**CS355 Mini Project Assignment – AUTUMN 2020**

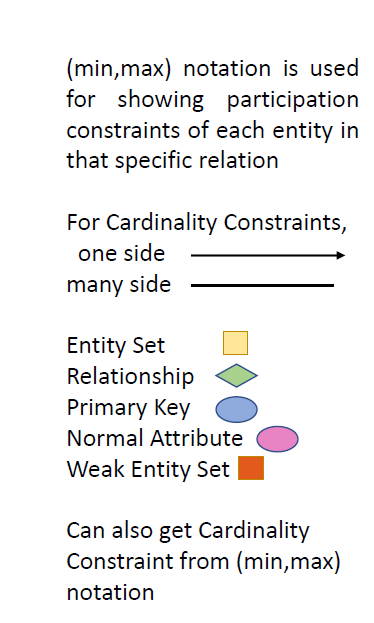
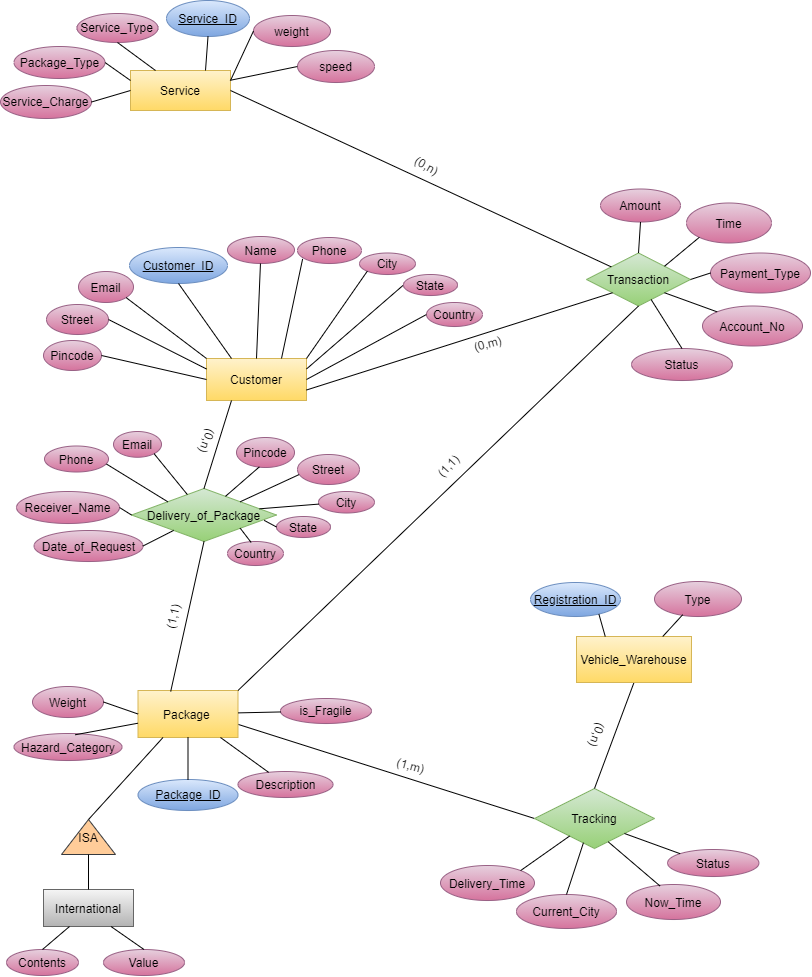
**PACKAGE DELIVERY COMPANY DATABASE**

**Name - Chandrawanshi Mangesh Shivaji**

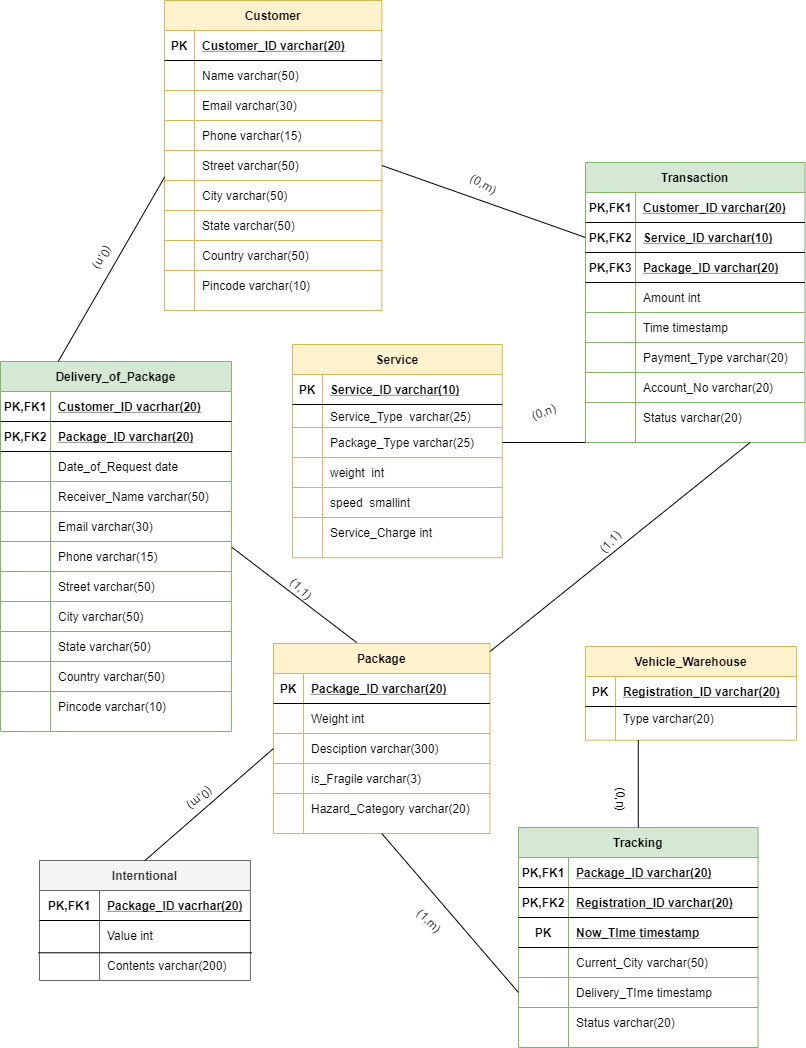
**Roll No. - 1801CS16**

**Date – 30/11/2020**

**ER DIAGRAM:**



**RELATIONAL MODEL PICTORIAL REP.:**



**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,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 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,Status1);

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.