**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

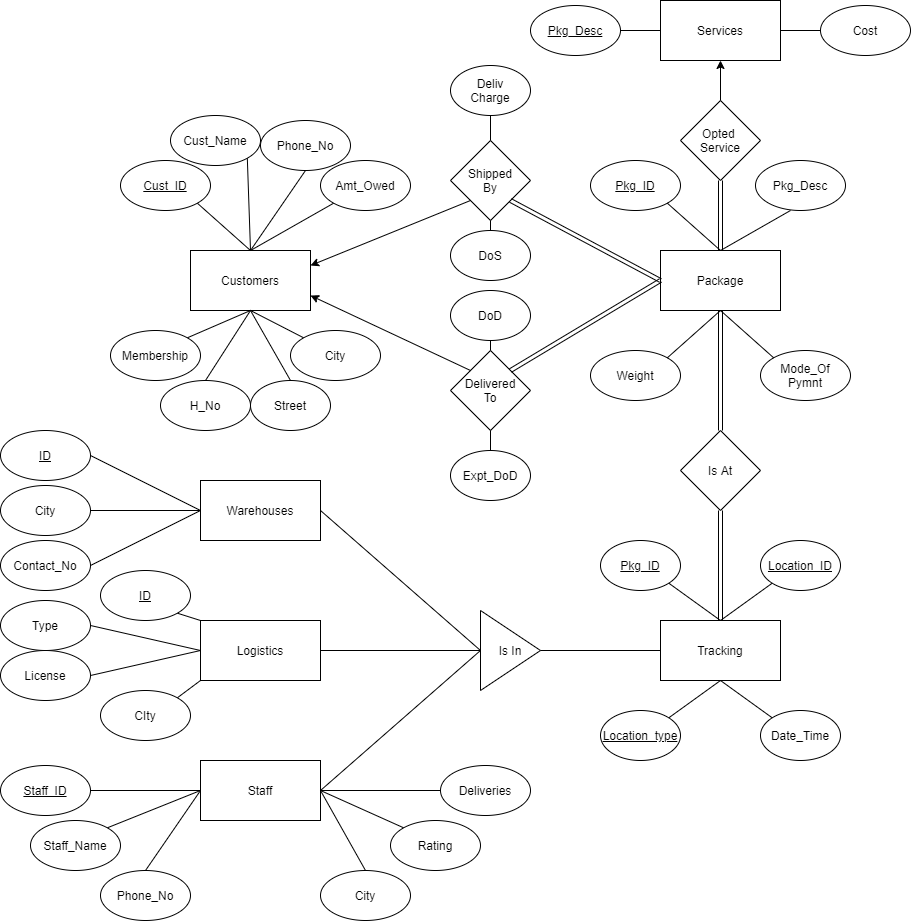
# Database

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**

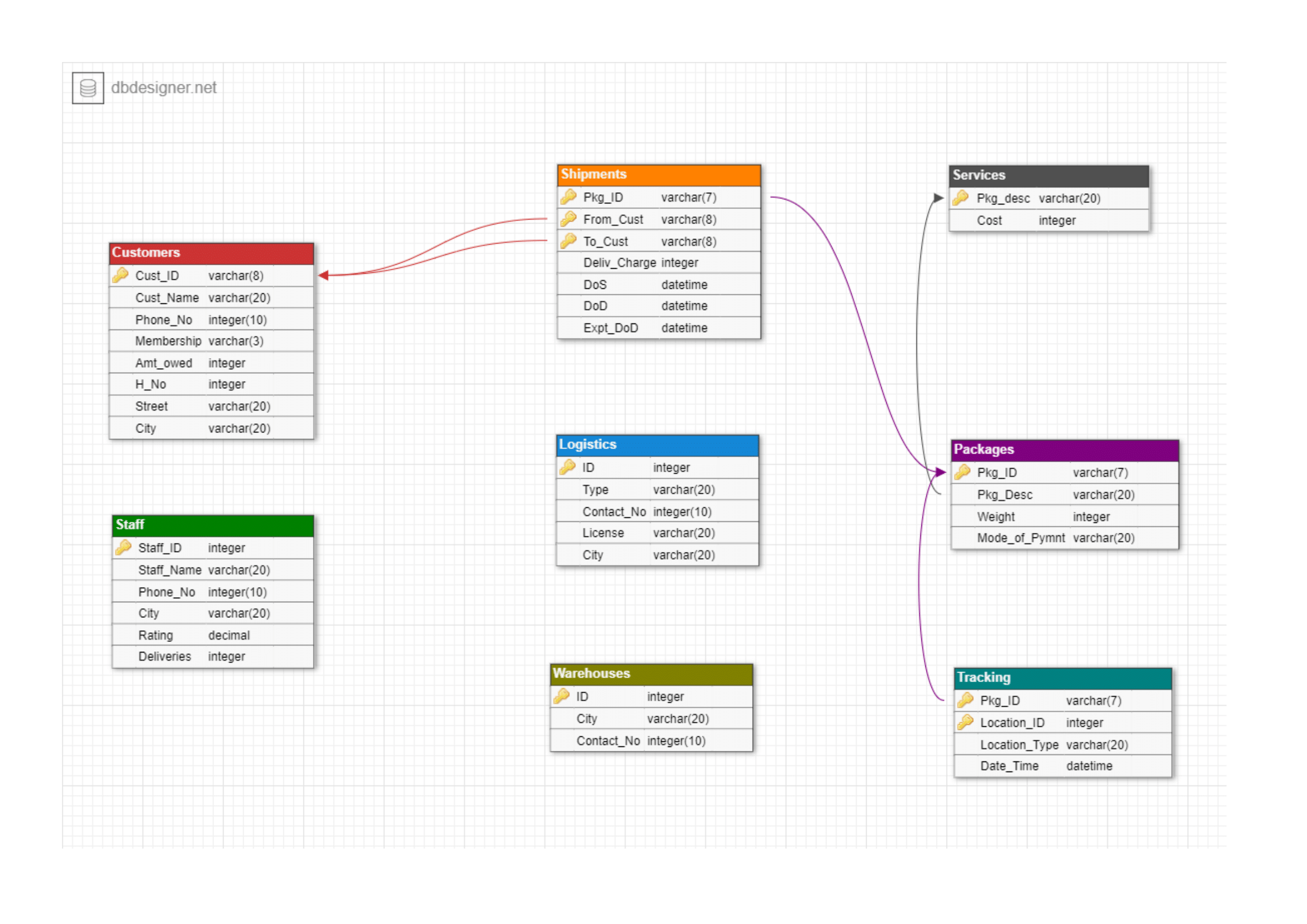
# E-R Diagram



**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



**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.**
2. **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"));

1. **Find all recipients who had a package on that truck at the time of the crash.**

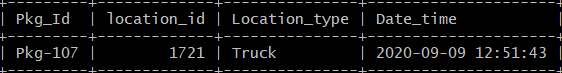
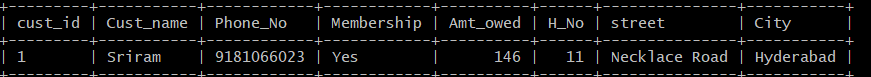
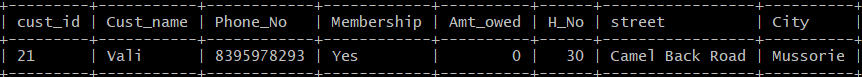
**Query:**

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"));

1. **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;



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

**Query:**

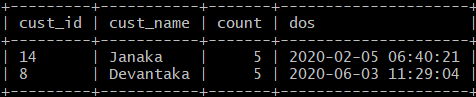
select cust\_id, cust\_name, count, dos from customers join

(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;



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

**Query:**

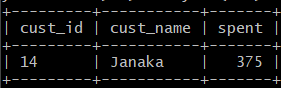
select cust\_id, cust\_name, spent from customers join

(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;



1. **Find the street with the most customers.**

**Query:**

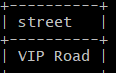
select street from customers join

(select count(\*) as count, cust\_id from customers group by street order by count(\*) desc) d

on customers.cust\_id

in (d.cust\_id)

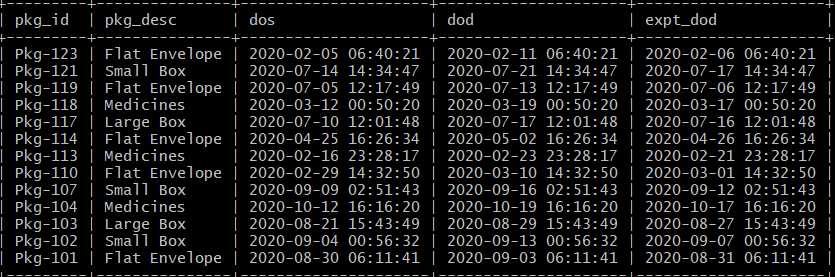
where count in (select max(count) from (select count(\*)as count from customers group by street) x);



1. **Find those packages that were not delivered within the promised time**

**Query:**

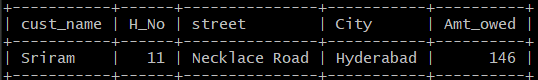
select packages.pkg\_id, pkg\_desc, dos, dod, expt\_dod from packages inner join shipments on packages.pkg\_id = shipments.pkg\_id where packages.pkg\_id in (select Pkg\_id from shipments where dod > expt\_dod);



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

**Query:**

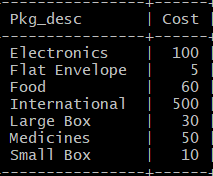
select cust\_name, H\_No, street, City, Amt\_owed from customers where cust\_id = 1;



1. **A bill listing charges by type of service**

**Query:**

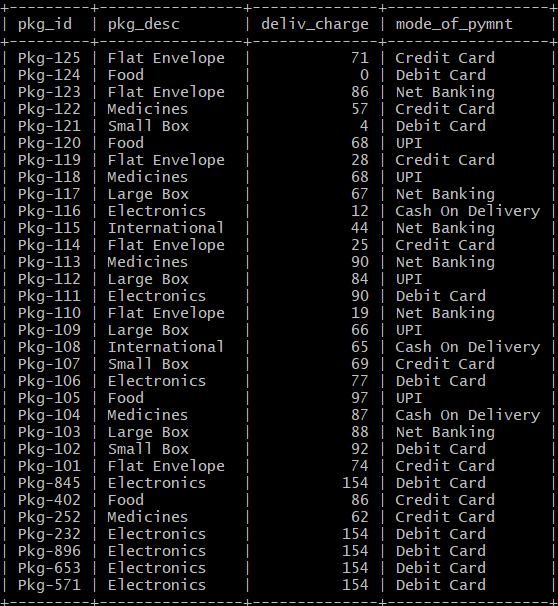
select \* from services;



1. **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;



# 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,

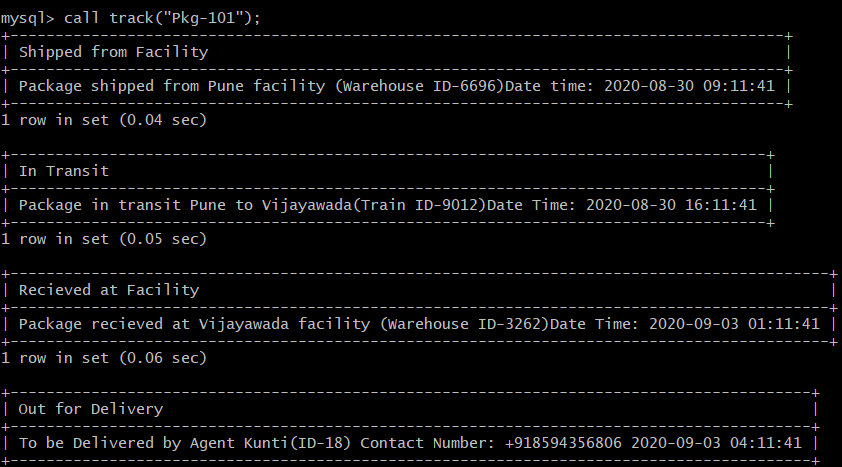
"(ID-", delivery\_id, ")",

" Contact Number: +91", delivery\_phn, " ", temp) as "Out for Delivery";

end $$

delimiter ;

## Output:



1. **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 ;

1. **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 ;

1. **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 ;

1. **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 ;

1. **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;