CS355 Mini Project Assignment – AUTUMN 2020

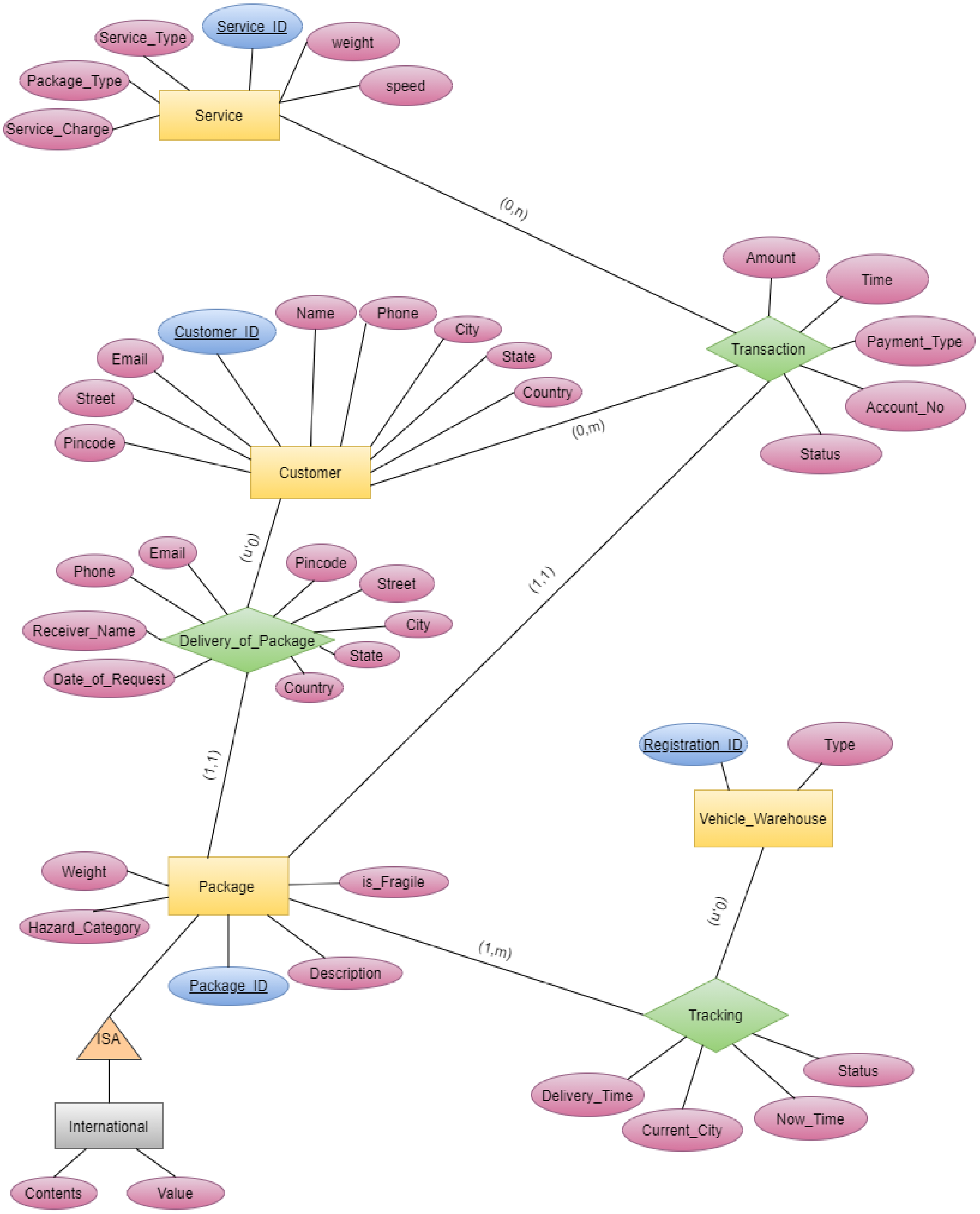
PACKAGE DELIVERY COMPANY DATABASE

Name - Chandrawanshi Mangesh Shivaji

Roll No. - 1801CS16

Date – 30/11/2020

ER DIAGRAM:



(min,max) notation is used for showing participation constraints of each entity in that specific relation

Entity Set 🟨

Relationship ◆

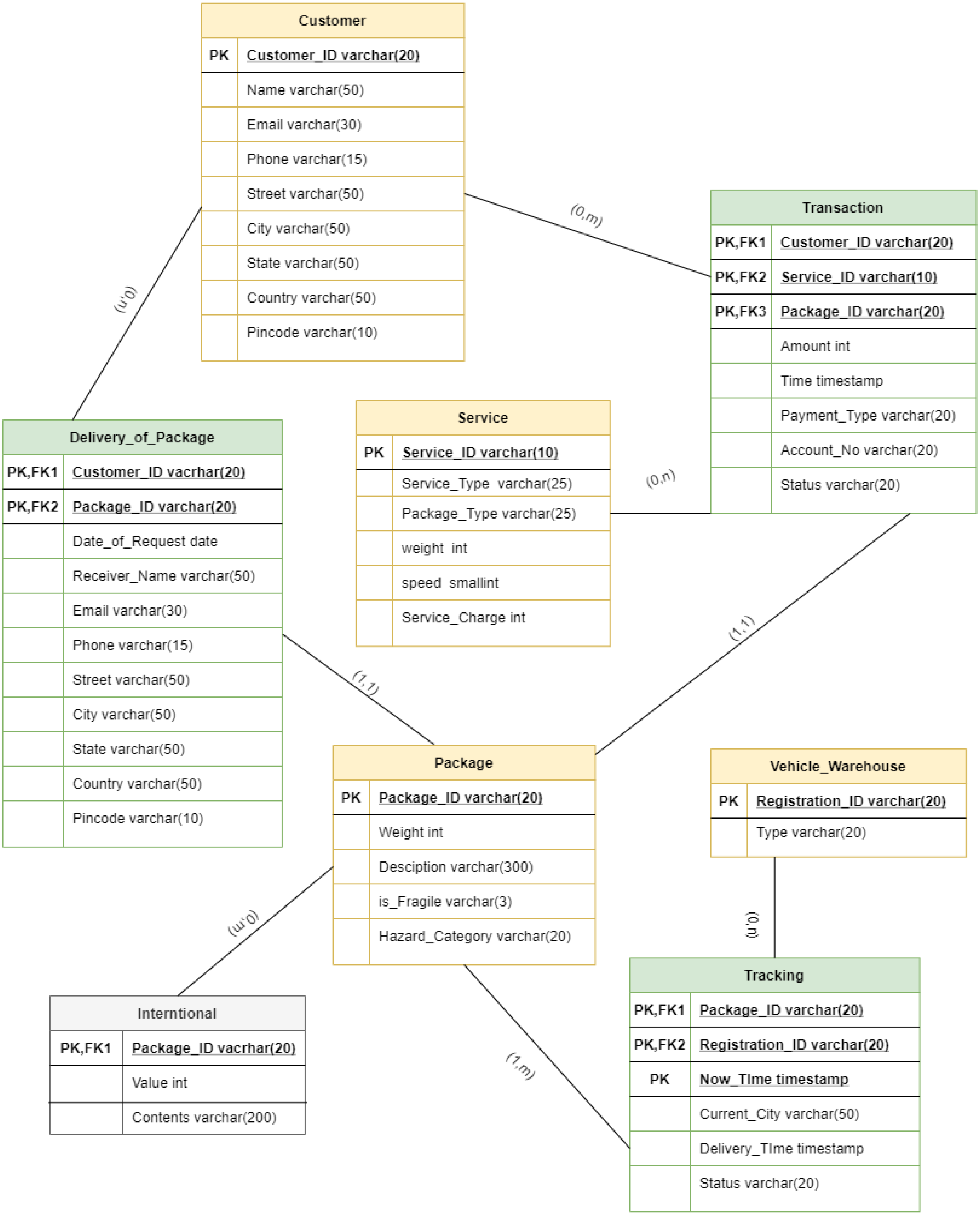
Primary Key 🔵

Normal Attribute ●

Weak Entity Set 🟧

Can also get Cardinality Constraint from (min,max) notation

RELATIONAL MODEL PICTORIAL REP.:



Customer	
PK	Customer_ID varchar(20)
	Name varchar(50)
	Email varchar(30)
	Phone varchar(15)
	Street varchar(50)
	City varchar(50)
	State varchar(50)
	Country varchar(50)
	Pincode varchar(10)

Transaction	
PK,FK1	Customer_ID varchar(20)
PK,FK2	Service_ID varchar(10)
PK,FK3	Package_ID varchar(20)
	Amount int
	Time timestamp
	Payment_Type varchar(20)
	Account_No varchar(20)
	Status varchar(20)

Delivery_of_Package	
PK,FK1	Customer_ID vacrhar(20)
PK,FK2	Package_ID varchar(20)
	Date_of_Request date
	Receiver_Name varchar(50)
	Email varchar(30)
	Phone varchar(15)
	Street varchar(50)
	City varchar(50)
	State varchar(50)
	Country varchar(50)
	Pincode varchar(10)

Service	
PK	Service_ID varchar(10)
	Service_Type varchar(25)
	Package_Type varchar(25)
	weight int
	speed smallint
	Service_Charge int

Package	
PK	Package_ID varchar(20)
	Weight int
	Desciption varchar(300)
	is_Fragile varchar(3)
	Hazard_Category varchar(20)

Vehicle_Warehouse	
PK	Registration_ID varchar(20)
	Type varchar(20)

Interntional	
PK,FK1	Package_ID vacrhar(20)
	Value int
	Contents varchar(200)

Tracking	
PK,FK1	Package_ID varchar(20)
PK,FK2	Registration_ID varchar(20)
PK	Now_TIme timestamp
	Current_City varchar(50)
	Delivery_TIme timestamp
	Status varchar(20)

CREATE ALL TABLES:

```
// is used to add description comments

// Sign in as root
mysql -u root -p
- Password

// Create Database for Package Delivery Company
create database Package_Delivery_Company;

use Package_Delivery_Company;

// Create all the required tables

create table Package(
        Package_ID varchar(20),
        Weight int unsigned,
        Description varchar(300),
        is_Fragile varchar(3) NOT NULL,
        Hazard_Category varchar(20),
        constraint Package_pk primary key (Package_ID)
);

create table International(
        Package_ID varchar(20),
        Value int unsigned,
        Contents varchar(200),
        constraint International_pk primary key (Package_ID),
        constraint International_fk1 foreign key (Package_ID) references Package(Package_ID)
);

create table Customer(
        Customer_ID varchar(20),
        Name varchar(50),
        Email varchar(30),
        Phone varchar(15),
        Street varchar(50),
        City varchar(50),
        State varchar(50),
        Country varchar(50),
        Pincode varchar(10),
        constraint Customer_pk primary key (Customer_ID)
);

create table Delivery_of_Package(
        Customer_ID varchar(20),
        Package_ID varchar(20),
        Date_of_Request date,
        Receiver_Name varchar(50),
        Email varchar(30),
        Phone varchar(15),
        Street varchar(50),
        City varchar(50),
        State varchar(50),
        Country varchar(50),
        Pincode varchar(10),
        constraint Delivery_of_Package_pk primary key (Customer_ID,Package_ID),
        constraint Delivery_of_Package_fk1 foreign key (Customer_ID) references Customer(Customer_ID),
        constraint Delivery_of_Package_fk2 foreign key (Package_ID) references Package(Package_ID)
);

create table Service(
        Service_ID varchar(10),
        Service_Type varchar(25),
        Package_Type varchar(25),
        Weight int unsigned,
        Speed smallint unsigned,
        Service_Charge int,
        constraint Service_pk primary key (Service_ID)
);

create table Transactions(
        Customer_ID varchar(20),
        Service_ID varchar(10),
        Package_ID varchar(20),
        Amount int unsigned,
        Time timestamp,
        Payment_Type varchar(20),
        Account_No varchar(20),
        Status varchar(20),
        constraint Transactions_pk primary key (Customer_ID,Service_ID,Package_ID),
        constraint Transactions_fk1 foreign key (Customer_ID) references Customer(Customer_ID),
        constraint Transactions_fk2 foreign key (Package_ID) references Package(Package_ID),
        constraint Transactions_fk3 foreign key (Service_ID) references Service(Service_ID)
);

create table Vehicle_Warehouse(
        Registration_ID varchar(20),
        Type varchar(20),
        constraint Vehicle_Warehouse_pk primary key (Registration_ID)
);


create table Tracking(
        Package_ID varchar(20),
        Registration_ID varchar(20),
        Now_Time timestamp,
        Current_City varchar(50),
        Delivery_Time timestamp,
        Status varchar(20),
        constraint Tracking_pk primary key (Package_ID,Registration_ID,Now_Time),
        constraint Tracking_fk1 foreign key (Registration_ID) references Vehicle_Warehouse(Registration_ID),
        constraint Tracking_fk2 foreign key (Package_ID) references Package(Package_ID)
);
```

FILL ALL TABLES USING PROCEDURES:

```
delimiter $$

// Procedure to fill Package table

create procedure procPkgDummyData(in n int)
begin
    declare Package_ID1 varchar(20);
    declare Weight1 int unsigned;
    declare Description1 varchar(300);
    declare is_Fragile1 varchar(3);
    declare Hazard_Category1 varchar(20);
    declare iter int;
    declare chk int;
    declare random_num int;

    set iter = 0;

    while(iter < n) do

        set Package_ID1 = concat('pkg_',lpad(conv(floor(rand()*pow(36,8)), 10, 36), 8,
0),lpad(conv(floor(rand()*pow(36,8)), 10, 36), 8, 0));

        set Weight1 = floor(rand()*999999) + 1;
        set Description1 =
concat('desc_',substring('ABCDEFGHIJKLMNOPQRSTUVWXYZ',floor(10*rand() + 1),10));

        set random_num = floor(rand()*4 + 1);

        if (random_num = 4) then
            set is_Fragile1 = 'YES';
        else
            set is_Fragile1 = 'NO';
        end if;

        set random_num = floor(rand()*1000 + 1);

        if (random_num = 250) then
            set Hazard_Category1 = 'EXPLOSIVE';
        elseif (random_num = 500) then
            set Hazard_Category1 = 'FlAMMABLE';
        elseif (random_num = 750) then
            set Hazard_Category1 = 'POISONOUS';
        elseif (random_num = 1000) then
            set Hazard_Category1 = 'RADIOACTIVE';
        else
            set Hazard_Category1 = 'NONE';
        end if;

        select exists (
            select Package_ID
            from Package
            where Package_ID = Package_ID1
        ) into chk;

        if (chk = 0) then
            insert into package
                values(Package_ID1,Weight1,Description1,is_Fragile1,Hazard_Category1);
            set iter = iter + 1;
        end if;

    end while;
end $$

delimiter ;

call procPkgDummyData(1000);
```

// **Procedure to fill international table**

```
delimiter $$

create procedure procIntrDummyData(in n int)
begin
    declare Package_ID1 varchar(20);
    declare Value1 int unsigned;
    declare Contents1 varchar(200);
    declare iter int;
    declare chk int;
    declare random_num int;

    set iter = 0;

    while(iter < n) do

        select Package_ID
        into Package_ID1
        from Package
        Order by rand()
        limit 1;

        set Value1 = floor(rand()*9999) + 1;
        set Contents1 =
concat('desc_',substring('ABCDEFGHIJKLMNOPQRSTUVWXYZ',floor(10*rand() + 1),10));

        select exists (
            select Package_ID
            from International
            where Package_ID = Package_ID1
        ) into chk;

        if (chk = 0) then
            insert into International
                values(Package_ID1,Value1,Contents1);
            set iter = iter + 1;
        end if;
        end while;
end $$

delimiter ;

call procIntrDummyData(50);
```

// **Procedure to fill Customer table**

```
delimiter $$

create procedure procCustDummyData(in n int)
begin
        declare Customer_ID1 varchar(20);
        declare Name1 varchar(50);
        declare Email1 varchar(30);
        declare Phone1 varchar(15);
        declare Street1 varchar(50);
        declare City1 varchar(50);
        declare State1 varchar(50);
        declare Country1 varchar(50);
        declare Pincode1 varchar(10);
        declare iter int;
        declare chk int;
        declare random_num int;

        set iter = 0;

        while(iter < n) do

                set Customer_ID1 = concat('cus_',lpad(conv(floor(rand()*pow(36,8)), 10, 36), 8, 0),lpad(conv(floor(rand()*pow(36,8)), 10, 36), 8, 0));

                select exists (
                        select Customer_ID
                        from Customer
                        where Customer_ID = Customer_ID1
                ) into chk;

                set Name1 = substring('ABCDEFabcRSTUVdefghijklMNOPQWXYZmnopqrsGHIJKLtuvwxyz',floor(rand()*25)+1,floor(rand()*25+1));
                set Email1 = concat(substring('ABCDEFabcRSTUVdefghijklMNOPQWXYZmnopqrsGHIJKLtuvwxyz',floor(rand()*25)+1,floor(rand()*10)+1),'@domain.com');
                set Phone1 = lpad(floor(rand()*10000000000),10,'0');
                set Street1 = substring('ABCDEFabcRSTUVdefghijklMNOPQWXYZmnopqrsGHIJKLtuvwxyz',floor(rand()*25)+1,floor(rand()*20+1));
                set City1 = substring('ABCDEFabcRSTUVdefghijklMNOPQWXYZmnopqrsGHIJKLtuvwxyz',floor(rand()*25)+1,floor(rand()*6)+1);
        set State1 =
substring('ABCDEFabcRSTUVdefghijklMNOPQWXYZmnopqrsGHIJKLtuvwxyz',floor(rand()*25)+1,floor(rand()
*7)+1);
        set Country1 =
substring('ABCDEFabcRSTUVdefghijklMNOPQWXYZmnopqrsGHIJKLtuvwxyz',floor(rand()*25)+1,floor(rand()
*9)+1);
        set Pincode1 = lpad(floor(rand()*10000000000),8,'0');

        if (chk = 0) then
            insert into Customer

    values(Customer_ID1,Name1,Email1,Phone1,Street1,City1,State1,Country1,Pincode1);
            set iter = iter + 1;
        end if;

    end while;
end $$

delimiter ;

call procCustDummyData(1000);
```

// **Procedure to fill Delivery_of_Package table**

```
delimiter $$

create procedure procDoPDummyData(in n int)
begin
    declare Customer_ID1 varchar(20);
    declare Package_ID1 varchar(20);
    declare Date_of_Request date;
    declare Receiver_Name1 varchar(50);
    declare Email1 varchar(30);
    declare Phone1 varchar(15);
    declare Street1 varchar(50);
    declare City1 varchar(50);
    declare State1 varchar(50);
    declare Country1 varchar(50);
    declare Pincode1 varchar(10);
    declare iter int;
    declare chk int;
    declare random_num int;

    set iter = 0;

    while(iter < n) do

        select Package_ID
        into Package_ID1
        from Package
        Order by rand()
        limit 1;
        select Customer_ID
        into Customer_ID1
        from Customer
        Order by rand()
        limit 1;

        select exists (
            select Customer_ID,Package_ID
            from Delivery_of_Package
            where Package_ID = Package_ID1 and Customer_ID = Customer_ID1
        ) into chk;

        SELECT FROM_UNIXTIME(UNIX_TIMESTAMP('2018-01-01 00:00:000') + FLOOR(0 +
(RAND() * 63072000))) INTO Date_of_Request;
        set Receiver_Name1 =
substring('ABCDEFabcRSTUVdefghijklMNOPQWXYZmnopqrsGHIJKLtuvwxyz',floor(rand()*25)+1,floor(rand()
*25)+1);
        set Email1 =
concat(substring('ABCDEFabcRSTUVdefghijklMNOPQWXYZmnopqrsGHIJKLtuvwxyz',floor(rand()*25)+1,floo
r(rand()*10)+1),'@domain.com');
        set Phone1 = lpad(floor(rand()*10000000000),10,'0');
        set Street1 =
substring('ABCDEFabcRSTUVdefghijklMNOPQWXYZmnopqrsGHIJKLtuvwxyz',floor(rand()*25)+1,floor(rand()
*20)+1);
        set City1 =
substring('ABCDEFabcRSTUVdefghijklMNOPQWXYZmnopqrsGHIJKLtuvwxyz',floor(rand()*25)+1,floor(rand()
*6)+1);
        set State1 =
substring('ABCDEFabcRSTUVdefghijklMNOPQWXYZmnopqrsGHIJKLtuvwxyz',floor(rand()*25)+1,floor(rand()
*7)+1);
        set Country1 =
substring('ABCDEFabcRSTUVdefghijklMNOPQWXYZmnopqrsGHIJKLtuvwxyz',floor(rand()*25)+1,floor(rand()
*9)+1);
        set Pincode1 = lpad(floor(rand()*10000000000),8,'0');

        if (chk = 0) then
            insert into Delivery_of_Package

    values(Customer_ID1,Package_ID1,Date_of_Request,Receiver_Name1,Email1,Phone1,Street1,City1,
State1,Country1,Pincode1);
            set iter = iter + 1;
        end if;

    end while;
end $$

delimiter ;

call procDoPDummyData(1000);
```

// **Procedure to fill Service Table**

```
delimiter $$
```

```
create procedure procServDummyData(in n int)
begin
        declare Service_ID1 varchar(10);
        declare Service_Type1 varchar(25);
        declare Package_Type1 varchar(25);
        declare Weight1 int;
        declare Service_Charge1 int;
        declare Speed1 smallint;
        declare iter int;
        declare chk int;
        declare random_num int;

        set iter = 0;

        while(iter < n) do

                set Service_ID1 = concat('s_',lpad(conv(floor(rand()*pow(36,8)), 10, 36), 8, 0));

                select exists (
                        select Service_ID
                        from Service
                        where Service_ID = Service_ID1
                ) into chk;

                set random_num = floor(rand()*3);
                if(random_num < 1) then
                        set Service_Type1 = 'Prepaid';
                elseif(random_num < 2) then
                        set Service_Type1 = 'Postpaid';
                else
                        set Service_Type1 = 'Other';
                end if;

                set random_num = floor(rand()*4);
                if(random_num < 1) then
                        set Package_Type1 = 'Small Box';
                elseif(random_num < 2) then
                        set Package_Type1 = 'Large Box';
                elseif(random_num < 3) then
                        set Package_Type1 = 'Medium Box';
                else
                        set Package_Type1 = 'Flat Envelope';
                end if;

                set Weight1 = floor(rand()*999999) + 1;

                set Service_Charge1 = floor(rand()*10000) + 1;

                set Speed1 = floor(rand()*14) + 1;

                if (chk = 0) then
                        insert into Service
                values(Service_ID1,Service_Type1,Package_Type1,Weight1,Speed1,
Service_Charge1);
            set iter = iter + 1;
        end if;

    end while;

end$$
delimiter ;

call procServDummyData(500);
```

// **Procedure to fill Transactions Table**

```
delimiter $$

create procedure procTransDummyData(in n int)
begin
    declare Customer_ID1 varchar(20);
    declare Service_ID1 varchar(10);
    declare Package_ID1 varchar(20);
    declare Amount1 int;
    declare Time1 timestamp;
    declare Payment_Type1 varchar(20);
    declare Account_No1 varchar(20);
    declare Status1 varchar(20);
    declare Service_Charge1 int;
    declare Speed1 smallint;
    declare random_date date;
    declare iter int;
    declare chk int;
    declare random_num int;

    set iter = 0;

    while(iter < n) do

        select Package_ID
        into Package_ID1
        from Package
        Order by rand()
        limit 1;

        select Service_ID
        into Service_ID1
        from Service
        Order by rand()
        limit 1;

        select Customer_ID
        into Customer_ID1
        from Customer
        Order by rand()
        limit 1;

        select exists (
            select Customer_ID,Service_ID,Package_ID
            from Transactions
            where Package_ID = Package_ID1 and Service_ID1 = Service_ID and Customer_ID =
Customer_ID1
        ) into chk;

        select Service_Charge into Amount1
        from Service
        where Service_ID = Service_ID1;

        set random_num = floor(rand()*4);
        if(random_num < 1) then
            set Payment_Type1 = 'Credit Card';
        elseif(random_num < 2) then
            set Payment_Type1 = 'Debit Card';
        elseif(random_num < 3) then
            set Payment_Type1 = 'Internet Banking';
        else
            set Payment_Type1 = 'UPI';
        end if;

        select exists(
            select Date_of_Request
            from Delivery_of_Package
            where Customer_ID = Customer_ID1 and Package_ID = Package_ID1
        ) into chk;

        set Account_No1 = floor(rand()*999999999) + 1;

        set Status1 = 'Successful';

        select from_unixtime(unix_timestamp('2018-01-01 01:00:00')+floor(rand()*31536000)) into
Time1;

        if (chk = 0) then
            insert into Transactions

    values(Customer_ID1,Service_ID1,Package_ID1,Amount1,Time1,Payment_Type1,Account_No1,Statu
s1);
            set iter = iter + 1;
        end if;

    end while;

end$$
delimiter ;

call procTransDummyData(1000);
```

// **Procedure to fill Vehicle_Warehouse Table**

```
delimiter $$

create procedure procVeWaDummyData(in n int)
begin
    declare Registration_ID1 varchar(20);
    declare Type1 varchar(20);
    declare iter int;
    declare chk int;
    declare random_num int;

    set iter = 0;

    while(iter < n) do

        set Registration_ID1 = concat('reg_',lpad(conv(floor(rand()*pow(36,8)), 10, 36), 8,
0),lpad(conv(floor(rand()*pow(36,8)), 10, 36), 8, 0));

        select exists (
            select Registration_ID
            from Vehicle_Warehouse
            where Registration_ID = Registration_ID1
        ) into chk;

        set random_num = floor(rand()*4);
        if(random_num < 1) then
            set Type1 = 'Truck';
        elseif(random_num < 2) then
            set Type1 = 'Plane';
        elseif(random_num < 3) then
            set Type1 = 'Train';
        else
            set Type1 = 'Warehouse';
        end if;

        if (chk = 0) then
            insert into Vehicle_Warehouse
                values(Registration_ID1,Type1);
            set iter = iter + 1;
        end if;

    end while;

end$$
delimiter ;

call procVeWaDummyData(2000);
```

// **Procedure to fill Tracking Table**

```
delimiter $$
```

```
create procedure procTracDummyData(in n int)
begin

    declare Package_ID1 varchar(20);
    declare Registration_ID1 varchar(20);
    declare Now_Time1 timestamp;
    declare Current_City1 varchar(50);
    declare Delivery_Time1 timestamp;
    declare Status1 varchar(20);
    declare Start_Time timestamp;
    declare iter int;
    declare iter1 int;
    declare chk int;
    declare random_num int;

    set iter = 0;

    while (iter < n) do

        select Package_ID
        into Package_ID1
        from Package
        Order by rand()
        limit 1;

        select Registration_ID
        into Registration_ID1
        from Vehicle_Warehouse
        Order by rand()
        limit 1;

        set Delivery_Time1 = NULL;

        set Status1 = 'Out for Delivery';

        select TIMESTAMP(Date_of_Request) into Start_Time from Delivery_of_Package where
Package_ID = Package_ID1 limit 1;

        set iter1 = 0;

        set random_num = floor(rand()*5 + 1);

        while (iter1 < random_num) do

            set Current_City1 =
substring('ABCDEFabcRSTUVdefghijklMNOPQWXYZmnopqrsGHIJKLtuvwxyz',floor(rand()*25)+1,floor(rand()
*6)+1);

            select Registration_ID
            into Registration_ID1
            from Vehicle_Warehouse
            Order by rand()
            limit 1;

            select from_unixtime(unix_timestamp(Start_Time) + floor(500 + (rand() * 172800))) into
Now_Time1;

            set Start_Time = Now_Time1;

            set Delivery_Time1 = FROM_UNIXTIME(UNIX_TIMESTAMP(Start_Time) + FLOOR(0 +
(RAND() * 172800)));

            if (iter1 = random_num-1) then
                set Status1 = 'Delivered';
                set Delivery_Time1 = Start_Time;
            end if;

            insert into Tracking

    values(Package_ID1,Registration_ID1,Now_Time1,Current_City1,Delivery_Time1,Status1);

            set iter1 = iter1 + 1;

        end while;

    set iter = iter + 1;

    end while;

end $$

delimiter ;

call procTracDummyData(1000);
```

// **Size of All Tables (Can be Modified/Increased using above procedures)**

```
mysql> select count(*) from customer;
+----------+
| count(*) |
+----------+
|     1000 |
+----------+
1 row in set (0.20 sec)

mysql> select count(*) from delivery_of_package;
+----------+
| count(*) |
+----------+
|     1000 |
+----------+
1 row in set (0.03 sec)

mysql> select count(*) from international;
+----------+
| count(*) |
+----------+
|       50 |
+----------+
1 row in set (0.01 sec)

mysql>  select count(*) from package;
+----------+
| count(*) |
+----------+
|     1000 |
+----------+
1 row in set (0.00 sec)

mysql>  select count(*) from service;
+----------+
| count(*) |
+----------+
|      500 |
+----------+
1 row in set (0.04 sec)

mysql>  select count(*) from tracking;
+----------+
| count(*) |
+----------+
|     2967 |
+----------+
1 row in set (0.00 sec)

mysql>  select count(*) from transactions;
+----------+
| count(*) |
+----------+
|     1000 |
+----------+
1 row in set (0.14 sec)

mysql>  select count(*) from vehicle_warehouse;
+----------+
| count(*) |
+----------+
|     2000 |
+----------+
1 row in set (0.14 sec)

mysql>  select count(*) from vehicle_warehouse;
+----------+
| count(*) |
+----------+
|     2000 |
+----------+
1 row in set (0.00 sec)
```

// **Export Data in all Tables to csv files**

```
SELECT * INTO outfile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Package.csv'
fields terminated by ','
lines terminated by '\n' FROM Package;

SELECT * INTO outfile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/International.csv'
fields terminated by ','
lines terminated by '\n' FROM International;

SELECT * INTO outfile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Customer.csv'
fields terminated by ','
lines terminated by '\n' FROM Customer;

SELECT * INTO outfile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Delivery_of_Package.csv'
fields terminated by ','
lines terminated by '\n' FROM Delivery_of_Package;

SELECT * INTO outfile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Service.csv'
fields terminated by ','
lines terminated by '\n' FROM Service;

SELECT * INTO outfile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Transactions.csv'
fields terminated by ','
lines terminated by '\n' FROM Transactions;

SELECT * INTO outfile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Vehicle_Warehouse.csv'
fields terminated by ','
lines terminated by '\n' FROM Vehicle_Warehouse;

SELECT * INTO outfile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Tracking.csv'
fields terminated by ','
lines terminated by '\n' FROM Tracking;
```

// Given Queries

Assume a delivery truck (say truck no 1721) is destroyed in a crash.

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

```
select Name from Customer where Customer_ID in(
select Customer_ID from Delivery_of_Package where Package_ID in(
select distinct Package_ID from Tracking where Registration_ID = '1721' and status != 'Delivered'));
```

```
// For My Data
select Name from Customer where Customer_ID in(
select Customer_ID from Delivery_of_Package where Package_ID in(
select distinct Package_ID from Tracking where Registration_ID = 'reg_75URYFEQ0R7HF109' and status !=
'Delivered'));
```

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

```
select Receiver_Name from Delivery_of_Package where Package_ID in(
select distinct Package_ID from Tracking where Registration_ID='1721' and status != 'Delivered');

// For My Data
select Receiver_Name from Delivery_of_Package where Package_ID in(
select distinct Package_ID from Tracking where Registration_ID='reg_75URYFEQ0R7HF109' and status !=
'Delivered');
```

##Find the last successful delivery by that truck prior to the crash.

```
select Package_ID from Tracking where Delivery_Time in(
select max(Delivery_Time) from Tracking where Registration_ID='1721' and status='Delivered');


// For My Data
select Package_ID from Tracking where Delivery_Time in(
select max(Delivery_Time) from Tracking where Registration_ID='reg_75URYFEQ0R7HF109' and
status='Delivered');
```

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

```
select Customer_ID from Delivery_of_Package where year(Date_of_Request)=2019 group by Customer_ID
order by count(*) desc limit 1;
```

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

```
select Customer_ID from Transactions where year(Time)=2019 group by Customer_ID order by sum(Amount)
desc limit 1;
```

##Find the street with the most customers.

```
select Street from Customer group by Street order by count(*) desc limit 1;
```

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

```
select * from Package where Package_ID in(
select Transactions.Package_ID from Transactions, Tracking, Delivery_of_Package, Service where
Transactions.Package_ID = Tracking.Package_ID and Transactions.Service_ID = Service.Service_ID and
Delivery_of_Package.Package_ID = Transactions.Package_ID and
date_add(Delivery_of_Package.Date_of_Request,interval Speed day) < Tracking.Now_Time);
```

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

```
select distinct Customer.Customer_ID, Customer.Name, Transactions.Amount, Customer.Pincode,
Customer.Street, Customer.City, Customer.State, Customer.Country, Customer.Email, Customer.Phone
from Customer, Transactions
where Customer.Customer_ID = Transactions.Customer_ID and Transactions.Service_ID in(
	select Service_ID from Service where Service_Type = 'Postpaid')
union
select distinct Customer.Customer_ID, Customer.Name, 0, Customer.Pincode, Customer.Street,
Customer.City, Customer.State, Customer.Country, Customer.Email, Customer.Phone
from Customer, Transactions
where Customer.Customer_ID = Transactions.Customer_ID and Transactions.Service_ID in(
	select Service_ID from Service where Service_Type != 'Postpaid')
union
select distinct Customer.Customer_ID, Customer.Name, 0, Customer.Pincode, Customer.Street,
Customer.City, Customer.State, Customer.Country, Customer.Email, Customer.Phone
from Customer where Customer.Customer_ID not in (select Customer_ID from Transactions);
```

##A bill listing charges by type of service.

```
Select * from Service;
```

##An itemize billing listing each individual shipment and the charges for it.

```
Select * from Transactions;
```

EXTRA QUERIES

## Find Package ID all the non hazardous international shipments

```
select International.Package_ID
from International,Package
where International.Package_ID = Package.Package_ID and Hazard_Category = 'None';
```

## Find the name of all customers who have shipped international packages.

```
select Name from Customer
where Customer_ID IN(
          select Customer_ID
          from Delivery_of_Package
          where Package_ID IN
                  (SELECT Package_ID
                   FROM International));
```

ADDING INDEX

As in most of the cases for querying we are using or requiring only primary key attributes, till now we did not find any requirement for indices on other attributes.There can be an index on Receiver's Name Attribute as it is possible to have useful queries involving receiver details. So, I have created an index on Receiver Name in Delivery_of_Package Relationship Table

```
create index idx_DoP1 on delivery_of_package(Receiver_Name);

## Query involving Reciever Name
Find the Name of all recievers of packages weighing less than 1000 units

select Receiver_Name
from delivery_of_package,Package
where delivery_of_package.Package_ID = Package.Package_ID and Package.Weight < 1000;

+----------------+
| Receiver_Name  |
+----------------+
| jklMN          |
| TUVdefghi      |
| fghijklMNOPQWX |
| lMNOPQWX       |
+----------------+
4 rows in set (0.01 sec)

mysql> create index idx_DoP1 on delivery_of_package(Receiver_Name);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select Receiver_Name
    -> from delivery_of_package,Package
    -> where delivery_of_package.Package_ID = Package.Package_ID and Package.Weight < 1000;
+----------------+
| Receiver_Name  |
+----------------+
| lMNOPQWX       |
| TUVdefghi      |
| jklMN          |
| fghijklMNOPQWX |
+----------------+
4 rows in set (0.00 sec)
```

So, Query Speeds up a little.