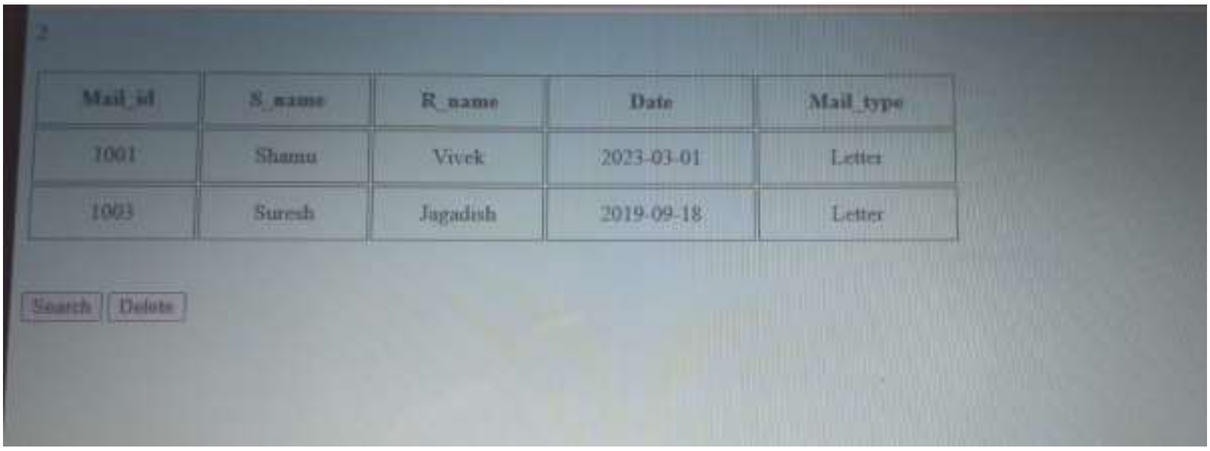
```
DECLARE
  CURSOR branch_cur IS SELECT branch_name, branch_address FROM branch;
  branch_rec branch_cur%ROWTYPE;
BEGIN
  OPEN branch_cur;
  LOOP
    FETCH branch_cur INTO branch_rec;
    EXIT WHEN branch_cur%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE(branch_rec.branch_name || ' is located at ' ||
branch_rec.branch_address);
  END LOOP;
  CLOSE branch_cur;
END;
```

# FRONTEND IMPLLEMENTATION



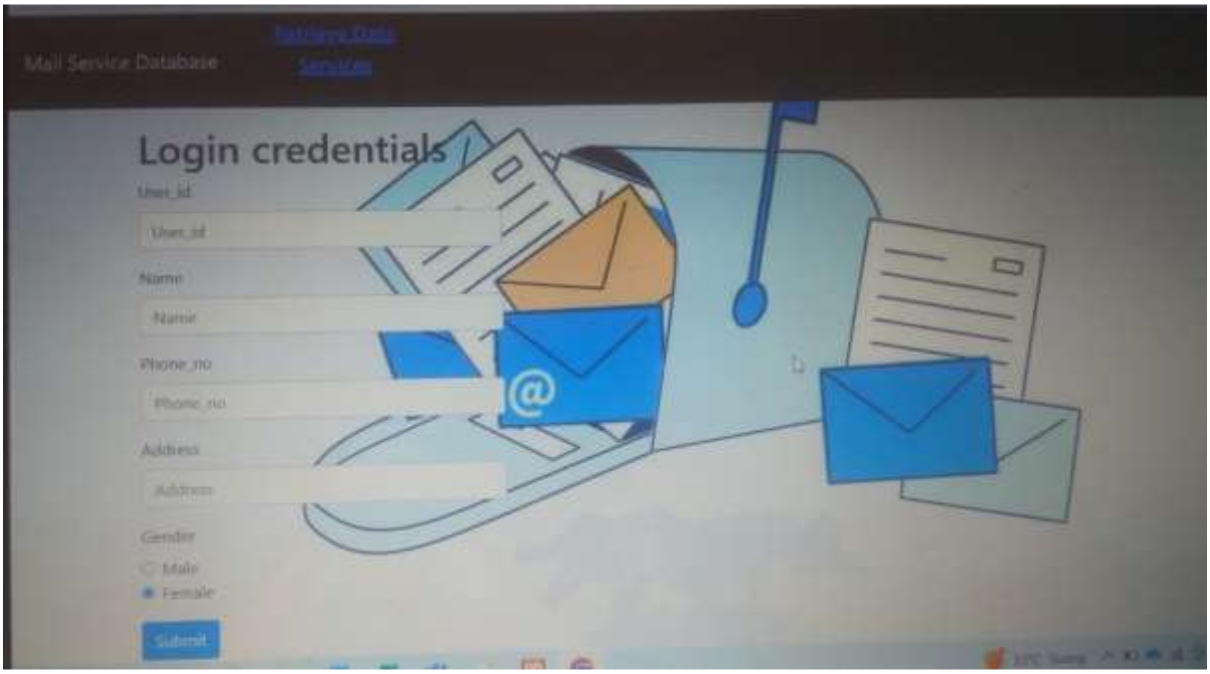
WELCOME TO MAIL SERVICE DATABASE!!





2

| Mail_id | S_name | R_name | Date | Mail_type |
|---|---|---|---|---|
| 1001 | Shamu | Vivek | 2023-03-01 | Letter |
| 1003 | Suresh | Jagadish | 2019-09-18 | Letter |



Login credentials

User_id

Name

Phone_no

Address

Gender

○ Male
● Female