# CS-355 Mini Project

Name: P. V. Sriram

Roll No. 1801CS37

As a part of the CS 355 Mini Project, I have designed a database management system for a package delivery company. This database system provides close to a realistic experience in the conceptual design, logical design, implementation, operation, and maintenance of a small relational database

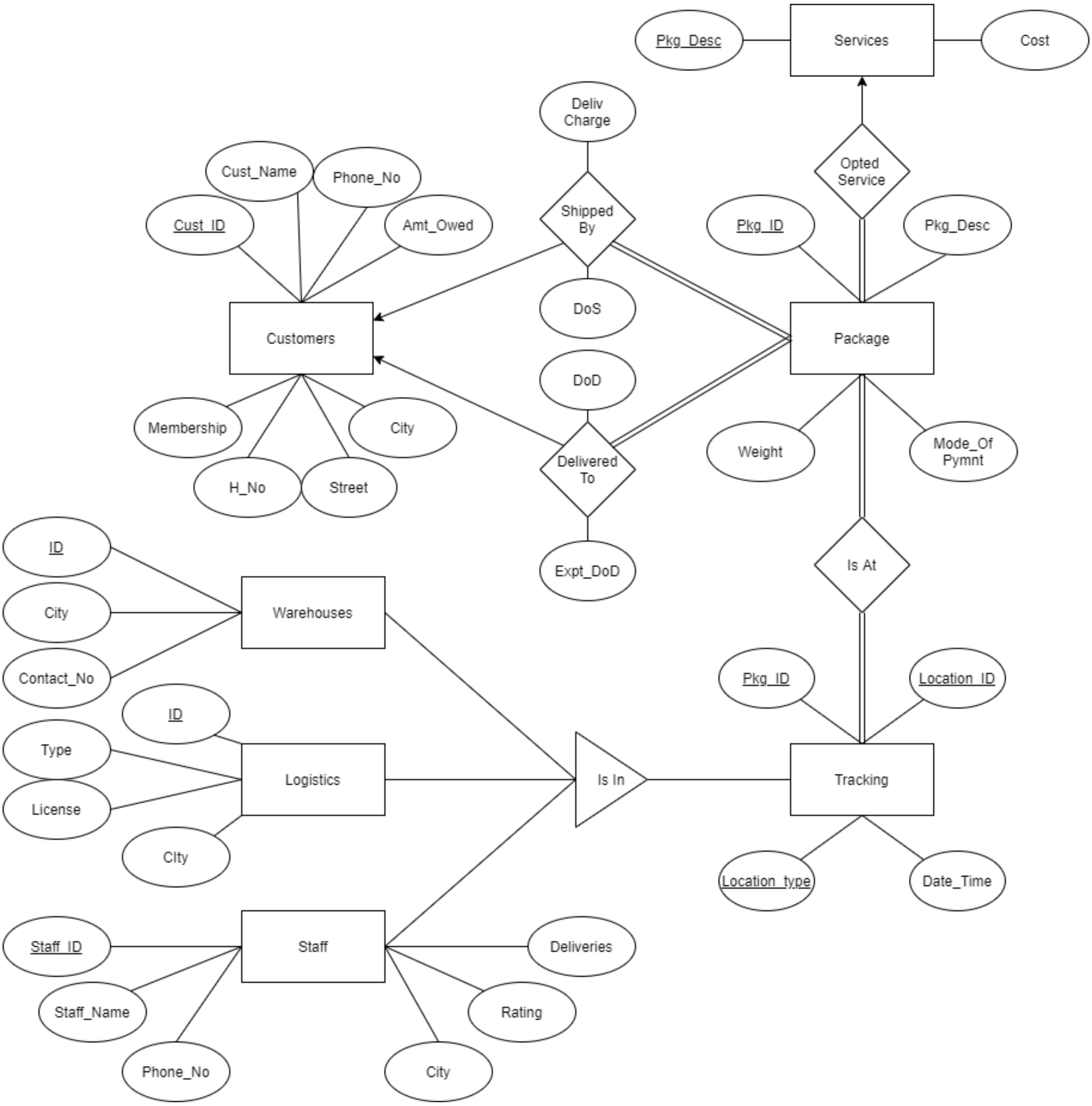
## Database

The database consists of the following tables:

1) Customers (Data of the customers who used the company's service)
2) Packages (Data about the packages shipped by the company)
3) Shipments (Data about the past and present shipment)
4) Tracking (Data about shipments throughout the whole process)
5) Logistics (Data about all the vehicles for intercity travel used by the company)
6) Warehouses (Data about the storage and warehouse facilities of the company)
7) Staff (Data about the delivery staff who work in the company)
8) Services (Data about the type of services offered by the company)

**For a detailed description of the database, please refer to readme.txt**

# E-R Diagram



### Structural Constraints of Relationships

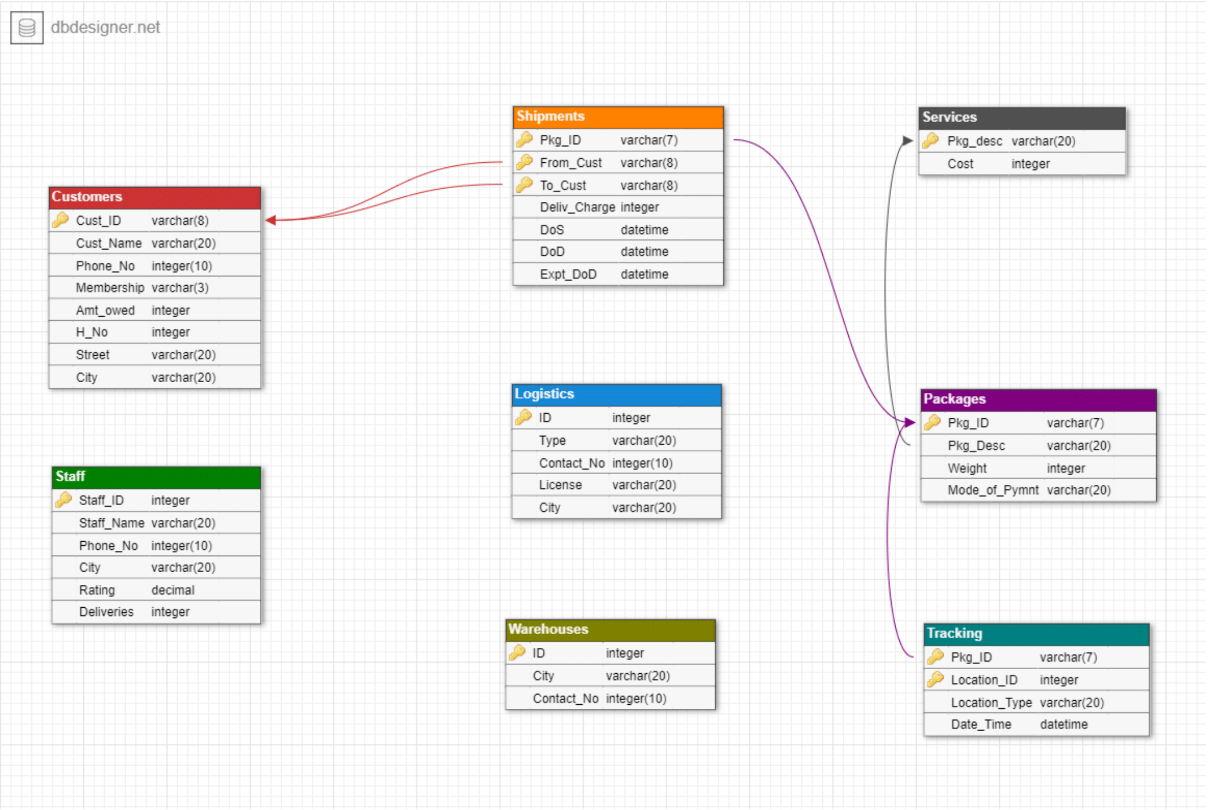
1) Customer to Package is a One-Many relationship, as a single customer can ship multiple packages but all the packages must be sent/received by only one customer. Customer is a partial participant as there could be a customer who hasn't used any service yet. Although packages are total participants as all packages are a part of a shipment and have to be definitely sent by someone.

2) Package to service is Many-One relationship, as a package is a part of a single service (opted by the customer) but there could be many packages which are a part of a service. Package is a total participant as the package should be a part of a service. Although, Service is a partial participant as all the services might not have been used by the packages.

3) Package to tracking is a Many-Many relationship, as throughout the shipment process, a package could be present at multiple locations, and also the locations can contain multiple packages at the same time. Both are total participants as every package has some location throughout the process and tracking table contains those locations where some or the other packages have been present.

4) Tracking table further contains sub groups. i.e. Warehouses, Logistics, Staff. Which essentially means that the package could either be present in the storage facilities (Warehouses) or vehicles (Logistics) or Delivery Agents (Staff).

## Relation Schema



| Customers | |
|---|---|
| 🔑 Cust_ID | varchar(8) |
| Cust_Name | varchar(20) |
| Phone_No | integer(10) |
| Membership | varchar(3) |
| Amt_owed | integer |
| H_No | integer |
| Street | varchar(20) |
| City | varchar(20) |

| Staff | |
|---|---|
| 🔑 Staff_ID | integer |
| Staff_Name | varchar(20) |
| Phone_No | integer(10) |
| City | varchar(20) |
| Rating | decimal |
| Deliveries | integer |

| Shipments | |
|---|---|
| 🔑 Pkg_ID | varchar(7) |
| 🔑 From_Cust | varchar(8) |
| 🔑 To_Cust | varchar(8) |
| Deliv_Charge | integer |
| DoS | datetime |
| DoD | datetime |
| Expt_DoD | datetime |

| Logistics | |
|---|---|
| 🔑 ID | integer |
| Type | varchar(20) |
| Contact_No | integer(10) |
| License | varchar(20) |
| City | varchar(20) |

| Warehouses | |
|---|---|
| 🔑 ID | integer |
| City | varchar(20) |
| Contact_No | integer(10) |

| Services | |
|---|---|
| 🔑 Pkg_desc | varchar(20) |
| Cost | integer |

| Packages | |
|---|---|
| 🔑 Pkg_ID | varchar(7) |
| Pkg_Desc | varchar(20) |
| Weight | integer |
| Mode_of_Pymnt | varchar(20) |

| Tracking | |
|---|---|
| 🔑 Pkg_ID | varchar(7) |
| 🔑 Location_ID | integer |
| Location_Type | varchar(20) |
| Date_Time | datetime |

### Foreign Keys

1) Shipments follows the foreign key constraint on Pkg_ID references to Packages (Pkg_ID). This is because, only the packages which are listed in the company's database are the ones to be shipped and therefore it has to be present in the Package database as well.

2) Shipments also follows the foreign key constraint on From_Cust and To_Cust referenced to Customers (Cust_ID). This is because, the company stores the data of all the customers who use the service, both sender and receiver. And therefore, all the customers related to a shipment are listed in customers' table.

3) Packages follows the foreign key constraint on Pkg_Desc referenced to Services (Pkg_desc). This is because each package is shipped through a particular plan offered by the company. Therefore, the Pkg_desc should lie in the services allowed by the company only for obvious reasons.

4) Tracking follows the foreign key constraint on Pkg_ID referenced to Packages (Pkg_ID). This is obviously true as the company is equipped and allowed only to track packages which are part of their shipment process only.

## Creating and Populating Database

### Customers

#### Create

```
create table Customers

(cust_id char(8),

Cust_name varchar(20),

Phone_No bigint,

Membership varchar(3),

Amt_owed int,

H_No int,

street varchar(20),

City varchar(20),

primary key(cust_id),
index idx_id (cust_id));
```

### Populate

```
Load data infile 'C:/ProgramData/MySQL/MySQL\ Server\ 8.0/Uploads/customers.csv' into table customers fields terminated by ',' enclosed by '"' lines terminated by '\r\n' ignore 1 rows;
```

## **Staff**

### Create

```
create table Staff

(Staff_ID int,

Staff_name varchar(20),

Phone_No bigint,

City varchar(20),

Rating decimal(3, 2),

deliveries int,

primary key(Staff_ID),

index idx_id (Staff_ID));
```

### Populate

```
Load data infile 'C:/ProgramData/MySQL/MySQL\ Server\ 8.0/Uploads/staff.csv' into table staff fields terminated by ',' enclosed by '"' lines terminated by '\r\n' ignore 1 rows;
```

## **Packages**

### Create

```
create table Packages

(pkg_id char(7),

pkg_desc varchar(20),

weight int,

mode_of_pymnt varchar(20),

primary key(pkg_id),
foreign key(Pkg_desc) references services(pkg_desc)

index idx_id (pkg_id));
```

### Populate

```
Load data infile 'C:/ProgramData/MySQL/MySQL\ Server\ 8.0/Uploads/packages.csv' into table packages fields terminated by ',' enclosed by '"' lines terminated by '\r\n' ignore 1 rows;
```

**Shipments**

### Create

```
create table shipments

(pkg_id char(7),

DoS datetime,

DoD datetime,

Expt_Dod datetime,

From_cust char(8),

To_cust char(8),

Deliv_Charge int,

primary key(pkg_id, To_cust, From_cust),

foreign key(pkg_id) references packages(pkg_id),

foreign key(From_cust) references customers(cust_id),

foreign key(To_cust) references customers(cust_id),

index idx_id (pkg_id));
```

### Populate

```
delimiter $$

create procedure populate_shipments()

begin

        declare count int;

        declare pkg_id char(7);

        declare from_cust int;
        declare to_cust int;

        declare charge int;

        declare dos datetime;

        declare dod datetime;

        declare expt_dod datetime;

        SET @MIN = '2020-01-29 00:53:27';

        SET @MAX = '2020-11-29 13:53:27';


        set count = 24;

        while(count >= 0) do

                select concat("Pkg-", count + 101) into pkg_id;

                select floor(rand() * 14 + 1) into from_cust;

                select floor(rand() * 11 + 15) into to_cust;

                select floor(rand() * 101) into charge;

                select TIMESTAMPADD(SECOND, FLOOR(RAND() * TIMESTAMPDIFF(SECOND,
@MIN, @MAX)), @MIN) into dos;

                select date_add(dos, interval floor(rand() * 10 + 1) day) into dod;

                select expt_deliv(pkg_id, dos) into expt_dod;

                insert into shipments values(pkg_id, from_cust, to_cust, charge, dos, dod,
expt_dod);

                set count = count - 1;

        end while;

end $$

delimiter ;
```

**Tracking**

### Create

```
create table Tracking

(Pkg_Id char(7),

Location_Id varchar(20),
Location_type varchar(20),

Date_time datetime

primary key(Pkg_Id, Location_Id),

index idx_id (pkg_id));
```

## Populate

```
delimiter $$

create procedure populate_tracking()

begin

        declare count int;

        declare pkg_id_ char(7);

        declare loc_id int;

        declare loc_typ varchar(20);

        declare date_time datetime;


        set count = 24;

        while(count >= 0) do

                select concat("Pkg-", count + 101) into pkg_id_;


                -- warehouse from

                select ID from warehouses where city in (select city from customers where cust_id in (select from_cust from shipments where pkg_id = pkg_id_)) into loc_id;


                set loc_typ = "Warehouse";


                select date_add((select dos from shipments where pkg_id = pkg_id_), interval 3 hour) into date_time;


                insert into tracking values(pkg_id_, loc_id, loc_typ, date_time);
```

```
            -- logistics from
            select ID from logistics where city in (select city from customers where cust_id in (select from_cust from shipments where pkg_id = pkg_id_)) into loc_id;

            select type from logistics where city in (select city from customers where cust_id in (select from_cust from shipments where pkg_id = pkg_id_)) into loc_typ;

            select date_add((select dos from shipments where pkg_id = pkg_id_), interval 10 hour) into date_time;

            insert into tracking values(pkg_id_, loc_id, loc_typ, date_time);

            -- warehouse to
            select ID from warehouses where city in (select city from customers where cust_id in (select to_cust from shipments where pkg_id = pkg_id_)) into loc_id;

            set loc_typ = "Warehouse";

            select date_add((select dod from shipments where pkg_id = pkg_id_), interval -5 hour) into date_time;

            insert into tracking values(pkg_id_, loc_id, loc_typ, date_time);

            -- delivery
            select staff_id from staff where city in (select city from customers where cust_id in (select to_cust from shipments where pkg_id = pkg_id_)) into loc_id;

            set loc_typ = "Out For Delivery";

            select date_add((select dod from shipments where pkg_id = pkg_id_), interval -2 hour) into date_time;
```

```
                insert into tracking values(pkg_id_, loc_id, loc_typ, date_time);

                set count = count - 1;
        end while;
end $$
delimiter ;
```

**Logistics**

## Create

```
 create table logistics
(ID int,
type varchar(20),
Contact_No bigint,
License varchar(20),
City varchar(20),
primary key(ID),
index idx_id (ID));
```

## Populate

```
Load data infile 'C:/ProgramData/MySQL/MySQL\ Server\ 8.0/Uploads/logistics.csv' into
table logistics fields terminated by ',' enclosed by '"' lines terminated by '\r\n' ignore 1 rows;
```

**Services**

## Create

```
create table services(
Pkg_desc varchar(20),
Cost int,
primary key(Pkg_desc),
index idx_desc (Pkg_desc));
```

### Populate

```
insert into services values

("Flat Envelope", 5),

("Large Box", 30),

("Small Box", 10),

("Medicines", 50),

("Food", 60),

("Electronics", 100),

("International", 500);
```

**Warehouses**

### Create

```
create table warehouses

(ID int,

City varchar(20),

Contact_No bigint,

primary key(ID),

index idx_id (ID));
```

### Populate

```
Load data infile 'C:/ProgramData/MySQL/MySQL\ Server\ 8.0/Uploads/warehouses.csv' into table warehouses fields terminated by ',' enclosed by '"' lines terminated by '\r\n' ignore 1 rows;
```

## Queries

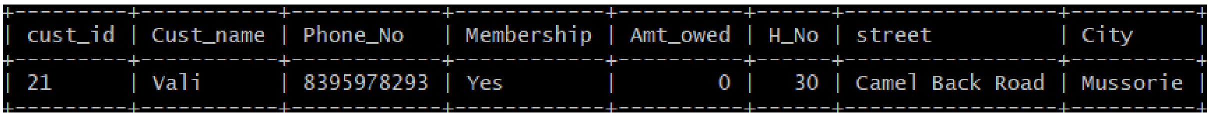
1) **Assume a delivery truck (say truck no 1721) is destroyed in a crash.**
   A) **Find all customers who had a package on that truck at the time of the crash.**
      **Query:**
      ```
      select * from customers where cust_id in (select from_cust from shipments where pkg_id in (select pkg_id from tracking where location_id = 1721 and location_type = "truck" and date_time > "2019-08-12 18:41:57"));
      ```
   B) **Find all recipients who had a package on that truck at the time of the crash.**
      **Query:**

select * from customers where cust_id in (select to_cust from shipments where pkg_id in (select pkg_id from tracking where location_id = 1721 and location_type = "truck" and date_time > "2019-08-12 18:41:57"));

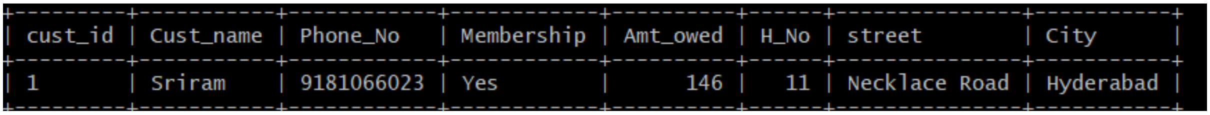
**C) Find the last successful delivery by that truck prior to the crash.**

**Query:**

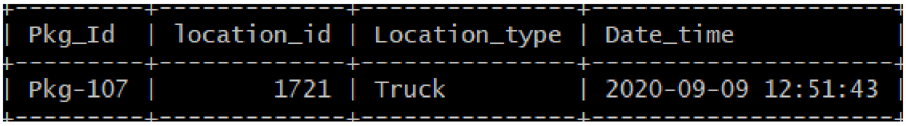
select * from tracking where location_id = 1721 and location_type = "truck" order by date_time and date_time is not null and date_time < "2019-08-12 18:41:57" desc limit 1;



```
+---------+-----------+------------+------------+----------+------+-----------------+----------+
| cust_id | Cust_name | Phone_No   | Membership | Amt_owed | H_No | street          | City     |
+---------+-----------+------------+------------+----------+------+-----------------+----------+
| 21      | Vali      | 8395978293 | Yes        |        0 |   30 | Camel Back Road | Mussorie |
+---------+-----------+------------+------------+----------+------+-----------------+----------+
```



```
+---------+-----------+------------+------------+----------+------+---------------+-----------+
| cust_id | Cust_name | Phone_No   | Membership | Amt_owed | H_No | street        | City      |
+---------+-----------+------------+------------+----------+------+---------------+-----------+
| 1       | Sriram    | 9181066023 | Yes        |      146 |   11 | Necklace Road | Hyderabad |
+---------+-----------+------------+------------+----------+------+---------------+-----------+
```



```
+---------+-------------+---------------+---------------------+
| Pkg_Id  | location_id | Location_type | Date_time           |
+---------+-------------+---------------+---------------------+
| Pkg-107 |        1721 | Truck         | 2020-09-09 12:51:43 |
+---------+-------------+---------------+---------------------+
```

2) **Find the customer who has shipped the most packages in the past year.**

**Query:**

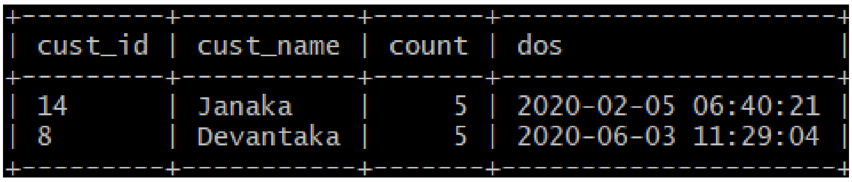
select cust_id, cust_name, count, dos from customers join

(select count(*) as count, from_cust, dos from shipments group by from_cust order by count(*) desc) d

on customers.cust_id

in (d.from_cust)

where count in (select max(count) from (select count(*)as count from shipments group by from_cust) x) and year(dos) = 2019;



```
+---------+-----------+-------+---------------------+
| cust_id | cust_name | count | dos                 |
+---------+-----------+-------+---------------------+
| 14      | Janaka    |     5 | 2020-02-05 06:40:21 |
| 8       | Devantaka |     5 | 2020-06-03 11:29:04 |
+---------+-----------+-------+---------------------+
```

3) **Find the customer who has spent the most money on shipping in the past year.**

**Query:**

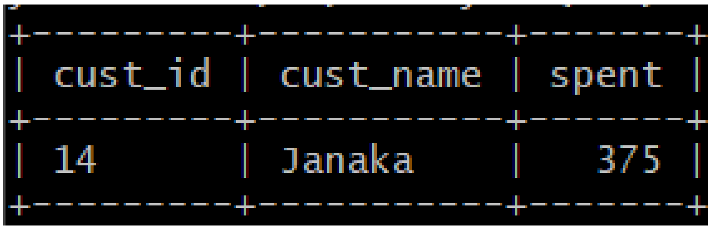
select cust_id, cust_name, spent from customers join

(select sum(deliv_charge) as spent, from_cust, dos from shipments group by from_cust order by spent desc) d

on customers.cust_id

in (d.from_cust)

where spent in (select max(spent) from (select sum(deliv_charge) as spent from shipments group by from_cust) x) and year(dos) = 2019;

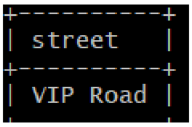


```
+----------+-----------+-------+
| cust_id  | cust_name | spent |
+----------+-----------+-------+
| 14       | Janaka    |   375 |
+----------+-----------+-------+
```

4) **Find the street with the most customers.**

**Query:**

```
select street from customers join
(select count(*) as count, cust_id from customers group by street order by count(*)
desc) d
on customers.cust_id
in (d.cust_id)
where count in (select max(count) from (select count(*)as count from customers
group by street) x);
```

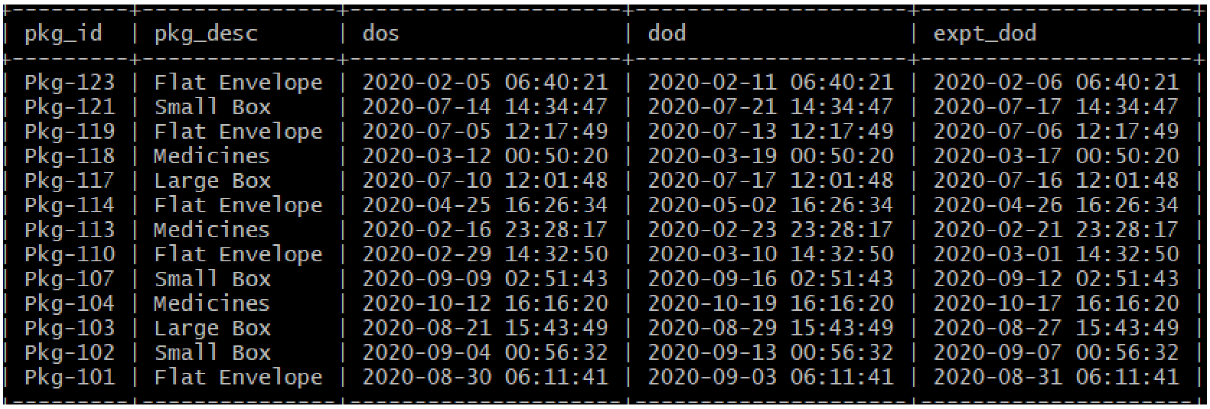


```
+----------+
| street   |
+----------+
| VIP Road |
```

5) **Find those packages that were not delivered within the promised time**

**Query:**

```
select packages.pkg_id, pkg_desc, dos, dod, expt_dod from packages inner join
shipments on packages.pkg_id = shipments.pkg_id where packages.pkg_id in (select
Pkg_id from shipments where dod > expt_dod);
```

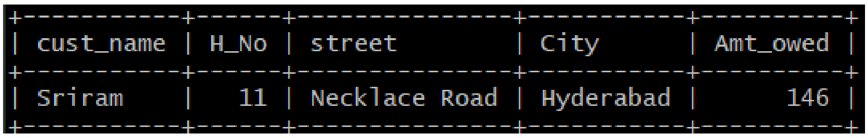


```
+---------+---------------+---------------------+---------------------+---------------------+
| pkg_id  | pkg_desc      | dos                 | dod                 | expt_dod            |
+---------+---------------+---------------------+---------------------+---------------------+
| Pkg-123 | Flat Envelope | 2020-02-05 06:40:21 | 2020-02-11 06:40:21 | 2020-02-06 06:40:21 |
| Pkg-121 | Small Box     | 2020-07-14 14:34:47 | 2020-07-21 14:34:47 | 2020-07-17 14:34:47 |
| Pkg-119 | Flat Envelope | 2020-07-05 12:17:49 | 2020-07-13 12:17:49 | 2020-07-06 12:17:49 |
| Pkg-118 | Medicines     | 2020-03-12 00:50:20 | 2020-03-19 00:50:20 | 2020-03-17 00:50:20 |
| Pkg-117 | Large Box     | 2020-07-10 12:01:48 | 2020-07-17 12:01:48 | 2020-07-16 12:01:48 |
| Pkg-114 | Flat Envelope | 2020-04-25 16:26:34 | 2020-05-02 16:26:34 | 2020-04-26 16:26:34 |
| Pkg-113 | Medicines     | 2020-02-16 23:28:17 | 2020-02-23 23:28:17 | 2020-02-21 23:28:17 |
| Pkg-110 | Flat Envelope | 2020-02-29 14:32:50 | 2020-03-10 14:32:50 | 2020-03-01 14:32:50 |
| Pkg-107 | Small Box     | 2020-09-09 02:51:43 | 2020-09-16 02:51:43 | 2020-09-12 02:51:43 |
| Pkg-104 | Medicines     | 2020-10-12 16:16:20 | 2020-10-19 16:16:20 | 2020-10-17 16:16:20 |
| Pkg-103 | Large Box     | 2020-08-21 15:43:49 | 2020-08-29 15:43:49 | 2020-08-27 15:43:49 |
| Pkg-102 | Small Box     | 2020-09-04 00:56:32 | 2020-09-13 00:56:32 | 2020-09-07 00:56:32 |
| Pkg-101 | Flat Envelope | 2020-08-30 06:11:41 | 2020-09-03 06:11:41 | 2020-08-31 06:11:41 |
+---------+---------------+---------------------+---------------------+---------------------+
```

6) **Take Customer ID and provide the details such as customer name, address, and amount owed**

**Query:**

```
select cust_name, H_No, street, City, Amt_owed from customers where cust_id = 1;
```

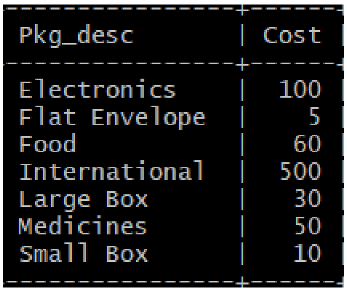


```
+-----------+------+---------------+-----------+----------+
| cust_name | H_No | street        | City      | Amt_owed |
+-----------+------+---------------+-----------+----------+
| Sriram    |   11 | Necklace Road | Hyderabad |      146 |
+-----------+------+---------------+-----------+----------+
```

7) **A bill listing charges by type of service**

**Query:**

select * from services;

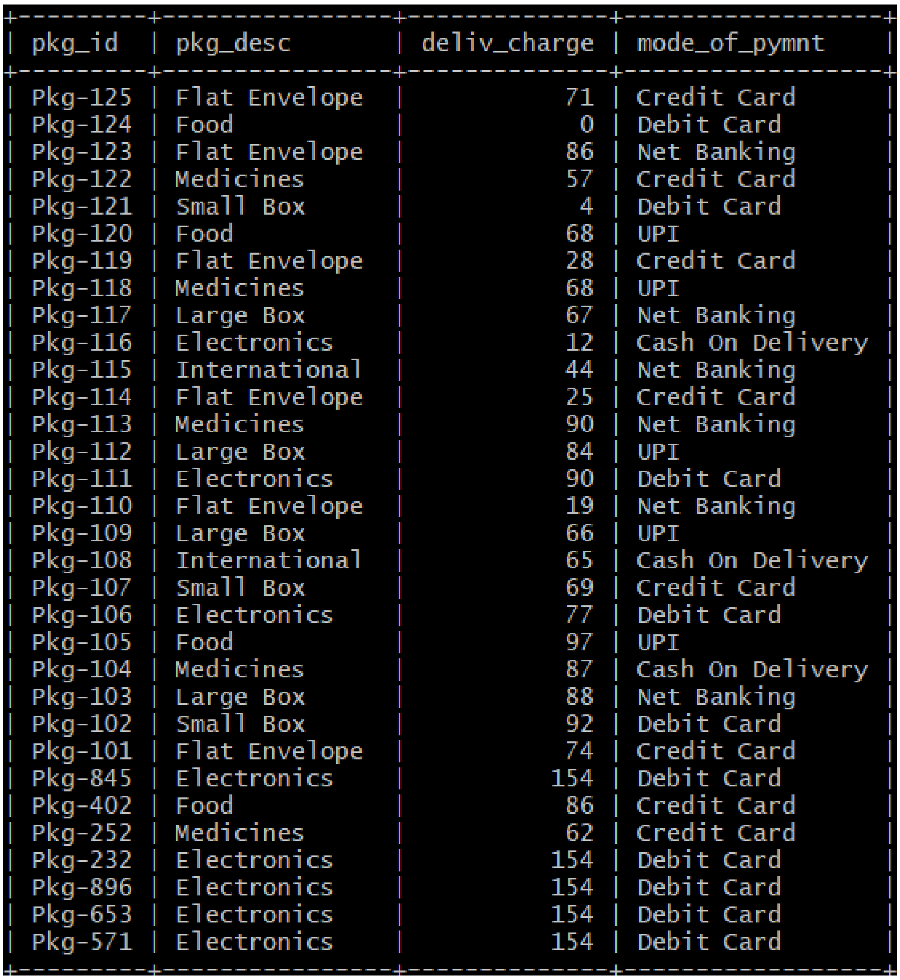


```
----------------+------
Pkg_desc        | Cost
----------------+------
Electronics     |  100
Flat Envelope   |    5
Food            |   60
International   |  500
Large Box       |   30
Medicines       |   50
Small Box       |   10
----------------+------
```

8) **An itemize billing listing each individual shipment and the charges for it.**

**Query:**

select shipments.pkg_id, pkg_desc, deliv_charge, mode_of_pymnt from shipments inner join packages on shipments.pkg_id = packages.pkg_id;



```
+---------+----------------+--------------+------------------+
| pkg_id  | pkg_desc       | deliv_charge | mode_of_pymnt    |
+---------+----------------+--------------+------------------+
| Pkg-125 | Flat Envelope  |           71 | Credit Card      |
| Pkg-124 | Food           |            0 | Debit Card       |
| Pkg-123 | Flat Envelope  |           86 | Net Banking      |
| Pkg-122 | Medicines      |           57 | Credit Card      |
| Pkg-121 | Small Box      |            4 | Debit Card       |
| Pkg-120 | Food           |           68 | UPI              |
| Pkg-119 | Flat Envelope  |           28 | Credit Card      |
| Pkg-118 | Medicines      |           68 | UPI              |
| Pkg-117 | Large Box      |           67 | Net Banking      |
| Pkg-116 | Electronics    |           12 | Cash On Delivery |
| Pkg-115 | International  |           44 | Net Banking      |
| Pkg-114 | Flat Envelope  |           25 | Credit Card      |
| Pkg-113 | Medicines      |           90 | Net Banking      |
| Pkg-112 | Large Box      |           84 | UPI              |
| Pkg-111 | Electronics    |           90 | Debit Card       |
| Pkg-110 | Flat Envelope  |           19 | Net Banking      |
| Pkg-109 | Large Box      |           66 | UPI              |
| Pkg-108 | International  |           65 | Cash On Delivery |
| Pkg-107 | Small Box      |           69 | Credit Card      |
| Pkg-106 | Electronics    |           77 | Debit Card       |
| Pkg-105 | Food           |           97 | UPI              |
| Pkg-104 | Medicines      |           87 | Cash On Delivery |
| Pkg-103 | Large Box      |           88 | Net Banking      |
| Pkg-102 | Small Box      |           92 | Debit Card       |
| Pkg-101 | Flat Envelope  |           74 | Credit Card      |
| Pkg-845 | Electronics    |          154 | Debit Card       |
| Pkg-402 | Food           |           86 | Credit Card      |
| Pkg-252 | Medicines      |           62 | Credit Card      |
| Pkg-232 | Electronics    |          154 | Debit Card       |
| Pkg-896 | Electronics    |          154 | Debit Card       |
| Pkg-653 | Electronics    |          154 | Debit Card       |
| Pkg-571 | Electronics    |          154 | Debit Card       |
+---------+----------------+--------------+------------------+
```

## Extra Features

1) **Procedure to Track a shipment**

I have additionally implemented an intuitive way to track a shipment using the package ID.

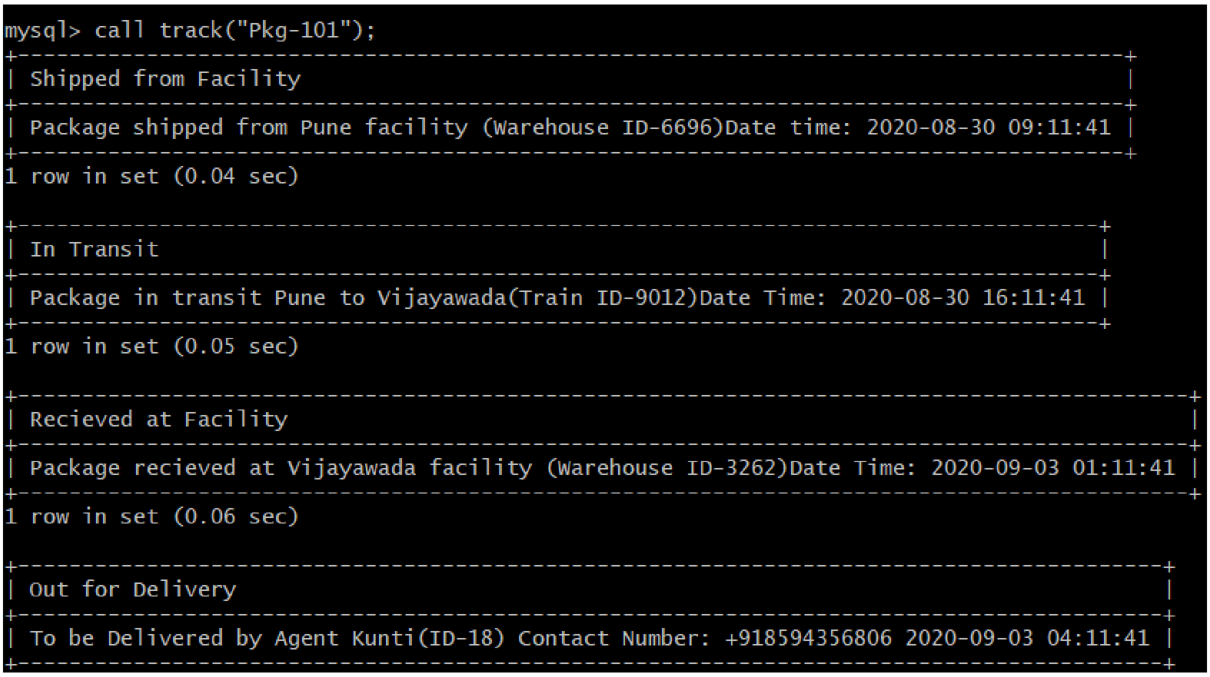
## Procedure:

```
delimiter $$

create procedure track(in pkg_id_ char(7))

begin

    declare warehouse_from int;

    declare warehouse_to int;

    declare transit int;

    declare vehicle varchar(20);

    declare from_city varchar(20);

    declare to_city varchar(20);

    declare delivery_name varchar(20);

    declare delivery_id int;

    declare delivery_phn bigint;

    declare temp datetime;


    select location_id from tracking where pkg_id = pkg_id_ and location_type =
"Warehouse" order by date_time limit 0,1 into warehouse_from;


    select city from warehouses where ID = warehouse_from into from_city;


    select location_id from tracking where pkg_id = pkg_id_ order by date_time limit 1,1
into transit;


    select location_type from tracking where pkg_id = pkg_id_ order by date_time limit
1,1 into vehicle;


    select location_id from tracking where pkg_id = pkg_id_ and location_type =
"Warehouse" order by date_time limit 1,2 into warehouse_to;


    select city from warehouses where ID = warehouse_to into to_city;
    select staff_name from staff where staff_id in (select location_id from tracking
where pkg_id = pkg_id_ and location_type = "Out For Delivery") into delivery_name;

    select staff_ID from staff where staff_id in (select location_id from tracking where
pkg_id = pkg_id_ and location_type = "Out For Delivery") into delivery_id;

    select Phone_no from staff where staff_id in (select location_id from tracking where
pkg_id = pkg_id_ and location_type = "Out For Delivery") into delivery_phn;

    select date_time from tracking where pkg_id = pkg_id_ and location_type =
"Warehouse" order by date_time limit 0,1 into temp;

    select concat("Package shipped from ", from_city, " facility (Warehouse ID-",
warehouse_from, ")Date time: ", temp) as "Shipped from Facility";

    select date_time from tracking where pkg_id = pkg_id_ order by date_time limit 1,1
into temp;

    select concat("Package in transit ", from_city, " to ", to_city, "(", vehicle," ID-",
transit, ")Date Time: ", temp) as "In Transit";

    select date_time from tracking where pkg_id = pkg_id_ and location_type =
"Warehouse" order by date_time limit 1,2 into temp;

    select concat("Package recieved at ", to_city, " facility (Warehouse ID-",
warehouse_to, ")Date Time: ", temp) as "Recieved at Facility";

    select date_time from tracking where pkg_id = pkg_id_ and location_type = "Out For
Delivery" into temp;

    select concat("To be Delivered by Agent ", delivery_name,

    "(ID-", delivery_id, ")",

    " Contact Number: +91", delivery_phn, " ", temp) as "Out for Delivery";
```

end $$

delimiter ;

## Output:



```
mysql> call track("Pkg-101");
+--------------------------------------------------------------------------------------+
| Shipped from Facility                                                                |
+--------------------------------------------------------------------------------------+
| Package shipped from Pune facility (Warehouse ID-6696)Date time: 2020-08-30 09:11:41 |
+--------------------------------------------------------------------------------------+
1 row in set (0.04 sec)

+------------------------------------------------------------------------------------+
| In Transit                                                                         |
+------------------------------------------------------------------------------------+
| Package in transit Pune to Vijayawada(Train ID-9012)Date Time: 2020-08-30 16:11:41 |
+------------------------------------------------------------------------------------+
1 row in set (0.05 sec)

+-------------------------------------------------------------------------------------------+
| Recieved at Facility                                                                      |
+-------------------------------------------------------------------------------------------+
| Package recieved at Vijayawada facility (Warehouse ID-3262)Date Time: 2020-09-03 01:11:41 |
+-------------------------------------------------------------------------------------------+
1 row in set (0.06 sec)

+-----------------------------------------------------------------------------------------+
| Out for Delivery                                                                        |
+-----------------------------------------------------------------------------------------+
| To be Delivered by Agent Kunti(ID-18) Contact Number: +918594356806 2020-09-03 04:11:41 |
+-----------------------------------------------------------------------------------------+
```

2) **Function to calculate an expected delivery date based on package description weight etc.**

## Function:

```
delimiter $$

create function expt_deliv(pkg_id_ char(7), service varchar(20), dos datetime)

returns datetime

deterministic

begin

    declare time_gap int;

    declare temp datetime;

    declare desc_ varchar(20);

    select pkg_desc from packages where pkg_id = pkg_id_ into desc_;

    case

            when desc_ = "Flat Envelope" and service = "Express" then set time_gap = 1;
            when desc_ = "Flat Envelope" and service = "Standard" then set time_gap = 3;
            when desc_ = "Small Box" and service = "Express" then set time_gap = 3;
            when desc_ = "Small Box" and service = "Standard" then set time_gap = 6;
            when desc_ = "Large Box" and service = "Express" then set time_gap = 6;
            when desc_ = "Large Box" and service = "Standard" then set time_gap = 9;
            when desc_ = "Medicines" and service = "Express" then set time_gap = 5;
            when desc_ = "Medicines" and service = "Standard" then set time_gap = 7;
            when desc_ = "Food" and service = "Express" then set time_gap = 4;
            when desc_ = "Food" and service = "Standard" then set time_gap = 5;
            when desc_ = "Electronics" and service = "Express" then set time_gap = 7;
            when desc_ = "Electronics" and service = "Standard" then set time_gap = 10;
            when desc_ = "International " and service = "Express" then set time_gap = 9;
            when desc_ = "International " and service = "Standard" then set time_gap = 12;
    end case;
    select date_add(dos, interval time_gap day) into temp;
    return(temp);
end $$
delimiter ;
```

**3) Function to calculate the Delivery charge, based on package description weight, express or standard delivery.**

### Function:

```
delimiter $$
create function charge(pkg_desc_ varchar(20), service varchar(20), weight int)
returns int
deterministic
begin
    declare charge int;
    select cost from services where pkg_desc = pkg_desc_ into charge;
    if service = "Express" then
            set charge = charge + weight * 0.2;
            set charge = charge + 0.15 * charge;
    else
            set charge  = charge + weight * 0.1;
    end if;
    set charge = charge + 0.2 * charge;
    return(charge);
end $$
delimiter ;
```

**4) Procedures to add a new row into shipment (which automatically adds into package and tracking table)**

Procedure:

```
delimiter $$
create procedure add_shipment(in from_cust char(8),in to_cust char(8), in dos datetime, in pkg_desc varchar(20), in weight int, in mode_of_pymnt varchar(20), service varchar(20))
begin
    declare pkg_id_ char(7);
    declare temp int;
    declare count_ int;
    declare expt_dod datetime;
    declare cost int;
    declare loc_id int;
    declare loc_typ varchar(20);
    declare date_time datetime;

    select floor(rand()*(999-125)) + 125 into temp;
    set pkg_id_ = concat("Pkg-",temp);
    select count(*) from packages where pkg_id = pkg_id_ into count_;
    while count_ > 0 DO
            select floor(rand()*(999-125)) + 125 into temp;
            set pkg_id_ = concat("Pkg-",temp);
            select count(*) from packages where pkg_id = pkg_id_ into count_;
    end while;

    insert into packages values(pkg_id_, pkg_desc, weight, mode_of_pymnt);

    select charge(pkg_desc, service, weight) into cost;
    select expt_deliv(pkg_id_, service, dos) into expt_dod;

    insert into shipments values(pkg_id_, from_cust, to_cust, cost, dos, NULL, expt_dod);

    select ID from warehouses where city in (select city from customers where cust_id = from_cust) order by rand() limit 1 into loc_id;

    set loc_typ = "Warehouse";

    select date_add(dos, interval 3 hour) into date_time;

    insert into tracking values(pkg_id_, loc_id, loc_typ, date_time);

    -- logistics from
    select ID from logistics where city in (select city from customers where cust_id = from_cust) order by rand() limit 1 into loc_id;

    select type from logistics where ID = loc_id into loc_typ;
    insert into tracking values(pkg_id_, loc_id, loc_typ, NULL);


    -- warehouse to
    select ID from warehouses where city in (select city from customers where cust_id = to_cust) into loc_id;


    set loc_typ = "Warehouse";


    insert into tracking values(pkg_id_, loc_id, loc_typ, NULL);


    -- delivery
    select staff_id from staff where city in (select city from customers where cust_id = to_cust) into loc_id;


    set loc_typ = "Out For Delivery";


    insert into tracking values(pkg_id_, loc_id, loc_typ, NULL);
end $$
delimiter ;
```

**5) Procedure to add new rows into rest of the tables**

## Procedure:

```
// Add logistics
delimiter $$
create procedure add_logistics(in type varchar(20), in Phn_No bigint, in lic varchar(20), in city varchar(20))
begin
    declare log_id int;
    declare count_ int;
    select floor(rand()*(9999-1000)) + 1000 into log_id;
    select count(*) from logistics where ID = log_id into count_;
    while count_ > 0 DO
            select floor(rand()*(9999-1000)) + 1000 into log_id;
            select count(*) from logistics where ID = log_id into count_;
    end while;
    insert into logistics values(log_id, type, Phn_No, lic, city);
end $$
delimiter ;

// Add Warehouse
delimiter $$
create procedure add_warehouses(in city varchar(20),in Phn_No bigint)
begin
    declare wr_id int;
    declare count_ int;
    select floor(rand()*(9999-1000)) + 1000 into wr_id;
    select count(*) from warehouses where ID = wr_id into count_;
    while count_ > 0 DO
            select floor(rand()*(9999-1000)) + 1000 into wr_id;
            select count(*) from warehouses where ID = wr_id into count_;
    end while;
    insert into warehouses values(wr_id, city, Phn_No);
end $$
delimiter ;

// Add staff
delimiter $$
create procedure add_staff(in name varchar(20), in Phn_No bigint,in city varchar(20))
```

```
begin
    declare stf_id int;
    declare count_ int;
    select floor(rand()*(100-25)) + 25 into stf_id;
    select count(*) from staff where staff_ID = stf_id into count_;
    while count_ > 0 DO
            select floor(rand()*(100-25)) + 25 into stf_id;
            select count(*) from staff where staff_ID = stf_id into count_;
    end while;
    insert into staff values(stf_id, name, Phn_No, city, 0, 0);
end $$
delimiter ;
```

**6) Procedure to update the tracking table for a particular package**

### Procedure:

```
delimiter $$
create procedure update_track(in pkg_id_ char(7), in date_time_ datetime)
begin
    update tracking set date_time = date_time_ where pkg_id = pkg_id_ and date_time
is NULL limit 1;
end$$
delimiter ;
```

## Storing data into CSV files

```
select * into outfile 'C:/ProgramData/MySQL/MySQL\ Server\
8.0/Uploads/customers_data.csv' fields terminated by ',' enclosed by '"' lines terminated
by '\r\n' from customers;


select * into outfile 'C:/ProgramData/MySQL/MySQL\ Server\
8.0/Uploads/logistics_data.csv' fields terminated by ',' enclosed by '"' lines terminated
by '\r\n' from logistics;
```

```
select * into outfile 'C:/ProgramData/MySQL/MySQL\ Server\
8.0/Uploads/packages_data.csv' fields terminated by ',' enclosed by '"' lines terminated
by '\r\n' from packages;


select * into outfile 'C:/ProgramData/MySQL/MySQL\ Server\
8.0/Uploads/services_data.csv' fields terminated by ',' enclosed by '"' lines terminated
by '\r\n' from services;


select * into outfile 'C:/ProgramData/MySQL/MySQL\ Server\
8.0/Uploads/shipments_data.csv' fields terminated by ',' enclosed by '"' lines terminated
by '\r\n' from shipments;


select * into outfile 'C:/ProgramData/MySQL/MySQL\ Server\
8.0/Uploads/staff_data.csv' fields terminated by ',' enclosed by '"' lines terminated by
'\r\n' from staff;


select * into outfile 'C:/ProgramData/MySQL/MySQL\ Server\
8.0/Uploads/tracking_data.csv' fields terminated by ',' enclosed by '"' lines terminated
by '\r\n' from tracking;


select * into outfile 'C:/ProgramData/MySQL/MySQL\ Server\
8.0/Uploads/warehouses_data.csv' fields terminated by ',' enclosed by '"' lines
terminated by '\r\n' from warehouses;
```